Development of Data Simulation and Processing GUI for Underwater Applications using PCI Add-On Card

Ramu Korukonda, K J Sreeja

Abstract: ***In application areas like data decoding, analog signal processing, telemetry, underwater vehicles etc, monitoring of analog and digital signals of various subsystems and mission parameters is very essential. Especially in the case of underwater vehicles, the performance evaluation is done offline because it is not possible to transmit data from vehicle either by cable or wireless. This calls for the onboard recording system to record different parameters and download those parameters for offline analysis. Parameters like navigation, control and target parameters should be transferred between vehicle and mothership for successful run, this is accomplished via surface, acoustic and fibre optic modem's depending on the conditions. This paper provides an overview of the development of an interactive simulation system and Graphical User Interface (GUI) for an underwater robotic vehicle. Computer simulations and graphical interfaces are effective ways for studying the behaviour of a system well before it is developed and deployed. It allows lot of flexibility in the design and greater scope for improvement without major financial commitments. The underwater vehicle and its sub-system properties, control inputs (parameter adjustments, maneuvers, etc), and the underwater environment are graphically represented on an user interface that can be easily accessed by the user for offline analysis and for further modifications.***

Keywords: ***Underwater vehicle, Onboard recording system, simulation system, GUI.***

I. INTRODUCTION

In Naval applications, underwater vehicles plays crucial role. The transformational natures of the Unmanned Underwater Vehicles and advanced technologies available have brought them to the forefront of defence capability planning. The vehicles that are netted, mobile, adaptable and reconfigurable will play pivotal role in the future multi-tired battle space complementing the traditional platforms while reducing threat of human loss. The autonomous nature of underwater vehicle requires continuous acquisition of data regarding the operating environment [7], decision-making capability and execution of essential functions to be managed without human intervention. This requires various onboard sensors, computers with intelligent algorithms, self–contained power source and control elements to be housed in a platform that can sustain underwater environment for prolonged periods [7]. But the telemetry is not possible in case of underwater vehicles as attenuation of high frequency electro- magnetic waves in the water (particularly in sea water it is too high) and at low frequencies abnormal subsystem dimensions are required (minimum length of antenna is half of the wave length). So the only possibility is offline analysis [6]. This offline analysis of vehicle is done by using onboard recording system.

In this paper, we exploit the serial, Tcp/ip communication functionalities of Visual basic6.0 to enable a PC to communicate with Power Pc microcontrollers to transmit control commands and receive sensory data. In addition, we utilize Matlab, Simulink [9] to develop an interactive GUI environment for testing the underwater vehicle performance by simulating analog and digital data.

Manuscript received on July, 2012.

Ramu Korukonda, Electronics & Communication Engineering, Vignan's Institute of Information Technology, Visakhapatnam, India,

K J Sreeja, Electronics & Communication Engineering, Vignan's Institute of Information Technology, Visakhapatnam, India,

II. SIMULATOR STRUCTURE

A. Simulator Architecture

The diagram of the simulator is shown in Fig.1. The simulator is combined of three parts [4], computer with the GUI of the onboard recording system, onboard recording unit and the computer with the 1553 bus simulation [11]. The GUI mainly operates in different modes, they are online-monitoring mode, data downloading mode and processing mode.

The recording system has two modes of operation, data recording and recovery. The recording system will acquire the vehicle parameters [9] from onboard computer system and other relevant information from the sensors attached to it and stores the data for post run analysis.

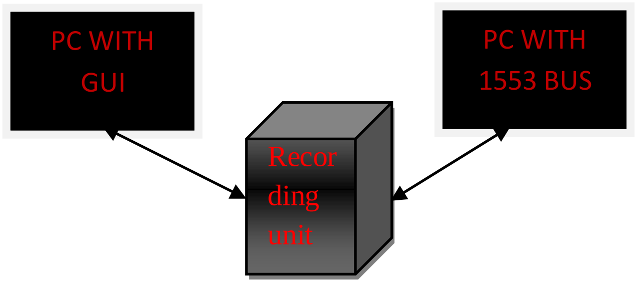


Fig.1 Architecture of the Simulator

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Recording system will monitor the health of the vehicle by acquiring health data of the sub-systems and if any of the systems fails then recording system will generate alarm for recovery. Recording system will initiate recovery action based on various conditions like malfunction of a major sub-system, completion of mission, run execution time or any recovery command given by the mother ship PC via onboard computer system.

MIL-STD-1553B [12], defines the term Time Division Multiplexing (TDM) as "the transmission of information from several signal sources through one communications system with different signal samples staggered in time to form a composite pulse train." In underwater vehicles, this bus is used to transfer data between multiple vehicle units (onboard computer system, recording system, control and guidance systems ...etc) over a single transmission media, with the communications between the different avionics boxes taking place at different moments in time, hence time division.

The user friendly GUI to interact with the onboard recording system [4], is developed in visual basic6.0. In GUI, different analog parameters (thruster voltage, thruster current, depth.. etc) and digital parameters (TTL's, Relay's corresponds to recovery aids after run termination) are simulated [6] to test working status of the recording system. The software implementations of different modes are described below.

III. ONLINE MONITORING MODE

The Graphical User Interface application [9] is mainly designed for controlling the operation and to get the status of the Unmanned under water vehicle by sending the commands from the GUI to UUV (unmanned underwater vehicle). By using online monitoring mode, we can test different subsystem's working status in emergency situations and health monitoring of the vehicle by sending different commands like ceiling cut-off (CCO), depth cut-off (DCO), subsystems failure . .etc.

For online monitoring [1], we are using the serial communication to transfer data between the GUI and recording unit. RS-232 [3] is the interface that uses to talk to and exchange data with physical serial devices. The following paragraphs describes how the connection is established, data transfer between devices is accomplished using visual basic

A. Establishing a Connection

The connection between the ports is established with the **portopen** [5] property of the mscomm1.

Mscomm1.portopen = True

Then the port is opened for connection. If this property is equal to false then the port is closed which closes the connection.

B. Sending and Receiving Data

Once the connection is established, the data will be sent to the target device using mscomm1's "output" property.

Mscomm1.output = "Hello"

The data is received at the remote port using "input" property.

Variable = Mscomm1.input

C. Monitoring the Connection

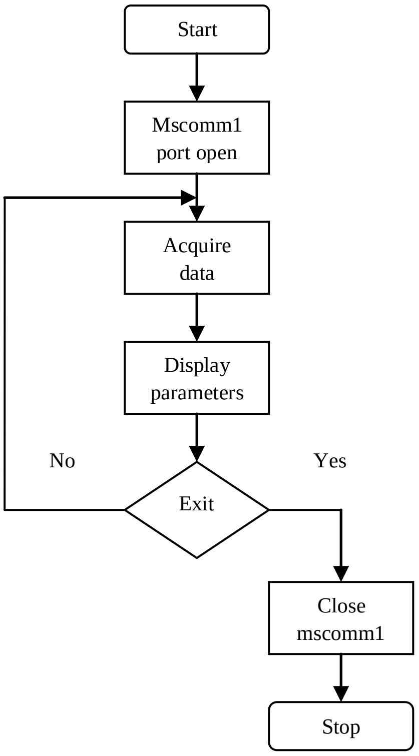
The OnComm event of mscomm1 is generated whenever the value of the CommEvent property changes, indicating that either a communication event or an error occurred.

Private Sub object_Oncomm ()

D. Serial Port Settings

The "settings" property is used to set different parameters like baud rate, start/stop bit, parity bit and send/receive threshold value according to the application.

Mscomm1.Settings = value



Flow chart for online monitoring mode

IV. DATA DOWNLOADING MODE

In downloading mode, the data recorded in onboard recording system during the vehicle run is downloaded to the test station pc by clicking the **download** command button on the GUI [9].

For data downloading, we are using the TCP/IP communication [13] instead of serial communication due to higher data transfer rates. The Visual basic WINSOCK [2] control is used to simplify network communications. Each Ethernet device is configured with a unique IP address and port number. The application program must make a "socket" connection [5] to the device using the IP address and port number assigned to it. Once the "socket" connection is established data can be transferred to and from the target device as a data stream. The contents of the actual data stream will depend on the target device.

A. Establishing a Connection

A socket connection can be established with the network target device using the WINSOCK control. Since the control can be used for either TCP or UDP communications the "Protocol" property must be set for TCP protocol (default) first. The IP address and port number [5], of the target device are stored in the “RemoteHost” and “RemotePort” properties. These properties can be set either at design time or at run time. Once the properties are set the “Connect” method is used to establish a connection with the target device.

```
Winsock1.Protocol = sckTCPProtocol
Winsock1.RemoteHost = “192.168.168.50”
Winsock1.RemotePort = 1070

Winsock1.Connect
```

B. Sending Data

Once the connection is established data can be sent to the target device using the “SendData” method.

```
Winsock1.SendData “DISPLAY=Hello World”
```

C. Receiving Data

The WINSOCK “DataArrival” event is used to capture data sent by the target device. Within the event the “GetData” method is used to capture the received data. Typically, the captured data is moved to a temporary buffer (data writing into a file) and then processed by another routine. Since the received data is a stream the processing routine may have to parse incomplete strings or blocks as they arrive.

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As
Long)
Dim buf As String
Winsock1.GetData buf
ProcessingRoutine buf

End Sub
```

D. Closing the Connection

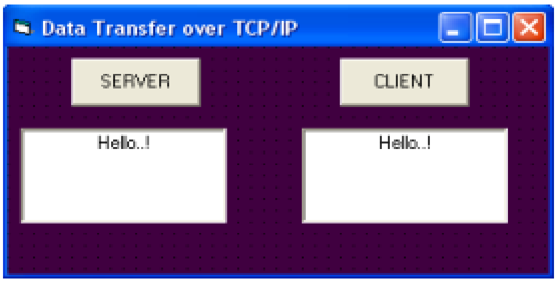
The established socket connection typically remains open for as long as it is needed to communicate with the target device. Either end of an established socket connection can close the connection when it is no longer needed. The application program can use the WINSOCK “Close” method to force the connection closed. If the target device initiates the close the WINSOCK “Close” event is triggered to notify the program that the connection has been closed.

```
Winsock1.Close
```

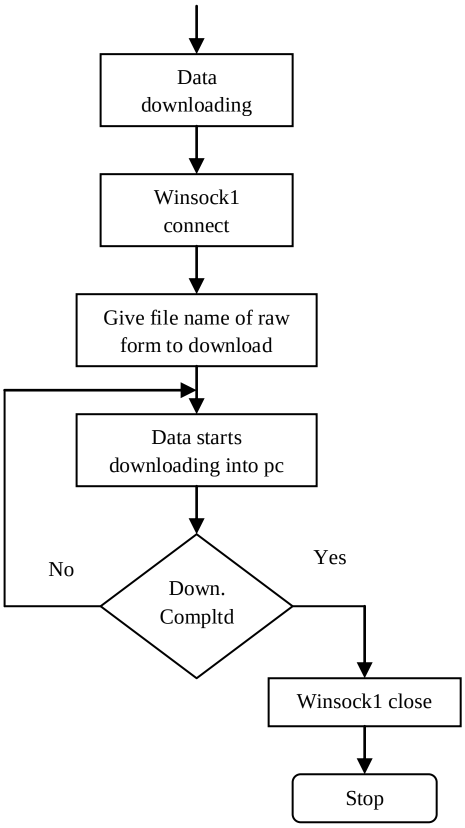
E. Monitoring the Connection

The WINSOCK control provides a “State” property which can be read at any time to determine the current connection status. An “Error” event is triggered whenever a connection error as occurred. Often the “State” property is used in conjunction with the “Error” event to determine what steps should be taken.

```
Private Sub Winsock1_Error(ByVal Number As Integer,
Description As String, ...)
Dim ms as String
ms = “Error=” & Number & “ “ & Description & vbCrLf
ms = ms & Winsock1.State
Msgbox ms,,”Connection Error”
End Sub
```



The above figure shows how the data is transferred between different systems and displayed in text boxes according to requirement using TCP/IP communication.


Data downloading
Winsock1 connect
Give file name of raw form to download
Data starts downloading into pc
Down. Compltd
No
Yes
Winsock1 close
Stop

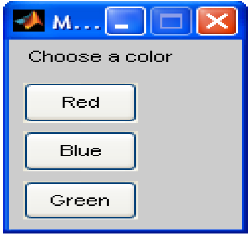
Flow chart for data downloading

In our application, the data stored in the recording system’s memory (usually NVRAM) during vehicle run is to be downloaded to the test station pc for post trial analysis. It is accomplished with the Power Pc 440 Embedded processor [14], which provides two Ethernet media access controllers (EMACs) that are generic implementations of the Ethernet Media Access Control (MAC) protocol complying with ANSI/IEEE Std 802.3 and IEEE 802.3u supplement. EMAC supports both half-duplex (CSMA/CD) and full-duplex operation for 10-Mbps and 100-Mbps operations.

V. PROCESSING MODE

In processing mode, the downloaded raw files are splitted into data files and address files according to requirement by clicking the **processing** command button. After processing, the graphs for different parameters such as depth, latitude, longitude, SONAR voltage and threshold voltage are plotted using Matlab7.0

[8] to analyze the behaviour of the vehicle during run. The matlab7.0 menu is used to select different parameters. Whenever a parameter is selected, it returns the corresponding data from the directory and that data can be plotted using **eval, plot** functions in Matlab7.0 [8].



The above sample figure is obtained with Menu function

K = menu (“title”; item1, item2, . ..)

K= menu(“choose a color”; Red, Blue, Green)

An integer value is returned into K if a color is selected. For example if blue is selected then ‘1’ is returned.

VI. 1553 BUS SIMULATION

To use a real 1553 bus, expensive cards must be purchased and additional cabling must be used to connect the various system components. The idea is to move functionality from hardware into software, in essence, create a completely software-based 1553 card simulator [11].

The bus operates asynchronously [12], employing a command/response protocol. Transmission of information on the bus is controlled by the bus controller, which initiates all transmissions. In addition to the bus controller, the standard allows for up to thirty-one remote terminals to be connected to the bus, which are recipients of commands from the bus controller. Remote terminals process these commands and send responses back to the bus controller. Fig.2 depicts sample 1553 bus architecture.

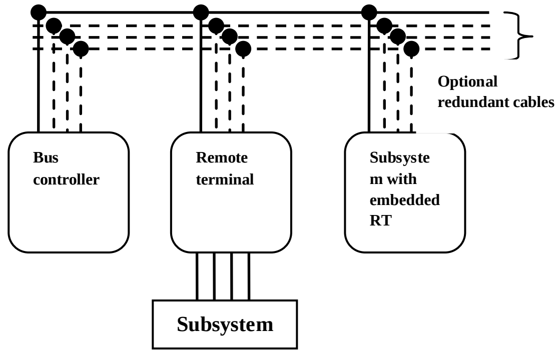


Fig.2 1553 Bus Architecture

The subsystem referred to in the figure is simply a device that is using the bus for data transfer. Bus monitors, terminals that receive bus traffic but do not respond to any message, may also be connected to the data bus. Information is transmitted on the bus in the form of messages [12], which consist of three types of words: command, data, and status. Each word consists of sixteen bits, most significant bit first, plus a three-bit time sync word and parity bit for a total of twenty bit times. Each bit time is one microsecond, as the transmission rate of the bus is one megabit per second. See Fig.3 for an illustration of the three word formats. The standard defines a variety of message formats [12]; however, each follows the same general pattern. The bus controller initiates all messages by the sending a command word to a remote terminal. If the remote terminal is to respond, it always responds first with a status word. The difference between the different message types is in the direction data flows (i.e. from bus controller to remote terminal, from remote terminal to bus controller, etc.). The maximum number of data words that may be transmitted per message is thirty two. See Figure 4 for an illustration of some of the 1553 message types.

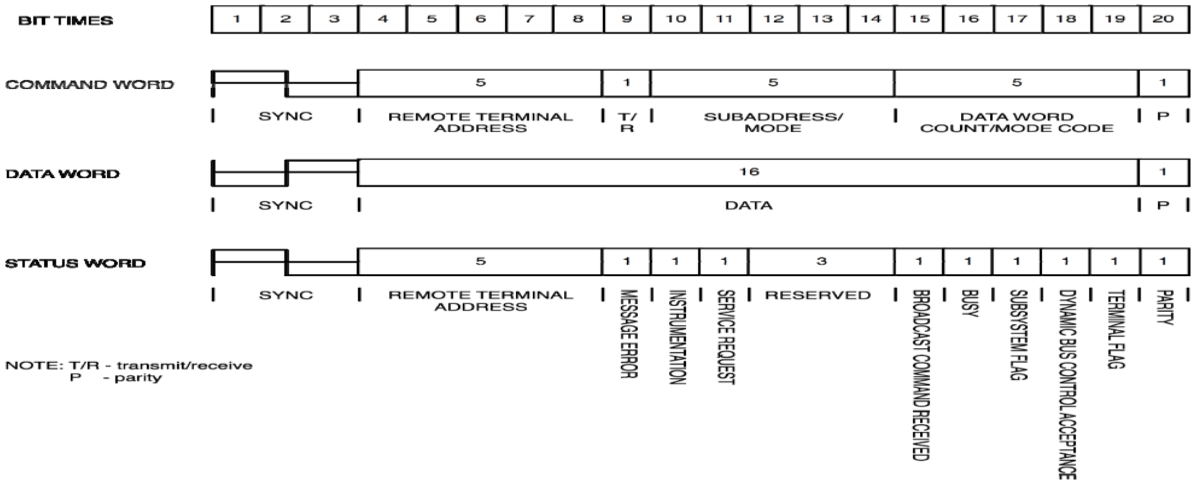


Fig.3 MIL-STD-1553Wordformat

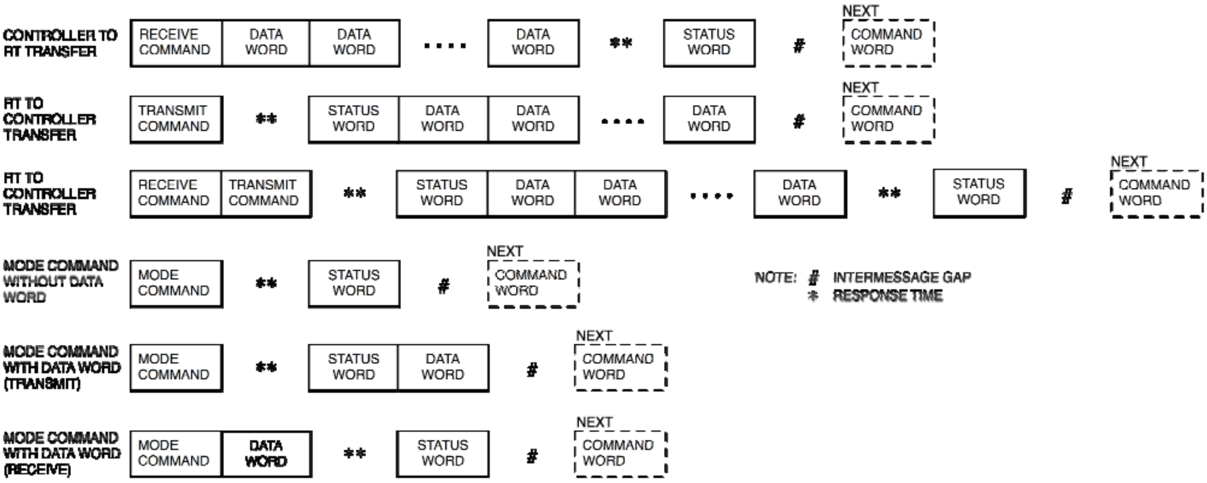


Fig.4 MIL-STD-1553 Message formats

VII. TEST RESULTS

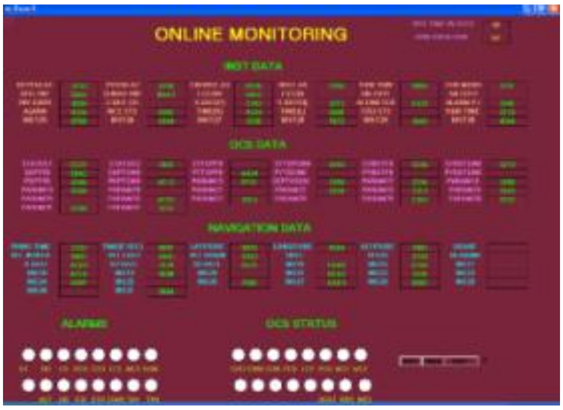


Fig.5 "Online monitoring" is used to test working status of the different subsystems.

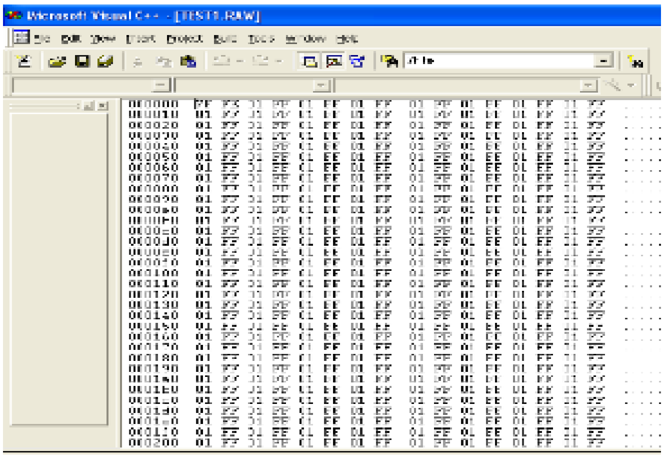
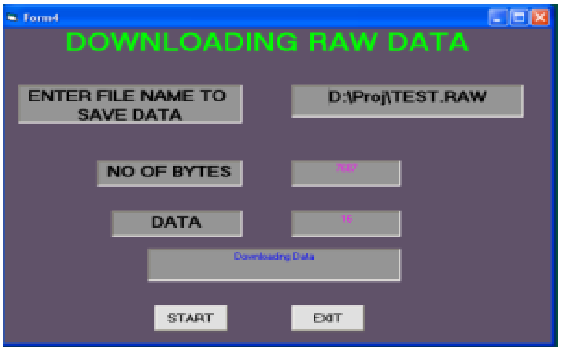

Microsoft Visual C++ - [TEST1.RAW]

Fig.7 Downloaded raw data from the onboard recording system


DOWNLOADING RAW DATA

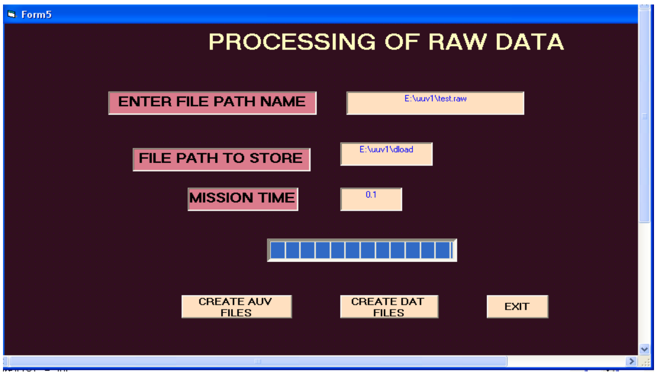
ENTER FILE NAME TO SAVE DATA: D:\Proj\TEST.RAW

NO OF BYTES: [illegible]

DATA: 16

Downloading Data

Fig.6 The data stored in onboard recording system is downloaded as raw data to test station pc


PROCESSING OF RAW DATA

ENTER FILE PATH NAME: E:\uuv1\test.raw

FILE PATH TO STORE: E:\uuv1\dload

MISSION TIME: 0.1

Fig.8 Processing of raw data for further analysis

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

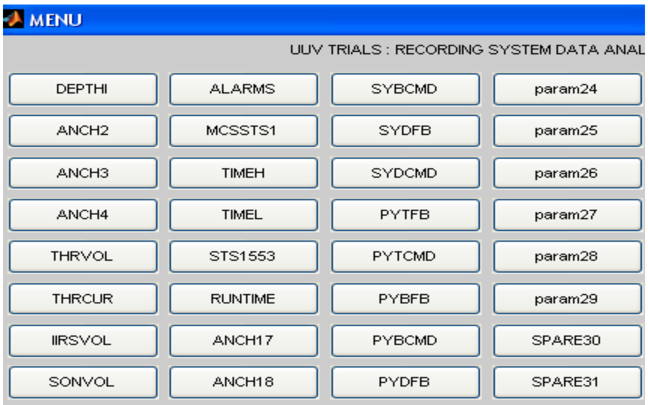


Fig .9 Graph analysis for processed onboard recording system data

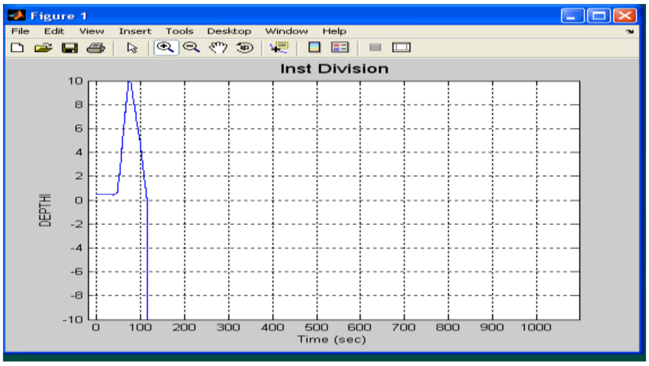


Fig.10 plot for the Depth parameter

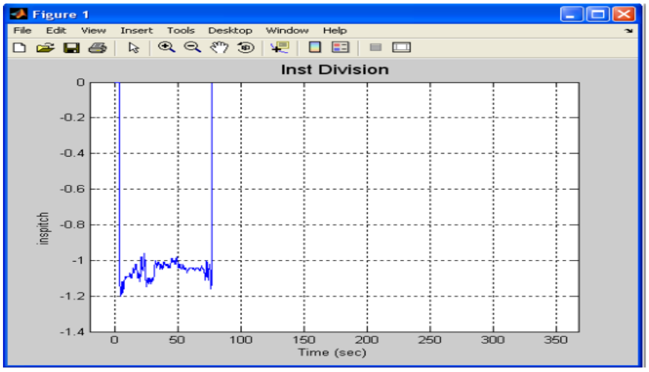


Fig.11 plot for the pitch parameter

X. CONCLUSION

The PCI Add-on card based Data Simulation and Processing GUI has been developed using Visual Basic6.0 and Matlab Simulink. It is used to interact with different subsystems in an underwater vehicle to test the working status of systems. The GUI is also used for online monitoring of recording system by simulating analog, digital data In offline, GUI can be used to simulate different real time parameters for testing onboard recording system. GUI provides effective ways for studying the behaviour of a system well before it is developed and deployed. It allows lot of flexibility in the design and greater scope for improvement without major financial commitments and to prevent possible damages of vehicle subsystems.

REFERENCES:

1. "Online Object Monitoring With Go4 V4.4" , J. Adamczewski-Musch, H. G. Essel, and S. Linev, IEEE Trans. On nuclear science vol.58 no. 4 Aug.2011
2. "TCP/IP Tutorial and Technical Overview", Lydia Parziale David, T. Britt Chuck Davis and Wei Liu, IBM Corp. , 8th edition in 2006
3. Richard Grier, Serial Communication Handbook.
4. "A New On-Board Data Handling System for Spacecraft Flight Control Simulator, Wu Jinjie, Hu Min, Gao Yudong,
2011 IEEE 5th International Conference on Cybernetics and Intelligent Systems (CIS)
5. "University of Phoenix - Computer Programming Concepts and Visual Basic" David I. Schneider, Pearson Custom Publishing special edition.
6. "Integrated simulation system for Rapid Development of Autonomous Underwater vehicles",D.P. Brutzman, Y. Kanayama & M.J. Zyda, Proc. of the IEEE Oceanic Engineering Society AUV92 Conference, Jun. 1992.
7."Control of an underwater vehicle using H-infinity synthesis", Issac kaminor,Antonio pascoal, 30th IEEE conference on decision and control, Brighton, England, Dec.1991
8. "http://www.mathworks.com/products/simulink/, website of The Math Works, Inc., developer and distributor of Simulink (access link for Simulink product information).
9. "Development of a Matlab-Based Graphical User Interface Environment for PIC Microcontroller Projects"Sang-Hoon Lee, Yan-Fang Li, and Vikram Kapila, Proc. Of ASEE, 2004
10. "A Hardware-in-the-loop Simulation System of Heave Compensation of Deepsea Mining", Qiong Hu, Shaojun Liu, Proc. Of ISOPE Ocean Mining Symposium, sept. 2009
11. **"**VIRTUAL MIL-STD-1553" Nicholas Downing, United Space Alliance, LLC, Houston, 25th Digital Avionics Systems Conference Oct. 15, 2006.
12. "Multiplex Applications Handbook", DOD, Washington, USA, Nov.1st 1988
13. "Network Primer and Programming Tutorial for the Model 2701 Ethernet- Based DMM/Data Acquisition System" Keithley Instruments, Inc., 2002.
14. "A Multiprocessor System-on-a-Chip Design Methodology for Networking Applications" Valentina Salapura, Christos J. Georgiou, Indira Nair IBM Research Division Thomas J. Watson Research Center, may 7th, 2004.

AUTHORS PROFILE

Ramu korukonda is pursuing his M. Tech degree in the Department of Electronics & Communications, Vignan's institute of Information and Technology, Duvvada, Visakhapatnam, Andhra Pradesh His Area of interests are Electronics, Communication and Networking.

Sreeja Kumari Jaya has received B.E. From S.R.K.R Engineering College in the year 2001. She has completed her M.Tech in the year 2003 from University of Kerala with Digital Image Computing as her specialization. She has seven years experience of teaching under- graduate and post-graduate students. Currently working as a Associate professor in Vignan's Institute of Information and Technology, Visakhapatnam, Andhra Pradesh. Her research interest area is digital image processing, communication and networking.