



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0089233
(43) 공개일자 2018년08월08일

(51) 국제특허분류(Int. Cl.)
H04L 29/08 (2006.01) G06Q 50/10 (2012.01)
G08C 17/02 (2006.01)
(52) CPC특허분류
H04L 67/12 (2013.01)
G06Q 50/10 (2015.01)
(21) 출원번호 10-2017-0014090
(22) 출원일자 2017년01월31일
심사청구일자 2017년01월31일

(71) 출원인
계명대학교 산학협력단
대구광역시 달서구 달구벌대로 1095, 계명대학교
산학협력관 201호(신당동)
(72) 발명자
방대욱
대구광역시 달서구 와룡로 169 101-1104 (감삼동,
월드마크웨스트엔드)
김경진
대구광역시 달서구 한실로 136, 212동 204호 (도
원동, 미리샘마을아파트)
(74) 대리인
특허법인아주

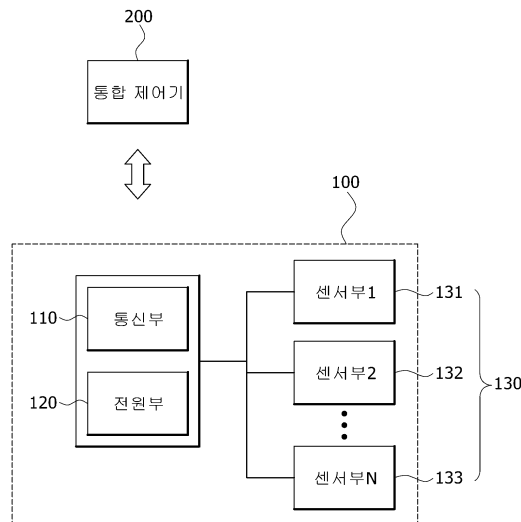
전체 청구항 수 : 총 9 항

(54) 발명의 명칭 스마트 관제 시스템을 위한 센서 디바이스 모듈

(57) 요약

본 발명은 스마트 관제 시스템을 위한 센서 디바이스 모듈에 관한 것으로, 관제 대상의 상태나 혹은 관제 대상 주변의 상태 정보를 적어도 하나 이상 측정하는 센서부, 센서 디바이스 모듈의 상위 레벨 제어기인 통합 제어기와, 유선 기반의 통신과 무선 기반의 통신 중 적어도 하나를 통해 네트워크를 구현하는 통신부, 및 상기 통신부 및 센서부에 구동 전원을 공급하는 전원부를 포함하며, 상기 통신부가 상기 센서부를 통해 측정된 센싱 정보를 상위 레벨 제어기인 상기 통합 제어기에 전달하되, 상기 통합 제어기의 지시나 요청에 따라 전달하거나, 미리 지정된 특정 이벤트가 발생할 경우에 통신부가 자체적으로 전달하도록 구현된다.

대 표 도 - 도1



(52) CPC특허분류

G08C 17/02 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 2016-D-0041-010100

부처명 교육부

연구관리전문기관 한국연구재단

연구사업명 산학협력 선도대학(LINC) 육성사업

연구과제명 산학협력 선도대학(LINC) 육성사업

기 여 율 1/1

주관기관 계명대학교

연구기간 2016.03.01 ~ 2017.02.28

명세서

청구범위

청구항 1

관제 대상의 상태나 혹은 관제 대상 주변의 상태 정보를 적어도 하나 이상 측정하는 센서부;

센서 디바이스 모듈의 상위 레벨 제어기인 통합 제어기와, 유선 기반의 통신과 무선 기반의 통신 중 적어도 하나를 통해 네트워크를 구현하는 통신부; 및

상기 통신부 및 센서부에 구동 전원을 공급하는 전원부;를 포함하며,

상기 통신부가 상기 센서부를 통해 측정된 센싱 정보를 상위 레벨 제어기인 상기 통합 제어기에 전달하되, 상기 통합 제어기의 지시나 요청에 따라 전달하거나, 미리 지정된 특정 이벤트가 발생할 경우에 통신부가 자체적으로 전달하도록 구현되는 것을 특징으로 하는 스마트 관제 시스템을 위한 센서 디바이스 모듈.

청구항 2

제 1항에 있어서, 상기 센서부는,

센서 보드에 일체형으로 탑재되어 구현되거나, 또는

수동 소자 및 능동 소자를 포함한 전자 소자와 함께 분리형으로 구성된 후 연결 소자를 이용해 상기 센서 디바이스 모듈에 연결되게 구현되는 것을 특징으로 하는 스마트 관제 시스템을 위한 센서 디바이스 모듈.

청구항 3

제 1항에 있어서,

상기 센서 디바이스 모듈, 및 이에 연결되는 센서부의 적어도 하나 이상의 센서는 각기 미리 지정된 비트 수의 아이디(ID)가 설정되는 것을 특징으로 하는 스마트 관제 시스템을 위한 센서 디바이스 모듈.

청구항 4

제 1항에 있어서, 상기 센서 디바이스 모듈의 통신부나 내부 프로세서는,

상기 통합 제어기의 지시나 요청 메시지에 의한 초기화,

상기 통합 제어기의 주기적인 폴링에 의한 센싱 정보의 전송, 및

상기 센서 디바이스 모듈의 이벤트 발생에 의한 비동기적인 센싱 정보의 전송을 수행하는 프로토콜이 구현되는 것을 특징으로 하는 스마트 관제 시스템을 위한 센서 디바이스 모듈.

청구항 5

제 4항에 있어서, 상기 센서 디바이스 모듈의 통신부나 내부 프로세서는,

상기 통합 제어기로부터 셋업 초기화(SETUP_INIT) 메시지를 수신하면, 센서 디바이스 모듈 및 적어도 하나 이상의 센서의 초기화 동작을 수행하고, 아울러 상기 통합 제어기에 적어도 하나 이상의 지정된 응답 메시지(RESPONSE_INIT, RESPONSE_STATUS, RESPONSE_DATA_INIT)를 출력하며,

상기 통합 제어기로부터 상기 센서 디바이스 모듈이나 센서에 대한 리셋 메시지(SETUP_INIT_SNODE, SETUP_INIT_SENSOR)를 수신하면, 센서 디바이스 모듈이나 센서를 리셋한 후 리셋 완료 응답(RESPONSE_RESET)을

상기 통합 제어기에 출력하도록 프로토콜이 구현되는 것을 특징으로 하는 스마트 관제 시스템을 위한 센서 디바이스 모듈.

청구항 6

제 5항에 있어서,

상기 셋업 초기화(SETUP_INIT) 메시지는,

통합 제어기 번호(cgno), 온실 내 위치(Location), 모니터 유형(monitorType), 유형별 설정값(monitor value), 임계유형(monitorEvent), 및 임계값(monitor Limit) 정보 중 적어도 하나 이상을 포함하고,

상기 센서 디바이스 모듈의 센서 초기화 동작은,

센서 디바이스 모듈 구성정보 설정, 센서 디바이스 모듈 상태정보 설정, 센서 상태정보 설정, 및 센서 HW 초기화 중 적어도 하나 이상을 포함하는 것을 특징으로 하는 스마트 관제 시스템을 위한 센서 디바이스 모듈.

청구항 7

제 4항에 있어서, 상기 센서 디바이스 모듈의 통신부나 내부 프로세서는,

상기 통합 제어기로부터 상태정보 요청 메시지(REQUEST_SNODE_INFO)를 수신하면, 상기 요청에 대응하는 상태정보가 포함된 응답 메시지(RESPONSE_STATUS)를 생성하여 출력하며,

상기 통합 제어기로부터 센서정보 요청 메시지(REQUEST_SNODE_VALUE)를 수신하면, 센서 읽기를 한 후에 응답 메시지(RESPONSE_DATA_PASSIVE)를 상기 통합 제어기에 출력하고, 이 과정에서 센서에 오류가 발생하면 에러 알람 메시지(NOTIFY_ERROR)를 상기 통합 제어기에 출력하고,

상기 통합 제어기로부터 상기 센서 디바이스 모듈에 대한 모니터링 유형설정 요청 메시지(SETUP_SNMODE)나 센서에 대한 모니터링 유형설정 요청 메시지(SETUP_SMODE)를 수신하면, 상기 각 모니터링 유형설정 요청 메시지(SETUP_SNMODE, SETUP_SMODE)에 따라 모니터링 유형을 설정한 후에 상기 통합 제어기에 응답 메시지(RESPONSE_SETUP)를 각기 출력하도록 프로토콜이 구현되는 것을 특징으로 하는 스마트 관제 시스템을 위한 센서 디바이스 모듈.

청구항 8

제 7항에 있어서,

상기 응답 메시지(RESPONSE_STATUS)는 센서 디바이스 모듈 ID(snid), 센서 디바이스 모듈 상태(status), 초기화 상태(isInit), 모니터 유형(monitorType), 유형별 설정값(monitorValue), 소프트웨어 버전(swVer), 프로파일 버전(profileVer), 통신오류 카운트(commErrorNuma), 서비스 오류 카운트(serviceErrorNum), 센서 ID(sid), 센서 상태(state), 임계유형(monitorevent)(임계기준), 및 임계치(monitorlimit) 중 적어도 하나 이상을 포함하는 것을 특징으로 하는 스마트 관제 시스템을 위한 센서 디바이스 모듈.

청구항 9

제 4항에 있어서, 상기 센서 디바이스 모듈의 통신부나 내부 프로세서는,

센서 디바이스 모듈이 최초 구동되면서 상기 통합 제어기에 연결 요청 메시지(REQUEST_CONNECTION)를 보내 통합 제어기에 연결하기 위한 연결정보를 받고,

실시간으로 측정된 센싱 정보를 알람 메시지(NOTIFY_DATA_ACTIVE)를 통해 상기 통합 제어기에 출력하며, 또한

이벤트 발생, 통신 상태, 및 오류 발생을 각기 알리는 메시지(NOTIFY_DATA_EVENT, NOTIFY_ALIVE, NOTIFY_ERROR)를 상기 통합 제어기에 자체적으로 출력하도록 프로토콜이 구현되는 것을 특징으로 하는 스마트

관제 시스템을 위한 센서 디바이스 모듈.

발명의 설명

기술 분야

- [0001] 본 발명은 스마트 관제 시스템을 위한 센서 디바이스 모듈에 관한 것으로, 보다 상세하게는 사물 인터넷(IoT : Internet of Things) 기반 원격 모니터링 및 관리 시스템의 구축 시 소규모 IoT 플랫폼에 해당하는 스마트 관제 시스템(또는 스마트 관제 플랫폼)에 적용할 수 있는 센서 디바이스 모듈에 관한 것이다.

배경 기술

- [0002] 최근 IoT(Internet of Things) 기술과의 융합을 통한 스마트 온실 혹은 스마트 팜과 같은 새로운 첨단 재배 시스템으로의 도약을 위해 스마트 관제(예 : 감시 제어, 장애 탐지 및 복구 등) 시스템(또는 스마트 관제 플랫폼)이 연구되고 있다.
- [0003] 이러한 스마트 관제 시스템의 가장 말단에 위치하고 있으면서 관제 대상이나 관제 대상 주변의 상황을 검출하는 중요한 역할을 수행하는 것이 센서이다.
- [0004] 통상적인 스마트 관제 시스템의 설계자는 관제할 대상(또는 시스템의 목적)에 따라 관제 대상을 제어하는 방법에 치중하여 시스템을 설계하고 있으나, 실제로는 센서의 특성에 따라 검출 정보가 달라질 수도 있고, 또한 센서에서 검출된 정보가 정확히 전달되지 못할 수도 있기 때문에 센서의 출력을 안정화시키기 위한(센서가 안정적인 결과를 검출하고, 또한 검출된 결과를 안정적으로 전달하기 위한) 부분에서 많은 시간적 투자 및 비용적 투자가 이루어지고 있다.
- [0005] 이에 따라 소규모 IoT 플랫폼에 해당하는 스마트 관제 시스템(또는 스마트 관제 플랫폼)의 설계 시 사용자의 시간적 부담 및 비용적 부담이 증가하는 문제점이 있으며, 따라서 스마트 관제 시스템(또는 스마트 관제 플랫폼)에 곧바로 적용할 수 있는 표준화된 인터페이스를 갖는 센서 디바이스 모듈이 필요한 상황이다.
- [0006] 본 발명의 배경기술은 대한민국 등록특허 10-1202614호(2012.11.13. 등록, 환경 모니터링용 스마트센서모듈)에 개시되어 있다.

발명의 내용

해결하려는 과제

- [0007] 본 발명의 일 측면에 따르면, 본 발명은 상기와 같은 문제점을 해결하기 위해 창작된 것으로서, 사물 인터넷(IoT : Internet of Things) 기반 원격 모니터링 및 관리 시스템의 구축 시 소규모 IoT 플랫폼에 해당하는 스마트 관제 시스템(또는 스마트 관제 플랫폼)에 적용할 수 있는 스마트 관제 시스템을 위한 센서 디바이스 모듈을 제공하는데 그 목적이 있다.

과제의 해결 수단

- [0008] 본 발명의 일 측면에 따른 스마트 관제 시스템을 위한 센서 디바이스 모듈은, 관제 대상의 상태나 혹은 관제 대상 주변의 상태 정보를 적어도 하나 이상 측정하는 센서부; 센서 디바이스 모듈의 상위 레벨 제어기인 통합 제어기와, 유선 기반의 통신과 무선 기반의 통신 중 적어도 하나를 통해 네트워크를 구현하는 통신부; 및 상기 통신부 및 센서부에 구동 전원을 공급하는 전원부;를 포함하며, 상기 통신부가 상기 센서부를 통해 측정된 센싱 정보를 상위 레벨 제어기인 상기 통합 제어기에 전달하되, 상기 통합 제어기의 지시나 요청에 따라 전달하거나, 미리 지정된 특정 이벤트가 발생할 경우에 통신부가 자체적으로 전달하도록 구현되는 것을 특징으로 한다.
- [0009] 본 발명에 있어서, 상기 센서부는, 센서 보드에 일체형으로 탑재되어 구현되거나, 또는 수동 소자 및 능동 소자를 포함한 전자 소자와 함께 분리형으로 구성된 후 연결 소자를 이용해 상기 센서 디바이스 모듈에 연결되게 구현되는 것을 특징으로 한다.
- [0010] 본 발명에 있어서, 상기 센서 디바이스 모듈, 및 이에 연결되는 센서부의 적어도 하나 이상의 센서는, 각기 미리 지정된 비트 수의 아이디(ID)가 설정되는 것을 특징으로 한다.

- [0011] 본 발명에 있어서, 상기 센서 디바이스 모듈의 통신부나 내부 프로세서는, 상기 통합 제어기의 지시나 요청 메시지에 의한 초기화, 상기 통합 제어기의 주기적인 폴링에 의한 센싱 정보의 전송, 및 상기 센서 디바이스 모듈의 이벤트 발생에 의한 비동기적인 센싱 정보의 전송을 수행하는 프로토콜이 구현되는 것을 특징으로 한다.
- [0012] 본 발명에 있어서, 상기 센서 디바이스 모듈의 통신부나 내부 프로세서는, 상기 통합 제어기로부터 셋업 초기화(SETUP_INIT) 메시지를 수신하면, 센서 디바이스 모듈 및 적어도 하나 이상의 센서의 초기화 동작을 수행하고, 아울러 상기 통합 제어기에 적어도 하나 이상의 지정된 응답 메시지(RESPONSE_INIT, RESPONSE_STATUS, RESPONSE_DATA_INIT)를 출력하며, 상기 통합 제어기로부터 상기 센서 디바이스 모듈이나 센서에 대한 리셋 메시지(SETUP_INIT_SNODE, SETUP_INIT_SENSOR)를 수신하면, 센서 디바이스 모듈이나 센서를 리셋한 후 리셋 완료 응답(RESPONSE_RESET)을 상기 통합 제어기에 출력하도록 프로토콜이 구현되는 것을 특징으로 한다.
- [0013] 본 발명에 있어서, 상기 셋업 초기화(SETUP_INIT) 메시지는, 통합 제어기 번호(cgno), 온실 내 위치(Location), 모니터 유형(monitorType), 유형별 설정값(monitor value), 임계유형(monitorEvent), 및 임계값(monitor Limit) 정보 중 적어도 하나 이상을 포함하고, 상기 센서 디바이스 모듈의 센서 초기화 동작은, 센서 디바이스 모듈 구성정보 설정, 센서 디바이스 모듈 상태정보 설정, 센서 상태정보 설정, 및 센서 HW 초기화 중 적어도 하나 이상을 포함하는 것을 특징으로 한다.
- [0014] 본 발명에 있어서, 상기 센서 디바이스 모듈의 통신부나 내부 프로세서는, 상기 통합 제어기로부터 상태정보 요청 메시지(REQUEST_SNODE_INFO)를 수신하면, 상기 요청에 대응하는 상태정보가 포함된 응답 메시지(RESPONSE_STATUS)를 생성하여 출력하며, 상기 통합 제어기로부터 센서정보 요청 메시지(REQUEST_SNODE_VALUE)를 수신하면, 센서 읽기를 한 후에 응답 메시지(RESPONSE_DATA_PASSIVE)를 상기 통합 제어기에 출력하고, 이 과정에서 센서에 오류가 발생하면 에러 알람 메시지(NOTIFY_ERROR)를 상기 통합 제어기에 출력하고, 상기 통합 제어기로부터 상기 센서 디바이스 모듈에 대한 모니터링 유형설정 요청 메시지(SETUP_SNMODE)나 센서에 대한 모니터링 유형설정 요청 메시지(SETUP_SMODE)를 수신하면, 상기 각 모니터링 유형설정 요청 메시지(SETUP_SNMODE, SETUP_SMODE)에 따라 모니터링 유형을 설정한 후에 상기 통합 제어기에 응답 메시지(RESPONSE_SETUP)를 각기 출력하도록 프로토콜이 구현되는 것을 특징으로 한다.
- [0015] 본 발명에 있어서, 상기 응답 메시지(RESPONSE_STATUS)는 센서 디바이스 모듈 ID(snid), 센서 디바이스 모듈 상태(status), 초기화 상태(isInit), 모니터 유형(monitorType), 유형별 설정값(monitorValue), 소프트웨어 버전(swVer), 프로파일 버전(profileVer), 통신오류 카운트(commErrorNuma), 서비스 오류 카운트(serviceErrorNum), 센서 ID(sid), 센서 상태(state), 임계유형(monitorevent)(임계기준), 및 임계치(monitorlimit) 중 적어도 하나 이상을 포함하는 것을 특징으로 한다.
- [0016] 본 발명에 있어서, 상기 센서 디바이스 모듈의 통신부나 내부 프로세서는, 센서 디바이스 모듈이 최초 구동되면서 상기 통합 제어기에 연결 요청 메시지(REQUEST_CONNECTION)를 보내 통합 제어기에 연결하기 위한 연결정보를 받고, 실시간으로 측정된 센싱 정보를 알림 메시지(NOTIFY_DATA_ACTIVE)를 통해 상기 통합 제어기에 출력하며, 또한 이벤트 발생, 통신 상태, 및 오류 발생을 각기 알리는 메시지(NOTIFY_DATA_EVENT, NOTIFY_ALIVE, NOTIFY_ERROR)를 상기 통합 제어기에 자체적으로 출력하도록 프로토콜이 구현되는 것을 특징으로 한다.

발명의 효과

- [0017] 본 발명의 일 측면에 따르면, 본 발명은 표준화된 인터페이스를 통해 사물 인터넷(IoT : Internet of Things) 기반 원격 모니터링 및 관리 시스템의 구축 시 소규모 IoT 플랫폼에 해당하는 스마트 관제 시스템(또는 스마트 관제 플랫폼)에 곧바로 적용할 수 있도록 함으로써 스마트 관제 시스템의 설계 시 시간적 부담 및 비용적 부담을 감소시킬 수 있도록 한다.

도면의 간단한 설명

- [0018] 도 1은 본 발명의 일 실시예에 따른 스마트 관제 시스템을 위한 센서 디바이스 모듈의 개략적인 구성을 보인 예시도.
- 도 2는 상기 도 1에 있어서, 통합 제어기의 제어에 의한 상기 센서 디바이스 모듈의 초기화 프로토콜을 설명하기 위한 예시도.
- 도 3은 상기 도 1에 있어서, 통합 제어기가 센서 디바이스 모듈의 상태 정보와 센서 정보를 요청하고, 센서 디바이스 모듈 및 센서의 모니터유형과 임계치를 설정하는 프로토콜을 설명하기 위한 예시도.

도 4는 상기 도 1에 있어서, 센서 디바이스 모듈이 이벤트에 의해 센서 정보를 통합 제어기에 알리는 프로토콜을 설명하기 위한 예시도.

도 5는 본 발명의 일 실시예에 따른 스마트 관제 시스템을 위한 센서 디바이스 모듈의 성능을 실험한 결과 데이터를 보인 예시도.

발명을 실시하기 위한 구체적인 내용

- [0019] 이하, 첨부된 도면을 참조하여 본 발명에 따른 스마트 관제 시스템을 위한 센서 디바이스 모듈의 일 실시예를 설명한다.
- [0020] 이 과정에서 도면에 도시된 선들의 두께나 구성요소의 크기 등은 설명의 명료성과 편의상 과장되게 도시되어 있을 수 있다. 또한, 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례에 따라 달라질 수 있다. 그러므로 이러한 용어들에 대한 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다.
- [0021] 도 1은 본 발명의 일 실시예에 따른 스마트 관제 시스템을 위한 센서 디바이스 모듈의 개략적인 구성을 보인 예시도이다.
- [0022] 도 1에 도시된 바와 같이, 본 실시예에 따른 스마트 관제 시스템을 위한 센서 디바이스 모듈(100)은, 통신부(110), 전원부(120), 및 센서부(130, 131, 132, 133)를 포함한다.
- [0023] 본 실시예에 따른 상기 센서 디바이스 모듈(100)은 상기 통신부(110)를 통해 상위 레벨 제어기인 통합 제어기(200)와 이더넷, RS485, 계측 제어기 통신망(CAN Controller Area Network) 등의 유선 기반의 통신과 지그비(Zigbee), 무선 랜(WLAN) 등의 무선 기반의 통신을 이용해 통신할 수 있다.
- [0024] 상기 센서 디바이스 모듈(100)은 상기 센서부(130, 131, 132, 133)를 통해 측정된 센싱 정보(또는 센싱 값)를 상위 레벨 제어기인 상기 통합 제어기(200)에 전달한다.
- [0025] 예컨대 상기 센서 디바이스 모듈(100)은 상기 센서부(130, 131, 132, 133)에 포함된 적어도 하나 이상의 센서(예 : 온도 센서, 습도 센서, 향기 센서, 가스 센서, CO₂ 센서, 풍향 센서, 이미지 센서, 조도 센서, 강우 센서 등)를 통해 측정된 센싱 정보(또는 센싱 값)를 상위 레벨 제어기인 상기 통합 제어기(200)에 전달한다.
- [0026] 상기 센서부(130, 131, 132, 133)는 관제 대상의 상태나 혹은 관제 대상 주변의 상태 정보를 수집하는 기능을 수행하며, 센서 보드(예 : PCB)에 일체형으로 탑재되어 구현되거나, 또는 전자 소자(예 : 수동 소자, 능동 소자 등)와 함께 분리형으로 구성된 후 연결 소자(예 : 커넥터)를 이용해 센서 디바이스 모듈(100)에 연결될 수 있다.
- [0027] 상기 센서부(130, 131, 132, 133)는 8비트의 센서 ID를 가지며, 상기 센서 디바이스 모듈(100)은 20비트의 모듈 ID를 가질 수 있다.
- [0028] 또한 통신을 위한 유무선 네트워크 구성에 있어서, 상기 통신부(110)는 통신 방식에 따른 미디어 액세스 프로토콜(MAC), 망(Topology), 및 네트워크 프로토콜(network protocol)의 구성과 운영을, 시스템을 구성하는 설치자 및 사용자에게 따라 달리할 수 있으며, 기존의 통신 표준을 준용하여 사용할 수 있다.
- [0029] 상기 통신부(110)는 상기 센서부(130, 131, 132, 133)를 통해 수집된 정보를 상위 레벨 제어기인 상기 통합 제어기(200)의 요청에 따라 전송할 수 있으며, 미리 지정된 특정 이벤트가 발생할 경우에도 상기 센서부(130, 131, 132, 133)를 통해 수집된 정보를 상위 레벨 제어기인 상기 통합 제어기(200)에 전송할 수 있다.
- [0030] 따라서 상기 통신부(110)는 소형(예 : 4비트, 8비트, 16비트 등) 프로세서(미도시, 예 : CPU)를 포함하여 구현될 수 있다.
- [0031] 상기 전원부(120)는 상기 통신부(110) 및 센서부(130, 131, 132, 133)에 구동 전원을 공급하며, 상시 전원 및 비상 전원(2차 전원)을 구비할 수 있다.
- [0032] 상기 통합 제어기(200)는 상기 센서 디바이스 모듈(100)로부터 정보를 더 상위 레벨의 제어기(미도시, 예 : 서버)로 전달한다.
- [0033] 또한 상기 통합 제어기(200)는 IoT 기반의 스마트 관제 시스템의 통신 네트워크 게이트웨이 기능을 수행할 수 있으며, 유지 보수에 대비하여 자신의 상태 및 센서 디바이스 모듈(100)의 상태를 내부 메모리(미도시)에 저장

및 유지할 수도 있다.

- [0034] 본 실시예에 따른 센서 디바이스 모듈(100)은 IoT 플랫폼의 상위 레벨 제어기인 상기 통합 제어기(200)와 유선 혹은 무선 통신 인터페이스를 이용하여 통신하는데, 상기 통합 제어기(200)는 센서 디바이스 모듈(100)을 초기화한 후, 상기 센서 디바이스 모듈(100)로의 주기적인 폴링에 의한 센싱 정보 전송 방법과 상기 센서 디바이스 모듈(100)로부터 비동기적으로 전송되는 이벤트에 의한 센싱 정보 전송 방법을 통해 정보를 수집한다.
- [0035] 상기와 같이 통합 제어기(200)와 센서 디바이스 모듈(100) 간의 폴링 및 이벤트에 의한 센싱 정보를 전송하기 위한 통신 방법은, 통합 제어기(200)의 지시에 의한 센서 디바이스 모듈(100)의 초기화, 통합 제어기(200)의 요청에 의한 센싱 정보의 전송(On Request에 의한 센싱 정보 전송), 및 센서 디바이스 모듈(100)의 이벤트에 의한 센싱 정보의 알림(이벤트에 의한 센싱 정보 전송)으로 구분할 수 있다.
- [0036] 즉, 상기 센서 디바이스 모듈(100)은 상기 통합 제어기(200)의 지시나 요청에 의한 상기 센서 디바이스 모듈(100)(또는 센서)의 초기화, 상기 통합 제어기(200)의 요청에 의한 센싱 정보의 전송(On Request에 의한 센싱 정보 전송), 및 센서 디바이스 모듈(100)의 이벤트에 의한 센싱 정보의 알림(이벤트에 의한 센싱 정보 전송)을 지원할 수 있도록 구현된다.
- [0037] 상기 센서 디바이스 모듈(100)의 구체적인 프로토콜 구현 방법에 대해서 도 2 내지 도 4를 참조하여 설명한다.
- [0038] 도 2는 상기 도 1에 있어서, 통합 제어기(200)의 제어에 의한 상기 센서 디바이스 모듈(100)의 초기화 프로토콜을 설명하기 위한 예시도이다.
- [0039] 도 2에 도시된 바와 같이, 통합 제어기(200)가 센서 디바이스 모듈(100)의 셋업 초기화(SETUP_INIT) 메시지를 통해, 통합 제어기 번호(cgno), 온실 내 위치(Location), 모니터 유형(monitorType), 유형별 설정값(monitor value), 임계유형(monitorEvent), 및 임계값(monitor Limit) 정보를 센서 디바이스 모듈(100)에 전달한다.
- [0040] 이에 따라 상기 셋업 초기화(SETUP_INIT) 메시지를 수신한 상기 센서 디바이스 모듈(100)(더 구체적으로 센서 디바이스 모듈의 프로세서나 통신부)은 센서 디바이스 모듈(100) 및 각 센서의 초기화 동작을 수행하면서 상기 통합 제어기(200)에 응답 메시지(예 : RESPONSE_INIT, RESPONSE_STATUS, RESPONSE_DATA_INIT)를 출력한다.
- [0041] 예컨대 상기 센서 디바이스 모듈(100)의 센서 초기화 동작은, 센서 디바이스 모듈 구성정보(예 : 통합제어기 번호, 센서 디바이스 모듈 위치, 통신오류 회수, 서비스오류 회수 등) 설정, 센서 디바이스 모듈 상태정보(예 : 모니터 유형, 유형별 설정값 등) 설정, 센서 상태정보(예 : 모니터 유형, 유형별 설정값, 임계기준, 임계값) 설정, 및 센서 HW 초기화(예 : 최초 센서 읽기, 센서 초기화 성공 유무 기록, 센서 디바이스 모듈 상태정보 설정 등)를 포함한다.
- [0042] 참고로 상기 응답 메시지(예 : RESPONSE_INIT, RESPONSE_STATUS, RESPONSE_DATA_INIT)에서 "RESPONSE_INIT"는 센서 디바이스 모듈(100)의 초기화 완료 응답, "RESPONSE_STATUS"은 센서 디바이스 모듈(100)의 상태정보 알림, "RESPONSE_DATA_INIT"은 센서 디바이스 모듈(100)의 환경정보 알림을 각기 의미한다.
- [0043] 그리고 상기 센서 디바이스 모듈(100)의 초기화 완료 응답 메시지(RESPONSE_INIT)에는 통합제어기 번호(cgno), 센서 디바이스 모듈 ID(snid), 센서 개수(sno), 온실 내 위치(location), 센서 ID(sensorid), 및 센서 유형(sensortype)이 포함된다.
- [0044] 예컨대 상기 센서 유형(sensortype)으로는 온도센서(TEMPERATURE), 습도 센서(HUMIDITY), 광량센서(ILLUMINATION), CO₂센서(CO2), 및 토양수분센서(MOISTURE) 등이 포함된다.
- [0045] 또한 상기 통합 제어기(200)는 상기 센서 디바이스 모듈(100)에 리셋(reset) 메시지(SETUP_INIT_SNODE)를 전송할 수 있으며, 또한 센서 리셋(reset) 메시지(SETUP_INIT_SENSOR)를 전송할 수 있다. 이에 따라 상기 센서 디바이스 모듈(100)은, 센서 디바이스 모듈의 리셋 완료 응답(RESPONSE_RESET), 및 센서의 리셋 완료 응답(RESPONSE_RESET) 메시지를 상기 통합 제어기(200)에 각기 출력한다.
- [0046] 아울러 상기 센서 디바이스 모듈(100)은 상기 리셋 완료 응답 메시지(RESPONSE_RESET)를 전송할 때 상술한 나머지 응답 메시지(RESPONSE_STATUS, RESPONSE_DATA_INIT)를 함께 전송한다.
- [0047] 도 3은 상기 도 1에 있어서, 통합 제어기(200)가 센서 디바이스 모듈(100)의 상태 정보와 센서 정보를 요청하고, 센서 디바이스 모듈 및 센서의 모니터 유형과 임계치를 설정하는 프로토콜을 설명하기 위한 예시도이다.

- [0048] 도 3에 도시된 바와 같이, 통합 제어기(200)가 센서 디바이스 모듈(100)에 상태정보 요청 메시지(REQUEST_SNODE_INFO)를 출력한다.
- [0049] 이에 따라 상기 센서 디바이스 모듈(100)은 상기 통합 제어기(200)에 응답 메시지(RESPONSE_STATUS)를 출력한다.
- [0050] 예컨대 상기 응답 메시지(RESPONSE_STATUS)에는 센서 디바이스 모듈 ID(snid), 센서 디바이스 모듈 상태(status), 초기화 상태(isInit), 모니터 유형(monitorType), 유형별 설정값(monitorValue), 소프트웨어 버전(swVer), 프로파일 버전(profileVer), 통신오류 카운트(commErrorNuma), 서비스 오류 카운트(serviceErrorNum), 센서 ID(sid), 센서 상태(state), 임계유형(monitorevent)(임계기준), 및 임계치(monitorlimit)가 포함된다.
- [0051] 여기서 상기 초기화상태(isInit)는 Fail(0), Success(1), 및 Unavailable(2)을 값으로 가지고, 상기 센서 디바이스 모듈 상태(status)와 센서 상태(state)는 Normal(1), Abnormal(2), 및 Unavailable(3)을 값으로 가진다.
- [0052] 상기 모니터 유형(monitorType)은 Passive(1), Active(2), Event(3), 및 Unavailable(4)을 값으로 가진다.
- [0053] 상기 유형별 설정값(monitorValue)은 모니터 유형이 PASSIVE(수동), ACTIVE(능동) 이면 시간(분)을 값으로 하고, EVENT 이면 트리거 유형을 값으로 한다.
- [0054] 상기 임계유형(monitorevent)(임계기준)은 없음(0), 이상(1), 이하(2), 초과(3), 미만(4), 동등(5)으로 구분하며, 임계유형과 임계치는 모니터 유형이 EVENT인 경우만 유효하다.
- [0055] 상기 센서 디바이스 모듈(100)의 센서정보 요청은 상기 통합 제어기(200)로부터 센서정보 요청 메시지(REQUEST_SNODE_VALUE)에 의해 이루어진다.
- [0056] 상기 센서정보 요청 메시지(REQUEST_SNODE_VALUE)를 수신한 상기 센서 디바이스 모듈(100)은 센서 읽기를 한 후에 응답 메시지(RESPONSE_DATA_PASSIVE)를 상기 통합 제어기(200)에 출력한다. 만약 이 과정에서 센서에 오류가 발생하면 에러 알람 메시지(NOTIFY_ERROR)를 상기 통합 제어기(200)에 출력한다.
- [0057] 예컨대 상기 응답 메시지(RESPONSE_DATA_PASSIVE)는 센서 디바이스 모듈 ID(snid), 센서 ID(sid), 및 액추에이터 설정값(value) 항목을 포함한다.
- [0058] 상기 통합 제어기(200)는 상기 센서 디바이스 모듈(100)에 대한 모니터링 유형설정 요청 메시지(SETUP_SNMODE), 또는 센서에 대한 모니터링 유형설정 요청 메시지(SETUP_SMODE)를 출력한다.
- [0059] 여기서 상기 센서에 대한 모니터링 유형설정 요청 메시지(SETUP_SMODE)는 센서 ID(id), 모니터 유형(type), 유형별 설정값(value), 임계기준(event), 및 임계값(limit) 항목을 포함한다.
- [0060] 이에 따라 상기 센서 디바이스 모듈(100)은 상기 각 모니터링 유형설정 요청 메시지(SETUP_SNMODE, SETUP_SMODE)에 따라 유형을 설정한 후에 상기 통합 제어기(200)에 응답 메시지(RESPONSE_SETUP)를 각기 출력한다.
- [0061] 이상으로 상기 도 2 및 도 3에서는 상기 통합 제어기(200)로부터의 지시나 요청에 따라 상기 센서 디바이스 모듈(100)이 응답하는 프로토콜에 대해서 설명하였다. 그러나 상기 통합 제어기(200)로부터의 지시나 요청이 없더라도 상기 센서 디바이스 모듈(100)은 이벤트에 의해 상기 통합 제어기(200)로 초기 연결을 요구하거나, 혹은 센서 값, 오류 발생, 및 통신 상태 등을 알려줄 수 있다.
- [0062] 도 4는 상기 도 1에 있어서, 센서 디바이스 모듈(100)이 이벤트에 의해 센서 정보를 통합 제어기(200)에 알리는 프로토콜을 설명하기 위한 예시도이다.
- [0063] 도 4에 도시된 바와 같이, 센서 디바이스 모듈(100)은 최초 구동되면서 통합 제어기(200)에 연결 요청 메시지(REQUEST_CONNECTION)를 보내 연결정보(예 : 통합제어기 번호, 위치, 모니터 유형 등)를 받아온다.
- [0064] 또한 상기 센서 디바이스 모듈(100)은 실시간으로 측정된 센서 값을 알림 메시지(NOTIFY_DATA_ACTIVE)를 통해 상기 통합 제어기(200)에 출력한다.
- [0065] 예컨대 상기 알림 메시지(NOTIFY_DATA_ACTIVE)는 통합 제어기 번호(cgno), 센서 디바이스 모듈 ID(snid), 센서 ID(sid), 센서 값(value), 모니터 유형(mode), 임계기준(criteria), 센서 유형(type), 트리거 유형(trigger), 및 알림 시간(time) 항목을 포함한다.

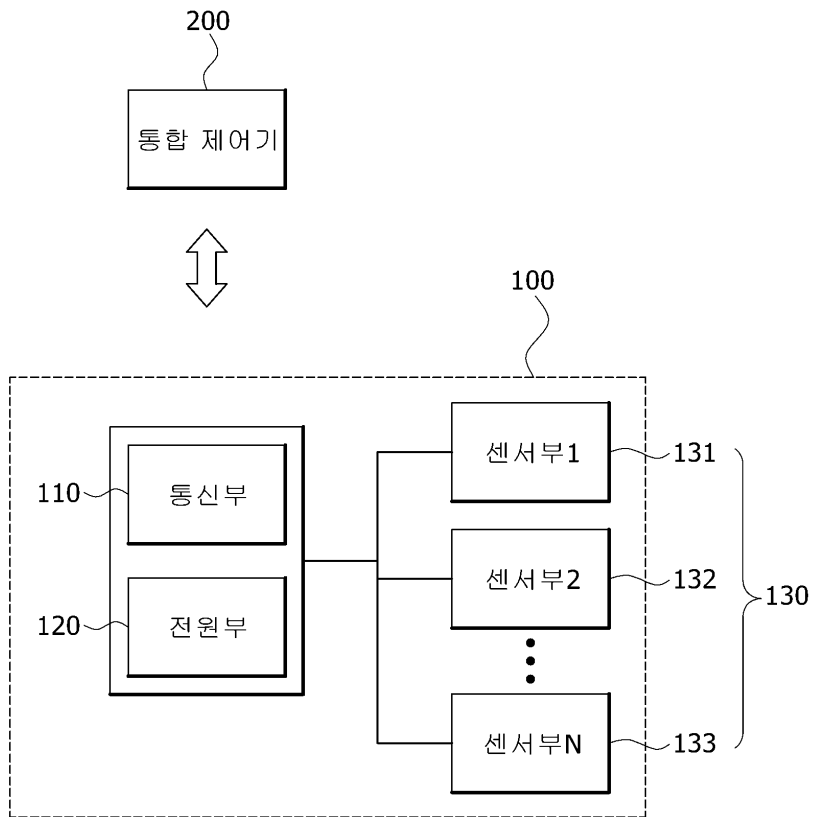
- [0066] 여기서 상기 트리거는 센서 디바이스 모듈(100)이 보내는 트리거 유형에 따라 통합 제어기(200)가 특정 액추에이터를 구동하는 기능이다.
- [0067] 예컨대 상기 트리거 유형으로는 천장 닫기(CLOSE_TOP_WINDOW), 측창 열기(OPEN_SIDE_WINDOW), 측창 닫기(CLOSE_SIDE_WINDOW), 유동팬 가동(OPEN_FLOW_FAN), 유동팬 중지(CLOSE_FLOW_FAN), 관수펌프 가동(OPEN_WATER_PUMP), 및 관수펌프 중지(CLOSE_WATER_PUMP) 항목을 포함한다.
- [0068] 이 외에도 상기 센서 디바이스 모듈(100)은 이벤트 발생, 통신 상태, 및 오류 발생을 각기 알리는 메시지(NOTIFY_DATA_EVENT, NOTIFY_ALIVE, NOTIFY_ERROR)를 상기 통합 제어기(200)에 출력한다.
- [0069] 도 5는 본 발명의 일 실시예에 따른 스마트 관제 시스템을 위한 센서 디바이스 모듈의 성능을 실험한 결과 데이터를 보인 예시도로서, 본 실시예에 따른 센서 디바이스 모듈(100)의 수행 시간을 프로파일링하기 위하여 64비트 PC에서 넷빈즈(NetBeans, 자바 개발 오픈 소프 IDE 프로그램) IDE(통합 개발 환경) 프로파일 기능을 사용하여 실험을 실시하였다.
- [0070] 좀 더 구체적으로 본 실시예에 따른 센서 디바이스 모듈(100)을 자바언어로 구현한 소스를 사용하여 센서 디바이스 모듈(100)의 실행 프로파일을 모니터링(도 6 참조)하였다.
- [0071] 이 실험 결과에 의하면 센서 디바이스 모듈의 구동(main), 센서 디바이스 모듈의 초기화 요청 HTTP 서비스(pool-1-thread-1), 센서 디바이스 모듈 센서 값 요청 HTTP 서비스(pool-1-thread-2), 반복주기 서비스(Thread-9), 센서노드 상태정보 요청 HTTP 서비스(pool-1-thread-3) 등이 모두 1초 이내로 처리되고 응답되어 안정적인 실시간 처리를 보장하고 있음을 알 수 있다.
- [0072] 상기와 같이 본 실시예에 따른 센서 디바이스 모듈은 리눅스 기반 라즈베리 파이와 같은 오픈 하드웨어에 탑재하여 구현할 수 있는 장점이 있으며, 기 구현된 온실 제어 시스템으로 실험한 결과에 의하면, 본 실시예에 따른 센서 디바이스 모듈은 IoT 플랫폼에서 통합 제어기와 연결되어 데이터 전달에 안전성과 실시간을 보장하고 있음을 알 수 있다.
- [0073] 이상으로 본 발명은 도면에 도시된 실시예를 참고로 하여 설명되었으나, 이는 예시적인 것에 불과하며, 당해 기술이 속하는 분야에서 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 따라서 본 발명의 기술적 보호범위는 아래의 특허청구범위에 의해서 정하여져야 할 것이다.

부호의 설명

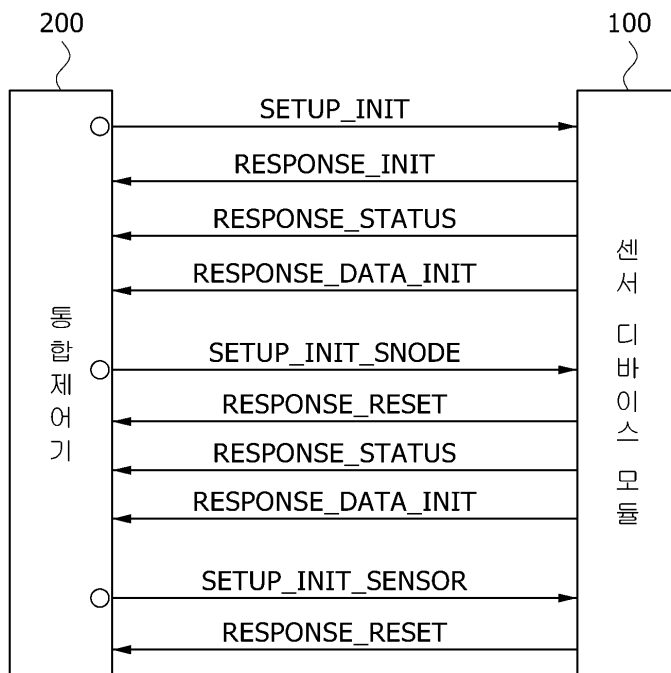
- [0074] 100 : 센서 디바이스 모듈
110 : 통신부
120 : 전원부
130, 131, 132, 133 : 센서부
200 : 통합 제어기

도면

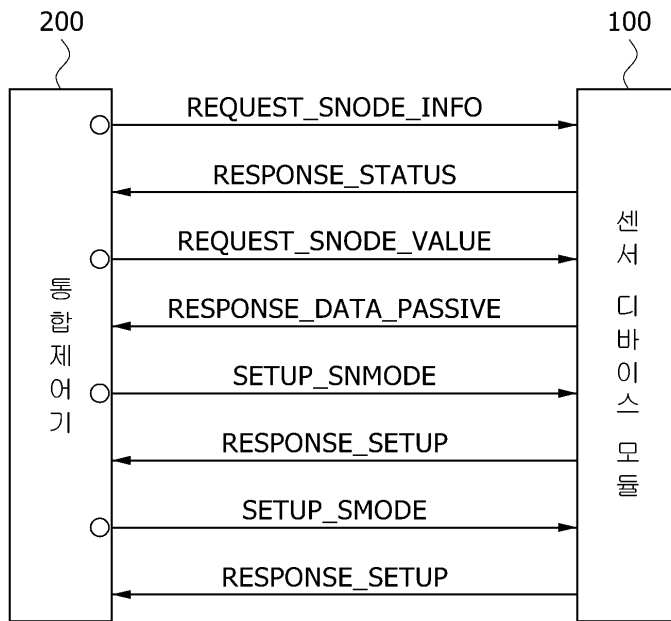
도면1



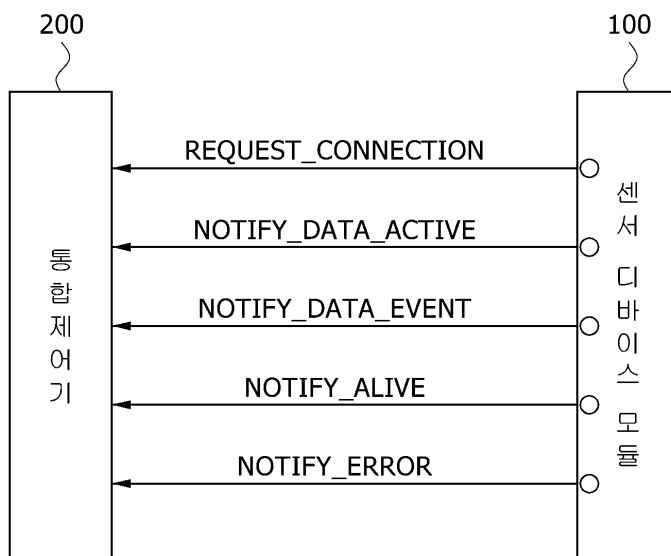
도면2



도면3



도면4



도면5

Call Tree - Method	Total Time ... ▼	Total Time	Total Time (CPU)
main		918 ms (100%)	918 ms
SensorNodeMain.main (String[])		918 ms (100%)	918 ms
SensorNodeMain.runServer ()		648 ms (70.6%)	648 ms
SensorNodeMain.getHostAddress ()		269 ms (29.4%)	269 ms
Self time		0.000 ms (0%)	0.000 ms
pool-1-thread-1		634 ms (100%)	453 ms
java.lang.Thread.run ()		634 ms (100%)	453 ms
pool-1-thread-2		479 ms (100%)	446 ms
java.lang.Thread.run ()		479 ms (100%)	446 ms
Thread-9		94.8 ms (100%)	94.8 ms
kmu.dssl.sensor.PeriodicManager.run ()		94.8 ms (100%)	94.8 ms
kmu.dssl.sensor.HttpClient.invokePut		84.5 ms (89.2%)	84.5 ms
java.lang.Integer.toString (int)		10.2 ms (10.8%)	10.2 ms
java.lang.Thread.sleep (long)		0.000 ms (0%)	0.000 ms
Self time		0.000 ms (0%)	0.000 ms
pool-1-thread-3		10.3 ms (100%)	10.3 ms
java.lang.Thread.run ()		10.3 ms (100%)	10.3 ms