DESIGN AND DEVELOPMENT OF PRESETTER SOFTWARE AND MIL-STD-1553B REMOTE TERMINAL SIMULATOR FOR HEAVY WEIGHT UNDERWATER VEHICLE

[1]Shaik Khadar Bibi, [2]Mr. K. Siva Kumar, [3]Mr.N.S.S.S. Girish Kumar, [4]Mr. Arvind Sastry
[1] PG Scholar, [2]Scientist – E, C.S.S Division, NSTL, [3]Assistant professor, [4]Scientist – C, C.S.S Division, NSTL
[1]Department of Computer Science and Engineering (CSE)
[1]Gayatri Vidya Parishad College of Engineering (A), Visakhapatnam, India

Abstract: The heavy weight underwater vehicle consists of various sub-systems such as recording system to generate alarms and record data, homing system to detect the targets, motor and battery for propulsion, actuation system for rudder control, and gyro system to measure own position and attitudes of the vehicle. In underwater vehicles, it is very essential to establish the communication between different subsystems to transmit the critical parameters. The interface between subsystems must be a network standard technology which is well defined with no loss of data, required speed, command response, less hardware for connectivity and ruggedness. The MIL-1553B standard provides the facility to combat this requirement. In the development stage of any embedded system for under water vehicle, it is required to test the system with different subsystems in the virtual environment. The main objective of this paper is to develop the MIL-STD-1553B simulator for recording system and to develop Presetter software using RS-422 interface, which is a standard for serial communication. To process the entire simulation and to retrieve data from the recording system and Presetter, a user-friendly GUI is developed using visual C-Sharp. The developed system can be operated either in fiber optic cable mode or autonomous mode based on requirement specification.

Keywords: **-MIL-STD-1553B, Simulation, RS-422 Interface, Visual C-Sharp.**

I.INTRODUCTION

In application areas like under water vehicles, integration of flight control systems, missile interfacing, and telemetry etc., the system requires bulk range of signals to be processed, which includes inputs and outputs between different subsystems. Also, in most of the systems the data is recorded, downloaded and analyzed. The communication between systems is efficiently performed by RS-422 and MIL-STD-1553B with the former being most prevalent for overall system reliability and performance.

So far the vehicles that we have are autonomous underwater vehicles (AUV). That is we have to control and guide the vehicle autonomously. So if any high priority task is identified, we cannot communicate to the vehicle to perform the operation dynamically. To overcome this limitation, now the vehicle is designed in such a way that it can operate in two modes i.e. cable mode and autonomous mode.

The paper emphasizes on the design and development of presetter software and MIL-STD-1553B recording system using Microsoft visual c-sharp Integrated Development Environment (IDE). The developed system will operate in cable mode and if there is any communication breakage, then it can operate autonomously as per the preset parameters.

1.1. MIL-STD1553B Bus

The 1553 data bus is a dual-redundant, bidirectional, Manchester II encoded data bus with a high bit error reliability. It is developed due to the growing complexity of integrated avionics systems, ships, satellites, missiles, space station programs etc. Due to its deterministic and error-free communication, it is being used widely in most of the modern military aircrafts, launch vehicles and other aerospace vehicles. In MIL-STD-1553B bus, 31 devices can be connected, of which one device must be Bus Controller (BC) and other devices are Remote Terminals (RT).

There are three different types of bus elements, they are
1. Bus Controller (BC): It is a Master device, decides which RT can communicate on the bus and give commands to it.
2. Remote Terminal (RT): It is Slave device that responds to the controller commands.
3. Bus Monitor (BM): It is Slave device, receives bus traffic and extracts selected information by monitoring communication on the bus.

1.2. Architecture

The heavy weight underwater vehicle consists of a bus controller onboard system and six remote terminals they are Recording System (RS), AS (Actuation System), Gyro System (GS), HS (Homing System), Motor controller, WHS (Wake Homing System) attached to the bus. As shown in figure 1, all the bus communications are controlled and initiated by a main bus controller onboard system. Remote terminals attached to the bus respond to controller commands.

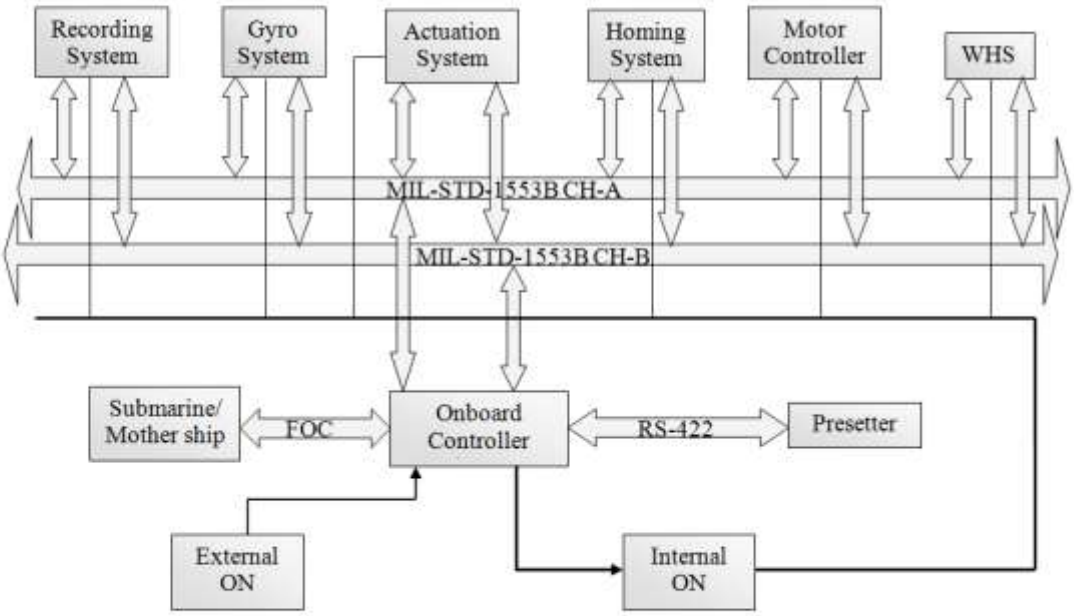

Figure 1: Presetter and Recording System (RS) interface with all the other subsystems in the vehicle

II.METHODOLOGY

The complete software is based on the foreground and background approach. All the interrupt service routines are executed in the foreground system, whereas main loop is executed in the background system. In foreground approach, two interrupt service routines are executed i.e., timer interrupt and bus interrupt.

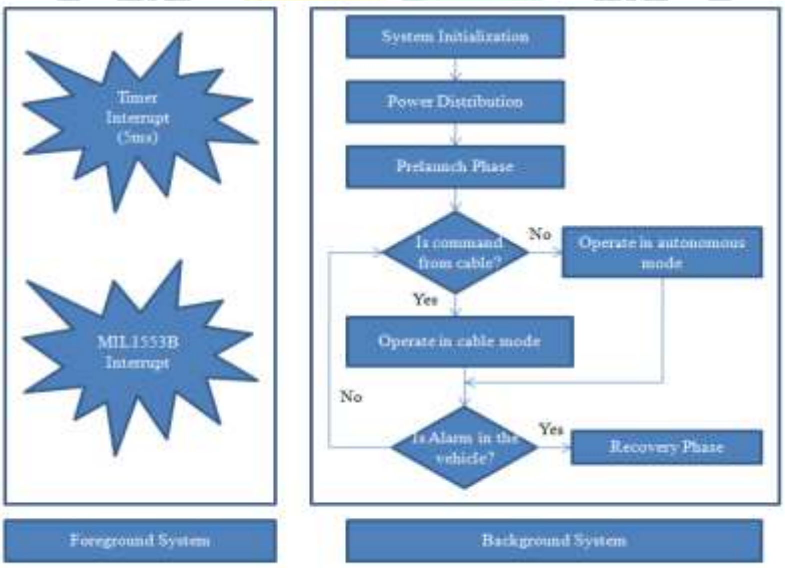

Figure 2: Concept of Execution

Figure 2 shows how the systems are designed and the concept of execution. In the foreground process when timer interrupt is generated for every 5ms then subsequent parameters in the GUI is updated and when the command is received from the bus then MIL-STD-1553B interrupt is generated. In background process, system initialization module performs initialization of peripherals i.e. timer, interrupt controller and serial controllers. Power distribution module distributes the power to all the sub-systems. In Pre-Launch phase, it checks the health of all the systems. If everything is fine then vehicle is launched and it can be operated either in cable mode or autonomous mode. In case of cable breakage, the mission operates autonomously as per preset parameters. RS records all critical parameters in NVRAM so that the data can be retrieved and later analyzed. If any abnormal condition occurs then alarm is generated by IDAS and, in few cases, the torpedo can be recovered.

III.DESCRIPTION

3.1. Presetter Software

The main functionality of presetter software is to initialize and guide the vehicle as per the mission requirements. In addition to that, it distributes the power to all the other subsystems in the vehicle such as RS, AS, GS, HS, Motor controller and WHS. It also performs prelaunch sequence and recovery operations. It operates in four different states.

3.1.1. System Initialization: - In this mode initialization of processor, serial port, timer and MIL-1553B interrupt is done.

3.1.2. Power Distribution: - This process distributes power to all the subsystems through relays.

3.1.3. Prelaunching Sequence: - Before dropping a missile, prelaunch sequence is done in order to perform a safe launch of a torpedo. This process has been further divided into eleven sub-processes as shown below.

3.1.3.1. Health Checks: Performs the health check of all the subsystems (RS, AS, GS, HS, MC and WHS), presetter and onboard controller. If any of the subsystem health is not ok then presetter will abort the mission.

3.1.3.2. Torpedo Configuration: To test the torpedo the bit is set to exercise so that the torpedo can be regained after testing. To launch the torpedo in real-time the bit is set to combat.

3.1.3.3. GPS Parameters: Presetter sends the GPS parameters to onboard controller and asks for validation of GPS parameter. Then onboard controller aligns the vehicle according to the given parameters and sends the GPS parameter echo to presetter.

3.1.3.4. Preset Parameters: Presetter sends the preset parameters to onboard controller and asks for validation of presetter parameter. The onboard controller sends the preset parameters echo to presetter.

3.1.3.5. Parameter ok: Onboard controller checks if all the parameters are within the required range or not and sets parameter ok and sends it to the presetter.

***3.1.3.6. RLG Alignment*:** Within 200 seconds RLG aligns the torpedo according to the GPS parameters. If the alignment is not done within the timestamp of 200 seconds then presetter will abort the mission.

3.1.3.7. Power Change over (PCO): If everything is ok, presetter changes the power from external battery to internal battery.

3.1.3.8. Heading Reference: Onboard controller acquires the heading reference from RLG and sends it to presetter.

3.1.3.9. Clear to Fire Command: Torpedo is fired when fire command is received from onboard controller. If fire command is not received and when time out occurs, mission will be aborted.

3.1.3.10. Stop Bolt Indication: Presetter sends a stop bolt indication to onboard controller and waits for an echo command.

3.1.3.11. Abort Operation: The presetter continuously checks for any run termination condition, and based on that, abort command will be generated. Run termination conditions include health status, alignment not received or any communication failure in the vehicle.

3.1.4 Recovery State: - There are two modes of operation exercise mode and combat mode. The exercise mode is used during testing and combat mode is used in real-time. The presetter is set to exercise mode so that the torpedo can be recovered after testing.

3.2. Recording System Simulator

In order to design and develop an embedded system for an underwater vehicle, there is a necessity to test the system with different subsystems in virtual environment. The main objective of this MIL-STD-1553B RS simulator is to display and record the data in NVRAM, so that it enables the users for further analysis. It also generates alarms like water contact (WC), leak contact sense (LCS), depth cut-off (DCO), cable cut-off (CCO) etc.

All bus communications are controlled and initiated by a main bus system onboard controller. Remote terminals such as RS attached to the bus responds to the controller commands.

The 1553 messages contain command word, data word and status word. The data portion has control commands and parameter values for different sub-systems, which are of utmost important for system performance analysis.

3.2.1. Alarms Generated by RS:-

After launching the torpedo, if any abnormal condition arises, then the RS generates some of the major alarms to indicate the problem to onboard controller. In case if any of the alarms is generated then the mission will be aborted. Alarms generated by RS are:

3.2.1.1. Water Contact (WC): This alarm is generated when the torpedo gets in contact with water.

3.2.1.2. Main Battery (MB): This alarm is generated if there is any problem in the main battery. The problem can be due to environmental conditions, physical component changes etc.

3.2.1.3. Exercise Battery (EB): This battery is used during testing. If there is any problem in the battery then this alarm is generated.

3.2.1.4. Depth Cut-off (DCO): This alarm is generated when the torpedo exceeds the depth cut-off limit given by presetter.

3.2.1.5. Ceiling Cut-off (CCO): This alarm is generated when the torpedo exceeds the range of ceiling cut off given by presetter.

3.2.1.6. Leak Contact Sense (LCS): This alarm is generated when there is any leakage of water into the torpedo.

IV.DESIGN AND IMPLEMENTATION

4.1. Presetter software

The presetter controls and guides the vehicle as per the mission requirements. During runtime presetter operates in twelve different states as shown in the figure 4.1.

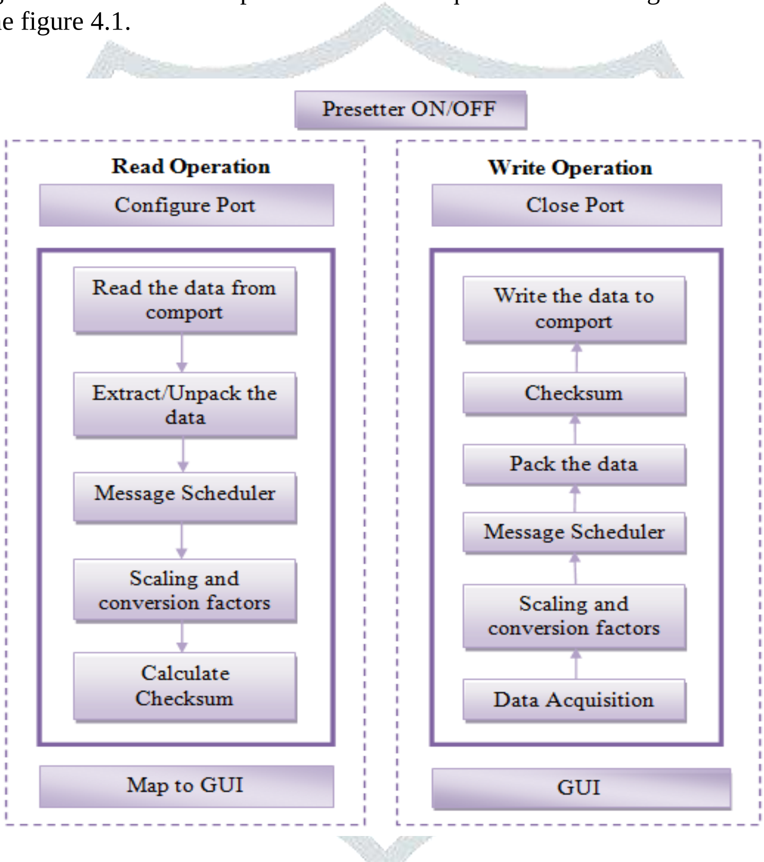

Figure 4.1: Implementation of presetter software

4.1.1. Configure Port:-

- The port settings like COM port number to 1, bits per sec to1920, data bits to 8, parity to none, stop bit to 1 are set.
- The port is opened by using method comport.Open ().

4.1.2. Read the data from port: - Entire data is read from the port buffer.

4.1.3. Unpack the data: - The data received from onboard controller is unpacked and required bits are extracted.

4.1.4. Message Scheduler: - All the messages will be scheduled as per the given dataset.

4.1.5. Scaling and conversion factor: - The data is scaled and converted into engineering units using a scaling factor.

Example: Scaling factor for heading reference is Heading value*180*2^{-15}.

4.1.6. Checksum: - For every message checksum is calculated, the packet is then retransmitted if there is any error.

4.1.7. Map to GUI: - The final result will be displayed on GUI.

4.1.8. Data Acquisition: - Data is acquired from onboard controller using RS-422 communication interface.

4.1.9. Pack the data: - Presetter packs all the relevant data and sends to onboard controller via RS-422 interface.

4.1.10. Write the data to port: - The packet is written to the port buffer as per prelaunching sequence.

4.1.11. Close port: - When the entire process is completed, the port will be closed.

4.1.12. Presetter On/Off: - When the port is open, presetter indication should be set to ON. If fire command or any abnormal condition occurs, it should be switched off.

4.2 Recording System (RS) Simulator

This Simulator is implemented by using USB 1553 avionics device and Acextreme SDK. The steps processed for implementing RS simulator is shown in the figure 4.2.

4.2.1. Device Configuration: USB 1553 Avionics Device (EmaceBU69092) is initialized and configured.

4.2.1.1. Initialize: The resources such as memory and register space for a particular mode of operation are initialized. It allows the memory on the USB 1553 avionics device to be accessed through a device driver. This memory can be selected by inputting ACE_ACCESS_CARD into the wAccess parameter of initialize function.

4.2.1.2. Configure device:

- The device is initialized to Multi-RT mode. Then the data tables are initialized to default values, and the memory structures are created. All RT sub addresses are illegalized after this function has been called.
- MT-I mode for a specific logical device number is configured.

4.2.1.3. Set Remote Terminal (RT) address: The remote terminal address is set in order to know where the packet is being sent or received.

4.2.1.4. Create and map sub address blocks: Data block for each and every sub address are created and mapped.

- **Create:** An RT data block is created and identified by the nDataBlkID input parameter. After the data block is created, it can be used by any sub address of different remote terminals.
- **Map:** The Data Block is mapped with one of the 32 sub addresses of the RT.

4.2.2. Device Start: When the device start function is enabled, all the required registers are aligned to enhancing mode code handling with the remote terminals operation being initiated. Then the enabled remote terminals will respond to messages on the 1553 bus. By this it transmits the device from ready state to run state.

4.2.3. Process 1553 Commands: The bus controller OC sends commands to the remote terminal RS for every time gap of 5ms. RS responds to the commands received from onboard controller.

4.2.4. Read from 1553 buffer: Read operation is done on the data present in the buffer of MIL-STD-1553 avionics USB hardware.

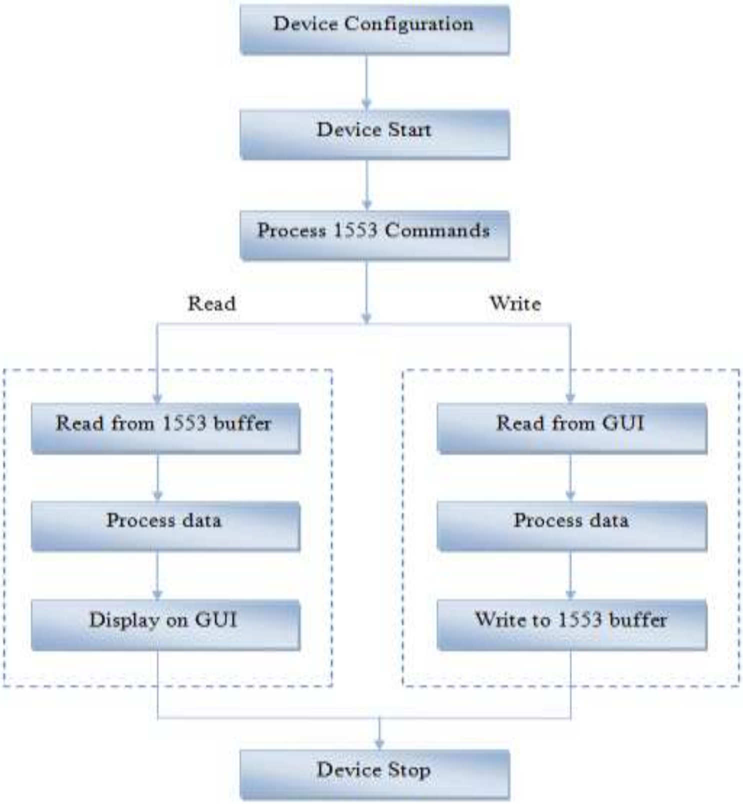

Figure 4.2: Implementation of recording system simulator.

4.2.5. Data pre-processing

The messages are scheduled (i.e. RT-RT, BC-RT and RT-BC).

4.2.5.1. Packet Parser: Packet parser time stamps each dataset, with time provided in the packet, or in absence of that with a local system time. the parser parses the received strings to extract data values.The following features are supported

- Binary or ASCII values.
- Big-endian of Little-endian for binary values.
- Double byte data (WORD).
- Signed or unsigned data.

4.2.5.2. Scaling and Conversion: Scaling factor is applied to the extracted data and then the value is converted to Engineering units.

4.2.6. Display on GUI: The GUI displays different observed parameters to the user.

4.2.7. Read from GUI: The messages are read from the GUI.

4.2.8. Write to 1553 buffer: Write operation is done on 1553 buffer as per the given dataset.

4.2.9. Device Stop: RT/MT-I mode is stopped from capturing messages and the device is set to Ready state.

V.Results

5.1. Presetter Software

5.1.1. RLG Alignment:-

The RLG alignment time starts from 200 seconds and goes on decreasing until alignment over is received from onboard controller. If not ,then the presetter will abort the mission.

Figure 5.1.1: RLG Alignment time and Heading Reference

5.1.2. Clear to fire:-

When the Clear To Fire command is set,then the torpedo gets fired.

Figure 5.1.2: Clear To Fire

5.2.Recording System Simulator:

USB 1553 Avionics Device (EmaceBU69092) is initialized and configured. RS responds to the onboard controller commands. It generates alarms

Figure 5.2.1:Device Configuration

Figure 5.2.2: Communication between onboard controller and RS

Figure 5.2.3: Acquiring different parameters from HS and AS.

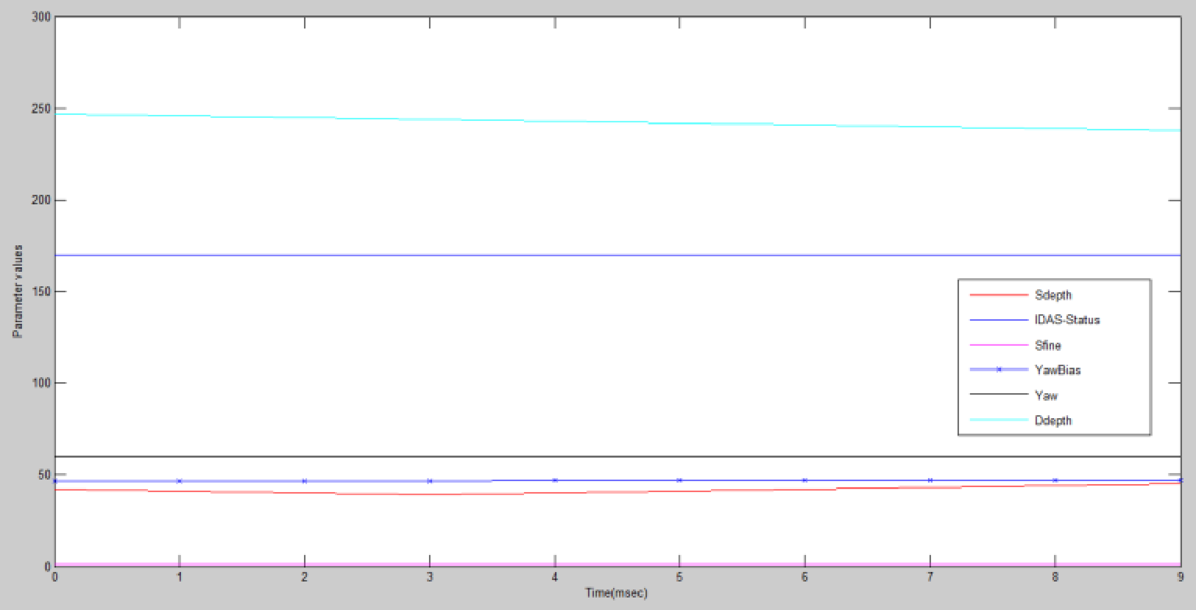

Figure 5.2.4: Plotting data in Matlab.

V.CONCLUSION

A Graphical User Interface which is developed using visual C-sharp software is user friendly and the process of data retrieval is very fast and easy. These systems can be used in under water applications.The proposed system operates in dual mode i.e. cable mode and autonomous mode.The presetter software is developed and tested in virtual environment using RS-422 interface. Simulation of recording system with different subsystem messages has been implemented and tested. The response to both the systems issued by the onboard controller in virtual environment using BU69092U USB module is simple and flexible when compared to the other simulation procedures.

VI.ACKNOWLEDGEMENT

This research work would have been incomplete, had it not been for the guidance of Mr. K. Siva Kumar, Scientist- 'E', CSS division, Naval Science and Technological Laboratory (NSTL). I would like to thank my guide Mr.N.S.S.S.Girish Kumar, without whose encouragement, this project would not have been a success.

VII.REFERENCES

[1] http://www.ddc-web.com/Documents/dguidehg.pdf, 6th edition by DDC, 2003.
[2] https://www.lammertbies.nl/comm/info/RS-422.html, Lammert Bies, 2017.
[3] https://science.howstuffworks.com/transport/engines-equipment/question419.htm, a division of InfoSpace, 2018.
[4] http://www.ddc-web.com/Products/196/Default.aspx, 1995.
[5] http://www.ddc-web.com/News/Press/Beta_3_Stub_Coupler.aspx, 1995.
[6] https://www.abaco.com/system/files/files/mil-std-1553_tutorial_a-wp-711b_0.pdf, Abaco Systems, 2016.
[7] https://rwhiston.wordpress.com/2011/10/13/9/, Robert Whiston weblog, 13 October 2011.
[8] https://www.encyclopedia.com/social-sciences-and-law/political-science-and-government/naval-and-nautical-affairs/torpedo-weapon, Encyclopedia.com. 18 May 2018.
[9] http://www.altadt.com/support/tutorials/mil-std-1553-tutorial-and-reference/, Alta Data Technologies LLC Rio Rancho, NM USA, 2017.
[10] https://www.computerhope.com/jargon/s/seriport.htm, computer hope, 30 October 2017.
[11] T.Hyakudome, "Design of Autonomous Underwater Vehicle", International Journal of Advanced Robotic Systems, Vol.8, No.1, pp.122-130, 2011.
[12] G.Mok Lee, "Development of an Autonomous Underwater Vehicle ISiMI6000 for Deep-sea Observation", International Journal of Geo-Marine Sciences, Vol.42, No.8, pp.1034-1041, 2013.
[13] K Siva Kumar, "Design and Development of Real Time Mission Software for wire guided underwater vehicle", International Journal of Engineering Research and Application, Vol.6, Issue.11, pp.69-71, 2016.
[14] Mario Prats, "An Open Source Tool for Simulation and Supervision of Underwater Intervention Missions", Intelligent Robots and Systems, Vol.7, Issue.12, pp.2577-2582, 2012.
[15] X.Wang, "Design and development of a rudder load simulator for the onshore testing of autonomous underwater vehicle", IEEE, Vol.18, Issue.21, pp.1-6, 2017.