Tshwane University of Technology

We empower people

DEVELOPING A RELIABILITY BASED MODEL FOR SELECTING TEST PROBE RECEPTACLE INTERFACES

By

JACOB JACOBUS GREEFF

Submitted in partial fulfillment of the requirements for the degree

MAGISTER TECHNOLOGIAE: ELECTRICAL ENGINEERING

In the

Department of Electrical Engineering

FACULTY OF ENGINEERING AND THE BUILT ENVIRONMENT

TSHWANE UNIVERSITY OF TECHNOLOGY

Supervisor: Prof Y Hamam

Co-Supervisor: Mr R Aylward

November 2014

ABSTRACT

DEVELOPING A RELIABILITY BASED MODEL FOR SELECTING TEST PROBE RECEPTACLE INTERFACES

This project is concerned with the creation of a reliability based model that allows a designer to enter design requirements in the form of percentage importance on a number of design parameters and then the model outputs a recommendation as to what the best methodology would be to apply to the specific problem.

The model is created by doing an intense qualitative study of each of the different methodologies being evaluated in the model for different qualitative parameters including cost, ease of signal conditioning implementation, maintainability etc. as well as creating a system that will allow a number of samples of the methodologies being evaluated to be placed under severe stress and automatically tested over a period of time to allow for failures in time to be measured and from that a relative reliability ranking of the different methodologies to obtained.

This qualitative and quantitative approach to creating the selection model, as well as the design techniques employed in the creation of the relative reliability tester support the need for further work in the refinement of the model, as well as in the opportunities for further exploring reactive measurements that can be taken with the circuitry designed for this project.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and appreciation to: My supervisor, Prof. Hamam and my co-supervisor Mr. Aylward for their guidance and assistance in getting this project together, as well as the Tshwane University of Techology for the financial assistance.

I would also like to thank Mr. Adrian Storie for his guidance and support during the duration of the project, as well as Mr. Hannes Van Niekerk for his mechanical design input. I would also like to thank my partners Helen and Stella as well as my parents Japie and Lee for their unwavering emotional support during the duration of this project. Without you, this would not have been possible.

DECLARATION BY CANDIDATE

"I hereby declare that the thesis submitted for the degree MTech: Electrical Engineering,at the Tshwane University of Technology, is my own original work and has not been previously submitted to any other institution of higher education. I further declare all sources cited or quoted are indicated and acknowledged by means of a comprehensive list of references".

J.J. Greeff

Copyright© Tshwane University of Technology 2014

CONTENTS PAGE

ABSTRACT i

ACKNOWLEDGEMENTS ii

DECLARATION iii

CONTENTS iv

LIST OF TABLES x

LIST OF FIGURES xi

LIST OF FORMULAE xiv

GLOSSARY xv

CHAPTER 1

1 **INTRODUCTION AND BACKGROUND** 1

1.1 **TESTING IN ELECTRONIC PRODUCT MANUFACTURING** 1

1.2 **PROBLEM DESCRIPTION** 2

1.3 **PROBLEM STATEMENT** 3

1.3.1 **Sub problem 1** 4

1.3.2 **Sub problem 2** 4

1.3.3 **Sub problem 3** 4

1.4 **AIM OF THE STUDY** 4

1.5 **HYPOTHESIES** 4

1.5.1 **Hypothesis 1** 5

1.5.2 **Hypothesis 2** 5

1.5.3 **Hypothesis 3** 5

1.6 **DELIMINATION** 5

1.7 **ASSUMPTIONS** 6

1.8 **MODEL CREATION METHODOLOGY** 6

1.9 **VALUE OF THIS STUDY** 7

1.10 **OVERVIEW OF THE DISSERTATION** 8

1.11 **SUMMARY** 10

CHAPTER 2

2 **LITERATURE SURVEY AND METHODOLOGY OVERVIEW** 11

2.1 **TESTING AS PART OF THE MANUFACTURING PROCESS** 11

2.2 **ICT SYSTEMS** 15

2.3 **RELIABILITY AND RELIABILITY PREDICTIONS** 23

2.4 **ENVIRONMENTAL STRESS SCREENING** 26

2.4.1 **Burn in** 32

2.4.2 **Temperature cycling** 32

2.4.3 **Thermal shock** 34

2.4.4 **Mechanical shock and vibration** 34

2.4.5 **Electrical stress** 35

2.4.6 **Other stresses** 35

2.5 **ENVIRONMENTAL STRESS EFFECTIVENESS** 35

2.6 **SUMMARY** 37

CHAPTER 3

3 **CONCEPTUAL DESIGN AND CONSIDERATIONS** 38

3.1 **DEFINING FAILURE AND PROPER ASSEMBLY** 38

3.1.1 **Wire wrap method** 39

3.1.2 **Solder cup connection method** 41

3.1.3 **Crimp connection method** 42

3.1.4 **Back plane connection with soldered receptacles** 43

3.1.5 **Back plane connection with double ended receptacles** 44

3.1.6 **Back plane with receptacle soldered in place** 45

3.1.7 **PCB on Probe platen** 46

3.1.8 **Double PCB mount** 47

3.2 **FIXTURE FAILURE VERSUS UUT FAILURE** 48

3.3 **PRELIMINARY TEST SYSTEM DESIGN** 50

3.3.1 **Overview** 50

3.3.2 **Hardware and equipment** 52

3.3.3 **Software and data budget** 56

3.4 **PRELIMINARY MODEL DESIGN** 58

3.5 **SUMMARY** 60

CHAPTER 4

4 **DESIGN AND DATA COLLECTION** 61

4.1 **RELIABILITY DATA COLLECTION SYSTEM (HARDWARE)** 61

4.1.1 **Sample and hold circuitry** 61

4.1.2 T**emperature measurements** 63

4.1.3 **Thermal chamber driving** 65

4.1.4 **Control relays** 67

4.1.5 **Signal Generation** 68

4.1.6 **Interfacing** 68

4.1.7 **Pneumatic control** 69

4.2 **RELIABILITY DATA COLLECTION SYSTEM (SOFTWARE)** 70

4.2.1 **Labview VI design** 70

4.2.2 **Teststand sequence design** 72

4.2.3 **Running the application** 73

4.2.4 **Message flow during runtime** 75

4.2.5 **Drivers** 76

4.3 **QUALITATIVE PARAMETERS** 78

4.3.1 **Maintainability** 78

4.3.2 **Ease of Implementation of Signal conditioning** 79

4.3.3 **Likeliness of system change** 79

4.3.4 **Price** 80

4.3.5 **Implementation speed** 81

4.4 **QUALITATIVE PARAMETER DATA COLLECTION** 81

4.4.1 **Wire wrap method** 82

4.4.2 **Solder cup method** 83

4.4.3 **Crimp connection method** 84

4.4.4 **Back plane connection with soldered sockets method** 84

4.4.5 **Back plane connection with double ended receptacles method** 86

4.4.6 **Back plane with receptacle soldered in place method** 86

4.4.7 **PCB on Probe platen method** 87

4.4.8 **Double PCB mount method** 88

4.5 **QUALITATIVE PARAMETER DATA RANKING** 88

4.6 **PRICE DATA COLLECTION** 89

4.7 **SUMMARY** 91

LIST OF TABLES PAGE

Table 3.1: TDA DAQ Message structure (all fields 8 bit) 54

Table 3.2: Data collected for each test sequence 56

Table 4.1: Message structure 77

Table 4.2: Tool usage across methodologies 80

Table 4.3: Qualitative parameter ranking 89

Table 4.4: Component price table 90

Table 5.1: Test system downtime 93

Table 5.2: Temperature probe channels 95

Table 5.3: Potted versus unpotted temperature error 99

Table 5.4: Signal chain mapping 101

Table 5.5: Relative reliability rankings 113

LIST OF ILLUSTRATIONS PAGE

Illustration 1.1: Flow chart of project flow 9

Illustration 2.1: The idealised test generation process 12

Illustration 2.2: Example of a simplified manufacturing line(electronic product) 13

Illustration 2.3: Generic manufacturing test flow with test points 14

Illustration 2.4: ICT vendors by country 15

Illustration 2.5: Test probe tip types 20

Illustration 2.6: Proper probe and force cancellation placement 21

Illustration 2.7: Probe connection methodologies showing various options 22

Illustration 2.8: Bath tub curve 25

Illustration 2.9: Infancy mortality period 26

Illustration 2.10: Common temperature profile for a temperature cycle 33

Illustration 2.11: Hot and cold box set for thermal shock 34

Illustration 3.1: Test signal chain 38

Illustration 3.2: Wire wrap wrappost wiring 40

Illustration 3.3: Solder cup mounting 41

Illustration 3.4: Voltage drop test 42

Illustration 3.5: Receptacle mechanical drawings 44

Illustration 3.6: Double ended receptacle with spring rear end 45

Illustration 3.7: Back plane with receptacle soldered in place 45

Illustration 3.8: PCB on probe platen mount 46

Illustration 3.9: Double PCB mount 48

Illustration 3.10: Rudimentary test system 49

Illustration 3.11: Relative reliability tester high level layout 52

Illustration 3.12: Proposed heat flow in the chamber 53

Illustration 3.13: Proposed signal chain and wiring design 55

Illustration 3.14: Proposed software test flow 57

Illustration 4.1: Sample and hold circuit 61

Illustration 4.2: Thermal probe design 64

Illustration 4.3: Temperature Probe placement in chamber 64

Illustration 4.4: Heating array in chamber 66

Illustration 4.5: Peltier cooler placement between heat syncs 67

Illustration 4.6: UUT Board indicating board markings and spacing 69

Illustration 4.7: Chamber housing receptacles 70

Illustration 4.8: Sample and hold custom electronics 71

Illustration 4.9: Mechanics outer view with power supplies 71

Illustration 4.10: Labview/Teststand overhead architecture 72

Illustration 4.11: User interface of custom VI 72

Illustration 4.12: Application life cycle overhead view 75

Illustration 4.13: Console generated event flow 78

Illustration 4.14: Sequence generated message flow 78

Illustration 4.15: Wire wrap example 84

Illustration 5.1: Ambient temperature graph over full test 96

Illustration 5.2: Inner chamber temperature, unpotted probe only 98

Illustration 5.3: Inner chamber temperature, potted probes 99

Illustration 5.4: Potted versus unpotted plots showing thermal inertia 100

Illustration 5.5: Voltages measured on channel 1 over the course of the test 103

Illustration 5.6: Crimped channel failure on test sequence 7 (channel 1) 103

Illustration 5.7: Voltages measured on channel 2 over the course of the test 105

Illustration 5.8: Crimped channel failure on test sequence 7 (channel 2) 106

Illustration 5.9: Voltages measured on channel 4 over the course of the test 107

Illustration 5.10: Crimped channel failure on test sequence 10 (channel 4) 108

Illustration 5.11: Voltages measured on channel 5 over the course of the test 109

Illustration 5.12: Voltages measured on channel 6 over the course of the test 109

Illustration 5.13: UUT wear indicating triangular impacts from probes 111

Illustration 5.14: Dents in test probe tips 112

Illustration 5.15: Heating element failure 113

LIST OF FORMULAE PAGE

Formula 2.1: Test coverage formulae 17

Formula 2.2: Probability function 23

Formula 2.3: Failure rate 24

Formula 2.4: Mean time between failures 24

Formula 2.5: Arrhenius equation 32

Formula 2.6: Failure rate in relation to MTBF and detection efficency 36

Formula 2.7: Stress adjustment factor 37

Formula 3.1: Preliminary model formula 59

Formula 4.1: Capacitive reactance 62

GLOSSARY

MTBF	Mean Time Between Failure
JTAG	Joint Task Action Group
UUT	Unit Under Test
DAQ	Data Acquisition (often used to describe a data acquisition device)
FIDES	a consortium formed to review some aspects of reliability measurements. A function of the ImdR (nstitut pour la Maitirise des Risques) FIDES is not an acronym, but the latin word for faith.
PCB	Printed Circuit Board
LRE	Labview Runtime Engineering
TSE	Teststand Engine

CHAPTER 1

1 INTRODUCTION AND BACKGROUND

1.1 TESTING IN ELECTRONIC PRODUCT MANUFACTURING

Manufacturing is a large part of the make up of South Africa's economic system. It has been identified by the DTI as one of the sectors that needs to be developed to bring it up to international standards (Economic Sectors And Employment Cluster, 2010:88). Testing is a fundamental part of the manufacturing line, and since by its definition, a test is a comparison against a known good standard, the reliability and accuracy of test equipment cannot be in question or else all manufactured products that come off the line are in question.

Arguably, the two most important tests outlined above are the ICT and the functional/performance test, and on many manufacturing lines these are the only tests implemented. The reason for this, is that the populated circuit board manufacturing process is where the majority of errors in the manufacturing line can creep in, and so an in depth test step at this point will catch the largest number of errors. The functional/performance test step is also implemented, though in many situations this is not fully implemented but rather just as a “power on test” to see if the unit powers up and starts running, since in this case everything is put together and the full system can be tested. A functional test does not pinpoint which of the subsystems for a specific function are not working, but does indicate if one of them is not working if that functional test fails.

A functional test can be implemented by merely powering up the UUT and allowing an operator to use it in a specified way, ensuring all the functionality is tested, so for many

products this test does not require a jig as such. The in circuit manufacturing test however is more often than not implemented as a test jig with a bed of nails interface to the UUT.

The electrical signals that pass between the UUT and the electrical test system pass through a number of generic points, one of them being the test probes and their receptacles. In this study, a number of different test probe receptacle connection methodologies are evaluated, and a selection model is created to allow the design engineer to choose the correct connection methodology for their application. The following connection methodologies are examined:

1. Wire wrapping - where a thin wire is tightly wrapped around the receptacle's square post to make a tight mechanical connection.
2. Solder connection – where a solid or multi core wire is connected to the receptacle by soldering it in place.
3. Crimp connection – where a wire is mechanically crimped in place on the back of the receptacle.
4. Back plane connection with solder receptacles – where a wire wrap receptacle has its back end post inserted into a female header receptacle that is soldered to a PCB that is placed behind the Bakelite plate.
5. Back plane connection with double ended probe – the receptacle spring loaded pin is pressed onto a PCB pad that is placed behind the Bakelite plate.
6. Soldered into PCB – where a wire wrap receptacle is soldered directly into a PCB by its wire wrap post
7. PCB on top – where a receptacle, of any kind, is soldered into a PCB that sits on top of the Bakelite, so it is in fact the body of the receptacle that is soldered in place, not the connection point.

8. Double PCB mounting – where a receptacle is soldered to two PCBs that align perfectly, not into the Bakelite, which may then not be necessary.

1.2 PROBLEM DESCRIPTION

Designing a test system is a complicated and time consuming process. Reliability data exists for a number of the generic components in a standard bed of nails test system, but since test fixtures have a workable life that spans the lifetime of the product that are tested, this stands to reason. This long life however means that some components are either over specified driving up their costs, or they are under specified which causes them to fail at some future point. This study aims to find a selection model for selecting one of the components in the ICT test system namely the test probe receptacle.

The model that is needed is reliability based, and as such reliability data will be collected for all of the connection methodologies that are being studied, but it is not solely based on reliability and as such a number of other parameters are taken into account.

1.3 PROBLEM STATEMENT

A test probe receptacle mounting methodology selection model is required that can simplify the design of ICT test systems. This model should be reliability based, but should also take into account other quantitative and qualitative factors that would be considered during the design process.

On the following page the sub problems that can be derived from the problem statement are defined.

1.3.1 Sub problem 1

The relative reliability of the receptacle mounting methodologies is required, and to determine these values a test system is required to place those receptacle mountings under thermal stress while repeatedly testing them until they reach a failure point.

1.3.2 Sub problem 2

The model creation requires the qualitative analysis of the different mounting methodologies so that the model can be updated with information relating to the maintenanance, pricing and ease of design.

1.3.3 Sub problem 3

Due to the large number of test cycles that need to be run, a fully automated testing system is required that would allow the system to continue testing and only notify the operator if specific intervention is required.

1.4 AIM OF THE STUDY

The aim of the study is to simplify the design engineer's burden in designing an ICT test system, by allowing them to simply fill in the required parameters into a selection model, and for that model to indicate which receptacle methodology is most suitable to the specific design problem.

1.5 HYPOTHESIES

From the problem statement the need is for a receptacle mounting methodology selection model. On the following page, the hypothesies that form the basis for this study are derived from the sub problems.

1.5.1 Hypothesis 1

Relative reliability is a sufficient measurement used to compare the reliability of different receptacle mounting methodologies. It is possible to obtain these relative reliabilities by subjecting the different mountings to repeated test cycles under temperature stress.

1.5.2 Hypothesis 2

Beyond quantitative measures like the reliability of the receptacle mountings, qualitative data like the relative ease of maintenance, design, and price will also be important in the creation of a reliability based model for the selection of a mounting in a design process.

1.5.3 Hypothesis 3

An Automated pneumatic test system can be designed and controlled through a Labview program, written for that purpose. The operator does not need to be present for the tests to continue running, and only when it becomes necessary for operator intervention does the system need to notify the operator as such.

1.6 DELIMINATION

For the purposes of this study, a dedicated test system will be built that will subject the receptacle mountings to temperature stress while the system is repeatedly cycled and tested. For the design of that system, only the mounting systems for each of the connection methodologies, the interface boards to them, the thermal control system and the temperature sensing hardware will form part of the system. The DAQ system will be a commercial system purchased and not designed for this project.

Of the software that will be created for the system other than for the creation of the graphs, there will be no specific reporting system created for accessing the data in the form of engineering reports.

1.7 ASSUMPTIONS

It is assumed that relative reliability between connection methodologies is sufficient in the creation of the selection model. The absolute reliability prediction model for each of the studied connection methodologies will not be created. It is also assumed that the receptacle connection is more reliable than the test probe that is inserted into it, and as such, a failure of the test probe or any other part of the signal chain connected to the receptacle is not considered a failure of the part. It is also assumed that the specifications of the TDA DAQ system are sufficient for the creation of the model.

For the reliability measurements, it is assumed that 30 000 test cycles with the receptacles mountings under thermal stress will be sufficient to get a measure of the reliability of each of the connection methodologies. 30 000 cycles is the recommended number of cycles where a full service should be performed on a fixture, including replacing the test probes and servicing the receptacles, information gained from Mr. Adrian Storie, Managing director of Test Fixture Technologies.

1.8 MODEL CREATION METHODOLOGY

In this study, a reliability based model will be created for the appropriate selection of one of the components of the ICT test system, namely the test probe receptacle. To create the model, the following steps will be followed:

1. An analysis will be done of the different connection methodologies, paying specific attention to the failure modes of those receptacles.

2. A test system will be developed which will allow for an accelerated life test to be done on a set of receptacles, with the goal of collecting relative reliability data for those receptacles. Since the model will be comparing the different receptacles to each other, and the receptacles will have an identical mission profile, it is not necessary to collect absolute reliability data.
3. Pricing will be collected for each of the receptacles, as well as cost to implement per test point and repair/replace costs will be taken into account for different numbers of probes per ICT system.
4. Qualitative notes will also be made comparing the maintainability as well as other less quantitative advantages and disadvantages between the different receptacle methodologies.

From the collected data and pricing information, a selection model will be created that includes the qualitative data from the relative reliability data, as well as the qualitative selection information.

1.9 VALUE OF THIS STUDY

The value of this study is in the simplification of the design engineer's job in creating an ICT test system. This will translate into a saving in both cost and time, as well as minimising failures in time of components that are incorrectly specified.

There is also value to be gained in the process of building the receptacle methodology selection model as this can form the basis of further research in this area in the form of refinements to the model, as well as in building a basis for creating models of other mechanical components in the ICT system that are selected in the same way.

1.10 OVERVIEW OF THE DISSERTATION

Chapter 1

Sets out the foundation of the study, sets out the problem statement and hypothesis and gives boundaries to the study by stating the assumptions and methodologies to be used, as well as giving a goal by stating the aims of the study.

Chapter 2

Provides an overview of the available literature and a theoretical background in testing and the purpose of the ICT test system on the manufacturing line, and what impact reliability measurement has on the manufacturing of products.

Chapter 3

Deals with the higher level design of the test system to collect the relative reliability data, and also outlines the qualitative data that will be collected and the methods used to evaluate them.

Chapter 4

Presents the hardware design of the test system for the collection of the relative reliability data, as well as interpretation of the qualitative data collected for use in the model.

Chapter 5

Presents the receptacle mounting methodology relative reliability data, as well as the mounting selection model.

Chapter 6

Presents the results of the study, as well as recommendations for further work that can be done in the field.

The flow chart below tracks the process followed in this project in the creation of the relative reliability tester as well as the creation of the receptacle methodology selection model.

1.11 SUMMARY

In this chapter the outline of this project is put forward, as well as giving the reader some background as to why the project is undertaken as well as highlighting the value that will be gained at the conclusion of it.

The following chapter consists of an in depth literature review of the concepts used with regards to reliability testing throughout this project, as well as a discussion of where ICT testers fall in the manufacturing process.

CHAPTER 2

2 LITERATURE SURVEY AND METHODOLOGY OVERVIEW

2.1 TESTING AS PART OF THE MANUFACTURING PROCESS

The first step in creating a test system, is the formulation of an overhead test plan. Crowson (Crowson 2005:231) proposes that the following elements form part of the test plan:

1. The location that the test will be performed.
2. Environments to be used in the test .
3. The equipment that will be used.
4. The sequence of tests performed, and the procedures that will be used.
5. check sheets for recording results.
6. the criteria for each step that will determine if the unit passes or fails.
7. Summary records of all test results obtainable.
8. signatures and identification stamps for the tester.

Although this study is concerned only with the interface between test probe receptacle and electrical system in the design of test equipment, it is important to have context for this decision. It should also be noted that there is a difference between test equipment, and a test strategy which needs to encompass the entire process and strategy for a product from start of manufacturing to the end of is life cycle. That being said, once a product has been designed and is handed over to the test department, it is done by starting off with a test requirements analysis.

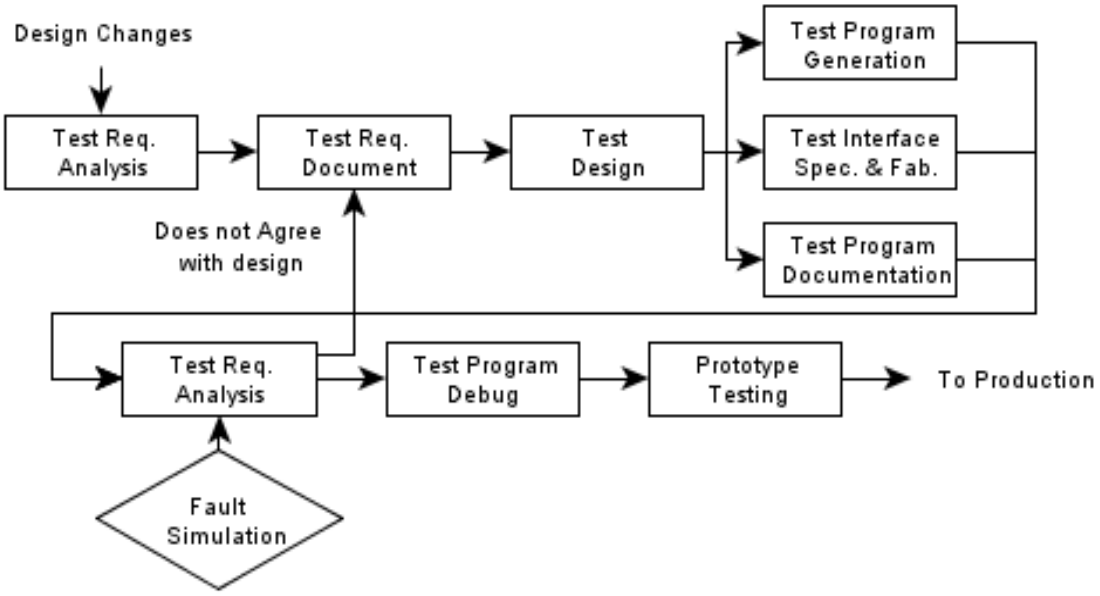


Illustration 2.1: The idealised test generation process (Scheiber, 2001:231)

Once the requirements analysis has been done, a test specification, or test requirements document can be written, and it is against this specification that the test system will be designed. This process will need to be followed for each test system that is in place on the manufacturing line, since each step in the manufacturing process will have its own unique test requirements.

In a standard manufacturing line for an electronic product, there are a number of discrete steps, involving either assembly, SMD (Surface Mount) soldering, through-hole soldering, cleaning etc. and between each of these steps there is an opportunity for either testing or inspection. An example of a generic manufacturing line for an electronic product can be seen on the following page.

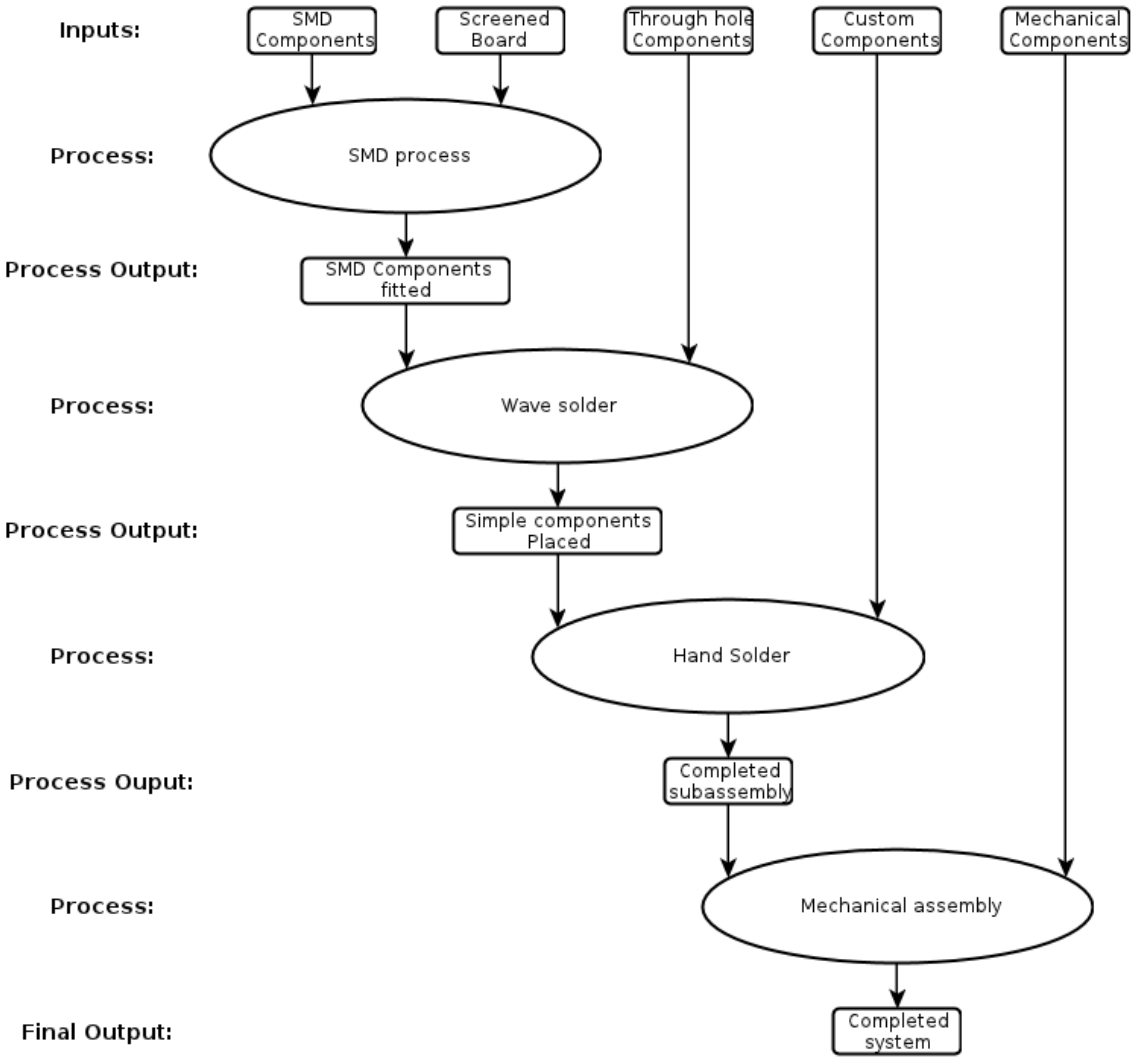


Illustration 2.2: Example of a simplified manufacturing line (electronic product)

The generic view of the manufacturing line for an electronic product above assumes the following things:

1. That the "electronic product" consists of a single circuit board that is fitted into a mechanical assembly.

2. The mechanical assembly can consist of a shell, as well as some mechanical interface hardware, for example a silicon keypad or perspex view window with LCD etc.

3. The assumption is made above that the circuit board can be manufactured with a single SMD step, a single wave solder step and an optional hand solder step. This may not be the case for boards where there are components fitted on both sides of the circuit board,

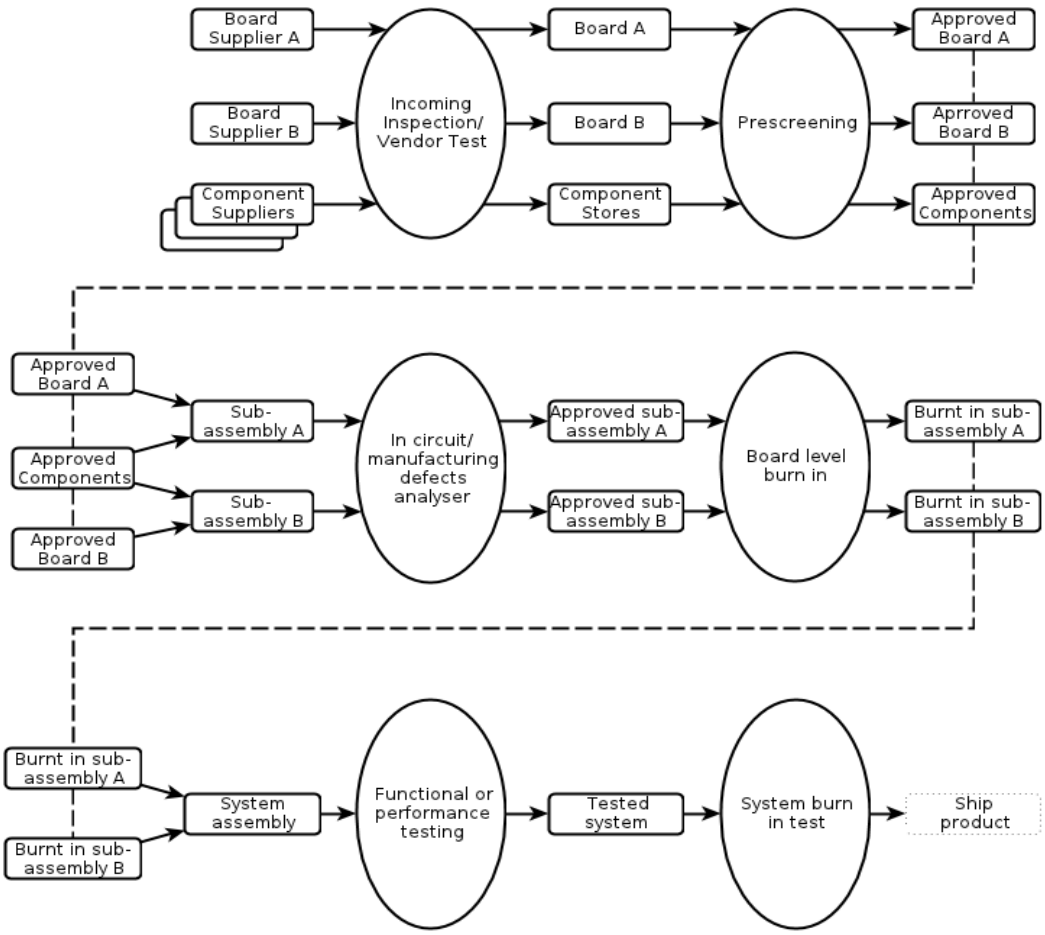


Illustration 2.3: Generic manufacturing test flow with test points

r if the design requires components to be fitted vertically in a certain order.

Looking at the above generic manufacturing line, a number of points in the line can be tested to ensure the greatest level of test coverage as described by Scheiber (2001:45).

A test can be defined as:

" an experiment of arbitrary complexity that will pass if the tested properties of a device (or set of devices) and associated connections are all acceptable, and may fail if any tested property is not acceptable" (Herd, Parker, and Follis, 2011:2)

2.2 ICT SYSTEMS

For many years ICT has been thought of as a dead or outdated technology since it has been around for a very long time, but looking at the breakdown of ICT solution providers by country put forward by Balangue (2011:2), there is still a vibrant market for it:

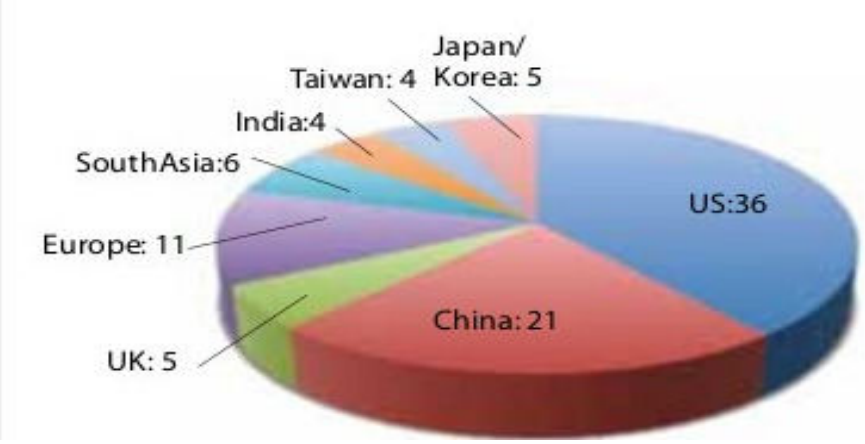


Illustration 2.4: ICT vendors by country (Balangue, 2012:2)

The reason for this according to Parker (Parker, 2011:1) is:

1. The large body of knowledge in industry. There are a vast number of engineers that understand ICT technology.
2. There is no alternative with a comparable test coverage.
3. ICT gives the best actionable information ie Since each test system is designed with a specific test subject in mind, the steps taken on failure can be customised to that subject.

4. It's scary to think of not having it ie fear of change in industry.
5. Existing infrastructure that is already paid up.
6. Test engineers can easily work with PCB designers to increase test access.
7. Its worth it ie the value is still there, even with so many other options.

The above statements refer both to custom and generic ICT systems, since whatever system a company chooses to go with, there is generally a fairly large capital cost initially which tends to lock a company into certain type of test system, the notable exception to this being companies that choose from the outset to focus solely on boundary scan technologies for testing which will make large changes in the design methodology used by the company but may cut testing costs down the line, even if sacrifices are made on test coverage.

Often the terms BON, Bed of Nails, and ICT, In Circuit Test, are used interchangeably, this is however not completely correct. The bed of nails refers to the actual pin bed of test probes that make contact with the UUT. The ICT uses a BON to connect to each of the test points, but then attempts to measure each of the components on the board separately, independent of the surrounding circuitry (Scheiber, 2001:62). Not all test systems referred to as ICT or BON testers can fully be considered full ICT systems since they often take a number of short cuts by doing different kinds of tests and referring to them as full coverage – for example, in the case of a memory module on a UUT, a test system might use an on board micro-controller to instead read a block of memory and write a block of memory on the memory module as a test. This is not a full test, but rather a functional test as it does not necessarily test all of the functionality of that memory module or even that every pin of the module is soldered on to the board – if a pin isn't used for those operations it is not

tested. This however is an acceptable trade off in many instances, but it is important to note as it should be taken into account when the test coverage of the test system is calculated. Herd, Parker and Follis (2011:1) show that a test system's coverage can be calculated using the formulas:

$$\text{DeviceCove rage} = \frac{(\#\text{devices with test s})}{(\#\text{of devices})}$$

$$\text{ShortsCove rage} = \frac{(\#\text{accessable nodes})}{(\#\text{of nodes})}$$

Formula 2.1: Test coverage formulae

but that this does not take into account some key problems like for example the memory module problem above but also:

1. redundant pins for power and ground: for example where a micro-controller has a number of Vcc input pins, and one of them has a bad solder joint. In this situation the board may pass the test, but in operation may have increased current drawn through the other Vcc pins thereby shortening the life of the product.

2. Parallel bypass capacitors: In many designs a number of bypass capacitors are used across the supply rails of each IC on the UUT, so in the event of a bad solder joint on one of these bypass capacitors, there are a number still in parallel to it so it would still appear as if the board has passed, but in operation the product may fail due to fast transient signals that may be present on the board that were not present during the test.

3. Open circuit on unused pins: Often times a micro-controller or other integrated circuit may be placed on the board without all of the pins being used, or in some cases, pins may be unconnected in the package. This may not be a problem on face value, but it can lead to not only stray capacitance, but in the case of micro-controller and other programmable devices, a software upgrade may use a pin that was not previously tested

which on an older board could be open circuit and therefore the product will fail in the field on upgrade of the software. (Herd, Parker and Follis 2011:5)

The bed of nails together with the DAQ form the heart of the test system. The bed of nails is the mechanical interface between the UUT and the DAQ and consists of a, generally, flat plate of Bakelite or some other non conductive engineering material with a number of test probes and receptacles inserted into it. Each of these test probes on actuation of the test bed will be physically pressed on to the UUT at test points to connect, through suitable signal conditioning hardware, to the DAQ.

Scheiber describes four basic types of BON based test fixtures (2001:69-70):

1. Conventional or dedicated fixtures – permit the testing of only one, or one fixed set of different UUTs.
2. Universal fixtures – have a great number of test probes placed in a grid formation, usually on 100mil pitch, to allow for the testing of a whole family of boards. For each specific board to be tested, a plate is put in place which masks off the unused probes.
3. Surface mount fixtures – also have a grid of pins at 100mil spacing, but use a translator plate between the probe platen and the UUT to move the pins slightly so that they can be used closed together.
4. Clamshell fixtures – allow for the testing of points on both the top and bottom of the UUT.

A subtype of fixture often used, which can be a subtype of any of the above fixtures, are combination testers, which have test probes set at different heights. Different tests can therefor be performed by actuating only the different height-sets of probes (Scheiber,

2001:91). As an example, a first stage of the tester could be used to test the rail voltages of a UUT, by injecting power into it and then measuring the rail voltages and comparing to the required specification. Only once this test has passed, can the secondary set of probes then be actuated onto the UUT to perform a programming function.

There are a number of different methods for actuating the probes and receptacles onto the UUT. Commonly used methods are:

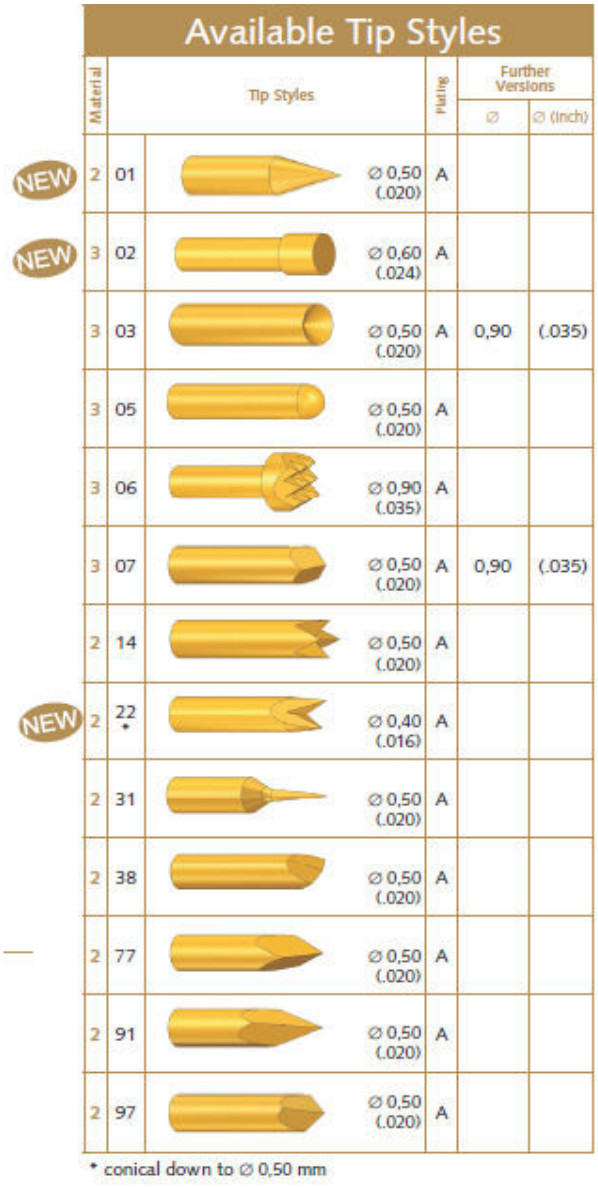
1. Vacuum – in a vacuum based fixture, the UUT is sucked down onto the stationary probe platen that holds the probes and receptacles. The advantage here is that there is no movement of the wiring and the probe receptacles, so this method is not examined in this study. The disadvantage however is that it is difficult to maintain a uniform vacuum on the UUT, and if there are any leaks in the system it can cause the UUT to actuate incorrectly and cause false failures.

2. Mechanical (manual) actuation – In a hand crank fixture, either the UUT is pressed down onto the probe set with a clamp which is actuated from above, or a handle is pulled down on the side of the fixture which lifts up the probe platen to actuate onto the UUT. This is cheapest kind of fixture to use, but can be vulnerable to operator induced failure due to the manual nature.

3. Mechanical (automatic) actuation – very similar to the manual fixture above, but in this instance, either a pneumatic or electromechanical actuator is used to actuate the probes onto the UUT. The advantage here is that there is less operator intervention, and the force of actuation can be more carefully controlled.

The choice of test probe used is obviously critical in the design of a suitable bed of nails and as such, test probe manufacturers make a variety of probes to suit every kind of test

solution. An example breakdown of different probe solutions can be found below from the Ingun test probe catalog (2011:25):



Available Tip Styles

	Material	Tip Styles		Plating	Further Versions Ø	Further Versions Ø (inch)
NEW	2	01	Ø 0,50 (.020)	A		
NEW	3	02	Ø 0,60 (.024)	A		
	3	03	Ø 0,50 (.020)	A	0,90	(.035)
	3	05	Ø 0,50 (.020)	A		
	3	06	Ø 0,90 (.035)	A		
	3	07	Ø 0,50 (.020)	A	0,90	(.035)
	2	14	Ø 0,50 (.020)	A		
NEW	2	22 *	Ø 0,40 (.016)	A		
	2	31	Ø 0,50 (.020)	A		
	2	38	Ø 0,50 (.020)	A		
—	2	77	Ø 0,50 (.020)	A		
	2	91	Ø 0,50 (.020)	A		
	2	97	Ø 0,50 (.020)	A		

* conical down to Ø 0,50 mm

Illustration 2.5: Test probe tip types (Ingun 2011:25)

In addition to having different different head shapes for different applications, test probes also come in different sizes and different types, for example double ended probes and switching probes.

Since the test probes are forced to make and break contact with the UUT every time a test is initiated and ended, the probe needs to be made from a durable material. Test probe bodies are generally made from either brass, steel or beryllium-copper with an additional plating of either gold, aurun, nickel, rhodium or silver (Ingun, 2010:13). Switching probes and other non-conducting probes generally have a nylon head, also for its durability.

Since test probes are spring loaded, it is important to note that they will exert an upward force onto the UUT when the test plate is actuated. This force is for each individual probe and is specified in the manufactures data-sheet. For example, the E-100 series test probes from Ingun will exert an upwards force of either 2.0N or 3.0N depending on the type chosen (Ingun, 2010:22). It is entirely possible on test systems with hundreds of test probes to be exerting a force of close to a ton onto the UUT. It is therefore important to have a suitable cover plate over the UUT which is in place during testing to press down onto the board and cancel out the force from the probes to ensure that the board doesn't mechanically warp.

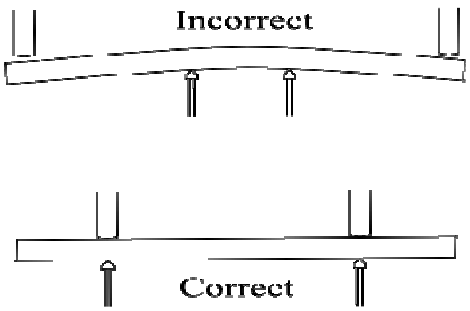


Illustration 2.6: Proper probe and force cancellation placement

Test probes need to be fully actuated to ensure proper contact to the UUT. In the case of the E-100 type probes referenced above the working stroke is 4.3mm, ie the pins need to be pressed down by 4,3mm once connected the UUT to ensure proper contact. A maximum stroke of 6.4mm for these probes is allowed, but it is advisable to stay at the working stroke length.

Test probes are generally user serviceable parts, as they are not fitted into the probe plate directly, but rather through a probe receptacle. Electrical connection to the test probes through the receptacle can be done in a variety of ways, depending on the vertical travel that the probe plate will do and how often the probes will need to be serviced. Common connection types include solder cup, wire wrap, SMD mount, on to a back plate circuit board, or crimp receptacles (Ingun, 2010:27) :

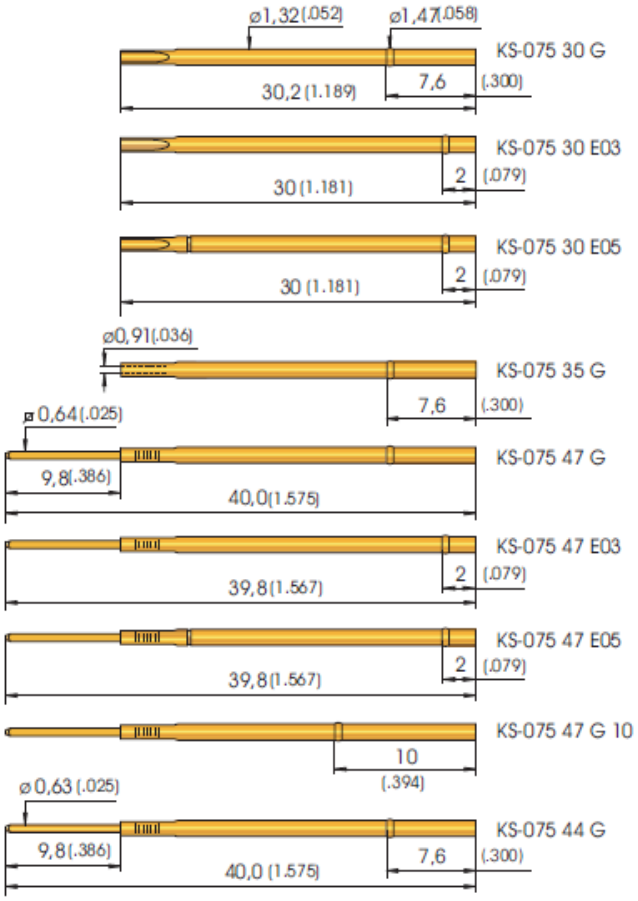


Illustration 2.7: Probe connection methodologies showing various options (Ingun, 2011:27)

2.3 RELIABILITY AND RELIABILITY PREDICTIONS

Reliability data is based on statistical measurement of a product in its mature stages, allowing the designer to go back and change some small things to make the product more reliable. Reliability needs to be a consideration during every stage of the product development cycle. As stated in the military handbook of reliability:

"Every aspect of an electronic system, from the purity of materials used in its component devices to the operator's interface has an impact on reliability. Reliability engineering must, therefore be applied throughout the system's development in a diligent and timely fashion, and integrated with other engineering disciplines." (MIL-HDBK-217F, 1995:3-1)

According to Tricker (1997:10), rigorous quality control procedures must be performed on the product during its manufacturing stage, but before this can be done, a prediction must first be made on the expected life of the product. This is done by collecting all available engineering data on the product's individual components, subsystems and mission profile and performing a statistical analysis on this data to get out a number of reliability predictors. The three main mathematical expressions used for indicating the reliabilty of a product are the probability function, failure rate and mean time between failures. As working definitions, the following are used:

Probability function: The probability function indicates the likelihood of the product performing as required, and is expressed as a percentage. The probability function can be calculated as follows:

$$p = \frac{a}{a+b} . 100$$

Formula 2.2: Probability function (Tricker, 1997:105)

where : p = probability function expressed as a percentage

a = number of units passing

b = number of units failing

Failure Rate: The failure rate expresses the number of failures per unit time

$$f = \frac{a}{b}$$ *Formula 2.3: Failure Rate(Tricker, 1997:106)*

where: f = failure rate expressed in failures per hour

a = failures

b = amount of time that the test ran in hours

Mean Time Between Failures: The mean time between failures is the inverse of the failure rate and expresses the average time between failures that was recorded during the test sequence.

$$MTBF = \frac{1}{f}$$ *Formula 2.4: Mean Time Between Failures(Tricker, 1997:107)*

where: MTBF = mean time between failures in hours

f = failures rate in failures per hour

The Probability function, Failure rate and Mean time between failures are all measures that can be used to determine the available "life" left in a product when it is currently in use. The data that is used is generally obtained from the datasheet of components during design/installation time and it is up the designer to ensure that similarly reliable components are used in a design since the design can only be as reliable as the least reliable component in the design.

The reliability data of a specific product often follows the bath-tub curve which indicates the likelihood of failure versus time for a specific product.

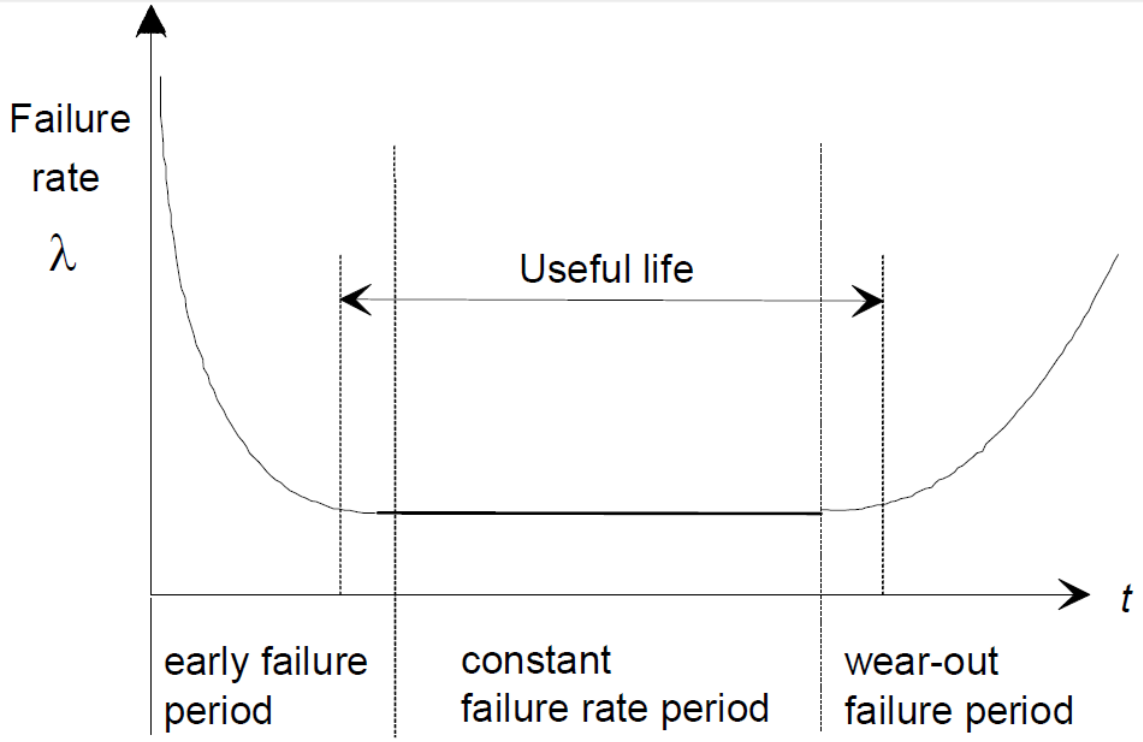


Illustration 2.8: Bath tub curve(EPSMA 2005:4)

From the traditional bath tub curve of failure, Davis and Davis as quoted by Scheiber (2001:200) break the infant mortality range down even further into 3 more sub sections namely Internal Defects, pre-shipment, external defects, within warranty, and external defects (outside warranty). They also found that the decay of the curve for infant mortality is related directly to the mean time between failures by the function:

$$F(x)=e^{-x/MTBF}$$

Formula 2.5: Infant mortality decay (Scheiber, 2001:200)

Where x is time.

The following page shows graphically the decay of the infancy mortality curve, as well as indicating the sub sections described above.

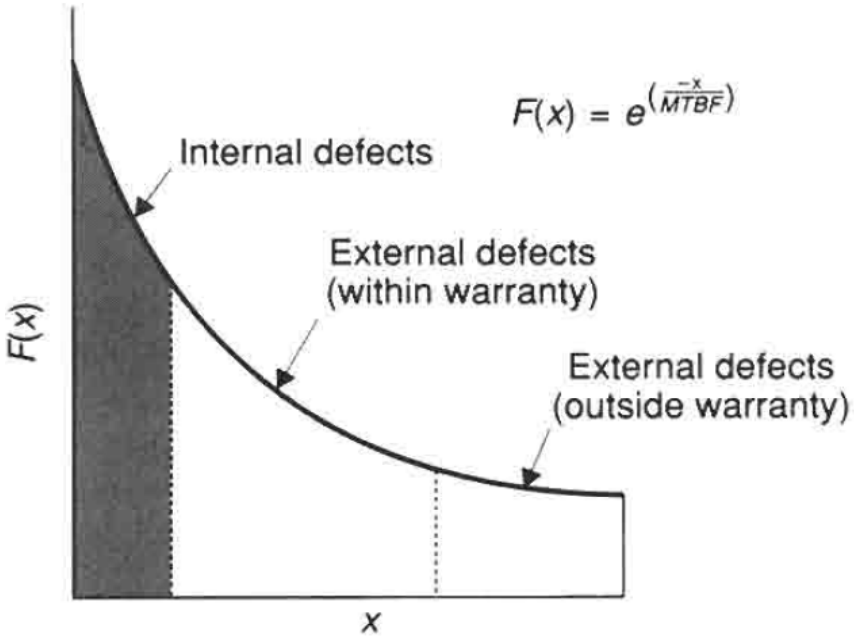


Illustration 2.9: Infancy mortality period (Scheiber, 2001:200)

2.4 ENVIRONMENTAL STRESS SCREENING

An environmental stress screening test is a test where the unit under test is subjected to very hard working conditions in terms of temperature, humidity etc, to enforce a much higher stress on the product than would normally appear in its infancy. In this way, those devices that were going to fail early can be screened out, and not be given the opportunity to fail in the field.

Scheiber (2001:201) suggests that the following are the advantages to doing an environmental stress screening on a product during manufacturing:

1. Reduces in warranty service costs incurred when the product fails under warranty.
2. Increases product reliability and decreases return rate.
3. Allows for better data collection in factory during manufacturing runs.
4. Allows better process control than with statistical process control only.

Scheiber (2001:202) also defines the following stress screening levels:

1. Static Screening: The unit is exposed to more extreme environmental pressure than it would normally experience, and may require the board to be powered up.
2. Dynamic screening: The unit is exposed to more extreme environmental pressure than it would normally experience, it is powered up and the inputs are also fed with voltages that would exercise the extreme cases the unit would be able to handle under normal operating conditions.
3. Exercised screening: The unit is exposed to more extreme environmental pressure than it would normally experience, it is powered up, simulated signals representing normal and in some cases extreme cases of input signals are fed into the unit and the outputs are put under heavy load. Exercised screening is the most rigorous screening level, and as such the one that will allow through the least number of units.

An important facet of the screening process that will be present during test is that of monitoring, which will allow the test system to check whether the unit is performing within the expected parameters.

It is important to point out that there are a number of different methodologies that can be applied to determine the acceleration factor that can be achieved by running the product at higher temperatures. The most common methodology for determining the failure rate of the completed system used is the parts stress method (MIL-STD-271F, 1991:128), but it is noted that there is significant work being done to greatly refine the data achieved by reliability prediction methods. Standard RL-TR-92-197 updates the failure rate prediction models for many of the more primitive models, Capacitors, Resistors, Inductors etc. (RL-

TR-92-197, 1992:ii) but still uses similar prediction methods to MIL-STD-271F. More Recently there has also been work done by the FIDES consortium to produce a more unified approach to reliability prediction that takes into account the technology the product forms a part of, well as the process used to create it and how it is used (FIDES, 2010:23). In a recent article in Reliability Edge Journal of Reliability (2008:9) the following methodologies are compared:

1. MIL-HDBK-217F notice 1 and 2 (1995).
2. Bellcore/Telcordia (2006).
3. RDF 2000 (2000).
4. SAE reliability Prediction Method (1987).
5. NTT Procedure (1985).
6. Siemens SN29500 (1999).
7. China 299B (1998).
8. PRISM (2000).

Wherein the distinction is made between different methodologies by first dividing them into three broad categories which distinguish between them by how the reliability data for the product is collected. The three categories are:

1. Empirical: a standards based approach that uses collected data and tables to predict the failure rate of the product based on previous data collected.
2. Physics of failure: requires an understanding of the mechanics of failure and some modeling to determine a mathematical relationship between factors contributing to the failure rate and a predicted rate. Examples of this include the Arrhenius equation, Eyrning model and Coffin-Manson model (Reliability Edge, 2008: 11-15).

3. Life testing: requires extensive real life field models to determine the failure rate from devices that have already failed in the field, in addition to the empirical data collected from one of the standards.

Reliability prediction is however not an infallible process and the military handbook of reliability prediction (MIL-STD-271F 1995:23) puts forth a number of limitations that need to be taken into account when doing this sort of prediction namely:

1. Failure rates of individual components are point estimates and based purely on available data, so may not be complete in all cases, and will not take into account all possible failure modes.

2. Even when the operational environments are similar for the product while in use as when the data for components was collected, there can be significant differences in stress due to different applications.

3. Electronic devices are constantly changing, and it is difficult to predict reliability of future devices with data for past devices, which could be relying on fundamentally different design techniques.

4. The manufacturing process of different products may have a large impact on the reliability of the products down the line. ie Two identical products that are used in the same application that were manufactured on different manufacturing lines with different processes may have different failure rates due to the contributions of the manufacturing process.

5. Reliability is directly influenced by correct application of the product by the user.

With that in mind however, a preliminary discussion of the parts stress method is in order. The parts stress method is one of two methods described in MIL-STD-271F, the other being the parts count method. The parts count method is simpler to implement than the

CHAPTER 3

3 CONCEPTUAL DESIGN AND CONSIDERATIONS

3.1 DEFINING FAILURE AND PROPER ASSEMBLY

The first step in creating a test system, is defining what is being tested and what the pass/fail modes for that specific UUT is. For the purposes of this study, test probe receptacle connection methodologies are being tested. It is therefor appropriate to define what will constitute correct operation, and what will be considered a failure mode.

The military handbook on stress screening (ESS) of electronic equipment describes the differences between latent and patent defects in electronic equipment (MIL-HDBK-344A,1993:4-6) and also includes a list of possible failure modes. There is also a distinction made between different kinds of patent defects, nl. Errors and precipitated latent defects:

Errors are caused by incorrect handling, faulty workmanship, or installation of faulty sub components, and as such are completely preventable. An initial inspection and simple test should identify all of the errors in the test system. The precipitated latent defects however will be more specific to each kind of receptacle connection methodology and will be discussed in the following section using the list in The military handbook on stress screening (ESS) of electronic equipment as a guide (MIL-HDBK-344A,1993:4-5).

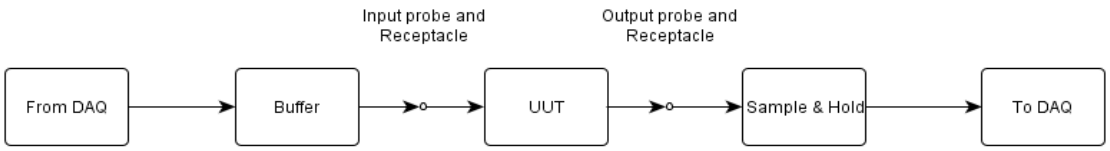


Illustration 3.1: Test signal chain

For each of the receptacle mounting methodologies explored below, the same signal chain is used as can be seen on the previous page. For this signal chain, a failure will be considered a failure of either the input or output probe's mounting, or any of the connecting hardware that the signal goes through between the buffer and the input receptacle or the output receptacle and the sample and hold. A failure of the buffer, input probe, UUT, output probe or sample and hold circuitry will not be considered a failure.

The following sub sections describe the different mounting methodologies that will be of relevance in this study in detail.

3.1.1 Wire wrap method

The wire wrap method for connecting test receptacles to the test system is a common method used which derives from a similar method that was popular with electronics manufacturers in the 1960s and 1970s which was developed after the second world war. The decline in use in commercial electronics is due to the emergence of solder-less breadboards, the decreasing cost of PCBs and the emergence of SMD technology (Wikipedia, wire wrap).

The connection head is a square wrappost that is between 0.0325in and 0.0355in (0.8255mm and 0.9mm) across the diagonal, and the connection to it is made by wrapping 7 stripped turns and ½ insulated turns of size 30 AWG wire onto it for class A connections and 7 stripped turns of size 30 AWG for class B connections (MIL-STD-1130C,2012:8). The wire should also be wrapped in such a way that there is no space between windings, and the wire should not be routed in any way that would facilitate the unwrapping of the wire.

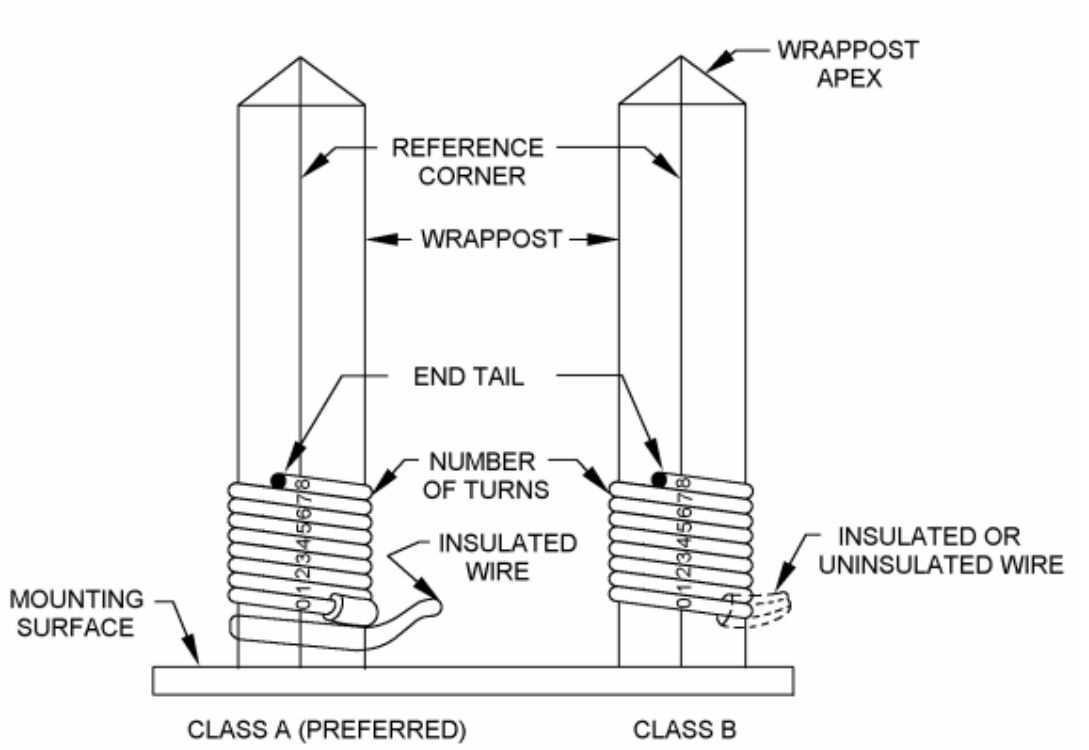


Illustration 3.2: Wire wrap wrappost wiring

The wire wrapping tool used to apply the wire to the wrappost applies up to 20 tons of pressure to the contact point between the wire and the wrappost, and if done correctly cold welds the two together. Since they aren't soldered, wire wrapped connections are immune to soldering faults like corrosion, dry joints and cold joints (Wikipedia, wire wrap).

The possible failure modes on a wire wrapped connection are mostly due to the very fine nature of the wire. Although the wire is unlikely to unwrap if properly routed, if there is excessive force on the wire connected to the wire it can snap as it is very thin. There is also an opportunity for the wire to unwrap if there are temperature fluctuations, but this is unlikely if the wire is correctly wrapped. The other end of the wire is also a possible failure

point where it connects to the rest of the ICT system, again due to the fact that it is very thin. A possible disadvantage is that the wire comes off of the wrappost at 90°.

3.1.2 Solder cup connection method

A receptacle with a solder cup connection has a small reservoir at the end of the receptacle which is used to solder a wire to. The advantage of solder cup connect receptacles is that they are cheaper than any other kind of receptical, and do not require any special tools to connect to other than a soldering iron. The possible failure modes of this methodology are due to their soldered nature and as such they are vulnerable to the weaknesses of solder joints, nl corrosion, dry joints and cold joints.

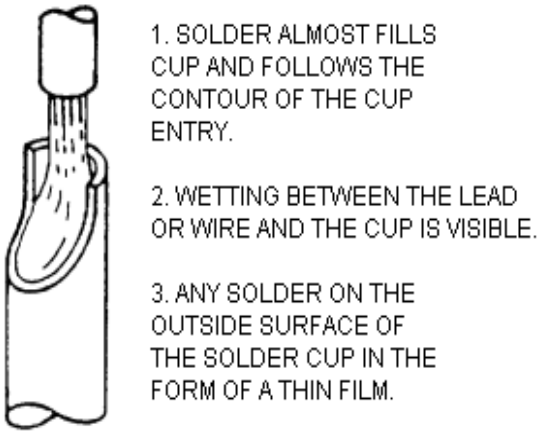


Illustration 3.3: Solder cup mounting(MIL-STD-2000A,1991:23)

Military Standard MIL-STD-2000A gives the correct method for soldering of wires and leads to solder cup connections, specifying that no more than 3 wires may be inserted into the cup, and that the solder contour should fill at least 75 percent of the mouth of the cup. Solder is allowed to overfill the cup, but solder on the outside of the cups should be in the form of a thin film and not a blob. (MIL-STD-2000A, 1991:48).

3.1.3 Crimp connection method

Receptacles with a crimp connection at the back come precrimped with a length of size 30 AWG wire attached from the manufacturer, or can be crimped using a dedicated tool. It offers a middle ground between soldered and wire wrapped connections, in that it allows the wire to come straight out the back of the receptacal and not off at 90° like the solder cup connection, but also gives a mechanical connection rather than a soldered one like the wire wrap receptacal. The failure modes for this connection methodology are again due to the thin wire that is used.

Military Detail Specification MIL-DTL-22520G gives a test method for the measurement of the quality crimped connections. A 1A current is fed through the wire, and a millivolt meter is placed on the body of the ferrule, and ½ in from the ferrule on the bare wire. For size 30AWG wire in a crimped connection this test should measure 6mV for silver or tin plated copper wire and 8mV for nickel plated wire (MIL-DTL-22520G,1997:9 Table I).

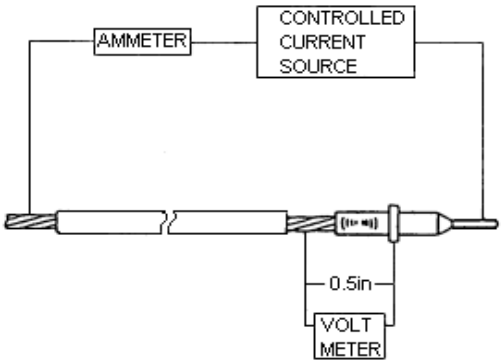


Illustration 3.4: Voltage drop test(MIL-DTL-22520G,1997:23)

The tensile strength of the connection needs to be 0.8lb for both silver or tin plated copper as well as nickel plated copper wire(MIL-DTL-22520G,1997:10 Table II). (specifications given in imperial units since the specification refers to imperial units).

3.1.4 Back plane connection with soldered receptacles

With a back plane connection, a PCB is laid out with the required wiring for the test fixture, and has single pin 2.54mm sockets soldered into it into which wire wrap receptacles are fitted. The advantage of this connection methodology is that the wiring is rigid, and by its nature fully documented since the PCB will be laid out using a schematic and netlist. It also means that if the PCB is kept up to date, then any changes that are made to the test fixture are also automatically tracked. The disadvantage is that a soldered connection is introduced in between the PCB and the receptacle, which introduces another point of failure. Also, the friction fit nature of the 2.54mm sockets means that on test fixtures that have a high pin count that the board can become difficult to remove.

Evaluating the Ingun KS-100 47 G receptacle (Ingun 2010:29) and the Samtec PHF-101-01-L-S socket (Samtec, 2011:218), the receptacle has a pin that is 10mm in length, and the socket is meant to receive a pin that is only 5.84mm in length. This means that the PCB needs to be mounted sufficiently far away from the probe platen that holds the receptacle in place to ensure a that the receptacle goes into the socket only the appropriate distance. On the following page can be seen the mechanical diagrams of both the wire wrap receptacle and PCB solder receptacle that will be used in this project to do the mounting.

The wire wrap receptacle is sourced from Ingun technologies and the PCB solder receptacle from Samtec.

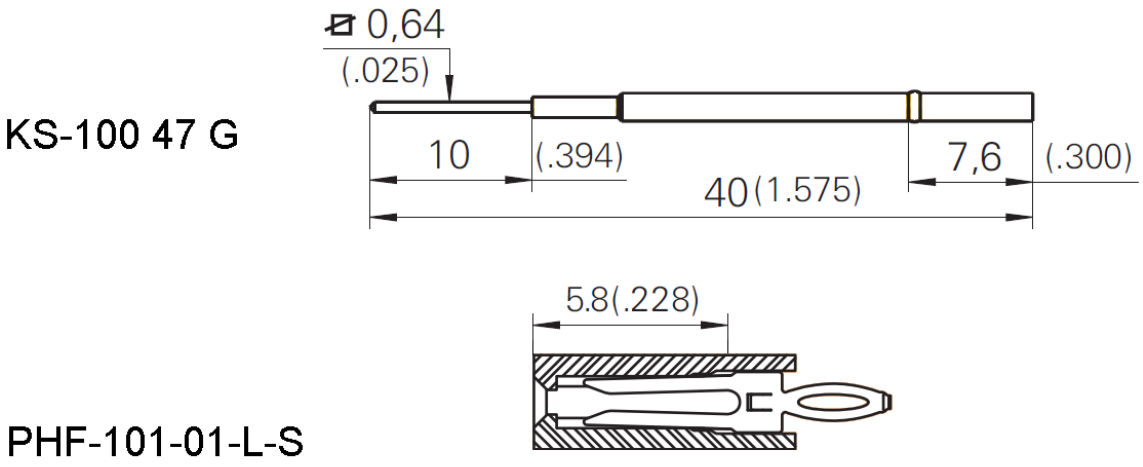


Illustration 3.5: Receptacle mechanical drawings (Ingun 2010:29) (Samtec, 2011:218)

3.1.5 Back plane connection with double ended receptacles

En evolution from the sockets soldered into the back plane, this mounting methodology uses a receptacle that has a weak spring built into the rear of the receptacle that then pushes into the back place PCB to ensure a proper connection. The advantage of this is that when the PCB is removed from its mountings, it is very easy to remove and makes servicing very simple. The disadvantage of this methodology is that the double ended receptacles are expensive when compared to other receptacles.

The DER-100 receptacle from Everet Charles (Everet Charles 2011:16) has a rear spring force of 3.5oz (0.973N), and when mounting the PCB it is important to compress the springs to the correct depth to ensure the optimal spring force is attained. The recommended travel is 3.5mm to achieve 99% of the spring force. A mechanical drawing of the rear of the double ended receptacle can be seen on the following page.

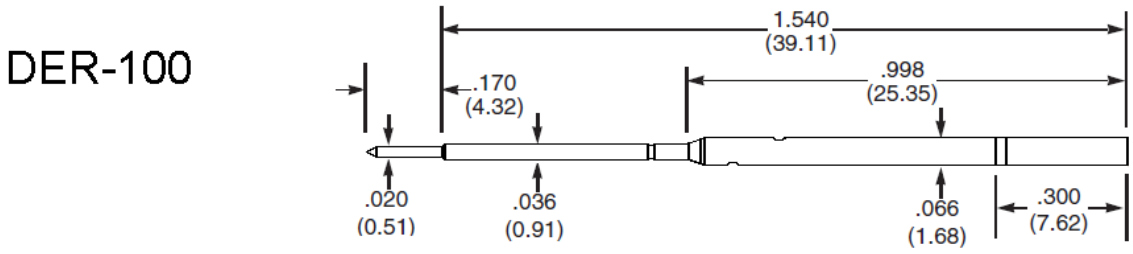


Illustration 3.6: Double ended receptacle with spring rear end (Everet Charles 2011:16)

3.1.6 Back plane with receptacle soldered in place

With a back plane connection, a PCB is laid out with the required wiring for the test fixture, then the wire wrap receptacles are soldered into place on the PCB. This means that a permanent connection is made, and it is very difficult to remove the PCB from the probe platen after it has been mounted there.

This methodology is used where insert style fixtures are used, where the wiring and the probe platen are integrally linked, and it is only done on fixtures where there is little chance that the UUT is going to change in any way. The advantage is that this is a very rugged mounting methodology since not only are the PCB standoffs holding the wiring board in place, but also

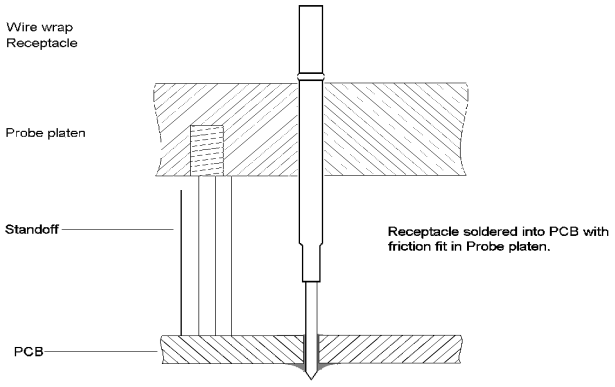


Illustration 3.7: Back plane with receptacle soldered in place

the soldered receptacles. The disadvantage however is that there is not really any opportunity to change the wiring PCB due the rigid nature of the construction.

3.1.7 PCB on Probe platen

With this methodology the wiring PCB is mounted on top of the probe platen, and the receptical is soldered into it from the top. The wiring PCB is then fitted securely to the probe platen. The receptacle can either be friction fitted into the probe platen or the receptacle can be

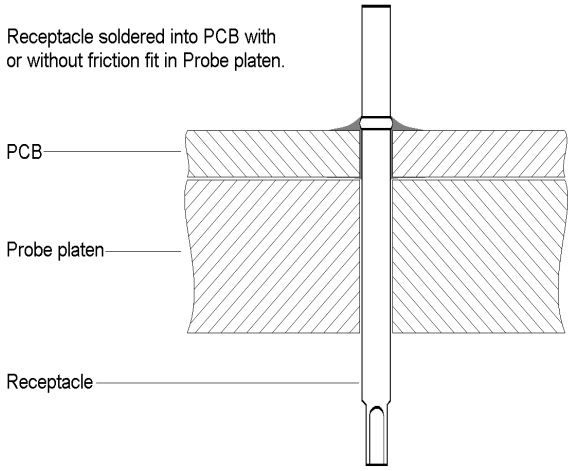


Illustration 3.8: PCB on probe platen mount

inserted slightly loosely. The advantage of friction fitting the receptacle into the probe platen is that there is a lot of mechanical stability. However making the hole 0.1mm larger allows the receptacle to freely move in and out of the hole so that the wiring board can be replaced and modified but sacrifices some of the mechanical stability.

A further disadvantage is that since the wiring board is now mounted right below the UUT with the probe platen on the other side of it, which means that if there are additional

components mounted on the wiring board to do intermediate signal conditioning there is a rigid height limit on which components can be used. The advantages are that of the PCB mounting techniques it is the cheapest, and it also has the lowest parts count which means there are less points of failure.

3.1.8 Double PCB mount

With this methodology the receptacle is mounted between a set of wiring boards. The PCBs need to be mounted rigidly together to form a rigid mechanical construct by using multiple PCB standoffs. The advantage of this methodology is that no probe platen is required. A generic lifting plate is used to actuate the probes the required amount upwards to the UUT. This means that standard PCB capture software can be used to create the wiring boards that then double as the probe platen. This represents a cost saving, as well as time saving in that when the UUT is designed the wiring boards can be created at the same time and tooled at the same time which decreases the time to market. The disadvantage is that the receptacle is not friction fitted into a probe platen, if not correctly designed, the wiring boards can be stressed and be caused to flex and break the solder points.

This is also the only methodology that allows a designer to design the fixture by and large without the access to mechanical software once a generic base has been established for the mechanical actuation. Adetailed discussion of this can be seen in the Authors previous work (Greeff, 2010).

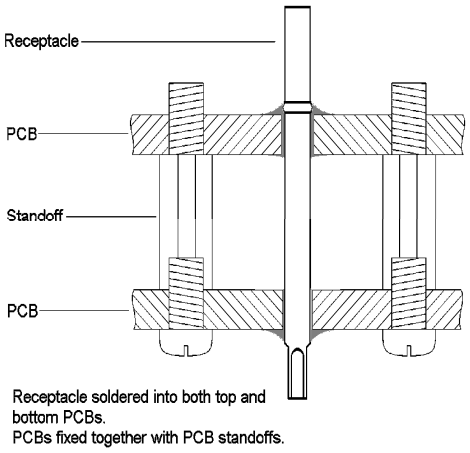


Illustration 3.9: Double PCB mounting

3.2 FIXTURE FAILURE VERSUS UUT FAILURE

On the manufacturing line, a measured failure on a test system, may not necessarily be indicative of a failure on the UUT. As an example: Assume a rudimentary test system that is testing a linear power supply (see Illustration 5).

To test the linear power supply, the following test steps are followed:

Visual Inspection: the unit is inspected to see if there are any obvious physical problems with the manufactured unit, specifically looking at the mechanical connectors J1 and J2 since these are not tested electrically when signals are injected and measured using test probes P1, P2 and P3 applied to test points TP1, TP2 and TP3 respectively.

Drain Capacitors: To ensure that the test starts from a known state, resistors are placed between TP1 and TP3 and TP2 and TP3 to allow the draining of capacitors C1 and C2. The resistors are applied using the relay control hardware in the test system.

Short/Opens testing: once the capacitors are drained, a shorts/opens test is applied to ensure that C1, C2 and U1 are not shorted out or completely open circuit (due to faulty soldering or component failure).

Apply Voltage and Load: 18V is applied to TP1 and the GND line of the supply is applied to TP3 which is the ground test point on the unit under test. The amount of current drawn from the supply is now measured to ensure it is within specified norms when a calibrated load is applied between TP2 and TP3.

Measure Voltage: The Voltage present on TP2 is now measured using the DAQ (Data Acquisition Device) with TP3 as the reference to see if the voltage is within the specified norms. (in this example, 12V +- 1%).

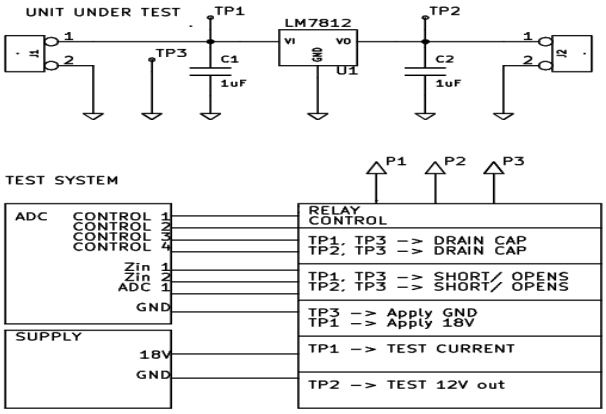


Illustration 2.10: Rudimentary test system

It can be seen from the above example, that possible failure modes of the complete system is either:

1. A component on the UUT has failed: If one of the components on the Unit under test fails, there will be a measurable failure since one of the measured signals, impedances or currents will not be within the specified norms.

2. The Control system fails: If one of the control relays fail and incorrect signals are applied to the UUT, incorrect values will be measured. For example, if the relay connecting TP1 to and TP3 tp the shorts/opens test fails, then the impedance measurement circuit on the DAQ will measure an open circuit on the Unit under test when in fact it is the relay control system that has failed.

3. DAQ failure: if the DAQ fails, then incorrect values can be measured. For example, if the ADC input port on the DAQ is blown and always measures 0V, then measuring the expected 12V +-1% on TP2, even if the UUT voltage is correct.

4. Connection failure: In this case either the wiring between the control hardware and the test probes fail, the test probe itself fails, the probe connection plate fails or receptacles holding the probes fail. If for example the test probe P2 fails so that it does not connect electrically to TP2, then obviously the test point will measure as open circuit.

Looking at this example, it can be seen that any failure in the test system will also manifest as a failure in the manufactured unit.

3.3 PRELIMINARY TEST SYSTEM DESIGN

3.3.1 Overview

In order to measure the relative reliability of different kinds of test probe receptacle interface methodologies, the receptacles will need to be run through a number of test cycles at an elevated temperature to induce a failure. This is achieved by running an automated test sequence inside of an environmental test chamber. Since the different receptacle interface methodologies are being compared to one another to create the selection method, absolute reliability is not as important as relative reliability. Since the only receptical mounting methodologies are being tested, the only area in the text fixture which will be temperature cycled will be the inner chamber of the fixture where the

receptacles are mounted. For each of the connection methodologies, at least 5 receptacles of that type will be tested.

It is with this in mind that the system will be designed using the techniques described by Neighbour et al as a guideline (Neighbour et al., 1990).

A standard box based pneumatic test fixture will be sourced and modified to allow for the heating and cooling of the inner chamber test system. The following modifications will be made:

1. The inner chamber will have multiple temperature sensors to allow the chamber temperature to be measured at multiple points during testing.
2. A heating element will be installed either inside the chamber, or outside of the chamber with a blower to allow for the heating of the test system's inner chamber.
3. Instead of using a compressor based cooling system, since the inner chamber has a limited volume, a Peltier cooler will be used as the cooling element.
4. The side of the chamber will be modified to allow all of the external electrical cabling can be inserted into the chamber, so that the measurement equipment is not subjected to the same stresses as the unit under test.
5. A custom probe platen will be designed that will allow for all of the tested methodologies to be mounted at the same time so they can be tested under the same conditions.
6. The pneumatic lifting system, the test probes, UUT, test instruments and other interface electronics will be thermally isolated so that they are not cycled in the same way as the receptacles. A silicon sleeve will be used the thermally isolate the chamber where possible leaks can occur between the inside and outside at this interface.

The inner chambers should have as small as possible internal volume to allow for the fastest temperature transitions. The temperature will also not just be held at a fixed point for the duration of the test, but rather moved through cycles similar to the profiles described in thermotron (1998:12).

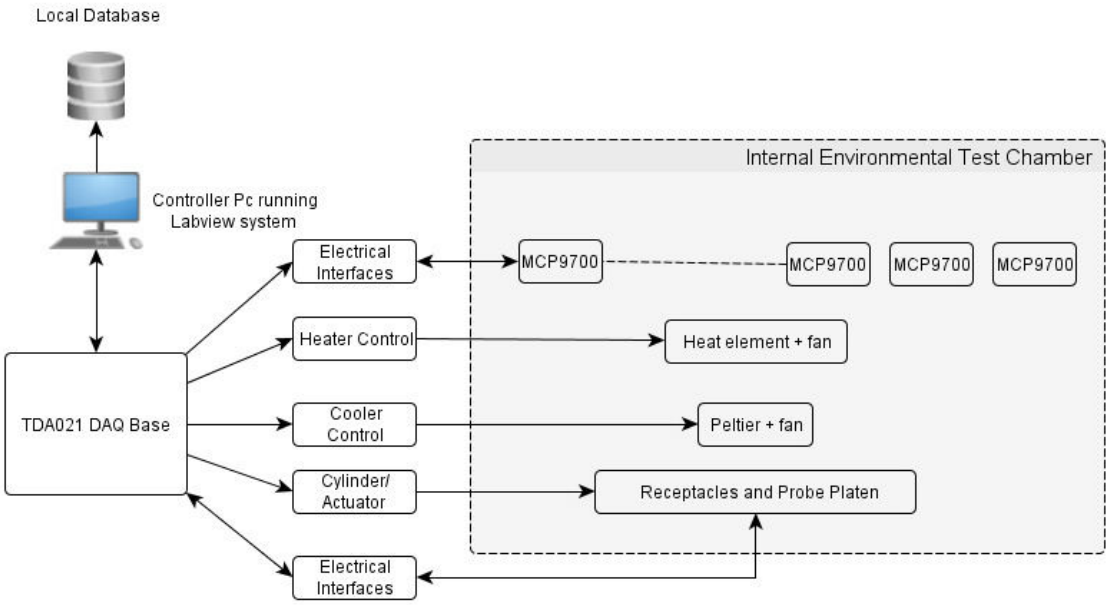


Illustration 3.11: Relative reliability tester high level layout

3.3.2 Hardware and equipment

Temperature measurement will be employed via the MCP9700 family of sensors. These sensors cover a wide temperature range (-40 to +150 °C) and that should cover all eventualities (Microchip 2009).

The heating element used will be a purely resistive load and will be a replacement element for a standard portable 12V oven/ boiler or a small 220V oven which will be either installed into the freezer, or if the temperature of the element is too high, externally and the heat blown in with a blower. To control the ramp gradient, the element will be controlled

via a PWM signal from a high current DC source, or with an SCR controlling an AC source.

the following strategy of temperature monitoring is proposed.

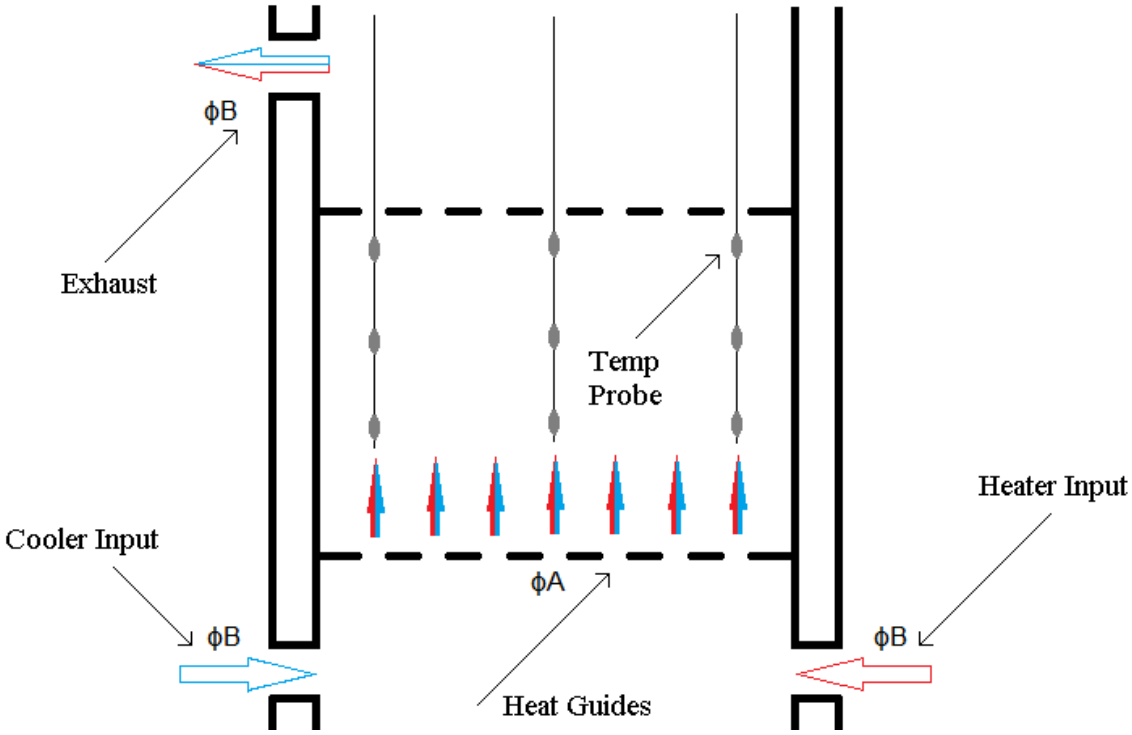


Illustration 3.12: Proposed heat flow in the chamber

Inside the chamber, plates will be fitted to restrict the heat flow, called respectively the top and bottom heat guide plates. In the above diagram øA defines the size of the heat guide holes in the plates. The heat will be pumped into the chamber via a fan, and the heater input and exhaust ports will both have øB sized holes. To ensure there is proper heat flow in the chamber, the sum of hole area of size øA holes in each of the plates should equal the area of øB. The cooler input will be mirrored on the other side of the chamber. The chamber temperature will be measured at multiple points to ensure that a proper temperature profile of the inner chamber can be drawn.

For data measurement, a TDA scalable DAQ system will be used. The TDA system allows for multiple DAQ devices to be inserted into a base system which can then be controlled from a single RS232 port. The Scalable DAQ system base has 8 slots into which different modules can be inserted and each is individually addressable. The TDA019D DAQ module will be used in this project.

Other DAQ devices are commercially available, but the main reason that the TDA system was chosen was a cost consideration due to the number of analog sample and hold chains that would be required (40 A/D channels with 8 pulsed outputs). Other available options were the NI USB-6225 and the Eagle R-USB-26C series of DAQ Devices. To measure on the same number of channels (without having a switch matrix to re-purpose channels), the following hardware options would be suitable:

1. 1x NI USB-6225 priced at R32,760
2. 1 x R-USB026C32-DC and 1x R-USB026C16-DC priced at R29,003 (in total)
3. 7 x TDA019D and 1xTDA021E Controller priced at R12,800 (in total)

Other higher specification options were also available from both eagle technologies and National instruments, but in terms of electrical specifications the selected hardware systems were both superior to the TDA system. However, since the main differentiation between different chains was gained largely through the sample and hold methodology used, the 10 bit TDA system was considered to be sufficient. It is also worth noting that the TDA021E controller was already available at the start of the project and did not need to be purchased. It is the author's belief though that both the Eagle Technology and National Instruments DAQ systems would have performed at least as well as the TDA system if they were chosen. Labview and Teststand were already available to the author, and it already included drivers for all of the above hardware systems.

The TDA message structure is as follows:

Start	Length	Reserved	Slot	Reserved	Id	Data	Checksum	End
0x61	variable	0x00	variable	0	variable	variable	variable	0x63

Table 3.1: TDA DAQ Message structure (all fields 8 bit) (TDA 2013:5)

The calculations for each of the fields can be found in the TDA Scalable DAQ manual (TDA 2013:5,6). To test the connectivity through the signal path containing the receptacle, a 500Hz square wave will be injected onto each point, and then measured through a Sample and Hold circuit to measure if there are any breaks in the signal path. All signals to and from the measurements and control will be generated and measured through the DAQ system.

All interface electronics between the DAQ system, the heating and cooling systems and the receptacles will be custom designed for the purposes of this project.

The signal chain for each of the connection methodologies will allow for 5 input and 5 output receptacles for the signal to flow through. This means that there are 2 receptacles on each line that can fail.

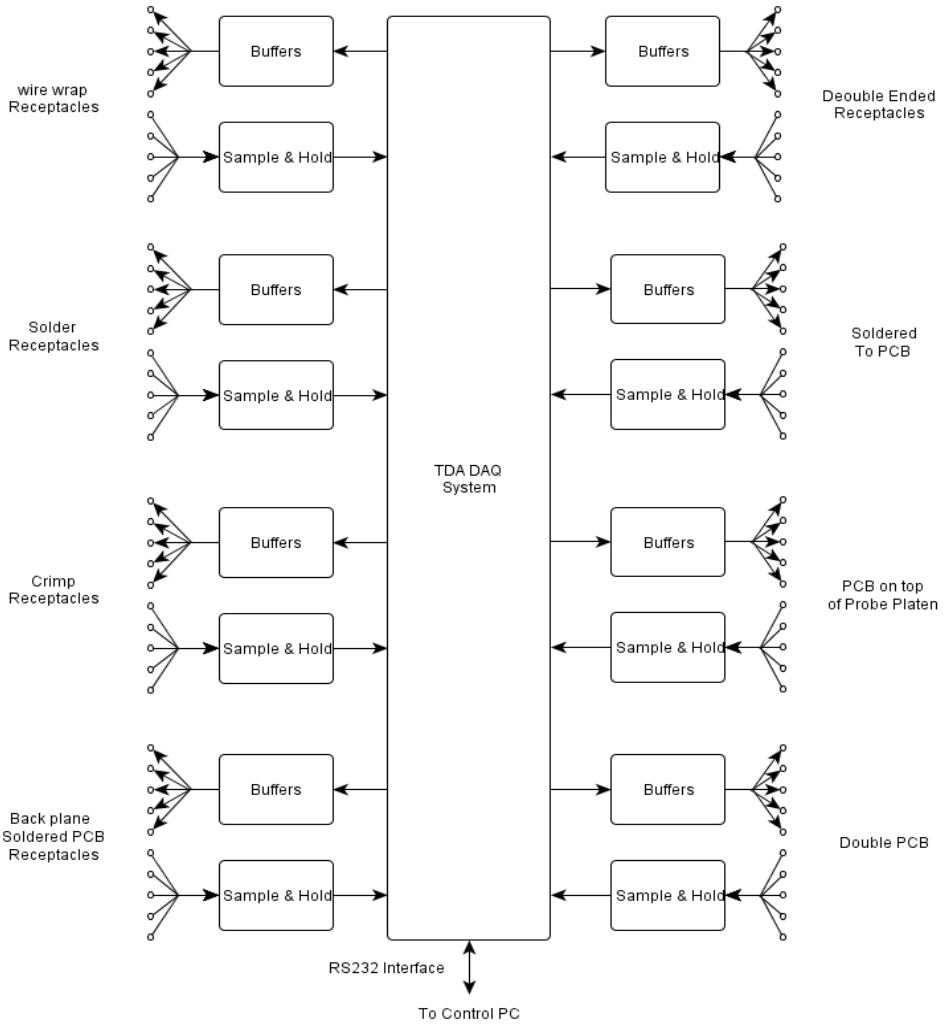


Illustration 3.13: Proposed signal chain and wiring design

3.3.3 Software and data budget

The control of the test system and the logging of data collected will be done in national instruments Labview. The described protocol for the TDA Daq system will be implemented, and the control of the heating and cooling elements, as well as the injection and measurements of signals will be done through this system.

Logging of data will either be done by using National Instruments Teststand, or with a custom based logging platform which writes to a database or csv file.

The following data points will be collected (for each test cycle) as a minimum:

Input	**Unit**	**Data Size**
Ambient temperature	°C	10 bits (2 bytes)
Heater input temperature	°C	10 bits (2 bytes)
Cooler input temperature	°C	10 bits (2 bytes)
Central chamber temperature	°C	10 bits (2 bytes)
Exhaust temperature	°C	10 bits (2 bytes)
Signal level (for each test point)	V	2 bytes x 40 points
Timestamp	time	4 bytes
Total		174 bytes

Table 3.2: Data collected for each test sequence

Test cycles will run continuously until there is a failure event, which will warrant inspection to identify whether the failure was a valid signal chain failure, or a failure that does not indicate a receptacle failure (for example, the failure of a test probe or sacrificial board).

Should a receptacle fail, it will no longer contribute to the test data (the signal line through that receptacle will then be considered inactive). The tests will run to a specified time limit, or until all of the receptacles have failed.

A flow chart of the proposed test sequence can be found on the following page.

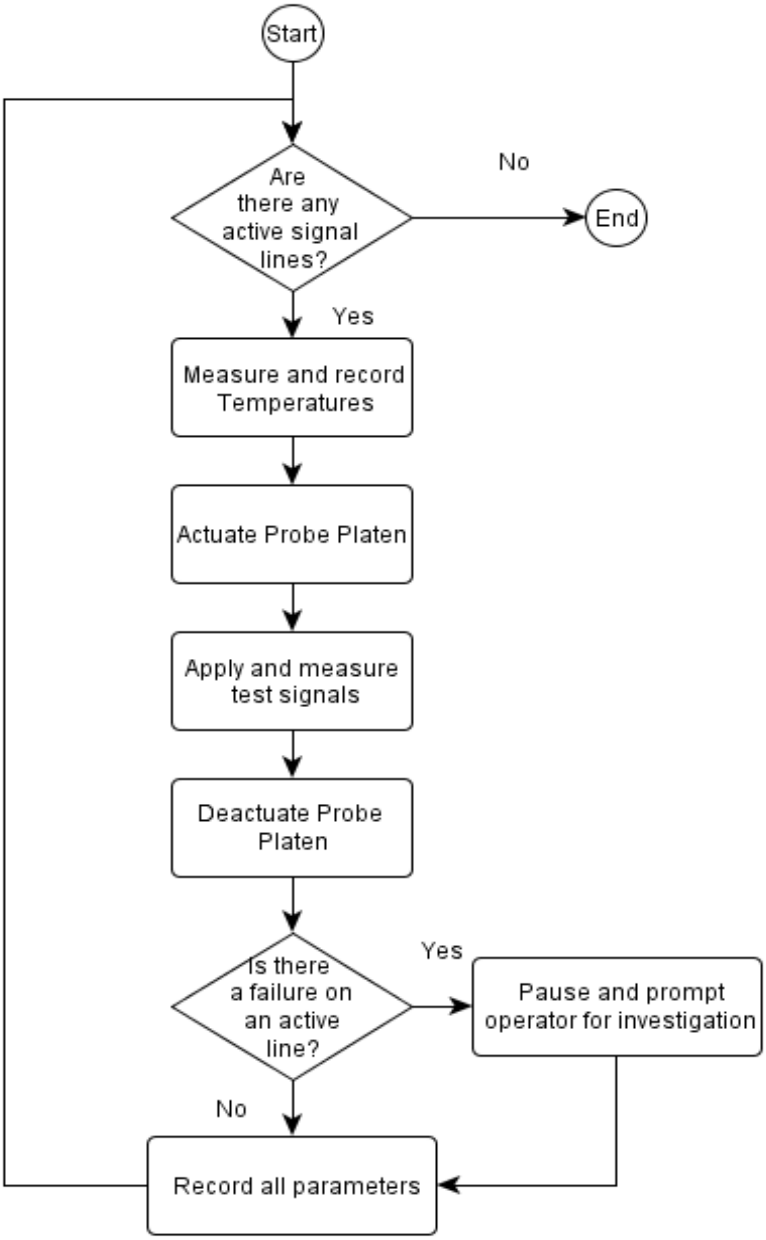


Illustration 3.14: Proposed software test flow

Assuming that 30 000 test sequences will be taken as the end of the test (if not all receptacles have failed before this point), the minimum data size will be 174x30000 = 5098 kbytes.

3.4 PRELIMINARY MODEL DESIGN

The selection model that will be created in this study will allow for the design engineer who is designing the test system to select the most appropriate receptacle for the system he is designing. To do this, the following parameters are evaluated:

1. Mounting methodology relative reliability.
2. Maintainability/Downtime in event of failure.
3. Amount of signal conditioning required
4. Likeliness of the system needing to be change
5. Price.

The mounting methodology reliability will be based on data obtained from the system described in this chapter, but the other 4 parameters will be collected on a quantitative basis by evaluating the methods as they are applied in the creation of the test system.

From this, the proposal is that the model be designed in the following way.

1. The above parameter data is collected for each of the 8 mounting methodologies.
2. For each of the above parameters the methodologies are then ranked from one to eight where one represents the best performing, and eight represents the worst performing methodology in that parameter (or the other way around with eight being the best and one being the worst).
3. The input to the model will then be from the user where he/she assigns a percentage value of importance to each of the 5 parameters above, which represent the importance of each of those parameters to the specific design problem the model is being applied to.

Once the above data is collected, the appropriate mounting methodology can be selected by generating a score for each of the 8 mounting methodologies using this formula:

$$Score = (RR)(RR_{method}) + (M)(M_{method}) + (I)(I_{method}) + (C)(C_{method}) + (P)(P_{method})$$

Formula 3.1: Preliminary model formula

Where:

$RR_{\%}$ = Relative reliability percentage importance assigned

RR_{model} = Relative reliability ranking 1-8 of specific mounting methodology

$M_{\%}$ = Maintainability percentage importance assigned

M_{model} = Maintainability ranking 1-8 of specific mounting methodology

$I_{\%}$ = Implementation ease (signal conditioning) percentage importance assigned

I_{model} = Ease of implementation (of signal conditioning) ranking 1-8 of specific mounting methodology

$C_{\%}$ = Likeniness the system will change percentage importance assigned

C_{model} = Likeniness the system will change ranking 1-8 of specific mounting methodology

$P_{\%}$ = Price percentage importance assigned

P_{model} = Price ranking 1-8 of specific mounting methodology

Of the 8 methodologies then analysed, the one with the highest (or lowest depending on how the rankings are assigned) score will be the most appropriate to the requirements with those percentage assignments that the user inputs into the model..

3.5 SUMMARY

In this chapter each of the receptacle mounting methodologies was investigated. The conceptual design of the relative reliability tester was also explored, and a method was put forward for the creation of the receptacle mounting methodology selection model.

In the Chapter 4, a qualitative analysis will be done of each of the parameters for each of the mounting methodologies, and they will be ranked according the model requirements put out in this chapter. The design of the relative reliability tester will then also be fully described.

CHAPTER 4

4 DESIGN AND DATA COLLECTION

4.1 RELIABILITY DATA COLLECTION SYSTEM (HARDWARE)

The main indication that a receptacle had failed in any of the methodologies was characterised by its inability to accurately carry the injected signal through the UUT. To this end, the electronic design of the system had that as its core measurement goal.

4.1.1 **Sample and hold circuitry**

See Annexure A for full simulation data of sample and hold circuitry. Since the signal that was being carried by the receptacles through the UUT needed to be accurately measured, a single waveform measurement was considered to be an insufficient measurement of the signal. In order to convert the waveform into a single measurement, a sample and hold circuit was developed that would allow the circuit to be gated for a fixed period (in the case of the test 100ms) to charge up a hold capacitor, measure the stored charge, and then once measured dump the stored charge to facilitate fast measurements. Since any changes

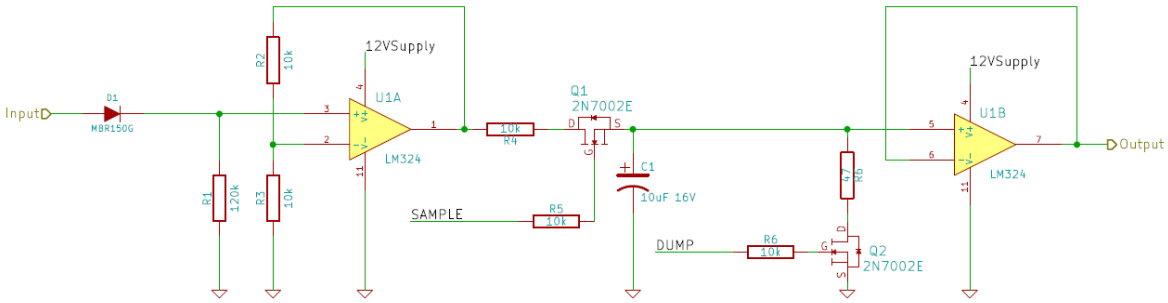


Illustration 4.1: Sample and hold circuit (see Annexure A)

in the reactance (capacitive or inductive), this will effect the measured single point value.

The input amplifier U1A is a non inverting amplifier that doubles the size of the incoming 500Hz 5V p-p square wave. The following failure conditions are considered:

a. Break in input line, series capacitance added.

b. Input line moves and a paralell capacitance is formed.

The input signal is therefor assumed to be affected as follows.

With series capacitance added

if the input line gains a series capacitance then the input will be effectively moving through an impedance divider circuit with R1. Looking at the fundamental wave and harmonics of the incoming square wave the reactance would be:

$$X_c = \frac{a}{(\pi C)}$$

Formula 4.1: Capacitive reactance (Hughes, 1995:227)

where a = 1/2f (at frequency of harmonic)

thus:

a_f = 0.0001 (at fundamental)

a_{3rd} = 0.0003333...(at 3rd harmonic)

a_{5th} = 0.0002 (at 5th harmonic)

a_{7th} = 0.000142 (at 7th harmonic)

etc.

where C is the value of the series capacitance that has formed. When the SAMPLE signal is pulled high, the output of U1A is then gated to charge up C1 through R4, and since the integral of the input signal will be less than that of a signal that does not have a series capacitance in it, the resultant single point measurement value will be lower.

With paralell capacitance added

If an input line through the receptacle somehow fails or moves so that it forms a parallel capacitor by coming into close contact with a ground wire, the effect on the circuit will be as follows. Since the output from the signal buffer is through a 74HC244, the output impedance is very low. Since the time constant to charge the parasitic capacitance will be:

$$t = RC$$

The parasitic capacitance that is formed will be charged up almost instantaneously in the high section of the incoming square wave signal. When the signal drops low however, the paracitic capacitor can only discharge through R1 which is at 120kΩ which will mean the discharge time is significantly slower than the charge time. The result will be that once the signal is used to charge the hold capacitor C1, that the single point measurement will be higher than when compared with a line with no parasitic capacitance.

Interpreting single point measurements

Since the relative reliability tester will only detect when a line fails, and not be used as an absolute capacitance meter, the amount of deflection will be used as a failure indicator rather than using the value to determine the capacitance or inductance. Using the circuit in that way falls outside of the scope of this project, but can be evaluated for further development.

4.1.2 **Temperature measurements**

Temperature measurements will be done using custom temperature probes to attempt to protect the wiring behind the probe itself. The probe is based around the MCP9700 thermistor IC from microchip (2009:1). The front of the IC is encased in thermal epoxy, with good thermal conductivity of 1.1W/mK Electrolube (2003:3) with the front of the ic as close as possible to the surface of the epoxy surface. Once the IC is encased in thermal

epoxy, the interface wires are soldered in place and the IC is inserted into a tube that is of the same diameter as the thermal epoxy front end. From there, the rear of the tube is filled with casting silicon.

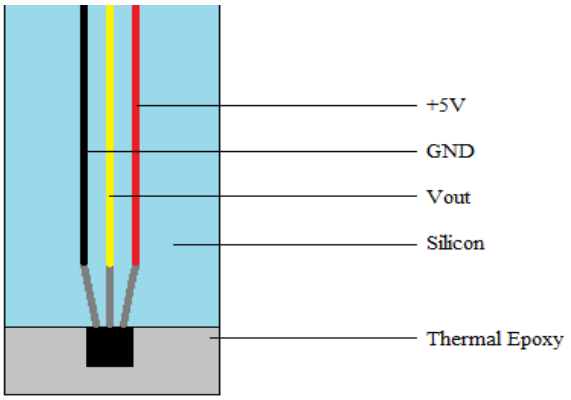


Illustration 4.2: Thermal Probe Design

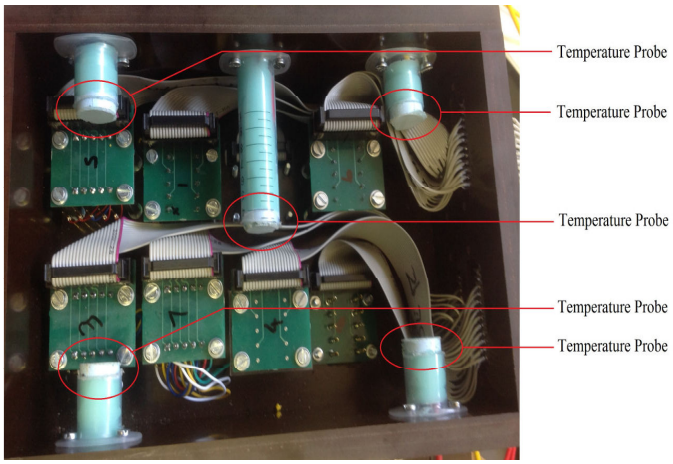


Illustration 4.3: Temperature Probe placement in chamber

5 probes were created to measure across the cross sectional diameter of the whole inner surface of the thermal chamber.

The MCP9700 output is directionally propotional to the the case temperature of the IC. At 0°C the output is 500mV and the temperature coefficient is 10mV/°C. Because no additional signal conditioning required, the output is directly connected to the Daq input through a 10k resistor.

4.1.3 **Thermal Chamber driving**

The thermal chamber is either driven to be heated or cooled. The heating system is achieved by running a number of 1W resistors in series/paralell with a heat dispersion plate on top of them. The heating resistors are mounted below the central heat dispersion sync so that when the fan is turned on the air flow is sucked over the resistors and through the heat syncing element. The cooling is achieved by running the same fan and pulling air over the resistors as well (cooling them at the same time) and then below the sync running a peltier cooler.

In the initial design proposal for the heating element the idea was put forward to use either a large resistor or an oven element to do the heating of the chamber. The problem with this idea was that a single failure of the heating element would put the chamber out of commission. Instead of using one large resistor to generate the heat, an array of 45 resistors are used as 15 sets of 3 series lines. In this way a failure of one resistor chain does not cause a complete failure of the heating element as a whole. The mounting of the resistors is also done in a zig zag pattern to attempt to place as little as possible strain on the solder joints between the resistors so that they don't fail as quickly during the temperature cycling.

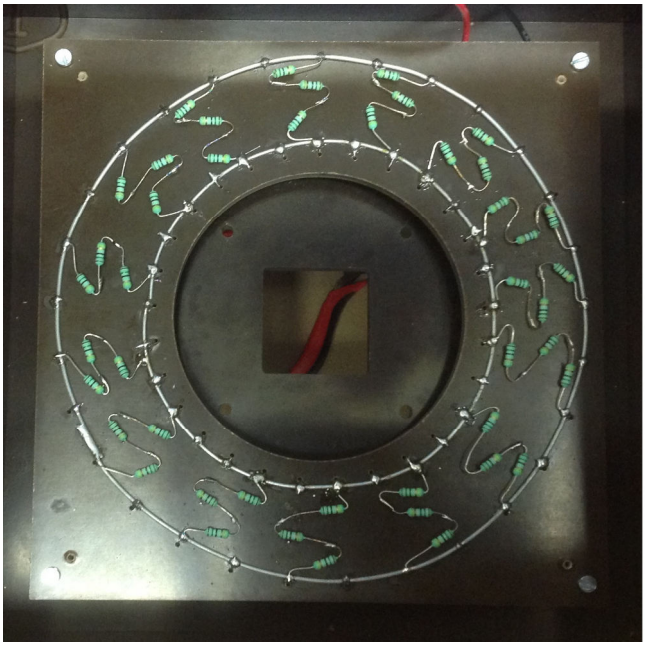


Illustration 4.4: Heating array in chamber

The Peltier cooler is fitted in the middle of the heating array to keep the heating and cooling elements as close as possible together to maintain similar thermal currents inside the chamber for both the heating and cooling cycle.

For the cooling element, A CP 1.4-127-06L Peltier cooler from Laird technologies (previously Melcor) was sandwiched between two Intel 775 socket CPU heat syncs. Two aluminium adapter plates were machined to ensure 100% coverage of the peltier faces, as well as to achieve maximum surface area contact to the heat syncs. Thermal paste was also applied to all faces to ensure proper thermal contact. The Hot side Heat sync has a fan mounted on it to cool it down as the Peltier pumps heat from inside the chamber outwards.

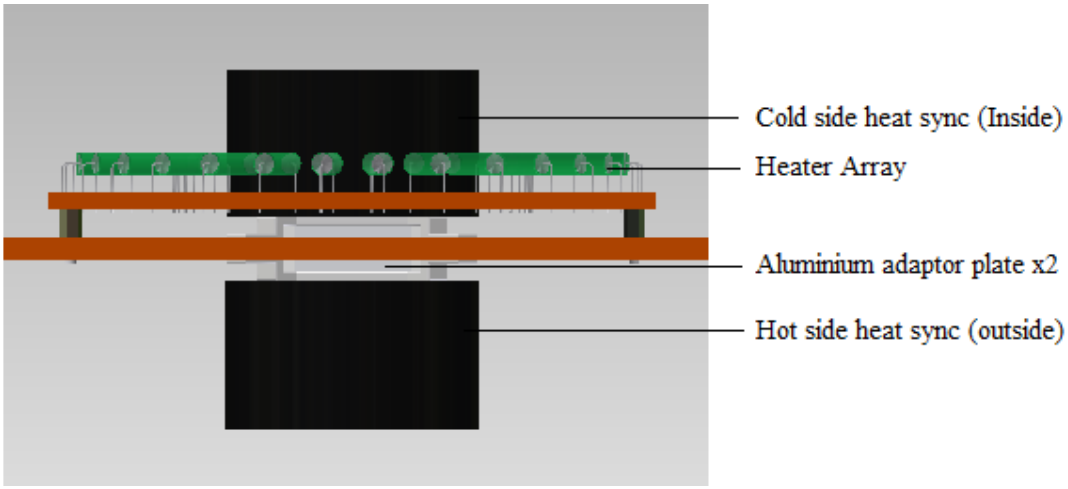


Illustration 4.5: Peltier cooler placement between heat syncs

During implementation and initial testing, the inner fan was removed from the peltier heat sync since in the above arrangement it is right in the airflow path of the heating elements, and the MTBF of the fan is lower than any of the other components of the fixture.

4.1.4 **Control Relays**

The Cooler, Fan, Heating array and pneumatic actuator are all individually controllable from the DAQ by way of the control pcb (see annexure B) through a ULN2003A darlington array which drive relays on the Relay pcb.

The Relay PCB runs two different kinds of relays, nl the FP2-D3002 and the G5LA1E12DC. The smaller FP series relays are used to control loads that need to be run simultaniously and are controlled as a single load – nl the Heater and Heat fan set, and the Peltier and cold fan set. In this instance the Darlington switches the smaller DPDT relay, and this relay actuates both the larger relays in turn. Where a single load is driven, the relay is actuated directly (nl. The Actuate relay).

4.1.5 **Signal Generation**

The Signal used to drive the receptacle chain is a square wave. To ensure that the signal fed into groups of receptacle chains (co-ordinated with the channels on the sample and hold circuits) are in receiving the same signal in phase, the signals are fed to the chains in paralell.

To ensure as sharp as possible a square wave, the signal is generated by the TDA019D Daq, and then fed into a 74HC244 buffer IC, and each of the receptical signal chain is run through its own diode to ensure they don't interfere with each other.

The Buffer chip adds the ability to place the outputs into a high impedance state (by disabling the “enable” pin of the IC) which can be used to isolate the capacitance stored on the receptical signal chain (which will be present due to the paracitic capacitance) since there will be a high impedance load on each side of the capacitance in that instance (the operational amplifier in the output side, and the high impedance buffer on the input side).

4.1.6 **Interfacing**

Using the information gathered in Chapter 3, each of the interfacing methodologies is implemented seperately on its own PCB. The probe layout is identical for each of the methodologies so that identical sacrificial UUT PCBs can be used.

The sacrificial UUT board is also implemented on a PCB. Each UUT has 5 sets of parallel tracks with an input terminal and output terminal so that two signal chains are tested at once.

Test probes are placed 12.7mm apart, and separate parallel lines 5.08mm apart. The reason for this spacing is twofold. The 5.08mm spacing is due to the fact that 100mil is the

minimum recommended spacing for 100mil probes (the reason they are designated as such), and the 12.7mm is to key the board so it is inserted correctly into the fixture. The UUT board also has provision made to mark out the number of test cycles the board was subjected to, as well as marking the date it was run so it can be easily identified later.

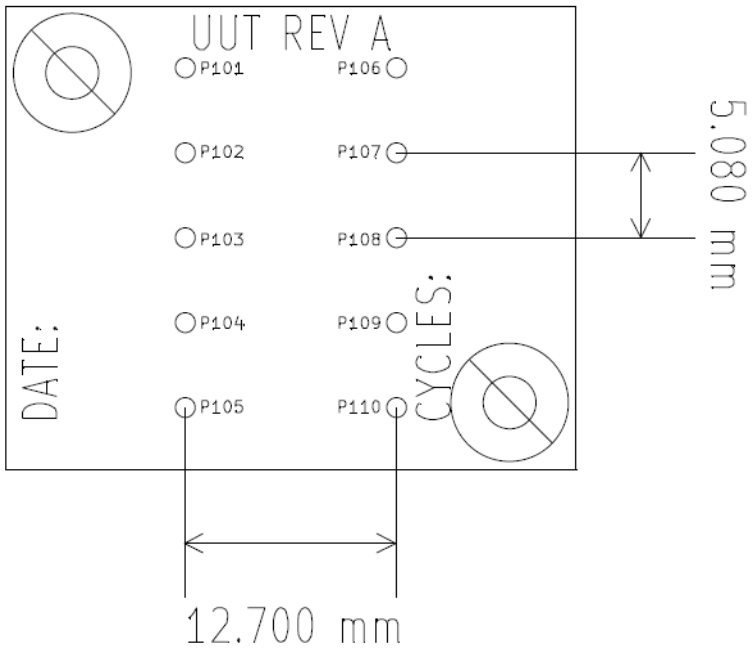


Illustration 4.6: UUT Board indicating board markings and spacing

4.1.7 **Pneumatic control**

The pneumatic ramps used to actuate the test probes the required 6mm are custom made pneumatic ramps purpose built by TFT for test probe actuation. The ramps are actuated by a pneumatic value driven through the Relay board, and is powered by 6 bar of compressed air.

The mechanical design of the relative reliability tester allows for chamber housing the test probe receptacles to be independently driven by supporting the chamber solely on the 4 guide pins onto the ramps and using 4 guide pillars in bushes (mounted on the PCB support plate) to keep movement exactly vertical.

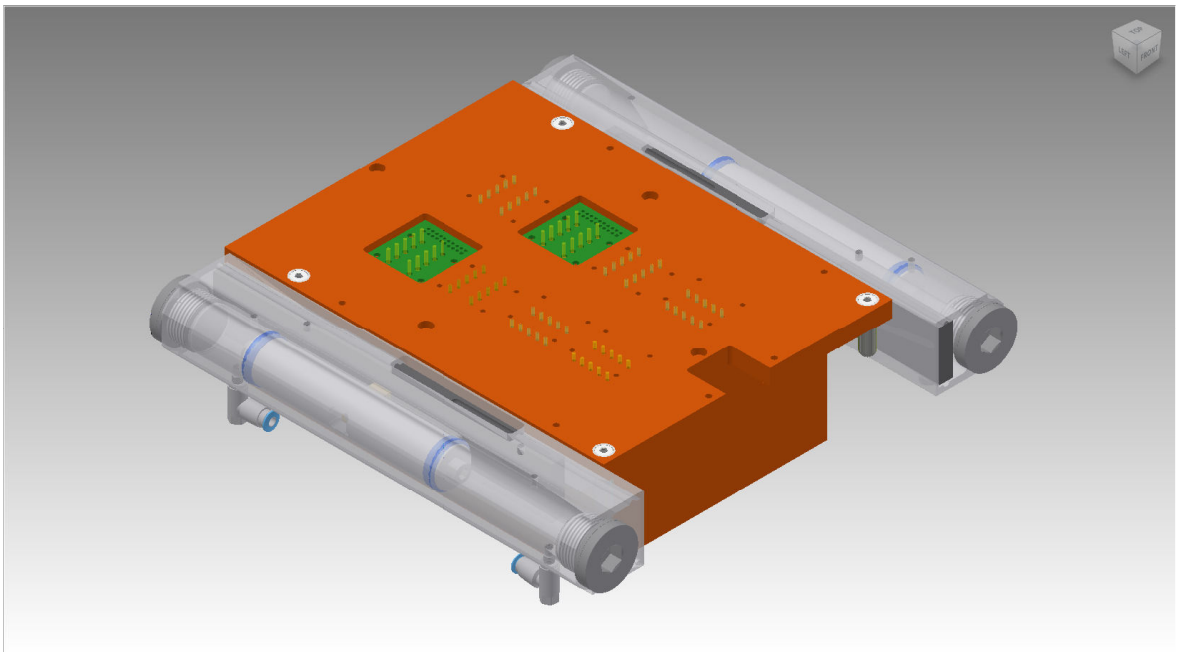


Illustration 4.7: Chamber housing receptacles (with pneumatic ramps for reference)

Overleaf are images of the fully assembled custom electronic section of the system as well as the mechanical assembly of the full system:

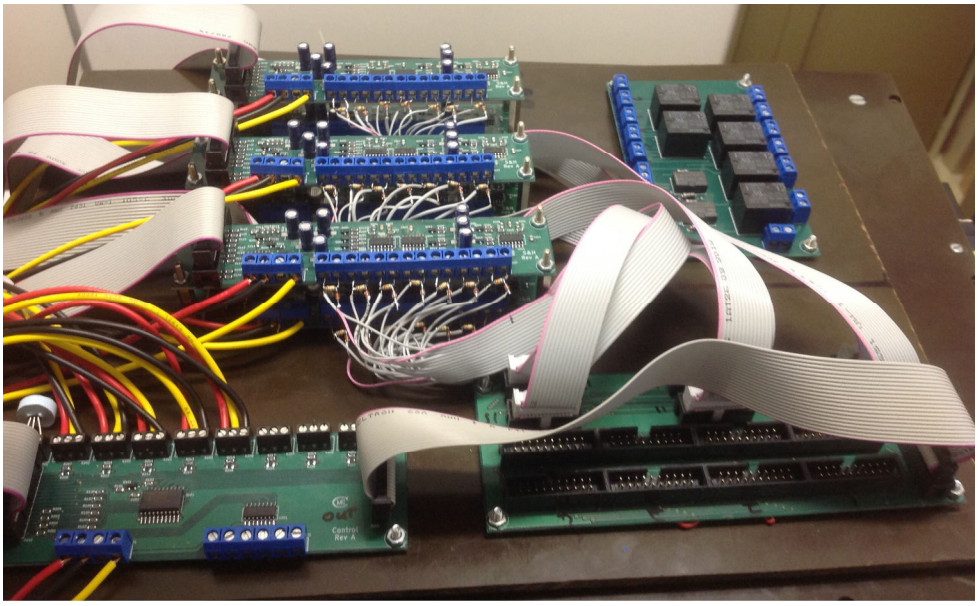


Illustration 4.8: Sample and Hold and custom electronics

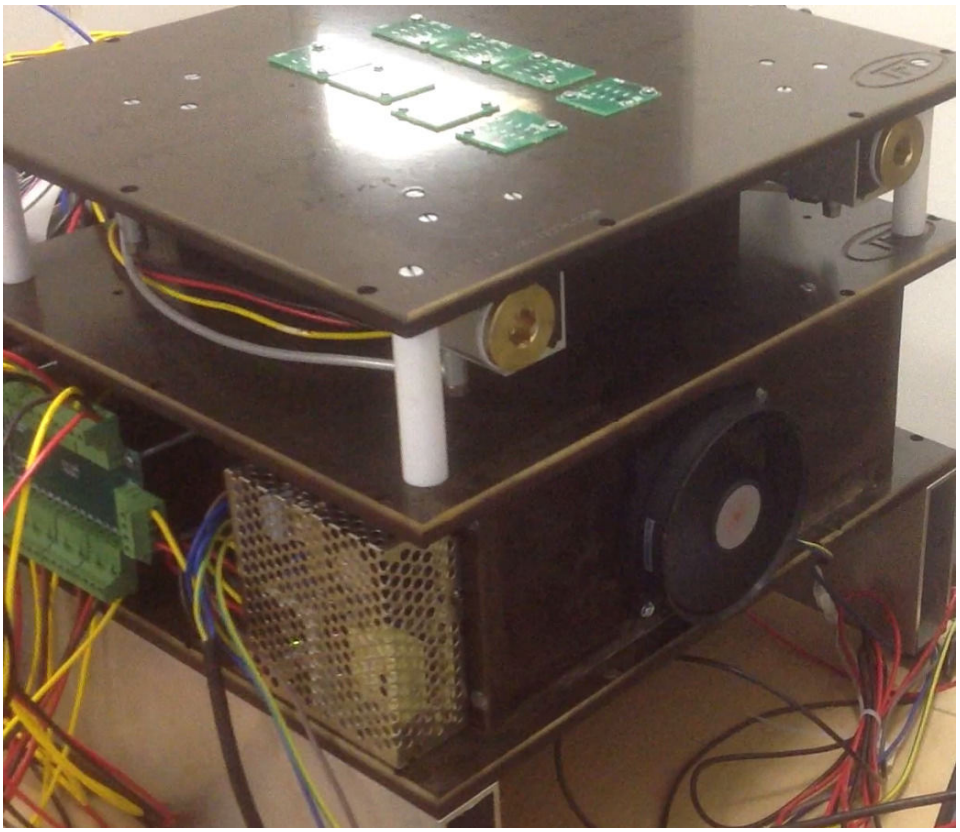


Illustration 4.9: Mechanics outer view with power supplies

4.2 RELIABILITY DATA COLLECTION SYSTEM (SOFTWARE)

The Software is written in labview/Teststand. The software is packaged as an exe with a local database for storage of the test results. Code documentation can be found in Annexure C of this thesis, and the code can be found on the accompanying CD.

4.2.1 **Labview VI design**

The software for driving the Relative reliability tester's architecture can be found on the following page.

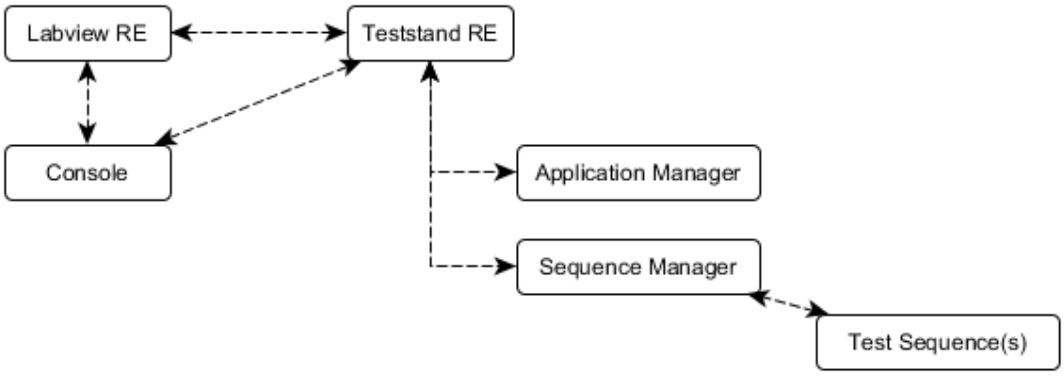


Illustration 4.10: Labview/Teststand overhead architecture

The main components are as follows:

- Console application – this is a labview application that runs against the Labview Runtime engine and has access to the Teststand engine. The console application allows the operator to trigger events to start the test sequence, stop it, pause it etc, as well as will display any required info to the operator during runtime.

- Labview Runtime Engine – allows standalone applications written in labview to run and handles the low level interactions with the PC. Events triggered in the runtime engine will propogate through to any applications running an eventing architecture (like the console application) – this is used extensively in this project.
- Teststand Runtime Engine – the test executive. Allows the console application through the Teststand API to control the test flow. Also will trigger events in the Labview runtime engine (which can then propogate through to any applications running against it). The main components in the TRE are the application manager and the sequence manager. The application manager handles the life of the engine and any sessions running against it, and the sequence manager loads and unloads sequences as well as allowing them to be executed (againse the TRE).
- Test Sequence(s) – the test sequences are linear scripts that follow a strict form of Startup, Main loop and cleanup. During execution, a sequences startup section runs first, which includes things like setting up instruments, creating log files, etc. Then the main section runs, which can either be run through once for a single test device or multiple times throughout the test session. The main sequence area contains the main body of the code. Then finally once the test session ends, the cleanup area is run which includes code to clean up the session, close off file handles etc.

The Teststand API also allows the sequence through the sequence manager to trigger events in the Teststand Runtime Engine and the Labview Runtime Engine. The graphical dashboard of the VI can be seen overleaf.

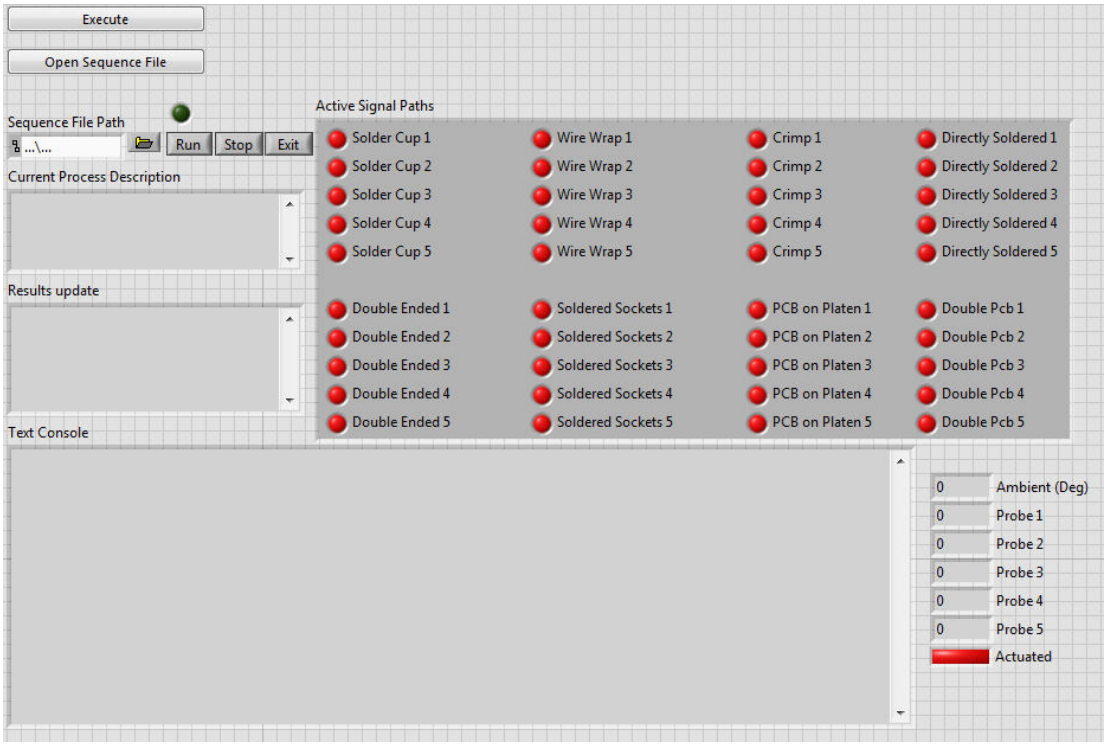


Illustration 4.11: user interface of the custom VI

4.2.2 **Teststand Sequence design**

The sequences written in the teststand IDE and run against the teststand engine form the heart of the software as they control the flow of the testing. The system contains 6 sequences – One master sequence that will spawn two seperate threads that each house a sequence, and three in line sequences that are used in the main sequence to control the active and inactive parts of the tests as well as the actuation sequence. See Annexure C for the code documentation for the sequences, Annexure D for the flow charts of all of the above sequences and the accompanying CD for the actual test sequences which can be found in “TestSequence.seq”.

4.2.3 **Running the application**

When the console application is started up, loads the sequence into the sequence manager, and then awaits the users input to trigger the start of the testing session, registers the console application for all of the appropriate events (which are user generated from the console) and messages (which are application generated from the teststand engine and from the sequence), and then allows the teststand application manager to start.

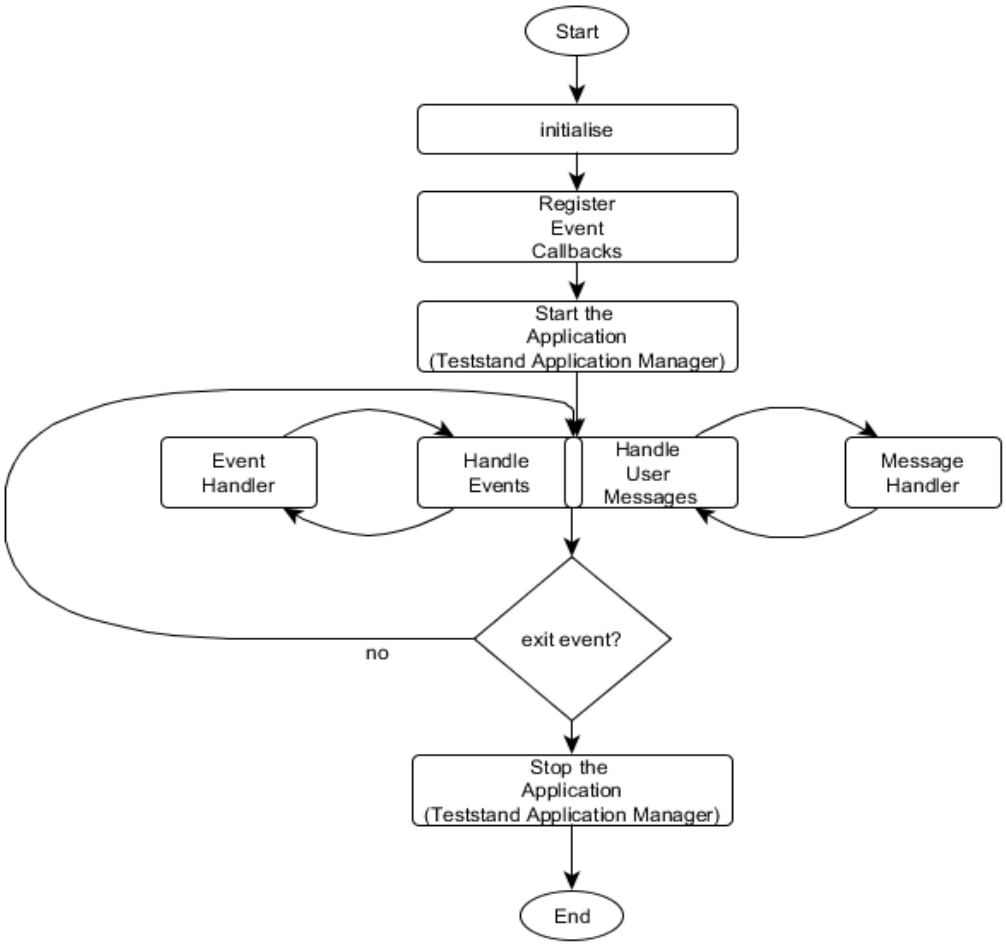


Illustration 4.12: Application life cycle overhead view

Once all of the applications are started and all events registered for, the application goes into an event monitoring mode where it will sit in a loop and await events from the user in the form of front panel interaction, or messages from the teststand engine.

The events that are registerd for are as follows (see consoleTopLevelVI.vi):

- Open button pressed – runs the “openSequenceFile” method on the Application manager. This is called to open the sequence file “testsequence.seq”.
- Execute button pressed – runs the “execute” command on the sequence file manager.
- Stop button pressed/panel closed – runs the “shutdown” command on the application manager (stops the current sequence and closes the reference).

The messages that are registered for are as follows (see HandleMessage.vi):

- Heartbeat – this is triggered from the heartbeat sequence thread and merely triggeres the message once a second to indicate to the operator that the application is running and there are no errors present by flashing the heartbeat indicator on the operator console.
- Update Console – appends a string to the console text area and automatically scrolls it when data is sent from one of the sequences. Since the message is run against the LRE and triggered from the teststand API, any of the sequence threads can update the console independently and don't have to worry about access error conditions since messages are parsed from a queue and one at a time – the console is therefore inherently thread safe.
- Update test description – similar to the console message above, but the test description is triggered only from the mainsequence thread so that it can indicate to the operator what the current test is being run.

- Update Results – similar to the console message above, but the test results are posted every time a test is completed.
- Clear console/Clear test description/clear results – three seperate messages which allow for the clearing of any of the text output areas from any of the sequence threads.
- Update Active Signal Chains – the signal chains that are currently active (and still part of the test) are indicated to the operator at all times. When a chain fails and is there for disqualified from the test, this needs to be updated so that the operator can see what the state of the test is. A chain is only removed from the active list once it is established that it is a legitimate failure of the signal chain, as opposed to a failure of a consumable component (Ie. The UUT boards/probes).
- Actuate pneumatics – indicates that the pneumatics have changed state, and updates the console so that the operator can see what state the fixture is in.

4.2.4 **Message flow during runtime**

As seen above, during runtime, messages are sent from the sequence to the console application and events are triggered back and forth between the LRE and TSE. The diagrams below indicate the flow of information when events are triggered from the console application which the sequence as the end destination, as well as when messages are triggered in the opposite direction when they are generated in the sequence and then sent with the console application as the final destination.

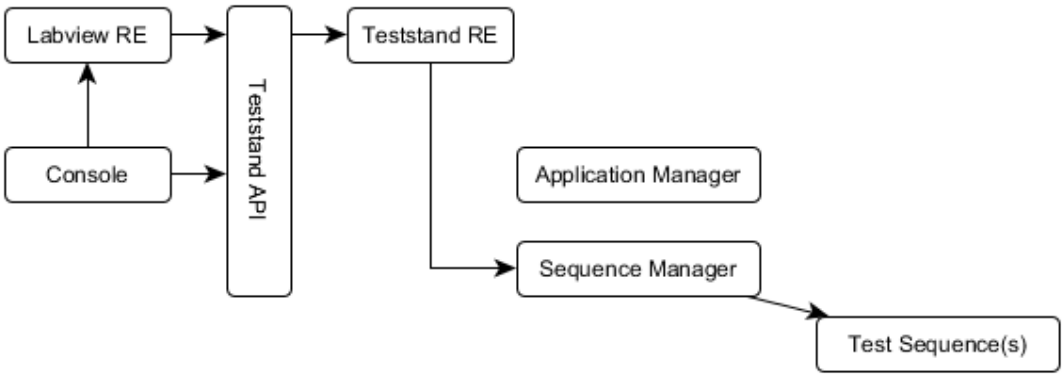


Illustration 4.13: Console generated event flow

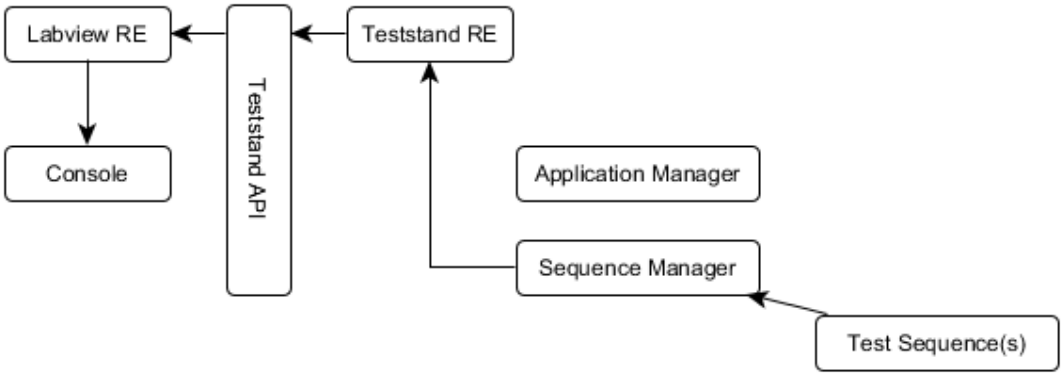


Illustration 4.14: Sequence generated message flow

4.2.5 Drivers

Teststand sequences have the ability through the LRE to also call labview vis directly in order to control the flow of the test sequence. In this project, the drivers for the fixture are all controlled directly by the test executive – so information is displayed to the user through the console, but the actual low level control is all applied to the Daq devices directly through the LRE from the sequence.

All of the interactions with the jig are done through the TDA019 DAQ devices, and as such is all applied using the DAQ's protocol. Since the TDA019 DAQ series does not support VISA, the code was written to interact with it directly. The code for the protocol can be found in TDAJigDriverV2.vi. The following table shows the seperate parts of the protocol messages:

Start	Length	Source	Destination	Type	Msg ID	Data X	Check Sum	End
(a)	variable	0 (rsvd)	Daq Slot	0 (rsvd)	variable	variable	calculated	(c)

Table 4.1: Message structure

- Start and end characters are fixed as ascii “a” and “c” respectively.
- The length of the message is calculated as the number of bytes after the start character up until the end character. So if the entire message is held in an array, the length would be (arraySize -1)
- The destination of the message is the daq slot number from 1 – 8
- the message id can be found in the daq manual
- the data size is dependant on the message
- checksum is calculated as follows (pseudo code) :

```
char SerialComms::chkSumCalc( char buffer[], char size, char arrayOffset){
  char chkSum = 0;
  for(int i = 0 + arrayOffset; i < size + arrayOffset; i ++) {
   chkSum+=buffer[i];
  }
  chkSum = ~chkSum;
  chkSum = chkSum + 1;
```

return(chkSum);

} *(TDA 2013:7)*

4.3 QUALITATIVE PARAMETERS

In the creation of the Receptacle Mounting Methodology Selection Model, the following qualitative parameters are taken into account:

a) Maintainability

b) Ease of implementation of signal conditioning required

c) Likeliness that the system will change

d) Price.

Each of these parameters is considered in depth, and then analysed for each of the mounting methodologies.

The methodologies are also split in some parameters between wired (wire wrap, soldered, crimped) and PCB based (all methodologies which use a rear or front mounted PCB) methodologies since these two groups also have their own advantages and disadvantages that are more generic to the group.

4.3.1 Maintainability

The maintainability of the receptacle relates to how much downtime will be experienced should a failure occur on one of the test probe receptacles. The failure could require merely resoldering of a wire, or the catastrophic damaging of the receptacle itself, so this is not a measure of one single failure mode, but rather a general measure of how easy it would be to fix a problem should it arise. For each of the receptacle mounting methodologies the same failures will be taken into account.

A failure that requires maintenance of the receptacle will be defined as a component failure that result in an impact on the signal integrity of the signal that is carried through that receptacle when that component is integral to the connection methodology. For example, all of the methodologies discussed will require a test probe inserted into them to connect to the UUT, so the test probe will not be a component that is specific to the mounting methodology, but if a wrapped wire snaps from the wrappost of a wire wrap receptacle, this will be considered a failure.

4.3.2 Ease of Implementation of Signal conditioning

Assuming signal conditioning is required (if it is not, then the percentage importance assigned to this parameter can be 0%), then this parameter describes how easy it is to add that signal conditioning to the system and interface it to the receptacles. The assumption is made that if there is signal conditioning to be done, that it will be done on a dedicated PCB, so the cost of tooling and manufacturing this PCB is not taken into account. For methodologies that directly interface to a backplane PCB, this will be assumed to be where the signal conditioning is done, and as such, the amount of space required for the mounting methodology compared to PCB real estate will be taken into account.

4.2.3 Likeliness of system change

This parameter describes how easy it will be to make changes to an already commisioned system, be it completely rewiring that system, or making a single change on a receptacle to take it to a different point. This could be as simple as unsoldering one wire and soldering it somewhere else, to having to replace the entire probe platen with a whole new set of receptacles and a new wiring board. The qualitative measure therefor is a combination of

all of the above spectrum of scenarios as observed from the creation of the reliability data collection system.

For methodologies that require a wiring harness (wire wrap receptacles, soldered receptacles, crimped receptacles), military specification MIL-C-545224D for cable and wiring harness assemblies will be used as a guide for the creation of those harnesses and the assumption is made that should that receptacle mounting methodology be chosen that the same specification will be followed.

4.3.4 Price

The price parameter is not just a relative ranking of the pricing of different receptacles, but more holistically the price of implementation of the mounting methodology. So prices taken into account are the receptacle pricing, the amount of machining required to create the probe platen, the cost of interface wire and wiring boards etc. For the different receptacle mounting methodologies, the following tools are a requirement:

Tool	**WW**	**S C**	**BPsr**	**BPde**	**BPds**	**T**	**D**
Wire Wrap Tool	√						
Crimp Tool		√					
Soldering Iron		√	√		√	√	√

Table 4.2: Tool usage across methodologies

In the model, provision will be made to include the cost of the tool if it is not already available. Again, the rand to rand comparison isn't done of each of the parameters, but

rather the ranking is done based on evaluating all of these separate costs for each of the methodologies.

4.3.5 Implementation speed

Different methodologies can be physically implemented at different rates. On system that has a small number of test points and is only implemented once, but if a system needs to be wired a number of times for many test station and contains a large number of test points this parameter can become significant. The assumption is made in the analysis below that a hand manual method will be used for all assembly, and robotic soldering stations, automatic wire wrapping machines etc. will fall outside of the scope of this project. The speed of implementation is ranked separately for wired and PCB based methodologies since they are offer different advantages and disadvantages.

4.4 QUALITATIVE PARAMETER DATA COLLECTION

For each of the receptacle connection methodologies, the above 5 parameters were analysed, and each connection methodology ranked based on the experiences from the relative reliability data collection system. For each of the mounting methodologies, it is assumed that a probe platen can be machined on a CNC machine, so the amount of machining is used in the pricing parameter. It is noted that the qualitative parameters collected are based on the author's experience in implementing each of these methodologies for the purposes of this project. If someone else was to do it, there may be a difference in the rankings for the parameters based on that person's individual skill set and experience.

4.4.1 Wire wrap method

The wire wrap method requires a special tool in order to do the wrapping but this is a simple method of connection. For a small number of probes this is a good methodology, as it is simple to wire, cheap and easy to maintain. For large number of probes however where there are a large number of wires that need to be maintained, this will be a problem since as the wire density increases, the ability to physically reach the receptacle back end becomes difficult. The advantage though, is that (unlike solder cups) the tool isn't heated, so there is less chance to damage the wires that are connected to other receptacles during wiring and maintenance. The same applies with the ability of the system to change – if the number of receptacles is small, then the change will be very simple to make, but if there are large numbers of receptacles, the change could be very difficult to implement (if another test point needs to be added, it could require a complete change of the probe platen as the drilling process could easily damage all of the wiring). It is important to follow the guidelines set out in MIL-C-45224D to minimise this disadvantage.

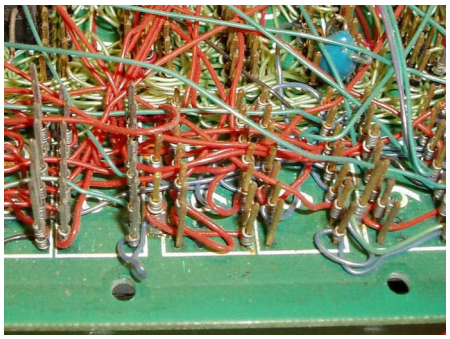


Illustration 4.15: wire wrap example (wikipedia, wire wrap)

The wire wrap receptacle is friction fit into the probe platen, and as such, the amount of machining is minimal. Signal conditioning will have to be applied on a separate board, and then wired onto the wrapposts so it offers no inherent benefits compared to other methodologies. The wire wrapping of wrapposts is done once they are friction fit into the probe platen, and once they are in place can be done quite quickly.

4.4.2 Solder cup method

Wiring of solder cups requires no special tooling (other than a soldering iron), so it is a very simple method to implement. Should a large number of probes be required however, it could be very difficult to maintain, as in the case of a breakage to solder onto a solder cup receptacle that has wires running close to it could be difficult without damaging those wires. This can be mitigated by using wires with a temperature resistant silicon insulation but this does increase the overall cost of the system. The converse benefit however is that a change to a wires position can be done without any special tooling as desoldering and then resoldering the wire is a simple procedure.

The solder cups suffer in the same way as the crimp and wire wrap receptacle methodologies in being difficult to change as drilling a new receptacle will be difficult to do without possibly damaging all of the wiring harnesses at the back, so it is not a very suitable methodology for systems that are likely to change (though a lot more suitable than the crimping methodology). It is important to follow the guidelines set out in MIL-C-45224D to minimise this disadvantage. Properly soldering the solder cup receptacles is slower than either the wire wrap or crimping wiring methodologies since the wire is first stripped, then pretinned, then the solder cup is pretinned and then the solder connection is made to ensure a good contact. There is also the safety aspect to consider.

Signal conditioning if required will have to be done on a seperate board, so this mounting methodology offers no inherent benefits to applying signal conditioning.

4.3.3 Crimp connection method

Wiring of crimp receptacles requires a special crimping tool to wire. They are very quick to wire, but suffer from the same wiring disadvantages as the other wired methodologies (wire wrap and soldered methodologies) and thus offer no inherent benefit to signal conditioning implementation. An additional disadvantage of the crimp receptacles is that they cannot be repaired in the same way as the soldered or wire wrap methodologies and since they are friction fit into the Bakelite probe platen, to change the system may require making an entirely new probe platen. Although they cannot be rewired, in the same way as soldered or wire wrapped receptacles, if a new receptacle needs to be added, the wiring should conform to MIL-C-45224D to simplify this process.

An advantage of the crimping process is the speed in which a test system can be wired. Since the crimping can be done before the receptacles are hammered in place into the probe platen, this means that the wiring can mostly be done outside by making off the required number of receptacles first and then placing them all at once. It is faster than either the soldering or wire wrapping processes.

4.4.4 Back plane connection with soldered sockets method

The back plane soldered receptacle methodology is slow to assemble and requires a specific technique. In order to assure that all of the PCB mount header sockets are alligned with the wire wrap receptacles in the probe platen, the header sockets are first fit onto the receptacles. After all the sockets are fit, then the PCB is fit onto them and they are all

soldered in place. This is a time consuming process, but it ensures that the PCB can be removed and refitted onto the receptacles.

As with other PCB based methodologies, it does offer some advantage to being able to fit signal conditioning on the back of it. This is slightly different though to the other PCB based methodologies, since the through hole nature of the header sockets means that a through hole needs to be placed on the mounting board that is large enough to accommodate the socket. The holes used in this project are 0.81mm with an annular ring of 1.4mm, which offers the same size constraint as the. Sockets also need to be chosen that are appropriately sized, since if the header socket is larger in one of its dimensions than can be accommodated by the 2.54mm spacing of the 100mil receptacles, this could mean that receptacles cannot be placed next to each other if the test points they need to contact are at that spacing.

The maintenance of this methodology is very simple, since it is possible to remove the rear PCB, it means that if there is a solder point failure, or the rear PCB gets damaged it can be completely replaced in a short period of time. The same applies with changes to the system as by simply making a change on the rear mounted PCB the changes are not only made but documented. It is slightly harder to replace the rear PCB on this methodology when compared to the double ended receptacle mounting, since the header sockets also apply a large amount of grip to the receptacle rear ends so with larger numbers of test points will cause difficulty.

4.4.5 Back plane connection with double ended receptacles method

Once implemented, the Double ended receptacle mounting methodology is the fastest to assemble and disassemble. The main disadvantage of the methodology though is the cost, since not only are the double ended receptacles very expensive when compared to other receptacle types, but there is an additional alignment block needed to ensure the connections are made in the same place every time with the small rear mounted probe.

The advantage of this methodology is similar to the other PCB based methodologies, with the added advantage however that the pad used to press the double ended receptacle into on the signal conditioning board is much smaller than any of the other methodologies, meaning that routing and signal conditioning implementation on this board is the easiest to do, and also that receptacles that are smaller than 100mil can be easily accomadated. Of all the methodologies investigated, this seemed the most appropriate for boards where large amounts of signal conditioning would be done.

Maintenance and change management are also very simple to accommodate because of the ease of assembly and dissasembly of the system, but the cost of the methodology is a big disadvantage.

4.4.6 Back plane with receptacle soldered in place method

Together with the top mounted PCB on Probe platen method, this is the cheapest of the PCB mounting methodologies to implement. It has the advantage of having the second most amount of space to place receptacles (after the double ended receptacle methodology), since the wire wrap receptacles can be spaced closer together than the header sockets, but they do still need slightly more space than the pads used for the double

ended receptacles. It is also very easy to assemble since the wiring board slides over the header pins easily and is soldered in place just a simply, but has the disadvantage that once the PCB is in place it becomes very very difficult to remove since every pin will need to be desoldered again to remove it. The price is the major advantage in this methodology, but it suffers greatly when change management and maintenance are required.

4.3.7 PCB on Probe platen method

The Top mounted PCB on probe platen is as cost effective a PCB mounting methodology as the Back plane with receptacle soldered in place, but has the added advantage that since the receptacles are not rigidly mounted in the probe platen, it is as easy to assemble and disassemble the system as the double ended receptacle mounting methodology.

The disadvantage of this methodology however when compared to all the other PCB based methodologies comes in with the signal conditioning. Since the board is mounted right below the support base that holds the UUT, no high components can be added for signal conditioning purposes and in addition to that, the hole required to fit the receptacle body is significantly larger than any of the other required holes (An annular ring of 2.7mm was used with a center drill size of 1.9mm).

Maintenance and change management are very simple, since the system can be changed very easily by just removing the retaining screws and replacing the board assembly. It does also open up the oppurtunity however to have multiple versions of the same UUT, or slightly different UUTs to be accomodated since the receptacles are removed at the same time as the board, which is an advantage shared by only the Double PCB methodology.

4.4.8 Double PCB mount method

As with the PCB on probe platen methodology, the Double PCB mount methodology does have the disadvantage of being able to swop out the Receptacles at the same time as the PCB, which means that multiple versions of a UUT can be supported on the same Fixture. The additional advantage however, is that other than all other methodologies, the Double mounted PCBs mean that no actual prober platen is needed, and the boards themselves give the receptacles stability.

Signal conditioning implementation is difficult because both the top and bottom PCBs accommodate the body of the receptacle (An annular ring of 2.7mm was used with a centre drill size of 1.9mm), but since there are 2 PCBs instead of just one, this is not as much of a disadvantage as the PCB on probe platen method. Although swopping out the double PCB mount methodology boards is simpler than many of the other methodologies, the actual implementation of the boards is much more time consuming than with many of the others when the PCBs are used to replace the probe platen as seen in Greeff (2010, 81).

4.5 QUALITATIVE PARAMETER DATA RANKING

Following from the previous section, the experiences in implementing each of the methodologies with reference to the above parameters are ranked from 1 – 8 for each of the methodologies compared to each other (where 8 is the best, and 1 is the worst).

MOUNTING METHOD	ABILITY TO MAINTAIN SYSTEM	EASE OF SIGNAL CONDITION	ABILITY TO CHANGE SYSTEM	PRICE	SPEED

Wire wrapping	4	1	5	8	3
Solder Cup	2	1	4	7	2
Crimp Connection	3	1	1	6	4
Back plane with soldered receptacles	6	6	7	2	1
Back plane with double ended receptacles	7	8	8	1	4
Back plane with receptacle soldered in place	1	7	1	4	3
PCB on Probe platen	8	4	8	3	3
Double PCB mount	5	5	6	5	2

Table 4.3: Qualitative parameter ranking

4.6 PRICE DATA COLLECTION

On the following page a table is presented that shows current prices for the various components used in the implementations of the methodologies, as well as the prices for the required tooling. Prices were obtained from the following suppliers:

1. Test fixture technologies (receptacles)
2. Communica (Soldering iron, crimp tool, wire wrap tool)
3. Digikey (SAM1213-01-N PCB receptacle)
4. WH Circtuits (PCBs)

Prices are based on current numbers (accurate at time of doing the project), but these prices will be adjustable in the final model to allow for the values to be changed.

COMPONENT	**PRICE**
Crimp receptacle	R8.50
Wire wrap receptacle	R8.60
Solder cup receptacle	R8.60
Double ended receptacle	R34.56
SAM1213-01-N receptacle	R8.58
Wire wrap tool	R389.00
Crimp tool	R620.00
Soldering iron	R450.00
Pcb Tooling (excluding per cm^2 cost)	R1,100

Table 4.4: Price table

Note: Prices current during August 2014

Some of the prices impact multiple methodologies (for example, the wrappost wire wrap receptacles are used by the wire wrap, receptacle soldered into pcb and the header soldered into pcs methodologies), and some of the tooling is specific to just one kind of methodology (for example a wire wrapping tool). This means that in the model it is important to note whether the user actually already owns the required tool, because if he does, that cost doesn't need to be taken into account (and owning a soldering iron, but not a wire wrap tool may mean that a solder cup methodology will be more suitable if price is an important parameter in the decision making process).

Also, the PCB based methodologies work on the assumption that signal conditioning needs to be done, but if that is not the case (for example if it is only used for wiring), then the PCB still needs to be created, but only to connect to the receptacles. In that instance, it may be more suitable to use a wired methodology if price is an important parameter in the decision making process.

4.7 SUMMARY

In this chapter the relative reliability testing system was discussed in detail and the qualitative parameters were collected for each of the mounting methodologies.

The hardware design was discussed in detail and different options were explored (but finally discounted due to budgetary constraints). The software design is also shown, and the final outputs of the hardware and software design processes are shown through photographs and screenshots. Although the choices made during the design phases in both sections met with the requirements of the project, it is important to note that these choices were not explored completely and there are a number of different ways in which the same goals could have been achieved as could be seen by the other hardware systems discussed, or by using a different language entirely for the software (The same APIs used could also be called from C/C++ and the same goals achieved). It is therefore the author's opinion that the work describes a complete project, but is not a full overview of the field of test system development.

In Chapter 5 the data collected from the relative reliability tester is collected and analysed, and a more comprehensive receptacle mounting methodology selection model is presented.

CHAPTER 5

5 RESULTS AND CONCLUSION

After extensive testing of the relative reliability tester to ensure that the concept will work, the tester was finally employed to do a long term stress test of the receptacle methodologies.

The final test was started on the 13th of November 2014, and completed on the 17th of November 2014. The test sequence followed is described extensively in chapter 4 and the Appendices. This chapter follows the analysis of the data obtained, as well as the conclusions drawn from it. Over the course of the test, the temperature chamber was cycled between 25 and 55 ºC, and the probe platen was actuated (and therefor the probes, UUTs, and recepticals stressed) 44 632 times.

5.1 RESULTS – RELATIVE RELIABILITY TEST

5.1.1 Test Times and data logging, Initial conditions.

During the test, there were 5 events that forced the system to be stopped so that something could be changed or fixed. The test started running at 19:09 13/11 and was ended at 07:26 17/11 for a total running time (not counting stoppages) of 3 days, 03 hours and 35 minutes. In this time, the full test sequence was run 5 579 times, and this equates to 44 632 impacts (as during each exercise sequence the probe platen is pneumatically actuated 8 times (once for each kind of receptacle mounting methodology). The table on the following page highlights the times were when the system was not running either due to failure or manual stoppage to do maintenance as well as the reason for the down time.

Stop Time	Restart Time	Down Time	Reason:
13/11 22:39	13/11 23:01	42 mins	Had to move the running application to a different laptop
14/11 09:45	14/11 10:24	39 mins	Paused the test to remove equipment from lab, and move the pneumatic hoses around.
14/11 14:16	14/11 15:02	46 mins	Air supply hose burst – had to be repaired.
14/11 04:31	14/11 04:33	2 mins	Accidentally unplugged the USB cable to the DAQs while making the off site backup.
16/11 02:42	16/11 08:15	06 hours 33 mins	Labview memory error - "insufficient memory to complete current operation"
Total Stoppage Time		08 hours 42 mins	

Table 5.1: Test system downtime

Every time the automated test system is started up, a new test log is created. The raw data collected over the course of the test is stored in the following files:

1. Log_2014_11-13_07_09PM.csv
2. Log_2014_11_13_11_01PM.csv
3. Log_2014_11_14_10_24AM.csv
4. Log_2014_11_14_10_24AM.csv
5. Log_2014_11_14_04_33PM.csv
6. Log_2014_11_16_08_15AM.csv

To create the graphs below from all of the above log files, the log files were appended together using the AppendLogs.vi. The resulting appending of all of the log files was stored as Append.csv.

The memory error unfortunately gave inconclusive results – the error reported that Labview didn't have sufficient memory to complete the operation it was busy with (error message found in the morning), but in the Log file, the temperature measurement thread had failed at 16/11 02:11 and stopped logging data, but the main sequence and heartbeat continued for another half an hour to 16/11 02:42 before the application was forced to stop by the Teststand engine. The temperature thread failed during a heating cycle, so when the test was restarted in the morning, the lab as well as the inside of the chamber had been heated for a long time – this is probably what caused the failure of one of the 15 heating elements.

After the initial tests, it was also found that one of the sample and hold boards (specifically the card attached to channel 3 of the signal chains) had failed. It would appear to have been a static discharge that caused the card to fail during the installation process. Unfortunately a spare card was not available and the timing was such that the test was about to commence. The decision was made to exclude channel 3 from the data set, as channel 3 serves 7 of the signal chains, and then the receptical methodology that is connected to channel 6 (the double pcb mount) had its 3rd line excluded from the test as well to ensure that all of the methodologies are affected the same amount by the failure. There were therefore only 4 signal chains active at the start of the test for each of the methodologies.

During the initial tests it was also found that the 5th signal chain of the Directly soldered receptacles had failed. During the moving of the system, the rear tail solder connection had

cracked and caused the joint to become faulty. As this happened during normal use though, it was counted as the first failure of reliability during the test.

5.1.2 Temperature Chamber Data

There were 8 temperature measurements recorded during the automated test run at 8 second intervals. The 8 measured channels in the log file correspond to the following:

Channel	Position and Description
1 (Potted)	Probe measuring the ambient temperature outside the chamber
2 (Potted)	Probe in chamber – above wire wrap mounts
3	Failed
4 (Potted)	Probe in chamber – above solder cup mounts
5 (Potted)	Probe in chamber – between ribbon cables and double pcb mounts
6 (Potted)	Probe in chamber – between ribbon cables and soldered recepticle mounts
7 (Not Potted)	Probe measuring the ambient temperature outside the chamber
8 (Not Potted)	Probe in center of chamber to read inner chamber temperature.

Table 5.2: Temperature probe channels

When the system was installed into the lab and the pretests were run, the third channel temperature sensor failed the calibration tests and so although the data was read from it and recorded, it was not used.

During initial testing it was found that although the potted temperature probes measured temperature correctly, their thermal inertia was very high due to the low thermal conductivity of the thermal epoxy used on the front end. To offset this, two additional

unpotted probes were added. The first was added in the center of the chamber to get an unshielded internal temperature reading, and the second outside to get an unshielded ambient reading. A comparison is done between the potted and unpotted values to gauge the effect that the thermal epoxy had on the accuracy of the potted probes versus their unpotted counterparts.

The following graph shows the ambient temperature of the potted and unpotted probes over the full test cycle. Due to the fact that the test system was housed in a small room, the heating effect from the inside of the chamber can also be seen on the ambient temperature on the outside as it was heating up the room it was housed in (albeit not by nearly as much as the inside of the chamber). The ambient measurements were taken close enough to the test system that as nearly 5 ºC in the ambient temperature is measured as the chamber

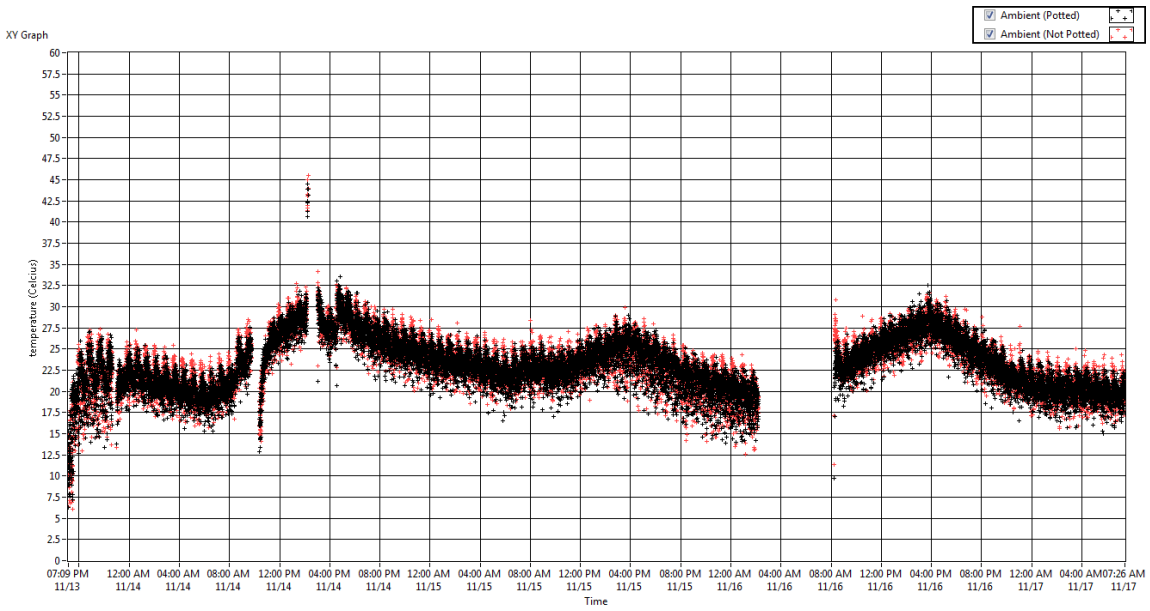


Illustration 5.1: Ambient temperature graph over full test

cycles its temperature up and down.

Evaluating the above graph – the following was noted.

1. At 11/14 10:24 the temperature at start up when the lab was cleared read very low on start up. The suspected reason is that while the test was running at other times the door was kept closed to try and maintain as stable a temperature as possible, and in openning it up it allowed all of the hot air to escape and cool down to a dead start position.

2. At 11/16 08:15 when the test system was switched on again it was very hot both inside and outside of the chamber since when the temperature chamber thread failed it failed with the heaters in the on position which allowed it to cook in the lab till the next morning when the sequence was restarted.

3. Although the potted and unpotted probes follow the same general curve, the thermal inertia of the potted probes cause it to read slightly differently. The method of comparison of the probe data be found in DifferenceAndPercentileOfDataBelow.vi. It was calculated by taking the absolute difference of each datapoint between the two probes and the checking what the difference is of the 95^{th} percentile. It was found that 95% of the data correspond to within 3.42ºC and 90% of the data to within 2.74ºC .

4. Although the room was air conditioned, it seems that the air conditioner was not powerful enough to overcome either the rapid changes of the temperature chamber, or completely overcome the effects of the heating and cooling in the day.

The inner chamber temperature was also measured with both an unpotted probe and a set of potted probes and the following graph shows the temperature inside the chamber of the unpotted probe:

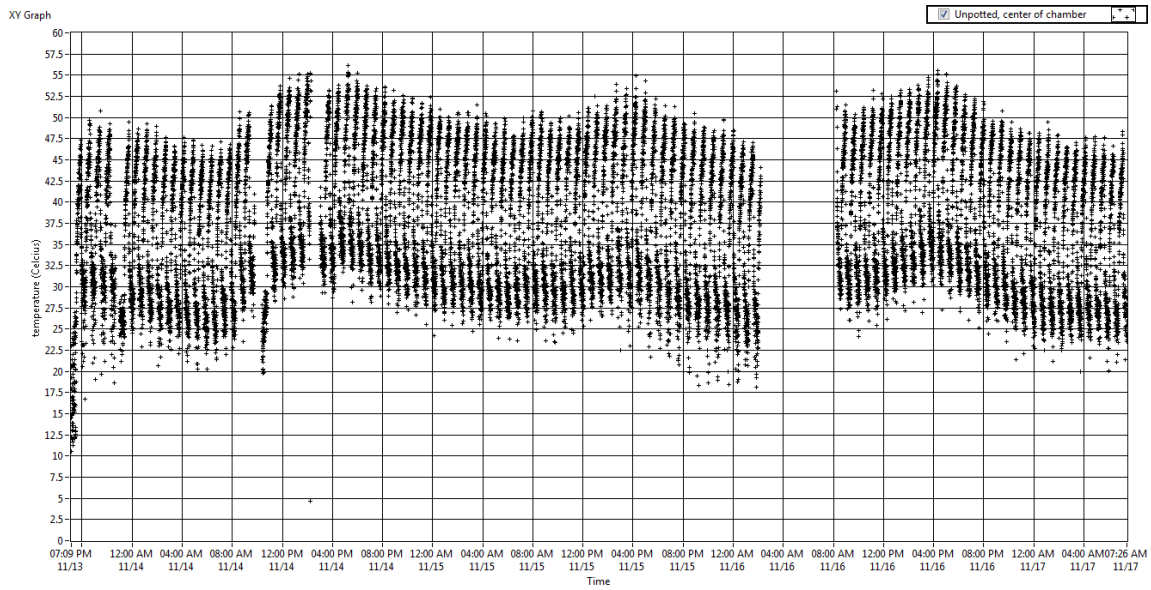


Illustration 5.2: Inner chamber temperature, unpotted probe only

Evaluating the above graph – the following was noted.

1. The general trend of the graph follows the same up and down motion as the ambient temperature as is expected, but the positive and negative swing obtained inside the chamber is very large. The temperature cycles between approximately ambient, and then 25ºC above ambient.

2. As with the ambient temperature, the inner chamber temperature also managed to cool down almost completely when the test was stopped and everything disconnected at 11/14 10:24, but it did recover within one heating cycle.

The following graph shows the inner temperature of the chamber as measured on the four potted probes:

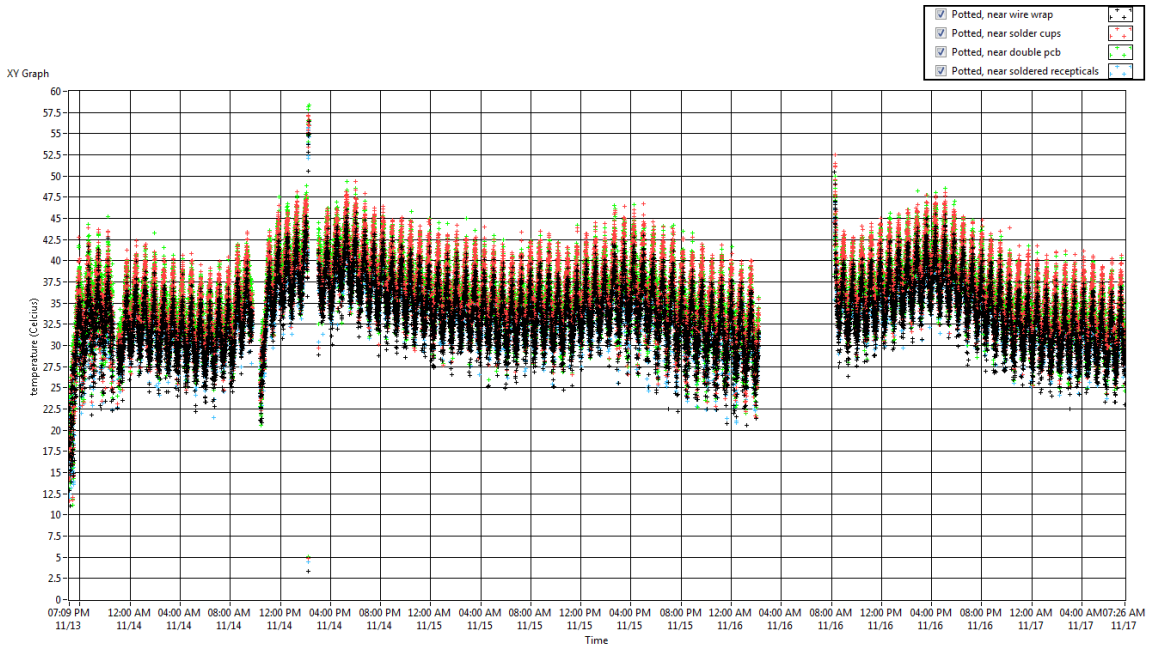


Illustration 5.3: Inner chamber temperature, potted probes

Evaluating the above graph – the following was noted.

1. The potted chamber probes follow the same general curve as the unpotted inner chamber probe, but due to the thermal inertia, they show a much lower swing of only about 17ºC above ambient as opposed to the 25ºC that was obtained from the unpotted probe.

2. Using the same technique of comparison as used on the ambient potted and unpotted probe, the following results were obtained:

Channel	**90th Percentile**	**95th Percentile**
Near wire wrap	11.34ºC	12.12ºC
Near solder cups	8.6ºC	9.29ºC
Near double pcb	8.9ºC	9.77ºC
Near soldered recepticals	11.53ºC	12.12ºC

Table 5.3: Potted versus unpotted temperature error

3. When running the same comparison between the potted probes on the "left" (wire wrap and soldered receptacles) and the "right" (solder cups and double PCB) it is found that the "left" probes differ by 2.93ºC on the 90th percentile and 3.81ºC on the 95th percentile, while the "right" probes differ by 2.24ºC on the 90th percentile and 2.73ºC on the 95th percentile. Left and right in this instance is merely how they are orientated in the box when it is viewed from above. The close correlation between these probes is probably due to thermal currents caused by the layout of the probes, receptacle mounts and cabling inside the chamber.

The impact of the thermal inertia can most easily be seen by comparing one of the potted probes inside the chamber over only a few of the thermal cycles.

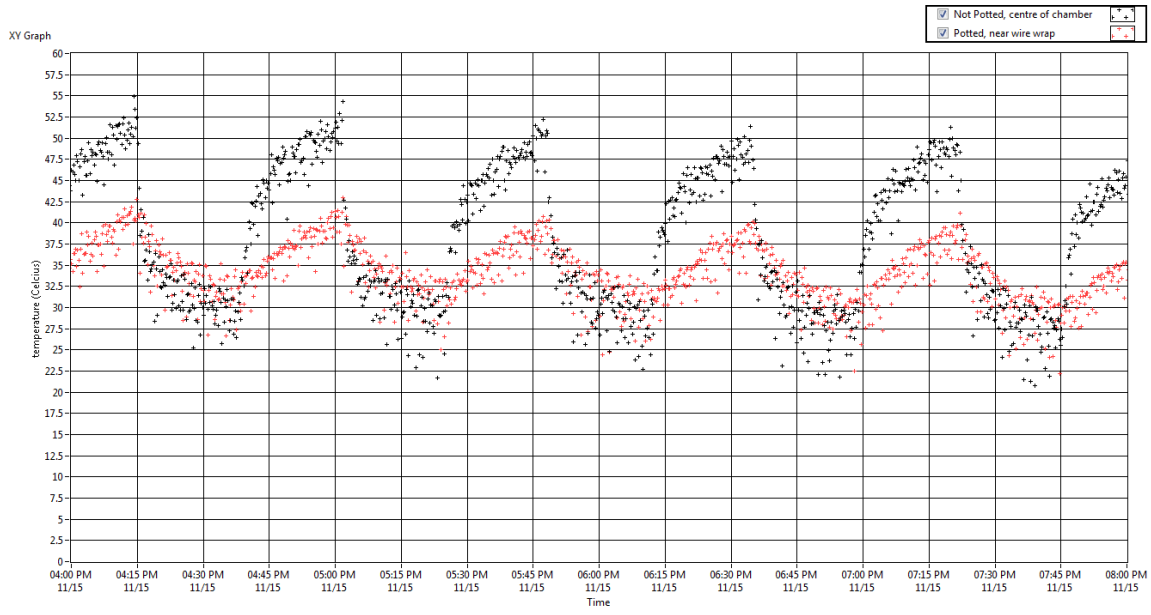


Illustration 5.4: Potted versus unpotted plots showing thermal inertia

Evaluating the graph below, the following can be seen:

1. The two probes follow the same up and down profile as the chamber heats up and cools down but the unpotted probe registers the rapid changes in temperature that occur at the start of the heating and cooling phases much faster than the potted probe.

2. The thermal inertia affects the probe in both the heating and cooling phases, but it does affect it more during the heating cycle where ΔT is at its largest. This also explains why there would be much closer correlation between the potted and unpotted ambient probes since the ΔT is much smaller there.

Using formula 2.5 (Arrhenius equation) found in section 2.4, it can be seen that the acceleration factor for the test using a normal temperature of 20°C and the peak temperature of 55°C that the acceleration factor on failure rate would have been at peak 38.95. So over the test time of 3 days, 03 hours and 35 minutes, this would equate to 122 days, 15 hours, 58 minutes.

5.1.3 Signal chain conductivity

The mapping of signal chains to receptacle methodologies was as follows:

Signal chain	**Double pcb**	**Direct solder**	**Wire wrap**	**Pcb on top**	**Double ended**	**Solder cups**	**Soldered recepticals**	**Crimped**
1	Ch6-1	Ch1-1	Ch1-2	Ch1-3	Ch1-4	Ch1-5	Ch1-6	Ch1-7
2	Ch6-2	Ch2-1	Ch2-2	Ch2-3	Ch2-4	Ch2-5	Ch2-6	Ch2-7
3	Ch6-3*	Ch3-1*	Ch3-2*	Ch3-3*	Ch3-4*	Ch3-5*	Ch3-6*	Ch3-7*
4	Ch6-3	Ch4-1	Ch4-2	Ch4-3	Ch4-4	Ch4-5	Ch4-6	Ch4-7
5	Ch6-4	Ch5-1	Ch5-2	Ch5-3	Ch5-4	Ch5-5	Ch5-6	Ch5-7

Table 5.4: Signal chain mapping

*not included in test set due to failure of Sample and Hold card for channel 3.

During the test the a square wave is applied to each of the channels (1-6) in turn which has a 50% duty cycle, and a 2 ms period and a 5V peak value. During the Low test, no voltage is present, and to pass the signal chain must read 0V to ensure the signal chain starts with no charge on it (this would only be of relevance for a specific failure mode though where a dry joint causes sufficient charge for some small charge to be detected by the integration of the sample and hold circuitry).

During a high test, the above signal is applied, and a sample and hold measurement is performed on each of the signal chains connected to the channel. The sample window is allowed to stay open for 150ms, and a single voltage measurement is taken corresponding to the integral of the waveform applied over the sample window. For this test nominally the sample window was long enough for the square wave to charge up the sample capacitor to 2.3V. This means that if a failure should occur, and instead of a square wave a 5Vdc signal is applied, the sample capacitor could only charge up to 4.6V which would leave a 0.4V safety margin on the inputs to the DAQ analog inputs (which are only rated between 0-5V). The high test will pass for a signal chain however if it measures above 2.2V and below 2.6V. See Annexure A for a description of how the output voltage will change based on deformation of the incoming square wave due to changes in the reactance of the signal chain.

The graph on the following page indicates the voltages measured on the signal chains connected to channel 1:

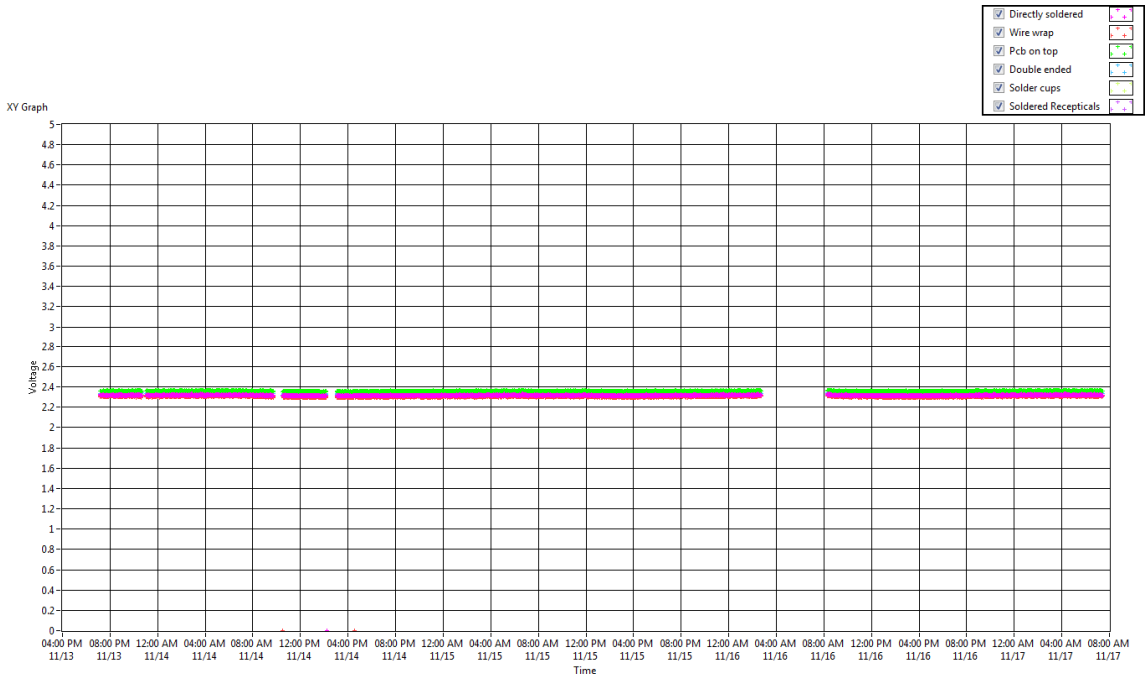


Illustration 5.5: Voltages measured on channel 1 over the course of the test

As expected on all of the lines that passed through the test, the resistance didn't vary at all. From the above data, the standard deviation from the mean value for each of the signal chains was less than 50mV across the entire test. The only signal chain not included in the above graph is that of the Crimped channel which failed during the early stages of the test. The failed crimped channel's plot is below.

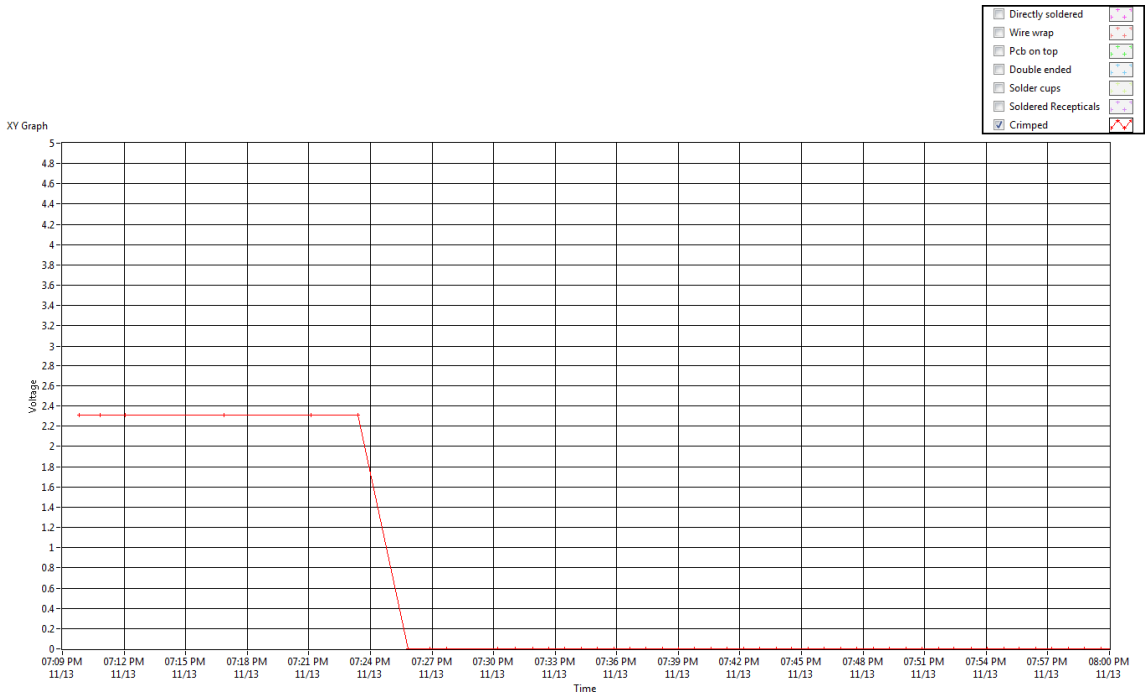


Illustration 5.6: Crimped channel failure on test sequence 7 (channel 1)

The channel 1 crimped receptical only managed to get through 6 test sequences and failed on the 7th. It therefor managed to survive (at least) 48 actuations before failing.

The graph on the following page indicates the voltages measured on the signal chains connected to channel 2.

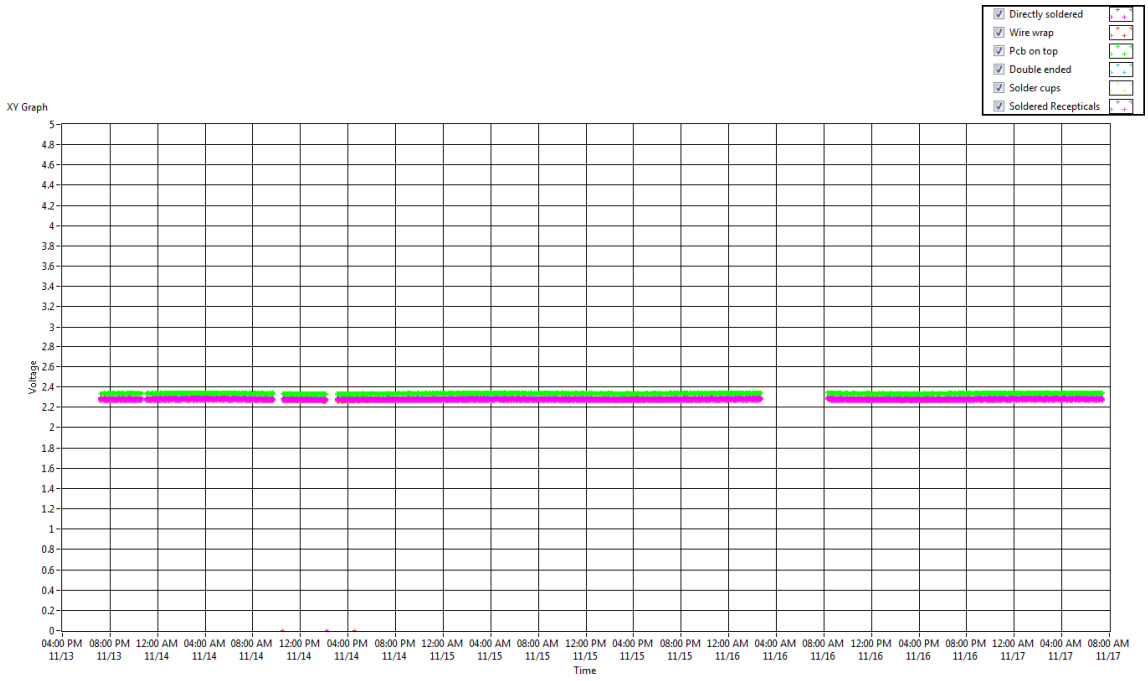


Illustration 5.7: Voltages measured on channel 2 over the course of the test

As expected on all of the lines that passed through the test, the resistance didn't vary at all. From the above data, the standard deviation from the mean value for each of the signal chains was less than 50mV across the entire test. The only signal chain not included in the above graph is that of the Crimped channel which failed during the early stages of the test. The failed crimped channel's plot is on the following page.

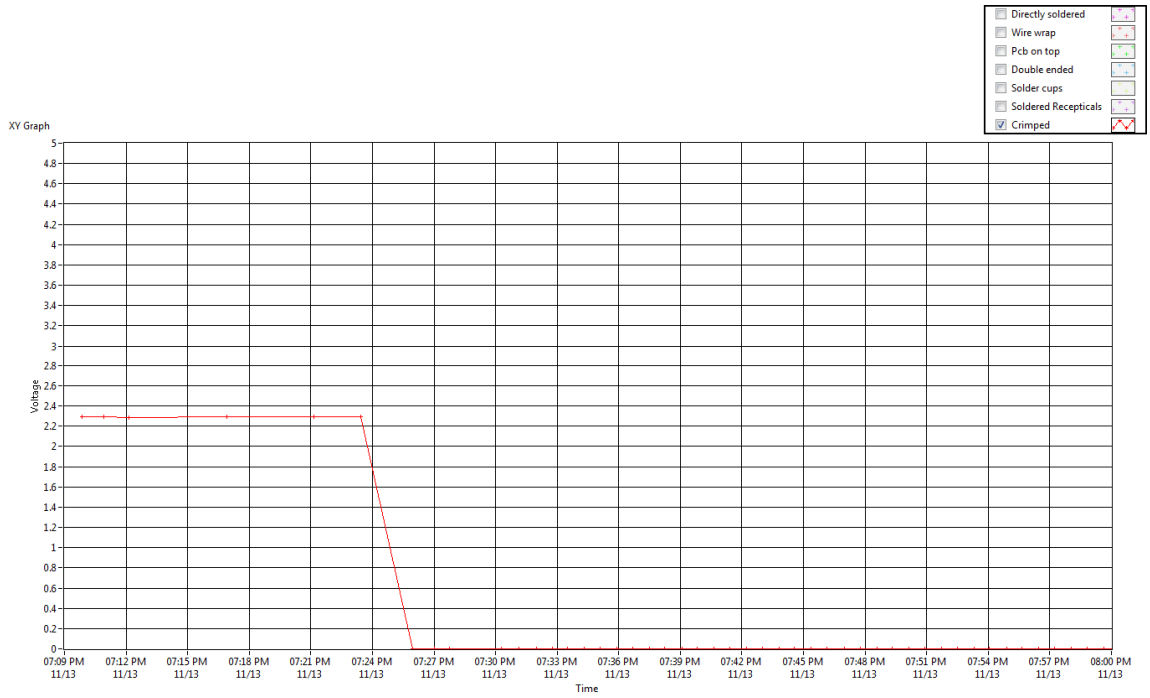


Illustration 5.8: Crimped channel failure on test sequence 7 (channel 2)

As with the channel 1 failure, the second channel crimped receptical also failed on the 7^{h} sequence, and therefor survived at least 48 actuations.

Channel 3 was not included in the test, so no graphs are available for it.

The below graph indicates the voltages measured on the signal chains connected to channel 4:

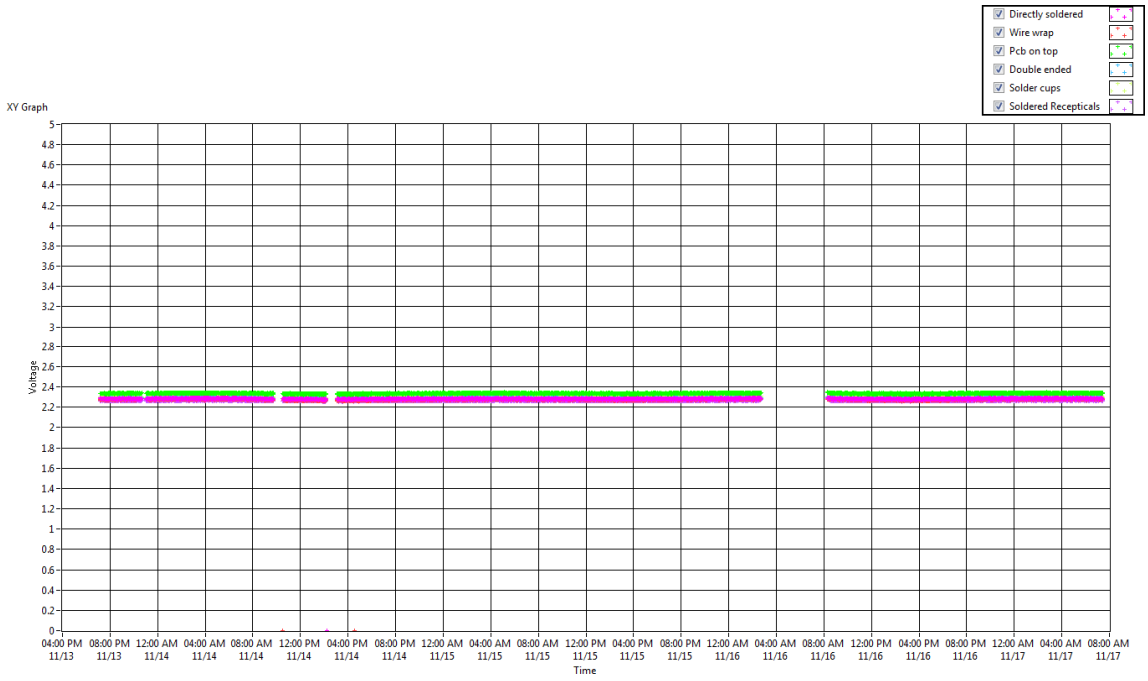


Illustration 5.9: Voltages measured on channel 4 over the course of the test

As expected on all of the lines that passed through the test, the resistance didn't vary at all. From the above data, the standard deviation from the mean value for each of the signal chains was less than 50mV across the entire test. The only signal chain not included in the above graph is that of the Crimped channel which failed during the early stages of the test. The failed crimped channel's plot is on the following page.

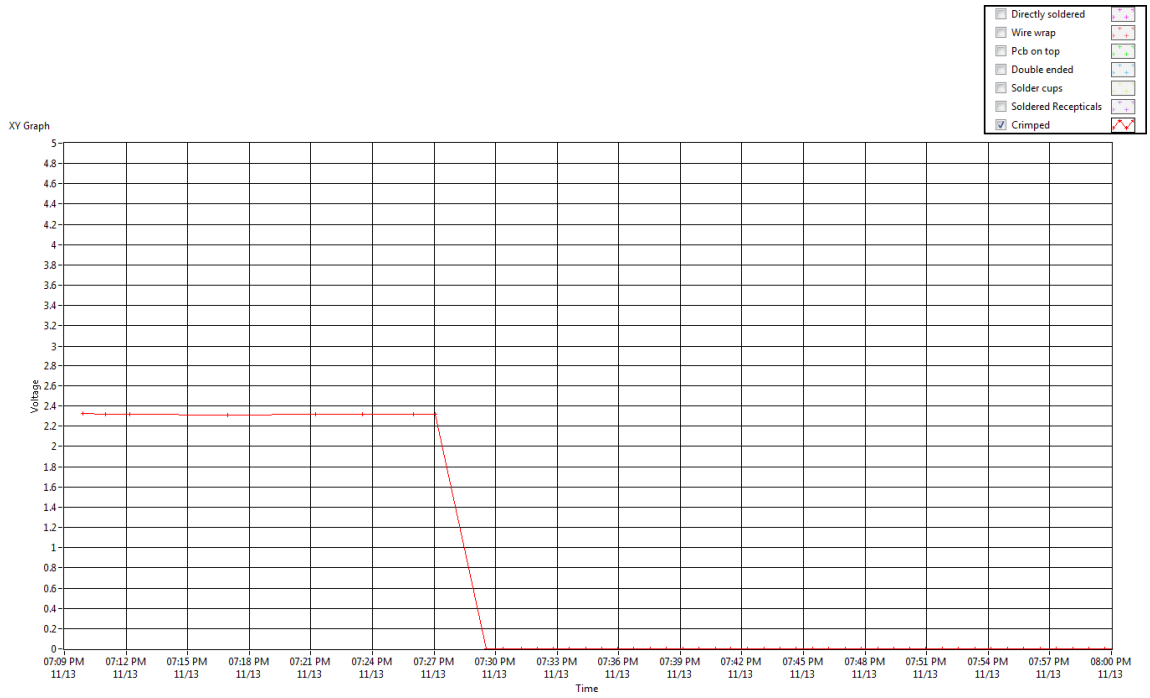


Illustration 5.10: Crimped channel failure on test sequence 10 (channel 4)

The Channel 4 crimped connection failed on the 10th sequence, and so survived 72 actuations before failing.

The graph on the following page indicates the voltages measured on the signal chains connected to channel 5.

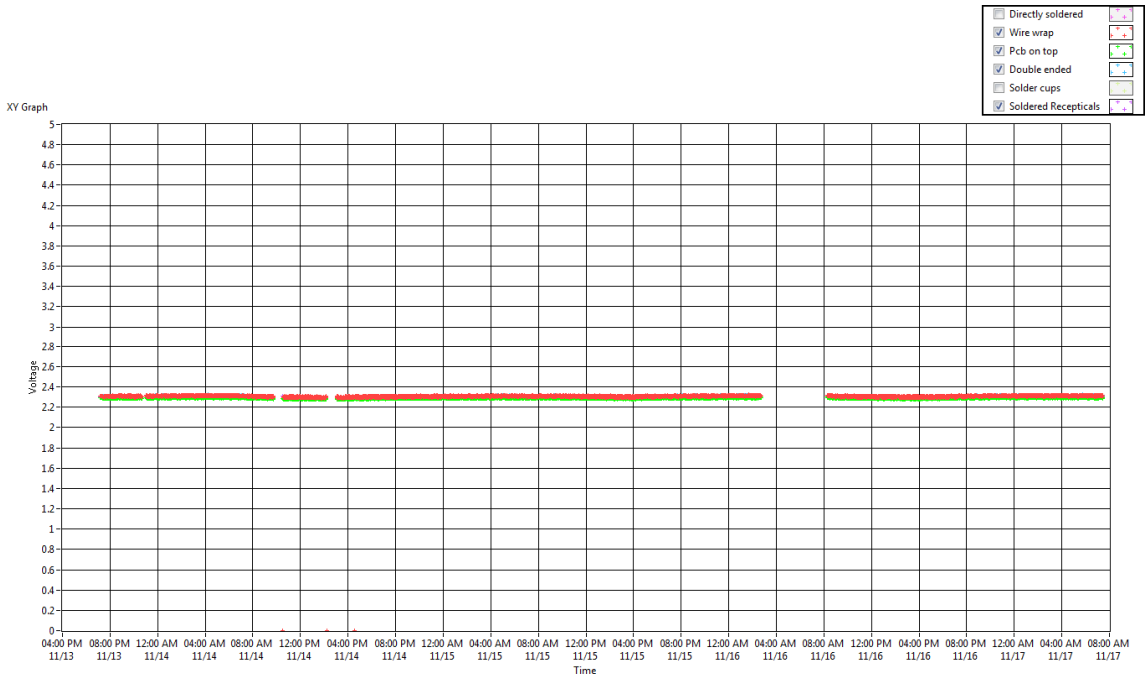


Illustration 5.11: Voltages measured on channel 5 over the course of the test

As expected on all of the lines that passed through the test, the resistance didn't vary at all.

From the above data, the standard deviation from the mean value for each of the signal chains was less than 50mV across the entire test. On Channel 5, a Crimped, Directly soldered as well as a solder cup signal chain failed. The Directly soldered receptacle survived some of the initial tests, but failed when the test system was moved to the lab for the long term stress test – it is therefor assumed to be the first one to fail during the first sequence, the same is true of the solder cup receptacle. The crimp receptacle on channel 5 passed the initial tests both in and out of the lab, but on the first official sequence it failed and is therefor allocated zero passed sequences.

The graph on the following page indicates the voltages measured on the signal chains connected to channel 6:

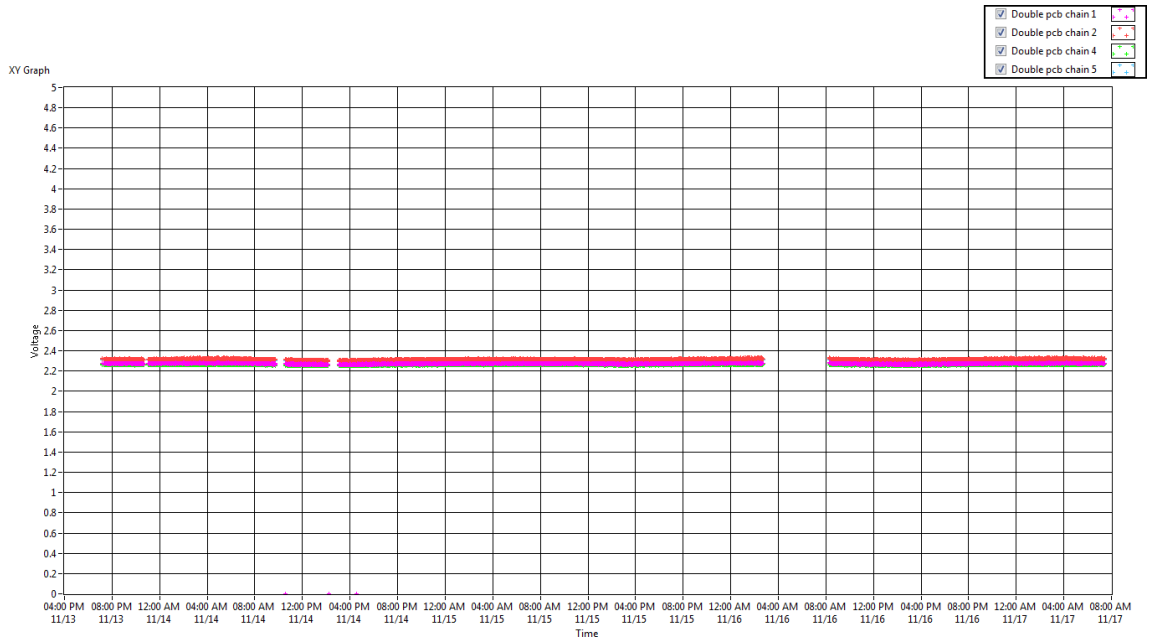


Illustration 5.12: Voltages measured on channel 6 over the course of the test

None of the signal chains connected to channel 6 failed the test.

5.1.4 Mechanical wear on UUT/Probe interface

Although the test probes themselves and the simulated UUT boards were not subjected to the same thermal stress as the receptacle mountings, the results obtained from them were the most surprising. During the initial design, the UUT was identified as the part that would wear first due to industry consultation that indicated that the sharp nature of the test probes tends to damage the test points. To minimise this, the UUTs were designed to have plated through hole test points to allow the test probes to hit at three points (due to the pyramidal head) as opposed to a single point as with a test pad. The test probes were also assumed to need at least some replacing as at the time of specifying and purchasing them the recommended service interval was at 30 000 actuations (and the test ran for 44 632 actuations).

Over the entire test not a single failure could be attributed to a worn test point or a failed/worn test probe. There was significant wear on the test points in the shape of triangular dents where the probes kept impacting them (see image below):

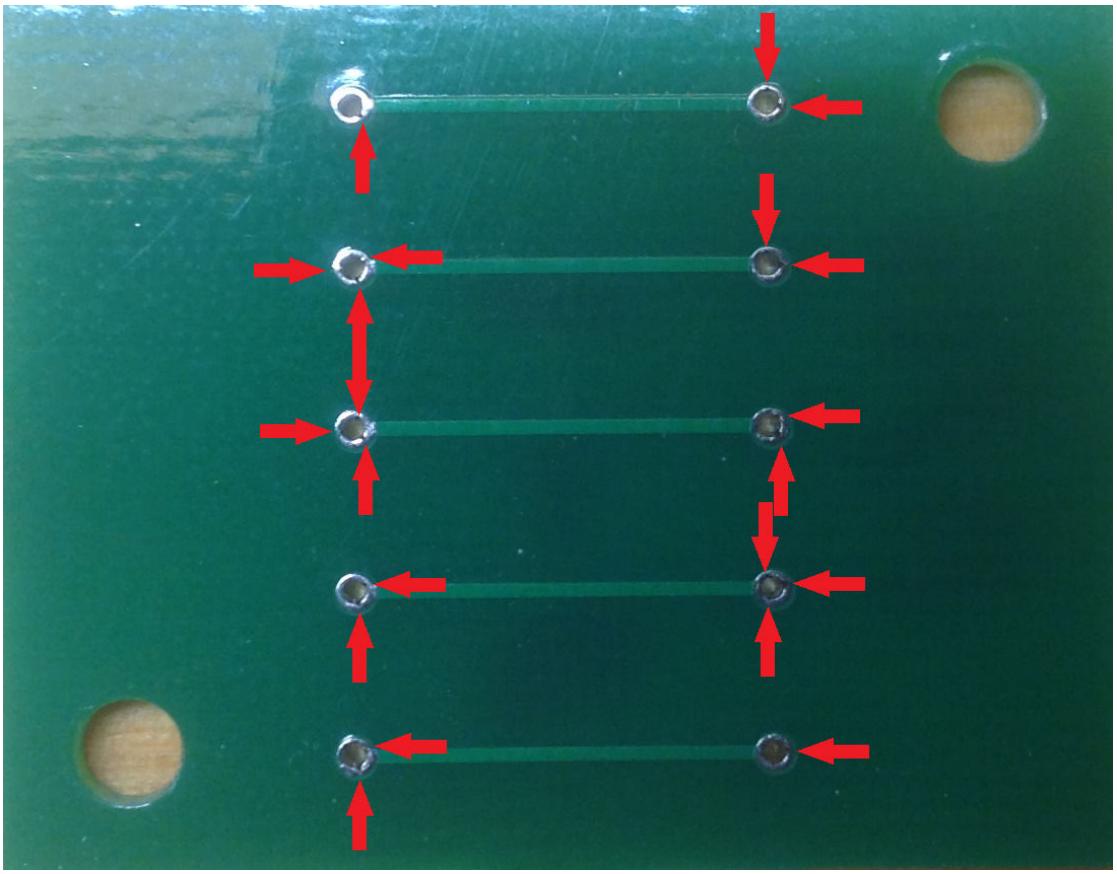


Illustration 5.13: UUT wear indicating triangular impacts from probes

The same is true of the test probes. Although significant wear can be seen on edges of the triangular tips of the probes, none of them seemed to have any impact on the contact resistance of the UUT/Probe interface. It should however be noted that there were no chemicals present (for example, flux and pcb cleaner) during the test which could under normal circumstances have an impact on the reliability of the UUT/Probe interface.

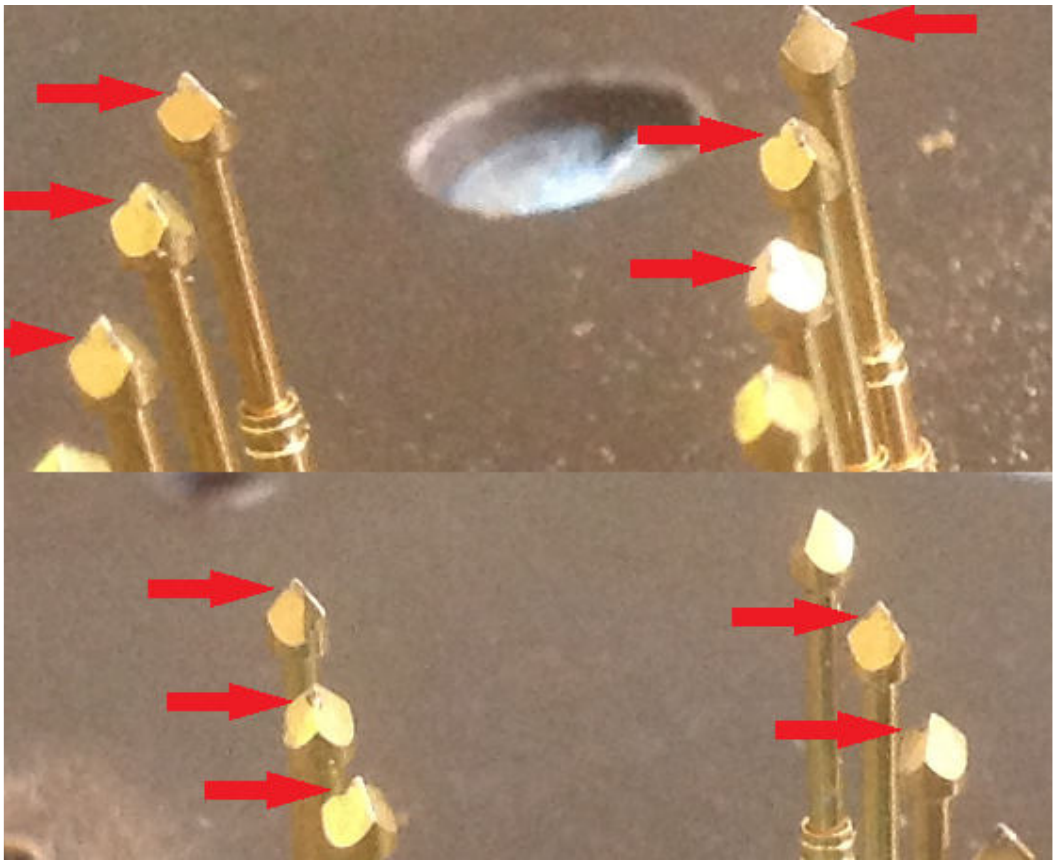


Illustration 5.14: Dents in test probe tips

5.1.5 Heating element failure

When the test system was opened up for inspection after the test, it was found that one of the solder connections on one of the 15 heating elements had failed. This was most likely due to the failure that occurred 16/11 02:42 which caused the system to fail in the "heating" state and allowing the heating elements to run non stop for 6 hours. It was surprising that more of the resistors/solder connections didn't fail, but since it was only 1 of the 15 chains, it had minimal impact on the chamber's ability to heat itself up.

Illustration 5.15: Heating element failure

5.2 RELATIVE RELIABILITY CONCLUSIONS

5.2.1 Relative reliability ranking

From the data above, the following failure matrix was obtained:

Methodology	Failures	Ranking	Notes
Direct solder	1	5	Failed the pretest
Wire wrap	0	8	none
Pcb on top	0	8	none
Double ended	0	8	none
Solder cups	1	6	Failed on first sequence, but passed pretest
Soldered recepticals	0	8	none
Crimped	4	1	All the recepticals failed in the test
Double pcb	0	8	none

Table 5.5: Relative reliability rankings

5.2.2 The Final Receptacle Methodology Selection model Test

Using the qualitative data and sample pricing collected in chapter 4, and the relative reliability data collected after the completion of the testing cycle, the model is finalised. The final model is implemented in Excel, and can be found in "model.xls".

To use the model, the inputs (marked in green) are set to reflect the users requirements, and the suitability ranking (marked in yellow) indicates the suitability of each methodology to the specific requirements. The suitability ranking is a ranking of the suitability score of each of the methodologies based on the requirements entered, and it is calculated as follows:

Assign out of 100% a value of importance to each of the following parameters: Maintainability ($M_{\%}$), Ease of implemetation ($S_{\%}$), Ease of implementing signal conditioning ($I_{\%}$), Ability to change and update system ($C_{\%}$), Price ($P_{\%}$) .

Add the Number of probes(N) to be used in the system.

Suitability: $M_{\%}(M_{methodology}) + S_{\%}(S_{Methdology}) + I_{\%}(I_{Methodology}) + C_{\%}(C_{Methodology}) + P_{\%}|(Pf_{Methodology})$

where Pf = Price function for the methdology

For the price function, calculate the cost of adding the receptacles using the price of implementation collected, and then if the methodology requires one of the special tools and the tool is not available, then add that to the final value of the price function.

To test the suitability of the model, three sample case studies are evaluated.

5.2.2.1 Case Study A: A Small Startup company

The company in question is still in its infancy. Although it owns the rights and designs to their own product, they are very price conscious as they are not yet a trading business so their capital is largely sourced from investment and there is no solid income stream yet.

Pricing is the main driving factor for decisions made during the design of their test system, but they have time and in house design capacity available, so if it takes a bit longer to implement the system that's not a problem. They would like the system to be fairly reliable, but the price is definitely the most important. Using the above information, the following choices are made:

Ability to maintain the system: 5%

Initially, they will have low production numbers, so the system is not likely to break often. If it does however go down, if it takes longer to fix and has some delay in availabilty, it would not be a huge problem to work around it.

Ease of implementing signal conditioning: 5%

The company has access to in house engineers that can work around some of the constraints imposed by some of the mounting methodologies – the additional engineering time is not as expensive as the capital outlay of a more expensive mounting methodology, as the engineers have a lot of idle time at the moment.

Ability to change the system: 10%

Although the same engineering resources will still be available to change the system if required by future versions, its slightly more important to take into account due to the fact that the product being developed is still in its early stages of production, so the design may still change.

Pricing: 60%

The cost of the system has been flagged as the highest concern.

Speed of implementation: 5%

Again, the availability of highly skilled resources that are at points idle means that they can be repurposed into implementation and testing of the test fixtures.

Relative Reliability: 15%

Although it is not too much of a problem if the system breaks that it would take a long time to fix, it is more important for it to be reliable in the long run as there is likely to only be one of the fixture initially (Due to the price concern and the low production numbers), so the methodology chosen should take into account the reliability as a more important factor.

The model predicts that the company should look at wire wrap receptacles (due largely to the pricing), but stay away from double ended receptacles. The full model results are available in the table overleaf.

Table 6 : Case Study A

Parameters	Percentage Importance
Ability to maintain system	5
Ease of implementing signal conditioning	5
Ability to change system	10
Pricing	60
Speed of implementation	5
Relative Reliability	15
Number of test points	100

Tooling – do you own it:	0 for yes, 1 for no
wire wrap tool	1
soldering iron	1
crimping tool	0

Methodology	Suitability Score	Suitability Rank
wire wrap	6.3	8
solder cup	5.15	5
crimp	5.45	6
Backplane, soldered receptcles	3.75	2
Backplane, double ended receptcles	3.55	1
Backplane, receptcles soldered in place	3.8	3
PCB on probe platen	5.75	7
double PCB mount	4.2	4

Methodology	Ability to maintain system
wire wrap	4
solder cup	2
crimp	3
backplane with soldered receptcles	6
backplane with double ended receptcles	7
backplane with receptcles soldered in place	1
PCB on probe platen	8
double PCB mount	5

Methodology	Ease of implementing signal conditioning
wire wrap	1
solder cup	1
crimp	1
backplane with soldered receptcles	6
backplane with double ended receptcles	8
backplane with receptcles soldered in place	7
PCB on probe platen	4
double PCB mount	5

Methodology	Ability to change system
wire wrap	3
solder cup	4
crimp	1
backplane with soldered receptcles	7
backplane with double ended receptcles	8
backplane with receptcles soldered in place	8
PCB on probe platen	6
double PCB mount	6

Methodology	Pricing
wire wrap	7
solder cup	6
crimp	2
backplane with soldered receptcles	1
backplane with double ended receptcles	4
backplane with receptcles soldered in place	3
PCB on probe platen	5
double PCB mount	5

Methodology	Speed of implementation
wire wrap	2
solder cup	4
crimp	4
backplane with soldered receptcles	1
backplane with double ended receptcles	4
backplane with receptcles soldered in place	3
PCB on probe platen	3
double PCB mount	2

Methodology	Relative Reliability
wire wrap	6
solder cup	6
crimp	1
backplane with soldered receptcles	8
backplane with double ended receptcles	8
backplane with receptcles soldered in place	5
PCB on probe platen	8
double PCB mount	8

Methodology	Price per point	Notes
wire wrap	R 8.60	From TFT
solder cup	R 8.60	From TFT
crimp	R 8.50	From TFT
backplane with soldered receptcles	R 17.18	From TFT + Digikey(SAM1213-01-N = $0.78)
backplane with double ended receptcles	R 34.56	
backplane with receptcles soldered in place	R 8.60	
PCB on probe platen	R 8.50	
double PCB mount	R 8.50	

Methodology	Price factor
wire wrap	R 1,240.00
solder cup	R 1,310.00
crimp	R 850.00
backplane with soldered receptcles	R 4,327.79
backplane with double ended receptcles	R 5,615.79
backplane with receptcles soldered in place	R 3,460.79
PCB on probe platen	R 3,459.79
double PCB mount	R 3,913.39

Methodology	Price factor ranking
wire wrap	7
solder cup	6
crimp	8
backplane with soldered receptcles	2
backplane with double ended receptcles	1
backplane with receptcles soldered in place	4
PCB on probe platen	5
double PCB mount	3

note: for all pricing assume $1 = R11

Tooling	Price	Notes
wire wrap tool	R 389.00	from communica
soldering iron	R 450.00	from communica
crimping tool	R 620.00	from communica

has Pcb tooling and 1x board added
has Pcb tooling and 1x board added
has Pcb tooling and 1x board added
has Pcb tooling and 1x board added
has Pcb tooling and 2x board added

PCB Costing	Price	Notes
PCB	R 1,059.79	Based on 390mmx265mm panel with multiple boards on it. double layer 35um copper, silkscreen and solder mask.
PCB x2 (at R756.75 each)	R 1,513.50	Based on 390mmx265mm panel with multiple boards on it. double layer 35um copper, silkscreen and solder mask.
loding	R 1,100.00	

Note: Two boards have a lower per unit cost (and also lower as the units become much more)

5.2.2.2 Case Study B: A military contractor

The military contractor is not at all cost conscious as reliability is the core concern of the test system as well as the final product. The level of complexity of their hardware, and the amount of signal conditioning required however lead them to indicate that the signal conditioning implementation, reliability and ability to change are by far the most important requirements to them. Taking into account the requirements, the following parameters are chosen:

Ability to maintain the system: 5%

The product being manufactured is of very high value, but very low production numbers are expected. The system is not likely to break, and if it does, the down time is not a problem.

Ease of implementing signal conditioning: 25%

The system will be required to handle a lot of signal conditioning hardware, due to the complex nature of the product being tested. A methodology that simplifies this is going to be more suitable.

Ability to change the system: 30%

The nature of complex designs is that they may go through more design iterations than simpler systems. It is forseeable that there would need to be a change in the signal conditioning later down the line as the product is changed.

Pricing: 5%

The cost of the system is negligible compared to the cost of the products it is testing.

Speed of implementation: 5%

Military projects run over long periods of time as accuracy and reliability of a much greater concern than time to market. The speed of implementation is not considered as a crucial parameter.

Relative Reliability: 30%

Reliability is stated as very important as the product being tested will need to conform to very high specifications and faults in the test equipment reading incorrectly could lead to very expensive rework on the tested product.

The model predicts that the company should look at double ended receptacles, but stay away from crimped connections. The full model results are available in the table overleaf.

Table 7: Case Study B

Parameters	Percentage Importance
Ability to maintain system	5
Ease of implementing signal conditioning	25
Ability to change system	30
Pricing	5
Speed of implementation	5
Relative Reliability	30
Number of test points	200

Tooling – do you own a:	0 for yes, 1 for no
wire wrap tool	1
Soldering iron	1
crimping tool	1

Methodology	Suitability Score	Suitability Rank
wire wrap	4.9	4
solder cup	3.8	2
crimp	1.5	1
Backplane, soldered receptacles	6.45	6
Backplane, double ended receptacles	7.4	8
Backplane, receptacles soldered in place	3.95	3
PCB on probe platen	6.6	7
double PCB mount	5.95	5

Methodology	Ability to maintain system	Ease of implementing signal conditioning	Ability to change system	Pricing	Speed of implementation	Relative Reliability
wire wrap	4	1	5	8	3	8
solder cup	2	1	4	7	2	6
crimp	3	1	1	6	4	1
backplane with soldered receptacles	6	6	7	2	1	8
backplane with double ended receptacles	7	8	8	1	4	5
backplane with receptacles soldered in place	1	7	1	4	3	8
PCB on probe platen	8	4	8	3	3	8
double PCB mount	5	5	6	5	2	8

Methodology	Price per point	Notes
wire wrap	R 8.60	From TFT
solder cup	R 8.60	From TFT
crimp	R 8.50	From TFT
backplane with soldered receptacles	R 17.18	From TFT + Digikey(SAM1213-01-N = $0.78)
backplane with double ended receptacles	R 8.50	
backplane with receptacles soldered in place	R 8.50	
PCB on probe platen	R 8.50	
double PCB mount	R 8.50	

Methodology	Price factor	Price factor ranking
wire wrap	R 2,109.00	8
solder cup	R 2,170.00	7
crimp	R 2,320.00	6
backplane with soldered receptacles	R 6,045.79	2
backplane with double ended receptacles	R 4,923.79	4
backplane with receptacles soldered in place	R 4,923.79	4
PCB on probe platen	R 4,309.79	5
double PCB mount	R 4,763.50	3

Tooling	Price	Notes
wire wrap tool	R 389.00	from communica
Soldering iron	R 450.00	from communica
crimping tool	R 620.00	from communica

note, for all pricing assume $1 = R11

PCB Costing	Price	Notes
PCB	R 1,099.78	based on 390mmx953mm panel with multiple boards on it. double layer 35um copper, silkscreen and solder mask.
PCB x2 (at R756.75 each)	R 1,513.50	based on 390mmx953mm panel with multiple boards on it. double layer 35um copper, silkscreen and solder mask.
tooling	R 1,100.00	

Note: Two boards have a lower per unit cost (and also lower as the units become much more)

5.2.2.3 Case Study C: A high volume production facility

The Factory manager requires a system that is reliable and easy to maintain and needs to be able to be updated quickly as new revisions of a product are added to the line. He also has access to a vast array of tools to ensure things run smoothly in the factory. He identifies as the most important factors the ability to maintain the system, the ability to update the system, speed of implementation and reliability as the most important factors, but considers them very similar in terms of importance. Taking into account the requirements, the following parameters are chosen:

Ability to maintain the system: 30%

Factory downtime is detrimental to the high volume production facility. If a test fixture breaks, it must be swopped out and brought back into service as soon as possible.

Ease of implementing signal conditioning: 5%

A system can take a long time to reach the production line, but once it is on the line it must perform well. It is acceptable to the factory manager that the signal conditioning can take some additional time to implement, as long as the final product conforms to the other requirements.

Ability to change the system: 30%

Once products reach the manufacturing line, and are produced in large quantities as they are successful, the likelyhood increases that the product being manufactured will be updated rather than replaced with a different product alltogether. Due to this, it is desirable that the test fixture be easy to update.

Pricing: 5%

The high volume of product manufactured means that the cost of the fixture can be easily amortised over a larger number of products. As such, the cost impact of a more expensive methodology is easier to absorb.

Speed of implementation: 25%

Due to the large production numbers, it may become necessary for a second or third test station to be put down to increase through put. In this case, it is important for the system to be easy to replicate quickly.

Relative Reliability: 25%

Factory down time is flagged as a very large concern, so the reliability is marked as important.

The model predicts that the company should look at receptacles mounted on top of the probe platen, or double ended receptacles. They should stay away from crimped receptacles.

The full model results are available in the table overleaf.

Table 8: Case Study C

Parameters	Percentage Importance
Ability to maintain system	30
Ease of implementing signal conditioning	5
Ability to change system	30
Pricing	5
Speed of implementation	5
Relative Reliability	25
Number of test points	50

Tooling – do you own a:	0 for yes, 1 for no
wire wrap tool	0
Soldering iron	0
crimping tool	0

Methodology	Suitability Score	Suitability Rank
wire wrap	5.85	4
solder cup	4.2	3
crimp	2.9	1
Backplane, soldered receptacles	6.8	6
Backplane, double ended receptacles	7.95	7
Backplane, receptacles soldered in place	3.15	2
PCB on probe platen	8	8
double PCB mount	6.15	5

Methodology	Ability to maintain system
wire wrap	2
solder cup	3
crimp	6
backplane with soldered receptacles	7
backplane with double ended receptacles	8
backplane with receptacles soldered in place	1
PCB on probe platen	8
double PCB mount	5

Methodology	Ease of implementing signal conditioning
wire wrap	1
solder cup	1
crimp	1
backplane with soldered receptacles	6
backplane with double ended receptacles	8
backplane with receptacles soldered in place	7
PCB on probe platen	4
double PCB mount	5

Methodology	Ability to change system
wire wrap	5
solder cup	4
crimp	1
backplane with soldered receptacles	7
backplane with double ended receptacles	8
backplane with receptacles soldered in place	1
PCB on probe platen	8
double PCB mount	6

Methodology	Pricing
wire wrap	8
solder cup	7
crimp	6
backplane with soldered receptacles	2
backplane with double ended receptacles	1
backplane with receptacles soldered in place	4
PCB on probe platen	3
double PCB mount	5

Methodology	Speed of implementation
wire wrap	3
solder cup	2
crimp	4
backplane with soldered receptacles	1
backplane with double ended receptacles	4
backplane with receptacles soldered in place	3
PCB on probe platen	2
double PCB mount	1

Methodology	Relative Reliability
wire wrap	8
solder cup	8
crimp	1
backplane with soldered receptacles	8
backplane with double ended receptacles	8
backplane with receptacles soldered in place	5
PCB on probe platen	8
double PCB mount	8

Methodology	Price per point	Notes
wire wrap	R 8.60	From TFT
solder cup	R 8.60	From TFT
crimp	R 8.50	From TFT
backplane with soldered receptacles	R 17.18	From TFT + Digikey(SAM1213-01-N = $0.78)
backplane with double ended receptacles	R 34.56	
backplane with receptacles soldered in place	R 8.60	
PCB on probe platen	R 8.50	
double PCB mount	R 8.50	

Methodology	Price factor
wire wrap	R 430.00
solder cup	R 420.00
crimp	R 425.00
backplane with soldered receptacles	R 3,018.79
backplane with double ended receptacles	R 3,887.79
backplane with receptacles soldered in place	R 2,589.79
PCB on probe platen	R 2,584.79
double PCB mount	R 3,038.50

Methodology	Price factor ranking
wire wrap	7
solder cup	6
crimp	8
backplane with soldered receptacles	3
backplane with double ended receptacles	1
backplane with receptacles soldered in place	4
PCB on probe platen	5
double PCB mount	2

Tooling	Price	Notes
wire wrap tool	R 399.00	from communica
Soldering iron	R 450.00	from communica
crimping tool	R 620.00	from communica

note: for all pricing assume $1 = R11

Methodology	Suitability Score	Suitability Rank
has Pcb tooling and 1x board added	5.85	4
has Pcb tooling and 1x board added	4.2	3
has Pcb tooling and 1x board added	2.9	1
has Pcb tooling and 1x board added	6.8	6
has Pcb tooling and 1x board added	7.95	7
has Pcb tooling and 1x board added	3.15	2
has Pcb tooling and 1x board added	8	8
has Pcb tooling and 2x board added	6.15	5

PCB Costing	Price	Notes
PCB	R 1,059.79	Based on 390mmx353mm panel with multiple boards on it, double layer 35um copper, silkscreen and solder mask.
PCB x2 (at R756.75 each)	R 1,513.50	Based on 390mmx353mm panel with multiple boards on it, double layer 35um copper, silkscreen and solder mask.
tooling	R 1,100.00	

Note: Two boards have a lower per unit cost (and also lower as the units become much more)

5.3 **SUMMARY**

In this chapter the data collected from the relative reliability tester was analysed and conclusions drawn on the relative reliability of the different mounting methodologies based on that. There was also an exploration of the impact on the thermal inertia of the temperature probes due to the thermal epoxy used on the front end.

After the data was analysed, a final version of the receptacle mounting methodology selection model was presented and three different example cases were presented to test the model.

Chapter 6 will look at the final conclusions and evaluate the study based on the hypothesis posed in chapter 1. It will also look at some of the problems encountered in this project and also make some recommendations for further study.

CHAPTER 6

6 SUMMARY OF CONCLUSIONS AND RECOMMENDATIONS

6.1 SUMMARY OF CONCLUSIONS

In the previous chapter a model was presented that would simplify the selection of the methodology used to create the front end of a test adapter for a test fixture. The model took into account both qualitative parameters like the ease of maintenance and implementation as well as quantitative parameters like the relative reliability of the methodologies discussed.
This model should simplify the design process of test front ends.

6.2 PROBLEMS ENCOUNTERED

There were a number of obstacles during the course of this project, some of which related to finding an appropriate space in which to do the automated tests in. The final lab room that was used was not ideal specifically with regards to temperature fluxuations and should this sort of project be expanded upon, it would be good to identify a suitable location first since lab space is difficult to come by and expensive to rent.

Another problem encountered was in the implementation of the labview/teststand test architecture which was a bit more complex than initially envisioned but in the end once overcome was an absolute joy to work with.

The unresolved memory error though that was encountered and the lack of suitable error coding from the Labview side to actually diagnose the problem was something that was quite worrying – although it only happened once, a problem that happens once is bound to happen

again if it is not solved. Were this project to go on longer than the expected time, that would have been a major concern.

6.3 **HYPOTHESIS ADRESSED**

The initial problem statement was the need for a reliability based model for receptacle mounting methodology selection. Not only was the problem statement addressed directly in the creation of the model in chapter 5, but the sub problems were posed and hypothesis derived from them were also addressed as follows:

The first sub problem identified was that in order to create the model, a suitable way was required to measure the relative reliability of the different methodologies investigated in the course of creating the model. Chapter 3 and 4 dealt in depth with the design process followed in the creation of this relative reliability test system.

The second sub problem was that in order for the model to be created, not only would quantitative data need to be collected in the form of the relative reliability data, but there would also need to be a qualitative component to the model that needs to be taken into account based on the requirements of the user of the model. Chapter 4 explored in depth the different aspects of choosing a receptacle mounting methodology and from that created a relative ranking system which informed the creation of the model.

The third sub problem was related to the first in that once a system was designed to actually test the system, the sheer amount of data created would make it impractical to test manually. To overcome this significant time was spent in exploring different options for test automisation, and ultimately studying the Labview/Teststand test suite and creating a fully

automated system that could run completely autonomously and only require operator input should a failure arise.

6.4 **RECOMMENDATIONS FOR FURTHER STUDY**

The model was tested and found to deliver a suitable base for further work down the line, but as with any model there are always refinements to be made. The following recommendations could form the bases of future work in refining the model, as well as identifying some ideas that came about during the design phase (as well as results obtained from the testing and simulation phase) that could form the basis of future studies.

6.4.1 **Explore different pitches**

This study was concerned with the most commonly used pitch of receptical methodologies nl. 100mil, but this is not the only family of recepticals that is available. Recepticals can also be obtained for test probes that are set for pitches of 20mil, 30mil, 50 mil, 75mil etc. It would be useful to go through a similar comparison of the relative reliability, but also to explore the options on offer for the design of the signal conditioning since the finer pitch would allow for much more board real estate on pcb based mounting methodologies.

6.4.2 **Explore test point design**

At the start of this project, there was much consultation with people in industry and one of the main areas of concern raised was that during the stress test of the recepticals, it would be difficult to do a long term test due to the sharp test probes having a tendency to destroy test pads over time if a test is run multiple times.

However, during the reliability testing it was found that with the specific test point used on the sample UUT that failure on that interface was not a problem at all. It would be useful to explore the different failure modes that can present at the UUT/probe interface and to determine the cause for them since the initial assumption that they would just wear due to repeated impact was proven to be false.

6.4.3 **Expand the created model**

The model created should form a good base for future design, but since the parameters are all added into a single order equation to determine the suitability, it should be fairly easy to expand this and add many more qualitative and quantitative parameters to the system that are more specific to certain applications/use cases. For example, if a company does already have a system in place that makes different kinds of testers in house, it would be useful to take into account the impact of stock levels, as well as the relative skill levels of staff members that are available at the time for the implementation step to adjust the importance in the model based on the available capacity.

It would also be worth refining the current model in terms of the step size of the relative rankings of different parameters. Currently each of the parameters is rated out of 8, and each of the methodologies gets assigned an integer value between 1 and 8. This can however be refined by creating a questionnaire that can be distributed to various users of test recipticals that could have the qualitative parameters instead ranked with more precision based on a larger sample size of qualitative data.

6.4.4 **Explore reactance testing with sample and hold**

The sample and hold card designed for this project worked really well both in the simulation but also when tested on the bench. It should be possible using a digital potentiometer to adjust the RC constant of the sample and hold to build a more generic successive approximation reactance measurement device. This would work really well if implemented as a shield for an Arduino system.

6.5 **FINAL REMARKS**

This project has exposed the author to a fascinating world of software and automation and allowed for the exploration of many new technologies and software architectures which have already proven to be of benefit to him.

It is the author's belief that in the automation of somoe aspects of design it is in the interest of the designer to automate not only the work of the factory worker, but also of his or her fellow designers. The more we manage to employ technology to overtake our more mundane tasks, the further we shift the definition of what mundane tasks are and it is through this process that we are able to overcome more and more complex problems because we are not being overly burdened by simple ones.

REFERENCES

Scheiber, S. 2001. *Building a successful board test strategy. 2nd ed*. Boston, MA: Butterworth-Heinemann.

Hughes, E. 1995. *HUGHES Electrical Technology. 7th ed*. Essex, England: Longman.

Crowson, R. 2005. Factory Operations: Planning and Instructional Methods. Danvers, MA:CRC press.

Balangue, J. 2011. *In-circuit test – Still standing strong.* USA:Agilent Technologies.

Parker, K. 2011. *In-Circuit Test (ICT): The King Is Dead; Long Live the King!* USA:Agilent Technologies.

Hird, K., Parker, K., & Follis, B. 2011. *Test Coverage: What Does It Mean when a Board Test Passes?* USA:Agilent Technologies.

Tricker, R. 1997. *Quality And Standards In Electronics.2nd ed.* Boston, MA:Butterworth-Heinemann.

Technical Report, 1992. *RL-TR-92-197 Reliability Assessment of critical electronic components.* New York: IIT Research Institute.

Military Handbook 1995. *Military handbook reliability prediction of electronic equipment*. MIL-HDBK-217F notice 2. United states military, USA.

Military Handbook 1993. *Environmental stress screening (ESS) of electronic equipment.* MIL-HDBK-344A. United states military, USA.

Military Handbook 1960. *Sampling procedures and tables for life and reliability testing*. H108. United states military, USA.

Department of Defence standard practice 2012. *Connections, Electrical, Solderles wrapped*. MIL-STD-1130C. United states military, USA.

Department of Defence Detail Specification 1997. *Crimping Tools, Wire Termination, General specification*. MIL-DTL-22520G. United states military, USA.

Department of Defence Military Standard 1991. *Standard Requirements for soldered electrical and electronic assemblies*. MIL-STD-2000A. United states military, USA.

Department of Defence Military Specification 1991. *Military Specification Cable and harness assemblies, electrical, missile system: General specification for*. MIL-C-45224D. United states military, USA.

Ingun. 2010. *Test Probes Catalog 2010/2011*. Konstanz: Ingun Prufmitelbau GmbH.

Everet Charles Technologies. 2011. *Pogo Contacts*. Pomona, Calif: Dover.

Samtec. 2014. *Press fit headers & sockets*. USA, Samtec.

FIDES Guide, 2010. *FIDES Guide 2009 Reliability Methodology For Electronic Systems. Edition A*. France, FIDES group.

Applying Reliability DOE Techniques to Investigate Effects on Product Life. 2008. *Relibility Edge Journal of Reliabilit*y, 9(1), Q2

Morrison, G., Kallis, J., Stratton, L., Jones, I. & Lena, A. 1982. *RADC-TR-82-172 RADC Thermal Guide For Reliability Engineers.* New York: Hughes Aircraft Company.

Industrial Policy Action Plan, *2010. 2010/11 – 2012/13 Industrial Policy Action Plan*. South Africa:Economic Sectors And Employment Cluster.

Thermotron, 1998, *The Environmental Stress Screening Handbook.* Holland Michigan: Thermotron Industries.

Thermotron, 1998, *Fundamentals of accelerated stress screening.* Holland Michigan: Thermotron Industries.

Wikipedia, Wire wrap. Available from: http://www.en.wikipedia.org/wiki/Wire_wrap. [10 Aug 2013]

Microchip, 2009. *MCP9700/9700A MCP9701/9701A Low-Power Linear Active Thermistor ICs.* Chandler, Arizona: Microchip Technology Inc.

Electrolube, 2003. *Thermal Bonding Compound*. Derbyshire, UK. Electrolube, division of H K Wentworth.

Greeff, J.J. 2013. *TDA DAQ module user manual.* Gauteng, South Africa: TDA.

Greeff, J.J. 2013. *TDA Scalable DAQ user manual.* Gauteng, South Africa: TDA.

EPSMA, 2005. *Reliability Guidelines to Understanding Reliability Prediction.* Wellingborough, United Kingdom: European Power Supply Manufacturers Association.

Greeff, J.J. 2010. *Util Labs Generic Test Jig Framework.* Gauteng, South Africa: Tshwane University of Technology.

YAGEO, 2001. *Aluminium Electrolytic Capacitors Engineering Bulletin for SK Type.* Taiwan,

ANNEXURE A

SAMPLE AND HOLD DESIGN AND SIMULATIONS

The following simulations were used to verify the design of the relative reliability tester in software before testing the actual hardware. Simulations were done in various versions of LtSpice during design with the final waveforms being captured for this thesis on Version 4.21e.

A.1 SAMPLE AND HOLD

The Sample and hold circuitry is implemented to allow for a 100ms window of integration to be done over the input signal received from the receptacle signal chain. This is achieved by feeding the signal from the chain through an op amp buffer, and then using that signal to charge up a holding capacitor for the sample window time, and then measuring the voltage on that capacitor at the end of the window before dumping the voltage.

A.1.1 Measurement Linearity

Since the charge on the holding capacitor will needs to reflect the integral of the input signal, it stands to reason that the integration needs to be linear over the period. A capacitors charge curve however is not linear. This limitation can be overcome however by using the holding capacitor in its most linear region of the charging cycle which is the first time constant. For time constant

$$\tau = R \times C$$

so for 100ms time constant, a 10k resistor and 10uF capacitor are used.

The following page shows a simulation of the linearity of the charge voltage, when compared to a linear voltage ramp. The ramp moves from 70mV to 1.87V over 100mS, and it can be seen that the capacitors charge value follows the ramp with a deviation of ±80mV. Should the circuit need to be used for very accurate integration (for example if the

circuit was to be used to measure precise capacitance) then a software curve correction could be made to overcome the small non-linearity - This however is outside of the scope of this project. The spice circuit (and directives) used for the linearity calculations were as follows:

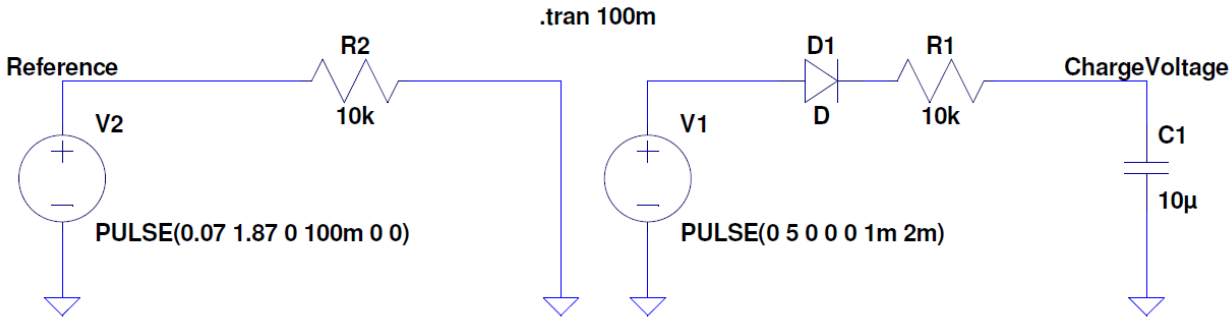


Illustration 1: Spice circuit for capacitor linearity calculations

And the referenced waveforms achieved are below:

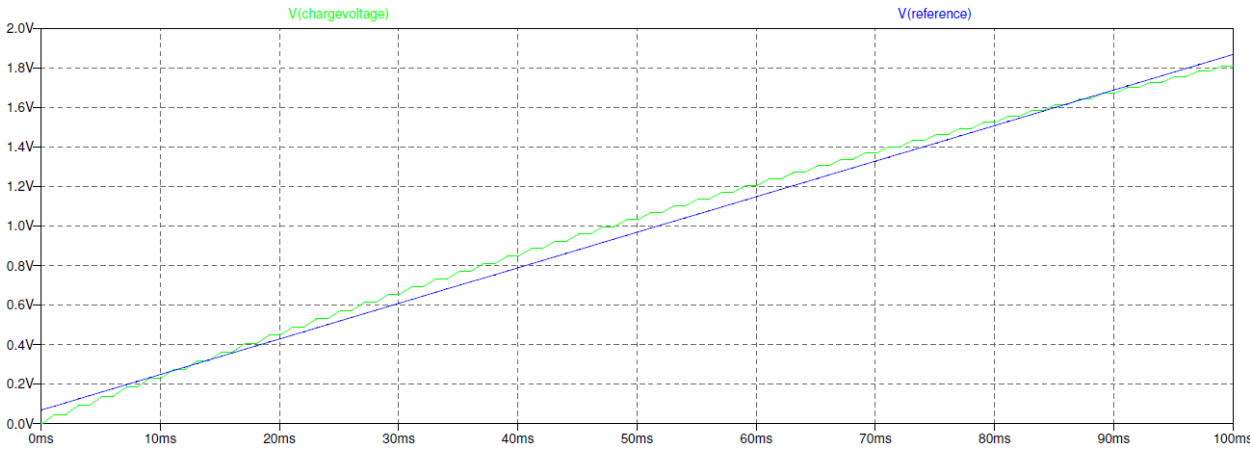


Illustration 2: Capacitor charge vs Reference

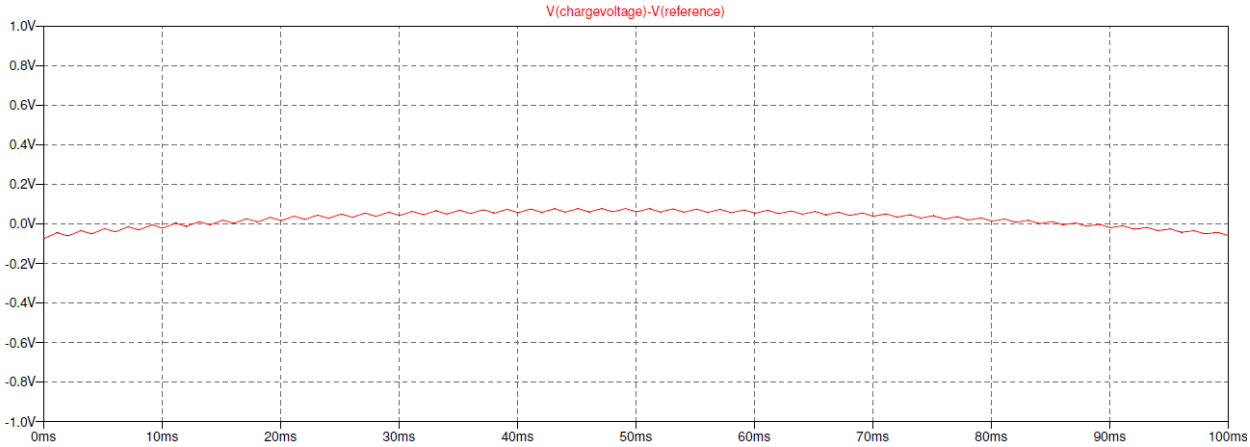


Illustration 3: Linearity deviation between charge voltage and reference

A.1.2 Ideal Sample and Hold circuit

The ideal circuit is as follows:

The 500Hz square wave used as the stimulus is voltage source Vin, and has a Vp-p of 5V,

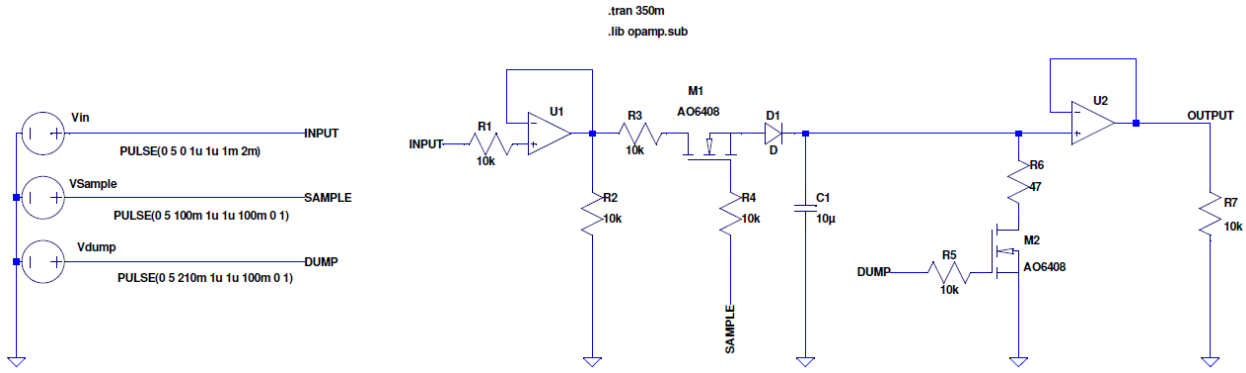


Illustration 4: Ideal Sample and hold Circuit

a 50% duty cycle and a period of 2ms. The rise and fall times are set at 1uS, which is done to give a more realistic pulse train.

The theory of operation is that the square wave will be buffered by U1, and then gated through an N Channel Mosfet M1 to charge up the holding capacitor C1 by applying a single 100ms pulse to its gate. This sample pulse is Vsample with a 5V peak voltage and has 1uS rise and fall times. D1 is placed in series with the mosfet so that while the M1 channel is opened during the sample phase, then C1 cannot discharge backwards.

The time constant for the charging current is set up by the combination of C1 and R3, and is the same as the time constant set up in the measurement linearity simulation.

During the sample time, C1 is allowed to charge up and represents the integral of the input signal. Once the sample pulse has been completed, M1 then goes closed, and the single point measurement can be taken on the OUTPUT line after U2 which acts as a second buffer to ensure that the measurement instrument doesn't impact the holding capacitor. During this measurement time the voltage on C1 has no path to ground with M1 and M2 both being off, and the buffer U2 having infinite input impedance (this however is only

true in the ideal circuit, non ideal op amps don't have infinite input impedance). This situation is held for 10ms to allow for the measurement.

Once the single point measurement is taken, the Dump signal Vdump is then applied to Mosfet M2. This then enables a low impedance path to ground for all of the charge on C1, effectively resetting the circuit.

The following waveforms show the ideal expected output for the ideal input signal.

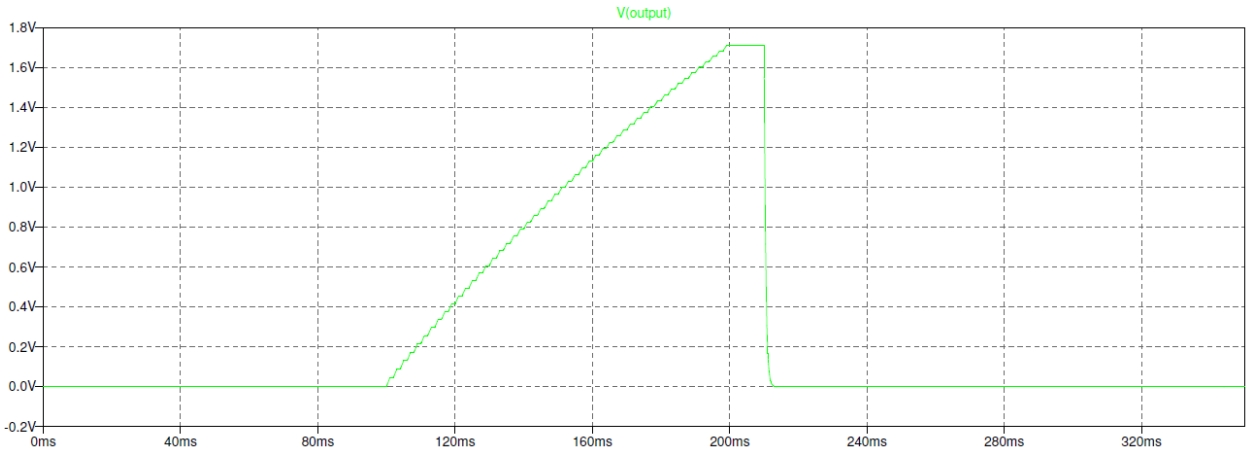


Illustration 5: Ideal output

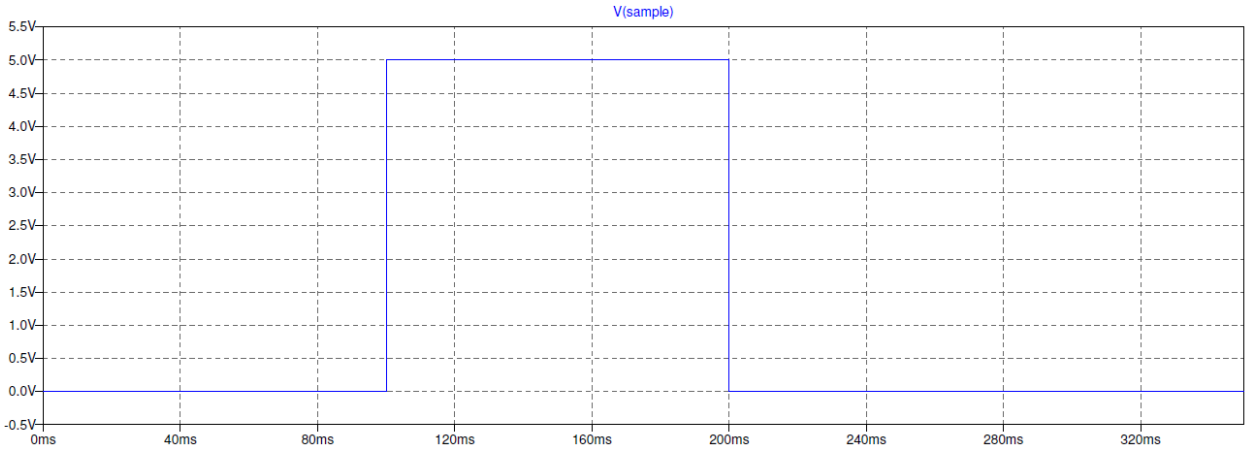


Illustration 6: Sample window

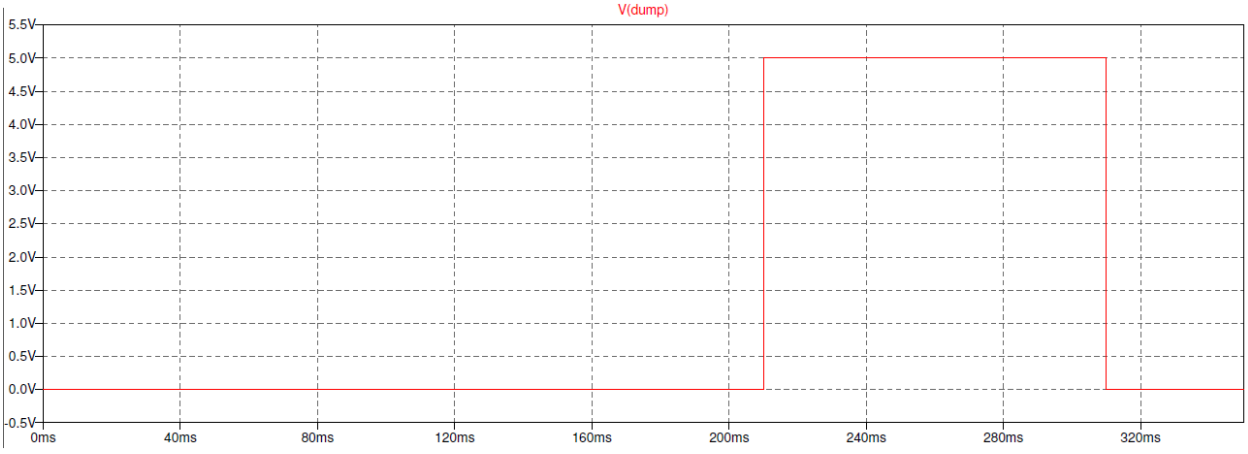


Illustration 7: Dump window

A.1.3 Non-ideal Sample and Hold circuit – no diode

The next circuit to be simulated was similar to the ideal circuit, but now using manufacturers models for the circuit elements. The decision was also made to attempt the simulation without the diode D1 in place to stop the feedback from the holding capacitor. The reason for this was that for the RC circuit, the current flowing while C1 is completely discharged during the start of the sample window will be 500uA, and this will become steadily less as the current decreases as C1 charges up. At these sorts of currents, it places the diode in the knee section of the forward voltage curve where the voltage drop across the diode is not constant but related to the amount of current across it.

The other option was to decrease the resistance of R1 in the RC circuit to increase the current flow into the holding capacitor C1, but this was decided against, because capacitors have much higher tolerances in capacitance value between different capacitors of the same brand and value. For example, the Yageo SK Type electrolytic capacitor range specifies a Capacitance tolerance of ± 20%, as opposed to the 1% of the resistive element. If the capacitance side of the RC equation was to become too dominant, then the differences between different Sample and Hold circuits would become too great. The opposite however is also relevant, since if the resistance element becomes greater, the current flow will also drop making the circuit more susceptable to noise (since the signal floor would become much lower).

The op-amp chosen to simulate the circuit was the LT1001 precision operational amplifier.

The dump signal was also changed to allow for a dump window to just before the sample window to ensure that the holding capacitor doesn't have any residual charge. This would not be a problem in the simulations, but in the actual circuit, this would ensure that ever sample operation happens from a known good state.

As expected, the peak output voltage was lower, however, with the simulation this was only 100mV less than the ideal circuit. The holding time was however worse than the ideal circuit, due to the leakage currents that are now taken into account. This was deemed not to be a problem though, because although the simulation offers 10ms of sample time, the actual measurement taken in hardware implementation would be done significantly faster (using a 200kHz ADC for example would mean the measurement could be taken at worst in 5uS) – the window was just left wider so that the waveforms could indicate the degradation of the held charge over time.

The Simulated circuit and corresponding waveforms are below.

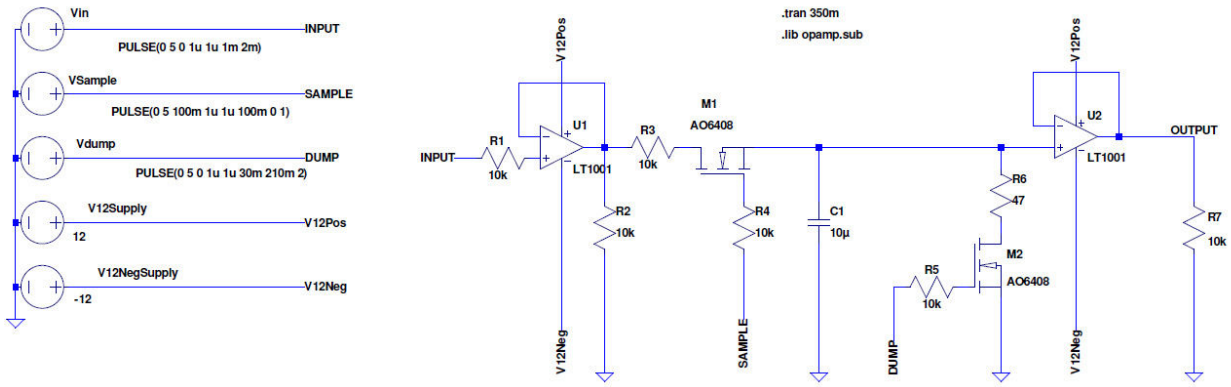


Illustration 8: Non-ideal Sample and hold circuit without diode

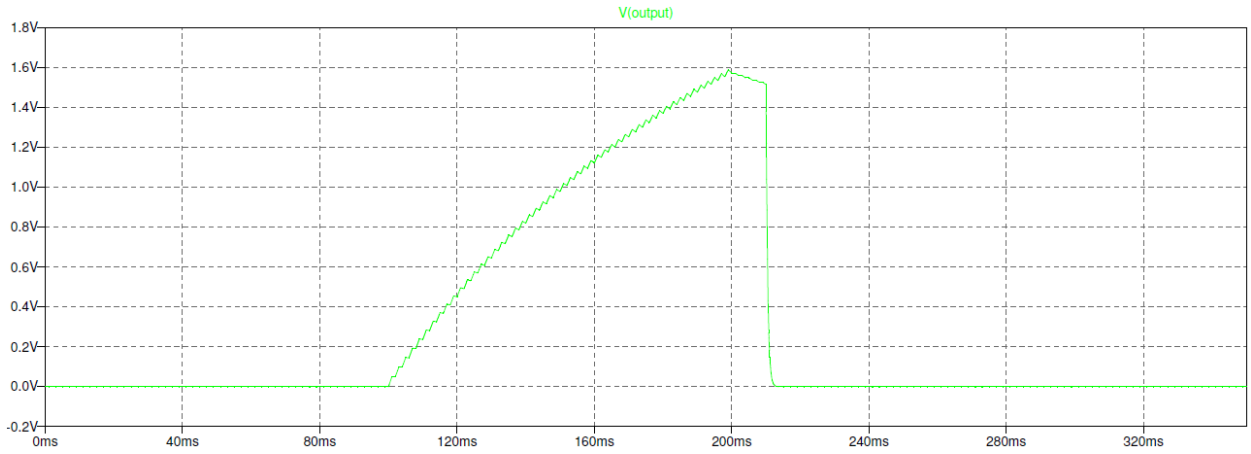


Illustration 9: Non-ideal output voltage

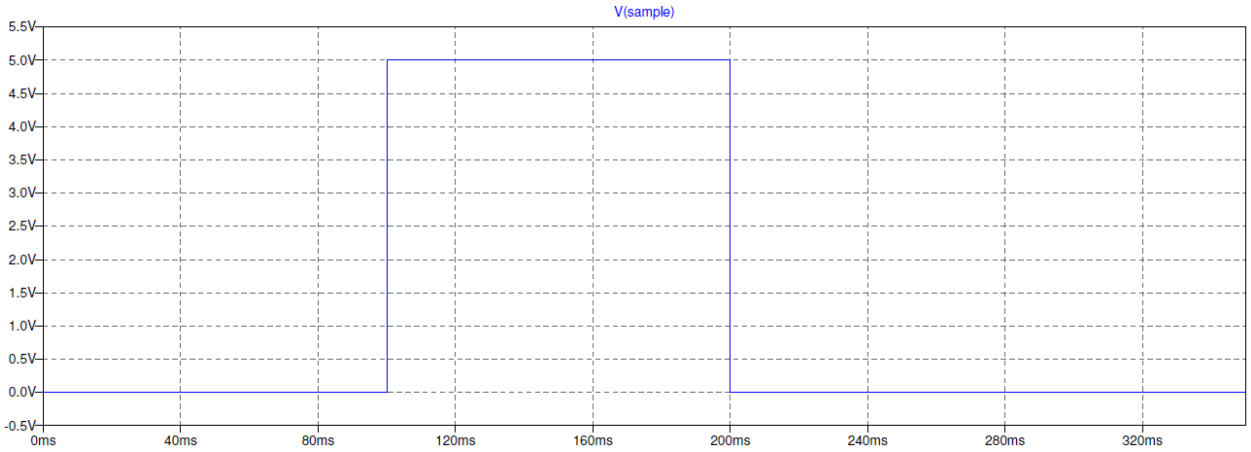


Illustration 10: Sample signal

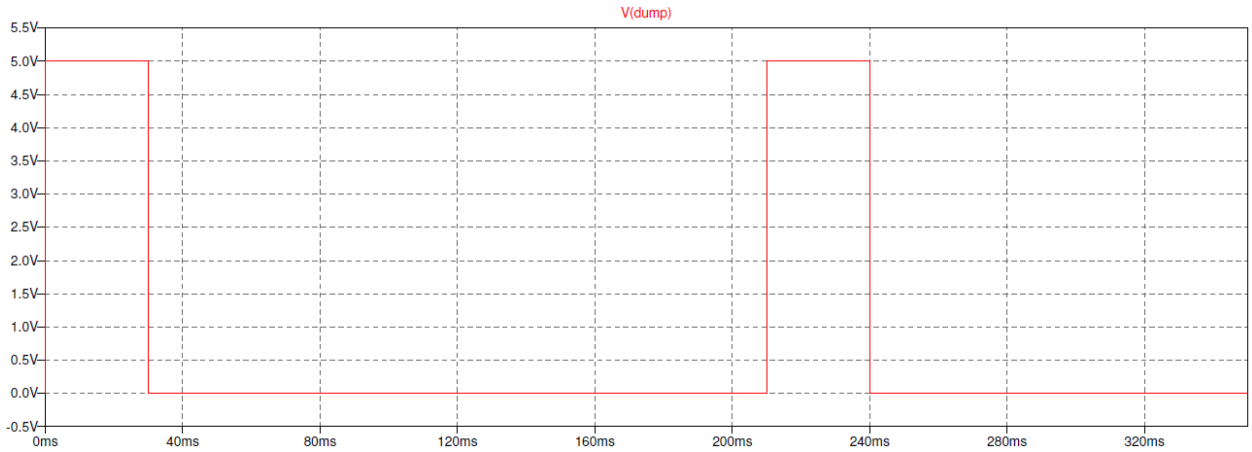


Illustration 11: Dump signal

A.1.4 Non-ideal Sample and Hold circuit – with input filter, parasitic capacitance and diode

Following what was learned from the non-ideal circuit simulation, the next step was to simulate the expected failure modes of the signal chain – ie. Adding series and paralel paracitic capacitances.

To simulate a small break in the signal chain the free air parallel plate model of capacitor was used with the formula:

$$C = \frac{(k \times \varepsilon_0 \times A)}{d}$$

where:

k = free space relative permittivity (approximately 1)

ε_0 = permittivity of space 8.854 x 10^{-12} F/m

A = Area of plates (wire thickness)

d = seperation between plates

Assuming a small break in the wire which places the wire 10um apart, and a 1mm^2 cross sectional surface area of wire, this will give a capacitance of 884pF. For the simulations, a parasitic capacitance then of 0.8nF was used.

For the input filter, to ensure that when parasitic capacitance is introduced during a failure that the capacitor doesn't simply drain into the signal generator, a series diode was used added after the signal generator (this won't necessarily impact the simulation, because the voltage sources don't sink current, but in the implemented hardware it would be beneficial as the signals are being generated with CMOS logic, so during the 0V portion of the duty cycle the output of the CMOS gate will be sinking current). A 120k resistor (R1, R12, R22) is also added to GND to create an RC time constant for the series/parallel parasitic capacitor.

Using this circuit, the clean signal chain (the signal chain with op-amps U1 and U2) measured 1.63V after the sample window, the signal chain with added parallel parasitic capacitance 2.24V and the signal chain with added series parasitic capacitance 192mV. From this it can be seen, that any rise in the output voltage compared to the reference signal will correspond to higher signal chain parallel capacitance, and a lower output voltage compared to reference signal will correspond to a broken line and series capacitance.

Since the signal is only ever positive, the decision was made to move from a dual rail precision op amp to a single supply precision op amp from the same family, ie the LT1006.

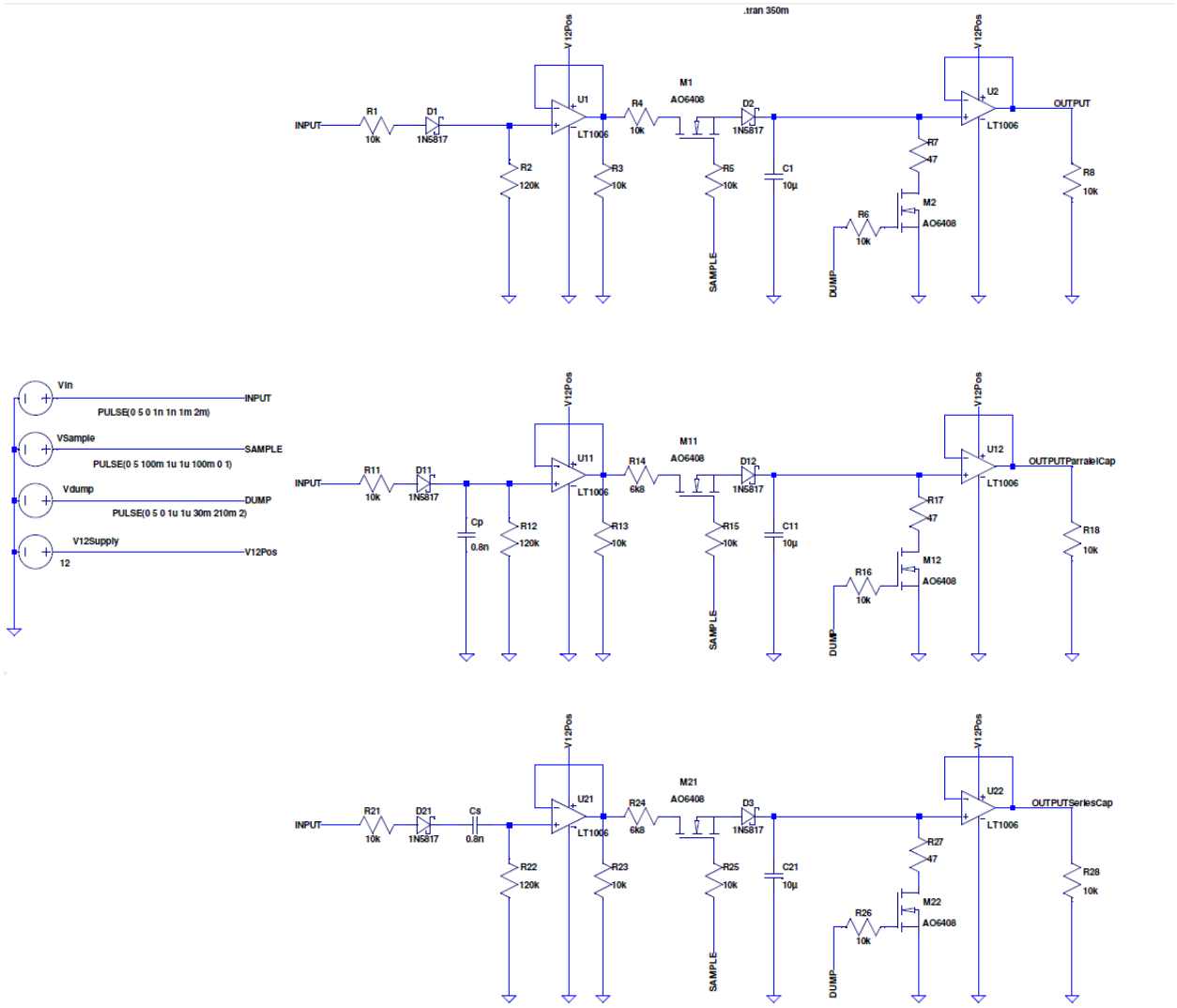


Illustration 12: Sample and Hold circuits with simulated breaks

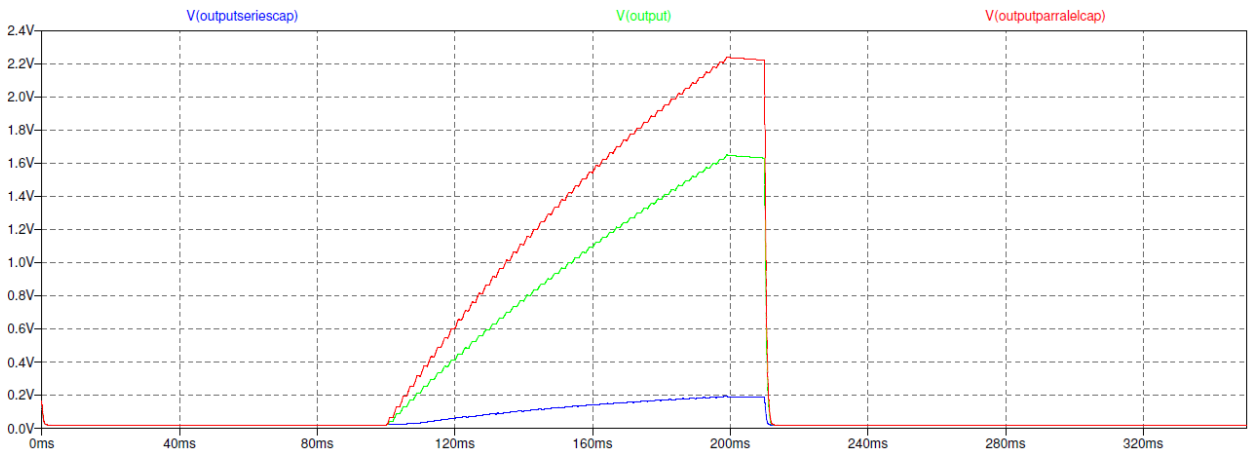


Illustration 13: Comparing Outputs of clean chain, with series parasitic capacitance and with paralell paracitic capacitance

A.1.5 Useable range of capacitive measurement

A.1.5.1 Paralell sweep simulation

Following the simulations of adding the parasitic capacitance, the next step was to simulate between what ranges the circuit would be able to determine whether the capacitance in the circuit had changed.

To simulate this a sweep simulation was done with both the parallel and series parasitic circuits. To change the range of the circuit, all that would need to be done would be to change the value of the input resistor (R12). If this was done programatically then the circuit could be used as a successive approximation capacitance meter.

For the 120k resistor and paralell parasitic capacitance the useable range was found to be between 1nF and 50nF – between 1nF and 20nF the capacitance relates linearly to the output voltage so if the circuit is to be used as a capacitance meter it would need to used in this range (for 120k input resistor). The reason for this when looking at the distortion on the input waveform when the RC circuit formed between Cp and R12 is that between 1nF and 20nF the square wave manages to get up to its maximum voltage (5V – D11 volt drop) so the full voltage is charged onto Cp. Beyond 20nF, the resistor doesn't allow Cp to charge quickly enough to reach the maximum. Beyond the linear point, the output voltage of the sample and hold circuit still goes higher, but it is no longer linearly related to the value of the paracitic capacitance. If the circuit is to be used to actually measure the absolute level of capacitance (rather than just relative changes like is being done in this application), it is important to note the linearity with relation to the R12 input resistor (as this is the part that will be programatically changed to move to different ranges).

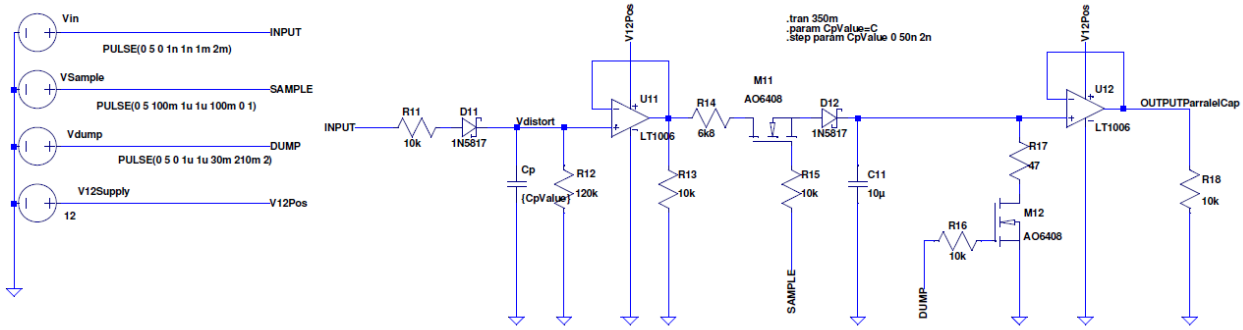


Illustration 14: Sweeped parasitic capacitor simulation using step directive

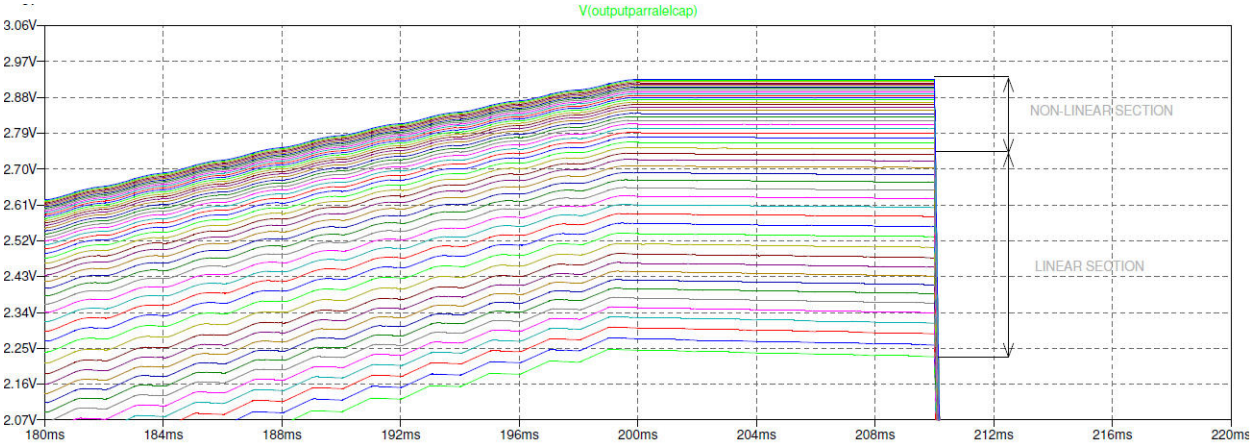


Illustration 16: Detail of output voltage showing linear and non linear portions of the output

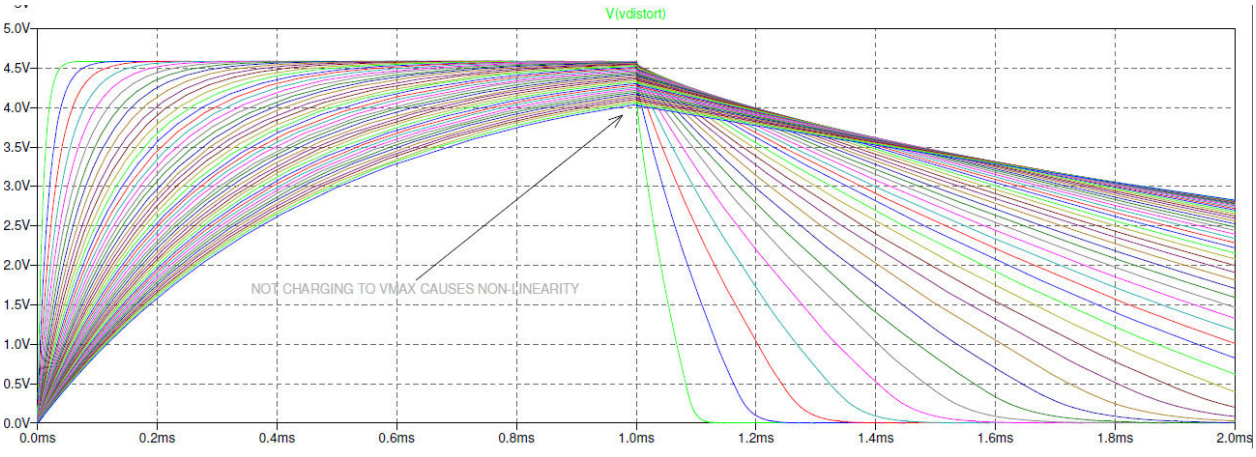


Illustration 17: Detail of distortion on input signal causing non linearity in output

ANNEXURE B

RELATIVE RELIABILITY TESTER DESIGN SCHEMATICS

The relative reliability tester consists of the following Pcb assemblies:

B.1 CONTROL BOARD

The control board houses connections to 2 of the TDA019 DAQs. The DAQ connections are as follows:

DAQ CHANNEL	FUNCTION	DESCRIPTION
IO1	Output	Channel 1 square wave signal to Buffer U101
IO2	Output	Channel 2 square wave signal to Buffer U101
IO3	Output	Channel 3 square wave signal to Buffer U101
FCIO1	Input	Temperature sensor Channel 3
FCIO2	Output	Buffer U101 Mode select (active/High Impedance)
FCIO3	Output	Channel 6 square wave signal to Buffer U101
FCIO4	Output	Channel 5 square wave signal to Buffer U101
FCIO5	Output	Channel 4 square wave signal to Buffer U101
DI1	Input	Temperature sensor Channel 1
DI2	Input	Temperature sensor Channel 2

Table 1: P102 DAQ Connection

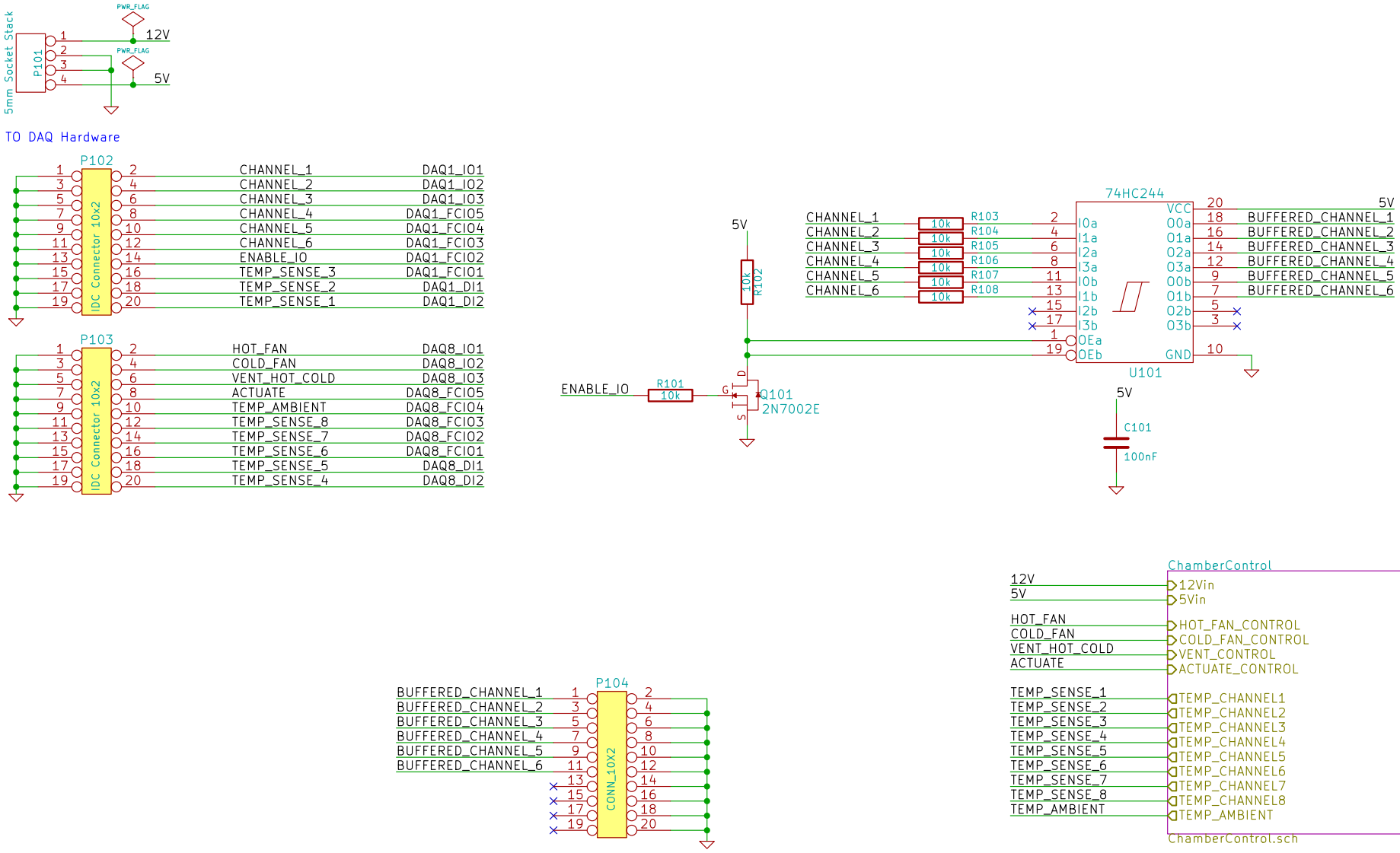
DAQ CHANNEL	FUNCTION	DESCRIPTION
IO1	Output	Actuate Heater
IO2	Output	Actuate Cooler
IO3	Output	Actuate Vent
FCIO1	Input	Temperature sensor Channel 6
FCIO2	Input	Temperature sensor Channel 7
FCIO3	Input	Temperature sensor Channel 8
FCIO4	Input	Temperature sensor Channel Ambient
FCIO5	Output	Actuate pneumatic drive
DI1	Input	Temperature sensor Channel 5
DI2	Input	Temperature sensor Channel 4

Table 2: P103 DAQ Connection

From the above DAQ connections, the 6 500Hz square wave channels are received, which are routed to U101 which is a 74HC244 CMOS buffer. The Buffer can be enabled or disabled by DAQ P102 as well.

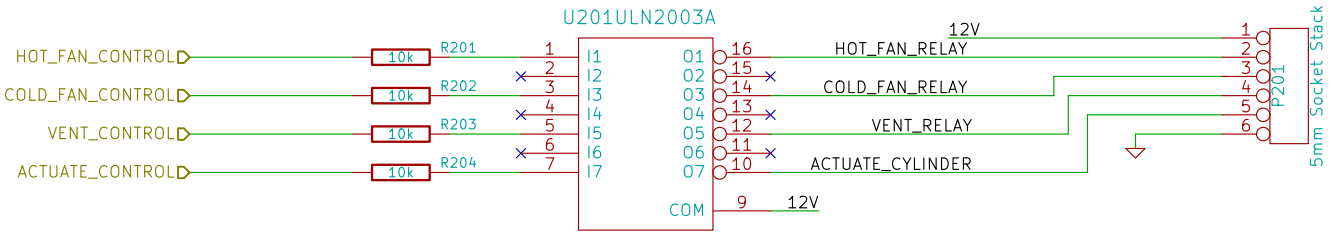
The Control channels which are used to actuate the heating and cooling as well as actuating the pneumatic drive are used to switch relays on the Relay card using U201, a ULN2003A darlington array.

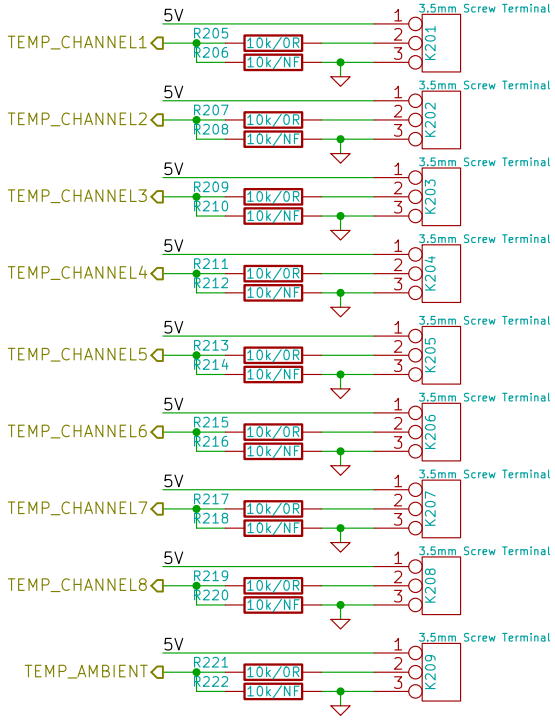
The temperature channels are fed in through the control board from the thermal probes and have optional resistor dividers on the board.



June 2013		
JJ Greeff		
Tshwane University of Technology		
File: Control.sch		
Sheet: /		
Title: Control		
Size: A4	*Date: 7 nov 2013*	*Rev: A*
KiCad E.D.A.		Id: 1/2

The Hot Relay will control both a heating element and a fan on a DPDT relay.
The Cold Relay will control both a Peltier module and a fan on a DPDT relay.
The VENT relay controls a pneumatic cylinder which diverts airflow between HOT and COLD inputs.
The Actuate relay controls a pneumatic cylinder which either engages or disengages the probes.





To potted MCP9700 sensor modules.
One sensor for each probe receptacle type.

File: ChamberControl.sch		
Sheet: /ChamberControl/		
Title:		
Size: A4	Date: 7 nov 2013	Rev:
KiCad E.D.A.		Id: 2/2

B.2 INTERFACE BOARD

The Interface boards contain the connections to the receptacles themselves as described in Chapter 3, section 3.1.1 – 3.1.8.

Each of the receptical mounting methodologies is applied on its own PCB, and each board has 10 receptacles connected to it – 5 for the incoming side of the receptical signal chains and 5 for the outputs of the receptical chains.

All the input and outputs on each of the boards are routed to a single 20 way ribbon connector, which is routed to the wiring board.

UUT

UUT.sch

DoublePcbInterface

DoublePcbInterface.sch

Crimp

Crimp.sch

PcbOnProbePlaten

PcbOnProbePlaten.sch

SolderCup

SolderCup.sch

DirectlySoldered

DirectlySoldered.sch

WireWrap

WireWrap.sch

Header

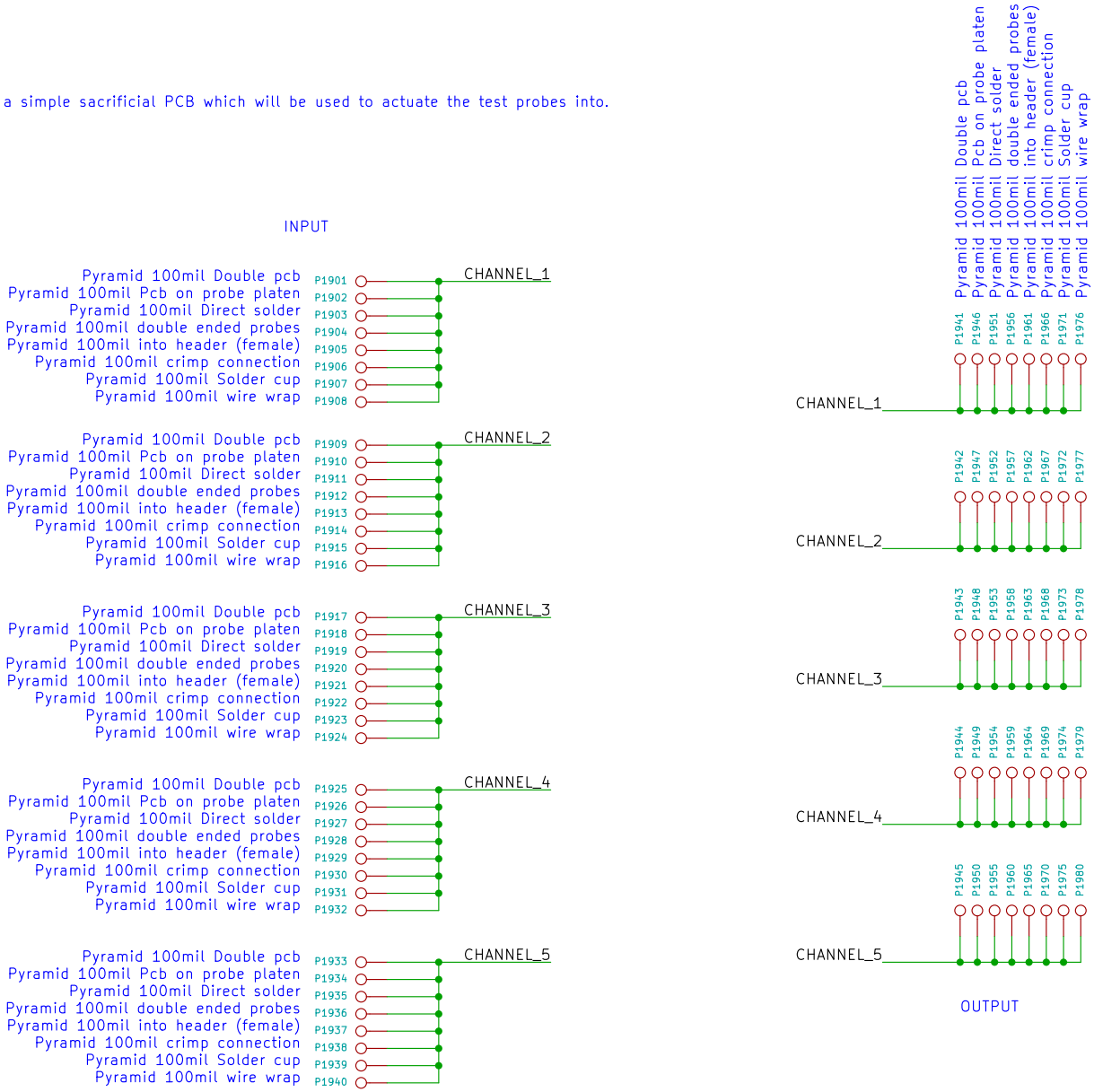
Header.sch

DoubleEndedProbes

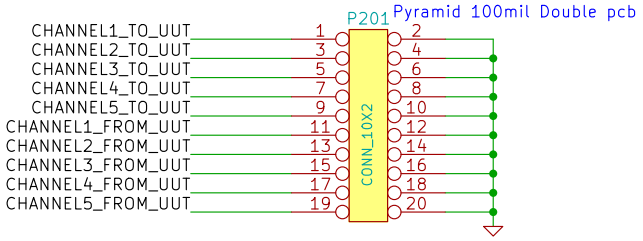
DoubleEndedProbes.sch

June 2013		
JJ Greeff		
Tshwane University of Technology		
File: InterfaceBoards.sch		
Sheet: /		
Title: Interface		
Size: A3	*Date: 9 nov 2013*	*Rev: A*
KiCad E.D.A.		Id: 1/10

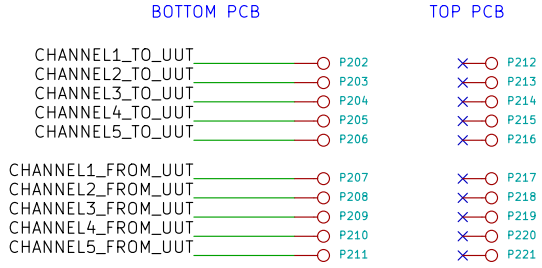
the UUT module is a simple sacrificial PCB which will be used to actuate the test probes into.



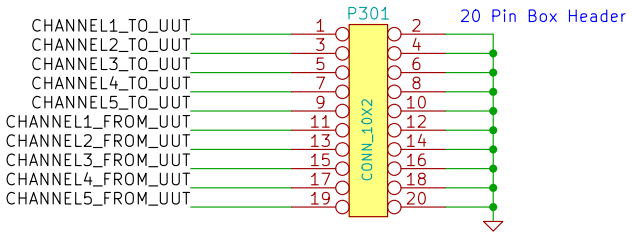
File: UUT.sch		
Sheet: /UUT/		
Title:		
Size: A4	Date: 9 nov 2013	Rev: A
KiCad E.D.A.		Id: 2/10



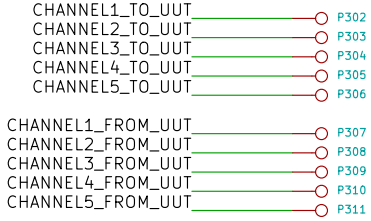
Note: The Double PCB mounting technique requires 2 PCBs, the bottom one containing all the tracks and interface connector and the top PCB only having holes to mount the receptacles into.
Both top and bottom PCB connections should be holes large enough for the receptacle body to be soldered into, but smaller than the retaining ring.



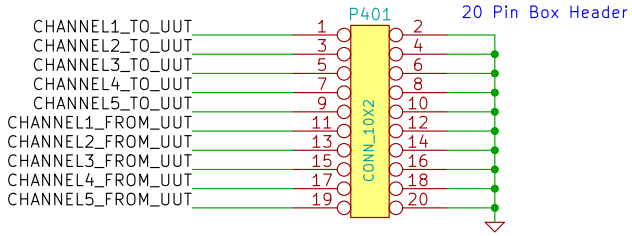
File: DoublePcbInterface.sch		
Sheet: /DoublePcbInterface/		
Title:		
Size: A4	*Date: 9 nov 2013*	*Rev: A*
KiCad E.D.A.		Id: 3/10



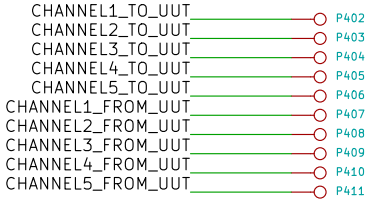
Note: With the Probe Platen mounting technique, the probe platen is used to allign the receptacle, and the recepticle is soldered into the PCB mounted on top of it. The mounting hole on the PCB should be large enough for the receptacle to fit through and small enough so it can stop the retaining ring. Since the track containing PCB is on the top of the probe platen the interface ribbon connector needs to move through a slot on the Probe Platen.



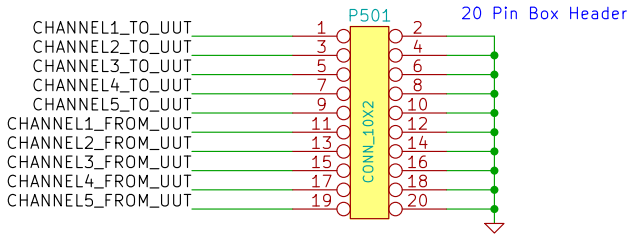
File: PcbOnProbePlaten.sch		
Sheet: /PcbOnProbePlaten/		
Title:		
Size: A4	*Date: 9 nov 2013*	*Rev: A*
KiCad E.D.A.		Id: 4/10



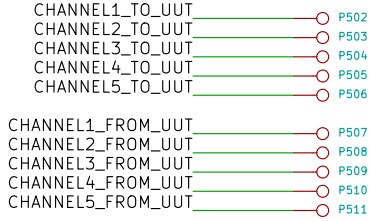
Note: With the Directly soldered technique, a wire wrap receptacle is friction fitted into the probe platen, and the wire wrap pin is soldered into the PCB. The mounting hole should fit a standard box header pin (wire wrap pin).



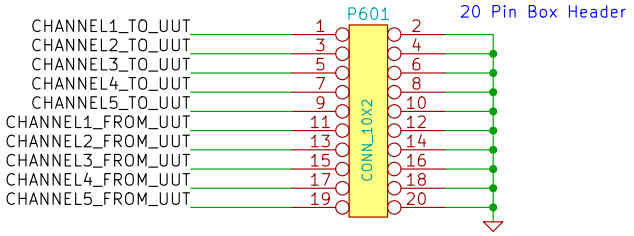
File: DirectlySoldered.sch		
Sheet: /DirectlySoldered/		
Title:		
Size: A4	Date: 9 nov 2013	Rev: A
KiCad E.D.A.		Id: 5/10



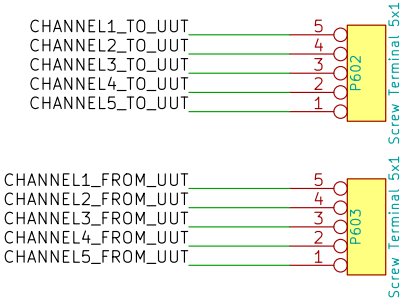
Note: With the header mounting technique, the wire wrap recepticle isfriction fitted into the probe platen and the pin is fitted into a female header pin socket.



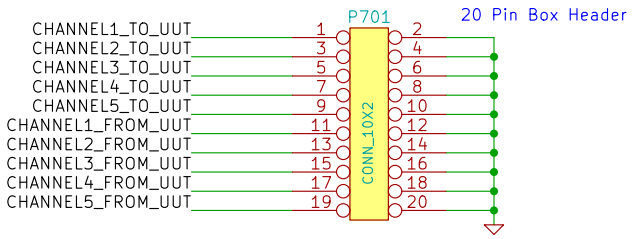
File: Header.sch		
Sheet: /Header/		
Title:		
Size: A4	Date: 9 nov 2013	Rev: A
KiCad E.D.A.		Id: 6/10



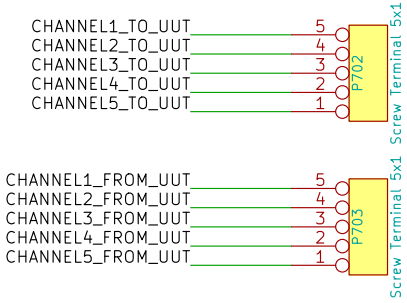
Note: With the crimp mounting technique, a crimped recepticle is friction fitted into the Probe Platen, and then the crimped wire can be connected to the PCB by either soldering it, or with screw terminal connections.



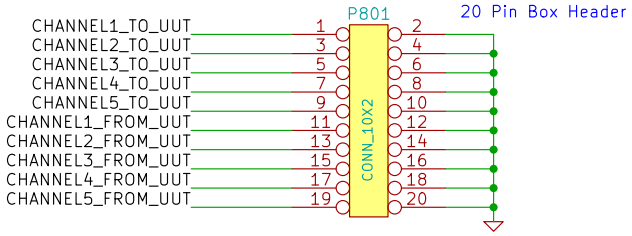
File: Crimp.sch		
Sheet: /Crimp/		
Title:		
Size: A4	Date: 9 nov 2013	Rev: A
KiCad E.D.A.		Id: 7/10



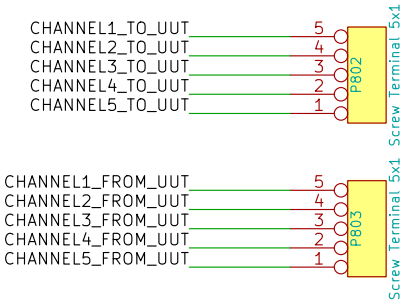
Note: With the Solder Cup mounting technique, a solder recepticle is friction fitted into the Probe Platen, and then the soldered wire is either soldered into the PCB or attached with screw terminals.



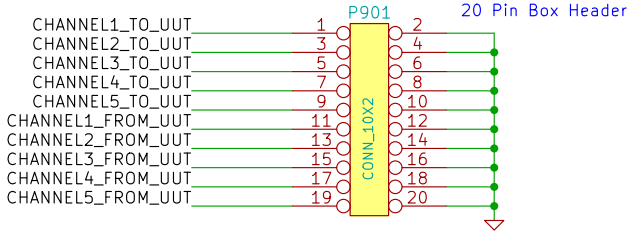
File: SolderCup.sch		
Sheet: /SolderCup/		
Title:		
Size: A4	Date: 9 nov 2013	Rev: A
KiCad E.D.A.		Id: 8/10



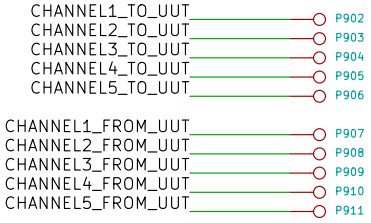
Note: With the Wire Wrap mounting technique, a Wire Wrap recepticle is friction fitted into the Probe Platen, and then the wire wrapped wire is either soldered into the PCB or attached with screw terminals.



File: WireWrap.sch		
Sheet: /WireWrap/		
Title:		
Size: A4	Date: 9 nov 2013	Rev: A
KiCad E.D.A.		Id: 9/10



Note: With the double ended probe mounting technique a double ended recepticle is friction fitted into the probe platen and then the receptacle spring loaded pin is pushed onto a pad on the PCB.

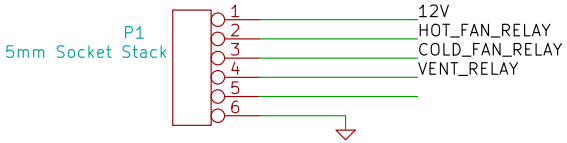


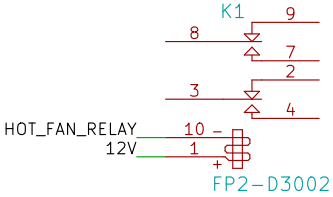
File: DoubleEndedProbes.sch		
Sheet: /DoubleEndedProbes/		
Title:		
Size: A4	*Date: 9 nov 2013*	*Rev: A*
KiCad E.D.A.		Id: 10/10

B.3 RELAY BOARD

The relay board is driven by the darlington driver chip on the Control board and is used to drive a number of G5LA1E12DC Relays which are wired to the Cooler, Heater Array and Penumatic drive.

The Hot Relay will control both a heating element
and a fan on a DPDT relay.
The Cold Relay will control both a Peltier module and
a fan on a DPDT relay.
The VENT relay controls a pneumatic cylinder which
diverts airflow between HOT and COLD inputs.
The Actuate relay controls a pneumatic cylinder which
either engages or disengages the probes.





12V
VENT_RELAY

ACTUATE_RELAY

June 2013 JJ Greeff *Tshwane University of Technology*		
File: Relays.sch Sheet: / *Title: Relays*		
Size: A4	*Date: 8 nov 2013*	*Rev: A*
KiCad E.D.A.		Id: 1/1

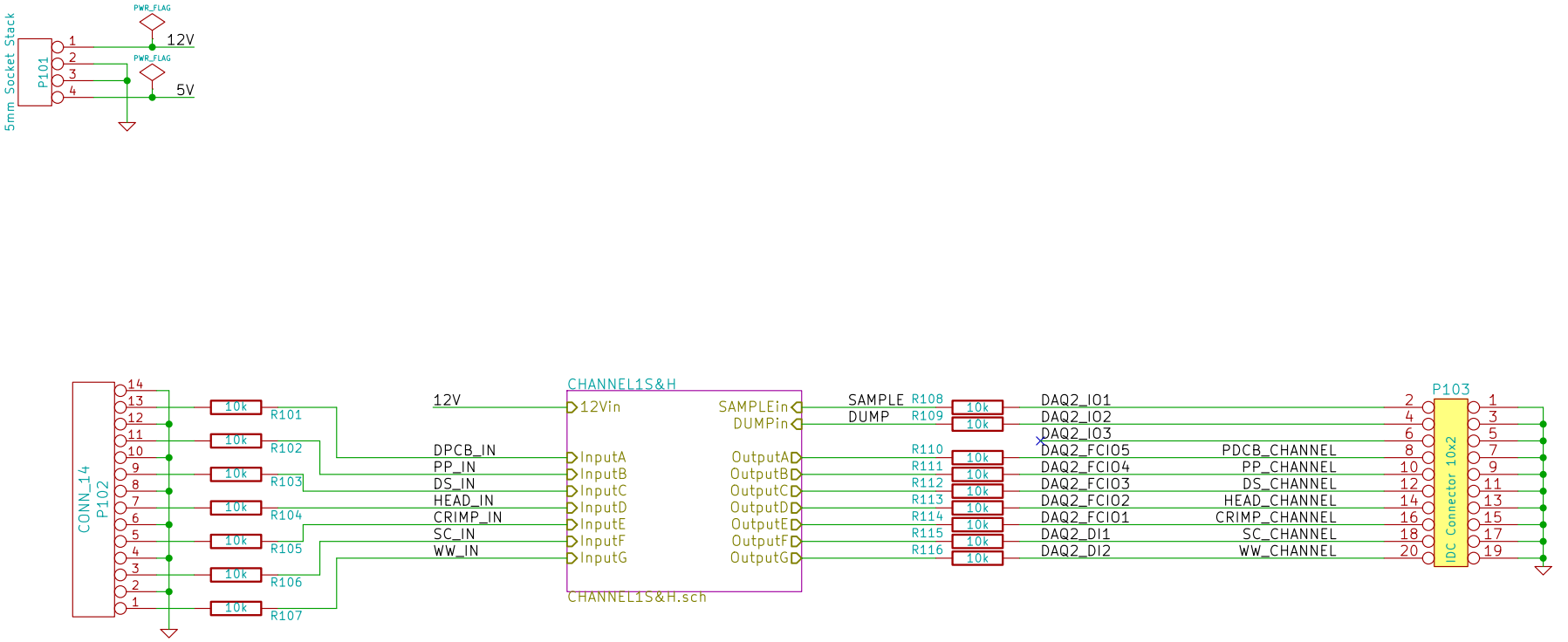
B.4 SAMPLE AND HOLD BOARD

The Sample and Hold board holds the sample and hold circuitry simulated in Annexure A and described in Chapter 4 section 4.1.1.1. Each Sample and hold board has 7 channels of Sample and hold circuits.

Each Sample and Hold Board connects to one of the TDA019 Daq Devices through ribbon cable P103. The Daq connections are as follows:

DAQ CHANNEL	FUNCTION	DESCRIPTION
IO1	Output	Sample signal input for all 7 channels
IO2	Output	Dump signal input for all 7 channels
IO3	NC	Not connected
FCIO1	Input	Sample and hold channel 1
FCIO2	Input	Sample and hold channel 2
FCIO3	Input	Sample and hold channel 3
FCIO4	Input	Sample and hold channel 4
FCIO5	Input	Sample and hold channel 5
DI1	Input	Sample and hold channel 6
DI2	Input	Sample and hold channel 7

Table 3: P103 DAQ Connection



June 2013		
JJ Greeff		
Tshwane University of Technology		
File: SampleAndHold.sch		
Sheet: /		
Title: Sample and Hold		
Size: A4	*Date: 26 jul 2014*	*Rev: A*
KiCad E.D.A.		Id: 1/2

12Vin
SAMPLEin
DUMPin
12VSupply
SAMPLE
DUMP

InputA
R201 10k/OR
R202 10k/NF
U201A LM324
R209 10k
Q201 2N7002E
C201 10uF 16V
R213 10k
R221 47
Q205 2N7002E
R217 10k
U202A LM324
OutputD

InputB
R203 10k/OR
R204 10k/NF
U201B LM324
R210 10k
Q202 2N7002E
C202 10uF 16V
R214 10k
R222 47
Q206 2N7002E
R218 10k
U202B LM324
OutputC

InputC
R205 10k/OR
R206 10k/NF
U201C LM324
R211 10k
Q203 2N7002E
C203 10uF 16V
R215 10k
R223 47
Q207 2N7002E
R219 10k
U202C LM324
OutputA

InputD
R207 10k/OR
R208 10k/NF
U201D LM324
R212 10k
Q204 2N7002E
C204 10uF 16V
R216 10k
R224 47
Q208 2N7002E
R220 10k
U202D LM324
OutputB

InputE
R225 10k/OR
R226 10k/NF
U203A LM324
R231 10k
Q209 2N7002E
C205 10uF 16V
R234 10k
R240 47
Q212 2N7002E
R237 10k
U204A LM324
OutputE

InputF
R227 10k/OR
R228 10k/NF
U203B LM324
R232 10k
Q210 2N7002E
C206 10uF 16V
R235 10k
R241 47
Q213 2N7002E
R238 10k
U204B LM324
OutputF

InputG
R229 10k/OR
R230 10k/NF
U203C LM324
R233 10k
Q211 2N7002E
C207 10uF 16V
R236 10k
R242 47
Q214 2N7002E
R239 10k
U204C LM324
OutputG

C208 100nF
C209 100nF
C210 100nF
C211 100nF
C212 100nF
C213 100nF
C214 100nF

U203D LM324
U204D LM324

File: CHANNEL1S&H.sch		
Sheet: /CHANNEL1S&H/		
Title:		
Size: A3	Date: 26 jul 2014	Rev:
KiCad E.D.A.		Id: 2/2

B.5 WIRING BOARD

The Wiring board contains the wiring connections between each of the 8 Interface boards and the Sample and hold Boards (outputs) and the buffered channels (inputs).

Each of the 8 Interface boards has 5 Input/Output signal chains for a total of 40 signal chains. Since each Sample and hold board has 7 channels, this means there are 42 available channels over 6 sample and hold boards. The wiring board routes each of these signal chains to one of the channels.

To Pyramid 100mil Double pcb

P1

BUFFERED_CHANNEL_1 1 2
BUFFERED_CHANNEL_2 3 4
BUFFERED_CHANNEL_3 5 6
BUFFERED_CHANNEL_4 7 8
BUFFERED_CHANNEL_5 9 10
PDCB_CHANNEL1 11 12
PDCB_CHANNEL2 13 14
PDCB_CHANNEL3 15 16
PDCB_CHANNEL4 17 18
PDCB_CHANNEL5 19 20

CONN_10X2

To Pyramid 100mil Pcb on probe platen

P2

BUFFERED_CHANNEL_1 1 2
BUFFERED_CHANNEL_2 3 4
BUFFERED_CHANNEL_3 5 6
BUFFERED_CHANNEL_4 7 8
BUFFERED_CHANNEL_5 9 10
PP_CHANNEL1 11 12
PP_CHANNEL2 13 14
PP_CHANNEL3 15 16
PP_CHANNEL4 17 18
PP_CHANNEL5 19 20

CONN_10X2

To Pyramid 100mil Direct solder

P3

BUFFERED_CHANNEL_1 1 2
BUFFERED_CHANNEL_2 3 4
BUFFERED_CHANNEL_3 5 6
BUFFERED_CHANNEL_4 7 8
BUFFERED_CHANNEL_5 9 10
DS_CHANNEL1 11 12
DS_CHANNEL2 13 14
DS_CHANNEL3 15 16
DS_CHANNEL4 17 18
DS_CHANNEL5 19 20

CONN_10X2

To Pyramid 100mil into header (female)

P4

BUFFERED_CHANNEL_1 1 2
BUFFERED_CHANNEL_2 3 4
BUFFERED_CHANNEL_3 5 6
BUFFERED_CHANNEL_4 7 8
BUFFERED_CHANNEL_5 9 10
HEAD_CHANNEL1 11 12
HEAD_CHANNEL2 13 14
HEAD_CHANNEL3 15 16
HEAD_CHANNEL4 17 18
HEAD_CHANNEL5 19 20

CONN_10X2

To Pyramid 100mil crimp connection

P5

BUFFERED_CHANNEL_1 1 2
BUFFERED_CHANNEL_2 3 4
BUFFERED_CHANNEL_3 5 6
BUFFERED_CHANNEL_4 7 8
BUFFERED_CHANNEL_5 9 10
CRIMP_CHANNEL1 11 12
CRIMP_CHANNEL2 13 14
CRIMP_CHANNEL3 15 16
CRIMP_CHANNEL4 17 18
CRIMP_CHANNEL5 19 20

CONN_10X2

To Pyramid 100mil Solder cup

P6

BUFFERED_CHANNEL_1 1 2
BUFFERED_CHANNEL_2 3 4
BUFFERED_CHANNEL_3 5 6
BUFFERED_CHANNEL_4 7 8
BUFFERED_CHANNEL_5 9 10
SC_CHANNEL1 11 12
SC_CHANNEL2 13 14
SC_CHANNEL3 15 16
SC_CHANNEL4 17 18
SC_CHANNEL5 19 20

CONN_10X2

To Pyramid 100mil wire wrap

P7

BUFFERED_CHANNEL_1 1 2
BUFFERED_CHANNEL_2 3 4
BUFFERED_CHANNEL_3 5 6
BUFFERED_CHANNEL_4 7 8
BUFFERED_CHANNEL_5 9 10
WW_CHANNEL1 11 12
WW_CHANNEL2 13 14
WW_CHANNEL3 15 16
WW_CHANNEL4 17 18
WW_CHANNEL5 19 20

CONN_10X2

To Pyramid 100mil double ended probes

P8

BUFFERED_CHANNEL_6 1 2
BUFFERED_CHANNEL_6 3 4
BUFFERED_CHANNEL_6 5 6
BUFFERED_CHANNEL_6 7 8
BUFFERED_CHANNEL_6 9 10
DEP_PROBE1_CHANNEL6 11 12
DEP_PROBE2_CHANNEL6 13 14
DEP_PROBE3_CHANNEL6 15 16
DEP_PROBE4_CHANNEL6 17 18
DEP_PROBE5_CHANNEL6 19 20

CONN_10X2

To Control Board

P9

BUFFERED_CHANNEL_1 1 2
BUFFERED_CHANNEL_2 3 4
BUFFERED_CHANNEL_3 5 6
BUFFERED_CHANNEL_4 7 8
BUFFERED_CHANNEL_5 9 10
BUFFERED_CHANNEL_6 11 12
13 14
15 16
17 18
19 20

CONN_10X2

To Sample and Hold

P10

1 2 PDCB_CHANNEL1 DAQ2_FCIO5
3 4 PP_CHANNEL1 DAQ2_FCIO4
5 6 DS_CHANNEL1 DAQ2_FCIO3
7 8 HEAD_CHANNEL1 DAQ2_FCIO2
9 10 CRIMP_CHANNEL1 DAQ2_FCIO1
11 12 SC_CHANNEL1 DAQ2_DI1
13 14 WW_CHANNEL1 DAQ2_DI2

CONN_7X2

P11

1 2 PDCB_CHANNEL2 DAQ3_FCIO5
3 4 PP_CHANNEL2 DAQ3_FCIO4
5 6 DS_CHANNEL2 DAQ3_FCIO3
7 8 HEAD_CHANNEL2 DAQ3_FCIO2
9 10 CRIMP_CHANNEL2 DAQ3_FCIO1
11 12 SC_CHANNEL2 DAQ3_DI1
13 14 WW_CHANNEL2 DAQ3_DI2

CONN_7X2

P12

1 2 PDCB_CHANNEL3 DAQ4_FCIO5
3 4 PP_CHANNEL3 DAQ4_FCIO4
5 6 DS_CHANNEL3 DAQ4_FCIO3
7 8 HEAD_CHANNEL3 DAQ4_FCIO2
9 10 CRIMP_CHANNEL3 DAQ4_FCIO1
11 12 SC_CHANNEL3 DAQ4_DI1
13 14 WW_CHANNEL3 DAQ4_DI2

CONN_7X2

P13

1 2 PDCB_CHANNEL4 DAQ5_FCIO5
3 4 PP_CHANNEL4 DAQ5_FCIO4
5 6 DS_CHANNEL4 DAQ5_FCIO3
7 8 HEAD_CHANNEL4 DAQ5_FCIO2
9 10 CRIMP_CHANNEL4 DAQ5_FCIO1
11 12 SC_CHANNEL4 DAQ5_DI1
13 14 WW_CHANNEL4 DAQ5_DI2

CONN_7X2

P14

1 2 PDCB_CHANNEL5 DAQ6_FCIO5
3 4 PP_CHANNEL5 DAQ6_FCIO4
5 6 DS_CHANNEL5 DAQ6_FCIO3
7 8 HEAD_CHANNEL5 DAQ6_FCIO2
9 10 CRIMP_CHANNEL5 DAQ6_FCIO1
11 12 SC_CHANNEL5 DAQ6_DI1
13 14 WW_CHANNEL5 DAQ6_DI2

CONN_7X2

P15

1 2
3 4
5 6 DEP_PROBE1_CHANNEL6 DAQ7_FCIO3
7 8 DEP_PROBE2_CHANNEL6 DAQ7_FCIO2
9 10 DEP_PROBE3_CHANNEL6 DAQ7_FCIO1
11 12 DEP_PROBE4_CHANNEL6 DAQ7_DI1
13 14 DEP_PROBE5_CHANNEL6 DAQ7_DI2

CONN_7X2

June 2013		
JJ Greeff		
Tshwane University of Technology		
File: Wiring.sch		
Sheet: /		
Title: Wiring		
Size: A3	*Date: 29 oct 2013*	*Rev: A*
KiCad E.D.A.		Id: 1/1

B.6 SACRIFICIAL UUT BOARD

The sacrificual UUT boards only contain 5 tracks and 10 test points – 5 test points on the left to connect to the “input” side probe of the signal chain, and 5 test points on the right to connect to the “output” side probe of the signal chain.

the UUT module is a simple sacrificial PCB which will be used to actuate the test probes into.

File: UUT.sch		
Sheet: /		
Title:		
Size: A4	Date: 8 apr 2014	Rev: A
KiCad E.D.A.		Id: 1/1

ANNEXURE C

SOFTWARE

The relative reliability tester software consists of both Labview Vis to implement the communication between the TDA019D Daq system, as well as a teststand sequence used to control the automated testing process.

Sequence Documentation is generated using Teststand 2012 SP1.

All of the used Vis and sequences can be found on the accompanying CD.

C.1 Teststand

The following pages house the generated Teststand documentation.

Sequence File Documentation Created On:

- **Date:** **17 November 2014**
- **Time:** **03:35:08 PM**
- **By Operator:** **administrator**

Path:	C:\projects\MTech\TestSequence.seq
Version:	0.0.0.0
Type:	Normal
Module Load Option:	Use step load option
Module Unload Option:	Use step unload option
Model Option:	Use Station Model
Number of sequences:	6
Total number of steps:	395

Sequence File Globals:		
HighPassCeilig	Number	2.7
HighPassFloor	Number	2
LowPassPoint	Number	0.3
pollTemperatureThreadActive	Boolean	False
heartbeatThreadActive	Boolean	False
DaqPort	String	"COM3"
LogFileSemaphore	String	"Semaphore1"
ComPortSemaphore	String	"Semaphore2"
LogFile	String	""
TestingActive	Boolean	False
ActiveChannels	Array of Booleans[0..39]	...

Sequence: MainSequence

Type:	Normal
Disable Results for All Steps:	False
Goto Cleanup on failure:	No Action
Number of Setup steps:	28
Number of Main steps:	240
Number of Cleanup steps:	5

Locals:		
SequenceNumber	Number	0
MeasuredPoints	Array of Numbers[0..6]	...
MsgIdToSend	Number	0
ResultList	Array of Result[0..empty]	...

Setup

Step: GetMessageId - clear description

StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage

StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Ping Description", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - clear results

StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage

StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Ping Description", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - clear console

StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage

StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Ping Description", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Create Log File for this session

StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\CreateLog.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Create Log file Semaphore

StepType, Adapter:	Semaphore, <None>
Description:	Create(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Create Com Port Semaphore

StepType, Adapter:	Semaphore, <None>
Description:	Create(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: set up the I/O functions for all of the Daq devices

StepType, Adapter:	Label, <None>
Description:	
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Record Results:	Disabled

Step: Aquire Semaphore

StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Set Slot 1 functions

StepType, Adapter:	Action, LabVIEW

Description:	Action, builds\console\My Source Distribution.llb\SetIoFunction.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Set Slot 2 functions	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetIoFunction.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Set Slot 3 functions	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetIoFunction.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Set Slot 4 functions	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetIoFunction.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Set Slot 5 functions	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetIoFunction.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Set Slot 6 functions	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetIoFunction.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Set Slot 7 functions	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetIoFunction.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Set Slot 8 functions	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetIoFunction.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Enable Channel Outputs	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetBufferOutputEnableState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Set the test mode to Active	
StepType, Adapter:	Statement, <None>
Description:	FileGlobals.TestingActive = True
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Spawn Temperature Monitor Thread	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call Poll Temperatures in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	New Thread

Step: Spawn Heartbeat Thread	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call Heartbeat in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	New Thread

Step: What are the UUT Serials? Are they starting from a baseline?	
StepType, Adapter:	Label, <None>
Description:	
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Record Results:	Disabled

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Actuate Low (ensure you start in a low state)	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetActuatorState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Main

Step: While	
StepType, Adapter:	NI_Flow_While, <None>
Description:	FileGlobals.TestingActive
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Increment sequence number	
StepType, Adapter:	Statement, <None>
Description:	Locals.SequenceNumber++
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log Sequence number to file	
StepType, Adapter:	Action, LabVIEW

Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - update Console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Stressing the connections \n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Stress test	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call Exercise in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - update Console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Stress test complete \n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - Update Active signal chains	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Actuate Low (ensure you start in a low state)	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetActuatorState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Wait for actuation to complete	
StepType, Adapter:	Wait, <None>
Description:	TimeInterval(2)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - update Console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Starting Test set \n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Actuate High	
StepType, Adapter:	Label, <None>
Description:	
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Record Results:	Disabled

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Actuate High	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetActuatorState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Wait for actuation to complete	
StepType, Adapter:	Wait, <None>
Description:	TimeInterval(2)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: _____Test Chain 1 steps_____ (if there are active lines in it)	
StepType, Adapter:	Label, <None>
Description:	
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Record Results:	Disabled

Step: GetMessageId - Log to console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Channel 1 testing\n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: If	
StepType, Adapter:	NI_Flow_If, <None>
Description:	FileGlobals.ActiveChannels[0] || FileGlobals.ActiveChannels[5] || FileGlobals.ActiveChannels[10] || FileGlobals.ActiveChannels[15] || FileGlobals.ActiveChannels[20] || FileGlobals.ActiveChannels[25] || FileGlobals.ActiveChannels[30]
Comment:	if any of the CHANNEL 1 parts are still active, run this part of the test set.
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Pinmappings	
StepType, Adapter:	Label, <None>
Description:	
Comment:	The channel is mapped as follows: 1 - directly soldered, 2 wire wrap, 3 pcb on top, 4 double ended, 5 solder cups, 6 soldered recepticles, 7 crimped
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Record Results:	Disabled

Step: ensure Channel 1 is off	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetChannelState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Query all the channels	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\GroupSampleAndHoldMeasurement.vi
Comment:	Mass query sample and hold all (active) lines low (slot 5 is connected to channel 1)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - Log to console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Channel 1 low test done\n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Check Channel 1 Directly Soldered	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(15, Locals.MeasuredPoints[0]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[15]

Step: Check Channel 1 Wire Wrap	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(5, Locals.MeasuredPoints[1]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[5]

Step: Check Channel 1 Pcb On Top	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(30, Locals.MeasuredPoints[2]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[30]

Step: Check Channel 1 Double Ended	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(20, Locals.MeasuredPoints[3]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[20]

Step: Check Channel 1 Solder Cups	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(0, Locals.MeasuredPoints[4]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[0]

Step: Check Channel 1 Soldered Receptacles	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(25, Locals.MeasuredPoints[5]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[25]

Step: Check Channel 1 Crimped	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(10, Locals.MeasuredPoints[6]) in <Current File>

Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[10]

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log low voltages	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Comment:	If the channel is active (and passed) it will log the measured voltage, otherwise it will log an X
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Enable Pulsing	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetChannelState.vi
Comment:	Enable Chain 1 pulsing
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Query all the channels	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\GroupSampleAndHoldMeasurement.vi
Comment:	Mass query sample and hold all (active) lines high (slot 5 is connected to channel 1)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - Log to console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Channel 1 high test done\n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Check Channel 1 Directly Soldered	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(15, Locals.MeasuredPoints[0]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[15]

Step: Check Channel 1 Wire Wrap	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(5, Locals.MeasuredPoints[1]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[5]

Step: Check Channel 1 Pcb On Top	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(30, Locals.MeasuredPoints[2]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[30]

Step: Check Channel 1 Double Ended	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(20, Locals.MeasuredPoints[3]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[20]

Step: Check Channel 1 Solder Cups	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(0, Locals.MeasuredPoints[4]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[0]

Step: Check Channel 1 Soldered Receptacles	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(25, Locals.MeasuredPoints[5]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[25]

Step: Check Channel 1 Crimped	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(10, Locals.MeasuredPoints[6]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[10]

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)

Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: ensure Channel 1 is off	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetChannelState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log High voltages	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Comment:	If the channel is active (and passed) it will log the measured voltage, otherwise it will log an X
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: _____Test Chain 2 steps_____	
StepType, Adapter:	Label, <None>
Description:	
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Record Results:	Disabled

Step: GetMessageId - Log to console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Channel 2 testing\n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - Update Active signal chains	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: If	
StepType, Adapter:	NI_Flow_If, <None>
Description:	FileGlobals.ActiveChannels[1] || FileGlobals.ActiveChannels[6] || FileGlobals.ActiveChannels[11] || FileGlobals.ActiveChannels[16] || FileGlobals.ActiveChannels[21] || FileGlobals.ActiveChannels[26] || FileGlobals.ActiveChannels[31]
Comment:	if any of the CHANNEL 2 parts are still active, run this part of the test set.
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Pinmappings	
StepType, Adapter:	Label, <None>
Description:	
Comment:	The channel is mapped as follows: 1 - directly soldered, 2 wire wrap, 3 pcb on top, 4 double ended, 5 solder cups, 6 soldered recepticles, 7 crimped
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Record Results:	Disabled

Step: ensure Channel 2 is off	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetChannelState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Query all the channels	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\GroupSampleAndHoldMeasurement.vi
Comment:	Mass query sample and hold all (active) lines low (slot 7 connected to channel 2)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - Log to console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Channel 2 low test done\n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)

Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Check Channel 2 Directly Soldered	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(16, Locals.MeasuredPoints[0]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[16]

Step: Check Channel 2 Wire Wrap	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(6, Locals.MeasuredPoints[1]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[6]

Step: Check Channel 2 Pcb On Top	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(31, Locals.MeasuredPoints[2]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[31]

Step: Check Channel 2 Double Ended	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(21, Locals.MeasuredPoints[3]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[21]

Step: Check Channel 2 Solder Cups	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(1, Locals.MeasuredPoints[4]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[1]

Step: Check Channel 2 Soldered Receptacles	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(26, Locals.MeasuredPoints[5]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[26]

Step: Check Channel 2 Crimped	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(11, Locals.MeasuredPoints[6]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[11]

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log low voltages	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Comment:	If the channel is active (and passed) it will log the measured voltage, otherwise it will log an X
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Enable Pulsing	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetChannelState.vi
Comment:	Enable Chain 2 pulsing
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Query all the channels	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\GroupSampleAndHoldMeasurement.vi
Comment:	Mass query sample and hold all (active) lines high (slot 7 is on channel 2)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - Log to console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Channel 2 high test done\n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Check Channel 2 Directly Soldered	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(16, Locals.MeasuredPoints[0]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Flow Properties:	Precondition: FileGlobals.ActiveChannels[16]

Step: Check Channel 2 Wire Wrap	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(6, Locals.MeasuredPoints[1]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[6]

Step: Check Channel 2 Pcb On Top	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(31, Locals.MeasuredPoints[2]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[31]

Step: Check Channel 2 Double Ended	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(21, Locals.MeasuredPoints[3]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[21]

Step: Check Channel 2 Solder Cups	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(1, Locals.MeasuredPoints[4]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[1]

Step: Check Channel 2 Soldered Receptacles	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(26, Locals.MeasuredPoints[5]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[26]

Step: Check Channel 2 Crimped	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(11, Locals.MeasuredPoints[6]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[11]

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: ensure Channel 2 is off	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetChannelState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log high voltages	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Comment:	If the channel is active (and passed) it will log the measured voltage, otherwise it will log an X
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: _____Test Chain 3 steps_____ (skipped due to blow board)	
StepType, Adapter:	Label, <None>
Description:	
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Record Results:	Disabled

Step: GetMessageId - Log to console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Channel 3 skipping due to broken board\n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - Update Active signal chains	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "", 0, False)

Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: _____Test Chain 4 steps_____	
StepType, Adapter:	Label, <None>
Description:	
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Record Results:	Disabled

Step: GetMessageId - Log to console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Channel 4 testing\n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - Update Active signal chains	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: If	
StepType, Adapter:	NI_Flow_If, <None>
Description:	FileGlobals.ActiveChannels[3] || FileGlobals.ActiveChannels[8] || FileGlobals.ActiveChannels[13] || FileGlobals.ActiveChannels[18] || FileGlobals.ActiveChannels[23] || FileGlobals.ActiveChannels[28] || FileGlobals.ActiveChannels[33]
Comment:	if any of the CHANNEL 4 parts are still active, run this part of the test set.
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Pinmappings	
StepType, Adapter:	Label, <None>
Description:	
Comment:	The channel is mapped as follows: 1 - directly soldered, 2 wire wrap, 3 pcb on top, 4 double ended, 5 solder cups, 6 soldered recepticles, 7 crimped
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Record Results:	Disabled

Step: ensure Channel 4 is off	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetChannelState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Query all the channels	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\GroupSampleAndHoldMeasurement.vi
Comment:	Mass query sample and hold all (active) lines low (slot 4 connected to channel 4)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - Log to console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Channel 4 low test done\n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Check Channel 4 Directly Soldered	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(18, Locals.MeasuredPoints[0]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[18]

Step: Check Channel 4 Wire Wrap	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(8, Locals.MeasuredPoints[1]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[8]

Step: Check Channel 4 Pcb On Top	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(33, Locals.MeasuredPoints[2]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[33]

Step: Check Channel 4 Double Ended	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(23, Locals.MeasuredPoints[3]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[23]

Step: Check Channel 4 Solder Cups

StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(3, Locals.MeasuredPoints[4]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[3]

Step: Check Channel 4 Soldered Receptacles

StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(28, Locals.MeasuredPoints[5]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[28]

Step: Check Channel 4 Crimped

StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(13, Locals.MeasuredPoints[6]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[13]

Step: Aquire Semaphore

StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log low voltages

StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Comment:	If the channel is active (and passed) it will log the measured voltage, otherwise it will log an X
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore

StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Aquire Semaphore

StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Enable Pulsing

StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetChannelState.vi
Comment:	Enable Chain 4 pulsing
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Query all the channels

StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\GroupSampleAndHoldMeasurement.vi
Comment:	Mass query sample and hold all (active) lines high (slot 4 is on channel 4)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - Log to console

StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage

StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Channel 4 high test done\n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore

StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Check Channel 4 Directly Soldered

StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(18, Locals.MeasuredPoints[0]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[18]

Step: Check Channel 4 Wire Wrap

StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(8, Locals.MeasuredPoints[1]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[8]

Step: Check Channel 4 Pcb On Top

StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(33, Locals.MeasuredPoints[2]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[33]

Step: Check Channel 4 Double Ended

StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(23, Locals.MeasuredPoints[3]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[23]

Step: Check Channel 4 Solder Cups

StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(3, Locals.MeasuredPoints[4]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[3]

Step: Check Channel 4 Soldered Receptacles

StepType, Adapter:	Sequence Call, Sequence

Description:	Call CheckHighPass(28, Locals.MeasuredPoints[5]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[28]

Step: Check Channel 4 Crimped	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(13, Locals.MeasuredPoints[6]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[13]

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: ensure Channel 4 is off	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetChannelState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log high voltages	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Comment:	If the channel is active (and passed) it will log the measured voltage, otherwise it will log an X
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: _____Test Chain 5 steps_____	
StepType, Adapter:	Label, <None>
Description:	
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Record Results:	Disabled

Step: GetMessageId - Log to console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Channel 5 testing\n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - Update Active signal chains	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: If	
StepType, Adapter:	NI_Flow_If, <None>
Description:	FileGlobals.ActiveChannels[4] || FileGlobals.ActiveChannels[9] || FileGlobals.ActiveChannels[14] || FileGlobals.ActiveChannels[19] || FileGlobals.ActiveChannels[24] || FileGlobals.ActiveChannels[29] || FileGlobals.ActiveChannels[34]
Comment:	if any of the CHANNEL 5 parts are still active, run this part of the test set.
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Pinmappings	
StepType, Adapter:	Label, <None>
Description:	
Comment:	The channel is mapped as follows: 1 - directly soldered, 2 wire wrap, 3 pcb on top, 4 double ended, 5 solder cups, 6 soldered recepticles, 7 crimped
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Record Results:	Disabled

Step: ensure Channel 5 is off	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetChannelState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Query all the channels	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\GroupSampleAndHoldMeasurement.vi
Comment:	Mass query sample and hold all (active) lines low (slot 8 connected to channel 5)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - Log to console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Channel 5 low test done\n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Check Channel 5 Directly Soldered	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(19, Locals.MeasuredPoints[0]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[19]

Step: Check Channel 5 Wire Wrap	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(9, Locals.MeasuredPoints[1]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[9]

Step: Check Channel 5 Pcb On Top	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(34, Locals.MeasuredPoints[2]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[34]

Step: Check Channel 5 Double Ended	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(24, Locals.MeasuredPoints[3]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[24]

Step: Check Channel 5 Solder Cups	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(4, Locals.MeasuredPoints[4]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[4]

Step: Check Channel 5 Soldered Receptacles	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(29, Locals.MeasuredPoints[5]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[29]

Step: Check Channel 5 Crimped	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(14, Locals.MeasuredPoints[6]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[14]

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log low voltages	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Comment:	If the channel is active (and passed) it will log the measured voltage, otherwise it will log an X
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Enable Pulsing	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetChannelState.vi
Comment:	Enable Chain 5 pulsing
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Query all the channels	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\GroupSampleAndHoldMeasurement.vi
Comment:	Mass query sample and hold all (active) lines high (slot 8 is on channel 5)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - Log to console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM

Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Channel 5 high test done\n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Check Channel 5 Directly Soldered	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(19, Locals.MeasuredPoints[0]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[19]

Step: Check Channel 5 Wire Wrap	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(9, Locals.MeasuredPoints[1]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[9]

Step: Check Channel 5 Pcb On Top	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(34, Locals.MeasuredPoints[2]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[34]

Step: Check Channel 5 Double Ended	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(24, Locals.MeasuredPoints[3]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[24]

Step: Check Channel 5 Solder Cups	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(4, Locals.MeasuredPoints[4]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[4]

Step: Check Channel 5 Soldered Receptacles	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(29, Locals.MeasuredPoints[5]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[29]

Step: Check Channel 5 Crimped	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(14, Locals.MeasuredPoints[6]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[14]

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: ensure Channel 5 is off	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetChannelState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log high voltages	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Comment:	If the channel is active (and passed) it will log the measured voltage, otherwise it will log an X
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: _____Test Chain 6 steps_____	
StepType, Adapter:	Label, <None>
Description:	
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Record Results:	Disabled

Step: GetMessageId - Log to console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage

StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Channel 6 testing\n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - Update Active signal chains	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: If	
StepType, Adapter:	NI_Flow_If, <None>
Description:	FileGlobals.ActiveChannels[35] || FileGlobals.ActiveChannels[36] || FileGlobals.ActiveChannels[37] || FileGlobals.ActiveChannels[38] || FileGlobals.ActiveChannels[39]
Comment:	if any of the CHANNEL 6 parts are still active, run this part of the test set.
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Pinmappings	
StepType, Adapter:	Label, <None>
Description:	
Comment:	The channel is mapped as follows: 1 to 5 = P101 to P105 chains on the Double PCB mount system
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Record Results:	Disabled

Step: ensure Channel 6 is off	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetChannelState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Query all the channels	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\GroupSampleAndHoldMeasurement.vi
Comment:	Mass query sample and hold all (active) lines low (slot 3 connected to channel 6)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - Log to console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Channel 6 low test done\n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Check Channel 6 P101	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(35, Locals.MeasuredPoints[0]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[35]

Step: Check Channel 6 P102	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(36, Locals.MeasuredPoints[1]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[36]

Step: Check Channel 6 P104	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(38, Locals.MeasuredPoints[2]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[34]

Step: Check Channel 6 P105	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckLowPass(39, Locals.MeasuredPoints[3]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[39]

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log low voltages	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Comment:	If the channel is active (and passed) it will log the measured voltage, otherwise it will log an X
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Enable Pulsing	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetChannelState.vi
Comment:	Enable Chain 6 pulsing
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Query all the channels	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\GroupSampleAndHoldMeasurement.vi
Comment:	Mass query sample and hold all (active) lines low (slot 3 connected to channel 6)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - Log to console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Channel 6 high test done\n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Check Channel 6 P101	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(35, Locals.MeasuredPoints[0]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[35]

Step: Check Channel 6 P102	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(36, Locals.MeasuredPoints[1]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[36]

Step: Check Channel 6 P104	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(38, Locals.MeasuredPoints[2]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[34]

Step: Check Channel 6 P105	
StepType, Adapter:	Sequence Call, Sequence
Description:	Call CheckHighPass(39, Locals.MeasuredPoints[3]) in <Current File>
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Flow Properties:	Precondition: FileGlobals.ActiveChannels[39]

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: ensure Channel 6 is off	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetChannelState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log High voltages	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Comment:	If the channel is active (and passed) it will log the measured voltage, otherwise it will log an X
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Check if there are any active channels left	
StepType, Adapter:	Statement, <None>
Description:	FileGlobals.TestingActive = FileGlobals.ActiveChannels[0] ||FileGlobals.ActiveChannels[1]||FileGlobals.ActiveChannels[2]||FileGlobals.ActiveChannels[3]||FileGlobals.ActiveChannels[4]||FileGlobals.ActiveChannels[5]||FileGlobals.ActiveChan
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Wait	
StepType, Adapter:	Wait, <None>

Description:	TimeInterval(5)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - check for stop	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Cleanup

Step: wait for other threads to end before stopping	
StepType, Adapter:	Label, <None>
Description:	
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Record Results:	Disabled

Step: While	
StepType, Adapter:	NI_Flow_While, <None>
Description:	FileGlobals.pollTemperatureThreadActive || FileGlobals.heartbeatThreadActive
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Message Popup	
StepType, Adapter:	Message Popup, <None>
Description:	NameOf(Step)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Wait	
StepType, Adapter:	Wait, <None>
Description:	TimeInterval(3)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

---

Sequence: Poll Temperatures	
Type:	Normal
Disable Results for All Steps:	False
Goto Cleanup on failure:	Use Station Option
Number of Setup steps:	6
Number of Main steps:	49
Number of Cleanup steps:	6

Locals:		
AmbientAverage	Number	0
Sensor1Average	Number	0
Sensor2Average	Number	0
Sensor3Average	Number	0
Sensor4Average	Number	0
Sensor5Average	Number	0
Sensor6Average	Number	0
Sensor7Average	Number	0
iteration	Number	0
AmbientPoints	Array of Numbers[0..4]	...
Sensor1Points	Array of Numbers[0..4]	...
Sensor2Points	Array of Numbers[0..4]	...
Sensor3Points	Array of Numbers[0..4]	...
Sensor4Points	Array of Numbers[0..4]	...
Sensor5Points	Array of Numbers[0..4]	...
Sensor6Points	Array of Numbers[0..4]	...
Sensor7Points	Array of Numbers[0..4]	...
TestPointsUpAndDown	Number	100
DataPointCount	Number	0
GoingUP	Boolean	False
MsgIdToSend	Number	0
ResultList	Array of Result[0..empty]	...

Setup

Step: set thread state active	
StepType, Adapter:	Statement, <None>
Description:	FileGlobals.pollTemperatureThreadActive = True
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Turn off Vent Fan	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetVentFanState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Cooler off	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetCoolerState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Heater off	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetHeaterState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore

StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Main

Step: While	
StepType, Adapter:	NI_Flow_While, <None>
Description:	FileGlobals.TestingActive
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: For	
StepType, Adapter:	NI_Flow_For, <None>
Description:	Locals.iteration = 0; Locals.iteration < 5; Locals.iteration += 1
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: GetMessageId - Update Results	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, ".", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Read the temperatures	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\ReadTemperatures.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Wait 1 second	
StepType, Adapter:	Wait, <None>
Description:	TimeInterval(1)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Calculate Ambient Average	
StepType, Adapter:	Statement, <None>
Description:	Locals.AmbientAverage = (Locals.AmbientPoints[0] + Locals.AmbientPoints[1] + Locals.AmbientPoints[2] + Locals.AmbientPoints[3] + Locals.AmbientPoints[4])/5
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Calculate Sensor 1 Average	
StepType, Adapter:	Statement, <None>
Description:	Locals.Sensor1Average = (Locals.Sensor1Points[0] + Locals.Sensor1Points[1]+ Locals.Sensor1Points[2] + Locals.Sensor1Points[3] + Locals.Sensor1Points[4])/5
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Calculate Sensor 2 Average	
StepType, Adapter:	Statement, <None>
Description:	Locals.Sensor2Average = (Locals.Sensor2Points[0] + Locals.Sensor2Points[1]+ Locals.Sensor2Points[2] + Locals.Sensor2Points[3] + Locals.Sensor2Points[4])/5
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Calculate Sensor 3 Average	
StepType, Adapter:	Statement, <None>
Description:	Locals.Sensor3Average = (Locals.Sensor3Points[0] + Locals.Sensor3Points[1]+ Locals.Sensor3Points[2] + Locals.Sensor3Points[3] + Locals.Sensor3Points[4])/5
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Calculate Sensor 4 Average	
StepType, Adapter:	Statement, <None>
Description:	Locals.Sensor4Average = (Locals.Sensor4Points[0] + Locals.Sensor4Points[1]+ Locals.Sensor4Points[2] + Locals.Sensor4Points[3] + Locals.Sensor4Points[4])/5
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Calculate Sensor 5 Average	
StepType, Adapter:	Statement, <None>
Description:	Locals.Sensor5Average = (Locals.Sensor5Points[0] + Locals.Sensor5Points[1]+ Locals.Sensor5Points[2] + Locals.Sensor5Points[3] + Locals.Sensor5Points[4])/5
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Calculate Sensor 6 Average	
StepType, Adapter:	Statement, <None>
Description:	Locals.Sensor6Average = (Locals.Sensor6Points[0] + Locals.Sensor6Points[1]+ Locals.Sensor6Points[2] + Locals.Sensor6Points[3] + Locals.Sensor6Points[4])/5
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Calculate Sensor 7 Average	
StepType, Adapter:	Statement, <None>
Description:	Locals.Sensor7Average = (Locals.Sensor7Points[0] + Locals.Sensor7Points[1]+ Locals.Sensor7Points[2] + Locals.Sensor7Points[3] + Locals.Sensor7Points[4])/5
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - Update Results	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM

Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Temp:" + Str(Locals.AmbientAverage) + ", "+ Str(Locals.Sensor1Average) + ", "+ Str(Locals.Sensor2Average) + ", "+ Str(Locals.Sensor3Average) + ", "+ Str(Locals.Sensor4Average) + ", "+ Str(Locals.Sensor5Average)+ ", "+ Str(Locals.Sensor6Average) + ", " + Str(Locals.Sensor7Average)+ " \n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log Temperatures to File	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Since this tests on a 7 secon interval (in its own thread), just let it log 100 up and 100 down and change direction accordingly	
StepType, Adapter:	Label, <None>
Description:	
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed
Record Results:	Disabled

Step: If	
StepType, Adapter:	NI_Flow_If, <None>
Description:	Locals.GoingUP
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Cooler off	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetCoolerState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Heater on	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetHeaterState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Turn off Vent Fan	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetVentFanState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log Going Up to file	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Else	
StepType, Adapter:	NI_Flow_Else, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Heater off	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetHeaterState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Cooler on	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetCoolerState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Turn on Vent Fan	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetVentFanState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>

Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log Going Down to file	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Check for up or down on temperature curve	
StepType, Adapter:	Statement, <None>
Description:	Locals.DataPointCount = ((Locals.DataPointCount + 1) == Locals.TestPointsUpAndDown)? (0):((Locals.DataPointCount + 1))
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: If	
StepType, Adapter:	NI_Flow_If, <None>
Description:	Locals.DataPointCount == 0
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Change Chamber Direction	
StepType, Adapter:	Statement, <None>
Description:	Locals.GoingUP=!(Locals.GoingUP)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: GetMessageId - update Console	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "Temperature Chamber going " + ((Locals.GoingUP)?("UP"):("DOWN"))+ "\n", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Cleanup

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Turn off Vent Fan	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetVentFanState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Cooler off	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetCoolerState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Heater off	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetHeaterState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: set thread state inActive	
StepType, Adapter:	Statement, <None>
Description:	FileGlobals.pollTemperatureThreadActive = False
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Sequence: Heartbeat	
Type:	Normal
Disable Results for All Steps:	False
Goto Cleanup on failure:	Use Station Option
Number of Setup steps:	1
Number of Main steps:	7
Number of Cleanup steps:	1

Locals:		
MsgIdToSend	Number	0
ResultList	Array of Result[0..empty]	...

Setup

Step: set thread state active	
StepType, Adapter:	Statement, <None>
Description:	FileGlobals.heartbeatThreadActive = True
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Main

Step: While	
StepType, Adapter:	NI_Flow_While, <None>
Description:	FileGlobals.TestingActive
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: GetMessageId - Heartbeat	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\convertMsgIdAndNumber.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 1, "", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Wait a second	
StepType, Adapter:	Wait, <None>
Description:	TimeInterval(1)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: SendMessage	
StepType, Adapter:	Action, ActiveX/COM
Description:	Action, Call Thread{RunState.Thread}.PostUIMessageEx (Locals.MsgIdToSend, 0, "", 0, False)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Wait a second	
StepType, Adapter:	Wait, <None>
Description:	TimeInterval(1)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Cleanup

Step: set thread state inActive	
StepType, Adapter:	Statement, <None>
Description:	FileGlobals.heartbeatThreadActive = False
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

---

Sequence: CheckLowPass	
Type:	Normal
Disable Results for All Steps:	False
Goto Cleanup on failure:	Use Station Option
Number of Setup steps:	0
Number of Main steps:	21
Number of Cleanup steps:	0

Parameters:		
channelNumberInDialog	Number (by reference)	0
VoltageMeasured	Number (by reference)	0

Locals:		
selectResult	Number	0
ResultList	Array of Result[0..empty]	...

Main

Step: If	
StepType, Adapter:	NI_Flow_If, <None>
Description:	Parameters.VoltageMeasured > FileGlobals.LowPassPoint
Comment:	If the voltage is too high, its a failure event.
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log Event	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: TestDialogBoxFailEvent	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\DialogBoxFailure.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Select	
StepType, Adapter:	NI_Flow_Select, <None>
Description:	Locals.selectResult
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Case	
StepType, Adapter:	NI_Flow_Case, <None>
Description:	1
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: disable the channel	
StepType, Adapter:	Statement, <None>
Description:	FileGlobals.ActiveChannels[Parameters.channelNumberInDialog] = False
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	

Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Case	
StepType, Adapter:	NI_Flow_Case, <None>
Description:	2
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log Probe Failure Event	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Case	
StepType, Adapter:	NI_Flow_Case, <None>
Description:	3
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log UUT Failure Event	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

---

Sequence: CheckHighPass	
Type:	Normal
Disable Results for All Steps:	False
Goto Cleanup on failure:	Use Station Option
Number of Setup steps:	0
Number of Main steps:	21
Number of Cleanup steps:	0

Parameters:		
channelNumberInDialog	Number (by reference)	0
VoltageMeasured	Number (by reference)	0

Locals:		
selectResult	Number	0
ResultList	Array of Result[0..empty]	...

Main

Step: If	
StepType, Adapter:	NI_Flow_If, <None>
Description:	(Parameters.VoltageMeasured > FileGlobals.HighPassCeilig)||(Parameters.VoltageMeasured <FileGlobals.HighPassFloor)
Comment:	If the voltage is too high, its a failure event.
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log Event	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: TestDialogBoxFailEvent	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\DialogBoxFailure.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Select	
StepType, Adapter:	NI_Flow_Select, <None>
Description:	Locals.selectResult
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Case	
StepType, Adapter:	NI_Flow_Case, <None>
Description:	1
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: disable the channel	
StepType, Adapter:	Statement, <None>
Description:	FileGlobals.ActiveChannels[Parameters.channelNumberInDialog] = False
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Case	
StepType, Adapter:	NI_Flow_Case, <None>
Description:	2
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log Probe Failure Event	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Case	
StepType, Adapter:	NI_Flow_Case, <None>
Description:	3
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Log UUT Failure Event	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\WriteLogToFile.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.LogFileSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

---

Sequence: Exercise	
Type:	Normal
Disable Results for All Steps:	False
Goto Cleanup on failure:	Use Station Option
Number of Setup steps:	0
Number of Main steps:	10
Number of Cleanup steps:	0

Locals:		
cycles	Number	0
ResultList	Array of Result[0..empty]	...

Main

Step: For	
StepType, Adapter:	NI_Flow_For, <None>
Description:	Locals.cycles = 0; Locals.cycles < 8; Locals.cycles += 1
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Actuate Low (ensure you start in a low state)

StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetActuatorState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Wait	
StepType, Adapter:	Wait, <None>
Description:	TimeInterval(1)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Aquire Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Acquire(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Actuate High	
StepType, Adapter:	Action, LabVIEW
Description:	Action, builds\console\My Source Distribution.llb\SetActuatorState.vi
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Release Semaphore	
StepType, Adapter:	Semaphore, <None>
Description:	Release(FileGlobals.ComPortSemaphore)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: Wait	
StepType, Adapter:	Wait, <None>
Description:	TimeInterval(1)
Module Load and Unload Options:	Preload when execution begins, Unload when sequence file is closed

Step: End	
StepType, Adapter:	NI_Flow_End, <None>
Description:	
Module Load and Unload Options:	Load dynamically, Unload when sequence file is closed

ANNEXURE D

SEQUENCE FLOW CHARTS

The relative reliability tester software consists of both Labview Vis to implement the communication between the TDA019D Daq system, as well as a teststand sequence used to control the automated testing process.

The following flow charts describe the 6 sequences used against the teststand engine to control the relative reliabilility tester.

D.1 Flow Charts.

See following pages for the flow charts of the testsand sequences in the following order:

1. Main Sequence Flow Chart
2. Temperature Chamber Control Thread
3. Heartbeat Thread
4. Exercise sequence (Pneumatic Stress Test)
5. High state test sequences
6. Low state test sequence

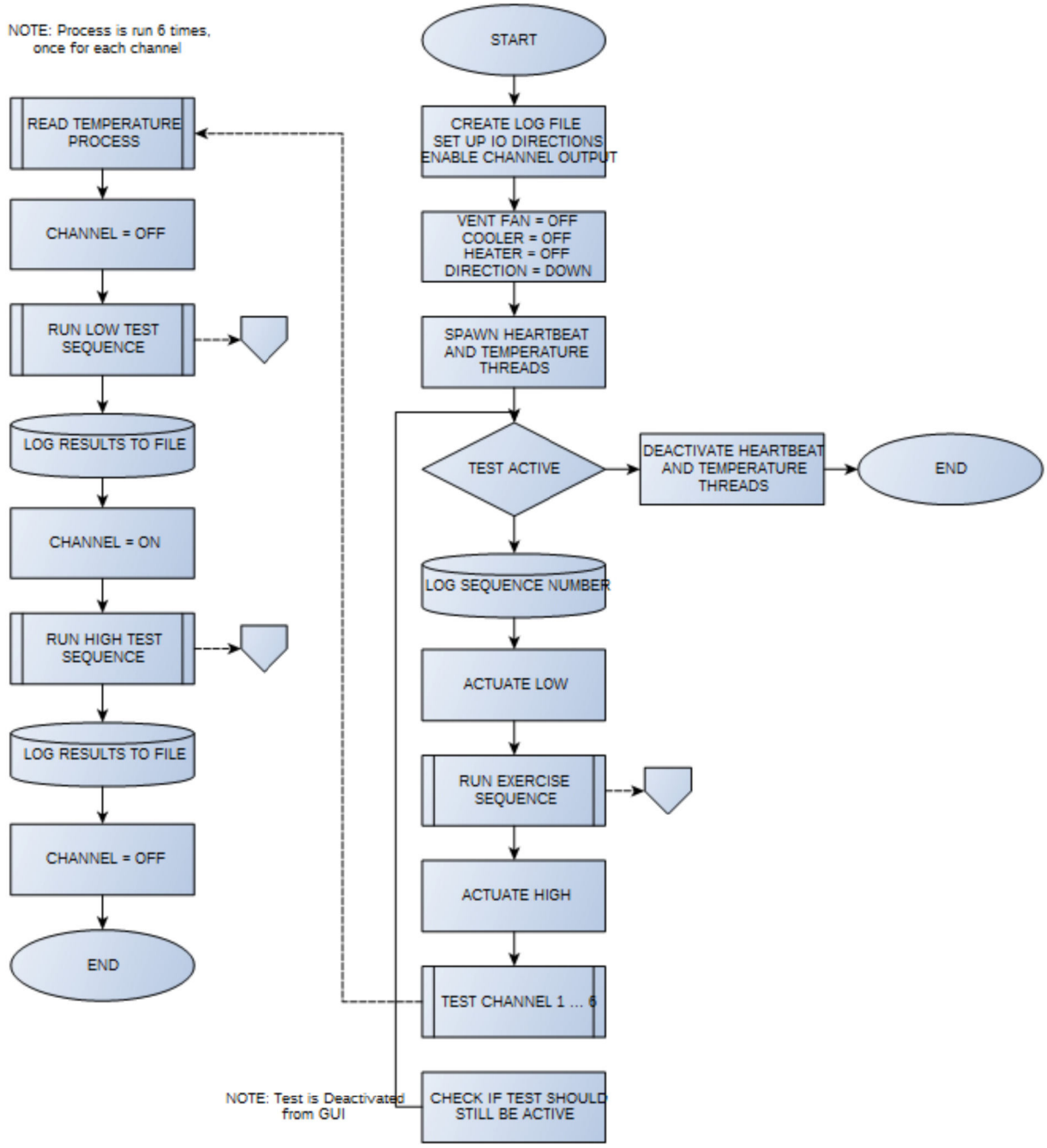


Illustration 1: Main Sequence Flow Chart

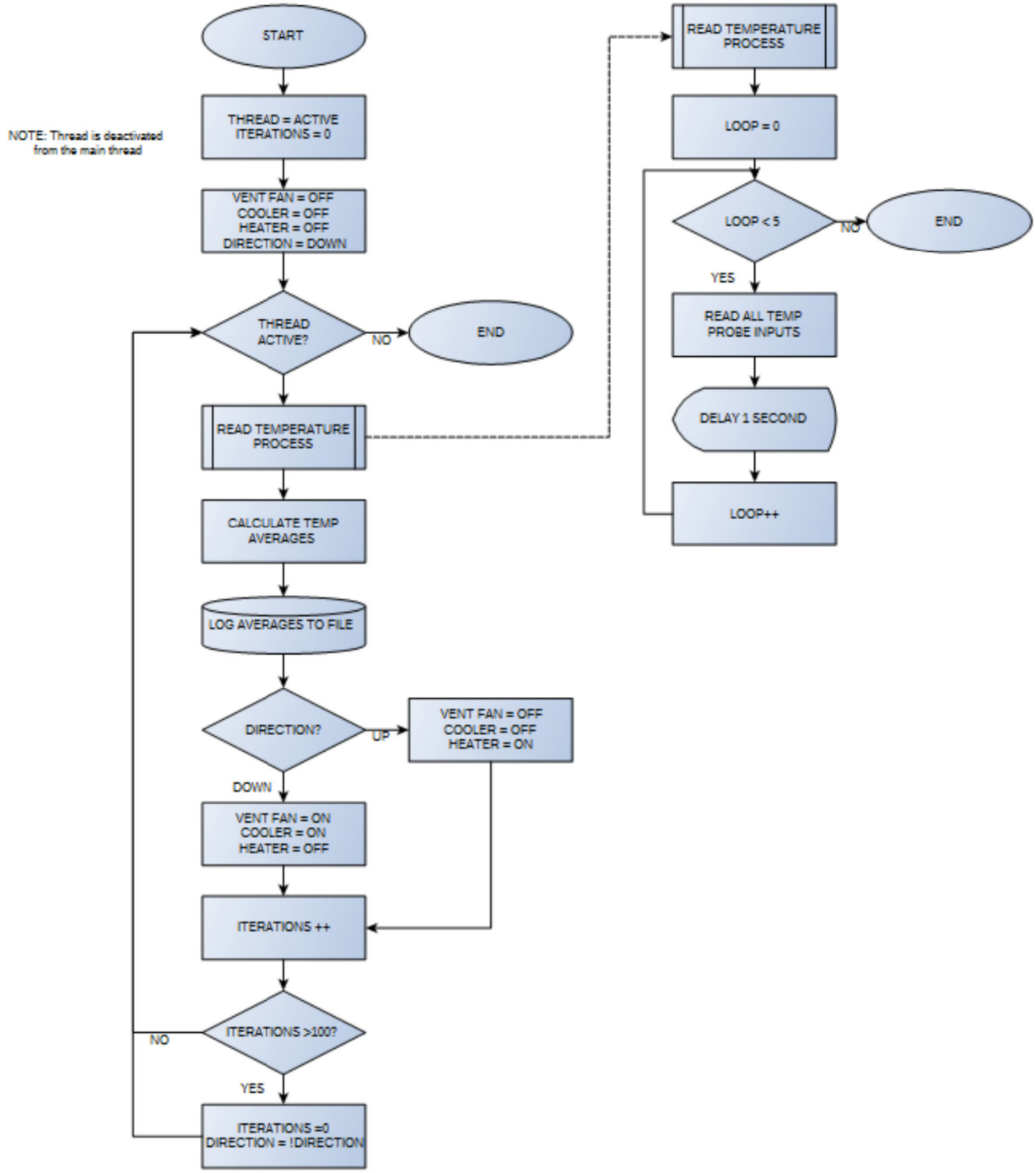


Illustration 2: Temperature chamber control thread sequence

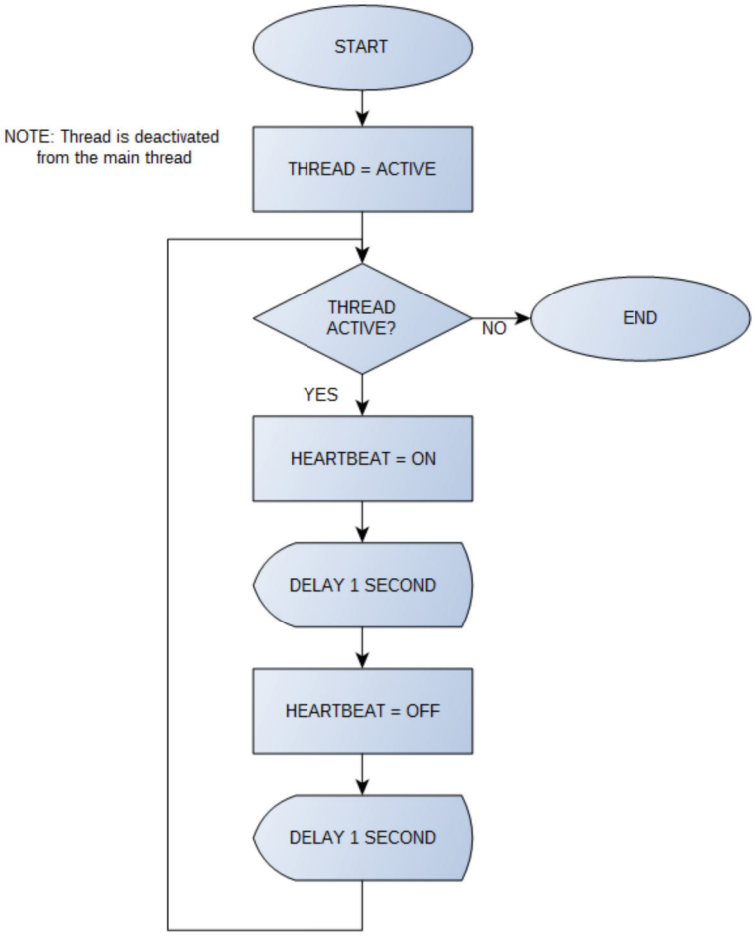


Illustration 3: Heartbeat thread sequence flow chart

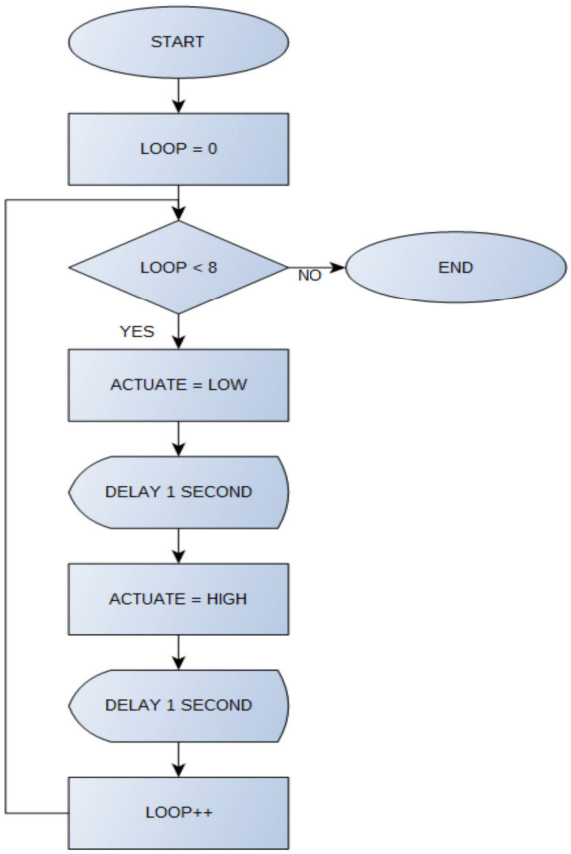


Illustration 4: Exercise (stress test) sequence

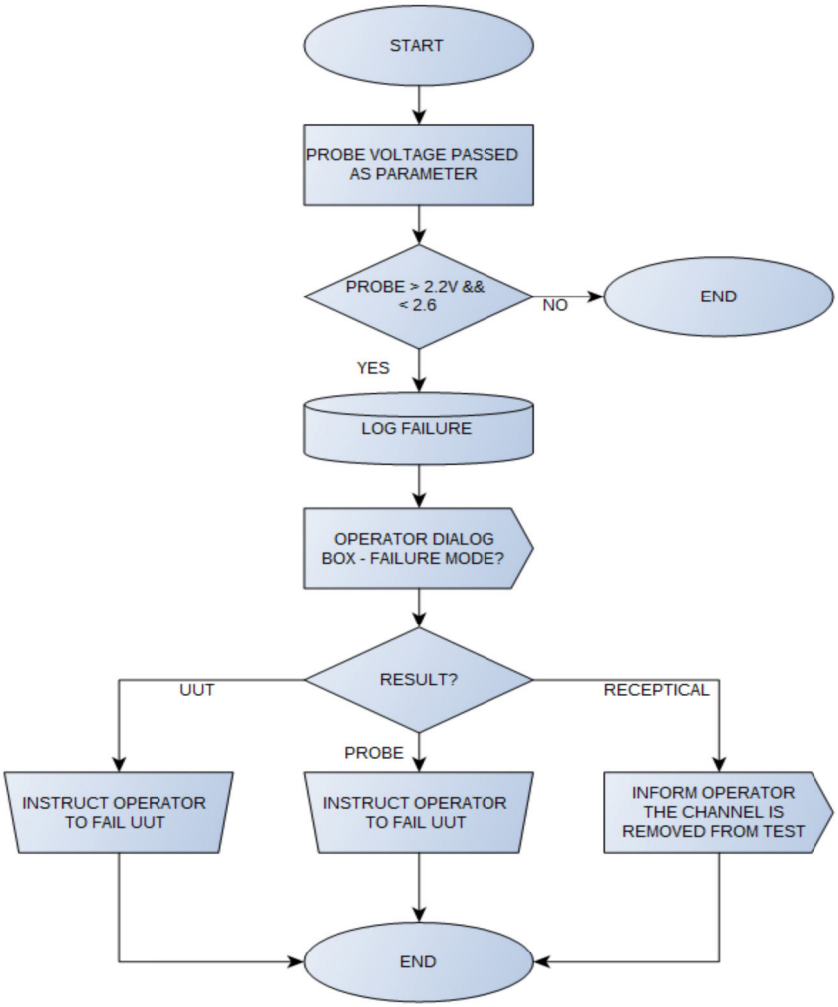


Illustration 5: High test sequence

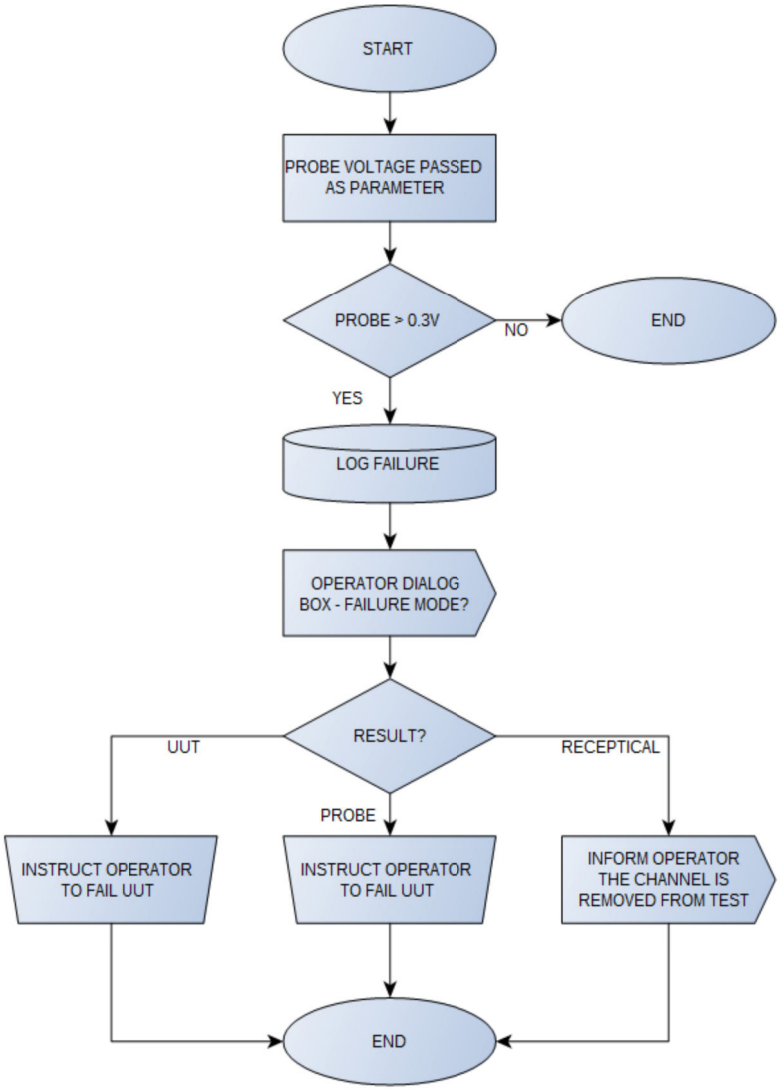


Illustration 6: Low test sequence