

DESIGN AND DEVELOPMENT OF MIL-STD-1553 BUS MONITORING SYSTEM FOR AN UNDERWATER VEHICLE

¹PVSL Meghana, ²Mr.NVS Lakshmipathi Raju, ³Mr. B Suresh Kumar,
¹Student, ²Associate Professor, ³Scientist - D, C.S.S Division, NSTL
¹Department of Computer Science and Engineering (CSE)
¹Gayatri Vidya Parishad College of Engineering, Visakhapatnam, India

Abstract: The underwater vehicle is an autonomous system which is impelled by an electrical propulsion system. The purpose of the underwater vehicle is to acquire, attack and destroy the enemy targets such as ships and submarines. The underwater vehicle consists of various sub-systems such as homing control system(HC), motor and battery, inertial navigation system(INS), actuation system, instrumentation system. Most of the latest underwater systems are integrated on MIL-STD-1553B bus, and the message activity on this bus takes place in real-time environment. In several test setups (in military applications), a tremendous volume of data, in the form of messages, are passed among various independent sub-systems which includes telemetry, actuator and sensor information. The data portray the estimations of different system parameters, which are of extreme importance for fault identification and analysis. Amid simulation run, all the critical messages need to be captured, stored, displayed, monitored and recorded for analysis. The data is received and stored by instrumentation system which is an embedded system with constrained memory limit because of which it cannot record all the sub-system parameters for longer mission times. This paper underlines development and advancement of MIL-1553B bus monitoring system to monitor and capture the real-time data on the bus for post mission analysis during system integration checks of underwater vehicle using visual basic 6.0 based graphical user interface (GUI).

Keywords- actuator, MIL-STD-1553B, simulation, subsystem, telemetry

I. INTRODUCTION

Underwater vehicle plays a crucial role in marine applications. The transformational idea of the unmanned underwater vehicle has brought propelled innovations to the forefront of defense capability planning. The autonomous nature of underwater vehicle requires continuous acquisition of data in regards to the working condition, decision making capability and execution of essential functions to be overseen without human intervention. This requires various onboard sensors, computers with intelligent algorithms, self-contained power source and control components to be housed on a platform that can sustain underwater environment for prolonged periods. Fig. 1.1 is a sample block diagram of underwater vehicle which consists of various sub-systems such as motor controller and battery for propulsion, homing system to detect the targets, inertial navigation system to measure own position and attitudes, instrumentation system for data recording, onboard computer to control and guide the underwater vehicle and actuation system for rudder control. In addition to these subsystems a presetter is also present. All these complex sub-systems need to be interfaced so that the mission sequence and logics are executed with tight real-time constraints.

During simulation, an enormous volume of data is generated from various modules produced by different sub-systems. These sub-systems are introduced into the test bed for performance evaluation by replacing their simulated versions. The sub-systems communicate among themselves in the form of messages.

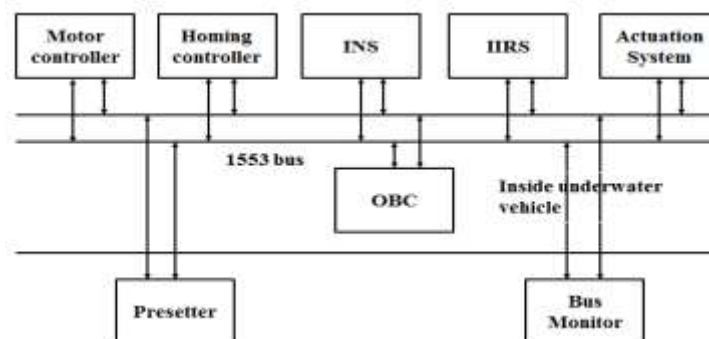


Fig. 1.1: Block diagram of underwater vehicle

In most of the systems, during user trails, the data can be recorded in one of the sub-systems. The paper emphasizes on the design and development of MIL-STD-1553 Bus Monitoring (BM) System, which could display and record parameters on the longer run, thus providing the user with maximum flexibility. The system is developed using Microsoft Visual Studio.Net Integrated Development Environment (IDE). The communication bus among the subsystem is MIL-STD-1553B with the former being most prevalent for overall system reliability and performance.

1.1.MIL-STD-1553B Bus

MIL-STD-1553B bus is a military standard that defines the electrical and protocol traits for a data bus. Its capabilities include

- The message formats
- Data words, command and status words
- Redundant dual bus, assist for both intelligent and non-intelligent remote terminals
- Error recovery.

Because of its deterministic and error free communication it is being utilized broadly in the majority of the advanced military, dispatch, aviation and submerged vehicles. In MIL-STD-1553B bus, 31 devices can be associated, out of which one device must be bus controller (BC) and other devices are remote terminals (RT). The monitor terminal (MT) can view each and every message, however it cannot send or get any of them.

1.2. Word formats

Communication between the subsystems in the bus is achieved by messages. A message is constituted by words. The protocol has three-word formats, namely command, data and status words each 20 bits wide. These are used for the communication on the bus.

1.2.1. Command word: A command word has 6 fields. The first field is sync it is used to check the validity of the message it can either be 1 or zero. Sync field uses 3 bits for 1 the first one and half bits are high and the next one and half bits are low and for 0 the first section low and next section high [8]. Remote Terminal (RT) Address the next five bits following the sync shall be the RT address. Each RT is assigned a unique address. The next bit following the address shall be the transmit/receive (TIR) bit, which indicate the action required by the RT. A logic zero indicate the RT to receive, and logic one indicate the RT to transmit. The next five bits following the transmit/receive bit is used for either an RT sub address or mode control, as is dictated by the individual terminal requirements. Following five bits is the quantity of data words to be either sent out or received by the RT. A maximum of 32 data words may be transmitted or received in any one message block. The last bit in the word is used for parity over the preceding sixteen bits. Odd parity is commonly used.

1.2.2. Data word: The Data Word (DW) contains the actual information that is being transferred within a message. The first three-bit time contains a data sync. This sync pattern is the opposite of that used for command and status words and therefore is unique to the word type. Data words can be transmitted by either a remote terminal (transit command) or a bus controller (receive command). Transmit and Receive, by convention, references the remote terminal. The next sixteen bits of information are left to the designer to define. The only standard requirement is that the most significant bit (MSB) of the data be transmitted first. The last bit (bit time 20) is the word parity bit. Only odd parity is used.

1.2.3. Status word: A remote terminal in response to a valid message transmits only the status word (SW). The status word is used to convey to the bus controller whether a message was properly

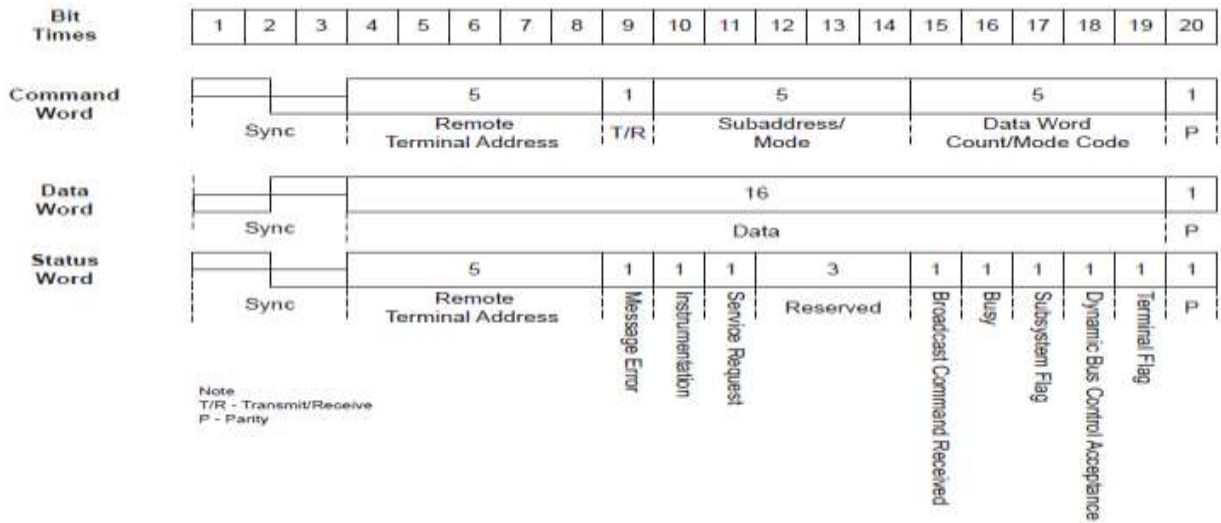


Fig. 1.2: Word formats of 1553B Protocol

1.3. Message Formats

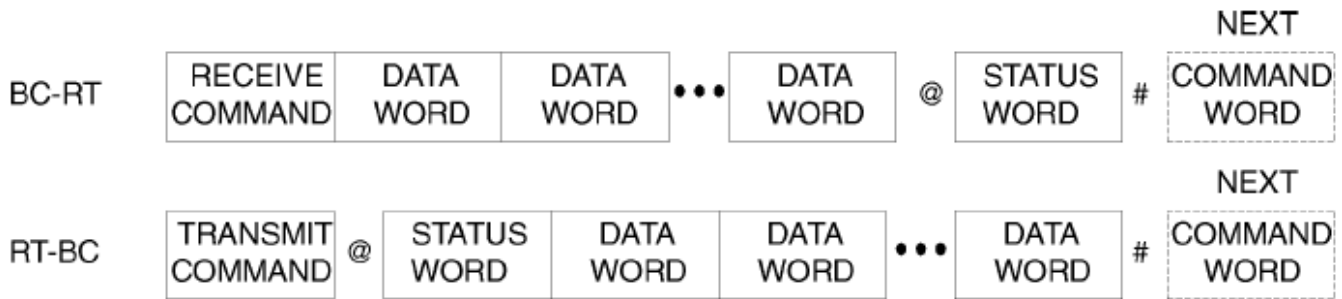


Fig. 1.3: Message formats of 1553 bus

The exchange of data is based on message transmissions. Message formats are classified as:

1.3.1. Bus Controller to Remote Terminal: The bus controller to remote terminal (BC-RT) message is referred to as the receive command, since the remote terminal is going to receive data. The bus controller outputs a command word to the terminal defining the Sub address of the data and the number of data words it sends. Following this (without any gap in the transmission), the number of data words specified in the command word is sent. The remote terminal upon validating the command word and all of the data words issues its status word within the response time requirements.

1.3.2. Remote Terminal to Bus Controller: The remote terminal to bus controller (RT-BC) message is referred to as a transmit command. The bus controller issues only a transmit command word to the remote terminal. The terminal, on validating the command word, transmits its status word followed by the number of data words requested by the command word.

II. LITERATURE REVIEW

A few works on MIL-STD-1553B are mentioned below: Amrutha V., Anu James, Sreejesh K.V, presented a paper which provides an overview of the Mil-Std-1553B bus controller implementation [12]. Jorge Rivera et al, presented a paper which provides an overview of the facility capabilities, and a detailed analysis of the results of validation and testing of remote terminals developed for Army and Tri-service programs [6]. An Peng proposed a method based on an Intelligent Node System to use Gigabit Ethernet to serve as a 1553B bus data transmission channel [7]. Gigabit Ethernet is built on the Ethernet standard technology. It has the advantages of fast communications and strong adaptability. So, it can serve as a 1553B bus data transmission channel. The 1553B bus protocol is introduced and analyzed, including data link layer, physical layer, and functional analysis. High-end FPGA chip is used as the core chip in the system designing. In its internal hardware, there are integrated Gigabit Ethernet physical layer. Gigabit Ethernet and 1553B bus has its own protocol and data frame formats along with the physical interface. The communication between them was achieved through

the bus conversion nodes in the proposed system. Michael Hegarty presented a paper on leveraging MIL-STD-1553's physical layer for use in aircraft data networks [8]. He presented an overview of the unique benefits of MIL-STD-1553's 1 Mbps physical layer and proposed new solutions based on derivatives of MIL-STD-1553. The derivatives of MIL-STD-1553 include both higher data rates and the use of 1553's physical layer with other protocols. Jemti Jose et al, proposed a method for implementation of the Military data bus standard MIL-STD-1553 onto a Xilinx based FPGA platform [9]. In this method, the protocol controller is modelled as state machine in HDL. In 1992 Ni Yu De et al, presented a paper on ideas and methods about the design and development of a high performance optical data bus based on MIL-STD-1773 [10]. The paper also proposes methods to avoid impulsive interference efficiently. By the use of large core and large numerical aperture fiber optics. He proposes the advantages of using high power LEDs and a low noise optical receiver structure to get high signal- to noise ratio at decision time. There are some critical technical problems to be solved before the MIL-STD-1773 data bus is used in aircrafts, e.g., optical modulation, optical source/detector and fibers, network architecture, and optical network components. In 1992 Jian Guo Zhang presented a paper which discusses about the key issues on designing high-reliability optical fiber transmission systems for avionics data buses [11].

III.METHODOLOGY

3.1. Architecture of Bus Monitoring System

The architecture of bus monitoring system is shown in fig. 3.1. 1553 device is a hardware equipment consisting of 1553 card. The application (monitor terminal) communicates with 1553 card through device drivers and application interface library provided by the software development kit (SDK) along with the card. SDK provides the framework for developing real time applications for series of MIL-STD-1553 components. The 1553 device can be configured as bus controller, bus monitor and 32 remote terminals.

The 1553 card monitors single bus at a time, stores the opted packets and discards the others, the selected packets are essentially the packets containing data for future analysis and display. Synchronization is done through the specific SYNC message. The card

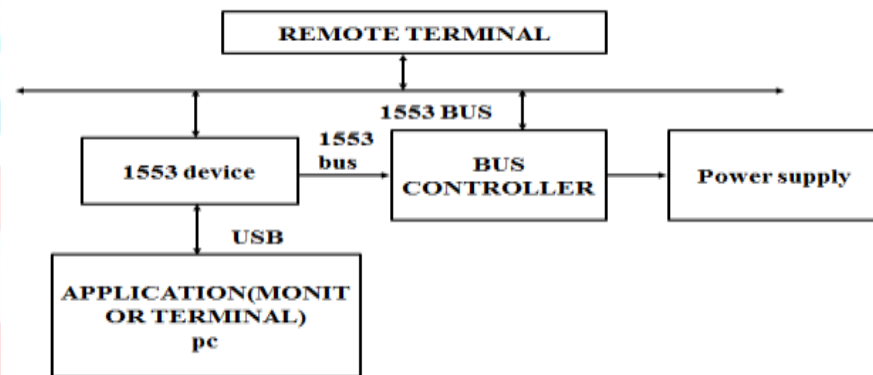


Fig. 3.1:Block diagram of Bus Monitoring System

implements a message buffer, which stores the messages on the bus temporarily and is accessed from the software. The IIRS sub system is simulated and the messages present on the bus need to be captured and displayed.

3.2. High-Level API Flow for a Single 1553 Logical Device

3.2.1. Logical Device number (LDN): A logical device number is allocated to each channel using DDC card manager. This card allows user application to reference a virtual device, but not the specific hardware channel directly.

3.2.2. ace initialize (): Each channel is initialized using ace initialize function.

3.2.3. ace free (): Using ace free function the channel can be reset.

3.2.4. Configuring Operating Modes: BC, RT, MT are the three configuring operating modes.

3.2.5. Access 1553 data: access to 1553 data is done by configuring MT-I mode. The MT-I mode allows the user to monitor all 1553 traffic present on the bus.

3.3. Bus Monitor System configuration and Data access

3.3.1. aceMTIConfigure: It configures frequency of message reception and event notifications. Parameters include hardware buffer sizes and selectable events. The events help in controlling the size and frequency of the 1553 packets supplied to the user.

3.3.2. aceMTDisableRTFilter: Traffic from specific RT’s can be disabled using aceMTDisableRTFilter.

3.3.3. aceMTiStart: This function initializes all command and data stack pointers, monitor structures, and monitor registers necessary to run the device in monitor (MT-I) mode. After this function has been called, the device is left in a Run state.

3.3.4. acexMTIGetCh10DataPkt: All 1553 traffic can be processed by the user by calling this function. It can selectively block until new 1553 data is available.

3.3.5. acexMTIStop: This function stops the Monitor from capturing messages and puts the device into the Ready state.

3.4. 1553 Bus Monitor 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 a virtual environment. The main objective of this MIL-STD-1553B bus monitor simulator is to send and receive data from IIRS and to capture, monitor and display the data present on the bus which enables users for further analysis. It displays alarms like water contact (WC), leak contact sense (LCS), depth cut-off (DCO), cable cut-off (CCO) etc. This Simulator is implemented by using MIL-STD-1553B AceXtreme hardware. A user-friendly GUI is developed and designed as per interface control document provided by NSTL using visual C#.

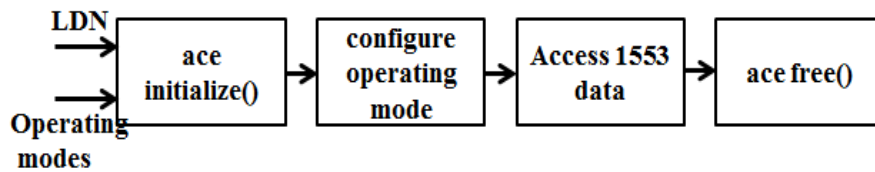


Fig. 3.2: API Flow

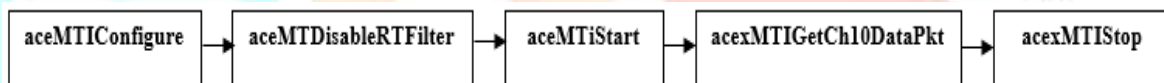


Fig. 3.3: Bus Monitor (MTI) Mode - Configuration and Data Access API Flow

IV.DESIGN AND IMPLEMENTATION

4.1. Dataflow

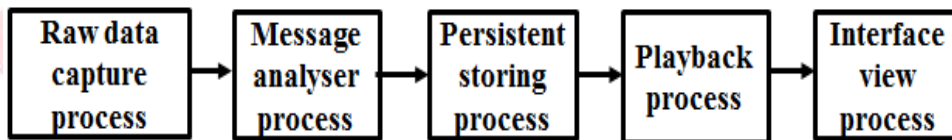


Fig. 4.1: Data Flow in 1553 Busx

In integrated standalone system, all the modules are part of single software. As mentioned in fig. 4.1, the data capture process captures raw data from MIL-STD-1553 bus. Then the message analyser process kicks in and extracts data from the messages. The raw data and extracted processed data is stored to persistent storage by the storing process. The playback process reads one of these previously stored data and directly populates the processed data structure. The interface view process accesses the processed data structure and displays different observed parameters to the user.

4.2 Design of Bus Monitor Simulator

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

4.2.1.1 Initialize device: This function initializes hardware resources such as memory and register space for a particular mode of operation.

4.2.1.2 Configure device: The device is configured to MT-I mode. Configure MT-I mode for a specific logical device number

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 RT and sub address blocks: Create and map data block for each and every sub address.

- *Create:* An RT data block is created and identified. 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 sub addresses of the IIRS RT.

4.2.2 Device Start: When the device start function is enabled, all the required registers are set to aligned enhancing mode code handling, then the remote terminals operation gets started. The enabled remote terminals will respond to messages on the 1553 bus. The device transits from ready state to run state.

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

4.2.4 Data Pre-processing: The input module is responsible for configuring and controlling the input port. The input selector module is basically a multiplexor. The pre-processing unit shown in fig3.2 takes the input from test configuration module. The configuration module enables user to specify port communication protocol parameters. the data pre-processing unit consists of following modules:

4.2.4.1 Packet Parser: Packet parser time stamps each dataset, with time provided in the packet, or in absence of that with a local system time. Depending on the data setup, it identifies one complete set of the parameters, partial or damaged packets are discarded. With a flexible approach to build data strings, the parser parses the received strings to extract data values.

The following features are supported:

- Binary or ASCII values
- Big Endian or Little Endian for binary values
- Single byte data (BYTE)
- Double Byte data (WORD)
- 4-byte data (DWORD)
- 8-byte data (ULONGLONG)
- Signed or unsigned data

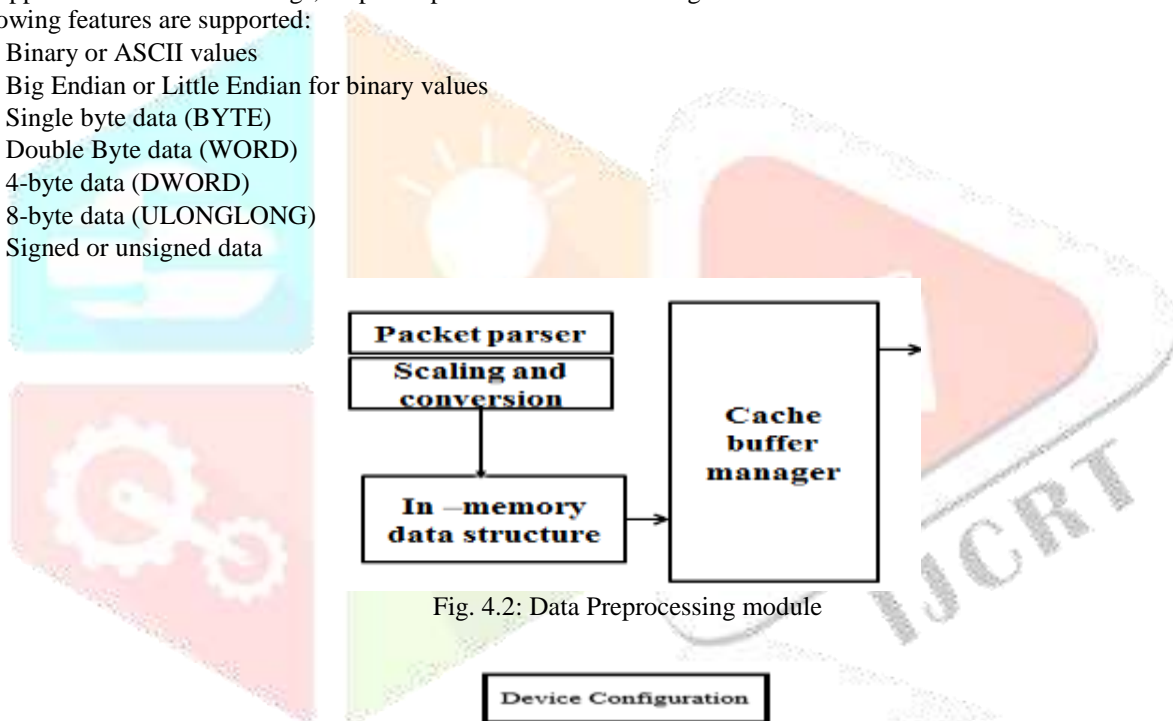


Fig. 4.2: Data Preprocessing module

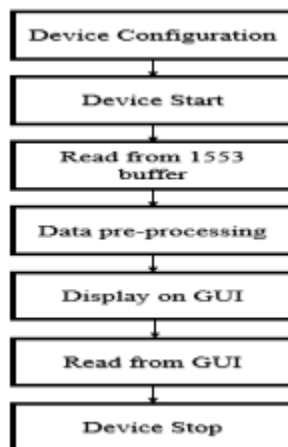


Fig. 4.3: Implementation of bus monitoring system

4.2.4.2 Scaling and Conversion: In Scaling and Conversion, the data is scaled and offset is added ($\text{value} = \text{scale} * \text{data} + \text{offset}$). The value is then converted to engineering units.

4.2.4.3 In-memory data structures: The scaled and converted data sets are stored into an in-memory data structure. This uses a circular queue and a linked list, to be decided depending on the amount of data to be cached and available memory. It takes the help of cache buffer manager for allocation of memory.

4.2.4.4 Cache buffer manager: Cache buffer manager provides the functionality of application specific memory management. It contains raw data in which as soon as the input selector sends data, it is stored in this module. The input modules become free and go back for receiving the next set of data. This has the highest priority to ensure that no input data is lost. It also contains processed data in which the scaled and converted datasets of in-memory data structures are stored as per need.

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

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

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

4.3 Results

4.3.1 Interface Control Document (ICD)

ICD was provided by NSTL. Based on the message list, the parameters in the interface are designed, captured and displayed

Table.4.3.1 1553B message list

S. No	Message ID	Tx / Rx	SA	Description	Word Count
IIRS (RT: 01)					
1	M101	Rx	01	Preset parameters for logging	12
2	M102	Rx	02	Homing parameters for OBC	28
3	M103	Tx	03	IIRS alarms to OBC	03
4	M104	Rx	04	Mission Parameters from OBC	25
6	M106	Rx	06	OBC Prelaunch Flags	06
7	M107	Rx	07	BLDC parameters from OBC	12

4.3.2 Outputs



Fig. 4.4: Device Configuration

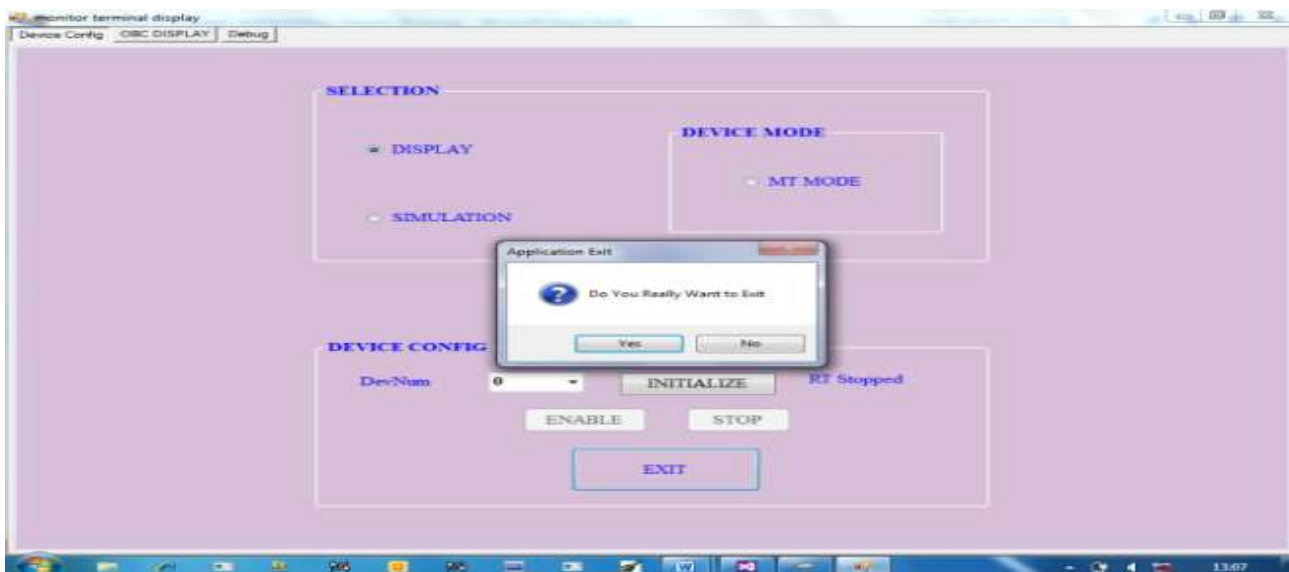


Fig. 4.5: Application exit



Fig. 4.6: Display of bus monitoring system parameters



Fig. 4.7: Display of Debug information

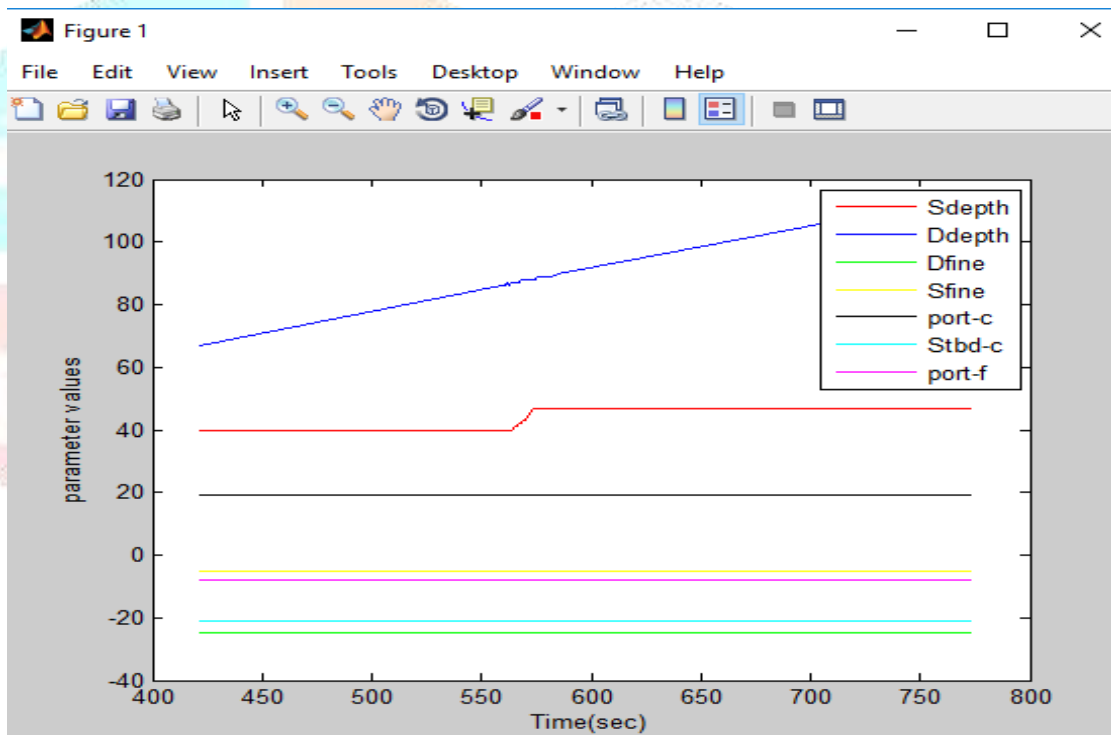


Fig. 4.8: Graph plotted against time and homing control parameters

The outputs show the simulation of bus monitoring system. Initially, the device is configured and the required monitor mode is chosen. Once the mode is selected, the device is initialized and enabled; thus capturing, displaying and monitoring all the six sub address parameters. Homing Control data parameters constituted in sub address 2 are stored in a file. A plot of homing control data parameters against time is depicted in the fig. 4.6 using MATLAB.

V. CONCLUSION

The bus monitoring system has been developed using the standard components (hardware, OS, software environment), such that it can be implemented and used effectively in under water survey missions such as detection and mapping of obstructions, rocks submerged wrecks, in most of the laboratory installation and user environment. The familiarity of the OS, GUI controls and interfacing with standard tools like MATLAB is an added benefit. The system acquires various sensor parameters such as pitch, roll, yaw, pitch rate,

roll, depth and speed information. Fast and fault tolerant storage system eliminates the error prone manual process of data back-up and may be used effectively in installations, where a large number of runs need to be analyzed.

VI. FUTURE SCOPE

The system may be further enhanced, by incorporating the data gateway system within a truly real-time environment like RTLinux. As a further refinement, the gateway system may be implemented as an embedded system over RISC architecture, providing a faster data capture capability.

VII. ACKNOWLEDGMENTS

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

I would also like to show my gratitude to the Organization, especially the HR department for giving me this opportunity to work along with the esteemed employees of Naval Science and Technological Laboratory.

VIII. REFERENCES

- [1] <http://www.ddcweb.com/Products/172/Default.aspx>
- [2] www.ddcweb.com/Documents/dguidehg.pdf
- [3] MIL-STD-1553B tutorial by condor engineering
- [4] www.milstd1553.com
- [5] https://www.abaco.com/system/files/files/mil-std-1553_tutorial_a-wp-711b_0.pdf
- [6] D. S. D'Avino, and J. Rivera, and C. DeLong, "MILSTD-1553 validation and test facility," Digital Avionics Systems Conference, 1990. Proceedings., IEEE/AIAA/NASA 9th, pp. 436-440, October 1990.
- [7] An Peng, "Intelligent node of 1553B bus based on Gigabit Ethernet protocol," Intelligent Control and Automation (WCICA), 2010 8th World Congress on, vol., no., pp.953,956, 7-9 July 2010.
- [8] Hegarty, M, "Leveraging MIL-STD-1553's physical layer for use in aircraft data networks," Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th, vol., no., pp.7B1-1,7B1-8, 16-20 Oct. 2011.
- [9] J. Jose, and S. Varghese., "Design of 1553 protocol controller for reliable data transfer in aircrafts," Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference on, pp.686-691, Nov 2012.
- [10] Jian-Gao Zhang, "Design of new optical fiber transceivers for a MIL-STD-1773 data bus," Aerospace and Electronics Conference, 1992. NAECON 1992., Proceedings of the IEEE 1992 National, vol., no., pp.1186,1192 vol.3, 18-22 May 1992.
- [11] Ni Yu-De; Li Zheng; Zhang Jian-Guo, "A high performance fiber optic data bus based on MIL-STD- 1773," Digital Avionics Systems Conference, 1992. Proceedings., IEEE/AIAA 11th, vol., no., pp.400,406,5-8 Oct 1992.
- [12] Amrutha V., Anu James, Sreejesh K.V, "Mil-Std-1553b Bus Controller Implementation: A Review," International Journal of Industrial Electronics and Electrical Engineering, ISSN: 2347-6982 Volume-3, Issue-8, Aug.-2015.