

数理工学の泉

Demo

Outline

1. Demonstration
2. Introduction
 - a. Raspberry Piとは？
 - b. インスタで映えるための数理工学
3. Method
 - a. ハードウェアを3分クッキング
 - b. 音声処理
 - c. 光の制御
 - d. 水の制御
4. Discussion
 - a. もっとキラキラしたい！

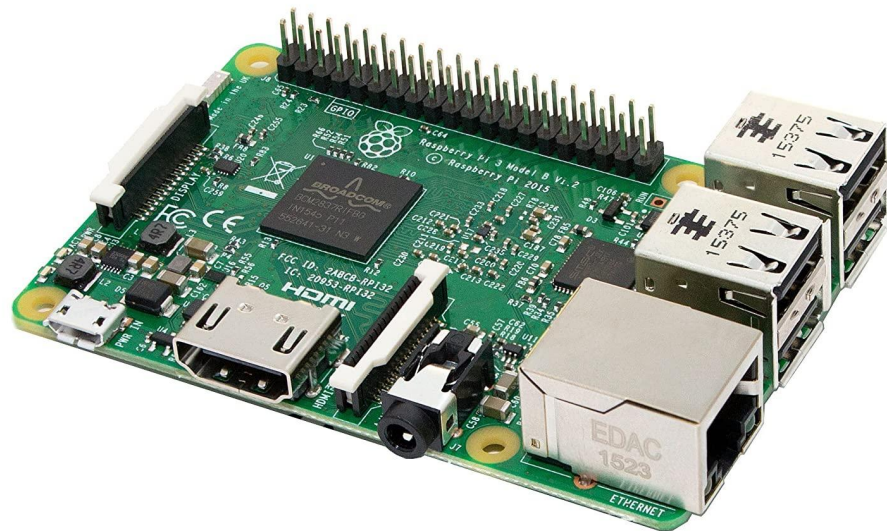
Introduction

Raspberry Piは小型のコンピュータ

材料はこちら

- CPU1個
- USBポート.....4個
- GPIOピン.....26ピン
- シリアル通信.....USART, SPI, I2C
- 無線通信.....WiFi, Bluetooth
- etc...

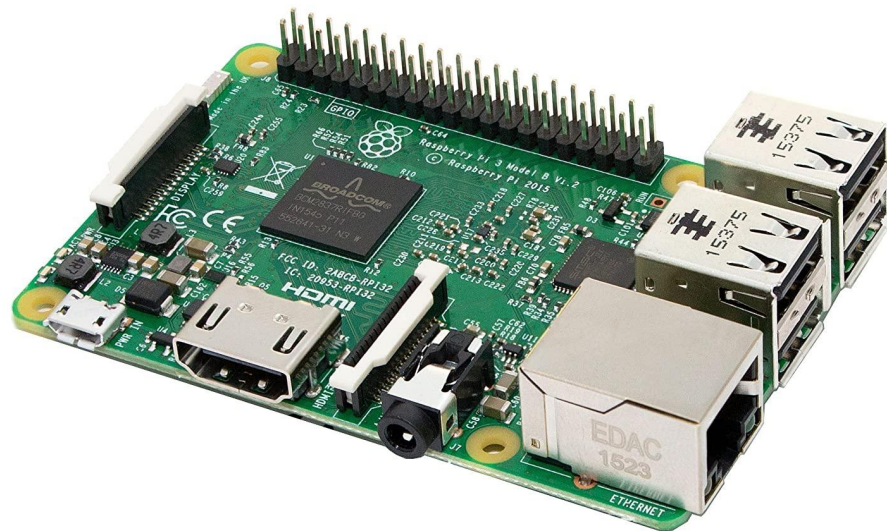
Raspberry Pi



Raspberry Piは電子工作向き 得意なこと

- GPIO多数！
 - 多数のLEDを制御可能
- 充実のシリアル通信機能！
 - 豊富なセンサーを使用可能
- WiFi 使用可能！
 - IoT家電も作れる

Raspberry Pi



Mathematical Engineering for Instagram

- せっかくならハードウェアを作りたい
- (オンライン前提で)「映える」動画を撮りたい
- 数理工学要素:「学習」「制御」を入れない

Concept:

現実世界の媒体を数理工学で操り,
アートを作り上げる



音



光



水!

音楽に合わせて噴水ショーをする!!

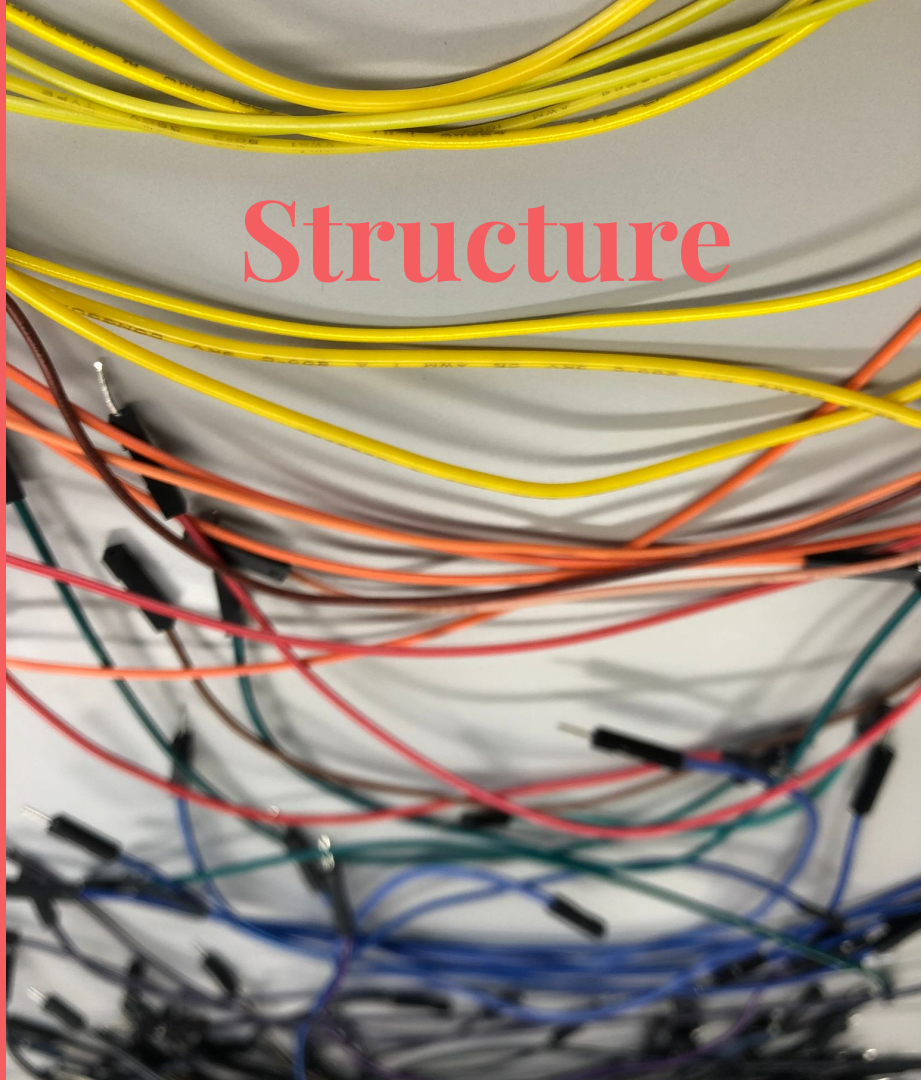
—

Method

材料はこちらに用意しています

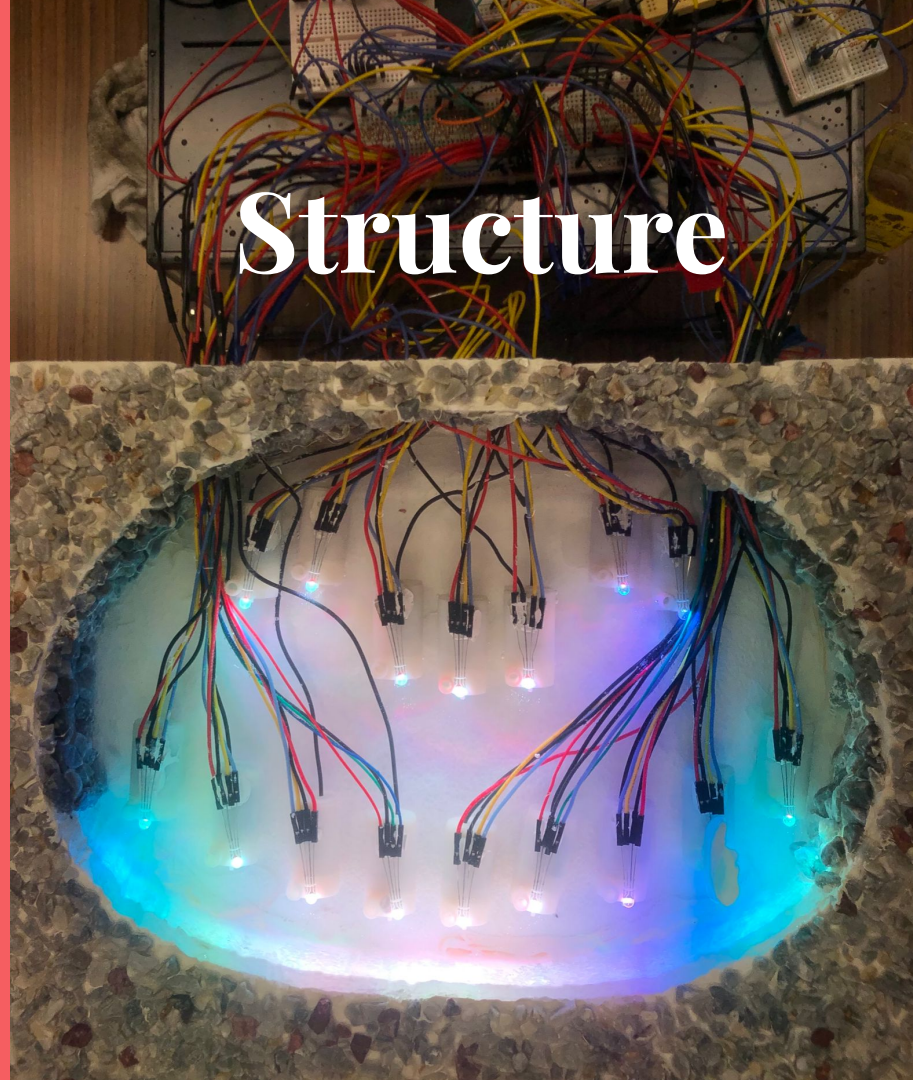
- 回路
 - ラズパイ2個
 - フルカラーLED16個
 - モーター16個
 - トランジスタ.....12個
 - ケーブル.....多数
 - etc...
- 噴水本体
 - 発泡スチロール
 - 小石
 - エポキシ樹脂

Structure



できあがったものがこちらになります

Structure



Sound Processing

```
def signal_handler(signal, frame):
    """
    This function is called when Ctr + C is pressed and is used to output the
    final buffer into a WAV file
    """
    # write final buffer to wav file
    if len(all_data) > 1:
        wav_filename = 'outstr' + ".wav", f' {len(all_data)}
        sys.exit(0)

signal.signal(signal.SIGINT, signal_handler)
```

```
def record_audio(block_size, devices, use_yeelight_bulbs=False, fs=8000):
```

```
    # initialize the yeelight devices:
    bulbs = []
    if use_yeelight_bulbs:
        for d in devices:
            bulbs.append(Bulb(d))
    try:
        bulbs[-1].turn_on()
    except:
        bulbs = []

    # initialize recording process
    mid_buf_size = int(fs * block_size)
    pa = pyaudio.PyAudio()
    stream = pa.open(format=FORMAT, channels=1, rate=fs,
                     input=True, frames_per_buffer=mid_buf_size)

    mid_buf = []
    count = 0
    global all_data
    global outstr
    all_data = []
    outstr = datetime.datetime.now().strftime("%Y_%m_%d_%I:%M%p")

    # load segment model
    [classifier, mu, std, class_names,
     mt_win, mt_step, st_win, st_step, _] = aT.load_model("model")

    [clf_energy, mu_energy, std_energy, class_names_energy,
     mt_win_en, mt_step_en, st_win_en, st_step_en, _] = \
        aT.load_model("energy")
```

音を調理していきます

- Spotifyから5000曲
 - 感情とその強さがラベル
- pyAudioAnalysisで130次元のベクトルに
- ベクトルからラベルをSVMで予測
 - RBFカーネルによるカーネル法
 - フレーズ毎に予測
- 訓練済みモデルをRaspberry Piに載せる
 - パソコン・スマホから入力可能

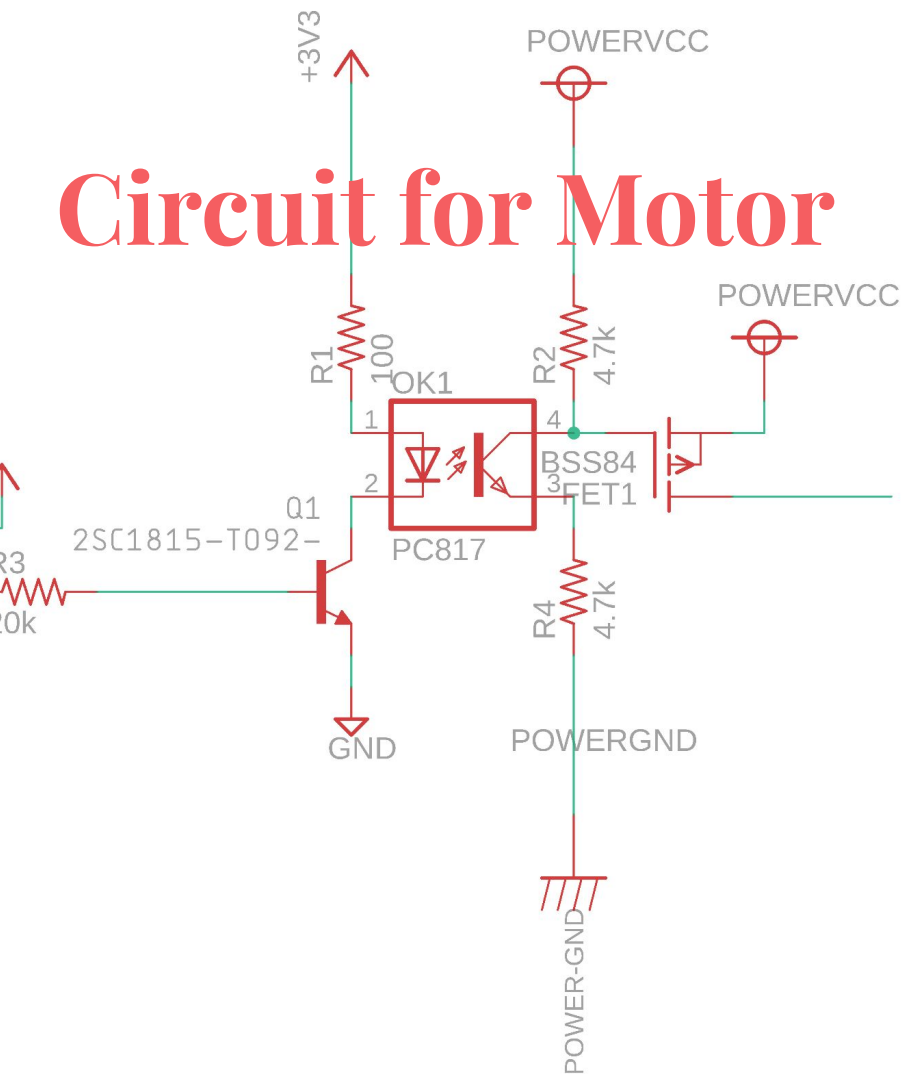
Circuit for LED



鮮やかに仕上げていきます

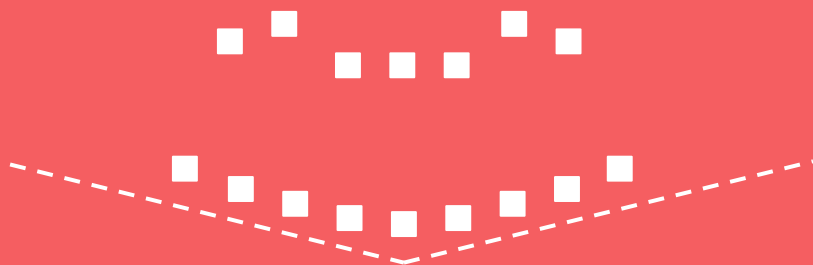
- 感情とその強さを出力
 - RGBで色指定
- 16個のLEDライト
 - 各噴出口に1つずつ配置
- Raspberry PiでPWM制御
 - 多数のLED→大出力が必要
 - トランジスタで出力増幅
 - PWMで明るさ調整

Circuit for Motor



水を入れていきます

- 16個の小型ポンプ
 - 配置は以下



- Raspberry Piでモーター制御
 - フォトカプラで絶縁
 - パワーMOSFETで駆動
 - 放熱板で排熱

Discussion

本日の工程を振り返りましょう

- 音声認識する噴水を作成
 - Raspberry Piで制御
 - モーターとライト16個ずつ
 - ひとまず形になった
- もっと映えるためには
 - 16個のライトを独立に制御
 - コントラスト
 - 噴水の高さを電圧で調整
 - スマホから操作
 - 学習手法の改善
 - 既存のものを繋いだので冗長



Discussion

Fin.

References

tyannak. color_your_music_mood. https://github.com/tyiannak/color_your_music_mood

Hackernoon. How to Use Machine Learning to Color Your Lighting Based on Music Mood.

<https://hackernoon.com/how-to-use-machine-learning-to-color-your-lighting-based-on-music-mood-bi163u8l>

Raspberry Pie!!!!!!!!!!!!

サクサクラズベリーパイを焼きました



Questions and Answers



各人の担当

- 筐体設計・作成
- 筐体設計・作成
- 環境構築・回路組み立て
- 環境構築・工作、料理
- 回路設計・制御、ソフトウェア