



신한대학교
전자공학과

라이다를 활용한 자율 주행 배달 로봇

캡스톤 디자인(1) 5조

20200840 정성빈, 20221713 장형중, 20221714 최정우

목차

01

목적 및 기대효과

1. 자율주행 배달로봇의 목적
2. 상용화 되었을때의 기대효과

02

블럭도, 흐름도

1. 자율주행 배달로봇 하드웨어 블럭도
2. 자율주행 배달로봇 흐름도

03

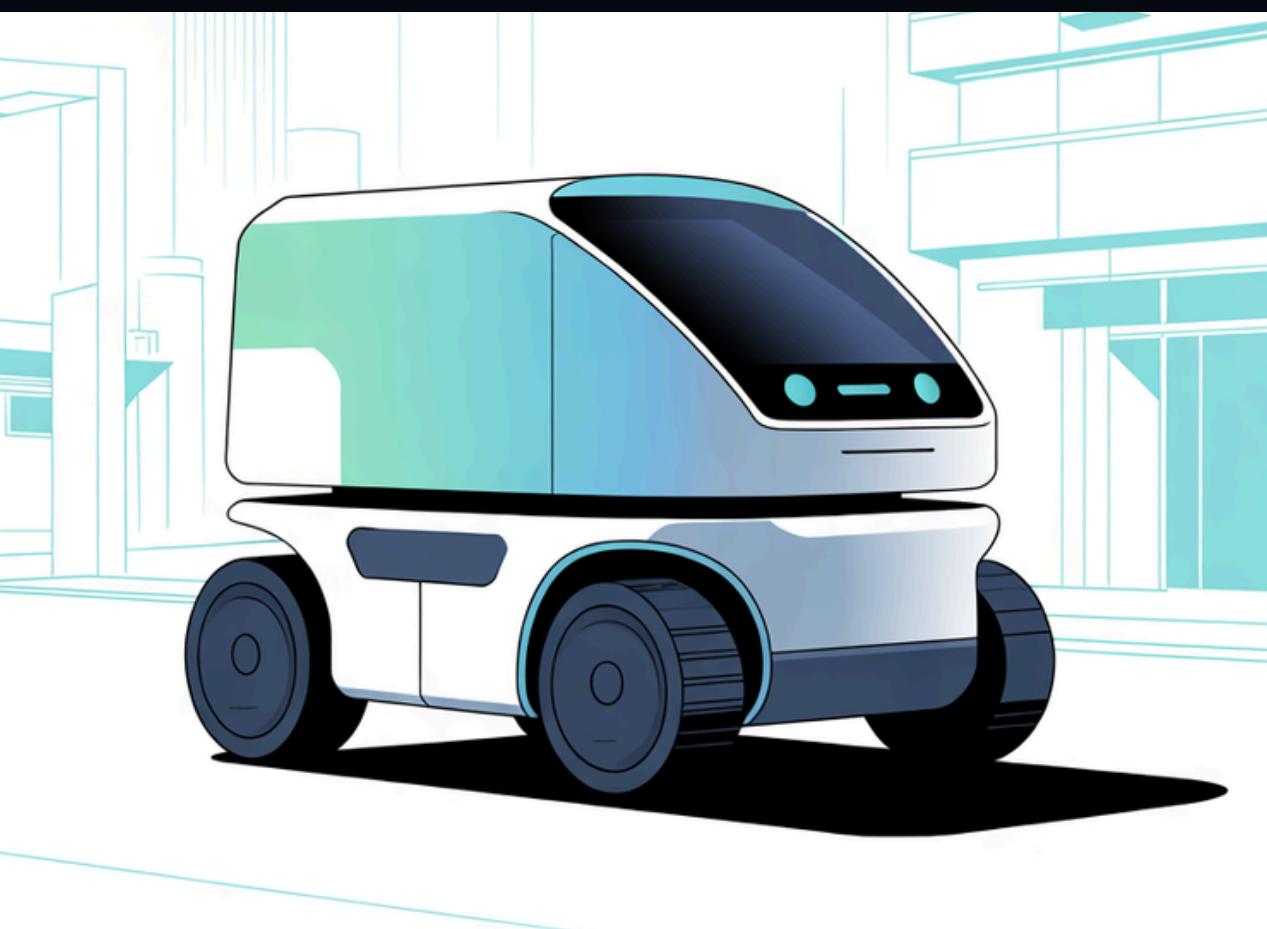
기술, 부품, 회로설명

1. 장형중(ubuntu, ros, gimp)
2. 최정우(motor, 블루투스, 스위치)
3. 정성빈(can, tof, 스피커, pcb)

04

통합결과

1. 최종 로봇 사진
2. 구현 영상(이찬주 교수님 연구실)



목적

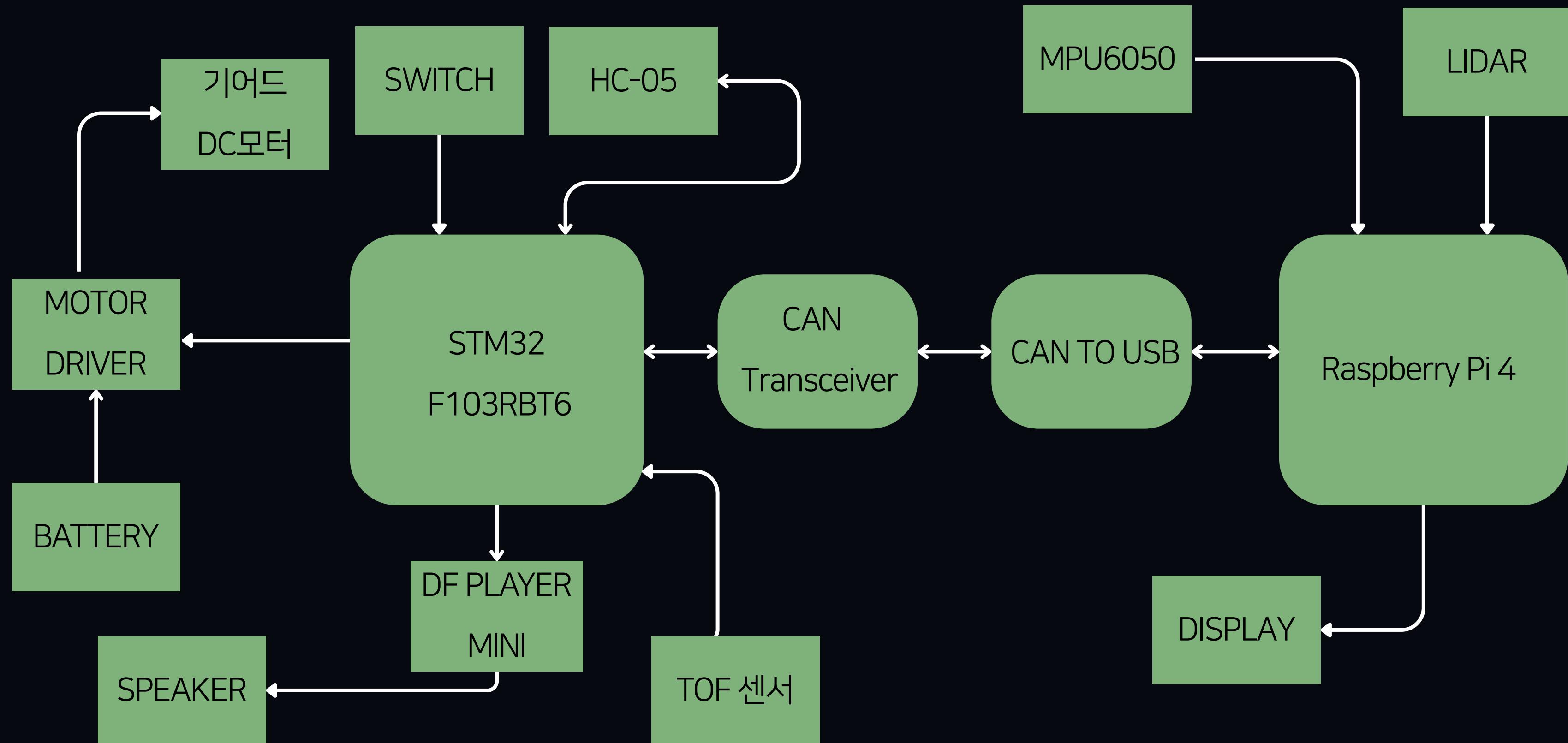
- 코로나 19이후 비대면 서비스에 대한 수요 증가
- 인건비 인상으로 인한 물류, 배달자동화 수요 증가
- 사람이 직접 수행하던 단순 운반 업무를 로봇이 대체함으로
업무 효율성을 높이고 인력 부담 완화

기대효과

- 비대면 서비스 확대에 따른 안전한 업무 환경 조성
- 인력 부담을 줄이고 인건비 상승 문제를 완화
- 신체적 제약이 있는 사용자에게 도움을 줄 수 있는 기술



하드웨어 블럭도



Section 3.

장형중

1. UBUNTU 20.04LTS
2. ROS 1 (HECTOR SLAM)
3. GIMP

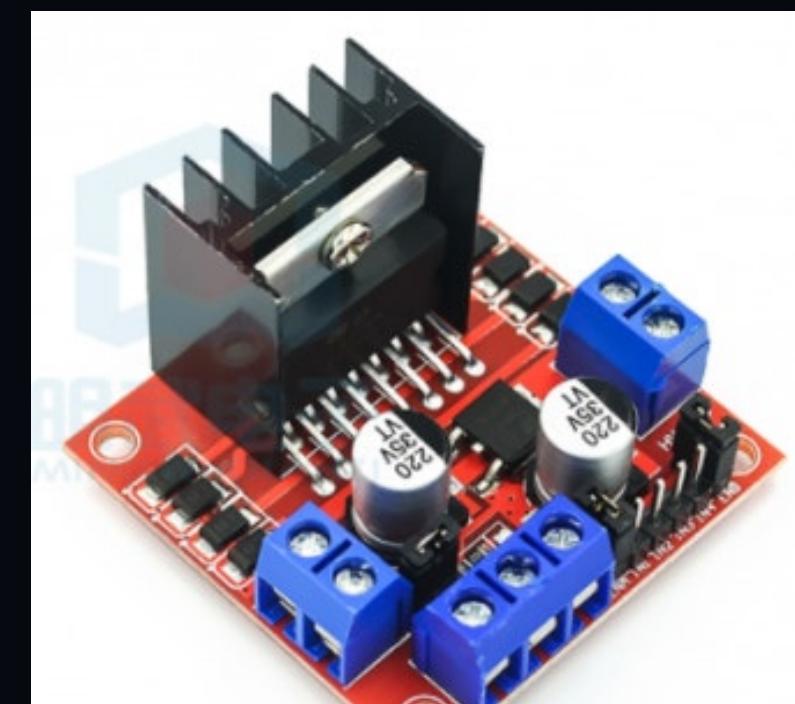


영상대체

Section 3.

최정우

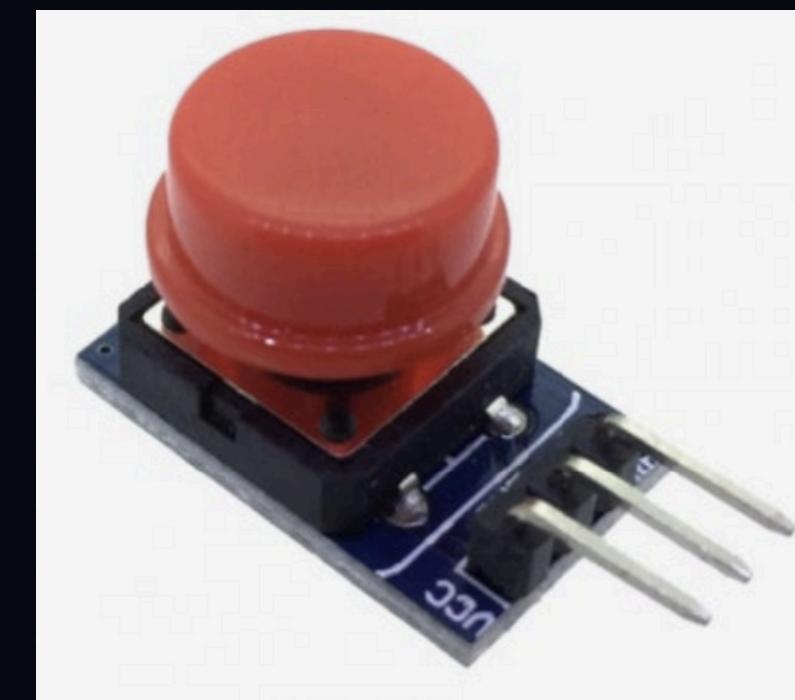
1. MOTOR



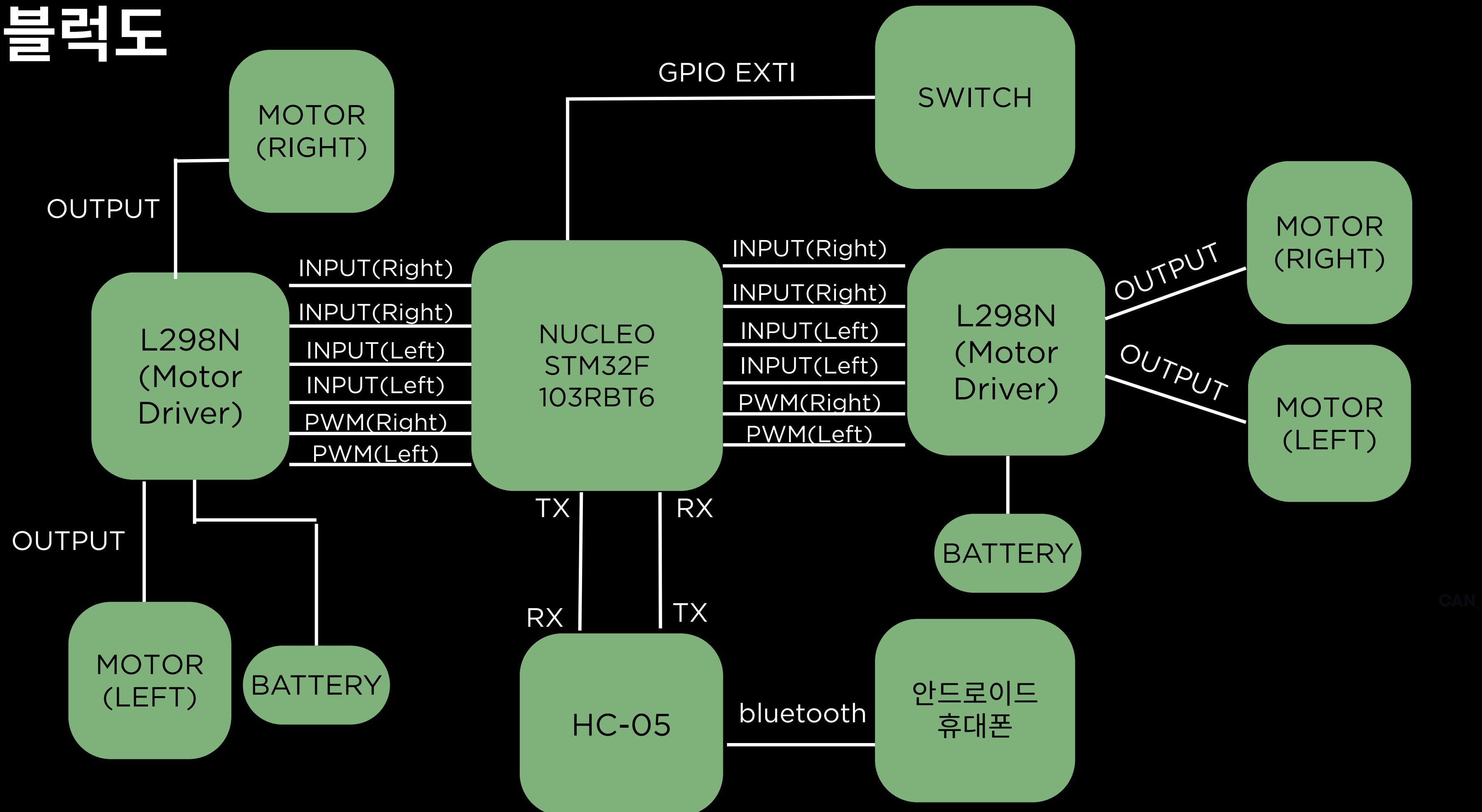
2. BLUETOOTH

3. SWITCH

STM32 센서 융합

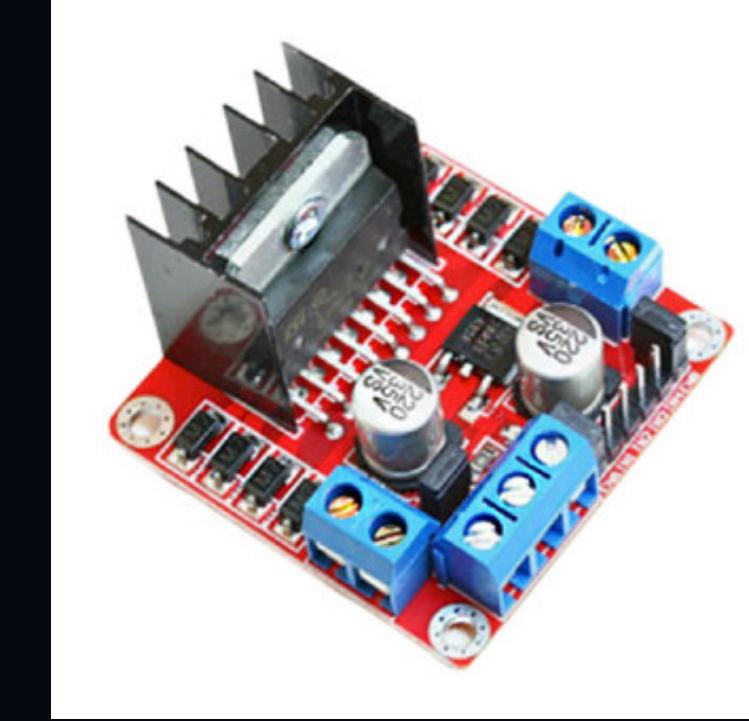


블럭도



1. MOTOR

TIMERS	TIM2 사용
사용 pin	5V, PC0~4(INPUT), PA0(CH1),PA1(CH2), PC5~8(INPUT), PA8(CH1),PA9(CH2)
외부전원	11.1V 배터리 사용
모터	JGA25-370(130RPM)
모터드라이버	L298N



L298N



JGA25-370

1. MOTOR

```
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_2);

HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, 1);
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, 0);
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_2, 1);
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_3, 0);

__HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_1, right_pwm); // 오른쪽
__HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_2, left_pwm); // 왼쪽

__HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_1, 0); // 오른쪽
__HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_2, 0);
```

모터 전진 PC0 = HIGH, PC1 = LOW, PC2 = HIGH, PC3 = LOW

모터 후진 PC0 = LOW, PC1 = HIGH, PC2 = LOW, PC3 = HIGH

앞쪽 모터는 PWM = 5000고정값(속도 유지 위해 사용)
뒷쪽 모터는 방향 조절을 위해 TOF의 따른 모터 속도 변경

CH1, CH2 PWM값은 0 ~ 10000사이에서 조절

2. HC-05(blueooth)

통신방식	USART3 사용
사용 pin	3.3V, GND, PC10 (TX), PC11(RX)
통신속도	9600Bits
범위	0 ~ 10M까지 가능
데이터 처리 특징	비동기(직렬 통신 방식)



HC-05

Android Studio

블루투스를 연결 및 해제하는
버튼 설계

블루투스 연결 버튼 ->

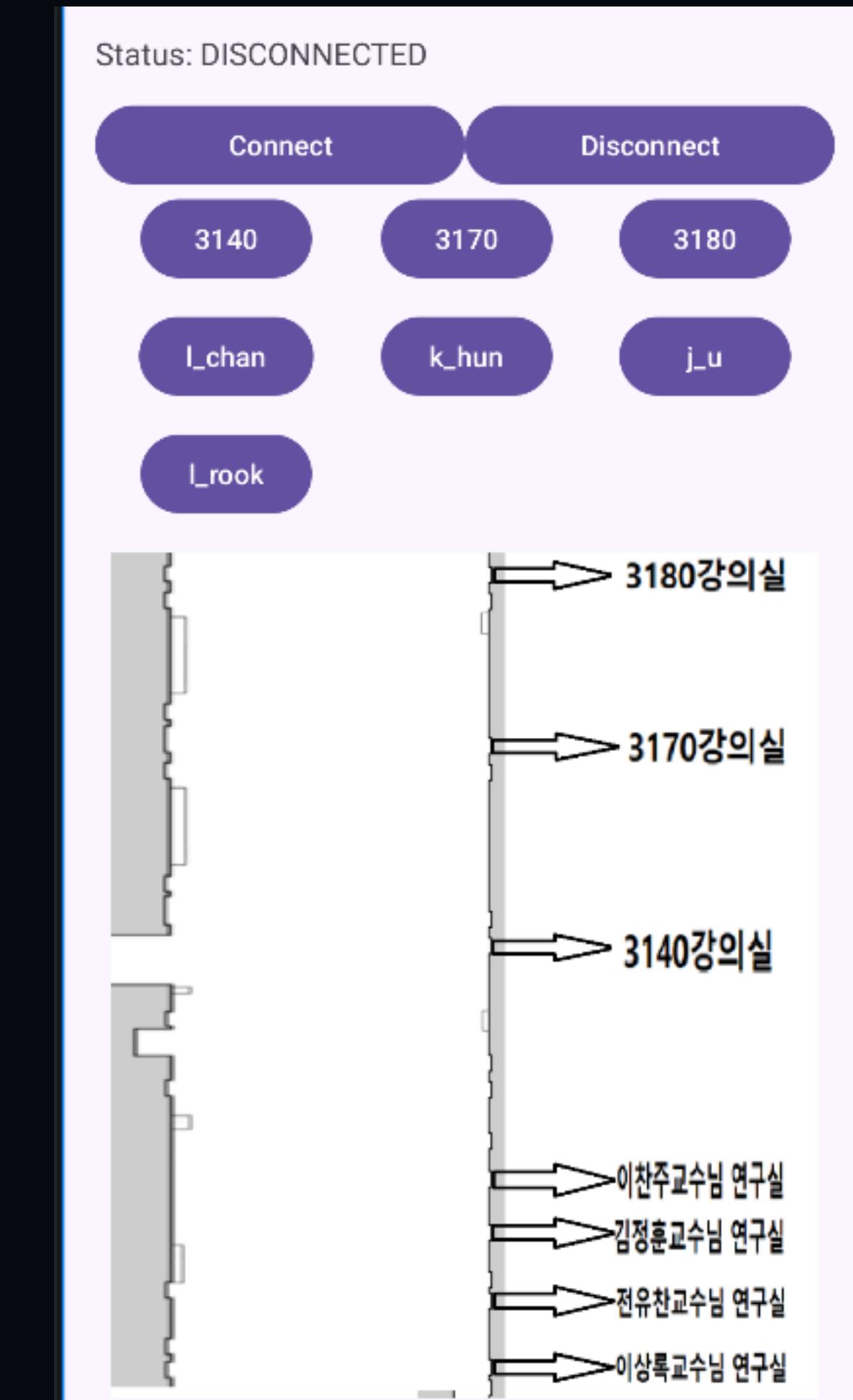
사용자가 원하는 목표 위치를
선택 할 수 있는 버튼 설계

강의실 및
교수님 연구실 버튼 ->

시각적 자료를 위한 지도 생성

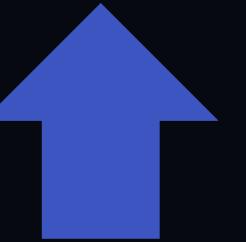
3층 복도 지도 ->

※Android Studio 환경에서 앱 구축



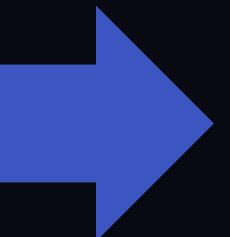
2. BLUETOOTH

```
// Bluetooth 수신 인터럽트 시작  
HAL_UART_Receive_IT(&huart3, &bt_rx_data, 1);
```



Bluetooth Start Code

1. 1 ~ 8 중 선택
2. switch문으로 넘어가서 해당값 case로 들어감
3. 선택된 case의 C파일로 이동하여 동작
4. 이후 0으로 초기화

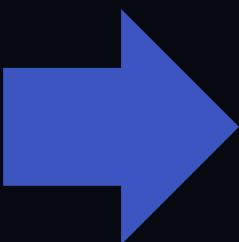


```
// Bluetooth 데이터 수신 처리 (명령 실행)  
if (bt_data_received) {  
  
    int cmd = atoi((char*) bt_rx_buffer);  
  
    switch (cmd) {  
        case 1:  
            l_chan(); break; //이찬주 교수님 연구실  
        case 2:  
            k_hun(); break; //김정훈 교수님 연구실  
        case 3:  
            j_yoo(); break; //전유찬 교수님 연구실  
        case 4:  
            l_rok(); break; //이상록 교수님 연구실  
        case 5:  
            class_3140f(); break; //3140강의실  
        case 6:  
            class_3170f(); break; //3170강의실  
        case 7:  
            class_3180f(); break; //3180강의실  
        case 8:  
            motor_stop(); break; //모터정지  
        default:  
            motor_stop(); break;  
    }  
  
    bt_data_received = 0;  
}
```

2. USART

UART 설정

통신속도: 9600BIT
Word Length: 8BIT
Parity: NONE
Stop Bits: 1



Mode Asynchronous

Hardware Flow Control (RS232) Disable

Configuration

Reset Configuration

DMA Settings GPIO Settings

User Constants NVIC Settings

Parameter Settings

Configure the below parameters :

Search (Ctrl+F)

Basic Parameters

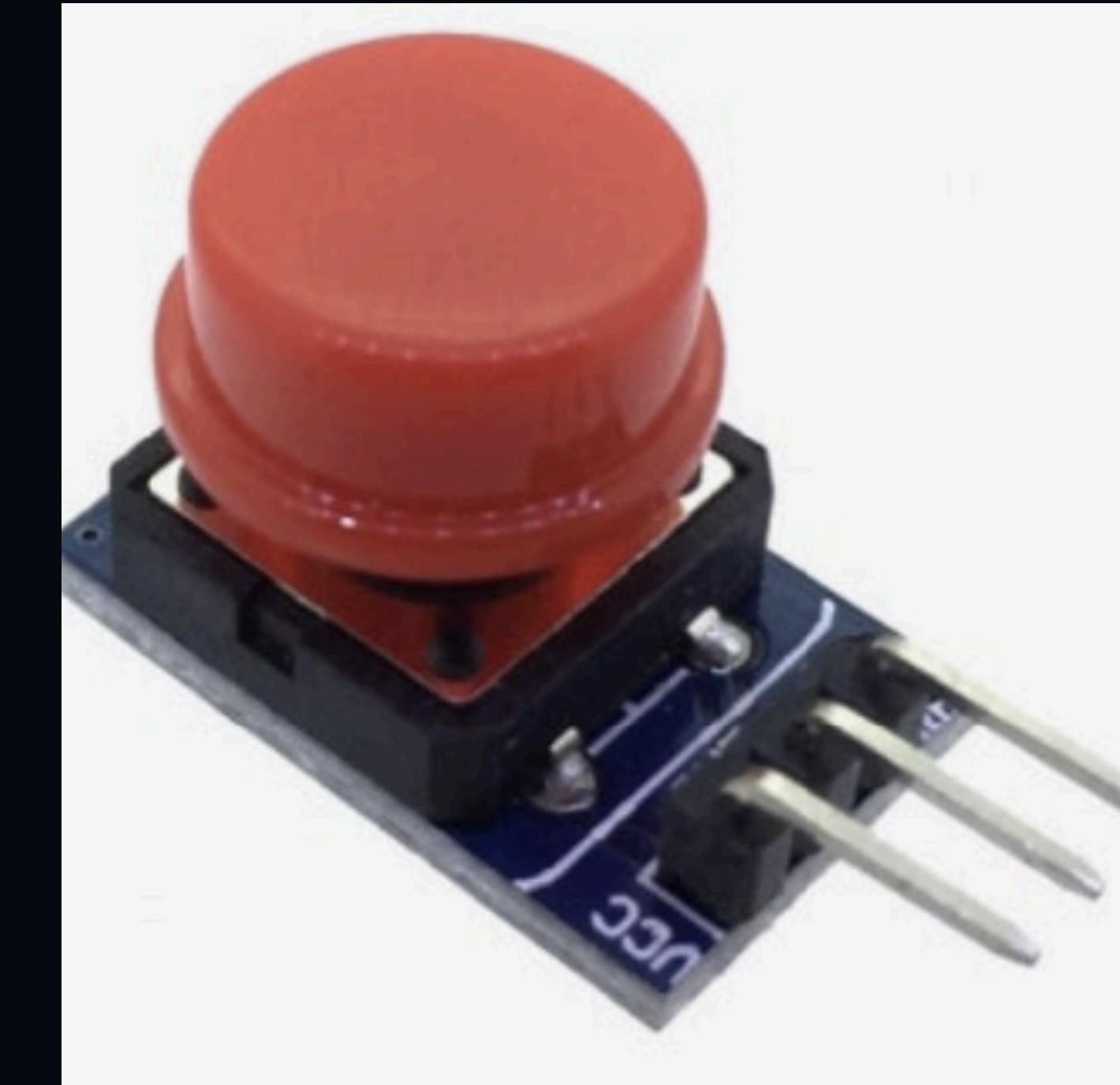
Baud Rate	9600 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1

Advanced Parameters

Data Direction	Receive and Transmit
Over Sampling	16 Samples

3. SWITCH

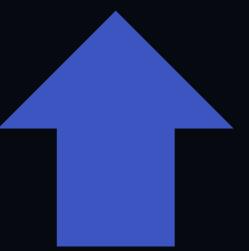
사용 방식	GPIO EXTI
사용 pin	PC13(GPIO), GND
GPIO Mode	INPUT Mode
Pull 설정	Pull - up
스위치 상태	기본상태 → HIGH(1) 눌린상태 → LOW(0)



SWITCH

3. SWITCH

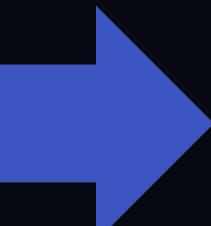
```
static inline int Button_PC13_Pressed(void)
{
    // 풀업 구성: 평상시 HIGH, 눌리면 LOW
    return (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET); //RESET
}
```



pull-up 설정으로 인한 기본상태는 HIGH ->
INT 발생시 GND로 연결되어 LOW로 떨어짐

CUBE IDE 환경설정에서 GPIO Port 설정

1. GPIO Mode를 Input으로 설정
2. pull-up 설정



GPIO Mode and Configuration

Configuration

Group By Peripherals

RCC	SYS	TIM	USART				
GPIO	CAN	I2C					
PC3	n/a	Low	Outp...	No p...	Medi...	BIN2	<input checked="" type="checkbox"/>
PC5	n/a	Low	Outp...	No p...	Medi...	AIN1...	<input checked="" type="checkbox"/>
PC6	n/a	Low	Outp...	No p...	Medi...	AIN2...	<input checked="" type="checkbox"/>
PC8	n/a	Low	Outp...	No p...	Medi...	BIN1...	<input checked="" type="checkbox"/>
PC9	n/a	Low	Outp...	No p...	Medi...	BIN2...	<input checked="" type="checkbox"/>
PC1...	n/a	n/a	Input ...	Pull-up	n/a		<input checked="" type="checkbox"/>

Search Signals Search (Ctrl+F) Show only Modified Pins

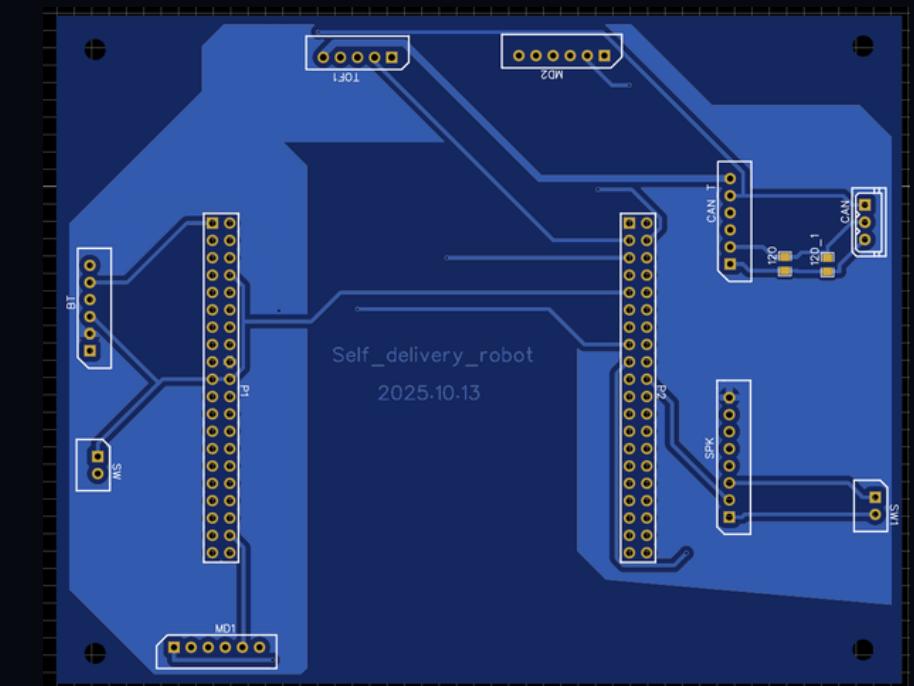
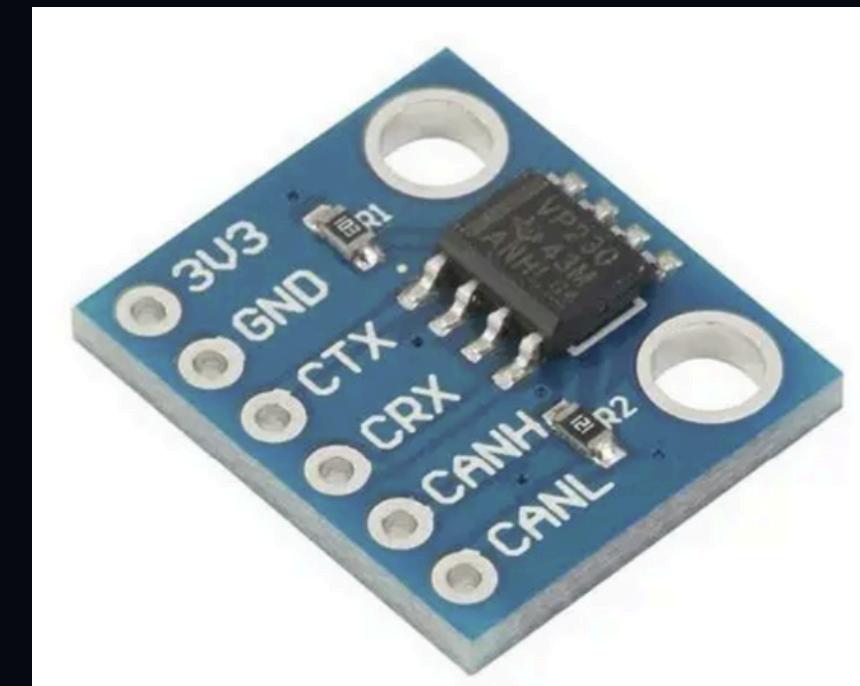
PC13-TAMPER-RTC Configuration :

GPIO mode	<input type="button" value="Input mode"/>
GPIO Pull-up/Pull-down	<input type="button" value="Pull-up"/>
User Label	<input type="text"/>

Section 3.

정성빈

1. CAN

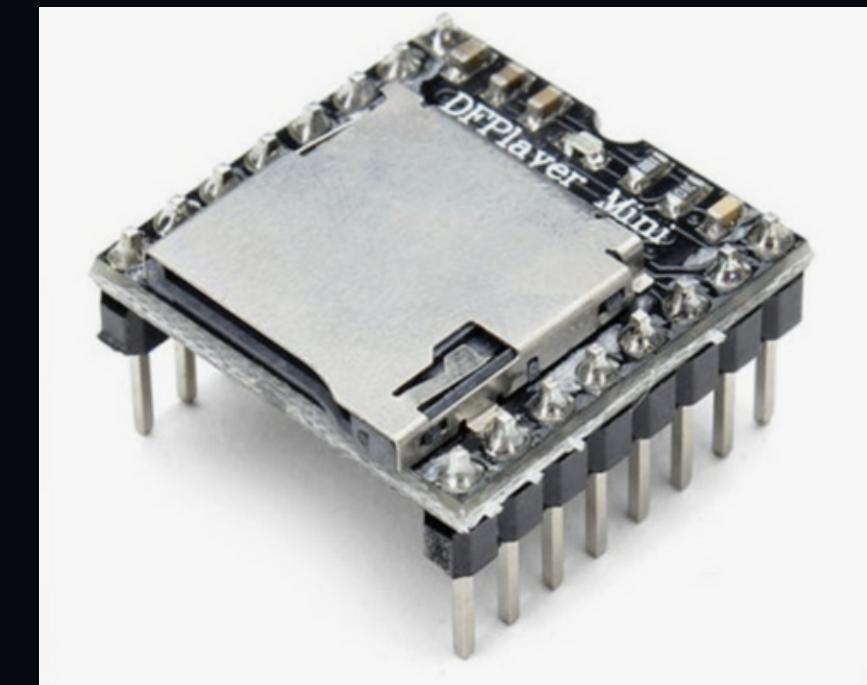


2. ToF

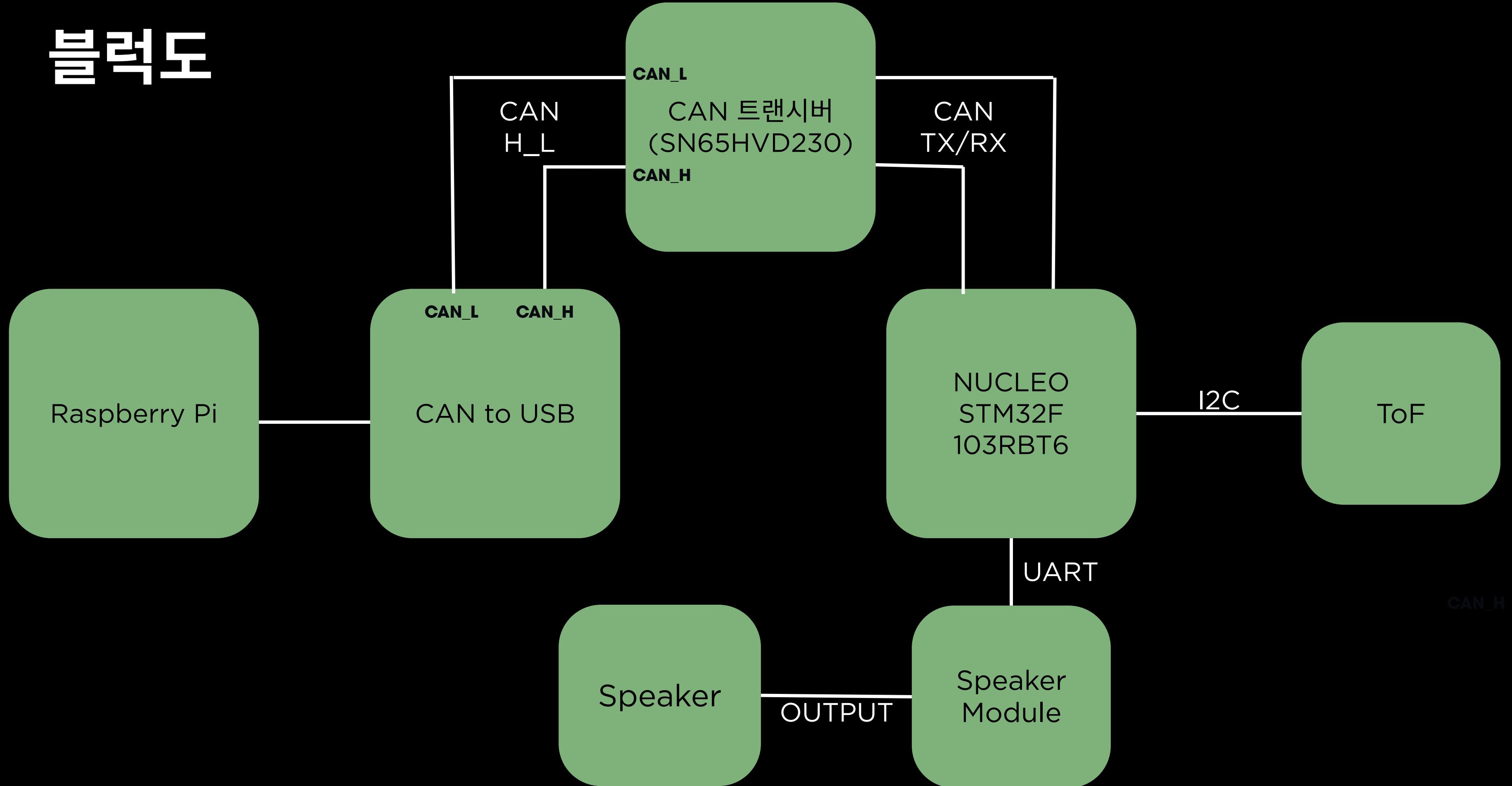
3. Speaker

4. PCB

통신 & STM 센서 제어

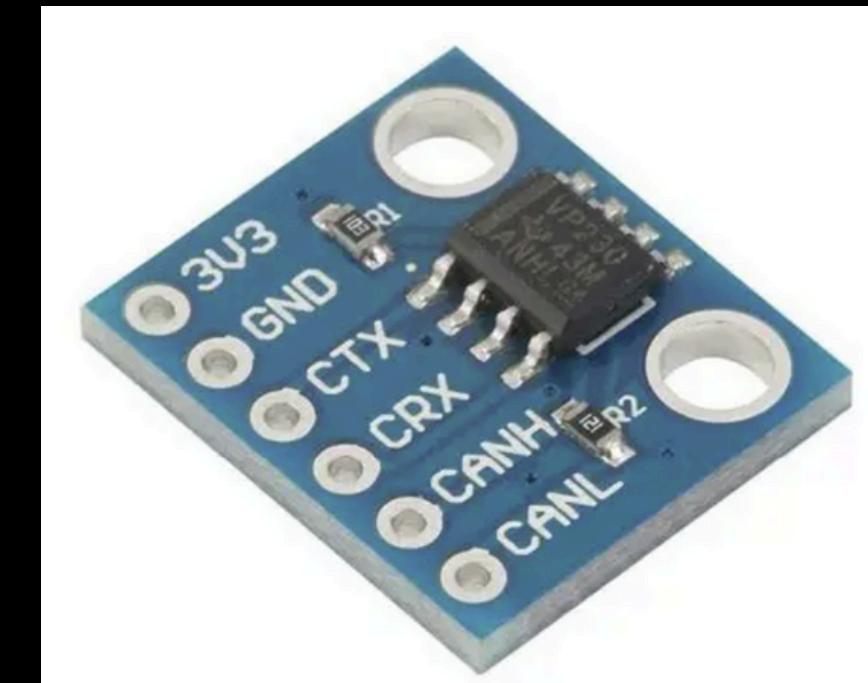


블럭도



1. CAN

CAN 트랜시버	SN65HVD230
CAN to USB	라즈베리파이 - 트랜시버 연결
전송 속도	500kbps
종단 저항	120Ω 2개 사용
사용 전압	3V



SN65HVD230



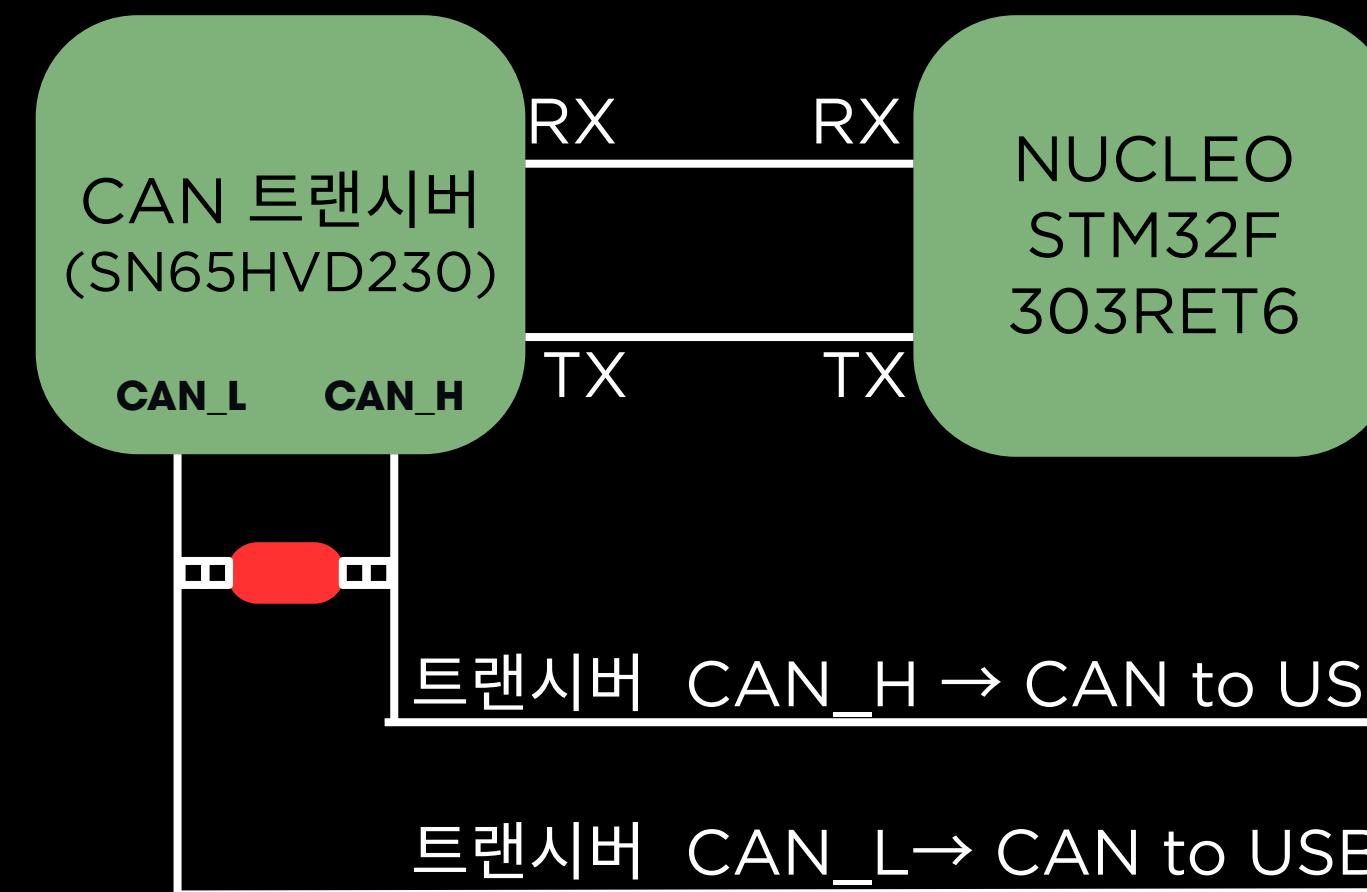
CAN to USB 분석기

1. CAN

RX-RX, TX-TX 연결 이유

CAN 통신은 양방향으로 데이터를 송수신하기 때문에,
트랜시버는 MCU와의 양방향 신호 전달을 위해
 $TX \leftrightarrow TX, RX \leftrightarrow RX$ 를 연결해야 함

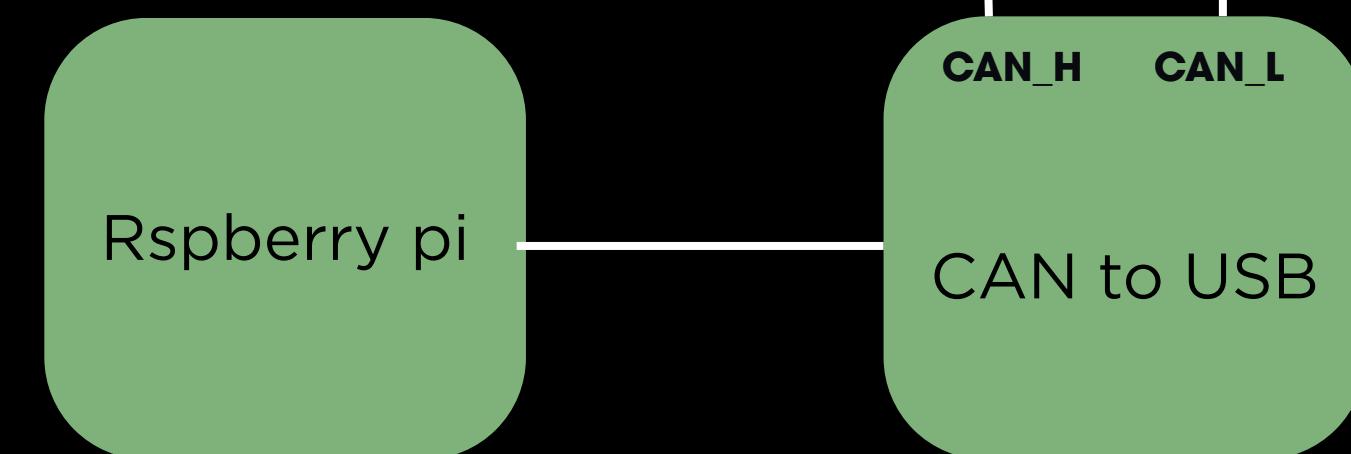
MCU - 트랜시버 전달을 위해 $TX-TX$ 연결
트랜시버 - MCU로 전달을 위해 $RX-RX$ 연결



120Ω 종단 저항

전송선로의 임피던스에 맞게 각 끝단에 120Ω 저항을 달면,
신호가 끝에서 흡수되어 반사되지 않아
신호 품질이 좋아지고 통신 에러가 줄어듦

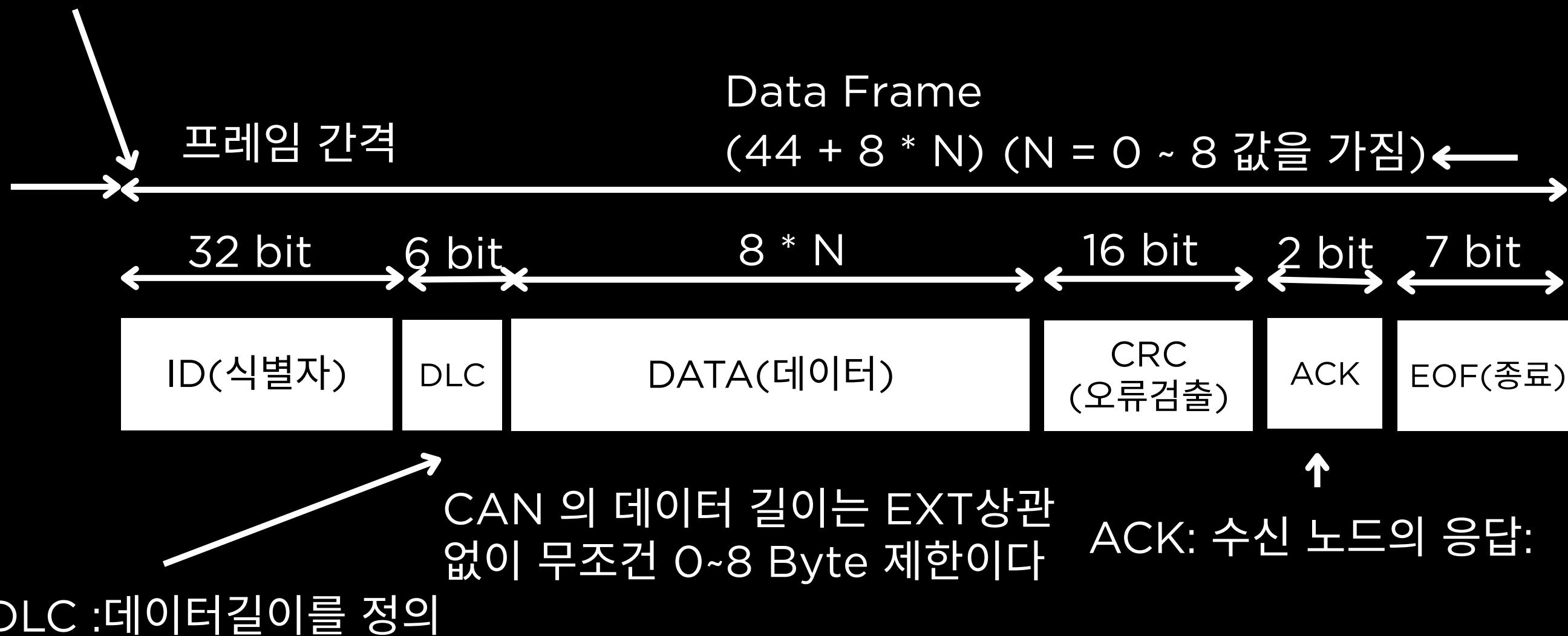
저항 1개 : 일부 반사 발생, 불안정
저항 3개 : 저항값 낮아져서 신호 감쇄 심함



1. CAN 통신의 원리

- 11 bit Standard ID의 경우 - Data Frame

SOF (Start of Frame): 전송 시작을 알림



1. CAN

```
void CAN_filter(void){  
CAN_FilterTypeDef sFilterConfig;  
    sFilterConfig.FilterBank = 0;  
    sFilterConfig.FilterMode = CAN_FILTERMODE_IDMASK;  
    sFilterConfig.FilterScale = CAN_FILTERSCALE_32BIT;  
    sFilterConfig.FilterIdHigh = 0x0000;  
    sFilterConfig.FilterIdLow = 0x0000;  
    sFilterConfig.FilterMaskIdHigh = 0x0000;  
    sFilterConfig.FilterMaskIdLow = 0x0000;  
    sFilterConfig.FilterFIFOAssignment = CAN_FILTER_FIFO0;  
    sFilterConfig.FilterActivation = ENABLE;  
    sFilterConfig.SlaveStartFilterBank = 14;  
  
    if (HAL_CAN_ConfigFilter(&hcan, &sFilterConfig) != HAL_OK) {  
        Error_Handler();  
    }  
}
```

- BANK : 0~13개의 공간 중 0번 저장
- Mode : IDMASK와 IDLIST 중 MASK는 범위를 지정해서 필터링하고 LIST는 정해진 ID만 필터링 한다.
- High와 Low에서 0x0000은 모든 값을 다 받는다.
- FIFO는 선입선출로 0번과 1번의 공간 중 0번에 저장한다.
- ENABLE : 활성화 코드

CAN Filter 설정 코드

2. ToF

측정거리	최대 약 2m
정밀도	%3 이내
시야각	25°
응답 속도	초당 최대 수십 회 측정
전원 전압	2.6~3.3V
인터페이스	I2C



ToF

2. ToF

```
VL53L1X_ERROR VL53L1X_GetDistance(uint16_t dev, uint16_t *distance)
{
    VL53L1X_ERROR status = 0;
    uint16_t tmp;

    status |= (VL53L1_RdWord(dev,
        VL53L1_RESULT_FINAL_CROSSTALK_CORRECTED_RANGE_MM_SDO, &tmp));
    *distance = tmp;
    return status;
}
```

ToF 거리 측정 함수

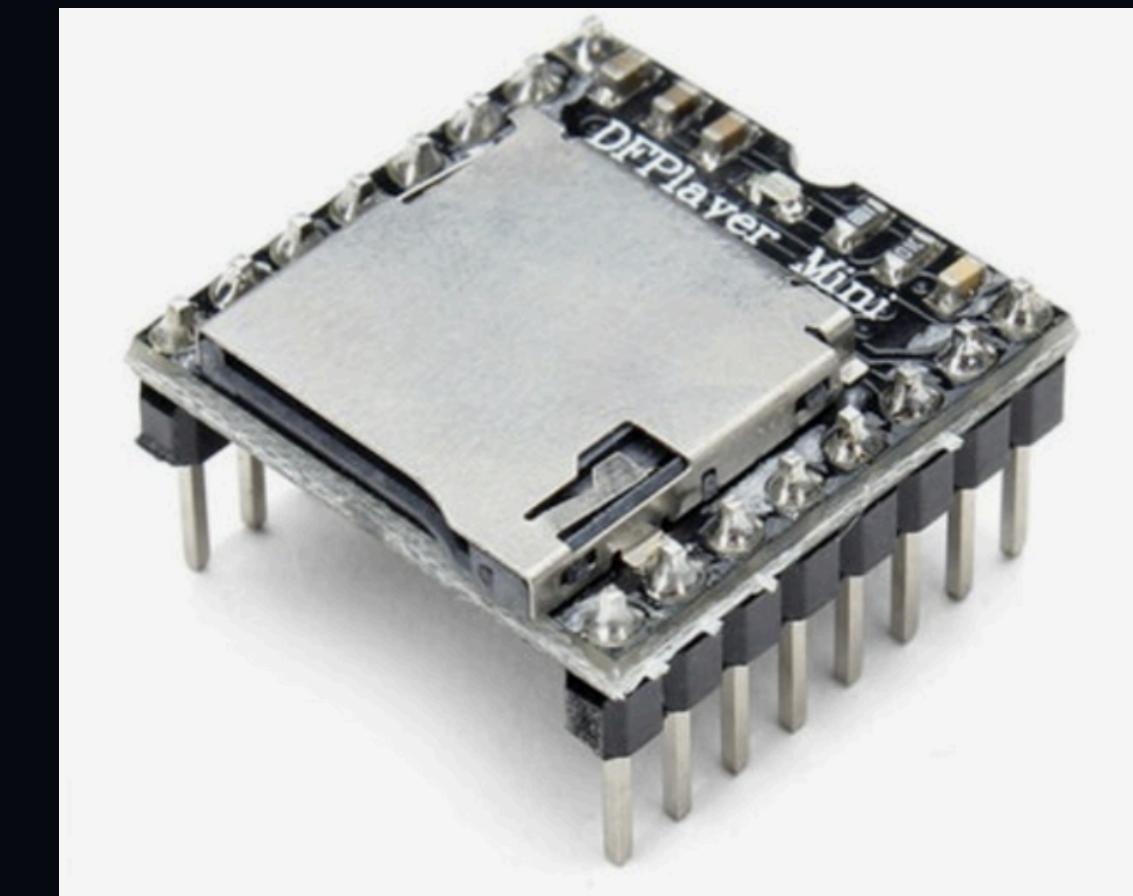
ToF 센서를 동작하기 위해 ST에서 제공하는 드라이버 코드 필요

- vl53l1_platform.c : 거리 측정, 모드 설정 등 ST 공식 API
- VL53L1X_api.c : I2C Read/Write, Delay, GPIO 등 MCU별 실제 하드웨어 제어

위의 코드에서 측정한 distance의 값으로 사용자 코드에서 활성화를 한 후 사용이 가능

3. Speaker

저장 매체	MicroSD 카드
제어 방식	UART, I/O 핀 제어
재생 모드	순차재생, 반복재생, 랜덤재생, 지정파일
통신 속도	기본 9600bps
전원 전압	3.2 ~ 5V
오디오 출력	3W 모노 출력 (스피커 직접 연결 가능)



Speaker

3. Speaker

```
static HAL_StatusTypeDef DF_SendFrame(uint8_t cmd, uint16_t param)
{
    uint8_t ph = (param >> 8) & 0xFF;
    uint8_t pl = (param & 0xFF);
    uint16_t sum = 0xFF + 0x06 + cmd + 0x00 + ph + pl;
    uint16_t cks = (uint16_t)(0x10000 - sum);
    uint8_t f[10] = {0x7E, 0xFF, 0x06, cmd, 0x00, ph, pl,
                    (uint8_t)(cks>>8), (uint8_t)cks, 0xEF};
    return HAL_UART_Transmit(&huart1, f, sizeof(f), 100);
}
```

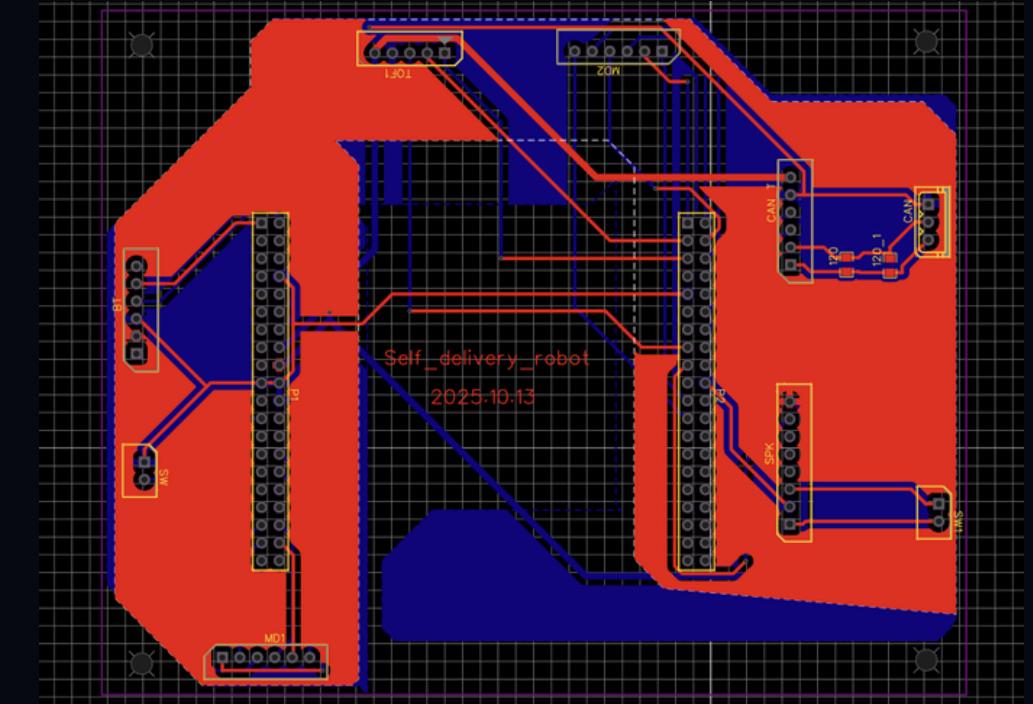
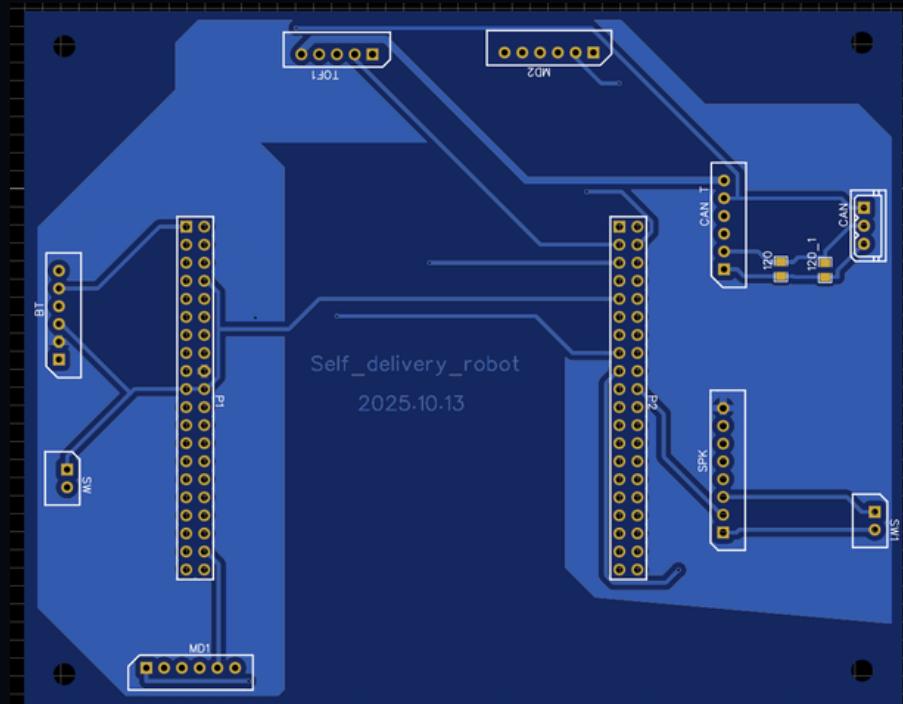
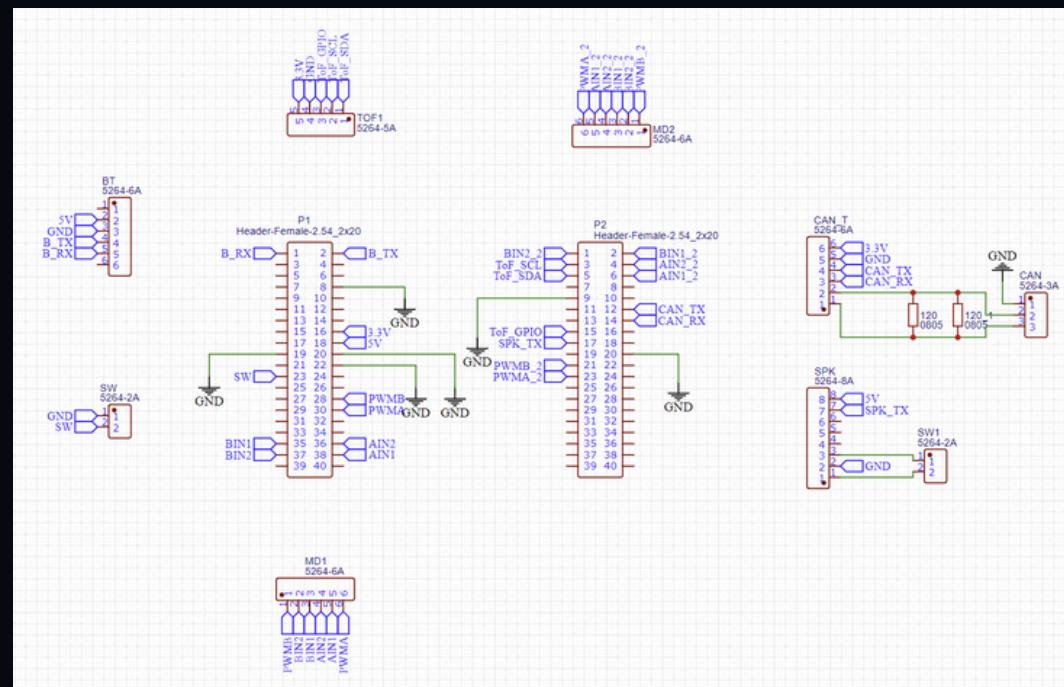
UART 전송 코드

```
static uint8_t spk_initied = 0;
if (!spk_initied) {
    Speaker_Init();
    spk_initied = 1;
}

Speaker_Play(7);
```

SD 카드에서 원하는 음성 선택 후 출력

4. PCB

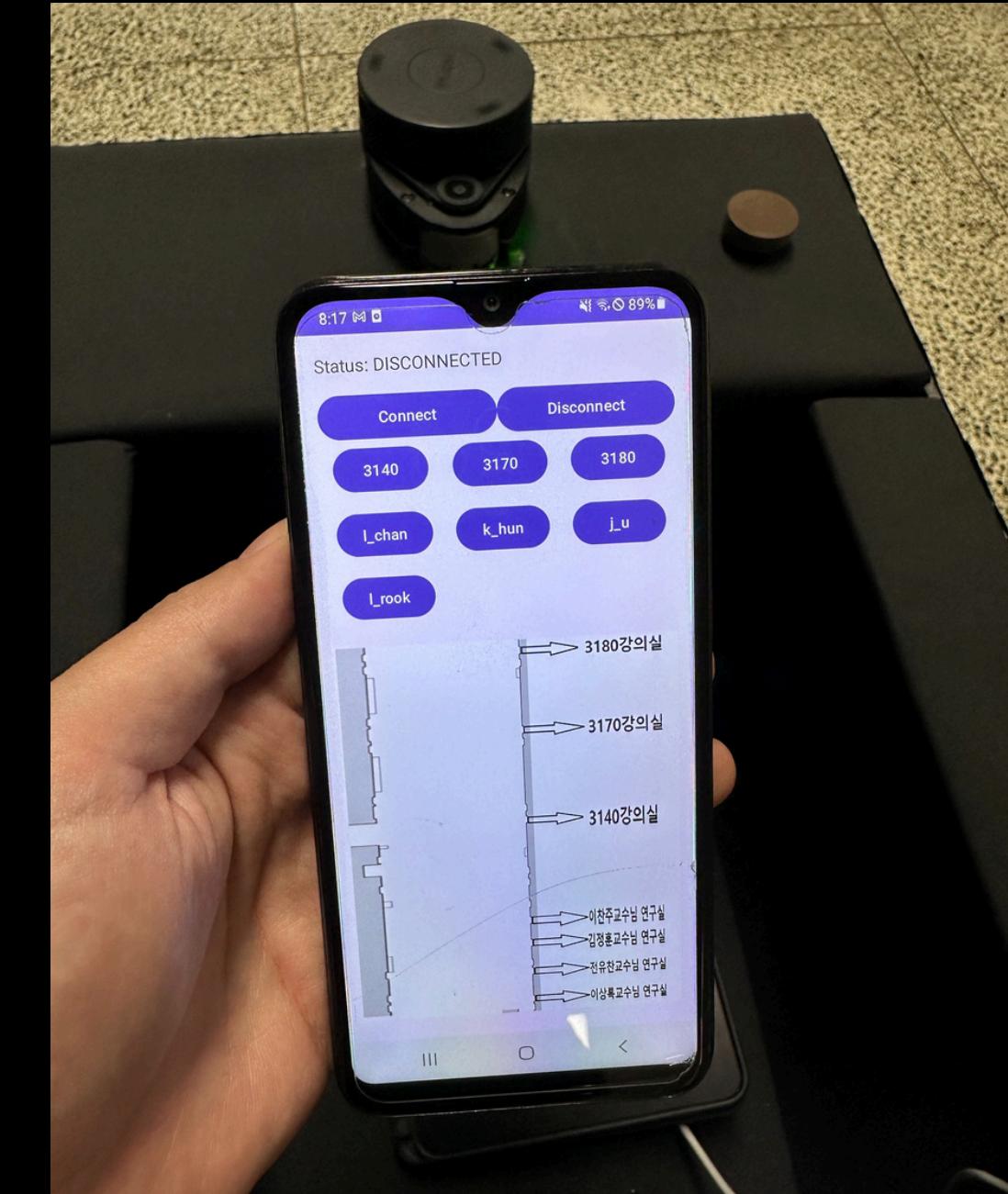


EasyEDA 프로그램 사용
배달로봇 전용 PCB 제작

Section4. 통합 결과



배달로봇 후면

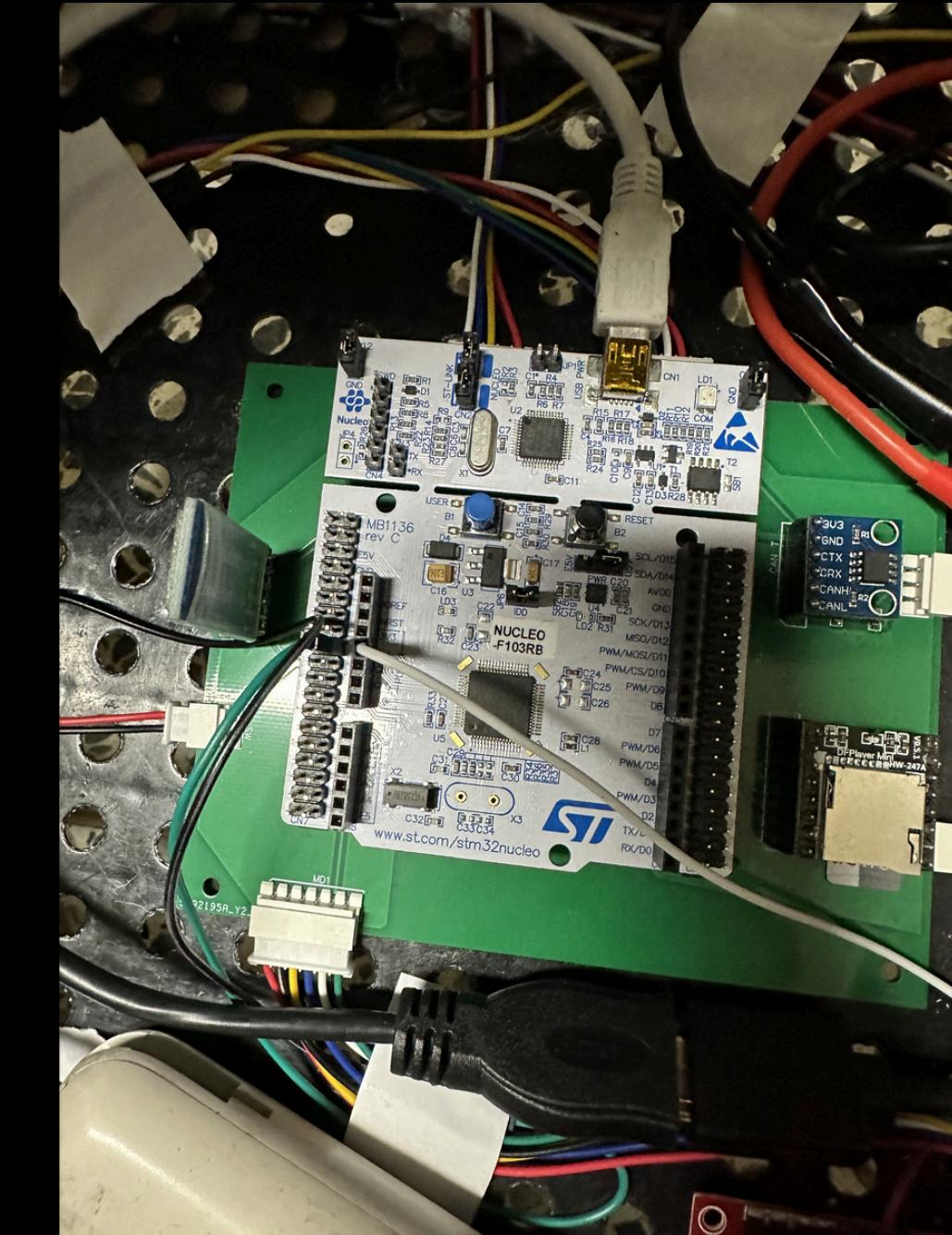


안드로이드 휴대폰 앱

Section4. 통합 결과

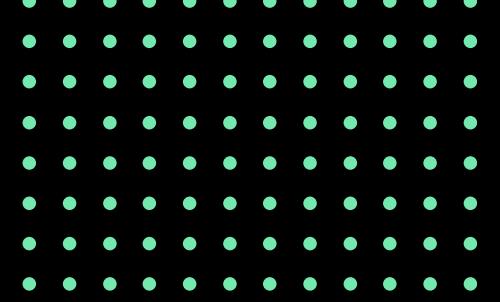


ToF센서



PCB 최적화

Section4. 통합 결과



동작 동선

중앙현관 → 3170 강의실 →
장애물 감지 → 중앙현관 복귀

THANK YOU

20200840 정성빈, 20221713 장형중, 20221714 최정우