LivePin

Safiya Bainazar, Crystal Huang, Tedd Jung

Department of Electrical and Computer Engineering,
Carnegie Mellon University

***Abstract*—**LivePin is a cost-effective 3D reconstruction system that transforms camera-captured objects into impressions on a motorized pin art board. Using 32 servos driven by real-time depth data, each pin automatically adjusts to create precise 3D forms with ±0.25 cm height accuracy and a 50 mm displacement range. The system operates with end-to-end latency under 120 seconds, enabling smooth, real-time interaction. Modernizing the nostalgic pin art toy, LivePin delivers dynamic, hands-free 3D displays that are significantly more affordable and energy-efficient than conventional 3D technologies, providing an innovative and sustainable solution for museums, art installations, and educational environments.

***Index Terms*—**depth sensing, embedded systems, mechatronics, servo motor control, 3D visualization, depth cameras, rack-and-pinion, CAD, 3D printing, servo motor, stepper motor, gantry

I. Introduction

Modern digital media and interactive displays are overwhelmingly two-dimensional, limiting how people perceive and engage with depth and spatial content. D is essential to fully convey physical form. Existing 3D visualization technologies, such as holographic, volumetric, or autostereoscopic displays, are expensive and inaccessible to most users.

LivePin is an affordable alternative that transforms the image of an object into a physical, tactile 3D display. Using a depth camera to capture an image, the system reconstructs a 3D surface by actuating a 32 x 32 array of pins that form the captured object's shape and depth. Each pin's height corresponds to a measured depth point, forming 3D relief that can be visually observed as a physical surface. The image is developed sequentially, row by row, using servo driven rack and pinion actuators controlled by an STM32 microcontroller, with a Raspberry Pi handling image capture and processing. While the actuation process is not real time, it allows users to see the three-dimensional structure emerge gradually and physically.

Compared to competing systems such as MIT's inFORM [1] or commercial holographic displays, LivePin offers a modular, low-cost, and visually intuitive alternative. LivePin's design prioritizes accessibility, education utility, and physical visualization over speed, creating a platform that can demonstrate the relationship between depth sensing, computation, and mechanical actuation. The project's goals are to accurately reproduce 3D surfaces with less than 2.5 mm height error for 95% of pins, complete full scene rendering within 120 seconds, and maintain a safe, scalable, and robust system architecture.

II. Use-Case Requirements

The LivePin system aims to transform real-world objects into 3D impressions through automated pin actuation, suitable for museums and galleries. To achieve this, several quantitative requirements in resolution, latency, and accuracy must be met to ensure interactive and reliable performance.

Figure 1. Original Greyscale Image

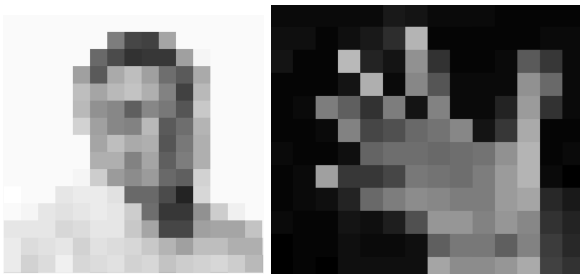

Figure 2. Reduced to 16 x 16 Greyscale Resolution

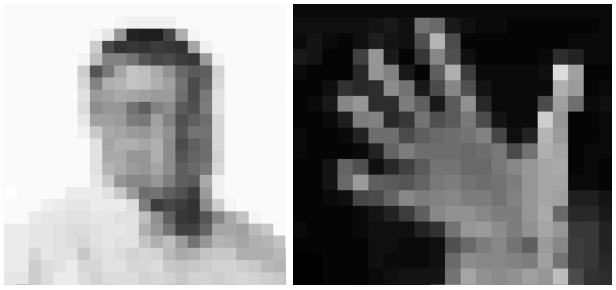

Figure 3. Reduced to 24 x 24 Greyscale Resolution

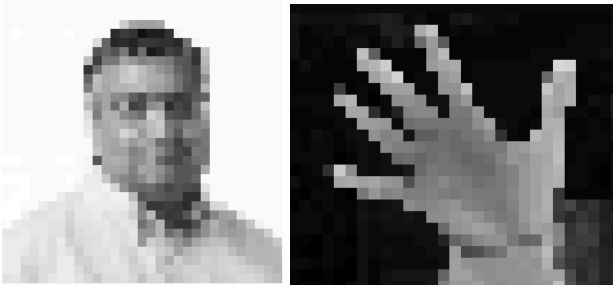

Figure 4. Reduced to 32 x 32 Greyscale Resolution

The system must be a 32 x 32 pin array (1024 pins), which would provide sufficient resolution to model recognizable objects while remaining cost-effective and energy efficient. Such resolution is illustrated in figures 1 – 4. When the image was reduced to a 16 x 16 or 24 x 24 resolution, many distinct

facial features were blurred and it is difficult to definitively say the image is of a pixelated hand. While you can distinguish that there are 5 fingers, it would be difficult for a user to distinguish if it is a hand or something else with 5 appendages. The 32 x 32 resolution image is able to display distinct facial features, allowing users to easily recognize the person and hand displayed. The arrangement of pins is comparable to traditional toy pin art boards. Each pin will support a 50 mm displacement range, enabling detailed depth reproduction across diverse shapes and surfaces. We chose this 50 mm displacement range to allow a wide variety of depths while also giving us a reasonable distance to display an image. Depth differences below 10 mm are difficult to perceive visually. Depth differences around 50 - 60 mm is enough to convey contrast and represent silhouettes and subtle topological differences. Furthermore, most pin toys also have a pin displacement of around 50 mm.

To ensure the engagingness of the device, all pins must complete actuation within 128 seconds of image capture, achieving an end-to-end latency under 120 seconds. Each pin actuator will actuate in ≤4 seconds per row, allowing for smooth and synchronized motion during image updates and ensuring a pleasant experience for the user.

For perceptual accuracy, 95% of pins must reach their target height within ±0.25 cm. This precision maintains recognizable 3D representations in line with Johnson's criteria [4], which indicate that faces become unrecognizable when more than 25% of key samples are inaccurate. For LivePin, this means that we must maintain pin accuracy high enough (≤5–10% errors) so that features are no longer recognizable.

Together, these requirements ensure that LivePin provides a responsive, precise, and sustainable 3D display experience suitable for museums, art installations, and educational environments.

III. Architecture and Principle of Operation

A. Overall Physical System

Our device is a depth-driven pin display that renders a 3D relief using actuators. The overall physical system is shown in Fig. 5 and Fig. 6. A rigid V-slot frame supports a leadscrew gantry and a carriage with 32 actuators that concurrently set pin heights across a pin array. The carriage moves down the Z-axis to actuate a selected row to its commanded heights, retracts, and the system indexes to the next row. This row-wise actuation scales naturally to 32 x N without increasing actuator count. The reset mechanism is mounted on the pin board (Fig. 7) , as a piece of polycarbonate with 2 push pull actuators. The actuator carriage will move to a safe position before the reset actuators retract the polycarbonate, causing all the pins to be pushed back.

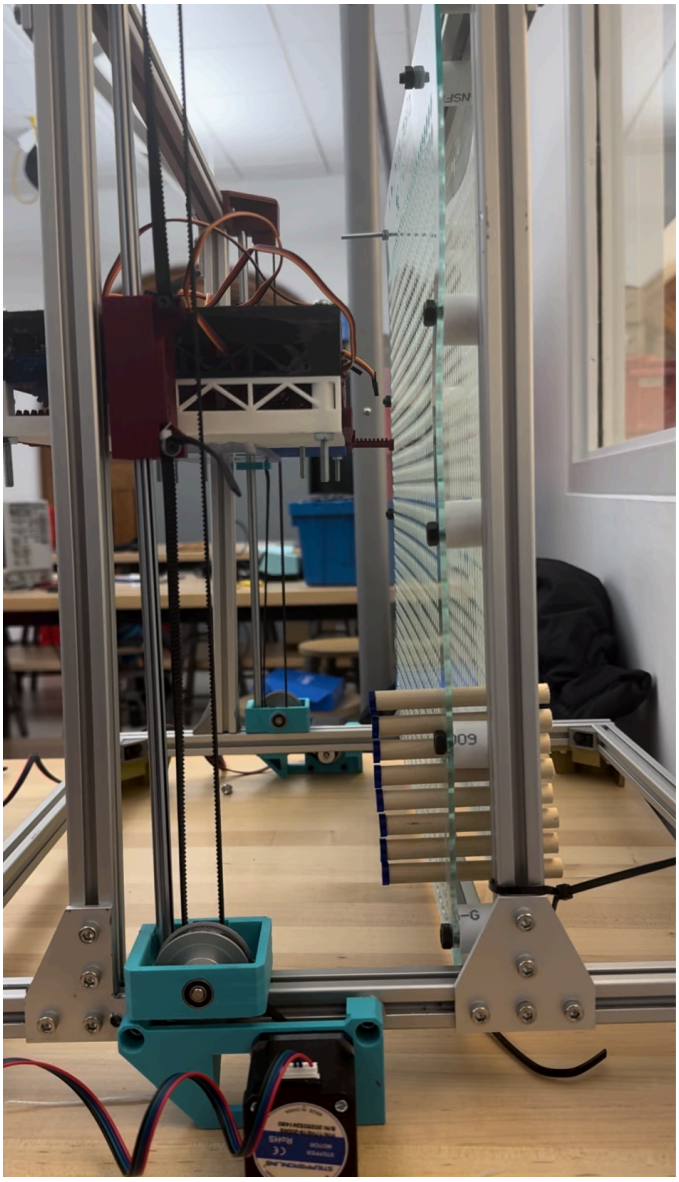

Figure 5. The Gantry

The Gantry (Figure 5) is a belted system, powered by two nema 17 stepper motors on 3:1 gearboxes. This gear reduction was optimal for increasing torque, which was necessary when moving the heavy carriage up the gantry. The Gantry system is also equipped with 3D printed belt tensioners and linear rods, which are responsible for the travel of the carriage.

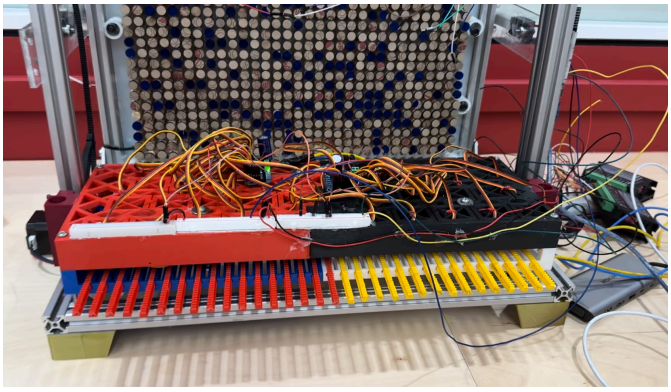

Figure 6. The Carriage

The Carriage (Figure 6) is a 3D printed housing for all the actuators. The actuators consist of a G90 positional servo, 3D printed racks and 3D printed pinions. The carriage also has two linear ball bearings press fit into their mounts that allow for it to freely travel up and down the linear rod.

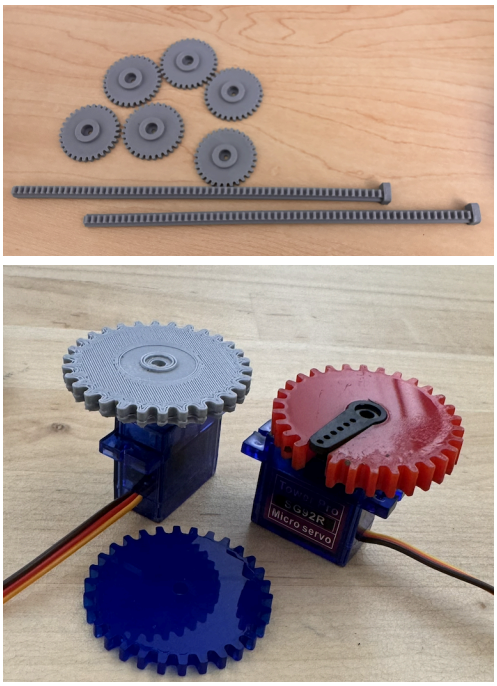

Figure 23. Rack and Pinion

A key component of our system is the rack and pinion mechanism (Figure 23). In each actuator, a position-controlled servo motor drives a circular pinion gear that meshes with a linear rack. When the servo rotates, the pinion gear turns about its axis, and the engagement between the pinion teeth and the rack teeth converts this rotational motion into linear translation of the rack. Clockwise rotation of the pinion advances the rack forward, while counterclockwise rotation retracts it. The linear displacement of the rack is directly proportional to the angular rotation of the servo, determined by the pinion pitch radius. This deterministic kinematic relationship allows the system to map a commanded servo angle to a predictable pin height.

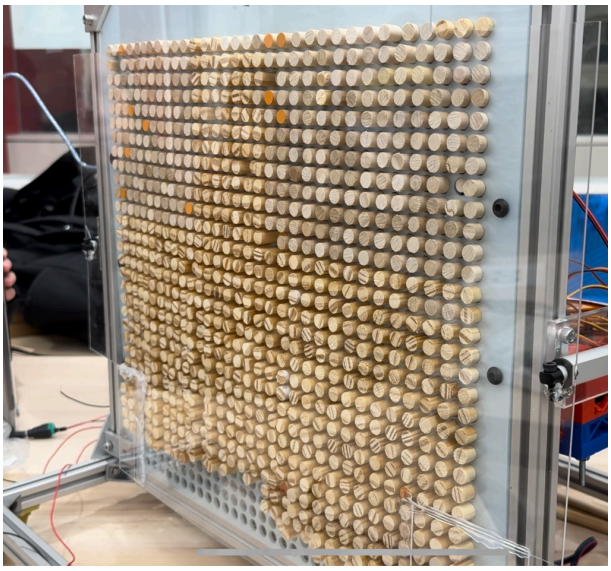

Figure 7. The Pin Board

The Pin Board (Figure 7) consists of the housing for all 1024 pins and the reset mechanism. The reset mechanism is powered by two linear actuators. A piece of acrylic is mounted on to the actuators and allows for the pins to be reset and another image to be created.

Figure 8. Camera Setup

The Camera Setup (Figure 8) consists of a depth camera and monitor.

B. Changes Since the Design Report

Since the initial design report, several refinements were made across the LivePin system as the project progressed from conceptual design to full system integration. While the overall system architecture remained unchanged, practical fabrication, integration, and testing revealed opportunities to improve robustness, manufacturability, and reliability. In total, four major design changes were implemented: updates to the frame, carriage, actuator selection, and pinion fabrication.

First, the frame was redesigned to increase overall rigidity and stability. The base footprint was expanded to better support the increased mass of the actuator carriage and pin board, reducing vibration during gantry motion. During integration, we also determined that one set of frame supports included in the original design was unnecessary, as the remaining aluminum extrusion structure provided sufficient stiffness. Removing these supports simplified assembly and reduced weight without compromising structural integrity.

Second, the carriage underwent significant redesign to accommodate larger and more robust pinions. The updated carriage includes strategically placed cutouts that reduce overall mass while maintaining structural strength. These cutouts also improved cable routing and accessibility, simplifying wiring and reducing strain on electrical connections. The lighter carriage reduced load on the gantry system and improved motion reliability.

Third, we switched from continuous-rotation servos to position-controlled servos for pin actuation. Although continuous servos were initially attractive due to their simplicity, testing revealed that time-based motion control introduced unacceptable variability in pin positioning due to differences in load, friction, and motor speed. Position-controlled servos provide closed-loop angular control, allowing each pin to be driven to a precise and repeatable height using a single PWM command. This change significantly improved actuation determinism and reduced cumulative error. The rationale for this decision is discussed in more detail in the design trade-off analysis.

Finally, the pinions were redesigned to improve mechanical durability. Early prototypes used 3D-printed pinions, which frequently failed under load due to limited strength and inconsistent print quality. To address this issue, we transitioned to laser-cut pinions, which provided greater rigidity, tighter tolerances, and improved consistency across all actuators. This change reduced mechanical failures and improved long-term reliability of the rack-and-pinion mechanism.

Together, these design updates reflect the transition from a proof-of-concept design to a more robust, manufacturable system. Each change was driven by empirical testing and integration experience, resulting in improved reliability, ease of assembly, and overall system performance.

C. Principles of Engineering

LivePin applies core principles of mechanical and electrical engineering to convert sensed depth information into controlled physical motion. Mechanically, the system relies on rack-and-pinion actuation, servo motor control, and structural load distribution to convert rotational motion into precise linear displacement across a 50-millimeter range. The gantry and carriage design ensure repeatable alignment and stable motion while supporting the combined mass of the actuators and pins.

Control-system engineering principles govern how motion is sequenced and coordinated across the device. Pins are actuated row-by-row to limit actuator count while maintaining scalability, and safety-oriented design choices such as emergency stop behavior, reset sequencing, and safe-position motion planning prevent mechanical interference. Embedded systems engineering principles guide the partitioning of computation and control between the Raspberry Pi and the microcontroller, including deterministic timing, UART-based

synchronization, and structured command protocols to ensure reliable system operation.

D. Principles of Science

LivePin relies on scientific principles from optics, physics, and computer vision to sense and physically represent the three-dimensional structure of real-world objects. The Intel RealSense D455 depth camera applies stereo infrared imaging, estimating depth by measuring pixel disparities between paired image sensors. This process is governed by geometric optics and triangulation principles, producing a dense depth map in which each pixel corresponds to a measured distance from the camera to a point in the scene.

Scientific principles of signal noise and uncertainty also influence depth acquisition. Infrared sensing is subject to measurement noise, surface reflectivity variations, and occlusions, which can introduce depth bias or outliers. To address these effects, the vision pipeline applies spatial filtering to suppress high-frequency noise and improve depth stability before actuation. Region-of-interest selection further isolates the relevant physical object, reducing interference from background depth measurements and improving consistency.

Human perception and physical visualization limits also inform the system's scientific design. Small depth differences below approximately 10 mm are difficult for users to perceive, while depth variations on the order of 50 mm provide sufficient contrast to convey shape and topology. These perceptual constraints guided the choice of pin displacement range and justified downsampling the depth image to a 32×32 grid, preserving recognizable features while remaining within mechanical and power constraints.

Finally, classical mechanics underpins the physical interaction between the pins and the rendered object shape. Gravitational forces, friction between pins and guide surfaces, and contact forces during actuation influence how accurately depth information is translated into physical motion. These physical effects explain observed variations in pin behavior and motivate design decisions such as conservative force margins, rigid structural support, and controlled actuation speeds to ensure stable and repeatable physical reconstruction.

E. Principles of Math

Mathematical principles provide the foundation for mapping sensed depth data to mechanical actuation. Coordinate transformations are used to align the camera's depth frame with the physical geometry of the pin board. Each depth pixel is linearly mapped from camera space into a corresponding pin height within the allowable 50-millimeter displacement range.

Kinematic relationships further connect angular servo motion to linear pin displacement through the rack-and-pinion mechanism. These relationships allow commanded servo angles to be translated into expected pin heights using deterministic geometric models. Discretization and binning are applied when converting continuous depth values into a finite 32×32 heightmap, ensuring that mathematical mappings remain consistent with the system's physical resolution and actuator constraints.

F. Functional Architecture

A depth camera operates to capture an image, transferring the depth map to a computer, which filters the frame, maps depth to the pins and down samples to a 32 x 32 heightmap. The computer converts each row of the heightmap into 32 target heights and streams a row-ordered command sequence over to a microcontroller. The microcontroller hosts three logical drivers: an Actuator Driver that generates PWM signals for the 32 actuators, a Gantry Driver that commands engage/retract and homing, and a Reset Driver for returning the pins to its reference plane. Limit switches communicate with the stepper motors for homing and for cartesian checks, ensuring that the gantry is at the intended row. Debounced Capture and Reset buttons provide user control and safety.

G. Principles of Operation

1. Capture: User presses Capture. The camera acquires one depth image and sends it to the computer.

2. Heightmap: The computer denoises the image. Then maps each depth to a stroke of the actuator and bins them.

3. Row sequencing: the computer serializes row 0 to row 31 into target heights and transmits the sequence to the microcontroller.

4. Execution: For each row, the microcontroller positions for that row, drives all 32 actuators to their setpoints, then advances the row index.

5. Reset: After all rows are written to, the actuator carriage moves to a safe position, and the reset mechanism (2 actuators on a polycarbonate sheet) returns all pins to the home plane, by request of the user.

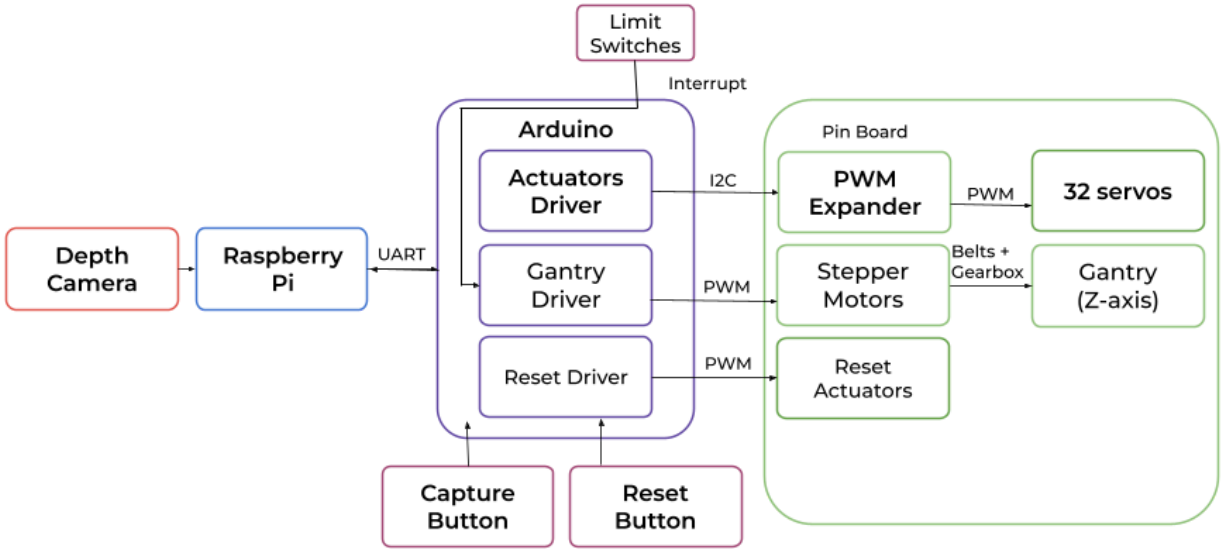

Figure 9. Block diagram

IV. Design Requirements

This section translates the use-case requirements into quantitative engineering specifications for the LivePin system. These design requirements define the performance of each subsystem to ensure that the system achieves the desired tactile 3D reconstruction with accuracy, responsiveness, and safety.

First is the actuator array design requirement. To achieve a cost-effective, compact implementation of the 32 x 32 pin array, LivePin employs **32 actuators**, each controlling a column of pins through a rack and pinion mechanism. Each pin must achieve a displacement range of **D = 50 mm,** with a pin height accuracy of **±0.25 cm.** Thus, for a pinion of radius ***r,*** the required angular resolution to be precise is

$$\Delta\Theta = \frac{\delta}{r}$$

Equation 1: Rack and Pinion Kinematic Relation

Equation 1 is the equation we will use to calculate the exact angle necessary for move the rack and pinion to the desired length.

The second design requirement is meeting the end-to-end latency target of less than 120 seconds. We can divide the latency into a simple addition equation:

$$T_{E2E} = T_{capture} + T_{processing} + T_{communication} + T_{mechanical}$$

Equation 2: Latency Equation

The end-to-end latency is divided into different latencies: the capture time of the depth camera, the processing time of the depth map into depth coordinates, the communication time between the Raspberry Pi and the STM32, and finally the mechanical action of the actuators and gantry system that pushes the pins.

Another design requirement we have is the size of the mechanical frame and gantry system. The frame dimensions are constrained by the layout of the servo motors (see Fig. 13), requiring a minimum width and height of **2 ft** each. The gantry will mount 32 servo motors in alignment to achieve uniform motion distribution across columns. As a result, it is crucial the frame is large enough to support this system and weight.

Another design requirement we have is the reset/stop button. We must ensure that the reset button's latency is less than 10 seconds. Mechanically, the actuators and gantry system need to move to a safe position before resetting. Otherwise, the carriage could interfere with the path of the resetting pins, causing damage to the system.

Each subsystem specification directly supports a use-case requirement: actuator precision and range ensure proper outputs; timing constraints ensure real-time response; and mechanical tolerances maintain perceptual recognizability. Validation will include end-to-end latency tests, pin accuracy measurements, and perceptual trials confirming that object recognition remains above 95 % under Johnson's criteria.

V. Design Trade Studies

A. Actuator Subsystem

One key design trade-off in our system is between the number of actuators and the overall project costs. Increasing the number of actuators allows more pins to be driven in parallel, reducing the total update latency for displaying a full image on the pin board. However, each actuator adds to the overall cost

of the system, not only through its own unit price but also through additional cost of shipping, power supply capacity, and wiring.

$$C_{total} = C_{fixed} + 5.95(N)$$

C_{total}: Total Costs

C_{fixed}: Costs associated with other subsystems

N: number of servo motors

Equation 3: Cost Model

Because our total budget is capped at $600, we cannot afford one actuator per pin, and we must reserve a portion of the budget for other essential components such as the mechanical frame, motors for the gantry subsystem, and other miscellaneous materials. Equation 3 illustrates the relationship between costs and the number of actuators. With component costs of $5.95 per servo motor and around $350 estimated fixed cost for other subsystems, we find that designs with 16 – 42 actuators satisfy the $600 cost constraint, image resolution, and latency requirements . Within this feasible region, we selected a 32 actuator configuration, corresponding to one actuator per column. This configuration offers a good balance between performance, cost, and complexity. It maintains manageable wiring and control overhead while providing a visually responsive update rate that meets our use-case requirements. Designs with fewer actuators would lower cost but risk longer update times and reduced visual continuity, while designs with more actuators would exceed the budget and increase the risk of wiring failures and power constraints.

Beyond costs, the number of servos also affects the resolution of the display. Spatially, the number of columns the pin board has is governed by how many columns of pins we can afford and physically pack. Less servos would mean fewer columns of pins, reducing image resolution.

In terms of power, thermal, and reliability implications, more actuators increase resolution, but raise instantaneous current draw and heat. 32 channels fit cleanly into a modest number of PWM outputs. Since wiring 32 servos directly to a STM32 is impractical, we used two PWM extenders.

Another design trade off we had to consider was the decision to use position controllable servos or continuous servos for pin actuation. Although both types are driven using PWM signals, they differ fundamentally in how the PWM command is interpreted. Position controlled servos use PWM to specify an absolute angular position, whereas continuous rotation servos interpret PWM as a velocity and direction command.
Continuous servos are more expensive at $7.50 per unit and would require time based motion control to achieve a desired pin height. This approach is sensitive to variations in laid, friction, supply voltage, and motor speed, making accurate and repeatable pin positioning difficult without additional sensing or frequent re-calibration. Any drift in speed over time would directly accumulate into position error.

In contrast, position controlled servos provide deterministic, closed loop position control, allowing each pin to be driven to a precise height using a single PWM command. This significantly simplifies the control logic, improves repeatability across updates, and reduces sensitivity to mechanical variability. Additionally, they are also cheaper. Thus, ultimately we decided to use position controlled servo motors. This decision changed the design of the carriage and pinions. Since position controlled servos are not able to rotate as far as continuous servos, the size of the pinion had to be large enough that one half rotation of the servo would result in the full stroke of the rack, thus pushing the pin to its full extrusion depth. After calculating how large the diameter of the pinion had to be to reach this max distance, it was found that the pinion had to be doubled in size, meaning the carriage geometry had to increase to accommodate this change. This relationship is modeled in equation 1.

We used 32 SG90 servo motors. These motors drew 10mA when idle, 100 - 250 mA during movement, and 300 - 650 mA when stalled or under heavy load [2]. Each servo motor required 5V.

$$A_{total\ current} = 0.65A * 32\ servos = 20.8A$$

Equation 4. Servo Motor Current Draw

Assuming heavy load, since our servo motors needed to drive a rack and pinion, each servo motor would draw upwards of 650 mA. This meant we needed a power source that could supply 5V with 20.8 Amps. We used an enclosed AC-DC switching power supply that was able to provide 5V and 30 Amps.

B. Camera and OpenCV Subsystem

When designing the Camera and OpenCV subsystem, we evaluated four depth sensing options available through lending: the OAK-D Pro, OAK-D Short Range, Intel RealSense LiDar L515, and the Intel RealSense Depth Camera D455. Because all commercially available depth cameras cost over $200, our selection was limited to these borrowed options. Among them, the Intel RealSense Depth camera was the best fit for our use case due to its balance of performance, software support and integration simplicity. The Intel RealSense LiDar L515, although capable of precise depth sensing, is no longer supported by Intel and poses long term software and driver compatibility risks. The OAK-D series, while powerful, includes features such as on board neural processing and stronger IR projectors that are unnecessary for our project.

The Intel RealSense D455 provides the optimal tradeoff between depth accuracy, operating range, latency, and ease of integration. The camera offers a usable range of 0.3 - 3 meters, making it ideal for short range interaction above the pin board. The model also interfaces easily with our Raspberry Pi via USB and are compatible with Intel's OpenCV, minimizing driver friction and setup complexity.

C. Frame and Gantry

When designing the frame and gantry system, the number of servos dictated the minimum size of the carriage and thus dictated the width of the frame. We also wanted to limit the size of the frame as our design requirement, to ensure that LivePin would be accessible and be able to be used in small scale settings. The tradeoff for this design decision was that as the number of servos goes up, increasing resolution of the pin board, the overall size of the device would also go up, creating additional costs and taking up more space.

As discussed in the use case requirements and as shown in Figure 1 – 4, we found that most basic images were distinguishable at a 32-pixel resolution, and students were even able to distinguish a face from a pixelated 32 x 32 image. Anything less than that, was only able to display very simple images, and any larger resolution would cause our frame and gantry to be too large and costly.

We capture this tradeoff with two simple relationships:

Carriage Width: $W = M_L + Ns + M_R$

N = number of column servos
s = pinion pitch = 12.5 mm
*Pitch is the distance between teeth on a gear
M_L = left side margin/tolerance = 10mm
M_s = right side margin/tolerance = 10mm

Equation 5. Carriage Width

Footprint constraint:

$$N_c \leq \frac{W_{max} - M_L - M_R}{s}$$

$$= N_c \leq \frac{W_{max} - M_L - M_R}{s} \approx \mathbf{34\ servos}$$

W_{max} = 1.5’ = 457.2 mm

Equation 6. Footprint Constraint

Our footprint constraint, which is our hard limit on how big the device is allowed to be. The max portal width we set is 1.5 feet, based on cost and modularity. To keep within our max size, our maximum number of servos is 34. To give us clearance for wires and taking into consideration the wall width of the carriage, we reduce this number by 2 servos.

Thus, selecting 32 columns satisfies the footprint requirement, preserves modularity, and meets recognizability needs, whereas pushing toward the 44-column width limit would add cost and stiffness burden with limited user-visible gain for our use case.

We selected two NEMA-7 stepper motors and TB6600 drivers for the gantry because the gantry needed deterministic, repeatable vertical positioning (row to row alignment) and sufficient torque margin to lift a relatively heavy carriage.

Mechanically, the carriage mass was ~0.5 kg, so the minimum lift force is

$$F = mg$$
$$4.8\,N = 0.5kg * 9.81$$

Equation 7. Gantry Force Calculation

With a belt / pulley lift, the torque required the drive pulley is

$$\tau = Fr$$

Equation 8. Torque

For a typical small pulley radius $r \approx 0.01\,m$, meaning the total torque needed was round 0.049 N.

$$\tau_{total} \approx 4.9N * 0.01m = 0.049\,N$$

Equation 9. Total Torque

Because we used two motors, the ideal load splits to ~ 0.025N per motor. Applying a practical safety factor (friction, misalignment, acceleration) of 2-3x yields a target capability on the order of 0.05-0.08 N*m per motor. Using steppers instead of DC motors also allows the gantry to move in precise step increments per row and hold position without drift, which is essential for repeatable row wise actuation.

Electrically, we paired the stepper motors with TB6600 drivers because they provide the voltage and current headroom needed for stepper motor torque and smooth motion. TB6600 modules accept a wide supply range (9-40 VDC) and can be configured up to 4.0 A output current, with selectable microstepping up to 1/32 [3]. The higher driver supply voltage improves current regulation at speed and helps maintain torque during motion, while microstepping reduces vibration and helps keep the motors in sync.

D. Reset Mechanism

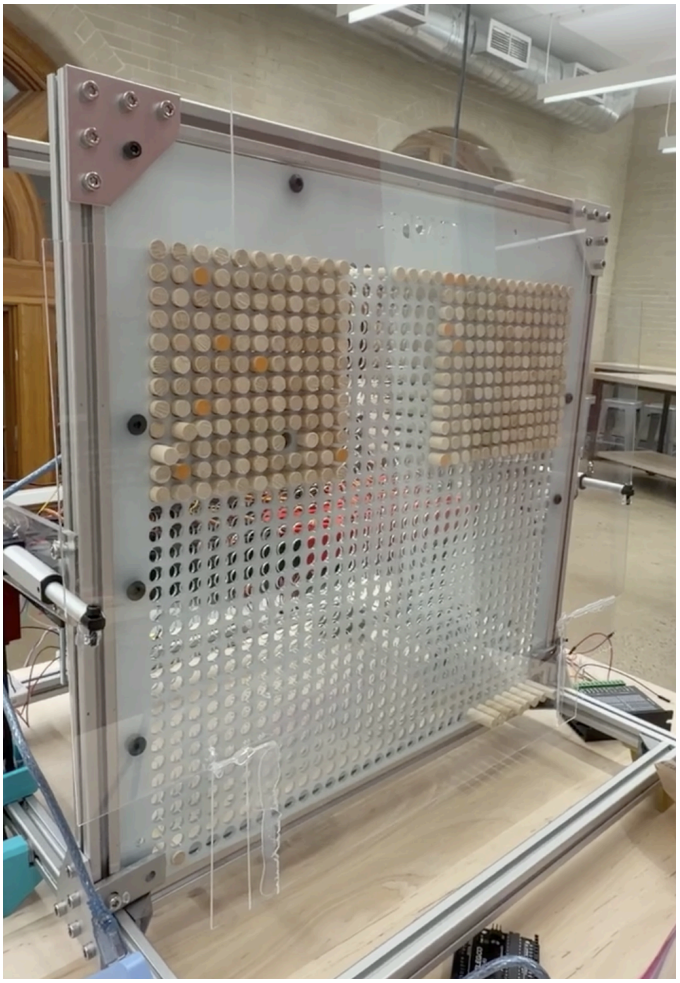

Figure 10: Fully Implemented Reset Mechanism

The reset mechanism (Figure 10) consists of two linear actuators with a 50 mm stroke that drive a shared 1/16″ thick clear acrylic sheet. This sheet sweeps across the pin board to reset all 1024 pins to a known baseline position. We selected linear actuators for this function because they provide direct, repeatable linear motion and are well suited for pushing a large number of pins simultaneously without requiring complex mechanical linkages.

According to the actuator datasheet [4], each linear actuator is rated to support up to 128 N of force, providing sufficient margin for the reset operation. Because the exact force required to simultaneously reset all 1024 pins is difficult to model analytically, due to friction, pin misalignment, and manufacturing tolerances, we minimized the load on the actuators by selecting the thinnest practical reset surface. A 1/16″ acrylic sheet was chosen to reduce mass and bending resistance while maintaining adequate stiffness to distribute force evenly across the pin array.

The linear actuators operate using a simple polarity-based control scheme, extending or retracting when 6 V is applied across their terminals with reversed polarity. To drive the two actuators efficiently and safely from a single controller, we used a DRV8835 dual H-bridge motor driver. The DRV8835 supports bidirectional control of two DC motors from a single compact IC and is compatible with the 6 V supply required by the actuators. This choice reduced component count, simplified wiring, and allowed both actuators to be driven synchronously, ensuring even and reliable reset motion across the pin board.

VI. System Implementation

A. Computer Vision

The computer vision system for LivePin was implemented using the Intel RealSense D455 depth camera and a Python script running on the Raspberry Pi, which captured a depth frame, applied preprocessing, and downsampled it to a 32×32 grid before exporting the values to a CSV file for the microcontroller. Debugging focused on ensuring the depth data was accurate, stable, and correctly aligned with the physical pin array. Early issues included noisy depth measurements, inconsistent scaling, and poor alignment between the depth frame and the pin board. To address this, we applied smoothing filters, performed calibration tests with known-depth objects, and introduced an ROI selection step that allows the user to crop the exact region of the depth map they want rendered. The ROI feature eliminated unwanted background depth data and ensured that the captured object was properly centered and scaled within the 32×32 grid. Through these improvements, the computer vision pipeline became reliable, accurate, and well-matched to the mechanical actuation system.

B. Frame and Gantry

The frame of LivePin adopts a portal-style gantry architecture inspired by Cartesian 3D printers. As shown in Fig. 5, the gantry is constructed from aluminum V-slot extrusions reinforced with corner gussets, which provide rigid mounting surfaces for the carriage plates. The gantry provides motion along a single vertical (Z) axis. Vertical translation is guided by wheels riding along the extrusion rails, while a threaded rod on the right side of the frame (Fig. 5) couples to the carriage through a nut block to constrain motion and maintain alignment.

Carriage motion is driven by two NEMA 7 stepper motors, each controlled by a TB6600 motor driver. The stepper motors actuate a pulley-and-belt system that lifts and lowers the carriage. Using two motors distributes the load across the carriage and helps maintain level motion during vertical translation.

At system startup, the stepper motors are commanded to drive the carriage downward until a limit switch is triggered, establishing a known reference position. From this home position, the motors advance the carriage upward in fixed step increments, aligning the carriage sequentially with each row of pins until the top of the pin board is reached. During initial

installation, we observed occasional belt slippage, which was resolved by adjusting belt tension to ensure reliable engagement throughout the full range of motion.

Synchronous operation of the two stepper motors is critical. If one motor advances while the other stalls or lags, the carriage tilts, leading to misalignment during row-wise pin actuation and potentially introducing damaging mechanical stress. During early bring-up, the stepper motors exhibited shuddering and inconsistent rotation, resulting in poor synchronization. Although initial adjustments to step timing improved smoothness, intermittent stuttering persisted and continued to disrupt synchronized motion.

Further investigation revealed that the issue was caused by incorrect phase wiring. Specifically, two motor phase connections were reversed, which prevented proper commutation. After swapping the appropriate phase wires (A− and B+), the stepper motors rotated smoothly and remained synchronized across the full range of motion.

The frame design itself also evolved over the course of the project. As shown in Fig. 12, the initial frame was designed around a smaller pin board and featured a uniform portal size. When the system was expanded to accommodate a 32×32 pin board with 32 servos, the carriage size increased substantially. To support the wider carriage and maintain rigidity, a third portal frame was added in the second revision of the design.

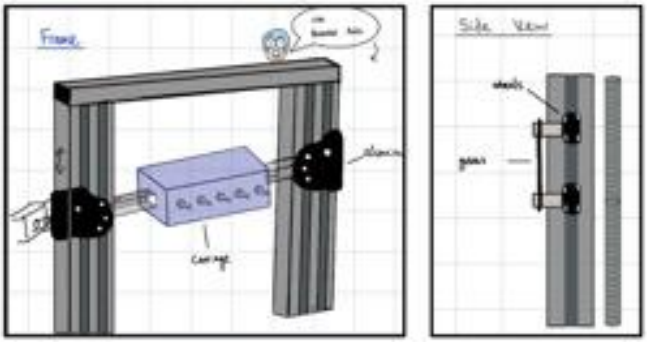

Figure 11: Concept Sketch of Gantry System. Left: front view with carriage; right : side view showing lead screw and guide wheels.

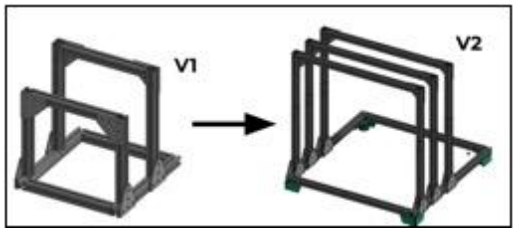

Figure 12: Onshape CAD of Frame Version 1 and Version 2

C. Carriage and Actuators

The carriage is a rigid container that mounts the row of actuators to the frame. It contains 32 rack-and-pinion servo motor modules. During a cycle the carriage receives a Z-engage command, the 32 servos move to their commanded strokes to set the row's pin heights, and then they retract. The carriage is moved to the next row by commanding the stepper motors to move a specified number of steps.

The primary design complexity of the actuator carriage was that each actuator, or rack in this case, had to perfectly align with its corresponding pin on the pin board. As shown in Fig.14a, ideally, we would have all the racks right next to each other but given that the servo is wider than the rack, we would either have to make a custom attachment on the rack (Fig.14b.) that would push multiple pins at a time. Another solution would be to place each servo in some geometric arrangement such that the racks are as close together as possible (Fig.14c.). This would require there to be multiple lengths of racks such that we can place the servos in varying positions.

Ultimately, we found a way to arrange the servos such that the racks are all in a line, in alignment with their corresponding pins. We also removed the need for multiple lengths of racks as we could just account for the furthest servo and use that length rack for every servo. This final design is shown in Fig.15 and Fig.16 with the top view being our organic placement of servos inside the carriage, and the front view showing that even with this nonlinear placement, we can align the racks and their pinions.

Each rack-and-pinion mechanism was driven by a servo motor, which rotated the pinion to extend or retract the rack and control pin height. To manage all 32 servo motors in a scalable and organized manner, we used two PCA9685 PWM expander boards. This approach significantly reduced the number of GPIO and timer peripherals required on the microcontroller while allowing individual servos to be addressed through a shared I2C interface.

During system integration, however, an error occurred in which a PWM output channel on one PCA9685 board was accidentally connected to a servo motor ground. Following this event, the PWM expander's onboard status LED became faint and eventually failed to illuminate, suggesting a potential hardware failure.

To diagnose the issue, we attempted to rotate the servos connected to the affected PCA9685 board and observed that they did not respond. The same servos functioned correctly when connected to the second, known good PWM expander, leading us to initially conclude that the first PCA9685 board had been damaged. At this stage, we did not fully disconnect all power sources (logic power, servo power, and USB) for an extended period before reattempting operation.

In response, we evaluated several fallback options: (1) splitting servo control between the STM32 and the Raspberry Pi and coordinating motion via UART, (2) reducing the total number of servos, (3) introducing an additional microcontroller to drive the remaining servos, or (4) switching to a microcontroller with a larger number of GPIO ports. Assigning servo control to the Raspberry Pi was deemed undesirable due to the added synchronization complexity and latency between signaling to the Raspberry Pi and actuation. Ultimately, we pivoted to using an Arduino Mega, which

provided sufficient GPIO resources to directly control 16 servos while maintaining centralized timing control.

Later in integration, repeated gantry motion caused mechanical wear that resulted in a snapped power wire supplying the servo motors. This event likely introduced transient voltage and current conditions on the servo power rail. Prior to this failure, we attempted to mitigate power instability by adding bulk decoupling capacitors to the servo power rails to smooth voltage fluctuations caused by simultaneous actuation of many servos. While these capacitors reduced observable voltage droop under normal operation, they could not prevent abnormal power behavior resulting from an intermittent or open power connection.

During this event, we observed abnormal behavior in which the microcontroller appeared partially powered despite not being directly connected to its primary power source, indicating back-powering through connected peripherals. This further suggests that the failure was dominated by wiring and power-domain interactions rather than insufficient decoupling or permanent component damage.

Given that the PCA9685 with I2C extension chip used separates logic power (VCC) and servo power (V+), these observations suggest that the PWM expander likely entered a temporary fault state, such as latch-up, brownout, or back-powering through signal lines, rather than suffering permanent hardware damage. After a period of disuse and subsequent clean reconnection of power and signals, both PCA9685 boards resumed normal operation and were successfully reintegrated into the system. This outcome indicates that the earlier failure was transient and caused by abnormal power conditions rather than irreversible component failure.

Ultimately in our final design, we used two PCA9685 PWM expanders controlled by an Arduino Mega.

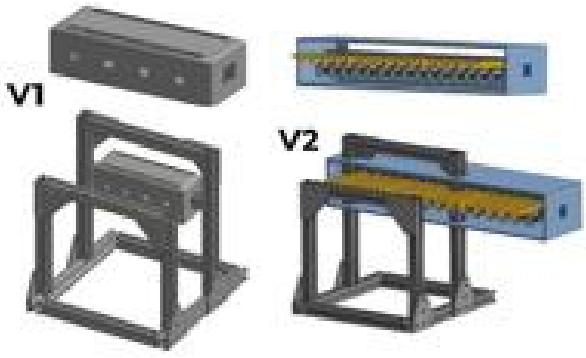

Figure 13: CAD of Carriage Version 1 and Version 2

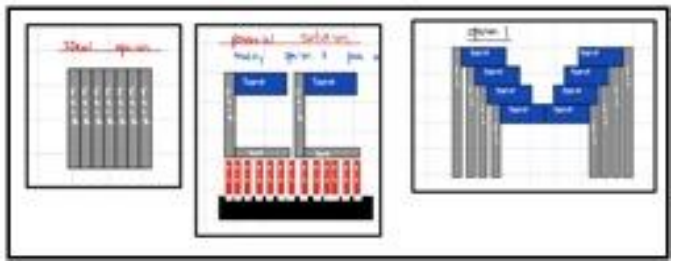

Figure 14 : a. Ideal Servo Config (left) b. Custom Head Config (middle) c. Organic Servo Config (right).

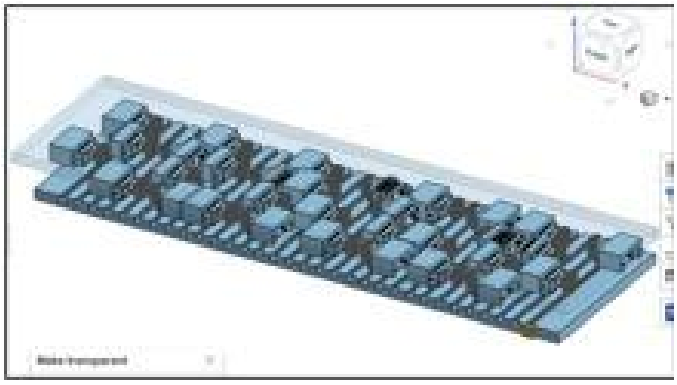

Figure 15: CAD of final servo placement inside carriage

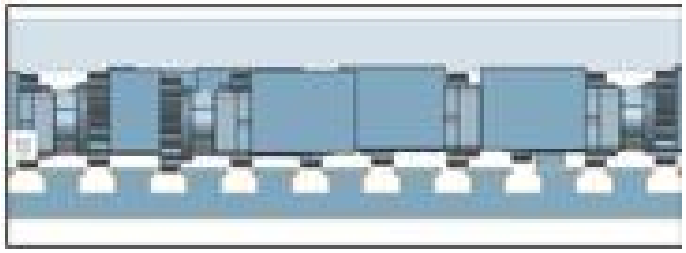

Figure 16: Front view of servo placement with rack slots

D. Pin Board

The pin board serves as the physical display surface of LivePin and is inspired by traditional pin-art toys in which pins are manually displaced to form 3D impressions. Early prototypes explored using a commercially available pin board; however, this approach became infeasible as the system evolved from an initial 16×11 display to a higher-resolution 32×32 array. This increase in resolution, combined with the physical width of the servo motors, imposed a minimum board footprint of approximately 1.5 ft × 1.5 ft, as shown in Fig. 15. At this scale, commercially available pin boards could no longer meet the system's resolution, alignment, or integration requirements.

As a result, we pivoted to designing a custom 32×32 pin board. The primary design requirements were low-friction, repeatable pin motion and precise row-and-column alignment with the gantry system. An initial concept using a fully

3D-printed pin board satisfied alignment requirements but resulted in excessive manufacturing time (on the order of one week per print) and increased material cost.

To improve manufacturability and reduce lead time, we redesigned the pin board as a stacked assembly of laser-cut acrylic plates combined with off-the-shelf ⅜" wooden dowels cut to a length of 50 mm to serve as pins. Laser cutting enabled rapid iteration and consistent hole tolerances, while wooden dowels offered a low-cost, readily available alternative to 3D-printed pins. This approach eliminated long print queues, reduced overall cost, and allowed individual pins to be easily replaced if damaged.

The resulting pin board is modular, serviceable, and maintains low-friction motion across all pins. Additionally, the design scales naturally to a 32×N configuration by extending the plate length without increasing actuator count, supporting future expansion while preserving the existing control architecture.

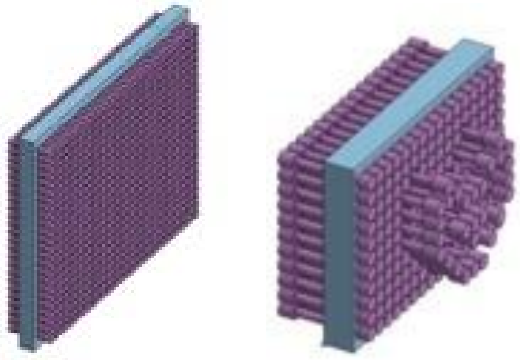

Figure 17: a. 32 x 32 pin board (left). b. 16x11 pin board(right)

E. Reset Mechanism

The reset mechanism was implemented using two linear actuators driven in parallel by a DRV8835 dual motor driver controlled by an Arduino. The purpose of this reset mechanism is to simultaneously push all 32x32 pins back to their initial position after an image has been fully actuated. To achieve synchronized motion, the two linear actuators were mounted on opposite sides of a rigid polycarbonate reset plate that spans the full width of the pin board. During a reset sequence, the Arduino sends complementary PWM and direction signals to the DRV8835, causing both actuators to extend or retract at the same time. Because the DRV8835 supports dual H-bridge outputs, both actuators receive identical voltage and timing, ensuring that the reset plate moves smoothly and remains level as it presses the pins back into their starting plane. Once the reset completes, the actuators retract the plate, clearing the path for the carriage to resume normal operation. This synchronized two-actuator system provided a reliable, mechanically simple, and cost-effective method for ensuring rapid and consistent pin resets. Although this design was mechanically straightforward, there were a few challenges during testing. First, the linear actuators moved slower than expected, which was attributed to power delivery limitations and insufficient current when both actuators are engaged at the same time. Additionally, one actuator would extend instead of retract and vice-versa because the directional wires were occasionally swapped. Thus, we had to debug and fix the retract and extend functions for both linear actuators to ensure that they were doing what they were designed to do. Despite these challenges, the reset mechanism was ultimately reliable enough to return the pin board to a neutral state.

VII. Test, Verification and Validation

A. Calibration of the Intel Realsense Depth Camera

Calibrating the Intel RealSense D455 was necessary because the accuracy of the entire LivePin system depends directly on the accuracy of the depth measurements coming from the camera. Since every pin height is mapped from a corresponding depth value, even small sensor errors such as depth bias, lens distortion, noise, or inconsistent scaling across the image would translate into physical inaccuracies on the pin board. Without calibration, a flat surface might appear slightly sloped, or an object might look too shallow or too deep, causing pins to actuate outside our ±2.5 mm accuracy requirement. By calibrating the camera and verifying its output with known reference objects, we ensured that the depth data was stable, correctly scaled, and consistent across the field of view, allowing the mechanical system to produce reliable and recognizable 3D reconstructions. To ensure that the Intel Realsense Depth Camera was properly calibrated, a Matplotlib diagram was generated and tested against the depth camera to ensure that a proper image was outputted. This graph was crucial to ensure that images were within the range of accuracy that we desired. As a result, it met the use case requirements.

B. Gantry Tests

The gantry subsystem is responsible for aligning the carriage with each row of pins and therefore plays a critical role in meeting both the pin accuracy and system latency requirements. To verify that the gantry met its design specifications and supported the intended use case, we performed a row-by-row alignment and repeatability test.

The gantry was first homed using a limit switch to establish a known reference position. From this home position, the gantry was commanded to move upward sequentially, stopping at each of the 32 row positions using a fixed number of stepper motor steps per row. This test sequence was repeated 10 times to evaluate repeatability and detect any cumulative positioning error.

At each row stop, a single actuator associated with that row was commanded to push a pin. Successful actuation of the correct pin served as a functional indicator that the carriage was aligned with the intended row.

The primary measured output of this test was row alignment correctness, defined as whether commanding the gantry to a specific row resulted in actuation of the corresponding pin. Across all test runs, the gantry correctly aligned with the intended row in 90.625% of row positioning attempts. In the remaining cases, the test failed by actuating two adjacent rows of pins simultaneously, indicating a vertical misalignment of the carriage relative to the target row.

Analysis of these failures showed that misalignment events were caused by intermittent belt slippage in the gantry drive system. Misalignment did not accumulate gradually across rows; instead, test runs typically fell into one of two patterns: either all 32 rows were aligned correctly, or a single row misalignment occurred that led to a cascading offset for subsequent rows. On average, failed runs exhibited misalignment in approximately three rows.

A 90.625% row alignment rate is near—but below—the threshold required to fully satisfy the pin height accuracy requirement. While approximately 90% of rows were actuated as intended, the remaining cases resulted in two rows being actuated simultaneously. In these instances, one row was actuated correctly, while the adjacent row was actuated incorrectly due to gantry misalignment. Because correct row alignment is a prerequisite for achieving the specified pin height accuracy of ±0.25 cm for at least 95% of pins, these misalignment events introduce systematic height errors across entire rows rather than isolated pins.

Despite these alignment errors, the gantry completed full traversals of all 32 rows without motor stalling or binding. This indicates that the gantry operates reliably within the mechanical latency budget defined in Equation 2 and does not introduce unmodeled delays into the mechanical latency term $T_{mechanical}$. Overall, this test demonstrates that the gantry provides generally repeatable vertical motion and supports consistent 3D image reconstruction in most cases. However, the observed belt-slippage-induced misalignment highlights a mechanical limitation of the current design. While performance is sufficient for demonstrating recognizable 3D impressions, improving belt tensioning or incorporating closed-loop position feedback would be necessary to fully meet the row alignment reliability required for robust, museum-grade operation.

C. Carriage Actuator Tests

The actuator subsystem translates commanded angular positions into precise pin displacements and therefore directly determines the system's height accuracy and depth reproduction capability. To evaluate whether the actuator design met the specified displacement range and accuracy requirements, we conducted a controlled actuator positioning and repeatability test.

For each test run, the gantry was first aligned to a single row to eliminate vertical misalignment as a confounding factor. Once aligned, all rack-and-pinion actuators in that row were commanded concurrently to move to five discrete angular setpoints: 0°, 45°, 90°, 135°, and 180°. Based on the rack-and-pinion kinematic relationship defined in Equation 1, these commands correspond to nominal pin displacements of 0 mm, 1.26 mm, 2.56 mm, 3.78 mm, and 5.0 mm, respectively. This sequence was repeated 10 times to assess repeatability.

The 0 mm (0°) position served as a critical test case to verify full retraction and reset capability, while the remaining four positions spanned the usable range of motion to evaluate intermediate and maximum displacement accuracy. The primary measured output of this test was displacement correctness, defined as whether each commanded angular position resulted in the expected pin displacement without stalling, overshoot, or failure to actuate.

Across all test runs, 25 of the 32 servo–rack-and-pinion subunits consistently reached all commanded heights, while 7 subunits consistently failed to actuate correctly. As a result, correct pin displacement was achieved for 78.125% of pins, which falls below the 95% pin accuracy requirement specified in the design requirements. Notably, these failures were systematic rather than random: the same actuator subunits failed repeatedly across trials, indicating a persistent mechanical limitation rather than transient control or timing errors.

While the kinematic model in Equation 1 accurately predicts the angular displacement required to achieve a given pin height, the measured results demonstrate that the current mechanical implementation does not reliably realize this model across all actuators. Because each actuator controls an entire column of pins, these inaccuracies propagate spatially across the display and have a disproportionate impact on the rendered image. Consequently, although the actuator subsystem demonstrates the capability to achieve the full displacement range in principle, its measured performance does not fully meet the quantitative accuracy requirement necessary for perceptually reliable 3D reconstruction.

Further investigation revealed that the observed systematic failures stemmed from multiple mechanical integration challenges. Due to the tight spacing of servo motors on the carriage, it was difficult to install all servos consistently.

Although the CAD design intended for servos to be secured using screws, the carriage geometry prevented reliable screwdriver access in several locations. As a result, some servos were secured using hot glue, leading to slight positional misalignments.

These misalignments were partially mitigated by adjusting screw and washer placement to raise or lower individual servos; however, this approach was not robust, as adjacent servos often required conflicting adjustments. Additionally, to conserve budget, a mix of servo motors was used, including several units obtained at no cost. These servos featured slightly different horn geometries. Although all pinions were manufactured uniformly via laser cutting, variations in horn geometry prevented consistent seating between pinions and servos, resulting in further pinion misalignment and degraded actuation reliability.

D. Pin Board Image Tests

A key use-case requirement for LivePin is that the physical pin-board output produces recognizable 3D representations for users in a museum or gallery setting. To evaluate whether the pin-board geometry and dowel-based display surface support this requirement independently of sensing and actuation errors, we conducted a human-subject recognizability test using manually actuated pin configurations.

In this test, images were recreated on the pin board through manual actuation of the wooden dowel pins, bypassing the camera, computer vision pipeline, gantry, and rack-and-pinion actuators. This approach isolated the perceptual capability of the pin-board resolution, pin spacing, and depth range without confounding factors introduced by sensor noise or mechanical inaccuracies.

A total of 47 participants were shown three manually constructed pin-board images: a heart, a house, and a portrait representation of Abraham Lincoln. Participants were asked to identify each image and then rate the Abraham Lincoln figure's similarity to a provided reference image on a scale from 1 (not similar) to 10 (very similar).

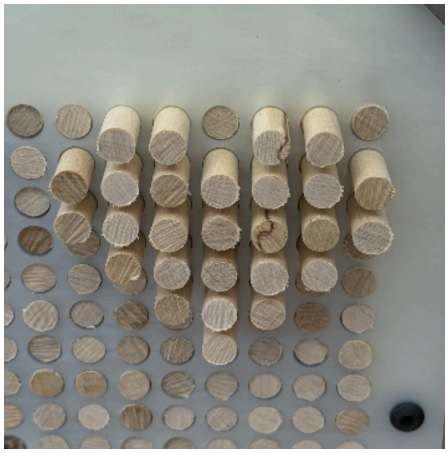

Figure 18. "Heart"

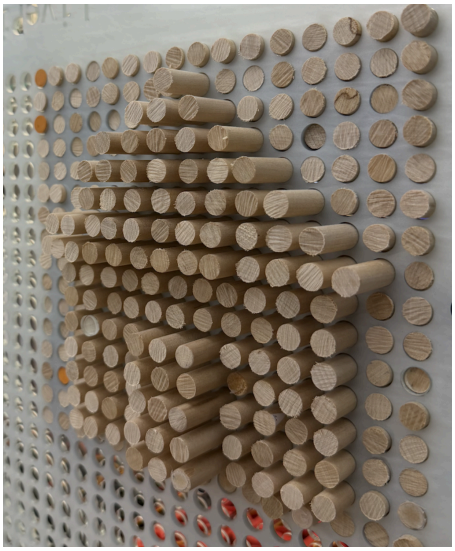

Figure 19. "House"

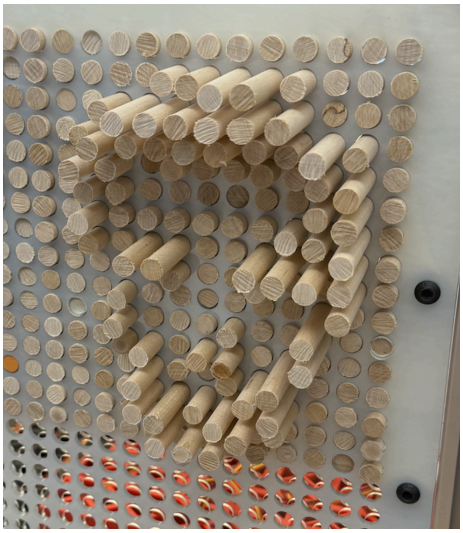

Figure 20. "Abraham Lincoln"

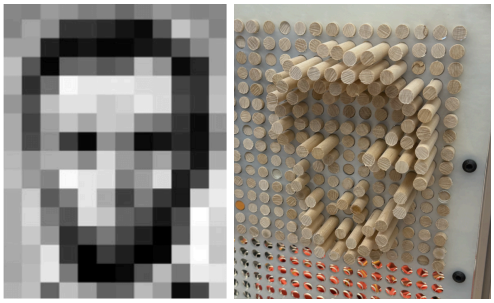

Figure 21. Reference Image vs Pin Board Recreation

The primary measured output of this test was image recognizability, defined as the percentage of participants who correctly identified the displayed image. Secondary outputs included subjective similarity ratings.

- 100% of participants correctly identified the heart image.
- 100% of participants correctly identified the house image.
- 91% of participants identified the third image as a skull, a face, or Abraham Lincoln, indicating high semantic recognition of facial structure.

Across all images, similarity ratings yielded a mean score of 6.85 and a median score of 7, indicating moderate to strong perceived similarity between the pin-board images and their references.

Qualitative feedback from participants noted that darker regions of the Abraham Lincoln image appeared inverted, suggesting that certain depth cues would be improved by pushing corresponding pins further back rather than forward. These results validate the use-case requirement that LivePin produce recognizable visual representations at a 32×32 spatial resolution using a dowel-based pin array. High recognition rates for simple geometric shapes and strong recognition of a complex facial image demonstrate that the selected resolution and pin spacing are sufficient for perceptual recognition.

The similarity scores and participant feedback also provide insight into the limits of depth representation and inversion effects, reinforcing the importance of accurate depth mapping and actuator precision in the full system. Importantly, because this test isolates the pin-board geometry, it confirms that the display surface itself does not constrain recognizability, and that remaining perceptual errors in the integrated system are likely attributable to sensing or actuation subsystems rather than the pin-board design.

By demonstrating that users can reliably recognize manually rendered images on the pin board, this test validates that LivePin's physical display medium is capable of supporting the intended museum and gallery use case. When combined with accurate depth sensing and reliable actuation, the pin board is therefore capable of producing engaging and interpretable 3D impressions suitable for public interaction.

E. Latency Tests

Latency testing was conducted to verify that LivePin met its system-level timing requirements from the depth capture to the full actuation. The test measured end-to-end latency beginning at the script initialization stage on the Raspberry Pi and ending once all 32 rows of pins were actuated. Image capture and depth processing using the Intel RealSense D455, including region-of-interest selection, filtering, downsampling, and CSV generation, consistently required approximately 10 seconds. The CSV file was then transmitted over UART to the STM microcontroller, which initiated row-by-row actuation of the pin array. Mechanical actuation of all 32 rows took approximately 90 seconds, resulting in a total system latency of 100 seconds. This performance exceeded the original requirement of completing actuation within 128 seconds and also satisfied the row-level requirement, with each row actuating in approximately 2 seconds. These measurements were obtained through repeated trials and timing instrumentation within the software pipeline, as well as visual confirmation of mechanical completion.

VIII. Project Management

A. Schedule

This project follows a structured timeline from September through December, as detailed in the full Gantt chart (Fig. 22) on page 20. The early weeks focus on proposal drafting, prototyping, and CAD design, followed by component testing and fabrication through October. Integration and subsystem testing are scheduled for November, with full system testing and final adjustments completed in early December. Major milestones include the completion of the actuation subsystem, gantry subsystem, computer vision module, reset subsystem, and final integration.

B. Team Member Responsibilities

Tedd worked on the computer vision module, the reset subsystem, the UART communication between Raspberry Pi and Arduino, and gluing endcaps on all the dowels. Crystal worked on the firmware (full program from startup to initiating reset), designed and built the circuit, cut 1024 dowels with a bandsaw, and assisted with laser cutting. Safiya worked on creating the detailed CAD design, fabrication of the rack and pinions, carriage box, gear box, and gantry subsystem.

C. Bill of Materials and Budget

See page 20.

D. TechSpark Usage

We will use TechSpark laser cutters to laser cut the acrylic that will hold the pins. We occasionally met in TechSpark to assemble components. We used a lot of tools from TechSpark like Allen wrenches, drills, and screwdrivers. We used the TechSpark woodshop's bandsaw to cut the dowels.

E. Risk Management (used to be Risk Mitigation Plans in Design Document)

A critical mechanical risk in our design is actuator reliability, since we are using cheap servos that may fail or degrade over time. Low cost servos often have limited mean time between failures due to issues such as gear wear, overheating, and inconsistent solder joints[CH1] . If even one actuator fails, it can lead to a distorted displayed image or interference with overall device functionality. If the servo fails after extending and fails to fully retract the rack and pinion, the actuator carriage will be unable to safely move to the next row as the position of the rack and pinion may interfere with the unactuated pin in the next row. To ensure long term reliability, we will have several spare servos.

Secondly, end to end latency could exceed 120 seconds. To mitigate this risk, we will parallelize actuation where the power budget allows, reduce per overhead (depth map calculations and batch I2C commands), and increase linear speed.

Third, when the device is being reset and the pins are pushed back, the pushing of the pins could damage the actuator carriage if it is in the path of the retracting pin. Furthermore the reset latency may exceed the 10 second goal. To mitigate this risk we will make the reset interrupt the highest priority, add hardware debouncing, and send the carriage to a predefined safe position.

Finally, recognizability may fall below the 90% target in user tests if the fine structure is lost when down sampling to 32 x 32. We will adapt the height map to make the height difference of pins outlining key features more dramatic, so salient depths are more noticeable. If recognition is still marginal, we can increase local sampling along feature lines without changing hardware.

IX. Ethical Issues

Although LivePin is primarily an interactive art and visualization device, there are several ethical considerations related to safety, accessibility, and responsible operation. The most direct ethical concern involves physical safety, since the system contains moving mechanical components including a gantry, actuators, and a large array of pins. In the case where a user places their hand or object inside the pin field during actuation, there is a risk of pinching, impact, or unintended force from the actuators. Individuals with limited mobility, children, or the visually impaired may be especially vulnerable to such accidents. To mitigate this, the system incorporates an emergency stop and a physical transparent frame that protects the front of LivePin. Though we didn't have time to implement these features, safeguards such as automatic motion interlocks, proximity sensors, or software-level safety thresholds could further reduce risk, all of which were ideas we had to make our system more safe.

A second ethical issue involves privacy. Although LivePin uses a depth camera, and not a traditional RGB camera, depth maps can still reveal recognizable human features. In scenarios where users interact with the system in public spaces like museums, capturing or storing depth data without consent could raise privacy concerns. As a result, we force the user to make critical decisions on the software side. For example, the GUI has buttons that force users to press buttons that clearly outline if they want to start the camera, capture the photo, and select the region they want to actuate on the pin board.

Lastly, there are accessibility and inclusivity concerns. Because the device is not tactile, people who are blind or cannot see well may not be able to interact with the machine as well as someone who has fair eyesight. However, LivePin is also very friendly for people who have low mobility. Because it does not require the user to move at all, it could be a very useful visualization tool for users who have very limited mobility.

Overall, while LivePin presents relatively low ethical risks, careful consideration of safety, privacy, reliability, and accessibility is essential for us to deploy our system responsibly.

X. Related Work

There are a few related works, specifically related to LivePin. The first, and our most direct inspiration, are the commercial toy pin boards. These toy pin boards allow users to press their hand or an object into a field of pins to produce a 3D impression. LivePin essentially modernizes this concept by adding an automated component for the pins.

An inspiration for the actuator system is from MIT's inFORM [1], which is a dynamic tactile display that utilizes a grid of actuated pins that can move up and down. While MIT's inFORM is very expensive and extremely sophisticated, LivePin is a more simplified, and cost-effective alternative to a similar idea.

Finally, the gantry and pulley system is inspired by 3D printers. 3D printers are able to move almost anywhere on the x, y, and z planes. While LivePin can only move up and down, it is nonetheless inspired by the very sophisticated systems of 3D printers.

XI. Summary

Overall, LivePin partially met its design specifications and successfully validated the core use case, while revealing clear mechanical and integration limits that constrained peak performance. At the system level, LivePin met its end-to-end

latency requirement, completing full image actuation in approximately 100 seconds, well below the 120-second target. User perceptual testing further confirmed that a 32×32 pin resolution and 50 mm displacement range are sufficient to produce recognizable 3D representations, validating the display geometry and pin-board design independently of sensing and actuation errors.

However, the system did not fully meet the quantitative pin accuracy specification of ±0.25 cm for 95% of pins. Actuator testing showed that only 78.125% of pin columns actuated reliably, with failures concentrated in a subset of servo–rack-and-pinion subunits. These errors were systematic rather than random and stemmed primarily from mechanical integration challenges, including tight servo packing, inconsistent servo horn geometries, and alignment errors introduced during assembly. Similarly, gantry testing demonstrated 90.625% row alignment accuracy, slightly below the level required to guarantee row-wise pin accuracy, with intermittent belt slippage identified as the dominant failure mode.

Despite these limitations, LivePin was able to consistently demonstrate 3D impressions, confirming that the overall system architecture is sound. The primary performance constraints arose not from sensing, computation, or control logic, but from mechanical tolerances and low-cost actuation hardware, which introduced alignment sensitivity and reduced repeatability.

A. Future work

LivePin successfully demonstrates the feasibility of a low-cost, automated pin-art display for tactile 3D reconstruction. However, several clear opportunities exist to improve robustness, accuracy, and scalability if development were to continue beyond the semester.

The most impactful improvements would be mechanical. On the gantry subsystem, replacing the belt-driven lift with a leadscrew or ball-screw mechanism would significantly reduce row misalignment caused by belt slip. Alternatively, adding closed-loop position feedback through encoders or additional limit switches would allow the system to detect and correct cumulative positioning errors in software. Improving belt tensioning hardware or using wider belts could also improve repeatability while preserving the current architecture.

For the actuator subsystem, redesigning the carriage for improved assembly access and tighter alignment tolerances would likely raise actuator reliability above the 95% accuracy threshold. Enforcing the use of identical servo models and horn geometries, along with keyed or guided pinion mounts, would reduce mechanical variability. Adding per-column calibration or compensation tables in firmware could further mitigate residual mechanical differences without requiring hardware changes.

From a control and sensing perspective, a more accurate or newer depth camera, combined with improved calibration routines, would reduce noise and bias in the depth map before actuation. Additionally, implementing adaptive depth mapping, such as exaggerating depth differences near edges or facial features, could improve perceptual quality without increasing mechanical resolution.

Finally, future iterations could explore modular expansion to larger arrays (e.g., 32×64) or faster update times through increased parallel actuation, provided that power delivery and thermal constraints are addressed. Together, these improvements would allow LivePin to fully meet its quantitative design specifications while preserving its low-cost, educational, and interactive design philosophy.

B. Lessons Learned

One of the most important lessons learned from this project is the need to balance ambition with feasibility, especially when working under strict time and budget constraints. While a 32×32 pin array enabled strong perceptual recognizability, it also significantly increased mechanical complexity, integration difficulty, and cost. In hindsight, starting with a lower resolution may have allowed more time to refine mechanical tolerances and actuator reliability.

This project also reinforced the importance of early and repeated mechanical testing. Many of the system's limitations were not due to sensing, computation, or control logic, but rather to mechanical integration challenges such as alignment tolerances, assembly accessibility, and variability in low-cost components. Future teams should prioritize mechanical prototypes early, even if they are rough, to expose these issues before committing to a final design.

Another key takeaway is that low-cost components require additional design margin. Cheap servos and 3D-printed parts can be effective, but they demand careful attention to alignment, redundancy, and calibration. Mixing hardware variants, even unintentionally, can introduce systematic failures that are difficult to correct in software.

For future teams pursuing similar interactive electromechanical systems, we strongly recommend prioritizing simplicity, modularity, and testability early in the design process.

Glossary of Acronyms

AC — Alternating Current
Electrical current that periodically reverses direction; used for mains power input to the system power supply.

ADC — Analog-to-Digital Converter

A circuit that converts analog signals into digital values (used implicitly in sensing systems).

API — Application Programming Interface
A software interface that allows programs to communicate with hardware or other software libraries.

CAD — Computer-Aided Design
Software used to design mechanical components such as the frame, carriage, pinions, and gantry.

CSV — Comma-Separated Values
A simple file format used to store the 32×32 heightmap data transmitted from the Raspberry Pi to the microcontroller.

DC — Direct Current
Electrical current that flows in a single direction; used to power motors, servos, and control electronics.

DRV8835 — Dual H-Bridge Motor Driver (Texas Instruments)
A motor driver IC used to control the bidirectional motion of the reset linear actuators.

ECE — Electrical and Computer Engineering
The academic department at Carnegie Mellon University under which this project was conducted.

GUI — Graphical User Interface
The user-facing software interface used to initiate capture, select regions of interest, and control system operation.

I2C — Inter-Integrated Circuit
A serial communication protocol used to control the PCA9685 PWM expanders.

IR — Infrared
Electromagnetic radiation used by depth cameras for stereo depth sensing.

MIT — Massachusetts Institute of Technology
Institution where the inFORM dynamic pin display system was developed.

NEMA — National Electrical Manufacturers Association
A standard defining motor frame sizes and mounting dimensions (e.g., NEMA-7 stepper motors).

PWM — Pulse Width Modulation
A control technique used to set servo motor positions and control motor drivers by varying signal duty cycle.

ROI — Region of Interest
A selected subsection of the depth image used for generating the pin heightmap.

RPi — Raspberry Pi
A single-board computer used for depth image capture, processing, and communication.

SG90 — Tower Pro SG90 Micro Servo Motor
A low-cost, position-controlled servo motor used to actuate the rack-and-pinion mechanisms.

STM32 — STMicroelectronics 32-bit Microcontroller
The embedded microcontroller responsible for actuator control, gantry motion, and system coordination.

TB6600 — Stepper Motor Driver Module
A high-current stepper motor driver used to control the gantry stepper motors.

UART — Universal Asynchronous Receiver/Transmitter
A serial communication protocol used to transmit heightmap data between the Raspberry Pi and microcontroller.

VCC — Voltage at the Common Collector
Logic supply voltage for integrated circuits.

V+ — Servo Power Supply
Dedicated power rail for servo motors, separate from logic power.

References

[1] Leithinger, D., Follmer, S., Olwal, A., Hogge, A., & Ishii, H. (2013, November 12). inFORM Daniel Leithinger*, Sean Follmer*, Alex Olwal, Akimitsu Hogge, Hiroshi Ishii. Tangible Media Group. Retrieved December 12, 2025, from https://tangible.media.mit.edu/project/inform/
[2] Tower Pro, “SG90 Micro Servo Motor Datasheet,” Tower Pro, Tech. Rep., n.d. [Online]. Available: https://www.towerpro.com.tw
[3] DFRobot, “TB6600 Stepper Motor Driver User Guide,” Version 1.2, DFRobot, Tech. Manual. [Online]. Available: https://www.dfrobot.com. Accessed: Dec. 2025.
[4] DFRobot, “6V Electric Push Rods—Product Overview,” DFRobot, Tech. Datasheet, Nov. 30, 2022. [Online]. Available: https://www.mouser.com/new/dfrobot/dfrobot-6v-electric-push-rod/. Accessed: Dec. 2025.

Description	Model #	Manufacturer	Quantity	Unplanned?	Cost	Total
Aluminum Extrusion Profile	s24011200wm0130	Uxcell	4	No	$14.99	$59.96
Angle Bracket Connection (L Type)	PL21-L	fenghe	4	No	$13.99	$13.99
T Shaped Joint Plates Bracket	n/a	EEASE	6	No	$7.00	$7.00
Corner Brace Support	1155	Adafruit	4	No	$0.95	$1.90
2’ x 2’ Acrylic Sheet	n/a	TechSpark	3	No	$7.00	$21.00
GT2 6mm wide Belt 4 meters	DPA_705	DiGiYes	10 Meters	No	$7.59	$7.59
20 Tooth Pulley	TBL6MM205	WINSINN	5	No	$6.99	$6.99
20 Tooth Smooth Idler	6MMHL20T5T	WINSINN	5	No	$8.99	$8.99
F695-2RS Bearings	F695-2RS	KABOBEARING	10	No	$6.99	$6.99
Linear Guide Rail	B0D54M2NR8	Uxcell	1	No	$12.99	$12.99
Stepper Motor	3-17HS19-2004S1	STEPPERONLINE	3	No	$32.99	$32.99
8mm Linear Rod	B0BNL4VVW4	Litoexpe	4	No	$6.99	$13.98
Linear Ball Bearings	LM8LUU	HiPicco	4	No	$9.99	$9.99
3:1 Gear Reduction with Belts	DWIBFBTHD-200MM-20T5-60T5	JNMING	2	No	$37.78	$75.56
50mm x 5mm pin	a22072500ux0432	Uxcell	10	No	$8.09	$8.09
Micro servo	169	Adafruit	32	No	$5.95	$190.40
PWM extenders	PCA9685PW/Q900,118	NXP USA Inc.	2	No	$2.88	$5.76
Raspberry Pi	B07TD42S27	Raspberry Pi	1	No	n/a	n/a
STMicroelectronics NUCLEO-F401RE	NUCLEO-F401RE	STMicroelectronics	1	No	n/a	n/a
Push Pull Actuator	1738-FIT0805-ND	DFRobot	2	No	$27.00	$54.00
5V 30Amp Power Supply	5V 30A Power Supply	Velain	1	Yes	n/a	n/a
Stepper Motor Drivers	TB6600	Handson Technology	2	Yes	$18.99	$37.98
Reset Actuator Driver	DRV8835	Texas Instruments	2	Yes	n/a	n/a
Wooden Dowels	n/a	Walmart and Michaels	3816 inches	No	varying	~$110

Grand Total $690.07

* We used everything we ordered.

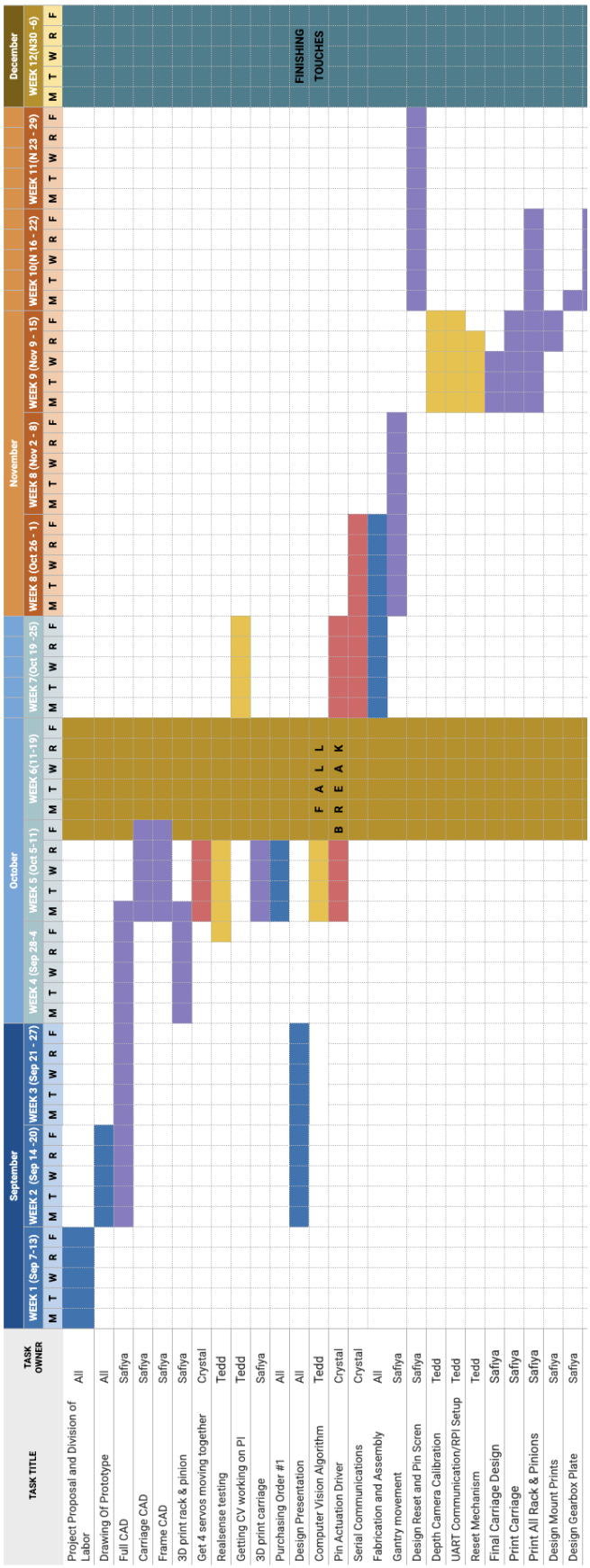

Figure 22. Gantt Chart