

# Microprocessor

- Homenetwork를 이용한 제어  
안전 솔루션

**김종현** 교수님

#1조

2011253070 정경주

2012253068 신민우

2014253060 이창재

# 목차

## 1. 주제 및 목표

- 주제 선정 이유

## 2. 구현기능 설명

## 3. 조별 일지

## 4. Flow Chart

## 5. 주요 함수 및 핵심 알고리즘

## 6. 토의

## 7. 고장 내역 및 없는 부품

## 8. 부록

# 1. 주제 및 목표

- 실험실 내부나 2차 화재의 위험이 있는 건물 내의 화재 발생 시를 가정하여 출입문(도어락)을 봉쇄 및 전원 차단, 전등 소등을 봉쇄 및 전원 차단, 전등 소등

## 주제 선정 이유

SK텔레콤이 건설 현장에 사물인터넷(IoT) 기술 기반의 종합 안전 솔루션을 국내 최초로 상용화했다.

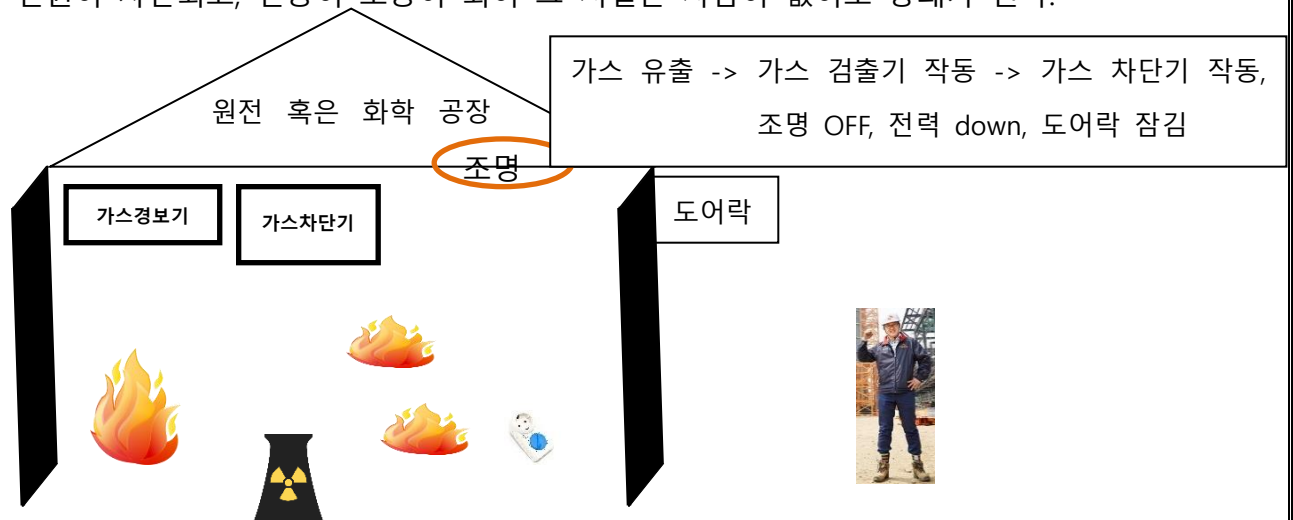
26일 SK텔레콤은 대우건설과 함께 위례신도시 우남역 푸르지오 건설현장에서 '스마트 건설' 솔루션을 시연했다.

스마트 건설은 IoT 네트워크 및 플랫폼을 기반으로 건설 현장 구조물에 지능형 CCTV, 가스·진동·화재 감지 센서 등을 설치해 사고 예방과 사고 발생 시 즉각적 대응을 돕는다.

- 건물의 화재 발생 및 원전 등 안전 사고 발생 시 각종 연기 확산과 2차 피해를 예방하기 위한 시스템을 홈 오토메이션 키트로 구현 해보고자 한다

## 2. 구현기능 설명(부록에 최종 구현 사진 첨부)

- 가스 경보기가 연기 및 가스를 인식하게 되면 도어락이 잠김과 동시에, 가스 차단기 작동, 콘센트 전원이 차단되고, 전등이 소등이 되어 그 시설은 사람이 없어도 봉쇄가 된다.



### 3. 조별 일지

11.16 – 키트 수령 후 같이 식사를 하며 주제 선정 토의

- 나왔던 주제: 비상 상황을 가정하여 건물 봉쇄하는 시스템  
잠을 잘 때 움직이기 싫은 상황에서 조명 조절  
방 출입 시 자동으로 조명 및 전원 키는 시스템 등

11.21 – 개인 공부 및 개발환경 구축(CD가 없어서 검색만 함)

- TinyOS2X 환경 구축 및 Cygwin 설치(최신 버전 설치 했으나 작동 안됨)

11.23 – 개인 공부 및 개발환경 구축(CD가 없어서 다른 팀에게 빌림)

- TinyOS2X 환경 구축 및 Cygwin 설치

11.28 – 예제 코드 테스트

- 적외선을 이용한 동작 센서 실행 등 테스트

11.30 – 주제 선정 완료 및 세부 조사

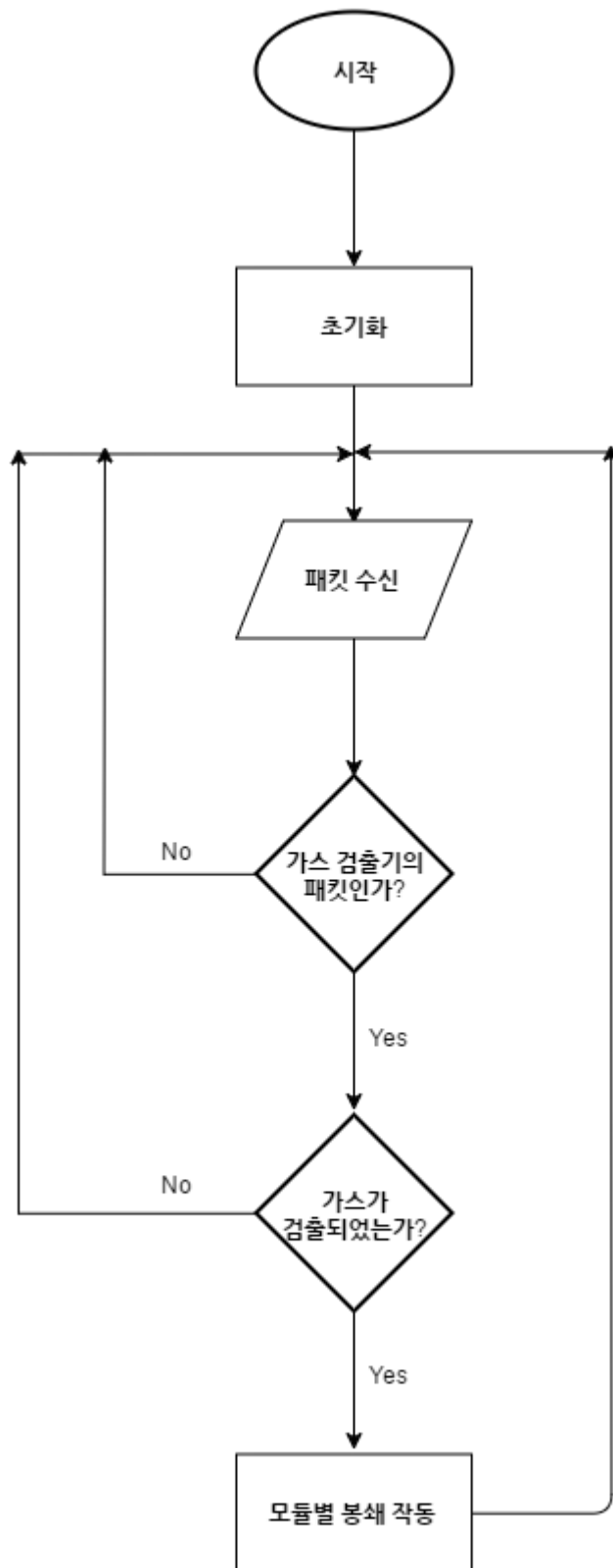
- 선정된 주제: 가스 검출기가 가스를 인식하면 도어락을 잠그고 전원 차단 및 전등 소등
- 선정 이유: 2차 피해를 예방 하기 위한 시스템을 홈 오토메이션 키트로 구현해보고자 함

12.1 – 목표 기능 구현 및 코드 작성 확인

12.3 – 보고서 작성, PPT 작성, 최종 작동 확인

\* 모든 조원 참석

## 4. Flow Chart



## 5. 주요 함수 및 핵심 알고리즘

### 1) 가스검출기

Configuration 파일(Ch6\_GasDetector\_101C.nc)

```
...
Components GasDetectorC;
    Ch6_GasDetector_101M.GasDetector->GasDetectorC;
}
```

DIO 보드(2 Digital Input & 3 Digital Output)의 Input 포트 단자를 통해 들어오는 가스 감지기의 신호를 인터럽트로 처리하고, 가스가 발생시 해당 정보를 알려주는 GasDetectorC 컴포넌트를 선언하고 있습니다. GasDetectorC 컴포넌트는 GasDetector라는 인터페이스를 제공하는데, 위에 서술한 코드는 모듈 파일과 해당 인터페이스 간의 와이어링 하는 것입니다.

Module 파일(Ch6\_GasDetector\_101M.nc)

```
...
event void GasDetector.EscapeOfGas() {
    user_header_t uh;                                // 메시지 헤더
    uh.id        = TOS_NODE_ID;
    uh.cmd       = COMM_INT;                          // 경보 메시지
    uh.sub       = COMM_INT_GAS_DETECT;               // 가스 검출기
    uh.length    = 0;                                // 데이터 영역 x
    atomic gasStatus = GAS_ESC;                      // 가스 검출
    // 주변에 데이터 보낼 준비
    memcpy(call AMSend.getPayload(&CMDmsg), &uh, sizeof(user_header_t));
    // 주변에 데이터 BROADCAST로 보냄
    call AMSend.send(AM_BROADCAST_ADDR, &CMDmsg, sizeof(user_header_t));
}
...
}
```

GasDetectorC 컴포넌트에 의해 가스가 검출되었을 때, EscapeOfGas() 이벤트 함수가 호출됩니다. 해당 함수가 호출될 경우 user\_header\_t uh 패킷에 cmd 필드에는 인터럽트와 관련된 COMM\_INT 타입을 넣고 sub 필드에는 가스 검출과 관련된 COMM\_INT\_GAS\_DETECT 타입을 넣어 RF로 전송합니다.

## 2) 도어락

Module파일(Ch4\_DoorLock\_31M.nc)

```
...
int flag;      // 도어락 잠금여부용
...
// 전원 공급시 초기화 함수
event void Boot.booted() {
    atomic {
        ...
        flag = 0;      // locked
    }
...
}
...
// RF로 들어오는 패킷 수신 함수
event message_t* Receive.receive(message_t* msg, void* payload, uint8_t len) {
...
if (pUH->id == 101) /*패킷 정보의 id가 101(가스 검출기) 이면*/ {
    switch(pUH->cmd) /*패킷 정보의 cmd(명령) 정보 확인*/ {
        case COMM_INT:
            if(pUH->sub == COMM_INT_GAS_DETECT) /*가스 검출*/ {
                if (flag == 1) {
                    flag = 0;
                    RelayChange(NOT_CONTROL, RELAY_ON, 500);
                }
            }
            break;
        }
    }
    return msg;
}
...
}
```

RF로 들어오는 패킷들 중 가스 감지 패킷을 구분하고 그 패킷을 받게 될 경우 문을 잠그는 동작을 수행하도록 하였습니다. Receive.receive 함수는 RF로부터 데이터를 받게 되었을 때 호출되는 함수입니다. 이 함수 안에 id가 101이며 cmd는 COMM\_INT이고 sub 필드로는 COMM\_INT\_GAS\_DETECT 타입의 패킷이 들어올 경우에 문이 잠겨 있지 않다면 문을 잠그기 위해 RelayChange(NOT\_CONTROL, RELAY\_ON, 500)를 호출합니다.

### 3) 콘센트

Module파일(Ch5\_Outlet\_61M.nc)

```
...
int flag;
...
// 전원 공급시 초기화 함수
event void Boot.booted() {
    atomic {
        ...
        flag = 1;    // power-on
    }
    ...
}

...
// RF로 들어오는 패킷 수신 함수
event message_t* Receive.receive(message_t* msg, void* payload, uint8_t len) {
    ...
    if (pUH->id == 101) /*패킷 정보의 id가 101(가스 검출기)이면*/ {
        switch(pUH->cmd) /*패킷 정보의 cmd(명령) 정보 확인*/ {
            case COMM_INT:
                if(pUH->sub == COMM_INT_GAS_DETECT) /*가스 검출*/ {
                    if(flag == 1) /*콘센트 전원이 공급중이면*/ {
                        flag = 0;    // 콘센트 전원 차단 플래그 설정
                        // 콘센트 전원 차단
                        RelayChange(NOT_CONTROL, RELAY_OFF, Relay_time);
                    }
                }
                break;
            }
        }
        return msg;
    }
    ...
}
```

도어락과 마찬가지로, 가스 감지 패킷을 구분하고 그 패킷을 받게 될 경우 콘센트 전원 차단을 수행하도록 하였습니다.



## 4) 전등

Module파일(Ch8\_Dimmer\_181M.nc)

```
...
// RF로 들어오는 패킷 수신 함수
event message_t* Receive.receive(message_t* msg, void* payload, uint8_t len) {
...
if (pUH->id == 101) /*패킷 정보의 id가 101(가스 검출기) 이면*/ {
    switch(pUH->cmd) /*패킷 정보의 cmd(명령) 정보 확인*/ {
        case COMM_INT:
            if(pUH->sub == COMM_INT_GAS_DETECT) /*가스 검출*/ {
                call dimmer.setLevel(20); // 전등 소등
            }
            break;
    }
    return msg;
}
...
}
```

도어락과 마찬가지로, 가스 감지 패킷을 구분하고 그 패킷을 받게 될 경우 전등 소등을 수행하도록 하였습니다.

## 5) 가스차단기

Module파일(Ch7\_GasShutter51M.nc)

```
...
uint8_t bOpen = TRUE; // 가스차단기 잠금 여부 확인용 (열림)
...
// RF로 들어오는 패킷 수신 함수
event message_t* Receive.receive(message_t* msg, void* payload, uint8_t len) {
...
if (pUH->id == 101) /*패킷 정보의 id가 101(가스 검출기) 이면*/ {
    switch(pUH->cmd) /*패킷 정보의 cmd(명령) 정보 확인*/ {
        case COMM_INT:
            if(pUH->sub == COMM_INT_GAS_DETECT) /*가스 검출*/ {
                if(bOpen == TRUE) /*가스차단기가 현재 열려 있으면*/ {
                    bOpen = FALSE; // 가스차단기 닫힘 플래그 설정
                    call Motor.M1_backward(); // 가스차단기 닫힘
                    // 9초 뒤부터 패킷 수신 가능
                    call GasTimer.startOneShot(9000);
                }
            }
            break;
    }
    return msg;
}
...
}
```

도어락과 마찬가지로, 가스 감지 패킷을 구분하고 그 패킷을 받게 될 경우에 가스 차단기가 열려 있으면 닫히도록 하였습니다.

## 6. 토의 ( 1)추후 발전했으면 하는 방향, 2)구현하지 못한 기능, 3)계획에 비해 새롭게 추가한 기능)

1) 홈 오토메이션 키트의 경우 부품의 기능이 제한적이기 때문에 부품의 기능에 맞추어 주제를 정하다 보니 생각할 수 있는 주제가 한정적이었다. 하지만 추후 프로젝트에서 좀 더 폭 넓은 부품으로 프로젝트를 진행할 수 있다면 다음과 같은 주제로 진행하고자 한다.

- 스마트폰과 홈 오토메이션 시스템을 연동하여 원격으로 시스템 제어
- > 가스차단기, 전원, 도어락 등을 스마트폰과 연동하여 사용자가 스마트폰을 이용해 원격으로 제어

2)

(1) 원래는 좀 더 정밀한 동작 센서가 주어진다면, 동작 센서에 동작을 인식 시키는 횟수에 따라 다르게 작동하는 홈 오토메이션 시스템을 개발하려고 했다.

- > 예를 들어 동작 센서가 동작을 한번 인식하면 조명 OFF, 동작을 두 번 인식하면 조명을 끄지 않고 조도를 낮춤. 이를 통해 잠을 잘 때 혹은 움직이지 않고 단순한 움직임으로 삶의 편리성을 만들기 위해 노력을 했었다. 하지만 이를 구현하지 못하여 주제를 안전 솔루션으로 바꿨다.

(2) 센싱 가능한 영역이 조도, 습도, 전압 레벨으로 정해져 있어 아쉬웠습니다.

- > 소리 등의 좀 더 다양한 것을 감지할 수 있는 센서가 제공되면 좋겠습니다.

3) 도어락의 경우엔 도어락이 현재 잠겨 있는지 열려 있는지 구분이 안되고, 콘센트의 경우엔 현재 전원이 연결 되어있는지 차단 되어있는지 구분이 안되는 등, 하드웨어적으로 내부 상태 판별이 불가능하였습니다.

- > 소프트웨어 상에서 flag를 두어 구현하였습니다.

## 7. 고장 내역 및 없는 부품

1) 키트에서 모든 것이 구비 되어 있다고 생각하였으나 없는 것도 있어서 혼란스러웠다. (설치 CD)

-> 같은 키트를 사용하는 다른 조의 조연을 얻어 CD가 필요함을 알았다.

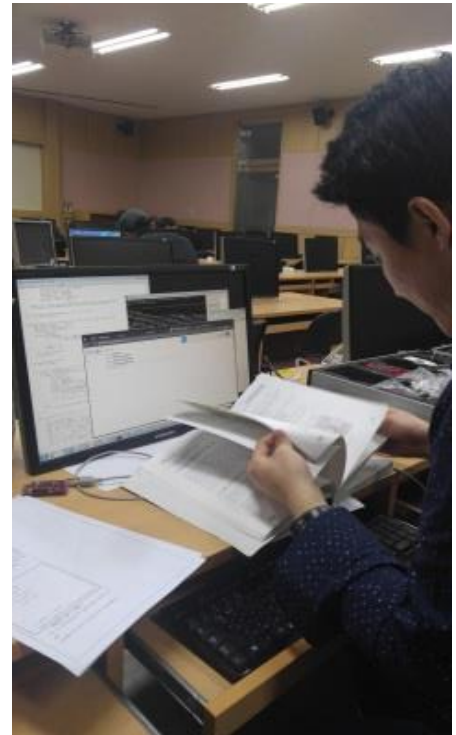
-> 다른 조의 조원에게 파일을 얻어 다운로드를 하여 진행할 수 있었다.

2) 일반 PC에 ZigbeX 모듈을 연결하고자 할 때, 키트에 기본적으로 포함 되어있는 usb 연장 케이블이 없어 정상적인 usb 연결이 힘들었습니다.

-> usb 연장 케이블을 개별 구입하였습니다.

## 8. 부록. 프로젝트 과정

### <그동안의 조모임>



개인 공부 및 회의를 진행하는 사진

## 부품 확인



AC input 보드의  
ZigbeX 노드와  
DIO 보드, 가스 경보기  
간의 연결



Dimmer 보드와 램프 간의 연결



AC input 보드의 ZigbeX 노드와 도어락 장비 간의 연결



AC input 보드의 ZigbeX 노드와 콘센트 장비 간의 연결



<최종 구동>

