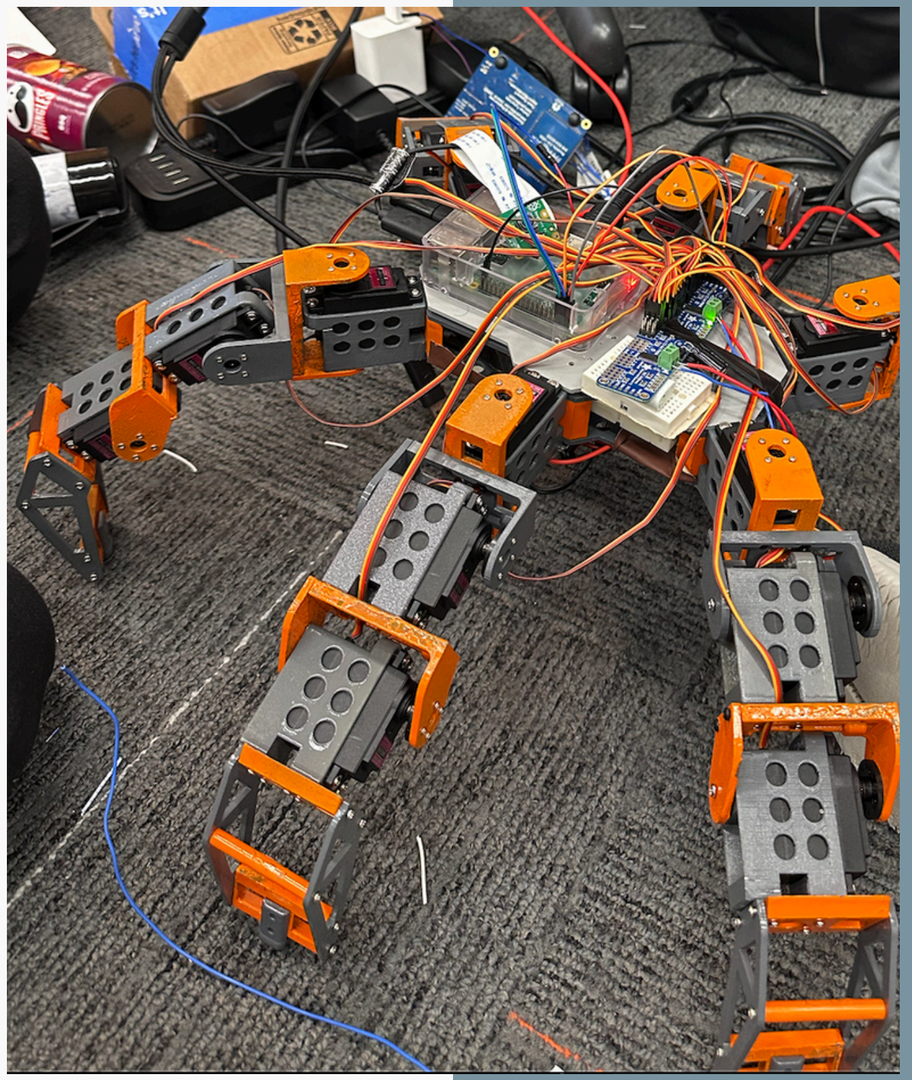
Rescue-feet

Gesture Controlled Hexapod with Live stream for rescue operations

Team 8: Archit Sharma, Sai Pranay, Samyu Kamtam

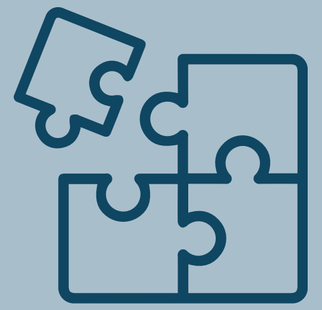
Project Overview

- Six-legged (18 DOF) locomotion & biologically inspired gait
- Gesture control for remote and convenient operation
- Live stream for teleoperation



Advantages

- Enhanced mobility
- Remote controlled operation
- Stability and redundancy
- Real time video streaming
- User-friendly
- Modular & scalable design



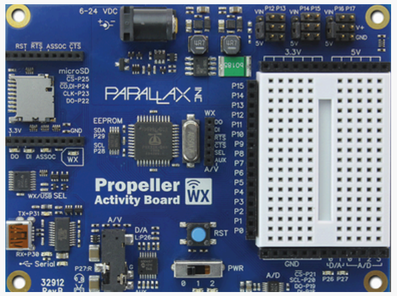
Disadvantages

- Complexity (Design and control)
- Cost
- Energy consumption
- Connectivity issue
- Payload limitations

System architecture

Gesture control - Propeller

Gesture control for navigation using flex sensor and Memsic accelerometer

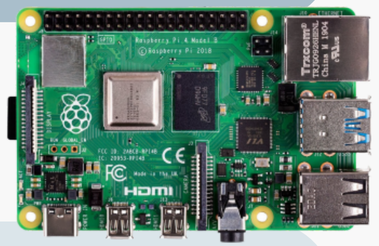


WiFi

Hexapod

Gait calculation for servo control

Rasperry pi



WiFi

Live stream

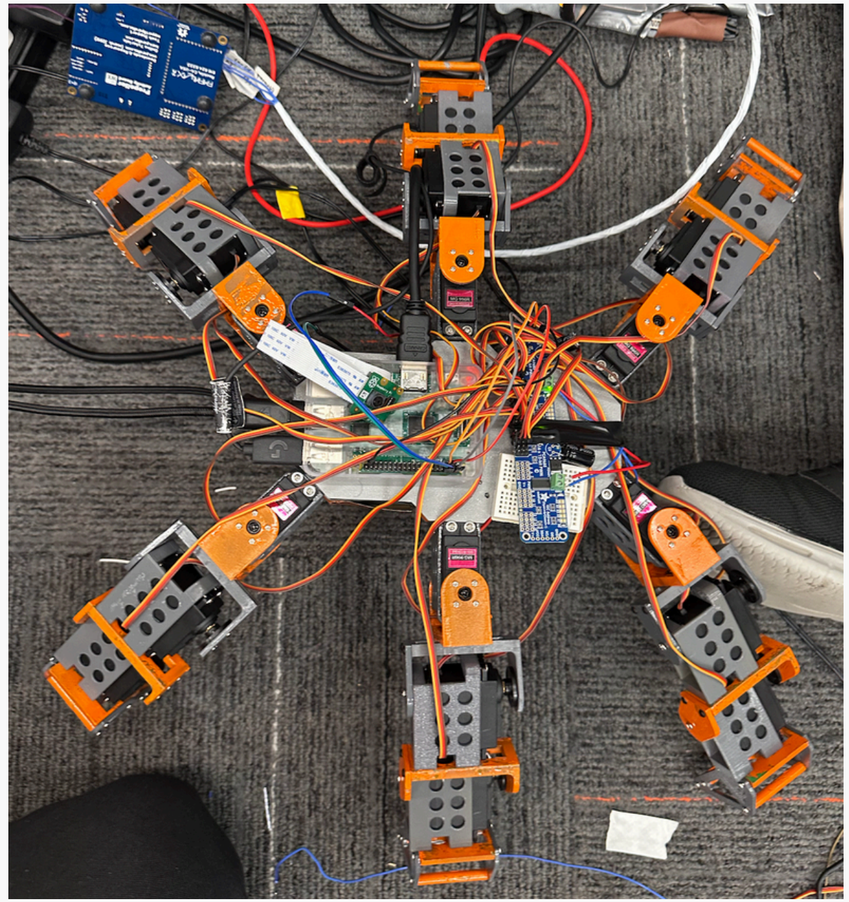
Picam live video capture

Hexapod

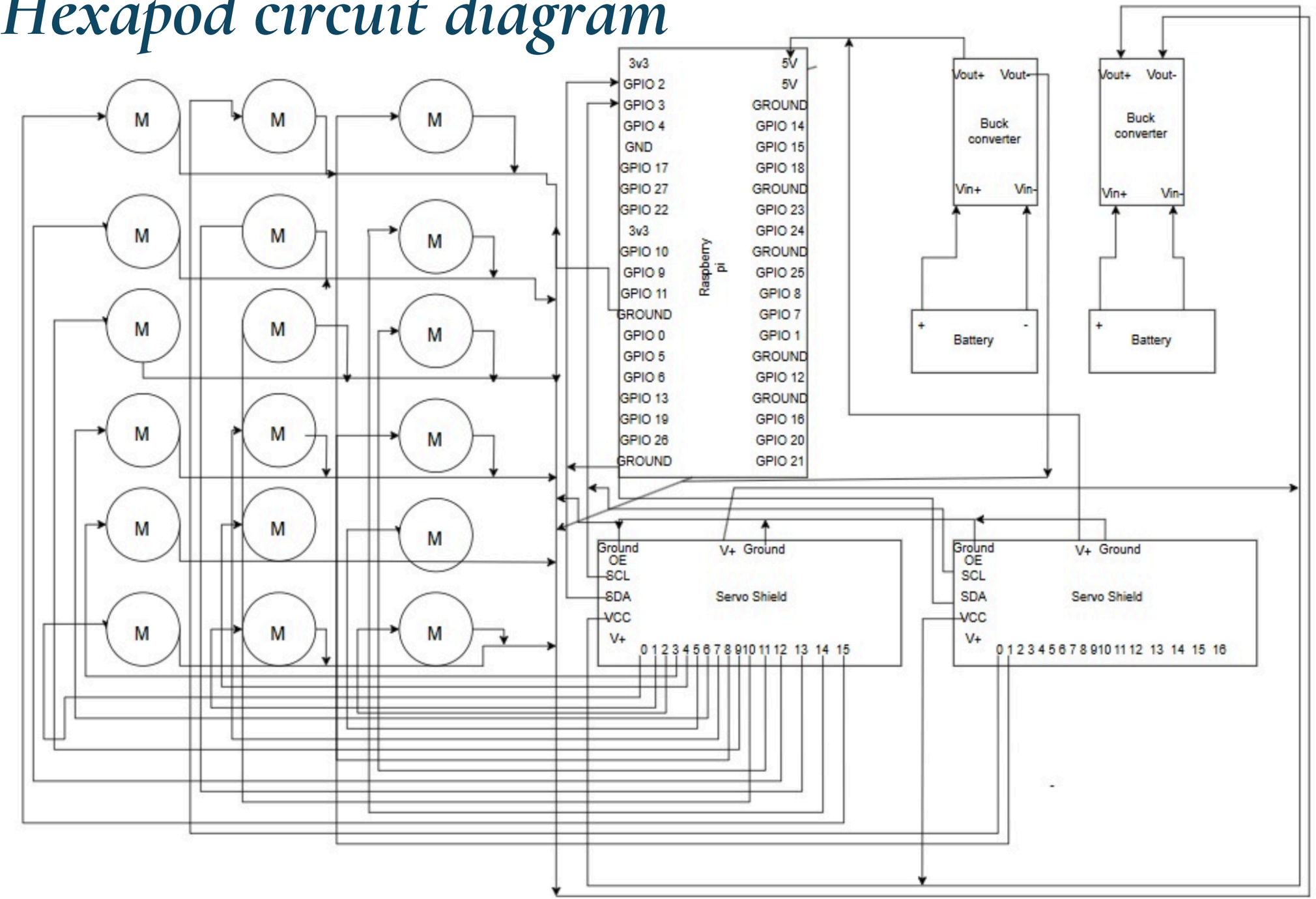
- **Servo**: The servo motors receive PWM signals from the I2C-based driver to perform precise joint movements.

- **I2C**: I2C communication connects the Raspberry Pi to the servo driver for synchronized multi-servo control.

- **Power distribution**:
  - A buck converter steps down 8.3V to 6V to safely distribute power to all 18 servo motors.
  - The Raspberry Pi is powered by stepping down 8.3V to 5V using a buck converter to ensure stable and safe operation.

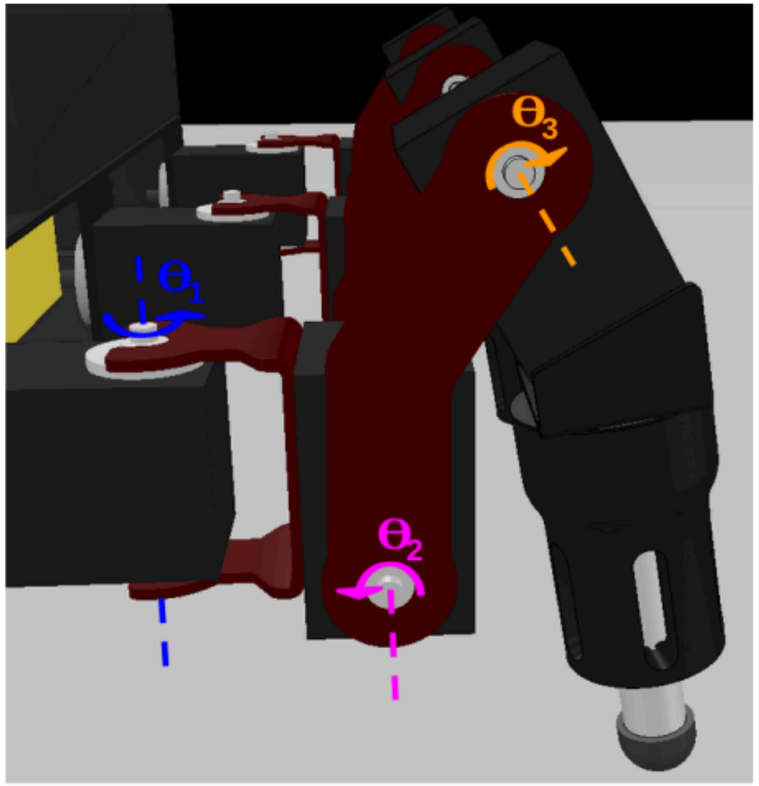


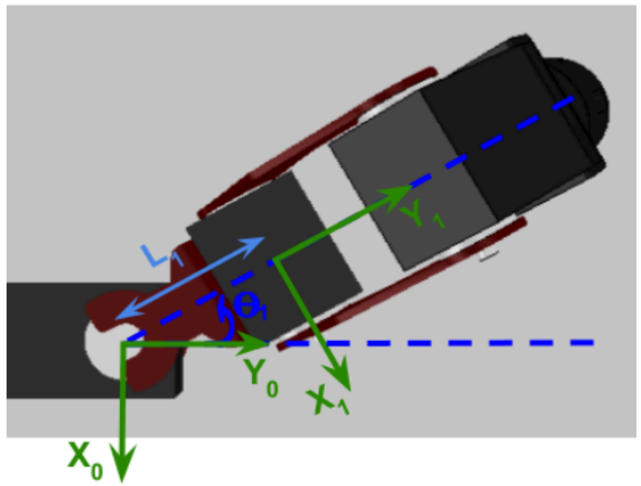
Hexapod circuit diagram



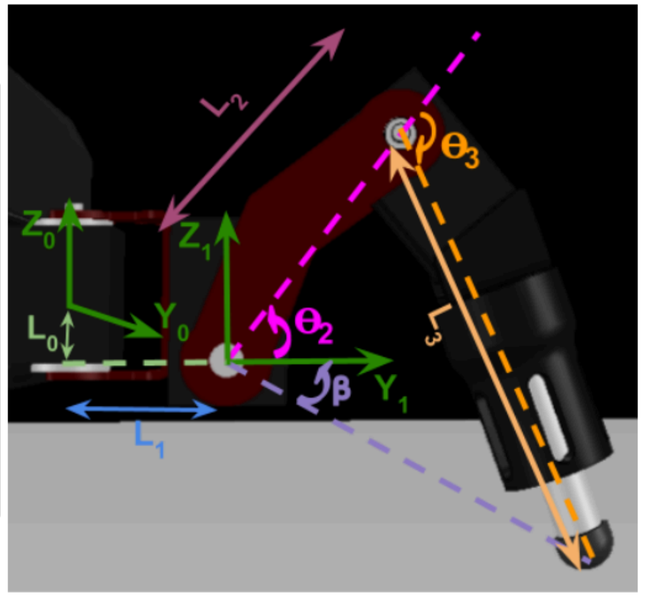
Rasperry Pi 3b

- OS: Raspbian 64 bit
- used for servo control and camera live stream





(a) Top view.



(b) Side view.

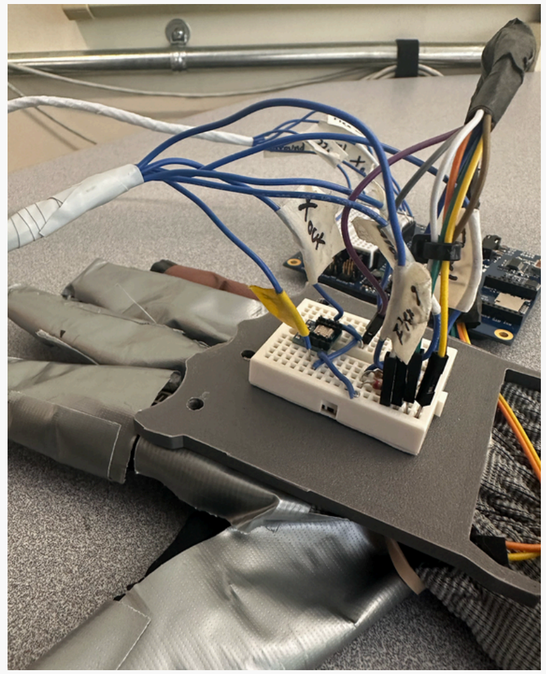
Kinematics

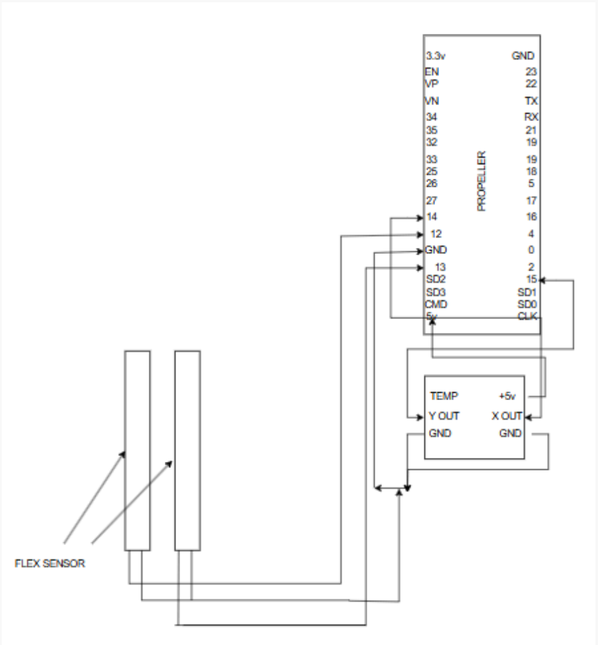
The angle of rotation of Joints 1, 2, and 3 are:

- $\theta 1 = \arctan\left(\frac{Y_0}{X_0}\right)$
- $\theta 2 = \arccos\left(\frac{-L3^2+L2^2+X_0^2+Y_0^2+Z_0^2}{2*L2*\sqrt{X_0^2+Y_0^2+Z_0^2}}\right) + \arctan\left(\frac{Z_0}{\sqrt{X_0^2+Y_0^2}}\right)$
- $\theta 3 = -(\pi - \beta) = -\arccos\left(\frac{X_0^2+Y_0^2+Z_0^2-L2^2-L3^2}{2*L2*L3}\right)$

Gesture control

- **Cogs for Performance:**
  - Multiple cogs on the Propeller read flex sensor and MX2125 accelerometer values while simultaneously encoding and transmitting data to the laptop.
- **Propeller Specs & Sensors:**
  - 8-core Parallax Propeller enables true parallel processing.
  - Each cog has 2KB RAM for fast local tasks.
  - Interfaced with MX2125 MEMS accelerometer and flex sensors for gesture input.



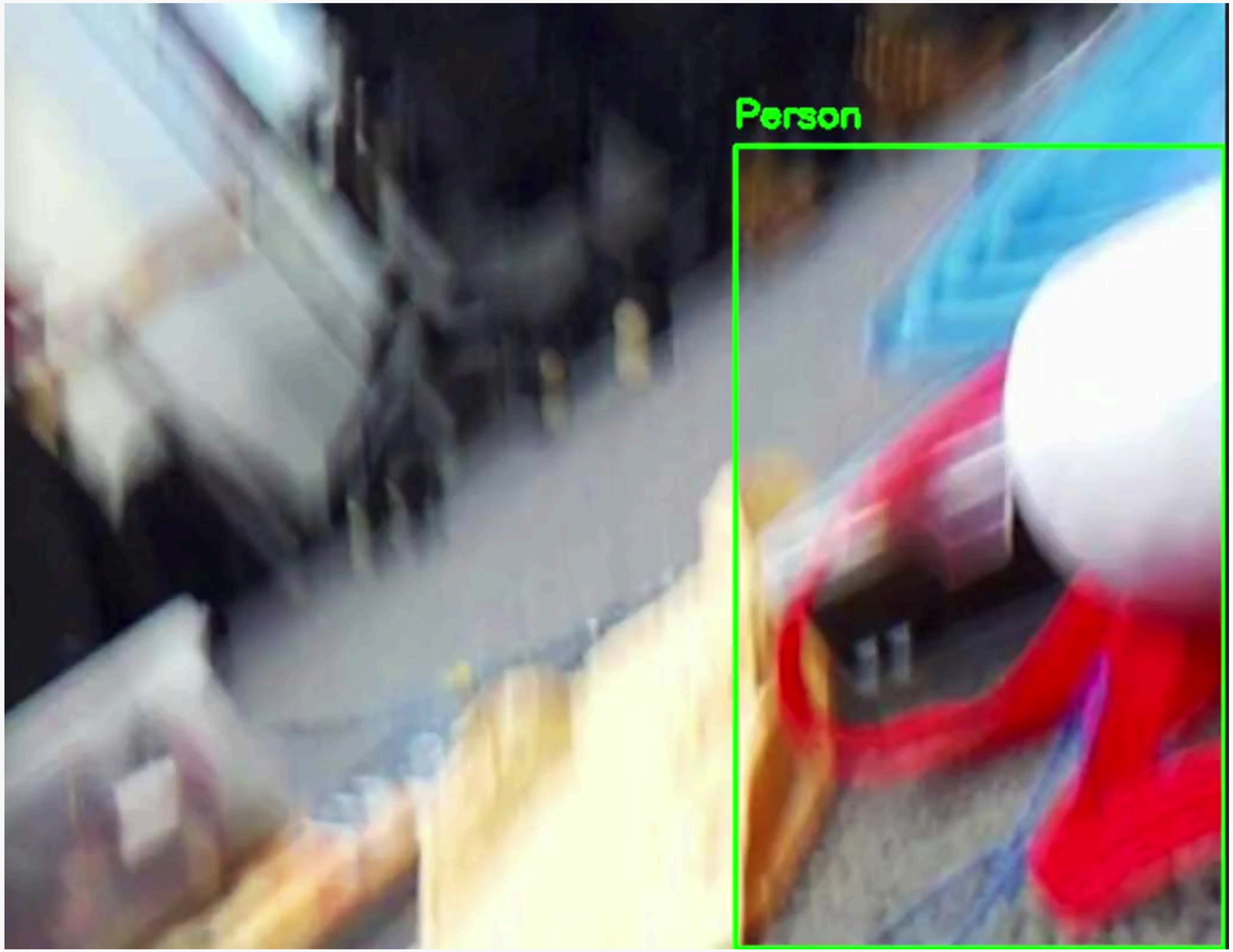


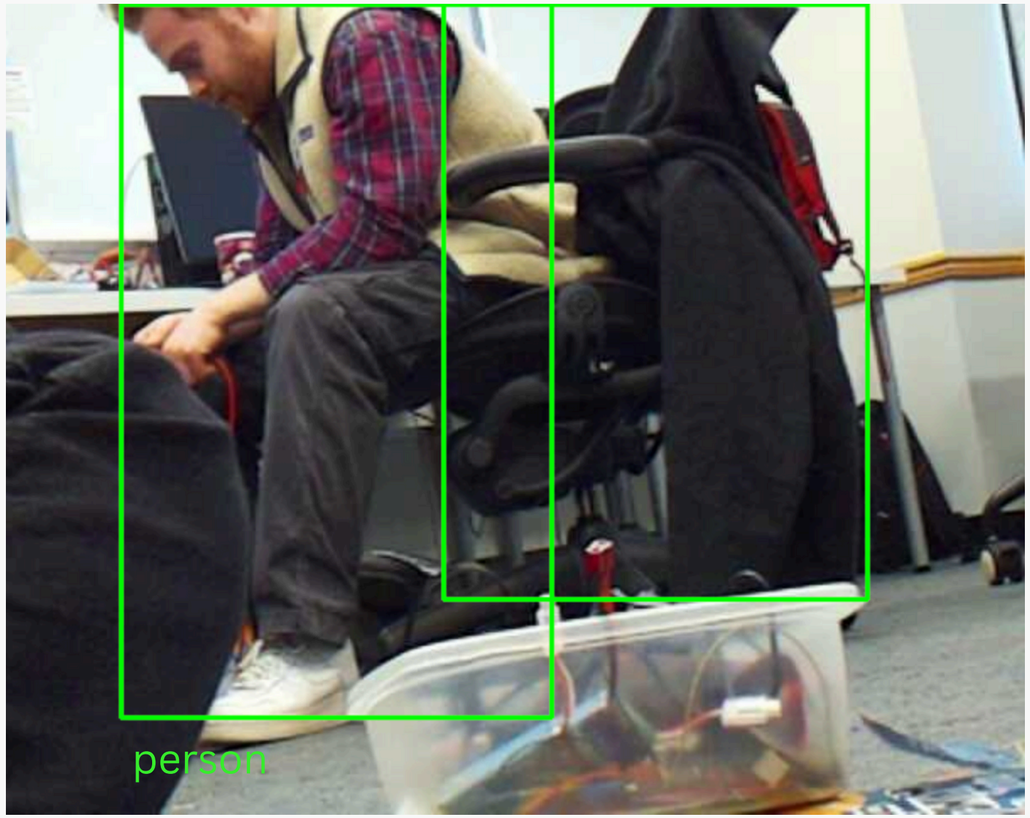
Encoder logic

State = flex1*18 + flex2*9 + x*3 + y

Movement	Flex 1 (*18)	Flex 2 (*9)	X (*3)	Y (*1)	Encoded value	Final value
Idle	0	0	1	1	4	1
Front	1	0	1	1	22	2
Front right	1	0	0	1	19	3
Front left	1	0	2	1	25	4
Back	0	1	1	1	13	5
Back right	0	1	0	1	10	6
Back left	0	1	2	1	16	7
Bounce	0	0	1	0	3	8
Kill	1	1	1	1	31	9

Live video stream





Basic object detection algorithm that decodes MJPEG that is streamed and displays the live video overlaid with detections using YOLO algorithm.

Conclusion

- Revolutionize Search and Rescue operations
- Enhances disaster response
- Scalable
- Has versatile applications
- User centric design with gesture based control
- Challenging with respect to system complexity and energy efficiency

Thank you

- https://www.raspberrypi.com/documentation/computers/camera_software.html#advanced-rpicam-apps
- https://www.youtube.com/watch?v=qs3KhLDUBmk
- https://github.com/shillehbean/youtubep2/blob/main/stream_usb_camera.py
- https://randomnerdtutorials.com/raspberry-pi-mjpeg-streaming-web-server-picamera2/