

**FREE/LIBRE AND OPEN SOURCE SOFTWARE FOR EDUCATION,
IIT BOMBAY**

Internship-Screening task Report On
PCB Design of a Simplified Arduino Clone using eSim



Submitted by
G.Vijay Venkatesh (2022505026)
III year, B.E., Electronics and Instrumentation Engineering



**MADRAS INSTITUTE OF TECHNOLOGY CAMPUS
ANNA UNIVERSITY
CHROMEPET, CHENNAI – 600044**

ACKNOWLEDGEMENT

I would like to thank the Registrar of Anna University for giving the golden opportunity to do an internship at the esteemed institution.

I express my sincere gratitude to my guide Prof. Kannan Moudgalya and Dr. Sai Subramanian for their guidance and support throughout the project.

I owe my sincere thanks to Mr.Sumanto Kar and Mr.Rajesh for giving us access and guidelines to various components available in eSim.

We are thankful for our university management, for providing us an opportunity to take up this internship; Dr.K.Ravichandran, Dean of MIT campus, Dr.S.Srinivasan, Professor and Head, EIE and Dr. J. Prakash, Professor for being a motivation for taking up new opportunities and projects.

ABSTRACT

This project involves the design and development of a custom Arduino-compatible embedded system based on the ATmega328P microcontroller. The aim is to create a compact and reliable printed circuit board (PCB) tailored for specific applications such as automation and sensor interfacing. The development process began with breadboard testing, where the microcontroller and essential components like the external crystal oscillator, decoupling capacitors, reset circuitry, and USB-to-serial interface were assembled and verified for proper functionality. Successful prototyping led to schematic design in eSim/KiCad, capturing all key elements including voltage regulation, input/output headers, and screw terminals such as the widely available green 5.00 mm pitch Altech AK300 blocks.

With the schematic finalized, appropriate footprints were assigned and the PCB layout was created in KiCad, ensuring optimal placement, routing, and adherence to design rules. Consideration was given to ground planes, trace widths, and silkscreen clarity to enhance manufacturability and debugging. After passing electrical and design rule checks, Gerber and drill files were generated for fabrication. Upon receiving the manufactured PCB, comprehensive testing was carried out. This included visual inspection, continuity testing, microcontroller programming, and verification of all functional blocks including power supply, input/output interfaces, and communication protocols. The board performed successfully under all tests, validating the complete transition from breadboard prototype to a fully functional, tested PCB ready for deployment in embedded applications.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
1	1.1 Intrdocution	6
2	2.1 Circuit Schematics 2.1.1 Core and Power supply 2.1.2 Oscillator and Reset Circuitry 2.1.3 I/O and Connector Interfaces 2.2 Breadbard-Test: Procedure 2.3 Breadboard-Test: Output	7 8 8 9-11
3	3.1 PCB Design using eSim 3.1.1 Schematic (.sch file) 3.1.2 Annotations and ERC 3.1.3 Footprints 3.1.4 Tracing and Auto-Routing 3.1.5 Design Rules Checker 3.2 PCB Printing and Testing 3.2.1 Gerber Files 3.2.2 PCB Printing using Gerber 3.2.3 PCB Components Assembly 3.2.4 PCB Validation and Testing	12 13 14 16 18 19 20 20 22
4	4.1 Links for Downloading Software 4.2 References	23 24

LIST OF FIGURES

Figure 1: Breaboard Test.....	10
Figure 2: Schematic file	13
Figure 3: Footprints.....	16
Figure 4: Free Routing	17
Figure 5: Kicad Tracing	17
Figure 6: Design Rules.....	18
Figure 7: Gerber File	19
Figure 8: Printed Circuit Board.....	20
Figure 9: Soldering and back layer of PCB	21
Figure 10: PCB Front layer	22

CHAPTER-1

1.1 INTRODUCTION:

Microcontrollers form the backbone of modern embedded systems, enabling compact, programmable control in a wide range of applications from industrial automation to consumer electronics. Among them, the ATmega328P stands out as a cost-effective and versatile solution, widely adopted in prototyping platforms such as the Arduino Uno. While development boards like Arduino are ideal for early-stage experimentation, they are not optimized for cost, size, or specific application needs in final products. This limitation drives the need to design custom hardware that replicates the essential functionality of these platforms while tailoring the layout and peripheral connections for targeted use cases.

This project addresses the design and implementation of a custom Arduino-compatible PCB based on the ATmega328P microcontroller, with the objective of transitioning from breadboard-level prototyping to a manufacturable, reliable, and application-specific hardware solution. The design integrates essential components including clock circuitry, reset logic, power management, I/O headers, and screw terminals, enabling a robust interface with external devices such as sensors and actuators. The workflow involves schematic capture, PCB layout, footprint assignment, design rule validation, and Gerber file generation, followed by physical testing of the fabricated board.

Through this process, the project not only demonstrates a complete embedded hardware development cycle but also provides a flexible platform for deploying custom firmware in real-world environments, where reliability, ease of connection, and compact form factor are critical.

CHAPTER-2

2.1 CIRCUIT SCHEMATICS

2.1.1 Core and Power Supply:

At the heart of the system lies the ATmega328P-PU microprocessor, widely known for its ease of integration with Arduino-based development environments. This microcontroller is configured to operate independently as an Arduino Uno/Duemilanove-compatible system.

Power is supplied via an onboard voltage regulation module featuring the 78L05 linear regulator (designated as IC2). This regulator is responsible for stepping down an unregulated input voltage (commonly 9V from a battery or external adapter) to a consistent and stable 5V output, which is essential for the microcontroller's operation. To enhance the performance and stability of the voltage regulator, a $0.33\mu F$ ceramic capacitor (C2) is placed at the input side, while a $0.1\mu F$ capacitor (C3) is positioned at the output. These capacitors suppress noise and filter transient voltage spikes, thereby improving system robustness under variable load conditions.

The ATmega328P receives this regulated 5V through its VCC and AVCC pins, with ground (GND) connections provided for both digital and analog sections. To ensure stable operation and suppress high-frequency noise, two $0.1\mu F$ decoupling capacitors (C1 and C4) are mounted close to the microcontroller's power supply pins. These act as local energy reservoirs, maintaining clean voltage levels during fast switching events within the MCU.

2.1.2 Oscillator and Reset Circuitry:

Accurate timing and reliable operation of the ATmega328P require a stable external clock source. In this implementation, a metallic 16 MHz crystal oscillator is used in conjunction with two 22pF ceramic capacitors. The crystal is connected between the XTAL1 and XTAL2 pins (pins 9 and 10) of the microcontroller. The capacitors are each tied from these pins to ground, forming a standard Pierce oscillator configuration. This setup generates a precise and low-jitter clock signal required for time-critical operations such as UART communication and PWM generation.

The reset circuitry plays a crucial role during programming and error recovery. A momentary push-button switch can be connected between the RESET pin and GND, allowing manual resets. The aforementioned 10k Ω resistor ensures the pin returns to a high state after a reset. During bootloader burning using an Arduino Nano as ISP, a 0.1 μ F capacitor is temporarily placed between the RESET pin and the DTR line from the Nano. This resolves the auto-reset issue commonly encountered during serial uploads, enabling smoother in-system programming experiences.

2.1.3 I/O and Connector Interfaces:

To maximize usability, all digital (D0–D13) and analog (A0–A5) pins of the ATmega328P are broken out through clearly labeled 0.1" pitch male headers. These headers facilitate direct interfacing with sensors, actuators, and other peripherals.

Dedicated headers include:

- PINS: Standard digital and analog I/O breakout.
- SERIAL: TX, RX, VCC, and GND pins for connecting to an FTDI or

USB-to-Serial module for sketch uploads and serial communication.

- ICP: Programming interface (MOSI, MISO, SCK, RESET) for uploading the bootloader or firmware via an Arduino ISP.
- EXTPWR: A keyed 3-pin header for input power, designed to prevent accidental reverse polarity connections.
- BBPWR: Power rails output (+5V and GND) for supplying power to a breadboard or external modules.

These interfaces enable the custom board to support both development and standalone modes with ease.

2.2 Breadboard Test - Procedure:

The prototype validation process was conducted using a step-by-step breadboard test, allowing the isolation and verification of each subsystem.

Initially, the power regulation circuit was constructed. A 9V battery was connected to the input of the 78L05 regulator. Voltage measurements taken at the output confirmed a clean and steady 5V output. To ensure reliability, the regulator was tasked with powering an Arduino Nano (with USB disconnected), and no voltage dips or instability were observed under light to moderate load.

The ATmega328P microcontroller was placed on the breadboard along with the 16 MHz crystal and the two 22pF capacitors. Decoupling capacitors and the pull-up resistor on the RESET pin were also included. An Arduino Nano, loaded with the "ArduinoISP" sketch, was connected to the ATmega328P using the SPI lines and RESET. The wiring followed the standard ISP pinout:

- Nano D10 → ATmega RESET

- Nano D11 → ATmega MOSI
- Nano D12 → ATmega MISO
- Nano D13 → ATmega SCK

Two status LEDs were wired to D9 and D7 of the Nano. The D9 LED provided a “heartbeat” pulse, while the D7 LED indicated bootloader burning status.

Within the Arduino IDE, “Arduino as ISP” was selected under the programmer menu, and the board was set to “Arduino Duemilanove w/ ATmega328P.” The bootloader was successfully burned, with the D7 LED lighting up during the process. An LED was connected to pin 19 (digital pin D13) of the ATmega chip. The “Blink” sketch was uploaded via the “Upload Using Programmer” option. Upon successful upload, the LED began blinking at a 1-second interval.

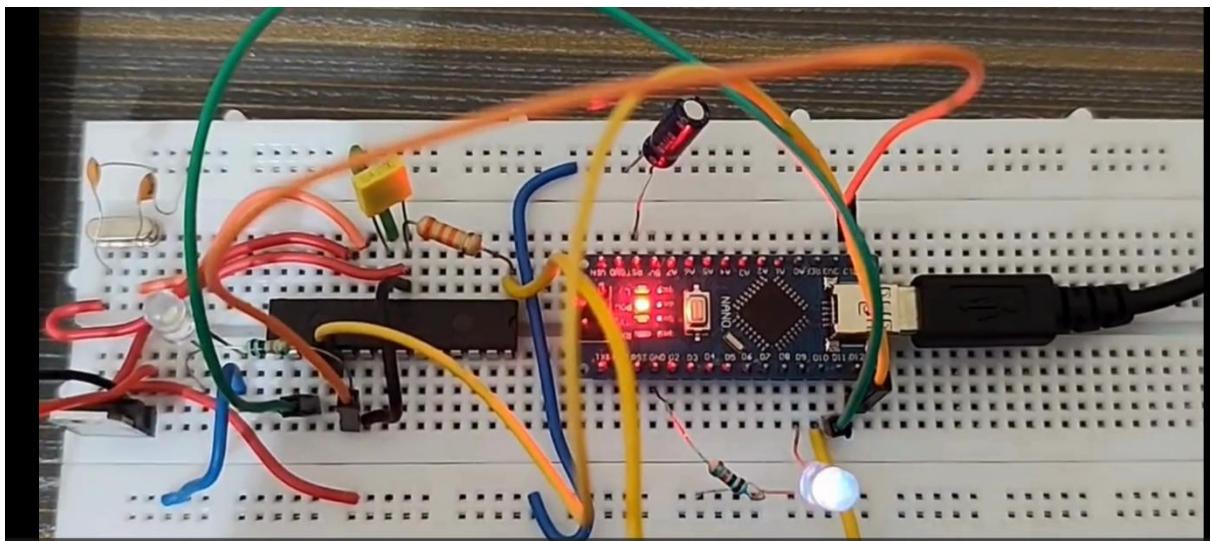


Figure 1: Breadboard Test

2.3 Breadboard Test - Output:

The test results demonstrated the correct functioning of all components and confirmed that the microcontroller system was configured as intended. The power regulator provided a consistent 5V even with brief load changes. The ATmega328P booted up without issue and responded correctly to programming commands sent via the ISP interface.

The D13 LED's blinking pattern validated the proper upload of the sketch to the target chip rather than the Nano. To further confirm, the connection between Nano D13 and the ATmega328P was removed after programming; the LED on pin 19 continued to blink independently, verifying successful program execution.

Additional functional tests were conducted by connecting analog sensors to A0 and A1 and observing serial data through the FTDI interface. These tests validated not only ADC functionality but also UART transmission reliability. Overall, the breadboard prototype served as a reliable foundation for transitioning the design to a fabricated PCB.

CHAPTER-3

3.1 PCB DESIGN USING eSim:

3.1.1 Schematic(.sch) file:

In this stage, the breadboard-tested circuit was formally translated into a digital schematic within eSim. The process started by adding the **ATmega328P-PU** microcontroller symbol along with the **78L05 voltage regulator** for generating a stable 5V supply. Support components such as:

- **0.33μF and 0.1μF capacitors** for regulator stabilization,
- **0.1μF decoupling capacitors** near VCC and AVCC pins,
- **10kΩ pull-up resistor** for the RESET pin, and
- A **16 MHz metallic crystal oscillator** with two **22pF capacitors** to GND,

were all placed with attention to logical grouping and future PCB trace routing.

Key considerations during schematic design:

- **Signal grouping:** Power, I/O, oscillator, and programming signals were clearly sectioned.
- **Net naming:** Nets such as VCC, GND, MOSI, MISO, SCK, TX, and RX were named for easier traceability.
- **Connector planning:** Headers for FTDI, ISP, and breadboard power distribution (BBPWR) were included and labeled.
- **Visual clarity:** Components were aligned, spaced logically, and named consistently for clean circuit comprehension.

This schematic served as the master design for both simulation and physical board layout.

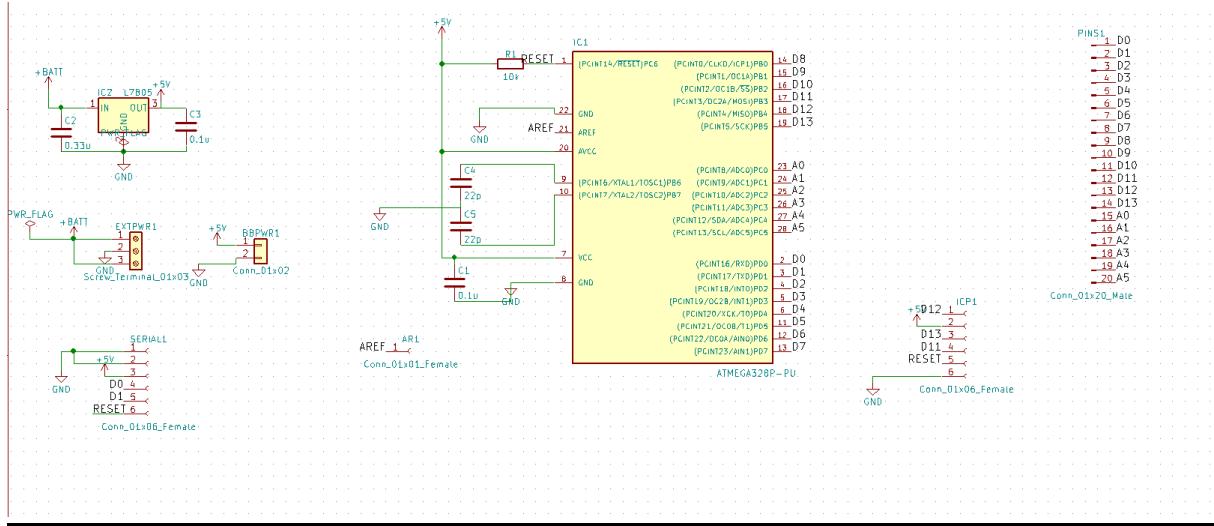


Figure 2: Schematic file

3.1.2 ANNOTATIONS AND ERC:

Annotation is the step where every component is assigned a **unique reference designator** (e.g., R1 for resistor, C1 for capacitor, U1 for IC). This is essential for PCB layout, netlist generation, and for matching components during manufacturing and assembly.

Once all components were annotated, an **Electrical Rules Check (ERC)** was performed to ensure schematic integrity. The ERC automatically identified:

- **Unconnected pins**, which could indicate errors or incomplete wiring,
- **Power pin warnings**, ensuring VCC and GND were properly routed,
- **Floating input pins** on ICs, which might lead to unpredictable behavior.

Warnings such as no-connect flags on unused pins were verified manually. For example, unused analog pins on the ATmega328P were either explicitly marked or tied to ground through a resistor if needed to avoid floating nodes.

3.1.3 FOOTPRINTS:

Footprint assignment is a crucial step in transitioning from schematic design to PCB layout. Each schematic symbol must be linked to a physical representation (footprint) that defines the pad layout on the PCB for soldering.

In this project, careful consideration was given to selecting **through-hole (THT)** components, which are beginner-friendly and suitable for hand soldering. The following footprints were assigned in the eSim environment:

- **ATmega328P-PU(IC1):**

Assigned the Housings_DIP:DIP-28_W7.62mm footprint, which supports standard Dual Inline Package (DIP-28) chips with a 7.62 mm row spacing—ideal for prototyping and socket mounting.

- **7805 Voltage Regulator (IC2):**

Assigned the TO_SOT_Packages_THT:TO-220-3_Vertical footprint, a common vertical package for power regulators, allowing proper heat dissipation and mechanical stability.

- **0.1µF, 0.33µF, 22pF Capacitors (C1–C5):**

Capacitors were mapped to two different footprints:

- ➔ Capacitors_THT:C_Rect_L7.0mm_W6.5mm_P5.00mm for larger ceramic/polyester capacitors (used for C1 and C3),
- ➔ Capacitors_THT:C_Disc_D4.7mm_W2.5mm_P5.00mm for disc-type ceramic capacitors (used for C4 and C5).

- **Crystal Oscillator 16 MHz (OSC1):**

Mapped to Crystals:Crystal_HC49-U_Vertical, a standard vertical package for HC-49 crystal oscillators, providing stable mechanical placement and minimal trace length to MCU pins.

- **10kΩ Resistor (R1):**
AssignedResistors_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P1
0.16mm_Horizontal, matching standard axial resistor dimensions for horizontal placement.
- **Header Connectors (AR1, BBPWR1, ICP1, PINS1, SERIAL1):**
 - ➔ AR1:
Pin_Headers:Pin_Header_Straight_1x01_Pitch2.54mm – single-pin female header.
 - ➔ BBPWR1:
Pin_Headers:Pin_Header_Straight_1x02_Pitch2.54mm – for supplying breadboard power.
 - ➔ ICP1 and SERIAL1:
Pin_Headers:Pin_Header_Straight_1x06_Pitch2.54mm – for FTDI (serial) and ISP (programming) headers.
 - ➔ PINS1:
Pin_Headers:Pin_Header_Straight_1x20_Pitch2.54mm – general-purpose I/O expansion header.\

- **External Power Terminal Block (EXTPWR1):**
Used Connectors_Terminal_Blocks:TerminalBlock_Altech_AK300-2_P5.00mm, a robust 2-pin screw terminal block for external 7–12V DC power input. This ensures firm wire connections for field applications.

Each of these footprints was selected not just for electrical compatibility but also for ease of assembly, signal integrity, and space optimization on the PCB. Through-hole parts facilitate reliable manual soldering and offer mechanical strength for connectors and regulators.

```

1      ARI - Conn_01x01_Female : Pin-Headers:Pin_Header_Straight_1x01_Pitch2.54mm
2  BBPWR1 -           Conn_01x02 : Pin-Headers:Pin_Header_Straight_1x02_Pitch2.54mm
3      C1 -           0.1u : Capacitors_THT:C_Rect_L7.0mm_W6.5mm_P5.00mm
4      C2 -           0.33u : Capacitors_THT:C_Rect_L7.0mm_W6.5mm_P5.00mm
5      C3 -           0.1u : Capacitors_THT:C_Rect_L7.0mm_W6.5mm_P5.00mm
6      C4 -           22p : Capacitors_THT:C_Disc_D4.7mm_W2.5mm_P5.00mm
7      C5 -           22p : Capacitors_THT:C_Disc_D4.7mm_W2.5mm_P5.00mm
8  EXTPWR1 - Screw_Terminal_01x02 : Connectors_Terminal_Blocks:TerminalBlock_Altech_AK300-2_P5.00mm
9      IC1 -          ATMEGA328P-PU : Housings_DIP:DIP-28_W7.62mm
10     IC2 -          L7805 : TO_SOT_Packages_THT:TO-220-3_Vertical
11     ICP1 - Conn_01x06_Female : Pin-Headers:Pin_Header_Straight_1x06_Pitch2.54mm
12     OSC11 - Conn_01x02 : Crystals:Crystal_HC49-U_Vertical
13     PINS1 - Conn_01x20_Male : Pin-Headers:Pin_Header_Straight_1x20_Pitch2.54mm
14     R1 -           10k : Resistors_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
15  SERIAL1 - Conn_01x06_Female : Pin-Headers:Pin_Header_Straight_1x06_Pitch2.54mm

```

Figure 3: Footprints

3.1.4 TRACING AND AUTO ROUTING:

With all footprints placed and the schematic imported into the PCB layout editor, the next task was to route electrical traces. Traces define the copper paths that interconnect pads on the board.

Initially, manual component placement was done to minimize crossovers and optimize space. Important decisions included:

- Placing decoupling capacitors close to IC power pins,
- Minimizing trace length for the crystal oscillator, to reduce noise and timing issues,
- Keeping power and ground traces wider than signal lines to reduce voltage drop.

After initial placement, the auto-router in eSim (based on FreeRouting) was used to generate initial traces. While the auto-router provided a baseline, critical nets (e.g., power lines, clock lines) were manually re-routed to ensure signal integrity and better layout quality.

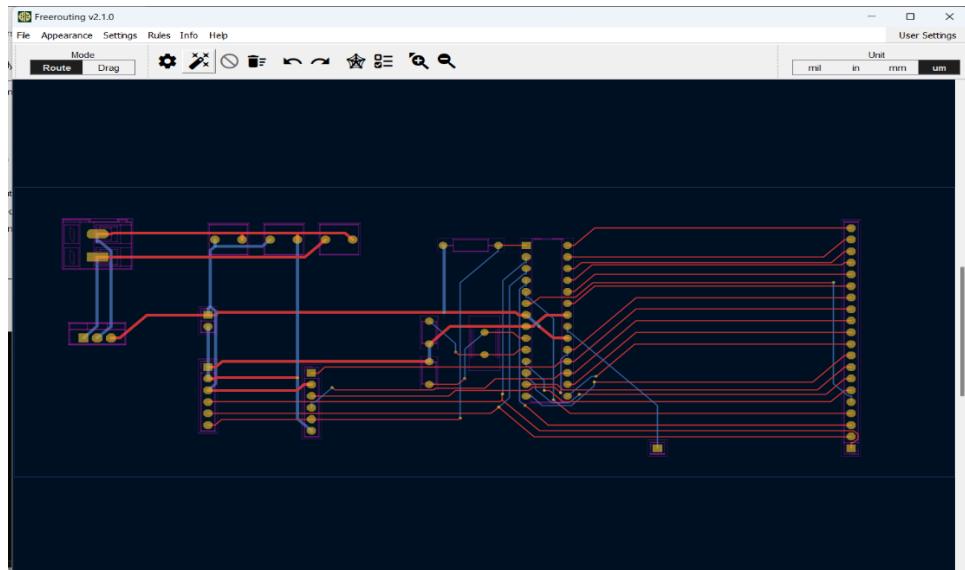


Figure 4: Free Routing

Routing constraints included:

- Minimum trace width of 0.25 mm,
- Trace clearance of 0.25 mm,
- Ground connections routed using a ground fill (GND plane) for noise reduction and better current return path.

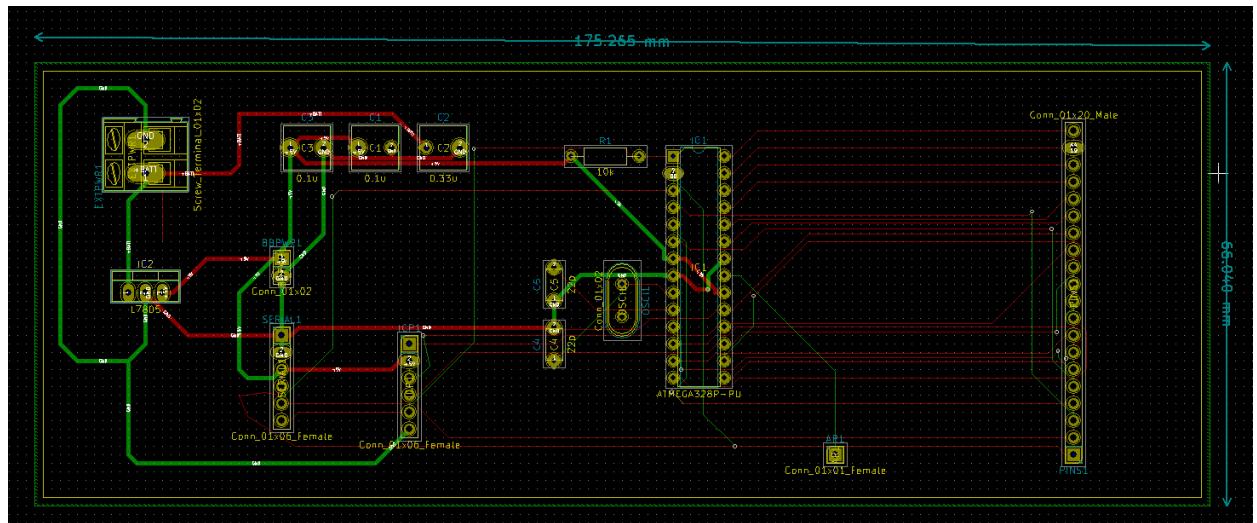


Figure 5: Kicad Tracing

3.1.5 Design Rules Checker:

After routing, the Design Rules Checker (DRC) was run to ensure that the board met standard manufacturing and electrical constraints. DRC validated:

- Minimum trace widths and spacing,
- Clearance between vias, pads, and traces,
- Proper pad connection to all nets,
- No unintended shorts or floating pads.

Any errors were resolved by re-routing the affected traces or adjusting component placement. For example, in one instance, the auto-router placed a trace too close to a pad, which was corrected by manually adjusting the trace path. Additionally, silkscreen overlaps with pads were corrected to avoid printing errors.

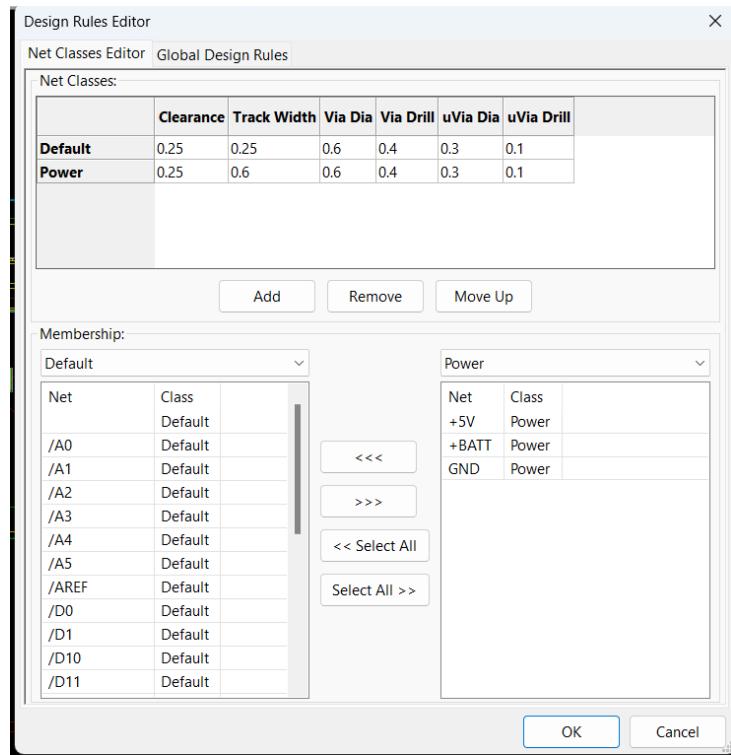


Figure 6: Design Rules

3.2 PCB PRINTING AND ASSEMBLY:

3.2.1 Gerber Files

Once the PCB design passed all checks, the Gerber files were generated. Gerbers are standard files used by PCB manufacturers to fabricate the physical board. Each layer (e.g., top copper, bottom copper, silkscreen, solder mask) is exported as a separate Gerber file.

Using eSim's Gerber generation tool, the following layers were exported:

- Top copper layer (.gtl)
- Bottom copper layer (.gbl) (if applicable)
- Silkscreen layer (.gto/.gbo)
- Solder mask layers (.gts/.gbs)
- Drill file (.drl) specifying via and hole dimensions

All files were then previewed using GerbView or an online Gerber viewer to confirm their correctness. The Gerber files were zipped and prepared for submission to a PCB manufacturer or could be used for in-house etching.

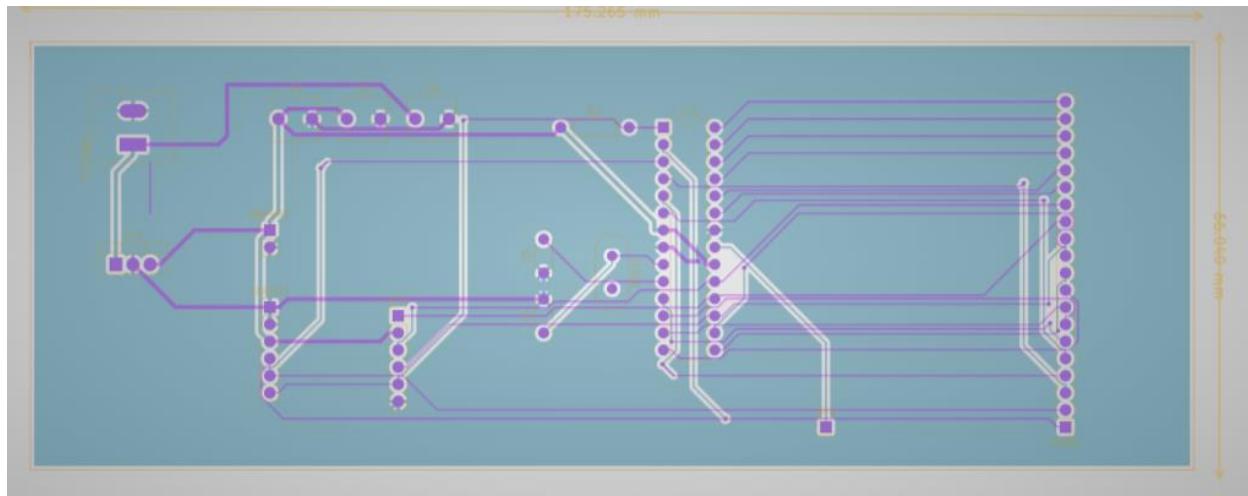


Figure 7: Gerber File

3.2.2 PCB Printing using Gerber:

After the design and routing of the PCB were finalized in eSim, the next crucial step was the generation of Gerber files for fabrication. Gerber files are industry-standard formats that describe the different layers of a PCB, ensuring that manufacturers have precise information to produce the physical board. Each layer of the PCB—copper traces, silkscreen text, solder mask, and drill patterns is exported as an individual file.

Key aspects of Gerber generation and printing:

- Files were generated using the “Export to Gerber” option in eSim.
- Standard extensions included .gbr for layers, .drl for drill files, and .gml or .gm1 for the board outline.
- A Gerber viewer (e.g., Gerbv or online tools) was used to verify the layout before submission.
- All Gerber files were zipped together and submitted to a PCB fabrication service.
- The manufacturer used these files for photoplotting, etching, drilling, and applying a solder mask, resulting in a professionally manufactured PCB.

The accuracy and completeness of these files directly impact the quality of the fabricated PCB, making this step essential in bridging the gap between digital design and a physical product.

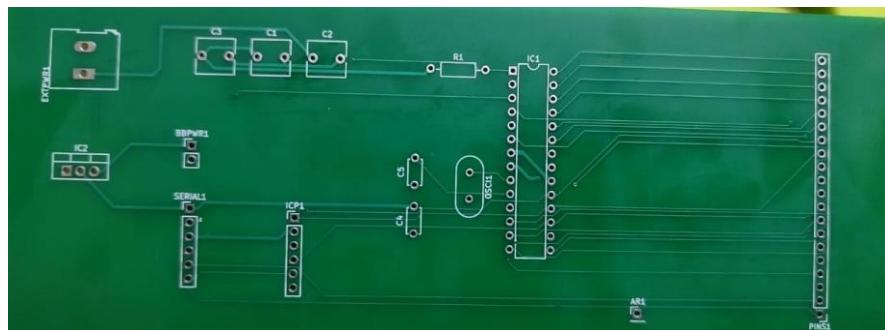


Figure 8: Printed Circuit Board

3.2.3 PCB COMPONENTS ASSEMBLY:

Once the fabricated PCB was received, the next phase was the assembly of components using the Through-Hole Technology (THT) method. THT components have leads that pass through pre-drilled holes in the PCB and are soldered on the opposite side, offering robust mechanical bonding.

Steps and considerations in THT assembly:

- Components like resistors, capacitors, regulators, microcontrollers, and connectors were placed as per the silkscreen markings.
- Leads were slightly bent to keep components in place before soldering.
- Solder joints were made shiny and conical, ensuring strong electrical connections.
- Excess leads were clipped using flush cutters after soldering.
- Special care was taken for sensitive ICs like the ATmega328P by using a 28-pin connector to solder.
- Heat sinks or tweezers were used when soldering components prone to damage.

Through-hole soldering not only ensured reliable electrical contact but also added mechanical strength, making it ideal for prototyping and DIY hardware

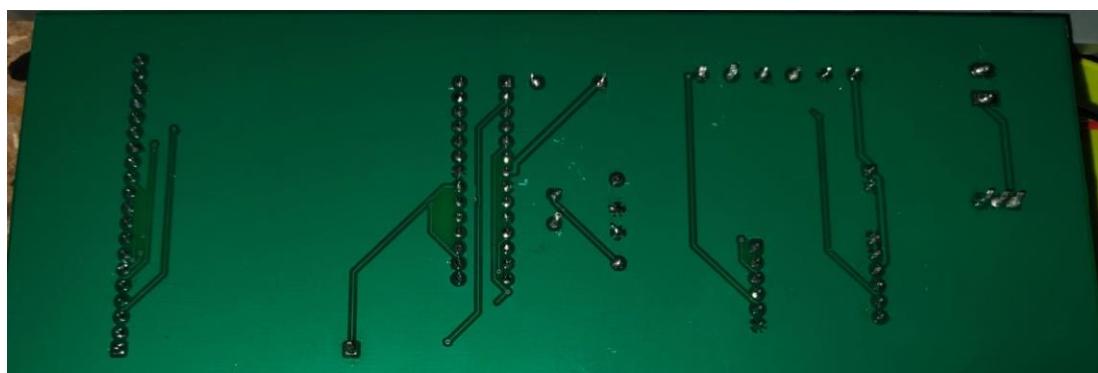


Figure 9: Soldering and back layer of PCB

3.2.4 PCB VALIDATION AND TESTING:

Before deploying the assembled PCB in a functional setup, thorough validation and testing were conducted to ensure the board's reliability and correct operation. This phase is vital for identifying any soldering issues, component misplacements, or design flaws.

PCB testing process involved:

- Visual Inspection: To check for solder bridges, missing components, and polarity issues.
- Continuity Testing: Using a multimeter to ensure all traces were properly connected.
- Short Circuit Check: Especially between VCC and GND lines to prevent component damage.
- Initial Power-On: Performed using a current-limited power supply to safely observe behavior.
- Microcontroller Programming: A basic sketch (like Blink) was uploaded via the serial interface to test the ATmega328P.
- Functional Verification:
 - ➔ Digital pin outputs (e.g., LED blink).
 - ➔ Analog input readings.
 - ➔ Serial communication integrity.

Any abnormalities were corrected by re-soldering joints or replacing defective components. Once all tests passed, the board was deemed ready for deployment in real-world applications.

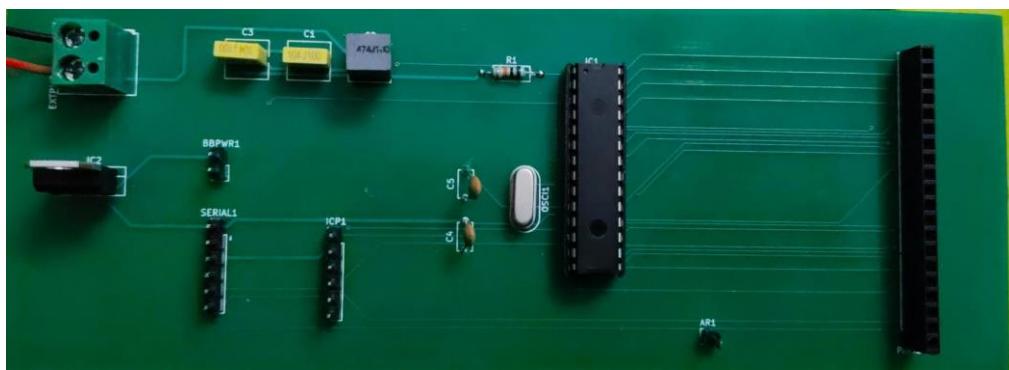


Figure 10: PCB Front layer

CHAPTER 4

4.1 LINKS FOR DOWNLOADING SOFTWARES:

eSIM:

<https://esim.fossee.in/downloads>

User-Manual:

https://static.fossee.in/esim/manuals/eSim_Manual_2.3.pdf

Installation Guide : <https://github.com/FOSSEE/eSim/blob/sky130-dev/INSTALL>

freerouting-2.1.0:

<https://freerouting.org/freerouting/using-with-kicad>

Java-Oracle:

<https://www.oracle.com/java/technologies/downloads/#java8>

4.2 REFERENCES:

1. Spoken Tutorial Videos for eSim : https://spoken-tutorial.org/tutorial-search/?search_foss=eSim&search_language=English
2. https://static.fossee.in/esim/manuals/eSim_Manual_2.3.pdf
3. R. Paknikar, S. Bansode, G. Nandihal, M. P. Desai, K. M. Moudgalya and A. Jha, "eSim: An Open Source EDA Tool for Mixed-Signal and Microcontroller Simulations," 2021 4th International Conference on Circuits, Systems and Simulation (ICCSS), Kuala Lumpur, Malaysia, 2021, pp. 212-217, doi: 10.1109/ICCSS51193.2021.9464198.
Link: <https://ieeexplore.ieee.org/abstract/document/9464198/>
4. <https://freerouting.org/freerouting/manual>