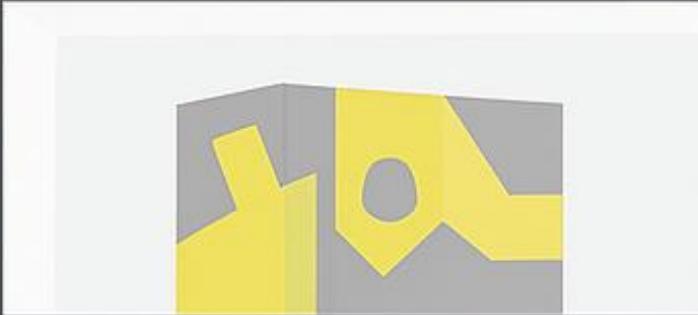


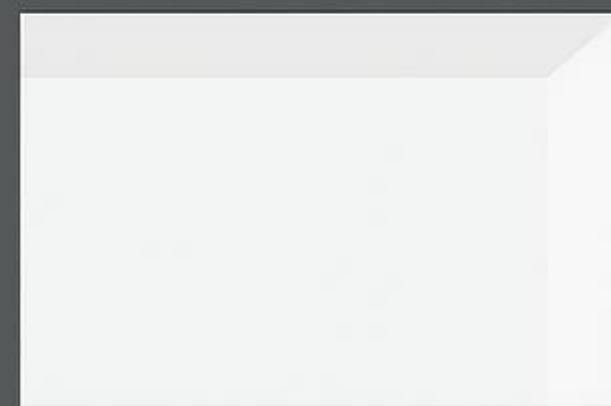
스마트 사물함



SCREEN



PRODUCT



#임억

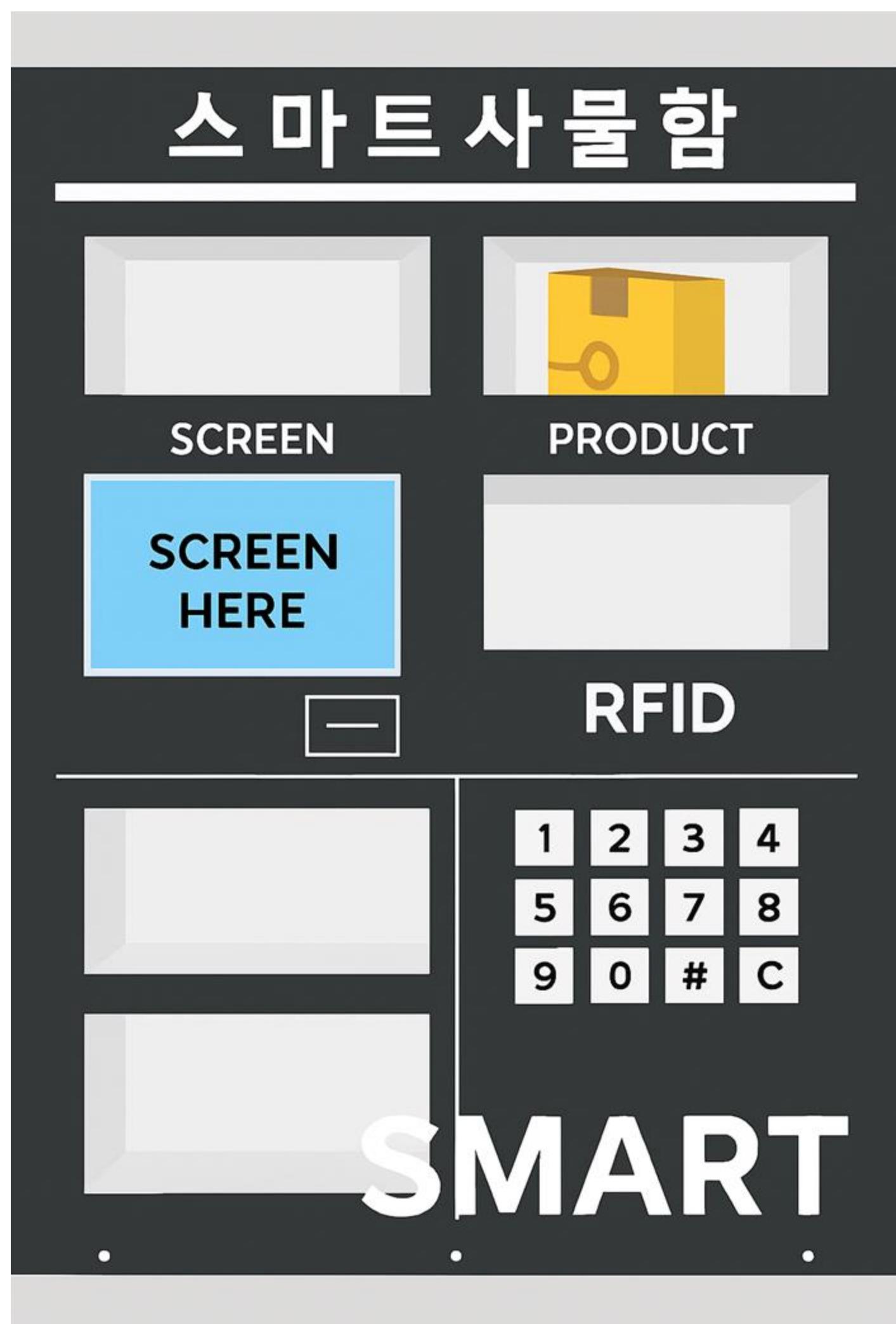
| | | | |
|---|---|---|---|
| 1 | 2 | 4 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 0 | # | C |

RFID SMART

15조
조민재
허진경

목 차

1. 프로젝트 개요
2. 구성
3. 흐름도
4. 주요 코드
5. 시연영상
6. 향후 계획



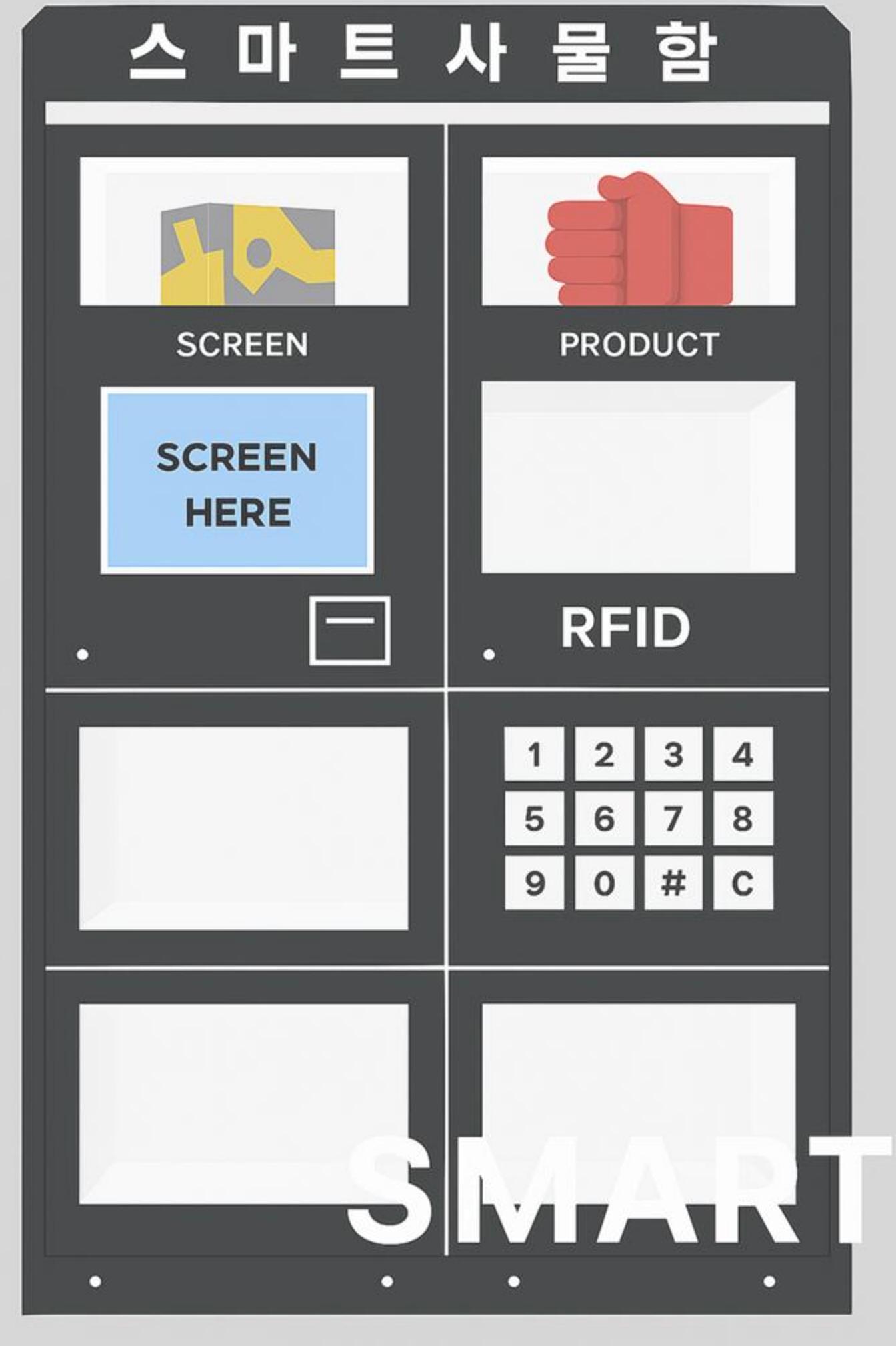
프로젝트 개요

스마트 사물함 프로젝트

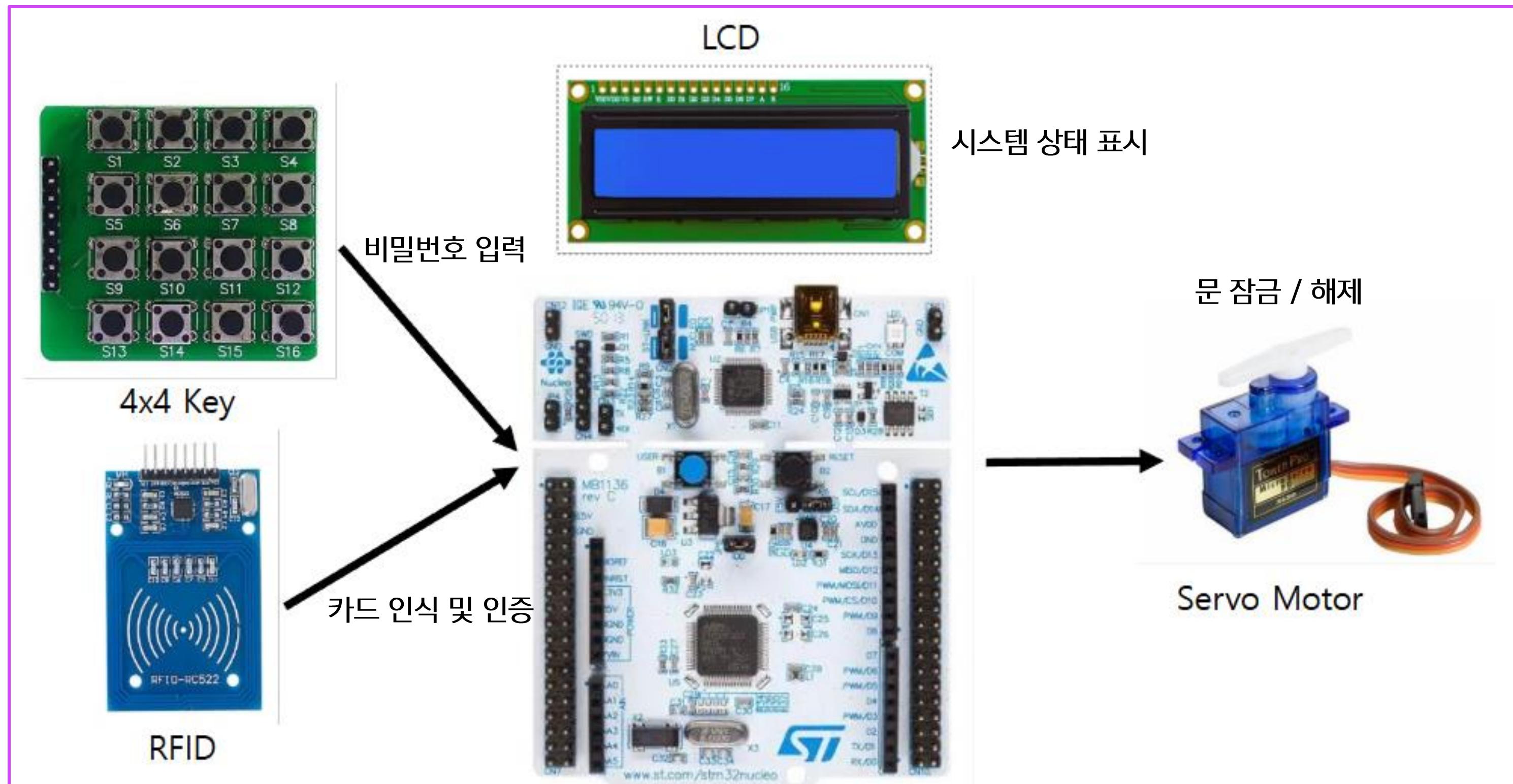
STM32F411RE 마이크로컨트롤러를 중심으로 4x4 키패드, RFID, LDC, 서보 모터를 결합하여 구현한 스마트 사물함 프로젝트입니다. 이 프로젝트는 기존 기계식 사물함의 한계를 넘어선 지능형 보안 시스템을 구축하고자 합니다.

프로젝트 목적

- RFID 카드와 비밀번호 이중 인증 방식 제공
- 인증 성공 시 서보 모터로 문 자동 제어
- LCD 화면을 통한 직관적인 상태 표시
- 사용자 편의성과 보안성을 극대화



구성



구성

GPIO 설정

Configuration

Group By Peripherals

GPIO Single Mapped Signals I2C RCC SPI SYS TIM USART NVIC

Search Signals

Show only Modified Pins

| Pin Name | Signal on Pin | GPIO output le... | GPIO mode | GPIO Pull-up/P... | Maximum outp... | User Label | Modified |
|----------|---------------|-------------------|----------------|--------------------|-----------------|------------|-------------------------------------|
| PA0-WKUP | n/a | Low | Output Push... | Pull-up | Low | R1 | <input checked="" type="checkbox"/> |
| PA1 | n/a | Low | Output Push... | Pull-up | Low | R2 | <input checked="" type="checkbox"/> |
| PA4 | n/a | n/a | Input mode | Pull-up | n/a | C1 | <input checked="" type="checkbox"/> |
| PA5 | n/a | n/a | Input mode | Pull-up | n/a | C2 | <input checked="" type="checkbox"/> |
| PA6 | n/a | n/a | Input mode | Pull-up | n/a | C3 | <input checked="" type="checkbox"/> |
| PA7 | n/a | n/a | Input mode | Pull-up | n/a | C4 | <input checked="" type="checkbox"/> |
| PB0 | n/a | Low | Output Push... | Pull-up | Low | R3 | <input checked="" type="checkbox"/> |
| PB10 | n/a | Low | Output Push... | Pull-up | Low | R4 | <input checked="" type="checkbox"/> |
| PC8 | n/a | High | Output Push... | No pull-up and ... | Low | RST | <input checked="" type="checkbox"/> |
| PC9 | n/a | Low | Output Push... | No pull-up and ... | Low | CS(SDA) | <input checked="" type="checkbox"/> |

- R1~4 / C1~4 : 4x4 Key
- CS(SDA) : RFID 데이터 전송라인
- RST : RFID RESET

TIM1 설정

TIM1 Mode and Configuration

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings DMA Settings

Configure the below parameters :

▼ Counter Settings

| | |
|---|-------------|
| Prescaler (PSC - 16 bits value) | 83 |
| Counter Mode | Up |
| Counter Period (AutoReload Register - 16 bits value) | 19999 |
| Internal Clock Division (CKD) | No Division |
| Repetition Counter (RCR - 8 bits value) | 0 |
| auto-reload preload | Disable |

서보 모터 PWM 주기

- Prescaler : $84 - 1 = 83\text{Mhz}$
- Counter Period : $20000 - 1 = 19999$
- 타이머 주기 = $(\text{Prescaler} + 1) \times (\text{Period} + 1) / \text{Timer_Clock}$
- $84\text{hz} \times 20000 / 84\text{Mhz} = 50\text{Hz} \rightarrow 20\text{ms}$

구성

SPI 설정

SPI2 Mode and Configuration

Mode

Mode Full-Duplex Master

Hardware NSS Signal Disable

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings DMA Settings

Configure the below parameters :

Search (Ctrl+F)

Basic Parameters

| | |
|--------------|-----------|
| Frame Format | Motorola |
| Data Size | 8 Bits |
| First Bit | MSB First |

Clock Parameters

| | |
|---------------------------|--------------|
| Prescaler (for Baud Rate) | 8 |
| Baud Rate | 5.25 MBits/s |
| Clock Polarity (CPOL) | Low |
| Clock Phase (CPHA) | 1 Edge |

Advanced Parameters

| | |
|-----------------|----------|
| CRC Calculation | Disabled |
| NSS Signal Type | Software |

SPI2 Mode and Configuration

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings DMA Settings GP

Search Signals

Pin Name Signal on Pin GPIO output level GPIO mode GPIO Pull-up/P... Maximum

| | | | | | |
|------|-----------|-----|-------------------|--------------------|-----------|
| PB13 | SPI2_SCK | n/a | Alternate Func... | No pull-up and ... | Very High |
| PB14 | SPI2_MISO | n/a | Alternate Func... | No pull-up and ... | Very High |
| PB15 | SPI2_MOSI | n/a | Alternate Func... | No pull-up and ... | Very High |

- 84MHz / Prescaler 8 = 10.5Mhz
- RFID SPI 최대 속도에 맞춤

전체 시스템 흐름도

1. 전원 ON 및 초기화
2. LCD에 모드 선택 화면 표시
(1: RFID, 2: 비밀번호)
3. 사용자가 입력한 모드에 따라 분기
 1. RFID 모드: 카드를 읽고 등록/인증
 2. 비밀번호 모드: 키패드로
비밀번호 입력 후 인증
4. 인증 성공 시 문 열림, # 입력 시 문

잠김

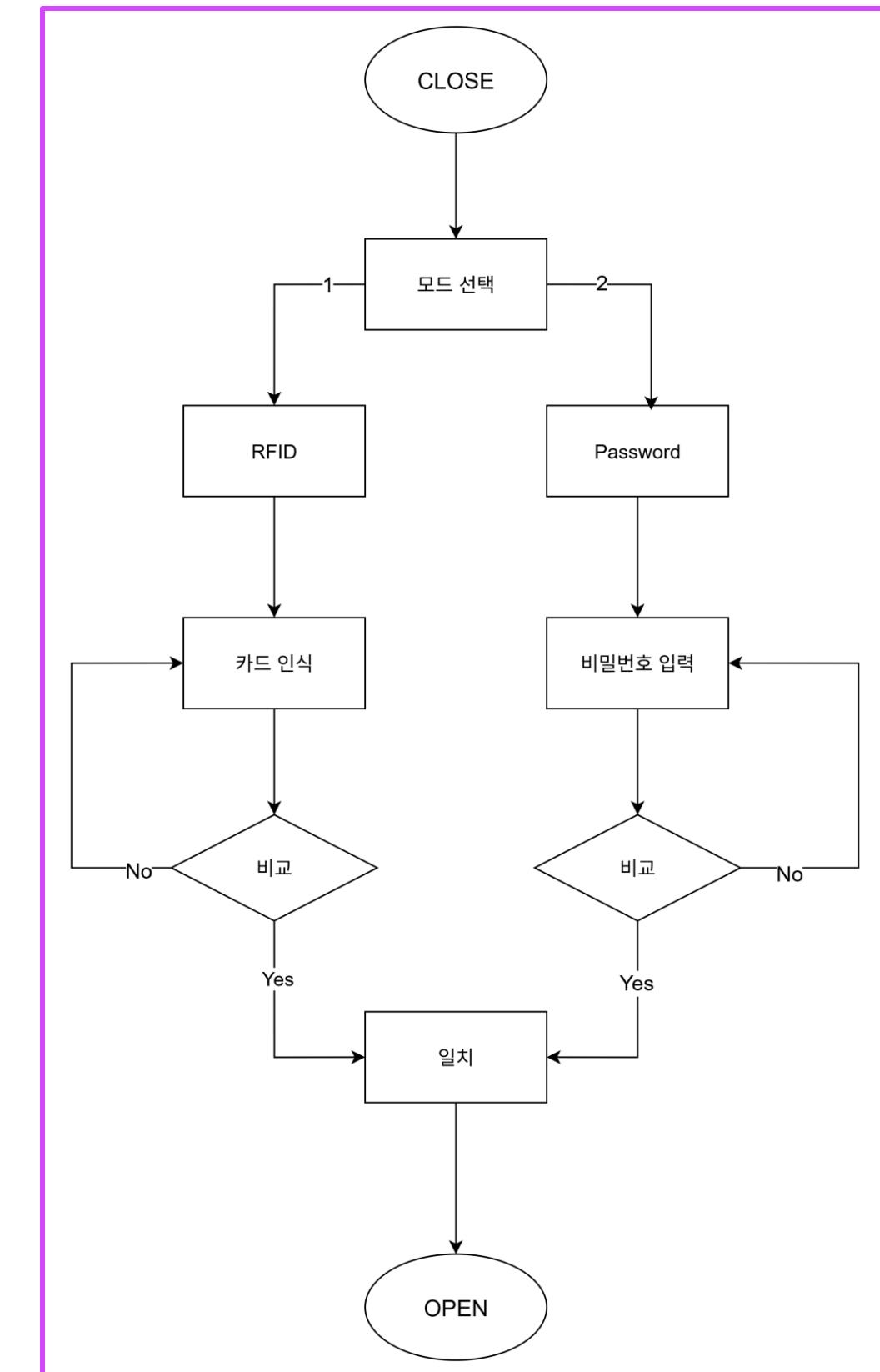


그림 2. 전체 시스템 동작 순서도

키패드 입력 흐름도

1. 비밀번호 모드 진입 시 "Enter Password:" 출력
2. 숫자 4자리 입력 → 로 표시됨
3. D 입력 시 비밀번호 확인
4. 일치 → 문 열림
5. 불일치 → "Access Denied" 출력
6. A 입력 시 비밀번호 변경 루틴 실행

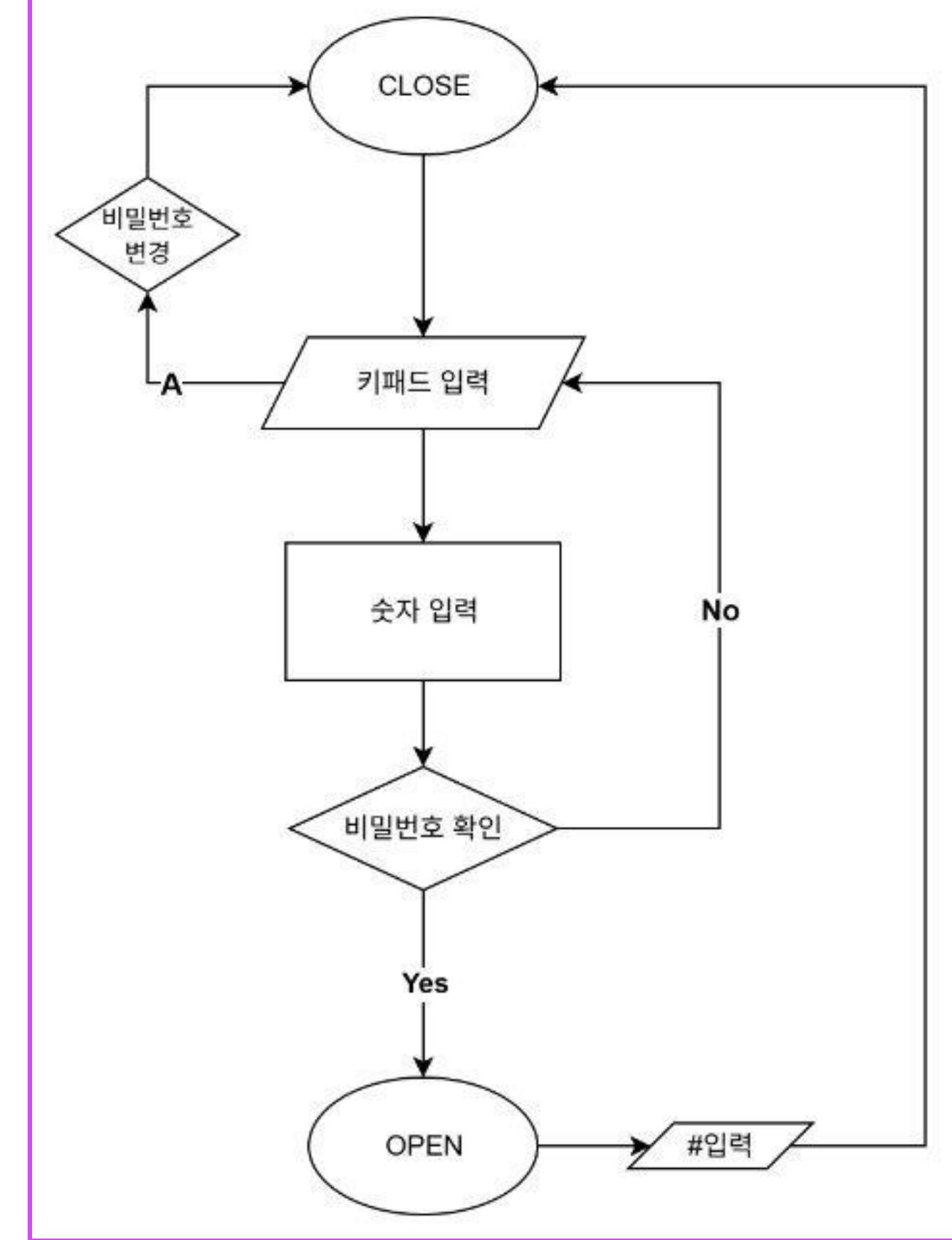


그림 3. 키패드 입력 순서도

RFID 모드 흐름도

1. RFID 모드 진입 시 카드 대기 메시지

출력

2. 카드 감지 → 등록 여부 확인

- 등록 안된 경우: 첫 카드 등록
- 등록된 경우:
 - 일치 → 문 열림
 - 불일치 → "Access Denied"

3. 문이 열린 상태에서 # 누르면 문 잠김

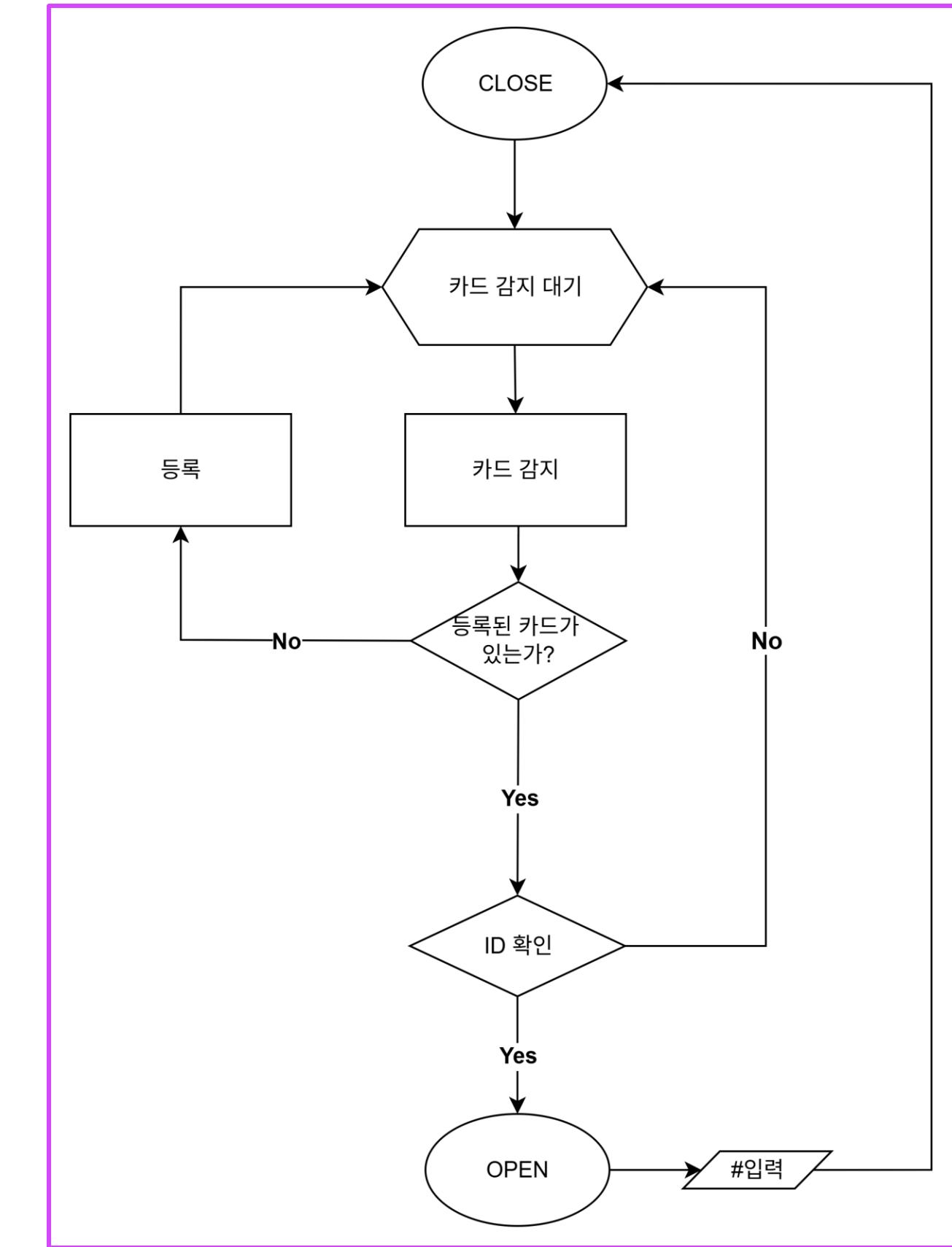


그림 4. RFID 모드 순서도

주요 코드 : RFID / 비밀번호 선택

```
// 모드 선택 대기
char key = keypad_get_key();
if (key == '1') {
    system_mode = 1; // RFID 모드
    lcd_clear();
    lcd_set_cursor(0, 0);
    if (!card_registered) {
        lcd_send_string("First: Register");
        lcd_set_cursor(1, 0);
        lcd_send_string("Present Card...");
    } else {
        lcd_send_string("RFID Mode");
        lcd_set_cursor(1, 0);
        lcd_send_string("Present Card...");
    }
    HAL_Delay(1000);
}
else if (key == '2') {
    system_mode = 2; // 비밀번호 모드
    lcd_clear();
    lcd_set_cursor(0, 0);
    lcd_send_string("Password Mode");
    HAL_Delay(500);
    lcd_clear();
    lcd_set_cursor(0, 0);
    lcd_send_string("Enter Password:");
    index_ = 0;
    i = 0;
    memset(entered, 0, sizeof(entered));
}
```

'1' 누르면 RFID

'2' 누르면
비밀번호

코드 1. RFID / 비밀번호 보안 방식 선택

주요 코드 : RFID 카드 읽기

```
uint8_t read_rfid_card(uint8_t* card_data)
{
    uint8_t status;
    uint8_t str[5];

    // 1단계: 카드 감지
    status = MFRC522_Request(PICC_REQIDL, str);
    if (status != MI_OK) {
        return 0; // 카드 없음
    }

    // 2단계: UID 읽기
    status = MFRC522_Anticoll(str);
    if (status != MI_OK) {
        return 0; // UID 읽기 실패
    }

    // UID 복사 (4바이트만)
    for (int i = 0; i < 4; i++) {
        card_data[i] = str[i];
    }

    return 1; // 성공
}
```

코드 2. 카드 읽기

```
uint8_t compare_rfid_card(uint8_t* card_data)
{
    for (int i = 0; i < 4; i++) {
        if (card_data[i] != authorized_card[i]) {
            return 0;
        }
    }
    return 1;
}
```

코드 3. 등록된 카드와 ID 비교

주요 코드 : 비밀번호 입력

```
void handle_key_input(char key, uint8_t* door_open, uint8_t* cursor_index)
{
    if (*door_open)
    {
        if (key == '#')
        {
            servo_lock();
            *door_open = 0;

            lcd_clear();
            lcd_set_cursor(0, 0);
            lcd_send_string("Door Locked");
            HAL_Delay(1000);

            lcd_clear();
            lcd_set_cursor(0, 0);
            lcd_send_string("Enter Password:");

            index_ = 0;
            *cursor_index = 0;
            memset(entered, 0, sizeof(entered));
        }
        return;
    }

    if (key == 'C')
    {
        lcd_clear();
        lcd_set_cursor(0, 0);
        lcd_send_string("Enter Password:");
        index_ = 0;
        *cursor_index = 0;
        memset(entered, 0, sizeof(entered));
    }
    else if (key >= '0' && key <= '9')
    {
        if (index_ < 4)
        {
            lcd_set_cursor(1, (*cursor_index));
            lcd_send_data('*');
            entered[index_++] = key;
            (*cursor_index)++;
            // 딜레이 제거 - 빠른 입력 가능
        }
    }
}
```

'#' 누르면 잠금

'C' 누르면 입력된
비밀번호 지우기

```
else if (key == 'D')
{
    if (index_ == 4)
    {
        if (strcmp(entered, password) == 0)
        {
            servo_unlock();
            *door_open = 1;

            lcd_clear();
            lcd_set_cursor(0, 0);
            lcd_send_string("Door Open");
            lcd_set_cursor(1, 0);
            lcd_send_string("Press # to Lock");
        }
        else
        {
            lcd_set_cursor(0, 0);
            lcd_send_string("Access Denied");
            lcd_set_cursor(1, 0);
            lcd_send_string("");
            HAL_Delay(1000);

            lcd_clear();
            lcd_set_cursor(0, 0);
            lcd_send_string("Enter Password:");
        }
    }

    index_ = 0;
    *cursor_index = 0;
    memset(entered, 0, sizeof(entered));
}

else if (key == 'A')
{
    changePassword();

    lcd_clear();
    lcd_set_cursor(0, 0);
    lcd_send_string("Enter Password:");

    index_ = 0;
    *cursor_index = 0;
    memset(entered, 0, sizeof(entered));
}
```

'D' 누르면
비밀번호
확인

'A' 누르면
비밀번호
변경

주요 코드 : 비밀번호 변경

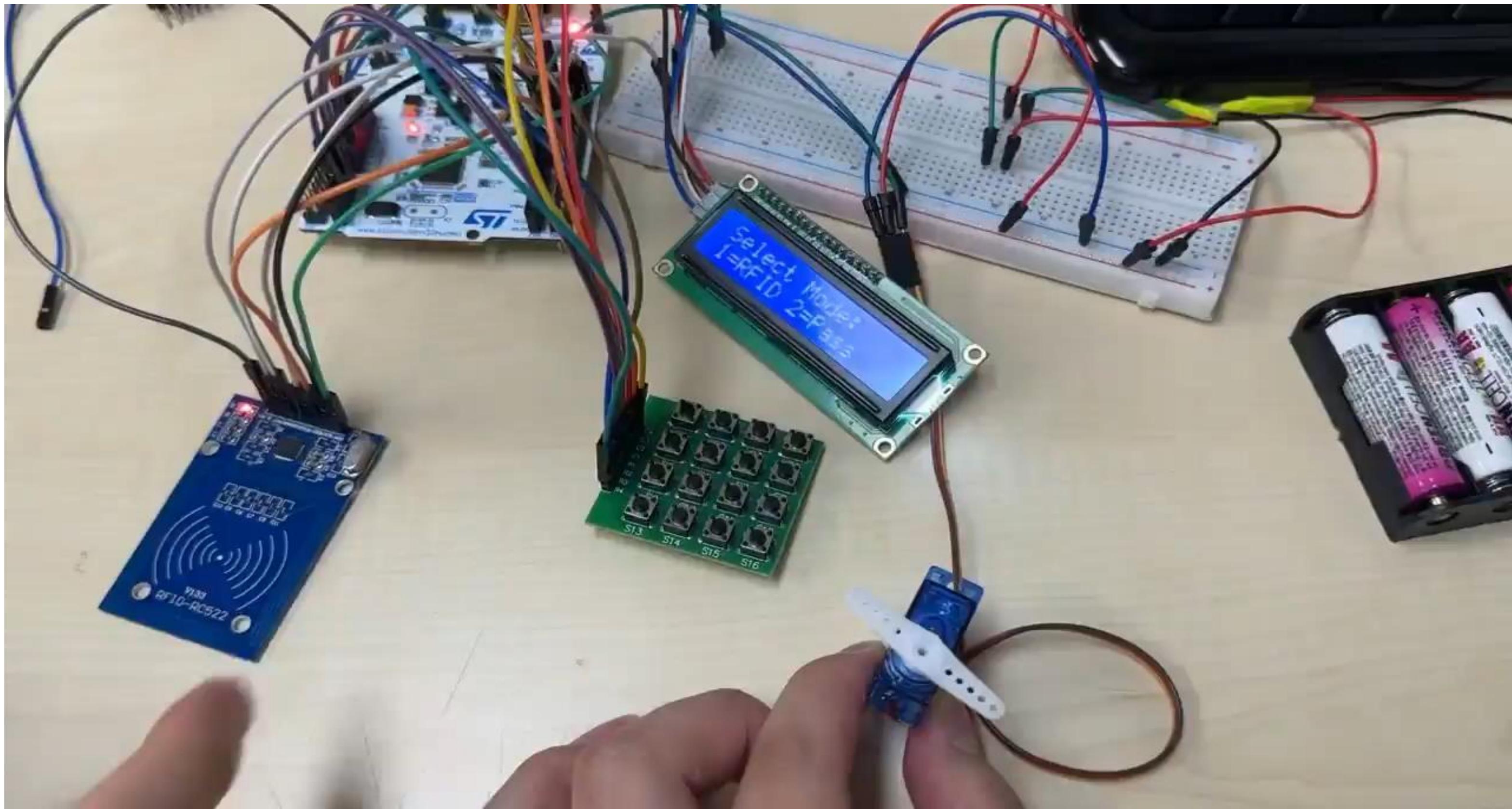
```
int changePassword(void){  
    char newPass[5];  
  
    lcd_clear();  
    lcd_set_cursor(0,0);  
    lcd_send_string("Change Password?");  
    lcd_set_cursor(1,0);  
    lcd_send_string("A=Yes, B=No.");  
  
    char choice = 0;  
    while(choice != 'A' && choice != 'B') {  
        choice = keypad_get_key();  
        HAL_Delay(10);  
    }  
  
    if(choice == 'B'){  
        lcd_clear();  
        lcd_send_string("Canceled");  
        HAL_Delay(1000);  
        return 0;  
    }  
  
    lcd_clear();  
    lcd_set_cursor(0,0);  
    lcd_send_string("New Password:");  
  
    for(int i = 0; i < 4; i++){  
        char k = 0;  
        while((k = keypad_get_key()) == 0){  
            HAL_Delay(10); // 짧은 딜레이만 유지 (키패드 안정성을 위해)  
        }  
  
        if(k >= '0' && k <= '9'){  
            newPass[i] = k;  
            lcd_set_cursor(1,i);  
            lcd_send_data('*');  
        } else {  
            i--;  
            continue;  
        }  
    }  
}
```

'A' 누르면
변경 실행

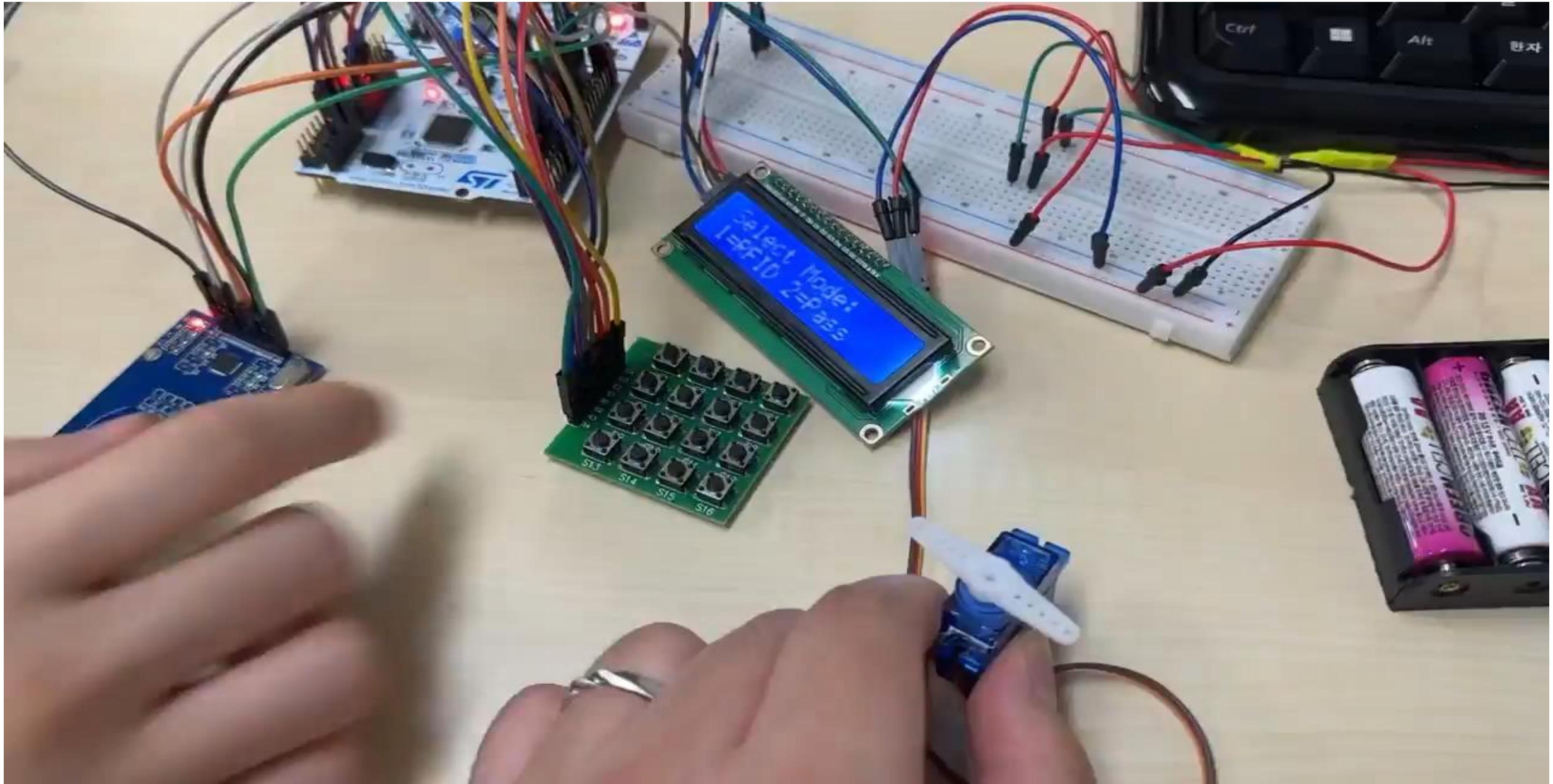
'B' 누르면
변경 취소

```
newPass[4] = '\0';  
if(strcmp(newPass, password) != 0) {  
    for(int i = 0; i < 4; i++){  
        password[i] = newPass[i];  
    }  
  
    lcd_clear();  
    lcd_set_cursor(0,0);  
    lcd_send_string("Password Updated");  
    HAL_Delay(1500);  
    return 1;  
} else {  
    lcd_clear();  
    lcd_set_cursor(0,0);  
    lcd_send_string("Same Password!");  
    HAL_Delay(1500);  
    return 0;  
}
```

RFID 모드 영상



Password 모드 영상



향후 계획

1. 보안 강화

- ID값 및 비밀번호를 Flash Memory에
저장하여 전원이 OFF시에도 저장 가능

2. 메모리 절약

- RFID 인터럽트 방식으로 메모리 사용 줄임