



채팅과 코딩이 병합된 마이크로컨트롤러 협력학습시스템 개발

한새론*, 최관순**¹, 이순흠***, 김동식**², 안달**³, 전흥구**⁴

Development of Microcontroller Collaborative Learning System with Merged Chatting and Coding

Saeron Han*, Kwansun Choi**¹, Sunheum Lee***, Dongsik Kim**², Dal Ahn**³, and Heunggu Jeon**⁴

이 논문은 2017학년도 순천향대학교 교수 연구년제에 의하여 연구하였음.

요 약

인터넷 정보사회가 도래함에 따라 협력을 통해 문제를 해결하려는 흐름은 교육분야에서 활발히 진행되고 있다. 이러한 상황에 맞추어 본 연구에서는 협력학습 시스템을 개발하게 되었다. 제안한 마이크로컨트롤러 협력학습시스템(MCLS)은 유무선망이 연결된 네트워크 환경 하에서 PC 또는 모바일 기기를 이용하여 언제 어디서나 자신의 진도에 맞추어 원하는 개별학습을 할 수 있을 뿐만 아니라, 채팅을 통하여 서로 의견을 나누고 협력하여 마이크로컨트롤러에 관련된 프로그래밍 과정과 실행결과를 함께 살펴볼 수 있도록 하였으며, 본 논문에서 제안된 마이크로컨트롤러 협력학습시스템의 유효성을 입증하기 위하여 PC 또는 모바일 기기를 이용하여 원격학습자들 상호간의 채팅과 코딩을 통해 마이크로컨트롤러와 관련된 문제를 해결하는 과정을 예로서 제시하였다.

Abstract

As the Internet information society arrives, the trend toward solving the problems through cooperation is actively underway in the field of education. In this study, we developed a collaborative learning system. The proposed Microcontroller Collaborative Learning System(MCLS) is designed to make the desired individual learning based on their own pace at anytime and anywhere by using personal computers or mobile devices with wired or wireless network environment. Furthermore, the distant learners can examine coding process and execution result simultaneously by sharing and collaborating their opinion through the general chat program. The procedure of solving problem of microcontroller through chatting and coding between distant learners will be illustrated as examples in order to prove the validity of the proposed MCLS.

Keywords

collaborative learning system, microcontroller, virtual/remote laboratory, SNS

* 파루씨앤씨

- ORCID: <http://orcid.org/0000-0003-0104-8711>

** 순천향대학교 전기공학과

- ORCID¹: <http://orcid.org/0000-0001-6802-5819>

- ORCID²: <http://orcid.org/0000-0001-5112-7392>

- ORCID³: <http://orcid.org/0000-0003-0556-0771>

- ORCID⁴: <http://orcid.org/0000-0003-2335-8155>

*** 순천향대학교 정보통신학과

- ORCID: <http://orcid.org/0000-0002-1626-4276>

• Received: Nov. 27, 2018, Revised: Mar. 06, 2019, Accepted: Mar. 09, 2019

• Corresponding Author: Kwansun Choi

Dept. of Electrical engineering/college of Engineering/Soonchunhyang University, Soonchunhyang-ro 22, Shinchang-myeon, Asan, Chungnam, 31538, S., Korea, Tel.: +82-41-530-1329, Email: cks1329@sch.ac.kr

I. 서 론

인터넷과 모바일 환경의 급속한 발전으로 인하여 현대인들의 실생활은 오프라인에서 시간과 공간의 제약이 없는 온라인 중심으로 변화되었다. 그 결과로서 인터넷 정보사회가 도래함에 따라 개인이 중심이 되어 문제를 해결하는 방식보다는 여러 사람들이 상호간에 협력하여 문제를 해결하는 협업 방식이 더 효율적인 문제해결 방식으로 자리 잡고 있는 추세이다. 이러한 흐름은 산업뿐만 아니라 학문 분야에서도 융합이 이슈화되면서 상호간의 협력이 절대적으로 필요하게 되었으며, 이에 부응하기 위하여 공학교육에 있어서의 학습방법도 새롭게 변화하고 있다.

학습자와 교수자, 학습자와 학습자간의 상호작용 방식의 틀을 의미하는 학습구조는 개별학습, 경쟁학습, 협력학습, 일체학습 등으로 분류할 수 있는데 이 중에서 협력학습 구조가 학습 효과가 가장 높은 것으로 알려져 있다[1]. 협력학습은 공동의 학습목표를 이루기 위해 그룹 구성원끼리 활발한 학문적 상호작용을 통하여 학습 효과를 극대화시킬 수 있다. 학습목표 달성을 위해 학습자와 학습자 혹은 학습자와 교수자 사이의 인지적인 협력에 초점을 두는 형태로 학습자들이 개별적으로는 해결하기 어려운 과업을 서로 협동하면서 공동으로 과업에 참여하여 공동의 학습목표를 달성한다. 이를 위하여 SNS를 통한 정보의 공유, 상호작용, 주도적인 참여가 필수적이며 인터넷과 모바일 기기를 활용하는 것이 중요한 매개수단이 되는 것이다[2][3].

이와 같은 협력학습 필요에 맞추어 실제 실험과 실험 교육에 도움을 주는 원격 실험실에 협력학습 시스템을 도입하는 것이 필요하므로[4], 본 논문에서는 인터넷이나 이동형 기기인 스마트폰을 이용한 협력학습 시스템을 적용하여 시간과 장소에 제약 없이 원격 실험실을 공유함으로써 학습자간 협력학습 환경을 지원하는 방법을 제안하였다. 또한 이를 위한 방법으로 학습자간 협력학습을 통해 구현하고자 하는 프로그램 개발의 기본이 되는 프로그램 코드를 공유하는 방법을 제시하며, 실시간으로 그룹 참여자간의 문제해결을 위한 대화를 할 수 있도록 대화방을 연동하는 방법을 제시하고 이를 구현하였

다. 제안한 마이크로컨트롤러 협력학습 시스템(MCLS, Microcontroller Collaborative Learning System)은 Java와 Android, Apache, 포트포워딩, 텔넷, 소켓통신 기술을 사용하여 구현하였으며, 카톡(Kakao Talk)과 같이 일반적으로 많이 사용하는 채팅 프로그램과는 달리 채팅과 코딩을 동시에 할 수 있다는 장점이 있다.

이와 같은 필요에 의해 구현된 논문의 구성을 살펴보면 논문의 2장에서는 협력학습시스템의 개발 동향에 대해 다루었으며, 3장에서는 마이크로컨트롤러 원격실습실(MRLS, Microcontroller Remote Laboratory System)의 전체적인 구조와 기능별로 구분된 하드웨어 시스템과 이를 운영하기 위한 협력학습시스템 설계 및 관리시스템에 대하여 기술하였으며, 마지막으로 4장에서는 결론을 기술하였다.

II. 협력 학습시스템의 개발 동향

최근의 기술 발전은 기업 및 조직의 직원들의 협력 및 교육에 영향을 미쳤을 뿐만 아니라 학습 및 교육에도 큰 영향을 미치고 있다. 소셜 미디어, 게임, 가상 세계, 웹 2.0과 같은 새로운 트렌드는 새로운 방향으로 학습을 유도하고 있다. 모바일 장치, 사물 인터넷 및 클라우드 컴퓨팅과 같은 기술을 이용함으로써 공동 작업 개념을 훨씬 더 중요하고 실현 가능하게 만들었다. 이러한 협력학습에 관련한 세계적인 연구 동향을 살펴보면 다음과 같다.

컴퓨터 과학 교육을 위한 원격 협업학습 도구로 원격 교사와 공동으로 소프트웨어 엔지니어링 실습 과정을 지원하는 클라우드 기반 도구이다. 이 시스템은 클라우드 스토리지 플랫폼을 사용하여 멀티미디어 학습 자료를 공유하고 프로젝트 개발주기를 보다 효율적으로 관리한다. Virtual Debug Laboratory 라는 원격 협업 구성 요소는 동일한 팀에서 학생들의 디버깅 경험을 향상시키고 공유하도록 설계되었다[5].

미국의 루이지애나 주 뉴 올리언즈의 DU(Dillard University) 생물학과와 Bangalore의 BMS 공과 대학 화학 공학과(BMSCE)에서 개발한 협력시스템은 모바일 기술을 사용하여 개발된 시스템으로 쓰기 및 의사소통 및 협업 및 공유 리더십을 배울 수 있는 환경을 제공하고 있다[6].

컴퓨터 네트워크 교육에 있어서 많은 독립형 에뮬레이터를 사용하는 것을 개선하기 위해 개발된 HVLab는 네트워크 에뮬레이터와 가상교육기술을 통합한 웹기반 하이브리드 가상실험실로서 학생들이 혼자서 실습할 수 있도록 여러 네트워크 에뮬레이터를 제공하여 공유하고 학생들 간의 실험에 관련하여 토론을 촉진하는 협력학습모듈 등으로 구성되었다[7].

인도의 ABC들은 Scratch와 Alice와 같은 많은 비주얼 프로그래밍 도구가 갖고 있는 코드 블록, 애니메이션, 장면, 캐릭터 등을 통한 시각적인 프로그래밍 인터페이스를 제공하며, 학생들이 스스로 개념을 학습하는 특징을 갖춘 code.org와 Solo Learn C ++ 도구의 개념을 접목하여 새로운 협동 학습을 위해 개발되었다[8].

프랑스에서 개발된 PLACIS 시스템은 엔지니어링을 위한 협업 플랫폼으로 공동 프로젝트를 통해 능동적인 학습 및 교육을 촉진하는 시스템이다. 미래의 학습 플랫폼과 같은 새로운 도구 개발, 분산된 학습 환경 적응형, 협업형, 시맨틱 및 소셜 웹 3.0을 기반으로 개발되었다[9].

소프트웨어 공학 과정에서 구현된 Github와 같은 사이트는 교수용 미디어의 도움으로 협업 방법을 결합한 SCL(Student-Centered Learning)을 기반으로 한 학습 방법으로, 이는 웹 기반 소프트웨어로서 응용 프로그램 소프트웨어 프로젝트의 개발을 위한 공동 작업 매체로 사용하여 학생들이 학습목표를 달성하기 위하여 상호간에 협력할 수 있도록 도와준다[10].

그러나 지금까지 개발되어 사용되고 있는 여러 가지 협력학습시스템은 대부분의 경우 대중적인 SNS 프로그램을 이용하여 학습자 상호간에 토론을 하거나 의견을 공유하는 방식을 채택하는 것이 대부분이다[11]. 실제로 온라인상에서 공학교육을 진행할 때 소스코드에 대한 프로그래밍 과정이 요구되는데 이때 소스코드의 수정이나 소스코드의 실행 과정 등에 대한 학습자들 간의 상호 토의가 범용의 SNS 프로그램을 통하여 수행할 수 없다.

따라서 본 연구에서는 온라인 협력학습을 진행하는데 있어 학습자들 상호간의 채팅뿐만 아니라 소스코드 프로그램 코딩과 실행결과를 동시에 공유할 수 있는 협력학습시스템을 제안하고자 한다. 제안된 협력학습시스템은 채팅을 통한 상호토론과 소스코드의 수정 및 실행과정이 동시에 이루어지도록 설계되었기 때문에 학습의 연속성과 효율성이 매우 높다. 본 논문에서 제안한 협력학습방식은 많은 공학분야에서 활용될 수 있지만 유효성을 입증하기 위하여 공학 분야에서 매우 중요한 MRLS를 예로서 선정하고 MCLS를 구축하였다.

III. MCLS와 관리 시스템의 설계 및 구현

3.1 MRLS

그림 1에 나타난 MRLS은 원격지에 설치된 장비를 시간과 장소에 무관하게 인터넷을 이용하여 실험을 수행하는 시스템으로 학습자는 실제 장비가 없어도 원격으로 실제 실험을 진행할 수 있다는 장점이 있다[12]-[14].

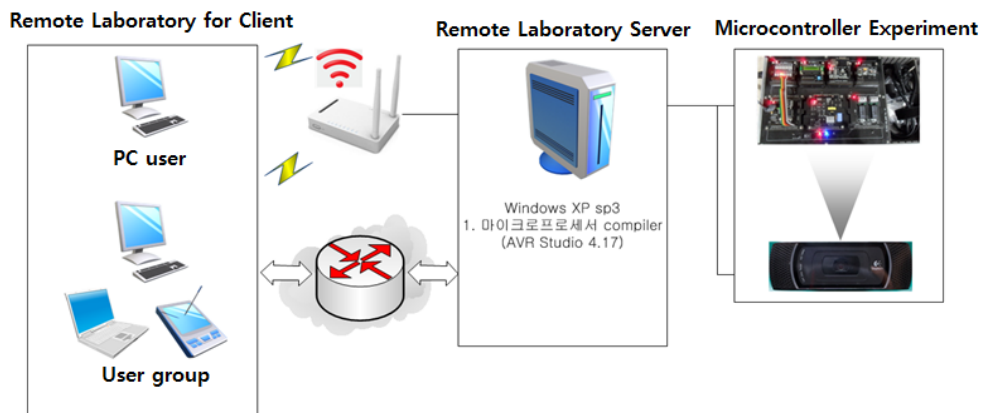


그림 1. 마이크로컨트롤러 원격실험시스템 구성
Fig. 1. Configuration of microcontroller remote laboratory system

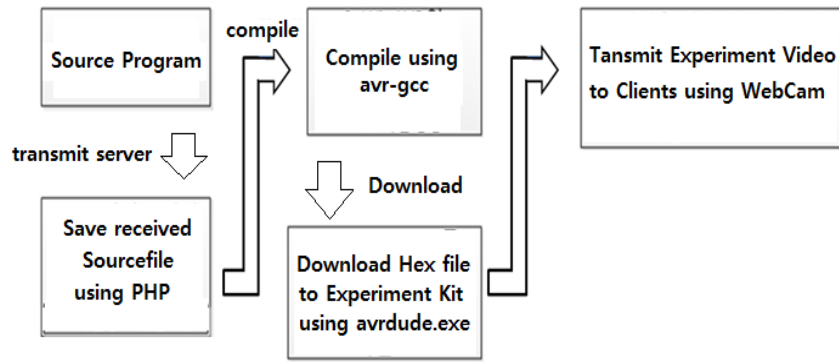


그림 2. MRLS에서의 데이터 흐름도
Fig. 2. Data flowchart of MRLS

본 논문에서 구현된 MRLS에서 학습자들은 LED, LCD, RELAY 등의 모듈로 이루어진 마이크로컨트롤러의 각 PORT와 현재의 결선상태를 확인하고 마이크로컨트롤러 실습을 진행하기 위한 소스코드를 클라이언트 프로그램을 이용하여 작성하고 동작유무를 웹캠을 통해 확인 할 수 있다.

또한 학습자가 작성한 소스코드를 Apache 웹서버에 업로드하면 웹서버에서 소스코드를 수신하고 웹서버에 있는 php파일에서 소스코드를 학번과 컴파일 시간으로 이루어진 ID+compile Time.c 파일로 저장한다. 파일명에 학번과 컴파일 시간이라는 요소를 넣고 학생들의 학습내용을 데이터베이스에 저장함으로써 나중에 자신이 학습하던 내용을 데이터베이스로부터 호출하여 협력학습을 연속적으로 진행할 수 있도록 하였다.

지금까지 기술한 MRLS에 대한 데이터 흐름도를 그림 2에 도시하였다. MRLS의 구현을 위하여 실제 적용된 알고리즘은 모두 자바언어를 이용하여 구현하였다. 학습방 서버는 Java Swing을 이용하여 구현하였으며, 클라이언트가 서버에 접속하기 위해 Java 소켓통신을 사용하였다. PC용 클라이언트 프로그램은 Java Applet으로 구현되어 통신을 위해 자바 소켓통신을 사용하였으며, 스마트폰용 클라이언트 앱은 안드로이드용으로 제작되어 데이터 통신을 위해 자바 소켓통신을 사용하였다.

클라이언트에서 컴파일 버튼을 클릭하면 작성된 소스코드가 서버로 전송되어 컴파일되는데 이때 전송을 위하여 Java의 HttpURLConnection 클래스와 OutputStreamWriter, PrintWriter 클래스를 이용하여

서버에 데이터를 보내고 실제 특정 URL에 데이터 전송을 시행하였다. 서버에서 데이터 수신은 php 파일에서 이루어지는데 각 데이터의 이름은 studentNum과 experiment, code이며 각 데이터의 값은 m_studentNum과 strDetailExperiment, m_code이다. 또한 각 변수는 학번, 실험 모듈, 소스코드에 대한 정보를 담고 있다. 작성된 소스코드를 서버로 보내어 컴파일되어 생성된 hex파일이 마이크로컨트롤러로 다운로드 되면 webcamXP라는 웹캠 스트리밍 프로그램을 이용하여 웹캠을 특정 포트에 스트리밍하게 하여 클라이언트에서 마이크로컨트롤러의 동작모습을 웹캠을 통해 볼 수 있도록 하였다.

3.2 MCLS

본 논문에서는 3.1절에서 구현한 MRLS에서 협력학습의 개념을 도입하여 MCLS를 구현하였다. PC 환경에서 MCLS에 참여하기 위해서는 웹 브라우저를 통해 접속하고 학번과 비밀번호를 이용하여 로그인하여 “마이크로컨트롤러(ATmega128)” 학습방을 만들면 된다. 한편, 모바일 스마트폰에서는 ‘협력학습’ 탭을 선택하여 “마이크로컨트롤러(ATmega128)”을 선택하면 된다.

세 명의 학생이 모여 MRLS를 이용해 협력학습을 하는 경우를 예로 제시한다. 참여하는 세 명의 학습자를 편의상 HAN, LEE, KIM 이라고 한다. 협력학습에서는 모든 학생이 동일한 지식이 있는 것이 아니다. 예를 들어 HAN은 FND의 각 세그먼트를 제어하기 위해 대입하여야 할 값을 알고 있으며,

KIM은 FND를 제어하는 방법을, LEE는 LED를 제어하는 방법을 알고 있어서 FND 코드가 만들어진 후 LED 기능이 동시에 동작하도록 할 수 있다고 가정하자. 이와 같이 학습자 개개인은 부분적인 지식을 가지고 있지만 협력학습을 통해 서로의 지식을 적절히 잘 조합한다면 공동의 목표를 달성할 수 있다.

MCLS에 접속하면 학습자들은 마이크로컨트롤러 협력학습 학습방(①)을 더블클릭하고 ‘참여’ 버튼을 클릭하면 학습방에 대한 정보가 나타난다. 학습자가 학습방에 참여한 후 그룹 채팅을 통해 의견을 교환하여 코드작업을 할 사람을 한 명 정한 다음 협력학습을 진행하면 된다.

다음은 세 명의 학습자가 한 그룹을 이루어 MCLS을 이용해 숫자를 1~8의 순서로 LED와 FND에 동시에 디스플레이 되는 과정에 대한 대화내용을 나타내었다.

[HAN] 8개의 LED를 순서적으로 점등시키면서 FND에 1~ 8의 숫자를 표현해봅시다.
 [LEE] 좋아요.
 [KIM] LED만큼 간단한건 없죠. 신호를 주면 켜지니까요.
 [HAN] 그렇죠. 각 비트에 1을 주면 켜지게 되겠죠. LED에 PORTB를 연결한다고 했을 때 제일 처음 해야 할것은 포트의 방향 설정이에요.
 $DDRB = 0xFF;$
 이렇게 하면 되요. 저건 포트 B의 모든 비트를 출력으로 쓰겠다는 뜻이에요. 1씩 더하면 처음에는 LED가 하나가 켜지고 두 번째는 2가 되니까 2번 LED만 켜질 것이고 3번째는 3이 되니까 첫 번째 LED와 두 번째 LED가 켜지겠네요. 우리는 0번 비트 LED부터 1번, 2번, 3번 이런 식으로 차례로 켜고 싶은데 어떻게 해야 할까요? 아는 사람 있나요?
 [LEE] 비트 이동을 하면 간단히 할 수 있어요. $0x01 \ll i$ 값을 루프마다 1씩 비트 $0x01 \ll i$ 이동하는 거예요.
 [HAN] 그렇지요. 그렇게 하면서 마지막 자리에 왔을 때 led 변수의 값을 1로 돌려주면 되겠네요.
 [KIM] FND는 Dynamic 방식으로 해보는 것이 어떨까요?
 [HAN] 그래요. FND를 켜기 위한 16진수 값은 $0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80$ 이예요.

```
[KIM] : 제가 FND 코딩을 추가해볼게요.
#define F_CPU      16000000UL
#include <avr/io.h>
#include <util/delay.h>
unsigned char fnd[16]={0xC0, 0xF9, 0xA4, 0xB0,
                      0x99,0x92,0x82,0xF8, 0x80};

int main( ){
    int i = 0, j = 0;
    DDRC = 0xFF; // FND의 출력
    DDRD = 0xFF; // FND의 선택 출력
    PORTD = 0xFF;
    while(1){
        for ( i = 0; i < 9; i++) {
            PORTB = 0x01 << i;
            for(j = 0; j < 50; j++) {
                PORTD = 0xFE;
                PORTC = fnd[i];
                _delay_ms(2);
                PORTD = 0xFF;
                PORTD = 0xF8;
                PORTC = fnd[i];
                _delay_ms(2);
            }
            .....
        }
    }
}
```

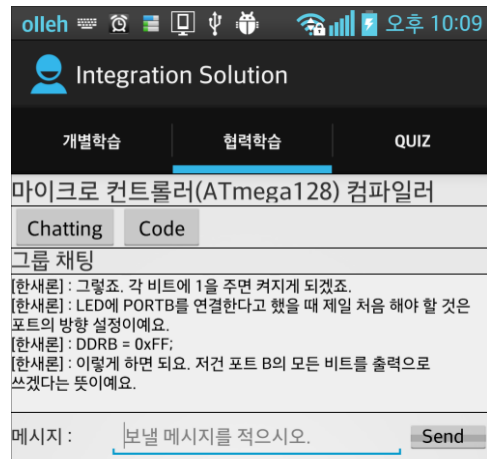


그림 3. 협력학습을 위한 그룹채팅
 Fig. 3. Group chatting for collaborative learning

그림 3에 MCLS에서 협력학습을 위한 그룹채팅 과정을 화면으로 캡처하였다. 대화 및 코드 메시지 전송시 적용된 프로토콜 문자는 “TALK”와 “CODE_CHANGE”로 클라이언트에서 보내는 메시지 앞에 프로토콜 문자를 붙여서 보내면 서버에서 이를 분석하여 그룹에 참여한 학습자들에게 전송해준다. 이때 각 토큰 분리자로 ‘/’ 문자를 추가하였으며, 서버는 메시지를 받았을 때 ‘/’ 문자를 기준으로 토큰을

분리한다. 사용자 ID가 학번으로 쓰여서 독립적이기 때문에 방 제목을 메시지에 포함시키지 않았으며, 그 대신에 학습방을 분류하기 위해 학습방 이름과 달리 뒤에 사용 모듈명이 사용되었다. 표 1에 메시지 형식을 보였다.

표 1. 메시지 형식
Table 1. Message format

TALK	User ID	message	
CODE_CHANGE	User ID	Line No.	Code

그림 4에서처럼 클라이언트가 메시지를 받으면 대화 메시지인지 코드 메시지인지 분석하여 대화 메시지이면 채팅창에 메시지를 보여주고 코드 메시지이면 전송받은 메시지를 바탕으로 코드를 동기화해주게 된다. 이러한 과정을 거쳐 클라이언트는 채팅으로 의견교환과 동시에 코드가 동기화되어 협력 학습을 할 수 있다.

그림 5는 알고리즘의 소스 코드를 채팅과정을 통해 얻어지는 과정의 예를 보여주고 있다. 모든 소스 프로그램을 작성하는 코딩 작업이 끝나면 코드 입력창 우상단의 'compile' 버튼을 이용해 컴파일 할 수 있고 컴파일 결과를 그림 6에서와 같이 확인할 수 있다. 컴파일이 성공적으로 이루어지면 서버에 작성한 소스코드에 대한 기계어코드(헤사파일)가 생성된다. 원격학습자들은 PC에서는 'Download', 스마트폰에서는 'Download to Kit' 버튼을 클릭하여 생성된 헤사파일을 실습키트로 다운로드할 수 있다.

헤사파일이 원격지에 위치한 실습키트장비로 다운로드 되면 그림 7과 같이 새로운 탭 또는 새로운

창을 통해 웹캠을 통해 실습키트가 동작하는 모습을 볼 수 있게 된다.

```

소스 Compile
#include <avr/io/h>
#include <util/delay.h>

unsigned char font[16]={0xC0,0xF9,0xA4,0x80,0x99
int main( ){
    int i=0 , j=0 ;
    DDRB = 0xFF;
    DDRC = 0xFF;
    DDRD = 0xFF;
    PORTD = 0xFF;
    while(1){
        for( i=0; i<9; i++){
            PORTB = 0x01<<i ;
            for(j=0; j<50; j++){
                PORTD = 0xFE ;
                PORTC = font[i] ;
            }
        }
    }
}
    
```

그림 5. 소스 프로그램
Fig. 5. Source program

```

Output Download to Kit
C:\APM_Setup\htdocs\ISFUS\03_MicroController\ISFUS_source\LED_20161106071925\20109036_2016_11_06_07_21_50.c is saved.
link success!
compile success!
    
```

그림 6. 컴파일 결과
Fig. 6. Result of compile

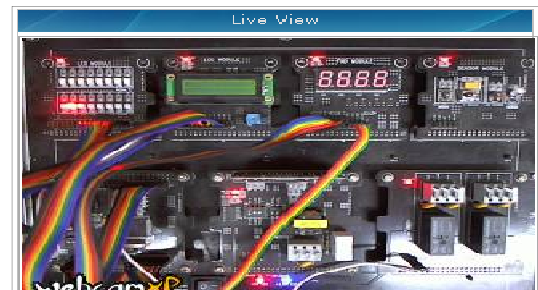


그림 7. MCLS에서의 실행 결과
Fig. 7. Execution result in MCLS

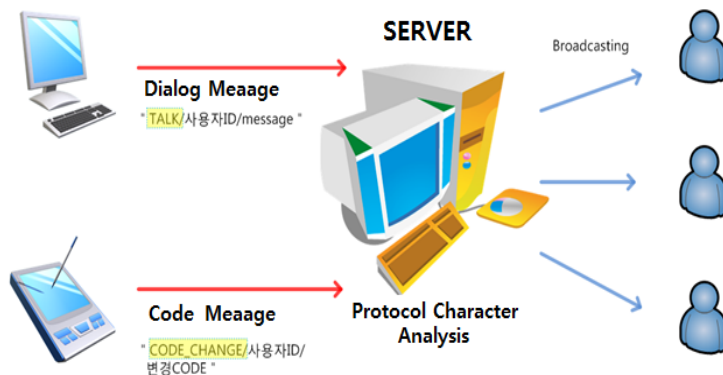


그림 4. MCLS에서의 프로토콜
Fig. 4. Protocol in MCLS

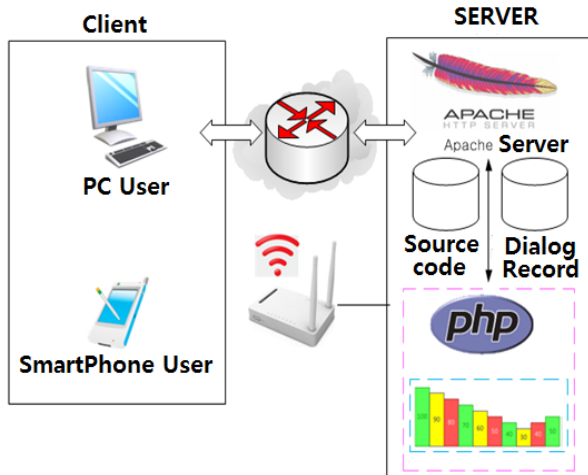


그림 8. MCLS 관리 시스템
Fig. 8. MCLS management system

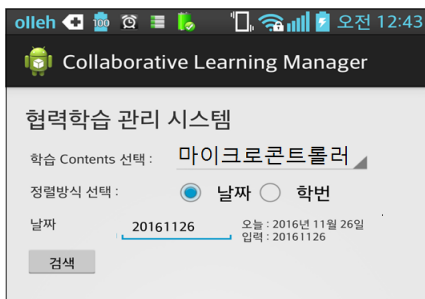


그림 9. MCLS 관리 시스템의 검색창
Fig. 9. Search Window of MCLS management system

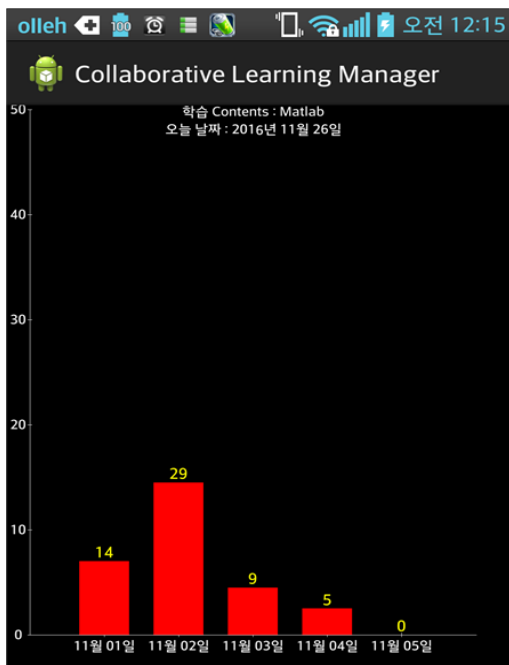


그림 10. 날짜별 협력학습 분량 검색
Fig. 10. Searching the amount of collaborative learning by date

이와 같이 구현된 MCLS를 이용하면 원거리에 떨어져 있는 학습자들 간에 다양한 아이디어와 지식이 통합되어 개별적으로 해결하기 어려운 문제들을 해결해 나갈 수 있다.

2.3 MCLS의 관리 시스템 설계 및 구현

제안된 MCLS의 효율적인 운영을 위하여 교수자의 관리 및 지도가 필요하기 때문에 본 논문에서는 MCLS와 함께 관련된 학습 데이터를 관리하고 학습 지도를 위한 협력학습 관리 시스템의 데이터 흐름도는 그림 8과 같다.

스마트폰용 MCLS 관리 시스템에서 온라인 협력 학습을 날짜로 검색할 수 있도록 설계하였으며 이를 그림 9에 나타내었다. 또한 그림 9에서 날짜를 입력하여 검색을 하게 되면 그림 10에 나타낸 것과 같은 그래프를 통해 MCLS의 해당 날짜에 이루어진 협력학습 사용 현황이 디스플레이 되도록 설계하였다.

그래프를 통하여 협력학습의 진행 정도를 볼 수 있을 뿐만 아니라 그래프 위에 수치 값을 통하여 협력학습에 참여한 학습자들의 참여도를 평가할 수 있도록 하였다. 또한 그래프에서 날짜별로 시스템의 사용빈도와 함께 참여한 학습자들의 상세 사용내역을 볼 수 있도록 지원함으로써 선택한 날짜에 학습한 마이크로컨트롤러의 소스코드와 협력학습 과정에서 학습자들간에 나누었던 채팅 대화 내용이 데이터베이스로부터 호출되어 제공되도록 설계하였다.

IV. 결 론

최근 컴퓨터 기술과 유무선 인터넷의 발전에 따라 모바일기반 협력학습이 새로운 교육 패러다임으로 제시되고 있다. 기존의 유선 네트워크 환경에서 개인 컴퓨터를 활용하여 학습하던 단계에서 벗어나 무선 네트워크 환경에서 시간과 장소의 제약 없이 컴퓨터, PDA, 노트북, 스마트폰 등을 사용하여 학습자들의 학습 효과를 높이는 단계로 진일보하고 있다.

지금까지 개발되어 사용되고 있는 여러 가지 협력학습시스템은 대부분의 경우 대중적인 SNS 프로그램을 이용하여 학습자 상호간에 토론을 하거나 의견을 공유하는 방식을 채택하는 것이 대부분이다. 실제로 온라인상에서 공학교육을 진행할 때 소스코

드에 대한 프로그래밍 과정이 요구되는데 이때 소스코드의 수정이나 소스코드의 실행과정 등에 대한 학습자들 간의 상호 토의가 범용의 SNS 프로그램을 통하여 수행할 수 없다는 단점을 가진다.

따라서 본 연구에서는 온라인 협력학습을 진행하는데 있어 학습자들 상호간의 채팅뿐 만 아니라 소스코드 프로그램 코딩과 실행결과를 동시에 공유할 수 있는 협력학습시스템을 제안하여 MRLS를 통하여 그 유효성을 입증하였다. 제안된 MCLS는 채팅을 통한 상호토론과 소스코드의 수정 및 실행과정이 동시에 이루어지도록 설계되었기 때문에 학습의 연속성과 효율성이 매우 높다.

더욱이 본 논문에서 제안한 협력학습방식은 많은 공학 분야에서 활용될 수 있기 때문에 여러 다양한 분야에 적용이 가능하다는 장점을 가진다. 또한 서버와 MRLS 구축을 위한 실습키트 구매비용을 제외하고는 MCLS 구축 비용이 전혀 들지 않으며, 협력학습 뿐만 아니라 개인학습을 위해서도 사용이 가능하다는 장점을 가진다.

References

- [1] SaeRon Hann, SeongHwan Kim, KwanSun Choi, Jong-Sik Lim, Dongsik Kim, SunHeum Lee, and HeungGu Jeon, "Implementation of a Remote Electrical Circuit Laboratory using ATmega128", *The Journal of Korean Institute of Information Technology*, Vol. 8, No. 11, pp. 215-224, Nov. 2010.
- [2] Luis Ramírez-Donoso, Juan S. Rojas-Riethmuller, Mar Pérez-Sanagustín, and Andrés Neyem, "Enhancing collaborative learning in higher education online courses through a mobile game app", *IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 103-108, Apr. 2017.
- [3] L. M. Sánchez Ruiz, S. Moll, M. D. Roselló, and N. Llobregat-Gomez, "Mathematics micro flipped and micro collaborative learning with sophomore Aerospace Engineering students", *World Engineering Education Forum - Global Engineering Deans Council (WEEF-GEDC)*, pp. 1-4, Nov. 2018.
- [4] A. J. Shalini Lakshmi and M. Vijayalakshmi, "Next Generation Classroom Learning Using Mobile Cloud Computing", *Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM)*, pp. 1-6, Feb. 2017.
- [5] Qing Ding and Sitan Cao, "A Cloud-Based Learning Tool for Graduate Software Engineering Practice Courses With Remote Tutor Support", *IEEE Journals & Magazines*, Vol. 5, pp. 2262-2271, Feb. 2017.
- [6] Julie Basu Ray and Samita Maitra, "Collaborative Interdisciplinary Teaching and Learning Across Borders, Using Mobile Technologies and Smart Devices", *5th IEEE International Conference on MOOCs, Innovation and Technology in Education (MITE)*, pp. 77-82, Nov. 2017.
- [7] Wu-Yuin Hwang, Michaele Haregot, and Chaknarin Kongcharoen, "Web-based hybrid virtualization laboratory to facilitate network learning: HVLab", *2nd International Conference on Information Technology (INCIT)*, pp. 1-6, Nov. 2017.
- [8] Sneha R Deshmukh and Vijay T Raisinghani, "ABC: Application Based Collaborative approach to improve student engagement for a programming course", *IEEE Ninth International Conference on Technology for Education (T4E)*, Vol. 2, pp. 20-23, Dec. 2018.
- [9] Jeffrey M. Kurcz, Ting-Wen Chang, and Sabine Graf, "Improving Communication and Project Management through an Adaptive Collaborative Learning System", *IEEE 15th International Conference on Advanced Learning Technologies*, pp. 61-63, Jul. 2015.
- [10] Azizah Zakiah and Mohamad Nurkamal Fauzan, "Collaborative Learning Model of Software Engineering using Github for informatics student", *4th International Conference on Cyber and IT Service Management*, pp. 1-5, Apr. 2016.
- [11] Didac Gil de la Iglesia, Juan Felipe Calderón, Danny Weyns, Marcelo Milrad, and Miguel

Nussbaum, "A Self-Adaptive Multi-Agent System Approach for Collaborative Mobile Learning", IEEE Transactions on Learning Technologies, Vol. 8, No. 2, pp. 158-172, Apr.-Jun. 2015.

[12] Yassine Khazri, Mohammed Moussetad, and Ahmed Fahli, "Implementing a Remote experience for engineering education", International Conference on Interactive Collaborative Learning (ICL), pp. 48-50, Sep. 2015.

[13] Parkhomenko Anzhelika, Gladkova Olga, Eugene Ivanov, Aleksandr Sokolyanskii, and Sergey Kurson, "Development and application of remote laboratory for embedded systems design", Proceedings of 12th International Conference on Remote Engineering and Virtual Instrumentation (REV), pp. 69-73, Feb. 2015.

[14] H. Mostefaoui and A. Benachenhou, "Design of a remote electronic laboratory", International Conference on Interactive Mobile Communication Technologies and Learning (IMCL), pp. 160-162, Nov. 2015.

이 순 흠 (Sunheum Lee)



1983년 2월 : 고려대학교
전자공학과(공학사)
1985년 2월 : 고려대학교 대학원
전자공학과(공학석사)
1989년 8월 : 고려대학교 대학원
전자공학과(공학박사)
1991년 9월 ~ 현재 : 순천향대학교

공과대학 정보통신공학과 교수

관심분야 : 모바일, 컴퓨터네트워크, 멀티미디어 콘텐츠, 임베디드시스템

김 동 식 (Dongsik Kim)



1986년 2월 : 고려대학교
전기공학과(공학사)
1988년 2월 : 고려대학교 대학원
전기공학과(공학석사)
1992년 8월 : 고려대학교 대학원
전기공학과(공학박사)
1992년 9월 ~ 현재 : 순천향대학교

공과대학 전기공학과 교수

관심분야 : 웹기반 가상실험실/원격실험실, 비선형제어, 지능제어시스템 설계

안 달 (Dal Ahn)



1990년 2월 : 서강대학교
전자공학과 초고주파
전공(공학박사)
1993년 이후 ~ 현재 : 순천향
대학교 전기공학 교수
관심분야 : 초고주파 수동소자,
각종 Filter, 부품 및

마이크로웨이브 회로 설계

전 흥 구 (Heunggu Jeon)



2003년 2월 : 순천향대학교
전자공학과 공학박사
2004년 8월 ~ 2005년 7월 :
한국통신정보기술
2010년 3월 ~ 2013년 :
안양대학교 전기공학과 겸임교수
2013년 3월 ~ 현재 : 순천향대학교

전기공학과 시간강사

관심분야 : 임베디드시스템, 가상 교육 콘텐츠개발, 영상처리

저자소개

한 새 론 (Saeron Han)



2014년 2월 : 순천향대학교
전기통신시스템공학과(공학박사)
2016년 11월 : 국립농업과학원
POST DOC.
2016년 11월 ~ 2017년 9월 :
위쉬정보기술
2017년 11월 ~ 현재 : 파루씨앤씨

재직 중

관심분야 : 임베디드시스템, 웹기반 하드웨어제어

최 관 순 (Kwansun Choi)



1994년 8월 : 서강대학교
전자공학과(공학박사)
1990년 3월 ~ 현재 : 순천향대학교
전기공학과 교수
관심분야 : 영상처리, GIS,
가상교육, 임베디드시스템, RFID,
웹기반 하드웨어제어,

교육용콘텐츠개발