Design Document

ST 4001 Senior Thesis

Thesis Title: Smart Foot-Controlled Mouse with UI-Aware Assistance

Student Name(s): Zhihao Cheng

Hao Liu

Jiongye Liu

Chaoxiang Yang

Major: EE&ME

Supervisor Name: Dr. Weeliat Ong

Contents

1 Introduction

1.1 Problem and Solution Overview

Modern human-computer interaction depends heavily on hand-operated devices such as mice and keyboards. While these input devices are effective for most users, they present substantial accessibility barriers for people with upper-limb disabilities or limited hand mobility. Tasks such as moving a cursor, clicking a small button, or dragging an object on a graphical interface require fine motor control that many users cannot easily perform with conventional hand-based devices. Prior work has explored alternative desktop input using foot-operated devices, ranging from early planar foot cursor control to more recent studies of foot-operated input in modern desktop applications [3, 2].

Existing assistive input technologies partially address this issue, but each has clear limitations. A practical alternative must provide continuous cursor movement, reliable command input, and sufficient precision for real desktop tasks. At the same time, target-acquisition research has shown that software-side assistance techniques such as semantic pointing and dynamically adaptive cursors can significantly improve the selection of small graphical targets [4, 5]. These ideas are consistent with the general movement-time constraints captured by Fitts' law [6].

To address this problem, our project proposes a smart wearable foot-controlled mouse with UI-aware assistance. The hardware uses multiple force-sensitive resistors embedded in a slipper-like wearable structure to sense foot pressure distribution. A microcontroller processes the sensor readings and converts them into two-dimensional cursor movement and mouse commands such as click, double-click, and drag. In addition, a PC-side UI-aware assistance module improves precise interaction by reducing cursor gain near dense interface regions and by assisting target acquisition for small clickable elements. Compared with a basic foot-only pointing system, the proposed design aims to provide higher practical usability, better small-target selection performance, and a more complete alternative to a traditional mouse. Recent open-source tactile sensing work also supports the feasibility of wearable pressure-based systems in shoe-sole-like applications [1].

1.2 Visual Aid

Figure 1 illustrates the complete usage scenario of the system. The user wears a slipper-like foot interface containing pressure sensors and auxiliary buttons. Sensor signals are sent to an embedded controller, which translates foot gestures into USB HID mouse commands. These commands are transmitted to a computer, where a software assistance module monitors interface context and improves fine-grained cursor selection for icons, buttons, and links. The figure emphasizes how the device is used in a real computer-accessibility setting rather than only showing internal hardware blocks.

Fig. 1. Conceptual usage scenario of the smart foot-controlled mouse with interface-aware assistance: The user operates the computer cursor with foot input while the PC-side assistive module enhances graphical interface interactions.

Figure 1: Conceptual usage of the smart foot-controlled mouse with UI-aware assistance. The user controls the computer cursor through wearable foot input, while the PC-side assistance module improves precise interaction with graphical user interfaces.

1.3 High-Level Requirements

- The system shall provide continuous two-dimensional cursor control with an end-to-end update latency of no more than 80 ms and a cursor update rate of at least 50 Hz during normal operation.
- The system shall correctly execute basic mouse operations, including single click, double-

click, and drag, with an overall command success rate of at least 95% and a mis-trigger rate of no more than 5% under standard testing conditions.

- The UI-aware assistance module shall improve the successful selection rate of small on-screen targets (16 px or smaller in width or height) by at least 20% compared with baseline foot-only control.

2 Design

2.1 Block Diagram

The system consists of five main subsystems: the sensing subsystem, the embedded processing subsystem, the communication subsystem, the power and wearable structure subsystem, and the PC software subsystem. The sensing subsystem measures foot pressure distribution through multiple force-sensitive resistors and auxiliary buttons. The embedded processing subsystem samples and filters sensor data, interprets gestures, and generates cursor velocity and mouse command events. The communication subsystem transmits these commands to the PC through USB HID. The power and wearable structure subsystem provides regulated electrical power and maintains stable physical placement of sensors and electronics. Finally, the PC software subsystem performs UI-aware assistance and parameter tuning. Figure 2 presents the system-level architecture of the proposed design.

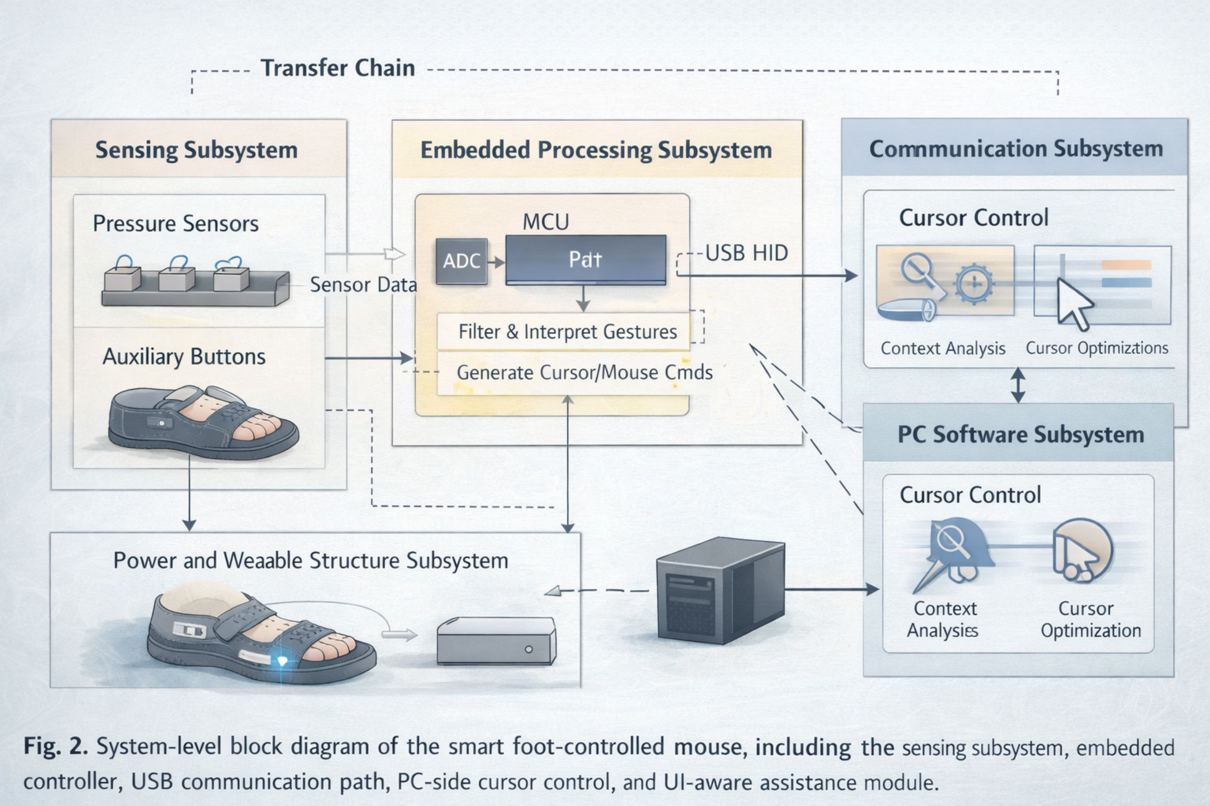

Figure 2: System-level block diagram of the smart foot-controlled mouse, including the sensing subsystem, embedded controller, USB communication path, PC-side cursor control, and UI-aware assistance module. This figure was AI-generated for documentation of the proposed architecture.

2.2 Physical Design

The physical prototype will take the form of a slipper-like wearable interface. Four force-sensitive resistors will be placed under major pressure regions of the foot to detect directional intent, while auxiliary buttons may be placed at accessible positions for explicit command triggering if needed. The microcontroller board, power-conditioning circuit, and wiring will be fixed to a lightweight frame or side-mounted enclosure attached to the slipper structure. The design will prioritize mechanical stability, user comfort, and ease of repeated use. The physical structure must keep sensors aligned with the intended foot-contact regions and prevent excessive shifting during operation. This design direction is also supported by recent wearable tactile sensing toolkits demonstrated on gait-monitoring shoe soles [1]. Figure 3 shows the teams preliminary hand-drawn structural concept of the wearable device.

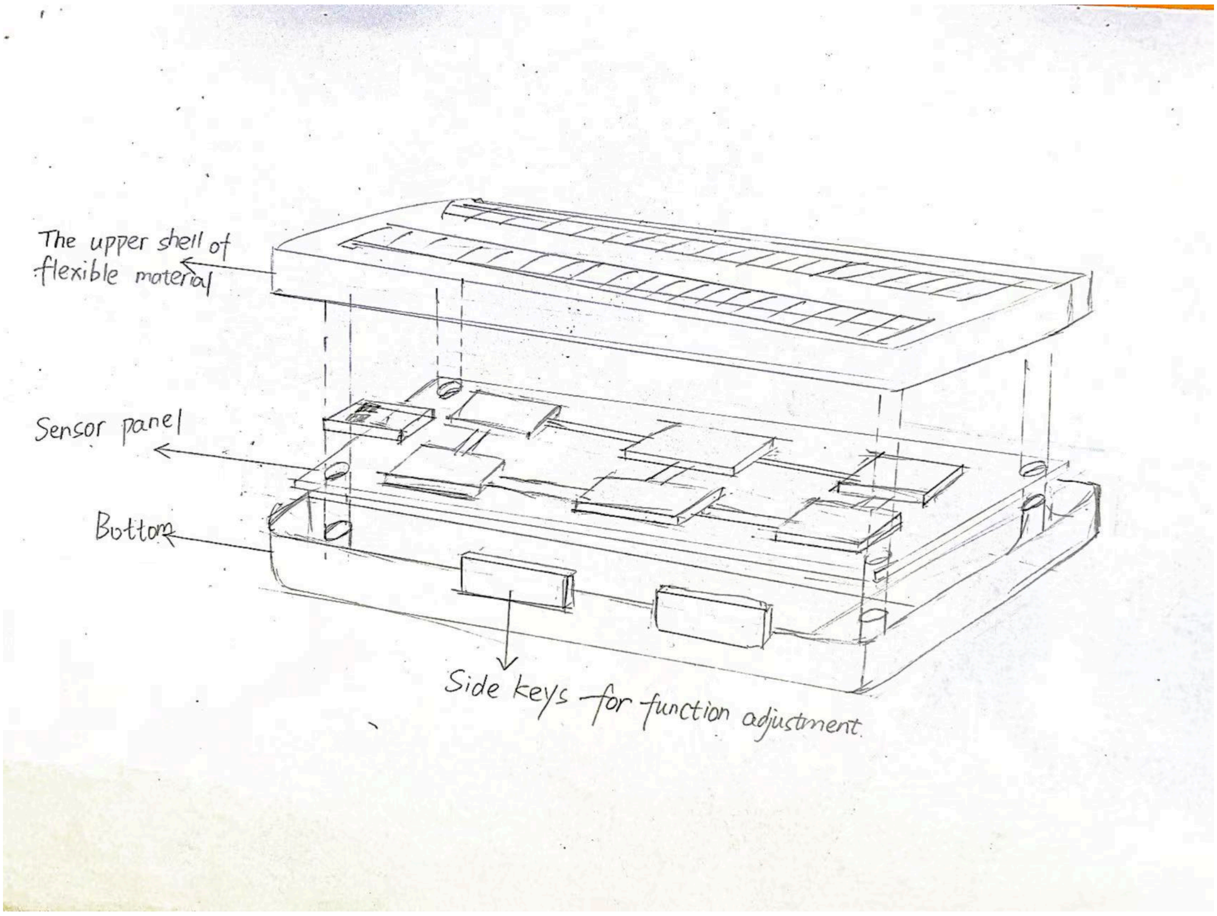

Figure 3: Preliminary hand-drawn physical design sketch of the smart foot-controlled mouse, showing the upper shell, sensor panel, buttons, and side keys for function adjustment.

2.3 Subsystem 1: Sensing Subsystem

The sensing subsystem is responsible for detecting foot pressure distribution and translating it into measurable analog electrical signals. The current design uses four force-sensitive resistors as the main continuous sensing elements. These sensors are positioned to capture left-right and front-back pressure changes, which are used to infer intended cursor movement. Additional buttons may be included to simplify explicit commands such as click or drag initiation if gesture-only triggering is found to be insufficiently reliable. Commercial FSR elements are suitable for this role because they are thin polymer thick-film sensors whose resistance decreases with applied force and are intended for human-machine interface type applications [8].

The subsystem contributes directly to the overall design by serving as the primary human input layer. Without stable and distinguishable pressure readings, the rest of the system cannot generate reliable cursor motion or mouse commands. Therefore, the sensing subsystem must provide repeatable, low-noise, and sufficiently separated signals for different foot gestures.

Sensing Subsystem Requirements

- The subsystem shall provide at least four independent analog sensing channels for foot pressure acquisition.
- After filtering, the short-term noise variation of each sensing channel shall be no greater than 5% of full-scale output under a constant load condition.
- Under repeated loading at the same foot position, the variation in normalized sensor reading shall remain within ±5%.

Sensing Subsystem Verification

- Connect all sensing channels to the embedded ADC and confirm that four independent channels can be read simultaneously without channel loss.
- Apply a constant reference load to each sensor for 30 s, record the filtered output, and verify that peak-to-peak variation does not exceed 5% of the sensor full-scale range.
- Repeat the same loading condition ten times at a fixed position and verify that the normalized output remains within ±5% across trials.

2.4 Subsystem 2: Embedded Processing Subsystem

The embedded processing subsystem consists of a microcontroller with ADC capability and firmware for data acquisition, filtering, calibration, gesture interpretation, and mouse event generation. Sensor readings are sampled at a fixed rate and processed using smoothing filters, normalization, and dead-zone thresholds. Directional pressure imbalance is mapped to cursor velocity, while specific gesture patterns or auxiliary button events are mapped to click, double-click, and drag operations.

This subsystem is central to the design because it converts raw foot pressure signals into meaningful control signals. It determines whether the system can respond fast enough, reject noise, and interpret user intent reliably. The embedded subsystem must therefore meet both timing and correctness requirements.

Embedded Processing Subsystem Requirements

- The controller shall sample sensing channels at a rate of at least 100 Hz.
- The embedded processing time from sensor acquisition to HID command generation shall be no more than 20 ms per cycle.
- The controller shall support a user calibration process that stores neutral pressure offset and sensitivity parameters for at least one operating profile.

Embedded Processing Subsystem Verification

- Measure the actual sampling interval using timestamped serial debug output and verify an effective rate of at least 100 Hz.
- Toggle a debug pin at the start and end of each processing cycle, observe the signal on an oscilloscope, and verify that per-cycle processing time does not exceed 20 ms.
- Run the calibration routine, store the resulting parameters, reboot the controller, and confirm that the calibration profile is successfully retained and reused.

2.5 Subsystem 3: Communication Subsystem

The communication subsystem sends cursor and mouse-event information from the embedded controller to the PC. The baseline implementation uses wired USB communication and emulates a standard HID mouse device so that the system can work with common desktop operating systems without a custom driver. This approach reduces complexity and improves development reliability for the first prototype. The design follows the standard USB HID device model, which is intended to support USB-compatible human interface devices and standard host-side extraction of HID reports [7].

This subsystem contributes to the overall design by ensuring that the embedded interpretation results are recognized by the host computer in a standard and low-latency way. If communication is unstable or incompatible, the device cannot function as a practical assistive input tool.

Communication Subsystem Requirements

- The system shall enumerate as a standard USB HID mouse on a common desktop operating system without requiring manual driver installation.
- The communication link shall support a command/report update rate of at least 50 Hz during continuous cursor movement.
- During a continuous 1-hour operation test, the subsystem shall experience no unintended disconnects.

Communication Subsystem Verification

- Connect the prototype to a PC and verify that the operating system recognizes it as a mouse-class HID device automatically.
- Send continuous cursor updates from the controller and confirm, through timestamped logging, that the delivered update rate is at least 50 Hz.
- Operate the device continuously for 1 hour and verify that no disconnect, freeze, or spontaneous reconnection occurs.

2.6 Subsystem 4: Power and Wearable Structure Subsystem

The power subsystem provides regulated power to the sensors, controller, and auxiliary circuits. In the baseline prototype, the system is powered through USB, with local regulation as needed for logic and sensing stability. The wearable structure subsystem mechanically integrates the sensors, electronics, and cables into a slipper-like device that remains stable during use and does not introduce discomfort or unsafe edges.

This subsystem supports the entire design by providing electrical stability and physical usability. Since the device is worn on the foot, mechanical fit and stability are especially important. If the structure shifts significantly or the power rail becomes noisy, sensing accuracy and user comfort will both degrade.

Power and Wearable Structure Subsystem Requirements

- The subsystem shall provide a regulated logic supply within 3.3 V ± 5% during normal operation.
- The peak-to-peak ripple on the logic supply shall be less than 50 mV under full system load.
- During 30 minutes of continuous use, the sensor placement relative to the foot-contact regions shall not shift by more than 5 mm.
- No exposed component or enclosure surface shall exceed 45°C during normal operation.

Power and Wearable Structure Subsystem Verification

- Measure the logic supply using a multimeter and verify that the voltage remains within 3.3 V ± 5%.
- Observe the regulated supply with an oscilloscope under full operating load and verify that ripple remains below 50 mV peak-to-peak.
- Mark the intended sensor positions, perform 30 minutes of representative use, and verify that the sensor locations have shifted by no more than 5 mm.
- Use an IR thermometer or thermocouple to verify that the surface temperature of the device remains below 45°C.

2.7 Subsystem 5: PC Software and UI-Aware Assistance Subsystem

The PC software subsystem receives mouse input from the embedded hardware and modifies cursor behavior to improve precision during difficult interactions. The UI-aware assistance module monitors cursor context and applies gain reduction or target-oriented assistance when the user approaches small interface elements. The module will also provide parameter tuning for sensitivity, dead-zone size, and assistance strength. Its design is informed by prior GUI target-acquisition techniques such as semantic pointing and the bubble cursor, both of which show that cursor behavior can be adapted dynamically to improve interaction with small or dense targets [4, 5]. The general motivation for improving small-target acquisition is also consistent with Fitts' law [6].

This subsystem is the main source of novelty in the project. A purely foot-based mouse can achieve basic cursor movement, but precise desktop interaction remains difficult when targets are small or visually dense. By incorporating interface-aware assistance, the system improves real-world usability rather than only demonstrating raw input capability.

PC Software and UI-Aware Assistance Requirements

- The software shall provide adjustable cursor gain and dead-zone parameters that can be modified without reflashing embedded firmware.
- When assistance mode is enabled, the system shall reduce the mean cursor speed by at least 30% within a configurable proximity region around small targets.
- In standardized target-acquisition tasks with targets of size 16 px or smaller, assistance mode shall improve successful target selection rate by at least 20% compared with baseline foot-only mode.

PC Software and UI-Aware Assistance Verification

- Open the parameter tuning interface, adjust gain and dead-zone settings, and verify that changes take effect immediately during use.
- Run a test application containing small clickable targets, record cursor velocity with and without proximity to targets, and verify at least a 30% reduction in mean speed within the assistance zone.
- Conduct repeated target-selection trials under baseline and assistance modes and verify that the successful acquisition rate improves by at least 20% for targets of 16 px or smaller.

2.8 Requirements and Verification Table

The course R&V guidance recommends that requirements and verification procedures be organized by functional block, with each requirement paired with a specific verification procedure. In addition, the verification should clearly define the equipment used, the test steps, and the success criterion. Therefore, Table 1 presents the formal Requirements and Verification table for the smart foot-controlled mouse.

Table 1: Requirements and verification table for the smart foot-controlled mouse.

ID	**Subsystem**	**Requirement**	**Equipment**	**Verification Procedure**	**Success Criterion**
S1	Sensing	Provide at least four independent analog sensing channels for foot pressure acquisition.	MCU, ADC input, PC serial monitor	Connect all sensing channels to the MCU ADC. Apply force independently to each FSR and record ADC responses through serial output.	All four sensing channels respond independently with no missing channel.
S2	Sensing	After filtering, the short-term noise variation of each sensing channel shall be no greater than 5% of full-scale output under constant load.	Reference load, MCU, PC serial monitor	Apply a constant reference load to each FSR for 30 s. Record filtered sensor output and compute peak-to-peak variation.	Peak-to-peak variation $\leq 5\%$ of full-scale range.

ID	**Subsystem**	**Requirement**	**Equipment**	**Verification Procedure**	**Success Criterion**
S3	Sensing	Under repeated loading at the same foot position, the variation in normalized sensor reading shall remain within ±5%.	Reference load, ruler/template, PC logging	Apply the same load at the same position ten times. Record normalized readings and calculate deviation from the mean.	Maximum deviation from the mean $\leq \pm 5\%$.
E1	Embedded Processing	Sample sensing channels at a rate of at least 100 Hz.	MCU, PC serial terminal	Output timestamps through the serial port for consecutive samples over at least 10 s. Compute effective sampling rate.	Average sampling rate $\geq$ 100 Hz.
E2	Embedded Processing	Processing time from sensor acquisition to HID command generation shall be no more than 20 ms per cycle.	Oscilloscope, MCU debug firmware	Toggle a debug GPIO pin at the start and end of each processing cycle. Measure pulse width using an oscilloscope.	Per-cycle processing time $\leq$ 20 ms.
E3	Embedded Processing	Support a user calibration process that stores neutral offset and sensitivity parameters for at least one operating profile.	MCU, host computer	Run the calibration routine, save parameters in non-volatile memory, reboot the device, and reload the profile.	Saved calibration profile is retained and produces the same neutral behavior after reboot.

ID	**Subsystem**	**Requirement**	**Equipment**	**Verification Procedure**	**Success Criterion**
C1	Communication	Enumerate as a standard USB HID mouse on a common desktop OS without manual driver installation.	Prototype, Windows/macOS computer	Connect the prototype to the host computer and observe device recognition. Record screenshots of device detection and cursor movement.	Host OS automatically recognizes the device as a HID mouse and cursor control functions correctly.
C2	Communication	Support a command/report update rate of at least 50 Hz during continuous cursor movement.	Prototype, host computer, logging software	Send continuous cursor reports and log report timestamps on the host side for at least 30 s. Compute average update rate.	Average update rate $\geq$ 50 Hz.
C3	Communication	During a continuous 1-hour operation test, the subsystem shall experience no unintended disconnects.	Prototype, host computer	Operate the device continuously for 1 hour while generating cursor movement and command events. Observe system stability.	No disconnect, freeze, or spontaneous reconnection occurs during the test.
P1	Power / Structure	Provide a regulated logic supply within 3.3 V ± 5% during normal operation.	DMM, powered prototype	Measure the regulated logic rail during active system operation.	Measured voltage remains within 3.135–3.465 V.

ID	**Subsystem**	**Requirement**	**Equipment**	**Verification Procedure**	**Success Criterion**
P2	Power / Structure	Peak-to-peak ripple on the logic supply shall be less than 50 mV under full system load.	Oscilloscope, prototype	Probe the 3.3 V rail while the system runs under maximum sensing and communication activity.	Ripple $<$ 50 mV peak-to-peak.
P3	Power / Structure	During 30 minutes of continuous use, sensor placement relative to the foot-contact regions shall not shift by more than 5 mm.	Ruler or caliper, prototype	Mark initial sensor locations, perform 30 minutes of representative use, then measure displacement.	Sensor displacement $\leq$ 5 mm.
P4	Power / Structure	No exposed component or enclosure surface shall exceed 45°C during normal operation.	IR thermometer or thermocouple	Run the system continuously for 30 minutes and measure exposed surface temperatures.	All exposed surfaces remain below 45°C.
U1	PC Software / UI Assistance	Adjustable cursor gain and dead-zone parameters shall be modifiable without reflashing firmware, and updated settings shall take effect within 1 s.	Host computer, software interface	Open the parameter tuning interface, modify gain and dead-zone values, and observe cursor behavior immediately after the change.	Parameter change takes effect without reflashing and within 1 s.

ID	Subsystem	Requirement	Equipment	Verification Procedure	Success Criterion
U2	PC Software / UI Assistance	When assistance mode is enabled, the mean cursor speed shall be reduced by at least 30% within a configurable proximity region around small targets.	Host computer, logging software, test application	Run a target-based test application. Record cursor velocity with assistance disabled and enabled when entering the assistance region near targets.	Mean cursor speed in the assistance zone is reduced by at least 30%.
U3	PC Software / UI Assistance	For targets of size 16 px or smaller, assistance mode shall improve successful target selection rate by at least 20% compared with baseline foot-only mode.	Host computer, target-selection test program, trial log	Conduct repeated target-acquisition trials under baseline mode and assistance mode. Compute success rate for each mode.	Assistance mode improves successful target selection rate by at least 20%.

2.9 Points Summary

According to the course R&V guidance, the points summary should be organized as a separate table and should distribute 50 demo points across the functional blocks in the system block diagram. In order to make partial credit assignment clearer during the final demo, the points are further subdivided to the requirement level in Table 2.

Table 2: Proposed points summary for final demo evaluation.

ID	Subsystem	Requirement Basis of Credit	Points
S1	Sensing	Four independent sensing channels function correctly.	3
S2	Sensing	Channel noise remains within the specified 5% bound.	4
S3	Sensing	Repeated loading produces repeatable readings within ±5%.	3
E1	Embedded Processing	Sampling rate reaches at least 100 Hz.	3
E2	Embedded Processing	Per-cycle processing latency does not exceed 20 ms.	4
E3	Embedded Processing	Calibration profile is stored and successfully reused.	3
C1	Communication	Device enumerates correctly as USB HID mouse.	3
C2	Communication	Communication update rate reaches at least 50 Hz.	3
C3	Communication	No unintended disconnect occurs during 1-hour test.	2
P1	Power / Structure	Logic rail remains within 3.3 V ± 5%.	3
P2	Power / Structure	Supply ripple remains below 50 mV.	2
P3	Power / Structure	Sensor placement shift remains within 5 mm after use.	3
P4	Power / Structure	Surface temperature remains below 45°C.	2
U1	PC Software / UI Assistance	Software parameters can be changed live without reflashing.	3
U2	PC Software / UI Assistance	Assistance zone reduces cursor speed by at least 30%.	4
U3	PC Software / UI Assistance	Assistance improves small-target success rate by at least 20%.	5
Total			**50**

2.10 Tolerance Analysis

One of the most critical challenges in this project is preventing false cursor movement caused by sensor noise and small unintended foot-pressure fluctuations. Let the normalized left-right pressure difference be defined as

$$\Delta P = \frac{P_R - P_L}{P_R + P_L + \varepsilon},$$

where P_R and P_L are the filtered right and left pressure readings and ε is a small constant introduced to avoid division by zero. Cursor motion is generated only when the magnitude of ΔP exceeds a dead-zone threshold T:

$$v_x = \begin{cases} 0, & |\Delta P| \leq T \\ k_x\,(|\Delta P| - T)\,\operatorname{sgn}(\Delta P), & |\Delta P| > T \end{cases}$$

where v_x is the horizontal cursor velocity and k_x is the motion gain.

Assume that, after filtering, the normalized short-term noise on each side is bounded by 0.02 full scale. In the worst case, the differential noise may therefore reach approximately

$$|\Delta P_{\text{noise}}| \leq 0.02 + 0.02 = 0.04.$$

To guarantee that noise alone does not create cursor motion, the dead-zone threshold must satisfy

$$T > 0.04.$$

We select

$$T = 0.06,$$

which provides a 50% safety margin above the worst-case estimated differential noise. At the same time, intentional user input is expected to generate normalized pressure differences well above 0.06, typically above 0.12 during deliberate directional loading. Therefore, the selected threshold suppresses false motion while still preserving sufficient usable control range. The need for such thresholding and calibration is also consistent with pressure-sensing practice, because FSR response depends not only on force but also on the readout electronics and mechanics of integration [8, 1].

This analysis demonstrates that the embedded dead-zone design can feasibly reject expected sensor fluctuation and maintain stable neutral cursor behavior, which is essential to the success of the system.

3 Cost and Schedule

3.1 Cost Analysis

Table 4 summarizes the estimated prototype cost for one smart foot-controlled mouse unit. To make the cost analysis more concrete, the major electronic parts are listed with specific manufacturers and part numbers wherever the implementation choice has already been narrowed down. For custom structural materials and miscellaneous passive components, estimated costs are used because these items will be fabricated or assembled by the team.

Table 4 caption: Preliminary prototype parts cost for the smart foot-controlled mouse.

Part	**Manufacturer / Part Number**	**Function**	**Qty.**	**Unit Cost**	**Subtotal**
Force-sensitive resistor	Interlink Electronics FSR 402 (30-81794)	Foot pressure sensing	4	$6.39	$25.56
Microcontroller board	SparkFun Pro Micro 5V/16MHz (DEV-12640)	ADC + USB HID control	1	$22.50	$22.50
Momentary pushbutton switch	SparkFun 12mm Square Pushbutton (COM-09190)	Auxiliary click / mode input	3	$0.75	$2.25
Prototype board	SparkFun ProtoBoard - Square 1" Single Sided (PRT-08808)	Signal conditioning / soldered prototype	2	$2.95	$5.90
USB data cable	Adafruit USB A to Micro-B Cable, PID 592	Power + USB communication	1	$2.95	$2.95
Passive components	Resistors, capacitors, headers, wire, heat shrink (team estimate)	Signal conditioning / interconnect	1 set	$8.00	$8.00
Wearable structure materials	Slipper base, EVA foam, Velcro straps, enclosure materials (team estimate)	Mechanical support / wearable housing	1 set	$12.00	$12.00
Total Prototype Parts Cost					**$79.16**

Representative current retail prices used above are based on published listings for the Interlink FSR 402 sensor at DigiKey, the SparkFun Pro Micro board, the SparkFun 12 mm pushbutton switch, the SparkFun ProtoBoard, and the Adafruit USB A-to-Micro-B cable. Custom wearable materials and miscellaneous passive parts are estimated by the team. :contentReference[oaicite:0]index=0

For labor cost, we use a more conservative student-prototype estimate of \$15/hour rather than a full professional engineering wage. Assuming each of the four team members contributes approximately 60 hours to design, build, integration, and testing, the labor cost per person is

$$\text{Labor Cost per Person} = (\$15/\text{hour}) \times 2.5 \times 60 = \$2250.$$

Therefore, the total labor estimate is

$$4 \times 2250 = \$9000.$$

The estimated grand total, including prototype parts and labor, is therefore

$$\$9079.16.$$

3.2 Schedule

Table 3 should show the weekly schedule and division of labor.

Table 3 caption: Proposed schedule and division of work for the smart foot-controlled mouse.

Week	**Main Task**	**Responsible Member(s)**
Week 1	Finalize system architecture, select MCU and sensors, define sensor layout	All members
Week 2	Build first sensing prototype, test FSR response and placement	Zhihao Cheng, Hao Liu, Chaoxiang Yang
Week 3	Implement ADC sampling, filtering, and serial debugging	Zhihao Cheng, Hao Liu
Week 4	Implement cursor mapping and HID mouse output	Hao Liu, Chaoxiang Yang
Week 5	Build wearable slipper structure and integrate hardware mechanically	Jiongye Liu
Week 6	Implement click, double-click, and drag logic; begin calibration testing	Zhihao Cheng, Hao Liu, Chaoxiang Yang
Week 7	Develop PC-side UI-aware assistance prototype	Hao Liu, Chaoxiang Yang
Week 8	Perform subsystem verification and debugging	All members
Week 9	Conduct target-selection experiments and compare assistance vs. baseline	All members
Week 10	Final integration, prepare report, demo, and presentation	All members

The work allocation follows the teams current role distribution. Jiongye Liu focuses on power hardware and wearable structure. Zhihao Cheng, Hao Liu, and Chaoxiang Yang focus on sensing hardware, communication hardware, embedded control, and software integration.

4 Discussion of Ethics and Safety

This project is motivated by the goal of improving accessibility in human-computer interaction. In this sense, it aligns with the IEEE Code of Ethics by supporting inclusive technology development and by attempting to reduce barriers faced by users with disabilities. The design objective is not only technical performance, but also fairness and practical usability.

Several ethical issues must be considered. First, the system should not make interaction less transparent to the user. Although the UI-aware assistance module modifies cursor behavior, it must not override user intent unpredictably. Assistance must therefore remain adjustable and limited to improving target acquisition rather than taking autonomous actions on behalf of the user. Second, the system should remain affordable and reproducible so that it does not become another inaccessible assistive technology solution. Third, if the software logs interaction traces for evaluation, these data should be stored only for testing and must not include unnecessary personal information.

The project also presents several safety considerations. Electrically, the system must operate only from low-voltage USB power and use regulated logic supply rails to avoid unsafe voltage exposure. Mechanically, the wearable structure must avoid sharp edges, loose components, and unstable mounting that could cause discomfort or tripping. Thermally, the controller and surrounding enclosure must remain below safe surface temperature during normal use. Cable routing must also avoid creating a snagging hazard during foot motion.

To mitigate these risks, the prototype uses low-voltage USB power only, secure wire insulation, enclosed electronics, lightweight mounting, and temperature verification during testing. The design will follow standard USB low-voltage electrical practice for the wired prototype. If a future wireless version is developed, radio-regulation considerations such as FCC Part 15 compliance would also need to be addressed. Overall, the design decisions are intended to protect both the user and the developers during assembly, testing, and demonstration.

5 References

References

[1] D. Murphy, J. Zhu, P. P. Liang, W. Matusik, and Y. Luo, "WiReSens Toolkit: An Open-source Platform towards Accessible Wireless Tactile Sensing," *arXiv preprint arXiv:2412.00247*, 2024.

[2] N. Schönwerth, A. Schmid, R. Wimmer, and N. Henze, "Exploring Foot-Operated Input for Desktop Applications," in *Proceedings of Mensch und Computer 2025*, Chemnitz, Germany, 2025, pp. 270–280, doi: 10.1145/3743049.3743069.

[3] G. Pearson and M. Weiser, "Exploratory Evaluation of a Planar Foot-Operated Cursor-Positioning Device," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Washington, DC, USA, 1988, pp. 13–18, doi: 10.1145/57167.57169.

[4] R. Blanch, Y. Guiard, and M. Beaudouin-Lafon, "Semantic Pointing: Improving Target Acquisition with Control-Display Ratio Adaptation," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Vienna, Austria, 2004, pp. 519–526, doi: 10.1145/985692.985758.

[5] T. Grossman and R. Balakrishnan, "The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Portland, OR, USA, 2005, pp. 281–290, doi: 10.1145/1054972.1055012.

[6] P. M. Fitts, "The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement," *Journal of Experimental Psychology*, vol. 47, no. 6, pp. 381–391, 1954, doi: 10.1037/H0055392.

[7] USB Implementers Forum, "Device Class Definition for Human Interface Devices (HID), Version 1.11," Jun. 27, 2000. [Online]. Available: `https://www.usb.org/document-library/device-class-definition-hid-111`

[8] Interlink Electronics, "FSR 400 Series Data Sheet," [Online]. Available: `https://www.interlinkelectronics.com/data-sheets`