University of Illinois Urbana-Champaign

ECE 445 Spring 2022

ECE Senior Design

Design Proposal

Pranav Goel **Ben Xie** **Hongru Wang**

TA: **Pooja Bhagchandani**

Team 47

February 11, 2022

Contents

1 Introduction

1.1 Problem

High productivity is something many people try to achieve with little success. One of the most powerful systems to optimize their productivity output is to make a to-do list. This system is often used via apps people access from many devices like phones, laptops, etc. However, these apps can ultimately lower productivity than increase it. Checking your to-do list can be a multi-step process that takes your focus away from your original task and results in you working on a completely different task. This makes it easy to forget to check the app or interact with it entirely. In addition, having your to-do list hidden on your tablet or phone makes it easier to ignore, especially when notifications are often dismissed for no reason other than cleaning up the lock screen. It can also allow you to get distracted since less productive apps become easily accessible. There could be many factors in one's working environment that can lower productivity, such as CO_2 levels [**?**], temperature [2], and humidity [3]. Many people don't even realize that something is wrong and will continue pushing onwards, attributing their lack of focus and concentration to factors such as lack of sleep or stress [4]. Boiling this down, we need a better method of keeping track of daily tasks. We also need something that continuously monitors one's working environment and informs them of issues affecting optimal productivity conditions.

1.2 Solution

To give people a better way to monitor their daily tasks, as well as their working environment, we propose building a desktop device that can display one's to-do list in addition to monitoring environmental factors such as CO_2 levels, temperature, and humidity. This provides a constant reminder of what needs to be done on your desk, making it easier to check and difficult to ignore. Being a physical device on a desk will also allow it to collect data about the environment. An integrated motion sensor will also allow it to track sitting time.

This device will use an e-ink screen to display tasks due to its readability and low idle power consumption. Because e-ink displays have low refresh rates, we will also include individually addressable RGB LEDs to communicate some information in real-time. We will use a speaker to alert the user non-visually of upcoming deadlines and environmental warnings. We'll include a rotary encoder and buttons to interact with the device physically. The primary function of these will be to change the status of the tasks between to-do, in-progress, and finished. The status for each task will be shown by the LEDs along the edge of the display. We will also include a variety of sensors in the device to measure the environmental factors that were outlined above. Everything will connect to an ESP-32 MCU which will handle controlling the device as well as communicating with a server that contains all of the tasks. The tasks will be sent to the server via a website.

1.3 Visual Aid

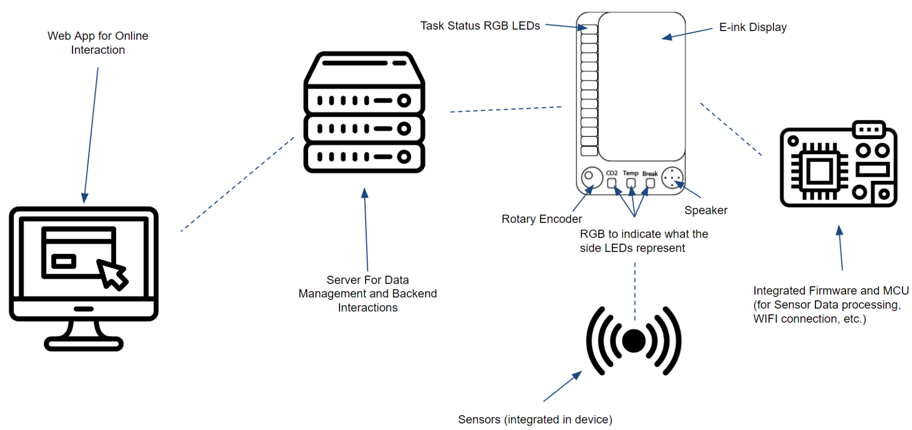

Figure 1: Visual Layout of how Project will Work

1.4 High Level Requirements

- **Requirement 1**: The device must be able to display at least 12 tasks simultaneously with newly created tasks displaying on the device within 1 minute.
- **Requirement 2**: The device must be able to measure and send the environmental CO_2 level, temperature level, and humidity level to a server at least once every 15 minutes.
- **Requirement 3**: The device must be able to determine if user is present or not within 5 minutes of the current status changing.

2 Design

2.1 Block Diagram

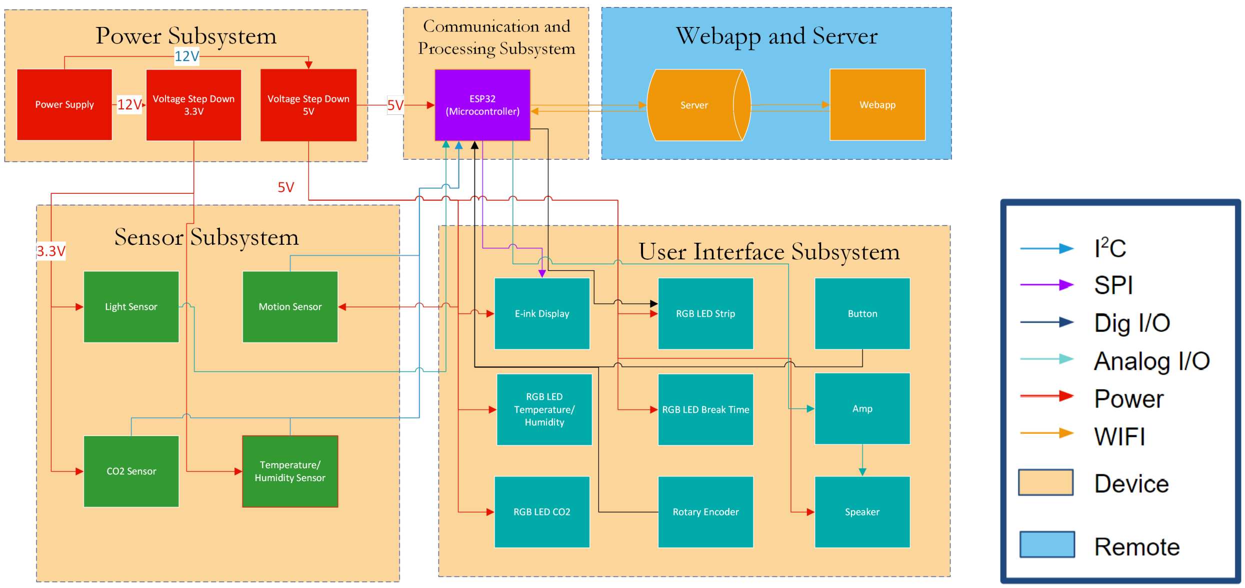

Figure 2: Block Diagram and Legend Describing

2.2 Subsystem Overview

2.2.1 Power

Overview The power subsystem will be responsible for providing power to the Processing, UI, and Sensing subsystems. It will have to step down $12V_{DC}$ provided from a standard AC/DC adapter to $3.3V_{DC}$ for the ESP-32, sensors, and e-ink display. It will also have to step down $12V_{DC}$ to $5V_{DC}$ for the RGB LEDs. This subsystem also needs to protect the circuit from unexpected voltages.

- **Requirement 1**: $5V_{DC}$ step down converter must maintain a 5V output with 5% accuracy

- **Requirement 2**: $3.3V_{DC}$ step down converter must maintain a 3.3V output with 5% accuracy

2.2.2 User Interface

Overview The UI subsystem includes all the ways the user might physically interact with the device. This consists of ways the user will physically control the devices, as well as ways the device can display information.

Outputs The primary way of displaying information will be on a 7.5 inch e-ink display connected to the Processing and Communication subsystem over SPI. Because e-ink displays have a slow refresh rate, a strip of RGB LEDs to the left of the display will be used to communicate information that is needed quickly, such as the task statuses, the currently selected task, relative temperature and humidity, and relative CO_2 levels. What's being displayed on these LEDS will be shown using a separate bank of LEDs under the display. There will also be a speaker that notifies the user of any incoming tasks.

- **Requirement 1**: The e-ink display must be able to completely refresh in under 15 seconds.
- **Requirement 2**: The e-ink display must be able to show at least 12 tasks.
- **Requirement 3**: Each RGB LED needs to be able to show at least 6 colors (Red, Orange, Yellow, Green, Blue, Violet)
- **Requirement 4**: The speaker must be able to produce a sound above 50 dB

Inputs Inputs: The inputs will consist of a rotary encoder, and two buttons. This should allow the user to control the device and do things such as changing the task status or what information is shown on the RGB leds. These will connect to digital inputs on the ESP-32.

- **Requirement 1**: There needs to be at least 2 buttons and one rotary encoder on the device.

2.2.3 Sensing

Overview The sensing subsystem will measure the environment around the device, and send that information via I2C, SPI, or an analog signal to the Processing and Communication subsystem.

eCO_2 sensor The first sensor we will use is an eCO_2 sensor. Normal CO_2 sensors are often bulky, expensive, and noisy, so we chose to use an eCO_2 sensor like the SGP40 which measures the volatile organic compounds in the air, and calculates the approximate CO2 levels based on that. The data will be sent to the Processing and Communication subsystem over I2C

- **Requirement 1**: The eCO_2 sensor must be accurate to within 20% of the actual value

Temperature and Humidity sensor We also want to measure temperature and humidity. Many manufacturers combine these into one sensor, such as the SHT40, which will also communicate with the Processing and Communication subsystem over I^2C.

- **Requirement 1**: The measured temperature needs to be within $\pm 1°C$ of the actual temperature between $10°C$ and $40°C$.

- **Requirement 2**: The measured relative humidity needs to be within ±3% between 25% and 75% relative humidity.

Motion Sensor We plan to use a radar-based motion sensor to detect seat occupancy. We plan to use the RCWL-0516 radar module for this. This sensor has an analog output, which will be read by an ADC on our ESP-32.

- **Requirement 1**: The motion sensor must be able to detect movement within 2 meters of the device.

Ambient Light Sensor We will also include an ambient light sensor to dim the LEDs to match the ambient light level, especially when the lights are off. This can be accomplished by using a photoresistor, which will use another ADC on the ESP-32.

- **Requirement 1**: The ambient light sensor needs to be able to output a value proportional to the actual light intensity.

2.2.4 Webapp and Server

Overview The Webapp and Server subsystem will primarily be used to input and store tasks. The user will be able to type in the task's name and the due date, and that information will automatically be sent to the server. The server will then send all the tasks to the device over WiFi to be displayed. The Website will also need to read the stored environmental data from the server and provide an easy to interpret visualization. We will use React.js to build user interfaces of the simple page web application based on UI components from an established UI Framework. We also will use Node/Express as our server-side web framework.

We will write our own API to retrieve and update data from servers. If time is an issue we will use the established Google Tasks API.

- **Requirement 1**: The server must be able to store at least 12 tasks simultaneously
- **Requirement 2**: The server must be able to store at least 24 hours of sensor data

2.2.5 Processing and Communication

Overview The Processing and Communication subsystem will gather information and disseminate it to the other subsystems. It'll also handle communication to the server over WiFi. It's primary focus will be taking information from the server and sending it to the display in the UI subsystem. It will also be tasked with sending the sensor information and any task updates done through the UI buttons on the device to the server.

- **Requirement 1**: The MCU must be able to receive server information within 2 minutes of data sent
- **Requirement 2**: The MCU must be able to send relevant data to other subsystems within 2 minutes of it being called for

2.3 Tolerance Analysis

The RCWL-0516 radar sensor has a sensitivity of about 7 meters [5]. But the user usually would only be about 0.4 to 1.25 meter away from the sensor on the desk [6]. Generally, the preferred viewing distance is between 0.5 and 1m from the eye to the front surface of the computer screen (about an arm's length). We don't want the motion in the background be detected by the sensor so that we want to set a triggering threshold. Specifically, we want to control the sensitivity so it only goes on when there is very nearby movement, like 0.25 to 1.4 meters range. Except for the distance, another potential issue is that the sensor might mot be able to detect small body movement. For example, the sensor might not detect a very slight body movement so that the sitting time of the user might be miscalculated. We might conduct multiple tests to solve the above issues by changing the value of a certain register (R9 with Factory parameters- 220 ohm) to receive a optimal sensitivity range.

In addition, according to the feedback from other users using the RCWL-0516 radar sensor, having a lot of other electronic devices such as a computer close by reduces reliability. We need to identify the equipments that may interfere the sensor and move the sensor away from the interfering equipments.

3 Ethics and Safety

The user of our project will be directly involved with the operation of the device; We must adhere to the guidelines from Section I.1 of the IEEE Code of Ethics that "to hold paramount the safety, health, and welfare of the public..." [7] and the safety regulation in Sections 1910.302 - Electric utilization systems by Occupational Safety and Health Administration [8] , so, it is important that we must ensure a safe and reliable product. There are several components in our product that, if mishandled or constructed poorly, might pose a risk to the user's safety. The power supply and circuits are some of the most important components to consider. The risks associated with the unstable power supply will be reduced by a rigorous power electronics design, and we would make sure that the users use our product in a safe environment.

Furthermore, we must consider that our product will be available to everyone in accordance with the Section II of IEEE Code of Ethics [7], and that our users can equally operate our device. The mission of our project is to give people a better way to monitor their daily tasks and their working environment. We welcome everyone to use our product and we hope that our users could improve their productivity as well as efficiency with our product.