

# 최종 보고서

(2023학년도 2학기)

과제명	IoT 기술이 접목된 스마트 금고 제작		
과목명	IoT실습과응용		
조명(조번호)	대도 (3조)		
연구참여자 (담당분야)	20190895 김찬영 (조장, 총괄, 기획, 프로그래밍)		
	20190917 신대철 (프로그래밍, 실험 및 성능개선)		
	20190940 정재호 (부품 구입, 실험 및 성능개선)		
	20190954 허진환 (프로그래밍, 보고서 작성)		
담당교수	교수 정복래	제출일	2023년 12월 13일

## 목 차

1. 설계과제 제목 .....	3
2. 연구목적 .....	3
3. 설계과제의 필요성 .....	3
4. 설계과제의 목표 .....	4
5. 설계과정 .....	5
6. 제작 .....	12
7. 시험 .....	17
8. 평가 .....	27
9. 추진체계 .....	30
10. 설계 추진일정 .....	30
11. 결론 .....	31

### 부록

- 가. 프로그램 소스코드 (커멘트 포함)
- 나. 프로그램 전체 순서도 이미지 (SVG File)
- 다. [Github Link](#)

## 1. 설계과제의 제목 :

IoT 기술이 접목된 스마트 금고 제작

## 2. 연구목적

- 1) Linux(Raspbian) 환경에서의 C언어 프로그래밍을 통한 gcc환경에서의 프로그래밍 역량 향상
- 2) C언어 프로그래밍을 통한 IoT 장치 제어 능력 향상
- 3) 하드웨어와 소프트웨어가 결합된 프로젝트 진행을 통한 공학적 설계 능력 향상
- 4) 특정한 기능을 수행하기 위한 시스템 개발인 임베디드 환경에 대한 이해
- 5) 여러 센서 및 모듈의 응용 능력 향상

## 3. 설계과제의 필요성

최근 경기 불확실성으로 현금·순금 등을 보유한 사람들이 증가하고 개인의 소중한 물건이나 문서, 현금과 같은 자산이 늘어나며 이에 따른 수요가 늘어나고 있다. 금고는 물건을 보관하고 보호하기 위한 장치이기에 시중에 있는 금고를 분석하면 지문인식, 홍채인식, 안면인식 음성 인식 등 생체 인식과 전자식 비밀번호, 카드키 시스템 보안 카메라 기능으로 높은 보안과 안전성을 보장해주고 있다. 이러한 측면들에서 금고는 현대인들에게 매력적으로 다가오며, 수요가 늘어나고 있기에 본 프로젝트의 주제를 금고로 설정하였다.

#### 4. 설계과제의 목표

본 과제에서 달성하고자 하는 성능 목표는 다음과 같으며, 이외에도 본 과제에서 아래와 같은 현실적 제한 요소들을 달성한다.

- 1) 스마트폰과 금고와 블루투스 통신을 통해 통신하도록 설정
- 2) 사용자에게 현재 금고의 상태가 어떤지 LCD를 통해 출력
- 3) 정확하게 지문인식 센서가 동작하도록 실험하여 프로그램 작성
- 4) 난수로 설정되는 LED와 합쳐지는 암호로 정상적으로 금고 잠금 해제되는지 확인하여 프로그램 작성
- 5) CDS셀, 초음파 센서로 금고가 개방되었을 경우를 잘 인식하도록 프로그램 작성
- 6) 금고가 개방되었을 경우 카메라로 개방한 인물을 촬영(캡처)할 수 있음
- 7) 실제 모형을 제작하여 최종 시연 시 발표할 수 있도록 프로젝트 진행

표 1. 본 프로젝트의 현실적 제한요소 항목

현실적 제한 요소들	내 용 (Content)
경제	<ul style="list-style-type: none"> <li>▪ 최소한의 자원을 사용하여 제품의 경제성 고려</li> <li>▪ 부품 구매 시 계획에 맞춰 구매하여 프로젝트 비용 절감</li> <li>▪ 개발되는 제품의 가격 경쟁력 확보</li> <li>▪ 유지 보수 비용이 발생할 수 있음을 고려</li> </ul>
편리	<ul style="list-style-type: none"> <li>▪ 사용자에게 직관적으로 사용하도록 유도</li> <li>▪ LCD, LED 등 출력 장치의 명확성 확보</li> <li>▪ 예상치 못한 기술 장애로 인한 편의상의 문제 발생 가능성 고려</li> </ul>
윤리	<ul style="list-style-type: none"> <li>▪ 개발제품이 불법적, 비윤리적으로 사용되지 않도록 진행</li> <li>▪ 사용자의 생체 정보 등 개인정보를 수집하는 과정에서 발생할 문제 고려</li> </ul>
사회	<ul style="list-style-type: none"> <li>▪ 스마트 금고 사용 중 발생한 문제에 대한 법적 책임과 규제에 대한 논의</li> <li>▪ 스마트 금고 기술에 대한 접근성이 낮은 지역이나 계층에 대한 사회적 격차가 발생함을 고려</li> </ul>

## 5. 설계과정

### 5.1 설계 기초이론

#### 5.1.1 라즈베리파이

라즈베리파이(영어: Raspberry Pi)는 영국 잉글랜드의 라즈베리 파이 재단이 학교와 개발도상국에서 기초 컴퓨터 과학의 교육을 증진시키기 위해 개발한 신용카드 크기의 싱글 보드 컴퓨터이다. 라즈베리 파이는 모두 동일한 비디오코어 IV GPU와, 싱글코어 ARMv6에 호환되는 CPU 또는 신형의 ARMv7에 호환되는 쿼드코어(라즈베리 파이 2), 1 GB의 RAM(라즈베리 파이 2), 512 MB(라즈베리 파이 1 B와 B+), 또는 256 MB(모델 A와 A+, 구형 모델 B)의 메모리를 포함한다. 이들은 SD 카드 슬롯 (모델 A 와 B) 또는 부팅 가능한 매체와 지속적인 정보 저장을 위한 마이크로SDHC를 갖추고 있다. 라즈베리 파이는 리눅스 커널 기반 운영 체제를 사용한다. Raspbian이라는 라즈베리 파이에 최적화된 데비안 계열의 자유 운영 체제가 현재로서는 가장 권장되는 시스템이다.

#### 5.1.2 Servo Motor

서보모터(Servo Motor)는 범용 기계와 비교해 보면 핸들을 돌리는 손에 해당하는 부분으로 머리에 해당되는 정보처리회로(CPU)의 명령에 따라 공작기계 테이블 등을 움직이게 하는 모터이다. 물체의 위치 · 방위 · 자세 · 회전 속도 등을 제어량으로 하고 목표치의 변화에 뒤따르도록 구성된 자동제어계를 서보기구(servo機構)라하며, 보통 피드백(feed back) 회로를 가지고 있어, 출력의 검출부 · 목표치와의 오차 증폭부 · 조작부 · 제어 대상 등으로 구성된다. 이러한 서보계의 조작부에 사용되는 것이 서보모터인데, 전기식(직류 · 교류) · 유압식 · 전기유압식으로 된 것들이 있다.

#### 5.1.3 광학식 지문인식 센서

CCD 카메라형: 유리판에 손가락을 얹으면 유리판 바깥표면으로 초점이 셋팅된 CCD 카메라가 흑백 사진을 찍는데, 이때 지문에 의해 도장 찍듯이 볼록 나온 면이 유리에 붙어있으므로 지문만 초점이 맞아 나머지는 흐려진다. 이것의 윤곽선을 추출한다. 스마트폰 지문인식 방식 중 '광학식'이라고 한다면 대부분 이 방식이다.

#### 5.1.3 초음파 센서

초음파의 특성을 이용하거나 초음파를 발생시켜 가까운 거리에 있는 물체 혹은 사람의 유무, 거리 및 속도 측정 등에 사용된다. 초음파 소자는 고유 진동에 상응하는 교류전압을 가하면 압전 효과에 의해서 효율이 좋게 진동해서 초음파를 발생시킨다.

- 광학 센서 및 IR 센서와 달리 색상에 독립적으로 작동한다. 즉, 물체의 색상이 측정 정확도에 영향을 미치지 않음을 의미한다.

- 유리 및 물과 같이 반투명하거나 투명한 타겟도 측정에 영향을 미치지 않는다.
- 넓은 범위의 물체 감지 및 거리 측정을 제공한다.

초음파 센서의 동작 원리는 물체에 반사된 음파를 그대로 센서로 입력 (진동)시켜서 발생된 전압을 회로에서 처리해 측정 거리를 계산한다.

- 파장과 분해능 : 파장이 짧으면 거리 방향의 분해능이 높고 정밀도가 높아진다.
- 파장의 반사 : 물체의 유/무 감지를 위해 초음파가 물체에 닿으면 반사된다.
- 온도 영향 : 물체까지의 거리를 정밀하게 측정하기 위해서는 온도에 따른 보정이 필요하다.

\*아두이노에서도 초음파 센서의 정밀도를 확보하기 위해서 온도센서를 같이 사용하기도 함.

### 5.1.4 블루투스

근거리 무선 통신 기술. 줄여서 BT, 블루라고도 부른다. 과거 피쳐폰 시절에는 PAN(Personal Area Network)이라는 이름으로 탑재되기도 했다. 블루투스의 무선 시스템은 ISM(Industrial Scientific and Medical) 주파수 대역인 2400~2483.5MHz를 사용하고 있다. 이 중에서 위 아래 주파수를 쓰는 다른 시스템의 간섭을 막기 위해서 2400MHz 이후 2MHz, 2483.5MHz 이전 3.5MHz까지의 범위를 제외한 2400MHz 이후 2483.5MHz 이전 3.5MHz까지의 범위를 제외한 2402~2480MHz, 총 79개 채널을 사용하고 있다. 블루투스는 같은 주파수 대역을 사용하기에 서로간의 전파 간섭이 생길 수 있는데 이를 예방하기 위해서 주파수 호핑(Frequency Hopping)을 사용하는데 주파수 호핑이란 많은 수의 채널을 특정 패턴에 따라서 빠르게 이동하여 데이터를 조금씩 전송하는 방식이다. 블루투스는 기기간에 동기화가 되어야만 작동하게 되어 있는데 마스터와 슬레이브 구성으로 연결된다. 마스터와 슬레이브와의 동기화 외에 슬레이브-슬레이브간의 통신은 불가능하다.

### 5.1.5 CDS

- 빛의 양을 측정하는 센서
- 조도 센서에도 동작 방식이나 구성에 따라 다양한 종류가 있음
- 빛의 세기에 따라 저항값이 바뀌고 이에 따라 전압값을 측정하면 빛의 세기를 디지털로 변환할 수 있음
- 라즈베리 파이의 ADC 기능을 통하여 아날로그 신호를 디지털로 변환하여 측정

실습에 사용할 온/습도 센서 - CDS

- CDS 센서는 카드뮴과 황의 화합물인 황화 카드뮴(CdS)를 사용한 광전도 소자를 이용한 센서로 광도전 효과, 반도체에 빛을 조사하면 반도체 중 캐리어 밀도가 증가하여 도전율이 증가하는 현상으로 외부로부터의 빛의 에너지에 의하여 가전자대의 자유

전자가전도대에 여기되어 도전성을 나타내게 되는 현상을 이용

- 반도체에 빛이 닿으면 전자와 정공이 증가하고 조사된 빛 에너지에 비례하여 전류가 증가하는 원리를 이용

- CdS는 극성 없이 교류회로에 사용 가능하며 제작은 세라믹 기판에 CdS분말을 소결하여 제작

- CdS는 광도전체를 좌우 전극으로 둘러싸서 저항체를 구성

\* CdS의 황화카드뮴이 광도전체 역할을 하게 되며 빛 에너지가 입사하면 내부적으로 광도전 효과가 발생하고 두 전극 사이에 내부저항이 작아지게 됨

- CDS센서의 구조 및 기호CDS 센서는 소형이며, 가격이 저렴하고 무극성 소자이므로 회로구성이 간단반응시간이 늦어 즉각적인 반응을 필요로 하는 응용에는 적합하지 않음카메라의 노출계나 조명의 자동 점멸 장치 TV수상기의 자동휘도 조정장치 등에 사용

- 조도센서 제어 방법

조도 센서는 측정된 조도를 해당하는 포트 핀으로 출력출력 값은 아날로그 이므로 ADC(Analog to Digital Converter) 기능을 통하여 이를 디지털 수치로 변환하여 확인 아날로그 입력 핀(Analog in 0~7)을 통하여 센서에서 출력되는 값을 수신입력 값은 아날로그-디지털 변환 칩(MCP3208)을 통해 A/D 변환하여 디지털 값을 라즈베리 파이에서 인식많은 양의 빛이 들어갈 경우, 수치가 높고, 적은 양의 빛이 들어갈 경우, 수치가 낮음

### 5.1.6 CCD (카메라)

디지털 카메라의 하나로, 전하 결합 소자(CCD)를 사용하여 영상을 전기 신호로 변환함으로써 디지털 데이터로 플래시 메모리 등의 기억 매체에 저장하는 장치. CMOS형 카메라보다는 화질이 우수하나 전력 소모와 가격 면에서 불리하다.

CCD는 이미지를 이루는 점(點)을 표현하는 화소(畵素)가 같은 범위에 몇개 들어있는냐에 따라 성능이 구별된다. 우리가 흔히 디지털 카메라를 고를 때 300만 화소나 400만 화소나를 따지는 것은 바로 이 CCD에 들어간 화소수(畵素數)를 말한다. 같은 범위에 화소가 많을수록 더 선명한 이미지를 얻을 수 있지만, 화소의 집적도(集積度, degree of integration)뿐 아니라 CCD 자체의 크기도 화질(image quality)에 큰 영향을 준다.

### 5.1.7 LED

발광 다이오드, 순방향으로 전압을 가했을 때 발광하는 반도체 소자

충격에 강하고 점등과 소등이 쉽다는 장점이 있으며,

광량 조절과 색상조합이 쉬운편이어서 다양한 전자제품에서 많이 사용됨

제어방법 : LED를 제어하기 위해 GPIO (General-purpose Input Output) 포트를 사용

(GPIO는 범용적으로 사용되는 입출력 포트 - led, motor, keypad 등 에서도 공통적으로 사용)

임베디드 시스템을 구성하는 과정에서 주변장치, 소자의 동작을 제어하기 위해서 적절하게 신호를 보내야한다.

이 신호를 입력&출력하기 위해 포트를 할당시켜준다.

예시)

#define LED\_PIN\_1 5 : 1번째 LED소자를 GPIO기판의 5번 단자와 연결

#define KEYPAD\_PB1 20 : 1번째 키패드버튼 소자를 GPIO 20번 단자와 연결

LED는 양극에 5볼트의 전압을 인가하고 음극에 0볼트를 유지하면 LED 다이오드에 전류가 흐르게 되어 램프가 점등

프로그램의 관점에서는 #define LED\_ON 1 #define LED\_OFF 0 과 같이 값을 할당하여 사용

LED를 점등하고 싶으면 상태값에 1으로 설정. LED를 소등하고 싶다면 상태값을 0으로 설정

LED 제어 관련하여 저항을 읽는 법을 알고 있어야 한다

저항에는 4개의 선이 존재한다. 각각의 선은 왼쪽에서부터 첫번째숫자, 두번째숫자, 0의 갯수, 저항값의 오차의 의미를 가진다.

### 5.1.8 키패드

스위치(Switch, 개폐기)는 전류의 흐름을 막거나 계속 흐르게 하는 용도의 장치, 일반적으로 켜진 상태인 1, 꺼진 상태인 0의 상태를 가지며 전기 회로를 닫거나 열어서 전류의 흐름을 허용 또는 차단

- 누름 버튼 스위치 : 버튼을 누르면 전기 회로가 닫히고 버튼을 놓으면 회로가 열림
- 토글 스위치 : 토글 스위치는 스위치를 눌러서 상태를 변경, 다시 누르면 다시 원래 상태로 돌아감
- DIP 스위치 : DIP 스위치는 일렬로 배열된 작은 스위치, 이진형태의 설정에서 주로 사용됨
- 로터리 스위치 : 회전하는 원판 모양의 스위치로 여러 개의 전기적 연결을 선택함
- 실습 장비의 스위치는 2X2 (Button Module) 와 4X4 (Keypad Module) 푸쉬 버튼 스위치(Push Button Switch)로 버튼을 누르면 전류를 계속 흐르게 하여 신호의 상태를 제어

제어 방법

- 푸쉬 버튼 스위치의 경우, 버튼을 눌러 제어
- 푸쉬 버튼은 탄성이 있어 누르면 다시 튀어나오려고 하며, 눌림 상태에 Low 신호를 인가 하고, 눌리지 않은 상태에 High 신호를 인가 (Pull-up 저항 내장)
- 스위치를 제어하기 위해서 앞서 학습한 GPIO를 사용



- LED가 GPIO의 출력을 통해 On/Off 했다면, 스위치는 입력, 즉 Input을 통해 라즈베리 파이에 신호가 수신
- Keypad 모듈의 푸쉬 버튼 스위치를 눌렀을 때, Low 신호가 발생되며, 라즈베리 파이의 입력 핀과 연결된 케이블을 통해 신호가 전달되고, 이를 확인한 라즈베리 파이에서 신호에 따라 동작
- 푸쉬 버튼 스위치는 해당 출력 핀에 5V의 전압이 인가되어 있다가 스위치를 누름으로써 전류가 흐르게 되고, GND와 연결되어 Low 상태가 되며, 프로그램에서는 전달되는 전압 신호를 0과 1로 받아들여 상태에 따라 특정 동작을 제어

### 5.1.9 Buzzer

소리 신호 알림 장치이며, 비퍼(Beeper)라고도 호칭. 짧은 음향 신호를 생성하여 소리를 발생시키는 장치이다. 주로 전자 기기나 프로젝트에서 알림이나 경고음을 내기 위해 사용된다. Buzzer는 간단한 방식으로 동작하며, 전기적 신호를 받아 진동판을 흔들어서 소리를 발생시킨다. 주로 특정 주파수의 전기적 신호를 Buzzer에 전달하여 원하는 음을 만들어낸다. 코드에서 'softToneWrite' 함수를 사용하여 Raspberry Pi의 GPIO 핀에 연결된 Buzzer에 주파수 값을 전달하여 소리를 생성한다. 버저는 얼마만큼의 주파수를 전달해 주는가에 따라 출력되는 소리가 달라진다. 또한, 버저의 종류에 따라 주파수에 따른 소리의 높낮이를 가짐. 실습 모듈은 압전방식을 사용하는 피에조(Piezo) 스피커이다.

### 5.1.10 Text LCD

텍스트 LCD란 액정을 이용하여 화소에 도달하는 빛을 선택적으로 투과시키거나 차단시켜 문자를 표시하는 시각적 전달장치이다. 주로 임베디드 시스템이나 프로토타이핑 프로젝트에 사용된다. 라즈베리 파이와 같은 임베디드 시스템에서 textlcd 모듈은 GPIO 핀을 이용하여 제어된다. 코드에서는 WiringPi 라이브러리의 lcdInit, lcdClear, lcdPosition, lcdPuts 등의 함수를 사용하여 LCD에 텍스트를 표시하고 제어한다.

- TEXT LCD 구조는 LCD 패널(표시기)과 제어기가 함께 모듈 형태로 구성
- 제어기 내부에는 명령(Instruction) 레지스터, 데이터(Data) 레지스터, AC(Address Counter), BF(Busy Flag), 문자발생램(CGRAM), 문자발생롬(CGROM), 데이터표시램(DDRAM) 포함
- 제어기에서 버스를 통하여 데이터를 전송함으로써 표시

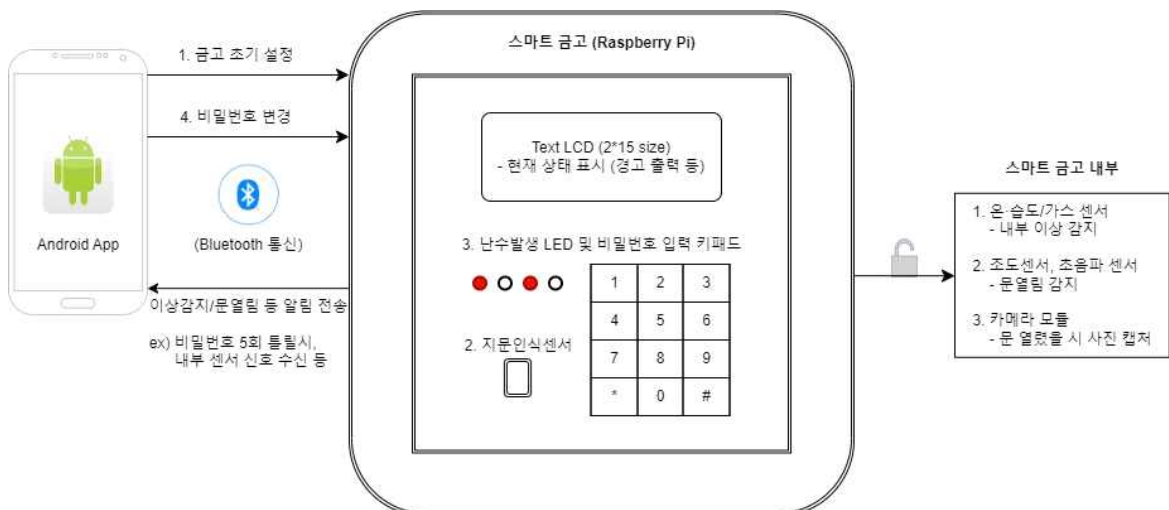
### 5.2.1 소프트웨어 기능블록도 (개념설계)

스마트폰 Application을 통해 스마트 금고(Raspberry Pi)와 Bluetooth 통신하여 금고 초기 설정(초기 비밀번호 설정, 지문인식센서를 통한 지문 등록 등)을 마친 후, 1차 잠금인 지문인식 성공한 경우에 한하여 비밀번호 입력 키패드가 활성화되어 2차 잠금인 비밀번호 입력이 가능하게 하였다.

만약 지문/비밀번호가 일정 횟수 불일치할 경우, 해당 경고 알림음이 부저로 출력됨과 함께 경고 문구가 스마트폰 Application 알림, 금고의 Text LCD로 출력되며 일정 시간동안 잠금 해제 시도가 불가능하게 된다.

스마트 금고 내부에는 온·습도 센서와 가스센서가 있어 내부 환경에 대한 이상을 감지하고, 문열림을 감지하여 잠금 모터가 잠기는 시점과 문열림에 대한 알림 문구를 전송하게 하는 조도센서와 초음파 센서가 존재하며, 카메라 모듈을 통해 문 열렸을 시 해당 인물을 찍을 수 있도록 영상 캡처한다.

그림 1. 스마트 금고 동작 기능 블록도



## 5.2.2 기능적 요구사항 명세서

ID	요구사항	내용	설명	우선순위
R_01	금고 현황 출력	LCD를 통해 금고의 전반적 현황을 파악가능	1. LCD에는 금고의 잠금 상태, 마지막 문 열림 시간, 비밀번호 변경 안내 등의 정보가 표시 2. 이상 감지 시 해당 금고의 온도 및 습도를 표시	2
R_02	금고 초기 설정	초기 설정은 편리하게 스마트폰을 활용하여 비밀번호 설정 및 재설정가능	1. 금고의 전원을 켜 후 페어링된 스마트폰을 이용하여 초기 비밀번호를 설정 가능 2. 비밀번호를 잊었을 경우, 연결된 스마트폰을 사용하여 비밀번호를 초기화하거나 변경	3
R_03	오입력 방지 모드	사용자가 안전하게 비밀번호를 입력가능하도록 기능 구현	1. 지문인식을 통해 키패드를 활성화시키고, LCD에 비밀번호 입력 안내 표시 2. 시간에 따라 변화하는 4개의LED를 활용하여 사용자는 허수입력 후 비밀번호 입력하여 사용자가 항상 다른 비밀번호를 입력 3. 비밀번호를 입력하는 동안 LCD창에는 비밀번호를 노출시키지 않음	4
R_04	문 상태 감지	현재 금고의 문 상태 감지 기능	1. 조도센서와 초음파 센서로 현재 문의 상태를 감지 2. 문이 열리면 내부에 있는 카메라 모듈로 사진을 촬영해 누가 문을 열었는지 확인 3. LCD에는 문이 열렸는지 표시하고 스마트폰에 알림 전송	1
R_05	이상감지 시스템	금고의 이상이 감지될 경우 사용자에게 알림	1. 비밀번호가 5회 이상 틀렸을 때 이상감지 2. 온습도 센서 및 가스 센서를 통해 이상감지 3. 감지가 되면 스마트폰에 경고 알림을 전송하고 금고를 잠금상태로 전환 후 부저음 과 LCD에 경고음 및 경고창 출력	5

### 5.2.3 전체 순서도 (상세설계)

전체 순서도는 소스코드와 함께 부록으로 첨부하였습니다.

## 6. 제작(Implementation)

### 6.1 제작과정

#### 6.1.1 라즈베리파이 사용자 정의 함수 목록 (iot.c)

함수명	설명
void setup()	아두이노와 라즈베리파이 간 시리얼 통신 set-up
char loop_getc()	아두이노와 라즈베리파이 간 시리얼 통신, 라즈베리파이는 수신 받은 char형 변수를 return
int KeypadRead(void)	키패드 입력받는 함수
void ServoControl(unsigned char str)	서보모터 컨트롤 함수
void Change_FREQ(unsigned int freq)	Buzzer 주파수 변경
void STOP_FREQ(void)	Buzzer 주파수 정지
void Buzzer_Init(void)	Buzzer 초기화
void btnsound(int mute)	버튼 입력 시 소리 출력 함수. mute 값이 0이면 소리가 출력, 1이면 소리 출력하지 않음.
void Correct_FREQ(int mute)	정상적으로 비밀번호 입력 시 소리 출력
void Close_FREQ(int mute)	문이 닫힐 경우 소리 출력
void Incorrect_FREQ(int mute)	틀릴 경우 소리 출력
void lcd_menu(int disp, int lcd_stat)	금고의 메인 메뉴 LCD 출력, LCD_STAT에 따라 화면 출력 변경됨(스크롤 기능)
void lcd_inputpw(int disp, int t)	PW 입력하는 LCD 화면 출력. PW는 화면상에 *로 표시됨.
void lcd_sound(int disp, int mute)	LCD상에 소리 설정 화면 출력 함수
void init_led(void)	LED 초기화
void setLed(int leds)	LED 번호에 따라 LED를 켜는 함수
int decimalToBCD(int decimal)	십진수 숫자를 BCD (Binary-Coded Decimal)로 변환하는 함수
void led_BCD(int number)	BCD로 표시된 숫자를 LED에 출력하는 함수
void getCurrentTime(struct tm *t, char *displayTime)	시간을 받아오는 함수
void getCurrentDate(struct tm *t, char *displayTime)	날짜 받아오는 함수
int time_led(struct tm curr, char buf_lcd[10])	시간(분)에 따른 LED 출력. 현재 시간이 16분일 경우, 숫자 6의 BCD인 0110 형태로 LED가 출력된다.
int pwcmp(int str1[], int str2[])	정수형 배열 비교 함수. 비밀번호가 맞는지 확인할 때 사용한다. 비밀번호 6자리에 대해서만 동작함.
int mincmp(int buf[], int min)	비밀번호 뒤에 입력한 숫자가 현재 시각의 분과 일치하는지 확인하는 함수. 예를 들어 현재 시각이 17분일 경우, LED는 0111 형식으로 켜질 것인데, 이는 숫자 7을 의미하며 LED가 3개 켜져있으니 비밀번호 6자리 뒤에 777을 붙여서 입력해야 금고의 문이 열린다. LED의 점등 개수만큼 LED가

	나타내는 BCD코드를 비밀번호 뒤에 추가로 입력하면 된다.
<code>void captureImage()</code>	금고 문을 열었을 시 카메라 촬영 함수. 저장 위치와 저장 파일 형식은 /project/capturelog/openlog/MMDD_HHMM.jpg 으로 설정하였다.
<code>void capturefailedImage()</code>	금고 비밀번호 5번 틀렸을 경우 카메라 촬영 함수. 파일명 형식 지정: /project/capturelog/failedlog/MMDD_HHMM.jpg

### 6.1.2 라즈베리파이 main function 동작 과정 (iot.c)

- (1) 라즈베리파이 GPIO SETUP
- (2) TEXT LCD 초기화
- (3) Buzzer 설정
- (4) 키패드 핀을 입력으로 설정
- (5) LED 초기화
- (6) 여러 변수 선언(tmp, password, lcd\_stat 등 프로그램 동작에 필요한 변수들)
- (7) 아두이노로부터 문자 'a' 를 수신받기 위한 루프(지문초기등록 완료 시 아두이노는 'a' 를 라즈베리파이로 송신한다)
- (8) 아두이노로부터 6자리의 정수를 수신해 비밀번호를 초기 설정함. 통신 형식은 문자열이지만, 문자열 ASCII Value로부터 49를 빼면 그것이 실제 정수의 값이 된다. 이를 비밀번호 배열에 저장한다.(1 << 수신한 값 - 49 형식으로 라즈베리파이에 저장됨)
- (9) 초기 지문 등록 및 비밀번호 등록이 종료되면 바로 금고 메인메뉴 동작을 시작한다. main 함수 내의 while(1) 부분에서는 in\_key와 past\_key를 사용하여, 키패드의 버튼이 눌리는 순간을 인식하기에 연속적인 입력을 방지하고, 사용자의 요청에 맞는 동작을 수행한다.
- (10) 1번 메뉴는 시간 확인 메뉴로, 라즈베리파이를 기준으로 시간대를 설정하여 현재 시간을 출력한다. 현재 시간에 맞는 LED 또한 출력된다.
- (11) 2번 메뉴는 비밀번호 입력 메뉴이다. 사용자는 등록된 지문과 일치하는지 여부를 확인한 다음 비밀번호를 입력할 수 있다. 2번 메뉴로 진입하게 되면, 먼저 main 함수에서 아두이노로부터 문자 'b' 를 수신받기 위해 while(1) 대기한다. 그 이후 키패드에서 비밀번호 6자리를 입력 및 LED에 따른 추가 난수를 입력한 다음

- 엔터(BE)를 입력하면 pwcmp()함수와 mincmp()함수를 통하여 확인하고, 맞을 경우 문을 열어준다.
- (12) 문을 열 때 서보모터가 회전하고, 문을 연 사람이 누구인지 기록을 남기기 위하여 captureImage()함수로 카메라 촬영한다. 그 다음, while(1)문을 활용하여 아두이노로부터 'c' 값을 받기 전 까지 대기한다. 'c' 값은 아두이노에 장착된 CDS, 초음파 센서를 통해 문이 닫혔다고 인식이 되면 아두이노가 라즈베리파이로 송신하는 값이다. 따라서 사용자가 문을 닫기 전 까지 서보모터로 금고의 문을 잠그지 않도록 동작한다. 사용자가 문을 닫으면 서보모터가 문을 잠그도록 동작한다.
- (13) 비밀번호가 틀렸을 경우, passwd\_cnt를 통해 얼마나 틀렸는지 계속 카운트한다. 이 값이 5가 되었을 경우, 금고에서 계속 틀리는 사람의 사진을 capturefailedImage()함수를 사용하여 촬영하고, 5초동안 금고의 동작을 정지한다.
- (14) 3번 메뉴는 비밀번호 변경 메뉴이다. 비밀번호 변경 또한 2번 메뉴처럼 지문인식을 거쳐서 동작하도록 설정하였다. 사용자는 원래의 비밀번호 6자리만 입력하고, 이 때는 LED에 관한 값을 입력하지 않도록 하였다. 비밀번호가 맞을 경우, 새 비밀번호를 입력하도록 하였다.
- (15) 4번 메뉴는 음소거 기능으로, mute의 값을 변경하는 기능이다. mute의 초기값은 0으로 설정하였으며, mute가 1일 경우 소리가 출력되지 않는다. 소리를 출력하는 부분이 있는 사용자 정의 함수의 파라미터로 mute를 사용하여 그 함수들 내부에서 mute가 1일 경우 소리가 출력되지 않도록 프로그래밍하였다.
- (16) 사용자가 금고를 사용하면서 발생할 수 있는 예외에 대한 처리를 진행하였다. 비밀번호 입력 및 변경 시 사용자가 숫자를 6자리보다 적게 입력할 경우, 비밀번호 메뉴를 탈출하고 다시 초기 메뉴를 실행하도록 설정하였다. 또한, in\_key와 past\_key 기법을 활용하여 키패드의 입력이 여러 번 눌리는 것을 방지하였다.

## 6.1.2 아두이노 main loop 동작 과정 (arduino\_iot.ino)

아두이노->라즈베리파이로 송신하는 value	조건	설명
'a'	아두이노에서 초기 지문이 등록되었을 경우	가장 처음 프로그램이 실행 되고 동작하는 부분으로, 지 문 등록을 하는 부분이다.
6자리 문자열	블루투스 연결된 스마트폰에 서 아두이노로 이 값을 전송 받았을 경우	지문등록 이후 비밀번호 초 기 설정 부분이다. 6자리 문 자열로 숫자를 입력받는다.
'b'	지문인식 결과 지문이 일치 할 경우	지문이 일치할 경우에만 동 작하고, 일치하지 않으면 계 속 지문인식을 반복하도록 한다.
'c'	CDS 값과 초음파 센서 값을 통해 문이 닫혔다고 판단되 었을 경우	센서의 값을 &&연산으로 조 건을 걸어, 두 값이 모두 조 건을 충족할 경우에만 문이 닫혔다고 판단한다.

- (1) 지문인식 센서 설정 단계
- (2) 블루투스 통신 설정
- (3) 아두이노 시리얼 통신에 대한 설정
- (4) 아두이노 메인 동작 void loop()
- (5) while(!getFingerprintEnroll())을 통해 아두이노에 연결된 지문인식 센서의 초기 등  
록이 완료될 때 까지 반복한다. 완료될 경우, Serial.println("a"); 코드를 통하여  
라즈베리파이에 문자 'a' 를 전송한다.
- (6) 다음은 아두이노와 블루투스로 연결된 스마트폰에서 arduino bluetooth control 어  
플리케이션을 실행해 아두이노로 6자리 숫자를 전송한다. 이를 받은 아두이노는  
그것을 라즈베리파이로 문자 형태로 전송한다. 6자리 이상 입력했을 경우, 문자열  
의 맨 뒤 6자리만 입력받는다.
- (7) 그 이후 라즈베리파이의 금고 초기 설정이 마무리되고 메뉴 프로그램이 동작하게  
된다. 금고에서 2번 메뉴를 누르면 사용자가 지문인식을 해야 되는데, 이것을 위  
하여 아두이노에서 while문 코드를 사용해 지문인식을 계속해서 반복한다. 지문인  
식 결과 값이 일치할 경우 'b' 를 라즈베리파이로 전송하고, 지문인식을 사용하  
가 하지 않았을 경우에는 조도 센서와 초음파 센서를 활용해 문이 열린 상태인지,  
닫힌 상태인지 판단하는 코드가 동작한다. 문이 닫힌 상태일 경우는 CDs 값이 80  
아래이며 초음파 센서 값이 13cm 아래일 경우 닫혔다고 판단하고, 라즈베리파이

로 ‘c’ 를 보내도록 한다.

## 6.2 제작시 문제점 및 개선사항

### 6.2.1 아두이노와 라즈베리파이 시리얼 통신 간 발생하는 버퍼 관련 문제 발생

아두이노에서 라즈베리파이로 상태에 관한 정보를 전송해야 하는데, 이 과정에서 예기치 못한 값이 전송됨을 확인하였다. 그 이유를 조사한 결과 문이 닫혔음을 판단하는 과정에서 ‘c’ 를 한 번만 보내도록 프로그래밍하였지만 통신 버퍼에 ‘c’ 값이나 알 수 없는 값들이 여러 번 보내지는 것을 확인하였다. 우리가 의도한 동작은 지문확인-단힘-열림-단힘과 같이 각 단계가 한 번씩 인식되고 통신을 하는 것이었는데, 실제로 시리얼 통신 과정에서 지문인식-단힘-단힘-단힘-열림-열림-열림-단힘-단힘-단힘-지문인식과 같이 각 상태가 여러 번 인식되어 통신 버퍼가 우리가 의도치 않게 입력되는 것이었다. 이것을 해결하기 위하여 아두이노 내부에서 check라는 변수를 사용하여, 문이 열린 상태는 check를 0으로 설정, 닫힌 상태는 1로 설정하여 check가 0에서 1로 바뀌는 경우에만 문이 닫혔다고 인식하여 시리얼 통신으로 ‘c’ 를 송신하도록 설정하여 해결하였다. 그 이외에도 아두이노에서 라즈베리파이로 송신하는 과정에 예외를 없애기 위하여 프로그래밍하였다.

### 6.2.2 온/습도 센서 및 가스 센서 사용 불가 문제 발생

설계 초기 단계에서 온/습도 센서와 가스 센서를 사용하여 금고 내부의 이상을 감지하는 경우 경보를 울리는 시스템을 설계를 하려고 하였으나, 온/습도 센서의 경우는 라즈베리파이의 GPIO 포트 부족으로 인해 추가하지 못하였으며, 가스 센서의 경우 코드에 추가하여 동작을 시도하였으나 계속 값이 0으로 출력되었다. 센서 감도의 문제라 생각하고 십자 드라이버로 센서 옆에 위치한 가변저항을 조정하였으나 값의 변화를 얻을 수 없었다. 따라서 해결 방법을 찾을 수 없으며, 큰 동작은 무리 없이 작동하였기 때문에 위의 두 센서를 사용하지 않도록 팀원들과 합의하여 결정하였다.

## 6.3 지도교수 지도내용 및 지적사항 조치결과

### 6.3.1 지도교수 Comment: GPIO 포트가 부족한 상황의 발생을 고려

이를 해결하기 위하여 초음파, 블루투스, CDS, 지문인식 등 센서를 통해 값을 받는 것은 아두이노로 연결하였으며 라즈베리파이로 시리얼 통신하여 값을 넘겨주는 방식을 채택하였다. 결과적으로, 라즈베리파이의 GPIO 포트를 확보할 수 있었다.



## 7. 시험

그림 1. 라즈베리파이 GPIO 포트 연결 및 아두이노와 연결

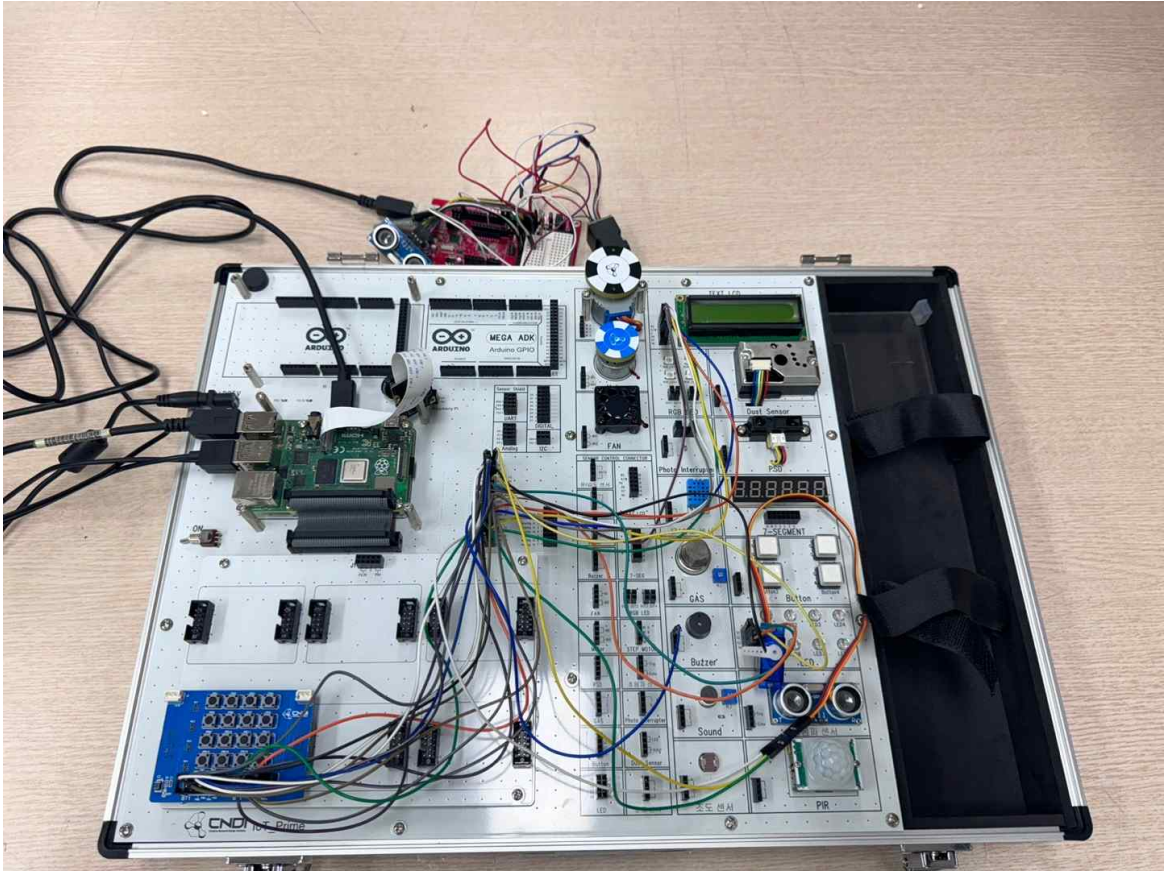


그림 2. 모형의 외부, 내부

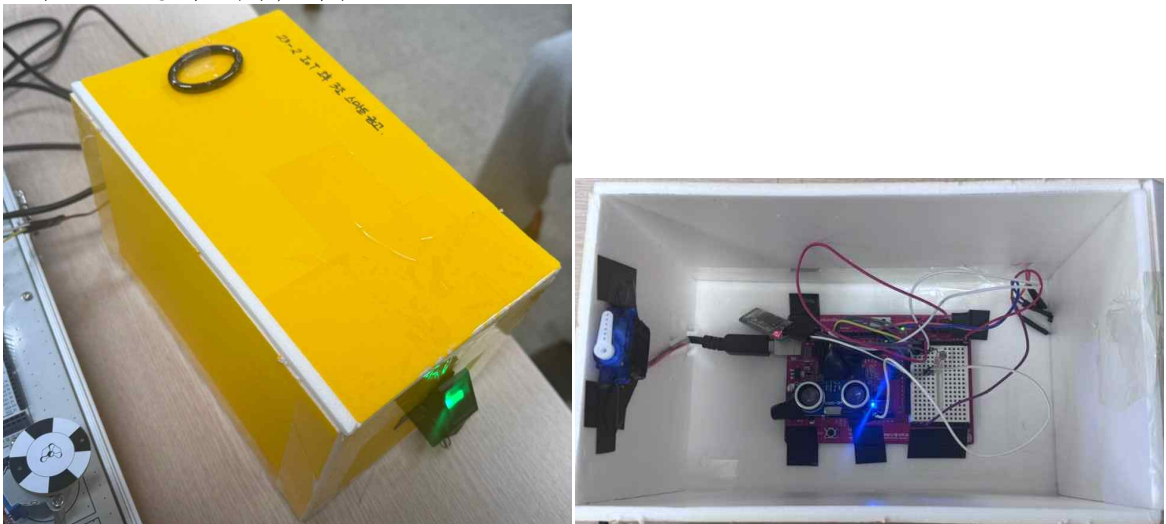


그림 3. 스마트폰과 아두이노 어플리케이션으로 블루투스 연결

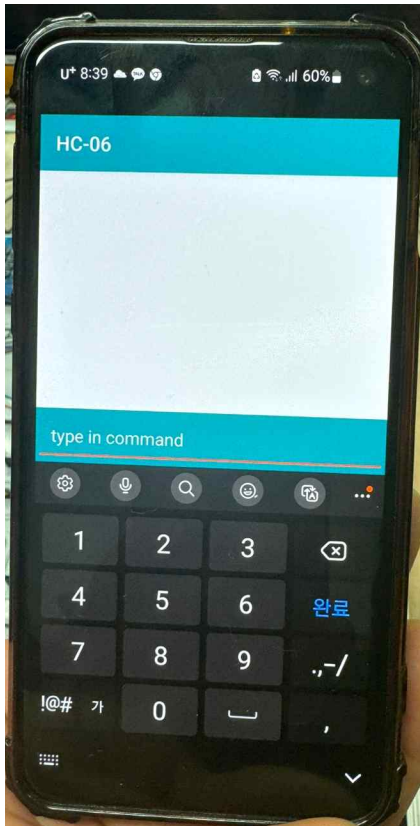


그림 4. 금고 전원 ON 이후 지문 등록 단계



그림 5. 아두이노에 연결된 지문인식 센서 등록 완료 후 라즈베리파이 ‘a’ 수신

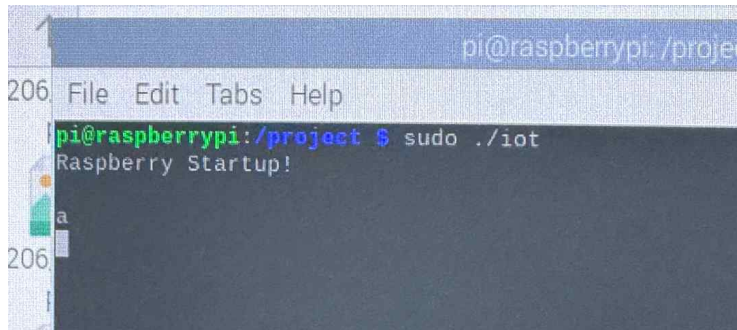


그림 6. 스마트폰에서 아두이노로 초기 비밀번호 전송(1 2 4 8 16 32는 키패드 상에서 1 2 3 4 5 6과 같음)

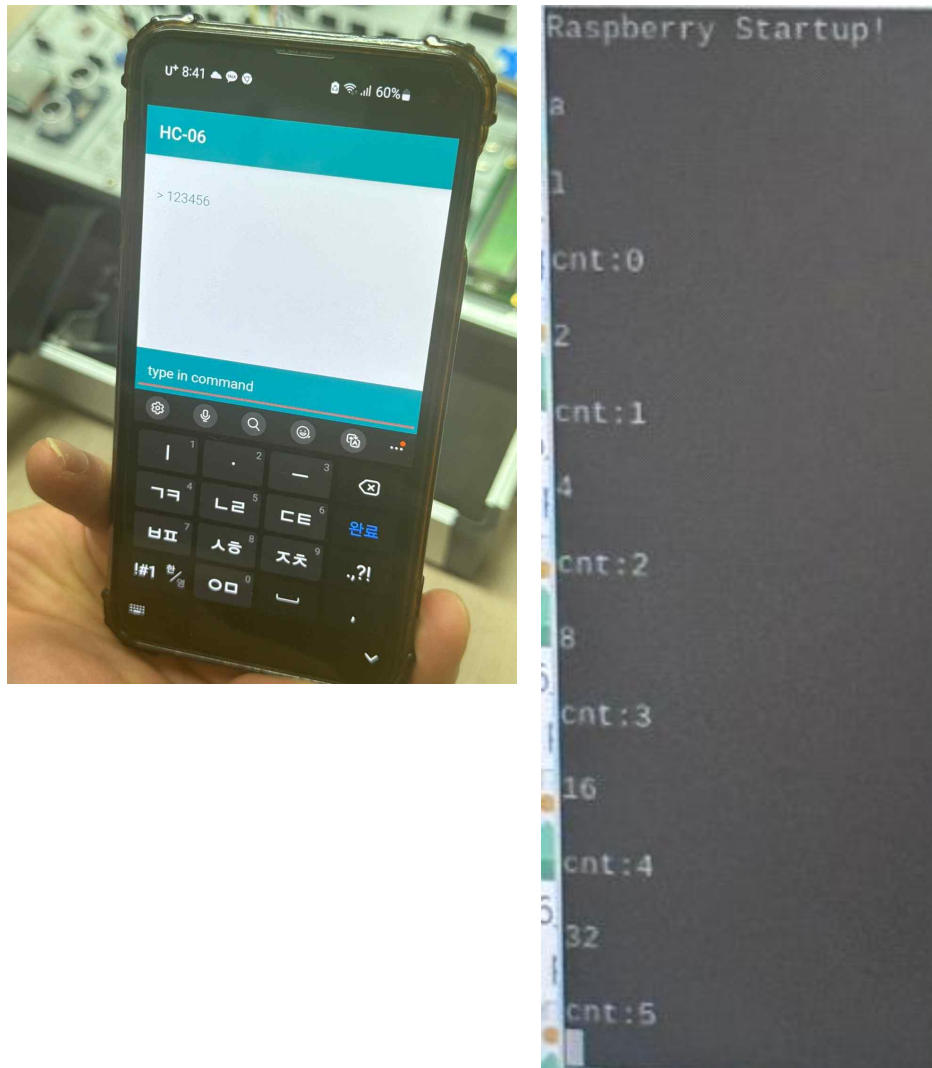




그림 7. 초기 설정 완료 후 금고 메인메뉴 Text LCD에 출력됨



그림 8. 스크롤 기능 확인

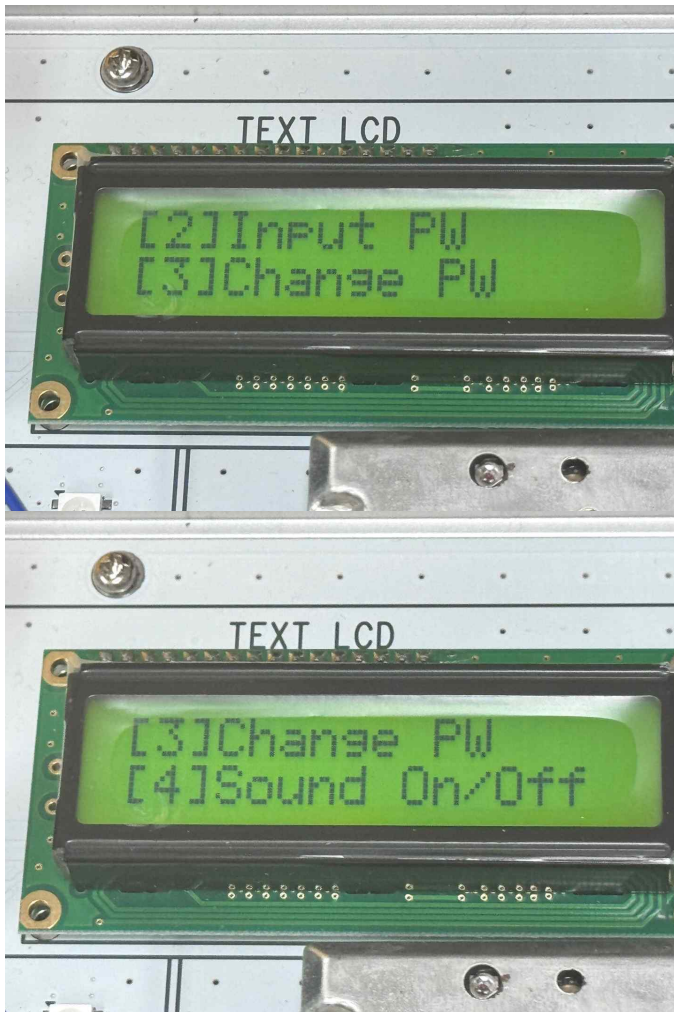


그림 9. 1번 메뉴인 현재 시간 선택 시 정상적으로 Text LCD와 LED가 출력됨  
(42분이기 때문에 2에 해당하는 BCD코드 0010이 출력됨. LED1이 낮은 자리, LED4가 높은 자리)

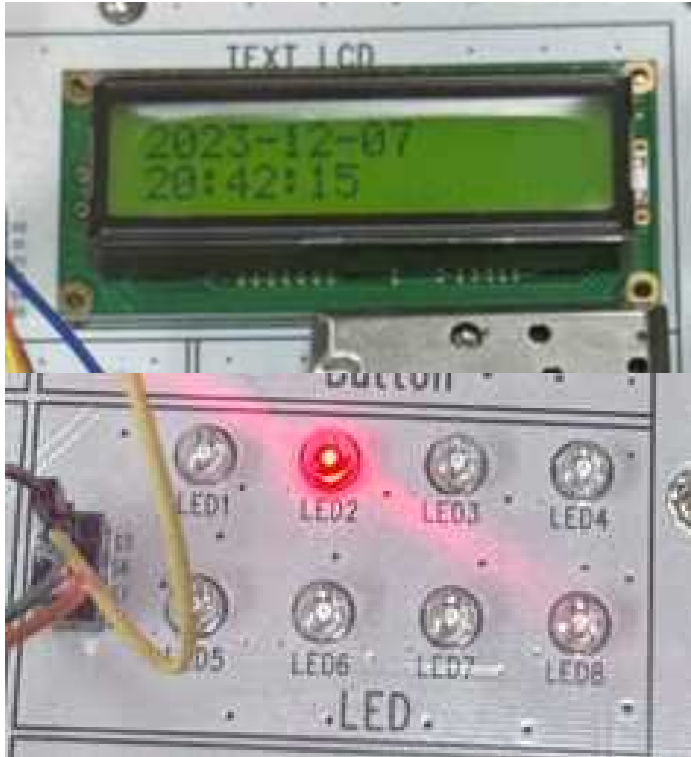


그림 10. 2번 메뉴인 비밀번호 입력 메뉴 선택 시 먼저 지문인식을 통과해야 한다.



그림 11. 지문인식 이후 패스워드 입력 화면. 현재 시간이 42분이었으니 비밀번호는 1234562를 입력하였다. 만약 43분이었다면, 3분에 대한 BCD가 0011이기 때문에 LED가 두 개 점등될텐데, 그러면 12345633으로 입력하면 된다. LED의 점등 개수만큼 LED가 나타내는 BCD코드를 비밀번호 뒤에 추가로 입력하면 된다.

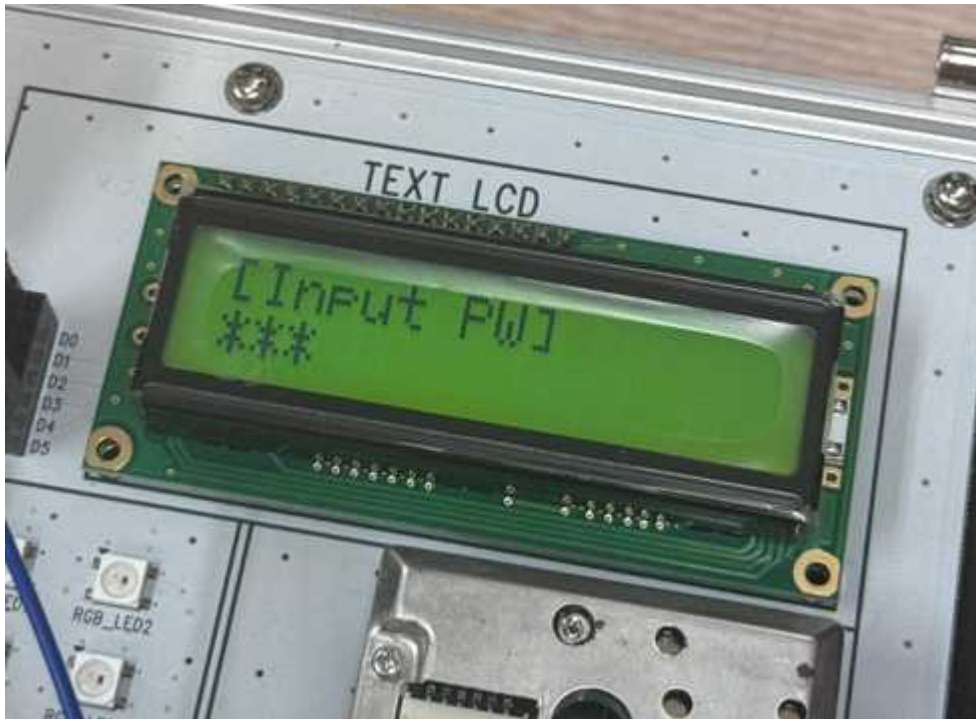


그림 12. 비밀번호가 일치해 문이 열린 모습





그림 13. 문이 열리고 카메라로 사진이 찍히는 모습



그림 14. 문을 닫으면 자동으로 문 닫힘을 감지해 서보모터가 문을 잠금



그림 15. 비밀번호 5회 이상 틀린 경우, 사진 촬영





그림 16. 3번 메뉴인 비밀번호 변경 과정

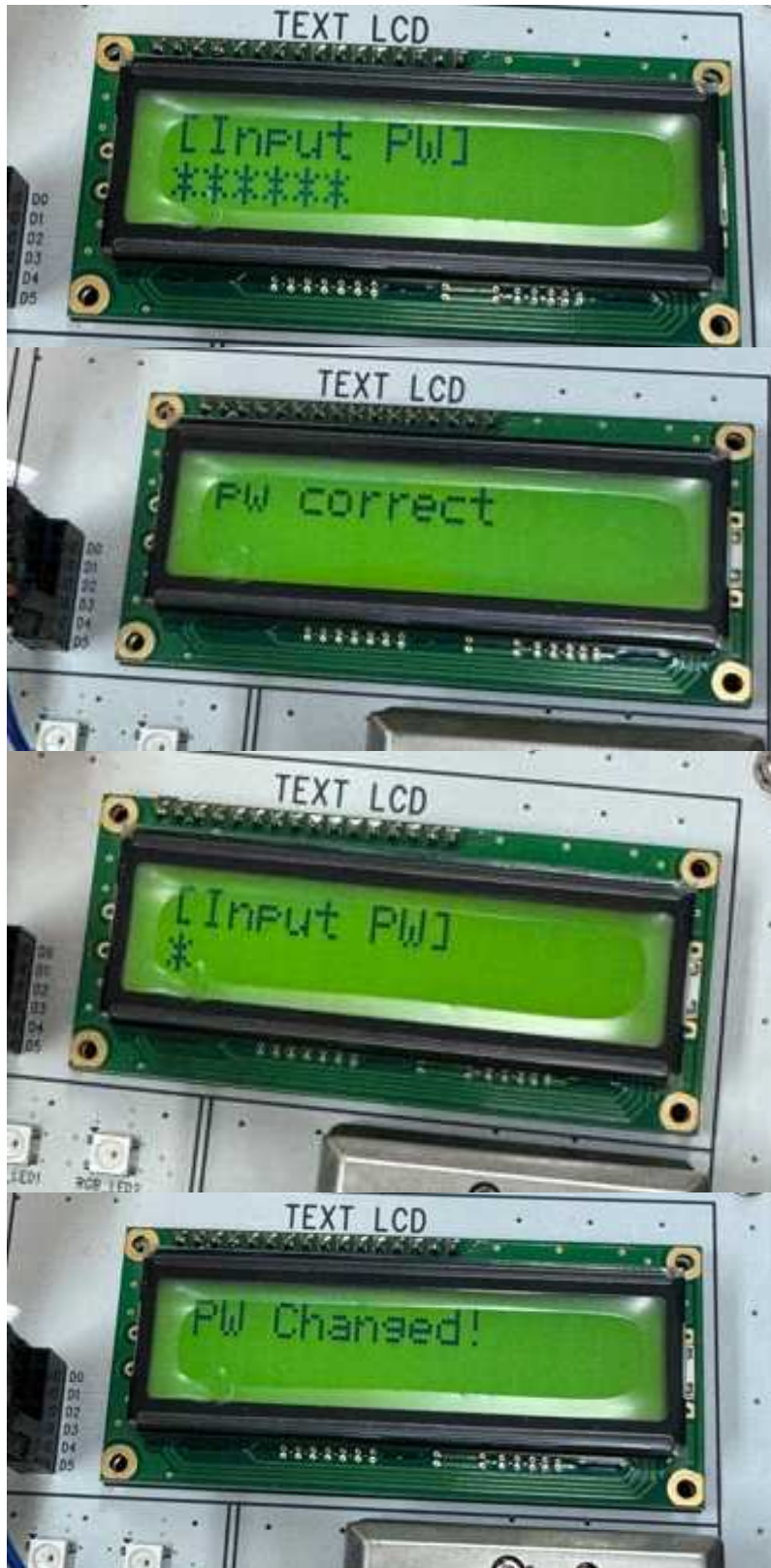


그림 17. 4번 메뉴인 사운드 On/Off 기능



## 8. 평가

### 8.1 정량적/정성적 목표달성도 평가

항목	목표값(내용)	달성률 (%)	비고
1	스마트폰과 금고와 블루투스 통신을 통해 통신하도록 설정	100%	
2	사용자에게 현재 금고의 상태가 어떤지 LCD를 통해 출력	100%	
3	정확하게 지문인식 센서가 동작하도록 실험하여 프로그램 작성	100%	
4	난수로 설정되는 LED와 합쳐지는 암호로 정상적으로 금고 잠금 해제되는지 확인 하여 프로그램 작성	100%	
5	CDS셀, 초음파 센서로 금고가 개방되었을 경우를 잘 인식하도록 프로그램 작성	100%	
6	금고가 개방되었을 경우 카메라로 개방한 인물을 촬영(캡처)할 수 있음	100%	
7	실제 모형을 제작하여 최종 시연 시 발표할 수 있도록 프로젝트 진행	100%	

## 8.2 현실적 제한요소 달성도 평가

현실적 제한 요소들	목표	달성결과
경제	<ul style="list-style-type: none"> <li>- 최소한의 자원을 사용하여 제품의 경제성 고려</li> <li>- 부품 구매 시 계획에 맞춰 구매하여 프로젝트 비용 절감</li> <li>- 개발되는 제품의 가격 경쟁력 확보</li> <li>- 유지 보수 비용이 발생할 수 있음을 고려</li> </ul>	<ul style="list-style-type: none"> <li>- 최소한의 자원을 사용하여 제품의 경제성 확보 완료</li> <li>- 부품 구매 시 계획에 맞춰 구매 완료</li> </ul>
편리	<ul style="list-style-type: none"> <li>- 사용자에게 직관적으로 사용하도록 유도</li> <li>- LCD, LED 등 출력 장치의 명확성 확보</li> <li>- 예상치 못한 기술 장애로 인한 편의상의 문제 발생 가능성 고려</li> </ul>	<ul style="list-style-type: none"> <li>- 최대한 직관적으로 모형 제작</li> <li>- LCD, LED 등 출력 장치 정확한 작동 구현</li> </ul>
윤리	<ul style="list-style-type: none"> <li>- 개발제품이 불법적, 비윤리적으로 사용되지 않도록 진행</li> <li>- 사용자의 생체 정보 등 개인 정보를 수집하는 과정에서 발생할 문제 고려</li> </ul>	<ul style="list-style-type: none"> <li>- 개발제품이 불법적, 비윤리적으로 사용되지 않게 제작</li> </ul>
사회	<ul style="list-style-type: none"> <li>- 스마트 금고 사용 중 발생한 문제에 대한 법적 책임과 규제에 대한 논의</li> <li>- 스마트 금고 기술에 대한 접근성이 낮은 지역이나 계층에 대한 사회적 격차가 발생함을 고려</li> </ul>	<ul style="list-style-type: none"> <li>- 스마트 금고 기술에 대한 접근성을 확대하기 위해 최소한의 부품만을 사용하여 제작하여 비용을 최소화 시킴</li> </ul>

### 8.3 기능적 요구사항 달성도 평가

ID	요구사항	내용	우선 순위	달성률 (%)	비고
R_01	금고 현황 출력	LCD를 통해 금고의 전반적 현황을 파악 가능	2	100%	
R_02	금고 초기 설정	초기 설정은 편리하게 스마트폰을 활용하여 비밀번호 설정 및 재설정이 가능	3	90%	재설정 은 스마트폰 구현X 키패드 사용
R_03	오입력 방지 모드	사용자가 안전하게 비밀번호를 입력이 가능하도록 기능 구현	4	100%	
R_04	문 상태 감지	현재 금고의 문 상태 감지 기능	1	100%	
R_05	이상감지 시스템	금고의 이상이 감지될 경우 사용자에게 알림	5	50%	온습도 - GPIO 포트 부족 가스 - 센서 고장

### 8.4 문제점 및 해결방안

#### 8.4.1 아두이노 상에서 지문인식과 블루투스 통신을 동시에 사용함에 있어서 문제 발생

아두이노 내부에서 스마트폰-아두이노 블루투스 통신과, 지문인식센서-아두이노 시리얼 통신이 같은 포트(9600)를 사용해야 하는데, 동시에 사용이 불가하여 아두이노 내부 코드에서 각 통신을 하는 타이밍을 다르게 프로그래밍하였다. 실제 프로젝트를 진행하면서 원래 제작 의도와 달라진 부분이었지만, 동작 결과 자체는 문제가 없다고 판단하였다.

## 9. 추진체계

본 과제의 추진을 위한 팀은 4인으로 구성되며 다음과 같은 역할을 담당한다. 이론 연구 단계에서부터 과제 종료시까지 협력과 보완을 통해 팀의 목표를 달성할 수 있도록 한다.

팀장 김찬영 : 프로젝트 총괄 및 기획, 암호 관련 함수 프로그래밍

팀원 신대철 : 센서 및 모듈 프로그래밍, 실험 및 성능 개선

팀원 정재호 : 센서 및 모듈 관련 자료 조사, 실험 평가 및 오류 수정

팀원 허진환 : 부품 조사 및 구입, 센서 및 모듈 프로그래밍, 실험 평가

## 10. 설계 추진 일정 : 2023년 11월 15일 ~ 2023년 12월 13일

수행 내용		일정 (1주 단위)			
		1	2	3	4
목표와 기준 설정	<ul style="list-style-type: none"> <li>- 설계목표 설정</li> <li>- 목표 달성 방법에 대한 논의</li> <li>- 소프트웨어 요구사항 파악, 기능 블록도</li> </ul>				
합성	<ul style="list-style-type: none"> <li>- 기능별 구현방법 결정</li> <li>- 적용할 이론 및 기술</li> <li>- 구매 목록 작성</li> </ul>				
분석	<ul style="list-style-type: none"> <li>- 세부 기능 구현 방법 결정</li> <li>- 목표달성가능성 확인</li> </ul>				
제작	<ul style="list-style-type: none"> <li>- 비밀번호 난수 생성 관련 함수 프로그래밍</li> <li>- 실제 동작 관련 함수 프로그래밍</li> <li>- 센서 및 모듈 제어 함수 작성</li> </ul>				
시험/ 평가	<ul style="list-style-type: none"> <li>- 시험 및 검증</li> <li>- Trouble shooting</li> <li>- 센서 및 모듈 동작 점검</li> </ul>				
결과	<ul style="list-style-type: none"> <li>- 결과 보고 및 시연</li> </ul>				

## 11. 결론

내부 이상 감지 시스템을 제외한 스마트 금고의 기본적인 기능을 수행하는 요구사항을 대부분 충족하였으며, wiringPi.h 라이브러리를 이용한 IoT 장치 및 센서들을 해당 요구사항에 맞게 제어하였다. 금고 현황이 Text LCD에 각 상황에 맞추어 알맞게 출력되고 있으며, 적절한 딜레이 시간과 lcd\_clear() 함수를 통하여 사용자가 현 상황을 알아보기에도 적합하게 출력하였다. 또한, 금고 잠금장치인 서보 모터를 제어함에 있어 기존 수업 때 실습했었던 도어락에서는 일정 시간 딜레이를 주고 서보 모터 각도를 원위치시키는 방식으로 도어락 잠금장치를 설계했었지만, 본 프로젝트에서는 최대한 실생활에서 사용할 수 있도록 문이 닫힘을 초음파/조도 센서로 감지하여 문이 완전히 닫혔다고 판단되어질 때 ‘스마트하게’ 닫히도록 설계한 것 같아 실생활에 한 발짝 더 나아간 프로그래밍 실력을 갖추 수 있게 되었다. 또한, 아두이노와 라즈베리파이, 핸드폰을 각각 유기적인 통신으로 연결하여 사용자가 제어하기 쉽게 하였으며 최대한 많은 횟수의 디버깅을 통해 사용자가 불편을 느끼지 않도록 노력하는 과정을 거쳐 보다 사용자 친화적인 IoT 설계에 한 발짝 다가간 느낌이 들었다. 마지막으로, 본 프로젝트를 진행하면서 키트가 한대 뿐이고, 모듈화가 힘든 프로젝트였었지만, 최대한 함수 위주의 프로그래밍을 하여 협업 환경에서의 프로그래밍 실력도 갖추게 되었다.

## 부록

- 가. 프로그램 소스코드 (iot.c, arduino\_iot.ino)
- 나. 프로그램 전체 순서도 이미지 (SVG File)
- 다. [Github Link](#)