Department of Electronic and Telecommunication Engineering

University of Moratuwa

EN2160 - Electronic Design Realization

Variable Frequency Drive for AC Motors

Design Document

Dabare A.D.T.	210089P
Cooray P.L.L.K.	210088L

This report is submitted as a partial fulfillment of module EN2160

2024.06.23

Contents

1 General

Variable Frequency Drives (VFDs) are electronic devices used to control the speed and torque of AC motors by varying the motor's input frequency and voltage. They are widely used in various industrial and commercial applications to optimize motor performance, enhance energy efficiency, and improve process control. The operation of a VFD can be broadly divided into three main stages: rectification, DC bus, and inversion.

Watch this video to understand how a VFD generally works.

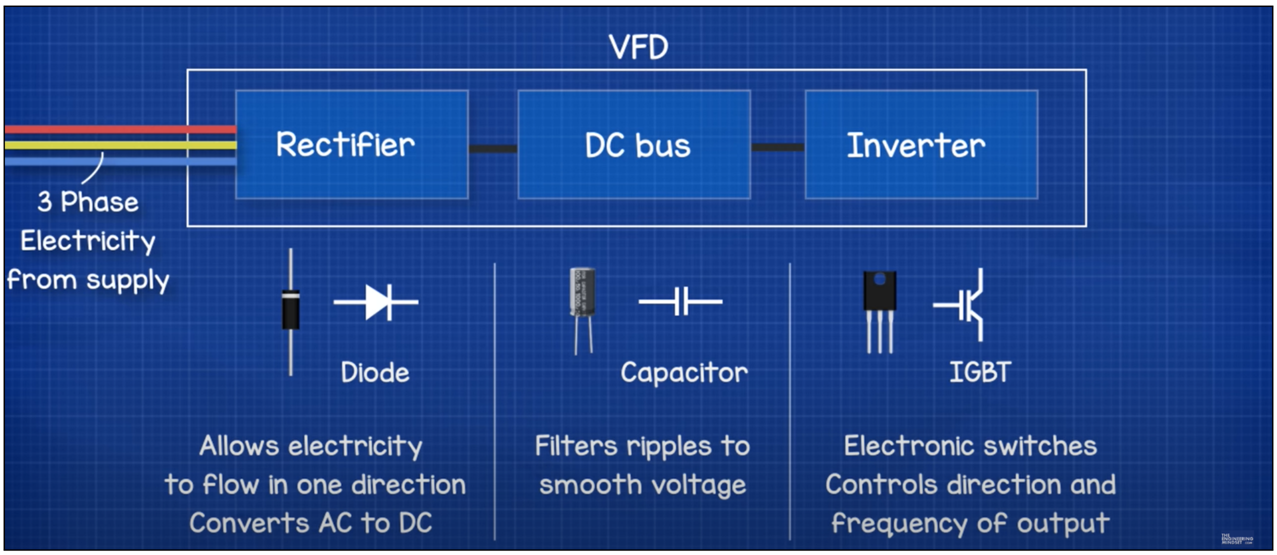


Figure 1: Block Diagram of a Variable Frequency Drive

1. Rectification Stage: In the first stage, the VFD converts the incoming AC power (usually at a fixed voltage and frequency, such as 230V, 50/60 Hz) into DC power. This is accomplished using a rectifier circuit, typically composed of diodes arranged in a full-wave bridge configuration. The rectifier outputs a pulsating DC voltage, which is then smoothed out using capacitors to reduce ripple.

2. DC Bus Stage: The rectified DC voltage is fed into the DC bus, which consists of large capacitors that store the electrical energy and filter out any remaining AC components. The DC bus serves as a stable intermediate link between the rectifier and inverter stages. Additional components, such as bleeding resistors and protection circuits (varistors, fuses), are included to enhance safety and reliability

3. Inversion Stage: In the final stage, the inverter converts the DC power back into AC power with variable frequency and voltage. This stage uses power electronic switches, such as Insulated Gate Bipolar Transistors (IGBTs), to generate Pulse Width Modulated (PWM) signals. By adjusting the duty cycle of these PWM signals, the VFD controls the effective output voltage and frequency supplied to the motor. This allows for precise control over motor speed and torque.

1.1 How VFDs Work

1. Frequency and Voltage Control:
   - The desired motor speed is set through a user interface, such as a potentiometer or a digital control panel.
   - The VFD adjusts the output frequency to control the motor speed. Higher frequencies increase motor speed, while lower frequencies reduce it.
   - To maintain optimal motor performance, the VFD also adjusts the output voltage in proportion to the frequency, following the V/f (Voltage/Frequency) control method.
2. Pulse Width Modulation (PWM):
   - The VFD generates PWM signals by switching the IGBTs on and off at high frequencies. The width of each pulse is varied to control the effective output voltage.
   - The PWM signals are filtered to produce a smooth AC waveform with the desired frequency and voltage.
3. Protection and Feedback:
   - VFDs include various protection features, such as overcurrent, overvoltage, and thermal protection, to safeguard the motor and the drive itself.
   - Feedback mechanisms, such as current sensing and temperature monitoring, are used to adjust the VFD's operation and ensure safe and efficient performance.

1.2 Advantages of VFDs

- Energy Efficiency: By matching motor speed to the actual load requirements, VFDs significantly reduce energy consumption compared to motors running at constant speed.
- Process Control: VFDs offer precise control over motor speed, enhancing the accuracy and efficiency of industrial processes.
- Reduced Mechanical Stress: Soft starting and stopping capabilities reduce mechanical stress on the motor and associated equipment, extending their lifespan.
- Flexibility: VFDs can be used with various types of AC motors and applications, providing versatile solutions for different industries.

2 Electronics Design

The electronics design of a Variable Frequency Drive (VFD) for AC motors is pivotal to its functionality and efficiency. This design encompasses the integration of power electronics, control circuits, and protection mechanisms to regulate the motor's speed and torque. By converting fixed frequency and voltage input into variable output, the VFD enables precise control over the motor's performance. Essential components such as rectifiers, inverters, and pulse width modulation (PWM) techniques are employed to achieve this conversion. The electronics design also incorporates protection against overcurrent, overvoltage, and thermal overload, thereby enhancing the reliability and longevity of both the drive and the motor.

2.1 Power Circuit

The Power Circuit is responsible for handling and converting the high-voltage input to the appropriate levels required by the motor and other electronic components. This includes rectification of AC input to DC, and AC to DC converters. This section ensures the delivery of smooth and consistent power to the motor and other circuits, which is critical for precise motor control.

2.1.1 Rectifier

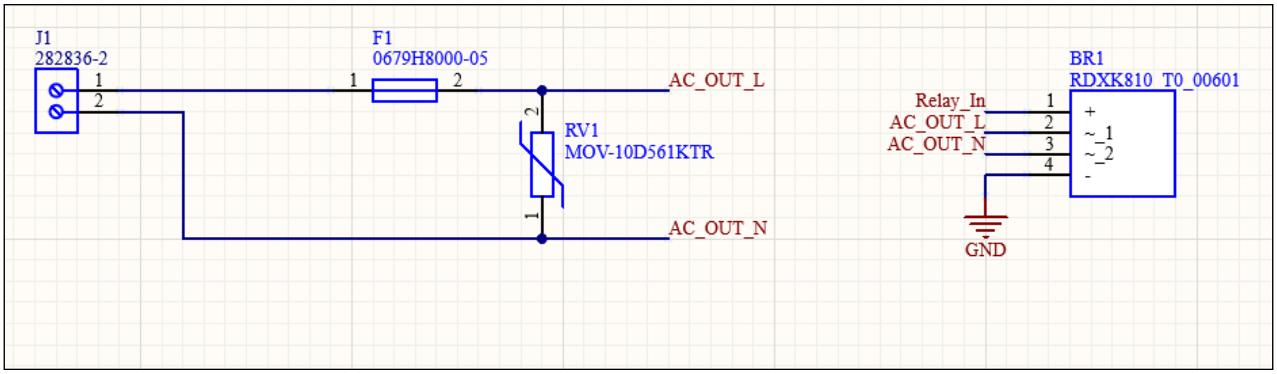


Figure 2: Rectifier

The power circuit of the Variable Frequency Drive (VFD) begins with the conversion of an input voltage of 230V AC to DC. This is achieved using a full-wave rectifier(BR1), which ensures that the entire waveform of the AC input is utilized, providing a more efficient and smoother DC output. For protection, a varistor(RV1) is employed to safeguard the circuit from voltage spikes, which can occur due to transient events or power surges. Additionally, a fuse(F1) is integrated into the circuit to provide overcurrent protection, ensuring that the system is protected from excessive current that could potentially cause damage to the components. This initial conversion and protection stage is crucial for the reliable operation of the VFD, setting the foundation for subsequent stages of the design.

2.1.2 Smoothing Capacitors

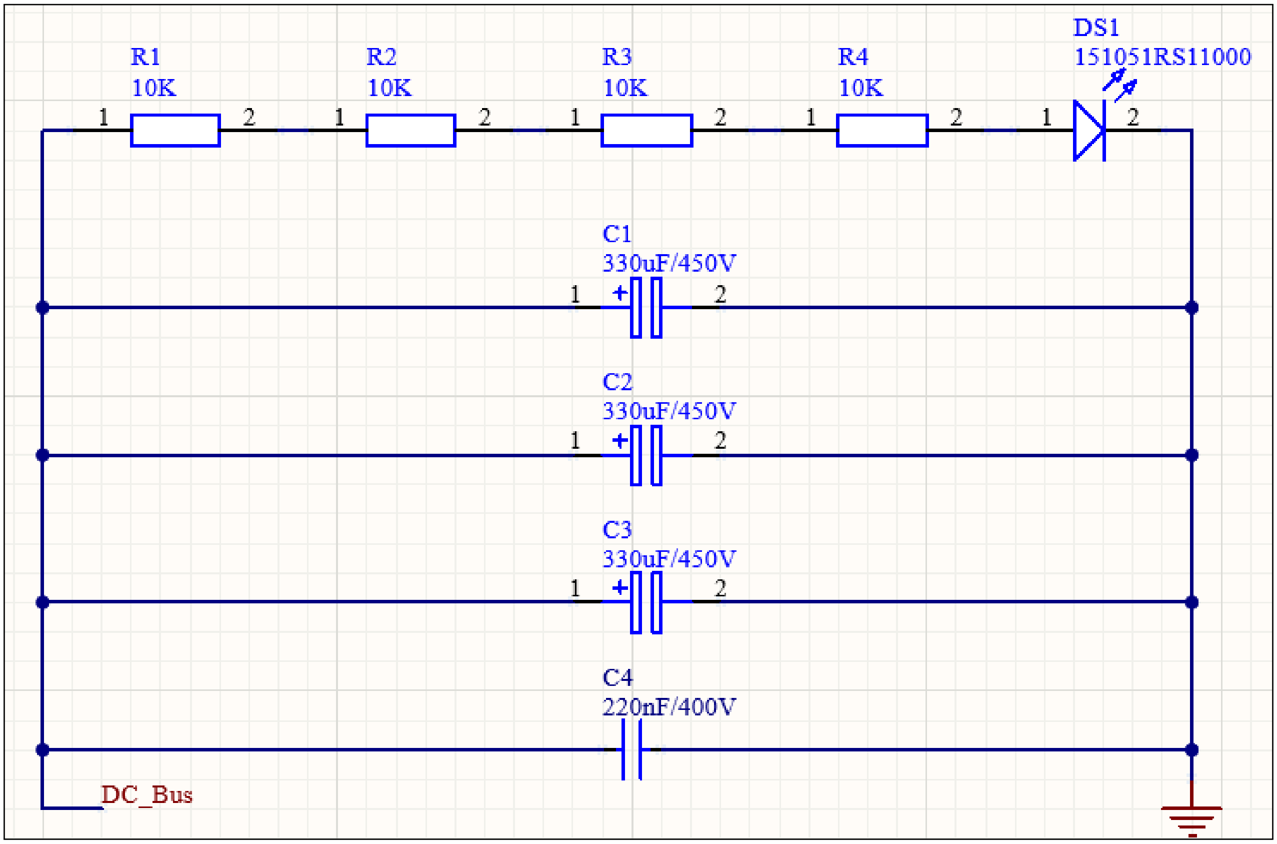


Figure 3: Rectifier

Following the rectification stage, the next component of the power circuit involves the smoothing of the DC output. This is accomplished using three smoothing capacitors(C1-C3), which are essential for reducing the ripple voltage present in the rectified DC signal. The capacitors store and release electrical energy to smooth out fluctuations, resulting in a more stable DC output that is suitable for the subsequent stages of the VFD.

To ensure safety and functionality, four bleeding resistors(R1-R4) are connected which will serve to discharge the capacitors when the power is turned off, preventing any residual charge that could pose a hazard or interfere with the operation of the circuit. An LED indicator is also incorporated into this stage, providing a visual indication that the capacitors are discharging. This LED offers a simple yet effective means of confirming that the discharge process is active, thereby enhancing the safety and monitoring capabilities of the power circuit.

2.1.3 AC-DC Converters

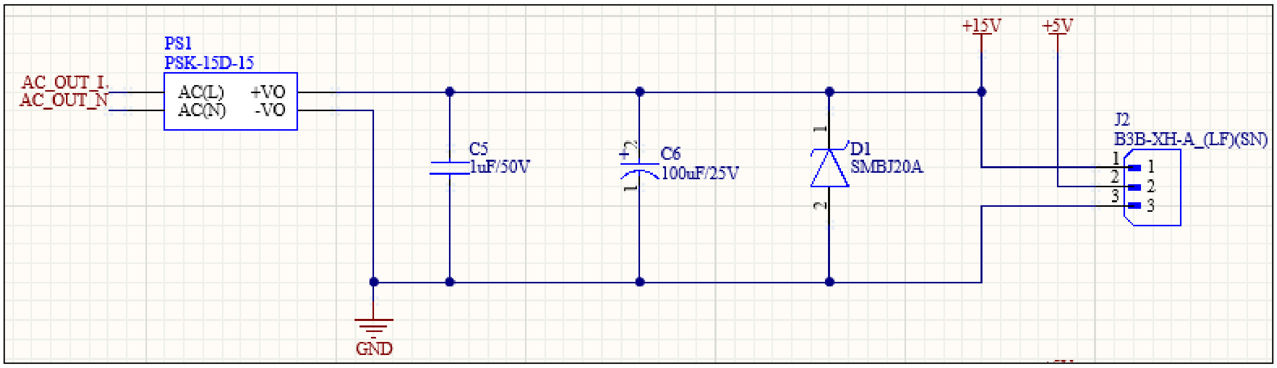


Figure 4: 15V AC-DC Converter

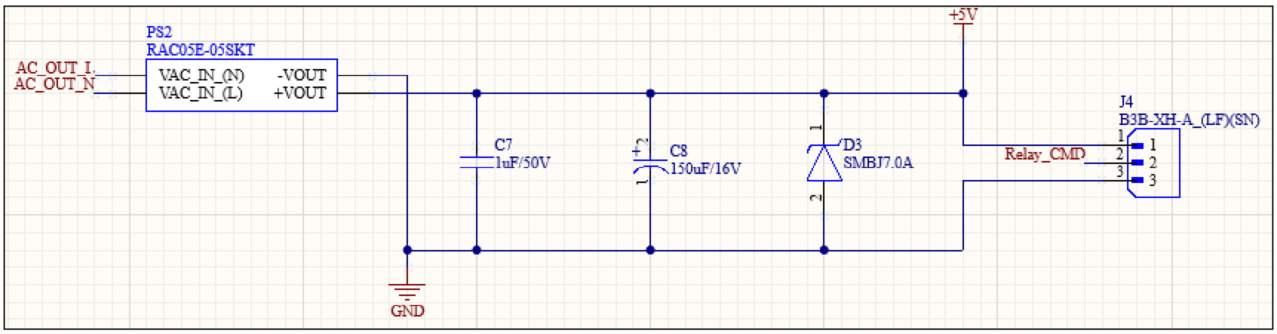


Figure 5: 5V AC-DC Converter

The design incorporates two AC-DC converters to supply the necessary operating voltages for various components within the VFD. Specifically, a 15V converter(PS1) and a 5V converter(PS2) are used to power the microcontroller, Intelligent Power Module (IPM), and the cooling fan. These converters ensure that each component receives the appropriate voltage, which is critical for their optimal performance. To maintain the integrity of the power supply and minimize noise, various filtering components(C5-C8, D1-D2) as recommended by the manufacturers are included in the circuit. These filtering components help in reducing electrical noise and ensuring stable operation of the VFD's sensitive electronic parts.

2.1.4 By Pass Relay Circuit

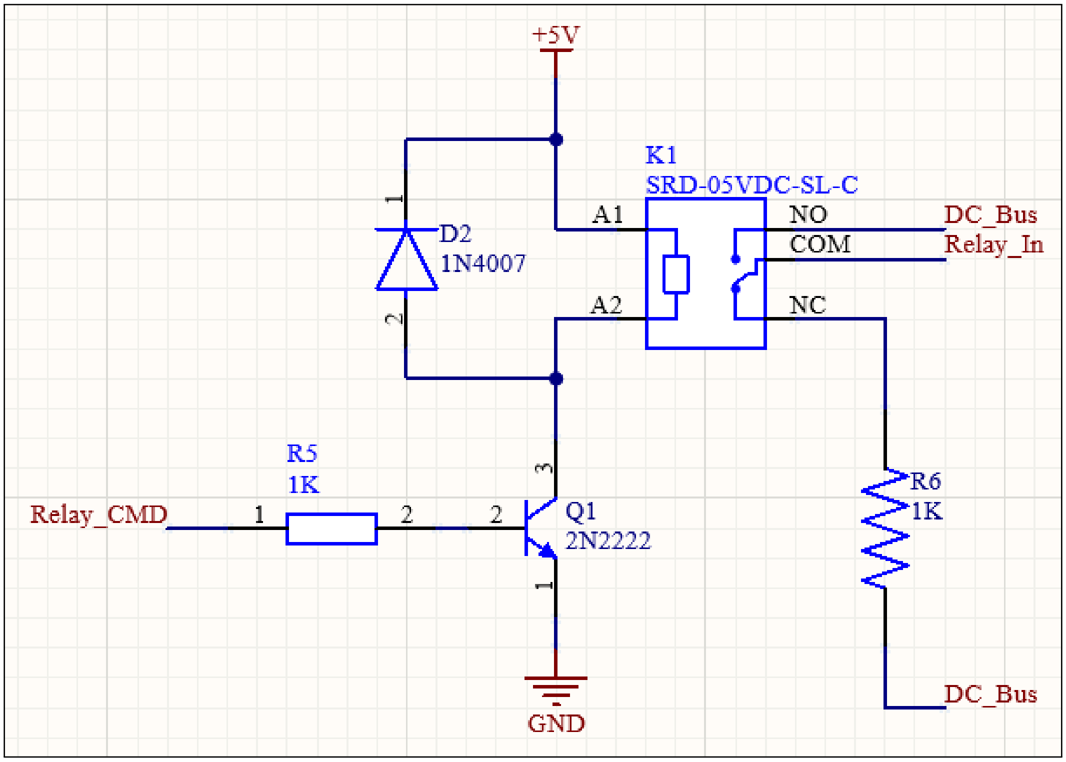


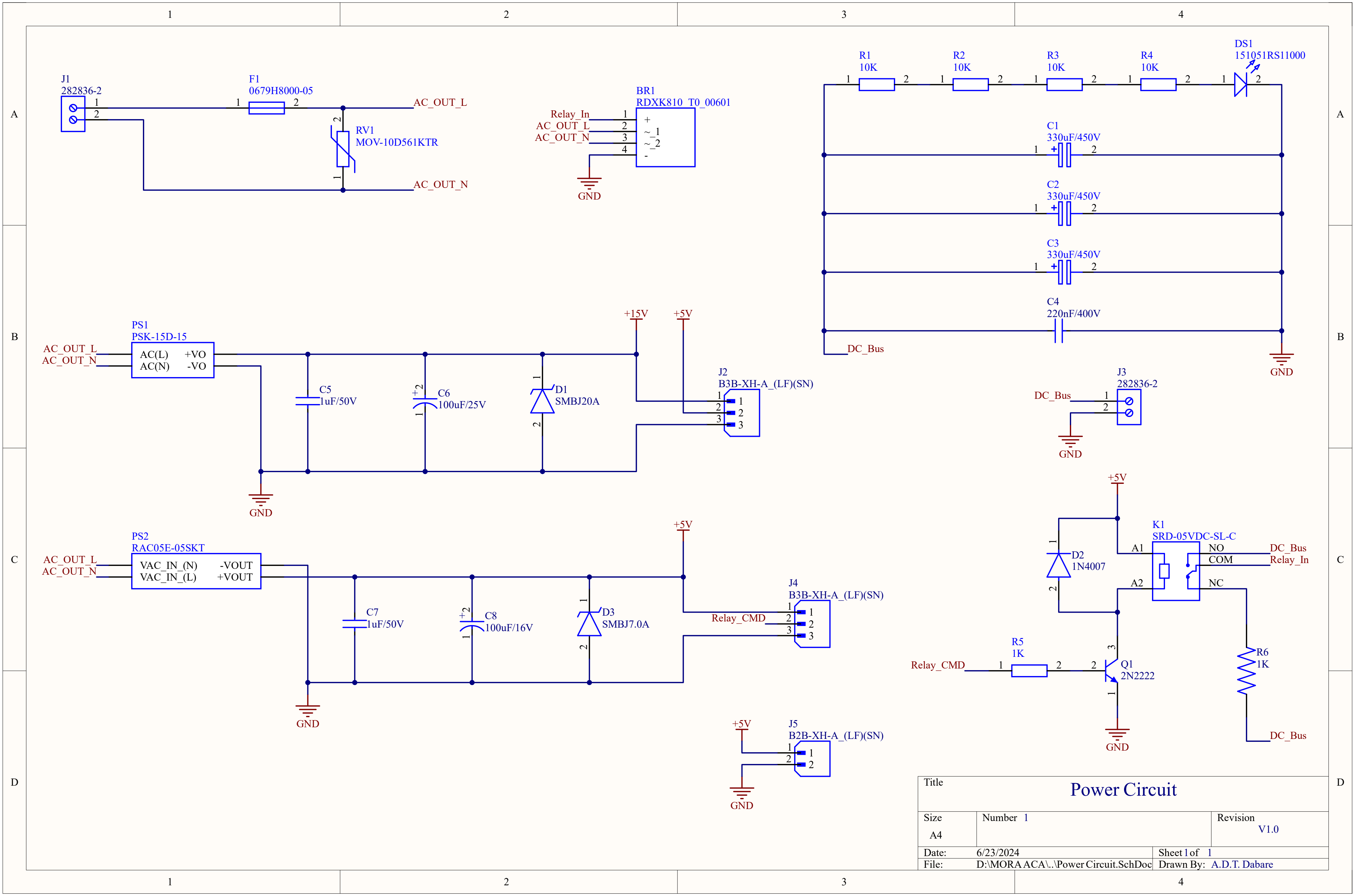
Figure 6: 5V AC-DC Converter

To manage the high inrush current when initially charging the capacitors, an initial charging resistor(R6) is employed. This resistor limits the current, preventing potential damage to the components. After approximately 8 seconds, the initial charging resistor is bypassed by a 5V relay(K1). The relay is activated by a command from the microcontroller, ensuring that the capacitors are fully charged before the resistor is bypassed.

However, the relay's coil requires a significant current of around 150mA, which an Arduino microcontroller cannot directly supply. To address this, an NPN transistor(Q1) is used to amplify the current. When the NPN transistor reaches saturation, it drives the relay effectively. This arrangement allows the low-current output from the microcontroller to control the higher-current requirement of the relay.

To protect the circuit from the voltage spike generated when the relay coil is de-energized, a diode(D2) is placed across the coil. This voltage spike, known as inductive kickback, results from the collapse of the magnetic field in the relay coil and can cause electromagnetic interference (EMI). The diode provides a path for the induced current, thereby preventing potential damage to the sensitive electronic components controlling the circuit. This protective measure ensures the longevity and reliability of the VFD system.

2.1.5 Schematic Design of the Power Circuit


J1
282836-2
F1
0679H8000-05
AC_OUT_L
AC_OUT_N
RV1
MOV-10D561KTR
BR1
RDXK810_T0_00601
Relay_In
AC_OUT_L
AC_OUT_N
GND
R1
10K
R2
10K
R3
10K
R4
10K
DS1
151051RS11000
C1
330uF/450V
C2
330uF/450V
C3
330uF/450V
C4
220nF/400V
DC_Bus
GND
PS1
PSK-15D-15
AC(L) +VO
AC(N) -VO
AC_OUT_L
AC_OUT_N
+15V
+5V
C5
1uF/50V
C6
100uF/25V
D1
SMBJ20A
J2
B3B-XH-A_(LF)(SN)
GND
J3
282836-2
DC_Bus
GND
PS2
RAC05E-05SKT
VAC_IN_(N) -VOUT
VAC_IN_(L) +VOUT
AC_OUT_L
AC_OUT_N
+5V
C7
1uF/50V
C8
100uF/16V
D3
SMBJ7.0A
J4
B3B-XH-A_(LF)(SN)
Relay_CMD
GND
+5V
J5
B2B-XH-A_(LF)(SN)
GND
+5V
D2
1N4007
K1
SRD-05VDC-SL-C
A1
A2
NO
COM
NC
DC_Bus
Relay_In
R5
1K
Relay_CMD
Q1
2N2222
GND
R6
1K
DC_Bus
Title
Power Circuit
Size
A4
Number 1
Revision
V1.0
Date: 6/23/2024
File: D:\MORA ACA\..\Power Circuit.SchDoc
Sheet 1 of 1
Drawn By: A.D.T. Dabare

2.1.6 PCB Layout of the Power Circuit

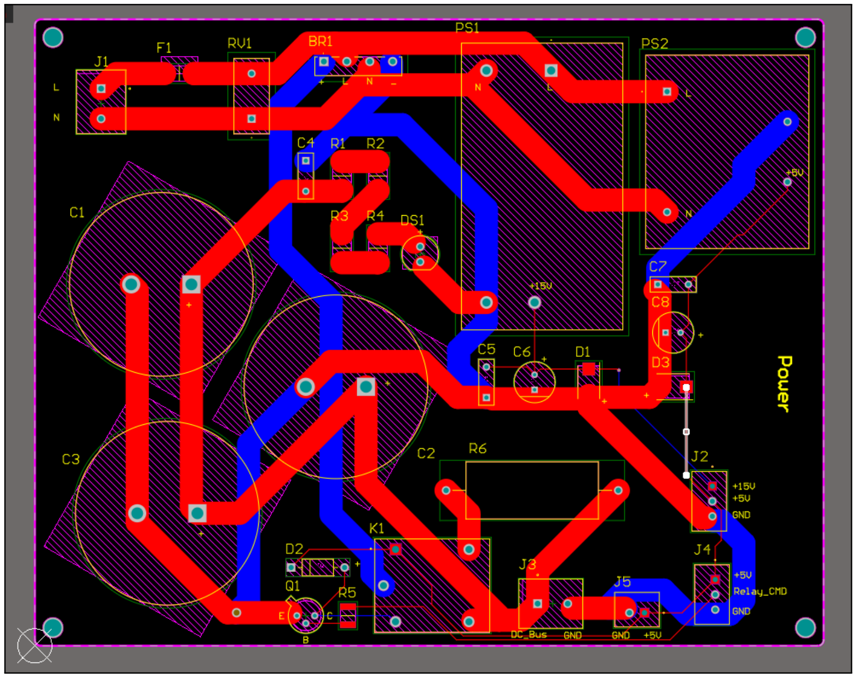


Figure 7: Power Circuit Bottom Layer PCB 2D Layout

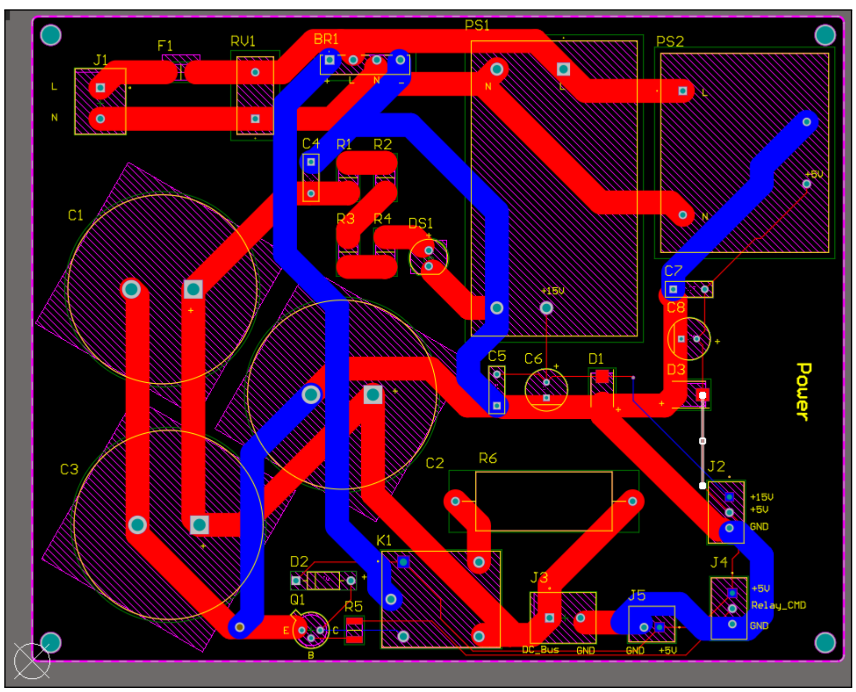


Figure 8: Power Circuit Bottom Layer PCB 2D Layout

- Dimensions - 131mm x 104mm
- Outer Copper Weight - 2 Oz
- 10A Trace Width - 4mm

- 1A Trace Width - 0.3mm

Inputs:

Current	10	Amps
Thickness	2	oz/ft^2

Optional Inputs:

Temperature Rise	10	Deg C
Ambient Temperature	25	Deg C
Trace Length	100	mm

Results for Internal Layers:

Required Trace Width	9.36	mm
Resistance	0.00270	Ohms
Voltage Drop	0.0270	Volts
Power Loss	0.270	Watts

Results for External Layers in Air:

Required Trace Width	3.60	mm
Resistance	0.00701	Ohms
Voltage Drop	0.0701	Volts
Power Loss	0.701	Watts

Figure 9: 10A Trace Width Calculation

Inputs:

Current	1	Amps
Thickness	2	oz/ft^2

Optional Inputs:

Temperature Rise	10	Deg C
Ambient Temperature	25	Deg C
Trace Length	100	mm

Results for Internal Layers:

Required Trace Width	0.391	mm
Resistance	0.0646	Ohms
Voltage Drop	0.0646	Volts
Power Loss	0.0646	Watts

Results for External Layers in Air:

Required Trace Width	0.150	mm
Resistance	0.168	Ohms
Voltage Drop	0.168	Volts
Power Loss	0.168	Watts

Figure 10: 1A Trace Width Calculation

Please visit this Website for trace width calculation.

2.1.7 Component List of the Power Circuit

Item	Quantity	Reference	Part
1	2	J1, J3	5.08mm Pitch 2-Pin 2-Way Screw Terminal Block PCB Mount
2	1	F1	Surface Mount Fuses 8A 350 VAC 72 VCD
3	1	RV1	Varistors 180pF 560volts 10%
4	1	BR1	Bridge Rectifier (8A/1000V,Trr:250ns)
5	3	C1-C3	Aluminum Electrolytic Capacitors - Snap In 450VDC 330uF
6	1	C4	0.22uf 400V- 224 Mylar Capacitor
7	4	R1-R4	Thick Film Resistors - SMD ResPowerQ 2512 10k 5% 2W
8	1	DS1	5 mm LED
9	1	PS1	AC/DC Power Modules 15 Vdc, 1 A, 15 W
10	1	PS2	AC/DC Power Modules 5W 90-264Vin 05Vout 1A
11	1	C5, C7	Ceramic capacitor Through Hole 1uF 50V
12	1	C6	Aluminum Electrolytic Capacitor Through Hole 100uF 25V
13	1	C8	Aluminum Electrolytic Capacitor Through Hole 100uF 16V
14	1	D1	ESD Suppressors / TVS Diodes 20volts 5uA 18.5 Amps Uni-Dir
15	1	D3	ESD Suppressors / TVS Diodes 20volts 5uA 18.5 Amps Uni-Dir
16	1	D2	Through Hole 1N4007 Diode
17	1	Q1	Through Hole 2N2222A NPN Transistor TO-92
18	1	R5	Thick Film Resistors - SMD 1210 1K Ohm Anti- Pulse AEC-Q200 10%
19	1	R6	Wirewound Resistors - Through Hole 5W 1K Ohm 5%
20	1	K1	5VDC Miniature Relay 10A 250VAC / 10A 30VDC
21	2	J2, J4	JST wire connectors 2.54mm with socket - 3 pin

2.2 IPM Circuit

The Intelligent Power Module (IPM) Circuit is the heart of the VFD, integrating the power devices with the necessary drive and protection circuitry. The IPM combines high-speed switching capabilities with protection features such as overcurrent, short circuit, and thermal shutdown. This circuit is crucial for maintaining the safe and efficient operation of the power components, enhancing the overall performance and reliability of the VFD. It also interfaces with the MCU to receive control signals and provide feedback on the system's status. We have utilized the STGIPQ8C60T-HZ as our Intelligent Power Module (IPM).

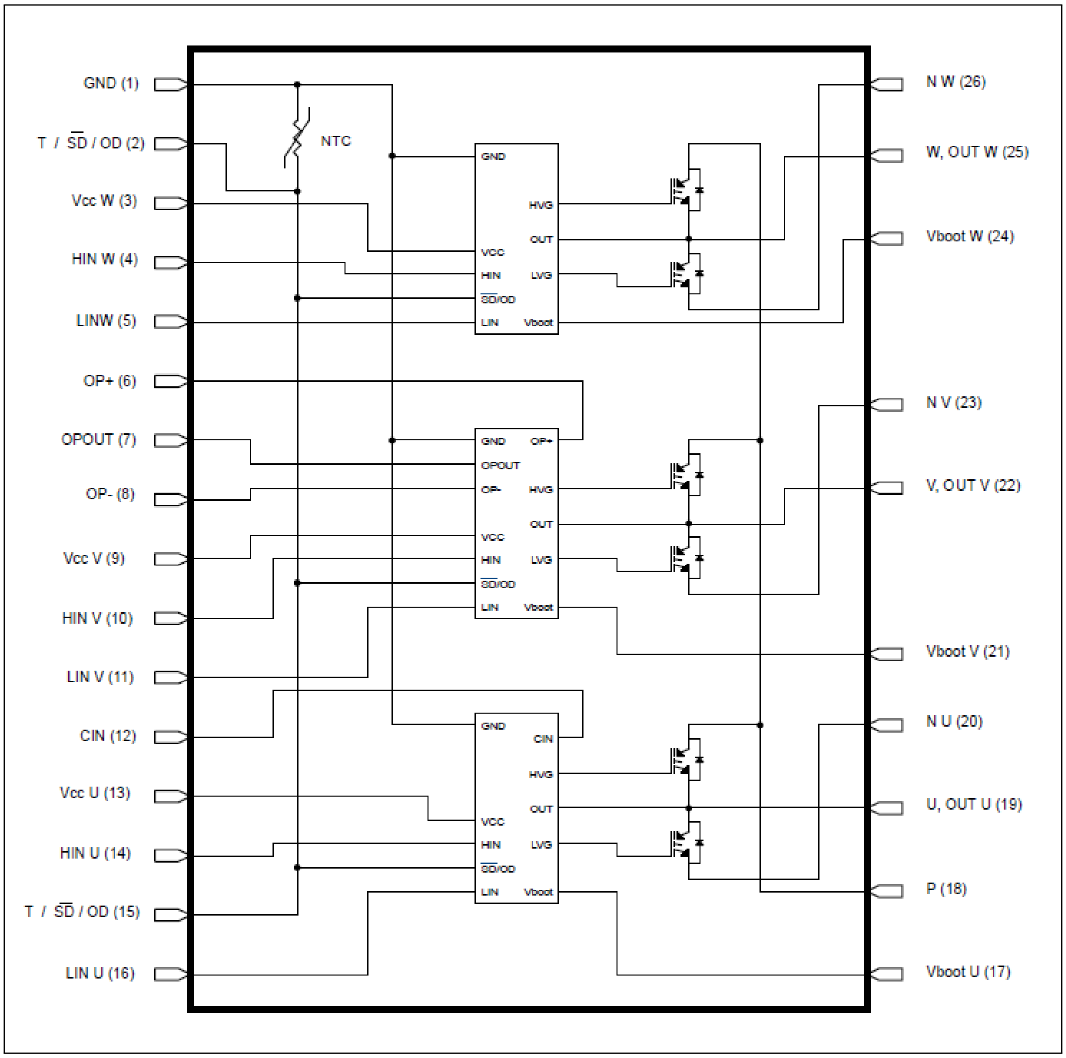


Figure 11: Internal Circuit of the IPM

2.2.1 Power Stage:

The IPM's power stage comprises six very fast Insulated Gate Bipolar Transistors (IGBTs) and their corresponding freewheeling diodes. These components are fundamental in switching the high currents required to drive AC motors efficiently. The IGBTs handle the main switching operations while the freewheeling diodes provide a path for current when the IGBTs are turned off, preventing voltage spikes and ensuring smooth operation.

2.2.2 Driving Network:

To control the IGBTs effectively, the IPM includes three high-voltage gate drivers. These drivers ensure precise switching of the IGBTs by providing sufficient voltage and current to their gates. Gate resistors are also incorporated to control the switching speed and reduce noise. Additionally, three bootstrap diodes supply the necessary voltage to the high-side gate drivers, enabling them to operate efficiently.

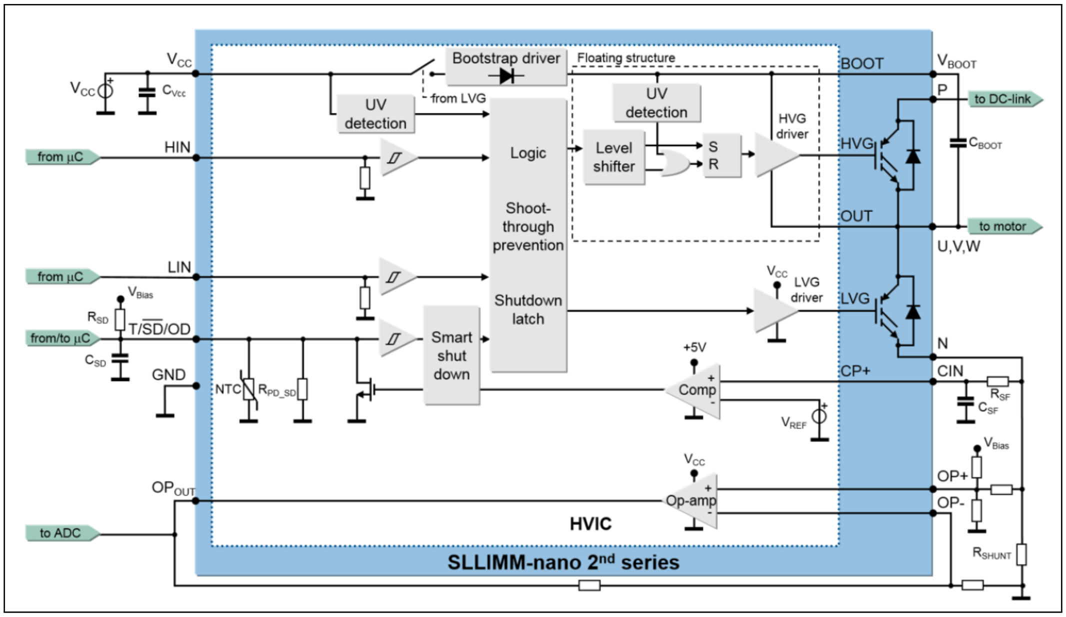


Figure 12: High Voltage Gate Driver Block Diagram

2.2.3 Protection and Optional Features:

- Op-amp for Advanced Current Sensing: The IPM integrates an operational amplifier (op-amp) for advanced current sensing. This feature allows accurate monitoring of the current flowing through the IGBTs, enabling the VFD to maintain optimal performance and protect against overcurrent conditions.

- Comparator for Fault Protection: A comparator is included to detect faults such as overcurrent and short-circuits. This feature provides rapid response to abnormal operating conditions, triggering protective measures to prevent damage to the IPM and connected components.

- Smart Shutdown Function: The IPM is equipped with a smart shutdown function that can deactivate the power stage in case of critical faults or errors. This function enhances the safety and reliability of the VFD by preventing potentially hazardous situations.

- NTC Thermistor: An NTC (Negative Temperature Coefficient) thermistor is employed for temperature sensing. It monitors the temperature of the IPM and surrounding components, helping to prevent overheating and ensuring thermal stability during operation.

- Dead-time, Interlocking Function, and Undervoltage Lockout: The IPM includes features such as dead-time management to prevent simultaneous conduction of IGBTs, interlocking functions to ensure proper sequencing of operations, and undervoltage lockout to protect against insufficient supply voltages.

These integrated features collectively enhance the performance, reliability, and safety of the VFD by providing robust power handling capabilities, precise control of switching operations, and comprehensive protection against various operational anomalies.

2.2.4 Input RC Filter

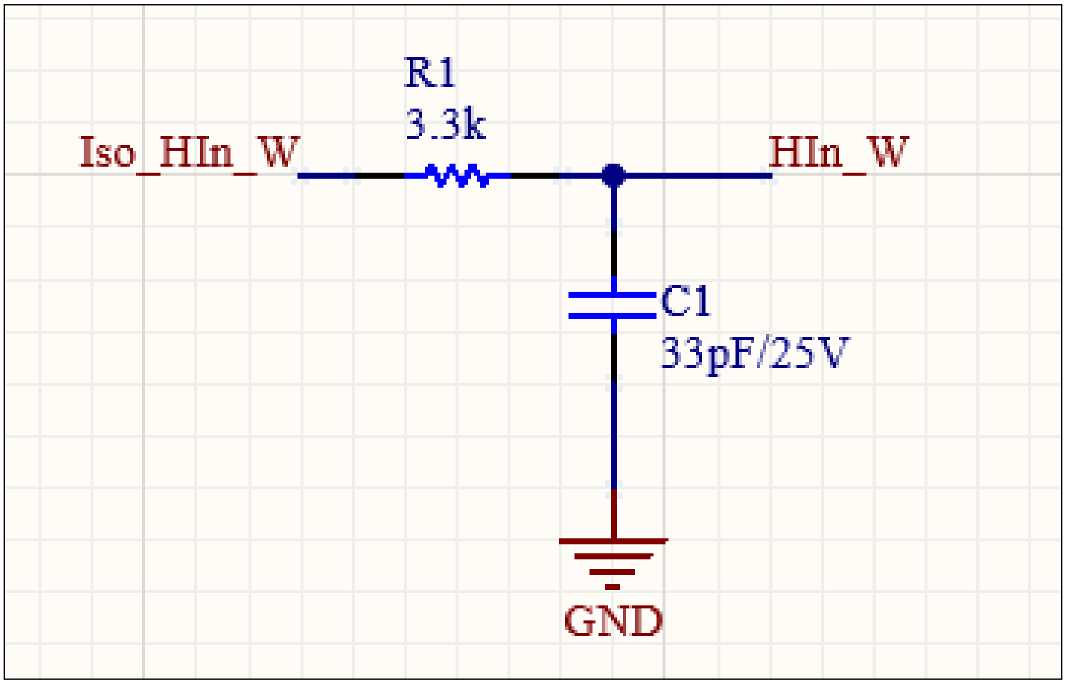


Figure 13: Input Signal Management

The STGIPQ8C60T-HZ Intelligent Power Module (IPM) uses active-high logic for its HIN and LIN input signals. Each high-side input includes a typical 375 kΩ pull-down resistor to prevent floating inputs. To prevent input signal oscillation, wiring should be as short as possible. Additionally, it is recommended to use RC filters (R1, C1) on each input signal with a time constant of about 100 ns. These filters should be placed as close to the IPM input pins as possible to filter out high-frequency noise and ensure signal stability. Given that we have six PWM input signals, each requires an RC filter to prevent signal oscillation. These steps help maintain clean and stable input signals, enhancing the reliability of the VFD system.

2.2.5 Decoupling and Bootstrap Capacitors

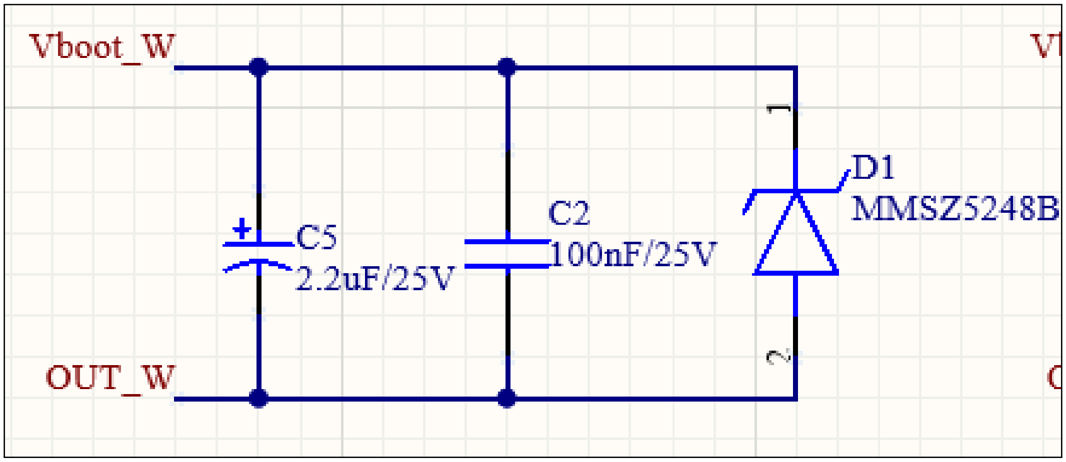


Figure 14: Bootstrap Circuit

To enhance the stability and performance of the STGIPQ8C60T-HZ Intelligent Power Module (IPM), it is recommended to use decoupling capacitors and additional components in the bootstrap circuit.

A decoupling capacitor (C2) with a value between 100 to 220 nF, featuring low Equivalent Series Resistance (ESR) and low Equivalent Series Inductance (ESL), should be placed in parallel with each bootstrap capacitor (Cboot). These decoupling capacitors help to filter high-frequency disturbances and ensure the smooth operation of the IPM. Both Cboot and C3 should be placed as close as possible to the U, V, W, and Vboot pins to maximize their effectiveness.

The negative electrodes of the bootstrap capacitors(C5) should be connected directly to the U, V, and W terminals, and kept separate from the main output wires. This separation helps to minimize noise and ensure accurate voltage levels.

To prevent overvoltage conditions, it is suggested to place a zener diode (D1) in parallel with each bootstrap capacitor (C5). This zener diode will clamp any excessive voltage, protecting the IPM and associated components from potential damage.

By implementing these measures, the VFD system can maintain stable operation, filter out high-frequency noise, and protect against overvoltage, thus enhancing overall reliability and performance.

2.2.6 Internal Non-Inverting Comparator

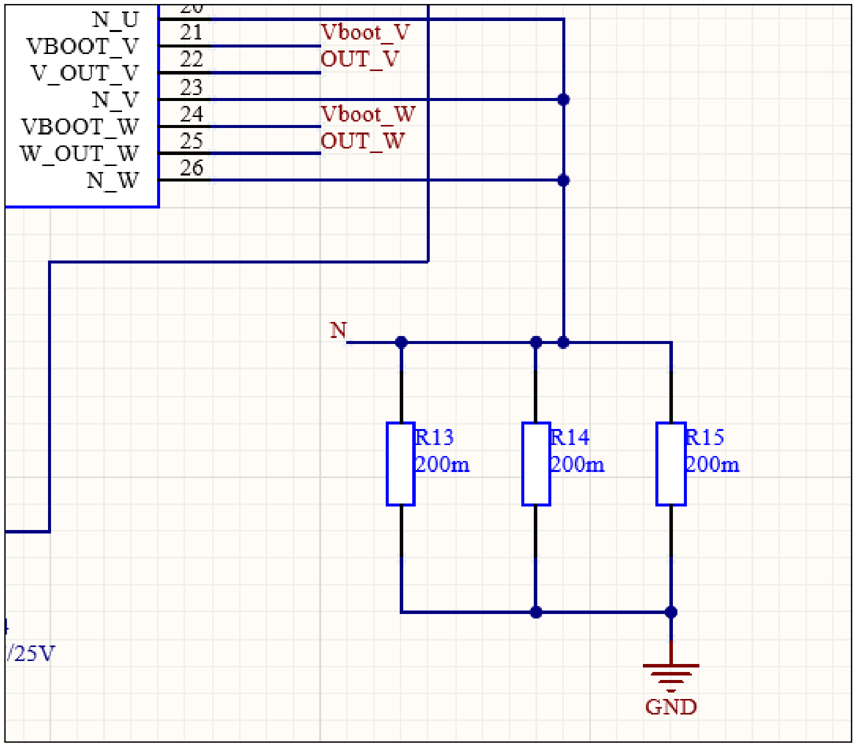


Figure 15: Shunt Resistors

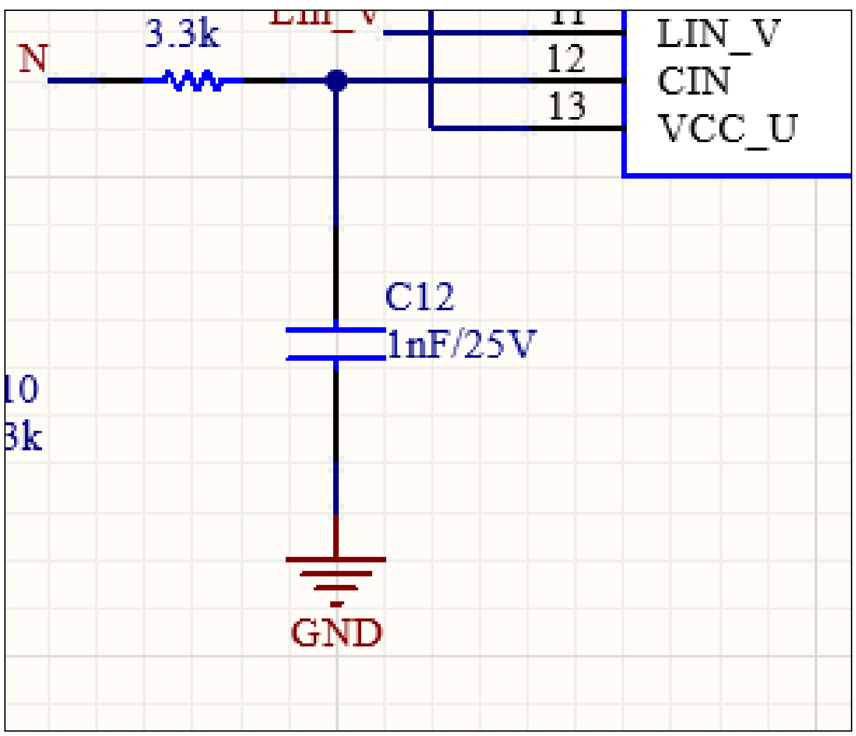


Figure 16: Cin Pin Configuration

The STGIPQ8C60T-HZ Intelligent Power Module (IPM) features an internal non-inverting comparator accessible via the CIN pin. This comparator can be used for short-circuit current detection, leveraging the current sensing shunt resistor connected to each phase leg.

Current Sensing: The shunt resistor, placed in each phase leg, allows for the detection of short-circuit currents through the internal comparator at the CIN pin. It is crucial to select a shunt resistor that meets the specific detection levels required for your application.

Noise Elimination: To filter out noise, an RC filter with a time constant of approximately 1 microsecond should be connected to the CIN pin. This filter helps to ensure accurate current sensing by eliminating high-frequency disturbances.

Connection Considerations: The connection length between the shunt resistor and the CIN pin should be minimized to reduce potential noise pickup and signal degradation.

Overvoltage Protection: If the voltage applied to the CIN pin exceeds the specified reference voltage (VREF) detailed in the datasheet, the IPM will automatically shut down. Additionally, the T/SD/OD pin will be pulled down, signaling the microcontroller of the shutdown event. This feature helps protect the IPM and the overall system from overcurrent conditions.

By implementing these guidelines, the internal comparator can effectively detect short-circuit conditions, enhance the reliability of the current sensing mechanism, and protect the IPM from potential overcurrent damage.

2.2.7 NTC Thermistor / Shutdown / Open Drain

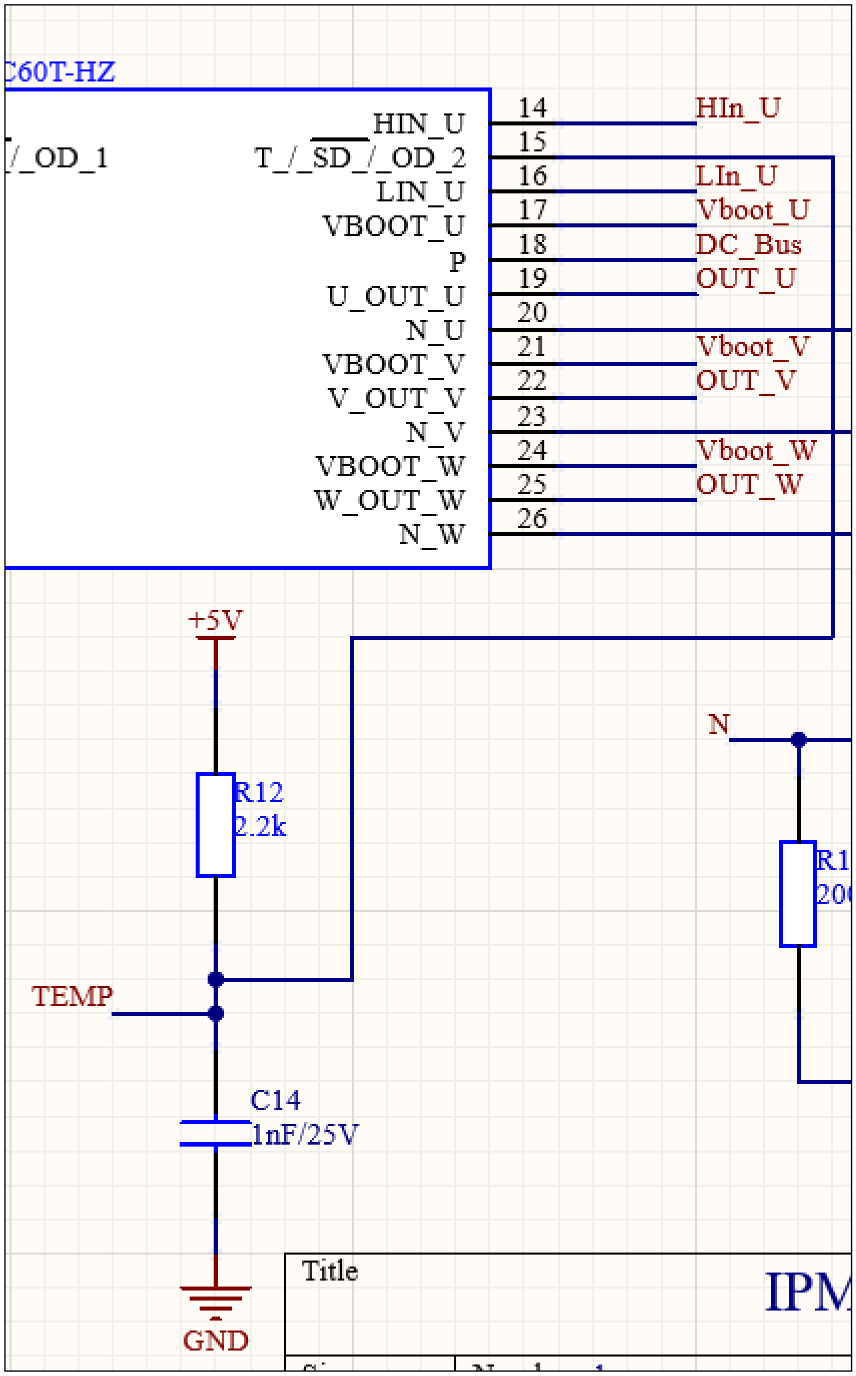


Figure 17: T/SD/OD Pin Configuration

The STGIPQ8C60T-HZ Intelligent Power Module (IPM) features T / SD / OD pins that provide shutdown and open drain functionalities, as well as temperature monitoring capabilities through an integrated NTC thermistor.

Pin Availability and Flexibility: There are two T / SD / OD pins, located on opposite ends of the IPM package to offer higher flexibility in PCB layout. Only one of these pins needs to be used for proper device operation.

Enable/Disable Function: The T / SD / OD pins serve as enable/disable controls for the IPM. These pins operate with active-low logic, meaning the device will shut down if a voltage below a specific threshold is applied. When this occurs, each half-bridge of the IPM goes into a tri-state mode.

Integration with Internal Comparator: The T / SD / OD pins are linked to the status of the internal comparator, which handles short-circuit protection and smart shutdown functions. If the comparator detects a fault, the T / SD / OD pin is pulled down, functioning as a FAULT indicator.

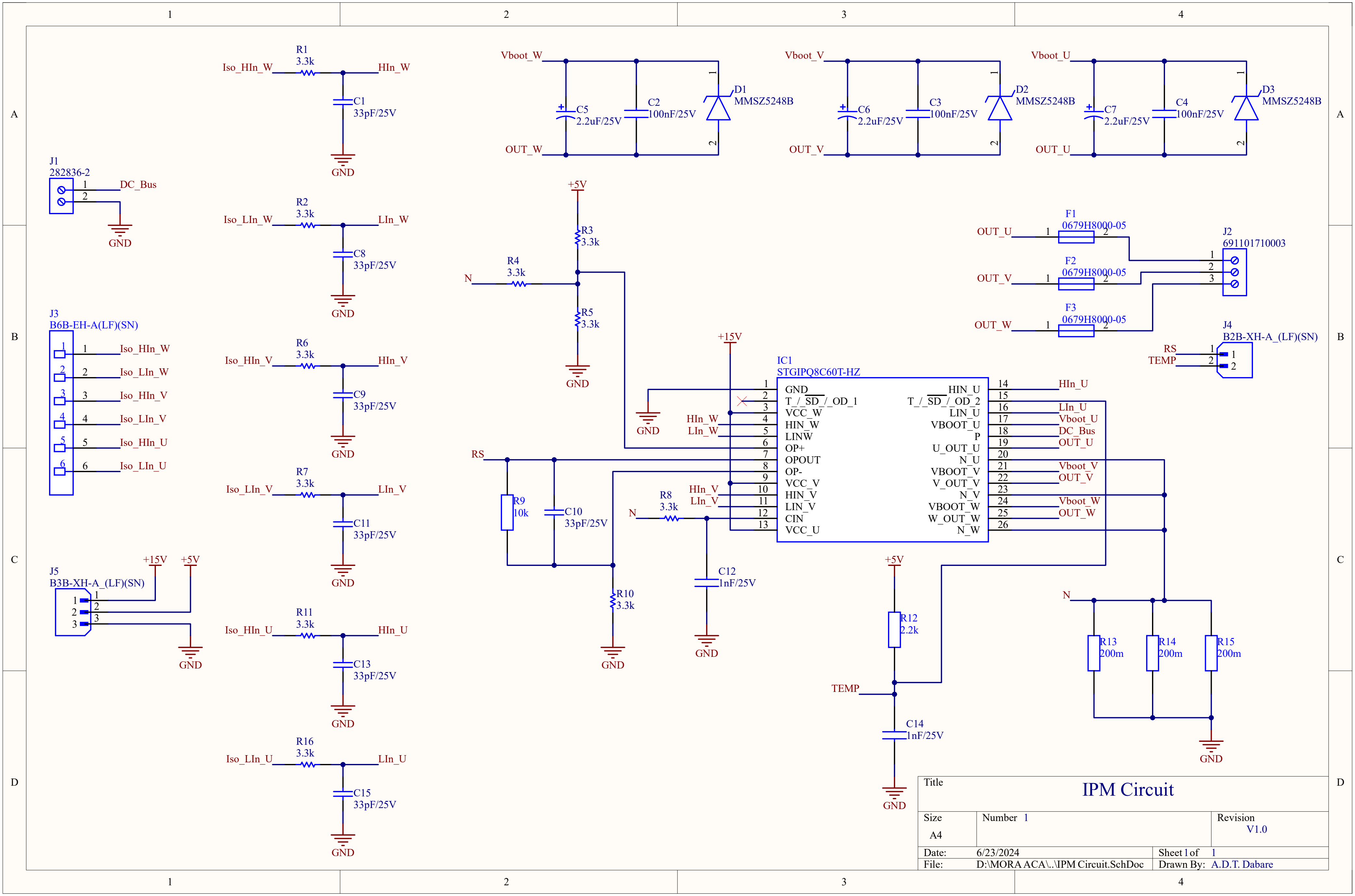
Open Drain Configuration: When the comparator triggers, the T / SD / OD pins are configured in an open drain mode. The voltage at these pins should be pulled up to the 3.3V or 5V logic power supply through an appropriate pull-up resistor. For 3.3V MCU power supplies, use a 1 kΩ resistor (RSD), and for 5V supplies, use a 2.2 kΩ resistor.

Temperature Monitoring: The T / SD / OD pins can also be used for temperature monitoring due to the co-packaged NTC thermistor. To prevent undesired shutdowns, a capacitor (CSD) with a time constant no higher than 1 microsecond should be used in conjunction with the pull-up resistor.

By implementing these guidelines, you can ensure reliable operation, effective fault detection, and protection, as well as accurate temperature monitoring of the IPM.

Refer this document for a detailed description of the SLLIMM-nano 2nd series providing guidlines for motor drive control.

2.2.8 Schematic Design of the IPM Circuit



R1 3.3k
Iso_HIn_W
HIn_W
C1 33pF/25V
GND
R2 3.3k
Iso_LIn_W
LIn_W
C8 33pF/25V
R6 3.3k
Iso_HIn_V
HIn_V
C9 33pF/25V
R7 3.3k
Iso_LIn_V
LIn_V
C11 33pF/25V
R11 3.3k
Iso_HIn_U
HIn_U
C13 33pF/25V
R16 3.3k
Iso_LIn_U
LIn_U
C15 33pF/25V
J1 282836-2
DC_Bus
J3 B6B-EH-A(LF)(SN)
J5 B3B-XH-A_(LF)(SN)
+15V
+5V
Vboot_W
OUT_W
C5 2.2uF/25V
C2 100nF/25V
D1 MMSZ5248B
Vboot_V
OUT_V
C6 2.2uF/25V
C3 100nF/25V
D2 MMSZ5248B
Vboot_U
OUT_U
C7 2.2uF/25V
C4 100nF/25V
D3 MMSZ5248B
R3 3.3k
R4 3.3k
R5 3.3k
N
F1 0679H8000-05
F2 0679H8000-05
F3 0679H8000-05
J2 691101710003
J4 B2B-XH-A_(LF)(SN)
RS
TEMP
IC1 STGIPQ8C60T-HZ
GND
T_/_SD_/_OD_1
VCC_W
HIN_W
LINW
OP+
OPOUT
OP-
VCC_V
HIN_V
LIN_V
CIN
VCC_U
HIN_U
T_/_SD_/_OD_2
LIN_U
VBOOT_U
P
U_OUT_U
N_U
VBOOT_V
V_OUT_V
N_V
VBOOT_W
W_OUT_W
N_W
R9 10k
C10 33pF/25V
R8 3.3k
C12 1nF/25V
R10 3.3k
R12 2.2k
C14 1nF/25V
R13 200m
R14 200m
R15 200m
Title IPM Circuit
Size A4
Number 1
Revision V1.0
Date: 6/23/2024
Sheet 1 of 1
File: D:\MORA ACA\..\IPM Circuit.SchDoc
Drawn By: A.D.T. Dabare

2.2.9 PCB Layout of the IPM Circuit

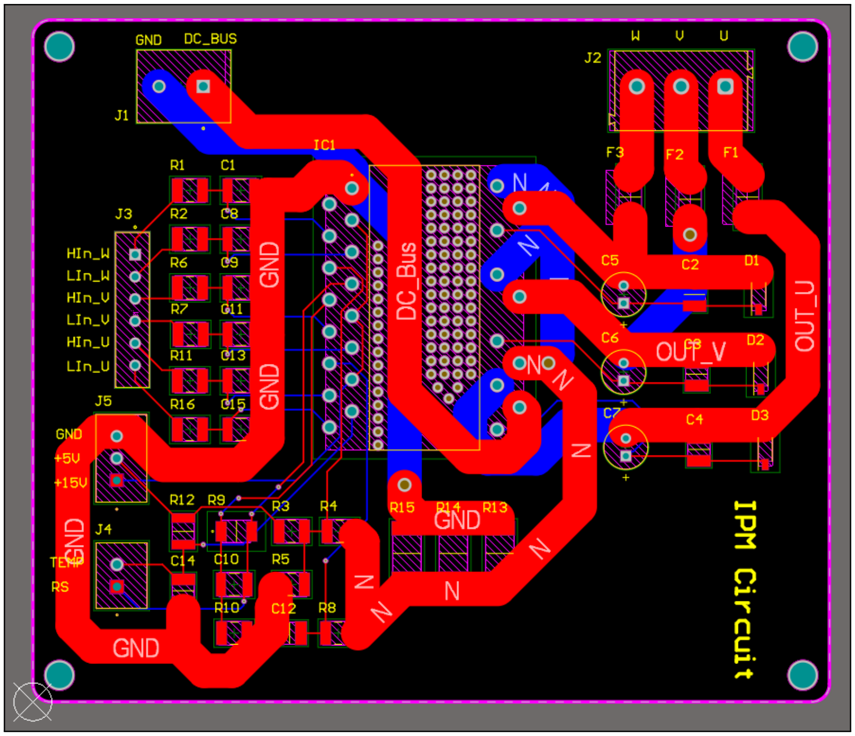


Figure 18: IPM Circuit Top Layer PCB 2D Layout

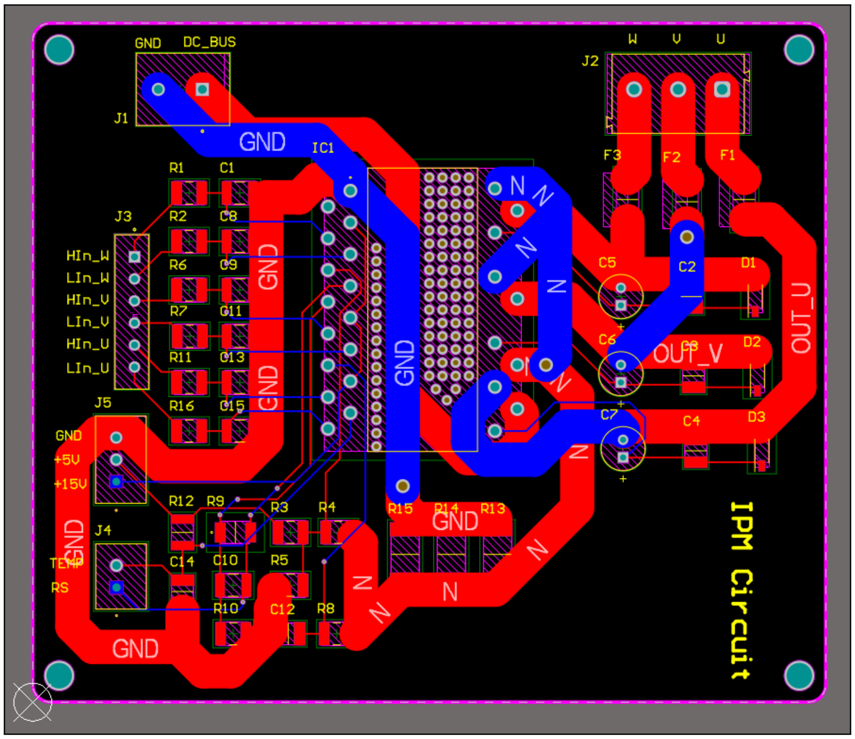


Figure 19: IPM Circuit Bottom Layer PCB 2D Layout

- Dimensions - 90mm x 77mm
- Outer Copper Weight - 2 Oz

- 10A Trace Width - 4mm
- 1A Trace Width - 0.3mm

Inputs:

Current	10	Amps
Thickness	2	oz/ft^2

Optional Inputs:

Temperature Rise	10	Deg C
Ambient Temperature	25	Deg C
Trace Length	100	mm

Results for Internal Layers:

Required Trace Width	9.36	mm
Resistance	0.00270	Ohms
Voltage Drop	0.0270	Volts
Power Loss	0.270	Watts

Results for External Layers in Air:

Required Trace Width	3.60	mm
Resistance	0.00701	Ohms
Voltage Drop	0.0701	Volts
Power Loss	0.701	Watts

Figure 20: 10A Trace Width Calculation

Inputs:

Current	1	Amps
Thickness	2	oz/ft^2

Optional Inputs:

Temperature Rise	10	Deg C
Ambient Temperature	25	Deg C
Trace Length	100	mm

Results for Internal Layers:

Required Trace Width	0.391	mm
Resistance	0.0646	Ohms
Voltage Drop	0.0646	Volts
Power Loss	0.0646	Watts

Results for External Layers in Air:

Required Trace Width	0.150	mm
Resistance	0.168	Ohms
Voltage Drop	0.168	Volts
Power Loss	0.168	Watts

Figure 21: 1A Trace Width Calculation

Please visit this Website for trace width calculation.

2.2.10 Component List of the IPM Circuit

Item	Quantity	Reference	Part
1	1	J1	5.08mm Pitch 2-Pin 2-Way Screw Terminal Block PCB Mount
2	1	J2	5.08mm Pitch 3-Pin 3-way Pluggable Screw Terminal Block PCB Mount
3	3	F1-F3	Surface Mount Fuses 8A 350 VAC 72 VCD
4	1	IC1	IGBT Modules SLLIMM nano 2nd series IPM, 3-phase inverter, 8A, 600V
5	1	J3	JST wire connectors 2.54mm with socket - 6 pin
6	1	J5	JST wire connectors 2.54mm with socket - 3 pin
7	1	J4	JST wire connectors 2.54mm with socket - 2 pin
8	3	R13-R15	Current Sense Resistors - SMD TLRP 2512 3.0W R200 1% 25PPM 4K RL
9	11	R1-R8, R10, R11, R16	Thick Film Resistors - SMD CRGCQ 1210 3K3 5% SMD Resistor
10	7	C1, C8-C11, C13, C15	Multilayer Ceramic Capacitors MLCC - SMD/SMT 25V 33pF X8R 1210 10%
11	2	C12, C14	Multilayer Ceramic Capacitors MLCC - SMD/SMT WCAP-CSGP 1000pF 1210 10% 25V MLCC
12	3	C2-C4	Multilayer Ceramic Capacitors MLCC - SMD/SMT KGM32RR71E104MU NEW GLOBAL PN 25V .1uF X7R 1210 20%
13	3	D1-D3	Zener Diodes 500mW,ZENER,SOD-123,18V
14	1	R12	Thick Film Resistors - SMD 1/2watt 2.2Kohms 5%
15	1	R9	Thick Film Resistors - SMD 1210 10Kohms 1% AEC-Q200

2.3 MCU Circuit

The Microcontroller Unit (MCU) Circuit acts as the brain of the VFD, executing the control algorithms and handling communication with other system components. It processes input from sensors, user commands, and feedback from the IPM circuit to generate precise PWM signals that control the inverter stage. The MCU also manages system diagnostics, fault detection, and user interface, ensuring that the VFD operates within its specified parameters and responds appropriately to any changes in operating conditions. We are using the ATmega328-PU as our microcontroller.

2.3.1 PWM Signals

The MCU generates six PWM signals that are used to control the IPM (Intelligent Power Module). These PWM signals modulate the voltage and frequency supplied to the AC motor, enabling precise control over its speed and torque.

2.3.2 Isolators

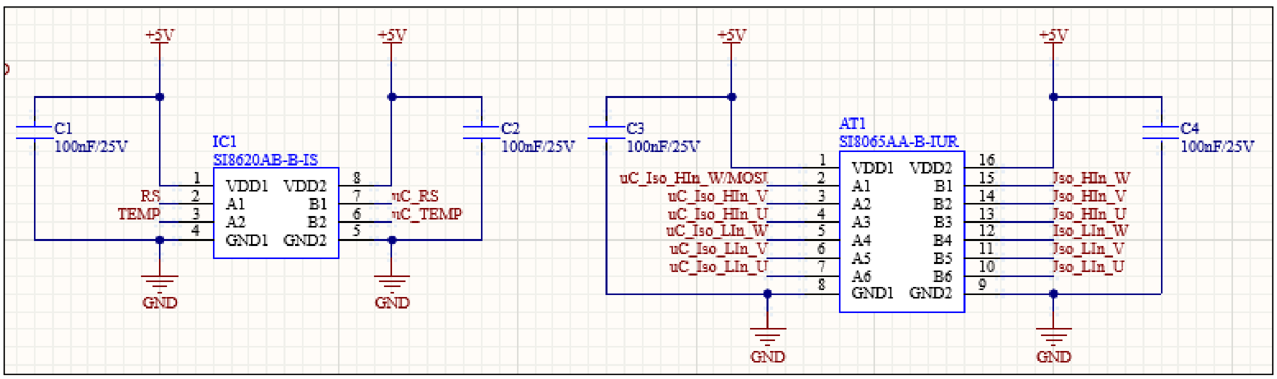


Figure 22: Isolators

To ensure safety and proper signal integrity, two isolators are used in the circuit. These isolators separate the high and low voltage sides, protecting the MCU from potential high voltage surges and ensuring that the PWM signals and feedback signals are transmitted without interference. (Since the IPM has integrated an application-specific type HVIC inside the module, direct coupling to the MCU terminals without an optocoupler is possible.)

2.3.3 Other Input/Output Pins

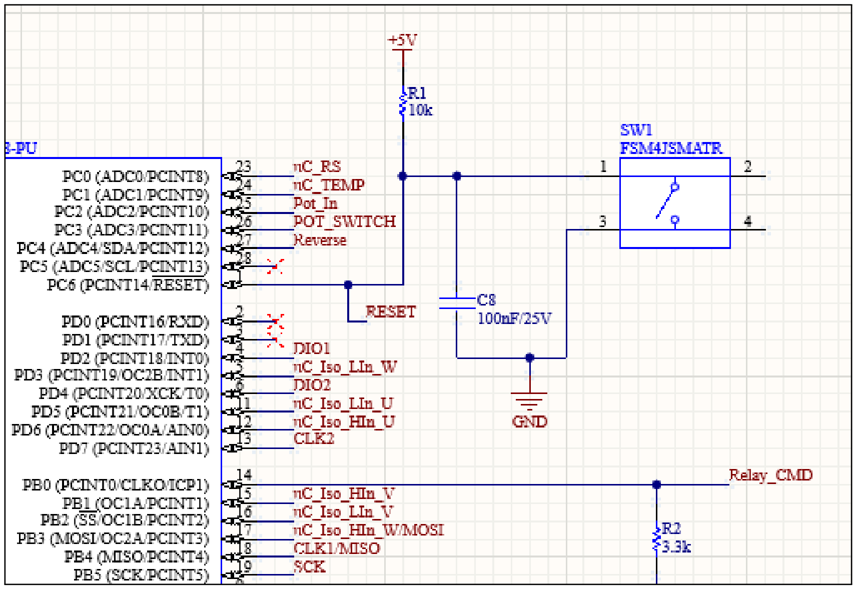


Figure 23: Other Input/Output Pins

- POT_SWITCH: This pin is connected to a rocker switch, which signals the MCU to start generating PWM signals. The switch acts as an on/off control for the VFD system.
- POT_In: This pin is connected to a potentiometer, allowing the user to adjust the motor speed. By varying the resistance, the potentiometer changes the input voltage, which the MCU interprets to adjust the frequency of the PWM signals, typically ranging from 20 Hz to 120 Hz.
- Display Control Pins: The MCU also has pins (DIO1, DIO2, CLK1, CLK2) dedicated to controlling two 7-segment, 4-bit digital LED displays (TM1637). These displays show the frequency of the PWM signals and the current flowing through the motor, providing real-time feedback to the user.
- Relay_CMD: This pin generates a signal to bypass the initial charging resistor in the power circuit after an 8-second delay. This command activates a relay to ensure the capacitors are fully charged before the resistor is bypassed, preventing high inrush currents.

2.3.4 Oscillator

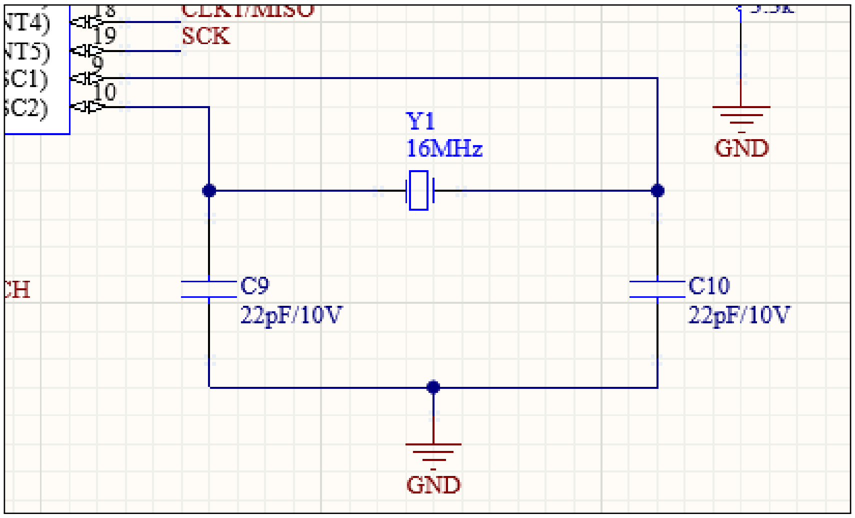


Figure 24: Oscillator

The MCU uses a 16 MHz oscillator(Y1) to provide a stable clock signal. This oscillator ensures precise timing for the generation of PWM signals and other time-sensitive operations within the MCU.

2.3.5 Input Power and Decoupling Capacitors

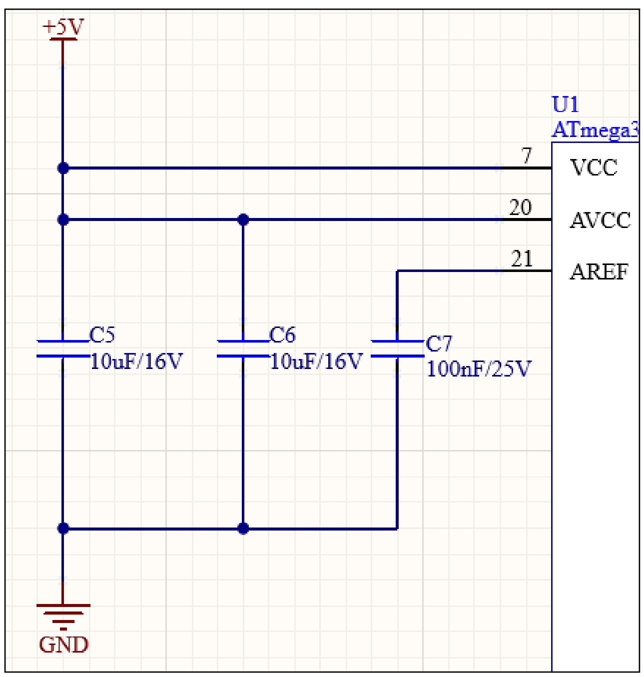
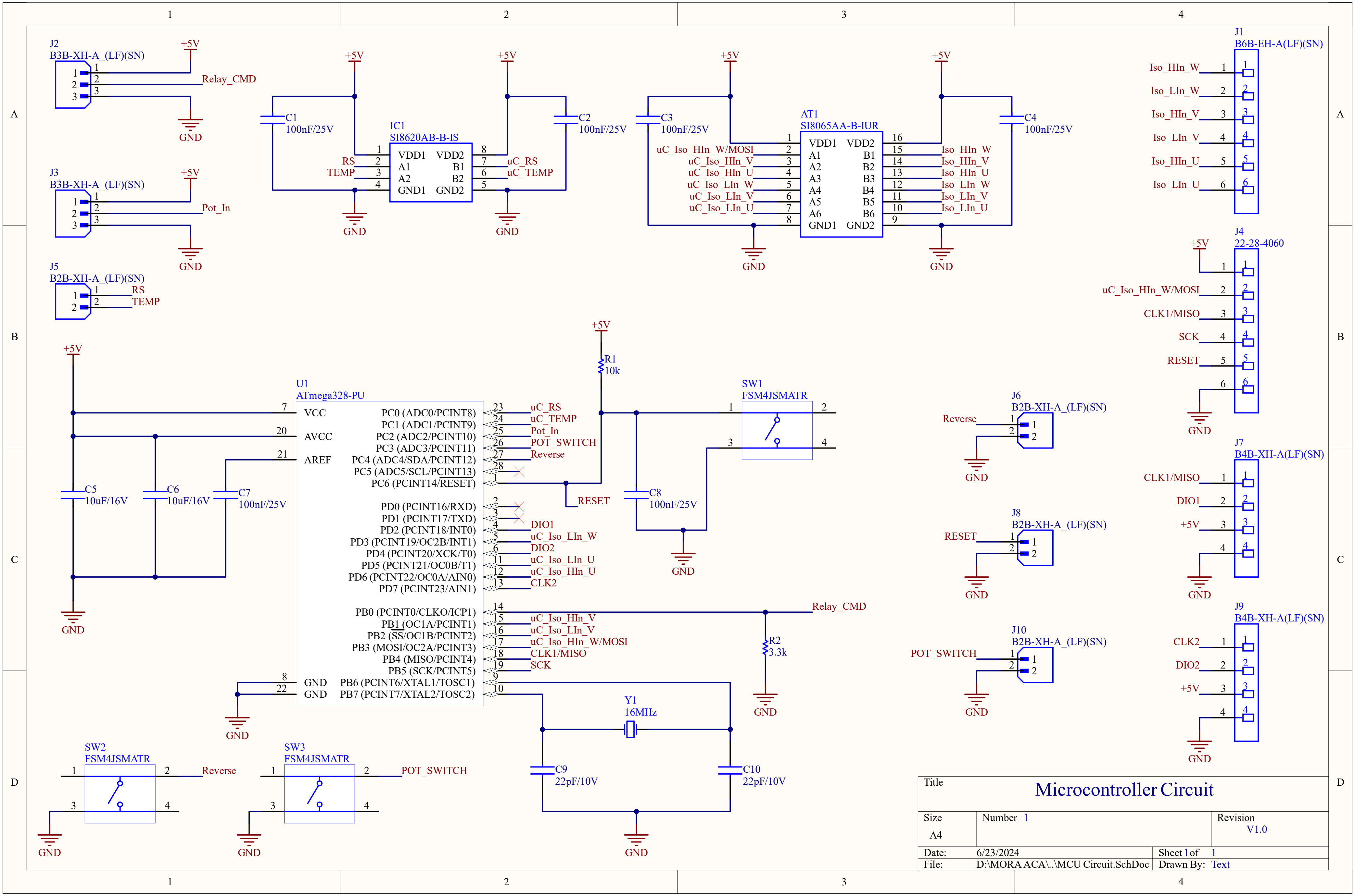


Figure 25: Oscillator

To ensure stable operation, the MCU circuit is equipped with decoupling capacitors (C5-C7). These capacitors filter out noise and stabilize the power supply, preventing voltage fluctuations from affecting the MCU's performance.

2.3.6 Schematic Design of the MCU Circuit


J2
B3B-XH-A_(LF)(SN)
+5V
Relay_CMD
GND
J3
B3B-XH-A_(LF)(SN)
Pot_In
J5
B2B-XH-A_(LF)(SN)
RS
TEMP
C1
100nF/25V
IC1
SI8620AB-B-IS
VDD1 VDD2
A1 B1
A2 B2
GND1 GND2
uC_RS
uC_TEMP
C2
100nF/25V
C3
100nF/25V
AT1
SI8065AA-B-IUR
uC_Iso_HIn_W/MOSI
uC_Iso_HIn_V
uC_Iso_HIn_U
uC_Iso_LIn_W
uC_Iso_LIn_V
uC_Iso_LIn_U
Iso_HIn_W
Iso_HIn_V
Iso_HIn_U
Iso_LIn_W
Iso_LIn_V
Iso_LIn_U
C4
100nF/25V
J1
B6B-EH-A(LF)(SN)
J4
22-28-4060
CLK1/MISO
SCK
RESET
R1
10k
SW1
FSM4JSMATR
U1
ATmega328-PU
VCC
AVCC
AREF
GND
PC0 (ADC0/PCINT8)
PC1 (ADC1/PCINT9)
PC2 (ADC2/PCINT10)
PC3 (ADC3/PCINT11)
PC4 (ADC4/SDA/PCINT12)
PC5 (ADC5/SCL/PCINT13)
PC6 (PCINT14/RESET)
PD0 (PCINT16/RXD)
PD1 (PCINT17/TXD)
PD2 (PCINT18/INT0)
PD3 (PCINT19/OC2B/INT1)
PD4 (PCINT20/XCK/T0)
PD5 (PCINT21/OC0B/T1)
PD6 (PCINT22/OC0A/AIN0)
PD7 (PCINT23/AIN1)
PB0 (PCINT0/CLKO/ICP1)
PB1 (OC1A/PCINT1)
PB2 (SS/OC1B/PCINT2)
PB3 (MOSI/OC2A/PCINT3)
PB4 (MISO/PCINT4)
PB5 (SCK/PCINT5)
PB6 (PCINT6/XTAL1/TOSC1)
PB7 (PCINT7/XTAL2/TOSC2)
POT_SWITCH
Reverse
DIO1
DIO2
CLK2
C5
10uF/16V
C6
10uF/16V
C7
100nF/25V
C8
100nF/25V
J6
B2B-XH-A_(LF)(SN)
J7
B4B-XH-A(LF)(SN)
J8
B2B-XH-A_(LF)(SN)
J9
B4B-XH-A(LF)(SN)
J10
B2B-XH-A_(LF)(SN)
R2
3.3k
Y1
16MHz
C9
22pF/10V
C10
22pF/10V
SW2
FSM4JSMATR
SW3
FSM4JSMATR
Title
Microcontroller Circuit
Size
A4
Number 1
Revision
V1.0
Date: 6/23/2024
Sheet 1 of 1
File: D:\MORA ACA\..\MCU Circuit.SchDoc
Drawn By: Text

2.3.7 PCB Layout of the MCU Circuit

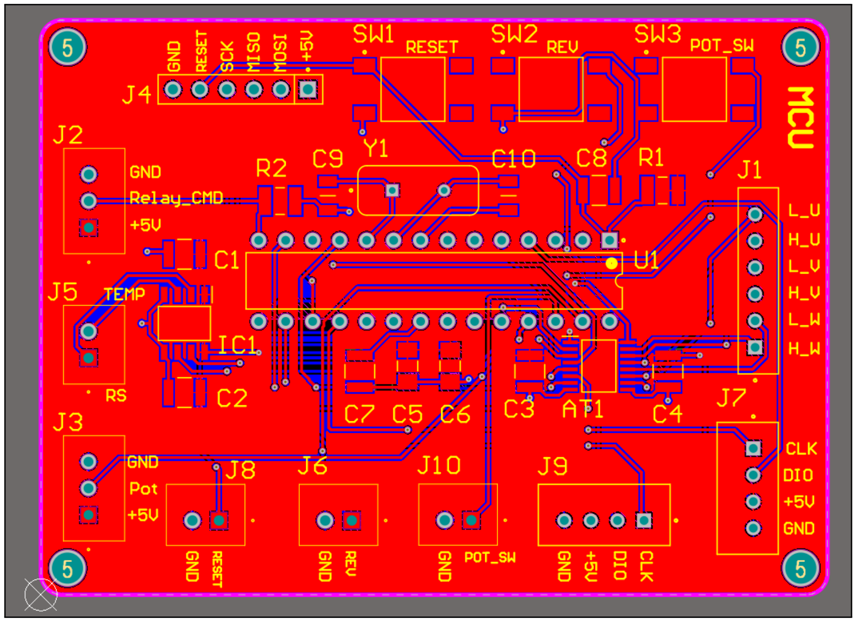


Figure 26: MCU Circuit Top Layer PCB 2D Layout

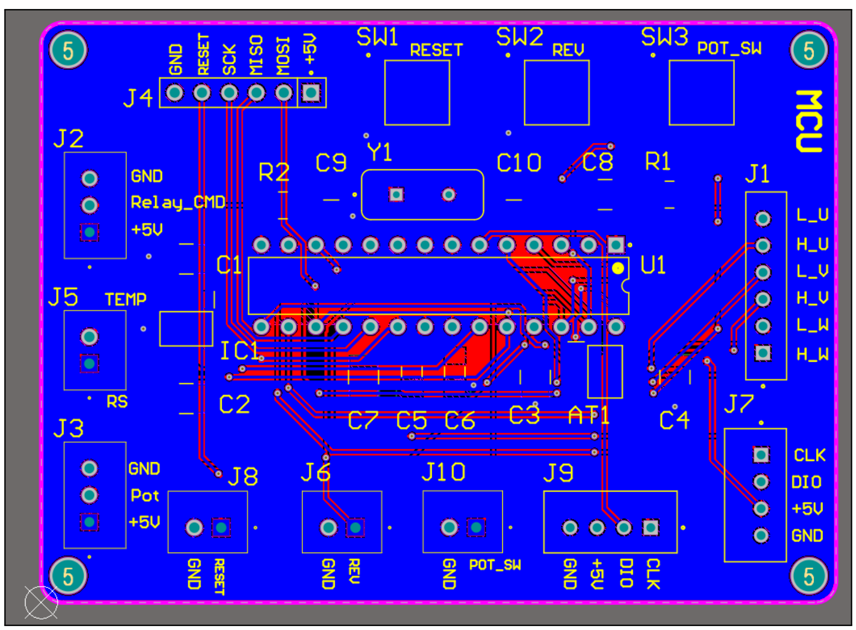


Figure 27: MCU Circuit Bottom Layer PCB 2D Layout

- Dimensions - 74mm x 54mm
- Outer Copper Weight - 1 Oz
- 1A Trace Width - 0.35mm

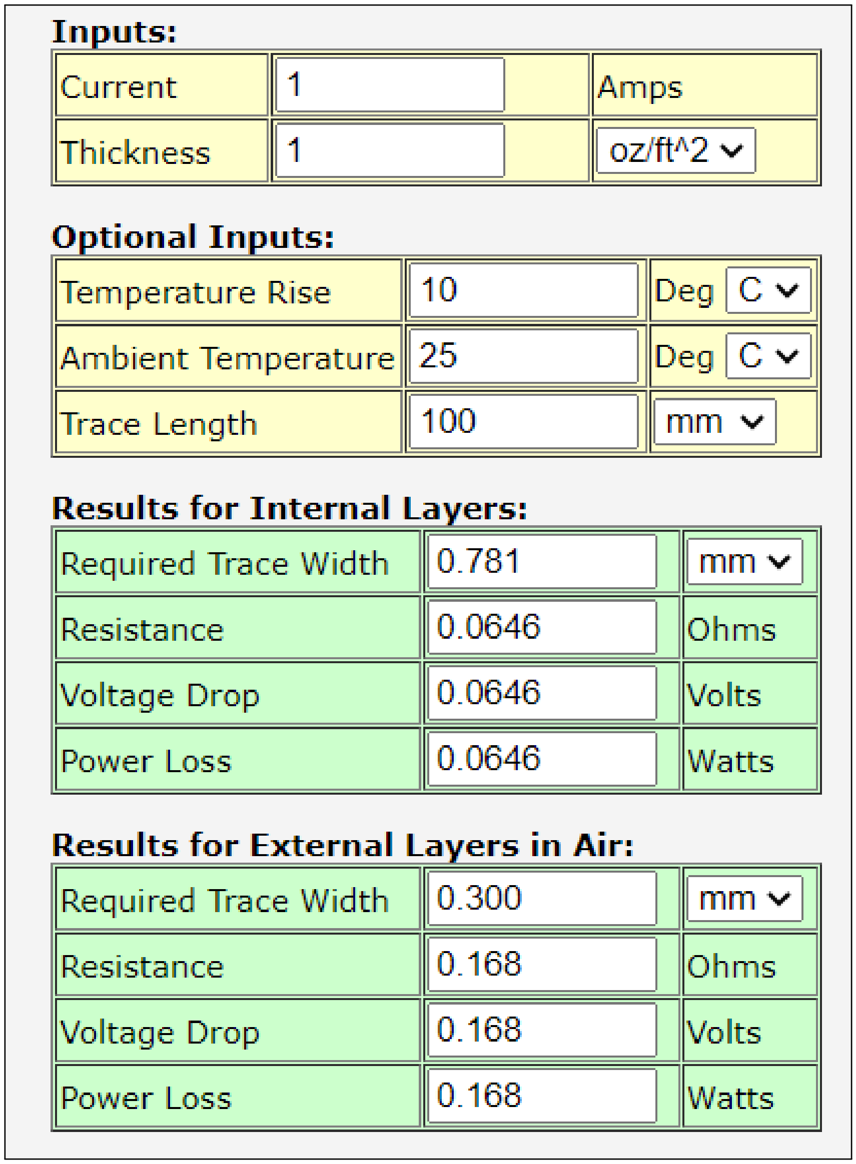


Figure 28: 1A Trace Width Calculation

Please visit this Website for trace width calculation.

2.3.8 Component List of the MCU Circuit

Item	Quantity	Reference	Part
1	1	J1	JST wire connectors 2.54mm with socket - 6 pin
2	2	J2, J3	JST wire connectors 2.54mm with socket - 3 pin
3	4	J5, J6, J8, J10	JST wire connectors 2.54mm with socket - 2 pin
4	1	J4	Long 6-pin Single row male headers 2.54mm
5	1	U1	Atmel ATMEGA328P-PU Microcontroller
6	1	IC1	Digital Isolators 2.5 kV 2-channel digital isolator
7	1	AT1	Digital Isolators 6 ch 1 kV digital isolator
8	3	SW1-SW3	Tactile Switches SPST OF(0N) RND SMT MINI PB TACT SWITCH
9	1	Y1	16MHz Crystal Oscillator - Through Hole
10	6	C1-C4, C7, C8	Multilayer Ceramic Capacitors MLCC - SMD/SMT KGM32RR71E104MU NEW GLOBAL PN 25V .1uF X7R 1210 20%
11	1	R2	Thick Film Resistors - SMD CRGCQ 1210 3K3 5% SMD Resistor
12	1	R1	Thick Film Resistors - SMD 1210 10Kohms 1% AEC-Q200
13	2	C5, C6	Multilayer Ceramic Capacitors SMD/SMT 16V 10uF 1210
14	2	C9, C10	Multilayer Ceramic Capacitors MLCC - SMD/SMT WCAP-CSGP 22pF 1206 5% 10V MLCC

3 Photographs of the PCB

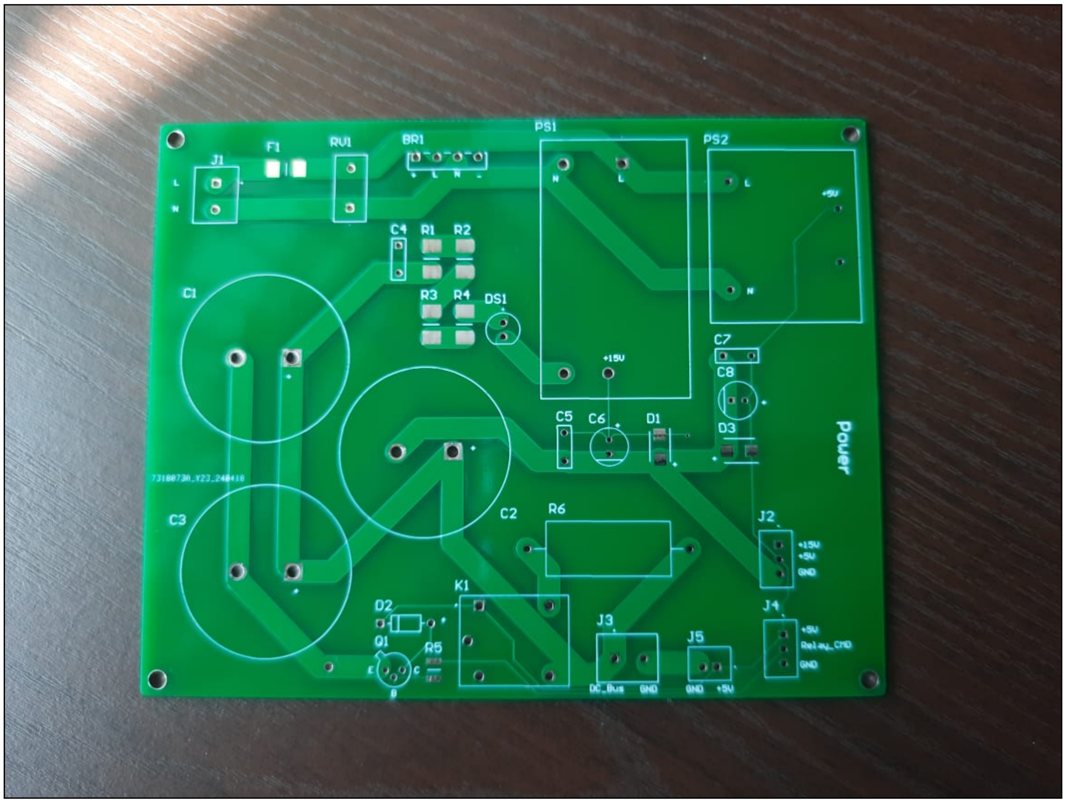


Figure 29: Bare PCB of the Power Circuit

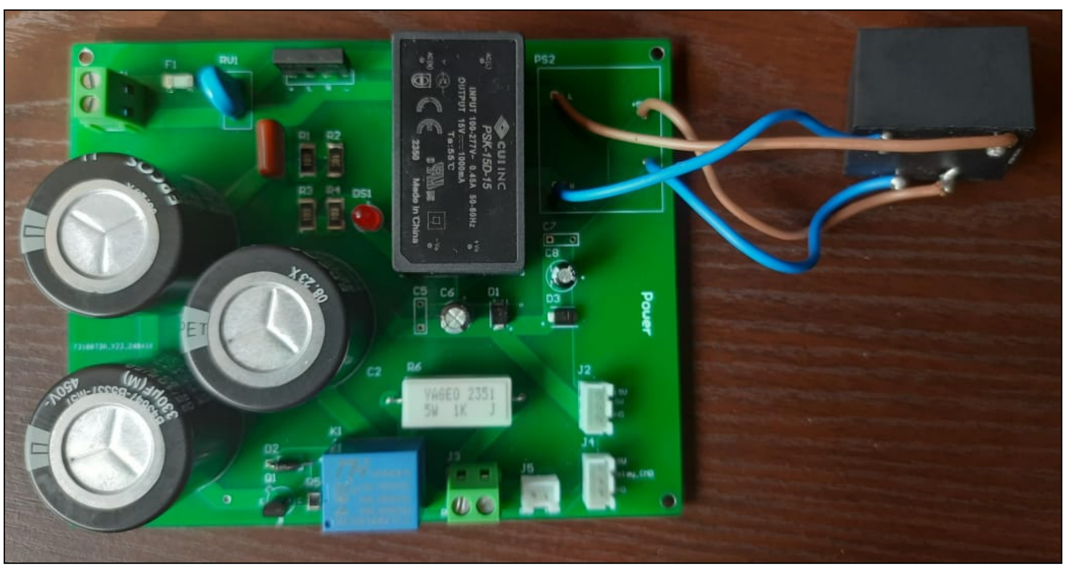


Figure 30: Soldered PCB of the Power Circuit

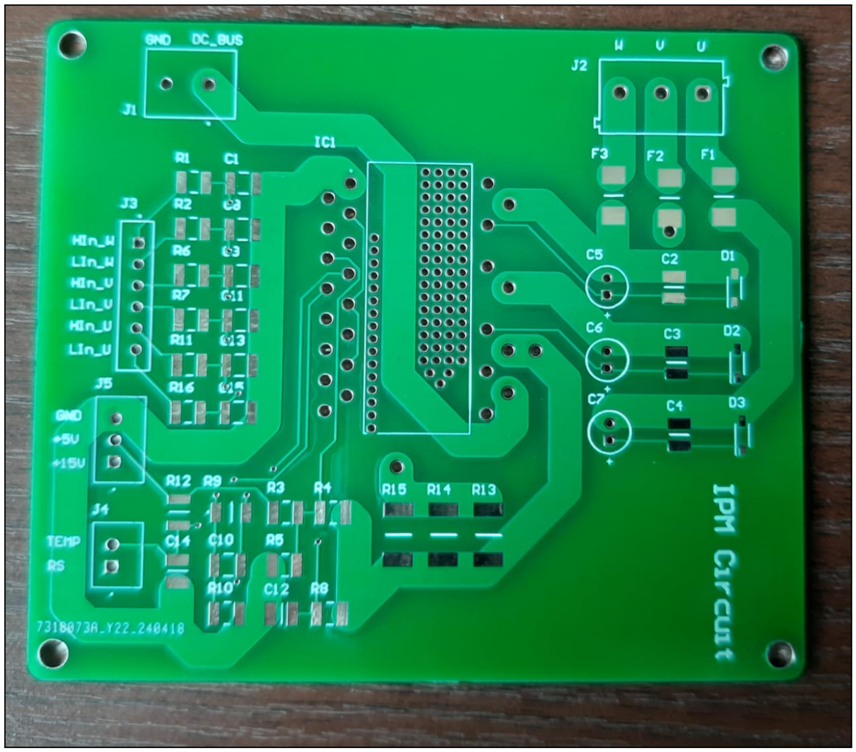


Figure 31: Bare PCB of the IPM Circuit

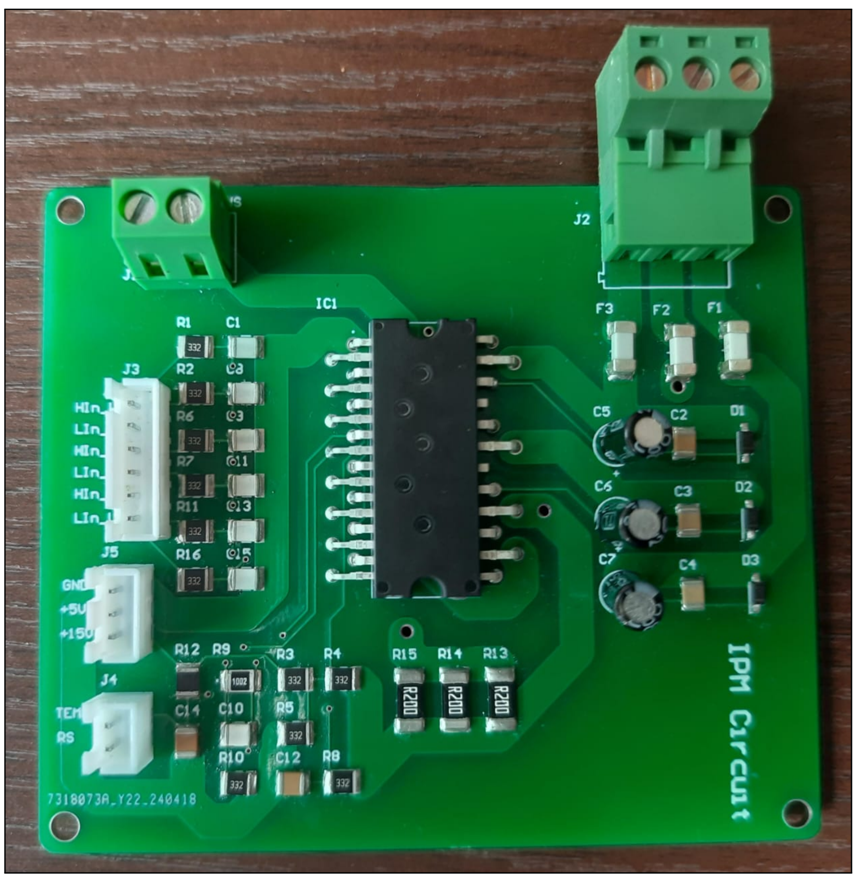


Figure 32: Soldered PCB of the IPM Circuit

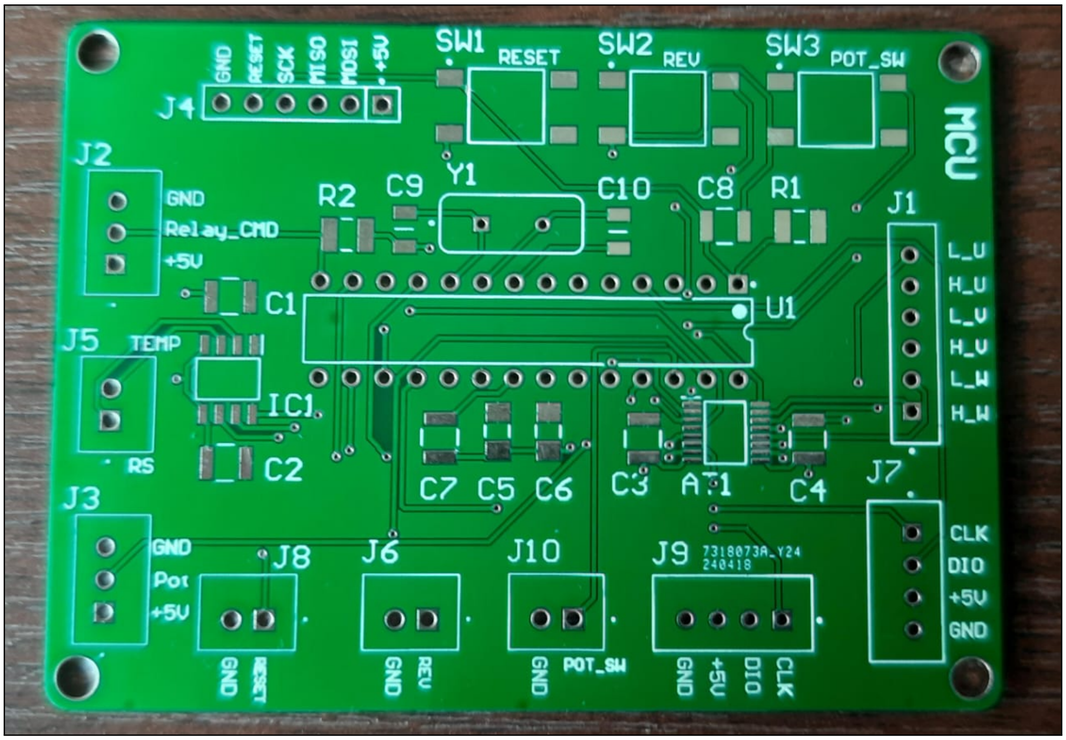


Figure 33: Bare PCB of the MCU Circuit

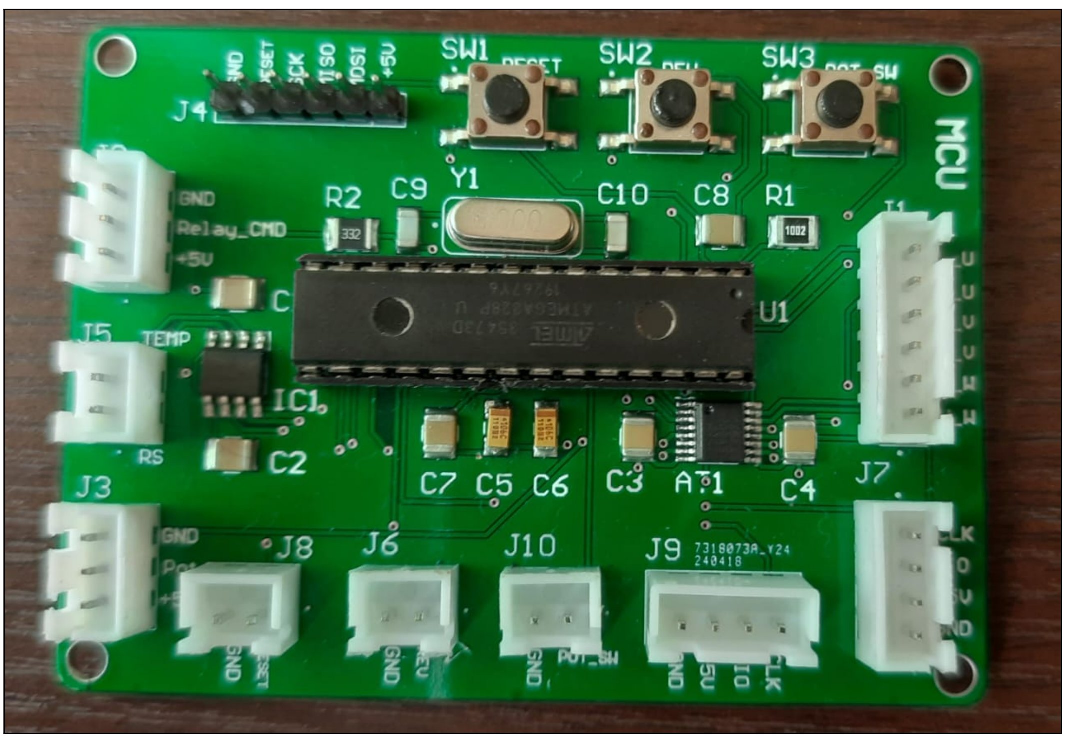


Figure 34: Soldered PCB of the MCU Circuit

4 Enclosure Design

4.1 Enclosure Assembly

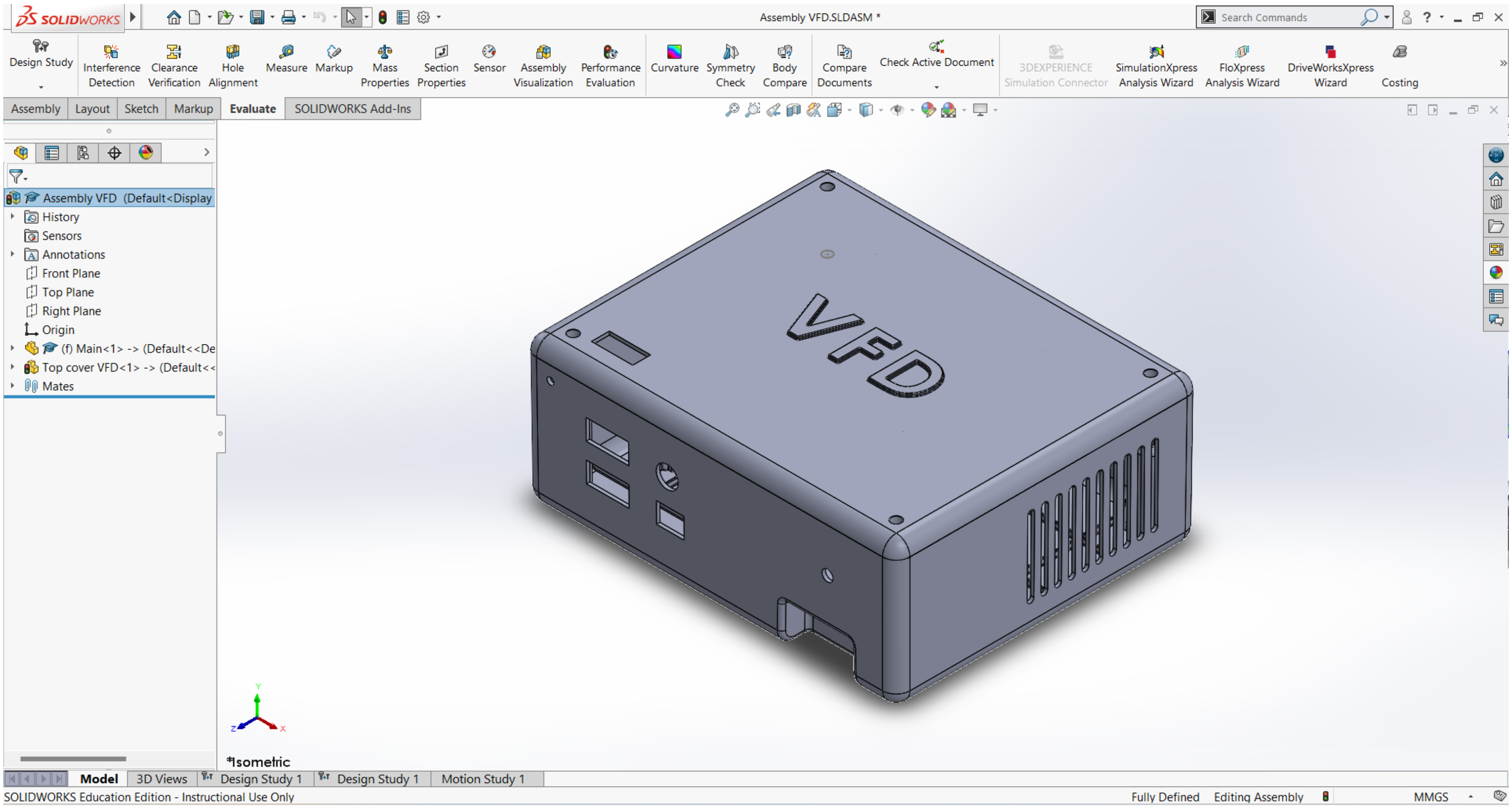


Figure 35: Isometric View

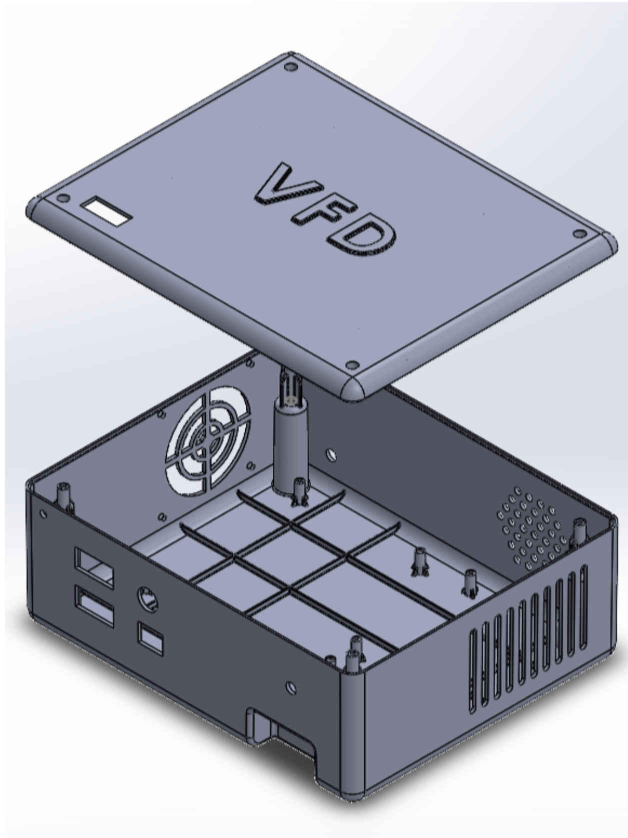


Figure 36: Exploded View - 1

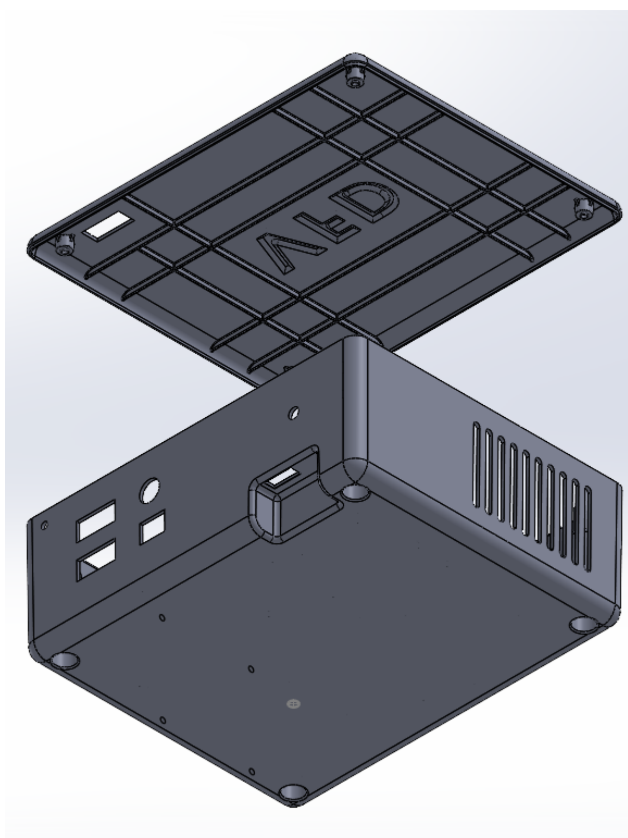


Figure 37: Exploded View - 2

4.2 Bottom Part

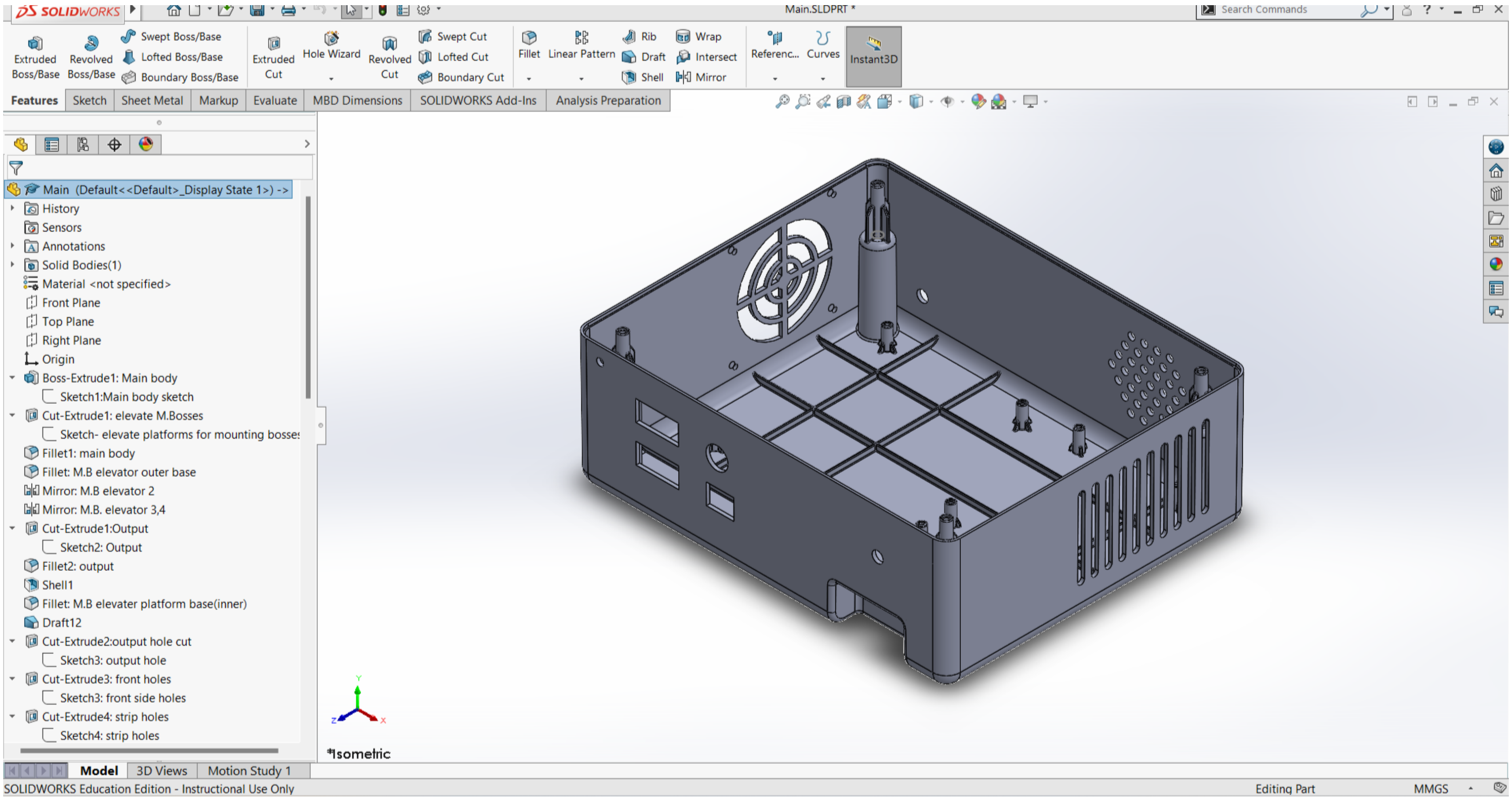


Figure 38: Isometric View

4.2.1 Bottom Part Mold Design

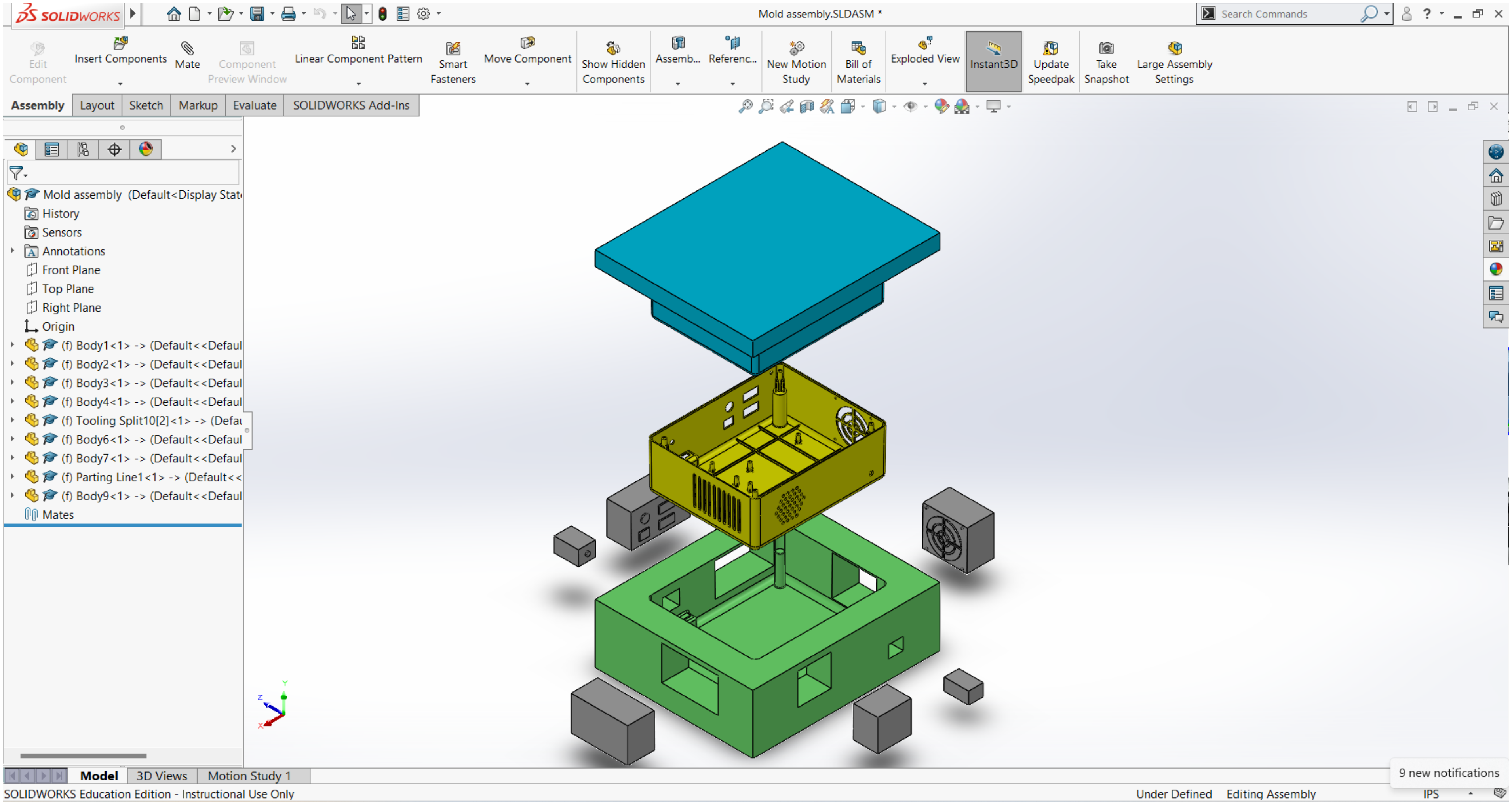


Figure 39: Mold Design Exploded View - 1

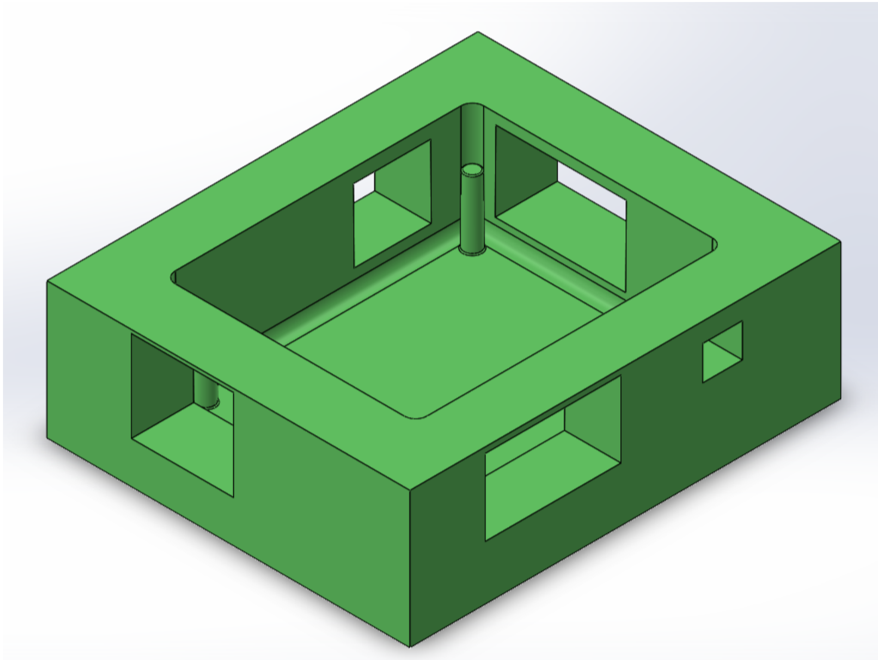


Figure 40: Mold Cavity

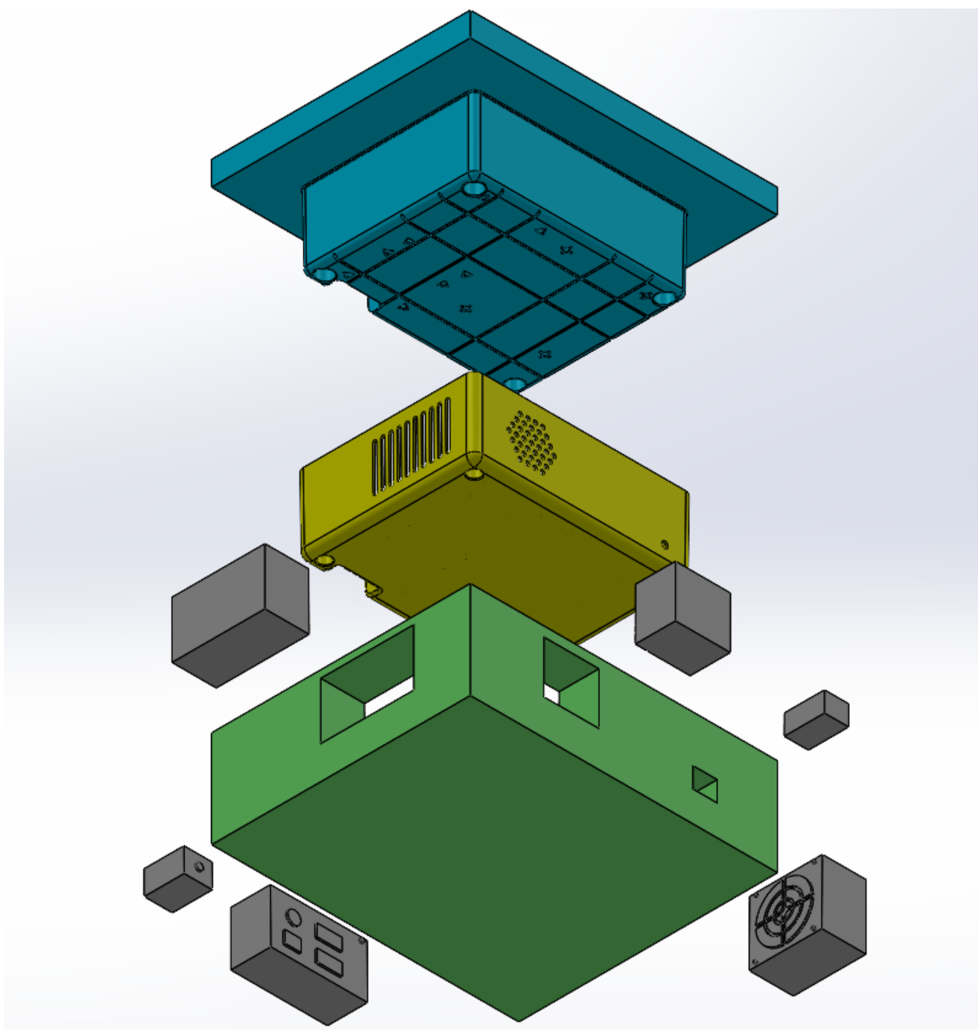


Figure 41: Mold Design Exploded View - 2

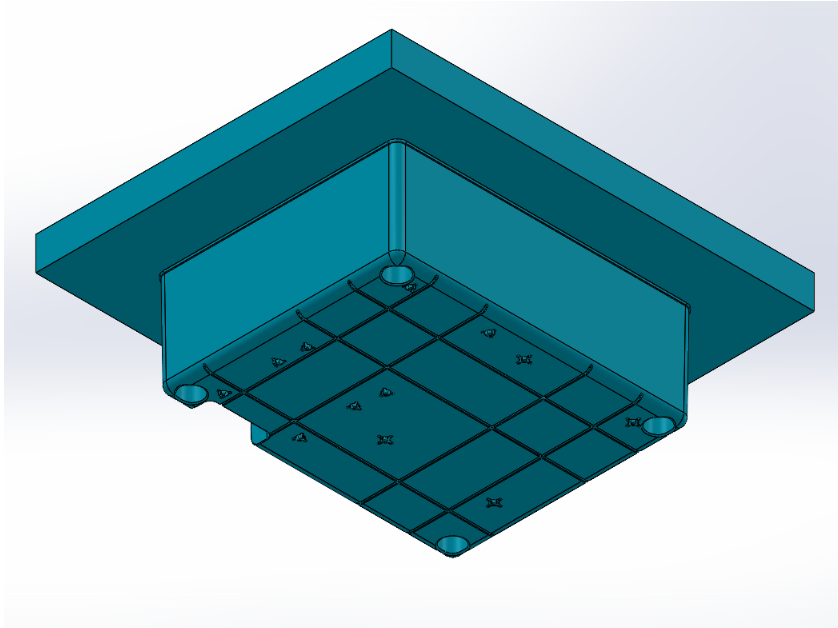
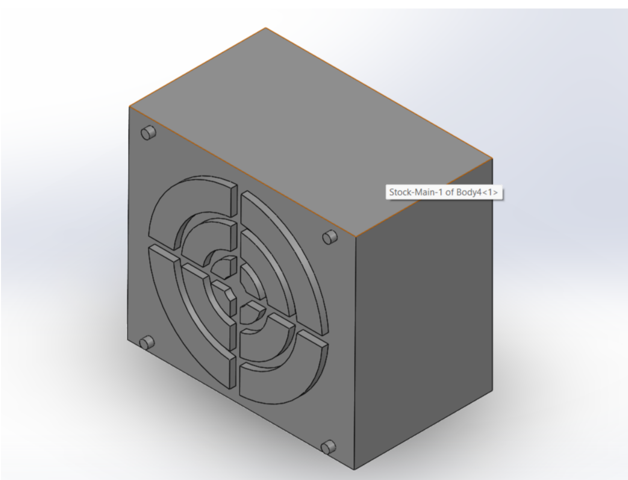
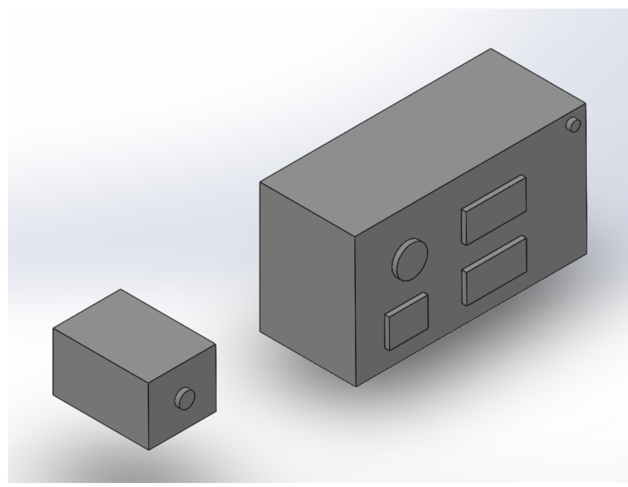


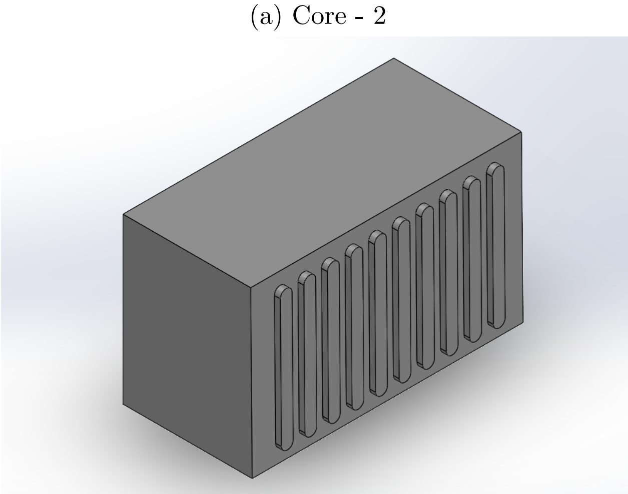
Figure 42: Core - 1



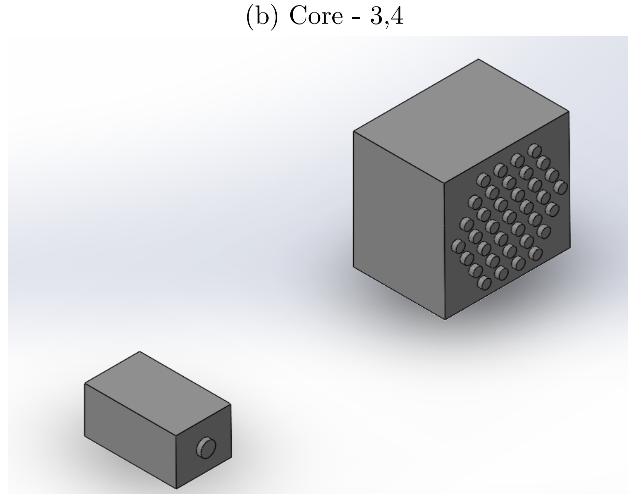
(a) Core - 2



(b) Core - 3,4



(c) Core - 5



(d) Core - 6,7

Figure 43: Cores of Bottom Part Mold

4.3 Top Part

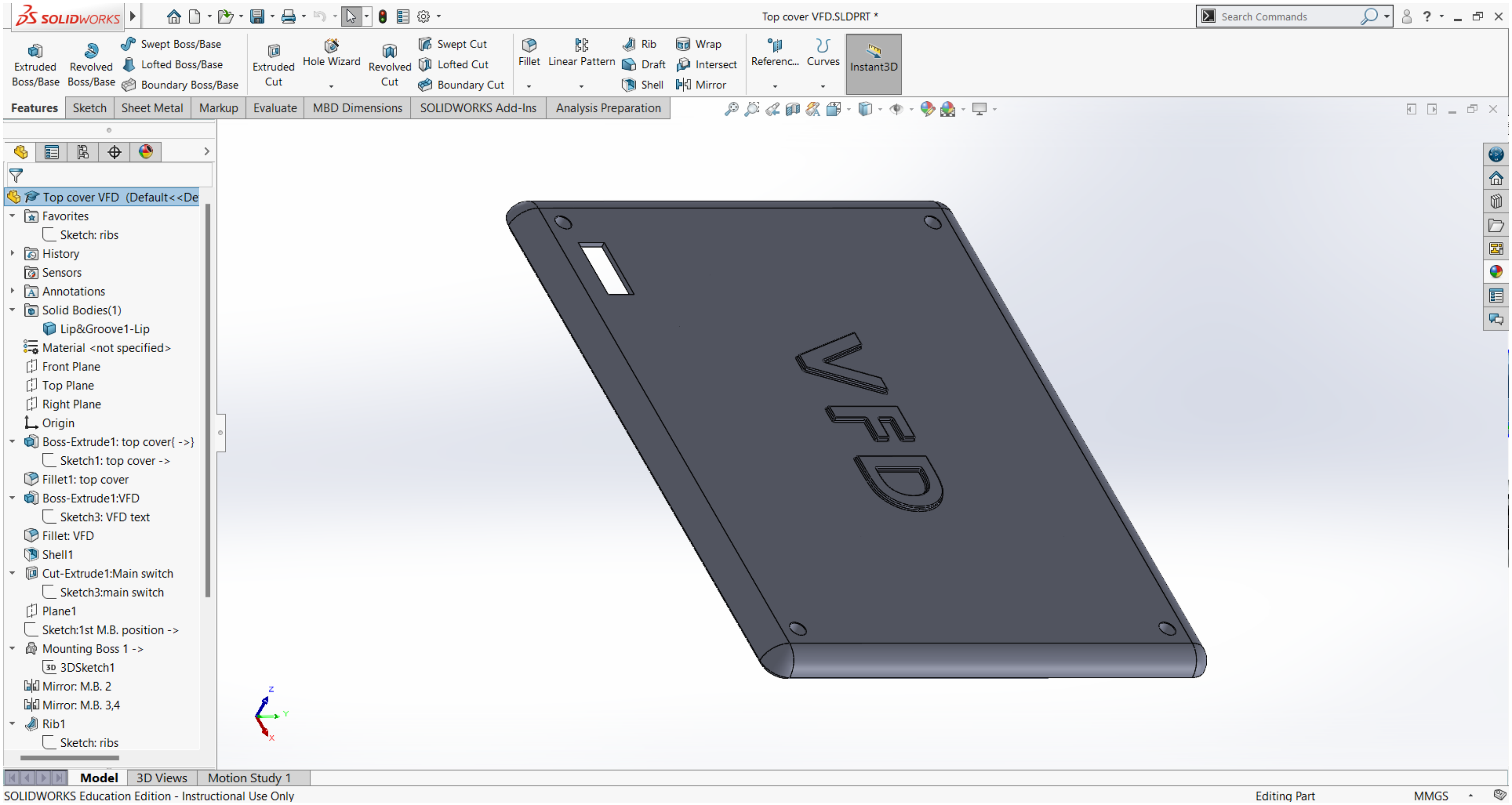


Figure 44: Isometric View

4.3.1 Top Part Mold Design

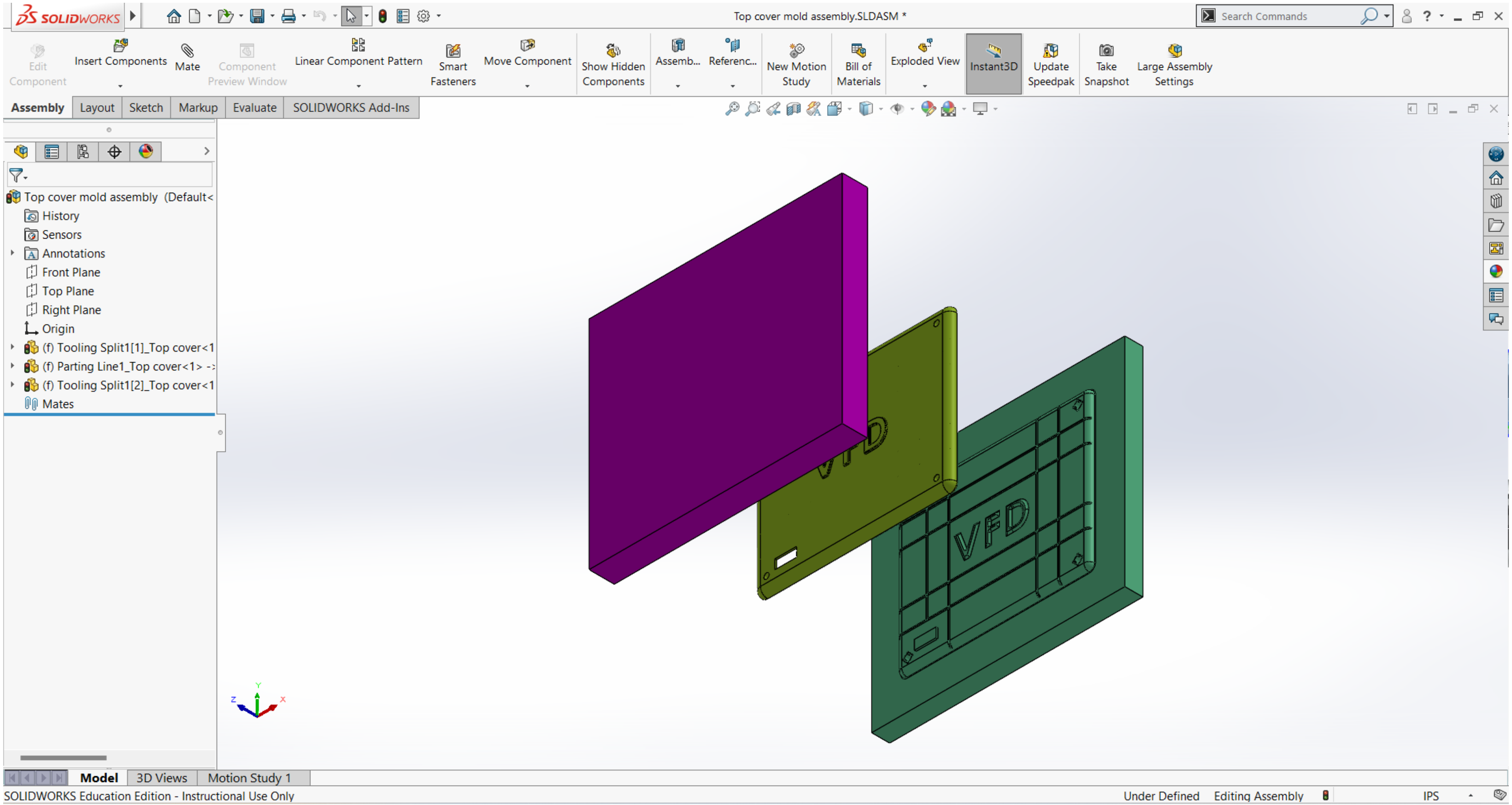


Figure 45: Mold Design Exploded View - 1

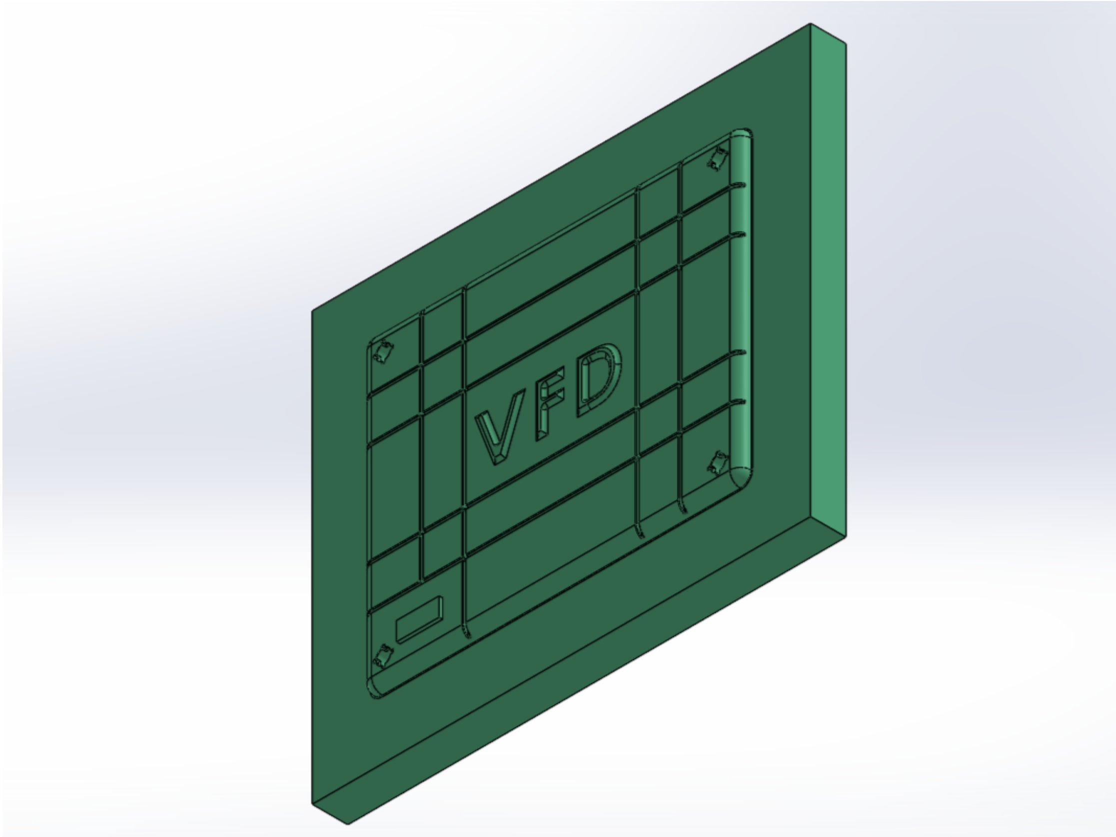


Figure 46: Mold Core

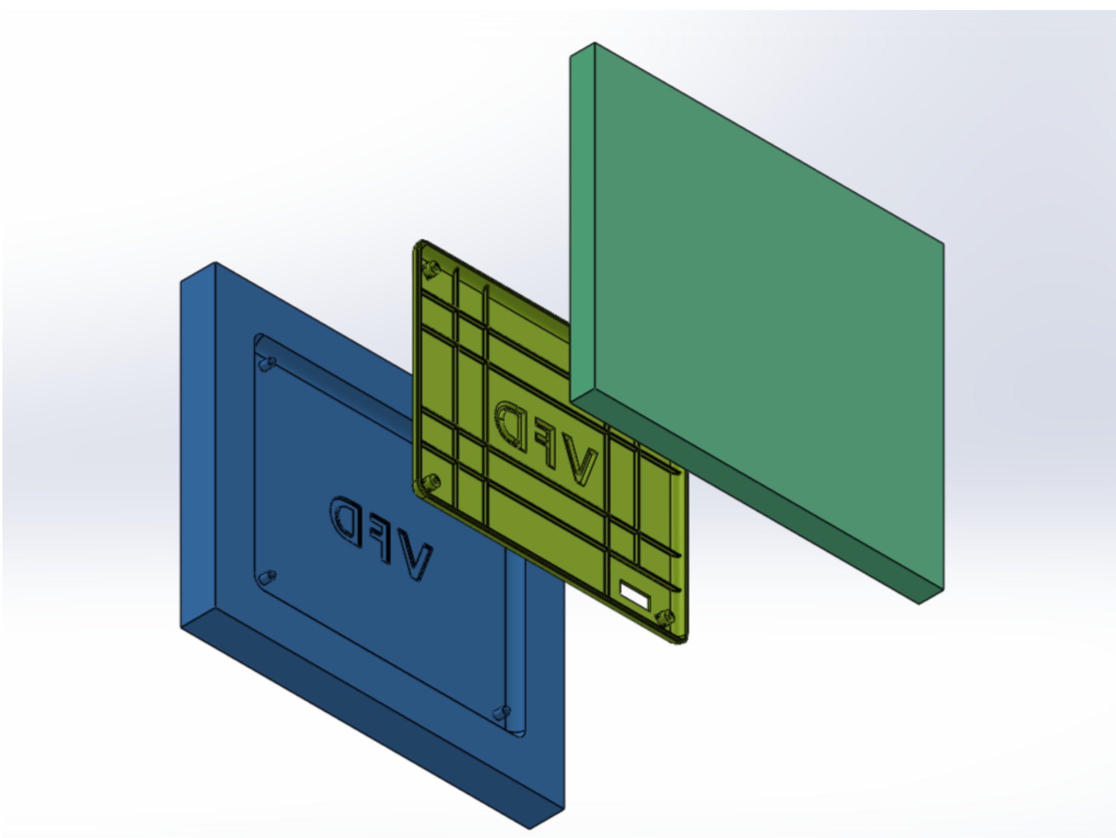


Figure 47: Mold Design Exploded View - 2

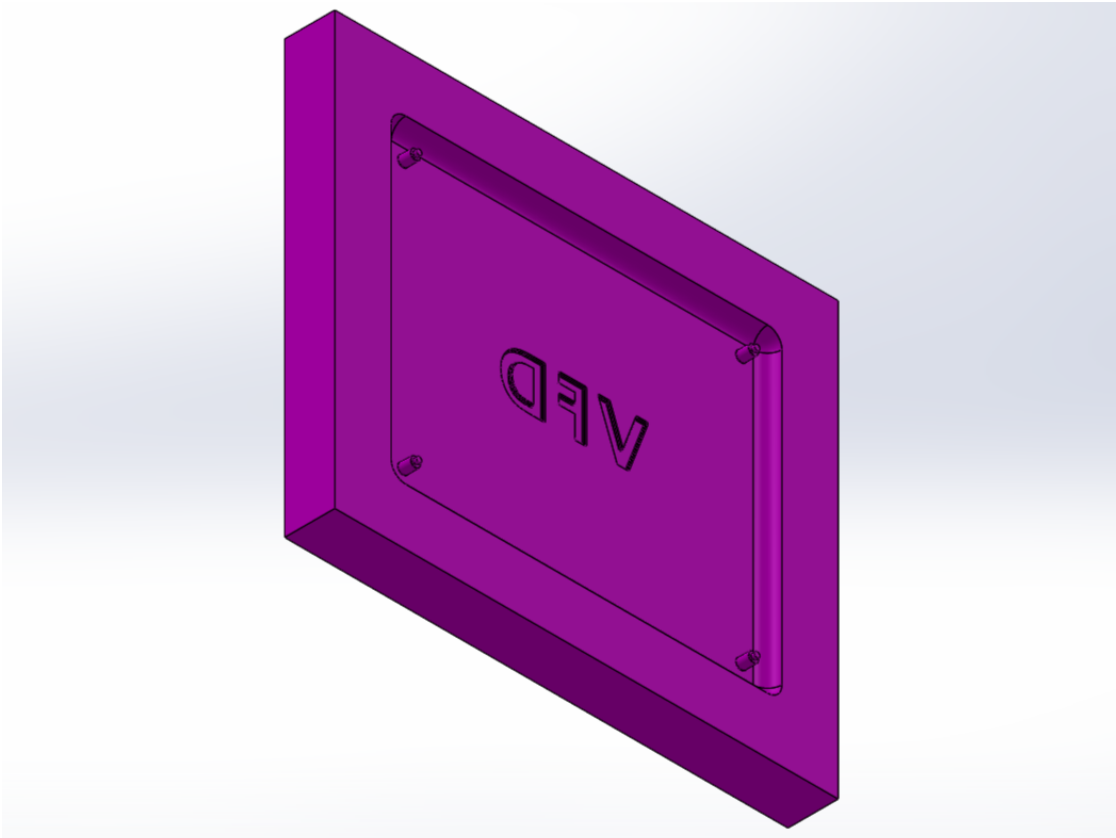


Figure 48: Mold Cavity

4.4 Material and Mass Properties

4.4.1 Mass Properties of The Bottom Part Using ABS Material

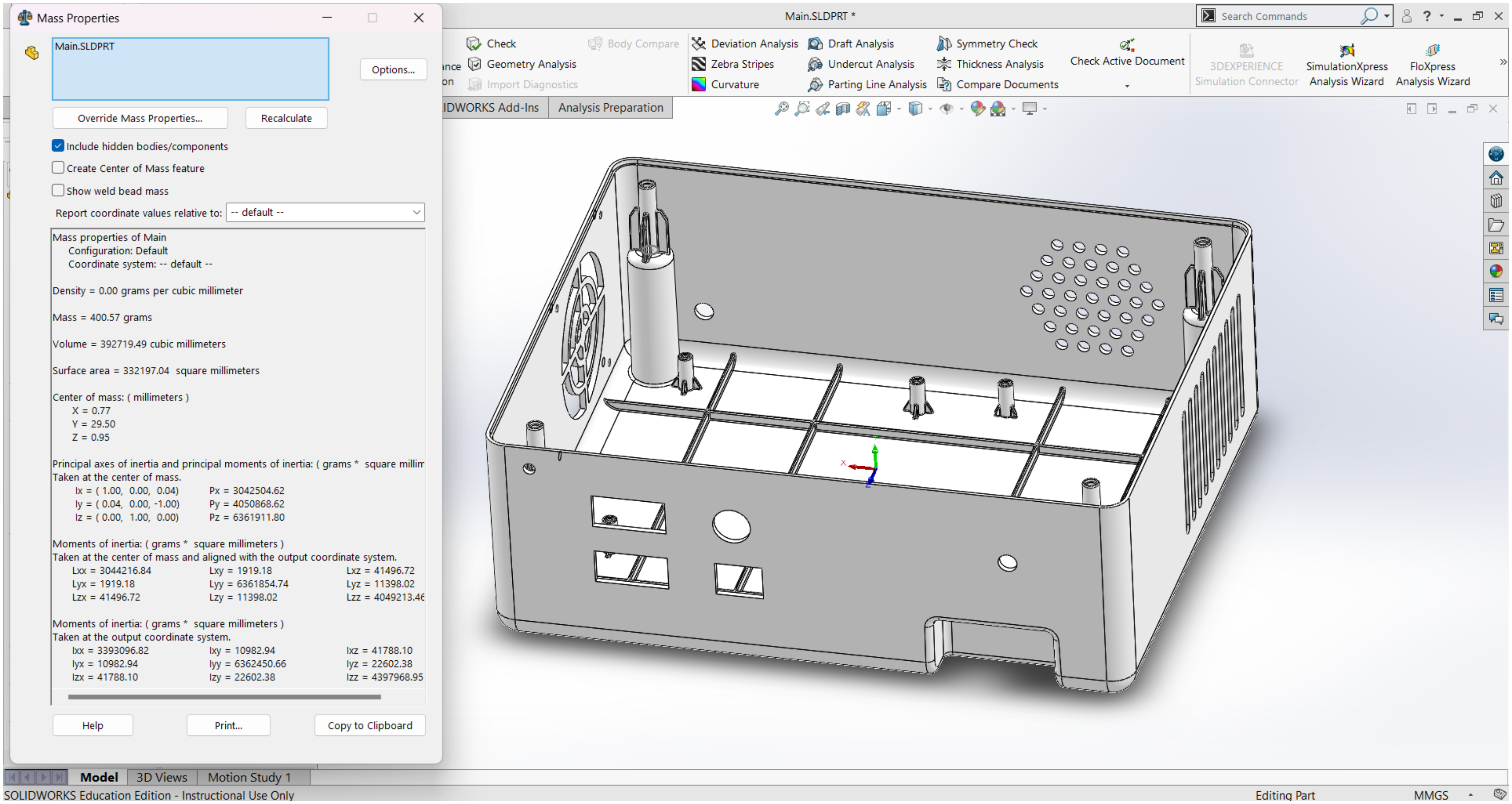


Figure 49: Mass Properties

4.4.2 Mass Properties of The Top Part Using ABS Material

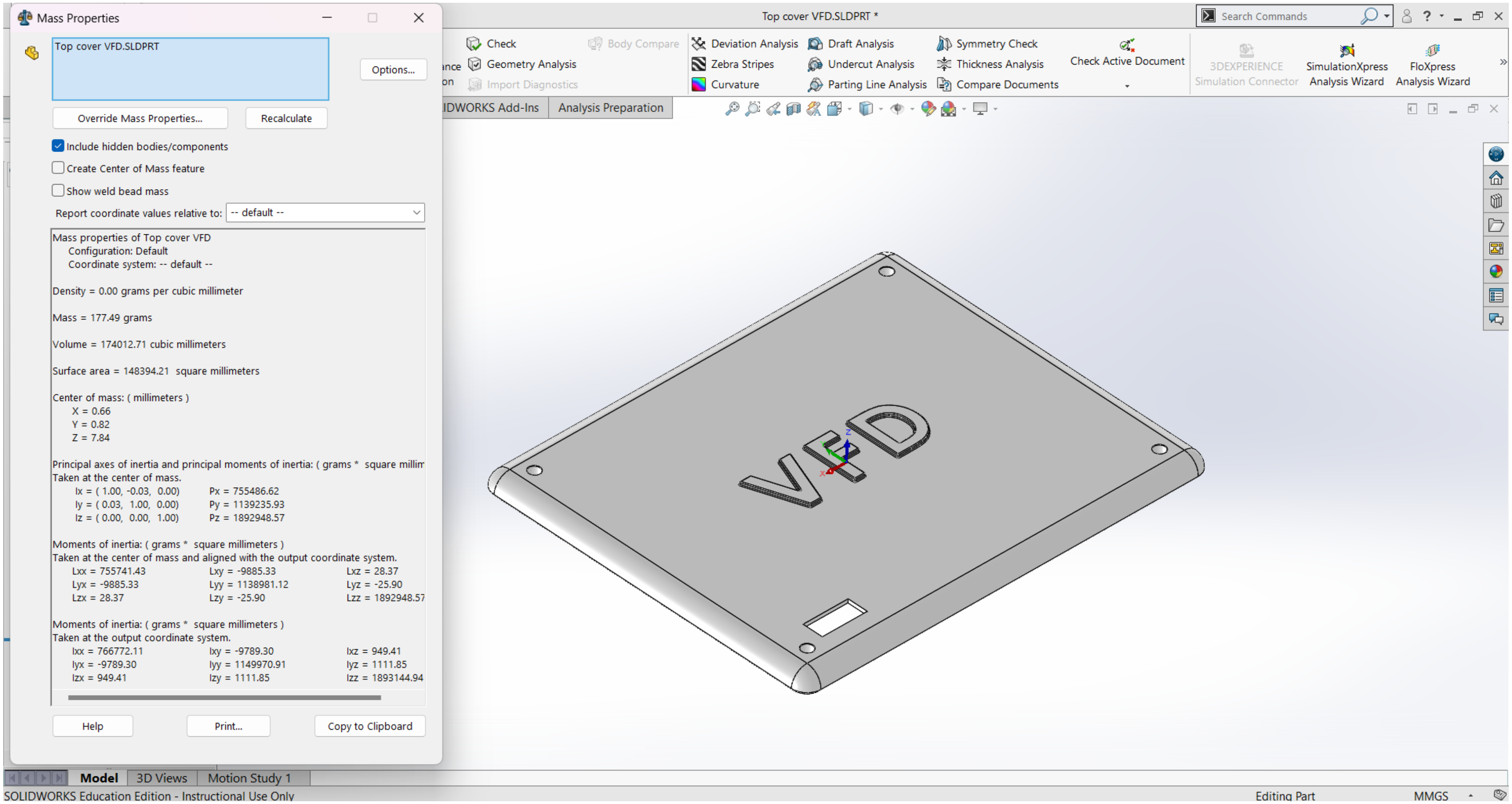
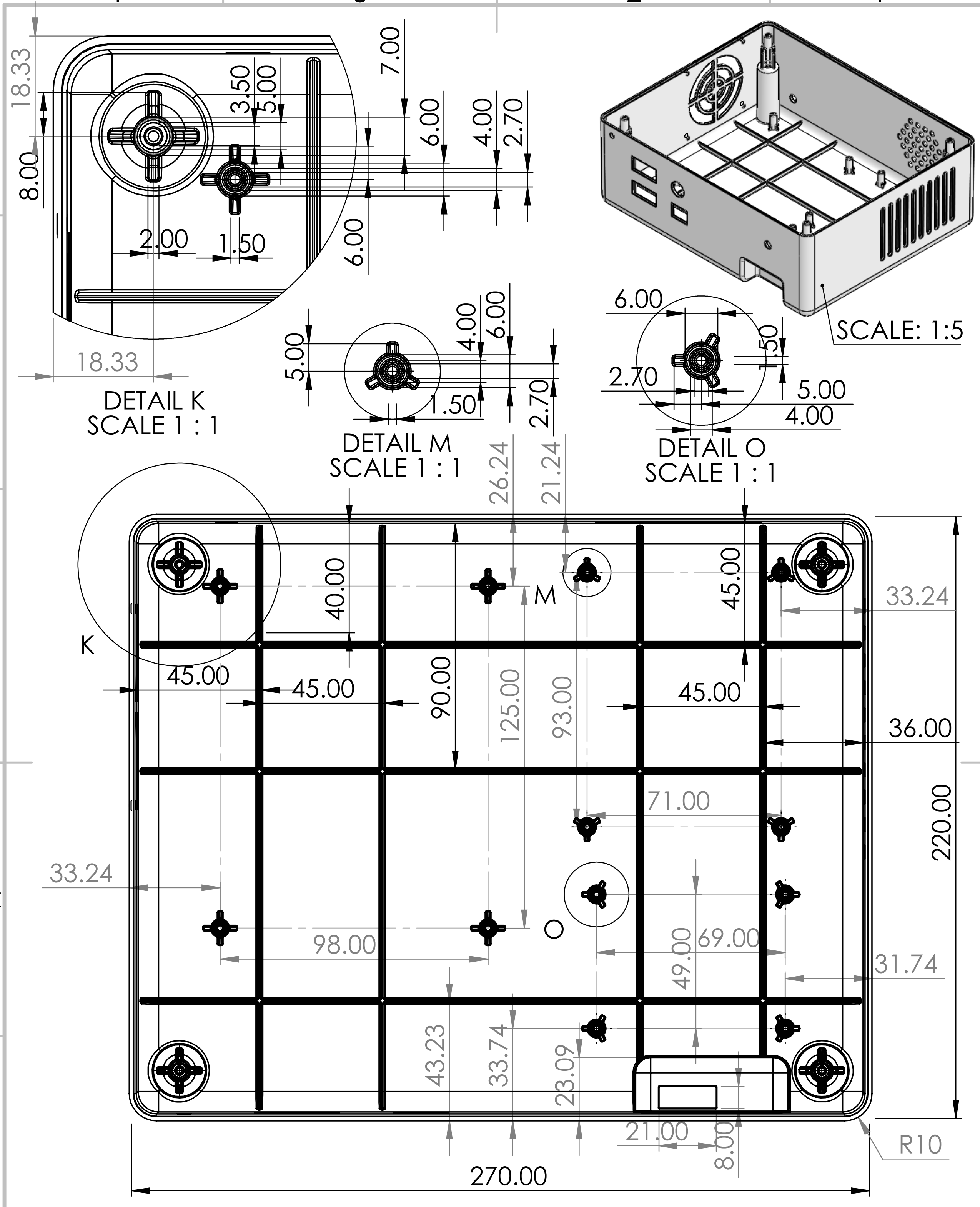


Figure 50: Mass Properties

5 Detailed Design Drawings



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH: 6/8/2024	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING
			REVISION

	NAME	SIGNATURE	DATE
DRAWN	Lahiru Cooray		6/8/2024
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

Bottom Part

MATERIAL: ABS

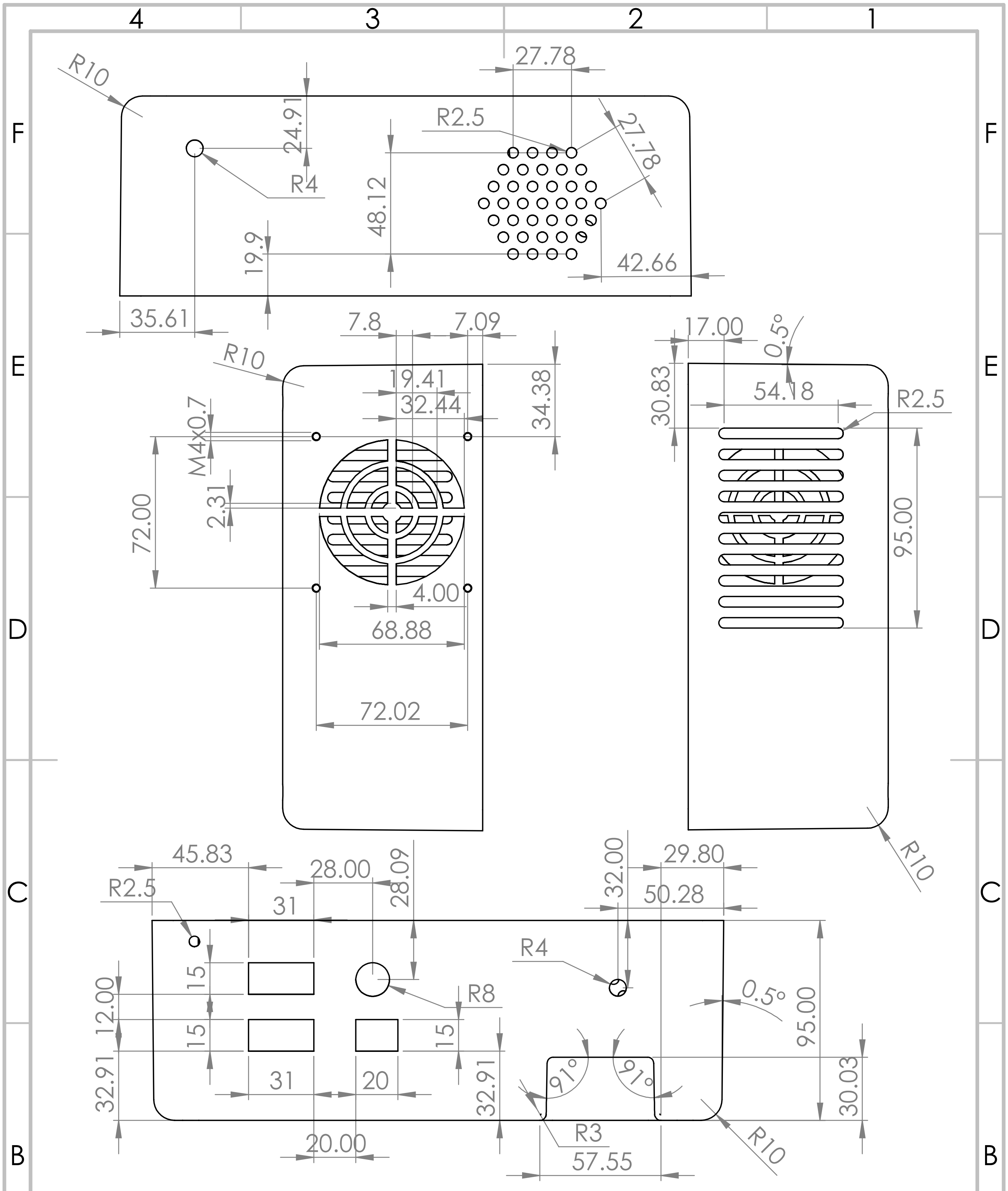
DWG NO. 01

A4

WEIGHT:

SCALE:1:2

SHEET 1 OF 4



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH: 6/8/2024	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING / REVISION

	NAME	SIGNATURE	DATE
DRAWN	Lahiru Cooray		6/8/2024
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE: **Bottom Part**

MATERIAL: ABS

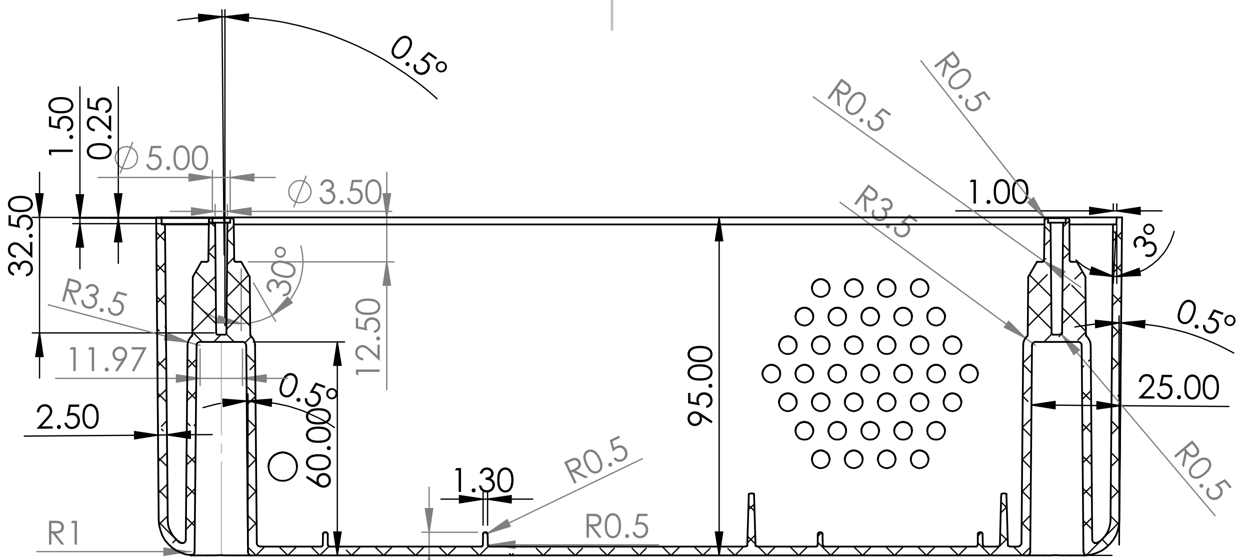
DWG NO. 02

A4

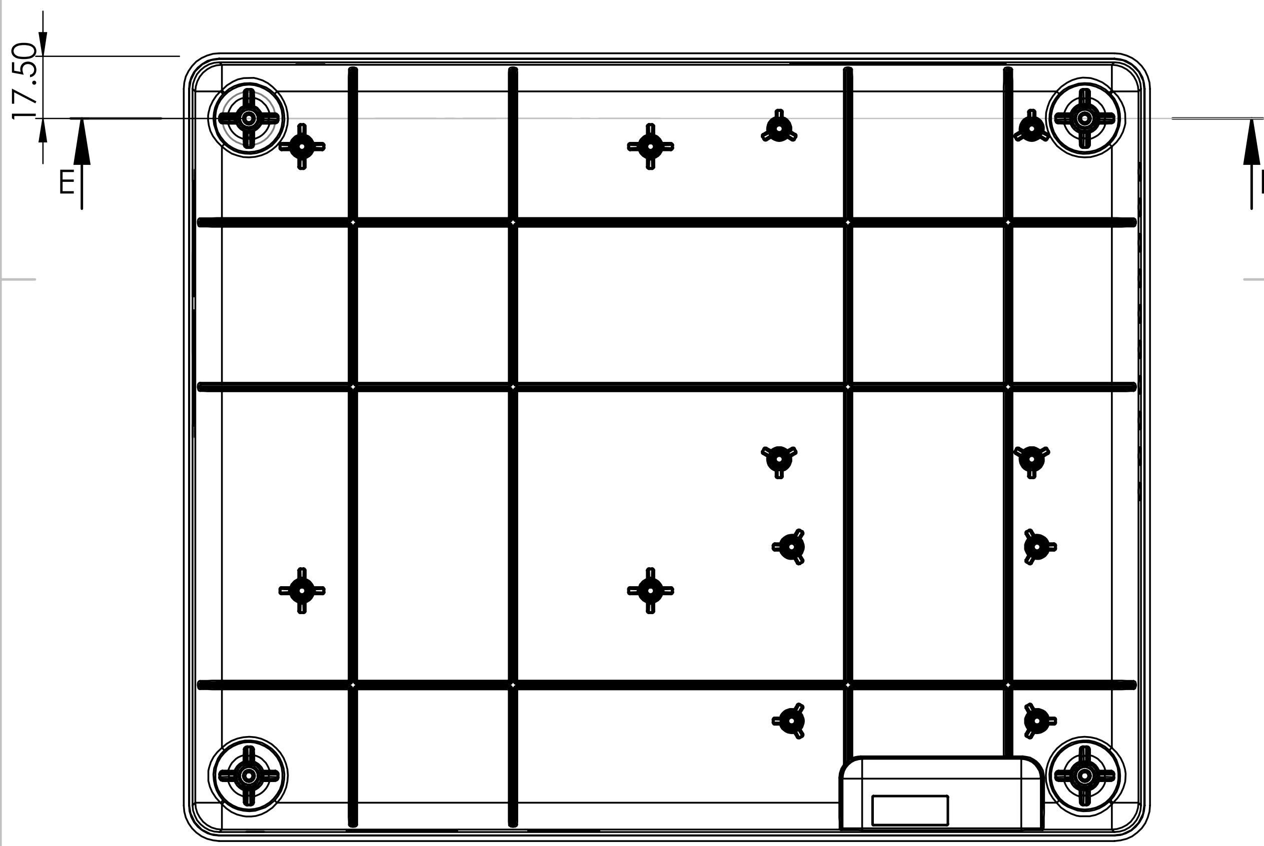
WEIGHT:

SCALE:1:2.5

SHEET 2 OF 4



SECTION: E-E
SCALE 1 : 2



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH: 6/8/2024	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION		

	NAME	SIGNATURE	DATE
DRAWN	Lahiru Cooray		6/8/2024
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

Bottom Part

MATERIAL: ABS

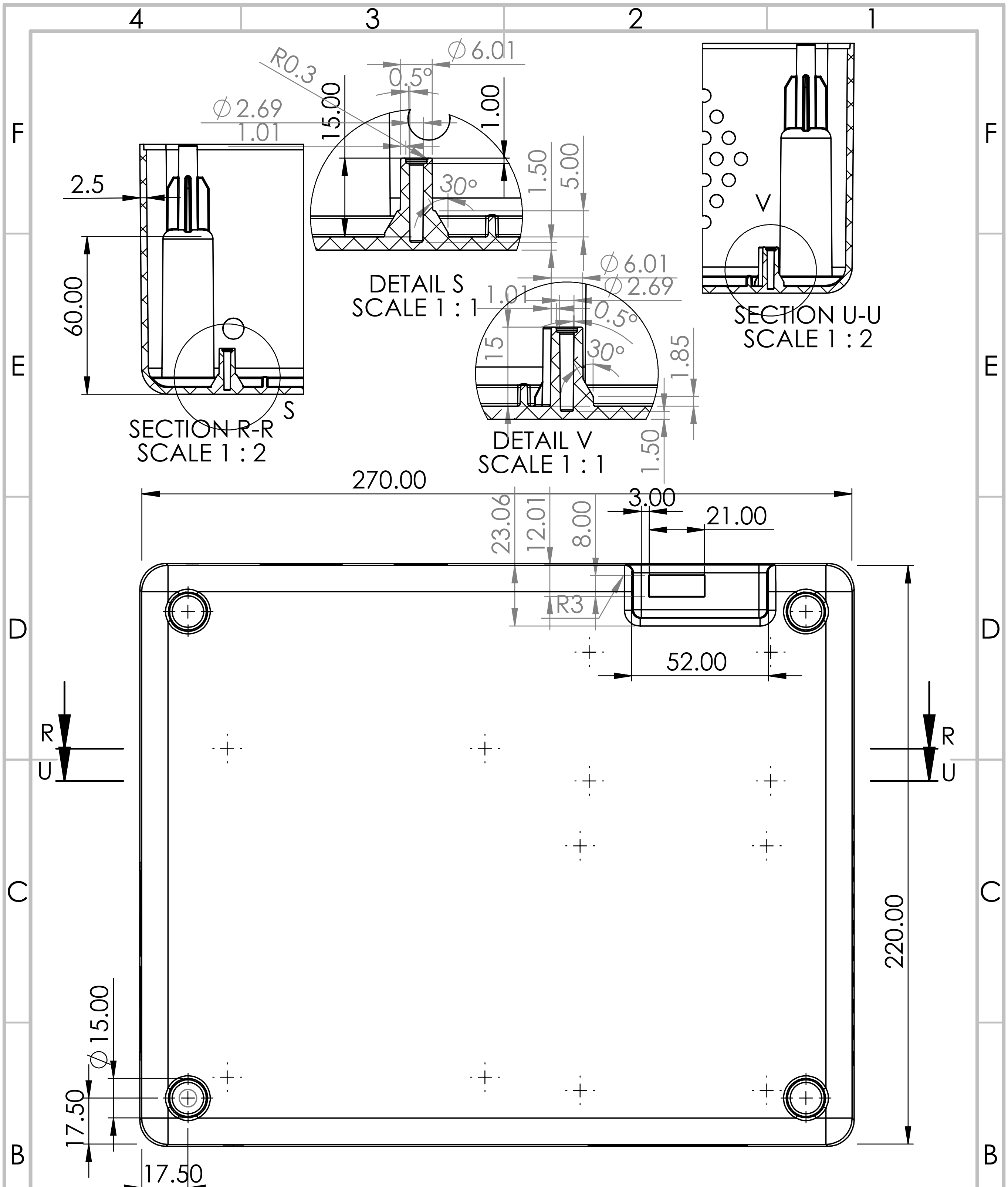
DWG NO. 03

A4

WEIGHT:

SCALE:1:2

SHEET 3 OF 4



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH: 6/9/2024	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN	Lahiru Cooray		6/9/2024
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

Bottom Part

MATERIAL: ABS

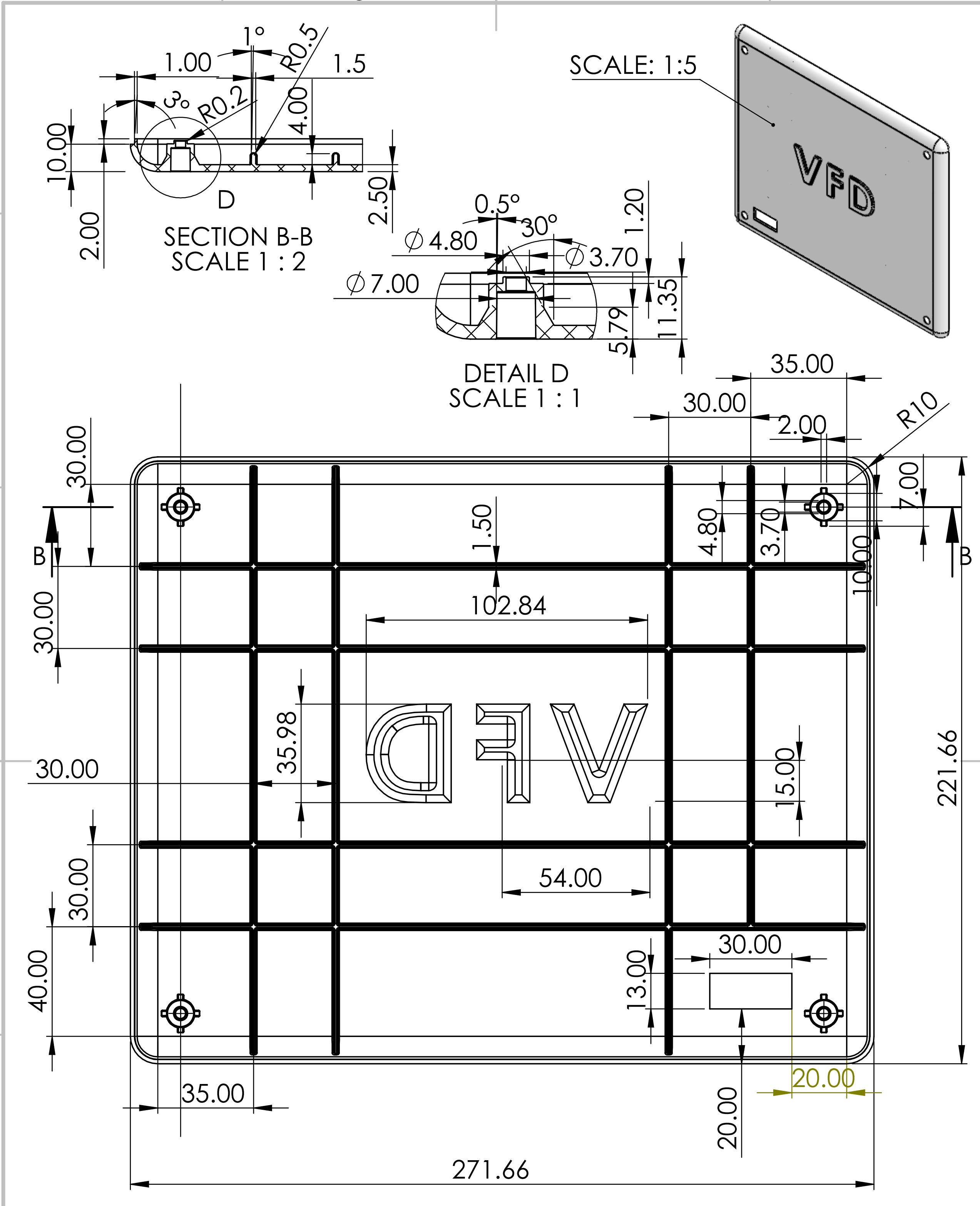
DWG NO. 04

A4

WEIGHT:

SCALE:1:2

SHEET 4 OF 4



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH: 6/9/2024	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING / REVISION

	NAME	SIGNATURE	DATE
DRAWN	Lahiru Cooray		6/9/2024
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE: **Top Part**

MATERIAL: ABS

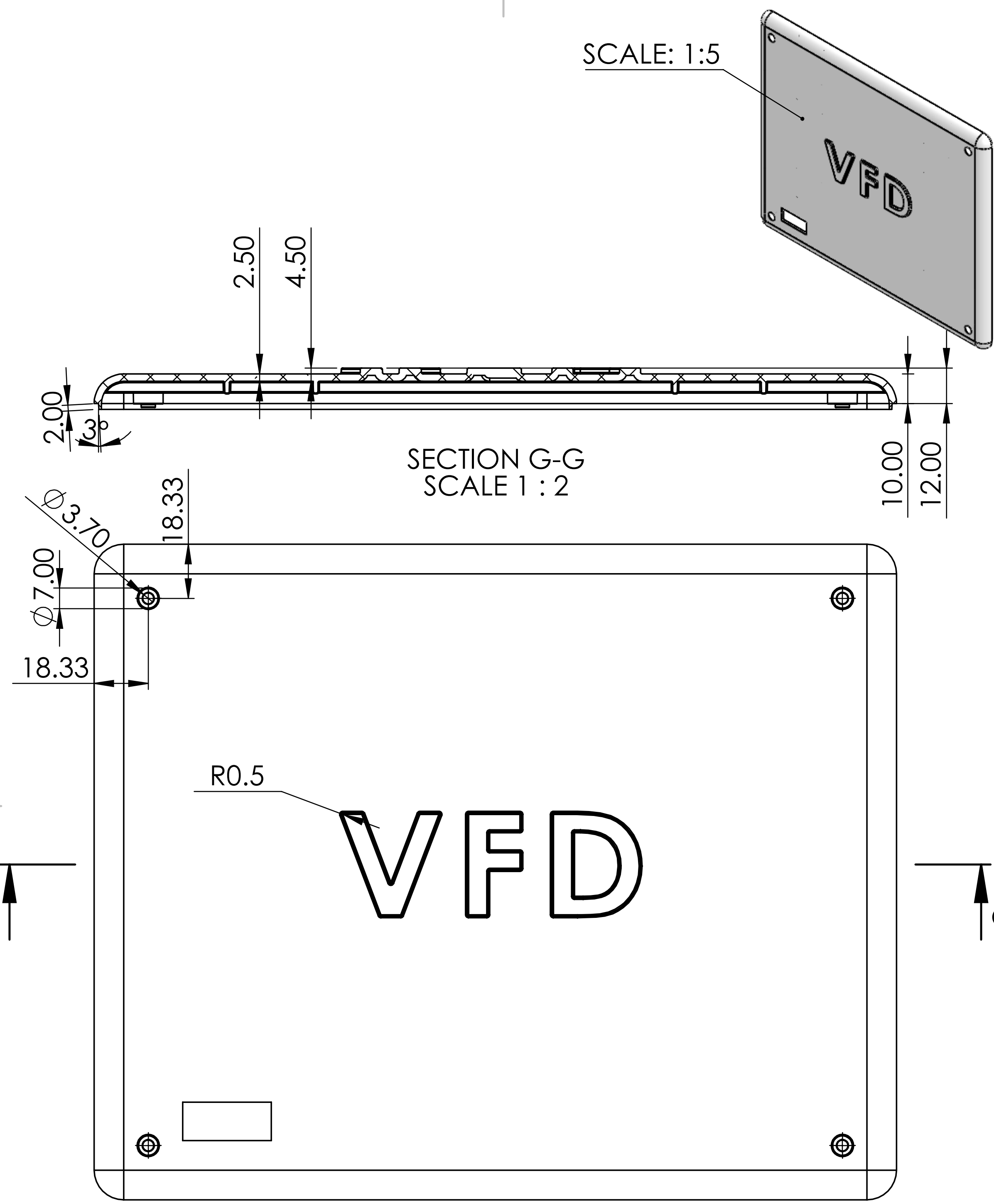
DWG NO. 01

A4

WEIGHT:

SCALE:1:2

SHEET 1 OF 2



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH: 6/9/2024

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN	Lahiru Cooray		6/9/2024
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

Top Part

MATERIAL: ABS

DWG NO. 02

A4

WEIGHT:

SCALE:1:2

SHEET 2 OF 2

6 Photographs of the Physically Built Enclosure

Figure 51: 3D printed Bottom and Top parts

Figure 52: 3D printed Bottom Part

7 Photographs of the Circuit Integration

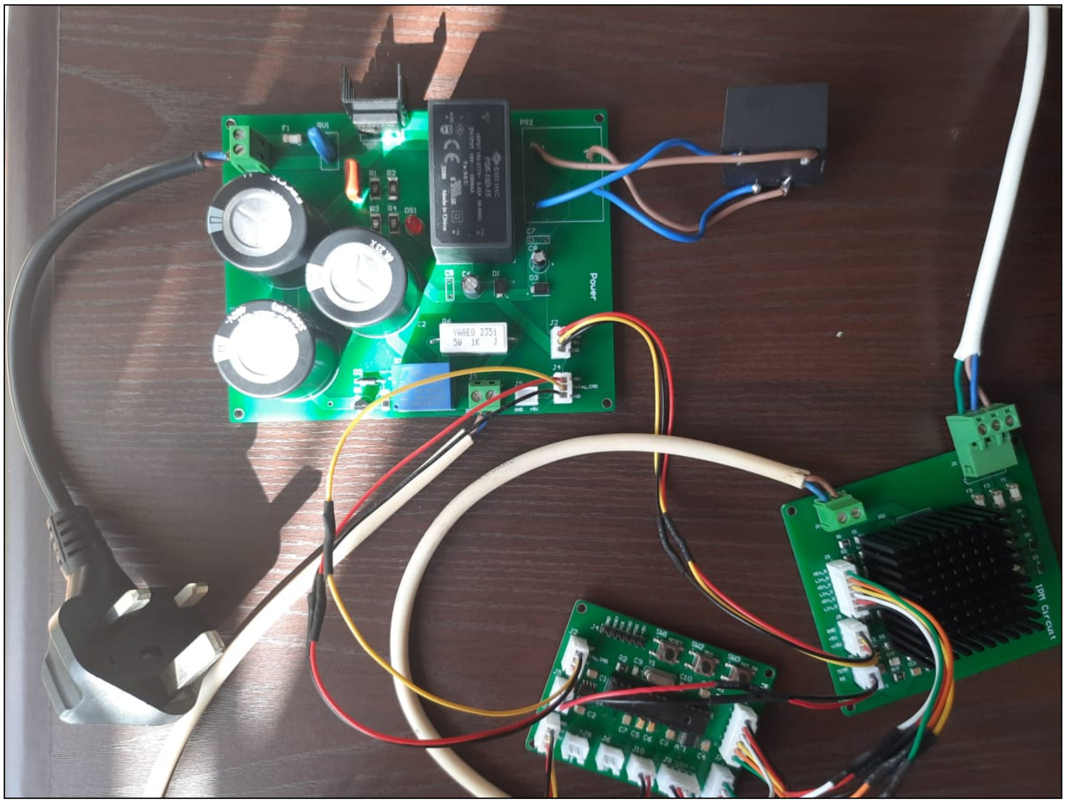


Figure 53: Power Circuit and IPM Circuit

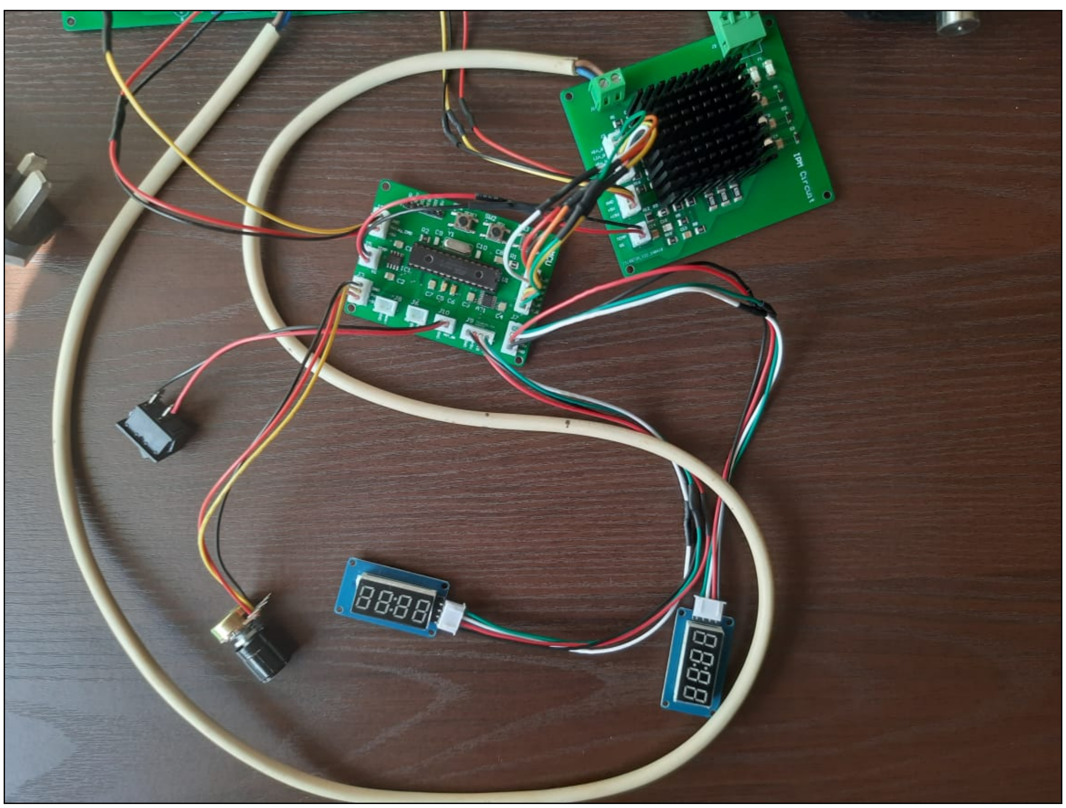


Figure 54: IPM Circuit and MCU Circuit

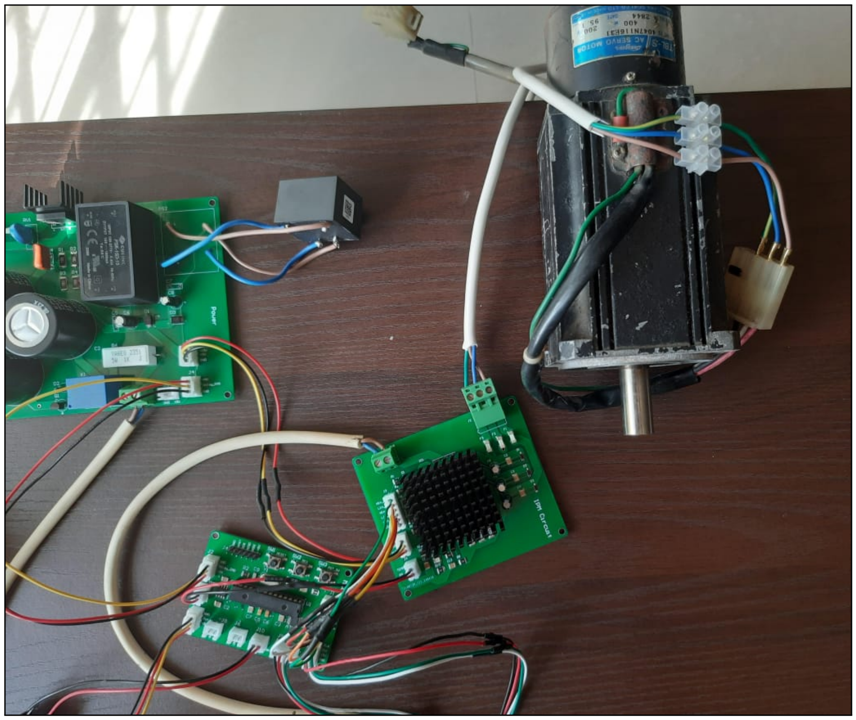


Figure 55: IPM Circuit and the Servo Motor

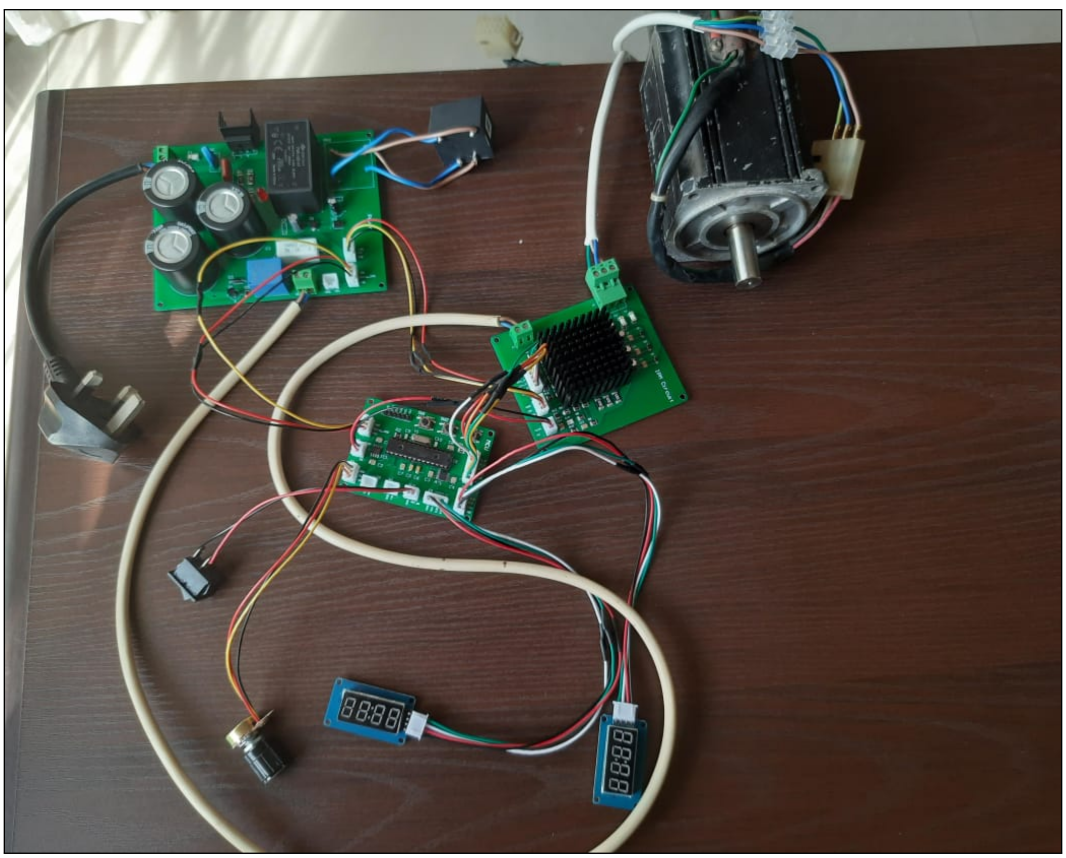


Figure 56: Whole Circuit

8 Photographs of Testing the PCBs

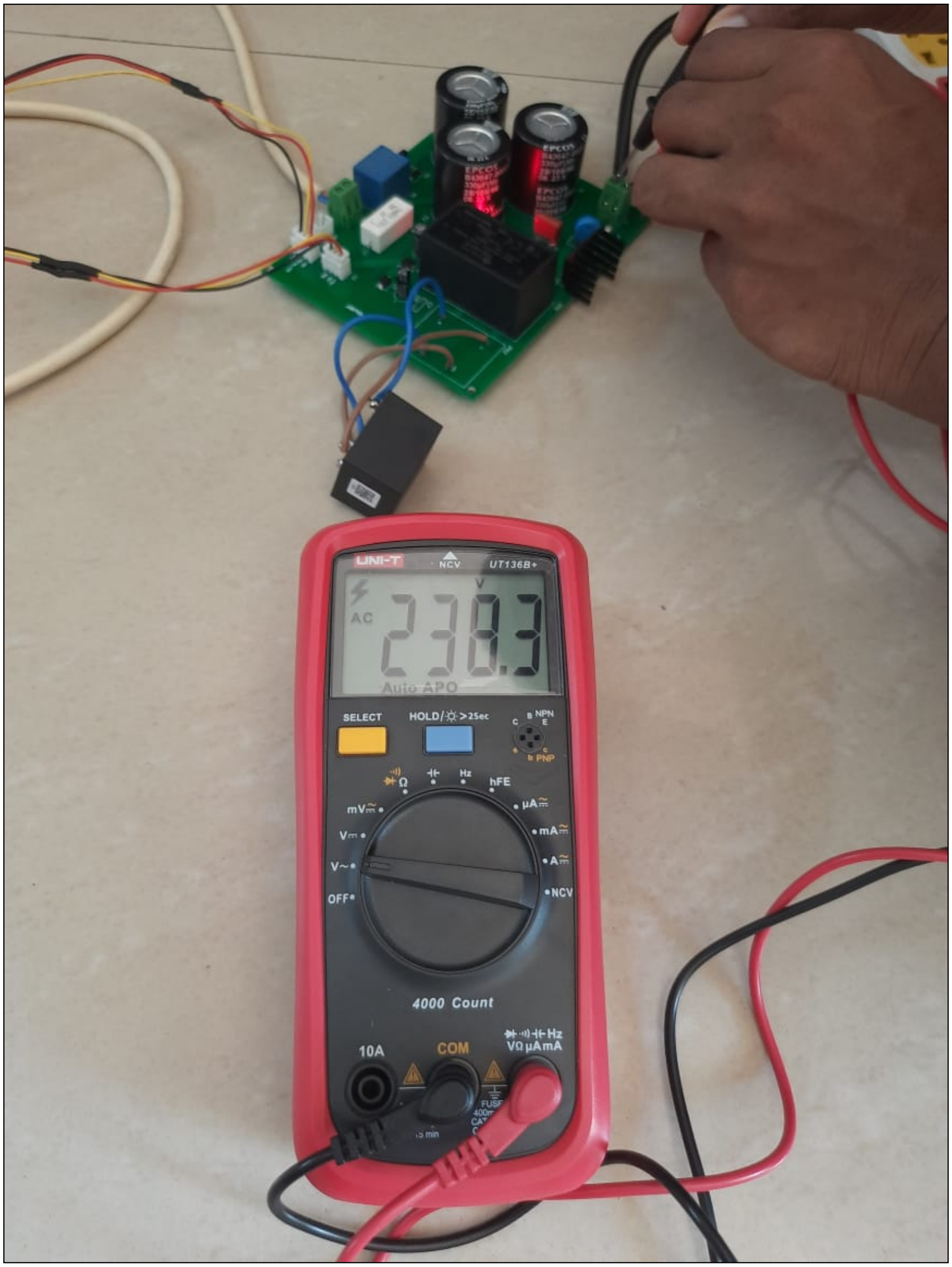


Figure 57: Input Voltage of the VFD 230V AC

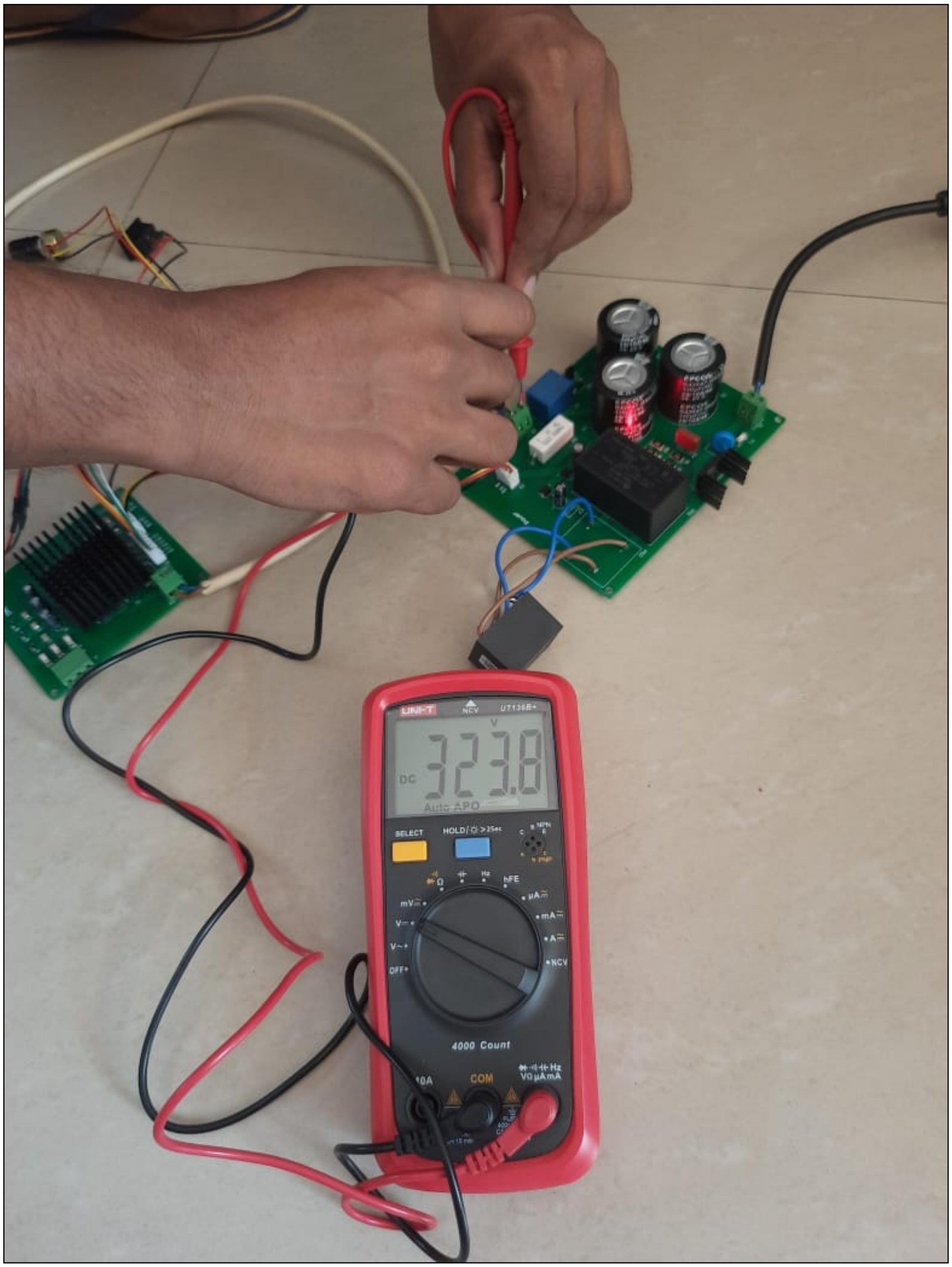


Figure 58: Rectified Voltage 325V DC

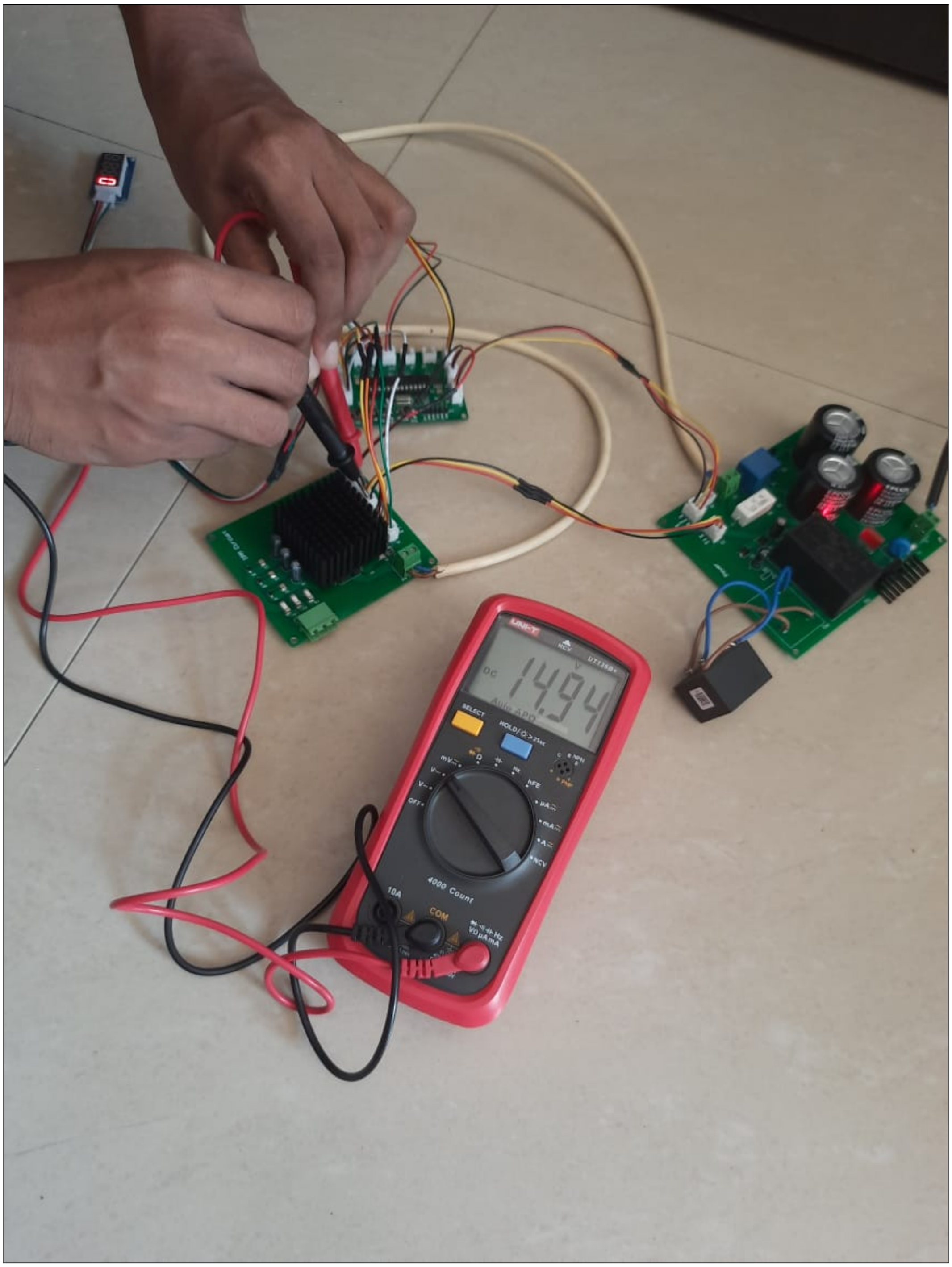


Figure 59: AC-DC Conversion 15V for the IPM

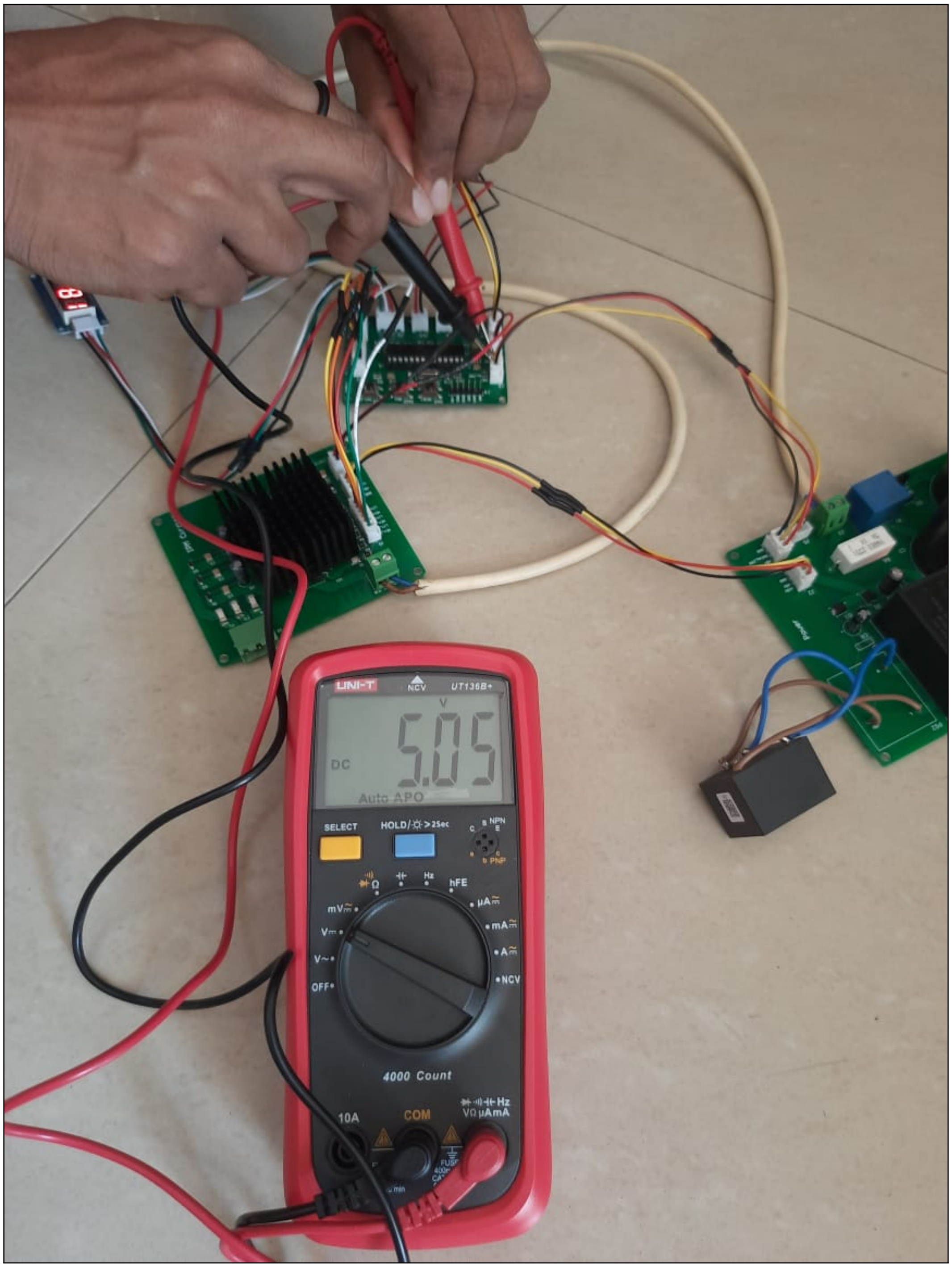


Figure 60: AC-DC Conversion 5V for the MCU

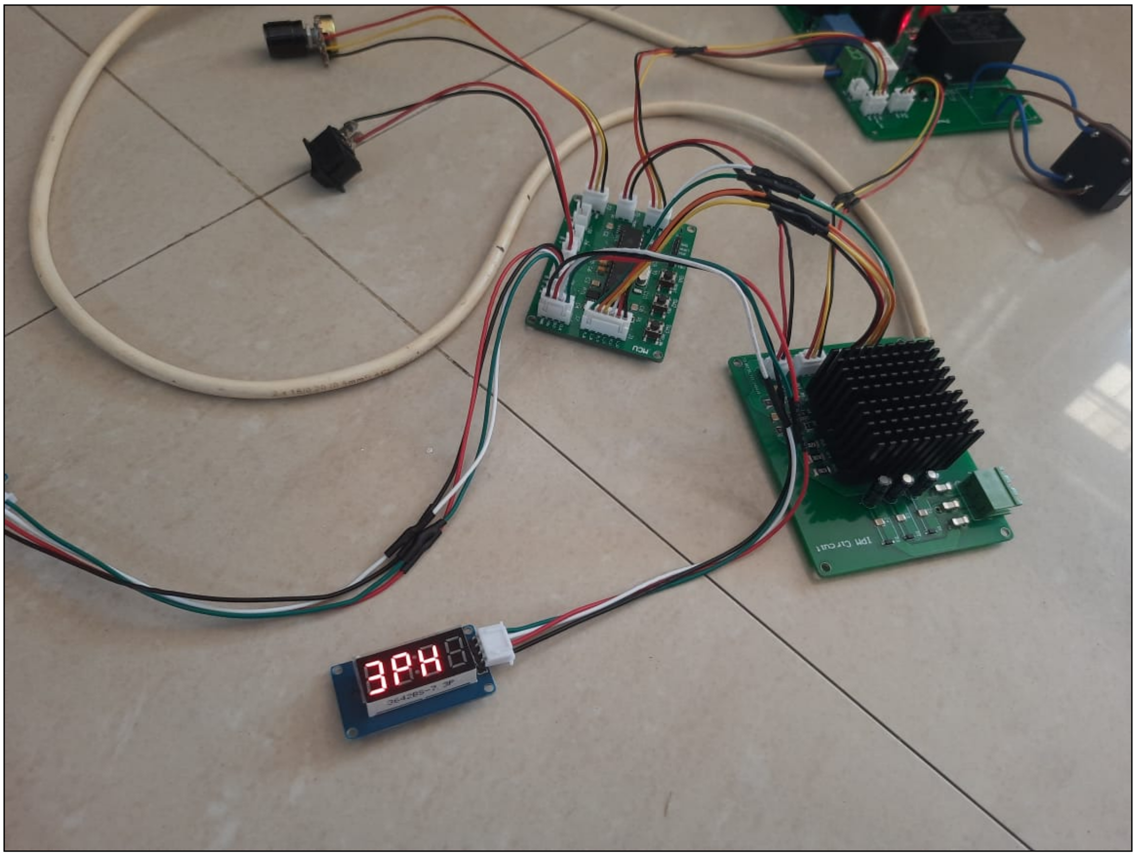


Figure 61: Initialization - Ready to send PWM signals

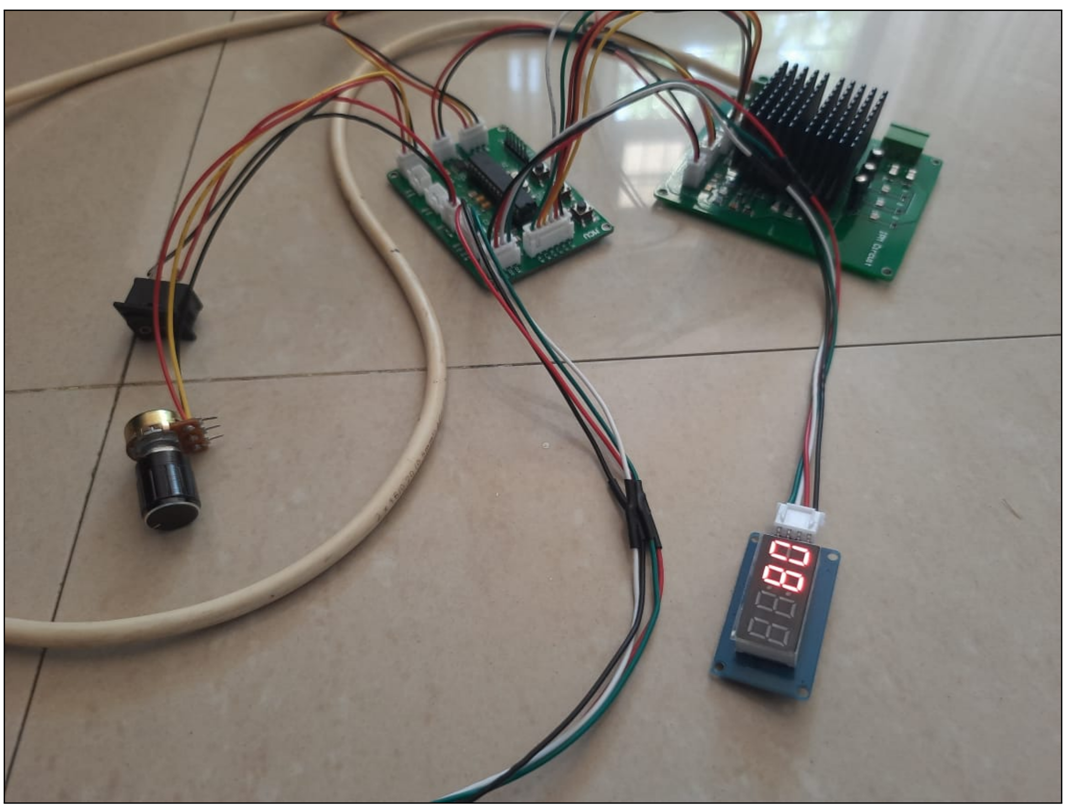


Figure 62: Adjusting the frequency for 80Hz using the potentiometer

9 Detailed Programming Information

9.1 C++ Code for The Micro-controller

```
/* MIT License

Original code by Matan Pazi
Copyright (c) 2023 Matan Pazi

Modifications by Lahiru Cooray
Copyright (c) 2024 Lahiru Cooray

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE. */

#define F_CPU 16000000UL
#include <stdint.h>
#include <avr/interrupt.h>
#include <avr/io.h>
#include <util/delay.h>

#define CLK1 12
#define DIO1 2
#define CLK2 7
#define DIO2 4
#define CURR_INPUT A0
#define POT_INPUT A2
//
#define THREE_PH 0
#define PWM_RUNNING 2
#define PWM_NOT_RUNNING 1
#define PWM_NOT_SET 0
#define DEADTIME_ADD  5
#define DEADTIME_SUB  2
#define SHORT_CLICK     10
```

```
#define LONG_CLICK      500
#define POT_SWITCH_SAMPLES  2
#define BOOT_CAP_CHARGE_TIME   160
#define RELAY_CHARGE_WAIT      2000000
#define DISPLAY_BLINK   100
#define MIN_PWM_VAL   6
#define BUTTON_IS_PRESSED      (!((PINC >> PINC4) & 1))
#define POT_SWITCH_IS_ON       (!((PINC >> PINC3) & 1))


#define SEG_A   0b00000001
#define SEG_B   0b00000010
#define SEG_C   0b00000100
#define SEG_D   0b00001000
#define SEG_E   0b00010000
#define SEG_F   0b00100000
#define SEG_G   0b01000000
#define SEG_DP  0b10000000

#define TM1637_I2C_COMM1    0x40
#define TM1637_I2C_COMM2    0xC0
#define TM1637_I2C_COMM3    0x80

#define DEFAULT_BIT_DELAY  100

const uint8_t THREE_PHASE[] = {
    SEG_A | SEG_B | SEG_C | SEG_D | SEG_G,
    SEG_A | SEG_B | SEG_E | SEG_F | SEG_G,
    SEG_B | SEG_C | SEG_E | SEG_F | SEG_G,
    0,
    };
const uint8_t SPACE[] = {
    0,
    };

const float Sine[] = {125.0,179.0,223.0,246.0,246.0,223.0,179.0,125.0,71.0,27.0,6.0,6.0
,27.0,71.0,125.0};
const uint8_t Sine_Len = 15;
const uint8_t Min_Freq = 20;
const uint8_t Max_Freq = 120;
const uint16_t Base_Freq = 1041;
const float Min_Amp = 0.1;
const float Max_Amp = 1.0;
const float V_f = 1.66667;
const float VBus = 230.0;
bool  Phase_Config = 0;
bool  Config_Editable = 0;
int16_t Sine_Used[] = {125,179,223,246,246,223,179,125,71,27,6,6,27,71,125};
uint8_t  Click_Type = 0;
uint8_t  PWM_Running = PWM_NOT_SET;
uint32_t Timer = 0;
```

```
uint32_t Click_Timer = 0;
uint32_t Pot_Switch_Off_Timer = 0;
uint32_t Pot_Switch_On_Timer = 0;
uint32_t Display_Timer = 0;
uint32_t Timer_Temp = 0;
uint32_t Timer_Temp1 = 0;
uint32_t Init_PWM_Counter = 0;
uint16_t Curr_Value = 0;
uint8_t Sine_Index = 0;
uint8_t Sine_Index_120 = Sine_Len / 3;
uint8_t Sine_Index_240 = (Sine_Len * 2) / 3;
uint8_t OVF_Counter = 0;
uint8_t OVF_Counter_Compare = 0;

const uint8_t digitToSegment[] = {
  0b00111111,    // 0
  0b00000110,    // 1
  0b01011011,    // 2
  0b01001111,    // 3
  0b01100110,    // 4
  0b01101101,    // 5
  0b01111101,    // 6
  0b00000111,    // 7
  0b01111111,    // 8
  0b01101111,    // 9
  0b01110111,    // A
  0b01111100,    // b
  0b00111001,    // C
  0b01011110,    // d
  0b01111001,    // E
  0b01110001     // F
  };

static const uint8_t minusSegments = 0b01000000;


class TM1637Display {
private:
    uint8_t m_pinClk;
    uint8_t m_pinDIO;
    uint8_t m_brightness;
    unsigned int m_bitDelay;

    void bitDelay();
    void start();
    void stop();
    bool writeByte(uint8_t b);
    void showDots(uint8_t dots, uint8_t* digits);
    uint8_t encodeDigit(uint8_t digit);

public:
```

```
    TM1637Display(uint8_t pinClk, uint8_t pinDIO, unsigned int bitDelay =
       DEFAULT_BIT_DELAY);
    void setBrightness(uint8_t brightness, bool on = true);
    void setSegments(const uint8_t segments[], uint8_t length = 4, uint8_t
       pos = 0);
    void clear();
    void showNumberDec(int num, bool leading_zero = false, uint8_t length =
       4, uint8_t pos = 0);
    void showNumberDecEx(int num, uint8_t dots, bool leading_zero, uint8_t
       length, uint8_t pos);
    void showNumberBaseEx(int8_t base, uint16_t num, uint8_t dots = 0, bool
       leading_zero = false, uint8_t length = 4, uint8_t pos = 0);
};

TM1637Display Display1(CLK1, DIO1);
TM1637Display Display2(CLK2, DIO2);

void Button_Click();
void Display(uint8_t PWM_Running, bool Blink, uint8_t Desired_Freq);
void Pot_Switch_State_Check();
void Reverse_3_Phase();
void Wait_A_Bit(uint32_t Executions_To_Wait);
void Pwm_Disable();
void Pwm_Config();
void ADC_init();
uint16_t analogRead(uint8_t channel) ;


void setup()
{
  cli();
  CLKPR = (1 << CLKPCE);
  CLKPR = (1 << CLKPS0);
  sei();
  Display1.setBrightness(0x02);
  Display2.setBrightness(0x02);
  Display1.clear();
  Display2.clear();
  PORTC = (1 << PORTC3) | (1 << PORTC4);
  DDRD = (1 << DDD6) | (1 << DDD5) | (1 << DDD3);
  DDRB = (1 << DDB3) | (1 << DDB2) | (1 << DDB1) | (1 << DDB0);

  Wait_A_Bit(RELAY_CHARGE_WAIT);

  PORTB = (1 << PORTB0);
}


void loop()
{
   uint8_t Desired_Freq;
   uint8_t OVF_Counter_Compare_Temp;
```

```
    float Amp;
    ADC_init();
    Curr_Value = analogRead(0) << 3;
    Desired_Freq = ((uint8_t)(analogRead(2) >> 3));
    if (Desired_Freq < Min_Freq) Desired_Freq = Min_Freq;
    else if (Desired_Freq > Max_Freq) Desired_Freq = Max_Freq;
    OVF_Counter_Compare_Temp = (uint8_t)(Base_Freq / Desired_Freq);
    {
       OVF_Counter_Compare = OVF_Counter_Compare_Temp;

    }
    Amp = ((float)(Desired_Freq) * V_f) / VBus;
    if (Amp < Min_Amp) Amp = Min_Amp;
    else if (Amp > Max_Amp) Amp = Max_Amp;
    {
       for (int i = 0; i < Sine_Len; i++)
       {
          Sine_Used[i] = (int16_t)(Amp * Sine[i]);
       }
    }

    Pot_Switch_State_Check();
    if (BUTTON_IS_PRESSED) Button_Click();
    if (PWM_Running != PWM_NOT_SET) Display(PWM_Running, Config_Editable,
       Desired_Freq);
    Timer++;
}


int main()
{
     setup();
     while (1)
     {
         loop();
     }
     return 0;
}


void Button_Click()
{
  if (BUTTON_IS_PRESSED)
  {
    if (Timer - Timer_Temp1  > 1) Click_Timer = 0;
    else Click_Timer++;
    Timer_Temp1 = Timer;
    if (Click_Timer > SHORT_CLICK)
    {
      Reverse_3_Phase();
    }
    Click_Timer = 0;
```

```
  }

  else Click_Timer = 0;
}


TM1637Display::TM1637Display(uint8_t pinClk, uint8_t pinDIO, unsigned int
   bitDelay)
    : m_pinClk(pinClk), m_pinDIO(pinDIO), m_bitDelay(bitDelay)
{
    DDRB &= ~(1 << m_pinClk);
    DDRB &= ~(1 << m_pinDIO);
    PORTB &= ~(1 << m_pinClk);
    PORTB &= ~(1 << m_pinDIO);
}

void TM1637Display::bitDelay() {
    for (uint16_t i = 0; i < m_bitDelay; ++i) {
        _delay_us(1);
    }
}

void TM1637Display::start() {
    DDRB |= (1 << m_pinDIO);
    bitDelay();
}

void TM1637Display::stop() {
    DDRB |= (1 << m_pinDIO);
    bitDelay();
    DDRB &= ~(1 << m_pinClk);
    bitDelay();
    DDRB &= ~(1 << m_pinDIO);
    bitDelay();
}

bool TM1637Display::writeByte(uint8_t b) {
    uint8_t data = b;
    for (uint8_t i = 0; i < 8; i++) {
        DDRB |= (1 << m_pinClk);
        bitDelay();
        if (data & 0x01) {
            DDRB &= ~(1 << m_pinDIO);
        } else {
            DDRB |= (1 << m_pinDIO);
        }
        bitDelay();
        DDRB &= ~(1 << m_pinClk);
        bitDelay();
        data = data >> 1;
    }
    DDRB |= (1 << m_pinClk);
```

```
    DDRB &= ~(1 << m_pinDIO);
    bitDelay();
    DDRB &= ~(1 << m_pinClk);
    bitDelay();
    uint8_t ack = (PINB & (1 << m_pinDIO)) != 0;
    if (ack == 0) {
        DDRB |= (1 << m_pinDIO);
    }
    bitDelay();
    DDRB |= (1 << m_pinClk);
    bitDelay();
    return ack;
}

void TM1637Display::showDots(uint8_t dots, uint8_t* digits) {
    for (int i = 0; i < 4; ++i) {
        digits[i] |= (dots & 0x80);
        dots <<= 1;
    }
}

uint8_t TM1637Display::encodeDigit(uint8_t digit) {
    return digitToSegment[digit & 0x0f];
}

void TM1637Display::setBrightness(uint8_t brightness, bool on) {
    m_brightness = (brightness & 0x7) | (on ? 0x08 : 0x00);
}

void TM1637Display::setSegments(const uint8_t segments[], uint8_t length,
   uint8_t pos) {
    start();
    writeByte(TM1637_I2C_COMM1);
    stop();
    start();
    writeByte(TM1637_I2C_COMM2 + (pos & 0x03));
    for (uint8_t k = 0; k < length; k++) {
        writeByte(segments[k]);
    }
    stop();
    start();
    writeByte(TM1637_I2C_COMM3 + (m_brightness & 0x0f));
    stop();
}

void TM1637Display::clear() {
    uint8_t data[] = { 0, 0, 0, 0 };
    setSegments(data);
}

void TM1637Display::showNumberDec(int num, bool leading_zero, uint8_t length
   , uint8_t pos) {
```

```
    showNumberDecEx(num, 0, leading_zero, length, pos);
}

void TM1637Display::showNumberDecEx(int num, uint8_t dots, bool leading_zero
    , uint8_t length, uint8_t pos) {
    showNumberBaseEx(num < 0 ? -10 : 10, num < 0 ? -num : num, dots,
        leading_zero, length, pos);
}

void TM1637Display::showNumberBaseEx(int8_t base, uint16_t num, uint8_t dots
    , bool leading_zero, uint8_t length, uint8_t pos) {
    bool negative = false;
    if (base < 0) {
        base = -base;
        negative = true;
    }
    uint8_t digits[4];
    if (num == 0 && !leading_zero) {
        for (uint8_t i = 0; i < (length - 1); i++) {
            digits[i] = 0;
        }
        digits[length - 1] = encodeDigit(0);
    } else {
        for (int i = length - 1; i >= 0; --i) {
            uint8_t digit = num % base;
            if (digit == 0 && num == 0 && leading_zero == false) {
                digits[i] = 0;
            } else {
                digits[i] = encodeDigit(digit);
            }
            if (digit == 0 && num == 0 && negative) {
                digits[i] = minusSegments;
                negative = false;
            }
            num /= base;
        }
    }
    if (dots != 0) {
        showDots(dots, digits);
    }
    setSegments(digits, length, pos);
}



void Display(uint8_t PWM_Running, bool Blink, uint8_t Desired_Freq)
{
  if (PWM_Running == PWM_RUNNING)
  {
    if (Desired_Freq > 99) Display1.showNumberDec(Desired_Freq, false, 3, 1)
        ;
    else
```

```
    {
      Display1.setSegments(SPACE, 1, 1);
      Display1.showNumberDec(Desired_Freq, false, 2, 2);
    }
    if (Curr_Value > 999) Display2.showNumberDec(Curr_Value, false, 4, 0);
    else
    {
      Display2.setSegments(SPACE, 1, 0);
      Display2.showNumberDec(Curr_Value, false, 3, 1);
    }
  }
  else
  {
    Display_Timer++;
    if (Blink && (Display_Timer == DISPLAY_BLINK))
    {
      Display1.clear();
      Display2.clear();
    }
    else if (Display_Timer > (2*DISPLAY_BLINK))
    {
      Display1.setSegments(THREE_PHASE);
      Display_Timer = 0;
    }
  }
}


void Wait_A_Bit(uint32_t Executions_To_Wait)
{
  volatile uint32_t Timer_Temp2 = 0;
  while (Timer_Temp2 < Executions_To_Wait)
  {
    Timer_Temp2++;
  }
}


void Pwm_Disable()
{
   PWM_Running = PWM_NOT_RUNNING;
   cli();
   Init_PWM_Counter = 0;
   TCCR0A = 0;
   TCCR0B = 0;
   TCCR1A = 0;
   TCCR1B = 0;
   TCCR2A = 0;
   TCCR2B = 0;
   sei();
}

```

```
void Pwm_Config()
{
   if (Phase_Config == THREE_PH)
   {
       cli();
       PWM_Running = PWM_RUNNING;
       GTCCR = (1<<TSM)|(1<<PSRASY)|(1<<PSRSYNC);
       TCCR0A = (1 << COM0A1) | (1 << COM0B1) | (1 << COM0B0) | (1 << WGM00)
          ;
       TCCR0B = (1 << CS00);
       TIMSK0 = (1 << TOIE0);
       OCR0A = 0;
       OCR0B = 127;
       TCCR1A = (1 << COM1A1) | (1 << COM1B1) | (1 << COM1B0) | (1 << WGM10)
          ;
       TCCR1B = (1 << CS10);
       OCR1A = 0;
       OCR1B = 127;
       TCCR2A = (1 << COM2A1) | (1 << COM2B1) | (1 << COM2B0) | (1 << WGM20)
          ;
       TCCR2B = (1 << CS20);
       OCR2A = 0;
       OCR2B = 127;
       TCNT0 = 0;
       TCNT1H = 0;
       TCNT1L = 0;
       TCNT2 = 0;
       GTCCR = 0;
       Init_PWM_Counter = 0;
       Wait_A_Bit(BOOT_CAP_CHARGE_TIME);
       sei();
   }

}


void Pot_Switch_State_Check()
{
   if (POT_SWITCH_IS_ON)
   {
      if (PWM_Running != PWM_RUNNING)
      {
        if (Timer - Timer_Temp  > 1) Pot_Switch_On_Timer = 0;
        else  Pot_Switch_On_Timer++;
        Timer_Temp = Timer;
        Pot_Switch_Off_Timer = 0;
        if (Pot_Switch_On_Timer > POT_SWITCH_SAMPLES)
        {
          Pwm_Config();
          Pot_Switch_On_Timer = 0;
          Pot_Switch_Off_Timer = 0;
          Display1.clear();
```

```
          Display2.clear();
        }
      }
  }
  else
  {
      if (PWM_Running != PWM_NOT_RUNNING)
      {
        if (Timer - Timer_Temp  > 1) Pot_Switch_Off_Timer = 0;
        else  Pot_Switch_Off_Timer++;
        Timer_Temp = Timer;
        Pot_Switch_On_Timer = 0;
        if (Pot_Switch_Off_Timer > POT_SWITCH_SAMPLES)
        {
          Pwm_Disable();
          Pot_Switch_On_Timer = 0;
          Pot_Switch_Off_Timer = 0;
          Display1.clear();
          Display2.clear();
        }
      }
  }
}


ISR (TIMER0_OVF_vect)
{
   OVF_Counter++;
   if (OVF_Counter >= OVF_Counter_Compare)
   {
      if (Sine_Index == Sine_Len) Sine_Index = 0;
      if (Sine_Index_120 == Sine_Len) Sine_Index_120 = 0;
      if (Sine_Index_240 == Sine_Len) Sine_Index_240 = 0;
      //
      if ((Sine_Used[Sine_Index] - DEADTIME_SUB) < MIN_PWM_VAL)
      {
         OCR0A = 0;
      }
      else
      {
         OCR0A = uint8_t(Sine_Used[Sine_Index] - DEADTIME_SUB);
      }
      OCR0B = OCR0A + DEADTIME_ADD;

      if ((Sine_Used[Sine_Index_120] - DEADTIME_SUB) < MIN_PWM_VAL)
      {
         OCR1A = 0;
      }
      else
      {
         OCR1A = uint8_t(Sine_Used[Sine_Index_120] - DEADTIME_SUB);
      }
```

```
        OCR1B = OCR1A + DEADTIME_ADD;

        if ((Sine_Used[Sine_Index_240] - DEADTIME_SUB) < MIN_PWM_VAL)
        {
            OCR2A = 0;
        }
        else
        {
            OCR2A = uint8_t(Sine_Used[Sine_Index_240] - DEADTIME_SUB);
        }
        OCR2B = OCR2A + DEADTIME_ADD;

        OVF_Counter = 0;
        Sine_Index++;
        Sine_Index_120++;
        Sine_Index_240++;
    }
}


void Reverse_3_Phase()
{
    if (Phase_Config == 0)
    {
        Pwm_Disable();
        uint8_t tempIndex = Sine_Index_120;
        Sine_Index_120 = Sine_Index_240;
        Sine_Index_240 = tempIndex;
        PORTB &= ~((1 << PORTB0) | (1 << PORTB1) | (1 << PORTB2));
        PORTD &= ~((1 << PORTD3) | (1 << PORTD5) | (1 << PORTD6));
    }
}


void ADC_init()
{
    ADMUX |= (1 << REFS0);
    ADMUX &= ~(1 << REFS1);
    ADCSRA |= (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0);
    ADCSRA |= (1 << ADEN);
}


uint16_t analogRead(uint8_t channel)
{
    channel &= 0b00000111;
    ADMUX = (ADMUX & 0xF8) | channel;
    ADCSRA |= (1 << ADSC);
    while (ADCSRA & (1 << ADSC));
    return ADC;
}
```

10 References

For further information about our Variable Frequency Drive (VFD) project for AC motors, please visit our GitHub repository at https://github.com/LahiruCooray/VariableFrequencyDrive_Project. This repository contains detailed documentation, code, and resources related to our VFD project, offering insights into its development and implementation.

11 Signed Declaration by Other Group Members

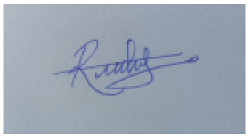
This document, detailing the design and development of a Variable Frequency Drive (VFD), has been thoroughly reviewed and cross-checked by an independent group to ensure accuracy and completeness. The contents include an overview of general VFD operation, detailed electronics design including schematics and PCB layouts, enclosure design, comprehensive programming information, photographs of the PCBs and enclosure, as well as daily log entries documenting the project's progress. The review and cross-checking process confirms that all aspects have been meticulously checked and validated.

Cross-checked by (members of Zero Gravity Lifting Device project - Group G) :

Name - Abeyrathna S.M.S.M.B. (210005H)

Signature - 

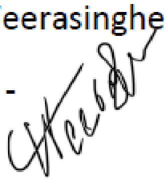
Name - Abeywardhane R.N. (210015M)

Signature - 

Name - Kumarasinghe R.D. (210321X)

Signature - 

Name - Weerasinghe C.N. (210687X)

Signature - 

12 Appendix A

This section describes the timeline of the project, including the time period of the project's tasks and the details of the tasks performed during that period.

1st February 2024: Deciding on the project and preparing project proposal: We, Lahiru Cooray and Danidu Dabare, a team of two electronic and telecommunication undergraduates, gathered to exchange ideas and thoughts on possible project ideas. Following a thorough discussion, we decided on the Variable Frequency Drives for AC Motors project as it aligns well with the learning outcomes of the module. A brief project proposal was created.

1st February 2024 - 9th February 2024: Reviewing progress and planning next steps: Here's a paraphrased version of the text: Throughout this week, we each conducted independent searches for resources related to the project. These resources encompassed existing products, research papers, and videos demonstrating these products in use within the industry.

9th February 2024 - 15th February 2024: Creating stakeholder map, observing users, and identifying needs: During this time, we identified the project's stakeholders, both internal and external, which helped us understand how to meet each of their needs. After that, we began observing users online, discovering instances where they had previously used similar products. We also compiled a list of requirements for the project, including user expectations and the functional specifications of the final product.

15th February 2024 - 22nd February 2024: Stimulation of ideas: During this period, our team brainstormed innovative ideas to make our product stand out and provide users with additional, convenient features.These concepts would help us develop better conceptual designs

1st March 2024 - 7th March 2024: Development of conceptual designs: In the subsequent stage of the project, we developed conceptual designs intended as potential foundations for the final product. We meticulously aligned these designs with our list of requirements to ensure they met all specified needs. Our process yielded three conceptual designs, each featuring block diagrams depicting the final product and sketches outlining the proposed features. Following thorough deliberation, we evaluated these designs and ultimately selected one as our preferred choice.

11th March 2024: Review: We conducted a comprehensive review of our work and the progress achieved thus far to ensure that our conceptual design aligned closely with our project goals.

12th March 2024 - 15th March 2024: Deciding on components During this time, we decided on what components will be used on the product. This was especially for main components like the microcontroller, IPM module, AC-DC converters.

15th March 2024 - 31st March 2024: Circuit design: During this period, we focused on designing the circuit for the product. We consulted existing schematics to enhance our understanding and proceeded to design the power circuit, IPM circuit and microcontroller circuit.

1st April 2024 - 12th April 2024: PCB Design: During this phase, we meticulously designed the Printed Circuit Board (PCB) using Altium, ensuring strict adherence to all PCB design standards and ensuring manufacturability.

1st April 2024 - 12th April 2024: Component Selection: Parallel with the PCB design, we also selected the rest of the components for the project such as resistors, capacitors, connectors, heat sinks, etc.

17th April 2024 - 19th April 2024: Enclosure Design: After completing the PCB, we began designing the enclosure. We sequenced this step after the PCB design to ensure that the enclosure size could be precisely determined based on the PCB's dimensions and components

20th April 2024: Sending the PCB for manufacture: The PCB design files were sent to JLCPCB, China to be manufactured..

30th April 2024: Arrival of manufactured PCB

7th May 2024: Soldering: On this day, we soldered the components onto our three PCBs.

As of 5th June 2024, we have successfully completed the PCB assembly and the 3D printed enclosure for our Variable Frequency Drives for AC Motors project. We were not able to carry out our testing phase successfully due to the strike held by the non-academic staff and the laboratories being closed.

13 Appendix B

Declaration of Non-Usage of Arduino Libraries or Arduino Type Coding

We, A.D.T. Dabare and P.L.L.K. Cooray, declare that the project titled "Variable Frequency Drive (VFD) for AC Motors," does not utilize any Arduino libraries or Arduino type coding in its current form. Specifically, the previously used Arduino library TM1637Display.h for the display has been removed, and the current version of the code is independent of Arduino libraries or Arduino-specific coding practices.