ISSN NO. 2320-5407

Journal homepage: http://www.journalijar.com
Journal DOI: 10.21474/IJAR01

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH

RESEARCH ARTICLE

GRAPHICAL USER INTERFACE FOR AUTONOMOUS SURFACE VEHICLES.

Nitin Dabholkar, Dr. B. T. Salokhe and Narayan Satelkar
Tatyasaheb Kore Institute of Engineering and Technology, Warananagar, Kolhapur.

Manuscript Info

Manuscript History:

Received: 11 April 2016
Final Accepted: 13 May 2016
Published Online: June 2016

Key words:
monitor, sub system, mission, RF Modem, Wireless LAN, base station, communication

***Corresponding Author**

Nitin Dabholkar.

Abstract

The operation of autonomous vehicles of every type land, sea and air requires control and continues monitoring of sub systems parameters and mission parameters is very essential[1]. The monitoring of such vehicles is done using a RF Modem or Wireless LAN. The autonomous vehicles acquires and records the parameters of sensors which are required for navigation and control of the vehicle. The parameters as recorded by the system varies from system but some common parameters includes Position from the Global positioning system GPS, pose from the Attitude Heading Reference system (AHRS), the scientific data along with the health of the System. These parameters are usually updated on the base station which could be a mother-ship or Land. In this paper we describe the Graphical user Interface developed for the Autonomous Surface Vehicle and the communication protocol between the surface vehicle and the base station.

Introduction:-

The GUI plays an important role not only in monitoring of the system but also in controlling the vehicle as per the user requirement. Most of the vehicles which are commercially available perform mission and the control of such vehicles is assumed by a user, by means of (GUI) which could also have a joystick as an interactive device[2]. The communication between the two is generally by means of RF Modem, Wireless LAN or Satellite Phones. The satellite phones are generally used in system which has endurance of few months or the communication is used for updating position for recovery otherwise the communication is mostly restricted to wireless LAN and RF Modem. The GUI designed is using the tools and packages of visual studio which could be deployed on windows based system. The communication between the GUI and the underwater vehicle is full duplex wherein commands and status is exchanged[3]. The GUI also makes use of KML file generated by the Google Earth software for generating the mission.

ASVs are robots which are designed to navigate autonomously along programmed transects on the sea surface, holding course with good accuracy in the face of sea currents and wind. ASVs are not designed to dive below the sea surface, and are therefore much simpler data platforms to design and construct than AUVs. The surface vehicle is propelled by means of twin thrusters which uses of the differential thrust for turning or Heading control. The communication of the ASV is on RF-Modem and Wireless LAN. The user can switch over between the two channels as and when required. Normally, the priority is given to the Wireless LAN as the bandwidth is on a higher side however the range is limited as compared to the RF-Modem. The communication software is implemented as such that the protocol is not aware of which communication system is used.

Hardware architecture:-

The Hardware architecture of the ASV is depicted in block diagram in Figure 1. The block diagram can be visualized in two parts ASV hardware and Science Node. The ASV hardware like many surface vehicles consists of a RF modem and a WIFI to communicate to the base station and a GPS for reference positioning and an Attitude, Heading Reference system (AHRS) for knowing the orientation of the vehicle. The ASV hardware uses the AHRS and the GPS for Guidance i.e. to move in a programmed transact from one point to another. The main computer of the ASV has a communication port which enables it to communicate to the science node which is explained in detail below. The Dissolved oxygen (DO) Sensor is attached to the digital port, and a provision is kept on the hardware to accommodate two more digital sensors. The spare digital ports as depicted on the block diagram can be interfaced to sensors like Chlorophyll and Turbidity and Conductivity, Temperature and Depth (CTD). This methodology of interfacing sensors verified the functioning of the algorithm just by modification of the configuration file. One of the COM port is connected to the main computer of the ASV to transmit and receive data and commands for operation. The hardware also has provision to interface analog sensors for future expandability. The data is stored in files on the internal flash memory. The web camera is interfaced on the USB bus, enabling live feed–through the WIFI network.

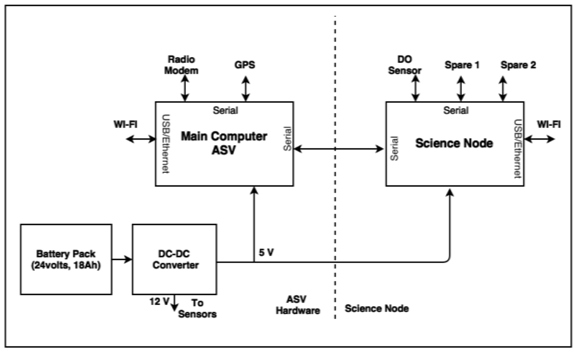

Figure 1:- Block Diagram of ASV.

The platform used "BeagleBone Black" is a low-cost, community-supported development platform for developers and hobbyists[4]. The BeagleBone Board is a low-power open-source hardware single-board computer produced by Texas Instruments. This Board was also designed with open source software development in mind. The board is based on Cortex-A8 32 bit processor operating at 1 GHz and has 512MB of RAM. It has a 4 Gb on board flash storage. Other peripherals connectivity includes USB host, Ethernet, HDMI and 96 pin header which could be programmed for various functionality.

Software implementation:-

The ASV software has been designed using the visual studio package and is compatible with windows based operating system. ASV System software design comprises of the following modules:

1. Main GUI module.
2. Connection Module.
3. FTP Module.
4. Controller Settings Module.
5. Mission Creation Module.
6. Guidance and Navigation Display Module.
7. Vehicle status and scientific information module.

Communication and Control of the ASV in a user-friendly way is a must if the ASV is to be operated by users from fields other than engineering. In order to allow users a very user-friendly and intuitive interface we have developed a Graphical User Interface from which it will be possible to control the ASV and upload/download data.

A screenshot of the ASV_GUI program is shown below in Figure 2. The ASV_GUI program keeps getting updated with information from the ASV when the radio-link between the ASV and the PC is available. The current position of the ASV along with the vehicle and the scientific payload sensors gets updated on the main page.

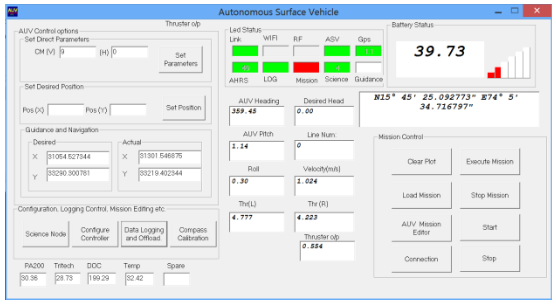

Figure 2:- Graphical User Interface.

The GUI also contains a battery monitor control. This battery monitor control indicates the status of the battery bank on the ASV in the form of led status. The color of the led turns RED informing the user to take preventive action.

Besides this other important information that is either critical to the functioning of the AUV or is very useful to the user without the user having to look for exact values is displayed using the LED controls. In case a sensor is malfunctioning that LED control will turn "Red" instead of "Green". Other colors are also used to indicate other states, such as "Blue" when a mission has been loaded by the AUV_SERVER. The buttons on the main GUI with functionality is as under:

1. **Connection:** To establish the connection with server running on ASV.
2. **ASV Mission Editor:** User create mission by giving required parameters.
3. **Data Logging:** For configuring the data logging of various vehicle and scientific sensors
4. **Configure Controller:** This is to configure all controllers with different parameter
5. **Start ASV:** Before starting any operation user need to start the ASV. For this connection should be established with server.
6. **Stop ASV:** Stop the ASV.
7. **Execute Mission:** It will start the mission on ASV. User must confirm that he has offloaded a mission on server.
8. **Stop Mission:** If any problem occurs in between mission execution, user can stop mission at the moment.
9. **Set Parameter:** User can set parameter any time by this option. If user set these parameters in between mission, it will stop the mission and will work in open loop.
10. **LEDs:** LEDs show the status of different hardware and sensors.

Client/Socket Connection:-

The following dialog as sown in Figure 3 is used to begin communication with the ASV. The IP address to connect on WIFI is 192.168.0.20 and on the RF is 192.168.2.1, and the ASV_SERVER is waiting on Port 4000.

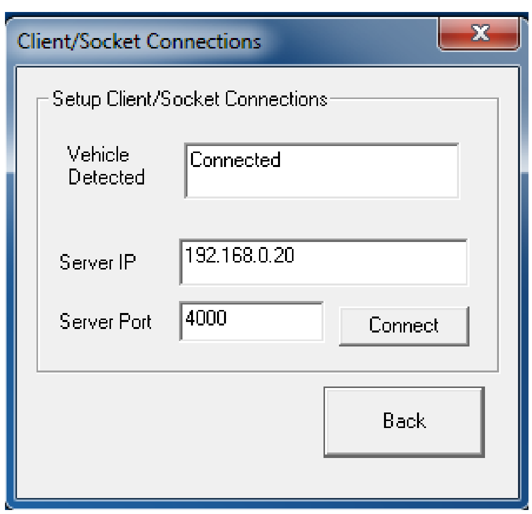

Figure 3:- Client/Server Connection.

This communication connection uses TCP-IP sockets over a PPP link and wireless LAN. This is expandable to any form of communication which could even be an acoustic, other link. However before attempting connection it is necessary that the user has established this PPP link or WIFI connection with the ASV.

Data Logging.

The user reaches this dialog if he/she chooses to perform some data-logging of specific parameters.

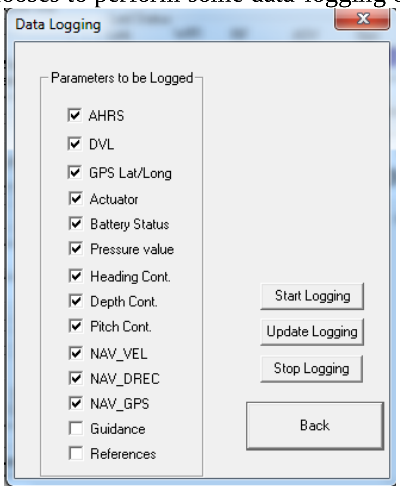

Figure 4:- Data Logging.

The Controller Configuration Dialog:-

The ASV autopilots and Guidance can be controlled directly from the ASV_GUI.

Here it is possible to change gains of the PID controllers and also to change the basic structure of the PID controllers. Acceptance radius for the Way-point guidance controller can also be changed here.

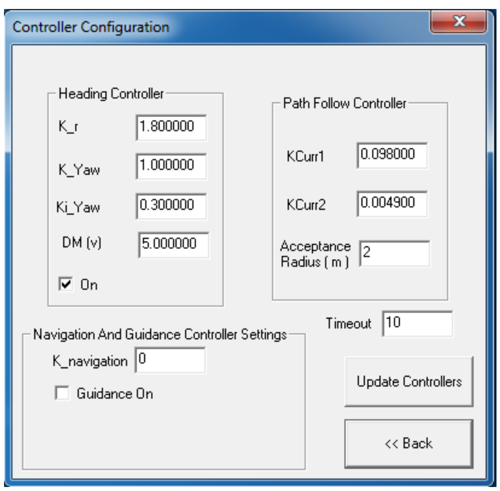

Figure 5:- Configure Controller.

The Mission Editor Dialog:-

We also have a mission editor dialog (Figure 6) where it is quite simple for the user to create a mission for execution on the ASV. All the primitives that can be used to create mission files can be selected from the Drop-down Menu. According to the number of parameters that each command takes, the edit boxes beside the command box automatically become editable. The user can then type in values that correspond to the amount he/she wants to set and then add or edit commands. The entire Mission File is listed in the List-box.

The user can also Load/Save Mission files or upload them to the ASV using the File Upload/Download dialog.

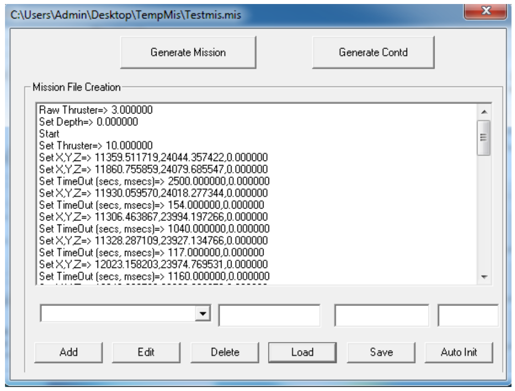

Figure 6:- Mission Generation Snapshot.

Future developments:-
The present GUI control a single vehicle at a time for its operation. Provision to control many such surface vehicles from the same GUI is being explored.

Conclusions:-
The data transfer between the GUI and the surface vehicle is updated every one hertz. The GUI allows to visualize the parameters along with the update rates of the sensors and a provision for building a mission. The GUI updates the KML file which is used by Google earth for updating and displaying the track of the vehicle.

Acknowledgements:-
The authors would also like to thank the professors and the Principal of Tatyasaheb Kore Institute of Engineering and Technology, Warananagar-Kolhapur for their continues encouragement and support.

References:-
1. Autonomous profiling device to monitor remote water bodies, Madhan, R.; Dabholkar, N.A.; Navelkar, G.S.; Desa, E.S.; Afzulpurkar, S.; Mascarenhas, A.A.M.Q.; Prabhudesai, S.P., Curr. Sci.: 102(2); 2012; 169-176.
2. Graphical User Interface Design Essentials & Process, by Reena Saini, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 9, September 2013
3. Exploring the Design Space for Adaptive Graphical User Interfaces by Krzysztof Z. Gajos, Mary Czerwinski , Desney S. Tan and Daniel S. Weld
4. https://en.wikipedia.org/wiki/BeagleBoard