RESEARCH ARTICLE OPEN ACCESS

Design and Development of Real Time Mission Software for wire guided underwater vehicle

K Siva Kumar*
*(Department of Computer Science& Systems Engineering, Andhra University, Visakhapatnam-03

ABSTRACT
This paper describes design and development of Real Time Mission Software (RTMS) for wire guided underwater vehicle. RTMS is ARM controller based real time embedded system. The main role of RTMS is to acquire sensors data, control and guide the vehicle as per the mission requirements. It performs Pitch control, Roll Control, Yaw Control and depth control operations as part of vehicle stabilization and it executes both active guidance and passive guidance algorithms to track the vehicle towards the target. In addition to that it will also distribute the power to subsystems, perform launch sequence and operate recovery operations. RTMS software receives the commands from submarine through fiber optic cable. RTMS operates either in fiber optic cable mode or autonomous mode based on user requirement.
Keywords: ARM controller, control, embedded system, guidance, real time

I. INTRODUCTION

This paper talks about design and development of Real Time Mission Software (RTMS) for underwater vehicle which is guided through fiber optic cable. The existing systems probably getting operated autonomously [1] so that there is no communication from the mother ship or submarine. The RTM software can receive commands from mothership/submarine via fiber optic cable or it can be operated autonomously.

1.1. Real Time Systems (RTS): A real time system that must satisfy explicit (bounded) response-time constraints or risk severe consequences including failures. The correctness of the system depends not only on the logical correctness but also the time at which the results are produced. It should respond to externally generated input stimuli within a finite and specified period. Real time systems can be classified as ,

1.1.1. Hard Real Time Systems: An overrun in response time leads to potential loss of life and/or big financial damage. Many of these systems are considered to be safety critical systems.

1.1.2. Soft Real Time Systems: Soft real time systems are the systems in which deadline overruns are tolerable but not desired. There are no catastrophic consequences of missing one or more deadlines.

1.2. Characteristics of RTS : This section describes the important characteristics of a Real Time System. All the characteristics which are described below may not applicable to every Real Time System

1.2.1. Time Constraints: Every real time task is associated with some time constraints. One form of

1.2.2. time constraints that is very common is deadlines associated with tasks. A task deadline specifies the time before which the task must complete and produce the results.

1.2.3. Correctness Criterion: The notion of correctness in real time systems is different from that used in the context of traditional systems.in real time systems ,correctness implies not only logical correctness of the results ,but the time at which the results are produced is important. A logically correct result produced after the deadline would be considered as an incorrect result

1.2.4. Embedded: A vast majority of real time systems are embedded nature. An embedded computer system is physically "embedded" in its environment and often controls [2] it. The sensors of the real time computer collect data from environment, and pass them on to real time computer for processing. The computer in turn passes information (processed data) to actuators to carry out necessary work on the environment.

1.2.5. Safety-Criticality : For traditional non-real time systems safety and reliability are independent issues. However, in many real-time systems these two issues are intricately bound together making them safety-critical. A safe system is one that doesn't cause any damage even when it fails. A reliable system on the other hand, is one that can operate for long duration of time without exhibiting any failures.

1.2.6. Concurrency: A real time system [3] usually needs to respond several independent events within very short and strict time bounds. The real time system must process data from all the sensors concurrently, otherwise signals may be lost and the system may malfunction.

1.2.7. Stability: Under overload conditions, real-time systems need to continue to meet the deadlines of the most critical tasks, though the non-critical tasks may not be met. This is in contrast to the requirement of fairness for traditional systems even under overload conditions.

1.2.8. Exception Handling: Many real time systems work round the clock and often operate without human operators. When there are no human operators, taking corrective actions on a failure becomes difficult. Even if no corrective actions can be taken immediately, it is desirable that a failure doesn't result in catastrophic situations. A failure should be detected ad the system should continue to operate in a degraded mode rather than shutting off abruptly.

System Overview

Real Time Mission Software (RTMS) is the Master controller of the entire Mission of the underwater vehicle. RTMS is TI ARM controller based real time embedded system. The main functionality of RTMS software is to control and guide the vehicle as per the mission requirements. In addition to that, it also distribute the power to all the other subsystems in the vehicle such as Recording system (RS),Sensor Package(SP) Speed Control System (SCS) and Actuation System (AS) .It performs Prelaunch sequence and recovery operations also.

To meet mission requirements, RTMS interfaces with the following subsystems.

1) Recording System
2) Sensor Package
3) Actuation System
4) Submarine/mother ship
5) Speed Control System
6) Presetter
7) Pressure Transducers
8) Debug Monitor

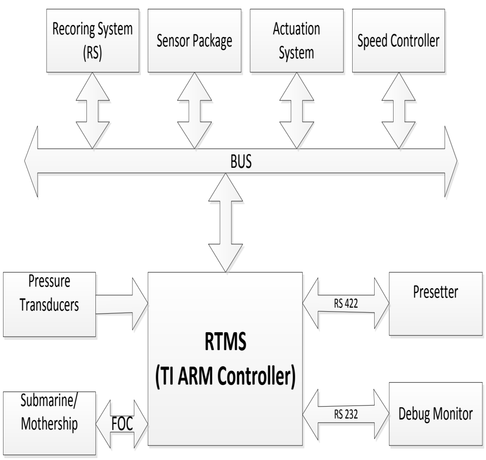


Figure 1.

Figure 1 shows how RTMS interfaces with the other subsystems in the vehicle.

II. METHODOLOGY

The complete mission software has been developed using foreground and back ground approach [4]. In foreground all interrupt service routines will be executing whereas main while loop will be executing in back ground process. In foreground two interrupt service routines will be executing i.e one for timer interrupt and second for communication bus interrupt. Whenever timer generates (ex 10ms) interrupt, subsequent soft timers are created so that it can be used in tasks scheduling in background process.

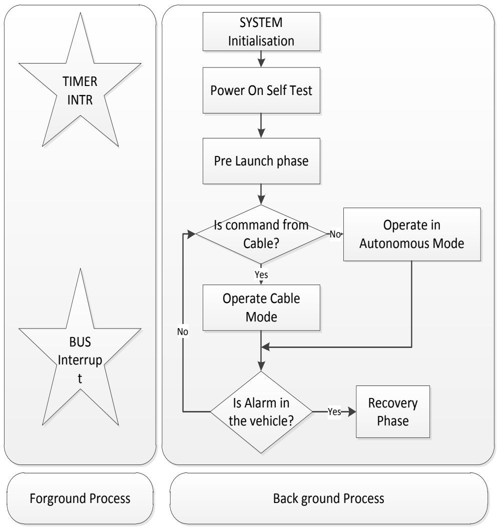
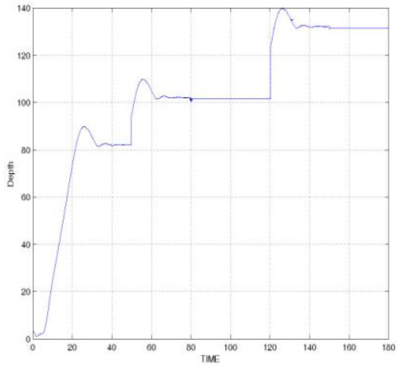


Figure 2.

Fig.2 shows how the complete RTMS is designed. In background process, System initialization module performs initialization of on board peripherals i.e Timer, Interrupt controller, ADC, DAC and serial controllers. Power On self-Test performs health check of internal structural components. In Pre Launch phase, it checks the health of on board subsystems. Then once vehicle is launched, it can be operated either in cable mode or autonomous mode.

RTMS can get commands from mother ship/submarine via fiber optic cable in turn it generates commands as per the requirement. Based on the commands vehicle change it depth,roll,Yaw and pitch as per user requirement. If Cable break happens , RTMS can operate autonomously as per preset parameters. In turn it records all critical parameters in NVRAM so that, after completion of the mission data can be retrieved from memory. The same data can be used for post trial analysis.

III. RESULTS



Figure

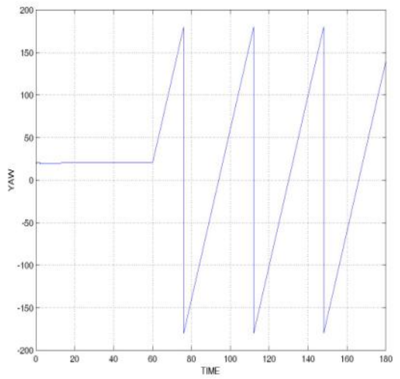


Figure 4.

Fig. 3 depicts the running depth of a vehicle in the mission. RTMS received the depth change commands from mother ship/submarine via fiber optic cable in turn the depth has been changed as per requirement. Initially vehicle depth has been set to 80m, and then commands are received from platform to change the depth to 100m and 130m respectively. The RTMS has responded the commands and corresponding control commands are fed to actuation system so that vehicle has settled to ~100m and ~130m as per user requirements.

Fig 4. Depicts the yaw plot where initially its running at ~20 deg . At 60th sec command is sent from mother ship to perform circular search in turn RTMS has received the command and performed the circular search pattern as shown in Fig 4..

IV. CONCLUSION

The proposed system has capability in operating in dual mode i.e. cable mode and autonomous mode.

The System can be used in under water applications. The system can be used in underwater survey missions such as detection and mapping of obstructions, rocks submerged wrecks.

REFERENCES

Journal Papers:

[1]. Xiaoming Qian,Peihuang Lou, An embedded control system for automatic guided vehicle,ICIRA'12 Proceedings of the 5th international conference on Intelligent Robotics and Applications - Volume Part III, Springer-Verlag Berlin, Heidelberg ©2012

[2]. J Yuh,Design and Control of Autonomous Underwater Robots : A Survey, Journal Autonomous Robots archive,Volume 8 Issue1,January 2000

Books:

[3]. RajKamal, EmbeddedSystems: Programming and Design (Tata McGraw-Hill,2011)

Chapters in Books:

[4]. Daniel W Lewis, Fundamentals of Embedded Software Where C and Assembly Meet (Ed 1st),Concurrent Software Chapter 7(Prentice Hall Inc. Upper Saddle River, New Jersey 07458 ;2002)