

IoT デバイス構築による温熱環境調整システムの開発

Development of Thermal Environment Adjustment System using IoT Device

深井裕二*

Yuji Fukai

概要

北海道などの寒冷地住宅において暖房機器を使用する場合、天井付近に暖気が溜まり反対に床付近は低温となる。このような上下温度分布での快適性を改善するために、部屋内部の空気を循環させる方法がある。シーリングファンは天井付近の暖気を部屋全体に循環させることに適しており、これを自動運転し快適性の維持と省エネルギー化を目的とした温熱環境調整システムを開発した。本システムでは温度分布の計測に IoT センサを用い、さらにファンの回転速度制御のための IoT デバイスを独自に構築した。

1. はじめに

IoT (Internet of Things) は多様な物がインターネットに接続され情報交換することで相互制御が可能な仕組みである。この仕組みを実装した IoT デバイスにはセンサやアクチュエータなどの種類がある。これらは計測や制御を目的とする機器であり、企業・施設・家庭などあらゆる場面で活用されている。IoT デバイスはコンピュータやスマートフォンとの情報交換ができ、通信手段として近距離無線の Bluetooth や Wi-Fi あるいは LPWA (Low Power Wide Area) に分類される長距離無線などの通信方式を用いる。コンシューマ向け IoT デバイスは市場規模も大きく、例えば温度センサ (図 1) は小型で安価なものが市販されている。その活用法として、スマートフォンに管理アプリを導入し、計測データの

グラフ化や記録などを用いる住環境管理などがある。住環境の室温調整は生活に欠かせないものであり、特に寒冷地では暖房機器を用いて室温を一定に維持することが一般的である。また、暖房機器周辺と部屋の隅との温度差を緩和するために、部屋の空気を循環させるサーキュレータ機器を併用することがある。本研究では、温度センサ型の IoT デバイスを用いてサーキュレータを効率的に運転するための温熱環境調整システムを開発した。本システムの構成に関し、既存の IoT デバイスの採用に加え、柔軟な処理機能実装のためにマイクロコンピュータを用いた IoT デバイスの構築を行った。

2. システム化の背景および目的

一般的なファンヒーターなどの暖房機器では、設定温度を保つべく自動的に運転の稼働・停止や強・弱運転の切り替えを行っている。こうした自動化は人の操作を不要にして快適性を維持する機能である。しかし、暖房機器が放出した暖気は人が位置する空気層に行き渡るわけではなく、暖房機器前方で上昇し天井付近の層に留まるものと考えられる。こうして天井付近と床付近の温度差が増した状態で部屋全体が暖まっていく。暖房機器は裏面あるいは側面に搭載されている温度センサを用いて周囲温度が設定温度に到達すると暖房出力を減少・停止するようなエコ運転機能を有するものが多い。エコ運



図 1 Inkbird 製 温湿度スマートセンサ IBS-TH1

* 北海道科学大学工学部情報工学科

転中は断続的な送風であるため空気循環は少なく天井付近の暖気はより拡散されにくい状態と言える。住居内では人は椅子に座っていることが多く、部屋の高さの半分以下に体が位置していることになる。体の周囲を快適温度にする際、天井付近はそれより高温であり、この不要な高温化のために非効率なエネルギー消費が起こっている。

居住空間における不均一な温度分布は人の快適性を損なう場合がある。温熱快適性の6要素として、空気温度、湿度、気流、放射温度、代謝量、着衣量がある。例として、寒冷期の暖房使用時に窓付近で生ずる冷たい下降気流などは、寒気を感じさせ快適性を低下させるものである。温熱快適性の6要素を用いた快適温熱指標として、Fangerが提唱した予測平均温冷感申告（PMV, Predicted Mean Vote）⁽¹⁾がある。ISO-7730(2005)⁽²⁾では室内の上下温度分布における温熱的快適性に関し、PMVに基づく基準として、床上0.1m（足元）と1.1m（椅座位での頭部）の位置での推奨温度差を規定している。これは快適性によりカテゴリA~Cに分類されており、不満率と上下温度差の各最大値がA:3%, 2°C, B:5%, 3°C, C:10%:4°Cと示されている。

上下温度分布の改善策として、市販のサーキュレータは部屋の空気を循環させ温度分布の均一化に効果がある。なかでも天井に設置するタイプのシーリングファン（CF）は羽根径が大きく大風量であり、部屋全体の空気循環に適している。冬季などの暖房時では、図2のように送風方向を上向きにすることで上部の暖気を効果的に下部へ循環させることができる。このとき回転速度を上げ風量を増加させると短時間で上下温度差が減少する。また、上部温度が低下した状態で風量が大きいと、体にあたる気流が不快感を与えることがある。よって温度分布に応じた適切な風量調整が望ましい。そこでIoTデバイスを用いて部屋の上下温度分布を計測し、CFを自動

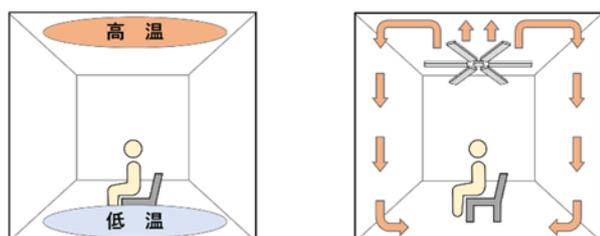


図2 シーリングファンによる暖房時の空気循環

運転するシステムを開発した。本システムは快適性維持と省エネルギー化を目的とし、上下温度差に応じて回転速度を適切に増減させるものである。

3. 温熱環境調整システムの開発

本システムの基本構成は、図3のように部屋の温度分布を計測するセンシング部とCFの回転速度を制御するコントロール部から成る。システムの主たる処理を担うホストPCはIoTデバイスとの情報交換やモニターへの情報表示を行う。また、家庭用CFとのインターフェースとしてマイクロコンピュータであるRaspberry Pi Zero⁽³⁾（RPZ）を用いた。

本システムでは常時運用を想定し、機器選定において省電力性を重視した。PCは超小型の省電力デスクトップパソコンであるECS製LIVA Q1L(Windows10 Pro)⁽⁴⁾（図4）を選択した。Windowsマシンの選択理由は、グラフィカルな表示機能や複雑・大規模な処理機能の実装において、開発生産性が高いことである。開発言語にはこのような開発に向いているC#を用いた。PC上のプログラムはIoTデバイスである温度スマートセンサからの温度情報をBluetooth通信で取得し、条件判断によってRPZに対しCF制御

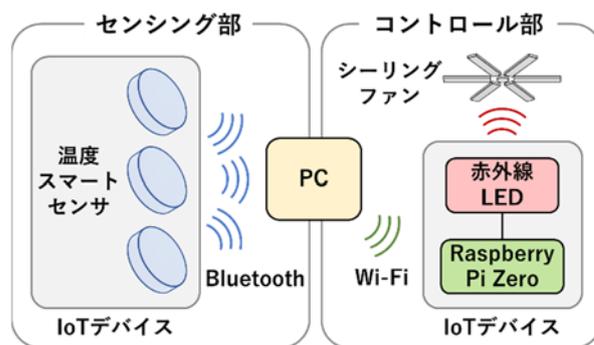


図3 温熱環境調整システムの構成



図4 超小型PC（左）、Raspberry Pi Zero（右）



図5 本システムのモニター表示 (7インチ)

の指示を送るものである。また、PCに接続した液晶モニターへのデータ表示機能を有している(図5)。

センシング部のIoTデバイスである温度センサにはInkbird製 IBS-TH1⁽⁵⁾(計測精度 $\pm 0.5^{\circ}\text{C}$)(図1)を屋内および屋外に設置した。これは乾電池で作動し、定期的な送信されるBLEアダプタイズパケットを利用しPCが温度を受信することができる。C#では標準機能を用いてBluetooth通信処理が記述でき、BLEデバイスのアドレスから個体識別しながら温度計測値を取得する。温度センサの屋内設置は、天井付近、座っている人の頭部(床上1.1m)と足元(床上0.1m)の高さとなる3か所とし、暖房機の正面や直上を避けた内壁面に両面テープで固定した。

コントロール部のIoTデバイスを構成するRPZ(図4)はマイクロコンピュータを搭載するワンボードコンピュータであり、microSDカードにインストールしたLinuxベースのRaspberry Pi OS⁽⁶⁾によって単体で稼働可能なマシンとなる。RPZはWi-Fi機能を有し、入出力ポート(GPIO)にセンサやアクチュエータなどのアナログデバイスを接続することで、カスタムメイドされたIoTデバイスとして機能させることができる。RPZでの開発言語には、OSに標準搭載されているPythonを用いた。プログラム内部で構成したWebサーバによりWi-Fi経由でPCからのCF制御命令を受け取る仕組みとなる。表1はCF制御命令を表しており、HTTPリクエストによって文字列コマンドとして送られる。ipとportはRPZのIPアドレスおよびWebサーバの待ち受けポートであり、URLパラメータのcommandに停止および速度指示のCF制御コマンドを与える。

表1 RPZへのHTTPリクエスト

```
GET http://<ip>:<port>?cmd=<command>
<command> ::= off|speed1|speed2|speed3|speed4
```

RPZは制御コマンドを受け取ると、GPIOに接続した赤外線送出モジュールを通じてCFを遠隔操作する。赤外線送出モジュールはリモコン信号を模倣的に送出するものであり、近赤外線(波長940nm)LEDをトランジスタによりドライブする装置として製作した。LEDは2芯コードで外部接続し自由な位置に設置できるようにしている。リモコン信号パターンは事前にCFのリモコン出力を記録したデータを用いた。これにはGPIOに赤外線受光素子を接続し、赤外線パルス信号のタイミングデータを記録するプログラムを作成した。これによってCFの速度設定および停止の信号パターンを得た。この手法を用いればリモコンを用いる多くの家電製品を自動操作できるものと思われる。

本システムにおけるCF制御の目的は、空気循環による頭部と足元間の温度差調整であり、天井と足元の温度差が大きいほど高速に回転させることで温度差の拡大を抑えるようにした。なお、CF制御は冬季などの寒冷時期における暖房運転時での稼働を想定しており、IoT温度センサによる室外温度の測定により寒冷時期であるかを判断している。また、発展として人が暑さを感じるような温度・照度の状況において、CFを逆回転させて扇風機のように人に風をあてる運転形態が考えられる。ただし、人の状態によっては涼しい快感となるか気流が不快感となるかは現状のシステムでは状態判断できない。今後、人の服装・代謝・活動の状態を把握できる何らかの手段を用い、多変量データから機械学習によって人が暑く感じる程度を判断する方法などが検討課題として考えられる。

4. システムの運用実験

本システムの動作および効果を確認するための運用実験を実施した。実験環境として、CFは4段階に回転速度を切り替え可能なものを使用しており、各風量は、大きい順に104, 73, 41, 27m³/minである。また、暖房機にはガスFF温風暖房機(暖房能力約9.3W, 8,000kcal/h)を使用している。本暖房

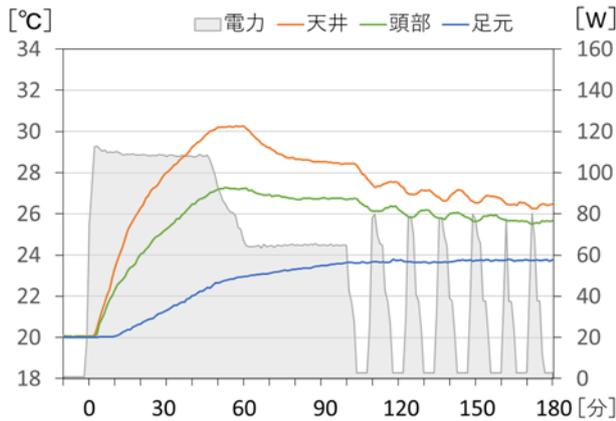


図6 実験計測データ (CF なし)

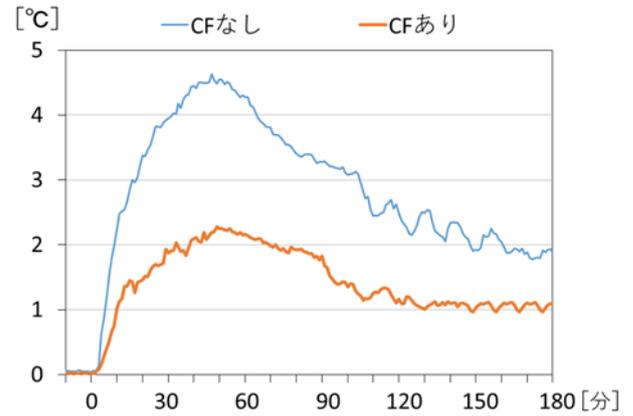


図8 頭部と足元間の温度差

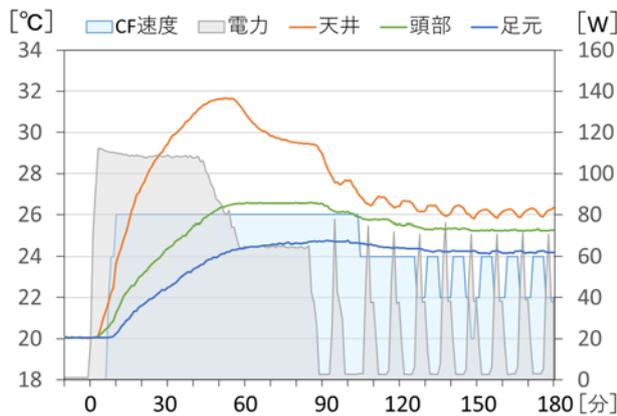


図7 実験計測データ (CF あり)

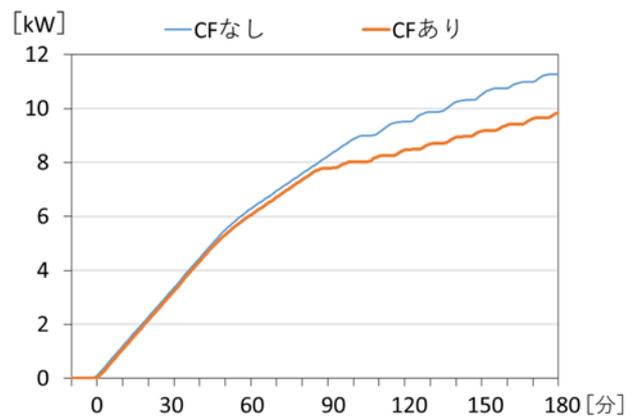


図9 暖房機器の累計消費電力

機はエコ運転機能により、室温と設定温度の差に応じて燃焼量が自動調整される。これにより設定温度になると運転を停止し、その後、設定温度を維持するために稼働・停止を繰り返すようになる。

実験は2回実施し、1回目はCFを全く稼働させず、2回目は本システムによってCFを自動運転させた。実験環境として、寒冷地（北海道札幌市）の一般住宅において、冬季の3月上旬の異なる日に実施した。実験条件は以下のとおりである。実施時刻は実内温度が低く、太陽光輻射熱の影響の少ない午前4:00頃とした。室内温度20°Cの状態暖房機器の設定温度25°Cにて運転を開始し、開始10分前から開始180分後までの屋内外の温度、暖房機器の消費電力およびCFの回転速度を計測データとして1分毎に記録した。本実験の計測中の屋外温度はそれぞれ平均-3.7°C、-4.5°Cであった。暖房機器の消費電力計測にはACコンセントに割り込ませるタイプのワット

チェッカーを用い、計測値を収集できるようにしている。この電力は主に暖房機器の送風ファンで消費されるものである。ファンの消費電力とガス消費量（暖房負荷）にはおおよその正相関があると捉え、電力をエネルギー消費総量のバロメーターとした。

図6および図7は実験1回目（CFなし）および2回目（CFあり）の計測結果である。グラフは、横軸に暖房運転開始を0とする時間軸をとり、天井付近、頭部、足元の各温度、暖房機器の消費電力およびCFの回転速度をOFF状態および4段階（1段階あたり右軸20W相当）で表している。どちらも室温上昇に至る40~50分まで暖房の強運転状態が続く、その後の弱運転はCFなしで100分頃まで続くが、CFありはそれより20分程度短い。これはCF運転により部屋下部の温度が早く上昇したためと思われる。また、CFありでは天井と足元の温度差に応じてCF回転速度が制御されている様子が見られる。

図 8 は、上下温度分布での温熱的快適性の評価として頭部と足元間の温度差を示したものである。CF なしではグラフの半ば程までは 3°C を超えており快適性が低い。一方、CF ありでは温度差は約半減し、ISO-7730 のカテゴリ A 相当である 2°C 以下の快適性の高い状態がほぼ維持されている。

図 9 は、暖房機器の運転開始後 1 分ごとの累計消費電力である。グラフの半ば程から差が大きくなり、180 分後において CF ありでは約 1.5kW の減少が見られる。これは CF による部屋下部の温度上昇による暖房機器の燃焼量低下あるいは燃焼停止による省電力効果であると考えられる。

5. システムの評価

CF による上下温度差の減少効果としては、図 8 より最大温度差となる 50 分近辺で約 50% に、温度差が安定する 180 分近辺で約 60% に減少している。本システムの制御により、暖房開始後の温度差の急上昇に対し、CF を最大回転させることで迅速に温度差を低下させている。また、温度差低下に応じ CF の回転速度を下げている。こうした動的な制御では早く部屋下部の温度を高め、上下温度差の小さい快適なゾーン（カテゴリ A）を維持できる。

このような制御を人の操作で行う場合、体感温度を頼りに主観的に判断すると、制御の精度が大雑把かつ不安定となり、室温計を見ながらの判断や暖房機の運転・停止に合わせた操作が煩雑な作業となる。また CF を手動操作した場合、1 回だけ操作し回転速度を高い状態のままにしておく、温度差低下後は涼しい気流を感じ、寒気を伴う不快感を受けることがあった。快適性を維持するためには、手動操作でも体感温度に応じて適切に速度調整する必要があるが煩雑さは解消されない。加えて手動調整は寒気をはっきり自覚した後になるため、温度分布に応じた即時的な自動調整よりも調整のタイミングが遅れる可能性がある。また手動操作に比べ安定した制御精度が得られるという点でも自動化は有効である。

暖房機器の消費電力について、図 9 より CF 運用により消費電力が抑えられ、これに伴い暖房機器の燃料消費量も抑えられ省エネルギー化につながっている。なお CF を全く使わない状況に比べ、CF 稼働そのものによる電力消費が生ずることになる。今回使用した CF の消費電力は、回転速度に応じ 14W, 5.4W, 2.5W, 1.7W である。手動操作によって CF を高い回転速度のままにするような運用形態に比べ、

本システムによる温度差に応じた回転速度制御は不要な消費電力を抑える働きがある。

次に、本システムの構成法について評価する。一般に IoT デバイスには相互の通信・制御による連携でシステム構築を容易にするものである。しかし実際に IoT デバイスを連携させるにはスマートフォン等を用いる場合が多く、IFTTT⁽⁷⁾などの Web サービスやアプリ等を用いた連携形態が利便性の高いものとして知られている。しかし、これらは主に非プログラミングによる処理構築であるため活用自由度が低い。本システムではシステム構成や機能における拡張性を重視し、PC と RPZ への機能分担およびプログラミング言語による開発を行った。管理・制御・記録・表示・連携等の機能は、ホスト PC に実装するのが容易で開発生産性が高い。また、複数のタスクや複雑な機能を動作させる場合、処理性能の面で PC は適している。

一般に IoT デバイスは情報収集や制御の処理機能を有するが、より高度な機能や独自の機能を実装したい場合、プログラミング可能なマイクロコンピュータが適している。RPZ は Python などで通信処理やマルチタスク処理等が容易に記述でき、IoT デバイス内に複雑な処理を完結させることができる。こうした RPZ 側に処理を独立実装する構成では、機能性とモジュール性を高め、システムの発展・拡張がしやすく、かつ障害対策やメンテナンスに有利である。もし将来 CF の機器変更が生じて、PC 側はリモコンなどの信号媒体には依存しないため、RPZ 側の修正や再構築のみで済む可能性が高い。

以上のように、本システムでは将来の拡張性を考慮し、ホスト PC を中心として IoT デバイスをシステムに追加していく構成法をとった。既存のアプリや Web サービスを利用せず、プログラミング技術を用いたことで、処理や構成の自由度を高めている。本手法は多機能化・複雑化に容易に対応できる。複雑な処理例として、マルチスレッディング、タイマー処理、複合的な条件判断処理などが挙げられ、PC および RPZ への実装は難しくない。また、温度などの環境系の計測値では、ゆらぎに対する平滑化やヒステリシスな制御などのデータ処理や状態遷移処理を行う場合がある。このような機能を IoT デバイスに実装する場合、プログラミング技術が不可欠である。RPZ ではセンサ、モータなどの部品を接続し、プログラミングにより様々な独自機能の実装が可能である。

本システムでは、運用時も RPZ 上のプログラムのバージョンアップ等を実施してきた。これにはリモートデスクトップツールである UltraVNC⁽⁸⁾を用いて PC から Wi-Fi 経由で RPZ を遠隔操作した。RPZ の OS には VNC サーバ機能が標準装備されており、PC に VNC クライアントをインストールすれば、遠隔操作によりプログラムの開発・デバッグ、機器の再起動などができる。これにより RPZ の IoT デバイスに対する設置場所からの取り外し、分解、ケーブル接続作業などが不要となる。また従来のマイクロコンピュータ開発で用いられた ROM ライタ装置やファームウェア転送ツールなどはプログラム変更時に不要であり、開発が容易かつメンテナンス性が高い。

6. まとめ

本研究では、温度分布に応じて GF を効率的に運転する温熱環境調整システムを開発した。本システムでは既存および独自構築した IoT デバイスを用いて、プログラミングによる機能実装を行った。本システムの自動調整機能は良好に動作し、運用実験において温熱環境維持の自動化システムとして有効であることがわかった。今後、IoT デバイスを用いて本システムを機能拡張し、より多彩な環境管理システムとして発展させたい。

参考文献

- (1) Fanger, P. O.: Thermal comfort: analysis and applications in environmental engineering, Danish Technical Press Copenhagen, 1970.
- (2) ISO 7730: Ergonomics of the thermal environment – Analytical determination and interpretation of thermal comfort using calculation of the PMV and PPD indices and local thermal comfort criteria, 2005.
- (3) Raspberry Pi Foundation: Raspberry Pi Zero, <https://www.raspberrypi.com/products/raspberry-pi-zero/>, 参照日 : 2021-11-25.
- (4) ECS: LIVA Q1L, https://www.ecs.com.tw/jp/Product/LIVA/LIVA_Q1L/overview, 参照日 : 2021-11-25.
- (5) Inkbird: Bluetooth Thermometer IBS-TH1, <https://inkbird.com/products/ibs-th1>, 参照日 : 2021-11-25.
- (6) Raspberry Pi Foundation: Operating system image, <https://www.raspberrypi.com/software/operating-systems/>, 参照日 : 2021-11-25.
- (7) IFTTT Inc.: IFTTT, <https://ifttt.com/>, 参照日 : 2021-11-25.
- (8) UltraVNC Team: UltraVNC VNC OFFICIAL SITE Remote Desktop Free Opensource, <https://uvnc.com/>, 参照日 : 2021-11-25.