

# 마이크로프로세서 II 텀프로젝트 보고서

## 프로젝트 명

< 고오급 엘리베이터 >

2018144014	서 동 현
------------	-------

# 목 차

## I. 제안 이유 및 필요성-----

## II. 개발 방법 개요

1. 최종 개발 목표 ----- 3
2. 전체 시스템 구성도 / 센서 및 액추에이터 포함한 H/W구성도
3. 동작 시나리오 제시(Flowchart 및 동작 개요 제시)

## III. 요소별 개발 내용 및 검증

1. 각 요소별 회로의 설계 / 구현 결과(회로도 및 동작 검증)
2. 각 요소의 동작을 위한 프로그램 설계 / 구현결과(code 및 동작검증)
3. 전체 동작 시나리오에 따른 구현

## IV. 전체 시나리오에 따른 개발 내용 및 검증

1. 동작 시나리오에 따른 동작 검증

## V. 팀프로젝트 수행에 대한 결론 및 기대효과

1. 초기 계획 대비 진행 내용 비교
2. 전체적인 구현에 대한 Overview
3. 프로젝트 진행중 어려웠던 내용과 어떻게 해결하였는지

## VI. 소요된 부품 List 및 수행된 일정 정리

# 1장. 제안 이유 및 필요성(산업적/학습 등에 대한 필요성)

## 1) 산업적 필요성

### - 범죄와 테러에 대한 사건 사고 방지

22.08.19 오피스텔 엘리베이터를 기다리던 와중 치한이 묻지마 폭행을 한 사건이 발생하였다. 이를 예방 및 대비하기 위해 아파트 현관을 인증된 사용자만 출입 할 수 있도록 한다.

관련 자료 : [\[단독\]"엘리베이터 기다리는데 뒤에서 돌려차기" 부산 귀갓길 행인 살인미수 | JTBC 뉴스](#)

### - 4차 산업혁명시대에 맞는 스마트 엘리베이터

오늘날 가정 및 산업에서 이용하는 엘리베이터 시스템은 엘리베이터가 처음 도입되었을 때 사용한 사용자가 버튼을 눌러야 구동하는 방식임. 4차 산업혁명이 도래됨에 따라 보다 스마트하고 사용자에게 편리하게, 기다리는 시간 없이 엘리베이터를 사용할 수 있어야 함.

관련자료 : 시간표 데이터를 이용한 엘리베이터 알고리즘 설계[한국정보통신학회 종합 학술대회 논문집]

22.05 | 122~124

## 2) 학습 필요성

### - 통신 시스템 이해 및 적용

마이크로프로세서2 에서 주로 학습한 통신 시스템(UART, SPI, TWI)을 활용하여 작품을 설계 및 구현한다.

### - 센서값에 따른 액추에이터 제어

마이크로프로세서1, 2를 융합하여 센서값을 통해 받아들이는 정보에 따라 특정 동작을 수행하도록 작품을 설계 및 제작한다.

## 2장. 개발 방법 개요

### 1) 최종 개발 목표

- 승인된 사용자만 건물 출입 할 수 있도록 보안 시스템을 1층 출입문에 구현한다. 승인여부는 RFID리더기 태그를 통해 승인된 사용자면 출입문이 자동으로 열리게 되며 엘리베이터를 자동으로 1층으로 호출 후, 카드 UID값에 따라 목적층수를 미리 눌러준다.

승인된 사용자가 아니라면 미승인된 카드라는 경고문구를 LCD 디스플레이에 출력한다.

- 출입문은 서보모터를 이용하여 게이트 open / close를 진행한다.

- 온도센서, 조도세서를 통해 외부환경에 따라 실내 온도 및 조명 환경을 제어한다.

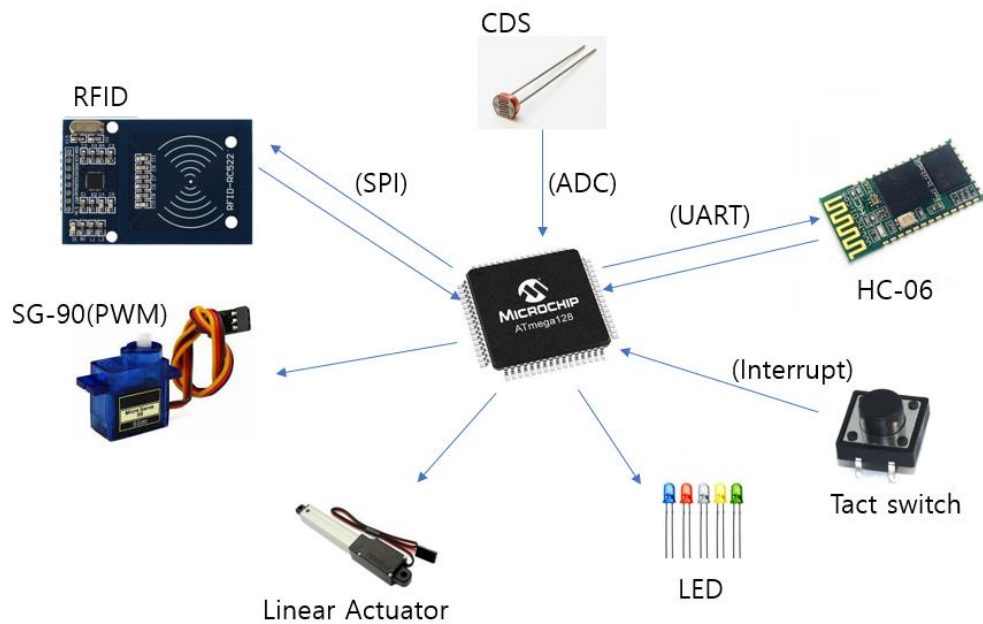
- 엘리베이터는 리니어 액추에이터를 이용하여 층 수에 맞게 구별한다.

- 부가적인 기능으로 낮과 밤에 따라서 조도의 값이 바뀌기 때문에 저녁이 되면 조도 값에 따라서 자동으로 복도 LED가 켜진다.

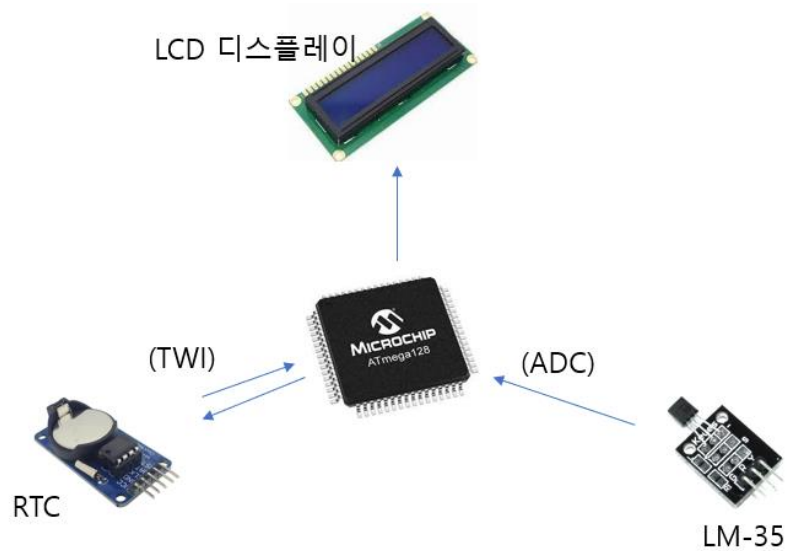
- 예외상황을 대비해 스위치로 엘리베이터를 1층으로 부를 수 있다.

## 2) 전체 시스템 구성도 / 센서 및 액추에이터 포함한 H/W 구성도

### ① H/W구성도

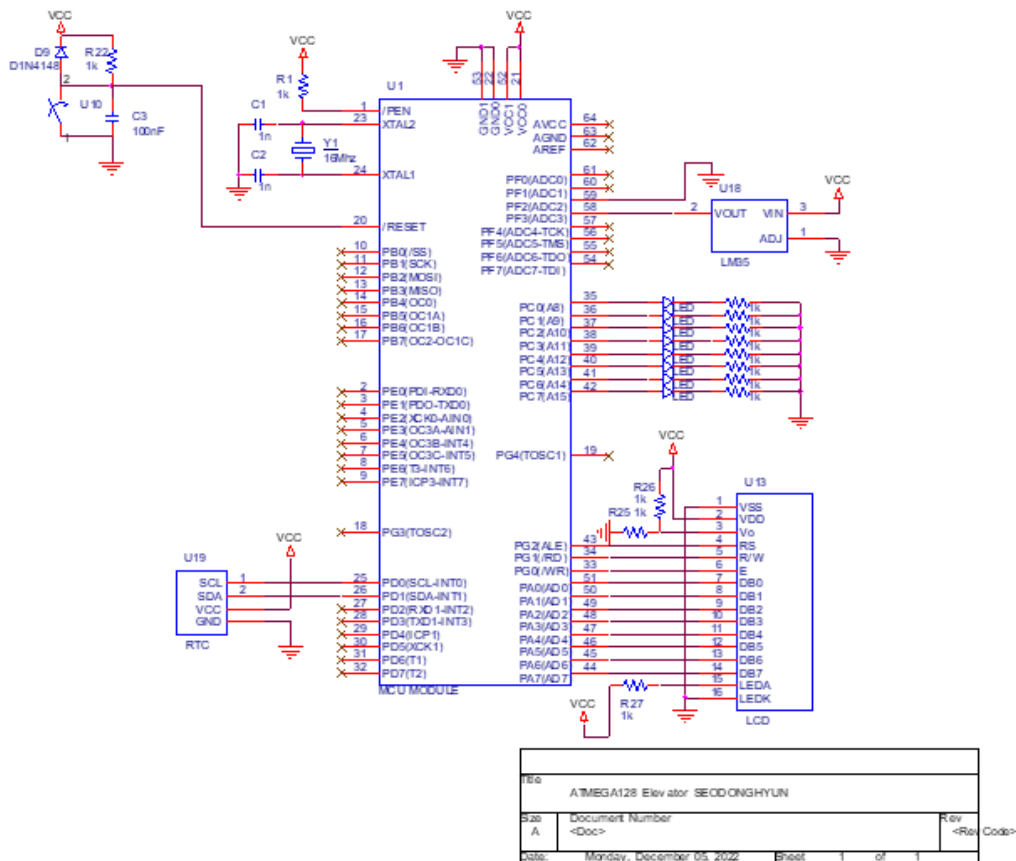
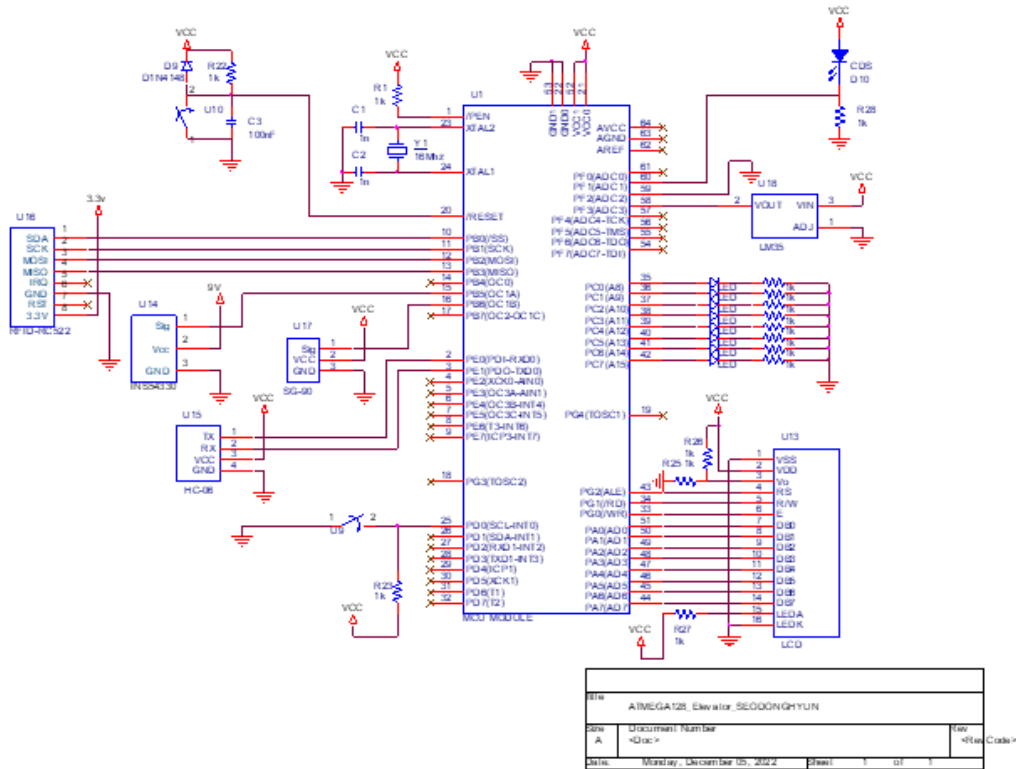


[동작부]



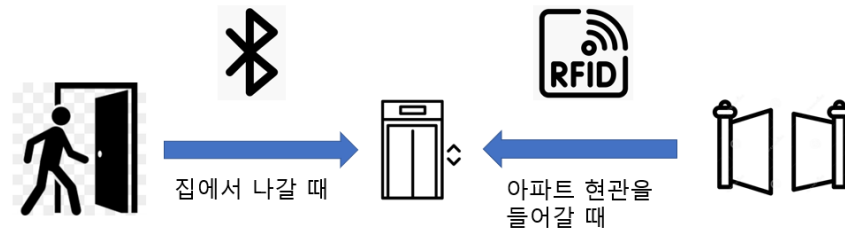
[인터페이스부]

## ② 회로도



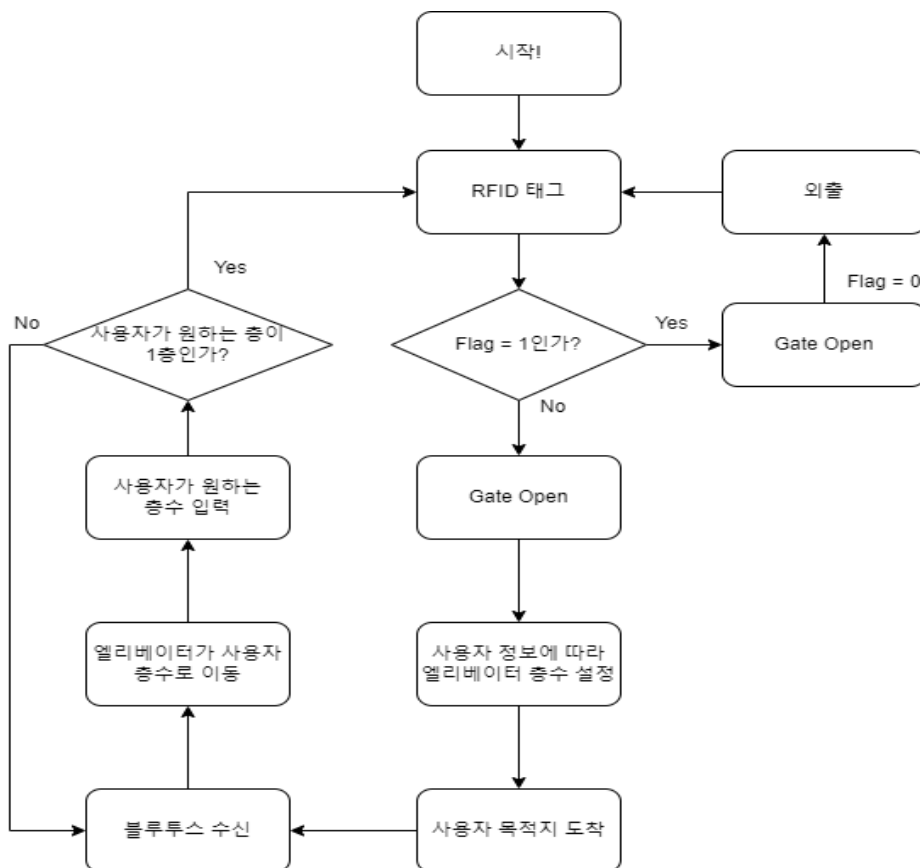
### 3) 동작 시나리오 제시 (Flowchart 및 동작 개요 제시)

#### ① 시스템 구성



사용자가 집에서 외출 할 때 휴대폰 블루투스를 통해 엘리베이터를 호출 할 수 있도록 한다. 반대로 실외에서 내부로 출입할 경우는 RFID카드 태그를 통해 인증된 사용자만 출입을 허용할 수 있도록 게이트를 Open한다.

#### ② 흐름도

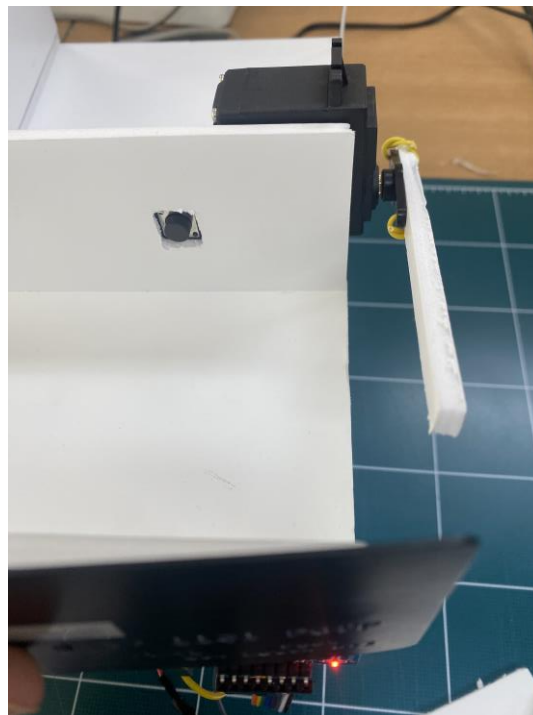


#### 4) 예외 시나리오 제시 및 대응방안 (Flowchart 및 동작 개요 제시)

##### ① RFID태그시 인증된 사용자가 아닐 경우.

현재 프로젝트에서 uidByte[0]의 값이 0x8C일 경우 2층 거주자, 0x23일 경우 3층 거주자로 등록을 하였다. 등록된 카드 정보에 따라 자동으로 엘리베이터가 사용자의 거주층수로 이동한다. uidByte[0]의 값이 0x8C, 0x23이 아닌 다른 값을 수신하면 경고음을 울린다. 이에 따른 예외처리는 다음과 같이 구현하였다.

```
}else{           // 인증된 카드가 아닐 경우 경고음 발생
    for(int k=0;k<500;k++){
        PORTG |= (1<<PG4);
        _delay_ms(1);
        PORTG &= 0xef;
        _delay_ms(1);
    }
}
```



등록되지 않는 카드가 태그 되면  
현관문(서보모터)가 열리지 않고 부저가 울리는 모습



② 엘리베이터를 수동으로 동작 시

예시 상황으로 RFID를 집에 두고 왔을 때를 대비해 1층에 외부인터럽트를 사용한 스위치를 만들어 엘리베이터를 사용 할 수 있도록 한다.

```
//택트스위치 (엘리베이터 1층)
ISR(INT5_vect)
{
    if(!(PINE &(1<<PE5)))
    {
        OCR1A = 250; // 1층
    }
}
```

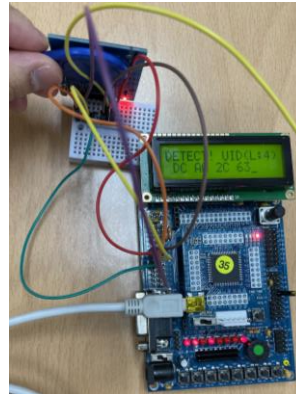
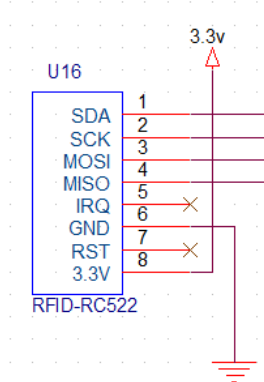


외부 인터럽트로 연결된 스위치를 누르면  
엘리베이터가 1층에 도달하는 모습

### 3장. 요소별 개발 내용 및 검증

#### 1) 각 요소별 회로의 설계 / 구현 결과(회로도 및 동작 검증)

##### ① RFID(SPI)



##### 8.1.2 Serial Peripheral Interface

A serial peripheral interface (SPI compatible) is supported to enable high-speed communication to the host. The interface can handle data speeds up to 10 Mbit/s. When communicating with a host, the MFRC522 acts as a slave, receiving data from the external host for register settings, sending and receiving data relevant for RF interface communication.

An interface compatible with SPI enables high-speed serial communication between the MFRC522 and a microcontroller. The implemented interface is in accordance with the SPI standard.

The timing specification is given in [Section 14.1 on page 78](#).

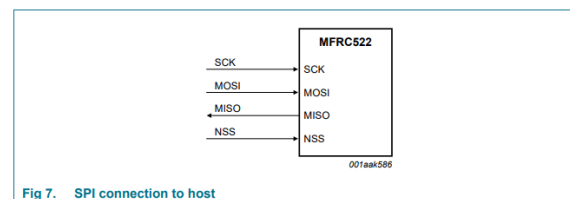
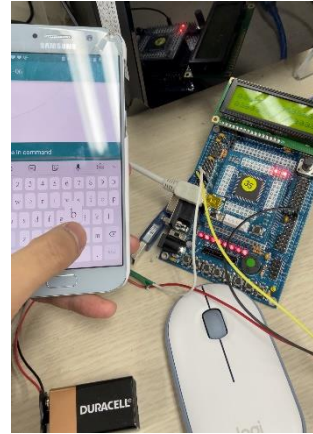
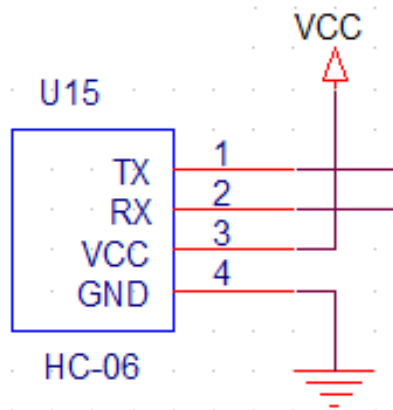


Fig 7. SPI connection to host

##### RC-522 Datasheet

RFID는 SPI통신으로 하나의 마스터와 다수의 슬레이브가 존재할 수 있으며 "SDA, SCL, MOSI, MISO"총 4개의 선을 이용하여 데이터의 송수신이 가능하다. SDA는 각 슬레이브에 대한 선택 신호로 마스터에 의해 선택되며 SCL은 마스터와 슬레이브 간의 동기 클럭 신호로 마스터에서 발생한다. MOSI는 마스터로부터 슬레이브로 전송되는 데이터 선이며, MISO는 슬레이브로부터 마스터로 전송되는 데이터 선이다. 송수신되는 데이터를 통해 인증된 사용자인지 아닌지 구별한다.

## ② HC-06(UART)



블루투스는 UART통신으로 Tx, Rx총 2개의 신호선을 사용하며 동시에 송수신이 가능한 전이중 통신방식이다. 현재 프로젝트에서 비동기식 통신으로 진행하였으며 Baudrate는 9,600으로 설정하였다. 스마트폰에서 신호를 보내면 Atmega128은 Rx인터럽트로 신호를 수신한다. 블루투스 통신을 통해 엘리베이터를 원하는 층으로 호출한다.



Putty를 사용해 블루투스 통신을 해서 엘리베이터를 사용자가 원하는 층으로 부를 수 있도록 제작.

### ③ RTC(I2C)

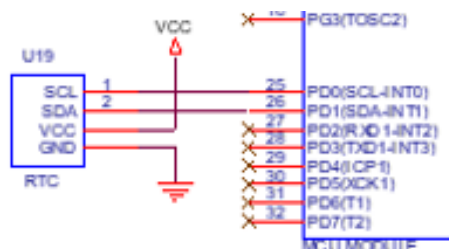
#### PIN DESCRIPTION

PIN	NAME	FUNCTION
1	X1	Connections for Standard 32.768kHz Quartz Crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance ( $C_L$ ) of 12.5pF. X1 is the input to the oscillator and can optionally be connected to an external 32.768kHz oscillator. The output of the internal oscillator, X2, is floated if an external oscillator is connected to X1.
2	X2	<b>Note:</b> For more information on crystal selection and crystal layout considerations, refer to <i>Application Note 58: Crystal Considerations with Dallas Real-Time Clocks</i> .
3	V <sub>BAT</sub>	Backup Supply Input for Any Standard 3V Lithium Cell or Other Energy Source. Battery voltage must be held between the minimum and maximum limits for proper operation. Diodes in series between the battery and the V <sub>BAT</sub> pin may prevent proper operation. If a backup supply is not required, V <sub>BAT</sub> must be grounded. The nominal power-fail trip point ( $V_{PF}$ ) voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as $1.25 \times V_{BAT}$ nominal. A lithium battery with 48mAh or greater will back up the DS1307 for more than 10 years in the absence of power at +25°C.  UL recognized to ensure against reverse charging current when used with a lithium battery. Go to: <a href="http://www.maxim-ic.com/qa/info/ul/">www.maxim-ic.com/qa/info/ul/</a> .
4	GND	Ground
5	SDA	Serial Data Input/Output. SDA is the data input/output for the I <sup>2</sup> C serial interface. The SDA pin is open drain and requires an external pullup resistor. The pullup voltage can be up to 5.5V regardless of the voltage on V <sub>CC</sub> .
6	SCL	Serial Clock Input. SCL is the clock input for the I <sup>2</sup> C interface and is used to synchronize data movement on the serial interface. The pullup voltage can be up to 5.5V regardless of the voltage on V <sub>CC</sub> .
7	SQW/OUT	Square Wave/Output Driver. When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square-wave frequencies (1Hz, 4kHz, 8kHz, 32kHz). The SQW/OUT pin is open drain and requires an external pullup resistor. SQW/OUT operates with either V <sub>CC</sub> or V <sub>BAT</sub> applied. The pullup voltage can be up to 5.5V regardless of the voltage on V <sub>CC</sub> . If not used, this pin can be left floating.
8	V <sub>CC</sub>	Primary Power Supply. When voltage is applied within normal limits, the device is fully accessible and data can be written and read. When a backup supply is connected to the device and V <sub>CC</sub> is below $V_{TP}$ , read and writes are inhibited. However, the timekeeping function continues unaffected by the lower input voltage.

RTC모듈은 I2C(TWI)통신으로 SCL(직렬클럭), SDA(직렬데이터)의 2개의 양방향 신호선을 이용한 통신방식이다. 1개의 데이터선을 이용하는 반이중 통신방식이며 SCL동기 클럭을 사용하므로 동기식 통신방식이다.



엘리베이터 맞은 편에  
시간을 확인할 수 있도록  
RTC값을 변수로 받아 LCD에 표현한 모습

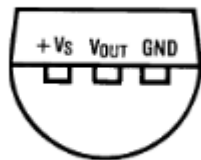


아트메가와 연결한 회로도 모습

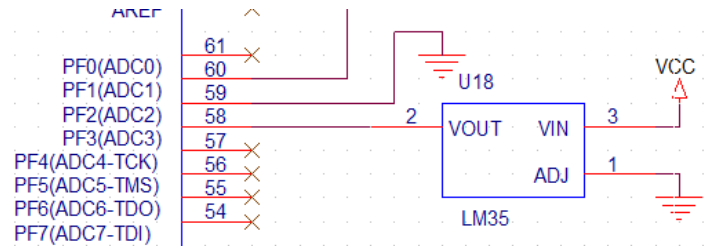
#### ④ LM-35 & CdS(ADC)

##### - LM35

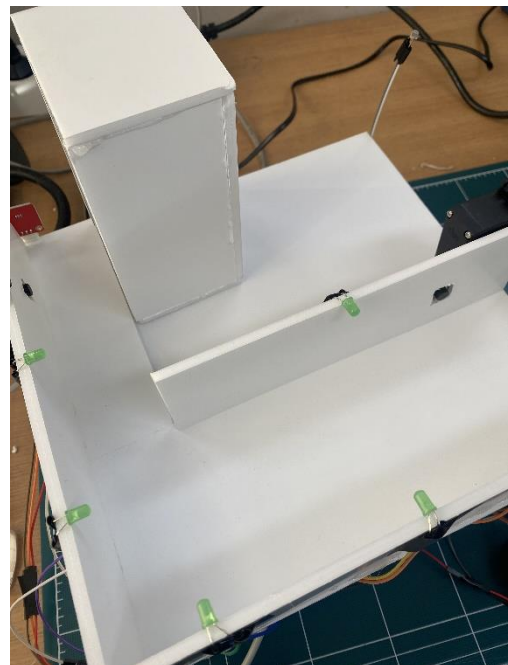
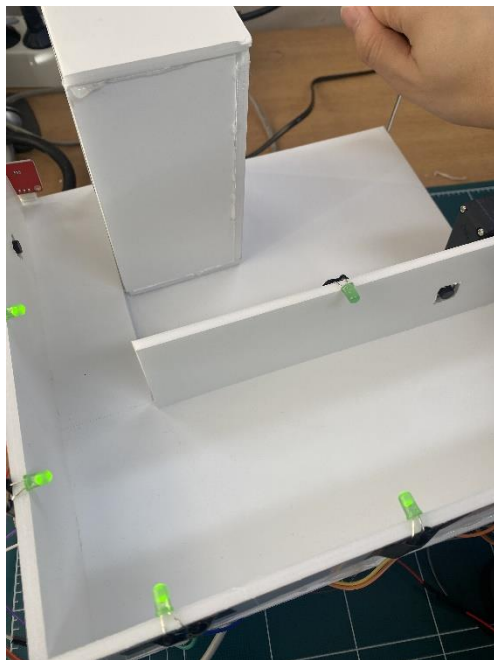
TO-92  
Plastic Package



BOTTOM VIEW  
DS005516-2

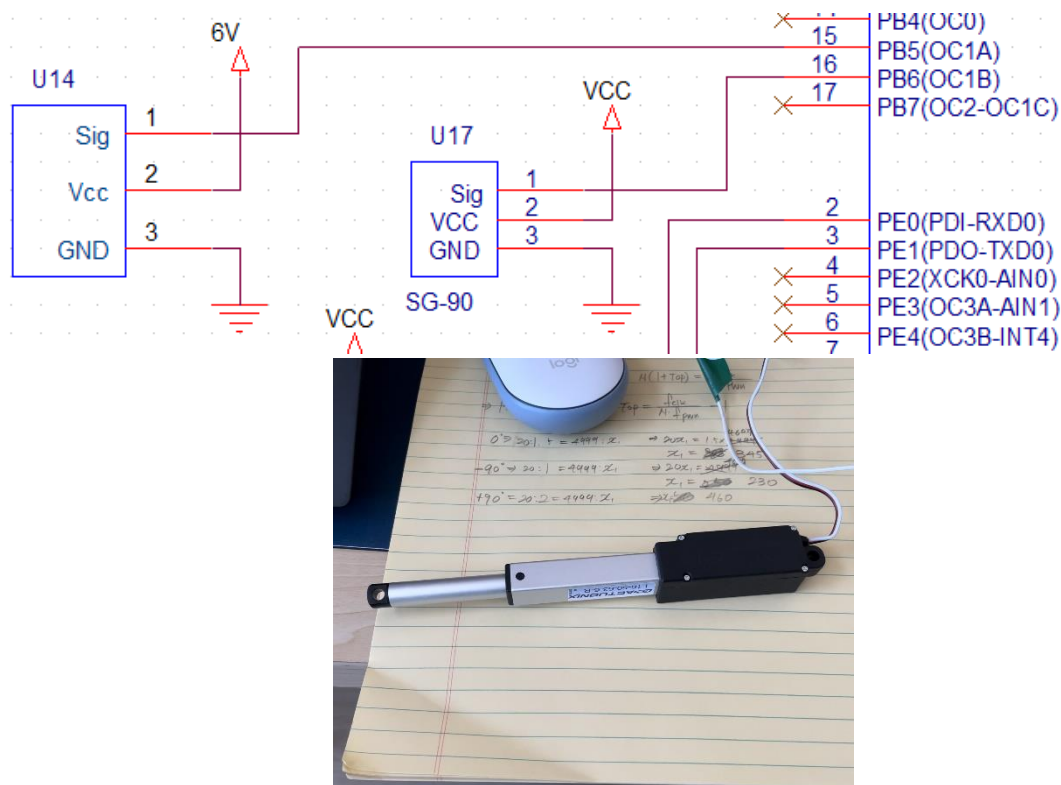


LM35는 출력전압이 섭씨온도에 선형적으로 비례하는 특징을 가는 정밀 IC 온도 센서이다. 동작전압은 datasheet상 4V-30V이기 때문에 시스템 전원 5V를 인가하였다. 현 프로젝트에서 ADC2, ADC3핀을 이용하여 차동입력채널과 10배 증폭모드를 이용하였다. A/D변환기의 분해능은 10비트이므로 A/D변환 출력은 1스텝당 4.88mV를 나타낸다. 위 회로도와 같이 진행하면 ADC2를 GND와 접지하여 ADC3의 신호를 GND기준으로 10배 증폭시킬 수 있다.



조도 센서의 값에 따라서 LED가 켜진 모습과 안켜진 모습을 확인할 수 있다.

## ⑤ Linear actuator / Servo moter



본 프로젝트에서 사용한 리니어 액추에이터는 엘리베이터의 높낮이를 조절하기 위해 사용하였다. 제품명은 Actuonix사의 L16R모델이며 1개의 신호선을 통해 PWM을 조절하여 스트로크 길이를 조절하는 제품이다. 해당 제품의 사용법은 다음과 같다.

### Option R – RC Linear Servo

WIRING: (see last page for pin numbering)

- 1 - White – **RC input signal** (RC-servo compatible)
- 2 - Red – **Power** (+6 VDC)
- 3 - Black – **Ground**

*Note: Reversing the polarity of pins 2 and 3 may permanently damage the actuator*

-R actuators are ideally suited to use in robotics and radio control models. The -R actuators or 'linear servos' are a direct replacement for regular radio controlled hobby servos. The desired actuator position is input to the actuator on lead 1 as a positive 5 Volt pulse width signal. A 1.0 ms pulse commands the controller to fully retract the actuator, and a 2.0 ms pulse signals it to fully extend. If the motion of the actuator, or of other servos in your system, seems erratic, place a 1-4Ω resistor in series with the actuator's red V+ lead wire.

L16 -R Linear Servos are the only 6 volt models in the L16 range because they are designed to work with typical RC receivers and battery packs. Consequently they also are compatible with Arduino control boards, VEX Microcontrollers and many other similar boards designed for robotics.

데이터시트를 확인 한 결과

PWM주기를 1ms로 설정하면

스트로크 길이를 최소로 하며

2ms로 진행하면 최대길이

(50mm)로 늘어나게 된다.

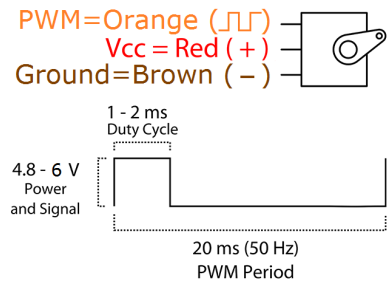
해당 제품의 동작 전압은 최소

6V부터 구동을 한다.

## - Servo moter(MG-90s)

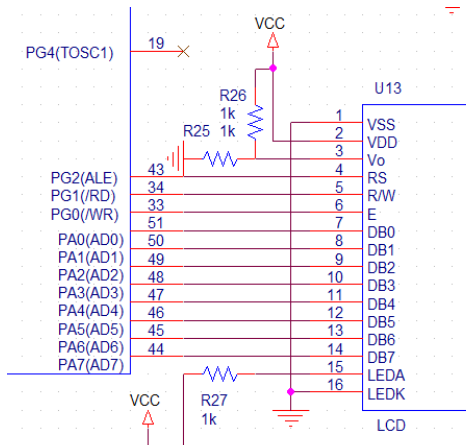
### Specifications

- Weight: 13.4 g
- Dimension: 22.5 x 12 x 35.5 mm approx.
- Stall torque: 1.8 kgf·cm (4.8V ), 2.2 kgf·cm (6 V)
- Operating speed: 0.1 s/60 degree (4.8 V), 0.08 s/60 degree (6 V)
- Operating voltage: 4.8 V - 6.0 V
- Dead band width: 5  $\mu$ s



서보모터또한 앞서 설명한 Linear actuator과 동일한 PWM주기를 갖는다. 구동전압은 4.8V-6V로 동작하기 때문에 역시 외부전원을 연결하여 구동을 시켰다.

## ⑥ LCD디스플레이



### 6. Interface pin description

Pin no.	Symbol	External connection	Function
1	Vss	Power supply	Signal ground for LCM
2	VDD		Power supply for logic for LCM
3	Vo		Contrast adjust
4	RS	MPU	Register select signal
5	R/W	MPU	Read/write select signal
6	E	MPU	Operation (data read/write) enable signal
7~10	DB0~DB3	MPU	Four low order bi-directional three-state data bus lines. Used for data transfer between the MPU and the LCM. These four are not used during 4-bit operation.
11~14	DB4~DB7	MPU	Four high order bi-directional three-state data bus lines. Used for data transfer between the MPU
15	LED+	LED BKL power supply	Power supply for BKL
16	LED-		Power supply for BKL

LCD디스플레이는 세부적으로 많은 표현을 해야하기 때문에 다른 소자보다 많은 핀개수를 사용한다. lcd.h을 이용하여 필요한 부분은 함수 호출을 통해 관리한다.

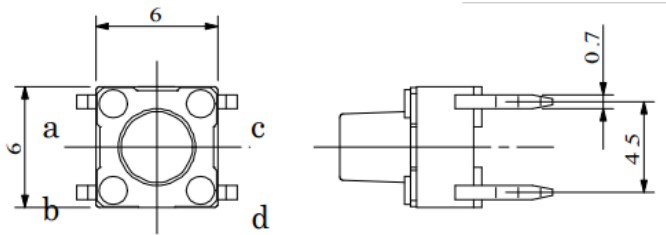


엘리베이터 맞은 편에 LCD를 부착해

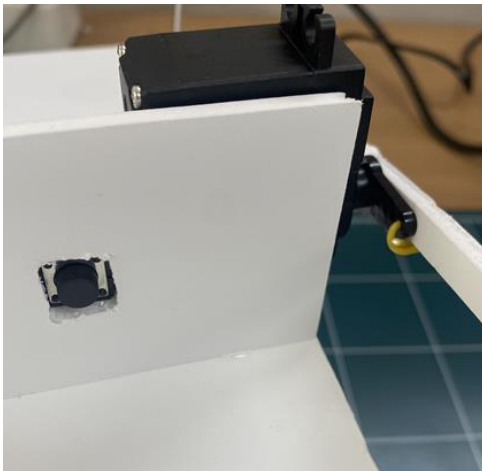
사용자가 정보를 얻을 수 있도록 한다.



## ⑦ Tact Switch



택트스위치는 외부 인터럽트를 이용하여 프로그램의 규약을 받지 않고 언제 어디서든 스위치를 누르게 되면 특정동작을 즉시 수행하는 목적으로 사용한다.



외부인터럽트 4번과 연결한  
서보모터 동작 스위치



외부인터럽트 5번과 연결한  
엘리베이터 동작 스위치



## 2) 각 요소의 동작을 위한 프로그램 설계 / 구현 결과(Code 및 동작 검증)

### <블루투스>

```
//블루투스 수신
ISR(USART0_RX_vect){
    ch = UDR0;
}
//블루투스
ISR(USART0_TX_vect){
    UDR0 = ch;
}
```

블루투스 동작을 위한  
인터럽트 서비스 루틴

```
//블루투스 초기화
void Init_USART0(void){
    UCSR0A = 0x00;
    UCSR0B = (1<<RXCIE0)|(1<<TXCIE0)|(1<<RXEN0) | (1<<TXEN0);
    UCSR0C = (1<<UCSZ01) | (1<<UCSZ00);
    UCSR0C &= ~(1<<UMSEL0);

    UBRR0H = 0;
    UBRR0L = 95;
}
```

블루투스 동작을 위한 초기화 함수

```

//블루투스 통신 시작//
if(ch == '1'){
    LCD_Pos(0,0);
    LCD_Str("1st Floor");
    OCR1A=250;    //-90도 1층
    ch = '0';
}
else if(ch == '2'){
    LCD_Pos(0,0);
    LCD_Str("2nd Floor");
    OCR1A=330;    // 2층
    ch = '0';
}
else if(ch == '3'){
    LCD_Pos(0,0);
    LCD_Str("3rd Floor");
    OCR1A = 400;    // 3층
    ch = '0';
}
else{
    ch = '0';
    LCD_Pos(0,0);
    LCD_Str("");
}
//블루투스 통신 끝//

```

main문 안에 있는 블루투스  
통신 동작 코드

블루투스 수신 값 UDR0를  
변수 ch에 저장해서 ch를  
통해서 엘리베이터를  
사용자가 원하는 층 수로  
올 수 있도록 변수를 저장.

OCR1A값에 따라서  
리니어 액추에이터의  
출력 파형(duty cycle)이  
결정된다.

# 전체적인 코드는 파일 첨부 하도록 하겠습니다 #

## <RFID>

```
//RFID 시작//
if(res == 2){
    if(uid.uidByte[0] == 0x8C) //2층 카드 태그
    {
        OCR1A = 330;
        LCD_Pos(0,0);
        LCD_Str("2st floor      ");
        OCR1B = 190; // GATE Open
        _delay_ms(3000);
        OCR1B = 250; // GATE Close
        _delay_ms(1000);
    }
    else if(uid.uidByte[0] == 0x23) //3층 카드 태그
    {
        OCR1A = 400;
        LCD_Pos(0,0);
        LCD_Str("3rd floor      ");
        OCR1B = 190; // GATE Open
        _delay_ms(3000);
        OCR1B = 250; // GATE Close
        _delay_ms(1000);
    }
}
else{
    //부저코드
    buz_sound(SOL);
}
//RFID 끝 //
```

## RFID의 동작 코드

UID 값으로 사용자를 인식해  
사용자의 정보가 저장된  
층으로 엘리베이터를 보낸다

또한 OCR1B의 값이 변경되어  
서보모터를 통해 현관문이  
열리고 3초 뒤에 서보모터가  
다시 닫히는 구조이다.

인증된 사용자가 아닐 경우  
부저가 울리면서 경고음이  
발생하도록 저장되어있다.

# 전체적인 코드는 파일 첨부 하도록 하겠습니다 #

## <RTC>

```
tinyRTC_init();

tinyRTC_setup(00, 30, 16, 5, 22, 12, 22);    // 초, 분, 시, 요일, 일, 월, 년
tinyRTC_set_date();
LCD_Init();

while (1)
{
    tinyRTC_read_date();

//시간 읽는 함수
void tinyRTC_read_date(void)
{
    uint8_t data[7];
    memset(data, 0, sizeof(data));
    while(i2c_readReg((SLAVE_ADDR << 1), 0x00, data, 7));

    //sprintf(Message,"%d.%d.%d.(%s) %d:%d:%d\r\n", bcd_to_decimal(data[6]),
                                //bcd_to_decimal(data[5]),
                                //bcd_to_decimal(data[4]),
                                //change_day(bcd_to_decimal(data[3])),
                                //bcd_to_decimal(data[2]),
                                //bcd_to_decimal(data[1]),
                                //bcd_to_decimal(data[0]));
    sprintf(Message,"%s%d:%d:%d",change_day(bcd_to_decimal(data[3])),
                                bcd_to_decimal(data[2]),
                                bcd_to_decimal(data[1]),
                                bcd_to_decimal(data[0]));

    LCD_Pos(0,0);
    LCD_Str(Message);
}
```

순서대로 현재 시간을 표시해주기 위해 read 함수를 만들어 저장했다.

## <ADC>

```
void ADC_Init(void){  
    ADCSRA = 0x00;  
    ADMUX=((ADC_AVCC_TYPE) | (0<<ADLAR) | (0<<MUX0));  
    ADCSRA = (1<<ADEN) | (1<<ADFR) | (3<<ADPS0);  
}
```

adc초기화 함수

```
unsigned int Read_ADC_Data_Diff(unsigned char adc_mux){  
    unsigned int ADC_Data = 0;  
  
    if(adc_mux < 8)  
        return 0xFFFF;  
  
    ADMUX &= ~(0x1F);  
    ADMUX |= (adc_mux & 0x1F);  
  
    ADCSRA |= (1<<ADSC);  
  
    while(!(ADCSRA & (1<<ADIF)));  
  
    ADC_Data = ADCL;  
    ADC_Data |= ADCH << 8;  
  
    return ADC_Data;  
}
```

차동 중단 방식을

이용하기 위한 함수

```
adcRaw = Read_ADC_Data_Diff(0b1101);  
adcmilliVoltage = (((float)adcRaw * 5000)/512)/10);  
Celsius = adcmilliVoltage / 10;  
  
sprintf(Message,"Temp(deg):%2.2f",Celsius);  
LCD_Pos(1,0);  
LCD_Str(Message);  
_delay_ms(100);
```

adc값을 받고 나서 섭씨 온도로 변환하기 위한 계산 식

```

//조도센서
unsigned int single_Read_ADC_Data_Diff(unsigned char adc_mux)
{
    unsigned int ADC_Data = 0;

    //if(adc_mux < 8)
    //return 0xffff; //양극 신호가 아닌 단극 mux 입력시 종료

    //ad 변환 채널 설정
    ADMUX &= ~(0x1f);
    ADMUX |= (adc_mux & 0x1f);

    ADCSRA |= (1<<ADSC); //ad변환 시작
    while(!(ADCSRA & (1<<ADIF))); //ad변환 종료 대기;

    ADC_Data = ADCL;
    ADC_Data |= ADCH << 8 ;

    return ADC_Data;
}
void ADC_Init(void)//프리러닝 모드로 초기화
{
    ADCSRA = 0x00; //adc 설정을 위한 비활성화
    ADMUX = ADC_AVCC_TYPE | (0<<ADLAR) | (0<<MUX0);
    //ADMUX = ADC_AVCC_TYPE | (0<<ADLAR);
    //REFS = 1, ADLAR = 0, MUX = 0 (ADC0 선택)
    ADCSRA = (1<<ADEN) | (0<<ADFR) | (3<<ADPS0);
    //1<<ADEN : AD변환 허가 , 1<<ADFR : 프리 러닝 모드 허가
    // 3<<ADPS0 : AC변환 분주비 설정 - 8분주비
    //ADSC비트는 실제 활용에서 ADC를 읽을 때 설정하여 변환한 후
    //데이터를 읽기 위해 초기화 과정에서는 설정하지 않음
}

```

단일 입력 방식으로 사용하기 위한 함수 및 초기화 함수

## <인터럽트>

```
//택트스위치 (서보모터)
ISR(INT4_vect)
{
    if(!(PINE &(1<<PE4)))
    {
        OCR1B = 190;           // GATE Open
        _delay_ms(3000);
        OCR1B = 250;           // GATE Close
        _delay_ms(1000);
    }
}
//택트스위치 (엘리베이터 1층)
ISR(INT5_vect)
{
    if(!(PINE &(1<<PE5)))
    {
        OCR1A = 250; // 1층
    }
}
```

인터럽트 서비스 루틴으로  
외부인터럽트 4번을 사용해  
서보모터를 제어한다.

인터럽트 서비스 루틴으로  
외부인터럽트 5번을 사용해  
리니어 액츄에이터를 제어  
한다

```
//외부인터럽트 (스위치)
void itrp_init(void)
{
    EIMSK = 0x30; //외부인터럽트 4,5번
    SREG |= 0x80;
    DDRE = 0x02;
}
```

EIMSK에서 외부인터럽트 4,5번을  
활성화 시키기 위한 비트 설정

## 4장. 전체 시나리오에 따른 개발 내용 및 검증

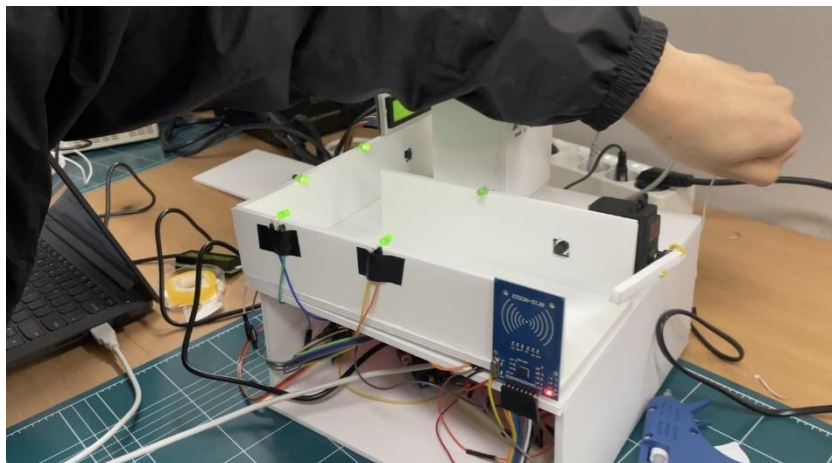
#전체 동작 영상은 맨 마지막 페이지에 링크로 첨부하였습니다#

### 1) LCD디스플레이에 현재시간 및 온도 표시



RTC모듈을 통해 시스템 시간이 계속해서 흐르며 온도센서(LM35)를 통해 수신된 현재 온도를 LCD디스플레이에 출력한다. 시간은 1초에 한번씩 update가 이루어지며 “(요일)시:분:초”의 표기법으로 디스플레이에 출력한다. 온도는 센서의 값이 변화될 때 마다 update가 이루어 진다.

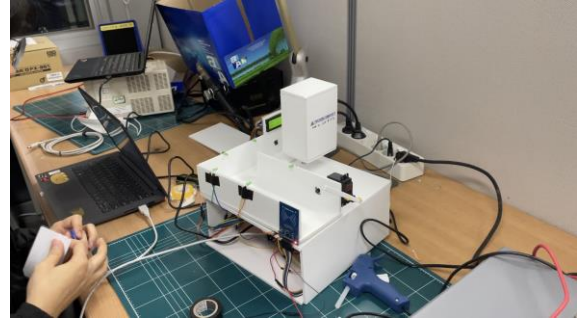
### 2) 외부조도에 따른 실내조명 제어



조도센서를 통해서 외부 조도가 낮을 경우(밤) 실내 조명(LED)을 자동으로 키고 외부 조도가 높으면 실내조명을 끈다. LED는 복도에 총 5개를 배치하였다.

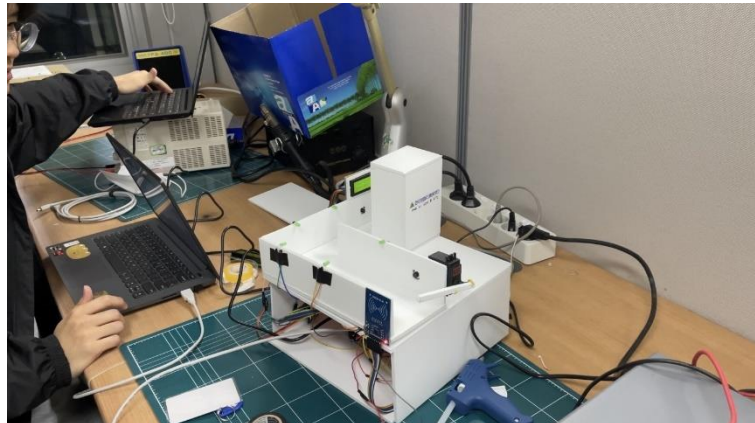


### 3) 카드 태그에 따라 게이트open, 엘리베이터 높이조절



RFID리더기에 인증된 카드가 태그되면 서보모터를 통해 게이트가 open되며 엘리베이터가 자동으로 사용자의 거주 층으로 높이를 조절한다. 카드의 UID의 첫번째 인덱스 값이 0x8C일 경우 엘리베이터가 2층으로 이동하며 첫번째 인덱스 값이 0x23이면 3층으로 이동한다. 인증되지 않은 사용자의 카드를 태그한다면 부저를 통해 경고음을 발생한다.

### 4) 블루투스 통신으로 엘리베이터 호출



HC-06블루투스 모듈을 이용하여 UART 시리얼 통신을 통해 데이터를 수신받는다. 사용자가 단말기(휴대폰, 노트북 등)에서 데이터를 전송하면 엘리베이터를 사용자의 거주 층 수로 호출 할 수 있다.

## 5) 외출시 스위치로 게이트 open



사용자가 실내에서 실외로 외출을 할 경우 게이트 옆에 있는 스위치 버튼을 눌러서 게이트를 open할 수 있다. 스위치는 언제든지 누를 수 있기 때문에 외부 인터럽트를 통해 시스템을 구현하였다.

## 5장. 텀프로젝트 수행에 대한 결론 및 기대효과

### - 초기 계획 대비 진행 내용 비교

처음 프로젝트를 계획했을 때 구상했던 내용의 가장 핵심은 RFID 카드 태그에 따른 모터 동작이 중요했고, 또 다른 포인트는 Bluetooth 동작을 만들어서 엘리베이터 사용에 있어 집으로 들어오는 사용자와 집에서 나가는 사용자를 구분하는 부분이었다. 집으로 들어갈 때에는 등록된 RFID 카드로만 현관문이 열리는 시스템이고 등록된 카드는 본인의 UID에 따라서 엘리베이터가 지정된 층으로 이동되는 동작까지 계획이었고 계획과 다른 차이점 없이 동작이 잘 구현되었음을 알 수 있다. 보안이라는 문제에 있어서 대중적으로 사용되는 엘리베이터를 현관문까지 연장시켜 RFID 태그 사용, 내부 switch를 사용해 보안성과 안정성을 높이는 기대효과를 얻을 수 있다.

부가적인 계획 중 서보모터를 현관문으로 사용하기로 했었는데 서보모터는 마이크로프로세서1 수업 때 사용했던 경험이 있어서 금방 동작 제어를 완료할 수 있었다. 계획에 없던 내용으로 내부에서 스위치를 사용해 서보 모터를 동작 제어하는 부분이 있었는데, 이 부분에서는 외부 인터럽트를

사용해야했다. 아트가 자체에 내장된 스위치는 PORTD에 있지만 아트가 자체를 하드웨어에 붙이기에는 하드웨어 프레임이 완성도가 떨어진다고 생각해서 아트의 데이터시트를 보면서 외부 인터럽트 4,5,6,7 번이 PORTE의 4,5,6,7번에 있는 것을 찾았고 PORTE의 상위 비트에서 출력신호를 끌어서 스위치에 연결하면 외부 인터럽트를 충분히 사용 할 수 있다고 생각했다. 이 계획은 순조롭게 잘 진행됐고 스위치는 슬라이드 스위치 보다는 탭 스위치가 동작 구현에 더 잘 어울린다고 생각이 들어서 탭 스위치로 PCB 기판에 납땀하고 연결해서 외부 인터럽트를 이용한 스위치는 해결되었다. 첫 계획때에는 하드웨어 프레임을 그렇게 세부적으로 생각하지 않았는데 직접 센서들을 제어하고 리니어 액추에이터를 동작시켜보니 생각보다 하드웨어 프레임 제작이 까다롭다는 것을 새롭게 알게 되었다.

각종 센서들을 이온 점퍼선과 전원선, RFID, Bluetooth 등등 한번에 여러가지를 사용하다보니 생각한 하드웨어 프레임 크기가 조금 협소하게 느껴졌다.

초기 계획에는 아트를 1개만 사용할 예정이었는데 여러가지 센서들을 사용하다보니 포트가 부족해서 아트를 2개 사용할 것으로 수정하였다.

포트 수가 여유로워져서 각종 센서들을 더 추가할 수 있었고 동작에 부가적인 기능을 넣을 수 있어서 하드웨어 프레임을 제작하고 직접 동작을 확인 할 때 완성도가 높다는 것을 확실히 느낄 수 있었다.

#### - 전체적인 구현에 대한 Overview

여러가지 구현을 하다보니 예상하지 못한 변수들도 많이 있었다고 생각한다.

서보모터와 리니어 액추에이터는 전류를 많이 잡아먹는 센서들이어서 아트에 직접 넣어주는 전원 이외에도 서보모터에는 배터리를 이용해서 외부 전원을 넣어주었고 리니어 액추에이터에는 파워 서플라이를 이용해서 외부 전원을 넣어주었다. 처음 외부전원을 리니어 액추에이터에만 넣어주었을 때에는 LCD 화면 점등 현상이 심하고 서보 모터 떨림 현상이 심했다.

이 문제를 해결하기 위해서 배터리를 추가했고 정말 없던 일처럼 점등 현상이

없어지고 서보 모터 떨림 현상도 해결됐다.

여러가지 센서와 모듈들을 사용하면서 데이터시트에 아주 조금 익숙해졌다고 생각하는데 RTC모듈, 온도 센서 모듈, 조도 센서 등 간단하지만 회로도 연결을 해주면서 공통 접지의 중요성도 알게되었다. 공통 접지를 하고서도 아트메가의 접지를 연결 안하는 경우도 있었고 예상하지 못한 변수 상황들이 많았다고 생각한다. 하드웨어 프레임을 제작할 때에는 여러가지 많이 깨닫게 되었는데 그 중 가장 크게 깨닫게 된 것은 하드웨어 구현할 때에도 크기, 수치 등 세부적인 내용까지도 계획하는 것이 중요하다는 것이 중요했다.

처음 계획할 때에는 큰 그림만 잡고 폼보드를 사용해 붙일 예정이었는데 크기를 어림잡아 하다보니 마지막에 점프선, 전원선, 센서, 모듈 등 여러가지를 한 번에 정리할 때가 되니 생각보다 하드웨어 프레임의 크기가 협소하다고 생각이 많이 들었다.

- 프로젝트 진행중 어려웠던 내용과 어떻게 해결하였는지.

가장 어려웠던 부분은 RFID 태그가 어려웠다. 예제 소스도 많이 없었고 아두이노를 사용할 때에는 헤더파일이 있기 때문에 쉽게 사용 할 수 있지만 AVR로 사용할 때에는 헤더파일을 직접 레지스터 비트 제어를 하는 방법으로 RFID 태그를 진행했다. 아두이노 헤더파일을 분석하고 rc522의 datasheet를 보면서 직접 레지스터를 찾았고 필요한 함수들만 조금씩 가져와 사용해서 문제를 해결 할 수 있었다. 이 과정이 글로 보면 조금 짧지만 대략 2주간 여러 자료를 보면서 긴 시간을 쏟았고 결국 해결 할 수 있어서 굉장한 뿌듯함을 느꼈다. 그렇게 시간이 조금 지난 뒤 RFID 관련으로 교수님께서 보충자료를 올려주셨고 확실히 내가 짠 코드보다 완성도가 높다는 것을 확실히 체감 할 수 있었고 아직 갈 길이 멀다는 걸 느꼈다.

- 진행 미비점 분석 및 발전방향

상용화가 되고 대중화가 되려면 앱인벤터를 사용해 블루투스 모듈을 사용하면 좋을 것 같다는 생각이 든다. 팀원 모두가 아이폰이라 계획 방향이었던 핸드폰

으로는 직접 연결해서 블루투스 통신을 완료하지 못한 점이 조금 아쉽긴하지만 대체방안으로 노트북 블루투스 통신을 이용해 동작을 완료시킬 수 있었다. 앱인벤터로 직접 블루투스 앱을 만들어서 엘리베이터 동작과 연동시켜 확인 하는 것이 앞으로의 발전 방향이라고 생각한다. 또한, 엘리베이터와 현관문을 연결시켜 프로젝트에서 소형화해 구현한 것처럼 더 나아가서는 주차장, 아파트 단지 전체로 범위를 폭넓게 넓혀서 연동한다면 4차 산업혁명 스마트 시대에 맞는 좋은 발전 방향을 가진다고 생각한다.

#### - 본 과제물의 기대 효과

본 프로젝트는 앞서 말했듯이 사건, 사고 방지와 4차 산업혁명 시대에 맞는 스마트 엘리베이터 라는 점에 초점을 맞추고 있다.

인증된 사용자만 아파트 현관을 출입할 수 있고 엘리베이터를 사용할 수 있기 때문에 외부인에 의한 사건, 사고 등을 방지하는 기대효과를 누릴 수 있다. 또한, 4차 산업 혁명 시대에 맞게 모든 동작을 수동적으로 해야하는 아파트 들이 대다수이지만 이 과제물의 취지가 대중적으로 상용화 된다면 수동적이고 직접적으로 동작하는 일들이 줄어들면서 삶 속에서 편리함을 체감할 수 있는 계기가 될 것이다. 엘리베이터, 현관문에 국한된 것이 아니라 아파트 전체, 주차장, 차단기 등등 광범위하게 범위를 넓혀간다면 4차 산업 혁명 시대에 맞게 스마트 엘리베이터, 스마트 아파트 등 여러 산업물로 발전할 가능성이 크다고 생각한다.

## 6장. 소요된 부품 List 및 수행된 일정 정리

### <소요 부품 List>

구성품	수량	기능
LED	5	조도 값에 따른 조명 역할
Tact switch	2	외부 인터럽트 수신 및 동작
CdS(조도 센서)	1	조도 값 수신
LM35(온도 센서)	1	현재 온도 값 수신
LCD	1	온도 및 날짜 표현
SG-90(서보모터)	1	출입문 역할
tinyRTC	1	현재 시간 측정 모듈
RC522-RFID	1	RFID 태그 모듈
Linear actuator	1	엘리베이터 동작 모터
HC-06(Bluetooth)	1	블루투스 통신 모듈

### <수행 일정 정리>

11/14 ~ 11/20	계획 수정 및 동작 검증
11/21 ~ 11/27	RFID, Bluetooth 동작 검증
11/28 ~ 12/04	RFID, 각종 센서 동작 검증
12/05 ~ 12/11	센서 동작 검증 및 H/W 프레임 제작
12/12 ~ 12/21	H/W 프레임 제작 및 보고서 작성

#전체 동작 영상 링크입니다#

<https://www.youtube.com/watch?v=nzGrvzZ4eBY>