

대형 물류창고 재난 상황 응급 구조 시스템



마이크로프로세서응용 (0642) 7조

마현준 / 선종민 / 장정욱 / 허무혁

목차

주제 , 배경 , 목표

제작 과정 : 모형제작 / 회의록

코드, 결과물

결론

이천서 또 물류센터 화재··건물 주저앉고 사방에 잿더미

22.05.23

구광 이천 물류창고 화재현장서 실종된 소방관 숨진 채 발견

21.06.19

용인 물류창고 지하서 화재...일부 근로자 연락 안돼

20.07.21

또 물류창고 화재...현장 투입됐던 소방관 3명 순직

22.01.06

프로젝트 목적

대피로 사전 인지

화재 조기 진압 시설



신속한 구조 대상자 위치 파악

동선 단축 / 구조 시간 최소화

중대재난사고 예방을 위한
조각돌을 찾아서

제한 조건 해결

✓ 실제 화재 상황 구현 불가

-> 키트의 버튼을 누르면 화재 발생 설정

✓ 심장 박동수 센서 대체 방법

-> 외부 신호 입력과 동일한 Timer를 사용

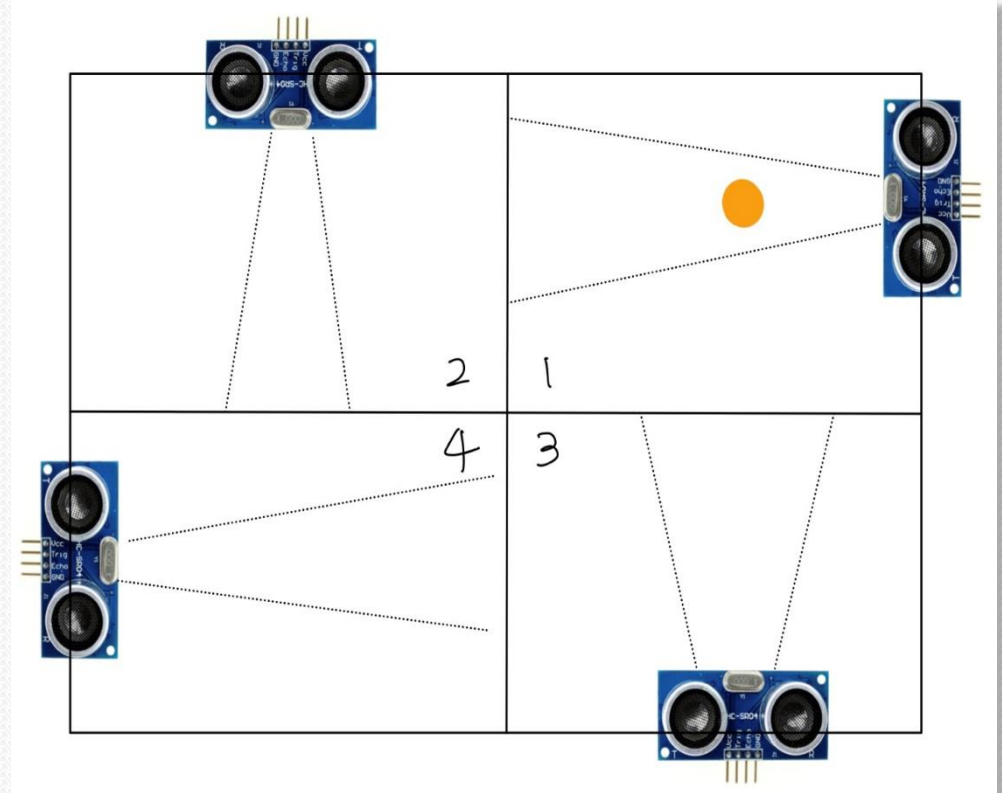
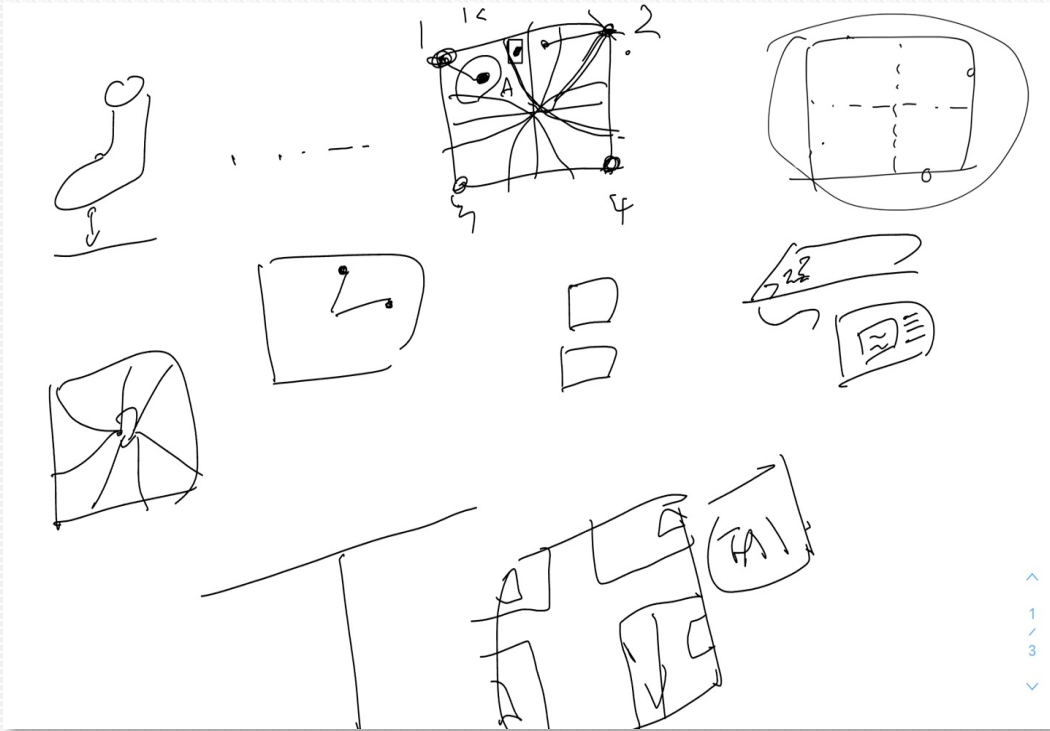
✓ 여러 구역에 구조 대상자가 존재할 경우?

-> 초음파 센서의 추가 배치

✓ 초음파 센서 개수가 늘어나면서 핀 부족 문제 발생

-> 키트 추가 사용

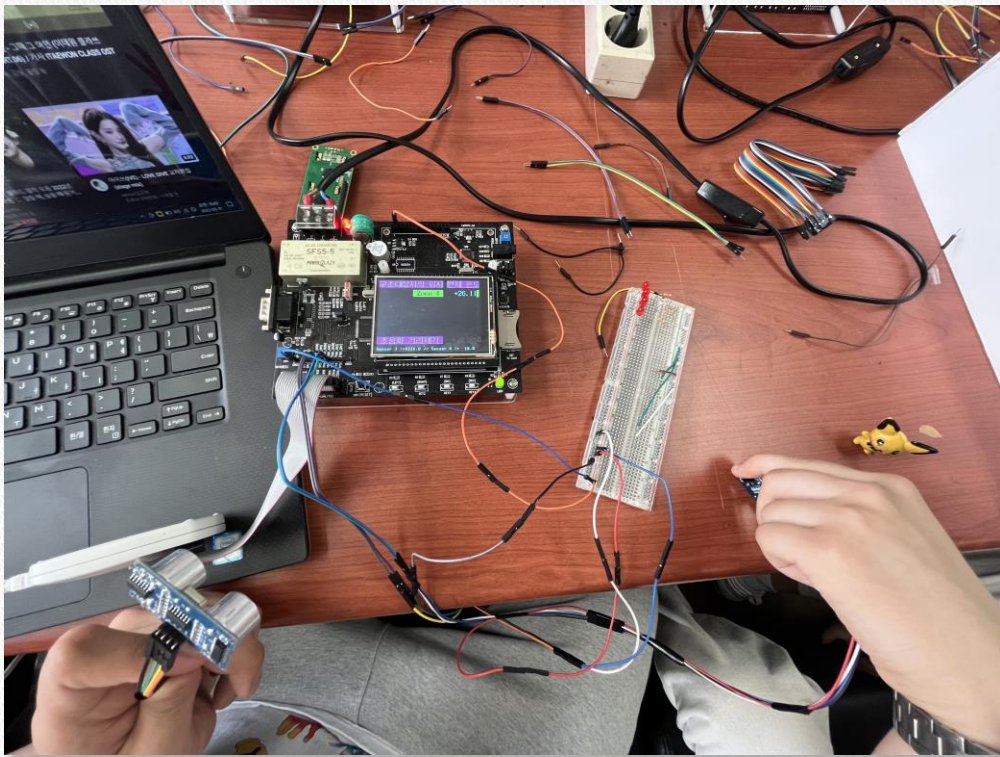
아이디어 창출



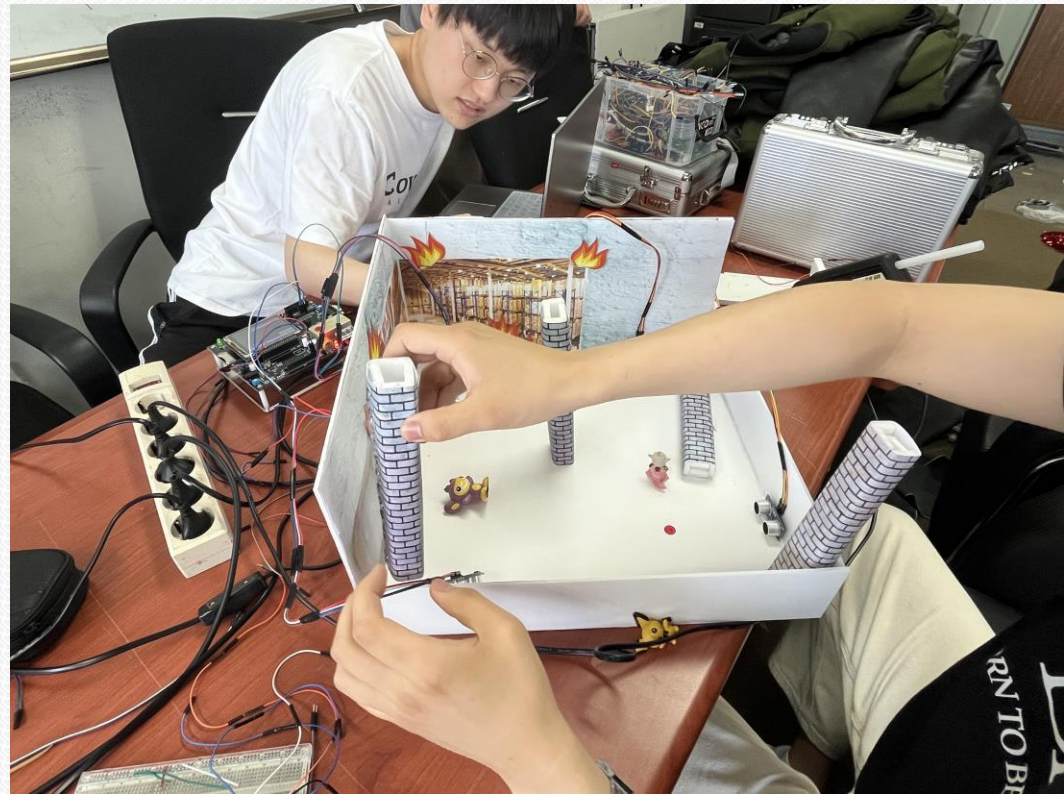
회로 구성

KIT 1	KIT 2	KIT 3
초음파 센서 1 (HC-SR04)	초음파 센서 3 (HC-SR04)	블루투스 모듈 (HC-06)
초음파 센서 2 (HC-SR04)	초음파 센서 4 (HC-SR04)	직렬유선통신 (RS-232C)
온도 센서 (LM35DZ)	직렬유선통신 (RS-232C)	
직렬유선통신 (RS-232C)		


제작 과정 : 모형제작



제작 과정 : 모형제작



제작 과정 : 회의록

 허무혁

지금 키트 3 휴대폰 메시지 보내는 거 통신 두번해서


컴퓨터로 보내는거 까지 했거든?

이제 코드는 다짤거 같고

ppt 만들고 최종보고서 작성하자

키트 작동하는 부분은 내가 ppt로 만들어서 올려줄테니까

오후 3:34


 허무혁

기술적인 한계랑

발표때 지적받았던 초음파 센서 문제 각자 찾아서


작업하자

오후 3:35

 마현준

굿굿

오후 3:35


 허무혁

그리고 기술이 필요한 이유랑 그런 거도 좀더 구체적으로 하고

오후 3:35


team_project_0516.c
유효기간: ~2022.05.31
용량: 4.41KB
저장 · 다른 이름으로 저장

오전 10:09

 선종민

key 1: 화재발생
key 2: zone 표시
key 3: 타이머 심장박동
key 4: 구조대상자가 위치 직접 표현

오전 10:52


 허무혁

Teamproject_0517.c
유효기간: ~2022.05.31
용량: 6.52KB
저장 · 다른 이름으로 저장

온도센서


코드

오전 11:22

 허무혁

Teamproject_0517 (2).c
유효기간: ~2022.05.31
용량: 6.97KB
저장 · 다른 이름으로 저장

오전 11:49

 허무혁


그거 진짜 너무많이깨져서 그나마 보완한게 그거

오후 11:05

왜그러지?

오후 11:05

전원을 너무 많이 써서 그런가

 허무혁

커서가 튀는거 같은데

오후 11:05

그니까

커서도 튀고


오후 11:05

그냥 TFT 컨트롤 자체가 오류 나는 모양인드

맞다 그리고 이거 1,2번 버튼은

누르면 사실 초음파 신호 무시하고 그냥 그 존에 있다고 나와야 하는거 아니야?


오후 11:07

 허무혁

꼭 누르면

그냥 계속 뜨고

오후 11:07

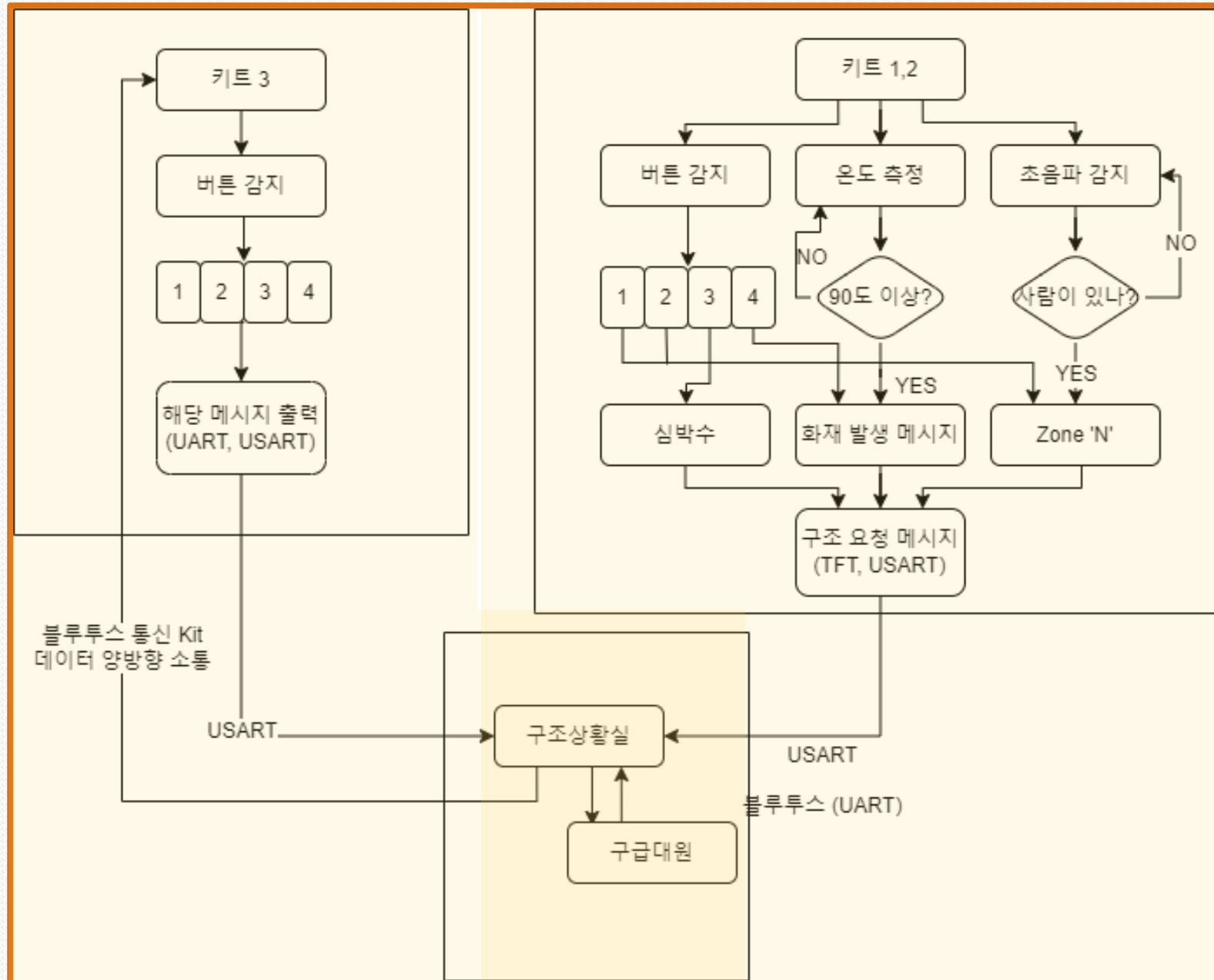
 허무혁

유아트로 출동하라고 모니터로 뛰우니까

괜찮을거 같은데

오후 11:08

블록 다이어그램 : 기능설명



키트 1 초음파 센서

```
void SetUltraSonic_func(void) {

    RCC->AHB1ENR |= 0x00000010;
    // 포트 E클락 온
    GPIOE->MODER &= 0xFF00FFFF;
    GPIOE->MODER |= 0x00110000;
    // 포트8출력모드(초음파의 트리거 신호)
    // 포트9 입력모드(에코신호를 키트의 입력으로 받음)
    // 포트9출력모드(초음파의 트리거 신호)
    // 포트10 입력모드(에코신호 키트의 입력으로 받음)
}
```

```
/*TIM2 Setting*/
void init_TIM2(void) {
    RCC->APB1ENR |= (1 << 0);           // Bit 0 TIM2EN
    TIM2->PSC = 72 - 1;                  // 1us
    TIM2->EGR = (1 << 0);                // Bit 0 UG
    TIM2->CR1 = (1 << 0);                 // Bit 0 CEN
}
```

```
/*TIM5 Setting*/
void init_TIM5(void) {

    RCC->APB1ENR |= (1 << 3);           // Bit 1 TIM5EN
    TIM5->PSC = 72 - 1;                  // 1us
    TIM5->EGR = (1 << 0);                // Bit 0 UG
    TIM5->CR1 = (1 << 0);                 // Bit 0 CEN
}
```

TIMER

타이머 CNT를 1us마다 작동하게 하여 초음파 센서 작동에 사용

키트 1 초음파 센서 1

```

void UltraSonic1_func(void) {

    GPIOE->BSRR = 0x01000100;
    //GPIOE_ODR 출력레지스터를 세트
    TIM2->CNT = 0;          //카운터 초기화

    while (TIM2->CNT < 12);
    // 12us 시간 대기 초음파 센서 작동 시간 (12us)기다리기 (Sensor 1)
    GPIOE->BSRR = 0x01000000;    //GPIOE_ODR 출력레지스터를 리셋

    while (!(GPIOE->IDR & 0x0200));
    //GPIOE_IDR (ECHO1 - PE8) 입력레지스터 SET을 기다린다 (Sensor 1)
    TIM2->CNT = 0;          //TIM2 카운터 초기화

    while (GPIOE->IDR & 0x0200);
    //GPIOE_IDR 입력레지스터 OFF를 기다린다. (Sensor 1)
    distance1 = (TIM2->CNT + 1) / 58;    // cm 거리 연산

    TFT_xy(10, 28);
    TFT_color(Cyan, Black);
    TFT_signed_float((float)distance1, 4, 1);

    if (distance1 <= 20) {
        TFT_string_large(0, 4, Black, Green, " Zone 1 ");
    }
    else {
        TFT_string_large(0, 4, Black, Black, "      ");
    }

    LED_toggle();          // 제대로 반복하여 동작하는지 확인하는 LED
    Delay_ms(200);
}

```

키트 1 초음파 센서 2

```

void UltraSonic2_func(void) {

    GPIOE->BSRR = 0x04000400;
    //GPIOE_ODR 출력레지스터를 세트
    TIM5->CNT = 0;

    while (TIM5->CNT < 12);
    // 12us 시간 대기 초음파 센서 작동 시간 (12us)기다리기 (Sensor 2)
    GPIOE->BSRR = 0x04000000;    //GPIOE_ODR 출력레지스터를 리셋

    while (!(GPIOE->IDR & 0x0800));
    //GPIOE_IDR (ECHO2 - PE10) 입력레지스터 SET을 기다린다 (Sensor 2)
    TIM5->CNT = 0;          //TIM5 카운터 초기화

    while (GPIOE->IDR & 0x0800);
    //GPIOE_IDR 입력레지스터 OFF를 기다린다. (Sensor 2)
    distance2 = (TIM5->CNT + 1) / 58;    // cm 거리 연산

    TFT_xy(31, 28);
    TFT_color(Cyan, Black);
    TFT_signed_float((float)distance2, 4, 1);

    if (distance2 <= 20) {
        TFT_string_large(9, 4, Black, Green, " Zone 2 ");
    }
    else {
        TFT_string_large(9, 4, Black, Black, "      ");
    }

    Delay_ms(200);
}

```

TIMER

각 초음파 센서의 범위 안에 물체가 감지된다면 LCD에 구역 정보 표시

키트 1 온도 센서

```
void init_ADC1(void) {
    GPIOA->MODER |= 0x000000C0;

    RCC->APB2ENR |= 0x00000100;

    ADC->CCR = 0x00000000;

    // ADCCLK = 54MHz/2 = 27MHz

    ADC1->SMPR2 = 0x00000200;
        // sampling time of channel 3 = 15 cycle
    ADC1->CR1 = 0x00000020;

    // 12-bit resolution, ADC EOC interrupt enable
    ADC1->CR2 = 0x1C000001;

    // right alignment, trigger by TIM2_trgo,single
    conversion, ADON = 1
    ADC1->SQR1 = 0x00000000;
        // total regular channel number = 1,
    ADC1->SQR3 = 0x00000003;
        // channel 3 (LM35DZ) :: ADC1에 CH3 연결

    NVIC->ISER[0] |= 0x00040000;
        // enable ADC interrupt
}
```

```
void init_TIM4(void) {

    RCC->APB1ENR |= 0x00000004;

    // enable TIM3 clock
    TIM4->PSC = 2159;

    // 108MHz/(2159+1) = 50kHz
    TIM4->ARR = 49999;

    // 50kHz/(49999+1) = 1Hz
    TIM4->CNT = 0;

    // clear counter
    TIM4->CR1 = 0x0005;

    // enable TIM2 and update event
    TIM4->CR2 = 0x0020;

    // update event가 TRGO로 사용됨
}
```

코드

ADC
TIMER

ADC1 (LM35DZ) 센서 사용
TIM4가 온도 센서의 트리거로 사용

키트 1 온도 센서

코드

```
void ADC_IRQHandler(void) {  
  
    result = ADC1->DR;                // AD변환값 읽기  
    temp = result * 330 / 4095;  
    TFT_xy(20, 4);  
    TFT_color(Cyan, Black);  
    TFT_signed_float_large((float)result * 330. / 4095., 2, 2); // TFT LCD에 온도 표시  
  
    if (temp > 90.0) {  
        TFT_string_large(18, 4, White, Magenta, "화재 발생");  
  
        ADC1->CR2 = 0x00000000;  
  
        TX3_string(" FIRE ! ");  
        TX3_char(0x0D); TX3_char(0x0A);  
        TX3_string(" Location : Yongin-si 351-00 ");  
        TX3_char(0x0D); TX3_char(0x0A);  
        TX3_string(" Employee : 300 ");  
        TX3_char(0x0D); TX3_char(0x0A);  
        TX3_string(" Type : Ware House ");  
  
        for( int i = 0; i < 50; i++){  
            Beep();  
            Delay_ms(20);  
        }  
    }  
}
```

90도 이상이면 USART3와 TFT를 통해 메시지 송출
USART의 경우 건물의 위치와 근무자의 수를 메시지에 포함
BEEP 경고음을 반복

ADC
TIMER

키트 1 버튼

```
void Key_enable(void)
{
    GPIOC->MODER &= 0xFFFFFFF00;
    // PC3,2,1,0 = input mode
    RCC->APB2ENR |= 0x00000001;
    // enable SYSCFG
    SYSCFG->EXTICR[0] = 0x00002222;
    // EXTI3,2,1,0 = PC3,2,1,0(KEY4,3,2,1)
    EXTI->IMR = 0x0000000F;
    // enable EXTI3,2,1,0 interrupt
    EXTI->RTSR = 0x00000000;
    // disable EXTI3,2,1,0 rising edge
    EXTI->FTSR = 0x0000000F;
    // enable EXTI3,2,1,0 falling edge
    NVIC->ISER[0] = 0x000003C0;
    // enable (9)EXTI3 ~ (6)EXTI0 interrupt
}
```

INTERRUPT

KEY 1,2,3,4에 대한 외부 인터럽트 허용

키트 1 - 버튼 1

```
void EXTI0_IRQHandler(void)
    /* EXTI0 interrupt function */
{
    /*여기에 key 1 기능을 적어주세요
       zone 1에 재난 발생*/
    TX3_string(" Zone 1 emergency ");
    TX3_char(0x0D); TX3_char(0x0A);

    disable_sensor1 = 1;
    TFT_string_large(0, 4, Black, Green, " Zone 1 ");

    while ((GPIOC->IDR & 0x00000001) != 0x00000001);
    // debouncing
    Delay_ms(20);
    EXTI->PR = 0x00000001;
    // clear pending bit of EXTI0
    NVIC->ICPR[0] = 0x00000040;
    // clear pending bit of (6)EXTI0
}
```

KEY 1 : 외부 인터럽트 0번

TFT와 USART3(RS-232C)를 통해
"ZONE1" 메시지를 출력

키트 1 - 버튼 2

```
void EXTI1_IRQHandler(void)
    /* EXTI1 interrupt function */
{
    /*여기에 key 2 기능을 적어주세요
       zone 2에 재난 발생*/
    TX3_string(" Zone 2 emergency ");
    TX3_char(0x0D); TX3_char(0x0A);

    TFT_string_large(9, 4, Black, Green, " Zone 2 ");
    disable_sensor2 = 1;

    while ((GPIOC->IDR & 0x00000002) != 0x00000002); //
    debouncing
    Delay_ms(20);
    EXTI->PR = 0x00000002;
    // clear pending bit of EXTI1
    NVIC->ICPR[0] = 0x00000080;
    // clear pending bit of (7)EXTI1
}
```

KEY 2 : 외부 인터럽트 1번

TFT와 USART3(RS-232C)를 통해
"ZONE2" 메시지를 출력

INTERRUPT

키트 1 - 버튼 3

```
void EXTI2_IRQHandler(void)
    /* EXTI2 interrupt function */
{
    /*여기에 key 3기능을 적어주세요
    심박수 동작*/
    run_flag = 1;

    SysTick->CTRL = 0x00000003;// interrupt enable

    while ((GPIOC->IDR & 0x00000004) != 0x00000004); //
debouncing
    Delay_ms(20);
    EXTI->PR = 0x00000004;
// clear pending bit of EXTI2
    NVIC->ICPR[0] = 0x00000100;
// clear pending bit of (8)EXTI2
}
```

KEY 3 : 외부 인터럽트 2번

SYSTICK TIMER 작동

KEY 4 : 외부 인터럽트 3번

TFT와 USART3(RS-232C)를 통해

“화재 발생” 메시지 출력 (화재위치와 근로자 수 포함)

키트 1 - 버튼 4

```
void EXTI3_IRQHandler(void)
    /* EXTI3 interrupt function */
{
    /*여기에 key 4기능을 적어주세요
    화재감지 동작 확인*/
    TFT_string_large(18, 4, White, Magenta, "화재 발생");
    ADC1->CR2 = 0x00000000;
    TX3_string(" FIRE ! ");
    TX3_char(0x0D); TX3_char(0x0A);
    TX3_string(" Location : Yongin-si 351-00 ");
    TX3_char(0x0D); TX3_char(0x0A);
    TX3_string(" Employee : 300 ");
    TX3_char(0x0D); TX3_char(0x0A);
    TX3_string(" Type : Ware House ");
    TX3_char(0x0D); TX3_char(0x0A);

    while ((GPIOC->IDR & 0x00000008) != 0x00000008);
// debouncing
    Delay_ms(20);
    EXTI->PR = 0x00000008;
// clear pending bit of EXTI3
    NVIC->ICPR[0] = 0x00000200;
// clear pending bit of (9)EXTI3
}
```

INTERRUPT

TIMER

USART

키트 1 SYSTICK TIMER

```
void SysTick_Handler(void)
    /* SysTick interrupt function */
{
    second++;
    // increment 1/100 second

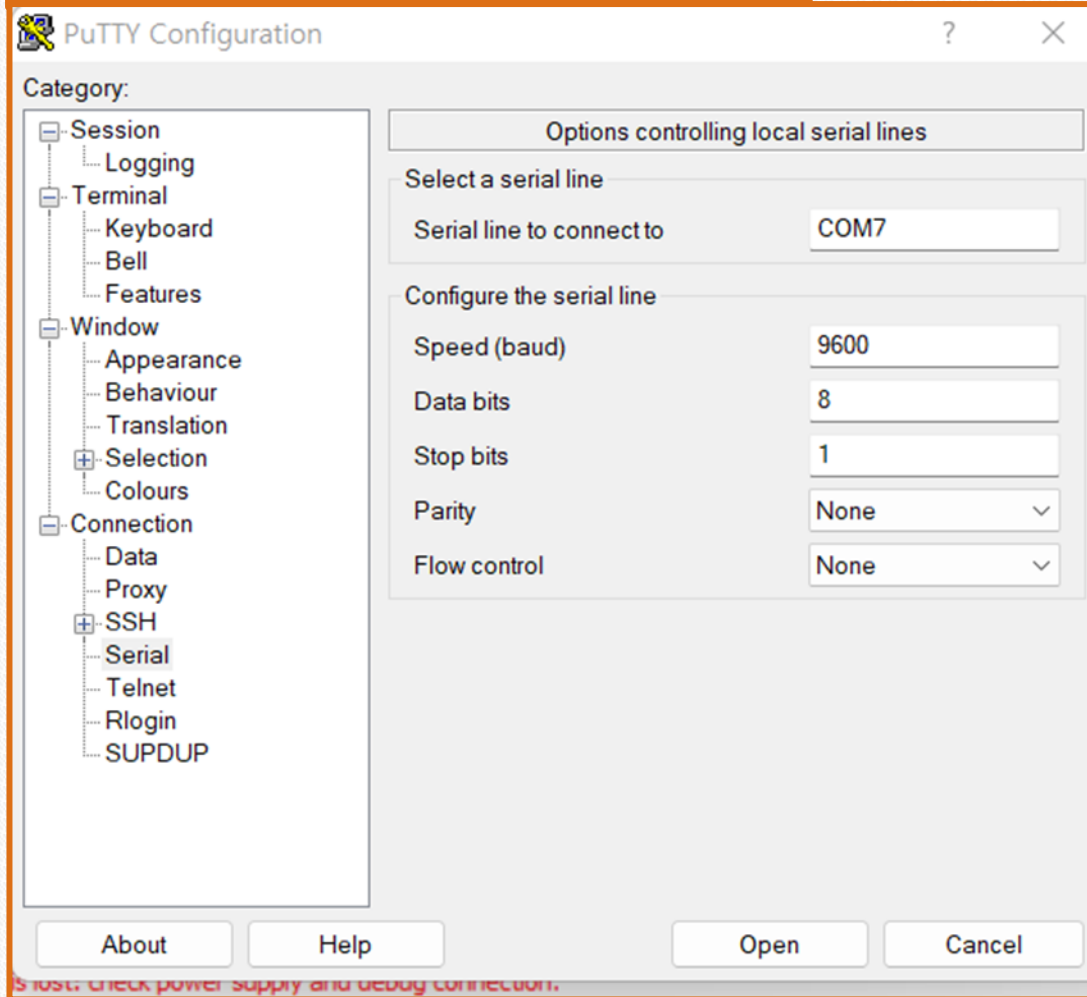
    if(100 - (second / 100) == 0 ){
        SysTick->CTRL = 0x00000001;// interrupt disable
        TFT_string(30, 14, Black, Green, " 심정지 " );
        TX3_char(0x0D); TX3_char(0x0A);
        TX3_string(" HeartRate stopped ");
        TX3_char(0x0D); TX3_char(0x0A);
    }

    if(second % 20 == 0){
        TFT_xy(30,12);
        TFT_unsigned_decimal(100 - (second / 100),1,2);
        TFT_string_large(0, 8, White, Magenta, "구조대상자의 건강상태");
        TFT_string(0, 12, Black, Green, "구조대상자의 심박수 : " );
    }
}
```

TIMER

KEY 3을 통해 SYSTICK TIMER가 작동
(SYSTICK TIMER는 심정지 상황을 묘사)
변수가 "0"이 되면 심정지로 판단하고 해당 상황을
TFT와 USART3을 통해 "HEARTRATE IS STOP" 출력

키트 1, 키트 3 USART3 통신



USART3의 세팅 값

BAUD RATE : 9600

DATA BITS : 8

STOP BIT : 1

NO FLOW CONTROL

USART

키트 1, 키트 3 USART3 통신

코드

```
void SET_USART3(void)
{
    GPIOB->MODER &= 0xFF0FFFFFFF;           // PB10 = USART3_TX, PB11 = USART3_RX
    GPIOB->MODER |= 0x00A00000;
    GPIOB->AFR[1] &= 0xFFFF00FF;
    GPIOB->AFR[1] |= 0x00007700;

    RCC->APB1ENR |= 0x00040000;              // enable USART3 clock

    USART3->CR1 = 0x0000000D;                // TE=RE=UE = 1, 8 data bit, oversampling
    by 16
    USART3->CR2 = 0x00000000;                // asynchronous mode, 1 stop bit
    USART3->CR3 = 0x00000000;                // 3 sampling bit
    USART3->BRR = 5624;                      // 9600 bps
    Delay_ms(1);
    RXD = USART3->RDR;                       // dummy read
}
```

USART

키트 1 UART7 블루투스 통신

```
void SET_UART7(void)
{
    GPIOA->MODER &= 0xFFFFCFFFF;    // PA8 = UART7_RX
    GPIOA->MODER |= 0x00020000;
    GPIOA->AFR[1] = 0x0000000C;    // AFIO 12
    GPIOE->MODER &= 0xFFFFCFFFF;    // PE8 = UART78_TX
    GPIOE->MODER |= 0x00020000;
    GPIOE->AFR[1] &= 0xFFFFFFF0;    // AFIO 8
    GPIOE->AFR[1] |= 0x00000008;
    RCC->APB1ENR |= 0x40000000;    // enable UART7 clock
    UART7->CR1 = 0x0000002D;    // TE=RE=UE = 1, 8 data bit, oversampling by 16
    UART7->CR2 = 0x00000000;    // asynchronous mode, 1 stop bit
    UART7->CR3 = 0x00000000;    // 3 sampling bit
    UART7->BRR = 5625;    // 9600 bps = 54MHz/5625
    Delay_ms(1);
    RXD2 = UART7->RDR;    // dummy read
}
```

BLUETOOTH

UART7 (Bluetooth) 세팅 값

PA8 부수적인 기능 12 / PE8 부수적인 기능 08 유의

BAUD RATE : 9600

DATA BITS : 8 / STOP BIT : 1 / NO PARITY / NO FLOW CONTROL

키트 3 - 버튼 1

```
void EXTI0_IRQHandler(void)
    /* EXTI0 interrupt function */
{ /*YES 송신*/
    TX7_char(0x0D); TX7_char(0x0A);
    TX7_string("Transmit : ");
    TX7_string(" YES!! ");
    TX7_char(0x0D); TX7_char(0x0A);
    TX3_char(0x0D); TX3_char(0x0A);
    TX3_string("Transmit : ");
    TX3_string(" YES!! ");
    TX3_char(0x0D); TX3_char(0x0A);
    while ((GPIOC->IDR & 0x00000001) != 0x00000001);
    // debouncing
    Delay_ms(20);
    EXTI->PR = 0x00000001; // clear pending bit of EXTI0
    NVIC->ICPR[0] = 0x00000040;
    // clear pending bit of (6)EXTI0
}
```

키트 3 - 버튼 2

```
void EXTI1_IRQHandler(void)
    /* EXTI1 interrupt function */
{ /*NO 송신*/
    TX7_char(0x0D); TX7_char(0x0A);
    TX7_string("Transmit : ");
    TX7_string(" NO ");
    TX7_char(0x0D); TX7_char(0x0A);
    TX3_char(0x0D); TX3_char(0x0A);
    TX3_string("Transmit : ");
    TX3_string(" NO ");
    TX3_char(0x0D); TX3_char(0x0A);
    while ((GPIOC->IDR & 0x00000002) != 0x00000002);
    // debouncing
    Delay_ms(20);
    EXTI->PR = 0x00000002; // clear pending bit of EXTI1
    NVIC->ICPR[0] = 0x00000080;
    // clear pending bit of (7)EXTI1
}
```

코드

KEY 1 : USART3과 UART7을 통하여

“YES” 메시지 전송

KEY 2 : USART3과 UART7을 통하여

“NO” 메시지 전송

INTERRUPT
BLUETOOTH

키트 3 - 버튼 3

```
void EXTI2_IRQHandler(void)
/* EXTI2 interrupt function */
{ /*테스트 버튼*/
    TX7_char(0x0D); TX7_char(0x0A);
    TX7_string("Transmit : ");
    TX7_string(" system test!! ");
    TX7_char(0x0D); TX7_char(0x0A);
    TX3_char(0x0D); TX3_char(0x0A);
    TX3_string("Transmit : ");
    TX3_string(" system test!! ");
    TX3_char(0x0D); TX3_char(0x0A);
    while ((GPIOC->IDR & 0x00000004) != 0x00000004);
    // debouncing
    Delay_ms(20);
    EXTI->PR = 0x00000004;// clear pending bit of EXTI2
    NVIC->ICPR[0] = 0x00000100;
    // clear pending bit of (8)EXTI2
}
```

KEY 3 : USART3과 UART7을 통하여

“SYSTEM TEST” 메시지 전송

KEY 4 : USART3과 UART7을 통하여

“화재 발생” 메시지 출력 (화재위치와 근로자 수 포함)

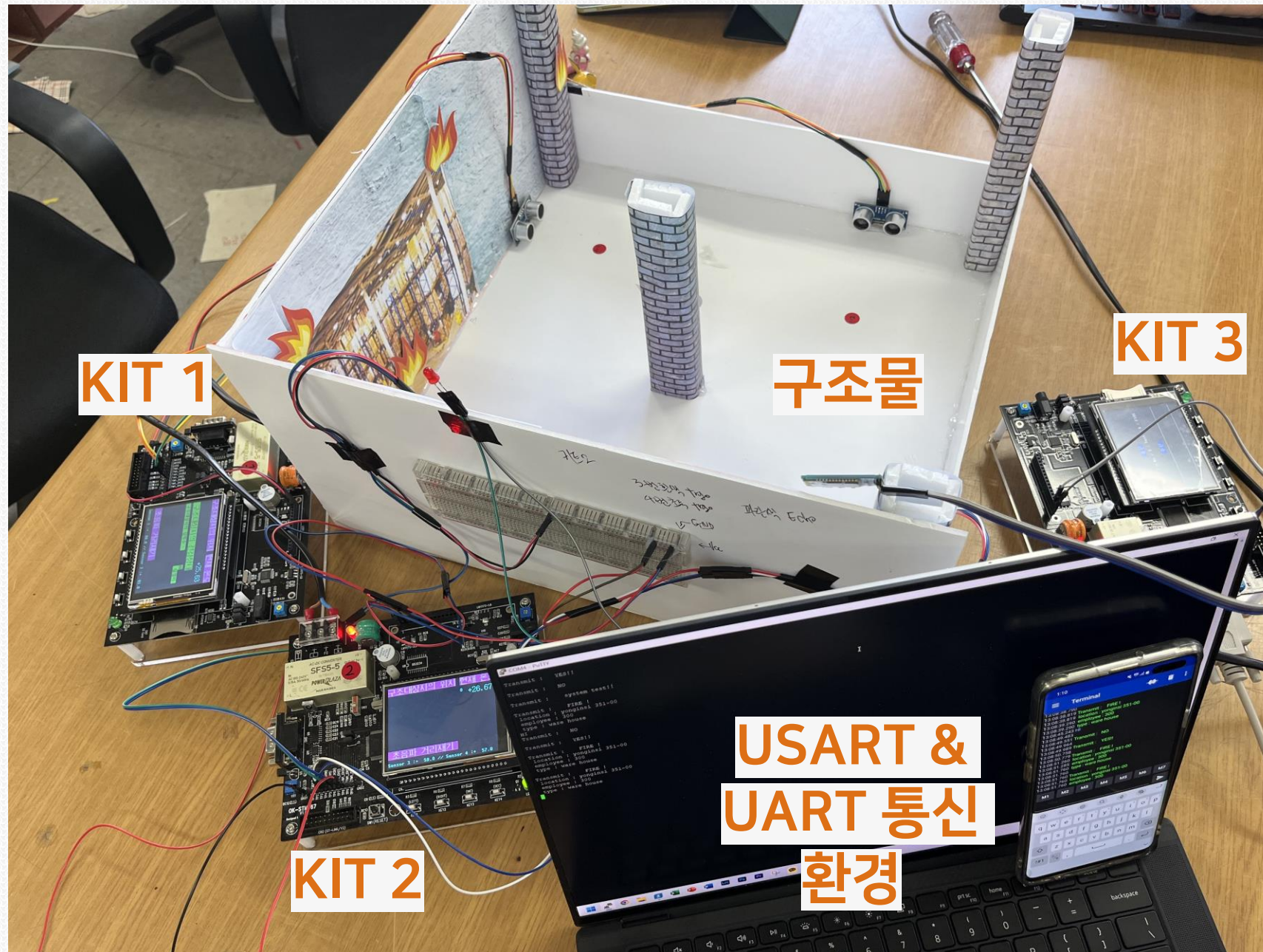
키트 3 - 버튼 4

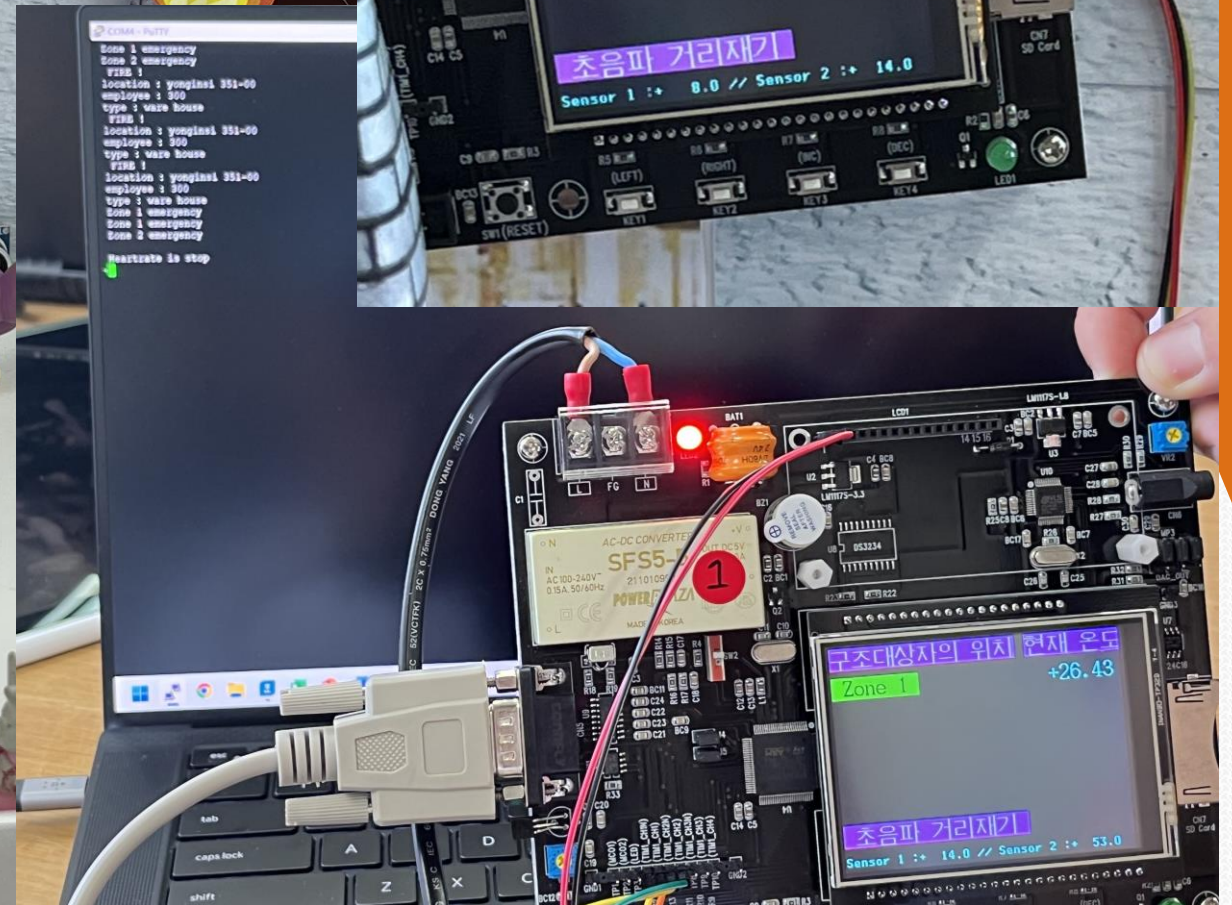
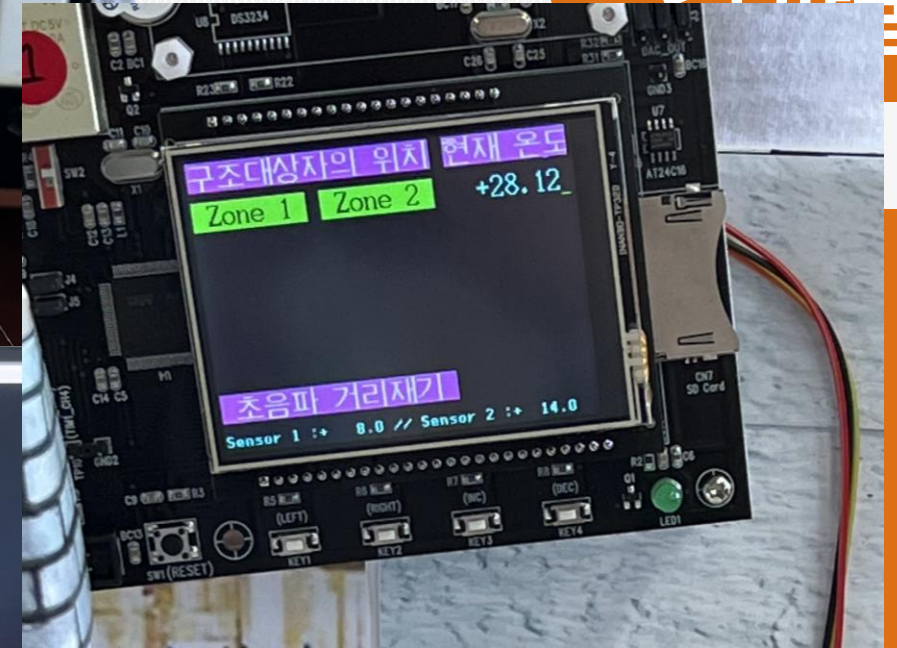
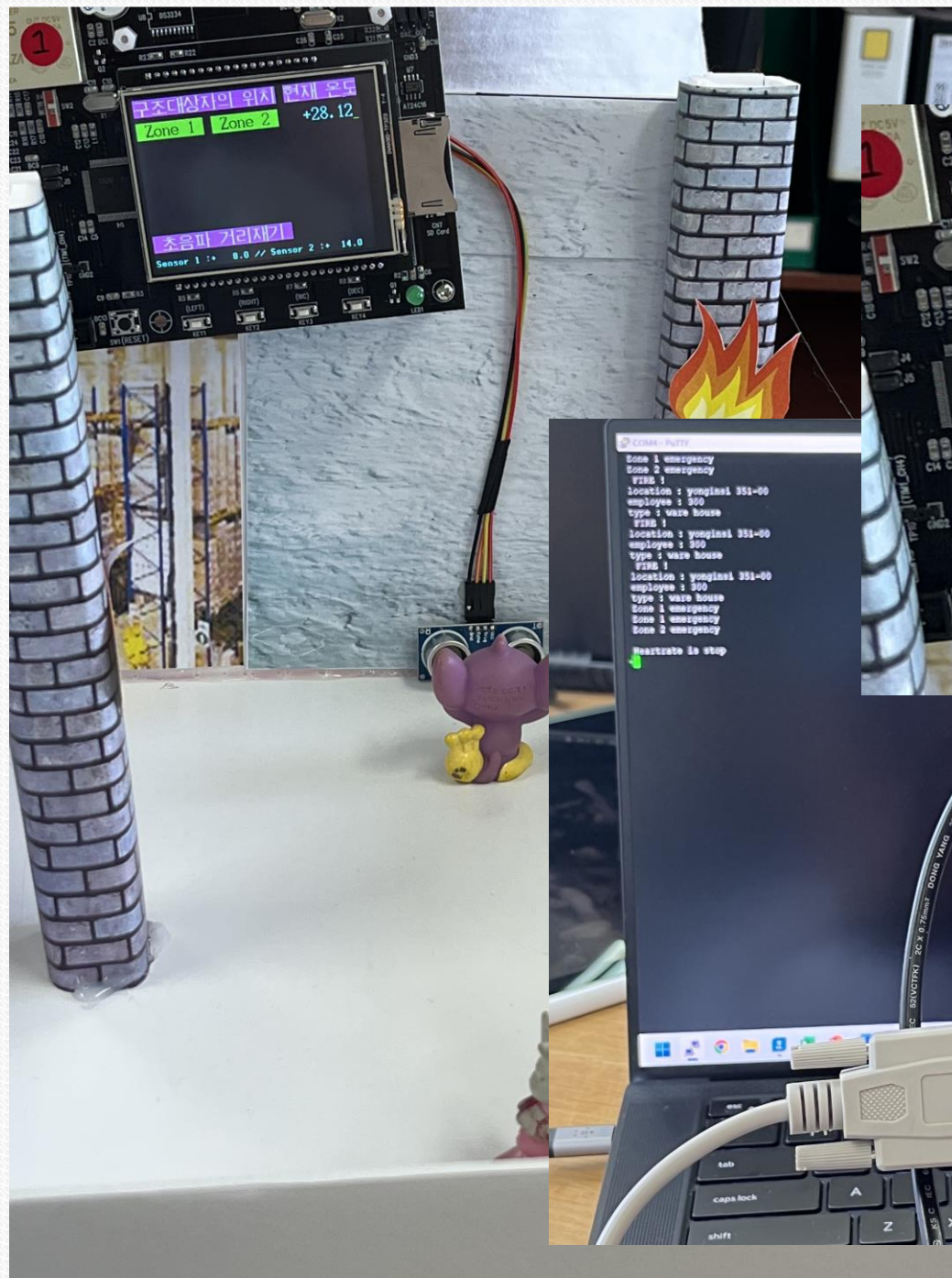
```
void EXTI3_IRQHandler(void)/* EXTI3 interrupt function */
{
    /*여기에 key 4기능을 적어주세요
    화재감지 동작 확인*/
    ADC1->CR2 = 0x00000000;
    TX3_char(0x0D); TX3_char(0x0A);
    TX3_string("Transmit : ");
    TX3_string(" FIRE ! ");
    TX3_char(0x0D); TX3_char(0x0A);
    TX3_string(" location : yonginsi 351-00 ");
    TX3_char(0x0D); TX3_char(0x0A);
    TX3_string(" employee : 300 ");
    TX3_char(0x0D); TX3_char(0x0A);
    TX3_string(" type : ware house ");
    TX3_char(0x0D); TX3_char(0x0A);
}
```

코드

INTERRUPT BLUETOOTH

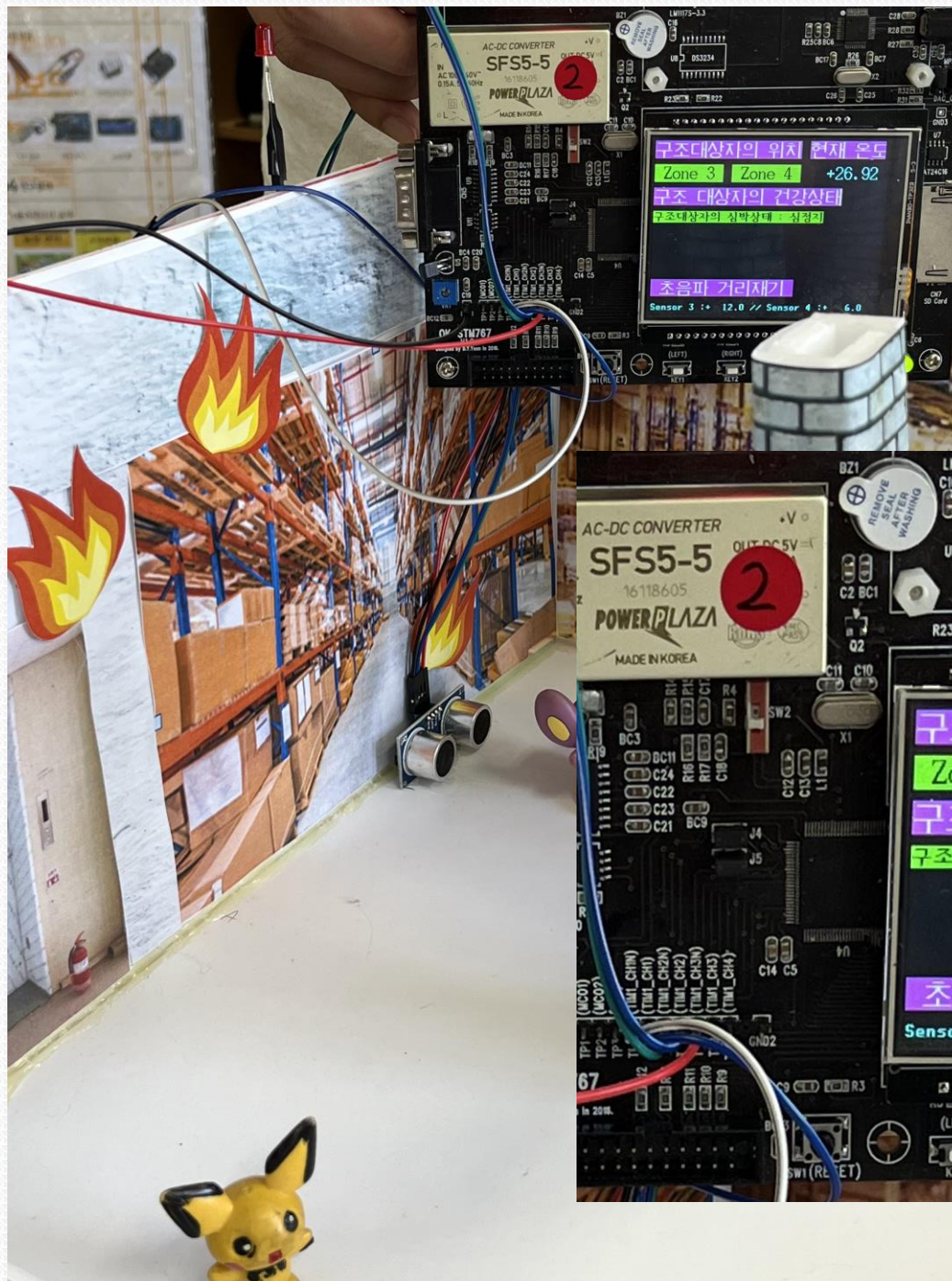
결과물





결과물

Kit 2



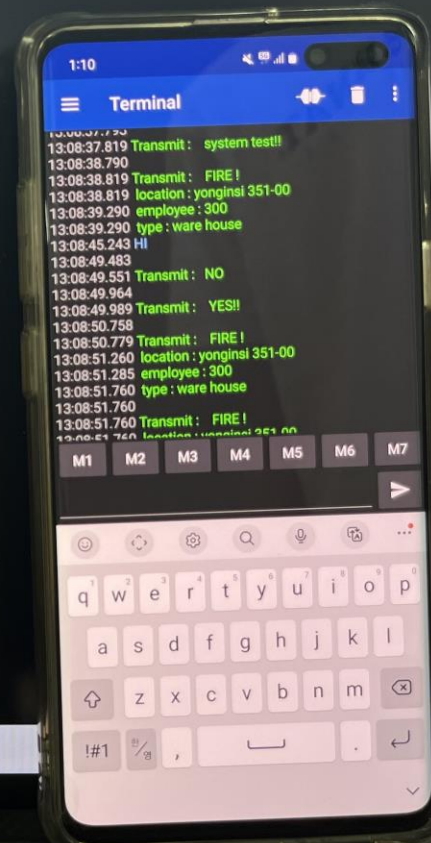
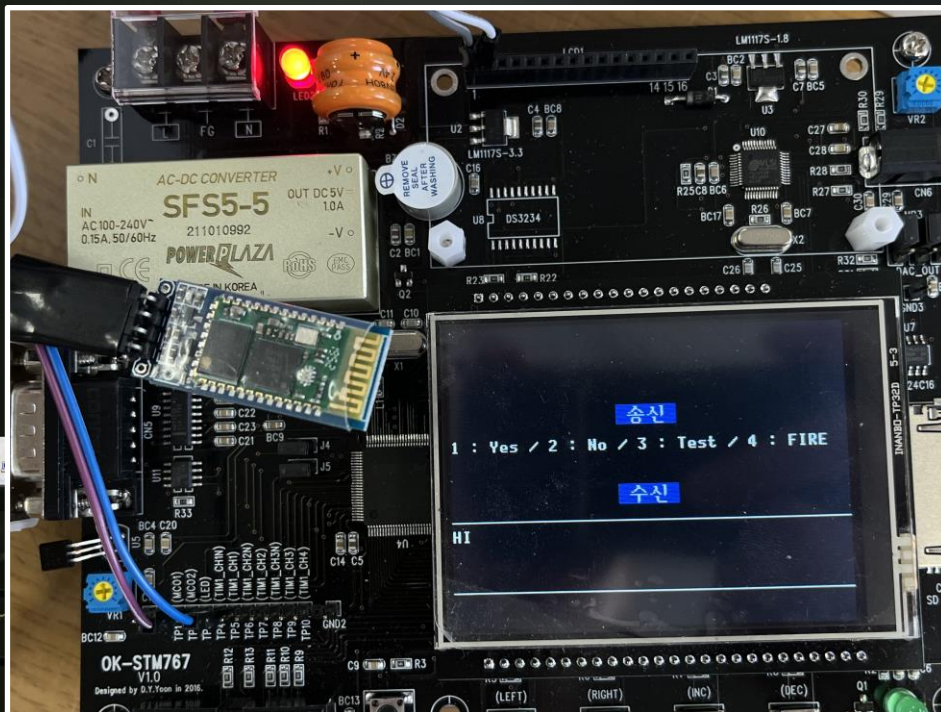
결과물

Kit 3

COM4 - PUTTY

```
Transmit : YES!!  
Transmit : NO  
Transmit : system test!!  
  
Transmit : FIRE !  
location : yonginsi 351-00  
employee : 300  
type : ware house  
HI  
Transmit : NO  
Transmit : YES!!  
  
Transmit : FIRE !  
location : yonginsi 351-00  
employee : 300  
type : ware house  
  
Transmit : FIRE !  
location : yonginsi 351-00  
employee : 300  
type : ware house
```

I



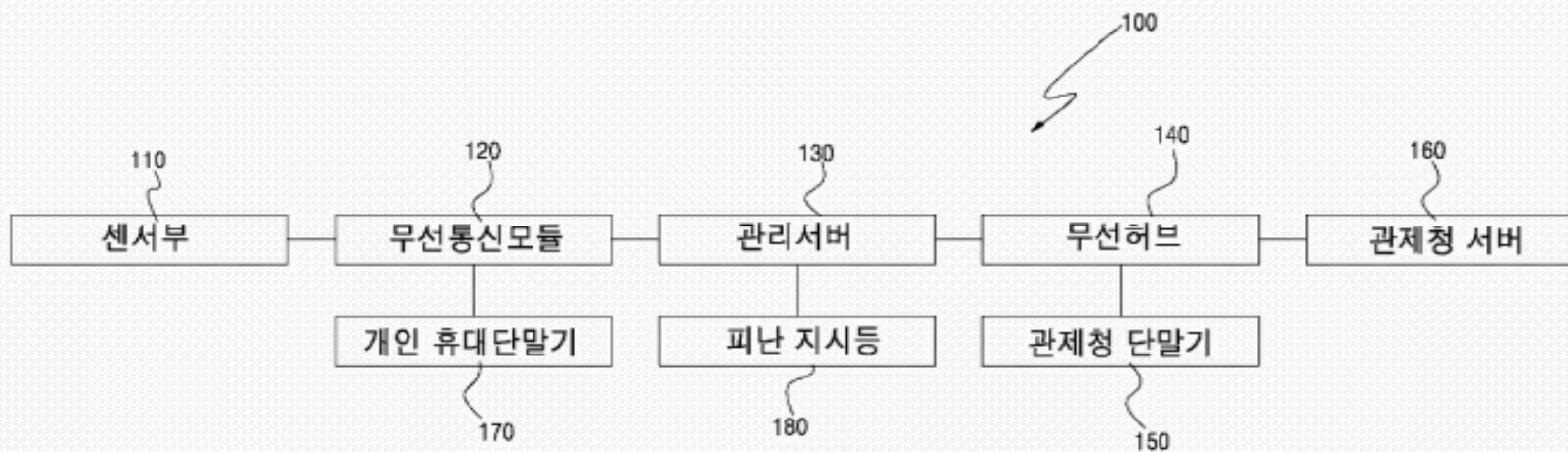
유사 프로젝트와 비교

- **발명의 명칭:** 건물 모델링 데이터를 이용한 화재감지시스템 및 화재대응방법
- **출원번호 :** 10-2014-0125280 (2001.07.12)
- **출원일자 :** 2014년 09월 19일 | IPC : G08B 17/06
- **출원인 :** 경일대학교 산학협력단 | 발명자/고안자 : 엄신조, 김근홍

(57) 요약

본 발명에 의한 건물 모델링 데이터를 이용한 화재감지시스템(100)은, 화재를 감지하기 위해 건물에 설치되도록 구성된 센서부(110)와, 상기 센서부(110)를 통해서 화재신호를 전송받고 탈출용 화재정보를 연산하여 발신하도록 구성된 관리서버(130)와, 상기 관리서버(130)와 통신하도록 구성된 것으로서 상기 탈출용 화재정보를 수신하도록 (뒷면에 계속)

대표도

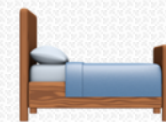




화재



지진



호텔



물류창고



회사



가정집