Design and Implement Remotely Operated Underwater Vehicle (ROV)

Keroles Shehata, Youssry Awad, Eid Alaa Eldein, Abdullah Ahmed, Abanoub Ibrahim, Hossam Shabaan, Mohamed Ahmed, Ahmed Mohamed and Eslam Adel
Assiut University, Egypt, kerocr90@gmail.com, yossrya87@gmail.com, eidelrafy98@gmail.com

Supervisor: Khalil Ali Khalil, Professor assistant at Mechanical Engineering
Assiut University, Egypt, khalil.ibrahim@aun.edu.eg

***Abstract*– The purpose of this project is to design and build a remotely operated underwater vehicle that is able to make explorations underwater using cameras and send clear images and live videos to the screen on the surface. In our project, we decided to concentrate on the objectives to be achieved, which are the motion of the vehicle thrusters underwater, controlling them by on off control system, determining the directions, sending images and videos from underwater to above to help exploration process.**

Keywords—Thrusters, Artelon, LabVIEW, ROS and PWM

I. Introduction

Nowadays, robots play a major role in every field and every industry. Robots are used in industrial, production, medical, military, archaeological and sea exploration [1]. The purpose of our project is to make a submarine type vehicle in order to track or visualize the seabed for archaeological, searching and exploring underwater life, creatures, resources and phenomena. The ROV (Remote-Operated Vehicle) is designed to be equipped with an on-board electronic system in order to achieve such functions. The project has passed by many steps to achieve its goals, from choosing the right materials crossing by the mechanical design, motors selection, sealing and watertight system and control part. We have used many methods to tune it together to obtain the required functions.

II. Setup Design

A. Conceptual design

The proposed design was to make a mechanical system that is not too heavy in weight to be easily operated [2] and reduce the load on the motors, which its selection was another issue to discuss, the watertight of the tool box and the electrical system was really challenging, sending and receiving the signal to the microcontrollers (Arduino and raspberry pi 4) between the operator on surface and under water, installing the vision system and receiving signal from it to the screen on surface.

B. Vehicle frame

The function of the frame is to hold all the mechanical and electrical components that we use in this project including the thrusters, electrical and electronic toolbox and the camera. The lighter the frame is the better. Our concern was to make it symmetrical and allow water to flow through in order to facilitate the movement.

This design consists of only three plates, two plates for the sides and one plate at the lower of the vehicle and handrails of aluminum used for making the frame firmer [3] as its used to fasten the two plates of the sides as shown in Fig.1

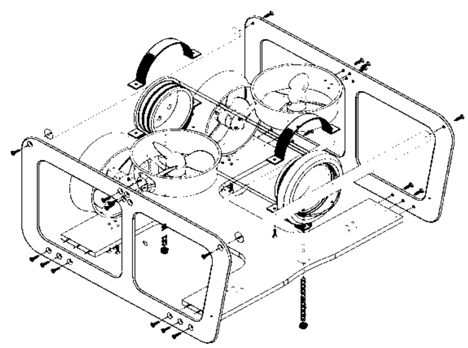

Fig. 1 Vehicle design

C. Frame Material

Choosing the right material for the frame of the vehicle was as important as any other part of the project. The best way to build a remotely operated underwater vehicle is to be half above water and half under water when there is no force impact on it. The chassis has also to be firm to endure the weight of other components and collisions. In choosing the right material that suites our aim in building a firm ROV, we had two main issues to put in mind: Choosing a material which density is close to the density of water (1g/ml) and Choosing a material with good fatigue resistance. So that we have decided to choose Artelon as our main material of the project. We used Artelon sheets with thickness of 1cm in our manufacturing.

D. Other Components

1) Tube: As we mentioned before the safety of the electrical and electronic components underwater is a great priority and that is the function of the tube. We used transparent acrylic tube. The reason for choosing a transparent material is the idea of being capable of seeing all different components mounted and to check easily its well assembly and not throwing the device with some disconnected cables.

2) The Dome: The semi sphere as shown in Fig.2 is also made out of acrylic and the reasons for choosing a transparent material: Being capable of seeing all different components mounted and to check easily and Install the camera at the inner surface of the semi sphere and the transparent material allows it to record and take picture in explore mission.

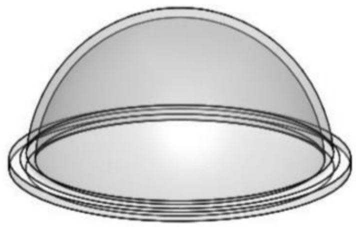

Fig. 2 The Dome

3) The back cap: these caps are designed and machined by us. The two caps are made out of Artelon, the same material the frame is made from. The outer diameter of the back cap is 16 cm; it has 10 holes for the tethers and the wires that is used to connect the electronic and electrical components with the operator on the surface

4) Front cap: the need arose to design a system that held the semi sphere (the dome) against the tube. So, it was decided to make the front cap. Logically, the front cap should have a bigger diameter so its outer diameter is 16 cm.

5) Clamp: It is formed by a semi-circular metal clamp made out of aluminum. Its function is to attach the tube to the main base. It is achieved by using two short semicircular cylinders through which is attached to the base.

E. Penetrators

we had to make sure of blocking every drop of water from getting inside of it. There is a tether used to transfer power and signal from the surface to under water. In order to deal with this situation, we made 10 holes in the back cap to pass the tether and wires through. We used cable glands direct penetrators. Its main function is to prevent water from passing from outside the tube to inside the tube. The type of penetrator we have used is PG-7 as shown in Fig. 3

F. Sealing

The most challenging part in the entire project, which is watertight and sealing. Preventing water from reaching the inside of the tube is a must, so we made lots of research to learn of others mistakes and avoid doing the same useless steps and learn the effective ways of sealing underwater.

To make the perfect watertight we had to know the exact parts, which can pass water through the tube, which were:

1) Penetrators' connection with the back cap.
2) Penetrators' connection with the wires.
3) Connection of the front cap with the tube.
4) The round edge of the front cap.

For the connection of the penetrators with both the back cap and wires we have used a water proof type of Epoxy (Araldite) consists of two tubes mixed together and applied on the connection's parts.

For the connection of the front cap with the tube, we have used a waterproof type of strong silicone sealant. We applied

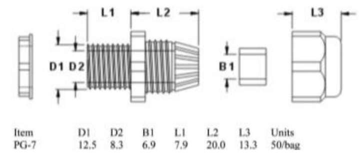

Fig. 3 PG-7 Penetrators

the silicone between the front cap and the tube right before fixing screws. For sealing the round edge of the front cap, we have used another type of Epoxy with an injection tube to help us apply the Epoxy on the edge perfectly.

G. Thrusters

The ROV's propulsion system is made up of two or more thrusters that propel the vehicle in a manner that allows navigation to the work site. Thrusters must be positioned on the vehicle so that the moment arm of their thrust force, relative to the central mass of the vehicle, allows a proper amount of maneuverability and controllability. We chose to use 4 thrusters of T200 thrusted [4] by Bluerobotics. The T200, shown in Fig. 4, Thruster is a low-cost high-performance thruster for marine robotics. To control the thruster, we need to use a speed controller so that we used a basic ESC as in [5].

The Basic ESC is a simple brushless sensor speed controller. It is based on the BLHeli ESC design with upgraded features and performance. An electronic speed controller (ESC) is necessary to run any three-phase brushless motor like Blue Robotics' thrusters and motors. This 30-amp ESC is sufficient for the T200 thrusters and runs the BLHeli'S firmware.

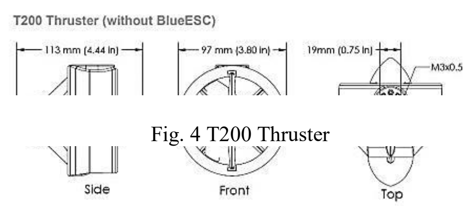

Fig. 4 T200 Thruster

H. Vision System

The video is considered the main sensor of ROV which is done through using different types of cameras. For our ROV, it is equipped with a (Raspberry Pi Camera module) [6] in the forward direction that are waterproofed in custom designed 3D printed cases that minimize space. The Camera is connected to the Raspberry Pi with a camera serial interface. The description of Camera is that it is a High Definition camera module compatible with all Raspberry Pi models. Provides high sensitivity, low noise image capture in an ultra-small and lightweight design. The camera module connects to the Raspberry Pi board via the CSI connector designed specifically for interfacing to cameras. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data to the processor.

I. Lighting systems

The lights provide illumination for the camera underwater; sunlight disappears rapidly underwater and many ROV missions occur at depths that are in complete darkness. A powerful led as in figure 46 is used as the source of lighting that help cameras to capture clear videos and photos especially in dark places underwater. The LEDs are put in housing, which is a circle part made of nylon PA6 with a groove to put the LED in. It is covered by a part made of transparent acrylic. The housing is screwed with bolts and nuts and is covered with silicon for tight sealing, and this makes it open and close easily. Lights like Lambs and Lasers.

III. Control System Design

The system has different sensors and microprocessors that builds the ROV. Those microprocessors and microcomputers often use programming languages to make different scripts, so in order to understand better the programming language election, these components will be briefly remembered and will introduced some software's specification. In our project we managed to control our vehicle with two different control systems using ROS and LabVIEW which controlling is done on the same PCB.

A. Arduino

Arduino is known for be one of the most powerful and sustained open source microprocessor of the market. Therefore, documentation and information about the different sensors can be acquired easily providing a strong base around the ROV's construction and specifications requirements. The Arduino is the responsible of thruster's movements

B. Raspberry Pi 4

Raspberry Pi is a microcomputer able to execute complex orders and it is used in a variety of applications, since more simple ones up to some complexes like fingerprint recognition. Is important to consider the difference between a microprocessor and a microcomputer. While the Arduino is good at making operations repeatedly at a fast rate, a microcomputer like the Raspberry can execute more complex programs, and it is built with the capability of having an operating system.

C. Python

Python is an open source programming language used now by a lot of different people in different applications. It has been decided to choose to use Python because of different reason. The first one and most important is that as mentioned before Python is the core component of the Raspberry ecosystem forcing us in some way to use it before anyone else.

D. Level Logic Converter

Its main function is to connect a 3.3V device to a 5V system. The bi-directional logic level converter is a small device that safely steps down 5V signals to 3.3V AND steps up 3.3V to 5V at the same time.

E. PWM control

Pulse width modulation (PWM) is a modulation technique that generates variable-width pulses to represent the amplitude of an analog input signal. The output-switching transistor is on more of the time for a high-amplitude signal and off more of the time for a low-amplitude signal. The digital nature (fully on or off) of the PWM circuit is less costly to fabricate than an analog circuit that does not drift over time.

F. Electronic Design

In Fig. 5 shows the block diagram of the proposed hardware that consists of two parts: the remotely controlled vehicle and the remote control.

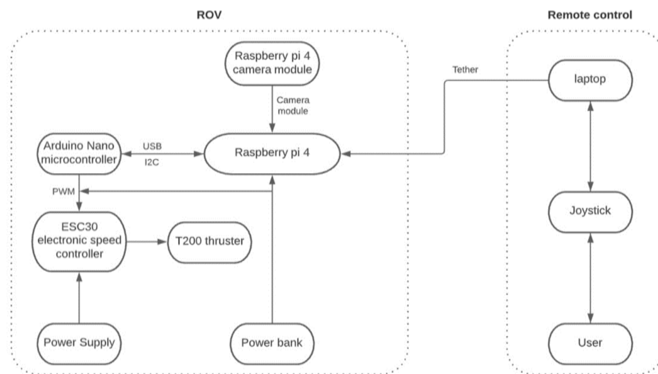

Fig. 3 Project block diagram

These two subsystems are communicated through an Ethernet cable so that the user can control the ROV from a personal computer, laptop, all with Virtual Network Computing (VNC) connectivity. The ROV hardware involves a Raspberry Pi 4 that is responsible for executing parallel tasks, such as:

1) acquisition of video by means of a digital camera.
2) measurement and recording of the different variables associated with the sensors of the system.
3) motors control in coordination with an Arduino microcontroller
4) battery voltage monitoring for energy consumption and internal temperature.
5) communication management with the remote control.

The used digital camera is a raspberry pi camera module 8 Megapixels. Besides, two power sources are integrated into the system, one with a power bank of 5 V with 10,000 mAh for the Raspberry Pi 4, an Arduino microcontroller, and digital sensors, and the other one is a power supply 12v to power four brushless motors. With these battery banks, the ROV has an autonomy of up to 2 h. The remote control shown in Figure 50 consists of an Ethernet network hub (Ethernet switch/router) interconnected by an Ethernet network cable to a computer or

SoC, from which the user controls the ROV through a touchscreen type graphical user interface (GUI).

G. Basic thruster control

ROV mobility is achieved through control of thrusters to reactively vector fluid for vehicle movement [7]. In the early days of ROVs, each electrical thruster motor (driving a propeller) was controlled individually via a rheostat linked directly to the thruster. This made for a dreadfully difficult control regime! As the technology evolved, control mechanisms arose allowing for a more intuitive human machine interface. Later iterations gave rise to the joystick for scaling thruster output, thus allowing thrust vectoring and, hence, finer control of vehicle movement. For direct human control of thrusters, a joystick is normally used. The joystick typically outputs a pulse width modulation signal (or some other electrical scaling signal) to direct motor output in some linear fashion. That signal can control power output to either an electric motor (for electrical thrusters) or a hydraulic valve pack/servo (for hydraulic thrusters).

H. High power board

It is a four of 5 DC/AC Blown Fuse indicator circuit with LED Display [14]. A Fuse is the equipment protects that use often most. Because of cheapness can use protect electronics expensive circuit. Generally, when fuse torn us can know immediately. but in sometimes Fuse is torn already we don't know. such as in electricity automobile system brake system, the system delays the electric current very much, etc. Fuse torn get into trouble at we must know for immediately the safety. The PCB is for providing the ROV with power, as it carries two 220V to 12V converters, it has 12 XT60s outlet connectors to power different devices inside the ROV. We used XT60 as it withstands high current in the high-power board. It is also characterized by the polarity and hence avoiding humans' mistakes if we connected the two terminals in a reverse way. In the figure below the real image of the PCB. Fig. 6 is the schematic diagram of the protection circuit.

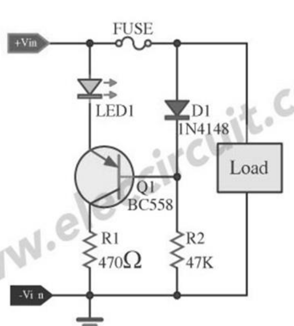

Fig. 4 Safety circuit

I. Controlling ROV Using ROS

The Robot Operating System (ROS) [8] is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. It is creating truly robust, general-purpose robot software is hard. From the robot's perspective, problems that seem trivial to humans often vary wildly between instances of tasks and environments. Dealing with these variations is so hard that no single individual, laboratory, or institution can hope to do it on their own.

We first chose the Ros version "Noetic" and selecting our platform which is Ubuntu 20.04 which is previously installed on both laptop and raspberry pi 4 then Setup sources.list, then install ROS and installing Joy package [9]. The joy package contains joy_node, a node that interfaces a generic Linux joystick to ROS. This node publishes a "Joy" message, which contains the current state of each one of the joystick's buttons and axes. To be able to bring up the joy node and display the data coming from the joystick over ROS and Using a Linux-Supported Joystick with ROS [10]. One of the other more important parts is to see exactly what the ROV actually sees. Without a camera it would be impossible to know the actual position or rotation that the ROV has. The camera view section connects directly with a raspberry pi camera [13] that is recording at a real time. In the website this recording is visible, and it is at real time allowing the user to see exactly what the ROV sees at every single moment. Then we can run the python3 camera code to show the camera capturing video. All the data and signal are being sent through the ethernet cable (Cat6) which is faster than any other type as it transferers 1Gigabyte/sec.

L. Communicating with Arduino [I2C Protocol]

Both the Arduino and the Raspberry Pi support I2C [11], however interfacing them can present a special challenge as they don't work at the same logic voltage levels. The Raspberry Pi uses 3.3-volt logic, whereas most Arduino's (including the Arduino Uno) make use of 5-volt logic. However, despite these voltage differences it is possible to interface the two devices. Before we examine the issues with mixing multiple I2C devices of different logic-levels it would be a good idea to make sure that we are familiar with the i2c as in [12] which operate them together.

M. Control Using LabVIEW

LabView is developed by National Instruments. A program in LabView is called a VI, which stands for Virtual Instrument. To create a VI (Virtual Instrument), LabView programming environment can be used. The user interface can be made by simply dragging and dropping objects, and arranging them in a required pattern to complete the path. Multiple functionalities can be added to the interface in the form of a diagram, similar to a flowchart, "wired" with the various structures and functions. So, in most LabView programs, no lines of code are written, the functionality of the program is provided by the diagram. That is why LabView is called a graphical programming language. LabView is an extremely powerful graphical programming system that is compatible with data collection and storage. People use it because it is convenient and no knowledge of conventional programming languages is required. Because of the appearance and physical instruments used in it, LabVIEW programs are called virtual instruments, or Vis. For data acquisition including data storing, data sorting, data analyzing and displaying LabVIEW contains a comprehensive set of tools. as well as it has tools to help you debug your code. An interface can be built on block diagram, and results will be displayed on front panel using controls and indicators. Buttons, potentiometers, knobs, and other input mechanisms are included in controls. Indicators include charts and graphs from oscilloscopes, LEDs to represent on or off, and other output displays. The block diagram contains a code, the output of which is available on front panel once you run the code. Once the front panel is displayed, you add code using VIs and structures to control the front panel objects. We can also communicate our LabView program with hardware such as we can acquire data from hardware and process it in our program, vision and motion from cameras and motion control devices.

Once you get done with the downloading and installation as in [15] part of LabView it will appear in you search bar. Depending upon the requirement of the project you can create any of the templates already provided by NI and then click finish, but according to our current requirement we will open Blank VI. LabVIEW displays two windows: the front panel window and the block diagram window both of them are explained below.

The graphical source code is to be built in the block diagram panel, also known as G code or block diagram code, for how the VI runs. Graphical representation of a code is known ad block diagram code. This code uses graphical representations of functions and includes inputs and controls to control the objects in the front panel. Objects on the Front panel appear as icon terminals on the block diagram. Controls and indicators are connected via wires and they collectively perform a specified task. Data transfers from one control unit to other via wires in the order given below:

1) from controls to VIs and functions

2) from VIs and functions to indicators

3) from VIs and functions to other VIs and functions.

The movement of data through the nodes on the block diagram determines the execution order of the VIs and functions. This +-movement of data is known as dataflow programming. The user interface in LabView is known as front panel of a VI. Combination of controls and indicators results in the formation of front panel, where controls are the input and indicators are the output terminals of the VI. We can select the required controls and indicators from the control palette. Push buttons, knobs, potentiometers, dials, and other input mechanisms are controls. Input data is simulated by the controls and then they supply data to the block diagram of the VI. Indicators include charts and graphs, LEDs, and other displays. Instrument output mechanisms are simulated by indicators and display data the block diagram acquires or generates.

Running a VI using the run button shown in the figure below or by pressing Ctrl+R executes the solution of the VI. During runtime the run button on the block diagram changes to a darkened arrow. Using the stop button the execution can be stopped during runtime, this button will stop running the VI after completing its current iteration. Using the pause button will pause the execution of the code after the completion of current iteration and pressing the pause button again will start the execution from where it was being paused

CONCLUSION

Based on today's explorations in seas and rivers, using robots become essential due to their features as they can perform underwater tasks in a very accurate level compared to humans' performance. The ROV design was done to reach up to 50m underwater, which can solve the problem of divers who can only reach 30m. The motion control, 3D position, and video capture are performed in parallel and they are processed by a main algorithm that was programmed to use connection between Raspberry Pi 4 and Arduino. The flexibility of mechanical design and low-cost hardware increases potential applications. The ROV proved to be capable of completing all task within a timely manner, and the team feels confident to make such a great project like that.

ACKNOWLEDGMENT

Thanks, and gratitude to Allah who gave us the power, patience and motivation to complete this project. We would like to express our thanks to the faculty staff of Mechanical and Mechatronics department for their support. First of all, we would like to express our appreciation to the head of Mechatronics and Robotics department Dr. Ali Ahmed Younis and our supervisor Dr. Khalil Ali Khalil Special thanks to Assiut university and ASRT for their efforts and Support. Finally, yet importantly, we owe our colleagues and friends special thanks for their love and support.

REFERENCES

[1] Design and implementation of a remotely operated vehicle for marine exploration and archaeological research, *Faculty of Nàutica of Barcelona, Barcelona, 5 October, 2019*

[2] Design and Construction of an ROV for Underwater Exploration, Article, Published: 6 December 2019

[3] Gland-PG7-IP68 | 3D CAD Model Library | GrabCAD. (n.d.). Retrieved July 16, 2019, from https://grabcad.com/library/gland-pg7-ip68-1

[4] T200 Thruster documentation https://www.robotshop.com/en/t200-thruster-no-esc.html

[5] Basic ESC Documentation (New) https://docs.bluerobotics.com/bescr3/

[6] Raspberry pi camera module documentation https://www.raspberrypi.org/documentation/hardware/camera/

[7] The ROV Manual A User Guide for Remotely Operated Vehicles Second Edition Robert D. Christ Robert L. Wernli, Sr.

[8] ROS/Network Setup http://wiki.ros.org/ROS/NetworkSetup

[9] Ubuntu install of ROS Noetic http://wiki.ros.org/noetic/Installation/Ubuntu

[10] Installing and Configuring Your ROS Environment http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment
[11] Basics of I2C communications https://www.circuitbasics.com/basics-of the-i2c-communication-protocol/
[12] Bi-Directional Logic Level Converter Hookup Guide https://learn.sparkfun.com/tutorials/bi-directional-logic-level-converter hookup-guide/all
[13] Opencv in Ubuntu using python https://docs.opencv.org/4.5.2/d2/de6/tutorial_py_setup_in_ubuntu.html
[14] 5 DC/AC Blown Fuse indicator circuit with LED Display https://www.eleccircuit.com/blown-fuse-indictor-led-display/
[15] Control a Servo Motor using LabVIEW https://www.labviewmakerhub.com/doku.php?id=learn:tutorials:libraries:linx:seeed-lotus-kit:control-servo-motor