**DAMNED: Digital Analog Modular NeoPixel Enable Display**

**- an extensible IoT platform**

1. **Features**

* ESP32-S3 core with 4MB Flash 2MB PSRAM
* Support for Adafruit CircuitPython and associated libraries
* I2C, I2S, SPI, UART connectivity
* Universal USB C power connector
* 24 NeoPixel RGB LED ring display
* Continuously rotatable stepper motor arm
* WiFi connectivity
* LiPo battery support
* Real time clock

1. **Applications**

* Environmental monitoring systems
* API interface devices
* Minimal games
* Wayfinding devices
* Musical instruments
* Time management systems
* Security alert and monitoring systems
* Training and quiz aides

1. **Description**

The DAMNED (Digital Analog Modular NeoPixel Enabled DIsplay) is an extensible IoT platform. Built on an ESP32-S3 core it can be programmed using CircuitPython. Multiple digital communication interfaces exist to allow for a wide range of compatible devices to augment the capabilities of the core DAMNED system. WiFi and a system RTC are available for real time, modern IoT applications that allow for interfacing with API’s. Analog input is also an option with several channels accepting signals from analog devices.

Core display output for data visualization includes a ring of 24 individually addressable RGB LEDs. These programmable LEDs allow for any light display from the most basic single color ring to complex configurations of moving patterns. A continuously rotatable stepper motor arm is also available with rotational speeds on the order of 60 RPM.

The device is housed in an enclosure that allows integrated access to all communications ports and power input. A magnetic faceplate allows for easy disassembly and the addition of a battery or replacement parts.

1. **Inputs and outputs**

[Provide a diagram that shows inputs and outputs of your device. This should be a simple box with arrows going in and out. The direction of the arrows should display the flow of power or information. Make sure you include wireless as well as wired connections.]

1. **Absolute maximum ratings**

|  | **Max** | **Unit** |
| --- | --- | --- |
| USB supply voltage | 6.5 | V |
| LDO regulator input voltage |  | V |
| LDO regulator output current |  | A |

Table 1: absolute maximum ratings

**6. Electrical Characteristics**

|  | **Min** | **Max** | **Unit** |
| --- | --- | --- | --- |
| USB supply voltage | 4.75 | 5.25 | V |
| LDO regulator input voltage |  |  | V |
| Operating temperature | 0 | 37.8 | °C |

Table 2: Recommended Operating Conditions

|  |  | **Unit** |
| --- | --- | --- |
| **Output voltage** |  | **V** |

Table 3: Nominal stepper motor supply output voltage with 5 V input and 1 A load

## **6.1 Measured Characteristics**

[This is where you list or provide graphs of characteristics you measured. You’ll want to include the turn on time of the voltage regulator as well as any ripple voltage or line regulation data you captured from the oscilloscope. ]

| **Vin (V)** | **Vout (V)** | **Iin (mA)** | **Iout (mA)** | **Pin (W)** | **Pout (W)** | **Peff (%)** |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |

Table 4: Stepper motor supply efficiency with 5 V input and 10 ohm load

| **Vin (V)** | **Vout (V)** | **In (mA)** | **Percent difference of Vout from 4 V** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table 5: Stepper motor supply input voltage range under 10 ohm load

| **Vin (V)** | **Vout (V)** | **Percent difference of Vout from 4 V** |
| --- | --- | --- |
|  |  |  |

Table 6: Dropout voltage required to maintain motor supply (4 V) to within 2% under 10 ohm load

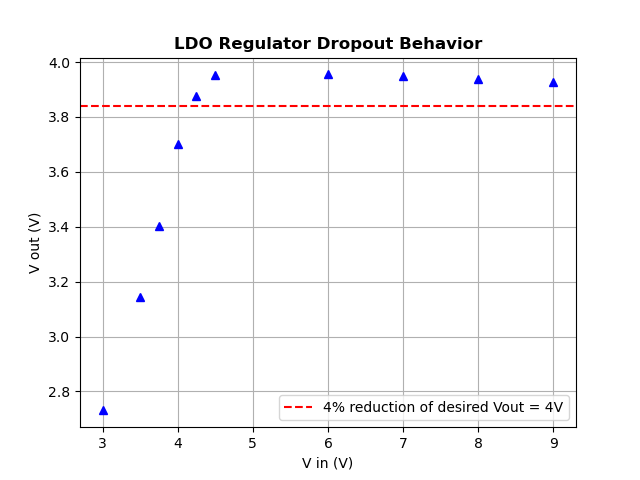
****

Figure 1: Vin vs Vout for stepper motor voltage regulator

**Ripple voltage and line regulation at 1 kHz**

| **Vin,pp (V)** | **Vout,pp (mV)** | **Line regulation  (ΔVout / ΔVin) x 100 (%)** |
| --- | --- | --- |
|  |  |  |

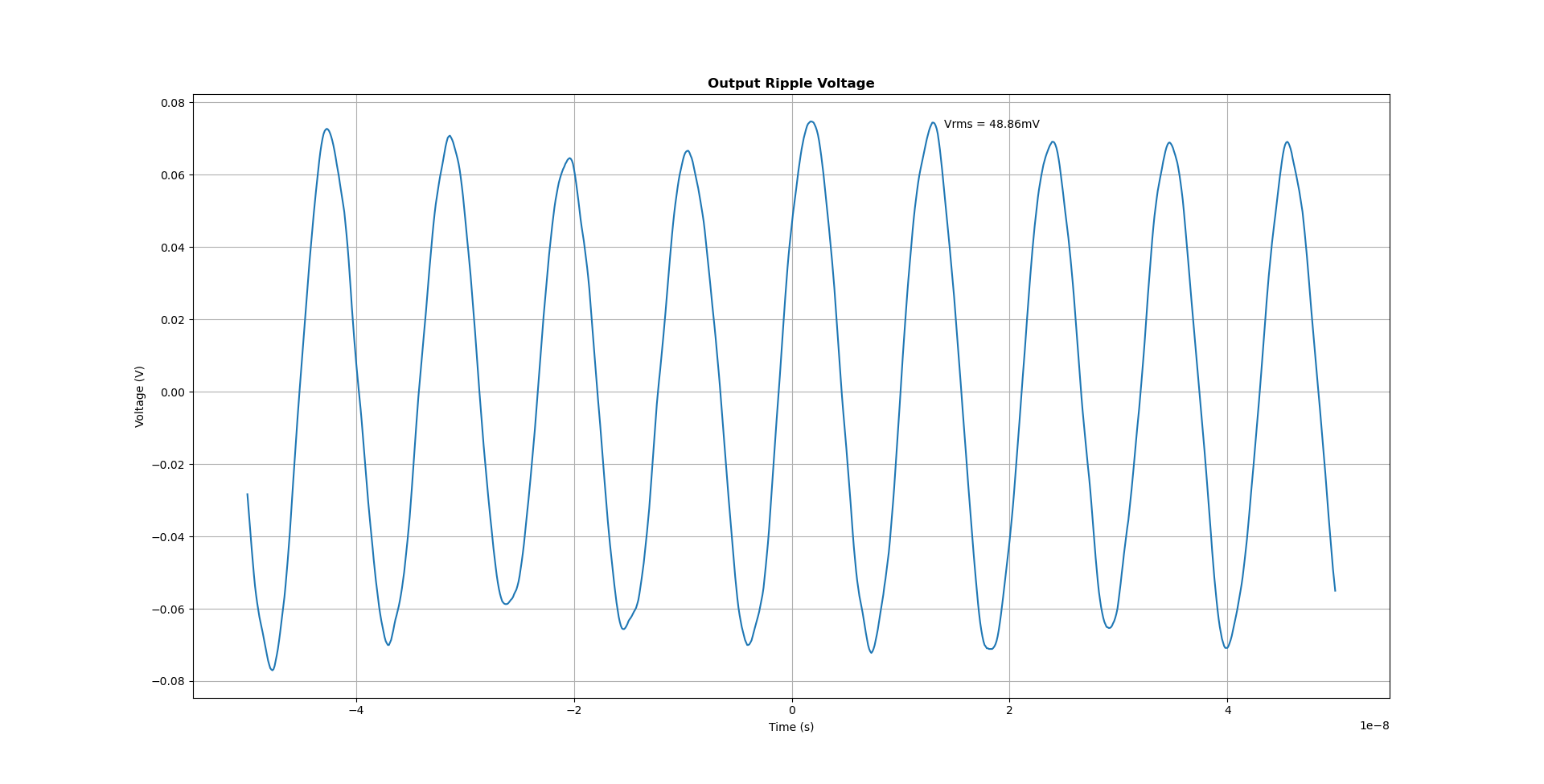
****

Figure 2: line regulation output ripple voltage

**Turn-on time of stepper motor voltage regulator is .**

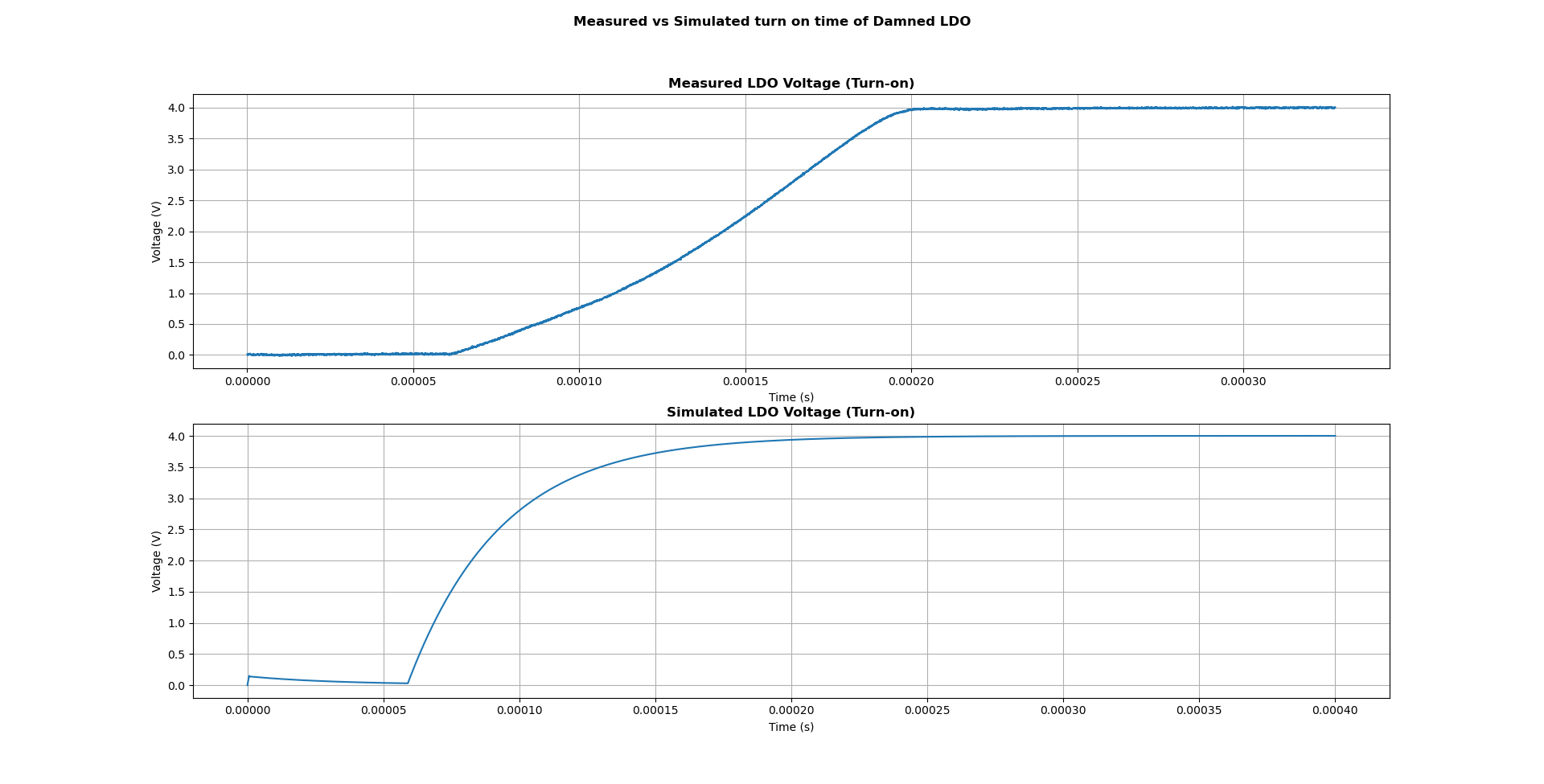
****

Figure 3: voltage regulator turn on, measured

## **6.2 Modeled Characteristics**

**Insert figure of LDO regulator circuit for motor control system, with figure label**

Figure 4: stepper motor voltage regulator circuit

**Line Regulation of motor control system**

|  | **Test conditions** | **Typical** | **Unit** |
| --- | --- | --- | --- |
| ΔVOUT/ΔVIN Output voltage line regulation | Vout = 5 V Iout = 1 A | 2.91 | %mV |

| **Vin (V)** | **Vout (V)** |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table 7: Input voltage vs output voltage at 1 A load, motor control system

| **Rload ()** | **Vin ()** | **Iin (A)** | **Vout (V)** | **Iout (A)** | **Peff (%)** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Table 8: Load Regulation and efficiency of motor control system

**Turn on time of motor control system**

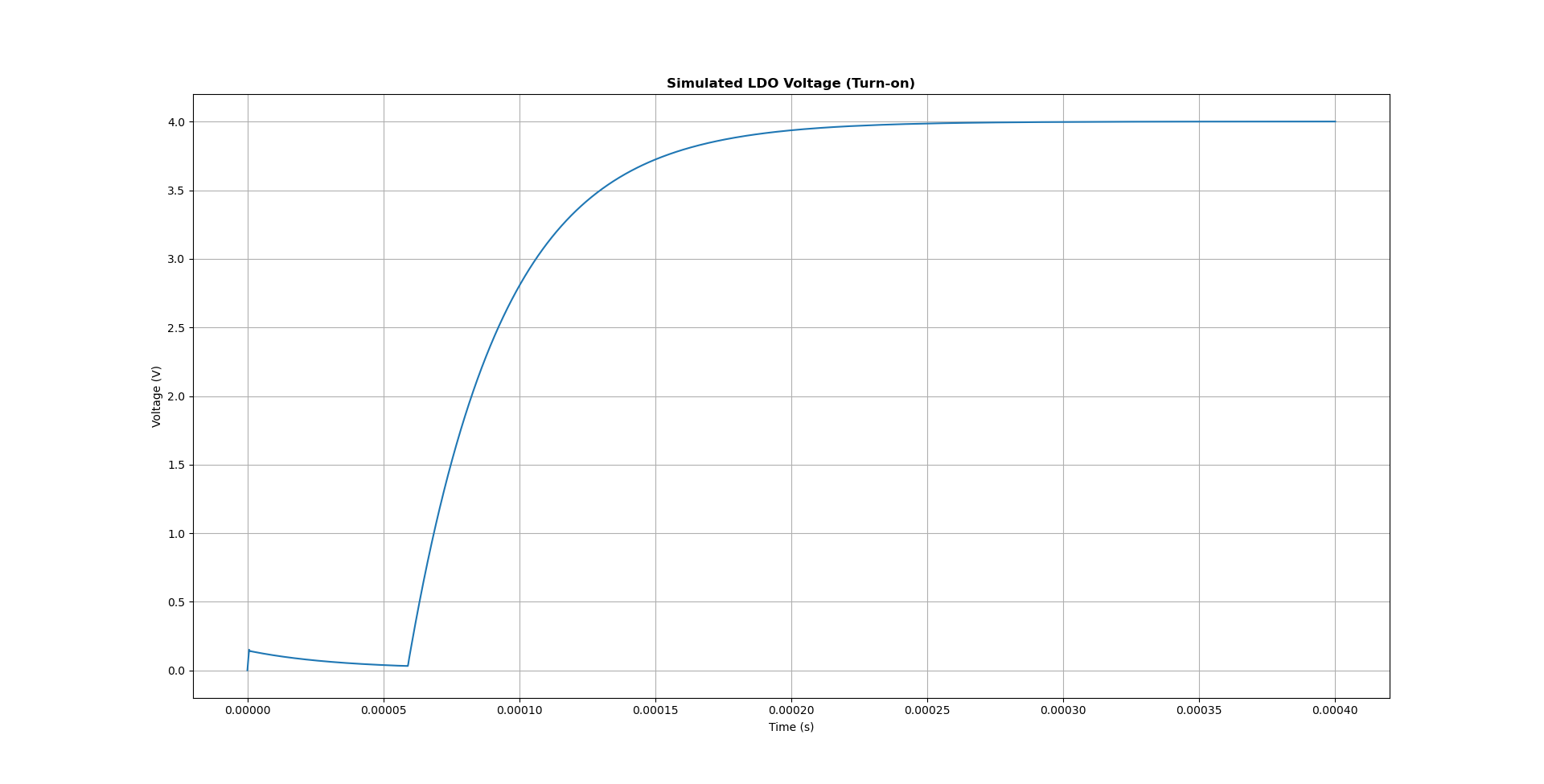
****

Figure 5: voltage regulator turn on, modeled

Typical turn on time for the motor control system power supply is X.

# **7. Detailed Description**

**7.1 Functional Block Diagram**

[Provide the functional block diagram of the DAMNED project and briefly walk users through the various parts or subsystems, very briefly explaining how each one contributes to the overall functionality. This block diagram can be copied directly from the course syllabus but **make sure you cite this as it is not your own work!** More detailed descriptions will be provided below.]

**7.2 Feature Description**

**7.2.1 Power System**

[Describe the operation of the power module. Make sure to include citations!]

**7.2.2 Sensors**

[Describe the sensor inputs, what protocols are used, any I2C addresses, and what sensors your device has attached. Make sure to include citations such as a link to the datasheet for the sensor you are using! ]

### **7.2.3 Software and Processor**

List the microcontroller and development board used in the DAMNED project. Provide some basic stats on the microcontroller such as processing speed, amount of memory, and any peripherals you are using in the DAMNED project.

List the software language used with the microcontroller and provide a bulleted list of library dependencies required to operate core functionality of the DAMNED

**7.2.4 Web connectivity**

[Discuss briefly how to connect to the internet and the module that performs this function. Describe where you are pulling data from and what data (if you are pulling data).]

The DAMNED features 2.4 GHz WiFi connectivity via an embedded ESP32-S3 microcontroller. The module supports 802.11 b/g/n network connectivity.

| **Device MAC address** |
| --- |
|  |

**7.2.5 Display**

**7.2.5.1 Lights**

[Describe how the lighted display works and information about colors and pixels. Make sure to include citations!]

Provide information on the lights on the DAMNED: what are they? How many are there? Which pin on the Feather is used to control them? Reference any libraries listed above in section 7.2.3

**7.2.5.2 Motor arm**

Provide information on the motor arm used on the DAMNED: How many steps per rotation are there? What is the degree moved per step? How is the arm controlled? For answers to some of these questions you might want to check out the motorFunctions.py library on Github.

**Stepper motor register to terminal mapping**

|  | **AO1** | **AO2** | **BO1** | **BO2** |
| --- | --- | --- | --- | --- |
| **h2A** |  |  |  |  |
| **h32** |  |  |  |  |
| **h2E** |  |  |  |  |
| **h36** |  |  |  |  |

# **8. Application and Implementation**

[Here describe a few ways the DAMNED project can be used. This section is divided into three sub-sections - applications, setup, and operation.]

## **8.1 Applications**

[Explain a typical application or two of the DAMNED project. Feel free to share your explanation with others in the class who have a different one and copy each others explanations. Include a simple diagram or a photo to help your explanation.]

Provide a brief description of how the DAMNED may be used to monitor and log temperature data. Include the part number for the sensors used as well as the associated drift, COV, and decay time values for the sensors.

Provide a brief description of how the DAMNED may be used to automatically set the real time clock, automatically set and continually update time, and display the temperature using the Neopixel ring and stepper arm, respectively. Give a summary of how time is mapped to the ring and temperature is mapped to the arm position.

Provide a brief description of how the DAMNED may be used for your own personal project. Be sure to include the sensors used, the data displayed, and how information is mapped between the sensors and the display (Neopixel ring + stepper motor arm).

## **8.2 Setup**

[Give the user detailed information about how to set up the DAMNED device. Assume they are technically competent, but are unfamiliar with this project. Photos are good to include here.]

Explain how a user would configure the DAMNED to connect to any WPA2 WiFi network with the associated password. Explain how to configure the DAMNED to write data to a specific ThingSpeak channel.

List the API calls used to access time and weather data and explain how those calls can be adjusted to meet the needs of the DAMNED located in any location.

Explain how a user would configure the DAMNED to push data to a public ThingSpeak channel. Include a link to your public ThingSpeak channel as an example.

Explain how any additional sensors would be connected that were used in your own project. Include addresses of any I2C sensors you used.

## **8.3 Operation**

[Describe how the device operates, including how you might switch sensors or Thingspeak channels, if it needs to be restarted regularly, how to calibrate it, or what normal operation looks like. Photos are good to include here.]

# **9. Mechanical Layout and Dimensions**

[Here you will provide diagrams for the mechanical dimensions of the case and the printed circuit board.]

## **9.1 Printed Circuit Board**

[Provide diagrams of the PCB - the diagrams should have dimensions. Use your KiCad files and if you’d like you can use the 3D view. Look at the TI datasheet to get ideas for how dimensions should be listed.]

## **9.2 Case**

[Same thing for the case. The CAD files with dimensions can be copied and pasted in this section. A photo wouldn’t hurt either.]

Include dimensions of the DAMNED 3D printed case here. You do not need to include dimensions of the laser cut pieces, only the 3D printed enclosure. You can use the SolidWorks design files from DA7 for this. [Here is a YouTube video](https://www.youtube.com/watch?v=58rqQbyrHbg) that shows you how to make a dimensional drawing from a SolidWorks part.

# **10 Citations**

[Wherever you cited other documents make sure you include in-text citations with a numbered list here in IEEE format]

There is a guide on Moodle that explains how to format citations using the IEEE format.