Journal of the Korea Academia-Industrial cooperation Society
Vol. 17, No. 11 pp. 51-56, 2016

http://dx.doi.org/10.5762/KAIS.2016.17.11.51
ISSN 1975-4701 / eISSN 2288-4688

그리드 기반 키 선분배 방식을 사용하는 공장 설비 모니터링 시스템 설계 및 구현

조양희[1*], 박재표[2], 양승민[3]
[1]숭실대학교 컴퓨터학과, [2]숭실대학교 정보과학대학원, [3]숭실대학교 컴퓨터학부

Design and Implementation of Factory Equipment Monitoring System using Grid-based Key Pre-Distribution

YANGHUI CHO[1*], JAEPYO PARK[2], SEUNGMIN YANG[3]
[1]Division of Computing, Soongsil University
[2]Division of Information Technology Management, Soongsil University
[3]Division of Computer Science & Engineering, Soongsil University

요 약 본 논문에서는 아두이노 기반의 공장 설비 모니터링 시스템을 제안한다. 제안하는 시스템은 아두이노 플랫폼을 기반으로 하며 온도, 습도 그리고 조도를 측정하는 환경 센서와 압력 센서를 이용하여 공장의 환경 및 설비의 상태를 모니터링 한다. 모니터링 데이터는 RF(Radio Frequency) 트랜시버를 통해 서버에 연결되어 있는 지그비 코디네이터로 전송된다. 호스트 서버에 저장된 환경 센서와 압력센서의 데이터를 이용하여 공장의 환경과 설비의 압력 상태를 확인하고 설정된 알람 상태에 도달하면 관리자에게 보고하도록 설계하였다. 그리드 기반 키 선분배 방식을 사용하여 센서 노드를 인증하고 데이터 키를 동적으로 생성하여 모니터링 정보를 보호한다. 추가적인 배선 작업이 필요 없는 지그비 무선 센서 네트워크를 적용하여 공장 설비 모니터링 시스템을 실제 구현함으로서 효율적인 공장의 작업 환경 모니터링이 가능하다. 또한 불량이 발생한 경우, 작업 환경을 역으로 추적하여 불량 원인 분석에 활용할 수 있다. 아두이노 플랫폼과 확장 보드를 이용하여 평탄도나 진동 같은 센서를 추가하거나 확장 보드에 연결된 포트로 제어 하는 등의 기능 확장이 용이하다.

Abstract In this paper, we propose an Arduino-based plant monitoring system. The proposed system is based on the Arduino platform, using an environmental sensor and a pressure sensor for measuring temperature, humidity and illuminance in order to monitor the state of the environment and the facilities of the plant. Monitoring data are transmitted to a ZigBee coordinator connected to a server through a radio frequency transceiver. When using a pressure sensor and the environment sensor data stored on the host server, checking the pressure in the environment of the plant and equipment is intended to report any alarm status to the administrator. Using a grid line-based key distribution scheme, the authentication node dynamically generates a data key to protect the monitoring information. Applying a ZigBee wireless sensor network does not require additional wiring for the actual implementation of a plant monitoring system. Possible working-environment monitoring of an efficient plant can help analyze the cause of any failure by backtracking the working environment when a failure occurs. In addition, it is easy to expand or add a sensor function using the Arduino platform and an expansion board.

Keywords : Remote Monitoring System, WSN, Security, Pre-Distribution

*Corresponding Author : Yang-Hui CHO(Soongsil Univ.)
Tel: +82-10-5621-0430 email: vanillo@realtime.ssu.ac.kr
Received August 25, 2016 Revised October 4, 2016
Accepted November 10, 2016 Published November 30, 2016

1. 서론

무선 센서 네트워크는 물리적 현상이나 주변의 환경을 모니터링하고 수집된 자료를 서버로 보내는 센서 노드로 구성된다. 센서 노드는 수집된 자료를 서버로 보내거나 서버로부터 명령어를 수신하는 RF 트랜시버 모듈과 물리적 또는 환경적 조건을 모니터링하는 ADC 모듈로 구성된다. ADC 모듈로 읽은 데이터를 계산하거나 액추에이터를 제어하는 것 보다 RF 통신에서 더 많은 전류를 소비한다[1].

무선 센서 노드의 경우, 배포와 설치가 쉽지만 계산 능력과 저장 공간이 제한되어 있다. 무선 통신의 특성 때문에 전송 데이터의 기밀성 보호에 취약하다[1-2]. 산업용, 군사용 그리고 의료용 어플리케이션에서는 전송되는 정보에 대해 보안이 요구되고 있다. 그러나 무선 센서 네트워크 표준 프로토콜은 에너지 효율, 효율적인 데이터 전송 그리고 라우팅에 중점을 두고 있다[3].

부피가 크고 무거운 공장 설비의 경우, 배치 후 이동이 없이 고정된 위치에서 사용하게 된다. 본 논문에서는 공장 설비 배치와 유사한 이차원 그리드 기반으로 다항식 키를 분배하고 사전 분배된 키와 동적으로 생성한 데이터 키를 사용하여 전송 데이터의 기밀성을 보호하는 공장 설비 모니터링 시스템을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 공장 설비 모니터링 시스템의 구성 및 설계에 대하여 기술하고, 3장은 제안한 모니터링 시스템의 구현을 통해 효율성 검증을 위한 실험 결과에 대해 기술하며, 마지막으로 4장에서는 결론을 맺는다.

2. 공장 설비 모니터링 시스템

본 논문에서는ADC 모듈에 연결된 센서로부터 온도, 습도 그리고 조도의 환경 데이터와 압력 데이터를 입력받는다. 마이크로 컨트롤러는 정해진 시간 동안 입력된 아날로그 데이터를 센서 테이블의 값으로 변환하여 평균값을 계산하고 서버로 전송한다. Fig.1은 본 논문에서 제안하는 시스템의 구성도를 나타낸다. 메인 프로세서는 아두이노 DUE[4]를 사용하였고, 온습센서는 SHT71[5], 조도센서는 BH1750FVI[6]와 압력센서 FSR-402[7]를 사용하였다. 지그비 모듈 RP-M100[8]을 사용하여 센서로부터 수집된 데이터를 무선으로 호스트 서버에 전송하도록 구성하였다. 서버로 보내진 데이터는 그래프를 통해 실시간으로 보여 주고, 설정된 알람 값에 도달하면 관리자에게 문자메시지를 전송한다.

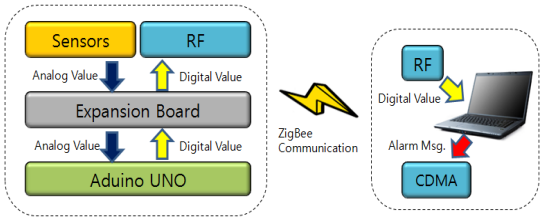

Fig. 1. Block Diagram of Factory Equipment Monitoring System

Fig. 2는 아두이노 DUE에 조립하여 센서 보드 연결을 편리하게 하고 485통신과 와이파이와 같은 유무선 통신 기능을 추가 할 수 있도록 설계된 확장 보드이다.

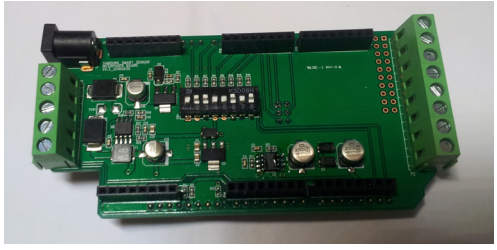

Fig. 2. Expansion Board

Fig. 3은 제안한 모니터링 시스템의 GUI 소프트웨어, 지그비 코디네이터 그리고 센서 노드의 제어 프로토콜이다. 모니터링 데이터를 받기 전에 관리자는 GUI 소프트웨어를 통해 지그비 코디네이터를 설정한다. 엔드 디바이스는 코디네이터에게 비콘(Beacon)을 요청하고 코디네이터로부터 비콘을 전송받은 엔드 디바이스는 자신의 NIB.MinJoinLQI값과 비교하여 값이 크거나 같으면 코디네이터에 조인을 시도한다. 엔드 디바이스가 조인에 성공하면 지그비 코디네이터는 서버로 엔드 디바이스의 주소를 전송한다. 새로운 엔드 디바이스 주소를 받은 서버는 엔드 디바이스의 아이디를 확인하고 키를 분배한다. 키를 분배 받은 엔드 디바이스는 정해진 시간 간격마다 센서로부터 입력받은 아날로그 데이터를 센서 테이블 값으로 변환하여 지그비 코디네이터로 전송한다.

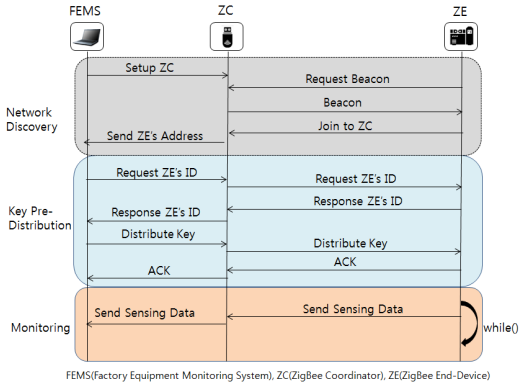

Fig. 3. Control Protocol for Factory Equipment Monitoring System

서버는 디바이스 아이디에 해당하는 좌표의 값을 이용하여 엔드 디바이스가 조인할 때 동적으로 생성된 해당 좌표의 다항식을 엔드 디바이스로 전송한다. Fig. 4와 같이 엔드 디바이스 ID 리스트는 이중 링크드 리스트로 관리하고, 자료구조는 디바이스 ID, X 좌표, Y 좌표, 동적 네트워크 주소로 구성된다.

```
typedef struct DeviceID
{
    int DeviceID;
    int x-coordinate;
    int y-coordinate;
    unsigned char NetAddr[4];
    struct DeviceID_LST* Next;
    struct DeviceID_LST* Prev;
}DeviceID_LST;
```

Fig. 4. Structure for End_Device ID

다항식 키의 자료구조는 Fig.5와 같다.

```
typedef struct Polynomial
{
    int constant;
    int x-coefficient;
    int x-value;
    int x-degree;
    int y-coefficient;
    int y-value;
    int y-degree;
}Polynomial_LST;
Polynomial_LST polynomial_list[];
```

Fig. 5. Structure for Polynomial Key

Fig. 6와 같이 25개의 센서 노드가 5*5 그리드 좌표를 구성하는 경우, Fig.4에서 polynomial_list의 크기는 5*5 배열이 된다. 각 배열의 항목은 좌표의 X축과 Y축의 값과 매칭 된다. 즉, (0, 0)좌표는 배열 polynomial_list[0][0]이 된다.

(0,4)	(1,4)	(2,4)	(3,4)	(4,4)
(0,3)	(1,3)	(2,3)	(3,3)	(3,4)
(0,2)	(1,2)	(2,2)	(2,3)	(2,4)
(0,1)	(1,1)	(1,2)	(1,3)	(1,4)
(0,0)	(0,1)	(0,2)	(0,3)	(0,4)

Fig. 6. 5*5 Grid

Fig.7은 모니터링 시스템에서 사용하는 메시지 포맷이다. 키 분배에 사용하는 메시지 패킷은 명령어, 엔드 디바이스의 ID, 다항식을 구성하는 상수(Contant), X항 계수(X-coefficient), X항의 값(X-value), X항의 차수(X-degree), Y항의 계수(Y-coefficient), Y항의 값(Y-value), Y항의 차수(Y-degree)로 구성된다.

Setup ZC message format

Cmd.	CR
7bytes	0x0D

Key pre-distribute message format

STX	Deli.	Cmd.	ID	Sep
1byte	1byte	1byte	2bytes	1byte
Con.	X-coefficient	X-value	X-degree	Deli.
2bytes	2bytes	2bytes	2bytes	1bytes
Y-coefficient	Y-value	Y-degree	Deli.	EXT
2bytes	2bytes	2bytes	1byte	1byte

Request ZE_ID message format

STX	Deli.	Cmd.	Deli.	EXT
1byte	1byte	1byte	1byte	1byte

Response ZE_ID message format

STX	Deli.	Cmd.	ID	Data	Ext
1byte	1byte	1byte	2bytes	2bytes	1byte

Sensing Data message format

STX	Deli.	Cmd.	ID	Data	EXT
1byte	1byte	1byte	2bytes	2bytes	1byte

(Deli.: Delimiter, Cmd : Command, ID : Device ID
Con. : Constant)

Fig. 7. Message Format

키 분배 도중 패킷 탈취로 인한 공격을 방지하기 위해 키 분배 패킷은 네트워크에 조인 할 때 동적으로 할당받는 네트워크 어드레스로 암호화하여 전송한다.

3. 공장 설비 모니터링 시스템 구현 및 실험

본 논문에서 제안한 시스템의 효율성 검증을 위해 Fig. 8와 같은 실험 환경을 구현하였다. 센서를 이차 그리드와 유사하게 배치하였고, 센서 보드와 지그비 트랜시버 모듈은 아두이노 보드에 각각 하나씩 연결된다. 지그비 코디네이터와 서버는 USB포트로 연결되고 아두노이 보드로부터 전송된 무선 데이터를 RF 트랜시버로 수신하여 시리얼 통신을 통해 서버 프로그램으로 데이터를 송수신 한다. 가상 USB시리얼 포트 드라이버를 지원하는 지그비 코디네이터는 서버 프로그램에 최대 10개가 연결될 수 있고, 하나의 코디네이터에 최대 250개의 엔드 디바이스가 연결될 수 있다.

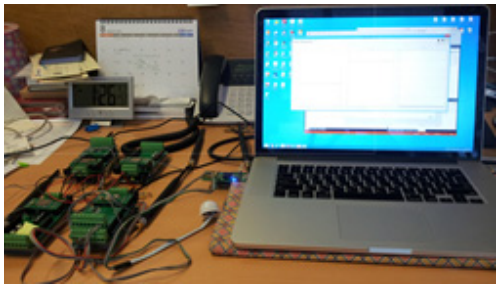

Fig. 8. Experiment Environment

서버에 전송된 데이터는 Fig. 9.과 같이 실시간으로 그래프로 표시되고, 알람 설정 값에 도달하면 시간과 데이터를 저장한다.

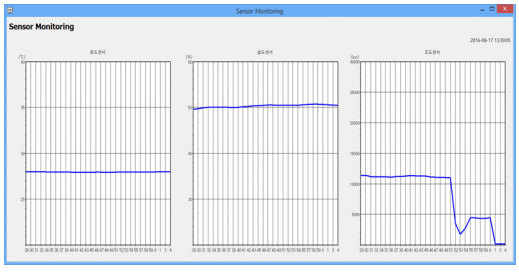

Fig. 9. Sensing Data Graph

다항식의 차수에 따른 연산 속도를 확인하기 위해서 다항식 연산을 하고 출력 포트를 토글 시킨 시간에서 다항식 연산을 수행하지 않고 출력 포트를 토글 시킨 시간을 빼면 다항식 연산 속도를 확인할 수 있다. Fig. 10은 출력 포트의 토글 시간을 확인하기 위한 아두이노 소스코드이다.

```
void setup(){
      pinMode(13, OUTPUT);
}

void compute_ToggleSpeed(){
      while(1){
      digitalWrite(13, HIGH);
      degitalWrite(13, LOW);
      }
}
```

Fig. 10. Arduino Simulation Code for Output Port Toggle

Fig. 11은 다항식 연산을 수행하고 출력 포트를 토글 시키는 아두이노 소스코드이다.

```
void setup(){
      pinMode(13, OUTPUT);
}

void compute_Polynomial(){
      long result;

      while(1){
      digitalWrite(13, HIGH);
      result = iCon + iX_co*pow(iX_val, iX-deg)
+ iY_co*pow(iY_val, iY_deg);
      digitalWrite(13, LOW);
       iCon + iX_co*pow(iX_val, iX-deg)
+ iY_co*pow(iY_val, iY_deg);
      }
}
```

Fig. 11. Arduino Simulation Code for Polynomial Computation

Fig.10의 소스 코드를 실행한 출력 포트의 포트 토글 지연 시간의 스코프 파형은 Fig.12과 같고, 지연 시간은 142.6KHz이다. 실험에서는 상수, X 계수, X 값, Y 계수 그리고 Y 값 고정으로 하고, X 차수와 Y 차수를 변경하여 연산 속도를 측정하였다.

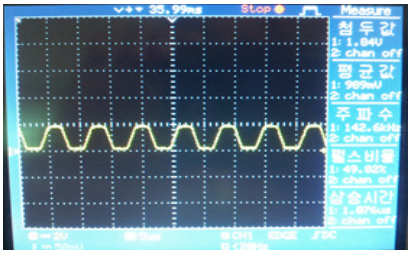

Fig. 12. Waveform of Output Port

사전에 분배받은 long형의 다항식의 키를 쉬프트 연산자를 사용하여 4바이트의 unsigned char 데이터 키를 생성한다. 데이터 키와 전송하려는 메시지 패킷을 XOR 연산을 수행하여 전송 버퍼에 넣는다. 데이터 키 생성 알고리즘은 Fig. 13과 같다.

```
#define MSG_LENGTH 8
#define KEY_LENGTH 4

void get_key(long poly_key, unsigned char* key)
{
poly_key = icon + iX_co*pow(iX_val, iX_deg) *
iY_co*pow(iY_val, iY_deg);
key[0] = poly_key >> 24;
key[1] = poly_key >> 16;
key[2] = poly_key >> 8;
key[3] = poly_key;
}

void encrypt_MSG(char* sendBuf, char* msg, char
msgLength){
        int j;
        get_key();

        j = 0;
        for(int i =0; i < msgLength; i++){
                if( j >= KEY_LENNGTH) j =0;
                else j++;
                sendBuf[i] = msg[i] ^ key[i];
        }
}
```

Fig. 13. Arduino Source code for Encryption Key Generation

다항식 차수에 따른 연산 속도 그래프는 Fig. 14와 같다. Cortex-M3 프로세서를 사용하고 SRAM사이즈 96KB 클럭 사이클이 84Mhz인 아두이노 DUE에서 X와 Y의 값의 범위를 0에서 10, X와 Y의 차수를 0에서 1000까지 변경하여 실행 속도를 측정하였다. X와 Y의 값이 2인 경우, 실행 속도는 6.97us로 차수의 변화에 차이가 거의 없었다. X와 Y의 값이 10이고 차수가 1000인 경우의 실행 속도는 491.88us였다.

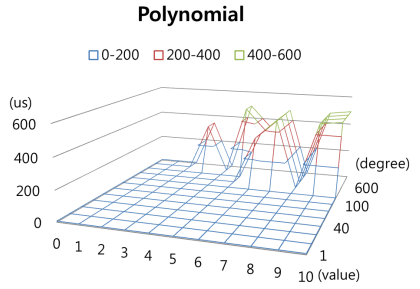

Fig. 14. Graph of Polynomial Computation

메시지 암호화 수행 시간 그래프는 Fig. 15와 같다. 데이터 메시지 패킷 사이즈가 8바이트로 암호화에 걸리는 시간은 8.75us, 2048바이트를 암호화 하는데 걸리는 시간은 1.08ms이다.

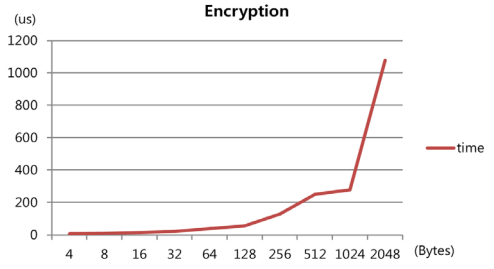

Fig. 15. Graph of Encryption

4. 결론

본 논문에서는 동적으로 생성된 다항식 키를 공장의 설비 배치와 유사한 그리드 기반 좌표를 사용하여 선분배하는 방식을 구현하였다. 아두이노 보드에 조립하여 사용할 수 있는 확장 보드를 설계하여 센서의 확장이 편리하다. 또한 확장 포트를 통해 유선 및 와이파이와 같은 이기종의 무선 통신 기능 확장이 가능하다.

공장 설비 배치와 유사한 그리드 기반 좌표를 이용하여 키 관리가 편리하며, 엔드 디바이스가 지그비 네트워크에 조인할 때마다 해당 좌표의 키를 동적으로 생성하여 분배하므로 키 탈취 공격을 예방할 수 있다. 어플리케

이션 레벨에서 모니터링 데이터 패킷을 선 분배된 키로 생성한 데이터 키와 동적 네트워크 주소로 이중 암호화 하므로 보다 안전하게 메시지를 전송할 수 있다.

Fig.14에서와 같이 X와 Y항의 값이 2인 경우 다항식 연산에 걸리는 7us 미만이고, Fig.15에서와 같이 8바이트 사이즈의 암호화에 걸리는 시간은 9us 미만으로 다항식 연산과 암호화 수행으로 인한 오버헤드가 크지 않다. 센서 노드의 성능에 맞춰 다항식 차수의 범위를 조절할 수 있도록 설계되어 의료, 군사, 산업 자동화 등 다양한 무선 센서 네트워크 환경에 적용 가능할 것으로 기대된다.

References

[1] Paolo Paolo Baronti, Prashant Pillai, Vince Chook, Stefano Chessa, Alberto Gotta, Y.Fun Hu, 'Wireless Sensor Networks: A Survey on the state of the art and the 802.15.4 and ZibBee Standards', Computer communications, vol. 30, no. 7, pp. 1655-1695, 2007. DOI: http://dx.doi.org/10.1016/j.comcom.2006.12.020

[2] Hyunjue Kim and Jong-Moon Chung., "USN Security Enhancement Using System IDs", Journal of the Institute of Electronics Engineers of Korea., vol. 46, no. 2, pp. 73-80, 2009.

[3] Hosein Marzi, Arash Marzi, "A Security Model for Wireless Sensor Networks", Computational Intelligence and Virtual Environments for Measurement System and Applications (CIVEMSA), 2014 IEEE International Conference on. IEEE, 2014. DOI: http://dx.doi.org/10.1109/civemsa.2014.6841440

[4] https://www.arduino.cc/en/Main/ArduinoBoardDue

[5] http://www.sensirion.co.kr/en/products/humidity-temperature/humidity-sensor-sht71/

[6] http://rohmfs.rohm.com/en/products/databook/datasheet/ic/sensor/light/bh1750fvi-e.pdf

[7] http://www.trossenrobotics.com/productdocs/2010-10-26-DataSheet-FSR402-Layout2.pdf

[8] http://www.firmtech.co.kr/default/img/eng/manual/zigmodule/RP-M100_Manual(Stack%20&%20Mac)_ver0.1(Eng).pdf

조 양 희(Yang-Hee Cho) **[정회원]**

- 2011년 2월 : 숭실대학교 컴퓨터학과 (공학석사)
- 2014년 2월 : 숭실대학교 컴퓨터학과 박사 수료

<관심분야>
무선 센서 네트워크, 보안, 원격 모니터링 시스템

박 재 표(Jae-Pyo Park) **[종신회원]**

- 1998년 2월 : 숭실대학교 컴퓨터학과 (공학석사)
- 2004년 2월 : 숭실대학교 컴퓨터학과 (공학박사)
- 2008년 3월 ~ 2009년 2월 : 숭실대학교 정보미디어 기술연구소 전임연구원
- 2010년 3월 ~ 현재 : 숭실대학교 정보과학대학원 교수

<관심분야>
컴퓨터 통신, 보안, 암호학, 멀티미디어 통신

양 승 민(Seung-Min Yang) **[정회원]**

- 1983년 2월 : Univ. of South Florida. 전산학과 (공학석사)
- 1986년 8월 : Univ. of South Florida. 전산학과 (공학박사)
- 1987년 3월 ~ 1991년 2월 : Univ. of Texas at Arlington. 조교수
- 1993년 3월 ~ 현재 : 숭실대학교 컴퓨터학부 교수

<관심분야>
실시간 시스템, 시스템 결함 허용, 센서 네트워크, 운영체제.