

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

いよてつ 雷 Takashimaya



目次



作品の概要



デモと応用



気付き

愛媛について



道後温泉



松山城



四国八十八カ所霊場
(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」

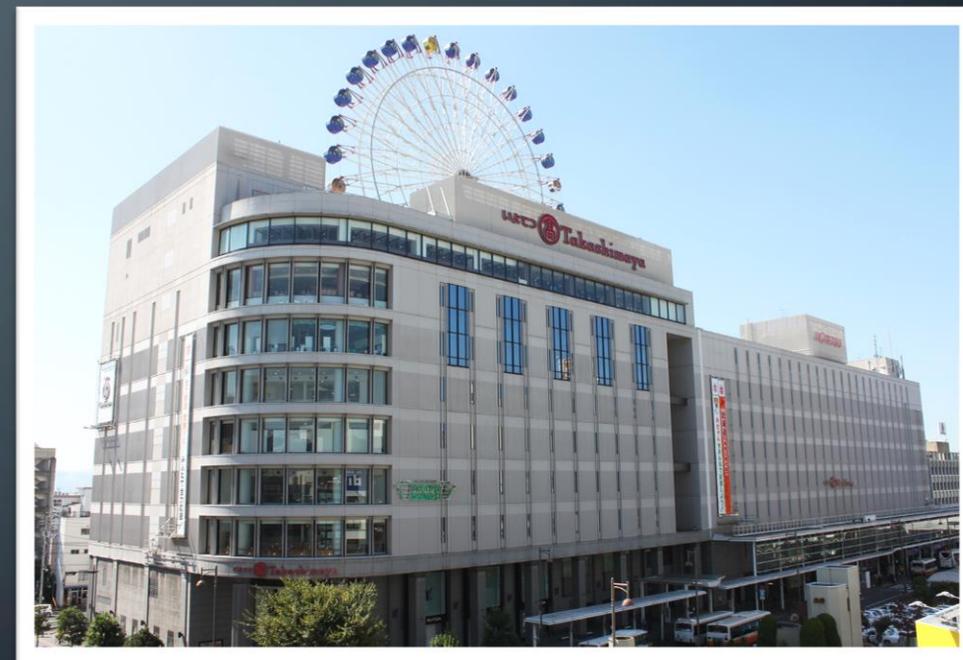
パーパス

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

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

愛媛県松山市の観覧車「くるりん」



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

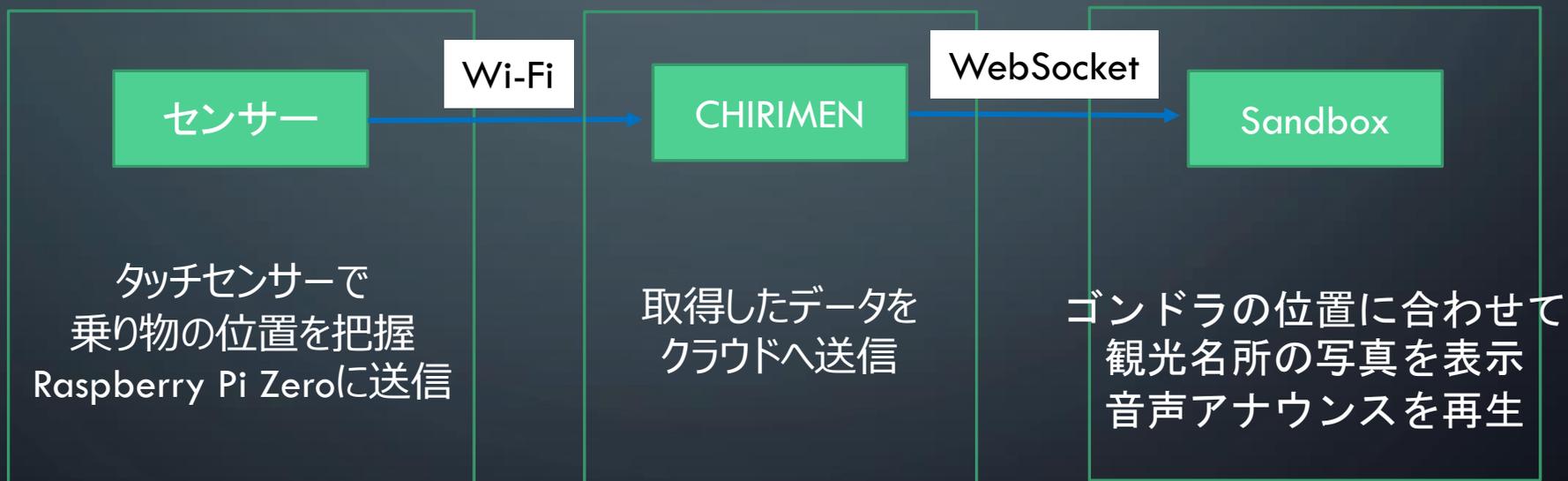
- ・CHRIMEN
- ・Sandbox

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

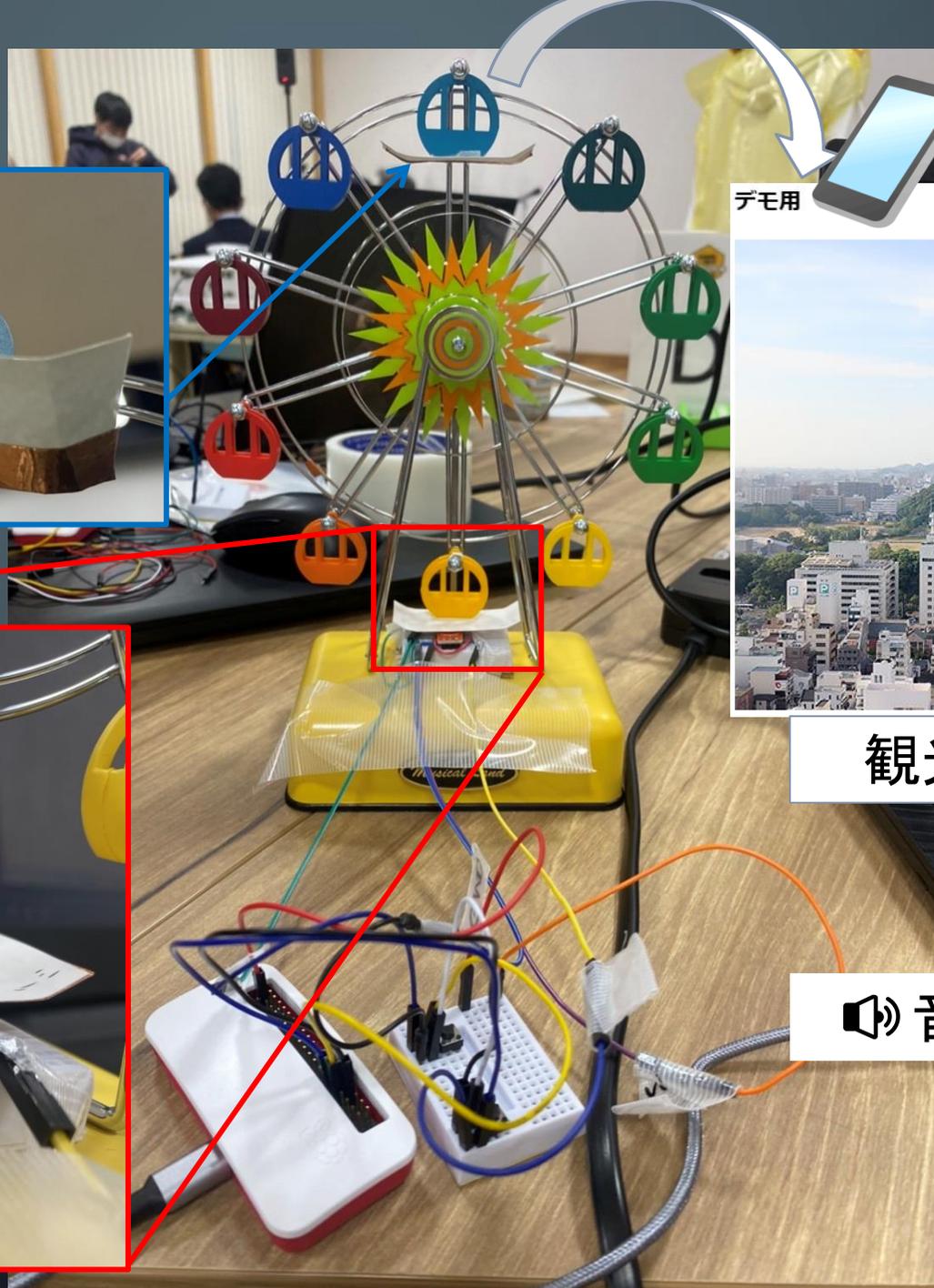
- ・ CHRIMEN
- ・ Sandbox

通信規格

- ・ Websocket

無線通信

- ・ Wi-Fi



デモ用



観光名所を表示



音声アナウンス

作品の応用ポイント

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

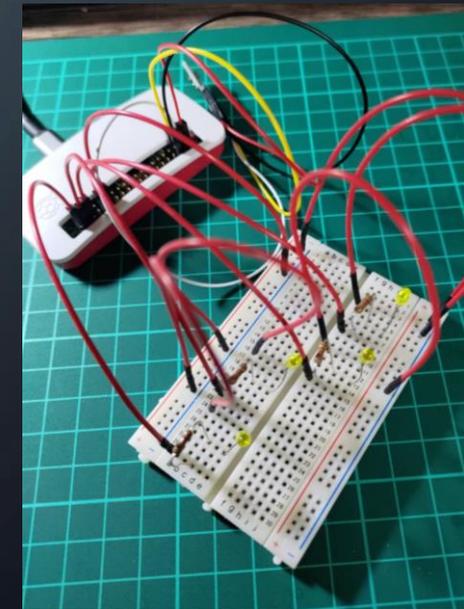
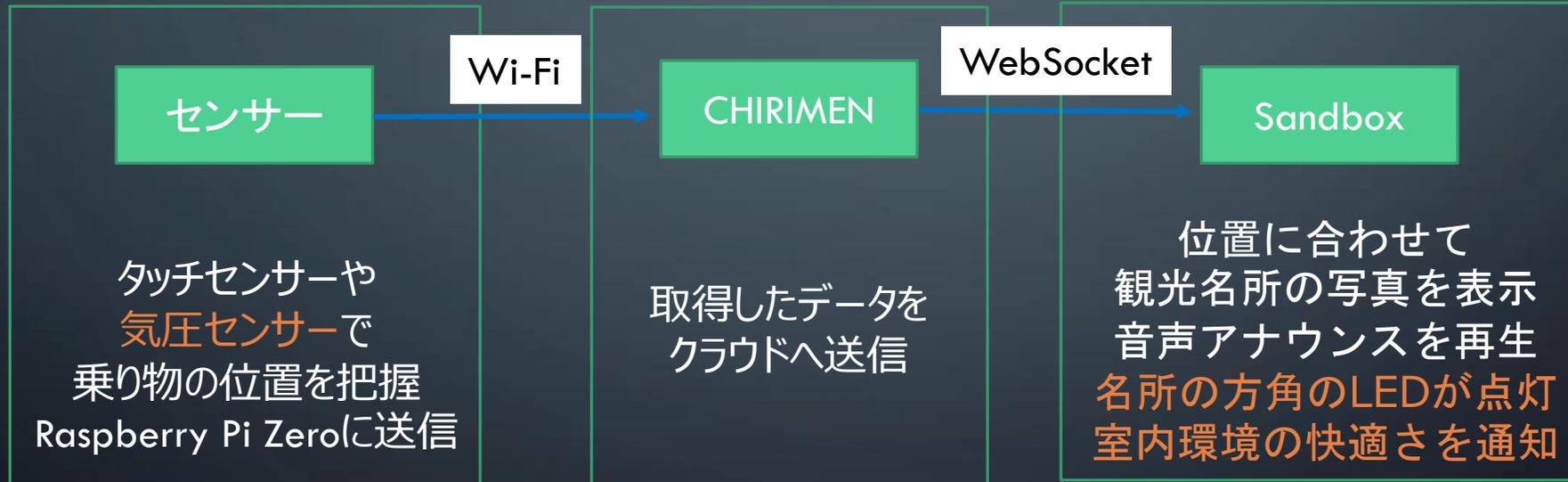
無線通信規格

- Wi-Fi

$$\text{不快指数} = 0.81 \times \text{気温} + 0.01 \times \text{湿度} \times (0.99 \times \text{温度} - 14.3) + 46.3$$

湿度	10%	20%	30%	40%	50%	60%	70%	80%	90%
10°C	40	41	42	43	44	45	46	47	48
11°C	41	42	43	44	45	46	47	48	49
12°C	42	43	44	45	46	47	48	49	50
13°C	43	44	45	46	47	48	49	50	51
14°C	44	45	46	47	48	49	50	51	52
15°C	45	46	47	48	49	50	51	52	53
16°C	46	47	48	49	50	51	52	53	54
17°C	47	48	49	50	51	52	53	54	55
18°C	48	49	50	51	52	53	54	55	56
19°C	49	50	51	52	53	54	55	56	57
20°C	50	51	52	53	54	55	56	57	58
21°C	51	52	53	54	55	56	57	58	59
22°C	52	53	54	55	56	57	58	59	60
23°C	53	54	55	56	57	58	59	60	61
24°C	54	55	56	57	58	59	60	61	62
25°C	55	56	57	58	59	60	61	62	63
26°C	56	57	58	59	60	61	62	63	64
27°C	57	58	59	60	61	62	63	64	65
28°C	58	59	60	61	62	63	64	65	66
29°C	59	60	61	62	63	64	65	66	67
30°C	60	61	62	63	64	65	66	67	68
31°C	61	62	63	64	65	66	67	68	69
32°C	62	63	64	65	66	67	68	69	70
33°C	63	64	65	66	67	68	69	70	71
34°C	64	65	66	67	68	69	70	71	72
35°C	65	66	67	68	69	70	71	72	73
36°C	66	67	68	69	70	71	72	73	74
37°C	67	68	69	70	71	72	73	74	75
38°C	68	69	70	71	72	73	74	75	76
39°C	69	70	71	72	73	74	75	76	77
40°C	70	71	72	73	74	75	76	77	78
41°C	71	72	73	74	75	76	77	78	79
42°C	72	73	74	75	76	77	78	79	80
43°C	73	74	75	76	77	78	79	80	81
44°C	74	75	76	77	78	79	80	81	82
45°C	75	76	77	78	79	80	81	82	83
46°C	76	77	78	79	80	81	82	83	84
47°C	77	78	79	80	81	82	83	84	85
48°C	78	79	80	81	82	83	84	85	86
49°C	79	80	81	82	83	84	85	86	87
50°C	80	81	82	83	84	85	86	87	88
51°C	81	82	83	84	85	86	87	88	89
52°C	82	83	84	85	86	87	88	89	90
53°C	83	84	85	86	87	88	89	90	91
54°C	84	85	86	87	88	89	90	91	92
55°C	85	86	87	88	89	90	91	92	93
56°C	86	87	88	89	90	91	92	93	94
57°C	87	88	89	90	91	92	93	94	95
58°C	88	89	90	91	92	93	94	95	96
59°C	89	90	91	92	93	94	95	96	97
60°C	90	91	92	93	94	95	96	97	98
61°C	91	92	93	94	95	96	97	98	99
62°C	92	93	94	95	96	97	98	99	100

不快指数の早見表



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。

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



```
// 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+
//chirimenest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("chirimenest", "chirimenSocket",
nodeWebSocketLib, "https://chirimen.org");
//チャンネルの取得+
//chirimenMedtea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開放+
//理由は通信に時間がかかるから+
channel = await relay.subscribe("chirimenMedtea5");
console.log("web socketリレーサービスに接続しました");
//spec.jsのsend("GET SENSOR DATA")を受け取り(1)spec.jsから受け取り)開放transmitSensorDataが起動+
channel.onmessage = transmitSensorData;
}+
}+
//spec.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(message){
  console.log(message.data);
  if (message.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(1)spec.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.innerHTML = 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+
tempDiv.innerHTML = mdata.temperature;
humidDiv.innerHTML = mdata.humidity;
presDiv.innerHTML = mdata.pressure;
//higher配列に10秒ごとの気圧をpush+
//データの個数によって表示方法と計算方法を定める+
+
//40回目から80回目までは計測中とする+
if(hiress.length>0){
  highTd.innerHTML = "データ集計中です";+
  updownTd.innerHTML = "--";+
} else if(hiress.length >= 90 && hiress.length < 100){+
  //80回目から100回目までの時は、存在するデータで計算+
  for(i = 0; i < hiress.length; i++){+
    if(hiress[i] > Maxpress){+
      //最大値更新+
      Maxpress = hiress[i];+
    }+
    if(hiress[i] < minpress){+
      //最小値更新+
      minpress = hiress[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+
highTd.innerHTML = 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.innerHTML = "最上部付近";+
} else if(high < 10){+
  //高さ10以下のとき+
  updownTd.innerHTML = "最下部付近";+
}
```

計算及び表示
(Sandbox)

ご清聴ありがとうございました！

