

# GAMIFIED FOCUS-DETECTION TAMAGOTCHI – GROUP 31

By

Khushi Murthy

Michael Yang

Noah Yi

Project Proposal for ECE 445, Senior Design, Fall 2026

TA: Tim Jiang

December 2026

Project No. 31

## Abstract

This report presents an interactive, desk smart companion designed to reinforce student productivity through immediate behavioral feedback. The system integrates host-side activity tracking, a custom control PCB, and a visual-mechanical actuation assembly. A background host application ingests active window titles, peripheral input frequency, and OpenCV-based presence detection to compute an explainable focus score, streamed periodically over USB-serial. We will use a STM32 mounted on a custom PCB to decode the incoming data to advance a deterministic, four-state behavioral state machine. The device drives a digital display to render facial expressions and commands micro-servo linkages to process physical gestures based on user focus.



**Figure 1: Visual representation of potential finalized desk companion.**

## Contents

|                                                                    |           |
|--------------------------------------------------------------------|-----------|
| <b>1. Introduction</b>                                             | <b>5</b>  |
| 1.1 Problem Statement                                              | 5         |
| 1.2 Proposed Solution                                              | 5         |
| 1.3 Goals & Indications of Success                                 | 5         |
| <b>2. Design</b>                                                   | <b>7</b>  |
| 2.1 Host PC System                                                 | 8         |
| 2.1.1 Computer Vision & Data Processing Logic                      | 9         |
| 2.2 Power Subsystem                                                | 11        |
| 2.2.1 USB-C Current Headroom Analysis (Including Motors & Sensors) | 12        |
| 2.2.2 3.3V Logic Regulation & Thermal Dissipation                  | 12        |
| 2.3 Control Subsystem                                              | 13        |
| 2.3.1 STM32F103C8T6 Microcontroller Core                           | 14        |
| 2.4 Sensing Subsystem                                              | 15        |
| 2.4.1 HC-SR04 Proximity Sensor vs. Tactile Button                  | 16        |
| 2.5 Display & Actuation Subsystem                                  | 17        |
| 2.5.1 ILI9488 3.5-inch IPS TFT Display                             | 18        |
| 2.5.2 Mechanical Actuation (HS-311 Servo)                          | 18        |
| 2.5.3 VJP12-25F190 Vibration Motor (Stretch Goal)                  | 18        |
| <b>3. Design Verification</b>                                      | <b>19</b> |
| 3.1 Power Management Subsystem                                     | 19        |
| 3.1.1 USB-C CC Pull-Down Verification                              | 19        |
| 3.1.2 3.3V Logic Rail Regulation                                   | 19        |
| 3.1.3 Motor Transient & Rail Isolation                             | 19        |
| 3.2 Control Subsystem                                              | 20        |
| 3.2.1 USB-Serial Communication & Data Ingestion                    | 20        |
| 3.2.2 State Machine Transitions                                    | 20        |
| 3.3 Actuation & Display Subsystem                                  | 20        |
| 3.3.1 Ultrasonic Proximity Level Scaling                           | 20        |
| 3.3.2 Hardware Button Debouncing                                   | 20        |
| 3.4 Display & Actuation Subsystem                                  | 21        |
| 3.4.1 Display Refresh & Rendering Stability                        | 21        |
| 3.4.2 Servo Positioning & Control                                  | 21        |
| 3.4.3 Vibration Motor Inductive Spike Suppression                  | 21        |
| 3.5 Host Software Subsystem                                        | 21        |
| 3.5.1 Computer Vision Presence Detection                           | 21        |
| <b>4. Costs</b>                                                    | <b>23</b> |
| <b>5. Conclusion</b>                                               | <b>24</b> |

|                            |           |
|----------------------------|-----------|
| 5.1 Uncertainties          | 24        |
| 5.2 Ethical considerations | 24        |
| 5.3 Future work            | 24        |
| <b>References</b>          | <b>26</b> |

## 1. Introduction

### 1.1 Problem Statement

In the digital age and current generation, social media and other digital distractions have shortened attention spans and have caused people to get easily distracted while working. Screen based notifications or other focus apps are easily ignored and mutable. Our project is to create a physical device to help you keep focus during your work, without being invasive.

### 1.2 Proposed Solution

We propose a desktop companion pet to encourage better desk habits through simple, physical reactions instead of easily dismissable screen pop-ups. It watches for user presence and activity on the computer, then responds with different moods and movements.

The physical device is driven by a custom printed circuit board utilizing an STM32F103C8T6 microcontroller, which manages five primary subsystems:

- **Visual Display:** A 3.5-inch IPS ILI9488 TFT screen that renders animated, 16-bit color facial expressions corresponding to the user's current focus state.
- **Physical Actuation:** A single HS-311 standard servo motor that provides physical gestures, specifically actuating a mechanical tail-wagging linkage.
- **Sensing & Interaction:** An HC-SR04 ultrasonic distance sensor and a hardware-debounced tactile push-button mounted to the top of the enclosure to allow direct, localized user interaction (e.g., "petting" or snoozing)
- **Power Management:** A 15W (5V, 3A) USB-C power architecture utilizing CC pin pull-down resistors to safely supply high current to the servo and display backlight, alongside an AMS1117-3.3 LDO for stable logic power.
- **Host Communication:** Full-Speed USB 2.0 serial communication (Virtual COM port) to receive periodic state updates from the host PC.

### 1.3 Goals & Indications of Success

High-Level Requirements:

- 1) The host PC software must aggregate webcam presence detection and peripheral input rates into a 0–100 focus score, transmitting a formatted command packet to the microcontroller at a fixed interval of  $2000\text{ms} \pm 100\text{ms}$  while consuming less than 10% of the host CPU.
- 2) The custom PCB must successfully negotiate a 5V, 3A power state via its USB-C Configuration Channel pins to maintain a 3.3V logic rail within  $\pm 0.1\text{V}$  while simultaneously supporting the 800mA peak stall current of the HS-311 servo and the 90mA draw of the TFT display backlight.

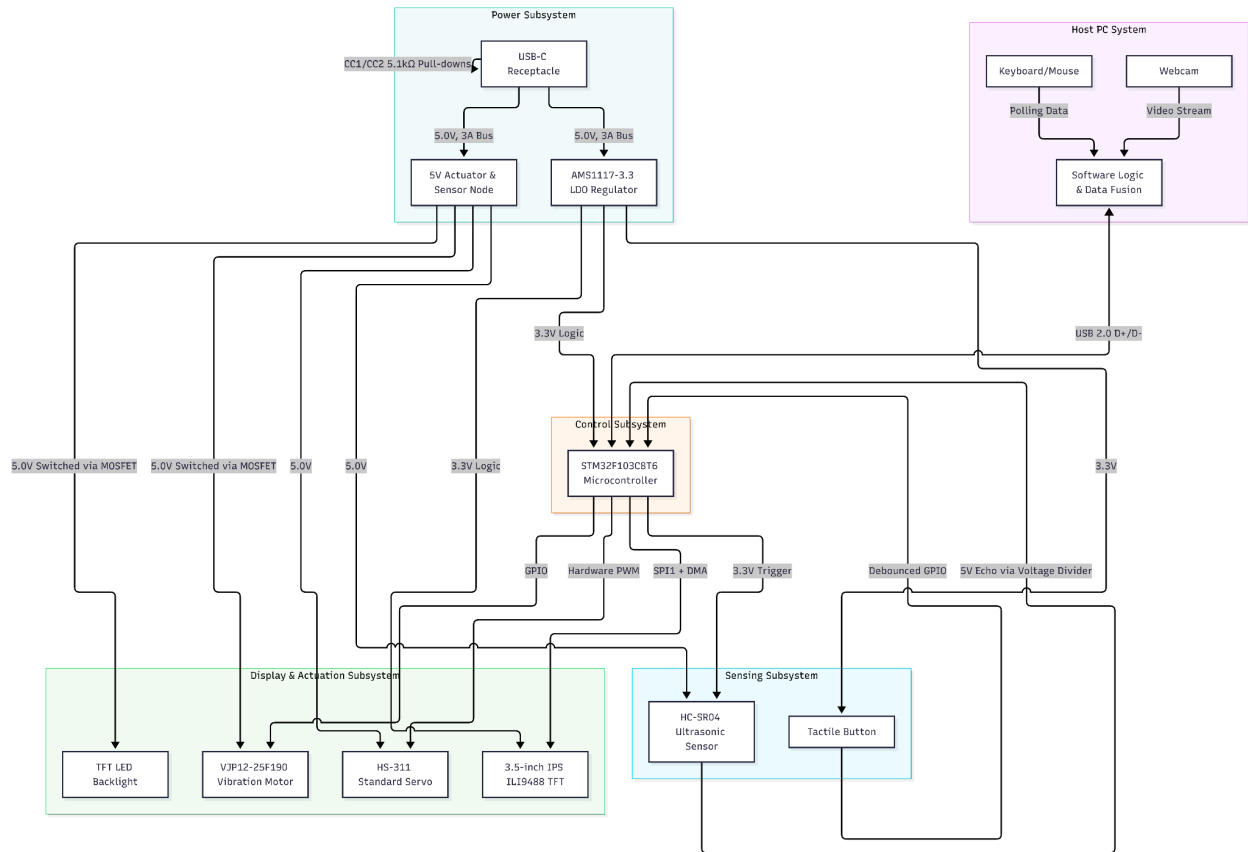
- 3) Upon receiving a state transition command, the STM32 microcontroller must utilize Direct Memory Access (DMA) to push visual facial updates to the TFT display within 100 ms and actuate the mechanical tail servo across a minimum 90-degree sweep within 500 ms to ensure immediate user feedback.

Stretch Goals:

- 1) **Haptic Feedback Integration:** Implement a VJP12-25F190 coin vibration motor driven by an N-channel MOSFET to provide physical, buzzing alerts when the user's focus score drops into a highly distracted state.
- 2) **Expanded Emotional & Behavioral States:** Develop a wider variety of specialized moods (such as energetic, curious, bored, tired, or celebratory) mapped to long-term work session data and complex desktop habits.

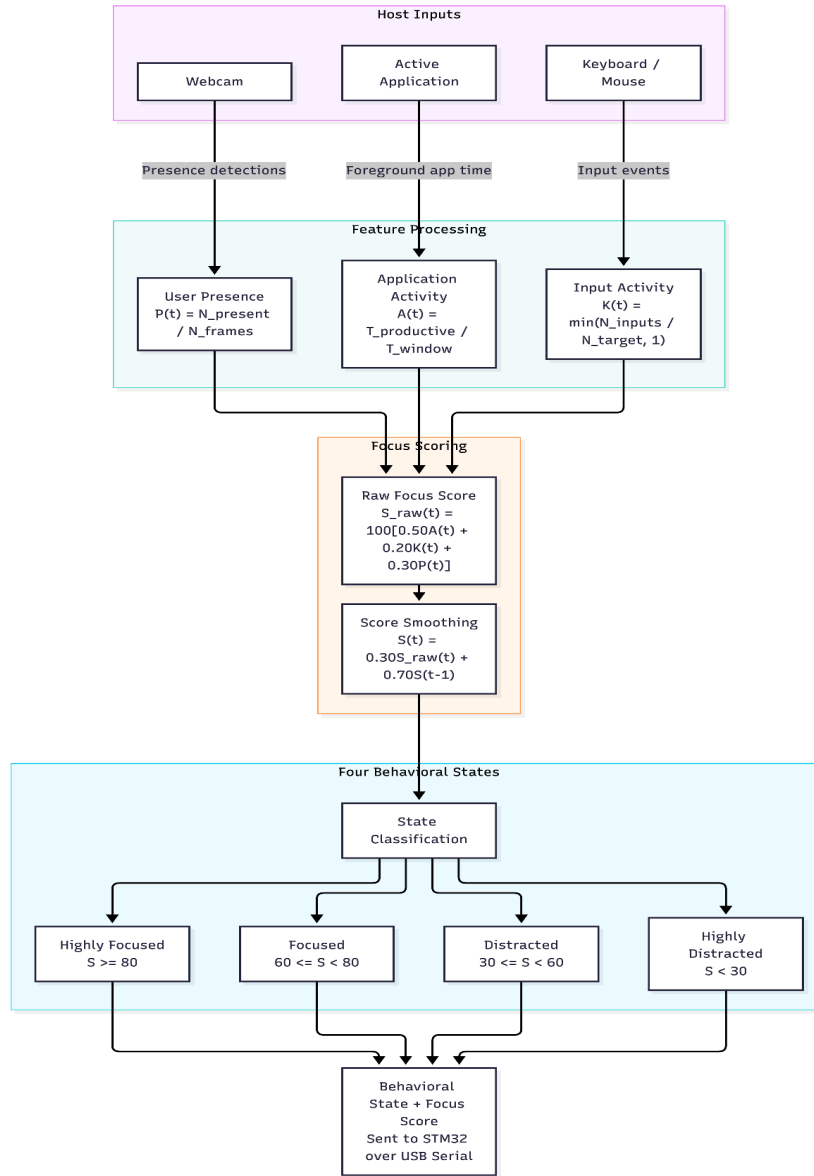
## 2. Design

The system architecture is divided into four primary functional subsystems. The Host PC System handles computationally heavy vision processing, while the custom embedded hardware manages the behavioral state machine and physical actuation.



**Figure 2: Overall system block diagram.**

## 2.1 Host PC System



**Figure 3: Host PC data processing flow.**

### Subsystem Overview:

The Host PC System performs computationally intensive activity monitoring and computer vision processing that would be unsuitable for the memory-constrained microcontroller. It collects three primary inputs: webcam-based user presence, active application information, and keyboard/mouse activity. These signals are normalized and combined into a focus score ranging from 0–100. The resulting score is smoothed to reduce short-term fluctuations, mapped to one of four behavioral states, and transmitted to the Control Subsystem over USB serial every two seconds.

## Subsystem Requirements:

- The software must capture webcam frames and execute OpenCV-based presence detection at a minimum rate of 0.5 Hz without exceeding 10% total CPU utilization on the host machine.
- The subsystem must also calculate a normalized focus score between 0 and 100 from application activity, peripheral activity, and user presence.
- The calculated focus score must map to exactly four behavioral states used by the embedded state machine.
- The subsystem must transmit a properly formatted command packet containing synchronization data, the focus score, behavioral state, and checksum over USB serial every  $2000\text{ms} \pm 500\text{ ms}$

### 2.1.1 Computer Vision & Data Processing Logic

Originally, we considered performing all activity classification on the embedded controller. This approach was not used because computer vision processing and operating-system activity monitoring would place excessive computational and memory requirements on the STM32. Instead, these tasks are performed by a lightweight host-side application.

**Application Activity:** Application activity is represented by the fraction of the scoring window during which the foreground application is classified as productive:

$$A(t) = T_{\text{productive}} / T_{\text{window}}, \quad (1)$$

where  $T_{\text{productive}}$  is the amount of time spent in applications classified as productive and  $T_{\text{window}}$  is the total duration of the scoring window.

**Keyboard and Mouse Activity:** Peripheral activity is normalized relative to a configurable target number of interactions:

$$K(t) = K_{\text{events}} / K_{\text{target}}, \quad (2)$$

where  $K_{\text{events}}$  is the number of keyboard and mouse events observed during the scoring window and  $K_{\text{target}}$  represents the activity level considered fully active. Capping the value at 1 prevents unusually high input frequency from disproportionately increasing the focus score.

**User Presence:** Rather than relying on a single webcam frame, presence is calculated as the fraction of sampled frames during the scoring window in which the user is detected:

$$P(t) = P_{\text{detected}} / P_{\text{total}}, \quad (3)$$

where  $P_{\text{detected}}$  is the number of frames in which the user is detected and  $P_{\text{total}}$  is the total number of processed frames. Using a fraction instead of a single binary observation reduces sensitivity to occasional missed detections caused by movement, lighting changes, or partial occlusion.

**Focus Score Calculation:** The three normalized measurements are combined using a weighted linear model:

$$S_{\text{raw}}(t) = 100 \times [w_A \cdot A(t) + w_K \cdot K(t) + w_P \cdot P(t)], \quad (4)$$

subject to  $w_A + w_K + w_P = 1$ . An initial weighting can be defined as giving application context the largest weight because active application usage provides a stronger indication of whether the user is performing the intended task than raw keyboard or mouse activity alone. The final weights will be adjusted experimentally during system testing.

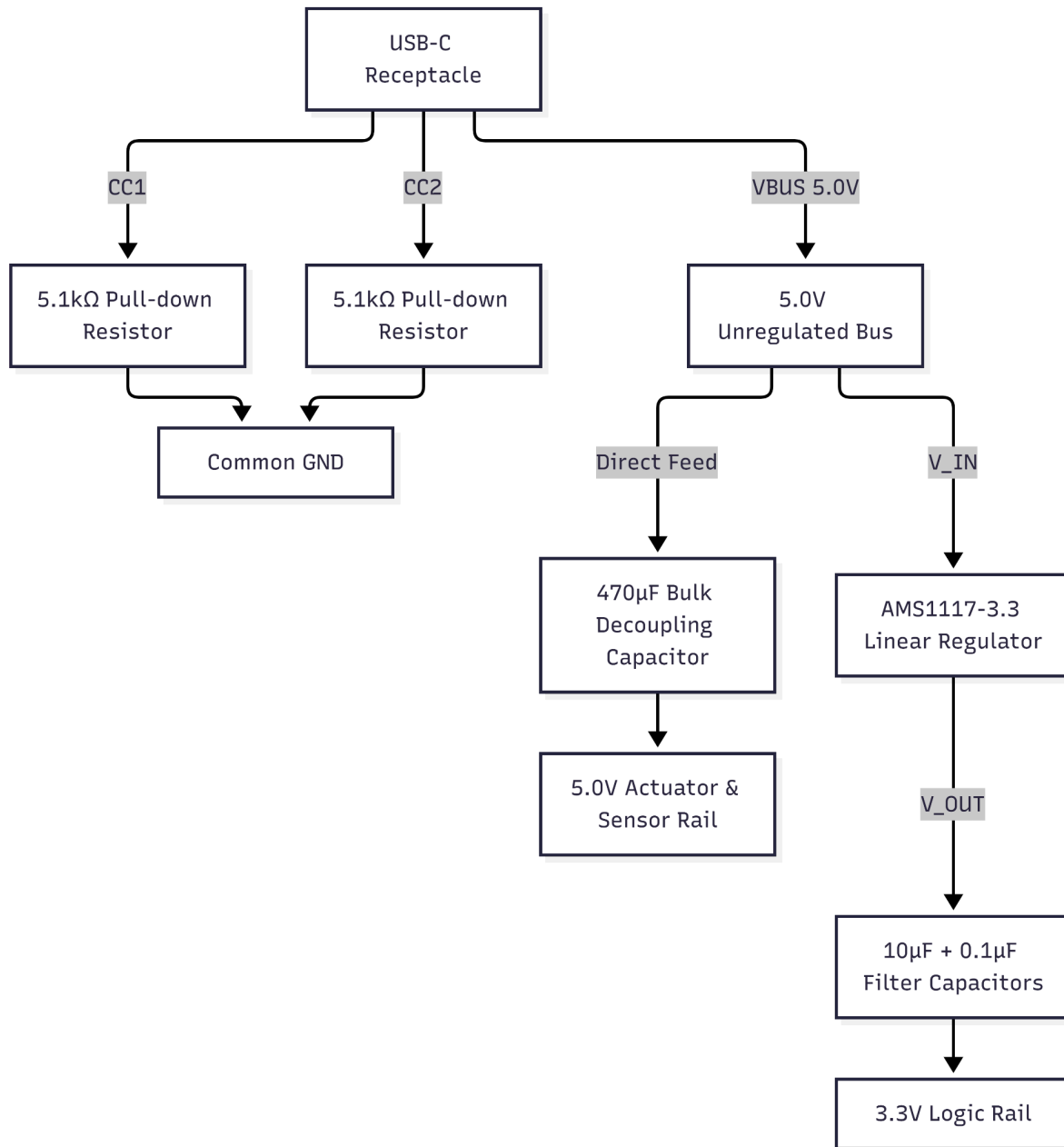
To prevent brief inactivity or noisy computer-vision measurements from causing rapid behavioral changes, the raw score is filtered using exponential smoothing:

$$S(t) = \alpha \cdot S_{\text{raw}}(t) + (1 - \alpha) \cdot S(t-1), \quad (5)$$

where  $S(t-1)$  is the previous filtered focus score and  $\alpha$  controls how strongly new observations affect the result. An initial value of  $\alpha = 0.30$  causes recent measurements to contribute 30% of the updated score while retaining 70% of the previous score.

**Behavioral Classification:** The filtered focus score is mapped to exactly four behavioral states: Highly Focused ( $S \geq 80$ ), Focused ( $60 \leq S < 80$ ), Distracted ( $30 \leq S < 60$ ), and Highly Distracted ( $S < 30$ ). These four states are transmitted to the STM32 and determine the corresponding display expression and physical actuation behavior. To further prevent rapid switching when the score lies near a state boundary, a small hysteresis margin can be applied. For example, after entering the Focused state at a score of 60, the score must fall below 55 before transitioning back to Distracted. Similar margins can be applied to the remaining boundaries. This directly addresses the proposal's concern that score boundaries should avoid erratic state flipping while remaining responsive to meaningful changes in user behavior.

## 2.2 Power Subsystem



**Figure 4: Power Subsystem architecture.**

### Subsystem Overview:

The Power Subsystem safely distributes power from the host computer to the device. To accommodate the high stall current of the HS-311 servo and the additional vibration motor stretch goal, it utilizes a USB-C receptacle to request up to 3A (15W) of passive power. It splits the 5.0V bus into a regulated 3.3V logic rail for the microcontroller/display logic and an unregulated 5.0V rail for all actuators, the sensor, and the screen backlight.

## Subsystem Requirements:

- The PCB must include independent  $5.1\text{k}\Omega$  ( $\pm 5\%$ ) pull-down resistors on both the CC1 and CC2 pins of the USB-C receptacle to successfully negotiate a 5V, 3A power state from a native USB-C host port.
- The LDO voltage regulator block must supply at least 150mA to the STM32 and ILI9488 logic components continuously at  $3.3\text{V} \pm 0.1\text{V}$ .
- The 5V actuator node must include a minimum of  $470\mu\text{F}$  of bulk decoupling capacitance to prevent transient current spikes from dropping the USB bus voltage below 4.5V.

### 2.2.1 USB-C Current Headroom Analysis (Including Motors & Sensors)

Worst-case system current draw ( $I_{total}$ ) occurs when the display is at maximum brightness, the HC-SR04 is firing, the vibration motor is active, and the HS-311 initiates movement from a dead stop.

1. **STM32F103C8T6 Logic:**  $\approx 50\text{mA}$
2. **ILI9488 TFT Backlight & Logic:**  $\approx 90\text{mA}$
3. **HS-311 Servo (Stall at 5V):**  $\approx 800\text{mA}$
4. **VJP12-25F190 Vibration Motor:**  $\approx 100\text{mA}$
5. **HC-SR04 Ultrasonic Sensor:**  $\approx 15\text{mA}$

$$I_{total} = 50\text{mA} + 90\text{mA} + 800\text{mA} + 100\text{mA} + 15\text{mA} = 1055\text{mA}$$

Available headroom ( $I_{margin}$ ) from the 3A USB-C standard:

$$I_{margin} = 3000\text{mA} - 1055\text{mA} = 1945\text{mA}$$

The maximum transient spike consumes 35.1% of the power budget, ensuring the system remains exceptionally stable even with the stretch goal hardware activated.

### 2.2.2 3.3V Logic Regulation & Thermal Dissipation

An AMS1117-3.3 linear low-dropout (LDO) regulator steps the 5V bus down for logic components. To prevent thermal shutdown in the unventilated 3D-printed enclosure, all heavy loads (servo, vibration motor, backlight) are driven directly from the 5V rail. Power dissipation of the LDO is modeled as:

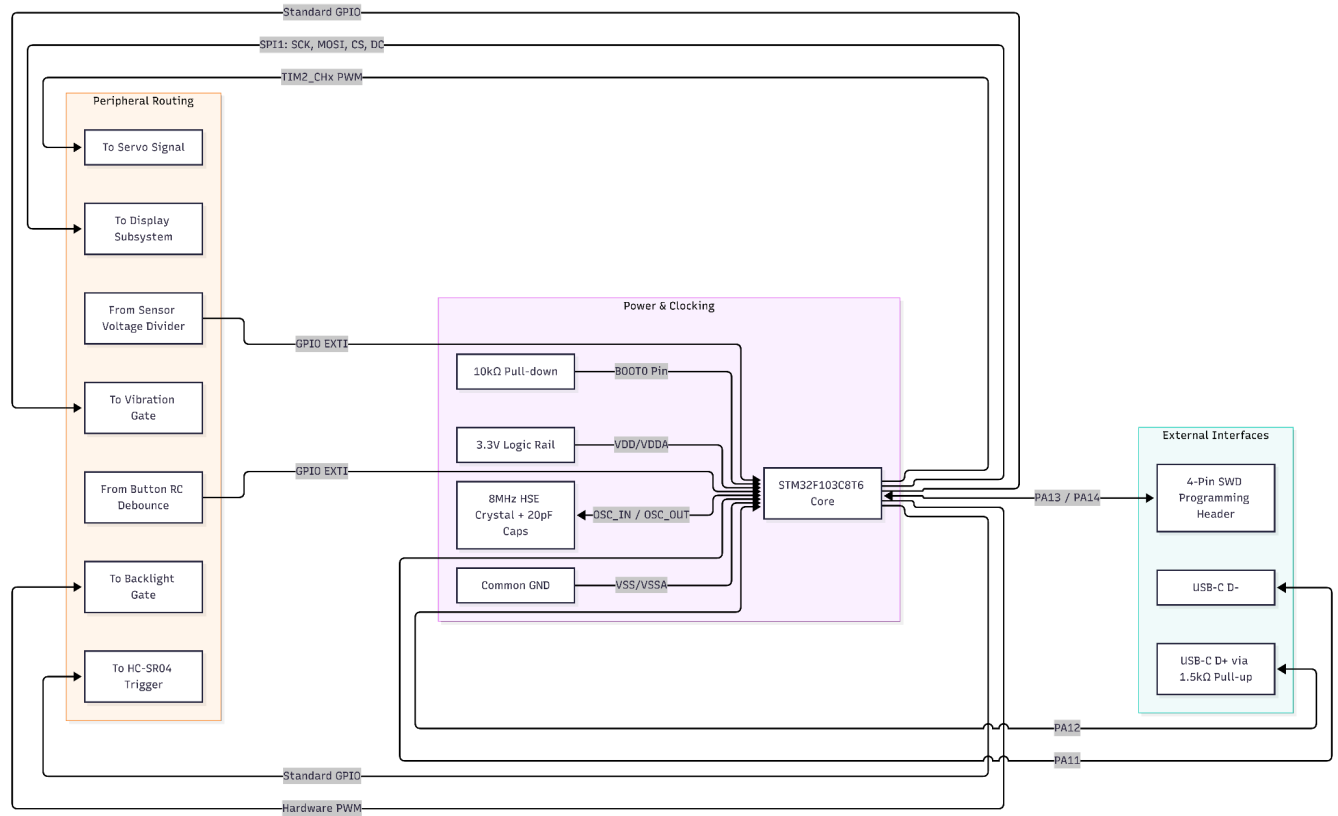
$$P_D = (5.25\text{V} - 3.3\text{V}) \times 0.050\text{A} = 0.097\text{W}$$

Assuming a maximum ambient enclosure temperature of  $40^\circ\text{C}$  and a junction-to-ambient thermal resistance ( $\theta_{JA}$ ) of  $90^\circ\text{C/W}$ :

$$T_J = 40^\circ\text{C} + (0.097\text{W} \times 90^\circ\text{C/W}) = 48.7^\circ\text{C}$$

This operating temperature remains safely below the  $125^\circ\text{C}$  maximum rating.

## 2.3 Control Subsystem



*Figure 5 Control Subsystem schematic.*

### Subsystem Overview:

Centered around the STM32F103C8T6 on a custom printed circuit board, this subsystem acts as the state-machine manager. It connects to the Power Subsystem to receive its 3.3V supply and maintains bi-directional USB 2.0 communication with the host. It translates incoming USB command packets and local sensor inputs into behavioral states, generating the necessary SPI and PWM signals for the Actuation subsystem.

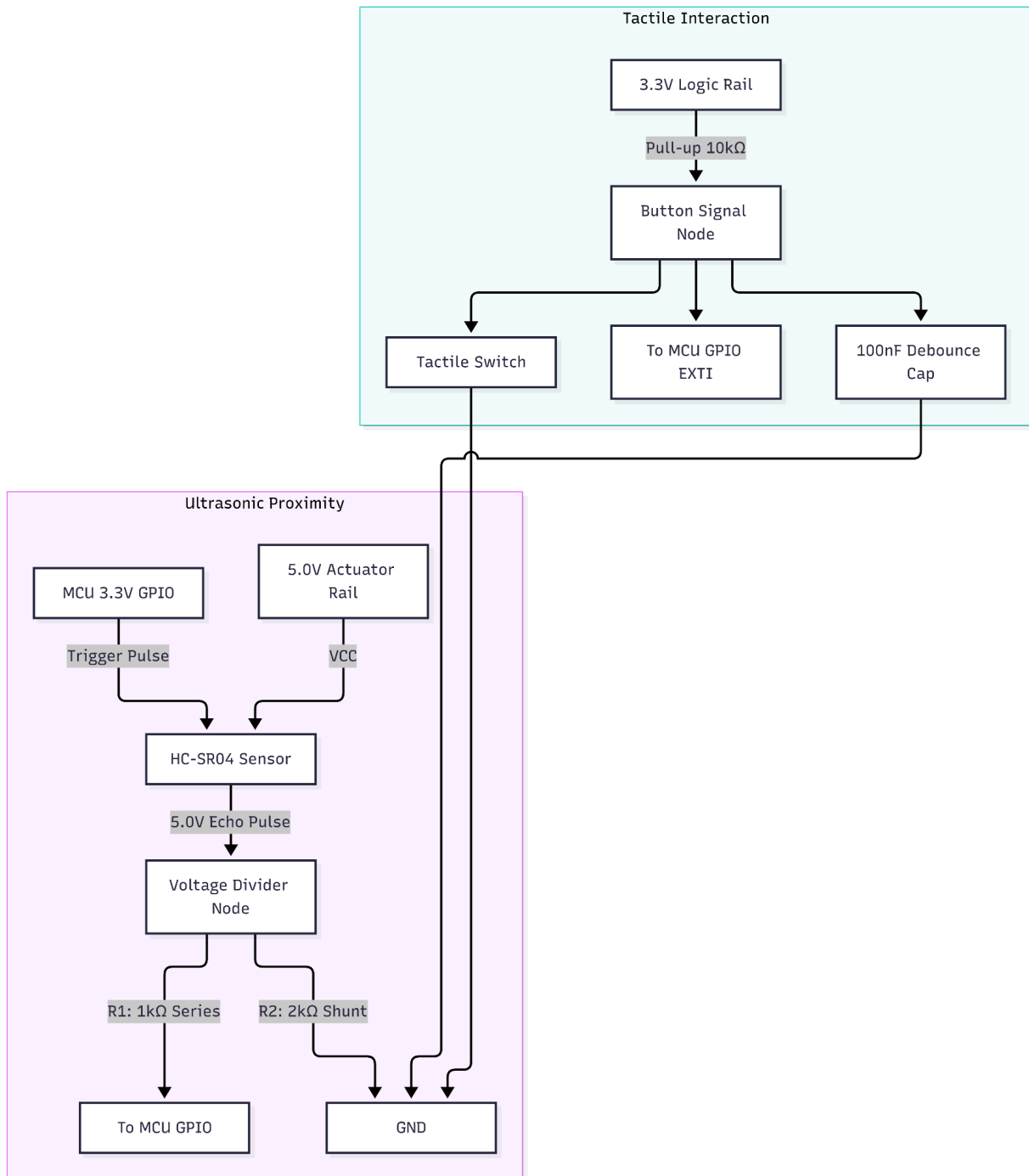
### Subsystem Requirements:

- The STM32 microcontroller must parse incoming USB command packets and evaluate internal state machine transitions within 10 ms of receiving data.
- The STM32 SPI peripheral must operate at a minimum clock frequency of 18 MHz and utilize DMA to prevent screen updates from blocking the main control loop.
- The hardware timers must generate a continuous 50Hz PWM signal for the servo control line and provide microsecond-resolution timing to measure the HC-SR04 Echo pulse.

### 2.3.1 STM32F103C8T6 Microcontroller Core

Because a 480x320 16-bit display requires ~300KB of RAM for a full frame buffer, the F103's 20KB of internal RAM requires strict memory management. The firmware utilizes Direct Memory Access (DMA) to stream small line-buffers (e.g., 64x64 pixel delta-updates) over the SPI bus. This architecture frees the CPU to simultaneously process incoming USB interrupts, poll the external user button, and calculate ultrasonic distances without introducing visual tearing.

## 2.4 Sensing Subsystem



*Figure 6: Sensing Subsystem interface.*

### Subsystem Overview:

This subsystem provides localized user interaction at the desk level, operating independently of the Host PC's webcam. It translates physical proximity or direct touch into logic-level interrupts

for the Control Subsystem, simulating the companion reacting to being "petted" or approached.

### Subsystem Requirements:

- The sensing circuitry must safely interface 5V peripheral signals with the 3.3V logic limits of the STM32 inputs.
- Physical button inputs must be hardware-debounced to prevent a single press from triggering multiple state machine transitions.

#### 2.4.1 HC-SR04 Proximity Sensor vs. Tactile Button

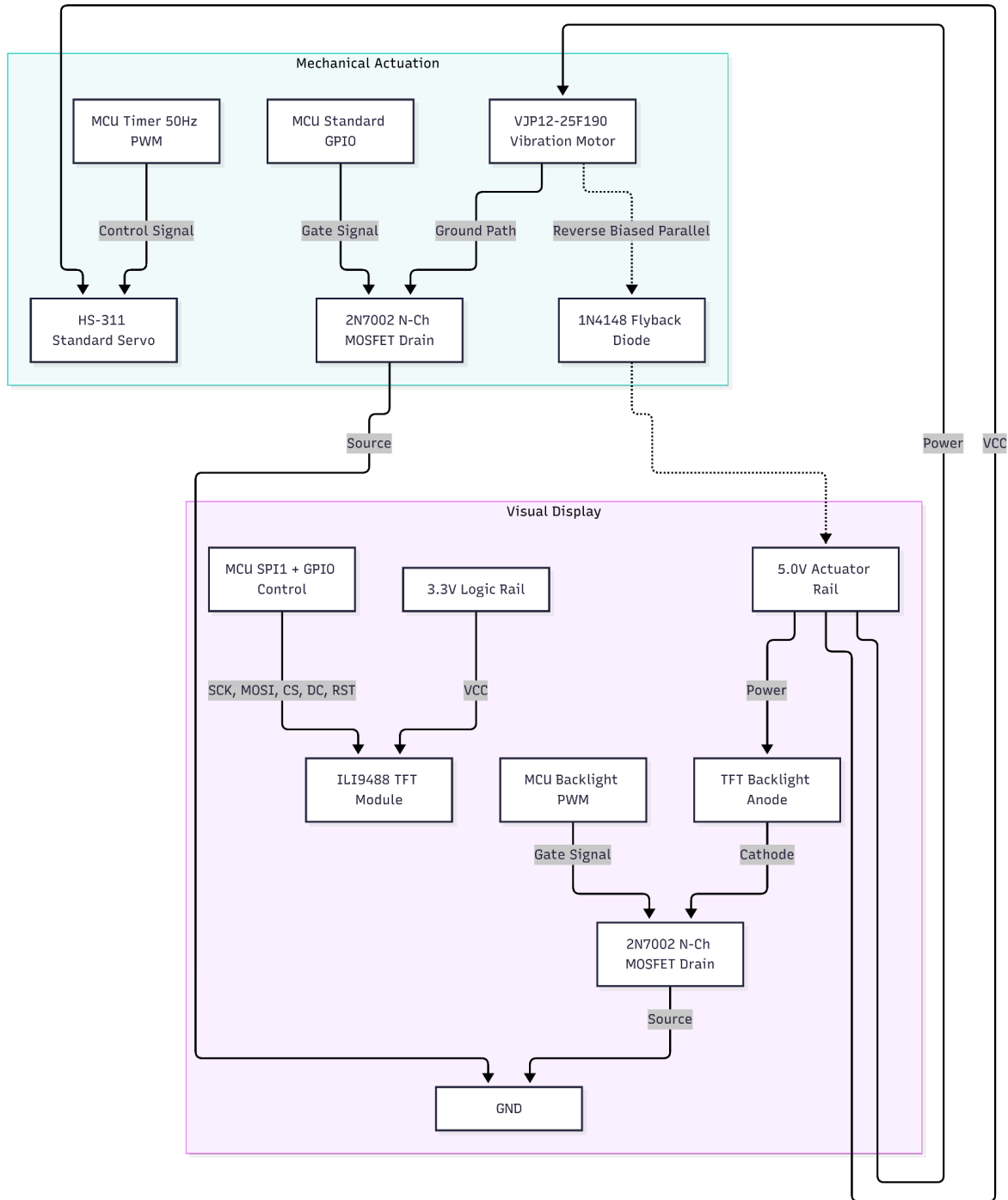
To enable direct user interaction, two alternatives were considered: a standard tactile push-button and an HC-SR04 ultrasonic distance sensor:

- The tactile button is highly reliable but requires a hardware RC debounce circuit (e.g., a 10k $\Omega$  pull-up resistor and a 100nF capacitor) to prevent signal bouncing.
- The HC-SR04 provides a more "lifelike" interaction (e.g., waving a hand over the pet) but operates on 5V logic. While the STM32 can safely provide a 3.3V Trigger signal (as the HC-SR04 interprets anything above 2.4V as high), the returning Echo pulse is 5V. To protect the STM32, a voltage divider is integrated onto the PCB for the Echo line:

$$V_{out} = V_{in} \cdot \left( \frac{R_2}{R_1 + R_2} \right) \quad (6)$$

Using  $R_1 = 1\text{k}\Omega$  and  $R_2 = 2\text{k}\Omega$ , a 5V input is successfully scaled to a safe 3.33V logic high. Both components can be mounted to the top of the enclosure and routed to the main PCB via a standard JST header.

## 2.5 Display & Actuation Subsystem



**Figure 7: Display and Actuation Subsystem circuitry.**

This physical feedback subsystem receives control signals from the Control Subsystem and power from the Power Subsystem to interact with the user. A 3.5-inch IPS SPI display renders the companion's 16-bit color facial animations, an HS-311 standard servo provides physical tail-wagging gestures, and a vibration motor provides tactile haptic feedback.

### Subsystem Requirements:

- The Display Subsystem must execute visual state transitions within 100 ms of receiving a new focus score from the Control Subsystem to maintain synchronized user feedback.
- The Actuation Subsystem must mechanically sweep the tail linkage across a minimum 90-degree arc within 500 ms upon receiving a state-transition command, without exceeding an 800mA current draw.
- The VJP12-25F190 vibration motor circuit must include an inductive flyback diode to protect the microcontroller from voltage spikes generated when the motor is disabled.

#### 2.5.1 ILI9488 3.5-inch IPS TFT Display

The ILI9488 module integrates internal Graphic RAM (GRAM) to hold the full 480x320 image buffer, requiring only spatial delta-updates over the 4-wire SPI bus. The backlight is controlled by an STM32 hardware PWM pin driving the gate of a 2N7002 N-channel MOSFET to adjust brightness dynamically based on the pet's focus state.

#### 2.5.2 Mechanical Actuation (HS-311 Servo)

The HS-311 standard servo drives a tail-wagging mechanism, powered directly from the 5V bus. The STM32 hardware timer governs the pulse width ( $T_{high}$ ), dictating the servo horn angle ( $\theta$ ):

$$T_{high} = \left( \frac{\theta}{180^\circ} \right) \cdot (T_{max} - T_{min}) + T_{min} \quad (7)$$

#### 2.5.3 VJP12-25F190 Vibration Motor (Stretch Goal)

To provide haptic feedback when the user's focus score drops, a VJP12-25F190 coin vibration motor is implemented. Because microcontrollers cannot source the ~100mA required to spin the motor, the STM32 GPIO drives an N-channel MOSFET acting as a low-side switch on the 5V rail. A 1N4148 Schottky diode is placed in parallel across the motor terminals (reverse biased). When the MOSFET turns off, this flyback diode provides a safe path for the collapsing magnetic field's reverse current, preventing a high-voltage inductive spike from destroying the MOSFET or resetting the STM32.

### 3. Design Verification

The design verification plan establishes quantitative requirements, tolerance bands, and step-by-step testing procedures for each subsystem. These procedures verify that the custom PCB, microcontroller, actuation hardware, and host software operate reliably within safe functional and electrical limits.

#### 3.1 Power Management Subsystem

##### 3.1.1 USB-C CC Pull-Down Verification

- **Requirement:** : The CC1 and CC2 lines must each provide appropriate pull-down resistance to ground to allow standard 5V power negotiation from a native USB-C port.
- **Verification:**
  1. Disconnect the board from any power source.
  2. Using a digital multimeter in resistance mode, measure the resistance from CC1 to ground, then repeat for CC2.
  3. Confirm both readings measure nominally around 5.1kohms within standard resistor tolerance.

##### 3.1.2 3.3V Logic Rail Regulation

- **Requirement:** The linear regulator must maintain a stable 3.3V output across logic loads without excessive heating under normal enclosure conditions.
- **Verification:**
  1. Power the PCB through the USB-C connector.
  2. Drive the display backlight and run active MCU routines to simulate normal operation.
  3. Measure the DC voltage between the 3.3V rail test point and ground using a multimeter to confirm its stably within standard margins.

##### 3.1.3 Motor Transient & Rail Isolation

- **Requirement:** Dynamic switching and stall events from the servo and vibration motor must not pull the main 5V rail low enough to cause microcontroller resets or inject disruptive noise into the 3.3V logic rail.
- **Verification:**
  1. Connect an oscilloscope across the 3.3V logic rail and the 5V power bus.
  2. Momentarily stall the servo to produce maximum mechanical load.
  3. Verify on the oscilloscope that voltage dips on the 5V bus do not disrupt downstream regulation and that logic rail ripple remains small enough to prevent brownout resets.

## 3.2 Control Subsystem

### 3.2.1 USB-Serial Communication & Data Ingestion

- **Requirement:** The microcontroller must reliably receive and parse incoming host telemetry packets over the USB connection without periodic dropouts or persistent checksum mismatches.
- **Verification:**
  1. Flash test firmware to echo received packet identifiers back over the serial interface.
  2. Run a host test script sending a continuous stream of formatted command packets containing sync bytes, data payloads, and checksums.
  3. Confirm via host logs that the microcontroller receives and validates the test packets without dropped transmissions.

### 3.2.2 State Machine Transitions

- **Requirement:** The microcontroller must transition states and update its output command buffers quickly upon receiving a valid host command packet so responsiveness feels instantaneous.
- **Verification:**
  1. Toggle a test GPIO pin upon packet arrival and clear it when the internal state logic finishes updating output registers.
  2. Measure the resulting pulse width on an oscilloscope.
  3. Confirm the execution time is within an acceptable window (sub-second, well below perceptible UI lag).

## 3.3 Actuation & Display Subsystem

### 3.3.1 Ultrasonic Proximity Level Scaling

- **Requirement:** The voltage divider on the ultrasonic sensor's Echo output line must safely scale the returning 5V pulse down to 3.3V logic levels.
- **Verification:**
  1. Connect an oscilloscope probe to the microcontroller's Echo input pin.
  2. Trigger the sensor to fire an ultrasonic pulse.
  3. Measure the peak voltage of the returned pulse to confirm it scales down to a safe 3.3V logic level without exceeding the pin's voltage rating.

### 3.3.2 Hardware Button Debouncing

- **Requirement:** The hardware RC filter on the user button must suppress mechanical contact bounce to prevent a single press from registering as multiple inputs.
- **Verification:**

1. Probe the debounced button signal line on an oscilloscope.
2. Press the physical button repeatedly under normal user conditions.
3. Inspect the waveform to ensure the signal transitions cleanly without switch chatter creating false intermediate edges.

### 3.4 Display & Actuation Subsystem

#### 3.4.1 Display Refresh & Rendering Stability

- **Requirement:** The display interface must transfer image frames smoothly via SPI and DMA without visual tearing, flickering, or blocking other time-critical MCU tasks.
- **Verification:**
  1. Run an animation sequence continuously cycling facial expressions.
  2. Visually verify the screen renders frames cleanly without display artifacts or freeze-ups.
  3. Toggle a debug pin at the end of each frame draw and observe the signal on an oscilloscope to verify a steady, fluid update rate.

#### 3.4.2 Servo Positioning & Control

- **Requirement:** The servo must respond smoothly to standard 50Hz PWM signals, rotating its output horn consistently according to the commanded duty cycle.
- **Verification:**
  1. Verify the PWM carrier period on an oscilloscope.
  2. Attach a visual indicator to the servo horn and issue commands corresponding to neutral and limit positions.
  3. Verify that the servo horn sweeps reliably between target positions without jitter or overshoot.

#### 3.4.3 Vibration Motor Inductive Spike Suppression

- **Requirement:** The flyback diode must suppress inductive turn-off transients across the vibration motor terminals to protect the switching transistor.
- **Verification:**
  1. Connect an oscilloscope probe across the switching transistor's drain and source.
  2. Pulse the vibration motor on and off.
  3. Observe the turn-off edge to verify that the flyback diode clamps inductive spikes safely near the 5V supply rail.

### 3.5 Host Software Subsystem

#### 3.5.1 Computer Vision Presence Detection

- **Requirement:** The host vision script must promptly detect when a user sits down or steps away from their desk under ambient room lighting, while maintaining reasonable host CPU usage.

- **Verification:**

1. Run the background detection pipeline under typical desk lighting conditions.
2. Perform multiple trials entering and leaving the webcam frame, noting the elapsed time before state transitions register in the application log.
3. Monitor system resource usage in the host task manager to confirm the script runs efficiently in the background.

#### 4. Costs

**Table 1 Parts Costs**

| <b>Part</b>                                                                                                                                          | <b>Manufacturer</b> | <b>Retail Cost (\$)</b> | <b>Bulk Purchase Cost (\$)</b>                                                                | <b>Actual Cost (\$)</b> |
|------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|-------------------------|-----------------------------------------------------------------------------------------------|-------------------------|
| XR-57P Plastic Electronics Enclosure<br><br><a href="https://www.polycase.com/xr-57p">https://www.polycase.com/xr-57p</a>                            | PolyCase            | 11.55                   | 1 - 9: \$11.55<br>10 - 49: \$10.09<br>50 - 99: \$8.78<br>100 - 249 \$7.71<br>250 - 499 \$7.24 | 11.55                   |
| HS-311 Servo Motor                                                                                                                                   | Servocity           | 16.99                   | 16.99                                                                                         | 16.99                   |
| IPS SPI Module 3.5 Inch TFT LCD Colorful Display Screen ILI9488 Drive IC Controller<br><br>IPS-9488 No Touch<br><br><a href="#">Link to purchase</a> | LiangHaoCai         | 19.78                   | 19.78                                                                                         | 19.78                   |
| <a href="#">Button</a>                                                                                                                               | APIELE              | 8.99                    | 8.99                                                                                          | 8.99                    |
| Ultrasonic Distance Sensor - HC-SR04                                                                                                                 | N/A                 | 6.04`                   | 6.04                                                                                          | 6.04                    |
| Vibration Motor - VJP12-25F190                                                                                                                       | Vytronics Inc       | 3.92                    | 3.92                                                                                          | 3.92                    |
| Multicolor Flashing LEDs (x8)<br><br>(ECE Supply Center)                                                                                             | BlueCell            | 0.85 each               | 0.85 each                                                                                     | 6.80                    |
| <b>Total</b>                                                                                                                                         |                     |                         |                                                                                               | <b>74.07</b>            |

## 5. Conclusion

### 5.1 Uncertainties

Our main priority is to make sure that the status tracking is robust yet accurate. We need to ensure that the computer vision pipeline reliably detects desk presence across dynamic real-world environments, such as changing ambient light levels, partial face occlusions, or non-standard webcam placement angles. Similarly, we need to fine tune the scoring system to find the optimal timing windows and score boundaries required to prevent rapid state flipping (hysteresis), ensuring the pet's behavioral transitions feel natural and encouraging rather than erratic or annoying. How exactly those measurements will be determined depends on testing and calibration once we develop our system, with a baseline setup serving as the starting point. We have a lot of factors to consider as weights, and finding a way to balance active application titles, keyboard/mouse polling activity, and webcam presence detection into a single reliable metric without triggering false distraction states during naturally low-input tasks (such as reading documentation or watching instructional videos) is an important goal that is not fully defined yet.

### 5.2 Ethical considerations

**Privacy and Surveillance Concerns:** The system utilizes a host webcam for desk-presence detection via computer vision and monitors active application titles on the desktop. IEEE Principle 1 and 9 focus on the welfare and safety of users, and we need to ensure that users do not feel like they are under surveillance. To address potential privacy anxieties, all video processing must occur entirely locally on the host machine in real-time, with raw video frames discarded immediately and never stored, logged, or transmitted externally. By extension, users should have full visibility and control over when the tracking software is active. Clear indicators (such as an LED light on the Tomogatchi) should show when monitoring is enabled, allowing users to easily pause or disable tracking during sensitive tasks (e.g., entering passwords or personal browsing)

**Psychological Impact:** Gamified productivity tools risk causing anxiety or guilt if users feel penalized or shamed by the pet's negative emotional states during burnout periods. IEEE Principles state that fairness and psychological wellbeing need to be emphasized in applications, so the system should encourage healthy work habits rather than toxic over-productivity, including adding in forgiving thresholds for temporary breaks.

**Environmental Sustainability:** The prototype involves electronic hardware (microcontrollers, PCBs, servos) and 3D-printed plastic enclosures. Future iterations should consider biodegradable or recycled filaments and energy-efficient power management to minimize e-waste.

### 5.3 Future work

**Advanced Biometric Integration:** We could incorporate additional sensors such as wearable heart-rate variability (HRV) monitors to measure stress and fatigue levels more accurately than just software heuristics

**Personalized ML Models:** We could transition from static focus thresholds to adaptive models that learn an individual user's unique work patterns, typing cadences, and focus rhythms over time to better represent individual characteristics, tendencies, and attributes

**Expanded Hardware Components:** We could add more micro-servos and multi-color LED configurations to enable a wider variety of expressive physical gestures, sounds, and emotional displays to enhance user experience.

## References

IEEE, "IEEE Code of Ethics," IEEE Policy 7.8. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8>