

ECE 445
Spring 2022
Final Report

Pacer Runner

Team 12
Aditya Singla (asingla4)
Mustafa Jamal (mustafa6)
Carlos Sanchez Alberti (carloss7)
TA: Daniel Ahn

Abstract

Pacer Runner is a wearable device attached to a running vest which helps our users get instantaneous feedback on their current pace. We utilize a GPS module to calculate speed and output that number to an LCD screen. Additionally, the device projects a line laser on the ground in front of the runner which changes its angle of projection based on a comparison between the runner's target and actual pace. Details about our project are discussed in this report below as well as the technical specifications.

Table of Contents

1. Introduction	4
1.1 Problem & Solution Overview	4
1.2 Visual Aid	4
1.3 High Level Requirements	5
1.4 Block Diagram and Descriptions	6
2. Design	8
2.1 Design Procedure	8
2.2 Design Details	9
2.2.1 Power	9
2.2.2 Angle Control	10
2.2.3 Control	10
2.2.4 User Interface	10
2.2.5 Tolerance Analysis	11
2.2.5.1 GPS Accuracy Tolerance	11
2.2.5.2 Angle-Control Tolerance	11
3. Verification	14
3.1 Power Subsystem	14
3.2 User interface	14
3.3 Control Subsystem	15
3.4 Angle Control Subsystem	15
4. Costs and Schedule	16
4.1 Schedule	16
4.2 Labor	16
4.3 Parts	17
4.4 Grand Total	17
5. Conclusion	18
5.1 Accomplishments	18
5.2 Uncertainties	18
5.3 Ethical Considerations	18
5.4 Future Work	19
Appendix A: PCB Design	20
Appendix B: Requirements and Verification Tables	21
References	24

1. Introduction

1.1 Problem & Solution Overview

Beginner runners know how difficult it is to keep a constant pace during a long run. Using a smartwatch or a smartphone, they can measure their pace and heart rate. However, an issue associated with these devices is that the runner doesn't receive any feedback until their training has ended, resulting in the runner not having any convenient way to maintain their pace throughout the run. Thus, they need a solution which can provide them with feedback instantly.

The solution involves a laser device that will be worn by the runner on their chest which would be capable of tracking their pace and giving real-time feedback on their performance. To do this, we will design a line laser device that changes the projection angle depending on whether the runner's pace is too fast, too slow or ideal. Thus, the pointer will move further if the runner decreases their pace and closer when they increase it.

1.2 Visual Aid

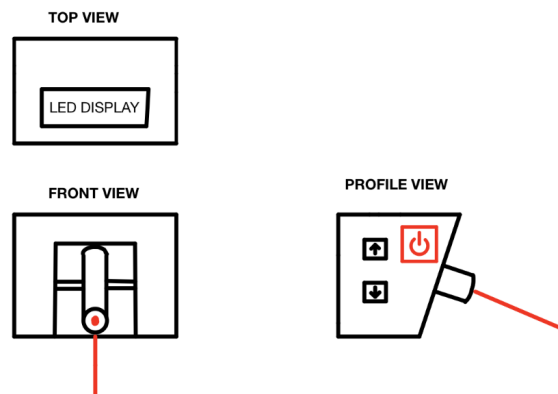


Figure 1: Orthographic Projection

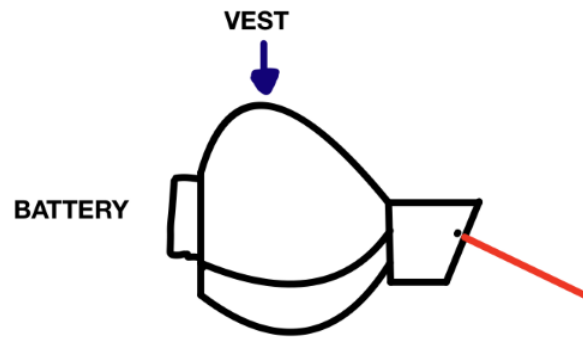


Figure 2: Vest Design

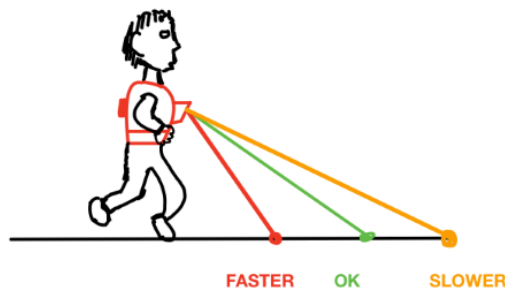


Figure 3: Working Illustration

1.3 High Level Requirements

- The servo-motor changes the angle of the laser in response to the runner's change in pace. (as shown in *Figure 3*)
 - ◆ There will be three possible positions with 0 degrees being vertically down (Refer to 2.6.2 *Angle-Control Tolerance*):
 - slower than target pace: servo-motor at 56 degrees
 - ideal pace: servo-motor at 45 degrees
 - faster than target pace: servo-motor at 27 degrees.
 - The GPS module will calculate the runner's pace accurate to 1 mi/hr.
 - The physical input buttons change the target pace, and the LCD display output will reflect these changes. The changes will be:
 - ◆ Increase pace button should increase the target pace by 0.5 mi/hr.
 - ◆ Decrease pace button should decrease the target pace by 0.5 mi/hr.
- The LCD display should always show the target pace.

1.4 Block Diagram and Descriptions

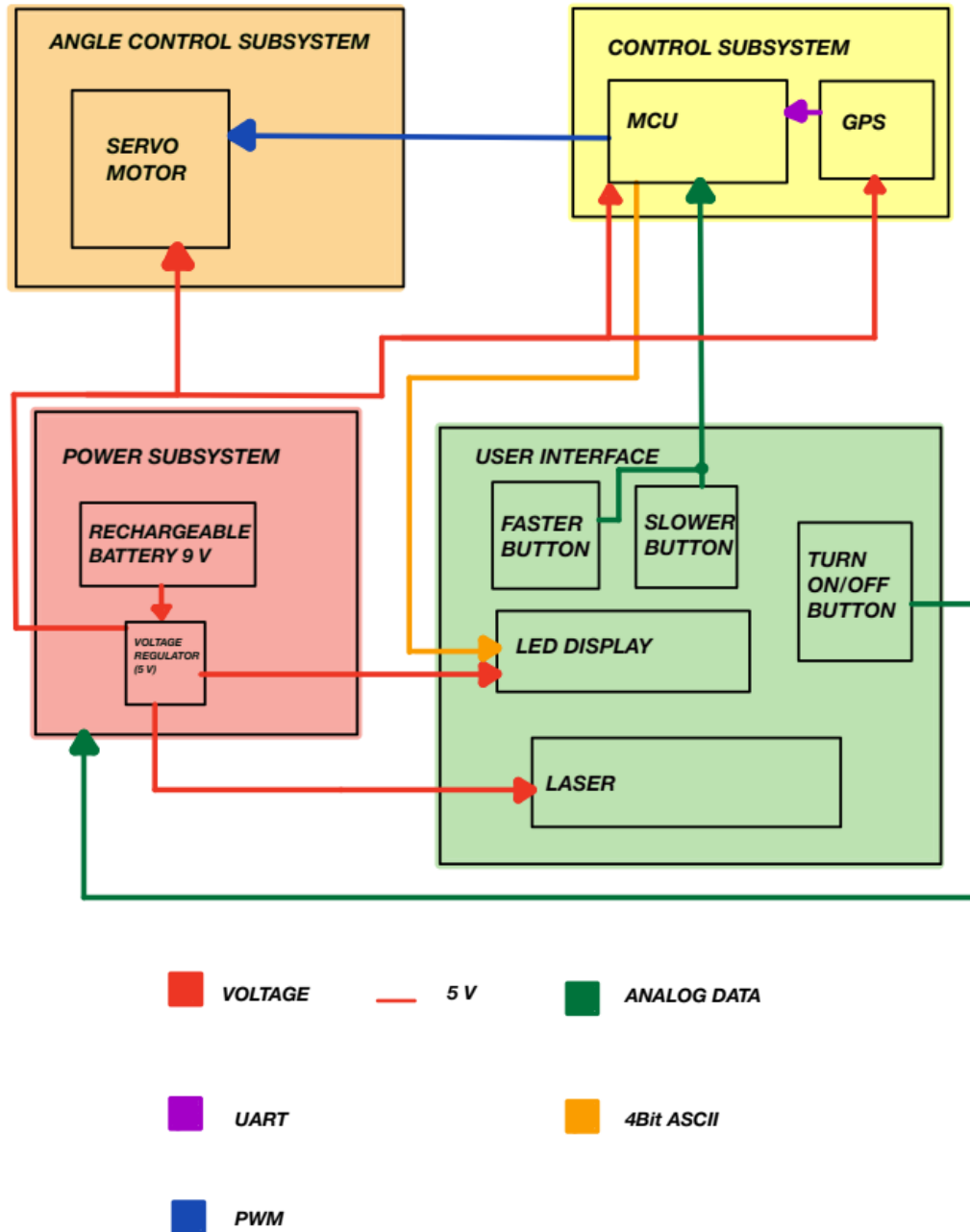


Figure 4: Block Diagram

Our Block Diagram consists of 4 separate subsystems namely Power, Control, Angle-Control and User Interface. The power subsystem provides energy to the components based on their individual needs and requirements. The voltage from the main battery is regulated by one

regulator, and then provided to all the other subsystems. The purpose of the Angle-Control subsystem is to rotate the servo-motor (on which the line laser is mounted) to project the line laser in different positions based on the pace of the runner. The Control subsystem receives power from the Power subsystem, processes input data from the User Interface subsystem, and outputs data to the Angle Control subsystem as well as the User Interface subsystem. It basically brings everything together and allows us to program the Microcontroller and the GPS. Additionally we have the User-Interface which controls the LCD, Line-Laser and the physical buttons. The goal for this subsystem is to enable interaction between the user and the device to enable functionality and ease of use. Lastly the Angle Control subsystem which takes in feedback from the microcontroller based on what speed the runner is going at and outputs a certain angle between 27/45/56 degrees.

2. Design

2.1 Design Procedure

In our design procedure our main focus was firstly, to make sure our product worked and secondly to make sure it is comfortable and usable by the runner. We tried to provide a well packaged product which provided accurate and effective feedback to the runner and help him/her in their run. We took a certain approach to our design but there are alternative possibilities which are discussed in more detail down below.

For our power subsystem we chose a 9 V battery to provide enough voltage to our laser light. From testing we realized that in order for it to be visible in ideal laboratory conditions a minimum of 6 V was required and 9 V battery was the most convenient. An alternative for this would be a battery of smaller voltage such as 6 V which could have worked as well.

The Control subsystem utilizes the ATmega 328 and a GPS module to send and receive signals based on user feedback. We initially tried using the ATmega 309 but quickly realized that it was not compatible with our PCB and had to switch to the 328. For our GPS module we used an external one connected via pins to the PCB but an alternative could be a surface mounted GPS which might save space and be more accurate.

The Angle-Control subsystem is where our project did not work as intended and our servo motor was not responding to the microcontroller. We could potentially explore two alternatives firstly by using a higher quality servo motor which does not have problems with electromagnetic interference. Another possibility is to completely replace the servo motor and instead have led lights of three different colors to indicate speed.

Lastly, the User Interface subsystem is where we use our LCD, physical buttons and line laser. Alternative approaches that we could have conducted for this subsystem would be using a brighter LCD as sometimes in intense sunlight the LCD was not as visible as we would have wanted. Additionally LCD would also flicker when viewed from certain angles, hence a higher quality LCD would solve these problems. Instead of the physical buttons we could use a slider for the pace making it easier for the runner to increase/decrease the target pace.

2.2 Design Details

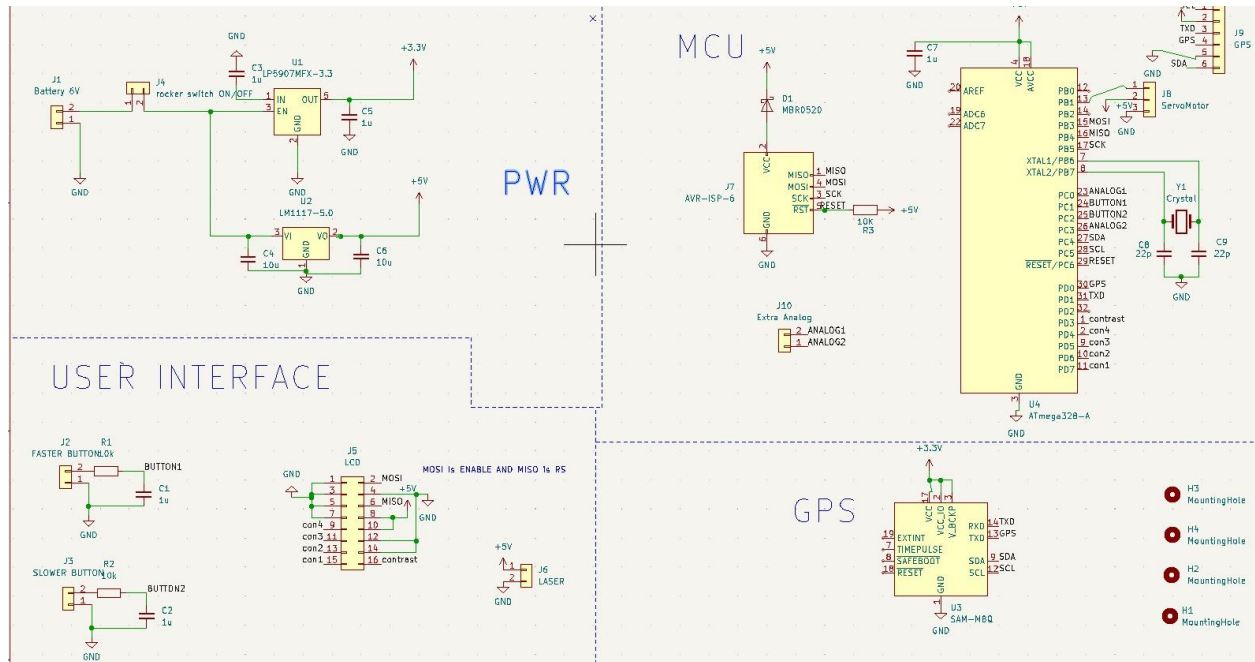


Figure 5: PCB Schematics

2.2.1 Power

The power subsystem provides energy to the components based on their individual needs and requirements. The voltage from the main battery is regulated by one of two regulators, and then provided to all the other subsystems.

To ensure that our device operates to its utmost potential we have to provide an adequate source of power and a voltage regulator. We will be using a rechargeable 9 V battery. The battery is extremely light, and should not be uncomfortable to wear. We chose this battery because our two main components, the laser and the servo-motor, both require a minimum voltage of 5 V for ideal functionality.

We would also require one voltage regulator to bring the voltage down to 5 V for different components in our device. We will use the LM1117 for the 5 V regulator.

2.2.2 Angle Control

The purpose of this subsystem is to rotate the servo-motor (on which the line laser is mounted) to project the line laser in different positions based on the pace of the runner.

The motor in this subsystem gets power from the Power subsystem and receives commands to rotate from the Control subsystem. Lastly, it is connected to the User Interface as the line laser is mounted on this subsystem.

This subsystem consists of a servo-motor (HS-311) which has an operating voltage of 4.8 to 6.0V. This motor has a maximum rotation of 202 degrees which meets our requirements. Based on the pace of the runner, the motor toggles between 27 degrees, 45 degrees, and 56 degrees. (Refer to *2.2.5 Angle Control Tolerance*)

2.2.3 Control

The Control subsystem will:

- take in the target pace from the User Interface
- calculate the runner's pace using a GPS module
- compare the runner's pace with the target pace
- and finally, adjust the servo-motor's angle based on this information.

The control receives power from the Power subsystem, processes input data from the User Interface subsystem, and outputs data to the Angle Control subsystem as well as the User Interface subsystem.

This subsystem consists of two main components:

- The microcontroller (ATMEGA328P) allows us to link and control various parts of our project.
- The GPS module (Beitan BN-880) will be used to calculate the runner's pace.

2.2.4 User Interface

The User Interface subsystem allows the user to interact with the device.

It receives inputs that are sent to, and processed, by the Control subsystem. A component in this subsystem, the line laser, is mounted on the Angle Control subsystem. Finally, it also receives output data (for the LCD display) from the Control subsystem.

This subsystem consists of three components:

- the LCD display,
- the physical buttons (turn on/off button, increase target pace button, decrease target pace button),
- the line laser.

2.2.5 Tolerance Analysis

2.2.5.1 GPS Accuracy Tolerance

For this project, the component which poses the highest risk of failure is the GPS system. This project requires that the GPS module can be used to successfully compute the runner's pace at any given moment. However, knowing the current limitations of a GPS module (i.e. **within 4.9 m**)[1], we will only be able to calculate the runner's location every ~5 m.

Now let us make some assumptions to calculate how to navigate this limitation.

Average speed of runner $\geq 2 \text{ m/s}$

GPS location updates every 5 m

Time between GPS updates $\leq 5/2 = 2.5 \text{ s}$

So we need to update our laser's position every 3 s

By moving to this non-instantaneous approach, we can navigate this issue in a practical manner. It is also more reasonable for the project overall, since we need not concern ourselves about overloading the motor in constant changes.

Another consideration regarding the GPS is how long it takes the module to ping off a satellite and be available to use. Based on this GPS module datasheet, the hot/cold starts can take anywhere from 1s - 32s, which means our project will require at least **45s** after turning on before it can be used.[2] This is something we need to consider, but a 45s start time is **not impractical**.

2.2.5.2 Angle-Control Tolerance

1. Glossary

- angle: the angle the servo-motor (mounted with the laser) must be set to, considering 0 degrees to be facing straight down.
In other words, at 0 degrees, the line laser will point straight down, and at 90 degrees, it points straight ahead
- laser horizontal: it means the horizontal distance between the runner and the laser line projection.
In other words, the distance between the runner's feet and the laser line. (projected by our laser)
- actual pace: the pace at which the runner is currently running at calculated by our device. In units of miles/hour or mi/hr.

- target pace: the pace at which the runner wants to run at, inputted by them through the user-interface subsystem. In units of miles/hour or mi/hr.

2. Introduction

Another important part of this project is the angle-control subsystem, and the exact angles we need to consider for each position.

The three different situations a runner can be in are:

1. **slow**: *actual pace* < *target pace* - 0.5 mi/hr
2. **fast**: *actual pace* > *target pace* + 0.5 mi/hr
3. **ideal**: *actual pace* is within 1 mi/hr of the *target pace*

Henceforth, we will just be referring to the above positions as *slow*, *fast*, *ideal*.

3. Requirements

There are two requirements for the angle-control subsystem:

- A. The line laser can be seen clearly by the runner.
- B. The actual length difference between the laser line in the three positions should be equidistant, so as to not be confusing for the runner.

4. Calculation

Let us begin this calculation by making a few assumptions, along with their necessary justifications.

- A. the height of the device from the ground is x m
- B. the distance between consecutive positions should also be $x/2$ m. (*fulfilling requirement B*)
 - a. If we take the average height of a runner to be 160 cm, then we can assume x to be ~125 cm.
 - b. Now based on our assumption of the line laser, the three positions it can be in will be:
 - i. 0.625 m from the runner ($1x/2$)
 - ii. 1.250 m from the runner ($2x/2$)
 - iii. 1.875 m from the runner ($3x/2$)
 - c. These three positions for the line laser will be comfortable for the runner to view, thus justifying our assumption. (*fulfilling requirement A*)
- C. the three positions should be as illustrated below

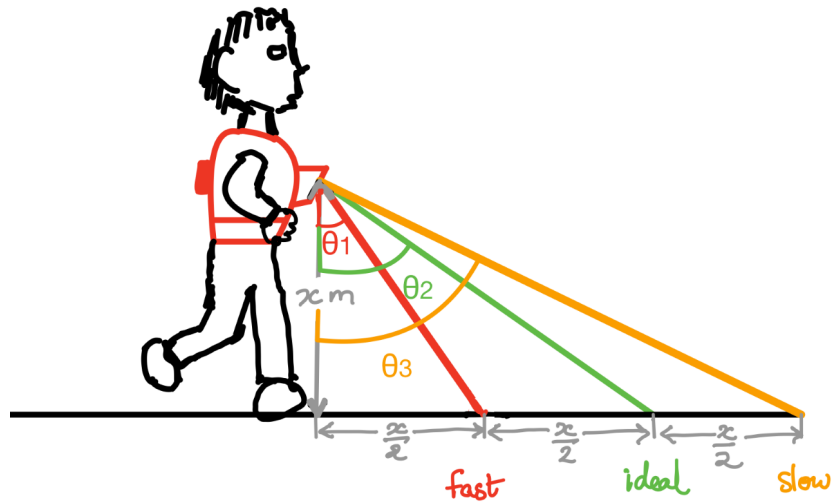


Figure 6: Sketch

Let us find the angles: $\theta_1, \theta_2, \theta_3$

$$\theta_1 = \tan^{-1}\left(\frac{x/2}{x}\right) = \tan^{-1}\left(\frac{1}{2}\right) = 26.56^\circ \approx 27^\circ$$

$$\theta_2 = \tan^{-1}\left(\frac{x}{x}\right) = \tan^{-1}(1) = 45^\circ$$

$$\theta_3 = \tan^{-1}\left(\frac{3x/2}{x}\right) = \tan^{-1}\left(\frac{3}{2}\right) = 56.31^\circ \approx 56^\circ$$

5. Findings

Thus, we should set up our servo motor such that:

- **fast:** servo-motor should be set at 27 degrees,
- **ideal:** servo-motor should be set at 45 degrees,
- **slow:** servo-motor should be set at 56 degrees.

This requires us to buy a servo-motor which only needs 60 degrees of motion.

3. Verification

3.1 Power Subsystem

In order to verify the Power subsystem, we connected the two probes from the Keysight 34461A Multimeter to the regulator on the PCB. Based on the Power Subsystem R&V table (Appendix B), we must check that the voltage provided by the output of the 5V voltage regulator (*LM1117DTX-5.0*) is $5 \pm 0.5V$. To do that we have to measure the voltage between the ground and the output pin of the regulator. We can see the pictorial verification in Figure 7.

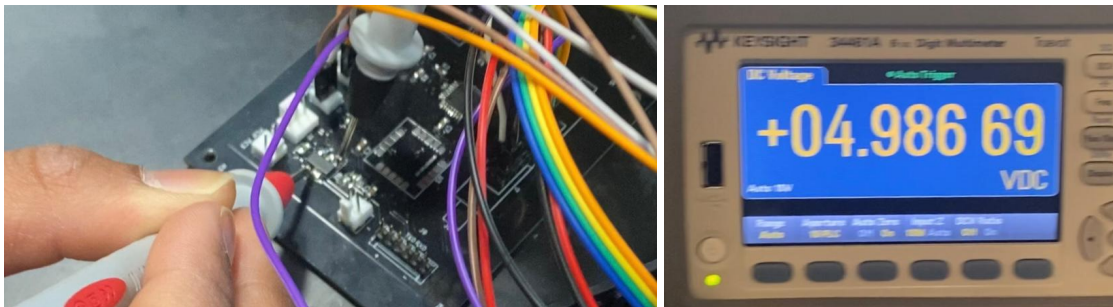


Figure 7: 5V regulator verification

3.2 User interface

The next subsystem that we have to verify is the User Interface Subsystem. This subsystem consists of two increase/decrease target pace buttons, a turn ON/OFF switch, the laser, and the LCD. It is important to verify the correct working of all the parts since they are important for the user to have a good experience with our device. To verify this subsystem, we are also going to follow the R&V tables that can be found in Appendix B.

First of all, we started by verifying the LCD. We verified that the LCD is able to display the target pace and the actual pace on the screen. We used a breadboard and an Arduino UNO to build and test the circuit, after which we implemented the same circuit on our PCB.

The second element to be verified was the buttons. We added the buttons to the breadboard circuit we used for verifying the LCD and programmed the Arduino such that the buttons changed the target pace by ± 0.5 . Both buttons were functional, and thus verified.

Thirdly, we tested the laser visibility in lab conditions. We connected the laser wires to a voltage source. For this experiment, we compared two lasers, one was a green line laser and the other one was a red line laser. Since both had a region of operation of 4.5V to 6V, we started by setting the output voltage to 4.5V, and then we increased this voltage to 6V. During these tests we concluded that the green laser was brighter. Thus we settled on the green line laser.

And lastly, we checked the ON/OFF rocker switch. This element is crucial for the safety of the device since having a kill-switch was an important part of our device. We tested it by building a simple circuit that was connected to this switch. We then verified that we were able to turn the circuit on/off with the rocker switch.

3.3 Control Subsystem

The Control Subsystem is the heart of our project, so we developed several tests to check the correct operation. These tests are described in the Control Subsystem R&V tables in Appendix B.

The first verification test was for the microcontroller. After soldering the microcontroller to the PCB, we connected the ISP port to an Arduino UNO, and flashed the MCU. We then verified that the microcontroller was able to interact with the rest of the components such as the LCD. We did this by programming our MCU with the same code used in verifying the LCD. Figure 8 illustrates this verification.



Figure 8: Correct operation of the MCU

Next, we verified the GPS Module with our device. We did this by uploading a program which calculated the pace of the user, and then displayed it on the LCD. We used a fitness tracker application called Strava in conjunction with our device. We covered a known distance in a set amount of time while tracking our walk with both Strava and our device. We noted the pace reflected on the LCD of our device at set intervals, and compared it with the same shown on Strava. Throughout the walk we had a constant pace of 2.5 mi/hr. Both Strava and our device reflected the same. Thus, we conclude that our GPS was accurate.

3.4 Angle Control Subsystem

This subsystem is composed of the servo-motor that moves to different positions depending on the pace of the runner. We needed to verify that the servo can be moved based on code. We programmed an Arduino UNO and manually entered different angles onto the Serial Monitor. We verified that the servo-motor moved to the angles inputted.

4. Costs and Schedule

4.1 Schedule

Table 1: Schedule

Week	Aditya	Mustafa	Carlos
2/28	PCB Design/ Design Review	PCB Design/ Design Review	PCB Design/ Design Review
3/7	PCB Design	PCB Design	PCB Design
3/14	Break	Break	Break
3/21	GPS Code	PCB Design (2nd)	PCB Design (2nd)
3/28	GPS Testing	GPS Testing	Solder PCB
4/4	Test Subsystems	Test Subsystems	Test Subsystems
4/11	Assemble Prototype	Assemble Prototype	Assemble Prototype
4/18	Full Device Test / Mock Demo	Full Device Test / Mock Demo	Full Device Test / Mock Demo
4/25	Demo	Demo	Demo
5/2	Presentation/ Final Paper	Presentation/ Final Paper	Presentation/ Final Paper

4.2 Labor

The average salary of a Computer Engineer graduate from UIUC is \$ 96,992. [3]

Based on that figure, if we consider there to be 261 working days in the year, and a normal work day to be eight hours, then the average hourly salary computes to: \$ 46.45.

Number of workers = 3

Hours per week = 10

Weeks required = 10

Student labor costs = $2.5 \times (\#workers) \times (hrs/ week) \times (weeks required) \times (avg salary)$
= $2.5 \times (3) \times (10) \times (10) \times (46.45)$
= **\$ 34,837.5**

Machine shop labor = \$70/hr x 5
= \$ 350

4.3 Parts

Table 2: Costs

S.No.	Part	Unit Cost	Quantity	Total
2.	Voltage Regulator- <i>LM1117DTX-5.0/NOPB</i>	\$ 0.78	1	\$ 0.78
3.	Microcontroller - <i>ATMEGA328P</i>	\$ 5.64	1	\$ 5.64
4.	Servo Motor - <i>HS-311</i>	\$ 13.49	1	\$ 13.49
5.	GPS Module - <i>Beitian BN-880</i>	\$ 31.50	1	\$ 31.50
6.	LCD Display - <i>Adafruit RGB Backlight 16x2</i>	\$ 8.58	1	\$ 8.58
7.	Push Buttons - <i>R-667845</i>	\$ 0.42	3	\$ 1.26
8.	Rocker Switch - <i>RF1-1A-DC-2-R-1</i>	\$ 1.17	1	\$ 1.17
9.	Laser- <i>VLM-520-28</i>	\$ 24.53	1	\$ 24.53
10.	Battery - <i>Energizer 9V Battery</i>	\$ 12.49	1	\$ 12.49
	Total			\$ 94.78

4.4 Grand Total

The grand total of the project comes out to **\$35,283**, with most of the cost in labor, and only \$ 94.78 to build the prototype.

5. Conclusion

5.1 Accomplishments

By the time of the demonstration, we were able to completely construct our whole device. The device was portable and durable, two important characteristics that a fitness related device must have.

The most complex part of our project: the GPS module used to calculate pace works flawlessly. This allows our device to accurately calculate the runner's speed upto an accuracy of 0.5 mi / hr. Our device also allows the user to interact with it easily, with responsive pace-changing buttons and a well-placed LCD display which gives feedback to the runner.

Additionally, we were able to accomplish most of our high level requirements which we set at the start of the semester. We also gained a lot of experience and expertise on PCB design and working with hardware throughout this project as well.

5.2 Uncertainties

Our project included only one failure: the angle-control subsystem. The servo-motor attached to the device did not work in conjunction with the rest of the subsystems. It worked when connected to the PCB in isolation, but would begin to malfunction when other components were introduced.

We believe this issue was due to electro-magnetic interference between the wiring of the other subsystems and the motor. This interference is known to cause problems with servo-motors, resulting in auto-pilot induced oscillations. [4]

Another reason the servo-motor malfunctioned could be due to improper connections within the PCB. The design of the PCB might be at fault due to the servo-motor's pins being too far away from the microcontroller.

5.3 Ethical Considerations

One specific ethical issue to consider is the collection of the runner's GPS data. We need to make sure that this data is not distributed, nor used in any manner outside of the runner's knowledge.

More generally, we need to make sure we follow the ACM and IEEE code of ethics to the best of our ability. [5][6]

Since this project will include software components, care is needed to properly credit all the resources we utilize. We must also be careful not to infringe on anyone else's copyright/intellectual property.

5.4 Future Work

Since our device already includes an accurate GPS module and algorithm, we think that a future extension to this device can be a mobile app. This app would allow the user to map out their run, as well as access different stats such as: the number of steps, average pace, and how many times they deviated from their target.

Additionally, our device would benefit with a more compact form factor. This can be easily achieved by refining our PCB, as well as opting for a surface mount GPS and solderable pins for the LCD display.

Finally, we recommend performing multiple surveys, which will provide us with feedback from users on different aspects which can be improved.

Appendix A: PCB Design

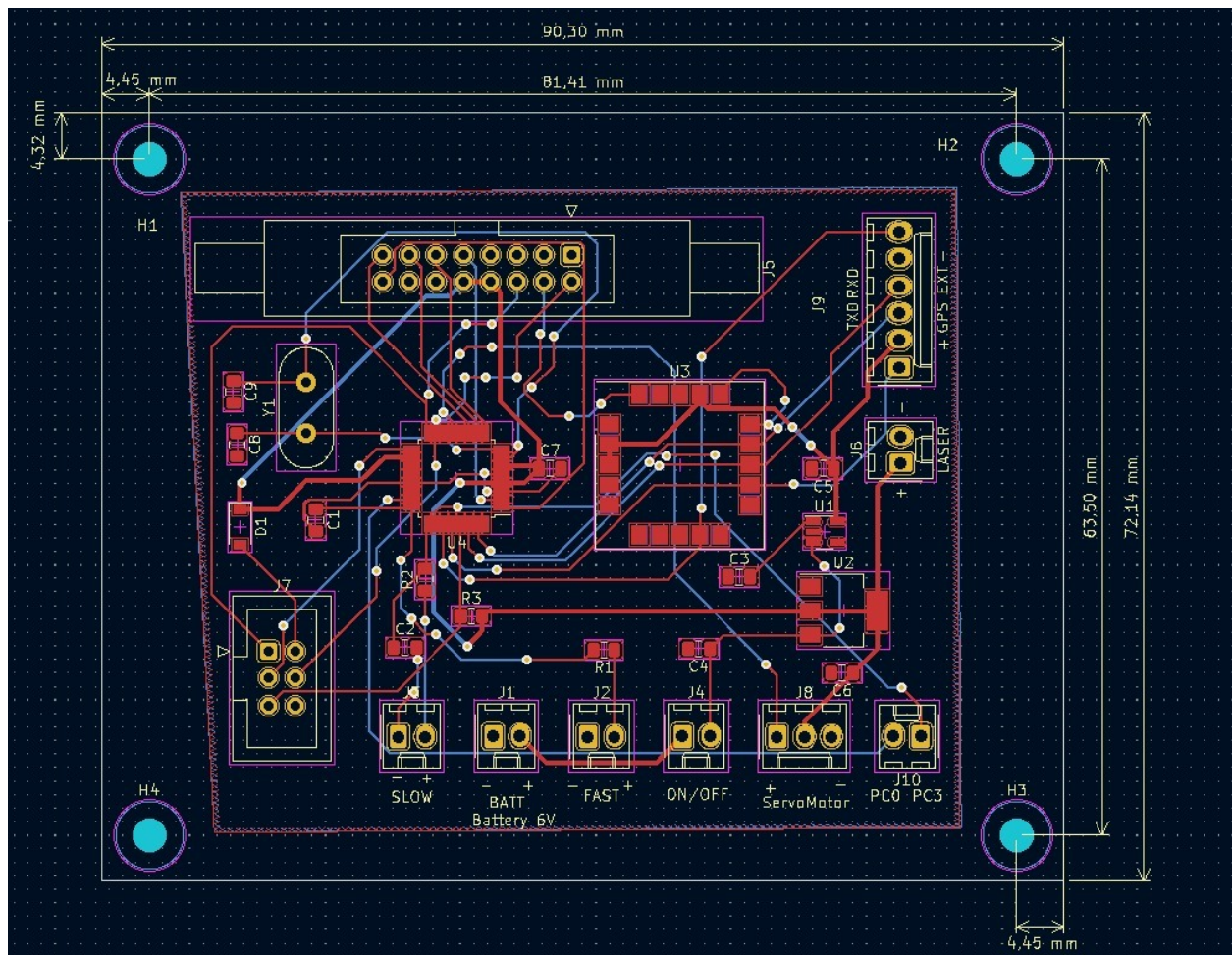


Figure 9: PCB KiCAD Design

Appendix B: Requirements and Verification Tables

Table 3: Power Subsystem R&V Table

Requirements	Verification	Verified?
1. Voltage Regulator 1 provides fixed 5.0 V +/- 0.5 V from 9 V source.	1. Measure the output voltage using an oscilloscope, ensuring that the output voltage stays within 0.5 V of 5.0 V.	Yes
3. The battery should provide at most a current of 600 mA +/- 50 mA. Since our device has an overall consumption of less than that amount.	3A. Connect the battery with a resistor of 10 Ohms. 3B. Measure the current using a multimeter to make sure that the current stays within 50 mA of 600 mA.	Yes

Table 4: Angle Control Subsystem R&V Table

Requirements	Verification	Verified?
1. Enter servo-motor angle as 27 degrees on the serial monitor via an arduino. The angle of projection of the laser should be 27 degrees.	1A. Program the device with the servo-motor program. 1B. Open the serial monitor, and enter 27. 1C. Verify that the angle of projection of the laser now moves to 27 degrees.	Yes (video)
2. Enter servo-motor angle as 45 degrees on the serial monitor via an arduino. The angle of projection of the laser should be 45 degrees.	2A. Program the device with the servo-motor program. 2B. Open the serial monitor, and enter 45. 2C. Verify that the angle of projection of the laser now moves to 45 degrees.	Yes (video)
3. Enter servo-motor angle as 56 degrees on the serial monitor via an arduino. The angle of projection of the laser should be 56 degrees.	3A. Program the device with the servo-motor program. 3B. Open the serial monitor, and enter 56. 3C. Verify that the angle of	Yes (video)

	projection of the laser now moves to 56 degrees.	
--	--	--

Table 5: Control Subsystem R&V Table

Requirements	Verification	Verified?
1. The microcontroller sends and displays data on the LCD screen.	1A. Connect the LCD display to the microcontroller using SPI. 1B. Verify that the LCD is capable of displaying the 3 digit speed of the runner.	Yes (picture)
2. The GPS module can track the user's location within 5 m.	2A. Program the GPS module to track a lap around the track for our runner. 2B. For the same lap use Strava, a running app on your phone to record the run which is known to give accurate data. 2C. After the run, compare the data side by side . If the margin of difference is within a circle of 5m radius around the runner the GPS is successful.	Yes (video)
3. The GPS module can track the user's speed within 0.5 mi/hr .	3A. Set the target pace as 5 mi/hr using the faster/slower buttons. 3B. Map out a lap that is 0.05 miles long, ask the runner to run this lap in 0.01 hrs / 36 seconds. 3C. Run this known lap, and verify that the speed on the LCD display sent by the GPS is within 0.5 of 5 mi/hr.	Yes (video)

Table 6: User Interface Subsystem R&V Table

Requirements	Verification	Verified?
1. The LCD displays the target pace set by the user.	<p>1A. Using the increase/decrease pace button, set the target pace to 5 mi/hr.</p> <p>1B. Verify that the LCD display displays 05.0</p>	Yes
2. The right button should increase target pace by 0.5.	<p>2A. Note the number shown on the LCD display. Click the increase pace button.</p> <p>2B. Verify that the number shown on the LCD display was incremented by 0.5.</p>	Yes
3. The left button should decrease target pace by 0.5.	<p>3A. Note the number shown on the LCD display. Click the increase pace button.</p> <p>3B. Verify that the number shown on the LCD display was decreased by 0.5.</p>	Yes

References

- [1] GPS Accuracy. "How accurate is GPS." GPS.gov. [Online].
<https://www.gps.gov/systems/gps/performance/accuracy>
- [2] GPS Receiver Module. "Datasheet" linxtechnologies.com. [Online]
<https://linxtechnologies.com/wp/wp-content/uploads/rxm-gps-rm.pdf>
- [3] ECE UIUC. "Salary Averages". Average Starting Salaries, Not Including Bonuses. [Online]
<https://ece.illinois.edu/admissions/why-ece/salary-averages>
- [4] DOC Ingeniamc. "Issues with Servo-Motors". EMI issues with Servo-Motors [Online]
<https://doc.ingeniamc.com/wiki/motion-wiki/electromagnetic-interference-issues-with-servo-drive-systems>
- [5] "ACM Code of Ethics and Professional Conduct." ACM.org. [Online]. Available:
<https://www.acm.org/code-of-ethics>.
- [6] IEEE Board of Directors. "IEEE Code of Ethics." IEEE Code of Policies, Section 7 - Professional Activities (Part A - IEEE Policies). [Online]. June 2020.
<https://www.ieee.org/about/corporate/governance/p7-8.html>.
- [6] WHO. "Health Risks from the Use of Laser Pointers" [Online]. Available:
<https://www.who.int/uv/resources/fact/fs202laserpointers.pdf>
- [7] E. I. S. Services, "Safety :: ECE 445," *Senior Design Laboratory*. [Online]. Available:
<https://courses.grainger.illinois.edu/ece445/guidelines/safety.asp>
- [8] "Basics of the SPI Communication Protocol," *Circuit Basics*, 13-Feb-2016. [Online].
<https://www.circuitbasics.com/basics-of-the-spi-communication-protocol>
- [9] "Interfacing 16x2 LCD with Atmega32 Microcontroller using Atmel Studio," electroSome, 20-May-2013. [Online]. Available:
<https://electrosome.com/interfacing-lcd-atmega32-microcontroller-atmel-studio/>
- [10] A. @ AEQ-WEB, "ATMega328 16x2 4-Bit LCD Display," *AEQ-WEB*, 07-Aug-2018. [Online]. Available: <https://www.aeq-web.com/atmega328-4bit-16x2-lcd-display-amtel-studio-c/?lang=en>.