



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0140100
(43) 공개일자 2016년12월07일

(51) 국제특허분류(Int. Cl.)

G06Q 50/10 (2012.01) C25D 5/08 (2006.01)
G01K 13/00 (2006.01) G06Q 50/04 (2012.01)
G08B 21/18 (2006.01) G08B 23/00 (2006.01)
G08B 25/10 (2006.01)

(52) CPC특허분류

G06Q 50/10 (2015.01)
C25D 5/08 (2013.01)

(21) 출원번호 10-2015-0076244

(22) 출원일자 2015년05월29일

심사청구일자 없음

(71) 출원인

인천대학교 산학협력단

인천광역시 연수구 아카데미로 119 (송도동)

(72) 발명자

김병수

인천광역시 연수구 해송로 70, 220동 901호 (송도동, 송도웰카운티2단지아파트)

(74) 대리인

특허법인충정

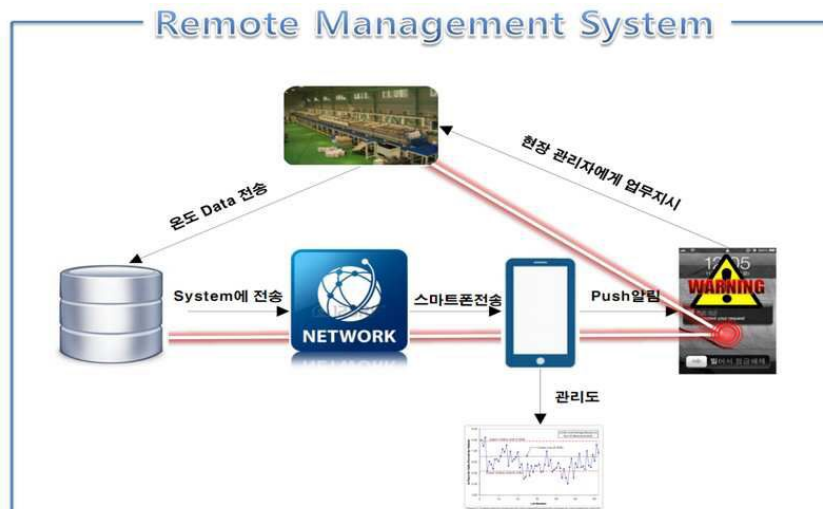
전체 청구항 수 : 총 3 항

(54) 발명의 명칭 도금공정 실시간 원격 모니터링 시스템

(57) 요약

본 발명은 전해물질의 온도를 측정하고, 측정된 온도에 해당하는 제1 온도 정보신호를 출력하는 온도센서, 상기 온도 센서로부터 출력된 제1 온도 정보신호를 수신하여, 상기 제1 온도 정보신호를 온도 신호 처리 알고리즘으로 처리하여 전해물질의 온도를 결정하고, 상기 결정된 온도에 해당하는 제2 온도 정보신호를 출력하는 중앙처리부 및 상기 중앙처리부로부터 출력된 제2 온도 정보신호를 수신하여 디스플레이하는 외부 단말을 포함한다.

대표도 - 도2



(52) CPC특허분류

G01K 13/00 (2013.01)

G06Q 50/04 (2013.01)

G08B 21/182 (2013.01)

G08B 23/00 (2013.01)

G08B 25/10 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 000000

부처명 교육부

연구관리전문기관 한국산업기술진흥원

연구사업명 산업단지캠퍼스 조성사업

연구과제명 송도산학그리드 캠퍼스 구축

기 여 율 1/1

주관기관 인천대학교 산학협력단

연구기간 2014.09.01 ~ 2015.06.30

명세서

청구범위

청구항 1

전해물질의 온도를 측정하고, 측정된 온도에 해당하는 제1 온도 정보신호를 출력하는 온도센서;

상기 온도 센서로부터 출력된 제1 온도 정보신호를 수신하여, 상기 제1 온도 정보신호를 온도 신호 처리 알고리즘으로 처리하여 전해물질의 온도를 결정하고, 상기 결정된 온도에 해당하는 제2 온도 정보신호를 출력하는 중앙처리부; 및

상기 중앙처리부로부터 출력된 제2 온도 정보신호를 수신하여 디스플레이하는 외부 단말

을 포함하는 도금공정 실시간 원격 모니터링 시스템.

청구항 2

제1항에 있어서,

상기 온도센서는 상기 측정된 전해물질의 온도를 보정하기 위한 온도정보처리부를 포함하는 도금공정 실시간 원격 모니터링 시스템.

청구항 3

제1항에 있어서,

상기 중앙처리부는 상기 수신된 제1 온도 정보신호와 미리 저장된 신호의 레벨을 비교하여 상기 수신된 제1 온도 정보신호의 레벨이 미리 저장된 신호의 레벨을 초과하는 경우 장애신호를 출력하는 장애신호 출력부를 포함하는 도금공정 실시간 원격 모니터링 시스템.

발명의 설명

기술 분야

[0001] 본 발명은 도금공정 실시간 원격 모니터링 시스템에 관한 것으로, 보다 상세하게는 손쉬운 사용 및 유지보수가 가능한 간편성과 비용의 최소화되며, 도금공정의 통합 운영/관리 체계를 구축하기 위한 도금공정 실시간 원격 모니터링에 관한 것이다.

배경 기술

[0002] 최근 남동공단 내의 금속가공 공장들의 화재 발생으로 인한 대형피해 사례가 지속적으로 발생되고 있는 상황이다.

[0003] 남동공단이 위치한 고잔동의 화재 원인을 살펴보면 2013-2014년도 화재발생건수 63건 중 58건이 예기치 못한 사고와 과열로 발생한 실화(失火)이다. 이는 총 화재발생건수인 63건의 92.1%를 차지한다.

[0004] 국민안전처 국가화재정보센터의 화재통계자료를 참고하면 남동공단이 위치한 인천광역시 남동구 고잔동의 화재발생빈도수가 가장 높다(연간 화재 발생건수의 약 19.5% 차지). 화재로 인한 재산피해액 또한 1,868,020천원으로 연간화재 피해액인 3,545,213천원의 약 52.6%가 공단밀집지역에서 발생함을 알 수 있다.

[0005] 도금공장의 경우 발화점이 낮은 액체들을 사용하기 때문에 낮은 온도에서도 불이 붙기 쉽고, 액체들이 많은 양을 재고로 있거나 사용 중이기 때문에 화재가 일어날 경우 그 화재규모가 크고 이에 따라 피해금액이 크다.

[0006] 도금공정의 특성상 24시간 동안 연속적인 전기화학적 처리과정을 가하기 때문에 전해물질의 이상온도로 인한 화재 위험이 항상 잠재적위험요소를 지니고 있다. 그렇기 때문에 지속적인 온도관리가 필수적인 요소이다. 그러나 중소기업의 특징인 소규모의 인력 및 자본으로는 야간근무자 배치가 어려운 상황이며, 급작스러운 이상온도 발생 시 신속한 대처가 어렵다.

- [0007] 도 1은 종래 도금업체에 화재예방시스템을 제공하는 업체가 소개하는 화재예방 시스템이다.
- [0008] 도 1을 참조하면, 도금업체 화재의 대부분이 도금액을 예열하는 과정에서 발생하는 것을 근거로 CCTV와 불꽃감지기를 통해 공장 내부의 실시간 영상 및 저장을 통해 화재 위험을 관리하고 있다.
- [0009] 이 시스템은 CCTV를 통해 실시간으로 공장 내부를 볼 수 있어 관리자가 상주하고 있다면 화재 여부를 확인할 수 있고 또한, 외부인의 침입을 확인할 수 있다.
- [0010] 그러나, 관리자가 상주해야만 하기 때문에 고정 인력이 필요하고, 외부에 있는 다른 관리자들은 별도의 연락이 있어야만 사고 소식을 접할 수 있는 문제가 있었다.

발명의 내용

해결하려는 과제

- [0011] 본 발명은 상기한 문제점을 해결하기 위해 발명된 것으로, 그 목적은 도금공정과 중○소기업의 특징을 고려하여 손쉬운 사용 및 유지보수가 가능한 간편성과 비용의 최소화를 할 수 있는 도금공정 실시간 원격 모니터링 시스템을 제공하는 것을 목적으로 한다.
- [0012] 또한, 본 발명의 다른 목적은 도금공정의 통합 운영/관리 체계를 제공하는 도금공정 실시간 원격 모니터링 시스템을 제공하는 것을 다른 목적으로 한다.
- [0013] 또한, 본 발명의 또 다른 목적은 공장 내 화재 발생 시 관리자의 빠른 인지와 대처를 통해 대형화재의 예방을 가능하게 하는 도금공정 실시간 원격 모니터링 시스템을 제공하는 것을 또 다른 목적으로 한다.

과제의 해결 수단

- [0014] 상기한 바와 같은 과제를 해결하기 위하여 본 발명은, 전해물질의 온도를 측정하고, 측정된 온도에 해당하는 제1 온도 정보신호를 출력하는 온도센서, 상기 온도 센서로부터 출력된 제1 온도 정보신호를 수신하여, 상기 제1 온도 정보신호를 온도 신호 처리 알고리즘으로 처리하여 전해물질의 온도를 결정하고, 상기 결정된 온도에 해당하는 제2 온도 정보신호를 출력하는 중앙처리부 및 상기 중앙처리부로부터 출력된 제2 온도 정보신호를 수신하여 디스플레이하는 외부 단말을 포함한다.
- [0015] 또한, 상기 온도센서는 상기 측정된 전해물질의 온도를 보정하기 위한 온도정보처리부를 포함할 수 있다.
- [0016] 또한, 상기 중앙처리부는 상기 수신된 제1 온도 정보신호와 미리 저장된 신호의 레벨을 비교하여 상기 수신된 제1 온도 정보신호의 레벨이 미리 저장된 신호의 레벨을 초과하는 경우 장애신호를 출력하는 장애신호 출력부를 포함할 수 있다.

발명의 효과

- [0017] 본 발명에 의하면, 도금공정과 중○소기업의 특징을 고려하여 손쉬운 사용 및 유지보수가 가능한 간편성과 비용의 최소화를 할 수 있는 도금공정 실시간 원격 모니터링 시스템을 제공할 수 있는 효과가 있다.
- [0018] 또한, 도금공정의 통합 운영/관리 체계를 제공하는 도금공정 실시간 원격 모니터링 시스템을 제공할 수 있는 효과가 있다.
- [0019] 또한, 공장 내 화재 발생 시 관리자의 빠른 인지와 대처를 통해 대형화재의 예방을 가능하게 하는 도금공정 실시간 원격 모니터링 시스템을 제공할 수 있는 효과가 있다.

도면의 간단한 설명

- [0020] 도 1은 종래 도금업체에 화재예방시스템을 제공하는 업체가 소개하는 화재예방 시스템이다.
- 도 2는 본 발명의 일실시예에 의한 도금공정 실시간 원격 모니터링 시스템의 구성을 나타낸 도면이다.
- 도 3은 본 발명의 일실시예에 의한 도금공정 실시간 원격 모니터링 시스템에서의 온도센서의 동작 및 온도센서의 구성을 나타낸 도면이다.
- 도 4는 본 발명의 다른 실시예에 의한 도금공정 실시간 원격 모니터링 방법을 나타낸 흐름도이다.

도 5는 Pt100용 측온저항센서 및 라즈베리파이B+를 나타낸 도면이다.

도 6은 MCP3208의 핀 정보와 외관을 나타낸 도면이다.

도 7은 라즈베리파이, Pt100용 측온저항센서 및 MCP3208을 연결한 회로도이다.

도 8은 다중 스레드 기반 Async Socket Server 개념도이다.

도 9는 Push Service의 개념도이다.

도 10은 Push Alarm을 수신한 모습을 나타낸 도면이다.

도 11은 본 발명의 일 실시예가 안드로이드 어플리케이션에서 실행된 모습을 나타낸 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0021] 본 발명을 충분히 이해하기 위해서 본 발명의 바람직한 실시예를 첨부 도면을 참조하여 설명한다. 본 발명의 실시예는 여러 가지 형태로 변형될 수 있으며, 본 발명의 범위가 아래에서 상세히 설명하는 실시예로 한정되는 것으로 해석되어서는 안 된다. 본 실시예는 당업계에서 평균적인 지식을 가진 자에게 본 발명을 보다 완전하게 설명하기 위하여 제공 되는 것이다. 따라서 도면에서의 요소의 형상 등은 보다 명확한 설명을 강조하기 위해서 과장되어 표현될 수 있다. 각 도면에서 동일한 부재는 동일한 참조부호로 도시한 경우가 있음을 유의하여야 한다. 또한, 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 공지 기능 및 구성에 대한 상세한 기술은 생략된다.
- [0022] 이하, 첨부된 도면을 참조하여 본 발명에 의한 실시예를 상세하게 설명한다.
- [0023] 도 2는 본 발명의 일실시예에 의한 도금공정 실시간 원격 모니터링 시스템의 구성을 나타낸 도면이고, 도 3은 본 발명의 일실시예에 의한 도금공정 실시간 원격 모니터링 시스템에서의 온도센서의 동작 및 온도센서의 구성을 나타낸 도면이다.
- [0024] 도 2를 참조하면, 본 발명의 일실시예에 의한 도금공정 실시간 원격 모니터링 시스템은 온도센서, 중앙처리부 및 외부 단말을 포함한다.
- [0025] 온도센서는 전해물질의 측정 및 온도 상승여부를 감지하고, 중앙처리부 및 외부 단말로 상기 측정된 온도 및 온도 상승여부에 해당하는 제1 온도 정보신호를 출력한다.
- [0026] 중앙처리부는 상기 온도센서로부터 출력된 제1 온도 정보신호를 수신하여 저장하고, 상기 제1 온도 정보신호를 온도 신호 처리 알고리즘으로 처리하여 전해물질의 온도를 결정하고, 상기 결정된 온도에 해당하는 제2 온도 정보신호를 출력한다.
- [0027] 또한, 중앙처리부는 상기 수신된 제1 온도 정보신호와 미리 저장된 신호의 레벨을 비교하여 상기 수신된 제1 온도 정보신호의 레벨이 미리 저장된 신호의 레벨을 초과하는 경우 장애신호를 출력하는 장애신호 출력부를 포함한다. 신호의 레벨을 비교할 때, 신호 자체의 레벨을 비교할 수도 있고, 수신된 신호를 숫자로 변환하여 변환된 숫자를 비교하는 것도 가능하다.
- [0028] 외부단말은 상기 중앙처리부로부터 출력된 제2 온도 정보신호를 수신하여 디스플레이한다.
- [0029] 도 3을 참조하면, 온도센서는 통신모듈, 온도정보처리부, 배터리, 수온센서 및 디스플레이부를 포함한다.
- [0030] 통신모듈은 상기 중앙처리부와 TCP/IP Socket통신로 통신을 하며, 도 3에는본 발명의 일실시예인 CDMA를 이용하여 무선통신을 하는 것을 나타내고 있다. 그러나 무선통신 방식은 CDMA이외 한정되는 것은 아니며 비용을 절약하기 위하여 WiFi를 이용하는 것도 가능하다. 즉, 무선통신 방식에 구애되는 것이 아니며 상기 중앙처리부와 무선통신이 가능하다면 어떠한 무선통신 방식도 가능하다.
- [0031] TCP/IP란 Transfer Control Protocol / Internet Protocol 의 약어으로써 컴퓨터간 통신에서 데이터를 전송할 때 예기치 않은 에러에 의해 데이터 손실이 일어나지 않도록 특정한 크기로 나누어 전송하고 이를 수신할 때 다시 원래의 정보로 변환하는 것을 약속해 놓은 통신 규약이다.
- [0032] 이 때, 특정한 크기로 나누어진 단위데이터를 패킷이라 칭하고 패킷의 구성은 헤더부분과 바디부분으로 나뉜다. 헤더에는 패킷의 정보(패킷 사이즈, 인코딩 방식 등)가 저장되어있고 바디에는 전송하고자 하는 데이터가 저장되어있다. 본 발명의 일실시예에서는 패킷의 크기를 4096Bytes로 설정하였다.

- [0033] 온도정보처리부는 이후 설명될 수온센서에서 측정된 전해물질의 아날로그 온도 신호를 디지털로 변환하며, 상기 측정된 전해물질의 온도를 처리하여 상기 측정된 전해물질의 온도를 보정하고 상기 중앙처리부와 무선통신을 제어하며 이후 설명될 디스플레이부에 정보를 표시하는 것을 관리한다.
- [0034] 배터리는 상기 온도센서에 공급되는 전원이 고장난 경우 상기 온도센서에 전원을 공급한다.
- [0035] 수온센서는 전해물질의 온도 측정 및 온도 상승 여부를 감지한다. 도금공정에서 일반적으로 사용되는 온도는 30도에서 70도 사이의 온도가 사용되므로 이를 충분히 포함할 수 있어야 한다. 예를 들면 수온센서는 0도에서 300도 사이에서 동작되는 수온센서가 사용될 수 있다.
- [0036] 디스플레이부는 상기 온도정보처리부에서 보정된 온도 및 상기 온도정보처리부에서 제공하는 정보를 표시한다.
- [0037] 도 3을 참조하여 온도센서의 동작을 검토하면, 온도센서가 전해물질의 온도를 측정하고, 측정된 온도 및 온도 상승여부를 상기 외부 단말 및 상기 중앙처리부로 송신한다. 상기 중앙처리부는 전해물질의 측정 온도에 해당하는 제1 온도 정보신호를 수신하고, 수신된 제1 온도 정보신호와 미리 저장된 신호의 레벨을 비교하여 상기 수신된 제1 온도 정보신호의 레벨이 미리 저장된 신호의 레벨을 초과하는 경우 장애신호를 상기 외부 단말로 SMS 문자전송 형태로 출력한다. 또한 상기 외부 단말은 상기 온도센서가 출력한 제1 온도정보신호를 수신하여 디스플레이한다.
- [0038] 도 4는 본 발명의 다른 실시예에 의한 도금공정 실시간 원격 모니터링 방법을 나타낸 흐름도이다.
- [0039] 도 4를 참조하면, 온도센서가 온도를 측정하고, 측정된 온도를 계산하고 계산된 데이터를 저장하고, 중앙처리부의 일실시예인 서버와 연결을 시도한다. 서버와 연결되면 패킷을 나누어서 데이터 베이스에 Insert한다.
- [0040] 본 발명의 일실시예에서 데이터의 저장 및 관리를 위한 데이터베이스는 MYSQL을 이용하고 모바일 운영체제인 안드로이드(Android)와의 통신을 위한 아파치 웹서버(Apache Webserver)와 연결한다.
- [0041] 안드로이드(Android)는 MYSQL에 직접접근이 지원되지 않으므로 웹 프로그래밍 언어인 PHP로 데이터를 아파치 웹서버(Apache Webserver)에 게시 후 데이터 표현방식인 JSON(JavaScript Standard Object Notation)을 이용해 안드로이드(Android)에서 데이터를 수신해 표현한다.
- [0042] 설정한 관리 상/하한을 벗어나게 되면 사용자의 스마트폰으로 Push Service를 실시간으로 보내 지속적 관리를 할 수 있게 한다.
- [0043] 도 5는 Pt100옴 측온저항센서 및 라즈베리파이B+를 나타낸 도면이고, 도 6은 MCP3208의 핀 정보와 외관을 나타낸 도면이다.
- [0044] 라즈베리파이(RaspberryPi B+)는 크기 85.60mm x 53.98mm의 소형 컴퓨터로서 데스크탑이나 랩탑에 비해 공간을 적게 차지한다. 따라서 진행하고자 하는 도금공정 상황 하에서 사용하기 적합하고 가격 또한 3달러로 매우 저렴한 편이다.
- [0045] 라즈베리파이(RaspberryPi B+)의 메모리(RAM)는 512MB로 Pt100옴 측온저항센서의 측정값을 처리하는 프로그램과 서버 컴퓨터와의 TCP/IP Socket통신을 하는 Client 프로그램을 구동하기에 충분한 성능을 지닌다. 4개의 USB 포트와 HDMI 연결이 지원되기 때문에 유지보수 및 개발에도 용이하다.
- [0046] 라즈베리파이(RaspberryPi B+)에는 하드웨어와 연결해 제어할 수 있는 GPIO(General Purpose Input Output) 인터페이스가 존재한다. 이를 이용해 Pt100옴측온저항센서와 연결하여 측정값을 받아들일 수 있다. 또한, 라즈베리파이의 운영체제인 라즈비안(Raspbian)에는 프로그래밍 언어인 Python 3.x 버전을 탑재하고 있어 Pt100옴 측온저항센서의 측정값을 처리하는 프로그램과 서버 컴퓨터와의 TCP/IP Socket통신이 가능한 프로그램을 제작할 수 있다.
- [0047] Pt100옴 측온저항센서의 외피는 불소수지 (PTFE)로 이루어져있어 내구성이 뛰어나고 도금 상황(액체)에서 사용하기 적합하다. 측정온도는 0-300도씨(이하 도 라 한다)까지 측정이 가능하므로 도금공정에서 일반적으로 다루는 온도(30~70도)를 충분히 포함하기 때문에 적합하다 판단되었다.
- [0048] Pt100옴 측온저항센서에서 측정되는 저항값은 아날로그 신호이고 라즈베리파이는 컴퓨터이기 때문에 디지털 신호를 사용한다. 따라서 라즈베리파이의 GPIO와 저항센서간의 데이터 송,수신을 위해선 A/D Converting(아날로그 -> 디지털 신호변환) 과정이 필요하다. 이를 위해서 도 7의 MCP3208 ADC(Analog to Digital Converter)를 이용한다.

[0049] 도 7은 라즈베리파이, Pt100옴 측온저항센서 및 MCP3208을 연결한 회로도이다.

[0050] 라즈베리파이 프로그램 상에서 온도계측을 시작하면 Pt100옴 측온저항센서에서 계측된 저항 값(Analog)이 MCP3208을 통해 변환된다. 이 때, MCP3208은 [수학식 1]의 방정식에 따라 Digital Output Code를 프로그램 상으로 전달하는데, 이 Digital Output Code를 이용하여 V_{IN} 을 계산하는 방정식을 프로그램상에 저장한다.

[0051] [수학식 1]

$$\text{Digital Output Code} = \frac{4096 \times V_{IN}}{V_{REF}}$$

$$V_{IN} = \text{analog input voltage}$$

$$V_{REF} = \text{reference voltage}$$

[0052]

[0053] V_{IN} 의 계산이 완료되면 온도 계산에 필요한 저항값을 V_{IN} 을 이용해 계산한다. 저항값(R_1)을 계산하기 위한 수식은 [수학식 2]와 같다.

[0054] [수학식 2]

$$R_1 = \frac{100(3.3 - V_{IN})}{V_{IN}} + R_a$$

[0055]

[0056] 여기서 상수 100은 Pt100옴 측온저항센서의 자체저항이며 3.3은 라즈베리파이 GPIO핀의 출력 전압 값이다.

[0057] 저항값(R_1)의 계산이 완료되면 [수학식 3]을 따라 온도를 계산한다.

[0058] [수학식 3]

$$Temp = \frac{(R_1 - 100)}{0.39} + T_a$$

[0059]

[0060] Pt100옴 측온저항센서는 IEC751 규격을 따르는데, 이 규격은 저항소자의 온도가 0도 일 때 저항값(R_0)이 100옴이고 저항소자의 온도가 100도 일 때 저항값(R_{100})은 139옴으로 나타난다. 이를 따르면 온도가 1도 상승할 때 저항은 약 0.39옴 상승하는 것을 알 수 있으며 이를 이용해 온도를 계산한다.

[0061] 저항을 계측할 때, 계측 시스템이 처한 환경 (분진, 이물질 등)에 의해 영향을 받아 실제 저항값과 계측된 저항값 사이에 오차가 발생할 수 있다. 이를 보정하기 위해 [수학식 2]에서 저항보정상수 R_a 를 사용하여 온도계산에 필요한 저항값을 산출한다.

[0062] 산출되어진 저항값을 이용해 최종적인 온도를 계산할 때, Pt100옴 측온저항센서의 상태와 리드선의 길이에 따라 계산되는 온도값에 편차가 발생할 수 있다. 이를 보정하기 위해 [수학식 3]에서 온도보정상수 T_a 를 사용하여 온도를 계산한다.

R_{α} 와 T_{α} 는 각 기기와 각 Pt100용 측온저항센서, 측정되는 온도대, 그리고 측정환경별로 다를 수 있어 이를 결정하는 실험이 필요하다.

R_{α} 및 T_{α} 는 결정실험

측정오차의 발생원인은 기기나 센서의 회로상태, 측정 환경, 온도대 등 다양하다. 이를 모두 고려할 수 있도록 측정시의 환경을 동일하게 설정하고 온도를 변화하며 실험할 수 있게 설정한 후 0도, -10도, 10도, -20도, 20도, -30도, 30도, -40도, 40도, -50도, 50도, -60도, 60도, -70도, 70도, -80도와 같이 구간별로 온도를 측정한다. 그리고 [테이블 1]에 주어진 특정 온도에서의 이상적인 저항값과 비교해 R_{α} 와 T_{α} 를 산출한다.

[테이블 1]

Temp	0	1	2	3	4	5	6	7	8	9
-20	92.16	91.77	91.37	90.98	90.59	90.19	89.8	89.4	89.01	88.62
-10	96.09	95.69	95.3	94.91	94.52	94.12	93.73	93.34	92.95	92.55
0	100	99.61	99.22	98.83	98.44	98.04	97.65	97.26	96.87	96.48
0	100	100.39	100.78	101.17	101.56	101.95	102.34	102.73	103.12	103.51
10	103.9	104.29	104.68	105.07	105.46	105.85	106.24	106.63	107.02	107.4
20	107.79	108.18	108.57	108.96	109.34	109.73	110.12	110.51	110.9	111.28
30	111.67	112.06	112.45	112.83	113.22	113.61	113.99	114.38	114.77	115.15
40	115.54	115.92	116.31	116.7	117.08	117.47	117.85	118.24	118.62	119.01
50	119.4	119.78	120.16	120.55	120.93	121.32	121.7	122.09	122.47	122.86
60	123.24	123.62	124.01	124.39	124.77	125.16	125.54	125.92	126.31	126.69
70	127.07	127.45	127.84	128.22	128.6	128.98	129.36	129.75	130.13	130.51
80	130.89	131.27	131.66	132.04	132.42	132.8	133.18	133.56	133.94	134.32
90	134.7	135.08	135.46	135.84	136.22	136.6	136.98	137.36	137.74	138.12
100	138.5	138.88	139.26	139.64	140.02	140.4	140.77	141.15	141.53	141.91

실험 결과

[테이블 2]

T01(라즈베리파이 1번)		
온도대	R_{α}	T_{α}
10 °C	-4.5	-2.0
20 °C	-4.0	-2.0
30 °C	-4.5	-2.0
40 °C	-4.2	-2.0
50 °C	-4.1	-2.0
60 °C	-4.8	-2.0
70 °C	-5.7	-2.0
80 °C	-5.9	-2.0

[0071] [테이블 3]

T02(라즈베리파이 2번)		
온도대	R_{α}	T_{α}
10 °C	-3.3	-2.0
20 °C	-3.5	-2.0
30 °C	-3.3	-2.0
40 °C	-3.3	-2.0
50 °C	-3.3	-2.0
60 °C	-3.3	-2.0
70 °C	-4.4	-2.0
80 °C	-4.8	-2.0

[0072]

[0073] 결과에 따라 각 온도대별 보정상수를 결정하여 프로그램상에 저장한다.

[0074] 저장 후 라즈베리파이는 PC 서버와의 통신을 시작한다.

[0075] 라즈베리파이 - PC 간 서버 통신

[0076] 라즈베리파이와 PC간의 통신을 위한 프로토콜(Protocol)은 TCP/IP Socket 통신을 사용한다. 서버의 운영체제는 (Operating System) Windows7이며 서버프로그램 개발 언어로는 C#을 사용한다.

[0077] TCP/IP란 Transfer Control Protocol / Internet Protocol 의 약으로써 컴퓨터간 통신에서 데이터를 전송할 때 예기치 않은 에러에 의해 데이터 손실이 일어나지 않도록 특정한 크기로 나누어 전송하고 이를 수신할 때 다시 원래의 정보로 변환하는 것을 약속해 놓은 통신 규약이다.

[0078] 이 때, 특정한 크기로 나누어진 단위데이터를 패킷이라 칭하고 패킷의 구성은 헤더부분과 바디부분으로 나뉜다. 헤더에는 패킷의 정보(패킷 사이즈, 인코딩 방식 등)가 저장되어있고 바디에는 전송하고자 하는 데이터가 저장되어있다. 이 시스템에서 다루는 데이터의 크기는 큰 용량이 필요하지 않아 패킷의 크기를 4096Bytes로 설정하였다.

[0079] 라즈베리파이에서 서버로 전송하는 데이터는 각 라즈베리파이에 부여한 명칭(예 : T01, T02...)과 온도가 측정된 시간, 그리고 온도 데이터를 각각 문자열(String), 문자열(String), 실수(Float) 형태로 서버 컴퓨터에 전송한다.

[0080] 라즈베리파이에서 데이터 송신을 하면 PC 서버에 저장된 서버 프로그램 내의 DoParseHeader 메서드와 DoParseBody 메서드에서 패킷의 헤더와 바디 부분을 수신하여 데이터베이스에 저장된다.

[0081] 도 8은 다중 스레드 기반 Async Socket Server 개념도이다.

[0082] 서버는 다수의 라즈베리파이에서 보내오는 데이터를 수신하기 위해 다중 스레드 기반의 Async Socket Server로 개발하였다.

[0083] 스레드란 하나의 프로그램 내에서 둘 이상의 작업이 필요할 때 사용하는 개념이다. 실시간으로 데이터를 수신하기 위해서는 라즈베리파이의 개수만큼 스레드가 열려있어야 하므로 다중 스레드 기반의 서버를 개발하였다.

[0084] Async Socket Server의 Async는 AsynchroNous의 약자으로써 비동기식 전송을 뜻한다. 비동기식 전송의 특징은 전송 한 번에 보낼 수 있는 데이터의 개수에 제한이 없고 기기 간의 전송 속도를 맞출 필요가 없다.

[0085] 서버에서 데이터를 수신할 때 온도를 판단해 관리 상한을 벗어나게 되면 GCM(Google Cloud Message)서버와 연동해 관리자의 스마트폰으로 Push Message를 보내게 된다.

[0086] PC 서버 - 스마트폰 간 통신

[0087] 도 9는 Push Service의 개념도이다.

[0088] Push Message를 보내려면 먼저 GCM서버와 통신을 해야한다. GCM 서버에서는 관리자의 스마트폰을 등록하고 스마트폰의 Device Token을 전달한다. 이 Device Token을 PC서버로 전송해 PC서버상에 저장하고 온도가 관리 상한을 벗어날 때 다시 GCM서버로 Push Message 요청을 보낸다. 마지막으로 GCM 서버에서 스마트폰으로 Push Alarm을

전송하면 개발한 안드로이드 어플리케이션 내부에서 미리 정의되어진 문구를 Push Alarm에 표시한다.

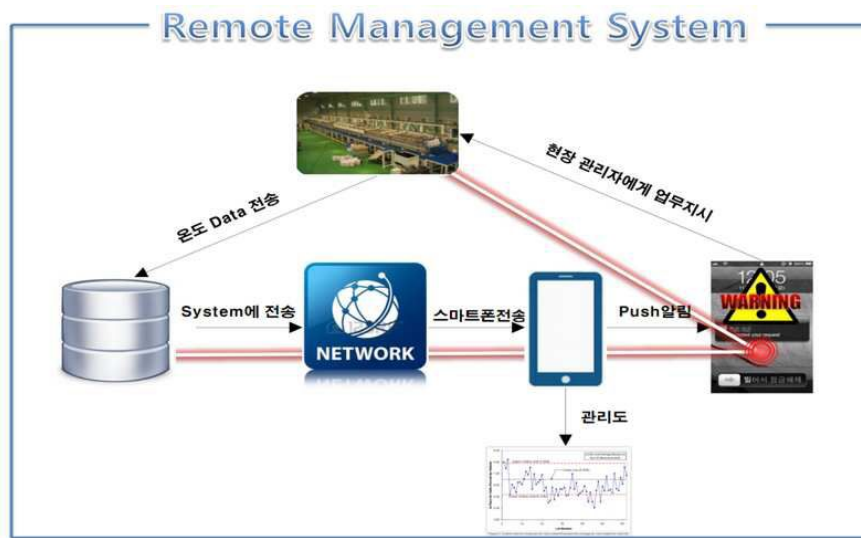
- [0089] 위 절차를 진행하여 어플리케이션에 적용시키면 온도가 관리상한을 벗어날 시 도 9와 같이 Push Alarm을 받을 수 있다.
- [0090] 도 10은 Push Alarm을 수신한 모습을 나타낸 도면이다.
- [0091] 스마트폰에서 데이터베이스에 저장되어진 온도 데이터를 조회하기 위해서는 Apache Web Server를 거쳐 JSON Parsing 과정이 필요하다.
- [0092] Apache Web Server는 웹서버용 소프트웨어로써 오픈소스 라이선스를 따르고 있어 무료로 사용이 가능하다. 데이터베이스에서 Apache Web Server를 통해 스마트폰으로 데이터를 전송한다. 그러나 스마트폰에선 데이터베이스에 직접 접근이 지원되지 않는다. 그러므로 JSON Parsing 과정을 거쳐 스마트폰에서 정상적으로 데이터를 조회할 수 있게 한다.
- [0093] JSON Parsing이란 Row/Column 형태로 이루어진 데이터베이스의 테이블을 {,}와 같은 기호로 구분되어진 JSON이란 데이터 형태로 변환하는 과정을 말한다. JSON과 마찬가지로 안드로이드도 JAVA 기반 개발플랫폼이기 때문에 서로간의 호환성이 우수하다.
- [0094] 도 11은 본 발명의 일 실시예가 안드로이드 어플리케이션에서 실행된 모습을 나타낸 도면이다.
- [0095] JSON Parsing을 거쳐 안드로이드 어플리케이션에 정상적으로 데이터를 전송하면 도 11과 같이 어플리케이션 상에서 온도 데이터를 조회할 수 있다.
- [0096] 이상에서 본 발명에 의한 바람직한 실시예를 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 청구의 범위에 기재된 본 발명의 사상 및 영역의 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

도면

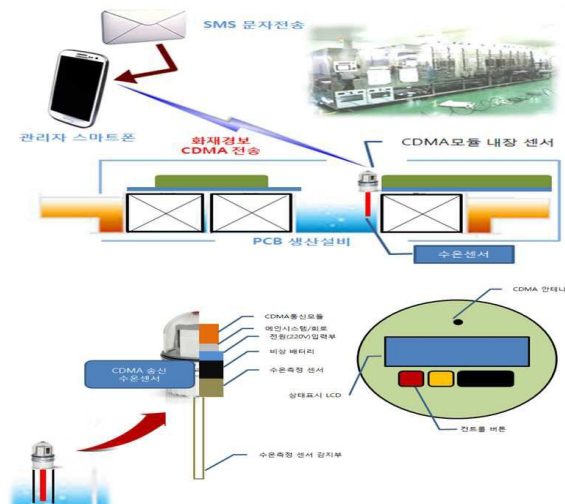
도면1



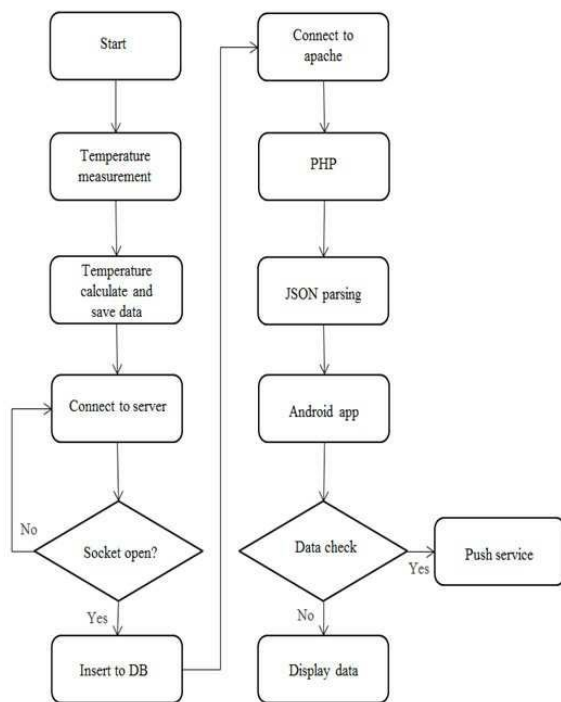
도면2



도면3



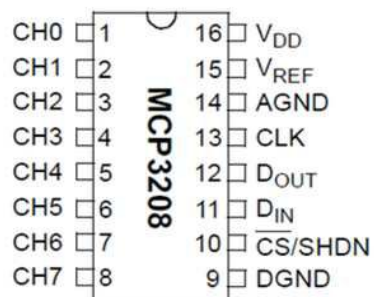
도면4



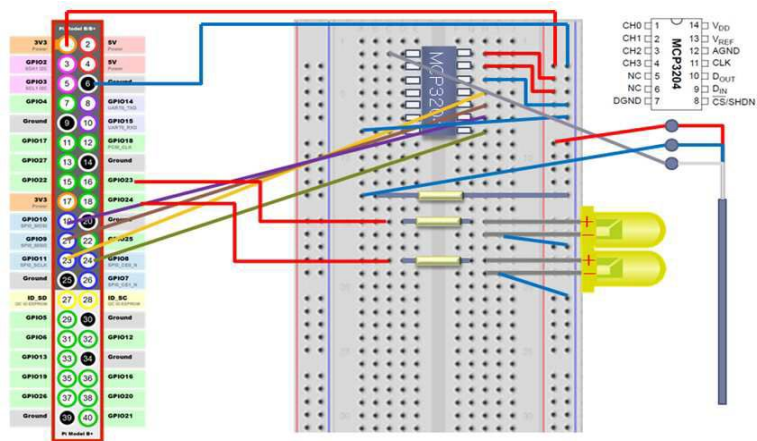
도면5



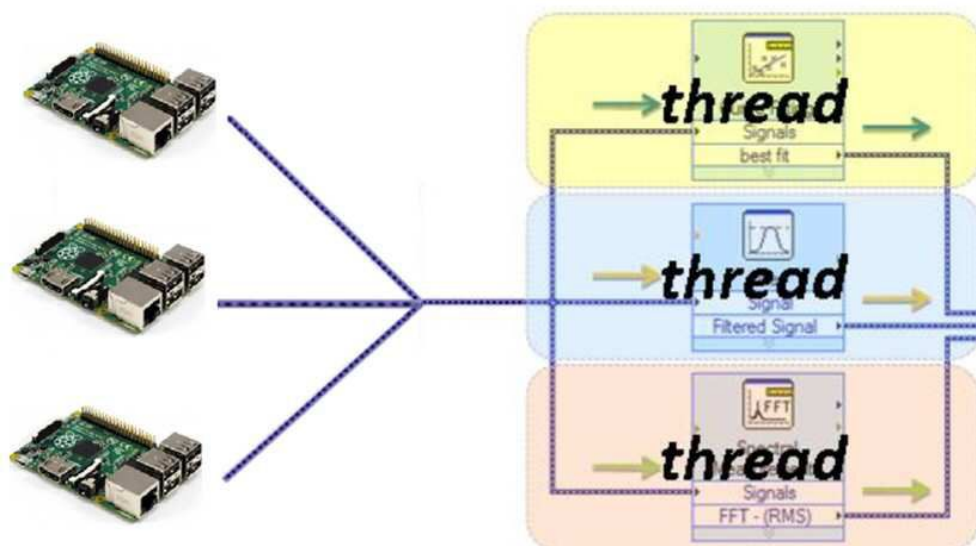
도면6



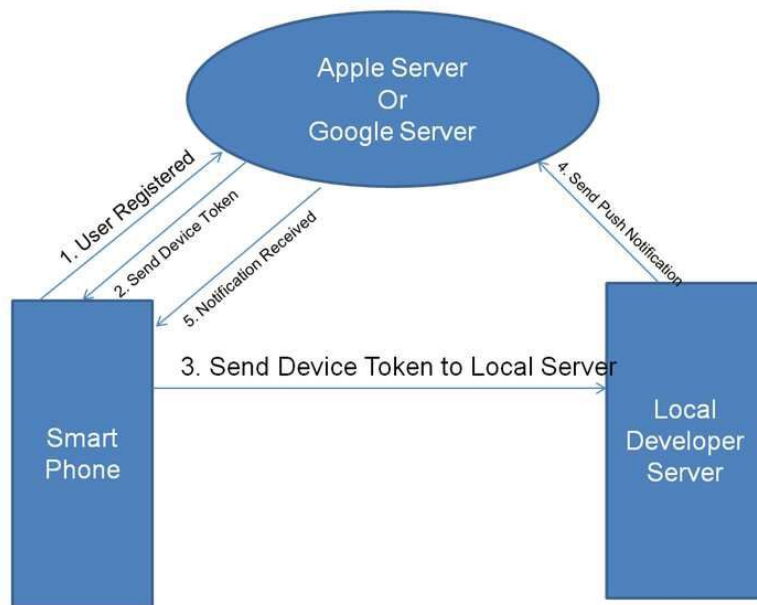
도면7



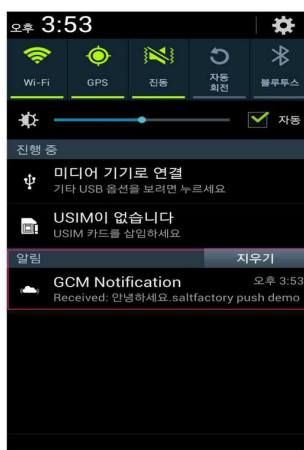
도면8



도면9



도면10



도면11

