ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Gesture Base Turn Signaling System

Team #20

SULTAN ALNUAIMI
(saltana2@illinois.edu)
EDAN ELAZAR
(eelazar@illinois.edu)
KAYLAN WANG
(kaylanw4@illinois.edu)

TA: Sanjana Pingali

May 1, 2024

Abstract

This document provides an outline and LaTeX template for report formatting in Senior Design. This document does not teach you what to include, or how to use LaTeX. Assumes a workable level of LaTeX proficiency.

Contents

1 Introduction

1.1 Problem Statement

Cyclists, skateboarders, and scooter riders often face challenges in signaling their intentions to drivers, especially in low-light conditions. According to the CDC, 1,000 cyclists die and 130,000 are injured every year on the road in the United States (cite here). These numbers don't include other riders sharing the road on things like skateboards and scooters. There are many interventions in place to prevent these accidents, such as fluorescent or retro-reflective clothing, or active lighting on the bicycle (required by law in most states) (cite here), but the traditional method of using hand signals is not always visible or practical, particularly at night or during adverse weather conditions. This lack of clear communication can lead to dangerous situations on the road, as other motorists may fail to recognize the cyclist's intended maneuvers, or if an accident occurs.

1.2 Proposed Solution

To address this issue, we propose the development of a gesture recognition-based turn signaling system for cyclists and scooter riders. This system will utilize IMUs containing 9 degrees of freedom (3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer), integrated into a jacket. Then the data from the sensors will be processed to identify the specific arm gesture made by the rider and activate corresponding LED signals. For example, if the rider extends their arm straight to the left, the left turn signal is activated, or if the rider indicates a stop (arm out and forearm down), then the brake light is activated, and so on. Additionally, the sensors will be able to detect when the rider has had an accident or a crash, and activate a hazard signal on the LEDs.

We propose placing an IMU below the on the left wrist, and another on the waist. The microprocessor will then receive and process the data from the IMU, and determine what kind of movement has been made in real time. Then, depending on the movement, it will output a specific signal to the LEDs to display on the front and back of the wearable.

The final product achieved all 3 high level requirements. The device is able to correctly detect predefined arm gestures (raising right/left arm for turn signals, forearm down for slowing down) with an accuracy of 90%. In addition, the device is able to correctly map the arm gestures into the different indications on the LEDs. The turn signal will be indicated by either the left or right side LED flashing red, while the brake/slow down signal will be indicated by all LEDs turning red. A crash or accident activates the hazard light, indicated by all LEDs flashing red. Lastly, the turn signals, brake lights, and hazard signals were all visible and easily identifiable from a distance of at least 250 feet to ensure that they are clearly visible at both day and night. All of these functionalities ensure that hand gestures are more visible to drivers and other people, thereby increasing the safety of the rider.

1.3 Visual Aid

Figure 1 is a mockup of the wearable device that we created. The device will be powered by a battery and will have 2 IMUs to detect the arm gestures. The system will be controlled by an ESP-32 microcontroller that will process the data from the IMUs and send a signal to the LEDs to turn on when the arm gestures are detected.

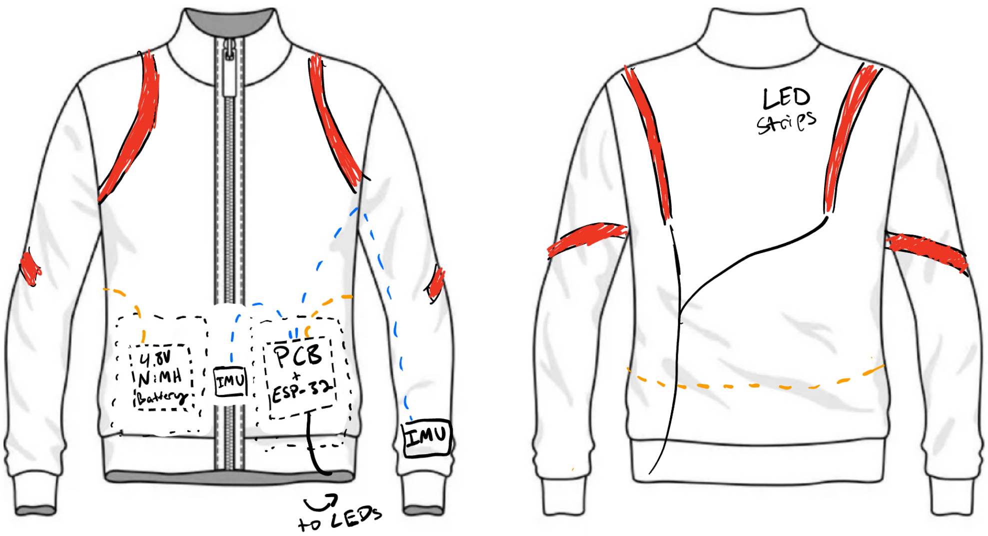

Figure 1: Visual Aid mockup of the wearable [1]

1.4 Block Diagram

As can be seen from the block diagram in figure 2, the device is split into 4 subsystems: Sensors, Control, Power, and LEDs. The power subsystem provides 4.8V to the LEDs through a BJT that gets a signal from the ESP-32. The Control Subsystem and the Sensor Subsystem receive 3.3 volts from the battery through a voltage regulator. The Control Subsystem receives real-time data from the 2 IMUs through I2C, processes the data, then sends a signal to the gate to turn the LEDs on when the IMU data is within a certain range.

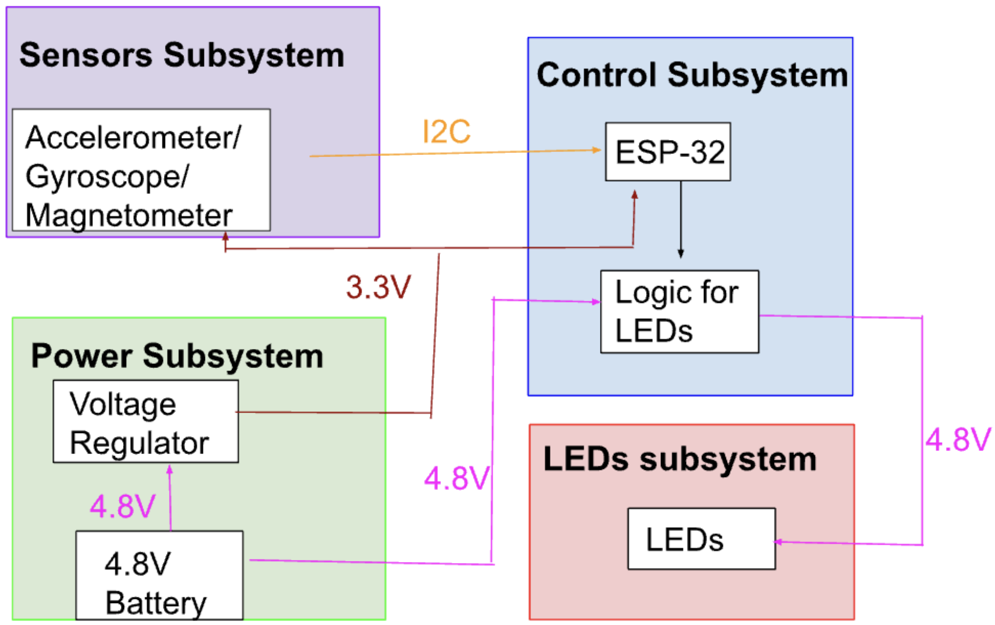

Figure 2: Block Diagram of the system

2 Design

2.1 Design Procedure

Our design consists of four different subsystems, each with their own functions and requirements. At the center of the system is the control subsystem consisting of the ESP-32 microcontroller along with some simple logic for controlling the LEDs. The goal of this subsystem is to establish communication with the sensors, receive the necessary data from them, determine which hand gesture is being activated, and activate the corresponding LEDs. The main design decisions that we had to make for the control subsystem were software related as the circuit is quite simple. We decided to simplify the algorithm for detecting the gestures due to some time constraints with the PCB order waves along with a problematic ESP-32 DevKit. We originally wanted to devise an algorithm that utilized all three sensors on the IMU, but ultimately decided that it would be too difficult to program, and we found that the magnetometer was all that was necessary to accurately detect the gestures.

The original plan was to use an IMU on each arm, but this approach proved to be difficult in terms of programming. Instead, we used one on the waist, which ensures that one of the IMUs is always stable. Then, we just take the difference in angles between the two IMUs, and set ranges for each hand gesture. We also had to discard our PCBs for our IMUs as we were unsuccessful in soldering the small IMUs onto the board. Therefore we decided to continue with the breakout boards.

For the power subsystem, we decided that the system should last at least one hour. We determined the current required to power the system to be 1864.2A total, so we needed a battery that could supply 5V with a capacity larger than 1864mAh, in addition to being rechargeable. The battery we used for the power subsystem was a 3000mAh 4.8V NiMH battery that provided ample voltage and current to power the LEDs. During the Design review, using an LED Driver was recommended to get a constant current to the LEDs. We researched several drivers and found one that fits the requirements. However, we also tested the LEDs without the driver, and the brightness was sufficient. The plan was to add the driver to get slightly brighter LEDs, but the driver only works with a PCB, and we were unable to get the LED drivers to function from the first and third order PCBs. Ultimately we decided to discard the LED driver in order to simplify the PCB and increase the odds of it functioning properly. We decided that a working PCB was more important than a slightly brighter LED.

2.2 Design Details

The LM317 datasheet [2] provides us with equation (1) which relates the output voltage to the resistances of R_1 and R_2 when using the schematic in figure 3

$$V_0 = V_{REF}\left(1 + \frac{R_2}{R_1}\right) + I_{ADJ} \times R_2 \quad [2] \tag{1}$$

Where R_1 and R_2 control the output voltage of the linear voltage regulator, and V_{REF} and I_{ADJ} are given to be 1.25V and 50μ A respectively. Using this equation, we were able to find the resistor values necessary to achieve an output of 3.3V, which are $R_1 = 314.2\Omega$ and $R_2 = 512\Omega$. Note that we also have a linear voltage regulator which outputs 1.8V, which we added with the intent of making our own IMU PCB, but we ended up using a breakout board which takes in 3.3V.

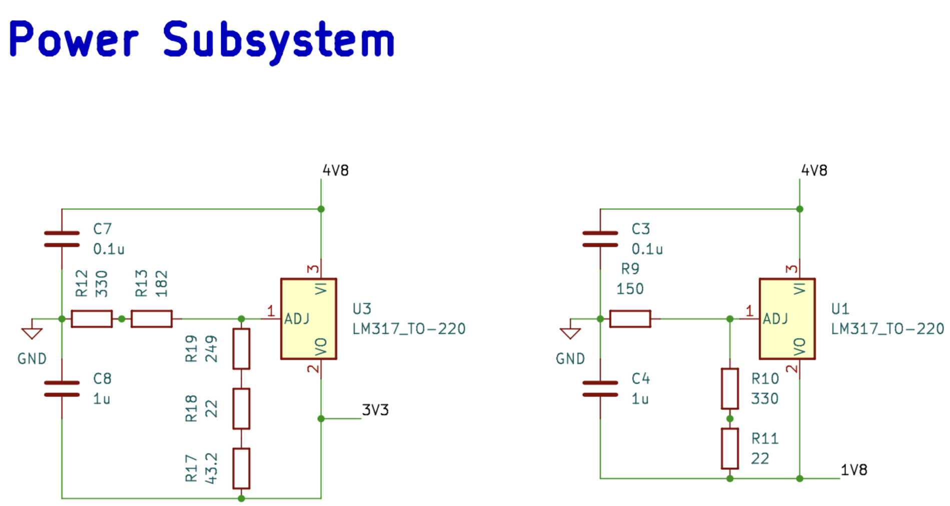

Figure 3: Schematic of the power subsystem

The control unit in figure 4 consists of the ESP32-S3 which we use to control our entire system. We needed to add a programming circuit for it and ensure it has the proper connections to all the other subsystems.

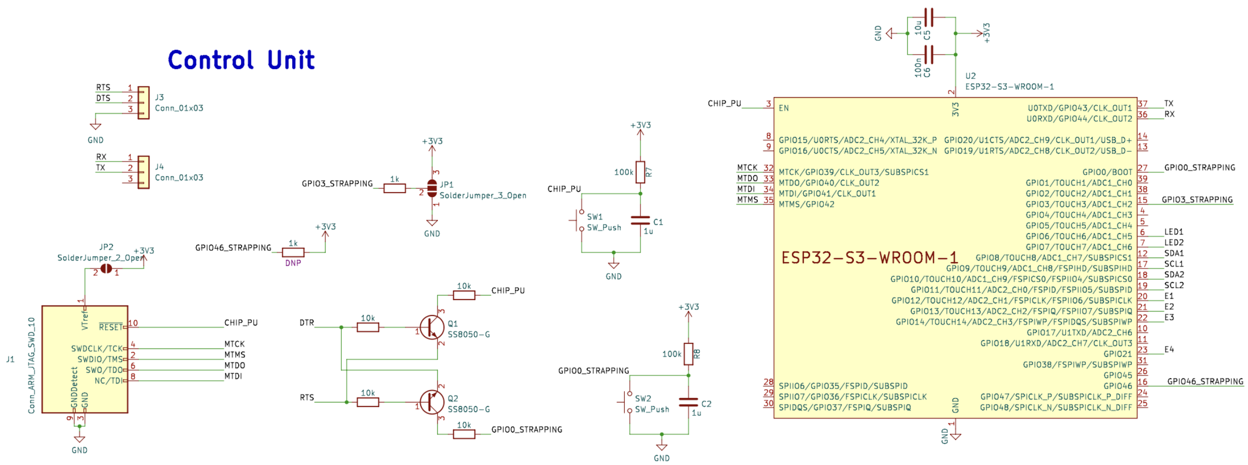

Figure 4: Schematic of the control subsystem

Our LED subsystem in figure 5 was intended to use an LED driver, so we designed a circuit for it which we have on our pcb. Due to time constraints however we did not have enough time to test it and ended up hotwiring the LEDs to bypass the LED driver.

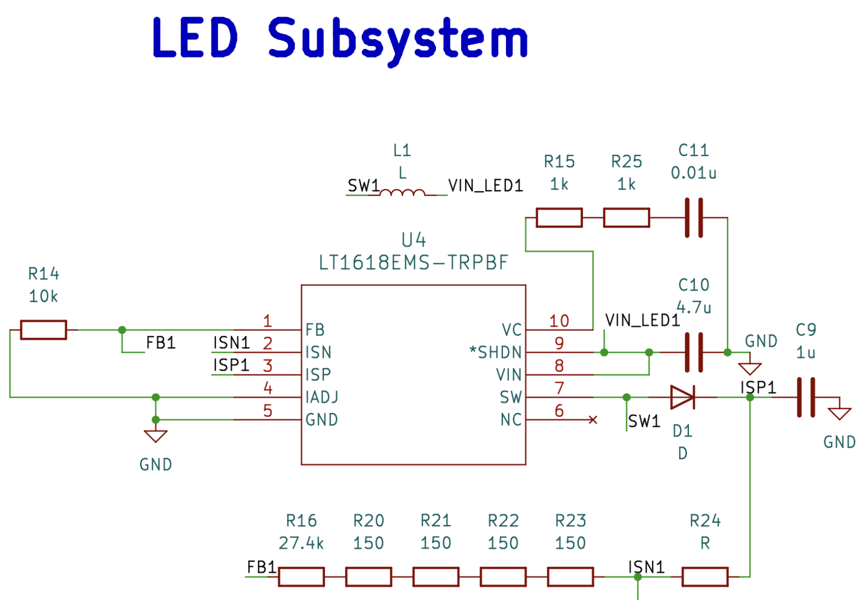

Figure 5: Schematic of the LED subsystem

The sensor subsystem in figure 6 consists of the ICM-20948 IMU which we use to detect the hand gestures. This schematic shows the suggested connections [3] for the IMU to communicate via I2C with the ESP32. Unfortunately, we were unable to get the IMU to work with the ESP32, so we had to use the breakout board which has a built-in voltage regulator and I2C communication.

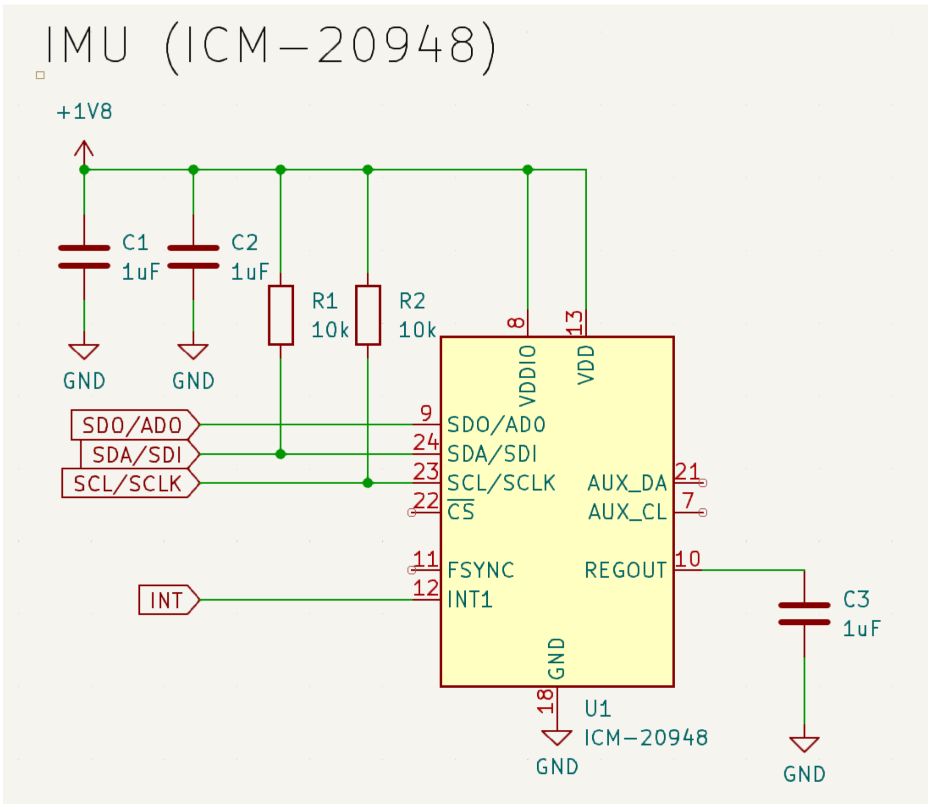

Figure 6: Schematic of the sensor subsystem

In order to power the system, we used a 3000mAh 4.8V NiMH battery. As shown in table 1, our total current draw is 1864.2 mA so our estimated battery life of the system is $3000 \text{ mAh}/1864.2 \text{ mA} = 1.61$ hours.

Table 1: Current Draw of System Components

Description	**Value**
ESP-32 worst case current draw	355 mA
2x ICM-20948 IMU	4.6 mA each = 9.2 mA total
2x LED strip	750 mA each = 1500 mA total
Total current draw	1864.2 mA

Heat tolerance for battery

- Total Internal Impedance = $4\,\mathrm{m}\Omega \times 4 = 16\,\mathrm{m}\Omega$.
- $I = 2\,\mathrm{A}$
- Time = 3600 seconds
- Mass = 0.228 kg for 4 cells
- Specific Heat Capacity: $900\,\mathrm{J/kg^\circ C}$
- Total Heat Generated = $I^2 \times R \times$ time $= 230.4\,\mathrm{J}$
- Change in temp = $\frac{\text{Heat}}{\text{mass}\times\text{specific heat capacity}} = 1.123°\mathrm{C}$

Assuming an ambient temperature of $25°$C, the temperature of the battery will be around $26.123°$C after one hour of operation. This is well within the operating temperature of the battery, so we can conclude that the battery will not overheat during operation.

During testing we found that after powering the system for one hour in an ambient temperature of $22°$C, the battery was $23.5°$C. This confirms our calculations that the battery will not overheat during operation.

3 Verification

3.1 Control Subsystem

3.2 Power Subsystem

3.3 Sensor Subsystem

3.4 LED Subsystem

4 Costs & Schedule

4.1 Costs

The total cost of parts will be approximately $\$234.96$ and the expected labor costs are calculated as $\$40$/hrs $* 2.5 * 60$hrs $= \$6,000$. This will be applied to all 3 team members so the total labor cost is $\$6,000 * 3 = \$18,000$. This comes out to a total cost of $\$18,234.96$.

Table 2: Bill of Materials

Description	**Manufacturer**	**Quantity**	**Price/unit**	**Total Price**	**Link**
velcro strips	VELCRO Brand	1	19.62	19.62	Link
ESP32	Espressif Systems	1	7.69	7.69	Link
LED strips	Aclorol	2	5.99	11.98	Link
Battery	Panasonic - BSG	1	40.50	40.50	Link
Voltage Regulator	STMicroElectronics	1	6.95	6.95	Link
IMU breakout board	Adafruit	3	18.50	55.50	Link
0.065 Ohm resistors	TFT Corp	4	0.55	2.20	Link
LED Driver	Analog Devices Inc	3	5.70	17.10	Link
1A Diode	onsemi	3	0.50	1.50	Link
Inductors	SAC Inc	3	0.82	2.46	Link
ICM-20948 IMU	TDK InvenSense	10	6.91	69.08	Link
Components Total	N/A	N/A	N/A	234.96	
Labor	N/A	3	40.00	18 000.00	
Total	N/A	N/A	N/A	18 234.96	

4.2 Schedule

5 Conclusion

5.1 Accomplishments

5.2 Uncertainties

5.3 Future Work

5.4 Ethics Considerations

References

[1] VectorStock, "Front back and side views blank jacket Royalty Free Vector," 2024, [Online; accessed 21-February-2024]. [Online]. Available: https://www.vectorstock.com/royalty-free-vector/front-back-and-side-views-blank-jacket-vector-4239549

[2] Texas Instruments, *LM317 3-Terminal Adjustable Regulator*, 2023, accessed: Feb. 22, 2024. [Online]. Available: https://www.ti.com/lit/ds/slvs044y/slvs044y.pdf

[3] TDK InvenSense, "DS-000189-ICM-20948 v1.3," https://invensense.tdk.com/wp-content/uploads/2016/06/DS-000189-ICM-20948-v1.3.pdf, 2016, accessed: [Online; accessed 22-March-2024].

Appendix A Example Appendix

An appendix can go here! Make sure you use the `\label{appendix:a}` above so that you can reference this section in your document.