

Physical Computing

Final Presentation

2조

2017012033 배수연

2018045587 성수아

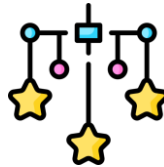
2018045696 이민형

Design Concept / Overall

스마트 유아용 침대



빛의 세기에 따른 차광효과

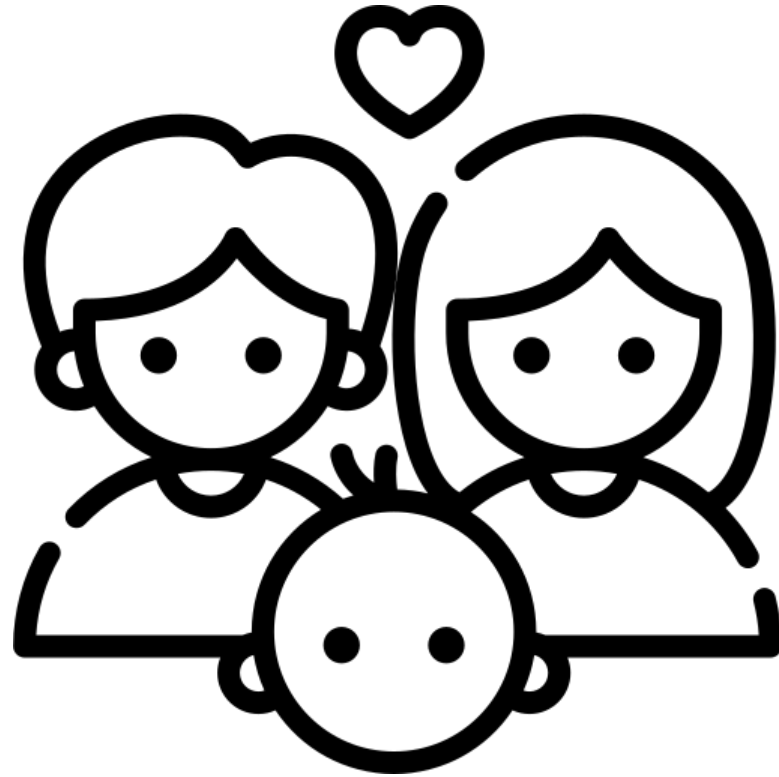


울음소리 들려오면 잔잔한 음악과 무드등



울음소리 들리면 프로세싱 알림과 모빌 원격 제어

User



어린 아이가 있는 가정

Context

아이가 오랫동안 울게 내버려두면 나타나는 결과

부정적인 결과

불안

아이가 울 때 반응이 없으면 아이는 버려진 느낌과 불안한 느낌을 갖게 된다.

질병

우는 아이를 달래주지 않으면 아이의 자율신경계는 시간이 흐를수록 과민해져 자율신경계가 과민한 아이는 천식 등의 호흡기질환, 소화기 장애, 근육 긴장, 두통, 만성피로 등의 질병에 걸리기 쉽다.

아이가 울 때 잘 달래준 결과

긍정적인 결과

스트레스 해소

부모가 아이의 울음에 관심을 기울여 주면 아이의 뇌에 효율적인 스트레스 반응시스템이 형성되어, 성장후에도 스트레스를 잘 견딜 수 있다.

안정감

부모가 우는 아이를 달래면 아이의 옥시토신의 분비를 자극하여 안정감을 느끼고 스트레스 호르몬이 낮아진다.

Context



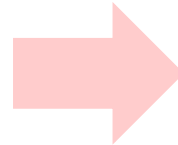
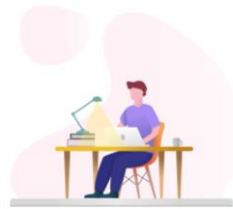
COVID 19로 인한

직장인들의 재택근무

다른 가사일



수면 중

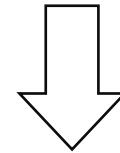


아이에게 온전히 집중할 수 없는
상황 발생

Context



어린이가 있는 **청각장애를 가진 부모**
아이의 울음에 즉각적인 반응이 어려움



UNIVERSAL DESIGN

Context

어린이가 있는 청각장애를 가진 부모

아이의 울음에 즉각적인 반응이 어려움

어린이의 울음에 잘 대처 가능한

스마트 유아용 침대 설계

UNIVERSAL DESIGN

Function



빛의 세기에 따른
차광효과



잔잔한 음악 재생



무드등 ON



아이 상태 알림



모빌 원격 제어

*프로세싱 화면을 통해 아이의 상태 알림

Function

Day



햇빛의 양에 따라 캡의 각도 조절



아이의 울음소리가 들리면
잔잔한 노래 재생



아이의 울음소리에 즉각적인
반응이 없을 시에는 프로세싱으로
아이 상태 알림



보호자의 스마트폰으로 모빌 원격제어 가능

Night



캡의 각도 조절 기능 OFF



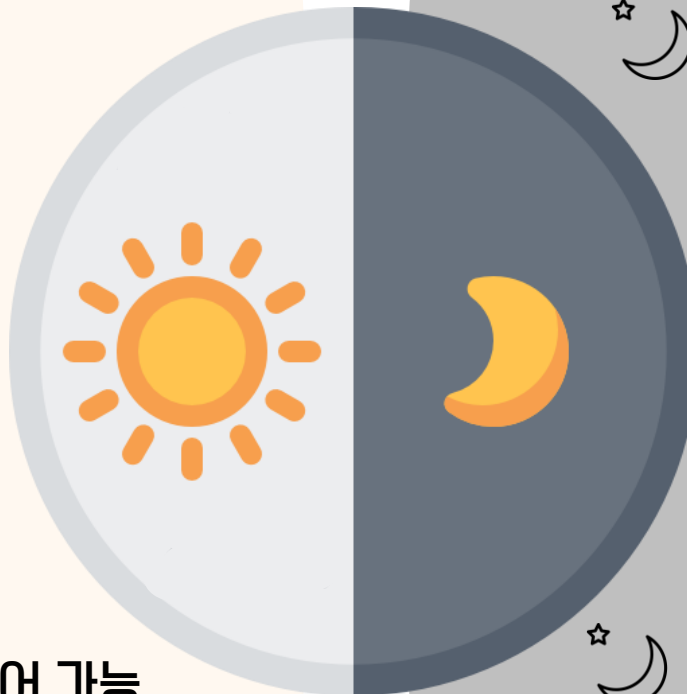
아이의 울음소리가 들리면
무드등 ON



아이의 울음소리에 즉각적인
반응이 없을 시에는 프로세싱으로
아이 상태 알림



보호자의 스마트폰으로 모빌 원격제어



Input & Output

Input

Output

조도 값



햇빛의 양에 따른 캡의 각도

일정 조도값 이상, 아이울음소리



잔잔한 노래소리

일정 조도값 이하, 아이울음소리



무드등 ON

아이 울음소리, 초음파센서 일정거리 이상



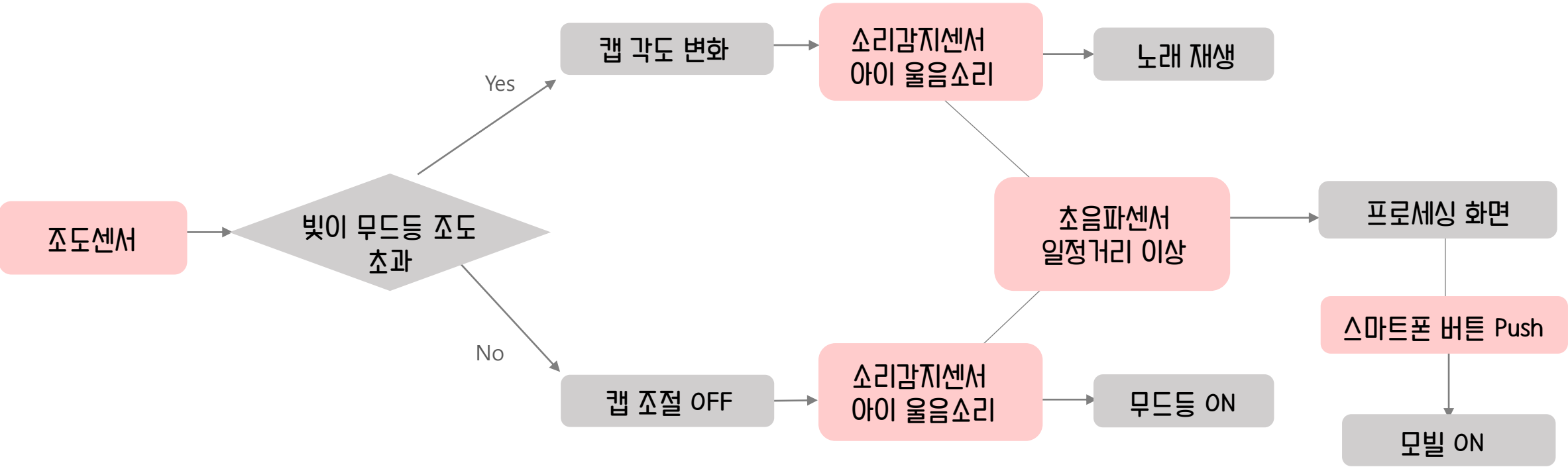
프로세싱 알림

스마트폰 어플 속 버튼

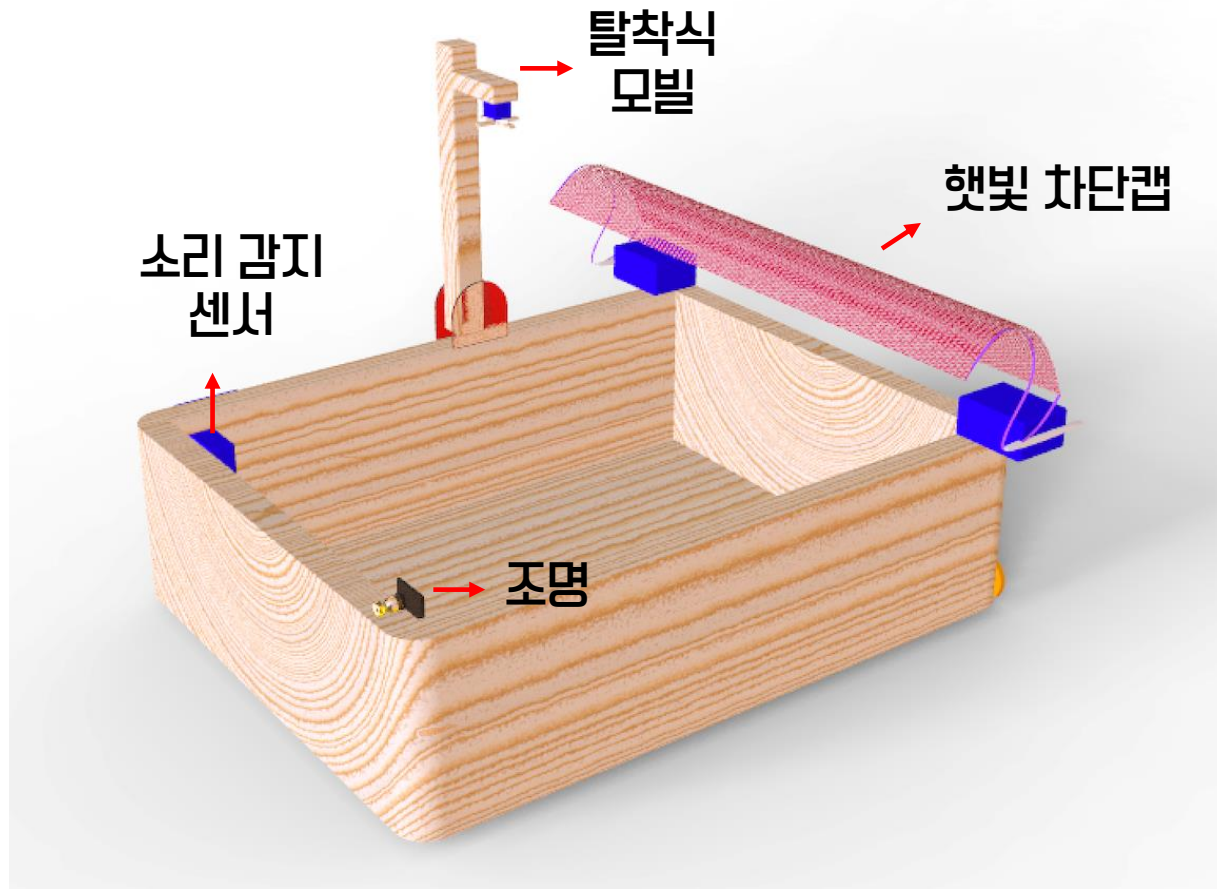


모빌 작동

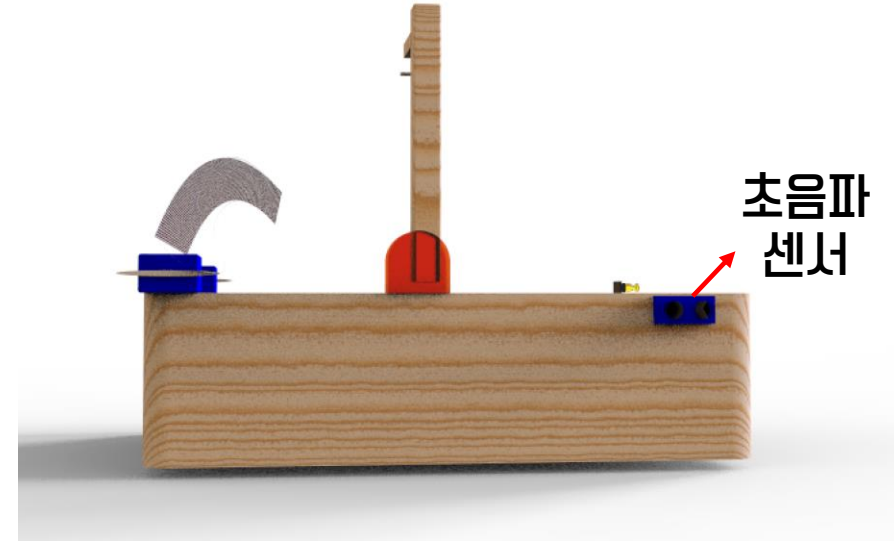
Interaction Flow Chart



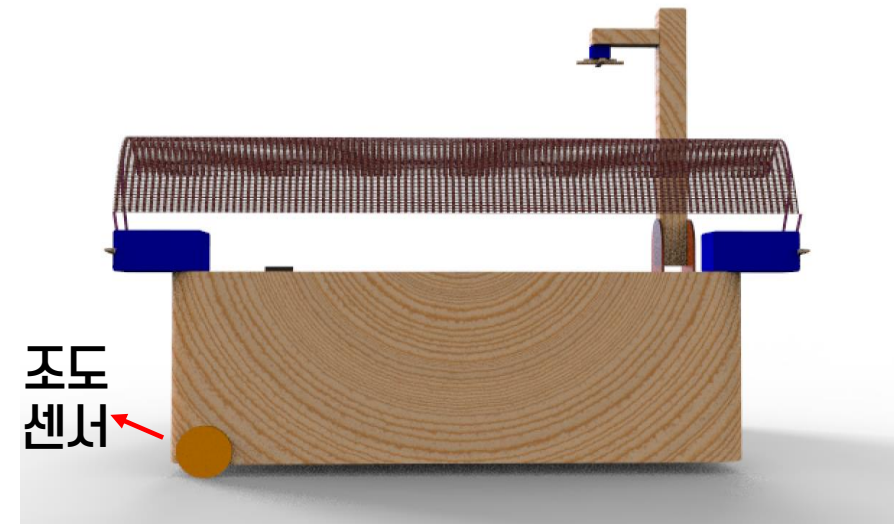
Prototype



Perspective



Left



Rear

Implementation

빛의 세기에 따른 차광효과

조도 센서를 통해 빛의 세기 값을 받고,
서보모터 2개를 이용하여 캡의 각도 조절

```
#include <Servo.h>
Servo myservo;
Servo myservo1;

int lightsensor = A0;

void setup() {
  Serial.begin(9600);
  myservo.attach(7); //캡1
  myservo1.attach(10); //캡2
}

void loop(){
  int light_default=analogRead(lightsensor);

  int ledlight=map(light_default,400,850,0,255);

  cap_move();

  void cap_move(){
    int light = analogRead(lightsensor);
    int a = map (light, 400, 850, 90, 140);
    myservo.write(a);
    delay(15);
    myservo1.write(200-a);
    delay(15);
  }
}
```

Implementation

빛의 세기에 따른 차광효과

- ✓ 최대 밝기 : 850
 - > 서보모터 각도 : 90 도
- ✓ 최소 밝기 : 400
 - > 서보모터 각도 : 140 도
- * 400 이하 값 > 밤



Implementation

무드등 ON

조도센서를 통해 빛의 세기가 일정 값 이하
소리감지 센서를 통해 아이 울음소리를 인식
LED 불 들어옴

```
int soundSensor = A1;  
int lightsensor = A0;  
int threshold = 400;
```

```
int r = 6;  
int g = 5;  
int b = 3;  
int sound = 0;
```

```
void setup() {  
  Serial.begin(9600);  
  pinMode(soundSensor, INPUT);}
```

```
void loop() {  
  int light_default=analogRead(lightsensor);  
  int ledlight=map(light_default,400,850,0,255);  
  sound = analogRead(soundSensor);
```

```
  if(sound > threshold) {  
    check_ultra(distance);  
    if(ledlight>100){  
      lightOn(0, 0, 0);  
    }  
    else{  
      lightOn(255, 153, 0);  
    }  
  }  
}
```

```
void lightOn(int red, int green, int blue){  
  analogWrite(r, red);  
  analogWrite(g, green);  
  analogWrite(b, blue); }
```

Implementation

아이의 상태 알림

소리감지 센서를 통해 아이 울음소리를 인식
초음파센서에 일정거리 안에 물체가 인식 안됨
Processing을 통해 알림 구현

Arduino

```
const int pingPin1 = 8; //trig
const int pingPin2 = 9;

const unsigned int BAUD_RATE=9600;
long duration;
unsigned long distance=0;

int soundSensor = A1;
int lightsensor = A0;
int threshold = 400;

int sound = 0;

void setup() {
  Serial.begin(9600);
  pinMode(soundSensor, INPUT);
  pinMode(pingPin2, INPUT);
  pinMode(pingPin1, OUTPUT); }
```

```
unsigned long
microseconds_to_cm(const
unsigned long microseconds){
  return microseconds/29/2;
}

void loop() {
  sound = analogRead(soundSensor);

  digitalWrite(pingPin1,LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin1,HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin1,LOW);
  duration = pulseIn(pingPin2, HIGH);

  distance=microseconds_to_cm(duration);}

void check_ultra(int distance){
  Serial.println(distance);
}
```


Implementation

아이의 상태 알림

소리감지 센서를 통해 아이 울음소리를 인식
초음파센서에 일정거리 안에 물체가 인식 안됨
Processing을 통해 알림 구현

Processing

```
import processing.serial.*;

Serial myPort;
PImage img1;
PImage img2;
float a;

void setup(){
  size(400, 400);
  img1 = loadImage("pic.png");
  img2 = loadImage("white.png");
  myPort = new Serial(this, Serial.list()[3], 9600);
  myPort.bufferUntil('\n');
}

void draw(){
  background(255);
  /*image(img2,0,0);
  tint(255,a);
  image(img1,0,0);*/

  if(a > 50){
    image(img1,0,0);
  }
  else{
    image(img2,0,0);
  }
}

void serialEvent(Serial myPort){
  String inString =
  myPort.readStringUntil('\n');

  if(inString != null){

    inString = trim(inString);

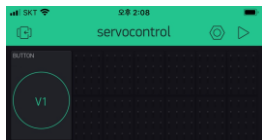
    float inByte = float(inString);
    a = map(inByte, 0, 50, 0, 255);
  }
}
```

Implementation

모빌 원격 제어

WiFi보드를 이용해 스마트폰 어플과 모빌 연결

스마트폰 어플 속 버튼을



이용해 모빌(서보모터) 원격제어

```
#define BLYNK_PRINT Serial
#include <Blynk.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Servo.h>
```

```
char ssid[] = "HY-DORM5";
char pass[] = "residence";
```

```
char auth[]
="_qwGSqDQeEwQ-
TCVUxL1P5QpF0qwPD0Q";
Servo servo;
int minAngle=0;
int maxAngle=180;
```

```
BLYNK_WRITE(V1)
{
  int pinData=param.asInt();
  if(pinData==HIGH)
  {
    servo.write(maxAngle);
    delay(30);
  }
  else if(pinData==LOW)
  {
    servo.write(minAngle);
    delay(30); }
}

void setup()
{
  // Debug console Serial.begin(115200);
  Serial.begin(115200);
  servo.attach(D6);
  Blynk.begin(auth, ssid, pass);
  servo.write(minAngle);
}

void loop()
{ Blynk.run(); }
```

Implementation

- ✓ 아이 상태 알림
- ✓ 무드등
- ✓ 모빌 원격 제어



Implementation

잔잔한 음악 재생

조도센서를 통해 빛의 세기가 일정 값 이상
소리감지 센서를 통해 아이 울음소리를 인식
Mp3 shield를 이용하여 잔잔한 음악 재생



```

#include <SPI.h>
#include <SdFat.h>
#include "SdFatUtil.h"
#include <SFEMP3Shield.h>

SdFat sd;
SFEMP3Shield MP3player;
union twobyte mp3_vol;

int soundSensor = A1;
int lightsensor = A0;
int threshold = 10;
int sound = 400;

void setup(){
  Serial.begin(9600);
  pinMode(soundSensor, INPUT);

  sd.begin(SD_SEL, SPI_HALF_SPEED);
  MP3player.begin();
}

void loop(){
  // Serial.println(analogRead(soundSensor));
  int light_default=analogRead(lightsensor);
  // Serial.print("led : ");
  // Serial.println(light_default);
  int ledlight=map(light_default,400,850,0,255);
  sound = analogRead(soundSensor);
  mp3_vol.word = MP3player.getVolume();
  mp3_vol.byte[1] = 2;
  MP3player.setVolume(mp3_vol.byte[1], mp3_vol.byte[1]);

  if(sound > threshold) {
    if(ledlight>100){
      MP3player.playTrack(1);
    }
    else{
      MP3player.stopTrack();
    }
  }
  else{
    MP3player.stopTrack();
  }
  parse_menu(Serial.read()); }

```

```

void parse_menu(byte key_command) { // 시리얼 창에서 키를 입력하면 조정되는 부분들 (볼륨, 일시정지 등)
  if(key_command == 's') {
    Serial.println(F("Stopping"));
    MP3player.stopTrack();
  }

  else if((key_command == '-') || (key_command == '+')) {
    // union twobyte mp3_vol; // create key_command existing variable that can be both word and double byte of
    // left and right.
    // mp3_vol.word = MP3player.getVolume(); // returns a double uint8_t of Left and Right packed into int16_t

    if(key_command == '-') { // note dB is negative
      // assume equal balance and use byte[1] for math
      if(mp3_vol.byte[1] >= 254) { // range check
        mp3_vol.byte[1] = 254;
      } else {
        mp3_vol.byte[1] += 2; // keep it simpler with whole dB's
      }
    } else {
      if(mp3_vol.byte[1] <= 2) { // range check
        mp3_vol.byte[1] = 2;
      } else {
        mp3_vol.byte[1] -= 38;
      }
    }
    // push byte[1] into both left and right assuming equal balance.
    MP3player.setVolume(mp3_vol.byte[1], mp3_vol.byte[1]); // commit new volume
    Serial.print(F("Volume changed to -"));
    Serial.print(mp3_vol.byte[1]>>1, 1);
    Serial.println(F("[dB]"));
  }

  else if(key_command == 'p') {
    if( MP3player.getState() == playback) {
      MP3player.pauseMusic();
      Serial.println(F("Pausing"));
    } else if( MP3player.getState() == paused_playback) {
      MP3player.resumeMusic();
      Serial.println(F("Resuming"));
    } else {
      Serial.println(F("Not Playing!"));
    }
  }
}

```


Implementation

잔잔한 음악 재생

✓ 아기 울음소리 감지

-조도센서 빛 0 > 노래 0

-조도센서 빛 X > 노래 X



THANK YOU