SENTRY TRACKING ATTACK TURRET A.K.A STAT PROJECT

SD2 CDR PRESENTATION

DEVAK SHARMA
CHRISTIAN SOLANET
CARLOS GARCIA
NICHOLAS MARTUCCI

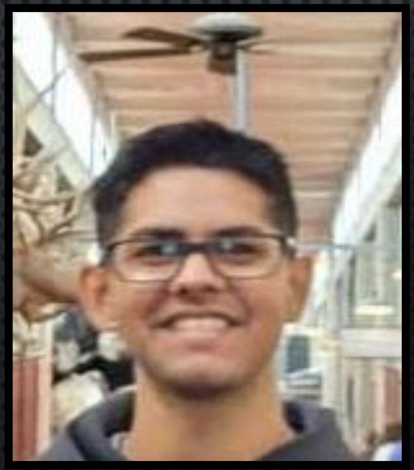
MEET THE TEAM AND REVIEWERS

Dr Wei Sun
Professer

Dr. Suboh Suboh
Assistant Lecturer

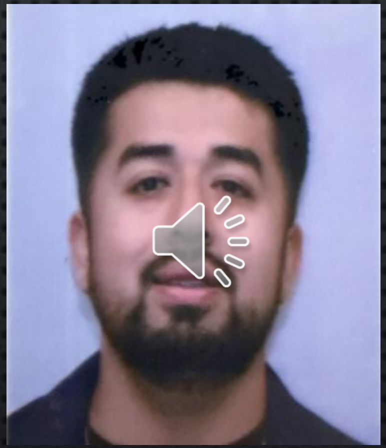
Dr. Qifeng Li
Associate Professor

Christian Solanet
Computer Engineering



Devak Sharma
Computer Engineering

Nicholas Martucci
Electrical Engineer



Carlos Garcia
Electrical Engineer

MOTIVATION/BACKGROUND

- Counter Rocket, Artillery and Missile (C-RAM)
- Uses sensors integrated with AI and live ammunition to track and predict path to intercept
- High Energy Laser Intercept Systems such as DragonFire by UK and various weapons in US Army

GOALS AND OBJECTIVES

Main Goals

- Build Stationary Turret with detection with in 120 degree area
- Turret Locks on to target within 2 seconds of waking up
- Access box signals object detection
- Hit target with (8in x 8in) 60% accuracy

Main Objectives

- Develop modular two-board control
- Implement dual-camera detection with HuskyLens integration
- Design custom power regulation and actuation control PCBs
- Design and fabricate structural turret housing using PETG

Advanced Goals

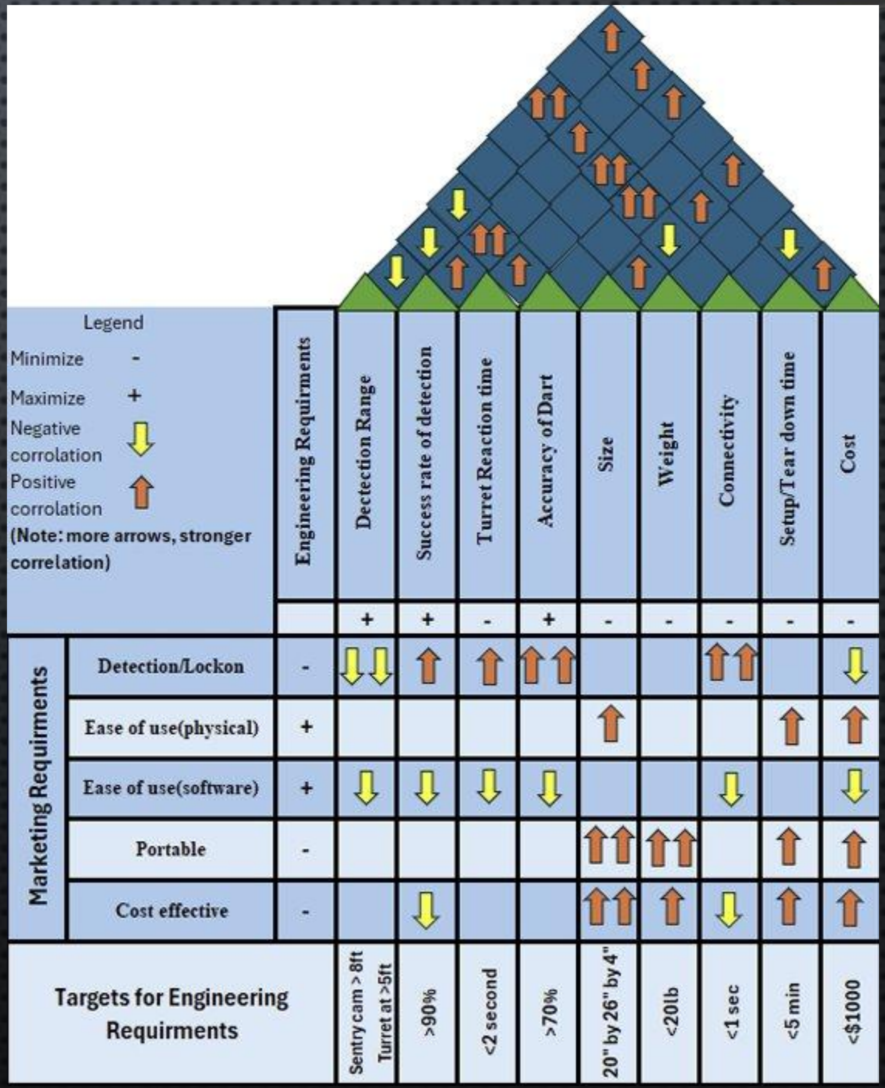
- Turret hits target with nerf dart with above 80% accuracy
- Be able to differentiate between friend or foe

Stretch Goals

- Be able to lock on to aerial target from sentry camera data
- Have lock on to target within 1 second of detection
- Be able to hit target in air with 75% accuracy

ENGINEERING SPECIFICATIONS/HOUSE OF QUALITY

Parameter	Description	Target Value/Unit
Sentry Range Detection	Sentry cameras must be able to detect an object that has entered its desired range and create a bounding box around said object.	Greater than eight feet
Success Rate of Description	This would be the success rate of sentry camera detection of object along with the turret detection and laser diode lock on.	Greater than 90%
Turret Reaction Time	The time that it takes the turret to wake up, position itself towards the object, detect the object, and determine if it should lock onto it must be quick.	Less than or equal to two seconds



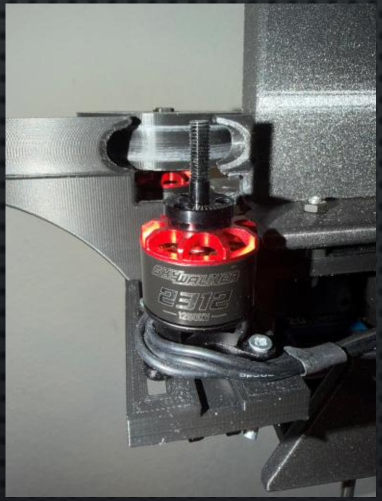
REGULATOR COMPARISON

	TPS Series	LM2679x Series	LMR604x Series
Layout Design	Extremely Difficult	Medium	Hard
Vin Range	3V - 16V	8V - 40V	3V - 36V
Vout Range	.6V to 7V	1.21V - 37V	1V - 20V
Current out max	4A	5A	3A
Package	SMD IC	Through Hole IC	SMD Underneath IC

LAUNCHER SELECTION

	Brushed DC Motor	Brushless Motor	Spring Loaded
Commutation:	Brushes	Electronic	
Control:	Encoder	ESC+PWM	Mechanical
Reload	Automated	Automated	Manual
Cost:	$	$$	$

	2312 980KV	2312SL 1250KV	2316SL 1400KV
Working Voltage:	11.1v – 14.8v	11.1v – 14.8v	11.1v – 14.8v
No Load Current @12V	~1.05A	~1.22A	~1.46A
Peak Current (Start up)	~35A	~45A	~50A
RPM:	~11,760	~15,000	~16,800
Cost	$24.99	$24.99	$24.99



PAN/TILT ACTUATION SELECTION

	Stepper Motors	Servo Motors	Brush/Brushless Motors
Range:	Unlimited	180-270	Unlimited
Position:	Stepper Driver	PWM	Encoder
Positional:	Open Loop	Closed Loop	Closed Loop
Cost:	$$	$$	$$$

	SF 55g	TD-8120MG	TD-8130MG
Working Voltage:	6V	7.2V	8.4V
No-load Speed	.15s/60deg	.14s/60deg	.22/60deg
Stall Current:	~1.9A	~2.7A	~3.4A
Position Range:	180 degree	270 degree	180 degree
Torque:	15Kg	22.8Kg	27Kg
Cost	$11.99	$17.99	$21.99

LASER SELECTION

	VLM-650-03	Adafruit 1056	VLM-520-29
Wavelength	650nm (red)	650nm (red)	520nm (green)
Power	~1mW	~5mW	~5mW
Driver Required	External Resistor	Integrated Driver	External Driver
Operating Voltage	2.6v – 5v	2.8v - 5.2v	7v – 10v
Typical Operating Current:	40mA	25mA	100mA
Cost	$25	$19	$63

IMU COMPARISON

Option	Features	Pros	Cons
6 DoF	Accel and gyro	Simpler to implement and cheap	No magnetometer
9 DoF	Accel, Gyro and Magnetometer	Better orientation and accuracy	Can drift with EMI and higher cost
Industrial IMU	Lowest noise and best stability	Extremely accurate	Extremely expensive and power hungry

IMU	Pro	Cons	Power Usage	Interface	Cost
MPU-6050	Cheap and available	Older and higher drift	Low	I2C and some SPI	~$6
ISM330DHCX	High stability and low noise	Expensive	Very low	I2C and SPI	~$20
LSM6DS3	Power efficient, widely used	Not as precise	Very low	I2C and SPI	~$10

MCU SELECTION

Features	ESP32-S3-WROOM-2-N32R16V	Arduino Uno R3 (ATMega328p)	Arduino Mega 2560 (ATMega2560)	Raspberry Pi Pico 2 (RP2350)
CPU/Lock Speed	Dual-core Xtensa LX7 at 240MHz	Single core 8-bit AVR RISC-based at 16 MHz	Single core 8-bit AVR at 16 MHz	Dual-core ARM Cortex-M33 at 150 MHz
RAM/Flash Memory	512 KB SRAM and 16 MB of flash memory	2 KB SRAM and 32 KB of flash memory	8 KB SRAM and supports 64,128, or 256 KB of flash memory	520 KB SRAM and up to 16 MB of flash memory
Pros	Dual-core CPU for parallel tasks, able to run TinyML, able to support multiple devices, supports Wi-Fi/Bluetooth	Low power, ease of integration, simple	Same as the ATMega328p, allows for more I/O tasks with high GPIO count	Provides more RAM for processing low resolution images, dual core allows for parallel tasks, and flexible options for GPIO
Cons	Limited RAM for high resolution ML, draws more power	Not enough RAM for any image tasks, slow clock speed	Too slow for real-time image processing, not enough memory storage	Does not support Wi-Fi or Bluetooth, slow USB transfer speeds since it uses USB 1.1
Interface Options	33 GPIO pins, DVP, SPI, UART, I2C	23 GPIO pins, supports UART, SPI, I2C	54 GPIO pins, can support UART, SPI, I2C	30-48 GPIO pins (depending on model), can support UART, I2C, SPI, or USB 1.1
Voltage	3.0-3.6V	2.7-5.5V	1.8-5.5V	1.8-5V
Current	Ranges from ~50 to 100mA	~10-20mA	~20m	~15-100mA
Cost	$5 - $15	~$2-5	~$1-5	~$1-10

SOFTWARE COMPARISON

Feature	ESP-IDF	Arduino	Zephyr
ESP32 Support	Yes	Yes	Yes
RTOS Integration	Native FreeRTOS	Wrapped FreeRTOS	Native Zephyr Kernel
Dual-Core Support	Yes	Limited	Limited
Library Ecosystem	Moderate	Large	Small
Build System	CMake	Arduino Build System	CMake & West
Learning Curve	Medium	Low	High
Deterministic Scheduling	Yes	Limited	Yes
Hardware Driver Access (I2C, SPI, etc.)	Yes	Limited	Yes

STATIONARY CAMERA COMPARISON

Features	Arducam Mega 3MP	Arducam Mega 5MP	OV2640	OV5640	HuskyLens
Interface	SPI	SPI	DVP	DVP	UART/I2C
Max Resolution	2048x1536	2592x1944	1600x1200	2592x1944	1600x1200
Frame Rate	15fps at full resolution, scalable	Slower at full resolution, scalable	15fps at UXGA. 30fps at SVGA	30fps at 1080p, 60fps at 720p	30fps, not scalable
Power	185mW-585mW	182mW-650mW	125mW-140mW	18mW-294mW	1.06W-1.15W
ESP32-S3 Multi-Cam Support	Yes	Yes	No	No	Yes
Features	Autofocus, IR cut filter, JPEG compression	Same as Arducam 3MP	Blob detection friendly	Autofocus, image compression	Built-in AI models
Cost	~$35	~$35-40	~$5	~$15-20	~$50-60

SOFTWARE – COLOR DETECTION ALGORITHM COMPARISON

Feature	YOLOv5-N (r6.1)	Faster R-CNN	Color-Based Blob Detection
Algorithm Type	Single-Stage Detector	Two-Stage Detector	Classical
Detection Method	CNN	Region Proposal Network and CNN	Pixel color thresholding and flood fill
Accuracy (mAP)	28.0%	> 70%	N/A
FLOPs (Billion)	4.5	~8-9	~0.002
Maximum Resolution	640x640	No maximum	No maximum
Computational Cost	Medium (Runs on CPU and GPU)	High (Runs mostly on GPU)	Low (Runs solely on CPU)
Model Size	7.2 MB	12.6MB (MobileNet)	Tens of Kilobytes
Training Requirement	Requires labeled dataset and training	Requires labeled dataset and training	No training required
Effect from external lighting / background	Low	Low	High
Implementation Complexity	Moderate	High	Simple

SOFTWARE – AI CAMERA COMPARISON

Feature	OAK-D	Arduino Nicla Vision	HuskyLens AI Machine Vision Sensor
AI Image Resolution	300x300, 416x416, 640x640	96x96, 128x128, 224x224	320x240
Depth Sensing	Yes	Yes	No
DFOV	81°	80°	70°
FPS	~20-30	~5-15	~30
AI Algorithms	Object detection, Facial recognition, Semantic segmentation, Pose estimation, Depth-based 3D localization, Gesture & hand tracking	Image classification, Simple object detection, Color or motion detection, Gesture recognition	Facial recognition, Object tracking, Color recognition, Line tracking, Object classification
Onboard Memory	8GB flash, 1GB RAM	2MB flash, 1MB RAM	8MB flash, 6MB RAM
Additional Peripherals	9-Axis IMU, Stereo depth perception,	6-Axis IMU, Time of Flight sensor, Microphone	2' IPS display, pre-programmed modes
Output	Image frames, depth maps, bounding boxes, labels, confidence scores, 3D coordinates, IMU data	Image frames, IMU data, bounding boxes, labels, confidence scores	Bounding boxes, labels, confidence scores
Price	$299	$80	$54

SOFTWARE – PROGRAM LANGUAGE COMPARISON

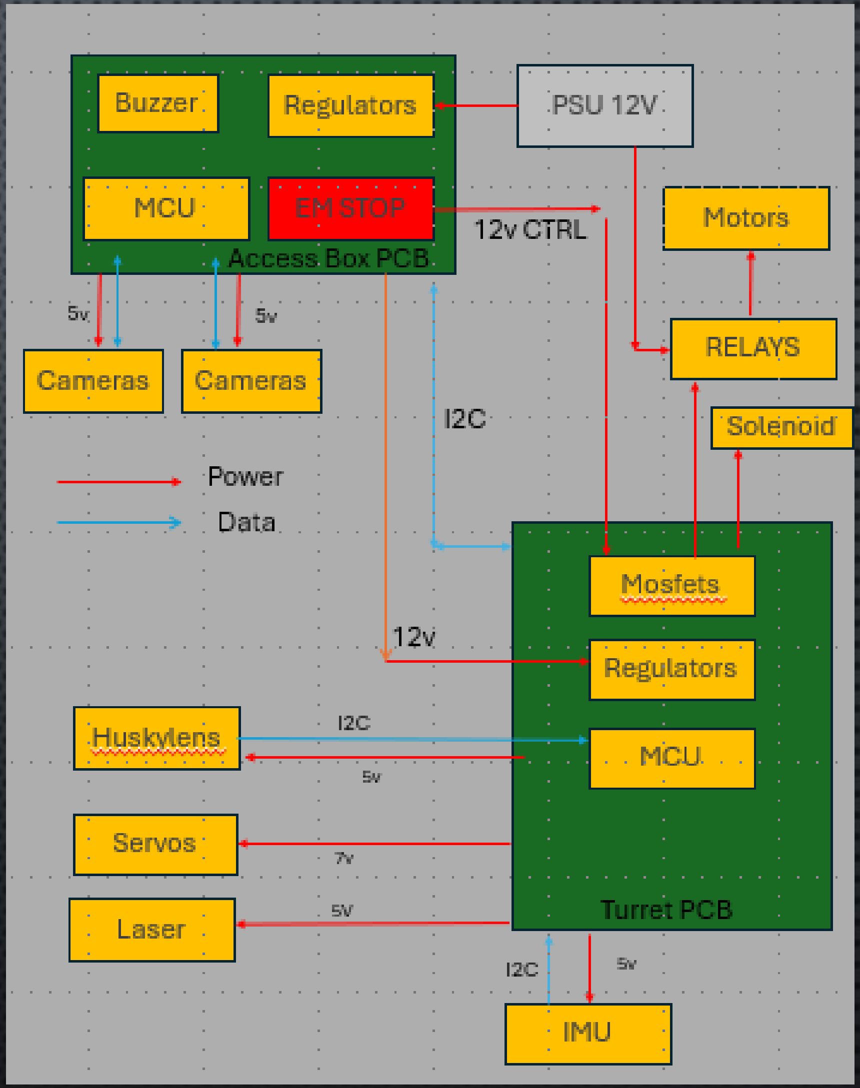
Feature	C	C++	Python
Speed	Highest	High	Low
Memory Usage	Lowest	Low	High
Hardware Control	Direct register level control	Direct register level control	Uses hardware APIs and libraries
Ease of Use	Low	Medium	High
Development Speed	Low	Medium	High

Software-Custom YOLO Model on HuskyLens 2

- S.T.A.T employs a custom YOLOv8 object detection model that is deployed onto the HuskyLens 2 (Kendryte K230 Processor, 6 TOPS KPU) to identify red balloon targets in real time. The model was developed to replace the HuskyLens's built-in color recognition algorithm, which produced significant amounts of false positives from other red objects (i.e., clothing or blanket). By incorporating both shape and color features, the YOLO-model approach significantly improves target detection across multiple environments.
- The model was trained on a 775-image dataset split 87/8/4% across train, validation, and test sets. The images were sourced from public red balloon datasets and from my own captured photos at various ranges (1-15ft) under multiple lighting conditions (e.g., direct sunlight, cloudy, inside under fluorescent lights) and backgrounds. The training was done in Google Colab using YOLOv8n architecture at 320x320 input resolution over 100 epochs, with data augmentation including horizontal flipping, brightness, exposure jitter, and up to 1px blur to simulate real deployment conditions. The trained model was exported to ONNX and converted to K230 KPU uint8 quantized format for on-device inference. The final model achieved a mAP of 0.95 and runs at 15-25 fps on the HuskyLens.

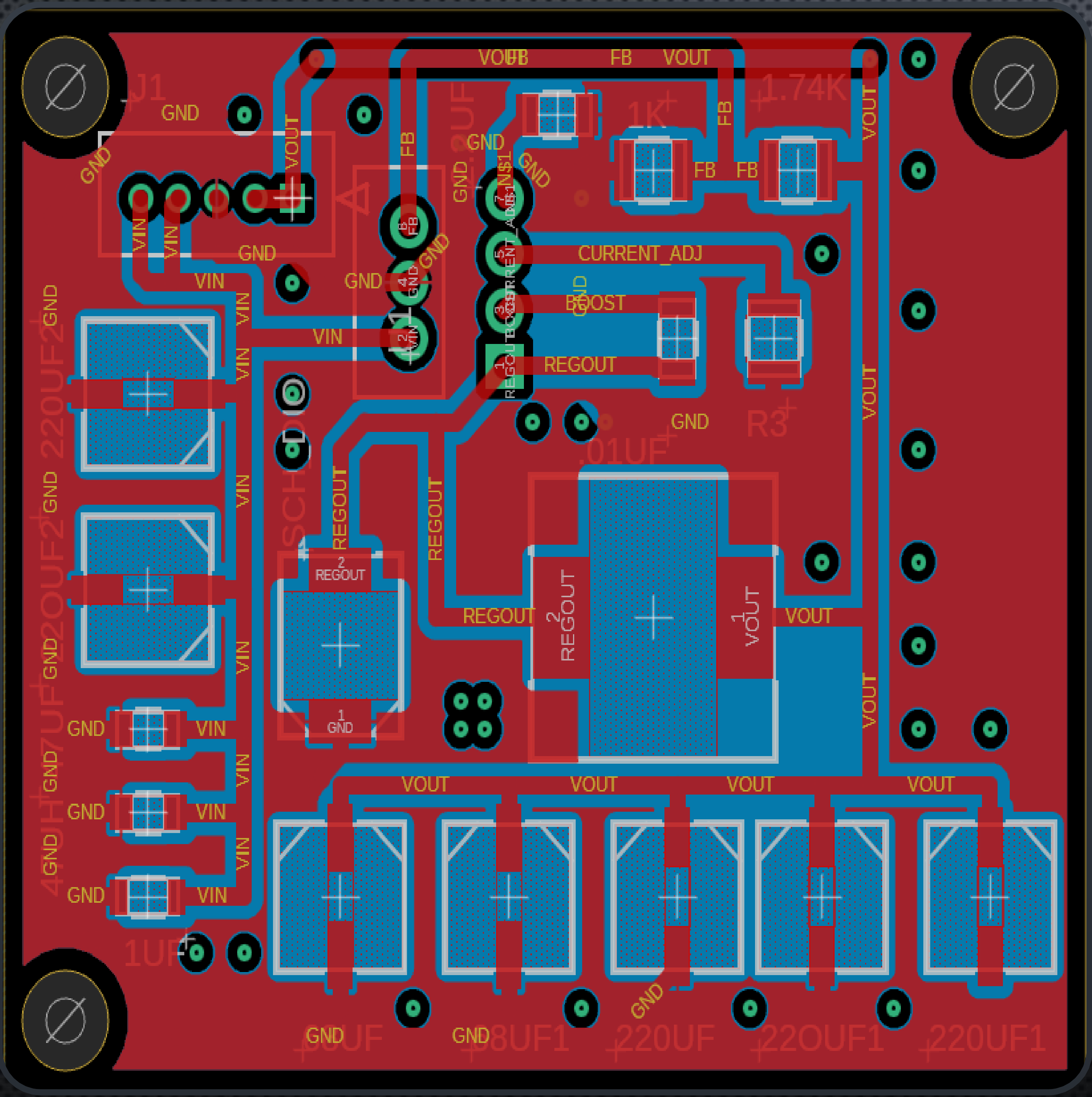
HARDWARE DIAGRAM

- Two-boards separating safety/control from actuation
- Hardware E-stop disables 12V actuation rail
- Dedicated regulators For Servos



POWER TABLE

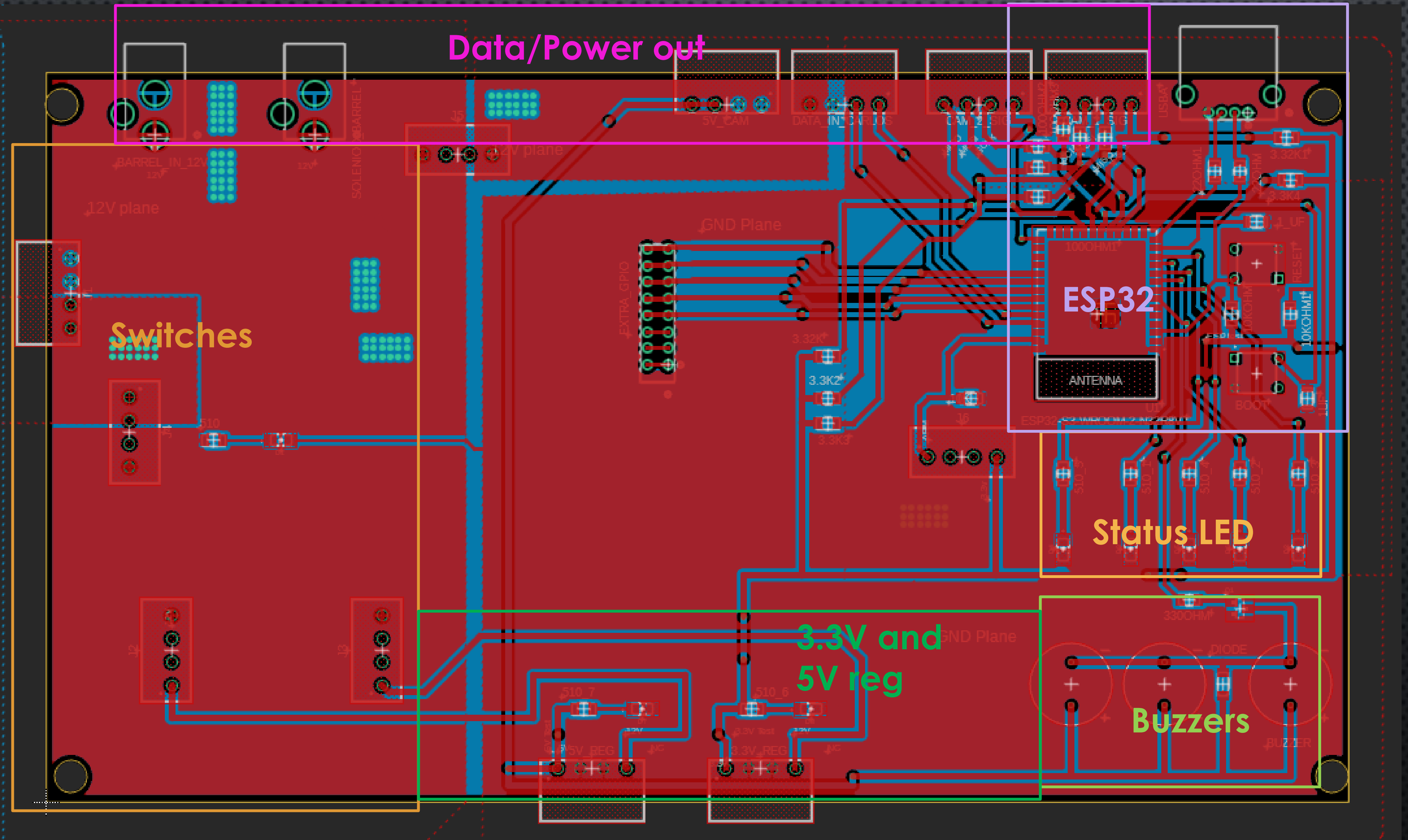
Component	Supply Voltage	Current	Quantity
Buzzer	5V	45mA	2x or 3x
LEDs	All Voltage	~20mA	5
IMU	3.3V	1.5mA	1x
Laser diode	5V	35mA	1x
HuskeyLens	5V	~320mA	1x
Cameras	5V	136mA	2x
ESP-32	3.3V	~100mA	2x
Servos	7.1V	~1.5A(Operating) ~3A (Stalled)	2x
Brushless DC Motors	12V	~2.4A (continuous)(Actual) ~45A on startup	2x
Power Usage normal operation = (~2A +2.3A*2)* 12.3V = ~85W			
Worst Case Total Power usage = ~587W			



PCB DESIGN - REGULATORS

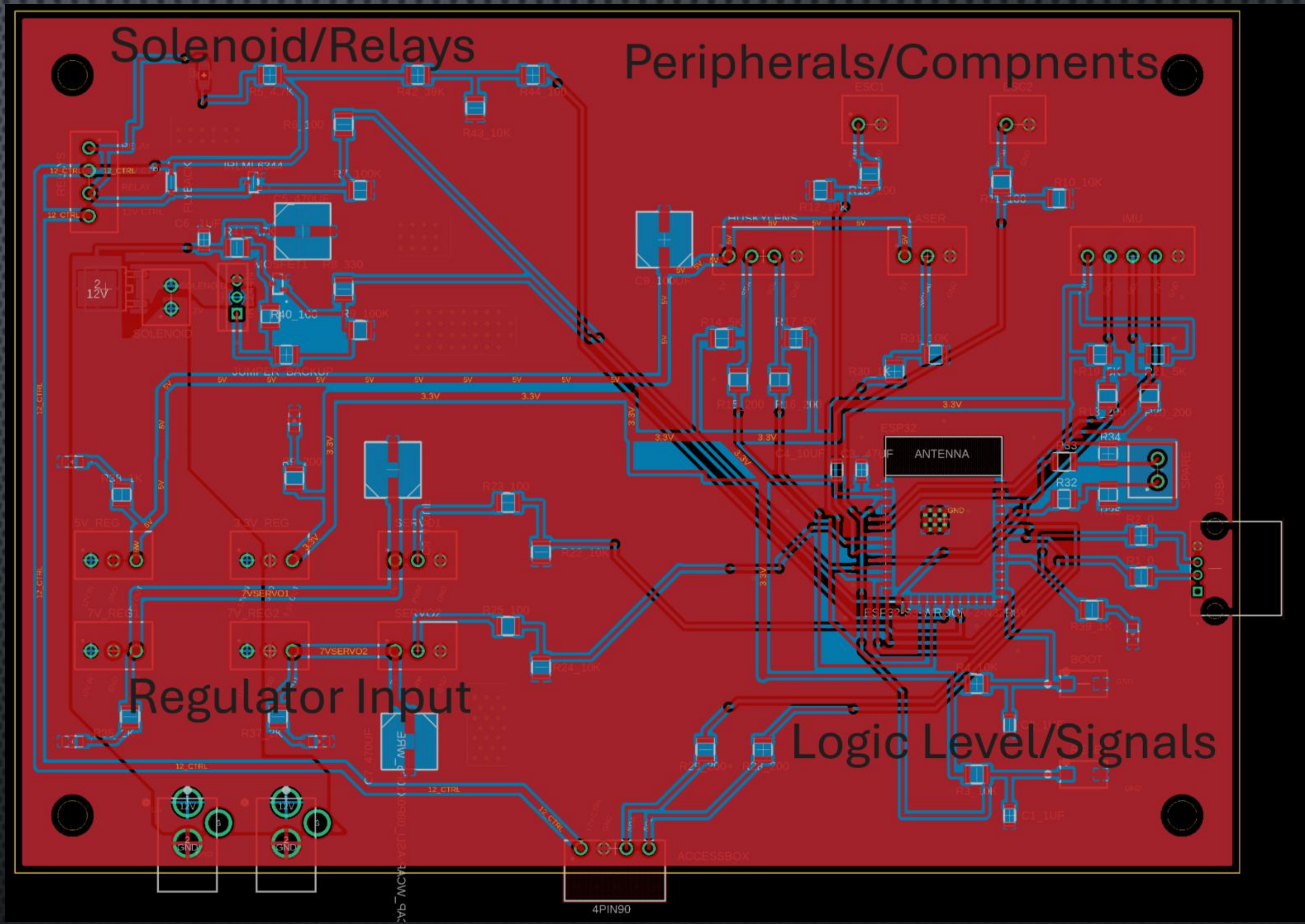
- CHANGE RESISTOR TO PRODUCE DIFFERENT OUTPUTS
- 5A REGULATOR BUT FINDING A SMALL 33uH INDUCTOR WAS DIFFICULT, ENDED UP WITH 3A INDUCTOR
- BULK CAPACITORS MEANT FOR LARGE NOISE REDUCTION
- 47uF AND 1uF MEANT TO HELP LOWER ESR TOTAL FROM BULK CAPS
- TIGHT REGOUT LOOP NEEDED
- SLIGHTLY HIGHER OUTPUT TO HELP WITH POWER LOSS

PCB DESIGN - ACCESS BOX

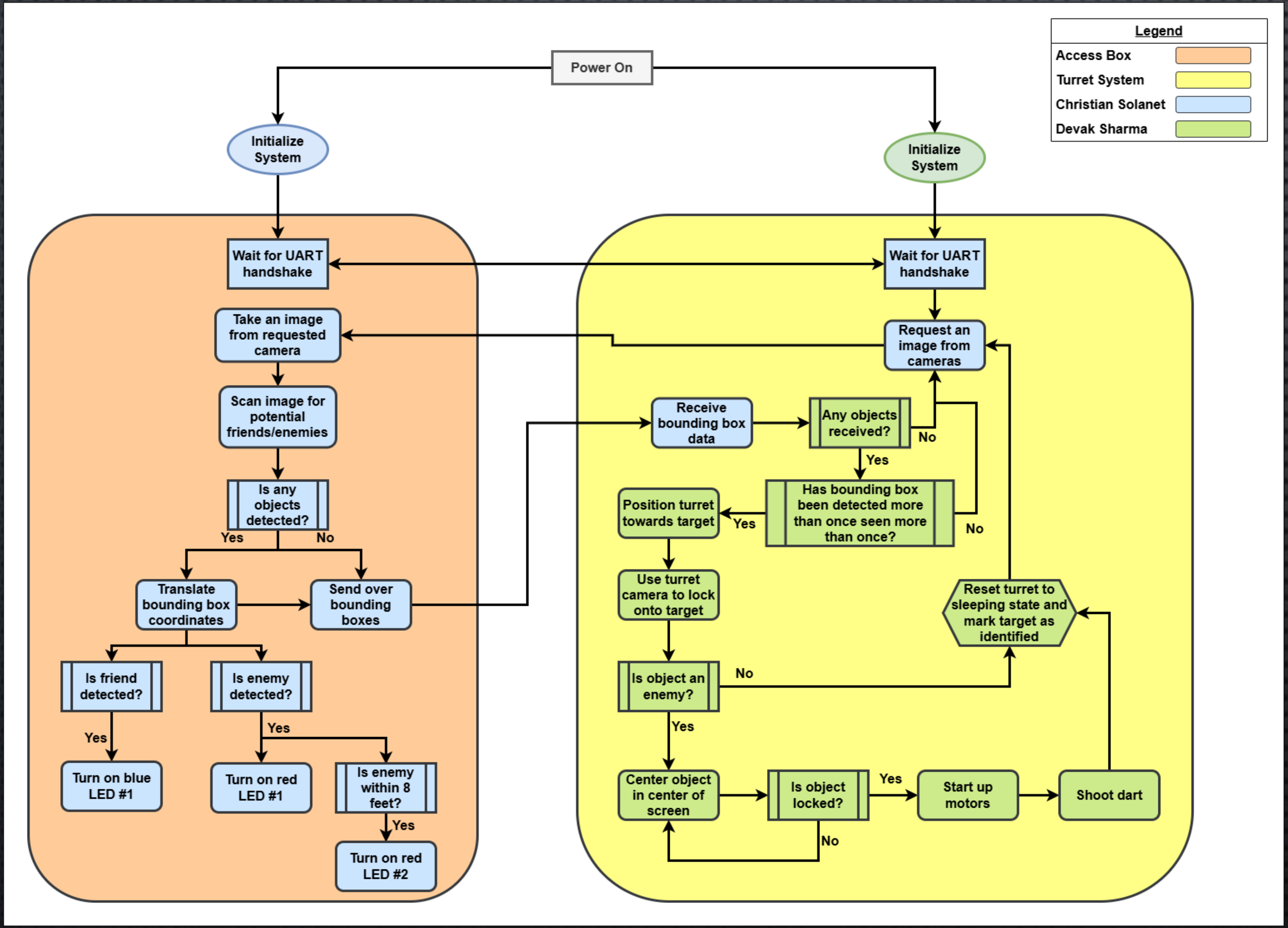


Turret PCB Design

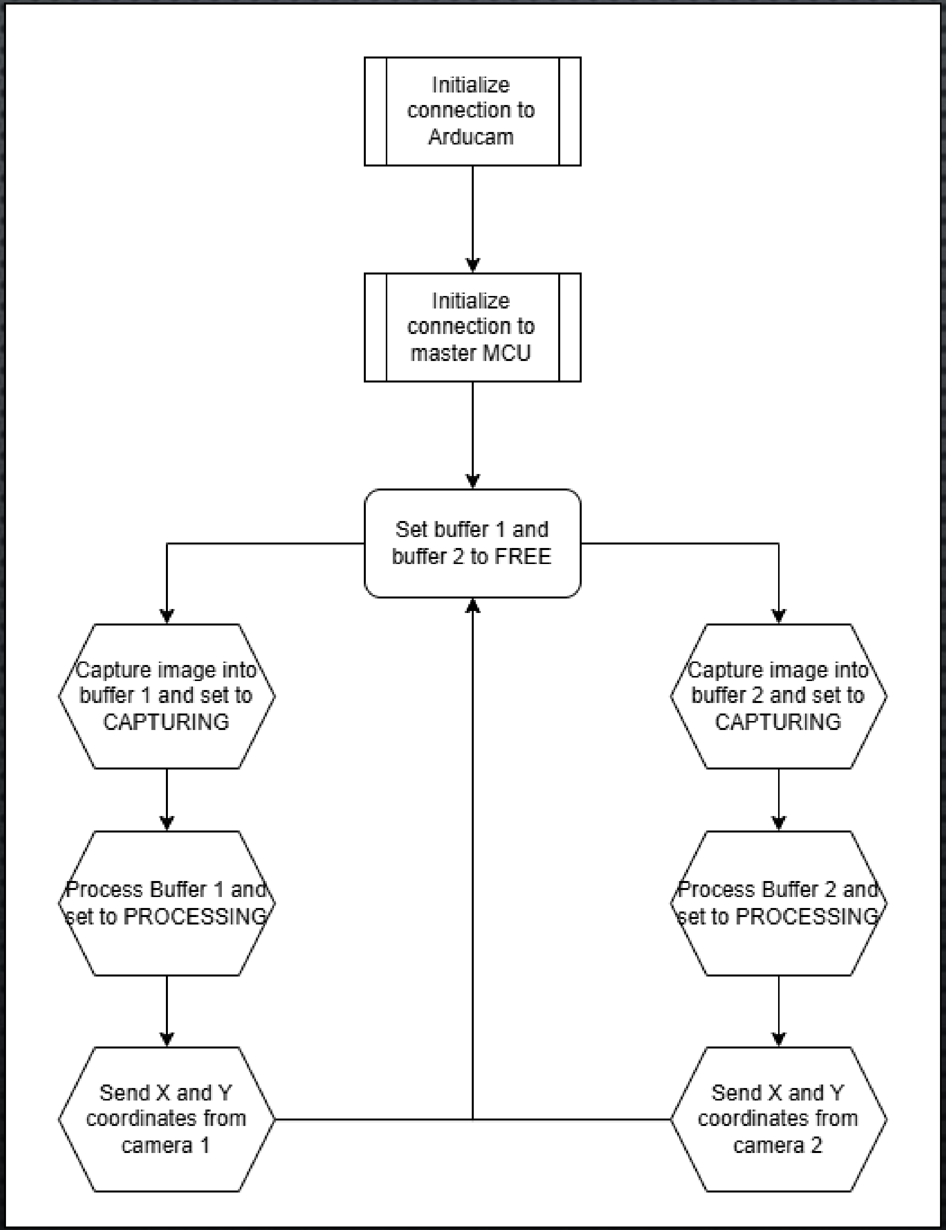
- • High-current motor phase wiring kept off-board; PCB carries only low-current control signals
- • MOSFET-driven inductive loads with flyback diode protection
- • Physical separation of high-current and low-signal routing regions
- • Decoupling capacitors placed at regulated rails and MCU supply pins

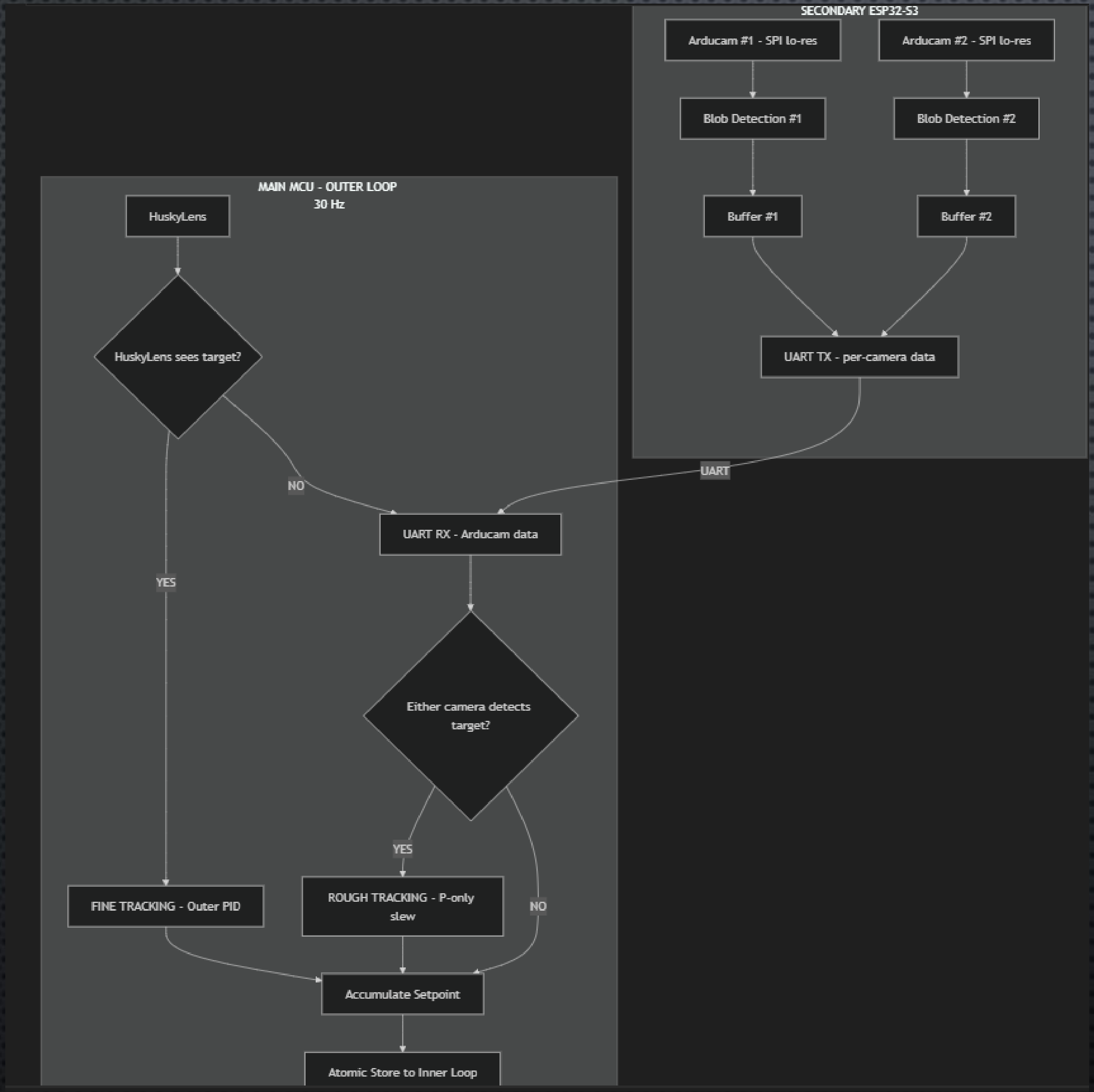


SOFTWARE – BASIC SYSTEM FLOW



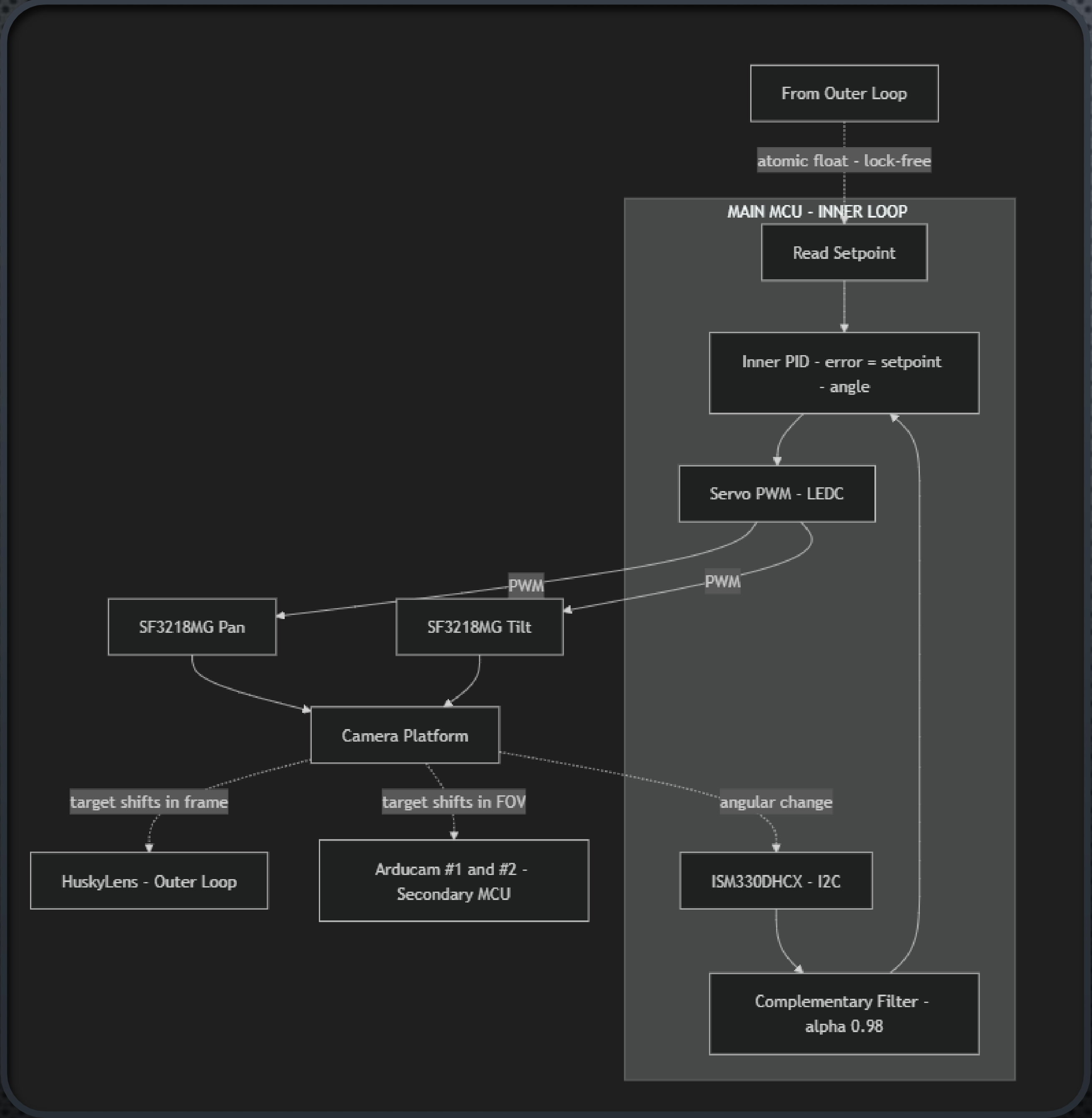
SOFTWARE – STATIONARY CAMERA CONTROL FLOW

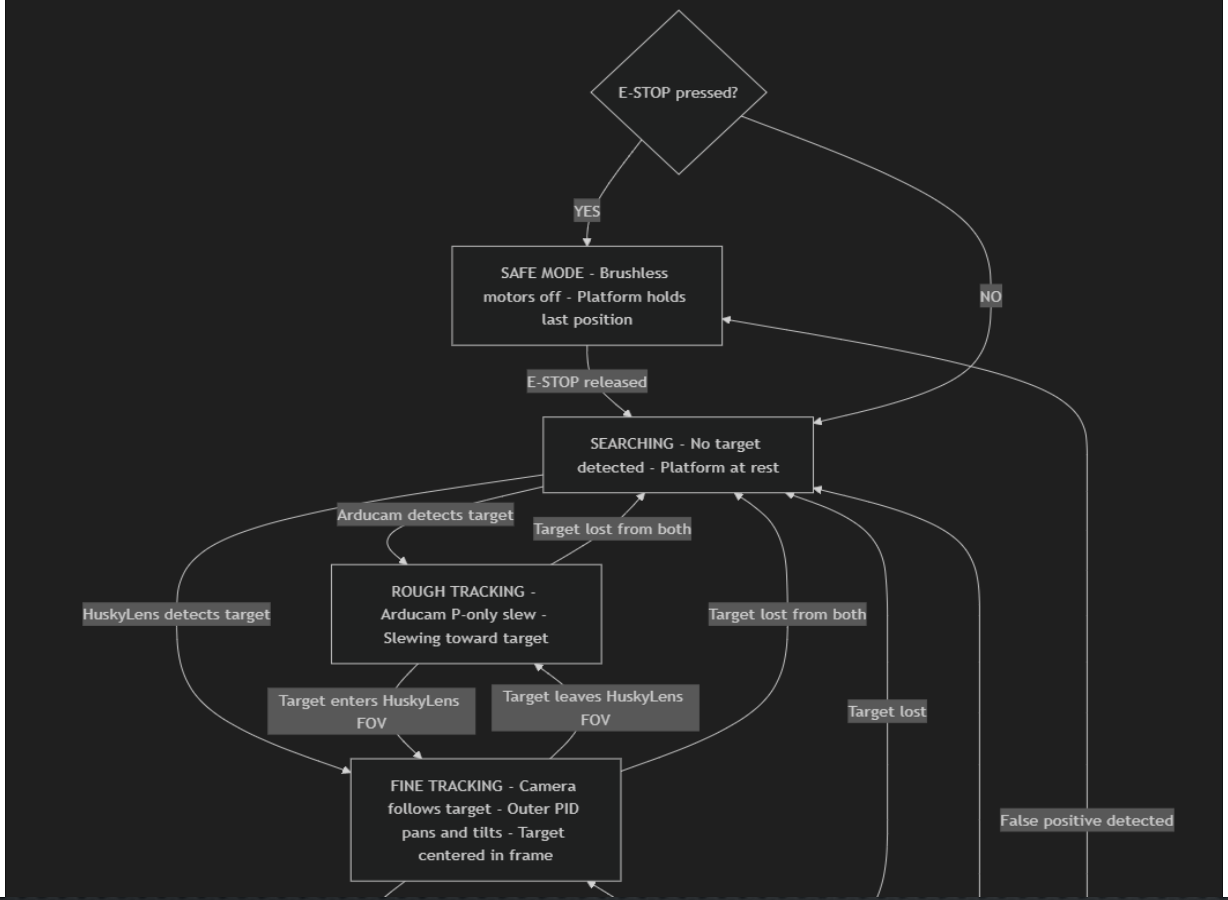


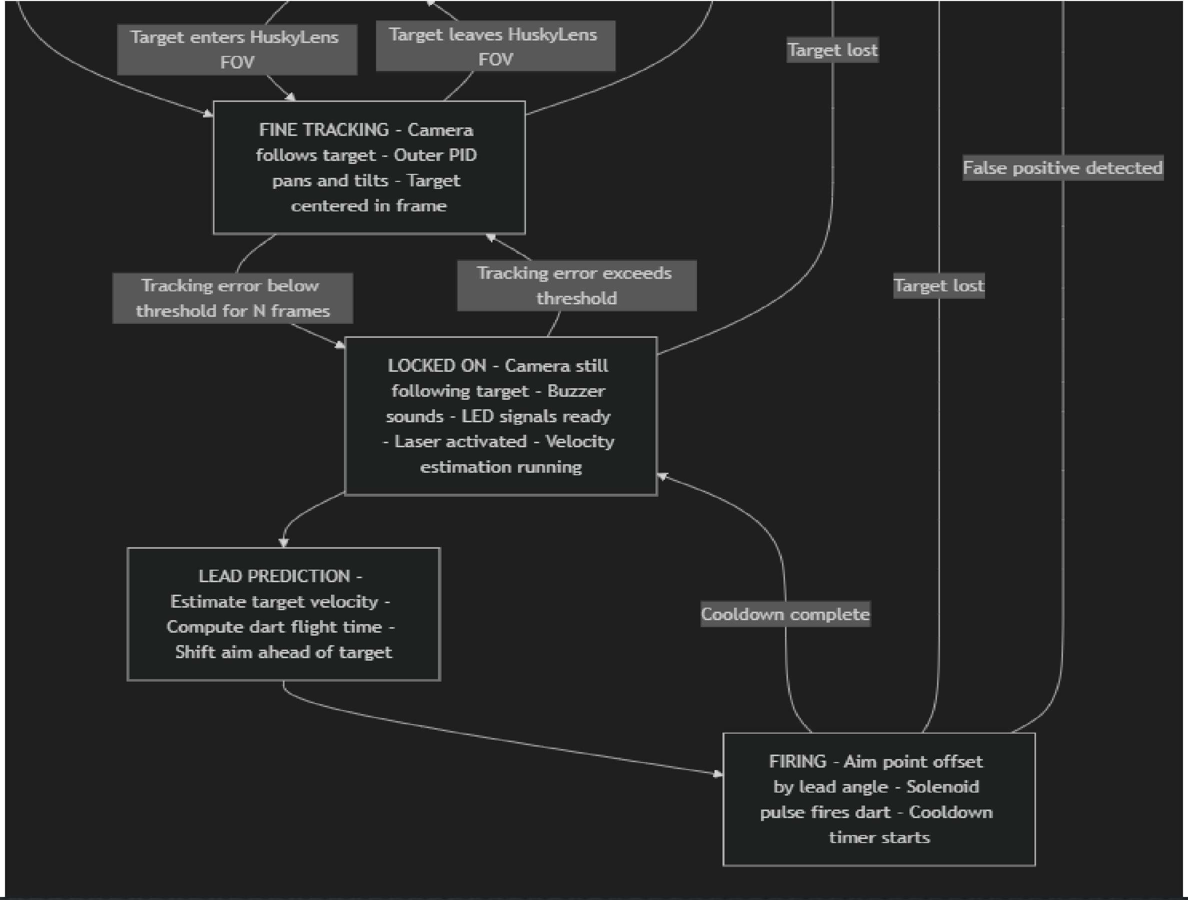


SOFTWARE – OUTER LOOP

SOFTWARE – INNER LOOP



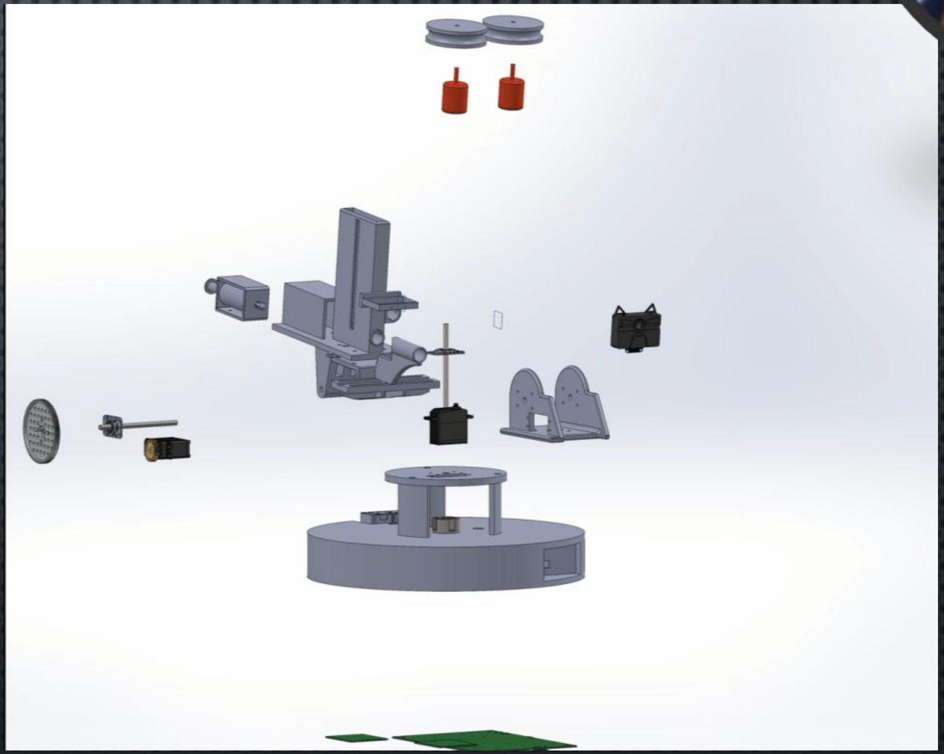




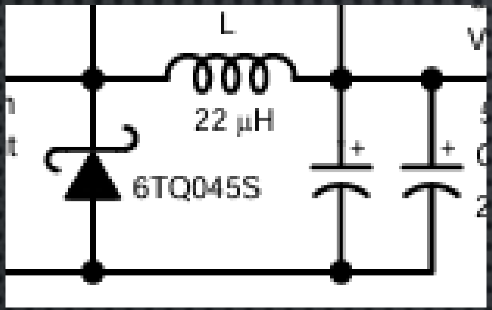
SOFTWARE – STATE MACHINE

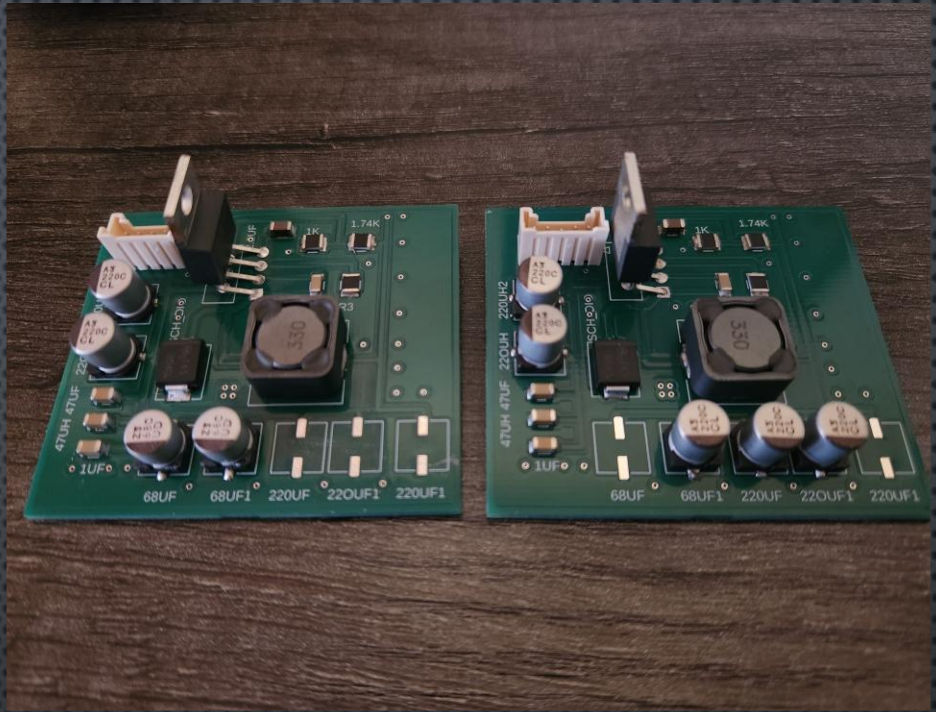
TURRET DESIGN

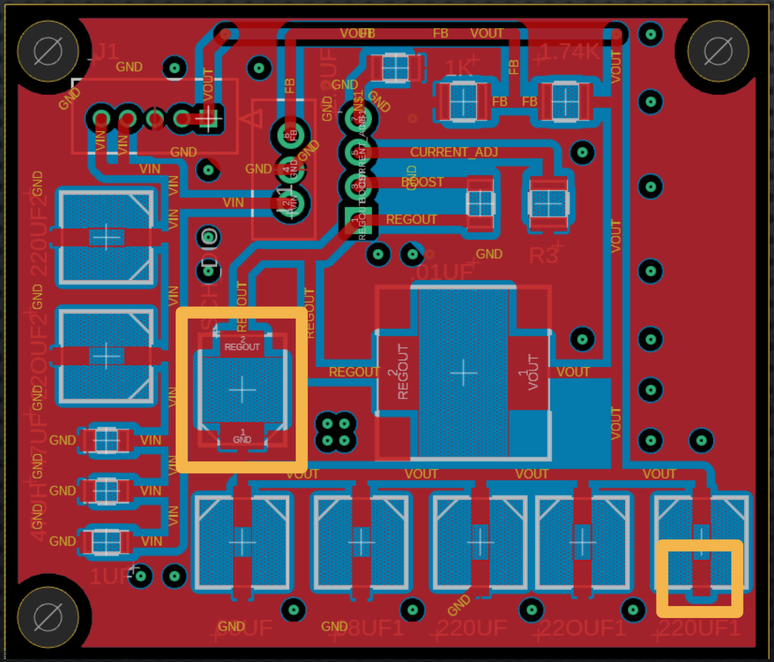
- DESIGN GOALS:
  - DUAL COUNTER-ROTATING FLYWHEELS ACCELERATE DART
  - ADJUSTABLE FLYWHEEL COMPRESSION
  - SOLENOID-DRIVEN DART INSERTION
  - GUIDED DART CHANNEL FOR ALIGNMENT
  - INTEGRATED MOUNTING STRUCTURE
  - 3:1 TILT GEAR FOR INCREASED TORQUE

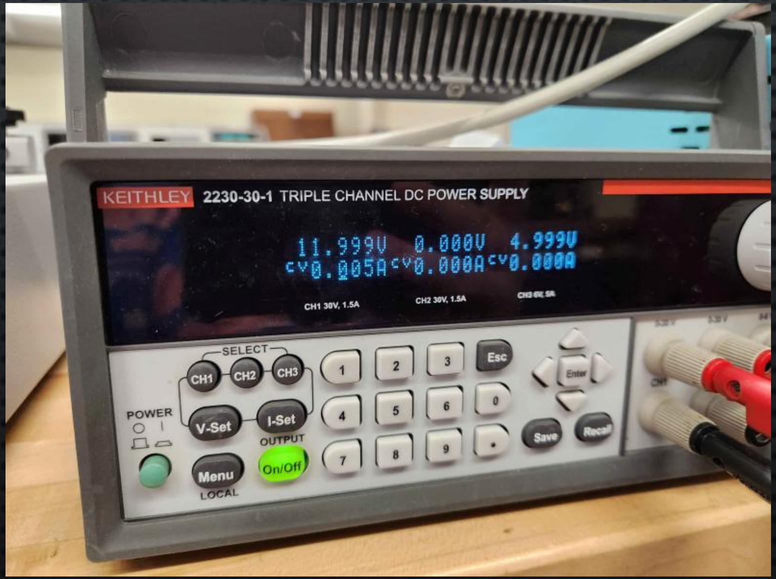


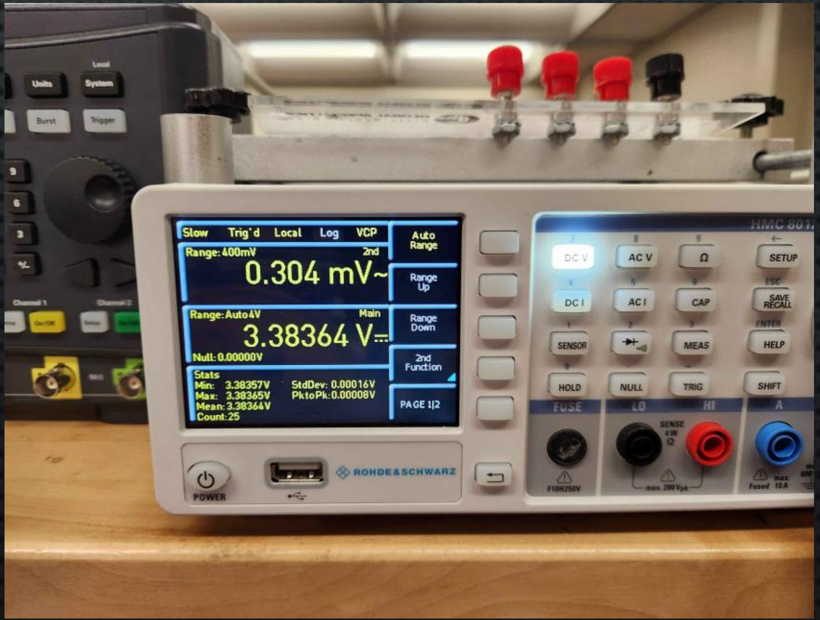
REGULATOR TESTING





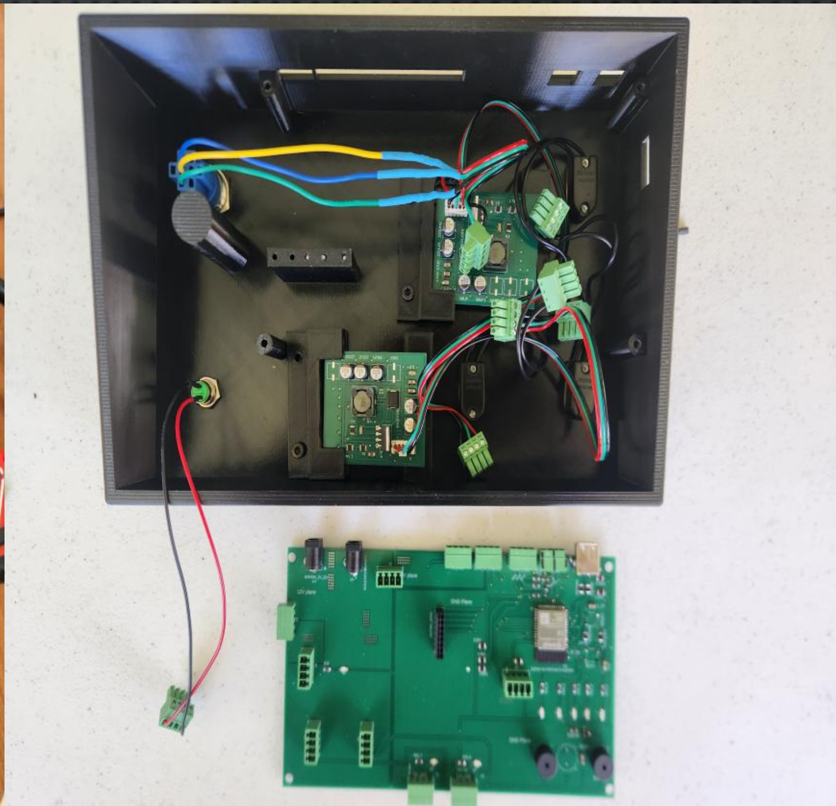






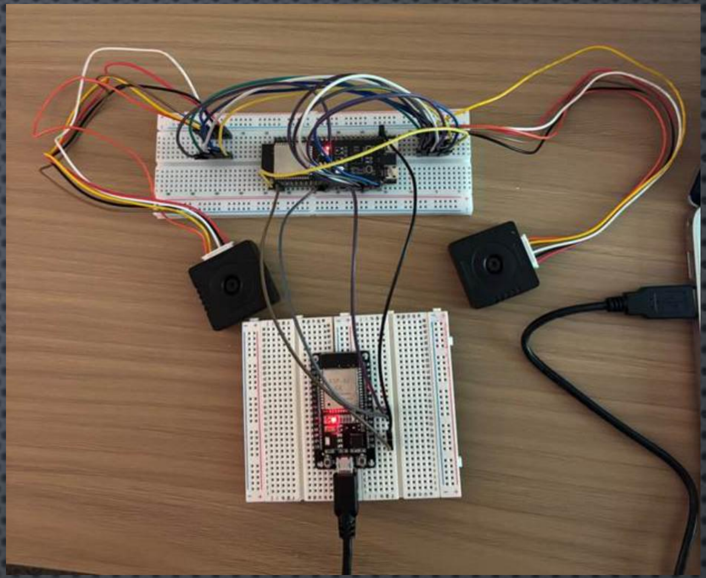
ACCESS BOX TESTING

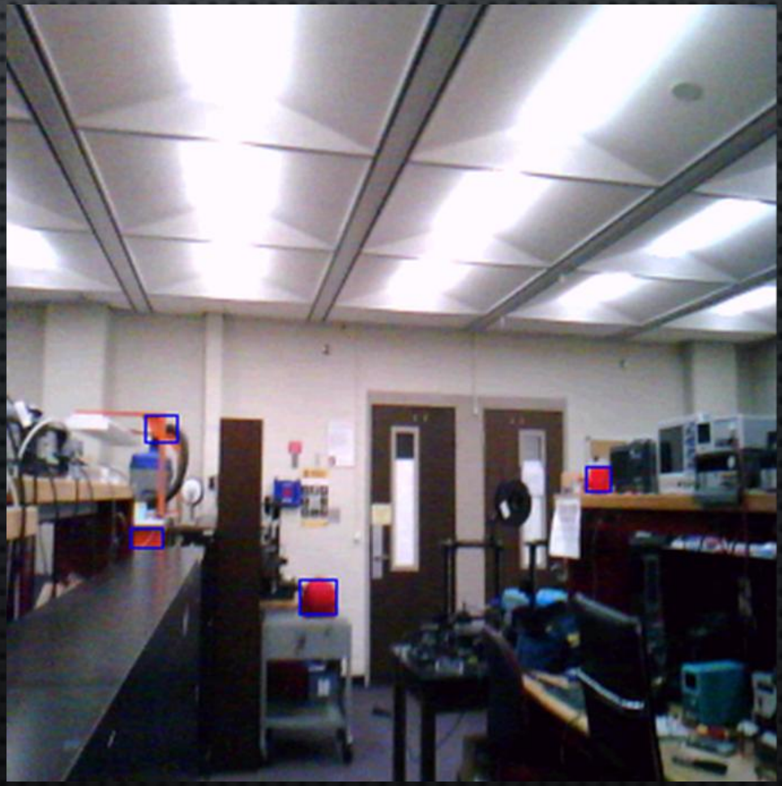
- Access box fully works
- Supplies power probperly to turret
- LEDs turn on for all status
- Emergency Button works as intended
- Arm key allows for tracking without shooting



SOFTWARE TESTING

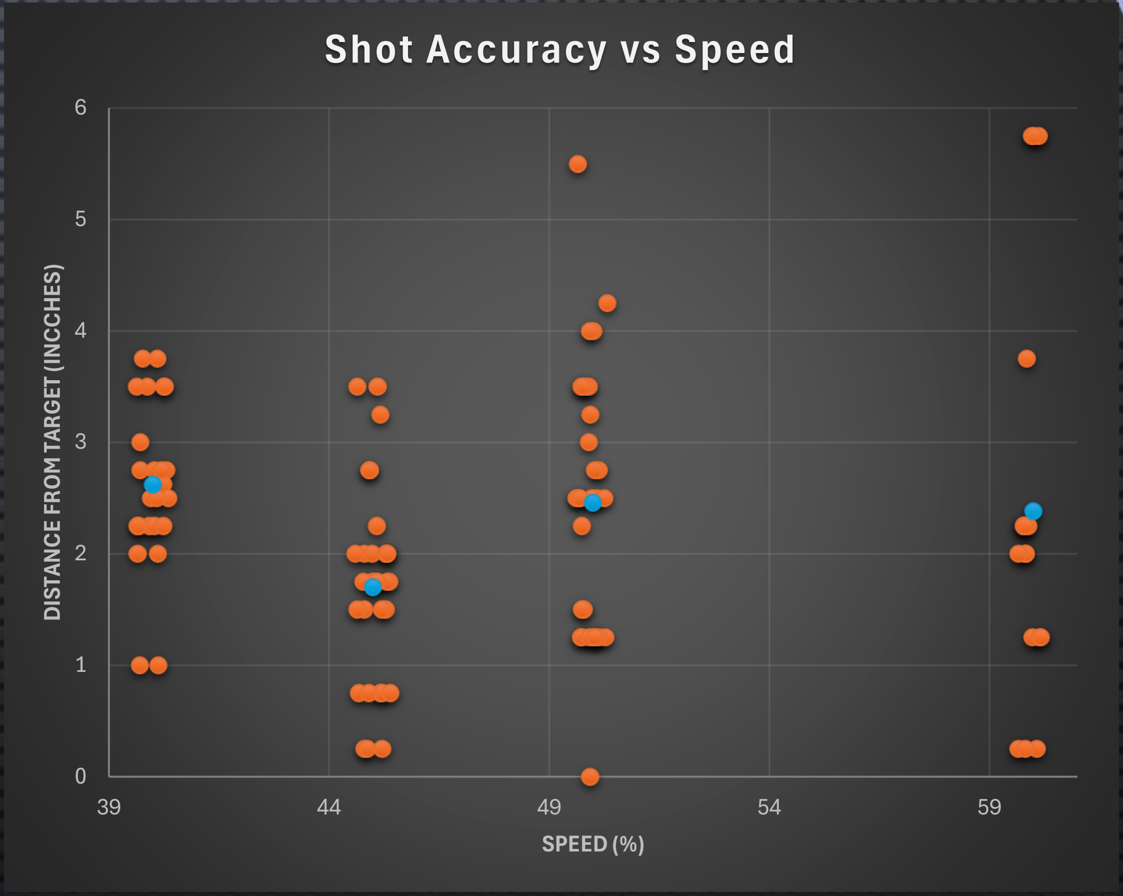
- To test the software, a simple python script was used to read the camera data as it was sent to main MCU.
- The stationary cameras successfully take photos and add bounding boxes around potential threat objects.
- Bounding box data is packaged and sent over to the main MCU.
- The total time from begin image capture to sending bounding box data at 320x320 resolution meets expectations.





BENCHMARK TURRET STATS

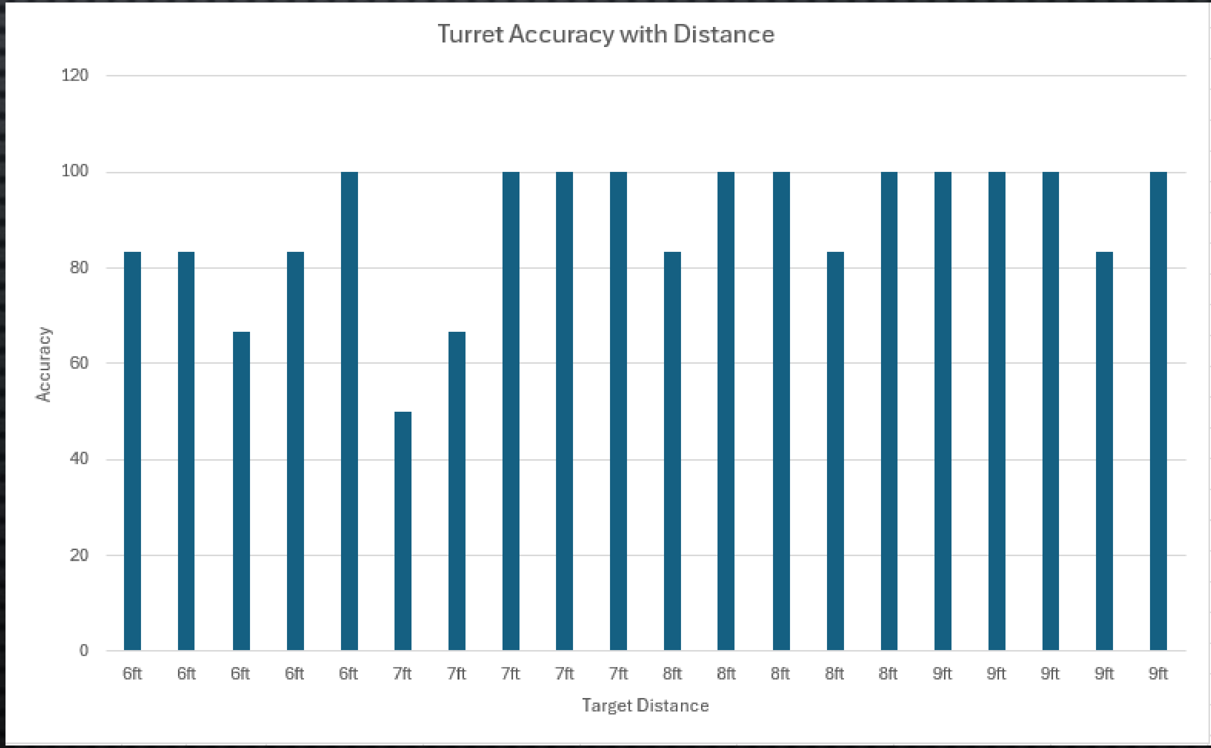
- DISTANCE FROM TARGET USED TO MEASURE ACCURACY
- EACH POINT REPRESENTS A SINGLE SHOT
- 45% SPEED SHOWS LOWEST AVERAGE ERROR
- HIGHER SPEEDS INTRODUCE GREATER SPREAD AND OUTLIERS
- INCREASED VIBRATION REDUCES CONSISTENCY AT HIGHER SPEEDS
- 45% SELECTED AS OPTIMAL BALANCE OF ACCURACY AND CONSISTENCY



TURRET ACCURACY TESTING

Distance	Hit	Miss	Lock time S1	Lock time S2	Lock time S3	Lock time S4	Lock time S5	Lock time S6	Average hit time	Accuracy
6ft	5	1	0.9	1.6	1.7	1	1	1.6	1.3	83.33333
6ft	5	1	1.7	1.7	1.6	1.7	1.5	0.98	1.53	83.33333
6ft	4	2	1.6	1	0.98	0.77	0.77	1.7	1.136666667	66.66667
6ft	5	1	1	0.98	1.5	1.9	0.98	1.6	1.326666667	83.33333
6ft	6	0	1.6	0.98	1.7	1.6	1.7	1.5	1.513333333	100
7ft	3	3	0.74	0.99	1.1	0.78	1.6	0.94	1.025	50
7ft	4	2	0.99	0.78	0.7	0.9	0.9	0.9	0.861666667	66.66667
7ft	6	0	2.5	2.1	1.3	1.4	1.9	1.8	1.833333333	100
7ft	6	0	2	1.3	1.8	1.8	1.3	1.3	1.583333333	100
7ft	6	0	1.4	1.4	2	1.7	1.7	1.3	1.583333333	100
8ft	5	1	1.3	2.2	2.1	1.3	2.1	1.3	1.716666667	83.33333
8ft	6	0	1.4	2.1	2.2	2.1	1.5	1.4	1.783333333	100
8ft	6	0	1.5	1.5	1.5	1.3	1.5	1.4	1.45	100
8ft	5	1	1.4	1.3	1.3	1.3	1.4	2.1	1.466666667	83.33333
8ft	6	0	2.2	2.1	2.1	1.3	1.3	1.3	1.716666667	100
9ft	6	0	1.3	2.15	2.2	2.4	2.3	2.2	2.091666667	100
9ft	6	0	1.3	2.25	2.2	2.3	1.4	1.3	1.791666667	100
9ft	6	0	1.4	1.8	2.2	1.9	2.4	1.8	1.916666667	100
9ft	5	1	1.3	2.1	1.27	1.3	2.1	1.4	1.578333333	83.33333
9ft	6	0	1.4	2.2	1.4	1.3	2.3	2.2	1.8	100

- Turret is suspectable to lighting conditions.
- Arducams could track but Huskeylens cant
- In Dark conditions may need a flashlight from phone shined on target to track



BUDGET

Subsystem	Part Name	Quantity	Price Per [$]	Total
Hardware	Microcontroller	1	$5.06	$5.06
Hardware	BLDC Motor	2	$24.99	$57.26
Hardware	High Torque Servo	3	$17.99	$53.97
Hardware	Electronic Speed Controller	2	$24.00	$48.00
All	IMU	1	$12.95	$12.95
Hardware	LEDs	5	$0.20	$1.00
Hardware	Buzzer	1	$0.54	$0.54
Hardware	Laser Diode	1	$18.95	$18.95
Hardware	SG90 Servo	1	$3.62	$3.62
Hardware	PSU (Mean-Well 600 12v)	1	$80.60	$80.60
Hardware	Emergency Button	1	$5.90	$5.90
Hardware	5V Regulator	1	$0.44	$0.44
Hardware	Mechanical Misc (bearings,springs)	1	$131.00	$131.00
Hardware	3.3V Regulator	1	$0.44	$0.44
Hardware	PLA Filament	1	$17.99	$17.99
Software	Husky Lens	1	$54.90	$54.90
Software	Cameras (OV5640)	2	$11.99	$24
Hardware	PCBs	3		$150.00
All	20% Cost Buffer	1	$120.15	$120.15
Total				**$1020.85**

WORK DISTRIBUTION

Engineers	Responsibilities
Electrical	
Nicholas Martucci	• PCB design of power system and access box controls • Modeling/building access box • Administrative work • Project lead
Carlos Garcia	• PCB design of turret • Modeling of 3D printed turret • 3D printing turret and mounts for cameras, sensors, laser, and solenoid
Computer	
Christian Solanet	• Software design for secondary MCU • Blob Detection Algorithm • Implementing two Arducam 3MP SPI cameras for rough tracking • Implementation of peripherals on the access box
Devak Sharma	• Software design for primary MCU • Implementation of double-axis PID controller and HuskyLens AI Vision Sensor with IMU for stable tracking • Tracking algorithm • Motor control

PLAN FOR THE FUTURE

- Use rasperry pi instead of ESP32
- Use stronger Cameras
- Use better AI tracker and upload more models
- Decrease wire usage and make cleaner look
- Use stepper motors rather then servos
- Move to a Moving target