

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

1 시스템 개요 및 기능(1/2)



◆ 주제

IOT 가스 누출 방지 시스템

◆ 개요

실시간 블루투스 가스 경보 알림 및 가스 농도 확인

LED 가스 경보 알림, 가스 밸브 자동 잠금

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



- 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)



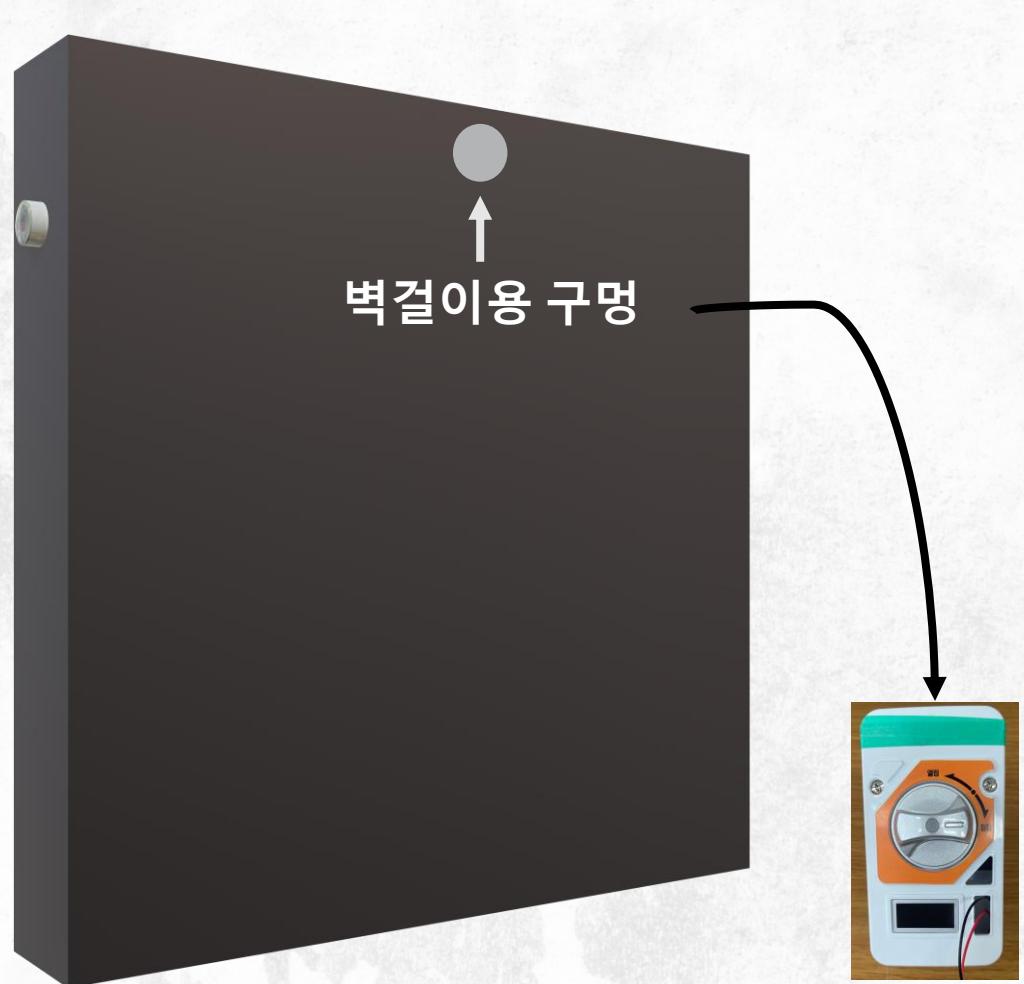
- 가스 경보시 가스 밸브 잠금 기능



제 품 완 성 도 안



제품 외관(정면 & 측면)



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

2 설계 사양(MCU 사양)



MCU : STM32F429ZIT6U

사양

- 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

설계 사양(입력 사양)



모델명 : SFS10-5C

제품 이름 : AC / DC 컨버터 • 출력 전압, 전류 : 5 V, 2 A

• 동작 전압 : 85 ~ 264 Vac



모델명 : GSET11-P110

제품 이름 : CO 센서

- 동작 전압 : 5 V
- 출력 전압 : 0.72V ~ 4.5V
- 감지 농도 범위 : 0 ~ 250 ppm
- 정확도 : ±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 연결 가능

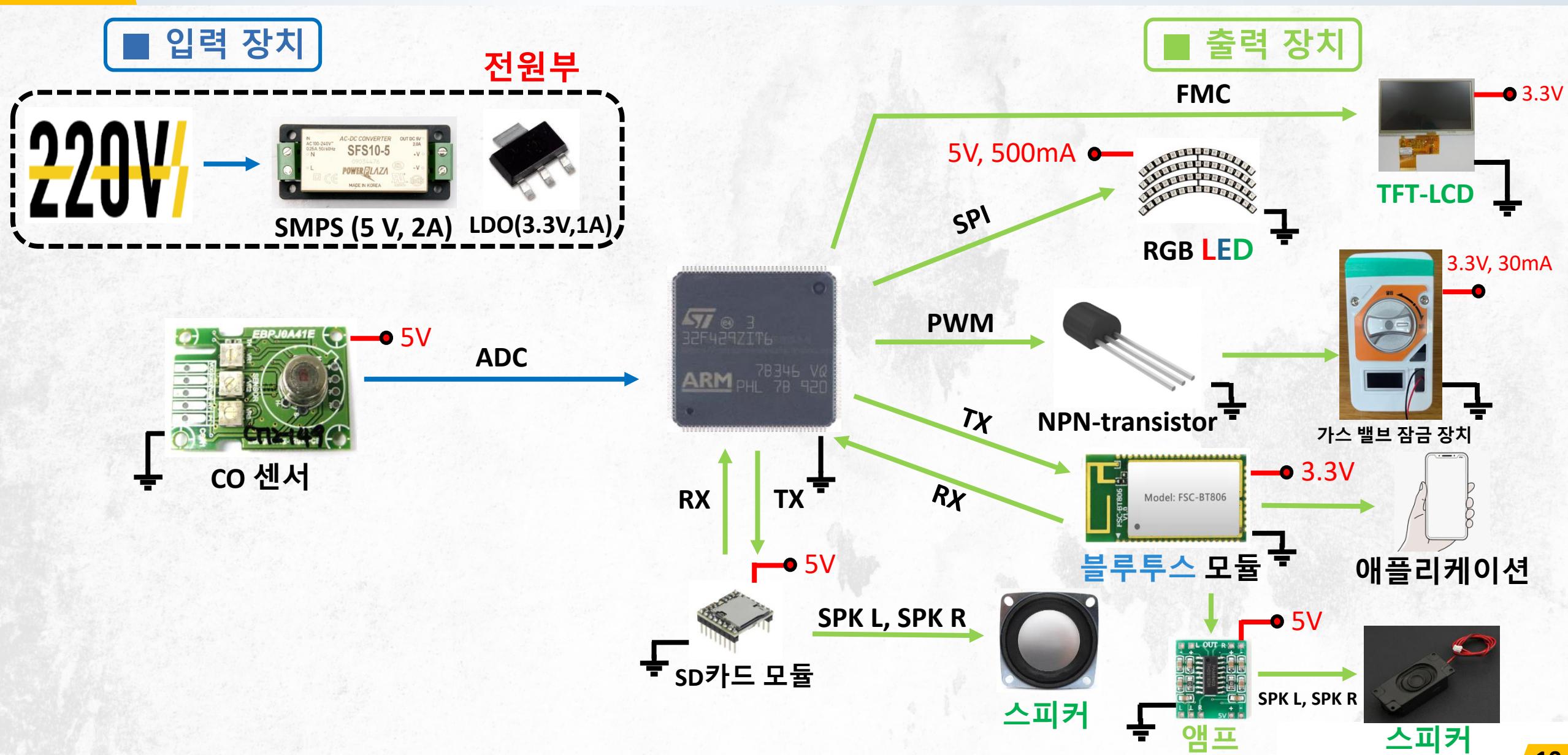


모델명 : DFR0299

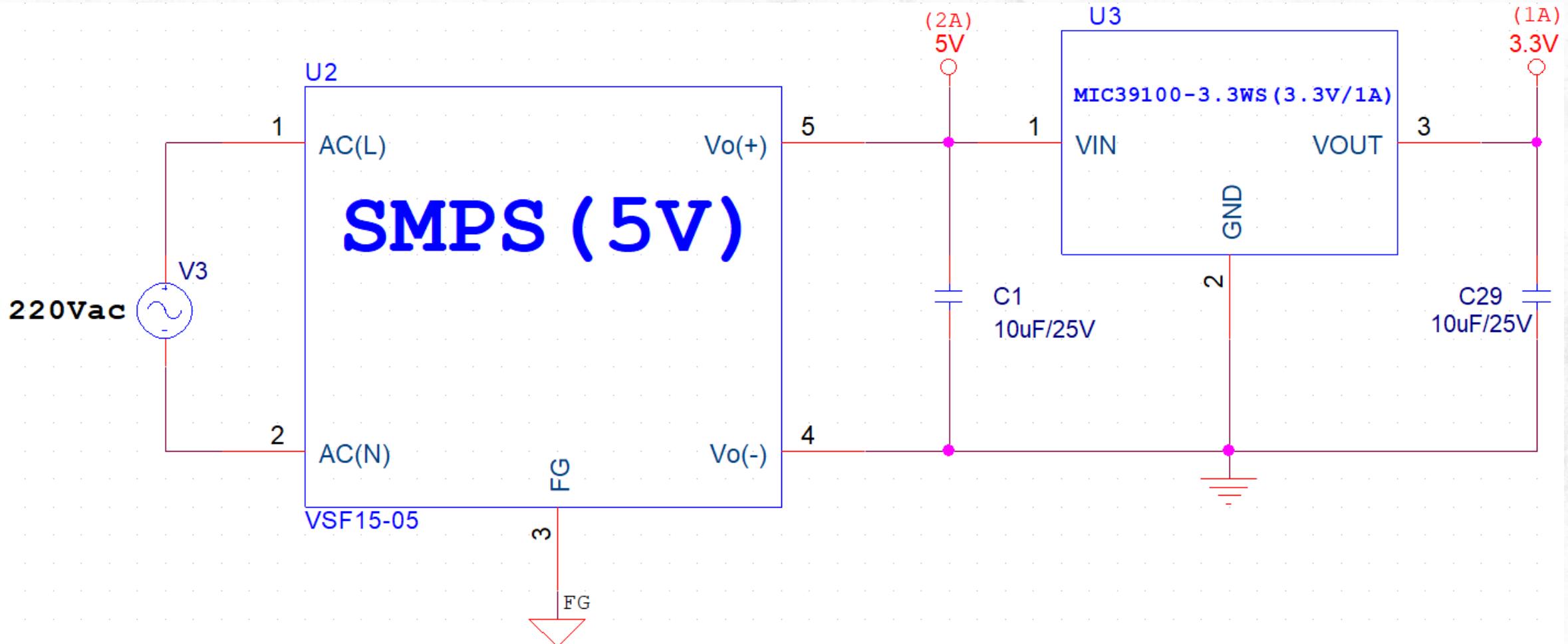
제품 이름 : SD카드 모듈

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

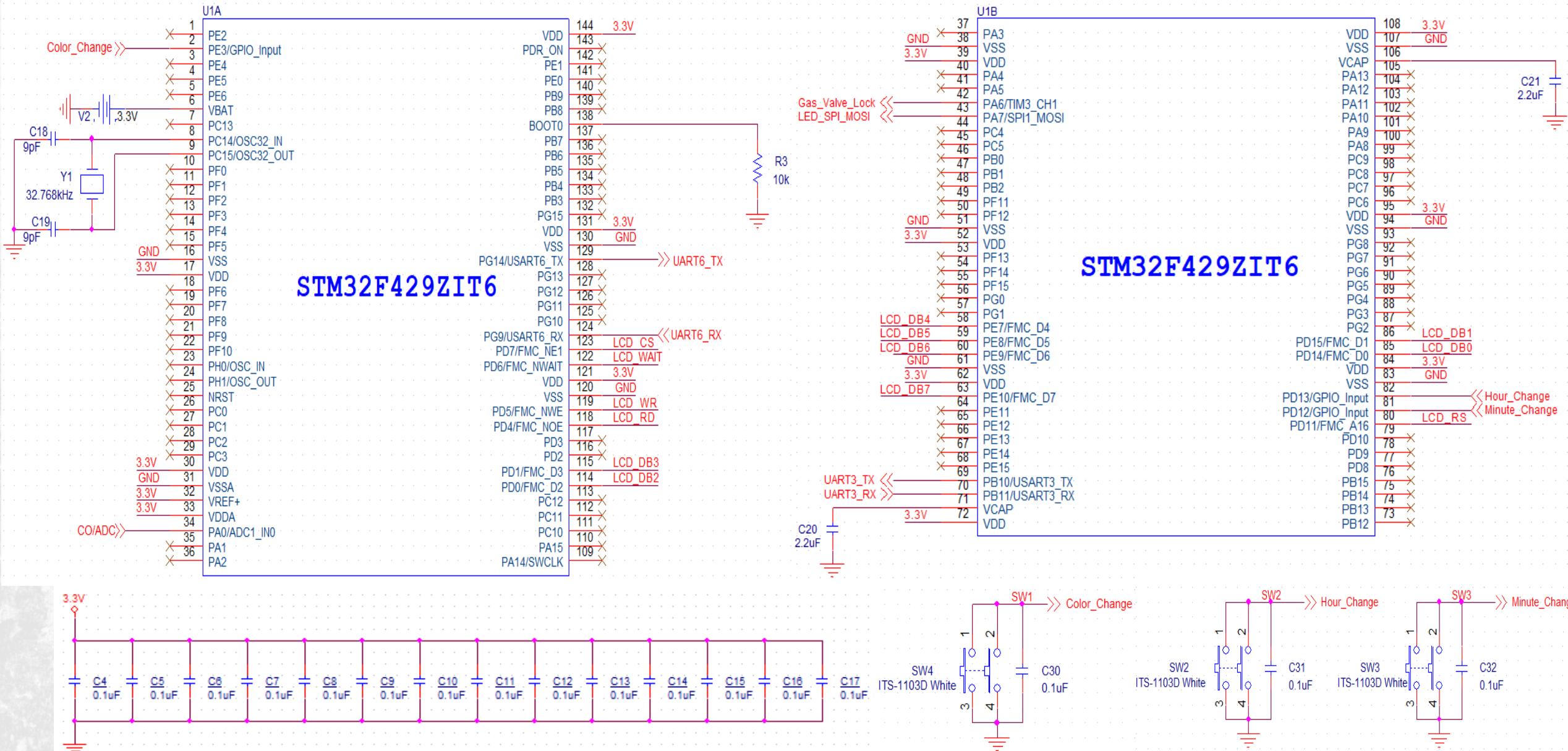
Block-Diagram



회로도 (Power)

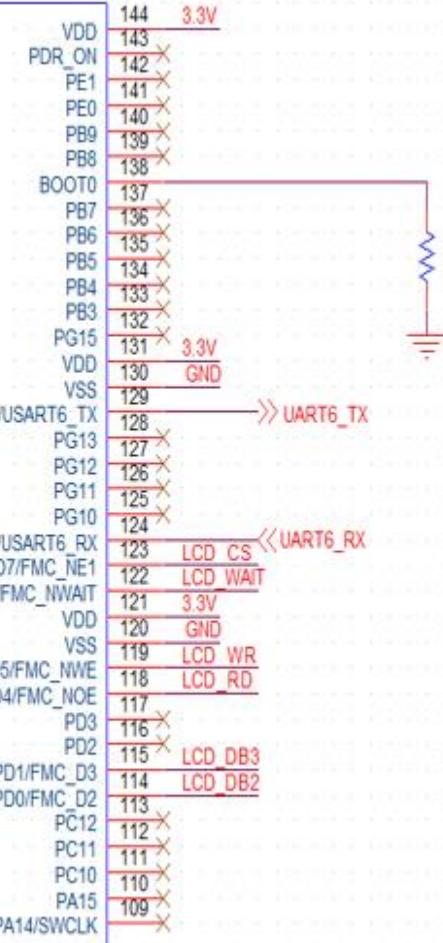


회로도 (CPU, Switch)



회로도 (TFT-LCD)

F429ZIT6



Gas_Valve_Lock

LED_SPI_MOSI

UART3_RX

UART3_TX

3.3V

C20

2.2uF

R3

10k

GND

3.3V

C21

2.2uF

GND

3.3V

VDD

VSS

LCD_DB7

63

GND

3.3V

VDD

VSS

3.3V

PA14

PA13

PA12

PA11

PA10

PA9

PA8

PA7

PA6

PA5

PA4

PA3

PA2

PA1

PA0

PC4

PC5

PC6

PC7

PC8

PC9

PC10

PC11

PC12

PC13

PC14

PC15

PC16

PC17

PC18

PC19

PC20

PC21

PC22

PC23

PC24

PC25

PC26

PC27

PC28

PC29

PC30

PC31

PC32

PC33

PC34

PC35

PC36

PC37

PC38

PC39

PC40

PC41

PC42

PC43

PC44

PC45

PC46

PC47

PC48

PC49

PC50

PC51

PC52

PC53

PC54

PC55

PC56

PC57

PC58

PC59

PC60

PC61

PC62

PC63

PC64

PC65

PC66

PC67

PC68

PC69

PC70

PC71

PC72

PC73

PC74

PC75

PC76

PC77

PC78

PC79

PC80

PC81

PC82

PC83

PC84

PC85

PC86

PC87

PC88

PC89

PC90

PC91

PC92

PC93

PC94

PC95

PC96

PC97

PC98

PC99

PC100

PC101

PC102

PC103

PC104

PC105

PC106

PC107

PC108

PC109

PC110

PC111

PC112

PC113

PC114

PC115

PC116

PC117

PC118

PC119

PC120

PC121

PC122

PC123

PC124

PC125

PC126

PC127

PC128

PC129

PC130

PC131

PC132

PC133

PC134

PC135

PC136

PC137

PC138

PC139

PC140

PC141

PC142

PC143

PC144

PC145

PC146

PC147

PC148

PC149

PC150

PC151

PC152

PC153

PC154

PC155

PC156

PC157

PC158

PC159

PC160

PC161

PC162

PC163

PC164

PC165

PC166

PC167

PC168

PC169

PC170

PC171

PC172

PC173

PC174

PC175

PC176

PC177

PC178

PC179

PC180

PC181

PC182

PC183

PC184

PC185

PC186

PC187

PC188

PC189

PC190

PC191

PC192

PC193

PC194

PC195

PC196

PC197

PC198

PC199

PC200

PC201

PC202

PC203

PC204

PC205

PC206

PC207

PC208

PC209

PC210

PC211

PC212

PC213

PC214

PC215

PC216

PC217

PC218

PC219

PC220

PC221

PC222

PC223

PC224

PC225

PC226

PC227

PC228

PC229

PC230

PC231

PC232

PC233

PC234

PC235

PC236

PC237

PC238

PC239

PC240

PC241

PC242

PC243

PC244

PC245

PC246

PC247

PC248

PC249

PC250

PC251

PC252

PC253

PC254

PC255

PC256

PC257

PC258

PC259

PC260

PC261

PC262

PC263

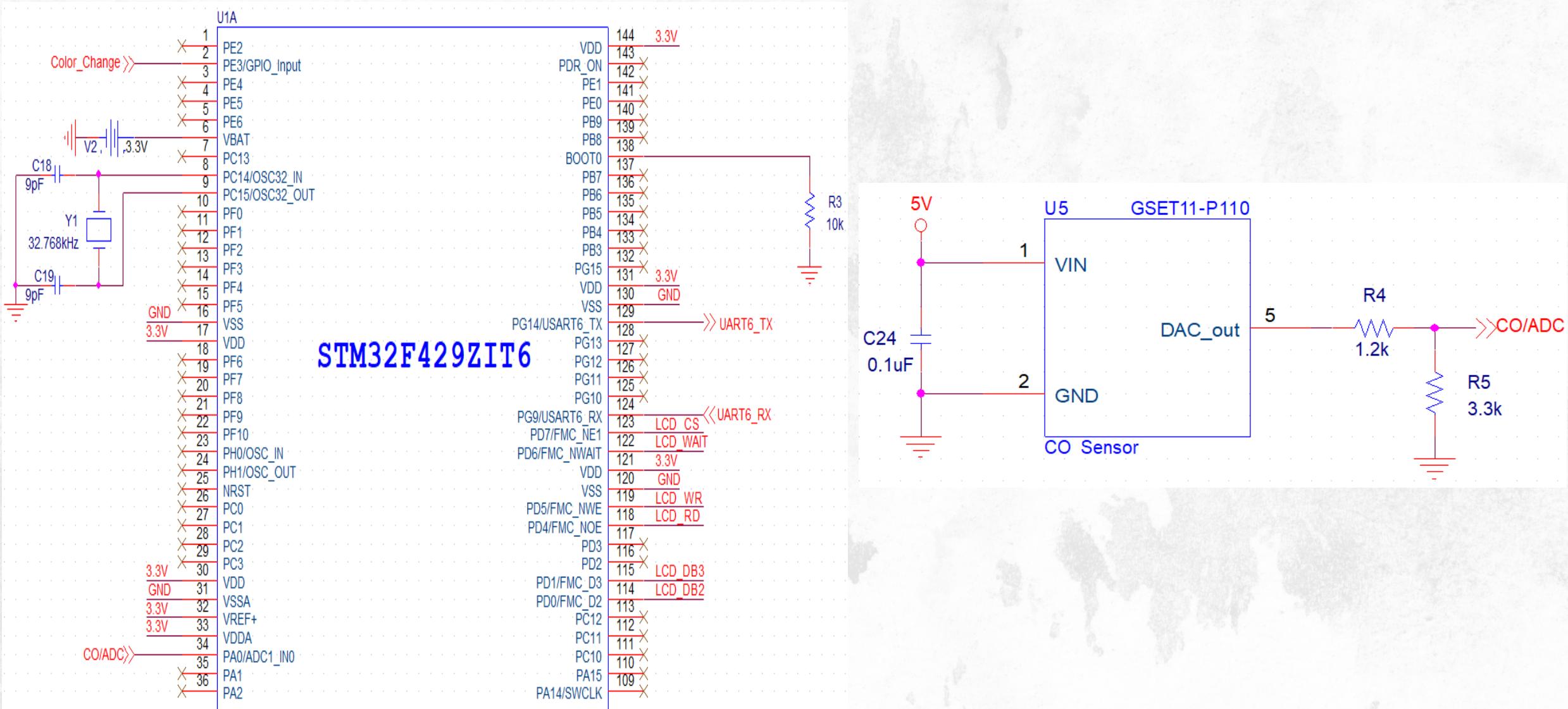
PC264

PC265

PC266

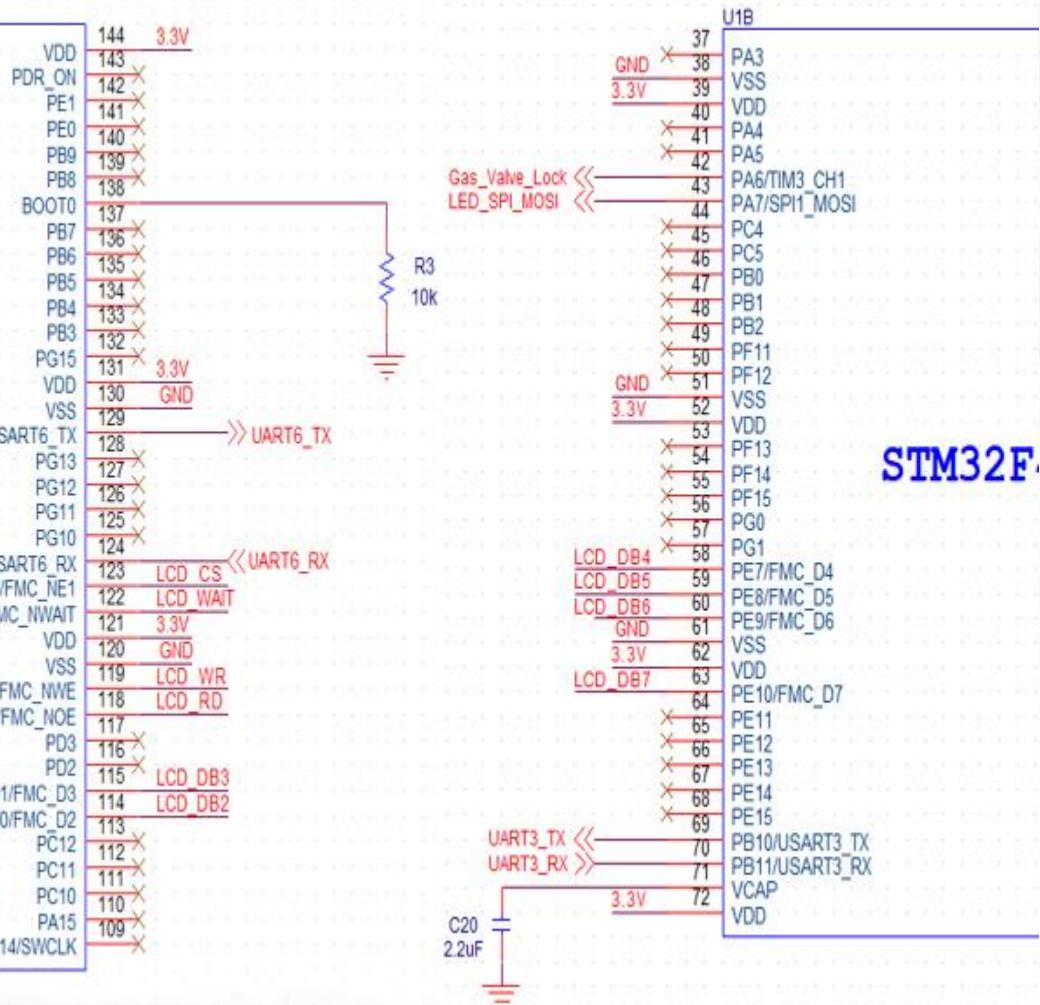
PC26

4 회로도 (CO_Sensor)

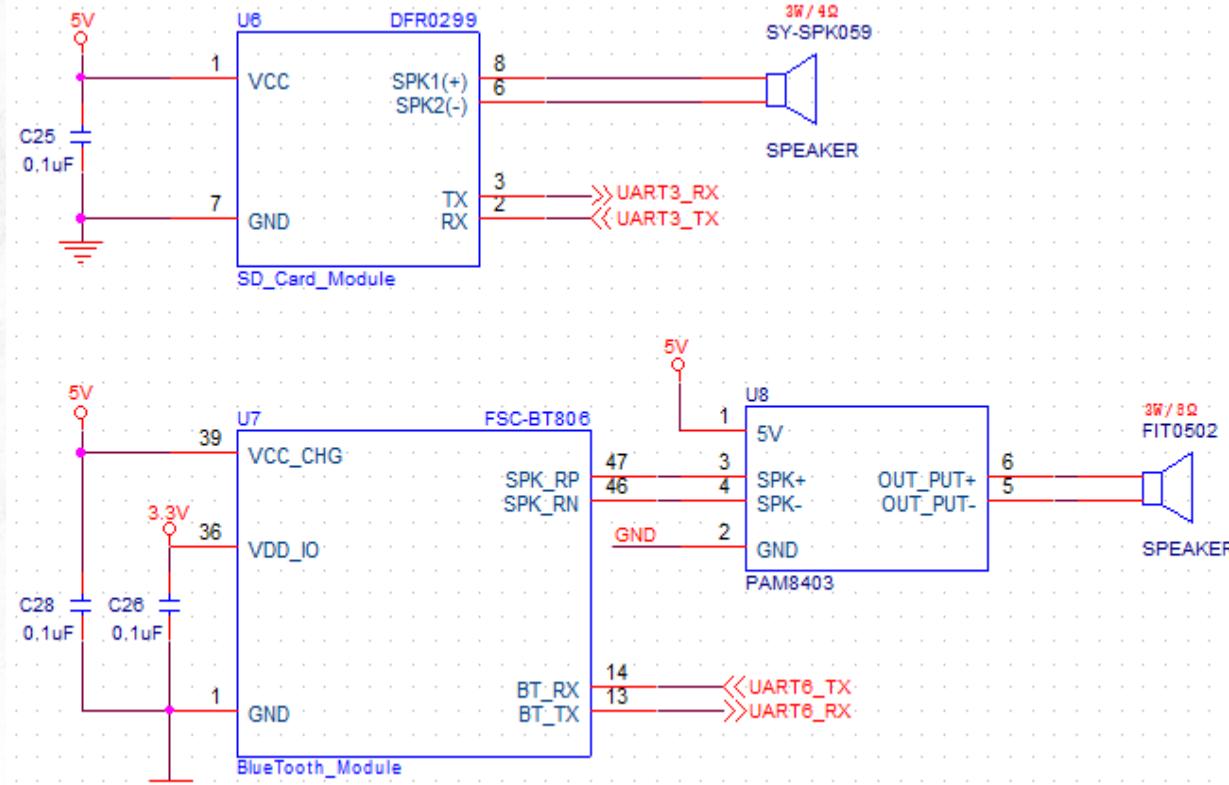


회로도 (SD_Card_Module, Bluetooth Speaker)

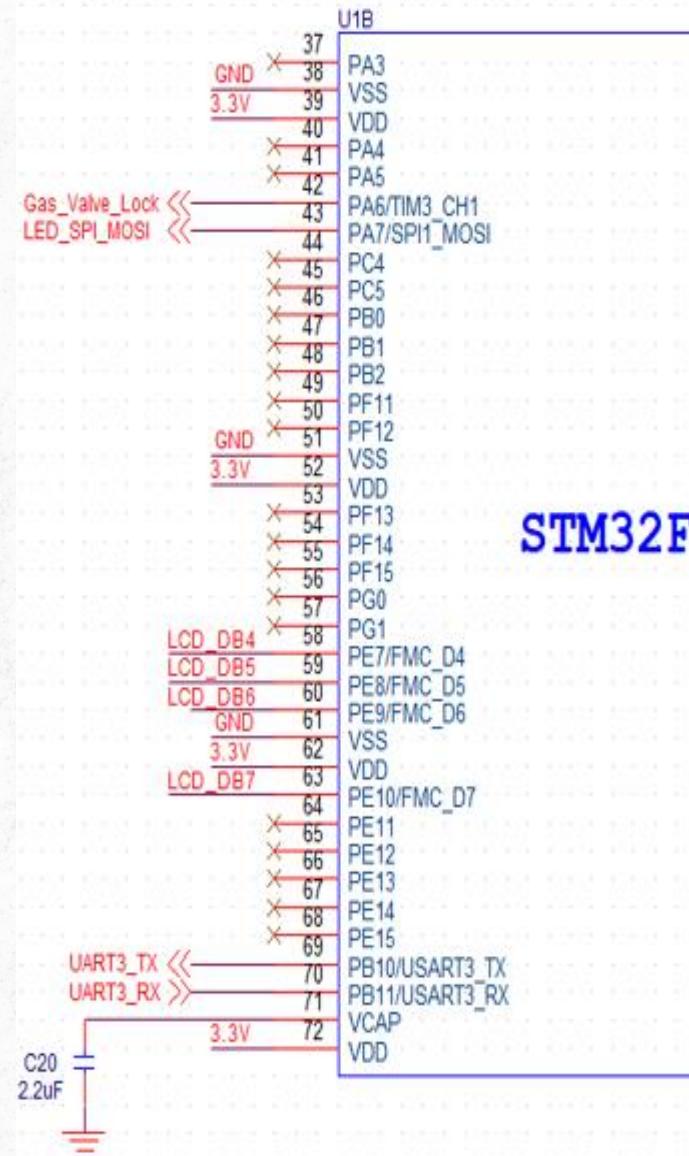
F429ZIT6



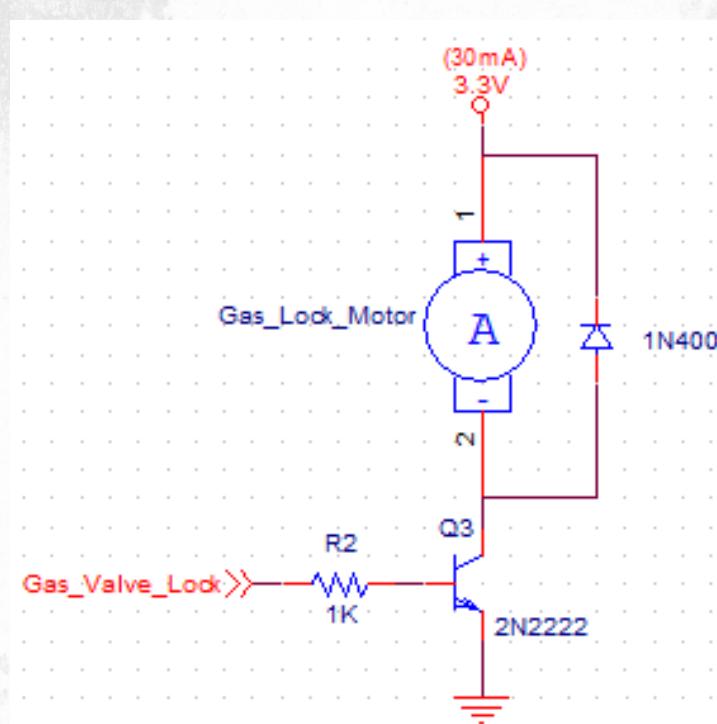
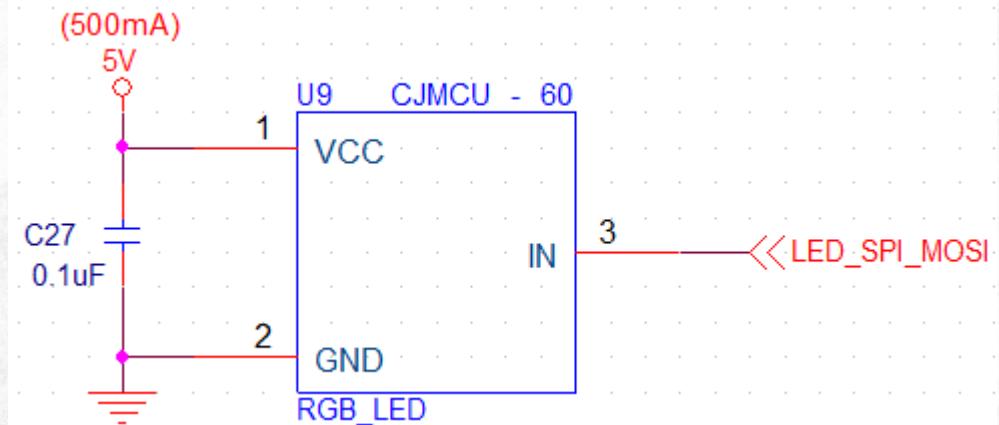
STM32F



회로도 (RGB_LED, Gas Valve)



STM32F



FirmWare 설명(IAR) (CO 센서)

```

HAL_ADC_Start(&hadc1); // ADC 변화 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));

```

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

(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

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

FirmWare 설명(IAR) (통신)

```

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

    // 블루투스 아ドレス 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); // 깨끗 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
}

```

FirmWare 설명(IAR) (통신)

```

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_RX_re_1 = 0; reset_RX_re_1 <=249; reset_RX_re_1++)
            {
                RE_re_1[reset_RX_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_RX_re_1 = 0; reset_RX_re_1 <=249; reset_RX_re_1++)
            {
                RE_re_1[reset_RX_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_RX_re_1 = 0; reset_RX_re_1 <=249; reset_RX_re_1++)
            {
                RE_re_1[reset_RX_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 int로 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_rx_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';
        }
    }
}

```

FirmWare 설명(Androide_Studio)

```

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")
}

```

실험 결과(IOT 시스템)(1/5)

◆ 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)

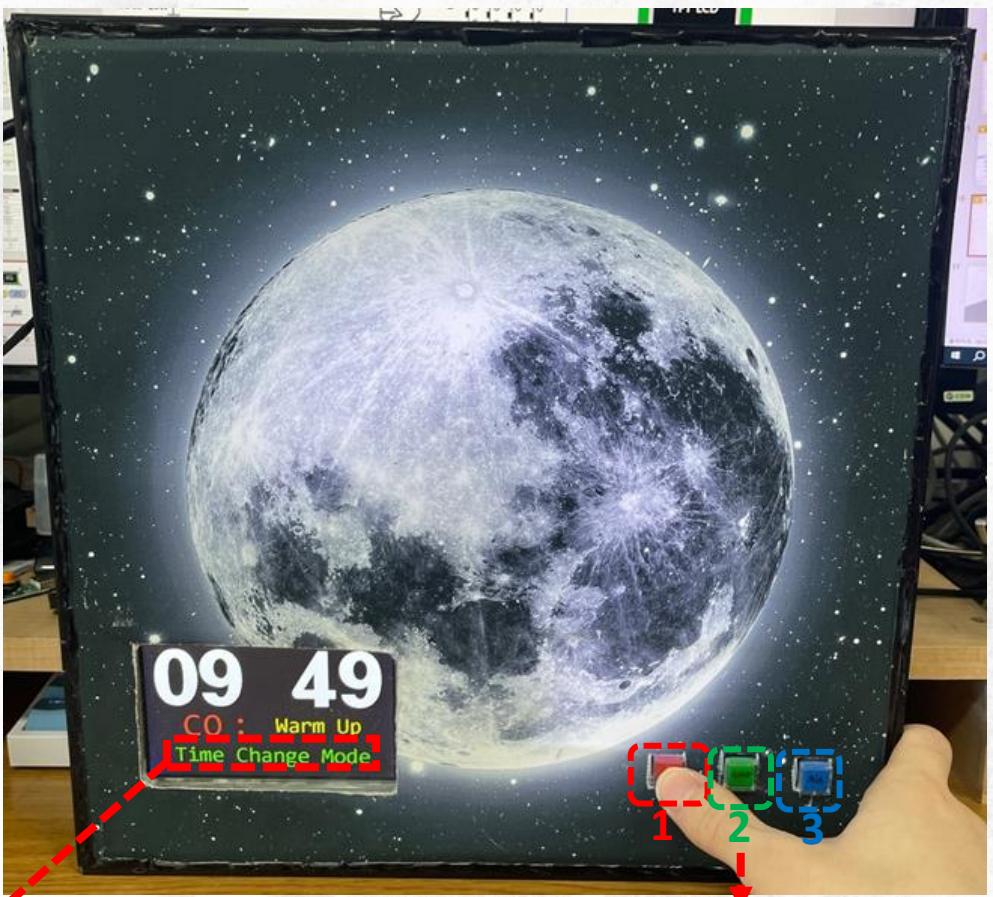
◆ CO 농도 30 ppm 미만 (평상시)



※ 스위치 기능

SW1. LED 색상 조절 가능 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

실험 결과(IOT 시스템)(3/5)

◆ CO 농도 30 ppm 미만 (평상시)



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



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

실험 결과(IOT 시스템)(4/5)

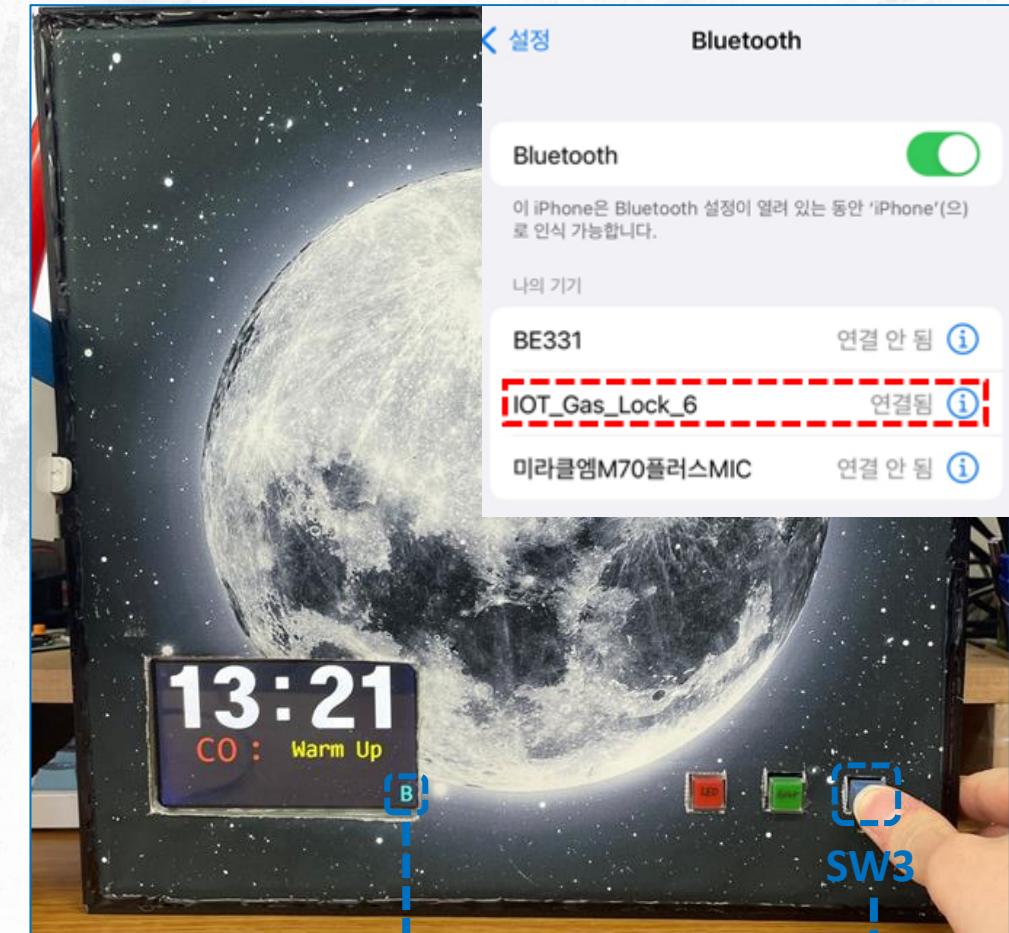
◆ CO 농도 30 ppm 미만 (평상시)



5분후
→



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



※ 스위치 기능
SW3 : 3초간 누르면 **블루투스** 연결(음악 재생 가능)

실험 결과(IOT 시스템)(5/5)

◆ CO 농도 30 ppm 이상 (가스경보시)

블루투스 연결이 해제되고, 스피커를 통해 가스 경보음을 재생한다.



가스 경보 시
가스 밸브 자동 잠금

LED 색상이 적색으로 출력

LCD에 가스 누출 감지 화면 및 CO 농도 출력

※ 스위치 기능
SW1. 가스 경보음 수동 정지 버튼

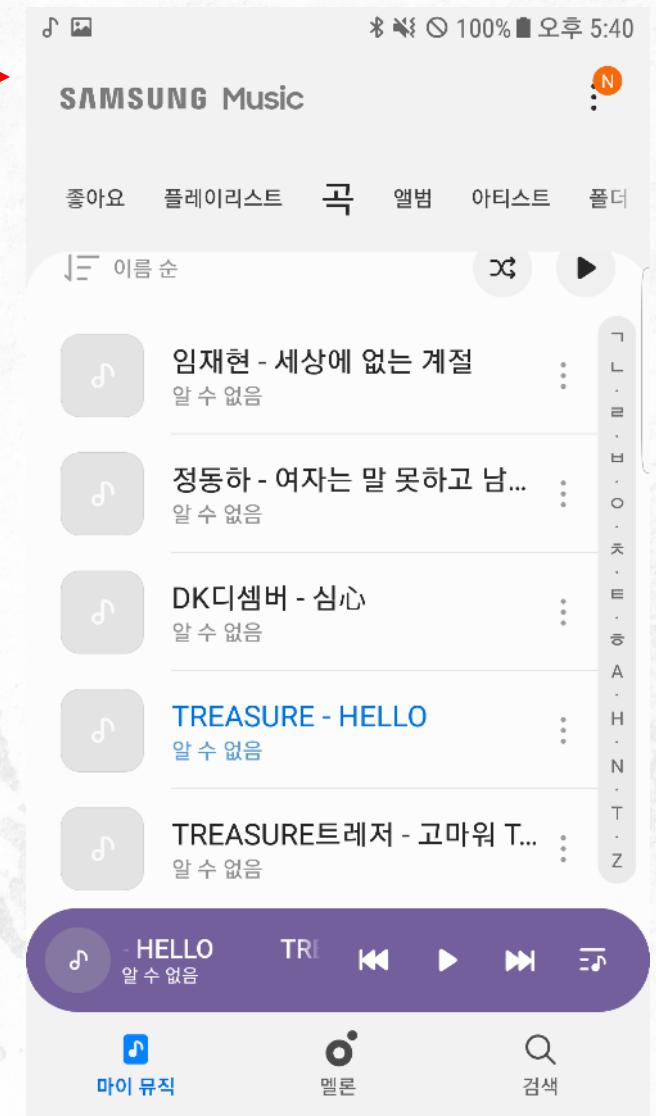
실험 결과(애플리케이션)(1/4)



APP 아이콘

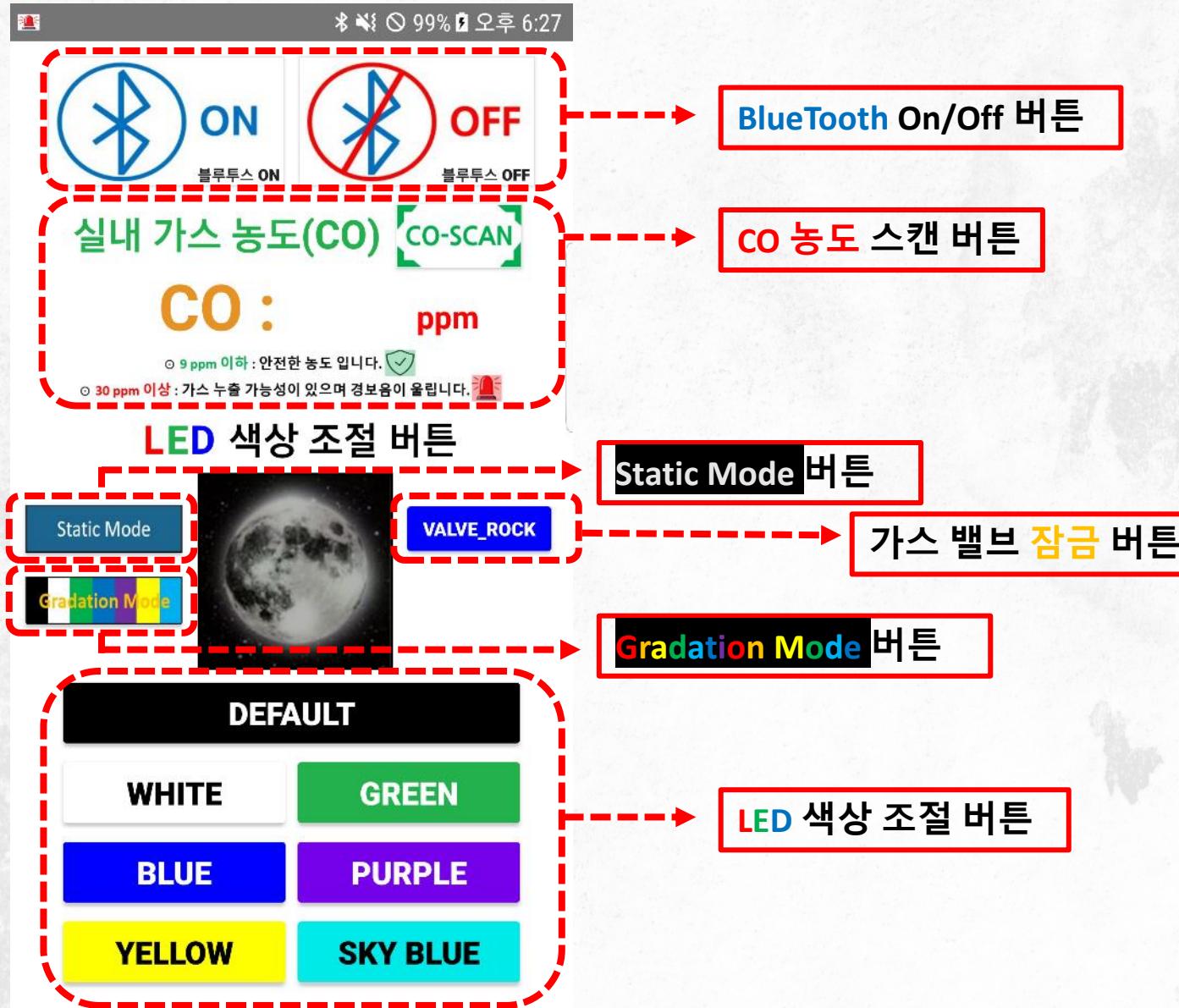


시작 버튼을 누르면 다음 화면으로 이동

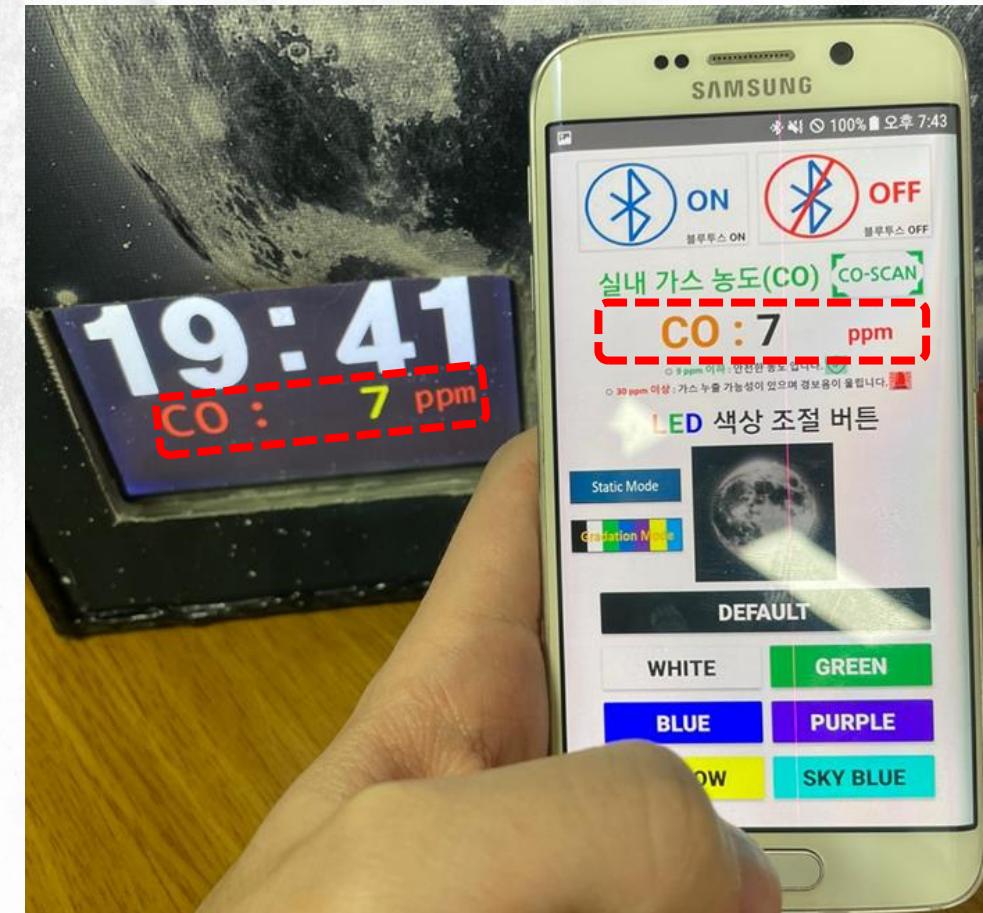


뮤직 버튼을 누르면 스마트폰의 뮤직으로 이동

실험 결과(애플리케이션)(2/4)



◆ CO 농도 30 ppm 미만 (평상시)



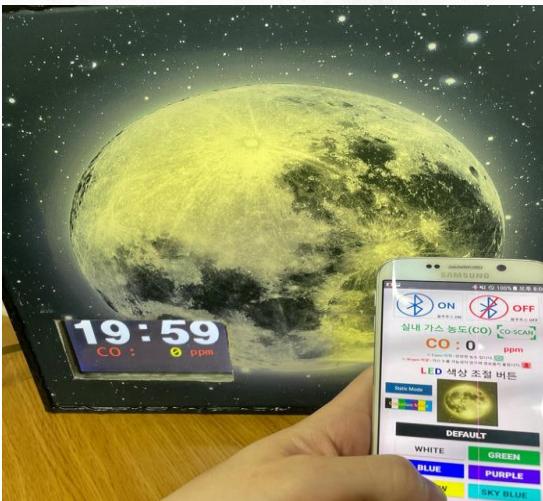
실시간으로 CO 농도 모니터링

실험 결과(애플리케이션)(3/4)

◆ CO 농도 30 ppm 미만 (평상시)

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

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



실험 결과(애플리케이션)(4/4)

◆ CO 농도 30 ppm 이상 (가스경보시)



스마트폰 진동과 함께 가스 누출 경보 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 사용 가능 할 것임

감사합니다!