**音声制御によるスマートホームシステム**

呉　華彬，厳　雪津（研究指導員　慎　祥揆　教授）

# Smart Home System with Voice Control

Huabin WU and Xuejin YAN　(Supervised by Prof. Sanggyu Shin)

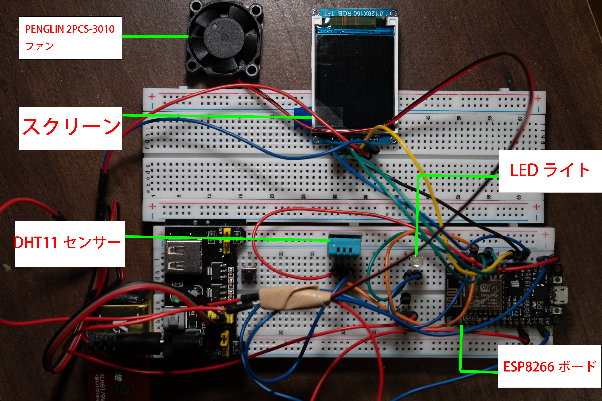


Fig.1 ホームシステム

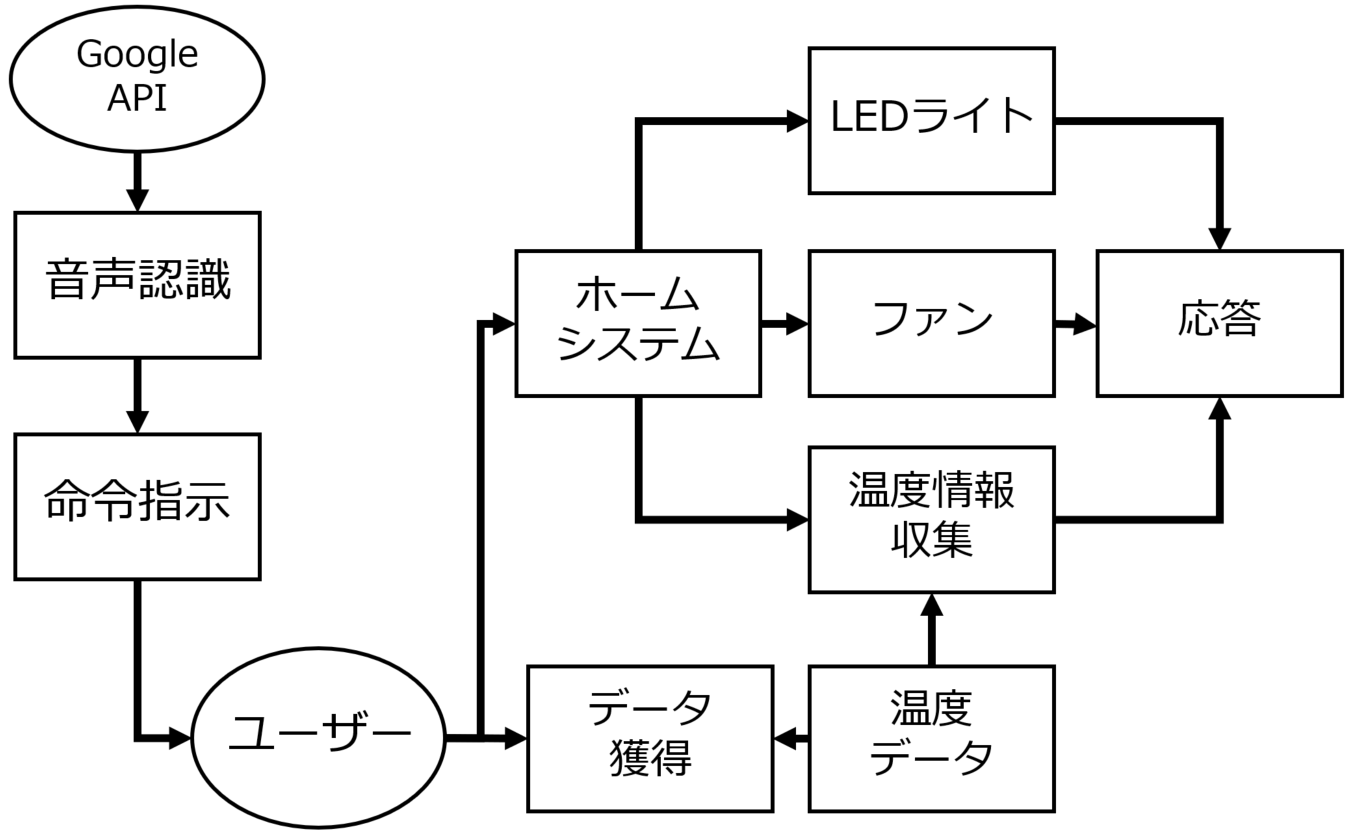


Fig.2 システムフローチャート図

**１．はじめに**

近年，音声認識技術は急速に発展しており，その応用はさまざまな分野に広がりつつある．特に，スマートホームや知能家具システムにおいては，ユーザーの声による操作が期待されている．これにより，従来のリモコンやスイッチに依存しない直感的なインターフェースが実現可能となる．本研究では，音声認識技術を用いて，知能家具システムの操作性向上を目指す．音声認識を活用することで，手がふさがっている場合や障害を持つユーザーでも簡単に家電や家具を操作できるようになる[1]．また，人工知能を組み合わせることで，ユーザーの生活習慣や好みに合わせた最適な提案が可能となる．本研究の目的は，音声認識を用いた操作システムを構築し，その有効性と操作性の向上を検証することである．

**２．システム概要**

本研究では，音声認識技術と知能家具システムを統合した次世代型家庭環境を提案する．ユーザーの音声を入力として取得し，音声認識エンジンにより命令を解釈する[2]．この認識結果に基づき，知能家具システムが対応する動作を行うことで，日常生活における利便性向上を図る．

**2.1　構築システムのハードウェア**

音声コマンドに基づいて操作可能な知能家具システムを構築するために，Wi-Fi機能を内蔵したマイクロコントローラ「ESP8266」を中心にハードウェアシステムを実装した（Fig. 1）．ESP8266は，強力な処理能力を持ち，Wi-Fiを使用してインターネット通信が可能であるため，本研究における音声コマンドによる制御システムに最適である．システムを実行する流れは以下のFig.2に示す．

温湿度センサーにはDHT11を採用した．このセンサーは，室内の温度および湿度をリアルタイムに計測し，そのデータをESP8266に送信する．温湿度データは音声コマンドを用いて取得することができ，例えば「現在の部屋の温度は？」といったコマンドに対応する．また，収集されたデータはファンやLEDの自動制御に利用される．

PENGLIN-２PCS-3010の５V小型ファンを採用した．音声コマンドを通じてオン/オフ制御や回転速度を調整することができる．ESP8266からPWM信号を供給することで，ファンの速度を動的に変更可能である．また，温湿度センサーから取得したデータに基づいて，自動的にファンの動作を制御するロジックを実装することで，エネルギー効率の向上を図った．

**2.2　ソフトウェア**

本研究では，音声認識のためのソフトウェア構築にPythonを用い，Googleが提供する音声認識APIを使用して音声のテキスト変換を実現する．また，Arduinoを使用して簡易サーバーを構築し，音声認識結果とハードウェア（ホームシステム）を連携させるシステムを開発した（Fig.3）．

**３．実験方法**

**3.1　音声認識テストの手順**

音声認識テストは，Googleの音声認識APIを利用した音声コマンドの認識精度を評価することを目的として実施した．まず，テストの環境を整えるため，音声入力を認識できる状態でシステムを起動し，静音環境と雑音環境という2つの条件を設定した．

次に，各音声コマンドを静音環境下で発話し，システムがコマンドを正しく認識したかどうかを確認した．同じ手順を，異なる話者が発話した場合にも繰り返して実施し，認識の汎用性を確認した．また，音楽や日常生活音といった雑音を背景にした環境を設定し，同様に音声コマンドを発話して，認識結果を検証した．

**3.2　各デバイスの動作確認手順**

音声認識によるデバイス制御が正しく動作しているかを確認するために以下の手順を実施する．

1. まず，LEDライトの動作確認を行う．音声コマンドを入力し，それぞれのコマンドに対してLEDが正しく点灯，消灯を行うかを観察する．
2. この結果はチェックリストを用いて「成功」もしくは「失敗」の形式で記録する．
3. 次に，ファンの動作確認を行う．音声コマンドを入力し，ファンが期待通りにオン・オフができるかを確認する．もしファンが正しく動作しない場合や予期しない動作をした場合には，その内容を詳細に記録する．

温湿度センサーについては，センサーが取得したデータが適切な形式で表示されるかを確認する．また，表示が正確であり，遅延が発生していないかを注意深く確認する．

**3.3　評価基準**

音声認識性能は以下の基準で評価する．まず，認識率（正確に解釈されたコマンドの割合）を算出する．全体の試行数に対する正確に解釈された音声コマンド数の割合を計算し，目標値を80%以上に設定する．

応答速度は，音声入力が終了してからデバイス動作が始まるまでの時間を測定し，その平均値を算出する．理想的な応答時間は1.5秒以内とし，これを超過する場合は改良が必要と判断する．さらに，雑音環境での認識率を測定し，静音環境における認識率との差から雑音耐性を評価する．この雑音耐性評価により，システムのロバスト性を確認する．

**４．実験結果**

**4.1　音声認識テストの手順**

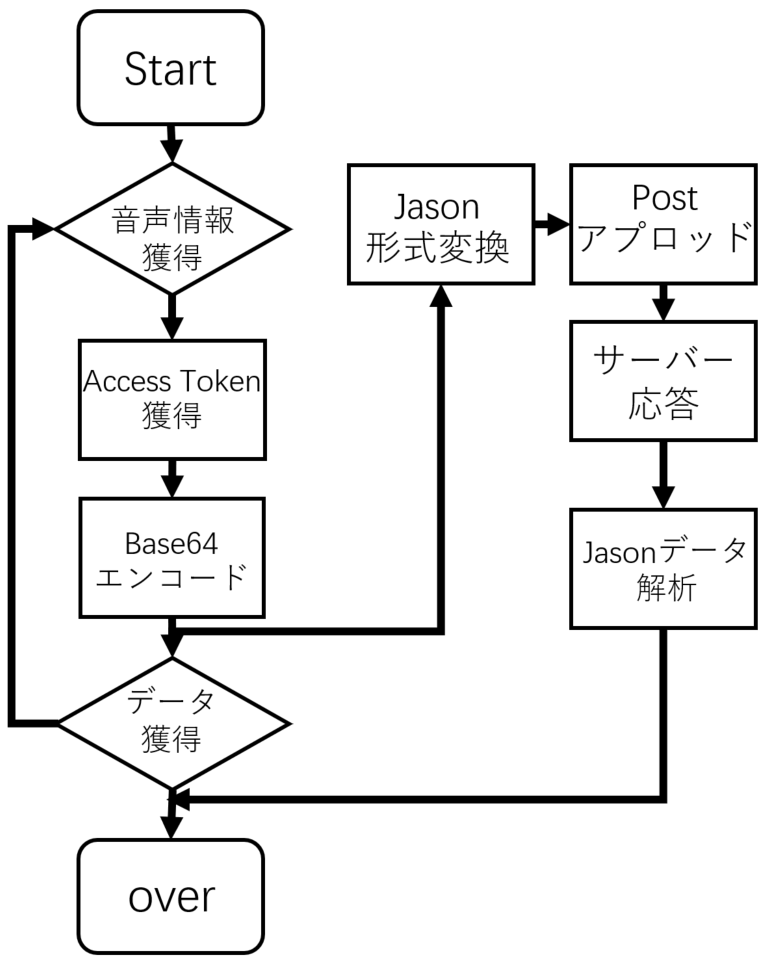


Fig.3 Google Speech-to-Text API フローチャート図

・静音環境での認識率

各音声コマンドを静音環境下で発話した際の認識率は表1に示したように平均95%であった．

認識率が最も高かったコマンドは「turn on the light」100%であり，最も低かったのは「tun on the fan」と「turn off the fan」85％であった．

・日常生活音環境での認識率

日常生活音を背景にした場合，認識率は82%であり，雑音の種類によって性能にばらつきが見られた．

Table 1　音声認識テストの結果

|  |  |  |
| --- | --- | --- |
| **条件** | **試行回数** | **認識率** |
| 静音環境 | 100回 | 95% |
| 日常生活音環境 | 100回 | 82% |

**4.2　各デバイスの動作確認結果**

表 2に示したようにデバイス全体は各20回で試行した。その中で一番認識率高かったのはLEDライト100%である。次は温湿度センサーの95%である。最後はファンのオン/オフ制御は85%である。各デバイスは理想的な応答速度以内であった。

Table 2　デバイスの動作確認結果

|  |  |  |  |
| --- | --- | --- | --- |
| **デバイス** | **試行回数** | **認識率** | **応答速度** |
| LEDライト | 20 | 100% | 2.83s |
| ファン | 20 | 85% | 2.4s |
| 温湿度センサー | 20 | 95% | 2.9s |

**5．おわりに**

本研究では，音声認識とハードウェア制御を統合したシステムを開発し，その性能を実験に通じて評価した．本システムは，PythonでGoogleの音声認識APIを利用して音声入力を処理し，ESP8266を介してハードウェアデバイスと連携する仕組みを採用した．今後の課題としては雑音環境や異なる話者での認識精度が低下する課題がある．これを改善するために，環境ノイズキャンセリング機能を追加するより高度なNLP（自然言語処理）技術やカスタムモデルのトレーニングを導入する必要がある．また，現在使用しているハードウェアに加えて，スマート家電やより多くのデバイスを制御する機能を追加することで，システムの実用性を向上させる．

参考文献

1. Arindom Chakraborty, Monirul Islam，Fahim Shahriyar，Sharnali Islam「Smart Home System: A Comprehensive Review」, March．2023
2. Bin Yuan「On the Security of Smart Home Systems: A Survey」, March．2023