

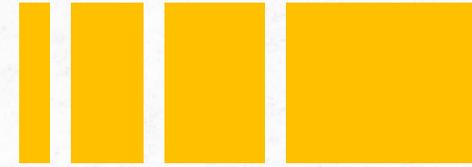


캡스톤 디자인2

IOT를 이용한 고기능 가스 누출 방지 시스템 설계

6조 유도우, 정성빈

목차 a table of contents



- 1 시스템 개요 및 기능
- 2 설계 사양
- 3 Block Diagram (입 · 출력)
- 4 회로도
- 5 Firmware 및 실험 결과
- 6 결론



◆ 주제

IOT 가스 누출 방지 시스템

◆ 개요

실시간 블루투스 가스 경보 알림 및 가스 농도 확인

LED 가스 경보 알림, 가스 밸브 자동 잠금

1 시스템 개요 및 기능 (2/2)

가스 누출 방지 시스템 기능(1/2)

무드등 및 블루투스 스피커 모드(평상시)

- 스마트폰 APP으로 LED 색상 조절 
- 블루투스 스피커 기능   Bluetooth
- 시간 및 실내 가스 농도(CO(일산화탄소)) 표시   
- 가스 농도를 스마트폰으로 확인 가능

가스 누출 방지 시스템 기능(2/2)

가스 경보 모드(가스 감지시)

- 가스 감지시 LED 색상이 적색으로 변함 
- LCD에 경고 표시가 출력되며 경보음 발생  
- 스마트폰 앱을 통해 가스 경보 알림 및 잠금 기능 (블루투스) 
- 가스 경보시 가스 밸브 잠금 기능 

1

제품 완성 도안



제품 외관(후면 & 측면)

※ 스위치 기능

1. LED 색상 조절 & 가스 경보 수동 정지 & 3초간 누르면 시간 변경 모드로 전환
2. 시간 변경 모드에서 시계의 시간(시) 조절 & 3초간 누르면 LED Gradation 모드
3. 시간 변경 모드에서 시계의 시간(분) 조절 & 3초간 누르면 블루투스 연결

사양



MCU : STM32F429ZIT6U

- CPU CORE : 32Bit / ARM[®] Cortex[®]-M4
- 최대 주파수 : 180MHz
- 메모리(FLASH) : 2MB
- 메모리(SRAM) : 256KB
- 동작 전압 : 1.8V ~ 3.6V
- 통신 : USART
- ADC / DAC : 12bit x 3 / 12bit x 2
- GPIO I/O 114개

2 설계 사양 (입력 사양)



모델명 : SFS10-5C

• 동작 전압 : 85 ~ 264 Vac

제품 이름 : AC / DC 컨버터

• 출력 전압, 전류 : 5 V, 2 A



모델명 : GSET11-P110

• 동작 전압 : 5 V

• 출력 전압 : 0.72V ~ 4.5V

제품 이름 : CO 센서

• 감지 농도 범위 : 0 ~ 250 ppm

• 정확도 : $\pm 10\%$

설계 사양 (출력 사양 1/2)



모델명 : LT043C-50A

제품 이름 : LCD Module

- 동작 전압 & 전류: DC 3.3 V, 155 mA
- 화면 크기, 해상도 : 4.3 인치, 480 x 272
- 인터페이스 : 8080 8bit Interface



모델명 : CJMCU - 60

제품 이름 : RGB LED 모듈

- 동작 전압 : 5 V
- 최대 소비 전류 : 500mA
- 램프 색상 : 빨간색, 녹색, 파란색(RGB)



모델명 : FIT0502

제품 이름 : 스피커

- 출력 정격 : 3 W
- 공칭 임피던스 : 8 Ω



모델명 : SY-SPK059

제품 이름 : 스피커

- 출력 정격 : 3 W
- 공칭 임피던스 : 4 Ω

설계 사양 (출력 사양 2/2)



모델명 : FSC-BT806

제품 이름 : 블루투스 5.0 모듈

- 동작 전압 : 3.3 V
- UART Interface 지원
- Stereo codec 지원
- 통신 거리 : 최대 40m



모델명 : PAM8403

제품 이름 : 디지털 앰프 모듈

- 동작 전압 : 5 V
- 4Ω 또는 8Ω 스피커 연결가능



모델명 : DFR0299

제품 이름 : SD카드 모듈

- 동작 전압 : 3.3 ~ 5 V
- UART Interface 지원

Block-Diagram

입력 장치

전원부

출력 장치

220V



SMPS (5 V, 2A) LDO(3.3V,1A)



CO 센서

5V
ADC



RX

TX

5V



SD카드 모듈

SPK L, SPK R

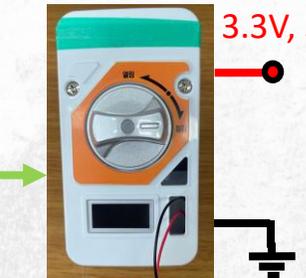


스피커

PWM



NPN-transistor



가스 밸브 잠금 장치

3.3V, 30mA

TX

RX



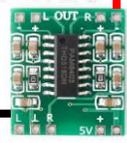
블루투스 모듈



애플리케이션

3.3V

5V



앰프

SPK L, SPK R



스피커

FMC



TFT-LCD

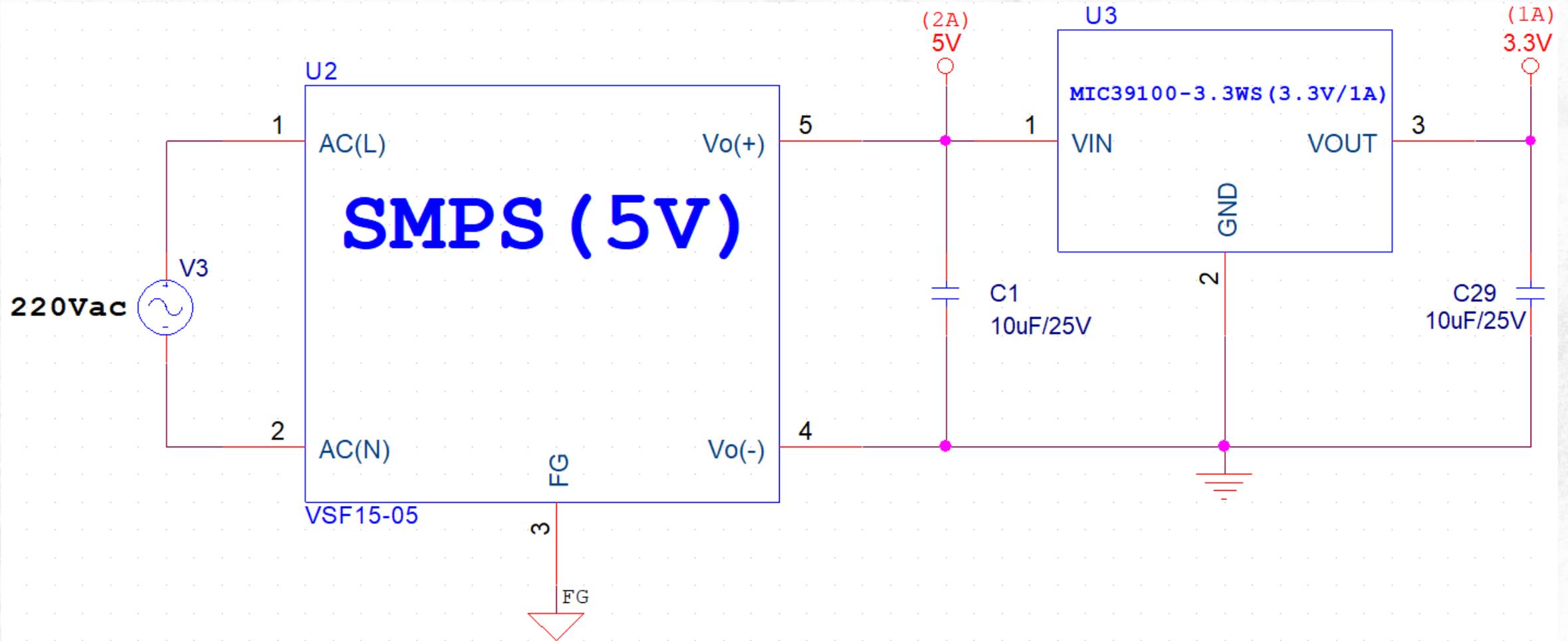
3.3V

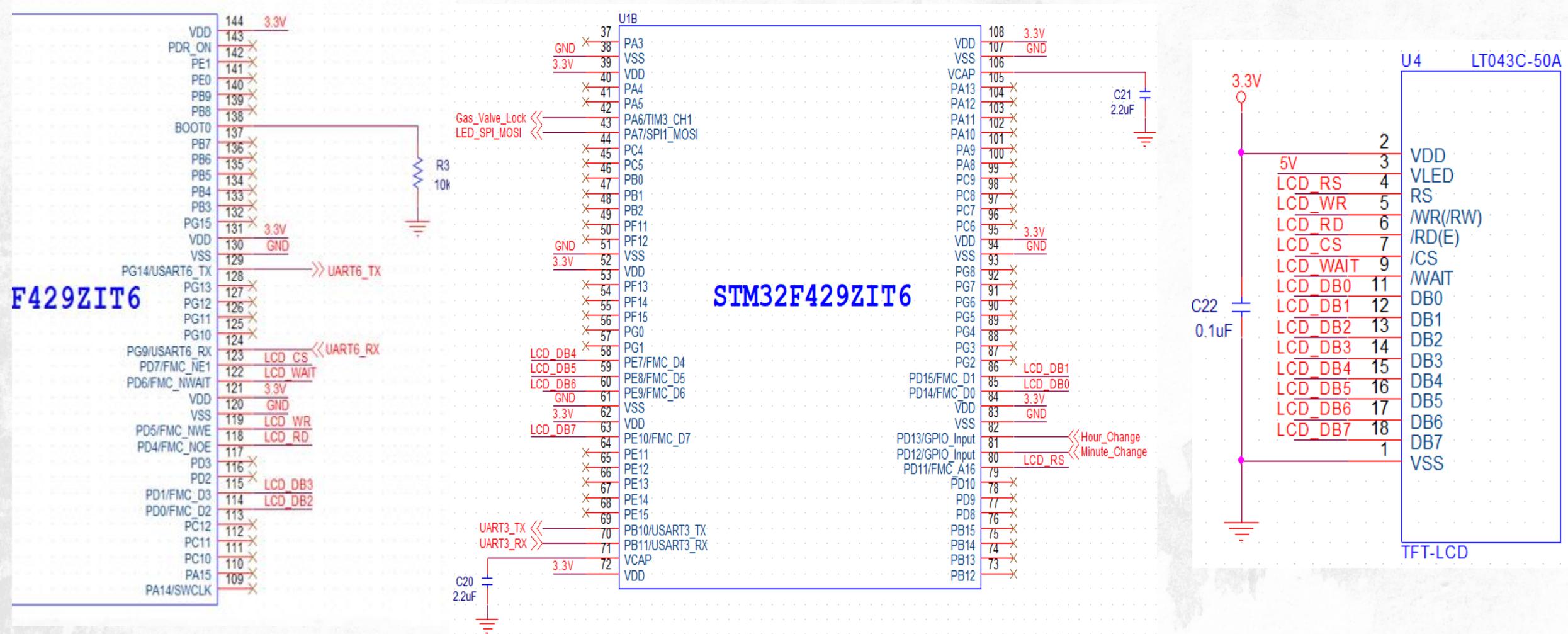
5V, 500mA

SPI

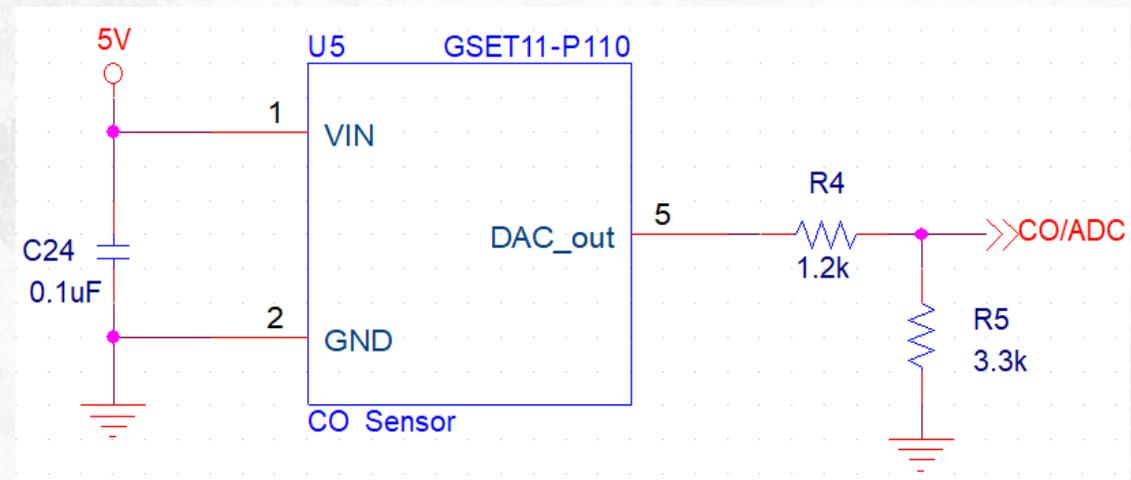
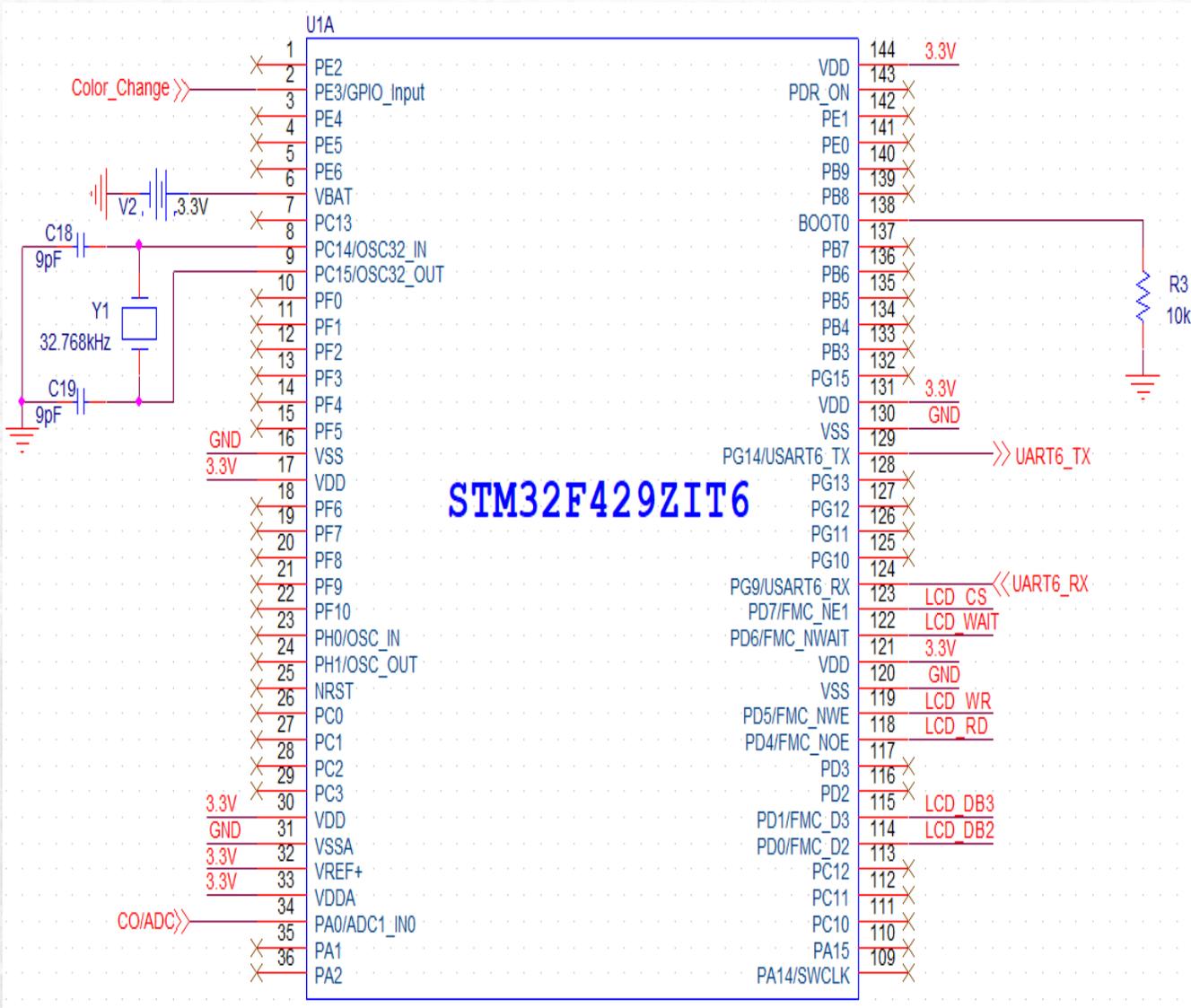


RGB LED

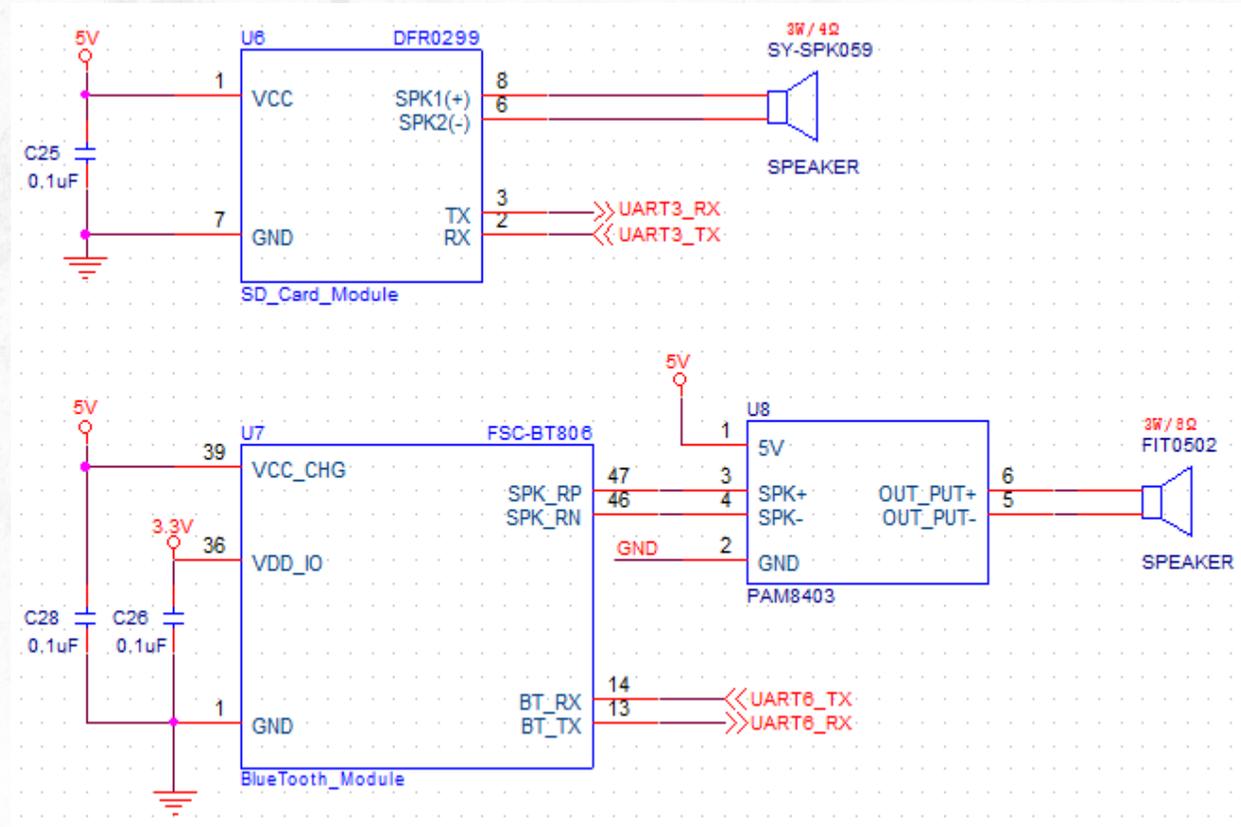
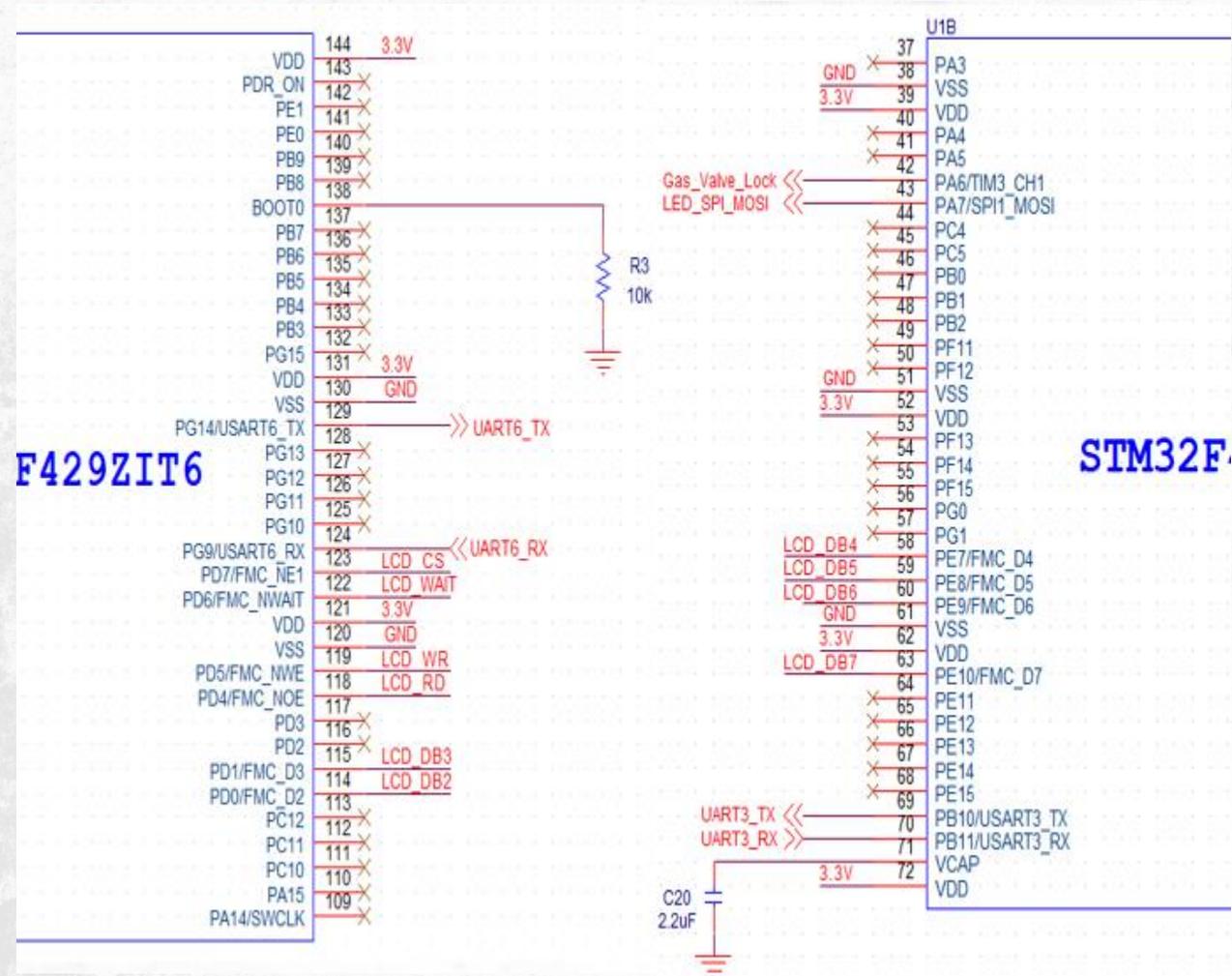


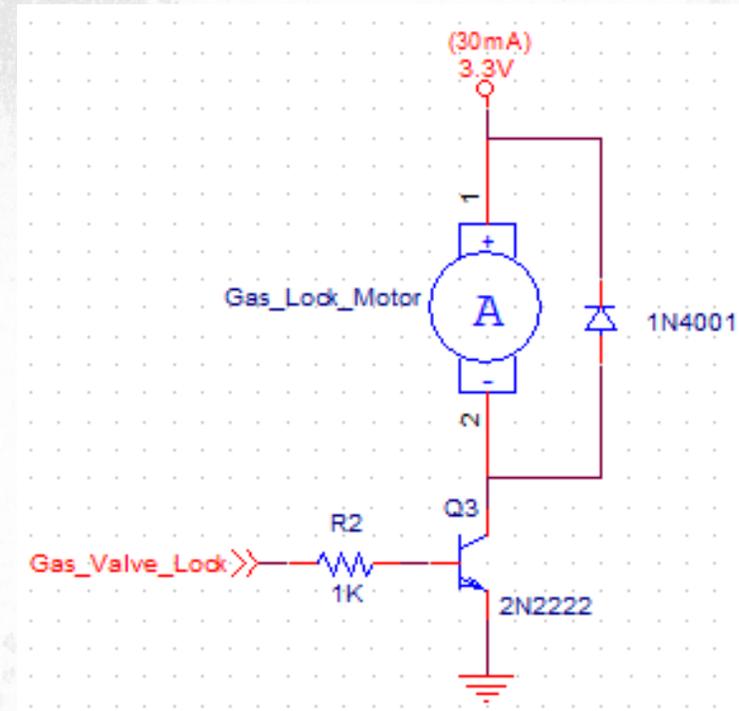
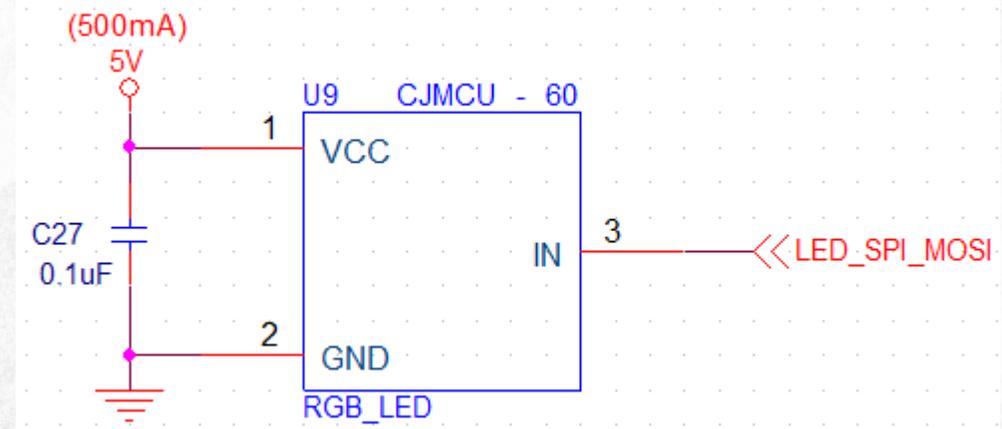
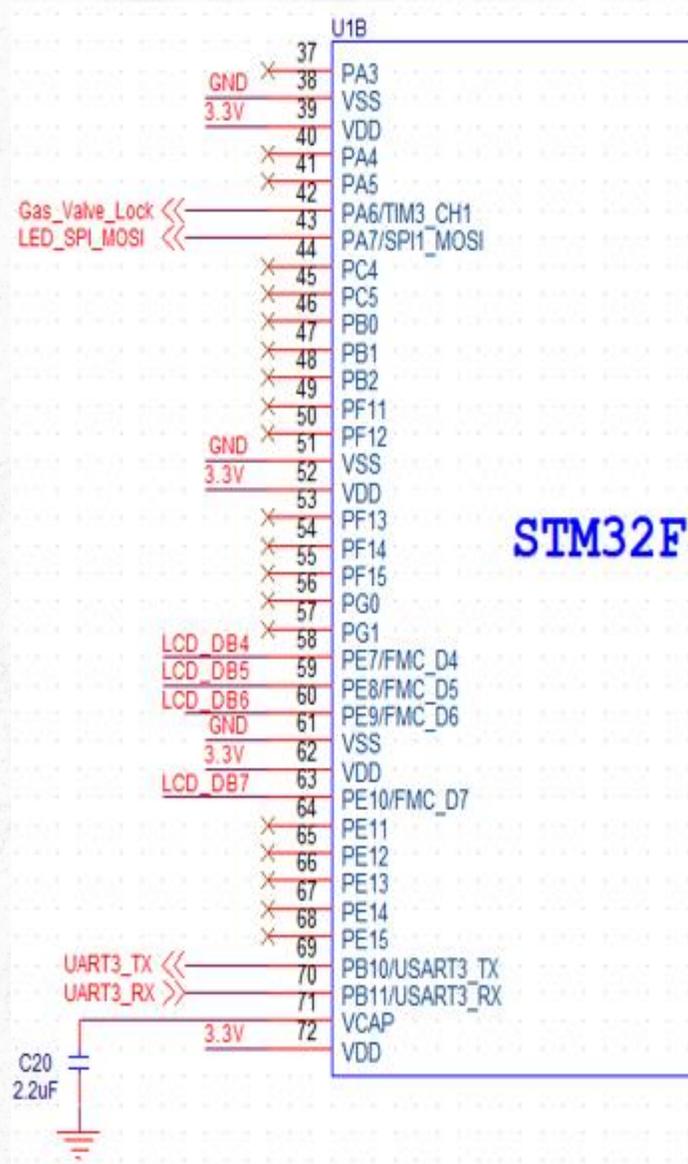


4 회로도 (CO_Sensor)



회로도 (SD_Card_Module, Bluetooth Speaker)





FirmWare 설명 (IAR) (CO 센서)

```

HAL_ADC_Start(&hadc1); // ADC 변환 시작

CO_adc1 = HAL_ADC_GetValue(&hadc1);

CO_final_adc += CO_adc1;

CO_adc2 = CO_final_adc / 1000.0;

//CO_AD_slop = (3300.0 - 720.0) / (4095.0 - 893.4545455);
//CO_AD_offset = 720.0 - (CO_AD_slop * 893.4545455);
//CO_AD_actual = (CO_AD_slop * CO_adc2) + CO_AD_offset;
//CO_gas = ((word)(CO_AD_actual) / 10) - 55; // CO 농도 값
//CO_gas = 7.448 - 19.757*(CO_adc2 / 910) + 16.388*(CO_adc2 / 910)*(CO_adc2 / 910) - 60;

static float CO_gas_test_num = 0;

CO_gas_test_num = CO_adc2/871;

CO_gas = 7.448 - (19.757*(CO_adc2 / 871)) + (16.388*(CO_adc2 / 871)*(CO_adc2 / 871));

```

$$\frac{3920}{4.5} = 871, \quad \frac{3920}{871} = 4.5V$$

(GSET11-P11X)

농도 (ppm)	출력 (Volt)	농도 (ppm)	출력 (Volt)
0	0.72	130	3.40
10	1.23	140	3.51
20	1.63	150	3.61
30	1.91	160	3.71
40	2.14	170	3.81
50	2.33	180	3.90
60	2.50	190	3.99
70	2.66	200	4.08
80	2.80	210	4.17
90	2.93	220	4.25
100	3.06	230	4.34
110	3.18	240	4.42
120	3.29	250	4.50

$$(ppm) = 7.448 - 19.757 * (Vout) + 16.388 * (Vout)^2$$

```

if(safe_discolor)
{
    static unsigned int reset_rx = 0;

    // 블루투스 어플리 버튼을 눌렀을 때 RX에서 받은 값에 따라 LED 색상이 변경 하는 부분
    for(int Ble_rx_cnt = 0; Ble_rx_cnt <= 1; Ble_rx_cnt++) // default
    {
        if(Re_cnt[Ble_rx_cnt] == 'i') // LED static mode
        {
            LED_Mode = 0;
            Re_cnt[0] = '\0';
            Re_cnt[1] = '\0';
            for(reset_rx = 0; reset_rx <= 249; reset_rx++)
            {
                RE_re_1[reset_rx] = '0';
            }
        }
        else if(Re_cnt[Ble_rx_cnt] == 'j') // LED gradation mode
        {
            LED_Mode = 1;
            Re_cnt[0] = '\0';
            Re_cnt[1] = '\0';
            for(reset_rx = 0; reset_rx <= 249; reset_rx++)
            {
                RE_re_1[reset_rx] = '0';
            }
        }
    }

    if(LED_Mode == 0)
    {
        if(Re_cnt[Ble_rx_cnt] == 'a')
        {
            Change_Count = 0;
            Re_cnt[0] = '\0';
            Re_cnt[1] = '\0';
            for(reset_rx = 0; reset_rx <= 249; reset_rx++)
            {
                RE_re_1[reset_rx] = '0';
            }
        }
    }
}

```

```

else if(LED_Mode == 1)
{
    if(Re_cnt[Ble_rx_cnt] == 'a')
    {
        Change_Count = 0;
        Re_cnt[0] = '\0';
        Re_cnt[1] = '\0';
        for(reset_rx = 0; reset_rx <= 249; reset_rx++)
        {
            RE_re_1[reset_rx] = '0';
        }
    }
    else if(Re_cnt[Ble_rx_cnt] == 'b') // white
    {
        Change_Count = 1;

        Re_cnt[0] = '\0';
        Re_cnt[1] = '\0';
        for(reset_rx = 0; reset_rx <= 249; reset_rx++)
        {
            RE_re_1[reset_rx] = '0';
        }
    }
    else if(Re_cnt[Ble_rx_cnt] == 'c') // green
    {
        Change_Count = 2;
        Re_cnt[0] = '\0';
        Re_cnt[1] = '\0';
        for(reset_rx = 0; reset_rx <= 249; reset_rx++)
        {
            RE_re_1[reset_rx] = '0';
        }
    }
    else if(Re_cnt[Ble_rx_cnt] == 'd') // blue
    {
        Change_Count = 3;
        Re_cnt[0] = '\0';
        Re_cnt[1] = '\0';
        for(reset_rx = 0; reset_rx <= 249; reset_rx++)
        {
            RE_re_1[reset_rx] = '0';
        }
    }
}

```

```

switch(Change_Count)
{
    case 0 :
        WS2812_SetAll(Red_Flag,Green_Flag,Blue_Flag); // 꺼짐 상태
        HAL_SPI_Transmit_DMA(&hspi1, ws2812_buffer, WS2812_BUFFER_SIZE);
        break;

    case 1 :
        //WS2812_SetAll(33,33,33); // White
        WS2812_SetAll(Red_Flag,Green_Flag,Blue_Flag);
        HAL_SPI_Transmit_DMA(&hspi1, ws2812_buffer, WS2812_BUFFER_SIZE);
        break;

    case 2 :
        //WS2812_SetAll(0,100,0); // Green
        WS2812_SetAll(Red_Flag,Green_Flag,Blue_Flag);
        HAL_SPI_Transmit_DMA(&hspi1, ws2812_buffer, WS2812_BUFFER_SIZE);
        break;

    case 3 :
        //WS2812_SetAll(0,0,100); // Blue
        WS2812_SetAll(Red_Flag,Green_Flag,Blue_Flag);
        HAL_SPI_Transmit_DMA(&hspi1, ws2812_buffer, WS2812_BUFFER_SIZE);
        break;

    case 4 :
        //WS2812_SetAll(50,0,50); // Purple

```

```

if(RE_re_1[R] == '@' || RE_re_1[R+1] == '@') // CO센서 데이터를 블루투스 모듈로 보내는 부분
{
    if (CO_TX >= 1)
    {
        CO_TX = 0;
        if(CO_gas >= 100)
        {
            sprintf(CO_tr_1, "AT+SPPSEND=3,%03d\r\n", CO_gas);
            HAL_UART_Transmit_IT(&huart6, CO_tr_1, 18);
            for(int reset_RE_re_1 = 0; reset_RE_re_1 <=249; reset_RE_re_1++)
            {
                RE_re_1[reset_RE_re_1] = '\0';
            }
        }
        else if(CO_gas >= 10 && CO_gas < 100)
        {
            sprintf(CO_tr_1, "AT+SPPSEND=3,%02d\r\n", CO_gas);
            HAL_UART_Transmit_IT(&huart6, CO_tr_1, 17);
            for(int reset_RE_re_1 = 0; reset_RE_re_1 <=249; reset_RE_re_1++)
            {
                RE_re_1[reset_RE_re_1] = '\0';
            }
        }
        else if(CO_gas < 10)
        {
            sprintf(CO_tr_1, "AT+SPPSEND=3,%01d\r\n", CO_gas);
            HAL_UART_Transmit_IT(&huart6, CO_tr_1, 16);
            for(int reset_RE_re_1 = 0; reset_RE_re_1 <=249; reset_RE_re_1++)
            {
                RE_re_1[reset_RE_re_1] = '\0';
            }
        }
    }
}
}

```

```

for(int Ble_rx_cnt_flag = 0; Ble_rx_cnt_flag <= 1; Ble_rx_cnt_flag++)
{
    if(Re_cnt[Ble_rx_cnt_flag] == 'h')
    {
        stop_check_sum = stop_tr_1[1]+stop_tr_1[2]+stop_tr_1[3]+stop_tr_1[4]+stop_tr_1[5]+stop_tr_1[6];
        stop_final_check_sum = ~stop_check_sum + 1; // check_sum의 2의 보수 값
        stop_tr_1[7] = (byte)(stop_final_check_sum >> 8); // final_check_sum은 int이므로 byte로 바꾸어주어야함.
        stop_tr_1[8] = (byte)(stop_final_check_sum);
        HAL_UART_Transmit_IT(&huart3, stop_tr_1, 10);

        HAL_UART_Receive_IT(&huart3, stop_re_1, 10);

        HAL_TIM_PWM_Stop(&htim3, TIM_CHANNEL_1);

        Re_cnt[0] = '\0';
        Re_cnt[1] = '\0';
        for(int reset_rx_flag = 0; reset_rx_flag <= 249; reset_rx_flag++)
        {
            RE_re_1[reset_rx_flag] = '\0';
        }
    }
}
}

```

```

static_mode_btn.setOnClickListener {
    if (isDeviceConnected(targetDeviceName)) {
        soundPool.play(sound, leftVolume: 1.0f, rightVolume: 1.0f, priority: 0, loop: 0, rate: 1.0f)
        Toast.makeText(
            context: this,
            text: "Static Mode",
            Toast.LENGTH_SHORT
        ).show()

        // BluetoothService 종료
        val stopIntent = Intent( packageContext: this, BluetoothService::class.java)
        stopService(stopIntent)

        val intent = Intent( packageContext: this, BluetoothService::class.java)
        intent.putExtra( name: "device", device)
        intent.putExtra( name: "static_mode", value: 1)
        startService(intent)
    }else {
        soundPool.play(sound, leftVolume: 1.0f, rightVolume: 1.0f, priority: 0, loop: 0, rate: 1.0f)
        Toast.makeText(
            context: this,
            text: "IOT_Gas_Valve_6와 블루투스 연결되지 않았습니다.",
            Toast.LENGTH_SHORT
        ).show()
    }
}

```

```

}else if(static_mode_data == 1){
    // 열 번째 데이터 전송 작업
    scope.launch {
        sendDataPeriodically_static_mode()
    }
}else if(gradation_mode_data == 1){
    // 열 한 번째 데이터 전송 작업
    scope.launch {
        sendDataPeriodically_gradation_mode()
    }
}

```

```

// static mode 데이터를 전송하는 함수
private fun sendDataPeriodically_static_mode() {
    sendDataOverBluetooth( data: "iiiiiiii")
    Log.d( tag: "데이터 전송 완료 : ", msg: "iiiiiiii")
}

```

```

// gradation mode 데이터를 전송하는 함수
private fun sendDataPeriodically_gradation_mode() {
    sendDataOverBluetooth( data: "jjjjjjjj")
    Log.d( tag: "데이터 전송 완료 : ", msg: "jjjjjjjj")
}

```

◆ 제품 외형

※ 가스 밸브 잠금 장치
수동으로 열림 / 잠김 가능
가스 경보 시 가스 밸브 잠금

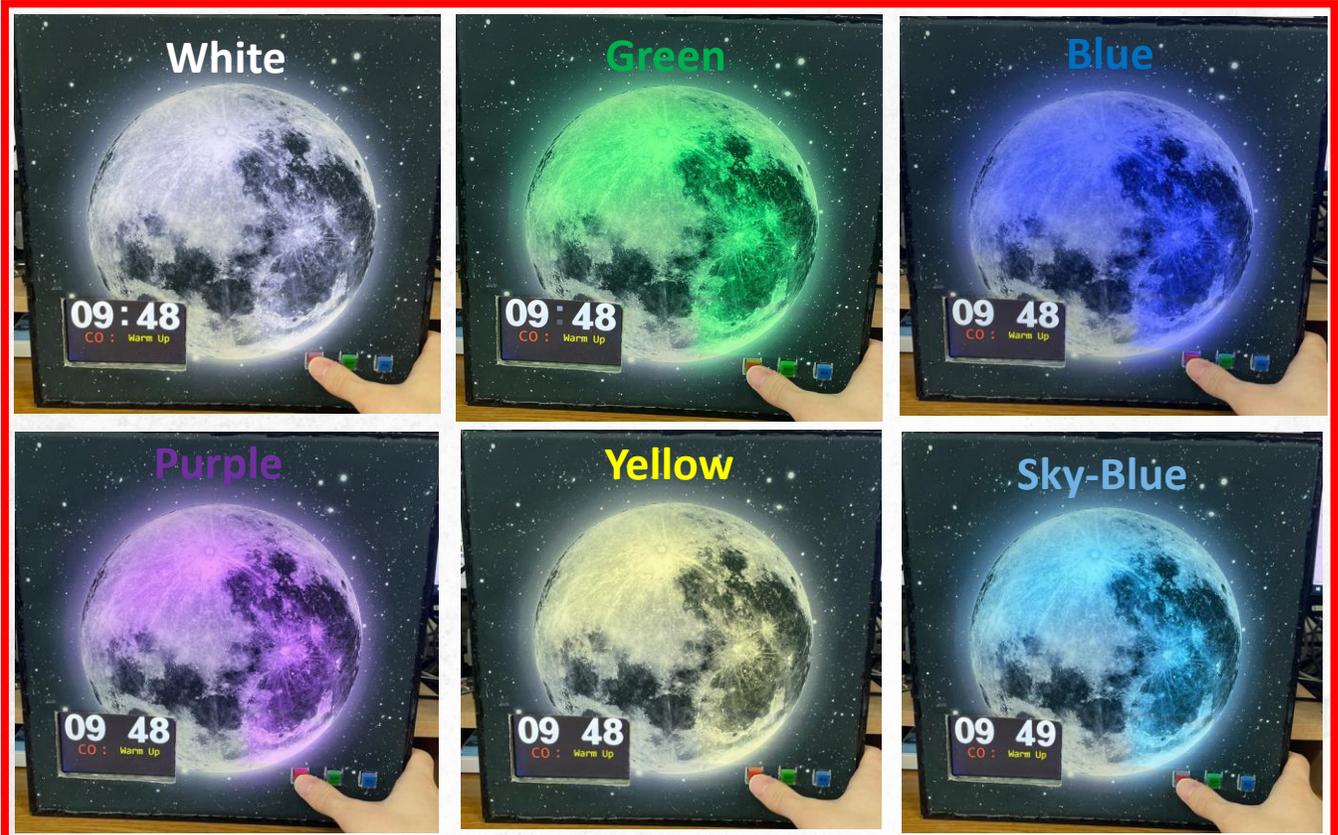


※ 스위치 기능

1. LED 색상 조절 & 가스 경보 수동 정지 & 3초간 누르면 시간 변경 모드
2. 시간 변경 모드에서 시계의 시간(시) 조절 & LED Gradation 모드
3. 시간 변경 모드에서 시계의 시간(분) 조절 & 3초간 누르면 블루투스 연결

실험 결과 (IOT 시스템) (2/5)

◆ CO 농도 30 ppm 미만 (평상시)



※ 스위치 기능
 SW1. LED 색상 조절가능 총 6가지 색상
 (White, Green, Blue, Purple, Yellow, Sky Blue)

Time Change Mode(시간 변경 모드)

※ 스위치 기능 → 한번 더 누르면 시간 저장 가능

1. 3초간 누르면 시간 변경 모드
2. 시간 변경 모드에서 시계의 시간(시) 조절
3. 시간 변경 모드에서 시계의 시간(분) 조절

5

실험 결과 (IOT 시스템) (3/5)

◆ CO 농도 30 ppm 미만 (정상시)



※ 스위치 기능
SW2 : 3초간 누르면 **Gradation Mode**

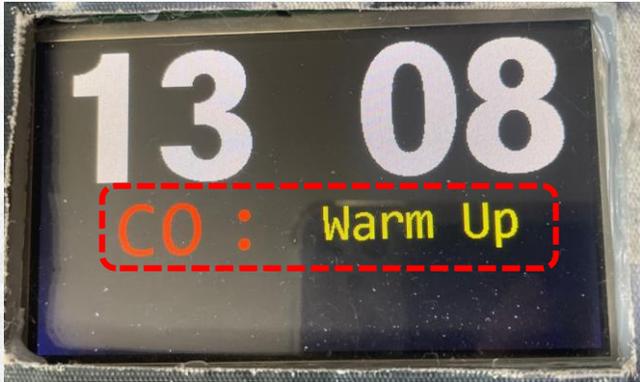


※ 스위치 기능
SW2 : 3초간 누르면 **Static Mode**

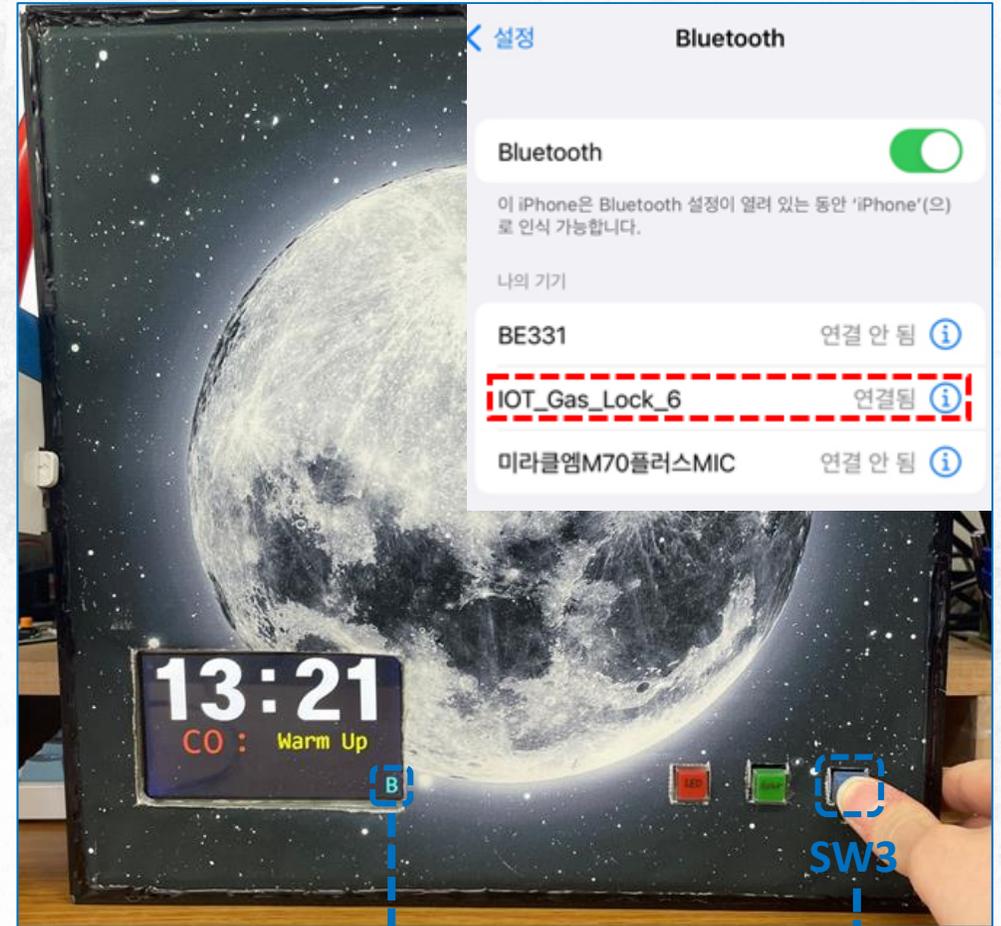
5

실험 결과 (IOT 시스템) (4/5)

◆ CO 농도 30 ppm 미만 (평상시)



CO 센서 Warming up 시간이 5분이기에 5분후에 CO 농도가 출력된다.



※ 스위치 기능
SW3 : 3초간 누르면 블루투스 연결(음악 재생가능)

5

실험 결과 (IOT 시스템) (5/5)

◆ CO 농도 30 ppm 이상 (가스경보시)

블루투스 연결이 해제되고, 스피커를 통해 가스 경보음을 재생한다.



LED 색상이 적색으로 출력

가스 경보시
가스 밸브 자동 잠금

※ 가스 누출 감지
※ CO : 22 ppm

SW1

※ 스위치 기능
SW1. 가스 경보음 수동 정지 버튼

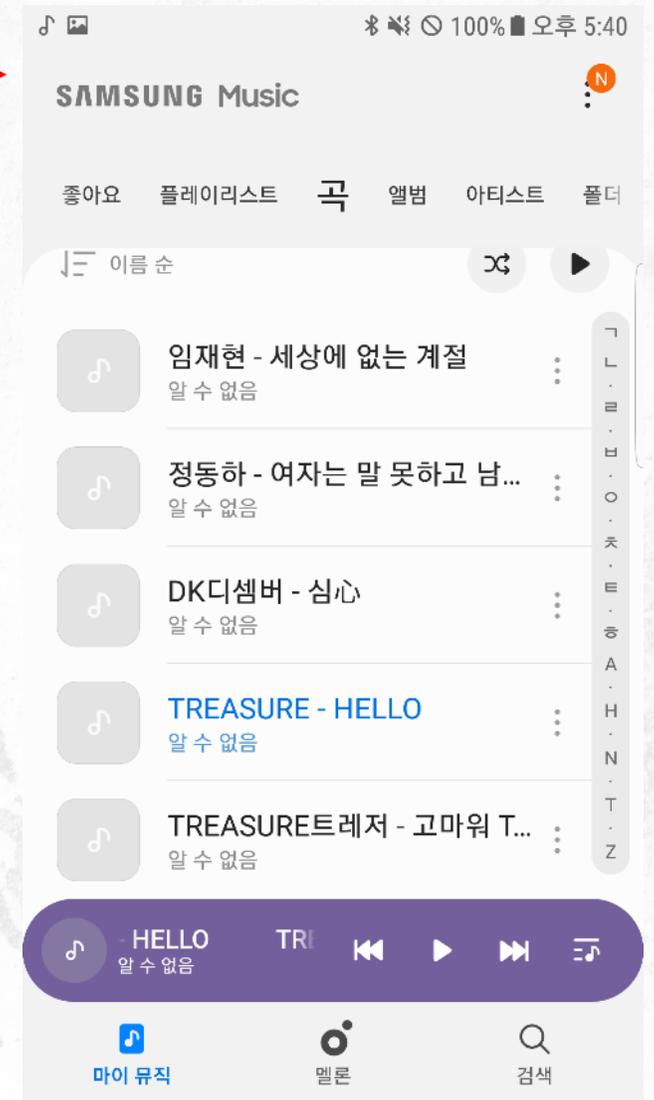
LCD에 가스 누출 감지 화면 및 CO 농도 출력



APP 아이콘

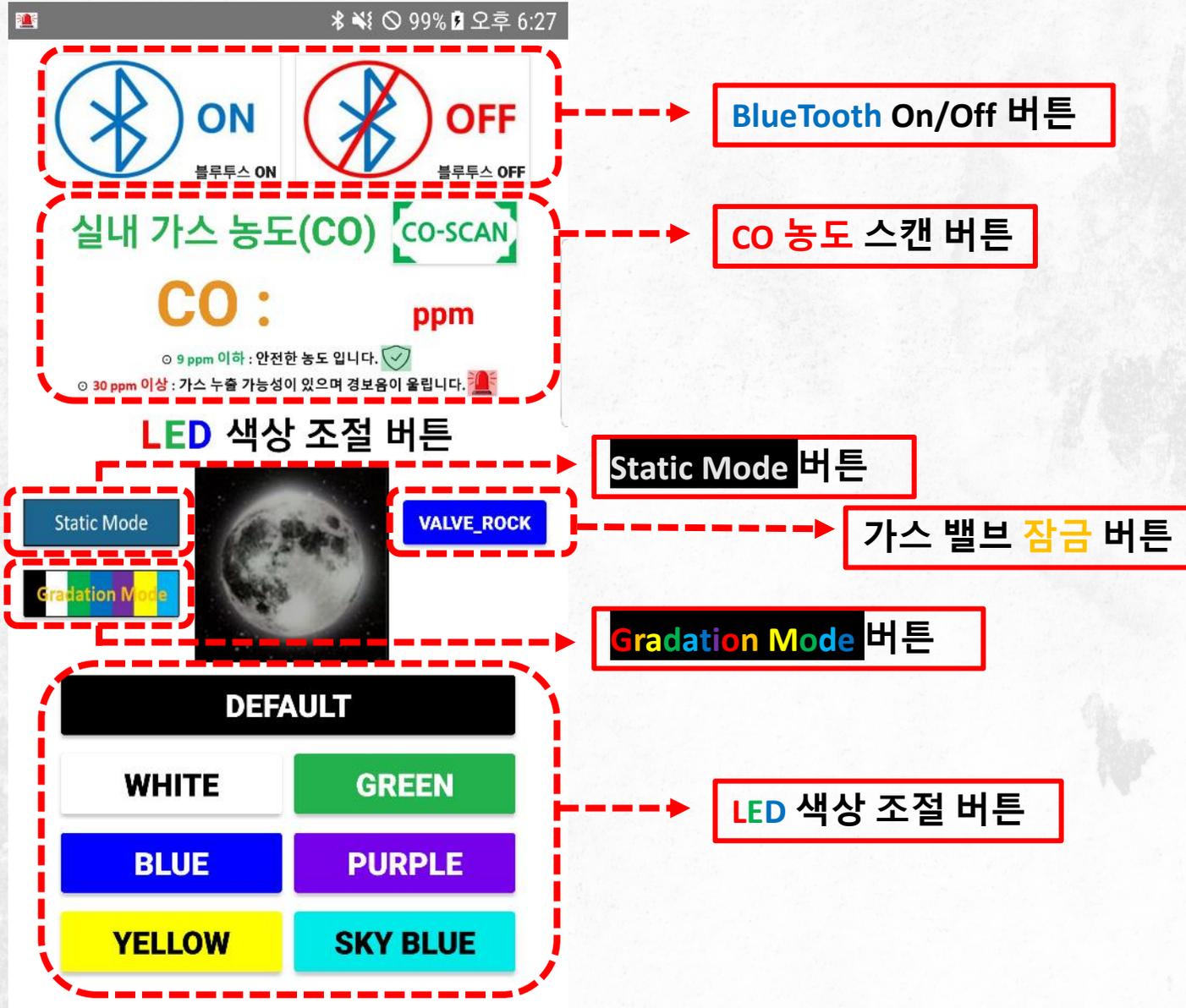


시작 버튼을 누르면 다음 화면으로 이동

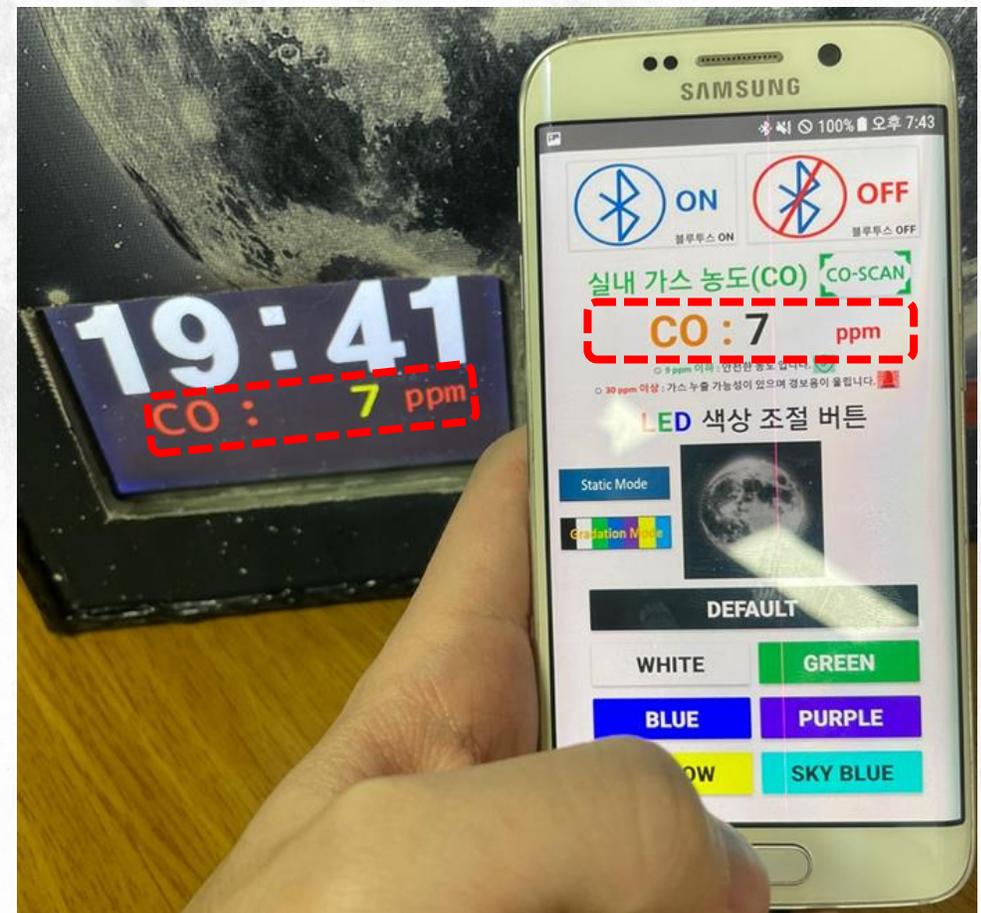


음악 버튼을 누르면 스마트폰의 음악으로 이동

실험 결과 (애플리케이션) (2/4)



◇ CO 농도 30 ppm 미만 (평상시)



실시간으로 CO 농도 모니터링

실험 결과 (애플리케이션) (3/4)

◆ CO 농도 30 ppm 미만 (정상시)

LED 색상 조절 버튼으로 색상 변경 가능
LED 모드 (Static Mode, Gradation Mode) 변경 가능

LED 색상 조절가능 총 6가지 색상
(White, Green, Blue, Purple, Yellow, Sky Blue)



실험 결과 (애플리케이션) (4/4)

◆ CO 농도 30 ppm 이상 (가스경보시)

Bluetooth ON / OFF

실내 가스 농도(CO) [CO-SCAN]

CO : 162 ppm

9 ppm 이하 : 안전한 농도입니다.

30 ppm 이상 : 가스 누출 가능성이 있으며 경보음이 울립니다.

LED 색상 조절 버튼

Static Mode / Gradation Mode

STOP

※ 가스 누출 감지
실내 가스 농도가 높습니다.
자동으로 가스 밸브를 잠급니다.

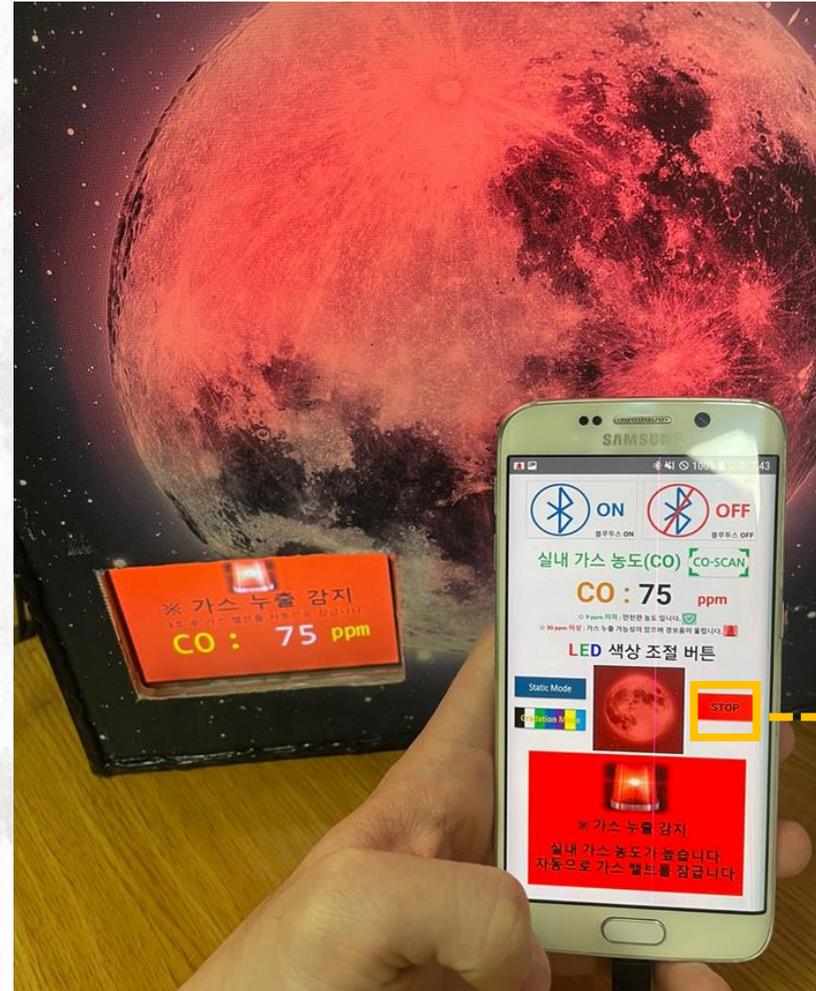
오후 7:44 | 10월 30일 (월)

IOT Gas Valve 7:44

가스 누출 알림!
실내 가스 농도가 높습니다! 자동으로 가스 밸브를 잠급니다!

알림 차단 / 모두 지우기

※ 가스 누출 감지
실내 가스 농도가 높습니다.
자동으로 가스 밸브를 잠급니다.
긴급전화만 가능합니다.



스마트폰 진동과 함께 가스 누출 경고 알림이 생성됨

- 시중에 가스 누출을 방지하기 위한 제품들이 많이 나와있지만, 블루투스 스피커 및 LED 무드등 기능을 하며 **칼라 TFT-LCD가 부착되어 CO 농도를 출력해주는 제품은 없음**
- 이 제품은 실내 CO 농도를 TFT-LCD를 통해 확인할 수 있고, 블루투스 모듈을 사용하여 MCU와 애플리케이션간 무선 통신을 통해 실시간으로 CO 농도를 확인할 수 있다
- 또한, 실내 CO 농도가 30ppm 이상이면 가스 누출 경보음과 함께, 가스 잠금 시스템의 모터가 동작하여 가스 밸브를 잠가 가스 누출을 방지할 수 있음
- 시중에서 판매되고 있는 가스 누출 방지 제품에 비해 부피가 큰 단점이 있지만, 가스 누출 방지 외 다양한 기능들로 실생활 속에서 실용적으로 사용가능 할 것임

감사합니다!