観覧車のための自動観光アナウンスキット

愛媛県　松山の魅力を発信し隊

白石　末廣　清家　北山　桐島

写真　くるりん｜松山市公式観光情報サイト｜四国松山 瀬戸内松山 (matsuyama-sightseeing.com)

目次

- 作品の概要
- デモと応用
- 気付き

写真　道後温泉　ホテル　メルパルク松山 日本最古の湯　道後温泉本館【楽天トラベル】(rakuten.co.jp)
shigureni free illust ｜ 素朴で可愛い、女の子のイラスト素材サイト

愛媛について


道後温泉


松山城


四国八十八ヵ所霊場
（51番札所 石手寺）


瀬戸内海

みかん


白くまピース

本イベントin愛媛のテーマ「愛媛を笑顔にするIoT」

パーパス

観覧車からの風景を遊覧中に、

各地点から見える観光地をアナウンス

することで、名所の魅力発信を支援するIoT

愛媛県松山市の観覧車 「くるりん」

アウトライン

- センサーで現在位置を把握

➡
- 各地点の観光名所を音声でアナウンス
- 観光名所の写真を表示

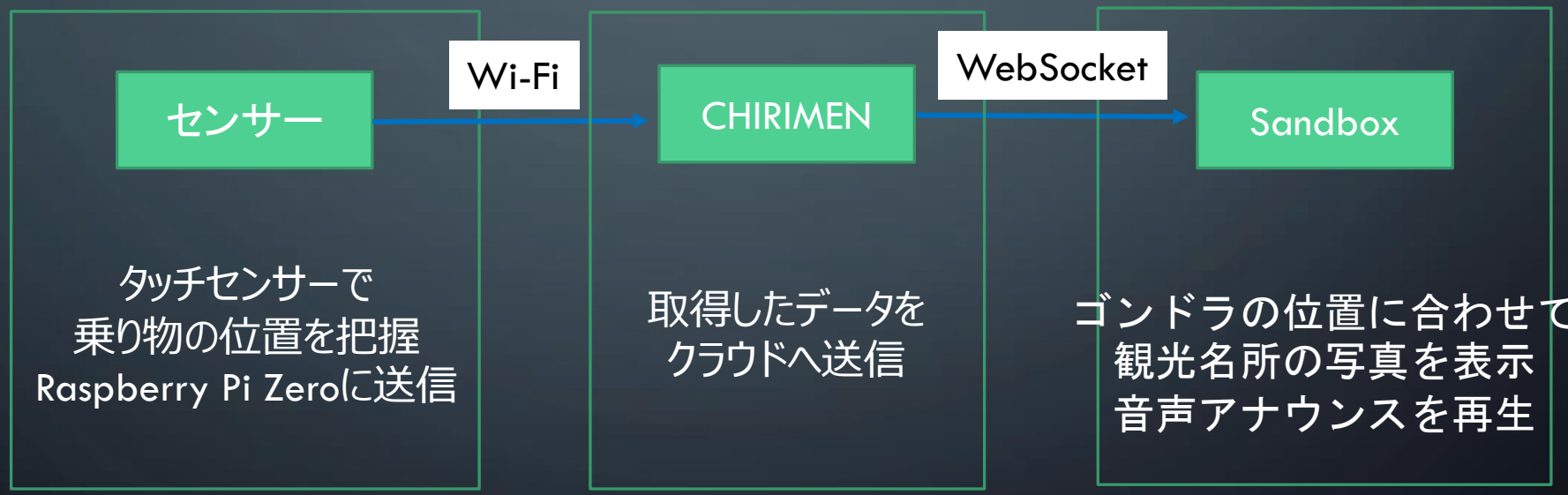
作品の構成

物品
・Raspberry Pi Zero
・タッチセンサー
・導電性銅箔シール

環境
・CHRIMEN
・Sandbox

通信規格
・Websocket

無線通信規格
・Wi-Fi



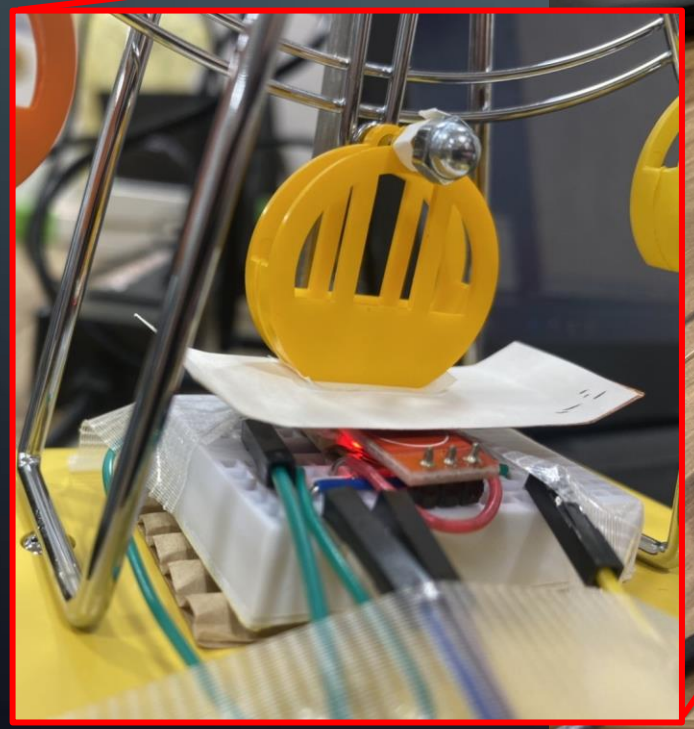
作品の構成

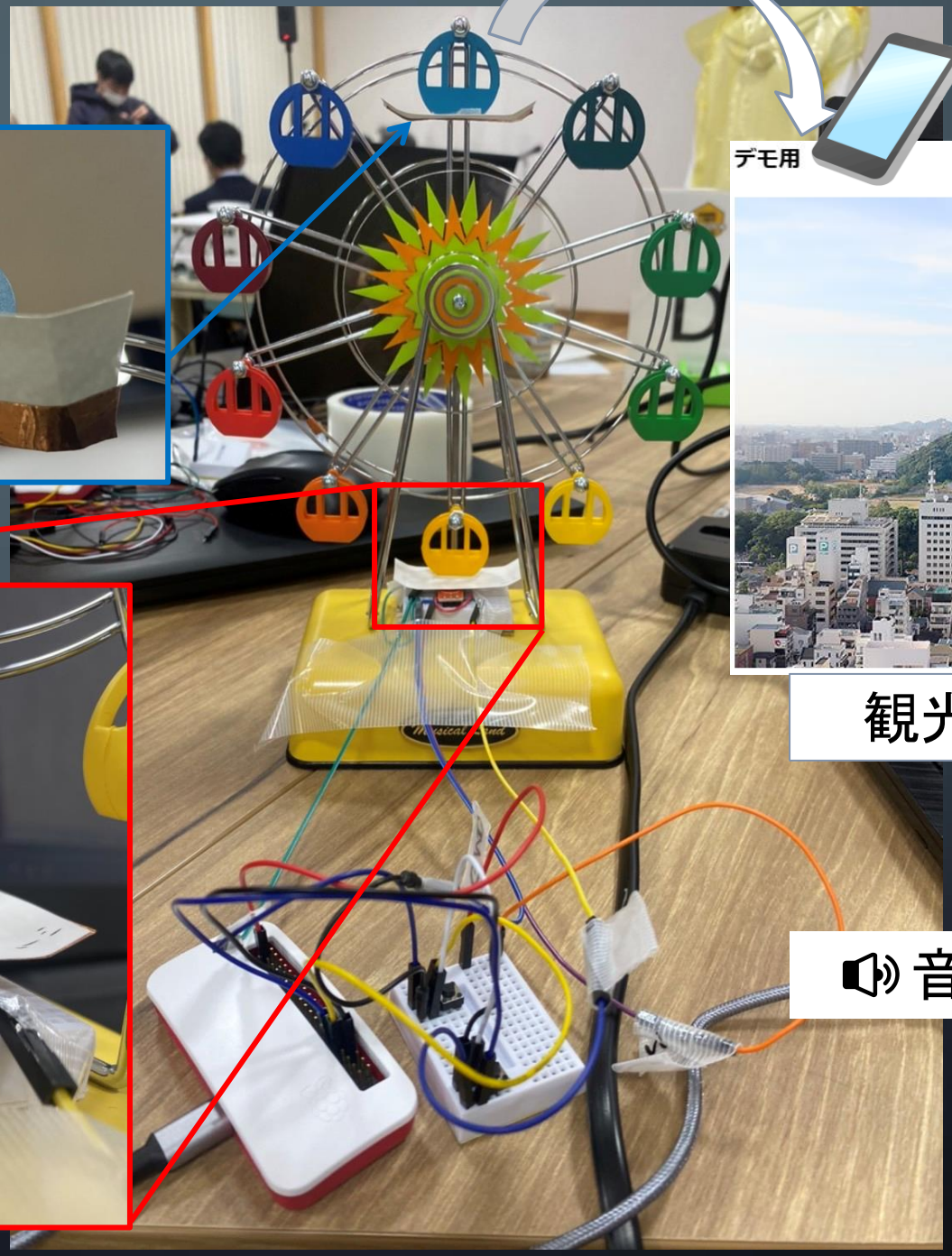
物品
- Raspberry Pi Zero
- タッチセンサー
- 導電性銅箔シール

環境
- CHRIMEN
- Sandbox

通信規格
- Websocket

無線通信
- Wi-Fi



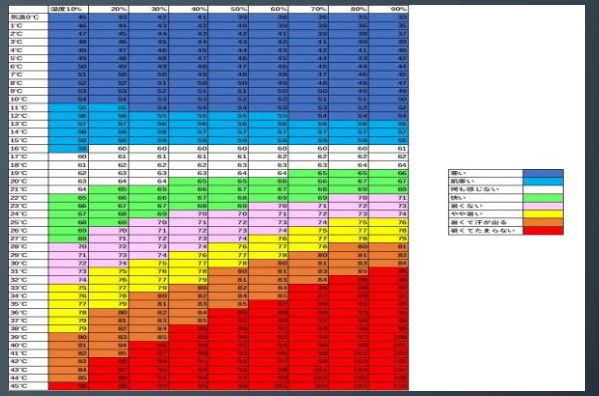


観光名所を表示

音声アナウンス

作品の応用ポイント

不快指数＝0.81×気温+0.01×湿度
x(0.99×温度－14.3)+46.3



不快指数の早見表

物品
- Raspberry Pi Zero
- タッチセンサー
- 導電性銅箔シール
- 温湿度気圧センサー
- LED発光ダイオード

環境
- CHRIMEN
- Sandbox
- Slack

通信規格
- Websocket

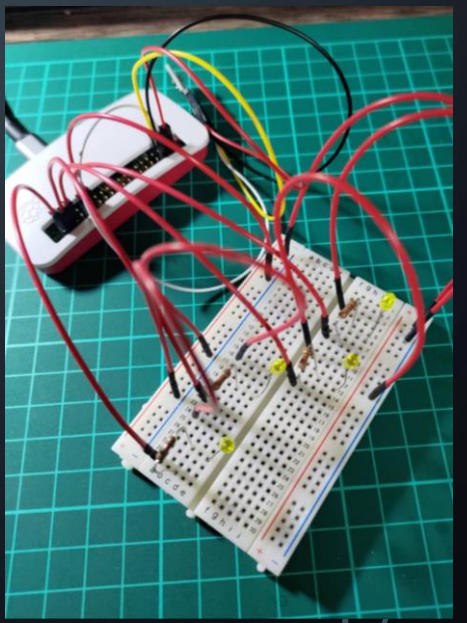
無線通信規格
- Wi-Fi

センサー → (Wi-Fi) → CHIRIMEN → (WebSocket) → Sandbox

タッチセンサーや
気圧センサーで
乗り物の位置を把握
Raspberry Pi Zeroに送信

取得したデータを
クラウドへ送信

位置に合わせて
観光名所の写真を表示
音声アナウンスを再生
名所の方角のLEDが点灯
室内環境の快適さを通知



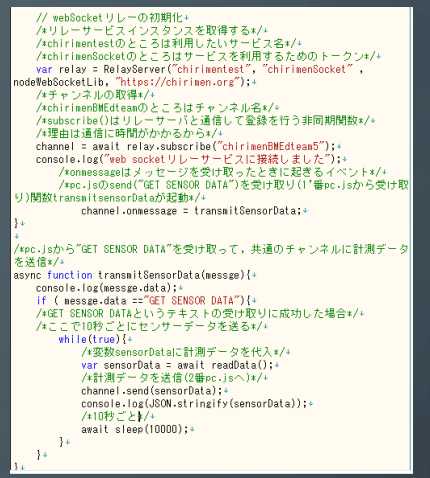
LEDで方角通知

気付き・所感

気圧センサーでゴンドラの高さを判別できることを確認したが、他機能と合わせるまでには至らなかった。

頂上付近でネットワークが不安定になる可能性。

実装を想定した場合、電源、配線、耐環境性、省電化など、ブラッシュアップを図りたい。



```
    // webSocketリレーの初期化↓
    /*リレーサービスインスタンスを取得する*/↓
    /*chirimentestのところは利用したいサービス名*/↓
    /*chirimenSocketのところはサービスを利用するためのトークン*/↓
    var relay = RelayServer("chirimentest", "chirimenSocket" ,
nodeWebSocketLib, "https://chirimen.org");↓
    /*チャンネルの取得*/↓
    /*chirimenBMEdteamのところはチャンネル名*/↓
    /*subscribe()はリレーサーバと通信して登録を行う非同期関数*/↓
    /*理由は通信に時間がかかるから*/↓
    channel = await relay.subscribe("chirimenBMEdteam5");↓
    console.log("web socketリレーサービスに接続しました");↓
        /*onmessageはメッセージを受け取ったときに起きるイベント*/↓
        /*pc.jsのsend("GET SENSOR DATA")を受け取り(1'番pc.jsから受け取
り)関数transmitsensorDataが起動*/↓
            channel.onmessage = transmitSensorData;↓
}↓
↓
/*pc.jsから"GET SENSOR DATA"を受け取って，共通のチャンネルに計測データ
を送信*/↓
async function transmitSensorData(messge){↓
    console.log(messge.data);↓
    if ( messge.data =="GET SENSOR DATA"){↓
    /*GET SENSOR DATAというテキストの受け取りに成功した場合*/↓
    /*ここで10秒ごとにセンサーデータを送る*/↓
        while(true){↓
            /*変数sensorDataに計測データを代入*/↓
            var sensorData = await readData();↓
            /*計測データを送信(2番pc.jsへ)*/↓
            channel.send(sensorData);↓
            console.log(JSON.stringify(sensorData));↓
            /*10秒ごと*/↓
            await sleep(10000);↓
        }↓
    }↓
}↓
```

データ受信側
（CHIRIMEN）

```
    messageDiv.innerText = JSON.stringify(mdata);↓
    console.log("mdata:",mdata);↓
    /*それぞれ対応しているhtmlのidに温度湿度気圧をテキスト形式で代入*/↓
    temTd.innerText = mdata.temperature;↓
    humTd.innerText = mdata.humidity;↓
    preTd.innerText = mdata.pressure;↓
    /*higherに10秒ごとの気圧をプッシュ*/↓
    hipress.push(mdata.pressure);↓
    /*データの個数によって表示方法と計算方法を変える*/↓
    ↓
    /*90回目から90回目までは計測中とする*/↓
    if(hipress.length<90){↓
        highTd.innerText = "データ集計中です";↓
        updownTd.innerText = "--";↓
    }else if(hipress.length >= 90 && hipress.length < 180){↓
    /*90回目から180回目までの時は，存在するデータで計算*/↓
        for(i = 0;i < hipress.length;i++){↓
            if(hipress[i] > Maxpress){↓
            /*最大値更新*/↓
                Maxpress = hipress[i];↓
            }↓
            if(hipress[i] < minpress){↓
            /*最小値更新*/↓
                minpress = hipress[i];↓
            }↓
        }↓
        /*最大最小がわかると高さの計算ができる*/↓
        high = [(Maxpress - mdata.pressure)/Maxpress - minpress]*100;↓
        /*高さを表示*/↓
        hithTd.innerText = high;↓
        /*上り，下り，最上部付近，最下部付近の表示*/↓
        if(high >= 90){↓
        /*高さ90以上のとき*/↓
            updownTd.innerText = "最上部付近";↓
        }else if(high <= 10){↓
```

計算及び表示
（Sandbox）

未完成のアイデア

- 気象データと連携してアナウンス内容をアレンジ
- 温湿度気圧や不快指数をDBで管理し、空調管理にフィードバック
- 衛星データを使った自己位置推定で他の乗り物への汎用性を拡大

ご清聴ありがとうございました！