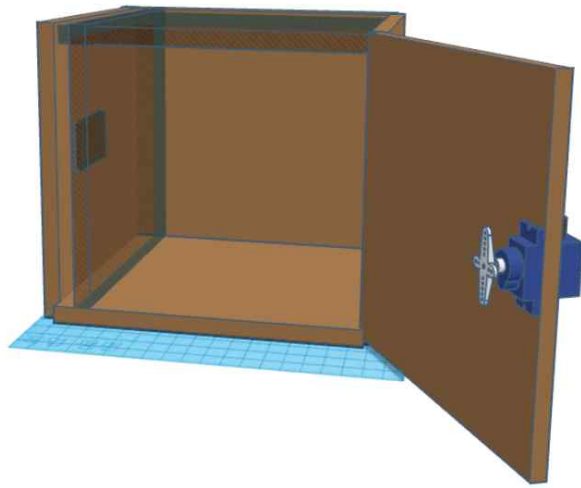


설계도=====



- > 20x20사이즈로 제작예정
- > 서보모터를 문 고리에 장착

열림상태일때



닫힘 상태일때



<서보 모터 구현법 기초>=====

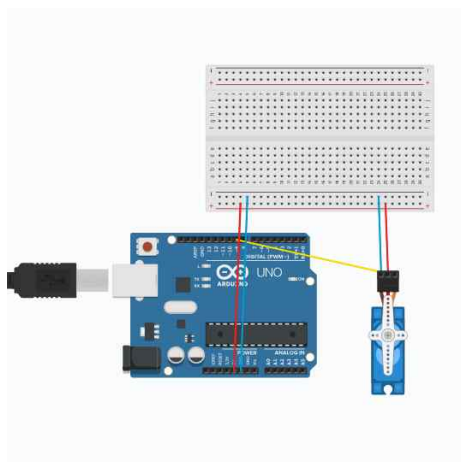
```
#include <Servo.h>  // 서보 라이브러리 추가
```

```
Servo myservo;  // 서보 객체 생성
```

```
int pos = 0;    // 서보모터의 현재 위치
```

```
void setup() {  
  myservo.attach(9);  // 9번 핀에 서보모터 연결  
}
```

```
void loop() {  
  // 잠금  
  for (pos = 0; pos <= 90; pos += 1) { // 0도에서 90도로 서서히 회전  
    myservo.write(pos);                // 서보모터를 pos 각도로 회전  
    delay(15);  
  }  
  delay(1000);                          // 1초 대기  
  
  // 해제  
  for (pos = 90; pos >= 0; pos -= 1) { // 90도에서 0도로 서서히 회전하여 초기 위치로  
    myservo.write(pos);  
    delay(15);  
  }  
  delay(1000);  
}
```



GND-갈색 연결/5V-빨간색 연결/~9-노란색 연결

간단한 회로도

Arduino Uno -- ESP32

5V ----- VCC

GND ----- GND

RX ----- TX0

TX ----- RX0

Arduino Uno -- 서보 모터

5V ----- VCC

GND ----- GND

D9 ----- Control Signal(9번핀)

Arduino Uno -- LED와 저항

D10 --(LED의 긴 다리)--> |LED (+)(빨간색) --(LED의 짧은 다리)| --(저항)-- GND

D11 --(LED의 긴 다리)--> |LED (+)(초록색) --(LED의 짧은 다리)| --(저항)-- GND

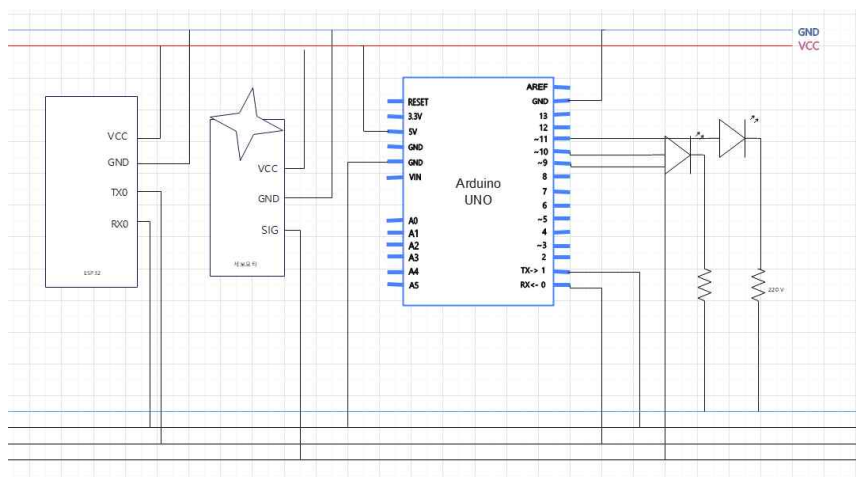
블루투스모듈

5V ----- VCC

GND ----- GND

RX(2번핀) ----- TX0

TX(3번핀) ----- RX0



앱인벤터

앱 인벤터와 서보모터를 직접 시리얼 통신으로만 연결하는 것은 직접적으로는 불가능합니다. 앱 인벤터는 스마트폰에서 실행되는 애플리케이션을 개발하기 위한 플랫폼이기 때문에, 스마트폰의 하드웨어 인터페이스 제한으로 인해 USB 시리얼 통신 같은 직접적인 하드웨어 연결을 지원하지 않습니다. 일반적으로 스마트폰과 외부 하드웨어(예: Arduino와 연결된 서보 모터) 간의 통신에는 다음과 같은 방법이 사용됩니다: 블루투스 통신

```
> AT
OK
> AT+VERSION
HC-06-20210910
> AT+NAMEfighteam
OKsetname
> AT+PIN1234
OKsetPIN
> AT+ROLE=S
OK+ROLE:S
```

아두이노 코드

```
#include <SoftwareSerial.h>
```

```
#include <Servo.h>
```

```
SoftwareSerial btSerial = SoftwareSerial(2, 3); // 2번핀은 TX, 3번핀은 RX에 연결하고
```

```
btSerial 이름으로 소프트웨어 시리얼 통신을 생성
```

```
Servo myServo; // 서보 모터 제어를 위한 서보 객체 생성
```

```
// LED 핀 설정
```

```
int redLedPin = 10; // 빨간색 LED 핀
```

```
int greenLedPin = 11; // 초록색 LED 핀
```

```
void setup() {
```

```
    btSerial.begin(9600); // 블루투스 시리얼 통신 시작
```

```
    myServo.attach(9); // 서보 모터를 디지털 핀 9번에 연결
```

```
    pinMode(redLedPin, OUTPUT); // 빨간색 LED 핀을 출력으로 설정
```

```
    pinMode(greenLedPin, OUTPUT); // 초록색 LED 핀을 출력으로 설정
```

```
}
```

```
void loop() {
```

```
    // 블루투스 시리얼 통신으로 받은 데이터가 있다면
```

```
    if (btSerial.available() > 0) {
```

```
        // 블루투스 시리얼 통신값을 btData 변수에 대입
```

```
        char btData = btSerial.read();
```

```
        if (btData == 'L') { // 'L'를 받았을 때(Lock)
```

```
            myServo.write(90); // 서보 모터를 90도 위치로 회전시켜 문을 닫기
```

```
            digitalWrite(redLedPin, HIGH); // 빨간색 LED 켜기
```

```
            digitalWrite(greenLedPin, LOW); // 초록색 LED 끄기
```

```
        }
```

```
        else if (btData == 'O') { // 'O'를 받았을 때(Open)
```

```
            myServo.write(180); // 서보 모터를 180도 위치로 회전시켜 문을 열기
```

```
            digitalWrite(redLedPin, LOW); // 빨간색 LED 끄기
```

```
            digitalWrite(greenLedPin, HIGH); // 초록색 LED 켜기
```

```
        }
```

```
    }
```

```
}
```

LED초록색->OPEN(이용가능)

LED빨간색->LOCK(잠그기)

앱인벤터 코드

