SPOTLIGHT TRACKING WITH AUTOMATED RECOGNITION (S.T.A.R.)

Review Committee
Dr. Jaesung Lee
Dr. Wei Sun
Dr. Stephen Eikenberry
Dr. Andrew Klein

Faculty Advisors
Dr. Chung Yong Chan
Dr. Aravinda Kar

Meet the Team
Group 7

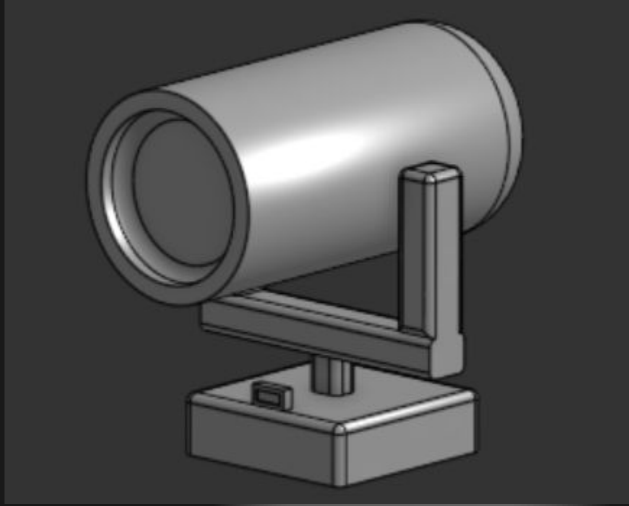
Lucio Ruben Villena
Computer Engineer

Travis Nguyen
Optical Engineer

Gage Anderson
Optical Engineer

Jerison Lau
Electrical Engineer

Project Overview



- **Automated Spotlight System**: STAR (Spotlight Tracking using Automated Recognition) is a motorized theatrical spotlight automatically tracks a subject.
- **Camera and illumination systems**: The STAR uses two separate optical systems: a camera system for subject tracking, and an LED illumination system.
- **Motors and control system:** The spotlight uses servo motors for gimbal movement, and an app to select subjects.

Motivation

- **Problem:** There are currently no tracking spotlights on the market that use a camera with human tracking to automatically follow a subject. Other technologies (RFID tracking chips) are very expensive and inaccurate

- **Proposed Solution:** is a low-cost, human detection based imaging spotlight system designed for performer tracking in live theatrical environments. The system targets a specific use case, allowing reliable performance without the cost or complexity of commercial follow-spot solutions.

Our Goals & Objectives

The primary goal of the S.T.A.R. project is to develop an automated spotlight system that can image, identify, and track a performer in real time.

This includes designing a multi-lens imaging system using commercial components, implementing human detection algorithms, and driving a motorized gimbal for accurate pan and tilt control, with system performance validated through stage-based latency testing.

Advanced and stretch goals focus on expanded tracking capability, improved motion control, and integration with existing lighting systems.

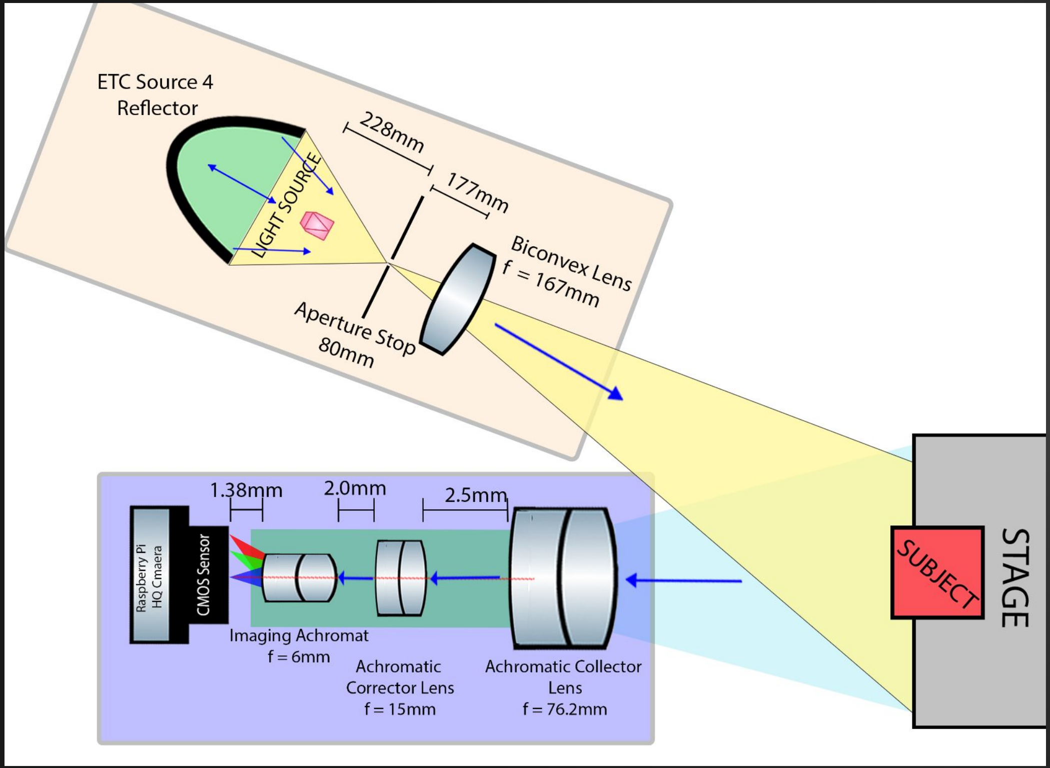
Specifications

Parameter	Description	Target Value & Unit(s)
Tracking Latency	Time delay between performer movement and spotlight response	≤ 300 ms (basic), ≤ 150 ms (advanced)
Tracking Accuracy	Angular error between spotlight center and the selected performer's center of mass	≤ 5° RMS at 10m
Detection Accuracy	The percentage at which a human subject is correctly identified upon first entering the field of view.	80%
Field of View (FOV)	The angular viewable area that the sensor can image	> 45°
Depth of Focus	The acceptable focus depth that a performer can move on a stage	> 1m
Tracking Range	Distance at which subject can be reliably detected and tracked	7.5m (± 1m)
Frames Per Second (fps)	How many frames are captured by the camera per second	50 fps @ 1920x1080p
Lighting Conditions	Minimum illumination where tracking is possible	≥ 1000 Lumens

Specifications cont.

Parameter	Description	Target Value & Unit(s)
Spotlight Beam Size	Effective beam diameter at 10m	0.5m - 1m adjustable (>5° half cone)
Pan/Tilt Range	Gimbal rotation coverage	Pan: 180° max, Tilt: 120° min
Pan/Tilt Speed	Spotlight movement speed	≥ 180°/s pan, ≥ 120°/s tilt
Operating Power Consumption	Wattage consumed by system during operation (NOT including illumination source)	~ 611Watts
System Weight	Total weight of system (NOT including illumination source)	≤ 20 lbs
System Cost	Total cost of prototype build (not including redundancies)	≤ $800 prototype

Optical Design



- Illumination system composed of a high power, broadband 3000k LED source
- Reflector absorbs heat (IR radiation) and reflects visible light
- Imaging plane at the gate allows for projection of images/patterns
- BK7 lens after the gate to focus output light

- Imaging system is stationary, placed at the base of the system
- Uses raspberry pi HQ camera
- Triple lens system, to ensure accurate, clear imaging

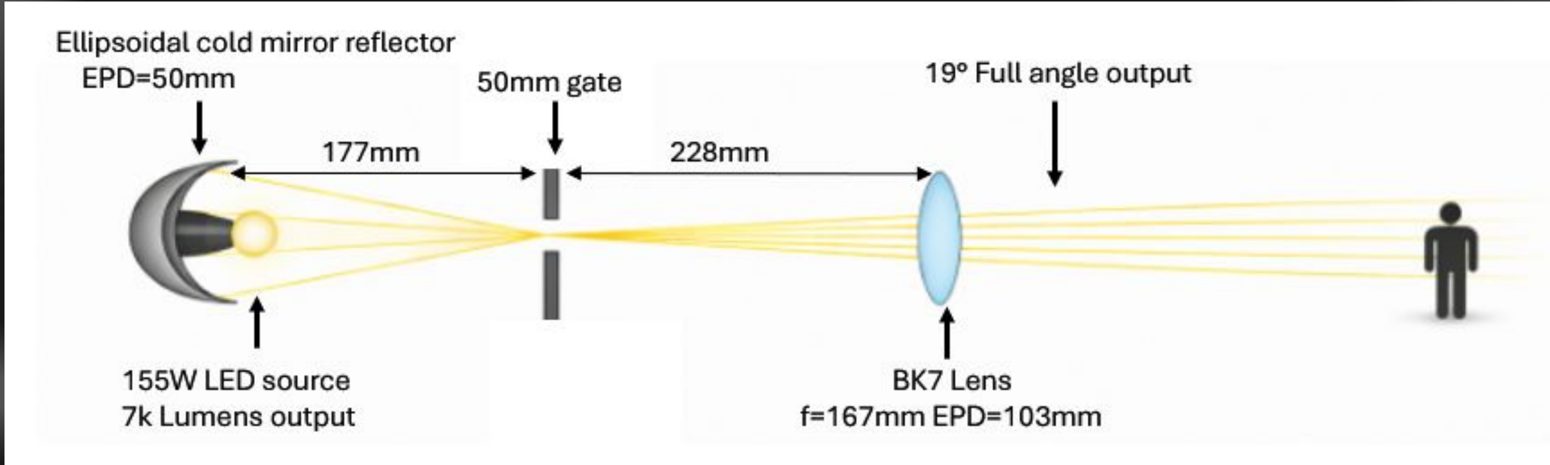
Illumination System

$$tan(\theta) = \frac{1.25}{7.5} = 9.5$$

- Calculations for the illumination system were made around the requirement that the full angle output be 19°. This was determined using basic geometric laws given two requirements: spot diameter=2.5m, and the distance from the source to the subject = 7.5m.

- 50mm gate size is a standard in stage lighting, so it was chosen for compatibility with existing accessories.

- The cold mirror reflector that was chosen fit the geometry of our system, so EPD=50mm and an output angle of 36°



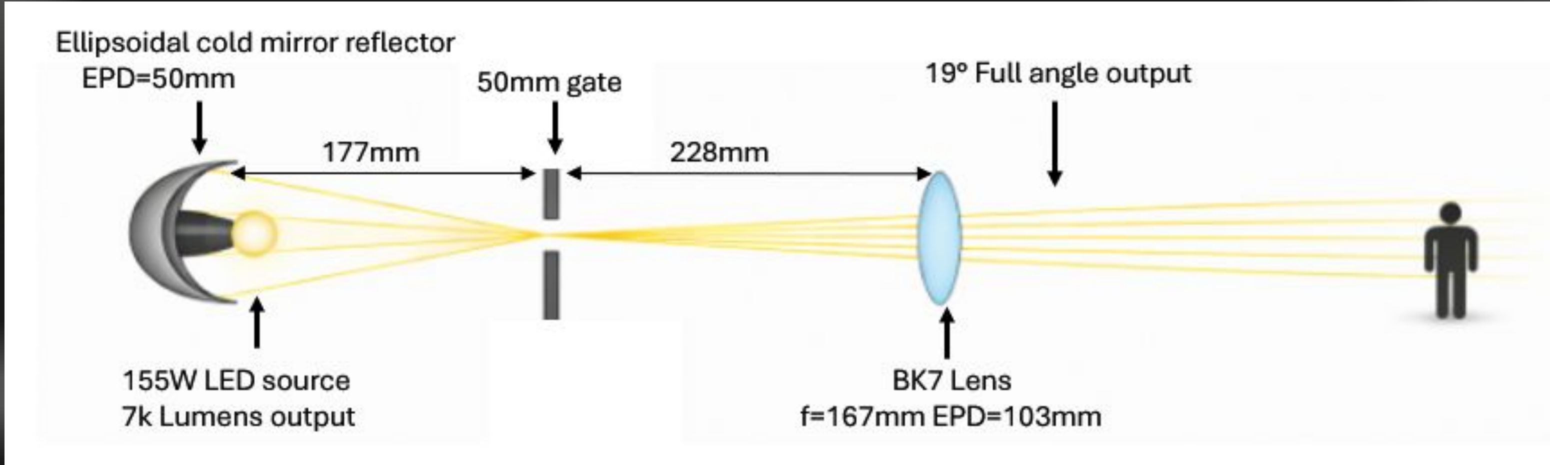
Illumination System

1. $$NA = n(sin(\frac{\Theta}{2})) \rightarrow NA = (sin(18)) = .31$$

2. $$NA = \frac{1}{2(f/\#)} \rightarrow f/\# = \frac{1}{2(.31)} = 1.618$$

3. $$f = \frac{r}{tan(\Theta/2)} \rightarrow f = \frac{25}{tan(8.5)} = 167mm$$

4. $$EPD = \frac{f}{f/\#} \rightarrow \frac{167}{1.618} = 103mm$$



Lens selection

Parameter	Calculated	Selected COTS Lens	Unit
Focal Length	167	171	mm
EPD	103	105	mm
Material	BK7	BK7	–

- We were able to obtain a lens that has almost identical parameters to the calculated specifications. The slight differences do not noticeably impact image quality or spot size.

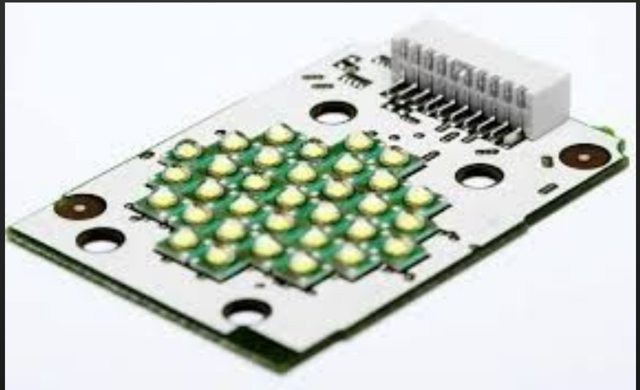
Source selection

- Three different styles of light sources were considered for selection:
  - 1. Tungsten-halogen lamp (Incandescent)
  - 2. Integrated RGBW Engine
  - 3. LED COB (Chip-on-board) Module

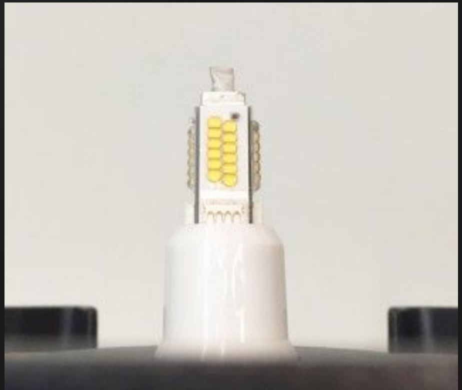
- Tungsten-halogen lamp provides the most aesthetically pleasing light due to its broadband emission, but produce extreme heat (500C)
- LED arrays are very compact and provide multiple color options, but require significantly more optics for collection, color mixing, and projection.
- Remote-phosphor LED light engine provides high output, and impressive temperature regulation due to the core being a copper heat pipe.

1

2.



3.

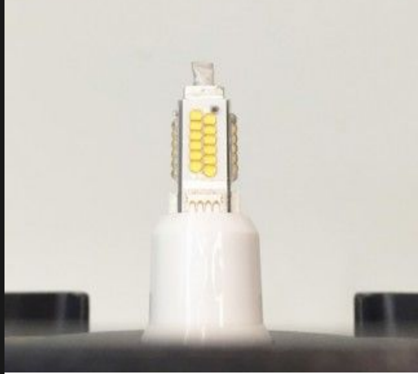


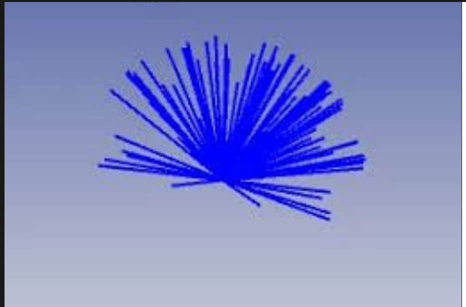
Source selection

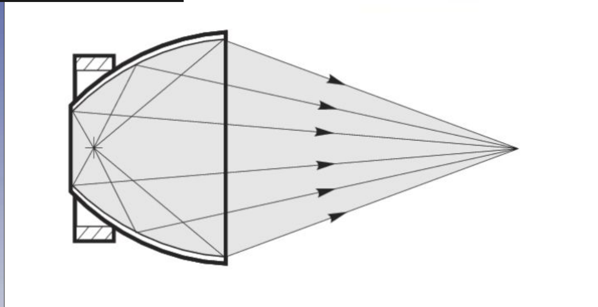
Criteria	HPL 757 Halogen Lamp	High-Power LED COB Module	Integrated RGBW Engine	Unit	Best Option
Luminous Output (lm)	≈ 5000	4500–6000	3500–5000	Lumens (lm)	LED COB Module
Efficiency (lm/W)	≈ 18	≈ 90	≈ 70	(lm/W)	LED COB Module
CRI	> 85	80–90	> 90	Percent	Integrated RGBW Engine
Thermal Load	High	Moderate	Low	-	Integrated RGBW Engine
Lifetime	300–500	25,000+	20,000+	Hours	LED COB Module
Cost	~$20	~$80	~$150	USD	HPL 757 Halogen Lamp
Integration Ease	Direct fit to housing	Requires adapter + driver	Custom driver needed	-	HPL 757 Halogen Lamp

Reflecting vs. Refracting source optics

- Using a lens to try to focus light from a lambertian emitter only collects about 5-10% of the output light
- An ellipsoidal reflector collects roughly 50-70% of output light
- Reflector chosen is the ETC Source 4WRD Ellipsoidal Reflector
- Reflector coating absorbs heat, reflects visible light. (cheaper than lens coating)

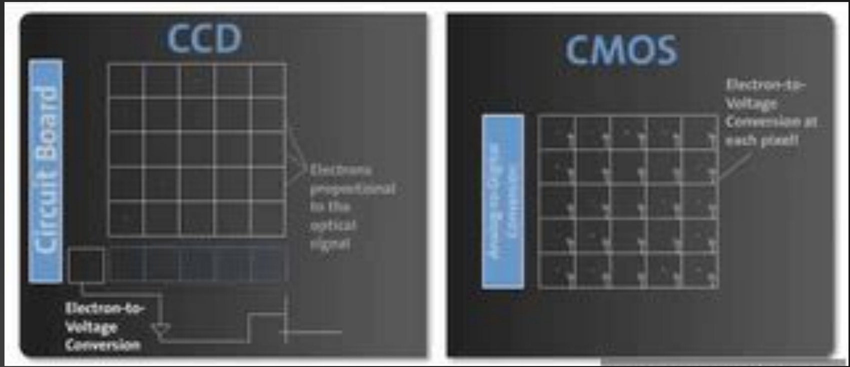






Sensor Technologies - CCDs vs CMOS

Specification	CCD	CMOS
Frame rate	Slower	Faster
Temperature control	Requires cooling	Does not require cooling
Power consumption	High power usage	Low power usage
Output noise level	Low	High
Cost	More expensive	Less expensive



The chosen technology between the sensors was the CMOS sensor, primarily due to its lower power usage and lower cost.

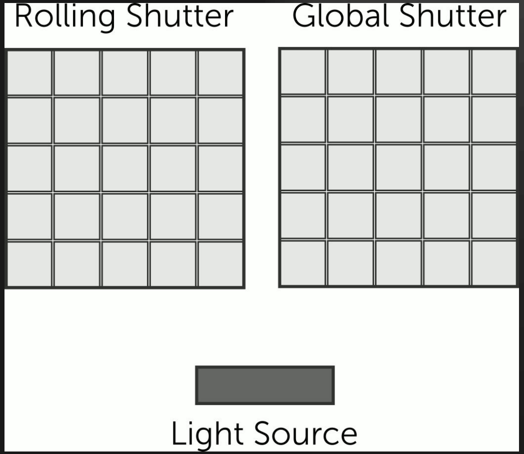
Not requiring cooling is a benefit as other systems within S.T.A.R. may already emit heat, causing an increase in dark current in CCD sensors

Sensor Technologies - Global vs Rolling Shutter

There are two main types of sensor technology that was considered:

1. Global Shutters - capture an entire scene and its pixels in an instant simultaneously
2. Rolling Shutters - reads and exposes images and pixels line-by-line sequentially

Specification	Global Shutter	Rolling Shutter
Sensor Size	Larger	More Compact
Cost	High	Significantly lower
Power Consumption	Higher	Lower
Availability	Limited	Widely available

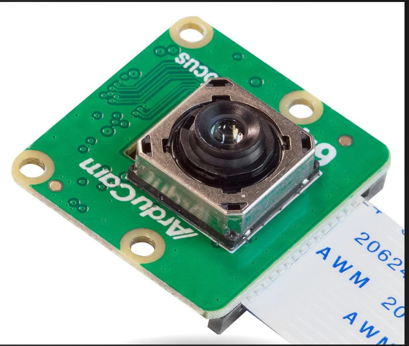


After consideration we ultimately chose for a rolling shutter sensor due to a cost vs performance analysis.

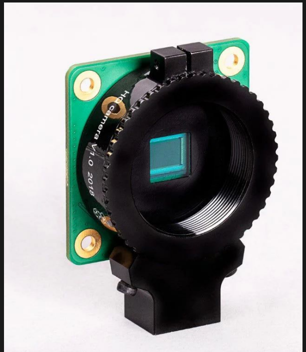
The smaller size and lower power consumption helps support our system despite a small cost in the imaging capabilities.

Sensor Part Comparison

Specification	ArduCam Pi Hawk-eye	Raspberry Pi HQ Camera
Net Price	$65	$50
Shutter Type	Rolling Shutter	Rolling Shutter
Sensor	Sony IMX686	Sony IMX477
Sensor Resolution	9152 x 6944 pixels	4056 x 3040 pixels
Still Resolution	64 megapixels	12.3 megapixels
Horizontal Image Area	Not Specified	6.287mm
Vertical Image Area	Not Specified	4.712mm
Pixel Size	0.8 μm × 0.8 μm	1.55 μm × 1.55 μm
Horizontal Field of View	74 degrees	Lens Dependent
Vertical Field of View	55 degrees	Lens Dependent
Lens Mount	Non-interchangable	C/CS or M12 Mount



ArduCam Pi Hawk-eye



Raspberry Pi HQ Camera

Ultimately chose the Raspberry Pi HQ Camera with the Sony IMX477 CMOS Sensor as the more cost-effective option with enough pixels to image our target.

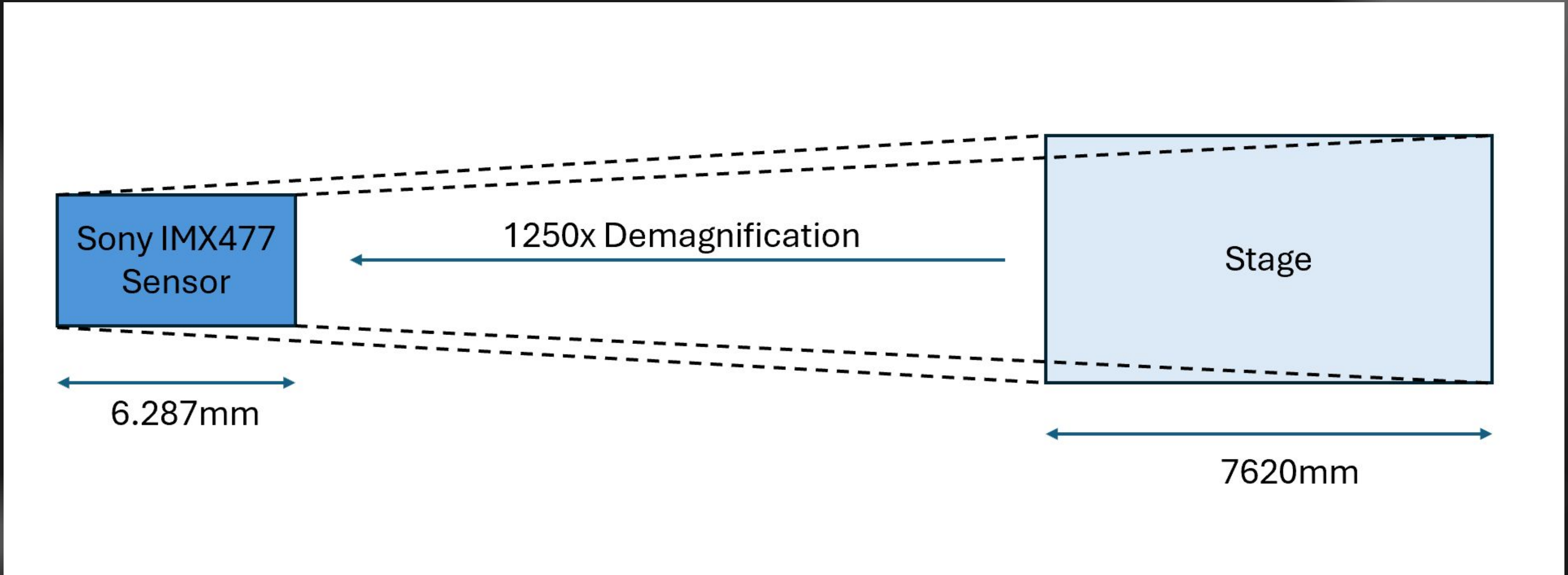
The transmittance is in the visible wavelength ranges, with a filter in the IR region

Lens System Parameters Calculations

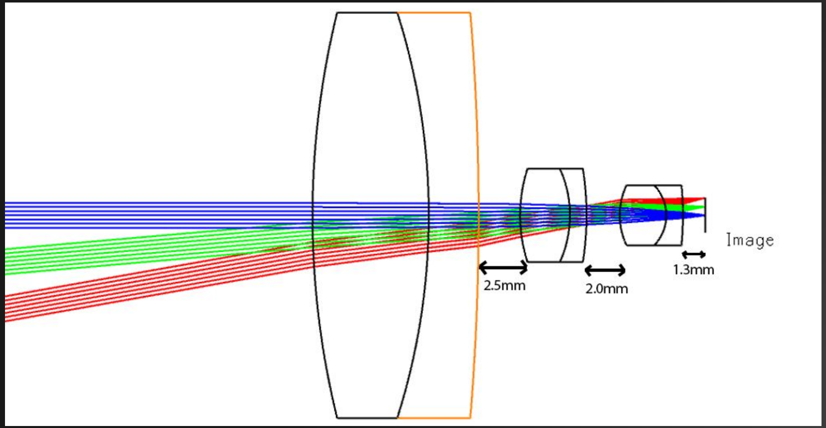
We want to image a stage with length of ~7.62m at a distance of 7.62m

$$M = \frac{h_i}{h_o} = \frac{L_{Sensor}}{L_{Stage}} = \frac{6.287mm}{7620mm} = 0.0008 \approx \frac{1}{1250}$$

The target magnification is a 1250x demagnification to image the entire FOV of the stage, alongside any performers on it to the incident sensor

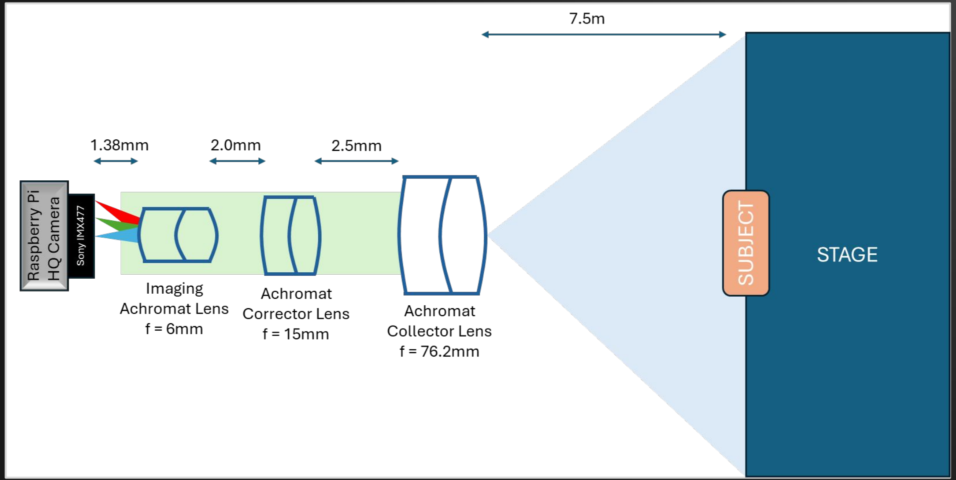


Lens Selection



Achromatic Doublets
Used for every lens to mitigate the amount of chromatic aberration appearing on the sensor.

Collector Lens
Large diameter to minimize vignetting and maximize light collection and capture the full field needed to image the stage.



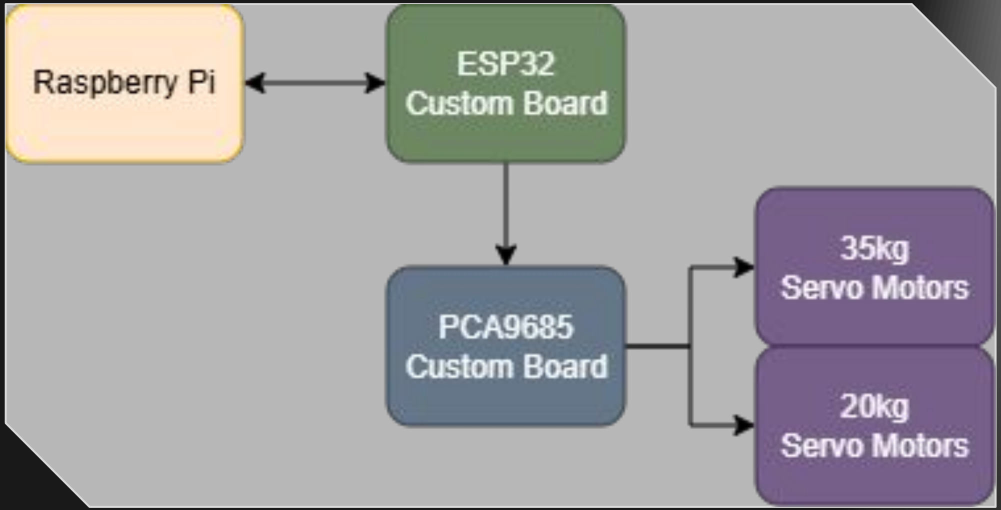
Corrector Lens
Ensures accurate imaging and accounts for other aberrations (spherical, coma, field curvature).

Imaging Lens
Provides proper effective focal length to achieve the desired magnification

Coatings
Specialized COTS coatings primarily used to transmit solely in the VIS wavelengths optimizing it for the 400-700nm range.

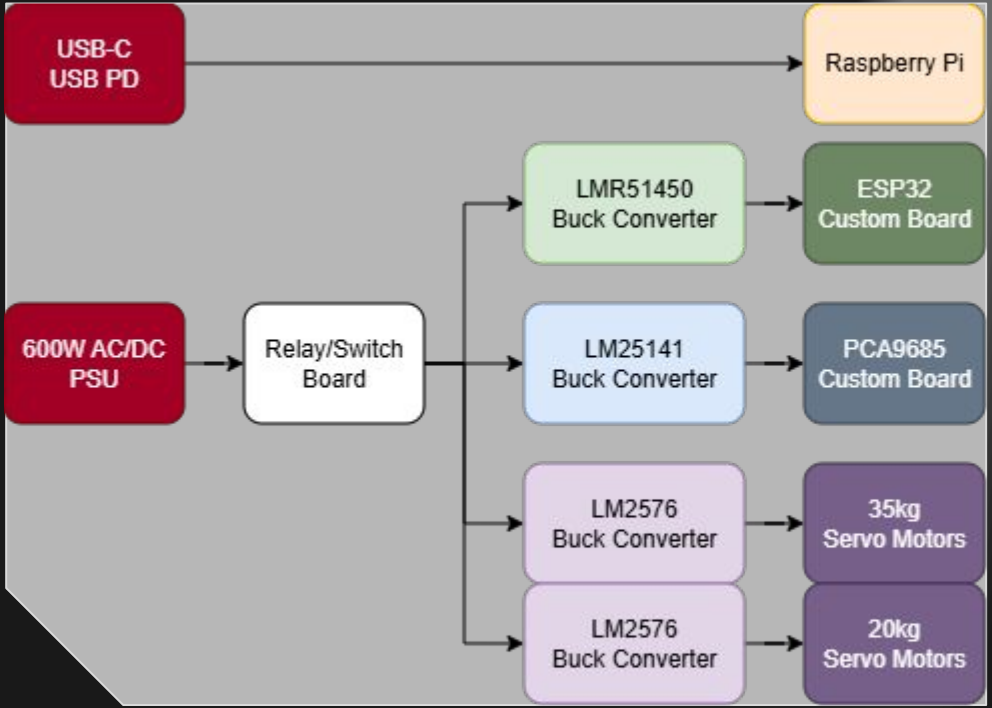
Hardware Block Diagram - High Level

- Raspberry Pi transmits/receives data to/from ESP32 through UART connection
- ESP32 transmits data to PCA9685 through I2C
- PCA9685 controls servo motors using PWM signals



Hardware Block Diagram - Power Delivery

- Raspberry Pi receives power independently through a USB-C power supply
- 600W PSU connects to custom switch board
- Various Texas Instrument LMxx ICs are used to step down the PSU output to rated values for our ESP32, PCA9685, and motors



Hardware Selection - SoM vs SBC

- SoM was chosen over SBC for the ability to design a custom carrier board
- Custom CM5 carrier was initially planned, high-speed PCB requirements made it impractical within our time and cost constraints.

Attribute	System-on-Module (SoM)	Single Board Computer (SBC)
Integration Level	Core computing components only (CPU, RAM, storage, GPU). Requires carrier board.	Fully integrated with I/O ports, power management, and interfaces on a single PCB.
Design Flexibility	Highly customizable—carrier board can be tailored for specific project needs.	Limited customization—fixed I/O layout and hardware configuration.
Scalability	Easier to upgrade by replacing the compute module without redesigning the carrier.	Typically requires replacing the entire board to upgrade hardware.
Manufacturing Cost	Higher initial design cost but scalable for production runs.	Lower cost for prototypes and small projects.
Thermal Management	Can be optimized with custom heat dissipation solutions.	Fixed thermal design based on vendor layout.

Hardware Selection - MCU vs FPGA

- FPGAs require specialized knowledge to develop increasing project risk and design effort.
- An MCU was selected as the most cost-effective, power-efficient, and development-friendly solution for the system.

Attribute	MCU (e.g., ESP32-S3)	FPGA
Programming Complexity	Easy—uses C/C++ or MicroPython.	High—requires HDL (VHDL/Verilog) and synthesis tools.
Development Speed	Rapid—large ecosystem and open-source libraries.	Slower—requires detailed logic design and timing closure.
Power Consumption	Low; ideal for embedded and battery-powered systems.	High; often requires cooling and significant power overhead.
Cost	Very low (<$10 for ESP32-S3).	High ($50–$200+ depending on FPGA).
Reason Chosen for S.T.A.R.	The ESP32-S3 provides real-time motor control, Wi-Fi, and I^2C communication in a compact, low-cost package.	FPGA unnecessary for S.T.A.R.'s moderate control and vision needs.

Hardware Selection - ESP32 vs STM32

- ESP32 was chosen for S.T.A.R. due to ease of development, integrated wireless connectivity, and cost effective pricing for units

Attribute	ESP32-S3	STM32 (e.g., STM32F4/F7)
Wireless Connectivity	Integrated Wi-Fi + Bluetooth 5.0	Requires external modules
I/O Interfaces	I^2C, SPI, UART, PWM, USB, ADC, DAC	Extensive peripheral support
Ease of Development	Excellent—supported by Arduino, ESP-IDF, and MicroPython	Good—requires STM32CubeIDE
Chosen for S.T.A.R.	Offers wireless connectivity, powerful dual-core processing, and efficient servo control via I^2C (PCA9685).	STM32 better for high-precision control but costlier and lacks wireless integration.

Hardware Selection - Servo vs Stepper

- Servos feature a simplified control architecture, allowing direct PWM control.
- Servo motors were selected for S.T.A.R. because they provide smooth, precise motion with integrated positional feedback.

Attribute	Servo Motors	Stepper Motors
Rotation Range	Typically limited (e.g., 180° –270°), suitable for fixed-orientation systems	Capable of continuous 360° rotation
Motion Smoothness	Smooth, continuous motion ideal for fluid pan/tilt tracking	Discrete stepwise motion can cause vibration or "jitter"
External Circuitry Required	Minimal (PWM controller such as PCA9685)	Moderate to high (stepper driver, current regulation, optional encoders)
Torque Characteristics	High torque at low speeds; torque chosen via rated servo strength	Torque drops off rapidly at higher speeds
Suitability for S.T.A.R.	Excellent—supports smooth tracking, stability, and simplified integration	Less ideal due to vibration, complexity, and tracking instability

Hardware Selection - Motor Controller

- Direct PWM generation using the ESP32 deemed impractical for multi-servo control due to processor load.
- The Adafruit PCA9685 selected for its 16-channel capacity and I2C interface, enabling precise and synchronized servo control.

Attribute	PCA9685 (Adafruit)	TLC5940
Control Type	12-bit PWM over I2C	12-bit PWM over SPI
Communication Interface	I2C (addressable up to 62 devices)	SPI (faster, but fewer devices)
Ease of Integration	Very easy—widely supported by Arduino, ESP32, and Raspberry Pi	Moderate—requires more configuration
Power Handling	Separate servo power rail; supports up to 6V	External current sink required
Chosen for S.T.A.R.	Offers high channel count, reliable timing, and I2C compatibility with ESP32-S3 for smooth servo control.	TLC5940 lacks flexibility for multi-servo synchronization.

Hardware Selection - Buck vs LDO

- Buck converters offer high efficiency when stepping down large voltage differences, minimizing heat generation.
- LDOs were not suitable because they dissipate excess voltage as heat.

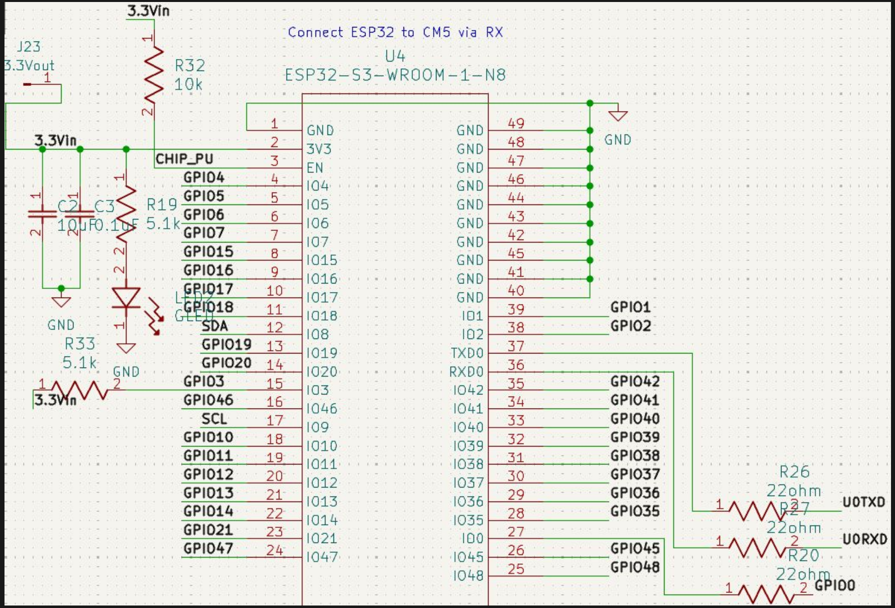
Attribute	Buck Converter (Switching)	LDO (Linear Regulator)
Heat dissipation	Low (most energy converted, small losses).	High (dissipates (Vin−Vout)×I as heat).
Transient response	Good, but requires loop compensation; faster switching designs respond well.	Very fast and clean transient response for small load steps.
Complexity and BOM	Higher: inductor, diode/synchronous FETs, switching IC, output/input capacitors, layout care.	Lower: single component, few external caps.
Transient response	Good, but requires loop compensation; faster switching designs respond well.	Very fast and clean transient response for small load steps.
Efficiency	High (typically 80–95% depending on load and design).	Low when Vin ≫ Vout (efficiency = Vout/Vin).

Hardware Design - Power Budget Table

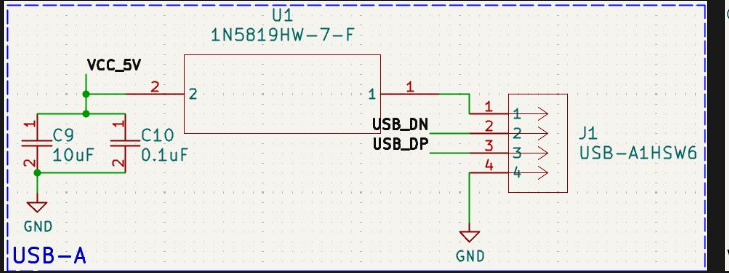
- The S.T.A.R. system requires a minimum power supply of 24V and 15A.
- A 600W (24V 25A) PSU was chosen to meet the voltage and current requirements for the system

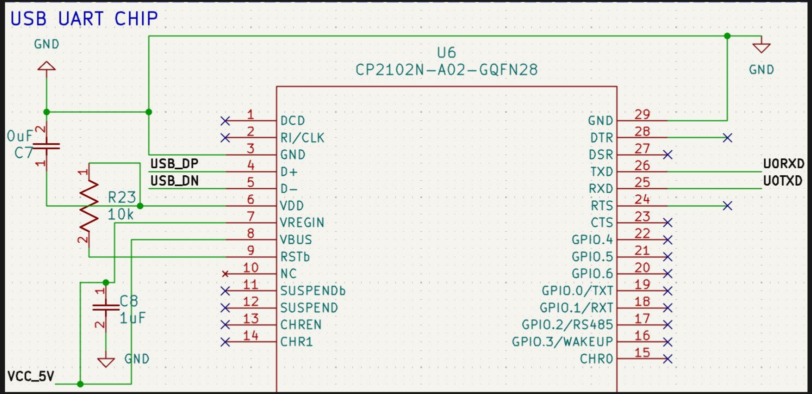
Component	Voltage Requirement	Current Requirement
ESP32	3.3V	1A
PCA9685	5V	400mA
35kg Servo Motors (3x)	~8V	3.9A / Each (Stall)
20kg Servo Motors (2x)	~7V	2.7A / Each (Stall)
TOTAL	~24V	~15A

Overall Schematics - ESP32



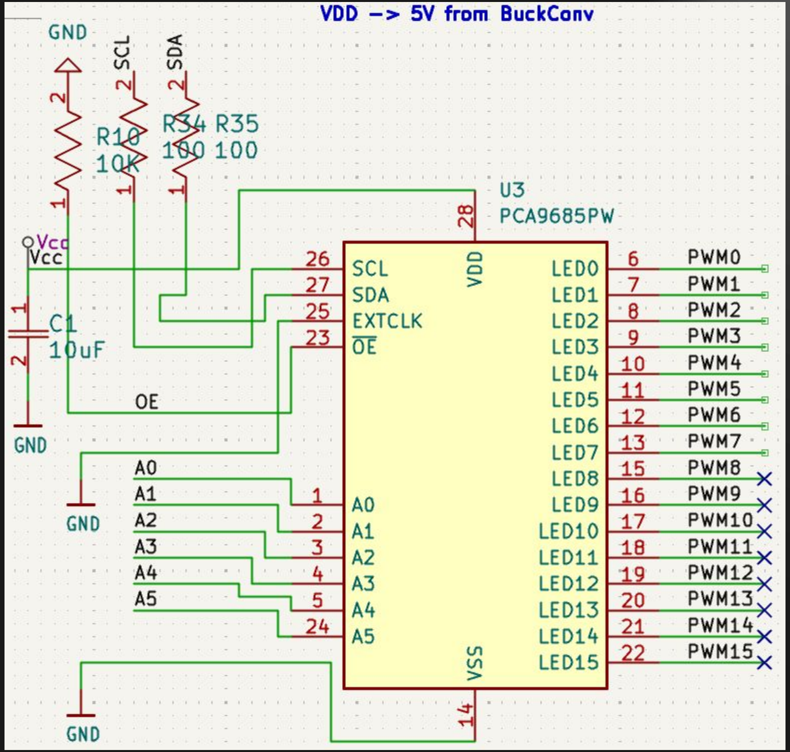
- Custom ESP32 board
- USB-A interface for programing MCU
- Custom board for ESP32 was chosen for S.T.A.R. due to ease of development and readily available documentation

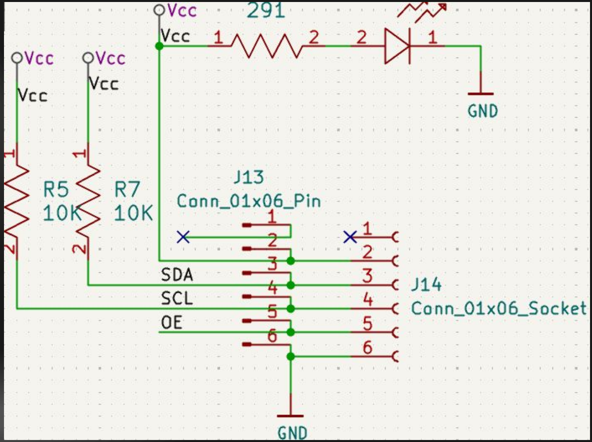




Overall Schematics - PCA9685

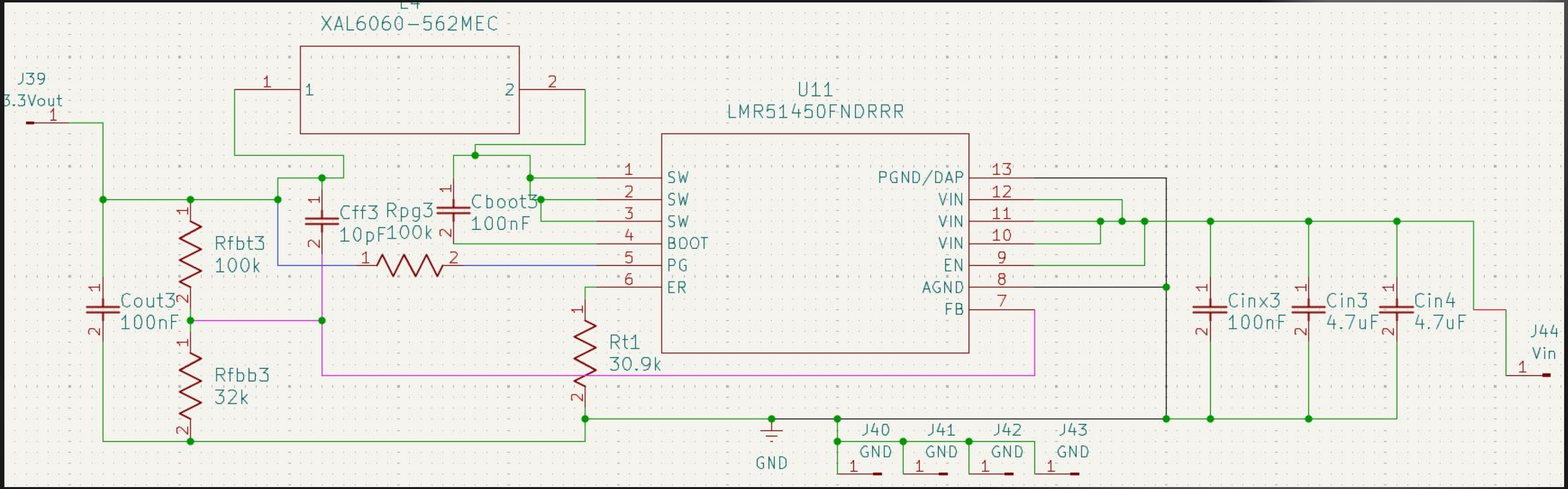
- Custom motor controller board was chosen for S.T.A.R. to meet PCB requirements
- No power rail for motors. External source needed





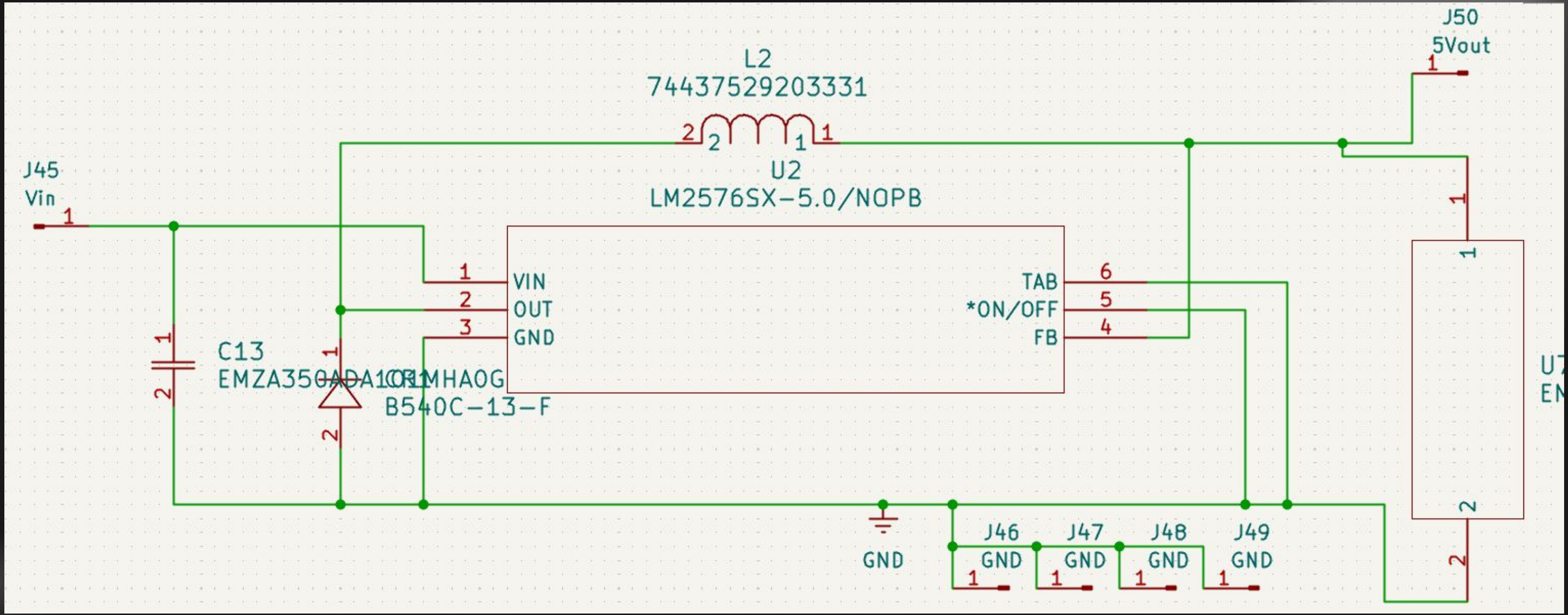
Overall Schematics - 3.3V Regulator

- TI LMR51450 used for its robust (36Vin, 5Aout) ratings
- High voltage rating needed because of 24V PSU
- Circuitry was designed using TI WEBENCH power designer



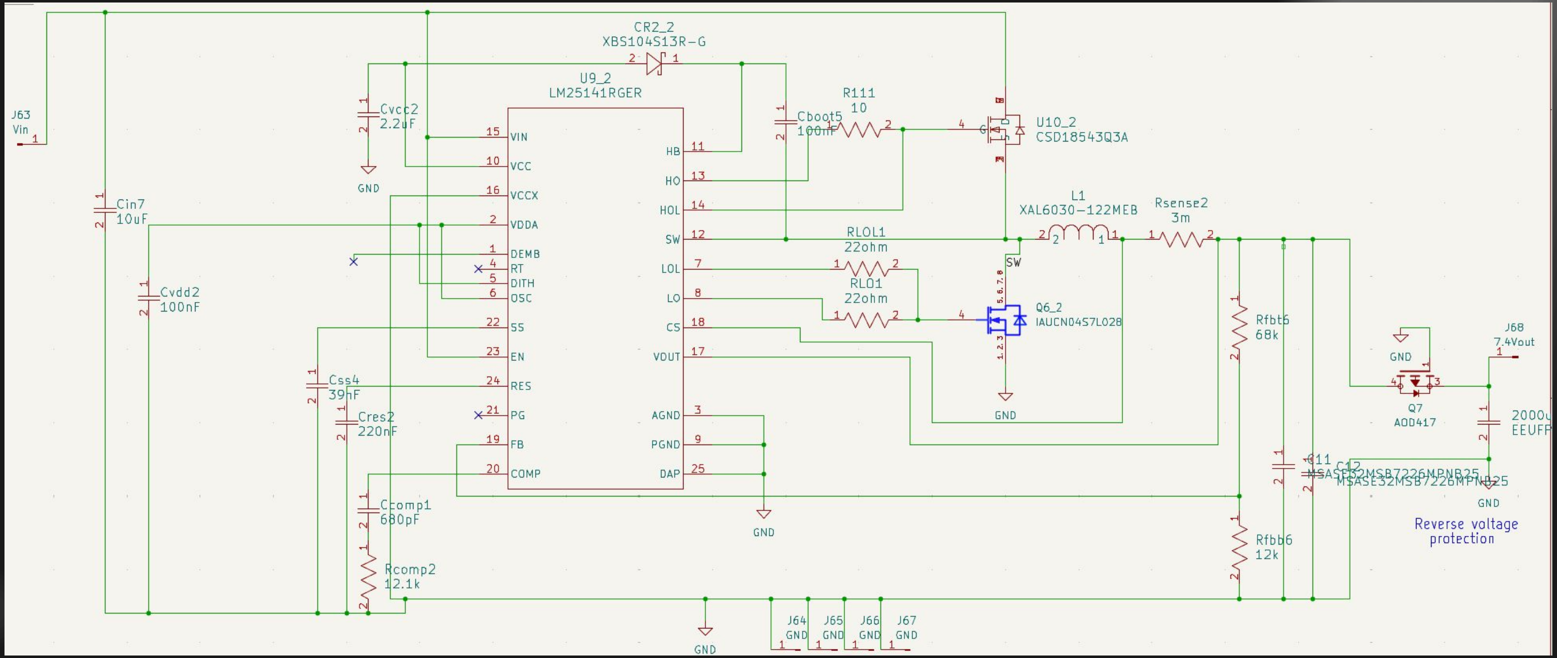
Overall Schematics - 5V Regulator

- TI LM2576 used. Recommendation from Dr. Weeks because of availability and simplicity.
- Rated for 45Vin and 3Aout
- Circuitry was designed using TI WEBENCH power designer

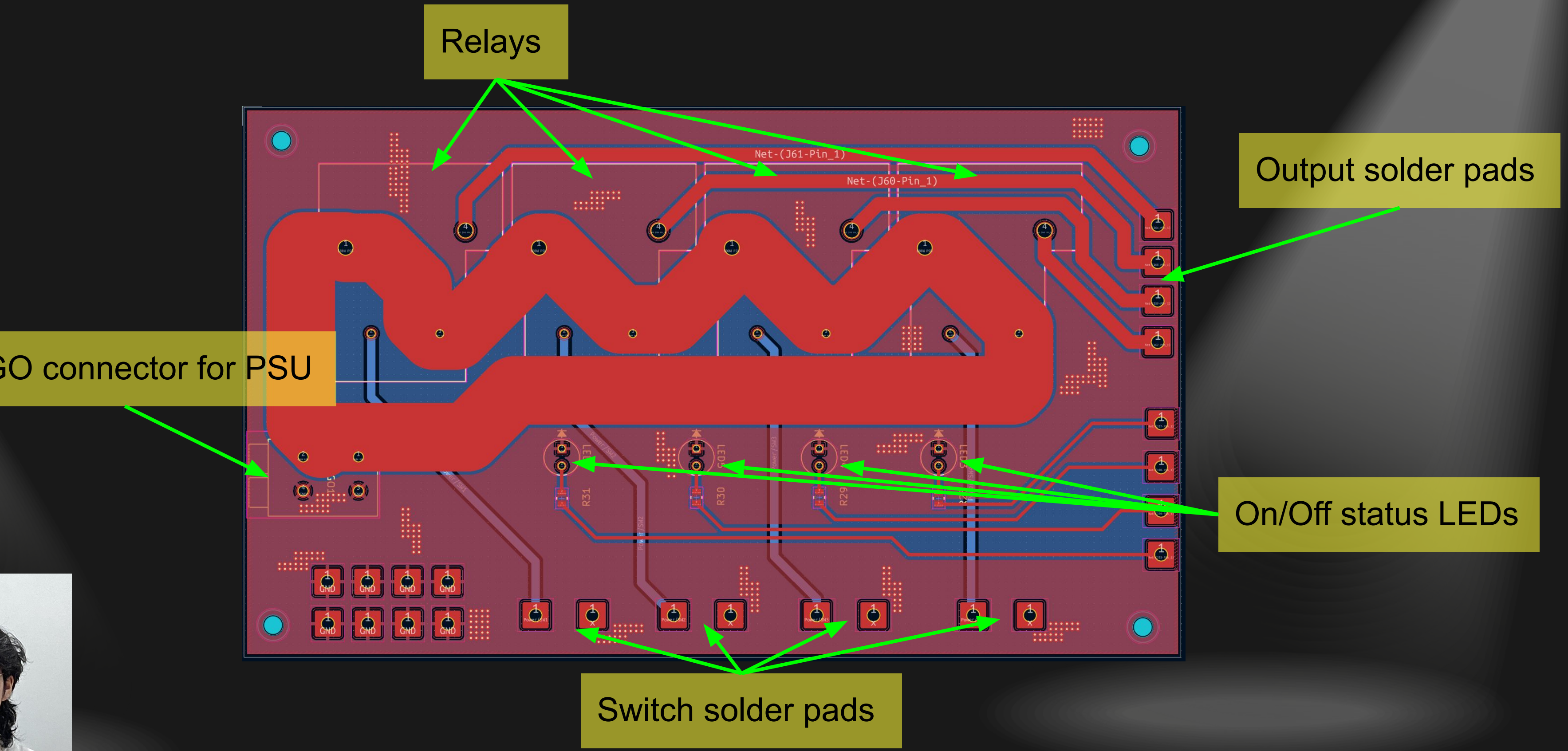


Overall Schematics - 8V Regulator

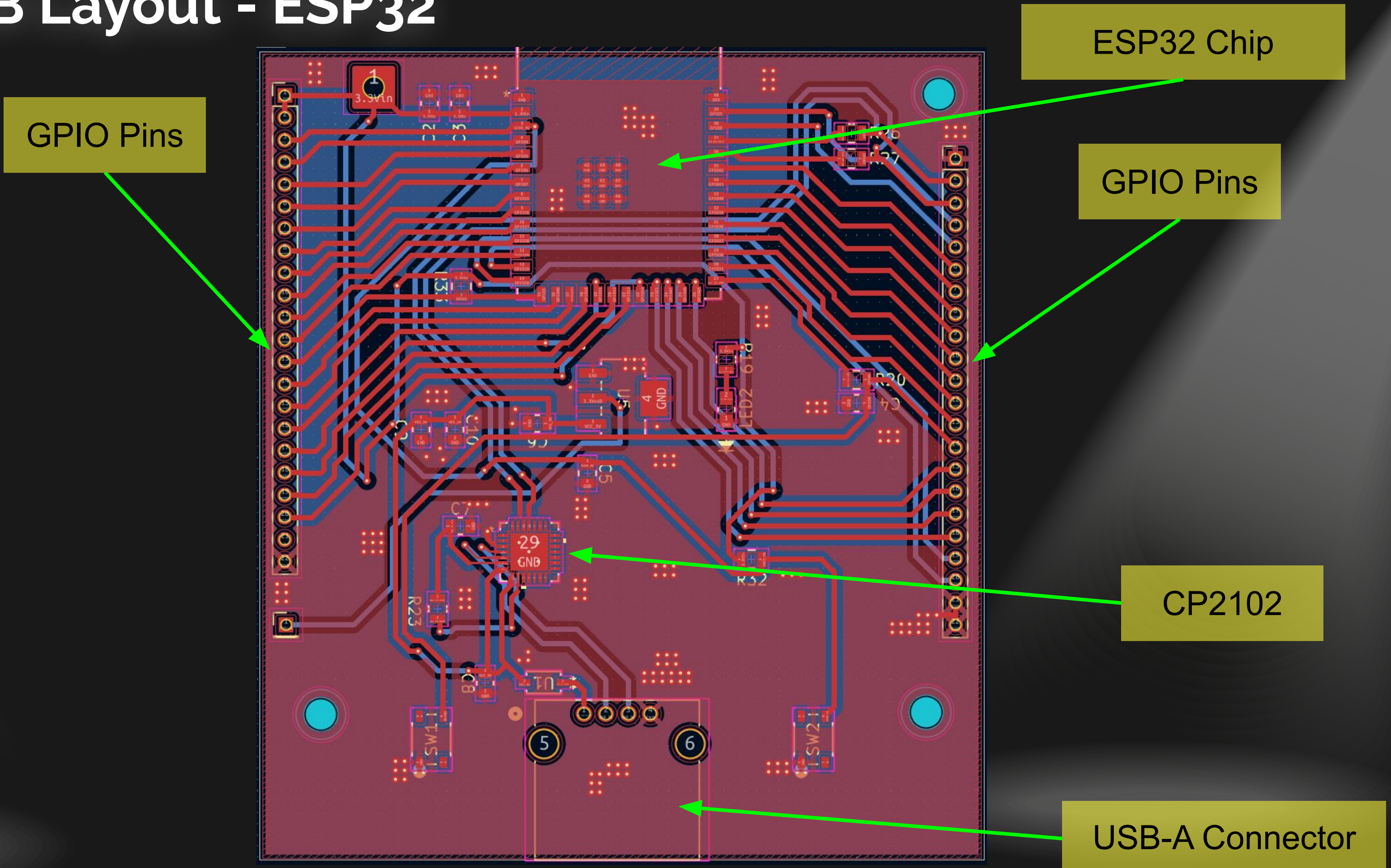
- TI LM25141 used. Rated for a whopping 30Aout.
- Output current has to be very high to support the combined stall current of our motors
- AOD417 and 2000uF cap used at output for reverse voltage protection



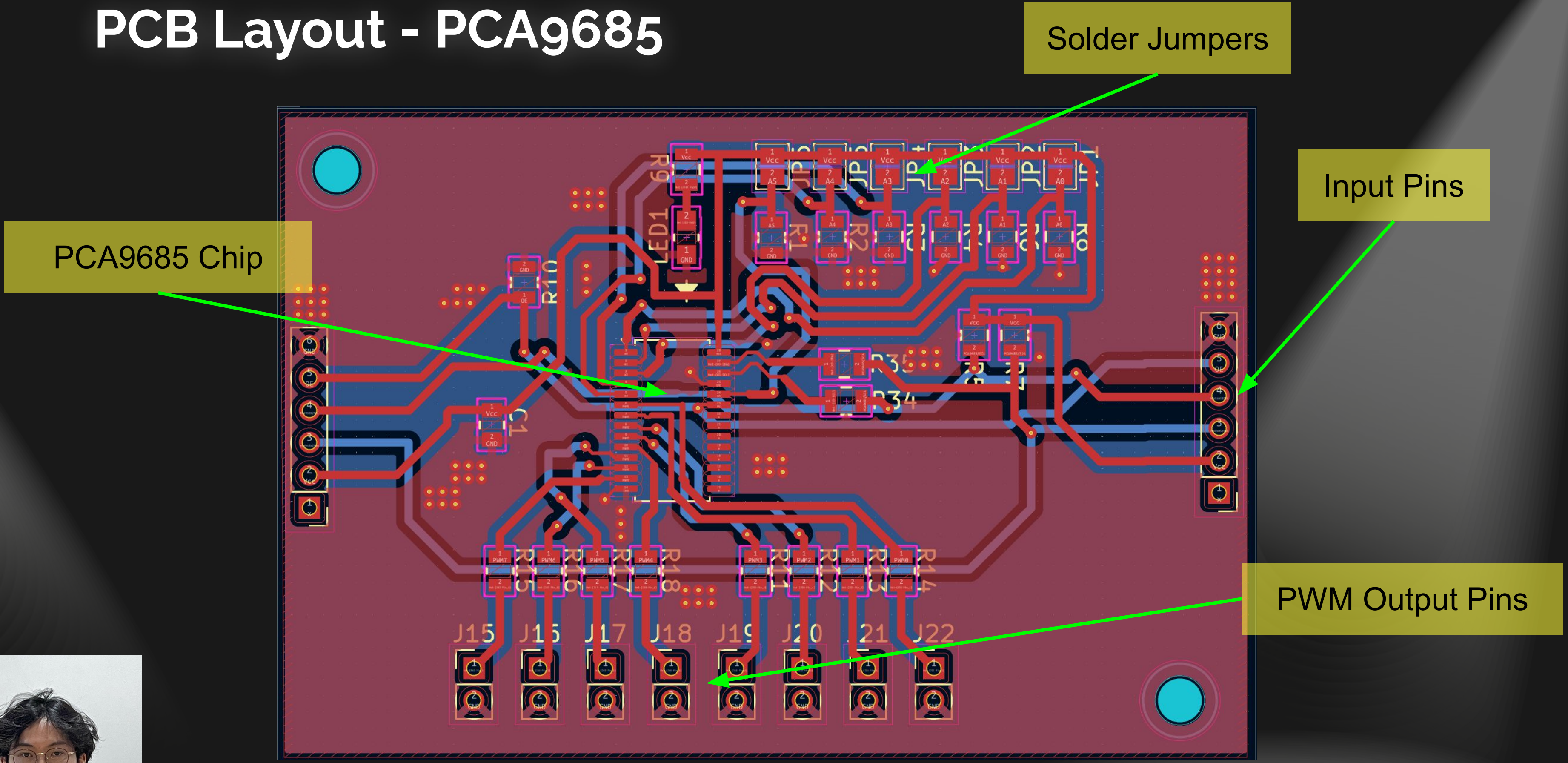
PCB Layout - Terminal Board



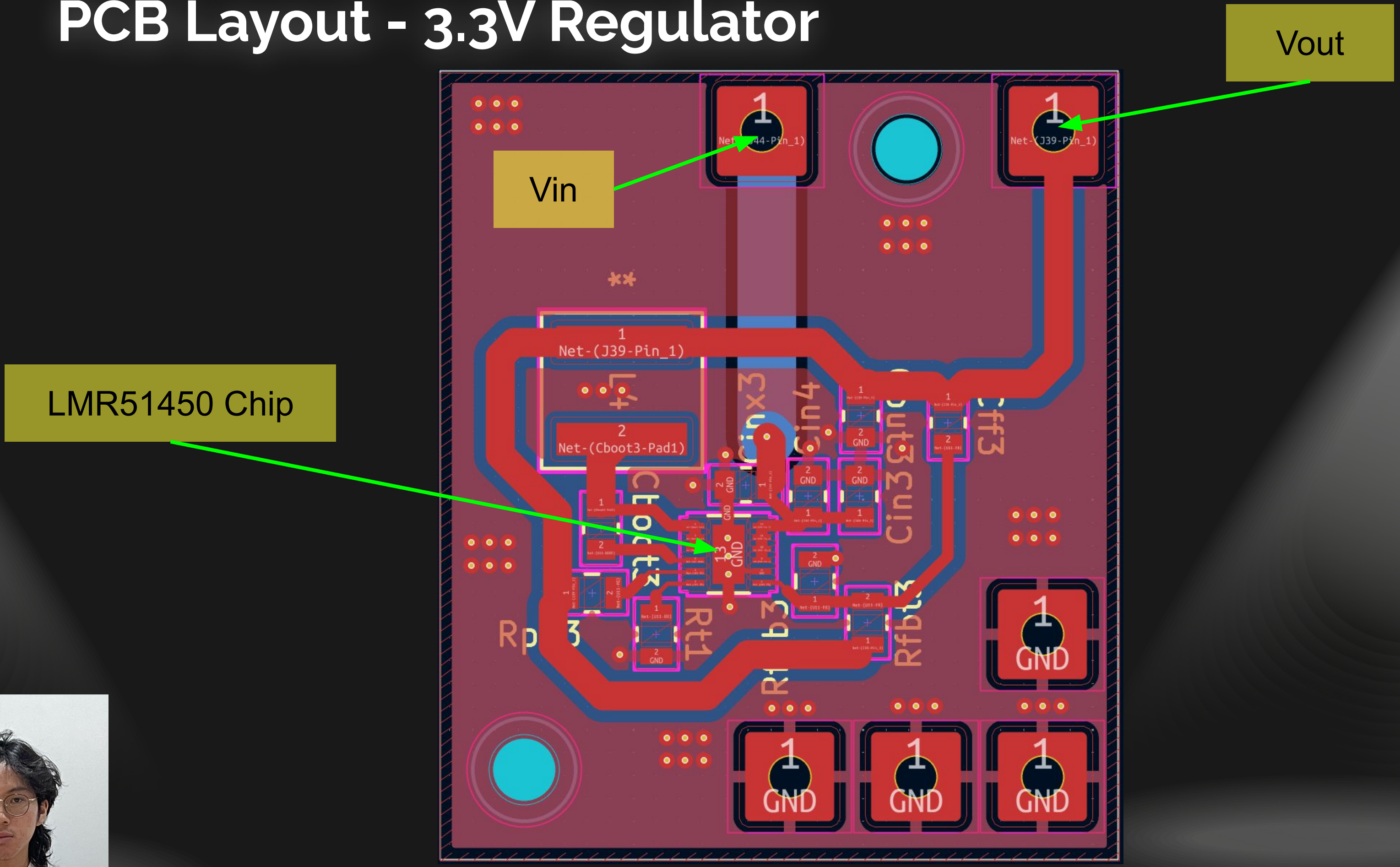
PCB Layout - ESP32



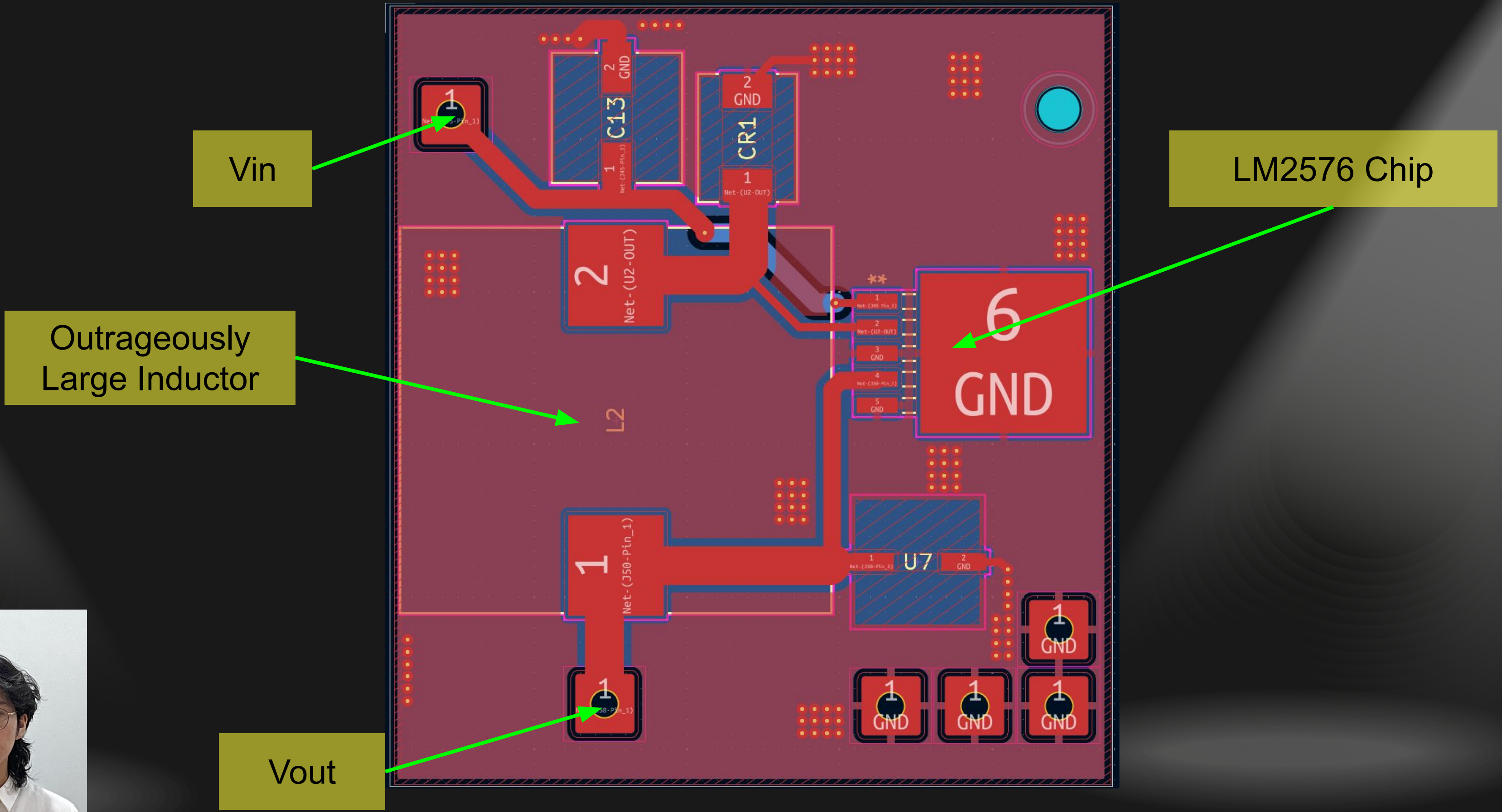
PCB Layout - PCA9685



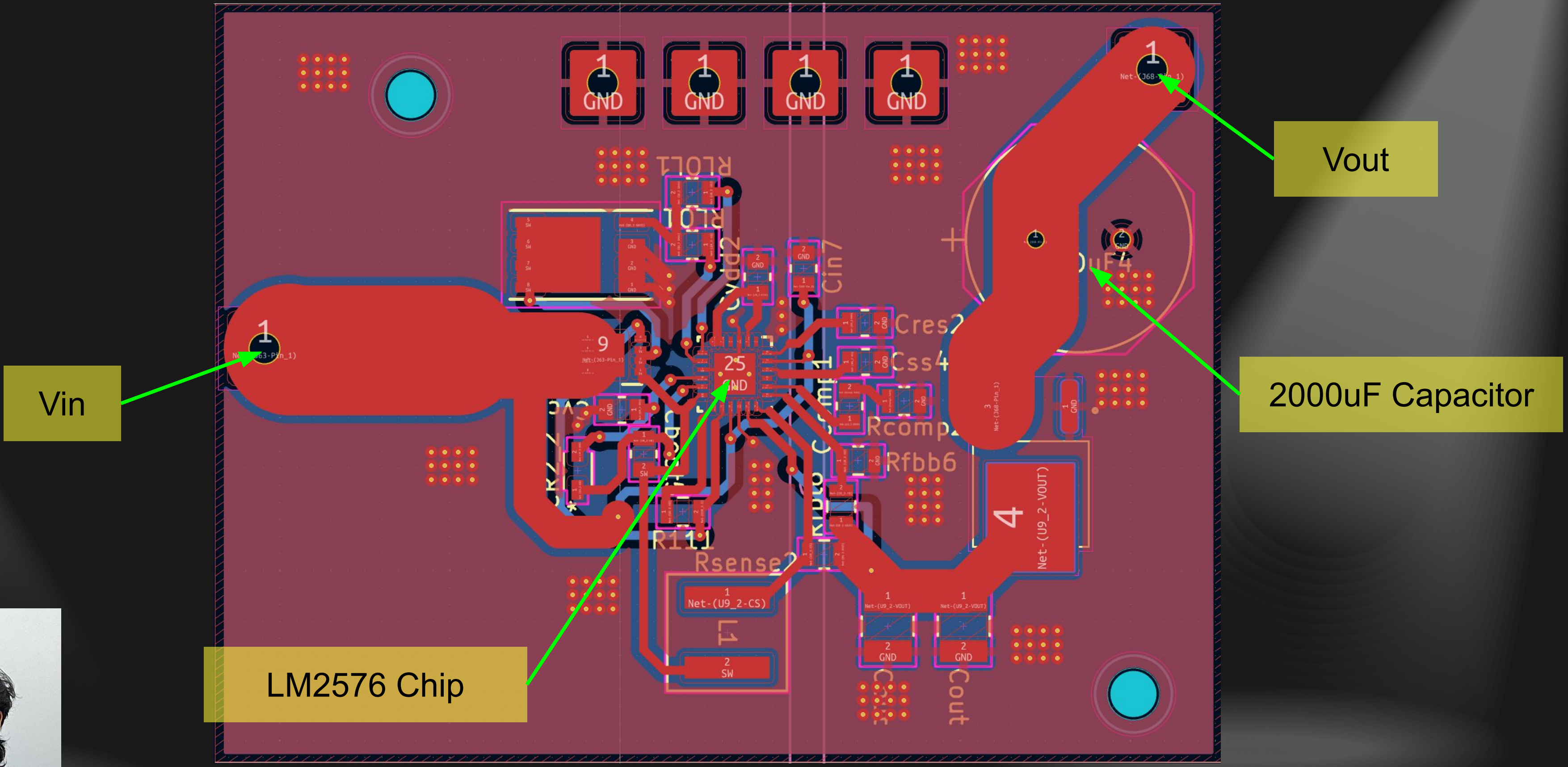
PCB Layout - 3.3V Regulator



PCB Layout - 5V Regulator



PCB Layout - 8V Regulator



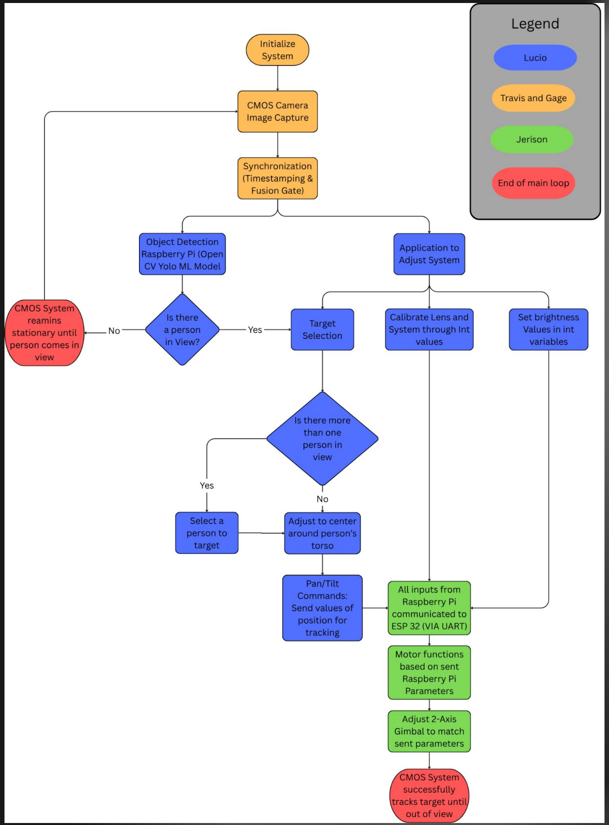
Software Design

Objectives

- Detect and track a single performer in real time
- Drive pan/tilt gimbal toward performer center
- Maintain smooth spotlight motion (<300 ms latency)
- Communicate target error data to MCU in real time

Pipeline

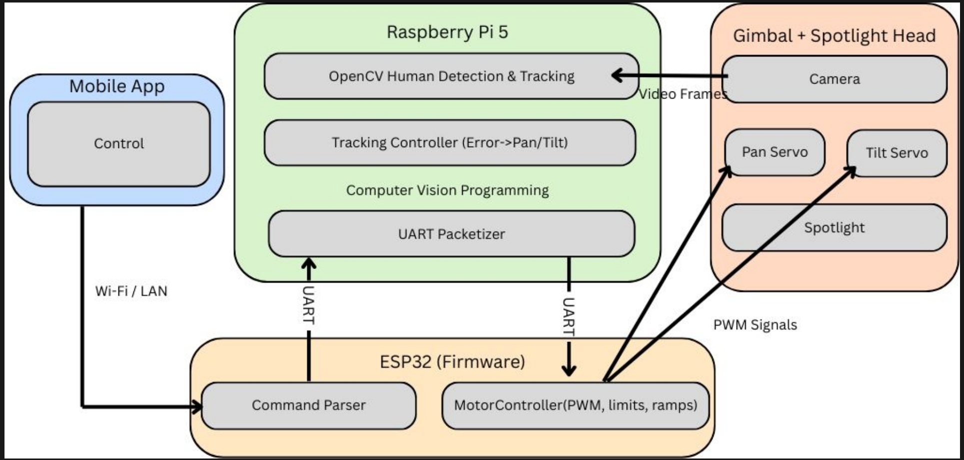
- Camera -> Detection -> Tracking -> Target Coordinates



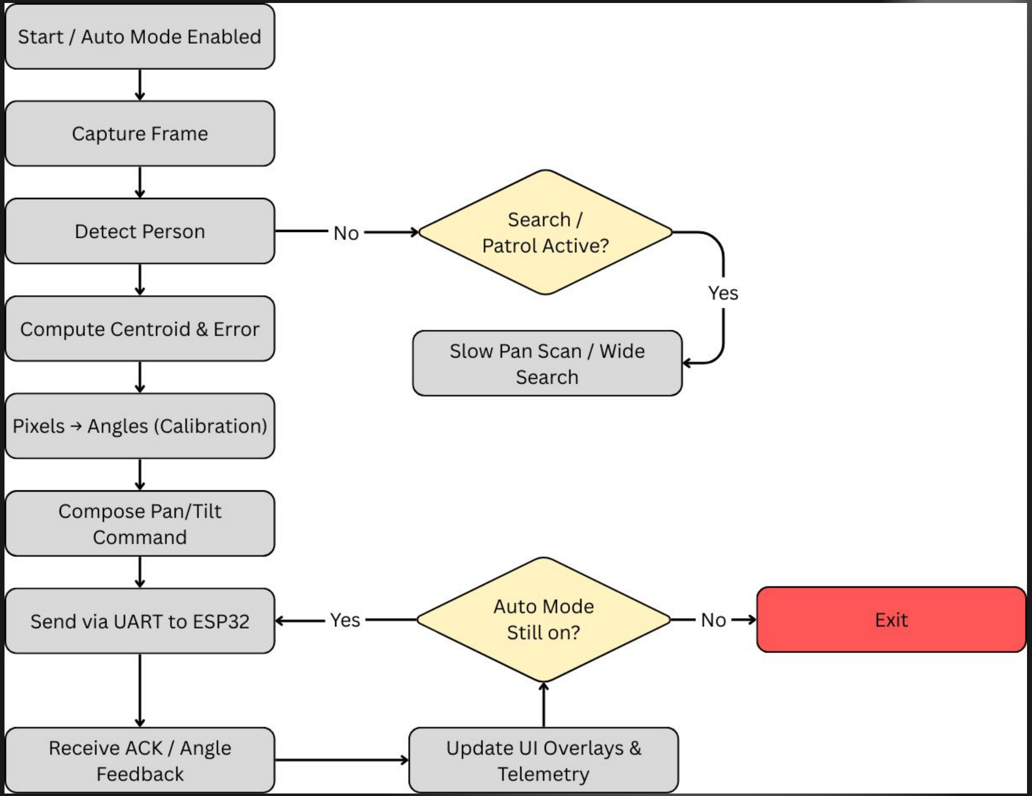
Software Design

Components

- Computer Vision - Detection
- ESP32 - Motorization
- Mobile App - Control



Computer Vision



Role:

- Capture live video from camera
- Detect and track performer
- Output target position data for motor control

Computer Vision

Raspberry Pi CM5

- Linux Based Embedded Computing
- Supports Native Camera interfaces
- Compatibility with Machine Learning Frameworks

NVIDIA Jetson Nano

- Integrated CUDA-capable GPU
- Excel in deep learning workloads
- High cost and power consumption

Criterion	Raspberry Pi CM5	Nvidia Jetson Nano
Primary Role	Embedded Vision Programming	GPU-accelerated edge AI
Compute Capability	Quad-Core ARM Cortex-A76	Quad-Core ARM Cortex-A57
Compute Capability	Moderate-High (CPU + GPU)	High (CUDA GPU)
Power Consumption	4 - 8 W	5 - 10 W
Embedded Suitability	High	Medium
Cost Efficiency	High	Medium

Computer Vision

OpenCV

- Computer Vision Library
- Supports camera Input, object tracking, and image preprocessing
- Optimized for ARM-based systems

Criterion	OpenCV (Selected)	TensorFlow Lite	PyTorch/Torch Script	MediaPipe
Primary Role	End-to-end CV toolkit	Edge inference for TF models	Research DL; scripted deployment	Prebuilt perception graphs
Edge Readiness	High (CPU/NEON; broad backends)	High (quantization, delegates)	Medium (heavier runtime)	High (mobile/edge focused)
Pipeline Breadth	Wide (capture → inference → tracking)	Narrow (inference only)	Narrow → Medium	Medium (task-focused)
Integration Effort	Low	Medium	Medium–High	Medium

Computer Vision

Criterion	YOLO (Selected)	MobieNet-SSD	HOG + SVM
Detection Accuracy	High	High	Very High
Inference Latency	Low (~33–66 ms / frame)	Very Low (~25–50 ms / frame)	Medium (~100–200 ms / frame)
Throughput	15-30 FPS	20-40 FPS	5-10 FPS
Robustness	High	Medium	Low
Resource Usage	Moderate(~300-500 MB)	Low (~200-350 MB)	Moderate (~150-250 MB)
Modern Relevance	State of the art	Widely Used	Legacy

YOLO (You Only Look Once)

- Performs object localization and classification in a single forward pass
- Delivers high detection accuracy with low end-to-end latency
- Well-suited for real-time inference on embedded platforms

Communication

Role:

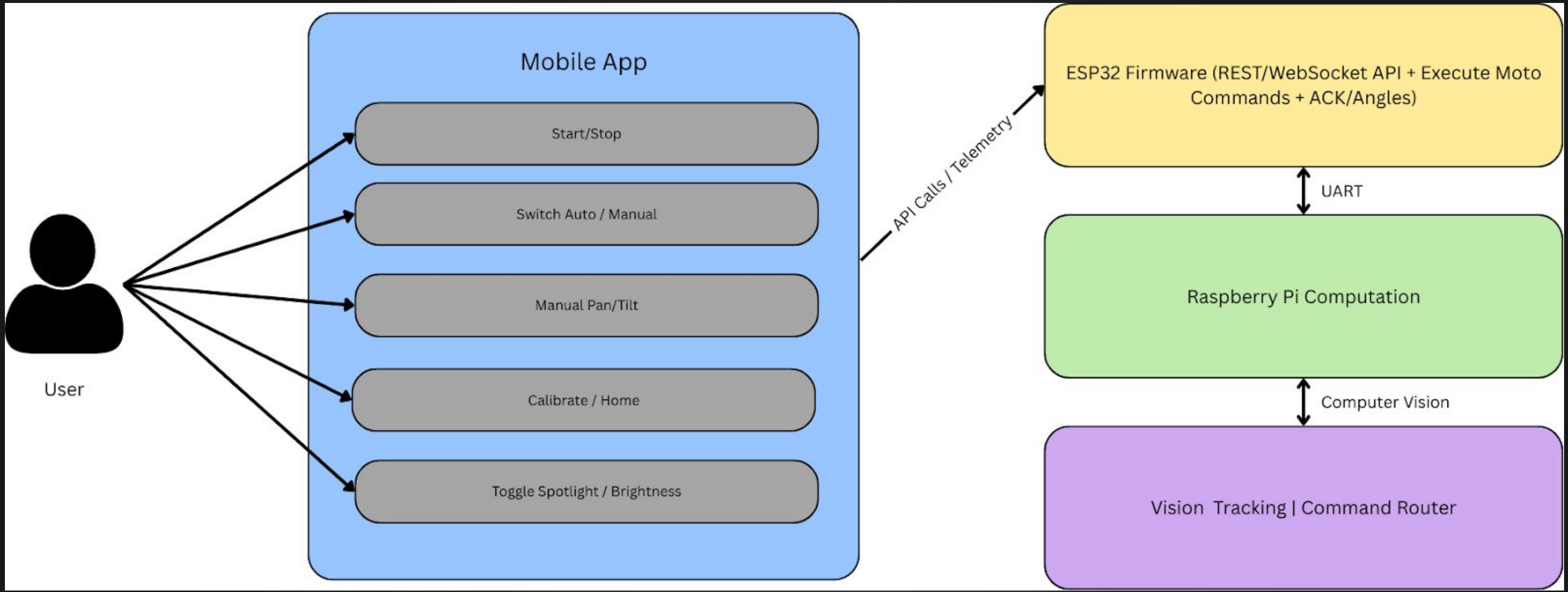
- Communication between ESP32 and Raspberry Pi CM5

UART

- Simple, point-to-point serial communication protocol
- Low and predictable latency for real-time data exchange
- Well-suited for reliable MCU-to-processor communication in embedded systems

Method	Wires	Clocking	Practical Rate	Determinism & Jitter
UART	TX, RX, GND	Asynchronous	115.2 kbps–1 Mbps	Good with framed packets and acks
I2C	SCL, SDA (+pull-ups), GND	Synchronous (open-drain)	100 kHz/400 kHz/1 MHz	Bus arbitration; moderate jitter under load
SPI	SCLK, MOSI, MISO, CS, GND	Synchronous (push-pull)	1–20+ MHz	Very good, clocked

Mobile App



Role:

- Provide a user interface for system control
- Allow mode switching and manual override
- Display system status and feedback

Communication

HTTP/REST

- Stateless request–response communication model
- Simple and reliable for configuration and system commands
- Higher overhead per message due to headers

WebSocket

- Persistent, bi-directional communication channel
- Enables low-latency, real-time data streaming
- Reduced overhead after initial connection

Criterion	HTTP/REST	WebSocket	MQTT
Setup cost	Request per action; keep-alive helps	One upgrade; then steady state	Broker + client connects
Push capability	No (client-pull only)	Yes (bi-directional)	Yes (publish/subscribe)
Per-msg overhead	High (headers)	Low (frame header)	Low (tiny header)
Backpressure	Ad-hoc	Event-loop awaits/queues	Broker buffer + QoS acks
Reconnect	Per request	Heartbeats & fast resume	Retained messages + QoS
Best use here	Config, logs	Primary real-time control	Future multi-client/IoT

Mobile App

React Native
- Cross Platform Development
- Javascript
- Single codebase

Flutter
- Cross Platform Development
- Compiled rendering engine
- Dart

Native Android and iOS Development
- Highest level of platform integration
- Separate Codebases
- Additional Language and Framework

Criterion	React Native (Selected)	Flutter	Native Android / iOS
Performance	High	High	Very High
Codebase Count	Single	Single	Multiple
Development SPeed	High	Medium	Low
Ecosystem Support	Very Large	Medium	Platform Specific

Mobile App

FastAPI

- Asynchronous Python web framework
- REST API and WebSocket Communication
- Low Latency Command Handling

Criterion	FastAPI	Flask	Node.js
Async Support	Yes	Limited	Yes
WebSocket Support	Yes	Yes	Yes
Latency Suitability	High	Medium	High
Embedded Fit	High	Medium	Medium

PCB Testing

- PCB testing will be done multiple ways.
  - Lab Tests (Oscilloscope, Thermal Camera, Multimeter, etc.)
  - Connectivity Tests (UART, I2C, etc.)
  - Operational Tests (Full motor control, Terminal board operation, etc.)
- Successful and unsuccessful tests will be recorded and revisions will be made to improve the functionality of the system
- Tests will be conducted multiple times to insure system reliability and robustness

Human Detection Software Testing

- The human detection algorithm was tested using live camera input under representative lighting conditions.

- The system successfully identified all four group members within a single frame with high confidence.

- Detection results demonstrated reliable multi-person recognition and stable tracking.

- These results validate the software’s suitability for integration with real-time tracking and control.

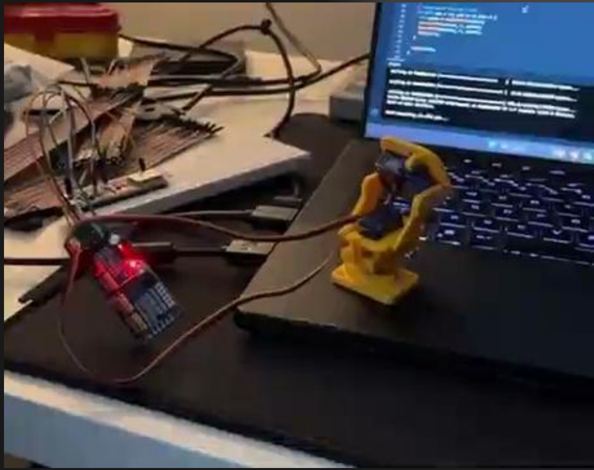
Gimbal Testing

- A scaled-down gimbal prototype was designed and 3D printed to validate the mechanical concept before full-scale fabrication.

- Low-torque servo motors were used to demonstrate basic pan and tilt motion and confirm control logic.

- The prototype verified axis alignment, range of motion, and structural feasibility under dynamic movement.

- Results from the mini gimbal testing informed design refinements for the full-size motor system.



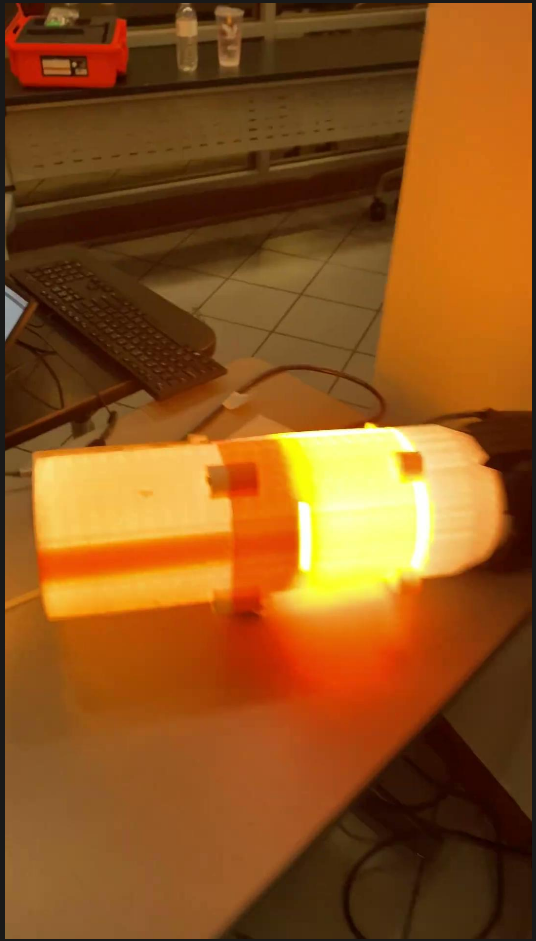
Illumination System Testing

- The measured spot size matches the expected value of approximately 2.5 m at the target distance.

- The illumination exhibits a uniform intensity distribution across the beam with a warm, consistent color temperature.

- Imaging results show sharp beam edges with minimal chromatic aberration, indicating good optical alignment and lens performance.

- The system maintains stable output during continuous operation, demonstrating consistent optical and thermal behavior.



Budget

Part Name	Vendor	$/unit or lot	QTY	Note
ESP32-S3 Chip	Digikey	$5.00	1	
Raspberry Pi Compute Mod. 5	Digikey	$65.00	3	65-80 depending on model
Servo Motors (shutter mech.) 4x	Amazon	$20.00	1	
Servo Motors (high torque)	Amazon	$60.00	3	50-110 depending on model
Portable Monitor	Amazon	$60.00	1	
RP Camera HQ	Digikey	$50.00	1	
HPL757 Lamp	ETC	$19.00	1	
Illumination lens	ETC	$60.00	1	
Source Four Reflector	ETC	$65.00	1	
VIS-NIR Achromatic Lens	Edmund Optics	$79.38	1	
Uncoated Plano-Concave Lens	Edmund Optics	$28.62	1	
Misc. PCB components	Digikey	$150	1	Rough estimate
PCB Fabrication	JLCPCB	$300	1	Rough estimate
EST. TOTAL		$1,104.00		Will most likely exceed est. total due to unforeseen costs

Work Distribution

Photonics Engineer	Responsibilities
Gage	Illumination Lens design
	Source Selection
	Illumination Gate System
	Housing Design

Computer Engineer	Responsibilities
Lucio	Computer vision algorithms
	System UI
	System logic coding
	Motor command code
	Senior Design website

Photonics Engineer	Responsibilities
Travis	Imaging Camera Selection
	Imaging lens design
	Imaging Housing Design
	Magnification and lens calculations

Electrical Engineer	Responsibilities
Jerison	PCB Design
	Motor Logic for Gimbal
	Illumination Shutter System
	3D Design and Printing
	Circuitry and Electrical Components

Progress and Plan for Completion

- **Progress:**
  - All major subsystems have been successfully developed and validated independently.
  - The human tracking, imaging system, illumination system, and motorized gimbal each operate as intended.
- **Plan for Completion:**
  - Finalize and assemble the full-scale gimbal system.
  - Validate system operation through on-site testing at the UCF Theatre.
  - Implement coordinate mapping to enable accurate spatial pointing of the spotlight.
  - Complete development and integration of the control application.
  - Test and validate custom PCBs under full system operation.