

- DIFFERENTIAL CONDUCTANCE -
- MEASUREMENT APPARATUS -
- TECHNICAL MANUAL -

DOCUMENT REVISION 1.0
JULY 17, 2012

SOFTWARE VERSION 3.9
MAY 10, 2012

MOTIVATION

When applying a voltage to a general quantum mechanical system, specifically, a planar junction, the current through the system is always a function of the voltage input. Knowledge of this function tells us about the system itself, such as the density of states on either side of the junction. Here, an apparatus is constructed to automatically measure this function.

DOCUMENT REVISION HISTORY

<i>Rev.</i>	<i>Date (YMD)</i>	<i>Description</i>	<i>Author</i>	<i>Permanent Email</i>
1.0	2012-07-17	Initial Creation	Frank Zhao	frankzhao47@hotmail.com

Differential Conductance Measurement Apparatus Technical Manual

TABLE OF CONTENTS

Section	Page
I. SYSTEM OVERVIEW	5
I.1 – Introduction	5
I.2 – Overview	5
I.3 – Operating Principles	6
II. INSTRUMENTS AND THE BLUE BOX	7
II.1 – Motivation	7
II.2 – Overall Schematic	7
II.3 – The Blue Box	9
III. SOFTWARE OVERVIEW	12
III.1 – Motivation	12
III.2 – Control Program Features and Functionality	12
III.2.A – Instrumentation Functionality and Feature List	12
III.2.B – Measurement Functionality and Feature List	13
III.2.C – Data Handling	13
III.2.D – Automation	13
III.3 – Measurement Sequence	14
III.4 – Main Program User Interface	15
III.4.A – Communications Settings Panel	16
III.4.B – Instrument Configuration Panel	17
III.4.C – Advanced Measurement Configuration Panel	18
III.4.D – User Commands Panel	19
III.4.E – Measurement Progress Status Display Panel	20
III.4.F – Temperature Control Panel	21
III.4.G – Cycle Parameters Panel	22
III.4.H – Measurement Stabilization Panel	23
III.4.I – File Management Panel	24
III.4.J – Current Status Panel	25
III.4.K – Measurement indicator Display Panel	26
III.4.L – Stabilization Display	27
III.4.M – Data Plots	28
III.4.N – Queue	29
IV. SOFTWARE CODE AND ALGORITHMS	30
IV.1 – Motivation	30
IV.2 – General Program Layout	30
IV.3 – Measurement	32
IV.3.A – Load Queue Parameters	32
IV.3.B – Calculate Supply Voltages	34
IV.3.C – Stabilize Temperature	35
IV.3.D – Stabilize Magnetic Field	36
IV.3.E – Generate Filenames	37
IV.3.F – Generate Header File	38
IV.3.G – Configure LIA	39
Section	Page

<i>IV.3.H – Set DC Voltage</i>	<i>40</i>
<i>IV.3.I – Wait</i>	<i>40</i>
<i>IV.3.J – Auto-Gain and Auto-Stabilize</i>	<i>41</i>
<i>IV.3.K – Measure</i>	<i>42</i>
<i>IV.4 – List of SubVI</i>	<i>43</i>
<i>IV.4.A – SubVI Conventions</i>	<i>43</i>
<i>IV.4.B – Magnet Control SubVI</i>	<i>44</i>
<i>IV.4.C – Temperature Controller SubVI</i>	<i>45</i>
<i>IV.4.D – Agilent 34401A Multimeter Customized SubVI</i>	<i>46</i>
<i>IV.4.E – HP3457A Digital Multimeter SubVI</i>	<i>47</i>
<i>IV.4.F – SR810 Lock-In Amplifier SubVI</i>	<i>48</i>
<i>IV.4.G – Queue SubVI</i>	<i>49</i>
<i>IV.4.H – File-Handling SubVI</i>	<i>49</i>
<i>IV.4.I – General Programming SubVI</i>	<i>50</i>
<i>V. APPENDIX: COMMUNICATING WITH INSTRUMENTS</i>	<i>51</i>
<i>V.1 – NI VISA</i>	<i>52</i>
<i>V.2 – Connections in Specific Protocols</i>	<i>52</i>
<i>V.2.A – RS-232 Serial Ports</i>	<i>52</i>
<i>V.2.B – GPIB Ports</i>	<i>53</i>
<i>V.2.C – Ethernet Ports</i>	<i>53</i>
<i>V.3 – General Procedure for Connecting to Devices</i>	<i>54</i>

I. SYSTEM OVERVIEW

I.1 – INTRODUCTION

The behavior of electrons in materials can be directly studied by measuring the material's response to an applied voltage. For conventional conductors, such as metals, the current passing through it is directly proportional to the voltage difference applied, by Ohm's Law: The proportionality constant is the conductance of the material. Here, we can see that the conductance of a material is simply the derivative in the I-V plot:

When the material in question is no longer a conventional conductor, Ohm's Law breaks down. However, some general relationship between applied voltage and current still exists, as some general function. We can then generalize the conductance of the sample to be the derivative above, except now with the caveat that conductance is now itself a function of the voltage applied:

Knowledge of the conductance curve, as a function of applied voltage, provides much knowledge about the sample itself. In order to measure this quantity with relative ease, the Differential Conductance Measurement Apparatus was put together between Summer 2011 and 2012.

I.2 – OVERVIEW

The Differential Conductance Measurement Apparatus comprises of a set of hardware instruments, linked together via a connector box (the "Blue Box"), all controlled via a LabVIEW Script on a PC computer.

On the hardware side, The system uses the instruments:

- Stanford Instruments SR810 Lock-in Amplifiers (x2)
- Agilent 34401A Digital Multimeter
- HP3457A Digital Multimeter
- Stanford Instruments SR570 Low-Noise Current Preamp
- BK Precision 1787B Single Channel Power Supply
- LakeShore 331 Temperature Controller
- American Magnetics Model 430 Power Supply (for MARTI)

The LabVIEW Script on the PC communicates with all these instruments, coordinating their actions and recording the measurements.

I.3 – OPERATING PRINCIPLES

The system applies, to the sample, an oscillatory signal superimposed on a constant voltage offset:

First, note that the constants V_0 and V_1 are both measurable using our instruments. We can now express the voltage as a small variation around a mean:

At the same time, the current going across the sample is some function of the applied voltage:

from which we can do the Taylor expansion:

Note now, given the form of the applied voltage $V(t)$, and assuming that the amplitude in voltage is small enough so that V_0 is constant in the range over which $V(t)$ oscillates, then we can assume that the current has the form:

Now, matching terms from the Taylor expansion and the expected form of $I(t)$, we see that:

and that:

The first equation gives us the I-V curve. From here, if we simultaneously measured the time-averaged voltage V_0 and the corresponding (constant, ie. time-averaged) current I_0 , we would have known the full I-V profile of the sample. In practice, these are measured using a pair of DC voltmeters and a *Current to Voltage Converter (SR570)*

The second equation tells us that the derivative dI/dV has to do with the applied sinusoidal part of the voltage. Using the fact that dV is a deviation from the mean value, we have $dV = V_1 \sin(\omega t)$, from which we can divide dI from both sides of the equation above, and it follows that:

Indeed, it is easy to convince oneself that this is completely intuitive. But since the amplitudes A and B are measurable from the Lock-In Amplifiers, the differential conductance dI/dV can easily be measured.

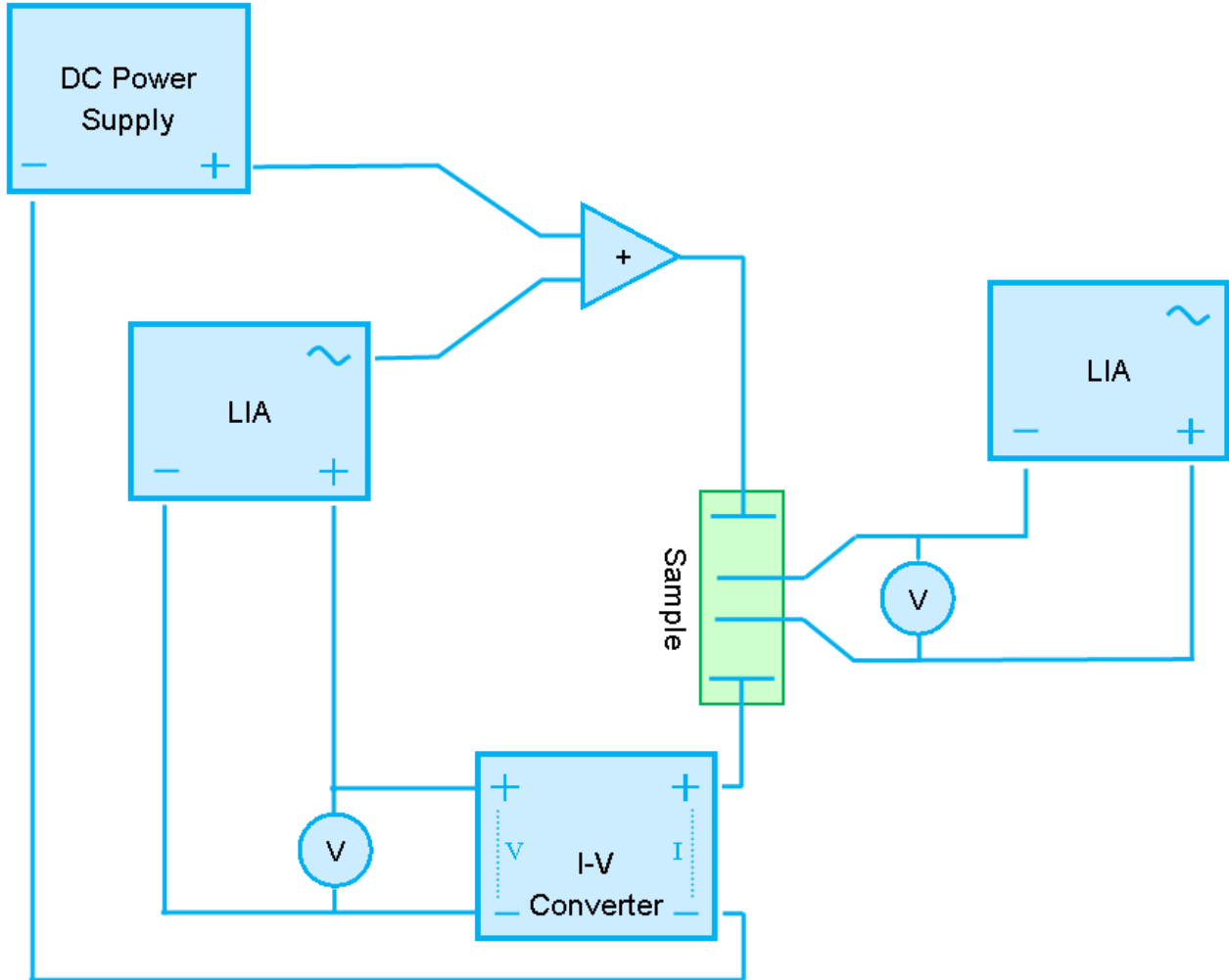
II. INSTRUMENTS AND THE BLUE BOX

II.1 – MOTIVATION

Here, we explain the fundamental roles that each instrument plays, and the wiring topology in the “Blue Box”.

II.2 – OVERALL SCHEMATIC

Overall the system is laid out¹:



Here, the *DC Power Supply* (*BK Precision 1787B*) applies a voltage, which is immediately added to the AC voltage signal applied by the *Lock-In Amplifier* (*LIA – Stanford Instruments SR810*). The adding is accomplished by a DC-biased transformer, inside the *Blue Box*, but for simplicity, for now, this is drawn as the triangular operational amplifier adder. The combined signal then travels through the sample, which is connected in a 4-point configuration.

At the sample, a pair of inner *Voltage Leads* senses the voltage drop across the functional part of the sample, where a *DC voltmeters* (*Agilent 34401A*) measures the DC part of the voltage

¹ Much credit goes to Achint Jain, at the time of writing at *ETH Zurich*, who worked out the system of measurement using two Lock-In Amplifiers.

drop, and a *Lock-In Amplifier (LIA – SR810)* measures the AC part of the voltage drop, at the same frequency as the applied signal.

At the same time, in the sample, the current is allowed to travel from the adder at Pin 1, through the sample, to the *Current-Voltage Preamp (Stanford Instruments SR570)* at Pin 4. The Current-Voltage Preamp effectively converts the current flowing through the input (drawn on the right hand side of the instrument) to a voltage. In reality, the two is related by a proportionality constant, set on the instrument. The voltage output is measured by a *DC Voltmeter (HP 3457A Digital Multimeter)* and the other *Lock-In Amplifier (SR710)*. To complete the circuit, the negative pin on the I-V converter goes to the negative terminal on the DC power supply.

This is the simplest instrument configuration. In actual use, several additions are usually made.

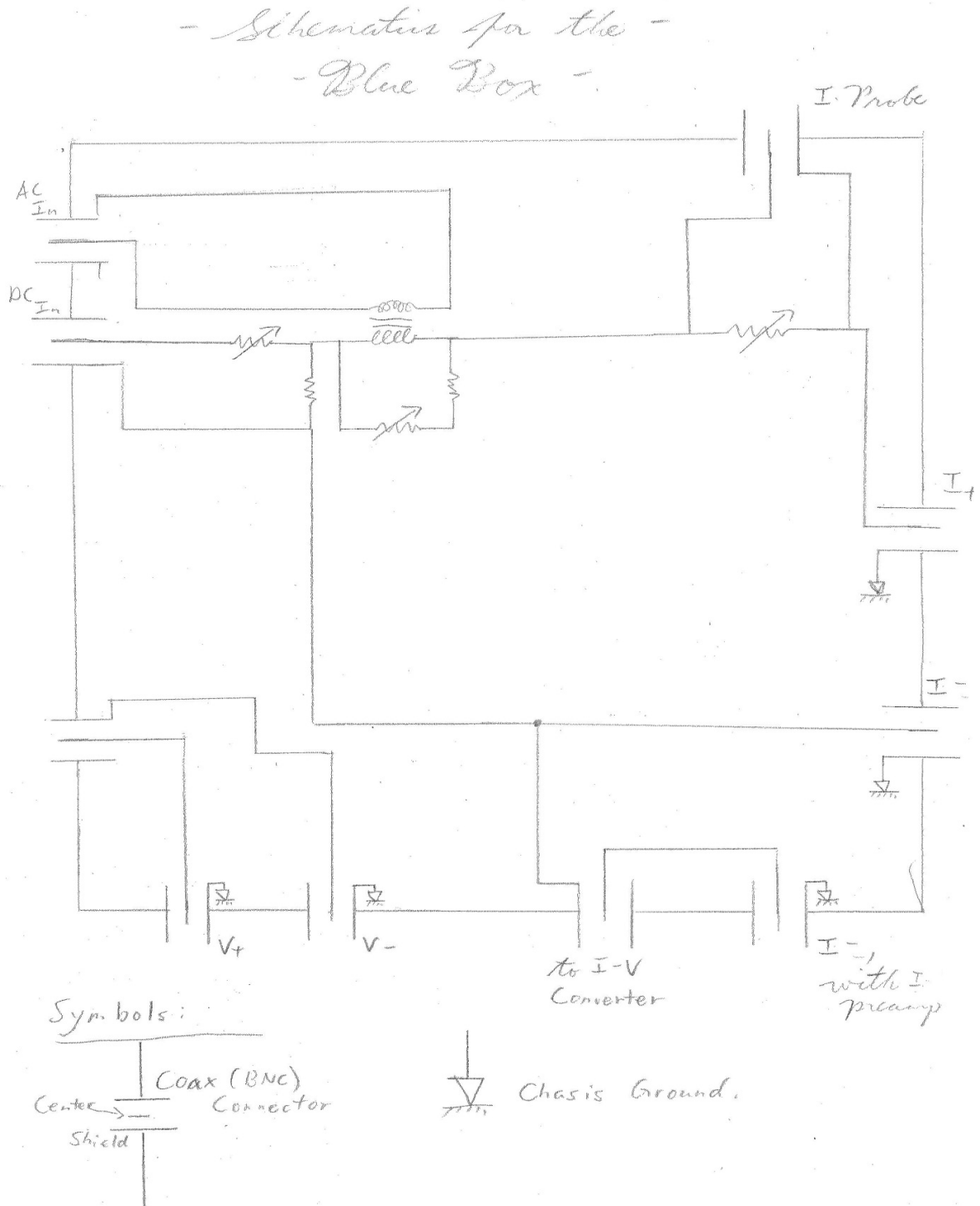
First, a *low-pass filter* is usually added before the DC voltmeters. This ensures that the signal going into the DC voltmeters are optimally filtered for noise, and so will not contain any oscillations from the AC signal.

Second, instead of the *I-V Converter*, a resistor, connected in series with the sample, between the sample and the adder, may be used to convert the current signal to a voltage. Alternatively, this resistor can be used to protect the sample from excessive heating.

Differential Conductance Measurement Apparatus Technical Manual

II.3 – THE BLUE BOX

The Blue Box acts as a custom connector box that connects all the instruments together. Schematically, it looks like:



The box has several functional purposes. First, it is a convenient way to *connect wires together in a way that is useful for the measurement*. Secondly, because the DC voltage source is designed to output high voltages up to 72V, with low accuracy, *the AC and DC signals has to be divided using voltage dividers*. Thirdly, the AC and DC signals needed to be *added together*.

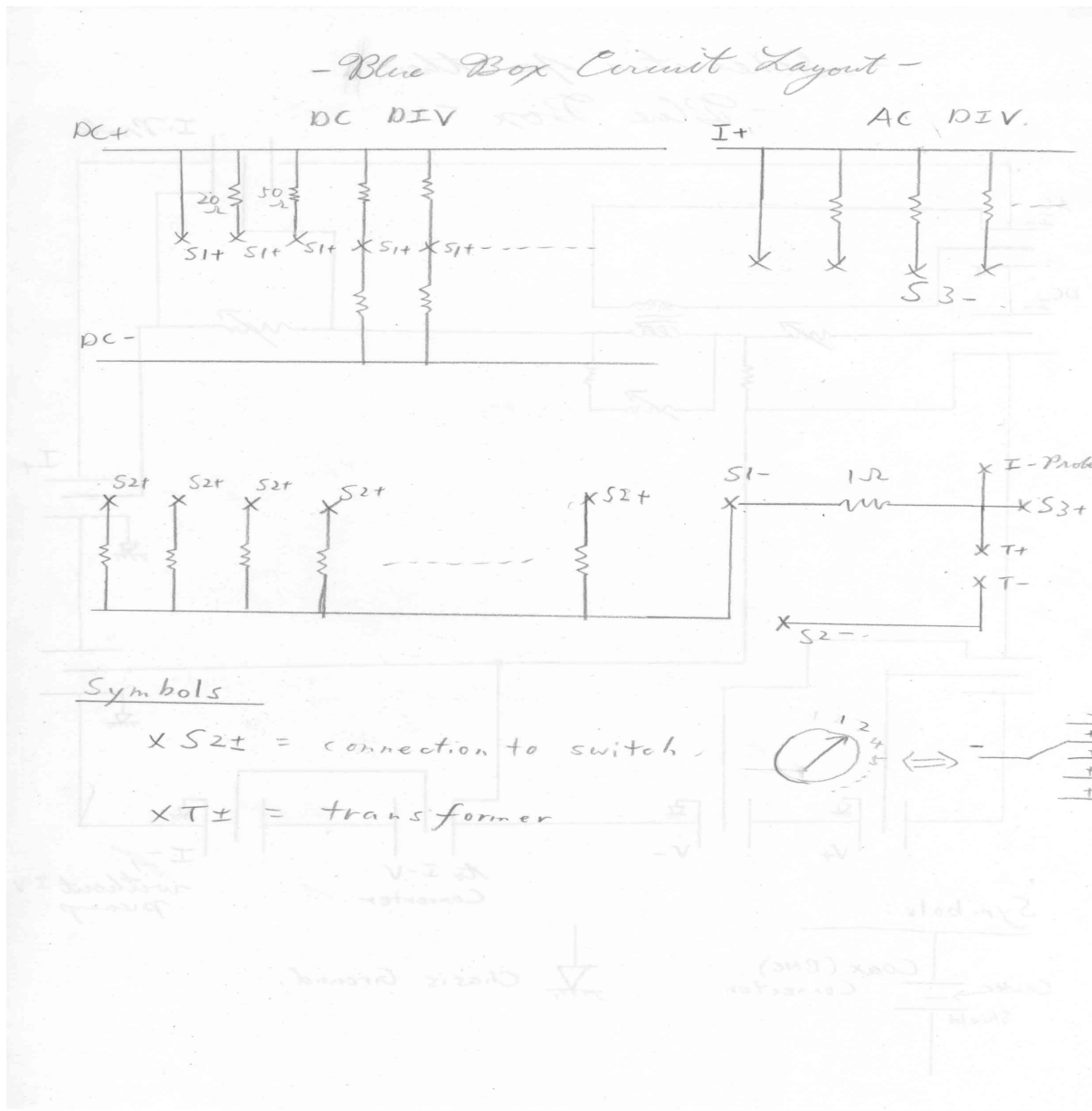
In the overall schematics of the Blue Box, the different functional parts are labeled in script letters in blue.

- Parts *A* and *C* are the voltage divider networks. The variable resistor is, in reality, one of many fixed resistors, one of which can be selected to be connected using a toggle switch on the box.
- Part *B* is the DC-biased transformer used to add the AC voltage into the DC one². It is a toroidal core power transformer, optimized for 60 Hz operation. The frequency response of the transformer is excellent at low frequencies (as opposed to audio transformers, which excel in the kHz range, and doesn't work at lower frequencies), enabling measurements at hundreds of Hz. At the same time, the symmetry of the toroid, together with the external shielding on the box, allows the transformer to pick up a minimum of EM radiation noise from the environment.
- Part *D* is the external resistor hinted earlier at the end of section II.2. This external resistor is in fact not a variable resistor. It is in fact one of several resistors soldered on to the circuit board, one of which is picked out and connected via another toggle switch on the box. The *I-Probe* port allows the voltage drop across this resistor to be measured, and thereby the current passing through the sample can be inferred. ***However, in standard operation, it is recommended that the I-V converter be used instead.*** The I-V Converter was found to produce a more repeatable reading.
- Part *E* is the output signal of the blue box. It goes to one of the contacts on the sample. Typically, this is Contact #1.
- Part *F* is the circuitry dealing with the I-V Preamp. Should the external resistor be used (*I-Pin*), Contact #4 on the sample should be connected to the *I-Connector*. Here, the current is simply sent to the negative pin on the DC Power Supply. However, *if the I-V converter is used*, the Contact #4 on the sample should be connected to the *I-With I-Preamp* connector. In this case, the current is sent into the input of the I-V Converter before it goes to the negative terminal in the DC power supply.
- Finally, Part *G* is a simple circuit that combines two Co-Axial cables into one. Since the input on the DC Multimeters usually takes in one co-ax input, it is built in for convenience. Canonically, it is connected to the *V+* and *V-* Contacts on the sample, and the combined output is connected to the LIA and the DC Voltmeter.

² The idea to use a DC-Biased Transformer belongs to Mr. Alan Stummer, who gave us numerous pieces of advice, for we are extremely grateful. The electrical box, as it stands now, could not have been designed without his help. The author originally intended to use a set of op-amps for adding the DC and AC signals.

Differential Conductance Measurement Apparatus Technical Manual

Inside the blue box, a circuit board was constructed to make Parts A, B, C and D to work. The other parts are done using actual wires inside the box. For convenience, the circuit board layout looks like:



III. SOFTWARE OVERVIEW

III.1 – MOTIVATION

The measurement is complex, so a software to automate the process is in order. The software is written using *National Instruments LabVIEW* software, in its unique language, known as G.

At first glance, the software serves to automatically:

- Coordinate & control all instruments
- Ensures that the sample is at the appropriate magnetic field and temperature
- Sets the appropriate applied voltage (AC amplitude, frequency, and DC offset)
- Measures resultant I-V and dI/dV curves
- Displays and saves the measurements to disk

III.2 – Control Program Features & Functionality

The Differential Resistance Measurement Control is a very complicated piece of software. It is written to make measurements as automatic and as easy as possible, while preserving enough user options so as to be as flexible as possible, without the need for the user to modify the code. As a result, sadly, the underlying code is relatively complex.

Since the basic features of the program will show up again in the user interface as well as the subsequent explanation of the code, we first go through a list of the functionality of this program.

The software is designed to be used with the Blue Box, as described in Section II.

III.2.A – Instrumentation Functionality and Feature List

- Automatic control of all relevant instruments:
 - Stanford Instruments SR810 Lock-in Amplifiers (x2)
 - Agilent 34401A Digital Multimeter
 - HP3457A Digital Multimeter
 - BK Precision 1787B Single Channel Power Supply
 - LakeShore 331 Temperature Controller
 - American Magnetics Model 430 Power Supply (for MARTI)
- With NO CONTROL for the following:
 - Stanford Instruments SR570 Low-Noise Current Preamp
 - Low-Noise Filters
- Automatic initialization and setting of all instruments
- Sample Temperature control.
- Sample Magnetic Field control.
- Can be used in either the I-V Converter or the External Resistor configurations

Differential Conductance Measurement Apparatus Technical Manual

III.2.B – Measurement Functionality & Feature List

- Automatically sets & waits for temperature and magnetic field to be stable before measurement.
 - Automatically takes dI/dV data at 0V bias, as the magnetic field sweeps
- Automatically calculates and sets correct DC voltages and AC amplitudes & frequencies, according to selected voltage dividers.
 - Automatically finds correct DC voltage to apply to achieve desired DC voltage drop on the sample, via a linear interpolation algorithm.
- Automatically configures the LIA and the HP Voltmeter for the correct voltage range, to avoid overload conditions.
- Automatically waits until electrical signal is stable before measurement.
- Data is averaged across multiple readings.

III.2.C – Data Handling

- Data for all runs are automatically saved to file, in Tab-Delimited ASCII files
 - Data files ready for direct import into OriginPro (via the import wizard), or Matlab, or directly pasted into Excel.
- Instrument and program settings automatically stored to a *header file* in ASCII format, which is human-readable in Notepad. This negates the need to record all settings manually. The header filename is the same as that of the data-file, but with a different extension.
- Data is saved after each data-point measurement – program can be stopped at any point, without fear of data loss.
- Data file name is automatically generated, containing:
 - A unique 3-digit Run ID for each measurement run.
 - So that each run can be easily identified.
 - Temperature of the sample at the beginning of the run.
 - Magnetic field in the chamber at the beginning of the run.
 - User-identified sample name.
 - Any other user-identified comments.

III.2.D – Automation

- If desired, user may define a list of measurements to be performed, at arbitrary temperature, magnetic field and electrical signal settings.
 - These measurements are performed in sequence, with no user intervention.

III.3 – MEASUREMENT SEQUENCE

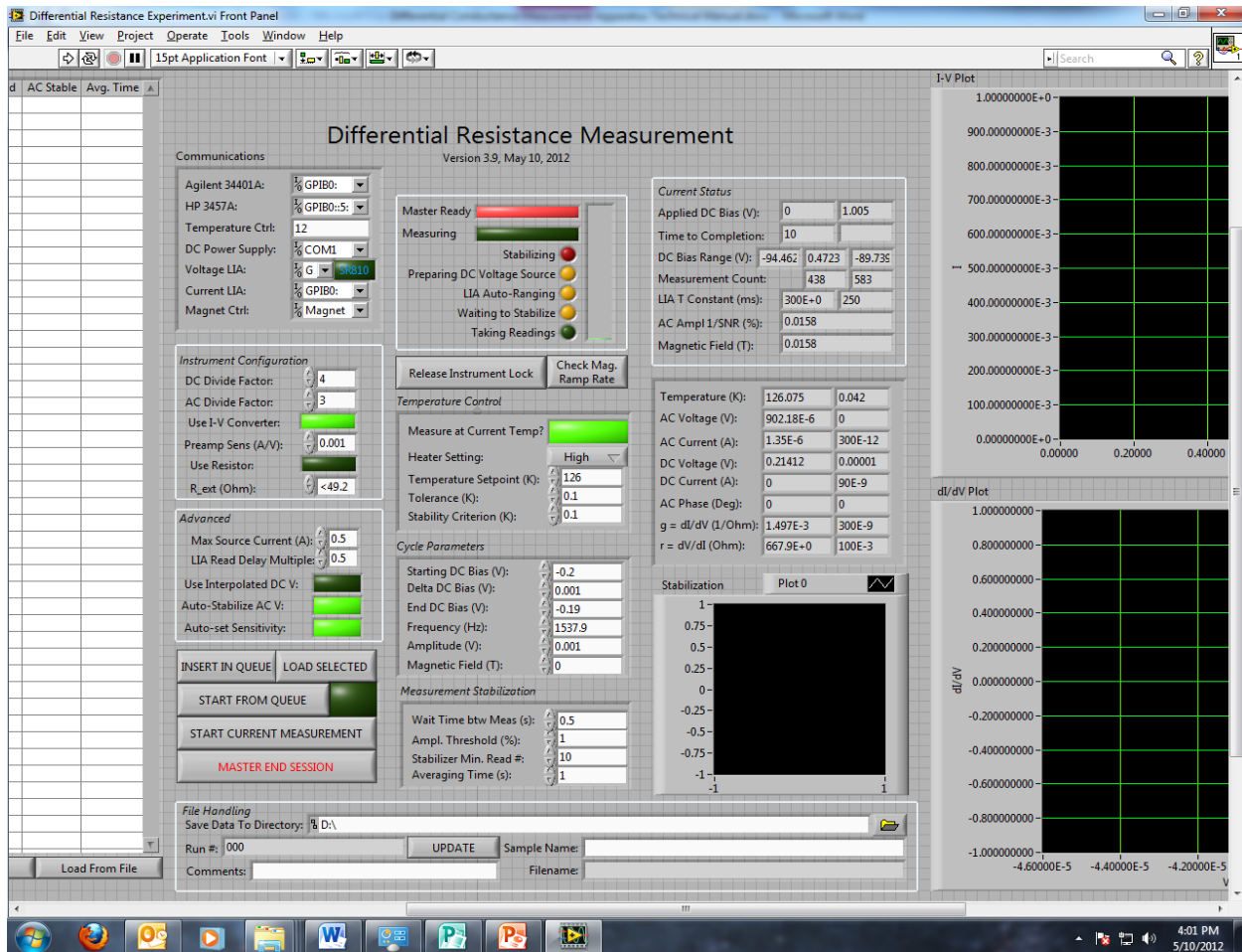
A measurement, automatic or manual, relies on performing a series of steps in succession. Here, the algorithm used to perform the measurement is discussed. Only a high-level summary of what occurs is sketched out here. A more detailed discussion follows when the code is discussed. Also, in the next section, the parameters of the measurement are defined, from the user interface.

Here, *text in italics* indicate if-conditions. If the condition is not met, then the step is skipped.

1. *If user commands it & if queue is nonempty,*
Load Parameters from Queue into Main Interface (on Right Hand Side)
2. From values supplied in the controls, calculate the appropriate voltages to supply.
3. *If not measuring at the current temperature*
Stabilize the temperature.
4. *If Linear Interpolated DC Voltage Estimation algorithm is on*
Perform the linear interpolation estimation algorithm.
5. *If the current magnetic field is not what is requested,*
Set the magnetic field to the correct value, and wait until the magnetic field reaches this value.
6. Generate appropriate filename for the datafiles.
7. Collect various experimental settings and parameters, and write it in the header file.
8. Configure the Lock-In Amplifiers for correct waveform output and settings.
9. Enter measurement loop.
For each datapoint in the measurement run,
(ie. for each different DC voltage,) do:
 1. Set the DC Power Supply to the appropriate voltage
 2. Wait for some time for readings to stabilize before adjusting gain/measuring.
 3. *If Auto-Set Sensitivity is enabled,*
Automatically set the sensitivity to both the HP Voltmeter and the LIAs.
 4. *If Auto-Stabilization is enabled,*
Wait until the LIA readings are stable.
 5. Take readings & calculate the differential conductance
 6. Display the data in charts
 7. Write the results to a new row in the data-file.
10. *If running from the Queue, and Queue is nonempty,*
Load next set of parameters, and repeat from A.
11. Finish the measurement.

Differential Conductance Measurement Apparatus Technical Manual

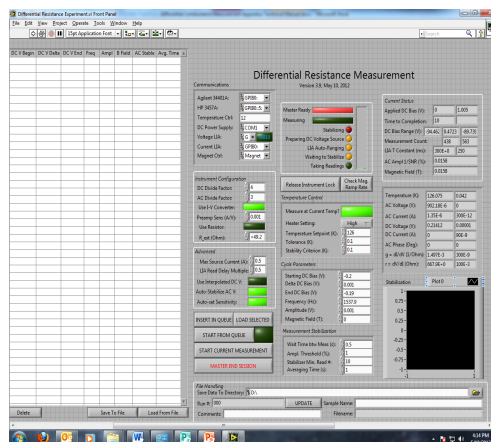
III.4 – MAIN PROGRAM USER INTERFACE



Description of Blocks

ID.	Label	Description
A	Queue	Loads pre-configured run parameters into the program
B	Communications	Sets the ports used to communicate to each device
C	Instrument Config.	Configures the instruments used by the measurement
D	System Settings	Sets various misc. settings used by the program
E	Buttons	Click to start or end measurements
F	Date-File Naming	Sets the directory in which data is saved, and the filenames used.
G	Status Indicator	Indicates which stage of measurement the software is in.
H	Temperature Control	Controls the temperature of the sample.
I	Cycle Parameters	Controls the electrical settings used for the measurement
J	Stabilization Parameters	Sets how long to wait for instruments to stabilize
K	Status Indicator	Indicates the current status of the system
L	Measurements	Displays the measured values
M	Stabilization Chart	Tracks how stable the LIA readings, temperature or magnetic field are
N	Plots	Plots the I-V diagrams and dI/dV data, as they come in.

III.4.A – Communications Settings Panel



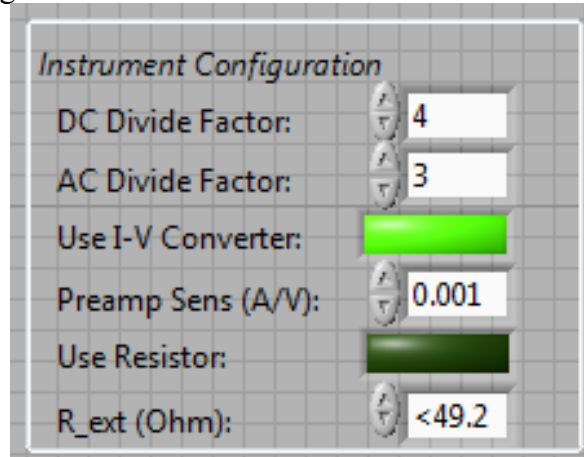
The Control Program communicates with all instruments via various ports on the PC. The majority of instruments communicate via the GPIB Port, where each instrument is assigned a unique address number. This number is selected on each instrument's communications setup menu. The DC Power Supply communicates via the RS-232 Serial Port (usually COM1 Port), and the MARTI Magnet Controller is connected via Ethernet. For more details, please see the Appendix on Communicating with Instruments.

The first time that MARTI is connected to a computer, its IP address needs to be added as an instrument under the NI Measurement and Automation Explorer (MAX). This is in the same folder, in the Start Menu, as LabVIEW.

For each instrument, the correct GPIB address needs to be selected. Furthermore, it is possible for the software to communicate with a Stanford Research Systems SR830 Dual-Channel Lock-In Amplifier, measuring the AC voltage drop across the sample. In this case, toggle the relevant LED button next to the Voltage LIA.

Differential Conductance Measurement Apparatus Technical Manual

III.4.B – Instrument Configuration Panel



The Instrument Configuration panel configures the system settings that are crucial for the measurement itself, but cannot be automatically read by the software. ***Incorrect configuration data here will result in erroneous results!***

In operating the measurement apparatus automatically, via the Queue, it is assumed that these manual configuration settings are not to be changed between measurement runs (because the human may not be present to change them!). Therefore, these settings are not loaded from the Queue.

Instrument Configuration Parameters

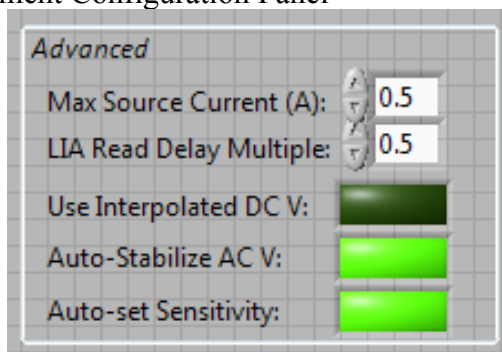
<i>ID.</i>	<i>Label</i>	<i>Description</i>
1	DC Divide Factor	DC Divide Factor set on the Blue Box DC Divider Toggle Switch
2	AC Divide Factor	AC Divide Factor set on the Blue Box AC Divider Toggle Switch
3	Use I-V Converter	Toggles whether the Current Preamp or the External Resistor is used.
4	Preamp Sens (A/V)	Conversion value, set by the I-V Converter, between current and voltage. If external resistor is used, this is automatically set according to #6.
5	Use Resistor	Toggle to use the External Resistor instead of the I-V Converter.
6	R_ext (Ohm)	Set the external resistor setting, from the label on the Blue Box. The resistance value is automatically loaded. Preamp sensitivity is automatically set, if #5 is true. If not, this value is simply recorded in the header file.

Exactly one of #3 and #5 Toggle Switches should be switched on.

Also, the Preamp Sensitivity (#4) is crucial. It is the conversion factor between the measured voltage reading from the I-V Converter output (or external resistor voltage drop), in volts, to the actual current, in amperes. An incorrect setting here will lead to all measured currents being off by a constant multiple. It is set by the I-V Converter.

Parameter #1 and #2 are used to set the DC Power Supply and the LIA Source to output a signal of the correct voltage, since it is necessary to compensate for the attenuation caused by the divider networks in order to supply the sample with the correct voltage drop.

III.4.C – Advanced Measurement Configuration Panel



The Advanced Measurement Configuration controls a series of program configuration settings.

Advanced Measurement Configuration Settings

ID.	Label	Description
1	Max. Source Current	Maximum current output from DC Power Supply.
2	LIA Read Delay Multiple	Sets the delay between readings from the LIA to be a multiple of the integration time, set on the LIA.
3	Use Interpolated DC V	Use an interpolation algorithm to find correct applied DC Voltage
4	Auto-Stabilize AC V	Controls whether to wait until LIA reading is stable before measuring
5	Auto-Set Sensitivity	Controls whether to check if the HP Voltmeter and the LIAs are configured to the correct measurement sensitivity range.

#1 is used to protect the circuit in the blue box from overheating. Since the voltage dividers dissipate some energy, it is necessary to ensure that not too much heat is being generated.

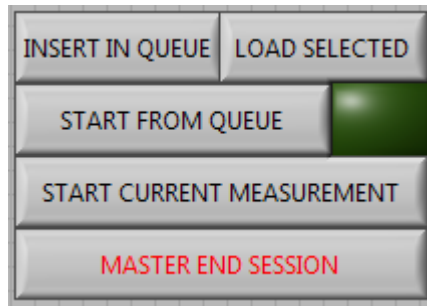
The *Interpolated DC V* option (#3) is an automatic algorithm that measures what voltages should be applied to the sample to achieve the user-specified voltage drop. Since a voltage divider is used to reduce the voltage reaching the sample, this applied voltage is related to the actual sample voltage drop by some multiplicative factor. When *Interpolated DC V* is turned off, the appropriate voltage to apply is estimated based on this factor only. When it is on, the estimated voltage is applied, and the actual voltage drop measured. Based on a linear interpolation between this data point, and the 0-crossing, a better estimate for the voltage is produced.

Auto-Stabilize AC V option (#4) and the *Auto-Set Sensitivity* option (#5) are useful for ensuring that the measured values are accurate. However, if the experimentalist watches the apparatus carefully, and makes sure that the LIA and HP Voltmeter sensitivity are well-adjusted, and the readings are stable, then disabling these options will save time.

Differential Conductance Measurement Apparatus Technical Manual

III.4.D – User Commands Panel

The User Commands section of the interfaces allow the user to start measurements and manage the queue.

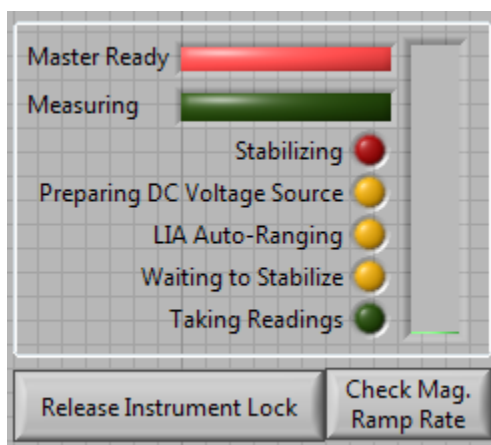


User Commands

<i>ID.</i>	<i>Label</i>	<i>Description</i>
1	Insert in Queue	Insert current measurement parameters into the Queue, (see III.4.N)
2	Load Selected	Load measurement parameters from selected row in the Queue. (see III.4.N)
3	Start from Queue	Run all measurements in the Queue in order, until the Queue is empty.
4	Running from Queue Indicator	Indicator is ON if the measurement is running from the Queue. Indicator is OFF if measurement parameters are from the main interface.
5	Start Current Measurement	Start the measurement in the main interface. When measurement is complete, stop. Ignore the Queue.
6	Master End Session	End execution of the program. Close communication ports with instruments.

III.4.E – Measurement Progress Status Display Panel

The Measurement Progress Status Display Panel displays the progress done on the current measurement.



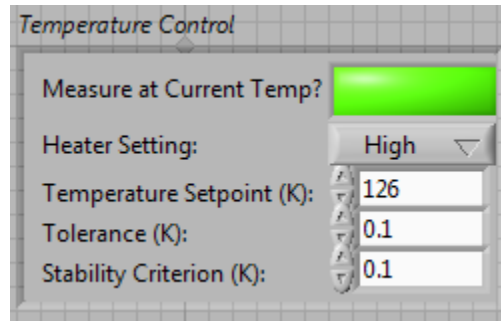
Status Indicators

<i>ID.</i>	<i>Label</i>	<i>Description</i>
1	Master Ready	GREEN when instruments are properly initialized. RED otherwise.
2	Measuring	ON when measurement taking place. OFF otherwise.
3	Stabilizing	ON when magnetic field or temperature is being ramped. OFF otherwise.
4	Prep. DC V Source	ON when computer is calculating appropriate voltage to send to DC Power Supply
5	LIA Auto-Ranging	ON when LIA and HP Voltmeter sensitivity are automatically adjusted.
6	Waiting to Stabilize	ON when LIA readings are monitored, until Stdev of last <i>N</i> readings are small.
7	Taking Readings	ON when actual measurement datapoint is being taken.
8	Progress Bar	Indicates the percentage of datapoints in the current run that are taken.
9	Release Instrument Lock	Send command to all instruments setting them to Local mode.
X	Check Mag. Ramp Rate	Verify the magnetic field ramp rate is at or below the maximum rated ramp rate, from the MARTI Manual.

Differential Conductance Measurement Apparatus Technical Manual

III.4.F – Temperature Control Panel

Sample behavior changes with temperature. The Temperature Control allows the user to change the



Temperature Control

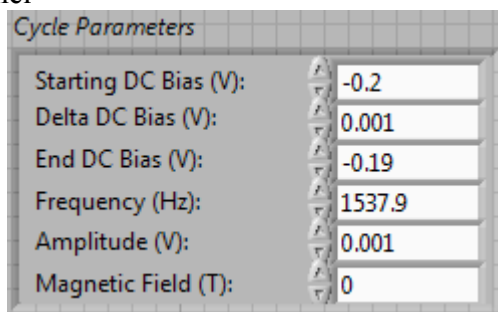
<i>ID.</i>	<i>Label</i>	<i>Description</i>
1	Measure at Current Temperature?	ON if measurement should occur at the current sample temperature. OFF if the system should wait for sample to stabilize to indicated temperature.
2	Heater Setting	Select between Low, Medium or High Heater Setting used by the T. Controller
3	Temperature Setpt.	Input desired temperature of the sample.
4	Tolerance	The range around the T. setpoint where sample temperature is allowed to stabilize.
5	Stability Criterion	Standard deviation of the temperature below which T is considered stable.

Unlike the Magnetic Field, it is difficult for the sample to be stabilized to a certain temperature within a short amount of time. Nominally, some waiting is involved, and often it is easier and faster for the human to manually stabilize the temperature. As such, the temperature control can be disabled using the *Measure at Current Temperature* option (#1).

When automation is warranted, it is often slow. The method by which the machine achieves temperature stability is to first of all heat/cool the sample until it is within the *Tolerance* (#4) of the desired *Temperature Setpoint* (#3). It then sets the heater output to be fixed at whatever value it was when this is achieved. Next, it waits for the fluctuations in temperature to be below the *Stability Criterion* (#5).

This complicated algorithm was devised because, originally, the PID algorithm on the temperature controller was not able to reliably stabilize the temperature to the desired value, without significant delays in time.

III.4.G – Cycle Parameters Panel



Cycle Parameters

<i>ID.</i>	<i>Label</i>	<i>Description</i>
1	Start DC Bias (V)	Sets the desired DC Voltage Drop at the Sample, at first data point in the run.
2	Delta DC Bias (V)	Sets the difference in voltage drop between successive data points.
3	End DC Bias (V)	Sets desired Sample Voltage Drop at last data point in the run.
4	Frequency (Hz)	Sets AC Frequency in Hertz.
5	Amplitude (V)	Sets amplitude of the AC waveform, in V, across the sample.
6	Magnetic Field (T)	Sets the desired magnetic field at the sample.

The Cycle Parameters pane sets the most important measurement parameters. All values are the desired values at the sample itself.

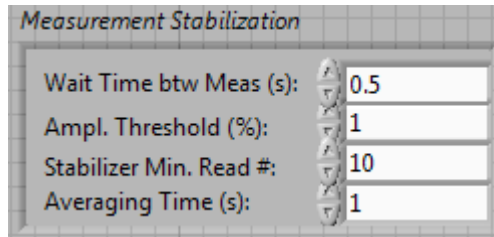
It is possible for the DC voltages to be swept backwards, by setting *Starting DC Bias* (#1) > *End DC Bias* (#3), and by ensuring *Delta DC Bias* (#2) is negative.

It is recommended that the *Frequency* (#4) be set near 200 Hz, so as to be away from the AC Power Line, but is still low enough to escape high frequency effects.

The *Amplitude* (#5) should be set to a value below the *Delta DC Bias* (#2). Otherwise, the result would be needlessly convoluted in the voltage domain.

Differential Conductance Measurement Apparatus Technical Manual

III.4.H – Measurement Stabilization Panel



Measurement Stabilization Parameters

<i>ID.</i>	<i>Label</i>	<i>Description</i>
1	Wait btw. Meas.	Sets seconds to wait for initial stabilization. Occurs before auto-setting sensitivity.
2	Ampl. Threshold	Signal-to-Noise tolerance of the last N readings, below which the signal is considered stable
3	Stabilizer Min. Read #	Defines <i>N</i> above in #2.
4	Average Time (s)	Sets number of seconds over which each reading is averaged.

The Measurement Stabilization parameters define settings having to do with when a signal is perceived to be stable. Increasing tolerances will generally result in quicker measurements, at the cost of lessened accuracy and precision.

Wait Time between Measurements (#1) was implemented in an effort to allow the LIA readings to settle before setting the sensitivity. This allows for a significantly better, and sometimes far faster, sensitivity adjustment. Its optimal value is sample-dependent.

The Auto-Stabilization algorithm works with *Amplitude Threshold (#2)* and *Stabilizer Minimum Readings (#3)* settings. Effectively, to see if the current LIA readings are stable, the program continuously reads from the LIA, at a speed of $0.5T$, or two readings per integration time period. The last *N* readings are then calculated for the Signal-to-Noise Ratio (SNR). If the SNR is below the *Threshold (#2)*, then the process ends, and measurement starts.

The program then takes continuous measurements in all instruments, for the time specified in *Averaging Time (#4)*. Readings are taken at the maximum permissible speed on all instruments, except the LIAs, where readings are taken at a fractional multiple of the integration time *T*. Usually, the slowest instrument is the old HP Voltmeter, where a 1s is sufficient for two separate measurements. However, the HP Voltmeter is able to make 7 readings in 2 seconds³. The separate readings are then averaged for the final reading, stored to file.

³ Because of the way the HP Voltmeter works, it requires about 0.5s to set up for the first measurement, then 0.21s for each subsequent reading. This information was found experimentally.

III.4.I – File Management Panel

The screenshot shows a software interface titled 'File Handling'. It contains several input fields and a button:

- 'Save Data To Directory': A text box containing 'D:\' with a folder icon on the right.
- 'Run #': A text box containing '000'.
- 'UPDATE': A button.
- 'Sample Name': A text box.
- 'Comments': A text box.
- 'Filename': A text box.

File Management

<i>ID.</i>	<i>Label</i>	<i>Description</i>
1	Save Data to Dir	Sets directory where all data is stored. Press the folder icon to browse.
2	Run #	Run ID, a 3-digit number unique in the folder to each measurement.
3	Update	Press to update to load the next run number.
4	Sample Name	User-provided label for the sample name
5	Comments	User-provided label for any comments
6	Filename	Computer-generated Filename

To aid automation, and make data-gathering as painless as possible, a unique name is generated for each experimental run. For this to work, the experimentalist selects a directory to store the data (#1), and the computer automatically generates a Run Number (#2) for each measurement in that directory.

The final filename is of the form:

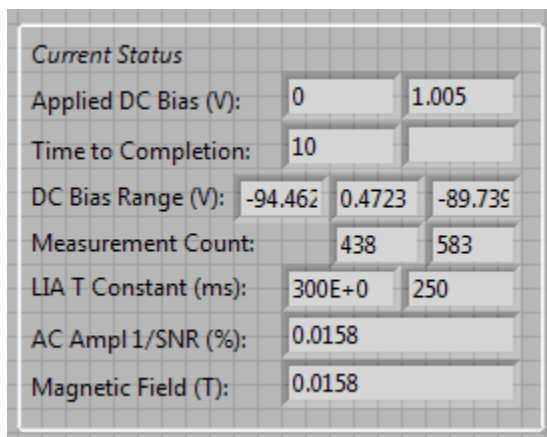
Format: <ID> - <TEMP>K, <B-Field>T, <Sample>, <Comments>.dat

Example: 052 - 001.0K, -2.000T, Sample, Comments.dat

Differential Conductance Measurement Apparatus Technical Manual

III.4.J – Current Status Panel

The Current Status panel displays settings used by the instruments. Each row shows related parameters. Due to spacing constraints, it was not possible to write down all labels.



Current Status

<i>ID.</i>	<i>Label</i>	<i>Description</i>
1	Applied DC Bias	<i>DC Voltage</i> applied by the DC Power Supply
2	---	<i>AC Voltage</i> applied by the LIA
3	Time to Completion	Estimated time, in minutes, until the measurement run is done.
4	---	Time when the measurement is done.
5	DC Bias Range	DC Power Supply Voltage. Boxes for Start, Voltage Delta, and End.
6	Measurement Count	Count of the datapoint currently being measured.
7	---	Total number of datapoints in the measurement run.
8	LIA T Constant	LIA integration time, in ms.
9	---	Delay time, in ms, between adjacent LIA reads. ⁴
A	AC Ampl 1/SNR	Larger of the two Noise-to-Signal Ratio of the two LIAs
B	Magnetic Field	Magnetic Field at the sample.

⁴ The delay time is one of several preset by the device. The program can only choose the one slightly longer than the one we want, which is a fractional multiple of the integration time T .

III.4.K – Measurement Indicator Display Panel

This panel displays the current measurements made by the instrument. *The first column always displays the averaged reading, while the second always displays the error (ie. the standard deviation).*

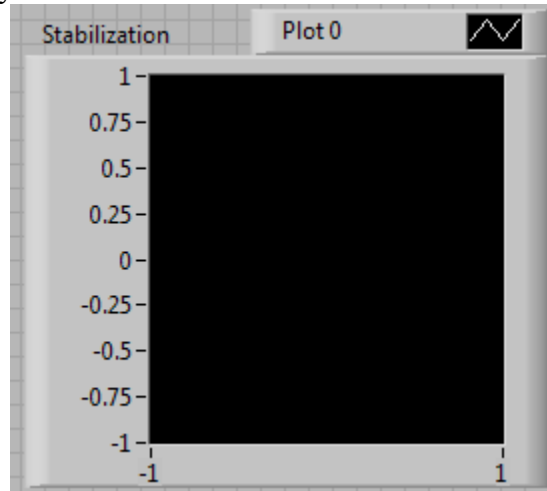
Temperature (K):	126.075	0.042
AC Voltage (V):	902.18E-6	0
AC Current (A):	1.35E-6	300E-12
DC Voltage (V):	0.21412	0.00001
DC Current (A):	0	90E-9
AC Phase (Deg):	0	0
$g = dI/dV$ (1/Ohm):	1.497E-3	300E-9
$r = dV/dI$ (Ohm):	667.9E+0	100E-3

Measurement Indicator Display Panel

<i>ID.</i>	<i>Label</i>	<i>Description</i>
1	Temperature	Temperature measured by the sensor.
2	AC Voltage	Amplitude of AC voltage drop across sample
3	AC Current	Amplitude of AC current across sample
4	DC Voltage	DC Voltage drop across sample
5	DC Current	DC current passing through sample
6	AC Phase	If using SR830 LIA, phase shift in voltage drop signal compared to the source.
7	dI/dV	Differential conductance of the sample, in 1/Ohm
8	dV/dI	Differential Resistance at the sample, in Ohms

Differential Conductance Measurement Apparatus Technical Manual

III.4.L– Stabilization Display



This display shows the instantaneous value of the parameter being stabilized. The color of the plot corresponds to the parameter itself. Temperature will be displayed in *Green*, Magnetic Field in *Blue*, while the AC Voltage and AC Current will be displayed in *White* and *Yellow*, respectively.

The x-axis displays the current measurement number. The y-axis is either the actual value, or a normalized value.

III.4.M – Data Plots

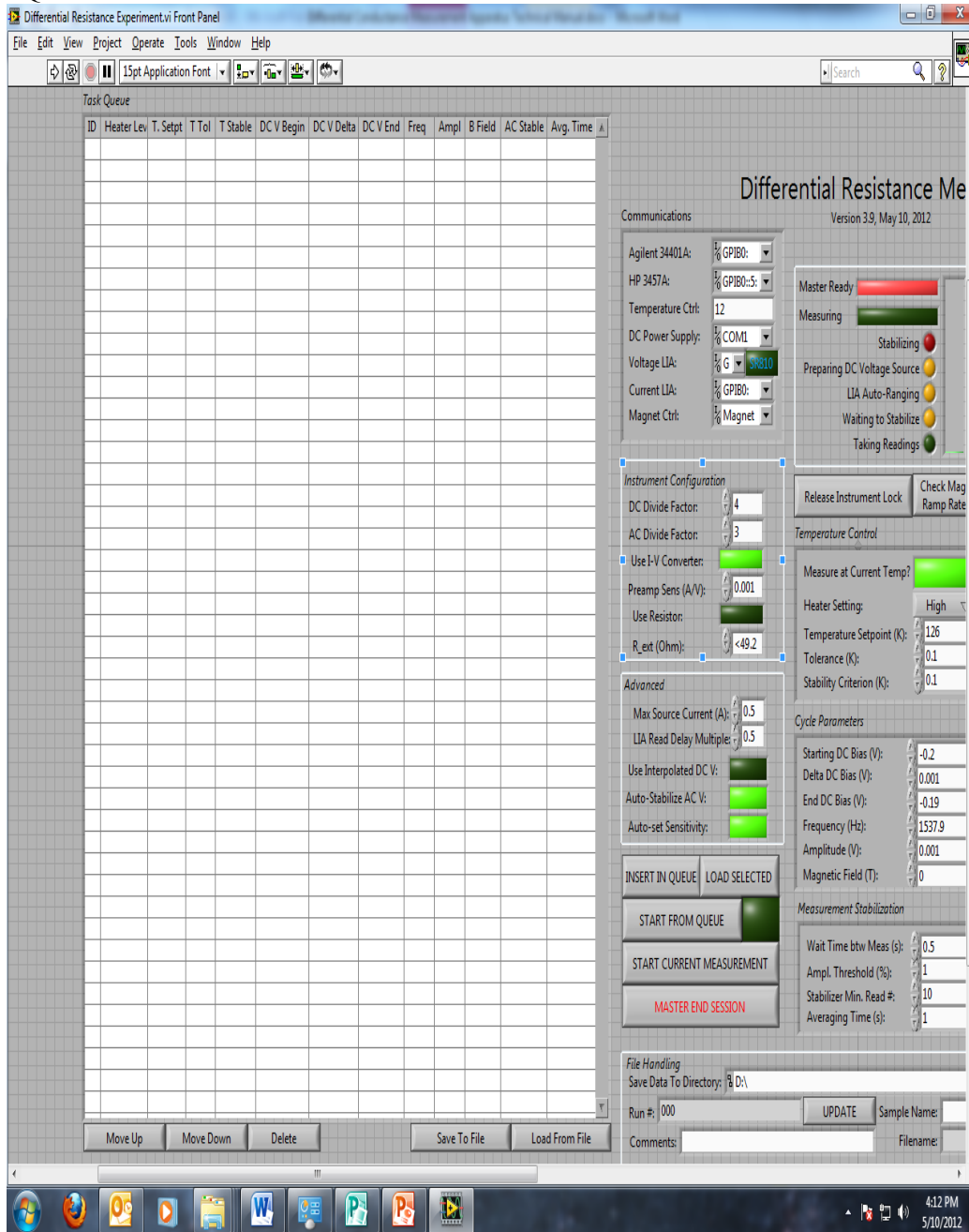
The measurement data is plotted in real time. Here, the top chart shows the DC Voltage Drop across the sample vs. the DC Current passing through. The bottom chart shows the DC Voltage Drop across the sample vs. the Differential Conductance through the sample.

The units on the axes are all standard SI base units. The x-axes are in Volts, while the y-axes are in Amperes and 1/Ohms respectively.



Differential Conductance Measurement Apparatus Technical Manual

III.4.N – Queue



The Task Queue records a list of future measurements to perform. Each row corresponds to a planned measurement run, and each column corresponds to a parameter for that run. The parameters are exactly the same as the ones on the right hand side of the screen.

The buttons on the left performs the specified activity on the currently-selected row. The right-hand-side buttons records the current list of measurements to a text file.

IV. SOFTWARE CODE AND ALGORITHMS

IV.1 – MOTIVATION

Knowledge of the user interface is enough for most operations, but if modifications to the code were necessary, it would be necessary to understand how the code works. Here, a summary of how the software works is presented.

The software is a rather complicated piece of code. In truth, a real programming language like C++ or C# would have been better suited to this program. However, since when the project started, the author had already had significant experience with instrument interfacing in LabVIEW, and the program was not intended to be this complex, the decision was made to use LabVIEW. As a result, the code is literally messy, and while the author had tried, as much as time constraints allowed, to clean this up, other experimental goals took higher priority. He apologizes in advance for this oversight.

Also, this description of the code was valid for Version 3.9 of the Program, finished on May 10, 2012. Later versions of the code will inevitably contain changes.

An outline of the algorithm was presented in Section III.3. Here, we will talk about the code that was actually used in implementing this algorithm. We will also talk about the algorithm in more detail.

IV.2 – GENERAL PROGRAM LAYOUT

Fundamentally, the program is *Event Based*. When the script first starts, it initializes all instruments, and waits for user input. After all parameters are put in, and the user presses certain buttons, these Events are triggered.

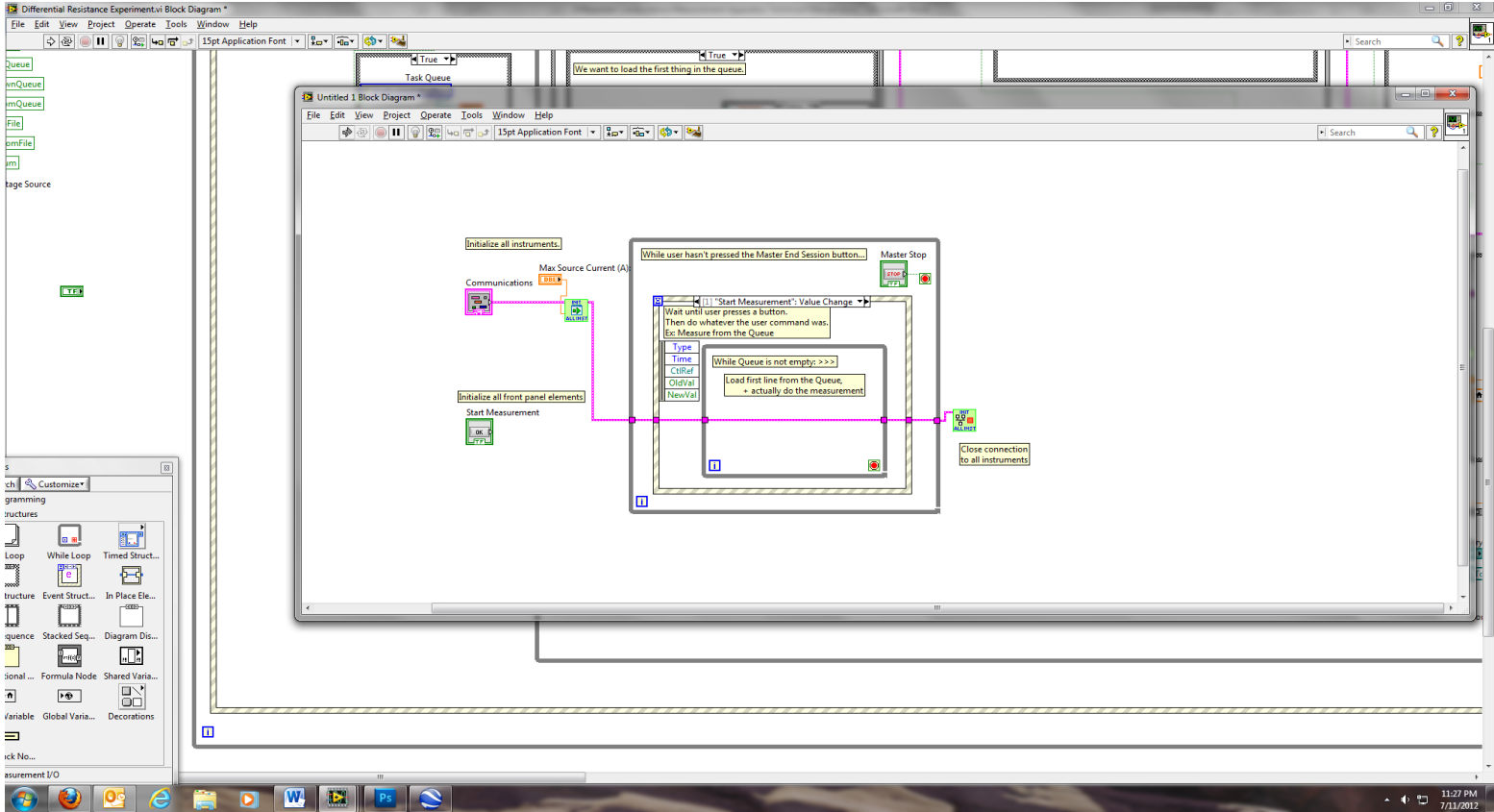
Events are like if-conditions, except it is more like a *when*-condition. An event is registered *when* a button is pressed, or a variable's value changes, or the mouse goes somewhere. A so-called *Event Structure* contains many such conditions, and one of which might be triggered at any given time. When this happens, the code contained inside the triggered condition will run.

Since the program continuously waits for events to occur, it always executes in a while loop, which only terminates when the user stops the program. When the program starts, it first initializes all instruments, then enters this event loop.

In LabVIEW, the structure looks like many layers of an onion. Recall that LabVIEW commands run whenever all necessary inputs are known. In this instance, one could safely read the code from left to right, from outside in.

In reality, the code is slightly more complicated. But the complication is just a variation on a theme – the basic structure here is the backbone of the whole program.

Differential Conductance Measurement Apparatus Technical Manual



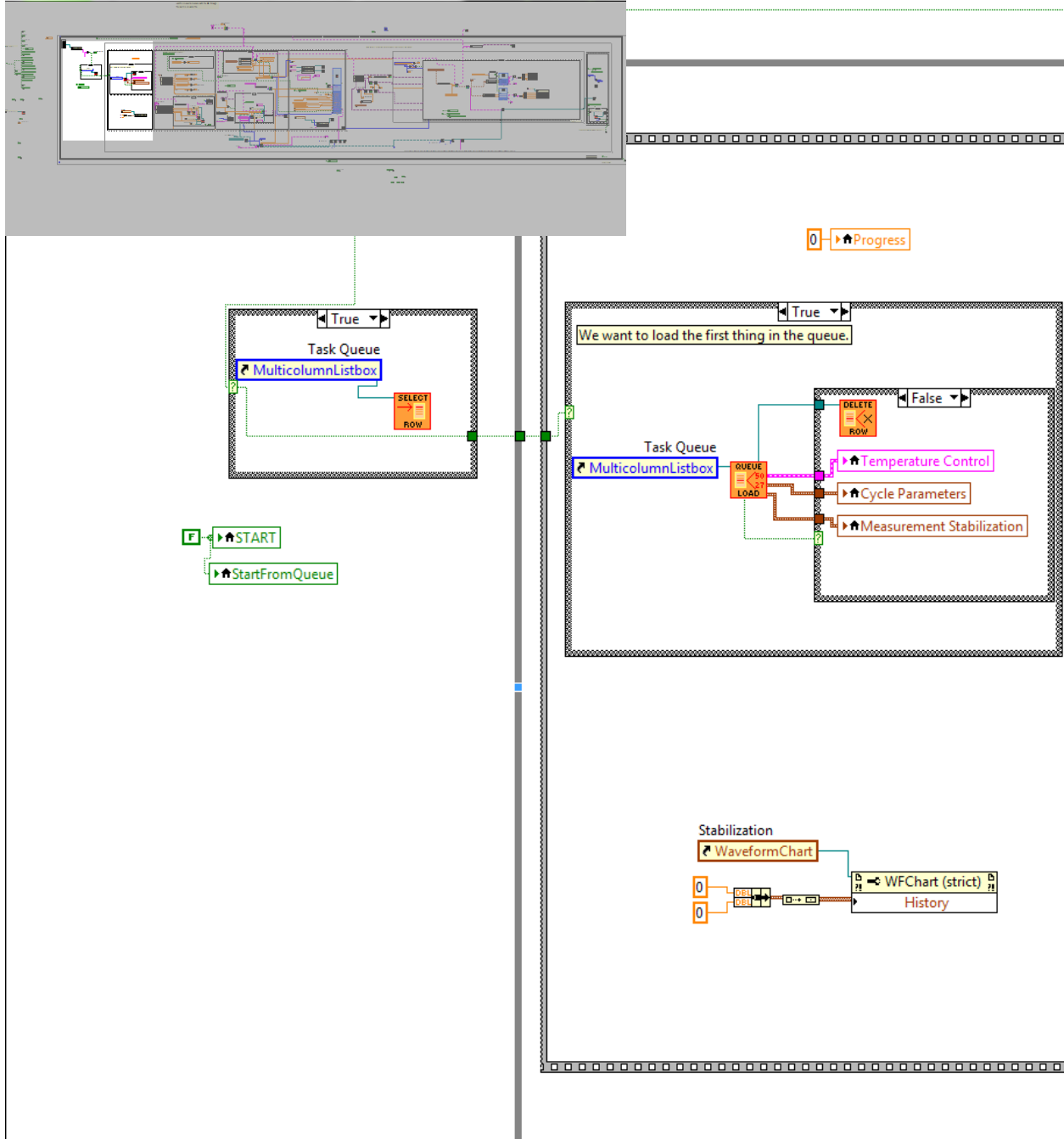
When the program starts, it initializes all instruments. The Master Ready indicator goes Green, and the program waits for user input, in the form of events. When the master Stop button is pressed, the loop exits and all connections to instruments are closed.

IV.3 – MEASUREMENT

The most complicated part of the program is the main Measurements Algorithm. A sketch of the steps involved is outlined in Section III.3. Here, we will look at the code implementing this. We will go step-by-step, in the order of Section III.3. Labels are added, for clarity, in *blue script*.

IV.3.A – Load Queue Parameters

If queue is nonempty, load parameters from queue into Main Interface.



The first thing here is to tell if the user wanted to use the queue or not. In *A*, the label of the button that triggered the event is loaded. First, only one of two buttons (“Start from Queue” and

Differential Conductance Measurement Apparatus Technical Manual

“Start from Measurement”, known to the system as “Start”) will trigger this measurement event. If the button is not simply called “Start”, then we must be using the queue.

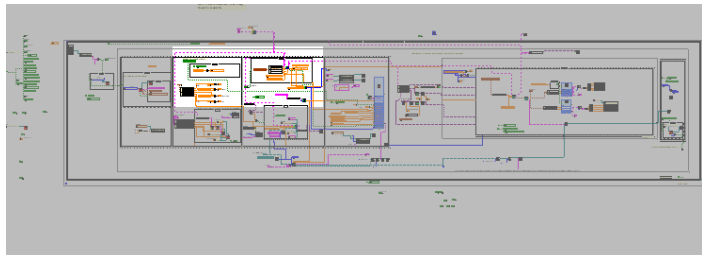
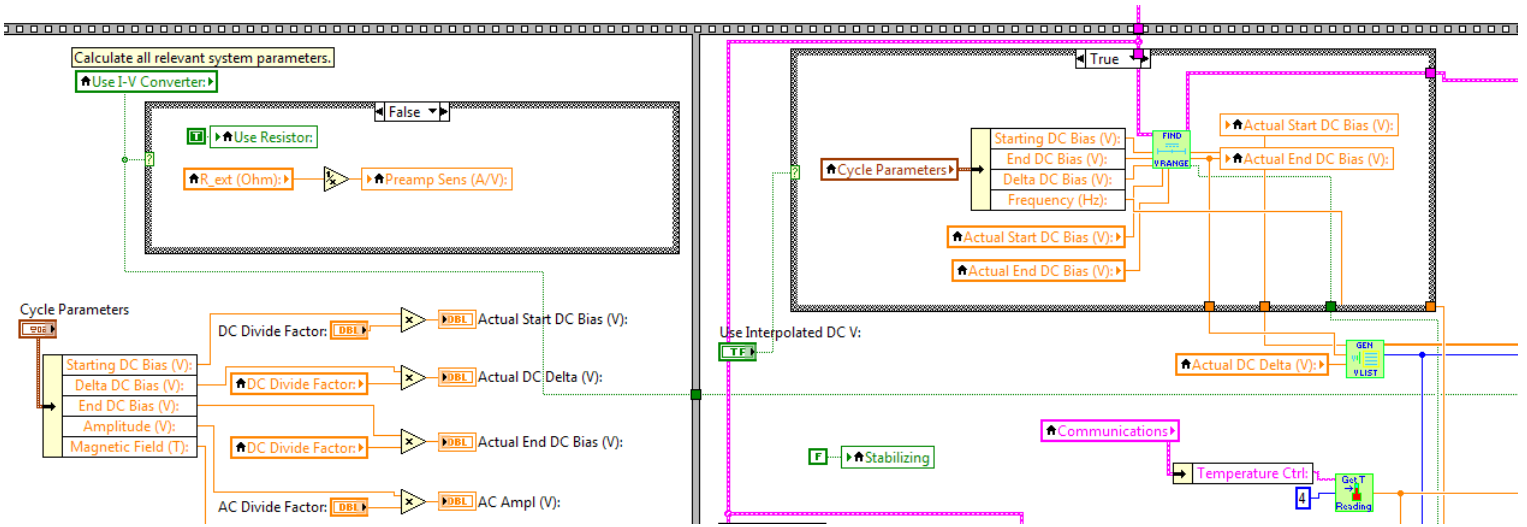
If we are using the queue, in [E](#), select the first row. Otherwise, do nothing.

In [C](#), We then load the selected row’s data into the controls in the UI, then deletes the row. When the program calculates the parameters to send to the instruments, it uses the values in the controls on the primary UI. This step ensures that the code is measuring the first thing in the queue.

At the same time, in [D](#), the stabilization chart is cleared.

IV.3.B – Calculate Supply Voltages

From values supplied in the main UI, calculate what voltages to apply from the Power Supply.



Having loaded the settings from the Queue, the first thing to do is to calculate the actual operating parameters the instruments require.

If the external resistor is used to sense current, \mathcal{A} calculates the coefficient between measured voltage and actual current through the sample. If the I-V Preamp is used, no changes are made.

The power supply and LIA will need to know the AC amplitude and DC voltages to apply. At \mathcal{B} , the parameters are estimated based on the value of the AC and DC voltage dividers.

The estimated DC voltages are not perfectly tuned to generate the sample voltage drop requested. If the user turned on *Interpolate DC V*, code at \mathcal{C} will perform this interpolation. In effect, the computer will apply the estimated voltages, and measure the actual voltage drop across the sample. We also know that when 0V at the power supply results in 0V across the sample. The optimal voltage to apply is then an interpolation between these datapoints. If *Interpolate DC V* was disabled, nothing happens.

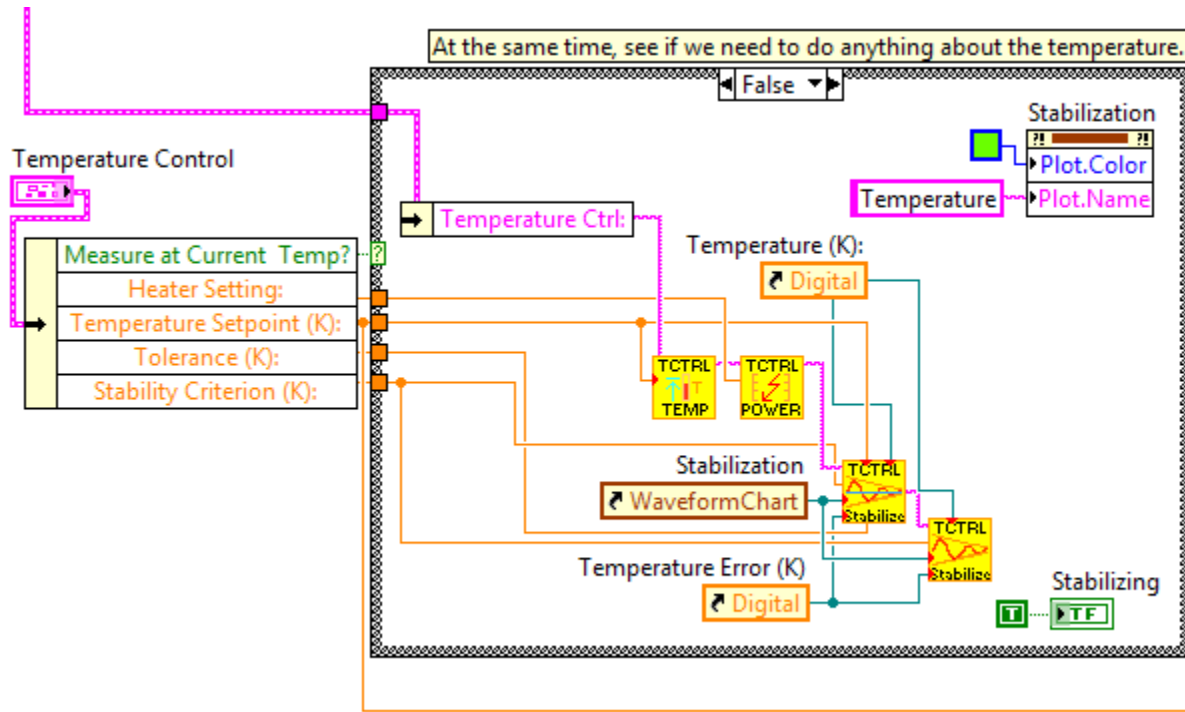
For instance, if the user requested to probe the sample at 0.1V, and the DC voltage divider has a factor of 0.1, 1V will be the estimated voltage at the power supply. However, due to contact resistances, the voltage drop at the sample might only be 0.07V. The Interpolated DC V algorithm will detect this automatically, and linearly interpolate for the power supply voltage resulting in a 0.1V drop across the sample.

At \mathcal{D} , an array of voltages is generated. Effectively, this generates the array (in Matlab syntax) $vList = vMin:vDelta:vMax$;

Differential Conductance Measurement Apparatus Technical Manual

IV.3.C – Stabilize Temperature

If not measuring at the current temperature, then stabilize the temperature.



Here, if *Measure at Current Temperature* is Enabled, nothing happens. Otherwise, the computer proceeds to stabilize the temperature to the setpoint.

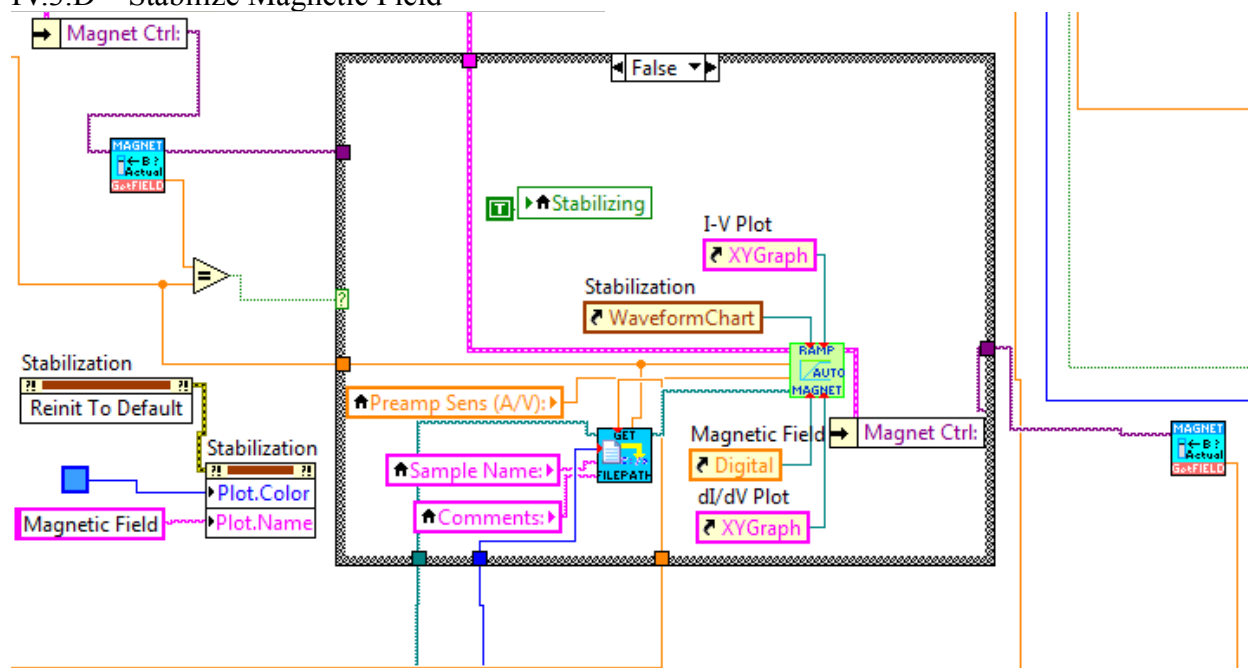
First, at \mathcal{A} , the color of the stabilization chart is set to green.

Then, the first two commands at \mathcal{B} sets the temperature controller to the desired temperature setpoint, and turns the heater on to the level specified by the user.

At \mathcal{C} , the computer waits for the temperature to go to the setpoint. The SubVI will only exit when the temperature averaged from the 40 previous readings (at 0.5s apart) are within *Tolerance* of the setpoint, **and** when the standard deviation of the last 40 samples are below the *Stability Criterion*. When this is done, it will command the heater to output a constant power.

Finally, with the heater set to constant power, the sample will reach an equilibrium temperature. \mathcal{D} looks for the standard deviation in temperature (across the last 40 readings, taken 0.5s apart) to reach below the *Stability Criterion*. This ensures that the sample temperature is stable.

IV.3.D – Stabilize Magnetic Field



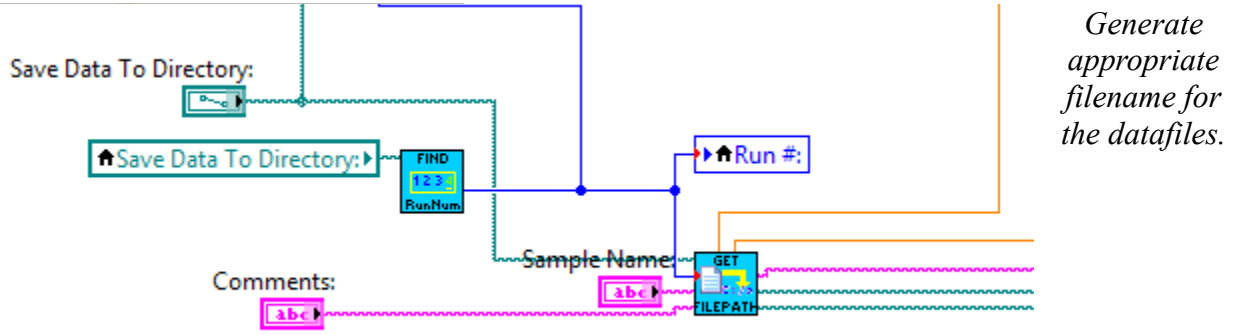
Set the magnetic field to the correct value, and wait until the magnetic field reaches this value.

At \mathcal{A} , the magnetic field value is read and compared to the setpoint. If they agree, nothing is done. Otherwise, at \mathcal{B} , a filename for the ramp datafile is generated. At \mathcal{C} , the magnet is commanded to go to the setpoint, while the DC power supply is configured for 0V, and readings are continuously taken. The information recorded is saved into the ramp datafile, for analysis.

The SubVI at \mathcal{C} quits when the magnetic field reaches the setpoint.

Differential Conductance Measurement Apparatus Technical Manual

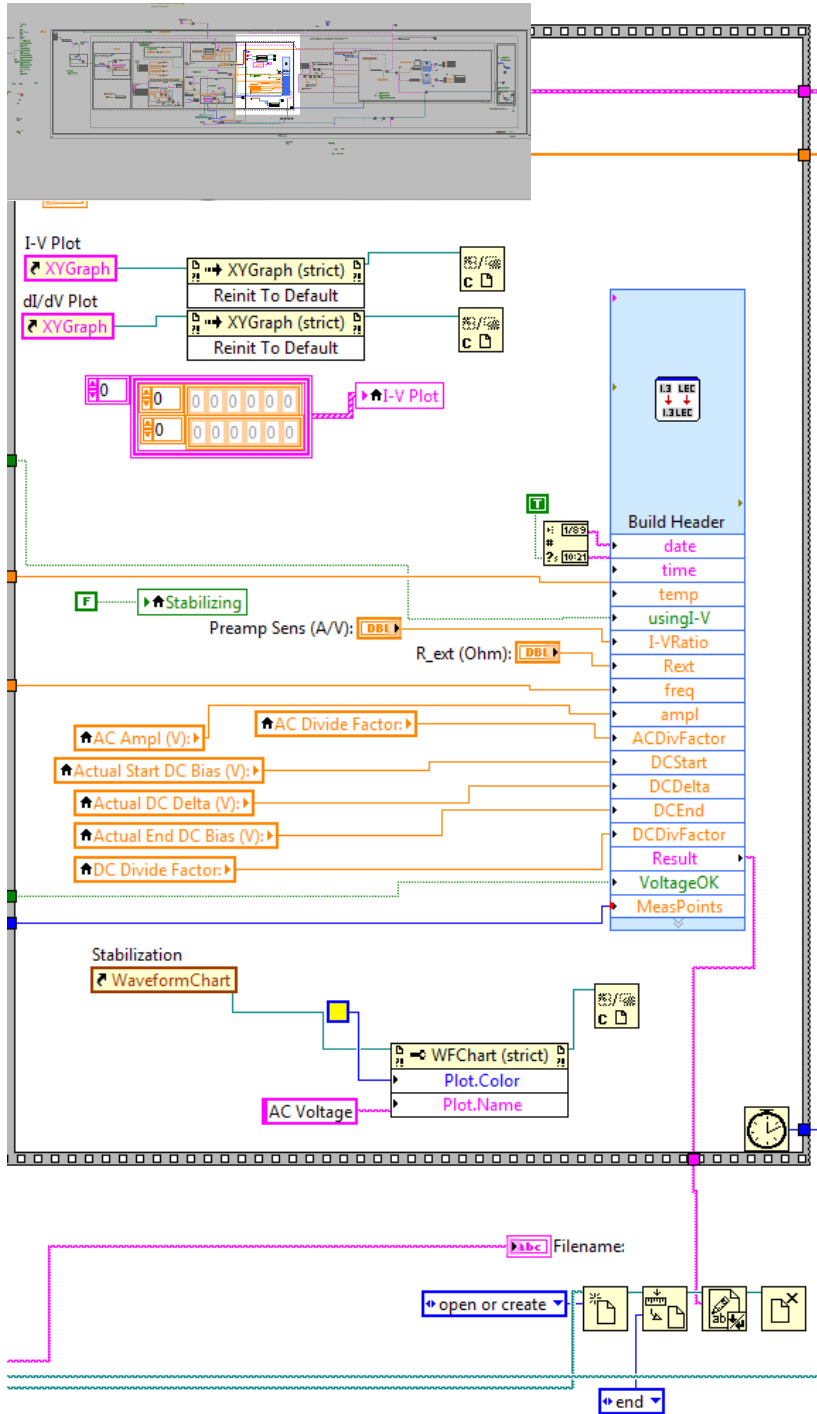
IV.3.E – Generate Filenames



The program then generates the filenames for each of the files. First, at *A*, it looks for all previously generated files in the folder, and generates the next available Run ID. *B* will generate the rest of the filenames, with the temperature and magnetic field information.

IV.3.F – Generate Header File

Generate Header File based on Parameters. Save this data.



\mathcal{A} clears the contents of the I-V and dI/dV plots.

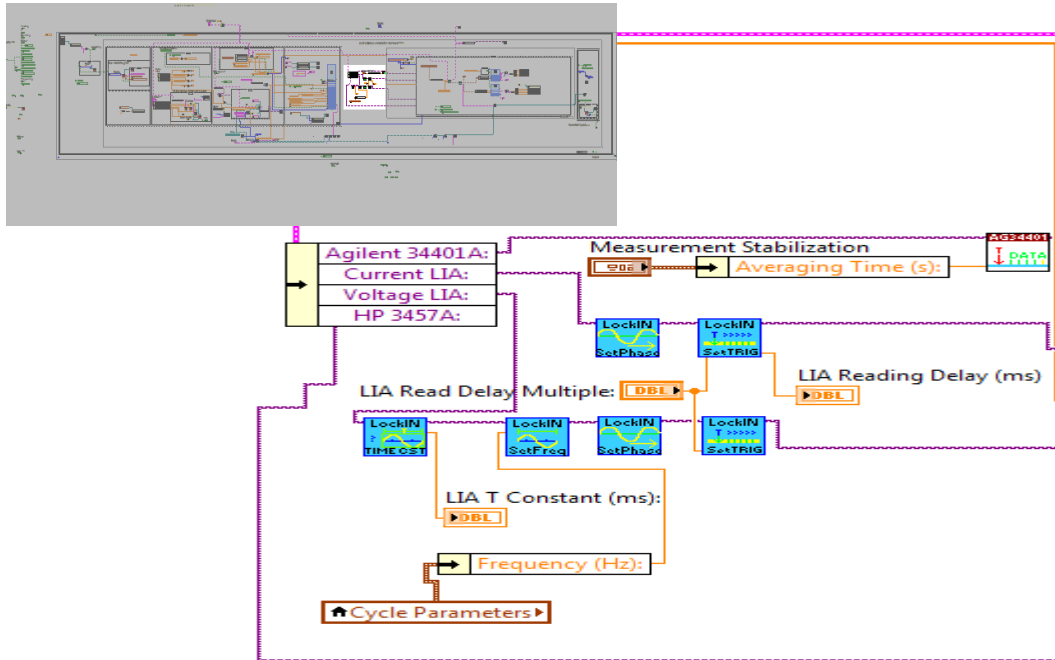
\mathcal{B} collects the information relevant to the header file, and writes it to a long string. The string then goes to \mathcal{C} , where it is saved to the header file.

The measurement start time is recorded at \mathcal{D} , to estimate the total time required to complete the measurement.

Differential Conductance Measurement Apparatus Technical Manual

IV.3.G – Configure LIA

Configure the Lock-In Amplifiers for correct waveform output and settings.

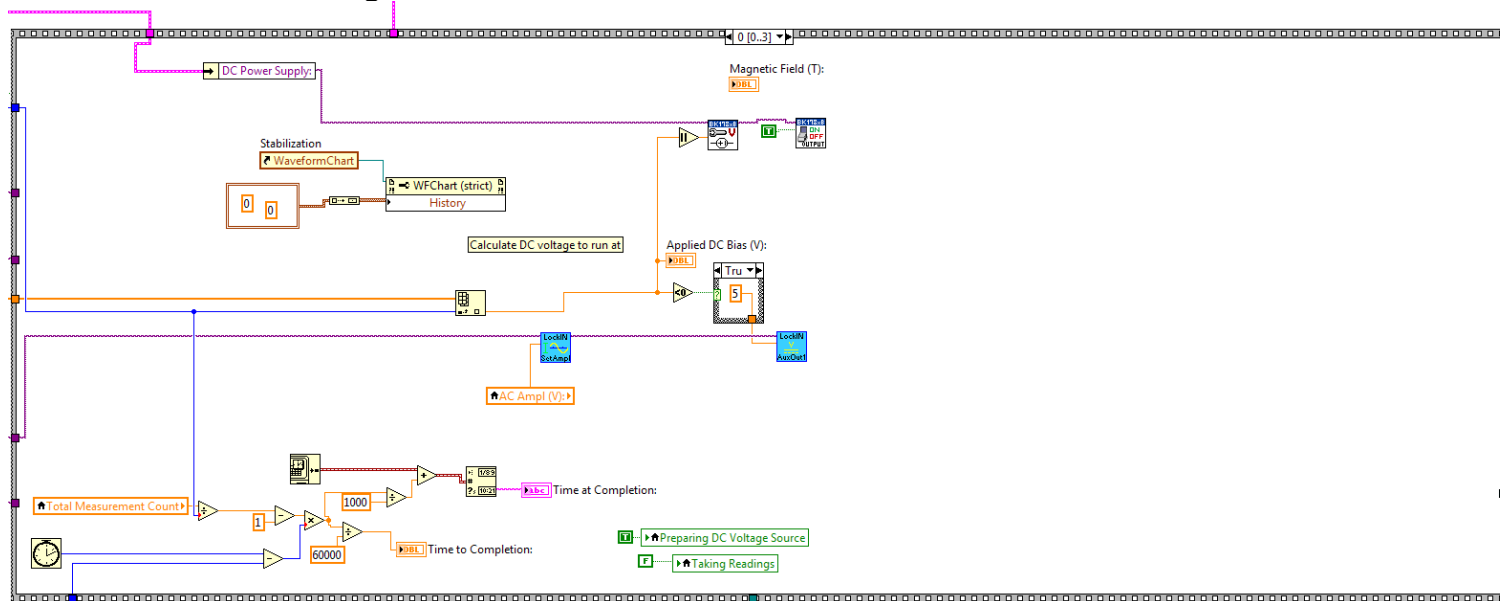


This part of the code is largely self-explanatory. It configures the LIA to the frequency and amplitude of the desired signal waveform, and configures the trigger so that it will take data at the correct frequency, relative to the integration time.

Also, the trigger for the Agilent DC Voltmeter is also configured.

After this step, the program enters the Measurement Loop. For each DC voltage that needs to be probed, it performs the steps necessary to measure the electrical properties requested.

IV.3.H – Set DC Voltage



Configure the DC Supply to output the correct voltage.

For each voltage, we first clear the stabilization chart at \mathcal{A} . We then calculate the time remaining in the measurement, based on how much time was required to get the data points already completed, at \mathcal{B} .

It turns out the DC power supply cannot output negative voltages. To compensate, an electrical relay was constructed⁵ to flip the polarity of the pins from the supply output. The relay is controlled by an output on the LIA, at \mathcal{C} .

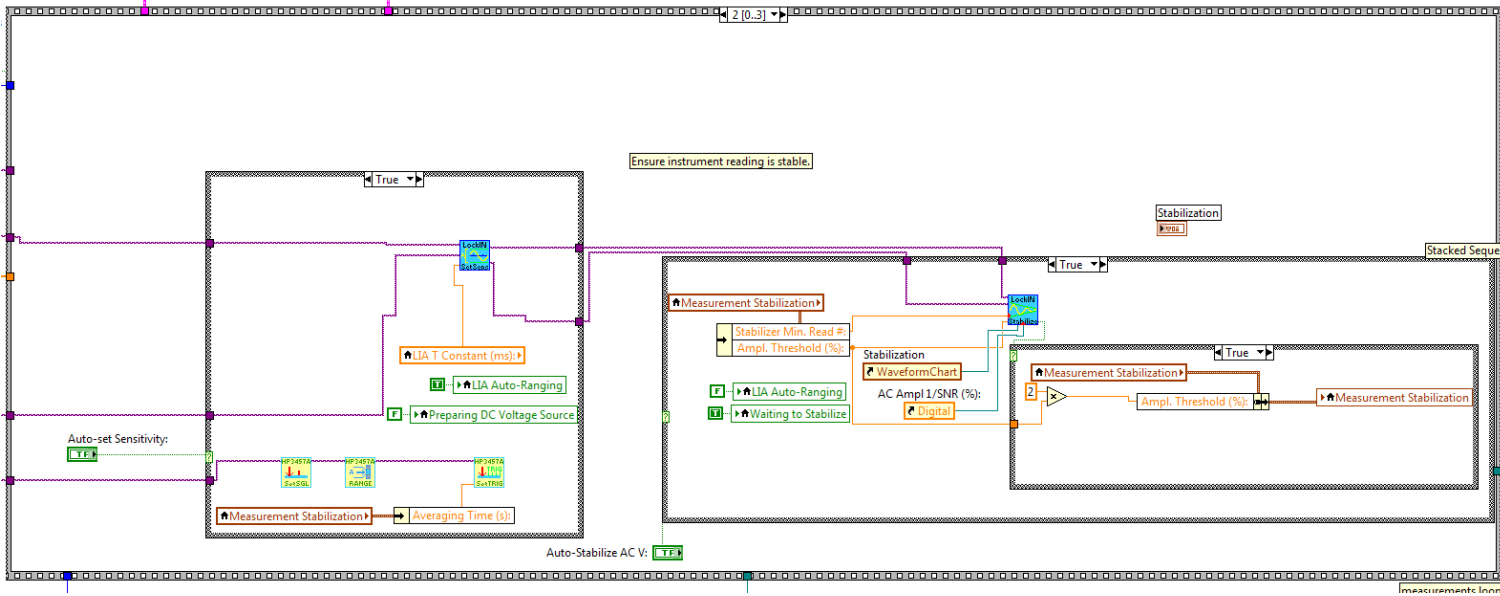
Finally, the supply is commanded to output the correct voltage, and is turned on at \mathcal{D} .

IV.3.I – Wait

The computer waits for the signal to stabilize before adjusting instrument sensitivity.

⁵ Construction by Phil Russ, who, at the time of this writing, is a PhD. Candidate at the University of Illinois at Urbana-Champaign.

IV.3.J – Auto-Gain and Auto-Stabilize



Sensitivity on the LIA and the HP Voltmeter are automatically adjusted.
The LIA signals are then monitored until the signal is stable.

The LIA and HP Voltmeter are then auto-adjusted for sensitivity. The adjustment algorithms essentially takes a reading, and compares it to the current range. If it is $< 10\%$ or $> 90\%$ of the upper limit, then the sensitivity is adjusted to the lowest level where the reading is smaller than the limit. This is repeated, until no adjustments are necessary.

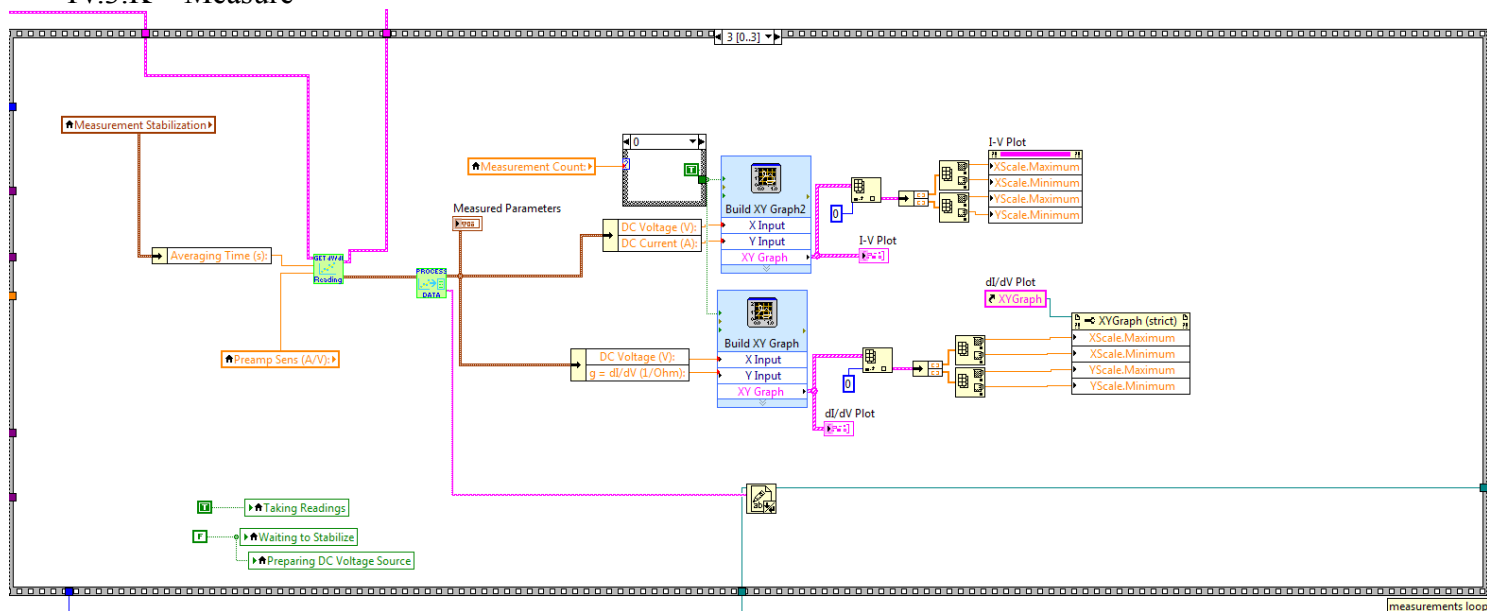
\mathcal{A} applies this algorithm to both LIA, while being careful to take readings at least 1 integration time apart. \mathcal{B} applies this to the HP Voltmeter. The first subVI here readies the instrument for single readings, while the third reverts it to taking multiple readings per trigger.

\mathcal{C} is the auto-stabilize algorithm. It works by monitoring the standard deviation of the last N readings (N is set by the user). Once it falls below a threshold, the signal is deemed stable, and the subVI quits.

If the stabilization is taking too long ($> 60s$), then the stabilization code will auto-abort, and the threshold will be doubled, so that future measurements will not take forever. This occurs at \mathcal{D} .

The auto-stabilize and auto-gain functionalities can be disabled, in which case nothing happens here.

IV.3.K – Measure



Take the readings.

With the signals stable, it is time to take the reading. \mathcal{A} is responsible for communicating with the instruments, triggering the readings, and averaging the readouts.

The averaged quantities are sent to \mathcal{B} , where the conductance and resistance are calculated. The measured quantities are also converted to an ASCII string, where they are eventually saved to a datafile.

The relevant quantities are then picked out and plotted at \mathcal{C} , and the code at \mathcal{D} auto-scales the plots, so that the full plot is shown.

After this step, this data point is measured. The program then loops around to take the next data-point, or, if the measurement is complete, it will search for additional parameters for runs in the queue (assuming it was running from the queue in the first place). If the queue is empty, or if the program is not running from the queue, it beeps, closes the measurement data file, and returns to the event loop.

This completes the description of the program's algorithm.










Differential Conductance Measurement Apparatus Technical Manual

IV.4 – LIST OF SUBVI

Here, we will attempt to list all SubVI written for the control program, for reference. Before we begin, however, we shall discuss some conventions.

IV.4.A – SubVI Conventions

LabVIEW is graphical, so whereas in C, naming conventions such as Hungarian spelling is used to aid readability, in LabVIEW, graphical aids are used instead. First, SubVIs are classified according to their purpose. SubVI of each type will then have its unique color scheme, for easy identification. Second, a short description is written on the SubVI label, and a descriptive symbol is usually drawn in.










<i>ID</i>	<i>Dealing With</i>	<i>Background Color</i>	<i>Text Color</i>	<i>Ex.</i>
1	Queue	Orange	Black	
2	Temperature Controller	Yellow	Black	
3	Programming Functions	Green	Blue	
4	Magnetic Field	Light Blue, with Blue & Red Stripes	White	
5	File-Handling	Aqua	Black	
6	Lock-In Amplifiers	Aqua	Black	
7	Agilent Multimeter	White, with Brown Stripe	White	
8	HP Multimeter	Yellow	Green	
9	DC Power Supply	White, with Blue Stripe	White	

Having established this list, we shall now attempt to list all SubVIs. Note that some SubVIs will be used inside others, and these we attempt to identify as Low-Level SubVI. These are not necessarily easy to find, however, so not all will be identified.

Also, the control software for the BK 178xB series DC Power Supplies, as well as the Agilent 34401A are already written, by the manufacturer. These software are directly downloaded from the National Instruments Developer Zone, and no description will be provided for these SubVIs.

IV.4.B – Magnet Control SubVI










.Magnet Control SubVI

<i>ID</i>	<i>Symbol</i>	<i>Name</i>	<i>Description</i>	<i>Low Level</i>
1		<i>TCP Open Port</i>	Opens connection to Magnet Controller	Y
2		<i>TCP Close Port</i>	Closes connection to Magnet Controller	Y
3		<i>Send Command</i>	Sends an ASCII command to the Magnet	Y
4		<i>Query</i>	Send a command, then listen for the string response	Y
5		<i>Get Current Field</i>	Queries the magnet controller for current magnetic field	N
6		<i>Get Status</i>	Gets current status. Consult the manual for the command “State?” for full list of responses	N
7		<i>Get Quench Status</i>	Queries for quench status. 0 for operational, 1 for quenched.	N
8		<i>Start Ramp</i>	Start ramping the magnetic field to the setpoint.	N
9		<i>Pause Ramp</i>	Stop ramping the magnetic field. Keep field constant at its current value.	N
10		<i>Get Target Field</i>	Queries for magnetic field setpoint.	N
11		<i>Set Target Field</i>	Set the magnetic field setpoint.	N
12		<i>Get Segment Ramp Rate</i>	Queries for ramp rate for any given segment number.	N
13		<i>Set Segment Ramp Rate</i>	Set the ramp rate for any particular segment.	N
14		<i>Get Field Units</i>	Queries for current units for the field. Returns an integer for either kGauss or Tesla.	N
15		<i>Set Field Units</i>	Set the magnetic field unit to either kGauss or Tesla	N
16		<i>Get Ramp Time Units</i>	Queries for units for the time. 0 for seconds, 1 for minutes.	N
17		<i>Set Ramp Time Units</i>	Set units in ramp time.	N
18		<i>Wait Until Setpoint Reached</i>	Wait until the magnetic field is stable, at the setpoint.	N

Differential Conductance Measurement Apparatus Technical Manual

IV.4.C – Temperature Controller SubVI

.Temperature Control SubVI




<i>ID</i>	<i>Symbol</i>	<i>Name</i>	<i>Description</i>	<i>Low Level</i>
1		<i>Read From Temperature Controller</i>	Reads a response from the temperature controller. No query is sent.	Y
2		<i>Write To Temperature Controller</i>	Sends a string to the temperature controller.	Y
3		<i>Query Temperature Controller</i>	Queries the temperature controller for one of several parameters	N
4		<i>Set Target Temperature</i>	Sets the target temperature for the temperature control loop.	N
5		<i>Set Heater Manual Power</i>	Set heater to a constant power level. PID control is disabled.	N
6		<i>Set Heater Power Range</i>	Set heater to one of three power ranges. Power level commands are defined relative to these.	N
7		<i>Set PID</i>	Set the PID coefficients for the control loop.	N
8		<i>L311 Wait Until Near Setpoint</i>	Waits until the temperature is fluctuating near the setpoint.	N
9		<i>L311 Wait Until Stable</i>	Waits until temperature fluctuation is smaller than some threshold.	N

IV.4.D – Agilent 34401A Multimeter Customized SubVI

Agilent 34401A is a widely used multimeter, and it has a superb set of control drivers already written and published on the National Instruments Developer's Zone. These drivers are used in our programming.

A few esoteric commands are implemented using customized SubVIs. Only these are documented here. These deal with ways in which the multimeter can be sent to take N sequential datapoints, store them to memory, and later recall them.

Agilent 34401A Multimeter Control SubVI

<i>ID</i>	<i>Symbol</i>	<i>Name</i>	<i>Description</i>	<i>Low Level</i>
1		<i>Set Multi Sample</i>	Configures the multimeter for multiple readings after a single trigger.	N
2		<i>Trip Trigger</i>	Triggers the multimeter for measurement. Readings taken according to <i>SetMultiSample</i> .	N
3		<i>Agilent Fetch</i>	Fetches the data previously stored in the multimeter's memory.	N

Differential Conductance Measurement Apparatus Technical Manual









IV.4.E – HP 3457A Digital Multimeter SubVI

Unlike the Agilent 34401A, the HP3457A is a very old instrument. While it can achieve an extra digit of precision than the Agilent multimeter, its operation is slower and very strange, compared to the modern standard.

This instrument is strange in that it has little autoranging capabilities in remote operation mode (at least in the way that we use it here), and it behaves very differently when read one measurement at a time (although setting it to take multiple readings after one trigger is similar to the Agilent 34401A). In single reading mode, it is triggered whenever the computer GPIB bus is set to “listening” mode – as opposed to most instruments, which respond to some fixed command. Whereas most instruments gives a reading after a verbal command, this one only responds to the computer looking at it weird.

Still, a set of drivers are written for this instrument:

HP 3457A Digital Multimeter Control SubVI
















<i>ID</i>	<i>Symbol</i>	<i>Name</i>	<i>Description</i>	<i>Low Level</i>
1		<i>Init</i>	Initializes the instrument to measure DC Voltage.	N
2		<i>Set Range</i>	Sets the DC voltage range to a specific level	Y
3		<i>Autorange</i>	Configure the DC voltage range for the reading.	N
4		<i>Set Single Reading</i>	Sets the instrument to return 1 reading at a time.	N
5		<i>Get Single Reading</i>	Takes one single reading from the instrument.	N
6		<i>Set Trigger</i>	Sets the instrument to take multiple readings after a trigger signal	N
7		<i>Get Memory Reading Count</i>	Gets the number of readings stored in instrument memory.	N
8		<i>Fetch</i>	Gets all readings stored in memory.	N

IV.4.F – SR 810 Lock-In Amplifier SubVI

Please note that there are a few pieces of now-decommissioned code that may not function as advertised.

A difficulty arises with the sensitivity settings here. Like the voltmeters, the LIAs only measure signals below some maximum sensitivity level. This can be automatically adjusted by the program. However, in order to do so, the LIA has several preset sensitivity levels, and each of these has a unique index (see manual for the command “SENS”). A few functions are devoted to dealing with converting these indices to actual voltage limits.











SR810 Lock-In Amplifier Control SubVI

<i>ID</i>	<i>Symbol</i>	<i>Name</i>	<i>Description</i>	<i>Low Level</i>
1		<i>LockInInit</i>	Initialize the lock-in amplifier.	N
2		<i>SetOutputAmplitude</i>	Set the LIA output signal waveform's amplitude.	N
3		<i>SetOutputFreq</i>	Set the LIA output signal waveform's frequency.	N
4		<i>LockInSetPhase</i>	Set the current phase offset for the LIA to 0	N
5		<i>SetAuxOut</i>	Set the auxiliary voltage output port #2 to a particular voltage.	N
6		<i>GetSensitivityIndex</i>	For a upper limit in amplitude, find the index to the sensitivity setting just above this limit.	Y
7		<i>LockInReadSensitivity</i>	Query the LIA for the current sensitivity used, in volts.	N
8		<i>ReadTimeConst</i>	Read the integration time constant.	N
9		<i>AutSetSensitivityFor2Inst</i>	Sets the sensitivity for two SR810 LIAs simultaneously.	N
10		<i>SetSensitivityToIndex</i>	Set LIA gain sensitivity settings to a particular level. Level is in index units (see manual)	N
11		<i>LockInWaitUntilStable</i>	Waits until stdev of the last N measurements in the LIAs are both below a threshold.	N
12		<i>LockInAmpRead</i>	Read the amplitude and phase of the signal.	N
13		<i>LockInReadPhase</i>	Read the phase in the AC signal.	N
14		<i>ConfigTrigger</i>	Configure the trigger to take multiple readings some time apart after a trigger signal.	N
15		<i>FetchData</i>	Download all stored data from memory. A separate version of this exists for the SR830	N

Differential Conductance Measurement Apparatus Technical Manual



IV.4.G – Queue SubVI

Queue Utilities SubVI

<i>ID</i>	<i>Symbol</i>	<i>Name</i>	<i>Description</i>	<i>Low Level</i>
1		<i>InitQueue</i>	Initializes the queue to contain an empty array of strings.	N
2		<i>CleanUpID</i>	Sorts the ID list in the queue so IDs are ascending with no gaps.	Y
3		<i>SelectRowFromQueue</i>	Selects a row, identified by its index.	Y
4		<i>GetQueueSize</i>	Gets the number of rows in the queue	Y
5		<i>SwapQueue</i>	Swap two given rows in the queue.	Y
6		<i>InsertIntoQueue</i>	Insert a set of parameters, defined from the primary UI, into the queue.	N
7		<i>LoadFromQueue</i>	Load a set of parameters from the queue into the primary UI.	Y
8		<i>DeleteRowFromQueue</i>	Deletes the selected row from the queue	N
9		<i>LoadQueueFromDisk</i>	Load the queue from an ASCII file stored on disk. Written without the B-field in mind!	N
10		<i>SaveQueueToDisk</i>	Save the queue from an ASCII file stored on disk. Written without the B-field in mind!	N

IV.4.H – File-Handling SubVI



















File Handling SubVI

<i>ID</i>	<i>Symbol</i>	<i>Name</i>	<i>Description</i>	<i>Low Level</i>
1		<i>FindNextRunNumber</i>	Find the next available run number in the current directory	N
2		<i>MakeFileName</i>	Generate filenames for data files.	N

IV.4.I – General Programming SubVI

Certain blocks of code were placed into SubVI for readability and ease of maintenance. Here we list these SubVI.

General Programming SubVI

<i>ID</i>	<i>Symbol</i>	<i>Name</i>	<i>Description</i>	<i>Low Level</i>
1		<i>AutoRampMagneticField</i>	Ramps magnetic field to a target while recording 0-bias conductance	N
2		<i>CheckInstRangeIsGood</i>	Returns 1 if the measured value is within 10 – 90% of a limit.	N
3		<i>CloseAllInst</i>	Close connections to all instruments	N
4		<i>FindAppliedVoltage</i>	Perform the applied voltage interpolation algorithm (see III.4.C)	N
5		<i>CheckInstRangeIsGood</i>	Returns 1 if the measured value is within 10 – 90% of a limit.	N
6		<i>CloseAllInst</i>	Close connections to all instruments	N
7		<i>FindAppliedVoltage</i>	Perform the applied voltage interpolation algorithm (see III.4.C)	N
8		<i>Format6DataIntoLineString</i>	Formats a series of 6 floating-point numbers into a tab-delimited string.	N
9		<i>FormatDataIntoLineString</i>	Formats a series of 4 floats into a tab-delimited string.	Y
10		<i>GenerateVoltageList</i>	Generates an array of numbers, from Begin to End, with spacing Delta.	N
11		<i>GetOneReading</i>	Take 1 reading from every instrument. Written to acquire ramp data.	N
12		<i>InitAllInst</i>	Initialize all instruments for operations.	N
13		<i>InitInstrumentsForRamp</i>	Initialize the instruments for measurements while ramping T or B-field.	N
14		<i>MakeRampDataString</i>	Takes the floating-point readings from T or field ramping and reformat into string.	N
15		<i>Measure DC Voltage</i>	Measures the voltage from the Agilent voltmeter. Includes averaging algorithm.	N
16		<i>MeasureTemperature</i>	Reads temperature from the temperature controller. Includes averaging.	N
17		<i>ProcessesData</i>	Processes data from dI/dV reading into a tab-delimited string. Calculates conductance.	N
18		<i>TakeReading</i>	Takes readings from every instrument for the dI/dV measurement. Uses trigger/fetch method	N

V. APPENDIX: COMMUNICATING WITH INSTRUMENTS

Thanks to some of the features of the LabVIEW coding environment, communicating with instruments is fairly straightforward. Depending on the computer and the instrument, there are four major interfaces connecting to an instrument:

- **GPIB Port:**
Oldest of the interfaces, this is still in common use among laboratory equipment. This system is very robust, with good error-handling capabilities. Several instruments can be daisy-chained together to a single GPIB port on a computer, and each instrument will have its own unique “address” – a number that identifies the instrument on the network. Few parameters need to be configured in a GPIB interface, and it is easy to get this working. However, the cable is very bulky, and the GPIB card is relatively expensive (about \$600 per card).
- **RS-232 Serial Port:**
RS-232 was a standard way for computers to communicate with external peripherals before the Universal Serial Bus (USB) replaced it. Today, some computers still contains a RS-232 port for backwards-compatibility, and it is still used as a standard interface for scientific instruments. Implementations of this interface varies between instruments, however. Be ware that both the computer and the instrument must be aware of the exact settings used on the interface – baud rate (ie. transmission speed), handshaking protocol, parity, etc. Consult the instrument manual and configuration menu for exactly how this is set on each instrument. Also, pay particular attention to the “termination” character used in each instrument. More on this later.
Data transmission over RS-232 is slightly more prone to errors than the GPIB Port, especially in the absence of handshaking, although this is rarely a concern except when transmitting numbers in pure binary format (ie. not as ASCII strings). However, RS232-to-USB adaptors are fairly cheap, and serial cables relatively lightweight (plus cables are easily made by hand, compared to GPIB cables, which are very complicated). By comparison to the GPIB Bus, each instrument gets a unique (hardware) COM Port instead of a software address.
- **Ethernet TCP/IP Protocol:**
Newer and fancier instruments may have the option of connecting the instrument to the computer via the Ethernet port, through the TCP/IP Protocol. This allows the instrument to plug right into the router/hub/switch (but ensure the IP and port settings are correct).
When using multiple instruments over Ethernet, it is possible to plug all instruments into the same hub/router as the computer, and to configure all IP addresses so that there are no conflicts. This makes it very convenient and inexpensive to connect to multiple instruments, as both ethernet cables and hub/routers are common items. The disadvantage is that only newer and fancier instruments features an Ethernet port.
- **USB:**
Newer instruments sometimes connect via USB. USB is actually a very complicated bus, and often it is necessary to install low-level driver software for the computer to identify the instrument.

V.1 – NI VISA

Because of the many ways an instrument can be hooked up to a computer, LabVIEW created NI VISA. Effectively, NI VISA *encapsulates* the many underlying protocols into a single interface, so that all instruments can be treated the same, and only a NI VISA resource needs to be selected at runtime to specify the instrument being used.

Furthermore, LabVIEW can automatically scan a system for all connected instruments. Connections over Serial Ports and GPIB Ports can often be auto-detected, and instruments connected via Ethernet can be added in the *Measurements and Automation Explorer (MAX)*, which comes with LabVIEW.

In practice, a general VISA session needs to be *opened*, *used*, and finally *closed*. Configuration settings are often defined when the resource is *opened*. Not closing a VISA connection will lead to it being unusable in other applications later (such as HyperTerminal). Usually the ports are closed when the program quits, not earlier.

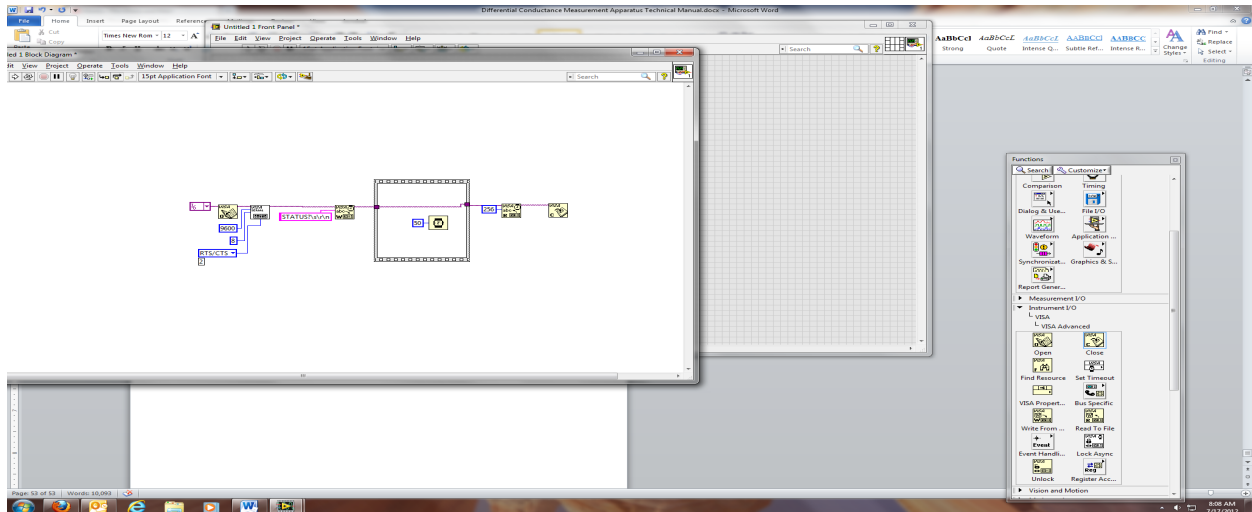
V.2 – CONNECTIONS IN SPECIFIC PROTOCOLS

V.2.A – RS-232 Serial Ports

Serial ports are inexpensive (in both the PC interface card and the cable), but is slower and less reliable than GPIB or Ethernet. Each instrument is connected on a separate DB-9 connector, which are very common.

Configuring for RS-232 is trickier than some of the other protocols. Both the instrument and the computer must know the exact configuration to use when communicating – akin to people picking a language before they speak to each other. These parameters include baud rate, parity, handshaking, etc.

Below, as an example, we have opened a VISA resource, then configured the relevant parameters in the first SubVI. We then sent a command to the device, waited for 50ms, and read the device's response. Finally, we closed the port.



The port connection settings can often be adjusted in the instrument's setup menu. Furthermore, usually a few of the settings cannot be adjusted, in which case they should be listed in the instrument's manual.

Differential Conductance Measurement Apparatus Technical Manual

Another thing to notice is the *termination character*. Commands to instruments are often ASCII strings, and it is sometimes difficult for the instrument to know when a command has ended. As such, all commands should be ended by *termination characters*. These are usually the *carriage return character* '\r', the *newline character* '\n', or a combination of both (such as \r\n). These special characters essentially refer to the user pressing *Enter* on the keyboard. Refer to the manual for the exact configuration, or if it is not known, try different combinations until one is found to work. Often, if the termination character is incorrect, the device will give no response.

When defining the command string, be sure to enable '\ Codes Display, so that the newline characters will not be understood literally as a slash, then an 'n'. Furthermore, ensure the slash is oriented correctly.

V.2.B – GPIB Ports

GPIB Ports almost need no configuration – one could simply specify the instrument address (user-adjusted in the instrument's setup menu), open the port and communicate. Termination characters are usually handled by the bus and LabVIEW, and the user won't have to worry about it. Remember to close the port when the program quits.

V.2.C – Ethernet Ports

Ethernet ports are like GPIB ports – a minimum of configuration is necessary, but since the instrument is not connected on a specific card, it is often necessary to add the IP address of the instrument prior to use. To do this, open the *Measurement and Automation Explorer*, and create a new device by right-clicking the Remote Systems icon. Here, the IP address of the device will need to be automatically entered. It is helpful to give each instrument a descriptive name. Thereafter, the user can always select that resource from the user interface of any LabVIEW program, and the IP address will no longer need to be entered.

Sometimes one has to pay attention to termination characters with these instruments.

V.3 – GENERAL PROCEDURE FOR CONNECTING TO DEVICES

Many devices will have *drivers*, or sets of SubVI's sending various commands, already implemented. They are usually published on the *National Instruments Developer Zone*. Simply search by manufacturer and model.

If a device does not have a good set of drivers already written, it is necessary to write them. It is helpful to first of all attempt to get the correct configuration settings to the device. Attempt to send a simple query command to the device, and read the response. A common command is the Identify Device command **IDN?* implemented across most instruments. Use the instrument's setup menu to set as many parameters as possible to known values, and pick the rest off from the manual. Guess the rest. This is the trickiest part of the process.

Once a response can be read, the correct configuration settings are known. Now it is a simple matter of writing the different SubVI's required to operate the instrument. Focus first of all on writing a SubVI to write a command to the device. This could be as simple as, for example, wrapping the command with termination characters. Then, write a command that will read a string from the device. Also, if specific configuration settings or initialization commands are needed, write VIs for opening, initializing and closing ports.

Finally, it is a simple matter of writing different commands to the device. Use the "Fundamental VI's" above, and write different commands to perform commonly needed tasks. That's it!