

임베디드 시스템을 이용한 양어장 모니터링 시스템의 설계 및 구현

Design and Implementation of Fish Farm Monitoring System using Embedded System

이준택
금오공과대학교 컴퓨터공학과

Joon-Taek Lee
Dept. of Computer Eng., Kumoh National Institute of Technology

조수현
금오공과대학교 컴퓨터공학과

Soo-Hyun Cho
Dept. of Computer Eng., Kumoh National Institute of Technology

신동현
금오공과대학교 컴퓨터공학과

Dong-Hem Shin
Dept. of Computer Eng., Kumoh National Institute of Technology

김영학
금오공과대학교 컴퓨터공학부

Young-Hak Kim
Dept. of Computer Eng., Kumoh National Institute of Technology

중심어 : 임베디드 시스템, Linux, 웹카메라, Video4Linux, RS-232C, 감시 시스템

요약

현재까지는 해양 어류의 생태와 양식과정 등을 연구하기 위해서 연구자가 직접 육안으로 실험환경을 관찰하였으며, 양어장의 영상, 온도와 같은 상태 등의 정보를 실험실 내에서만 모니터링 할 수 있었다. 기존의 이러한 방식은 과학적이고 체계적인 방법으로 어류의 생태를 관찰할 수 없고, 많은 연구 인력이 필요하다.

따라서 본 논문에서는 리눅스를 운영체제로 사용한 임베디드 시스템을 이용하여 원격지의 실험환경 상태를 언제, 어디서든 인터넷을 통하여 실시간으로 전송 받아, 그 상태를 종합적으로 검사 및 분석할 수 있는 양어장 모니터링 시스템을 설계 및 구현한다.

Abstract

Up to now, researchers have observed experiment environment to study about raising procedure and an ecology of the ocean fishes with naked eye. Then they can monitor status information in only laboratory such as temperatures, image of the fish farm, etc. Such the existing way as this can't observe an ecology of the ocean fishes with scientific and systematic method and moreover it needs many of researchers.

So in this paper, we design and implement a fish farm monitoring system which receives status of remote experiment environment through the internet whenever you want it and wherever you want it, and monitors and analyzes its status using embedded system of which the operating system is linux.

I. 서론

인터넷은 전 세계 어느 곳, 어떠한 컴퓨터 시스템에서도 접속할 수 있는 장점을 가진다. 또한 원격지에서 자신의 컴퓨터 혹은 서비스 제공업자의 서버에 접속하여 필요한 정

보를 얻을 수 있다. 이런 이유로 인터넷을 기반으로 하는 다양한 응용 서비스들이 등장하고 있다[1].

일반적인 해양어류 등의 생태를 연구하기 위해서 연구자는 실험실에서 육안으로 관찰하기 때문에 과학적인 방법으로 어류의 생태를 관찰할 수 없고 많은 연구 인력을 필요로 한다. 이런 문제점을 해결하기 위해 양어장의 관리를 자동화하여 인력을 줄 일 수 있는 양어장 자동제어장치[2],[3]

※ 본 연구는 경상북도, 중소기업청에서 지원하는 10차(2002)년도 금오공과대학교 산학연 공동기술개발 컨소시엄과제로 수행되었음

가 개발되었다. 하지만 원격지의 실험실 상태를 그래프로만 나타내고 있으며 실제 어류의 움직임과 같은 영상 정보를 확인 할 수는 없다. 또한 인터넷을 이용한 모니터링과 관리 자동화를 이룬 환경 모니터링 및 감시 시스템[4]이 개발되었으나, 전용 프로그램이 필요하여 프로그램이 설치되지 않은 곳에서는 사용할 수 없는 문제점이 있다.

따라서, 본 논문에서는 연구실뿐만 아니라 어디에서든 원격지의 상태를 모니터링 할 수 있는, 인터넷 기반의 원격 모니터링 시스템을 구현하였다. 핵심기능은 다양한 각도에서 캡처된 영상과 온도 센서를 통해 측정된 수온을 클라이언트로 전송하는 것이다. 기존에 구현된 시스템들과는 달리 웹을 통해서 접근이 가능하기 때문에 별도의 클라이언트 프로그램이 필요 없으며, 인터넷에 연결된 PC라면 언제 어디에서든 시스템이 구축된 곳의 상태를 실시간으로 파악할 수 있는 장점이 있다.

이러한 시스템의 활용 분야는 아동 관찰, 교통량 관찰, RS-232C[5]를 통해 연결된 원격 기기의 제어 등에 사용될 수 있다[6]. 본 논문의 구성은 다음과 같다. 2장에서는 임베디드 시스템의 이해에 대해 알아보고, 3장은 어류 양식장 모니터링 시스템의 개요 및 구조를 살펴보고, 4장에서는 구현 결과 및 시스템의 장점들을 살펴본다. 끝으로 5장에서는 결론을 설명한다.

II. 임베디드 시스템

이 장에서는 본 논문에서 구현을 위해 사용된 임베디드 시스템과 임베디드 리눅스에 대해서 살펴본다.

1. 임베디드 시스템의 개요

임베디드 시스템은 컴퓨터 하드웨어와 소프트웨어를 모두 포함하는 개념으로서 특수 목적을 위해 만들어진 시스템이다. 일상생활에서도 임베디드 시스템을 자주 접할 수 있는데, 예를 들면 가정용 전자제품, 자동차의 제어 시스템, 공장 자동화 시스템, 네트워크 라우터 등과 같은 다양한 분야에 임베디드 시스템이 적용되고 있다. 임베디드 시스템은 어떤 목적에 사용될지 모르는 다양한 주변 장치를 갖추고 있는 범용 컴퓨터 시스템과는 달리 특정한 목적을 수행하기 위해 설계되고, 그 특정한 목적을 수행하기 위해 필요로 하는 하드웨어와 소프트웨어로 구성된다.

2. 과거의 임베디드 시스템

임베디드 시스템은 최근에 생겨난 전혀 새로운 개념이 아니며, 오래전부터 많은 연구와 개발이 수행되었다. 임베디드 시스템의 활성화는 전자계산기만을 위한 집적회로를 사용하는 것을 대신하여 여러 가지 일반적인 요구 사항까지 처리할 수 있는 마이크로프로세서를 개발하면서 시작되었으며, 임베디드 시스템은 소프트웨어의 변경만으로도 다양한 시스템을 개발할 수 있는 환경을 제공하였다. 과거의 임베디드 시스템은 일반적으로 보드(Board)에 시스템이 구현되는 형태였으며, 단순한 반복적인 작업만을 수행하였다.

3. 임베디드 시스템의 변화

과거에는 임베디드 시스템이 한 두가지의 목적만을 위해서 설계되었기 때문에 대부분 간단한 형태로 제작되었으며, 기기의 수명이 다할 때까지 같은 동작을 하는 소프트웨어가 ROM에 내장된 형태가 일반적이었다. 그러나 현재에는 수행되는 기능이 다양해지고 복잡해짐에 따라 소프트웨어의 장점이라 할 수 있는 유연성과 기능 확장의 용이성이 극대화 되었으며 복잡해진 기능을 제어하기 위한 독자적인 운영체제도 필요하게 되었다. 또한 저장 매체의 기술 발전도 빠른 속도로 이루어져 상대적으로 비싸졌던 소프트웨어를 담아내고, 운영체제를 구동할 수 있을 만큼의 충분한 크기의 메모리가 개발 되었다. 그리고 프로세서, 메모리, MPEG(Moving Picture Experts Group) 인코더/디코더와 같은 특화된 처리기, 입출력 장치 등 다양한 종류의 하드웨어를 하나의 칩 안에 구현하여 제품화 될 정도의 급속한 발전을 하였다.

4. 임베디드 시스템의 미래

이전 임베디드 시스템[7]은 지금에 비해 상대적으로 단순하였으며, 일반 대중들에게 팔 물건들이 전통적인 가전제품에 국한되었다. 그러나 세계적으로 정보통신기술 등이 발전함에 따라 새로운 수요가 창출되었다. 따라서 고성능, 다양한 기능을 원하게 되었으며 새로운 가전제품의 개념과 인터넷 기기, 모바일 기기 등과 같은 다양한 형태의 임베디드 시스템이 좀 더 복잡하고 특화된 형태로서 출현할 것이며 그 수요 또한 급증함이 예상된다.

5. 임베디드 리눅스

오픈 소스 정책을 표방한 리눅스가 여러 플랫폼으로 확산되면서, 부수적으로 임베디드 시스템에 리눅스를 이식할

수 있게 되었다. 또한 고성능, 다기능을 제공하는 범용 CPU를 채택한 임베디드 시스템이 늘면서 성능을 최대한으로 향상시킬 수 있는 리눅스가 각광 받게 되었다.

III. 시스템의 설계

본 논문에서 구현한 양어장 모니터링 시스템의 개요와 구성요소 및 작동 시나리오를 살펴본다.

1. 시스템의 개요

본 시스템의 각각의 관계를 나타낸 논리적 구성은 그림 1과 같으며 전체적인 물리적 구성은 그림 2와 같다. 본 논문에서 구현한 어류 양식장 모니터링 시스템은 서버로부터 전송된 이미지와 온도를 확인할 수 있는 JAVA 애플릿 형태로 만든 클라이언트, 그리고 개별적인 작업을 하는 세 개의 서버 프로세스로 이루어져 있다. 서버 프로세스는 각각 영상을 캡처하고 온도를 온도센서로부터 측정하는 모듈, 클라이언트와의 사이에 제어 채널을 개설하여 제어신호를 주고받는 모듈, 클라이언트로 영상과 온도 데이터를 전송하는 모듈로 구성된다.

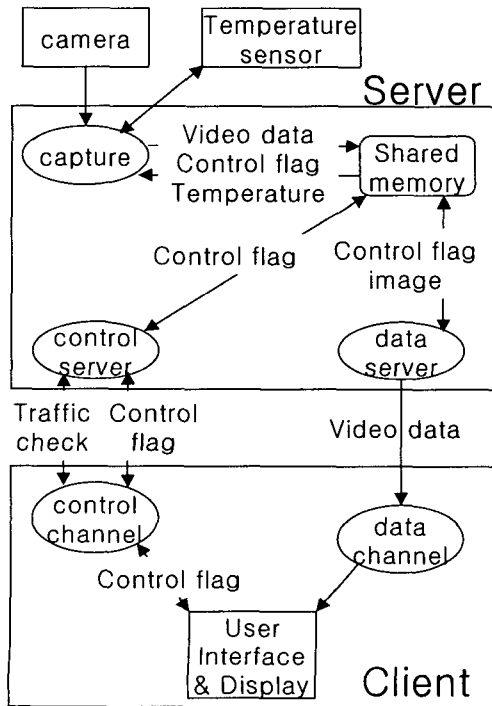


그림 1. 시스템 논리적 구성도

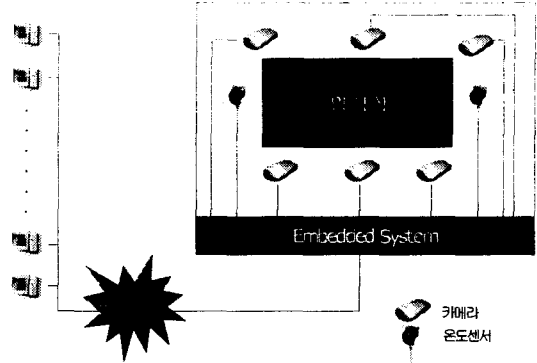


그림 2. 시스템의 물리적 구성

본 논문에서 구현한 시스템은 실시간 전송을 목적으로 하고 있으므로 네트워크의 트래픽과 연결된 클라이언트의 수에 따라 이미지의 압축 비율, 단위 시간당 데이터 전송량을 조절할 필요가 있다. 이를 위해 본 시스템에서는 일정 시간마다 네트워크 속도를 측정하여, 이미지의 압축 비율과 시간당 전송량을 조절하여 서버의 작업량을 줄임으로서 클라이언트에게 안정적인 전송을 보장할 수 있다.

2. 시스템 구성

본 시스템은 그림 2에서의 같이 서버 측과 클라이언트 측으로 구성된다. 서버 역할을 하는 리눅스를 포팅한 임베디드 시스템[8]에 범용 USB[9] 카메라 4대를 USB허브를 통하여 연결하고, 수온 및 상온을 측정 할 수 있는 온도센서를 RS-232C를 통하여 연결하였다. 웹브라우저에 삽입된 애플릿 코드를 통해 서버에 접속하여 영상과 온도 정보를 전송 받고, 클라이언트에서 보내는 제어 신호를 서버로 송신하며 이에 대한 응답을 수신한다.

2.1. 서버 부분

서버 프로그램은 크게 다음의 세 가지 모듈로 나누어진다.

2.1.1. 데이터 수집 모듈

온도센서로부터 RS-232C를 통해 온도를 측정하고 USB 카메라에서 영상을 캡처하여 JPEG[10] 형태나 MPEG[11]형태로 압축하며 시스템의 공유 메모리(Shared Memory) 영역으로 복사한다.

2.1.2. 제어 신호 송수신 모듈

트래픽 체크, 로그인, 모니터링 할 카메라 선택, 영상 검색, 특정 온도가 되면 다이얼로그 박스를 띄워서 알람기능을 하는 알람 온도 설정, 온도 측정 ON/OFF, 동영상 저장

시작 신호, 동영상 저장 정지 신호 등과 같은 클라이언트에서 보내는 제어 신호를 받아서 전체 시스템의 흐름을 총괄한다.

2.1.3. 데이터 송신 모듈

카메라에서 캡처해서 압축한 정지영상인 JPEG 포맷의 데이터와 측정된 온도 데이터를 혼합해서 연결된 클라이언트로 다른 제어 신호가 들어 올 때까지 지속적으로 송신한다.

2.2. 클라이언트 부분

사용자가 이용하게 되는 클라이언트 부분은 PC상에서 별도의 전용 프로그램을 설치할 필요가 없다. 이는 JAVA를 사용하여 구현한 애플릿이 웹페이지에 삽입된 형태로서 본 시스템을 처음 접하는 사용자도 손쉽게 조작 할 수 있도록 구현하였다. 클라이언트는 제어 신호를 서버로 보내어 원하는 영상과 온도 데이터를 전송 받을 수 있다. 필요한 경우 동영상을 서버 쪽에 파일로 저장하여, 후에 검색을 통해서 다시 영상을 확인 할 수 있다.

2.3. 작동 시나리오

그림 3은 본 시스템의 동작 순서를 나타내며, 구체적인 내용은 다음과 같다.

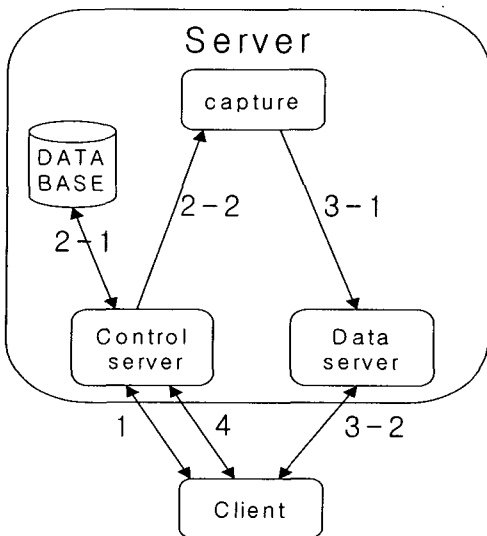


그림 3. 작동 시나리오

- 1) 클라이언트는 제어 채널을 통해 로그인 정보를 보낸다.
- 2) 제어신호 송수신 모듈은 로그인 정보를 데이터베이스의 정보와 비교하여 일치한다면, 데이터 수집 모듈을 동작시킨다.
- 3) 클라이언트는 제어신호 송수신 모듈로부터 인증 승인 메시지를 받게 되면 데이터 송신 모듈로부터 영상과 온도 데이터를 수신한다.
- 4) 클라이언트는 영상과 온도를 감시하면서 제어 채널을 통해 전체 시스템을 제어한다.

IV. 구현

본 장에서는 시스템 구현에 사용된 실험환경에 대해 알아본다. 그리고 본 시스템의 구현 모델이 되는 클라이언트/서버간 통신을 위해 사용된 프로토콜에 대한 정의와 작동 원리를 살펴본다. 실제 구현부분은 클라이언트와 서버부분으로 나눠 설명한다.

1. 구현 및 테스트 환경

표 1. 구현 및 테스트 환경

하드웨어
- LETOK-850 보드
- IBM 호환 PC(Pentium 667, 128 Main memory)
- 범용 USB 카메라 4대
- 4포트 USB 허브
- 수온 측정용 온도센서
소프트웨어
- gcc 3.2
- ffmpeg(free JPEG/MPEG encoding library) [12]
- mysql 3.23
- J2SDK-1.3.1 [13]
- JMF(JAVA Media FrameWork) 2.0
- boa web server [14]
- Hart Hat Linux Cross Development Kit 1.2 [15]
- kernel 2.4.9

표 1은 본 시스템을 구현함에 있어 사용된 하드웨어와

소프트웨어의 목록을 나타낸다. 실제 실험 환경의 구성은 그림 4와 같으며 범용 USB 카메라 4대를 USB 허브를 통하여 오른쪽 아래의 LETOK-850 보드 USB 포트에 연결하였다. 그리고 크로스 개발환경 구축을 위해 시리얼 포트를 이용하여 데스크탑 PC와 연결하였다.

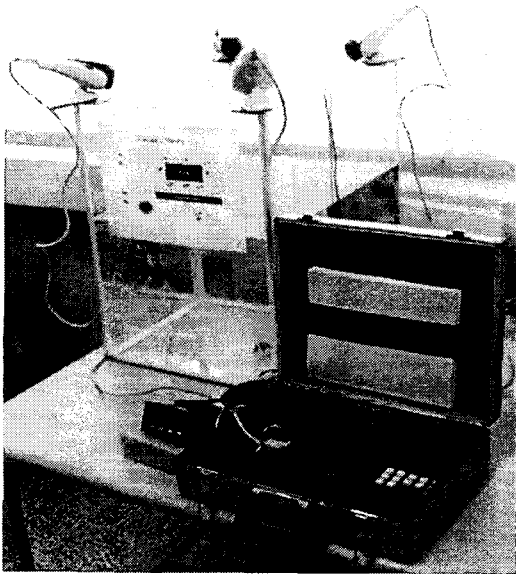


그림 4. 시스템의 실험 환경

표 2는 본 논문에서 구현한 시스템에서 사용한 LETOK-850 보드의 하드웨어 사양을 표로 나타낸 것으로서 RISC 칩인 PowerPC 850(50Mhz)를 CPU로 사용하고 있으며 16MB의 SDRAM과 4MB의 플래시(Flash) 메모리를 가지고 있다. 또한 외부 인터페이스로 RS-232C와 USB를 가지고 있고 두개의 네트워크 인터페이스를 가지고 있는 임베디드 시스템 보드이다. 구현에 있어서 LETOK-850보드의 플래시 메모리에 본 시스템에 적합하도록 설정한 리눅스 커널의 압축된 형태가 저장되며, 전원이 켜진 후에는 SDRAM으로 리눅스가 로드 되어 시스템 전체를 제어하게 됨으로서 일반 PC의 환경과 동일하게 동작한다. 그림 5는 LETOK-850 보드의 CPU 블록 다이어그램을 나타낸다.

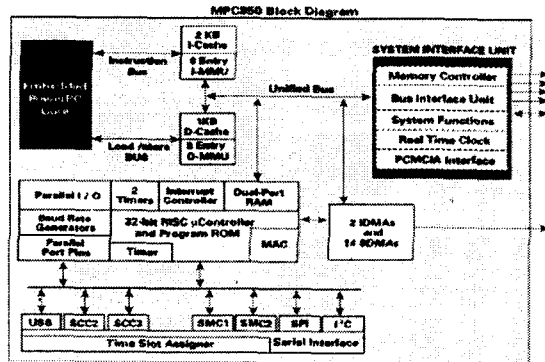


그림 5. LETOK-850 Block Diagram

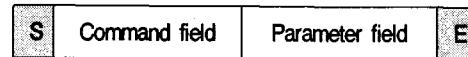
표 2. LETOK-850 하드웨어 사양

항 목	사 양
CPU	Motorola PowerPC 850DE RISC Processor 50MHz
Memory	SDRAM Hynix SDRAM (16MB - 8x2)
	Flash Intel TE28F320 Flash Memory (4MB)
Interface	LCD HG12605 Graphic LCD(128x64)
	Serial RS-232C and RJ11 Serial port (내장형 UART)
	USB USB Slave
	Ethernet Dual 10Base-T on Board Ethernet
	BDM S/W Download, Fusing
	Dip Switch 8EA
	LED 8EA
	확장 I/O port 8bit address 16bit data, 2개의 Interrupt 사용 가능

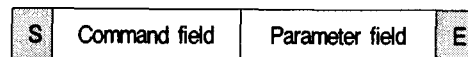
2. 프로토콜 정의

본 논문에서 구현된 시스템은 클라이언트/서버 모델로서 서로간의 통신이 필요하다. 따라서 통신을 위한 문자열 (String) 기반의 간단한 프로토콜을 정의한다.

1. 명령 메시지 포맷



2. 리턴 메시지 포맷



리턴 메시지는 3개의 필드로 구성되며 각 필드들은 "(" (파 이프)로 구분한다.

Acknowledge Message.

- 0 : OK
- 1 : Failed
- 2 : Undefined

명령 및 리턴 메시지의 공통적인 메시지 필드로서, 메시지의 시작에 "S", 메시지의 마지막에 "E"를 추가하여 메시지의 처음과 끝을 구분한다. 세부적인 명령 메시지들은 4개의 기본 필드로 구성되며 각 필드들은 "|" (파이프)로 구분한다. Parameter field 는 다시 여러 개의 Parameter 들로 구성될 수 있으며 각 Parameter 들 또한 "|" 로 구분하고 모든 메시지들의 타입은 문자열로 한다.

표 3은 정의된 메시지 규칙과 그 예로서 Command field 에 따른 각각의 기능과 세부적인 Parameter field의 내용에 대해서 설명한다.

표 3. 클라이언트/서버 메시지 규칙 및 사용 예

수행 기능	Command field	Parameter field
Login	1	ID, Password
Ex) "S 1 ID Password E"		
영상 관찰	2	camera number
Ex) "S 2 camera number E"		
영상 저장 시작	3	.
Ex) "S 3 E"		
영상 저장 종료	4	.
Ex) "S 4 E"		
영상 검색	5	검색어
Ex) "S 5 2003/01/01 2003/01/02 검색어 E"		
알람 설정	6	설정 온도
Ex) "S 6 설정 온도 E"		
알람 해지	7	OFF
Ex) "S 7 E"		

3. 시리얼 통신 프로토콜

RS-232C를 통해 연결된 온도센서로부터 온도를 측정하기 위해서는 온도센서의 펌웨어에 정의된 메시지 형식에 맞추어 명령을 보내야한다. 펌웨어는 명령에 해당하는 결과를 같은 형식에 맞추어 임베디드 시스템으로 전송한다. 다

음은 송수신되는 메시지의 기본적인 구조를 나타낸다.

Stx	Record	Etz
-----	--------	-----

표 4는 각각의 필드에 대한 설명으로서 실제 메시지 내에 포함되어야 할 값들이다.

표 4. 온도센서 송수신 메시지 규칙

<Stx>	메시지의 시작(아스키 코드 02h)
<Record>	명령어와 파라미터의 조합
<Etz>	메시지의 끝(아스키 코드 03h)

메시지의 Record 필드는 다시 세 개의 필드로 나누어지는데 첫 번째가 주 명령, 두 번째가 주 명령에 따른 부 명령, 세 번째는 메시지의 종류에 따라 메시지에 포함 될 수도, 없을 수도 있는 명령에 따른 세부적인 값이다.

실제 온도센서의 펌웨어가 지원하는 동작은 여러 가지가 있지만 본 논문에서는 실제 온도 측정을 위해 사용한 명령어에 대해서만 언급하였다. 표 5에서 M은 온도 측정을 지시하는 주 명령을 나타내며, 1과 2는 채널을 선택하는 부 명령을 나타낸다. 그리고 Values는 메시지를 보낸 후 얻게 되는 값이며, Replies는 펌웨어로부터의 응답 메시지에 포함되는 송신된 메시지에 대한 확인 값을 의미한다.

표 5. 온도측정 세부 명령

COMMAND M		
Sub-command	Values	Replies
1	채널 1을 측정할 값	ack/nak
2	채널 2를 측정할 값	ack/nak

4. 클라이언트 구현

그림 6은 윈도우즈 운영체제를 사용하는 PC에서 접속한 화면이다. 그림 6에서 보이는 로그인 화면에 ID와 패스워드를 입력하고 로그인 버튼을 누르면 서버로 접속을 시도한다. 구현된 로그인 웹페이지는 일반적인 로그인 기능을 제공하는 PHP나 JSP와 같은 스크립트 언어를 사용한 동적 웹페이지가 아니라 JAVA를 이용해 구현한 애플릿이 삽입된 형태이다.

사용자 인증을 받기 위해서 서버와의 소켓을 개설하고 ID와 패스워드를 서버 쪽으로 보내어 인증을 받게 되면 실제

사용자가 이용하게 될 사용자 인터페이스 윈도우를 화면상에 출력한다. 그런 후, 데이터 서버로 접속을 시도하여 데이터를 수신하게 된다.

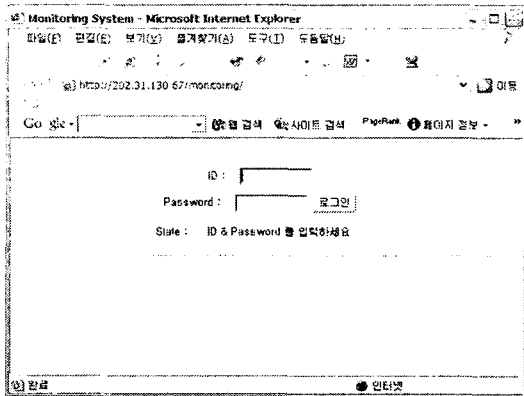


그림 6. 로그인 화면

그림 7은 인증 과정 후에 볼 수 있는 사용자 인터페이스 윈도우이다. 좌측에 보이는 것과 같이 320x240 픽셀 크기의 JPEG 형태의 정지화상을 전송 받아 연속적으로 보여줌으로서 동영상 효과를 줄 수 있다. 이는 본 시스템의 목적인 모니터링을 위해서 정지 영상의 전송만으로도 어느 정도의 효율성을 얻을 수 있다. 오른쪽 상단은 수신한 온도를 표시하는 부분이고, 그 아래에 있는 체크박스는 알람기능을 ON/OFF 하는 기능이다. 저장시작, 저장종료 버튼은 사용자가 MPEG 형식으로 현재 모니터링하고 있는 영상을 저장/저장종료 할 때 사용한다.

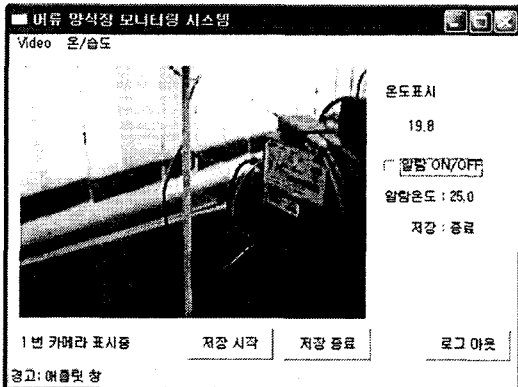


그림 7. 사용자 인터페이스

그림 8은 사용자가 알람 온도를 설정하는 팝업 윈도우이다. 사용자가 온도를 설정 해 놓으면 설정된 온도를 데이터 베이스에 저장하고 사용자 인증 시에 클라이언트 프로그램에서는 저장된 값을 불러와 비교한다. 만약, 온도센서를 통해 측정된 온도가 설정된 온도 이상의 값이 되면 사용자에게 알려준다.

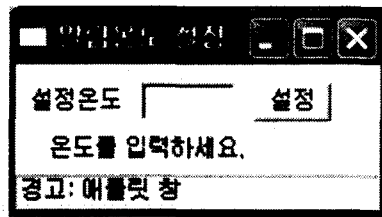


그림 8. 알람온도 설정

그림 9는 사용자가 원하는 시점의 카메라 영상을 저장하여 완료할 때 저장될 파일에 대한 설명을 입력하는 윈도우이다. 이 설명은 나중에 사용자가 저장된 파일을 검색해 볼 수 있는 키워드가 된다.

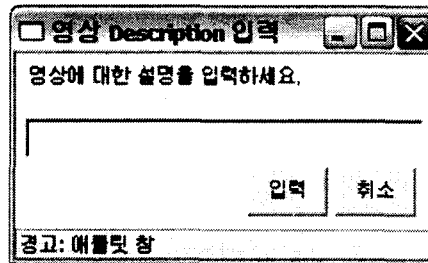


그림 9. 영상 저장

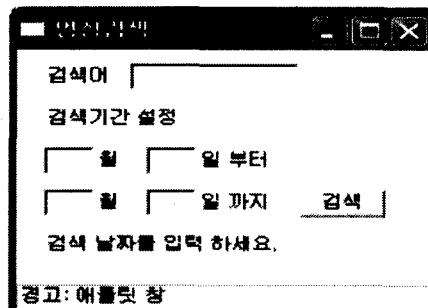


그림 10. 영상 검색

그림 10은 저장된 파일을 검색하는 윈도우이다. 저장된 그림의 설명부분과 날짜 범위를 입력하여 지난 영상들을 볼 수 있게끔 구현하였다. 그림 11은 입력된 조건에 일치하는 파일이름의 리스트들을 보여주며, 사용자가 해당 파일을 선택하면 그림 11의 오른쪽과 같이 파일의 내용을 볼 수 있다.

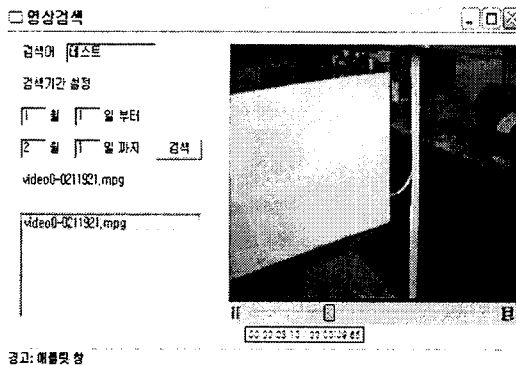


그림 11. 영상 검색 결과

5. 서버 구현

서버 프로그램의 구동 환경을 위해 Hard Hat Linux CDK를 사용하여 LETOK-850보드에 적합한 크로스 플랫폼 개발 환경을 구축하였다.

서버 프로그램은 운영을 위한 별도의 UI가 존재하지 않으며 세 개의 데몬 프로세스가 각자 다른 일을 하면서 공유 메모리를 통하여 서로간의 데이터를 주고받는다. 서버의 세 개 프로세스는 최초 대기 상태에서 클라이언트가 접속을 하면 다음 3단계의 순차적인 작업을 수행한다.

첫 번째, 제어 프로세스는 사용자 인증 요청을 받고 ID와 패스워드를 데이터베이스의 저장된 정보와 비교한다. 일치하면 접속 승인 메시지를 보내고 공유메모리를 통하여 온도 측정 및 영상 캡처 프로세스에게 JPEG 형태의 정지영상과 온도 데이터를 하드웨어로부터 얻을 수 있는 제어 신호를 보낸다. 두 번째, 온도 측정 및 영상 캡처 프로세스는 JPEG 형태의 정지영상과 온도 데이터를 하드웨어로부터 얻는다. 마지막으로, 데이터 전송 프로세스는 인증을 거친 클라이언트가 접속하면 공유 메모리 영역의 데이터를 읽어서 클라이언트로 전송한다.

초기화 작업 이후, 접속하는 클라이언트에 대한 서버 프로세스들의 동작은 제어 프로세스가 대기 상태에 있던 다른 프로세스들을 공유 메모리의 데이터 변경을 통해 활성

화시키는 것을 제외한, 나머지는 위의 작업들과 동일하다. 다음은 사용자의 인증을 위한 기본적인 정보와 저장되는 파일의 정보 유지를 위해 사용된 데이터베이스 테이블 구조이다.

표 6에서 uid는 사용자의 ID, passwd는 비밀번호, alram은 사용자가 설정한 알람신호를 보내게 되는 설정 온도의 값이다. 표 7에서 uid는 파일을 저장한 사용자의 ID이고, description은 저장된 파일에 대한 설명 부분이며 나중에 검색을 위해 사용되는 필드이다. 그리고 filename은 저장된 파일의 이름이고, 끝으로 date에는 파일이 저장된 날짜를 나타낸다.

표 6. 사용자 정보 테이블

Field	Type	Null	Key	Default	Extra
id	int(11)		PK	NULL	auto_increment
uid	varchar(10)				
passwd	varchar(10)				
alram	varchar(10)				

표 7. 저장된 동영상 파일정보 테이블

Field	Type	NULL	Key	Default	Extra
id	int(11)		PK	NULL	auto_increment
uid	varchar(10)				
description	tinytext				
filename	varchar(10)				
date	timestamp(14)	YES		NULL	

V. 결론

본 논문에서는 리눅스를 운영체제로 사용하는 임베디드 시스템 기반의 어류 양식장 모니터링 시스템을 설계하고 구현하였다. 웹이라는 가장 접근하기 쉬운 환경을 사용하여 공간의 제약성을 극복할 수 있었으며 네트워크 상태와 서버와 연결된 클라이언트의 수를 고려하여 서버의 작업량을 조절함으로써 안정적인 전송을 보장할 수 있었다. 그리고, 실시간 모니터링뿐만 아니라 영상의 저장이 가능해 자료를 남김으로서 후에 사용자가 검색을 통해 저장되어 있는 영

상을 다시 검토 해 볼 수 있는 기회를 제공하였다.

범용 CPU를 채택한 임베디드 시스템을 사용하여 기존의 자동화된 양어장 관리시스템의 이점은 살리고, 시스템의 확장성을 증가시켰으며, 전체 시스템 구성에 드는 비용을 줄여 기존의 소규모 양어장 시스템을 대체하려는 시도에서 본 논문은 가치가 있다고 하겠다.

참 고 문 헌

- [1] 이명옥, 이은미, "웹캠:새로운 인덱스검색 알고리즘을 이용한 웹기반 원격 녹화 보안 시스템", 정보처리학회 논문지 C, Vol.9-C No. 1, pp. 9-16, 2002.
- [2] <http://aquasysauto.com/aquasysauto.com/aquasysauto.com/main.htm>.
- [3] <http://www.cbtech.co.kr/2-5.htm#2>.
- [4] http://cimec.pknu.ac.kr/detail5_8_4.htm.
- [5] Serial Programming Guide for POSIX Operating Systems, <http://www.easysw.com/~mike/serial/serial.pdf>.
- [6] 이정배, 김인홍, "원격 영상감시 및 제어 자동화", 한국정보처리학회지, Vol. 4, No. 4, pp. 100-110, 1997.
- [7] <http://www.embedded.org/>.
- [8] <http://www.linuxdevices.com>.
- [9] Universal Serial Bus Revision 2.0 specification, http://www.usb.org/developers/docs/usb_20.zip.
- [10] <http://www.jpeg.org>.
- [11] <http://mpeg.telecomitalia.com>.
- [12] <http://ffmpeg.sourceforge.net>.
- [13] <http://java.sun.com>.
- [14] <http://www.boa.org>.
- [15] <http://www.nvista.com>.

이 준 택(Joon-Taek Lee)

준회원



2002년 2월 : 금오공과대학교 컴퓨터공학과(공학사)
2002년 2월 ~ 현재 : 금오공과대학교 컴퓨터공학과(석사과정)
<관심분야> : 운영체제, 임베디드 시스템 등

조 수 현(Soo-Hyun Cho)

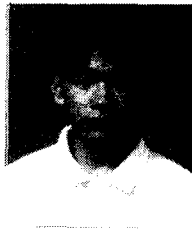
정회원



2000년 2월 : 금오공과대학교 컴퓨터공학과(공학사)
2002년 2월 : 금오공과대학교 컴퓨터공학과(공학석사)
2002년 2월 ~ 현재 : 금오공과대학교 컴퓨터공학과(박사과정)
<관심분야> : 병렬/분산처리, 이동 에이전트, 병렬 프로그래밍, Cluster Computing

신 동 헌(Dong-Hern Shin)

준회원



2001년 2월 : 금오공과대학교 컴퓨터공학과(공학사)
2002년 2월 ~ 현재 : 금오공과대학교 컴퓨터공학과(석사과정)
<관심분야> : 운영체제, 임베디드 시스템 등

김 영 학(Young-Hak Kim)

정회원



1984년 2월 : 금오공과대학교 전자공학과(공학사)
1989년 2월 : 서강대학교 전자계산학과(공학석사)
1997년 2월 : 서강대학교 전자계산학과(공학박사)

1989년 ~ 1997년 : 해군사관학교 전산학과 교수
1998년 ~ 1999년 : 여수대학교 멀티미디어학부 교수
1999년 ~ 현재 : 금오공과대학교 컴퓨터공학부 조교수
<관심분야> : 병렬 알고리즘, 분산 및 병렬처리 등