Design of an Automated Experimentation and Data Processing Software Suite for the ADiR Sensor

A Major Qualifying Project

Submitted to the faculty

of

Worcester Polytechnic Institute

Worcester, Massachusetts, USA

In partial fulfilment of the requirements for the

Degree of Bachelor of Science

on this day of

October 13th, 2007

by

____

Ryan Hollister

____

Rachelle Horwitz

____

Mandela Kiran

Abstract

Analog Devices, Inc., Limerick, Ireland, has developed an infrared sensor for non-contact thermometry applications. We developed a suite of software that automated and expedited testing and data processing procedures for four experiments conducted on the sensor. For the pressure and qualification experiments, we reduced the human-machine interaction time by 92% and 90%, respectively. The experimentation software we created for the angular response and lens focusing experiments enabled engineers to conduct experiments that had previously been impracticable.

Acknowledgements

We would first like to thank Analog Devices Inc., Limerick, for providing us with a unique opportunity to work with an upcoming technology. We would like to extend our heartfelt gratitude toward John Reidy for coordinating the project and for helping us settle in the company. We would also like to thank Brendan Cawley and Eamon Culhane for sharing their knowledge of the detectors and for continuously providing us with detectors to work with and the resources necessary to evaluate them for analysis and implementation. Lastly, we would also like to extend our thanks to Claire Leahy for her support in the software development process.

We would further like to thank Luke Pillans from the Newbury, UK, office for his dedication to the project and his interest in our learning and our project outcomes. Without his knowledge in infrared, we would not have been able to automate the experimentation software to obtain more repeatable data from which ADI can draw accurate conclusions. Furthermore, through sharing his own work, he was able to guide us toward a better understanding of the detector characteristics so that we could come up with a more intuitive software package.

We would next like to thank our advisors Professor Rick Vaz and Professor Rick Brown for their continuous advice and assistance through the entire duration of this project. They provided valuable assistance in setting up the goals for our project and in determining the best way to organize and display the abundance of data obtained.

Finally, we would like to thank Charlotte Tuohy, the local coordinator for the WPI Limerick Project Center, for providing us with the necessary information to make our stay in Ireland comfortable.

Table of Contents

Table of Figures

Executive Summary

Between 2002 and 2003, SARS had afflicted thousands of people throughout the world. Since it was highly contagious and doctors needed a way to measure patients' body temperatures, they used non-contact thermometers, which quickly and accurately displayed the patients' body temperatures.

Since the SARS epidemic, non-contact temperature sensing has become increasingly ubiquitous because it is inexpensive, fast, and accurate. Unlike contact temperature sensors, non-contact temperature sensors can provide the temperature of moving, hazardous, and physically inaccessible objects. Many non-contact temperature sensors are based on the principles of infrared (IR) thermometry, which is the measurement of the IR radiation that is emitted by all objects. Since the amount of IR radiation emitted is directly proportional to temperature of the object emitting the radiation, a sensor that detects the power of the radiation can also detect the temperature of the object.

Analog Devices, Inc. (ADI), a company that specializes in analog, mixed-signal, and digital signal processing ICs has designed its own cost-effective, high performance sensor that measures infrared (IR) radiation. ADI's IR sensor, also known as the ADiR sensor, is unique because when it is manufactured with a bolometer and enclosed in a vacuum, it is more sensitive than the current IR sensors available on the market. The ADiR sensor can also be based on thermopiles, which are heat transducers that most companies use in their IR sensors. For both types of heat transducer, ADI has developed sensors of three different sizes and spatial resolutions. Depending on the specific application, one type of sensor technology may be more appropriate than the other.

Before our project began, ADI possessed software to communicate with some of the testing equipment. However, some of these modules required modifications to enable them to operate properly and most efficiently. There were some pieces of equipment for which ADI completely lacked automation software modules. The benefits of automated experiments include faster tests, more repeatable results, and no need for employees to monitor the experiments.

The main goal of this project was to facilitate the experimentation on the ADiR sensor by developing a suite of software modules that would automate several testing and data processing procedures. To accomplish this goal, we manually conducted experiments on the ADiR sensor, and used LabVIEW, Excel, and Visual Basic to ultimately create the automation software suite. We initially performed the experiments manually to gain experience with the equipment and to minimize the number of glitches in the software. After repeatedly performing the experiments, we investigated ways to improve the display of the results and identified areas where software could expedite the experiment. Finally, we created the suite of automated software, which yielded faster and more repeatable results.

Our first objective was to understand the operation and properties of the four types of sensors: 3x3 and 11x11 thermopiles and bolometers. We accomplished this by performing two experiments: pressure and vacuum degradation experiments. The purpose of the pressure experiment was to confirm that the sensors become more sensitive at near-vacuum pressures. ADI can also use the results from our pressure experiment to improve their theoretical models that predict the behavior of different sensors at various pressures. We conducted the vacuum degradation experiment to compare the changes in the behaviors of bolometers and thermopiles as the vacuum inside the sensors degrades. We were particularly interested in comparing the sensors' baselines offsets and sensitivities to changes in temperature before and after we stored the parts at different temperatures for one week. We found that bolometers are more sensitive to changes in pressure than thermopiles are.

Our second objective was to automate the software for the pressure experiment. When we conducted the pressure experiment before we automated it, we spent approximately five minutes per part setting up the experiment and sixty minutes per part conducting the test. To conduct the test, we would manually set the radiation source to a specified temperature, adjust the pressure in the vacuum chamber, prevent the sensor from being exposed to radiation for a few seconds, subsequently re-expose it to radiation, and gather data from the sensor. We repeated this process approximately thirty times per part. After we automated the pressure test, we still spent five minutes per part setting up the experiment, but it was unnecessary for us to spend sixty minutes conducting the experiment because the software autonomously carried out the test.

Our third objective was to conduct an angular response experiment to determine whether the diffractive pattern on a lens would enable the ADiR sensor to detect motion. This required that we develop LabVIEW software to automatically move a rotational stage through 180^{o}. After we created the software and conducted the experiment, we found that when the diffractive pattern was on the lens, the center pixel was more responsive than the others, and after we removed the diffractive pattern, each of the pixels responded uniformly. Thus, we concluded that the lens enabled the sensor to detect motion more adequately.

Our fourth objective was to create software that moves the sensor to the point in three-dimensional space where it receives the highest signal when exposed to radiation. We accomplished this by creating LabVIEW virtual instruments (VIs) to move a three-axis translational stage and simultaneously take readings from the sensor. In the first version of this software, we moved the stage to 512 points in space and sampled the sensor's readings ten times at each point. Since this comprehensive scan required three hours to complete, we developed a more efficient algorithm that utilizes input from the user regarding the focal length to scan a smaller area. This required less time than the comprehensive scan. Due to time constraints and hardware issues, we have not been able to obtain repeatable results with this algorithm.

Our final objective was to create software in Visual Basic that could be imported into Excel to graph data obtained from a test that separates properly functioning parts from non-functional parts and characterizes the working parts. Instead of the users manually importing the data into Excel and generating seven graphs, the users simply select the part for which they want to generate the graphs. Before our software, manually generating multiple graphs for one part required ten minutes. Our software allows users to generate graphs for each part in less than thirty seconds.

We developed several recommendations that ADI can use to improve their experiments. The two most pertinent recommendations were that ADI cover the sensors while they are being tested to prevent them from breaking, and that ADI improve the test setup by using more reliable equipment. If ADI follows these recommendations, there will be a reduced need to repeat experiments due to inaccurate equipment or a broken part.

We hope that by accomplishing our five objectives, we have aided ADI in introducing the ADiR sensor to the market in the near future. Equipped with software that allows for faster, more repeatable experiments that test the ADiR sensor, ADI can obtain data more efficiently. Since the abundance of data gathered from the experiments we automated can be used to further enhance the ADiR sensor, ADI can commercialize the ADiR sensor in less time.

1. Introduction

Advances in technology have enabled non-contact temperature sensing to become faster, less expensive, and more ubiquitous. Non-contact temperature sensing is useful in numerous applications that range from human body thermal imaging to motion detection. This technology is especially useful because it can measure the temperature of a moving, hazardous, physically inaccessible, and/or hot object. For example, during the SARS outbreak, doctors and nurses used non-contact thermometers to conduct fever screening on patients without subjecting themselves to the SARS coronavirus (Ng, 2005). Analog Devices, Inc. (ADI), a company that designs and manufactures signal processing-related ICs, has designed its own cost-effective, high performance sensor that measures infrared (IR) radiation.

ADI's IR sensors, also known as ADiR sensors, are based on two different technologies, and for each technology, ADI has developed sensors of three different sizes and spatial resolutions. Depending on the specific application, one type of sensor technology may be more appropriate than the other. Although ADI has designed and prototyped these six different varieties, they have not been able to conduct extensive experiments to determine the characteristics and capabilities of each of the six varieties, such as sensors' responses to varying pressures and their use in motion detection.

ADI uses a graphical-based programming language called LabVIEW to run the experiments and gather data. ADI chose LabVIEW because it provides the flexibility and rapid prototyping needed for controlling different equipment. Before our project began, ADI possessed LabVIEW modules to communicate with some of the equipment. However, some of these modules required modifications to enable them to operate properly and most efficiently. There were some pieces of equipment for which ADI completely lacked automation software modules.

Ideally, all of the experimenting and data processing procedures would be automated, as automated experiments are more repeatable and often require less time. When the project began, ADI had not yet fully automated several of their experimenting and data processing procedures. ADI possessed LabVIEW modules to communicate with some of the experimentation equipment, such as the ADiR sensor itself, which was used in all of the experiments, and the temperature forcer, which was used in the vacuum degradation experiment. However, the module that communicated with the sensor during the vacuum degradation experiment only functioned when one type of technology was under test; it did not work for the other type. ADI completely lacked automation software modules for equipment that was used in three experiments: the pressure, angular response, and lens focusing experiments. Since ADI lacked equipment to communicate with the vacuum controller and shutter during the pressure experiment, the experiment required sixty-five minutes of human-machine interaction time before we developed the automation software and procedure. Before our automation software for the one-axis rotational stage, which was used in the angular response experiment, ADI had used different equipment that did not yield precise results. Since conducting the lens focusing experiment was impracticable before we developed the automated software, ADI had not yet conducted that experiment. ADI also lacked data processing software, which converted raw data into meaningful results. For one test, the qualification test, which separated properly working parts from non-functional parts and characterized the working parts, converting raw data into graphs in Excel required approximately five minutes per part. Since engineers run the qualification test on dozens of parts in one sitting, the engineers spend an excessive amount of time processing the data.

The main goal of this project was to facilitate the experimentation on the ADiR sensor by expediting the testing process and reducing the need for employees to constantly monitor the experiments. We accomplished this first manually by conducting experiments on the ADiR sensor. This provided us with experience with the equipment we would be automating and minimized the number of glitches in the software. After repeatedly performing the experiments, we investigated ways to improve the display of the results and identified areas where software could expedite the experiment. Finally, we used LabVIEW, Excel and Visual Basic to automate the experiments and data processing procedures for five of the experiments. For two of the experiments, the pressure and qualification experiments, we reduced the human-machine interaction time by 92% and 90%, respectively. Our modification to a third experiment, the vacuum degradation experiment, allowed engineers to test two types of technologies instead of one. The experimentation software we created for the angular response and lens focusing experiments enabled engineers to conduct experiments that had previously been impracticable. Our accomplishments will hopefully enable ADI to introduce the ADiR sensor to the market in the near future.

2. Background

As MEMS technology improves, devices utilizing these advancements, such as non-contact IR thermometers, become more commonplace and less expensive. Non-contact thermometry is the measurement of an object's temperature without making physical contact with it. This technology entered the limelight during the SARS outbreak, when doctors and nurses used non-contact thermometer to conduct fever screening on patients without subjecting themselves to the SARS coronavirus (Ng, 2005). Another benefit of this technology is that it can measure objects that have quickly changing temperatures, such as the tires of Formula 1 racing car while they are heating during a race.

The Extech 42500 Infrared Thermometer is an example of a non-contact IR thermometer that costs around $80. It is a handheld device that can be used to measure the temperature of objects, from refrigerators to tires. After the user simply points the device at a target, the device displays the temperature of the target within one second (http://www.extech.com/instrument/products/400_450/42500.html). The following sections provide the technical background necessary to understand how the major components of an IR sensor, specifically the ADiR sensor, operate.

2.1 ADiR Sensor and IC

The ADiR sensor is unique because it is more sensitive than current sensors on the market, since it is manufactured at near-vacuum pressures. Unlike ADI's competitors, who have been unable to seal the IR sensor at near-vacuum pressures, ADI possesses such technology and processes. Since a vacuum is the perfect insulator to conduction and convection, it is used to increase the resistance of the electrical analog shown in Figure 1.

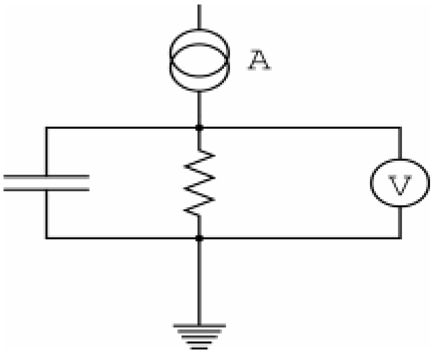


Figure 1: Electrical analog of the detector (Pillans).

This increase in the resistance of the system increases the signal. ADI also has proprietary resistor material that has a high temperature coefficient of resistance (TCR). This high TCR allows the resistors to be more sensitive to temperature changes (Pillans).

Since ADI is a leader in signal processing units, they are able to leverage their existing technology such as the ADT7301 temperature sensor and AD7794 analog-to-digital converter (ADC) to place them ahead of the competition. The IR sensor being designed by ADI integrates the sensor, AD7794, Programmable Gain Amplifier (PGA), fuse memory (ROM), and ADT7301 into one MEMS IR-to-digital converter. The fuse memory on the 3x3 array contains 1kbit memory in thin film fuse memory (TFFM). The memory contains 128 8-bit words that are addressable via a 7-bit address bus and a master fuse that prevents writing to ROM after being blown. ADI plans to release a single sensor ("pixel") device along with 3x3 and 11x11 array variations of this device. The different variations will allow the sensors to penetrate different markets as this report will discuss in Section 2.3. These different variations use the same logic except the arrays of pixels are multiplexed to allow a controller to read only one pixel or scan multiple pixels in the array. The system diagram for the system based on one type of IR transducer that ADI uses, a bolometer, is illustrated below in Figure 2.

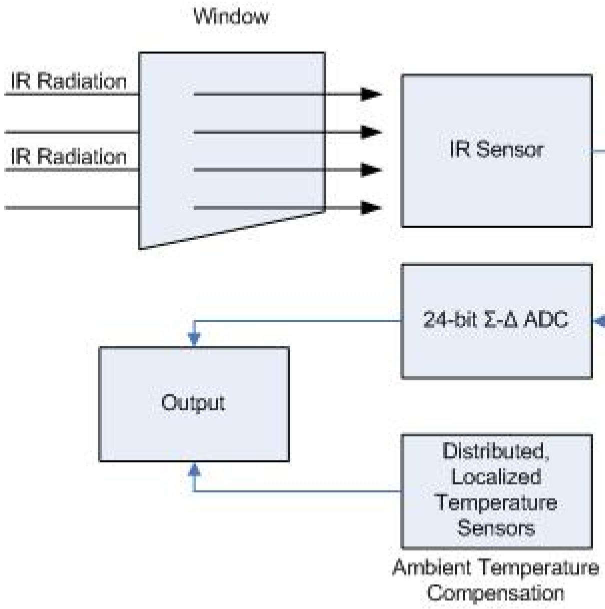

Window
IR Radiation
IR Radiation
IR Sensor
Wheatstone Bridge Arrangement
24-bit Σ-Δ ADC
Programmable Gain Amplifier
Output
Distributed, Localized Temperature Sensors
Ambient Temperature Compensation

Figure 2: ADiR system diagram.

IR is captured by the sensor through a "window" and is converted to a voltage by the Wheatstone bridge arrangement. The "window" that the radiation passes through is a diffractive single step phase lens, which focuses the signal onto the center pixel. The signal is then amplified by the PGA and converted to a digital signal by an AD7994. The ICs also contain a dedicated auxiliary analog input channel, an on-chip implicit reference and a high speed serial peripheral interface (SPI). The sensing elements of the system are thermally isolated from the rest of the devices by using ADI's TMEMS technology. This technology also allows the mixed-signal circuits to share the same silicon as the detector. ADI is targeting two package types; 8-lead TO-39 and 28-lead plastic LFCSP.

2.1.1 Capabilities and Specifications

The IC has three different variations, but there are only two different dies. The single and 3x3 array variations are made from the same die with slightly different register configurations. The Mode Register allows the customer to set the IC to two different single pixel configurations or two different array configurations. The IC can be configured to use only one pixel or be a combined array so that the 3x3 array acts as one big sensor. The remaining two modes treat the sensor as nine individual pixels and allow the user to either scan all nine pixels or only scan one of the three rows. The pixel numbers and arrangement is displayed in Figure 3.

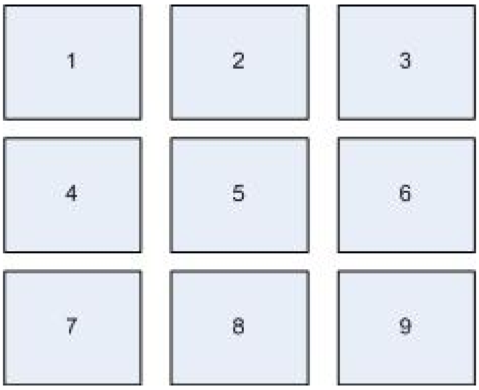


Figure 3: Pixel arrangement.

The IC allows for the pixel configurations described above and also allows for different operating modes independent of the pixel mode. These modes are standby, idle, calibrate, IR-to-digital and voltage-to-digital. When the part is initially powered, it is in standby mode. Normally the user would then calibrate the part by writing to the OP_MODE register. Once the part completes calibration it switches to idle mode and the DOUT pin is driven low to indicate that calibration is complete. From the idle state, the user can then write the OP_MODE register to switch to either IR-to-digital or voltage-to-digital mode. This switch can be made by the user at any time.

To power-down the part, the user can set the part to either idle or standby mode. If the user places the part in idle mode, the calibration data will not be lost and will be read the next time the part is powered. If the user places the part in standby then the calibration data that is stored in the ROM will be lost.

2.1.2 ADT7301 Bandgap Temperature Sensor

The ADT7301 is a 13-bit digital temperature sensor that is accurate to +/- 1°C. This temperature sensor is isolated from the radiation so that it can report a baseline temperature that will be used to determine the level of radiation. The bandgap type of temperature sensing is commonly used in electronics. The bandgap works on the principle that the forward voltage of a silicon diode is temperature-dependant.

Since the ADT7301 is not exposed to radiation, the difference between the temperature reported by the IR sensor and the temperature reported by the ADT7301 can be used to determine the amount of radiation being collected. The ADT is used to gather the baseline reading and is used as an offset of the IR sensor temperature. (ADT7301 Datasheet) This principle explains how the IR sensor functions.

2.1.3 AD7794 24-bit ADC

The AD7794 is a 6-channel 24-bit Sigma Delta ADC with an on-chip amplifier and reference. The chip is used to convert the analog signal from the bolometer or thermopile into a binary stream readable by the computer software. The binary stream is a digital signal effective up to 23 bits. (AD7794 Datasheet)

2.1.4 Configuration of the ADiR Sensor

The ADiR sensor has been developed in three different configurations: single-pixel, 3x3, and 11x11. The users need to decide which configurations are best suited to their applications. These different configurations are discussed in the following sub-sections.

The single pixel configuration can be accomplished in two different ways. One way would be to choose a single pixel from the array and only use the value reported by that pixel. The other option would be to place the sensors into a combined array and use the combined values of all nine sensors. In this mode, the final value is the total sum of all the nine pixels. The user would set the pixel mode by writing to the PIX_MODE[1:0] register which is bits 7 and 6 of the MODE[7:0] register. A value of ‘00’ would place the sensor in a single pixel mode while a value of ‘01’ would put it into a combined array mode.

The other option for the 3x3 sensor array is to use all nine pixels. This is done by multiplexing the outputs of the nine sensors. This mode will be inherently slower since there are more pixels to scan but will allow for greater spatial resolution than the single pixel mode. When using the array configuration, the user will need to choose either the full nine pixel scan or to scan only the middle three (pixels 4, 5 and 6). The PIX_MODE[1:0] will need to be configured accordingly with '10' being a full scan and '11' being only a line scan.

The business case for the 11x11 pixel array is still being investigated, so specific configuration and specification details are not yet defined. Test samples were available from which we collected data. The 11x11 array is very similar to the 3x3 array, except the 11x11 array contains more pixels. Another difference is that the pixels for the 11x11 array are much smaller than the 3x3 pixels. The 11x11 array allows for greater spatial resolution just as the 3x3 array allows for greater resolution than the single pixel. ADI expects that the 11x11 configuration will be very similar to the configuration described above for the 3x3 array.

The IC provides an internal offset and gain calibration routine. To execute the routine, the user writes to the OP_MODE register and waits for the DOUT pin to be driven low, which signals that calibration has been completed. Calibration is assumed to be performed after power-on but it is not required. Calibration calculates the offset coefficients to 24 bits and the individual pixel offset values to 16 bits. The offset value used during pixel conversions is equal to the sum of the calibrated offset value for the system and the scaled pixel offset.

2.2 IR Thermal Sensors

Non-contact IR thermal sensors are based on four laws of physics, which are described in Appendix A. The sensors operate by absorbing IR photons, which raise the temperature of the detector. The change in temperature is then measured as an electrical signal. There are several characteristics that are unique to each type of IR temperature sensor. Five characteristics that change depending on the IR thermal sensor are the transducer technology on which they are based, response time, response to a vacuum, distance-to-spot ratio, and the array size.

Even though all the IR thermal sensors perform the same task, their use in a specific application is based on their individual properties. ADI is building two types of prototypes that use two different technologies: thermopiles and bolometers.

A thermopile is a temperature sensor that is composed of several thermocouple junction pairs connected in series to amplify an induced voltage produced by each thermocouple. A thermocouple is made of two different materials. The junction in the thermocouple where the absorption of thermal radiation takes place is called the “active” or “hot” junction, while the junction that is connected to a voltage meter is called the “reference” or “cold” junction. The temperature at the reference junction must be known before the temperature at the active region can be derived (Kreith, p. 174). Figure 4 depicts a thermocouple.

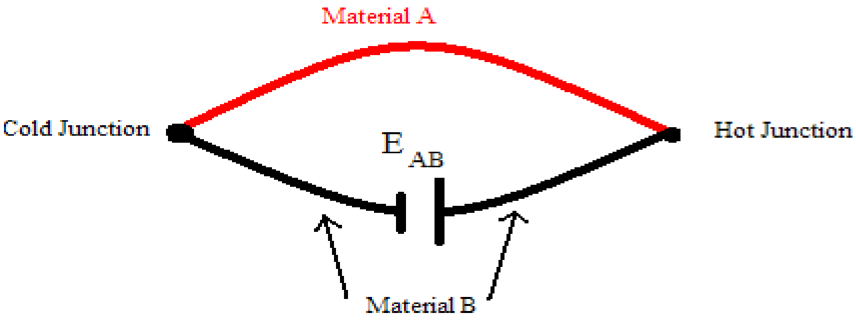


Figure 4: Thermocouple (Weckmann).

Figure 5 illustrates the active and reference regions in a magnified photograph of a thermopile used by ADI.

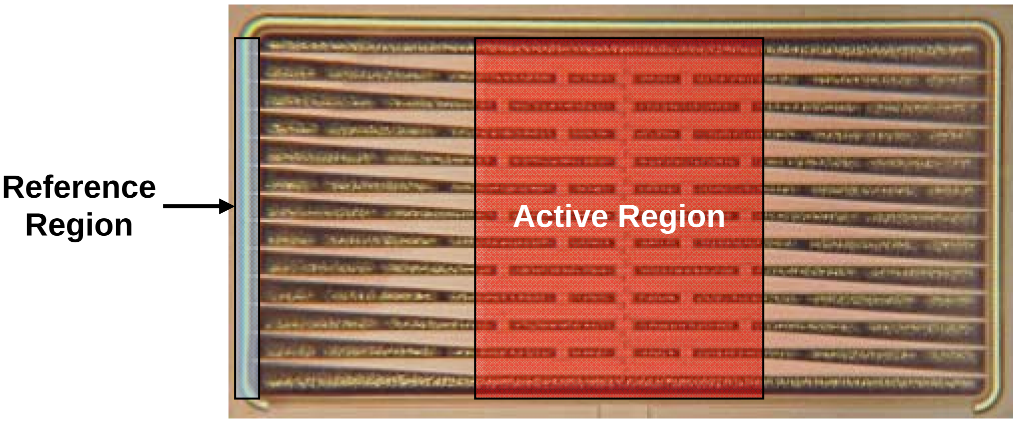


Figure 5: Magnified photograph of a thermopile (Pillans).

ADI is using a metal and a poly-silicon for the two materials, which are connected by the cold and the hot junctions (Pillans). The electromotive force (E_{AB}) between the junctions can be measured, and from this measurement, the difference in temperature between the cold and hot junctions can be deduced (Weckmann).

As a result of the IR radiation hitting the active junction, a differential is created in the temperature of the active and the reference junction, finally producing an electromotive force directly proportional to the temperature differential. This is commonly known as the Seebeck thermoelectric effect. This electromotive force can be used to calculate the change in temperature as shown in Equation 1 (Weckmann).

$$\Delta T = \frac{E_{AB}}{S_{AB}}$$

Equation 1: Electromotive force equation.

where E_{AB} (V) is the electromotive force. S_{AB} is the difference between the Seebeck coefficients of Metals B and A. ΔT (Kelvin) is the difference in temperature. (Weckmann)

Bolometers are resistance thermometers that respond to IR radiation. A typical bolometer comprises a sensing element and a resistor. The incoming IR radiation heats the resistor, causing its resistance to change (Neuzil, 2001). ADI is using bolometers made of resistors with a high temperature coefficient of resistance (TCR) so that they can respond to extreme temperatures. These resistors are placed in a Wheatstone bridge arrangement where two out of the four resistors are exposed to radiation. An external voltage source is applied to calculate a temperature from the measured voltage across the resistors in the Wheatstone bridge (Pillans).

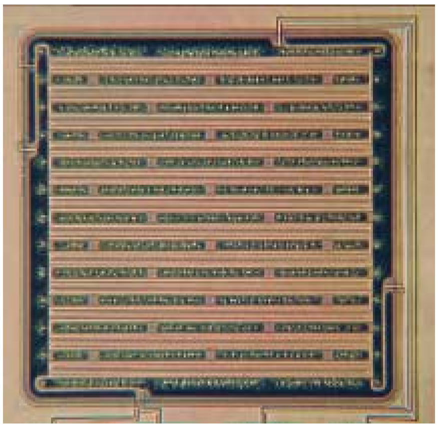


Figure 6: Magnified photograph of a bolometer (Pillans).

ADI is considering using a bolometer over a thermopile. Although under normal atmospheric pressure, thermopiles appear more responsive than bolometers, once the sensors are insulated in a vacuum, the bolometer quickly surpasses the thermopile in responsiveness (Pillans). ADI is exploiting this property of bolometers to create a more sensitive IR sensor.

A disadvantage of a bolometer is that it usually needs a matched pair of resistors in adjacent arms of a Wheatstone bridge circuit. If the temperature of the instrument changes, both sensors are affected in the same way and the bridge deflects. Measuring the output voltage by adding an external circuit has many drawbacks, as it is not easy to manufacture two resistors that possess the same properties (Brendan Cawley, personal communication, 16/08/2007). It is also difficult to estimate the self-heating effect in the resistors, which must carry a current in order for their resistances to be measured (Neuzil, 2001). The thermopile induces an emf and therefore does not need an external power supply like the bolometer. (Brendan Cawley, personal communication, 16/08/2007) The thermopile has the advantage of measuring temperature differences directly without any offset because if no radiation hits the detector, there is no temperature differential and hence the output signal is zero (Weckmann).

2.2.1 Operation in a Vacuum vs. Operation at Other Pressures

Ideally, the temperature measurement would occur in a vacuum because the only type of heat exchange that takes place there is radiation (Kirkpatrick). Over time, the substrate reaches the temperature of its environment, even if the substrate is in a vacuum. This state is called thermal equilibrium, which is undesirable because it reduces the accuracy of the IR sensor as no power is received. Placing the substrate in a vacuum greatly reduces the speed at which equilibrium is attained (Pillans).

Figure 7 is a simplified diagram of the heat exchange between the substrate, detector, and object providing the majority of the IR radiation.

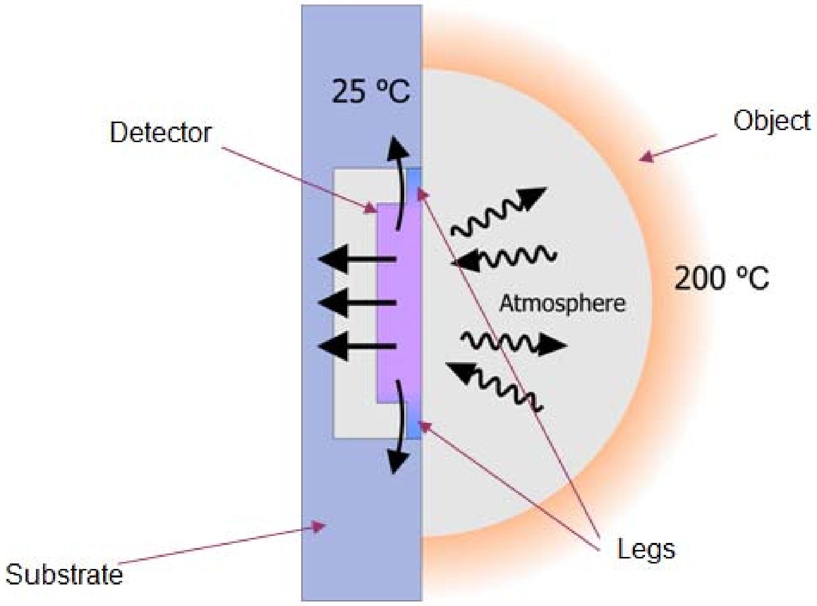


Figure 7: Heat exchange between the substrate, detector, and object (Pillans).

Eventually, the temperature of the detector reaches a temperature that is between that of the substrate and that of the object. Some of the signal from the object is lost through two sources: the legs of the detector and the atmosphere. The legs are necessary because they hold the detector in place, but they provide a thermally conductive path from the substrate. Since attaining a perfect vacuum is physically impossible, the atmosphere is also thermally conductive, so some of the signal is lost through here as well (Pillans).

2.2.2 Response Time

The sensor does not instantly respond to a change in radiation; there is always a delay. This delay is known as the response time of the detector. The response time is determined by four variables: the radiation power, thermal capacity of the detector, thermal impedance between the detector and substrate, and temperature of the substrate. This system is illustrated in Figure 7 (Pillans). In Figure 7, the radiation power is the constant current source, the thermal capacity of the detector element is the capacitance, the thermal impedance between the detector and substrate is the resistance, the temperature difference between the detector and substrate is the voltage source, and the temperature of the substrate is ground (Pillans).

There is a trade-off between the amplitude of the signal and the response time: if one increases, the other decreases. If the length of detector's legs is increased or the number of legs is decreased, the response time improves while the signal's amplitude diminishes (Pillans).

2.2.3 Distance-to-Spot Ratio

The distance-to-spot ratio is the ratio of the distance between the sensor and the target to the diameter of the spot that is being measured. For example, a sensor with a distance-to-spot ratio of 10:1 would absorb IR from a spot that is 1 cm in diameter if the sensor were placed 10 cm away from the target. A high distance-to-spot ratio indicates a high optical resolution, and it allows the user to measure smaller targets that are farther away from the sensor.

Distance-to-spot ratio is an important parameter because the target must completely fill the spot that the device is measuring. If this does not occur, the reading will be inaccurate because background radiation will interfere with the measurement, as illustrated in Figure 8.

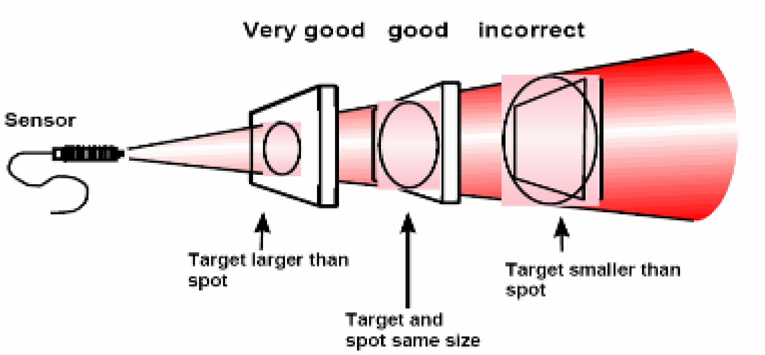


Figure 8: Target and spot sizes (Raytek).

In the leftmost target and spot in Figure 8, the target is larger than the spot, so the reading is accurate. In the middle target and spot, the target and spot are the same size. Consequently, the reading is still accurate, but it would be better if the target were larger than the spot. In the rightmost target and spot, the target is smaller than the spot, so background radiation interferes and the reading is inaccurate (Raytek).

2.2.4 Array Size

Several IR sensors can be put together to form an array, where each pixel detects the amount of IR in a different location. Larger arrays have better spatial resolution, which results in a more accurate image. They are also able to detect more motion across the field of view of the array, and if the target has multiple temperatures, larger arrays are better able to detect hot spots and cold spots (Pillans).

2.3 Applications

Reduced costs and technological breakthroughs have allowed IR sensors to become increasingly ubiquitous. ADI is planning to enter this field because they see a great potential in this technology. Since IR thermometry is an emerging market, other applications may be discovered in the future.

IR thermal sensors can be used in several settings: commercial settings, manufacturing settings, and settings where accurate temperature measurement is impossible or hazardous due to physical inaccessibility, extreme temperatures, or other causes (Raytek). Table 1 shows common specific applications of IR thermometry (Albuquerque Industrial, Inc.)

Table 1: Common IR thermometer applications (Albuquerque Industrial, Inc).

Common Infrared Thermometer Applications
Cement Kiln - burning zones; preheaters
Combustion or Incinerator - hot gases and utility boilers, rotary kiln
Energy conservation - insulation and heat flow studies; thermal mapping
Environmental - meteorological, biological, agriculture studies, remote airborne sensing, ground, water & ice surface temperatures
Filaments - annealing, drawing, heat treating
Food - baking, candy-chocolate processing, canning, freezing, frying, mixing, packing, roasting
Furnaces - flames, boiler tubes, catalytic crackers
Glass - drawing, manufacturing/processing bulbs, containers, TV tubes, fibers
Maintenance - appliances, bearings, current overloads, driving shafts, insulation, power lines, thermal leakage detection
Metals (ferrous and non-ferrous)- annealing, billet extrusion, brazing, carbonizing, casting, forging, heat treating, inductive heating, rolling/strip mills, sintering, smelting
Quality Control- printed circuit boards, soldering, universal joints, welding, metrology
Paint - curing, drying
Paper - coating, ink drying, printing, photographic emulsions, web profiles
Plastic Bulk - blow molding, RIM, film extrusion, sheet thermoforming, casting
Plastics Thin Films - photographic film materials, insulating films, PE, PP, PS, PA
Remote sensing (thermal mapping) - clouds, earth surfaces, lakes, rivers, roads, volcanic surveys
Rubber - calendering, casting, molding, profile extrusion, tires, latex gloves
Silicon - crystal growing, strand/fiber, wafer annealing, epitaxal deposition
Textile - curing, drying, fibers, spinning
Vacuum chambers - refining, processing, deposition

ADI plans to focus its IR thermometry efforts on automotive and high-volume applications, which require a low-cost chip that can be easily produced.

IR can also be used to detect the composition of a mixture of gases. This is possible because different gases absorb different wavebands. When the absorption pattern of a particular gas is known, the gas being measured can be identified (Colls, p. 16). This application is particularly important when toxic gases have to be identified. New regulations have banned the use of CFCs in air conditioners and refrigerators, but gases that are toxic to humans are now being used in new air conditioners. ADI hopes that their IR sensor will be incorporated into air conditioning and refrigeration units for the use of toxic gas detection (Pillans).

2.4 Previous Testing and Software Development

When the 2006 group began this project, the single-pixel, 3x3, and 11x11 ICs had been produced and testing software had been developed, but the 3x3 and 11x11 prototypes had not been fully tested. The 2006 ADI MQP group accomplished five main goals from which the 2007 group is building upon: verifying the behavior of each of the detectors when used with the USB evaluation board and the AES software, calibrating the detectors to produce an absolute temperature reading, developing and implementing tests that further characterize the sensors, improving the AES, and developing a demonstration that reveals the capabilities of the 3x3 and 11x11 arrays. The goals that are most applicable to the 2007 project are the development and implementation of tests, calibration of the detectors, and the improvements to the AES (MacMath).

It was necessary for the 2006 group to calibrate the ADT7301 for the 3x3 array because it provided an offset and gain that would need to be eliminated for an accurate temperature reading. To accomplish this task, the 2006 group used the temperature forcer to create temperatures between -25^{o}C and 100^{o}C, measured the reading from the ADT, and developed a calibration equation. Since only an offset and a gain were present, a linear calibration equation was ideal (MacMath).

The 2006 group conducted several tests that identified variables that would need to be controlled in future testing. These tests consisted of an angular response test, pixel response test, and the heated aperture plate test. The angular response test showed that the signal from the center pixel in the 3x3 array was significantly higher than the signal from the side pixels. When the angular response test had been completed, the ’06 group concluded that the best way to measure the response of the individual pixels was to take measurements solely from the center pixel. Further testing was carried out only on the center pixel. The purpose of the pixel response test was to test the effect of IR that was unrelated to the object whose temperature was being measured. The group used apertures of different sizes to provide this variable. Results showed that apertures of different sizes provided different offsets, which was expected. For apertures of different sizes, it was possible to convert the output from the

AD7794 into a temperature difference between the target and the ADT7301. There are several equations that convert the code to a temperature for a given aperture size. The last test, the heated aperture test, was carried out to verify the data used to calibrate the detectors. Results showed that the temperature of the aperture plate slightly affected the readings, but this problem will eventually be resolved by replacing the aperture plate with a small metal that will be in thermal equilibrium with the sensor (MacMath).

The 2006 group improved the AES by enabling it to communicate with all of the pixels in each of the arrays, increasing the speed, and simplifying the user interface by creating four modes. They created four modes: temperature mode, imager mode, real-time mode, and analysis mode. Temperature mode applies only to the 3x3 array. There are two graphs: one that displays the sensor's temperature, and one that displays temperature of the target. The user is able to enter the known calibration data for the ADT7301 and AD7794, as well as the pixel number under investigation. The generated data can be saved in an Excel, MATLAB, or comma-separated formats. Imager mode applies only to the 11x11 array. In this mode, thermal image data can be collected and saved in Excel, MATLAB, or comma-separated value format. The image itself can be saved as a JPEG file. Real-time mode was unchanged; it consists of the two graphs that display the AD7794 data and the ADT7301 data. Analysis mode contains the same two graphs as real-time mode, but it also displays data such as the maximum and minimum values collected, the mean, and the spread. Additionally, it contains a box that converts the codes from the AD7794 to an input voltage, a keypad that allows the user to select the pixel under investigation, a switch to display the AD7794 data at one time, and a box that allows the user to save the data in Excel, MATLAB, or comma-separated value formats. In this mode, the user also has the option of saving the data as hexadecimal values in Excel mode (MacMath).

The 2006 group conducted extensive testing on the 3x3 arrays and greatly improved the AES software. However, there are several characterization tests that they were unable to perform on any of the sensors, these being pressure, aging, and optimal position tests. During the testing process we found areas where LabVIEW software could reduce the time and resources needed to obtain meaningful results. In order to accomplish this we chose to automate certain aspects that required human supervision. These tests and their automation are the main focus of this report.

3. Goals and Objectives

The goal of this project was to facilitate the experimentation on the ADiR sensor. We planned to accomplish this by expediting the testing and data processing procedures while eliminating the need for employees to manually conduct the experiment or graph the data. The following are the detailed objectives and sub-objectives that we aimed to achieve:

- Understand the operation and properties of the four types of sensors: 3x3 and 11x11 thermopiles and bolometers. We would accomplish this by performing two experiments:
    1) A pre-automation pressure experiment. This would be performed by measuring the sensitivity of the sensors at different pressures between near-vacuum and atmospheric pressure.
    2) A vacuum degradation experiment. Before conducting the vacuum degradation experiment, we would need to modify a particular software module. After modifying the software module, we would perform the vacuum degradation experiment twice for two different purposes.
        - During the first vacuum degradation experiment, we would compare the responses of the 11x11 thermopiles and bolometers to one another when stored at three different temperatures: 0-4^{o}C, room temperature, and 150^{o}C. Every two-three days, we would gather data by measuring the sensors' sensitivities and the changes in their baseline readings at room temperature.
        - For the second vacuum degradation experiment, we would perform the same test on 3x3 bolometers that had undergone a new manufacturing process. To verify that the new process produced the desired results, we would store four of these sensors at 150^{o}C for two nights. We would compare the pre-storage sensitivities of each of the four sensors to their post-storage sensitivities, as well as the change in two of the sensors' baseline readings at room temperature.
- Develop LabVIEW virtual instruments (VIs) or Visual Basic for Applications (VBA) programs to expedite the testing process and improve the repeatability of the results. We would create automation software for five experiments:

1) The pressure experiment. Our goal was to reduce the human-machine interaction time for each part from sixty-five minutes, the time the experiment and setup required before automation, to five minutes, the time only the setup requires. To automate this test, we would need to automate two pieces of equipment: the vacuum controller, and the shutter. The vacuum controller would be used in the pressure test to control the pressure inside the vacuum chamber. Our VI would use the configuration parameters to set, maintain, and vary the pressure in the vacuum chamber. The shutter served as an object where the surface temperature was different from that of the heat being emitted from the radiation source. To automate the shutter, our VI would control it by rotating it down, into to a position where it would cover the sensor to block the radiation, and subsequently rotating it up to re-expose the sensor to the radiation.
2) The vacuum degradation experiment. Since ADI already possessed automated software for this experiment that could only be used on bolometers, our objective was to provide the flexibility needed that would enable the use of the software to test thermopiles as well.
3) The angular response experiment. Before we began this project, ADI did not possess any software to conduct the angular response experiment. If they were to conduct the angular response experiment manually, the test would probably require approximately a half hour, and the results would not be as repeatable. Our objective was to use LabVIEW to improve the repeatability of the results and to decrease the testing time to ten minutes.
4) The lens focusing experiment. Like the angular response experiment, ADI did not possess any software to determine where the sensor obtains the best signal in three-dimensional space when it is exposed to a radiation source. Our goal for this test was to develop a LABVIEW that would successfully and quickly.
5) The qualification test. We planned to develop a VBA program to automatically extract and process results from the qualification test using Microsoft Excel. Our intent was to reduce the time required to create graphs from ten minutes per part to one minute per part.

4. Testing Methodology

We conducted numerous experiments throughout the project: pressure, aging, angular response, lens focusing, and qualification experiments. Each of these required that we either create automation software or modify the software ADI possessed before our project began. In addition to verifying the functionality of our software, the tests we conducted provided ADI with useful information about the ADiR sensor. ADI used our results from the pressure test to improve their theoretical model regarding the behavior of the sensitivity of the IR sensor at different pressures. During the first vacuum degradation experiment, we compared the responses of the bolometers those of the thermopiles when they were stored at both cold and hot temperatures. The second time, we compared the responses of 3x3 bolometers that were manufactured by a new process to those of bolometers that were manufactured by a different process. ADI used our results from the angular response experiments to determine whether removing a diffractive pattern from the lens decreased the directivity of the sensor. The purpose the lens focusing experiment was to find the location in three-dimensional space where the sensor obtains a maximum response. We conducted the final test, the qualification test, for two reasons: to separate the working parts from those that were the non-functional, and to characterize the operational parts.

We communicated with the evaluation board by using a previously designed set of VIs in National Instruments' LabVIEW. LabVIEW allows for a high-level block diagram approach to programming and communicating with the IR sensor and lab instrumentation. Since the various physical instruments in the lab use different communication protocols, we needed different pieces of software for communication. ADI provided us with the necessary VIs to interact with and display the data output by the evaluation board. In addition to the VIs for interacting with the evaluation board, ADI also provided us with VIs for the temperature forcer and blackbody source. Although these VIs were functional, portions of the VIs required modifications to work with some of the equipment. These VIs and the necessary modifications are described in Section 4.2.1.

4.1. Pressure Experiment

The purpose of the pressure experiment was to measure the responses of the sensors at very low pressures and to improve the theoretical model. According to this model, when the pressure in a sensor is very low, the 3x3 bolometers are more sensitive than 11x11 bolometers. We tested this theory by using a LabVIEW VI to measure and observe the sensors' responses to a step input.

4.1.1. Setup of the Pre-Automation Pressure Experiment

The setup of the pressure experiment used four main components: the blackbody source, vacuum controller, vacuum chamber, and LabVIEW. This setup is summarized in Figure 9.

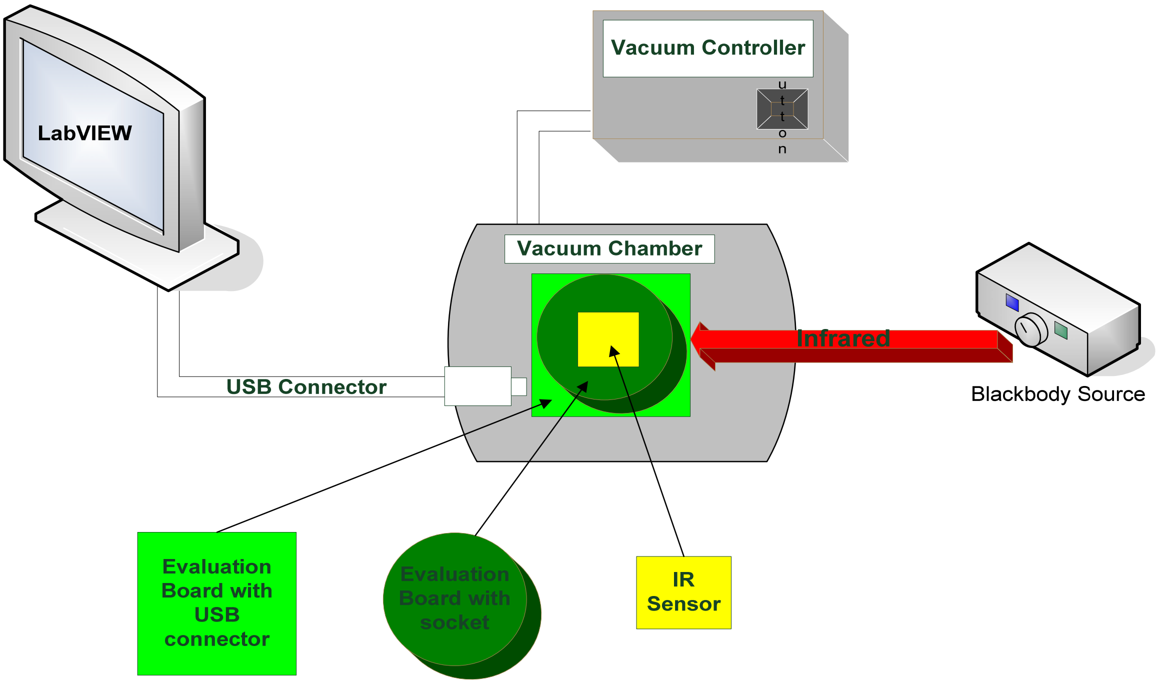


Figure 9: Setup of the pressure experiment.

We placed the blackbody source in front of the vacuum chamber, which housed the ADiR sensor and evaluation board. To set the pressure in the vacuum chamber, we used the vacuum controller. Inside the vacuum chamber, the ADiR sensor and evaluation board were subjected to variations in pressure ranging from atmospheric pressure to near-vacuum pressures. We used LabVIEW to observe the response of the sensor.

The blackbody source that ADI uses to emit radiation is the Mikron M31, a unit that consists of two pieces: the source and the digital temperature controller. Figure 10 shows a photograph of the unit.

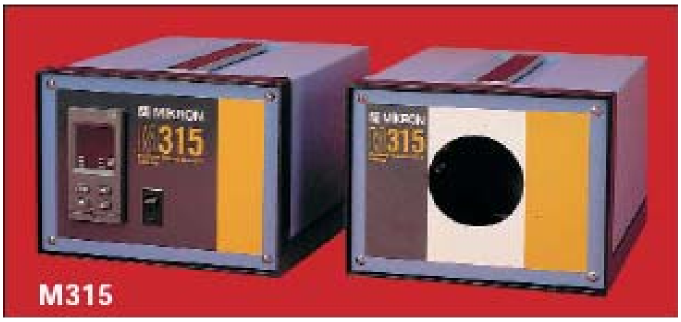


Figure 10: Mikron M315 blackbody source (Instruction Manual for Model M315 Blackbody Calibration Source).

The temperature controller can set the blackbody source to temperatures ranging between 10^{o}C and 300^{o}C. The blackbody source maintains a temperature within +/- 0.1^{o}C of the set temperature. For the pressure experiments, we set the temperature to 300^{o}C.

ADI uses the Pfeiffer Vacuum Controller RVC300 for pressure control and gas flow adjustment, as shown in Figure 11. Additional information about the vacuum controller can be found in Appendix B.

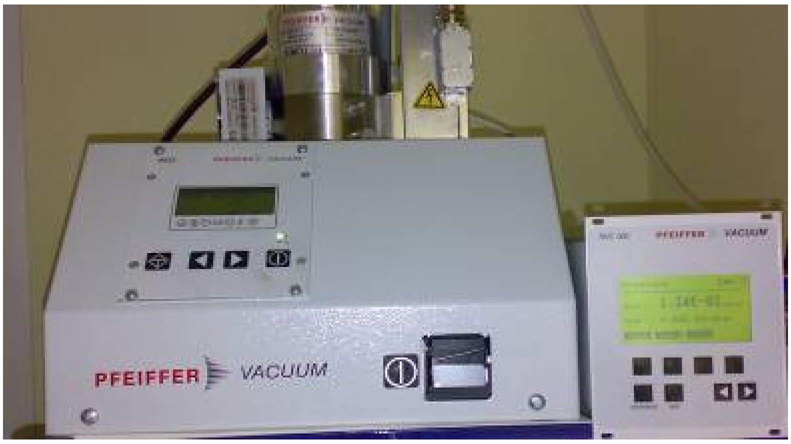


Figure 11: Vacuum controller.

Before we automated the LabVIEW software, we manually adjusted the pressure inside the vacuum chamber by pressing the buttons on the vacuum controller. This was no longer necessary after we automated the software, since our automated software was fully responsible for communicating with the vacuum controller.

When we performed pressure experiment, we mounted the ADiR sensor onto an evaluation board and placed it in an airtight vacuum chamber that could maintain the pressure applied. We allowed radiation to enter the vacuum chamber by placing a blackbody source in front of a window on the vacuum chamber.

The front panel of the VI used to gather data is shown below in Figure 12.

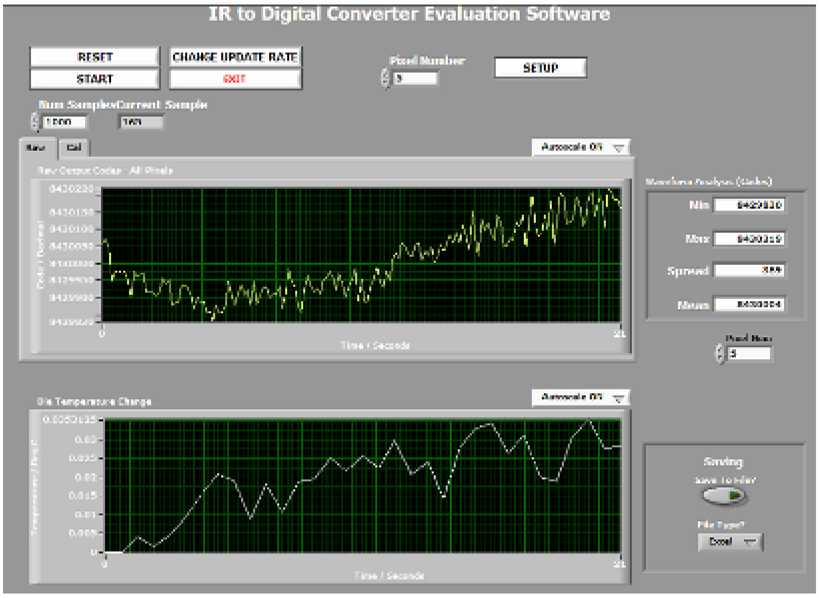
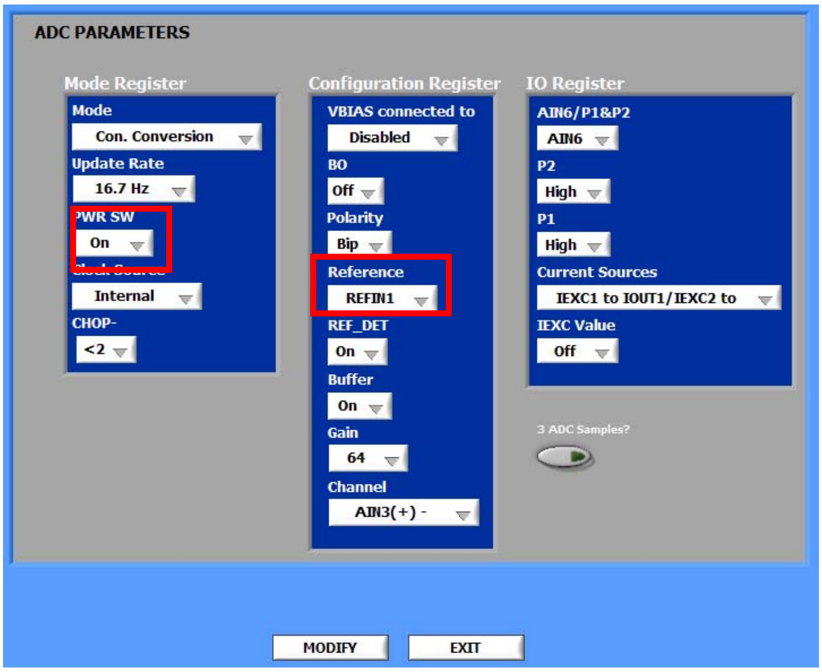


Figure 12: Front panel for collecting data from IR sensor.

This screen allowed us to initiate data collection, program the Cypress USB controller on the evaluation board, change the update rate, or open the setup screen. We needed to program the USB controller each time we plugged the cable into the evaluation board. Clicking the button labelled “Reset” quickly programmed the USB controller but did not provide any feedback to the user that the programming was successful.

The top graph in Figure 12 was the graph used to observe the response of the sensor to a step input. The y-axis corresponds to the power (in ADC units) received by the sensor and the x-axis is the time elapsed.

After programming the controller, we verified the settings in the setup menu. Clicking the “Setup” button on the main screen opens the window shown in Figure 13.



ADC PARAMETERS

Mode Register

Mode: Con. Conversion
Update Rate: 16.7 Hz
PWR SW: On
Clock Source: Internal
CHOP-: <2

Configuration Register

VBIAS connected to: Disabled
BO: Off
Polarity: Bip
Reference: REFIN1
REF_DET: On
Buffer: On
Gain: 64
Channel: AIN3(+) -

IO Register

AIN6/P1&P2: AIN6
P2: High
P1: High
Current Sources: IEXC1 to IOUT1/IEXC2 to
IEXC Value: Off

3 ADC Samples?

Figure 13: Setup window.

The two important parameters in the setup window were “PWR SW” and “Reference”. If we were testing a bolometer, we set PWR SW and Reference to “On” and “REFIN1” respectively. In the case of a thermopile, “Off” and “Internal” respectively were the correct settings.

Changing the “Pixel Number” and “Pixel Num” settings on the main screen changed the data displayed on the graph and calculated data in the “Waveform Analysis” accordingly. The numbers for these fields correspond to the pixel numbers as shown in Figure 12. The number of samples to collect from the pixel could be limited by setting the “Num Samples” field to a particular value.

Once we had properly configured the software, we pressed “Start” to initiate the data collection. As the program collected data, the top graph automatically scaled to display all the collected data. The bottom graph continuously updated with the data collected from the ADT. Once the set number of samples was collected or we pressed “Stop,” the fields in “Waveform Analysis” populated with values calculated from the collected data. The minimum power, maximum power, spread (maximum power minus minimum power), and mean power were displayed.

4.1.2. Procedure for the Pre-Automation Pressure Experiment

To measure the responsiveness of each of the 3x3 and 11x11 bolometers and thermopiles, we conducted step response tests several times for each part. Before adjusting the pressure, we needed a baseline to provide a control. To obtain this baseline, we conducted a single step response on each “uncracked” part. To qualify as an “uncracked” part, the seal of the part must be intact. We obtained only one reading for each uncracked part; it was unnecessary to vary the pressure because the seal prevents the vacuum inside the part from escaping. After we obtained a step response for the uncracked parts, the surface of each part was cracked, exposing the bolometer or thermopile to the pressure inside the vacuum chamber. When we tested the cracked parts, we varied the pressure inside the vacuum chamber.

To conduct a step response, we set the blackbody radiation source to 300°C and configured the VI to collect data from the center pixel, which is pixel 5 in the 3x3 sensors and pixel 61 in the 11x11 sensors. We collected a few seconds of data before we blocked the radiation from the blackbody source by placing a notebook at room temperature between the blackbody and the sensor. After a few more seconds, we lifted notebook, exposing the sensor again to the blackbody source. About a second after we lifted the notebook, we pressed the “Stop” button in the VI.

During this process, the sensor continuously read the amount of radiation it was receiving; when we placed the notebook in front of the blackbody source, the power received by the sensor decreased, and when we lifted the notebook, the power received by the sensor increased. This is shown in Figure 14.

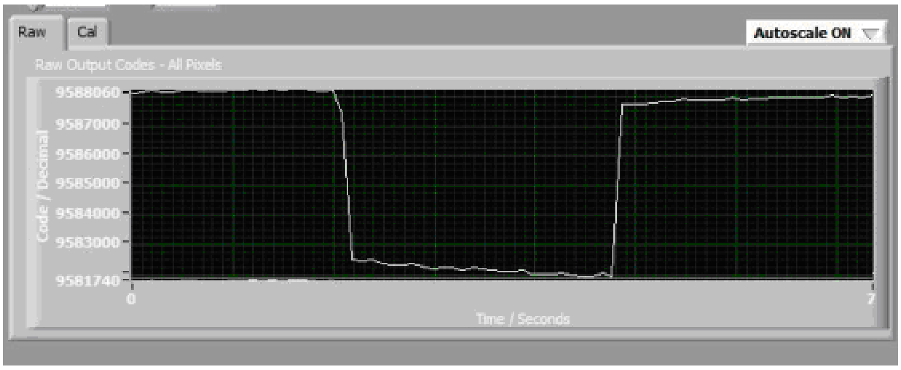


Figure 14: Screenshot of step response.

We measured the response using LabVIEW, which recorded it in the spread textbox as shown previously in Figure 14. For the cracked parts, we repeated this process for various pressures between 1μBar to 1000 mBar, as set by us when using the vacuum controller. For each pressure, we stored the spread in an Excel spreadsheet and calculated the pressure quotient. The formula used to find the pressure quotient is given in Equation 2.

$$q = \frac{s}{B}$$

Equation 2: Equation for the pressure quotient.

In Equation 2, ‘q’ is the pressure quotient, ‘s’ is the spread given by LabVIEW for a specific pressure, and ‘B’ is the baseline spread for the same part before it was cracked. Each of the pressure quotient points was plotted on the y-axis of a graph, and the pressure was plotted on the x-axis of the same graph. We compared the shapes of the curves for bolometers and thermopiles, as well as the maximum pressure quotients achieved by both thermopiles and bolometers.

4.1.3. Pressure Experiment: Automation

After conducting the pressure experiment manually, which was a tedious and time-consuming process, we realized that we could use LabVIEW software to remove the need for an engineer to conduct the entire 60-minute experiment. Automating the experiment would also improve the reliability of the results by measuring the response of the sensor at specific, user-defined pressures. Before automating the pressure experiment, it was impossible to obtain readings at a single pressure because we were unable to prevent the pressure from fluctuating significantly while we were taking a reading. We decided to automate the two pieces of equipment used in the test that required human-machine interaction: the vacuum controller and the shutter.

To eliminate the need for an engineer to manually set the pressure inside the vacuum chamber by pushing the buttons on vacuum controller every five minutes, we automated the vacuum controller so that the software could automatically set the pressure. We used an RS232 cable to allow the computer to communicate with the vacuum controller. To allow the user to set the pressure values before beginning the experiment, we modified a LabVIEW VI provided by the manufacturers of the vacuum controller.

Before we automated the pressure experiment, we used a lab notebook to produce a step input by blocking the IR every time we reached a pressure at which we needed a reading of the sensor’s response. For each part, we would produce a step input approximately thirty times. To automate this process, we used a shutter that we controlled by using a LabVIEW VI provided by NXT Lego Company. This shutter automatically blocked the IR as soon as the vacuum inside the vacuum chamber reached the user-specified desired values.

Once we had automated both the vacuum controller and shutter individually, we combined both of the modules into a single LabVIEW VI, ran the post-automation pressure experiment, and compared our results to from the pre-automation experiment to determine whether they were similar. If the results were similar, we would be able to conclude that we had successfully automated the pressure experiment.

With the exception of the LabVIEW software used in the experiments, the setup of the post-automation experiment was nearly identical to the setup of the pre-automation experiment. The only difference was the addition of a shutter in the post-automation experiment, as depicted in Figure 15.

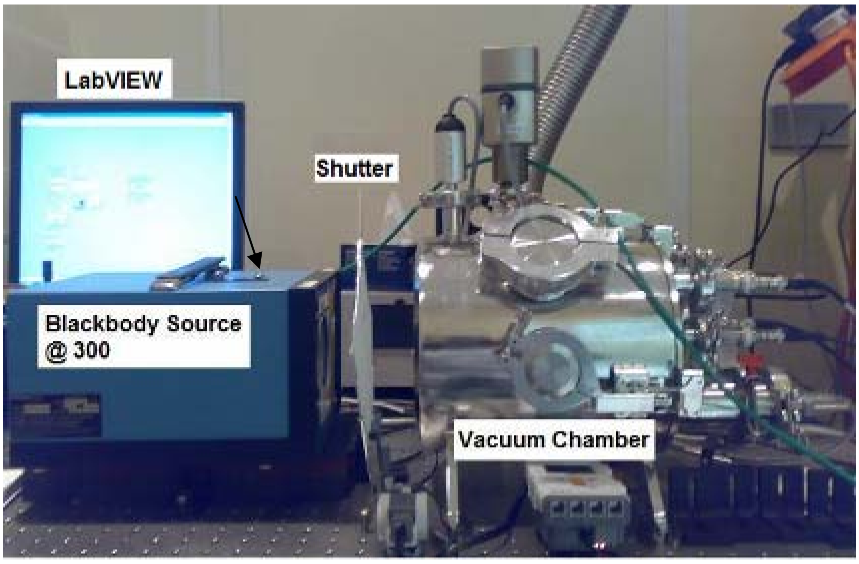


Figure 15: Photograph of the setup of the post-automation pressure experiment.

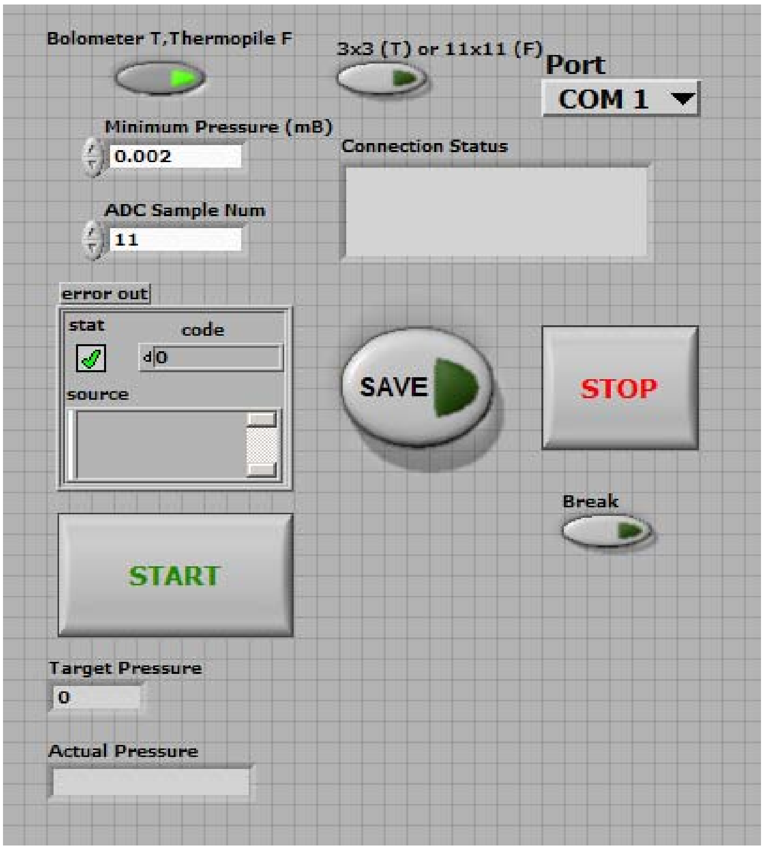
We placed the shutter between the blackbody source and the vacuum chamber in a configuration such that the shutter's default position allowed the sensor to be exposed to the blackbody radiation, but when the pressure in the vacuum chamber reached the desired value, the software would move the shutter to a position that prevented the radiation from reaching the sensor.

After combining the modified VIs for controlling the vacuum controller and the shutter, we conducted the pressure experiment using our automation software to verify that the results obtained from our automated pressure experiment were similar to the results from the pre-automation pressure experiment. Before starting the pressure experiment, we re-configured some of the PID parameters of the vacuum controller. The main purpose of this was to prevent the pressure inside the vacuum chamber from fluctuating after we had set the target pressure. Initially, we set all the parameters to zero and observed the response of the vacuum controller to change in pressures. We then started increasing the PID parameter values in small increments until we obtained stable readings for the pressure inside the vacuum chamber. Table 2 describes these changes to the PID parameters.

Table 2: PID parameters.

PID Parameter	Value
Kp	10
Tn	5
Tv	1.1

The front panel of the automated pressure experiment is shown below in Figure 16.



Port: COM 1

Minimum Pressure (mB): 0.002

ADC Sample Num: 11

error out — stat: ☑ code: 0

Target Pressure: 0

Figure 16: Front panel of the automation software for the pressure experiment.

This software is user friendly and flexible, as it allows the user to choose between a bolometer and thermopile of 3x3 or 11x11 pixels. To run our software on an 11x11 bolometer, we pushed the “Bolometer T, Thermopile F” button to indicate that we were testing a bolometer, and did not press the “3x3 (T) or 11x11 (F)” button because we were testing an 11x11 sensor. The “Minimum Pressure (mB)” text box contains the user-specified minimum pressure at which the software samples the sensor. When we tested our post-automation software, we set this to 0.002. The “ADC Sample Num” text box contains the number of samples the software takes from the sensor at each user-specified target pressure. For example, when the pressure inside the vacuum chamber reaches 0.002 mBar, the software gathers the step response from the sensor eleven times before communicating with the vacuum controller to change the pressure to another value. After we entered these parameters, we pressed the “START” button, and when the experiment had finished, we saved the data to an Excel file, which automatically plotted the pressure curve. If we had wanted to stop the experiment earlier, we could have pressed the “Break” button. The experimentation software for the pressure experiment will be discussed in detail in Section 6.1.1, and the data processing software that plotted the pressure curves will be discussed in Section 6.2.1.

4.2. Vacuum Degradation Experiment

The purpose of the vacuum degradation experiment was to determine whether storage at different temperatures would affect the sensors' offsets at room temperature or their sensitivities. Since vacuum degrades more quickly at high temperatures than at low temperatures, we investigated different rates of vacuum degradation by exposing the sensors to extreme temperatures, both cold and hot. Out of the six bolometers and six thermopiles that we tested, we put two of each type in a 150^{o}C oven, another two of each in a refrigerator where the temperature was between 0^{o}C and 4^{o}C, and the remaining two of each at room temperature. Before we placed the sensors in the oven, refrigerator, or at room temperature, we conducted two tests on them: the temperature cycling test, which provided results regarding the sensor's offset and gain at room temperature (10^{o}C and 30^{o}C) , and the step response test, which provided results regarding the sensor's sensitivity. We conducted the temperature cycling test and step response test on the sensors every two or three days during the seven days of storage. After seven days, we made several comparisons: the responses to changes in room temperature of the thermopiles versus those of the bolometers, the step responses of the thermopiles versus those of the bolometers, and changes in the sensors' readings for given temperatures. The following sections detail the purpose, setup, and procedures for the temperature cycling test and the step response test.

4.2.1. Temperature Cycling Test

When we conducted the temperature cycling test, we used the temperature forcer to vary the temperature of the sensor's environment from 10^{o}C to 30^{o}C over a period of twenty minutes and collected data for each pixel in the arrays. We used LabVIEW software to output commands that control the temperature forcer and to receive data from the sensor.

The purpose of the temperature cycling test was to analyze the sensors' responses to changing temperatures that they are likely to encounter while they are being used. The sensors' responses to the temperature cycling before being stored in an oven or refrigerator provided baselines. We used this baseline to determine whether an offset or change in gain was present after storage.

The main components in the temperature cycling test were the sensors, foam stabilizer, temperature forcer, and LabVIEW software. The evaluation board on we placed the sensors was connected to a computer via a USB cable, and the computer was connected to the temperature forcer via a GPIB-to-USB cable.

To contain the evaluation board and sensor and provide thermal insulation, we used a previously constructed foam stabilizer. A photograph of this foam stabilizer with the evaluation board and a sensor is shown in Figure 17.

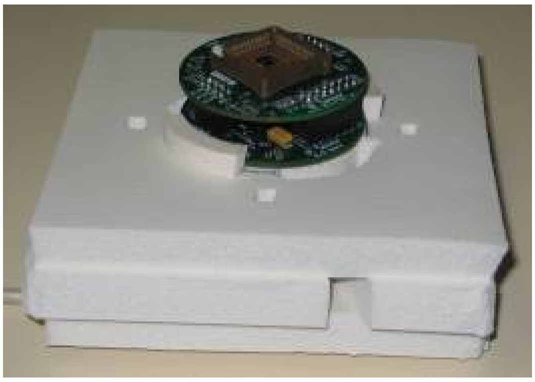


Figure 17: Foam stabilizer, evaluation board, and sensor.

The temperature forcer used was a Temptronic TP04200A. When we conducted the tests, we ensured that the temperature forcer was in engineering mode. A photograph of this machine is shown in Figure 18.

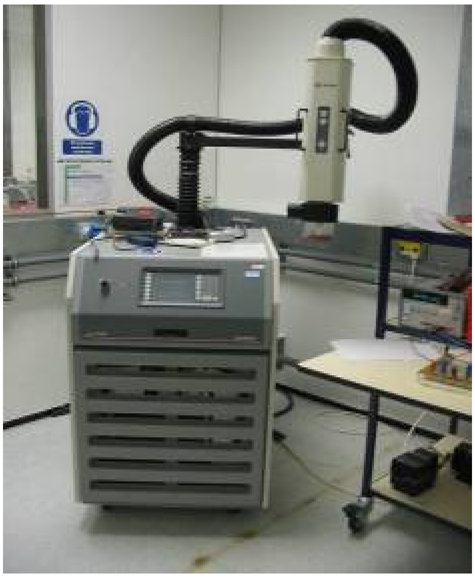
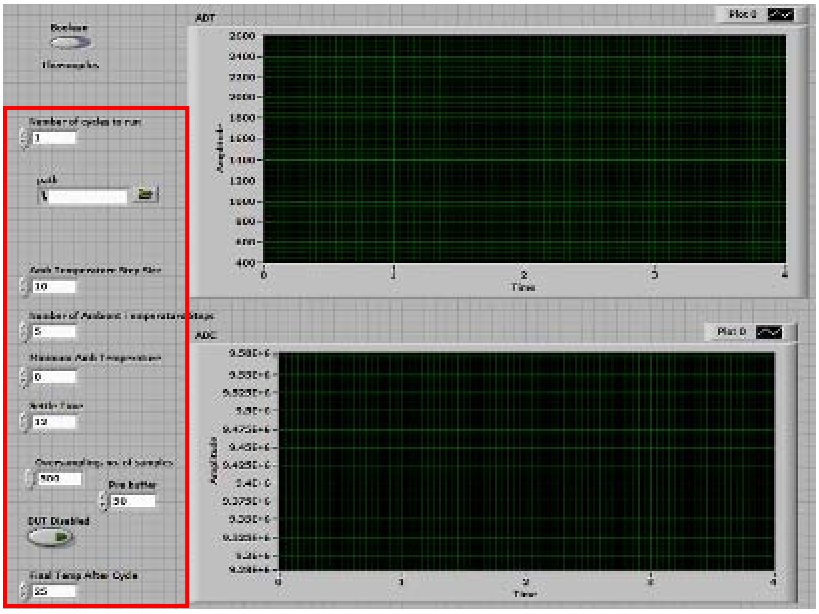


Figure 18: Temperature forcer.

As discussed previously in Section 4.2, we used the temperature forcer to precisely and accurately control the air to which the IC is exposed. In addition to the temperature, the duration of time for the exposure can be controlled with the temperature forcer. GPIB allows up to 15 devices to share a single 8-bit parallel electrical bus by daisy chaining connections.

The LabVIEW VI for the temperature forcer allows for various parameters to be entered. A screenshot of the VI is shown below in Figure 19.



Boolean
Thermopiles
Number of cycles to run: 1
path
Amb Temperature Step Size: 10
Number of Ambient Temperature Steps: 5
Minimum Amb Temperature: 0
Settle Time: 10
Oversampling: no. of samples: 500
Pre buffer: 50
DUT Disabled
Final Temp After Cycle: 25

Figure 19: Front panel of the temperature forcer VI.

The original VI provided was only for bolometers; we needed a method to switch between bolometers and thermopiles. To accomplish this, we added a button to the front panel. We stored the state of the button into a global variable, which was then connected to a sub-VI that collects the data from the evaluation board. Next, we needed to swap the original sub-VI for an updated one that had an additional input. The additional input is of type Boolean (T/F), and corresponds to bolometers and thermopiles respectively.

The parameters highlighted in Figure 19 alter the test in different ways. Table 3 describes their purpose.

Table 3: Temperature forcer test parameters.

Devices	The type of IR sensor to be tested (Bolometer or Thermopile)
Number of cycles to run	The number of times that the test will run.
Path	Path for the output file.
Amb. Temperature Step size	Increase between each temperature set point.
Number of Ambient temperature steps	Number of set points per test.
Minimum Ambient temperature	Starting set point.
Settle time	Time at each set point.
Oversampling	Number of samples to average per reading
Prebuffer	Number of samples to take before starting
DUT Disabled	Disables Device Under Test temperature reading. (On or Off)
Final temperature after cycle	Temperature for the forcer to return to upon completion of the test.

To prevent air at room temperature from interfering with the tests, we inserted insulating foam into the duct of the temperature forcer.

When running the tests, we placed a tube over the foam stabilizer. A photograph of the entire setup is provided in Figure 20.

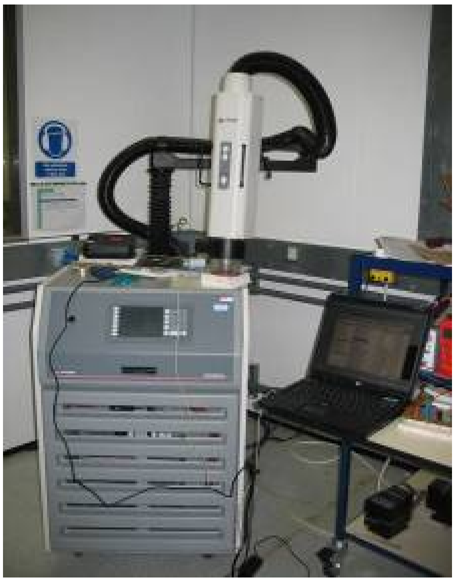


Figure 20: Setup for the temperature cycling test.

After modifying the LabVIEW software to automatically adjust the settings for bolometers and thermopiles, we ran the tests by inputting the particular parameters from Table 4 into the appropriate VI.

Table 4: Parameters used for the temperature cycling test.

Device	"Bolometer" or "Thermopile," depending on which is being tested
Number of cycles to run	1
Path	Leave blank
Ambient Temperature Step Size	20
Number of Ambient Temperature Steps	2
Minimum Ambient Temperature	10
Settle Time	4
Oversampling	10
Prebuffer	10
DUT Disabled	On (i.e. the light is lit)
Final Temperature After Cycle	25

After we pressed the "Run" button in LabVIEW and gave the output file a name, the tests required approximately twenty minutes to complete. We repeated this process for each of the sensors.

For each pixel in the arrays, we obtained measurements from the ADC and ADT. The ADT measurement represented the temperature measurement of the die and the ADC measurement represented the temperature of the air. We observed the differences between these measurements over time for bolometers and thermopiles that were stored at room temperature, refrigerator temperature, and oven temperature.

4.2.2. Step Response Test

The setup and procedure for the step response test were nearly identical to those mentioned in the pressure experiment, which is described in Section 4.1.2. However, the purpose of the step response test was different from that of the pressure experiment as it enabled us to identify changes in the sensor's sensitivity to a change in temperature. One of the differences in the procedure of the step response test compared to the pressure experiment was that the step response test was conducted only at atmospheric pressure, so the vacuum controller did not play a role in this test. Another difference was that the sensors were never cracked in the step response test. Instead of calculating the pressure quotient as described in Section 4.1.2, we calculated a delta by subtracting the spread obtained before storage from the spread obtained after storage. These deltas for bolometers and thermopiles stored at room temperature, refrigerator temperature, and oven temperature were compared to one another to determine which of the two showed the least amount of change in their delta over time.

4.3. Angular Response Experiment

The purpose of the angular response test was to observe the behavior of the ADiR sensor when rotated through 180° in increments of 1°. We used LabVIEW to develop a VI that would rotate the rotational stage and record the response value at every increment. We conducted this test for a sensor with and without a diffractive lens pattern. Figure 21 depicts the lens with the diffractive pattern, when it was unpolished, and without the diffractive lens, when it was polished.

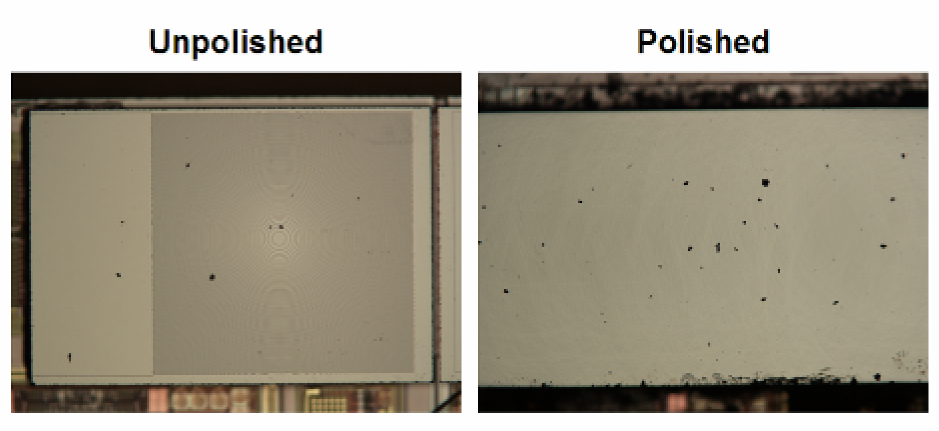


Figure 21: Photographs of the lens pattern with the diffractive lens (unpolished) and without the diffractive lens (polished).

The purpose of the diffractive lens is to focus the radiation on the center pixel. When this occurs, the sensor exhibits directivity. To determine whether the diffractive lens was working properly, we conducted this test on a sensor with and without a diffractive lens pattern. The results obtained from this test could be used to determine whether the ADiR sensor exhibits angular response. This characteristic is important because the ADiR sensor could be used in motion detection applications if it demonstrates this property.

4.4.1 Setup of the Angular Response Test

The setup of the angular response test is depicted below in Figure 22.

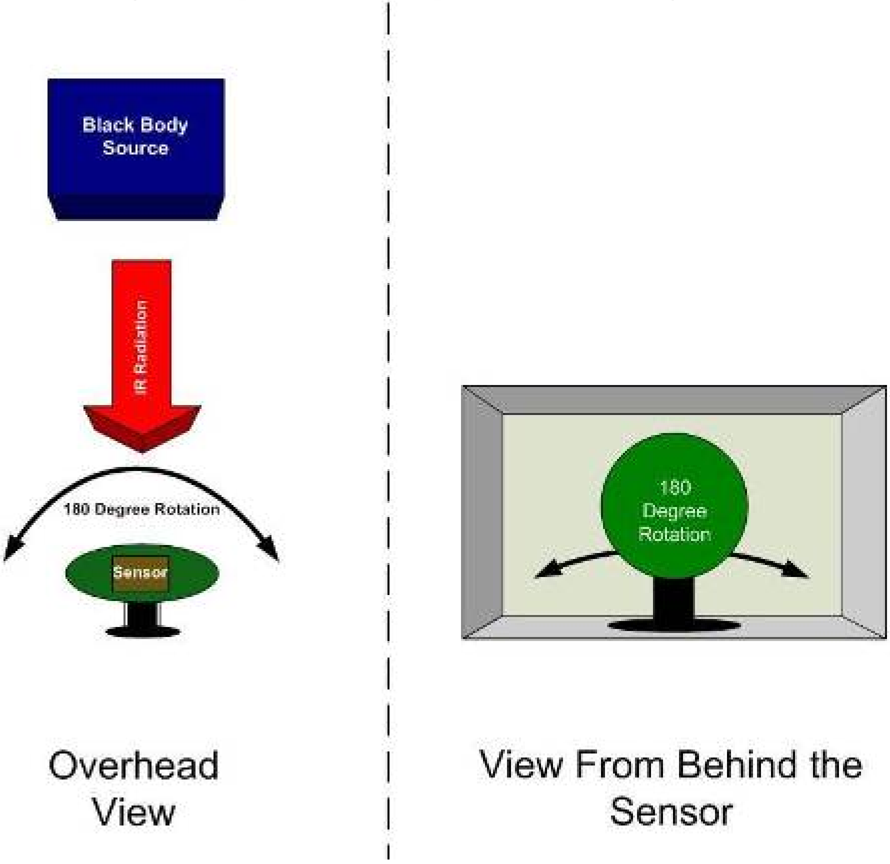


Figure 22: Setup of the angular response test.

We placed the blackbody source in front of the rotational stage, shown in Figure 23, which housed the ADiR sensor.

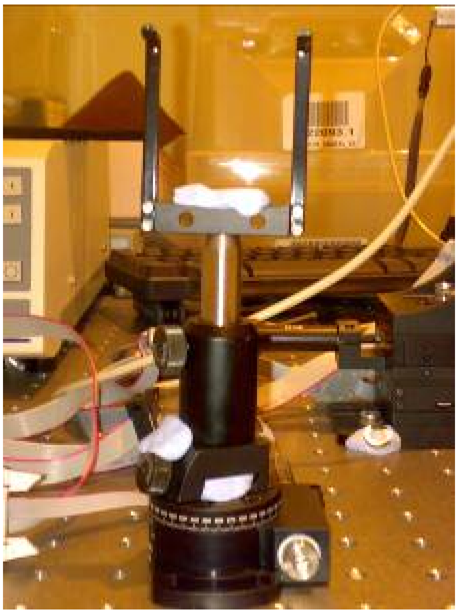
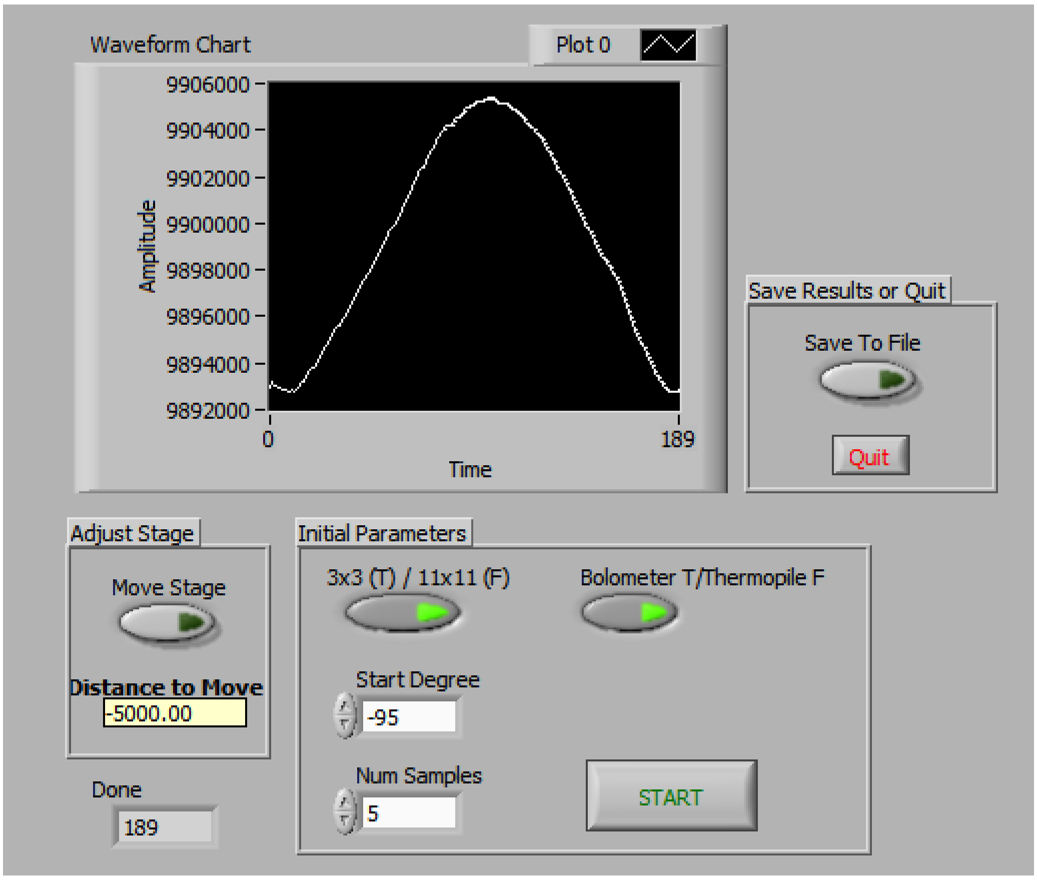


Figure 23: Photograph of rotational stage.

We specifically placed the sensor in the U-shaped holder, shown in Figure 23. An aluminium box with a hole in the middle behaving like an aperture was then placed in between the sensor and the blackbody so that the radiation would focus mainly on the sensor. We used the LabVIEW VI that we created to control the rotational stage so that it rotated from 0° to 180°, gathering data at every 1° increment.

Our software is very flexible and allows users to test for 3x3 or 11x11 bolometers or thermopiles. It also lets the users configure the position of the rotational stage by allowing them to click the "Move Stage" button. The software also shows the percentage of the test completed. A screenshot of the VI is shown below in Figure 24.



Distance to Move: -5000.00

Start Degree: -95

Num Samples: 5

Done: 189

Figure 24: Screenshot of the angular response VI.

4.4.2 Procedure for the Angular Response Test

After creating the software to automatically move the rotational stage and gather data at 1^{o} increments, we tested an ADiR sensor by mounting it on the stage. Before starting the test, we rotated the stage so that the sensor was perpendicular to the blackbody source. We then pressed the "START" button to begin the test, and when the test had completed, we saved the data into an Excel file by clicking the "Save To File" button. After we ran this test for the first time, we realized that there were some unnecessary and time-consuming modules in our

VI, and that if we removed these modules, the test time would be reduced significantly. We removed the unnecessary modules and re-ran the test.

4.4. Lens Focusing Experiment

The main purpose of the lens focusing experiment was to locate the point in three-dimensional space where the response of the ADiR sensor reaches a maximum. This peak position, or the focal point, is depicted in Figure 25.

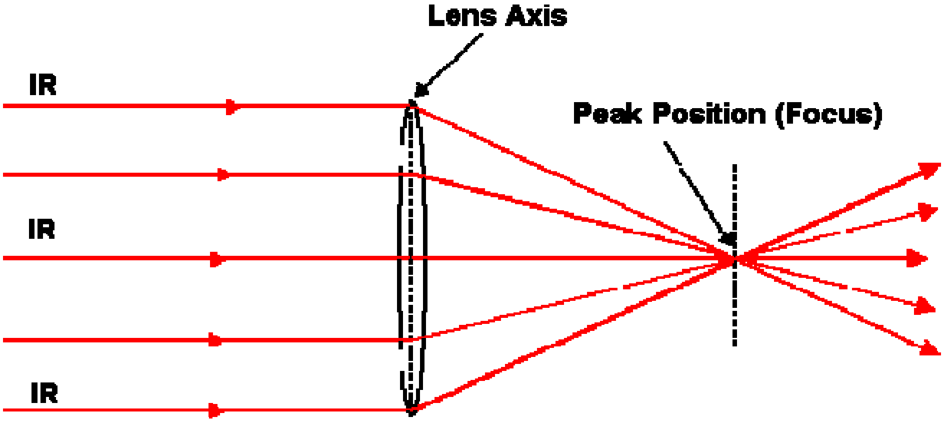


Figure 25: Position of the focus relative to the lens.

To move the sensor to different locations in three-dimensional space, we programmed a three-axis translational stage.

4.4.1. Setup of the Lens Focusing Experiment

We placed the blackbody source in front of the three-axis translational stage, which housed the ADiR sensor. We used the LabVIEW VI that we created to obtain the response of the ADiR sensor and control the movements of three-axis translational stage in three dimensions. A photograph of the setup is provided in Figure 26.

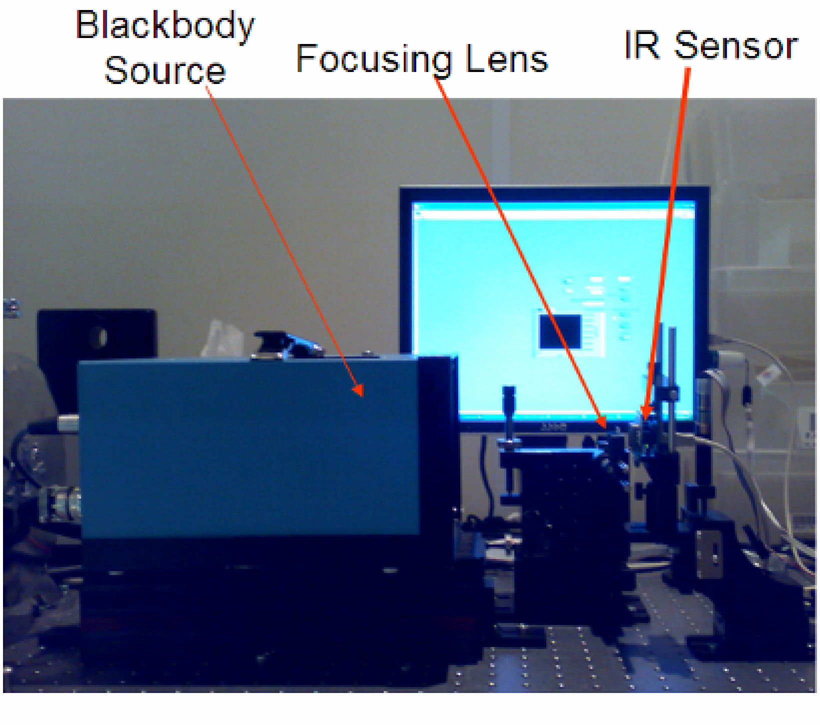
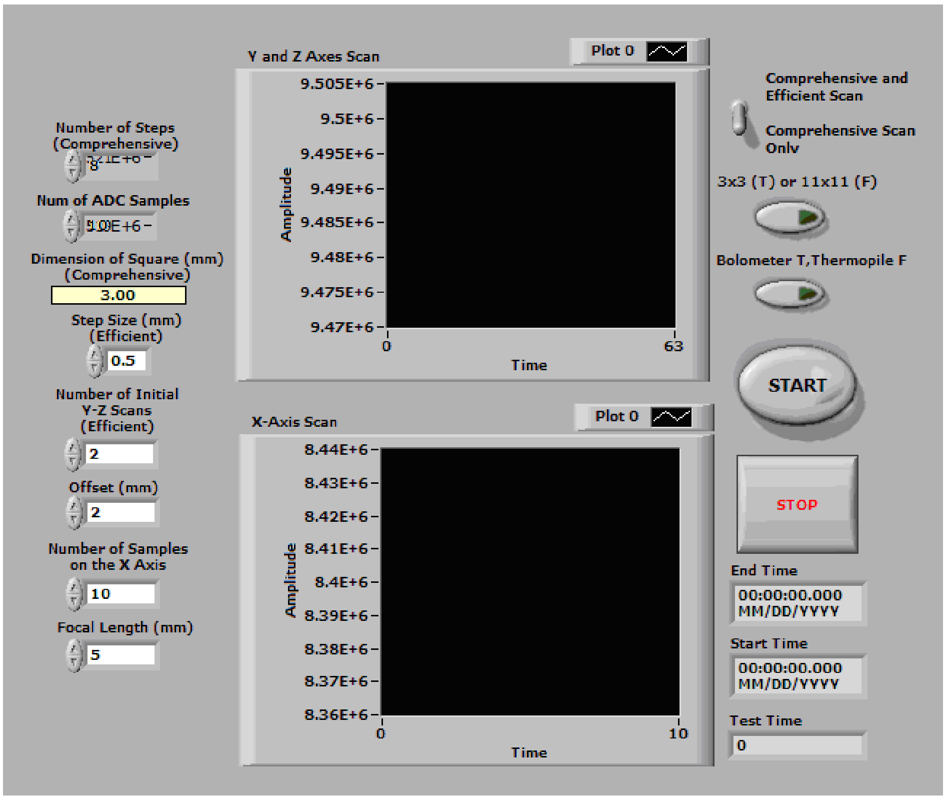


Figure 26: Photograph of setup for the lens focusing experiment.

Our software for this experiment is very flexible because it allows users to conduct a comprehensive scan, which samples the sensor at equally spaced points in three-dimensional space, or an efficient scan, which utilized information input by the user regarding the focal length to find the focus more quickly. We have also included two switches that let the user test all four combinations of 3x3 and 11x11 bolometers and thermopiles. A screenshot of the VI is shown below in Figure 27.



Number of Steps (Comprehensive) 8

Num of ADC Samples 5.0E+6

Dimension of Square (mm) (Comprehensive) 3.00

Step Size (mm) (Efficient) 0.5

Number of Initial Y-Z Scans (Efficient) 2

Offset (mm) 2

Number of Samples on the X Axis 10

Focal Length (mm) 5

End Time 00:00:00.000 MM/DD/YYYY

Start Time 00:00:00.000 MM/DD/YYYY

Test Time 0

Figure 27: Front panel of our VI for the lens focusing experiment.

4.4.2. Procedure for the Lens Focusing Experiment

We completed the lens focusing experiment by creating the comprehensive scan, analyzing the results from that scan, developing an algorithm for the efficient scan, and implementing the algorithm in a VI. After creating the software to automatically move the translational stage to user-specified points in a cube, as shown in Figure 28, we tested our software on the ADiR sensor by mounting the sensor on the stage.

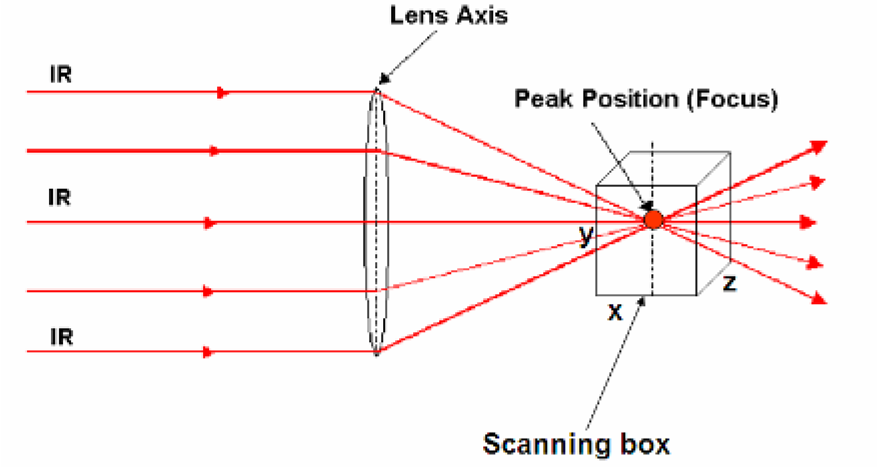


Figure 28: Scanning cube relative to the peak position and lens.

When we created the complete scan, we first programmed the translational stage to move to a starting position, which we designated to be the origin: the forward-bottom-right-hand corner. After moving to the origin, the translational stage moves the sensor to a user-specified number of equally spaced points inside the cube and gathers the sensor's response. After we analyzed the data we gathered from this experiment, we developed the algorithm for the efficient scan. This algorithm is based on information that the user inputs about the focal length. The sensor first scans the y- and z-axes. When the software finds where the sensor's response is the highest in that plane, the software moves the sensor to that point and moves it along the x-axis, which is the axis perpendicular to the focal point.

4.5. Qualification Tests

The purpose of the qualification was twofold: to separate the properly working sensors from the non-functional ones, and to characterize the operational sensors. We placed one sensor at a time through a tester, which provided us with information that enabled us to determine whether the part was functional. The tester also gathered data, which we graphed in Excel after the completion of the test. Although we did not create software to run the test, we developed software to expedite the data processing that occurs after the completion of the qualification test.

4.5.1. Setup of the Qualification Test

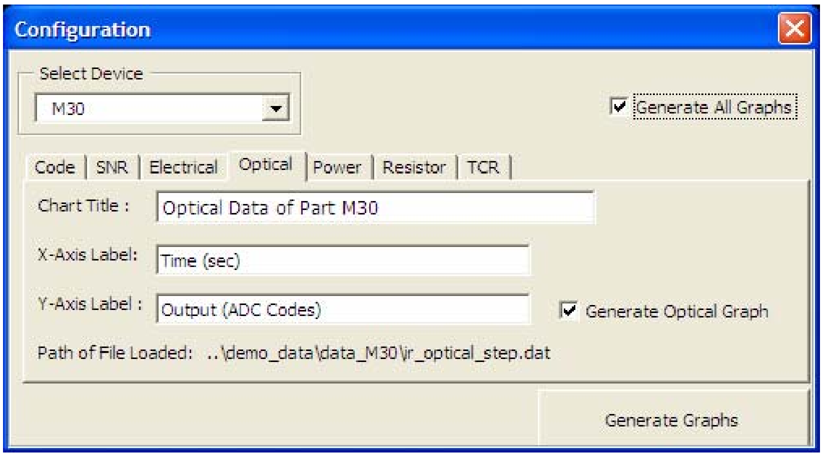
There were three main pieces of equipment involved with this test besides the ADiR sensor: the CTS5340HSC F1 tester, Temptronic temperature forcer, and computer. The tester measured different characteristics of the ADiR sensor, such as the resistance of each of the resistors in the Wheatstone bridge and output codes of the ADC. We used the temperature forcer to change the temperature to which the ADiR sensor was exposed.

4.5.2. Procedure of the Qualification Test

To run the test, we first placed the part into the tester, aligned the temperature forcer with the sensor, and set the temperature forcer to 25°C. After pushing the "Test Part" icon, and waiting for the tester to complete some of the tests, we determined whether the part was functional by comparing the power of the center pixel to -40dB. If the power of the center pixel was higher than -40dB, the part was functional, so we pressed "1" to continue with the test. If the power was lower, we pressed "0", removed the part from the tester, and tested another part. After seven or eight minutes, when prompted, we set the temperature forcer to

100^{o}C, waited a minute for the sensor to heat, and hit “Return”. After approximately twenty seconds, the software displayed “Pass” or “Fail”. If the software displayed “Fail”, we ended the test and removed the part from the tester. If it displayed “Pass”, we set the temperature forcer to 6^{o}C, waited approximately one minute for the sensor to cool, re-set the temperature to 25^{o}C, saved the data, removed the part, and began the test again for another part.

To create the graphs after the qualification test, we ran our VBA program in Excel. When we opened the program, we clicked on the “Launch Program” icon, as shown below in Figure 29.



Configuration

Select Device: M30

☑ Generate All Graphs

Chart Title : Optical Data of Part M30

X-Axis Label: Time (sec)

Y-Axis Label : Output (ADC Codes)

☑ Generate Optical Graph

Path of File Loaded: ..\demo_data\data_M30\ir_optical_step.dat

Figure 29: Screenshot of the configuration screen in our data processing software for the qualification test.

Next, we selected the first part in the list of available parts, and checked “Generate All Graphs”. Finally, we clicked the “Generate Graphs” button on the bottom right-hand corner of the window. We repeated this process for each of the parts we tested.

5. Results and Analysis

This section discusses the results obtained from the pre- and post-automation pressure experiments, vacuum degradation experiment, angular response experiment, lens focusing experiment, and qualification test. Although we extensively analyze the results from the pressure, vacuum degradation, angular response, and lens focusing experiments, we do not extensively analyze the results from the qualification test because our goal for that particular test was simply to create software and analyze the results.

5.1. Pre-Automation Pressure Experiment

As mentioned in Section 4.1, we conducted a pressure experiment before and after we automated the software that obtained the data and controlled the pressure inside the vacuum chamber. The sole purpose of the pre-automation pressure experiment was to confirm that the ADiR sensor is more responsive at lower pressures near vacuum. Our results also provide experimental data that will aid ADI in improving their theoretical model to make better predictions. The complete results are shown in Appendix C.

5.1.1. Pre-Experiment Voltage Regulator Testing

Before conducting the pre-automation pressure experiment, we needed to determine whether we would use a 3.3V regulator or a 5V regulator. Before our voltage regulator test, ADI was using a 3.3V regulator, but they were considering the use of a 5V regulator. A voltage regulator was necessary because it prevents noise from coupling onto the ADC when directly powered by the USB supply. To determine which voltage regulator to use, we compared the step response of a cracked 3x3 bolometer when supplied by two different voltages: 3.3V and 5V.

Figure 30 shows the comparison of the step response of a cracked 3x3 bolometer when supplied by a 3.3V regulator and 5V regulator.

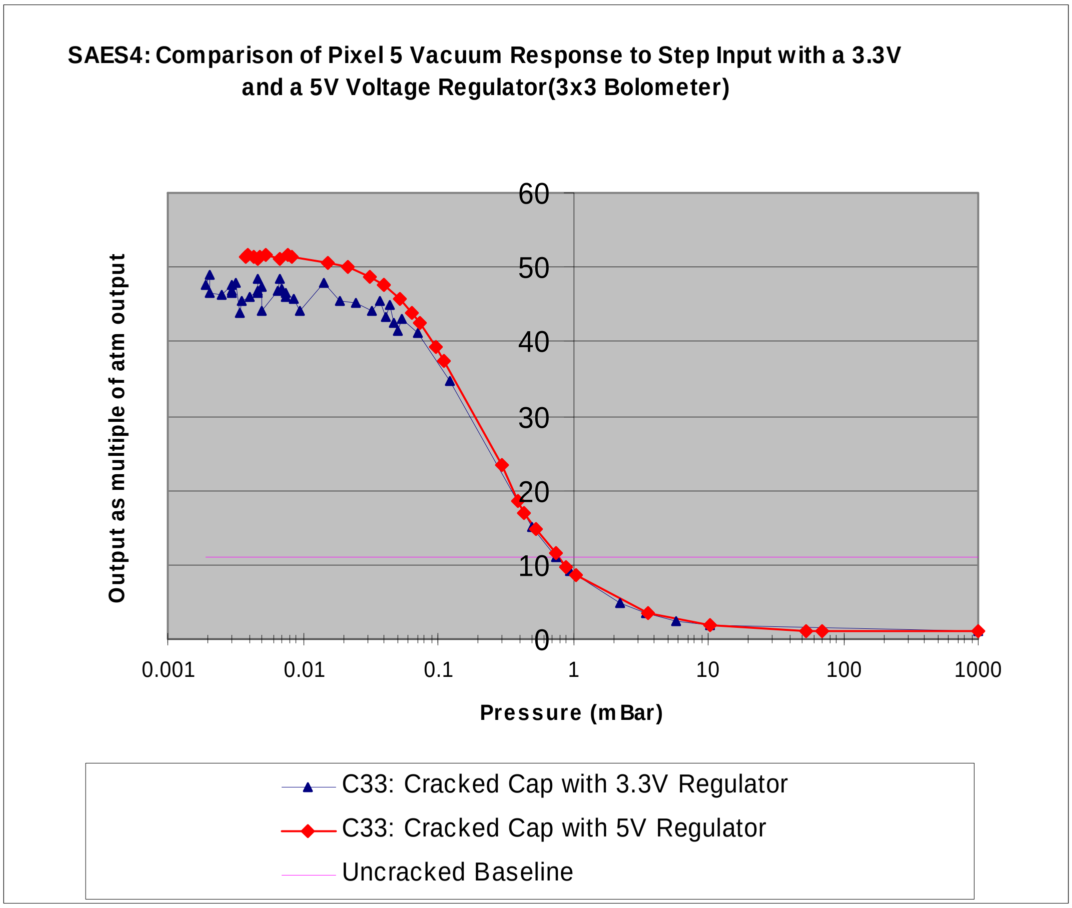


Figure 30: Comparison of the step responses of a cracked 3x3 bolometer when powered by a 3.3V regulator vs. a 5V regulator.

When we examined Figure 30, we found that the 5V regulator removed noise more effectively than the 3.3V regulator. Consequently, ADI replaced the 3.3V regulator with the 5V regulator. It was unnecessary for us to test regulators with ratings above 5V because 5V is the maximum voltage the USB could supply.

5.1.2. 3x3 Bolometer Theoretical Model vs. 3x3 Experimental Model

ADI has designed a theoretical model for the pressure experiment that is used to compare the behavior of the bolometer- and thermopile- based sensors. They use the simulated results from this model and compare them to the results obtained by experimentation. Figure 31 provides the theoretical model for the 3x3 bolometer-based sensors, and shows their sensitivities at varying pressure readings.

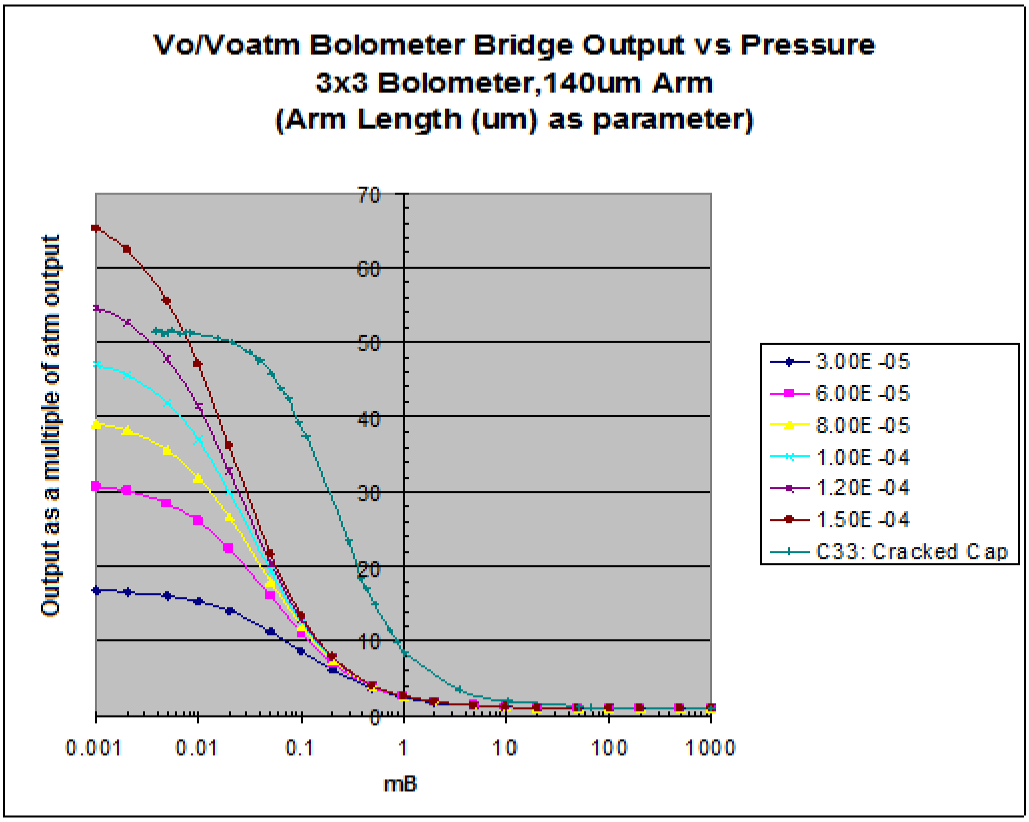


Figure 31: Theoretical model for the 3x3 bolometer.

According to this model, as the pressure decreases, the step response of the bolometer-based sensor increases in a shape that resembles a backwards "S" when plotted on a semi-logarithmic scale.

Figure 32 shows the experimental results we obtained from the step responses from various 3x3 bolometers when we varied the pressure.

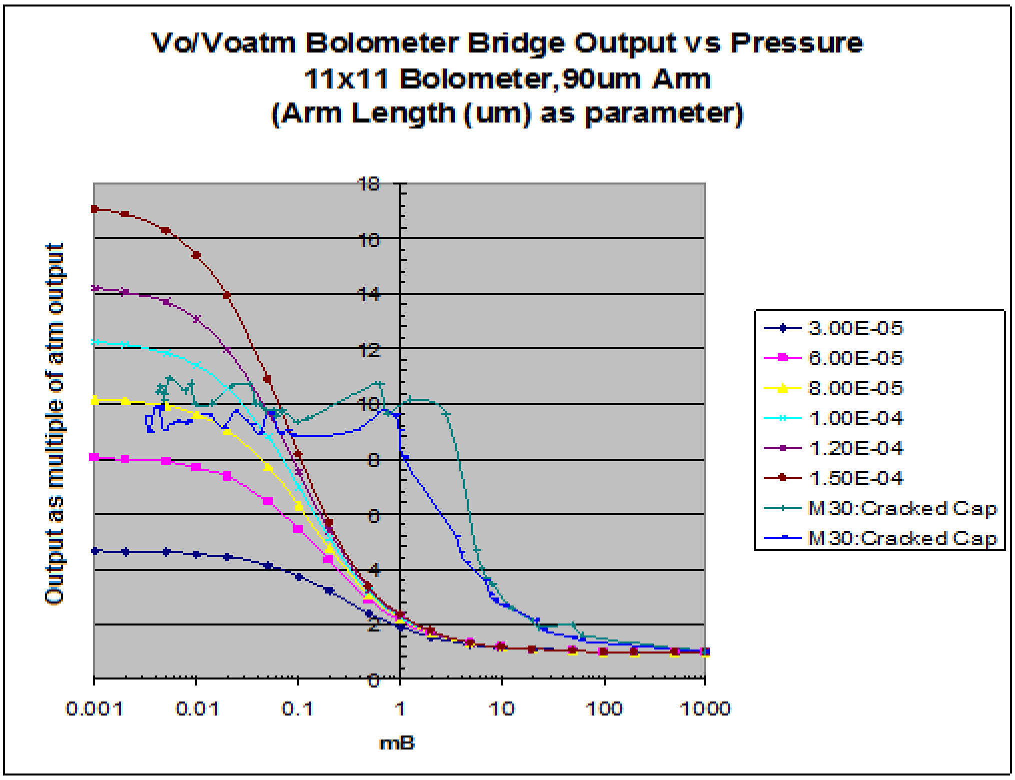


Figure 32: Experimental results for the 3x3 bolometer.

Like the theoretical model in Figure 31, the curves in Figure 32 exhibit the backwards "S" shape when plotted on a logarithmic scale. Also, the maximum outputs achieved by each of the bolometers are similar to the maximum outputs predicted by the theoretical model. The difference between Figure 31, the theoretical model, and Figure 32, the experimental results, is that there is a right shift in the experimental pressure curve. There are two possible causes of this discrepancy. One is our inability to maintain the same pressure inside the sensor as that in the vacuum chamber. This issue can be resolved by allowing the vacuum chamber more time to settle at a particular pressure value before we take a reading. Another possible cause is an inconsistency with the manufacturing process. ADI is currently investigating this problem and attempting to increase the uniformity among all parts.

Figure 32 also shows that the 3x3 bolometers generally exhibited the same behavior. Part I5's relatively inconsistent behavior could have been due to human error or an inconsistency in the part itself. However, since the other five parts behaved in the same fashion, as predicted by the theoretical model, we believe that we have provided data that can be used to improve the accuracy of ADI's theoretical model for 3x3 bolometers.

5.1.3. 11x11 Bolometer Theoretical Model vs. 11x11 Experimental Model

We found that the experimental behavior of the 11x11 bolometers was similar to their predicted behavior from the theoretical model. Figure 33 shows the theoretical pressure curves for any 11x11 bolometer as well as experimental results obtained for part M30.

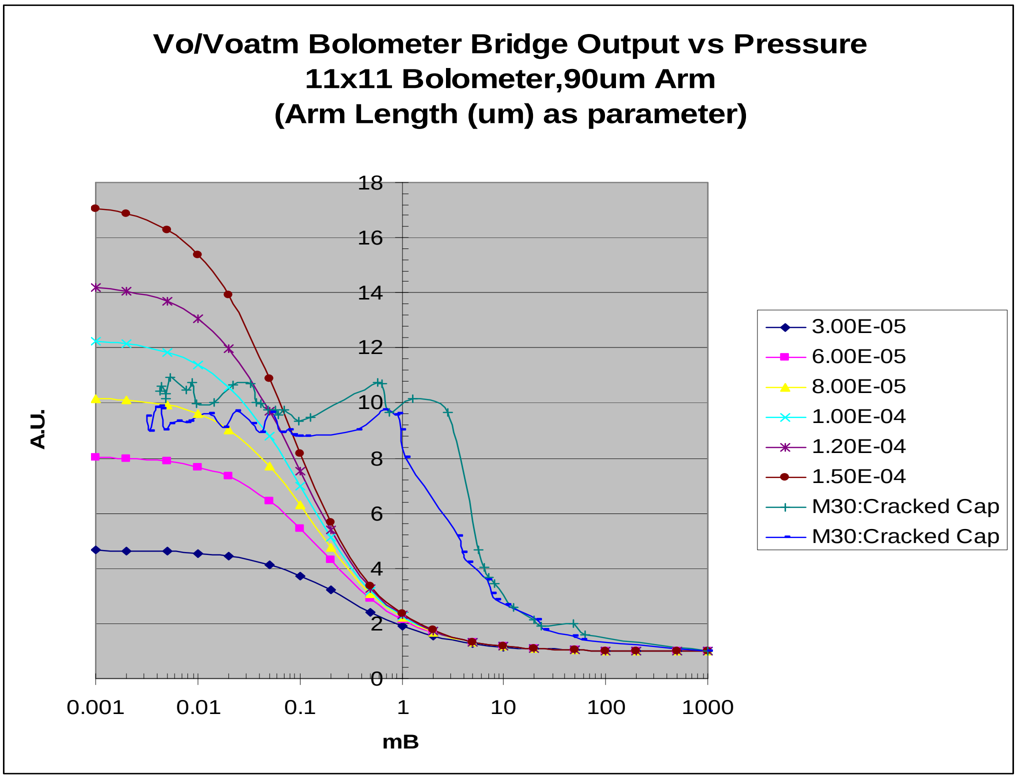


Figure 33: Theoretical model for the 11x11 bolometer.

Although the theoretical curves in Figure 33 exhibit the same shape as the curves in the theoretical model for the 3x3 bolometers, the theoretical output for the 11x11 bolometers is significantly lower; it ranges between four and sixteen instead of between fifteen and sixty-five.

Figure 34 shows the experimental results obtained for two 11x11 bolometers.

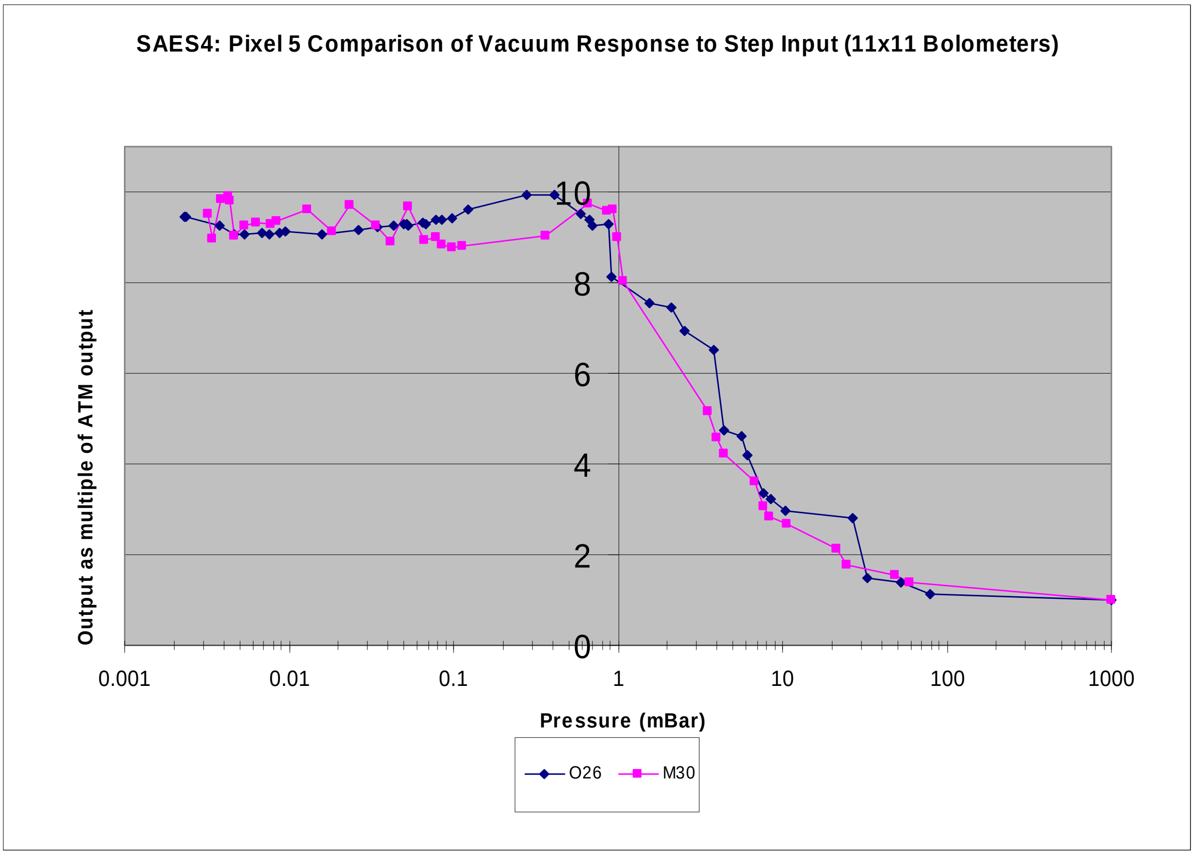


Figure 34: Experimental results for 11x11 bolometers.

The curves shown in Figure 34 are similar to the theoretical curves shown in Figure 33. Like the difference between the theoretical model and the experimental results for the 3x3 bolometers, the difference between the theoretical model and experimental results for the 11x11 bolometer is that the experimental results are shifted to the right. The explanation for this discrepancy is identical to what was described in the previous section.

5.1.4. 3x3 Bolometers vs. 11x11 Bolometers

Figure 35 shows the comparison of the sensors response from the 3x3 and 11x11 bolometers on the same graph.

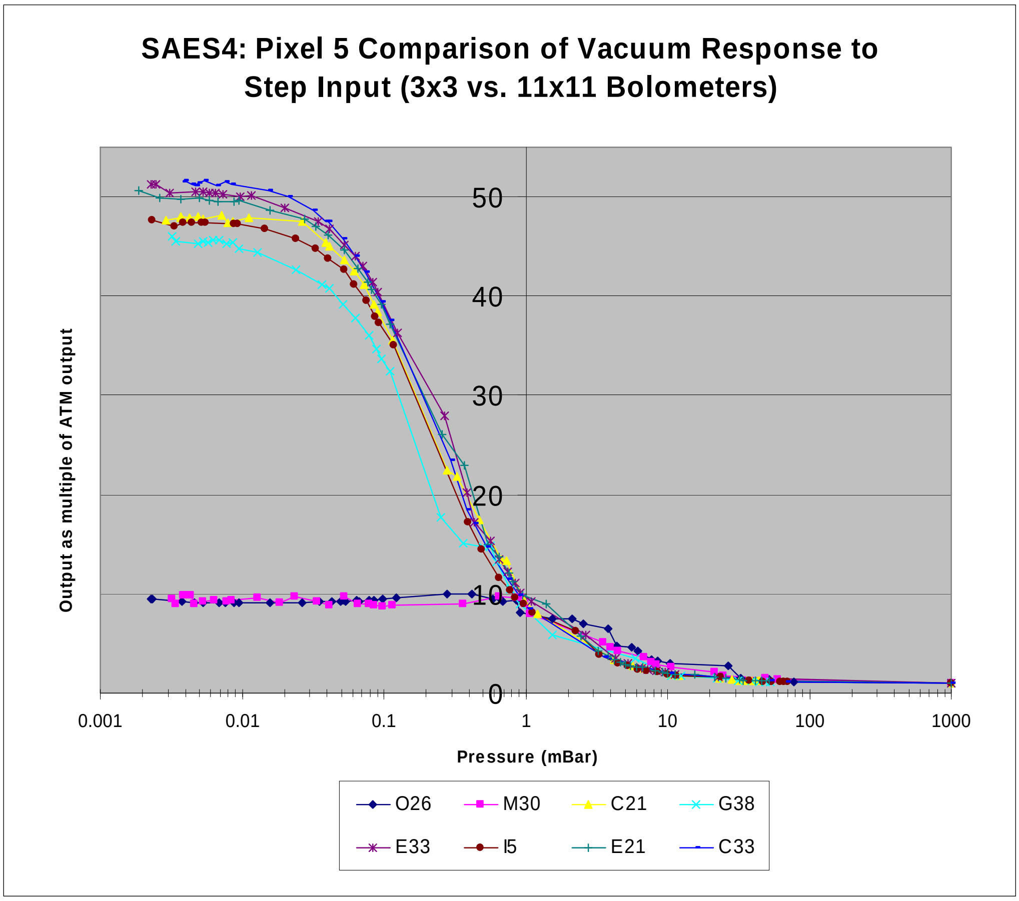


Figure 35: 3x3 bolometers vs. 11x11 bolometers

As we can see from Figure 35, the 3x3 bolometer-based sensors are more sensitive at pressures near vacuum, as their output response values are around 50, while those of the 11x11 bolometer-based sensors is 10 This is due a difference in the size of the pixels: the pixel size in the 3x3 bolometers is larger than that of the 11x11 bolometer. Since the size of pixel 5 in the 3x3 bolometer is larger than the size of pixel 61 in the 11x11 bolometer, the 3x3 bolometers obtain a higher response to IR, making them more responsive at lower pressures.

We have tested approximately twenty bolometers and thermopiles to accurately compare the experimental and theoretical results. These individual plots have been attached in Appendix D.

5.2. Post-Automation Pressure Experiment

To confirm that our results from the post-automation experiment were valid, we conducted two pressure experiments on part C33, a 3x3 bolometer: one using the pre-automation setup and procedure, and the other using our post-automation setup and procedure. When we compared the results, we found that our automated software not only decreased the human-machine interaction time for the part from sixty-five minutes to five minutes, but it also provided results at pressures equally spaced on a semi-logarithmic scale. As we can see from Figure 36, before we automated the pressure experiment, we were unable to gather a significant amount of data at pressures between 0.1 mBar and 1 mBar.

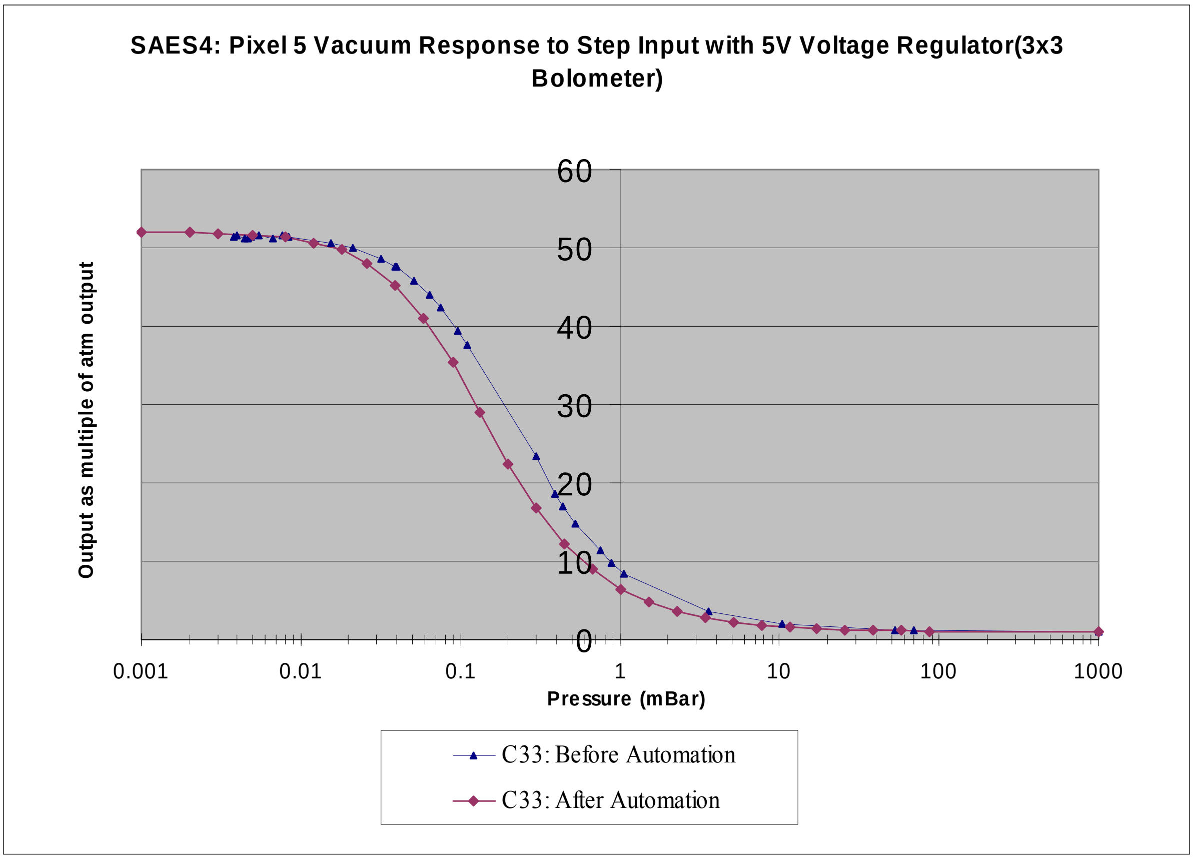


Figure 36: Comparison between the pre- and post- pressure experiment results for part C33.

Our automation software solved this problem by simply setting the desired target pressure on the vacuum controller. Before automation, we manually tightened or loosened a valve to increase or decrease the pressure inside the vacuum chamber. The results from the automated pressure experiment can be seen in Figure 36.

We observed that the post-automation pressure experiment produced similar results to the pre-automation pressure experiment. Figure 36 shows that the sensor’s highest response is located at low pressures, between 0.01 mBar and 0.001 mBar. In both the pre- and post-automation experiments, the output at these pressures is approximately 52. This similarity in the maximum output suggests that our post-automation pressure experiment is repeatable. Also, we observed that the behavior of the sensor during the post-automation pressure experiment was consistent with the behavior of the sensor during the pre-automation experiment; the sensor's response in both experiments increased as the pressure inside the vacuum chamber decreased. The pressure curves in Figure 36 are similar to those predicted by the theoretical model, confirming the repeatability post-automation pressure experiment.

5.3. Vacuum Degradation Experiment

As mentioned in Section 4.2, we conducted the aging test on two groups of parts: 11x11 bolometers and thermopiles, and 3x3 bolometers. When we tested the 11x11 bolometers and thermopiles, we compared their responses to IR when stored in extreme temperatures to one another. Since some of the 3x3 bolometers were manufactured using a new process, we compared the responses of the bolometers manufactured with the new process to the bolometers manufactured with the old process.

5.3.1. 11x11 Bolometers and Thermopiles

Figure 37 and Figure 38 contain the step response graphs for the 11x11 bolometers and thermopiles, respectively. Step response indicates the sensitivity of the ADiR sensor when exposed to IR due to a change in temperature. The y-axis corresponds to the sensitivity of the ADiR sensor when measured on different days. Since several of the parts broke during testing, we decided to include only the data points from functional parts: M15, M35, M27, O7, O31 and D19.

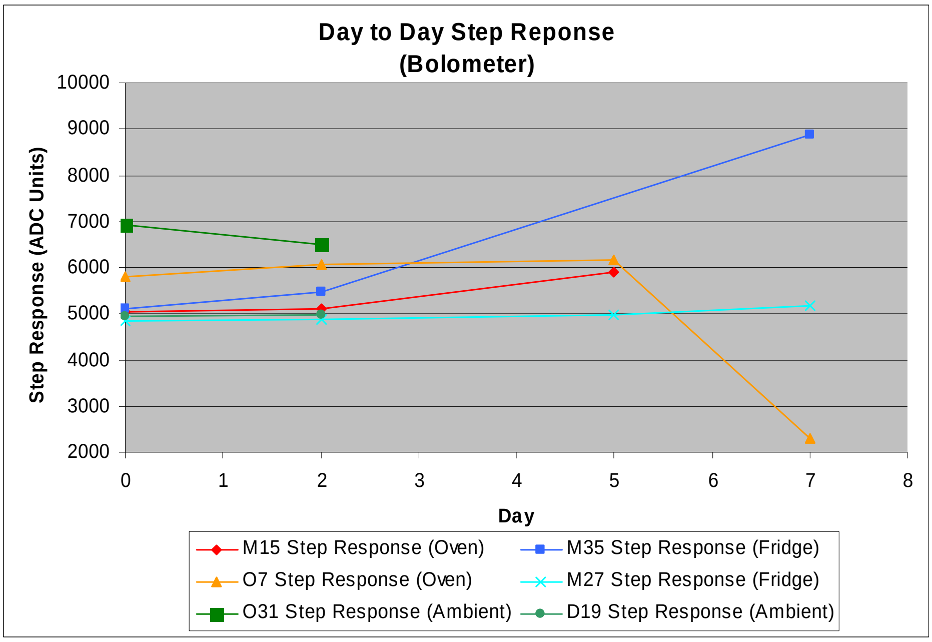


Figure 37: Day to day step response of bolometers.

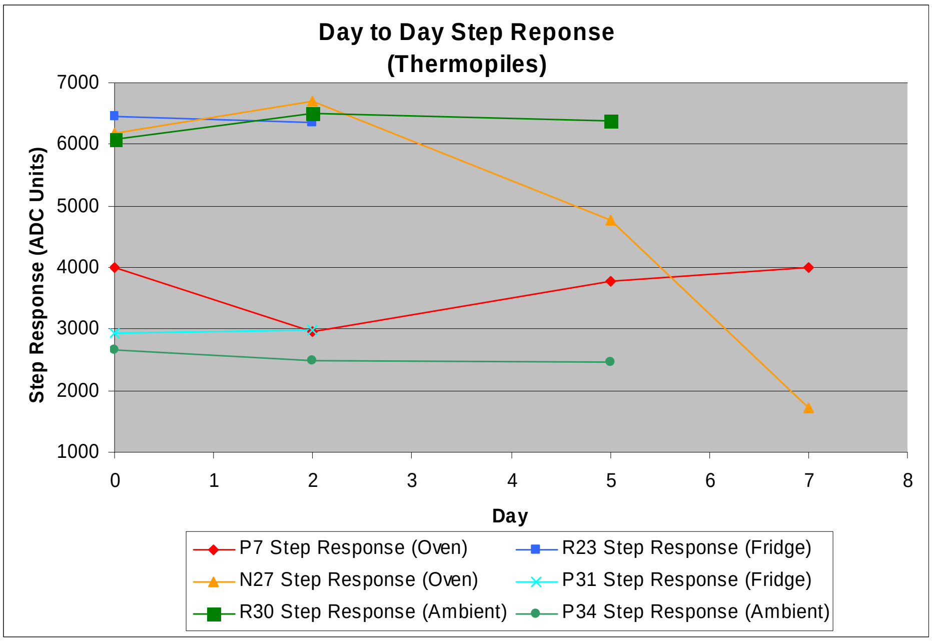


Figure 38: Day to day step response of thermopiles.

The results from the step response tests show that storage at high temperatures decreases the step response for both thermopiles and bolometers, and storage at room temperature and in a refrigerator does not affect the step responses of either the bolometers or thermopiles. Since step response is a good measure of sensitivity, we determined that storage of the ADiR sensors at high temperatures reduced the sensitivities of both the bolometer and thermopile based sensors.

Figure 39 and Figure 40 plot the absolute value of the difference between the sensors' offsets on day n (n = 0, 2, 5, and 7), and the initial day (day 0). When we exposed the sensors to temperatures of 10°C and 30°C, LabVIEW obtained the output from all 121 pixels in each of the sensors. We took the median of all of the 121 pixels for each sensor at both 10°C and 30°C. We collected data points on the day 0 and then on days 2, 5, and 7. Each day, we calculated the offsets for each sensor by subtracting the median reading at 10°C on that day from the median reading at 10°C on day 0, and repeating this process for each sensor at both 10°C and 30°C. The median offsets of the thermopiles are shown in Figure 39.

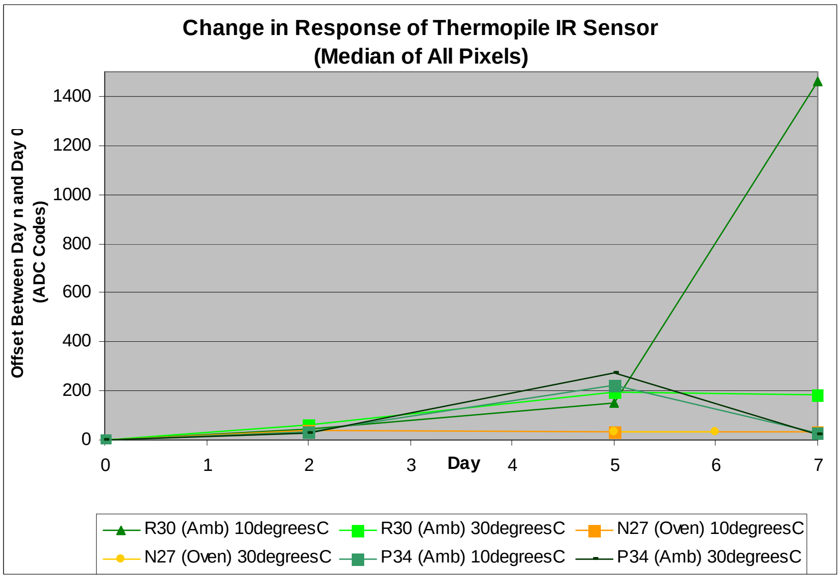


Figure 39: Change in thermopile response.

The median offsets of the bolometers are shown in Figure 40.

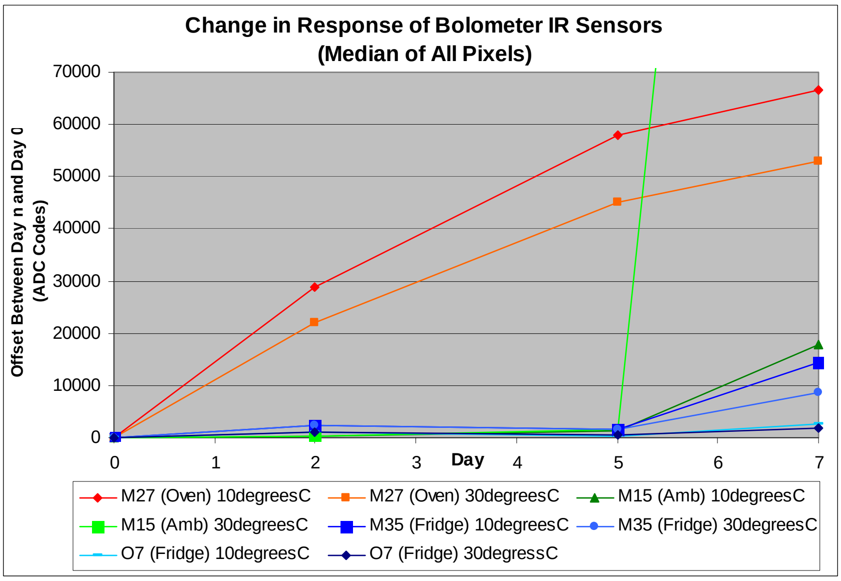


Figure 40: Change in bolometer response.

In Figure 39, the offsets of the thermopiles placed in the oven did not change significantly.

However, in Figure 40, the offsets of the bolometers placed in the oven changed considerably. Therefore, we concluded that the effect of aging is more prominent on bolometers than on thermopiles. Even though the aging was simulated, these results gave us a fair indication that in the long run, thermopile based sensors would be more stable than the bolometer based ones. This characteristic of the thermopiles could serve as a unique selling point in the automotive and industrial sectors.

During the temperature forcing test, we found that several parts were unresponsive. The 30^{o}C values appeared to be different than the values expected. We confirmed that these parts were broken when we found them to be unresponsive when exposed to the blackbody source in the lab. At the end of the last day only two thermopiles and three bolometers out of the original twelve were left. Table 5 summarizes the days when each part broke, if applicable.

Table 5: Days when each part broke.

Type	Part Number	Storage Location	Day Broken
Bolometer	M15	Oven	7
Bolometer	O7	Oven	-
Bolometer	O31	Room	5
Bolometer	D19	Room	5
Bolometer	M35	Refrigerator	-
Bolometer	M27	Refrigerator	-
Thermopile	P7	Oven	-
Thermopile	N27	Oven	-
Thermopile	R30	Room	7
Thermopile	R23	Room	7
Thermopile	P31	Refrigerator	5
Thermopile	P34	Refrigerator	5

Since more than half of the parts broke, we believed that there was a problem with our test setup. When we observed that one part broke while it was in the temperature forcer and that the bond wires in each of the sensors appeared bent, we hypothesized that the air flowing from the temperature forcer was breaking or shorting the bond wires. To solve this problem, we decided to place a cover over the sensor in the tests on the 3x3 bolometers.

5.3.2. 3x3 Bolometers

When we conducted the aging test on five 3x3 bolometers that were made by the new manufacturing process, we stored them in an oven for three days. We measured the offsets of two of the sensors at the same temperatures: 10^{o}C and 30^{o}C. To obtain readings at one of these temperatures, we placed each sensor under the temperature forcer before and after storage and used LabVIEW to gather samples of the ADC readings from each pixel. To save time, ADI instructed us to conducted the temperature cycling test on only two of the five 3x3 bolometers. Figure 41 shows the offsets of one of the bolometers at 10^{o}C before and after being stored in the oven.

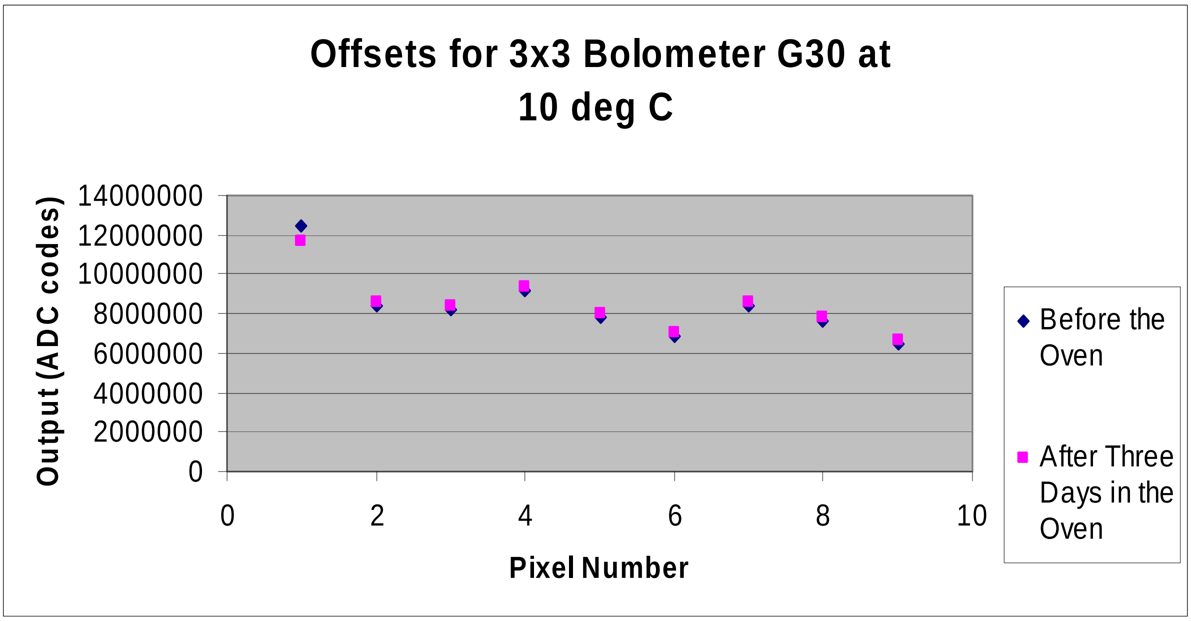


Figure 41: Offset of a 3x3 bolometer manufactured by the new process at 10^{o}C.

We found that for both of the parts, the change in the offset was very small at both 10^{o}C and 30^{o}C. The graphs of the offsets for part G30 at 30^{o}C and for part I34 at 10^{o}C and 30^{o}C can be found in Appendix E.

We found that the step responses for the parts did not significantly change, either. Figure 42 shows the step responses of the 3x3 bolometers that were made by the new manufacturing process.

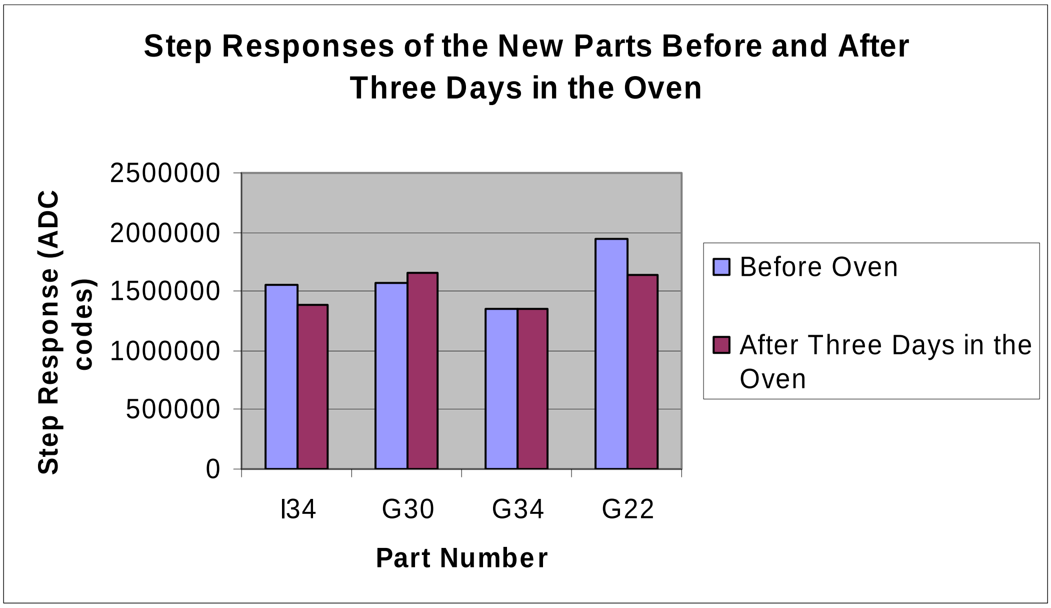


Figure 42: Step responses of the 3x3 bolometers manufactured by the new process.

ADI compared our results obtained for the parts made by the new manufacturing process to their results for the 3x3 bolometers made by the old manufacturing process and informed us that the newly manufactured parts displayed less sensitivity to heat exposure than the parts made by old manufacturing process.

5.4. Angular Response Experiment

The main purpose of the angular response test was to observe the change in sensitivity of the IR sensor with and without the multi-stepped lens technology when rotated 180^{o}. This lens on the sensor is a diffractive single-step phase lens that focuses radiation on the center pixels. Since this was the first time that this lens technology had been used on the sensor, it was essential for us to perform this test to determine whether the lens technology functioned properly. To perform this test, we developed a LabVIEW VI that would move the one-axis rotational stage and communicate with the computer to process the data.

A full description of the test setup utilized for this experiment is available in Section 4.4. We first tested a sensor that had the lens pattern on it and saw that the center pixel was more responsive than the outside pixels when rotated on the horizontal axis. Our results are shown in Figure 43. In pixels 1-3 and 7- 9, the response was not as strong as that in pixels 4-6, which form the center row of the sensor. From Figure 43, we can infer that the lens operated as expected by focusing the IR onto the center pixels. This lens technology could be used in applications that require motion detection because it provides angular response.

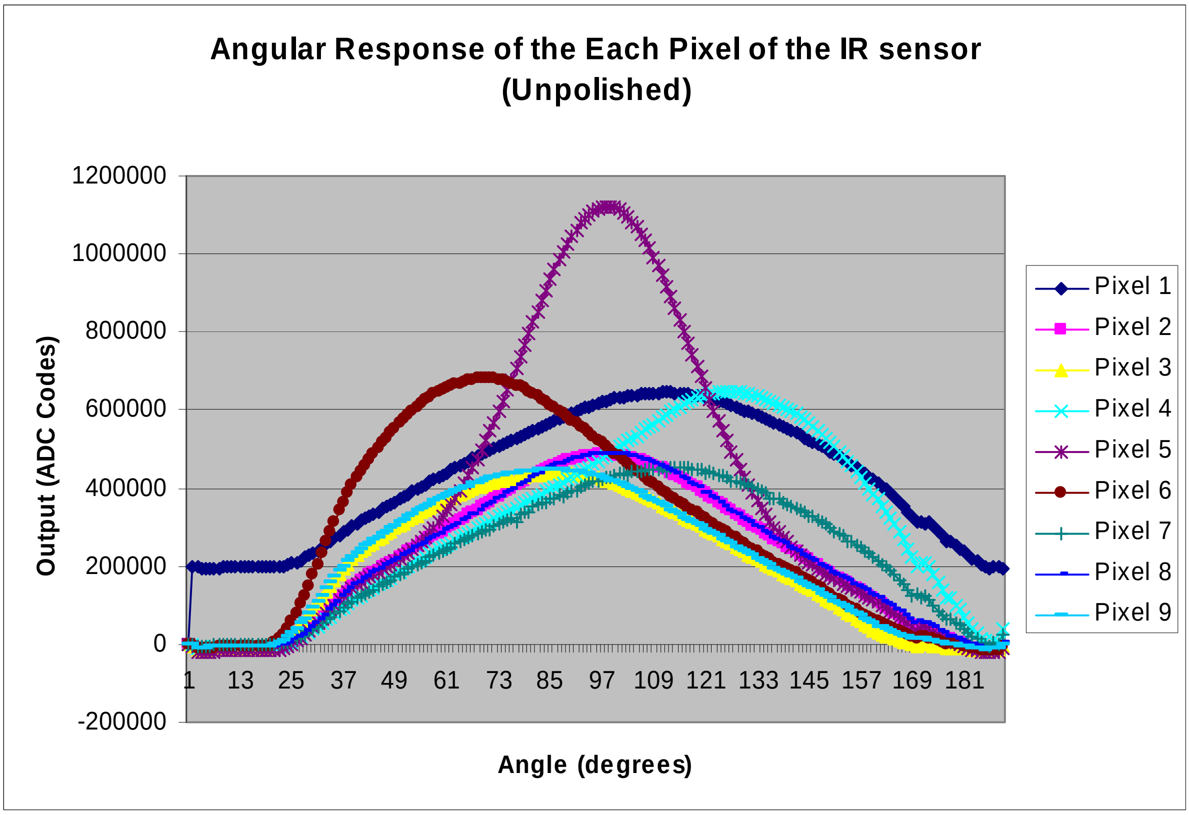


Figure 43: Angular response of the unpolished lens before we automated the test.

Figure 44 shows the angular response of the sensor after we removed the lens by polishing it.

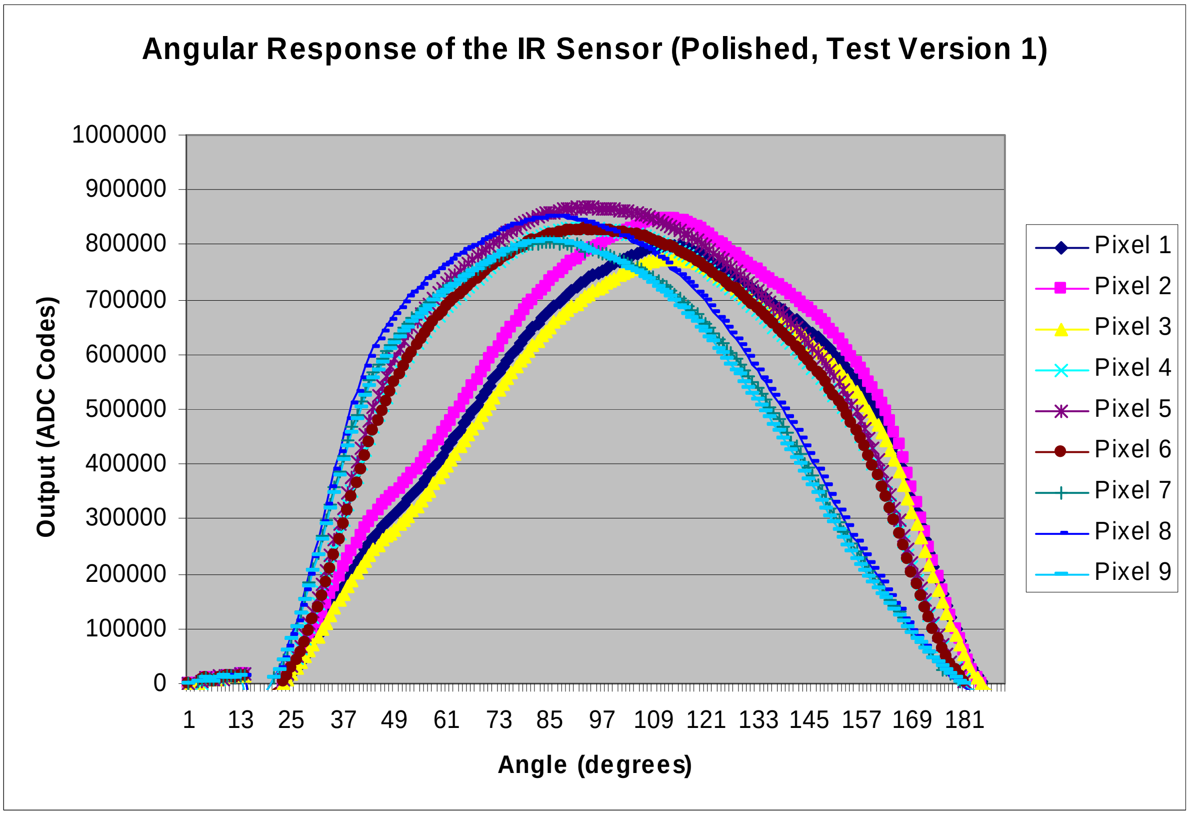


Figure 44: Angular response of the sensor with the lens polished.

We expected that without the lens, the pixels would exhibit uniform angular response, and our results from Figure 44 confirmed this hypothesis.

After we had completed the testing, we realized that we could expedite the process by removing unnecessary modules from our LabVIEW VI. When we ran the first version, it took around thirty minutes for the test to complete whereas after we made the intended modifications it took only five minutes. We plotted the results from this quicker test and found that the results were consistent with those from the slower test.

5.5. Lens Focusing Experiment

The main purpose of the lens focusing experiment was to find the position of the ADiR sensor in three-dimensional space where it obtains the highest response to IR radiation. For the comprehensive scan, we placed the ADiR sensor in front of the lens that focuses the IR radiation provided by the blackbody source. To perform this test, we created a LabVIEW VI that moved the sensor and recorded its response as it moved in all the three dimensions.

A full description of the test setup utilized for this experiment is available in Section 4.5.1. The x-axis in this experiment refers to the movement of the translational stage, which houses the sensor, toward or away from the lens. This has been shown in Figure 45.

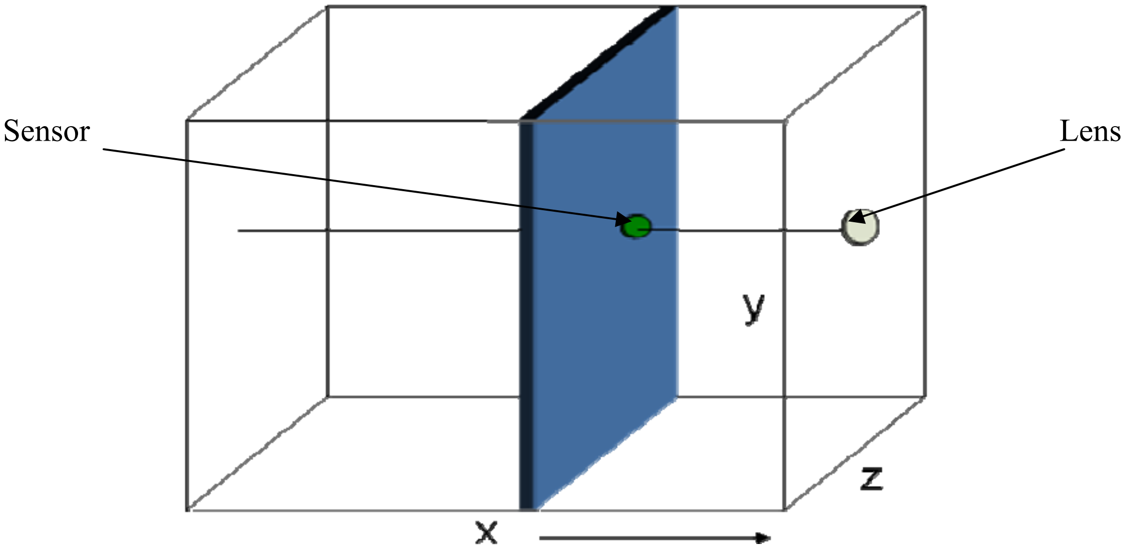


Figure 45: x-axis representation of the translational stage.

Figure 46 below shows the focused peak position of the ADiR sensor as seen from the x-axis.

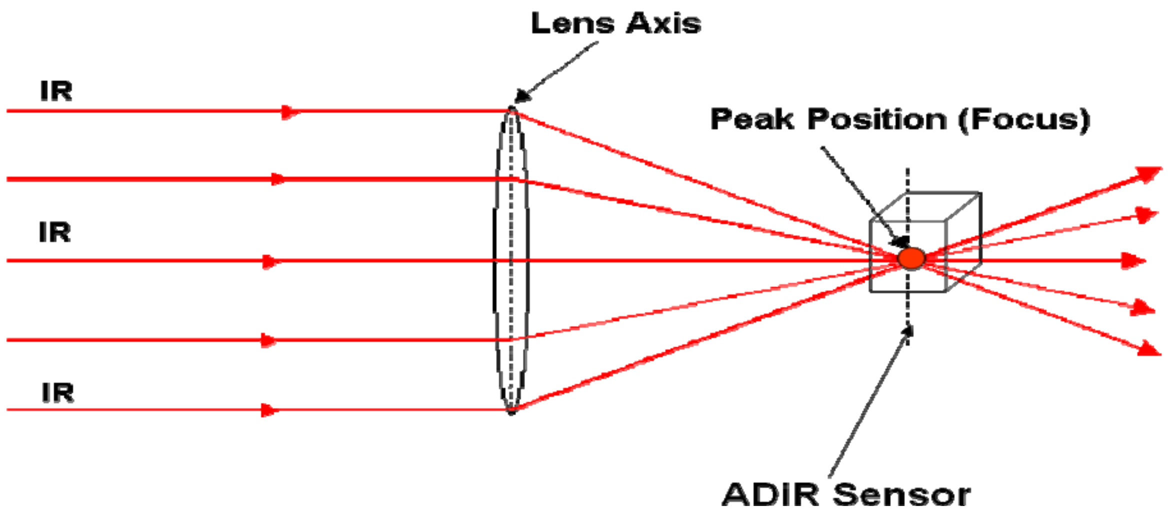


Figure 46: ADiR sensor at focal point.

The ADiR sensor is most responsive when it is at the focal point. This focal point is the peak position and can been seen in Figure 47 by the warm colors. As we move away from the center in any direction, we see that the image becomes darker, suggesting that we are moving away from the peak position. The resolution of the picture is three-thirteenths of a millimeter as the square is a 3x3 mm box divided into thirteen sections.

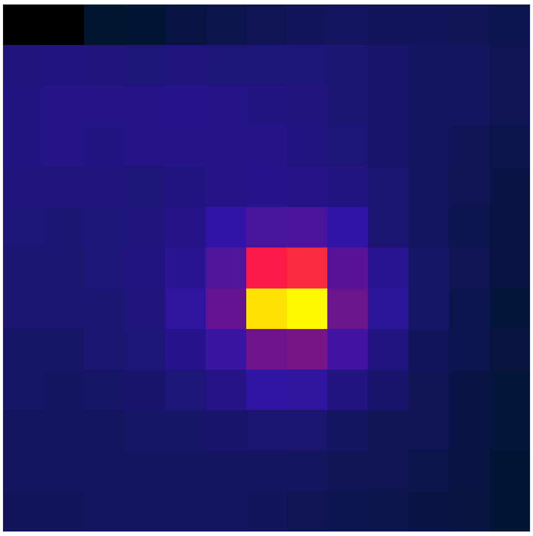


Figure 47: Focused image of the x-axis.

From Figure 48, we can see a peak in the center of the blurry image. The main reason for this blurry image is that this has not been taken at the focal point of the lens. This image is out of focus as the sensor is beyond the focal point as shown in Figure 49.

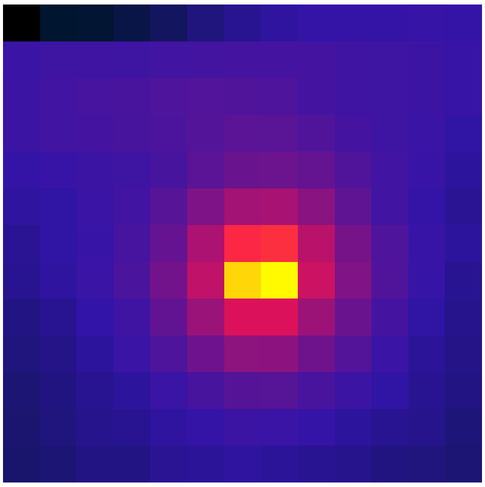


Figure 48: Wideslice image of the x-axis

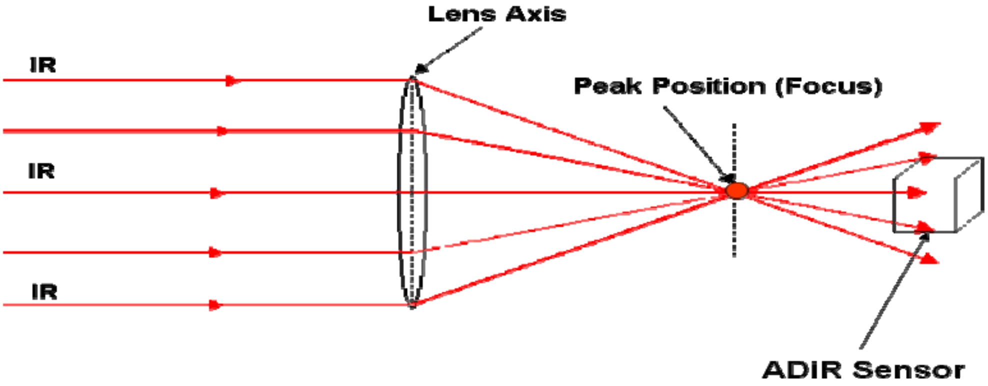


Figure 49: ADiR sensor beyond focal point.

The y-axis in this experiment refers to the movement of the translational stage vertically from the lens. This has been shown in Figure 50.

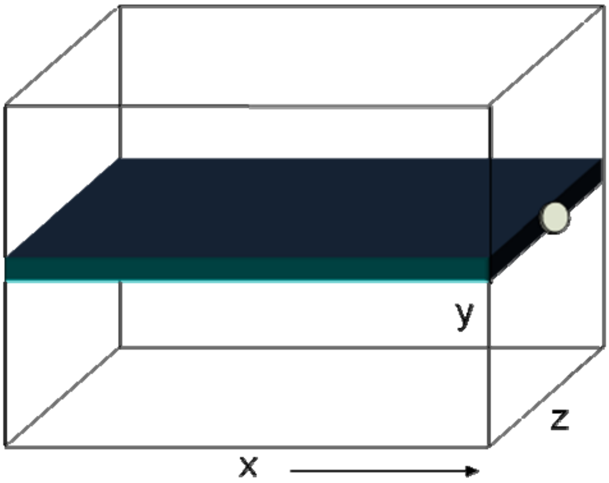


Figure 50: y-axis representation of the translational stage.

Figure 51 and Figure 52 show the focused peak position of the ADiR sensor as seen from the y- and z-axes, respectively. We can see in both the images that as we move from left to right the peak, displayed in warmer colors like yellow is observed in the center. On either side of the yellow, we see the fading nature because the lens no longer focuses to a point on the sensor.

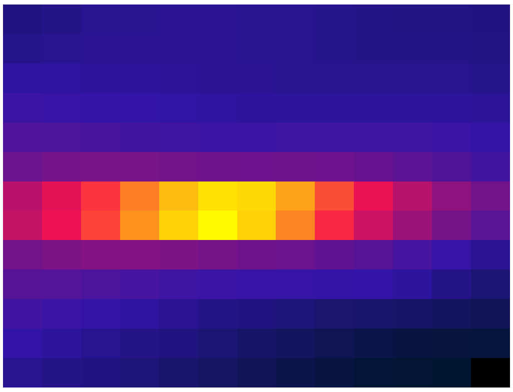


Figure 51: Focused image of y-axis.

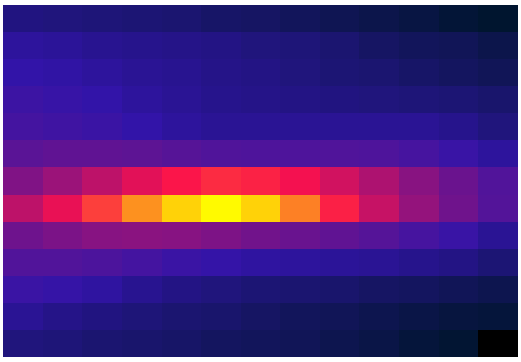


Figure 52: Focused image of z-axis.

We compared the results of the comprehensive scan with the results of the efficient scan to assess the functionality of the efficient scan. Figure 53 and Figure 54 shows a clear peak along each axis as well as a gradual increase in the magnitude of the peak relative to the first scans of the y- and z- axis scans. This suggests that the sensor is approaching the peak positions shown in Figure 47 and Figure 48. Figure 53 shows the results from the first and second scan of the y-axis.

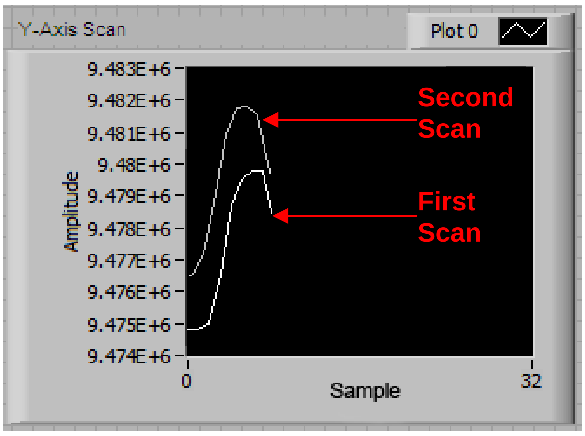


Figure 53: Results of the y- axis scan using the efficient algorithm.

Figure 54 shows the results from the first and second scan of the z-axis.

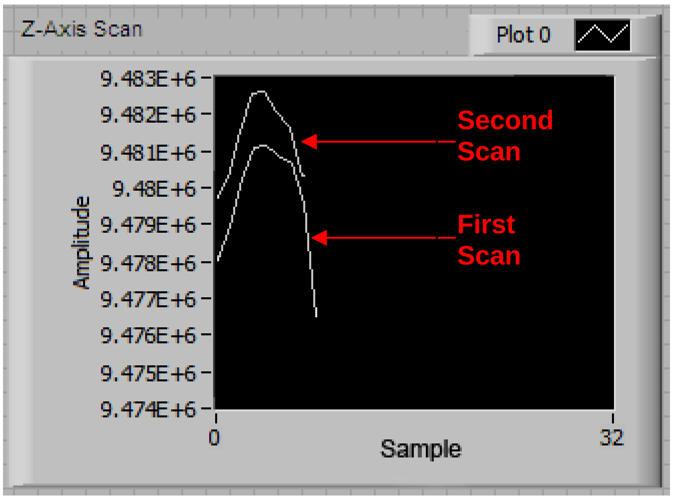


Figure 54: Results of the z-axis scan using the efficient algorithm

After the scan of the y- and z- axes, the software scans the x-axis. The results of the x-axis scan are shown in Figure 55. The figure shows an anomaly for the first data point; this could be due to a problem with the resetting of the evaluation board. Ignoring this first data point, the figure shows a clear peak in the middle of the scan.

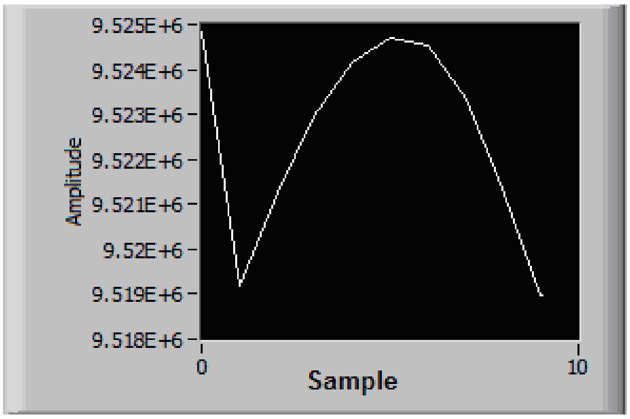


Figure 55: Result of the X-axis scan using the efficient algorithm.

5.6. *Qualification Test*

The purpose of the qualification test was twofold: to separate working parts from those that are non-functional, and to characterize the working parts. The qualification test outputs twenty-three files for each working part. The first graph, an example of which is shown in Figure 56, will be used to estimate the RMS noise seen by the ADC.

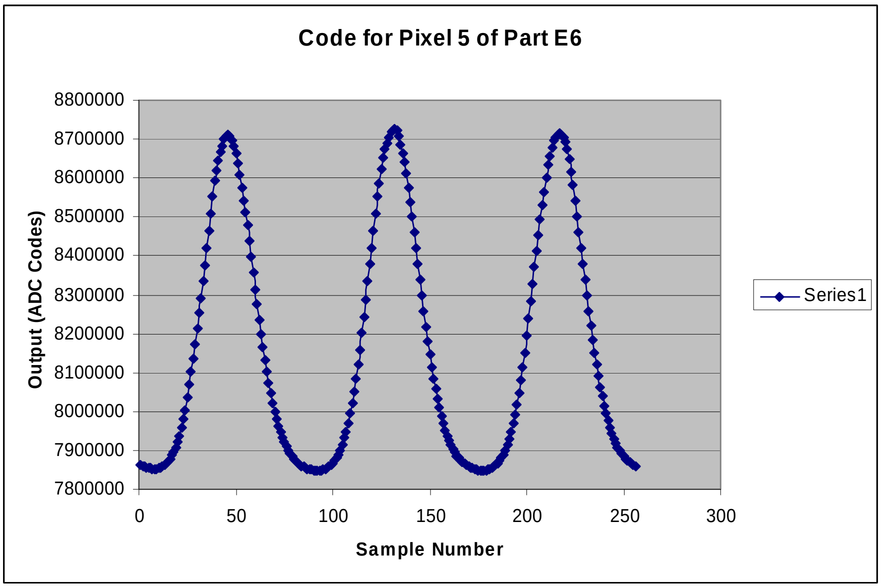


Figure 56: Raw Data for FFT.

The graph displays a part’s center pixel’s output code in response to a sine-squared IR source. The FFT of this wave will be used to calculate the RMS noise.

The second graph, shown in Figure 57, is the FFT of the data from the first graph

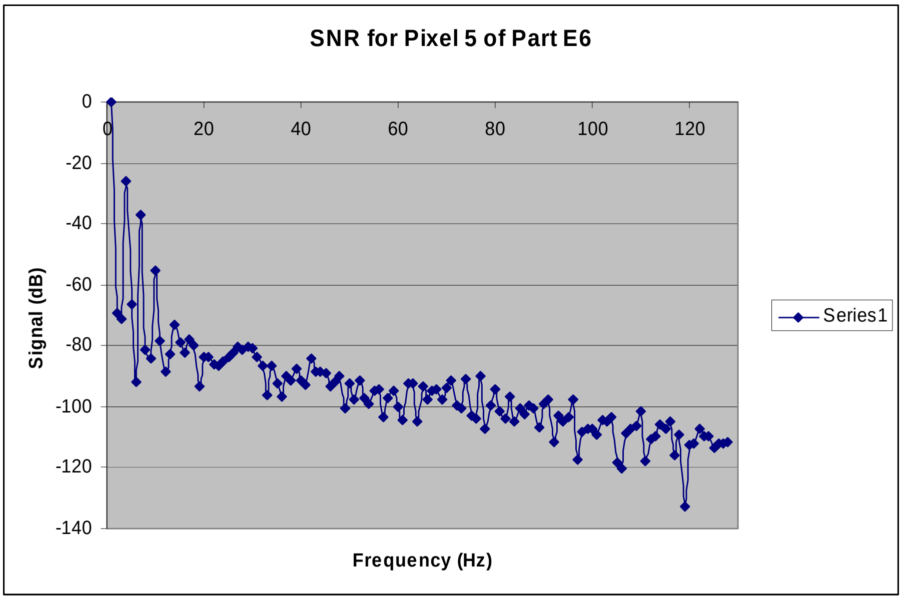


Figure 57: FFT.

The second peak in Figure 57, at 4 Hz, is the signal from which the SNR will be calculated. The next few peaks are the harmonics, which are caused by the sine-squared function, as well as the difference in the time required for the sensor to heat and cool. Since they are the harmonics, they are disregarded in the SNR. The noise from which the SNR is calculated is located at frequencies other than DC, the fundamental frequency, and the harmonics, and it is caused by the inability of the source to cool as quickly as it heats. Once ADI knows the magnitudes of the signal and noise in dB, they can convert the values from dB and calculate the SNR.

The third graph, shown in Figure 58, is a plot of the power of the fundamental frequency for each pixel.

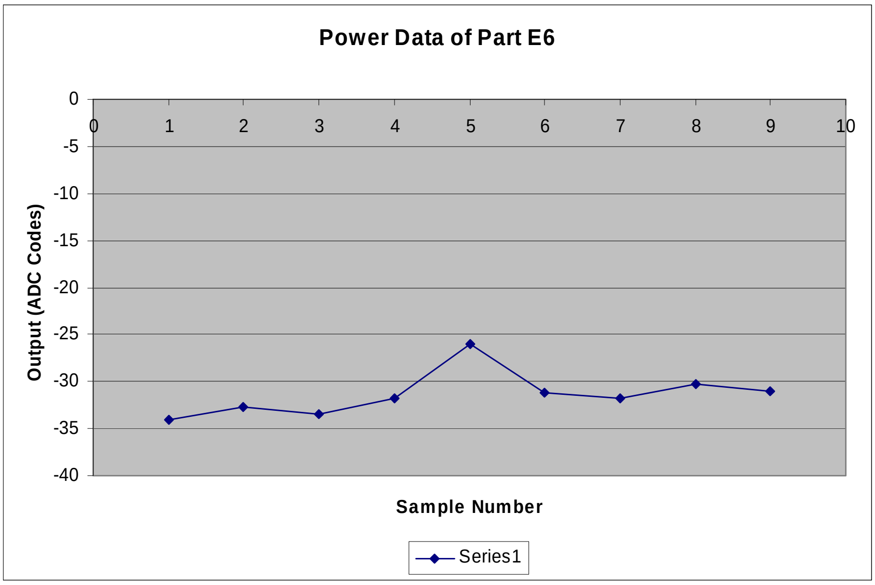


Figure 58: Power of the fundamental frequency for each pixel.

Since the signal in pixel 5 was the highest, its power was also the highest.

Figure 59 shows a schematic of the Wheatstone bridge arrangement that was used to gather data about the resistors.

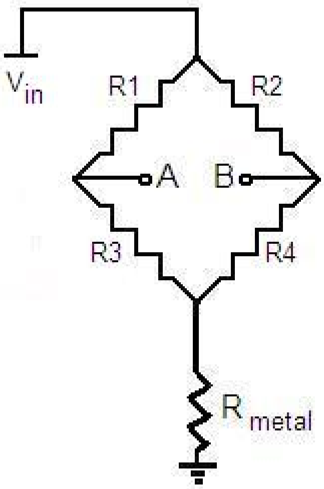


Figure 59: Wheatstone bridge.

Figure 60 is a graph of R1 in the Wheatstone bridge under both 3V and 100mV.

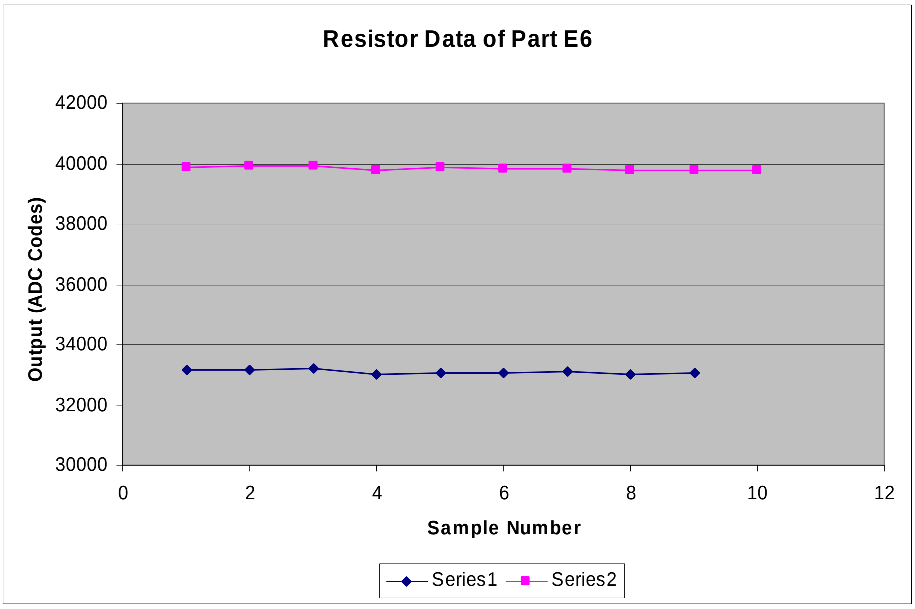


Figure 60: Value of one of the resistors in the Wheatstone bridge under 3V (pink) and 100mV (blue).

ADI will use the graph in Figure 60 to analyze the self-heating effect of the resistors. At higher voltages, the self-heating effect is greater.

Figure 61 graphs the temperature coefficient of resistance (TCR) mismatch for each of the nine pixels. TCR is a measure of the amount a resistor changes as a function of temperature, and TCR mismatch is the difference between the TCR of the resistors on the left and right hand sides of the Wheatstone bridge. TCR mismatch is undesirable because it decreases the accuracy of the response obtained from the sensor.

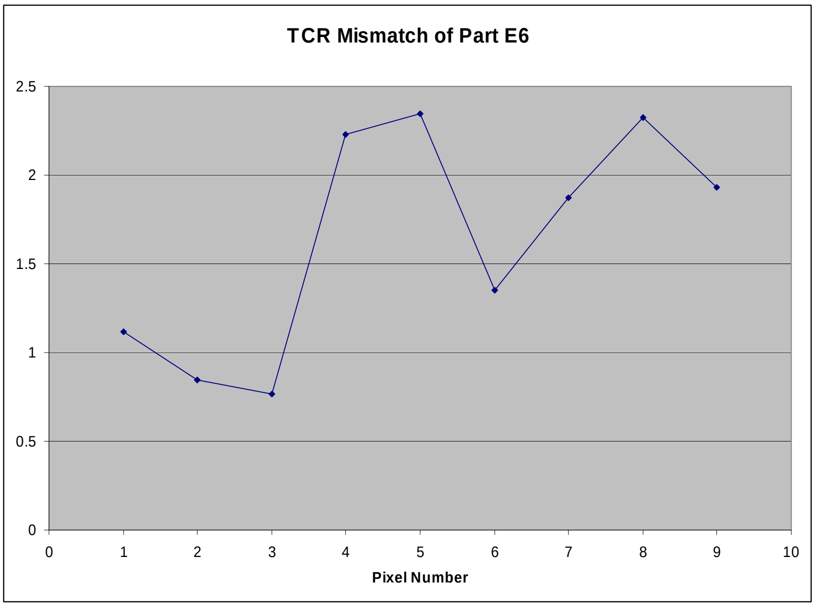


Figure 61: TCR mismatch for each pixel.

The tester obtains the data plotted in this graph by measuring the resistance of each of the four resistors in the Wheatstone bridge at 25°C and then at 100°C. Equation 3 is then used to calculate the TCR mismatch.

Equation 3: Equation for TCR mismatch.

$$TCR\_mismatch = \frac{R_{hot} - R_{cold}}{R_{cold} * Temp\_Change} * 100$$

For this test, R_{hot} is the resistance of one of the resistors at 100°C, R_{cold} is the resistance of the same resistor at 25°C, and Temp_Change is 75°C.

Figure 62 is a graph of the resistance of R2 when an electrical step is applied at R3. This graph is important because it shows how the value of one of the resistors changes as heat is electrically applied.

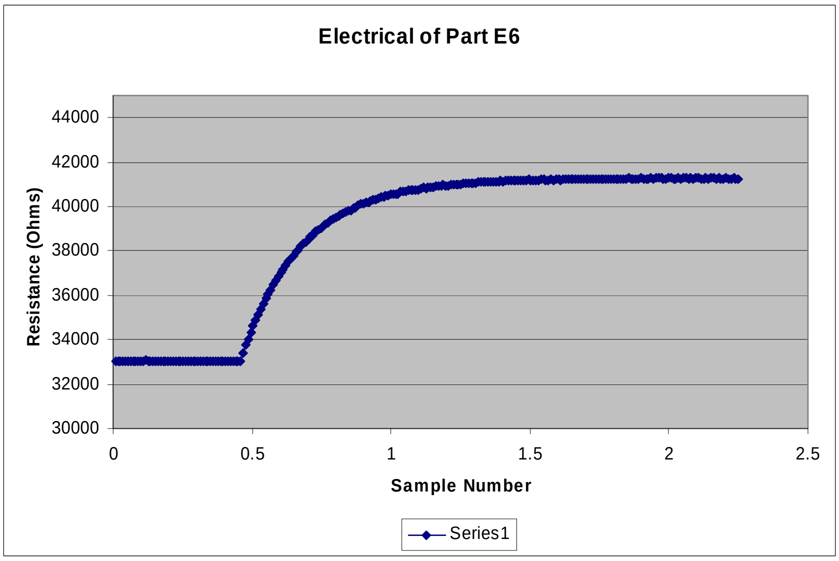


Figure 62: R_2's resistance in response to an electrical step input.

Figure 63 shows where the voltages are being input when the electrical step response is conducted.

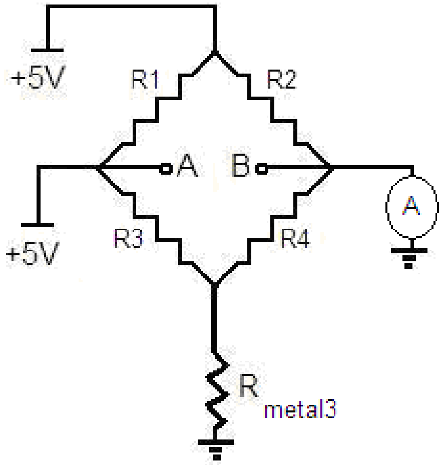


Figure 63: Schematic of electrical step response.

When a step input of 5V is applied to the R_1R_3 node, the current through R_{metal3} doubles. This causes a step change in the error of the measurement. To remove this additional error, we calculated the change in R_2 after the step change had been applied, and added this offset to the each of the subsequent values of R_2.

The last parameter measured by the tester that we graphed in Excel was the sensor's optical response. In this test, a radiation source is switched on and V_{AB}, depicted in Figure 63, is measured. An example of the graph generated for the optical step response is shown in Figure 64.

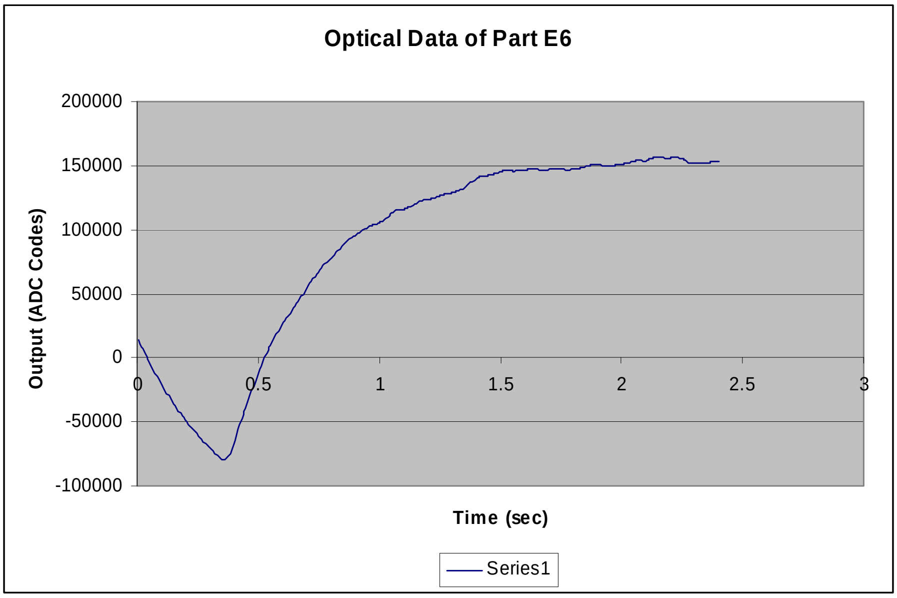
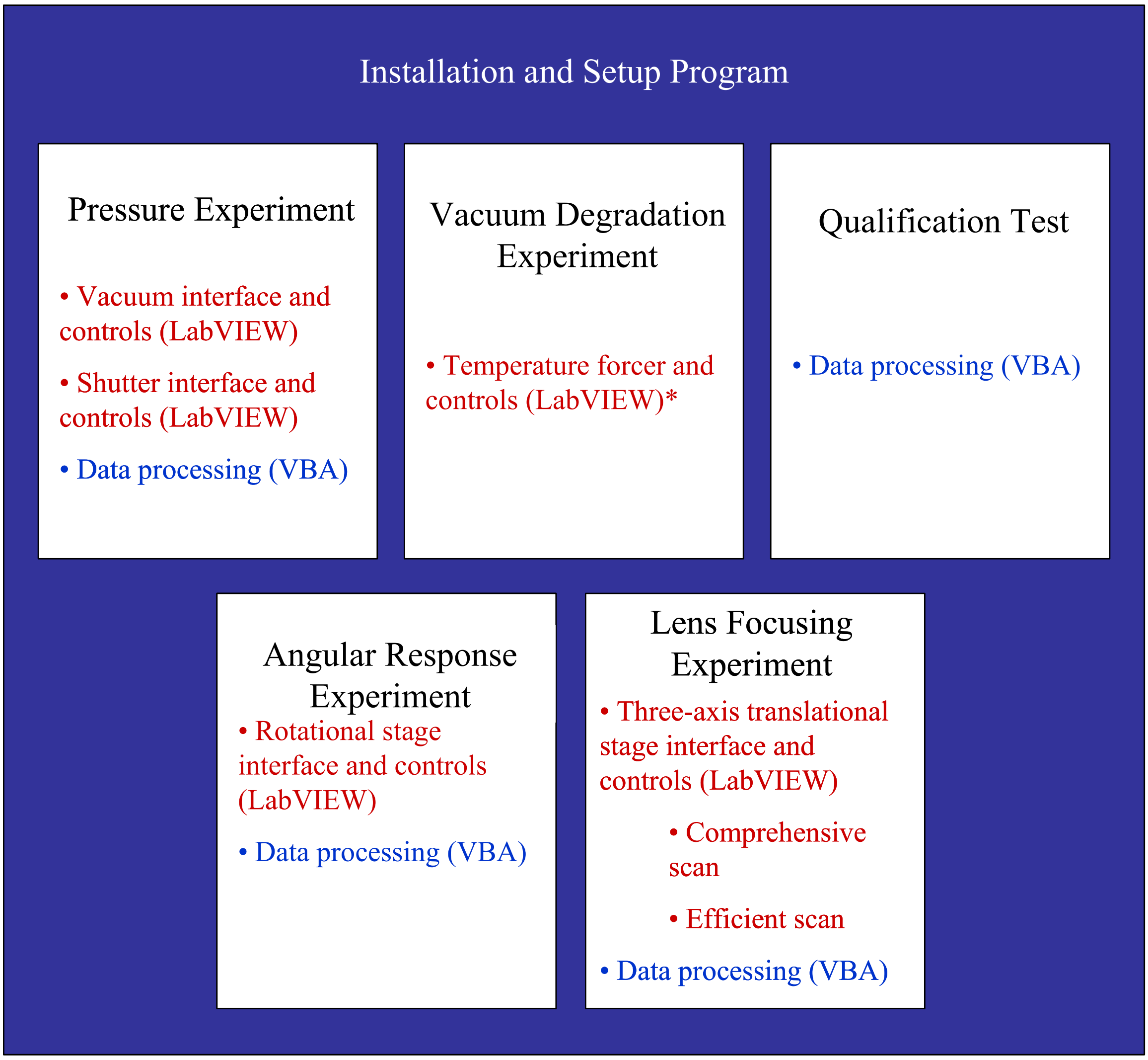


Figure 64: Optical step response graph.

In Figure 64, some of the ADC codes are negative because of the drift caused by the TCR mismatch. In this case, the TCR drift was negative, so the data points before the step response appear negative

6. ADiR Experimentation and Data Processing Suite

Our deliverable to ADI consisted of a suite of tools that simplify and reduce the time necessary to conduct the five experiments described in Section 4.5. This section will describe how these tools combine LabVIEW VIs and Visual Basic applications to systematically perform the experiments and quickly process the raw data. The components of the suite are outlined below in Figure 65.



Installation and Setup Program

Pressure Experiment
- Vacuum interface and controls (LabVIEW)
- Shutter interface and controls (LabVIEW)
- Data processing (VBA)

Vacuum Degradation Experiment
- Temperature forcer and controls (LabVIEW)*

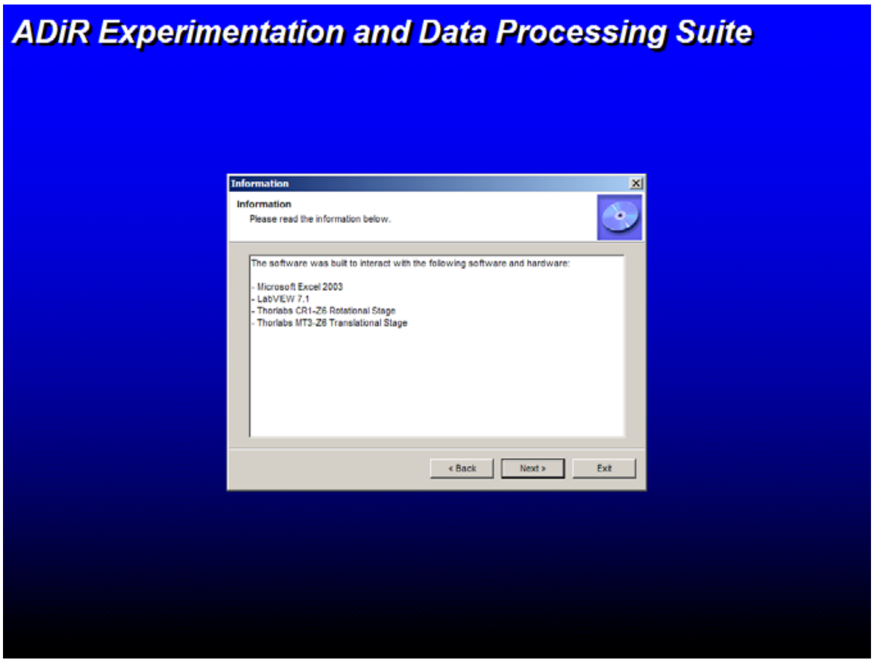
Qualification Test
- Data processing (VBA)

Angular Response Experiment
- Rotational stage interface and controls (LabVIEW)
- Data processing (VBA)

Lens Focusing Experiment
- Three-axis translational stage interface and controls (LabVIEW)
  - Comprehensive scan
  - Efficient scan
- Data processing (VBA)

Figure 65: Software hierarchy. * denotes a piece of software that ADI provided for us to modify.

The installation and setup program provides ADI with a quick way to deploy the five tools onto any PC and to ensure that all the components are present. Once completed, the installation program places the files in a proper directory as well as creates shortcuts to each component in the Windows start menu. A screenshot of the installation program can be seen in Figure 66. The installation program was created to be straightforward and intuitive to any PC user.



Please read the information below.

The software was built to interact with the following software and hardware:

- Microsoft Excel 2003
- LabVIEW 7.1
- Thorlabs CR1-Z6 Rotational Stage
- Thorlabs MT3-Z8 Translational Stage

Figure 66: Screenshot of software installation.

6.1. Experimentation Interfaces and Controls

Before our project began, ADI possessed LabVIEW modules to communicate with the IR evaluation board and temperature forcer. However, the software for the temperature forcer required slight modifications and the software to communicate with a vacuum controller, three-axis translational stage, rotational stage, and blackbody source was not present. We created the interface to communicate with all these devices and then created or modified the software to execute the pressure, vacuum degradation, angular response, and lens focusing experiments.

6.1.1. Pressure Experimentation Software

The pressure experimentation software automates the methodology outlined in Section 4.1 . To accomplish this, the vacuum chamber and the shutter needed to work in a controlled way. Pfieffer and Lego both provided the necessary communication VIs to control their respective equipment, implementation of our procedure was the part missing.

A flow chart of the software is shown below in Figure 67.

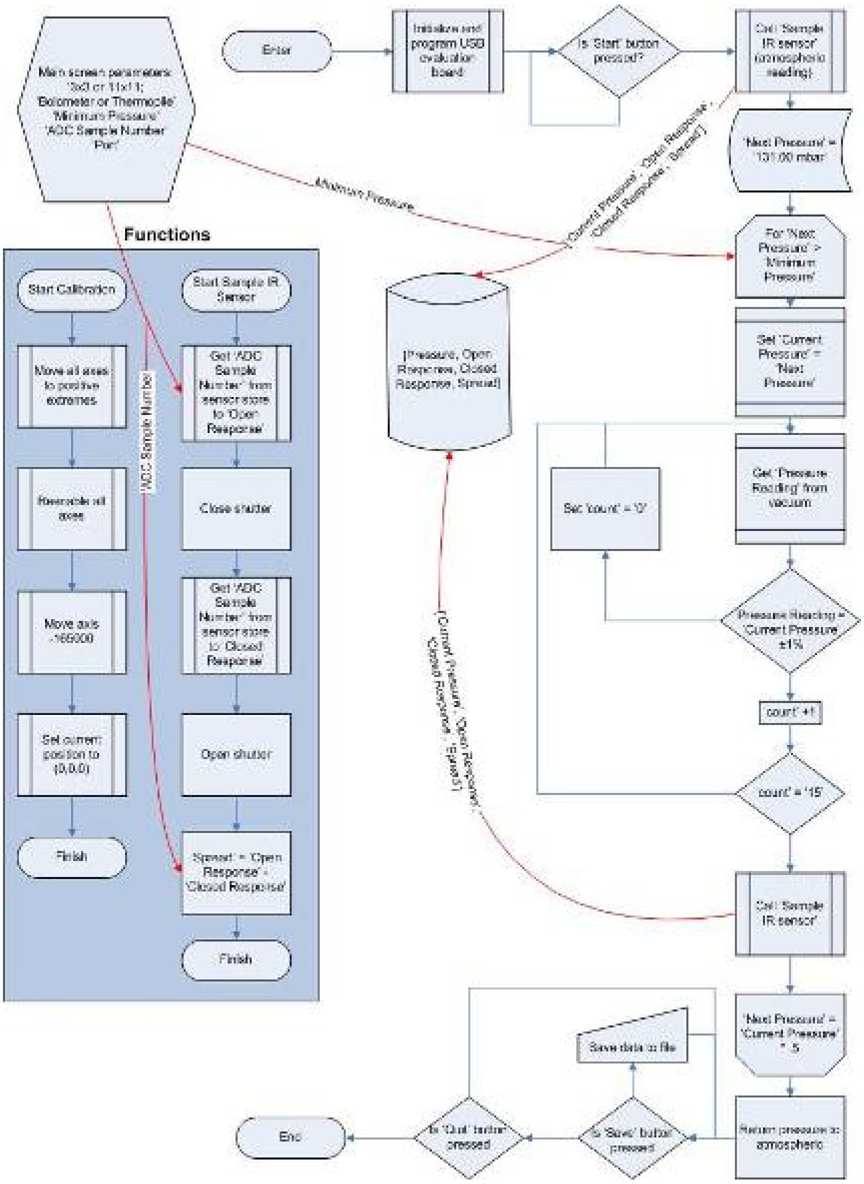

Main screen parameters:
'3x3 or 11x11';
'Bolometer or Thermopile'
'Minimum Pressure'
'ADC Sample Number
Port'
Enter
Initialize and program USB evaluation board
Is 'Start' button pressed?
Call 'Sample IR sensor' (atmospheric reading)
'Next Pressure' = '101.00 mbar'
Minimum Pressure
['Current Pressure', 'Open Response', 'Closed Response', 'Spread']
For 'Next Pressure' > 'Minimum Pressure'
Set 'Current Pressure' = 'Next Pressure'
Get 'Pressure Reading' from vacuum
Set 'count' = '0'
Pressure Reading = 'Current Pressure' ±1%
'count' +1
'count' = '15'
[Pressure, Open Response, Closed Response, Spread]
['Current Pressure', 'Open Response', 'Closed Response', 'Spread']
Call 'Sample IR sensor'
'Next Pressure' = 'Current Pressure' * .5
Return pressure to atmospheric
Save data to file
Is 'Save' button pressed
Is 'Quit' button pressed
End
Functions
Start Calibration
Move all axes to positive extremes
Reenable all axes
Move axis -165000
Set current position to (0,0,0)
Finish
'ADC Sample Number'
Start Sample IR Sensor
Get 'ADC Sample Number' from sensor store to 'Open Response'
Close shutter
Get 'ADC Sample Number' from sensor store to 'Closed Response'
Open shutter
'Spread' = 'Open Response' - 'Closed Response'
Finish

Figure 67: Flowchart of pressure experimentation software.

The front panel prompts the user to set various parameters to configure the pressure experiment. These parameters are listed in the top left corner of the flowchart: 3x3 or 11x11 array, thermopile- or bolometer-based sensor, “Minimum Pressure,” and “ADC Sample Number,” and “Port.” The “Minimum Pressure” was the minimum pressure that would be achieved inside the vacuum controller. “ADC Sample Number” was the number of samples the software would take from the sensor before opening or closing the shutter. An increase in “ADC Sample Number” decreased the amount of noise in the measurements

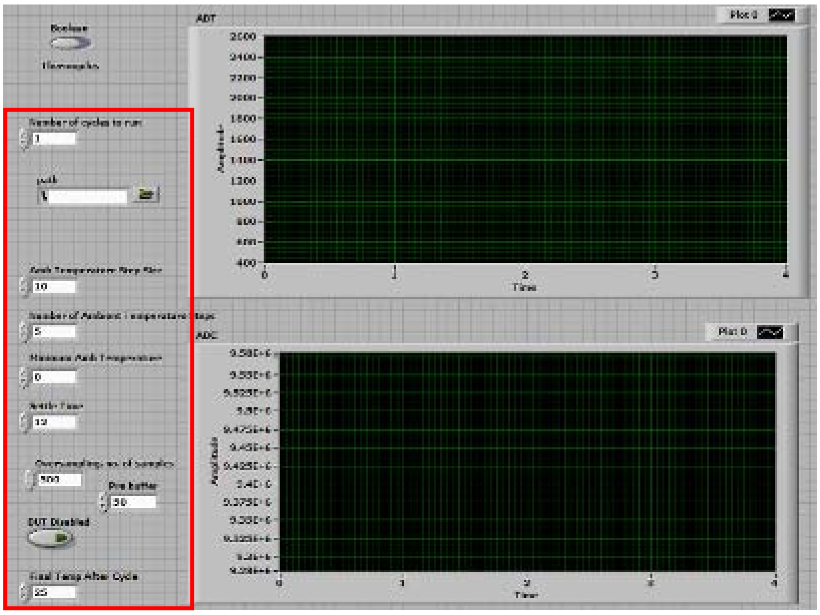
When the test begins, the software takes the initial reading before the pressure inside the vacuum chamber is set to a value other than atmospheric pressure. Taking a reading consists of collecting “ADC Sample Number” samples of the sensor’s response, closing the shutter, collection another “ADC Sample Number” samples of the sensor, and then reopening the shutter. To minimize the amount of noise in the data, the software averages the two groups of samples individually and stores the average of the samples taken when the shutter was open to “Open Response”, and the average of the samples taken when the shutter was closed to “Closed Response”. The software calculates the difference between “Open Response” and “Closed Response”, and stores this difference as the “Spread”. Next, all three of those values are stored in an array. This process is repeated for each pressure point: changing the pressure in the vacuum chamber, waiting for the pressure to reach the desired target pressure, taking a reading, and adding the newly obtained “Open Response,” “Closed Response”, and “Spread” values to the array.

The software reduces the pressure inside the vacuum controller by 50% after a successful “ADC Sample Number” of readings is taken. If the new target pressure is below “Minimum Pressure”, then the test is completed. If it is not, the software continuously polls the vacuum controller for the current pressure of the chamber, waiting for the pressure inside the vacuum controller to reach the target pressure. We found that even after we calibrated the vacuum controller, it still showed some irregularities when attempting to reach a lower pressure. These irregularities included overshooting of the pressure when descending to a lower value, instability of the actual reading, and inconsistent rates of depressurization. To ensure that the reading was taken at the desired value, we implemented a counter that would increment when a new consecutive reading was within ±1% of the desired target pressure. When the counter reached fifteen, the software took a reading from the IR sensor. This practice, combined calibrating the vacuum controller, greatly increased the repeatability of the test.

From this point, the user has the option to save the data to a CSV file and manipulate it manually or use the Pressure Experiment Data Processing tool we created.

6.1.2. Vacuum Degradation Experiment

As discussed in Section 4.2, we used the temperature forcer to precisely and accurately control the temperature of the air to which the IC was exposed. In addition to the temperature, we can also control the duration of time for the exposure by using the LabVIEW VI. The temperature forcer can be connected to a PC through a General Purpose Interface Bus (GPIB) to USB converter. GPIB allows up to 15 devices to share a single 8-bit parallel electrical bus by daisy chaining connections. The LabVIEW VI for the temperature forcer allows for various parameters to be entered. A screenshot of the VI is shown below in Figure 68.



Boolean

Thermopiles

Number of cycles to run: 1

path

Amb Temperature Step Size: 10

Number of Ambient Temperature Steps: 5

Minimum Amb Temperature: 0

Settle Time: 12

Oversampling: no. of samples: 500

Pre buffer: 50

DUT Disabled

Final Temp After Cycle: 25

Figure 68: Front panel of the temperature forcer VI.

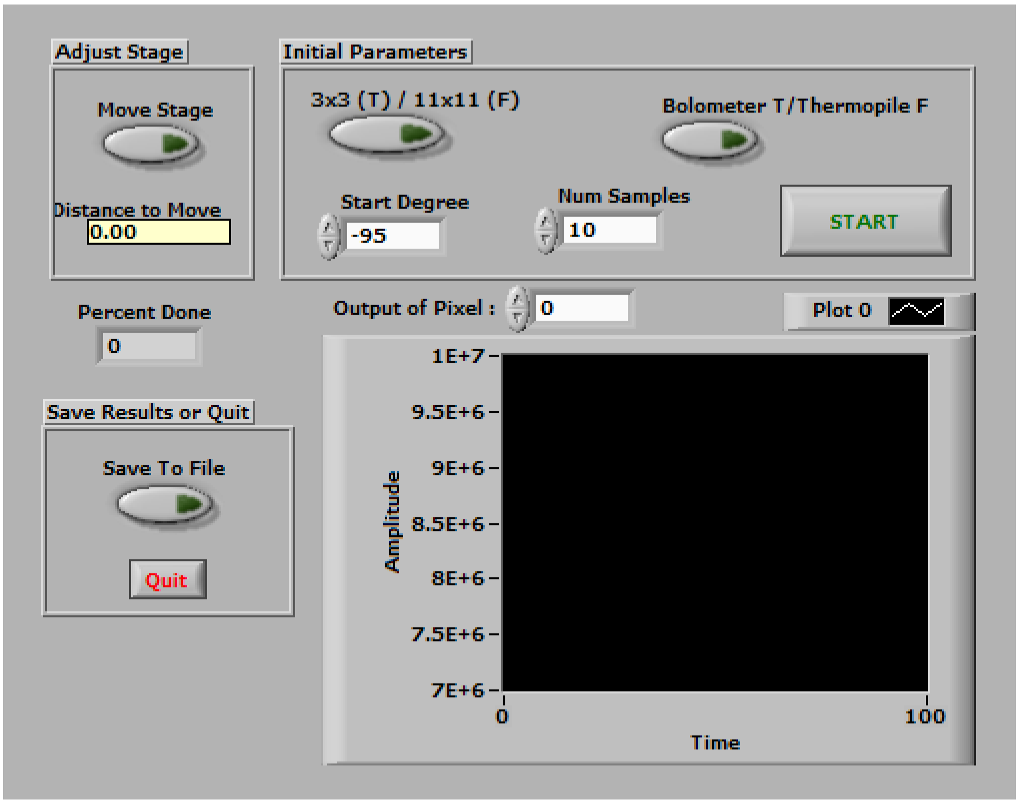
Since the original VI only allowed for testing of bolometers, a way to switch between bolometers and thermopiles was needed. To accomplish this, we added a switch to the front panel and stored the state of the button into a global variable. We then connected this global variable to a sub-VI that collects the data from the evaluation board. The original sub-VI needed to be swapped with an updated one which had an additional input. The additional input is of type Boolean (T/F), and corresponds to bolometers and thermopiles respectively. The parameters highlighted in Figure 68 alter the test in different ways. Table 6 describes the purpose of each parameter.

Table 6: Temperature forcer test parameters.

Devices	**The type of IR sensor to be tested (Bolometer or Thermopile)**
Number of cycles to run	The number of times that the test will run.
Path	Path for the output file.
Amb. Temperature Step size	Increase between each temperature set point.
Number of Ambient temperature steps	Number of set points per test.
Minimum Ambient temperature	Starting set point.
Settle time	Time at each set point.
Oversampling	Number of extra samples to take after the desired amount has been collected
Prebuffer	Number of samples to discard before collecting actual samples
DUT Disabled	Disables Device Under Test temperature reading. (On or Off)
Final temperature after cycle	Temperature for the forcer to return to upon completion.

6.1.3. Angular Response Experimentation Software

The angular response experiment described in Section 4.4 did not exist prior to this project. The 2006 group created a similar experiment using Lego's NXT devices, which are not nearly as precise as the Thorlab's CR1-Z6 rotational stage used this year. The VIs to communicate with the rotational stage were provided by Thorlabs. The software we created allows the user to enter in numerous parameters to prepare the device, configure the experiment, and save the results. The main screen of the VI that we created is shown in Figure 69.



Distance to Move: 0.00

Start Degree: -95

Num Samples: 10

Percent Done: 0

Output of Pixel : 0

Figure 69: Angular response test main screen.

Since the rotational stage can theoretically move forever in one direction, it is often necessary to adjust the stage to direct the IR sensor at the source before running the experiment. The "Move Stage" button allows the user to move the stage in either direction prior to the start of the test. The first button under "Initial Parameters," labelled "3x3 (T)/11x11 (F)," allows the user to specify whether the part being tested is a 3x3 array or an 11x11 array. To indicate that the part being tested is a 3x3 array, the user should press the button. If the part being tested is an 11x11 array, the user should not press the button. The user can press second button under "Initial Parameters", "Bolometer T/Thermopile F", to indicate that the bolometer is being tested. If a thermopile is being tested, the user does not press this button. "Start Degree" is the angle, relative to the sensor's initial position, from which the first sample will be collected. It is important to understand that this parameter is used in different areas of the test.

The second parameter is "Num Samples". This parameter refers to the number of samples that will be collected from the sensor at each degree. The median of these samples will be taken to produce a single reading per degree. These readings are displayed in real time on the main screen.

Once the experiment is configured, the user presses "Start" and the software automatically conducts the remainder of the experiment. Below, in Figure 70, is a flowchart of the final software.

The experiment begins with the sensor facing the blackbody source and the stage at 0^o degrees. Once the user presses the start button, the stage will move to the angle specified by "Start Degree". From this angle, the software will begin the process of sampling the IR sensor and incrementing the stage one degree. This process is repeated 'n' number of times where (n = |Start Degree| * 2).

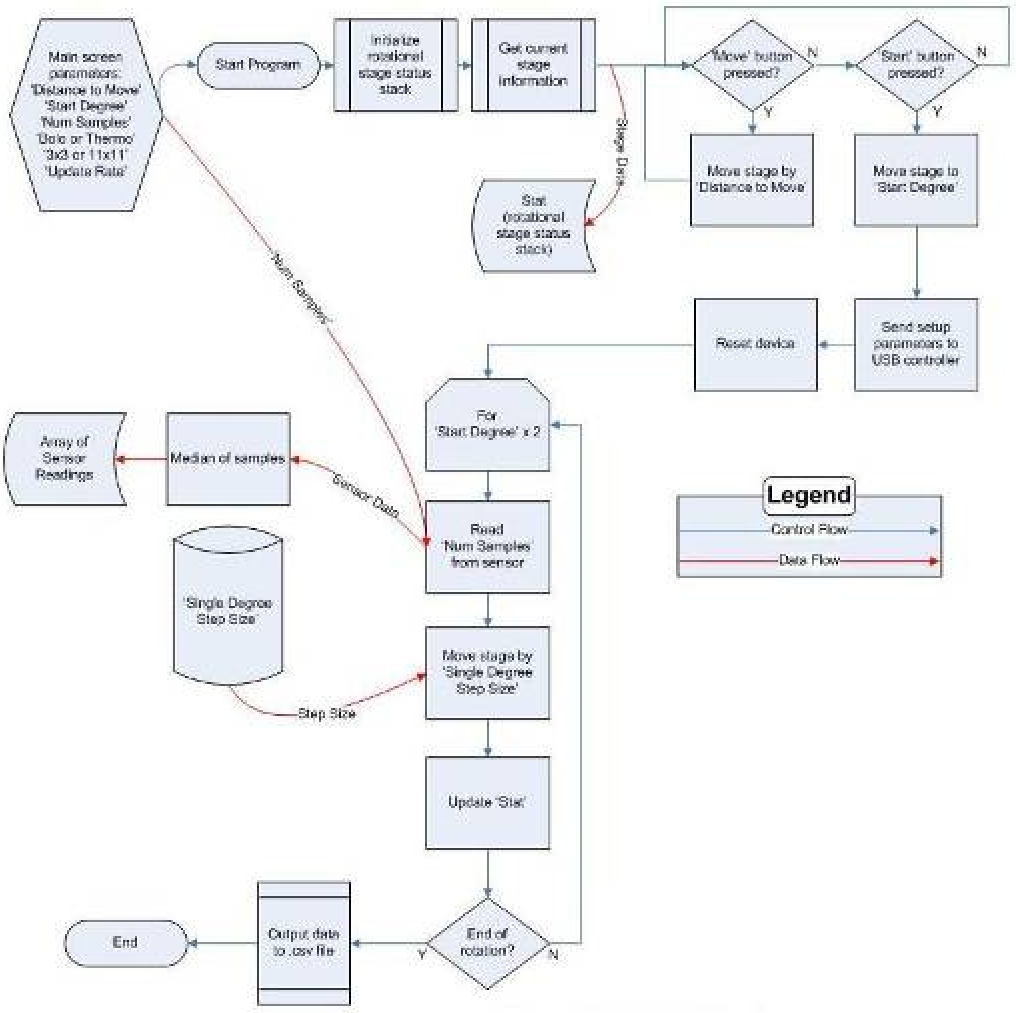

Main screen parameters: 'Distance to Move' 'Start Degree' 'Num Samples' 'Bolo or Thermo' '3x3 or 11x11' 'Update Rate'
Start Program
Initialize rotational stage status stack
Get current stage information
'Move' button pressed?
N
Y
'Start' button pressed?
N
Y
Stage Data
Stat (rotational stage status stack)
Move stage by 'Distance to Move'
Move stage to 'Start Degree'
Send setup parameters to USB controller
Reset device
'Num Samples'
For 'Start Degree' x 2
Array of Sensor Readings
Median of samples
Sensor Data
Read 'Num Samples' from sensor
'Single Degree Step Size'
Move stage by 'Single Degree Step Size'
Step Size
Update 'Stat'
End of rotation?
Y
N
Output data to .csv file
End
Legend
Control Flow
Data Flow

Figure 70: Flowchart of the angular response software.

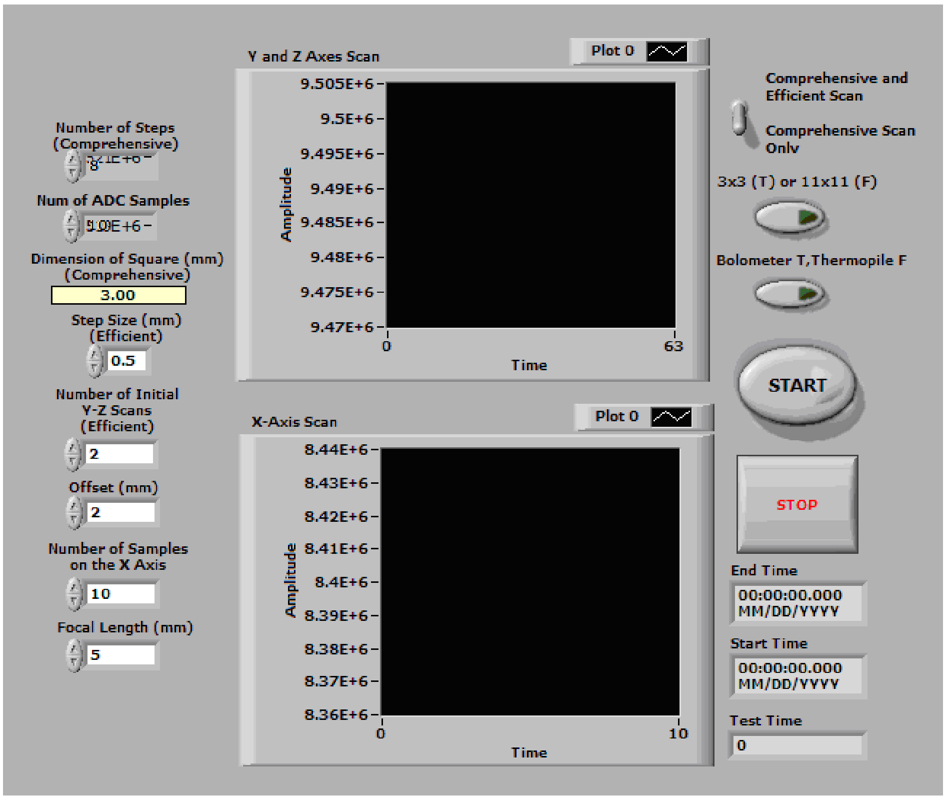
The initial version of our software required approximately 30 minutes to complete. The process of the angular response test is to run the software to collect initial results with the lens intact, polish away the lens, and then run the software again with the lens removed. This means that approximately an hour was spent collecting data.

To investigate methods that would reduce the time required for the experiment, we examined the IR sensor VIs and found that the “Send setup parameters” and “Reset” blocks shown in Figure 70 were inside the For loop. Extracting the resetting and configuration process out of the VI that sampled the sensor yielded a significantly faster test: we reduced the time required to test one part from 30 minutes to 5 minutes. Since we reduced the time needed to collect data for each part by 83%, the polishing process became the most time consuming part of the experiment.

The main screen also has a “Percent Done” text field to indicate the status of the experiment. Once the experiment is complete, the user can save the data collected to a Comma Separate Values (CSV) file, which can be easily imported into Microsoft Excel. The importing and processing of this data has been automated as well. The software is discussed in Section 6.2.3.

6.1.4. Lens Focusing Experimentation Software

The peak detection software scans a three dimensional space to find the location of highest IR response. ADI possesses the Thorlabs MT3-Z6 three-axis translational stage and the LabVIEW VIs to communicate with the device. Two algorithms for scanning the cube were developed. The first is a simple algorithm that scans each point of the cube in increments set by the user. Since this first method took more than three hours for a 10x10x10 cube, we developed an algorithm that would efficiently find the peak position in the cube in less time. We added an algorithm prior to this comprehensive scan to locate the part close to the peak position. The additional algorithm can be turned on or off with a switch. A picture of the front panel can be seen in Figure 71.


Number of Steps (Comprehensive): 8

Num of ADC Samples: 1.00E+6

Dimension of Square (mm) (Comprehensive): 3.00

Step Size (mm) (Efficient): 0.5

Number of Initial Y-Z Scans (Efficient): 2

Offset (mm): 2

Number of Samples on the X Axis: 10

Focal Length (mm): 5

End Time: 00:00:00.000 MM/DD/YYYY

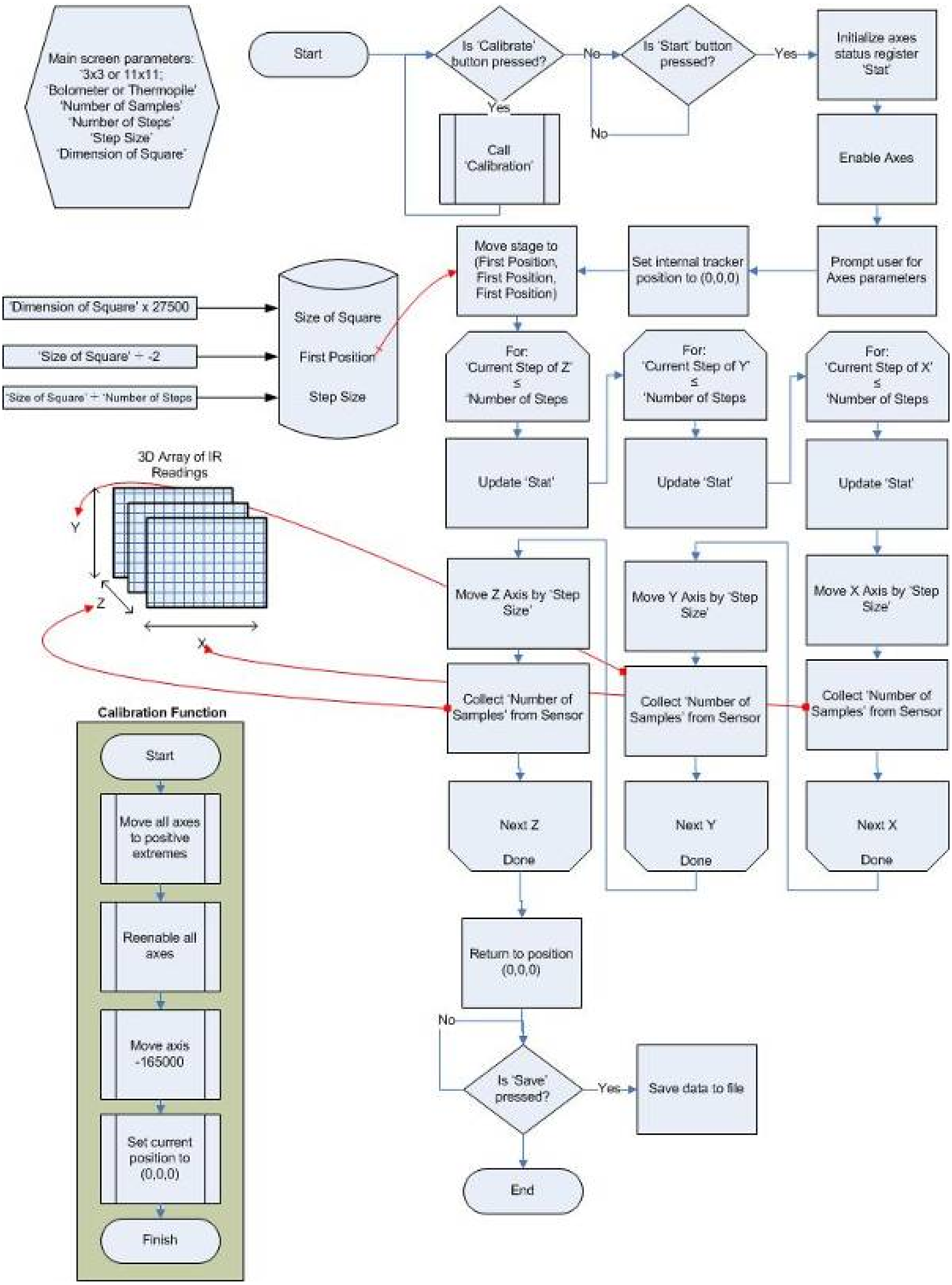
Start Time: 00:00:00.000 MM/DD/YYYY

Test Time: 0

Figure 71: Front panel of the lens focusing experimentation VI.

6.1.4.1. Comprehensive Scan Algorithm

The comprehensive scan serves two purposes: to provide an effective method of finding the position of peak response within a three dimensional square, and to yield a set of data that can be used to judge the effectiveness of the efficient algorithm. A flowchart of the software is shown below in Figure 72.



Main screen parameters: '3x3 or 11x11; 'Bolometer or Thermopile' 'Number of Samples' 'Number of Steps' 'Step Size' 'Dimension of Square'
Start
Is 'Calibrate' button pressed?
No
Yes
Call 'Calibration'
Is 'Start' button pressed?
No
Yes
Initialize axes status register 'Stat'
Enable Axes
Prompt user for Axes parameters
Set internal tracker position to (0,0,0)
Move stage to (First Position, First Position, First Position)
'Dimension of Square' x 27500
'Size of Square' ÷ -2
'Size of Square' ÷ 'Number of Steps'
Size of Square
First Position
Step Size
For: 'Current Step of Z' ≤ 'Number of Steps
For: 'Current Step of Y' ≤ 'Number of Steps
For: 'Current Step of X' ≤ 'Number of Steps
Update 'Stat'
Update 'Stat'
Update 'Stat'
3D Array of IR Readings
Y
Z
X
Move Z Axis by 'Step Size'
Move Y Axis by 'Step Size'
Move X Axis by 'Step Size'
Collect 'Number of Samples' from Sensor
Collect 'Number of Samples' from Sensor
Collect 'Number of Samples' from Sensor
Next Z
Done
Next Y
Done
Next X
Done
Return to position (0,0,0)
No
Is 'Save' pressed?
Yes
Save data to file
End
Calibration Function
Start
Move all axes to positive extremes
Reenable all axes
Move axis -165000
Set current position to (0,0,0)
Finish

Figure 72: Flowchart for the comprehensive scan algorithm.

The front panel allows the user to configure the test. The parameters such as "3x3 or 11x11" and "Bolometer or Thermopile" allow the user to configure the type of part under test. When sampling the IR sensor, the software takes "Number of Samples" samples and then averages the gathered samples to minimize noise. The user defines the size of the cube to scan. This parameter is measured in millimeters and is defined by the corresponding textbox, “Dimension of Square,” on the front panel. The resolution of the cube is defined by “Dimension of Square” divided by “Number of Steps”. For example, a 4mm cube with 10 steps would have 1000 points at a resolution of 0.4mm, which is the “Step Size”.

We designed simple yet important function to “calibrate” the platform to allow the user to position the three axes in the middle of their ranges. This function also calibrates the internal trackers of the axes to (0, 0, 0). All movements of the stage will be relative to this initial position during the test. If the user chooses not to calibrate, the axes will start at the zero value that is currently set in the axes controller.

When the user presses the “Start” button, the experiment begins. The three axes move to the negative extreme, relative to the origin, which is the bottom right-hand corner of the cube. From this position, the process of sampling the sensor then incrementing the sensor along the x axis continues for “Number of Steps” times. Once the “Number of Steps” is reached, the x-axis resets and the y-axis is incremented by “Step Size”. This process of sampling the sensor, moving the sensor’s position on the x-axis until “Number of Steps” is reached, resetting the x-axis, and incrementing the y-axis will continue until the limit of the y-axis is reached. Once the limit of the y-axis is reached, the z-axis will increment by “Step Size” and the process will restart beginning with the scan of the x-axis. The behavior described will conclude once the position (“Number of Steps,” ”Number of Steps,” ”Number of Steps”) has been reached. Upon completion, the software returns the stage to position (0,0,0) and allows the user to save the collected data in CSV format.

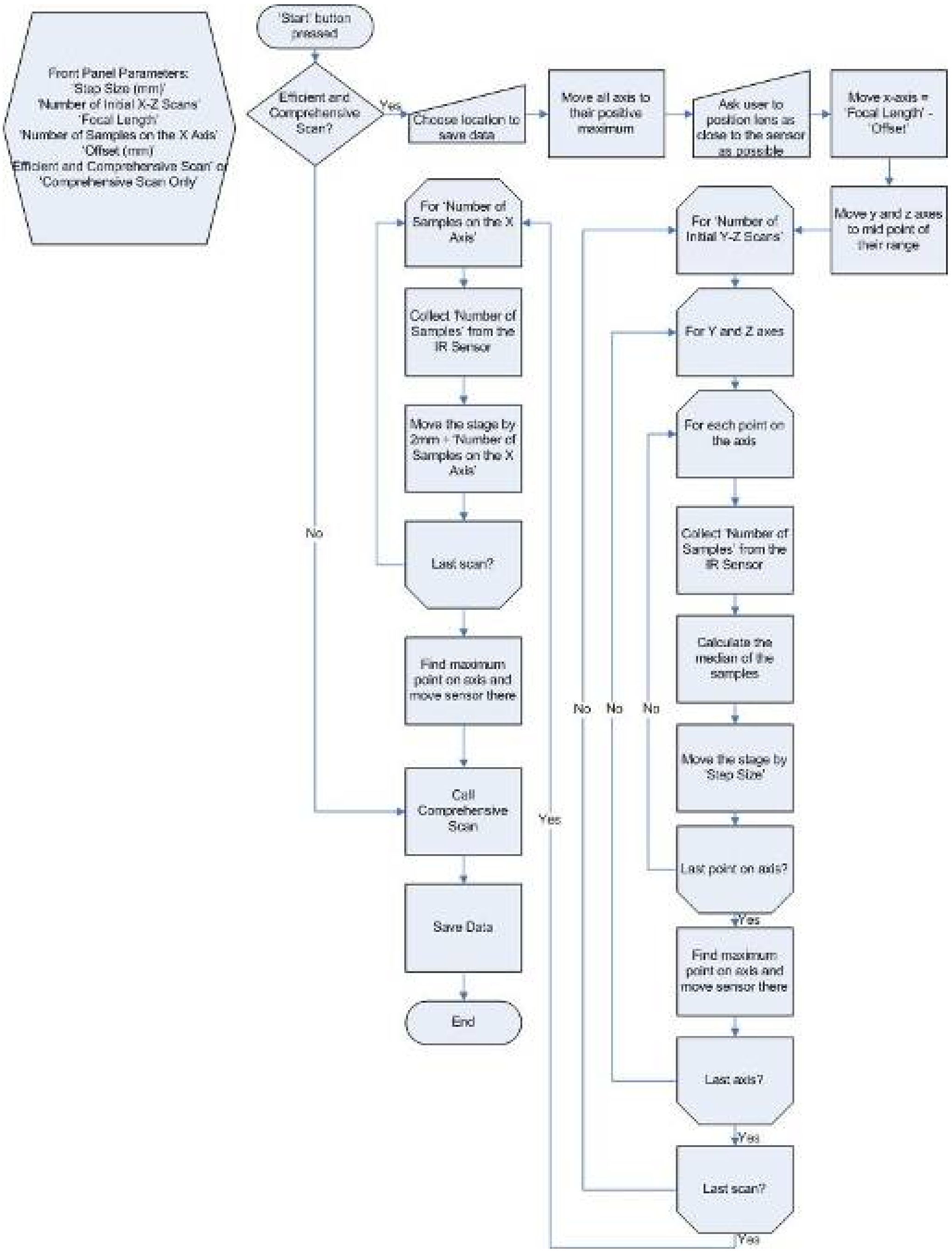
6.1.4.2. Efficient Scan Algorithm

The smart scan was designed to balance efficiency and accuracy. This is accomplished by combining the complete scan algorithm with an initial three-axis scan. Since the x-axis cross sections are fairly symmetric, an initial scan of the y- and z-axes should position the sensor in the middle of the cross section. To increase accuracy, this scan should be performed at least two times. The parameter “Number of Initial Scans” will determine how many times the y and z axes are scanned before scanning the x-axis, this parameter is shown in Figure 71.

With the approximately centered the x-axis can be scanned for the focal point. To adjust the resolution of the x axis scan, “Number of Samples on the X Axis” can be increased. Since the focal point is relatively small, this parameter should be set to no lower than ten. The “Step Size” should be 0.5mm. Due to limitations of the lens and sensor holders, the lens cannot come closer than 0.2mm to the sensor. The software uses the “Offset” parameter to take this into consideration. A picture of the limitation is shown in Figure 79.

The last parameter is “Focal Length (mm)”. This refers to the focal length of the lens. This parameter needs to be within ± 1mm since the x axis will be scanned for the peak.

Once the parameters are configured, the user can then press the “Start” button to being testing. A flow chart describing the software is shown below in Figure 73.



Front Panel Parameters:
'Step Size (mm)'
'Number of Initial X-Z Scans'
'Focal Length'
'Number of Samples on the X Axis'
'Offset (mm)'
'Efficient and Comprehensive Scan' or
'Comprehensive Scan Only'
'Start' button pressed
Efficient and Comprehensive Scan?
Yes
Choose location to save data
Move all axis to their positive maximum
Ask user to position lens as close to the sensor as possible
Move x-axis = 'Focal Length' - 'Offset'
Move y and z axes to mid point of their range
For 'Number of Initial Y-Z Scans'
For Y and Z axes
For each point on the axis
Collect 'Number of Samples' from the IR Sensor
Calculate the median of the samples
Move the stage by 'Step Size'
Last point on axis?
Yes
Find maximum point on axis and move sensor there
Last axis?
Yes
Last scan?
Yes
No
No
No
For 'Number of Samples on the X Axis'
Collect 'Number of Samples' from the IR Sensor
Move the stage by 2mm ÷ 'Number of Samples on the X Axis'
Last scan?
Find maximum point on axis and move sensor there
Yes
No
Call Comprehensive Scan
Save Data
End

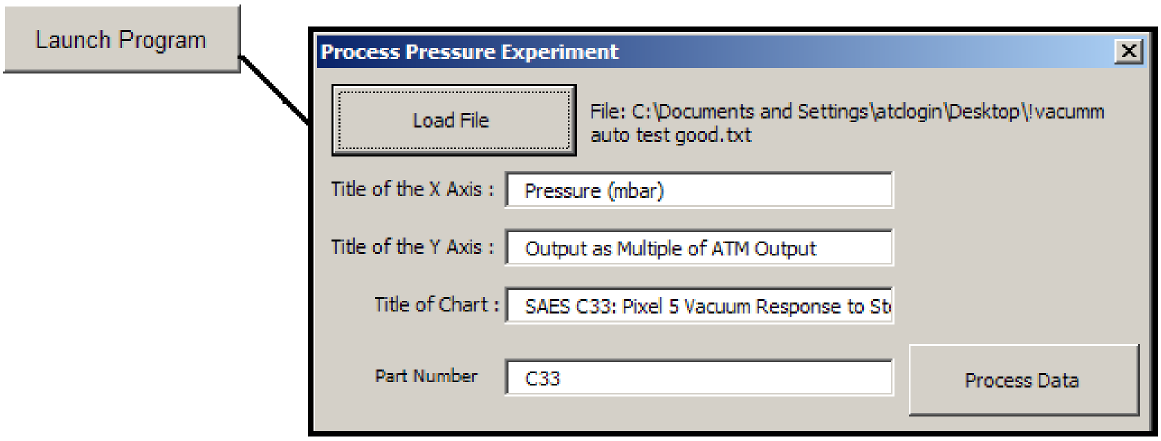
Figure 73: Flowchart of efficient scan.

6.2. *Data Processing Software*

With the reduced experimentation time, the users will be able to generate data more quickly. To compliment the suite of experimentation tools, we created a set of data processing applications that transform raw data into analyzable form. Using our acquired knowledge of IR sensors and the resources available at ADI, we were able to determine the best display methods. These tools are not intended to generate presentation-level reports, but rather to present the engineer with data that can be quickly interpreted and require minimal refinement to be acceptable to deliver to customers, management, or other engineers.

6.2.1. Data Processing of the Automated Pressure Experiment

The VI for the automated pressure experiment discussed in Section 6.1.1 allows the user to save the collected data to a CSV formatted file. The user can import the CSV formatted file into Microsoft Excel. Our software imports the CSV file and manipulates it to display the response of the ADiR sensor as the pressure changes. When the users utilize the data processing tool for the pressure experiment, they can view the results of the pressure experiment within seconds. Launching the program brings the user to the screen shown below in Figure 74.



File: C:\Documents and Settings\atclogin\Desktop\!vacumm auto test good.txt

Title of the X Axis : Pressure (mbar)

Title of the Y Axis : Output as Multiple of ATM Output

Title of Chart : SAES C33: Pixel 5 Vacuum Response to St

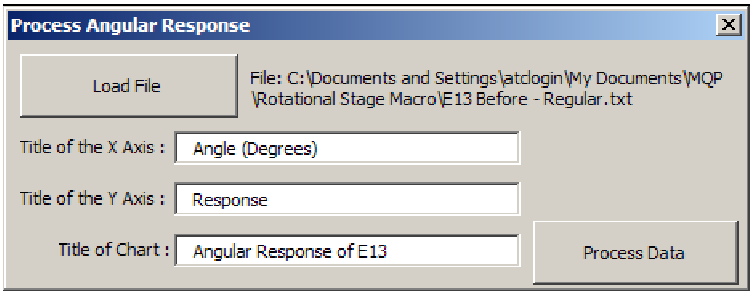
Part Number C33

Figure 74: Screenshot of the data processing software for the automated pressure experiment.

To use the software, the user first clicks "Load File'" to select the text data file, the next steps are to enter the titles for the two axes, the chart title, and the part number. After setting these parameters, the user clicks the "Process Data" button the bottom right corner of the screen. Clicking the button prompts the user for a location and filename to which the software will save the generated Excel file. Finally, the user clicks "OK" and the program imports the data, calculates the pressure quotient, generates the pressure curve, and then saves the file to the specified location. The code for this application is provided in Appendix E.

6.2.2. Data Processing of Angular Response Experiment

The VI for the angular response experiment discussed in Section 6.1.3 allows the user to save the collected data to a CSV formatted file, which can then be imported into Microsoft Excel and manipulated to show the angular response of each pixel. Like the data processing software for the pressure experiment, our software can generate the results within seconds. Launching the program brings the user to the screen shown below in Figure 75.



File: C:\Documents and Settings\atclogin\My Documents\MQP\Rotational Stage Macro\E13 Before - Regular.txt

Title of the X Axis : Angle (Degrees)

Title of the Y Axis : Response

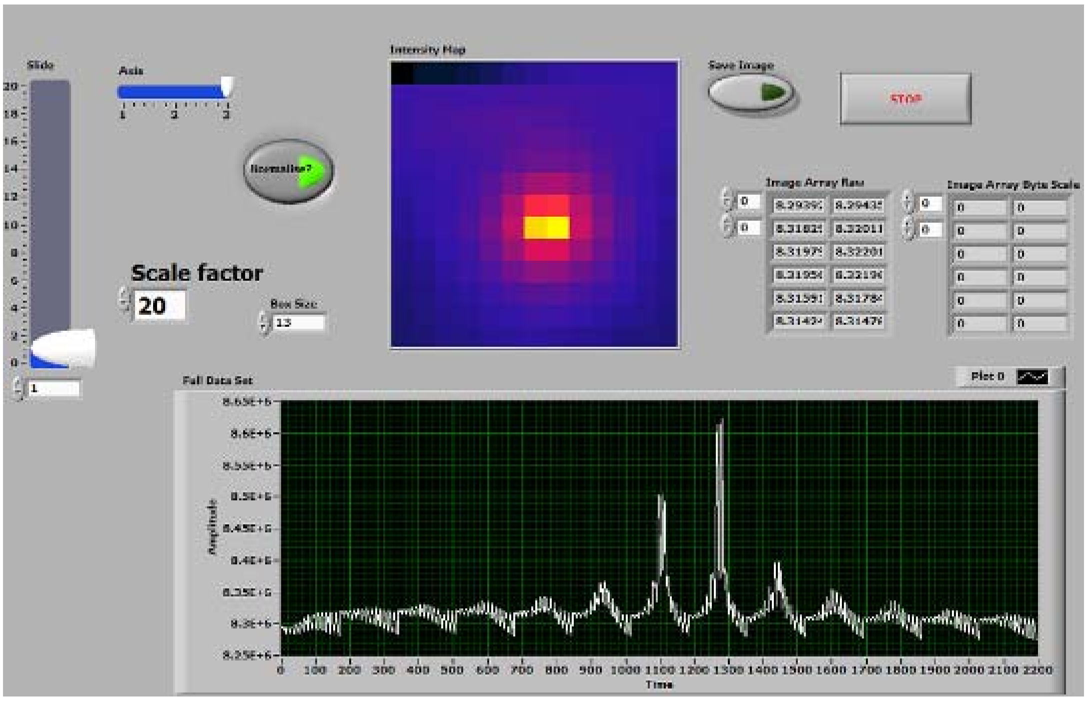
Title of Chart : Angular Response of E13

Figure 75: Data processing tool for the angular response experiment.

To use the software, the user first clicks “Load File” to select the CSV data file, enters the titles for the x-axis, y-axis, and chart, and clicks “Process Data.” Clicking the button prompts the user for a location and filename to which the software will save the generated Excel file. Finally, the user clicks “OK” and the program imports and normalizes the data, generates the angular response, and then saves the file to the specified location. The code for this application is provided in Appendix F.

6.2.3. Data Processing of Lens Focusing Experiment

The data processing of the lens focusing experiment is done by using a LabVIEW VI developed by Luke Pillans. A screen shot of the front panel is shown below in Figure 76.


Scale factor 20

Box Size 13

Figure 76: Front panel of the data processing VI for the lens focusing experiment.

The user is prompted to select the data file after clicking "Run" in LabVIEW. The "Box Size" parameter needs to be configured to the dimension of the square that was scanned. The user can then move the sliders to rotate the data being displayed. The vertical slider, labelled "Slide", allows the user to scroll through the cross sections. The software also allows the user to save the "Intensity Map" as an image by pressing the "Save Image" button.

6.2.4. Data Processing of Qualification Test

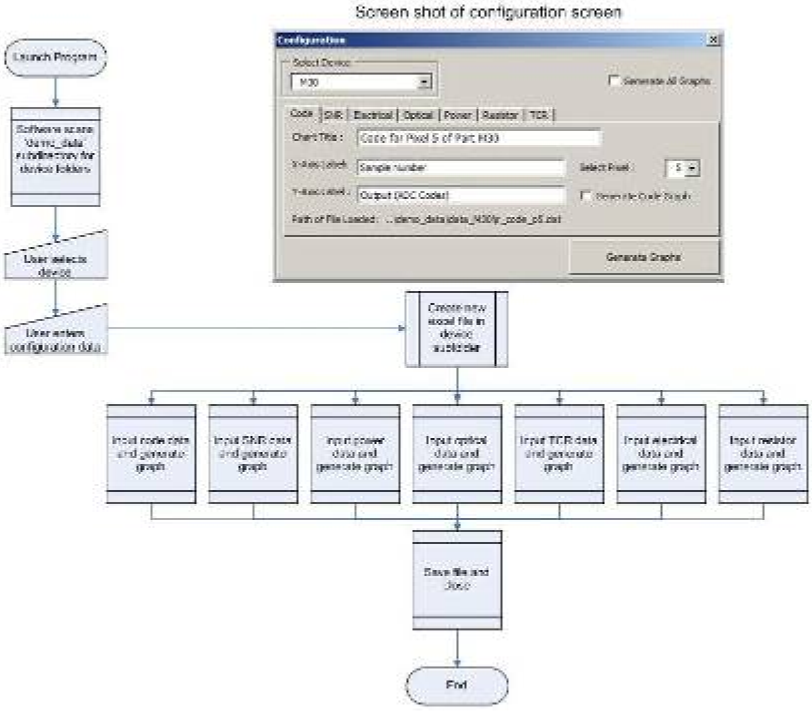
The qualification test produces seven sets of data. The output data is outlined below in Table 7.

Table 7: Qualification data.

ir_code_p*.dat
ir_elec_step.dat
ir_optical_step.dat
ir_snr_p*.dat
power.dat
resistor_data.dat
tcr.dat

* indicates a pixel number place holder, a separate file is generated for each pixel

The software scans the ".\demo_data" subfolder for files with the format: "data_*", where '*' is a placeholder for the part number. Once each part is found, it then scans those subfolders to verify that each file is present. With the part selected from the drop down menu, the user can configure the graphs accordingly. A flowchart of the software along with a screenshot of the configuration panel is shown below in Figure 77.



Screen shot of configuration screen
Configuration
Code
SNR
Electrical
Optical
Power
Resistor
TCR
Generate Graphs
Launch Program
Software scans 'demo_data' subdirectory for device folders
User selects device
User enters configuration data
Create new excel file in device subfolder
Input code data and generate graph
Input SNR data and generate graph
Input power data and generate graph
Input optical data and generate graph
Input TCR data and generate graph
Input electrical data and generate graph
Input resistor data and generate graph
Save file and close
End

Select Device: M30

☐ Generate All Graphs

Chart Title : Code for Pixel 5 of Part M30

X-Axis Label : Sample number

Select Pixel : 5

Y-Axis Label : Output (ADC Codes)

☐ Generate Code Graph

Path of File Loaded : ...\demo_data\data_M30\r_code_p5.dat

Figure 77: Flowchart of qualification data processing.

The code for this application is provided in Appendix G.

7. Recommendations

After testing and characterizing the ADiR sensor using the test equipment provided, we feel that there is still room for improvement in the various testing procedures designed by ADI. Our recommendations could help ADI make more accurate conclusions by eliminating hardware and equipment issues.

We recommend that ADI use a protective cap to prevent the exposed bond wires from breaking during tests. At the moment, the fragile bond wires are physically exposed, as shown inside the red ovals in Figure 78.

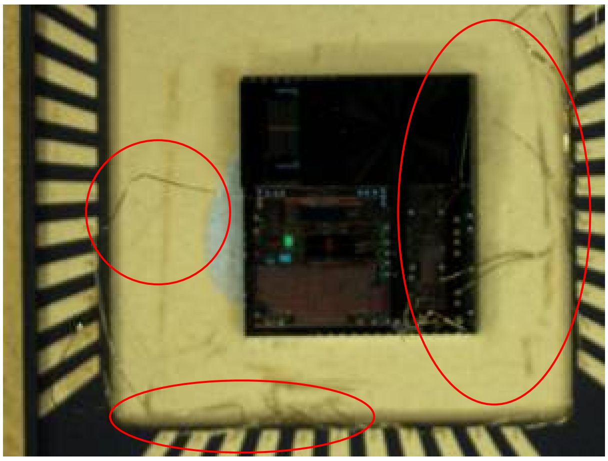


Figure 78: Broken bond wires.

As a result, it is very easy to damage them physically and thus destroy the part. Since all of our parts had exposed bond wires, several of them broke, so for some of the experiments we conducted, such as the vacuum degradation experiment, we took a measurement on one part and when it broke, we replaced it with another part and took a second measurement on the new one. Protecting the bond wires would help ADI in obtaining data results from one single part instead of using different parts and finding ways to compare their individual characteristics.

We recommend that ADI improve the test setups by providing more reliable equipment that provides better shielding from IR sources that could interfere with the experiment. Currently, we use a makeshift unit that holds the evaluation board and sensor in place while we conduct the lens focusing experiment. As seen in the photograph of this unit, provided in Figure 79, although it is difficult for the evaluation board to move during the lens focusing experiment, if an engineering removes the board and replaces it, it is likely that the board would have shifted or rotated. This causes inconsistencies in the results. Since the holders of the IR sensor and focusing lens are rather large, it is difficult to scan the full range of the focusing lens. This can be seen in Figure 79 and can be solved by obtaining more appropriate holders.

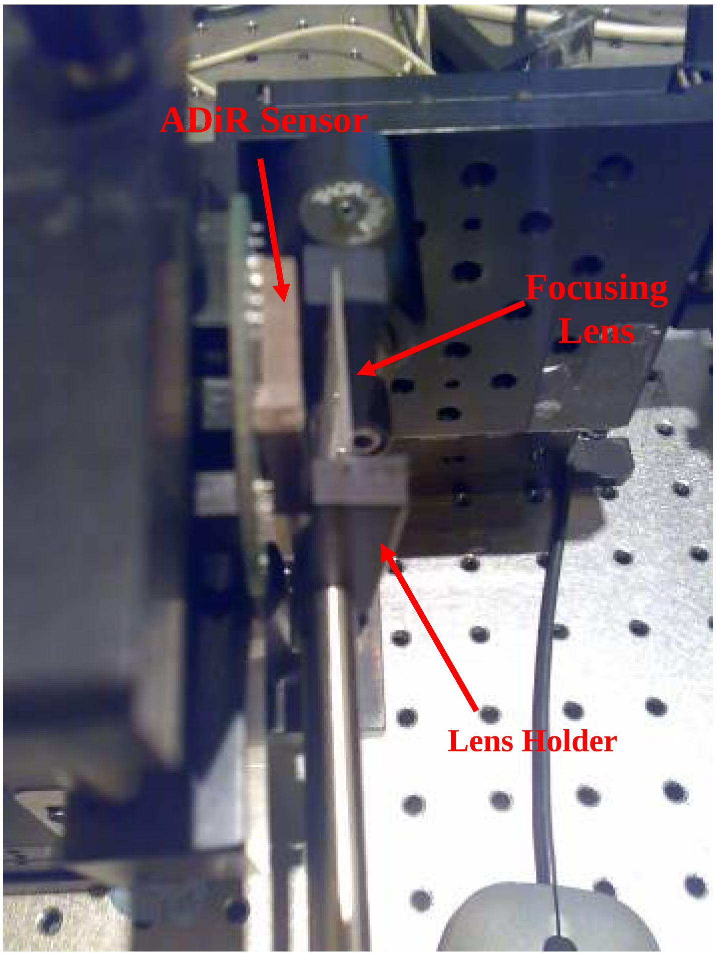


Figure 79: Photograph of the gap when the lens holder is touching the sensor.

Another potential problem with the setups for the pressure, angular response, and lens focusing experiments is that the monitor is also an IR source, so the radiation that it emits could affect the results. To minimize or eliminate this effect, ADI should shield the monitor or any other hot object in the room.

We recommend that ADI investigate the efficient algorithm that we developed. The results we obtained from it appear promising. If we were allowed more time, we would conduct more tests using the algorithm to verify that it yields the same results as the comprehensive scan, but does not require as much time.

We also recommend that ADI conduct the vacuum degradation experiment on more 11x11 and 3x3 parts to verify our results. Since we were left with only five of the twelve original parts from the vacuum degradation experiment on the 11x11 bolometers and thermopiles, we do not believe we have enough data to draw accurate conclusions; our data only suggests that bolometers are more sensitive to changes in vacuum than thermopiles are.

ADI should conduct the vacuum degradation experiment on additional 3x3 bolometers, too, since we tested the bolometers over a period of two days. Continuing the experiment for a longer period of time would provide more data, which would lead to more accurate conclusions.

8. Conclusions

ADI has developed an IR sensor that can be used in non-contact thermometry and motion detection applications. To characterize the performance of these IR sensors, ADI has designed a series of testing procedures. We have successfully developed a suite of automation software that has made these testing procedures faster, more repeatable, and more reliable. Hence, the results are more consistent and there is no longer a need for employees to monitor the tests. This section summarizes the results of our contributions.

ADI can use `the data we gathered from the pressure experiments to enhance their theoretical model regarding the behavior of the ADiR sensor when subjected to varying pressures is accurate. We conducted the pressure experiment on twenty parts to first understand the behavior of the ADiR sensor and then to confirm the accuracy of ADI's theoretical model that at low pressures, the 3x3 bolometers are more sensitive than 11x11 bolometers. We found that the pressure curves we generated experimentally were similar to the pressure curves that were predicted by the theoretical model because they both exhibited the same shape, which showed that the bolometer-based sensors are more sensitive at low pressures than at high pressures. However, there was a small discrepancy between our experimentally-generated pressure curves and the theoretical model: our experimentally-generated pressure curves were skewed to the right. Two possible explanations for this discrepancy are experimental error and an inconsistency in the manufacturing process of the sensor. This experimental error could have been caused by our inability to maintain the same pressure inside the vacuum chamber as the pressure inside the ADiR sensor. To remove this discrepancy, we automated the software that controls the vacuum controller and provides accurate pressure measurements. The manufacturing error could be due to the difference in material constants used in the manufacturing process, which ADI will investigate.

We successfully automated the pressure experiment by configuring the vacuum controller to set pressure inside a vacuum chamber and automating a shutter that provides a step input at varying pressures. This automation reduced the human-machine interaction time from sixty-five minutes to five minutes per part. When engineers use our software, the only time they spend on the experiment is the five-minute setup; they no longer need to manually conduct the sixty-minute experiment.

We successfully improved the software for the vacuum degradation experiment by enabling the user to test both bolometers and thermopiles in the temperature forcer.

Before our change to the aging software, ADI could test aging only on bolometers. We solved this problem by creating a button that allowed users to specify whether they were testing a bolometer or a thermopile.

The results from the first vacuum degradation experiment suggested that bolometers age more rapidly than thermopiles. We also concluded that storage in an oven expedites the aging of both bolometers and thermopiles. However, while we were conducting these experiments, five of the twelve parts on which we experimented broke because of the force of the air from the temperature forcer. We decided to rectify this issue by modifying the setup of the next vacuum degradation experiment.

The results from the second vacuum degradation experiment implied that covering the sensor with a cap successfully prevented the temperature forcer from breaking the parts. Since we were coving the sensor with a cap, we assumed that the sensor would require more time to adjust to the temperature, so we compensated by increasing the "settling time" parameter from four minutes to ten minutes. Consequently, the length of the experiment increased, but none of the parts broke during the experiment.

The results from the second vacuum degradation experiment also suggested that a new manufacturing process used on 3x3 bolometers improved their responses. When compared to the 3x3 bolometers that were manufactured by the older process, the 3x3 bolometers manufactured with the new process exhibited increased sensitivities and reduced baseline offsets**.**

We developed a software suite in Microsoft Excel using VBA to process the data collected from qualification tests. The purpose of the qualification tests was to separate and characterize the properly working parts from those that did not function. Since ADI did not possess a quick method to perform data processing on each part that underwent the qualification experiment, we decided to create a software package that would enable them to import the data from a CSV file into an Excel file and create the graphs necessary for analysis. The result was that instead of spending hours to manually create these graphs one by one, the users could take advantage of our VBA program and obtain results within minutes.

We successfully automated the angular response experiment. The angular response experiment required that we move the rotational stage 180° and obtain ten readings from the sensor at each 1° increment. Since it is tedious to move the rotational stage one degree at a time with one VI and obtain ten data points with a separate one, we decided to combine the two functionalities and create a single VI that would both move the rotational stage through 180° and acquire readings from the sensor. After we finished developing the first version of the software that combined the two functionalities, we ran it and found that it required 30 minutes to complete. We made some additional modifications to the software because we realized that there were unnecessary software modules that reduced the speed of the experiment. The second version of our software required only five minutes to complete this experiment. We confirmed that our second version operated properly by verifying that it yielded the same results as the first version.

The results from the angular response experiment suggested that the diffractive lens pattern would enable the sensor to be used in motion-detection applications. When we conducted the angular response experiment on the ADiR sensor and used lens that contained a diffractive pattern, we found the sensor exhibited angular response. After we removed this lens pattern, we found that each of the pixels responded uniformly, which meant that they did not exhibit an angular response. Since angular response enables the sensor to detect motion, we believe that the diffractive lens pattern would enable the sensor to be used in motion-detection applications.

We automated the lens focusing experiment to find the location in three-dimensional space where the center pixel obtains a maximum signal, relative to a radiation source. We accomplished this by creating a VI that moves the translational stage holding the sensor in all three dimensions. First, we developed an algorithm that moves the stage to points that span a large volume in three-dimensional space and samples the sensor's responses at each of those positions. Not only did this take three hours, but it also required that the user analyze the data to find the maximum signal, and move the stage to the location of this maximum signal. After we finished this slow scan, we developed another algorithm that determined the location of the maximum signal by scanning a smaller volume in the three-dimensional space. The results from this algorithm appear promising.

Using our automated software to conduct experiments on the sensor and collect data, ADI will be able to obtain data about the sensor's performance significantly more quickly than they had been able to previously. The increase in the speed at which ADI obtains results will enable them to further develop the sensor more quickly, and therefore, they will be able to introduce it to the market sooner.

9. Bibliography

"AD7794 Datasheet." Analog Devices, Inc., 2007.

"ADT7301 Datasheet." Analog Devices, Inc., 2007.

Barron, W.R. "Principles of Infrared Thermometry." Retrieved August 15, 2007 from http://www.omega.com/temperature/Z/pdf/z059-062.pdf.

Braunovic, Milenko. *Electrical Contacts: Fundamentals, Applications and Technology*. Chapter 11: Monitoring Technologies. Retrieved August 14, 2007 from ENGnetBASE, www.engnetbase.com.

Cawley, Brendan (brendan.cawley@analog.com), personal communication, August 17, 2007.

Extech 42500 Mini -4 Degree to 500 Degree Fahrenheit and -20 Degree to 260 Degree Celsius Infrared Thermometer. Retrieved September 4, 2007 from amazon.com: http://www.amazon.com/Extech-42500-Farenheit-Infrared-Thermometer/dp/B0000WU19S/ref=pd_bbs_7/104-0329731-6879945?ie=UTF8&s=hi&qid=1188906379&sr=8-7

Gruner, Klaus-Dieter. "Principles of Non-Contact Temperature Measurement." *Raytek®*. Retrieved August 14, 2007 from http://www.raytek-europe.com/admin/file_handler/9c2cae67f7a4e392f2d05da4fe4daa26/1014336830/IR_Theory_RevB_LR.pdf.

"Instruction Manual for Model M315 Blackbody Calibration Source." Retrieved 20 August, 2007 from http://www.mikroninfrared.com/manuals/M315man.PDF

Kirkpatrick, Allan. "Heat Transfer Mechanisms." Retrieved September 6, 2007 from http://www.engr.colostate.edu/~allan/heat_trans/page4/page4f.html.

Kreith, F., et. al. (2000). *The CRC Handbook of Thermal Engineering*. Retrieved August 14, 2007 from ENGnetBASE, www.engnetbase.com.

MacMath, Lauren K., et. al. (2006). "Analysis and Implementation of the Integrated 3x3 and 11x11 Pixel Array Infrared Detectors for the ADiR Product Line." http://www.wpi.edu/Pubs/E-project/Available/E-project-101406-021118/unrestricted/ADI_2006_MQP_Final_Report.pdf

"Mini IR Thermometer." http://www.extech.com/instrument/products/400_450/42500.html

Neuzil, Pavil, and Mei, Ting. (2001). http://ieeexplore.ieee.org/iel5/7361/28464/01271268.pdf

Ng, DK. (2005). "Non-contact infrared thermometry temperature measurement for screening fever in children." *Annals of Tropical Paediatrics*. 25(4):267-75. Retrieved September 20,

2007 from
http://www.ncbi.nlm.nih.gov/sites/entrez?cmd=Retrieve&db=PubMed&list_uids=16297301&dopt=AbstractPlus.

Pillans, Luke (luke.pillans@analog.com). “ADIR Bolometer Technology.” Presented at General Technical Conference, May 1-3, 2007.

Stephenson, Robert et. al. (2002). *Measurement, Instrumentation, and Sensors Handbook*. Temperature Measurement. Retrieved August 14, 2007 from ENGnetBASE, www.engnetbase.com.

Weckmann, Stephanie. (1997). “Dynamic Electrothermal Model of a Sputtered Thermopile Thermal Radiation Detector for Earth Radiation Budget Applications.” Retrieved August 17, 2007 from http://scholar.lib.vt.edu/theses/available/etd-8497-205315/unrestricted/chap2.pdf.

Appendix A: Laws of Physics Applicable to the Project

Electromagnetic radiation is emitted by the constantly moving charges that are present in any object that has a temperature above absolute zero. IR radiation, like any radiation, can be absorbed, transmitted, reflected, or scattered (Kreith, p. 66). There are four physical laws that govern radiation and are applicable to this project: Planck's Law, Wien's Law, the Stefan-Boltzmann Law, and Kirchhoff's Law.

Planck's Law is one of the laws of physics on which the ADiR sensor is based. Planck's Law states the spectral radiance (W_λ), or amount of radiation emitted, as a function of wavelength (λ), temperature (T), and emissivity (ε), where emissivity is the ratio of the amount of energy radiated by a material to the amount of energy radiated by a blackbody at the same temperature. There are several ways to express Planck's Law, one of which is provided in Equation 4:

$$W_\lambda = \frac{\varepsilon(\lambda)C_1}{\pi\lambda^5\left(e^{C_2/(\lambda T)}-1\right)}$$

Equation 4: Planck's Law (Stephenson, p. 89).

where $\varepsilon(\lambda)$ is the emissivity, $C_1 = 3.74 * 10^{-12}$ W/cm^2, and $C_2 = 1.44$ cm*K (Stephenson, p. 89).

A blackbody is an object that has an emissivity of 1; it absorbs all of the radiation that strikes it and does not reflect any radiation. Real objects have an emissivity that is less than 1. Non-metallic objects often have higher emissivities than metallic objects (Stephenson, p. 89). Using Planck's Law and assuming a blackbody source, spectral radiation curves can be generated, as shown in Figure 80.

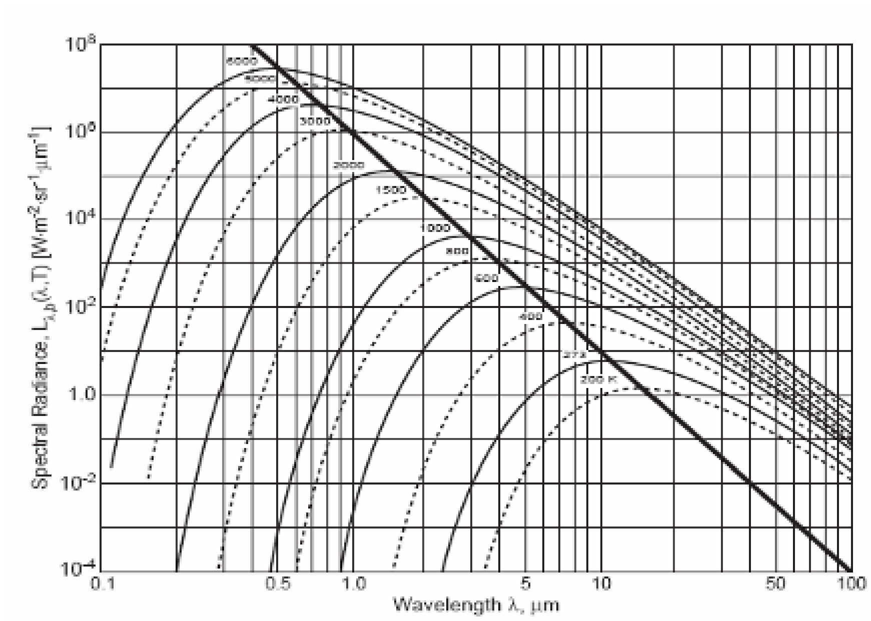


Figure 80: Spectral radiation graph (Stephenson, p. 89).

This graph is applicable to IR thermometry because IR temperature sensors, such as the ADiR sensor, operate at a particular wavelength. Based on the measured radiation intensity and given wavelength, IR sensors can detect the temperature of an object.

Wien's Law is a simpler version of Planck's Law; since Planck's Law is mathematically complicated, Wien's Law (Equation 5) is often used as an approximation.

$$W_\lambda = \frac{C_1}{\pi}\varepsilon(\lambda)\lambda^{-5\frac{C_2}{e^{\lambda T}}}$$

Equation 5: Wien's Law (Stephenson, p. 90).

The most probable wavelength at a certain temperature can be derived by setting the first derivative of Equation 5 to zero. The result is shown in Equation 6.

$$\lambda_m = \frac{2898}{T}$$

Equation 6: Wien's Displacement Law.

where λ_m is the wavelength in microns and T is the temperature in Kelvin (Stephenson, p. 90).

Equation 6 shows that wavelength and temperature are inversely proportional to one another; the shorter the wavelength emitted, the higher the temperature (Sandberg p. 90). For this project, the most important property of Wien's Displacement Law is that it describes the location of the radiation peaks on the spectral radiation graph as shown in Figure 80, by the thick black line. By knowing the location of these radiation peaks, ADI could decide the wavelength on which ADiR sensor would operate. Although it appears that the best wavelength would be where the peaks of the curves are located, ADI decided to use a wavelength in the long-wave IR (LWIR) region because the effect of sunlight on the measurement is smaller in this region than in the medium-wave IR region (MWIR) (Pillans).

The Stefan-Boltzmann Law (Equation 7) shows that the power of the radiation emitted by a target (Φ_{b0}) is directly proportional to the area (A), the emissivity, and the temperature of the target, assuming the body receiving the radiation is at absolute zero:

$$\Phi_{b0} = A\varepsilon\sigma T^{4}$$

Equation 7: Stefan-Boltzmann Law.

where $\sigma = 5.67*10^{-8}$ W/m^{2}K^4 (Stephenson, p. 91).When the temperature of the target increases, the power of the emitted radiation measured grows very quickly (Sandberg, p. 90). This is applicable to the project because the relationship between radiation and temperature is not perfectly linear; calibration must be performed on the sensor.

Kirchhoff's Law states that for any object, the sum of its reflectivity (ρ), transparency (γ), and emissivity is equal to 1 (Sandberg, p. 90).

$$\varepsilon + \gamma + \rho = 1$$

Equation 8: Kirchhoff's Law.

If an object is opaque, its transparency will be zero. Equation 9 is a simplified version of Equation 8, but it works only for opaque objects.

$$\varepsilon = 1 - \rho$$

Equation 9: Kirchhoff's Law for opaque objects.

Some materials, such as glass and silicon, are exceptions to this equation, but if selective spectral filtering is used, the temperature of these objects can be measured in the region at which they are opaque to IR (Barron).

Kirchhoff's Law is applicable to the project because all of the testing the group is conducting is performed on a blackbody, which has an emissivity of nearly 1, and therefore a reflectivity and transparency of approximately zero. The results of the tests would change if the IR sensor were tested on other objects.

Appendix B: Pfeiffer Vacuum Controller RVC300

ADI uses the Pfeiffer Vacuum Controller RVC300 for pressure control and gas flow adjustment, as shown in Figure 11.

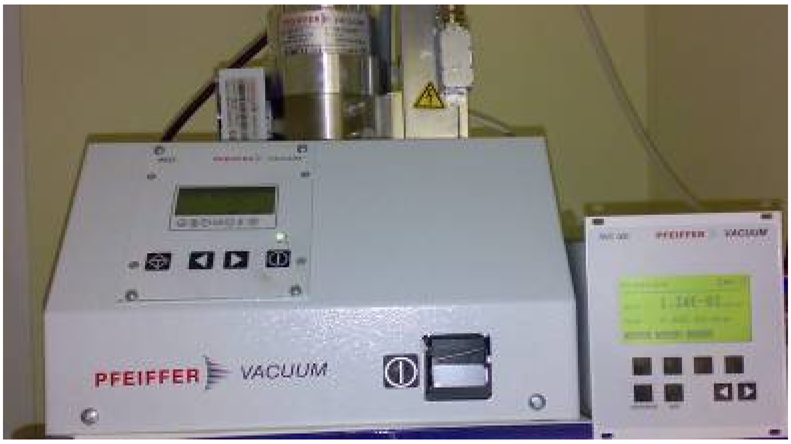


Figure 81: Vacuum controller.

The ADiR sensor was subjected to a variation in pressure in order to create a near-vacuum using a pressure sensor and a control valve. To increase the pressure within the vacuum system, the valve plates close as shown in Figure 82.

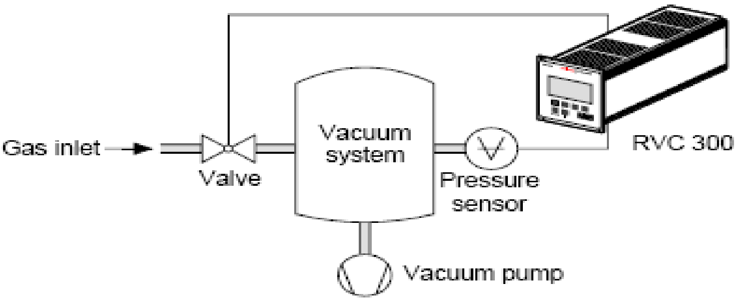


Figure 82: Block diagram of increasing pressure flow (Pfeiffer Vacuum Manual).

In order to release the pressure, the plates of the valve move into the open position as shown in Figure 83.

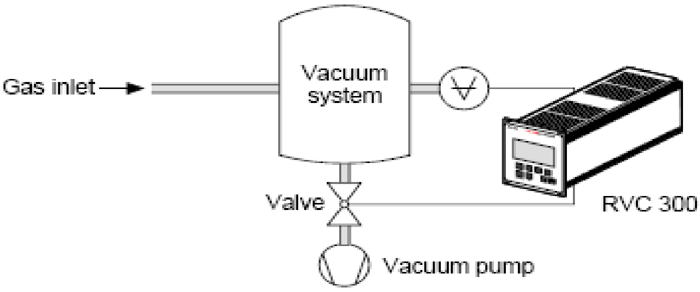
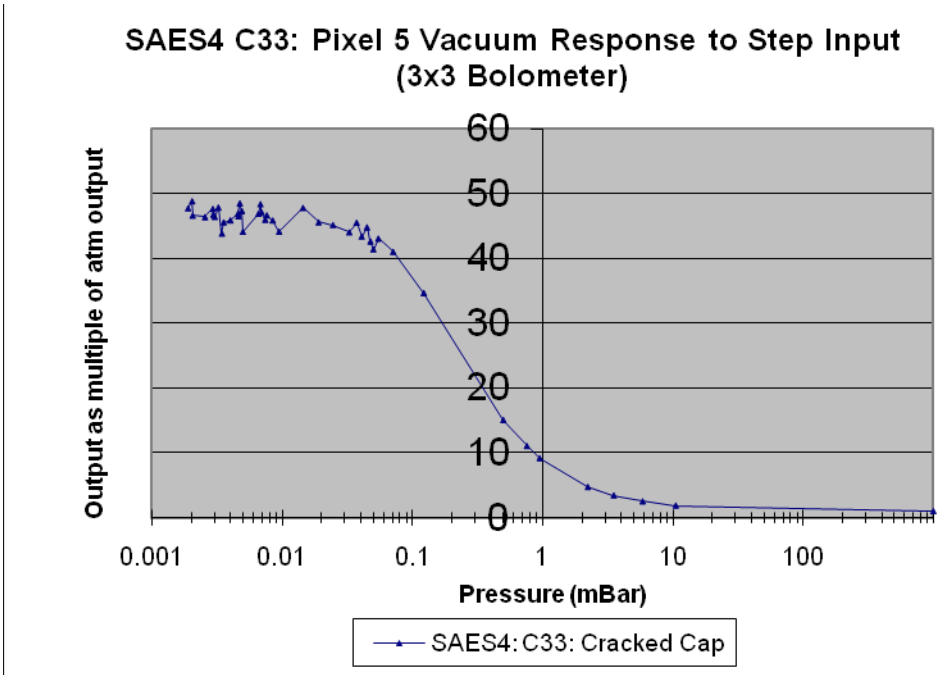
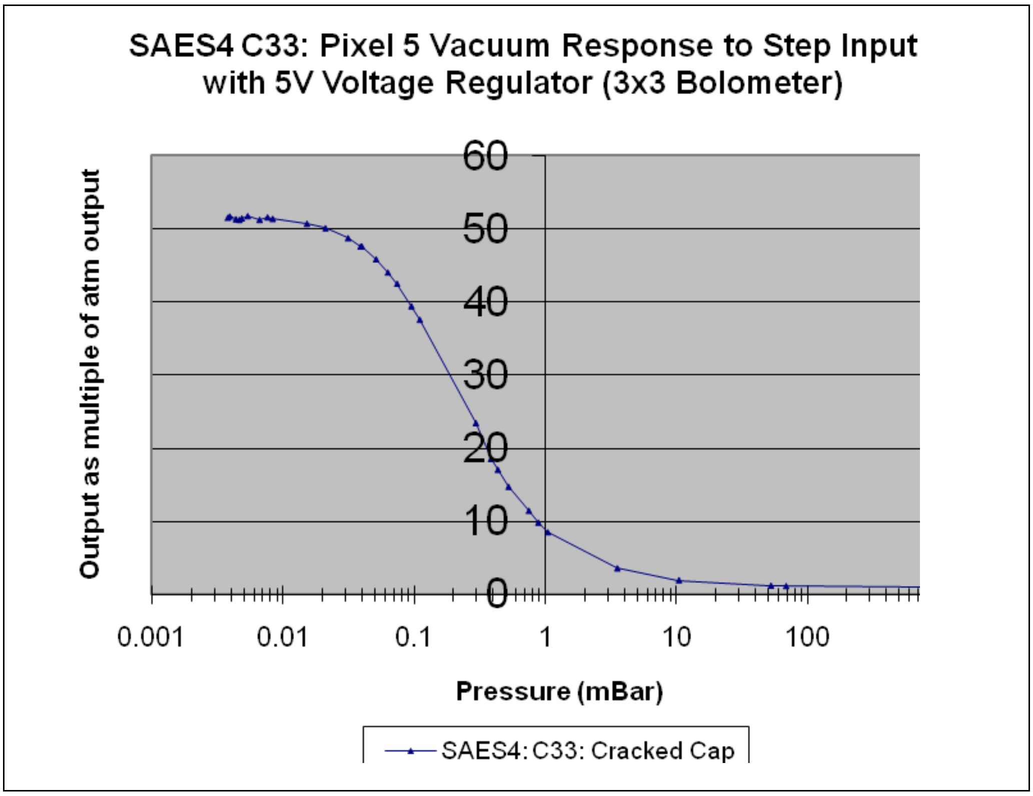


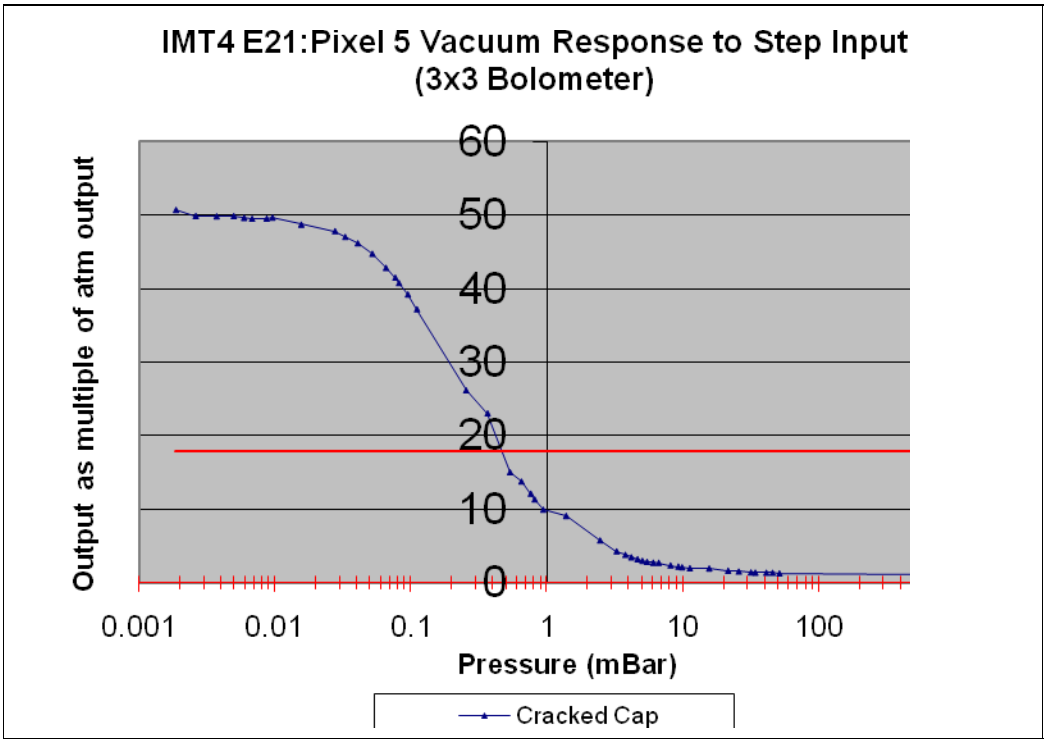
Figure 83: Block diagram of decreasing pressure flow (Pfeiffer Vacuum Manual).

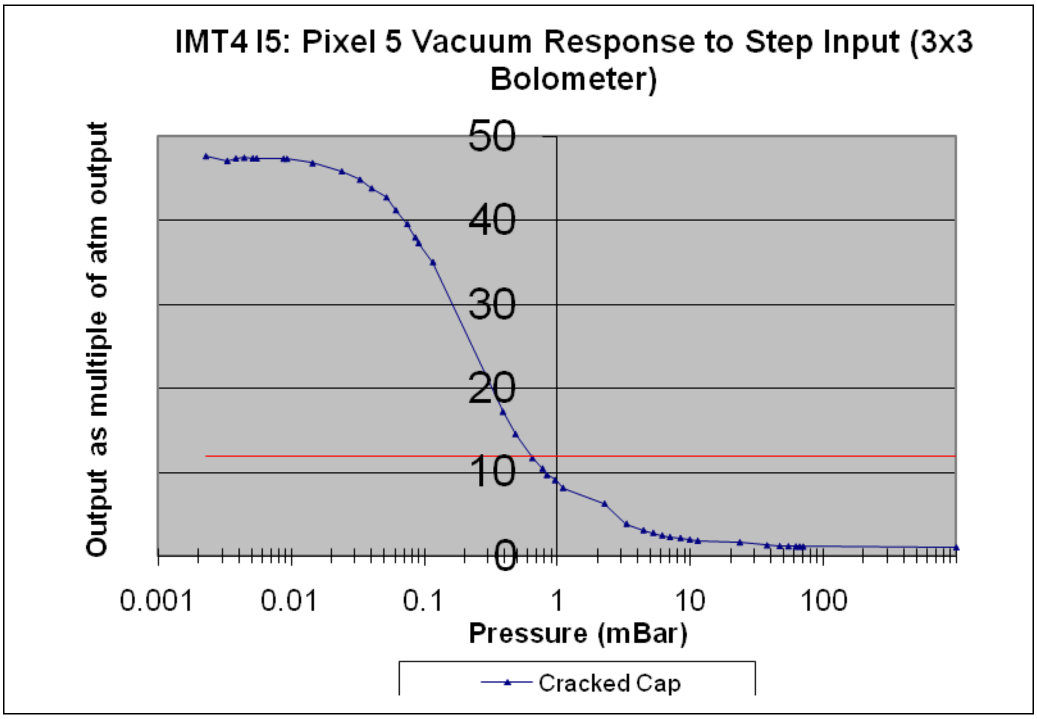
Appendix C: Pressure Experiment Data

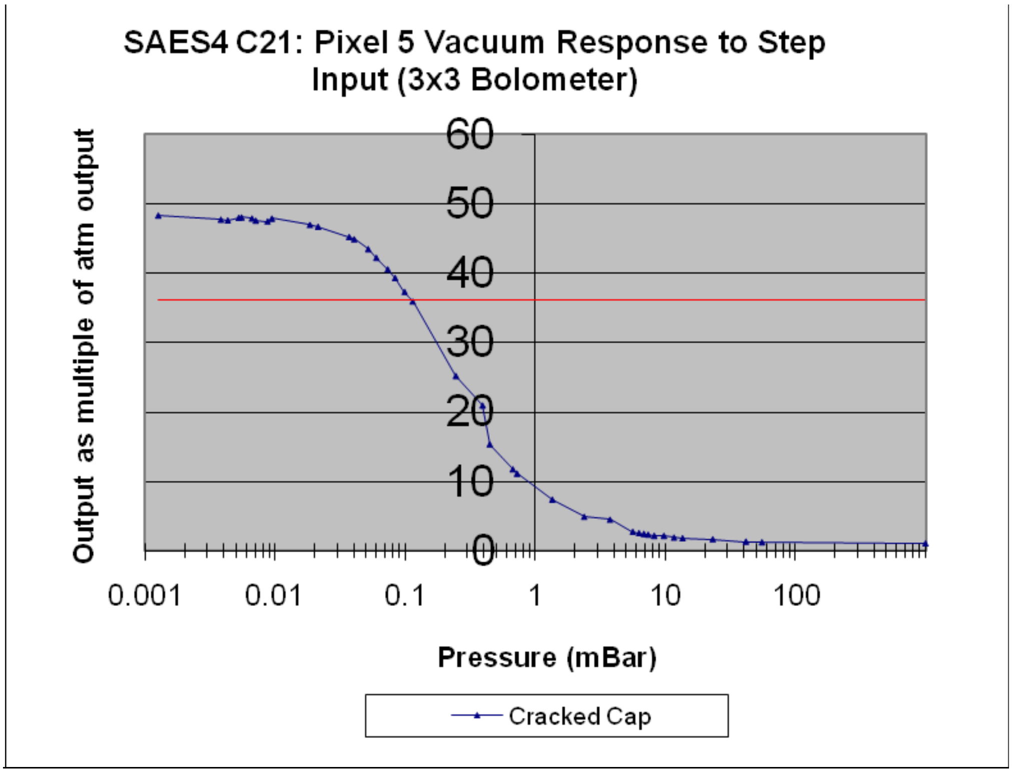
The following graphs contain the pressure curves for various 3x3 and 11x11 bolometers. The straight lines across some of the graphs represent the baseline, which we took before we cracked their caps and created the pressure curves.

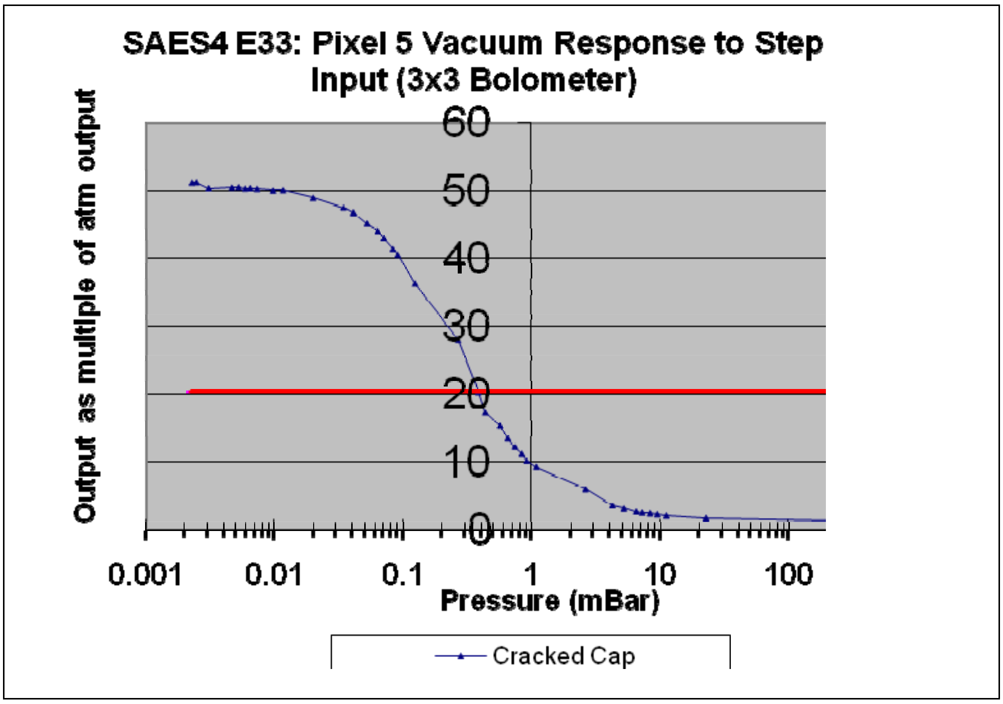


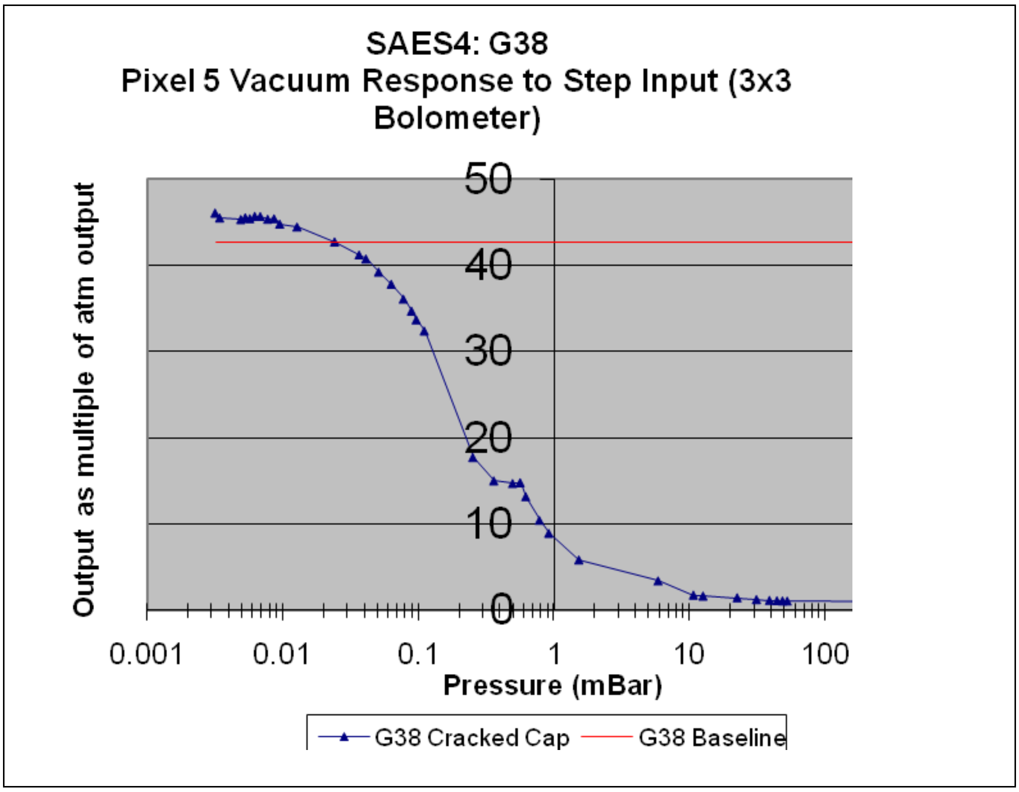


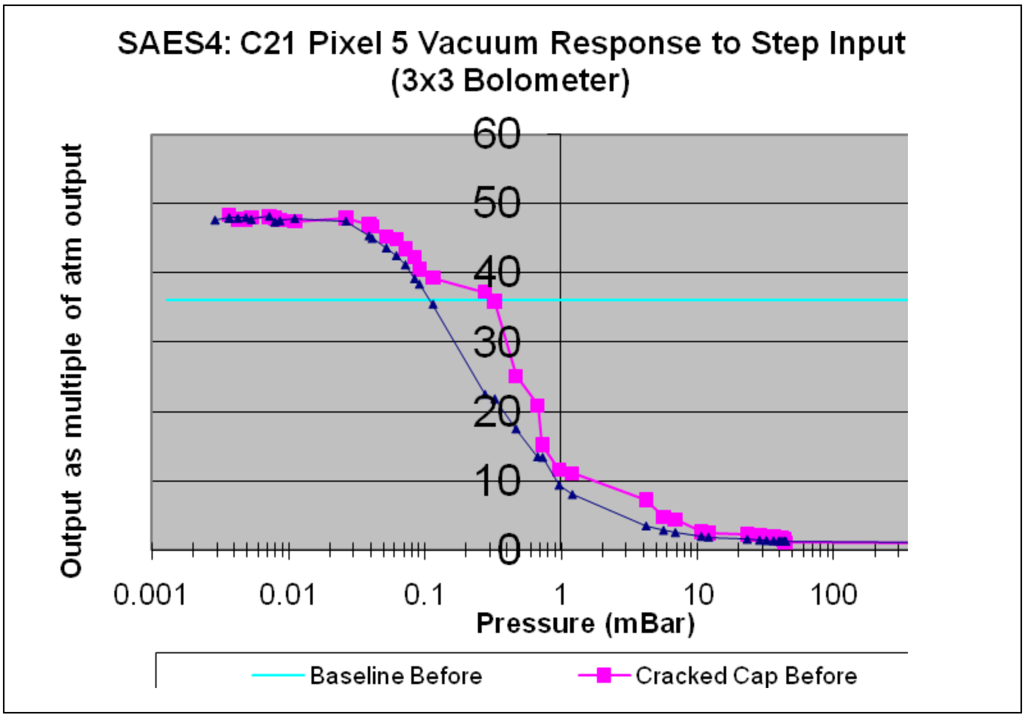

IMT4 E21:Pixel 5 Vacuum Response to Step Input (3x3 Bolometer)
Output as multiple of atm output
60
50
40
30
20
10
0
0.001
0.01
0.1
1
10
100
Pressure (mBar)
Cracked Cap

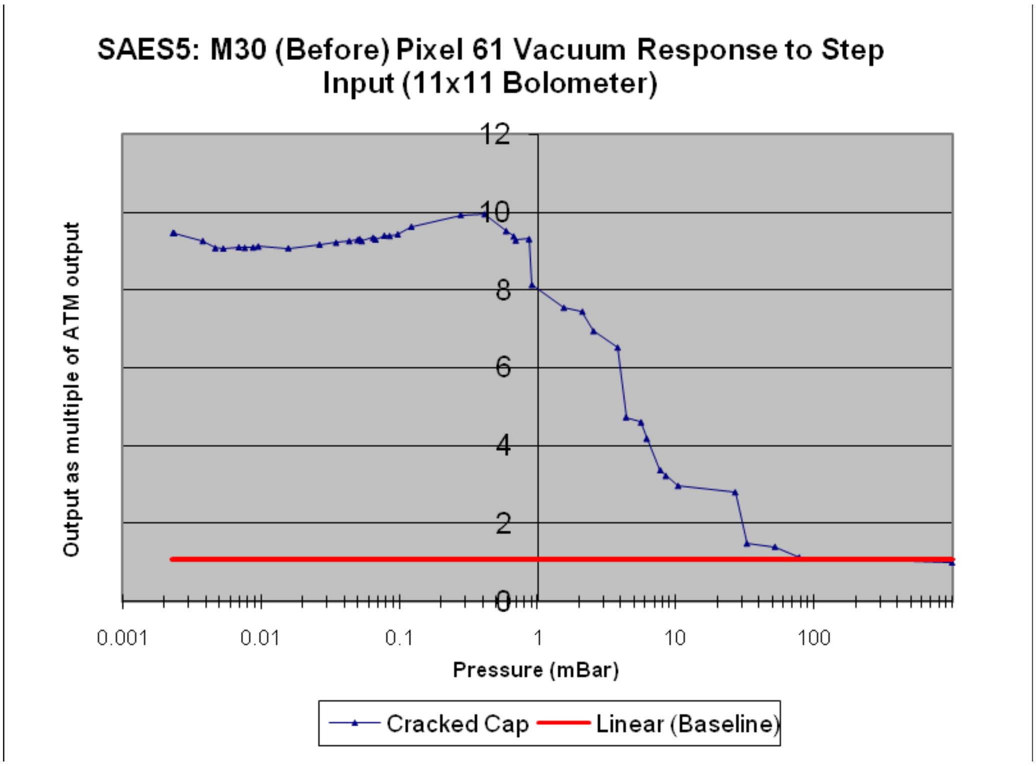

IMT4 I5: Pixel 5 Vacuum Response to Step Input (3x3 Bolometer)
Output as multiple of atm output
50
40
30
20
10
0
0.001
0.01
0.1
1
10
100
Pressure (mBar)
Cracked Cap

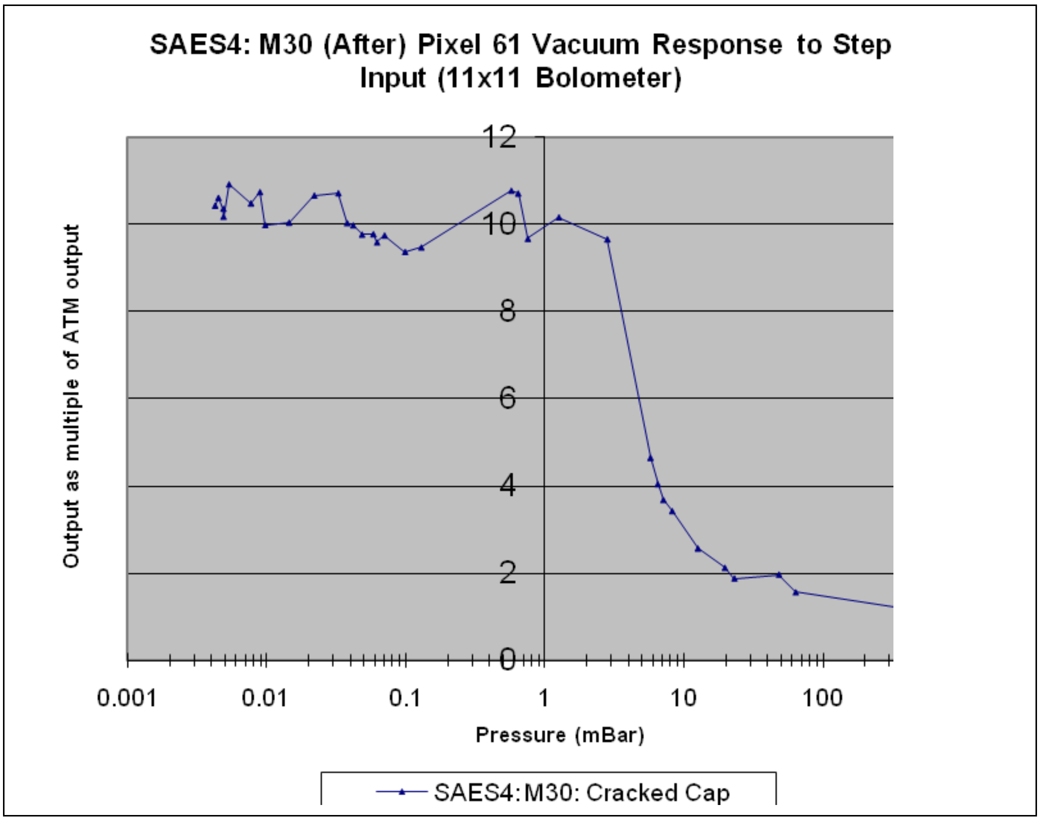

SAES4 C21: Pixel 5 Vacuum Response to Step Input (3x3 Bolometer)
Output as multiple of atm output
60
50
40
30
20
10
0
0.001
0.01
0.1
1
10
100
Pressure (mBar)
Cracked Cap

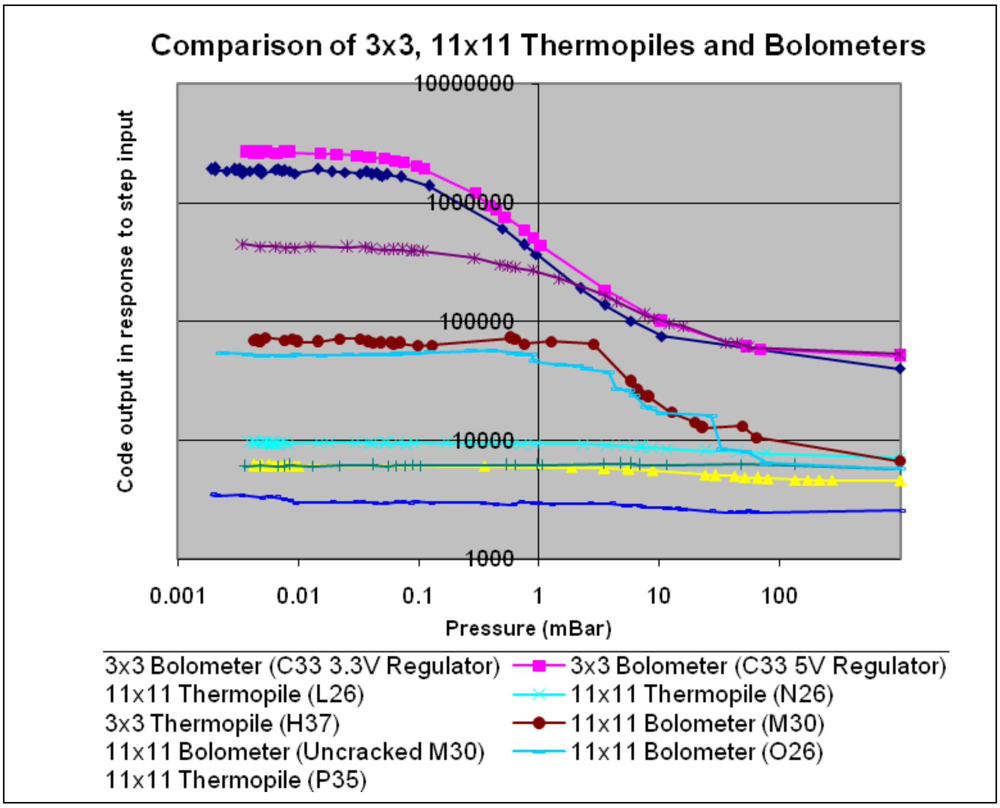

SAES4 E33: Pixel 5 Vacuum Response to Step Input (3x3 Bolometer)
Output as multiple of atm output
60
50
40
30
20
10
0
0.001
0.01
0.1
1
10
100
Pressure (mBar)
Cracked Cap

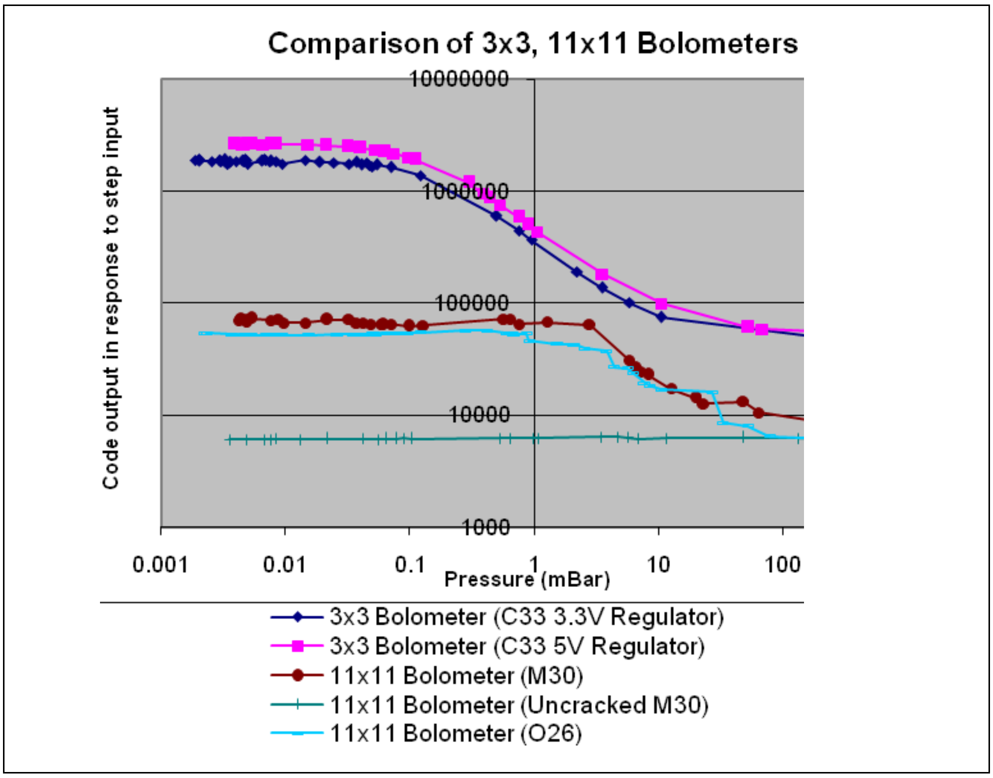

SAES4: G38
Pixel 5 Vacuum Response to Step Input (3x3 Bolometer)
Output as multiple of atm output
50
40
30
20
10
0
0.001
0.01
0.1
1
10
100
Pressure (mBar)
G38 Cracked Cap
G38 Baseline


SAES4: C21 Pixel 5 Vacuum Response to Step Input (3x3 Bolometer)
Output as multiple of atm output
60
50
40
30
20
10
0
0.001
0.01
0.1
1
10
100
Pressure (mBar)
Baseline Before
Cracked Cap Before


SAES5: M30 (Before) Pixel 61 Vacuum Response to Step Input (11x11 Bolometer)
Output as multiple of ATM output
12
10
8
6
4
2
0
0.001
0.01
0.1
1
10
100
Pressure (mBar)
Cracked Cap
Linear (Baseline)


SAES4: M30 (After) Pixel 61 Vacuum Response to Step Input (11x11 Bolometer)
Output as multiple of ATM output
12
10
8
6
4
2
0
0.001
0.01
0.1
1
10
100
Pressure (mBar)
SAES4:M30: Cracked Cap


Comparison of 3x3, 11x11 Thermopiles and Bolometers
Code output in response to step input
10000000
1000000
100000
10000
1000
0.001
0.01
0.1
1
10
100
Pressure (mBar)
3x3 Bolometer (C33 3.3V Regulator)
3x3 Bolometer (C33 5V Regulator)
11x11 Thermopile (L26)
11x11 Thermopile (N26)
3x3 Thermopile (H37)
11x11 Bolometer (M30)
11x11 Bolometer (Uncracked M30)
11x11 Bolometer (O26)
11x11 Thermopile (P35)


Comparison of 3x3, 11x11 Bolometers
Code output in response to step input
10000000
1000000
100000
10000
1000
0.001
0.01
0.1
1
10
100
Pressure (mBar)
3x3 Bolometer (C33 3.3V Regulator)
3x3 Bolometer (C33 5V Regulator)
11x11 Bolometer (M30)
11x11 Bolometer (Uncracked M30)
11x11 Bolometer (O26)

Appendix D: Temperature Forcer Results for Part G30

Figure 84 and Figure 85 contain the results for part G30, which was manufactured with a new process, when it was placed in the temperature forcer. We tested G30 at 10^{o}C and at 30^{o}C.

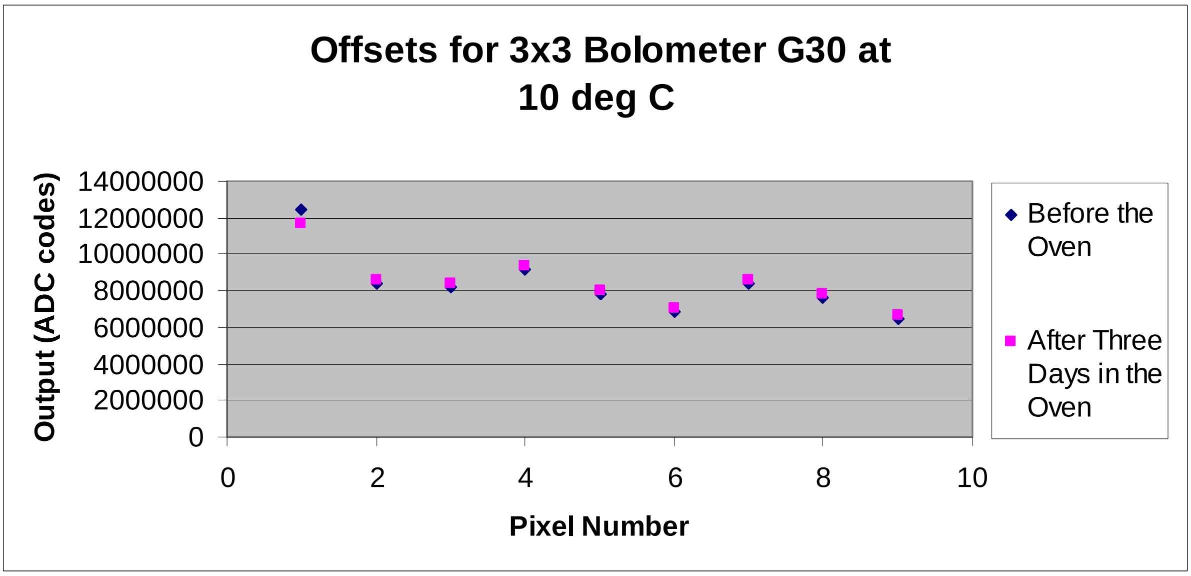


Figure 84: Offsets for 3x3 bolometer G30 at 10 degrees C

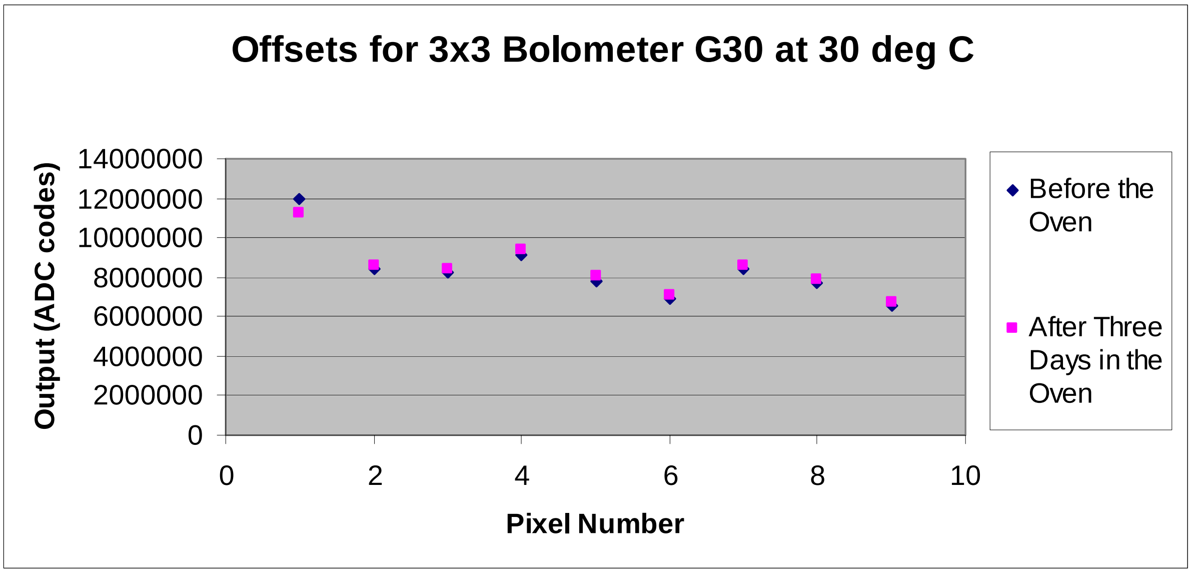


Figure 85: Offset for 3x3 bolometer G30 at 30 degrees C

We noticed that the graphs appear nearly identical.

Appendix E: Visual Basic for the Automated Pressure Experiment

```
Dim filechosen As Boolean
Dim path2file As String
Dim path2putfile As String
Dim allfiles As FileDialogFilter



Private Sub CommandButton1_Click()

With Application.FileDialog(3)
        .AllowMultiSelect = False
        .Title = "Select Pressure Experiment Data"
        .Show
        If .SelectedItems.Count < 1 Then
            MsgBox "No file selected"
            filechosen = False
        ElseIf .SelectedItems.Count = 1 Then
            filepathLabel.Caption = "File: " + .SelectedItems.Item(1)
            path2file = .SelectedItems.Item(1)
            CommandButton2.Enabled = True
        End If
End With
End Sub

Private Sub CommandButton2_Click()
filechosen = False
With Application.FileDialog(2)
        .AllowMultiSelect = False
        .Title = "Select Destination Folder"
        .Show
        If .SelectedItems.Count < 1 Then
            MsgBox "No file selected"
        ElseIf .SelectedItems.Count = 1 Then
            filechosen = True
            path2putfile = .SelectedItems.Item(1)
        End If
End With
If filechosen Then
Call procData
Call saveWkBk
End If
End Sub
Function genNewFile()
Workbooks.Add
```

```
End Function

Private Sub Label3_Click()

End Sub

Private Sub Label4_Click()

End Sub

Private Sub UserForm_Click()

End Sub
Public Sub procData()
    Workbooks.Add
        Range("A5").Select
    With ActiveSheet.QueryTables.Add(Connection:= _
        "TEXT;" + path2file, _
        Destination:=Range("A5"))
        .Name = "!vacumm auto test good"
        .FieldNames = True
        .RowNumbers = False
        .FillAdjacentFormulas = False
        .PreserveFormatting = True
        .RefreshOnFileOpen = False
        .RefreshStyle = xlInsertDeleteCells
        .SavePassword = False
        .SaveData = True
        .AdjustColumnWidth = True
        .RefreshPeriod = 0
        .TextFilePromptOnRefresh = False
        .TextFilePlatform = 437
        .TextFileStartRow = 1
        .TextFileParseType = xlDelimited
        .TextFileTextQualifier = xlTextQualifierDoubleQuote
        .TextFileConsecutiveDelimiter = False
        .TextFileTabDelimiter = True
        .TextFileSemicolonDelimiter = False
        .TextFileCommaDelimiter = False
        .TextFileSpaceDelimiter = False
        .TextFileColumnDataTypes = Array(1, 1, 1, 1)
        .TextFileTrailingMinusNumbers = True
        .Refresh BackgroundQuery:=False
    End With
    Range("A4").Select
    ActiveCell.FormulaR1C1 = "Pressure (mBar)"
    Range("B4").Select
    ActiveCell.FormulaR1C1 = "Spread (Codes)"
    Range("C4").Select
    ActiveCell.FormulaR1C1 = "Max (Codes)"
```

```
Range("D4").Select
ActiveCell.FormulaR1C1 = "Min (Codes)"
Columns("A:A").EntireColumn.AutoFit
Columns("B:B").EntireColumn.AutoFit
Columns("C:C").EntireColumn.AutoFit
Columns("D:D").EntireColumn.AutoFit
Range("A4:D35").Select
With Selection
   .HorizontalAlignment = xlCenter
   .VerticalAlignment = xlBottom
   .WrapText = False
   .Orientation = 0
   .AddIndent = False
   .IndentLevel = 0
   .ShrinkToFit = False
   .ReadingOrder = xlContext
   .MergeCells = False
End With
Columns("C:C").Select
Selection.Insert Shift:=xlToRight
Range("C4").Select
ActiveCell.FormulaR1C1 = "Response"
Range("C5").Select
ActiveCell.FormulaR1C1 = "B5/$B5$"
Range("C5").Select
ActiveCell.FormulaR1C1 = "=RC[-1]/R5C[-1]"
Range("C5").Select
Selection.AutoFill Destination:=Range("C5:C35"), Type:=xlFillDefault
Range("C5:C35").Select
Range("A:A,C:C").Select
Range("C4").Activate
Charts.Add
ActiveChart.ChartType = xlXYScatterSmooth
ActiveChart.SetSourceData Source:=Sheets("Sheet1").Range("A4:A35,C4:C35"), _
   PlotBy:=xlColumns
ActiveChart.SeriesCollection(1).Name = "=""" + partNumber.Text + """"
ActiveChart.Location Where:=xlLocationAsObject, Name:="Sheet1"
With ActiveChart
   .HasTitle = True
   .ChartTitle.Characters.Text = partNumber.Text
   .Axes(xlCategory, xlPrimary).HasTitle = True
   .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = XTitle.Text
   .Axes(xlValue, xlPrimary).HasTitle = True
   .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = YTitle.Text
End With
ActiveChart.SeriesCollection(1).Select
ActiveChart.ChartArea.Select
ActiveChart.Axes(xlCategory).Select
With ActiveChart.Axes(xlCategory)
   .MinimumScaleIsAuto = True
```

```
        .MaximumScaleIsAuto = True
        .MinorUnitIsAuto = True
        .MajorUnitIsAuto = True
        .Crosses = xlCustom
        .CrossesAt = 1
        .ReversePlotOrder = False
        .ScaleType = xlLogarithmic
        .DisplayUnit = xlNone
    End With
    ActiveChart.ChartArea.Select
    ActiveSheet.Shapes("Chart 1").IncrementLeft 159.75
    ActiveSheet.Shapes("Chart 1").IncrementTop -101.25
    ActiveChart.HasLegend = True
    ActiveChart.Legend.Select
    Selection.Position = xlBottom
    ActiveSheet.Shapes("Chart 1").ScaleWidth 1.06, msoFalse, msoScaleFromTopLeft
    ActiveSheet.Shapes("Chart 1").ScaleHeight 1.68, msoFalse, msoScaleFromTopLeft
End Sub
Function saveWkBk()
    ActiveWorkbook.SaveAs Filename:= _
        path2putfile _
        , FileFormat:=xlNormal, Password:="", WriteResPassword:="", _
        ReadOnlyRecommended:=False, CreateBackup:=False
    ActiveWorkbook.Close
End Function
```

Appendix F: Visual Basic for the Angular Response Experiment

```
Dim filechosen As Boolean
Dim path2file As String
Dim path2putfile As String
Dim allfiles As FileDialogFilter



Private Sub CommandButton1_Click()

With Application.FileDialog(3)
        .AllowMultiSelect = False
        .Title = "Select Angular Response Data"
        .Show
        If .SelectedItems.Count < 1 Then
            MsgBox "No file selected"
            filechosen = False
        ElseIf .SelectedItems.Count = 1 Then
            filepathLabel.Caption = "File: " + .SelectedItems.Item(1)
            path2file = .SelectedItems.Item(1)
            CommandButton2.Enabled = True
        End If
End With
End Sub

Private Sub CommandButton2_Click()
filechosen = False
With Application.FileDialog(2)
        .AllowMultiSelect = False
        .Title = "Select Destination Folder"
        .Show
        If .SelectedItems.Count < 1 Then
            MsgBox "No file selected"
        ElseIf .SelectedItems.Count = 1 Then
            filechosen = True
            path2putfile = .SelectedItems.Item(1)
        End If
End With
If filechosen Then
Call procData
Call saveWkBk
End If
End Sub
Function genNewFile()
Workbooks.Add
```

```
End Function

Private Sub Label3_Click()

End Sub

Private Sub UserForm_Click()

End Sub
Public Sub procData()
    Workbooks.Add
    With ActiveSheet.QueryTables.Add(Connection:= _
        "TEXT;" + path2file, Destination:=Range("A1"))
        .Name = "new data 90 deg"
        .FieldNames = True
        .RowNumbers = False
        .FillAdjacentFormulas = False
        .PreserveFormatting = True
        .RefreshOnFileOpen = False
        .RefreshStyle = xlInsertDeleteCells
        .SavePassword = False
        .SaveData = True
        .AdjustColumnWidth = True
        .RefreshPeriod = 0
        .TextFilePromptOnRefresh = False
        .TextFilePlatform = 437
        .TextFileStartRow = 1
        .TextFileParseType = xlDelimited
        .TextFileTextQualifier = xlTextQualifierDoubleQuote
        .TextFileConsecutiveDelimiter = False
        .TextFileTabDelimiter = True
        .TextFileSemicolonDelimiter = False
        .TextFileCommaDelimiter = False
        .TextFileSpaceDelimiter = False
        .TextFileColumnDataTypes = Array(1, 1, 1, 1, 1, 1, 1, 1, 1)
        .TextFileTrailingMinusNumbers = True
        .Refresh BackgroundQuery:=False
    End With
    Range("A1:A190").Select
    ActiveWindow.SmallScroll Down:=-210
    Range("K1").Select
    ActiveCell.FormulaR1C1 = "=RC[-10]-R1C[-10]"
    Range("K1").Select
    Selection.AutoFill Destination:=Range("K1:K190"), Type:=xlFillDefault
    Range("K1:K190").Select
    Selection.AutoFill Destination:=Range("K1:S190"), Type:=xlFillDefault
    Range("K1:S190").Select
    ActiveWindow.SmallScroll Down:=-171
    Range("L1").Select
    Range("K1:K190").Select
```

```
Charts.Add
ActiveChart.ChartType = xlXYScatterSmooth
ActiveChart.SetSourceData Source:=Sheets("Sheet1").Range("K1:K190"), PlotBy _
    :=xlColumns
ActiveChart.SeriesCollection.NewSeries
ActiveChart.SeriesCollection.NewSeries
ActiveChart.SeriesCollection.NewSeries
ActiveChart.SeriesCollection.NewSeries
ActiveChart.SeriesCollection.NewSeries
ActiveChart.SeriesCollection.NewSeries
ActiveChart.SeriesCollection.NewSeries
ActiveChart.SeriesCollection.NewSeries
ActiveChart.SeriesCollection(1).Name = "=""Pixel 1"""
ActiveChart.SeriesCollection(2).Values = "=Sheet1!R1C12:R190C12"
ActiveChart.SeriesCollection(2).Name = "=""Pixel 2"""
ActiveChart.SeriesCollection(3).Values = "=Sheet1!R1C13:R190C13"
ActiveChart.SeriesCollection(3).Name = "=""Pixel 3"""
ActiveChart.SeriesCollection(4).Values = "=Sheet1!R1C14:R190C14"
ActiveChart.SeriesCollection(4).Name = "=""Pixel 4"""
ActiveChart.SeriesCollection(5).Values = "=Sheet1!R1C15:R190C15"
ActiveChart.SeriesCollection(5).Name = "=""Pixel 5"""
ActiveChart.SeriesCollection(6).Values = "=Sheet1!R1C16:R190C16"
ActiveChart.SeriesCollection(6).Name = "=""Pixel 6"""
ActiveChart.SeriesCollection(7).Values = "=Sheet1!R1C17:R190C17"
ActiveChart.SeriesCollection(7).Name = "=""Pixel 7"""
ActiveChart.SeriesCollection(8).Values = "=Sheet1!R1C18:R190C18"
ActiveChart.SeriesCollection(8).Name = "=""Pixel 8"""
ActiveChart.SeriesCollection(9).Values = "=Sheet1!R1C19:R190C19"
ActiveChart.SeriesCollection(9).Name = "=""Pixel 9"""
ActiveChart.Location Where:=xlLocationAsObject, Name:="Sheet1"
ActiveSheet.Shapes("Chart 1").IncrementLeft -145.5
ActiveSheet.Shapes("Chart 1").IncrementTop -99#
ActiveSheet.Shapes("Chart 1").ScaleWidth 1.47, msoFalse, msoScaleFromTopLeft
ActiveSheet.Shapes("Chart 1").ScaleHeight 1.74, msoFalse, msoScaleFromTopLeft
ActiveSheet.Shapes("Chart 1").ScaleWidth 1.1, msoFalse, msoScaleFromTopLeft
ActiveSheet.Shapes("Chart 1").ScaleHeight 1.14, msoFalse, msoScaleFromTopLeft
ActiveChart.Axes(xlCategory).Select
ActiveChart.Axes(xlValue).Select
With ActiveChart.Axes(xlValue)
    .MinimumScale = 0
    .MaximumScaleIsAuto = True
    .MinorUnitIsAuto = True
    .MajorUnitIsAuto = True
    .Crosses = xlAutomatic
    .ReversePlotOrder = False
    .ScaleType = xlLinear
    .DisplayUnit = xlNone
End With
ActiveChart.ChartArea.Select
With ActiveChart
```

```
    .HasTitle = True
    .ChartTitle.Characters.Text = ChartTitle1.Text
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = XTitle.Text
    .Axes(xlValue, xlPrimary).HasTitle = True
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = YTitle.Text
  End With
  ActiveChart.HasLegend = True
End Sub
Function saveWkBk()
  ActiveWorkbook.SaveAs Filename:= _
    path2putfile _
    , FileFormat:=xlNormal, Password:="", WriteResPassword:="", _
    ReadOnlyRecommended:=False, CreateBackup:=False
  ActiveWorkbook.Close
End Function
```

Appendix G: Visual Basic for the Qualification Test

```
Dim working_directory As String 'Path of ProcessData.xls
Dim partList As New Collection  'Collection of parts with data
Dim flag As Boolean             'Basic flag used in 'Generate All Graphs' logic
Dim curChart As String


Private Sub CheckBox1_Click()

End Sub

Private Sub CodeYes_Click()
flag = True
If CodeYes = False Then GenAllGraphs = False
End Sub


Private Sub DeviceList_Change()
If DeviceList.Text <> "" Then
   If (Dir(working_directory + "\demo_data\data_" + DeviceList.Text + "\ir_code_p5.dat") >
"") Then CodesFileLoaded.Caption = ".." + "\demo_data\data_" + DeviceList.Text +
"\ir_code_p5.dat" Else MsgBox "Sorry no code data!"
   If (Dir(working_directory + "\demo_data\data_" + DeviceList.Text + "\ir_snr_p5.dat") >
"") Then SNRFileLoaded.Caption = ".." + "\demo_data\data_" + DeviceList.Text +
"\ir_snr_p5.dat" Else MsgBox "Sorry no SNR data!"
   If (Dir(working_directory + "\demo_data\data_" + DeviceList.Text + "\ir_elec_step.dat") >
"") Then ElectricalFileLoaded.Caption = ".." + "\demo_data\data_" + DeviceList.Text +
"\ir_elec_step.dat" Else MsgBox "Sorry no electrical data!"
   If (Dir(working_directory + "\demo_data\data_" + DeviceList.Text +
"\ir_optical_step.dat") > "") Then OpticalFileLoaded.Caption = ".." + "\demo_data\data_" +
DeviceList.Text + "\ir_optical_step.dat" Else MsgBox "Sorry no optical data!"
   If (Dir(working_directory + "\demo_data\data_" + DeviceList.Text + "\power.dat") > "")
Then PowerFileLoaded.Caption = ".." + "\demo_data\data_" + DeviceList.Text +
"\power.dat" Else MsgBox "Sorry no power data!"
   If (Dir(working_directory + "\demo_data\data_" + DeviceList.Text + "\resistor_data.dat")
> "") Then ResistorFileLoaded.Caption = ".." + "\demo_data\data_" + DeviceList.Text +
"\resistor_data.dat" Else MsgBox "Sorry no resistor data!"
   If (Dir(working_directory + "\demo_data\data_" + DeviceList.Text + "\tcr.dat") > "") Then
TCRFileLoaded.Caption = ".." + "\demo_data\data_" + DeviceList.Text + "\tcr.dat" Else
MsgBox "Sorry no TCR data!"
   CodeTitle.Text = "Code for Pixel 5 of Part " + DeviceList.Text
   SNRTitle.Text = "SNR for Pixel 5 of Part " + DeviceList.Text
   TCRTitle.Text = "TCR of Part " + DeviceList.Text
   ElectricalTitle.Text = "Electrical of Part " + DeviceList.Text
   ResistorTitle.Text = "Resistor Data of Part " + DeviceList.Text
   PowerTitle.Text = "Power Data of Part " + DeviceList.Text
   OpticalTitle.Text = "Optical Data of Part " + DeviceList.Text
   If configForm.Height = 63 Then
   For i = 1 To 39
   configForm.Height = configForm.Height + 4
   configForm.Width = configForm.Width + 6
   Next i
   End If
   GenAllGraphs.Visible = True
End If

End Sub

Private Sub ElectricalYes_Click()
flag = True
If ElectricalYes = False Then GenAllGraphs = False
End Sub



Private Sub Frame1_Click()

End Sub

Private Sub GenAllGraphs_Click()
If (flag = True And GenAllGraphs = False) Then

Else
   If GenAllGraphs = True Then
      CodeYes = True
      SNRYes = True
      ElectricalYes = True
      PowerYes = True
      ResistorYes = True
      TCRYes = True
      OpticalYes = True
   Else
      CodeYes = False
      SNRYes = False
      ElectricalYes = False
      PowerYes = False
      ResistorYes = False
      TCRYes = False
      OpticalYes = False
   End If
End If
   flag = False
End Sub

Private Sub genBtn_Click()
If ((CodeYes Or SNRYes Or PowerYes Or ResistorYes Or TCRYes Or OpticalYes Or
ElectricalYes) = False) Then
```

```
MsgBox "You haven't selected any data to graph!"
Else
Call genNewFile
If CodeYes Then Call genCodesGraph
If SNRYes Then Call genSNRGraph
If ElectricalYes Then Call genElectricalGraph
If TCRYes Then Call genTCRGraph
If PowerYes Then Call genPowerGraph
If ResistorYes Then Call genResistorGraph
If OpticalYes Then Call genOpticalGraph
Call saveWkBk
End If
exitYN = MsgBox("Would you like to close this tool?", vbYesNo, "Exit?")
If exitYN = vbYes Then Unload Me
End Sub

Private Sub OpticalYes_Click()
flag = True
If OpticalYes = False Then GenAllGraphs = False
End Sub

Private Sub PowerYes_Click()
flag = True
If PowerYes = False Then GenAllGraphs = False
End Sub

Private Sub ResistorYes_Click()
flag = True
If ResistorYes = False Then GenAllGraphs = False
End Sub

Private Sub SNRYes_Click()
flag = True
If SNRYes = False Then GenAllGraphs = False
End Sub

Private Sub TCRYes_Click()
flag = True
If TCRYes = False Then GenAllGraphs = False
End Sub

Private Sub TextBox15_Change()

End Sub

Private Sub UserForm_Click()

End Sub

Private Sub UserForm_Initialize()
```

```
configForm.Height = 63
configForm.Width = 144
working_directory = ActiveWorkbook.Path
Call getPartNumbers
For d = 1 To partList.Count
splitArray = Split(partList(d), "_")
DeviceList.AddItem (splitArray(1))
Next d
For i = 1 To 9
CodePixel.AddItem (i)
SNRPixel.AddItem (i)
Next i
DeviceList.SetFocus
End Sub

Function getPartNumbers() As Collection
   Dim FSO As FileSystemObject
   Dim SubFld As Folder
   Dim Sub2fld As Folder
   Dim splitArray() As String

   Set FSO = New FileSystemObject

   For Each SubFld In FSO.GetFolder(ActiveWorkbook.Path).SubFolders
      For Each Sub2fld In SubFld.SubFolders
         partList.Add (UCase(Sub2fld.Name))
      Next Sub2fld
   Next SubFld
End Function
Function saveWkBk()
   ActiveWorkbook.SaveAs Filename:= _
      working_directory + "\demo_data\data_" + DeviceList.Text + "\Data.xls" _
      , FileFormat:=xlNormal, Password:="", WriteResPassword:="", _
      ReadOnlyRecommended:=False, CreateBackup:=False
   ActiveWorkbook.Close
End Function
Function genCodesGraph()
Sheets("Codes").Select
   With ActiveSheet.QueryTables.Add(Connection:= _
      "TEXT;" + working_directory + "\demo_data\data_" + DeviceList.Text + "\ir_code_p"
+ CodePixel.Text + ".dat", Destination:=Range( _
      "A2"))
      .Name = "ir_code_p5"
      .FieldNames = True
      .RowNumbers = False
      .FillAdjacentFormulas = False
      .PreserveFormatting = True
      .RefreshOnFileOpen = False
      .RefreshStyle = xlInsertDeleteCells
      .SavePassword = False
      .SaveData = True
      .AdjustColumnWidth = True
      .RefreshPeriod = 0
      .TextFilePromptOnRefresh = False
      .TextFilePlatform = 850
      .TextFileStartRow = 1
      .TextFileParseType = xlDelimited
      .TextFileTextQualifier = xlTextQualifierDoubleQuote
      .TextFileConsecutiveDelimiter = False
      .TextFileTabDelimiter = True
      .TextFileSemicolonDelimiter = False
      .TextFileCommaDelimiter = False
      .TextFileSpaceDelimiter = False
      .TextFileColumnDataTypes = Array(1)
      .TextFileTrailingMinusNumbers = True
      .Refresh BackgroundQuery:=False
   End With
   Range("A2").Select
   Range(Selection, ActiveCell.SpecialCells(xlLastCell)).Select
   Charts.Add
   ActiveChart.ChartType = xlXYScatterSmooth
   ActiveChart.SetSourceData Source:=Sheets("Codes").Range("A2:A257"), PlotBy _
      :=xlColumns
   ActiveChart.Location Where:=xlLocationAsObject, Name:="Codes"
   With ActiveChart
      .HasTitle = True
      .ChartTitle.Characters.Text = CodeTitle.Text
      .Axes(xlCategory, xlPrimary).HasTitle = True
      .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = CodeX.Text
      .Axes(xlValue, xlPrimary).HasTitle = True
      .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = CodeY.Text
   End With
End Function

Function genSNRGraph()
Sheets("SNR").Select
   With ActiveSheet.QueryTables.Add(Connection:= _
      "TEXT;" + working_directory + "\demo_data\data_" + DeviceList.Text + "\ir_snr_p" +
SNRPixel.Text + ".dat", Destination:=Range( _
      "A2"))
      .Name = "ir_code_p5"
      .FieldNames = True
      .RowNumbers = False
      .FillAdjacentFormulas = False
      .PreserveFormatting = True
      .RefreshOnFileOpen = False
      .RefreshStyle = xlInsertDeleteCells
      .SavePassword = False
      .SaveData = True
      .AdjustColumnWidth = True
      .RefreshPeriod = 0
      .TextFilePromptOnRefresh = False
      .TextFilePlatform = 850
      .TextFileStartRow = 1
      .TextFileParseType = xlDelimited
      .TextFileTextQualifier = xlTextQualifierDoubleQuote
      .TextFileConsecutiveDelimiter = False
      .TextFileTabDelimiter = True
      .TextFileSemicolonDelimiter = False
      .TextFileCommaDelimiter = False
      .TextFileSpaceDelimiter = False
      .TextFileColumnDataTypes = Array(1)
      .TextFileTrailingMinusNumbers = True
      .Refresh BackgroundQuery:=False
   End With
   Range("A2").Select
   Range(Selection, ActiveCell.SpecialCells(xlLastCell)).Select
   Charts.Add
   ActiveChart.ChartType = xlXYScatterSmooth
   ActiveChart.SetSourceData Source:=Sheets("SNR").Range("A2:A257"), PlotBy _
      :=xlColumns
   ActiveChart.Location Where:=xlLocationAsObject, Name:="SNR"
   With ActiveChart
      .HasTitle = True
      .ChartTitle.Characters.Text = SNRTitle.Text
      .Axes(xlCategory, xlPrimary).HasTitle = True
      .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = SNRX.Text
      .Axes(xlCategory).MinimumScale = 0
      .Axes(xlCategory).MaximumScale = 130
      .Axes(xlValue, xlPrimary).HasTitle = True
      .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = SNRY.Text
   End With
End Function

Function genPowerGraph()
Sheets("Power").Select
   With ActiveSheet.QueryTables.Add(Connection:= _
      "TEXT;" + working_directory + "\demo_data\data_" + DeviceList.Text + "\power.dat" _
      , Destination:=Range("A2"))
      .Name = "power"
      .FieldNames = True
      .RowNumbers = False
      .FillAdjacentFormulas = False
      .PreserveFormatting = True
      .RefreshOnFileOpen = False
      .RefreshStyle = xlInsertDeleteCells
      .SavePassword = False
      .SaveData = True
      .AdjustColumnWidth = True
      .RefreshPeriod = 0
```

```
        .TextFilePromptOnRefresh = False
        .TextFilePlatform = 437
        .TextFileStartRow = 1
        .TextFileParseType = xlDelimited
        .TextFileTextQualifier = xlTextQualifierDoubleQuote
        .TextFileConsecutiveDelimiter = False
        .TextFileTabDelimiter = True
        .TextFileSemicolonDelimiter = False
        .TextFileCommaDelimiter = False
        .TextFileSpaceDelimiter = False
        .TextFileColumnDataTypes = Array(1)
        .TextFileTrailingMinusNumbers = True
        .Refresh BackgroundQuery:=False
    End With
    Range("A2:A10").Select
    Charts.Add
    ActiveChart.ChartType = xlXYScatterLines
    ActiveChart.SetSourceData Source:=Sheets("Power").Range("A2:A10"), PlotBy _
        :=xlColumns
    ActiveChart.Location Where:=xlLocationAsObject, Name:="Power"
    With ActiveChart
        .HasTitle = True
        .ChartTitle.Characters.Text = PowerTitle.Text
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = PowerX.Text
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = PowerY.Text
    End With
    ActiveChart.HasLegend = True
    ActiveChart.Legend.Select
    Selection.Position = xlBottom
End Function

Function genTCRGraph()
Sheets("TCR").Select
    With ActiveSheet.QueryTables.Add(Connection:= _
"TEXT;" + working_directory + "\demo_data\data_" + DeviceList.Text + "\tcr.dat" _
        , Destination:=Range("A2"))
        .Name = "tcr_1"
        .FieldNames = True
        .RowNumbers = False
        .FillAdjacentFormulas = False
        .PreserveFormatting = True
        .RefreshOnFileOpen = False
        .RefreshStyle = xlInsertDeleteCells
        .SavePassword = False
        .SaveData = True
        .AdjustColumnWidth = True
        .RefreshPeriod = 0
        .TextFilePromptOnRefresh = False
        .TextFilePlatform = 437
        .TextFileStartRow = 1
        .TextFileParseType = xlDelimited
        .TextFileTextQualifier = xlTextQualifierDoubleQuote
        .TextFileConsecutiveDelimiter = True
        .TextFileTabDelimiter = False
        .TextFileSemicolonDelimiter = False
        .TextFileCommaDelimiter = False
        .TextFileSpaceDelimiter = True
        .TextFileColumnDataTypes = Array(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, _
        1, 1)
        .TextFileTrailingMinusNumbers = True
        .Refresh BackgroundQuery:=False
    End With
    Range("W3:W11").Select
    Charts.Add
    ActiveChart.ChartType = xlXYScatterLines
    ActiveChart.SetSourceData Source:=Sheets("TCR").Range("W3:W11"), PlotBy _
        :=xlColumns
    ActiveChart.Location Where:=xlLocationAsObject, Name:="TCR"
    With ActiveChart
        .HasTitle = True
        .ChartTitle.Characters.Text = TCRTitle.Text
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = TCRX.Text
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = TCRY.Text
    End With
    ActiveChart.HasLegend = True
    ActiveChart.Legend.Select
    Selection.Position = xlBottom
End Function

Function genOpticalGraph()
Sheets("Optical").Select
    With ActiveSheet.QueryTables.Add(Connection:= _
"TEXT;" + working_directory + "\demo_data\data_" + DeviceList.Text +
"\ir_optical_step.dat" _
        , Destination:=Range("A2"))
        .Name = "ir_optical_step_4"
        .FieldNames = True
        .RowNumbers = False
        .FillAdjacentFormulas = False
        .PreserveFormatting = True
        .RefreshOnFileOpen = False
        .RefreshStyle = xlInsertDeleteCells
        .SavePassword = False
        .SaveData = True
        .AdjustColumnWidth = True
        .RefreshPeriod = 0
        .TextFilePromptOnRefresh = False
        .TextFilePlatform = 437
        .TextFileStartRow = 1
        .TextFileParseType = xlFixedWidth
        .TextFileTextQualifier = xlTextQualifierDoubleQuote
        .TextFileConsecutiveDelimiter = False
        .TextFileTabDelimiter = True
        .TextFileSemicolonDelimiter = False
        .TextFileCommaDelimiter = False
        .TextFileSpaceDelimiter = False
        .TextFileColumnDataTypes = Array(1, 1, 1, 1)
        .TextFileFixedColumnWidths = Array(8, 8, 1)
        .TextFileTrailingMinusNumbers = True
        .Refresh BackgroundQuery:=False
    End With
    Columns("C:C").Select
    Selection.Delete Shift:=xlToLeft
    Range("E3").Select
    ActiveCell.FormulaR1C1 = "=MAX(C[-4])/COUNT(C[-4])"
    Range("E3").Select
    Selection.Copy
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Range("A2").Select
    Application.CutCopyMode = False
    ActiveCell.FormulaR1C1 = "=R[-1]C+R3C5"
    Range("A2").Select
    Selection.AutoFill Destination:=Range("A2:A351"), Type:=xlFillDefault
    Range("A2:A351").Select
    ActiveWindow.SmallScroll Down:=-354
    Columns("B:B").Select
    Selection.Delete Shift:=xlToLeft
    Range("A2:B351").Select
    Charts.Add
    ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
    ActiveChart.SetSourceData Source:=Sheets("Optical").Range("A2:B351"), PlotBy _
        :=xlColumns
    ActiveChart.Location Where:=xlLocationAsObject, Name:="Optical"
    With ActiveChart
        .HasTitle = True
        .ChartTitle.Characters.Text = OpticalTitle.Text
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = OpticalX.Text
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = OpticalY.Text
    End With
    ActiveChart.HasLegend = True
    ActiveChart.Legend.Select
    Selection.Position = xlBottom
```

```
End Function

Function genElectricalGraph()
Sheets("Electrical").Select
   With ActiveSheet.QueryTables.Add(Connection:= _
      "TEXT;" + working_directory + "\demo_data\data_" + DeviceList.Text +
"\ir_elec_step.dat" _
      , Destination:=Range("A2"))
      .Name = "ir_elec_step_4"
      .FieldNames = True
      .RowNumbers = False
      .FillAdjacentFormulas = False
      .PreserveFormatting = True
      .RefreshOnFileOpen = False
      .RefreshStyle = xlInsertDeleteCells
      .SavePassword = False
      .SaveData = True
      .AdjustColumnWidth = True
      .RefreshPeriod = 0
      .TextFilePromptOnRefresh = False
      .TextFilePlatform = 437
      .TextFileStartRow = 1
      .TextFileParseType = xlDelimited
      .TextFileTextQualifier = xlTextQualifierDoubleQuote
      .TextFileConsecutiveDelimiter = True
      .TextFileTabDelimiter = False
      .TextFileSemicolonDelimiter = False
      .TextFileCommaDelimiter = False
      .TextFileSpaceDelimiter = True
      .TextFileColumnDataTypes = Array(1, 1)
      .TextFileTrailingMinusNumbers = True
      .Refresh BackgroundQuery:=False
   End With
   Range("B253").Select
   ActiveCell.FormulaR1C1 = "=R[-2]C[-1]/COUNT(R[-251]C[-1]:R[-2]C[-1])"
   Range("B253").Select
   Selection.Copy
   Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
      :=False, Transpose:=False
   Range("A2").Select
   Application.CutCopyMode = False
   ActiveCell.FormulaR1C1 = "=R[-1]C+R253C2"
   Range("A2").Select
   Selection.AutoFill Destination:=Range("A2:A251"), Type:=xlFillDefault
   Range("A2:A251").Select
   ActiveWindow.SmallScroll Down:=-66
   Range("A2:B251").Select
   Charts.Add
   ActiveChart.ChartType = xlXYScatterSmooth
   ActiveChart.SetSourceData Source:=Sheets("Electrical").Range("A2:B251"), PlotBy _
      :=xlColumns
   ActiveChart.Location Where:=xlLocationAsObject, Name:="Electrical"
   With ActiveChart
      .HasTitle = True
      .ChartTitle.Characters.Text = ElectricalTitle.Text
      .Axes(xlCategory, xlPrimary).HasTitle = True
      .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = ElectricalX.Text
      .Axes(xlValue, xlPrimary).HasTitle = True
      .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = ElectricalY.Text
   End With
   ActiveChart.HasLegend = True
   ActiveChart.Legend.Select
   Selection.Position = xlBottom
End Function

Function genResistorGraph()
Sheets("Resistor").Select
   With ActiveSheet.QueryTables.Add(Connection:= _
     "TEXT;" + working_directory + "\demo_data\data_" + DeviceList.Text +
"\resistor_data.dat" _
      , Destination:=Range("A2"))
      .Name = "resistor_data"
      .FieldNames = True
      .RowNumbers = False
      .FillAdjacentFormulas = False
      .PreserveFormatting = True
      .RefreshOnFileOpen = False
      .RefreshStyle = xlInsertDeleteCells
      .SavePassword = False
      .SaveData = True
      .AdjustColumnWidth = True
      .RefreshPeriod = 0
      .TextFilePromptOnRefresh = False
      .TextFilePlatform = 437
      .TextFileStartRow = 1
      .TextFileParseType = xlDelimited
      .TextFileTextQualifier = xlTextQualifierDoubleQuote
      .TextFileConsecutiveDelimiter = True
      .TextFileTabDelimiter = False
      .TextFileSemicolonDelimiter = False
      .TextFileCommaDelimiter = False
      .TextFileSpaceDelimiter = True
      .TextFileColumnDataTypes = Array(1, 1, 1, 1)
      .TextFileTrailingMinusNumbers = True
      .Refresh BackgroundQuery:=False
   End With
   Range("A4:A12").Select
   Charts.Add
   ActiveChart.ChartType = xlXYScatterLines
   ActiveChart.SeriesCollection.NewSeries
   ActiveChart.SeriesCollection(2).XValues = ""
   ActiveChart.SeriesCollection(2).Values = "=Resistor!R13C1:R21C1"
   ActiveChart.Location Where:=xlLocationAsObject, Name:="Resistor"
   With ActiveChart
      .HasTitle = True
      .ChartTitle.Characters.Text = ResistorTitle.Text
      .Axes(xlCategory, xlPrimary).HasTitle = True
      .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = ResistorX.Text
      .Axes(xlValue, xlPrimary).HasTitle = True
      .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = ResistorY.Text
   End With
   ActiveChart.HasLegend = True
   ActiveChart.Legend.Select
   Selection.Position = xlBottom
   ActiveChart.Axes(xlValue).Select
   With ActiveChart.Axes(xlValue)
      .MinimumScale = 30000
      .MaximumScaleIsAuto = True
      .MinorUnitIsAuto = True
      .MajorUnitIsAuto = True
      .Crosses = xlAutomatic
      .ReversePlotOrder = False
      .ScaleType = xlLinear
      .DisplayUnit = xlNone
   End With
End Function

Function genNewFile()
Workbooks.Add Template:=ActiveWorkbook.Path + "\Data.xlt"
End Function
```

Appendix H: Software Installation Guide

```
Experimentation and Data Processing Suite


How to use this document
========================

To view Readme.txt on screen in Notepad, maximize the Notepad window.

To print Readme.txt, open it in Notepad or another word processor, and then
use the Print command on the File menu.

Contents
========

- Minimum System Requirements

- Installing The Experimentation and Data Processing Suite
    * Windows 2000/XP/Vista Compatible

- Uninstalling The Experimentation and Data Processing Suite
    * Windows 2000/XP/Vista Compatible


Minimum System Requirements
=================================

233 MHz Intel Pentium class or better processor
At least 128 MB of RAM
Windows XP Service Pack 2 or Vista
Please note: A more powerful computer will deliver better performance.


Installing The Experimentation and Data Processing Suite
=================================================================

1.  Launch the Setup program located in the folder Experimentation and Data Processing
Suite

2.  A welcome screen will pop up, providing the user with the option of continuing with the
setup or exiting.
   To proceed further with the installation, click "Next". Click "Exit" to stop the program
from installing.
   Press "Esc" to cancel the installation of the software at any point of time.

3.  An information screen will pop up, describing the hardware required to interact with the
software for data processing.
```

It will also inform the user of the software required to run various experiments.

4. After the information screen, the directory screen prompts the user for a location to install the software.
The default location for this is "C:\Program Files\Experimentation and Data Processing Suite". The software requires
21 Mb of free hard disk space.

5. Click "Next" to proceed with the installation.

6. After clicking the "Next" button, wait approximately 2 minutes for the software to install.

7. Click "Finish" to complete the software installation.

8. The software will be in the Startup menu in the folder, "ADiR Software Suite".

9. To run a program, navigate to the "ADiR Software Suite" window and click on program you desire to run.

Uninstalling The Experimentation and Data Processing Suite
==

1. From the Windows Start Menu, select Settings | Control Panel.

2. Double click on the "Add/Remove Hardware" icon in the Control Panel. NOTE: this is called "Uninstall a Program" in Vista.

3. Select "Next" to begin the wizard.

4. Select "ADiR Experimentation and Data Processing Suite" and click "Uninstall."

5. The program will be removed from the computer.