Master's Thesis

석사 학위논문

Study Of Reduced Graphene Oxide Property and Application

Seong Yong Woo (우 성 용 禹 成 龍)

Department of

Robotics Engineering

DGIST

2016

Master's Thesis

석사 학위논문

Study Of Reduced Graphene Oxide Property and Application

Seong Yong Woo (우 성 용 禹 成 龍)

Department of

Robotics Engineering

DGIST

2016

Study of Reduced Graphene Oxide Property and Application

Advisor : Professor Sang Jun Moon

Co-advisor : Professor Ho Chun Lee

By

Seong Yong Woo

Department of Robotics Engineering

DGIST

A thesis submitted to the faculty of DGIST in partial fulfillment of the requirements for the degree of Master of Science in the Department of Robotics Engineering. The study was conducted in accordance with Code of Research Ethics[1]

01. 08. 2016

Approved by

Professor San Jun Moon (Signature)
(Advisor)

Professor Ho Chun Lee (Signature)
(Co-Advisor)

[1] Declaration of Ethical Conduct in Research**:** I, as a graduate student of DGIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

Study Of Reduced Graphene Oxide Property and Application

Seongyong Woo

Accepted in partial fulfillment of the requirements for the degree of Master of Science.

11. 24. 2015

Head of Committee Prof. Sang Jun Moon (인)

Committee Member Prof. Ho Chun Lee (인)

Committee Member Prof. Won Gu Lee (인)

MS/RT
201323012

우 성 용. Seong Yong Woo. Study of Reduced Graphene Oxide Property and Application Department of Robotics Engineering. 2016. P88. Advisors Prof. Moon, San Jun, Co-Advisors Prof. Lee, Ho Chun.

ABSTRACT

.

Graphene Oxide is a substance capable of working in a typical laboratory or indoor environment. Usage is also simple and also because of the toxic materials in contact is very easy to work. If the non-conductive nature of the process itself have only column characteristics or by laser energy reduction processes occurring in the applied part "Reduced Graphene Oxide" has electrical conductivity.

While as the non-processed surface of non-conductive properties is manufactured in a form to enable this with the two characteristics simultaneously. The electrical resistance value varies according to the amount of energy input. That is the difference in electrical resistance occurs. When the process is repeated for the same position as the laser has a built-in property of the resistance value decreases. Because of this characteristic it has been utilized in the semiconductor and memory, which appears to be the next circuit making the composite itself.

In this study, the trying to make the application of the application fields to see a change with the amount of laser energy thereto and refocus the electrical characteristics of rGO.

rGO is utilized to expect sensor which using swell property in reduction processor and mixing nano particle. Particularly, recent rGO research is bio and blood base clinic (medical field) to target. This method is coping with the sensor using an expensive reactive substances that are produced by the will of the existing proteins and various enzymes to the sensor using the physical properties, the enzyme or the protein sensor a breakthrough resolved in a matter of persistent storage and expiration date. In fabricating the sensor, so that utilized rGO in glucose measurement is expected to be opened, the length of the large quantity of production at a lower price.

These should be capable of quick and easy to manufacture device to mass production. In this study, we propose a method using direct laser scribe. Laser equipment is easy to manufacture and consists of a configuration using the industrial robot or other general-purpose motor. In addition, easily and expeditiously making processor can manufacture sensor with a laser scribe. There shall be no restrictions on the disposal or

storage of materials. The storage and disposal easier and self-produced and mass production of the sensor should be presented several experimental and environmental conditions for access to the produce.

This study presents a laser scriber capable of easily and quickly making how to create the rGO with the non-toxic GO. rGO sensor performance analysis that manufactured through laser fabrication process. These will be proposed the availability, future development direction and improvement of the glucose measurement.

Keywords: Graphene, Graphene Oxide, Reduced Graphene Oxide, glucose, blood sugar, diabetes, laser cutting

Contents

List of Figures

List of Tables

1. INTRODUCTION

1.1 Direct Laser Scribe

The requirement for the production of micron, submicron and even nanometer structures at high-speed and low-unit cost is characteristic to the manufacturing of most high-tech devices. The continuing development of pulsed laser materials processing positions it as a key enabling technology, providing high-resolution, high accuracy, high-speed and flexible production [1]. Direct laser scribing technique has been researched a lot in the lab [2]. Laser write equipment is the same shape and structure. However, the laser source is different from the source of laser scriber equipment using the high power of the laser relative to the writing device.

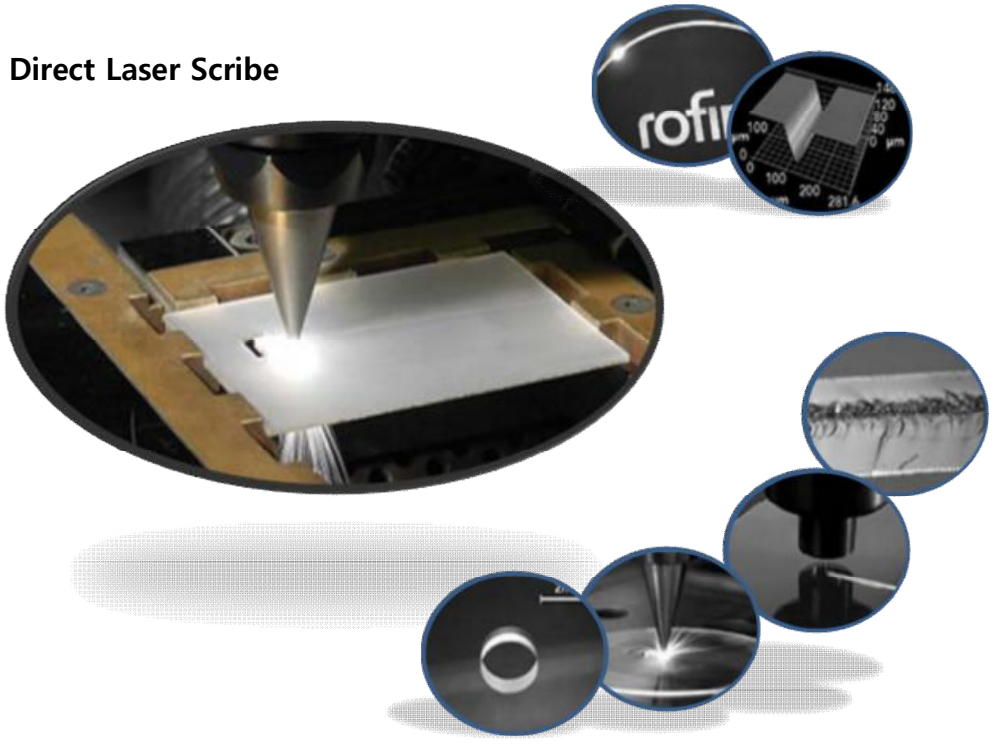

Figure 1-1 How to use a variety of laser scribe : Marking, Inside Scribing, Separation, Cutting, Drilling [3]

Laser scriber is utilized in various applications. For example, marking, inside scribing, cutting, separating, and drilling [3]. The system through the continuous development of technology integration, simple and compact cooling system is became to an easy device to work.

Design tools can be manufactured using a commercial CAD programs [4]. AUTO-CAD program is widely used and generally, to create 2D drawings to work. The reason for 2D work is due to only use the letter and cutting function. If the use in the lithography equipment, the 3D drawing should work.

The moving unit is required for driving the spatial coordinates. 2D, or 3D robot consisting of a motor that makes the driving. Therefore, the difference in speed, accuracy and repeatability of the operation depends heavily on the ability of the robot. The accuracy of the driving motor of the robot permitting, it is possible to obtain a high accuracy of the results. Other factors affecting the accuracy of the focal point of the laser. Precisely correct the focus, the focus is the injection of energy to generate a thin powerful laser.

In this study, by making the laser scriber is to be utilized a high precision drive and the electrode draw. Overall, this configuration is to drive the robot and laser driver. The robot driver utilizes industrial linear robots and laser driver utilizes the DVD-writer to easily save.

1.2 Reduce Graphene Oxide Sequence

Graphene is due to several new excellent properties since it was discovered in 2004 by a carbon atom of a further two-dimensional material has received much attention of researchers. Zero effective mass and high charge moving rate, does not scatter in the room temperature [5].The thinner the material known as Graphene that can conduct electricity or heat. In addition, a strong and flexible material. A method of manufacturing the Graphene are mechanical separation method using a scotch tape [5-9], CVD(Chemical Vapor deposition) method [10], a chemical separation method [11-14], and epitaxial synthesis methods [15]. Working to the highest quality and mass production of Graphene. Research to develop an application device for Graphene properties are being made in various ways [9, 16-19].

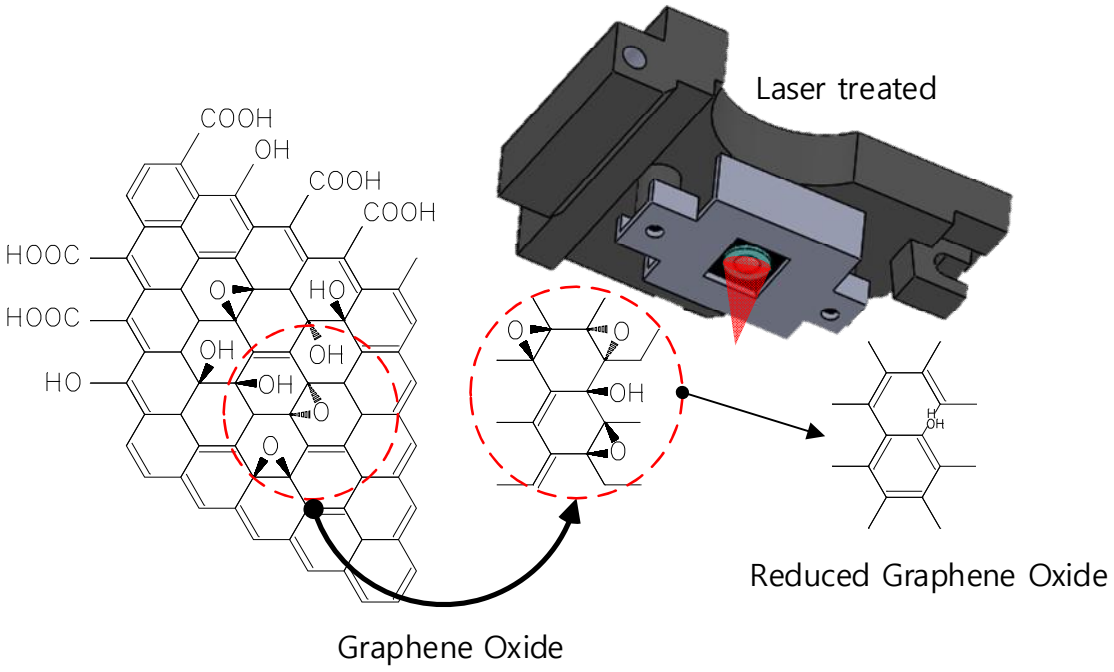

Figure 1-2 Graphene oxide reduction process using direct laser scriber

Graphene Oxide(GO) is prepared by treating Graphene with strong aqueous oxidizing agents such as fuming nitric acid/potassium chlorate or sulphuric acid/potassium permanganate[20]. Most recently GO has been used as the positive electrode in primary lithium cells[21] , for the preparation of membranes and as a model system for membrane states[22-24]. Electrode possesses the lowest charge transfer resistance and highest electro catalytic activity among all of the electrodes used Reduced Graphene Oxide (rGO) [25]. rGO will receive a lot of attention for meaningful research [26].

Figure 1.2 shows the rGO conversion by a laser. Receiving the energy from the laser to the GO is transformed to the rGO. Turns to rGO coupling structure is changed. Have a GO and other physical and electrical characteristics. In particular, the electrical properties, but GO is in non-conductive material, the conductive rGO is changed. There is no difference between the D band and the G value in Raman spectroscopic results. However, the value of the G band is very slightly shifted [27].

Electrical Features vary on the number of laser operation [27]. When the operation of the laser several times the resistance value drops. Therefore, the resistance value due to the number of laser irradiation is changed as table 1.1. Due to the various properties of rGO, it has been utilized in many fields.

Time	State	Resistance(kΩ)
0	GO	580,000
1	rGO	8.2
2	rGO	4.8
3	rGO	2.3

Table 1.1 Impedance measurements of the Focal length[27]

In this study, the electrical and chemical properties of the identified rGO through multiple analysis system.It will be utilized to be applied to the field through the analyzed data.

1.3 Reduce Graphene Oxide used in glucose measurement

Glucose is an important metabolic material in the human body and index of variation in glucose concentration can be used to directly reflect the human body health [28]. Diabetes is one of the most common chronic diseases, in recent years, increases rapidly that occurs. The development of new glucose sensors provides a measure of the ease by an effective blood sugar. Currently various glucose measurement method, for example, the photo acoustic spectroscopy, Raman spectroscopy, infrared spectroscopy [29-31], capacitive detection [32] , colorimetry [33], surface plasmon resonance biosensor [34] and electrochemiluminescene [35]. Electrochemical glucose sensors has attracted considerable interest because of the high selectivity, sensitivity and short response time. Electrical measurements of glucose: enzymes and non-enzymes way can be measured [36]. Glucose measurement using the enzyme is inconvenient because of the influence of temperature and pH [37]. In addition, limiting the application method and the high production cost is a problem.

Noble metals, such as Au [38], Pt[39] , Ag[40] and Pt-PB[41] alloy has been used for the electrochemical oxidation of glucose, and indicate a high sensitivity. However, there are also some disadvantages method using a noble metal. For example, chloride ion poisoning[42], rapid deactivation, low selectivity and a high cost.

Graphene is consist of monolayer carbon atom material with high surface area, superior mechanical and electrical properties [43-45]. This property has potential applications in nano-electronics, sensor, capacitors and a variety of high technology [46-48]. However, mass production of Graphene is needed is an alternative to other hard problem. So, the production easier, faster, cheaper prices is to use the oxidation Graphene. However, possible to use oxide

Graphene is a special treatment because there is no electrical conductivity. It is possible to change the electrical properties through the reduction process.

This study will use a reductive way through the introduction by the laser scriber device. Subjected to the reduction process using a laser scriber rGO it can be utilized as a sensor. Method using a catalyst to enhance a more properties can be thought.

2. SYSTEM DESIGN

2.1 Direct Laser scribing.

Direct laser scribe device is a common CD, DVD system from being used for cutter and engraver. Using the linearity of the high power laser is irradiated to object by driver. Moving the laser to the part of the object space coordinates of the object by moving the process to the desired shape. Figure 2.1 is displayed briefly direct laser writing system. In addition, the linear motion robot, and built-in moving the 3axis to hold the form of the processing object, the robot is equipped with a driver amplifier for each axis, and each amplifier passes the command by the controller. Computer(Top-level controller) is connected to the each controller and user interfacing.

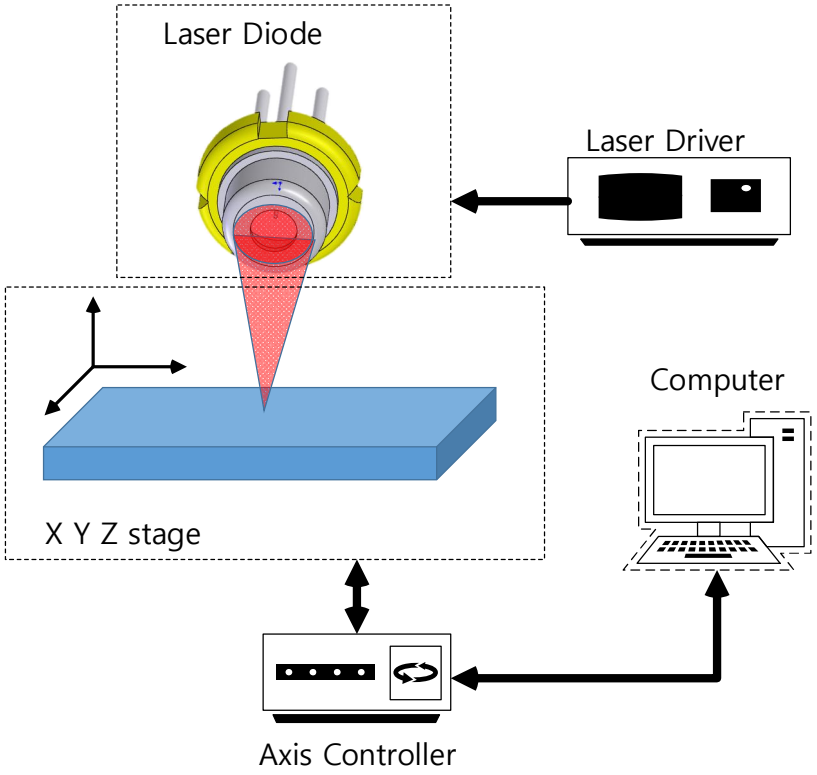

Figure 2-1 Concept image of direct laser scribe System

In this study, using a robot for two spatial coordinates plotted on the system. One is to use a small size and the micro-system is suitable Nano Max-Ts (THORLABS) product was configuring the system, the other way to use an industrial motor drive unit were to be a large amount of processing. Each drive motor has a driver and host controller, the control is possible using the NI-LAB VIEW systems.

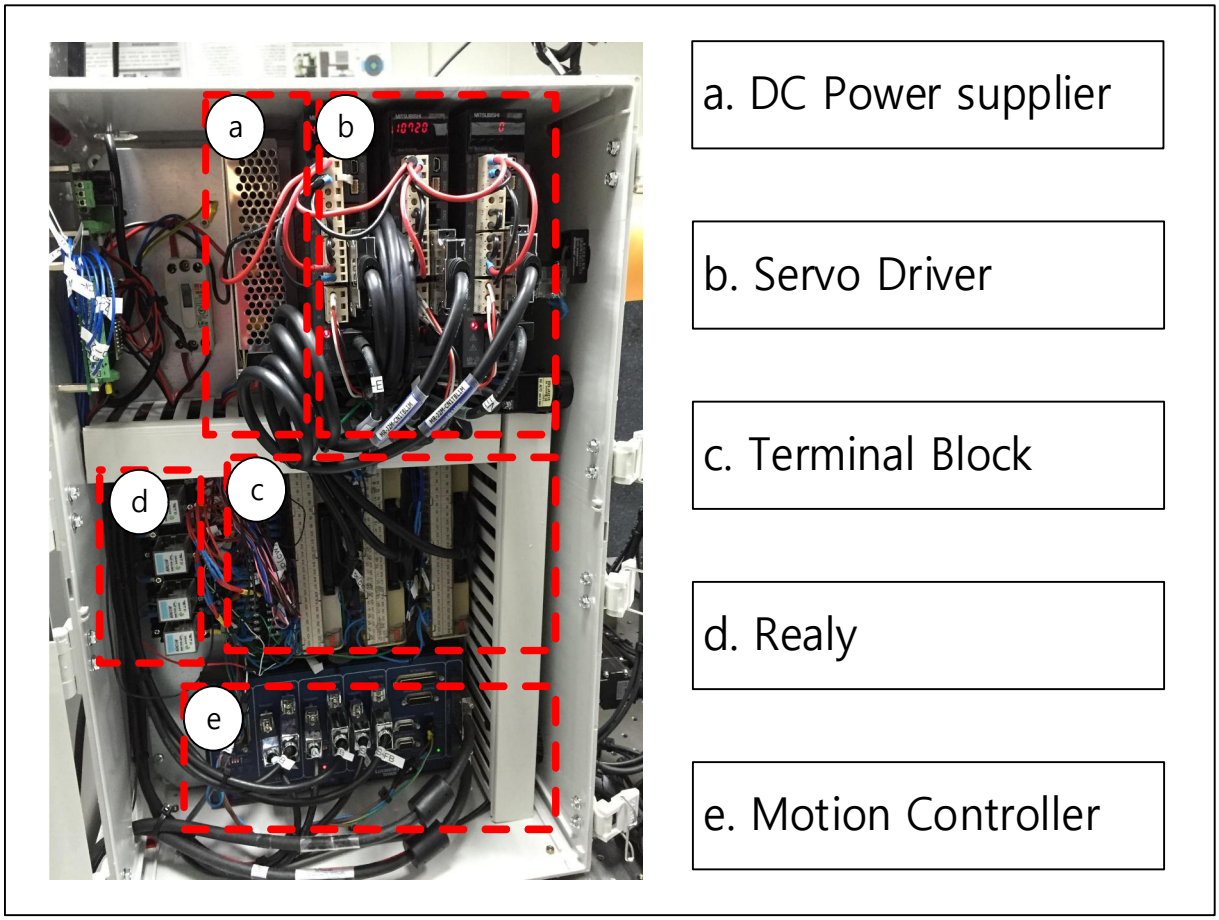

Figure 2-2 Figure Internal configuration control box : (a)Control input and output signals and DC power, (b) servo motor driver for driving, (c) Terminal block for the control signal and output signal cables, (d) Relay for controlling the brake and servo ON, (e) The motion controller to manage the servo and output signal

Commercial systems include a separate control box. The control box has a power supply wiring portion, terminal blocks, relays, control are located. In the case of commercial motor phase power supply and three-phase power, depending on the type used. In this study, using a single-phase power. (b.)Rotary motor servo is driven by servo driver(MR-J3-10A type, Mitsubishi).

(a.)24V DC power supply to control power use a DC power supply and a built-in a load circuit. Such as the 24V power source to operate the brakes on each axis, or operates as the power source for driving the (c.)GP I / O terminal of CN1, to control power to the (e.)Motion controller is also used. (d.)The relay is controlled by a large power as a power supply which switches on internal or built-in magnet, it is used to separate the control unit and the load. Built in the control box relays the role of driving the ON / OFF of the servo motor and the brake. CN1 terminal is connected with the servo driver, and thus the I / O configuration to be connected to the motion controller and relays. Since all of the critical features to the terminal, connected to the note is required. In particular, the control pulse input section for the driving of the motor "CW +, CW-, CCW +, CCW-" Line Driver receives the input signal. Emergency stop. SERVO-ON, a limit signal is to be input must be driven Each input-output terminal is the signal line by using the photo coupler as a problem of the external noise are separated independently. Motion controller with NI products (NI-PCI-7350, National Instruments) receives the signal from the servo drive control with pulse sensor around the whole drive. The available internal I / O mapping program, can easily configure automatic production system using the feature. However, because the limit signals only get PNP type, caution is necessary when applying the sensor. If in order to connect to the NPN type equipped with a pull-up resistor to the signal output terminal should a change in signal output. Limit signal sensor is capable of power-on of the wide range of 5V ~ 24V. In this study, an additional circuit in the output due to the transformation of the Line Driver

signals and must supply the 5V supply to provide power to additional circuitry. When the control power input to the different types, resistance to external noise, but this advantage, the wiring is complicated, it has a disadvantage.

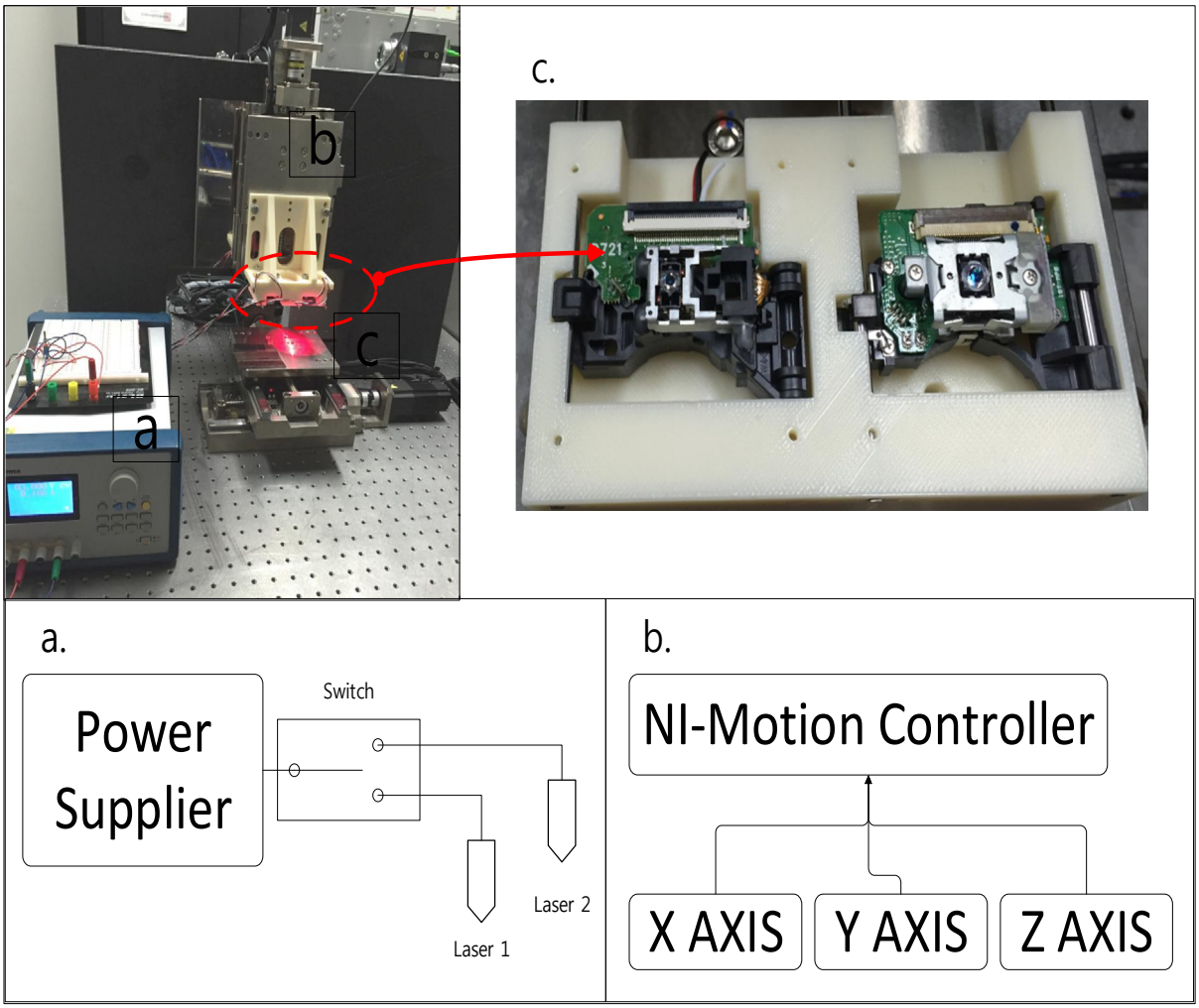

Figure 2-3 Internal configuration control box : (a)By selecting the laser source operating the switching modules), (b)3-axis motion controller and servo control functions with built-in driver,, (c)2 laser fixing jig head

Configuration of the system using industrial equipment was configured, as shown in Figure 2.3. (b.) 3-axis stage is tied as the y-axis and x-axis, Z-axis was mounted on a separate base. Each servo motor (HF-KP13B, Mitsubishi) has a built-in brake and encoder. The NI

Motion Controller is equipped with a separate terminal board UMI-7774(National Instruments) for the connection of the control panel. Cables and connectors to connect the laser power supply and laser head located around to the driver.

Optical Pick Up(OPU) was equipped with a dual-head type, housing was the creation using 3D printers(uPrinter SE Pluse, Stratasys). The laser source was used as the (c.)OPU which is built on a commercial DVD-MULTI(Super-Multi GH-24NS90, LG) products. A laser built into the OPU is different from the output in accordance with the operation method, (a.)CD when a difference between playing a DVD and a CD are "C" and "GND" connecting terminal, and when using DVD "D" and "GND" for use. The wavelength band of each of the source is the CD when the wavelength of 780 nm, during operation in the DVD 650 nm. Bout focal length also differ, and the "C" mode when more than 1mm, "D" mode when the working area is caused to 1mm or less. If you work in read mode and is driven by small power, Write mode when this is driven by a higher power.

2.2 Motion Device Setup

2.2.1 Servo Pack Setup

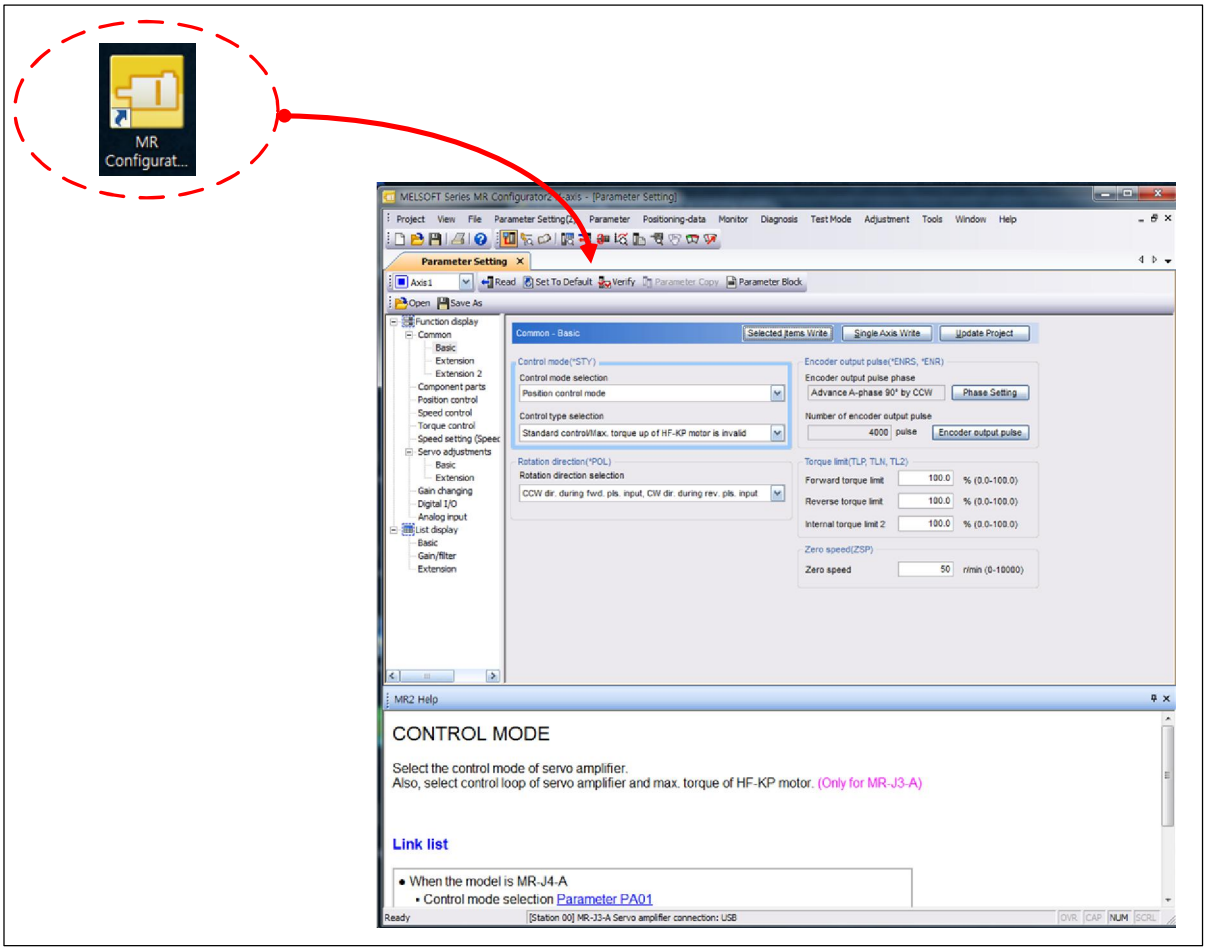

Figure 2-4 Servo Driver Setup : Setting up a program for driving the servo motor "MR Configurator"

Figure 2.2 is a program called "MR Configurator" to use when setting up a dedicated servo drives from Mitsubishi. Through a program for setting up and related to the driving of the servo motor, it is possible to determine the various problems details. In addition to providing test drives and auto-tuning functions and features to enable reliable operation of the

servo motor. In particular, the setup parameters should have made the necessary settings for the drive. If not set, the servo driver in the continuously hold the error condition.

First, an input parameter block to 000C in the menu. The setting of the parameters of the interior, Basic, is set Gain / Filter, Extension, I / O settings allow the user to change. Once this is complete, set the Block setting on the Common Basic menu. Basic settings for the control mode, and to the type of signal to the command from the control a significant change depending on the control mode. Because it information on the position command pulse from the controller through a set in the position control mode. Changing the settings will be saved immediately by pressing the "Write Parameter" button.

Encoder output pulse setting selects the "Set according to the number of input pluses per revolution" among two check boxes on the inside. When this setting is enabled, you can put a number next to "Number of input pulse per revolution" on the inside. Type "3600" by setting the pulse value of the motor wheel, comes in from the external input pulse 3600, it rotates the first wheel. That is, it has a resolution obtained by dividing the one rotation angle of 360 ° to 3600. As a result, it will have a resolution of 0.1 ° per pulse.

There is set the direction of rotation "CCW, dir" and, "CW, dir" two kinds of selection, and, the setting should be set considering the driving direction of the machine. This unit was set to "CCW, dir". "Position Control" menu, set the 100pulse the "In-Position range". However, if high accuracy is required Entering a low value. However, too low values may be slow in-position determination time. "Command pulse input states" There is a total of six options. Largely divided into Negative and Positive, and each menu to Pulse 1 pieces that can be input to the direction and speed "CW, CCW" approach, different menus will receive respectively the speed and direction "Pulse and Direction" scheme, which is a phase difference signal there is an outgoing "A phase / B phase" method.

2.2.2 Controller Setup

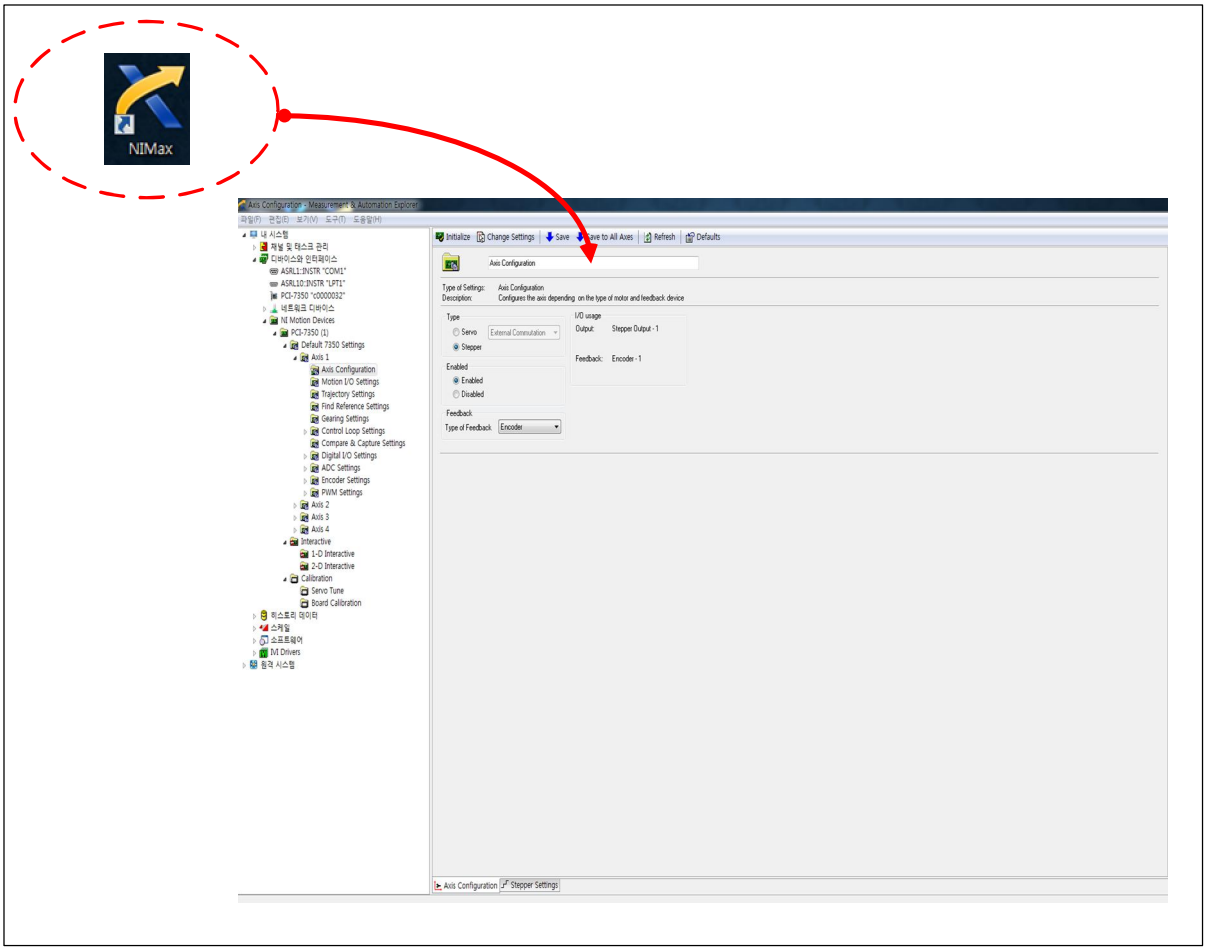

Figure 2-5 Setting controller with NI-MAX for test driving.

The NI devices can be configured and driven through the NI-MAX, which oversees its own device driver. The controller was used in the NI-PXI 7350 of the PCI type, and external devices connected to the panel was used NI-UMI 7774. The controller settings to configure format from the motor, the menu is On "STEP MOTOR", "SERVO MOTOR". In the motor control servo motor controller settings, but select the "STEP MOTOR". Matters related to the encoder is set to "None". The output of the control pulse is set to enter the

"3600", the above values have to be inputted in the same way as the control pulse of the relevant settings servo driver.

Setting the limit sensors are hardware settings are required depending on the type of sensor connected, but special requirements. In general, the domestic use of the NPN-type limit sensor, but NI controller has a basic input sensor of the PNP type. Because of the NPN type to connect the sensor to the NI controller, attached (in the case of a 5V voltage) pull-up resistors 500 ~ 1kohm to the sensor output signal line must be aware of the signal it becomes possible.

After setting up the sensor test drive, it is possible in the "Interactive" menu. The motion profile can be configured to increment or absolute value through the setting of the coordinate system. If the drive is without error in the function, all settings will be no problem.

2.3 Laser Setup

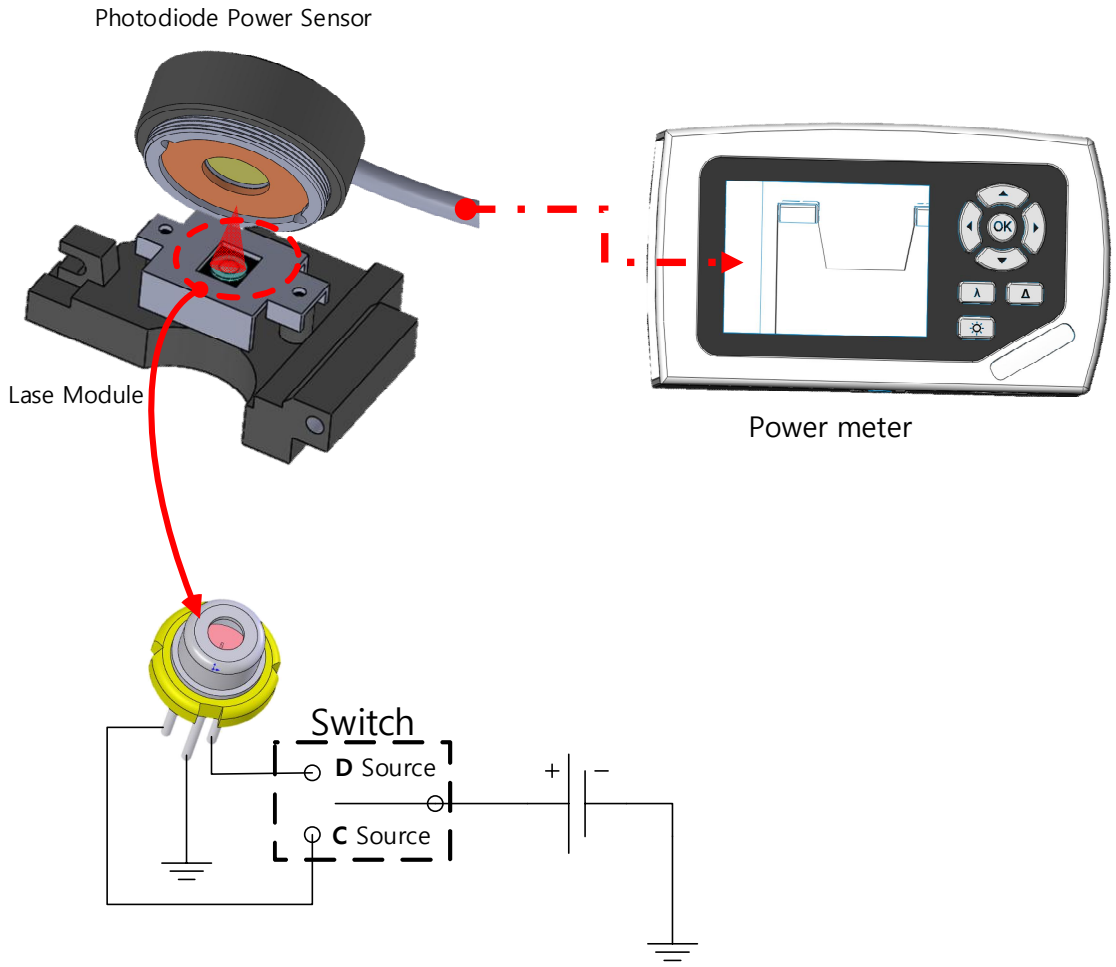

Figure 2-6 Laser driver circuit and check the output equipment.

The laser module using a built-in commercial OPU DVD read / write, there is a laser for CD and DVD inside the OPU. OPU has its own specifications for the lens driving unit to control the sensor and to determine the focal length of the laser focus. But this is not used in the lens driving and hanging in the lock circuit and programs for the operation of the focus sensor. Focus control lens motor is a separate controller is needed in order for a Voice Coil Motor(VCM) for multi-axis drive. So may be used only a laser diode which can be used only when the input power. When the laser is input to the DC power from outside the drive are possible, more than 3 V can be driven at a voltage of 4V or less. If applying the more power is necessary to use careful because this is break down.

Selection of the CD laser (C Source) and the DVD laser (D Source) are readily controlled by controlling the switches mounted on the outside. Because laser source has a different wavelength and power, there is a need for careful in the use. Power of each laser varies according to the voltage input, it is possible to make the power is confirmed by a separate instrument. To check the power was used Photodiode Power meter (PM100D, THORLABS) and a dedicated sensor (S120C, THORLABS).

2.4 Impedance Measurement Setup

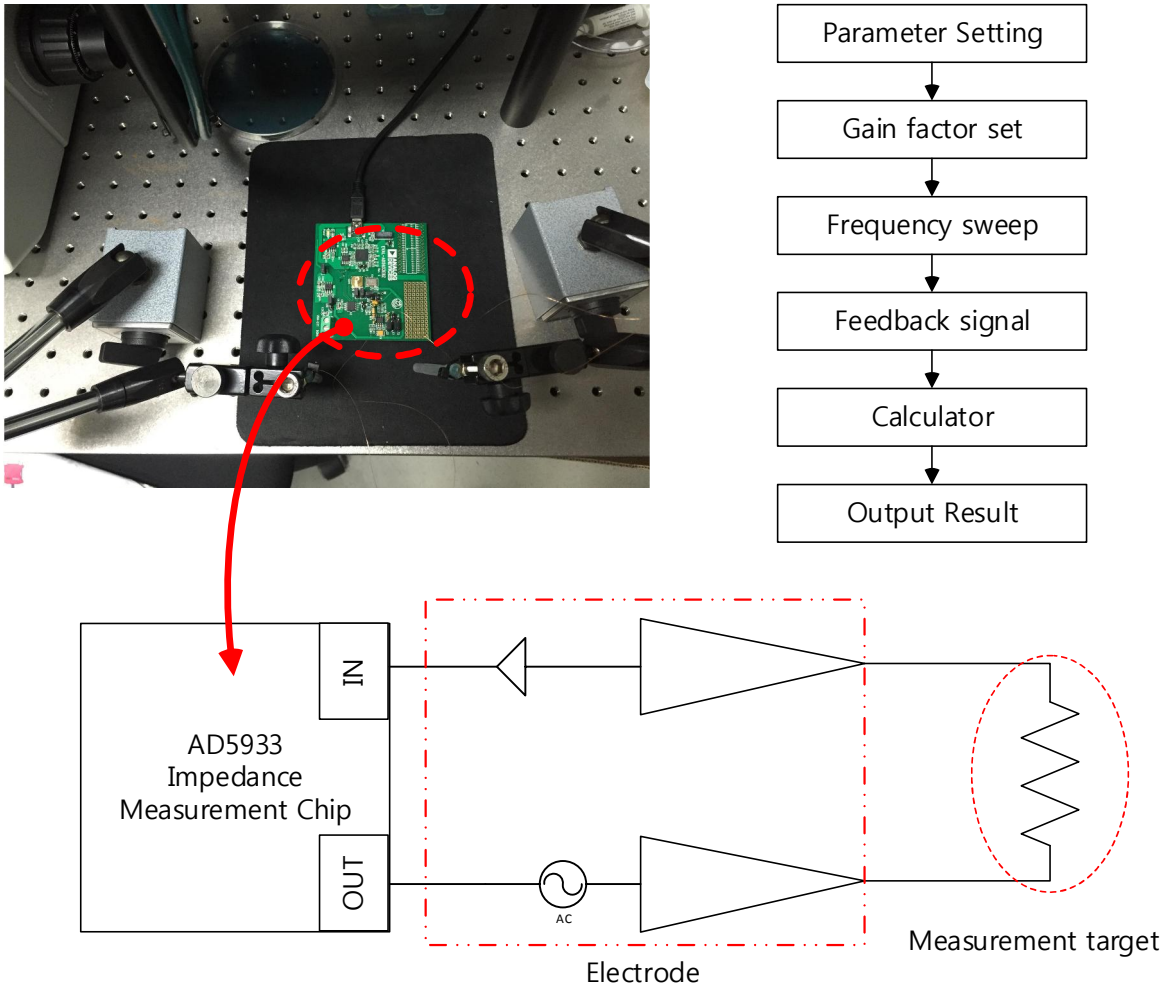

Figure 2-7 Low-frequency impedance measurements using a commercial environment impedance measurement chip(AD5933).

Impedance measurement is an important factor when using the signal line or a sensor. Does not form a line is used as impedance matching for the whole reflection signal and a distortion occurs in the signal, the sensor output value of the characteristic impedance at a particular frequency band, and to see the change to be able to play the role as a sensor. A method for the purposes of this study, because the sensor or the signaling must be the characterization. Impedance measurements in the analysis for each frequency band of a large

area, impedance measuring chip (AD5933, Analog Device) of Figure 2.5 it is possible to analyze the frequency band of 200 kHz.

Before measuring, the internal parameter settings should be preceded. Start measurement frequency increment, and put the number is set to the entire measuring range. AD5933, and to set the input voltage of 0.2, 0.4, 1, and supports 2Vpp voltage signal.

Parameter write, and perform calibration process. When the calibration, and the fixed resistor connected to the electrode, the resistance to input the calibration parameter performs a measurement sequence once. The value measured in the sequence is called the gain factor, within the Imaginary, the calculation is done by Real value.

Calculating a value for each order are, first, the Real and Imaginary impedance measuring chip measurement value based on a signal sent back from the signal. Using the measured value and Real Imaginary Magnitude value is calculated by applying the formula (2-1). Impedance-known value of the formula (2-2) to provide the calibration resistance is calculated Gain factor. When used to calculate the unknown impedance of the formula (2-3). After obtaining Real, Imaginary values of the measurement sequence, and calculates the Magnitude. Calculate the Impedance-unknown pre-computed using the Gain Factor and Magnitude.

$$Magnitude = \sqrt{Real^2 + Imaginary^2} \quad \textbf{(2-1)}$$

$$Gain\ Factor = \frac{1}{Impedance_{know} * Magnitude} \quad \textbf{(2-2)}$$

$$Impedance_{unknow} = \frac{1}{Gain\ Factor * Magnitude} \quad \textbf{(2-3)}$$

AD5933 can not handle formulas. However, the output of the signal using a DSP and Real, Imaginary value derivation is possible. Treatment of the formula is to be used a Micro Control Unit (MCU) mounted on the outside. The MCU and the AD5933 is exchanged each value through the I2C communications. Each value is received through the communication processing by the internal program on the MCU. This study was carried out by utilizing PSOC5(CY8C5866AXI, Cypress) calculates the impedance.

3. Preparation of sample and experiment

3.1 Production of Graphene Oxide film

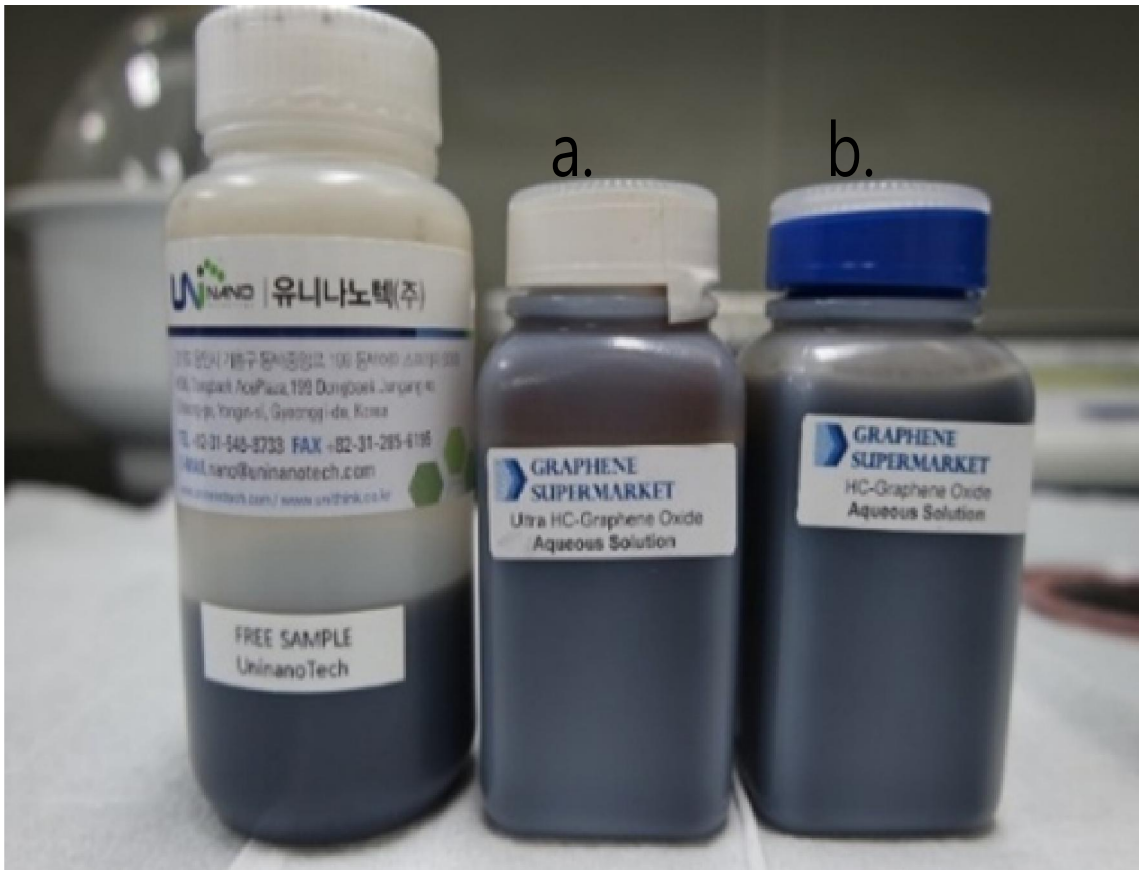

Figure 3-1 Graphene Oxide solutions used in graphene Oxide Film : (a)Ultra-Highly Concentrated Single Layer Graphene Oxide, (b) Highly Concentrated Single Atomic Layer Graphene Oxide

GO for the production of the films were prepared two solutions (Figure 3.1). The concentration level is to determine the nature of the other two solutions, respectively, to prepare a film. (a) Ultra-Highly Concentrated Single Layer GO(175ml, GRAPHENE SUPER MARKET) has a concentration of 6.2 g / L, (b) Highly Concentrated Single Atomic Layer GO(175ml, GRAPHENE SUPER MARKET) of 5g/ L. Due to the differences in concentration of Ultra-Highly Concentrated GO it is stronger than nature gather the central

part Highly Concentrated GO. Both the liquid does not spread, due to the aggregation properties to even the distribution is not easy.

PET to create a highly flexible film quality but may result in peeling. Si-wafer and, using a glass -Wafer, it is possible to obtain a film in a fixed film quality of GO.

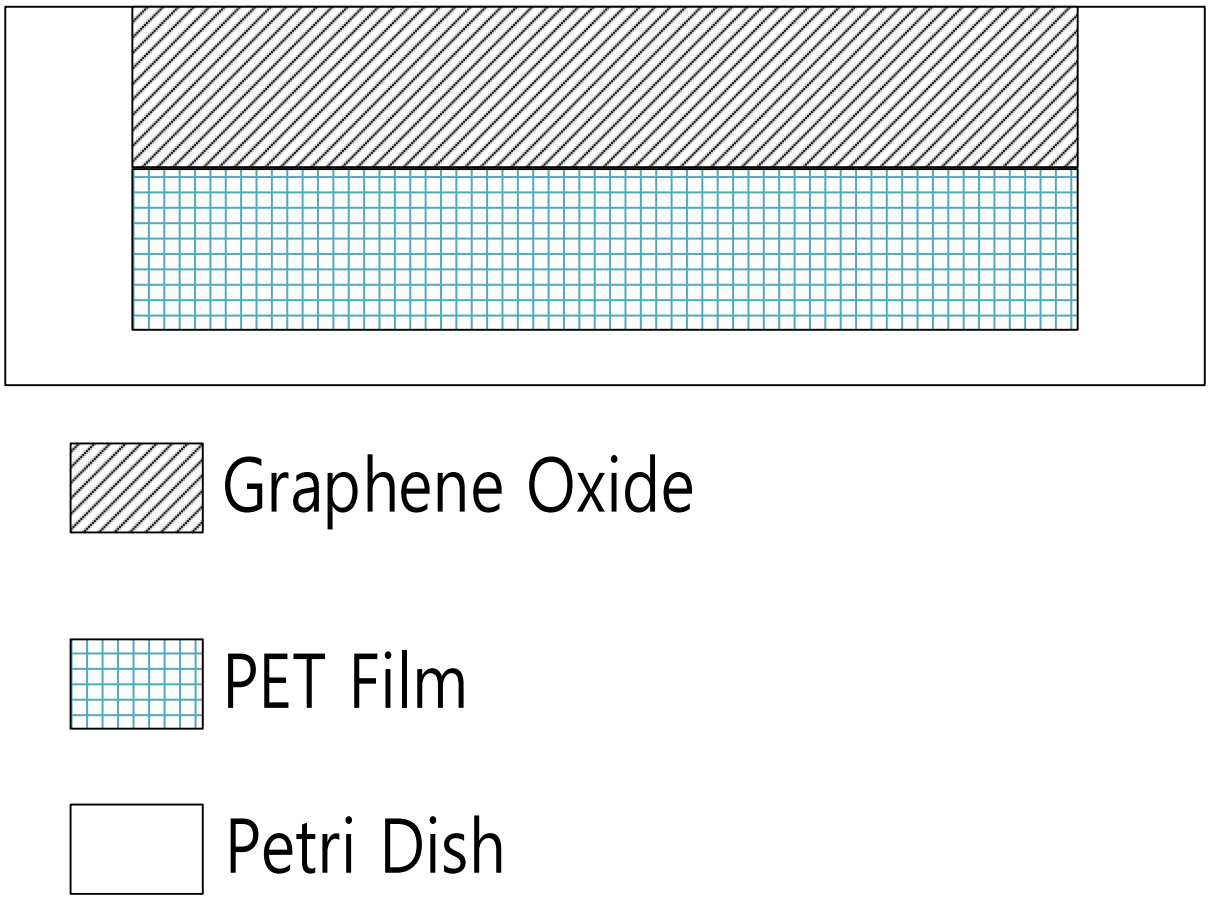

Figure 3-2 Configuration of the graphene Oxide film

First, remove the foreign object by flushing the petri-dish and Pet film. After removing all the water goes to the next step. Production on the film is as shown in Figure 2, which puts PET film in the mold act petri dish. Then, the main surface is not scattered about aggregation the GO solution. However, GO solution and the most closely to prevent from entering through this PET film, even when placed in the GO from the center toward the outer wall sprayed must spread. The whole process is possible to work at room temperature, it is easy to work in the laboratory.

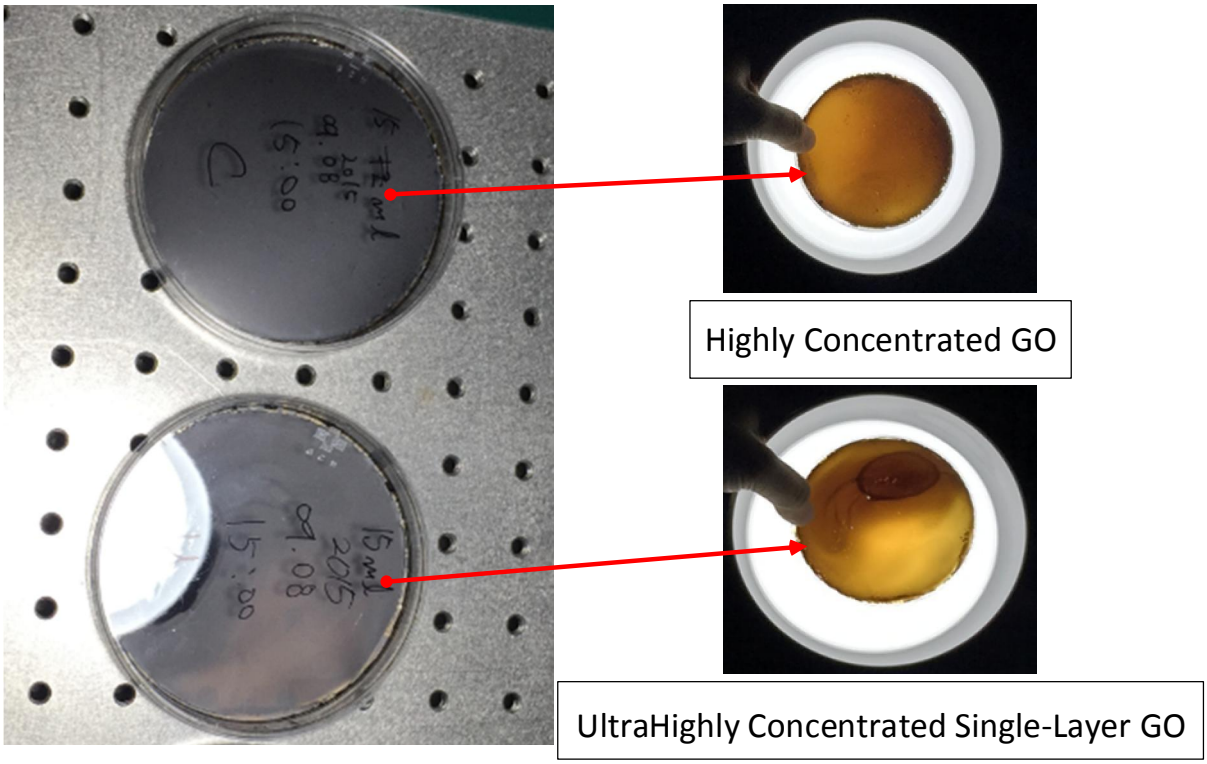

Figure 3-3 GO film after the drying process

Figure 3.3, the film is being made by using the above method. The PET film placed in the Petri dish was cut into a circle. Next, the Ultra-Highly Concentrated Single Layer GO and Highly Concentrated Single Atomic Layer GO applied to each dose of 15ml. Spread on a PET film as a whole GO solution. After then, the drying were put into a vacuum chamber. The dry time was measured and confirmed by eye. The drying time varies according to the amount of GO solution was applied on the PET film. However, what you careful is that this phenomenon occurs in one large aggregation, the amount of the applied solution, petri dish is tilted, so careful is needed, or receives an external vibration is easily generated.

3.2 Production of the test sample a variety of laminate structure

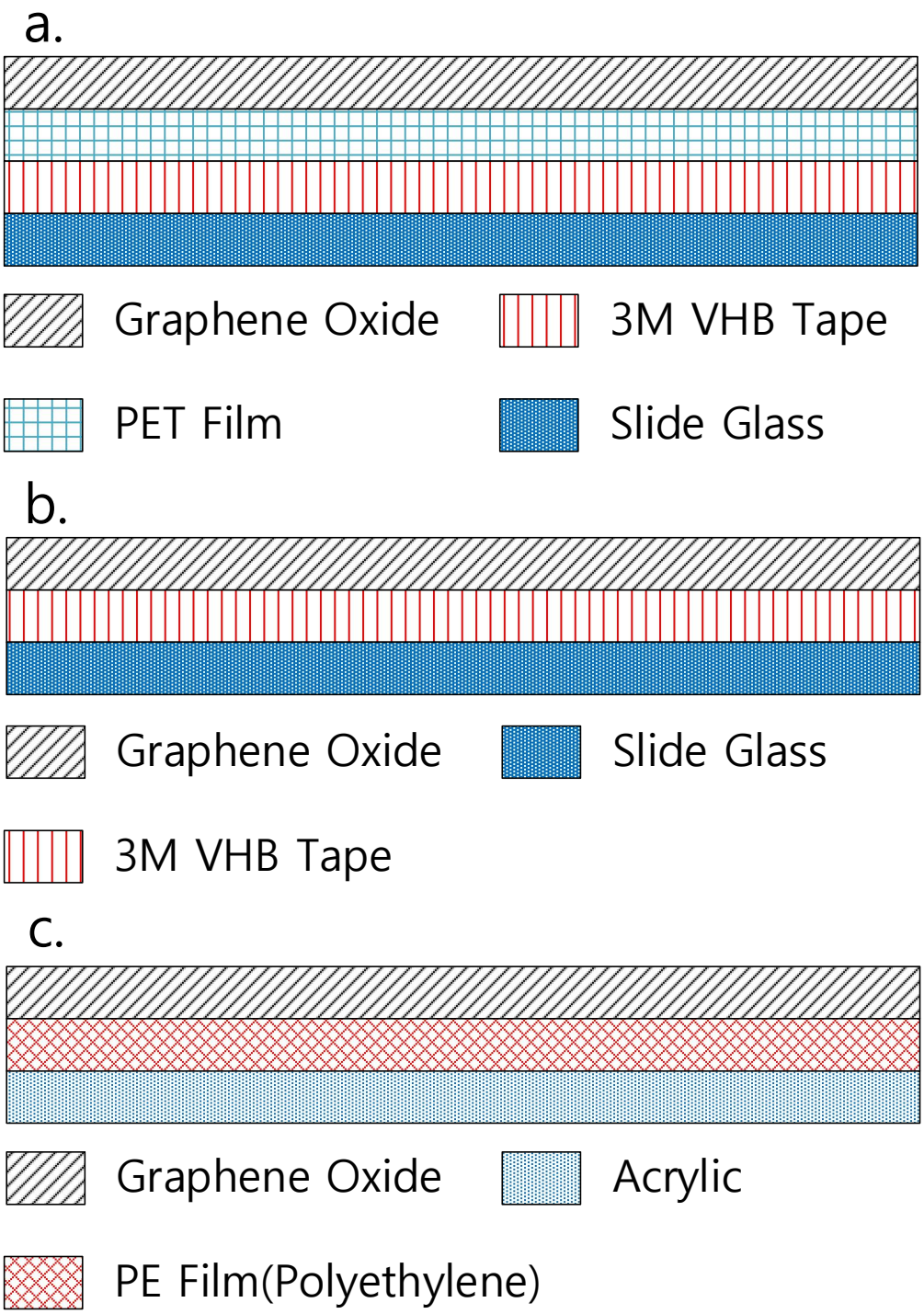

Figure 3-4 The other sample configurations for the GO film at the bottom : (a) Using the slide glass substrate attached to the VHB tape for fixing GO film. (b)Using the slide glass substrate must apply to the GO on the VHB tape, (c)Acrylic should apply to the substrate using the GO on the PE protective film.

Figure 3.4 shows a shape of a sample prepared for the experiment. Each sample differs according to whether or not to use a PE film, a sample (a) was used as a slide glass as substrate in conventional GO film, the sample (b) was applied to the GO given the VHB tape on a slide glass, without the PET film. Sample (c) was applied on the PE film GO, in use as a protective film with good workability by using the acrylic with a substrate.

The reason for the solid substrate is to minimize the change in focus during operation of a laser. VHB Tape was used to secure the GO film without the adhesive on the slide glass. In addition, when the conductive material is coated with a VHB tape piezoelectric property occurs at a higher voltage, when the conversion is expected to be in the GO to rGO similar characteristics have when converted into a conductive material.

3.3 Direct laser scriber work and motion profile

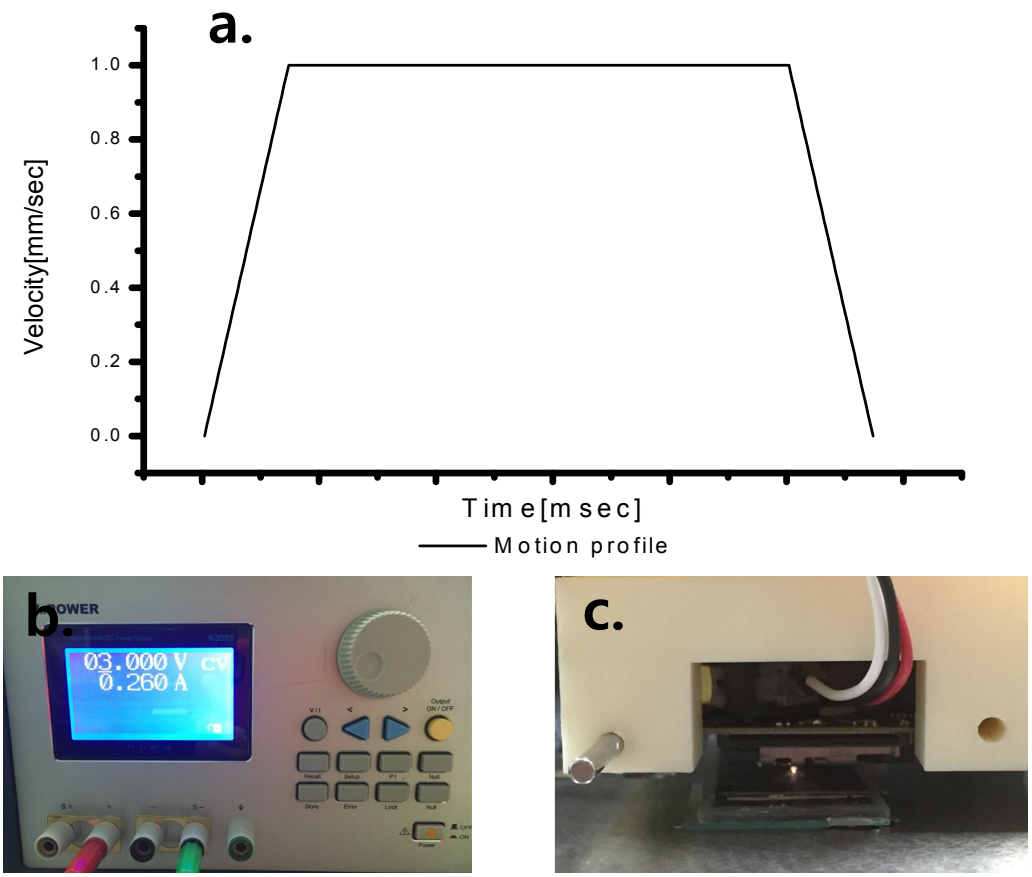

Figure 3-5 Direct operation of the laser scriber: (a) Setting the drive shaft of the main profile, (b) Laser power setting 3.0V, 260mA, (c) the process of converting the GO to rGO.

To drive the direct laser scriber, and sets the motion profile of the drive shaft, as shown in Figure 3.5 (a). The drive speed is set to 1 mm / sec, the acceleration and the deceleration was set to 277.8mm / s. Acceleration and deceleration are set to a uniform motion to the driving speed in about 3.6 msec.

The laser power supply 3 V DC voltage from the power supply as a voltage source, as shown in Figure 3.5 (b). The current is automatically set according to the load. Finally, the driving current was set to 260mA.

Setting the controller to enter the motion profile is prepared in standby mode. Laser is converted into C source switch activates the power supply. Next, operating the drive shaft by operating the laser writing is started, as shown in Figure 3.5 (c).

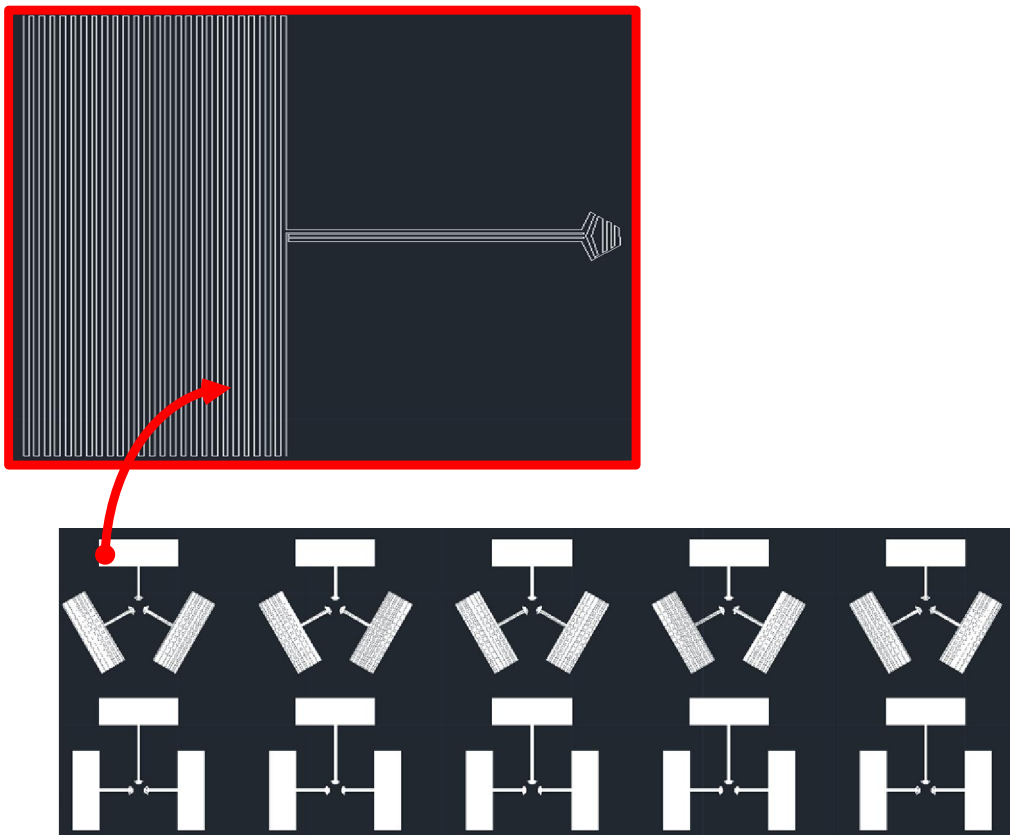

Figure 3-6 Setting the pattern for the write path

Draw a pattern based on a predetermined motion profiles. Figure 3.6 is a CAD drawing input to the NI-Motion Assist to draw a pattern. It can be driven to generate a DXF file using the Auto Cad. Each pattern is designed by the design for using as an electrode.

4. APPLICATION

4.1 Glucose detecting sensor design

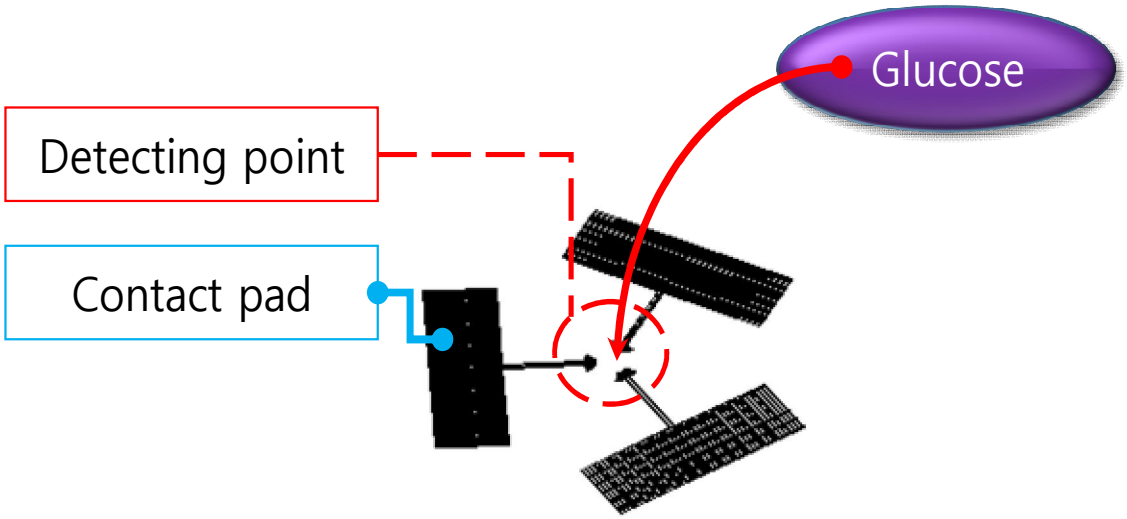

Figure 4-1 Glucose detecting sensor design and configuration

rGO Production of Sensors Using Laser patterns can be done very easily and quickly. The shape of electrode draw with a CAD program and, sensor pattern is made with the laser irradiation. Pattern design is a three-electrode method and the structure of the electrode include "reference electrode(RE)", "working electrode(WE)", "counter electrode(CE)" site. RE and WE performs the potential controlled, CE and WE must check the output for Current flows. Electrode has a radial shape, and the center of the sensing area.

The glucose is the form of sugar and it have the properties of an electrolyte electrically. The injection of glucose to measure the signal detecting point. Glucose that is produced of a material with the ions can be unique to the current output.

4.2 Performance evaluation of the glucose sensor

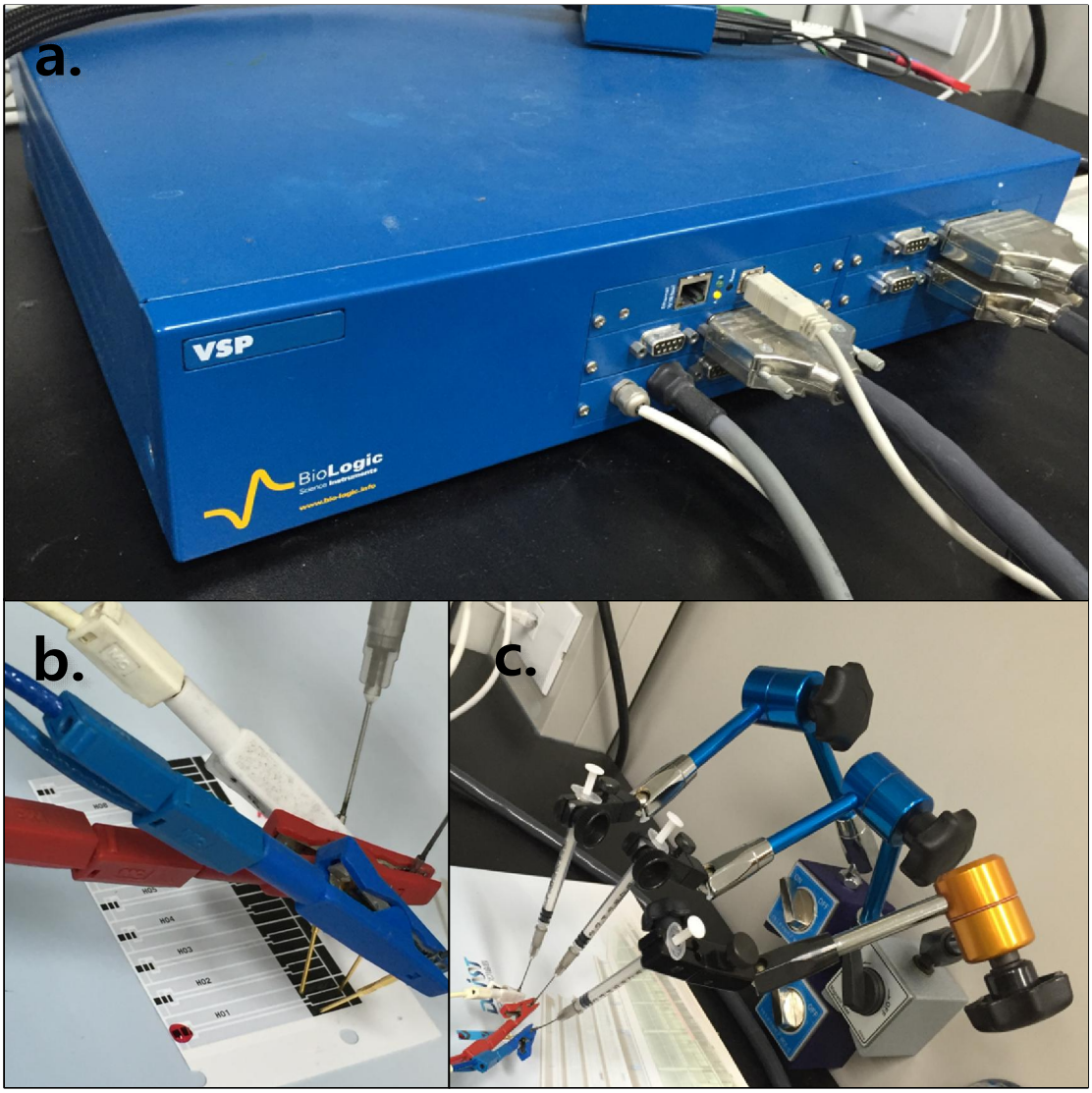

Figure 4-2 Measuring laboratory environment : a. C-V measurement equipment, b. Connecting probe, c. Magnet base for fixing probe

rGO for performance evaluation of the glucose sensor using the measuring device for Figure 4-2(a.) C-V meter (POTENTIOSTAT-VSP, BioLogic). A device for measuring current according to an applied voltage. The probe was connected using the Figure 4-2(b.) gold coating(R-1-RP RSS75RP, IDI). Fixing the probe was used the Figure 4-2(c.) magnet base (WCE-C, BLUE BIRD).

5. RESULTS AND DISCUSSIONS

5.1 Direct laser scriber drawing pattern

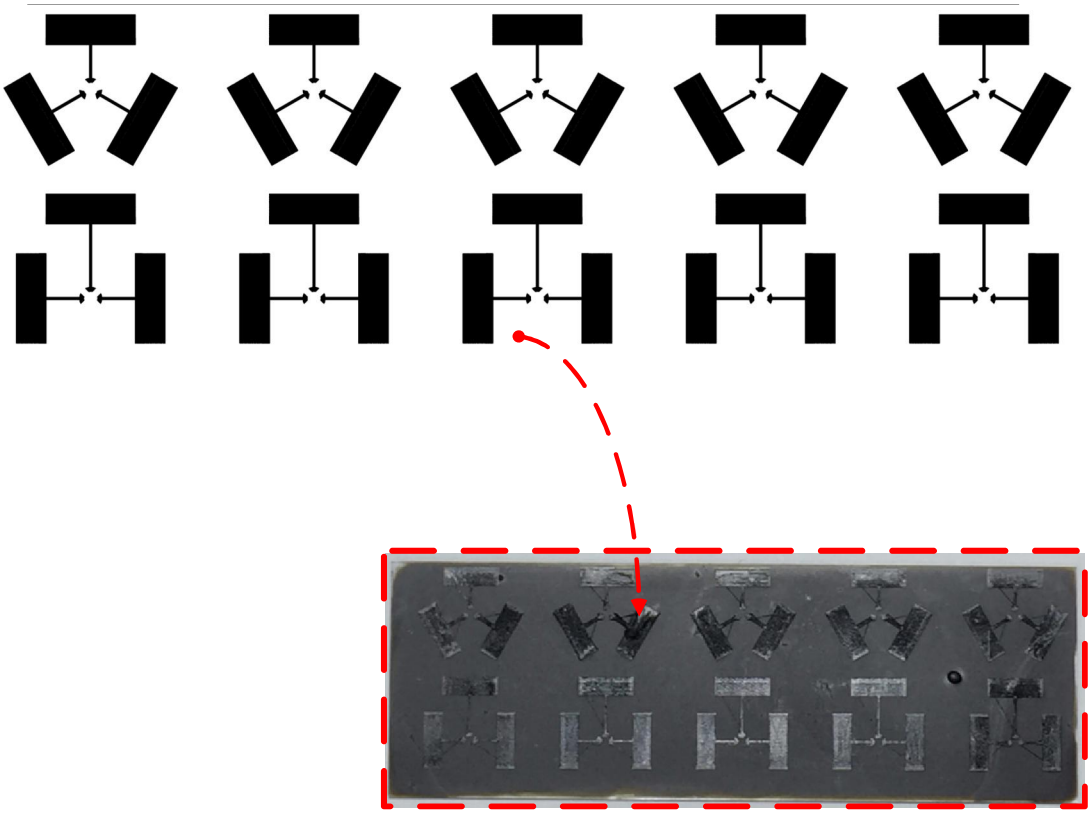

Figure 5-1 Laser scribing results with a CAD drawing pattern

GO in the work space to install the sample was driving a laser scriber. Pattern utilizing the Auto Cad drawings were output as shown in Figure 5-1. The size and shape has been achieved as the contents of the drawing sheet. However, some of the unnecessary pattern is observed by manually operating the laser. Required pattern can be resolved by installing the switching hardware

5.2 Electrical characterization of the rGO

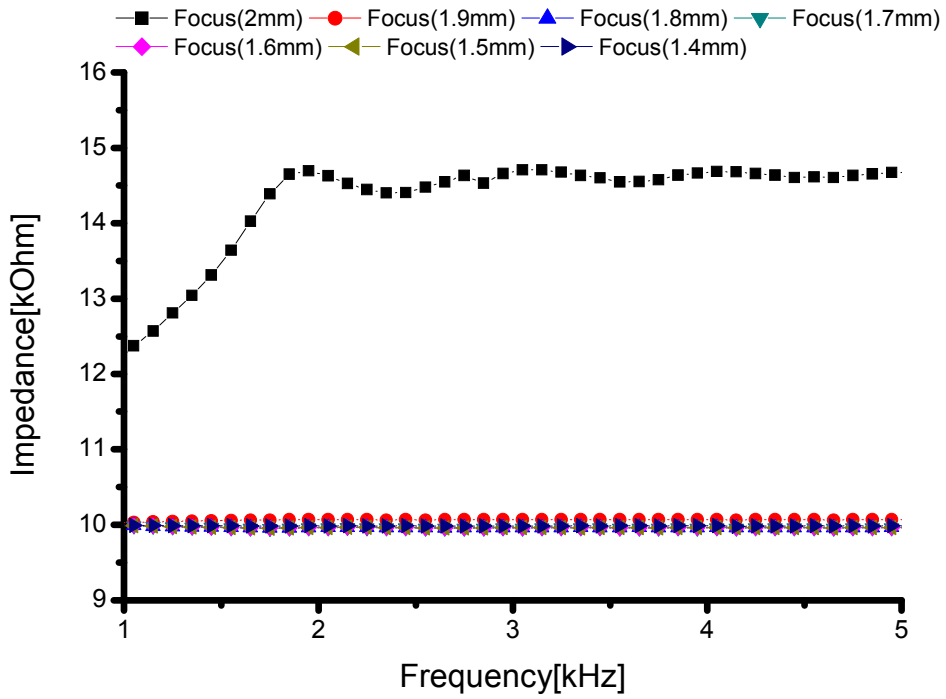

Figure 5-2 Impedance measurement result of 1kHz to 5kHz

It produced the rGO by adjusting the Z-axis location by the motion control. The OPU itself specific to the focusing distance is 1.2 ~ 1.7mm, but the driving conditions was determined by 2 ~ 1.4 mm.

The sample of rGO was measured by impedance measuring chip (ad5933). Parameter was set from 10Hz to 50050Hz, the data interval frequency 50 Hz. Voltage swing range is set to 1Vpp, which was Calibration using 100kΩ resistance.

Results were measured as shown in Figure 5.1 and 5.2. 2mm distance focusing state of the sample showed that the impedance value is not stabilized below 2 kHz. Samples of 1.9mm or less came out both stable impedance measurement range.

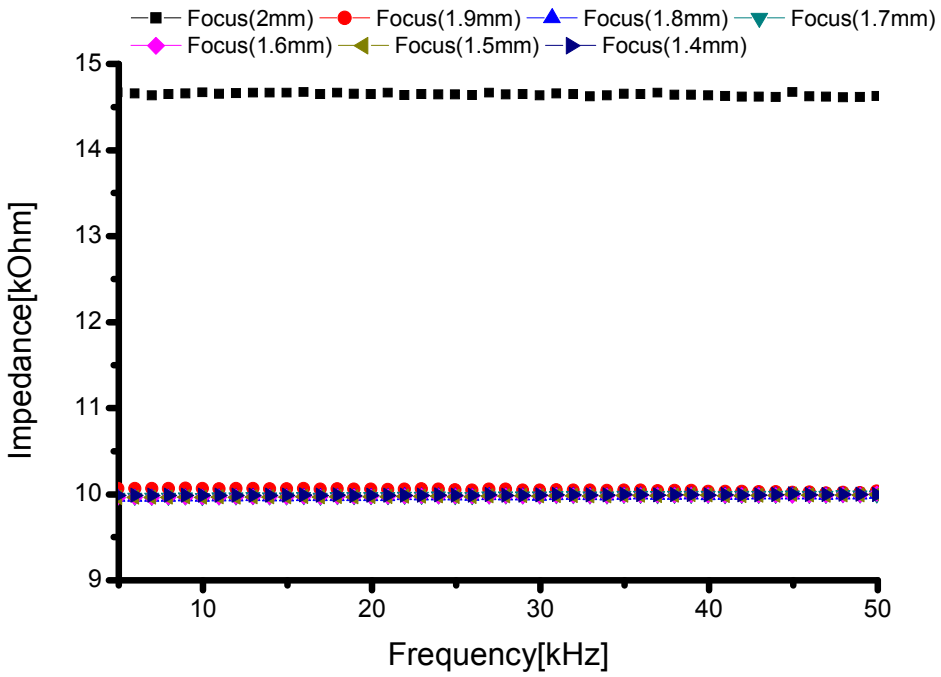

Figure 5-3 Impedance measurement result of 5kHz to 50kHz

Focal length (mm)	Maximum Impedance (kΩ)	Minimum Impedance (kΩ)	Average Impedance (kΩ)	Standard Deviation
2	14.708	10.072	14.598	333.753
1.9	10.075	9.975	10.05	16.063
1.8	10.027	9.955	9.977	10.147
1.7	10.039	9.929	9.975	10.641
1.6	10.029	9.954	9.975	10.43
1.5	10.024	9.953	9.975	10.292
1.4	10.006	9.978	9.989	4.595

Table 5.1 Impedance measurements of the Focal length

Figure 5-2, 5-3 and Table 5.1, it confirmed the trend of impedance measurements based. Samples having a focal distance of 1.9mm or less were fixed resistance ability of the 10 KΩ. Samples of 2mm focal distance has a high resistance of 14kΩ, showed a low stability in a low frequency band until 2kHz.

Impedance of the 2mm focus distance sample has large variation at 14.7 ~ 10.07kΩ. The value compared with the other samples can be seen that in the larger standard deviation out. Samples of 1.9mm or less is the difference between the maximum and minimum values represented by more than 100 Ω. It shows a small change in width of less than 20 standard deviations. Electrical impedance looking at the measurement results, according to the focus distance, and the area for the difference between the values generated, and a region in which the saturation is to occur.

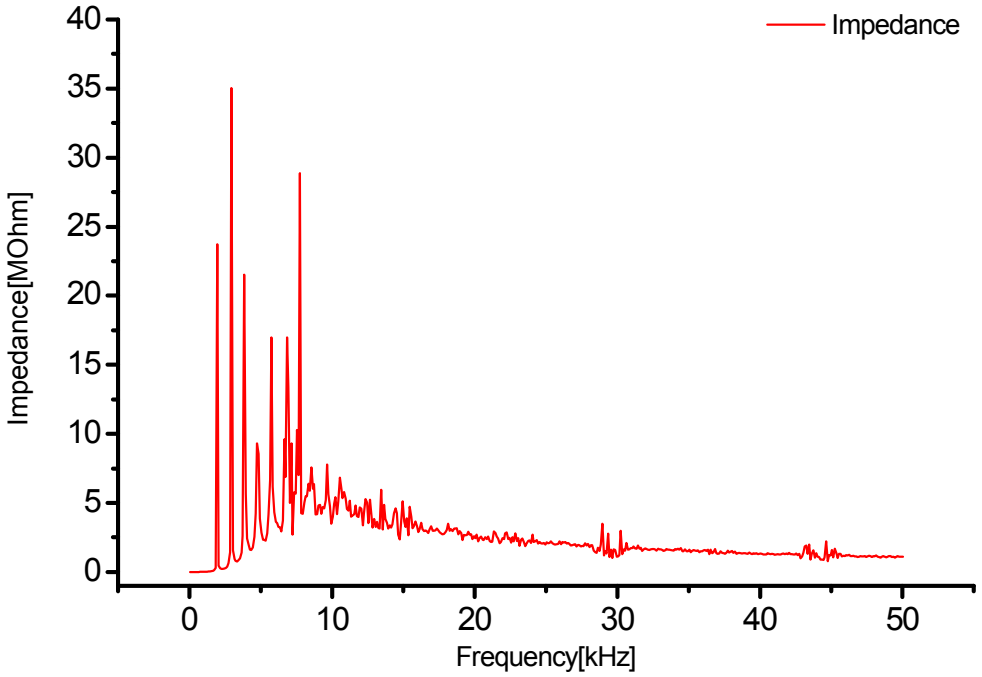

Figure 5-4 Specific case impedance measurement results

This characteristic impedance at a particular sample, as shown in Figure 5-4 were measured. A high impedance peak at a specific low-frequency band is output, and a low impedance in the high frequency region have been output. 38.573 Ω minimum impedance

value at 250 Hz was measured, 35MΩ maximum value measured at 2.95kHz. A crack was observed on the surface in common with the output of a specific sample. The size of the result value of each output was different.

5.3 Optical Imaging Characteristics of rGO

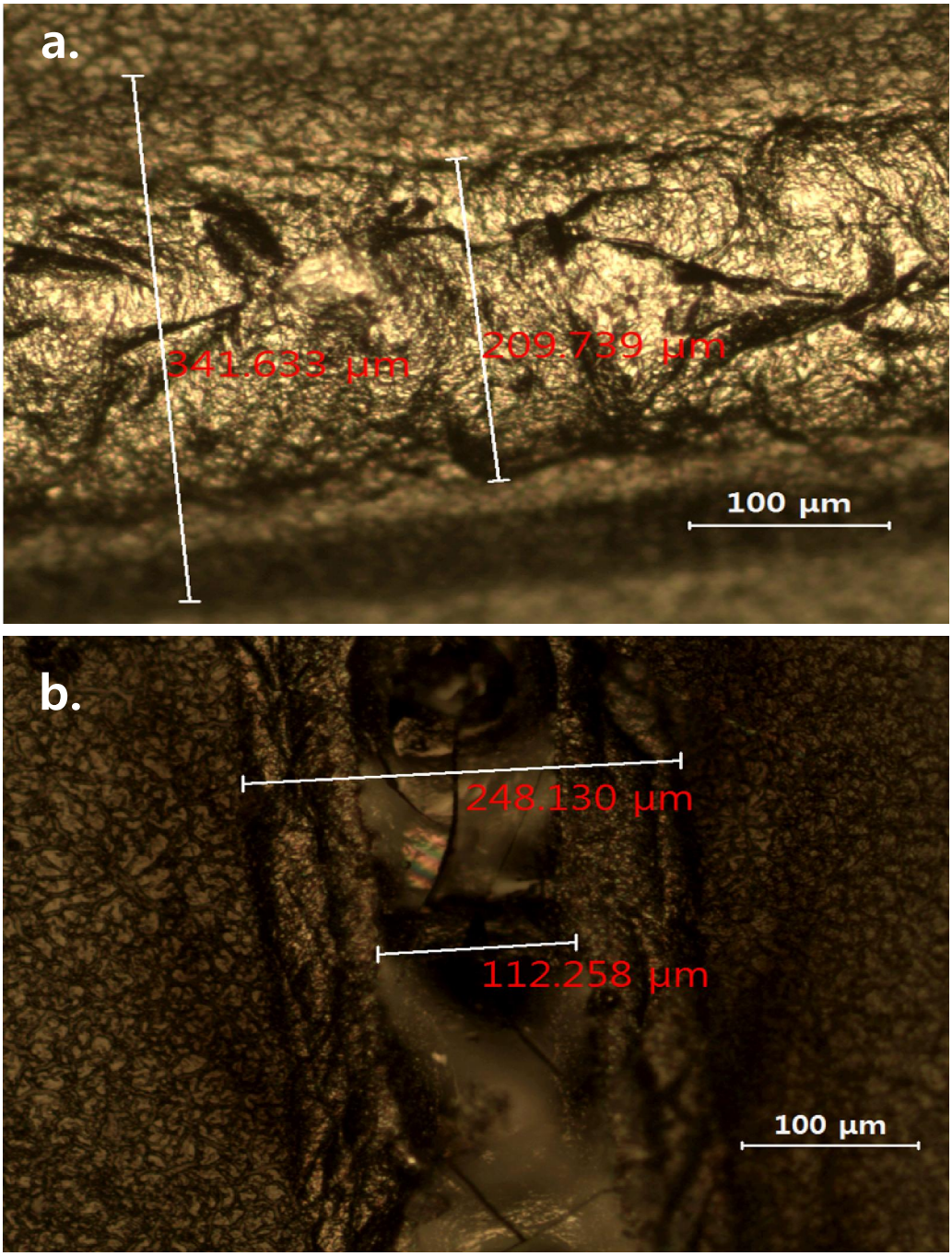

Figure 5-5 rGO measured using an optical microscope photograph : (a) In the form of a general rGO, (b)glossy unusual form on the surface

Figure 5-5 is an optical micrograph (BX60M, OLYMPUS) of each focal distance when (a) 1.7mm and (b) 1.2mm. In the case of (a) rises as the reduction in the bloated GO to rGO. The overall transformation width 345um, the swollen size 216.3um. This phenomenon seems swollen and cleavage. The case of (b) total deformation 260.89um, a fully deformed interval 128.773um. This aspect of the traditional rGO other glassy material is observed in a full deformation range. In addition, cracks were observed in some sections. Samples glass film is formed, additional cracks were caused by an external impact. This phenomenon occurred only in the sample containing the PET film.

5.4 SEM Imaging Characteristics of rGO

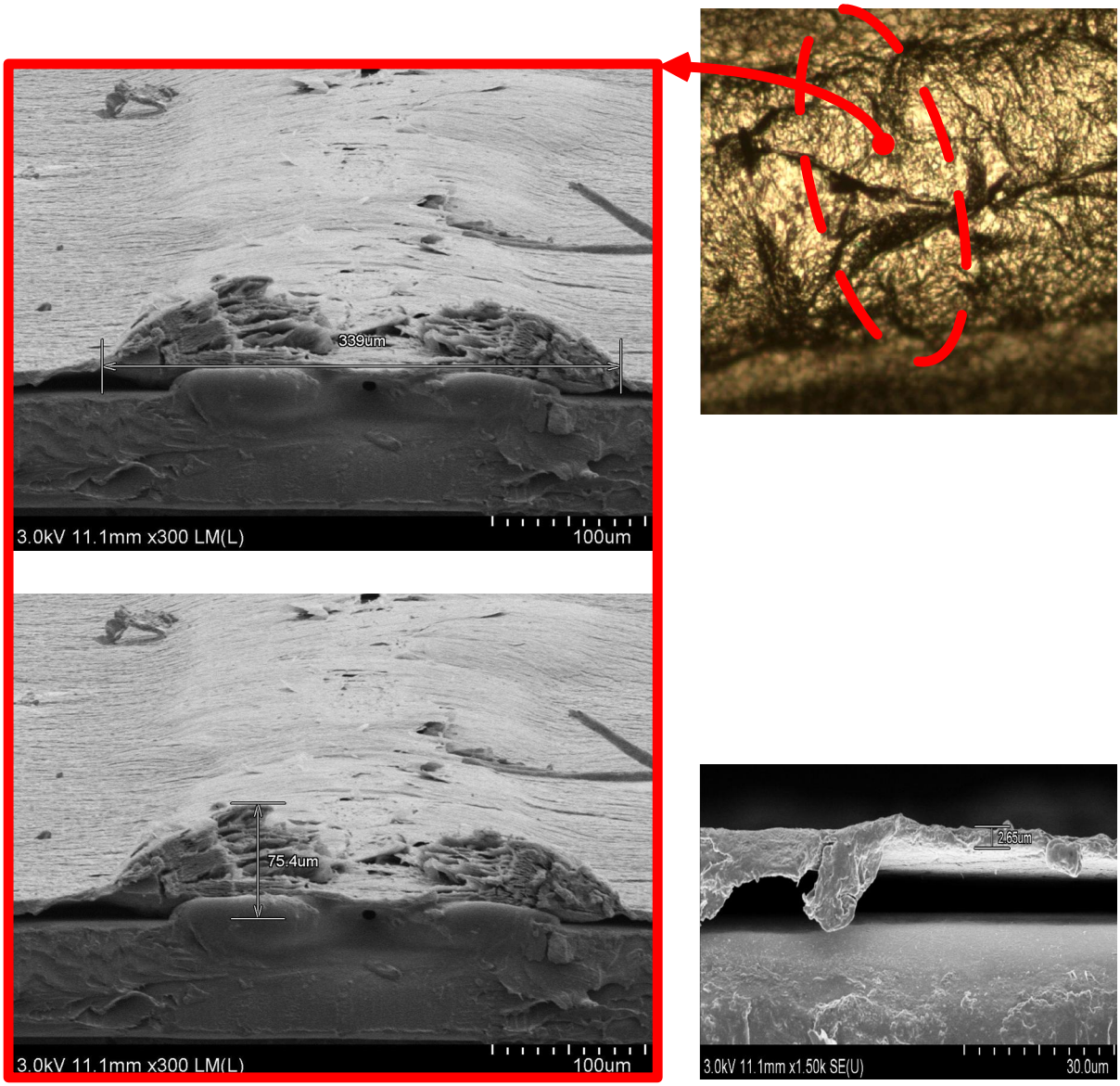

Figure 5-6 SEM image results of rGO: side face measurement of the general rGO

The GO film was confirmed by SEM (SU8020, Hitachi). The results of the image (Figure 5-6) taken from the side, while receiving the laser energy was observed in the form of surface modified with rGO. The laser spot was gone puffed up GO film. Width 339um, the

specifications for the thickness 75.4um been confirmed. GO original thickness of the film was 2.65um.

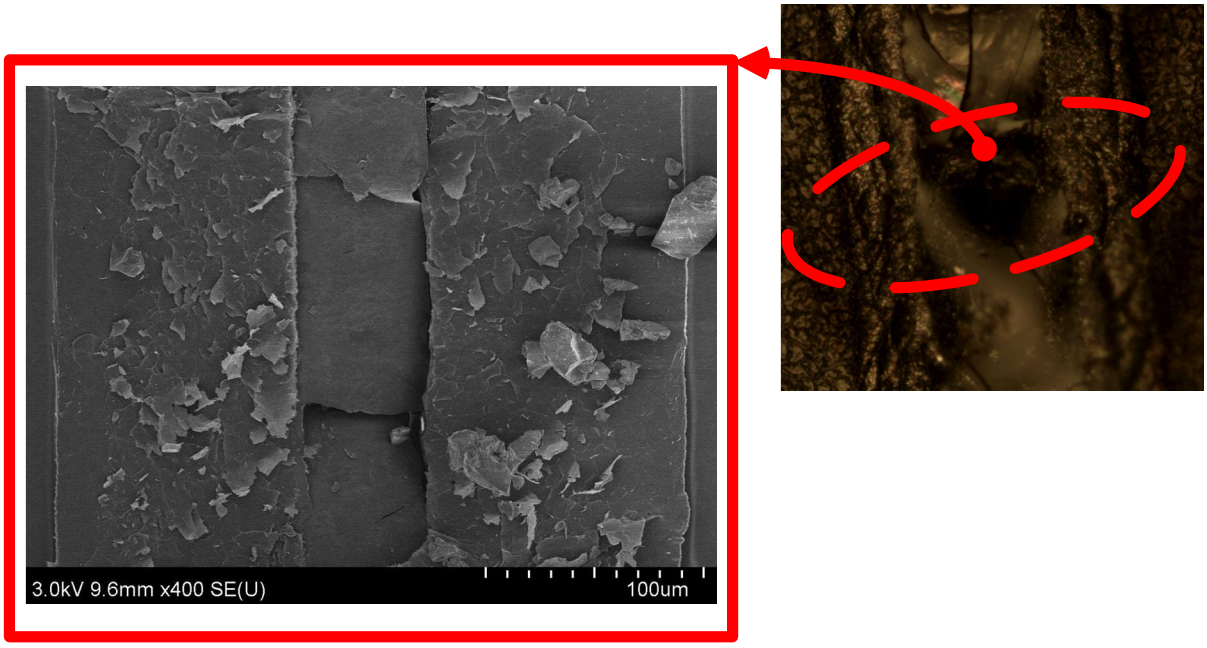

Figure 5-7 SEM image results of rGO: The upper image on the outside of the crystal localized rGO

If the outer surface of the sample crystal screen in a light microscope(Figure 5-7), it was possible to obtain only the top image. The reason was not observed to cause damage to the film side of the sample operation.

5.5 Raman spectroscopy characteristics of rGO

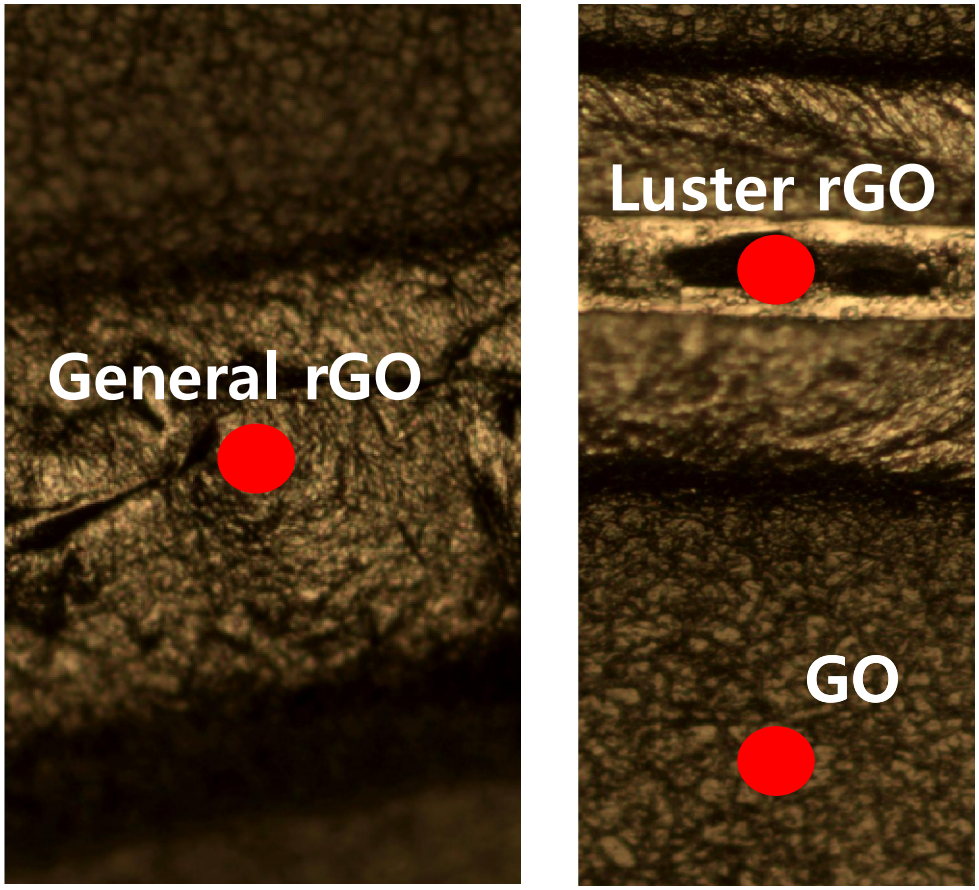

Figure 5-8 Raman spectroscopy measures the position of each sample

The position and focus of the test substance in the Raman analysis is very important. Measuring positions of the sample was set as shown in Figure 5-8 rGO to the check through a Raman spectrometer that is normally formed. It was measured at a center position of each sample. Retention the laser Raman spectroscopy (NICOLET ALMECA XR, Thermo Scientific) measurements, which was set at 780nm.

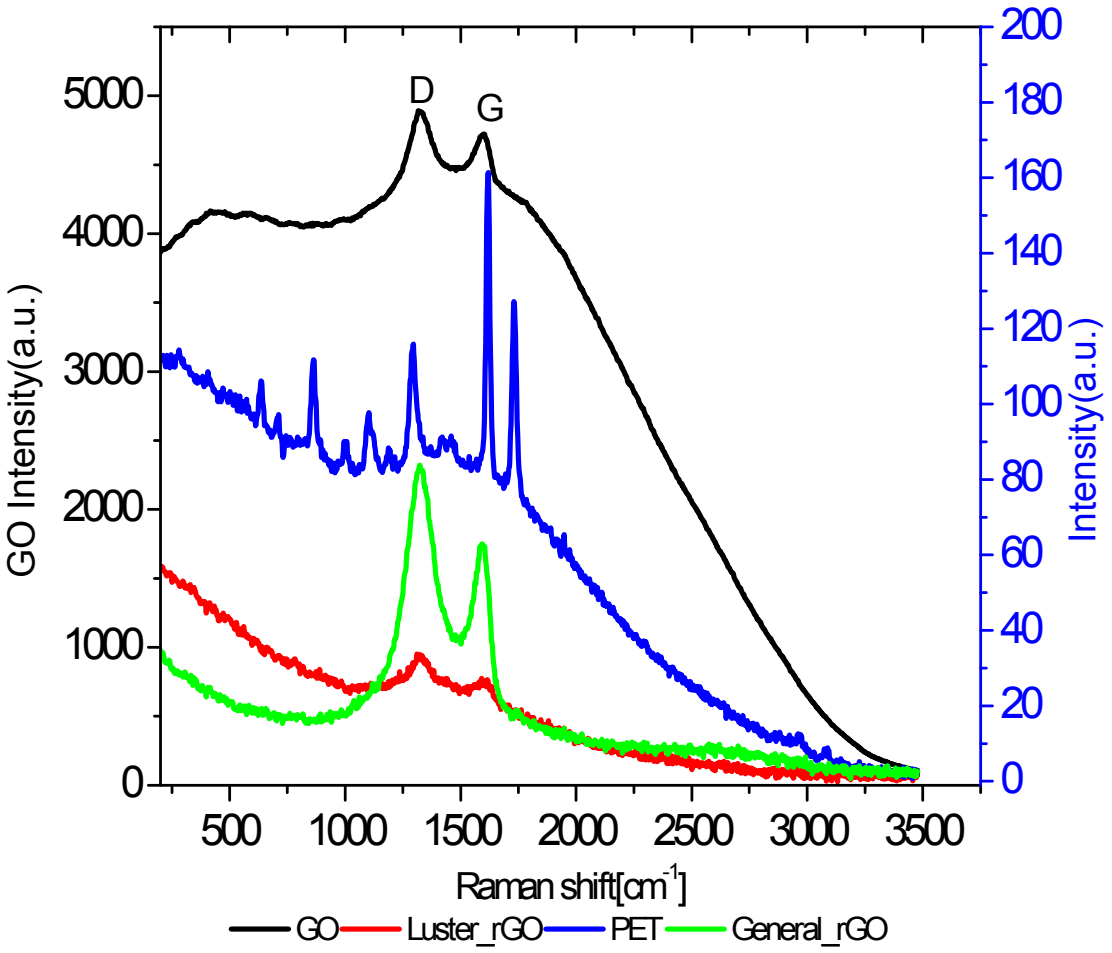

Figure 5-9 Results of Raman spectroscopy measurement

Peak in Raman spectroscopy (Figure 5-9) refers to the specific molecule is vibrating. Report the location of the peak value can tell that the same material. For the GO and rGO the peak value should be generated from the same Raman shift position. This is because the main material contains graphene. So GO, General rGO, Luster rGO both the peak value should be a normal occurrence at the same location.

In other words, laser processing and luster of the surface of the crystallized rGO phenomenon is independent of the PET.

5.6 Glucose detecting sensor result

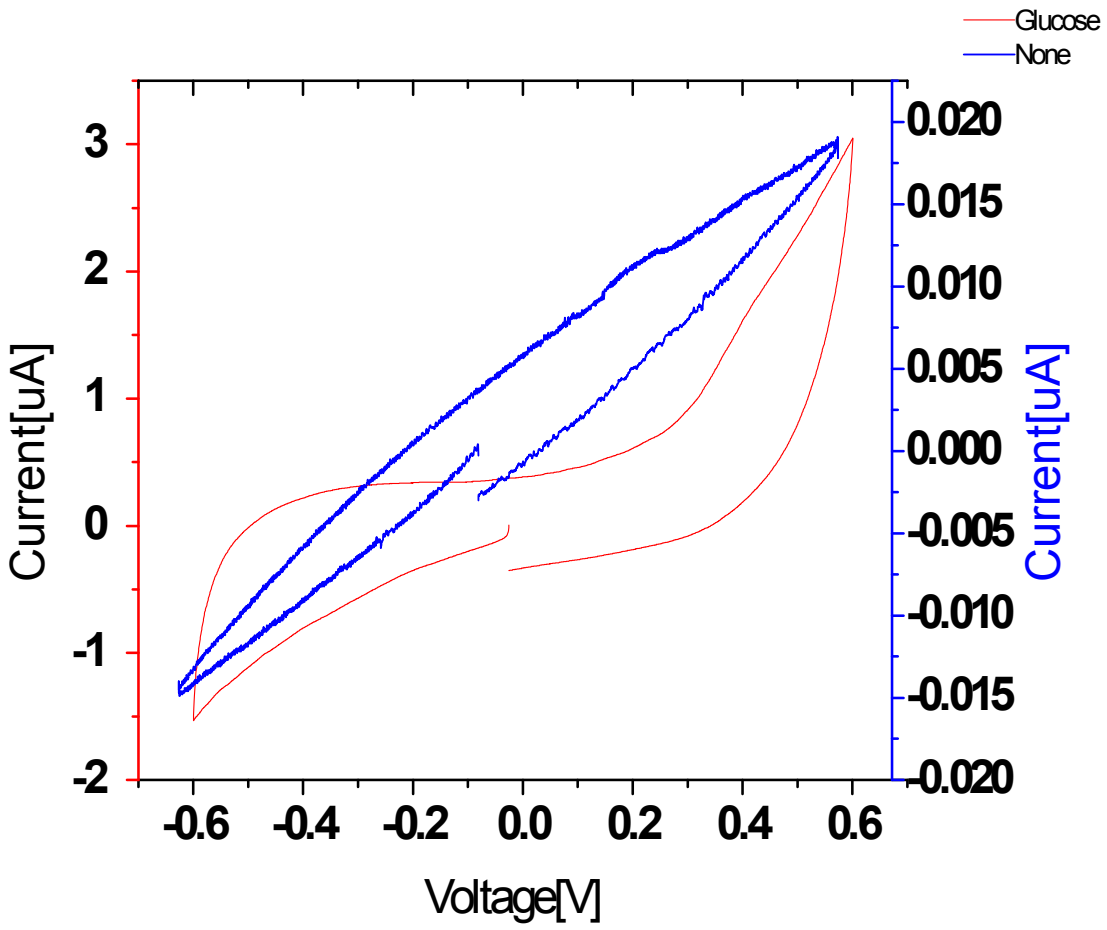

Figure 5-10 Result of C-V measurement

In order to use the CV measurement was set to setting value of the equipment. Scan to set the voltage value (Ew) 10.0mV / sec, and the voltage application range was set -0.6V to +0.6V.

The first measurement is to determine the value without the measured substance to the pattern of the blue color confirmed the graph shown in Figure 5-10. Second, the measured signal for inserting a glucose to sensor pattern. Result of glucose sensing was output as a red graph. Each of the minimum and maximum current output value was up to 3.05uA, at least -1.03uA.

5.7 Discussion

Graphene oxide itself is a non-conductive material. However, as the external energy is injected through the oxide to the reducing process of a conductive material is formed in the graphene. Graphene has excellent thermal conductivity and electrical conductivity.
rGO process is easily done via laser or thermal energy to the GO.

The laser is easy to create micro-sized pattern. In particular, the focal length determine the size of the pattern elements to affect the largest. additional factors are generated according to the type of laser and kind of the lens. In practice, the OPU that used in this study is a built-in voice coil motor to adjust the focal length of the lens, but could not be used. The reason has not be used for a sensor for calculating the integrated focal length. If there will be use of the built-in sensor it is possible, to generate a more accurate pattern.

rGO appears to be proportional to the resistance of the energy injected into the laser. So the focus distance mentioned above is very important. In the impedance measurement it appears to be also used as resistance for common circuit. However, the frequency characteristic of the high impedance output irregular value that occurred in some samples is required to identify the cause of the occurrence. If the guess, cracks occurred in the pattern seems to be major cause. The control cracks generated at this time is not possible, and it is difficult to analyze.

GO is converted to rGO as flame is generated. This phenomenon appears to be caused by meeting a column of the laser and oxygen atom in the reduction process. However, caution is required when prolonged exposure of the laser, because it may also burn GO.

the output value of Luster rGO in raman spectroscopy analysis showed the typical GO properties, there are parts that are not yet clear.Will the strength of the first signal is too low, because PET has not been identified for the spectral characteristics it melts. If these

physical bonds seems a new spectral properties, it is expected because this will be a very interesting experiment. rGO and GO has properties that dissolve flowing water. However, molten rGO to PET film is larger in the resistance to the external liquid. When the electrode is assumed to be produced in consideration of the support layer for rGO.

C-V characteristics has emerged to change the current apparent only rGO. However, the third electrode pattern is due to thermal variations, the resistor was connected to each other. Because it requires an additional study on the distance of the electrode. The nano-particles as a catalyst in conventional studies are expected to have better results come out.

6. CONCLUSIONS

Direct laser scribe was made a pattern to make the application device. Device is successfully driven, by mounting the additional module. The configuration of each module is a feedback servo motor device, a laser unit and indicator for measuring Z-axis distance.

Overall, the stage is configured to operate a commercial motor driver and controller can be easily implemented using the NI Motion Assist. Accuracy was set to 1.3um, it can be increased in accordance with the additional set. However, the accuracy of NANO-level implementation of the structure and the motor has a low performance.

Impedance measurement using the AD5933 chip is available in the 200kHz range. Low price of a measurement system, but it was realized a satisfactory precision. It has an advantage to change and control the possible locations of the drive via an external MCU program. The electrical properties of the materials was through with rGO. In general, the resistance of the rGO samples was measured.10kΩ. The most significant factor contributing to the resistance value was a power of the laser. Power of the laser is the focal length associated with, a change in the width and the resistance value of the pattern occurs according to a fine distance control. The vertical adjustment of the lens is available in the OPU internal VCM, but can not use the feature. If possible, VCM will drive in the future research, the thickness of pattern and quality will be improved.

Due to the swelling phenomenon in the optical properties of the pattern was observed that the increase in surface area. The glass membrane was observed in a particular sample, but does not determine the exact reason. However, it is expected to be caused by the melting shape of the PET film for using support layer. However, the unique characteristics of PET

were not observed in Raman spectroscopy. For the sample glass membrane is formed, it is necessary to analyze the components through additional experiments.

The glucose measurement application saw the possibility of utilization by report of the C-V graphs. In the normal state appears linearity, but current output when glucose injection to pattern is increased 150 times. However, the voltage conditions requires a long time because it is difficult to measure the active time. In subsequent experiments, it is necessary to reduce the voltage conditions times to the measurement in order to measure the instantaneous reactive.

In subsequent studies, it will be making an active laser drive system to the development of integrated hardware that automatically adjustable laser focal length. It will be also create a boundary condition to determine the CV output value corresponding to the graph of the glucose concentration. Next, it will be miniaturized, integrated glucose measurement device using the created conditions.

REFERENCES

[1] H. Booth, "Recent applications of pulsed lasers in advanced materials processing," *Thin Solid Films,* vol. 453, pp. 450-457, 2004.

[2] M. Thiel, J. Fischer, G. Von Freymann, and M. Wegener, "Direct laser writing of three-dimensional submicron structures using a continuous-wave laser at 532 nm," *Applied Physics Letters,* vol. 97, p. 221102, 2010.

[3] A. H. ROLAND MAYERHOFER, CHRISTOPH RÜTTIMANN, "Laser cutting, drilling, and structuring of brittle materials," *Industrial Laser Solution,* 2015.

[4] J.-Y. Cheng, C.-W. Wei, K.-H. Hsu, and T.-H. Young, "Direct-write laser micromachining and universal surface modification of PMMA for device development," *Sensors and Actuators B: Chemical,* vol. 99, pp. 186-196, 2004.

[5] A. K. Geim, "Graphene: status and prospects," *science,* vol. 324, pp. 1530-1534, 2009.

[6] M. J. Allen, V. C. Tung, and R. B. Kaner, "Honeycomb carbon: a review of graphene," *Chemical reviews,* vol. 110, pp. 132-145, 2009.

[7] A. K. Geim and K. S. Novoselov, "The rise of graphene," *Nature materials,* vol. 6, pp. 183-191, 2007.

[8] P. Blake, E. Hill, A. C. Neto, K. Novoselov, D. Jiang, R. Yang*, et al.*, "Making graphene visible," *Applied Physics Letters,* vol. 91, p. 063124, 2007.

[9] K. S. Novoselov, A. K. Geim, S. Morozov, D. Jiang, Y. Zhang, S. a. Dubonos*, et al.*, "Electric field effect in atomically thin carbon films," *science,* vol. 306, pp. 666-669, 2004.

[10] K. S. Kim, Y. Zhao, H. Jang, S. Y. Lee, J. M. Kim, K. S. Kim*, et al.*, "Large-scale pattern growth of graphene films for stretchable transparent electrodes," *Nature,* vol. 457, pp. 706-710, 2009.

[11] I. K. Moon, J. Lee, R. S. Ruoff, and H. Lee, "Reduced graphene oxide by chemical graphitization," *Nature communications,* vol. 1, p. 73, 2010.

[12] S. Park and R. S. Ruoff, "Chemical methods for the production of graphenes," *Nature nanotechnology,* vol. 4, pp. 217-224, 2009.

[13] D. Li, M. B. Mueller, S. Gilje, R. B. Kaner, and G. G. Wallace, "Processable aqueous dispersions of graphene nanosheets," *Nature nanotechnology,* vol. 3, pp. 101-105, 2008.

[14] H. A. Becerril, J. Mao, Z. Liu, R. M. Stoltenberg, Z. Bao, and Y. Chen, "Evaluation of solution-processed reduced graphene oxide films as transparent conductors," *ACS nano,* vol. 2, pp. 463-470, 2008.

[15] C. Berger, Z. Song, X. Li, X. Wu, N. Brown, C. Naud*, et al.*, "Electronic confinement and coherence in patterned epitaxial graphene," *Science,* vol. 312, pp. 1191-1196, 2006.

[16] A. A. Balandin, S. Ghosh, W. Bao, I. Calizo, D. Teweldebrhan, F. Miao*, et al.*, "Superior thermal conductivity of single-layer graphene," *Nano letters,* vol. 8, pp. 902-907, 2008.

[17] C. Gómez-Navarro, R. T. Weitz, A. M. Bittner, M. Scolari, A. Mews, M. Burghard*, et al.*, "Electronic transport properties of individual chemically reduced graphene oxide sheets," *Nano letters,* vol. 7, pp. 3499-3503, 2007.

[18] D. A. Dikin, S. Stankovich, E. J. Zimney, R. D. Piner, G. H. Dommett, G. Evmenenko, *et al.*, "Preparation and characterization of graphene oxide paper," *Nature,* vol. 448, pp. 457-460, 2007.
[19] Y. Zhang, Y.-W. Tan, H. L. Stormer, and P. Kim, "Experimental observation of the quantum Hall effect and Berry's phase in graphene," *Nature,* vol. 438, pp. 201-204, 2005.
[20] H. He, J. Klinowski, M. Forster, and A. Lerf, "A new structural model for graphite oxide," *Chemical Physics Letters,* vol. 287, pp. 53-56, 1998.
[21] P. Touzain, R. Yazami, and J. Maire, "Lithium-graphitic oxide cells part II: High specific surface area graphitic oxide as cathode material for lithium batteries," *Journal of Power Sources,* vol. 14, pp. 99-104, 1985.
[22] S. A. Kliewer, K. Umesono, D. J. Noonan, R. A. Heyman, and R. M. Evans, "Convergence of 9-cis retinoic acid and peroxisome proliferator signalling pathways through heterodimer formation of their receptors," *Nature,* vol. 358, pp. 771-774, 1992.
[23] F. Abraham and M. Goulian, "Diffraction from polymerized membranes: flat vs. crumpled," *EPL (Europhysics Letters),* vol. 19, p. 293, 1992.
[24] T. Hwa, E. Kokufuta, and T. Tanaka, "Conformation of graphite oxide membranes in solution," *Physical Review A,* vol. 44, p. R2235, 1991.
[25] X. Q. Zuo, R. Y. Zhang, B. Yang, G. Li, H. B. Tang, H. J. Zhang, *et al.*, "NiS nanoparticles anchored on reduced graphene oxide to enhance the performance of dye-sensitized solar cells," *Journal of Materials Science-Materials in Electronics,* vol. 26, pp. 8176-8181, Oct 2015.
[26] X. F. Zou and W. J. Zhang, "Preparation and performance of a novel complex material Eu-modified reduced graphene oxide," *Rsc Advances,* vol. 5, pp. 55143-55149, 2015.
[27] H. Tian, Y. Yang, D. Xie, Y.-L. Cui, W.-T. Mi, Y. Zhang, *et al.*, "Wafer-scale integration of graphene-based electronic, optoelectronic and electroacoustic devices," *Scientific reports,* vol. 4, 2014.
[28] T.-M. Cheng, T.-K. Huang, H.-K. Lin, S.-P. Tung, Y.-L. Chen, C.-Y. Lee, *et al.*, "(110)-exposed gold nanocoral electrode as low onset potential selective glucose sensor," *ACS applied materials & interfaces,* vol. 2, pp. 2773-2780, 2010.
[29] S. Sridevi, K. Vasu, S. Asokan, and A. Sood, "Sensitive detection of C-reactive protein using optical fiber Bragg gratings," *Biosensors and Bioelectronics,* vol. 65, pp. 251-256, 2015.
[30] P. W. Barone, R. S. Parker, and M. S. Strano, "In vivo fluorescence detection of glucose using a single-walled carbon nanotube optical sensor: design, fluorophore properties, advantages, and disadvantages," *Analytical chemistry,* vol. 77, pp. 7556-7562, 2005.
[31] G. Spanner and R. Niessner, "New concept for the non-invasive determination of physiological glucose concentrations using modulated laser diodes," *Fresenius' journal of analytical chemistry,* vol. 354, pp. 306-310, 1996.
[32] Z. Cheng, E. Wang, and X. Yang, "Capacitive detection of glucose using molecularly imprinted polymers," *Biosensors and Bioelectronics,* vol. 16, pp. 179-185, 2001.
[33] M. a. Morikawa, N. Kimizuka, M. Yoshihara, and T. Endo, "New Colorimetric Detection of Glucose by Means of Electron-Accepting Indicators: Ligand Substitution of [Fe (acac) 3− n (phen) n] n+ Complexes Triggered by Electron Transfer from Glucose Oxidase," *Chemistry-A European Journal,* vol. 8, pp. 5580-5584, 2002.
[34] S. Wang and L. Wang, "Lanthanide-doped nanomaterials for luminescence detection and imaging," *TrAC Trends in Analytical Chemistry,* vol. 62, pp. 123-134, 2014.

[35] J. Kremeskotter, R. Wilson, D. Schiffrin, B. J. Luff, and J. S. Wilkinson, "Detection of glucose via electrochemiluminescence in a thin-layer cell with a planar optical waveguide," *Measurement Science and Technology,* vol. 6, p. 1325, 1995.
[36] K. E. Toghill and R. G. Compton, "Electrochemical non-enzymatic glucose sensors: a perspective and an evaluation," *Int J Electrochem Sci,* vol. 5, pp. 1246-1301, 2010.
[37] X. Lu, Z. Wen, and J. Li, "Hydroxyl-containing antimony oxide bromide nanorods combined with chitosan for biosensors," *Biomaterials,* vol. 27, pp. 5740-5747, 2006.
[38] A. K. Das and C. R. Raj, "Shape-controlled growth of surface-confined Au nanostructures for electroanalytical applications," *Journal of Electroanalytical Chemistry,* vol. 717, pp. 140-146, 2014.
[39] Y.-Y. Fang, Y.-C. Hsieh, and C.-W. Lin, "NANOSTRUCTURED Pt–Ir NON-ENZYMATIC GLUCOSE SENSORS," *Biomedical Engineering: Applications, Basis and Communications,* vol. 25, p. 1350048, 2013.
[40] F. Lorestani, Z. Shahnavaz, P. Mn, Y. Alias, and N. S. Manan, "One-step hydrothermal green synthesis of silver nanoparticle-carbon nanotube reduced-graphene oxide composite and its application as hydrogen peroxide sensor," *Sensors and Actuators B: Chemical,* vol. 208, pp. 389-398, 2015.
[41] P. Costamagna and S. Srinivasan, "Quantum jumps in the PEMFC science and technology from the 1960s to the year 2000: Part I. Fundamental scientific aspects," *Journal of Power Sources,* vol. 102, pp. 242-252, 2001.
[42] S. Luo, F. Su, C. Liu, J. Li, R. Liu, Y. Xiao*, et al.*, "A new method for fabricating a CuO/TiO 2 nanotube arrays electrode and its application as a sensitive nonenzymatic glucose sensor," *Talanta,* vol. 86, pp. 157-163, 2011.
[43] X. Jia, J. Campos-Delgado, M. Terrones, V. Meunier, and M. S. Dresselhaus, "Graphene edges: a review of their fabrication and characterization," *Nanoscale,* vol. 3, pp. 86-95, 2011.
[44] C. X. Guo, H. B. Yang, Z. M. Sheng, Z. S. Lu, Q. L. Song, and C. M. Li, "Layered graphene/quantum dots for photovoltaic devices," *Angewandte Chemie International Edition,* vol. 49, pp. 3014-3017, 2010.
[45] G. Wang, X. Shen, J. Yao, and J. Park, "Graphene nanosheets for enhanced lithium storage in lithium ion batteries," *Carbon,* vol. 47, pp. 2049-2053, 2009.
[46] X. Huang, X. Qi, F. Boey, and H. Zhang, "Graphene-based composites," *Chemical Society Reviews,* vol. 41, pp. 666-686, 2012.
[47] C. Liu, Z. Yu, D. Neff, A. Zhamu, and B. Z. Jang, "Graphene-based supercapacitor with an ultrahigh energy density," *Nano letters,* vol. 10, pp. 4863-4868, 2010.
[48] M. Zhou, Y. Zhai, and S. Dong, "Electrochemical sensing and biosensing platform based on chemically reduced graphene oxide," *Analytical Chemistry,* vol. 81, pp. 5603-5613, 2009.
[49] *Graphene oxide : physics and applications*. New York: Springer, 2014.

요 약 문

Reduced Graphene Oxide 속성과 응용에 관한 연구

Graphene Oxide(GO)는 일반적인 실험실 혹은 실내 환경에서 작업이 가능한 물질이다. 사용방법도 간단하고 물질 접촉에 따른 독성도 없어서 작업에 아주 용이하다.

자체 특성상 비전도성 특성을 가지지만 열이나 레이저로 가공을 하면 에너지가 가해진 부분에서 환원과정이 발생하여 “Reduced Graphene Oxide(rGO)”가 되면서 전기 전도성을 가진다. 당연히 가공이 안된 면은 비전도성 성질을 그대로 가지면서 두 가지 특성을 동시에 가진 형태로 제작이 가능하게 된다.

입력된 에너지의 양에 따라서 전기적 특성이 달라지는데, 즉 전기적 저항에 차이가 발생한다. 레이저로 같은 위치를 반복하여 가공을 하면 저항 값이 감소하는 특성을 내장 하고 있다. 이러한 특성으로 인하여 반도체 및 메모리에 활용이 되고 있으며, 향후 자체 복합 회로도 제작이 가능할 것으로 보인다.

본 연구에서는 rGO 의 전기적 특성을 재 조명하면서 레이저 에너지의 양과 그에 따른 변화를 보고 활용분야의 확인과 적용을 하려고 한다. rGO 프로세서 과정에서 형태상 부풀어 오르는 성질을 이용하여 분자간의 접촉 면적이 커지는 점을 착안하여 다양한 이온을 혼합하여 특정 센서로 활용을 모색하고 있으며, 특히 의료 분야에서 혈액을 대상으로 하는 센서의 개발을 주로 이루어 지고 있다. 이러한 방법은 기존의 단백질과 각종 효소에 의지하여 제작되고 있는 고가의 반응 물질을 이용한 센서를 물리적인 성질을 이용한 센서로 대처하며, 효소나 단백질 센서의 고질적인 보관과 유통기한의 문제에서 해결의 돌파구를 제시할 것으로 보여진다.

의료분야에서 효소를 사용한 측정분야에서 가장 크게 사용되고 있는 분야는 혈당의 측정 즉 포도당을 측정하는 것으로 당뇨병 환자를 주 타깃으로 하는 분야이다. 본 분야의

센서부분은 위에서 언급한 효소를 사용한 센서로 인하여 생산단가가 높고 유통기한으로 이한 보관의 주의가 필요한 문제가 있다. 때문에 포도당 측정에 있어서 rGO 를 활용한 센서를 제작할 경우 낮은 가격에 대량의 생산의 길이 열릴 것으로 예상이 된다.

이러한 대량 생산을 위해서는 쉽고 빠르게 제작이 가능한 기기가 있어야 하며 본 연구에서는 직접 레이저 가공기를 활용한 방법을 제안한다. 레이저 가공기는 산업용 로봇이나 기타 범용 모터를 활용한 구성으로 제작과 구성이 용이하다.

또한 소형화를 통해 센서를 개인적으로 프린터에서 인쇄를 하듯 센서도 필요할 때 가정이나 장소에 구애가 없이 제작이 가능하며, 재료의 독성이 없으므로 폐기나 보관시 제약이 없어야 한다. 이러한 보관과 폐기의 용이성에 쉬운 제작과 대량 생산 및 자가 생산으로 접근성을 위한 여러 실험과 환경조건을 제시를 해야 한다.

본 연구는 독성이 없는 GO 를 사용하여 rGO 를 생성하는 방법을 쉽고 빠르게 제작이 가능한 레이저 스크라이버의 사용과 함께 제작된 rGO 의 성능 분석과 포도당의 직접적인 측정을 통하여 포도당측정의 가능여부 및 향후 개선안과 방향을 소개할 것이다.

핵심어: 그래핀, 산화그래핀, 환원 산화그래핀, 포도당, 혈당, 당뇨, 레이저 가공기

APPENDIX

Ad5933 Source code

```
/* ======================================
 * PRJ_AD5933 OPeration
 * make by WOO Seong Yong
 * AD5933_4_fun.h
 * ======================================
*/
//#include <deviece.h>
#include <project.h>
#include "DATA_MAPPING.h"
#include "I2C_1_CMD.h"
#include "User_math.h"
#include <math.h>
#define Ref_resister          (100000.0)
#define WR_BUFFER_SIZE        (10u)
#define MCLK                  (16000000.0)
#define data_buff             (200)
#define COR_val               (1)
int NUMo_cycle          =     (0);//측정사이클 수, 안정화시간 계산에 사용 및
측정반복에 사용_Input val

uint8 Start_Q[4]                =      [49];//1, 2, 3만 사용
uint8 Inc_Q[4]                  =      [28];//1, 2, 3만 사용
uint8 ItoC[2]                   =      {0};
//double Manitude_arr[data_buff]     =      {0};//code
//double Sweep_phase_arr[data_buff]  =      {0};
//long double Admittance_arr[]  = {0};//(1/Impedance)
//long Frequency_arr[data_buff]      =      {0};
//int R_data[data_buff]             =      {0};
//int I_data[data_buff]             =      {0};
double G_factor[data_buff]        =      [28];
double Imp_data_arr[data_buff]    =      [28];
//extern uint8 W_Data_flg = 0; // data flg


//int    Tem_R_data               = 0;
//int    Tem_I_data               = 0;
int    Cal_Gf_flg               = 0;
//double  Magnitude               = 0.0 ;
//double  Sweep_phase             = 0.0 ;
double  Mid_phase               = 0.0 ;
double  Z_data                  = 0.0 ;
double  Imp_data                = 0.0 ;
double  Gain_f                  = 0.0 ;
double  Cal_Gain_f              = 0.0 ;
double  MT_Gain_f               = 0.0 ;
uint8   error                   = 0;
double  Feed_back_res           = 0.0 ;
int    array_num                = 0   ;
uint32 Start_Q_val              = 0   ;
uint32 Inc_Q_val                = 0   ;
//int loop_cnt                  = 0   ;
```

```
//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@

int8 CMD_ARR[13]          ={0};

int ST_HZ_Q              = 0;
int INC_HZ_Q             = 0;
int NUM_OF_INCVAL         = 0;

uint8 MOD_num             = 0;
uint8 SETTLING_TIME_VAL    = 0;
uint8 VRANGE_VAL           = 0;
uint8 PG_GAIN_VAL          = 0;
int8 ERROR_sys             = 0;

//+++++++++++++++++++++++++AD5933 control+++++++++++++++++++++++++++++++
void    CMD_Repeat_sweep();
uint8   Check_valid_data();
void    Init_and_Start_Frequency_Sweep();
void    Make_st_f_and_inc_f(uint32 st_f, uint32 inc_f);
uint8   CMD_Out_V_Range(uint8 Vr);
uint8   CMD_Pggain(uint8 Gain);
double  CMD_Temp();
void    Pow_down();

//++++++++++++++++++++++++CAL_data_fun+++++++++++++++++++++++++++++++++
void    calibration_G_factor(uint32 St_Hz, uint32 Inc_Hz, unsigned int
Num_Of_Inc, unsigned int Settime, uint8 V_Range, uint8 Pg_gain, uint8
G_flg);
uint32  Cal_Frequency_code(uint32 Start_F);
double    Cal_Imp_and_Phase_val(int cal_flg);
//void    Result_arr(int Num_of_Inc_Q, int stQQQ, int Inc_QQQ, int
cal_flg);
//void    Cal_R_val();
//void    Cal_I_val();
double  Data_cal_Phase(double img, double real);

//++++++++++++++++++++++++unit_cov_fun+++++++++++++++++++++++++++++++++
long    Conv_8bit_to_16bit(uint8 Lower, uint8 Upper);
void    Div_Start_Q(uint32 data);
void    Div_Inc_Q(uint32 data);
void    Conv_int_to_char(unsigned int data);
uint32  Data_proc(uint8 data1, uint8 data2);
unsigned int Conv_char_to_int(uint8 first, uint8 last);
double  Conv_1byte_to_2byte(uint8 Lower, uint8 Upper);
//++++++++++++++++++++++++Check_Sweep_CMP++++++++++++++++++++++++++++++
void    Check_Sweep_CMP(int CMD_code, unsigned int arr_nu, int stQQ, int
Inc_QQ);
//uint8   CMD_resweep(uint8 num);

//++++++++++++++++++++++++init_AD5933+++++++++++++++++++++++++++++++++
void init_AD5933(uint32 St_Hz, uint32 Inc_Hz, unsigned int Num_Of_Inc,
unsigned int Settime, uint8 V_Range, uint8 Pg_gain);

//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@

//++++++++++++++++++++++++init_AD5933+++++++++++++++++++++++++++++++++
```

```
/*
void init_AD5933(uint32 St_Hz, uint32 Inc_Hz, unsigned int Num_Of_Inc,
unsigned int Settime, uint8 V_Range, uint8 Pg_gain);
*/
//============================================================================
void init_AD5933(uint32 St_Hz, uint32 Inc_Hz, unsigned int Num_Of_Inc,
unsigned int Settime, uint8 V_Range, uint8 Pg_gain)
{
    //uint8 reg_data[2]={0};
    //uint8 size = 0;
    uint8 tem1 = 0;
    uint8 tem2 = 0;
    //printf("\n\r Init AD5933");
    //I2C_1_MasterClearStatus();

    //write_I2C_data(CNT_reg_2, CMD_Reset);//(0x81)//D7~D0 //(0x08)//D7~D0
```
리셋모드

```
    Start_Q_val = St_Hz                     ;
    Inc_Q_val   = Inc_Hz                    ;
    Mid_phase   = (St_Hz*1.0)+((Inc_Hz* Num_Of_Inc*1.0)/2.0);
    array_num   = Num_Of_Inc                ;//Num_Of_Inc+1                ;

    //printf("\n\r Start_Q_val = %ld",Start_Q_val);
    //printf("\n\r Inc_Q_val = %ld",Inc_Q_val);
    //printf("\n\r Mid_phase = %g",Mid_phase);
    //printf("\n\r array_num = %d",array_num);

    Make_st_f_and_inc_f(St_Hz, Inc_Hz);



    //Write Start Frequency
    Write_I2C_test(AD5933_add, ST_Hz_1, Start_Q[1]);//(0x82)//D23~D16
    Write_I2C_test(AD5933_add, ST_Hz_2, Start_Q[2]);//(0x83)//D15~D8
    Write_I2C_test(AD5933_add, ST_Hz_3, Start_Q[3]);//(0x84)//D7~D0

    //printf("\n\r Start Frequency = 0x%X %X %X",Read_I2C_test(AD5933_add,
ST_Hz_1), Read_I2C_test(AD5933_add, ST_Hz_2), Read_I2C_test(AD5933_add,
ST_Hz_3));
    CyDelay(1);

    //Write Number of Increments
    Conv_int_to_char(Num_Of_Inc+1);
    Write_I2C_test(AD5933_add, NUMoINC_1, ItoC[0]);//(0x88)//D15~D8
    Write_I2C_test(AD5933_add, NUMoINC_2, ItoC[1]);//(0x89)//D7~D0
    //printf("\n\r Number of Increments = 0x%X %X",Read_I2C_test(AD5933_add,
NUMoINC_1), Read_I2C_test(AD5933_add, NUMoINC_2));
    CyDelay(1);

    //Write Increments Frequency
    Write_I2C_test(AD5933_add, INC_Hz_1, Inc_Q[1]);//(0x85)//D23~D16
    Write_I2C_test(AD5933_add, INC_Hz_2, Inc_Q[2]);//(0x86)//D15~D8
    Write_I2C_test(AD5933_add, INC_Hz_3, Inc_Q[3]);//(0x87)//D7~D0
    //printf("\n\r Increments Frequency =
0x%X %X %X",Read_I2C_test(AD5933_add, INC_Hz_1), Read_I2C_test(AD5933_add,
INC_Hz_2), Read_I2C_test(AD5933_add, INC_Hz_3));
    CyDelay(1);
```

```
    //Write Number of settlingtime
    Conv_int_to_char(Settime);
    Write_I2C_test(AD5933_add, NUMoSETIME_1,
ItoC[0]);//(0x8A)//D15~D8((0x00)1배냐 (2배냐 4배냐 선택)
    Write_I2C_test(AD5933_add, NUMoSETIME_2,
ItoC[1]);//(0x8B)//D7~D0(숫자만)
    //printf("\n\r Number of settlingtime =
0x%X %X",Read_I2C_test(AD5933_add, NUMoSETIME_1), Read_I2C_test(AD5933_add,
NUMoSETIME_2));
    CyDelay(1);

   //Write Ext_osc
    Write_I2C_test(AD5933_add, CNT_reg_2, CMD_ex_Osc);//(0x81)//D7~D0
//(0x08)//D7~D0 Enable external Oscillator
    //printf("\n\r Enable external Oscillator = 0x%X",
Read_I2C_test(AD5933_add, CNT_reg_2));//Enable external Oscillator
    //CyDelay(50);

    //Write Standby mode
    Write_I2C_test(AD5933_add, CNT_reg_1, CMD_standby_mo);//(0x80)//D15~D8
//(0xB0)//D15~D12 대기모드
    //printf("\n\r Standby mode= 0x%X",Read_I2C_test(AD5933_add,
CNT_reg_1));


    //write PGAgain and Output Voltage Range
    tem1 = CMD_Out_V_Range(V_Range);
    tem2 = CMD_Pggain(Pg_gain);
    Write_I2C_test(AD5933_add, CNT_reg_1, (tem1+tem2));//(0x80)//D15~D8
    //printf("\n\r PGAgain and Output Voltage Range =
0x%X",Read_I2C_test(AD5933_add, CNT_reg_1));

    CyDelay(2);
    //Init_and_Start_Frequency_Sweep();
    //printf("\n\r Init CMP %d", (int)tem1);
    Write_I2C_test(AD5933_add, CNT_reg_1, CMD_St_Q_Init);         //주파수
설정
    //printf("\n\r Q_init? 0x10? = 0x%X ",Read_I2C_test(AD5933_add,
CNT_reg_1));
    CyDelay(2);
    Write_I2C_test(AD5933_add, CNT_reg_1, CMD_Q_sweep_start);   //주파수
진동시작
    //printf("\n\r Q_start? 0x20? = 0x%X ",Read_I2C_test(AD5933_add,
CNT_reg_1));

}
//=========================================================================
//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

//++++++++++++++++++++++++Check_Sweep_CMP++++++++++++++++++++++++++++++++
/*
void Check_Sweep_CMP(int CMD_code, int arr_nu, int stQQ, int Inc_QQ);
*/
//=========================================================================
void Check_Sweep_CMP(int CMD_code, unsigned int arr_nu, int stQQ, int
```

```
Inc_QQ)
{//main loop prg
    uint8 read_data = 0;
    unsigned int Num_of_inc_1 = 0;
    uint8 loop_inc = 0;
    uint8 run_tem = 0;
    int Q_display = 0;
    Num_of_inc_1 = arr_nu;
    if(CMD_code == 1)
    {
        CyDelay(10);
        //read_data = Read_I2C_test(AD5933_add, Sts_reg_1);
        //printf("\n\r read_data = %d", read_data);
        //printf("\n\r Num_of_inc = %d", Num_of_inc_1);
        run_tem = Read_I2C_test(AD5933_add, Sts_reg_1)&Cmp_q_shoot;
        while(!(run_tem))
        {
            read_data = Read_I2C_test(AD5933_add, Sts_reg_1);
            while(!(read_data &(Eff_val_ch)))//Eff_val_ch(0x02)?  Loop End?
            {
                read_data = Read_I2C_test(AD5933_add, Sts_reg_1);
                 //printf("\n\r CMP?? read_data = %x", read_data);
            }

                //printf("\n\r good !! if read_data = %x", read_data);
                //Cal_Imp_and_Phase_val(Cal_Gf_flg);
                //Result_arr(loop_inc, stQQ, Inc_QQ, Cal_Gf_flg);
                   if(Cal_Gf_flg == 0)
                   {//게인팩터 계산 전
                       Q_display = stQQ + (Inc_QQ*loop_inc);
                       G_factor[loop_inc] = Cal_Imp_and_Phase_val(Cal_Gf_flg);
                       printf(" [%d]Hz G_factor[%d] = %g",Q_display, loop_inc,
G_factor[loop_inc]);
                   }
                   else
                   {//게인팩터 계산 후
                       Q_display = stQQ + (Inc_QQ*loop_inc);
                       Imp_data_arr[loop_inc] =
Cal_Imp_and_Phase_val(Cal_Gf_flg);
                       printf("[%d]Hz ",Q_display);
                   }
                CyDelay(10);
               // printf("\n\r good !! array_num = %d loop_inc =%d Num_of_inc
= %d",arr_nu, loop_inc, Num_of_inc_1);
                loop_inc++;
                Num_of_inc_1--;

                //printf("\n\r Loop_inc = %d", loop_inc);
                Write_I2C_test(AD5933_add, CNT_reg_1, CMD_Q_Inc);//주파수 증가
                //printf("\n\r ");
                CyDelay(5);
                run_tem = Read_I2C_test(AD5933_add, Sts_reg_1)&Cmp_q_shoot;
        }
        //printf("\n\r good !! array_num = %d loop_inc =%d Num_of_inc
= %d",arr_nu, loop_inc, Num_of_inc_1);
        if(run_tem!=0)
        {//loop end
            array_num = 0;
```

```
            //printf("\n\r good !! All cle %d", loop_inc);
            //printf("\n\r G_F = %g", Cal_Gain_f);
            Pow_down();
        }
    }//if
    else
    {
        error = 0;
        //printf("\n\r NO CMD");
    }//else
}
//=========================================================================
//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
/*uint8 CMD_resweep(uint8 num)
{
    uint8 read_data = 0;
    //read_data = num & (Eff_val_ch); //while 조건문 갱신
    if((num & (Eff_val_ch))|(num & (Cmp_q_shoot)))
    {
        printf("\n\r good~!! get valid data ");
        return 0;
    }
    else
    {
        //printf("\n\r No~!! Do not get valid data ");
        CMD_Repeat_sweep();
        //CyDelay(10);
        read_data = Read_I2C_test(AD5933_add, Sts_reg_1);
        //printf("\n\r No~!! Do not get valid data = %d ",read_data);
        read_data = ((num & (Eff_val_ch))|(num & (Cmp_q_shoot)));//read_data
& (Eff_val_ch);

        //printf("\n\r No~!! Do not get valid data = %d ",read_data);
        if(read_data == 0x00)
        {
            return 1;
        }
        else
        {
            return 0;
        }


    }

}
*/
//++++++++++++++++++++++++unit_cov_fun++++++++++++++++++++++++++++++++++++
/*
long Conv_8bit_to_16bit(uint8 Lower, uint8 Upper);
void Div_Start_Q(uint32 data);
void Div_Inc_Q(uint32 data);
void Conv_int_to_char(unsigned int data);
uint32 Data_proc(uint8 data1, uint8 data2);
unsigned int Conv_char_to_int(uint8 first, uint8 last);
double Conv_1byte_to_2byte(uint8 Lower, uint8 Upper);
*/
//====================Conv_8bit_to_16bit============================
long Conv_8bit_to_16bit(uint8 Lower, uint8 Upper)
```

```
{
    long data16 = 0;
    data16 = Lower+(Upper*256);
    if(data16 > 0x7FFF)
    {
        data16 = data16-65536;//0x10000
    }
    return data16;
}
//=========================================================================

//=====================Conv_8bit_to_16bit============================
double Conv_1byte_to_2byte(uint8 Lower, uint8 Upper)
{
    double data_2byte = 0;
    data_2byte = Lower+(Upper*256);
    if(data_2byte > 0x7FFF)
    {
        data_2byte = data_2byte-65536;//0x10000
    }
    return data_2byte;
}
//=========================================================================

//=====================Div_Start_Q=====================================
void Div_Start_Q(uint32 data)
{
    uint32 Msb_first = 0    ;
    uint32 Mid_1    = 0    ;
    uint32 Mid_2    = 0    ;
    uint32 Lsd_last  = 0    ;
    //printf("\n\r Start_Q_data = %ld",data);
    Msb_first   = data & 0xff000000    ;
    Mid_1      = data & 0xff0000      ;
    Mid_2      = data & 0xff00        ;
    Lsd_last    = data & 0xff          ;
    Start_Q[0]  = Msb_first/(0x1000000) ;
    Start_Q[1]  = Mid_1/(0x10000)      ;
    Start_Q[2]  = Mid_2/(0x100)        ;
    Start_Q[3]  = Lsd_last/(0x1)       ;
    //printf("\n\r Start_Q[1] = 0x%x",Start_Q[1]);
    //printf("\n\r Start_Q[2] = 0x%x",Start_Q[2]);
    //printf("\n\r Start_Q[3] = 0x%x",Start_Q[3]);
}//4byte를 1byte로 4개로 나눔
//=========================================================================

//=====================Div_Inc_Q=======================================
void Div_Inc_Q(uint32 data)
{
    uint32 Msb_first = 0    ;
    uint32 Mid_1    = 0    ;
    uint32 Mid_2    = 0    ;
    uint32 Lsd_last  = 0    ;
    //printf("\n\r Inc_Q_data = %ld",data);
    Msb_first   = data & 0xff000000    ;
    Mid_1      = data & 0xff0000      ;
    Mid_2      = data & 0xff00        ;
    Lsd_last    = data & 0xff          ;
    Inc_Q[0]  = Msb_first/(0x1000000)  ;
```

```
    Inc_Q[1]  = Mid_1/(0x10000)       ;
    Inc_Q[2]  = Mid_2/(0x100)         ;
    Inc_Q[3]  = Lsd_last/(0x1)        ;
    //printf("\n\r Inc_Q[1] = 0x%x",Inc_Q[1]);
    //printf("\n\r Inc_Q[2] = 0x%x",Inc_Q[2]);
    //printf("\n\r Inc_Q[3] = 0x%x",Inc_Q[3]);
}//4byte를 1byte로 4개로 나눔
//=============================================================================

//======================Conv_int_to_char================================
void Conv_int_to_char(unsigned int data)
{
    uint32 Msb_first = 0   ;
    uint32 Lsd_last  = 0   ;
    Msb_first   = data & 0xff00      ;
    Lsd_last    = data & 0xff        ;
    ItoC[0]  = Msb_first/(0x100)     ;
    ItoC[1]  = Lsd_last/(0x1)        ;
}//2byte를 1byte로 2개로 나눔
//=============================================================================

//============================Data_proc================================
uint32 Data_proc(uint8 data1, uint8 data2)
{
    uint32 a = 0;

    a = (uint32)data1*256+data2;
    if(a > 0x7fff)
    {
        a = 0x10000-a;
    }
    return a;
}
//=============================================================================

//=========================Conv_char_to_int============================
unsigned int Conv_char_to_int(uint8 first, uint8 last)
{
    unsigned int Conv_val = 0;
    //int conv_tem1 = 0;
    //int conv_tem2 = 0;
    //printf("\n\r first = %d last = %d ", first, last);
    Conv_val = first*(0x100)+last*(0x01);
    if(Conv_val > 0x7fff)
    {
        Conv_val = 0x10000-Conv_val;
        //printf("\n\r some_error");
    }
    return Conv_val;
}
//=============================================================================
//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


//++++++++++++++++++++++++CAL_data_fun++++++++++++++++++++++++++++++++++
/*
uint32 Cal_Frequency_code(uint32 Start_F);
void Cal_Imp_and_Phase_val(int cal_flg)
void Result_arr(int Num_of_Inc_Q, int stQQQ, int Inc_QQQ, int cal_flg)
```

```
int Cal_R_val();
int Cal_I_val();
double Data_cal_Phase(double img, double real);
*/
//=============================================================

void calibration_G_factor(uint32 St_Hz, uint32 Inc_Hz, unsigned int
Num_Of_Inc, unsigned int Settime, uint8 V_Range, uint8 Pg_gain, uint8
G_flg)
{

    double tem_GF = 0.0;
    int cal_st_hz = 0;
    uint8 i = 0;


    if(G_flg == 0)
    {
        Gain_f = 0.0;
        Cal_Gf_flg = G_flg;
        cal_st_hz = (St_Hz+(Inc_Hz*Num_Of_Inc))/2;
        CyDelay(10);
        init_AD5933(cal_st_hz,10, 1, Settime, V_Range,Pg_gain);
        Check_Sweep_CMP(1, 1,cal_st_hz, Inc_Hz);

            tem_GF = G_factor[1];

        Cal_Gain_f = (tem_GF *1.0)*COR_val;

        Cal_Gf_flg = 1;
        printf("\n\r if Result of Calibration Gain_f = %g",Cal_Gain_f);

    }
    else
    {
        Cal_Gf_flg = G_flg;
        Cal_Gain_f = Cal_Gain_f;
        printf("\n\r else REsult of Calibration Gain_f = %g",Cal_Gain_f);
    }
}

//======================Cal_Frequency_code=============================
uint32 Cal_Frequency_code(uint32 Start_F)
{
   double Start_F_code = 0;
    Start_F_code = (Start_F/(MCLK/4.0))*(Cal_2_n(27));
    //printf("\n\r 2^27 = %ld",Cal_2_n(27));
    //printf("\n\r cal_Start_F = %ld", Start_F);
    //printf("\n\r cal_Start_F_code = %x", Start_F_code);
    //printf("\n\r cal_Start_F_code = %ld", (uint32)Start_F_code);
    return (uint32)Start_F_code;
}//시작주파수_증감주파수  연산;
//=============================================================================
//======================Cal_Imp_and_Phase_val=========================
double Cal_Imp_and_Phase_val(int cal_flg)
{
    int16 R_Tem_R_data = 0;//2byte
    int16 R_Tem_I_data = 0;//2byte
```

```
    double Magnitude  = 0.0;
    double Sweep_phase = 0.0;

    int8 R_val[2] = {0};//1byte
    int8 I_val[2] = {0};//1byte

    double R_Gain_f = 0.0;
    double R_Imp_data = 0.0;

    //int16 aa = 0;
    int8 read_data = 0;

    //Cal_R_val();
    //CAl_I_val();
    //aa = sizeof(aa);
    //temppp = sizeof(temppp);
   // printf("\n\r sizeof(int16) = %d sizeof(int8) = %d ",aa,temppp);
    read_data = Read_I2C_test(AD5933_add, Sts_reg_1);
    printf("\n\r CMP?? read_data = %x", read_data);
    CyDelay(10);
    R_val[0] = Read_I2C_test(AD5933_add,ME_REAL_data_1);//Uper D15~8
    CyDelay(10);
    R_val[1] = Read_I2C_test(AD5933_add,ME_REAL_data_2);//Lower D7~0
    CyDelay(2);
    printf("\n\r org R0= 0x%x R1= 0x%x ",R_val[0], R_val[1] );
    //printf("\n\r org R100= 0x%x R256= 0x%x ",R_val[0]*0x100,
R_val[0]*256 );
    R_Tem_R_data =
((R_val[0])*256)+((R_val[1]));//)&0xff);//Conv_1byte_to_2byte(R_val[1],
R_val[0]);
    printf(" org R= 0x%x ",R_Tem_R_data);
    //CyDelay(2);
    if(R_Tem_R_data > 0x7fff)
    {
        R_Tem_R_data = R_Tem_R_data & 0x7fff;
        R_Tem_R_data = R_Tem_R_data - 65536;
        //R_Tem_R_data = 0x10000 - R_Tem_R_data;
        printf("\n\r if org R= 0x%x ",R_Tem_R_data);
    }
    else if(R_Tem_R_data <= 0x7fff)
    {
        R_Tem_R_data = R_Tem_R_data;
        printf("\n\r else org R= %x ",R_Tem_R_data);
    }

    CyDelay(10);
    I_val[0] = Read_I2C_test(AD5933_add,ME_IMAG_data_1);//Uper D15~8
    CyDelay(10);
    I_val[1] = Read_I2C_test(AD5933_add,ME_IMAG_data_2);//Lower D7~0
    CyDelay(2);
    printf("\n\r org I0= 0x%x I1= 0x%x ",I_val[0], I_val[1] );
    //printf("\n\r org I100= 0x%x I256= 0x%x ",I_val[0]*0x100,
I_val[0]*256 );
    R_Tem_I_data = ((I_val[0]*256))+((I_val[1]));//&0xff);
    printf(" org I= 0x%x ",R_Tem_I_data);
    CyDelay(2);
    if(R_Tem_I_data > 0x7fff)
    {
        R_Tem_I_data = R_Tem_I_data & 0x7fff;
        R_Tem_I_data = R_Tem_I_data - 65536;
```

```
        //R_Tem_I_data = 0x10000 - R_Tem_I_data;
        printf("\n\r if org I= %x ",R_Tem_I_data);
    }
    else if(R_Tem_I_data <= 0x7fff)
    {
        R_Tem_I_data = R_Tem_I_data;
        //R_Tem_I_data = R_Tem_I_data - 65536 + 2048;
        printf("\n\r else org I= %x ",R_Tem_I_data);
    }
    CyDelay(2);

    Magnitude =
sqrt(((((double)R_Tem_R_data)*(1.0))*(((double)R_Tem_R_data)*(1.0)))+((((do
uble)R_Tem_I_data)*(1.0))*(((double)R_Tem_I_data)*(1.0))));
    CyDelay(10);

    Sweep_phase =  (Data_cal_Phase(((double)R_Tem_R_data*1.0),
((double)R_Tem_I_data*1.0)) - Mid_phase);


    error = 0;
    if(cal_flg == 0)
    {
        //printf("\n\r 1 R_data = %d / I_data = %d",R_Tem_R_data,
R_Tem_I_data);
        R_Gain_f = (1.0/(Ref_resister))/(Magnitude);
        return R_Gain_f;
        //게인팩터 계산 전에는 게인팩터를 리턴한다.

    }
    else if(cal_flg != 0)
    {

        R_Imp_data = (1/(Cal_Gain_f*Magnitude));//kohm
        printf("\n\r 2 R_data = %d/ I_data = %d/ Magnitude = %0.3f/Imp_data
= %0.3f",R_Tem_R_data, R_Tem_I_data, Magnitude, R_Imp_data);
        return R_Imp_data;
        //게인팩터가 계산이 되어 있다면 임피던스 값을 반환한다.
    }
}
//=====================================================================

//=====================Result_arr=====================================
/*void Result_arr(int Num_of_Inc_Q, int stQQQ, int Inc_QQQ, int cal_flg)
{


    //CyDelay(10);
    //R_data[Num_of_Inc_Q]         = Tem_R_data;
    //CyDelay(10);
    //I_data[Num_of_Inc_Q]         = Tem_I_data;
    //CyDelay(10);
    //printf("\n\r R_data[%d] = %d / I_data[%d] = %d",Num_of_Inc_Q,
R_data[Num_of_Inc_Q],Num_of_Inc_Q, I_data[Num_of_Inc_Q]);
    if(cal_flg == 0)
    {
        //Manitude_arr[Num_of_Inc_Q]      = Magnitude    ;

//calibration_G_factor(ST_HZ_Q  ,INC_HZ_Q ,NUM_OF_INCVAL ,SETTLING_TIME_VAL
```

```
 ,VRANGE_VAL ,PG_GAIN_VAL, cal_flg);
       G_factor[Num_of_Inc_Q] = Gain_f;
       //printf("\n\r Calibration-ing-");
    }
    else
    {
       if(Num_of_Inc_Q == 0)
       {
          //printf("\n\r NUM   Frequency   Magnitude   Sweep_phase
Impedance");
       }

       Manitude_arr[Num_of_Inc_Q]     = Magnitude    ;
       CyDelay(10);
       Sweep_phase_arr[Num_of_Inc_Q]   = Sweep_phase  ;
       CyDelay(10);
       Frequency_arr[Num_of_Inc_Q]     = stQQQ+(Inc_QQQ*Num_of_Inc_Q);
       //printf("\n\r Start_Q_val = %d  / Inc_Q_val = %d ",stQQQ, Inc_QQQ);
       Imp_data_arr[Num_of_Inc_Q]     =
1/(Cal_Gain_f*Manitude_arr[Num_of_Inc_Q]);
       printf("\n\r %d     %ld      %.3f   %.3f   %.3f",Num_of_Inc_Q,
Frequency_arr[Num_of_Inc_Q], Manitude_arr[Num_of_Inc_Q],
Sweep_phase_arr[Num_of_Inc_Q], Imp_data_arr[Num_of_Inc_Q]);
       //printf("\n\r Imp_data_arr[%d] = %g ",Num_of_Inc_Q,
Imp_data_arr[Num_of_Inc_Q]);
       //CyDelay(10);
       //printf("\n\r Magnitude_arr[%d] = %g / Sweep_phase_arr[%d]
= %g",Num_of_Inc_Q, Manitude_arr[Num_of_Inc_Q], Num_of_Inc_Q,
Sweep_phase_arr[Num_of_Inc_Q]);
       //CyDelay(10);
       //printf("\n\r  Frequency_arr[%d] = %ld", Num_of_Inc_Q,
Frequency_arr[Num_of_Inc_Q]);
       //CyDelay(10);
       //printf("\n\r Result End");
       printf("\n\r Cal_GF = %g",Cal_Gain_f);
    }



}*/
//=========================================================================

//======================Cal_R_and_I_val================================
/*void Cal_R_val()
{
    int R_val[2] = {0};
    //int Tem_R_data= 0;
    //Read R data
    //I2C_1_MasterSendStart(0x0D, 1);
    R_val[0] = Read_I2C_test(AD5933_add,ME_REAL_data_1);//Uper
    R_val[1] = Read_I2C_test(AD5933_add,ME_REAL_data_2);//Lower
    Tem_R_data = R_val[1]+R_val[0]*256;//Conv_1byte_to_2byte(R_val[1],
R_val[0]);
    CyDelay(10);
    if(Tem_R_data <= 32767.0)
    {
       //Tem_R_data = Tem_R_data;
    }
    else
    {
```

```
        Tem_R_data = Tem_R_data & 0x7FFF;
        Tem_R_data = Tem_R_data - 65536;
    }
    //printf("\n\r Tem_R_data = %d ",Tem_R_data);
    CyDelay(2);
    //return Tem_R_data;
}

void Cal_I_val()
{
    int I_val[2] = {0};
    //int Tem_I_data = 0;
    //Read I data
    //I2C_1_MasterSendStart(0x0D, 1);
    I_val[0] = Read_I2C_test(AD5933_add,ME_IMAG_data_1);//Uper
    //I2C_1_MasterSendStart(0x0D, 1);
    I_val[1] = Read_I2C_test(AD5933_add,ME_IMAG_data_2);//Lower
    Tem_I_data = I_val[1]+ (I_val[0]*256);
    //CyDelay(10);
    CyDelay(10);
    if(Tem_I_data <= 0x7fff)
    {
        //Tem_I_data = Tem_I_data;
    }
    else
    {
        Tem_I_data = Tem_I_data & 0x7FFF;
        Tem_I_data = Tem_I_data - 65536;
    }
   //printf("\n\r Tem_I_data = %d ",Tem_I_data);
    CyDelay(2);
    //return Tem_I_data;

}*/
//========================================================================



//=========================Data_cal_Phase=============================
double Data_cal_Phase(double img, double real)
{
    double theta = 0.0;
    double phase2 = 0.0;
    if((real > 0.0)& (img > 0.0))
    {
        theta = atan(img/real);
        phase2 = (theta * 180.0) / M_PI;
    }
    else if((real > 0.0) & (img < 0.0))
    {
        theta = atan(img/real);
        phase2 = ((theta * 180.0) / M_PI)+360.0;
    }
    else if((real < 0.0) & (img < 0.0))
    {
        theta = ((-1.0)*(M_PI))+atan(img/real);
        phase2 = (theta * 180.0) / M_PI;
    }
    else if((real < 0.0)&(img > 0.0))
    {
```

```
        theta = M_PI+atan(img/real);
        phase2 = (theta * 180.0) / M_PI;
    }
    return phase2;
//#endif
}

//=========================================================================

//++++++++++++++++++++++++++AD5933 control+++++++++++++++++++++++++++++++
/*
void CMD_Repeat_sweep();
uint8 Check_valid_data();
void Init_and_Start_Frequency_Sweep();
void Make_st_f_and_inc_f(uint32 st_f, uint32 inc_f);
uint8 CMD_Out_V_Range(uint8 Vr);
uint8 CMD_Pggain(uint8 Gain);
void CMD_Temp();
void Pow_down();
*/
//=========================CMD_Repeat_sweep=============================
void CMD_Repeat_sweep()
{
    //uint8 reg_data[2]={0};
    Write_I2C_test(AD5933_add, CNT_reg_1,CMD_Re_sweep);
}

//=========================================================================

//=========================Check_valid_data=============================
uint8 Check_valid_data()
{
    uint8 read_data = 0;
    //I2C_1_MasterSendStart(0x0D, 1);
    read_data = Read_I2C_test(AD5933_add, Sts_reg_1);
    while(!(read_data&(0x02)))
    {
        //I2C_1_MasterSendStart(0x0D, 1);
        read_data = Read_I2C_test(AD5933_add, Sts_reg_1);
        if(read_data&(0x02))
            {
            return 1;
        }
        else
        {
            return 0;
        }
    }
    return 1;
}
//=========================================================================

//=================Init_and_Start_Frequency_Sweep=====================
void Init_and_Start_Frequency_Sweep()
{
    //write Init sensor with start Frequency
    Write_I2C_test(AD5933_add, CNT_reg_1, CMD_Q_Inc);

    //write start Frequency sweep
    Write_I2C_test(AD5933_add, CNT_reg_1, CMD_Q_sweep_start);
```

```
}
//=============================================================================

//=========================Make_st_f_and_inc_f=========================
void Make_st_f_and_inc_f(uint32 st_f, uint32 inc_f)
{
    uint32 cal_st_f = 0;
    uint32 cal_inc_f = 0;
    cal_st_f = Cal_Frequency_code(st_f);
    cal_inc_f = Cal_Frequency_code(inc_f);
    Div_Start_Q(cal_st_f);
    Div_Inc_Q(cal_inc_f);
}//주파수 연산
//=============================================================================

//============================CMD_Out_V_Range==========================
uint8 CMD_Out_V_Range(uint8 Vr)
{
    uint8 temp1 = 0;

    if(Vr == 1)
    {
        temp1 = CMD_out_1vpp;
    }
    else if(Vr == 2)
    {
        temp1 = CMD_out_2vpp;
    }
    else if(Vr == 20)
    {
        temp1 = CMD_out_200mvpp;
    }
    else if(Vr == 40)
    {
        temp1 = CMD_out_400mvpp;
    }
    else
    {
        temp1 = 0;
    }
    return temp1;
}
//=============================================================================

//=============================CMD_Pggain============================
uint8 CMD_Pggain(uint8 Gain)
{
    uint8 temp1 = 0;
    if(Gain ==1)
    {
        temp1 = CMD_pg_x1;
    }
    else if(Gain == 5)
    {
        temp1 = CMD_pg_x5;
    }
    else
    {
        temp1 = 0;
```

```
    }
    return temp1;
}
//=====================================================================

//==================Temperature=========================================
double CMD_Temp()
{
    uint8 Data_buf   =  0 ;
    uint8 tempUper   =  0 ;
    uint8 tempLower  =  0 ;
    double tempature =  0 ;
        //Write Reset

    //Write_I2C_test(CNT_reg_2,CMD_Reset );
    Pow_down();
    Write_I2C_test(AD5933_add, CNT_reg_2, CMD_Reset);


    //write temperature reading
    Write_I2C_test(AD5933_add, CNT_reg_1, CMD_temp_me);
    CyDelay(100);
    Data_buf = Read_I2C_test(AD5933_add, Sts_reg_1);    //status reg read
    Data_buf = Data_buf & 0x01;           // Valid temperature measurement
    if(Data_buf != 1)
    {
        while(!(Data_buf&0x01))//1이 나올때 까지 loop
        {
            Write_I2C_test(AD5933_add, CNT_reg_1, CMD_temp_me);
            CyDelay(20);
            Data_buf  = Read_I2C_test(AD5933_add, Sts_reg_1);
            Data_buf  = Data_buf & 0x01;
            printf("\n\r Temperature loop Data_buf = %d ",Data_buf);
        }
    }
    else//온도측정 완료
    {
        //printf("\n\r Temperature Measurement Complete");
        tempUper  = Read_I2C_test(AD5933_add, Temp_data_1)& (0x3F);

        tempLower = Read_I2C_test(AD5933_add, Temp_data_2);
        //printf("\n\r tempUper = %X",tempUper );
        //printf("\n\r tempLower = %X",tempLower);

        tempature = (Conv_8bit_to_16bit(tempLower, tempUper))/32;
        printf("\n\r Temperature/32 = %f",(double)(tempature));
        //printf("\n\r Temperature = %d",tempature);
    }
    return (double)(tempature);
}
//=============================================================
//==================Power down=================================
void Pow_down()
{
    Write_I2C_test(AD5933_add, CNT_reg_1, CMD_pow_dw);
}
//=====================================================================
//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
/* [] END OF FILE */
```

```
/* ======================================
 * PRJ_AD5933 OPeration
 * make by WOO Seong Yong
 * DATA_MAPPING.h
 * ======================================
*/
//#include <project.h>

#define PI_num  (3.141592654)
#define SL_W    (0x1a)//AD5933은 고정 주소_변경불가 13번(0X0D) 사용 + 마지막bit 0
#define SL_R    (0x1b)//AD5933은 고정 주소_변경불가 13번(0X0D) 사용 + 마지막bit 1
#define AD5933_add  (0x0D)
//===========command_code==============================
#define Block_Write     (0xA0)
#define Block_Read      (0xA1)
#define Add_Pointer     (0xB0)
//=====================================================
//===========Control Register==========================
#define CNT_reg_1       (0x80)//D15~D12
#define CNT_reg_2       (0x81)//D11~D8

//========Mecro==================
//===0x80===
#define CMD_pg_x5           (0x00)//D8 PG gain X5설정
#define CMD_pg_x1           (0x01)//D8 PG gain X1설정
#define CMD_out_2vpp            (0x00)//D10~D9 출력전압 2Vpp
#define CMD_out_200mvpp         (0x02)//D10~D9 출력전압 200mVpp
#define CMD_out_400mvpp         (0x04)//D10~D9 출력전압 400mVpp
#define CMD_out_1vpp            (0x06)//D10~D9 출력전압 1Vpp
#define CMD_St_Q_Init       (0x10)//D15~D12 시작주파수 설정
#define CMD_Q_sweep_start   (0x20)//D15~D12 주파수 구동명령
#define CMD_Q_Inc           (0x30)//D15~D12 주파수 증감값
#define CMD_Re_sweep        (0x40)//D15~D12 명령없음
#define CMD_temp_me         (0x90)//D15~D12 온도측정
#define CMD_pow_dw          (0xA0)//D15~D12 파원다운모드
#define CMD_standby_mo      (0xB0)//D15~D12 대기모드

//===0x81===
#define CMD_ex_Osc          (0x08)//D3 Enable external Oscillator
#define CMD_Reset           (0x10)//D7~0 Reset
//========CNT_reg datamap========
#define CNT_reg_D0      (0)
#define CNT_reg_D1      (1)
#define CNT_reg_D2      (2)
#define CNT_reg_D3      (3)
#define CNT_reg_D4      (4)
#define CNT_reg_D5      (5)
#define CNT_reg_D6      (6)
#define CNT_reg_D7      (7)
#define CNT_reg_D8      (0)
#define CNT_reg_D9      (1)
#define CNT_reg_D10     (2)
```

```
#define CNT_reg_D11     (3)
#define CNT_reg_D12     (4)
#define CNT_reg_D13     (5)
#define CNT_reg_D14     (6)
#define CNT_reg_D15     (7)



//===========START FREQUENCY = ST_Hz_Val ===============
//ST_Hz_Val = [(START FREQUENCY)/(Device_CLK/4)] * 2^27
//시작 주파수는 계산식에 START FREQUENCY이며 해당 주소에
//입력값과는 상이하며 해당 레지스터에는 계산된 ST_Hz_Val을 넣는다.
#define ST_Hz_1         (0x82)//D23~D16
#define ST_Hz_2         (0x83)//D15~D8
#define ST_Hz_3         (0x84)//D7~D0

//===========FREQUENCY INCREMENT=======================
//INC_Hz_Val = [(FREQUENCY INCREMENT)/(Device_CLK/4)] * 2^27
//증분주파수는 측정시 주파수의 증가치를 가리키며 해당주소에
//원하는 주파수치에대한 계산값인 INC_Hz_Val값을 나눠 넣는다.
#define INC_Hz_1        (0x85)//D23~D16
#define INC_Hz_2        (0x86)//D15~D8
#define INC_Hz_3        (0x87)//D7~D0

//===========NUMBER OF INCRMENTs=======================
//몇회 증가를 할건지 지정한다.
// 즉 측정범위를 증분주파수와 해당값의 곱으로
//최대 주파수 영역을 알 수 있다.
//MAX값은 511(0X1FF)이다.
#define NUMoINC_1       (0x88)//D15~D8
#define NUMoINC_2       (0x89)//D7~D0

//===========NUMBER OF SETTLING TIME CYCLE==============
//SETTLING TIME=CYCLE*CYCLE 배수*ST_Hz_Val 주기시간
#define NUMoSETIME_1    (0x8A)//D15~D8
#define NUMoSETIME_2    (0x8B)//D7~D0

//========Mecro==================
//===0x8A===
#define CMD_cyl_def (0x00)//기본값
#define CMD_cyl_x2  (0x02)// X 2배
#define CMD_cyl_x4  (0x06)// X 4배
//==========
//========settime_reg datamap========
#define NUMoSETIME_D0   (0)
#define NUMoSETIME_D1   (1)
#define NUMoSETIME_D2   (2)
#define NUMoSETIME_D3   (3)
#define NUMoSETIME_D4   (4)
#define NUMoSETIME_D5   (5)
#define NUMoSETIME_D6   (6)
#define NUMoSETIME_D7   (7)
#define NUMoSETIME_D8   (8)
```

```
#define NUMoSETIME_D9   (9)
#define NUMoSETIME_D10  (10)
#define NUMoSETIME_D11  (11)
#define NUMoSETIME_D12  (12)
#define NUMoSETIME_D13  (13)
#define NUMoSETIME_D14  (14)
#define NUMoSETIME_D15  (15)


//===========Status Register============================
//상태확인
#define Sts_reg_1       (0x8F)//D7~D0

//========Mecro=================
//===0x8F===
#define Eff_temp_ch (0x01)//유효온도 측정
#define Eff_val_ch  (0x02)//유효 실제/가상 값 측정
#define Cmp_q_shoot (0x04)//주파수 조사완료
//==========

//========Status_reg datamap========
#define Sts_reg_D0  (0)
#define Sts_reg_D1  (1)
#define Sts_reg_D2  (2)
#define Sts_reg_D3  (3)
#define Sts_reg_D4  (4)
#define Sts_reg_D5  (5)
#define Sts_reg_D6  (6)
#define Sts_reg_D7  (7)
//===========Temperature data===========================
//온도 측정
#define Temp_data_1     (0x92)//D15~D8
#define Temp_data_2     (0x93)//D7~D0

//===========Real data==================================
//측정 값
#define ME_REAL_data_1  (0x94)//D15~D8
#define ME_REAL_data_2  (0x95)//D7~D0

//===========Imaginary data=============================
//측정 값
#define ME_IMAG_data_1  (0x96)//D15~D8
#define ME_IMAG_data_2  (0x97)//D7~D0

//+++++++++++++++++값 변환방법++++++++++++++++++++++++++

//DATA_val = sqrt[(Real^2)+(IMAG^2)]

//++++++++++++++++++++++++++++++++++++++++++++++++++++++

//입출력 전역변수=====================================

    /* [] END OF FILE */
```

```
/* ========================================
 * PRJ_AD5933 OPeration
 * make by WOO Seong Yong
 * I2C_1_CMD.h
 * ========================================
*/
#include <project.h>

void Write_I2C_test(uint8 add, uint8 reg, uint8 data );
uint8 Read_I2C_test(uint8 add, uint8 reg);

void Write_I2C_test(uint8 add, uint8 reg, uint8 data )
{
    uint8 status = 0;
    uint8 userArray[2] = {0};
    uint8 i = 0;
    userArray[0] = reg ;
    userArray[1] = data;
    status = I2C_1_MasterStatus();
    //printf("\n\r write status = %d ", status);
    if(status != 0)
    {
        I2C_1_MasterClearWriteBuf();
        I2C_1_MasterClearReadBuf();
        I2C_1_MasterClearStatus();
        status = I2C_1_MasterStatus();
        //printf("\n\r write CLe status = %d ", status);
    }
    status = I2C_1_MasterSendStart(add, I2C_1_WRITE_XFER_MODE);
    CyDelay(1);

    if(I2C_1_MSTR_NO_ERROR == status)   /* Check if transfer completed
without errors */
    {
       for(i=0; i<2; i++)
       {
          status = I2C_1_MasterWriteByte(userArray[i]);
        CyDelay(1);
          //printf("\n\r status = %d ", status);
          if(status != I2C_1_MSTR_NO_ERROR)
          {
            CyDelay(1);
             //printf("\n\r break");
             break;
          }
       }
    }
    CyDelay(1);
    I2C_1_MasterSendStop();      /* Send Stop */
}

uint8 Read_I2C_test(uint8 add, uint8 reg)
{
    uint8 status = 0;
    uint8 Data_result = 0;
    status = I2C_1_MasterStatus();
    CyDelay(1);
    if(status != 0)
    {
        I2C_1_MasterClearWriteBuf();
```

```
        I2C_1_MasterClearReadBuf();
        I2C_1_MasterClearStatus();
        status = I2C_1_MasterStatus();
        //printf("\n\r read CLe status = %d ", status);
    }
    Write_I2C_test(add, Add_Pointer, reg);
    CyDelay(1);
    status = I2C_1_MasterSendStart(add, I2C_1_READ_XFER_MODE);
    CyDelay(1);
    //printf("\n\r read status = %d ", status);
    if(I2C_1_MSTR_NO_ERROR == status)   /* Check if transfer completed
without errors */
    {
        CyDelay(1);
            Data_result = I2C_1_MasterReadByte(I2C_1_NAK_DATA);
    }
    I2C_1_MasterSendStop();     /* Send Stop */
    CyDelay(1);
    return Data_result;
}

/* [] END OF FILE */
```

```
/* ========================================
 * 2014~2015 code by Woo seongyong
 * I2C and UART
 * PRJ_AD5933 OPeration
 * make by WOO Seong Yong
 * UART_PSOC.h
 * ========================================
*/
#include <UART_1.h>
#include <errno.h>
#include <sys/stat.h>
#include <sys/unistd.h>
#define RxBufferSize       100                  //Uart buffer

uint8  ReceiveBuffer[RxBufferSize];
uint8  *RxReadIndex  = ReceiveBuffer;
uint8  *RxWriteIndex= ReceiveBuffer;

void Wait(void);
uint8  IsCharReady(void);
uint8  GetRxChar(void);
//CY_ISR(MyRxInt);



//Printf
================================================================================
============
#if (CY_PSOC3)
    /* For Keil compiler revise putchar() function with
       component which has to send data */
    char putchar( char c)
    {
        //UART_PutChar(c);
        UART_1_PutChar(c);
        return c;
    }
#else
    #if defined(__ARMCC_VERSION)

    /* For MDK/RVDS compiler revise fputc function */
    struct __FILE
    {
        int handle;
    };
      enum
    {
        STDIN_HANDLE,
        STDOUT_HANDLE,
        STDERR_HANDLE
    };

    FILE __stdin = {STDIN_HANDLE};
    FILE __stdout = {STDOUT_HANDLE};
    FILE __stderr = {STDERR_HANDLE};

    int fputc(int ch, FILE *file)
    {
        int ret = EOF;
        switch( file->handle )
```

```
        {
            case STDOUT_HANDLE:
                //UART_PutChar(ch);
            UART_1_PutChar(ch);
                ret = ch;
                break;
                case STDERR_HANDLE:
                ret = ch;
                break;
                          default:
                file = file;
                break;
        }
        return(ret);
    }

#elif defined (__ICCARM__)      /* IAR */
/* For IAR compiler revise __write() function for printf functionality */
size_t __write(int handle, const unsigned char * buffer, size_t size)
{
    size_t nChars = 0;
    for (/* Empty */; size != 0; --size)
    {
        //UART_PutChar(*buffer++);
        //UART_1_PutChar(*buffer++);
        ++nChars;
    }
        return (nChars);
}

#else  /* (__GNUC__)  GCC */

    /* For GCC compiler revise _write() function */
    int _write(int file, char *ptr, int len)
    {
        /*int i;
        for (i = 0; i < len; i++)
        {
            //UART_PutChar(*ptr++);
            UART_1_PutChar(*ptr++);
        }
        return(len); */
        int n;
            switch(file)
         {
                    case STDOUT_FILENO:
                    case STDERR_FILENO:
                           for(n = 0; n < len; n++)
                {
                               UART_1_PutChar(*ptr++);//
                           }
                       break;
                    default:
                           errno = EBADF;
                           return -1;
        }
        return len;
    }

    #endif  /* (__ARMCC_VERSION) */
```

```
#endif /* CY_PSOC3 */

/* Add an explicit reference to the floating point printf library to allow
   the usage of floating point conversion specifier */
#if defined (__GNUC__)
    asm (".global _printf_float");
#endif

void Wait(void)
{
       // Do nothing
}

uint8  IsCharReady(void)
{
       return !(RxWriteIndex == RxReadIndex);
}

uint8  GetRxChar(void)
{
uint8  Result;
       while(!IsCharReady()) Wait();
       Result = *RxReadIndex++;
       if (RxReadIndex >= ReceiveBuffer + RxBufferSize) RxReadIndex =
ReceiveBuffer;
       return Result;
}


/* [] END OF FILE */
```

```
/* ======================================
 * PRJ_AD5933 OPeration
 * make by WOO Seong Yong
 * User_math.h
 * ======================================
*/
#include <project.h>
#include <math.h>

uint32 Cal_2_n(uint8 num)   ;
int TR_ang(double ang)      ;
double COSINE(double dig)   ;
double SINE(double dig)     ;
double TAN1(double dig)     ;
int ACOSINE(double dig)     ;//0~180
int ASINE(double dig)       ;//-90~90
int ACTAN1(double dig)      ;//-90~90
double use_pow(double a, int b);


uint32 Cal_2_n(uint8 num)
{
    uint8 i = 0;
    uint32 result_cal = 1;
    for(i=0 ; i < num; i++)
    {
        result_cal = result_cal * (2);
    }
    return result_cal;
}
int TR_ang(double ang)
{
       int andd;
       andd = ((int)ang);
       if ((ang-andd) >= (0.5))
       {
       //     printf("\r\n 5");
       andd = andd+1;
       return andd;
       }
       else
       //     printf("\r\n 6");
       return andd;
}

//계산에 필요한 삼각함수들 math.h의 것을 불러 사용하지만 ================
//dig용으로 변환 사용하면 atan2는 많이 다름 ============================
double COSINE(double dig)
{
       double cs = 0;
       cs = cos(dig*(M_PI/180));
       return cs;
}
double SINE(double dig)
{
       double sine = 0;
       sine = sin(dig*(M_PI/180));
       return sine;
}
```

```
double TAN1(double dig)
{
	double tan1 = 0;
	tan1 = tan(dig*(M_PI/180));
	return tan1;
}

int ACOSINE(double dig)//0~180
{
	double cccs = 0;
	int temp_ac = 0;
	cccs = (180/M_PI)*acos(dig);

	temp_ac = ((int)cccs);
	if ((cccs-temp_ac) >= (0.5))
	{
	//	printf("\r\n 5");
	temp_ac = temp_ac+1;
	return temp_ac;
	}
	else
	//	printf("\r\n 6");
	return temp_ac;
}


int ASINE(double dig)                    //-90~90
{
	double aasine = 0;
	int temp_as = 0;
	aasine = (180/M_PI)*asin(dig);

	temp_as = ((int)aasine);
	if ((aasine-temp_as) >= (0.5))
	{
	//	printf("\r\n 5");
	temp_as = temp_as+1;
	return temp_as;
	}
	else
	//	printf("\r\n 6");
	return temp_as;
}

int ACTAN1(double dig)                   //-90~90
{
	double ACtan1 = 0;
	int temp_at = 0;
	ACtan1 = (180/M_PI)*atan(dig);

	temp_at = ((int)ACtan1);
	if ((ACtan1-temp_at) >= (0.5))
	{
	//	printf("\r\n 5");
	temp_at = temp_at+1;
	return temp_at;
	}
	else
	//	printf("\r\n 6");
```

```
	return temp_at;
}

double use_pow(double a, int b)
{
	double sol=1.0;
	int i=0;

	for(i=0;i<b;i++)
	sol = sol*a;
//	 printf("\r\n sol = %f",sol);
//	sol = a*a;
	return sol;
}

/* [] END OF FILE */


/* ========================================
 * PRJ_AD5933 OPeration
 * make by WOO Seong Yong
 * device.h
 * ========================================
*/
********************************************************************************
****/
#ifndef DEVICE_H
#define DEVICE_H
#include <project.h>
//#include <device.h>

#include <stdio.h>
#include <stdlib.h>		//malloc

#include "UART_PSOC.h"


#include "AD5933_4_fun.h"

#endif
/* [] END OF FILE */
```