The Journal of the Convergence on Culture Technology (JCCT)
Vol. 5, No. 4, pp.457-465, November 30, 2019. pISSN 2384-0358, eISSN 2384-0366

http://dx.doi.org/10.17703/JCCT.2019.5.4.457
JCCT 2019-11-58

안드로이드 플랫폼을 탑재한 스마트 온도제어기의 MMI 시스템 개발

Development of MMI System for Smart Temperature controller with Android Platform

이갑래*

Kap Rai Lee*

요 약 본 연구에서는 안드로이드 플랫폼을 탑재한 스마트 온도 제어기의 사용자 인터페이스(MMI)시스템에 대한 개발 방법을 나타낸다. 개발하는 스마트 온도제어기의 MMI 시스템은 온도 그래프 및 데이터를 원거리에서 모바일장치를 통하여 확인할 수 있도록 무선 WiFI 기능과 연동되는 시스템을 개발한다. 먼저, 스마트 온도제어기의 사용자 인터페이스(MMI)시스템의 하드웨어 인터페이스 회로설계방법을 나타내고, MMI 시스템의 운영 소프트웨어 개발 방법을 나타낸다. 스마트 온도제어기는 실시간 온도제어기 기능과 MMI 시스템 기능을 수행하는 이중 프로세서를 사용한다. 이어서 개발된 스마트 온도제어기의 MMI시스템의 실제 작동 시험을 통한 동작 성능을 평가한다. 안드로이드 플랫폼을 탑재함으로서 스마트온도제어기의 개발 기간을 줄일 수 있다.

주요어 : 스마트 온도제어기, MMI 시스템, 안드로이드 플랫폼, 이중 프로세서

Abstract This paper present developing methods of man-machine interface(MMI) system for smart temperature controller with android platform. This MMI system could communicate with mobile machine. Firstly we present electrical hardware design method of MMI system of smart temperature controller. Smart temperature controller is composed of dual processors. Secondly we develop operating software of MMI system using android development environment. And finally we present verification of MMI systems of smart temperature controller with android platform through field experiment. This MMI system with android platform has rapid development speed due to performance of android platform.

Key words : Smart temperature controller, Man-machine interface system, Android platform, Dual processors

Ⅰ. 서 론

넓은 온도 범위에서 작동하는 고급 온도제어기에 관한 연구로, 온도 제어 시스템의 다양한 제어 특성을 잘 반영하기 위하여, 오토 튜닝 기능을 적용하여 제어 시스템에 따른 제어 이득 값 설정이 스스로 튜닝 되도록 설계하여 사용하고 있다[1-8].

고급 온도제어기에서 사용하고 있는 제어기는 우수한 성능에도 불구하고 많은 문제점도 가지고 있다. 제어기는 특성상 사용하는 설정 항목이 많고, 사용자 인

*정회원, 평택대학교 스마트자동차학과 정교수
접수일: 2019년 9월 27일, 수정완료일자: 2019년 10월 21일
게재확정일자: 2019년 11월 6일

Received: September 27, 2019 / Revised: October 21, 2019
Accepted: November 6, 2019
*Corresponding Author: krlee@ptu.ac.kr
Dept. of Smart Automobile, Pyeongtaek Univ., Korea

터페이스(MMI)가 복잡하여 사용하기에 불편한 점을 들 수 있다. 제어기를 처음 접하는 사용자도 손쉽게 사용할 수 있도록 설계해야 하며, 다양한 제어대상의 특성에 대한 설정을 자동화하는 기능이 필요하다. 따라서 컬러 LCD 터치 스크린과 오디오 기능 등을 첨가하여 화려한 멀티미디어 기능을 갖는 사용자 인터페이스(MMI) 기능을 첨가함으로서 사용자가 사용하기 편하게 설계하여야 한다. 특히 모바일용 운영체제를 탑재하여 그래픽 정보를 개발함으로서 다양한 멀티미디어 전달에 개발 리소스를 많이 할애할 필요가 없으며, 모바일용 기기 어플 프로그램으로 손쉽게 이식 가능하게 해야 한다.

특히 사용자는 원거리에서 모바일장치 통하여 상태를 확인할 수 있어야 하며, 제어상태를 변경할 수도 있으며, 비상 알람에 대한 정보를 실시간으로 받아볼 수도 있는 고급 온도제어기에 대한 개발이 필요하다. 외형에서는 기존의 산업기기 형태에서 벗어나 스타일 시한 디자인 또는 수려한 그래픽 정보를 사용자에게 전달할 수 있는 것이 필요하다. 기존 시스템의 많은 키 입력은 도시 이미지와는 거리가 있으므로, 외부 키를 거의 없애고 LCD 터치패드를 사용해 설치되는 곳의 디자인을 세련되게 해야 하며, 터치 LCD를 사용하여 사용자들에게 많은 그래픽 정보를 전달할 수 있어야 한다.

따라서 본 개발에서는 스마트 온도제어기의 무선 WiFi 기능이 내장된 사용자 인터페이스(MMI)시스템을 개발한다. 기존 온도 제어기에서 음성 출력, 화면 디자인은 많은 시간을 요구하는 작업이므로, 안드로이드 플랫폼을 채택하여 개발함으로서 GUI 디자인에 우위를 점할 수 있게 하려고 한다. 온도 제어기능을 구현하기 위하여 빠른 처리속도를 수행할 수 있는 32bit RISC CPU를 사용할 것이며 터치패드를 사용해 입력키의 개수를 줄여 수려한 디자인이 가능하게 한다. 온도 그래프 및 데이터를 원거리에서 모바일장치를 통하여 확인할 수 있도록 무선 WiFI 기능과 연동되는 MMI 시스템 운용 프로그램을 개발한다. 이 어플리케이션은 모바일 기기와도 연동되도록 스마트폰용 어플리케이션 프로그램으로 쉽게 이식될 수 있도록 개발한다. 또한 MMI 시스템의 프로그램 수정 없이 모바일 스마트 기기에서 온도 제어기의 제어상태 및 비상알람에 대한 정보를 실시간으로 받아볼 수 있는 것이 가능하도록 서버시스템과 연동되는 것이 가능하도록 개발한다.

II. 스마트 온도제어기의 MMI 개발

1. 스마트 온도제어기 구성

스마트 온도제어기는 이중(Dual) 프로세서를 사용하여 설계한다. 실시간 온도제어를 하는 온도 제어기는 원칩 마이크로 컨트롤러를 이용하며 설계하며, WiFi 통신 기능을 포함한 MMI 시스템은 32bit RISC CPU를 이용하여 안드로이드 플랫폼을 탑재하여 구현한다. 실시간 온도 제어기와 MMI 시스템 보드는 병렬 또는 직렬 포트를 통하여 온도제어기 정보를 공유한다

스마트 정보기술이 적용된 스마트 산업용 기기에서는 기존의 산업용 기기가 하는 기능과 스마트 정보처리기능을 하나의 32bit Risc CPU를 사용하여 구현하고 있지만. 실시간 제어가 필요한 산업용 제어장치에서는 실시간 제어 성능을 고려한 제어기와 사용자 인터페이스 기능을 분리하여 설계하는 것이 필요하다. 스마트 산업용 기기에서 화려한 사용자 인터페이스 기능 작동 시에 사용자가 터치 입력을 잘못 입력함으로 인한 일어날 수 없는 조합이 발생 하여 오작동이 발생 할 경우에도 실시간 제어 성능은 영향을 받지 않고 동작하여야만 한다. 따라서 본 개발에서는 크게 실시간 제어를 위한 실시간 제어기와 스마트 정보처리를 위한 스마트 정보처리부(MMI 시스템부)를 분리하여 구성한다.

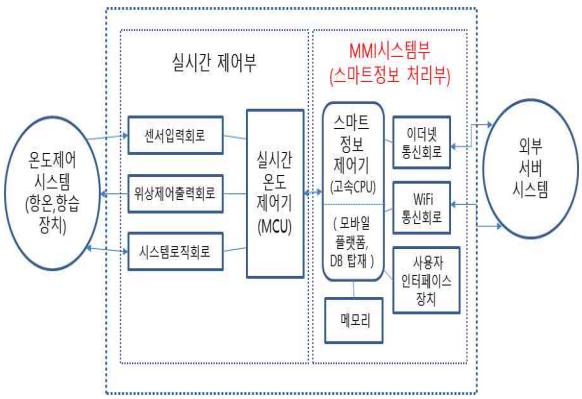

그림 1. 스마트 온도제어기 구조
Fig. 1. Structure of smart temperature controller

실시간 제어부는 센서입력회로, 위상제어출력회로, 시스템로직회로 및 마이크로 컨트롤러(MCU)로 구성하며, 이에 대한 상세한 제어기 설계 및 구현방법은 참고

문헌 [8]에 잘 나와 있다.

본 연구에서 나타내는 MMI 시스템부는 WiFi 통신 회로, 메모리(RAM, ROM), 고속 CPU, 사용자인터페이스 장치로 구성한다. WiFi 통신 회로는 인터넷 연결 및 외부 모바일기기와의 통신을 위한 통신회로이다. 사용자 인터페이스 장치는 사용자 프로그램이 동작하여 출력되는 LCD, 사용자가 입력하기 위한 터치스크린, 사용자 입출력회로로 구성된다. 고속 CPU는 사용자 인터페이스 프로그램을 수행하며, 실시간 제어기로부터 통신하여 전송받은 온도값과 시스템 정보값을 메모리에 누적 저장하며, 또한 외부 모바일 기기로부터 정보 요구 시에 전송하는 장치이다. 다음 그림은 MMI 시스템의 프로그램이 동작하는 흐름도이다.

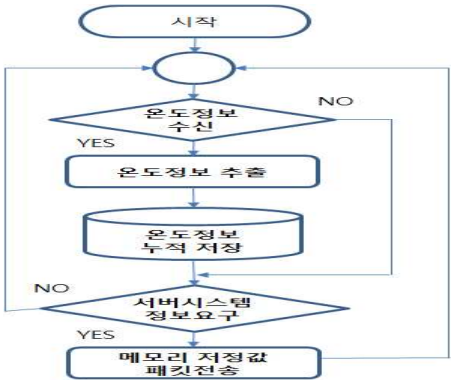

그림 2. MMI 시스템의 동작 흐름도
Fig. 2. Operation flowchart of MMI system

전원이 인가되면 스마트 정보처리기의 고속 CPU는 먼저 실시간 온도제어기의 마이크로 컨트롤러가 송신한 정보가 있는지를 주기적으로 수신 버퍼를 읽어서 통신 데이터로부터 약속된 통신 프로토콜을 사용하여 온도값과 시스템 정보값을 추출한다. 다음으로 추출된 온도값과 시스템 정보값을 탑재된 데이터베이스 엔진을 통하여 메모리에 누적 저장한다. 다음으로 사용자 인터페이스(UI)로부터 요구가 있을시 사용자 인터페이스 프로그램이 수행되어 온도값과 시스템 정보값을 LCD에 출력하게 된다. 또한 외부 모바일 기기부터 정보요구가 있을시 누적된 메모리 저장값을 데이터베이스(DB)엔진을 통하여 읽어와서 패킷 단위로 구성하여 외부 시스템으로 전송한다.

2. 스마트 MMI 시스템 개발

(1) 하드웨어 구성 및 인터페이스 설계

스마트 MMI 보드의 전체 하드웨어 구성은 그림 3과 같다.

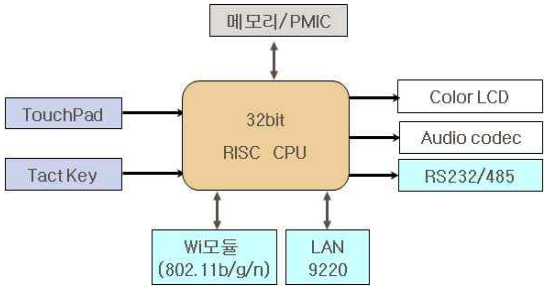

그림 3. MMI 시스템 하드웨어 구성
Fig. 3. Hardware structure of of MMI system

o CPU & Memory

본 시스템에서 사용하려는 32bit RISC CPU는 ARM Cortex이다. 휴대용 통신기기 및 산업용 제어시스템에 적합하며, 멀티미디어 환경에 최적화 되어 있다.

CPU, 메모리 및 PMIC가 포함된 CPU 모듈을 사용함으로서 고속 동작하는 영역을 실제 응용 보드에서 배제함으로써 신뢰성을 확보하고 추후 모듈의 저가개발에 대한 여지도 동시에 가지게 된다. 이에 따라 속도가 낮은 부품이 들어가는 어플리케이션 보드에 대해서는 물리적으로 분리함에 따라 PCB 층수를 줄일 수 있어 가격 경쟁력을 가지게 되었고 추후 파생모델을 개발하는데도 용이하게 하였다.

o LCD 및 UART

LCD는 한눈에 제어상태를 파악할 수 있는 넓은 화면과 깨끗한 화질을 위해 7인치 TFT LCD를 채택하였다. LCD Controller가 ARM Cortex에 포함되어 있으므로 Connector만을 통해 제어가 가능하며 주요 Signal로는 RGB line과 HSync, VSync Enable 등이 있다. Backlight는 PWM제어를 통해 밝기를 제어하게 된다.

실시간 온도제어기와 직접적인 RS-232통신을 통해서 제어 및 상태를 확인하게 된다. Power Switch를 통해서 High-current 및 Short등에 따른 전류에 대한 제한과 Overcurrent 발생 시 이를 알려주는 회로로 구성되어 USB에서 정해진 전류(500mA)이상이 되면 차단할 수 있도록 하였다.

o Audio Codec 및 Touch screen

특정한 이벤트가 발생했을 경우 음성 안내를 할 수

있도록 한다. I2C를 통한 Audio Codec을 제어하며 스피커 기능만을 사용한다. 제어방식은 mode pin을 GND로 하여 2-wire serial control mode로 한다.

Touch screen는 직접 I2C를 통해 키와 좌표데이터를 입력받게 된다.

o Power, SD card, Debug port

Power회로는 DC 전원을 입력받아서 필요한 전원을 LDO를 통해 5V, 3.3V등을 생성한다

SD memory로 booting을 하고 Android image, file system용도로 활용하도록 SD card slot이 부착된다. 전원은 PMIC로부터 3.3V를 공급받게 된다. Debug port는 boot monitoring, boot/os/file system upgrade를 위한 창구 역활을 위해 부착된다.

o WiFI

무선랜기능을 활용하여 내부 data를 전송하고 명령을 전달받기 위해 활용이 된다. SDIO #2를 통해 WiFi 모듈을 제어한다. 이 WiFi은 802.11b/g/n 표준을 따른다. 전원은 VBAT에 5V를 인가하고 VDD_IO에 PMIC로부터 공급을 받게된다. Bluetooth기능과 Combo이지만 여기서는 Bluetooth는 사용하지 않는다.

(2) 안드로이드 플랫폼 및 드라이브 설치

o 부트로드 빌드

본 개발보드에서 사용하는 부트로더는 U-boot 이다. 파워 PC와 ARM에 기반을 둔 임베디드 보드를 위한 부트로더이다. 시리얼 통신, 네트워크, USB, Flash를 지원하며, 각종 CPU에대한 명령어 체계가 동일하므로 다른 플랫폼에 이식성이 용이하다. make 명령을 통해 소스를 컴파일한다. 정상적으로 컴파일이 완료되면 u-boot.bin 이미지가 생성된다.

o. 커널 빌드

안드로이드 커널은 플랫폼의 핵심을 이루는 요소이다. 리눅스 커널과 비슷하나 Ashmen, Kernel Debugger, Binder, Power Management, Logger 등의 컴포넌트가 추가 확장되어 있다.

o. Android 빌드

안드로이드 플랫폼은 속도를 위한 사용자 인터페이스, 향상된 전원관리, 근거리 무선통신 기능들이 있으며, 네이티브 코드에서 입력이나 센서 이벤트를 직접 받아서 처리할 수 있는 개발자 기능들이 있다. 또한 자이로스코프, 회전, 선형가속, 중력, 기압계 등의 새로운 센서를 지원하는 API가 있다.

o. 드라이버 설치 및 다운로드

개발 보드에 전원을 인가하게 되면 새로운 하드웨어가 검색되어 드라이버 설치가 필요하다. USB 시리얼 포트 드라이버를 설치하고, ADB(Android Debug Bridge) 드라이버를 설치한다. ADB는 안드로이드 SDK에 포함되어 있는 기능으로 안드로이드 디버그에 관련된 툴로이를 이용해서 에뮬레이터 및 장치를 이용할 수 있고, 파일복사 및 어플리케이션 설치, 삭제 등의 작업을 할 수 있는 아주 유용한 유틸리티이다. 다운로드 환경이 구축이 되면 부트로더 이미지(u-boot.bin), 커널 이미지(zImage), 안드로이드 시스템 이미지(system.im, root.img)를 다운로드 한다

(3) 제어기 간의 통신 데이터 프로토콜 설계

실시간 제어기, MMI시스템, 모바일기기 간의 직렬 통신 데이터 프로토콜은 참조 논문 [8]과 유사하게 설계한다

(4) MMI 운용 프로그램 개발

개발 MMI시스템의 운용 프로그램인 챔버컨트롤러는 메인화면, 디바이스설정 화면, 디바이스상태 하면, 제어입출력그래프 화면으로 구성되어 있다. 운용 프로그램의 전체 동작은 설정메뉴에서의 설정된 메뉴에 따라 다음과 같이 동작한다.

- 메인화면 프로그램에서는 제어대상 시스템의 정보를 5개 채널를 통하여 동시에 나태내고 있다. 온도(또는 습도,기타)값 및 동작시간을 설정할 수 있으며, 동작중인 시스템의 현재 온도값을 나타낸다.
- 디바이스 설정 화면에서는 각 채널별로 제어기의 이득값을 설정 할 수 있다. 또한 출력 온도값 단위를 선택할 수 있다.
- 디바이스 상태 화면은 각 채널별로 문 열림, 냉각수,

온도가열, 시스템 운전 상태 등의 정보를 나타낸다.
- 제어입출력그래프 화면에서는 전체 시간대별 동작 추이를 나타내기 위하여 온도 출력값과 제어기 출력값을 그래프로 나타내고 있다.
- 이외에도 통신 포트 설정 화면에서는 통신포트 및 통신 보오레이트를 설정 할 수 있게 하였다.
- 각 화면에서 입력값 설정 및 상태값 출력시 필요에 따라 음성 메시지를 오디오에 출력 할 수 있게 하였다. 한다. 이 때 라이브러리 스택의 Media Framework 에 있는 MediaPlayer를 이용하여 오디오로 출력한다.

o 운용 메뉴 프로그램

운용 메뉴에 대한 설정 Activity 화면 프로그램의 개발 방법은 다음의 과정으로 이루어진다.

그림 4. 운용 메뉴 화면
Fig. 4. Operating monitor of MMI system

- 메뉴의 버튼은 설정 등의 메뉴는 Xml로 작성하며 setContView()를 호출하여 디바이스 설정 Activity화면을 뷰 위젯으로 채운다.

- 뷰 요소에 findViewById() 를 호출하여 주어진 리소스 id의 뷰를 찾는다. 또한 View의 onClickListener() 를 이용하여 뷰를 건드리거나 클릭했을 경우 해당하는 객체를 작동시킨다.

o 제어기 설정 화면 프로그램

메인 화면에 대한 설정 Activity 화면 프로그램의 개발 방법은 다음의 과정으로 이루어진다.

- mainscreen.xml은 폼을 구성한다. 온도입력을 위한 폼 구성요소, 습도입력을 위한 폼 구성요소와 START 및 전송 버튼을 배치한다. 폼 구성 용소인 뷰 클래스를 이용하여 폼을 설계한다. TextView, EditText, Button, TableRow 클래스들은 LinearLayout 클래스의 서브 클래스가 되어 LinearLayout 클래스의 속성을 상속받는다.

- MainScreen.java는 SerialPortActivity 클래스의 서브클래스로 정의되어 SerialPortActivity 클래스로부터 상속받는다. SerialPortActivity 클래스는 Activity 클래스로부터 상속받아 액티비티 생성시 입출력 통신에 관련된 메소드를 호출하여 통신관련 클래스를 정의한다.

-MainScreen.java는 mainscreen.xml에서 지정한 폼 구 성요소를 출력하고 전송버튼이 클릭되면 사용자가 입력한 정보를 추출하고 직렬통신으로 전송하는 코드를 작성한다. 또한 직렬통신으로 전송받은 데이터를 약속된 통신 규약에 따라 현재 온도 상태값을 추출하여 화면에 출력한다.

그림 5. 제어기 설정 화면
Fig. 5. Controller monitor of MMI system

o 포트 설정 화면

- 통신 포트 설정 화면에서는 통신포트 Serial 1 및 Serial 2 등을 선택할 수 있게 하였다.
- 또한 통신 보오레이트도 선택할 수 있게 하였다.

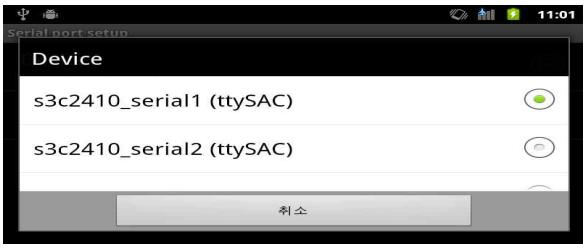

그림 6. 통신 포트 설정 화면
Fig. 6. Setting monitor of communication port

o 디바이스 설정 화면

- deviceset.xml은 레이아웃 내에 폼 구성요소를 배치한다. 채널번호 입력을 위한 폼 구성요소, 제어 이득값 설정을 위한 폼 구성요소, 온도표시를 위한 폼 구성요소와 확인(취소)버튼을 배치한다. 채널번호는 라벨, RadioGroup 및 RadioButton 클래스를 이용한다. 제어 어이득값은 라벨과 EditText 클래스를 이용한다. 온도표시는 라벨, RadioGroup 및 RadioButton 클래스를 이용한다. 확인 및 취소 버튼은 Button 클래스를 이용한다.

- DeviceSet.java는 SerialPortActivity 클래스의 서브클래스로 정의되어 SerialPortActivity 클래스로부터 상속받아 액티비티를 생성하고 deviceset.xml의 레이아웃을 화면에 출력한다.

- DeviceSet.java는 확인버튼이 클릭되면 사용자가 입력한 정보를 추출하고, 약속된 통신 규약에 따라 헤드 데이터를 첨가하여 직렬통신으로 전송하는 코드를 작성한다.

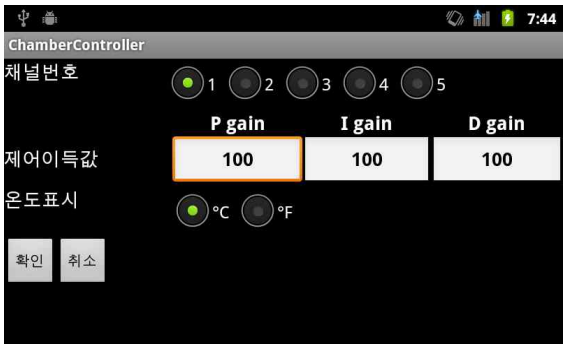

그림 7. 제어 디바이스 설정 화면
Fig. 7. Setting monitor of control device

o 디바이스 상태 화면

- devicestatus.xml은 레이아웃 내에 폼 구성요소를 배치한다. 채널번호 폼 구성요소, 도어 오픈 폼 구성요소, Water Low 폼 구성요소, Overtemp 폼 구성요소와 냉동기운전상태 구성요소를 배치한다. 폼 구성 용소인 뷰 클래스를 이용하여 폼을 설계한다. TextView, TableRow 클래스들은 TableLayout 클래스의 서브 클래스가 되어 TableLayout 클래스의 속성을 상속받는다.

- DeviceStatus.java는 SerialPortActivity 클래스의 서브클래스로 정의되어 SerialPortActivity 클래스로부터 상속받는다. SerialPortActivity 클래스는 Activity 클래스로부터 상속받아 엑티비티 생성 시 입출력 통신에 관련된 메소드를 호출하여 통신관련 클래스를 정의한다.

- DeviceStatus.java는 devicestatus.xml에서 지정한 폼 구 성요소를 출력하고, 직렬통신으로 전송받은 데이터를 약속된 통신 규약에 따라 디바이스 현재 상태값을 추출하여 화면에 출력한다. 또한 약속된 통신 규약에 따라 헤드 데이터를 첨가하여 응답하는 통신 코드를 작성한다.

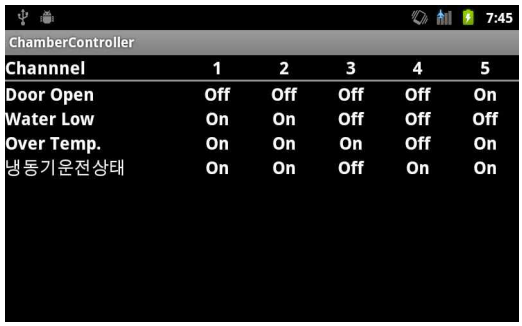

그림 8. 제어 디바이스 상태 화면
Fig. 8. State monitor of control device

o 제어기 성능 입출력 그래프 프로그램

- 입출력 그래프의 plot은 오픈소스인 Androidplot project Androidplot-core-0.4.4-release.jar를 사용한다.

- makePlotPretty() 함수를 호출하여 x축, y축 lable, step등 그래프 관한 기본 라벨 프로그램을 작성한다. seriesSetup() 함수를 이용하며 각 그래프에 대해 출력할 데이터를 연결한다. 이때 데이터는 tempHistory Series*이며 직렬통신으로 받으며, 통신으로 받은 데이타는 그래프와 text로 변경한다.

- 그래프는 10개의 값만 표시하고 가장 오래된 것은 지운다. 그래서 출력할 데이타를(tempHistorySeries*)에서 가장 오래된 것은 지운다.

o. 음성 출력 프로그램

sound 출력은 특정한 이벤트가 발생했을 경우 알리는 사운드를 출력한다. 또한 파라메터 설정 시 설정을 유도하는 사운드와 설정이 완료됨을 알리는 사운드를 출력한다. 사운드는 mp3 파일이며 MediaPlayer를 사용하여 play 한다. 사운드는 mp3 파일이며 MediaPlayer를 사용하여 play 한다. 사운드 출력의 경우 mp3 file을 위치를 MediaPlayer의 *audio_play*.setDataSource()로 play할 파일을 정하는 것으로 sound 출력 작업을 시작하며, *audio_play*.prepare()한 다음 *audio_play*.start()로 음악이 시작된다. 이미 음성이 출력되고 있는 경우 *audio_play*.stop() 및 *audio_play*.release() 함수를 이용하여, 그 음성을 중단하고 새로운 음성을 시작해야 한다. 메인화면이 종료될 때 사운드 자원을 반환해야 한다.

o. TCP/IP 통신 프로그램 작성

원격에서 모바일 장치를 통하여 확인할 수 있도록 웹서버와 통신을 위하여 TCP/IP socket통신을 사용한다. 기본적으로 server와 통신을 위한 기본조건인 Ethernet이 같은 network group에 있어야 한다. 안드로이드에서 인터넷통신을 하기 위해서는 먼저 AndroidManifest.xml에서 INTERNET permission을 추가해야한다. Client 동작 프로그램은 MMI시스템의 운용 상태 결과를 server로 보내는 동작을 구현하는 것이다. 이에 반응하는 server는 desktop이며 이를 처리할 프로그램은 ChammerConAccess이다. Client 동작 프로그램은 다음과 같다. 먼저 socket 통신을 하기 위하여 Server의 IP 및 port 번호를 android code에 추가하고 thread를 발생시켜야 socket통신이 가능하다. 다음, 운용 상태에 대한 데이터를 yte형태로 만들어 서버로 전송한다. 전송하는 코드는 out.write(w), out.flush() 함수를 이용하여 packet으로 서버에 전송한다.

o. WI-FI 연결

WIFI 설정은 안드로에서 제공하는 에플리케이션인 "설정" 프로그램으로 활성화할 수 있다. 먼저 Wi-FI를 click하여 Wi-Fi 블록을 동작 시킨다. 그 후 범위내에 있는 Wi-Fi를 검색하면 검색이 되며 가장 신호가 강한 무선 AP에 자동으로 연결된다. WEP 방식의 경우 자동으로 detect하며 비밀번호를 추가로 요구한다.

o. 언어 다중화

본 제품에서는 영어와 한글을 기본적으로 지원한다. 언어 전환은 기기 설정 프로그램으로 간단하게 할 수 있다. 그러나 안드로이드에서 동작하는 어플리케이션의 경우 미리 리소스를 각 언어에 대응하여야만 언어 전환이 가능하다. 다중언어를 지원하기 위해서서는 각국 언어의 문자열을 정의해야 한다(/res/values: default 언어, 영어, 한국어). 언어 전환은 먼저 select local을 선택하고, 이후 전환할 언어를 선택한다.

o. 모바일 스마트기기용 어플

MMI 시스템의운용 프로그램 중 "메인화면" 및 "디바이스상태 화면"은 모바일 스마트기기에 탑재하여 사용할 수 있다. 다만 스마트폰용일 경우 사용자 인터페이스 프로그램 중 가로 배열에 대한 레이아웃 화면을 세로 배열로 수정하여야 한다.

III. 성능 시험 평가

1. 시험 환경 구성

개발 시제품의 성능시험을 평가하기 위하여 항온항습장치 챔버와 본 개발시스템을 연결하여 성능 시험을 한다. 본 연구에서 개발된 스마트 온도제어기는 항온항습장치에 장착되며, 항온항습장치의 온도센서와 히터에 연결된다. 이 구성도를 나타내는 그림은 아래 그림과 같다.

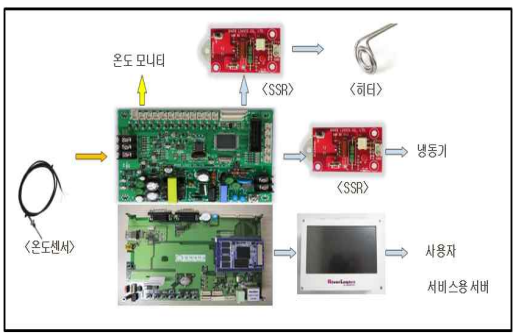

그림 9. 시험 환경 구성도
Fig. 9. Diagram of test environment

2. 온도제어기 MMI 시스템의 성능

o 채널별 온도값 입력 및 출력

아래 그림은 채널별 온도값 입력 및 출력값을 실시간으로 나타내고 있다. 5개 각 채널에 대한 입력 및 출력값을 실시간으로 나타내고 있다.

그림 10. 제어기 설정 화면 출력
Fig. 10. Output of controller monitor

o 채널별 디바이스상태 출력

아래 그림은 채널별 디바이스 상태 출력을 나타낸다. 5개 각 채널에 대한 디바이스 상태(문열림, 수조물 없음, 가열, 운전상태)을 나타내고 있다.

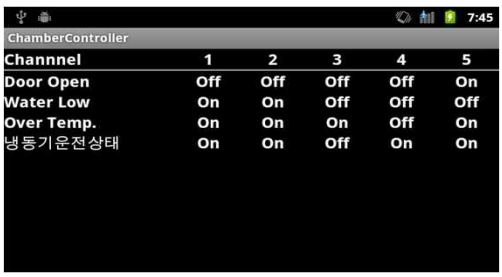

그림 11. 스마트제어기의 디바이스 상태 출력
Fig. 11. Output of device state for smart ontroller

o 채널별 온도값 궤적 출력

아래 그림은 채널별 온도값에 대한 입력 및 출력 궤적을 나타내고 있다.

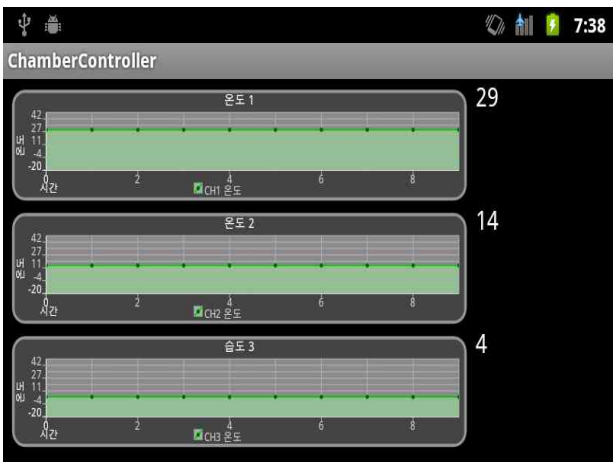

그림 12. 스마트 제어기의 온도값 출력
Fig. 12. Temperature output of smart controller

o 스마트폰 연동 제어기 성능

아래 그림은 본 연구에서 개발한 어플리케이션 프로그램을 WiFi 연결을 통한 스마트폰에서 제어한 원격 제어 및 출력 제어 응답을 나타내고 있다.

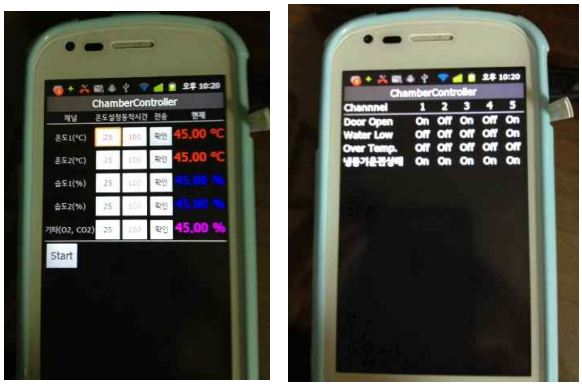

그림 13. 스마트 제어기의 모바일 기기 출력
Fig. 13. Output of smart controller via mobile device

V. 결 론

본 연구에서는 안드로이드 플랫폼을 탑재한 스마트 온도 제어기의 사용자 인터페이스(MMI)시스템에 대한 개발 방법을 나타내었다. 스마트 온도제어기의 MMI 시스템은 온도 그래프 및 데이터를 원거리에서 모바일장치를 통하여 확인할 수 있도록 무선 WiFI 기능과 연동되는 시스템을 개발하였다. 스마트 온도제어기의 사용자 인터페이스(MMI)시스템의 하드웨어 인터페이스 회로설계방법을 나타내고, MMI 시스템의 운영 소프트웨어 개발 방법을 나타내었다. 개발된 스마트 온도제어기의 MMI시스템의 실제 작동 시험을 통한 동작 성능을 평가하였다. 무선 WiFi 및 LCD 터치스크린 등

으로 수려한 디자인이 가능하며, 또한 기존 온도 제어기에서 음성 출력, 화면 디자인은 많은 시간을 요구하는 작업이므로, 안드로이드 플랫폼을 탑재하여 개발함으로서 GUI 디자인에 우위를 점할 수 있다.

References

[1] L. Wang, and W. R. Cluett, "Tuning PID controllers for integrating processes", IEE Proc. Control Theory Appl., 144, pp. 385- 392, 1997.
[2] K. Man-Seok, K. Jo-Hwan, C. Min-Koo, P. Jong-Oh, K. Jong-Wook, "Design of an Auto-Tuning IMC-PID Controller for a Heater System Using uDEAS," Korean Institute of Intelligent Systems Vol. 21, no. 4, pp. 530-535, 2011.
[3] L. Wang, "Automatic tuning of PID controllers using frequency sampling filters," IET Control Theory & Applications, vol 36, no 7, pp 985-995, 2017. DOI: 10.1049/iet-cta.2016.1284
[4] P. Poksawat, L. Wang, A. Mohamed, "Automatic tuning of attitude control system for fixed-wing unmanned aerial vehicles", IET Control Theory Appl., vol. 10, pp. 2233-2242, 2016. DOI: 10.1049/iet-cta.2016.0236
[5] Gyu-Hong Jung, "Auto Tuning of Position Controller for Proportional Flow Solenoid Valve," The Korean Society of Mechanical Engineers, vol 36, no 7, pp 797-803, 2012.
[6] S. Oh, "An Auto-tuning Algorithm of PI Controller Using Time Delay Element," Journal of the Institute of Electronics Engineers of Korea, vol 47(SC), no 6, pp 1-5, 2010.
[7] K. R. Lee, "Auto-tuning control using multi output sensor for wide band temperature control systems," Proceedings of KIIS Autumn Conference, vol. 25, no. 2, 2017.
[8] K. R. Lee, "Development of Auto-tuning Temperature controller with Multi-channel," The Journal of the Convergence on Culture Technology, vol. 4, no. 4, 2018.