

Kennesaw State University - AUV Team - 2023 through 2024

Task List

Each node has it's own tasks that need to be completed

Please make sure to note which tasks you are completing within this file before you push to the github

Tasks

☐ GCS

☒ GUI Creation

☐ Ping Function

- Is a subsystem algorithm that pings the external subsystems to verify that they are still connected to the network

☐ SSH Function

- Is a subsystem algorithm that allows the user to SSH into the external subsystems to start and stop the subsystems
- Calls the Ping Function to verify that the external subsystems are still connected to the network (runs before SSH into the external subsystems)

☐ Start Subsystem Function

- Is a subsystem algorithm that allows the user to start the external subsystems
- Calls the SSH Function to SSH into the external subsystems
- Starts Individual threads for each external subsystems
- CP = Camera Processing
- MV = Machine Vision
- MP = Movement Package
- SP = Sensor Package
- SM = State Machine

☐ Stop Subsystem Function

- Loops through all of the external subsystems and stops them and saves their log files

☐ Restart Subsystem Function

- Loops through all of the external subsystems and restarts them and saves their log files

☐ CP

☐ ZED SDK Installation

- The ZED SDK is only available for Ubuntu and Windows

☐ ZED SDK Configuration

☐ ZED Camera Calibration

☐ ZED Camera Stereo Vision

☐ ZED Camera Depth Sensing

☐ ZED Camera Streaming to Port

☐ Bottom Camera Calibration

☐ Bottom Camera Streaming to Port

☐ MV

☐ Camera Input

- Listen to the port that the CP is streaming to (only if AUV is toggled '--AUV')

☐ Object Detection

- Separate the objects from the background
- Detect the objects
- Classify the objects (calls the Object Classification function)
- Return the objects in a dictionary '{classified_object: [x, y, z, width, height, depth, confidence]}'

☐ Object Tracking

- If the object x, y, z values are different from the previous frame, then update the dictionary

☐ Object Classification

- Using YOLOV8, classify the object

☐ State Machine Output

- Send the dictionary to the State Machine

☐ MP

☐ State Machine Input

☐ Convert State Machine Input to PWM Output

☐ PWM Output

- Write the PWM values to the corresponding microcontrollers

☐ SP

☐ I2C Input

- Define the I2C addresses for each sensor
- Read the I2C values from the sensors (every 0.1 seconds)

☐ State Machine Output

☐ SM

☐ State Machine Creation

- Define the states
- Define the transitions
- Define the actions

☐ State Machine Output