

**Carbon Nanotube Based Vacuum Diode Characteristics at Elevated Temperatures**

by

Michael Moxley

A thesis submitted to the Graduate Faculty of  
Auburn University  
in partial fulfillment of the  
requirements for the Degree of  
Master of Science

Auburn, Alabama  
August 6, 2016

Keywords: Carbon nanotubes, CNT, Field Emission, Elevated Temperature

Approved by

Hulya Kirkici, Chair, Professor of Electrical and Computer Engineering  
Stuart Wentworth, Associate Professor of Electrical and Computer Engineering  
Mark Adams, Assistant Professor of Electrical and Computer Engineering

## Abstract

Since the discovery of carbon nanotubes twenty years ago, they have been studied heavily. Carbon nanotubes are known to have excellent mechanical, electrical, and thermal conductivity properties as well as for having a very large aspect ratio. Researchers are finding new ways to use carbon nanotubes in our everyday world. In this thesis, carbon nanotubes are put into a vacuum diode configuration and tested at different temperatures to see how the field emissions would change with temperature. The carbon nanotubes that were used in these experiments were randomly aligned Multi Walled Carbon Nanotubes (MWCNTs). These carbon nanotubes were grown in house using a chemical vapor deposition process on a silicon wafer.

## Acknowledgements

This thesis would not have been possible without Dr. Kirkici giving me the opportunity to do this research. I appreciate all the things I have learned from working as one of her students. I also want to thank all the professors who helped me in this process.

I would also like to thank Baha Yakupoglu and Michael Jung for teaching me how to use the machines to grow and test carbon nanotube samples.

## Table of Contents

Abstract .....	ii
Acknowledgements .....	iii
List of Figures .....	vi
List of Tables .....	viii
Chapter 1 Introduction .....	1
1.1 Carbon .....	1
1.2 Carbon Nanotubes .....	1
1.3 Synthesis of Carbon Nanotubes .....	3
1.4 Carbon Nanotube Applications .....	4
Chapter 2 .....	5
2.1 Vacuum Tube Diode .....	5
2.2 Thermionic Emission Theory .....	6
2.3 Field Emission Theory .....	6
2.4 Carbon Nanotube Diode .....	7
Chapter 3 Fabrication .....	9
3.1 Preparing Wafer .....	9
3.2 Photolithography .....	10
3.3 Sputtering .....	14
3.4 Chemical Vapor Deposition .....	16
Chapter 4 Field Emission Characteristics .....	22
4.1 Vacuum Chamber .....	22
4.2 Electrical Connections and Data Acquisition .....	23
4.3 Diode Configuration and Testing .....	25
4.4 Field Emission Testing Results .....	28
4.5 Thermionic Emission Contribution .....	36
4.6 High Temperature Bake and Field Emission Test .....	37
Chapter 5 Carbon Nanotube Patch Antenna .....	40
5.1 Design .....	40
5.2 Fabrication .....	42
5.3 Results .....	45

Chapter 6 Conclusion .....	48
References .....	49
Appendix .....	52

## List of Figures

Figure 1.1	Single walled and multiwalled carbon nanotube structure. . . . .	2
Figure 2.1	Circuit schematic of a vacuum tube . . . . .	6
Figure 2.2	Carbon nanotube diode setup . . . . .	8
Figure 3.1	Silicon wafer and wafer packaging label . . . . .	9
Figure 3.2	Spin rinse dryer machine and wafer holder . . . . .	10
Figure 3.3	The bottle of HMDS and the chamber used for the process . . . . .	11
Figure 3.4	The spinner and the hot plate to apply the photoresist. . . . .	12
Figure 3.5	Mask aligner to expose the photoresist to UV light. . . . .	13
Figure 3.6	Station to develop the photoresist . . . . .	14
Figure 3.7	Illustration of the sputtering chamber . . . . .	15
Figure 3.8	Chemical vapor deposition chamber . . . . .	17
Figure 3.9	SEM Image of the edge of the 5 minutes sputter iron CNT sample at 5k . . . . .	18
Figure 3.10	SEM Image of 5 minutes sputter iron CNTs at 10k. . . . .	18
Figure 3.11	SEM Image of 5 minutes sputter iron CNTs at 30k . . . . .	19
Figure 3.12	SEM Image of the edge of iron and carbon sputtered CNTs at 10k . . . . .	19
Figure 3.13	SEM Image of iron and carbon sputtered CNTs at 10k. . . . .	20
Figure 3.14	SEM Image of iron and carbon sputtered CNTs at 30k . . . . .	20
Figure 3.15	SEM Image of iron and carbon sputtered CNTs at 70k . . . . .	21
Figure 4.1	Image of a turbo pump and ion gauge. . . . .	22
Figure 4.2	Image of picoammeter and power supply . . . . .	24
Figure 4.3	The MATLAB field emission test GUI . . . . .	25
Figure 4.4	Carbon nanotube diode setup in a vacuum chamber . . . . .	26
Figure 4.5	Equivalent Circuit of carbon nanotube diode setup . . . . .	26

Figure 4.6	The CNT diode setup in the vacuum chamber with the heating element . . . . .	27
Figure 4.7	Voltage vs. Current plot from a CNT sample . . . . .	28
Figure 4.8	The field emission data in a plot of electric field verse current density . . . . .	29
Figure 4.9	Field emission data from iron sputtered CNT sample . . . . .	30
Figure 4.10	Fowler-Nordheim plot of iron sputtered CNT sample . . . . .	31
Figure 4.11	Field emission data from iron and carbon sputtered CNT sample 1 . . . . .	32
Figure 4.12	Fowler-Nordheim plot of iron and carbon sputtered CNT sample 1 . . . . .	32
Figure 4.13	Field emission data from iron and carbon sputtered CNT sample 2 . . . . .	33
Figure 4.14	Fowler-Nordheim plot of iron and carbon sputtered CNT sample 2. . . . .	34
Figure 4.15	CNT diode setup with CNT as anode . . . . .	35
Figure 4.16	Field emission data from CNT diode with CNT anode . . . . .	35
Figure 4.17	Fowler-Nordheim plot of CNT diode with CNT anode . . . . .	36
Figure 4.18	Field emission data comparing data from before and after an 850°C bake . . . . .	38
Figure 4.19	SEM image of CNTs at 5k after 850°C Bake . . . . .	38
Figure 4.20	SEM image of CNTs at 30k after 850°C Bake . . . . .	39
Figure 5.1	Rectangular Patch Antenna . . . . .	41
Figure 5.2	Ceramic Substrate with kapton tape mask . . . . .	43
Figure 5.3	2.4 GHz carbon nanotube patch antenna . . . . .	44
Figure 5.4	5 GHz carbon nanotube patch antenna . . . . .	45
Figure 5.5	Network analyzer used for S11 measurements . . . . .	46
Figure 5.6	S11 measurement for 2.4 GHz patch antenna . . . . .	46
Figure 5.7	S11 measurement for 5 GHz patch antenna. . . . .	47

## List of Tables

Table 4.1	Turn on Voltage and F-N Plot of Figures 9 and 10 . . . . .	31
Table 4.2	Turn on Voltage and F-N Plot of Figures 11 and 12 . . . . .	33
Table 4.3	Turn on Voltage and F-N Plot of Figures 13 and 14 . . . . .	34
Table 4.4	Turn on Voltage and F-N Plot of Figures 16 and 17 . . . . .	36
Table 4.5	Current Density Values from Thermionic Emission . . . . .	36
Table 5.1	Lengths of the Two Different Patch Antennas . . . . .	41



## **CHAPTER 1**

### **Introduction**

#### **1.1 Carbon:**

Carbon is the sixth element found on the periodic table and located on column 4. This means that carbon atoms have four valence electrons in its outer shell to form covalent bonds with itself, as well as other types of bonds with other elements. The diversity of bonds and their corresponding geometries enable the existence of structural isomers, geometric isomers, and enantiomers [1]. They are found in large, complex, and diverse structures and allow for endless variety of organic molecules [1]. Carbon is the fourth most abundant element in the solar system, after hydrogen, helium, and oxygen. Carbon is utilized in the development of the highest strength fibers, considered to be one of the best lubricants (graphite), known as the strongest crystal and hardest material (diamond). Essentially as a non-crystalline products (vitreous carbon), it is one of the best gas absorbers (activated charcoal) and one of the best helium gas barriers (vitreous carbon) [2].

There are many different ways carbon atoms can be bonded together, but the three most common forms are graphite, diamond, and the amorphous carbon that is the focus of this thesis. Carbon structures called fullerene are carbon atoms that form hollow structures like spheres or tubes. These hollow spheres are called buckyballs and the hollow tubes are called carbon nanotubes. The carbon atoms are joined together in hexagonal rings to form a graphene sheet.

#### **1.2 Carbon Nanotubes (CNTs):**

Carbon nanotubes are tubes of carbon in the form of graphite sheets that are rolled up. Nanotubes are perfectly straight tubules with diameters of nanometer size, and properties close to that of an ideal graphite fiber. Carbon nanotubes were discovered accidentally by Sumio Iijuma in 1991, while studying the surfaces of graphite electrodes used in electrical arc discharge [3]. In

general, carbon nanotubes can come in three different forms: Single Walled Carbon Nanotubes (SWCNTs), Double Walled Carbon Nanotubes (DWCNTs), and Multi Walled Carbon Nanotubes (MWCNTs). Single wall carbon nanotubes consist of a single layer of graphite that is rolled up. These carbon nanotubes have a diameter of only a few nanometers. Double wall carbon nanotubes consist of two layers of carbon that are rolled up to make a tube structure. The multiwall carbon nanotube consists of many layers of carbon. Each of the three different kinds of carbon nanotubes has different properties and is made using a different method.

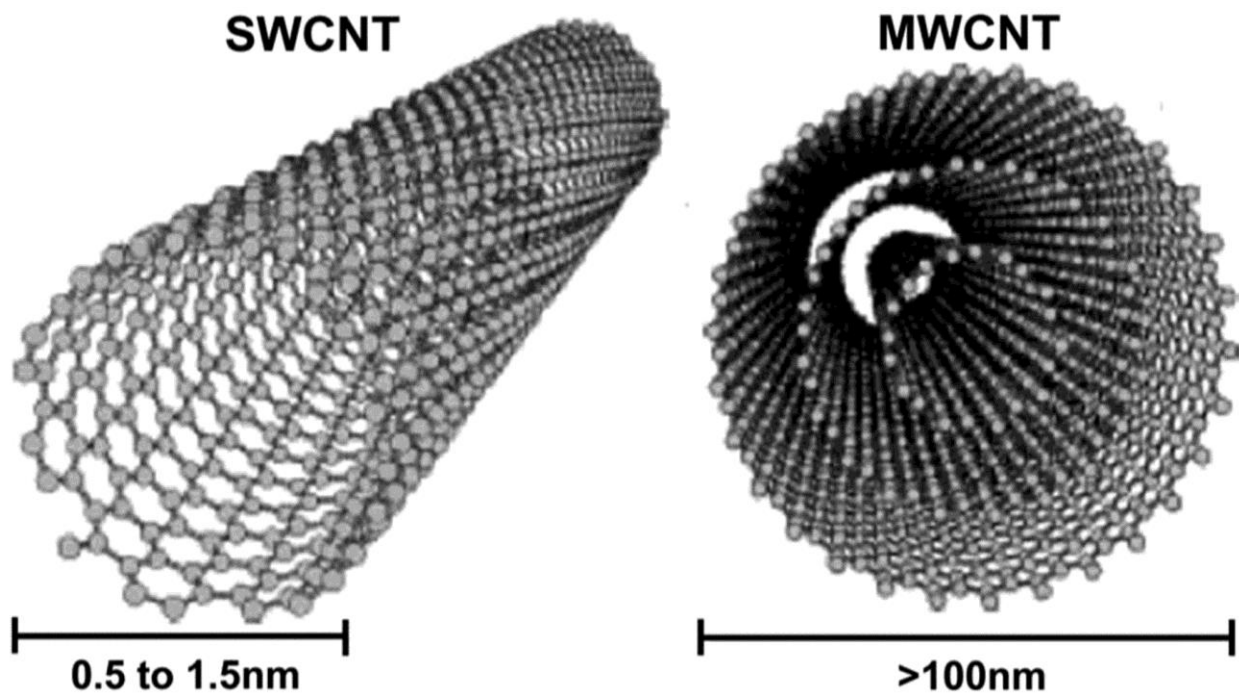


Figure 1.1 (Left) Single wall carbon nanotube structure (Right) The structure of a multi walled carbon nanotube structure

### **1.3 Synthesis of Carbon Nanotubes:**

There are several different ways to synthesis carbon nanotubes. The three most common ways are electric-arc discharge, laser vaporization, and chemical vapor deposition (CVD). The carbon nanotubes that were used in the experiments of this thesis involved synthesis using chemical vapor deposition.

The first carbon nanotubes were synthesized using the electric-arc discharge method. The electric-arc creates carbon nanotubes by striking an arc between graphite electrodes in an inert atmosphere (argon or helium). This process also produces carbon soot containing fullerene molecules [4]. The optimal yield of the nanotubes can be obtained under a helium atmosphere of 500 Torr, and a current roughly 100A [5].

In laser vaporization, a piece of graphite target is vaporized by laser irradiation under high temperature in an inert atmosphere. The quality of the carbon nanotubes depends on the reaction temperature with the best quality around 1,200°C [4]. The laser's vaporizing pulse lasts only ~10ns and carbon nanotube growth then can occur undisturbed from further excitation, even for single laser ablation events [6].

For chemical vapor deposition, a carbon containing gas (hydrocarbon source) is heated in a quartz tube that is enclosed in a furnace. The thermal energy initiates a chemical reaction and breaks up the carbon containing molecules. The freed carbon is then deposited on the substrate placed in the quartz tube. The substrate material can be Si, mica, silica quartz, or alumina. The nature and yield of the deposit obtained in the reaction is controlled by varying different parameters such as the nature of the catalytic metals and their supports, the hydrocarbon sources, the gas flow rate of feed gases, the reaction temperature, the reaction time, and others [4].

#### **1.4 Carbon Nanotube Application:**

Since the discovery of carbon nanotubes, we have learned that they have valued mechanical, physical, and electrical properties. Over the past two decades, researchers have discovered and adapted carbon nanotubes with many of the electrical devices we use today. Semiconducting carbon nanotubes have been proposed for such nanoelectronics applications as high-speed field-effect transistors (FETs), few or single-electron memories, and chemical/biochemical sensors [7]. With carbon nanotubes being both metallic and a semiconducting, an all-nanotube electronic device can be envisioned [1].

## CHAPTER 2

### **2.1 Vacuum Tube Diode:**

John Ambrose Fleming invented the first vacuum diode in 1904, and held the first patent for a vacuum type device. His device was a metal plate that was introduced in an incandescent lamp and the diode was used for detection in wireless communication [8]. This device utilized the thermionic emission phenomenon. A vacuum tube diode consists of a metal cathode and anode that are in a glass bulb. Inside the glass bulb is a vacuum. If a cathode is heated (thermionic emission), it is found that electrons from the cathode become increasingly active. As the temperature increases, they can actually leave the cathode and enter the surrounding space [9]. A voltage source is applied across the cathode and the anode. When the voltage is increased to a certain amount, the electrons that left the cathode are attracted to the anode. If there are gas molecules present in the space between the cathode and the anode where the electrons will travel, collisions will occur impeding the flow of electrons [9]. Vacuum tubes that was a breakthrough technology and used throughout the first half of the 1900s.

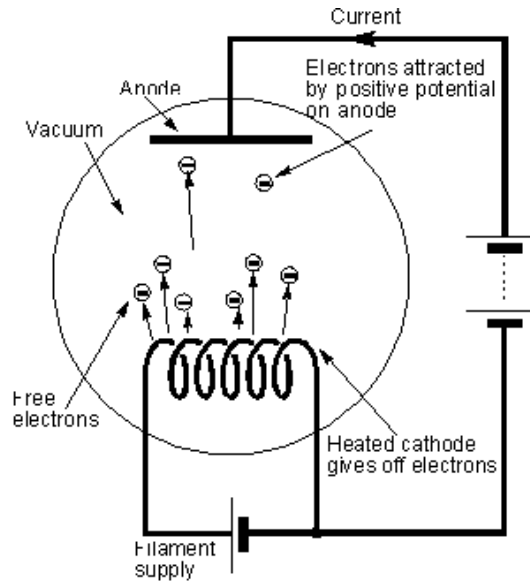


Figure 2.1 Circuit schematic of a vacuum tube [10]

## **2.2 Thermionic Emission Theory:**

The current emitted from a metal increases with the temperature of the metal and the applied field strength [24]. Millikan and his associates [12] have shown that as the higher temperatures are approached, at which ordinary thermionic emission begins, the strong field emission does become sensitive to temperature and finally blends into the thermionic. The equation for the current density of the thermionic emission can be represented as [25]:

$$J = AT^2 e^{-W/kT}$$

In that equation, A, which is the Richardson constant, is equal to  $1.2 \times 10^6 \text{ A/m}^2\text{K}^2$ , T is temperature in kelvin, W is the work function, and k is the Boltzmann constant.

## **2.3 Field Emission Theory:**

The main features of the phenomenon (field emission) of the extraction of electrons from cold metals by intense electric fields are well known. An approximate theory of the effect was first developed by Schottky [12]. The emission mechanism has been well described by wave mechanics: free electrons “tunnel” through the surface potential barrier of the metal when the latter

is altered by the applied field [13]. Normally at field strengths of the order of  $10^8$  V/m (below the electrical breakdown), the potential barriers at the surfaces of metals and semiconductors usually become very thin and result in field emissions of electrons due to the tunneling effect. [14]

The theory of cold cathode field emission of electrons from metals into vacuum is often described by Fowler-Nordheim (FN) theory, after the first authors to treat this effect as wave-mechanical tunneling through a triangular potential barrier [15]. Using a free-electron gas model to determine the tunneling probability expressed as a Taylor series expansion at the Fermi level, one obtains the following expression for the current density:

$$J = \frac{(q^3 E^2)}{(8\pi h \phi)} \exp \left[ \frac{-4(2m)^{1/2} \phi^{3/2}}{ehqE} \right]$$

Where  $h$  = Planck's constant,  $q$  = electronic charge,  $E$  = electric field,  $\phi$  = work function, and  $m$  = free electron mass [30]. In a Fowler-Nordheim plot ( $\log J/E^2$  vs  $1/E$ ), this equation is represented by a straight line.

## **2.4 Carbon Nanotube Diode:**

The structure of the carbon nanotube diode is very similar to the vacuum diode except that it uses carbon nanotubes as the cathode for the field emission. There is a metal plate for the anode, but instead of the metal cathode, carbon nanotubes are used. In the vacuum tube diode the cathode is heated for electron emission, but the carbon nanotubes do not have to be heated. The carbon nanotube diode is referred to as a cold cathode diode, since the cathode is not heated. The reason the carbon nanotubes do not need to be heated like the coil to start releasing electrons is because they are one of the best electron-emitting materials available. This is because high field enhancement is expected due to the high aspect ratio of the length to the diameter [11]. The field

emission properties of the carbon nanotubes are affected by the length of the CNTs and the spacing between them. It is important to characterize the effect of length and spacing on the field-emission properties in order to obtain the high and uniform field emission current at low electric field value [11].

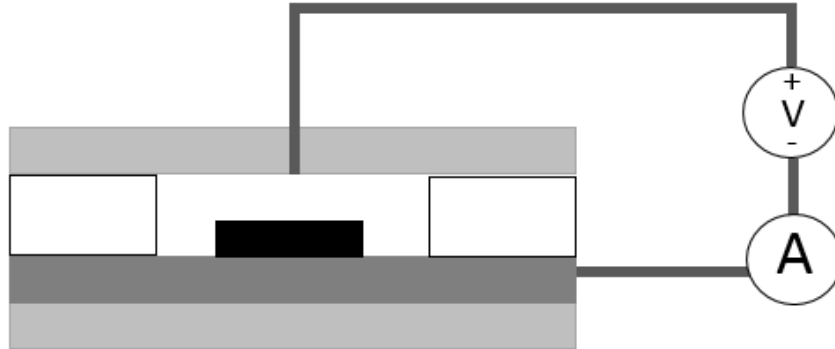


Figure 2.2 Carbon nanotube diode setup.

This diode setup was connected to a power source and an ammeter to read the current value. The power supply and ammeter were connected by GPIB cords to a computer. When a MATLAB program on the computer was run, the program controlled the power supply and recorded the data. Different voltage values were input into the program to sweep through and test the field emission of the carbon nanotubes from 0 to 1000 volts. For the experiments that were run, the carbon nanotube samples were grown on silicon substrates. Glass spacers were placed on each side of the carbon nanotube sample. The anode consisted of a ceramic block with a stainless steel rod in the middle of the block. This was placed directly above the carbon nanotubes with a metal contact on top for the electrical connection.



## CHAPTER 3

### Fabrication

#### 3.1 Preparing Wafer:

The first part in the fabrication of the carbon nanotubes is to clean and pattern the silicon wafer. These processes are done in a class ~100 clean room where a cleanroom suit or bunny suit is required to be worn.

The wafers that are used in the carbon nanotube fabrication are N-type, 100mm (4 inch), <100> silicon wafers. These silicon wafers were purchased from Silicon Quest. The wafers have a thickness between 475-575  $\mu\text{m}$ . The front side of these silicon wafers is polished with the back side etched. A picture of a wafer and the label from the wafers packaging can be seen in Figure 3.1.



Figure 3.1 (Left) N-type Silicon wafer. (Right) Wafer details from packaging.

Before the silicon wafer is ready for photolithography, it first is cleaned. The wafer is put into a wafer holder and then placed into a spin rinse dryer. The spin rinse dryer spins at a very high rate and directs water directly at the wafer. After a set amount of time, the water stops

shooting and the dryer continues to spin. After the spin rinse dryer, the wafer is placed into an oven at 120°C for a dehydration bake.

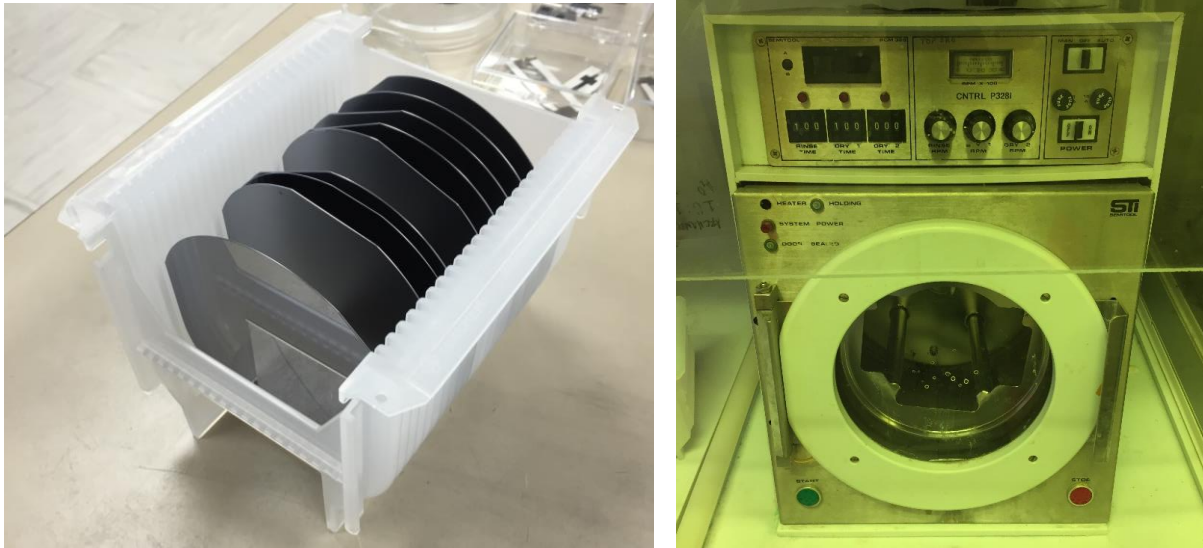


Figure 3.2 (Left) Wafer holder (Right) Spin rinse dryer machine.

### **3.2 Photolithography:**

The first step in the photolithography process is to apply HMDS (hexamethyldisilazane) by a vapor process. Applying HMDS to dehydrated wafers is one common method for achieving the surface hydrophobicity required to prevent photoresist delamination [16]. The silicon wafer is placed into a glass chamber and 5 drops of the HMDS are put into the chamber and the top is put on. The wafer sits in the chamber for 5 minutes.

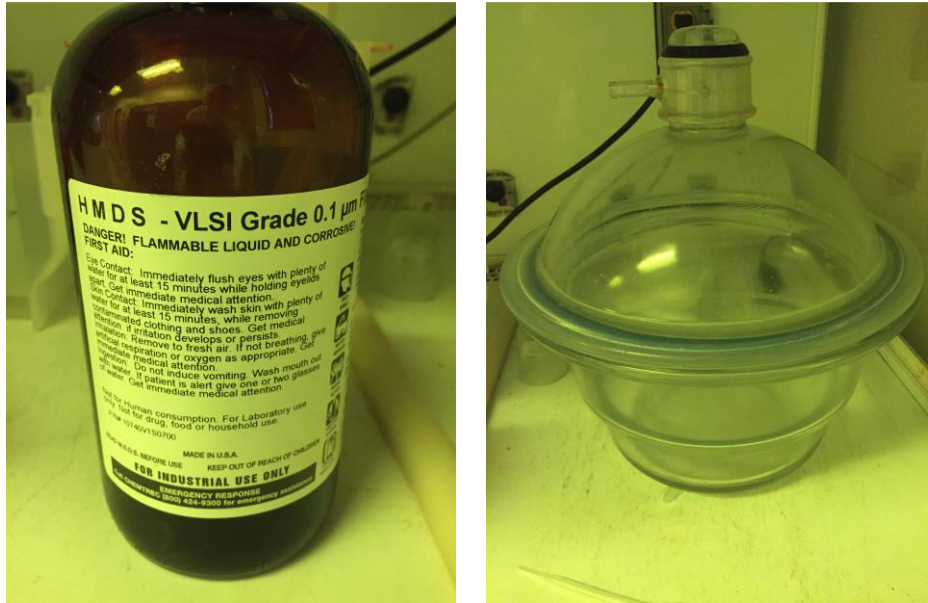


Figure 3.3 (Left) Bottle containing HMDS. (Right) Chamber for HMDS vapor process.

The next step is to apply photoresist on the wafer. The photoresist material is composed of three components: a photoactive compound, a base resin, and a suitable organic solvent system [17]. To apply the photoresist evenly on the wafer, the wafer is placed onto a spinning disc and a vacuum holds the wafer down. The spinner can be seen in Figure 3.4. The photoresist is poured into the center of the wafer. The wafer then spins at 3000 rpm for 30 seconds. This process will coat the wafer with a thin even layer of the photoresist. AZ P4620 was the photoresist that was used in this process. This photoresist is a positive photoresist. For a positive photoresist, the resist is exposed to UV light where the underlying material is to be removed. In these resists, exposure to UV light changes the chemical structure of the resist so that it becomes more soluble in the developer [18]. After the photoresist has been spun on, the wafer is placed on a heater for a soft-bake at 105°C for 1 minute. Soft-baking plays a critical role in the photo-imaging. The photoresist coatings become photosensitive, or imagable, only after softbaking [18].

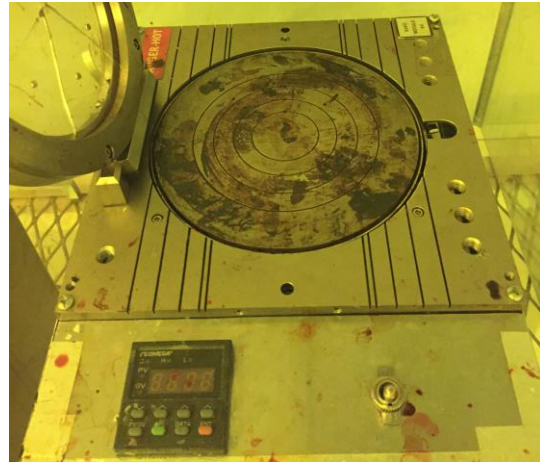


Figure 3.4 (Left) Spinner to apply photoresist. (Right) Heater for the soft bake

The next step in the photolithography process is to pattern the wafer. The wafer is placed into a machine called a mask aligner. This machine consists of a powerful UV light that is used to expose the photoresist. It also has a microscope, so multiple layers of masks can be lined up. In our process, we only used one mask. The mask can be seen in Figure 3.5. It consists of 5mm by 5mm squares. The UV light that goes through the square will expose the photoresist. For this process, the UV light was set to turn on for 30 seconds.

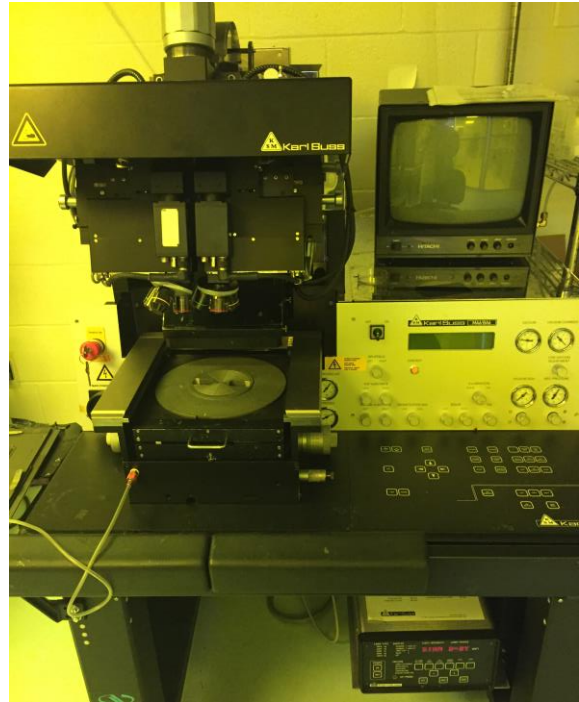
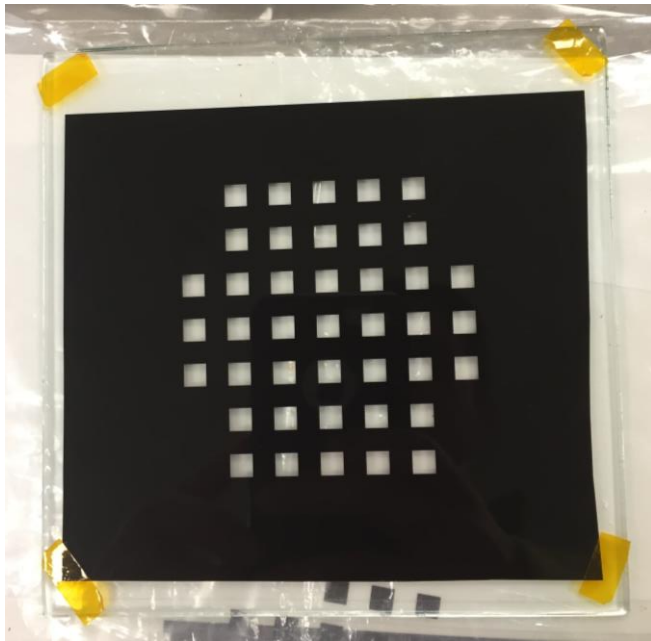


Figure 3.5 (Left) Photolithography Mask. (Right) Mask Aligner

The last part of the photolithography process is to develop the wafer. The developer that was used in this process was called AZ 400K Developer. This was mixed in with water at 2 parts water to 1 part developer. The wafer was placed into the developer and water mixture for approximately 45 seconds to 1 minute. The time the wafer was in the solution would be different for each run. The wafer is examined under the microscope to ensure that all the photoresist had washed away in the exposed areas. If some photoresist remains, the wafer is put back into the developer solution until it is gone.

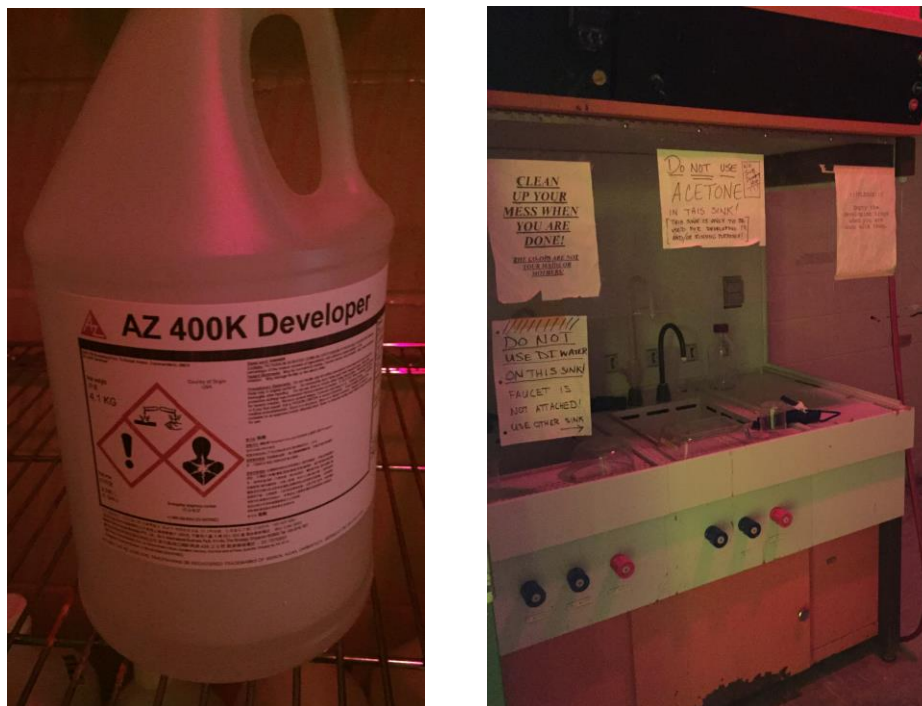


Figure 3.6 (Left) Developer Solution (Right) Station to develop the wafer

### **3.3 Sputtering:**

After the wafer has been patterned using the photolithography process, it is now ready for the catalyst metal layer to be sputtered. The catalyst layer is necessary for carbon nanotube growth. For our samples, iron was used as a catalyst. The catalyst layer was formed during the sputtering process. The sputtering process takes place in a vacuum chamber. A two inch disc of iron is then placed on the target area. The wafer is secured facing upside down, so it is directly in line with the iron target. The chamber is pumped down to around  $3-5 \times 10^{-6}$  Torr. Argon is pumped into the system and the chamber maintained a constant pressure of 70 mTorr. The sputtering process is described as follows [19]: a DC voltage is applied between the target and the substrate which ionizes argon atoms and creates a plasma, hot gas-like phase consisting of ions and electrons, in the chamber. This plasma is also known as a glow discharge. These ionized argon atoms are accelerated to the anode iron target. The collision with the target ejects iron atoms, which travel to the substrate and eventually settle on the wafer. Electrons released during argon ionization are

accelerated towards the anode substrate, subsequently colliding with the background argon atoms, creating more ions and free electrons in the process, continuing the cycle. An image of the sputtering process is show in Figure 3.7. The sputter process continues for 5 minutes. The deposited thin layer of iron only a few nanometers thick throughout this process.

The first tested samples are given a sputtering time of 5 minutes resulting in a few nm iron film. The other samples that are tested are sputtered with iron for 5 minutes and then sputtered with carbon for 8 minutes.

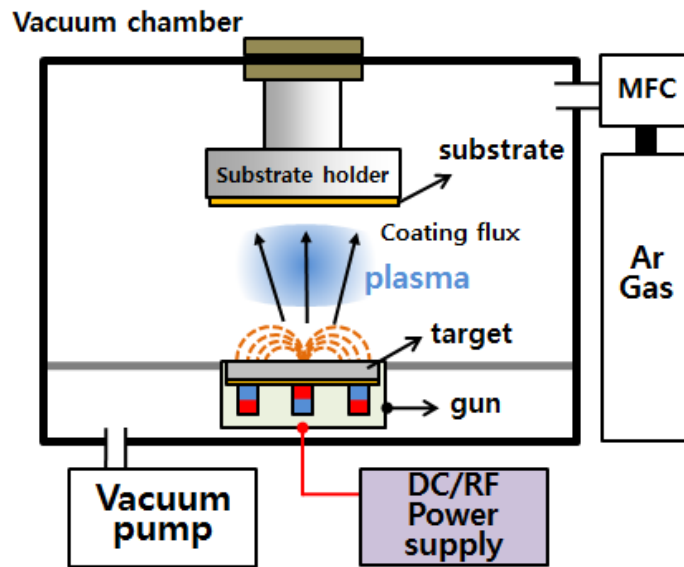


Figure 3.7 Illustration of the sputtering chamber.

After the sputtering process, the photoresist from the wafer needed to be completely removed. For this process, acetone and methanol were used to clean the photoresist from the wafer. The silicon wafer was put into a glass bowl and then acetone was poured into the bowl. The wafer was moved up and down in the acetone. This process removed most of the photoresist. This process is then repeated using methanol. The acetone and methanol baths are repeated a few times to ensure that all the photoresist is removed from the wafer. After the methanol bath, the

wafer was washed with deionized water. To ensure that all the photoresist was removed from the wafer, it was examined under a microscope.

The samples were then placed on a hot plate at 300°C for 4 hours for annealing. Annealing has two important functions. One is to eliminate oxidation of the metallic surface and the second is to form nucleation of metallic catalyst seeds to grow CNTs in the CVD furnace [20].

### **3.4 Chemical Vapor Deposition (CVD):**

The last process in growing the carbon nanotubes is for the wafer to undergo a chemical vapor deposition process. Chemical vapor deposition (CVD) is a widely used materials-processing technology. The majority of its applications involve applying solid thin-film coatings to surfaces, but it is also used to produce high-purity bulk materials and powders, as well as fabricating composite materials via infiltration techniques [21]. In our process, a mechanical roughing pump is connected to a quartz tube. A quartz plate is used to hold the silicon wafer in the quartz tube. This quartz plate is pushed to the center of the tube and kept there during the CVD experiment. The mechanical pump is turned on to pump down the tube. The quartz tube is heated up to 705°C. Using two gas flow controllers, one for argon and one for acetylene, with flow rates of 22ccm and 75ccm respectively, gas is introduced into the chamber. When the gases start flowing into the tube, the pressure begins to rise and then levels off at 70.8 Torr. This experiment is run for 40 minutes for each sample that was tested in this thesis. Varying the time of the CVD will vary how long the carbon nanotubes will grow on the substrate.



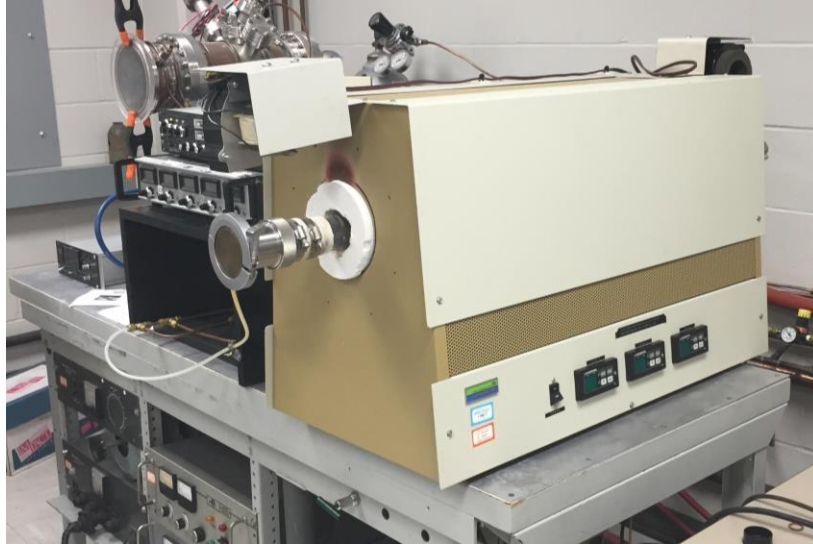


Figure 3.8 Chemical Vapor Deposition Chamber

The images below are the SEM pictures of the carbon nanotube samples grown using this method. The pictures taken are of two different carbon nanotube samples. The first sample was sputtered with iron for 5 minutes, annealed for 8 hours, and placed in the CVD furnace for 40 minutes. The second sample was sputtered with iron for 5 minutes and sputtered with carbon for 8 minutes, annealed for 8 hours, and placed in the furnace for 40 minutes. The pictures are of the edge of the carbon nanotube sample and images of the middle of the carbon nanotube sample. The second process, i.e. the carbon layer, enhances the carbon nanotube grown and results in growing vertically aligned carbon nanotubes [31]. However, in this case vertical carbon nanotubes did not grow. This can be attributed to not having the correct thickness of carbon over the iron catalyst.

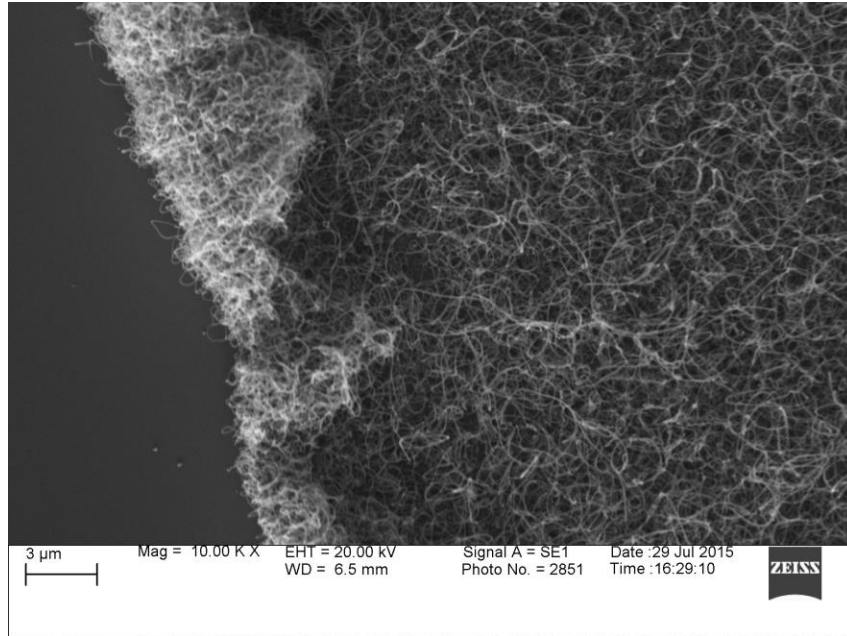


Figure 3.9 SEM Image of the edge of the 5 minute sputtered iron CNT sample at 5k

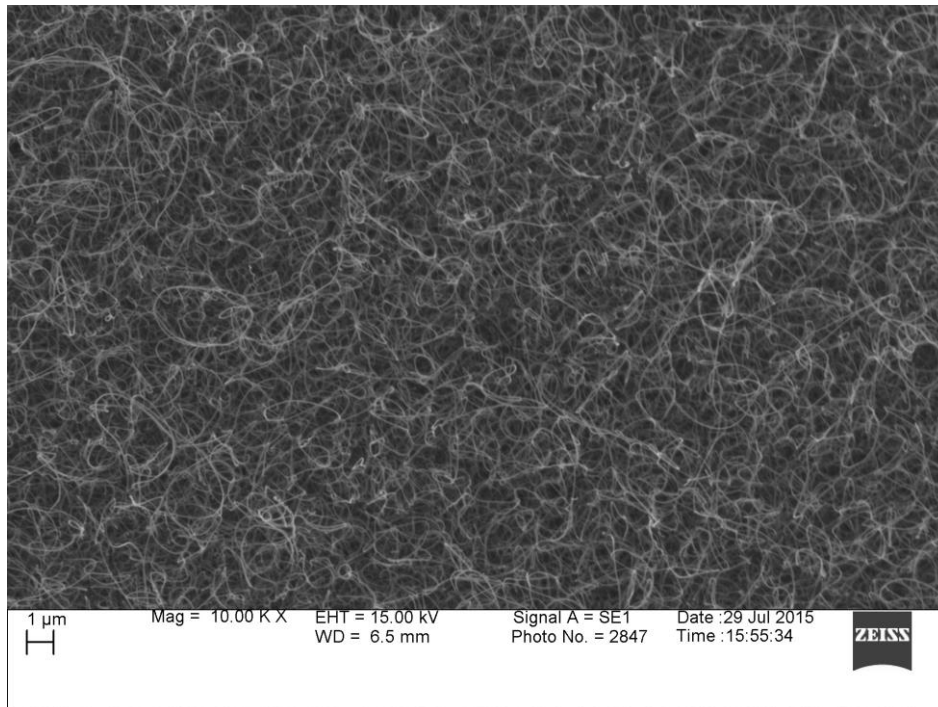


Figure 3.10 SEM Image of 5 minute sputtered iron CNTs at 10k

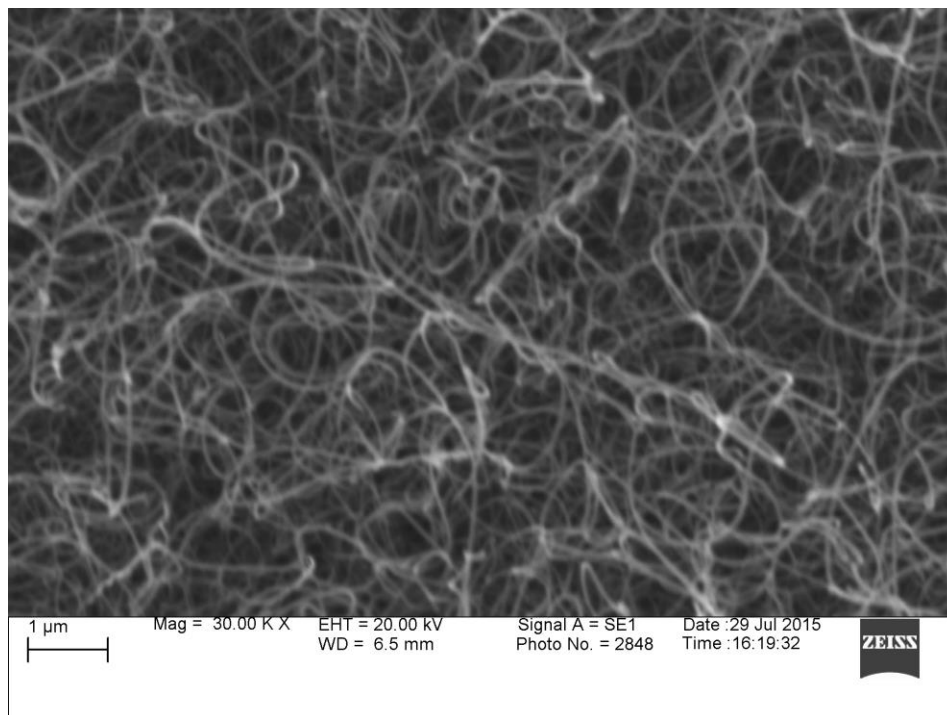


Figure 3.11 SEM Image of 5 minute sputtered iron CNTs at 30k

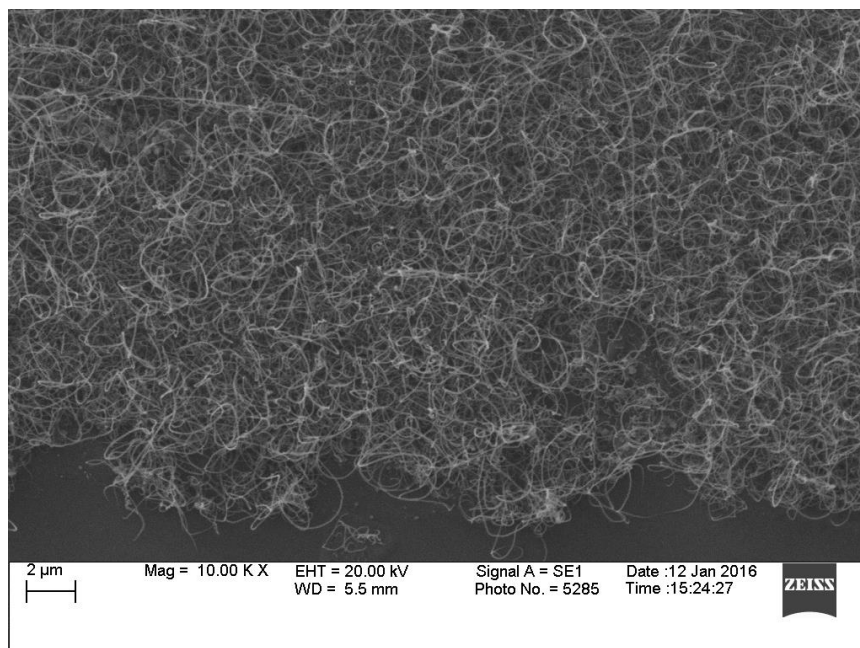


Figure 3.12 SEM Image of the edge of the 5 minute sputtered iron and 8 minute sputtered carbon CNTs at 10k

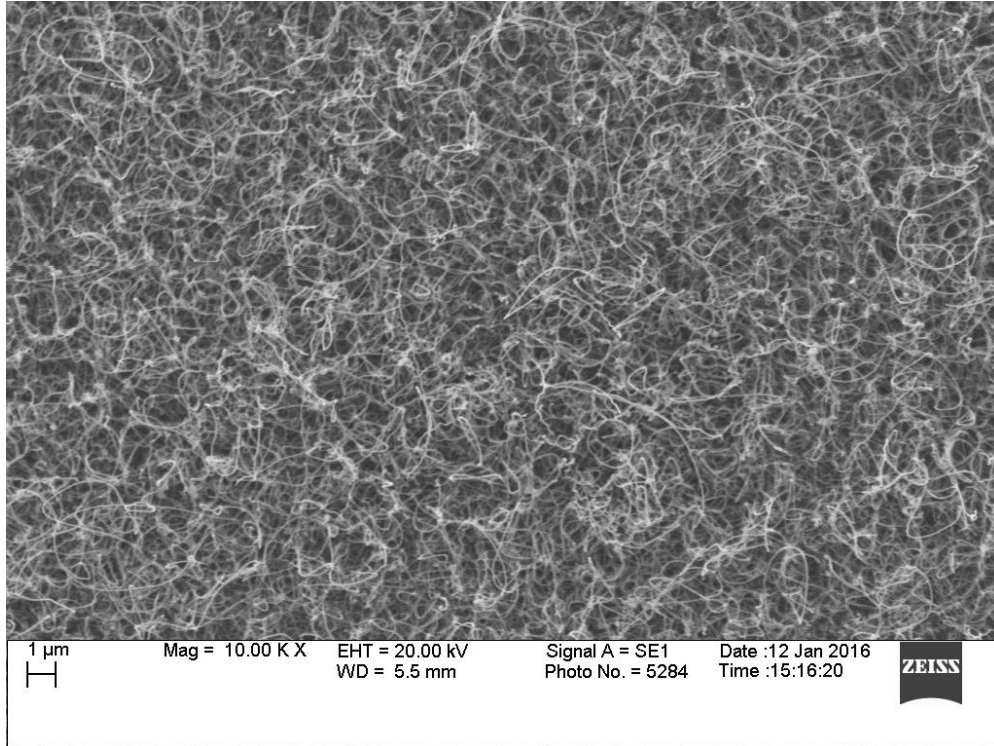


Figure 3.13 SEM Image of 5 minute sputtered iron and 8 minute sputtered carbon CNTs at 10k

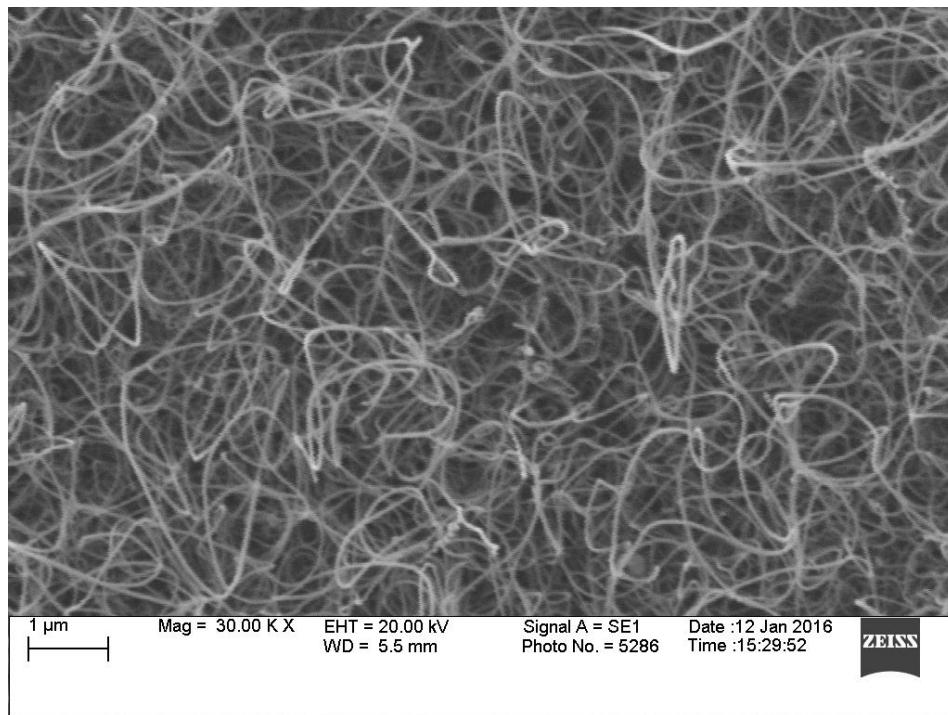


Figure 3.14 SEM Image of 5 minute sputtered iron and 8 minute sputtered carbon CNTs at 30k

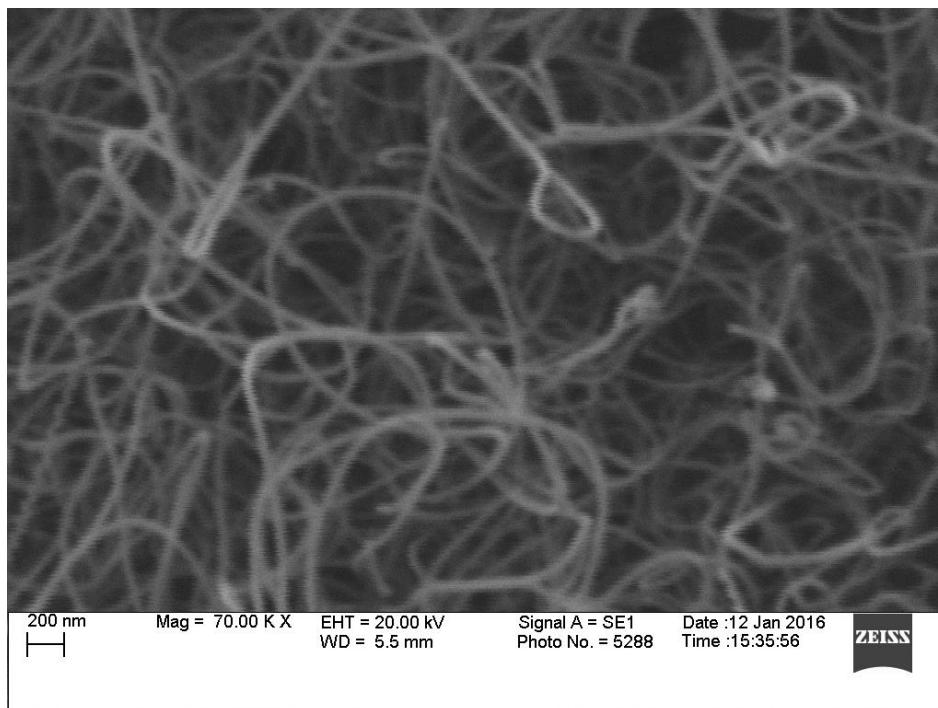


Figure 3.15 SEM Image of 5 minute sputtered iron and 8 minute sputtered carbon CNTs at 70k

## CHAPTER 4

### Field Emission Characteristics

#### 4.1 Vacuum Chamber:

Once the sample is ready to be tested for the field emission, it is placed in a vacuum chamber. All the tests are done in a vacuum pressure of  $3 \times 10^{-6}$  Torr.

These vacuum chambers use a mechanical pump to initially pump the chamber down to 10-70 mTorr, then the turbo pump can be turned on to further pump down the chamber to  $10^{-6}$  torr range. A turbo pump is made up of many layers of blades that are stacked on each other. Turbomolecular pumps work on the principle that gas molecules can be given momentum in a desired direction by repeated collision with a moving solid surface. In a turbomolecular pump, a rapidly spinning turbine rotor 'hits' gas molecules from an inlet of the pump towards the exhaust in order to create or maintain a vacuum [22].

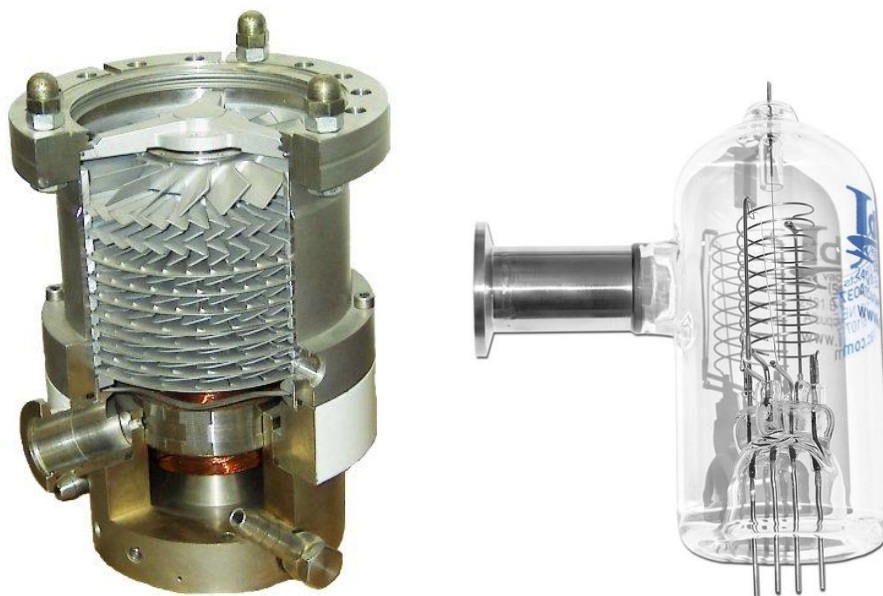


Figure 4.1 (Left) A turbo pump with the side cutout showing how the blades are laid out. (Right) An ion gauge used to measure pressure.

Figure 4.2 illustrates the layers of blades stacked on top of each other. The blades are very delicate and can break easily if not handled correctly. When turning off the turbo pump, the blades must come to rest before turning off the mechanical pump and venting the chamber. The ion gauge is used to measure the pressure after the turbo pump has been turned on. The ion gauge uses a filament that heats and releases electrons. The pressure that is indicated on the controller is based on the ionization of the gas molecules by a constant flow of electrons. The negative electrons are emitted at a controlled, selectable rate from a heated filament (cathode) and are accelerated toward a positively-charged wire grid (anode). As the electrons go from the cathode to the anode, they will collide with gas molecules and produce positive ions. At a constant filament-to-grid voltage and electron emission current, the rate at which positive ions are formed is directly proportional to the density of molecules (pressure) in the gauge for pressures below approximately  $1 \times 10^{-3}$  Torr [23].

#### **4.2 Electrical Connections and Data Acquisition:**

The vacuum chambers have electrical feedthrough connections to connect a power supply to the sample inside the vacuum chamber. The carbon nanotube diode is connected to a Stanford Research Systems, Inc Model PS325 power supply and a Keithley 485 autoranging picoammeter. The power supply and the picoammeter are connected to a computer through GPIB cables. The computer runs a MATLAB code that connects up to the power supply and controls the voltage output of the power supply. The circuit diagram of the setup is shown in Figure 4.7.



Figure 4.2 (Top) Picoammeter (Bottom) Power supply

The MATLAB code help to run the experiment and collect the data. The MATLAB code was written by Haitao Zhao, a former PhD student in our research group [32]. The power supply and the picoammeter have a GPIB port in the back, but a converter was used to connect it to the computer. A Prologix GPIB-USB Controller 6.0 was used to convert the GPIB to USB port. Address values were given to the power supply and picoammeter, so the computer could talk and control the devices. The power supply was set up as address 20 and the picoammeter as address 22. When the MATLAB code is run, a graphical user interface (GUI) pops up. In this GUI, there are nine boxes (as shown in Figure 4.3), to enter the voltage values to be tested. Each column can be set to increment the voltage values differently. In a normal run, the first column would be set to start at 0 Volts and end around 200 Volts with an increment of 50 Volts. The second column starts at 210 Volts and ends around 400-500 Volts with an increment of 10 Volts. The third column includes values starting at 550 Volts and going up to 800-1000 Volts and incrementing by 50 Volts. The reason behind this is that the values between 200-500 Volts are the most important, because this is the voltage range in which the carbon nanotube diode starts emitting electrons.



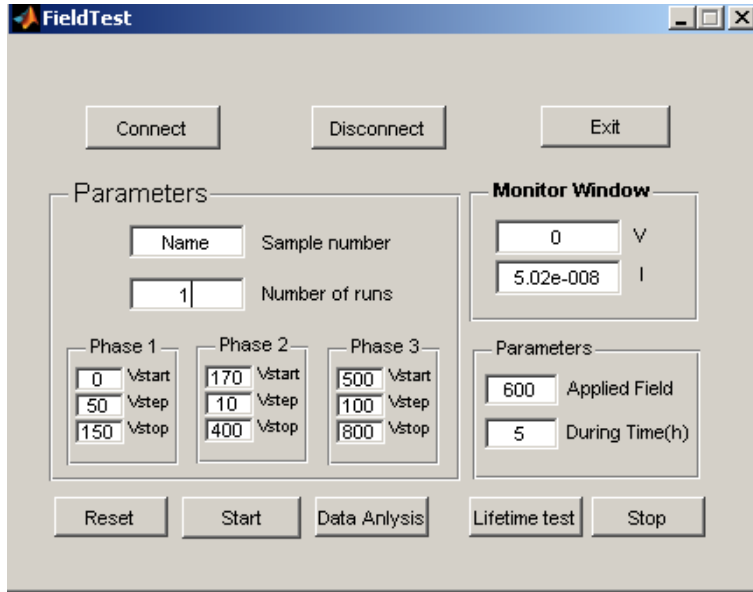


Figure 4.3 MATLAB Field Emission Test GUI

The first thing the code does is to set a variable equal to the serial port of the USB. Once the connect button is pressed, various GPIB commands are written to that variable containing the serial port for initial setup. When the start button is pressed, the voltage values that were entered into the GUI are saved in variables and the test begins to run. The different voltage values are sent to the power meter and the code reads and saves the current value from the picoammeter. This data is placed into an Excel file that can be used later to make plots and visually see the results. The MATLAB source code can be found below in the appendix.

#### **4.3 Diode Configuration and Testing:**

After the carbon nanotubes have been fabricated onto the silicon substrate, they are put into a diode configuration. The carbon nanotubes on the silicon represent the cathode part of the diode. The middle of the ceramic block was drilled out, so a stainless steel rod could be placed into the drilled hole. The drilled hole was precisely the diameter of the stainless steel rod, so that the rod would be a tight fit and would not fall out. The ceramic block and stainless steel rod were the anode of the diode. Glass spacers were placed down on the silicon for the ceramic block to be

placed on. The glass spacers have a thickness of  $140\mu\text{m}$ . The diode was then set up in a vacuum chamber as shown in Figure 4.4.

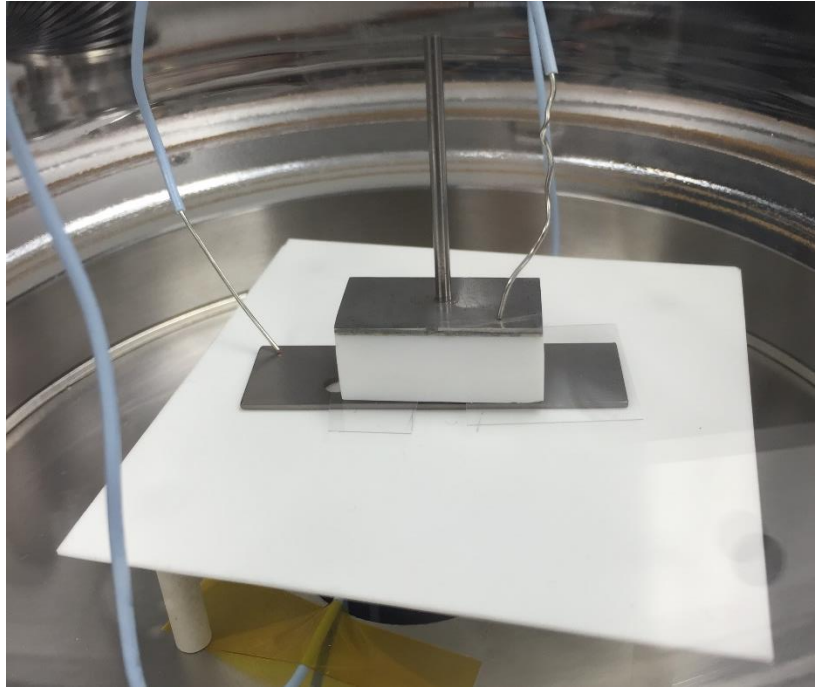


Figure 4.4 Carbon nanotube diode setup in a vacuum chamber

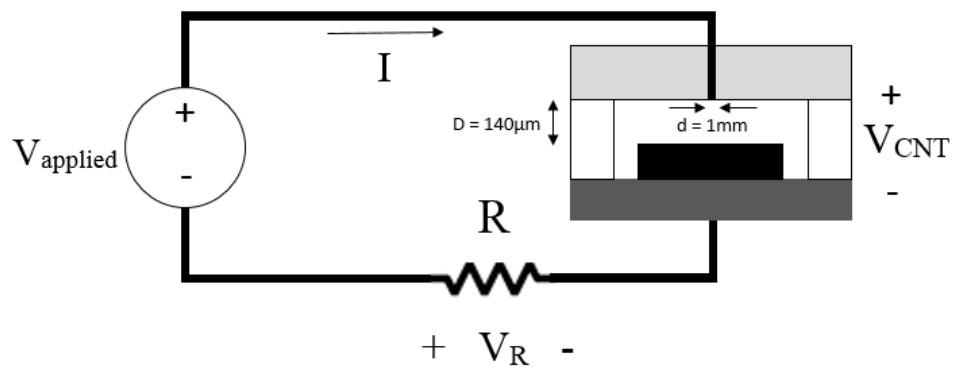


Figure 4.5 Equivalent Circuit

This carbon nanotube diode setup is in a small bell jar chamber. The leads that connect the diode to the power supply and current meter go straight down therefore requiring a small table.

The piece of silicon which is not visible in the picture is sitting on a piece of tungsten, so the connection could be made.

The carbon nanotube diode setup in Figure 4.1 is a vacuum chamber to test field emissions. A heater element was needed to be added into a vacuum chamber to run these experiments. For this reason, tungsten filament heater lamps were used. These tungsten filament bulbs were set up with a metal reflector, so that all the radiation would be targeted in one direction. For this heating element, three of these bulbs were connected in series. These bulbs were controlled by a controller that reads the temperature inside the vacuum chamber by a thermocouple. The thermocouple touches the black piece of ceramic the experiment is set up on. The desired temperature is set on the controller and it maintains that temperature.

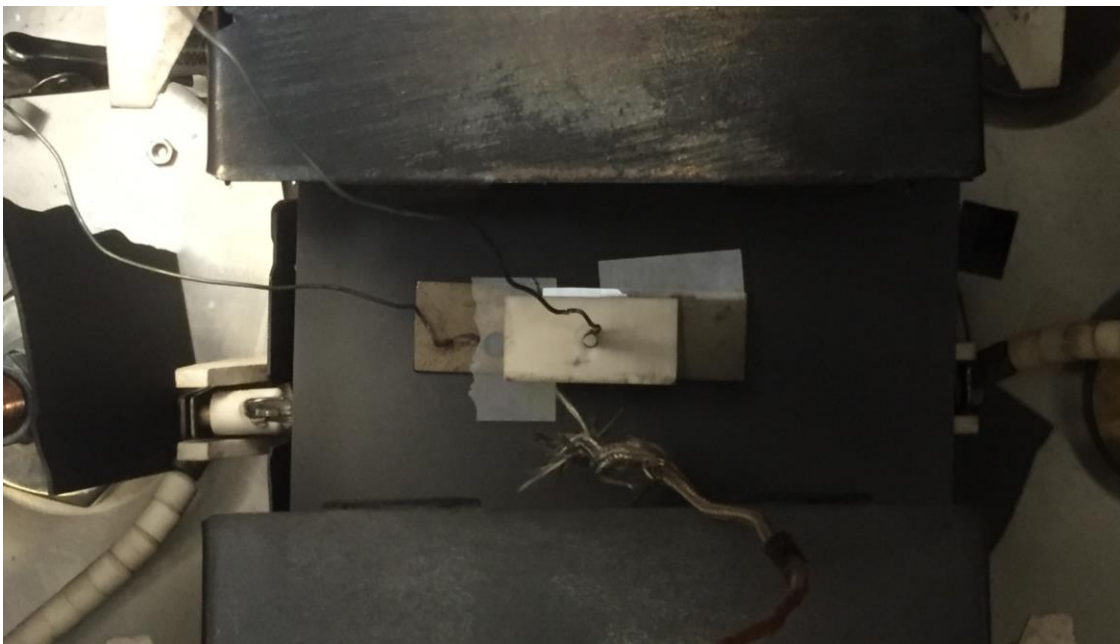


Figure 4.6 Carbon nanotube diode setup in vacuum chamber with heating element.

#### **4.4 Field Emission Testing Results:**

These tests were set up to see how the electron field emission from randomly aligned carbon nanotubes would change as the temperature increased from room temperature. As stated before, these field emission tests were performed in a very high vacuum ( $3-5 \times 10^{-6}$  Torr). When the tests are run, the MATLAB program saves the current values read at the picoammeter with the voltage values sent to the power supply. Excel was used to plot the data using a line scatter plot.

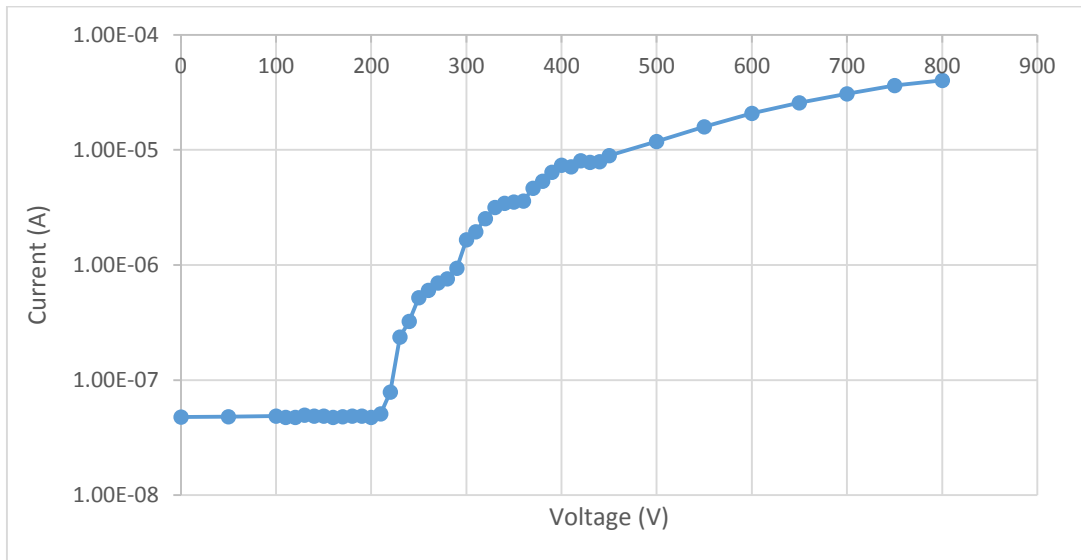


Figure 4.7 Voltage vs Current plot

For a better representation of this data, electric field and current density are used for the plot. The electric field (E) is found by subtracting the voltage of the power supply by the current multiplied by the resistor and then that value divided by the gap distance between the carbon nanotubes and the anode. The current density (J) is found by dividing the current by the surface area of the metal anode that is exposed to the carbon nanotubes. An example of this data can be seen in Figure 4.8.

$$E = \frac{V_{CNT}}{d_{gap}} = \frac{V_{Applied} - V_R}{d_{gap}} = \frac{(Voltage - Current * Resistance)}{Gap Distance (\mu m)} \left[ \frac{V}{\mu m} \right]$$

$$J = \frac{\text{Current}}{(\text{Surface Area of Anode})} \left[ \frac{\text{A}}{\text{mm}^2} \right]$$

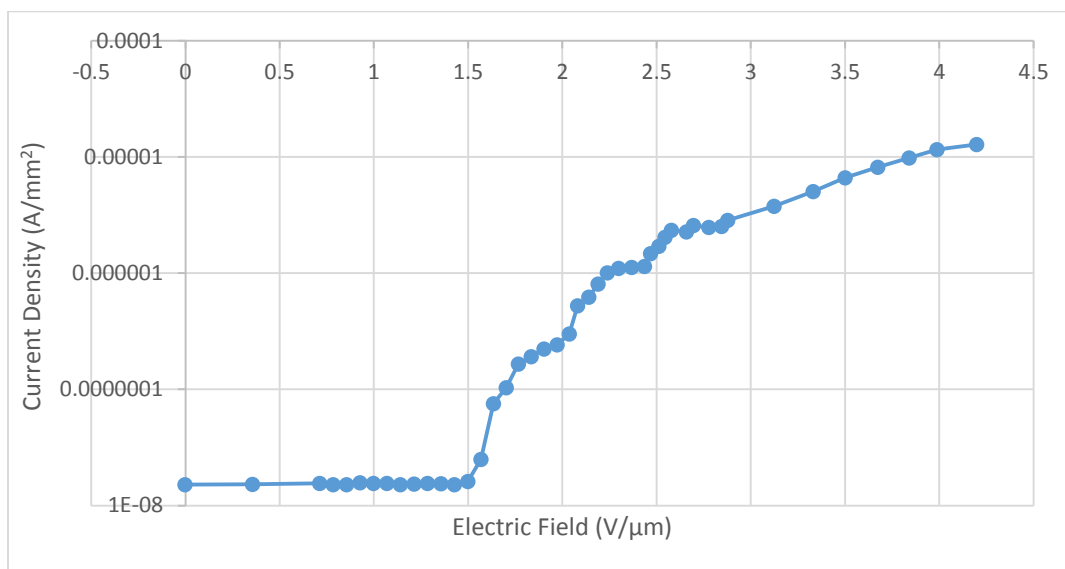


Figure 4.8 Field Emission Data, Electric Field vs Current Density

For the field emissions tests that were performed at different temperatures, four different types of carbon nanotube samples were used. In all of the samples presented in the thesis, as the temperature of the carbon nanotube diode increased, the turn on point would decrease.

The first carbon nanotube sample was made using a 5 minute sputtering, 8 hour anneal, and 40 minute chemical vapor deposition process. This sample was tested from room temperature to 300°C. For the data below in Figure 4.9, the data might not show the decreasing turn on voltage decreasing with the increase of temperature linearly, but the turn on voltage of the carbon nanotube diode is just as good or better than that of the carbon nanotube diode at room temperature.

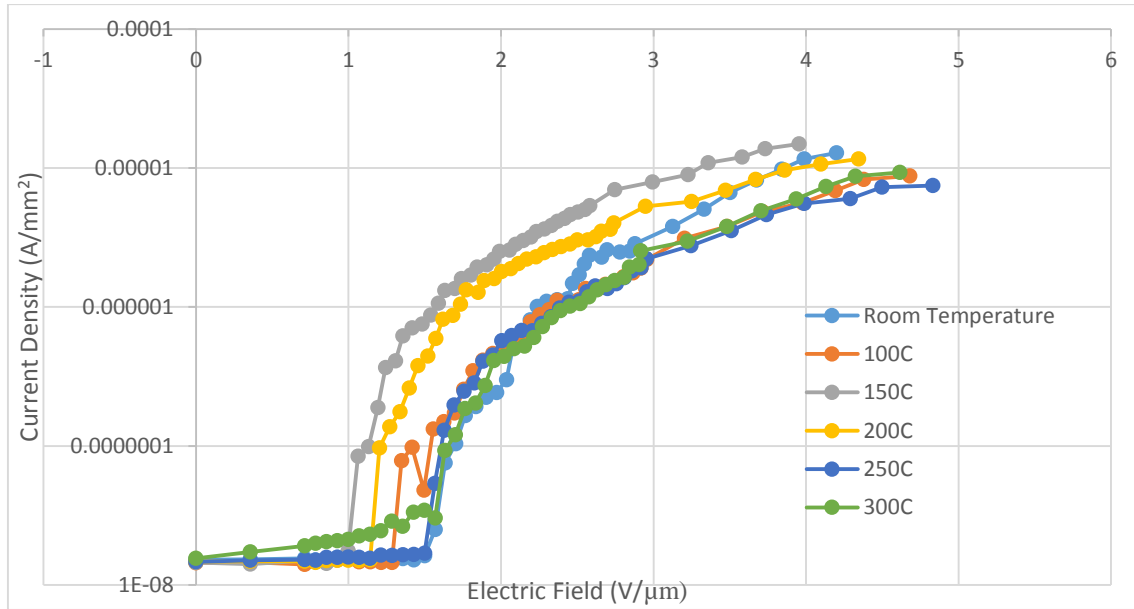


Figure 4.9 Field Emission Data from Fe sputtered CNT sample

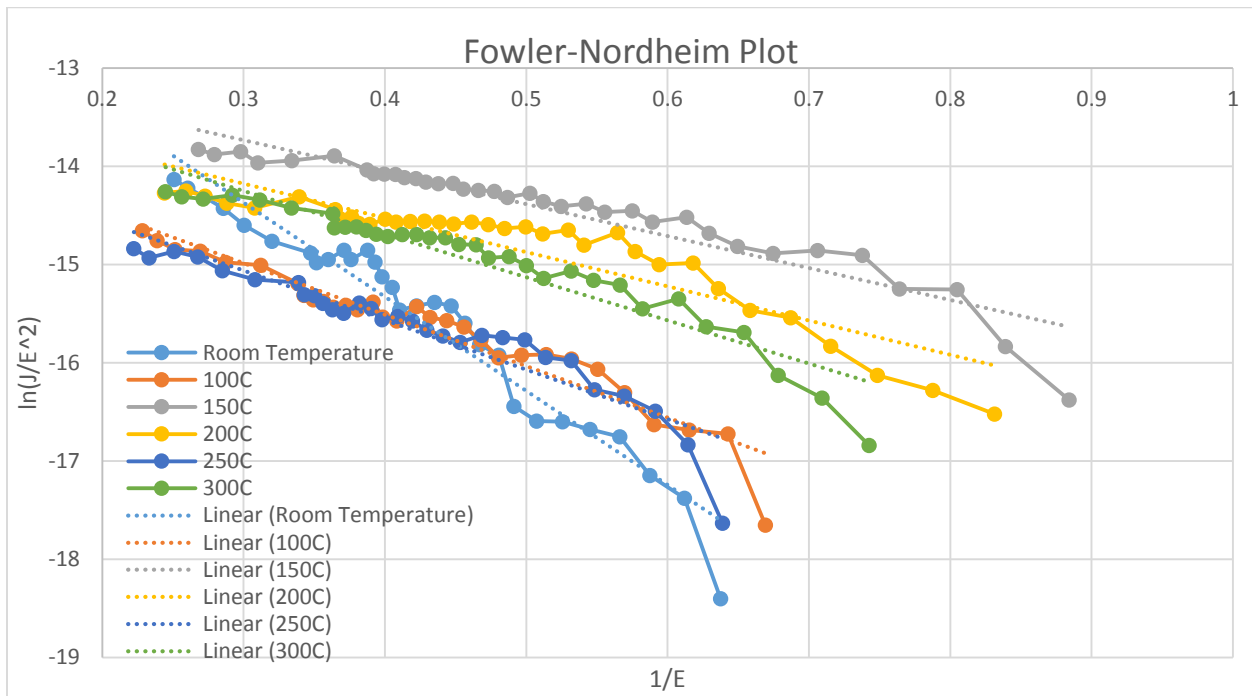


Figure 4.10 Fowler-Nordheim Plot of Field Emission Data in Figure 4.9

Temperature	$V_{\text{turn on}}$	Slope
Room Temperature	1.42 V	-9.5868
100C	1.35 V	-5.2408
150C	1.07 V	-3.2525
200C	1.21 V	-3.4824
250C	1.49 V	-5.0463
300C	1.56 V	-4.391

Table 4.1 Turn on voltage and slope of F-N Plot of Figures 4.9 and 4.10

For the next carbon nanotube sample, it was made using a 5 minute sputtering iron then an 8 minute sputtering carbon, 8 hour anneal, and 40 minute chemical vapor deposition process. This recipe has been used in the past to grow vertically aligned carbon nanotubes. Unfortunately, the carbon nanotubes did not grow in a vertically aligned orientation, but were randomly aligned. Two different carbon nanotube samples from this batch were tested. The SEM images of these samples can be seen in Figures 3.12-15. Figure 4.11 and 4.13 show the field emission data from these samples and different temperatures. In both of the samples, the turn on voltage of the device decreases as the temperature increases.

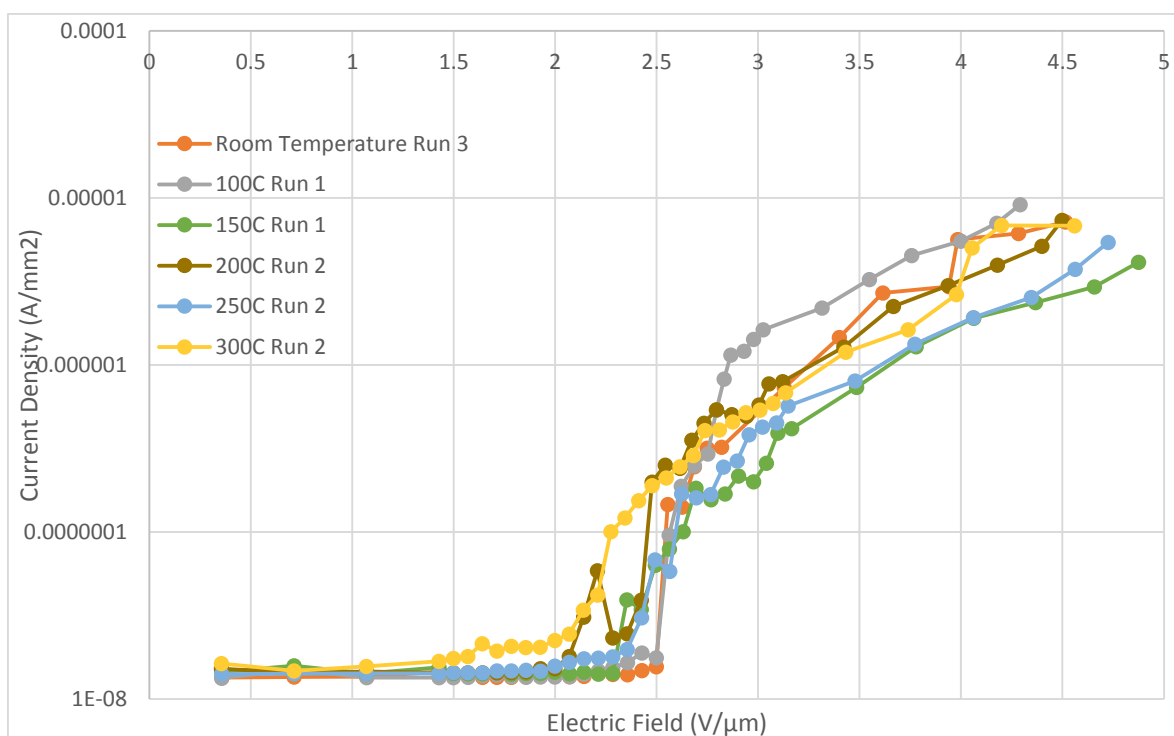


Figure 4.11 Field Emission Data for Sample 1 of Fe and C Sputtering

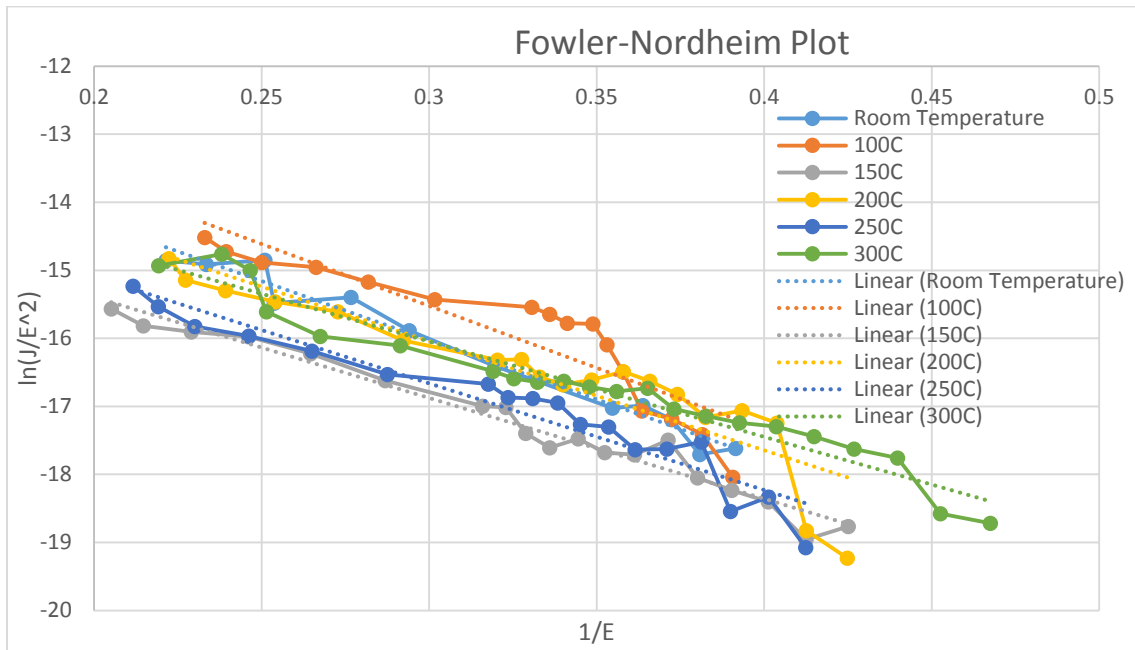


Figure 4.12 Fowler-Nordheim Plot of Field Emission Data in Figure 4.11

Temperature	$V_{\text{turn on}}$	Slope
Room Temperature	2.49 V	-17.389
100C	2.49 V	-18.223
150C	2.27 V	-14.802
200C	2.13 V	-16.205
250C	2.06 V	-15.674
300C	2.01 V	-14.023

Table 4.2 Turn on voltage and F-N plots of Figures 4.11 and 4.12



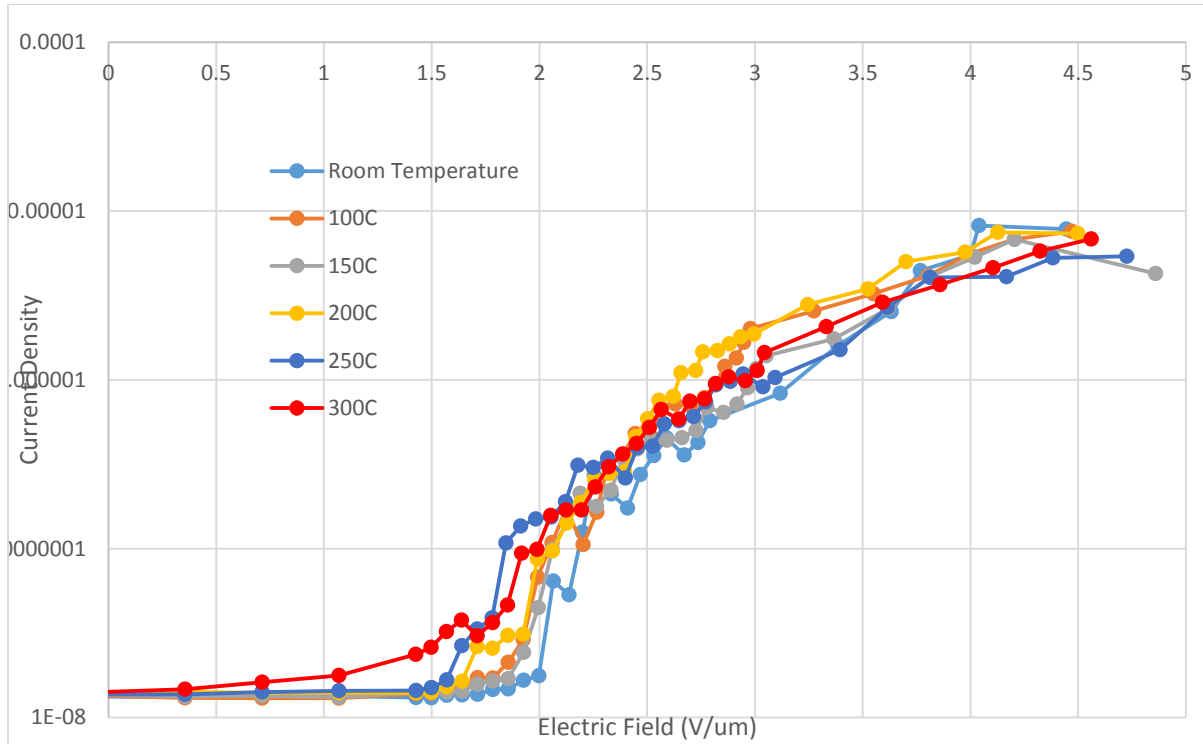


Figure 4.13 Field Emission Data for Sample 2 of Fe and C Sputtering

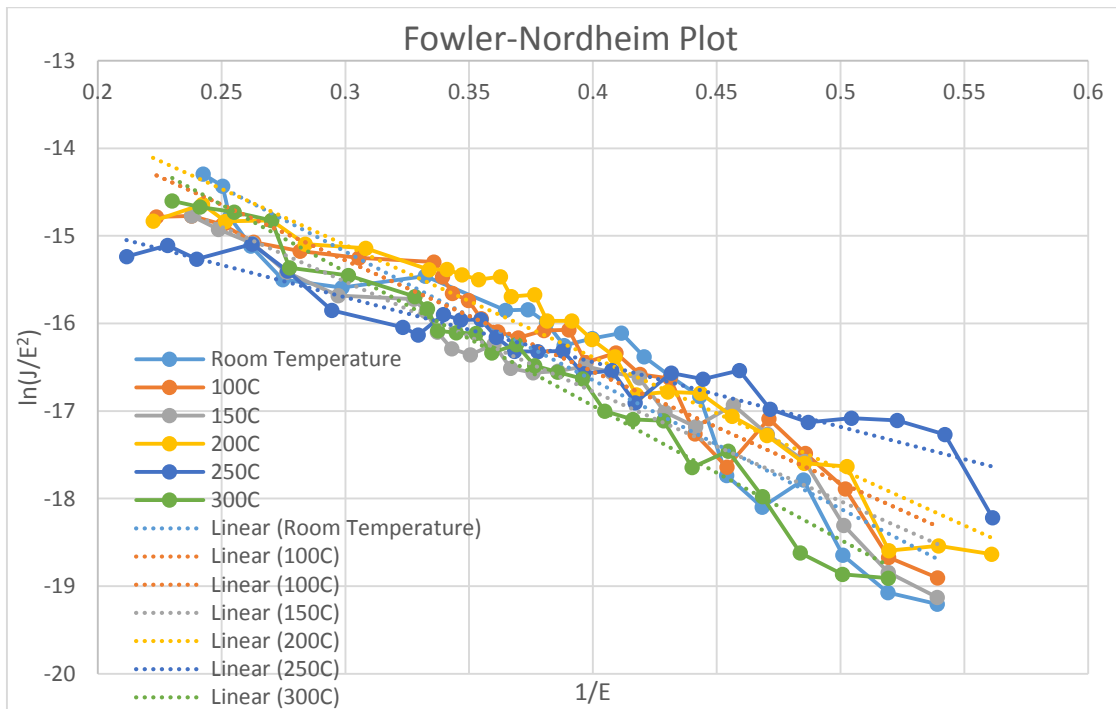


Figure 4.14 Fowler-Nordheim Plot of Field Emission Data in Figure 4.13

Temperature	$V_{\text{turn on}}$	Slope
Room Temperature	1.93 V	-14.679
100C	1.78 V	-12.708
150C	1.86 V	-12.536
200C	1.64 V	-12.574
250C	1.50 V	-8.296
300C	1.43 V	-15.294

Table 4.3 Turn on voltage and F-N plots of Figures 4.13 and 4.14

In this next data, a carbon nanotube sample was made using a 5 minute sputtering, 8 hour anneal, and 40 minute chemical vapor deposition process. But instead of using the piece of ceramic with the stainless steel rod another carbon nanotube sample was used at the anode to see how the field emission data would change. Data for this sample was taken from room temperature to 200°C. The data follows the same pattern as the other data with the turn on voltage decreasing as temperature increases. With this sample the turn on voltage is almost a half a volt less than the other samples presented in this thesis.

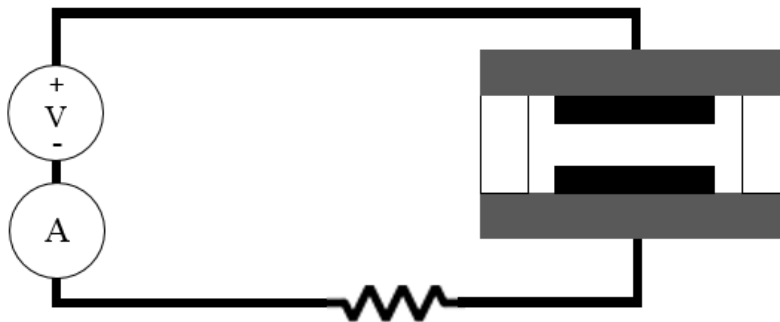


Figure 4.15 CNT Diode Setup with CNT as Anode

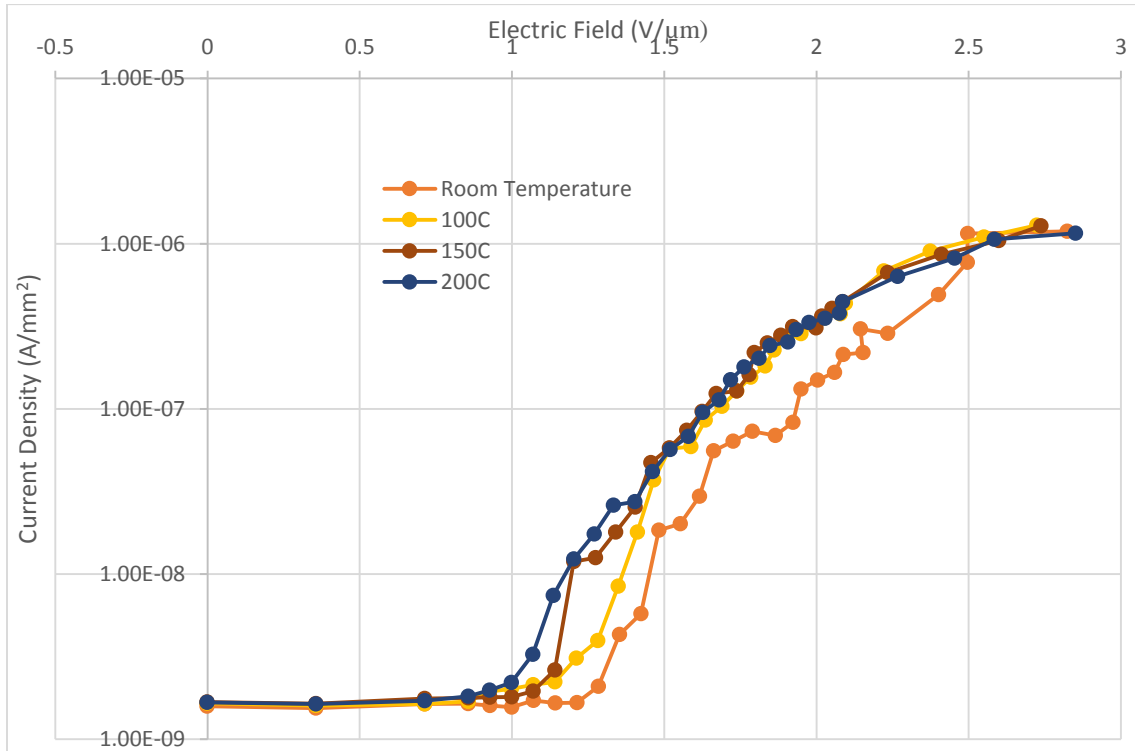


Figure 4.16 Field Emission Data of CNTs as anode

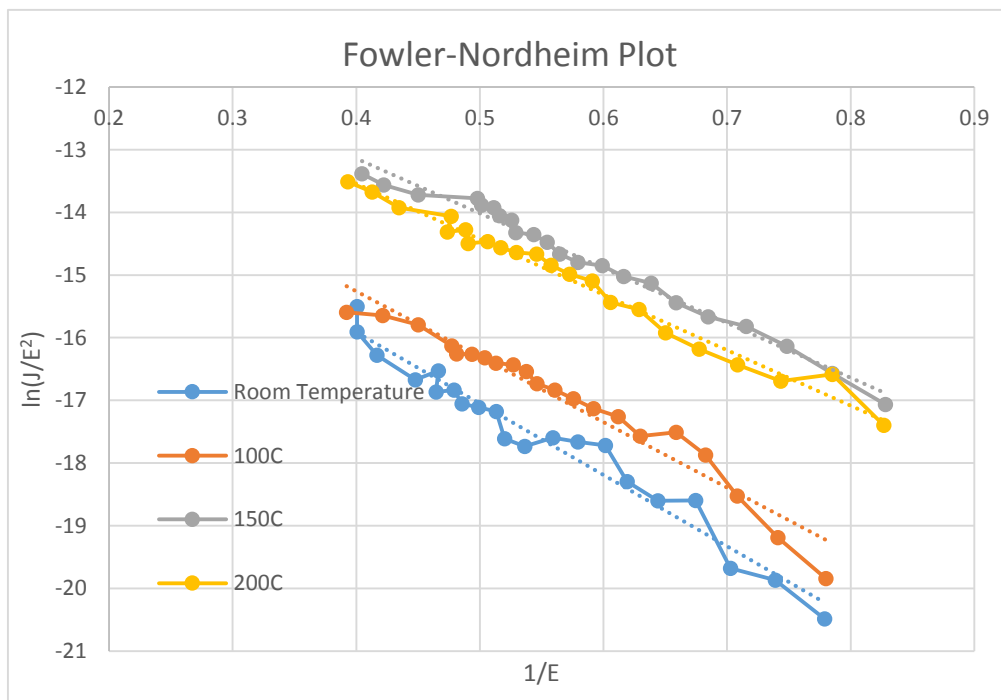


Figure 4.17 Fowler-Nordheim Plot of Field Emission Data in Figure 4.16

Temperature	V <sub>turn on</sub>	Slope
Room Temperature	1.21 V	-11.398
100C	1.14 V	-10.434
150C	1.07 V	-8.747
200C	0.93 V	-8.8501

Table 4.4 Turn on voltage and F-N plots of Figures 4.16 and 4.17

#### **4.5 Thermionic Emission Contribution:**

Using the Thermionic Emission equation described earlier, the current densities for each of the temperatures tested were calculated.

Temperature	Current Density
100C	$2.776 * 10^{-50} \text{ A/m}^2$
150C	$5.426 * 10^{-43} \text{ A/m}^2$
200C	$3.129 * 10^{-37} \text{ A/m}^2$
250C	$1.458 * 10^{-32} \text{ A/m}^2$
300C	$1.059 * 10^{-28} \text{ A/m}^2$

Table 4.5 Current Density Values from Thermionic Emission

The current density values from thermionic emission that were calculated from the equation show that the current density is so low that it did not contribute to lowering the turn on voltage of the carbon nanotube diode. The work function is a constant of 4.5 eV for these carbon nanotube samples. When the equation is solved with the temperature values used in the tests, the values are shown in Table 5.

#### **4.6 High Temperature Bake and Field Emission Test:**

There was also a test set up to see if the carbon nanotubes could survive at high temperatures. For this test, a sample is baked at 850°C for 30 minutes. Field emission data was taken from this carbon nanotube sample before and after the bake. The data can be seen in Figure

4.18. This shows that the carbon nanotube can survive at high temperatures. The field emission data from after the bake was better than the field emission data taken before the bake. The turn on voltage of the carbon nanotubes before the bake was 1.02V and the turn on voltage after the bake was 0.7V.

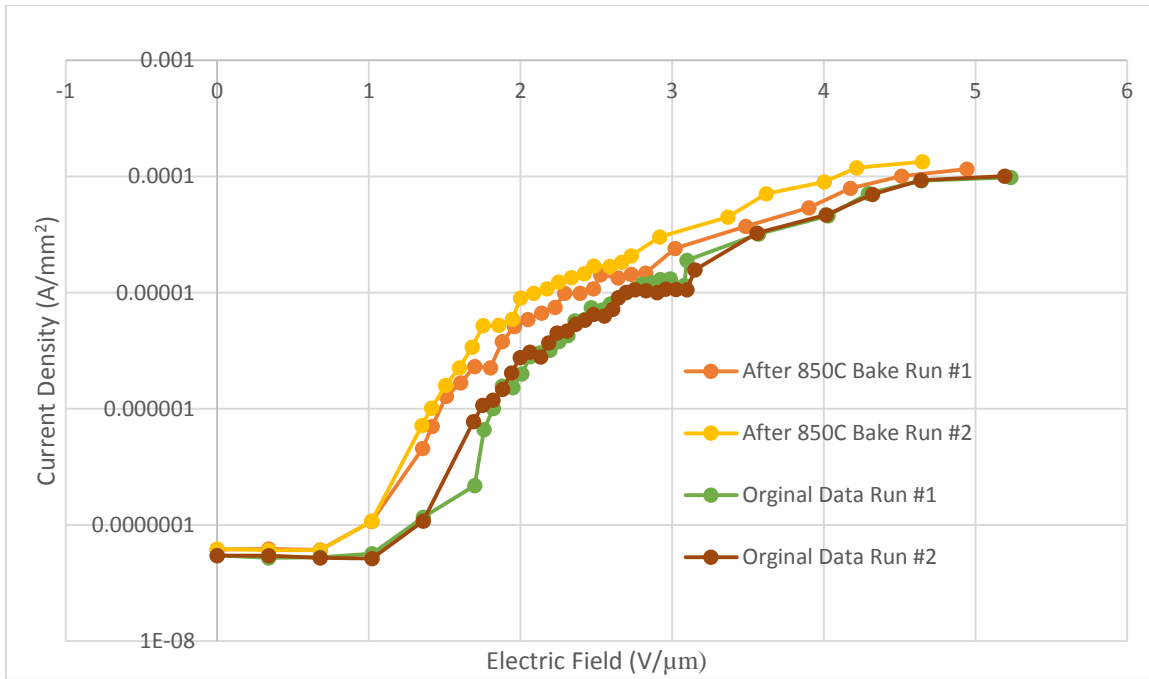


Figure 4.18 Field Emission Data from Sample after Bake

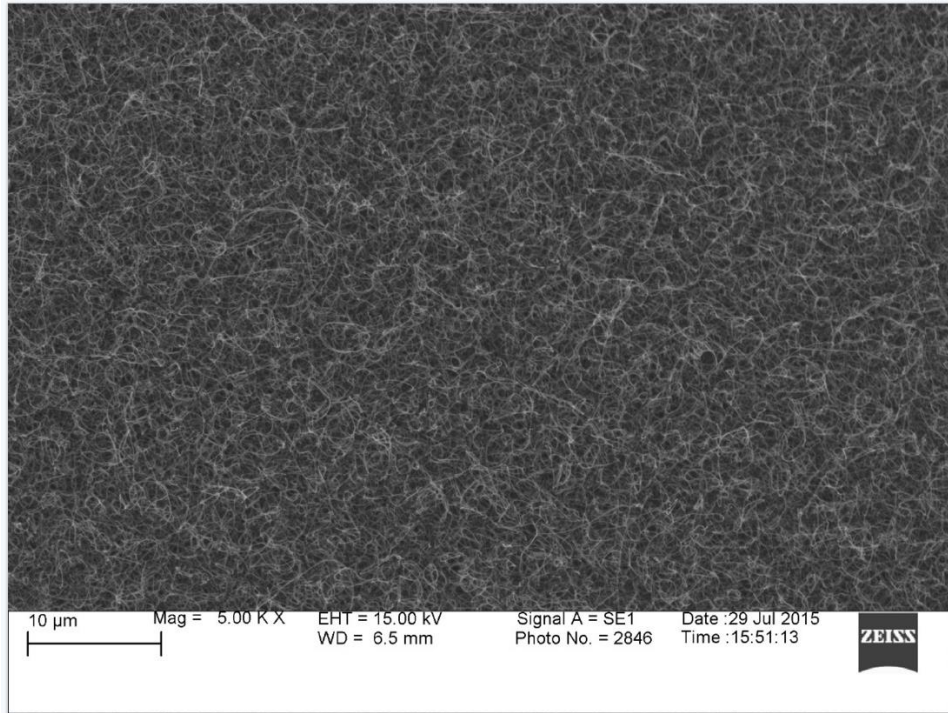


Figure 4.19 SEM Image of CNTs at 5k after the 850°C Bake

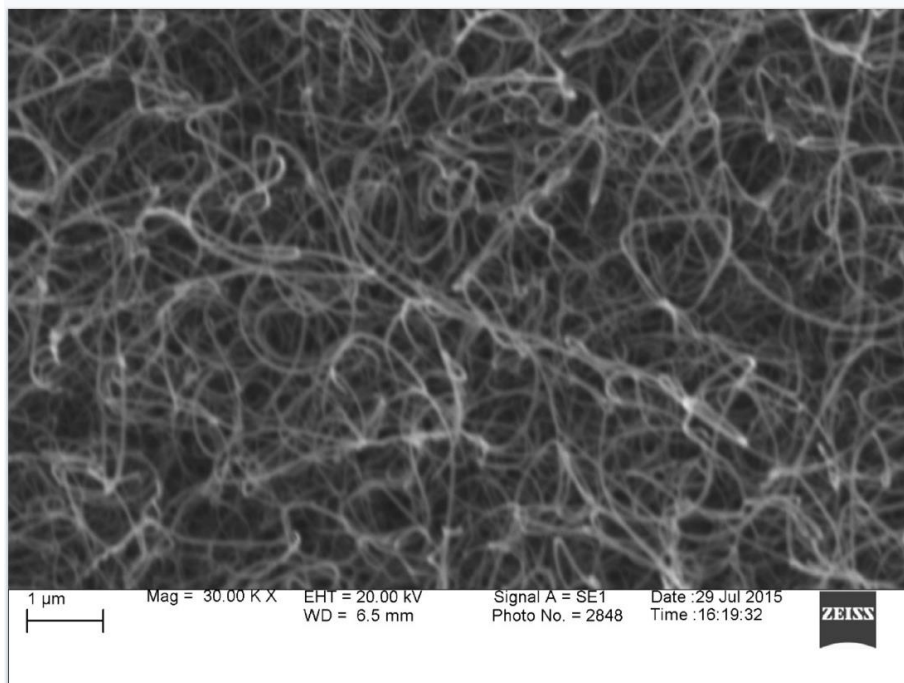


Figure 4.20 SEM Image of CNTs at 30k after the 850° Bake

## CHAPTER 5

### Carbon Nanotube Patch Antenna

#### **5.1 Design:**

Microstrip patch antennas have become the most in demand antenna based on their applications. Their merits include low fabrication weights and cost, and the ability to operate operating in high frequency range. The main drawback of microstrip antennas have their main drawback in terms of narrow bandwidth, low efficiency and relatively large size [26]. Most to all of the patch antennas use copper as the radiating material. A feedline is used to excite and radiate by direct or indirect contact. There are many different techniques of feeding. The four most popular techniques are coaxial probe fee, microstrip line, aperture coupling, and proximity coupling [27]. The carbon nanotube patch antennas used in this thesis use the microstrip feed method.

The geometry used in this thesis for the carbon nanotube patch antenna is a rectangular shape. There has been extensive research about this kind of patch antenna and there are formulas available to calculate the dimension of the radiating patch. The patch used in this thesis is a half-wave rectangular patch antenna (or half-wave patch). The length of this antenna is approximately half of the wave length in terms of the dielectric of the substrate material that is used [28].

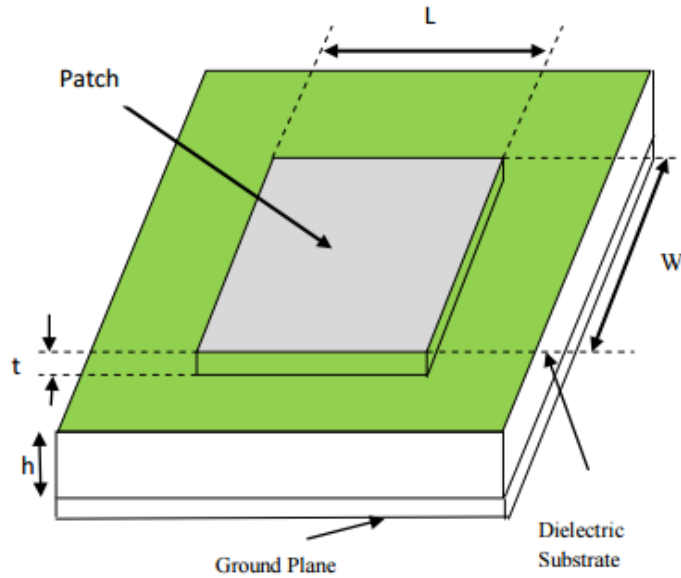


Figure 5.1 Rectangular Patch Antenna [29]

Two different antennas were designed for two different frequency values, 2.4 GHz and 5 GHz. The length of a patch antenna determines the frequency it will radiate at. The electric field distribution for a half-wave patch has fringing fields that extend beyond the patch. This fringing makes the patch effectively longer than its physical length, so the resonant length is less than a half-wavelength [28]. The calculation for patches' length is:

$$\text{Length} = 0.49 \frac{\lambda}{\sqrt{\epsilon_r}}$$

where  $\lambda$  is the wavelength and  $\epsilon_r$  is the substrate dielectric constant. The substrate used to make these carbon nanotube patch antennas is 99.6% Alumina Oxide (ceramic) which has a  $\epsilon_r = 9.8$ .

Frequency	Length of Patch
2.4 GHz	19.6mm
5 GHz	9.39mm

Table 5.1 Lengths of the Two Different Patch Antennas



Two different methods were taken to design the widths of the patches. The width W is used to adjust the input resistance [28]. The width for the 2.4 GHz patch was calculated using an equation for high efficiency [28]. This equation for width is:

$$\text{Width} = \frac{\lambda}{2} \left( \frac{\epsilon_r + 1}{2} \right)^{-1/2}$$

where  $\lambda$  is the wavelength and  $\epsilon_r$  is the substrate dielectric constant. Working this equation out gives a width of 26.88mm. The width of the 5 GHz was solved in a different way. Using the high efficiency equation, the antenna input impedance will be higher than the 50  $\Omega$  which is ideal so the antenna and output are matched. Using this equation:

$$Z_A = 90 \frac{\epsilon_r^2 + 1}{\epsilon_r} \left( \frac{L}{W} \right)^2 \quad [28]$$

with  $Z_A$  being the antenna impedance, the width of the patch can be found for the exact antenna impedance desired. Setting  $Z_A = 50 \Omega$  and  $L = 9.39\text{mm}$ , the width for 5 GHz is equal to 41.6mm. The high efficiency equation for width was used instead of designing the width from the antenna impedance equation for 2.4 GHz because the width from the antenna impedance equation turned out to be wider than the ceramic substrates available in the lab.

## **5.2 Fabrication:**

The fabrication of the carbon nanotube patch antenna is very similar to the fabrication of the carbon nanotube samples used in the field emission experiments. The substrate used to grow the carbon nanotubes on for the patch is 99.6% alumina oxide (ceramic). The only major difference in the fabrication process is masking the ceramic substrate.

The first part of the fabrication of the patch antenna is to create a mask for the ceramic substrate. Since masks were not available with the dimensions of the two antenna sizes for a

photolithography process, the masks were made from cutting Kapton tape and placing it on the ceramic substrate to create an open square of exposed ceramic with the dimensions of the antennas.

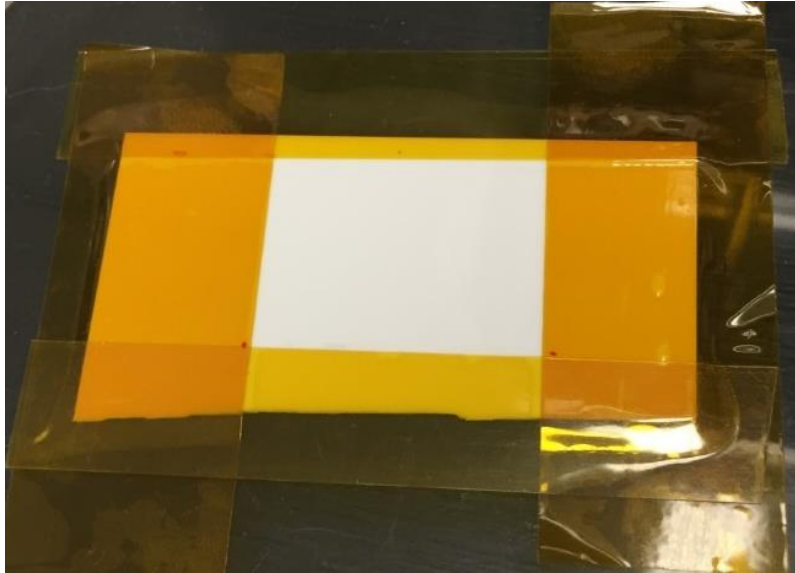


Figure 5.2 Ceramic Substrate with Kapton Tape Mask

The next step in the fabrication process is sputtering. This procedure was done exactly the same way as the carbon nanotube sample for field emission described in the sputtering section of Chapter 3. Iron was first sputtered onto the surface for 5 minutes and then carbon was then sputtered onto the surface for 8 minutes. This procedure was done for the 2.4 GHz and the 5 GHz patch antenna design.

The last step is to grow the carbon nanotubes in the chemical vapor deposition chamber. First the Kapton tape is removed from the ceramic sample and placed into the chamber. The same procedures were done as described in the chemical vapor deposition section of Chapter 3. The only part of the procedure that changed was the time the ceramic sample was left in the chamber. For these patch antennas, the ceramic sample was left in the chemical vapor deposition process for 1 hour.

The next part is to make the feed line and put the ground plane on the back of the ceramic. For the feed line, a piece of copper tape was cut thin to around 1mm and placed on the ceramic wafer at the edge of the carbon nanotubes to the edge of the ceramic. Using a small brush, silver paste was used to make the connection from the copper feed line to the carbon nanotube surface. Using a large sheet of copper tape, the back of the tape was peeled off the back side of the ceramic and placed onto the sticky side of the copper sheet. A knife was used to cut the extra copper around the ceramic substrate.

The last part of the fabrication process was to add an SMA connector to the copper feed line. To make a good connection, the SMA connector was soldered onto the copper feed line and also soldered to the copper back side. The finished antennas can be seen below in Figures 5.3 and 5.4



Figure 5.3 2.4 GHz Carbon Nanotube Patch Antenna

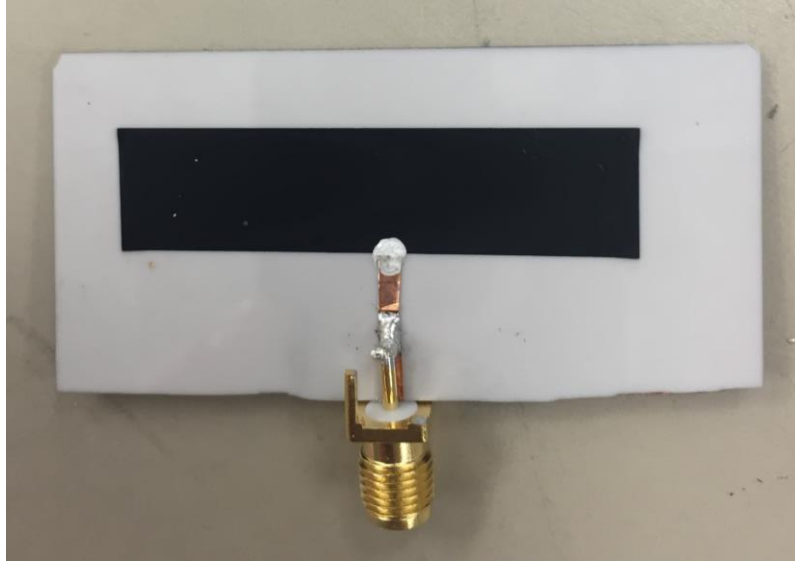


Figure 5.4 5 GHz Carbon Nanotube Patch Antenna

### **5.3 Results:**

The goal for this project was to determine if chemical vapor deposition grown carbon nanotubes could be used instead of copper as a radiating material for a patch antenna. Using a Hewlett Packard 8753C Network Analyzer with a Hewlett Packard 85047A S-Parameter Test Set, an S11 measurement was performed on both of the antennas.



Figure 5.5 Network Analyzer Used for S11 Measurements

Plotting the log magnitude of the S11 measurements for the two antennas gave the two plots in

Figure 5.6 and 5.7.

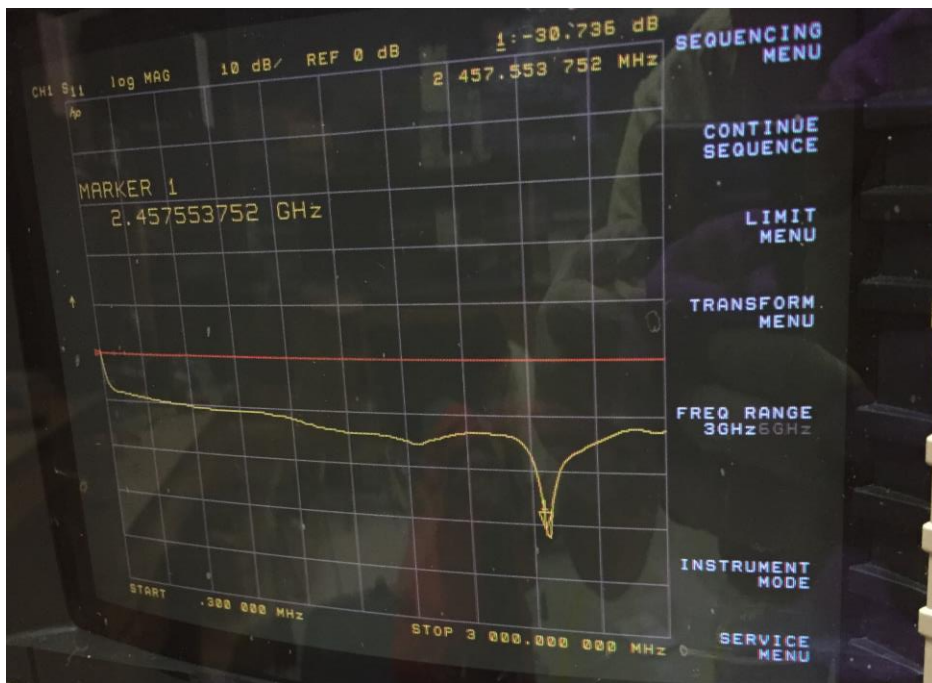


Figure 5.6 S11 Measurement for the 2.4 GHz Patch Antenna



Figure 5.7 S11 Measurement for the 5 GHz Patch Antenna

The two figures above show that both antenna are matched right around the frequency they were designed for. The 5 GHz antenna has a few small dips at other frequencies, but the frequency it is matched at is right at 5 GHz.

One problem with using carbon nanotubes as the radiating material for patch antennas is that the resistivity of the material can be large. If the resistivity is too large, the material will just absorb all of the wave trying to be transmitted. That is one of the main reasons to do a longer chemical vapor deposition process, which helped to decrease the resistivity of the carbon nanotubes.

## **CHAPTER 6**

### **Conclusion**

Carbon nanotubes are known to be one of the best materials for electron field emissions. The multiwalled carbon nanotubes that were used in this thesis were grown by using a set recipe. A silicon substrate would be patterned using photolithography processing in a clean room. Iron was then sputtered onto the substrate (Carbon was also sputtered in some of the samples), then the substrate was placed on a hot plate to anneal for 8 hours. The substrate was placed into a chemical vapor deposition chamber for 40 minutes to grow the carbon nanotubes. The carbon nanotubes were put into a diode setup in a vacuum chamber. A power supply and current meter were connected to test and measure the diode. The diode was tested in the vacuum chamber at various temperatures to see how the field emission properties would change. For just about every test, as the temperature of the diode increased, the turn on voltage of the diode decreased. Carbon nanotubes can also be used instead of copper for a patch antenna design.

## References:

- [1] M. O'Connell, *Carbon nanotubes*. Boca Raton, FL: CRC/Taylor & Francis, 2006.
- [2] H. Pierson, *Handbook of carbon, graphite, diamond, and fullerenes*. Park Ridge, N.J.: Noyes Publications, 1993.
- [3] Ajayan, Pulickel M., and Otto Z. Zhou. "Applications of carbon nanotubes." *Carbon nanotubes*. Springer Berlin Heidelberg, 2001. 391-425.
- [4] Awasthi, Kalpana, Anchal Srivastava, and O. N. Srivastava. "Synthesis of carbon nanotubes." *Journal of nanoscience and nanotechnology* 5.10 (2005): 1616-1636.
- [5] Seraphin, Supapan, et al. "Effect of processing conditions on the morphology and yield of carbon nanotubes." *Carbon* 31.5 (1993): 685-689.
- [6] Puretzky, A. A., et al. "In situ imaging and spectroscopy of single-wall carbon nanotube synthesis by laser vaporization." *Applied Physics Letters* 76.2 (2000): 182-184.
- [7] T. Dürkop, S. Getty, E. Cobas and M. Fuhrer, "Extraordinary Mobility in Semiconducting Carbon Nanotubes," *Nano Letters*, vol. 4, no. 1, pp. 35-39, 2004.
- [8] S. Okamura, *History of electron tubes*. Tokyo: Ohmsha, 1994.
- [9] I. Poole, "Vacuum Tube Theory | Thermionic Valve | Radio-Electronics.Com," *Radio-electronics.com*, 2016. [Online]. Available: <http://www.radio-electronics.com/info/data/thermionic-valves/vacuum-tube-theory/tube-tutorial-basics.php>.
- [10] Circuitstoday.com, "Vacuum tube diodes - Electronic Circuits and Diagram-Electronics Projects and Design," 2009. [Online]. Available: <http://www.circuitstoday.com/vacuum-diodes>. [Accessed: 27- Jan- 2016].
- [11] S. Jo, Y. Tu, Z. Huang, D. Carnahan, D. Wang and Z. Ren, "Effect of length and spacing of vertically aligned carbon nanotubes on field emission properties," *Appl. Phys. Lett.*, vol. 82, no. 20, p. 3520, 2003.
- [12] Fowler, Ralph Howard, and L. Nordheim. "Electron emission in intense electric fields." *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. Vol. 119. No. 781. The Royal Society, 1928.
- [13] Dyke, W. P., and W. W. Dolan. "Field emission." *Advances in electronics and electron physics* 8 (1956): 89-185.
- [14] K. Ghatak and S. Bhattacharya, *Fowler-Nordheim Field Emission From Semiconductors And Their Nanostructures*. Berlin: Springer Berlin, 2011.
- [15] R. Forbes, "Refining the application of Fowler–Nordheim theory," *Ultramicroscopy*, vol. 79, no. 1-4, pp. 11-23, 1999.



- [16] Imicromaterials.com, "HMDS," 2016. [Online]. Available: <http://www.imicromaterials.com/technical/hmds>.
- [17] Dill, Frederick H., et al. "Characterization of positive photoresist." *Electron Devices, IEEE Transactions on* 22.7 (1975): 445-452.
- [18] Www2.ece.gatech.edu, "Photolithography," 2016. [Online]. Available: <http://www2.ece.gatech.edu/research/labs/vc/theory/photolith.html>.
- [19] <http://www.directvacuum.com>, "Magnetron Sputtering Technology," [Online]. Available: [http://www.directvacuum.com/pdf/what\\_is\\_sputtering.pdf](http://www.directvacuum.com/pdf/what_is_sputtering.pdf).
- [20] Jung, Michael. *Carbon Nanotube Vacuum Tube Diode RFID Tag*. Thesis, Auburn University. Print.
- [21] Park, Jong-Hee, and T. S. Sudarshan, eds. *Chemical vapor deposition*. Vol. 2. ASM International, 2001.
- [22] Vacuum-guide.com, "Turbomolecular pump - how it works", 2016. [Online]. Available: <http://www.vacuum-guide.com/english/equipment/hw-turbomolecular.htm>.
- [23] *Introduction to Bayard-Alpert Ionization Gauges*, 1st ed. Boulder, Colorado: Granville-Phillips, 1991, pp. 1-2.
- [24] Murphy, Edward Leo, and R. H. Good Jr. "Thermionic emission, field emission, and the transition region." *Physical review* 102.6 (1956): 1464.
- [25] "Thermionic Emission," 2016. [Online]. Available: <http://www.physics.csbsju.edu/370/thermionic.pdf>.
- [26] G. KumarSharma and N. Sharma, "Analysis of a Rectangular Microstrip Patch Antenna with EBG Structures," *International Journal of Computer Applications*, vol. 84, no. 13, pp. 39-42, 2013.
- [27] Ramesh Garg, Prakash Bartia, Inder Bahl, Apisak Ittipiboon, "Microstrip Antenna Design Handbook," 2001, pp 1-68, 253-316 Artech House Inc. Norwood, MA.
- [28] W. Stutzman and G. Thiele, *Antenna Theory and Design*, 3rd ed. John Wiley & Sons, 2013, pp. 466-469.
- [29] Singh I, Tripathi VS. Micro strip patch antenna and its applications: a survey. *Int J Comp Tech Appl*. 2011; 2(5):1595- 1599.
- [30] M. Lenzlinger, "Fowler-Nordheim Tunneling into Thermally Grown SiO<sub>2</sub>," *J. Appl. Phys.*, vol. 40, no. 1, p. 278, 1969.
- [31] Tsai, Chung-Nan. "Selective and Non-selective Synthesis of Carbon Nanotubes (CNTs) by Chemical Vapor Deposition (CVD) Characterization: Catalysts and Underlayers Effects on Field Emission Properties." (2012).

[32] H. Zhao, "Design and construction of carbon nanotubes (CNTs) triggered pseudospark switch," Ph.D. dissertation, Auburn Univ., Auburn, AL, 2012.

## Appendix:

```
function varargout = FieldTest(varargin)
% FIELDTEST M-file for FieldTest.fig
%   FIELDTEST, by itself, creates a new FIELDTEST or raises the existing
%   singleton*.
%
%   H = FIELDTEST returns the handle to a new FIELDTEST or the handle to
%   the existing singleton*.
%
%   FIELDTEST('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in FIELDTEST.M with the given input arguments.
%
%   FIELDTEST('Property','Value',...) creates a new FIELDTEST or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before FieldTest_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to FieldTest_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help FieldTest

% Last Modified by GUIDE v2.5 12-Dec-2010 14:54:15

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @FieldTest_OpeningFcn, ...
                  'gui_OutputFcn', @FieldTest_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before FieldTest is made visible.
function FieldTest_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to FieldTest (see VARARGIN)

% Choose default command line output for FieldTest
handles.output = hObject;
handles.flag=0;

set(handles.edit5,'String',num2str(0));
set(handles.edit6,'String',num2str(20));
set(handles.edit7,'String',num2str(100));
set(handles.edit10,'String',num2str(120));
set(handles.edit9,'String',num2str(10));
set(handles.edit8,'String',num2str(260));
set(handles.edit13,'String',num2str(300));
set(handles.edit12,'String',num2str(80));
set(handles.edit11,'String',num2str(1200));
set(handles.edit16,'String',num2str(600));
set(handles.edit17,'String',num2str(5));
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes FieldTest wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = FieldTest_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

handles.flag=1;
sport = serial('COM3');
sport.Terminator = 'CR/LF';
fopen(sport);
fprintf(sport, '++mode 1');
fprintf(sport, '++addr 20');
fprintf(sport, '++auto 0');
fprintf(sport, 'HVON');

```

```

fprintf(sport, '++addr 20');
fprintf(sport, '++auto 0');
fprintf(sport, 'VSET 0'); % initialize the power source

handles.flag=1;
handles.sport=sport;

guidata(hObject, handles);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

if handles.flag==1

fprintf(handles.sport, '++addr 20');
fprintf(handles.sport, '++auto 0');
fprintf(handles.sport, 'VSET 0');
fprintf(handles.sport, 'HVOF');
fprintf(handles.sport, '++loc');
fclose(handles.sport);

handles.flag=0;
else
    msgbox('Connect device first');
end

guidata(hObject, handles);

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit2_Callback(hObject, eventdata, handles)
% hObject handle to edit2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
% str2double(get(hObject,'String')) returns contents of edit2 as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)
% hObject handle to edit4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
% str2double(get(hObject,'String')) returns contents of edit4 as a double

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)
% hObject handle to edit11 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit11 as text
% str2double(get(hObject,'String')) returns contents of edit11 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit11 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)
% hObject handle to edit12 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit12 as text
% str2double(get(hObject,'String')) returns contents of edit12 as a double

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit12 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)
% hObject handle to edit13 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit13 as text
% str2double(get(hObject,'String')) returns contents of edit13 as a double

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit13 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

```

```
    set(hObject,'BackgroundColor','white');  
end
```

```
function edit8_Callback(hObject, eventdata, handles)  
% hObject handle to edit8 (see GCBO)  
% eventdata reserved - to be defined in a future version of MATLAB  
% handles structure with handles and user data (see GUIDATA)  
  
% Hints: get(hObject,'String') returns contents of edit8 as text  
% str2double(get(hObject,'String')) returns contents of edit8 as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function edit8_CreateFcn(hObject, eventdata, handles)  
% hObject handle to edit8 (see GCBO)  
% eventdata reserved - to be defined in a future version of MATLAB  
% handles empty - handles not created until after all CreateFcns called  
  
% Hint: edit controls usually have a white background on Windows.  
% See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function edit9_Callback(hObject, eventdata, handles)  
% hObject handle to edit9 (see GCBO)  
% eventdata reserved - to be defined in a future version of MATLAB  
% handles structure with handles and user data (see GUIDATA)  
  
% Hints: get(hObject,'String') returns contents of edit9 as text  
% str2double(get(hObject,'String')) returns contents of edit9 as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function edit9_CreateFcn(hObject, eventdata, handles)  
% hObject handle to edit9 (see GCBO)  
% eventdata reserved - to be defined in a future version of MATLAB  
% handles empty - handles not created until after all CreateFcns called  
  
% Hint: edit controls usually have a white background on Windows.  
% See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function edit10_Callback(hObject, eventdata, handles)  
% hObject handle to edit10 (see GCBO)  
% eventdata reserved - to be defined in a future version of MATLAB  
% handles structure with handles and user data (see GUIDATA)
```



```

% Hints: get(hObject,'String') returns contents of edit10 as text
%   str2double(get(hObject,'String')) returns contents of edit10 as a double

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject   handle to edit10 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%   See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)
% hObject   handle to edit7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%   str2double(get(hObject,'String')) returns contents of edit7 as a double

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject   handle to edit7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%   See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)
% hObject   handle to edit6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%   str2double(get(hObject,'String')) returns contents of edit6 as a double

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject   handle to edit6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit5_Callback(hObject, eventdata, handles)
% hObject handle to edit5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
% str2double(get(hObject,'String')) returns contents of edit5 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes Start Button
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
if handles.flag==0
    msgbox('Connect device first');
else
handles.sport.Timeout = 10;
warning off all
%Sample Number
handles.samplenum=get(handles.edit1,'String');
%Number of Runs
handles.runnum=str2num(get(handles.edit4,'String'));
%Voltage values
handles.Vst1=str2num(get(handles.edit5,'String'));
handles.Vse1=str2num(get(handles.edit6,'String'));
handles.Vsp1=str2num(get(handles.edit7,'String'));
handles.Vst2=str2num(get(handles.edit10,'String'));
handles.Vse2=str2num(get(handles.edit9,'String'));
handles.Vsp2=str2num(get(handles.edit8,'String'));
handles.Vst3=str2num(get(handles.edit13,'String'));

```

```

handles.Vse3=str2num(get(handles.edit12,'String'));
handles.Vsp3=str2num(get(handles.edit11,'String'));
Con_power='VSET';

if
isempty(handles.samplenum)||isempty(handles.runnum)||isempty(handles.Vst1)||isempty(handles.Vse1)||is
empty(handles.Vsp1)||isempty(handles.Vst2)||isempty(handles.Vse2)||isempty(handles.Vsp2)||isempty(ha
ndles.Vst3)||isempty(handles.Vse3)||isempty(handles.Vsp3)
    msgbox('Incomplete parameter, please check ','Incomplete parameter','warn');
else
dataU=[(handles.Vst1+handles.Vse1:handles.Vse1:handles.Vsp1)
(handles.Vst2:handles.Vse2:handles.Vsp2) (handles.Vst3:handles.Vse3:handles.Vsp3) (handles.Vsp3-
handles.Vse3:-handles.Vse3:handles.Vst3) (handles.Vsp2:-handles.Vse2:handles.Vst2) (handles.Vsp1:-
handles.Vse1:handles.Vst1)];
%dataU=-1*dataU;
h = waitbar(0,'Please wait...');
for i=1:handles.runnum
    for j=1:length(dataU)
        number=1;

fprintf(handles.sport, '++addr 20');
fprintf(handles.sport, '++auto 0');
fprintf(handles.sport, [Con_power, num2str(dataU(j))]);

%Displays voltage value in the GUI
set(handles.edit14,'String',num2str(dataU(j)));

while 1
fprintf(handles.sport, '++addr 22');
fprintf(handles.sport, '++auto 1');
dataII =fread(handles.sport, 16);

if ~isempty(dataII)&& length(dataII)==16

temp=(char(dataII));
dataIII(number)=str2num(temp(6:16));
%Displays the current value in the GUI
set(handles.edit15,'String',num2str(dataIII(number)));
% pause(0.5);
number=number+1;
end

if number==6
    break;
end
pause(3)
end
dataI(j)=(dataIII(1)+dataIII(2)+dataIII(3)+dataIII(4)+dataIII(5))/5;

pause(1)

```

```

end
% xlsxwrite(['c:\fieldtest\' ,num2str(handles.samplennum),\'\',num2str(i)],[dataU;dataI]);
xlsxwrite([handles.samplennum, '-',num2str(i)],[dataU;dataI]);
clear dataI;
waitbar(i/handles.runnum,h)
pause(30)
end
close(h)

cf = 2000;           % carrier frequency (Hz)
sf = 22050;         % sample frequency (Hz)
d = 1.0;           % duration (s)
n = sf * d;         % number of samples
s = (1:n) / sf;     % sound data preparation
s = sin(2 * pi * cf * s); % sinusoidal modulation
sound(s, sf);
end
end

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

if handles.flag==0
    close all;
else
    msgbox('Please disconnect the device first!');
end

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
priorPorts = instrfind ; % finds any existing Serial Ports in MATLAB
delete(priorPorts) ; % and deletes them
msgbox('Reset finished');

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
field;

function edit14_Callback(hObject, eventdata, handles)

```

```

% hObject  handle to edit14 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit14 as text
%       str2double(get(hObject,'String')) returns contents of edit14 as a double

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
% hObject  handle to edit14 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit15_Callback(hObject, eventdata, handles)
% hObject  handle to edit15 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit15 as text
%       str2double(get(hObject,'String')) returns contents of edit15 as a double

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
% hObject  handle to edit15 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function pushbutton3_CreateFcn(hObject, eventdata, handles)
% hObject  handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  empty - handles not created until after all CreateFcns called

function edit16_Callback(hObject, eventdata, handles)
% hObject  handle to edit16 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit16 as text
% str2double(get(hObject,'String')) returns contents of edit16 as a double

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit16 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit17_Callback(hObject, eventdata, handles)
% hObject handle to edit17 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit17 as text
% str2double(get(hObject,'String')) returns contents of edit17 as a double

% --- Executes during object creation, after setting all properties.
function edit17_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit17 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global stopflg;
stopflg=0;
if handles.flag==0
    msgbox('Connect device first');
else

```

```

handles.sport.Timeout = 10;
warning off all

handles.appfield=str2num(get(handles.edit16,'String'));
handles.Dtime=str2num(get(handles.edit17,'String'));

handles.samplenum=str2num(get(handles.edit1,'String'));
Con_power='VSET ';

if isempty(handles.appfield)||isempty(handles.Dtime)||isempty(handles.samplenum)
    msgbox('Incomplete parameter, please check ','Incomplete parameter','warn');
else

num_points=1000;
fprintf(handles.sport, '++addr 20');
fprintf(handles.sport, '++auto 0');
fprintf(handles.sport, [Con_power, num2str(handles.appfield)]);

set(handles.edit14,'String',num2str(handles.appfield));

h = waitbar(0,'Please wait...');
for k=1:num_points
    guidata(hObject, handles);
    if stopflg==1;
        break;
    else

while 1
    fprintf(handles.sport, '++addr 22');
    fprintf(handles.sport, '++auto 1');
    dataII =fread(handles.sport, 17);

    if ~isempty(dataII)&&length(dataII)==17
        temp=(char(dataII));
        data_I(k)=str2num(temp(6:16));
        % temp=dataII(6:end);
        % data_I(k)=str2num(temp);
        set(handles.edit15,'String',num2str(data_I(k)));
        pause(handles.Dtime*3600/num_points);
        break;

    end

end

end
waitbar(k/num_points,h)
end
close(h)

```

```

fprintf(handles.sport, '++addr 20');
fprintf(handles.sport, '++auto 0');
a=0;
fprintf(handles.sport, [Con_power, num2str(a)]);
set(handles.edit14,'String',num2str(''));
set(handles.edit15,'String',num2str(''));
xlswrite('Lifetest.xls',[(1:k)*handles.Dtime*3600/num_points;data_I]);
clear data_I;

```

```

cf = 2000;           % carrier frequency (Hz)
sf = 22050;         % sample frequency (Hz)
d = 1.0;            % duration (s)
n = sf * d;         % number of samples
s = (1:n) / sf;     % sound data preparation
s = sin(2 * pi * cf * s); % sinusoidal modulation
sound(s, sf);
end
end

```

```

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global stopflg;
stopflg=1;

guidata(hObject, handles);

```