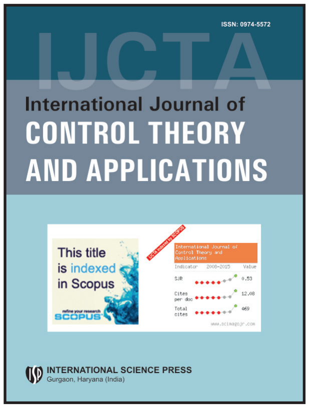

ISSN: 0974-5572
IJCTA
International Journal of
CONTROL THEORY
AND APPLICATIONS
This title is indexed in Scopus
INTERNATIONAL SCIENCE PRESS
Gurgaon, Haryana (India)

International Journal of Control Theory and Applications

ISSN : 0974-5572

© International Science Press

Volume 9 • Number 43 • 2016

Modular Data Logger Sub-System Design with Real Time Constraints for Weather Balloon Satellite

Bhushan C. Patil[a], **Amit Patwardhan**[b] **and Rabinder Henry**[c]

[a-c]Pralhad P. Chhabria Research Center, Pune-411057 India. Email: [a]bhushanpatil23@gmail.com; [b]amitpatwardhan8@gmail.com; [c]henrysal2000@gmail.com

Abstract: The main goal of the work presented here is to design a Modular Data logger system for low altitude weather balloon satellite that records temperature, humidity, particulate matters and infrequent anomalies for long durations. The data are obtained from sensors with respect to time and location. Multitasking embedded software with real time constraints have been developed to operate a data logging system based on an ARM based controller. The embedded software is implemented with combination of state machines and co-operative multitasking software. The results are obtained through Analog to Digital Converter and eight digital channels. These data are logged on to data storage system. Combination of State machine and Co-operative multitasking results in flexible multitasking system in which any number of channels can be added easily without affecting real time performance.

Keyword: Data acquisition, Real-time systems, Satellite communication, Embedded software, Microcontrollers.

1. INTRODUCTION

The Data logger is a subsystem in design of a Low Altitude Weather Balloon Satellite (LAWBS). LAWBS has been designed for monitoring temperature, humidity, particulate matters to perform urban air pollution analysis. Main objective of data logger sub-system is to record signals of interest from sensors and transfer data to main satellite system in real time. We have designed modular data logger system in which number of channels required can be reconfigured according to number of sensors connected to the system. At the same time the date collection, retrieval and storage happens in real time.

A real-time system is an application where the accuracy depends on the timelines and predictability of the outcome as well as the results based on computations. To assist the real-time data capturing, multitasking technique is adopted for software implementation that facilitates efficient inter-process communication and synchronization. A fast interrupt response time, asynchronous input and output (I/O), memory management functions, file synchronization, and facilities for satisfying timing requirement have been included in the system design [6][7].

2. SYSTEM HARDWARE

For designing data logger system we've chosen LPC2148 microcontroller because it has enough memory and flash storage for intermediate-level application, and it has a full range of advanced internal peripherals such as full-speed USB 2.0, 10bit ADCs, DAC, and standard serial buses like UART, SPI, and I2C. Though this microcontroller has too many resources than actual requirements per objectives, the system must support flexible increment in number of channels without affecting real time performance.

Complete hardware block arrangement of data logger system is shown in Figure 1. This system uses internal ADC of ARM7TDMI. Data logger system collects data from multiple ADC channels, 8 bit digital data inputs and transfers them to computing system via RS232 link [1][2].

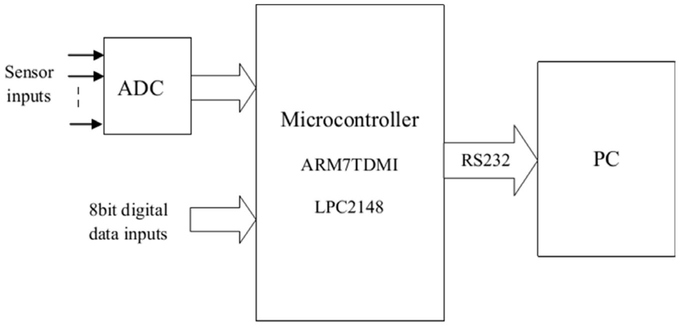

Figure 1: Hardware arrangement for data logger system

Blueboard-LPC2148 is used for designing data logger system. Functional block diagram of Blueboard is shown in Figure 2.

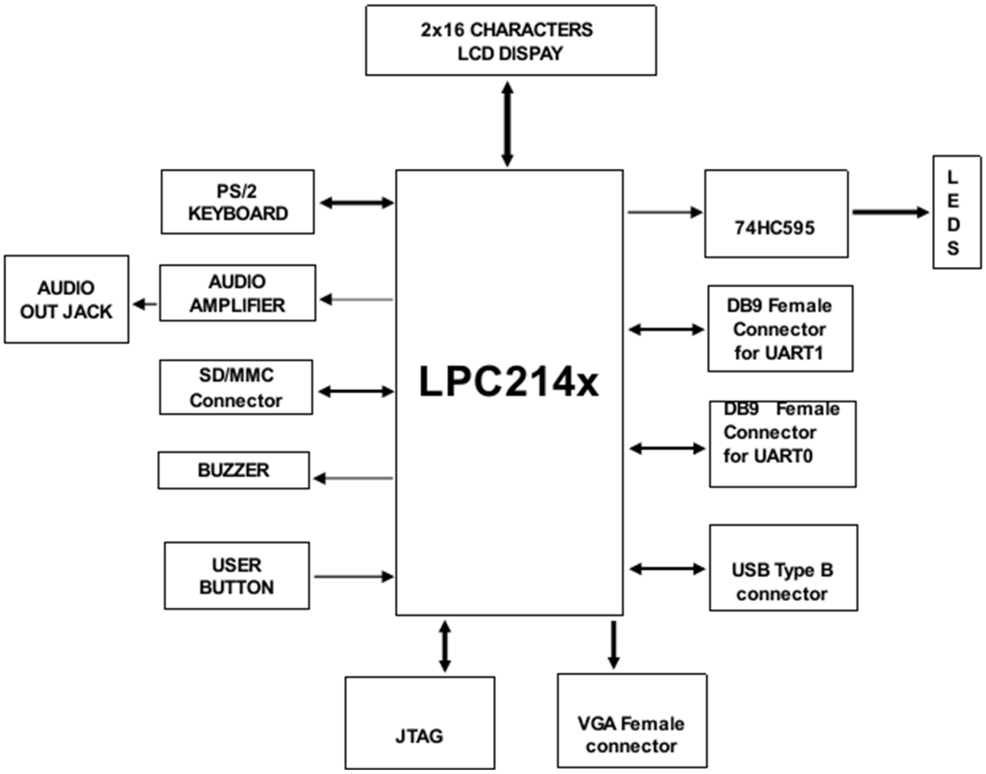

Figure 2: Functional block diagram of Blueboard

3. SYSTEM SOFTWARE

A. *Selection of System Software Design Approach:* State machine approach, co-operative multitasking and pre-emptive multitasking are more suitable for data logger's software development in-order to accomplish objectives of this design. The pre-emptive system is variable within the tasks creating completion problems. The cooperative system gives the designer the power to determine where and when the context switch occurs, but it suffers in its handling of faults. Both the systems suffer from complexity with respect to timing issues. Both require some specialized routines within the system to execute properly, and also certain special design by the designer to implement and optimize. The state machine approach uses state machines for the individual tasks with the calls to the state machine regulated by a hardware-driven timing system. Priorities are managed based on the current value in the state variables and the general state of the system. Communications are handled through a simple combination of handshaking protocols and overall system design [3][4][5].

Data logger system uses simple protocol for communication (i.e. RS232) and soft type real time system. Hence, in this design the approach is based on co-operative multitasking.

Data logger software is divided into states that take care of the following aspects:

- Communicating with the PC
- Provides status of ADC values to user
- Provides status of digital data inputs
- Handling control signals of ADC and capturing digital value generated by particular channel
- Capturing real time digital data from 8 bit bus
- LCD control and data display for on-board verification
- Configuration

B. *Software Constraints:* Normally above configuration of multiple-task workload might suggest the use of a RTOS, but in this project minimal system is implemented. The 'real world' events in this system won't be happening quicker than 1 millisecond apart and the CPU is clocked at 12MHz which equates to 12000 instructions per millisecond. So, scheduling time-based tasks at millisecond resolution enough to handle events comfortably.

C. *Software Implementation:* Final software implementation is arranged as shown in Figure 3. Arrangement of data logger software shows that scheduler is handling multiple tasks.

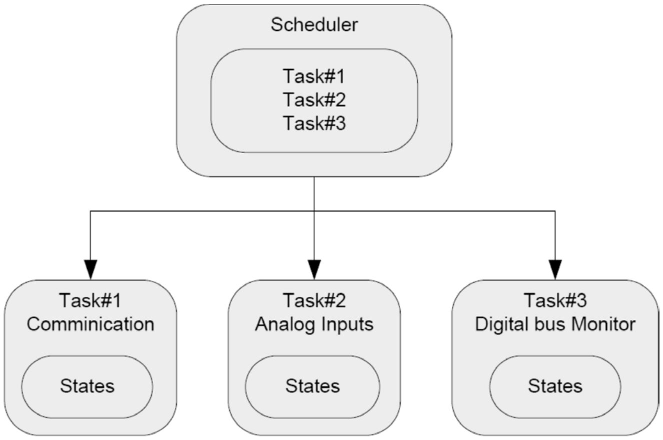

Figure 3: Software implementation

Process flow of complete data logger software implementation:

1. *Task:* Each task assigned for specific event (e.g. ADC control task is handling ADC related control signal and data generated). Each task can be queried to see if it can be run and if it can, it will be executed via a call to its running method [3][4][5].

2. *Task Scheduler:* The task scheduler is created with a fixed list of tasks. The run method iterates endlessly over the task list. Calling the run method on each turn to check whether it has to be re-run. If it does, the run method is called to execute the task. One very important note: after running a task, iteration flow of the task list gets a break and it start back from the top of the list. That gives the fixed task priority feature - if multiple tasks are runnable the earlier tasks on the list will always be dispatched first and the later tasks on the list will be lower priority. With this approach data logger provides a lightweight set of tasks that are scheduled in priority order. The code is both high performance and lightweight, two vital attributes considering the constrained environment it must operate in. Providing the various run methods are reasonably short, it will run tasks within less than one millisecond of when they are scheduled, which is perfectly adequate for a data logger system [3][4][5].

3. *Complete program flowchart:* Figure 4 & Figure 5 shows flowchart for task program & task scheduler respectively.

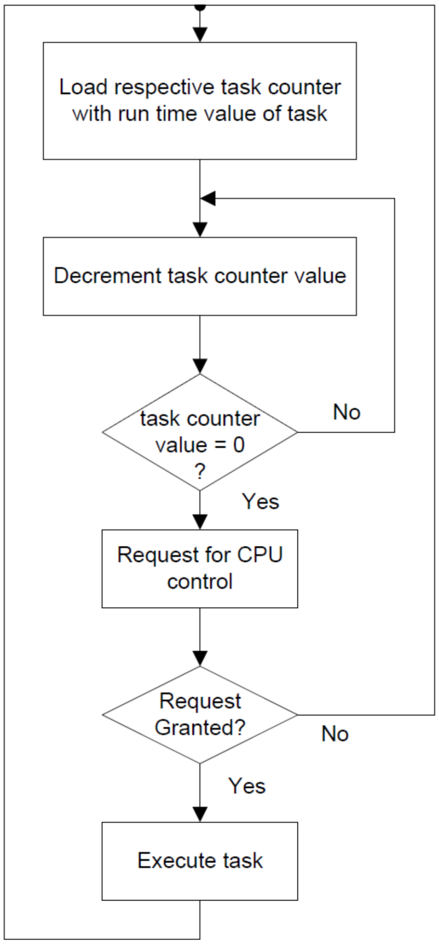

Figure 4: Flowchart for task program

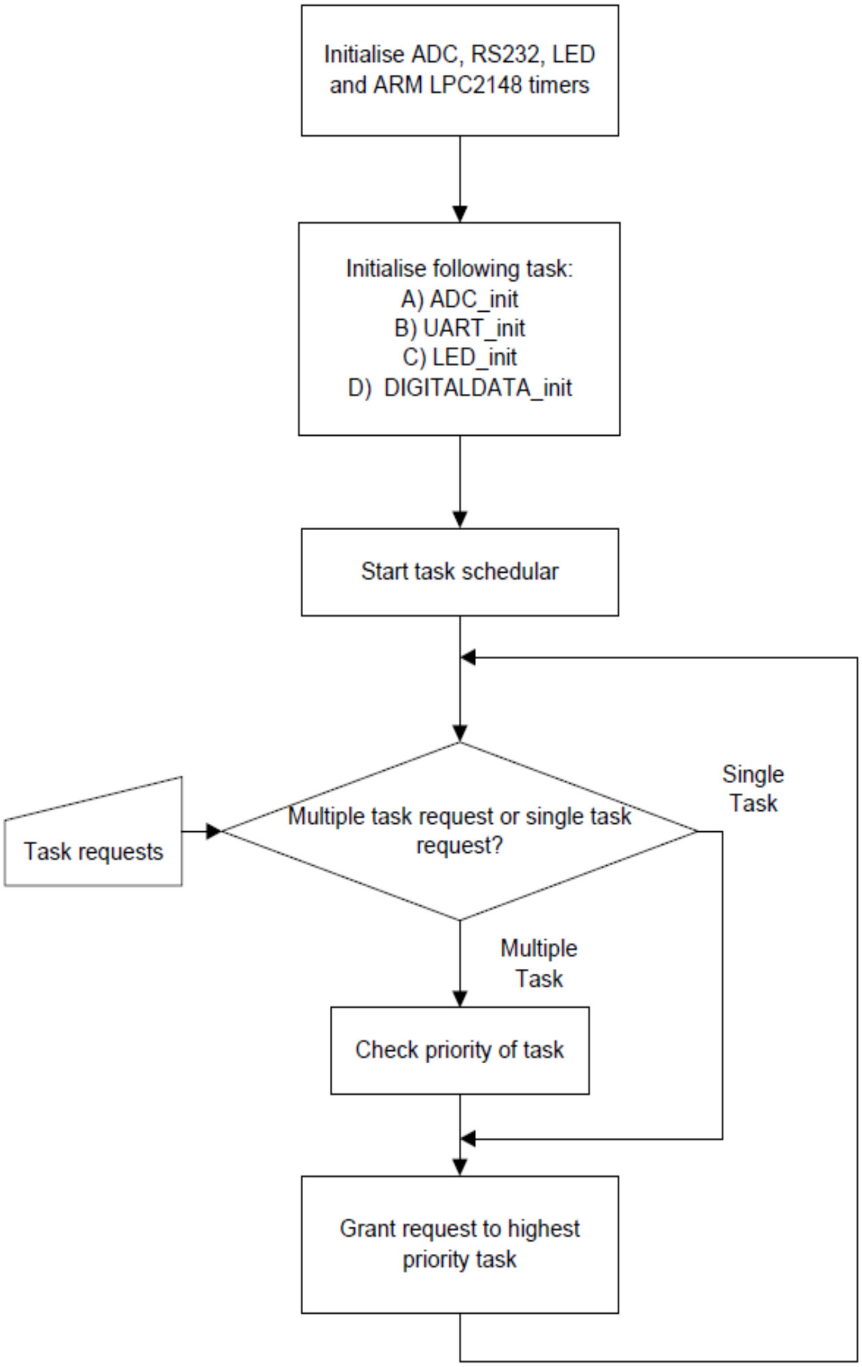
Initialise ADC, RS232, LED and ARM LPC2148 timers
Initialise following task:
A) ADC_init
B) UART_init
C) LED_init
D) DIGITALDATA_init
Start task schedular
Task requests
Multiple task request or single task request?
Single Task
Multiple Task
Check priority of task
Grant request to highest priority task

Figure 5: Flowchart for task scheduler

4. FINAL IMPLEMENTATION

Blueboard with NXP LPC2148 is used for hardware implementation. The firmware for the NXP LPC2148 is written in C language and compiled using GNU ARM tool chain. HEX file generation & programming of microcontroller was done by using make script for the GNU ARM assembler & linker [8]. Figure 6 below shows basic multitasking implementation displayed on LCD of Hello World string and ADC input for system test purpose.

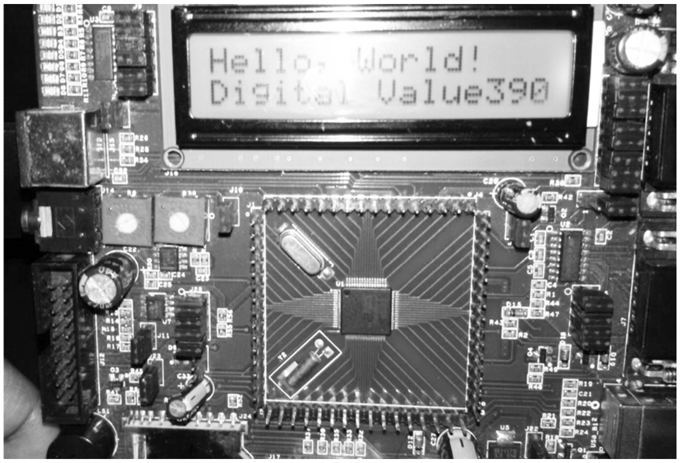

Figure 6: Basic system test

5. CONCLUSION

The data logger multitasking system is a flexible and simple solution to the collection of analog and digital data. It can be integrated with a large array of available sensors. A big advantage of the data logger software is the ease of code maintenance and improvement. The software functionality is contained in well-defined tasks that communicate via well-defined methods. In short, software implementations successfully perform the required data logging functions for analog and digital data.

REFERENCES

[1] Steve Furber, ARM system-on-chip architecture second edition, Addison-Wesley, Harlow, England, 2000.

[2] ARM Architecture Reference Manual, Copyright © 1996-1998, 2000, 2004, 2005 ARM Limited.

[3] David Stonier-Gibson, Understanding embedded microcontroller multitasking RTOS alternatives, SPLat Controls

[4] Silberschatz, Abraham; Galvin, Peter B.; Gagne, Greg (2010), Operating System Concepts (8th ed.), John Wiley & Sons (Asia).

[5] Keith Curtis, Embedded Multitasking With Small Microcontrollers, Copyright © 2006 Elsevier Inc., ISBN: 978-0-7506-7918-3

[6] Sylvia Goldsmith, A Practical Guide to Real-Time Systems Development. Prentice Hall, 1993

[7] National Instruments. Advantages of pc-based data logging, [Online] Available: http://www.ni.com/datalogger/advantages.htm, 2012.

[8] Bill Zimmerly, Install the GNU ARM tool chain under Linux, [Online] Available: http://www.ibm.com/developerworks/linux/library/l-arm-toolchain/