Aquaphoton ‘23

Electrical Mega Project

Sections:

1. Mega Project Overview:

This **Mega Project** is an overall **simulation** of how we as an Electrical & Software Team handle the ROV competition tasks. In this project you will be able to have a better understanding and implement **everything** that you’ve learnt during this training.

You will be working on a **GUI** that combines all of your **autonomous software tasks** that are based on **image processing** in an organized way, gives you control over your whole **electrical PCB system** and also gives you feedback from your **camera**, **sensors** and **motors**.

2. Technical Tasks:

2.1. GUI :

In this section you are required to make a Graphical User Interface (GUI) that combines all of the necessary **GUI features** and the required Software Tasks together in an organized way.

This is a **simulation** that will give you a better understanding of what a software team does in order to give the Pilot/Copilot a better user experience in order to

1. Navigate the ROV
2. Finish some required tasks
3. Keep being updated with the ROV status at all times.

Your GUI will be **scored** according to the following :

- **Version Control:**
  - You MUST use any version control system ex:(git/github) and push all of your commits ***Daily*** there
- **Main features :**
  - Live Camera Feed :
    - Should contain a place for live webcam feed that you're going to use in some software tasks.**Use multithreading**
  - ScreenShot button :
    - Gives you the option to take a screenshot from your webcam live feed at any time.
  - Indicator Icons :
    - You should have an indicator that changes its colors according to connectivity with your car.
    - A motion indicator (forward, backward..etc.)
    - Speed Indicators (low, medium, high).
    - Sensor readings from both the current and voltage sensors.
  - Serial Connection:
    - Your GUI should be able to receive and send data ***at the same time*** (Use Multithreading)
- **Overall Design :**
  Should have a nice looking overall design that is well organized and easy to use.

- **Modularity & Adaptability :**
  - Should be modular and have the ability to adapt to any changes that might happen. Ex: We can add another camera feed easily in the code without major changes

- **Multi-Window :**
  - **Your GUI (QMainWindow) should have push buttons each one should open a certain Software Task (Qdialog)**

- **Back-End :**

  "Any fool can write code that a computer can understand. Good programmers write code that humans can understand."
  – Martin Fowler

  - **OOP Concepts :**
    - Concepts like inheritance, polymorphism and ***Abstract classes(abc class)*** are strongly recommended for you to use as it will help you build a well organized code.

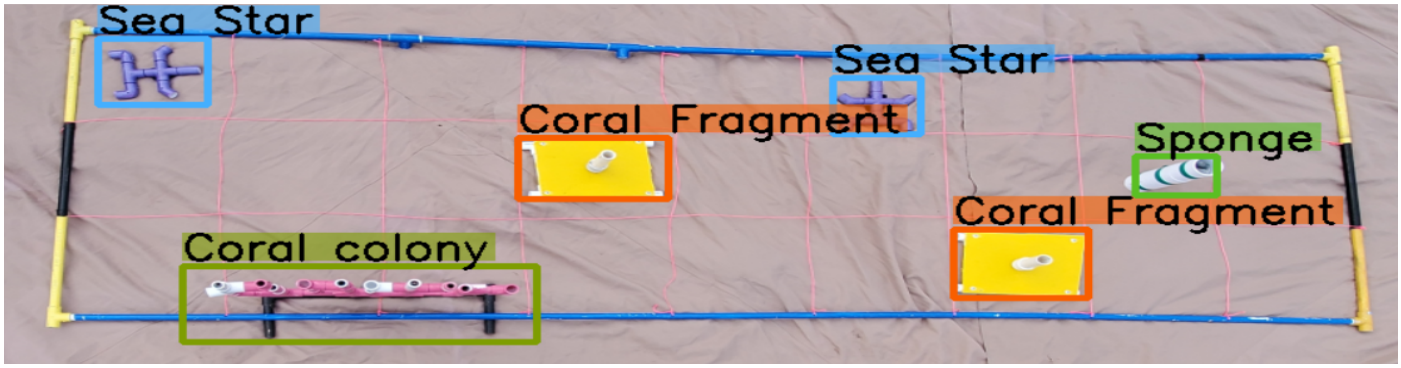
  - **Clean Code**
    - Use **PEP-8** naming conventions.
    - All of your Icons, Images, data paths must be relative not absolute.
    - Separate your front-end (UI) from you back-end (Logic)

2.2. Software Tasks :

In this section you are required to use your Image processing and software knowledge in order to finish the following tasks and represent them in your GUI in Sub Windows.

- **Stitching & Object detection :**
  - In this task you will be given five images, each image containing a different part of the same map. *Side Note : These five images are just for example, your program should be able to work with the same efficiency on images of any similar map.*
  - **First:** you are required to stitch these five images together and generate a correct image of the whole map.
  - **Second:** you will use the previously generated image to detect and identify the shapes shown in the whole map and count the number of each shape.
  - This task should be integrated in your GUI so that we can see the generated image and the number of each shape present in the map.
  - Link of the images : **https://drive.google.com/drive/folders/1ej68wm0PilBAo_fTtz8DEMfyhYM6rbXO?usp=sharing**

Example for the Output:



❖ **Stereo Vision :**

Stereo vision is the process of extracting 3D information from multiple 2D views of a scene. In ROVs, stereo vision facilitates the navigation system, measuring things underwater , and calculating how close the ROV is to a certain object.

➢ **You are required to implement a stereo vision algorithm in python to be able to calculate the length of an object in any image.**

➢ **You can find the pipeline to create a stereo vision system as well as the images you need here : https://drive.google.com/drive/folders/1C9XwKlAyHzGEXRXGFZKsDWWOp0iFerVe?usp=sharing**

➢ **Resources:**

- https://www.youtube.com/watch?v=KOSS24P3_fY
- https://www.youtube.com/watch?v=kxsvG4sSuvA&t=31s
- https://aaronolsen.github.io/about_me.html
- https://pramod-atre.medium.com/disparity-map-computation-in-python-and-c-c8113c63d701
- https://stackoverflow.com/questions/27856965/stereo-disparity-map-generation
- https://stackoverflow.com/questions/36172913/opencv-depth-map-from-uncalibrated-stereo-system
- https://www.cc.gatech.edu/classes/AY2016/cs4476_fall/results/proj3/html/sdai30/index.html

2.3. Hardware Tasks :

In this section you are required to build a remotely operated car that can be controlled wirelessly using the GUI you built. You are required to design and fabricate a **PCB** or multiple **PCBs** that controls this car. Your **PCB (PCBs)** should contains all of the following :

- **Bare microcontroller circuit :**

  A bare minimum circuit means the usage of the microcontroller without its development board. For example, you may use the Atmel ATmega328P microcontroller by itself without the arduino board.

  You may use any additional components needed for the microcontroller to be fully functional (Resistors, Capacitors, Crystals, Diodes, etc.).

  You must use the appropriate socket or base in your PCB so you can use this microcontroller in other projects as well. **DO NOT SOLDER THE MICROCONTROLLER DIRECTLY TO THE PCB!**

  Arduino IDE supports a wide variety of microcontrollers and different methods to program them, you can freely choose any of them.

- **Motor divers:**

  You are required to use a 2-channel motor driver to control the car. You can use any motor driver module or You can build one yourself (search for how to make a motor driver using relays). You should be able to control the speed of the motors.

- **ESCs [2 Channels] :**

  Your PCB should have the option to control 2 ESCs (electronic speed controller) which control 2 brushless motors instead of the motor drivers.

- **Sensors :**

  Your PCB should contain the following sensors:

  - Voltmeter Sensor :

    This sensor should be used to validate the Voltage going to the car.

  - Current Sensor up to 5 Ampere :

    This sensor detects electric current consumed by the car.

    Side Note : you don't have to purchase the sensor itself but you are required to add needed connections.

Side Note : you are required to show those readings on the GUI.

- **Motors :**
  - 2 DC Motor :

    You are required to have a working car frame which you can find in electronic stores or you can build yourself and it should be equipped with the required number of DC motors to make it able to navigate.
  - 2 Brushless Motors :

    This will be **provided to you by the team with their compatible ESCs** but you should prepare their connections on your PCB.
- **Indicators :**
  - LED Indicators :
    - You'll be using one **RGB LED** to indicate the Speed of the Motors [you'll adjust 1 color for each speed :Low, Med & High].
    - You should add other indicators to help the user easily use and debug the car. For example, an indicator for the input voltage.
- **Bonus points:**
  - Your design must be modular.
  - Your PCB should be well documented and easy to use.
  - You should make as much of the project without relying on premade modules.
  - Your product should be as inexpensive as possible.

2.4. PID report :

In this section you are required to write a report on PID control and how it is used to stabilize the motion of the ROV in water using the readings from the IMU sensor and pressure sensor.

3. Non Technical Tasks:

3.1. Daily Progress Reports :

You are required to write and deliver a report before **12:00 AM everyday** explaining your **progress** and the **detailed division of labor** between the team members on this particular day. This will be the **official** way of communication between us so that we will be able to help you and know where you guys are at.

3.2. Technical Report :

In this task, you are required to write a technical report explaining your whole Project. It should include the following :

- **Electrical System :**

  In this section, you are required to give a detailed explanation about everything in your **PCB** : Schematic connections, Layout Design & Routing.

- **Software tasks' Algorithms :**

  In this section, you are required to give a detailed explanation of all your software algorithms that you used in order to finish the **required tasks**.

  You should also design a flow chart for each required task and include them in this report.

- **GUI Features :**

  In this section, you should explain and show off all of your **GUI** features that make your GUI unique. You should also include the following:

  - **Flowchart** Explaining Back-End Algorithm
  - **Main Window Features** including screenshot of the design itself
  - **Sub Windows Features** including how you integrated the software tasks in this GUI.

4. Teams & Deliverables:

4.1. Teams :

In this Project, you guys will be working in teams as follows:

- **Team 1:**
  Ahmed Sakr, Ahmed Falah, Karim Sherif , Mohamed Ahmed Ebrahem, Seif Bassiouny
- **Team 2 :**

  Ahmed Sameh, Ahmed Khalafallah, Marwan Ahmed, Salma ahmed
- **Team 3 :**
  Ibrahim mohamed, Eyad Salama, Mahmoud Ahmed, Ahmed Anan, Dana Hamdy .
- **Team 4 :**

  Ahmed Aly, Abdelhamed Mohammed, Amr Alnas,
  Mohamed Elsherif
- **Team 5 :**
  Mahmoud hamada, Mohamed Ahmed Mohamed, Moustafa Ibrahim , Nada Mamdouh, Ziad Amr

4.2. Deliverables :

You are required to deliver the following :

- **Daily Progress Report**
  should be Delivered Every day before 11:59 PM
- All the Software Project Codes you developed for both **GUI** & **Software tasks.**
- **PCB** with its sticker & CAM outputs.
- **Technical Report.**
- **PID Report**

The Deadline for delivering these files is : 21/8 at 11:59 PM