1. 시스템 아키텍쳐

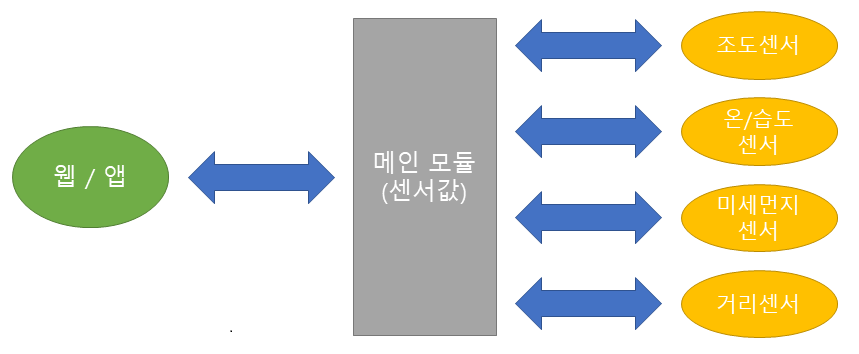


그림 . 센서 모니터링 시스템 아키텍쳐

센서모니터링 시스템을 블랙보드 아키텍쳐로 나타내면 그림1과 같다. 메인 모듈은 센서값을 받아서 저장하고 이걸 사용자에게 보여주게된다. IoT 시스템 대부분이 여러 독립된 컴포넌트들이 마스터와 작용하는 형태를 띄고 있어서 우리의 설계 또한 이런 아키텍쳐를 따르고자 했다.

텍스트, 지도이(가) 표시된 사진

자동 생성된 설명

그림 . 센서 모니터링 시스템 클래스 다이어그램

이것을 클래스 다이어그램으로 보면 Master에 Slave들과 Application이 연관되어 있다. 이 연관관계는 통신으로 이어지는데, 통신에 필요한 인증은 id, password로 하고 Slave같은 경우는 센서값도 실어서 통신한다. Master는 통신으로 받은 센서값을 DB에 저장하고 보여주거나 센서의 변화를 감지해서 Application에 알려주는 기능이 있다.

1. 개발 프로세스

* 위험분석  
  센서는 사용자가 원하는 위치에 자유롭게 설치할 수 있어야 하는데 콘센트가 없는 자리면 전원의 문제가 생긴다. 배터리를 이용하는 방법을 생각해둬야 한다.
* 일정

스크린샷이(가) 표시된 사진

자동 생성된 설명

* 팀 구조  
  김희주 : MCU 클라이언트의 센싱 기능 구현   
  백지훈 : MCU 클라이언트의 센싱 기능 구현, 어플리케이션 제작  
  임성용 : HTML5 + CSS로 프론트엔드 뷰 구현  
  형준혁 : 장고 혹은 NodeJS로 서버 구현
* 테스트 계획  
  (1) 슬레이브가 센서값을 서버로 송신하면 서버는 그것을 수신해서 DB에 저장할 수 있는지 테스트  
  (2) 웹이나 앱에서 DB에 있는 센서값들을 가져오는 것을 테스트  
  (3) 위 과정에 일어나는 통신이 지정된 ID, 비밀번호로 인증받아야만 가능하게 테스트

+ 코멘트

마스터에서 센서리스트를 관리해야 될 것 같음.  
사용자가 보고 제어할 수 있는 화면은 웹이나 앱 뿐이다.  
<id, password> 키 페어 관리는 어떻게? 키 페어는 수정도 가능

DB에 저장되는 센서값의 형태와 크기  
<시간 : String><슬레이브ID : char><값 : float>

GitHub 만들어야할듯