Hall Effect Sensor Project Report

Brody Brodkorb and Thomas Loeffler

ECE 343: Instrument Systems

Dr. Mitofsky

10 / 24 / 24

## Introduction:

This project focuses on designing a handheld device that incorporates a Hall effect sensor, a microcontroller, a display, and various other components to measure and display data such as rotational speed and distance. The device's design balances functionality, ease of use, and reliability while maintaining constraints such as being handheld and fitting within a specific size due to 3D printing limitations. Two application concepts were developed: a bike speedometer/odometer and a pizza cutter-like measuring tool, both utilizing the Hall effect sensor for accurate readings. Collaboration with Design Engineering Technology (DET) students allowed for innovative packaging solutions and interdisciplinary teamwork.

### Problem Statement:

To design and implement an electrical system for a handheld instrumentation device that includes a sensor, display, microcontroller, and other necessary electronic components. The system must accurately gather, process, and display information from the sensor, maintaining ease of use and reliability. Extra features such as power-on LED, analog filters, etc. are encouraged.

### Constraints and Requirements of the Design:

The design of the entire project, including the electronics and the packaging, did not have many constraints or requirements. This project encouraged us to be creative and think outside of the box with the task at hand. One of the guidelines for the project was the sensor selection. The sensor had to be analog, or digital that measured an analog value, such as distance, speed, etc. Additionally, the project had to be handheld and fit within 5.25”x 5.25”x 5.25” due to the space limitation of the 3D printers. Lastly, the project must be completed within a budget of $110.

### Design:

For the product's design, we started by selecting a digital Hall effect sensor. This is a type of proximity sensor that detects magnetic fields. We used this sensor with our chosen microcontroller to measure velocity and distance in two applications. The firmware was optimized for both applications, providing unique features for each product.

This processed information is displayed on a 16x2 LCD screen, showing the respective values for each application. Other user-facing components were also used: a button for resetting and other functions, a dial potentiometer providing a way to select from a range of options, a power-on LED, and an on/off switch. Behind the scenes, other analog components support the main hardware through filters, decoupling, and other functions. The design for the product's electronics needed to be compatible with both concepts from the DET teams.

## Package Design Concept:

### Bike Speedometer Team (1):

DET team 1 envisioned a bike speedometer/odometer. With features like a digital screen, reset button, and more, the device is easy to use and feature-rich. The product wouldn’t be handheld but rather something to be mounted to the handlebars of the user’s bike. This idea is also unique because the sensor will not be mounted directly to the PCB. To capture the measurements of speed and distance, the sensor needs to be near the wheel, mounted on the front shocks of the bike. This setup, coupled with a magnet mounted to the wheel, allows the sensor to detect rotations and calculate the advertised quantities. Additionally, a dial potentiometer was added to adjust the wheel size. Wheel diameter is imperative for speed and distance calculations, and creating a way for this to be adjusted and used on bikes with different wheel sizes is a significant advantage.

A black rectangular device with buttons and a screen

Description automatically generatedA grey rectangular object with wires connected to it

Description automatically generated

Fig. 1 Bike Team Design Renderings

### Pizza Cutter Measurement Tool Team (2):

The second DET team came up with a measuring tool similar in form and function to a pizza cutter. This concept took the idea of measuring rotational distance and coupled it with the handheld constraint of the project. The same components are used: a digital screen for displaying the measuring program, a button for determining when to measure, and a dial for changing the units. These components are contained within a rectangular extrusion next to the handle for a more condensed form factor and ease of use. Overall, this product would have a wide use case and be applicable for nearly any use.



Fig. 2 Bike Team casing pictures (\*\*PLACEHOLDER\*\*)

## Parts List:

Some components were carefully selected, while others were included based on simplicity or manufacturers’ suggestions. Listed below are the components that were selected with purpose, along with the reasoning behind the selection. See the circuit design section for greater detail on product selection. Following that is the complete parts list, with prices and reference designators.

* U1: Our microcontroller was selected for its low cost, high clock speed, and relatively small footprint
* U2: LM7805 5V regulator, robust TO-220 packaging [1]
* R1 and C1: Used together to create a low pass filter for the analog input to the LCD. See the Circuit Diagram section for more details
* DS1: Generic 16x2 LCD display, no I2C backpack
* D1: Clear LED with green light, indicating if power is on
* C2 and C3: Decoupling capacitors placed on the input and output side of the 5V regulator. Used for stabilizing the input and output signals, protecting from input spikes and output ripples. Values determined by datasheet suggestions. [1]
* N/A (J1): Industrial Hall effect sensor selected for rugged conditions like the bike application. A case with wire leads, not the traditional compact package to be soldered on a PCB.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reference designator: | Part name: | QTY: | Unit Price | Total Price |
| U1 | Raspberry Pi Pico | 2 | $5.00 | $10.00 |
| U2 | 5V regulator | 2 | $0.09 | $0.18 |
| SW1 | On off switch | 2 | $0.61 | $1.22 |
| SW2 | Button | 2 | $.10 | $0.20 |
| R1 | 150 W resistor | 2 | $0.01 | $0.02 |
| R2 | Resistor 20k | 2 | $0.01 | $0.02 |
| RV1 | Potentiometer | 2 | $0.71 | $1.42 |
| J1 | Terminal blocks (3) | 2 | $0.53 | $1.06 |
| J2 | 9V battery harness | 2 | $0.53 | $1.06 |
| DS1 | LCD display 16x2 | 2 | $8.11 | $16.22 |
| D1 | LED | 2 | $0.29 | $0.58 |
| C1 | 10 mF capacitor | 2 | $0.10 | $0.20 |
| C2 | .1 mF capacitor | 2 | $0.10 | $0.20 |
| C3 | .33 mF capacitor | 2 | $0.10 | $0.20 |
| N/A (J1) | Hall effect sensor | 2 | $7.25 | $14.50 |
| N/A (J2) | 9V battery | 2 | $2.34 | $4.68 |
| N/A | 3 wire cable, 2ft | 1 | $0.20 | $0.20 |
| N/A | PCB | 3.2”x2.5” | $5 x 8 sq. in. | $40.00 |
| N/A | Magnet | 6 | $1.876 | $11.26 |
| N/A | Rubber gasket | 2 | $0.95 | $1.90 |
| N/A | Female headers 2x20 | 2 | $0.95 | $1.90 |
| N/A | Female headers 16 | 2 | $0.68 | $1.36 |
| N/A | Male headers 16 | 2 | $0.29 | $0.58 |
|  | Grand Total: |  |  | $108.96 |

Fig. 3 Parts list (including costs)

## Circuit Design:

The circuit used in this project was not one of immense complexity. Due to the processing of the microcontroller and the use of primarily digital components, some elements were plug-and-play. On the other hand, several situations required more analysis.

### LCD:

#### Overview:

To start with the most complex component other than the microcontroller, the LCD screen has 16 pins, 12 of which need connections. Without the incorporation of an “I2C backpack,” the package provides 8 data pins, requiring parallel binary data transfer. This allows data transfer of 8 bits at a time, but fewer pins can be used if desired. We opted to use 4 data pins, which is the most commonly accepted setup. The processing time on the LCD circuitry, which turns binary into symbols on the screen, is slower than the time it takes to send the data. This means that the use of 8 pins for a speed increase is negligible, and using 4 pins simplifies the circuitry and the programming. [4]

#### Contrast pin:

Next on the list of used LCD pins is the contrast pin. This is an analog input pin, determining how the contrast is set on the display in relation to the background. It takes a range of 0-5V, and under a sequence of trial and error, the ideal input voltage is around 1.3V. This presents a challenge because unless we want to add hardware (potentiometer, or DAC), there is no way of doing this directly due to the lack of an analog output pin on the Pico. Fortunately, this problem can be overcome by a PWM signal from the microcontroller. Using the equation below, we can generate any analog “effective voltage” from 0 - 3.3V

Solving for the duty cycle with a V high of 3.3V (operating voltage of the microcontroller), and a V effective of 1.3, the duty cycle comes out to 39.4%. For simplicity’s sake, that is rounded up to 40%, producing an effective voltage of 1.32 volts. This makes no difference to the display compared to 1.3V. A frequency of 1kHz was also chosen for simplicity’s sake, once it had been tested and shown to work, no further adjustment was needed.

A low-pass filter was incorporated to smooth out the PWM signal given to the LCD. Although the rectified signal is not free of ripple, it gets very close to DC and greatly improves upon the PWM signal the LCD would receive without this filter. This first-order low-pass filter has one resistor (R1) and one capacitor (C1). The components are connected in series, with the resistor coming first and the capacitor connected to ground, as shown in Fig. 4 below. Vout is referenced across the capacitor. The formula for component value selection is below (Fig. 5):

A diagram of a circuit

Description automatically generated

Fig. 5 RC filter equation [6]

Fig. 4 Low-Pass Filter [6]

In the equation, represents the cutoff frequency, which is normally 10% of a PWM frequency [7]. This allows effective filtering of high-frequency components while passing the desired signal. R represents the resistance of the resistor, and C is the capacitance of the capacitor. The PWM frequency is 1kHz. Therefore, the cutoff frequency is 100Hz. With two unknown variables (R and C), we will set one and solve for the other. Since capacitor values are more specialized, and there is a wider range of resistor values to use from, we let C = 10 mF. The equation becomes:

For a 10 mF capacitor at 100Hz cutoff frequency, the equation tells us 159.2 is the required resistor value in ohms. Considering the ±10% tolerance in capacitor values and available components, a 150 W resistor was selected for the filter. Once the Vout node on the low pass filter is connected to the contrast pin of the LCD, all required connections have been made.

#### Other pins:

The other pins on the LCD are connected as indicated by the datasheet. Power and ground for the overall LCD power as well as the backlight. Read/write pin in grounded because we only write to the display, and the enable and register select pins are connected to the respective GPIO pins on the Pico. See the software section for more details on how these pins are used.

### Power-on LED:

Another simple analog circuit within the overall circuit is the connection of the “power on” LED. This LED is used for an indication that the microcontroller is receiving 5V from the regulator and is connected between 5V and ground with a current-limiting resistor. This LED has an abnormal voltage drop of 2.3, meaning the current limiting resistor only has 2.7 volts across it. The selection of the current limiting resistor was based on the highest resistance we could get while still maintaining saturation brightness on the LED. This was done to save energy by keeping the current as low as possible. By trial and error, it was found that using a 20k W resistor allowed for the desired brightness and low current flow of 14mA.

### Other components:

All the other components within the circuit had basic wiring to power and/or GPIO pins. The button had one connection to GPIO 14 and the other to ground, grounding that pin when the button was pressed. The 20k W potentiometer had a 3.3V supply and ground connection, with the variable pin connected to GPIO26 for the ADC calculation. Lastly, the hall effect sensor also has a 3.3V supply and ground connection, and the output pin is wired to GPIO 15.

### Circuit Diagram:

Below is our circuit diagram (Fig. 6) for more details on the setup and decisions made, see the “Circuit Design” section above.

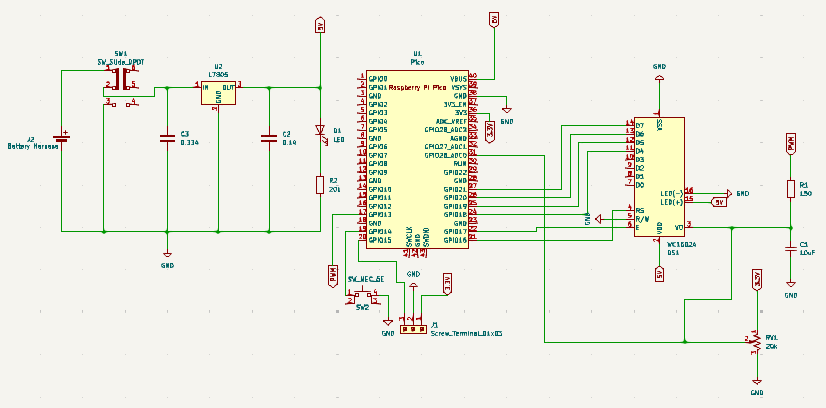


Fig. 6 Circuit Diagram (Placeholder)

## PCB Layout:

When designing the PCB layout, the primary focus was on the components visible to the user. Therefore, the potentiometer, button, power-on switch, power-on LED, and display had to be on the front of the PCB. The display was chosen to be at the top of the PCB and the other front components were spaced out across the front in a visibly pleasing and efficient way. The button was placed in the middle of the layout for easy pressing with the “pizza cutter” application.

The rest of the components were chosen to be placed on the backside of the PCB. This overall allows for the most appealing case design. The larger capacitors and voltage regulator were also placed away from other components, giving them space to be laid horizontally across the board for a slimmer case design. The spot for the battery leads was also chosen to be close to where the compartment for the battery will be.

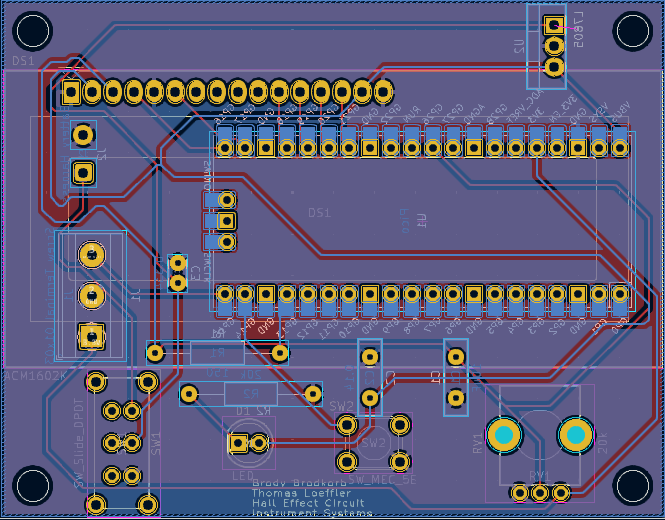


Fig. 7 PCB Raw Layout (front and back)

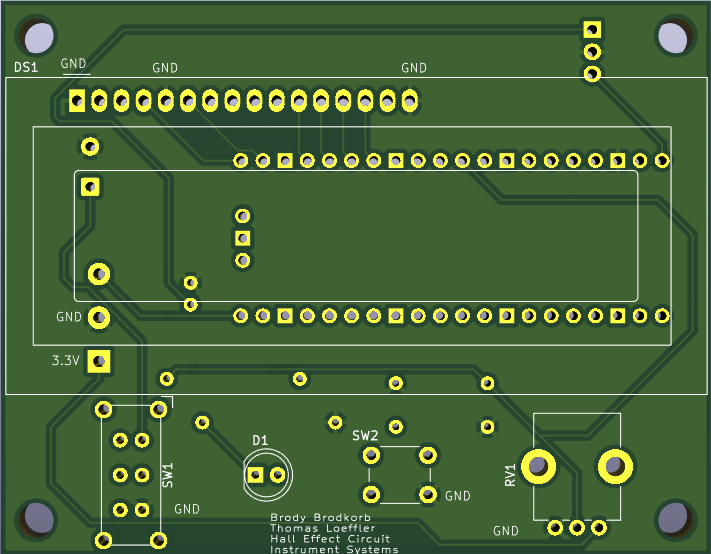


Fig. 8 PCB Front View

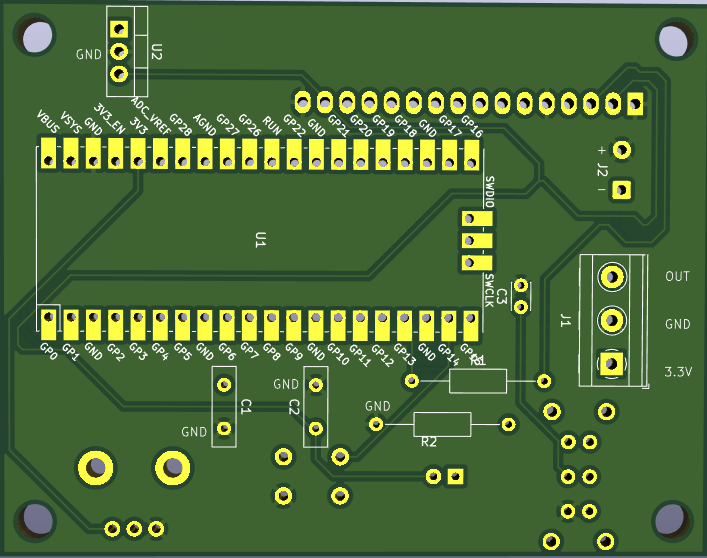


Fig. 9 PCB Back View

## Software Architecture:

The software used in this project was MicroPython, using Thonny as the IDE. Using a high-level language made most of the programming very easy and shorter in length. The only drawback to using MicroPython is that since it is a high-level language, it has many built-in functions and features that can also slow it down. Also, Python is an interpreted language, executed line by line. This can introduce other latencies compared to alternative languages like C/C++. However, due to the nature of our project, speed and efficiency can be sacrificed to make room for increased ease of programming and functionality.

For both projects, the bike speedometer and the pizza cutter measurement tool, four files were used for the overall program. Two of the files were imported for use in the LCD libraries. These files came from an LCD tutorial [5]. Importing these libraries allowed for extreme ease of use in terms of interfacing for the display. Both files are the same for both applications, but the next files vary slightly in syntax and function. Another one of the files was a user-made functions file. This file contained all the functions called in the main file, such as displaying the main menu, updating the speed/distance values, and generating the appropriate values from the ADC. Finally, the main file was the driver function. This file initialized all the pins and executed all the running code, like reading data from the sensor, button, and potentiometer. To make the main file easier to interpret and work with, several functions are called from the user-defined functions file. All four of these files are downloaded onto the Raspberry Pi Pico, and the main function runs automatically, calling the other files when needed.

## Sensor / Component Details:

### Hall Effect Sensor:

This project uses a digital hall effect sensor switch for detecting magnetic fields. A hall effect sensor is a device that can detect the presence of or change in the magnetic field. Within the sensor, there is a conductive plate that is a part of a current loop. Due to the strength of the magnetic field (B field), the magnet can alter the flow of charges on that plate, forcing the negative charges to one side and positive charges to the other side. (See Fig. 10)

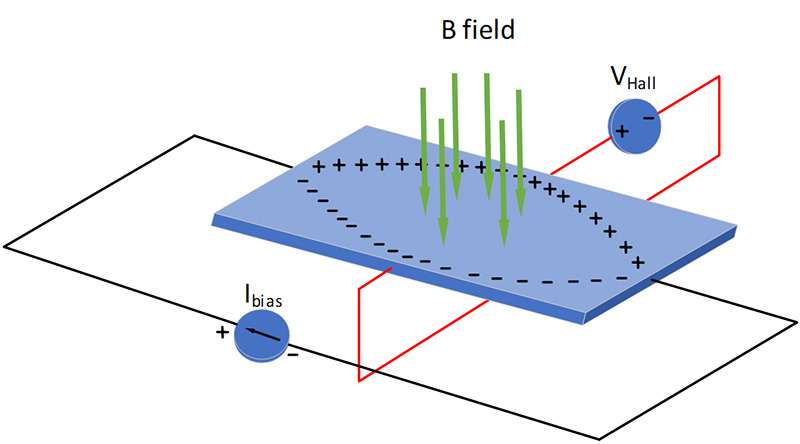


Fig. 10 Hall effect sensor diagram [3]

Take note that the magnetic field lines do not behave in a straight-on fashion as shown in the image, the depiction is for the sake of understanding. The difference in these charges creates a small potential difference between either side of the plate. This voltage is then amplified, and the output pin of the sensor will be pulled low when a magnetic field above the threshold strength is detected. This can be done using simple logic and op amps within the sensor. [3]

There are multiple types of hall effect sensors, the most common types being switch, latch, and linear. We used a switch so that a signal could be read every time a magnet passed by. This means the output pin is grounded when the magnet is near it, and then once the magnet is gone, Vcc is shown on the output. This is the most optimal setup for our application of detecting rotations. The latch sensor is one where the output toggles high/low whenever a magnet is detected, not changing states until the next time a magnet is passed by. Lastly, a linear sensor produces a voltage proportional to the magnetic field strength present.

Although the sensor is digital, an exception was made because it can measure analog values such as speed and distance. Using software, counts and time intervals are used for distance and speed calculations.

### Other Components:

The Hall effect sensor was the primary component that required an extensive amount of thought. Some thought was given to the microcontroller. The Raspberry Pi Pico was selected for multiple reasons, many relating to the ease of use and performance:

* Fast processor/clock speed
* Easy interface for programming and file transfer
* Allows for MicroPython, C, and C++ programming languages
* Affordable
* Many GPIO pins with a variety of functions (ADC, I2C, PWM)
* A large amount of flash memory (2MB)
* Lots of community support and resources

The only other component that was selected with specific intention was the clear LED. This was desirable for us because of the color difference between on and off. With our products being used outside in various lighting conditions and the potential for glare, the clear casing, and bright green light provided a noticeable difference between on and off. Whereas an LED with a green casing and green light might make it difficult to determine the difference between on and off. Other components, such as LCD screen, 5V regulator, etc., were somewhat generic and didn’t require too much intentionality with their selection.

## Battery Life Test:

Through a simple test, we measured the current draw from the battery to determine how long the battery would last under normal load/operating conditions. The measured current draw from the battery was 50.3 mA. Assuming an average 9V alkaline battery capacity of 550 mAh, the following calculations can be made: [2]

From the calculations, the device has an approximate run time of 11 hours. While this isn’t an amazing number compared to other low-power devices, it is still a substantial amount of time for our application and use case.

## Conclusion:

### Challenges and Lessons Learned Through Interdisciplinary Teamwork:

Through collaboration with the Design Engineering Technology students, many lessons were learned. Our means of communication was primarily through GroupMe, this allowed for quick questions and answers, as well as setting up meeting times. In conjunction, Microsoft Teams was also used for easy file and image sharing. It was eye-opening to see their side of the project and gain a deeper understanding of the subjects and challenges that they have to address. They must consider concepts related to the business and manufacturing side of the project, such as, “What can we do to make this casing easy and affordable to manufacture?” “How can we make this product easy to use for the customer?” etc. We also faced challenges on how to make sacrifices for the DETs. This included making changes like changing the positions of components or connectors, altering our PCB for mounting, and changing the function of hardware with software.

We also learned more about technical communication and how to clearly explain the complexities of our electrical system. The DETs were largely unaware of electrical hazards and concepts, so explaining our side clearly and effectively helped them understand better and plan accordingly for their side of the project. There were also moments of brainstorming and problem-solving where design and electrical ideas needed to come together. Communication and understanding from both sides were key in coming to a solution. Lastly, we as the electricals took the reins on the project in terms of leadership. We learned valuable lessons like project planning, setting up deadlines, and delegating tasks equally to all members of the team.

### What we learned:

This project provided so many opportunities to learn several aspects of electrical engineering. One of the more general takeaways was seeing how the new product development process looks from start to finish. Making all the decisions for component selection was something that proved to be more complex than we thought. We became more familiar with Digikey and more proficient in the component searching process, as well as looking at datasheets and seeking out components that had the desired characteristics.

We also spent a large amount of time looking at the datasheet for the Raspberry Pi Pico. Learning how to use that microcontroller was a long, but extremely beneficial process. We had to figure out how to wire it up using multiple internal power supplies, ground, and several GPIO pins. Learning how to code on it was not too complicated, but now we understand how to embed the files onto the microcontroller and make them run on an external power supply.

In addition to this, we had to import 2 LCD libraries for interfacing between the microcontroller and the LCD. Since the LCD display did not come with an I2C backpack, we had to perform parallel data transfer instead of serial. The microcontroller was I2C friendly, but without this backpack, we had to use the libraries and understand how they were used, which added another level of complexity to the software side of the project. Moreover, setting up the GPIO pins with components like the potentiometer and Hall effect sensor required yet another level of software setup, understanding, and code execution for correct and efficient use.

We also had to learn a lot about the PCB design process and how to use software (KiCad) to model what we wanted. One learning outcome was modeling our components with footprints and learning how to incorporate those with their symbols. Figuring out the most efficient way to place our components was also very important. We had to think about the most effective way to do this so that it allowed for the DET’s to configure the best possible case design. We also had to design how to route the tracks and where to put the labels, so that the layout has the highest possibility of success.

The hall effect sensor we used was not something as common as a temperature sensor. Once we had decided we wanted to use this sensor for magnetic field detection and rotational calculations, we dove deeper into how it actually works and breaking down the physics behind it. Lastly, our skills in time management were greatly strengthened. We quickly learned that tasks on projects like this take significantly longer than you would initially think.

### Advice for someone doing this project in the future:

Fortunately, we did not face many shortcomings or large problems in this project. The main advice to anyone looking to do this project in the future is to focus on your time management. Even if there is no timeline, plan on tasks and parts of the project taking longer than you think. We faced a bit of a time crunch when underestimating how long it would take to put together the demo. On a different note, two parts of the project that we did right were the microcontroller and the user input hardware. The Raspberry Pi Pico was a great choice for this project. It has several great features, and while there are a few drawbacks, you can’t beat it at its low price point of $5. Lastly, the use of a potentiometer and button turned out to be a wise decision. These components allowed for different types of user input, which greatly helped the architecture of the program and the versatility of the product

## References:

1. “LM7805.” *SparkFun Electronics*, www.sparkfun.com/datasheets/Components/LM7805.pdf. Accessed 2 Oct. 2024.
2. “Nine-Volt Battery.” *Wikipedia*, Wikimedia Foundation, 18 Sept. 2024, en.wikipedia.org/wiki/Nine-volt\_battery. Accessed 20 Oct. 2024.
3. Soltero, Manny. “What Is a Hall-Effect Sensor?” *Texas Instruments*, May 2021, www.ti.com/lit/ta/sszt164/sszt164.pdf?ts=1729423684901. Accessed 4 Sept. 2024.
4. “Specifications For Liquid Crystal Display.” AZ Displays, www.azdisplays.com/PDF/acm1602k.pdf. Accessed 2 Oct. 2024.
5. Staff, CircuitSchools. “Interfacing 16X2 LCD Module with Raspberry Pi Pico.” Circuit Schools, 11 Jan. 2022, www.circuitschools.com/interfacing-16x2-lcd-module-with-raspberry-pi-pico-with-and-without-i2c/#google\_vignette. Accessed 9 Oct. 2024.
6. “What Does a Low Pass Filter Do?” Hollyland, 5 July 2024, www.hollyland.com/blog/tips/what-does-a-low-pass-filter-do. Accessed 12 Oct. 2024.
7. Wagner, Jim. “Filtering PWM Signals.” *Ltwiki.Org*, Oct. 2009, ltwiki.org/images/8/82/PWM\_Filters.pdf. Accessed 12 Oct. 2024.

## Appendix: User Guides

### Bike Speedometer User Guide:

1. Mount the case to the handlebars.
2. Mount the hall effect sensor to the front “fork blades” (the vertical bars that connect the front wheel to the rest of the bike).
3. Remove the valve stem cap from the front wheel and install the provided valve stem cap.
4. Ensure the 9V battery is inserted, connected, and not dead.
5. Flip the switch to the on position (up)
6. Set your wheel diameter using the dial
7. Enjoy your ride!

Note: tap the button if you would like to reset the trip (distance).

### Pizza Cutter Measurement Tool User Guide:

1. Ensure the 9V battery is inserted, connected, and not dead.
2. Flip the switch to the on position (up)
3. Adjust the units if desired (Metric or Imperial)
4. Hold the button down to measure as long as needed
5. Let go of the button when the measurement is finished
6. If needed, repeat steps 4-5 to add to the measurement
7. Tap the button if you would like to reset the measurement

Note: The units may be changed after the measurement if needed