

## WSN 기반 야생 동물 침입 탐지 시스템 구현

이 형 봉<sup>1</sup> · 권 기 현<sup>2\*</sup><sup>1</sup>강릉원주대학교 컴퓨터공학과 교수<sup>2</sup>\*강원대학교 전자정보통신공학부 교수

# Implementation of a WSN-based Wild Animal Intrusion Detection System

Hyung-Bong Lee<sup>1</sup> · Ki-Hyeon Kwon<sup>2\*</sup><sup>1</sup>Professor, Department of Computer Science & Engineering, Gangneung-Wonju National University, Wonju 26403, Korea<sup>2</sup>\*Professor, Department of Electronics, Information & Communication Engineering, Kangwon National University, Samcheok 25913, Korea

### [요 약]

멧돼지, 고라니, 노루 등 야생동물에 의한 농작물 피해를 막기 위해 ICT를 기반으로 하는 다양한 방법들이 제안되고 있으나 대부분 실용성이 떨어지거나 적용 범위가 제한적이다. 이로 인해 농촌의 현장에서는 그물망이나 차양막을 이용하는 전통적 울타리로 농작물을 보호한다. 전통적 울타리 방법은 설치가 용이하고 견고하다는 장점이 있으나 멧돼지와 같은 강력한 힘을 가진 동물에 의한 국부적인 훼손 및 침입을 즉시 알 수 없다는 단점이 있다. 이 연구에서는 WSN을 기반으로 전통적 울타리의 파손이나 침입을 탐지하여 즉시 보고할 수 있는 야생동물 침입 탐지 시스템을 구현한다. 구현된 시스템은 이전에 제안된 방법들보다 구조가 간단하고 비용이 저렴하며 설치가 쉽다.

### [Abstract]

Various ICT-based devices have been proposed to prevent damage to crops caused by wild animals such as boar, elk, and roe deer, but most of them are not practical or have a limited scope of application. For this reason, in rural areas, farmers protect crops with traditional fences using nets or awnings. The traditional fence has the advantage of being easy to install and robust, but has the disadvantage that local damage and invasion by animals with strong power such as boar can not be immediately detected. In this study, we implement a wild animal intrusion detection system capable of detecting and reporting damage or intrusion of a traditional fence based on WSN. The implemented system has a simpler structure, lower cost, and easier installation than previously proposed methods.

**색인어** : 농작물 보호 울타리, 침입 탐지, 야생 동물, 무선 센서 네트워크**Keyword** : Crop protection fence, Intrusion detection, Wild animal, WSN(wireless sensor network)<http://dx.doi.org/10.9728/dcs.2021.23.1.149>

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Received** 04 November 2021; **Revised** 17 December 2021**Accepted** 17 January 2022**\*Corresponding Author; Ki-Hyeon Kwon****Tel:** +82-33-570-6400**E-mail:** [kweon@kangwon.ac.kr](mailto:kweon@kangwon.ac.kr)

## I. 서론

최근 인간의 개발 범위 확대에 의한 산지 등 서식처를 잃은 야생동물들이 농지에 접근하여 농작물을 해치는 사례가 급격하게 증가하고 있다는 연구 보고들이 많다[1, 2]. 이들 연구에 의하면 멧돼지, 고라니 등이 서식처 주변 주민들에게 매년 100억 이상의 농작물 피해를 주고 있다. 유해 야생동물 중에서도 멧돼지에 의한 피해가 가장 심하여, 농민들은 수입 감소를 무릅쓰고 재배 작물을 아예 멧돼지가 싫어하는 종류로 변경하기도 한다[3]. 특히 멧돼지의 경우 서식지 감소와는 다르게 그 개체 수가 크게 증가하는 추세이다. 이와 같은 야생동물에 의한 피해를 줄이거나 예방하기 방법으로 포획에 의한 개체 수 조절, 농작물 보호 울타리 설치, 동물별 기피제 살포, 서식처 먹이 주기 등의 활동이 이루어지고 있으나 그 효과가 제한적이라는 상황에 처해 있다.

ASF(african swine fever: 아프리카 돼지 열병) 바이러스에 감염된 DMZ 서식 멧돼지는 농작물 피해보다는 그 규모를 비교할 수 없을 만큼 소나 돼지 사육장에 치명적인 피해를 준다. 이를 예방하기 위해 ASF 바이러스에 감염된 멧돼지 사체가 발견된 지역의 사육장 근처에는 2중 3중의 울타리를 설치하거나 긴급 수렵을 허용한다[4]. 또한 DMZ 멧돼지의 남하를 감지하기 위해 드론을 활용하기도 한다[5].

현재 농촌에서 이루어지는 유해 야생동물 차단 방법은 대부분 그물망이나 차양막을 사용하여 설치하는 전통적인 울타리인데, 편리성과 효과 제고를 위해 다양한 ICT를 적용한 방법들이 제안되어 있다. 그러나 이러한 제안된 방법들이 안정성, 운영성, 관리성, 비용 등의 측면에서 만족스럽지 않아 실제 활용되는 경우는 많지 않다[6]. 이 논문에서는 농촌의 전통적인 울타리를 보조하여 강력한 힘을 가진 멧돼지 등에 의한 울타리의 파손이나 침입을 실시간으로 탐지하여 보고할 수 있는 WSN 기반 유해 동물 탐지 시스템을 제안하고 구현한다. 제안 시스템은 일렬로 배치된 여러 개의 센서 노드들을 선형 라우팅 경로로 연결하여, 별도의 특별한 센서 대신 울타리 사이를 연결하는 도선의 단락 체크로 침입을 간단하게 탐지할 수 있다.

이 논문은 2장의 관련 연구, 3장의 시스템 구현, 4장의 시스템 평가, 그리고 5장의 결론으로 구성된다.

## II. 야생동물 차단 방법 관련 연구

유해 야생동물 차단 방법은 크게 전통적인 방법과 ICT를 기반으로 하는 새로운 방법 등 두 가지로 나눌 수 있다.

### 2-1 전통적인 차단 방법

멧돼지, 고라니, 노루 등 비교적 몸집이 큰 야생동물로부터 농작물을 보호하기 위한 가장 일반적이고 전통적인 방법은

그림 1과 같이 그물망이나 차양막을 이용하는 울타리이다. 그물망은 일반 농가에서 노루망이라 부르기도 한다. ASF 바이러스에 감염된 DMZ의 멧돼지 남하를 차단하기 위해서는 그림 2와 같이 철제망, 일명 염소망으로 울타리를 설치한다. 이러한 전통적인 방법은 설치 과정이 어렵지 않고, 일단 설치되면 비교적 광범위한 영역을 보호할 수 있으며, 운영 및 관리상 기술적 어려움이 적다는 장점이 있다. 그러나, 야생동물의 국부적인 공격에 의해 부분적으로 구멍이 나거나 망가졌을 때 그 사실을 즉시 알 수 없다는 단점이 있다. 그 밖에 그림 2의 전기 울타리를 설치하는 경우도 있으나 사람의 감전 사고나 화재 위험성이 커 그 활용 빈도가 극히 낮다[7].



그림 1. 일반적인 야생동물 차단 울타리  
Fig. 1. Common Wild Animal Barrier Fence

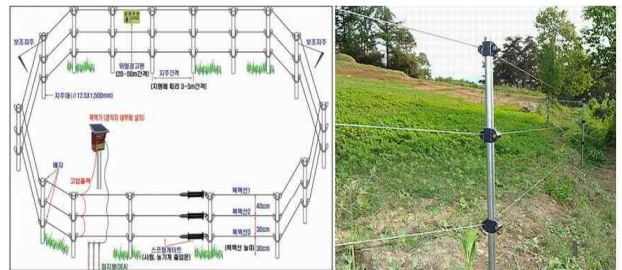


그림 2. 야생동물 차단용 전기 울타리  
Fig. 2. Electric Fence for Wild Animal Barrier

### 2-2 ICT 기반 탐지 및 차단

무선 통신, 인공지능 등 ICT의 발전에 힘입어 그 기술을 농작물 보호에 적용하려는 시도들이 이루어지고 있다. 연구 [8]은 스프링 구조의 발판 형태로 구성된 리미터 스위치를 이용하여 야생동물이 스위치를 밟았을 때 발생하는 인터럽트를 침입 탐지 신호로 삼아 해당 id를 ZigBee 등을 이용하여 서버로 전송하면, 서버에서 가장 가까운 위치의 LED 램프나 스피커와 통신하여 동작하도록 한다. 이 시스템의 단점은 탐지 영역이 리미터 스위치가 설치된 곳으로만 한정되고, 통신 방식이 수 백m 이상의 원거리 환경에는 적합하지 않다는 점이다. 연구 [9]는 자신의 위치를 알리는 비콘에 가속도 센서를 부착하여 주변에서 움직임이 감지될 경우 이를 야생동물의 침입 신호로 삼아 그 사실을 알리도록 한다. 이 연구는 사용 장치가 간단하고 설치가 용이하다는 장점이 있는 반면에 탐지된 움직임이 동물에 의한 것인지 바람 등 자연현상에 의

한 것인지에 대한 식별이 곤란하여 오류 보고 빈도가 높고, 탐지 대상 영역이 넓은 경우 2차적인 통신 선로 구비가 필요하다는 단점이 있다.

연구 [6]과 [10]은 객체의 종류와 위치를 추측하는 YOLO(You Only Look Once) 알고리즘을 이용하여 웹캠으로 캡처한 이미지 내에 야생동물이 존재하는가를 인공 지능 기법으로 판별하여 침입을 탐지한다. 이 연구는 최신 기술의 적용을 시도했다는 점에서 의의가 있으나, 이미지를 위한 대용량 전송 통신 및 이미지 처리를 위한 서버급 컴퓨터가 필요하고, 분석 결과의 오류 확률이 있으며, 웹캠의 시야를 벗어난 영역은 탐지할 수 없다는 단점이 있다. 연구 [11]은 특별히 사람에 의한 도난과 야생동물의 침입을 구별하기 위해 마그네틱 센서, 위치 변위 센서, 리미트 센서를 곳곳에 설치하여 세 가지 센서에 의한 탐지 패턴을 분석하는 방안을 제안하였다.

### Ⅲ. WSN 기반 경량 야생동물 침입 탐지 시스템

#### 3-1 제안 시스템 개관

이 연구에서 제안하는 시스템은 그림 1의 전통적인 울타리를 다음과 같이 보조하여 단점을 보완하는 데 주안점을 둔다.

- 그림 3과 같이 울타리의 옆면을 따라 노끈이나 고추 끈 등 얇은 로프를 연결하여 설치한다.
- 안정적인 무선 통신 가능 거리마다 단락 체크를 위한 지주를 세우고, 이곳에 무선 통신 센서 노드를 배치한다.
- 멧돼지 등의 강력한 힘에 의해 울타리가 밀려 파손되는 순간 로프에 연결된 지주가 기울어 센서 노드와 연결된 센싱 케이블이 단선되도록 한다.
- 센싱 케이블의 단선을 인지한 센서 노드는 현재 위치를 의미하는 id를 싱크 노드로 전송한다.

제안 시스템의 구현 모듈은 효과적인 무선 통신 방법과 단선 체크 방법 등 크게 두 가지로 요약할 수 있다. 게이트웨이나 모니터링 UI 등은 일반화되어 있으므로 이 연구에서는 다루지 않는다.



그림 3. 제안 시스템 개관도  
Fig. 3. Overview of the Proposed System

구현 시스템의 하드웨어로는 그림 4와 같이 ATmega 2560 MCU[12]의 패키지(이하 MCU)에 CC1100 RF 트랜시버[13] 패키지를 탑재한 프로토타입 센서 노드를 사용한다. 센서 노드의 사양은 표 1과 같고, CC1100의 주요 인터페이스 핀에 대한 MCU 핀 연결 및 핀 기능은 그림 5와 같다. MCU와 CC1100은 CS/SI/SO 핀을 이용하는 SPI(serial peripheral interface) 표준으로 통신한다. 즉, 마스터 MCU는 슬레이브 CC1100에 대하여 SPI 통신으로 명령어를 보내고 응답을 받는다. SPI 통신은 마스터와 슬레이브 사이에서 8비트 단위의 환형 맞교환으로 진행되므로, 특정 명령어에 대한 응답을 받기 위해서는 두 번의 통신이 필요하다. CC1100은 1차 명령어에 대하여 내부 상태를 돌려주도록 약속되어 있고, MCU는 설정에 따라 1 바이트 전송이 완료되었을 때마다 인터럽트 플래그를 표시해준다. 그림 6에 AVRStudio 4.19 개발 환경에서 구현된 MCU의 CC1100 접근을 위한 기본 프로시저 뼈대를 보였고(이하 모든 프로시저의 구현 환경 동일) 이때 매개 변수 addr은 CC1100의 내부 레지스터 번호나 버퍼의 위치를 지시한다.

표 1. 센서 노드 사양

Table 1. Specifications of the Sensor Node

Items	Specifications	
MCU	CPU	ATmega2560(7.3728 MHz, 30PPM)
	Flash	256KB
	SRAM	8KB
	Energy Cons.	active:8mA, sleep:15uA
	Ext. Memory	128KB
RF Transceiver	Model	CC1100(800~928MHz, 500Kbps)
	Antenna type	1/4 mono pole
Power	2 AA type lithium batteries(3.6V)	

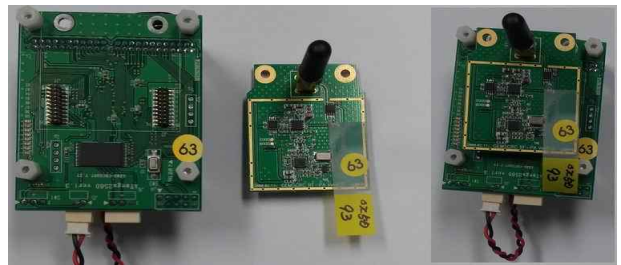


그림 4. 센서 노드 모습  
Fig. 4. Photo of the Sensor Node

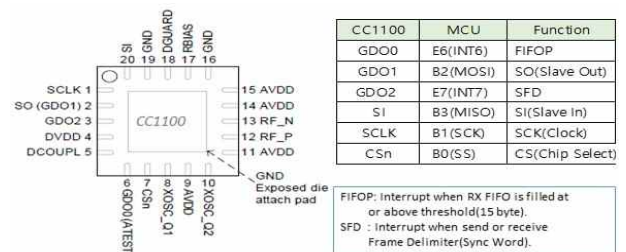


그림 5. CC1100 RF 트랜시버의 노출 핀  
Fig. 5. Exposed Pin of CC1100 RF Transceiver

```

uint8_t rf_cmd(uint8_t cmd) {
    outp(cmd, SPDR) // send command to SPI
    while (!(SPSR & SPIF)); // wait for interrupt flag
    return(inp(SPDR)); // return response byte
}
uint8_t rf_read(uint8_t addr) { // read 1 byte
    uint8_t data;
    cbi(PORTB, 0); // set CS to low(start SPI)
    rf_cmd_without_cs((addr | CC1100_R_SINGLE));
    data = rf_cmd(0); // send dummy to get data
    sbi(PORTB, 0); // set CS to high(stop SPI)
    return(data);
}
// read many bytes
uint8_t rf_read_burst(uint8_t addr,
    uint8_t buf[], uint8_t count) {
    cbi(PORTB, 0); // set CS to low(start SPI)
    rf_cmd((addr | CC1100_R_BURST));
    for (i = 0; i < count; i++)
        buf[i] = rf_cmd(0); // send dummy to get data
    sbi(PORTB, 0); // set CS to high(stop SPI)
    return(SUCCESS);
}
// write 1 byte
uint8_t rf_write(uint8_t addr, uint8_t data) {
    cbi(PORTB, 0); // set CS to low(start SPI)
    rf_cmd((addr | CC1100_W_SINGLE));
    rf_cmd(data); // send data
    sbi(PORTB, 0); // set CS to high(stop SPI)
    return(data);
}
// write many bytes
uint8_t rf_write_burst(uint8_t addr,
    uint8_t buf[], uint8_t count) {
    cbi(PORTB, 0); // set CS to low(start SPI)
    rf_cmd((addr | CC1100_W_BURST));
    for (i = 0; i < count; i++)
        rf_cmd(buf[i]); // send data
    sbi(PORTB, 0); // set CS to high(stop SPI)
    return(SUCCESS);
}
uint8_t rf_status() { // get CC1100 status
    cbi(PORTB, 0); // set CS to low(start SPI)
    status = rf_cmd(CC1100_SNOP);
    sbi(PORTB, 0); // set CS to high(stop SPI)
    return(status);
}
    
```

그림 6. CC1100 접근을 위한 기본 프로시저  
**Fig. 6. Primitive Procedure to Access CC1100**

**3-2 RF 프레임 송수신**

MCU가 프레임을 송신하기 위해서는 SPI 통신으로 그림 7의 MCU 프레임 포맷 부분을 CC1100의 TX 버퍼에 채운 후, 송신 명령어를 보낸다. 이때, CC1100의 상태는 수신 혹은 유힬 상태여야 하고, CCA(clear channel assessment)가 성공하면 그림 7의 RF 패킷 송신이 시작된다. 그림 8에 MCU의 패킷 송신 프로시저를 보았다. 프레임의 수신은 CC1100이 RF 패킷을 수신하여 RX 버퍼에 임계치 이상 채우고 그림 9의 FIFOP 인터럽트인 INT6으로 MCU에게 알리면, MCU가

인터럽트 핸들러에서 SPI 통신으로 읽어오으로써 이루어진다. 그림 9에 MCU 프레임을 수신하는 프로시저를 보였는데, 읽기 언더플로우가 발생하지 않도록 구현하였고, RX 버퍼의 마지막 두 바이트는 RSSI와 LQI이다.

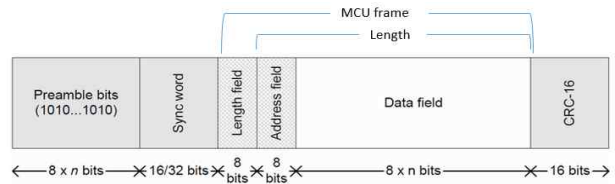


그림 7. CC1100 패킷 포맷  
**Fig. 7. Packet Format of CC1100**

```

uint8_t rf_tx_frame(uint8_t frame[]) {
    uint8_t status;
    rf_write_burst(CC1100_TXFIFO, frame, frame[0] + 1);
    cbi(PORTB, 0); // set CS to low(start SPI)
    status = rf_cmd(CC1100_STX); // TX if CCA
    sbi(PORTB, 0); // set CS to high(stop SPI)
    return(status);
}
    
```

그림 8 MCU 프레임 송신 프로시저  
**Fig. 8. MCU Frame Transmit Procedure**

```

ISR(INT_6_vect) {
    G_rx_len = rf_rx_frame(G_rx_frame);
    G_event |= EVENT_FIFOP;
}
uint8_t rf_rx_frame(uint8_t frame[]) {
    uint8_t framele, clen;
    cbi(PORTB, 0); // set CS to low(start SPI)
    clen = rf_cmd(CC1100_RXFIFO | CC1100_R_BURST);
    clen &= 0x0f; // current len of RX FIFO
    framelen = frame[0] = rf_cmd(0);
    for (clen--; clen = 0; ; ) {
        for (i=0; i < clen-2 && len < framelen; i++)
            frame[++len] = rf_cmd(0);
        if (len == framelen) break;
        sbi(PORTB, 0); // stop read burst mode
        cbi(PORTB, 0); // start again SPI
        clen = rf_cmd(CC1100_RXFIFO | CC1100_R_BURST);
        clen &= 0x0f; // current len of RX FIFO
    }
    frame[++len] = rf_cmd(0); // RSSI
    frame[++len] = rf_cmd(0); // LQI
    sbi(PORTB, 0); // stop burst read mode
    cbi(PORTB, 0); // start again SPI
    rf_cmd(CC1100_SFRX); // flush RX FIFO
    rf_cmd(CC1100_RX); // set RX enable
    return(framelen);
}
    
```

그림 9 MCU 프레임 수신 프로시저  
**Fig. 9. MCU Frame Receive Procedure**

**3-3 무선 통신 프로토콜**



일반적인 무선 센서 네트워크 개념은 그림 10과 같이 임의의 위치에 흩어진 다수의 센서 노드들이 서로 협조하여 센싱 데이터를 싱크 노드 방향으로 전달하도록 한다[14]. 이를 위한 대표적인 전송 프로토콜인 AODV[15]는 플러딩으로 라우팅 정보를 얻기 때문에 전송 과정이 느리고 에너지 소모가 크다. 그 외에 그림 11의 다양한 링크 토폴로지들을 구성하여 데이터를 전송하는 프로토콜들이 알려져 있다. 그런데, 그림 11의 토폴로지들은 전송 경로가 장거리 선형인 이 연구의 형태에 적합하지 않다. 따라서 이 연구에서는 [16]-[17]에서 제안된 시분할 기반 선형 토폴로지 개념에 그림 12와 같이 센서 노드를 id 순서에 따라 일렬로 배치한다는 조건을 부여함으로써 인위적 선형 토폴로지를 이루는 프로토콜 WLSP (waterfall-styled linear sensor network protocol)을 구현하여 활용한다. WLSP는 그림 13과 같이 id 순서에 따라 단말 노드부터 폭포수처럼 차례로 자신에게 지정된 활동 시간에만 깨어나 센싱 및 전달 활동을 하므로 에너지 소모가 적고 전송 지연이 짧다.

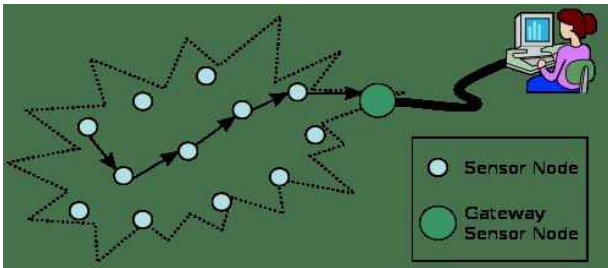


그림 10. 일반적인 무선 센서 네트워크 개념  
Fig. 10. General Concept of Wireless Sensor Network

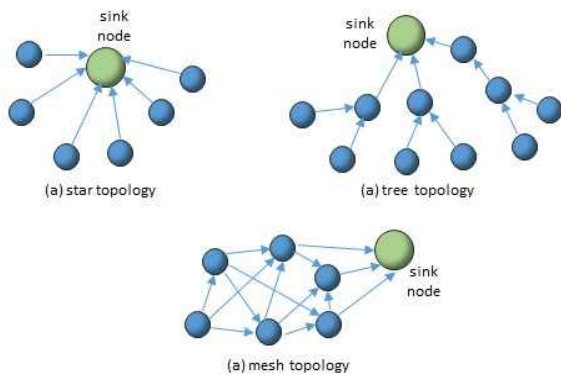


그림 11. 무선 센서 네트워크의 전형적인 토폴로지  
Fig. 11. Typical Topologies of Wireless Sensor Network



그림 12. 무선 센서 네트워크의 선형 토폴로지  
Fig. 12. Linear Topology of Wireless Sensor Network

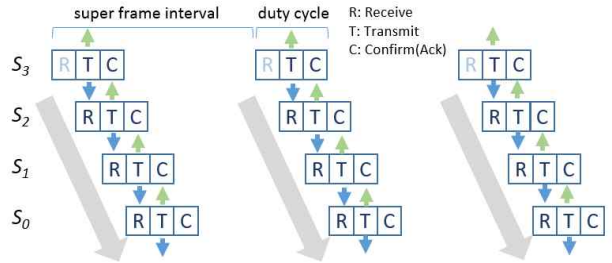


그림 13. WLSP 개념  
Fig. 13. Concept of WLSP

1) 타임 유닛

WLSP 센서 노드의 의무 사이클은 그림 13, 그림 14와 같이 수신(R), 송신(T), 확인(C) 등 크게 세 개의 타임 유닛으로 구성된다. 이 중 확인 구간에서는 다음 수신 노드가 정상적으로 활동하는지를 판단하고 필요한 경우 응답을 포함한다. 이들 타임 유닛은 그림 14와 같이 준비(ready), 송수신(rx/tx), 처리(process) 구간으로 세분화되어 시간 오차 및 동기화에 대비한다.

2) 슈퍼 프레임 동기화

두 노드 사이의 슈퍼프레임 동기화는 노드  $S_{j+1}$ 의 송신 패킷의 첫 부분이 노드  $S_j$ 에 도착하는 순간을 기준으로, 남아있는 슈퍼프레임 잔여기간  $sfi'$ 을 타이머에 설정함으로써 이룰 수 있다. RF 전송 시간을 무시한다면  $sfi'$  값은 명백하게  $sfi - t_0$ 의 수식으로 구할 수 있다. CC1100은 그림 7의 패킷 포맷 중 Sync word의 수신을 시작하는 순간 그림 5의 SFD 인터럽트인 INT7로 MCU에게 알려 동기화를 가능하게 한다. 그림 15에 SFD와 슈퍼프레임 타이머 인터럽트 핸들러 뼈대를 보였다. 확인 구간까지 마친 센서 노드는 슈퍼프레임 타이머가 만료될 때까지 절전 상태에서 기다린다.

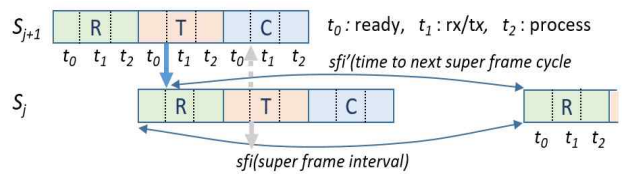


그림 14. WLSP의 슈퍼프레임 동기화  
Fig. 14. Super Frame Synchronization of WLSP

```
ISR(INT_7_vect) {
    set_sfi_timer(sfi - t_0); // set sfi timer
    G_event |= EVENT_SFD;
}
ISR(TIMER3_OVF_vect) {
    G_event |= EVENT_SFI;
}
```

그림 15. 동기화를 위한 SFD와 슈퍼프레임 타이머 인터럽트 핸들러  
Fig. 15. SFD and Super Frame Timer Interrupt Handler for Synchronization

### 3) 노드 참여

WLSP의 네트워크 토폴로지는 아래의 과정을 거쳐 형성된다.

- id 0인 싱크 노드가 슈퍼프레임 주기에 따라 의무 사이클의 T 타임 유니트에서 RF 프레임 송신한다.
- id 1인 노드가 id 0인 싱크 노드의 RF 송신 프레임을 스캔하여 자신의 슈퍼프레임 주기를 동기화한 후, 해당 구간에서 RF 프레임을 송신한다.
- 그림 14에서  $S_j$ 의 다음 노드  $S_{j+1}$ 가 WLSP에 참여하기 위한 슈퍼프레임 잔여 기간  $sfi$ 는 C 타임 유니트에서  $S_j$ 의 RF 프레임을 수신하는 순간을 기준으로  $sfi - (t_0 + t_1 + t_2 + t_0 + t_1 + t_2 + t_0)$ 의 식으로 얻을 수 있다.

### 3-4 침입 탐지 장치

본 제안 시스템에서는 침입 탐지를 위한 별도의 센서나 장치를 부착하지 않고, 그림 3과 같이 울타리를 따라 설치한 로프의 장력에 의해 로프의 이음 부분에서 발생하는 단선을 신호로 하여 침입을 탐지한다. MCU는 외부에 노출된 INT0 핀에 대하여 그 입력 값의 변화를 이벤트로 하는 인터럽트를 지원한다. 원하는 이벤트는 설정을 통해 상승(0->1: rising) 혹은 하강(1->0: falling) 변화 중 선택할 수 있다. 만약 다른 핀 INT1을 MCU 외부에서 도선으로 루프백시키고, MCU 내부에서 INT1 핀의 값을 변화시키면 그 변화가 INT0 핀의 입력으로 전달되어 인터럽트가 발생한다. 반대로 도선이 단절되어 있다면 인터럽트는 발생하지 않는다. 이 메커니즘을 이용하여 MCU는 외부 루프백 도선의 단절 여부를 간단하게 체크할 수 있다. 이를 위한 설정 및 체크 절차는 아래와 같고, 그 구현 프로시저는 그림 17과 같다.

- INT0 핀은 입력, INT1 핀은 출력으로 설정한다.
- 그림 16과 같이 INT0와 INT1 핀을 외부에서 연결하여 INT1의 출력을 INT0로 루프백시킨다.
- INT1 핀의 출력 값을 변화시켜 INT0 인터럽트가 발생하도록 한다.
- MCU는 INT1 핀의 출력값에 상승 변화를 준다.
- 외부 연결이 정상적인 경우에는 인터럽트가 발생하고, 단절되어 있는 경우에는 인터럽트가 발생하지 않는다.
- 인터럽트 발생 여부에 따라 단절 여부를 판단한다.

INT1(PORTD, 1)      INT0(PORTD, 0)

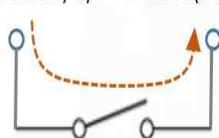


그림 16. MCU 핀을 이용하는 침입 탐지 메커니즘  
Fig. 16. Intrusion Detection Mechanism using MCU Pins

```
uint8_t check_fence() {
    sbi(DDRD, 1); cbi(DDRD, 0); // INT1 output, INT0 input
    sbi(EICRA, ISC01);
    sbi(EICRA, ISC00); // interrupt at INT0 rising edge
    G_event &= ~EVENT_FENCE;
    cbi(PORTD, 1); sbi(PORTD, 1); // make rising edge
    delay_003us();
    if (G_event & EVENT_FENCE) return(FENCE_OK);
    else return(FENCE_FAIL);
}
ISR(INT_0_vect) {
    G_event |= EVENT_FENCE;
}
```

그림 17. MCU 침입 탐지 프로시저  
Fig. 17. Intrusion Detection Procedure of MCU

## IV. 제안 시스템 시험 및 평가

### 4-1 시험을 위한 통합 알고리즘

III장에서 구현된 주요 프로시저를 통합한 센서 노드 펌웨어 알고리즘을 그림 18에 보였고, 주요 내용은 아래와 같다.

- 싱크 노드는 곧바로 슈퍼프레임 주기를 시작하고, 다른 노드들은 부모 노드를 스캔하여 자신의 최초 슈퍼프레임 잔여 시간을 설정한다.
- 노드들은 슈퍼프레임 주기로 깨어나, 아래의 단계를 진행한다.
  - R 유니트 타이머를 설정하고, RF를 준비한 후 R 유니트 동안 자식 노드로부터의 RF 패킷 수신을 기다린다.
  - 패킷 수신 시 SFD 인터럽트와 FIFOP 인터럽트가 시작될 두 차례로 발생하며, SFD 인터럽트 시점에서 자신의 슈퍼프레임 잔여 시간을 설정함으로써 시간 동기를 맞추고, FIFOP 인터럽트에서 RF 패킷을 수신한다.
  - 수신 패킷이 없는 경우 단말 노드가 된다.
  - R 유니트 종료 전 이벤트 발생 여부를 체크한다.
  - R 유니트가 끝나면, T 유니트의  $t_0$  만큼의 준비 시간을 기다린 후 부모 노드에게 RF 패킷을 송신한다.
  - 비 싱크 노드는  $t_0$  를 포함한 T 유니트 전체가 끝나면 C 유니트 동안 부모 노드에서의 송신 패킷을 엿들어 부모 노드의 정상 작동 여부를 확인한다
  - 싱크 노드는 게이트웨이에게 이벤트 여부 및 네트워크 상태를 보고한다.
- 만약 자식 노드로부터의 패킷 수신에 실패하면 depth를 1로 설정하여 자신이 단말 노드가 되며, 이후 노드들은 시차를 두고 차례로 초기 상태로 전환한다. 고장 노드가 복구되면 해당 노드부터 차례로 네트워크에 다시 참여한다.

```

main() {
    if (!lamSinkNode) // S0
        set_sfi_timer(sfi);
    else // Sj, j > 0
        set_rf_rx_mode();
        set_rf_pwr_on(); set_rf_rx_mode();
        while(1) {
            G_event = 0;
            while(!(G_event & EVENT_SFD));
            set_sfi_timer(sfi);
            while(!(G_event & EVENT_FIFOP));
            if (is_frame_of(Sj-1, G_rx_frame))
                break;
        }
        set_rf_pwr_down();
}

while(1) {
    while(!(G_event & EVENT_SF1))
        mcu_sleep();
    set_sfi_timer(sfi);
    set_sfp_timer(t0+t1+t2); // period of R unit
    set_rf_pwr_on(); set_rf_rx_mode();
    while(!(G_event & EVENT_SFD | EVENT_PHASE));
    if (G_event & EVENT_SFD) {
        set_sfi_timer(sfi); // time synchronization
        while(!(G_event & EVENT_FIFOP));
        if (is_frame_of(Sj+1, G_rx_frame))
            depth = depth_of(G_rx_frame) + 1;
        else error_restart();
    } else depth = 1; // terminal node

    fstat = check_fence();
    while(!(G_event & EVENT_PHASE)); // end of R unit
    set_sfp_timer(t0); // period of t0 of T unit
    pack_mcu_frame(fstat, depth, G_rx_frame, G_tx_frame);
    while(!(G_event & EVENT_PHASE)); // end of t0
    rf_tx_frame(G_tx_frame);
    set_sfp_timer(t1+t2); // period of T unit
    while(!(G_event & EVENT_PHASE)); // end of T unit

    if (!lamSinkNode) {
        set_sfp_timer(t0+t1+t2); // period of C unit
        while(!(G_event & EVENT_PHASE)); // end of C unit
        if (!(G_event & EVENT_FIFOP) ||
            !is_frame_of(Sj-1, G_rx_frame)) error_restart();
    }
    else send_report_to_gateway();
    set_rf_pwr_down();
}
}

```

그림 18. MCU 펌웨어 통합 알고리즘  
**Fig. 18. Integrated Algorithm of MCU Firmware**

#### 4-2 타임 유닛 설정

이 연구의 시험에서는 최대 RF 패킷 크기 64 바이트, CC1100의 상태 천이 시간, ATmega2560의 처리 속도, XTAL의 오차 한계 등을 고려하여 그림 14의 타임 유닛를 표 2와 같이 설정하였다. 다만, MCU 타이머의 카운터 속도가 ms 단위와 정확하게 일치하지 않으므로, 표 2에 보인 설정값과 실제 구현 환경 사이에는 미세한 시간차가 존재한다.

표 2. 타임 유닛 설정  
**Table 2. Configuration of Time Unit**

Items	Value
$t_0$	2.5ms
$t_1$	2.5ms
$t_2$	3.5ms
$sfi$	6s

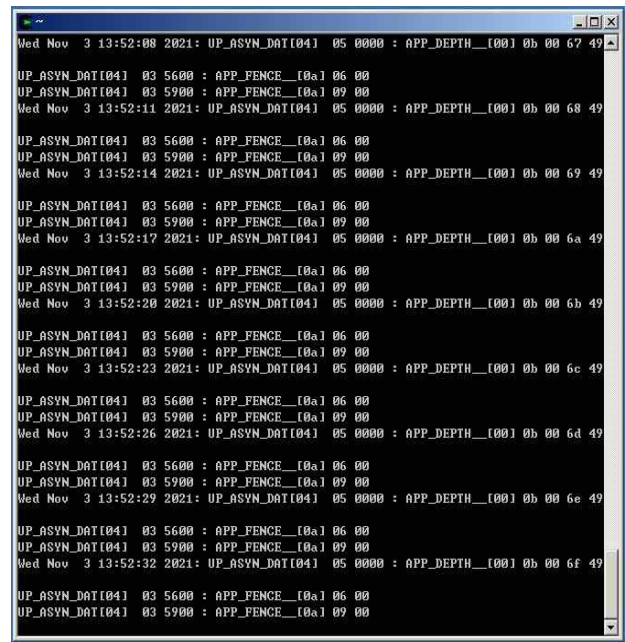


그림 20. 실험실 환경 노드 배치  
**Fig. 20. Node Deployment in Experimental Environment**

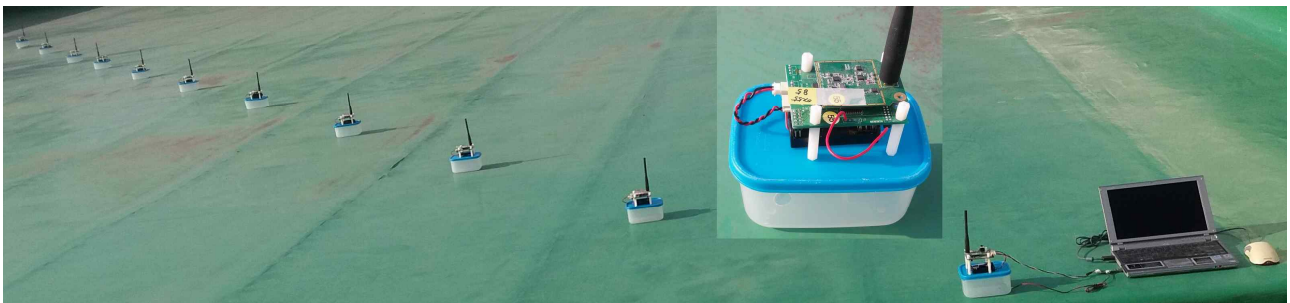


그림 19. 실험실 환경 노드 배치  
**Fig. 19 Node Deployment in Experimental Environment**





그림 21 필드 환경 노드 배치  
**Fig. 21** Node Deployment in Field Environment

#### 4-3 시험 환경 및 결과

##### 1) 실험실 환경 시험

구현 시스템의 신뢰성 및 기능 시험을 위해, 그림 19와 같이 싱크 노드를 포함하여 총 11개의 센서 노드들을 약 1.5m 간격으로 콘크리트 건물 옥상에 배치하여 그림 20의 콘솔 화면으로 14일 동안 관찰한 결과 네트워크 오류가 나타나지 않았고, 24시간마다 4회 실시한 모든 노드에서의 침입 탐지 기능이 정상적으로 작동하고 있음을 확인하였다. 그림 20은 PC에서 시리얼 통신을 통해 싱크 노드로부터 전달받은 WLSP의 패킷을 모니터링하는 콘솔 화면으로, 현재 참여 노드의 수인 DEPTH 11과 침입 탐지 발생 노드인 FENCE 06, 09를 보이고 있다.

##### 2) 필드 환경 시험

실제 현장 환경에서의 시험을 위해 실험실 환경에서 사용했던 11개의 센서 노드를 그림 21과 같이 40~50m 간격으로 실제 밭두렁 주변 약 450m에 걸쳐 배치하고 24시간 동안 관찰하였다. 관찰 결과 네트워크 오류가 나타나지 않았고, 시험 시작과 종료 시점에 실시한 모든 노드에서의 침입 탐지 실험에서 정확한 탐지 기능을 보였다.

#### V. 결론 및 향후 계획

농업의 기계화 및 자동화를 위한 제반 장치에 요구되는 최우선 요건은 설치와 운영이 용이해야 하고, 다음으로 비용이 크지 않아야 한다. 아무리 높은 수준의 최신 기술이 적용된 시스템이라 하더라도 설치 및 운영 과정에 고도의 전문가 개입이 필요하다면 그 활용은 제한적일 수밖에 없다. 이 연구에서는 구매 후 농민 스스로 설치와 운영이 가능할 정도로 간단한 WSN 기반 야생동물 침입 탐지 시스템을 구현하여 그 가능성을 검증하였다. 이번 연구에서는 침입 탐지 목적의 단방향 선형 WSN 프로토콜을 구현하였으나, 이후 연구에서는 상·하 양방향 통신을 번갈아 수행하는 양방향 WSN 프로토콜

을 구현하여 필요시 마이크 등 야생동물 퇴치 장치를 제어할 수 있는 시스템으로 확장할 예정이다.

#### 참고문헌

[1] S. H. Park, M. K. Kim, Y. C. Youn, "A Study on Mitigation and Adaptation Strategy to Wildlife Damage in Forest-dependent Rural Communities," *Journal of Environmental Policy and Administration Society*, Vol. 27, No. 2, pp. 11-37, June 2019. <http://dx.doi.org/10.15301/jepa.2019.27.2.11>

[2] S. H. Park, Y. C. Youn, "Estimating Economic Loss due to Wildlife Damage to Agriculture and Forestry Production Near Protected Areas: Case of Mountain Villages in Gurye-gun, Korea," *Journal of Korean Society of Forest Science*, Vol. 108, No. 4, pp. 618-627, 2019. <https://doi.org/10.14578/jkfs.2019.108.4.618>

[3] S. M. Lee, E. J. Lee, H. B. Park, C. W. Seo, "Factors affecting Crop Damage by the Wild Boar (*Sus scrofa*): A case study in Geochang County, Gyeongnam Province, Korea," *Korean Society of Environment and Ecology*, Vol. 32, No. 2, pp. 140-146, April 2018. <https://doi.org/10.13047/KJEE.2018.32.2.140>

[4] Y. S. Jo, C. Gortazar, *African swine fever in wild boar, South Korea, 2019*, Wiley Online Library, Wiley, 2020 [Internet]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1111/tbed.13532>.

[5] Maeil Business News Korea, *On the signs of the spread of ASF... Government uses Drones to search for wild boar*, 29 July 2021 [Internet]. Available: <https://www.mk.co.kr/news/economy/view/2021/07/732962/>.

[6] A. H. Lee, S. M. Park, J. S. Hong, "Development of a Yolo-Based System for Prevention of Wildlife Damage,"



- Proceedings of Korea Software Congress 2018*, pp. 1897-1899, December 2018.
- [7] C. S. Park, "Electric Shock Accident Cases and Safety Management of Electric Fence," *The Proceedings of the Korean Institute of Illuminating and Electrical Installation Engineers*, Vol. 24, No. 3, pp. 21-28, 2010.
- [8] J. H. Woo, "Design and Implementation of Farm Pest Animals Repelling System Based on Open Source," *Journal of Korea Multimedia Society*, Vol. 19, No. 2, pp. 451-459, February 2016.  
<http://dx.doi.org/10.9717/kmms.2016.19.2.451>
- [9] D. J. Jung, E. M. Jung, H. K. Kim, "A Study on the Prevention of Crops Damage Using Beacon System," *Journal of KIIT Conference*, pp. 307-310, June 2017.
- [10] Y. S. Ha, J. C. Shim. "Wild Animal Repellent System For Prevention of Crop Damage By Wild Boars," *Journal of Korea Multimedia Society*, Vol. 24, No. 2, pp. 215-221, February 2021.  
<https://doi.org/10.9717/kmms.2020.24.2.215>
- [11] T. G. Kang, S. H. Lee, J. G. Kim, et. al, "Anti-theft System for Agricultural Products which Discriminates between Humans and Wild Animals," *Proceedings of the KSAM & ARCs 2018 Autumn Conference*, pp. 98-98, 2018.
- [12] ATmega2560 [Internet]. Available: <https://www.microchip.com/en-us/product/ATmega2560>.
- [13] CC1100 [Internet]. Available: <https://www.ti.com/product/CC1100?keyMatch=CC1100>.
- [14] Sensor Networks [Internet]. Available: <http://en.wikipedia.org/wiki/File:WSN.svg>.
- [15] C. E. Perkins, E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," *Proceedings of the 2nd IEEE workshop on Mobile Computing Systems and Applications, New Orleans, LA*, pp. 90-100, February 1999.
- [16] D. S. Kim, S. Y. Lee, K. H. Won, D. J. Chung, and J. H. Kim, "Time-synchronized Forwarding Protocol for Remote Control of Home Appliances Based on Wireless Sensor Network," *IEEE Transactions Consumer Electronics*, Vol. 53, No. 4, pp. 1427-1433, 2007.
- [17] D. D. Caneva, and P. L. Montessoro, "A Synchronous and Deterministic MAC Protocol for Wireless Communications on Linear Topologies," *International Journal of Communications, Network and System Sciences*, pp. 925-933, 2010.



**이형봉(Hyung-Bong Lee)**

1984년 : 서울대학교 계산통계학과 (학사)  
 1986년 : 서울대학교 대학원 계산통계학과 (석사)  
 2000년 : 강원대학교 대학원 컴퓨터과학과 (박사)

1986년~1994년: LG전자 컴퓨터연구소

1994년~1999년: 한국디지털(주)

2004년~현 재: 강원원주대학교 컴퓨터공학과 교수

※ 관심분야 : 무선 통신 (Wireless Networks), 센서 네트워크 (Sensor Networks), 임베디드 시스템 (Embedded Systems), 사물 인터넷 (IoT)



**권기현(Ki-Hyeon Kwon)**

1993년 : 강원대학교 컴퓨터과학과 (학사)  
 1995년 : 강원대학교 대학원 컴퓨터과학과 (석사)  
 2000년 : 강원대학교 대학원 컴퓨터과학과 (박사)

1998년~2002년: 동원대학 인터넷정보과 교수

2002년~현 재: 강원대학교 전자정보통신공학부 교수

※ 관심분야 : 사물 인터넷 (IoT), 인공지능 응용