IoT모듈의 센서 진단을 위한 모바일 앱 설계 및 구현

김진홍 · 장시웅

동의대학교

Design and Implementation of mobile App for diagnosing sensor of IoT module

Jin-Hong Kim · Si-Woong Jang

Dong-Eui University

E-mail : 40490@deu.ac.kr/ swjang@deu.ac.kr

요 약

최근 IoT 모듈 즉, 사물 인터넷 기반 기술에 대한 연구가 활발히 진행되고 있으며, 다양한 제품들이 출시되고 있다. 대표적인 IoT 제품으로 가정에서 인터넷과 모바일 앱을 이용하여 카메라, 전등, 보일러 등을 제어하는 것을 볼 수 있다. 하지만 보통의 IoT모듈은 센서로 값을 추출하는 것이 목적이기 때문에 센서의 정확성과 고장 유·무를 판단하기 힘들다. 본 논문에서는 아두이노와 무선통신, 웹서버, 안드로이드 어플리케이션을 이용해 센서에서 추출된 값을 비교·분석한 표준편차를 이용해 센서의 고장 유·무를 판단 할 수 있는 모바일 앱을 설계 및 구현하고자 한다. 아두이노의 와이파이, 온습도센서 모듈 등을 이용하여 통신 연결을 하고 각종 환경, 제어정보들을 HTTP통신을 이용하여 웹서버와 통신하여 전달하고 제어한다. 이로써 사용자가 직접 IoT모듈에 가지 않아도 스마트폰 어플리케이션을 통해 센서 상태를 모니터링 및 진단하고 고장 유·무를 파악하여 교체시기를 알려주는 기능을 구현하였다.

키워드

무인 관리, IoT, 온습도 센서, 진단, 표준편차

Ⅰ. 서 론

사물인터넷에 관한 본격적 논의는 2005년 ITU(International Telecommunication Union)가 사물인터넷에 대한 보고서를 발간하면서 주목을 받아 시작되었으며, 2008년에서 2009년 사이에 Cisco, Gartner 등의 조사기관에서 사물인터넷이 유망키워드로 제시되면서 산업적 관심을 받게 되었다[1]. 세계적인 시장조사전문업체인 가트너에서는 2012년부터 2015년까지 지속적으로 주목할 만한 10대 전략 기술로 사물인터넷(IoT)을 발표했다[2]. 이러한 사물인터넷(IoT)의 핵심 기술은 센서(sensor)이다. 센서는 빛, 소리, 화학물질, 온도 등 내·외부에서 발생한 신호들을 수집하여 과학적인 방법 등으로 분석하여 내·외부의 각종 상태를 파악하는 장치이다[1,2]. 대부분의 IoT 센서 제품들이 센서로부터 값을 전달 받지만 그 값의 정확성에 대해서 판단을 하지 못하고 센서의 고장유무에 대해서도 센서에 직접 현상을 일으켜 이상유·무를 판단해야 하거나 여러 센서들을 이용한 Localized Faulty Sensor Detection 알고리즘을 이용하여 센서의 이상 유·무를 판단해야 한다.

본 논문에서는 하나의 센서를 이용해서 사용자가 스마트폰 어플리케이션을 통해 센서 값을 전달받고 표준편차를 이용해 그 값의 정확성과 센서의 고장유무를 판단하는 모바일 앱을 설계 및 구현하고자 한다.

Ⅱ. 관 련 연 구

2.1 Localized Faulty Sensor Detection 알고리즘

Localized Faulty Sensor Detection(LFSD) 알고리즘은 인접한 노드간의 센서 값 비교를 통해 센서의 이상 유·무를 판단하는 알고리즘이다. 각각의 센서 노드는 규칙적으로 자신의 센서값을 인접한 노드에게 전송하며, 알고리즘의 동작 순서는 그림3과 같이 5단계로 나뉘어 진다.

1번째 단계에서 각 노드들의 평가값 Cij는 0으로 설정되며, 평가값은 같은 시간에 측정된 현재 노드의 센서값 Si와 인접 노드로부터 측정된 센서값 Sj의 차와 미리 정의된 threshold 값보다 클 경우 1로 설정된다.

2번째 단계에서 각 노드의 상태값 Ti는 첫 번째 단계에서 결정된 인접 노드의 평가값을 토대로

변경되며, 상태값은 인접 노드의 상태에 따라 LG 또는 LF로 결정된다. 결정된 노드의 상태는 인접 노드에게 전송되며 수신된 인접 노드의 상태는 다음단계에서 상태를 결정하는데 사용된다.

3번째 단계에서는 인접 노드중 상태값 LG를 갖는 노드의 개수에 따라 노드의 GD 상태가 결정되며, 4번째 단계에서는 아직 결정되지 않은 노드의 상태값을 1~3번째 단계에서 결정된 인접 노드의 평가값과 상태값의 반복 비교를 통해 GD 또는 FT로 결정된다.

마지막으로 5번째 단계에서는 4단계까지 결정되지 못한 노드의 상태값이 인접 노드의 상태값을 통해 GD 혹은 FT로 결정된다[3].

Step 1: Each sensor S_i tests every member of $S_j \in N(S_i)$ to generate test $c_{ij}\{0, 1\}$ using the following method:
1: Each sensor S_i, set $c_{ij} = 0$ and compute d^t_{ij};
2: IF $|d^t_{ij}| > \theta 1$ THEN
3: Calculate $\Delta d^{\Delta t_l}_{ij}$;
4: IF $|\Delta d^{\Delta t_l}_{ij}| > \theta 2$ THEN $c_{ij} = 1$;

Step 2: S_i generates a tendency value T_i based upon its neighboring sensors' test value:
1: IF $\sum_{S_j \in N(S_i)} c_{ij} < \lceil |N(S_i)|/2 \rceil$, where $|N(S_i)|$ is the number of the S_i's neighboring nodes THEN
2: T_i = LG;
3: ELSE T_i = LF;
4: Communicate T_i to neighbors;

Step 3: Compare the number of S_i's LG neighboring nodes with different test results to determine its status:
1: IF $(\sum_{S_j \in N(S_i) \text{and} T_j = LG} (1 - 2c_{ij}) \geq \lceil |N(S_i)|/2 \rceil$ THEN
2: T_j = GD;
3: Communicate T_i to neighbors;

Step 4: For the remaining undetermined sensors, do the following steps in parallel for $Maxd$ cycles:
1: FOR $i = 1$ to n
2: IF T_i = LG or T_i = LF THEN
3: IF T_j = GD $\forall S_j \in N(S_i)$, THEN
4: IF $c_{ji} = 0$ THEN
5: T_i = GD;
6: ELSE T_i = FT;
7: ELSE repeat
8: Communicate T_i to neighbors;

Step 5: If ambiguity occurs, then the sensor's own tendency value determine its status:
1: FOR each S_i, IF $T_j = T_k = GD$ $\forall S_j, S_k \in N(S_i)$, where $j \neq k$, and IF $c_{ji} \neq c_{ki}$ THEN
2: IF $T_i = LG$ (or LF) THEN
3: $T_i = GD$ (or FT)

그림 1. Localized Faulty Sensor Detection 알고리즘

이 알고리즘은 필연적으로 여러 개의 센서가 필요 하다는 단점이 있기 때문에 하나의 센서로 기능하기에는 적합하지 않다.

2.2 IoT 센서 모듈

IoT 센서 모듈은 데이터를 얻기 위한 온습도 센서와 전원 제어를 위한 릴레이모듈, 웹서버와 통신을 위한 NodeMCU보드로 이루어져 있으며, 구성도는 그림 2와 같다.

1. 온습도 센서를 이용 온도·습도를 체크한다.
2. 현재 릴레이 모듈의 ON/OFF상태를 확인하여 저장한다.
3. NodeMCU 보드를 이용하여 WiFi허브에 연결하여 웹서버와 통신하여 데이터를 주고 받는다.
4. 웹서버와 HTTP 프로토콜을 이용해 통신하여 웹 페이지에 있는 제어 값을 전달받는다. 전달받은 제어 값에 따라 릴레이 모듈을 사용하여 온습도 센서의 전원을 제어한다.

그림 2. IoT 센서 모듈 구성도

2.3 표준편차

산포도의 하나인 표준편차는 자료가 평균을 중심으로 얼마나 퍼져 있는지를 나타내는 대표적인 수치이다. 표준편차의 단위는 자료의 단위와 일치한다. 표준편차가 0에 가까우면 자료 값들이 평균 근처에 집중되어 있음을 의미한다. 표준편차가 클수록 자료 값들이 널리 퍼져 있음을 의미한다.

확률변수 X의 평균이 μ, 표준편차가 σ인 정규분포를 따르면 확률밀도함수는

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

이다. 모집단에서 임의로 하나의 값을 취할 때이 값이

구간[μ-σ,μ+σ]에 속할 확률은 68.27%,

구간[μ-2σ,μ+2σ]에 속할 확률은 95.45%,

구간[μ-3σ,μ+3σ]에 속할 확률은 99.73%이다.

또한 다음과 같은 확률의 값은 신뢰구간을 구하는데 유용하게 쓰인다[4].

$P(\mu-1.65\sigma \leq X \leq \mu+1.65\sigma) = 0.90$

$P(\mu-1.96\sigma \leq X \leq \mu+1.96\sigma) = 0.95$

$P(\mu-2.58\sigma \leq X \leq \mu+2.58\sigma) = 0.99$

이 정보를 토대로 센서의 데이터를 1분 동안 2초당 1회씩 30개의 데이터를 받아 표준편차를 측정하여 센서 값의 이상유·무를 판별할 수 있다.

Ⅲ. 센서 진단 어플리케이션 설계

3.1 실험 장비

소켓통신을 사용하기 위해서는 인터넷 환경에 접속 가능한 IoT 센서 모듈이 필요하다. 또한 센서 모듈에서 송신된 데이터를 받기위한 안드로이드OS 기반의 스마트폰이 필요하다. IoT 센서 모듈은 WiFi를 이용하여 데이터를 송수신하고, 스마트폰은 자체 내장된 LTE를 사용하여 데이터를 송수신 한다.

3.2 센서 진단 어플리케이션 설계

먼저 어플리케이션을 설계할 때 IoT 센서 모듈에서 전송되는 데이터가 어떻게 출력되는지 확인해야 한다.

출력된 데이터들은 콤마(,)를 구분자로 사용하였다. 이 구분자를 이용하여 데이터를 나누면 온습도센서 전원여부, 온도, 습도로 나누어지며, 표1과 같다.

표 1. IoT센서 모듈 데이터

센서	데이터	설명
릴레이모듈	R+숫자	0 - 전원ON
		1 - 전원OFF
온도	숫자	온도 표시
습도	숫자	습도 표시

센서 진단 어플리케이션을 다음과 같이 설계하였다.

1. 센서 진단 알고리즘설계
2. 온·습도 센서 전원 확인 부분
3. 온·습도 확인과 갱신 버튼 부분
4. 최근의 온·습도 데이터 보기 부분
5. 센서의 이상 유·무 검사 부분

설계된 알고리즘의 구성은 그림3와 같다.

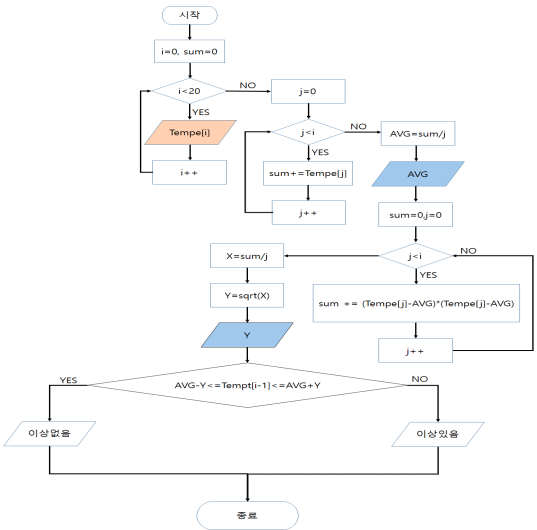

그림 3. 센서 진단 알고리즘 구성도

설계된 어플리케이션 화면은 그림4와 같다.

그림4에서 보이는 것처럼 갱신을 누르면 온·습도 센서에서 값을 가져와 보여주고 최근 온습도 버튼을 누르면 최근의 온·습도 데이터를 볼 수 있도록 설계하였다.

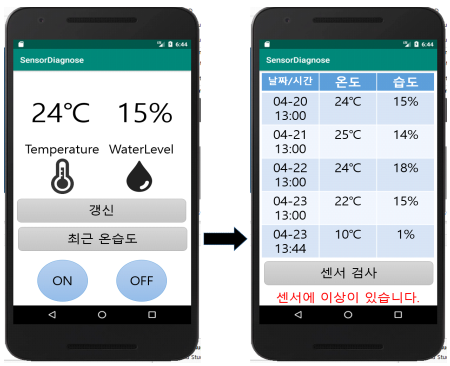

그림 4. 설계된 어플리케이션 화면

그림 3의 알고리즘을 사용하여 그림4에서 보이는 것처럼 센서 검사 버튼을 눌렀을 경우 최근 온도들의 표준편차를 구하여 마지막 온도가 표준에 속하지 않을 경우 센서에 이상이 있다고 판단 될 수 있다. 이상이 있을 경우 그림5의 왼쪽화면 ON/OFF 버튼을 이용하여 센서를 재부팅후 재검사 할 수 있도록 설계하였다.

Ⅳ. 결론

본 논문에서는 IoT 센서 모듈의 데이터를 이용하여 센서 진단 어플리케이션을 설계하였다. IoT 센서 모듈은 WiFi를 이용하여 센서 데이터를 송수신하여 정보를 제공하고, 스마트폰은 자체 내장된 LTE를 이용하여 센서 데이터 및 컨트롤 정보를 송수신한다.

송수신된 데이터를 표준편차를 이용해서 온·습도 센서의 이상 유·무를 판별하고 센서의 교체시기를 파악 할 수 있어 센서를 관리하는데 많은 도움이 될 것으로 예상된다.

또한, 개발된 시스템을 응용하면 온·습도 센서 뿐만 아니라 다른 센서의 이상유·무도 파악할 수 있다.

Acknowledgment

“이 논문은 2018년도 BB21+사업에 의하여 지원되었음.”

References

[1] Naver 용어로 보는 IT. 사물인터넷(Internet of Things) [Internet]. Available : https://terms.naver.com/entry.nhn?docId=3577301&cid=59088&categoryId=59096

[2] 방종현, 방효찬, 김세한, 김말희, 이인환, 최병철, 이강복, 강성수, 김호원, 사물인터넷의 미래, 한국, 전자신문사, 2017.

[3] 육의수, 윤성웅, 김성호, “Development of Fault Detection Algorithm Applicable to Sensor Network System”, 퍼지 및 지능시스템학회 논문지, 목포해양대학교, pp.760-765, 0712.

[4] Naver 수학백과. 표준편차 [Internet]. Available : https://terms.naver.com/entry.nhn?docId=3405400&cid=47324&categoryId=47324