

お天気通知システム

~*Raspberry Pi*の活用~

github:ekazyam(rumcoke)

構築の動機

本システムは以下の動機/思想に基づき、作成されたものである。

■生活の悩み

1. 朝は晴れてたのに帰りに雨が降ってて、残念な気持ちになる。
2. しかしTVで毎朝天気予報見るのは面倒である。
3. さらにスマホで検索するのはもっと面倒である。

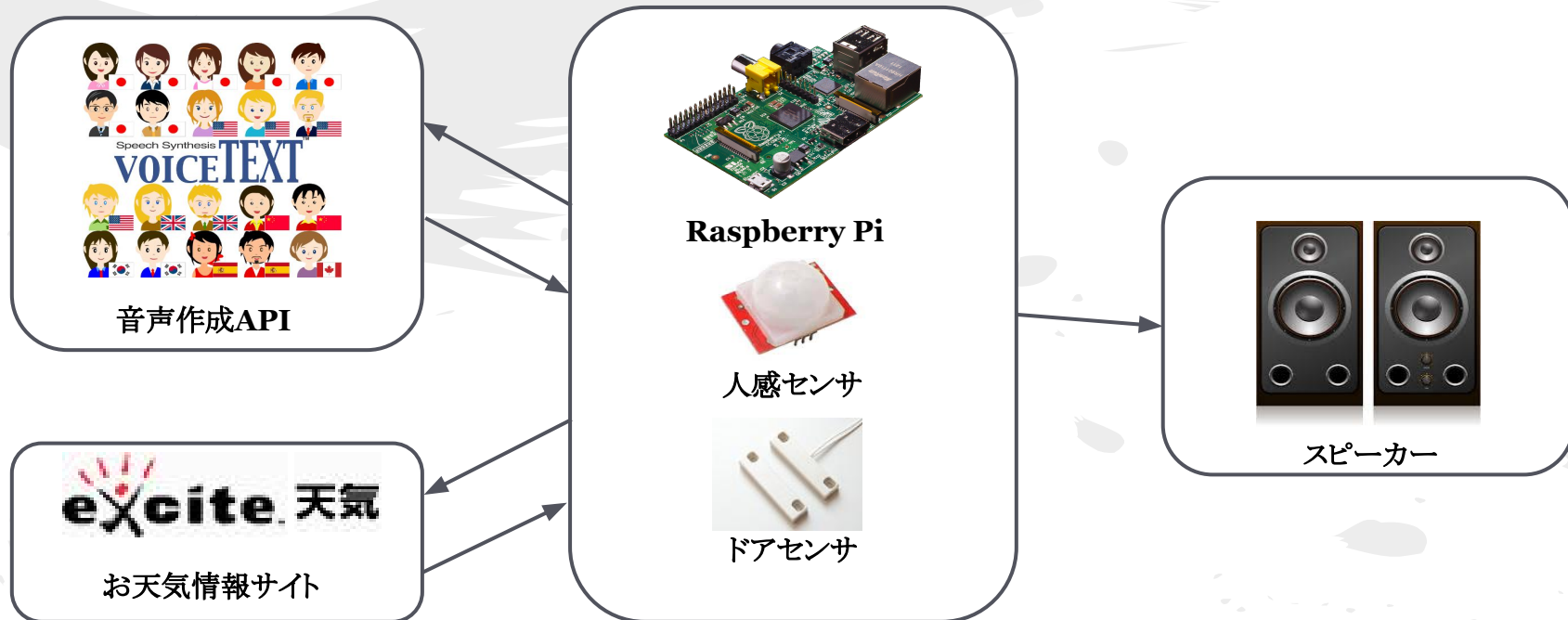
■こうなったらいい的な妄想の条件

1. 天気は自動で通知されるべきである。
2. 知るための動作をせずに自然と知ることができるべきである。

→生活スタイルを変化させずに、天気予報を知ることができるシステムを構築する。

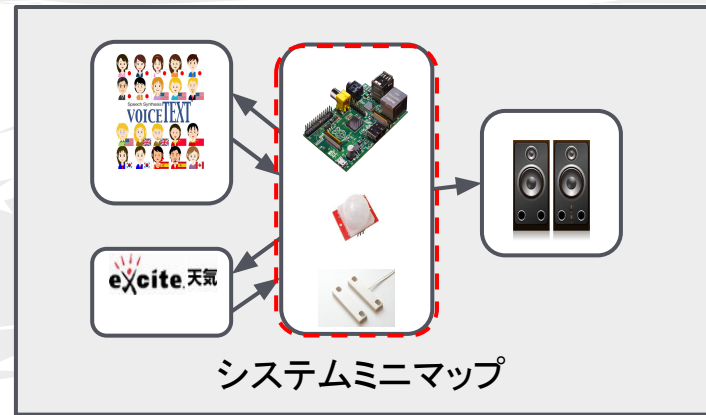
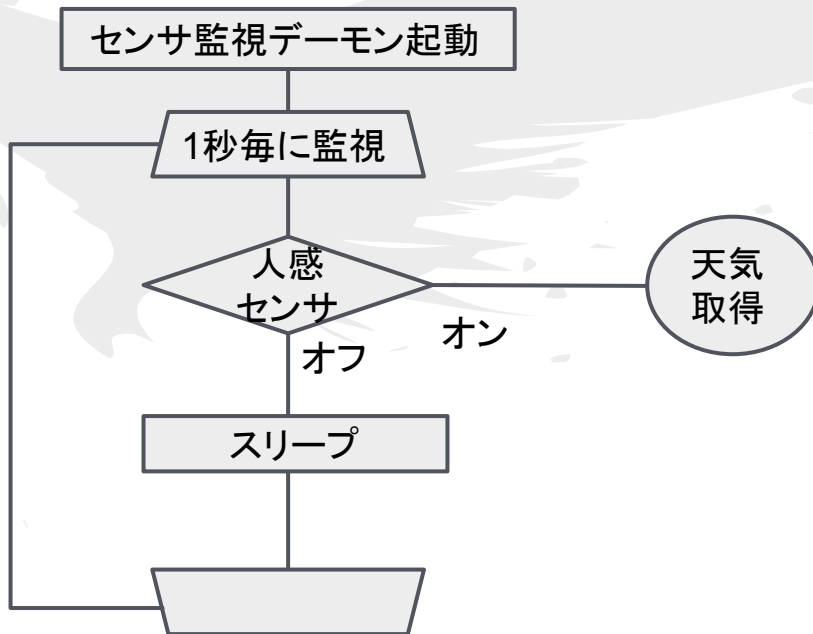
システム構成

下図の様に構成し、各サイト/APIを利用し取得したデータを音声通知するシステムを構築した。



処理フロー

～センサ感知から天気取得への連携～

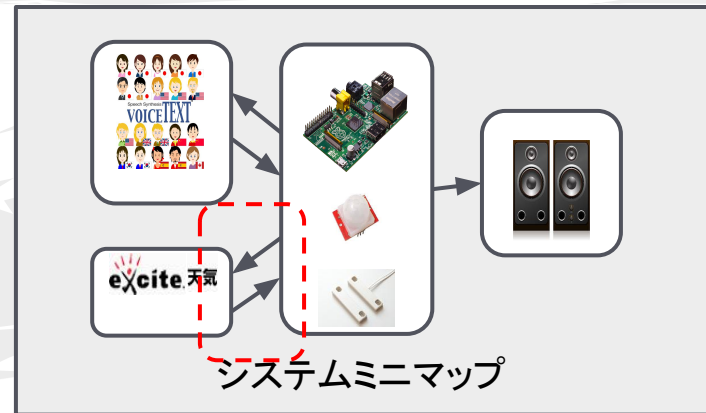
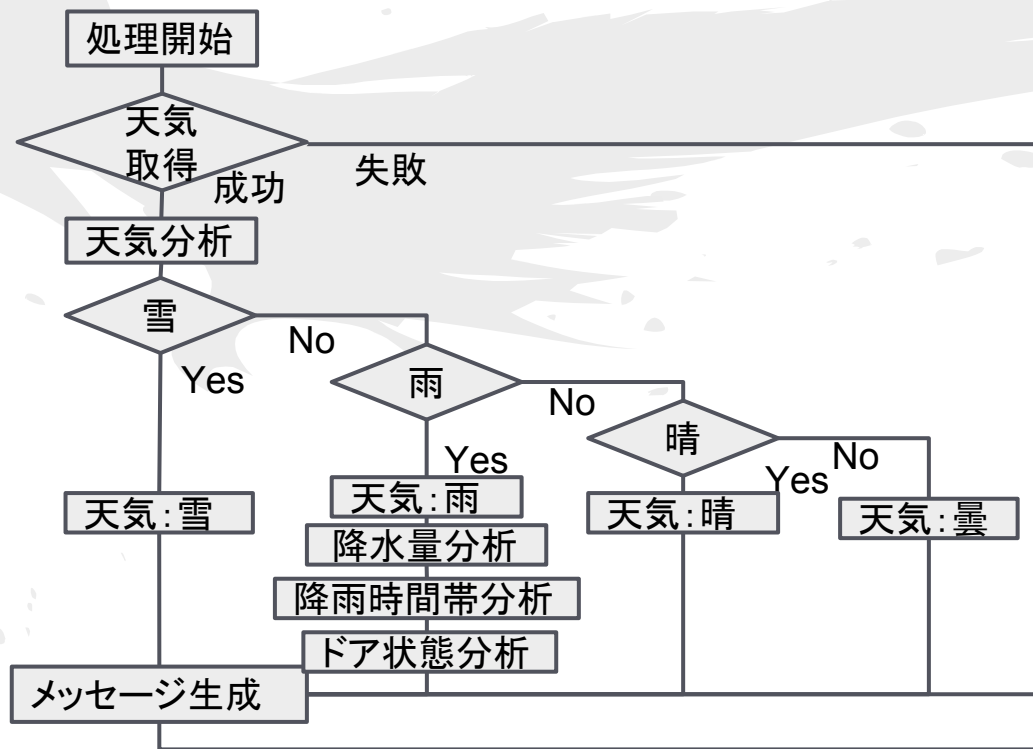


■概略

人感センサを検知するデーモンがシステムの起動と同時に常駐し、センサの情報を監視し続けることで、イベントの発生を他の処理へ受け渡す方法を確認した。

処理フロー

～天気取得から音声作成への連携～



■概略

取得したデータを分析し、音声通知ファイルの作成に必要なテキストデータの生成を行う。雨天の場合、降水量・時間帯を詳細に通知するように考慮を行った。

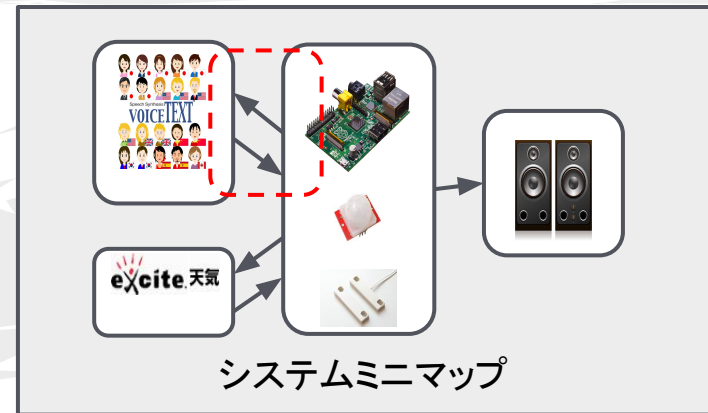
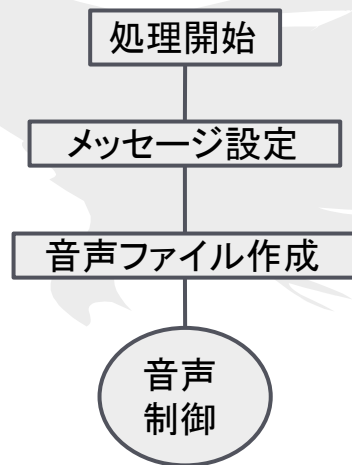
を
対
た。

音声
作成

また、ドアオープンセンサ追加したことにより、雨にする更なる考慮を施し

処理フロー

～音声作成から再生制御への連携～

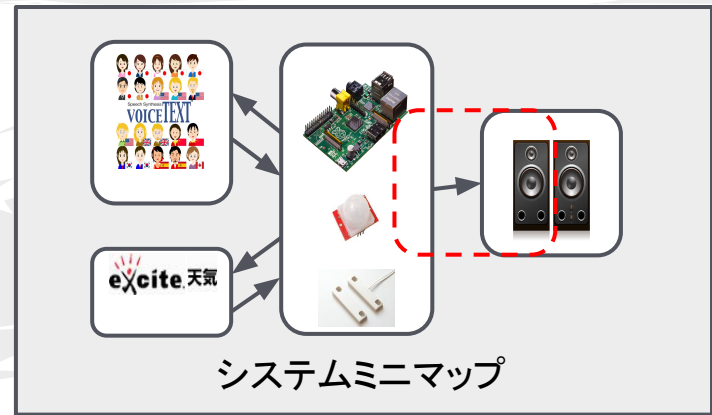
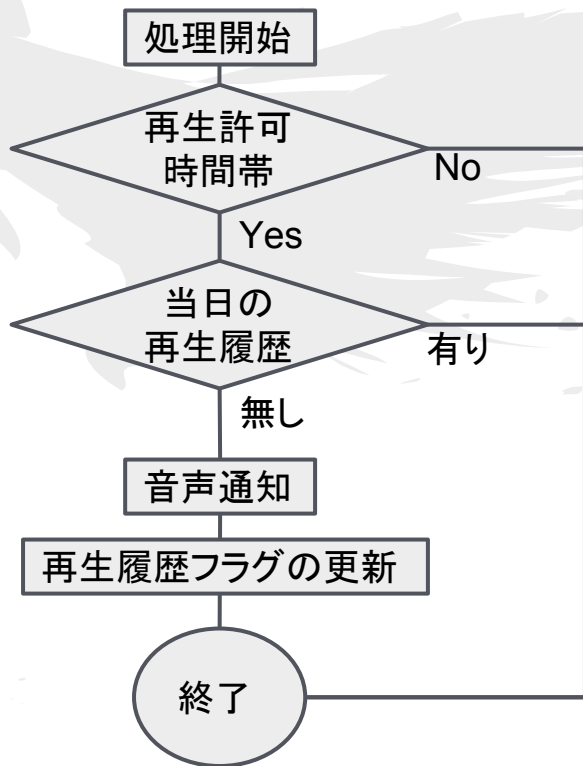


■概略

音声ファイル作成APIへテキストデータを送信することで、音声ファイルの生成を行う。
今回は、Voice TEXTサービスを利用し、音声ファイルの生成を行った。

処理フロー

~再生制御~



■概略

音声ファイルの再生を以下の条件に従い再生する。

1. イベント発生時に再生が許可された時間帯である。
2. イベント発生日に再生された履歴が存在しない。

仕様の策定にあたり、以下を考慮した。

1. 許可された時間帯の設定については、夜間等の外出を想定していない時間帯での通知を防止する為である。また、外出直後の最新データを反映することで、より精度の高い通知が期待できるものである。
2. センサによる感知のみをトリガーとして許容してしまうと、意図しない頻度で連続的に通知されることが想定される為、当日の通知回数を制限したものである。

システム化による恩恵

■生活面

1. 天気未確認時の傘忘れが激減した。
2. 天気を確認するという行為が日々の生活から無くなった為、煩わしさから解放された。

■技術面

1. デーモンの作成が初の試みであったので、他のモノ作りでも活かせる。
2. 音声の再生という行為が初の試みであったので、コマンドに詳しくなった。
3. 人感センサの実装/利用方法が理解でき、他のモノ作りで活かせる。

■その他

1. 天気データに詳しくなった。
例えば降雨量`omm`で雨天である等のイリーガルな状況の存在を知ることができた。

今後改良すべきポイント

■天気取得

1. **BB exite**の天気予報サーバがダウンしていた場合、別のサイトからデータを取得する。

■音声作成

1. **Voice TEXT**のサービス停止、サーバダウンに備え、別の音声作成API(**Aques Talk**等)を代替として用意する。

■音声通知

1. 聞き逃した場合に対応していない為、他の通知手段を考慮すべき。聞き逃した再の再通知はユーザが意図して行う動作であるため、自動通知ではなく、任意のタイミングで実行することを想定する。そのため、以下の様なUIを実装する。

★物理ボタンの設置、押下による再通知

★WebUIを用いたPC/スマートフォンからの再通知