



# 大学生の 一人暮らしを明るく

7班

あなたが一人暮らしで**さみしい**と感じるときはいつですか？





私たちが注目したこと

一人暮らしでの不安の解消



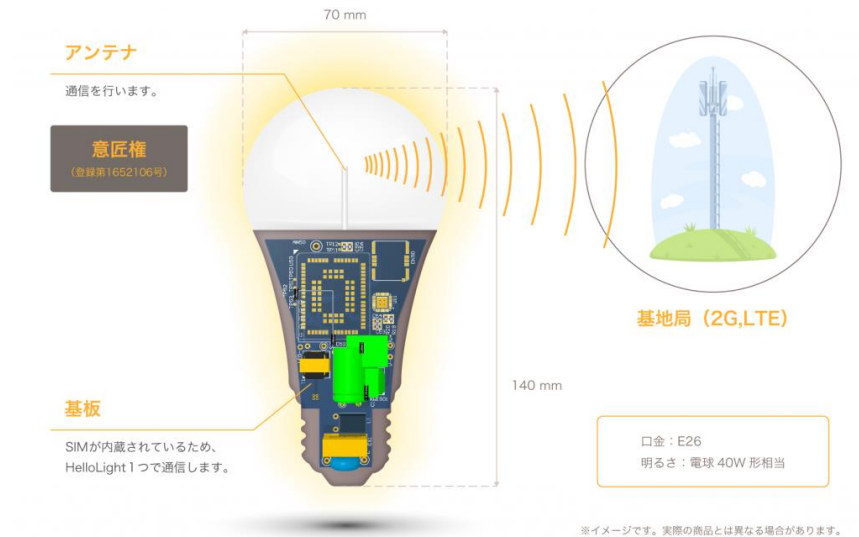
# なぜ一人暮らしに着目をしたのか？

## 現在のIoTの実例

高齢者の一人暮らしを見守るライト



大学生の一人暮らしにも応用できるのではないかと



↑ Hello Light

参考：

<https://soracom.jp/store/13804/>

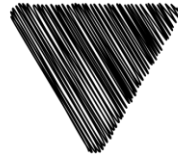
# 一人暮らしによる問題

一人暮らしへの寂  
しさに耐えられる  
か

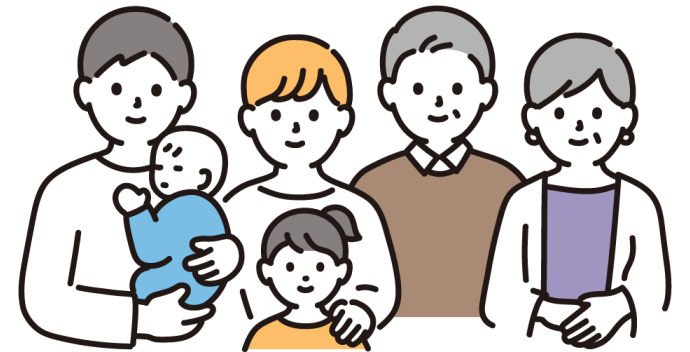
防犯に対する不安

一人でやらなけれ  
ばならないという  
ストレス

一人暮らしのさみしさ



家族を感じられるようにしたい！！



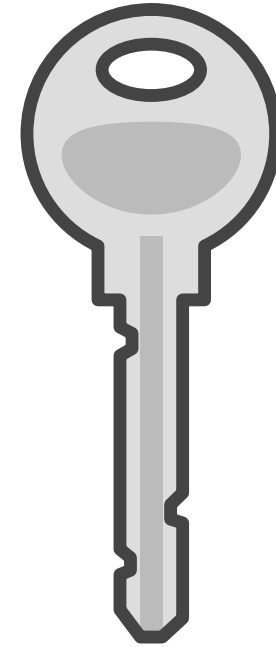
# プロトタイプ

## • 製品の説明

- キースタンドに家の鍵を差す
- キースタンドの中にある**光センサー**が反応
- その光センサーによって家族の録音した音声が出る

# プロトタイプ 製品のねらい

- **安心感**を生み出す
- **防犯**対策
  - 鍵の紛失防止





# プロトタイプ

## デザイン

家族を感じられるように**部屋に馴染んで**リアルさを追求

光センサー→キースタンド

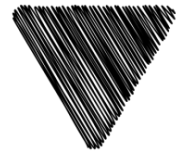


音声出力→写真立て



# IoTの要素

光センサーの数値を読み取って音声が出力される



遠隔で操作ができる  
データを収集し、後にAIに利用できる

# プログラミングの仕組み

## 1つ目

- 調べたところ、4秒以下の音声ファイルなら再生できるはずということが分かったので、4秒以下の音声を録音してarduinoに対応させるためにC言語に変換してヘッダファイルとして保存。
- インターネットのコードを参考に右の写真にあるように
- プログラムを組み、そこに音声ファイルのヘッダファイルを入れるて実行。

## 2つ目

- 音声ファイルをC言語に変換したものをそのままプログラムの中に書き込み実行。



```
shun | Arduino 1.8.19 (Windows Store 1.8.57.0)
ファイル 編集 スケッチ ツール ヘルプ
shun shun.h
#include <shun.h>
prog_uchar KUGI[] PROGMEM = {
  127,
  127,
  .
  . (8kHzサンプリング8bitのPCMデータを入れます)
  . |
  127,
  127
};
// *****
void setup() {
// ***-----***
// *** D03 OUTPUT PWM設定

// ***-----***
pinMode( 3,OUTPUT); // *** D03 PWM OUTPUT

TCCR2A = 0b00100011; // *** 高速PWM設定

TCCR2B = 0b00011001;
OCR2A = 255;

// *** 周期

// *** lcount=0.0625us
// *** 0.0625*256 -> 16us

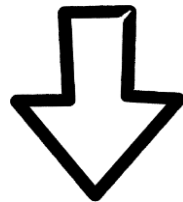
// *** -> 62.5kHz
OCR2B = 0; // *** Duty 0 - 255

// ***-----***
}
// *****
void loop() {
// ***-----***
int i;
// ***-----***
for(i=0; i<12000; i++){
// ***-----***
OCR2B = pgm_read_byte_near(&KUGI[i]);
delayMicroseconds(125); // *** 8kHzで再生

// ***-----***
}
// ***-----***
}
//
```

# 失敗と転換

1. 音声ファイルをC言語ヘッダファイルに書き換えてArudinoを起動するプログラムに読み込ませて音声を流そうとしたが、ヘッダファイルが読み込まれずできなかった。
2. ヘッダファイルをArudinoに直接入れてみたが一瞬音があっただけで音声として流れ  
なかったため失敗



この失敗からデザインはそのままにして録音した音声ではなく  
音階から作る電子音なら流せるのではないかと考えた。

# アイデアの転換

家族を感じさせるもの



一人暮らしを**楽しませる**もの

# 新プログラムの仕組み

- 明るさセンサーのプログラムとファミマの音声を出すプログラムをArduinoで作る。
- この2つのプログラムをPythonを使って連動させる。(明るさセンサーの数値が400以上で1、それ以下だった場合は0と設定)

```
if int_data>400:
    to_arduino = '1'
else:
    to_arduino = '0'
```

# 音階の設定

```
#define BEATTIME 310
#define PIN 11

#define LA 880
#define FA 740
#define MI 659
#define RE 587
#define DLA 440SS
```

使用する音階の周波数を指定

- 参考サイト

[音階周波数 \(tomari.org\)](http://tomari.org)

BEATTIMEで音の長さを調節

# 新プログラムの仕組み

- ・ファミマのプログラムの方でIF文を使い、1が送られてきたら音声になるようにする。

- ・最後に「EXIT」を入れることでプログラムを一回で終わらせるようにした。（鍵を指したままでも音声はなり続けない）

```
        // exit the loop
        exit(0); //0 is required to prevent error.
    }
    else if (from_pc=='0'){
        pinMode(11,LOW);
    }
```

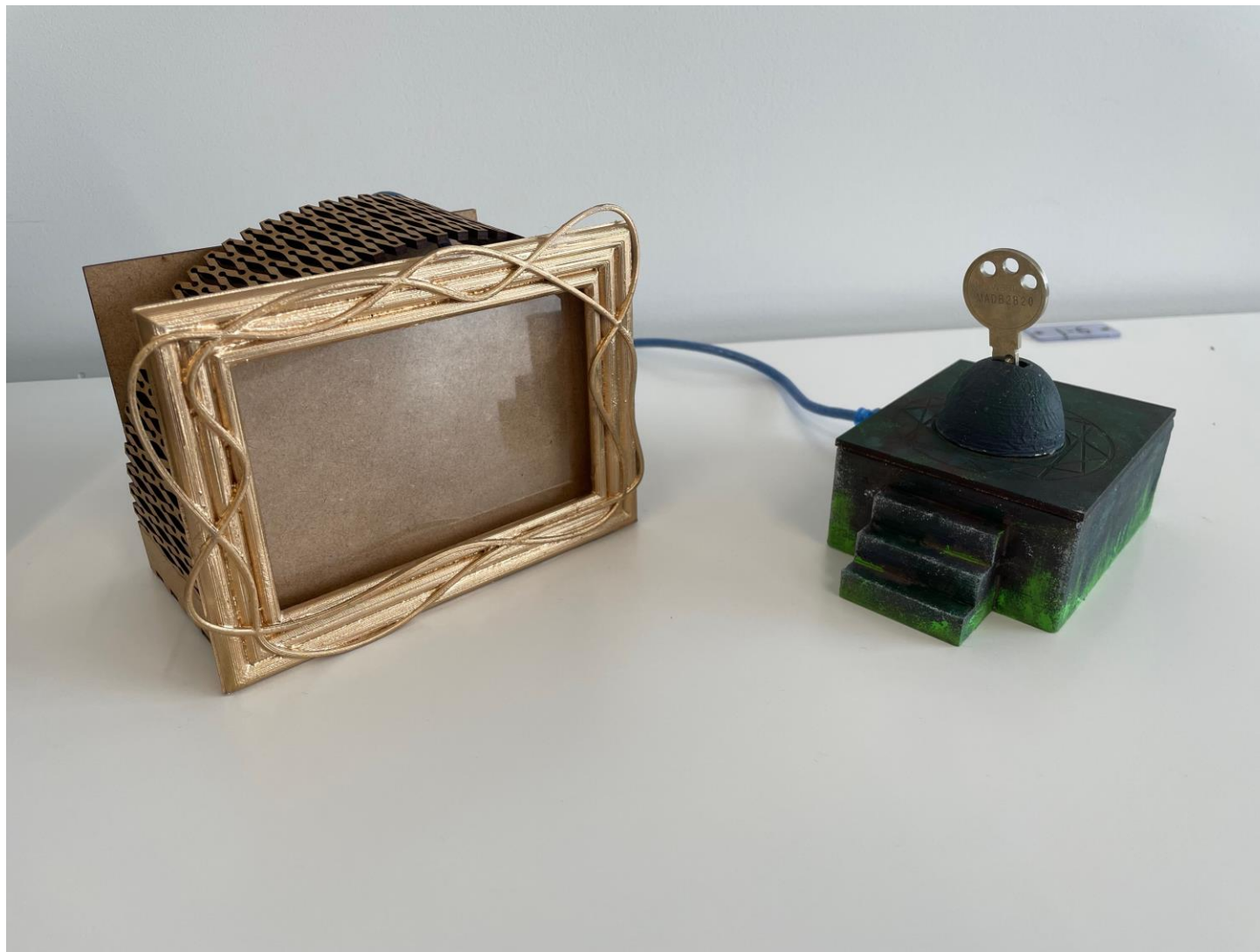
```
void loop() {

    if(Serial.available() > 0){
        from_pc = Serial.read();
    }

    //1が送られてきたらLEDを点灯、0だったら消灯
    if (from_pc=='1'){
        pinMode(11,HIGH);
        //ファ#レラレミララミファ#ミラレ
        tone(PIN,FA,BEATTIME); // ファ#
        delay(BEATTIME);
        tone(PIN,RE,BEATTIME); // レ
        delay(BEATTIME);
        tone(PIN,DLA,BEATTIME); // dラ
        delay(BEATTIME);
        tone(PIN,RE,BEATTIME); // レ
        delay(BEATTIME);
        tone(PIN,MI,BEATTIME); // ミ
        delay(BEATTIME);
        tone(PIN,LA,BEATTIME); // ラ
        delay(BEATTIME*2);
        tone(PIN,DLA,BEATTIME); // dラ
        delay(BEATTIME);
        tone(PIN,MI,BEATTIME); // ミ
        delay(BEATTIME);
        tone(PIN,FA,BEATTIME); // ファ#
        delay(BEATTIME);
        tone(PIN,MI,BEATTIME); // ミ
        delay(BEATTIME);
        tone(PIN,DLA,BEATTIME); // dラ
        delay(BEATTIME);
        tone(PIN,RE,BEATTIME); // レ
        delay(BEATTIME);
    }
}
```



# 写真



# 製品の効果

---

- ・一人暮らしを安心感があり、楽しいものにする
- ・音が聞こえることによる多少の**防犯対策**
- ・音楽による**非日常感**
- ・家の鍵の喪失防止

# 今後の展望

---

- ・ IOT要素の追加
- ・ 防犯対策の強化
- ・ 当初の計画である家族とつながる部分のブラッシュアップ

ご清聴  
ありがとうございました