LivePin

Tedd Jung, Crystal Huang, Safiya Bainazar

Department of Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*—** LivePin is a cost-effective, environmentally friendly 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*—** actuators, depth camera, Intel RealSense D455, linear actuator, OpenCV, pin based display, rack and pinion, Raspberry Pi, servo actuation, STM NUCLEO-F401RE

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. There is a clear need for an affordable medium that bridges the gap between digital imagery and real-world perception.

LivePin addresses this need by transforming 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 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.

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. 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.

To ensure near real-time interaction, 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/OR PRINCIPLE OF OPERATION

Overview (Fig.1 and Fig.4)

Our device is a depth-driven pin display that renders a 3D relief using actuators. The overall physical system is shown in Fig. 1 and Fig. 3. 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. 3) , 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.

Functional Architecture (Fig.2 and Fig.3)

A depth camera operates to capture an image, transferring the depth map to a Raspberry Pi, which filters the frame, maps depth to the pins and down samples to a 32 x 32 heightmap. The Raspberry Pi converts each row of the heightmap into 32 target heights and streams a row-ordered command sequence over UART to an STM32. The STM32 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. Actuator driver modules and reset devices are addressed on I2C. Debounced Capture and Reset buttons provide user control and safety.

Principle of operation

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

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

3. Row sequencing: the Raspberry Pi serializes row 0 to row 31 into target heights and transmits the sequence to the STM32.

4. Execution: For each row, the STM32 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.

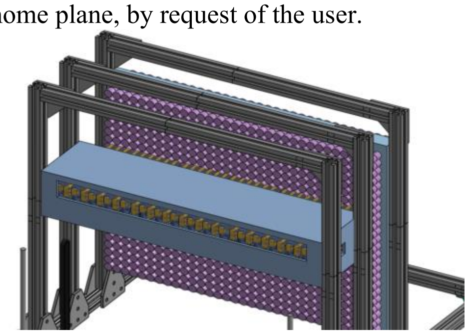

Figure 1: Carriage w/ Actuators and Custom Pin Board Mounted on Frame

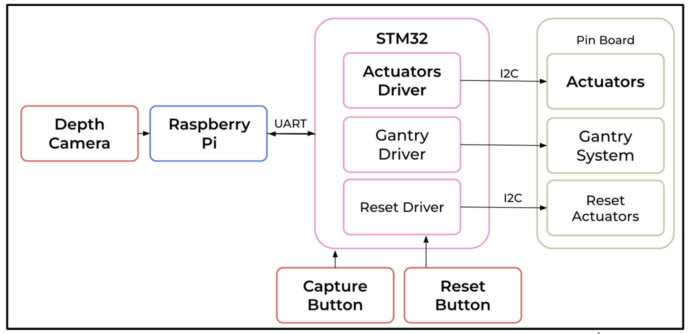

Figure 2: System Specification Approach – Block Diagram[1]

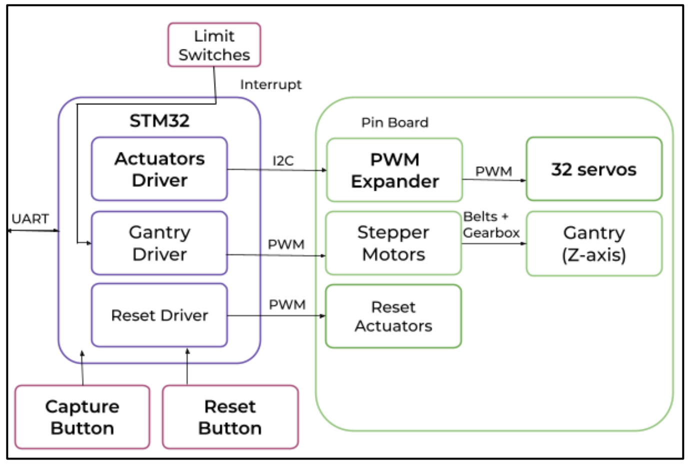

Figure 3: Subsystem Block Diagram

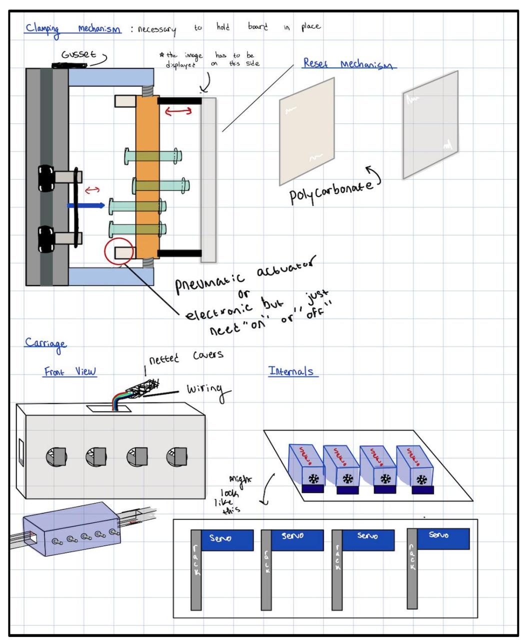

Figure 4: Concept sketch of full LivePin Design Side View[2]

IV. Design Requirements

This section translates the use-case requirements into quantitative engineering specifications for the LivePin system.

[1] Enlarged block diagram found on page 9.

[2] Enlarged concept sketch found on page 10.

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 pinon of radius ***r***, the required angular resolution to be precise is

$$\Delta\theta = \frac{\delta_x}{r}$$

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:

$$T_{E2E} = T_{capture} + T_{proc} + T_{comm} + T_{mech}$$

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. 8), 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.

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

Equation 3: Cost Model

C_{total}: Total Costs
C_{fixed}: Costs associated with other subsystems
N: number of servo motors

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 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 servo 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 less column 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 include two PWM extenders.

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.

Through user testing 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: $W_{car} = m_L + N_c s + m_R$
N_c = number of column servos
s = pin pitch = 12.5 mm
M_L = left side margin/tolerance = 10mm
M_R= right side margin/tolerance = 10mm

Footprint constraint:

$$N_c \leq \left\lfloor \frac{W_{\max} - m_L - m_R}{s} \right\rfloor$$

$= \left\lfloor \frac{457.2 - 10 - 10}{12.5} \right\rfloor \approx$ **34 servos**

W_{max} = 1.5' = 457.2 mm

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.

VI. System Implementation

A. Frame and Gantry

The frame of LivePin adopts a portal-style gantry inspired by cartesian 3D printers. As shown in Fig. 5, the portal consists of aluminum V-slot extrusions with corner gussets that provide mounting faces for the carriage plates. The gantry provides Z-axis motion only. The threaded-rod (right side of Fig.5) couples to the carriage through a nut block. Guide wheels ride on the vertical extrusion to guide the carriage up and down. In Figure 6, the original version of the frame was made we a smaller pin board in mind, and it had a uniform size between the two portals. In version 2 the carriage got a lot bigger because we changed our design to have 32 servos. Thus, we needed to add an additional portal frame to support the carriage.

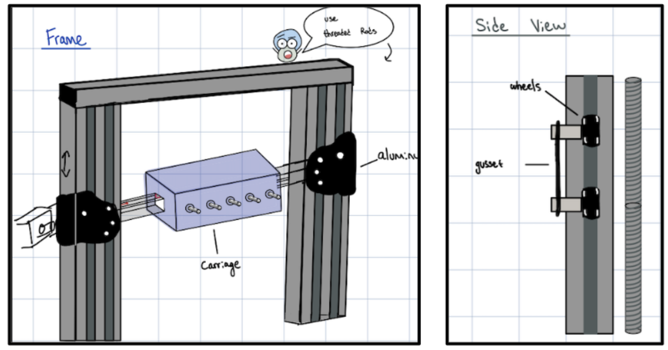

Figure 5: Concept Sketch of Gantry System. Left: front view with carriage; right : side view showing lead screw and guide wheels.

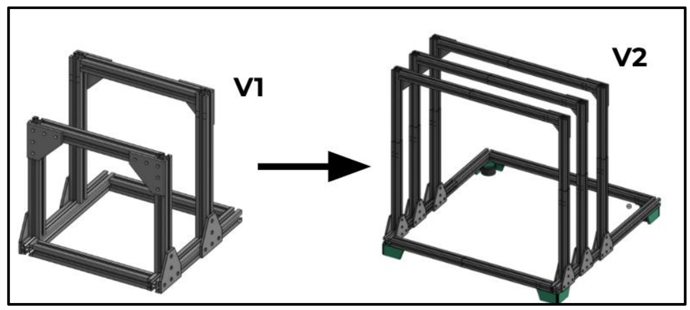

Figure 6: Onshape CAD of Frame Version 1 and Version 2

B. 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. A simple indexing mechanism advances to the next row.

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.8a, 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.8b.) 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.8c.). 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 lengths rack for every servo. This final design is shown in Fig.9 and Fig.10 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.

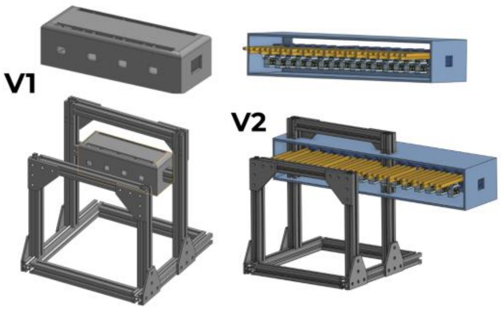

Figure 7: CAD of Carriage Version 1 and Version 2

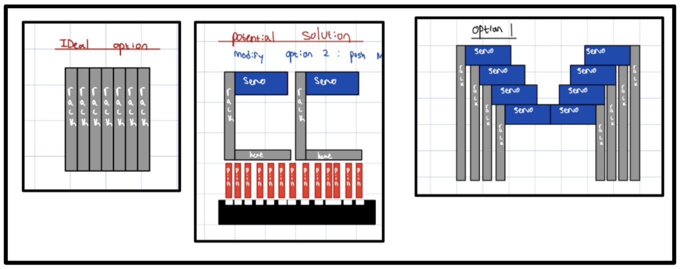

Figure 8: a. Ideal Servo Config (left) b. Custom Head Config (middle) c. Organic Servo Config (right).[3]

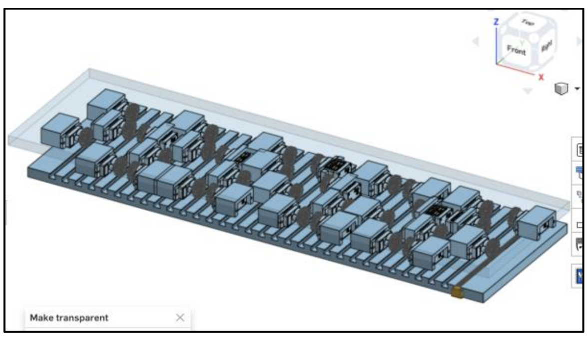

Figure 9: CAD of final servo placement inside carriage

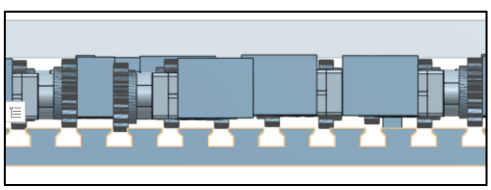

Figure 10: Front view of servo placement with rack slots

C. Pin Board

The pin board is the physical display surface of LivePin. It is inspired by the pin board toys that you manually push. We pivoted from buying a pin board toy to building a custom design after moving the display from 16x11 to 32 x 32 for a higher resolution image; this resolution change is shown in Fig.11. The pin board design must provide low-friction, repeatable motion for each pin, and maintain precise row/column alignment to the gantry.

Based on these constraints, we pivoted away from buying a pin board toy to creating a custom 32 x 32 pin board. A fully 3D- printed board met alignment needs but imposed a week-long print time and higher material cost. To improve manufacturability and lead time, we redesigned the board stack to use laser-cut acrylic plates with off-the-shelf wooden dowels as pins. Laser cutting provides us with a fast turnaround and consistent hole tolerances. Dowels eliminate long print queues, reduce cost, and are easily replaced. The resulting board is modular and serviceable, maintains low friction, and scales to 32 x N by extending plate length without increasing actuator count.

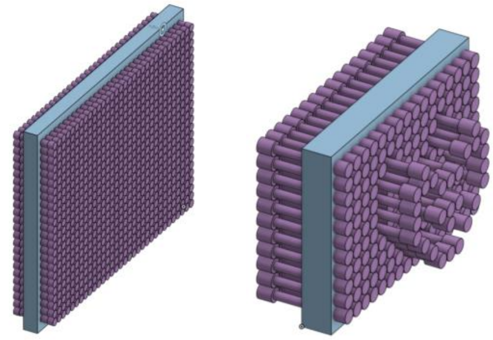

Figure 11: a. 32 x 32 pin board (left). b. 16x11 pin board(right)

VII. Test, Verification and Validation

The testing and verification plan for LivePin is designed to confirm that the implemented system meets both the design specifications and the use-case requirements. Each test targets a specific subsystem or performance metric and evaluates whether the real-world implementation aligns with theoretical design expectations and trade-offs. The test outputs directly correspond to measurable performance indicators such as accuracy, latency, and user perception, ensuring full validation of the system's intended functionality in interactive environments.

A. Actuator Accuracy Test

To verify that the actuator array achieves the required accuracy and displacement range, a hardcoded depth map will be used as the test input. The pins will be actuated according to the depth map data, and individual pin heights will be measured using a ruler. The system will be considered successful if over 95% of the pins actuate to ±0.25 cm of the expected height.

This test validates the reliability and accuracy of the actuator control algorithms and mechanical tolerances. By comparing measured pin positions against the depth map, discrepancies can be traced to either servo control errors or mechanical errors. These results confirm that the actuator subsystem can produce accurate and recognizable impressions, fulfilling both design and use-case accuracy requirements.

B. Tests for Use-Case Specification B: Latency

To evaluate the system's responsiveness, end-to-end latency will be measured from the moment an object is captured by the camera to the completion of all pin movements. This test

[3] Enlarged images can be found on page 11.

will use a variety of random 3D objects to ensure that it can produce images through a broad range of objects. The total latency must remain below **120 seconds** to satisfy the design specification for real-time operation in interactive exhibits.

If the measured latency exceeds this threshold, optimization efforts will focus on parallelizing actuation and improving the gantry speed. Additionally, the reset button latency will be evaluated by timing how long the system takes to return all pins to the neutral position after pressing the emergency stop button. This reset process must be completed within 10 seconds to ensure operational safety and reliability. Together, these tests verify that LivePin meets both technical and experiential performance expectations for its use case.

C. Tests for Use-Case Specification C: Image Resolution

To confirm that LivePin fulfills its use-case requirement of generating recognizable 3D impressions, a user study will be conducted. Fifty participants will be asked to identify the object or shape rendered on the pin board. The system will be considered successful if over 90% of participants correctly recognize the displayed forms.

This test directly links the quantitative performance metrics like pin accuracy, displacement precision, and latency to tangible outcomes. If recognition rates fall below target, the depth map generation and binning algorithms will be refined to better map depth gradients to the 32 x 32 pin array. Through this evaluation, LivePin's ability to deliver meaningful and engaging 3D representations is verified, demonstrating that the design effectively satisfies its intended use-case requirements in educational and museum settings.

VIII. Project Management

A. Schedule

This project follows a structured timeline from September through December, as detailed in the full Gantt chart (Fig. 16) on page 12. 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 is working on the computer vision module and reset subsystem. Crystal is working on the actuator subsystem as well as the communication protocols between the Raspberry Pi, STM32, and other subsystems. Safiya is working on creating the detailed CAD design, fabrication of the rack and pinions, carriage box, gear box, and gantry subsystem. We will all be working on the physical assembly of the project.

C. Bill of Materials and Budget

Please refer to Figure 12 for our Bill of Materials located on page 8.

D. TechSpark Use Plan

We will use TechSpark laser cutters to laser cut the acrylic that will hold the pins.

E. Risk Mitigation Plans

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. 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 the 120 second. 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. Related Work

There are a few related works that are similar yet very different from our project. The first is MIT's inFORM[2], which is a Dynamic Shape Display that can render 3D content physically, allowing users to interact digitally in a tangible way. This is a very cool idea that served as an inspiration for us to create a system that could actuate pins similarly to inFORM. However, inFORM is very difficult and physically impossible to replicate with our current budget, as it uses a 30 x 30 pin array that are individually actuated. This means that there are essentially 900 actuators, 1 for each pin, allowing the system to run with extremely low latency in real time.

Another system we took inspiration from is the flip-dot display. While our pin art board does not look like the flip-dot display, we still were inspired by how it is able to display information on the board efficiently and in a cost-effective way. Because of our novel idea with the pin art board, we realized that flip-dot displays are the closest alternative to a relatively cheap display board that outputs information.

X. Summary

LivePin is a low-cost, modular pin matrix display that converts a captured depth image into a 3D relief on a 32 x 32 array of pins. A depth camera provides a short range map, a Raspberry Pi processes it, an STM32 drives column actuators through a rack and pinion carriage to raise pins to command heights. For stakeholders, museum exhibit designers and visitors, the impact is an interactive medium that makes depth visible and touchable, revealing the full pipeline from depth sensing to actuation. Compared to holographic or volumetric displays, LivePin emphasizes affordability, serviceability, and interactive value.

The main implementation risks are hitting the quantitative requirements under budget. Technically, we must keep end to end latency less than 120 s, which depends on actuation time and per row travel; achieve ±2.5 mm pin accuracy for over 95% of pins; and maintain a 50 mm pin displacement. On the sensing side, we need depth accuracy at a short range. Reliability is another challenge, low cost servos can drift or fail, so we will unit test the servos before incorporating them and swap out malfuncting servos.

Glossary of Acronyms

CAD – Computer-Aided Design
STM32 – STM NUCLEO-F401RE

References

[1] IEEE, *IEEE Author Center: Author tools*, Accessed on Jan 17, 2022, [Online]. Available: https://newauthors.ieeeauthorcenter.ieee.org/author-tools/

[2] D. Leithinger, S. Follmer, A. Olwal, and H. Ishii, "Physical telepresence: shape capture and display for embodied, computer-mediated remote collaboration," *Proc. 27th Annu. ACM Symp. User Interface Softw. Technol. (UIST)*, pp. 461–470, Oct. 2014, doi: 10.1145/2642918.2647377.

[3] C. Lott, "A close look at how flip-dot displays really work," *Hackaday*, Mar. 3, 2023. [Online]. Available: https://hackaday.com/2023/03/03/a-close-look-at-how-flip-dot-displays-really-work/

[4] G. L. DesAutels, "A modern review of the johnson image resolution criterion," Optik, vol. 249, p. 168246, doi: 10.1016/j.ijleo.2021.168246.

Description	Model #	Manufacturer	Quantity	Cost	Total
Aluminum Extrusion Profile	s24011200wm0130	Uxcell	4	$14.99	$59.96
Angle Bracket Connection (L Type)	PL21-L	fenghe	1 pack (contains 4 pieces)	$13.99	$13.99
T Shaped Joint Plates Bracket	n/a	EEASE	1 pack (contains 6 pieces)	$7.00	$7.00
Corner Brace Support	1155	Adafruit	4	$0.95	$1.90
2' x 2' Acrylic Sheet	n/a	TechSpark	3	$7.00	$21.00
GT2 6mm wide Belt 4 meters	DPA_705	DiGiYes	10 Meters	$7.59	$7.59
20 Tooth Pulley	TBL6MM205	WINSINN	1 pack (contains 5 pieces)	$6.99	$6.99
20 Tooth Smooth Idler	6MMHL20T5T	WINSINN	1 pack (contains 5 pieces)	$8.99	$8.99
F695-2RS Bearings	F695-2RS	KABOBEARING	1 pack (contains 10 pieces)	$6.99	$6.99
Linear Guide Rail	B0D54M2NR8	Uxcell	1	$12.99	$12.99
Stepper Motor	3-17HS19-2004S1	STEPPERONLINE	1 pack (contains 3 pieces)	$32.99	$32.99
8mm Linear Rod	B0BNL4VVW4	Litoexpe	2 packs (1 pack contains 2 pieces)	$6.99	$13.98
Linear Ball Bearings	LM8LUU	HiPicco	1 pack (contains 4 pieces)	$9.99	$9.99
3:1 Gear Reduction with Belts	DWIBFBTHD-200MM-20T5-60T5	JNMING	2	$37.78	$75.56
50mm x 5mm pin	a22072500ux0432	Uxcell	1 pack (contains 10 pieces)	$8.09	$8.09
Micro servo	169	Adafruit	32	$5.95	$190.40
PWM extenders	PCA9685PW/Q900,118	NXP USA Inc.	2	$2.88	$5.76
24v 8 amp AC Adapter Power Supply	n/a	IDeATe	1	n/a	n/a
Raspberry Pi	B07TD42S27	Raspberry Pi	1	n/a	n/a
STMicroelectronics NUCLEO-F401RE	NUCLEO-F401RE	STMicroelectronics	1	n/a	n/a
Push Pull Actuator	1738-FIT0805-ND	DFRobot	2	$27.00	$54.00

Grand Total **$542.07**

Figure 12: Bill of Materials

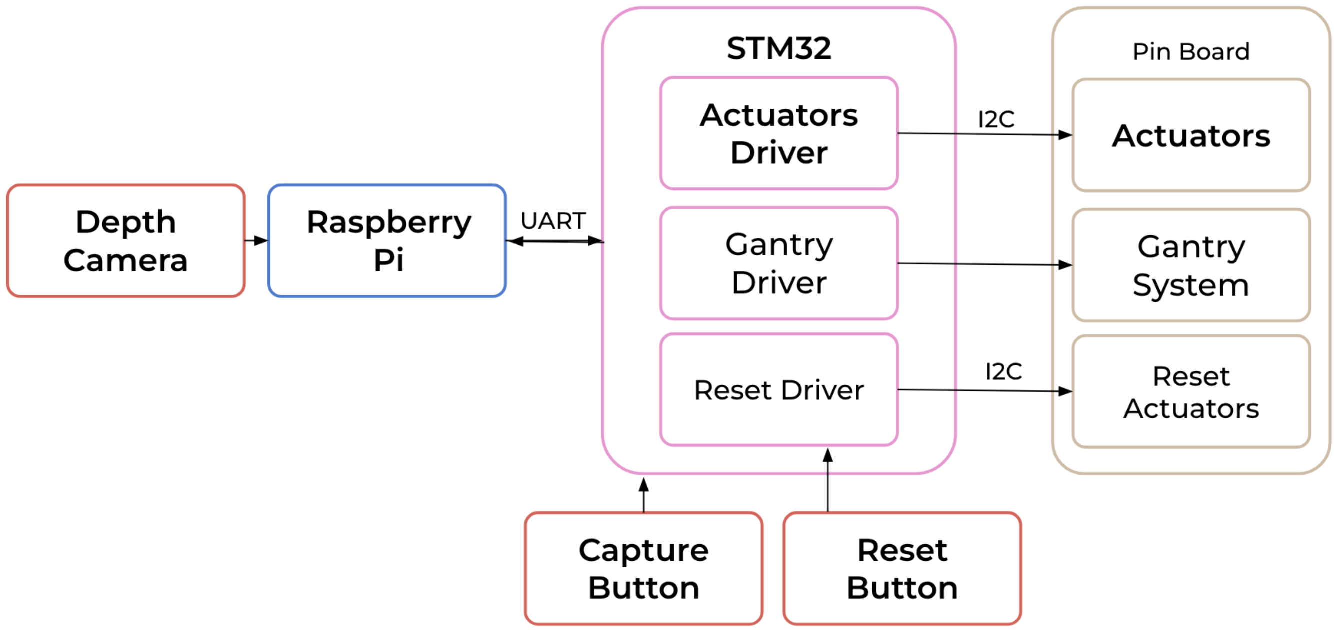

Figure 13: System Design Block Diagram

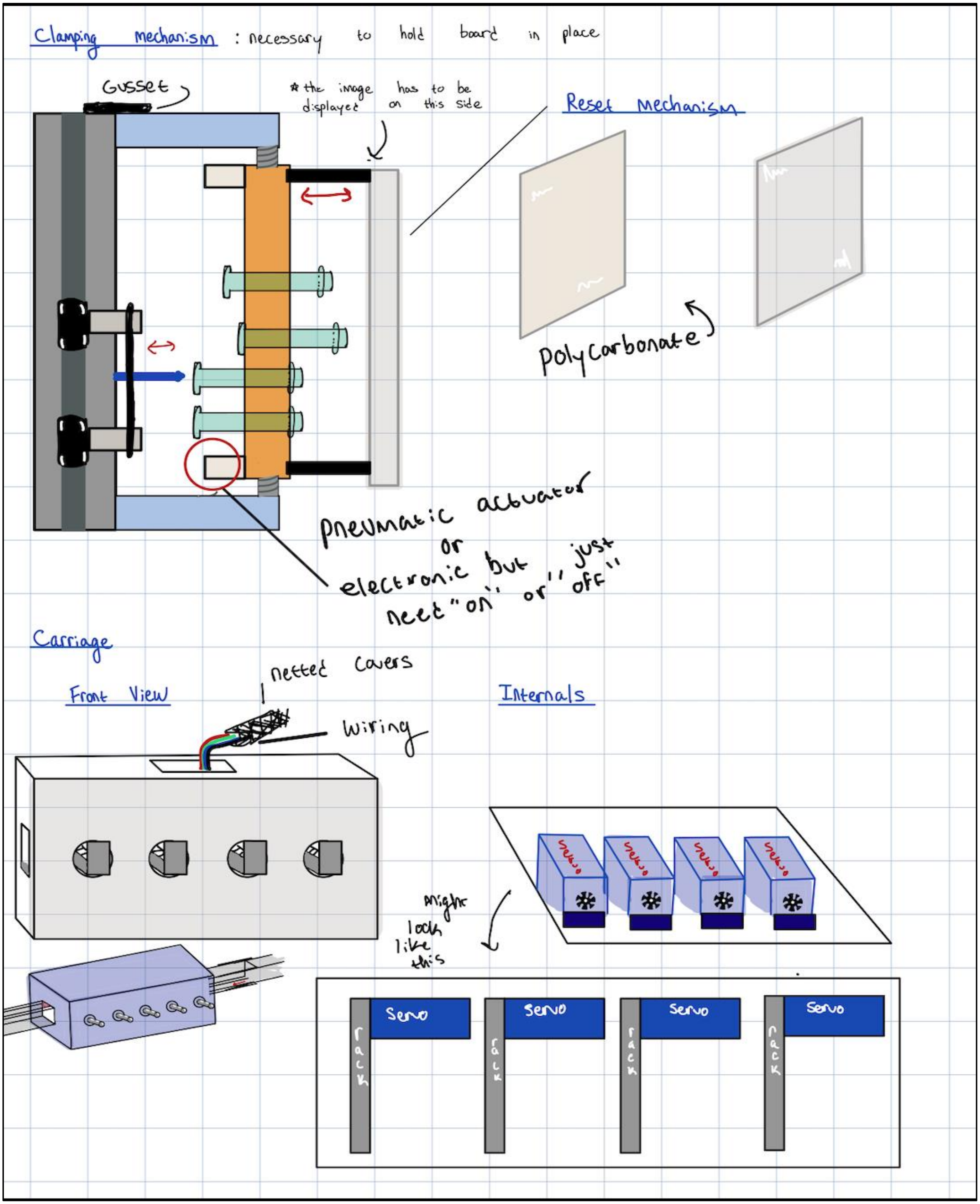

Figure 14: Concept Sketch

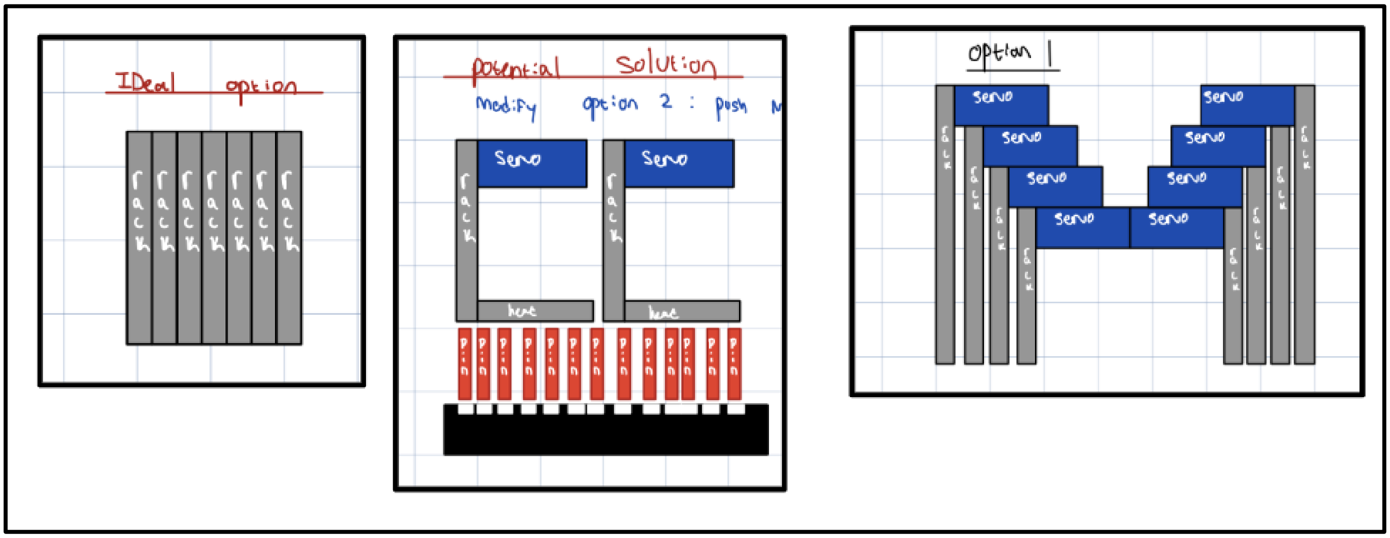
IDeal option
rack
potential Solution
Modify option 2 : push
Servo
rack
here
pin
Option 1
Servo
rack

Figure 15:

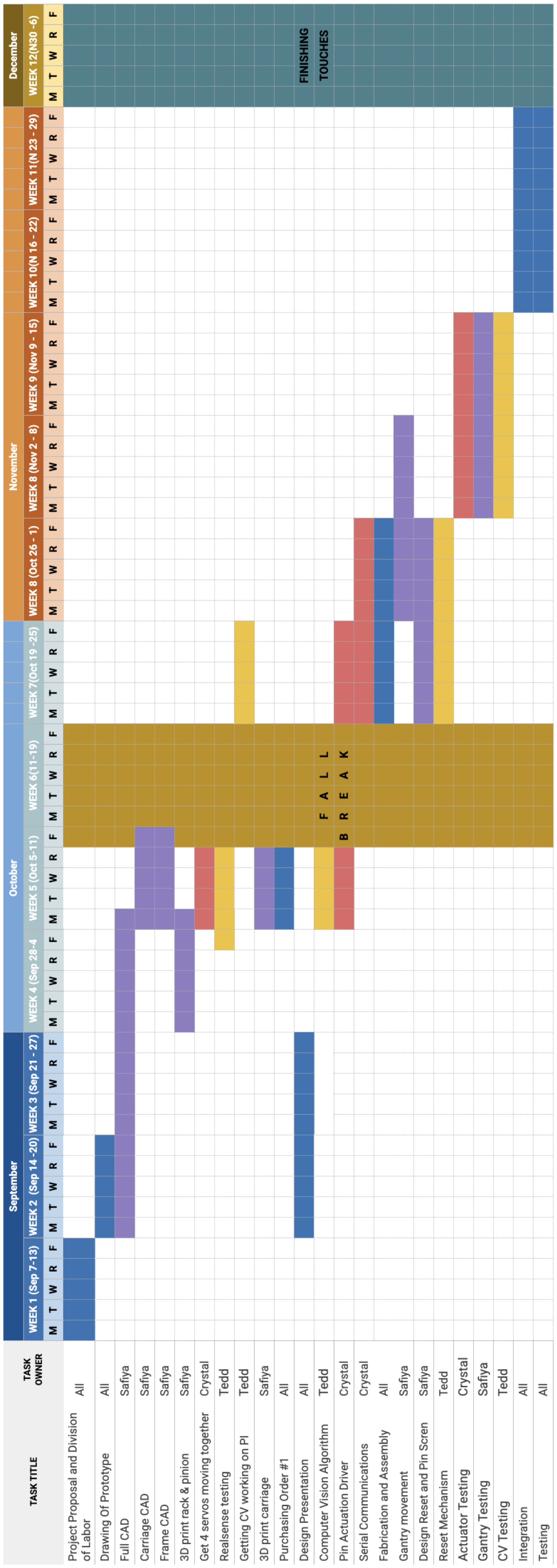

Figure 16: Gantt Chart