

Raspberry Pi 周辺の熱移動の見える化実験

秋元優貴*1・岡村友樹*1・加藤樹*1・楠目啄也*1・坂本知也*1・上園波輝*2

上田真也*3・横山有太*4・高田拓*5

Yuki AKIMOTO, Yuki OKAMURA, Tatsuki KATO, Takuya KUSUME, Tomoya SAKAMOTO,
Namiki UEZONO, Shinya UETA, Yuta YOKOYAMA, and Taku TAKADA

Summary

In order to understand the propagation characteristics of the heat generated from Raspberry Pi 3 onboard computer, 20 temperature sensors were placed in the equipment box, and the change of temperature over time was measured. Three parameters—the time to reach the constant equilibrium temperature, the temperature rise, and the equilibrium temperature—were derived from the acquired data by approximating the temporal change of the temperature at each measurement point as an exponential function. We also confirmed that the equilibrium temperature and rate of heat propagation in the equipment box can be varied experimentally with constant CPU loading rates of 0%, 200%, 300%, and 400%. Actual temperature control can be implemented by monitoring the values of the CPU temperature, CPU load, and clock frequency. As a result, we can understand the characteristics of heat propagation from the onboard computer to the inside of the equipment box, obtaining knowledge from the information and of the means necessary for designing the observation equipment.

1. 序 論

Raspberry Pi に代表されるオンボードコンピュータを用いた観測装置は、小型軽量であり、様々な場所に持ち運んで使用できる。一方で、計測を行うセンサ類には、適正な動作が想定されている温度や気圧の範囲がある。動作範囲を超えるとデータの異常やセンサの故障につながる可能性がある。短時間の計測時には問題とならないが、長時間・長期間の測定時には、機器に大きな問題を与えることも想定される¹⁾。笹岡ら 2018 では、高層などの低温下において、オンボードコンピュータによる発熱を装置ボックスの保温に利用できるかどうかを検討した²⁾。具体的には、常温と低温環境下において、Raspberry Pi を起動させ、装置ボックス内の温度変化を調べる実験を行った。さらに、CPU 負荷率を意図的に大きくして、装置ボックス内の温度変化を計測することで、 -20°C

の低温下において装置ボックス内の温度を 10°C 程度上昇させる効果があることを示した。一般に、装置ボックス内の熱源は局在しており、温度の空間勾配は小さくない。そのため、装置ボックス内の熱伝搬を把握することで、装置ボックス内の最適な配置や大きさなどが推定できる。

Raspberry Pi には、複数のモデルが存在するが、例えば、Raspberry Pi3 には、CPU に 1.2 GHz のクアッド（4つの）コアの ARM 系プロセッサが搭載されている。デフォルトではヒートシンク等による熱対策はなされておらず、動作温度範囲は $0\sim 70^{\circ}\text{C}$ である。 80°C 以上の温度になった場合は、クロック周波数をゆっくりと下げることで加熱を抑えるスロットリングという仕組みが機能する³⁾。 85°C 以上になると待機時と同じ 0.6 GHz 程度のクロック周波数となり、温度上昇がほぼ抑えられる。また、CPU 負荷率は、コンピュータが論理 CPU を使用している割合であるが、1つのコア

*1 高知工業高等専門学校 電気情報工学科 4 年生

*2 高知工業高等専門学校 電気情報工学科 5 年生

*3 高知工業高等専門学校 教育研究支援センター 技術専門職員

*4 高知工業高等専門学校 ソーシャルデザイン工学科 講師

*5 高知工業高等専門学校 ソーシャルデザイン工学科 准教授

あたり最大100%とするため、4つのコアの場合、最大 400%となる。

本論文では、装置ボックス内での、CPU から発生する熱の伝搬について調べる。装置ボックス内に、多数の温度センサを設置し、各点での温度の時間変化を計測する。得られたデータから、熱平衡の式を近似し、熱の伝搬を可視化する。また、装置ボックス内への熱伝搬と CPU 負荷率との関係について考察する。さらに、熱対流の影響を推定するため、低圧下での計測を行い、熱対流の効果が抑えられた場合の熱伝搬について述べる。

2. 計測装置と実験環境について

2. 1 計測装置について

実験に使用する計測装置の構成を図 1 に示す。オンボードコンピュータとして Raspberry Pi を使い、ブレッドボード上に、A/D コンバータ、アナログ温度センサ、気圧センサ、温湿度・気圧センサモジュールを配置した。Raspberry Pi の複数のモデルの中から、無線 LAN 機能が含まれている 3 Model B (最大消費電力 12.5 W、電源定格 5 V) を使用した。外付けの無線 LAN アダプタを利用する場合、アダプタとの接続部の発熱が無視できないため、無線 LAN 内蔵モデルとした。装置ボックス内の温度の勾配や流れを求めるため、20 個の温度センサを装置ボックス内に取り付け、A/D コンバータを介して温度データを取得した。温度センサはアナログセンサで、温度範囲 $-25 \sim 85^{\circ}\text{C}$ での計測精度が $\pm 3^{\circ}\text{C}$ であるが、本実験においては温度の相対的な変化を見るため十分な精度である。また、A/D コンバータとして、10ch 分の入力が可能な TLC2543 を 2 個用いて、計 20 個の温度センサの出力を得た。

今回使用した 20 個の温度センサに関しては、事前実験で同一温度計測時のオフセット値を定め (表 1 参照)、データ解析時はオフセット値を差し引いて利用している。基準としては、最低温度を記録した ch6 の平均値を使用しており、温度センサ間の最大温度差は約 2.26°C であった。

2. 2 実験環境について

今回の実験では、熱対流の影響を比較するため、常圧 (1 気圧) と低圧 (約 5 hPa) 下での計測を行った。図 2 (上) に示すように、常圧では縦 15.5 cm、横 23.0 cm、奥行 14.5 cm の大きさの発泡スチロール製の箱を装置ボックスとした。低圧では、図 2 (下) に示すように、縦 21 cm、横 21 cm、奥行 21 cm の真空デシケータを装置ボックスとした。真空デシケータ内からは、外部供給電源を利用できる。真空装置としては、アルバック社の油回転真空ポンプ (G-101S) を用いており、最小で約数 hPa 程度まで気圧を下げることができる。また、装置ボックス内には、図 2 に示すように、

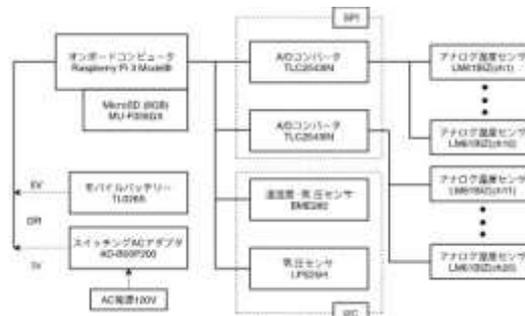


図 1 計測装置の構成図



図 2 実験装置の外観
(上) 常圧時の発泡スチロール製の箱内
(下) 低圧時の真空デシケータ内

表 1 同一温度計測時のセンサの平均温度と ch6 を基準としたオフセット

温度センサの示した温度の平均値									
ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	ch9	ch10
26.84	27.20	26.73	27.37	25.83	25.50	26.90	27.53	26.71	26.84
ch11	ch12	ch13	ch14	ch15	ch16	ch17	ch18	ch19	ch20
27.46	26.55	27.07	26.66	26.28	27.04	26.39	26.82	27.64	27.76
オフセット値									
ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	ch9	ch10
1.34	1.70	1.24	1.87	0.33	0.00	1.40	2.03	1.22	1.34
ch11	ch12	ch13	ch14	ch15	ch16	ch17	ch18	ch19	ch20
1.96	1.05	1.57	1.16	0.79	1.54	0.90	1.32	2.14	2.26

アナログ温度センサ 20 個を取り付けたブレッドボードを壁面と上下面に口の字型に配置した。熱平衡に至るまでの温度の時間変化を見るため、計測時間は 3 時間以上とした。

実験の手順としては、装置ボックス内に、Raspberry Pi やセンサ付きブレッドボードを図 3 のように配置する。装置ボックスのふたを閉めた後、Raspberry Pi を起動し、Tera Term により SSH 接続を行い、計測プログラムを実行する。その後、CPU 負荷率を上げる場合は、さらにコマンド実行し、計測の終了を待つ。指定時間が過ぎると自動的に計測を終えるようにした。計測プログラムでは 2 s 毎にセンサデータを取得し、CSV 形式としてマイクロ SD カードに保存する。保存されたデータは、計測終了後に、ファイル転送により PC に送る。真空デシケータを利用する場合は、真空デシケータのふたを閉めて低圧にした後、外部供給電源のスイッチを入れ、Raspberry Pi を起動させる。

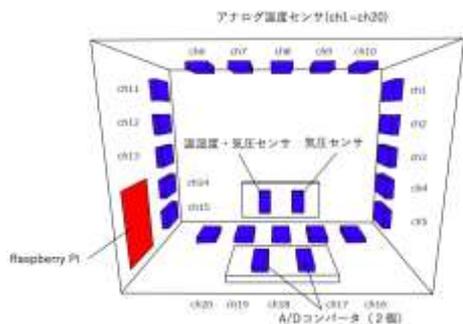


図 3 装置ボックス内のセンサ

3. 実験結果

計測した温度センサの時系列データの例を図 4 に示す。計測開始直後から温度は上昇し、時間が経つと熱平衡に達する。この温度変化は、 T を計測温度[°C]、 t を時間[s]とすると、指数関数 $T = a - be^{-\frac{t}{c}}$ で近似され、図 4 の例では、 $a=37.9$ 、 $b=5.8$ 、 $c=2173$ となった。 a は平衡温度 [°C]、 b は初期温度からの温度上昇[°C]、 c は平衡温度に達する時定数 [s]に相当する。測定位置や CPU 負荷率の値に関係なく、ほとんどの計測点で、同様の平衡温度に達する曲線で近似できた。ただし、実際のデータには、特に温度が高い場合や気圧が低い場合に、異常値が含まれることがある。図 5 に、異常値を含むデータ例を示す。図 5 (上) のように、高温で異常値が時折現れる場合には、正常と考えられる温度範囲のデータに対し、近似を行った。一方で、図 5 (下) のように、開始時刻以降、2 つの温度分布が常時現れている場合には、正しい温度変化を導出できないとして、近似を行わなかった。

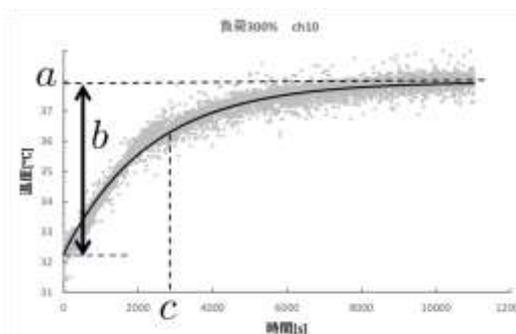


図 4 計測データ例
(CPU 負荷率 300%、ch10)

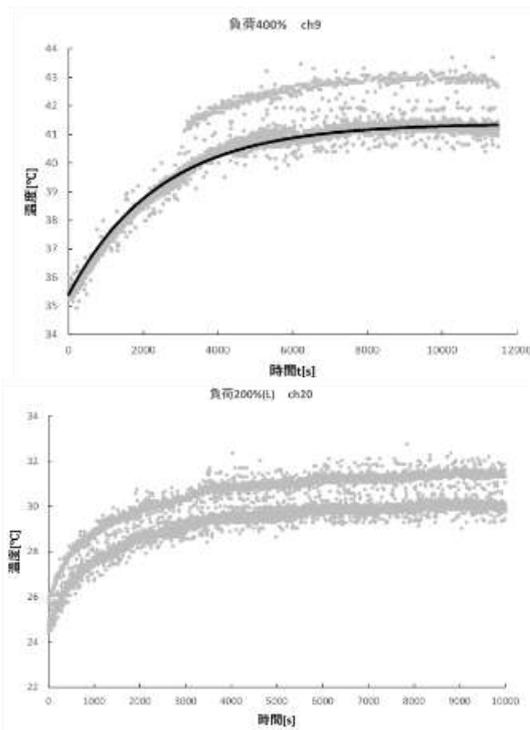


図 5 異常値を含むデータ例：
(上) 近似可能な場合
(下) 近似不可能な場合

3. 1 常圧実験における位置による温度変化の違い

各センサの位置による温度変化の違いを、図 6 に示す。図 6 (上)は、計測装置の下側(ch16~ch20)センサの温度変化であり、熱源からの水平距離が異なる場合を示している。初期温度はほぼ同じであり、全ての計測点で温度上昇が確認できる。平衡に達した温度は、熱源から近い側ほど高く、熱源から遠い側ほど低く、温度の空間勾配の向きは妥当である。図 6 (下)に、全てのセンサの最終温度を示す。下側のセンサに関しては、図 6 (左)で確認したように、熱源に近いほど温度が高い。上側のセンサに関しても、熱源に近いほど温度が高く、熱源から遠いほど温度が低い。鉛直方向(左側)の温度の違いに関しては、上側の温度が高く下側の温度が低い。鉛直方向(右側)の温度の違いも、同様の傾向をもつ。温められた空気が上昇し、空気の対流による熱が伝搬していると考えられる。一方で、熱源である Raspberry Pi の近傍

では、ch14 と ch15 の温度が低く、ch15 と ch20 の間には温度の不連続な領域ができています。熱の放射や対流を考えるだけでは、このような不連続な領域は発生しないため、3. 3 で詳細に調べる。図 6 は CPU 負荷率 300 % の例であるが、他の CPU 負荷率においても、空間位置による温度の違いに対する傾向はほぼ同じであった。

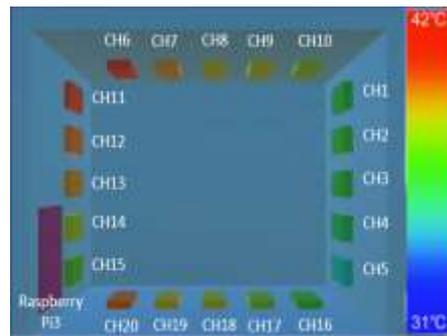
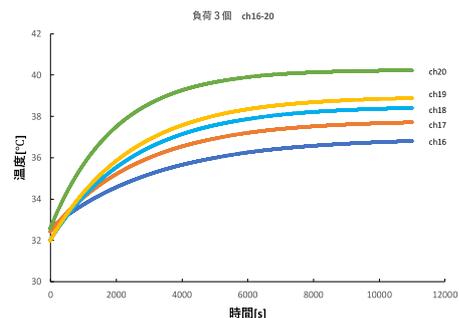


図 6 (上) 水平方向の位置 (下側) 毎の温度の時間変化

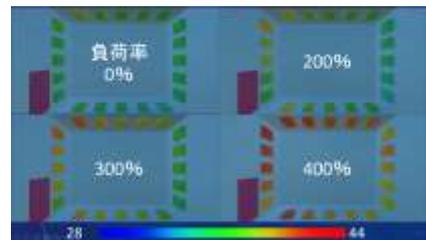


図 7 CPU 負荷率毎の計測装置内の最終温度

3. 2 CPU 負荷率による温度変化の違い

CPU 負荷率が異なる場合の装置内の最終温度を図 7 に示す。図 6 (下) で示したように、全ての CPU 負荷率において、水平方向の最終温度は、熱源に近いほど高く、熱源から遠いほど低い。鉛

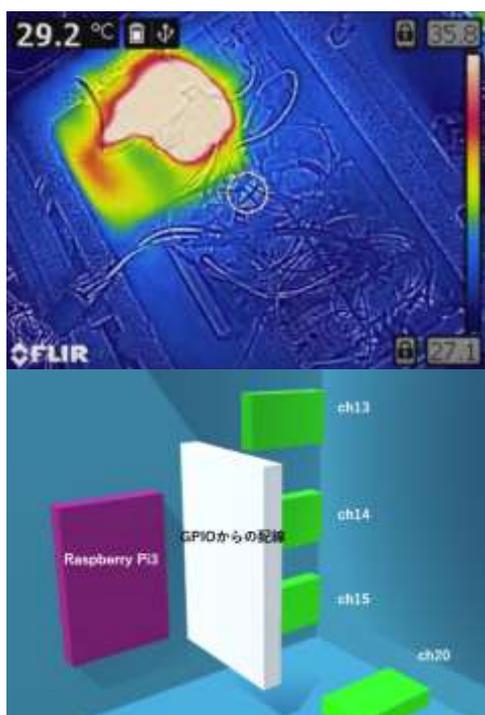


図 8 Raspberry Pi 周辺の
(上)サーモグラフィカメラによる撮像画像、
(下) 構造物の模式図

直方向の最終温度は、上側ほど高く、下側ほど低い。これらの傾向は、全ての CPU 負荷率でほぼ同様であった。一方で、CPU 負荷率が高いほど、最終温度は高い傾向がある。具体的には、最終温度は、CPU 負荷率 0% で 34.69~39.04°C、CPU 負荷率 200% で 34.04~42.18°C、CPU 負荷率 300% で 35.9~43.39°C、CPU 負荷率 400% で 37.99~46.45°C であった。また、CPU 負荷率 300% 時に見られた ch15 と ch20 付近の温度の不連続な領域は、全ての CPU 負荷率において観測された。

3. 3 Raspberry Pi 近傍の温度勾配

Raspberry Pi 近傍に位置する、ch15 と ch20 の間には温度の不連続な領域が見られた。その原因を探るため、Raspberry Pi 周辺をサーモグラフィカメラ (型番: FLIR C2、解像度: 80×60、30 万画素、計測温度範囲: -15~+150°C、温度精度: 2% @ 25°C) により撮像した。図 8 (上) に、真

上から見た装置内の温度分布を示す。Raspberry Pi 近傍およびその上方に温度が高い領域がある一方で、その右隣りに位置する鉛直方向のセンサ ch11~ch15 がある領域に熱が広がっていない。装置ボックス内の Raspberry Pi とセンサ周辺を確認すると、図 8 (下) の模式図に示すように、Raspberry Pi と、ch14 および ch15 の間に GPIO から配線が多く出ており、壁のようになっていることが分かった。この配線の壁により、熱が伝わりにくくなり、ch15 および ch14 の位置での温度上昇が小さくなったと考えられる。装置ボックス内における障害物などによる熱の伝搬特性に関しては、サーモグラフィカメラを利用することで、確認することができることが分かった。

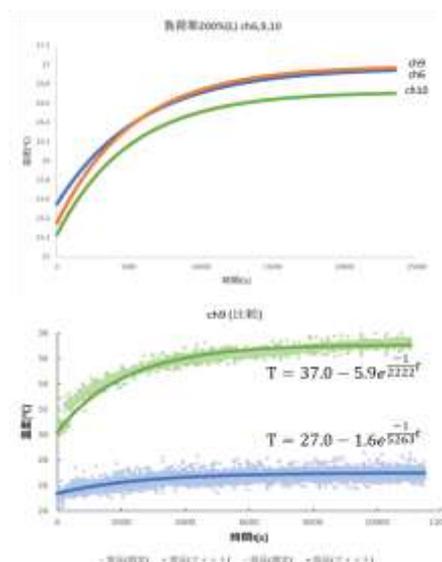


図 9 (上) 低圧時における、水平方向の温度の時間変化
(下) 低圧時と常圧時の比較 (CPU 負荷率 200%、ch9)

3. 4 低圧実験

常圧時と同様に、低圧下において、各センサ位置での温度の時間変化を調べた。温度の時間変化は、常圧時と同様に、指数関数 $T = a - be^{-\frac{t}{\tau}}$ でよく近似される。水平距離が異なる計測点での温度変化の例として、図 9 (上) に装置上側

(ch6~ch10) の温度変化を示す。ただし、低圧実験においては、図 4 (下) で示した指数関数での近似が不可能なセンサデータが多く、ch6、ch9、ch10 のみが指数関数による近似可能なデータであった。ch6 と ch9 の差ははっきりしないが、熱源から最も遠い ch10 は、他のセンサよりも温度が低い。

図 9 (下) で、低圧下 (約 5 hPa) と常圧下での温度の時間変化を比較する。低圧下では常圧下に比べて、温度上昇が小さく、時定数が大きくなる傾向がある。実際に、平衡温度 a [°C] と温度上昇 b [°C] は常圧時より低圧時の方がかなり低い。さらに、平衡温度に達する時定数 c [s] は、常圧時に比べ低圧時の方が大きくなっている。低圧下では、熱の伝搬量が少なく、伝搬はゆっくりであることが分かる。低圧時には、装置内の空気量が少なく、熱対流の影響が小さくなったためと考えられる。

4. Raspberry Pi の CPU 負荷率と熱伝搬の関係

本節では、CPU 負荷率と熱伝搬の関係を考察する。表 2 に、上側の水平方向位置における、CPU 負荷率毎の 3 つのパラメータ (平衡温度、温度上昇、平衡温度に達する時定数) をまとめた。各 CPU 負荷率での平衡温度(a)は、熱源に近い(ch6)側で大きくなっているが、それほど大きな差はない。各位置に関しては、CPU 負荷率が大きくなるほど、平衡温度は大きくなっており、3. 3 で述べた最終温度の位置による違いと同じ傾向を示している。温度上昇 (b) に関しては、熱源に近い側ほど大きい傾向はあるが、CPU 負荷率との関係は、ばらつきが大きく一定の関係があるとは言えない。平衡温度に達する時定数 (c) に関しては、熱源に近いほど時定数が小さく、熱の伝播が速かったことを示している。一方で、CPU 負荷率との関係は、0% から 300% までは時定数が順に小さくなっているが、CPU 負荷率 400% の時定数は 300% の時定数よりも大きく、負荷率 400% の熱伝搬が想定より遅かったことを示し、一般的

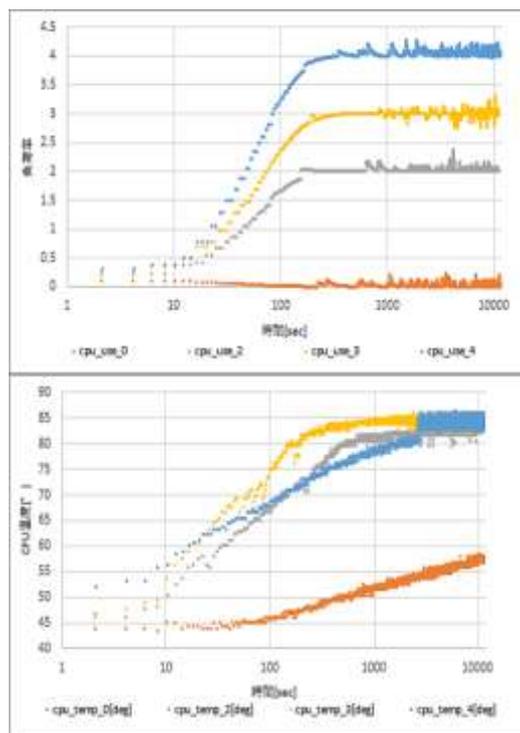


図 10 CPU 負荷率毎の
(上) CPU 負荷率 (実測値)、(下) CPU 温度

な描像と矛盾した結果となっている。

そこで、今回の装置ボックス内の熱源と考えている CPU の温度と、CPU 負荷率を確認する。図 10 (左) に CPU 負荷率 (実測値)、図 10 (右) に CPU 温度を示す。初動の変化を見やすくするため、横軸の時間は対数表示としている。時間の経過とともに、CPU 負荷率は指定した CPU 負荷率に収束していることが分かる。一方で、CPU 温度と CPU 負荷率との関係は複雑である。CPU 負荷率 0% では、ゆっくりと温度上昇しており、最終時刻でも CPU 温度は 60°C 以下である。CPU 負荷率 200% では、CPU 負荷率 100% よりも温度上昇が急であり、CPU 温度が 80°C になると温度上昇が急激に緩やかになり、85°C 程度まで緩やかに温度上昇している。これは、RaspberryPi3 のスロットリングという機能であり、クロック周波数を抑えることで、CPU 温度上昇を抑えているためと考えられる。CPU 負荷率 300% では、CPU 負荷率 200% よりやや急激な温度上昇をしながら、

CPU 温度は 80°C を経て 85°C に収束している。一方で、CPU 400% では、起動後の CPU 温度は最も高かったものの、CPU 負荷率 0% と同程度の温度上昇で 80°C に到達し、その後は、値が 82°C ~ 86°C 程度に突然切り替わっている。CPU 負荷率 400% 時には、CPU 負荷率の実測値が十分大きいにも関わらず、CPU 温度の上昇の仕方が特異であり、CPU のクロック周波数などが制限を受け、熱の発生が抑えられた可能性が考えられる。この点を検証するため、クロック周波数を取得するように設定し、再度実験を行った。CPU 負荷率 300%、400% で実験を行った結果を図 11 に示す。

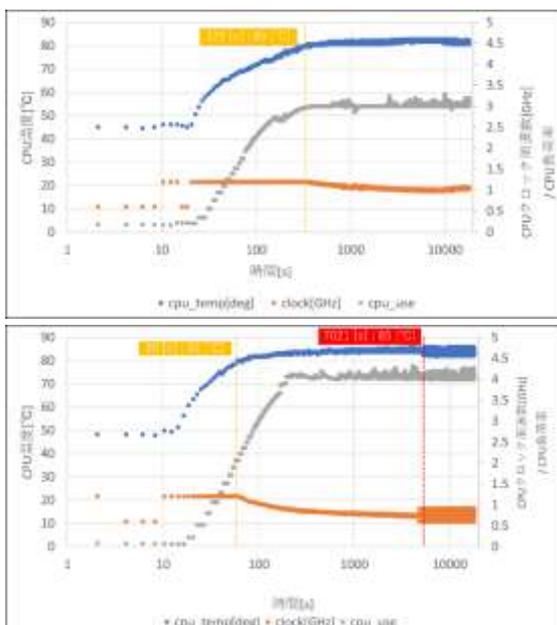


図 11 CPU 温度・CPU 負荷率・クロック周波数、

(上) 負荷率 300%、(下) 負荷率 400%

まず、CPU 負荷率 (灰色) に関しては、20 秒あたりから増加し始め、各々 300%、400% に到達後はほぼ一定となっている。CPU 負荷率の増加が始まると、CPU 温度 (青色) は、実験開始時の 40°C 台から急激に増加し始め、徐々にゆるやかになりつつ 80°C に達する。その後はゆるやかに上昇し、CPU 負荷率 300% では 82°C 程度に収束し、CPU 負荷率 400% では開始 7021 秒で 85°C に達し、

その後は、82°C ~ 86°C の間で振動していた。CPU のクロック周波数 (橙色) は、CPU 負荷率の上昇前には待機時の 0.6 GHz であったが、CPU 負荷率の上昇に伴って、1.2 GHz となり、80°C に達すると徐々に低下した。CPU 温度が 80°C 以上でのクロック周波数の低下はスロットリングの仕組みによるものである。負荷率 300% では 1.0 GHz 程度まで低下したが、負荷率 400% では 0.7 GHz 程度に低下し、CPU 温度が 85°C に到達した後は、待機時の 0.6 GHz と 0.9 GHz の間で振動していた。CPU 温度が 85°C 程度での振動は、前述した CPU 温度の増減に伴って、クロック周波数の切り替わりが発生していると考えられる。

図 10 では、CPU 負荷率 400% 時に、2 つの特異な現象が見られた。1 つは、CPU 温度が 80°C に到達直後に、82°C 程度へ突然切り替わり、その後は 82°C ~ 86°C の間で振動していた点である。図 11 では、突然の切り替わりは確認できなかったものの、82°C ~ 86°C の振動は確認され、クロック周波数の引き下げの下限で、待機時と動作時の振動を繰り返していると考えられることが分かった。もう 1 つは、CPU 負荷率 400% の方が、CPU 負荷率 400% よりも温度勾配が小さかった点である。ただし、図 11 では負荷率 400% の方が負荷率 300% よりも温度勾配が大きく、元々想定される結果が得られた。図 10 での CPU 負荷率 400% の振る舞いについては、今後も検討する必要がある。表 2 で得られた熱伝播の様子は、CPU 負荷率に単純に比例しなかったが、CPU 温度 80°C 以上では、CPU 温度が CPU 負荷率に加えて、クロック周波数などの変動に影響を受けることによる。

今回の計測を通して、CPU からの熱伝搬は、空気の対流により大きく影響を受けており、上方に温められた空気が留まっていた。装置ボックス内の最終温度で、10°C 程度の差があるため、送風機の装置内天井への設置などの対応により、より適正な温度管理が可能となる。また、CPU 負荷率の強制的な増減による温度管理に関しては、CPU 温度、CPU 負荷率、およびクロック周波数

などをモニタリングすることで、より効率的で効果的な温度管理が可能であることを示唆できたと考えている。

5. まとめ

本実験では計測装置内に 20 個の温度センサを配置し、Raspberry Pi より発生する熱の伝搬の様子を常圧、低圧下で計測した。取得データより、各測定点での温度の時間変化を指数関数で近似でき、平衡温度、温度上昇、平衡温度に達する時定数の 3 つのパラメータを導出できた。計測装置内の空間分布を調べることで、熱源である Raspberry Pi からの距離に反比例して、最終温度は小さくなること、空間的に高い場所ほど、温度が高い傾向があることが分かった。基本的には、CPU 負荷率を大きくするほど、ボックス内の平衡温度は高くなり、時定数が小さく（熱の伝播が速く）なる傾向が確認できた。ただし、CPU 温度は CPU 負荷率とクロック周波数の双方から影響を受けるため、CPU 温度、CPU 負荷率、及びクロック周波数をモニターして温度管理を実施する必要がある。また、装置ボックス内で見つかった温度の不連続な領域は、サーモグラフィカメラにより温度の空間分布を調べることで、配線が壁のように熱の伝播を遮っていることを確認できた。また、低圧下では、温度上昇が小さく、熱平衡に達する時定数が大きい特徴があり、常圧下に比べて、熱伝搬が抑えられることが分かった。

謝辞

実験実施にあたって、株式会社八光電機より助成を受けました。

参考文献

1) 竹内純人、コンセント単位での計測を可能としたフリーソフトウェアツールによる消費電力値自動収集システムの実装と改善、電気通信大学紀

要, 28.1, 61-69, 2016

2) 笹岡由唯, 上園波輝, 川上舞帆, 坪内麟太郎, 横山有太, 上田真也, 高田拓, 成層圏気球のための低温・低圧下におけるオンボードコンピュータ Raspberry Pi の発熱と保温性能に関する評価実験, 高知工業高等専門学校学術紀要, 63, 35-44, 2018

3) ZDNet ニュースサイト : (2018/11/26)
<https://www.zdnet.com/article/no-your-raspberry-pi-3-wont-overheat-in-everyday-use-says-its-creator/>