



Cleanliness

- 은가비

1901717 김민주 1901716 김민경
1901812 최윤희 1901789 이혜린

개발 개요

■ 목적 및 필요성

코로나 19로 인해 손 씻기의 중요성이 더욱 강조되고 있습니다. 그런데 어떠한 방법으로 손 씻기를 해야 하는지에 대해 많이들 자각하지 못하고 있습니다. 그래서 이번 프로젝트를 통해 손 씻기의 중요성과 제대로 된 손 씻기를 사람들이 쉽고 편하게 인식할 수 있게 하였습니다.

■ 과제선정 사유

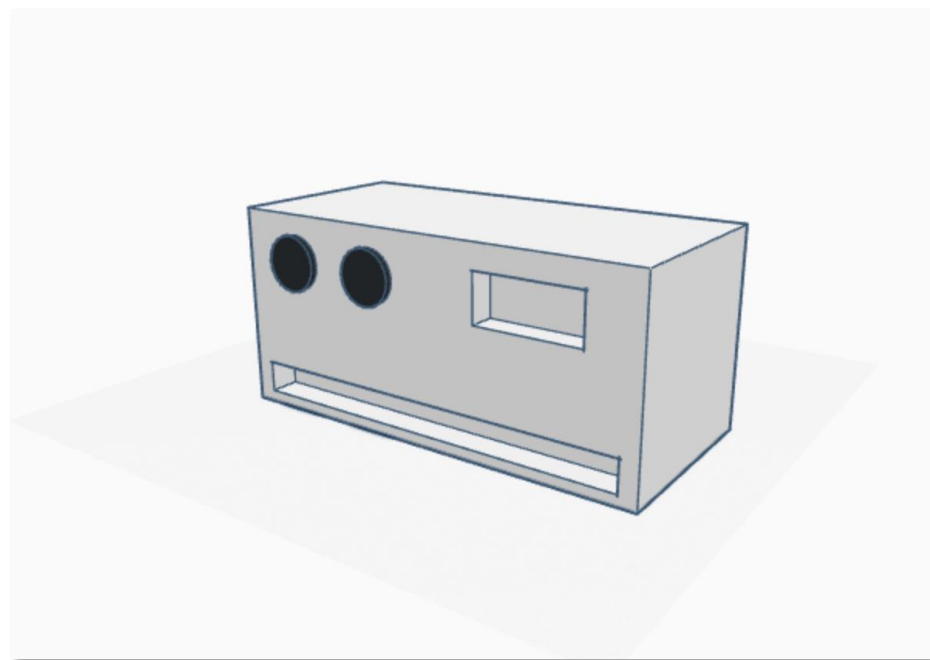
코로나 19로 손씻기의 중요성에 대한 인식은 높아졌으나 손을 씻는 것에만 의의를 두고 30초동안 흐르는 물에 손을 씻는 것을 잘 하지 않고 있습니다. 손 씻기에 관한 어플은 애플 워치와 같은 부가적인 스마트폰 악세사리를 구매해야 하며 애플 워치를 가지고 있는 사람만 사용할 수 있습니다.

그래서 저희는 공용화장실이나 개인 화장실에서 부착하여 그 화장실을 사용하는 사람들 모두 사용할 수 있는 장치를 만들고자 이 프로젝트를 기획하였습니다.

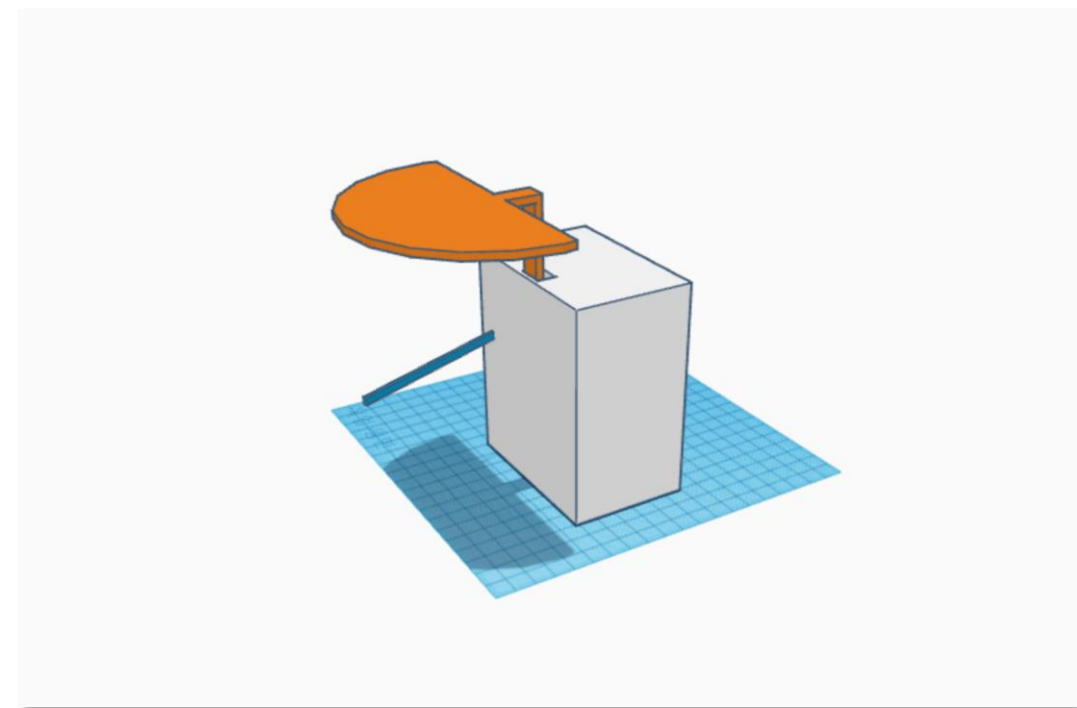


시연

제품



제품1 손씻기 타이머

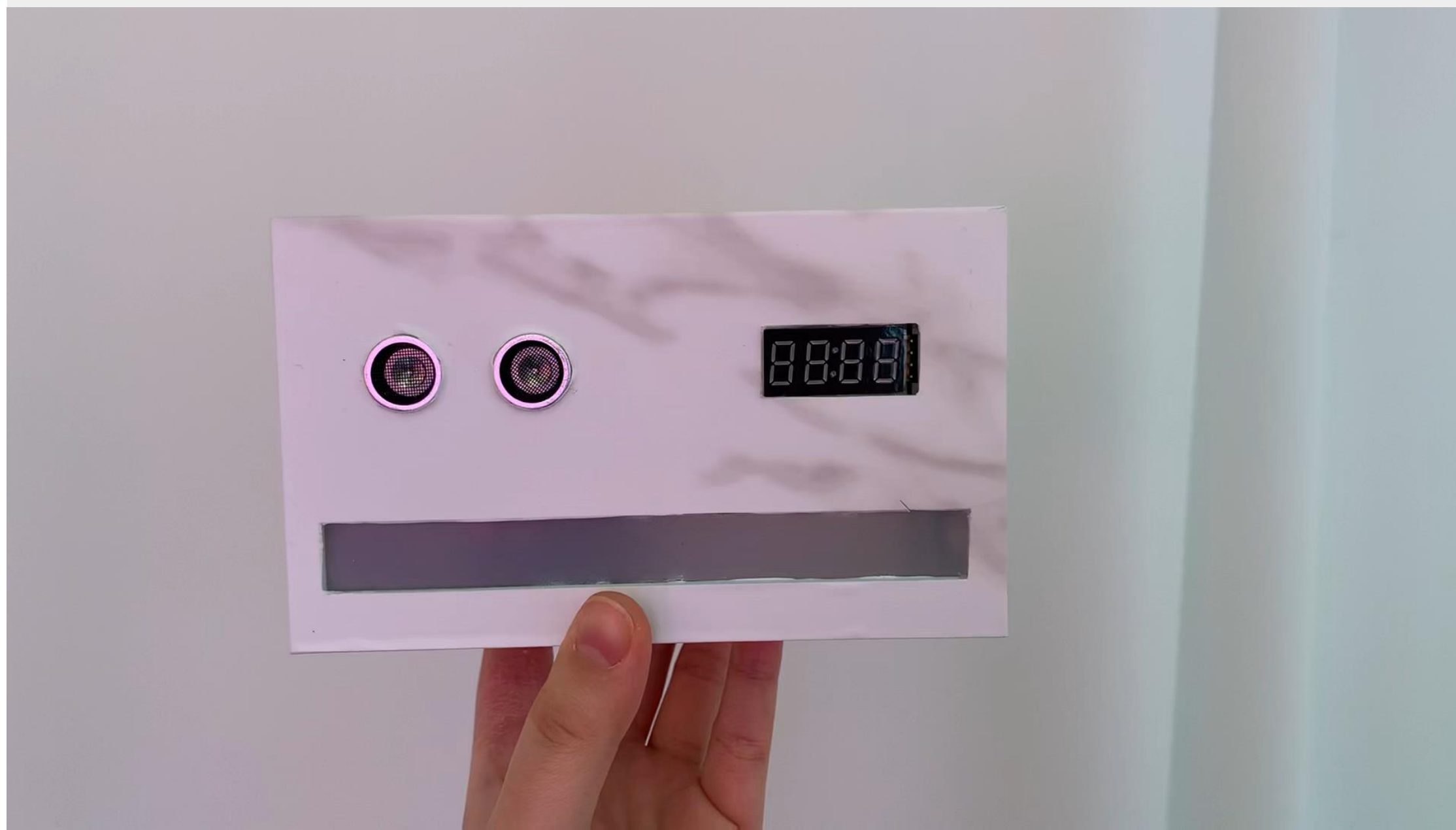


제품2 물 끄기

디자인

앞쪽의 초음파 센서로 사람이
있음을 감지하고, 오른쪽의
소리 센서로 소리가
나고있음을 감지한다.

타이머가 올라가면서
밑쪽에서 LED가 점등되며
시간이 흐름을 보여준다.



디자인(세부)

- 초음파센서

기계 앞에 사람이 있는가를 판단

- 소리센서

기계 앞에서 소리가 나는가를 판단

- LED모듈, LED등

30초의 시간이 흐르는 모습을 보여줌

- 부저

손씻기 실패 시 부저를 울려줌

- 전원선

제품1 (손씻기 타이머)

정면



상



좌측



우측




```

#include <SoftwareSerial.h> //시리얼 통신 라이브러리 호출
#include <Wire.h>
#include <Adafruit_NeoPixel.h>
#include <Arduino.h>
#include <TM1637Display.h>
#define CLK 4
#define DIO 3

Adafruit_NeoPixel strip = Adafruit_NeoPixel(5, 8, NEO_GRB + NEO_KHZ800); //LED strip 8번 핀

SoftwareSerial mySerial(9, 10); //시리얼 통신을 위한 객체선언 t r
String myString=""; //받는 문자열

TM1637Display display(CLK, DIO);
extern volatile unsigned long timer0_millis; //millis 함수 초기화 변수
int Sec1, Sec2, Min1, Min2 = 0;
unsigned long currentTime;
uint8_t data[] = { 0x0, 0x0, 0x0, 0x0 }; // 초기값으로 00:00 셋팅
uint8_t segto;

//출력핀(trig)과 입력핀(echo) 설정
int trigPin = 6;
int echoPin = 7;
int soundPin = A0;
int speakerPin = 5;
int tempo = 200; // duration 옵션 값 설정
int checksound = 0;
int checkdistance = 0;
int use = 0;

int Do = 262;
int Re = 294;
int Mi = 330;
int Fa = 349;
int Sol = 392;
int Ra = 440;
int Si = 494;
int Do2 = 523;

```

```

void setup() {
  Serial.begin(9600); // 시리얼 속도 설정
  pinMode(echoPin, INPUT); // echoPin 입력
  pinMode(trigPin, OUTPUT); // trigPin 출력
  pinMode(soundPin, INPUT);
  pinMode(speakerPin, OUTPUT);
  strip.begin(); //네오픽셀을 초기화하기 위해 모든LED를 off시킨다
  strip.show();
  display.setBrightness(0xF); //최대 밝기 (1~7 밝기 조절)
  mySerial.begin(9600); //블루투스 시리얼 개방
}

```

```

void loop() {

  long duration; // duration : 왕복소요시간
  float distance; // distance : 거리 (HC-SR04는 2cm ~ 4m까지 측정가능)
  int soundValue;

  currentTime = millis() / 1000; // '/1000'최하 단위를 초로 바꾸어 줌
  Sec2 = currentTime % 10; // 초단위 '일의 자리'만 분리하여 저장
  Sec1 = (currentTime / 10) % 10; // 초단위 '십의 자리'만 분리하여 저장
  Min2 = (currentTime / 100) % 10; // 분단위 '일의 자리'만 분리하여 저장
  Min1 = (currentTime / 1000) % 10; // 분단위 '십의 자리'만 분리하여 저장

  soundValue = analogRead(soundPin);
  digitalWrite(trigPin, HIGH); // trigPin에서 초음파 발생(echoPin도 HIGH)
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH); // echoPin 이 HIGH를 유지한 시간을 저장 한다.
  // 단위 : 마이크로초(μs)
  distance = ((float)(340 * duration) / 10000) / 2; // 초음파 속도 : 340 m/s

  Serial.print("소리 : ");
  Serial.println(soundValue);
  Serial.print("시간 : "); //시리얼모니터에 Echo가 HIGH인 시간을 표시
  Serial.print(duration);
  Serial.print("μs, 거리 : "); // 물체와 초음파 센서간 거리를 표시
  Serial.print(distance);
  Serial.println("cm");
  Serial.print("currentTime : ");
  Serial.print(currentTime);
  Serial.print(" checkdistance : ");
  Serial.print(checkdistance);
  Serial.print(" checksound : ");
  Serial.println(checksound);
  Serial.println();
}

```



```

if (use <= 5) {
  if (checkdistance <= 3 && checksound <= 5) {
    if (currentTime == 30) {
      tone(speakerPin, Do, 250);
      delay(200);
      tone(speakerPin, Mi, 250);
      delay(200);
      tone(speakerPin, Sol, 250);
      delay(200);
      tone(speakerPin, Do2, 250);
      delay(500);
      tone(speakerPin, Sol, 250);
      delay(200);
      tone(speakerPin, Do2, 250);
      delay(500);
      colorWipe(strip.Color(255, 0, 0), 100); //빨간색 출력
      colorWipe(strip.Color(255, 255, 0), 100);
      colorWipe(strip.Color(0, 255, 0), 100); //녹색 출력
      colorWipe(strip.Color(0, 255, 255), 100);
      colorWipe(strip.Color(0, 0, 255), 100); //파란색 출력
      colorWipe(strip.Color(255, 0, 255), 100);
      colorWipe(strip.Color(0, 0, 0), 100);
      timer0_millis = 0;
    }
    if (currentTime % 2 == 1) {
      if (currentTime / 6 == 0) strip.setPixelColor(0, 255, 0, 0);
      else if (currentTime / 6 == 1) strip.setPixelColor(1, 255, 255, 0);
      else if (currentTime / 6 == 2) strip.setPixelColor(2, 0, 255, 0);
      else if (currentTime / 6 == 3) strip.setPixelColor(3, 0, 255, 255);
      else if (currentTime / 6 == 4) strip.setPixelColor(4, 0, 0, 255);
    }
    else {
      if (currentTime / 6 == 0) strip.setPixelColor(0, 0, 0, 0);
      else if (currentTime / 6 == 1) strip.setPixelColor(1, 0, 0, 0);
      else if (currentTime / 6 == 2) strip.setPixelColor(2, 0, 0, 0);
      else if (currentTime / 6 == 3) strip.setPixelColor(3, 0, 0, 0);
      else if (currentTime / 6 == 4) strip.setPixelColor(4, 0, 0, 0);
    }
    strip.show();
  }
  else {
    if (currentTime != 0) {
      for (int i = 0; i < 5; i++) {
        strip.setPixelColor(i, 0, 0, 0);
      }
      strip.show();
      checkdistance = 0;
      checksound = 0;
      timer0_millis = 0;
      tone(speakerPin, Sol, 250);
    }
  }
}
}

```

```

else {
  for (int i = 0; i < 5; i++) {
    strip.setPixelColor(i, 0, 0, 0);
  }
  strip.show();
  timer0_millis = 0;
}

data[0] = display.encodeDigit(Min1); // 초시계
data[1] = display.encodeDigit(Min2);
data[2] = display.encodeDigit(Sec1);
data[3] = display.encodeDigit(Sec2);

segto = 0x80 | display.encodeDigit(Min2); // 초시계용 ':'도트
display.setSegments(data);
display.setSegments(&segto, 1, 1);
delay(500);
display.setSegments(data); //도트가 깜빡이게 하기 위해 반복 출력
delay(500);

if (distance > 80) checkdistance++;
else checkdistance = 0;

if (soundValue == 0) checksound++;
else checksound = 0;

if (distance > 80) use++;
else use = 0;

//Serial.println(sensor);
if(currentTime == 30) {
  //30초가 되면
  Serial.println("1");
  mySerial.println("1");
  delay(500);
}

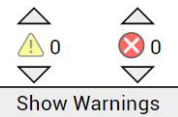
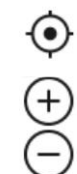
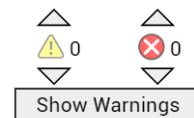
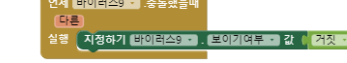
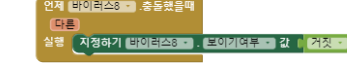
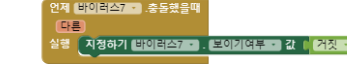
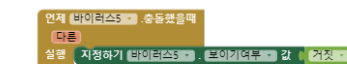
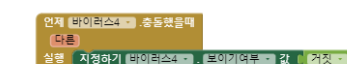
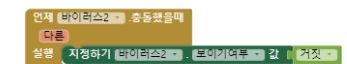
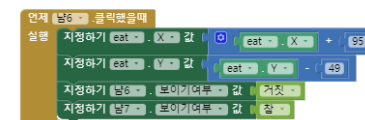
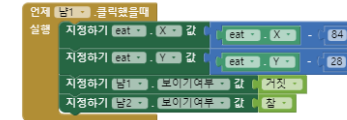
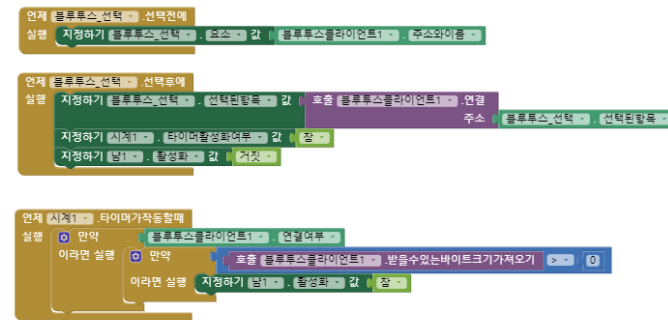
}

}

//NeoPixel에 달린 LED를 각각 주어진 인자값 색으로 채워나가는 함수
void colorWipe(uint32_t c, uint8_t wait) {
  for (uint16_t i = 0; i < strip.numPixels(); i++) {
    strip.setPixelColor(i, c);
    strip.show();
    delay(wait);
  }
}

```

앱인벤터 코드



제품 동작

<< 사람0 / 소리 0 >>

사람이 있고, 소리가 나는 상태라면
타이머가 30초까지 올라가며
LED등이 끝까지 점등 되며
성공했다는 알림음이 나온다

30초간 손씻기에 성공한 경우



※ 동영상은 시연 영상에서 확인하실 수 있습니다.

제품 동작

<< 사람or소리 X >>

사람이나 소리 둘 중 하나가 있다가 없어지는 경우,
삐 소리가 나며 타이머 리셋, LED 리셋

30초간 손씻기를 하지 못한건으로 손씻기 실패

<< 사람X / 소리 0 >>

사람이 없고, 소리가 나는 상태라면 타이머 및
LED등이 작동하지 않고 자동 종료됨

따라서 손씻기를 위해서 세면대에 선 것이 아닌
단순히 화장실을 이용하는 소음에 대한 반응을
하지 않음

<< 사람X / 소리 0 >>

사람은 있지만 소리가 나지 않는 경우 삐 소리가
나며 타이머 리셋, LED 리셋

단순히 거울을 보는 경우 반응을 하지 않음

제품1 (손씻기 타이머)



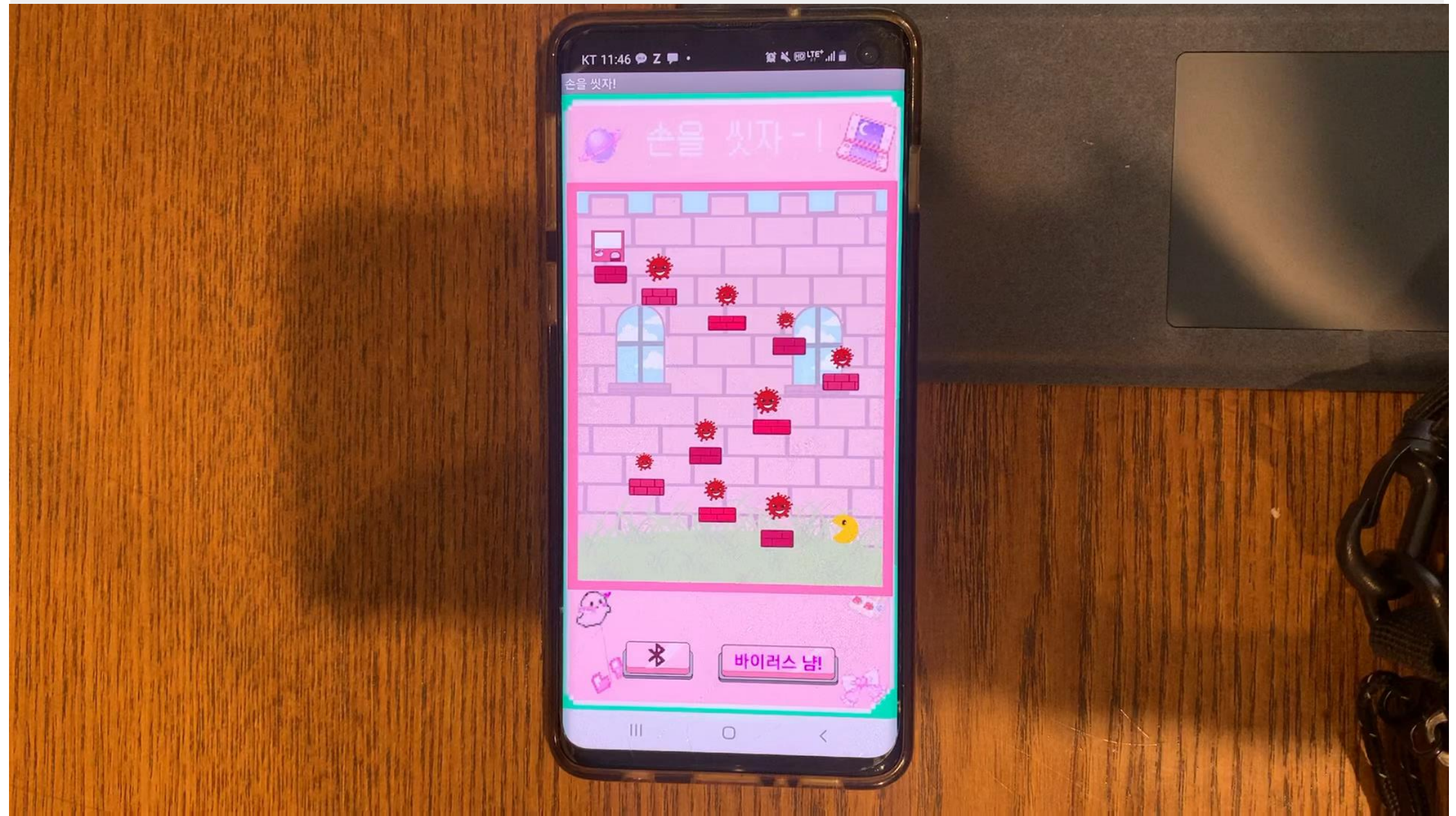
※ 동영상은 시연 영상에서 확인하실 수 있습니다.

어플

손씻기에 성공한 경우 버튼이 1회
활성화 됨

바이러스를 전부 다 해치우게
되면 뿔기로 이동하여 랜덤한
선물을 받을 수 있게 함

제품1 (손씻기 타이머)



※ 동영상은 시연 영상에서 확인하실 수 있습니다.

제품2 (물끄기 기능)



시제품

- 충격센서

물이 켜져있는가를 확인

- 블루투스센서

핸드폰과의 연결

- 서보모터

신호를 받으면 작동 -> 위쪽 누름판이
내려옴

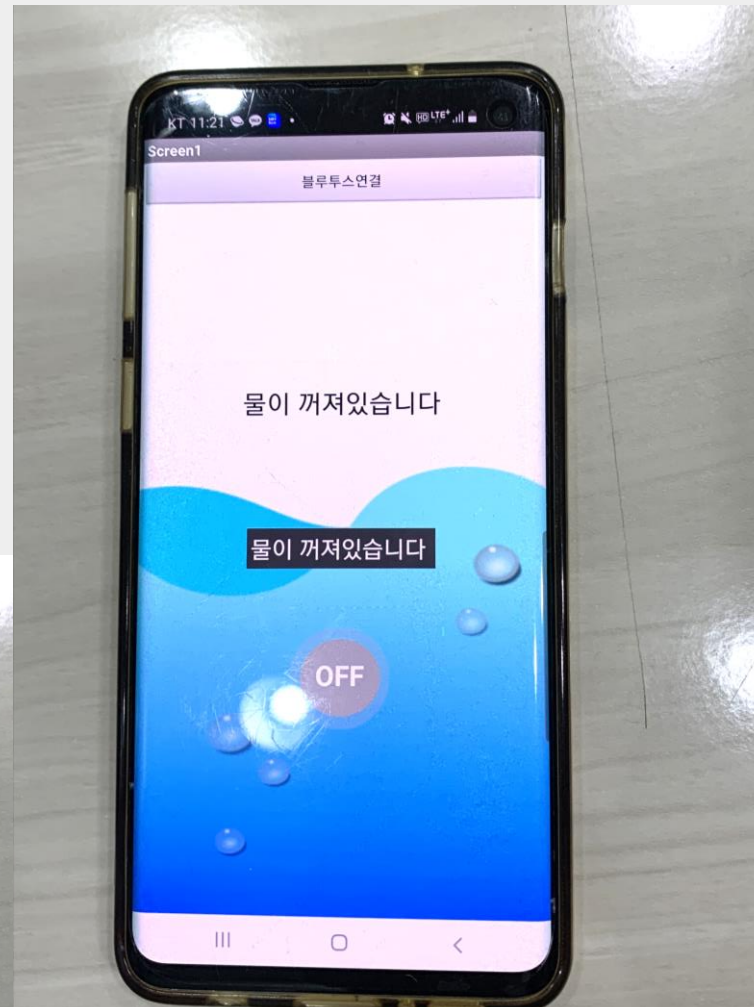
제품2 (물끄기 기능)



물 끄기 화면



버튼 비활성화



비활성화 버튼을 누름

어플

물이 켜져있는 상태로 충격센서애
충격이 가해지면 어플의 물끄기
버튼이 활성화 됨.

물끄기 버튼이 활성화 되어있을
때 버튼을 터치한다면 위에부분이
내려오며 물을 끌 수 있도록 제작.

아두이노 코드

```
final_ | 아두이노 1.8.16
파일 편집 스케치 툴 도움말

final_

#include <SoftwareSerial.h> //시리얼 통신 라이브러리 호출
#include "Servo.h" //서보 라이브러리

Servo myservo; //서보객체
int sensor = 0;
SoftwareSerial mySerial(4, 5); //시리얼 통신을 위한 객체 선언

String myString=""; //받는 문자열

void setup() {
  myservo.attach(6); //서보 시그널 핀설정
  Serial.begin(9600);
  pinMode(2, INPUT); //출력센서
  myservo.write(0); //서보 초기각도 0도 설정
  mySerial.begin(9600); //프로투스 시리얼 개팅
}

void loop() {
  sensor = digitalRead(2);
  while(mySerial.available()) //mySerial 값이 있으면
  {
    char myChar = (char)mySerial.read(); //mySerial int형식의 값을 char형식으로 변환
    myString+=myChar; //수신되는 문자열을 myString에 모두 붙임 (1바이트씩 전송되는 것을 모두 붙임)
    delay(5); //수신 문자열 끝감 방지
  }

  //Serial.println(sensor);
  if(sensor == HIGH ){
    //센서가 작동함것
    Serial.println("1");
    mySerial.println("1");
    delay(500);
  }

  if(!myString.equals("")) //myString 값이 있다면
  {
    Serial.println("input value: "+myString); //시리얼모니터에 myString값 출력

    if(myString=="on") //myString 값이 'on' 이라면
    {
      myservo.write(180); //각도 90도로 움직임
    } else {
      myservo.write(0); //각도 0도로 움직임
    }
    myString=""; //myString 변수값 초기화
  }
}
```

앱인벤터 코드

뷰어

전역변수 만들기 **딜레이** 초기값 0

언제 블루투스목록 선택전에
실행 지정하기 블루투스목록 요소 값 블루투스클라이언트1 주소와이름

언제 블루투스목록 선택후에
실행 지정하기 블루투스목록 선택된항목 값 호출 블루투스클라이언트1 연결 주소 블루투스목록 선택된항목
지정하기 시계1 타이머활성화여부 값 참

언제 시계1 타이머가작동할때
실행 만약 블루투스클라이언트1 연결여부
이러면 실행 만약 호출 블루투스클라이언트1 받을수있는바이트크기가져오기 0
이러면 실행 지정하기 수직배치2 보이기여부 값 참
지정하기 수직배치1 보이기여부 값 거짓
아니라면 지정하기 수직배치2 보이기여부 값 거짓
지정하기 수직배치1 보이기여부 값 참

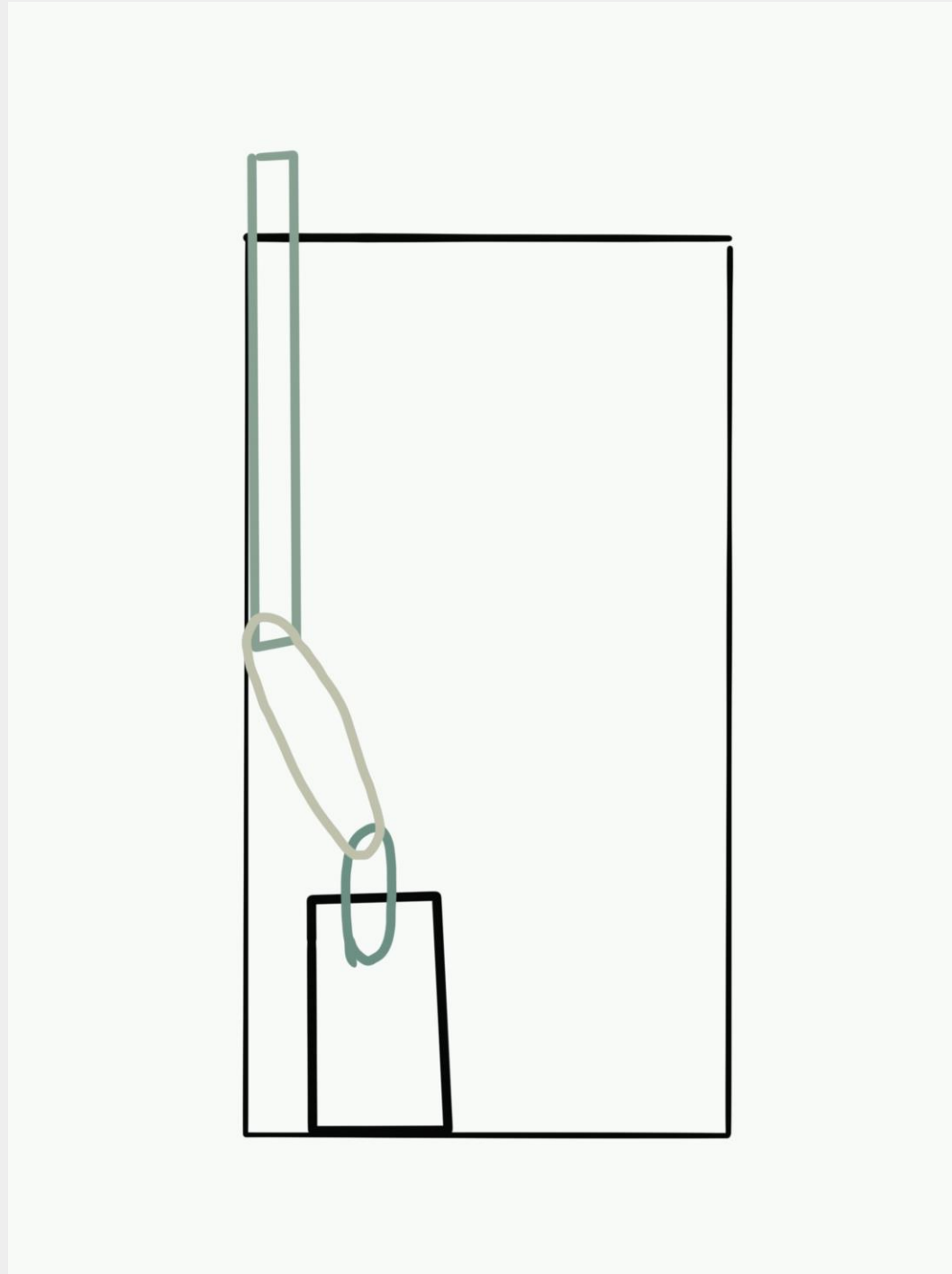
언제 버튼2 클릭했을때
실행 호출 블루투스클라이언트1 텍스트보내기 텍스트 on
지정하기 시계2 타이머활성화여부 값 참

언제 버튼2 터치업했을때
실행 지정하기 수직배치2 보이기여부 값 거짓
지정하기 수직배치1 보이기여부 값 참
지정하기 시계1 타이머활성화여부 값 거짓

언제 시계2 타이머가작동할때
실행 지정하기 전역변수 딜레이 값 가져오기 전역변수 딜레이 + 1
만약 가져오기 전역변수 딜레이 = 2
이러면 실행 호출 블루투스클라이언트1 텍스트보내기 텍스트 off
지정하기 시계2 타이머활성화여부 값 거짓
지정하기 전역변수 딜레이 값 0

언제 버튼1 클릭했을때
실행 호출 물이 꺼져있습니다 경고창보이기 알림 물이 꺼져있습니다

Show Warnings



작동 원리

물을 끌 때 작동되는 원리

서보모터가 180도 움직이며 물을
강한 힘으로 끌 수 있게 제작

서보모터의 각도에 따라 두개의
관절이 함께 움직이며 윗부분이
내려와 물을 끌 수 있음.



어플 작동

물이 켜져있는 상태로 충격센서애
충격이 가해지면 어플의 물끄기
버튼이 활성화 됨.

물끄기 버튼이 활성화 되어있을
때 버튼을 터치한다면 위에부분이
내려오며 물을 끌 수 있도록 제작.

“

감사합니다