Abstract – Design of on-line embedded web server is a challenging part of many embedded and real time data acquisition and control system applications. The World Wide Web is a global system of interconnected computer networks that use the standard Internet Protocol Suite (TCP/IP) to serve billion of users worldwide and allows the user to interface many real time embedded applications like data acquisition, Industrial automations and safety measures etc. This paper approached towards the design and development of on-line Interactive Data Acquisition and Control System (IDACS) using ARM based embedded web server. It can be a network, intelligent and digital distributed control system. Single chip IDACS method improves the processing capability of a system and overcomes the problem of poor real time and reliability. This system uses ARM9 Processor portability with Real Time Linux operating system (RTLinux RTOS) makes the system more real time and handling various processes based on multi tasking and reliable scheduling mechanisms. Web server application is ported into an ARM processor using embedded ‘C’ language. Web pages are written by Hyper text markup language (HTML); it is beneficial for real time IDACS, Mission critical applications, ATM networks and more.

Keywords - Embedded ARM9 Processor, Real Time Linux Operating system (RTLinux RTOS), Embedded web server, Interactive data acquisition and control system (IDACS).

I. INTRODUCTION

Online Interactive Data Acquisition and Control system plays the major role in the rapid development of the fast popularization and control in the field of measurement and control systems. It has been designed with the help of many electrical, electronic and high voltage equipments; it makes the system more complicated and not reliable. This paper approaches a new system that contains inbuilt Data Acquisition and Control system (DACS) with on-line interaction. It makes the system more reliable and avoids more complication. It is the great demand in consumer applications and many industries. The design of very fast data acquisition in plasma discharge application was discussed in [1]; this system replaces various complex cables which are used for acquisition and it uses FPGA and ARM processor for data acquisition and digital diagnosis. There are various digital DAC systems are available for the substitution of multisite job operation. A single worker can interact with the machine and collect various data from ongoing work in a single work station. The simplest design of data acquisition system is detailed in [2], which is based on Linux Operating system [3]; it is the popular choice for many embedded real time applications and PC systems. The design of flexible and networked data acquisition architecture was approached in [4], where the software resources are stored in local memory to avoid the level of resource usage and increases system’s efficiency. This system process the client based on dynamic manner by server response and it maintains separate data base with DAC controller. In [5] advanced traffic survey mechanism uses data collection process for post processing of vehicle’s position. Signal conditioning is the major part of any data acquisition unit. High level integration architecture was discussed in [6]; it allows signals to be conditioned, simultaneously acquired according to the external clock and triggers processed and transferred data to real time servers. Signal measurement from astrophysical sources is described in [7]; where the shared memory and internet protocols are used for data handling and process from remote users. It was developed with Global Positioning System (GPS) and Environmental monitoring system. Similarly depends on industry and its location General Packet Radio Service (GPRS) also used for data transmission through on-line. But this paper doesn’t use GPRS and GPS systems for data uploading into internet. It reduces the system complexity and effective for all kind of real time applications. Every real time embedded system should be run by real time operating systems. Even a small 8-bit microcontroller has the portability with RTOS is developed in [8]. In this paper Real time Linux Operating system is ported in ARM9 processor. Generally all ARM9 processors have the portability with any kind of higher end RTOSes. This RTLinux RTOS is very effective for many embedded
applications [9] & [10]. Here the embedded web server application is developed and ported into ARM9 with this setup. This single ARM board has been act as data acquisition unit, control unit, embedded web server and self diagnosis. All processes are allocated with essential resources and associated with reliable scheduling algorithms and internet protocols followed by ARM processor. This miniaturized setup reduces the complexity & size of system.

![Diagram of system overview](image1.png)

**Figure 1. System overview**

Fig.1 shows the overview of IDAC system. Every client can access the industry directly without any interaction with additional server and modules. IDACS shows Intelligent Data Acquisition and Control System. This system contains single ARM9 processor which is portable with Real Time Linux RTOS. ARM processor is the heart of this work. It handles two modes at same time, DAC and Web server. During DAC mode Processor can measure signals which are coming from various external sources and applications. And it can control the industry machineries by the control instruction sent by client via embedded web server. During signal measurements Analog to digital converter is very important, because almost every external source is giving analog signal only. While converting these analog to digital processor has to handle asynchronous interrupts. This system uses RTLinux so it can handle many interrupts in an efficient manner because RTLinux has preemptive kernel with required privilege levels. Similarly during web server mode processor will handle client request and response to the particular client by sending web pages, client can interact the industry by giving instruction in web page on its own web browser. This setup can be suitable for inter communication with other nodes via Ethernet and higher end ports. Ethernet programming and execution is very easy and adaptable with various applications. Embedded web pages are designed by HTML language.

II. **SYSTEM DESIGN**

Hardware design, Software design and Porting are the entire important steps in whole system design.

A. **Hardware design of the system**

1) **IDACS Design:**

IDACS design is the major part in hardware. ARM9 processor is a centre core of this system. The general hardware structure of the IDACS is shown in Fig 2. The online intelligent data acquisition and control system based on embedded ARM platform has high universality, each acquisition and control device equipped with 24-way acquisition/control channels and isolated from each other. Each I/O channel can select a variety of electrical and non electrical signals like current, voltage, resistance etc., Digital acquisition are done by special ADC. The measured data are stored in external memory in which the memory is act as a data base during web server mode. The ARM processor directly supports the Ethernet service and RS485 communication. Hence the data has been stored and controlled by some other PCs or network via RS485 & Ethernet. ARM processor has internal I2C module. So it has the ability to communicate with any other peripherals.

![Diagram of general structure of the IDACS](image2.png)

**Figure 2. General Structure of the IDACS**

I2C is the wired communication protocol to communicate with other processor or peripherals tho two wired link. This system has 128*64 LCD to display the information and measured parameters which makes the debugging and modification of the parameter easy. The Analog to digital interfacing module is independent with the embedded system, which is beneficial to the system maintenance and upgrade. As the embedded Ethernet interface makes the remote data exchange between the applications become very easy.
2) Analog to Digital Converter:

Fig 2. uses 16bit ADC chip AD7715. This is digital chip having I²C module internally. It has the ability to transfer the converted digital data to ARM processor. It needs only five lines, which are DOUT – Data output, DRDY – Data ready, DIN – Data Input, CS – Chip select and SCLK – system Clock. Converted digital data will be sending out by DOUT pin of the chip. This ADC chip is driven by 2.4576MHz crystal. It contains separate Reference signals Ref+ and Ref- and separate Analog input channels AIN+ and AIN-. During communication with ARM processor this ADC chip should be synchronized with the processor’s clock.

3) RS485 Communication:

RS-485 is a telecommunications standard for binary serial communications between devices. It is the protocol or specifications that need to be followed to allow devices that implement this standard to communicate with each other. This protocol is an updated version of the original serial protocol known as RS-232. While the original RS-232 standard allowed for the connection of two devices through a serial link, RS-485 allows for serial connections between more than 2 devices on a networked system.

The general network topology of RS485 is shown in Fig.3. Here N nodes are connected in a multipoint RS485 network. For higher speeds and longer lines, the termination resistances are necessary on both ends of the line to eliminate reflections. Use 100 Ω resistors on both ends. The RS485 network must be designed as one line with multiple drops, not as a star. RS-485 standard specifies up to 32 drivers and 32 receivers on a single (2-wire) bus. In this DACS system the RS485 communication is used to transfers the data between remote DACS to Embedded controller vice versa. New technology has since introduced "automatic" repeaters and high-impedance drivers and receivers such that the number of drivers and receivers can be extended to hundreds of nodes on a network. RS-485 drivers are now even able to withstand bus contention problems and bus fault conditions.

A RS-485 network can be constructed as either a balanced 2 wire system or a 4 wire system. If a RS-485 network is constructed as a 2 wire system, then all of the nodes will have equal ranking. A RS-485 network constructed as a 4 wire system, has one node designated as the master and the remaining nodes are designated as slaves. Communication in such a system is only between master and slaves and never between slaves. This approach simplifies the software protocol that needs to be used at the cost of increasing the complexity of the wiring system slightly.

B. Software design of the system

1) Real Time Linux:

RTCore is a POSIX 1003.13 PE51 type real-time kernel, something that looks like a multithreaded POSIX process with its own internal scheduler. RTCore can run a secondary operating system as a thread, using a small virtual machine to keep the secondary system from disabling interrupts. This is a peculiar model: a UNIX process with a UNIX operating system as a thread, but it provides a useful avenue to modularity. RTLinux is RTCore with Linux as the secondary kernel. RTCore with BSD UNIX as the secondary kernel. Real-time applications run as real-time threads and signal handlers either within the address space of RTCore or within the address spaces of processes belonging to the secondary kernel. Real-time threads are scheduled by the RTCore scheduler without reference to the process scheduler in the secondary operating system. The secondary operating system is the idle thread for the real-time system. The virtual machine virtualizes the interrupt controller so the secondary kernel can preserve internal synchronization without interfering with real-time processing. Performance is adequate to allow standard PC and single board computers to replace DSPs in many applications.

Figure 4. RTLinux Run time Model

Linux Tasks

Real time tasks

Linux Kernel

RTLinux

Hardware

Figure 4. RTLinux Run time Model
Unlike Linux, RTLinux provides hard real-time capability. It has a hybrid kernel architecture with a small real-time kernel coexists with the Linux kernel running as the lowest priority task. This combination allows RTLinux to provide highly optimized, time-shared services in parallel with the real-time, predictable, and low-latency execution. Besides this unique feature, RTLinux is freely available to the public. As more development tools are geared towards RTLinux, it will become a dominant player in the embedded market. RTLinux is a typical dual-kernel, one is Linux kernel, which provides various features of general purpose OS, other one is RTLinux kernel, which support hard real time capability. Fig 4 illustrates the RTLinux architecture.

2) RTLinux Porting:

Real time modules (Declarations)

```c
#include <rtl.h>
#include <time.h>
#include <rtl_sched.h>

pthreads_t tasks[2];
void *idacs_thr(void *arg);
void *webserv_thr(void *arg);
```

RTLinux Modules (Init & Cleanup)

```c
int init_module(void)
{
    pthread_create(&tasks[0], NULL, idacs_thr, NULL);
    pthread_create(&tasks[1], NULL, webserv_thr, NULL);
    return 0;
}

void cleanup_module(void)
{
    pthread_cancel(tasks[0]);
    pthread_join(tasks[0], NULL);
    pthread_cancel(tasks[1]);
    pthread_join(tasks[1], NULL);
}
```

RTLinux Modules (RT Threads)

```c
void *idacs_thr(void *arg)
{
    pthread_make_periodic_np(pthread_self(), gethrtime(), 500000000);
    while(1)
    {
        pthread_wait_np();
        rtl_printf("IDACS\n");
    }
    return 0;
}
```

3) Real Time Interrupts:

Hard Interrupts:

```c
rtl request irq(3) and
rtl free irq(3)
```

these functions are used for installing and uninstalling hard interrupt handlers for specific interrupts.

```c
#include <rtl_core.h>
int rtl_request_irq(unsigned int irq,
                    unsigned int (*handler) (unsigned int,
                    struct pt_regs *));
int rtl_free_irq(unsigned int irq);
```

Soft Interrupts:

```c
int rtl_get_soft_irq(
    void (*handler)(int, void *, struct pt_regs *),
    const char * devname);
void rtl_global_pend_irq(int ix);
void rtl_free_soft_irq(unsigned int irq);
```

The rtl get soft irq(3) function allocates a virtual irq number and installs the handler function for it. This virtual interrupt can later be triggered using rtl global pend irq(3). rtl global pend irq is safe to use from realtime threads and realtime interrupts. rtl free soft frees the allocated virtual interrupt.

4. TCP/IP in RTLinux:

LWIP is an implementation of the TCP/IP stack “Use of the LWIP stack is to reduce memory usage and code size, making LWIP suitable for use in small clients with very limited resources such as embedded systems”. Improvements achieved by LWIP in terms of processing speed and memory usage have been performed by means of violating the TCP/IP layers. Most TCP/IP implementations keep a strict division between the application layer and the lower protocol layers. As the barrier between the kernel and the application processes is not a strict protection, a more relaxed scheme for communication between the application and the lower layer protocols can be performed by means of shared memory.

In particular, the application layer can be made aware of the buffer handling mechanisms used by the lower layers. Therefore, the application more efficiently reuses buffers. Also, since the application process can use the same memory as the networking code the application can read and write directly to the internal buffers, thus saving the expense of performing a copy. As in many other TCP/IP implementations, the layered protocol design was used as a guide for the LWIP design and implementation. Each protocol is implemented as its own module, with a few functions acting as entry points into each protocol. Even though the protocols are implemented separately and as said
before, some layer violations are made in order to improve performance both in terms of processing speed and memory usage. Hence many benefits can be obtained using RTL-LWIP. Besides providing IPv6 and TCP protocols, RTL-LWIP is more suitable for embedded systems, not only for its code but for its memory usage improvements.

III. RESULTS AND DISCUSSIONS

Fig 5 & 6 shows the few Simulation and execution results of ARM web server based DACS system using RTLinux.

Fig.7 is the simple web page designed using HTML language. It is requested by the client to server. Then the internet processes these request and server response for client request with web page. Now the Client can know the status of industry machineries and can control the machines via its own browser from remote location. It is showed in Fig.8&9.

Hence, results show that the client can access the whole industry from any remote place via its own local browser. In industry the single ARM9 board acts as data acquisition and control system and as web server, so the system is compact with less complexity. This system replaces the traditional system for remote access and control by embedded web server with Real Time Linux operating system. And this system is adaptable with kernel level debugging. It can be done by GDB, DDD Linux debuggers. These debuggers satisfy hard real time requirements by the use of RTLinux core.
IV. MERIT OF THE SYSTEM

A. Existing System

The use of single chip Data acquisition system (DAS) method in Instrumentation and process control application is not only limited in processing capacity and also the problem of poor real time and reliability. General web server requires more resources and huge amount of memories. This system can only measure the remote signals and it cannot be used to control the process.

B. Proposed System

Limited processing capacity and the problem of poor real time and reliability of DAS system has been overcome by the substitution of embedded ARM processor for single chip method to realize interactive data acquisition and control (IDACS). This IDACS system can able to measure the remote signals and can control the remote devices through reliable protocols and communication network. This system uses RTLinux Multi-tasking operating system to measure and control the whole process. And the embedded web server mode requires less resource usage, high reliability, security, controllability and portability.

V. CONCLUSIONS

With the rapid development of the field of industrial process control and the wide range of applications of network, intelligence, digital distributed control System, it is necessary to make a higher demand of the data accuracy and reliability of the control system. This embedded ARM system can adapt to the strict requirements of the data acquisition and control system such as the function, reliability, cost, size, power consumption, and remote access and so on. This system operated by DACS mode to acquire the signals and control the devices remotely. Embedded web server mode is used to share the data with clients in online. Both modes are efficiently carried out by real time multi tasking operating system (RTLinux). This system can be widely applied to electric power, petroleum, chemical, metallurgy, steel, transportation, Electronic & Electrical industries, Automobiles and so on.

REFERENCES


[16] www.opensource.org

[17] www.embeddedarm.com

[18] www.fslabs.com

Mr.M.Manivannan is doing his M.E (Embedded Systems Technologies) in Anna University of Technology Coimbatore, Coimbatore. He has completed his B.E (ECE) degree from Government College of Engineering, Tirunelveli during 2008. In 2005 he has completed his Diploma in Electronics and Communication Engineering from K.L.N.M Polytechnic College, Madurai. He published research papers in Five National conferences and Three International Conferences. And he published a Research Paper in International Journal. His research areas include Embedded operating Systems, Kernel, RTOS and Embedded Control.

Mr.N.Kumaresan is doing his Ph.D (Embedded Systems) in Anna University of Technology Coimbatore, Coimbatore. He completed his M.E degree (Embedded Systems Technology) from Sri Ram Engineering College in 2006, B.E (ECE) from Government College of Engineering, Salem in 1996 and currently he is working as a Assistant Professor in Dept of ECE Anna University of Technology Coimbatore. He published three papers in National Conference and two papers in International Conference. And he published a Research Paper in International Journal. His research areas include Real time Embedded systems, Operating systems, Networking, Wireless Sensors and Mobile Communication. He has eleven years of excellent teaching experience.