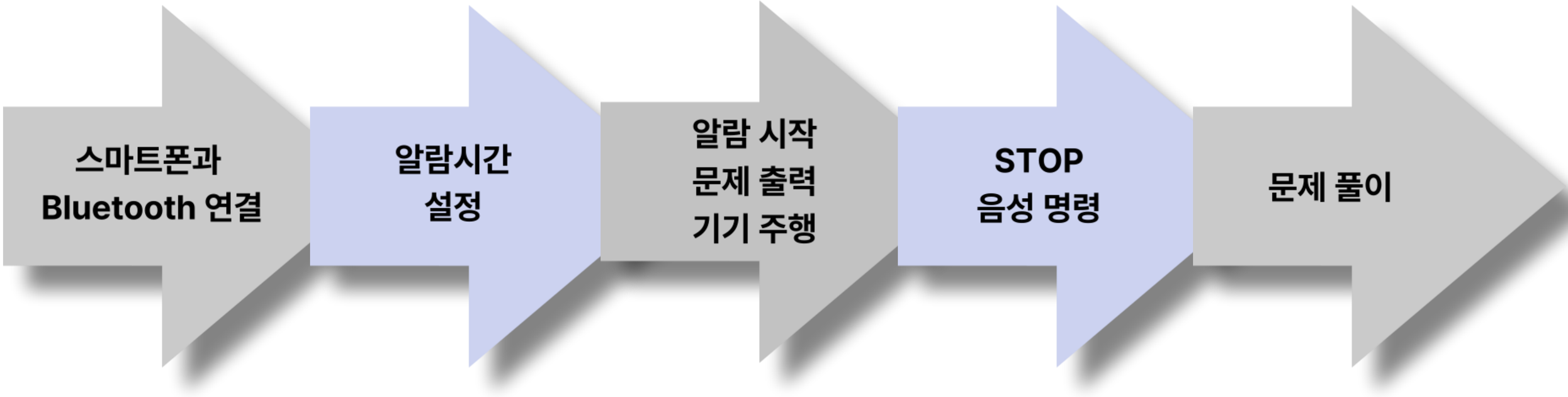
Run away Smart Alarm

임베디드시스템설계및실험 최종 결과 발표

9조
강선호
박형주
손봉국
김주송

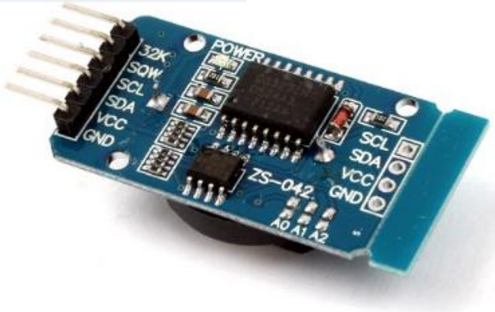
Run away Smart Alarm - 시나리오



Run away Smart Alarm - 사용센서

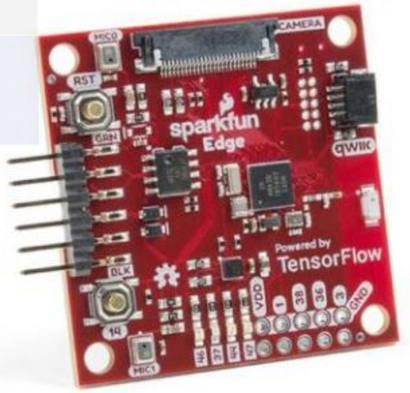
- **시간 측정 부분**

RTC 고정밀 리얼타임 클럭 모듈



- **음성 인식 부분**

SparkFun Edge Development Board

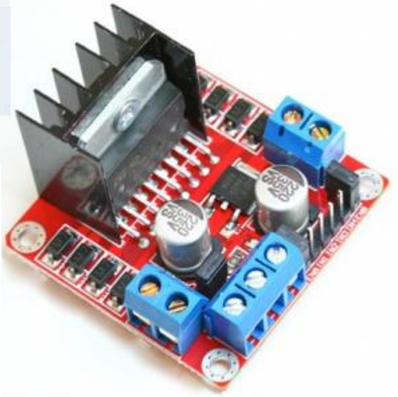


- **알람 발생 부분**

알람 경보용 피에조 부부저

- **기기 주행 부분**

DC모터 및 모터 드라이버 모듈



Run away Smart Alarm - 주요기능

- **시간 출력**

  RTC 고정밀 리얼타임 클럭 모듈을 이용하여 현재 시간을 출력한다.

- **알람 설정 및 알람 출력**

  사용자는 스마트폰을 이용하여 원하는 시간에 알람을 지정한다.
  기기는 설정된 시간에 알람을 출력하고 문제를 제시한다.

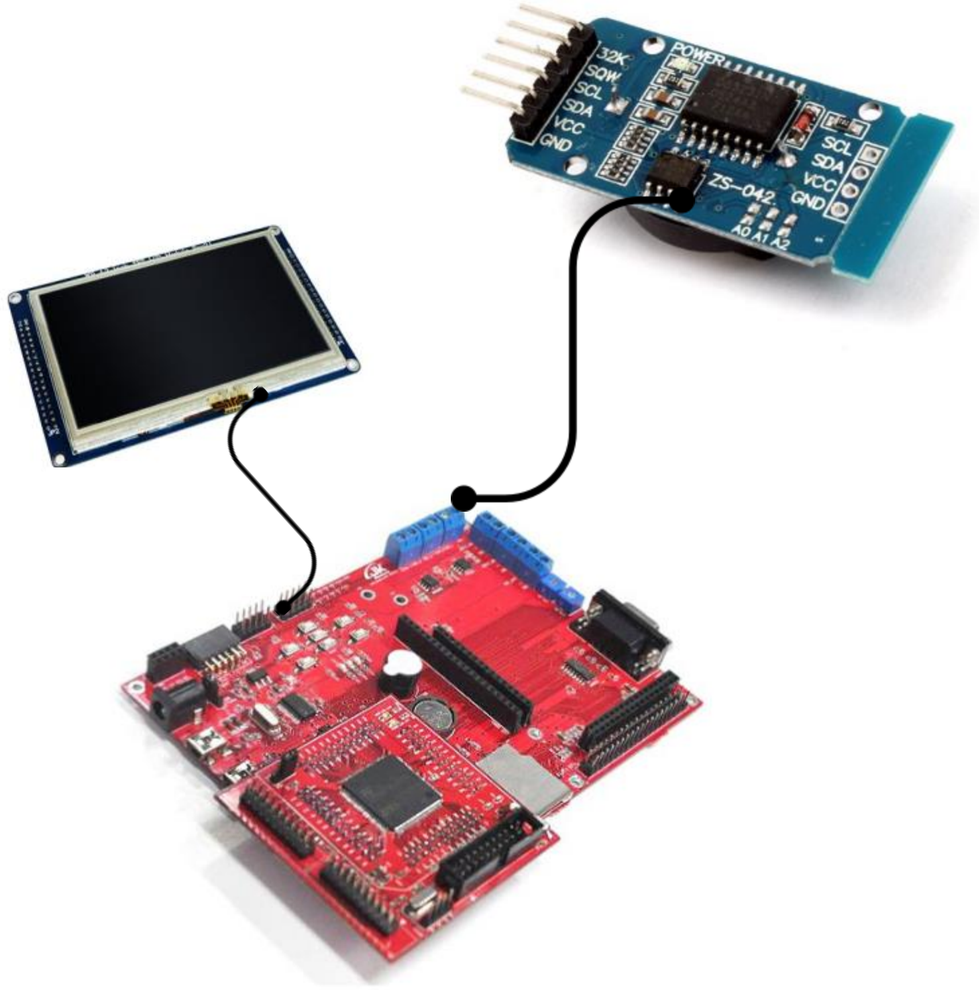
- **기기 주행**

  사용자가 지정한 알람이 울린 후, 기기는 장애물을 피하며 이동한다.
  사용자가 문제 풀이를 완료하면 기기는 동작을 멈춘다.

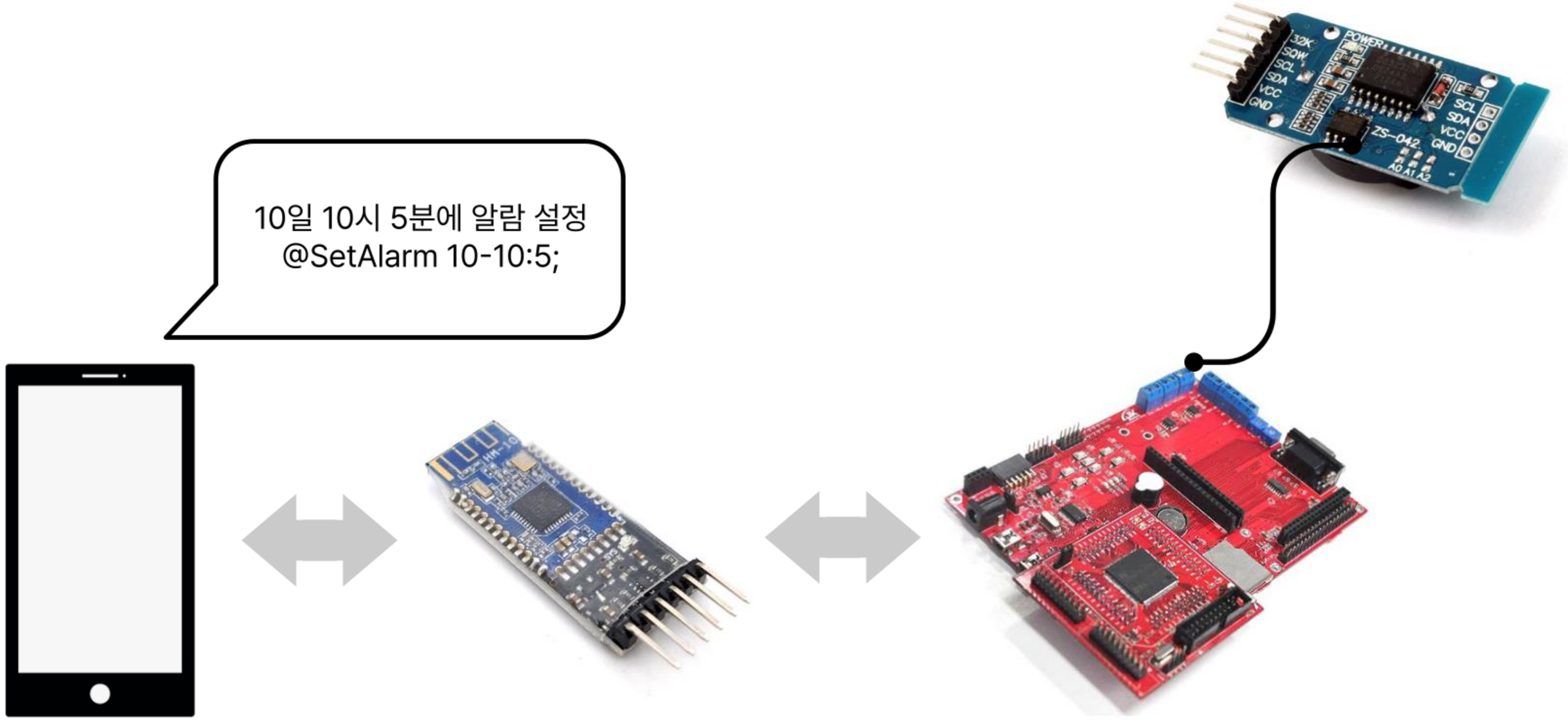
작동 원리

(1) 시간출력

날짜	23 / 12 / 22
현재 시각	12 : 33 : 45
알람 시각	12 : 40 : 0



(2) 알람설정

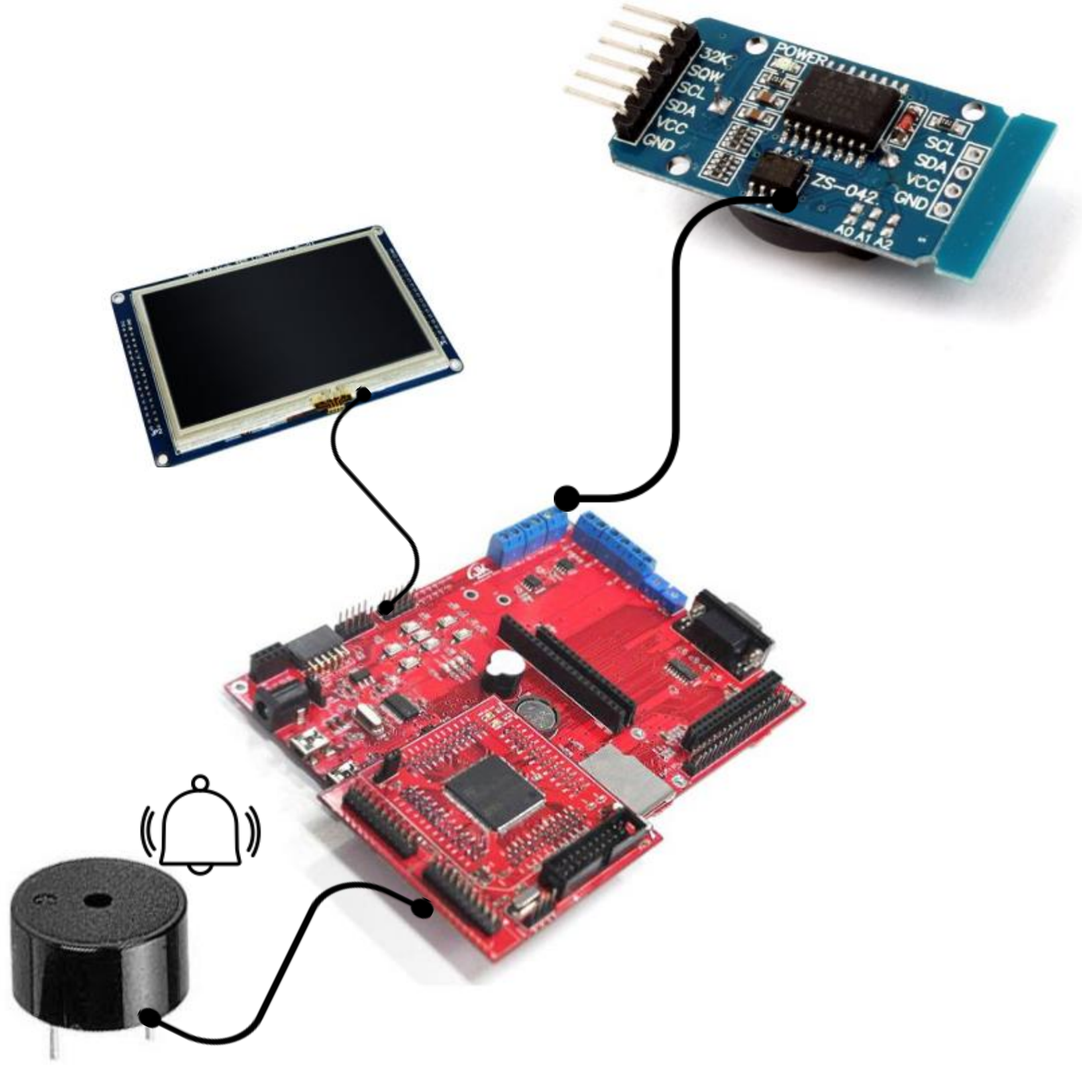

10일 10시 5분에 알람 설정
@SetAlarm 10-10:5;
32K
SQW
SCL
SDA
VCC
GND
POWER
ZS-042
SCL
SDA
VCC
GND
A0 A1 A2

(3) 알람출력

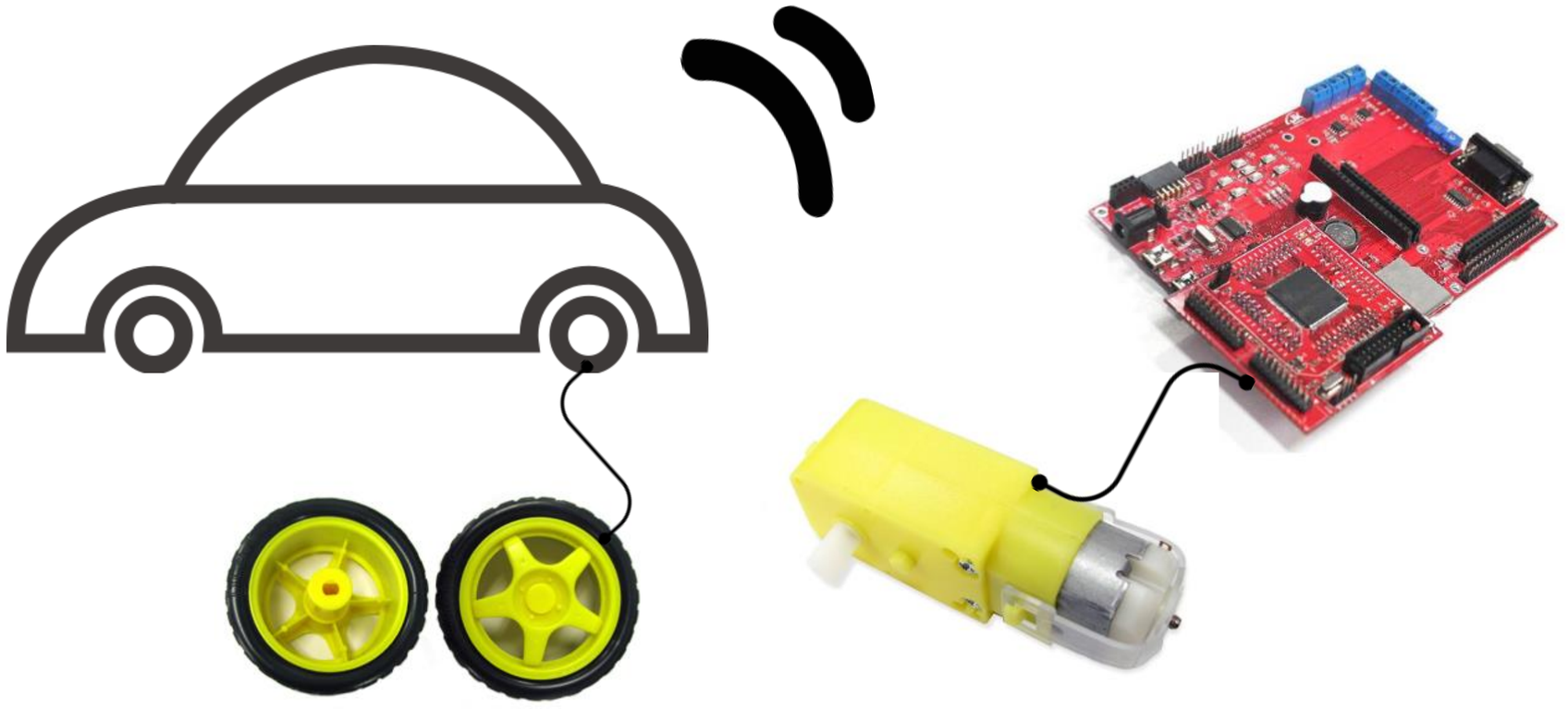
Solve the problem

10 x 30 = ?

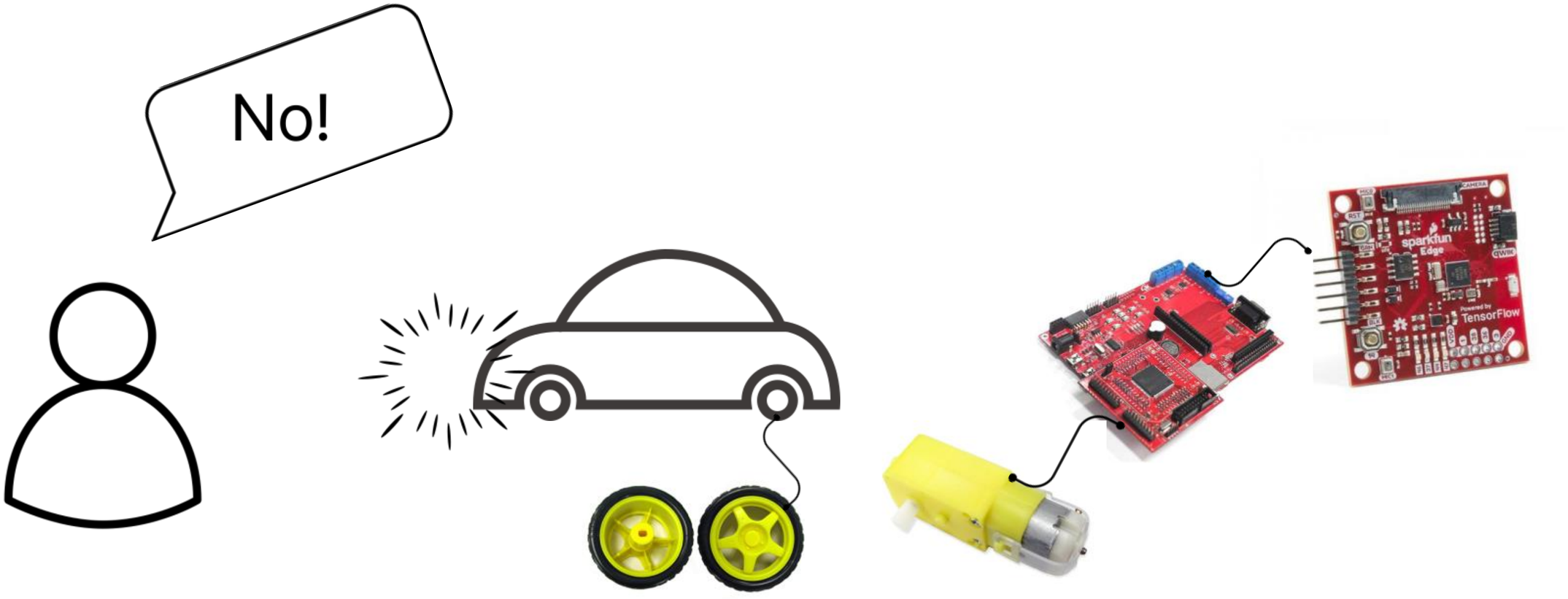
alarm_on



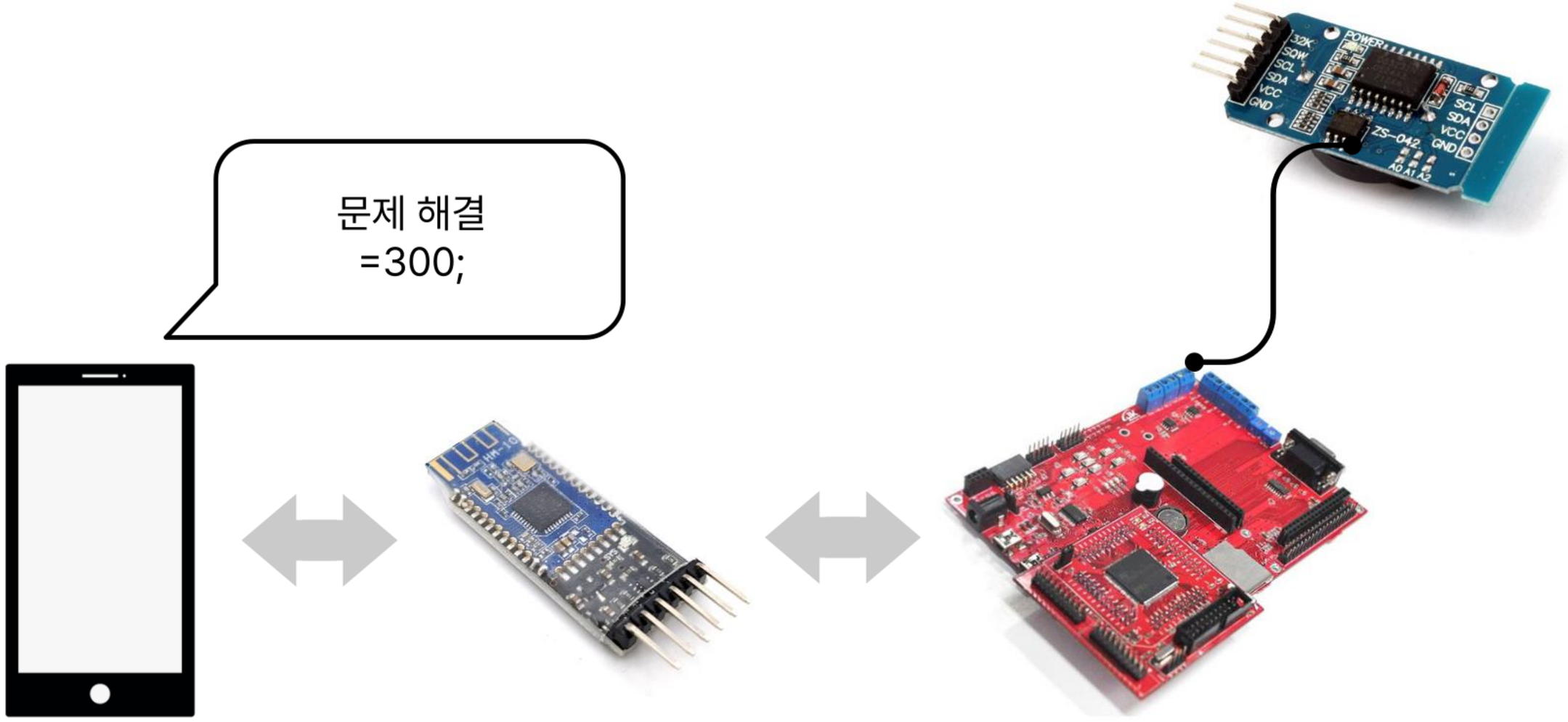
(4) 기기주행



(5) 음성인식

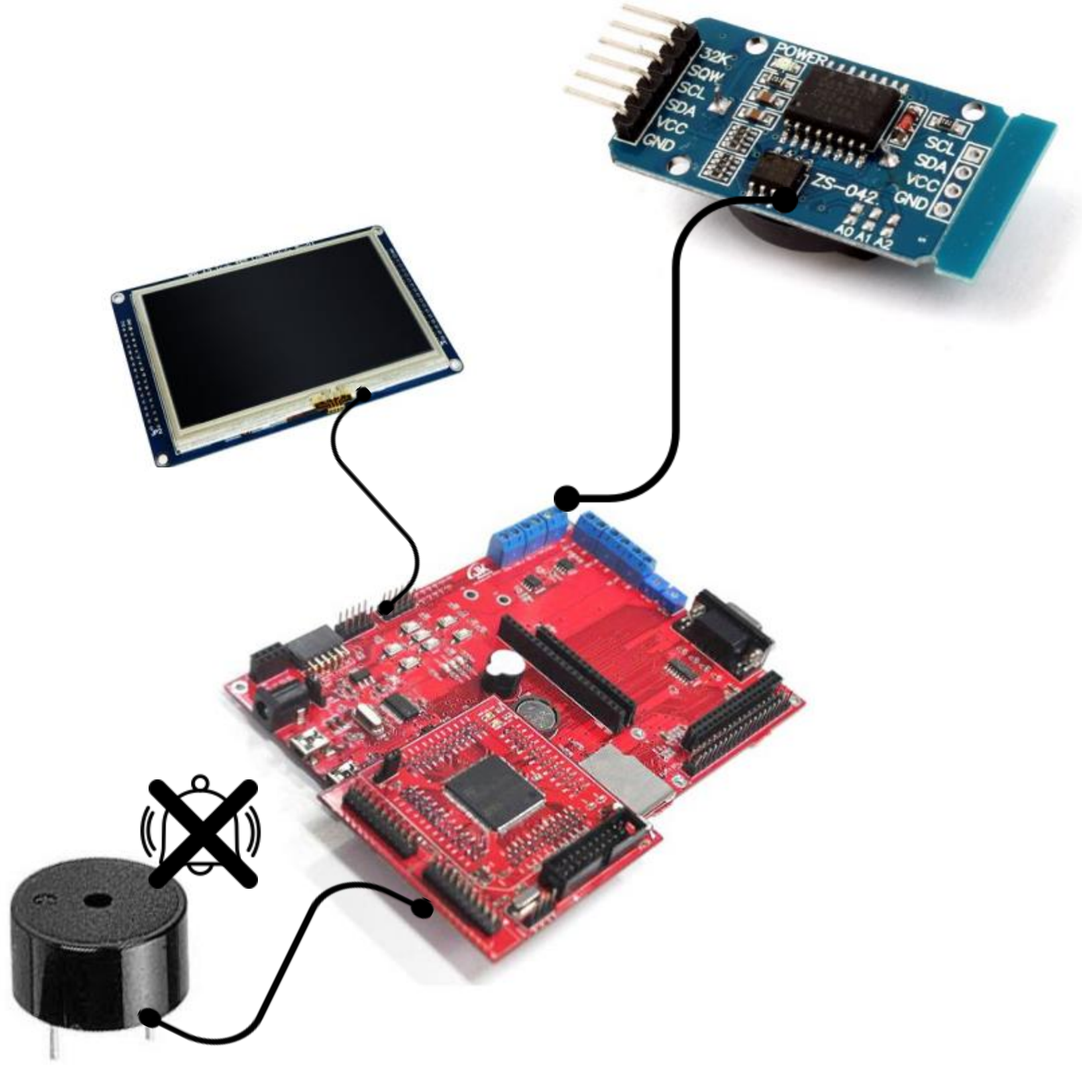


(6) 문제풀이 및 알람종료


문제 해결
=300;
32K
SQW
SCL
SDA
VCC
GND
POWER
SCL
SDA
VCC
GND
ZS-042
A0 A1 A2

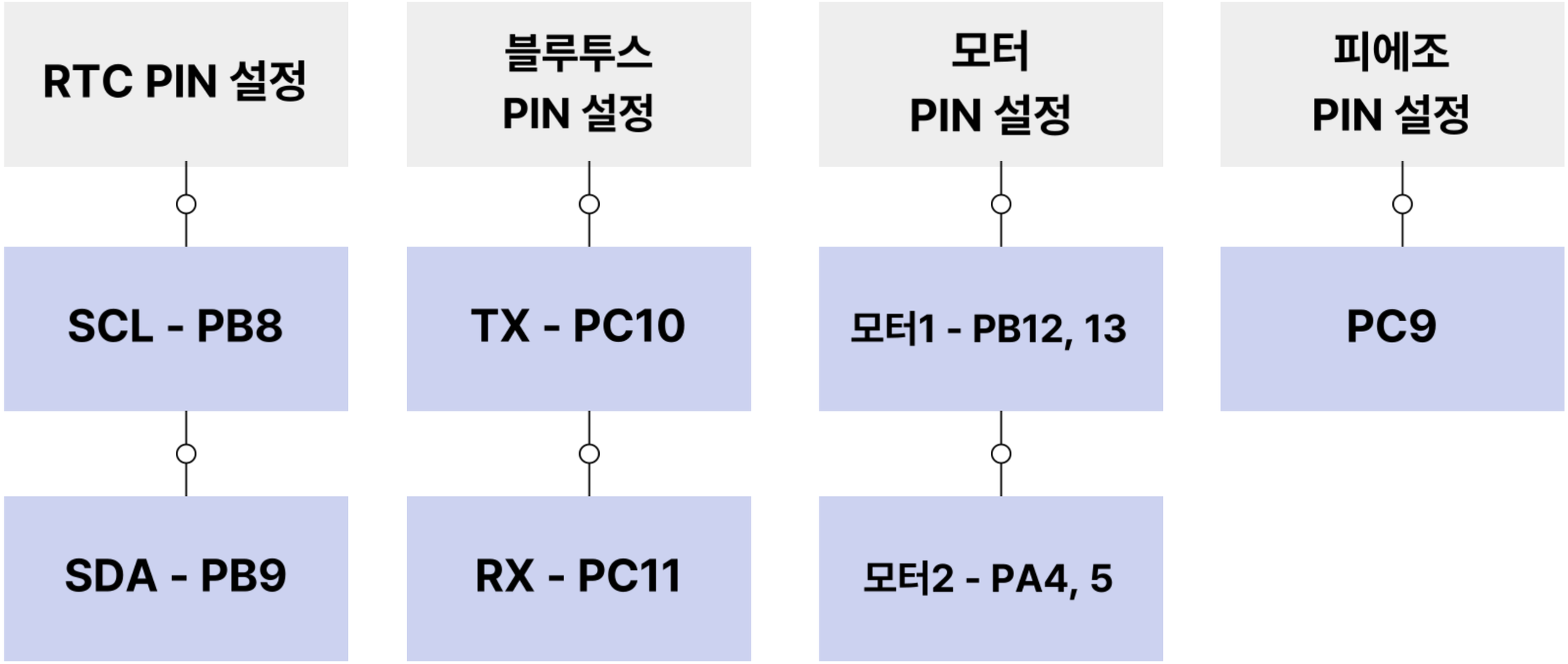
(6) 문제풀이 및 알람종료

날짜	23 / 12 / 22
현재 시각	12 : 45 : 00
알람 시각	12 : 40 : 0



코드 구성

(1) GPIO 핀 연결 + 클락 인가



(2) 알람출력

RTC 주소에 접근해 데이터를 읽은 후, 데이터 버퍼에 저장

```
void ds3231_read_time(ds3231_time *current_time)
{
    uint8_t ds3231_read_time_buff[7];
    //DS3231로부터 읽어 올 데이터를 저장할 데이터 버퍼 (초,분,시간,요일,날짜,달,년도-총 7개)

    /*
    특정 메모리 주소에 액세스해서 원하는 갯수만큼 데이터를 읽어와서 데이터 버퍼에 저장
    */
    HW_I2C_Read(I2C1, ds3231_addr,ds3231_sec_addr, 1, &ds3231_read_time_buff[0]);
    HW_I2C_Read(I2C1, ds3231_addr,ds3231_min_addr, 1, &ds3231_read_time_buff[1]);
    HW_I2C_Read(I2C1, ds3231_addr,ds3231_hour_addr, 1, &ds3231_read_time_buff[2]);
    HW_I2C_Read(I2C1, ds3231_addr,ds3231_day_addr, 1, &ds3231_read_time_buff[3]);
    HW_I2C_Read(I2C1, ds3231_addr,ds3231_date_addr, 1, &ds3231_read_time_buff[4]);
    HW_I2C_Read(I2C1, ds3231_addr,ds3231_month_addr, 1, &ds3231_read_time_buff[5]);
    HW_I2C_Read(I2C1, ds3231_addr,ds3231_year_addr, 1, &ds3231_read_time_buff[6]);
```

(2) 알람출력

RTC에서 현재 시각 읽기

```
void ds3231_write_time(ds3231_time *ds3231_write_time_struct)
{
    uint8_t write_buf[7];   //전송에 사용할 데이터 버퍼 배열 선언

    //입력한 10진수 시간 데이터를 2진수화
    write_buf[0]=decTobcd(ds3231_write_time_struct->sec);
    write_buf[1]=decTobcd(ds3231_write_time_struct->min);
    write_buf[2]=decTobcd(ds3231_write_time_struct->hour_select.hour);
    write_buf[3]=decTobcd(ds3231_write_time_struct->day);
    write_buf[4]=decTobcd(ds3231_write_time_struct->date);
    write_buf[5]=decTobcd(ds3231_write_time_struct->month);
    write_buf[6]=decTobcd(ds3231_write_time_struct->year);
```

(3) 알람설정

설정한 값에 따라 알람 설정

```
void ds3231_set_alarm1(ds3231_Alarm1 *alarm1_data)
{
    uint8_t alarm1_buff[4]; //DS3231에 전송할 알람1 데이터 버퍼

    alarm1_buff[0]=decTobcd(alarm1_data->sec);  //초
    alarm1_buff[1]=decTobcd(alarm1_data->min);  //분
    alarm1_buff[2]=decTobcd(alarm1_data->hour_select.hour); //시간
    alarm1_buff[3]=decTobcd(alarm1_data->day_date_select.value);    //날짜 or 요일값
```

(4) 알람실행

현재 시각이 알람 시각에 도달한 이후, 알람 실행

```
int alarm_check(ds3231_time *current_time, ds3231_Alarm1 *alarm1_data) {


  if (current_time->date >= alarm1_data->day_date_select.value) {
    if (current_time->hour_select.am_pm_24 == alarm1_data->hour_select.am_pm_24) {
      if (current_time->hour_select.hour >= alarm1_data->hour_select.hour) {
        if (current_time->min >= alarm1_data->min) {
          if (current_time->sec >= alarm1_data->sec) {
            if (Alarm_flag == 0) { // 알람 작동시
              Alarm_flag = 1;
              Alarm_ONOFF = 1;
            }
            else {
              Alarm_ONOFF = 0;
            }
          }
```

(5) 기기주행

알림 실행 후, 모터 제어 및 기기 수행

```
void setDirectionToFront(void) {
    GPIO_SetBits(GPIOB, LEFT_HIGH);
    GPIO_ResetBits(GPIOB, LEFT_LOW);
    GPIO_SetBits(GPIOA, RIGHT_HIGH);
    GPIO_ResetBits(GPIOA, RIGHT_LOW);
}

void setDirectionToBack(void) {
    GPIO_ResetBits(GPIOB, LEFT_HIGH);
    GPIO_SetBits(GPIOB, LEFT_LOW);
    GPIO_ResetBits(GPIOA, RIGHT_HIGH);
    GPIO_SetBits(GPIOA, RIGHT_LOW);
}
```

```
GPIO_SetBits(GPIOC, GPIO_Pin_9);
```

(6) 문제풀이

문제 설정 후 TFT-LCD에 출력

```
if(alarm_check(&ds_time_default, &alarm1_default)){

  int sec_problem = alarm1_default.sec;
  int min_problem = alarm1_default.min;
  int hour_problem = alarm1_default.hour_select.hour;


 problem1 = sec_problem + hour_problem;
 problem2 = min_problem * hour_problem;

  LCD_ShowString(70, 100, "Solve the problem", BLACK, WHITE);
  LCD_ShowNum(90, 150, problem1, 2, BLACK, WHITE);
  LCD_ShowString(110, 150, "X", BLACK, WHITE);
  LCD_ShowNum(130, 150, problem2, 2, BLACK, WHITE);
  LCD_ShowString(150, 150, "=", BLACK, WHITE);
  LCD_ShowString(170, 150, "?", BLACK, WHITE);
  LCD_ShowString(130, 300, "Alarm_on", 0x0000, 0xFFFF);
```

(7) 알람종료

제대로 된 정답이 입력된 후, 알람 종료

```
void is_answer(void) {

  char String[10];
  int st = 10-string_count;

  for (int i = 0; i < 10; i++) {
    String[i] = 0x30;
  }

  for (int i = 0; i < string_count; i++) {
    String[st+i] = (char)receive_string[i];
    receive_string[i] = 0;
 if (answer_correct) {
    LCD_Clear(WHITE);
    GPIO_ResetBits(GPIOC, GPIO_Pin_9); // 부저 종료
    Delay_little();
    LCD_ShowString(30, 300, "Alarm_off", 0x0000, 0xFFFF);
    LCD_Clear(WHITE);
    answer_correct=0;
    stopAllWheel();
    break;
 }
```

(+) 음성인식

수집된 음성 데이터를 이미지 파일로 변환

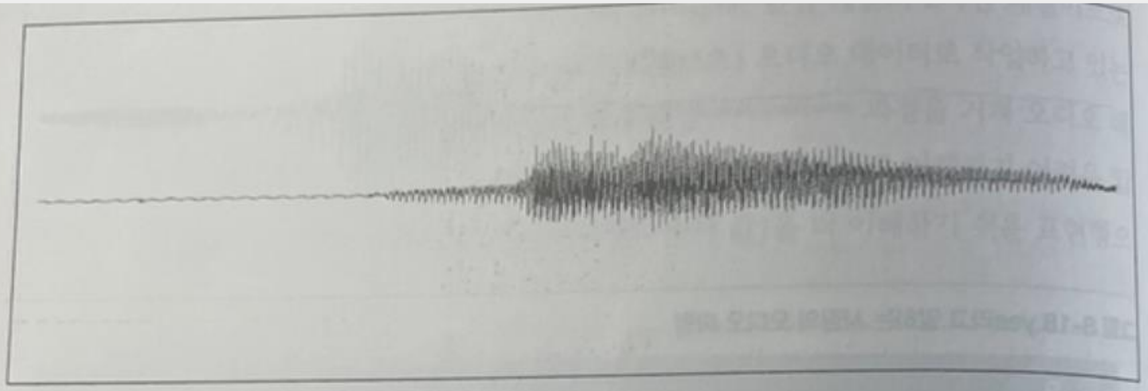


그림 8-21 no라고 말하는 사람의 다른 오디오 파형

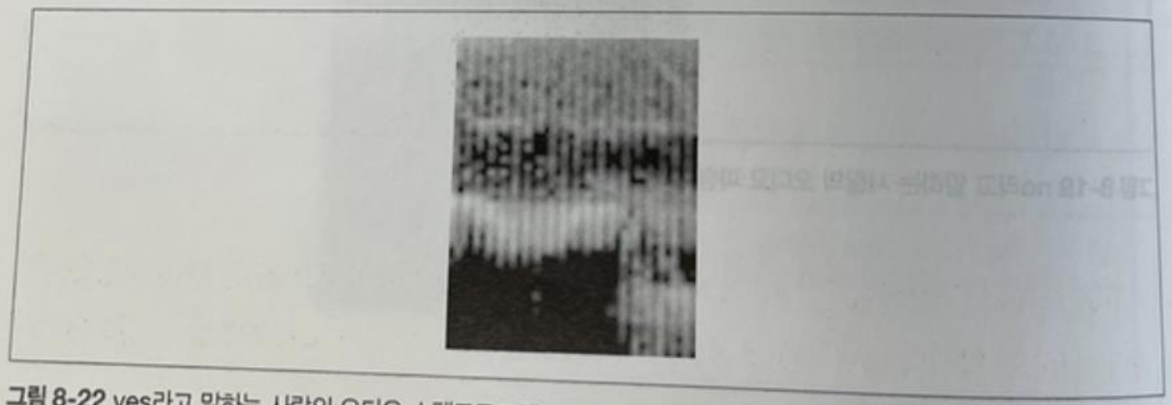


그림 8-22 yes라고 말하는 사람의 오디오 스펙트로그램

(+) 음성인식

Convolution 모델을 사용하여 훈련

Training

The following script downloads the dataset and begin training.

```
[ ]  !python tensorflow/tensorflow/examples/speech_commands/train.py
     --data_dir={DATASET_DIR} ₩
     --wanted_words={WANTED_WORDS} ₩
     --silence_percentage={SILENT_PERCENTAGE} ₩
     --unknown_percentage={UNKNOWN_PERCENTAGE} ₩
     --preprocess={PREPROCESS} ₩
     --window_stride={WINDOW_STRIDE} ₩
     --model_architecture={MODEL_ARCHITECTURE} ₩
     --how_many_training_steps={TRAINING_STEPS} ₩
     --learning_rate={LEARNING_RATE} ₩
     --train_dir={TRAIN_DIR} ₩
     --summaries_dir={LOGS_DIR} ₩
     --verbosity={VERBOSITY} ₩
     --eval_step_interval={EVAL_STEP_INTERVAL} ₩
     --save_step_interval={SAVE_STEP_INTERVAL}
```

(+) 음성인식

모델 양자화

```
!cat {MODEL_TFLITE_MICRO}
```

```
unsigned char g_model[] = {
  0x20, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x00, 0x00, 0x00, 0x00,
  0x14, 0x00, 0x20, 0x00, 0x1c, 0x00, 0x18, 0x00, 0x14, 0x00, 0x10, 0x00,
  0x0c, 0x00, 0x00, 0x00, 0x08, 0x00, 0x04, 0x00, 0x14, 0x00, 0x00, 0x00,
  0x1c, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00, 0x74, 0x00, 0x00, 0x00,
  0xcc, 0x42, 0x00, 0x00, 0xdc, 0x42, 0x00, 0x00, 0x70, 0x49, 0x00, 0x00,
  0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0x00,
  0x34, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0xdc, 0xff, 0xff, 0xff,
  0x0e, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x13, 0x00, 0x00, 0x00,
  0x43, 0x4f, 0x4e, 0x56, 0x45, 0x52, 0x53, 0x49, 0x4f, 0x4e, 0x5f, 0x4d,
  0x45, 0x54, 0x41, 0x44, 0x41, 0x54, 0x41, 0x00, 0x08, 0x00, 0x0c, 0x00,
  0x08, 0x00, 0x04, 0x00, 0x08, 0x00, 0x00, 0x00, 0x0d, 0x00, 0x00, 0x00,
  0x04, 0x00, 0x00, 0x00, 0x13, 0x00, 0x00, 0x00, 0x6d, 0x69, 0x6e, 0x5f,
  0x72, 0x75, 0x6e, 0x74, 0x69, 0x6d, 0x65, 0x5f, 0x76, 0x65, 0x72, 0x73,
```

(+) 음성인식

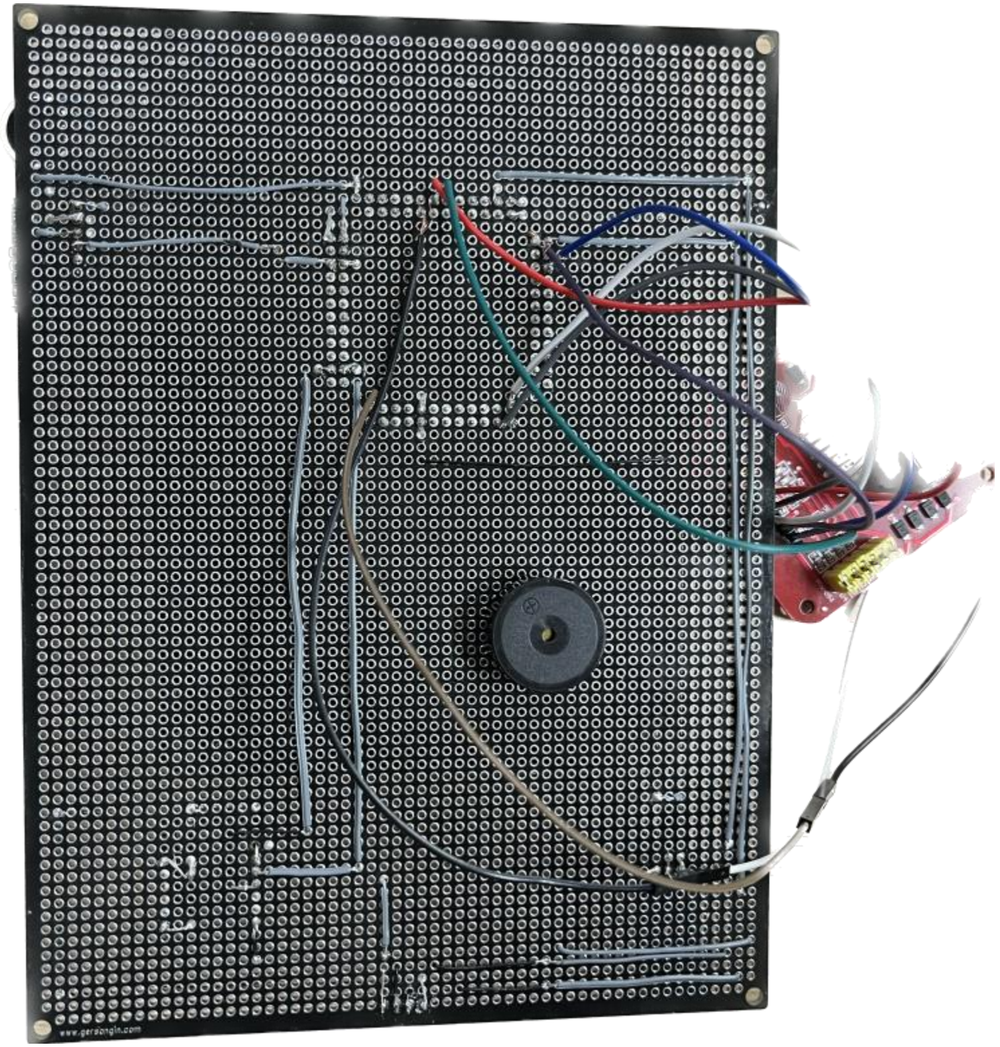
학습 결과

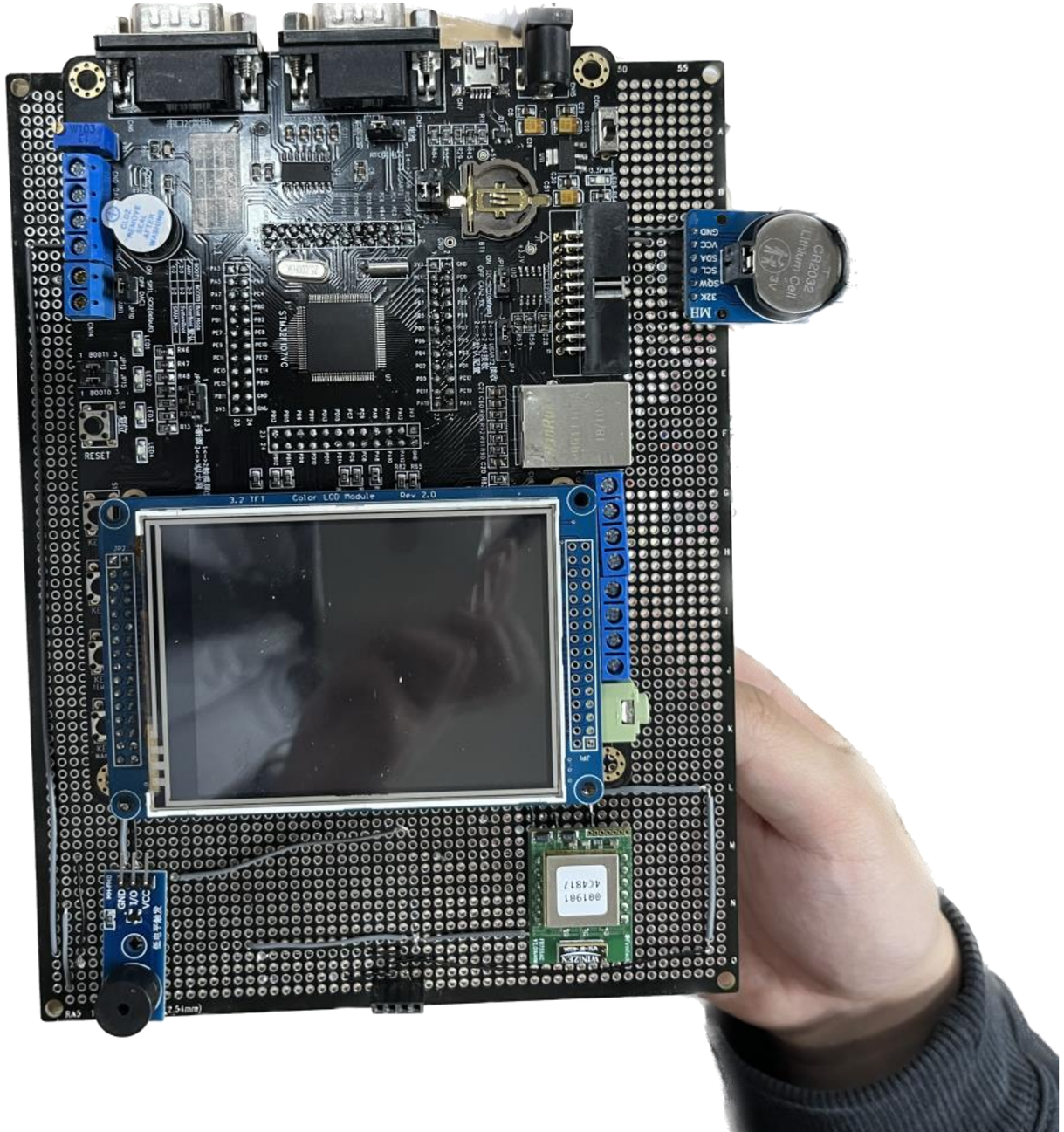
```
[ ]  # Compute float model accuracy
     run_tflite_inference(FLOAT_MODEL_TFLITE)

     # Compute quantized model accuracy
     run_tflite_inference(MODEL_TFLITE, model_type='Quantized')
```

```
Float model accuracy is 90.171990% (Number of test samples=1221)
Quantized model accuracy is 90.171990% (Number of test samples=1221)
```

```
d unknown (223) @39936ms
d unknown (213) @41984ms
d unknown (211) @46080ms
d unknown (216) @51264ms
d unknown (202) @56768ms
d unknown (210) @64000ms
d unknown (210) @68416ms
d yes (205) @69504ms
d unknown (227) @76352ms
d unknown (206) @78784ms
d unknown (201) @83584ms
d unknown (206) @89408ms
d yes (211) @91136ms
d unknown (208) @97024ms
d unknown (204) @107328ms
d unknown (206) @115904ms
d yes (207) @117312ms
d unknown (230) @123456ms
d unknown (225) @126592ms
d unknown (212) @128640ms
d no (225) @136576ms
d unknown (212) @139648ms
d unknown (202) @142400ms
```

Run away Smart Alarm - 회로구성





Run away Smart Alarm - 완성모습