Convert the Resulting samples into numeric Values Based On ARM9

C.SWATI

M.TECH

Abstract: In this paper we are implementing the data acquisition system card meanwhile overcoming the disadvantages of existing projects are μC/OS-II & ARM9.we are setuping the realtime parameters that improves the constraints of the system performance and efficiency, security.and we can increase the cpu utilization from 80% to 90%.

Keywords: Data acquisition, ARM9 (S3C2440), μC/OS-II.

I INTRODUCTION

The circuit of acquisition hardware, firstly, using multi modulation circuit to modulate the 12 channel signals (including the corresponding signal conversion, anti-interference, as well as a preliminary signal processing, etc.), this makes the signal collected from multichannel meet ARM9 core requirement. After multichannel choosing and ADC sampling, signal enters the ARM9 core. In order to achieve the whole system working state real-time monitoring; controlling; and adjustment setup the host computer monitor and on-site display touch screen, show the user all the required information in the superior computer display output, or output at the on-site LCD simultaneous display. Sampling in accordance with the sampling frequency band is divided into high frequency band (300~200); intermediate frequency band (200~100); low frequency band (less than 100) (units KSPS) in total 3-band, each band formed by the four external sampling channels. ARM 9core communicates with PC using UART and RS232 convert circuit. System work power supplied by unified power source, after the realization of voltage convert, it also provided power to the PC and ARM9 cores and the corresponding working circuit. With the rapid development of IT technology, electronic technology, communication technology and automatic control technology, high-speed real-time industrial data collection have become the crucial link in electronics and industrial control technology. This paper focus on multi-task algorithms in high-speed multi-task data acquisition; meanwhile meets the requirements of real-time, effectiveness and high speed, proposed a intelligent design, based on ARM9 and μC/OS-II operating system, to achieve the task priority dynamic scheduling, the dynamic setup of system working parameters, intelligent design within the system software for low-speed peripherals, system optimization, and improved system reliability has improved internal task synchronous. As an information exchange unit, our software system should be time-sensitive, therefore, a preemptive multithreaded operation system is useful to support real time desires. MC/OS-II is a real-time preemptive multitasking embedded OS kernel, which is popular with portable, scalable and easy to use. It provides a number of key functionalities needed by embedded applications, such as multi-task, synchronization, timer management, memory management. The series of advantages existed in μC/OS-II could encourage researchers to rapidly develop a project; meanwhile, it also enables the applications to be real-time. Additionally, its security and reliability is helpful to construct robust software system. In aspect of information exchange capacity, the performance of hardware plays a decisive role; once the components are selected, only improving the software can we obtain higher system performance. In this article, we did some research on μC/OS-II, for example, improving the interruption process to reduce memory usage, detecting the overflow of stack to avoid system crash and so on. Multi task data acquisition system proved to be successful were done on μC/OS-II.

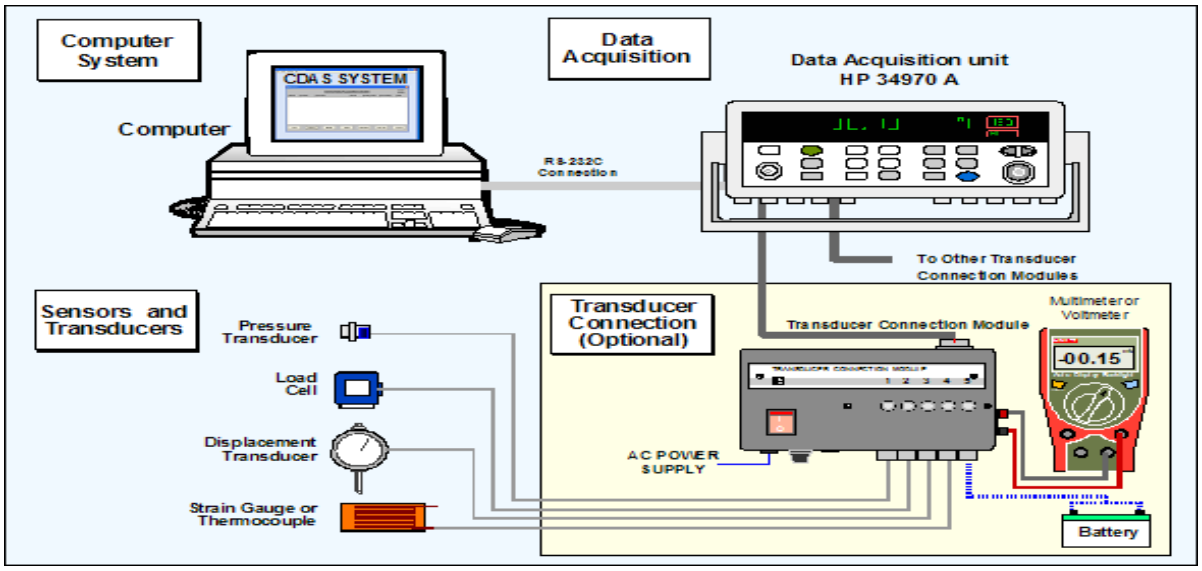

Fig1 Basic Data Acquisition system

Data acquisition is the process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer. Data acquisition systems (abbreviated with the acronym DAS or DAQ) typically convert analog waveforms into digital values for processing. The components of data acquisition systems include: Sensors that convert physical parameters to electrical signals. Signal conditioning circuitry to convert sensor signals into a form that can be converted to digital values. Analog-to-digital converters, which convert conditioned sensor signals to digital values. Data acquisition applications are controlled by software programs developed using various general purpose programming languages as BASIC, C, Fortran, Java, Lisp, Pascal. The S3C2440A is developed with ARM920T core, 0.13um CMOS standard cells and a memory compiler. Its low power, simple, elegant and fully static design is particularly suitable for cost- and power-sensitive applications. It adopts a new bus architecture known as Advanced Micro controller Bus Architecture (AMBA). The S3C2440A offers outstanding features with its CPU core, a 16/32-bit ARM920T RISC processor designed by Advanced RISC Machines, Ltd. The ARM920T implements MMU, AMBA BUS, and Harvard cache architecture with separate 16KB instruction and 16KB data caches, each with an 8-word line length. By providing a complete set of common system peripherals, the S3C2440A minimizes overall system costs and eliminates the need to configure additional components. After multichannel choice by the modulation circuit and ADC samples collection the collected 12 signals enters each internal task channel. Each acquisition channels are designed to: separate data collection, data storage, data processing and data submitted task, corresponds to an external acquisition channel, there is an independent internal data channels, and the channel data is only stored, processing and transmission in its very channel. Insofar as system scheduling, the relationship between the various internal tasks within the data channels are different sequence of execution time; the relative priorities, no data coupling. As for command scanning part, independent upper and lower machine scanning tasks complete PC monitor and lower computer touch-screen real-time monitoring. Once the user command is detected, it'll be immediately send to the command analysis and processing tasks which processing user commands. After processing, the extract command results will be sent to the user operating sampling channel, so as to run in accordance with user requirements. Dual data buffer provide double ensure for the sample smooth store mission, once the sample had achieved, There are available buffer, data can be immediately stored and then go for the next collection. This reduced the follow up data processing tasks data waiting time span, it can also quickly get access to dealing with the current sampling data and processing, sending it to the next level task in the possible shortest time. Sampling, data display, command scanning all involving peripheral equipment, which is of the slowest running task within the system as a whole, the two-sample data buffer set for the task can greatly improve system rate decline due to slow

sampling, the optimal design of display and command scanning part will be described in detail as follow. Command analysis of each band within the system .command scan function capture the user command, the user command is verified; analysis; extraction, and the extracted results will be sent to each radio frequency command wait queue. If the queue get the new command, the user command will be sent to various sampling task function belong to the queue, user command will immediately be implemented, including the channel switching; transform sampling period, change the current task priority, display specific channel data, etc. If not a new command, after waiting out, the sample task continue to work in accordance with the original way. This is also an intelligent design, by a very simple way to achieve —examining; supervision; operation; work status switch; channel switch; from single channel switch to real-time parallel collection of all channels in all the features.

a) DAQ hardware

DAQ hardware is what usually interfaces between the signal and a PC. It could be in the form of modules that can be connected to the computer's ports (parallel, serial, USB, etc.) or cars connected to slots (S-100 bus, AppleBus, ISA, MCA, PCI, PCI-E, etc.) in the motherboard. Usually the space on the back of a PCI card is too small for all the connections needed, so an external breakout box is required. The cable between this box and the PC can be expensive due to the many wires, and the required shielding. DAQ cards often contain multiple components (multiplexer, ADC, DAC, TTL-IO, high speed timers, RAM). These are accessible via a bus by a microcontroller, which can run small programs. A controller is more flexible than a hard wired logic, yet cheaper than a CPU so that it is permissible to block it with simple polling loops. For example: Waiting for a trigger, starting the ADC, looking up the time, waiting for the ADC to finish, move value to RAM, switch multiplexer, get TTL input, let DAC proceed with voltage ramp.

Fig 1.0 DAQ Hardware

b) *DAQ SOFTWARE*

DAQ software is needed in order for the DAQ hardware to work with a PC. The device driver performs low-level register writes and reads on the hardware, while exposing a standard API for developing user applications. A standard API such as COMEDI allows the same user applications to run on different operating systems, e.g. a user application that runs on Windows will also run on Linux.

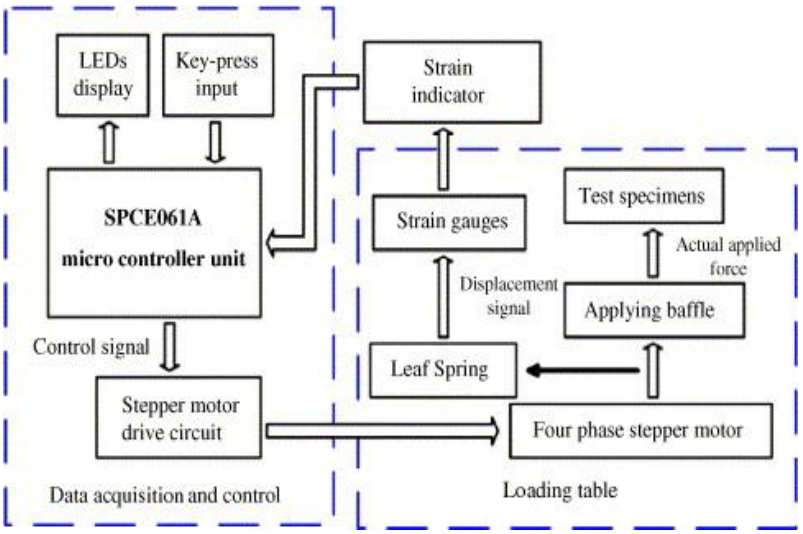

Fig 1. 2 DAQ Software

II SCHEDULING ALGORITHM OF THE INNER SOFTWARE

a) Task priority set up and sample task priority scheduling

μC/OS-II operating system is preemptive priority-based operating system, all tasks must be independent and priority unique[1]. Commands and data scanning were set to the highest priority and lowest priority respectively, the highest priority for scan function ensures that at any time responding to user commands, due to the slow running speed of display task, set it to lowest priority. As long as the system is reasonable designed, appropriate avoid low-priority tasks hungry can achieve the user valid information display. Inner priority rules set up as: among low frequency, medium frequency, high frequency band the priorities are in descending order. Apparently, low frequency's sampling period

is longer than high frequency, within the idle time of low frequency data acquisition, system can smoothly switch the right using CPU to other tasks. Priority of each task within each band descending in order: from sampling to data storage, data processing, to data submission To make the whole system to achieve dynamic priority scheduling, it can also modify other sampling. In software design, all the priority of each band sampling task and the sampling period are set up as global variable (their format are defined as "volatile" and been given independent variables' memory address), they will be set before entering the operating.

b) Task time limit set

In order to optimize time scheduling for the entire system, firstly state a few internal time work names, i386 system and the Linux2.5 operating system kernel beat rate are set to 1000 HZ [2], here the μC/OS-II operating system, the dispatch time OS_TICKS_PER_SEC also set to 1000. That is the task scheduling to the operating system in units of ms[2]. ARM9 kernel CPU operating frequency is set to 400 MHz, that is to say, the CPU machine cycles in units of ns. In μC/OS-II system, task switching, scheduling and delay are in unit of the beat rate system[1] , while the internal code is running in unit of machine cycle. Limit for internal task time unit can be beat rate, and can also be real time basis, obtain an absolute time difference. Within a single sampling channel only needs to set the task time limit. In order to achieve intelligent, using the waiting delay of message queue replace the sampling tasks time limit and the delay is the sampling period of channel. Other tasks including open up memory; data processing, data submit are all in turn waiting for superior message; limited to superior message and do not set the task time limit. The absolute time difference of bv display tasks time limit set in unit of ms.

IV CONCLUSION:

We implemented the data acquisition system with the practical results of high frequency, medium frequency, low frequency limits were tested. At a time all the tasks are running and priority wise tasks will be displayed in the HyperTerminal tool. Achieved sampling channel priority dynamic scheduling, real-time parameters acquisition have been set up here. All these determined in the case of cpu utilization from 80% to 90%.

REFERENCES

[1]Li, S. , Jiarong R luo. , Yichun C. Wu. , Guiming M. li. , Feng Wang and Yong Wang, "Continuous and Real – Time Data Acquisition Embedded System for EAST", Vol. 57, No. 2, April 2010.

[2]Jacker, K. and Mckinney, J, "TkDAS – A data acquisition system using RTLinux, COMEDI, and Tcl/Tk", in Proc. Third Real – Time Linux workshop 2001. [Online]. Available: The Real – Time Linux Foundations: 2001/papers. html.

[3]Siever, A. , Weber, S. , Figgins and Love, R. "Linux in a Nutshell". Sebastopol, CA: O' Reilly, 2005.

[4]Tao Lin, Hai Zhao, Jiyong Wang, Ganglia Han and Jindong Wang, "An Embedded Web server for [5]Equipment", School of Information Science & Engineering, Northeastern University, Shenyang, Liaoning, China.

[6]Zhou, Q. , Wu, W. , and Ma, Y. "The Embedded Data Acquisition System for Mossbauer Spectrum", in Proc. Third Real – Time Linux Workshop, Embedded Linux Expo Conf. Real – Time Embedded Comput. Conf. , Milan, Italy, Nov. 2001, pp26-29.

[7]The ARM Processor website. [Online]. Available: http://www. friendlyarm. comMPX 2010 Datasheet, 2007. [Online]. Available: http://www. atmicroprog.