DIP COATING UNIT

Major Project Report submitted

to

PUNJAB ENGINEERING COLLEGE, CHANDIGARH

For Fulfilment of the Academic Requirement for the
Award of the Grades

of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS & COMMUNICATION ENGINEERING

By

AYUSHMAAN BENIWAL	**(18105005)**
VANKSHU BANSAL	**(18105011)**
NIKHIL MEHTA	**(18105012)**
KANNAN GILL	**(18105016)**
PRANAV KAUSHAL	**(18105066)**

Under the guidance of

Dr. Gaurav Mani Khanal
Assistant Professor
Department of Electronics & Comm. Engineering
PEC Institute of Technology

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

PEC
EXPLORE INNOVATE EXCEL

2021 – 2022

DECLARATION

We, hereby, declare that the project titled **"DIP COATING UNIT"** is an authentic record of our own work carried out as major Project for the award of degree of B.Tech. Electronics and Communication Engineering, Punjab Engineering College (deemed to be University), under the guidance of Dr. Gaurav Mani Khanal, from August to December 2021.

Ayushmaan Beniwal	(18105005)
Vankshu Bansal	(18105011)
Nikhil Mehta	(18105012)
Kannan Gill	(18105016)
Pranav Kaushal	(18105066)

Date: ______________

Certified that the above statement made by the students is correct to the best of my knowledge and belief.

Under the guidance of:

Dr. Gaurav Mani Khanal
Assistant Professor
Department of Electronics & Comm. Engineering
PEC Institute of Technology

ACKNOWLEDGEMENT

We would like to take this opportunity to express our profound gratitude to our faculty mentor- **Dr. Gaurav Mani** for his guidance in completing the project “**Dip Coating Unit**” for his exemplary guidance, valuable feedback and constant encouragement throughout the duration of the project. Their valuable suggestions were of immense help throughout our project work. The perceptive criticism kept us working to make this project in a much better way. Working under his guidance was an extremely knowledgeable experience.

We are also grateful to the college and staff of the **ECE Department**, who taught the fundamentals and allowed us to take up this project which has been a great learning experience for us.

We would like to give our sincere gratitude to all the respondents for giving the information that was required, without which the project would have been incomplete.

Ayushmaan Beniwal	(18105005)
Vankshu Bansal	(18105011)
Nikhil Mehta	(18105012)
Kannan Gill	(18105016)
Pranav Kaushal	(18105066)

EXPLORE INNOVATE EXCEL

\

CONTENTS

CHAPTER 6: RESULTS AND OBSERVATIONS

CHAPTER 7: SCHEMATIC

CHAPTER 8: CODE

CHAPTER 9: APPLICATIONS AND LIMITATIONS

CHAPTER 10: CONCLUSION AND FUTURE SCOPE

CHAPTER 1

INTRODUCTION

A semiconductor is a material product usually comprised of silicon, which conducts electricity more than an insulator, such as glass, but less than a pure conductor, such as copper or aluminium. Their conductivity and other properties can be altered with the introduction of impurities, called doping, to meet the specific needs of the electronic component in which it resides.

Semiconductor device fabrication begins with the deposition of extremely thin films of material on silicon wafers.

Coating process describes the application of material to a substrate.

Dip coating is one of the most widely-used coating processes in industry and academia for producing thin films of semiconductor. To create a film, the substrate is first lowered into, and then withdrawn from, the solution. By controlling the speed of substrate withdrawal from solution using a programmable dip coater, you can vary the thickness of the deposited film. A high-precision motor means that the rate of withdrawal - and therefore the film thickness - can be controlled with a high degree of accuracy and reproducibility.

Because dip coating is suitable for roll-to-roll processing, it is used extensively in manufacturing. The Dip Coater allows researchers and academics to use the method in the lab. Dip coating is a straight-forward way of producing films, and it is therefore relatively easy to create high quality uniform coatings with little prior knowledge.

We aim to assist laboratories in semiconductor fabrication research by designing a portable, cheap and high quality device which allows us to change and maintain various parameters that influence the thickness of semiconductor coating like dip speed, retract speed, delay time, number of cycles, vibration, temperature and moisture.

1.1 Motivation

The idea of this project arose from the need to implement a dip coating device in the new semiconductor laboratory of PEC, Chandigarh. This device was not only needed to perform high quality fabrication of semiconductors but also to give a cheap and portable alternate to market available options in order to conduct research activities efficiently in the lab.

Finding this as a real good opportunity to implement the skills we have acquired during our course of engineering, we decided to take up this project and work on a dip coating device.

The thought of researching our topics of interest and use various technologies and domains of engineering collectively to come up with laboratory ready device was the driving source of motivation. Our project can be used in laboratory for training and testing purposes.

1.2 Need of the project

This project can be used to perform dip coating on a substrate at a much cheaper rate as compared to the market available options. Dip coating is very important step in electronics and for a lot of companies and laboratories, the main blocking factor in making their own substrate and conducting research by varying parameters is the high cost of dip coater. This project targets to resolve that issue and encourage the manufacture and testing of various properties on the thickness of coating on a substrate by giving them a much cheaper alternate.

This allows us to learn a lot of new concepts as well as implement the theoretical concepts we have learnt over the last 4 years. It will certainly give our juniors an opportunity to see the effect on thickness by varying different parameters and also encourage them to take such projects in the future.

1.3 Objectives

- With our project we aim to make a production ready dip coating unit with various user – given parameters like dip speed, retract speed, number of cycles and delay time in order to influence the thickness of coating on the substrate.

- We also plan to improve the user interface by adding lcd screen, matrix keyboard, web application for remote access, and pre-defined speed settings in our project so that minimal time is spent to understand the working of Dip coater Unit and emphasis can be laid on experiments and research.

- Experiment environment is very important in the process of Dip Coating. Hence we display and maintain a record of Temperature and Humidity of the environment.

- A dedicated website for the Dip Coater Unit will allow students to run Dip Coater Unit over the internet without the need to be physically present around there. The website will also maintain experiment logs.

- Focus is being paid to reduce vibrations so that they have minimal effect on coating.
- Steps to reduce accidental damage in case of hardware failure are also to be implemented.

- Since we want to implement this project in the new semiconductor fabrication laboratory of PEC, we want to use metallic or ceramic apparatus which can be used in a clean room.

- Finally we intend to reduce the size of model to make it portable and give a professional appearance.

1.4 Project Flow

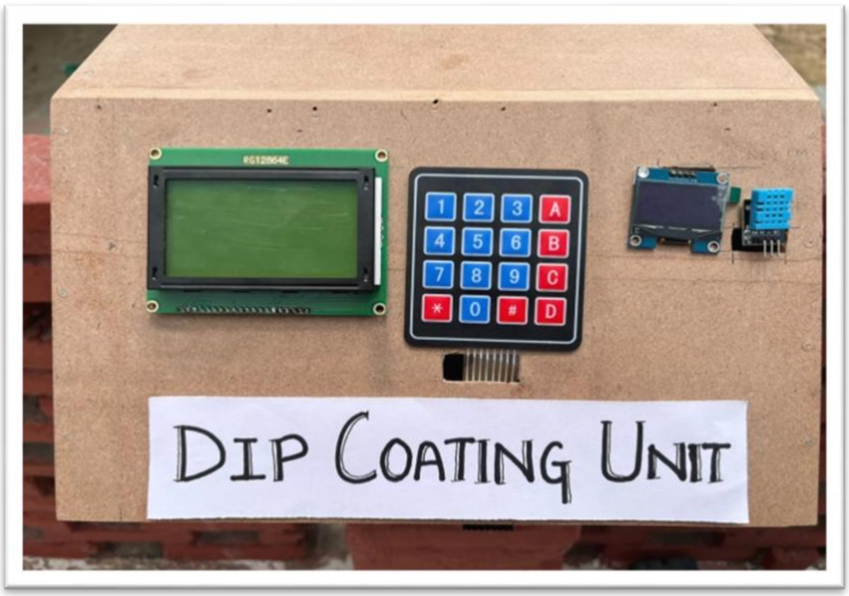
- Attach the substrate to an Alligator clip and place the coating solution in the solution container.
- Turn the power supply of Motor and Arduino on and read the instructions on the lcd display.
- The user can enter the required values of dip speed, retract speed, number of cycles, and the delay time using the matrix keyboard or he can use the dip coater website to do the same wirelessly.
- Arduino receives the instructions and has all the variables with it. It commands the stepper motor to start.
- Rotation of motor leads to downward movement of substrate based on the dip speed given.
- The motor stops there for the delay and allows coating solution to accumulate over the substrate.
- Substrate is then pulled up at retract speed and results can be seen on the LCD screen.

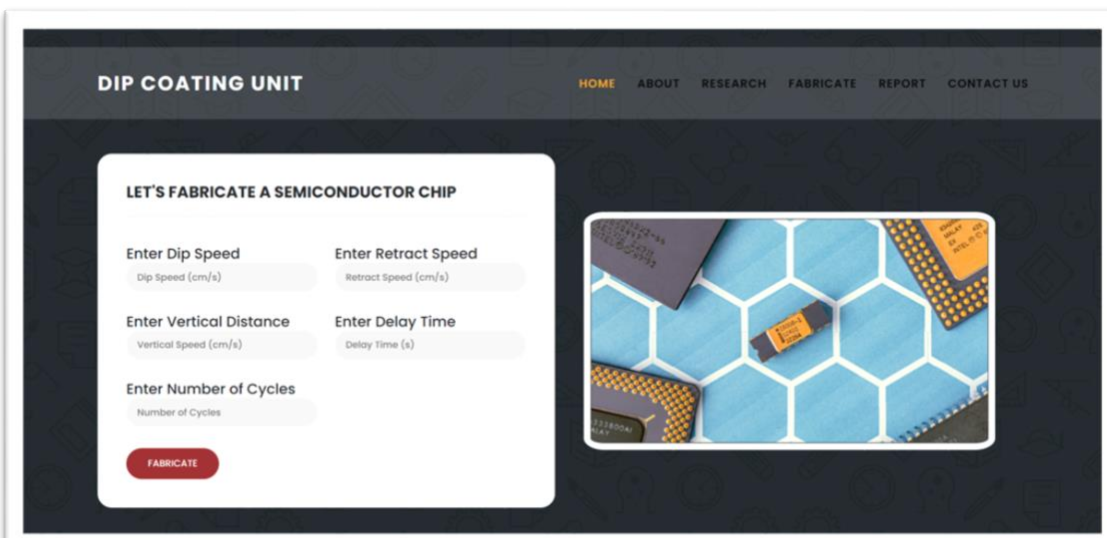
- This process is repeated for the number of cycles specified.

- The observed thickness shall be entered by student in the Research Data tab of the website in order to maintain experimental logs for future

- **INPUT**

  This Dip Coating Unit can take two type of inputs, Manual input through the matrix keyboard and wireless input through the website.





PEC
EXPLORE INNOVATE EXCEL

- **Model**

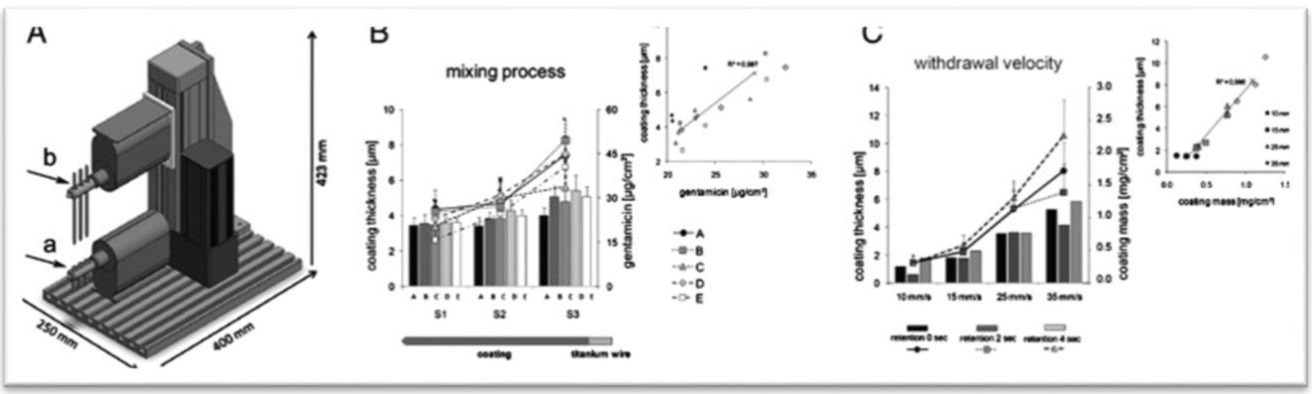
EXPLORE INNOVATE EXCEL

CHAPTER 2

LITERATURE REVIEW

2.1 Research Papers

Dip coating is a common technique for the optimization of metallic implants, whereas several steps during the procedure influence the coating. Dip coating is one of the most effective processes for the production of PSC. The solution substrate is immersed in the solution for effective formation of the material. Once the material is deposited then the substrate can be removed by evaporation which will result the thickness of the layer are unique. The major forces that are used for the dip coating process are force of inertia, viscosity, drag gravitational forces, and surface tension. The merits of dip coating are low cost and layer thickness can be easily adjusted. The drawbacks of dip coating are process is slow and it has the ability to block the screen, which will create major impact in the final product.



Immersion dip coating figure

Dip coating, also known as impregnation or saturation, is a popular way of creating a thin and uniform coating onto flat or cylindrical substrates. In this simple process, the substrate is dipped into a bath of the coating, which is normally of a low viscosity to enable the coating to run back into the bath as the substrate emerges. The excess material is then squeezed out by passing through nip rolls or a set of flexible doctor blades pre-calibrated to give a fixed net pickup of the resin. The solid content of the impregnating liquor and the absorption capacity of the fabric need to be considered in designing an impregnated fabric. In dip coating, the pickup is quite low, and penetration occurs into the interstices of the fabrics as well as in the yarns so it is frequently used on porous substrates.

The advantage of the process is that no stress in the fabric and no damage or distortion to the yarn occurs. Dip coating involves immersing a substrate in a precursor solution and then lifting it vertically from the solution with a velocity $v0$.

A wet film with thickness λ is dragged from the liquid upward along with the moving substrate. Despite its simple appearance, the dip-coating process involves a complex interplay between many counteracting factors: viscous drag upward on the liquid by the moving substrate, force of gravity on the wet film, surface tension in the concavely shaped meniscus, surface tension gradient along the height of the film due to drying effects, the disjoining (or conjoining) pressure (important for films less than 1-μm thick) and others (Brinker, 1992). The wet film thickness can be estimated from the relationship by Landau and Levich:

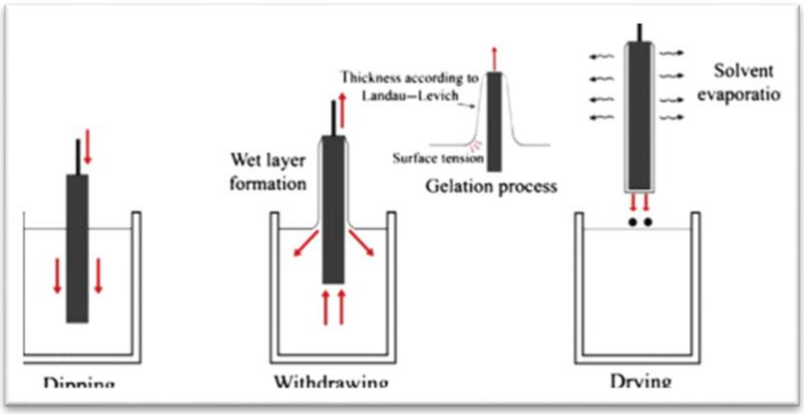
Dip coating is also known as slurry or vacuum slurry dip coating. This method is usually used for fabricating functional layers. Similarly to the screen-printing method, this method uses slurry in the fabrication process.

The slurry is a combination of ceramic powder, solvent, binder, and dispersant. In the dip-coating method the supporting layer is soaked in the slurry. Then the slurry jar comes down (or the supported layer comes up) and a film of slurry attaches to the supporting layer. After the coated slurry is dried in ambient temperature the supported layer with its new dip-coated layer are sintered.

The smoothness and thickness of the layer can be controlled by modifying the solid loads in the slurry and the draw-up speed of the supporting layer from the slurry jar. Layers with the thickness of a few microns to hundreds of microns can be produced using this method. The usual duration for soaking the supporting layer in the jar initially is about 30 seconds.

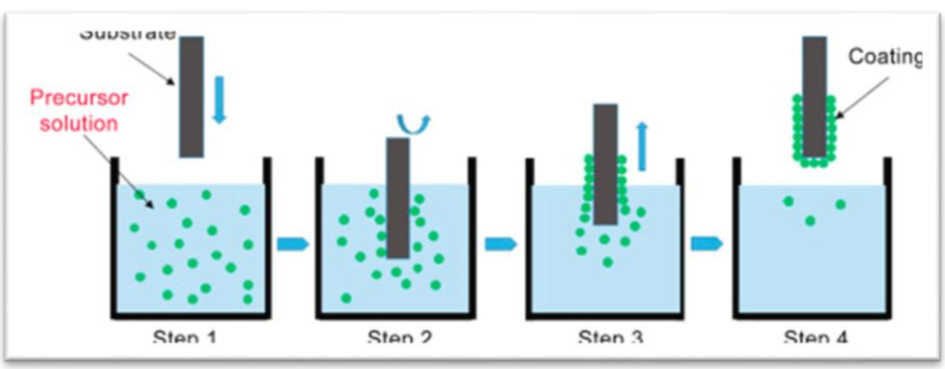
Yamaguchi reported that the density of the electrolyte layer coated on the anode-supported layer in a tubular SOFC is highly affected by the shrinkage of the sublayer during the cosintering process.

Figure below illustrates the dip-coating method.



Dip coating is a simple, low-cost, reliable and reproducible method which involves the deposition of a wet liquid film by immersion of the substrate into a solution containing hydrolysable metal compounds (or readily formed particles) and its withdrawal at constant speed into an atmosphere containing water vapour. After the removal of the substrate from the solution, a homogeneous liquid film is formed on the substrate's surface.

After drying at room temperature, the volatile solvents will be eliminated and possible chemical reactions will occur, resulting in a thin film of coating. Usually, after drying in a water atmosphere, the film needs hardening/chemical transformation by heat treatment.



Mesoporosity can be provided by the use of small nanoparticles (few nm to some tens of nm) in the precursor sol or by mixing the powder with a volatile component, for example, hydrogen peroxide or naphthalene, whose deposition in a compact, crack-free layer, gives rise to films with pore diameter between 2 and 50 nm (IUPAC designation of mesoporosity). The dip coating technique also allows the preparation of sophisticated architectures.

The coating thickness can also be varied and the influencing parameters are the viscosity of the solution, the rate of solvent evaporation and the angle at which the substrate is taken out. The evaporation depends on the nature of the solvent.

This method is similar to other wet chemical techniques, except the fact that the deposition is performed faster, within just a few seconds, depending on which solvents are used (Aegerter and Mennig, 2004).

Dip coating is intensively used in both industry and laboratory applications because this implies cheap raw materials and equipment, following easy steps and the results are good qualitatively. The main disadvantages are the non-uniformity of the coating thickness and the high-temperature sintering (usually >1000 °C), which can deteriorate mechanical properties of the metal implant and can lead to low bond strength.

HA is a versatile coating material that can be deposited on various metal substrates. Studies were already done on Ti–6Al–4V and cobalt alloys substrates, resulting in coatings with thicknesses varying from 0.05 to 0.5 mm. Interactions between the ceramic coating and metallic substrate were not observed. The starting material was chemically precipitated HA powder (Aminatun, 2015; Hijon, 2006).

The concentration of the suspension affects the thickness and homogeneity of the resulting thin film. The lower the concentration, the thinner the layer is generated and the stronger the adhesion between the coating material and the substrate is (Hijon, 2006; Mohseni, 2014).

Dip coating is also frequently used for the production of gloves. Fillers play a prominent role in dip coating formulations. The formulation must be designed to impart two opposite properties to the coating. It should have a relatively low viscosity to assist dipping process and the viscosity should rapidly increase after the form is withdrawn from the dipping tank.

Overview of Dip Coating

Dip coating involves the deposition of a liquid film via the precise and controlled withdrawal of a substrate from a solution using a dip coater. The dip coating process involves a minimum of four unique steps (or stages) followed an optional fifth curing step:

1. Immersion
2. Dwelling
3. Withdrawal
4. Drying

All these stages are essential in the dip coating process. However, the two critical points for determining the properties of the deposited film are the withdrawal and drying stages. The final film thickness is determined during these stages by the interplay between the entraining forces, draining forces, and the drying of the film. Films are formed in one of three regimes:

1. Viscous flow
2. Drainage
3. Capillary

The transitions between each of these regimes occur at varying values of withdrawal speed and solution viscosity. The combination of the three coating regimes ultimately determines the 'thickness vs withdrawal speed' behaviour for a thin film. By summing the contributions from the drainage regime and capillary regime, it is possible to obtain an equation that explains the thickness withdrawal speed relationship over a wide range of speeds. This also allows us to determine the minimum possible thickness that can be coated for a solution.

The substrate is lowered into a bath of solution until it is mostly or fully immersed. A short delay occurs before the substrate is withdrawn. During the withdrawal process, a thin layer of the solution remains on the surface of the substrate. Once fully withdrawn, the liquid from the film then begins to evaporate and leaves behind a dry film. For certain materials, a further curing step can be performed which forces the deposited material to undergo a chemical or physical change.

Dip Coating Theory

Withdrawal and Film Formation

The withdrawal stage of the dip coating process can be simply seen as the interaction of several sets of forces. These forces can be placed into one of two categories: draining forces and entraining forces. Draining forces work to draw the liquid away from the substrate and back towards the bath. Conversely, entraining forces are those that work to retain fluid onto the substrate. The balance between these sets of forces determines the thickness of the wet film coated onto the substrate. During the withdrawal stage, the formation of the wet film can be broken into four regions (shown in Figure 2).

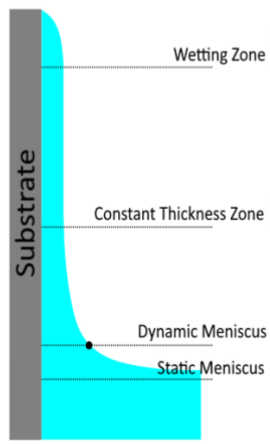


Figure 2. The dip coating film formation involves four distinct regions. These are the static meniscus, the dynamic meniscus, the constant thickness zone, and the wetting zone.

These four regions are:

1. The static meniscus, where the shape of the meniscus is determined by the balance of the hydrostatic and capillary pressures.
2. The dynamic meniscus, which occurs around the stagnation point. The stagnation point is where the entraining forces and draining forces are in equilibrium.
3. The constant thickness zone, where the wet film has reached a given thickness (h_0).
4. The wetting zone, which is the region where the wet film begins.

The dynamic meniscus and the flow of solution in this region determine the wet film thickness. Therefore, understanding the physics that underpins the curvature of the dynamic meniscus and the thickness of the stagnation point are important.

Figure 3 (below) shows the dynamic meniscus region. The transition between the static and dynamic meniscus occurs within the boundary layer (L). In this region, the forces from viscous flow impact the way the solution moves. Beyond the boundary layer, the draining forces are significantly greater than those of the viscous forces. In this region, it is the balance between the hydrostatic and capillary pressure that determines the meniscus.

The stagnation point occurs when the balance between the entraining forces and the draining forces are equal. It is the balance of these forces that determine the film thickness. There are three different coating regimes that are defined by which forces dominate the behavior of the coating - these are the viscous flow, draining, and capillary regimes.

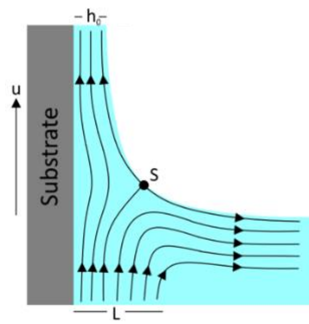


Figure 3. The flow of solution is determined by the balance of entraining and draining forces. At the dynamic meniscus, the entraining forces begin to impact solution flow until they are the dominant force.

Viscous Flow and Drainage Regime

The first coating regime is the viscous flow regime. This occurs for high velocities and viscous solutions. Here, the coating is dominated by viscous forces and gravitational attraction.

The equation for calculating the wet film thickness for dip coating while in the viscous flow regime.

Here, the entraining force is made up of the viscous forces acting on the solution as the substrate is withdrawn. This is given by the viscosity (η) and the rate of withdrawal speed of the substrate from the solution (U_0). The draining force is that of gravity, given by the density of the solution (ρ) and the gravitational constant (g). The constant (c) is related to the curvature of the dynamic meniscus. This constant is a property of the solution itself, and is strongly related to the rheological properties of the solution. For most Newtonian liquids, this constant is around 0.8.

In most situations, the withdrawal speeds or the viscosity of the solutions used will not be high enough for this approximation to be valid. When these two variables are reduced, the viscous force becomes weaker. The balance between the entraining forces and the draining force is then also dependent upon surface-tension driven movement of the solution. It is under these conditions that the coating is said to be within the drainage regime. The Landau-Levich equation (refer to Equation 2) describes the relationship between the wet film thickness and the withdrawal speed of the substrate (when surface tension is considered).

Equation 2. The Landau-Levich equation is a modified form of the viscous flow equation, which takes account of surface-tension driven flow.

The Landau-Levich equation is valid until very low withdrawal speeds are considered. When the speed is reduced to lower than approximately 0.1mm.s^{-1}, a third regime of coating occurs. This regime is known as the capillary regime. In the capillary regime, the rate at which solution is entrained onto the substrate (through viscous flow) is lower than the rate of evaporation. Therefore, the dynamics of drying are crucial to understanding the capillary regime.

2.2 Selecting Microprocessor

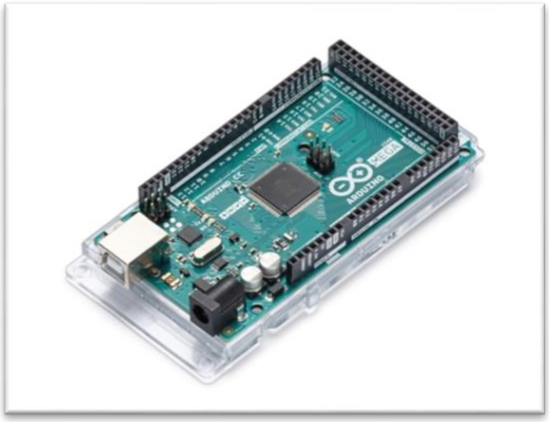
Need for a microprocessor in our project

A device that uses a microprocessor is normally capable of many functions, such as word processing, calculation, and communication via Internet or telephone. However, for the device to work properly, the microprocessor itself has to communicate with other parts of the device.

For our project we needed a microprocessor to take care of the working of all the components in the project. The microcontroller ensures that all the functionalities in the device are in synchronization with each other, like taking input from keyboard matrix and using that to run the motor with varied parameters like speed, distance and delay.

The microprocessor in our project controls and synchronizes all functionalities including speed control, delay, and distance motion control.

Our choice of microprocessor is Arduino Mega.



Why we chose Arduino Mega over any other microprocessor?

1. **Ready to Use:**

The biggest advantage of Arduino is its ready to use structure. As Arduino comes in a complete package form, there is no need to think about programmer connections for programming or any other interface. Just plug it into USB port of a computer and that's it.

2. **Examples of codes:**

Another big advantage of Arduino is its library of examples present inside the software of Arduino. So if we want to measure the voltage using Arduino. We can just plug in the Arduino and open the Read Analog Voltage example.

3. **Effortless functions:**

There is a vast library of readily available functions in the Arduino IDE which further contributed to the selection of Arduino as our first choice, another advantage of Arduino is its automatic unit conversion capability. That ensures that during debugging we don't have to worry about the units conversions. We can use all our force on the main parts of our projects and not worry about the side problems.

4. **Large community:**

There are many forums present on the internet in which people are talking about the Arduino. Engineers, hobbyists and professionals are making their projects through Arduino. We can easily find help about everything. Moreover the Arduino website itself explains each and every functions of Arduino.

EXPLORE INNOVATE EXCEL

CHAPTER 3

HARDWARE AND SOFTWARE USED

3.1 HARDWARE USED

1. ARDUINO MEGA

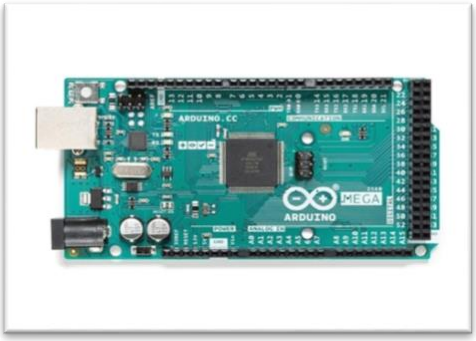


Fig. Arduino Mega

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila.

MICROCONTROLLER	ATmega2560
OPERATING VOLTAGE	5V
INPUT VOLTAGE (RECOMMENDED)	7-12V
INPUT VOLTAGE (LIMIT)	6-20V
DIGITAL I/O PINS	54 (of which 15 provide PWM output)
ANALOG INPUT PINS	16
DC CURRENT PER I/O PIN	20 mA
DC CURRENT FOR 3.3V PIN	50 mA
FLASH MEMORY	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
CLOCK SPEED	16 MHz
LED_BUILTIN	13
LENGTH	101.52 mm
WIDTH	53.3 mm
WEIGHT	37 g

2. L – CLAMP

Fig. L-Clamp

A clamp is a fastening device used to hold or secure objects tightly together to prevent movement or separation through the application of inward pressure. The product that we've used is L shaped that is why they it is called L clamp. They are very light in their weight and are available at different shapes and sizes in the market. The important advantage of these products is that their body is made up of stainless steel which makes them very strong.

3. STEPPER MOTOR

Fig. Stepper Motor

A stepper motor, also known as step motor or stepping motor is a brushless DC electric motor that divides a full rotation into a number of equal steps. The motor's position can be commanded to move and hold at one of these steps without any position sensor for feedback (an open-loop controller), as long as the motor is correctly sized to the

EXPLORE INNOVATE EXCEL

application in respect to torque and speed. The stepper motor used here is a NEMA 17, with a holding torque of 4.2 Kg-cm. Holding torque is the torque required to force a stationary rotor out of a magnetized core and into the next position.

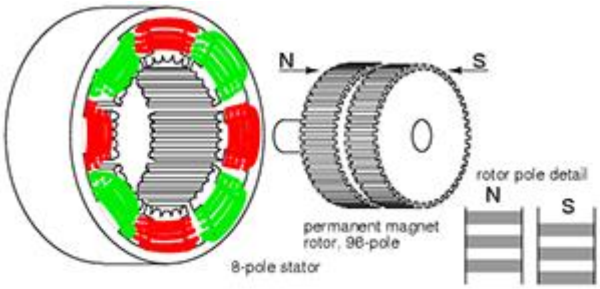
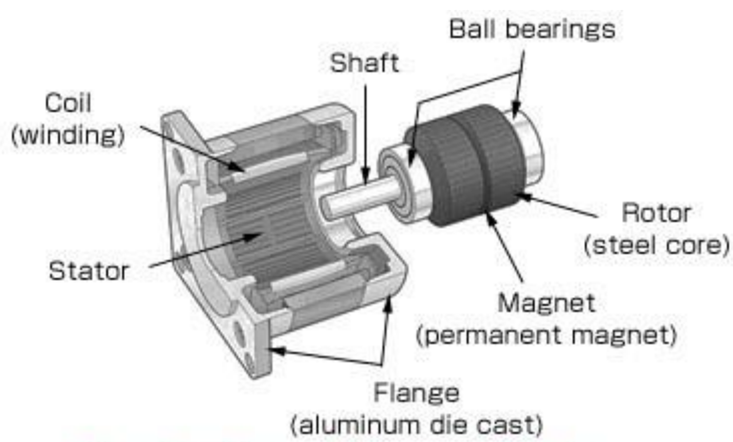


Fig. Holding Torque

It works on 12V – 24V DC and has a step angle of 1.8 Degree and operating current of 1.7A per phase. The invention of dedicated stepper motor driver cards and other digital control technologies for interfacing a stepper motor to PC-based systems are the reason for the widespread acceptance of stepper motors in recent times. Stepper motors become the ideal choice for automation systems that require precise speed control or precise positioning or both.

As we know that many industrial electric motors are used with closed-loop feedback control for achieving precise positioning or precise speed control, on the other hand, a stepper motor able to operate on an open-loop controller. This in turn reduces the total system cost and simplifies the machine design compared with servo system control.



Construction of Stepper Motor

Stepper Motor is a brushless electromechanical device which converts the train of electric pulses applied at their excitation windings into precisely defined step-by-step mechanical shaft rotation. The shaft of the motor rotates through a fixed angle for each discrete pulse. This rotation can be linear or angular. It gets one step movement for a single pulse input. When a train of pulses is applied, it gets turned through a certain angle. The angle through which the stepper motor shaft turns for each pulse is referred as the step angle, which is generally expressed in degrees.

The number of input pulses given to the motor decides the step angle and hence the position of motor shaft is controlled by controlling the number of pulses. This unique feature makes the stepper motor to be well suitable for open-loop control system wherein the precise position of the shaft is maintained with exact number of pulses without using a feedback sensor.

If the step angle is smaller, the greater will be the number of steps per revolutions and higher will be the accuracy of the position obtained. The step angles can be as large as 90 degrees and as small as 0.72 degrees, however, the commonly used step angles are 1.8 degrees, 2.5 degrees, 7.5 degrees and 15 degrees.

The direction of the shaft rotation depends on the sequence of pulses applied to the stator. The speed of the shaft or the average motor speed is directly proportional to the frequency (the rate of input pulses) of input pulses being applied at excitation windings. Therefore, if the frequency is low, the stepper motor rotates in steps and for high frequency, it continuously rotates like a DC motor due to inertia.

Like all electric motors, it has stator and rotor. The rotor is the movable part which has no windings, brushes and a commutator. Usually the rotors are either variable reluctance or permanent magnet kind. The stator is often constructed with multipole and multiphase windings, usually of three or four phase windings wound for a required number of poles decided by desired angular displacement per input pulse.

Unlike other motors it operates on a programmed discrete control pulses that are applied to the stator windings via an electronic drive. The rotation occurs due to the magnetic interaction between poles of sequentially energized stator winding and poles of the rotor.

4. COUPLING

Fig. Coupling (5x8)

Coupling Shaft can connect the driving shaft with the driven shaft. 5x8mm. First bore diameter is 5mm and the second bore diameter is of 8mm also.

It comes with the spiral cut in the middle along its length which makes it flexible so they can be fit to two shafts even if they are not perfectly co-linear and will help reduce binding effects.

Descriptions:

D (Outside diameter): 19mm

L (Inside Length): 25mm

ID (Inside diameter): 8mm

5. WIRES

Fig. Wires

A wire is a single usually cylindrical, flexible strand or rod of metal. Wires are used to bear mechanical loads or electricity and telecommunications signals. Wire is commonly formed by drawing the metal through a hole in a die or draw plate. Each cable length about 20cm or 8-inch. The male ends meant for insertion into standard 0.1 inch (2.54mm) female sockets and the female ends are meant for insertion onto standard 0.1 inch (2.54mm) male headers. The cables can be separated to form an assembly containing the number of wires you require for your connection and to support non-standard odd-spaced headers

6. STEEL ROD

Fig. Steel Rod

Metal rods are metals and alloys designed in the pattern of round bars or rod, rectangular or flat bars, square bars, hexagons, and other patterns of bar stock. These shapes also come in billet form and generally include a cross-section based on the shape of rod or bar stock.

Descriptions:

D (Outside diameter): 8mm

L (Inside Length): 300mm

7. ROTATING SCREW

Fig. Rotating Screw

A rotating jack has a lift shaft that moves a nut as it turns. The lift shaft is fixed to the worm gear. This causes the load, which is attached to the travel nut, to move along the lift shaft. A screw is a mechanism that converts rotational motion to linear motion, and a torque (rotational force) to a linear force.

8. OLED Display Screen

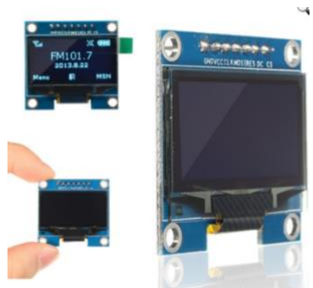


Fig. OLED Display Screen

This 1.3 Inch 128×64 OLED Display Screen Module offers 128×64 pixel resolution. They are featuring much less thickness than LCD displays with good brightness and also produce better and true colors. This OLED Display Module is very compact. The connection of this display with Arduino is made through SPI interface. The 1.3 Inch 128×64 OLED Display Screen Module produces blue text on black background with very good contrast when supplied with DC 2.8V supply. The OLED Display Modules also offers a very wide viewing angle of about greater than 160°.

Features:

1. No need of the backlight
2. The display is self-illuminating
3. Power requirement is low
4. Offers the large viewing angle
5. Full Compatible with Arduino
6. The ultra-low power consumption of around 0.08w at full screen lit.

9. TFT Display Screen

Fig. TFT Display Screen

2.4″ Inch Touch Screen TFT Display Shield adds a touch up to your Arduino project with a beautiful large touchscreen display shield with built-in microSD card connection. This TFT display is big (2.4″ diagonal) bright and colorful! 240×320 pixels with individual pixel control. It has way more resolution than a black and white 128×64 display.

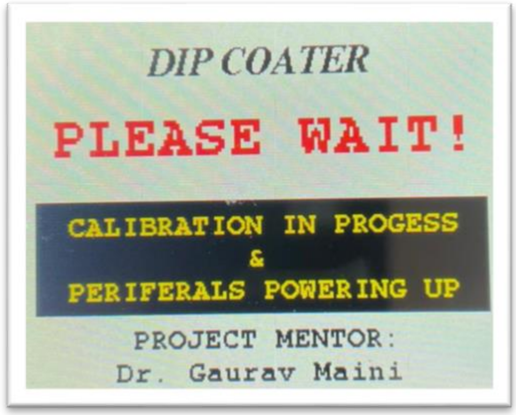
Features:

1. 2.4″ diagonal Touch LCD TFT display
2. 240×320 resolution, 18-bit (262,000) color
3. 8-bit digital interface, plus 4 control lines
4. 5V compatible! Use with 3.3V or 5V logic
5. Onboard 3.3V 300mA LDO regulator
6. Uses digital pins 5-13 and analog 0-3. That means you can use digital pins 2, 3, and analog 4 and 5. Pin 12 is available if not using the microSD
7. Works with any Arduino Uno and mega compatible boards

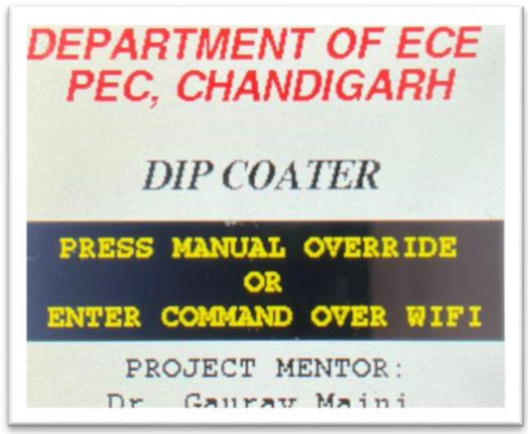
Graphic User Interface Construction

The user interface was made pixel by pixel using functions of the available U8glib library. They were tried to be made as intuitive as possible. Red color was used to express error. Green was for success. Yellow with black text was used to enhance the importance of the screen on which parameters were being entered. The screens were as under:

ENTRY SCREEN TO PRODUCE DELAY FOR CALIBRATION



MAIN SCREEN TO DECIDE FOR MANUAL OVERRIDE

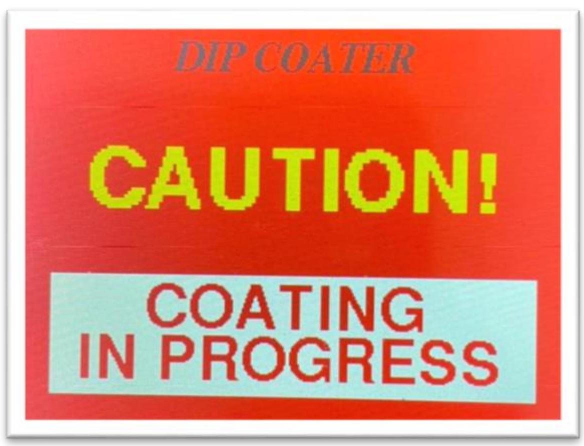


READY SCREEN TO SHOW THAT CALIBRATION HAS TAKEN PLACE SUCCCESSFULLY

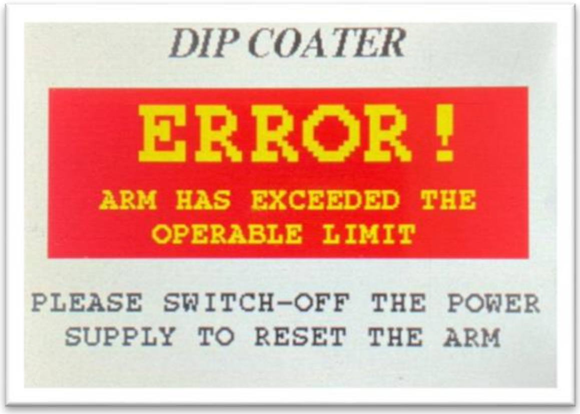
DATA RECEPTION SCREEN FOR WIFI MODE

DIP COATER PARAMETER ENTERING SCREEN FOR KEYPAD MODE

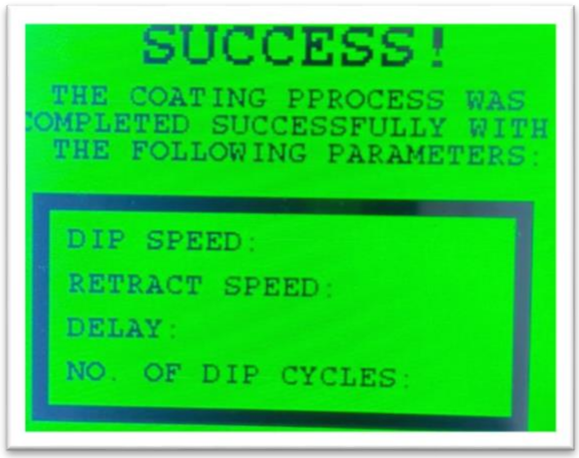
CAUTION SCREEN WHILE COATING IS TAKING PLACE



ERROR SCREEN WHILE END-STOP SWITCHES WER ACTIVATED



SUCCESS SCREEN TO SIGNIFY PROCESS PARAMETERS AND SUCCCESSFUL COMPLETEION OF COATING PROCESS



10. MATRIX KEYPAD

Fig. Matrix Keypad

A matrix keypad is a small compact input device that accepts user inputs and processed by Microcontrollers. You might have seen this in most commonly used devices like Calculators, Digital locks, Gas pumps and DIY projects. It comes in different types, one of them is membrane keypads, it is thinner in size and you can paste it on top of your creative projects.

Most of the applications of embedded systems require keypads to take the user inputs, especially in the case where an application requires more keys. With simple architecture and easy interfacing procedure, matrix keypads are replacing normal push buttons by offering more inputs to the user with the lesser I/O pins. As a Human Machine Interface (HMI) keypad plays a major role in vital microprocessor and microcontroller-based projects and equipment. Therefore, this article gives you a brief idea about the matrix keypad.

A Matrix keypad is the most commonly used input device in many of the application areas like digital circuits, telephone communications, calculators, ATMs, and so on. A matrix keypad consists of a set of push-button or switches which are arranged in a matrix format of rows and columns. These keypads are available in configurations like 3×4 and 4×4 based on the application it is implemented for.

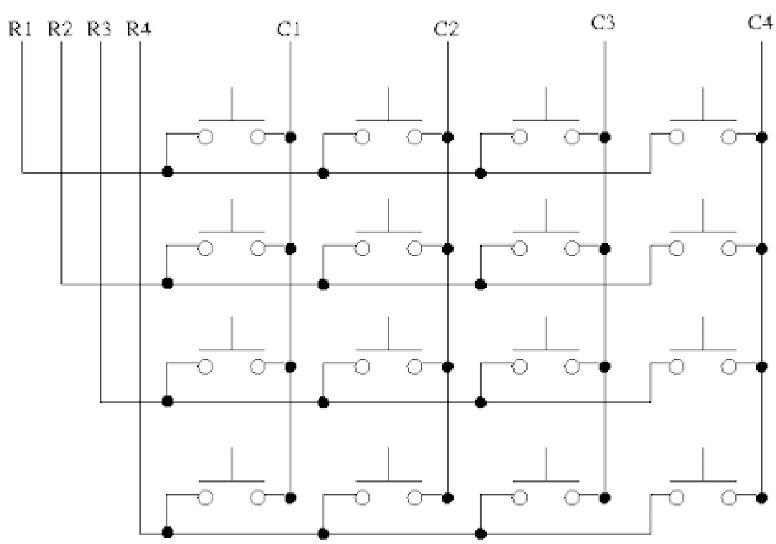


Fig. Matrix Keypad Schematic

The status of each key can be determined by a process called Scanning. There are many methods depending on how we connect our keypad with our controller, but the basic logic is the same. For the sake of explanation, let's assume that all the column pins (Col1 – Col4) are connected to the inputs pins and all the row pins are connected to the output pins of the microcontroller. In the normal case, all the column pins are pulled up (HIGH state) by internal or external pull-up resistors. Now we can read the status of each switch through scanning.

A logic LOW is given to Row1 and others (Row2 – Row-4) HIGH. Now each Column is scanned. If any switch belongs to the 1st row is pressed the corresponding column will pull down (logic LOW) and we can detect the pressed key. This process is repeated for all rows.

11. POWER SUPPLY

Fig. Power Supply

A power supply is an electrical device that supplies electric power to an electrical load. The primary function of a power supply is to convert electric current from a source to the correct voltage, current, and frequency to power the load. As a result, power supplies are sometimes referred to as electric power converters.

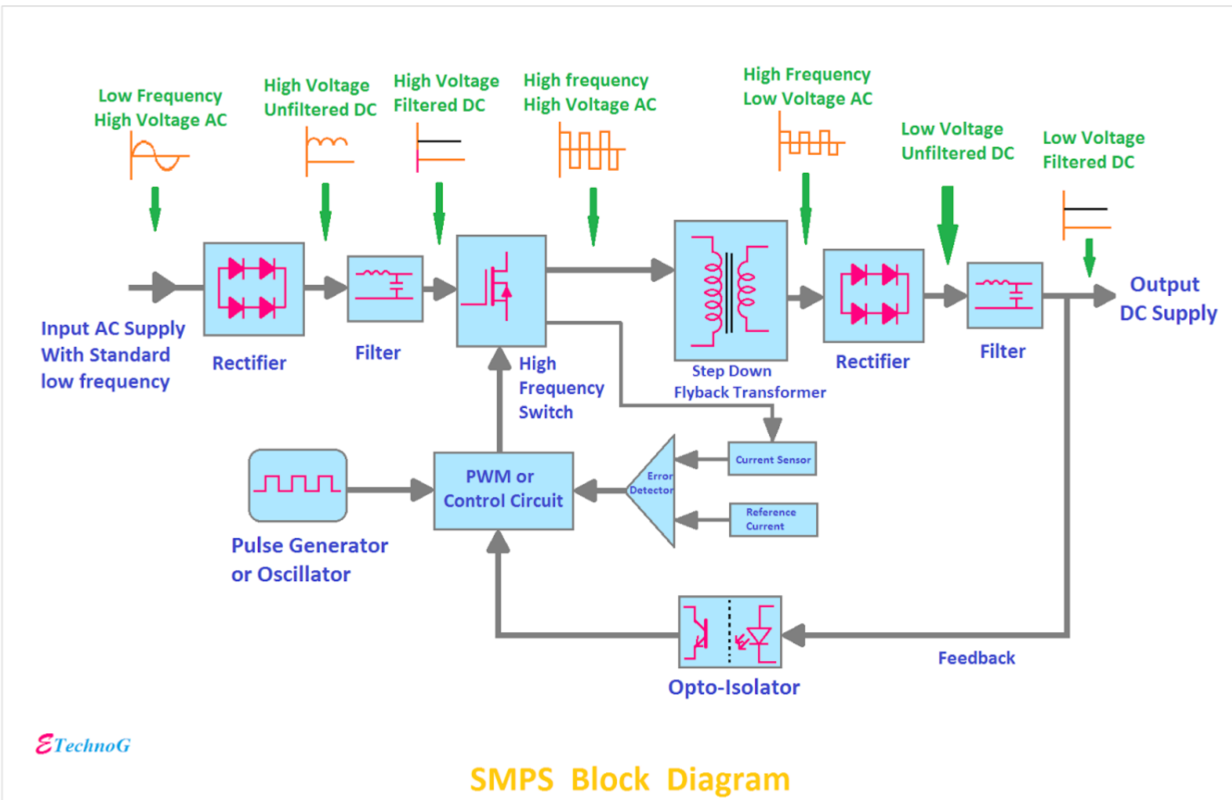
Power supplies are of many types these days, namely:

- Simple Full Bridge Rectification transformer based power supplies
- Switching Supplies: Buck, Boost, Buck-Boost Power supply
- Capacitive Power Supplies
- Switch Mode Power Supplies.

and many more.

A switched-mode power supply (switching-mode power supply, switch-mode power supply, switched power supply, SMPS, or switcher) is an electronic power supply that incorporates a switching regulator to convert electrical power efficiently.

Like other power supplies, an SMPS transfers power from a DC or AC source (often mains power, see AC adapter) to DC loads, such as a personal computer, while converting voltage and current characteristics. Unlike a linear power supply, the pass transistor of a switching-mode supply continually switches between low-dissipation, full-on and full-off states, and spends very little time in the high dissipation transitions, which minimizes wasted energy. A hypothetical ideal switched-mode power supply dissipates no power. Voltage regulation is achieved by varying the ratio of on-to-off time (also known as duty cycles). In contrast, a linear power supply regulates the output voltage by continually dissipating power in the pass transistor. This higher power conversion efficiency is an important advantage of a switched-mode power supply.



The supplies used here, in the project are SMPS power supplies. Hey basically use the technique of first rectifying 220V AC to DC and then using a MOSFET to switch this DC voltage and then a transformer to step it down to usable level and further on. This is only been done to reduce the size of the transformer and increase the efficiency. The two supplies used here are rated as 12V @ 5A and 5V @ 1A.

Calculations for the 12V @ 5A Power Supply

NEMA 17 Power Requirements = 12V @ 1.2A – 1.7A, thus the above will suffice.

Calculations for the 5V @ 1A Power Supply

Arduino Mega = 5V @ 150mA – 200 mA

Nokia LCD = 3.3V @ 50 mA

Total = 1 W + 0.165 W = 1.165W <5W

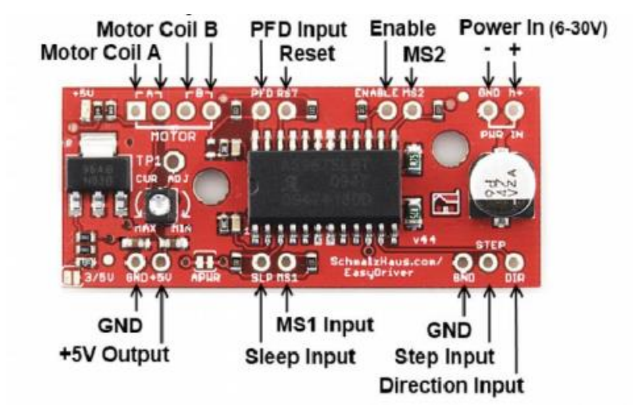
Thus the above will suffice.

12. ESP8266: WIFI MODULE

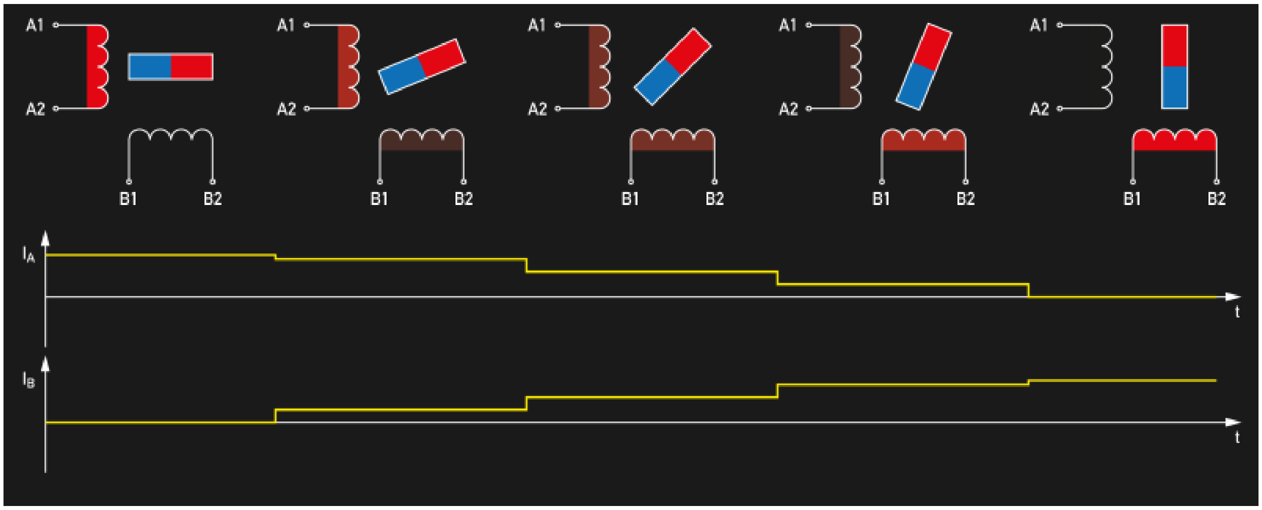
Fig. Wi-Fi Module

ESP8266 is a WIFI SOC (system on a chip) produced by Espressif Systems. It is a highly integrated chip designed to provide full internet connectivity in a small package. ESP8266 can be used as an external WIFI module, using the standard AT Command set Firmware by connecting it to any microcontroller using the serial UART, or directly serve as a WIFI-enabled micro controller, by programming a new firmware using the provided SDK. The GPIO pins allow Analog and Digital IO, plus PWM, SPI, I2C, etc. This board has been around for almost a year now, and has been used mostly in IoT contexts, where we want to add connectivity for example to an Arduino project. A wide adoption has been facilitated by the very modest price.

13. EASY DRIVER: STEPPER DRIVER



The Easy Driver stepper motor is a stepper driver unit which uses the A3967 IC to perform mircostepping operations on the stepper motor. This is achieved through Pulse width modulation.



Pinout of the Easy Driver:

- **GND :** There are three GND (Ground) pins on the Easy Driver. They are all connected together inside the board
- **M+ :** This is the power input to the Easy Driver. Connect this to the positive power supply lead. This should be a 6V to 30V, 2A (or more) power supply that is clean (low ripple).
- **A and B :** (four pins) These are the motor connections. A and B are the two coils of the motor, and can swap the two wires for a given coil (it will just reverse the direction of the motor). It defaults to 8 step micro stepping mode. (So if the motor is 200 full steps per revolution, you would get 1600 steps/rev using Easy Driver.) This setting can be easily overridden by tying the MS1 and/or MS2 pin to ground to set the driver to use 1/8, 1/4 or 1/2 microstep mode.
- **STEP :** This needs to be a 0V to 5V (or 0V to 3.3V if you've set your Easy Driver that way) digital signal. Each rising edge of this signal will cause one step (or microstep) to be taken.
- **DIR (Direction) :** This needs to be a 0V to 5V (or 0V to 3.3V if you've set your Easy Driver up that way) digital signal. The level if this signal (high/low) is sampled on each rising edge of STEP to determine which direction to take the step (or microstep).

14. END STOP SWITCH

Fig. End Stop Switch

Endstop Switch uses a lever switch to detect when it is activated. The switch is wired up so that when activated, it pulls the signal to LOW. There is also an LED on the board that will light up when the switch is activated. It uses a standard 4 pin 100″ pitch header and accepts a standard, old-style CD-ROM audio connector cable.

The mechanical endstop is a simple solution to a simple problem. We want to be able to detect when an X/Y/Z stage has reached its minimum or maximum. Instead of messing with flags or complicated light beam interruption, we use a mechanical switch.

If we place the switch in the path of the stage, then the stage itself will simply close the switch when it moves against it. Other than properly positioning the switch, we do not need to modify the stage at all. If you're worried about reliability, you can sleep well at night. The switches we use are rated for 1 million operations before failure. Since we only use the switches once per print, that means you'll be able to do one million prints before having to replace the switch

Features:

1. Not easy to pull off, plug it convenient, for ease of use, plug the other side to do the one, you can use the direct plug.
2. Quality switching elements.
3. Customized high-quality connection wires using 22AWG copper wire with ratings of current 2A, 300V voltage.
4. An insulating layer over a wide range of temperatures, up to 80 degrees.

15. HUMIDITY AND TEMPERATURE SENSOR

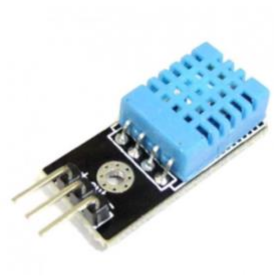


Fig. Humidity And Temperature Sensor

DHT11 digital temperature and humidity sensor is a composite Sensor contains a calibrated digital signal output of the temperature and humidity. Application of a dedicated digital modules collection technology and the temperature and humidity sensing technology, to ensure that the product has high reliability and excellent long-term stability. The sensor includes a resistive sense of wet components and an NTC temperature measurement devices, and connected with a high-performance 8-bit microcontroller.

Applications: HVAC, dehumidifier, testing and inspection equipment, consumer goods, automotive, automatic control, data loggers, weather stations, home appliances, humidity regulator, medical and other humidity measurement and control.

Features:

- Low cost,
- long-term stability,
- relative humidity and temperature measurement,
- excellent quality,
- fast response,
- strong anti-interference ability,
- long distance signal transmission,
- digital signal output, and
- precise calibration.

16. COMBINING EVERYTHING TOGETHER

- Every component of the circuit was combined together on a PCB perforated board through a meticulous soldering process. The Arduino MEGA board and Arduino UNO board were inverted for efficient soldering and wire management. This way, jumpers were eliminated from the circuit. Some important aspects or bottle necks where error could have crept in was:

  - The Wifi-Module connection with the Arduino Mega. The RX pin of the Arduino MEGA goes into the TX pin of the Wifi-Module through a resistive divider network and the TX pin of the Arduino Board goes in the RX pin of the module.

  - The Wifi module needs to be powered separately from a 3.3V regulator as the TFT LCD screen draws excessive amounts of current from the Arduino Mega hence not leaving ample current for the Wifi-module.

  - The Matrix Keyboard connections need care as they are very sensitive too erroneous touches.

  - The Accel Stepper driver IC gets heated very quickly to a high temperature this needing an efficient heat removal method.

  - The interrupt switch needs to be a push-button switch that helps it to not interrupt more than once whence pressed.

  - End-Stop switch placement should be done with care.

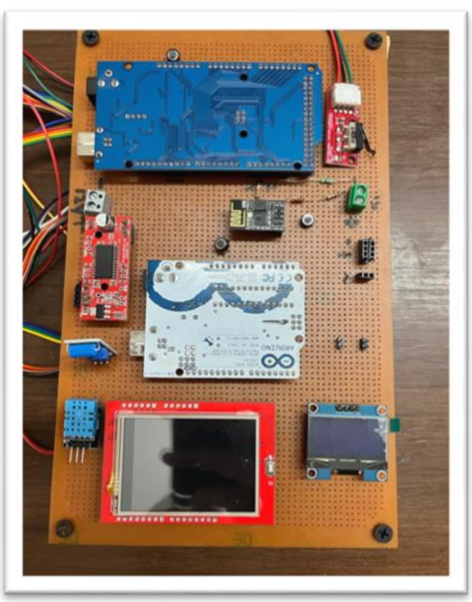


Fig. Hardware Unit

3.2 SOFTWARE USED

1. ARDUINO IDE

The **Arduino Integrated Development Environment (IDE)** is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program Arduino to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware. By default, Arduino is used as the uploading tool to flash the user code onto official Arduino boards.

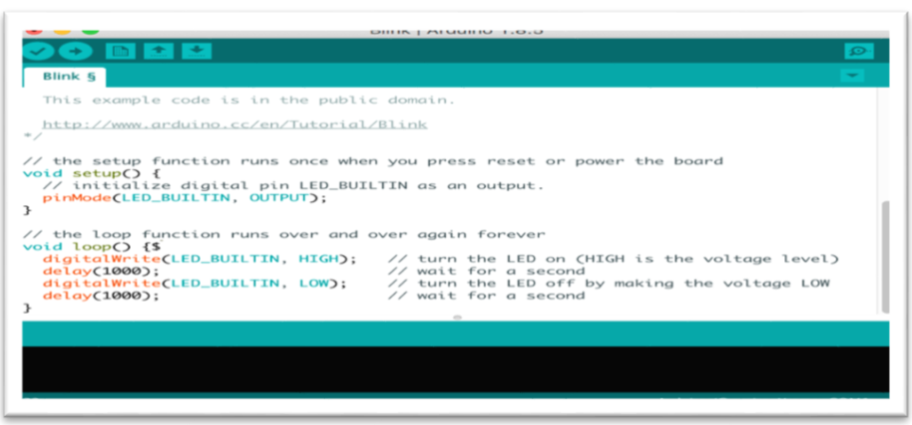


Fig: Screenshot of the Arduino IDE showing a blink program

2. HTML

HTML (Hyper Text Markup Language) is the computer language that facilitates website creation. The language, which has code words and syntax just like any other language, is relatively easy to comprehend and, as time goes on, increasingly powerful in what it allows someone to create. HTML continues to evolve to meet the demands and requirements of the Internet under the guise of the World Wide Web Consortium, the organization that designs and maintains the language; for instance, with the transition to Web 2.0.

HyperText is the method by which Internet users navigate the web. By clicking on special text called hyperlinks, users are brought to new pages. The use of hyper means it is not linear, so users can go anywhere on the Internet simply by clicking on the available links. Markup is what HTML tags do to the text inside of them; they mark it as a specific type of text. For example, markup text could come in the form of boldface or italicized type to draw specific attention to a word or phrase.

At its core, HTML is a series of short codes typed into a text-file. These are the tags that power HTML's capabilities. The text is saved as an HTML file and viewed through a web browser. The browser reads the file and translates the text into a visible form, as directed by the codes the author used to write what becomes the visible rendering. Writing HTML requires tags to be used correctly to create the author's vision. The tags are what separate normal text from HTML code. Tags are the words between what are known as angle-brackets, which allow graphics, images, and tables to appear on the webpage. Different tags perform different functions. The most basic tags apply formatting to text. As web interfaces need to become more dynamic, Cascading Style Sheets (CSS) and JavaScript applications may be used. CSS makes web pages more accessible and JavaScript adds power to basic HTML.

3. JAVASCRIPT

JavaScript is a programming language commonly used in web development. It was originally developed by Netscape as a means to add dynamic and interactive elements to websites. While JavaScript is influenced by Java, the syntaxis more similar to C and is based on ECMAScript, a scripting language developed by Sun Microsystems.

JavaScript is a client-side scripting language, which means the source code is processed by the client's web browser rather than on the web server. This means JavaScript functions can run after a webpage has loaded without communicating with the server. For example, a JavaScript function may check a web form before it is submitted to make sure all the required fields have been filled out. The JavaScript code can produce an error message before any information is actually transmitted to the server.

Like server-side scripting languages, such as PHP and ASP, JavaScript code can be inserted anywhere within the HTML of a webpage. However, only the output of server-side code is displayed in the HTML, while JavaScript code remains fully visible in the source of the webpage. It can also be referenced in a separate .JS file, which may also be viewed in a browser.

4. CSS

Stands for "Cascading Style Sheet." Cascading style sheets are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML.

CSS helps Web developers create a uniform look across several pages of a Web site. Instead of defining the style of each table and each block of text within a page's HTML, commonly used styles need to be defined only once in a CSS document. Once the style is defined in cascading style sheet, it can be used by any page that references the CSS file. Plus, CSS makes it easy to change styles across several pages at once. For example, a Web developer may want to increase the default text size from 10pt to 12pt for fifty pages of a Web site. If the pages all reference the same style sheet, the text size only needs to be changed on the style sheet and all the pages will show the larger text.

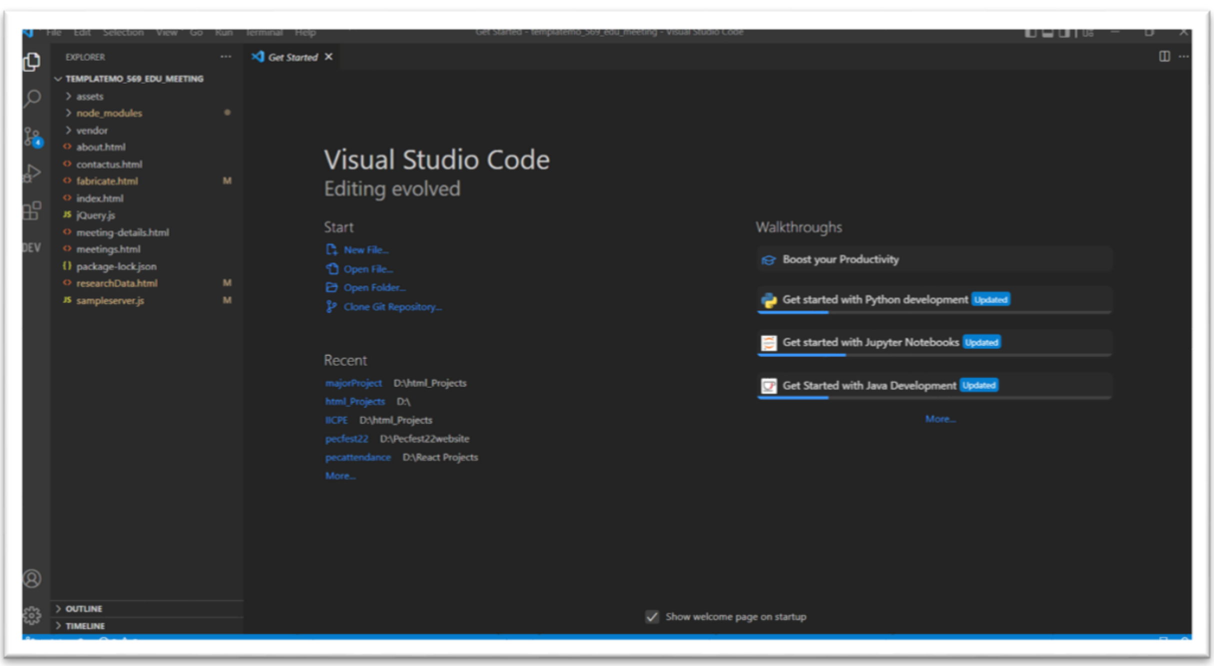
While CSS is great for creating text styles, it is helpful for formatting other aspects of Web page layout as well. For example, CSS can be used to define the cell padding of table cells, the style, thickness, and color of a table's border, and the padding around images or other objects. CSS gives Web developers more exact control over how Web pages will look than HTML does. This is why most Web pages today incorporate cascading style sheets.

5. POSTGRESQL

PostgreSQL is an open-source, object-relational database management system (ORDBMS) that is not owned or controlled by one company or individual. Because postgresSQL software is open-source, it is managed mostly through a coordinated online effort by an active global community of developers, enthusiasts and other volunteers.

First released in the mid-1990s, postgresSQL is written in C. Its primary competitors include Oracle DB, SQL Server and MySQL.

6. VS CODE



Visual Studio Code (famously known as VS Code) is a free open source text editor by Microsoft. VS Code is available for Windows, Linux, and macOS. Although the editor is relatively lightweight, it includes some powerful features that have made VS Code one of the most popular development environment tools in recent times. VS Code supports a wide array of programming languages from Java, C++, and Python to CSS, Go, and Docker file. Moreover, VS Code allows you to add on and even creating new extensions including code linters, debuggers, and cloud and web development support.

EXPLORE INNOVATE EXCEL

CHAPTER 4

WORKING OF THE PROJECT

WORKING

The working of the project can be divided into sub sections that are as under:

1. **The Mechanics.**
2. **The Electronics.**
3. **Overall Structure.**
4. **Motor Mechanism.**
5. **Rotor/ Shaft System.**
6. **The Web App**

The Mechanics

The Mechanics of this project has perhaps been some of the toughest ever to deal with. The reason is that as electronics students we are not well versed with the technicalities of the mechanical engineering domain. But, which working project is ever void of mechanical constraints and technicalities. The fact of the matter is that, the complexity involved in moving the die up and down made us learn a lot about the importance of understanding torque and basically the way how force is applied. The primary concern of this project was to grasp, firstly how to first move the semiconductor wafer up and down, with precision and also be able to do it without fail. This was a play of motors. Somewhere we knew with conviction that whenever we move something in projects, we need motors, but here we needed precision, so then came in the stepper motor. Secondly how to use rotational motion and convert it to vertical/ linear motion. To summarize, the system uses rotational motion through a stepper motor which is coupled to a threaded screw on which sits a stud which is further coupled to a wooden rack which is guided through two guide rails which guide its vertical motion. Thirdly, we realized that simply a threaded screw with a stud will not be sufficient to suffice for conversion of rotational to linear motion, will need guide rails on the side to actually help constrain motion in the rotational direction to perform the aforementioned conversion. The mechanics of the project had shown a major road block in the primitive stages of the project. This is explained as under.

Motion: Conversion from Rotational to Linear Motion

In physics, motion is the phenomenon in which an object changes its position. Motion is mathematically described in terms of displacement, distance, velocity, acceleration, speed, and time. The motion of a body is observed by attaching a frame of reference to an observer and measuring the change in position of the body relative to that frame with change in time. The branch of physics describing the motion of objects without reference to its cause is kinematics; the branch studying forces and their effect on motion is dynamics. Here, we talk about the motion of a motor, specifically a stepper motor. As things currently stand, by and large, robotic motion is created by motors. By employing motors in creative ways, an unimaginable amount of robotic automation is possible.

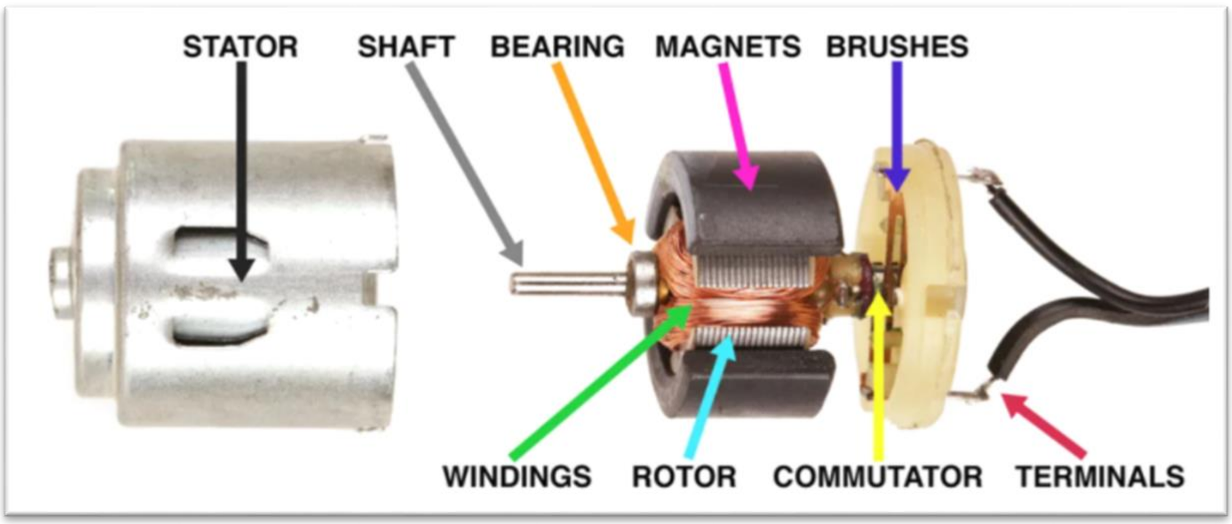


Fig. Stator Components

Here the motor that we used was the stepper motor, because we wanted precision in the rotational motion. As guided very correctly by our mentor, Dr Gaurav Mani Khanal, the dip coaters primary requirement was the

"Precision in dip speed, time of dipping and retraction speed"

This precision can only be achieved by a stepper motor because of the way it is constructed. Now after we knew that we needed a stepper motor to perform the motion, we knew that the next task was to convert this motion from rotational to linear.

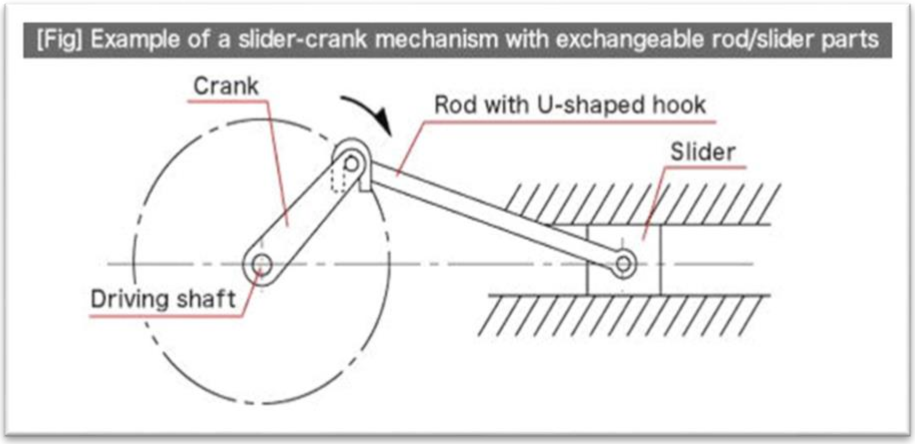


Fig. Slider Crank Mechanism

If we take notice of steam locomotives, they use their coal powered engines to drive the train. When we talk about pistons, they move up and down, but the drive a crank shaft which shows rotational motion. So if this is true them the converse is true too. This is basically where we got the idea of constraining the motion in the rotational direction to achieve vertical movement.

Here, the plate which is connected between the stud and the wooden platform to support the wafer dipping apparatus and the guide rails are constraining the rotational direction and allowing vertical motion. This is how we are converting rotational to vertical motion.

The Electronics

Though realizing the constraints of the Arduino Mega of the latency and other issues, we still decided to use it as the main micro-controller for prototyping purpose just by factoring its ease of use. Arduino has some sound advantages, such as large user community, free and broad ranges of libraries of codes, relatively low cost components, and so forth.

Its disadvantages are its small and a user has to work in a relatively small (or rather tiny) space. In many broad and multi-purpose projects, its required to look for third party sources in addition to Arduino scripts. Another disadvantage is that in many cases, some errors remain quite persistent and error messages may not be quite useful. Its usefulness for learning purposes and developing really complex boards is very sound.

The need of using the Mega was the part that it has many more Analog and Digital ports. We anticipated that in the future there might be OLED displays which use around 16-20 ports, so this will eventually help us to extend the project in a proper direction without any diversion.

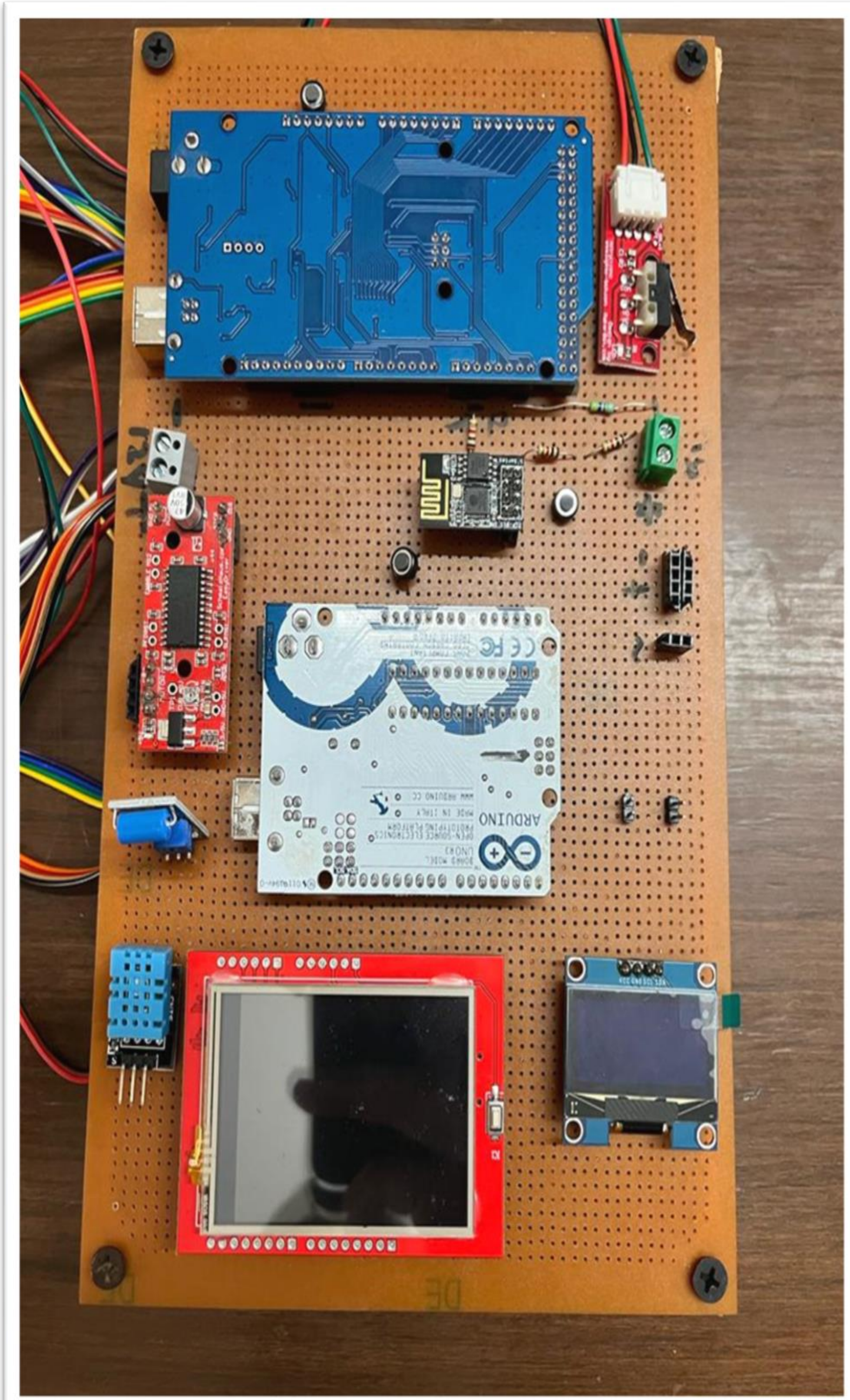


Fig. Electronic Components

Overall Structure

The overall structure is a teak ply board veneered with a brown coloured paint with lacquer on top. This saves the structure from damage due to inadvertent spillage of dipping contents. Basically the structure is a base of wood which has a perpendicular base of again wood to support vertical movement. The various rod clamps provide support to the guide metal rails and the L-clamps provide support to the stepper motor and the bevel stud. The dipping apparatus is supported by a wooden plank mounted on the rotating stud.

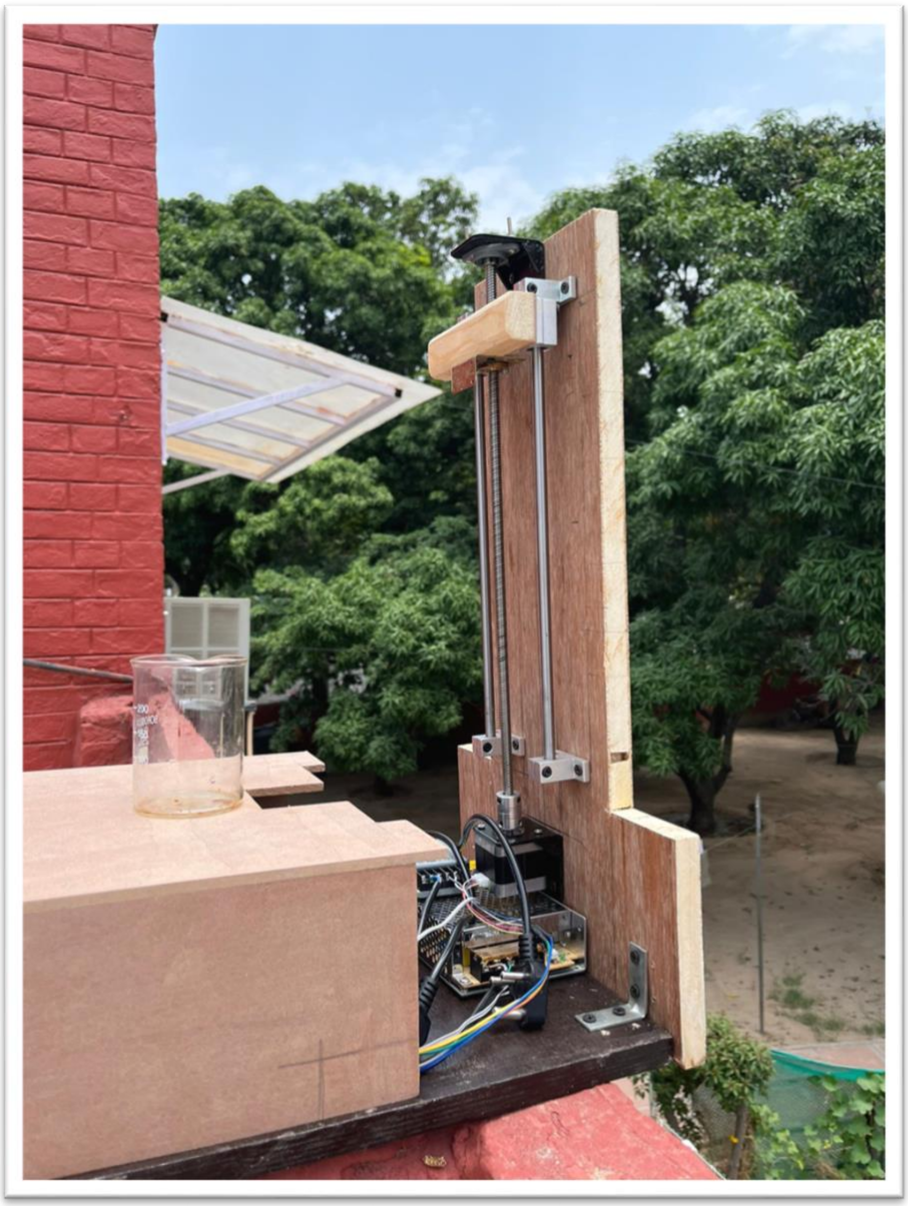


Fig. Overall Structure

Motor Mechanism

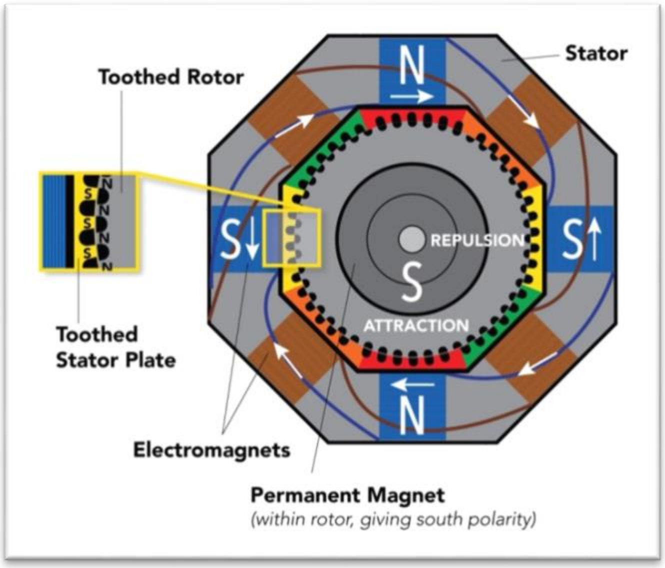


Fig. Stepper Motor Mechanism

The motor that we use here is a stepper motor, specifically a 42mm, 1.8-degree rotation motor. It's called the NEMA 17 stepper motor. A stepper motor, also known as step motor or stepping motor, is a brushless DC electric motor that divides a full rotation into a number of equal steps. The motor's position can be commanded to move and hold at one of these steps without any position sensor for feedback (an open-loop controller), as long as the motor is correctly sized to the application in respect to torque and speed.

Brushed DC motors rotate continuously when DC voltage is applied to their terminals. The stepper motor is known for its property of converting a train of input pulses (typically square waves) into a precisely defined increment in the shaft's rotational position. Each pulse rotates the shaft through a fixed angle.

Stepper motors effectively have multiple "toothed" electromagnets arranged as a stator around a central rotor, a gear-shaped piece of iron. The electromagnets are energized by an external driver circuit or a micro controller. To make the motor shaft turn, first, one electromagnet is given power, which magnetically attracts the gear's teeth. When the gear's teeth are aligned to the first electromagnet, they are slightly offset from the next electromagnet. This means that when the next electromagnet is turned on and the first is turned off, the gear rotates slightly to align with the next one. From there the process is repeated. Each of those rotations is called a "step", with an integer number of steps making a full rotation. In that way, the motor can be turned by a precise angle.

Fig. Stepper Motor

The circular arrangement of electromagnets is divided into groups, each group called a phase, and there is an equal number of electromagnets per group. The number of groups is chosen by the designer of the stepper motor. The electromagnets of each group are interleaved with the electromagnets of other groups to form a uniform pattern of arrangement. For example, if the stepper motor has two groups identified as A or B, and ten electromagnets in total, then the grouping pattern would be ABABABABAB.

Electromagnets within the same group are all energized together. Because of this, stepper motors with more phases typically have more wires (or leads) to control the motor.

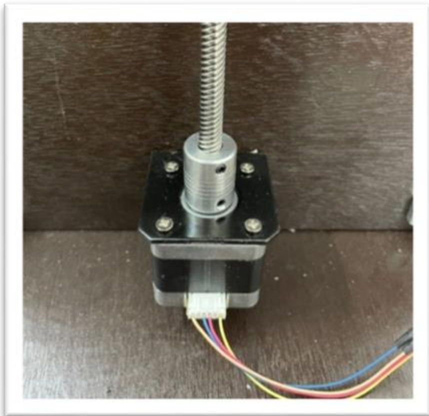


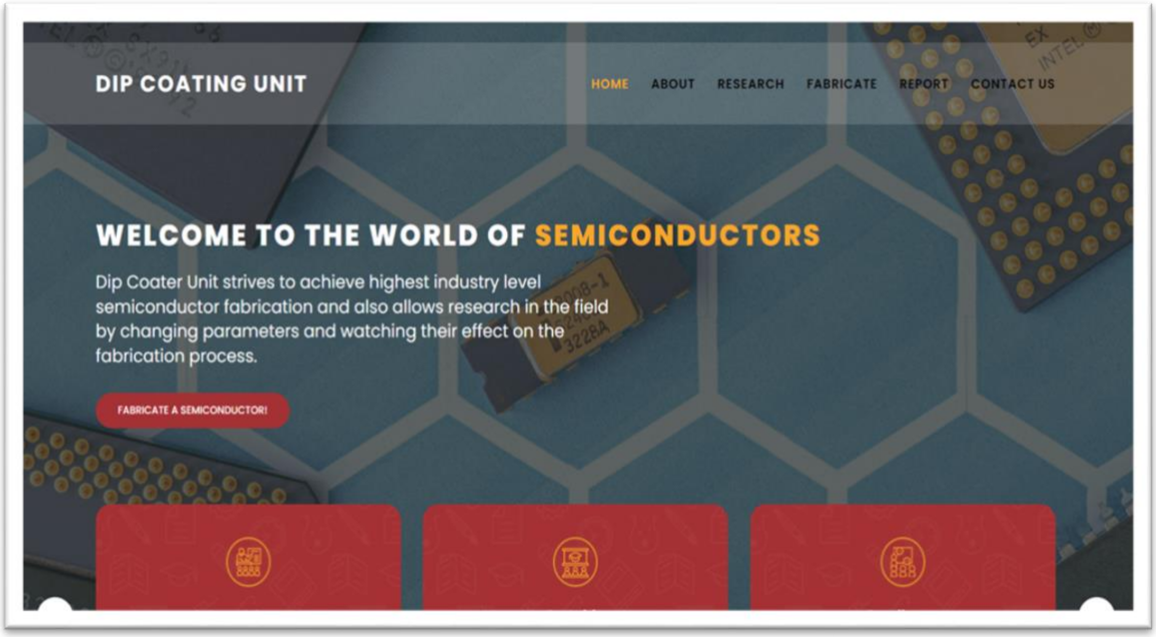
Fig. Stepper Motor attached to the couple

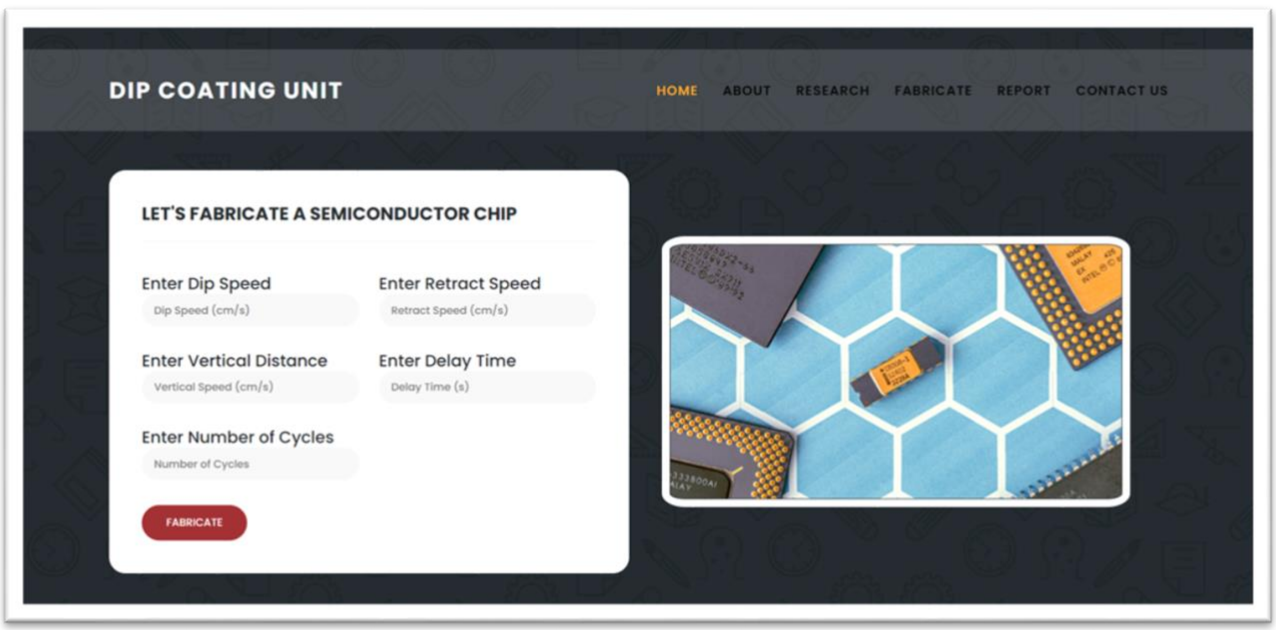
The Rotor/ Shaft System

The rotor/ shaft system is an amalgamation of a 5/8 coupling that is attached with the stepper motor and on the other end with a threaded screw which joins a beveled stud that holds the entire system and supports it. The rotor rotates the coupling which exerts torque on the threaded screw which finally rotates the constrained stud which causes the linear motion of the entire assembly.

The Web APP

- The web application for the project Dip Coater Unit has been made using HTML, CSS and JavaScript using the latest coding practices. A lot of emphasis has been paid to improve the user interface.
- Student can send the variable parameters like dip speed, retract speed, number of cycles and delay time to the Dip Coater Unit over the Wi-Fi without the need to be physically present there. The web application uses PostgreSQL for database and has highly efficient queries to provide best user experience to the people visiting the site.
- Dip Coating Unit app also has a server which is written using JavaScript. We have used Express.js or simply Express as back end Web Application Framework.
- The HTML web page sends an XMLHTTP request to our Express Server. The server receives the request and fetches/ stores data from PostgreSQL database depending on the query.
- Another XMLHTTP request is sent to Arduino which is connected to the internet via ESP8266 WIFI module.
- The Arduino takes this request and performs the required operation.
- The server receives a request from Arduino that the work is done along with the sensors data.
- The parameters entered by user as well as the data returned by Arduino are saved for future reference in our database.


DIP COATING UNIT
HOME ABOUT RESEARCH FABRICATE REPORT CONTACT US
WELCOME TO THE WORLD OF SEMICONDUCTORS
Dip Coater Unit strives to achieve highest industry level semiconductor fabrication and also allows research in the field by changing parameters and watching their effect on the fabrication process.
FABRICATE A SEMICONDUCTOR!


DIP COATING UNIT
HOME ABOUT RESEARCH FABRICATE REPORT CONTACT US
LET'S FABRICATE A SEMICONDUCTOR CHIP
Enter Dip Speed
Dip Speed (cm/s)
Enter Retract Speed
Retract Speed (cm/s)
Enter Vertical Distance
Vertical Speed (cm/s)
Enter Delay Time
Delay Time (s)
Enter Number of Cycles
Number of Cycles
FABRICATE


DIP COATING UNIT
HOME
ABOUT
RESEARCH
FABRICATE
REPORT
CONTACT US
ENTER THE THICKNESS OF CONDUCTED EXPERIMENT
Experiment Serial Number
Serial Number
Observed Thickness
Thickness
SUBMIT
S.no DateTime Dip Speed Retract Speed Distance Cycles Delay Temperature Humidity Vibrations Thicknes
1 2022-05-16 17:16:48.962348+05:30 3000 2000 5 2 5 23 30 -1 45
2 2022-05-16 17:18:00.696128+05:30 3000 3000 10 2 5 23 30 -1 24

CHAPTER 5

PROJECT HIGHLIGHTS

5.1 KEY FEATURES OF OUR PROJECT

- **Smooth Motion**

  With the stepper motor, your substrate will be immersed and withdrawn using smooth and precise movements, ensuring high-quality film coatings.

- **Wide Range of Speeds**

  Our Dip Coater can withdraw a substrate from solution at rates varying from as low as 0.01mm/s to as high as 50mm/s. This gives a wide range of coating thicknesses - all from a single dip-coating system.

- **Variable Withdrawal Speeds**

  The speed of withdrawal can be varied across the substrate length. This enables you to produce thickness gradients across a film for the quick optimisation of film thicknesses.

- **Compact Size**

  The small footprint of the system enables you to perform measurements even in the smallest and busiest labs! With a low bench area, you can be assured that the Dip Coater will fit in your lab.

- **Simple-to-Use Software**

  The in-built software and controls on the Dip Coater have been designed to make it easy for you to programme an experiment. By setting the immersion speed, delay time and withdrawal speed the entire dip coating process can be completed.

- **Full Colour Display**

  In any lighting condition, you can easily read the Dip Coater's coloured screen. The angled screen has been cleverly designed for comfortable viewing in the lab.

- **Quick-Release Clamp**

  Our quick release clamp design allows the user to quickly load and unload samples onto the dip coater arm.

- **Magnetic Rulers**

  We have attached a magnetic ruler beside the clamp for easy measurement while performing the fabrication.

- **Website enabled controls of dip coater**

  User can virtually send the variable parameters and can perform the fabrication process without being present in front of the dip coater.

- **Fabrication history will be stored in the website**

  Dip coating experiment history will be saved in the website for all the fabrication processes that will take place in the dip coater. This provides us with data to conduct research activities.

- **Accident Prevention**

  We have installed two end stop switches on the either side of the clamp and are checking the parameter data entered to prevent hardware damage.

- **Reduced load on Arduino**

  We have added checks to prevent empty requests and unauthorized access for our dip coater unit.

- **Ease of data entry**

  We are adding speeds in mm/s and converting it to steps per second as needed by the motor. Also we have added a scale for user convenience.

- **Variable parameters**

  Vertical distance and delay time can be entered by users apart from the dip speed and retract speed and their effect can be seen on coating thickness.

- **Additional variables to assist research**

  Moisture and Temperature has been added for every experiment in the logs which will help in the research process.

Fig. Final Project

5.2 COMPARISON OF THE PROJECTS

Major Project (7th Semester)	**Major Project (8th Semester)**
	
One Arduino Mega	Arduino Mega + Arduino Uno
Long Length of Bar	Short Length of Bar
Improper User Interface	Well Defined User Interface
Parameter Entry only through keypad	Parameter Entry through keypad and Website
No Temperature, Humidity, Vibration Sensor	Temperature, Humidity and Vibration Sensor
No Data Logging	Data Logging on the website
Stabilization not good	Excellent stabilization due to extremely small tolerances
Displacement and speed calibration not in place	Accurate displacement and speed calibration
Dip Cycles functionality is absent	Dip Cycle functionality is present
No end stop switch provision	End stop switch provision for preventing accidents

CHAPTER 6

RESULTS AND OBSERVATIONS

6.1 RESULTS

The results of this activity were that:

- The conversion of rotational to linear motion happened correctly and aided the smooth vertical movement of the apparatus.
- Thus the basic premise of the exercise was fulfilled. The expensive dip-coater found in the markets has a cheap substitute.
- Minimal vibrations were achieved by using perfect fit material and effective Arduino coding.
- The stepper motor served to be the greatest tool, as without it this exercise would have been futile.
- The guide rails seem to be an essential part of the apparatus, for aiding the smooth operation.
- The Arduino Mega helped us by providing extra ports and prevented a cluttered situation, which happens in the case of an Uno.
- The stud which has constrained rotational motion, attached to the wooden plank through a metal plate is by far the most crucial part of the project. It is perhaps also that point where there is maximal chance of failure.
- End Stop switch has been used to give accidental protection in case there comes some default in the hardware. If there is such a failure, end stop will stop the motor irrespective of the error there will not be any damage to the device or the surroundings.
- Temperature and Humidity sensor give additional information about the surroundings in which coating is done. This will give an additional topic to explore in our research of the dependence of coating thickness on various nearby factors.
- The web app allows us to run the Dip Coater Unit wirelessly over WIFI while maintaining a log of all experiments conducted. Data is the most important tool in any research and this data will drive our conclusions.
- Multiple checks have been added to prevent unauthorized access and empty requests from reaching Arduino.

6.2 OBSERVATIONS

The observations of this exercise are as under:

- The project has a breaking point that is the stud and the metal plate. Care needs to be taken in case using this method.
- This is a tedious yet effective exercise to understand the solution towards a cheap dip coating unit.
- The fabrication history which is being stored in the website can be used to conduct research in the field of semiconductors.
- Use of wood will make this equipment rust proof and also prevent students from any electric shock risk.
- Portable design and hidden wires make this device usable equipment for college laboratories and small fabrication industries.
- Due to involvement of lot of metallic and electronic components, it will take some time to figure out errors and correct them.

EXPLORE INNOVATE EXCEL

CHAPTER 7

SCHEMATICS

EXPLORE INNOVATE EXCEL

TFT DISPLAY to ARDUINO MEGA

ARDUINO MEGA	LCD
13	SD-SCK
12	SD-DO
11	SD-DI
10	SD-SS
23	LCD-DI
22	LCD-DO
29	LCD-D7
28	LCD-D6
27	LCD-D5
26	LCD-D4
25	LCD-D3
24	LCD-D2
A0	LCD-RD
A1	LCD-WR
A2	LCD-RS
A3	LCD-CS
A4	LCD-RST
A5	LCD-CS
3.3V	3.3V
5V	5V
GND	GND

Fig. TFT Display To Arduino Mega

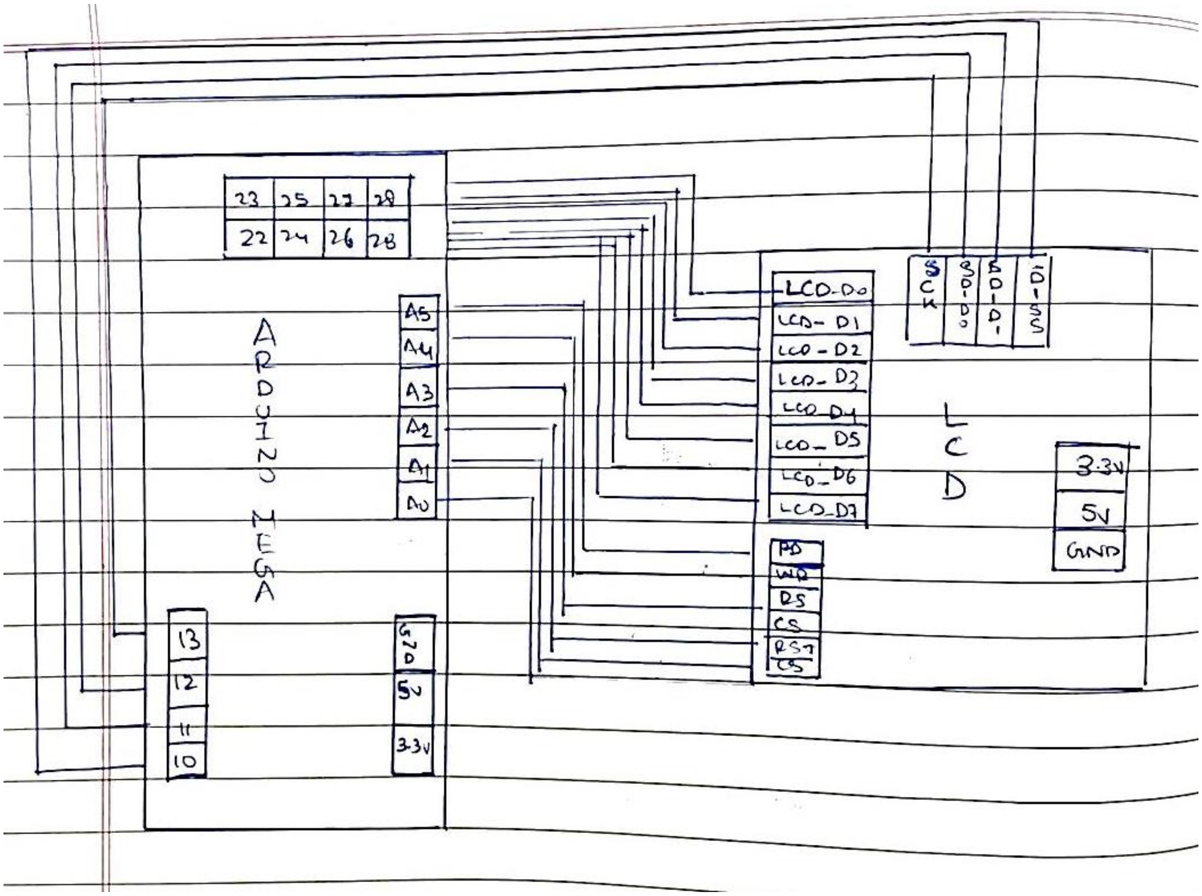


Fig. Schematic of TFT Display to Arduino Mega

MATRIX KEYBOARD with ARDUINO MEGA

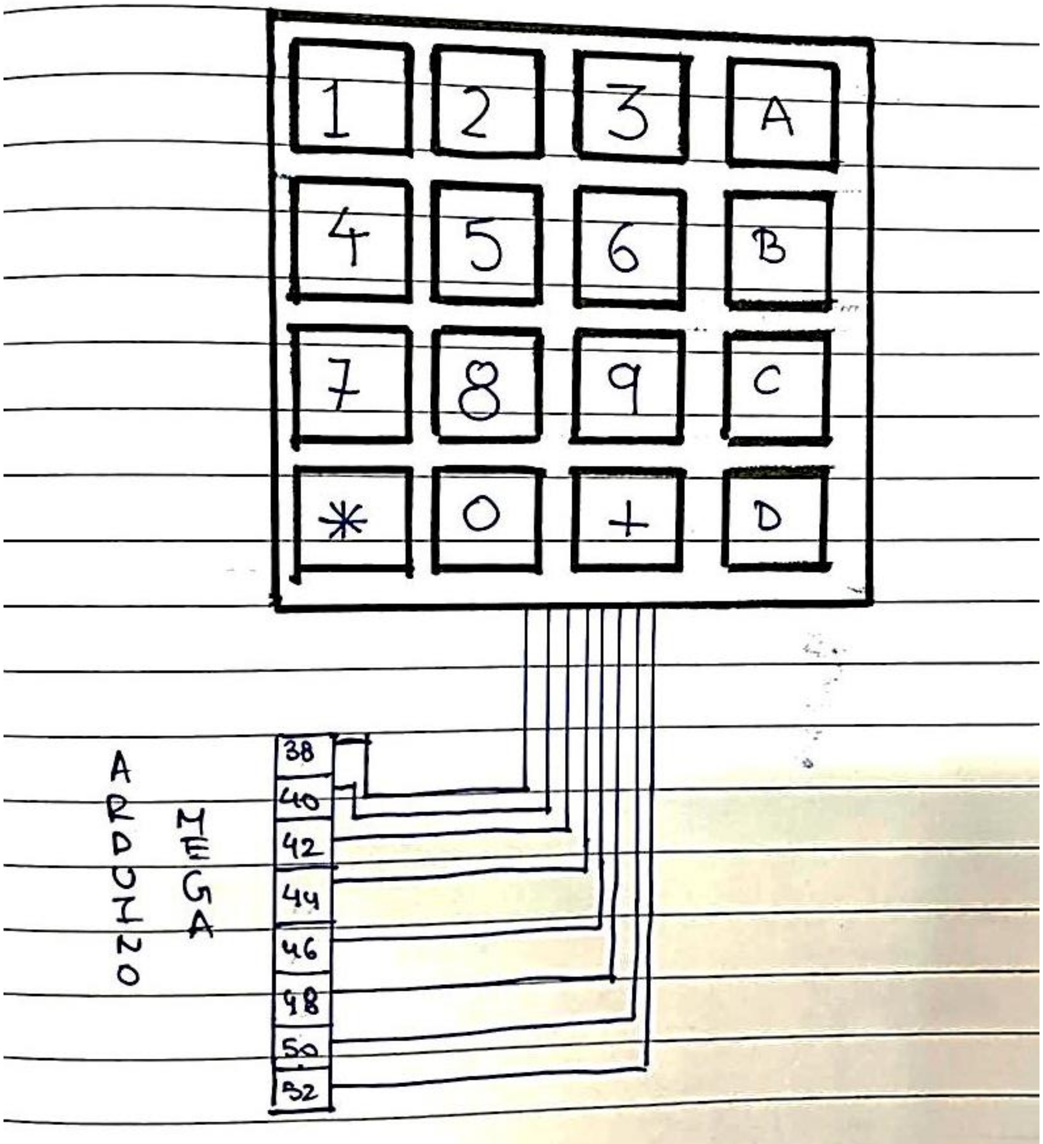


Fig. Schematic of Matrix Keyboard with Arduino Mega

END-STOP SWITCHED WITH ARDUINO MEGA

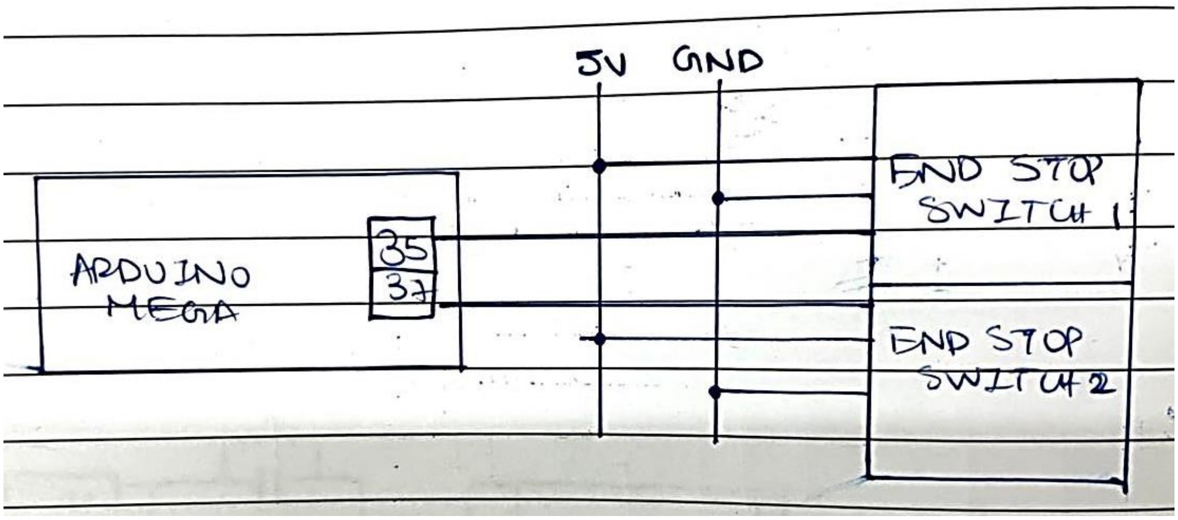


Fig. Schematic of End-Stop Switched with Arduino Mega

EASY DRIVER with ARDUINO MEGA

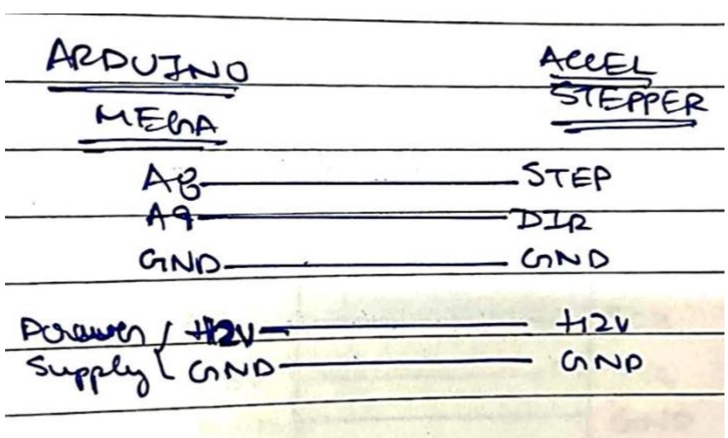


Fig. Connections of Easy Driver with Arduino Mega

ESP 8266 with ARDUINO MEGA

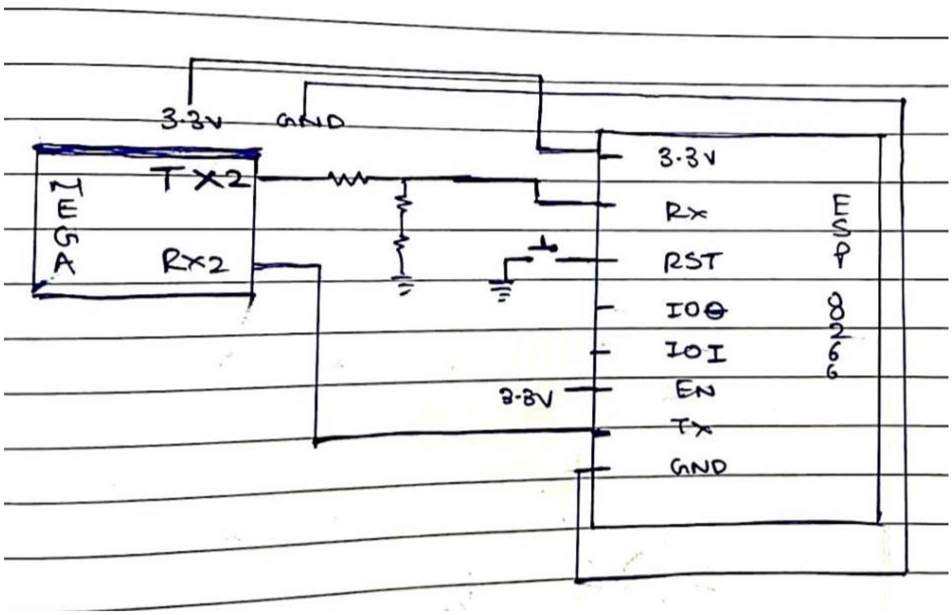


Fig. ESP 8266 with Arduino Mega

TEMPERATURE, HUMIDITY AND VIBRATION SENSOR

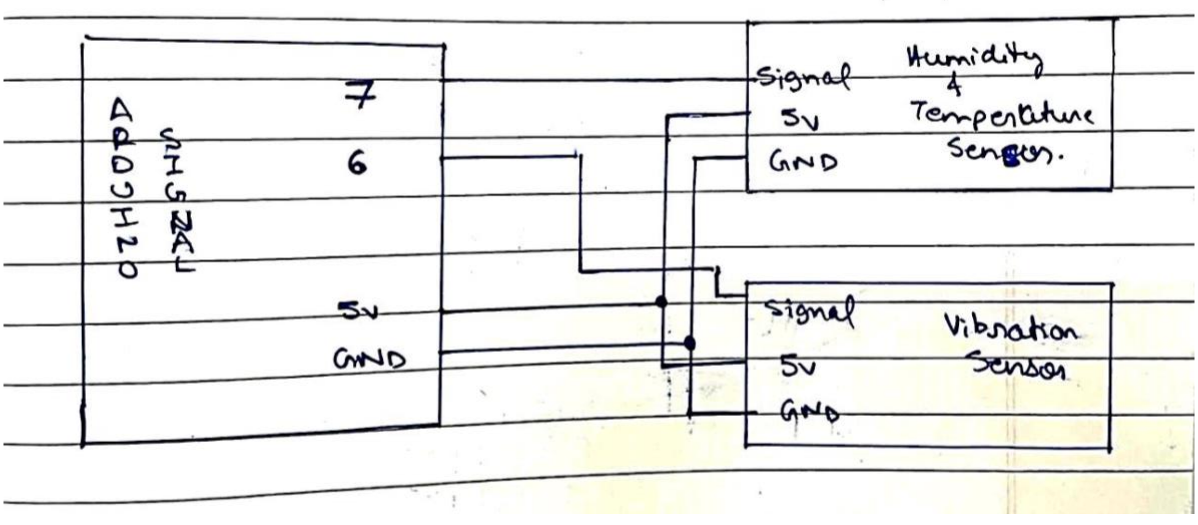


Fig. Connections of Temperature, Humidity and Vibration Sensor

EXPLORE INNOVATE EXCEL

OLED DISPLAY

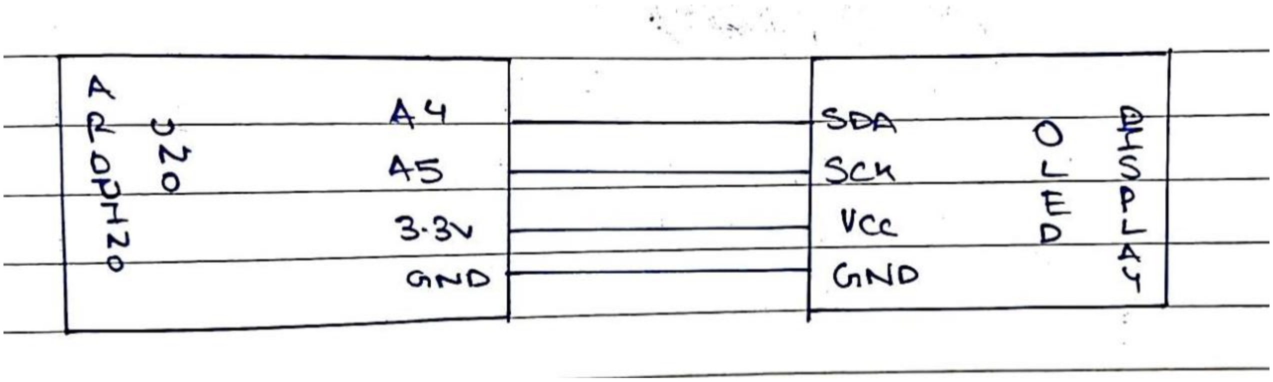


Fig. Connections OLED Display

EXPLORE INNOVATE EXCEL

CHAPTER 8

CODES

8.1 ARDUINO CODE

The code used in the Arduino IDE is as follows:

```
#include <AccelStepper.h> // AccelStepper Library for Stepper
Motor
#include <Keypad.h> // Keypad Library for Matrix Keyboard
#include <U8glib.h> // U8glib for Nokia LCD

// Speed set by the user
unsigned int enteredSpeed;
// Delay entered by the user
unsigned long enteredDelay;
// Setting speed or delay at current moment
bool isSpeed;
// Custom setting
bool isCustom;

// Prompts and units
const char speedStr[] = "Enter speed";
const char speedUnits[] = "RPS";
const char delayStr[] =  "Enter delay";
const char delayUnits[] = "sec";



// Keypad Setup
const byte ROWS = 4;
const byte COLS = 4;

char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

// Arduino pins connected to the row pins of the keypad
byte rowPins[ROWS] = {22, 24, 26, 28};

// Arduino pins connected to the column pins of the keypad
byte colPins[COLS] = {31, 33, 35, 37};

// Keypad object
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS,
COLS);
```

```
// U8glib Setup for Nokia LCD
// Backlight for LCD
#define backlight_pin 11

// Arduino pins connected to Nokia pins
U8GLIB_PCD8544 u8g(3, 4, 6, 5, 7);
// CLK = 3, DIN = 4, SCE = 6, D/C = 5, RST = 7



// Stepper motor setup
AccelStepper stepper(1, A0, A1);
// 1 = Easy Driver interface
// Arduino A0 connected to STEP pin of Easy Driver
// Arduino A1 connected to DIR pin of Easy Driver

void setup(void) {
  // Start serial communication for log
  Serial.begin(9600);
  Serial.println("Starting...\n");

  // Clear the display
  u8g.begin();
  delay(500);

  // Light up the LCD backlight
  // Set the backlight intensity(0 = bright, 255 = dim)
  analogWrite(backlight_pin, 100);

  // Set maximum acceleration in RPS
  stepper.setAcceleration(500);

  // Intialize global variables with initial values
  resetNumbers();
  isCustom = false;

  // Display home page
  homePage();
  delay(5000);
  // drawNokiaScreen(0, 0, speedStr, speedUnits);
}

void loop() {
  char pressedKey;

  if(isCustom == false) {
    menuPage1();
    delay(3000);
```

```
    menuPage2();

    pressedKey = keypad.waitForKey();
    // Logs
    Serial.print("Pressed key is: ");
    Serial.println(pressedKey);

    switch(pressedKey) {
      case 'A':
        enteredSpeed = 100;
        enteredDelay = 5;
        moveStepper();
        break;
      case 'B':
        enteredSpeed = 150;
        enteredDelay = 5;
        moveStepper();
        break;
      case 'C':
        enteredSpeed = 200;
        enteredDelay = 10;
        moveStepper();
        break;
      case 'D':
        isCustom = true;
        break;
    }

    if(isCustom == false) {
      return;
    }

    drawNokiaScreen(0, 0, speedStr, speedUnits);
  }

  // Get value of keypad button if pressed
  // This is blocking
  pressedKey = keypad.getKey();

  // If some key is pressed, check which one
  if(pressedKey != NO_KEY) {
    // Logs
    Serial.print("Pressed key is: ");
    Serial.println(pressedKey);

    // If we enter a digit, make change in speed or delay
    if(isDigit(pressedKey)) {
      checkNumber(pressedKey - '0');
    }
```

```
    // '*' is used for 'backspace'
    // '#' is used for 'enter'
    // If 'enter' is pressed and 'isSpeed' is true, we need to
input the delay from the user
            // Prompt to enter delay
            // else, rotate the motors
        else if(    else if(pressedKey == '*') {
      deleteNumber();
    }
else if(isSpeed == true && pressedKey == '#') {
      if(enteredSpeed != -1) {
        isSpeed = false;
drawNokiaScreen(0, 0, delayStr, delayUnits);
      }
    }
else if(pressedKey == '#') {
      if(enteredDelay != -2) {
        moveStepper();
        isCustom = false;
      }
    }
else if(pressedKey == 'B'){
      resetNumbers();
      drawNokiaScreen(0, 0, speedStr, speedUnits);
    } else {
      Serial.println("Wrong input, so do nothing");
    }
  }
}

void resetNumbers() {
  enteredSpeed = -1;
  enteredDelay = -2;
  isSpeed = true;
}

void checkNumber(int num) {
  // Make change in speed
  if(isSpeed == true) {
    if(enteredSpeed == -1) {
        if(num == 0) {
          return;
        }
        enteredSpeed = num;
    } else {
      enteredSpeed = enteredSpeed * 10 + num;
      // If > 4000, do not make any changes
      if(enteredSpeed > 4000) {
        enteredSpeed = -1;
```

```
        drawNokiaScreen(0, 0, speedStr, speedUnits);
        return;
      }
    }
    drawNokiaScreen(enteredSpeed, 0, speedStr, speedUnits);
  }
  // Make change in delay
  else {
    if(enteredDelay == -2) {
      if(num == 0) {
        return;
      }
      enteredDelay = num;
    } else {
      enteredDelay = enteredDelay * 10 + num;
    }
    drawNokiaScreen(enteredDelay, 0, delayStr, delayUnits);
  }
}

void deleteNumber() {
  if(isSpeed == true) {
    if(enteredSpeed == -1) {
        return;
    }
    enteredSpeed = enteredSpeed / 10;
    drawNokiaScreen(enteredSpeed, 0, speedStr, speedUnits);
    if(enteredSpeed == 0) {
      enteredSpeed = -1;
    }
  } else {
    if(enteredDelay == -2) {
        return;
    }
    enteredDelay = enteredDelay / 10;
    drawNokiaScreen(enteredDelay, 0, delayStr, delayUnits);
    if(enteredDelay == 0) {
      enteredDelay = -2;
    }
  }
}

void moveStepper() {

  // Calculate number of steps needed in mm. Need to edit this
  // int calculatedMove = ((dist * 1600) / 80);

  long calculatedMove = 50000;
```

```
  drawNokiaScreen(enteredSpeed, 0, "Motor running", "RPS");

  stepper.setMaxSpeed(enteredSpeed);

  // Go forward
  stepper.runToNewPosition(calculatedMove);

  // Change position
  drawNokiaScreen(enteredDelay, 50, "Coating", "sec");

  // Delay
  delay(enteredDelay * 1000);

  drawNokiaScreen(enteredSpeed, 50, "Motor running", "RPS");

  // Go back to original position
  stepper.runToNewPosition(0);

  // Reset all conditions
  resetNumbers();

  // Back to speed logic
  drawNokiaScreen(0, 0, "Finished", "");
  delay(2000);
  // drawNokiaScreen(0, 0, speedStr, speedUnits);
}

void displayInit() {
//  homePage();
//  delay(7500);
  menuPage1();
  delay(7500);
//  delay(7500);
//  menuPage2();
}


// Function to display on Nokia LCD
void drawNokiaScreen(int arg, int currPos, const char ask[],
const char units[]) {

  u8g.firstPage();

  do {
    u8g.drawHLine(0, 15, 84);
    u8g.drawVLine(52, 16, 38);
    u8g.drawHLine(0, 35, 84);
    u8g.setFont(u8g_font_profont11);
    u8g.drawStr(0, 10, ask);
```

```
    u8g.setPrintPos(0, 29);
    // Put entered number on Nokia LCD
    u8g.print(String(arg));
    u8g.drawStr(62, 29, units);
    u8g.drawStr(0, 46, "Position");
    u8g.setPrintPos(58, 47);
    // Display current position of stepper
    u8g.print(String(currPos) + "cm");
  } while(u8g.nextPage());
}

void homePage() {
    u8g.firstPage();
    do {
    u8g.setFont(u8g_font_gdr14);
    u8g.drawStr(0, 15, "Dip");
    u8g.drawStr(0, 31, "Coating");
    u8g.drawStr(0, 47, "Unit");
  } while(u8g.nextPage());
}

void menuPage1() {
  u8g.firstPage();
  do {
    u8g.setFont(u8g_font_gdr14);
    u8g.drawStr(0, 13, "Menu");
    u8g.drawHLine(0, 15, 84);
    u8g.setFont(u8g_font_profont11);
    u8g.drawStr(0, 28, "Format:");
    u8g.setFont(u8g_font_u8glib_4);
    u8g.drawStr(0, 40, "Speed(RPS), Delay(sec)");
  } while(u8g.nextPage());
}
void menuPage2() {
    u8g.firstPage();
    do {
      // Set font
      u8g.setFont(u8g_font_profont11);
      u8g.drawStr(0, 10, "A: (100, 5)");
      u8g.drawHLine(0, 12, 84);
      u8g.drawStr(0, 22, "B: (150, 5)");
      u8g.drawHLine(0, 24, 84);
      u8g.drawStr(0, 34, "C: (200, 10)");
      u8g.drawHLine(0, 36, 84);
      u8g.drawStr(0, 46, "D: Custom");
      u8g.drawHLine(0, 48, 84);
    } while(u8g.nextPage());
}
```

8.2 WEB APP CODE

- Server Code

```
const express = require('express');
const pgp = require('pg-promise')(/* options */)

var cors = require('cors')


const app = express();
const port = 8080;
app.use(cors())

const Pool = require('pg').Pool
const pool = new Pool({
    user:'postgres',
    host:'localhost',
    database:'postgres',
    password:'postgres',
    port:'5432'
})

app.get('/',(req,res)=>{
    pool.query('SELECT * FROM dataLogs')
  .then((data) => {
    console.log(data.rows);
    res.json(data.rows);
  })
  .catch((error) => {
    console.log('ERROR:', error)
  })
})

app.post('/thickness/:serialnumber/:thicknessvalue',(req,res)=>{
    var v1 = req.params.serialnumber;
    var v2 = req.params.thicknessvalue;

    pool.query("UPDATE dataLogs SET thickness = "+v2+" WHERE sno
= "+v1)
    .then((data)=>{
        console.log("updated");
    })
    .catch((error)=>{
        console.log(error);
    })
})
```

```
function helper(serialNumberCurrent,speed,delay){
    console.log("hi");
    console.log(serialNumberCurrent);
   pool.query("INSERT INTO dataLogs (sno, speed, delay)
VALUES  ("+ serialNumberCurrent+","+speed+","+delay+ ")")
   .then((data)=>{
       console.log("new experiment added");
   })
}

app.post('/experiment/:speed/:delay',(req,res)=>{
    var speed = req.params.speed;
    var delay= req.params.delay;

    var numberOfExp;
    pool.query("SELECT * FROM dataLogs")
    .then((data)=>{
        console.log(data);
        numberOfExp = data.rowCount;
        helper(numberOfExp+1,speed,delay);
    })


})

app.listen(port,()=>{
    console.log('listening');
})
```

- Front end Code (Fabricate Page)

```
<!DOCTYPE html>
<html lang="en">

  <head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="Template Mo">
    <link
href="https://fonts.googleapis.com/css?family=Poppins:100,20
0,300,400,500,600,700,800,900" rel="stylesheet">

    <title>Dip Coater Unit</title>
```

```
<!-- Bootstrap core CSS -->
<link href="vendor/bootstrap/css/bootstrap.min.css"
rel="stylesheet">

<!-- Additional CSS Files -->
<link rel="stylesheet" href="assets/css/fontawesome.css">
<link rel="stylesheet" href="assets/css/templatemo-edu-
meeting.css">
<link rel="stylesheet" href="assets/css/owl.css">
<link rel="stylesheet" href="assets/css/lightbox.css">
<!--

-->
  </head>

<body>
      <!-- ***** Header Area Start ***** -->
  <header class="header-area header-sticky">
    <div class="container">
        <div class="row">
            <div class="col-12">
                <nav class="main-nav">
                    <!-- ***** Logo Start ***** -->
                    <a href="index.html" class="logo">
                        Dip Coating Unit
                    </a>
                    <!-- ***** Logo End ***** -->
                    <!-- ***** Menu Start ***** -->
                    <ul class="nav">
                      <li><a href="index.html"
class="active">Home</a></li>
                      <li><a href="about.html">About</a></li>
                      <li><a
href="researchData.html">Research</a></li>
                      <li><a
href="fabricate.html">Fabricate</a></li>
                      <li><a>Report</a></li>
                      <li><a href="contactus.html">Contact
Us</a></li>
                  </ul>
                  <a class='menu-trigger'>
                      <span>Menu</span>
                  </a>
                    <!-- ***** Menu End ***** -->
                </nav>
            </div>
        </div>
    </div>
</header>
```

```
<!-- ***** Header Area End ***** -->


    </div>

    <section class="contact-us" id="contact">
     <div class="container mt-5">
       <div class="row">
         <div class="col-lg-6 align-self-center">
           <div class="row">
             <div class="col-lg-12">
               <form id="contact" action="" method="post">
                 <div class="row">
                   <div class="col-lg-12">
                     <h2>Let's fabricate a Semiconductor
chip</h2>
                   </div>
                   <div class="col-lg-6">
                     <h5>Enter Motor Speed</h5>
                     <fieldset>
                       <input name="Motor Speed" type="text"
id="speed" placeholder="Motor Speed" required="">
                     </fieldset>
                   </div>

                   <div class="col-lg-6">
                     <h5>Enter Delay time</h5>
                     <fieldset>
                       <input name="Delay Time" type="text"
id="delay" placeholder="Delay Time" required="">
                     </fieldset>
                   </div>
                   <div class="col-lg-12">
                     <fieldset>
                       <button name="enter" type="submit"
id="form-submit" class="button">Fabricate</button>
                     </fieldset>
                   </div>
                 </div>
               </form>
             </div>
           </div>

         </div>

         <div class="col-lg-6 align-self-center">
             <img class="m-3 projectimage"
src="./assets/images/semiconductor.jpg" alt="not found"/>
         </div>
```

```
        </div>
      </div>

    </section>




    <!-- Scripts -->
    <!-- Bootstrap core JavaScript -->
      <script src="vendor/jquery/jquery.min.js"></script>
      <script
src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

      <script src="assets/js/isotope.min.js"></script>
      <script src="assets/js/owl-carousel.js"></script>
      <script src="assets/js/lightbox.js"></script>
      <script src="assets/js/tabs.js"></script>
      <script src="assets/js/video.js"></script>
      <script src="assets/js/slick-slider.js"></script>
      <script src="assets/js/custom.js"></script>
      <script>
          $.ajaxSetup({timeout:1000});
          // btn =
document.querySelector('input[name="butname"]');
          var data1 = document.getElementById("speed");
          var data2 = document.getElementById("delay");
          var btn = document.getElementById("form-submit");
          // txt = document.querySelector('p');
          // btn.addEventListener('click', led1);
          btn.addEventListener('click', sendData);

          function sendData()
          {

              var spe = data1.value;
              var delay = data2.value;

              const xhr3 = new XMLHttpRequest();

              xhr3.open('post',"http://localhost:8080/experime
nt/"+spe+"/"+delay);
              xhr3.send();
          }
          </script>
      <script>
          //according to loftblog tut
          $('.nav li:first').addClass('active');
```

```
        var showSection = function showSection(section,
isAnimate) {
          var
          direction = section.replace(/#/, ''),
          reqSection = $('.section').filter('[data-
section="' + direction + '"]'),
          reqSectionPos = reqSection.offset().top - 0;

          if (isAnimate) {
            $('body, html').animate({
              scrollTop: reqSectionPos },
            800);
          } else {
            $('body, html').scrollTop(reqSectionPos);
          }

        };

        var checkSection = function checkSection() {
          $('.section').each(function () {
            var
            $this = $(this),
            topEdge = $this.offset().top - 80,
            bottomEdge = topEdge + $this.height(),
            wScroll = $(window).scrollTop();
            if (topEdge < wScroll && bottomEdge > wScroll) {
              var
              currentId = $this.data('section'),
              reqLink = $('a').filter('[href*=\\#' +
currentId + ']');
              reqLink.closest('li').addClass('active').
              siblings().removeClass('active');
            }
          });
        };

        $('.main-menu, .responsive-menu, .scroll-to-
section').on('click', 'a', function (e) {
          e.preventDefault();
          showSection($(this).attr('href'), true);
        });

        $(window).scroll(function () {
          checkSection();
        });
    </script>
</body>
```

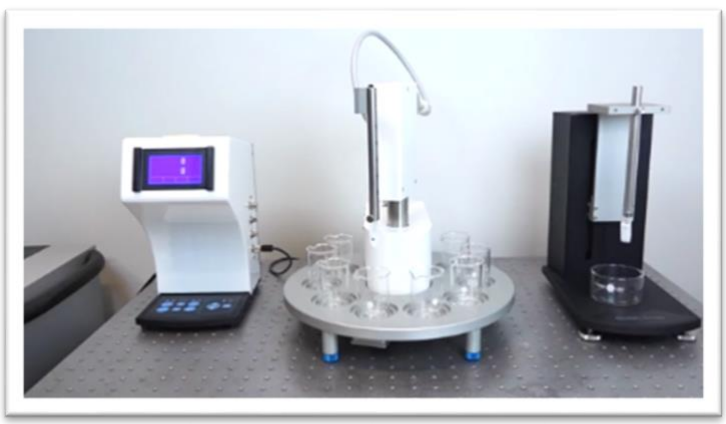
PEC
EXPLORE INNOVATE EXCEL

CHAPTER 9

APPLICATIONS AND LIMITATIONS

9.1 Applications

1. The ability to change the Dip speed, Retract speed, Delay time and number of cycles for the dip coating.
2. Multiple coating capabilities by rotating the arm with a motor.
3. Record of environment variables like temperature and humidity while maintaining experimental data for future references.
4. Dedicated website with the option to run device over the WIFI.
5. Accidental protection of device and environment with End stop switches.
6. Improved user interface with higher resolution LCD, OLED screen and a website.
7. Establishing predefined parameters to reduce turnaround times.
8. Research and training can be conducted in laboratories using this device.
9. The use of new types of coatings on different substrates can be tested.



PEC
EXPLORE INNOVATE EXCEL

9.2 Limitations

As a proof of concept, the model has a number of limitations that we plan to resolve

1. Due to a large metal component count, the project requires periodic maintenance.

2. In its current form, the model cannot be used in the clean room due to the low quality of the materials

3. Although we have reduced vibrations to a bare minimum, but due to involvement of large number of metallic components, there are some vibrations that might affect the quality of coating at higher speeds.

4. The model is too large; it can be made smaller.

5. Motion in the X-axis has not been incorporated due to technical constraints, which can be used for slant dip coating.

6. Due to involvement of many concepts of mechanics, electronics, software, structural design, currently the work is not decentralized completely and any bug if faced might need other components to be changed too.

EXPLORE INNOVATE EXCEL

CHAPTER 10

CONCLUSION AND FUTURE SCOPE

10.1 Conclusion

The project started with the need to make a dip coater unit so that we have a cheap alternative to the expensive options available in the market. This dip coater was aimed to be used in college laboratories or small business and encourage training and research of semiconductors in India. We wanted students and scientists with limited resources to have an option of buying a dip coater unit and apply their theoretical concepts.

We can positively say that after a work of 4 months, we were able to achieve our initial objectives and renew our objectives for future scope of this project.
We were able to make a dip coating device with a user interface and provide the option to vary the thickness of the coating depending on the dip speed, retract speed, delay time and number of cycles. A web app was also made to keep a record of all the experimental data along with the temperature and moisture. One can use the web app to run Dip Coater Unit as well over the WIFI.

Our project can be used in laboratories as we have tried to use top quality metal components of industry standard. This makes the project suitable for production as well.

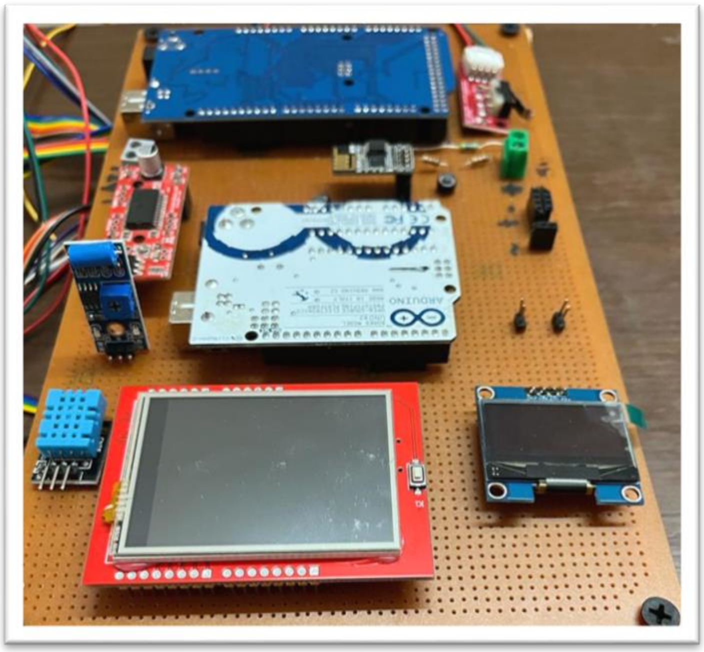


Fig. Hardware Components Soldered on PCB

Fig. Project Prototype

10.2 Future Scope

Given an opportunity to work on the same project in the future we would love to improve this project. Since the initial foundation is laid, we can work on more precise details and new features that we want to append to this project.

- We would like to reduce the usage of wood and eventually work to create a compact device overall. This will add portability to our project and make it suitable for clean room.
- Improved components and a better microprocessor like raspberry Pie can be used.
- Coding can be shifted to Python instead of the current Arduino Uno.
- A web Application with better database and use of latest technologies like React for remote monitoring and usage of the project is something we look forward to, as the need for scaling arises.
- We aim to reduce the vibrations present currently by enabling a vibration absorbing mechanism.
- Increased decentralisation of work so that it is easy to debug and correct an issue if we face one in future.
- Ultimately, this project is for users and we want them to have maximum benefit from it, so we will work on improving the user interface.

REFERENCES

EXPLORE INNOVATE EXCEL

1. “*Handbook of Research: How the dip coating machine works in the Industries and Companies*”, Elgar reference collection (2 ed.), Edward Elgar Publishing, ISBN

2. Pollack, Scott. "*What, Exactly, Is Dip coating?*"

3. YouTube reference link *f*

4. *https://link.springer.com/article/10.1186/s11671-017-2163-0*

5. *https://www.sciencedirect.com/science/article/pii/S0009250900000361*

6. *https://www.sciencedirect.com/science/article/abs/pii/S0167577X11008822*

7. *https://www.ijser.org/paper/Dip-Coating-System-Using.html*

8. *https://www.ijsdr.org/papers/IJSDR2004041.pdf#:~:text=smart%20dip%20coating%20system%20has%20been%20implemented%20successfully,a%20customized%20webpage%20for%20continuous%20and%20effective%20monitoring*

9. *https://tutorials-arduino.com/build-your-own-automatic-arduino-mega/#datetime*

10. *https://www.electronicsforu.com/electronics-projects/prototypes/smart-coating-system*

11. *https://ijecscse.org/papers/specialissue/entc/19.pdf*

12. *https://sci-hub.se/10.3390/app10113835*

13. *https://eee.yasar.edu.tr/wp-content/uploads/2015/06/Mirsat-Y%C4%B1lmaz-Poster.pdf*

14. *https://www.coating.com/blog/coating-control-systems/*

15. *https://www.kdnuggets.com/2019/05/machine-learning-applications-techniques.html*

16. *https://www.static-contents.youth4work.com/y4w/Documents/Portfolio/d5236a21-ff10-47b3-8ad8-3c5deafe45e1.pdf*