Design and Implementation of Free RTOS Based online Data Acquisition and Controlling System Using Cortex M3Core

Mohd Arshad
M.Tech in Embedded Systems
CVR College of Engineering (Autonomous)
Hyderabad (AP), India

Dhruva R. Rinku
Sr. Asst. Professor, ECE Dept.
CVR College of Engineering (Autonomous)
Hyderabad (AP), India

Abstract: Data Acquisition and Controlling System is one of the promising trends in the era of computing in today's system automation industry and control. The proposed project is one such attempt of designing FreeRTOS based online data acquisition and controlling system using Cortex-M3 core. In this project we will develop Ethernet device drivers for Cortex-M3 core to transmit the monitored sensor data to internet. The System can complete the remote monitoring and maintenance operations of equipment through the network using Web browser. By introducing Internet into control network, it is possible to break through the spatial temporal restriction of traditional control network and effectively achieve remote sensing, monitoring and real-time controlling of equipments. The main essence of this project is to design and implement a Data acquisition and controlling System using ARM CORTEX M3 CORE and TCP/IP Ethernet connection for industrial controlling applications. The real time analog voltages are converted into corresponding digital values using the ADC pins inbuilt in LPC 1768 Cortex M-3 and transfer them to the internet through Ethernet.

Keywords: LPC 1768 H-Plus Ex &Board (CORTEX-M3); GLCD; Free RTOS; sensors

1. INTRODUCTION

Computer communication systems and especially the Internet are playing an important role in the daily life. Using this knowledge many applications are imaginable. Home automation, utility meters, security systems can be easily monitored using either special front-end software or a standard internet browser client from anywhere around the world. Web access functionality is embedded in a device to enable low cost widely accessible and enhance user interface functions for the device. A web server in the device provides access to the user interface functions for the device through a device web page. A web server can be embedded into any appliance and connected to the Internet so the appliance can be monitored through the browser in a desktop.

Temperatures, Pressure, displacement, motion are the most often measured quantities. For example, some processes work only within a narrow range of temperatures; certain chemical reactions, biological processes, and even electronic circuits perform best within limited temperature ranges. So, it is necessary to measure the temperature and control if it exceeds some certain limit to avoid any misbehavior of the systems. To accurately control process temperature without operator involvement, a temperature control system relies upon a controller, which accepts a temperature sensor. Temperatures, Pressure, displacement, motion are the most often measured quantities. For example, some processes work only within a narrow range of temperatures; certain chemical reactions, biological processes, and even electronic circuits perform best within limited temperature ranges. So, it is necessary to measure the temperature and control if it exceeds some certain limit to avoid any misbehavior of the systems.

2. REQUIREMENT TO THE DATA ACQUISITION SYSTEM

The main component use in the Data Acquisition System is the ADC unit connected to the POT, temperature sensor, pressure sensor and humidity sensor, Gas sensor. The ADC is used to take analog inputs from sensors and is concerted into digital values. As we know the ADC in LPC1768 is of 12-bit resolution it will give a digital value of range 0 to 4095.we need to convert these digital values in sensors readings like temperature, volts etc., this is done by using some formulas. These values are continuously display on GLCD. Based on the sensor values the total Data Acquisition System works, the controlling of these sensor values when it exceeds the maximum limit is done by motors through relays.

2.1 H/W AND S/W SPECIFICATIONS

Hardware:

- LPC 1768 H-Plus Ex & Header Board
- Graphical LCD Display
- Sensors –Temperature, Pressure, humidity, And Gas
- J-Tag Debugger / Serial Cable
- L293D(motor driver IC),1000rpm Motors, Relays

Software:

- FLASH MAGIC Philips Serial ISP programming utility.
- KEIL µVision 4IDE MDK Embedded Cross Compiler

2.2 BLOCK DIAGRAM OF DATA ACQUISITION SYSTEM

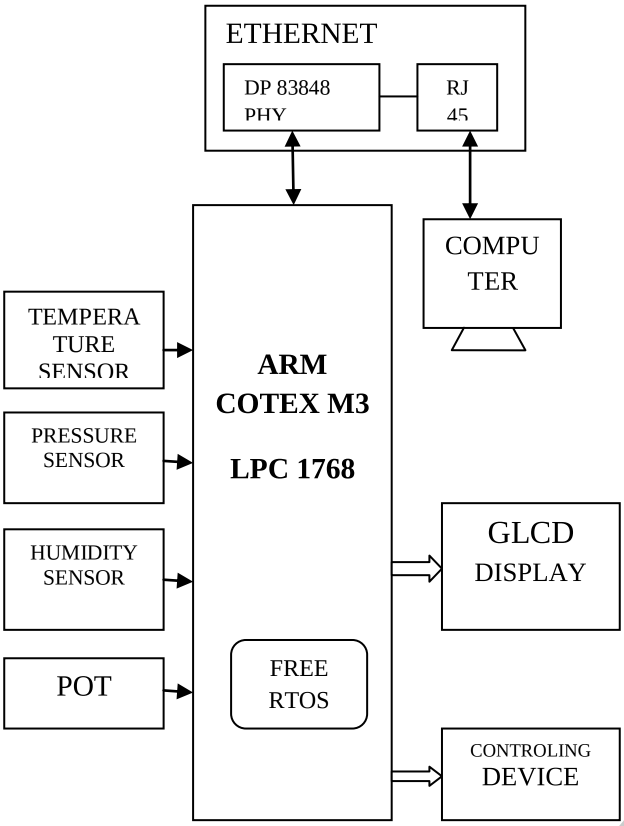

Figure1: Block Diagram of Data Acquisition System

3 ADC CONFIGURATION

An analog-to-digital converter (abbreviated ADC, A/D or A to D) is a device that converts a continuous physical quantity (usually voltage) to a digital number that represents the quantity's amplitude. The conversion involves quantization of the input, so it necessarily introduces a small amount of error. The inverse operation is performed by a digital-to-analog converter (DAC). Instead of doing a single conversion, an ADC often performs the conversions ("samples" the input) periodically. The result is a sequence of digital values that have converted a continuous-time and continuous-amplitude analog signal to a discrete-time and discrete-amplitude digital signal.

An ADC may also provide an isolated measurement such as an electronic device that converts an input analog voltage or current to a digital number proportional to the magnitude of the voltage or current. However, some non-electronic or only partially electronic devices, such as rotary encoders, can also be considered ADCs.

Figure 2:Analog to Digital converter

3.1 UART CONFIGURATION

The UART, universal asynchronous receiver / transmitter, is mainly used for the communication between computer systems. The communication can be in full duplex mode or half duplex mode. At the sender side, the data is given to the UART in bytes and the UART then transmits them as sequential individual bits. At the receiver side, the sequential bits received from the serial link are assembled into bytes and placed into the receiver register waiting for reading. To perform the function of converting between serial and parallel form, shift registers are used in the UART. The UART is of the asynchronous transmission type which means that there is no clock signal between the two communicated parties, instead the communication configuration (speed, number of data bits, parity bit or not, number of stop bits) is agreed in advance and special bits for the synchronization are transmitted?

Before the transmission of a data word, a special bit, the start bit, is transmitted to inform the receiver and also to synchronize the receiver's clock. The difference between the two clocks should not exceed 10 percent during the data bit transmissions. After that, the data bits are sent from the least significant bit to the most significant bit. And then, a parity bit which is used for the error checking may be sent and finally, one or more stop bits, which signal the end of the word, are sent.

3.2 ETHERNET TERMINOLOGY

Ethernet follows a simple set of rules that govern its basic operation. To better understand these rules, it is important to understand the basics of Ethernet terminology.

Medium; **Segment**; **Node**; and **Frame**

Frame: The nodes communicate in short messages called frames, which are variably sized chunks of information.

3.3 ETHERNET HARDWARE DESCRIPTION

The correct way to plug the connector is given in the figure. Press the connector in the direction shown and the connector will lock up properly when it is fully connected. An Ethernet straight through cable is used for testing. The recommended connection of the cable is also given.

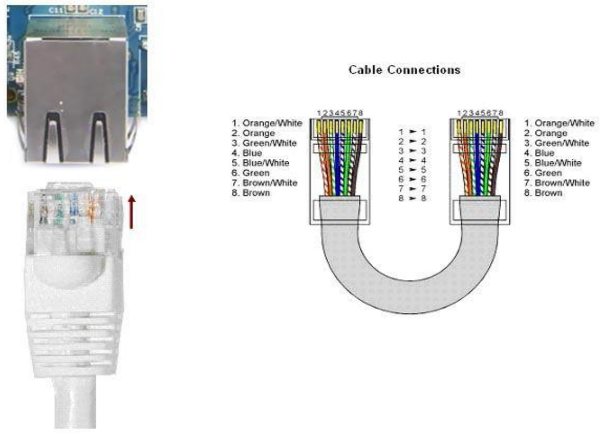

Figure3:Ethernet cable

4 FREE RTOS PROGRAMMING

Free RTOS is a free and open source Real-time Operating system developed by Real Time Engineers Ltd. Its design has been developed to fit on very small embedded systems and implements only a very minimalist set of functions: very basic handle of tasks and memory management, just sufficient API concerning synchronization, and absolutely nothing is provided for network communication, drivers for external hardware, or access to a file system. However, among its features are the following Characteristics: preemptive tasks, a support for 23 microcontroller architectures1 by its developers, a small footprint2 (4.3Kbytes on an ARM7 after compilation), written in C and compiled with various C compiler (some ports are compiled with gcc, others with openwatcom or borland c++). It also allows an unlimited number of tasks to run at the same time and no limitation about their priorities as long as used hardware can afford it. Finally, it implements queues, binary and counting semaphores and mutexes.

As there are several reasons for a task not to be running, the "Not running" state can be expanded as shows Figure 8.1. A task can be preempted because of a more priority task, because it has been delayed or because it waits for an event. When a task can run but is waiting for the processor to be available, its state is said "Ready". This can happen when a task has it needs everything to run but there is a more priority task running at this time. When a task is delayed or is waiting for another task (synchronisation through semaphores or mutexes) a task is said to be "Blocked". Finally, a call to vTaskSuspend() and TaskResume() or xTaskResumeFromISR() makes the task going in and out the state "Suspend".

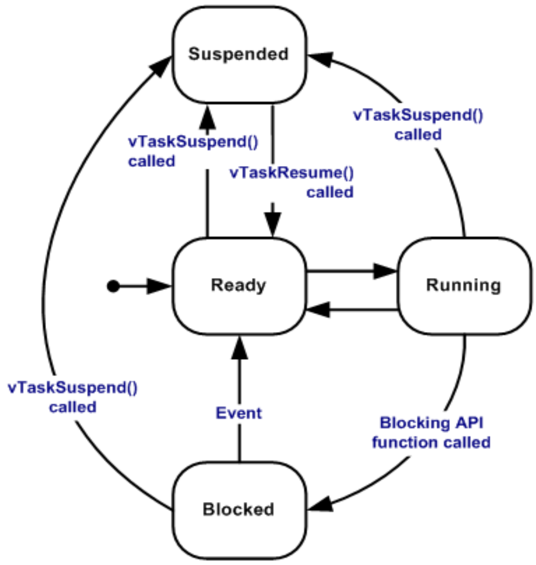

Figure4 Life cycle of a task

It is important to underline that a if a task can leave by itself the "Running" state (delay, suspend or wait for an event), only the scheduler can "switch in" again this task. When a task wants to run again, its state turns to "Ready" an only the scheduler can choose which "Ready" task is run at a given time.

5 HARDWARE IMPLEMENTATION RESULTS

In this project the results we are going to see in two parts. Initially, shown the results regarding GLCD. Figure5 shows total hardware kit and GLCD display. The Hardware unit consist of LPC 1768 Header Board, sensors, GLCD, Power supply unit, Motor.

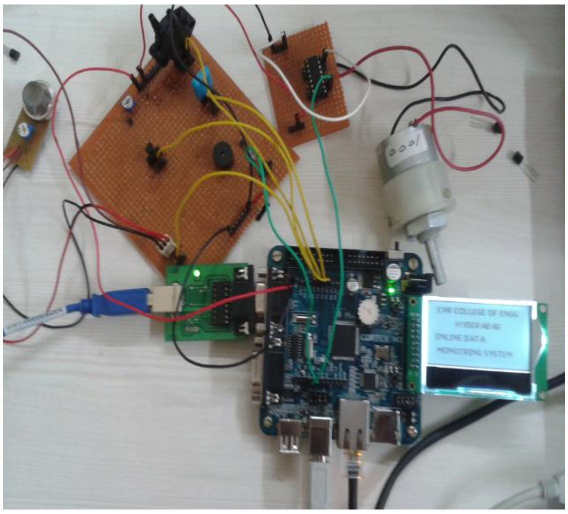

Figure5: Interfacing of Components.

The second part shows results regarding Ethernet. Figure6 shows project name, components name, sensors name

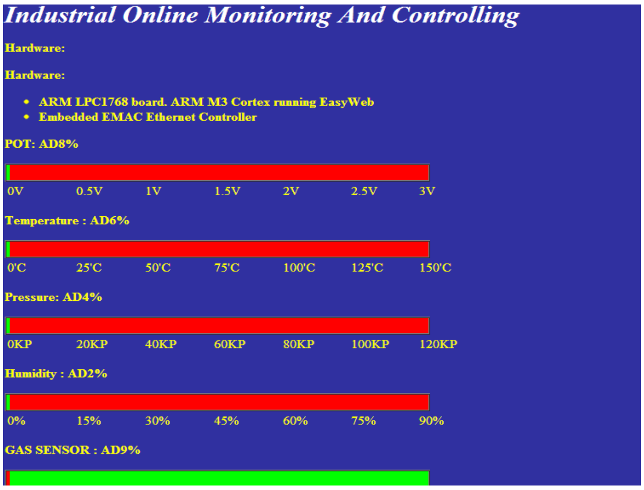

Figure6: Initial sensor value on webpage

The real time analog voltages are converted into corresponding digital values using the ADC pins inbuilt in LPC 1768 and transfer them to the GLCD. Figure7 shows the results regarding sensor values.

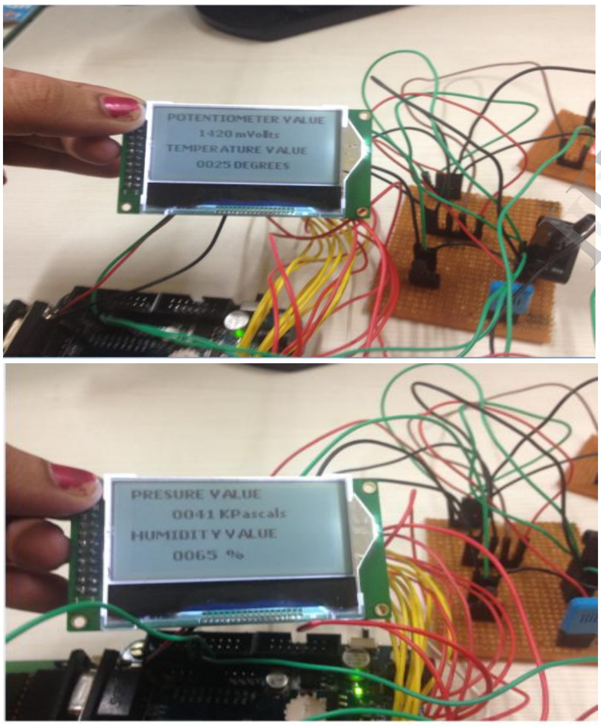

Figure7: Initial sensor value on GLCD

The real time analog voltages are converted into corresponding digital values using the ADC pins inbuilt in LPC 1768 and transfer them to the webpage. Figure8 shows the results regarding sensor values.

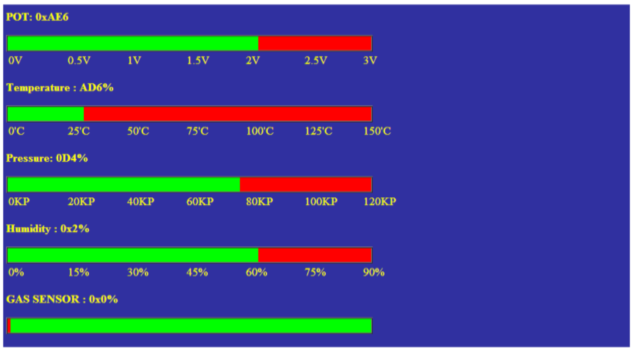

Figure8: Displaying corresponding values of the sensors

REFERENCES

[1] Alen Rajan and Aby K.Thomas; "ARM Based Embedded Web Server for Industrial Applications," *International Conference on Computing and Control Engineering*, April2012.

[2] Junhua Yang; Zhien Shang and Tao XinG, "Intelligence Monitoring System Based on ARM and Information Fusion," *International Conference on Electric Information and Control Engineering*, pp.487- 490, April 2011.

[3] Karia, D.C., Adajania, V., Agarwal, M and Dandekar, S.; "Embedded Web Server Application Based Automation and Monitoring System," *International Conference on Signal Processing, Communication, Computing and Networking Technologies*, pp.634-637, July 2011.

[4] Manivannan, M and Kumaresan, N; "Design of On-line Interactive Data Acquisition and Control System for Embedded Real Time Applications," *International Conference on Emerging trends in Computer and Information Technology*, pp.551-556, March 2011.

[5] Zhao Ruimei and Wang Mei; "Design of ARM-based Embedded Ethernet Interface," 2nd *International Conference on Computer Engineering and technology*, pp.V4-268-V4-270, April 2010.

[6] Zhan mei-qiong and Ji chang-peng; "Research and Implementation of Embedded Web Server," *Inetrnational Conference on Multimedia and Information Technology*, pp.123-125, Dec 2008.

[7] Rui Yang, Hong Cai and Ming zhang; "Research and Implement of Ethernet Interface Based on Embedded System," *Second International Symposium on Computational Intelligence and Design*, pp.288-291, Dec 2009.

[8] Wu Min-hua; "Research for the Embedded Web Server," *Microwave Conference*, pp.776-779, Sept 2008.

[9] Masato Shimano, Futoshi Okazaki, Yoshihiro Saito, Akiya Fukui,Takako Nonaka and Tomohiro Hase; "Small, Embedded Web Server for Home Appliances with Embedded MPU and Real-time Operating Syatem," pp.1-3, June 2007.

[10] Wu Xiguang, Li Bonian, Zhao Likai and Zhang Minghu; "An embedded real-time remote monitoring system based on B/S mode," *international Conference on Mechatronic Science, Electrical Engineering and Computer*, pp.2135-2138, Aug 2011