The ROV communication and control

Alin Ghilezan
Electronic and Telecommunication Department,
Constanta Maritime University, UMC,
Constanta, Romania
alin.ghilezan@yahoo.com

Mihaela Hnatiuc
Electronic and Telecommunication Department,
Constanta Maritime University, UMC,
Constanta, Romania
mihaela.hnatiuc@cmu-edu.eu

***Abstract*—The opportunity for the future engineers for direct hands-on work in projects and the liberty of trying their solutions and to apply personal ideas with a large variety of electronic components, it will lead to a progress in everyday life for the future generations. In this paper we present a remote controlled ROV (Remotely Operated Vehicle) based on Arduino Mega 2560 microcontroller which acquired the sensors data and controls the direction and acceleration. For the ROV localization it is used a Global Positioning System (GPS) module and an Inertial Measurement Unit (IMU) which provides the orientation in space and the heading as a deviation from the magnetic North. It connects trough WiFi at the Access Point which is identified by the name of the network and a password. The Access Point sends a string that consists of movement data and waits for a response from the ROV. In the message that the ROV sends are information's about the speed, latitude, longitude, an estimated roll, pitch and yaw. The ROV is controlled using LabView and Processing interfaces. A comparison of the results between the two interfaces and the type of data communication is presented.**

***Keywords*—ROV, communication protocol, data frame, interface**

I. Introduction

The new Internet of Thinks applications are remote control systems using ROV equipped with different specific sensors which communicate with other devices or users [1, 2, 3, 4, 5]. The system can be remote manipulated for fulfill different tasks demand by the user. The different solution of communication between ROV and command platform are tested using wire or wireless transmission [6, 7].

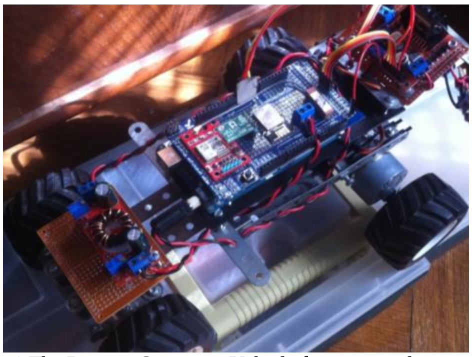


Fig.1 The Remote Operating Vehicle for terrestrial missions

The system presented in this paper (Fig. 1) is a practical achievement of the remote control vehicle ROV used for the different types of inspection and control in an unapproachable space. The ROV system applications which demand remote controls are the pipe inspection on the ship, risk missions or explorations inspections [8, 9]. The remote control system developed in our laboratory is designed in many stages. We are tested different type of command, control and communication. The ROV system is considered as the client (Slave) The first control of the ROV system is built with a microcontroller, Arduino card which is defined as Server (Master). This server communicates with a computer using a human machine interface. The second solution designed is to control the ROV system directly with the computer using human machine interface. Each method is analyzed function of the application and the software complexity.

II. System Description

The remote operated vehicle system is divided and tested in two sub-systems: the ROV system (client) and the command unit (server) (Fig.2). The ROV is equipped by Arduino Mega card which acquire the date of GPS module, IMU sensors, WiFi module and control four motors. The motors are commanded by DC motor drivers. The command unit is a computer which has connect a joystick. This unit establish the acceleration an orientation of ROV and receive the sensors data for processing. For the data transmission between these two sub-system are tested different communication protocols, implemented on the microcontroller and computer. As the human machine interface are tested two interfaces programmed in Processing and LabView 2013 software.

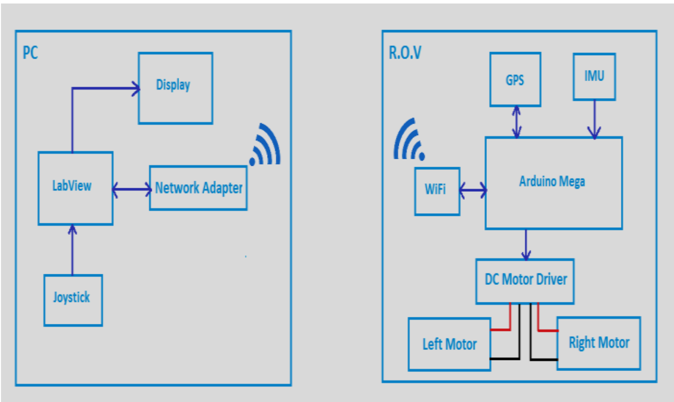


Fig.2. The block diagram of the remote controlled ROV system

The ROV components is powered from 7.2V and 2200mAh Lithium Polymer battery using a DC-DC converter which lowers the voltage to 6V (Fig.3).

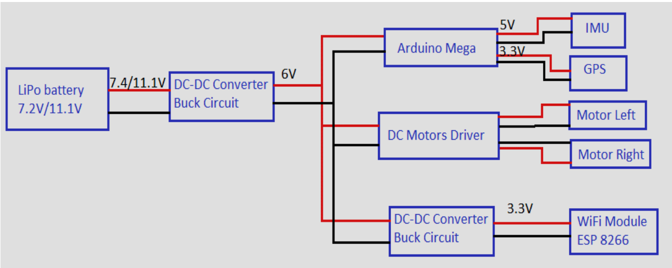


Fig.3. The power supply system for the ROV

The Arduino pins can provide up to 40mA so we can power the IMU and GPS modules directly from the 5V and 3.3V pins, but for the DC motors and the WiFi module it's necessary external power supply. Motor Driver use the same voltage as Arduino board. An DC-DC converter reduce the voltage from 6V to 3.3V (0.7A) for the WiFi module.

A. The system components detalies

The compatibility and the real time response are tested for the components used in the system.

The remote control communication is establish using WiFi connection generated with ESP 8266-12E module. This module communicates with Arduino by UDP connection.

The orientation is established with GPS module of the ROV - GYNEO6MV2 with mini antenna. Using TinyGPS++ library written for Arduino we can interrogate the GPS about latitude and longitude coordinates and speed using build-in functions.

The Inertial Measurement Unit of the ROV is an LSM6DS33 with 3-axis gyro, 3-axis accelerometer and an LIS3MDL 3-axis magnetometer onto a tiny 0.8″ × 0.5″ board. By I²C interface accesses nine independent rotation, acceleration, and magnetic measurements. These are used to calculate the sensor's absolute orientation. The MinIMU-9 board includes a voltage regulator and a level-shifting circuit that allow operation from 2.5V to 5.5 V. The data frame from the accelerometer, magnetometer and gyroscope are fused together by an Open Source algorithm provided by the producer.

The DC (Direct Current) motors are controlled using a dual H-bridge Motor Driver based on L298N chip.

Arduino Mega 2560 has 4 hardware Serial ports, two of them are used for Serial communication with the other modules (Serial 3 for WiFi and Serial 2 for GPS) and one is used for debugging and programing (Serial 0). Arduino card allows to achieved software serial ports, but we noticed more errors than on the hardware serial.

B. Communication and interfaces software description

In this case, the accuracy of communication is important. In this regard, several types of communications were tested, tracking the transfer time, errors, compatibility between the transmitting / receiving devices.

The first communication implemented has been between two Arduino using Ethernet Shield. The data transmission used TCP/IP protocol. One of microcontroller cards is Client which read the data of IMU sensors and the other microcontroller is server which read the joystick data. The Arduino server transmitted the data of the client to the computer using the USB communication.

The second communication tested the direct transmission between Arduino and computer using UDP protocol. The data are send to the computer using UTP (Unshielded Twisted Pair) cable, so a LAN (Local area network) network are established between the devices. The logic address allocated must be of the same IP class. The initialization of Ethernet shield is similar of the TCT/IP protocol but it is necessary to create a UDP instance and to declare the port which receive the data.

The last communication tested is between two microcontroller using RS232, first we tested the transmission between two Arduinos and then between Arduino and WIFI module. Then WiFi module send the message received of Arduino to the computer using UDP protocol.

Using many different sensors connected to Arduino client demand a resources organization (Fig. 4). Arduino is not build for multi-tasking, but the tasks can be synchronized using time period. Using millis() function in the main loop of Arduino software it can be organized to run certain function on different time intervals.

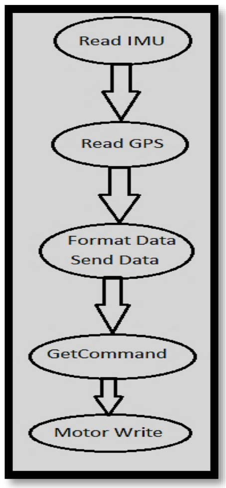
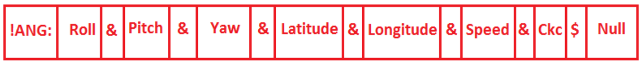


Fig. 4. Arduino flow chart

The IMU sensor subroutine achieves a double integration function of time to determinate the ROV rotation. This time interval it is the same with the execution subroutine period to read the values (200ms). The subroutine for motors controlling is executed permanently for ROV displacement. The execution subroutine for data reading of location command and transmission data to WiFi module are achieved simultaneously because it is possible to read and write to the serial port in the same time

In the testing step as remote control we developed an interface in Processing to the computer, that reads the analog values from a PS4 joystick connected on the USB port. The values are converted into a string, initiate an UDP connection with the ROV IP on the port 1001, sends the data string and reads UDP messages sent from ROV. The messages that are exchanged have a standard format with known header, separators between data, checksum and a special character at end. The header and the checksum are used to validate the data at the reception. The NULL character of the end communication is used in the communication between the WiFi module and Arduino.



The checksum is determinate using XOR function on each element of the string (without header) with the previous XOR result. So the checksum value is smaller than the sum of all the elements in the string and the separators „&" don't influence it.

The received message in remote control server consist in the coordination of the IMU sensor and GPS. The message sent by remote control using joystick values control the motors for establish the acceleration and direction (Fig.5).

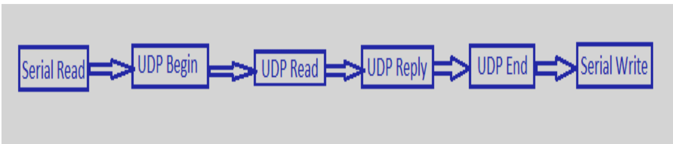


Fig.5. WiFi module flow chart

The WiFi module code reads orientation data on the serial, initiates the UDP connection, waits the command and replies with the orientation data. The module connects by HotSpot made from computer check the ID and Password of the network, than read UDP messages from a local port, gets the IP of the sender and replys to the same port and IP.

The ROV control by the user is achieved using interfaces. We build two interfaces in different programme to compare the efficiency and the facilities which it offers. One of them is built in Processing using JAVA software. The command data is read using gamecontrolplus.gui library which can read the joystick data connected to the USB port of computer. The data formatting is achieved in the same way as Arduino software. The data received in the interface are the position of the ROV.

The second interface tested is build LabView. The interfaces showed the position of the ROV and localisation on the map (Fig.6). On the Front Panel is display the joystick data, speed of ROV in clock indicator, a map localization indicates by GPS, the 3D position representation of the system indicated by IMU sensor and an artificial horizon. The interface has the possibility to write the IP address and to save the data in Excel.

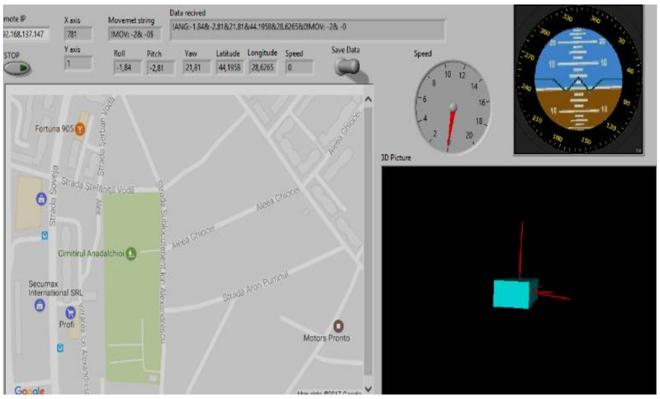


Fig.6. The ROV coordination showed using an interface using LabView

The artificial horizon is build using virtual instrument tested by National Instruments. This vi takes Roll, Pitch and Yaw data and it is used to rotate the framework.

The joystick communication is build using Input Device Control Function of LabView (Fig.7). The data are calibrated on the Y and Z axes and then converted in String with a dimension which can be choosed. The two string obtained of the two axes must be concatenated to respect the same structure described above.

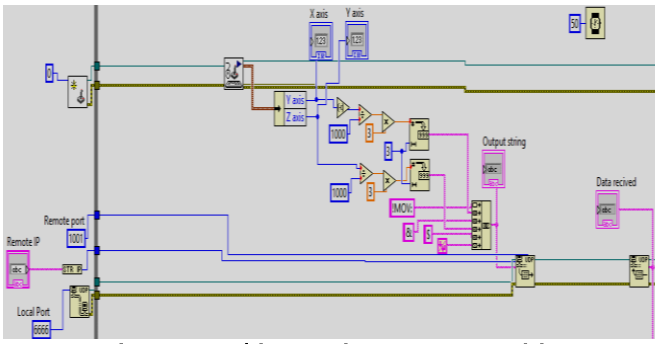


Fig.7. Bloc Diagram of the Joystick communication and the UDP initialization.

The UDP connection are build using UDP Function. With UDP Write virtual instrument it is send the data frame to the IP address assigned by the computer at the HotSpot conection and the ROV port defined in the program structure which run in WiFi module. The IP address it is written in Front Panel. Using UDP Read Function is read the ROV message and the message validation is checked comparing the header of the frame. The strings frame is divided in sub-strings which represent Roll, Pitch, Yaw, Latitude, Longitude and Speed, values used for orientation. The separator character is introduced in Match Regular Patern virtual instrument, which it is used for the header. The wrong reading is displayed as "0", so in each array there is a condition test using Index Array. A stack FIFO (First Input First Out) type is build. This has in the first position (most recent) a value different of 0.

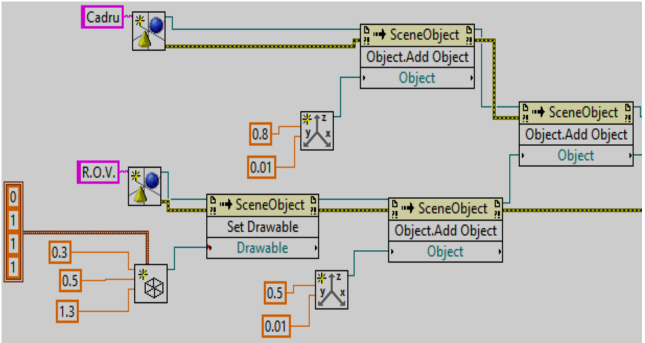


Fig.8.The Bloc Diagram of the 3D representation

The 3D representation is build using 3D Picture Control function (Fig.8). In the framework there is a coordination axes and a rectangular object which we can change the color and the dimension. The object can be rotate in three axes using Roll, Pitch and Yaw. The variables values must be converted in Radian to be compatible with Rotate Object function.

The map is build using a virtual instrument of National Instrument which made a connection to Google Map online. The data are saved to the computer using Write To Measurement File which it is introduce to the Case structure, so in the same mission we can have many files (many stages of the mission).

III. COMMUNICATION RESULTS

The message transmission integrity in TCP/IP protocol is identified in long distances cable. The disadvantage of this communication is the server and the client synchronization. This means, the client must reconnect to the server at each iteration and the server must be active all the time. This communication it is not very fast.

The UDP communication has the following advantages: better speed of transmission than TCP/IP and the link between the server and client is always active (this is a asynchronous communication) . The integrity of the message transmitted is not checked by the protocol this is a disadvantage of UDP communication. This disadvantage is eliminated using a software algorithm for the validation of message.

The RS 232 communication has a fast transmission, can be used in real time application and can be achieved between many devices. As disadvantage of RS232 communication is the limited cable length. Also at higher baud rate, the number of errors increase therefor it is necessary to find the optimal baud rate value. To eliminate these problems, it is used RS485 transmission protocol. This protocol allows a communication in long cable, shielded and twisted. This method is used in industry or underwater ROV [10, 11].

In point of view of human machine interfaces tested in this project Processing software offer the advantage of easier algorithm implementation but LabView software is oriented to the real time application and many of the Virtual instruments are open source and tested by Nation Instruments.

CONCLUSION

Three protocols are tested in this system to find the optimum and accuracy transmission methods between the sensors module and acquisition cards. Each protocols tested are compatible function of application difficulty and complexity.

The ROV sub-system can be controlled using a computer or Arduino card, depending of the environment where it is used. The interfaces used for control and communication are developed using open source function.

In hardware point of view, Arduino Mega are used to control the ROV because has an open source software, it's friendly used and it has enough input/output pins. Arduino has a big disadvantage: for complex application which demand complex algorithms, the microsystem has not enough memory and it isn't able to process in pipeline.

ACKNOWLEDGMENT

We would like to express our special thanks to APTE for sponsorship.

REFERENCES

[1] Zanella A., Vangelista L., "Internet of Things for Smart Cities", IEEE Internet of tThings Journal, VOL. 1, NO. 1, 2014

[2] Nefian A. V., Bouyssounouse X. Edwards L.,Kim T., Hand E., Rhizor J., Deans M., Bebis G. , Fong T., "Planetary rover localization within orbital maps" Image Processing (ICIP), 2014 IEEE International Conference on Image Processing (ICIP), ISBN: 978-1-4799-5751-4, 2015

[3] Chiuchisan I., Chiuchisan I., Dimian M., "Internet of Things for e-Health: An approach to medical applications", Proceedings of the 2015 International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM), 2015, pp. 1-5.

[4] Chiuchisan I., Costin H., Geman O., "Adopting the Internet of Things Technologies in Health Care Systems", Proceedings of the 2014 International Conference and Exposition on Electrical and Power Engineering (EPE), Workshop on Electromagnetic Compatibility and Engineering in Medicine and Biology, 2014, pp. 532-535.

[5] Brassitos E., Dubowsky S.,"Compact drive system for planetary rovers and space manipulators"2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Pages: 664 - 669, 2015

[6] Daigle M., Kulkarni C.S., "A battery health monitoring framework for planetary rovers", 2014 IEEE Aerospace Conference Pages: 1 – 9, 2014

[7] Yu C., Ju H. , Gao Y., Cui P., "A Bilateral Teleoperation System for Planetary Rovers", 2009 International Conference on Computational Intelligence and Software Engineering, Pag. 1 - 5, DOI: 10.1109/CISE.2009.5363354, 2009

[8] Anwar I., Mohsin M.O., Iqbal S.; Abideen Z. , Rehman A.; Ahmed N., "Design and fabrication of an underwater remotely operated vehicle (Single thruster configuration)", 2016 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Pages: 547 - 553, DOI: 10.1109/IBCAST.2016.7429932, 2016

[9] Conte G. , Orlando G. , "A variable structure control of a remotely operated vehicle", OCEANS '94. 'Oceans Engineering for Today's Technology and Tomorrow's Preservation.' Proceedings, Volume: 3, Pages: 111 - 115,1994

[10] Lazar I., Ghilezan A., Hnatiuc M., "Development of underwater sensor unit for studying marine life", 2016 IEEE 22nd International Symposium for Design and Technology in Electronic Packaging (SIITME),Pages: 82 - 85, DOI: 10.1109/SIITME.2016.7777249, 2016

[11] Lazăr I., Ghilezan A., Hnatiuc M., "Development of Tools and Techniques for Monitoring Underwater Artifacts", Proc. SPIE 10010, Advanced Topics in Optoelectronics, Microelectronics, and Nanotechnologies, vol.VIII, ISBN:978-1-5106-0424-7; ISSN: 0277-786X, 2016