

ECE 445  
SENIOR DESIGN LABORATORY  
DESIGN DOCUMENT

---

# Any-Surface Computer Stylus

---

**Team #86**

ETHAN FORSELL  
(ethanf2@illinois.edu)

JOHN BLEDSOE  
(johndb3@illinois.edu)

ALEX CAMAJ  
(acamaj2@illinois.edu)

TA: Manvi Jha

February 27, 2026

## **Abstract**

This project proposes the development of a stylus that can be used improve the quality and accessibility of digital drawing on desktop computer systems. Conventional computer mice are difficult to use as a drawing tool while alternatives like drawing pads are bulky, expensive, and often overkill for simple sketching tasks. The proposed device aims to combine the convenience of a computer mouse, including the ability to work on most surfaces, with the ergonomics of a pen. The implementation will involve fitting sensors into a ergonomic pen shape and creating a control module to read and convert sensor inputs as well as house any power converters. The expectation for success will be a reliable, cost-effective, and ergonomic tool that makes drawing on the computer easy for all.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem . . . . .	1
1.2	Solution . . . . .	1
1.3	Visual Aid . . . . .	2
1.4	High-Level Requirements . . . . .	2
<b>2</b>	<b>Design</b>	<b>3</b>
2.1	Block Diagram . . . . .	3
2.2	Subsystem Overview . . . . .	3
2.2.1	Sensing Subsystem . . . . .	3
2.2.2	Control Subsystem . . . . .	4
2.2.3	Power Subsystem . . . . .	5
2.3	Subsystem Requirements & Verification . . . . .	5
2.3.1	Sensing Subsystem . . . . .	5
2.3.2	Control Subsystem . . . . .	6
2.3.3	Power Subsystem . . . . .	7
2.4	Hardware Design . . . . .	8
2.4.1	Housing . . . . .	8
2.4.2	Optical Sensor Construction . . . . .	9
2.5	Software Design . . . . .	9
2.5.1	Sensor Processing Firmware . . . . .	9
2.5.2	USB HID . . . . .	10
2.6	Component Selection . . . . .	11
2.6.1	Force Sensor . . . . .	11
2.6.2	Optical Sensor . . . . .	11
2.6.3	MCU . . . . .	11
2.7	Tolerance Analysis . . . . .	12
2.7.1	Power Requirements . . . . .	12
2.7.2	Optical Sensor . . . . .	12
2.7.3	IMU . . . . .	13
<b>3</b>	<b>Cost &amp; Schedule</b>	<b>14</b>
3.1	Bill of Materials . . . . .	14
3.2	Schedule . . . . .	14
<b>4</b>	<b>Ethics and Impact</b>	<b>16</b>
4.1	Public Impact . . . . .	16
4.2	Engineering Standards . . . . .	16
4.3	Safety . . . . .	16
	<b>References</b>	<b>17</b>

# 1 Introduction

## 1.1 Problem

The ability to draw quickly and reliably on a digital document used to be a luxury but is becoming more and more of a requirement. Physical paper becomes harder to find in offices and academia. Yet, the need for hand-drawings is still present. Tasks like signing and modifying documents, note taking, sketching ideas, and even some interviews require the ability to draw on a screen. The old fashion way was to print the original document, make the drawings with a pen, and then scan the document to send back to the computer. Now, touch-screen tablets have taken over academia, but they are large and very expensive tools that are an unnecessary luxury that many cannot afford. Artists solved this problem with the drawing tablet, a touch sensitive surface and an attached stylus. But these are still bulky, expensive, often require special software, and have a learning curve. There is yet to be a solution that is readily available that improves on the ergonomics of drawing with the convenience of a computer mouse.

## 1.2 Solution

Our product tackles these problems by offering a light-weight, ergonomic stylus that can be used to write on any surface. We want our stylus to instantly connect to a computer and operate with the sensitivity of a regular computer mouse. We're going to use an optical sensor to keep track of movement and a pressure or force sensor on the tip of the stylus so that it writes like a regular writing utensil. We will add an auxiliary button on the side for right click and a scroll wheel so that the pen has full control over the computer when in hand.

### 1.3 Visual Aid

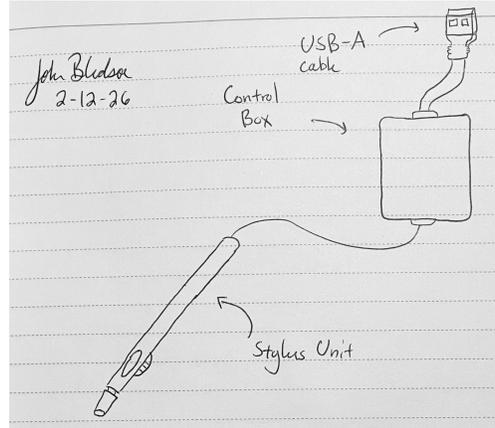


Figure 1: Visual Aid

### 1.4 High-Level Requirements

To deem our problem solved and our solution successful, we necessitate our product to be able to accomplish the following:

1. Writing speed test: We want our pen to be low latency enough to write as fast as a traditional pen and paper. We will quantify this by saying we want our stylus writing time for a given text to not increase by more than 25% of the time it would take with pen and paper.
2. Pen Writing Range: In order for our stylus to work like a regular pen, we need the cursor to move along with the stylus when you pick it up to maintain spacial continuity with the page. We would like the combined sensors to have enough spatial range to track the paper when the stylus tip is lifted at least 0.5 inches while keeping the optimal (non jittery) movement when close to the page.
3. Ergonomics and easy of use: We want our product to be easy to use and maneuver like a traditional pen or pencil. To accomplish this we are going to keep the stylus part of our product under 30 grams. This should allow for the expected ease of movement needed in a writing utensil.

## 2 Design

### 2.1 Block Diagram

At a high level, the proposed solution is to use the existing software interaction of a standard computer mouse while changing the hardware and housing to have the functionality of a stylus. The stylus is meant to be pen-shaped and lightweight so it will contain only the necessary sensors. This is represented by the first block of the block diagram shown in Fig. 2. To handle the power and processing for the pen is a separated processing unit, represented by the control and power subsystem.

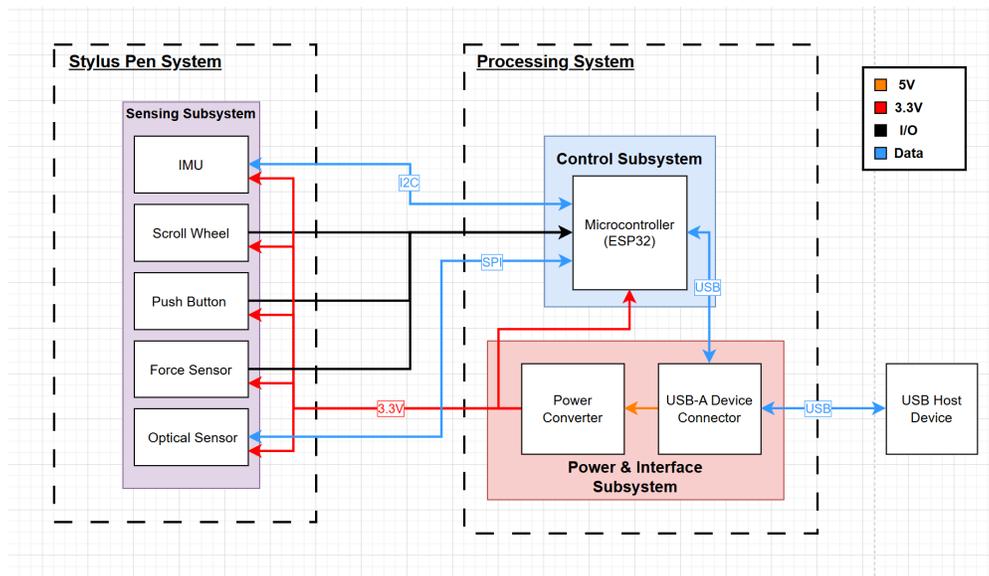


Figure 2: Block Diagram

### 2.2 Subsystem Overview

#### 2.2.1 Sensing Subsystem

In order to keep the physical size of the stylus to a minimum, the stylus will hold only the necessary sensors. The sensors include an optical sensor, a pressure sensor, a push button, scroll input, and an IMU for pen control in the air.

**Optical Sensor:** The optical sensor is responsible for tracking two dimensional movements and will also be the most difficult to implement given the variable height of the stylus off the surface and the need for precision. This also requires specific mechanical construction with a lens and light in order to focus a light into the optical sensor for optimal functionality. The light source that we will be using is a simple LED. However, we need that light to hit the surface on an angle for the optical sensor to get a good scan of the movement. For this we are going to use a fiber optic cable to accurately get the light to shine without taking much space up in our pen frame. The optical sensor will communicate with the microcontroller over SPI sending the change in X and Y position so the

cursor for the given device updates accordingly. The optical sensor will be primarily used when the pen tip is touching a surface.

**Pressure Sensor:** The pressure sensor will detect contact with the surface. This information will be sent to the microcontroller which will have a threshold to output a left click and hold. The best option for our pressure sensor is a force sensitive resistor (FSR) which changes resistance with applied force. This threshold will also be used to determine which movement sensor has control of the cursor. When the FSR threshold is exceeded, the optical sensor will control the cursor. Otherwise in the air we will use the IMU for control. We plan to mount the FSR inside the stylus and use a spring and movable tip mechanical system to get the force exerted on the resistor.

**Button and Scroll Wheel:** The current design will also include two buttons for additional functionality. One simple push-button will serve as a right click. We plan to use a regular mechanical switch (which will have to be de-bounced) and a cover for the button to have it sit flush with the stylus. The other will be a scroll wheel for moving the page. The scroll wheel will be implemented using a rotary encoder mounted on the stylus. These sensors will communicate to the microcontroller using I/O.

**IMU:** The final sensor inside the stylus pen frame will be the Inertial Measurement Unit (IMU). We have selected the IMU for control of the stylus pen when it is in the air. When the pen is off the surface, the optical sensor will be too far from the surface to accurately process the light from the fiber optic cable. This is when the IMU will take over control. In order for the IMU to be useful, we need to control the stylus cursor using our wrist tilt primarily as opposed to typical linear movement of a mouse. This is because the IMU measures acceleration and moving linearly will build up error in position calculations, rendering it not effective in control. This will still allow us to control the cursor accurately and comfortably, but the movement will be different. This will be communicating with the microcontroller using I2C.

### 2.2.2 Control Subsystem

All the sensor information will be sent to the processing box where a microcontroller will process the incoming information, convert it to the appropriate mouse signals, and send them to the computer. For this job, an ESP32 will work. It has plenty of inputs for our analog signals, can operate an optical sensor, includes various communication protocols, and has enough processing capability to handle all the sensor information at the same time. It will process SPI, I2C, Analog, I/O, and USB processing. This is the brain of the pen and it will fuse all the sensor data in order to make correct actions acting as a typical mouse. It will determine which sensor is in control of the cursor, whether to left click, right click, or scroll. This will generate the necessary D+ and D- signals for the USB protocol to talk to the USB Host device and translate the proper X and Y cursor positions.

The board will also be in charge of generating the necessary Human Interface Device (HID) signals. This will allow the pen to be recognized when it is plugged into any device and give it full control of the cursor and all of the button functionality.

### 2.2.3 Power Subsystem

The power subsystem consists of a USB-A connector and a power regulator. The power regulator is responsible for maintaining a 3.3V power sources to all of the sensors and the microcontroller. The USB-A connector serves as a power source and will receive a 5V input from the host device along with a maximum current limit of 500 mA. There is no battery required which allows this device to be used for as long as the user would like. The USB interface also allows the device to be used universally with any PC that has a USB-A host port.

This subsystem also communicates with the host device using the USB protocol. It acts as a bridge from the stylus pen to the host device. This communication will allow the host device to receive the data necessary for controlling the cursor and recognizing the pen as a HID. Once connected, the connector will then forward the data down to the microcontroller using the USB protocol and let it know of the connection status and if it is ready to start providing output data.

## 2.3 Subsystem Requirements & Verification

### 2.3.1 Sensing Subsystem

Table 1: Sensing Subsystem Requirements & Verification Methods

Requirement	Verification
The optical sensor must correctly output displacement in X and Y position with an error of no more than 5%	Move the stylus pen exactly 100mm along a ruler horizontally and then read the cumulative delta X count from the ESP32 terminal and convert that output using the sensor calibration (counts/mm). Confirm the position movement reported is within 105 to 95 mm. Repeat 5 trials and track data in a table.
The optical sensor must stop reporting X/Y displacement when the pen is lifted 3.0mm $\pm$ 0.5 mm from the writing surface	Confirm there is displacement output on the ESP32 serial monitor when moving the pen on the surface and then place a 3mm spacer block and verify that that all the counts being displayed are zero. Record the sensor output in a data log.
Continued on next page	

**Table 1 – continued from previous page**

Requirement	Verification
<p>When the stylus is more than 3mm the IMU must control the mouse in all 4 directions by tilting to the designated destination and stop moving when the pen is held flat within 5 degrees of horizontal.</p>	<p>With the pen held in the air above 3mm tilt it 45 degrees in each direction (left, right, forward, backward) and confirm the cursor on screen moves in each direction. Then return the pen back to 5 degrees within horizontal and confirm there is no cursor movement on screen. Save data with videos of trials as well as pass/fail trial date saved in a table.</p>
<p>The push button should register a right click event within 100ms of the physical press and de register when there is a release of the button with no more than one click event registered for each physical press.</p>	<p>Connect a logic analyzer to the push button GPIO pin and a USB protocol analyzer to the USB port. Press and release the button 10 times. Record the time from GPIO falling edge to right-click event in the HID stream and the number of click-down events per press in a table. All 10 trials must show latency 100 ms and exactly 1 click-down event per press.</p>
<p>The scroll wheel must register exactly 1 scroll event for each click and correctly distinguish upward from downward scroll direction.</p>	<p>Connect a serial monitor on the PC host device. Move the scroll wheel up and down for 10 detents each and verify there is 10 "+1" followed by 10 "-1" on the monitor. Save the data log.</p>
<p>When the stylus pen is dragged across a surface under constant pressure above our threshold, the FSR should maintain a constant left click without interruption for 100mm.</p>	<p>Attach enough rubber bands around the pen tip to trigger a force greater than the left click threshold. With those applied drag the mouse across the writing surface, monitor the voltage of the analog output of the FSR and ensure that a left click is being displayed on the PC monitor.</p>

### 2.3.2 Control Subsystem

Table 2: Control Subsystem Requirements & Verification Methods

Requirement	Verification
The ESP32 can enumerate as a USB HID mouse device within 5 seconds of being connected to a host device.	Connect the control box to a windows machine and observe the device manager. Measure the time from the USB plug in to the device appearance using a stopwatch.
When the stylus tip is transferred from the surface to above 3.0mm the control of the cursor switches from the optical sensor to the IMU	On the host device use a serial monitor and verify that once the pen is in the air the data values for the cursor are now coming from the IMU. Send cursor data with control interface in header. This can be recorded in a data log along with a video.
The microcontroller can record data from all 5 sensors and relay them over USB to the host device with less than 5% loss of data	Write a firmware test that polls all of the sensors for 60 seconds and log the data verifying that each sensor was recorded the same amount of times for the entire 60 seconds. Interact with each button to verify data changes upon physical interaction. Connect USB analyzer to host device and verify that there is a max of 5% packets lost within a 1 minute recording window.
After 5 minutes of inactivity the pen enters a low power mode where it polls the IMU for any movement and awakens when there is a movement	Leave the stylus untouched for 5 minutes connected to the host device. Measure the current draw through the USB. Confirm that the current has dropped when compared to the active state. Shake the stylus pen and then confirm the cursor now moves and the pen has entered back to normal mode. Record the active current, low power current, and wake time in a table.

### 2.3.3 Power Subsystem

Table 3: Power Subsystem Requirements & Verification Methods

Requirement	Verification
The power converter must be able to step down the 5V input to $3.3V \pm 0.3V$ for all of the subsystems	Using a digital multimeter, verify that the input to the sensors is within range.

Continued on next page

**Table 3 – continued from previous page**

Requirement	Verification
The entire stylus must draw no more than 500 mA to comply with USB 2.0 standards	Use a digital multimeter to check the current draw from the USB port and make sure it isn't over 500mA.
The USB-A Connector must be recognized when plugged into a USB host device.	Open the device manger on the host device that the pen is plugged into and confirm it is recognized as a HID and no error messages such as "Device Not Recognized" shows.

## 2.4 Hardware Design

### 2.4.1 Housing

To house the components and make the stylus comfortable in the hand a cylindrical housing is a must. For the components being used, the electronics will need a custom housing to hold the electronics at the right place and provide the mechanical functionality. These housing components can be 3D printed for proper sizing.

The stylus tip will need to be separate from the main body. This is so the tip can be slightly depressed in order for force exerted onto the tip to translate to force onto the force-sensitive resistor. A small rubber piece on the surface of the resistor's pad will ensure that force is spread evenly and provide a spring-like stiffness to the tip. The tip-piece will be held between the bottom of the housing and the bottom of the rubber stopper.

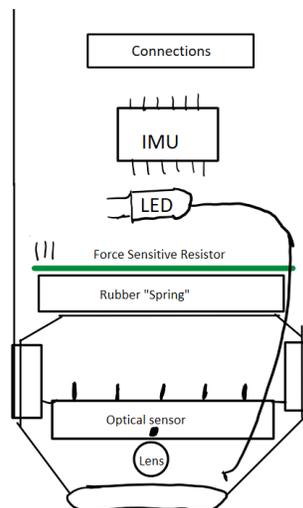


Figure 3: Housing Sketch

## 2.4.2 Optical Sensor Construction

The optical sensor will be housed in the tip of the stylus along with a lens. This sensor will be identical to that of a common computer mouse. It will be fixed such that when the stylus is pressed flat against the table the sensor is as close as possible to function smoothly. That leaves the rest of the range (about 2mm) as liftoff distance when writing.

The optical sensor requires the surface it is reading to be properly illuminated. In a common mouse this is done through mirrors and an LED to reflect the light at an angle onto the surface. This angle casts shadows of the peaks and valleys on the surface that the sensor is able to recognize, track, and translate to motion. To avoid the bulk of an LED and mirrors in the tip of the pen our design will utilize fiber optic cable to guide light from an LED located in the base of the stylus to the tip at as steep an angle possible with the given tip geometry.

## 2.5 Software Design

The software will be running on an ESP32 which will be responsible for interfacing with the 5 sensors in the pen, computing cursor movement, and communicating over USB to the host device. This firmware for this will be written in C using the ESP-IDF. We will take advantage of the dual core functionality of this chip and use one for processing sensor info and the other for USB interfacing. They will communicate using an RTOS queue where the sensor core will publish its data to a queue and the USB core will subscribe and link that to the USB host device.

### 2.5.1 Sensor Processing Firmware

The sensing core will function using a timer interrupt every 8ms to reach the necessary timing restraints for a smooth cursor movement. Depending on the current state we will read either the optical sensor or IMU output followed by the FSR, push button, and scroll wheel.

**States:** The pen will have a four state framework that allows it to perform in multiple environments.

**INITIALIZE:** All sensors will be initialized and the link between the host and the device will be established.

**SURFACE:** This state is the first state after initialization finishes. The optical sensor is in control of the cursor movements when in this state.

**AIR:** This mode is entered when the FSR reading is below the threshold and the IMU takes over the control.

**SLEEP:** This is the low power mode that the device enters when there is more than 5 minutes of inactivity.

**Transitions:**

- INITIALIZE → SURFACE OR AIR: After power up and all the sensors have been initialized the state will then transition to either SURFACE or AIR depending on the FSR pressure level.
- SURFACE → AIR: This happens when the pen FSR pressure transitions from above to below the threshold. The control then goes from the optical sensor to the IMU.
- AIR → SURFACE: This happens when the pen FSR pressure transitions from below to above the threshold. The control then goes from the IMU sensor to the optical sensor.
- AIR/SURFACE → SLEEP: When there is no change in the IMU or optical sensor output for 5 minutes the system will go into a low power sleep until the IMU detects movement again.

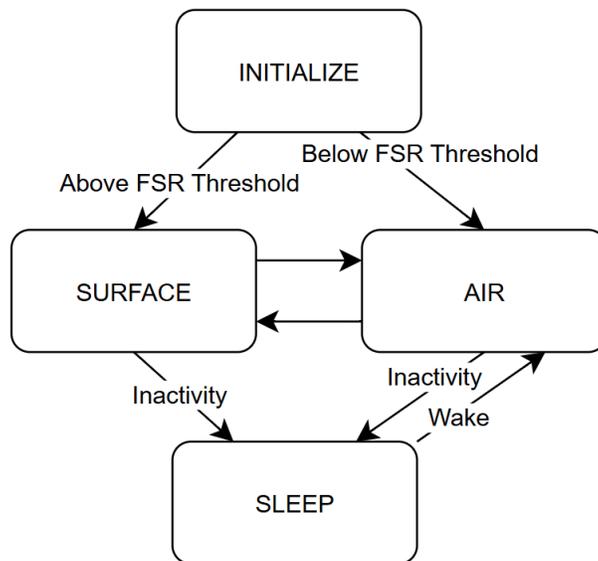


Figure 4: FSM

## 2.5.2 USB HID

The ESP32 has a naive TinyUSB stack that is included within the ESP-IDF framework. This allows the device to present itself as a standard HID mouse class device.

The HID report descriptor defines the mouse with three buttons left, right, and middle click, a 16 bit signed field for the X and Y displacement, and an 8-bit signed vertical scroll field. The report will be 4 bytes total. We will map this to our buttons using the framework:

- left click: FSR pen-down state
- right click: push button state
- X, Y displacement: outputs of state machine (either optical or IMU displacement)

- Scroll: scroll wheel accumulator value

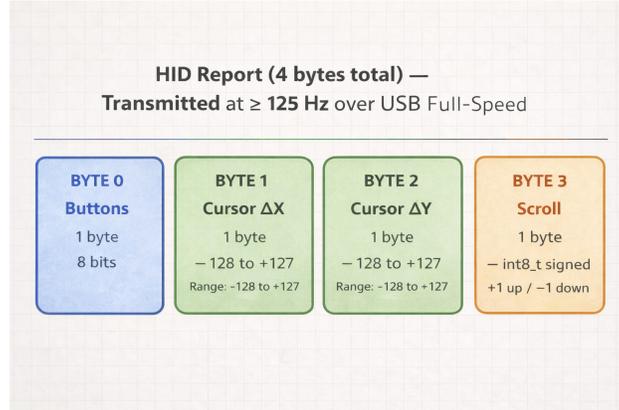


Figure 5: Report Layout

The USB Core will run the TinyUSB task continuously. Whenever there is a new packet that is available in the RTOS queue, published from the sensor core, the USB Core will dequeue it and reports the queue over USB to the host device. This must be submitted at a rate of 125 Hz in order to meet baseline cursor smoothness.

## 2.6 Component Selection

### 2.6.1 Force Sensor

There are many ways to detect if the stylus is making contact with the surface: a force sensitive resistor, pushbutton, hall sensors. We chose to use a force sensitive resistor. They are flat, lightweight, and simple making them perfect for use in our stylus. A force applied to the pad of the resistor reduces its resistance according to the curve shown in Fig. 6[1]. By tuning a voltage divider we can connect the sensor to a ADC pin on the ESP32 and convert to a serial value that can be used and tuned to our needs. A rubber piece will act to disperse the force on the tip across the entire pad of the resistor.

### 2.6.2 Optical Sensor

We opted to use a common computer mouse sensor in the MX8733B. This optical sensor implements many of the functions we need in a very compact function. It also has the advantage of being very inexpensive and has a wider 2mm range of operation before it detects liftoff, making writing possible. It also processes the sensor data into USB differential signaling and contains an on-board LED driver[2].

### 2.6.3 MCU

The ESP32 was chosen to provide the control to the circuit. It has a large number of pins for connecting all the sensors and buttons. It is compatible with analog signals,

SPI communication, and GPIO. The processor is plenty fast and capable for the software implementation. Additionally, the dev-board also comes with the voltage regulator to handle power conversion between 5v and 3.3v.

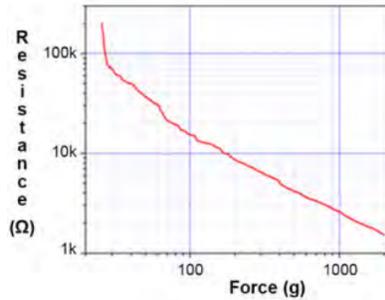


Figure 6: Report Layout

## 2.7 Tolerance Analysis

### 2.7.1 Power Requirements

Power is supplied through a USB cable. This power is nominally 5v but is guaranteed to range between 4.45-5.25 V minus the voltage drop across the cable, which should be negligible at the low currents being drawn.

The 5v from the USB connects to a regulator that drops the voltage to  $3.3\text{v} \pm 0.3\text{v}$ . The AMS1117 regulator is included on the ESP32 Dev Board.

The optical sensor (MX8733B) requires a 4.5-5.5v operation. So it can be connected directly to the USB provided 5v. It should draw maximum of 15mA during operation and 7.5mA when idle. The LED is driven from a circuit internal to the optical sensor IC so the power draw is included in the 15mA current [2].

The ESP32 operates at 3.3v and 44mA idle and up to 180mA during peak activity including wifi-transmission [3]. Because we are not using the wireless features the current spikes should be limited. This means that even at the highest power draw expected from all components, the regulator is under rated power and should not overheat.

### 2.7.2 Optical Sensor

The optical sensor has an operating distance. It first starts working when the bottom of the lens is 2mm off the surface and is unable to detect movement when lifted 4mm off the surface, giving a 2mm operating range. The sensor works most reliably when the surface is illuminated at an angle from a single light source. The stylus will need to block ambient light and supply its own for the best performance.

### 2.7.3 IMU

The IMU is responsible for sensing the less precise movements when the stylus is too far from the surface for the optical sensor. It uses tilt and acceleration data which is translated to XY movement. The sensor contains analog to digital conversion within the IC which is available in single nano-second time scales [4]. the SPI data must be translated to XY mouse controls within the microcontroller unit, which takes micro-second time scale. Even putting the delays together the sensor should react plenty-fast to feel instant to a user. The combination of tilt and acceleration may require some skill to use but with proper tuning should be able to move the cursor with the pen lifted off the table.

### 3 Cost & Schedule

#### 3.1 Bill of Materials

Description	Manufacturer	Quantity	Extended Price	Link
ESP-32	HiLetGo	1	\$7.95	<a href="#">Link</a>
Optical Mouse Sensor MX8733B And Lens	Shenzhen LIZE	1	\$2.00*	<a href="#">Link</a>
Force Sensitive Resistor 34-00004	Interlink Electronics	1	\$7.90	<a href="#">Link</a>
LED Bulb		1	\$0.45	<a href="#">Link</a>
1mm Fiber Optic Cable	Azimom	164ft	\$10.89	<a href="#">Link</a>
IMU ICM-40609-D	Interlink Electronics	1	\$4.29	<a href="#">Link</a>

Table 4: Bill of Materials

**Labor Costs:** We estimate that it will take approximately 80 hours (not counting writing reports) to design, create, and thoroughly test out product. At the average ECE graduate salary of \$80,000 (approximately \$38 per hour) we expect the total labor cost for this project to be:

$$\text{Total} = \$/\text{hour} * 2.5 * \text{team members} * \text{hours}$$

$$\$22800 = 38 * 2.5 * 3 * 80$$

#### 3.2 Schedule

<b>Week</b>	<b>Task</b>	<b>Person</b>
<b>FEB 23 - MAR 1</b>	Order Parts	Everyone
	Design Document	Everyone
<b>MAR 2 - MAR 8</b>	Test Sensors	Everyone
	Prototype	Everyone
	Design Review	Everyone
	Work on PCB design(s) + Order	Everyone
<b>MAR 9 - MAR 15</b>	Design and Print Housing	Everyone
	Pass Audit	Everyone
	Breadboard Demo	Everyone
	Team Evaluation 1	Individual
	PCB Order	Everyone
	Software Implementation	Everyone
<b>MAR 16 - MAR 22</b>	Spring Break	Everyone
<b>MAR 23 - MAR 29</b>	Last PCB Order	Everyone
	Assembly	Everyone
<b>MAR 30 - APR 5</b>	Individual Progress Report	Individual
	Final Assembly	Everyone
<b>APR 6 - APR 12</b>	Progress Demo	Everyone
	Team Contact Assessment	Everyone
<b>APR 13 - APR 19</b>	Bug Fixed + Verification	Everyone
<b>APR 20 - APR 26</b>	Mock Demo + Presentation	Everyone
	Final Tweaks	Everyone
<b>APR 27 - MAY 3</b>	Final Demo + Presentation	Everyone

Table 5: Project Timeline

## 4 Ethics and Impact

### 4.1 Public Impact

In creating this project, we are striving to better people's lives through making writing on computers more accessible and convenient. We want our product to theoretically be accessible to all people so we are trying to make it as cheaply as possible. We're also trying to make it as efficiently as possible from an environmental perspective. We'd like our product to be relatively harmless to the environment and perhaps even be beneficial helping to cut down on wasteful use of paper and printing. We don't foresee our product having any impact on culture or global factors.

### 4.2 Engineering Standards

Any engineering project is required to comply to the IEEE code of ethics [5]. While the impacts and risks associated with creating or using our stylus are few, decisions should still be made with regard to human safety, privacy, and capability. The three most important IEEE codes we'll have in mind while creating and using our product are Sec I, codes 1, 2, and 5. Additionally, we think ACM codes [6] 1.1, 1.6, and 2.1 apply handily to this project. We intend to take safety and privacy seriously and create a product that people feel comfortable using.

### 4.3 Safety

Besides taking our own safety into account when putting together the project, one concern that we have is the optical sensor used for tracking movement. The sensor is a laser that shoots a focused beam of light that could potentially cause harm if shined into someone's eye. Our sensor will probably be similar to that of a computer mouse but it could be more dangerous because our laser may need to be focused further from the point of emission so that the stylus can track at a distance. We want to make a point to say that our stylus should be used responsibly and in accordance with general laser safety rules. We have chosen to specifically abide by the LIA laser safety [7] guidelines.

While there is low risk at the low voltage required for this project, any power conversion should meet certain efficiency and thermal requirements. We will be sure to use good wiring practices to prevent shorts and other potentially issues.

Privacy is another important factor when it comes to electronic devices. One advantage of choosing a wired vs a wireless design is having a secure communication channel so no information can be read from the mouse by any third party actors. No user information is stored so the device itself is secure.

## References

- [1] *Fsr® 400 series data sheet*, [https://cdn2.hubspot.net/hubfs/3899023/Interlinkelectronics%20November2017/Docs/Datasheet\\_FSR.pdf](https://cdn2.hubspot.net/hubfs/3899023/Interlinkelectronics%20November2017/Docs/Datasheet_FSR.pdf), Accessed: 2026-02-27.
- [2] *Mx8733b usb single chip optical mouse sensor*, <https://datasheet4u.com/pdf-down/M/X/8/MX8733B-LIZEElectronic.pdf>, Accessed: 2026-02-27.
- [3] S. Prabhu, *Insight into esp32 sleep modes their power consumption*, <https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/>, Accessed: 2026-02-27.
- [4] *Icm-40609-d datasheet*, <https://invensense.tdk.com/wp-content/uploads/2022/07/DS-000330-ICM-40609-D-v1.2.pdf>, Accessed: 2026-02-27.
- [5] IEEE. "IEEE Code of Ethics," Accessed: Feb. 10, 2026. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8>.
- [6] ACM. "ACM Code of Ethics," Accessed: Feb. 27, 2026. [Online]. Available: <https://www.acm.org/code-of-ethics>.
- [7] LIA. "LIA Laser Safety Information," Accessed: Feb. 27, 2026. [Online]. Available: [https://www.lia.org/resources/laser-safety-information?srsltid=AfmBOorIHd2w8-x3POavt33yMtar3aLtTom\\_zEYMQNBG1uMVKjFXQS\\_X](https://www.lia.org/resources/laser-safety-information?srsltid=AfmBOorIHd2w8-x3POavt33yMtar3aLtTom_zEYMQNBG1uMVKjFXQS_X).