

BRON-S:
BLUETOOTH
REMOTE
OPERATED
NERF-SENTRY

GROUP 24

- JUAN LANDRON (C.P.E)
- ROSS ELLIS (C.P.E)
- DAANISH HABIBI (C.P.E)
- KYLE QUARTI (E.E)



PROJECT GOALS AND INTRODUCTION

- Wanted to design and build a mounted turret that would automatically detect hostile targets.
- As an alternative to the fully autonomous target detection and firing, we wanted to allow the user to manually aim and fire the turret if so desired.
- Focus on portability and ease of use.
 - Compact design
 - Low power
 - Wireless Connectivity



INSPIRATION AND INFLUENCES

- Grew up with NERF toys and wanted to create an automated NERF blaster.
- Wanted to add OpenCV experience to our resumes.



DESIGN SPECIFICATIONS



WIRELESS RANGE
OF ~10M



80% ACCURACY
WITHIN 5M



TARGET DETECTION
WITHIN 5M



CAN IDENTIFY A
TARGET IN UNDER
5 SECONDS



POWER SYSTEM DESIGN

- We are utilizing two separate power supplies for our project
 - Lithium Ion battery bank with fast charging capabilities
 - 10000mAh
 - 5v/4.8A rated output
 - Can supply power to Pixycam and Raspberry Pi
 - High current RC battery
 - 2100mAh
 - 7.2V/20C charge/discharge rate
 - Much higher power draw needed for the servos



CUSTOM PCB MCU SPECIFICATIONS

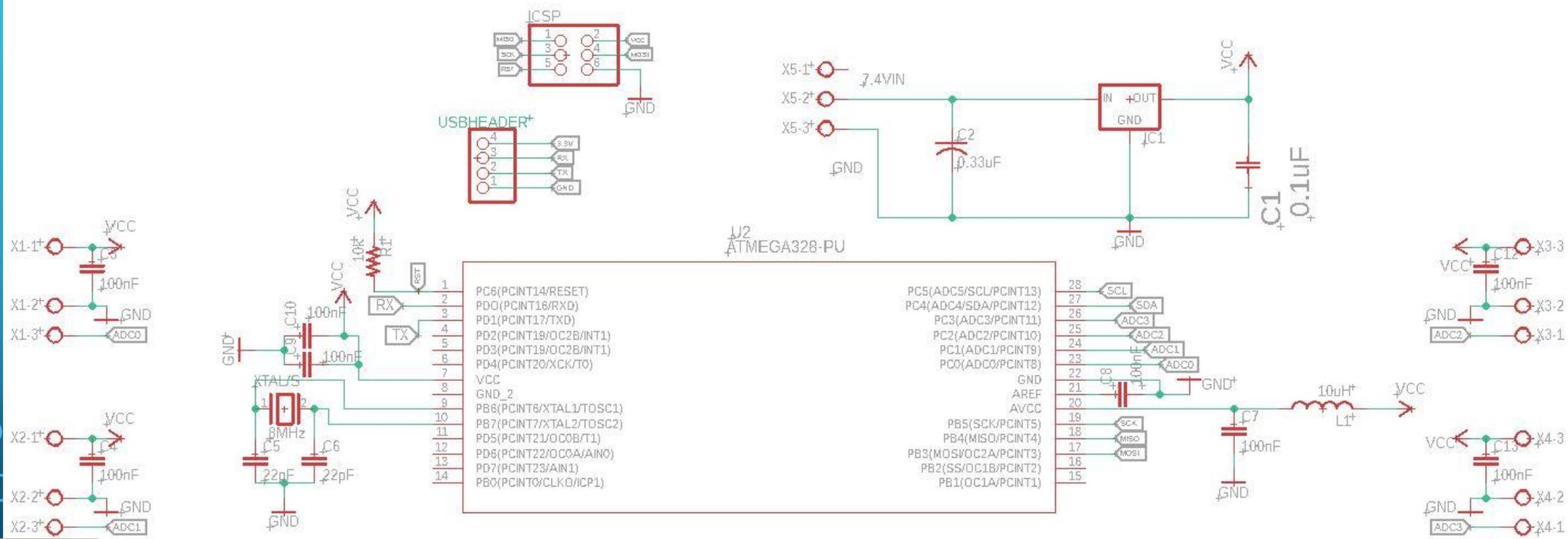
- Based on an Arduino Uno R3 reference design
- Using the Atmega328 because it was easy to obtain, and inexpensive
- Operating voltage of the Atmega328p is the same as our other components so there is no need for additional voltage regulation



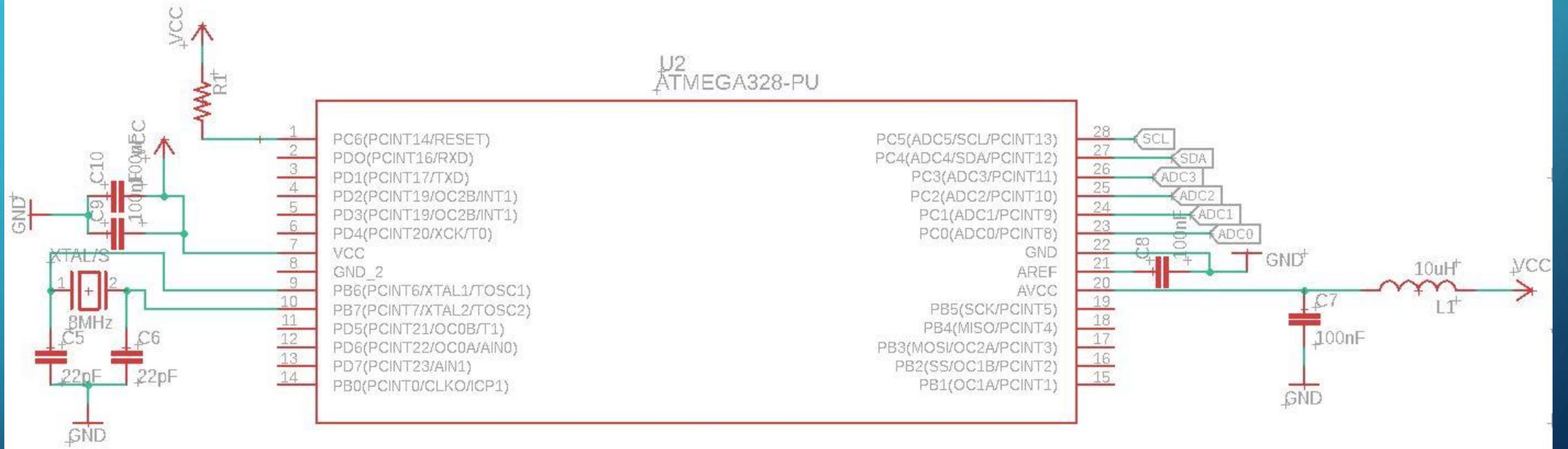
Microcontrollers	MSP432P401R	Atmega2560	Atmega328
Operating Voltage Range	1.62-3.7V	1.8-5.5V	2.7-5.5V
PWM Channels	20	12	6
UART/USART	4	4	1
SPI	8	1	1
Flash Memory	256KB	256KB	32KB
SRAM	64KB	8KB	2KB
ROM/EEPROM	32KB(ROM)	4KB(EEPROM)	1KB(EEPROM)
Maximum Clock Speed	48MHz	16MHz	16MHz



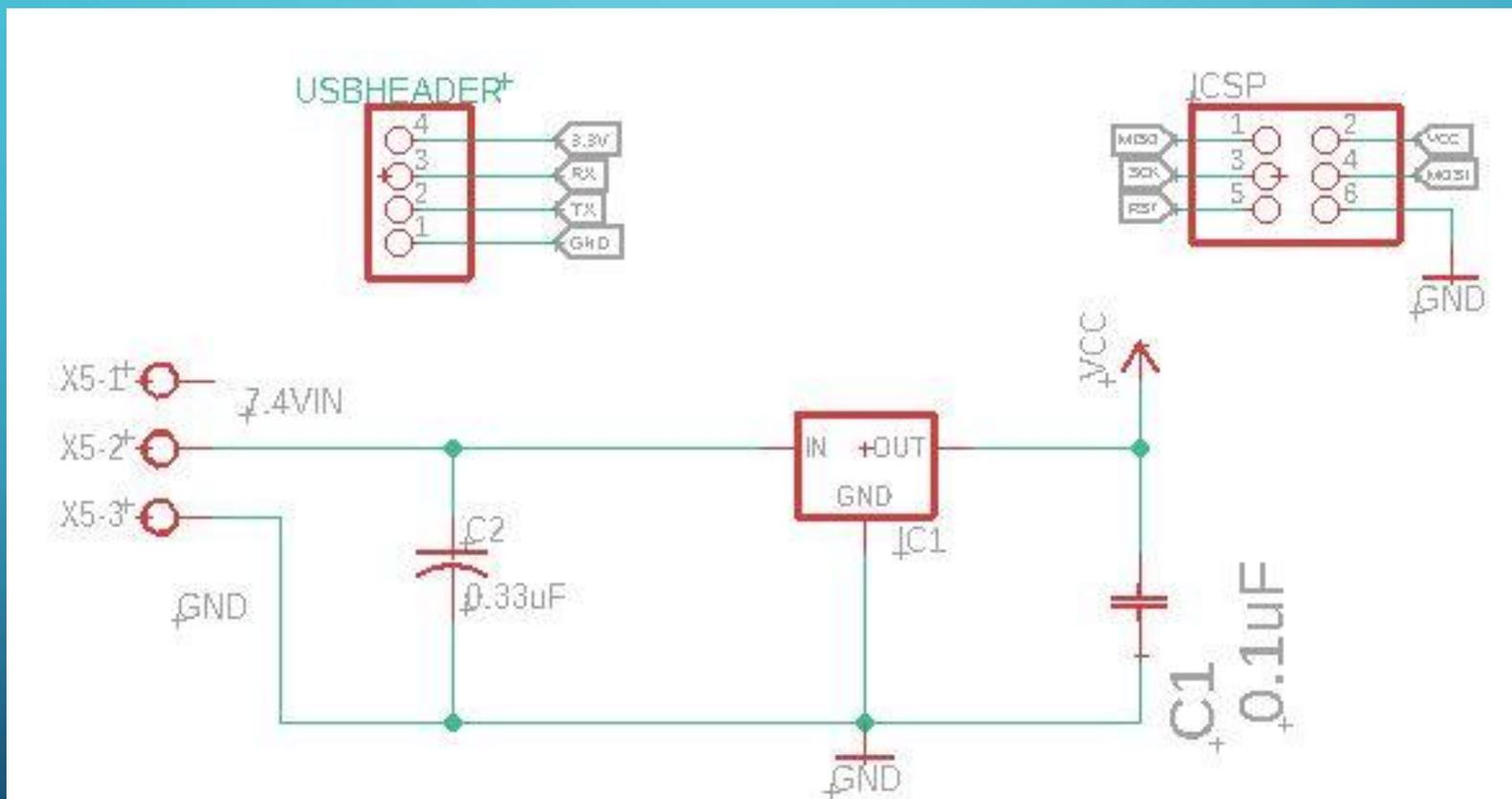
CUSTOM PCB SCHEMATIC



CUSTOM PCB: MCU



CUSTOM PCB: VOLTAGE REGULATOR



MOTORS AND SENSORS

- The accuracy of the Nerf gun limited the range at which we could engage targets.
- We decided to implement an ultrasonic sensor to act more as a proximity sensor.



Distance Sensors	Approximate Range	3D Mapping	Complex Structure Mapping
Ultrasonic Sensor	2cm - 500cm	no	no
IR sensor	100cm - 500cm	no	yes
LIDAR	Around 10m	Yes (if using multiple)	yes
LED Time-of-Flight	Around 4m	yes	yes



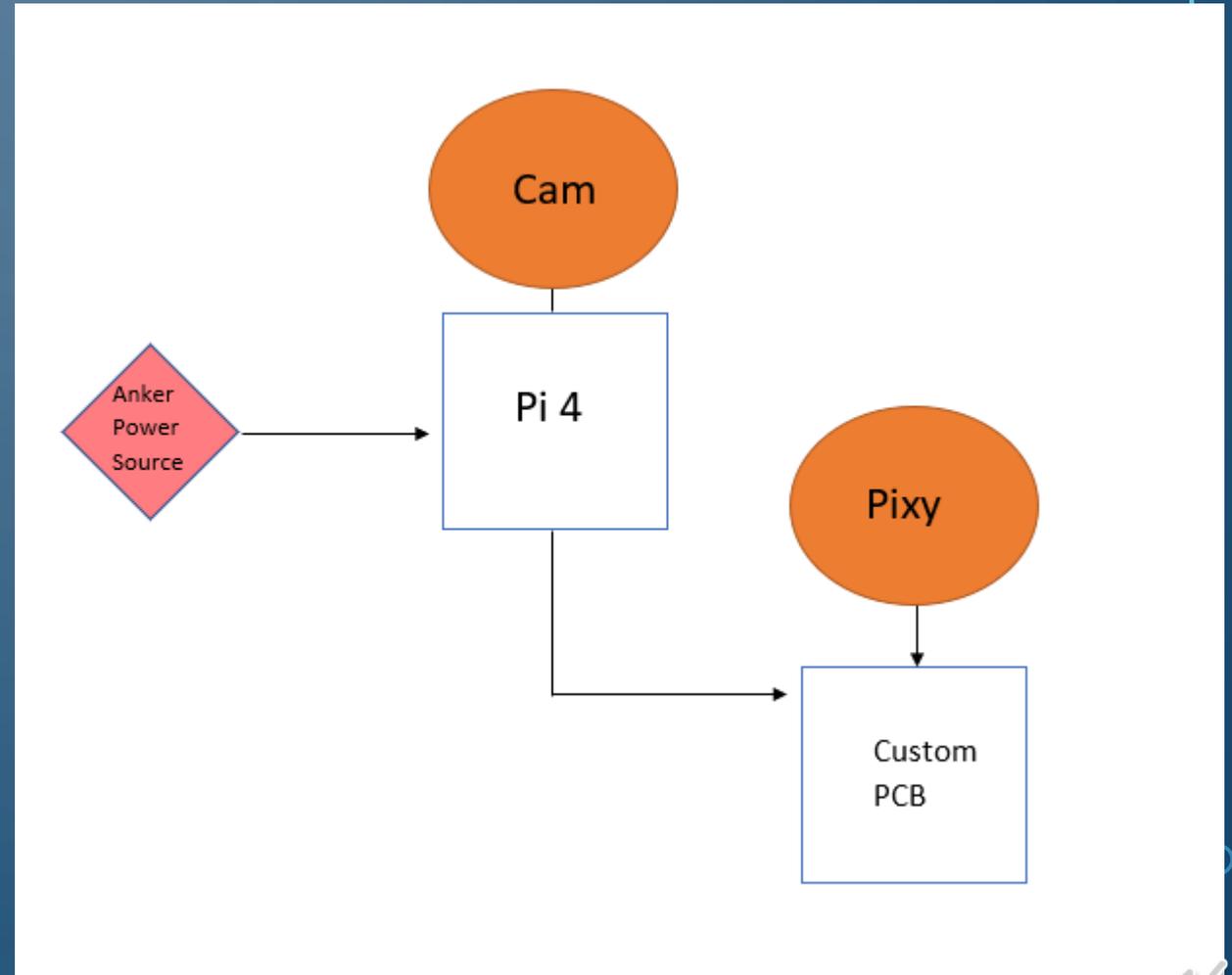
SERVO MOTORS

- Y-axis servo is a LJWRC 25Kg servo
 - Rated at 28Kg.cm at 6.4V or 23Kg.cm at 5V
 - Stall current of 2.67A
 - Extra stall torque was needed because of increased inertia on y-axis
- X-axis servo is a MG996R
 - Rated at 12Kg.cm at 6V and 10Kg.cm at 5V
 - Stall current is 2.5A



PROJECT SYSTEMS: SUBSYSTEM

- Subsystem handles all image processing data
- Two cameras being used:
 - Pixycam 2.0:
 - Customizable color detection
 - Webcam:
 - OpenCV facial recognition algorithm



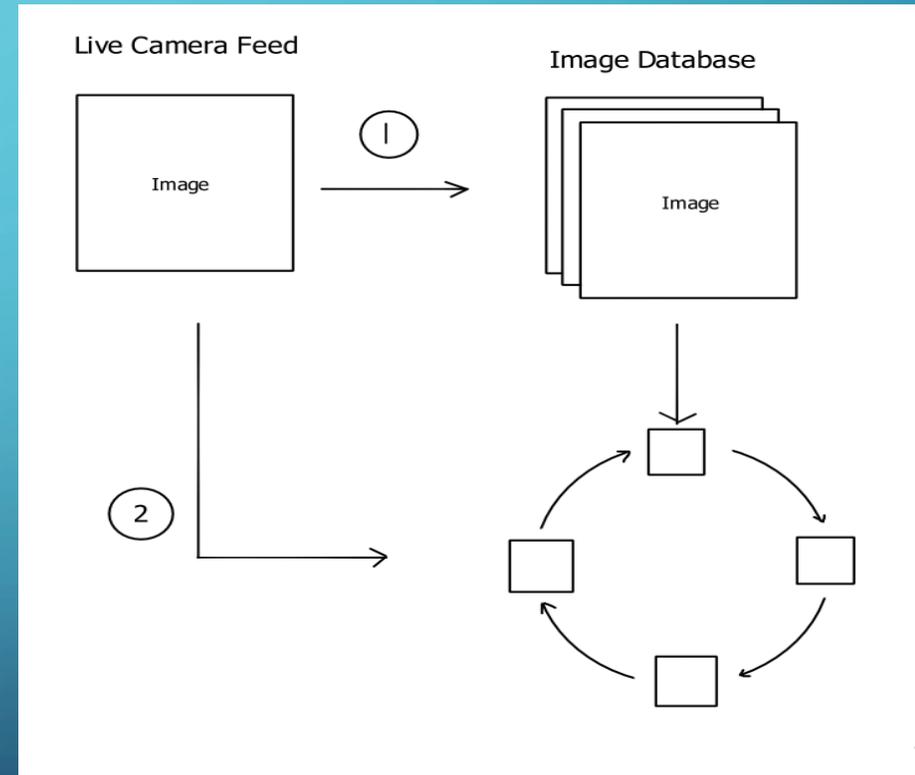
SECONDARY MCU SPECIFICATIONS

- *Why we chose the Raspberry pi4(2gb)*
 - *Offers a myriad of support*
 - *Can be programmed in essentially any language*
 - *Power draw compared to performance*
 - *Raspbian OS*
 - *USB ports offer easy communication between each of our devices*
 - *Has built in Bluetooth and wifi*



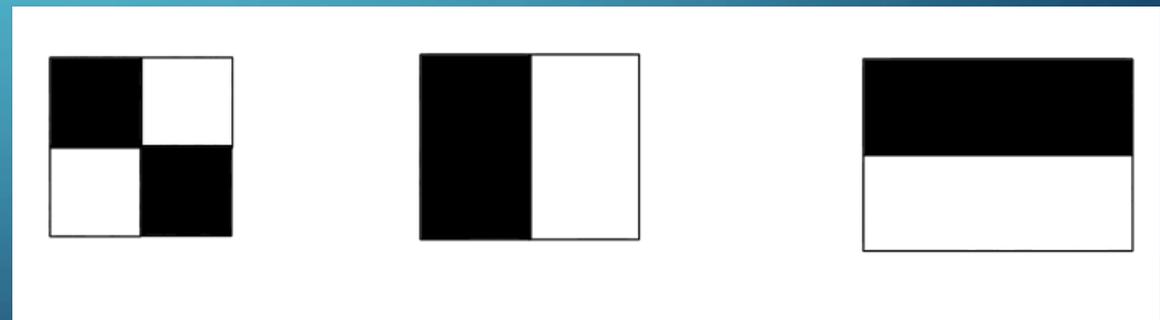
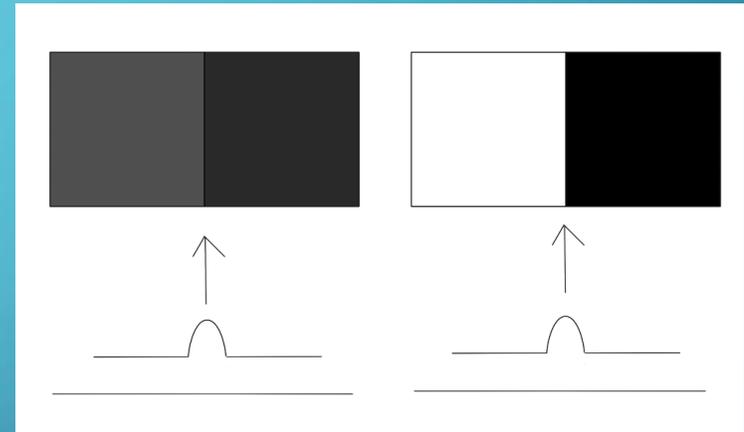
COMPUTER VISION

- OpenCV
- Coding in python
- *What limitations might we have*
 - Memory
 - Speed of tracking



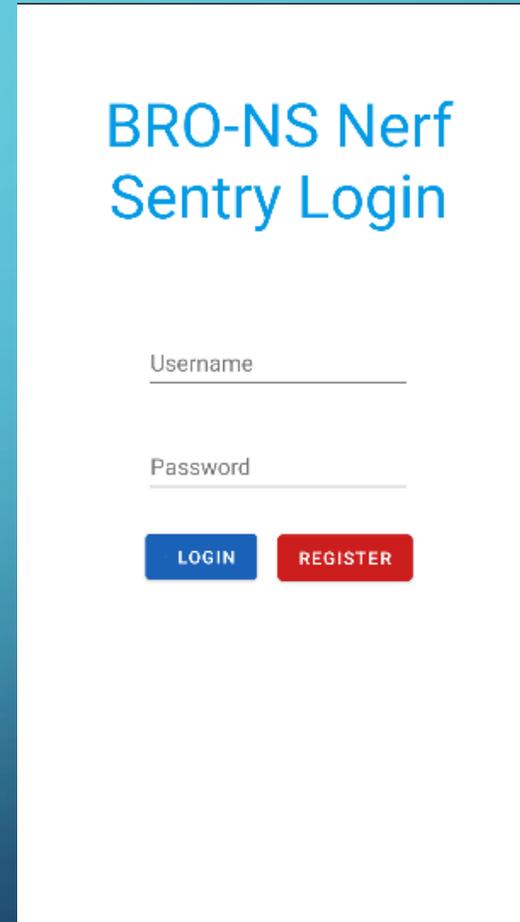
COMPUTER VISION

- *The image database has trained models*
- *Haar cascade*
- *Each face has a unique outline which is what the algorithm uses.*



ANDROID APPLICATION DEVELOPMENT

- Used Android Studios
- Programmed in Kotlin, interoperable with Java
- Reasons for choosing Android Studios:
 - UI creation
 - Access to android devices
 - Compatibility with Google Firebase
- Functionality:
 - Login and Sign out
 - Choice of mode of operation
 - Manual
 - Autonomous

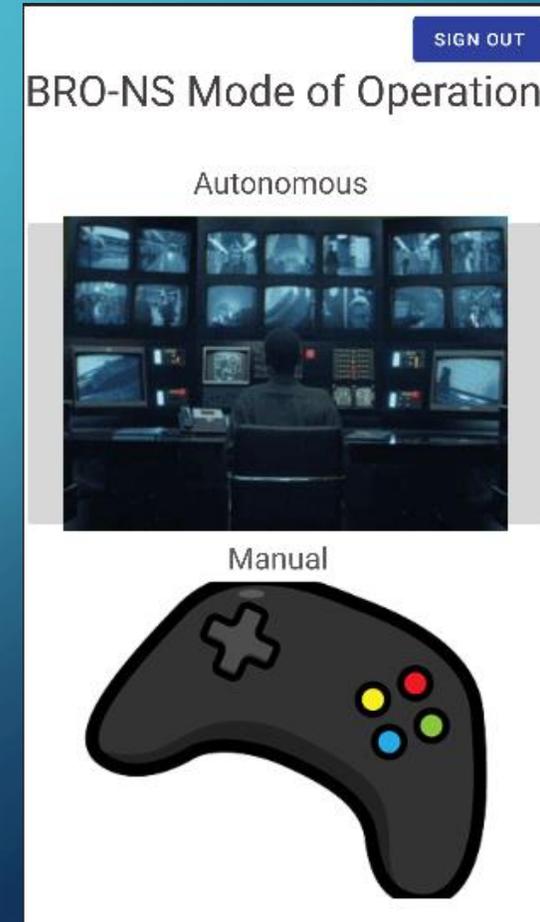


BRO-NS Nerf Sentry Login

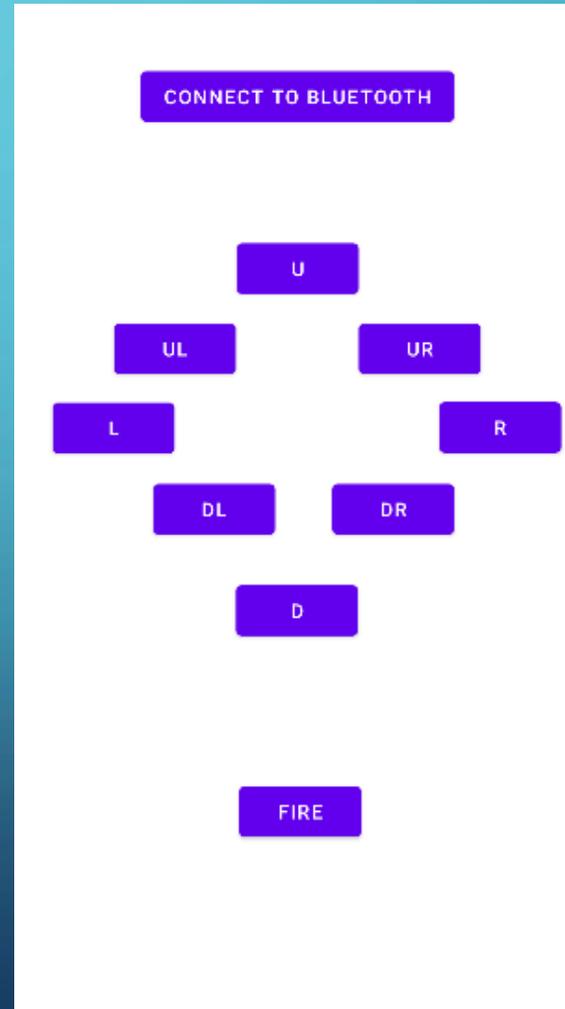
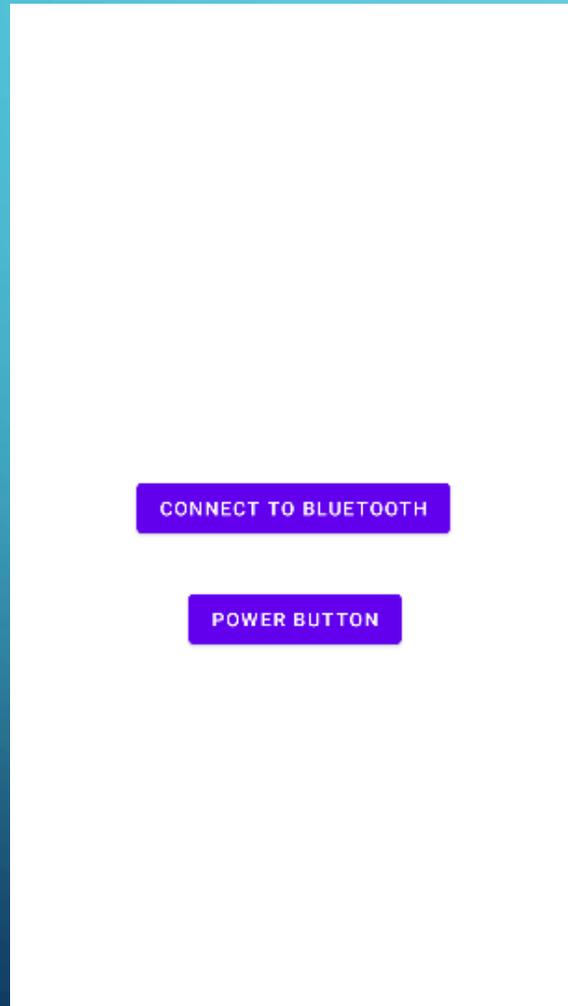
Username _____

Password _____

[LOGIN](#) [REGISTER](#)



ANDROID APPLICATION DEVELOPMENT CONT.

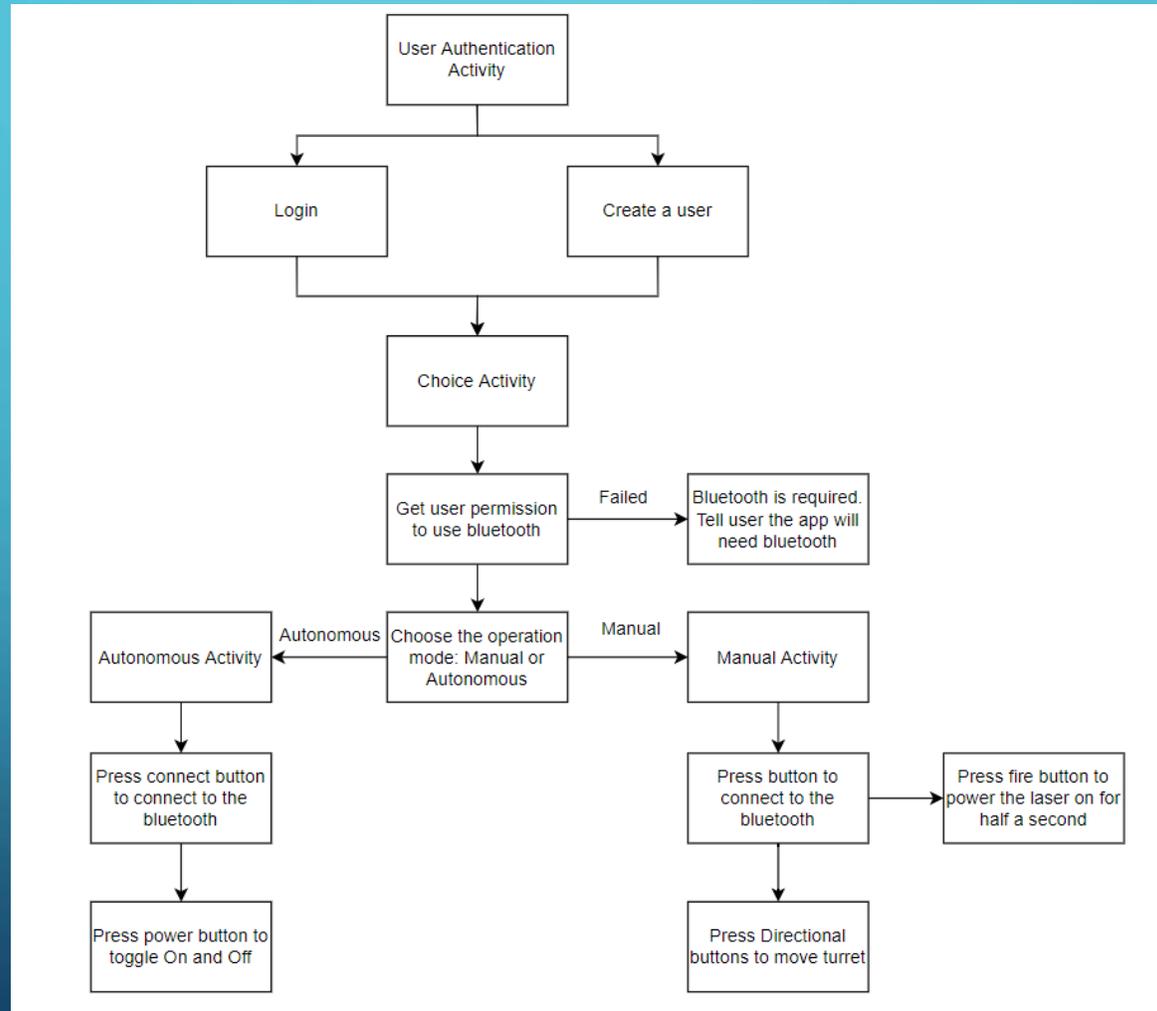


BACKEND: USER AUTHENTICATION

- Firebase is the dedicated user authentication server
 - Seamless connection between the app and the service
 - Built to have compatibility with Android Studios (Libraries and functions)
 - 10k per month free authentication uses on Spark Plan through Google Firebase
- Other options
 - Okta
 - 2100 per minute cumulative usage
 - AWS Amplify
 - 1000 build minutes per month

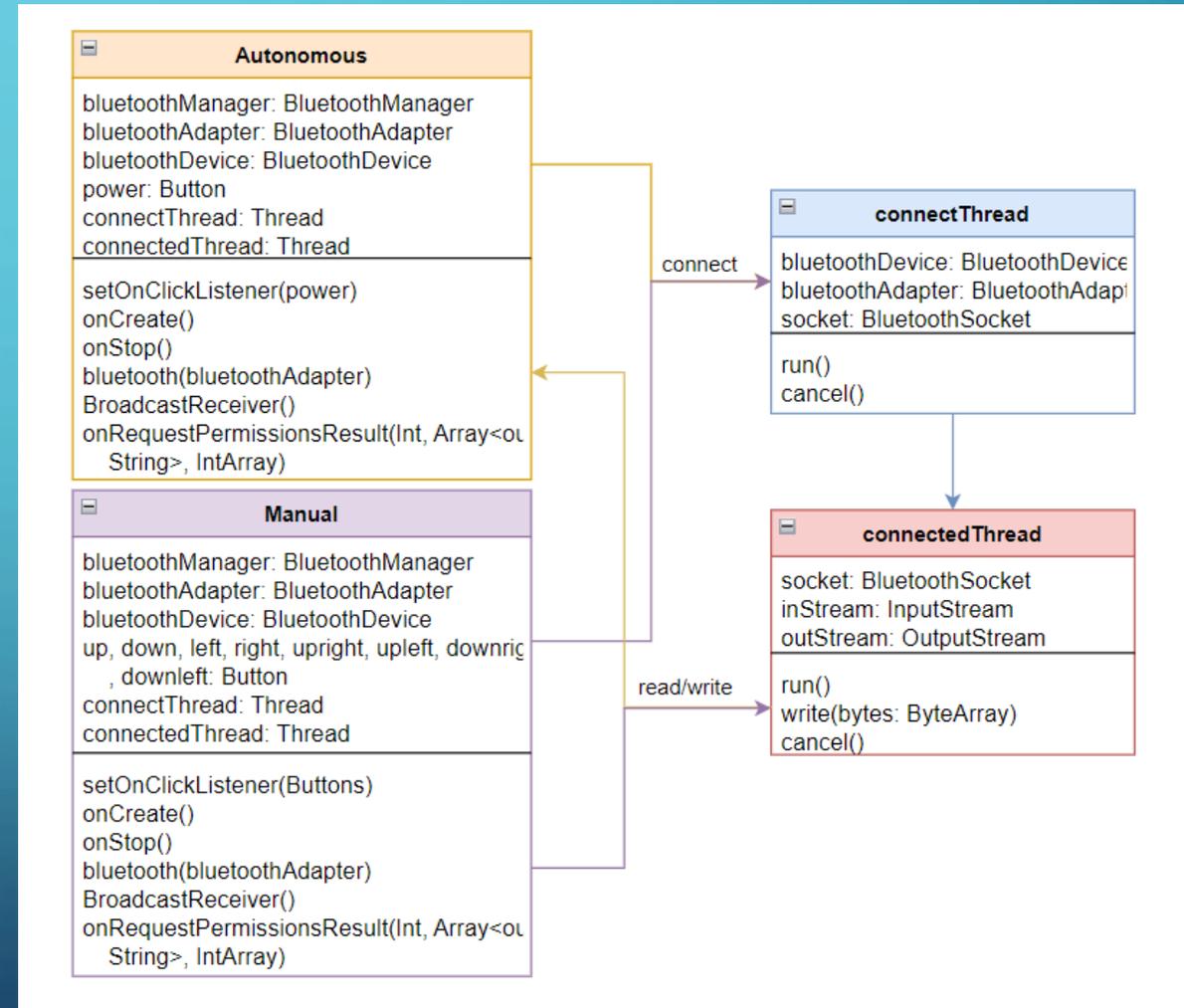


OPERATION OF ANDROID APPLICATION



OPERATION OF ANDROID APPLICATION CONT.

- Class Diagram of the android applications Bluetooth operation
- Bluetooth requires two separate threads
 - Connecting thread and operation thread



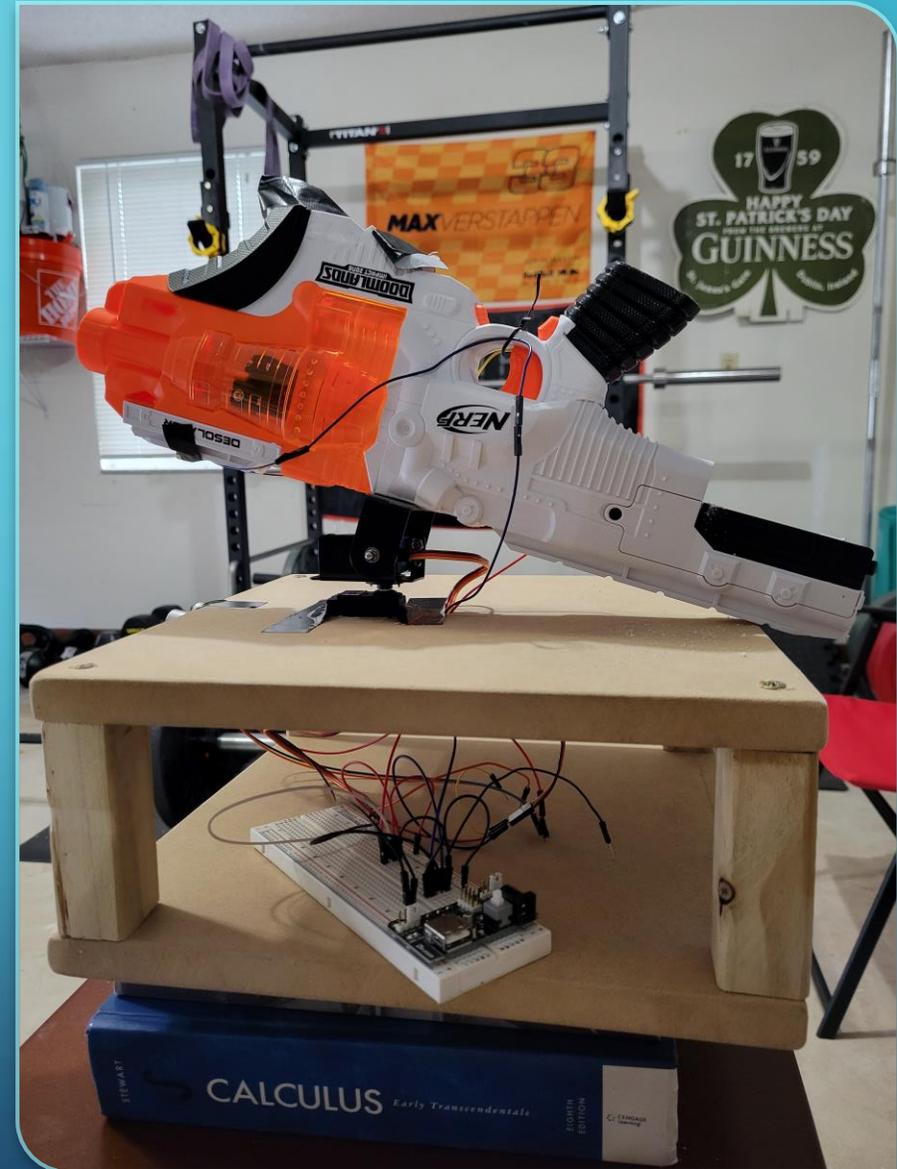
PIXYCAM

- Uses a hue-based color filtering algorithm to detect objects
- User friendly setup and operation
- Tracking is accurate and swift



CHASSIS DESIGN

- Wooden framed chassis
- Gun is mounted to the y-axis servo which is mounted to the x-axis servo through a custom bracket. The x-axis servo is mounted directly to the chassis.
- Electrical components are stored and mounted below the turret



TESTING APPROACH

- Building the two systems (primary and secondary) individually, then bringing the two together to test functionality
- We did all our software and hardware testing using the Arduino Uno and the Arduino IDE. This enabled us to quickly implement our code to our breadboarded MCU as they share the same chipset and coding libraries.



BUDGET

Quantity	Component	Price
1	Raspberry Pi 4	\$85.00
3	LM7086 Voltage regulator	\$4.46
5	Printed PCBs from JLCPCB	\$26.68
5	Printed PCBs from JLCPCB batch 2	\$36.75
5	Printed PCBs from OSHPARK	\$53
4	MG996R Servos	\$21
2	LJWRC Servos	\$40
1	Nerf Badlands Blaster	\$20
	Wood for chassis	\$10(scrap)
2	Arduinos for testing	\$45
	PCB components for 5 boards	\$26.87
		Total: \$368.76



GROUP MEMBER WORK DISTRIBUTION

- Computer Vision

- Primary: Danish
- Secondary: Ross

- PCB Design

- Primary: Kyle
- Secondary: Juan

- Bluetooth Programming & App design

- Primary: Juan
- Secondary: Ross

- Pixycam Programming

- Primary: Ross
- Secondary: Danish

