

IoT 환경에서 라즈베리파이를 이용한 데이터 모니터링

이동형 · 이강민 · 윤현성 · 정재훈 · 김남호*

부경대학교

Data Monitoring using Raspberry Pi in IoT Environment

Dong-Hyung Lee · Kang-Min Lee · Hyeon-Seong Yun · Jae-Hoon Jung · Nam-Ho Kim*

Pukyong National University

E-mail : nhk@pknu.ac.kr

요 약

IoT 기술이 대중화됨에 따라 점차 많은 데이터가 발생하고 있으며, 데이터의 다양성 또한 증가하고 있는 추세이다. 특히 스마트팩토리 및 Home IoT 시스템에서는 센서를 통해 각종 데이터를 실시간으로 수집하고 처리하므로 데이터처리의 중요성이 크다. 본 논문에서는 라즈베리파이를 통해 IoT 환경에서 센서에 의해 발생하는 다양한 데이터를 수집, 분석 및 모니터링하는 방법을 제시하였다. 또한 위와 같은 방법으로 수집, 분석한 데이터를 스마트미러 및 모바일 어플리케이션과 연동시킴으로 시스템의 유용성을 검증하였다.

ABSTRACT

As IoT technology becomes popular, more and more data is being generated, and the diversity of data is also increasing. In particular, in smart factory or Home IoT systems, data processing is very important because various data is collected and processed in real time through sensors. In this paper, we present a method for collecting, analyzing, and monitoring various data generated by sensors in IoT environment through Raspberry Pi. We also validate its usefulness by demonstrating that the above processed data can be operated in conjunction with smart mirror and mobile application.

키워드

라즈베리파이, IoT, 스마트팩토리, 데이터 모니터링

1. 서 론

오늘날 정보통신기술이 급속도로 발전을 이룸에 따라 IoT 시스템은 점차 대중화되고 사용량이 크게 증가하였다. 점차 다양한 시스템에 IoT를 적용함에 따라 수집하는 데이터의 양과 다양성은 증가하였으며, 데이터 처리의 중요성은 더욱 증가하였다. 특히 Home IoT 및 스마트팩토리 시스템에서는 센서를 통해 각종 데이터를 실시간으로 수집하고 처리하므로 데이터를 효과적으로 수집하고 모니터링하는 기술이 중요하다[1-2]. 본 논문에서는 라즈

베리파이를 이용하여 IoT 환경의 센서에 의해 발생하는 다양한 데이터를 실시간으로 수집하고 모니터링하는 연구를 진행하였다. 또한 수집한 데이터를 스마트 미러 및 모바일 어플리케이션과 연동함으로써 생활환경 및 편의성을 개선할 수 있음을 보인다.

본 논문의 이후 구성은 다음과 같다. 2장에서는 시스템 및 센서의 구성을 제시하며, 3장에서는 센서로부터 수신한 데이터를 라즈베리파이를 통해 수집 및 모니터링하는 방법을 제시한다. 4장에서는 위와 같은 방법으로 수집한 데이터를 스마트 미러 및 모바일 어플리케이션을 통해 활용하는 방법을 제시하였으며, 결론에서는 앞으로의 연구 및 개선

* corresponding author

방안을 제시하였다.

II. 시스템 개요

본 논문에서는 라즈베리파이를 이용하여 스마트 팩토리 및 Home IoT 시스템에 적용할 수 있는 모니터링 시스템을 구성하였다. 모니터링 대상은 온습도, 미세먼지 등의 환경적 요인과 가스 및 화재 등 특별한 요인들을 포함한다. 데이터 생성을 위한 센서로는 온습도 센서(DHT22), 미세먼지 센서(PMS7003), 불꽃 감지 센서(SEN040129), 가스감지 센서(MQ-2) 등을 사용하였다.

다양한 센서에 의해 생성된 데이터는 아두이노 메가(Arduino mega)를 통해 수집하고 라즈베리파이(Raspberry Pi 4 Model B)로 전송하였으며, 라즈베리파이에서는 데이터베이스로의 수집과 동시에 실시간으로 모니터링 하였다. 또한 수집한 데이터는 전처리과정 및 분석 과정을 거쳐 스마트 미러 및 모바일 어플리케이션과 연동하였으며, 시스템을 모니터링하고 제어하는데 사용하였다. 시스템 전체 구성은 그림 1과 같다.

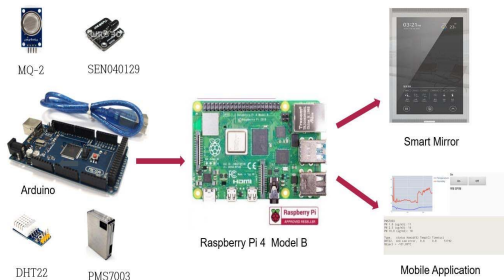


그림 1. 시스템 구성도

III. 데이터 수집 및 모니터링

데이터 수집 및 모니터링 과정에서는 아두이노에서 일차적으로 수집하고 전송한 센서 데이터를 라즈베리파이를 통하여 수집하고 실시간으로 모니터링하는 방법을 제시하였다. 또한 수집한 결과를 확인 및 분석하기 위한 전처리 과정에 대해서도 기술하였다.

3.1 데이터 수집

데이터 수집 과정은 라즈베리파이에서 Python 코드를 이용함으로써 진행되었으며, MariaDB 데이터베이스에 데이터를 저장하고 관리하였다. 아래 그림 2는 DHT22 센서를 통해 수집한 데이터를 MariaDB 서버로 실시간으로 수집 및 저장하기 위해 사용한 Python 코드이다.

```

arduino = serial.Serial('/dev/ttyACM0', 9600)
def input():
    data=arduino.readline()
    data=data.decode()[-2]
    data=data.split(',')
    return data
while True:
    try:
        with conn.cursor() as cur:
            DHT22_ins="insert into collect_data values(%s,%s,%s,%s)"
            while True:
                while datal=input():
                    data=input()
                    print(data)
                    if data[0]=="DHT22":
                        cur.execute(DHT22_ins,(data[0],time.strftime("%Y-%m-%d %H:%M:%S",
                                                                    time.localtime()),data[2],data[1]))
            conn.commit()
    
```

그림 2. Python을 이용한 데이터 수집 및 저장

Serial 라이브러리를 통해 수신하는 아두이노의 센서 데이터는 CSV(comma separated value) 형태로 구성되어 있다. 따라서 시리얼 데이터를 수신할 때 마다 decode 과정을 거친 이후 split을 통해 “,”를 기준으로 데이터의 종류를 구분하였다. 또한 구분된 데이터에 수집 시간을 추가함으로써 시계열을 생성하였으며, 미리 구성한 데이터베이스의 구조에 맞게 수집·저장하였다. 아래 그림 3은 DHT22 센서의 온도 및 습도 데이터를 수집시간과 함께 저장한 결과이다.

```

4 rows in set (0.001 sec)
MariaDB [raspi_db]> select * from DHT22;
+-----+-----+-----+-----+
| sensor | collect_time | temp | hum |
+-----+-----+-----+-----+
| DHT22 | 2021-04-08 17:09:22 | 26.1 | 41.3 |
| DHT22 | 2021-04-08 17:11:17 | 26.1 | 41.4 |
| DHT22 | 2021-04-08 17:11:47 | 26.1 | 41.5 |
| DHT22 | 2021-04-08 17:14:32 | 26.1 | 41.6 |
+-----+-----+-----+-----+
4 rows in set (0.000 sec)

```

그림 3. 수집 완료된 데이터

3.2 데이터 모니터링

데이터 모니터링 시스템은 수신한 데이터를 데이터베이스로 저장함과 동시에 실시간으로 확인할 수 있도록 구성하였다. 수신하는 데이터를 라즈베리파이에서 직접 확인하기 위해서는 Python Matplotlib 라이브러리를 이용하였으며, 원격 모니터링을 위해서는 Plotly 라이브러리 및 Webiopi 웹서버를 이용하였다.

Plotly 라이브러리를 통해서 수신한 데이터를 웹 서버에 주기적으로 업데이트함으로써 그래프와 데이터 프레임을 확인할 수 있도록 하였다. 또한 WebIOPi 웹서버를 통해서 시리얼 모니터링 기능과 더불어 Plotly 서비스와 연동하였으며, 모니터링 결과에 따라 Python 스크립트를 실행하고 제어하는 자동화 기능을 구현하였다.

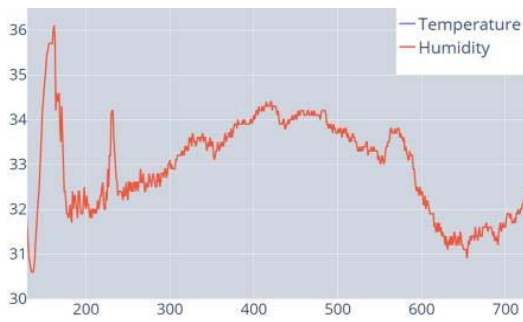


그림 4. 데이터 원격 모니터링

실시간 모니터링 과정 이후에는 전처리 및 분석 과정을 거쳤다. IoT 환경에서는 데이터통신 과정이나 센서 및 IoT 장비의 오류로 인해 잡음이 발생할 수 있으며, 이에 따라 이상치 및 결측치가 발생할 수 있다[3]. 따라서 수집한 데이터의 특정 부분을 모니터링하고 시스템을 제어하는 데 활용하기 위해서는 적절한 전처리 과정이 필요하다. 아래 그림은 데이터베이스에 수집·저장된 데이터 원본을 Pandas 라이브러리를 통해 인출하고 시각화 한 결과이다.

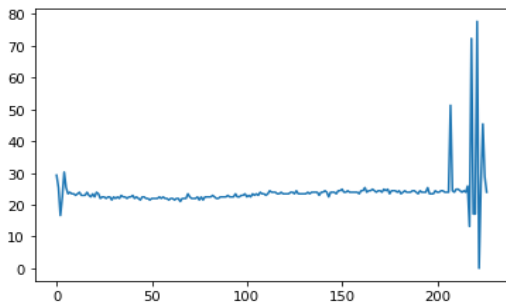


그림 5. 수집 데이터 그래프

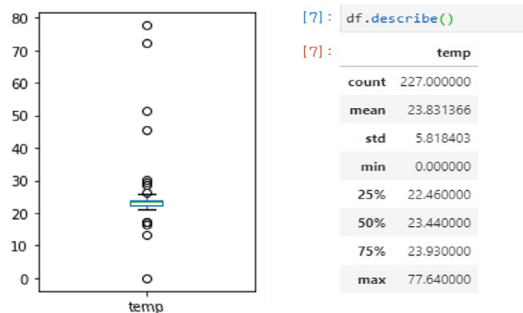


그림 6. 이상치 분석

저장된 데이터에는 센서 및 장치상의 오류로 발생한 결측치 및 이상치가 포함된 것을 확인할 수

있다. 위와 같은 데이터상의 오류는 전체 데이터의 통계적 측면에 부정적인 영향을 끼칠 뿐만 아니라 모니터링 이후 시스템을 제어하는 과정에도 부정적인 영향을 끼친다. 따라서 Pandas 라이브러리를 통해 이상치 및 결측치를 제거하는 과정을 거쳤다. 특히 이상치가 발생한 경우 이를 제거하기 위하여 IQR을 이용한 전처리 과정을 거쳤다. IQR은 사분위수의 상위 75%인 지점과 하위 25%지점 사이 구간을 의미한다. 즉, 상위 75%에서 1.5*IQR을 더한 값을 상한선, 하위 25%에서 1.5*IQR을 뺀 값을 하한선으로 제한하며, 해당 범위를 벗어난 값들은 이상치로 간주하여 제거하였다[4]. 아래 그림 7은 그림 5와 같이 수집한 데이터를 위와 같은 방법으로 전처리한 결과이다.

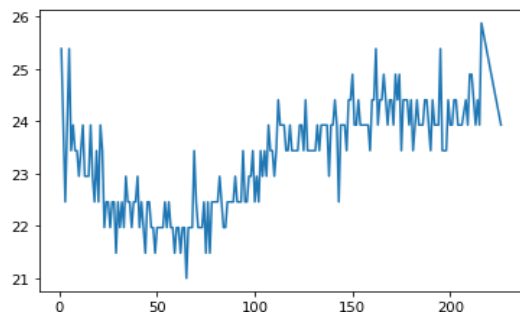


그림 7. 이상치 제거 데이터

IV. 스마트 미러 및 모바일 활용

위와 같은 과정을 통해 수집하고 분석한 데이터는 스마트 미러 및 모바일 어플리케이션과 연동하고 활용하였다. 스마트 미러는 거울 후면에 디스플레이를 위치시킴으로서 거울의 기능과 동시에 디스플레이에 나타난 정보를 이용할 수 있는 장치이다[5]. 본 시스템에서는 라즈베리파이 모니터에 하프미러 필름을 부착하여 거울의 기능과 동시에 디스플레이에 나타난 정보를 활용할 수 있도록 하였다.

스마트 미러에 디스플레이 하는 정보들은 Plotly 라이브러리를 통해 업로드한 온도, 습도 등의 그래프 및 센서 데이터와 연동하였다. 또한 모니터링 기능 이외에도 날씨, 뉴스정보 및 검색기능 등 다양한 편의기능을 제공하며, 구글 어시스턴트 모듈을 이용한 음성인식 기능도 추가하였다. 또한 스마트 미러는 내부에 적용할 수 있는 다양한 모듈을 제공하므로, 향후에는 음성인식 및 모션인식 기능과 IoT 기능을 연동함으로써 편의성의 확장을 시도할 계획이다.



그림 8. 스마트미러 실행 화면

모바일 어플리케이션을 통해서도 외부에서 데이터를 모니터링 하고 간단한 제어기능을 수행할 수 있도록 하였다. 모바일 어플리케이션은 MIT App Inventor 2를 사용하여 제작하였으며, WebIOPi 및 Plotly 서비스와 연동하여 간단한 모니터링 기능을 구현하였다. 또한 모바일의 On/Off 버튼을 통해 라즈베리파이에서 아두이노로 제어 신호를 보내고 주변 기기들을 제어할 수 있도록 하였다. 해당 기능들은 Remoteit 서비스를 이용함으로써 라즈베리파이와 같은 통신 환경이 아닐 때도 모니터링 및 제어를 수행할 수 있으며, 가스 및 화재센서 등으로부터 이상 신호가 감지되었을 때 자동으로 반응하는 기능 또한 구현하였다.

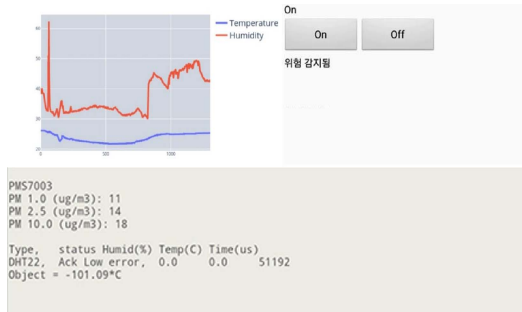


그림 9. 모바일 어플리케이션 활용

V. 결 론

본 논문에서는 라즈베리파이를 이용하여 IoT 환경에서 발생하는 다양한 데이터를 실시간으로 수집하고 모니터링 하는 시스템을 구현하였다. 수집한 데이터는 전처리 과정을 통해 필요한 정보를 도출할 수 있도록 하였으며, 모바일 및 스마트미러와 연동하여 생활환경과 편의성의 개선에 도움이 되도록 하였다.

센서에 의해 수집하는 온도, 습도, 미세먼지 등 환경적 요소는 건강을 관리하거나 생산 과정을 효율적으로 운영하는 데 있어서 중요하다. 따라서 본

시스템을 이용하여 다양한 IoT 센서 데이터를 모니터링 함으로서 환경 관리와 편의성 개선에 도움이 될 것을 기대한다. 또한 라즈베리 파이에서 파이썬 언어를 이용하여 구현한 시스템은 PC 등 다른 시스템에 쉽게 적용하고 이식할 수 있을 것을 기대한다.

본 논문에서 수집한 센서 데이터들은 시리얼 통신을 통해 수집되어 라즈베리파이와 연동하였지만, 향후에는 무선 원격 데이터 통신을 이용하여 넓은 범위의 데이터들을 수집하고 분석하는 시스템으로 발전시킬 계획이다.

References

- [1] H. W. Chu, Y. K. Suh, R. Lee, M. W. Park, R. Y. Jang, S. H. Lee, S. K. Song, "A Trie-based Indexing Scheme for Efficient Retrieval of Massive Spatio-Temporal IoT Sensor Data", *The Journal of KIISE*, Vol. 47, No. 12, pp. 1199-1207, Dec. 2020.
- [2] J. S. Kim, H. W. Chu, S. H. Kim, J. Y. An, J. Y. Lee, Y. K. Suh, R. Lee, M. W. Park, S. H. Lee, R. Y. Jang, S. K. Song, "Design and Implementation of a Layered Engine for Managing Heterogeneous Spatio-Temporal IoT Sensor Data", *In Proceedings of Symposium of the The Korean Institute of Information Scientists and Engineers*, pp. 130-132, Dec. 2020.
- [3] C. S. Chae, H. G. Kim, T. G. Kwon, J. H. Kang, "IIoT Big data Outlier removal S/W Framework design", *In Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, pp. 358-359, Nov. 2019.
- [4] J. H. Kang, H. W. Kang, C. S. Chae, H. G. Kim, H. S. Choi, "IIoT big data processing algorithm extension application design", *In Proceedings of Symposium of The Institute of Electronics and Information Engineers*, pp. 2130-2132, Aug. 2020.
- [5] J. W. Jeon, S. G. Jin, "SmartMirror Realization using GPU board", *In Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, pp.1294-1295, Jan. 2019.