# Design Decisions

As we discussed in the complete simulation report, we will use the topology consisting of a three-phase full bridge rectifier feeding a buck converter that eventually drives the motor. The main block diagram of the system is shown in Figure 1 and Figure 2. Moreover, to accomplish various bonuses, we have some other submodules.

The motor drive system is designed in a modular manner to make it easy to analyze and debug. As well as to avoid epic failures; for example, a total board failure due to the explosion caused by an arc born from not enough clearance in the AC mains line. Thanks to the modular approach, we were able to get rid of only the faulty parts and redesign only them, not the complete system.

Hence, the motor driver system consists of various subsystems, as the block diagrams suggest. Each subsystem will be handled in detail to guarantee the subsystems work properly not only standalone but also after the integration.

In Figure 3, the completely integrated and finished version of the system is shown.

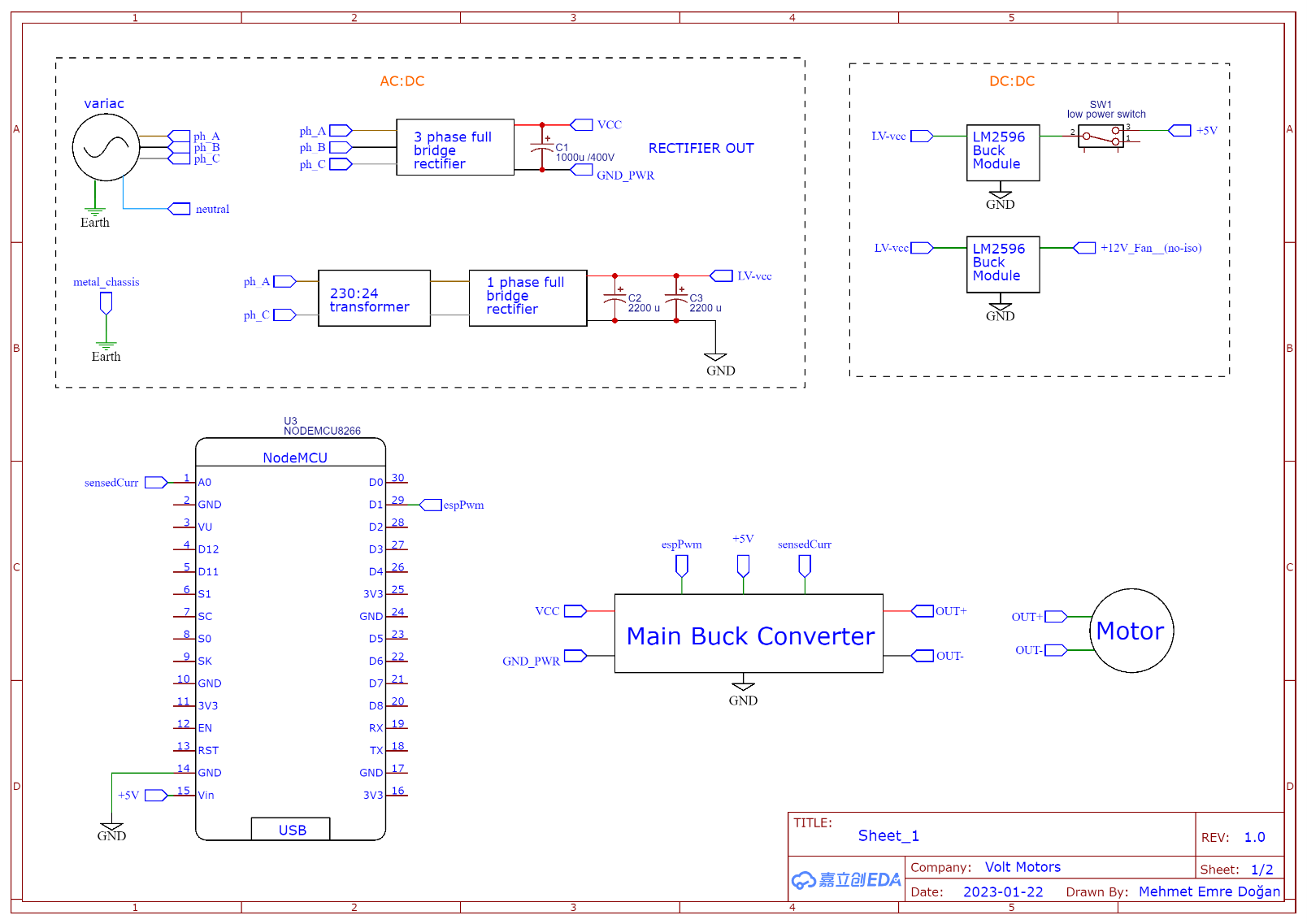
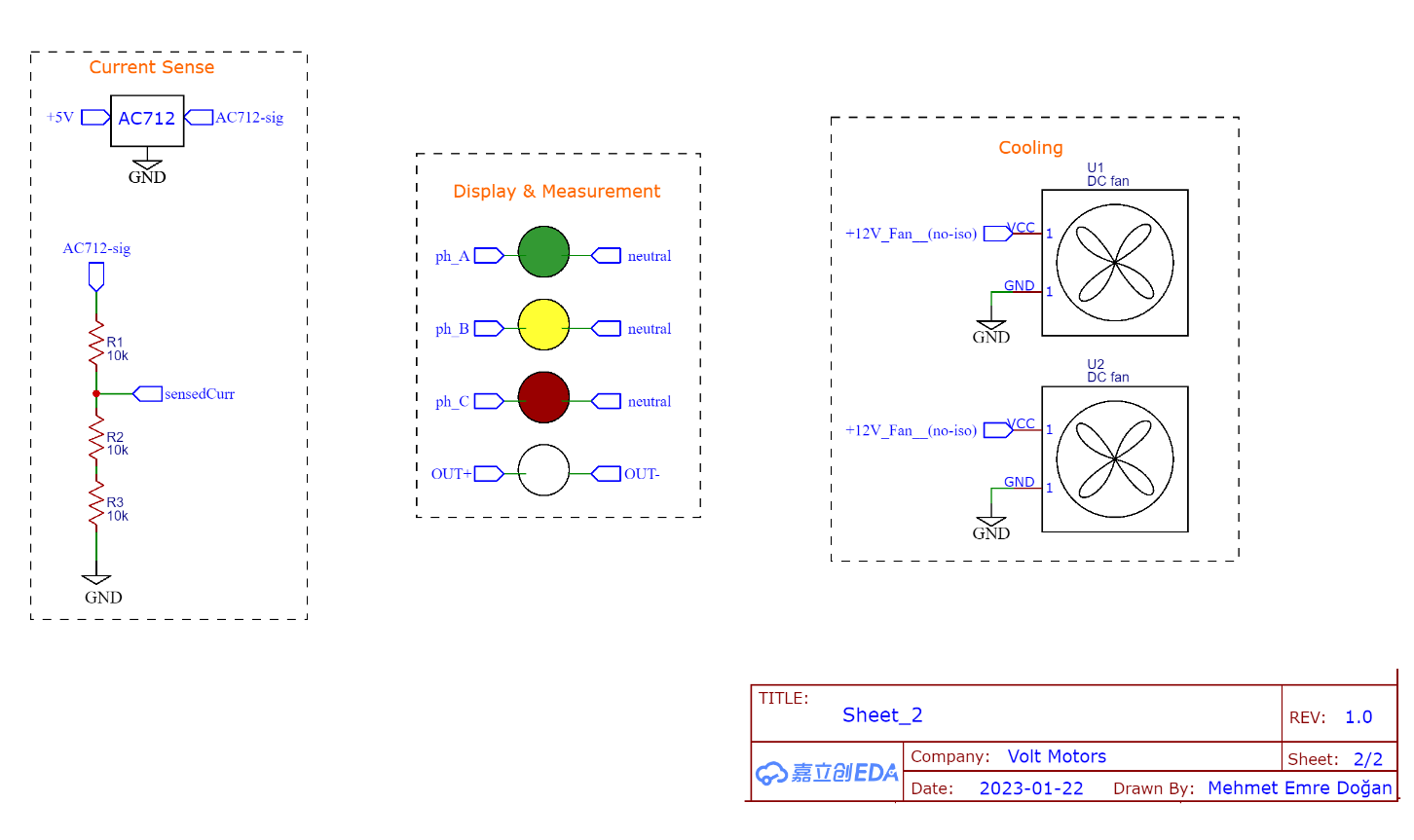


Figure 1: Main block diagram sheet 1

Figure 2: Main block diagram sheet 2

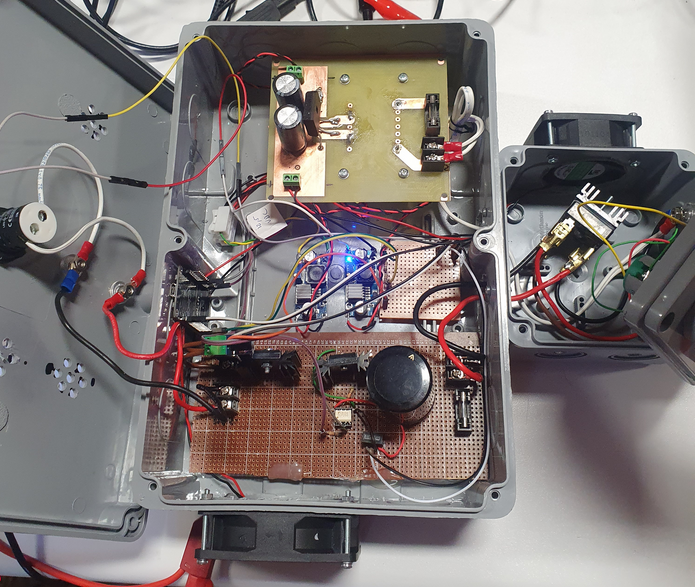


Figure 3: The completely integrated system from the demo day

## AC-DC Converter Subsystem

To drive the DC motor, we need DC power. Moreover, to run the MCU (Micro-Controller-Unit) we also need low-power DC voltage, which needs to be isolated from the mains DC power.

### 3-Phase Full Bridge Rectifier + Capacitor

This rectifier, shown in Figure 4, rectifies the mains AC power provided from the variac to DC 200V. Although the rated voltage of the motor is 180V DC, we rectify it to 200V DC because we plan o reach the rated voltage at 90% duty cycle. The discussion of why is provided in the Duty Cycle subsection of the MCU section. Then the voltage is smoothened (decreased its ripple) by the 1mF 400V polarized aluminum capacitor, shown in Figure 5, connected to the exit of the 3-Phase FBR.

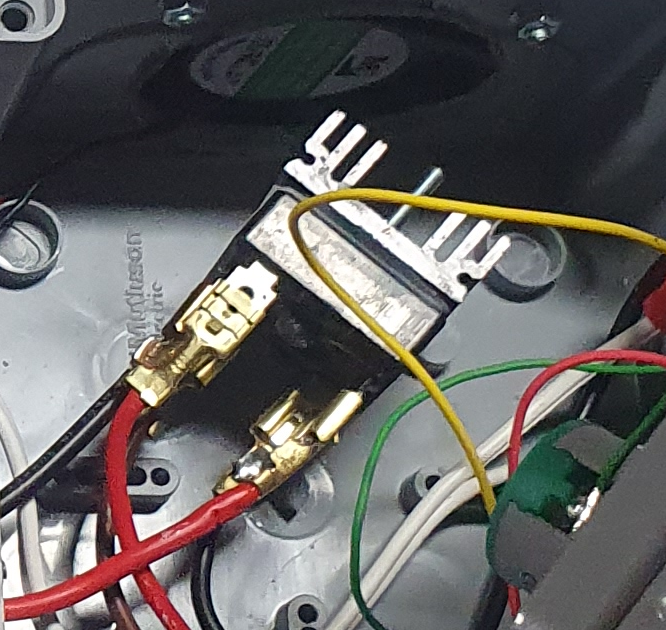


Figure 4: The 3-Phase FBR

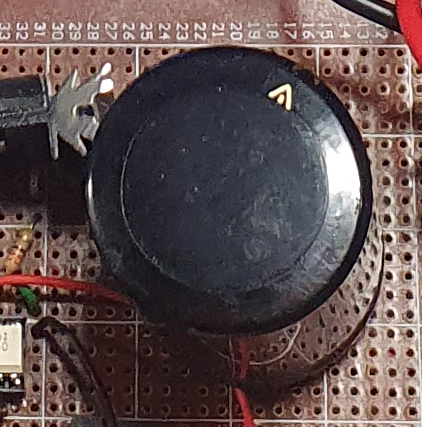


Figure 5: The main capacitor

### Low Power Adapter

The low-power adapter schematic is shown in Figure 6. Its job is to provide low DC voltage to the LM2596 buck regulators to power MCU, fans, sensors, optoisolator, and other low-power devices.

**

*Figure 6: Low Power Adapter Schematic*

This adapter, first, steps down the voltage using the transformer, then rectify with the single-phase FBR. Afterward, thanks to the large capacity of aluminum capacitors, the output ripple is smoothened. Moreover, in design, 2 ceramic capacitors are put to decrease the equivalent ESR of the capacitors subsystem. However, these are not soldered in implementation, since we did not have an ESR problem.

The fuse provides overcurrent protection and screwed terminal connectors ease making cable connections.

The VCC provided by this adapter is expected to be near 20VDC when two AC lines from the variac are connected to its AC\_IN. Note that, the variac is adjusted to 85 V AC line to neutral, resulting in 148 V AC line to line.

However, since the transformer is nearly unloaded, due to the low-power devices connected, this output voltage is expected to increase. However, thanks to the 36 V DC maximum input voltage and built-in feedback and filter system of LM2596 buck regulator modules, neither this voltage rise nor the ripples are a problem. LM2596 modules provide clean and consistent voltages to MCU and other low-voltage peripherals.

The PCB design and an image from the demo day are shown for the adapter in Figure 7 and Figure 8, respectively.

metin, parketme, çizgili içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 7: The low-power adapter PCB

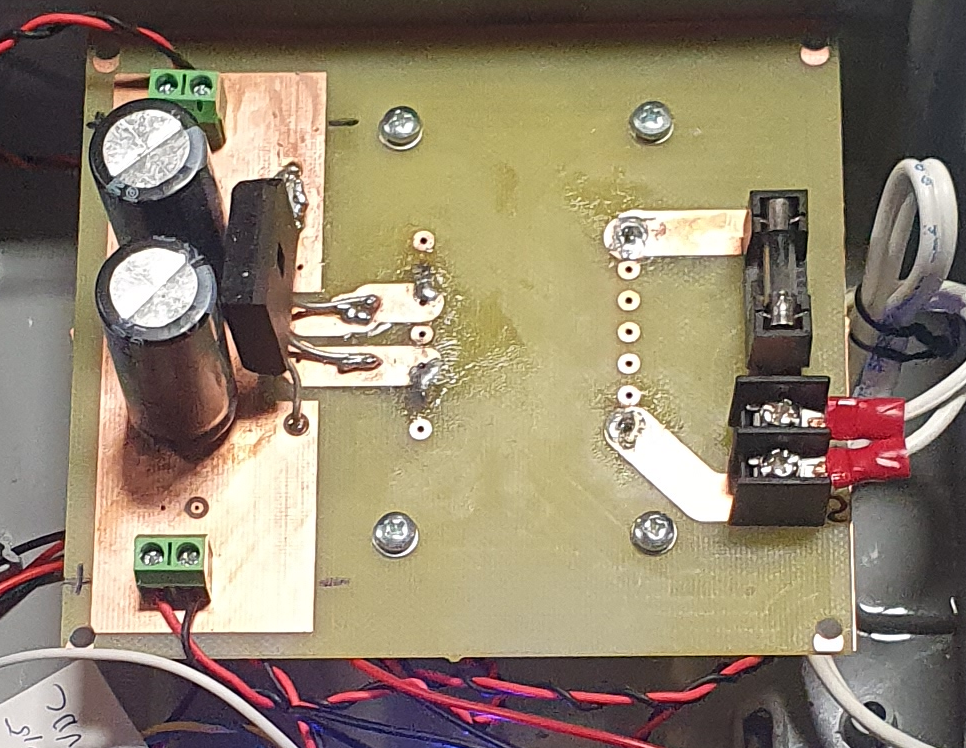


Figure 8: The low-power adapter from the demo day

## Main Buck Converter Subsystem

The main buck converter’s schematic is shown in Figure 9. The schematic includes 5 VDC and GND busbars to power the peripherals.

The output of the 3-Phase FBR is connected to the board using screw terminals. This voltage is filtered out via a 1 mF 400V aluminum capacitor, shown in Figure 5. The additional 10 nF ceramic capacitor is added to the design in case we had an ESR problem. But in implementation, it is not soldered since we had no ESR problems. Then this voltage passes a protective fuse and is called VCC, or VCC\_PWR.

The isolated DC-DC converter creates an isolated +12VDC from the mains 5VDC to feed into the optocoupler’s VCC and GND.

The optoisolator receives the PWM signal from the ESP and amplifies it in an isolated manner to drive the gate of the IGBT. 330 Ohm of resistors is connected in series to the LED part of the Opto to limit current. The Opto’s forward voltage is near 1.8 Volts and this resistor guarantees the ESP provides current to the Opto less than 10 mA.

The resistor between the Opto’s output and the gate of the IGBT is also 330 Ohms because it works. We first tried 330 Ohm and it worked perfectly, then we did not change it.

The freewheeling diode parallel to the output of the converter makes sure the current due to the inductance of the motor can have a path to circulate when the IGBT is on OFF duty. Therefore, the IGBT is protected from damages due to overvoltage caused by the motor inductance.

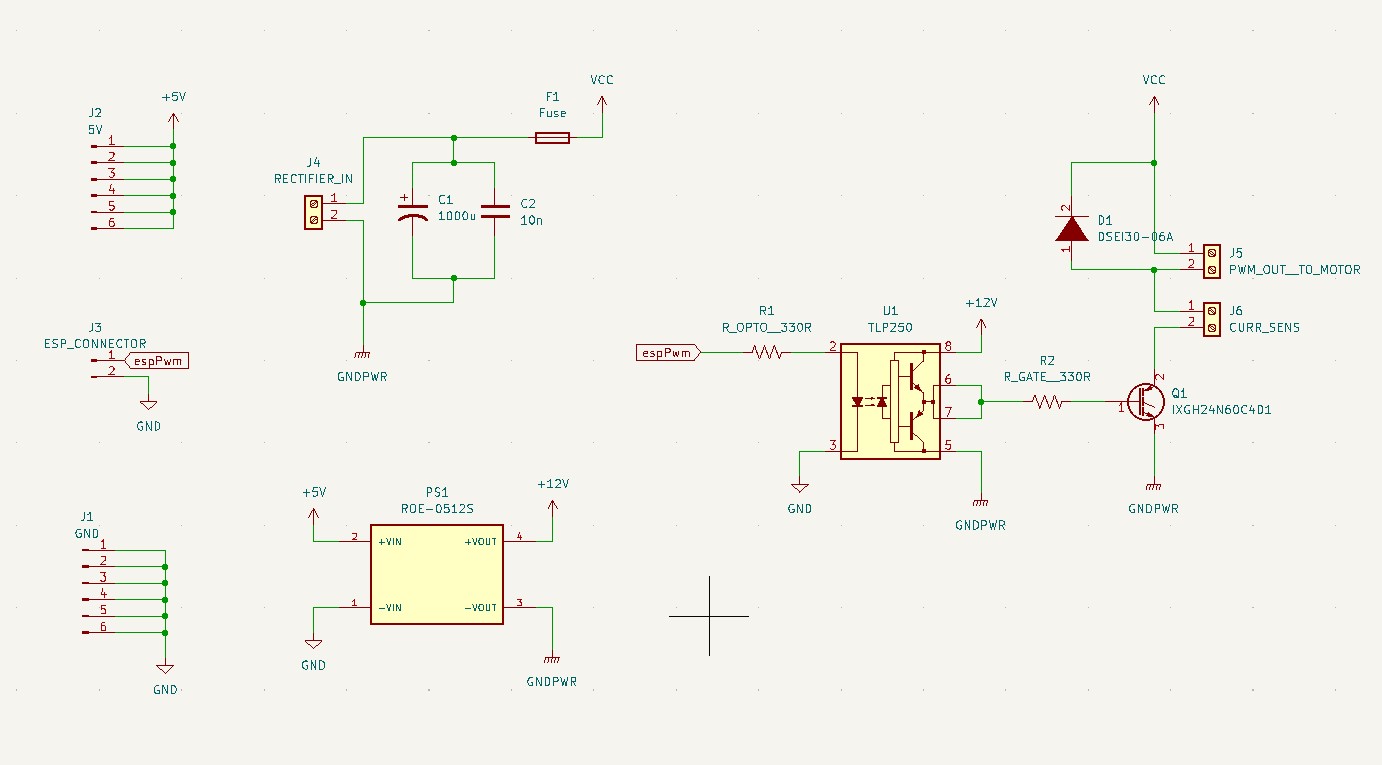
 The current sensor is connected via the screw terminals to measure the current drawn. The screw terminals ease the change of the sensor in case of failure.

Figure 9: main Buck Converter

metin, elektronik eşyalar içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 10: The failed main buck PCB

We designed a PCB for the Main Buck Converter, as shown in Figure 10. However, it failed due to some known and unknown reasons.

One of the known reasons is that the PCB has been wrongly produced. The positive pin of the Opto’s LED is shorted to the ground because of a manufacturing error. The error is understood by comparing Figure 11 with Figure 10.

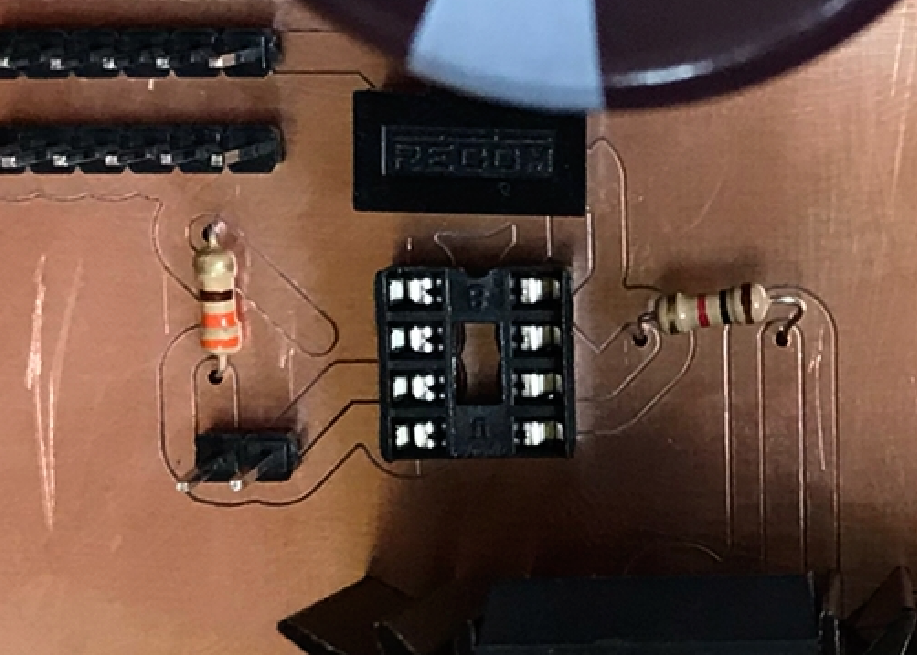


Figure 11: The Opto LED positive input, manufactured

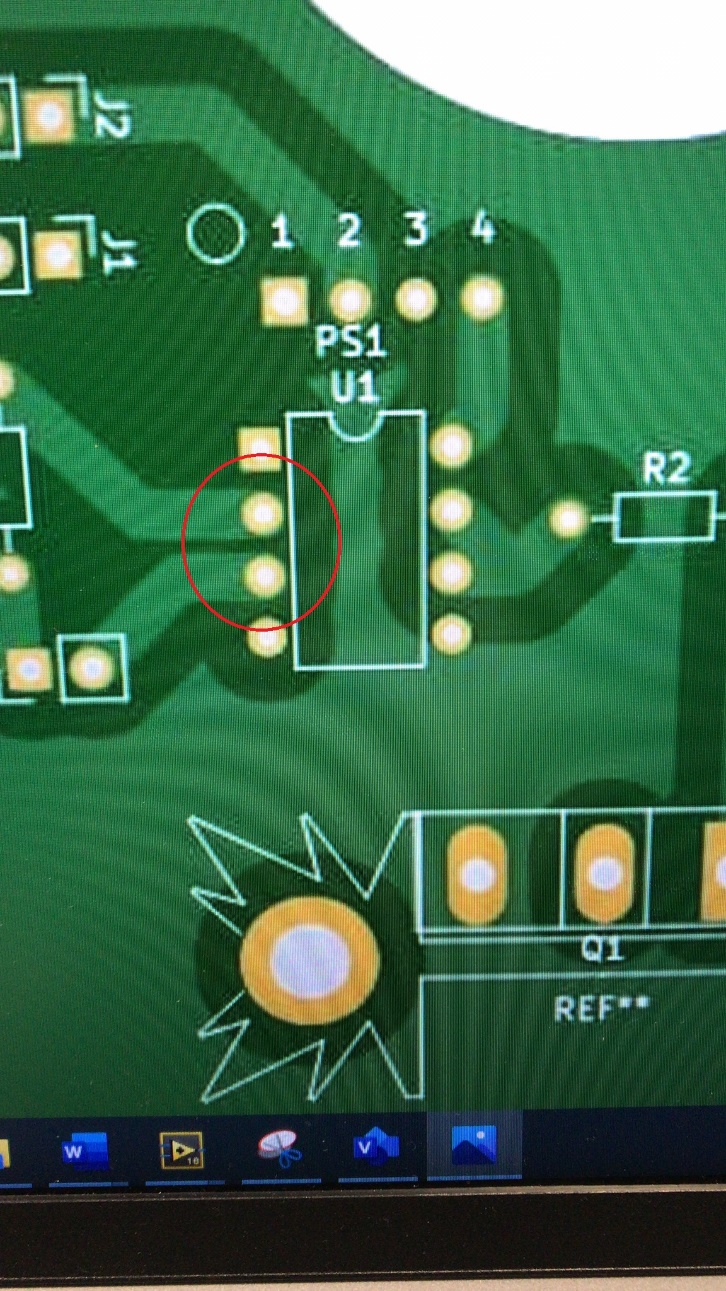


Figure 12: The Opto LED inputs, designed

Although we solved this problem by bending the optoisolator’s LED pins and space soldered these pins. However, unfortunately, the board was still not working due to the errors we were unable to diagnose. Afterward, we rebuilt the buck circuit on the stripboard before the demo, as shown in Figure 13.

Thanks to the modularity approach, we were able to use the rest of the designs without any problems.

bilgisayar içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 13: The buck converter on the stripboard

## MCU & Software Subsystem

The MCU subsystem is responsible to generate the PWM signal with the duty cycle that the users select via the modern Web GUI (Graphical User Interface), as shown in Figure 14. Moreover, as shown in Figure 14, the software has a WebSerisal to print timestamped important run-time data.

The online Web GUI demo is available [1]. Note that the Set Direction and Close Loop menus are inactive due to the system not including H-Bridge and Close Loop Speed Sensor. However, these menus exist for future versions of our design updates.

metin, ekran, iç mekan, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 14: WebGui with WebSerial from demo day

### Duty Cycle

The duty cycle is adjusted using the slide bar shown in Figure 15. The range of the used duty cycle is 0.1-0.9. That is, the motor reaches its rated speed and power with the duty cycle at 90%, instead of 100%. The reason for that approach is to avoid unstable switching at 0% (fully off) and 100% (fully on) by providing a safety margin. For example, due to environmental noise, the PWM signal may deviate slightly from the adjusted duty, and the safety margin makes sure this deviation will not change the state of the system.

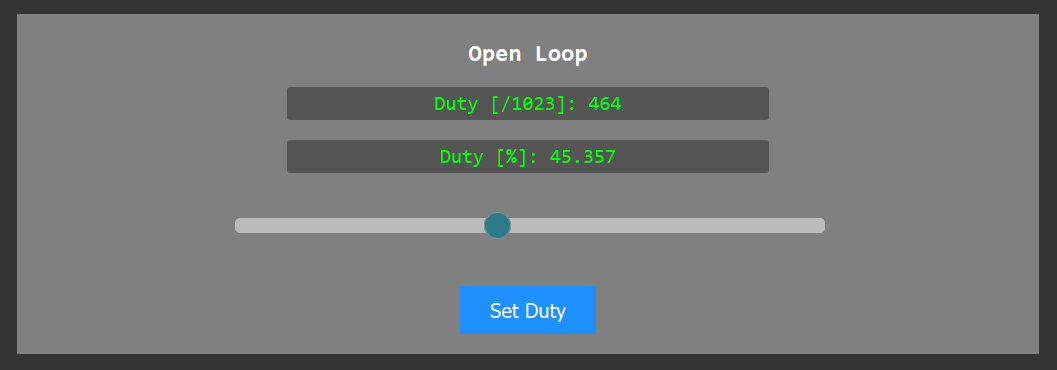


Figure 15: Open-loop duty selector slide-bar

### Switching Frequency

The switching frequency of the generated PWM signal is chosen as 1 kHz since the Main Buck Converter uses an IGBT to switch the motor. We know that the IGBTs are not capable of switching at high frequencies due to their high rise time and turn-on/off delay times, in addition to their huge switching energy. For example, we used IXGH24N60C4D1 N channel IGBT, whose switching energy is 1.46 mJ. On the other hand, LSIC1MO120E0080 N Channel SiC MOSFET only has 0.25 mJ of switching energy.

Moreover, designing a circuity capable of high switching frequency requires more attention and expertise. For example, the paths carrying the PWM signal should be properly engineered. However, due to our modular approach, we use jumper cables to carry the PWM signal.

As a result, on the first try, we used 1 kHz as the switching frequency. Afterward, we continued with that frequency because the integrated whole system worked pretty decent with a 1 kHz frequency.

### Power

The used board NodeMcu ESP8266 needs +5VDC from its Vin pin. The required voltage is supplied by an LM2596 buck regulator. Moreover, a small switch is attached in series to avoid booting bugs caused by the slowly increasing variac voltage. Thanks to that switch, we power on the ESP manually when the variac voltage reaches its rated value.

### Current Sensor

ACS712 Current sensor measures the motor current for debug purposes. Unfortunately, the existing ACS712 was burned due to misconnections, and replaced with another. However, since the current scale of the new one is different, the reading is wrong. The code needed to be updated but since it was the last day, we do not want to cause further bugs by editing the code, we did not edit.

In the future, a current feedback loop control could be added by just a software update thanks to the current sensor.

#### Voltage Divider

To convert the 5 V analog level of the current sensor to the 3.3 V analog level of the ESP, a resistor divider network is used.