

小型・省電力センサによるスマート養蜂システム

吉竹隆也¹ 佐藤証¹

概要： 養蜂のためのスマートシステムの実用化に向け、無線通信機能を備えた 32 ビットマイコン ESP32-WROOM-32 を実装した小型・省電力センサモジュールを開発した。マイコンを低消費電力の Deep Sleep モード動作させることで、モバイルバッテリーや乾電池で数ヶ月の動作を目指している。接続数の限られたモバイルルータでも、センサ側で通信を順次切り替えて数十台の接続でき、センサデータは軽量通信プロトコル MQTT で、Raspberry Pi サーバに送信されスマートフォン等で遠隔でのモニタが可能である。養蜂場に設置したセンサにより、季節、時刻、天候等による巣箱の重量変化から蜂蜜の採取や、逆に給餌をするタイミングを計ることが可能になり、温湿度は巣箱内のミツバチの状態を把握するのに有用であることが示された。

1. はじめに

ミツバチは世界の食料の 9 割を占める作物種の受粉に携わり、その 7 割を担っていると言われる。養蜂は持続可能な社会の実現を目的とした SDGs の活動にも大きく貢献し、ビルの屋上を利用した都市養蜂も静かなブームとなっている。しかし、養蜂は経験とカンに頼るところが大きく、プロであっても養蜂箱内のハチの状態を把握するには、図 1 のように巣枠を取りだして目視する内検の必要がある。しかしこれはミツバチにとって大きなストレスとなる。また、雨天は巣箱内が濡れてしまうので、内検は天気の良い日しか行えない。ミツバチやは眠らず夜も巣箱の中で巣作り、育児、蜜の濃縮等様々な活動を行っているが、夜間の状態も把握することはできない。



図 1 養蜂箱の内検の様子

農業の IoT 化は大規模農場や植物工場に浸透している一方、その農業に不可欠な養蜂の IoT 化に関する研究や製品化は遅れている。センサを実装した養蜂箱を用いて、ミツバチの生態を調べる研究もおこなわれているが[1]-[3]、専用の巣箱を用いており、一般の養蜂家が使用できるものではない。また国内では、温湿度を測定する製品が社から販売されているだけである[4]。

そこで本研究では、IoT 技術を養蜂現場で誰でも利用できるよう、小型で安価なセンサモジュールと、それを用いた養蜂システムを開発する。センサモジュールは省電力化

により、外部電源が不要なバッテリー駆動とした。また、Wi-Fi 経由でデータをシングルボードコンピュータの Raspberry Pi 上に実装した管理システムに送信し、スマートフォン等でどこでもデータをモニタ可能とする。そして開発したセンサを実際に養蜂現場に設置し、その有用性を評価する。

2. 従来の研究

IndusBee 4.0 [1]では巣枠に、重量(HX711)、温湿度(DHT11/22)、音響(SPH0645LM4H)の各センサ、そして Raspberry Pi Zero W を取り付けた図 2 のセンサを開発している。電力は外部電源から有線で供給するが、野山の養蜂場や都市養蜂を行うビルの屋上では電源が確保できないことも多い。また、巣枠にセンサやコードをつけたままだと、内検作業の邪魔なるだけでなく、採蜜のための巣枠を遠心分離機にかけることができないという問題がある。

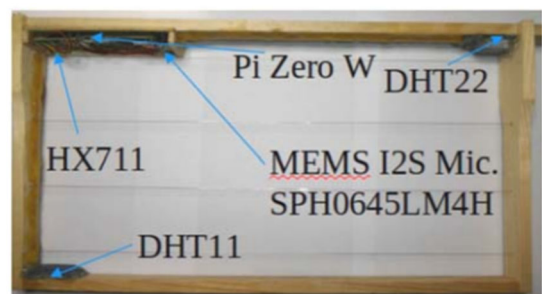


図 2 センサとマイコンを実装した巣枠 [1]

文献[2]では、温湿度・重量・音響センサに加えて二酸化炭素センサを導入している。図 3 左のように巣箱の裏側にセンサを搭載した大きなモジュールを取付け、図 3 右のように近くに気象センサを搭載した通信モジュールを設置している。大きなセンサと通信モジュール、そしてそれに必要な電源を供給する必要があるため、養蜂での利用を目的としたものではなく、ミツバチの生態観察用の実験システムである。

文献[3]では、図 4 のように巣門の両側に設置したマイク

¹ 電気通信大学

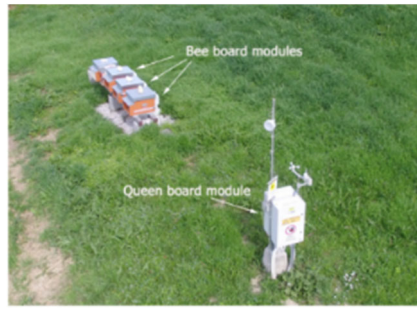


図3 巣箱に取付けたセンサ(左)と通信モジュール(右) [2]



図4 巣門の両脇に設置したマイク [3]

で集めたミツバチの音を、機械学習によってコオロギの音や周辺ノイズと区別する研究をしている。違う種類の音を識別するだけで、蜜蜂の状態の把握を行うものではない。また、SDカードに保存した音声データを後で解析するため、リアルタイムでの処理は行えない。

図5に本研究のシステム概要を示す。センサモジュールのプロセッサにはWi-FiとBluetoothを内蔵し、IoT用として広く使われている32ビットマイコンESP32-WROOM-32(以下ESP32) [5]を用いた。巣箱内外の温湿度、巣箱の重量をセンサで測定し、IoT向けの軽量通信プロトコルMQTT(Message Queuing Telemetry Transport)でモバイルルータに

よりインターネットを経由してデータをサーバに送信する。サーバにはスマートホーム用プラットフォーム Home Assistant [6]を搭載したRaspberry Piを用い、フローベースのプログラミング環境 Node-RED で全体の管理を行う。MQTTブローカ mosquitto [7]で受信したデータは時系列データベース Influxdb [8]に保存され、Grafana [9]やNode-RED [10]のダッシュボードで可視化されたデータは外部からスマートフォンやPCを用いて遠隔でのモニタが可能である。

図6に温湿度センサモジュールとそれを養蜂箱に設置した状態を、図7に重量センサと養蜂場での使用している様子を示す。温湿度センサはHTU21D [11]をESP32 miniボードとタブレットケースに封止し3,400mAhのリチウムイオン電池一本で駆動している。重量センサは市販の体重計の筐体とひずみセンサのロードセルを流用し、ロードセル用ADコンバータHX711とESP32 miniを組み込んだ。電源は、体重計の電池ボックスを流用して単4アルカリ電池三本で駆動している。30分に1回、10秒弱起動してセンサデータを送信し、それ以外の時間は、ESP32をDeep Sleepモードに移行すること

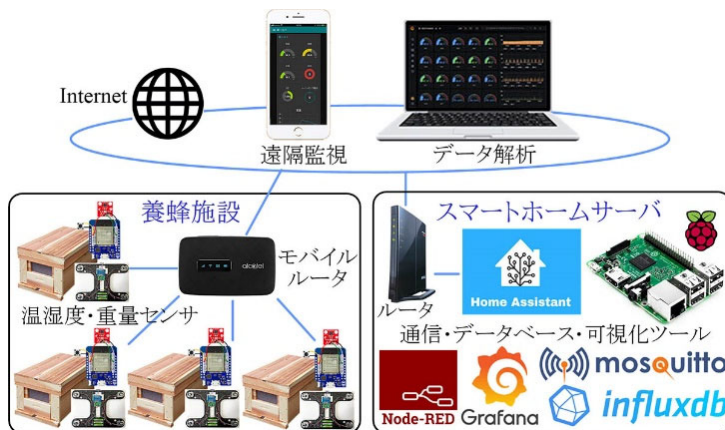


図5 スマート養蜂システムの概要



図6 温湿度センサモジュール



図8 温湿度センサとモバイルルータの接続

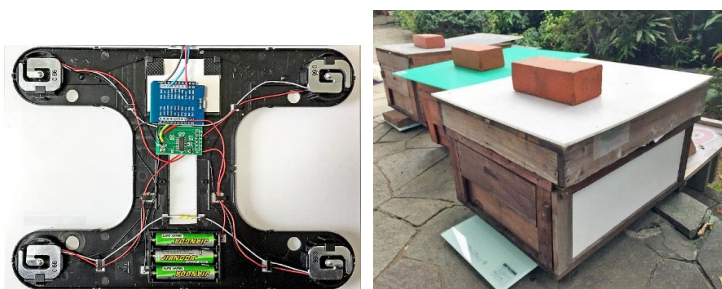


図7 重量センサモジュール



図9 Grafanaの湿度表示画面

で消費電力を削減した。

モバイルルータは同時に 10 台のデバイスしか接続できないが、Deep Sleep モードにしながらかデバイスを順次切り替えることで、20 台の温湿度センサと 20 台の重量センサの接続に成功している。図 8 に温湿度センサ 20 台をモバイルルータに接続している様子と、そのデータを Grafana で表示した画面を示す。センサモジュールは接続に失敗すると、約 30 秒間リトライを繰り返すようにプログラムされている。この接続の順番待ちにより、各モジュールが Deep Sleep モードに入るタイミングが少しずつずれていくことになる。それによって、全てのセンサの電源を同時に入れても、自動的に接続の順番とタイミングが調整されることになる。

温湿度センサをおよそ 22℃と 15℃の部屋で Deep Sleep モードに入れずに動作させたところ、22℃の部屋は問題なかったが 15℃の部屋は 16℃と 1℃高くなった。これは、ESP32 mini の発熱が短いワイヤを伝わって温湿度計に伝わっていることが原因であった。Deep Sleep モードで ESP32 mini を停止するサイクルを伸ばしていったところ、5 分でこのような温度の上昇はなくなった。なお、実際の運用での起動サイクルは 30 分~1 時間とするのでこのような発熱の問題は生じない。

体重計の天板は小さいため巣箱全体をそのまま載せることはできず、図 7 に示したように巣箱の後脚だけを載せ、前脚は地面についている。なお、巣箱内部に雨が入らないように後脚にはレンガ等を挟んで前に傾けるため、この状態でミツバチへの影響はない、後脚だけを体重計に載せても巣箱が水平であれば、前後の脚にかかる重量は同じであるが、しかしこのように傾けて設置すると、前部のほうが重くなる。この影響について検討した。

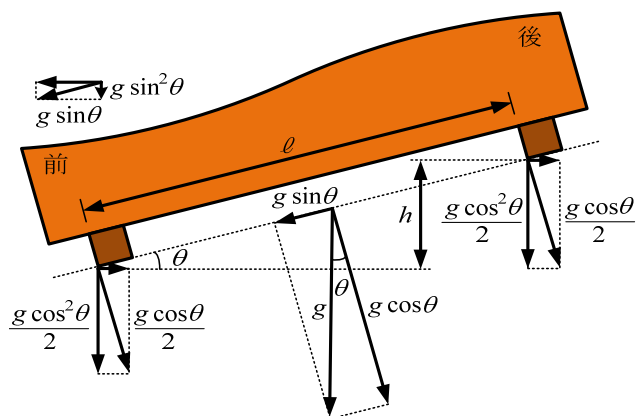


図 10 傾いた巣箱の前後の脚にかかる荷重の計算

重量 g の養蜂箱が水平におかれ、前後の脚の中央に重心があれば、それぞれの脚にかかる垂直方向の荷重は当然 $g/2$ となる。後部の脚が体重計の上に置かれ前に θ 傾いたとき、図 10 に示したように、両脚を結んだ線から垂直方向の成分は $g \cos q$ となり、これの半分がそれぞれの脚にかかるので $(g \cos q)/2$ となる。体重計にかかるのはその縦方向成分の $(g \cos^2 q)/2$ である。また、前部の脚には、 g の水平方向成

分 $g \sin q$ のさらに縦成分 $g \sin^2 q$ も加わることになる。

$$\text{前脚} : \frac{g \cos^2 \theta}{2} + g \sin^2 \theta$$

$$\text{後脚} : \frac{g \cos^2 \theta}{2}$$

前脚と後脚の荷重を合計すると、 $g(\cos^2 q + \sin^2 q) = g$ であるから巣箱の重量 g に等しいことが確認できる。

以上から、巣箱の傾きが θ で重量センサの値が x のとき、巣箱の重量 g は次式で得られる。

$$g = \frac{2x}{\cos^2 \theta}$$

図 10 に示したように脚の間隔を ℓ 、体重計の高さを h とすると三平方の定理から、

$$\cos^2 \theta = \frac{\ell^2 - h^2}{\ell^2}$$

したがって、

$$g = \frac{2x\ell^2}{\ell^2 - h^2}$$

となる。一般に使用されるラングストロス式養蜂箱の ℓ はおよそ 45cm で、体重計の高さは 2.4cm であった。この時の傾き 3.06° を上式に当てはめると、

$$g = 2x \times 0.997$$

レンガを挟むと約 6cm 高くなるが (傾き 7.66°)、それでも

$$g = 2x \times 0.982$$

である。したがって、傾きはさほど気にする必要はなく、センサの値の 2 倍を養蜂箱の重量と見積もればほぼよい。なお、体重計には図 7 のように 4 つの歪センサがあり、重い箱を端に載せると体重計が傾き、いくつかのセンサが地面から離れて浮いて正しく計測できなくなることがあるので注意が必要である。

上では巣箱の重心は前後の脚の中央にあるものと仮定したが、現在使用している巣箱の脚は前に寄っている。したがって重心は脚の中央より後ろにずれ、後脚にその分大きな荷重がかかることになる。巣箱が水平に置かれているとき、前後の脚の間隔を ℓ (図 10 では 45.0cm)、巣箱の重心 (巣箱の長さの半分の位置としているが、実際にはここからずれている可能性のあることに注意が必要) から前脚までを x (図 10 では 26.5cm)、後脚までを y (図 10 では 19.0cm) とすると、それぞれにかかる重量の比率は、 y/ℓ ($=19.5/45.0=43.3\%$) 及び x/ℓ ($=25.5/45.0=56.7\%$) となる。巣箱の脚の位置はものによって異なるので、これを測って計算するか、実際に前後の脚にかかる重量を測って比率を求めておく必要がある。

3. センサモジュールの低電力化

5V 昇圧回路を有したモバイルバッテリーアダプタに図 11 のように 3.7V×3,400mAh のリチウムイオン電池 (NCR18650B) を挿入し、温湿度または重量センサモジュール

ルの USB 端子に接続して 30 分間隔の Deep Sleep モード動作させた。ESP32 の Deep Sleep 時の消費電力は $3.3V \times 10 \sim 150\mu A$ と非常に小さく、Wi-Fi 通信時のセンサモジュール全体の消費電力の測定値は 133mA であった。したがって 30 分間に 10 秒間起動し、その間ずっと Wi-Fi 通信をしていたとしても、一時間の平均の電流は、

$$(130mA \times 20s + 150\mu A \times 3,580s) / 3,600s = 0.871mA$$

となる。電圧変換 $3.7V \rightarrow 5V \rightarrow 3.3V$ によるロスはあるが、バッテリー容量を $3.3V \times 3,400mAh$ としたときに、単純計算では

$$3,400mAh / 0.871mAh = 3,904h = 162 \text{ 日}$$

となり 5 ヶ月程度は動作するはずである。しかし実際には 2 週間ほどでバッテリーが切れてしまった。



図 11 モバイルバッテリー

そこで、ESP32 mini 上の主要コンポーネントの Deep Sleep 時の消費電力を詳細に調べたところ、3.3V レギュレータ ASM1117-3.3 [14] が定常的に $5V \times 4.5mA$ (データシートでは 5mA) を消費していることがわかった。なお、ESP32 の消費電力は $4.6\mu A$ と Deep Sleep モードの $10 \sim 150\mu A$ よりも小さかった。これは ULP が Deep Sleep 時にデータを保持するリアルタイムメモリを使用していなかったため、Deep Sleep よりも低消費電力の Hibernation モードに入った

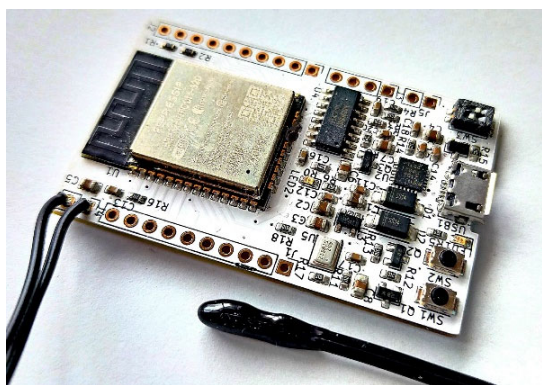


図 12 オリジナルの ESP32 センサモジュールとサーミスタ

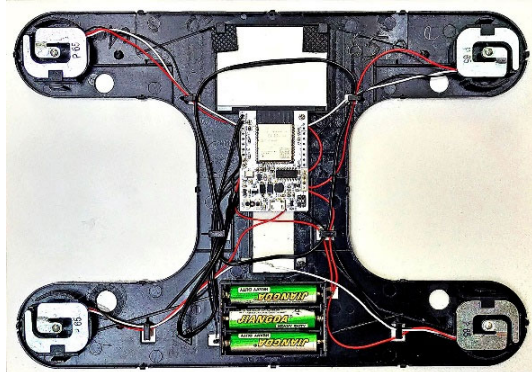


図 13 センサモジュールを実装した体重計

ものと考えられる。ESP32 mini 以外の市販ボードの利用も検討したが、ほぼ ASM1117-3.3 が使われていた。

そこで低電力化を主目的とした図 12 のオリジナルボードを作成した。ESP32 と各種センサを実装しながらサイズは $33.1mm \times 48.7mm$ と、ESP32 mini の $30mm \times 38mm$ より一回り大きいだけである。低損失 3.3V レギュレータの TLV75533PDBVR [15] を用いており、待機時の電流は $25\mu A$ と ASM1117-3.3 の $1/200$ である。USB コネクタは ESP32 へのコードの書き込みとデバッグに使い、モバイルバッテリーを接続してセンサデータの測定を行う際には信号線は使用されない。そのため、USB-シリアル変換 IC をサスペンドモードに入れられず、無駄な電力が消費されてしまう。そこで、バッテリー駆動時には USB-シリアル変換 IC に電力が供給されないように、モードを切り替えるスイッチを付加している。図 6 のタブレットケースに ESP32 ボードを内蔵したセンサモジュールは、ケースで遮られる面積部分の巣牌をミツバチが使用できないという欠点があった。そこで、センサボードは別の場所に置き、図 12 のようにサーミスタを接続してコードを温度測定位置まで引っ張ることとした。ボードには重量センサ用 AD コンバータも実装しており、図 13 のようにそのまま体重計に実装することができる。また小型の MEMS (Micro Electro Mechanical Systems) マイクも実装しており、巣箱内の音声のサーバへの MQTT 送信や、Bluetooth 接続で再生が可能である。

4. 養蜂施設での運用

開発したセンサは、図 14 の原宿の商業ビル屋上にある養蜂施設「BEETOPIA はらじゅく」[16] や図 15 の赤坂の地域ボランティア活動「みつばちあ」[17] の施設に設置し、データを取得した。

図 16 は BEETOPIA に設置した 7 台の重量センサで取得した、3 月末から 4 月初旬の 10 日間のデータの一部を Grafana で表示した画面である。桜のシーズンで天候にも恵まれたため、ミツバチが盛んに花蜜を集め、日毎に巣箱の重量が増加していることがわかる。右端で垂直に重量が落ちているのは、ここで巣箱から蜂蜜を採取したためである。なお、採取は春に 4-5 回行い、そのタイミングを見計らうのは非常に重要であるが、これまでは巣箱を開けて内検したり、図 17 のように後ろ手に巣箱を持ち上げて重さを調べたりする必要があった。巣箱の重量は常に単調増加しているわけではなく、詳しく見ると巣箱毎でも変化が異なる。一日の間では、夜は巣箱の中で蜜の水分を蒸発させているので朝にかけて次第に重量が減り、多くの働きバチが採蜜に外に出ている朝 10 時頃が最も軽い。天気の日には気温が上昇して花が開くことで採蜜量が増え、曇りの日は少ない。また図 16 の巣箱からの蜂蜜の採取の前日の天気は雨で、ハチは蜜が集められず食料として消費したため重量が減少している。このように天候の重量への影響は非常に大



図 14 BEETOPIA はらじゆくに設置した重量センサ



図 15 みつばちあの養蜂施設

きいことがグラフからよくわかる。

なお、図 13 のオリジナルのセンサモジュールを実装した体重計は、本稿執筆時点で1ヶ月半動作している。電源には容量およそ 1.5V×1,000mAh の単四アルカリ乾電池電池を3本用いており、全体でおよそ 4.5V×1,000mAh となる。図7のESP32 mini を実装した重量センサを、容量がおよそ3倍 (3.7V×3,400mAh) の図11のリチウムイオンバッテリーで駆動したときの持続期間が2週間程度であったので、オリジナルボードの省電力性能が示された。

図18は8月の「みつばちあ」の巣箱内外の温湿度と重量の、Node-RED ダッシュボードによるスマートフォン表示画面である。花の少ない夏はハチたちが消費する蜜の量が採取の量を上回り、重量が次第に減少している様子がわかる。なお縦のスパイクノイズは内検のために養蜂箱を開けたときの重量変化である。梅雨以降はこのように重量が減ることもあり、糖蜜や花粉を給餌するなどして群勢を維持しないと、越冬できないので注意が必要である。外気温は約30~40℃、外気湿度は約50~90%と大きく変化しているのに対して、巣箱内部は35℃で65%とほぼ一定に維持されている。暑い日中にハチたち巣門の前に並び、羽で空気を送り込んで換気して温度調節をしている。また冬眠しないハチは巣箱の中で固まって羽の筋肉を震わせて発熱し、35℃を保っているが、その温かい環境を好むダニの繁殖にもつながることとなる。ダニが繁殖している養蜂箱は蓋の内側に多くの水滴がついているようで、ダニに対抗するためにハチが温度を上げた結果発生した結露ではないかと見られている。このように重量や温湿度のモニターで、巣箱を開け



図 16 BEETOPIA はらじゆくの重量センサデータ



図 17 巣箱の重量のチェック

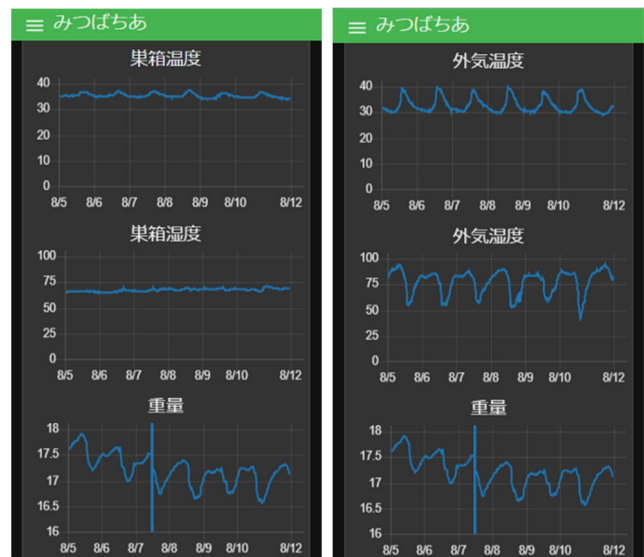


図 18 みつばちあの8月のセンサデータ

て内検しなくても、蜂蜜の採取の時期の判断や、群勢・健康状態等を推察することができ、センサモジュールの有効性が示された。

5. むすび

ミツバチの生態の研究のための特殊な養蜂箱ではなく、養蜂現場での実利用を目的に乾電池やモバイルバッテリーで駆動可能な低電力センサモジュールを開発し、性能及び有効性を評価した。IoT 向 32 ビットマイコン ESP32 と低損失レギュレータ、温度センサ、重量センサ、MEMS マイクを

実装したセンサモジュールを Deep Sleep モードで動作させることで、市販の ESP32 ボードに対して少なくとも 1/10 の消費電力を実現した。取得したセンサデータは Wi-Fi ルータを経由して IoT 向通信規格 MQTT で Raspberry Pi サーバに送信される。モジュールの接続を順次切り替えることで、同時接続数 10 台のモバイルルータでも、温湿度センサ 20 台と重量センサ 20 台の接続を可能とした。サーバが受信したデータは、フローベースプログラミング環境 Node-RED のダッシュボードや、時系列データベース InfluxDB とデータ可視化ツール Grafana を用いて、スマートフォンや PC で遠隔モニタを可能とした。

開発したセンサモジュールを 2ヶ所の養蜂施設に設置してデータを取得した。桜の開花以降、貯蜜により巣箱の重量が日に日に増加し、時間帯や天候によってことなる変化の詳細が把握できた。これらは春の巣箱からの蜂蜜採取や、梅雨以降で花の少ない時期の給餌のタイミングを計るのに非常に有用である。またミツバチは巣箱内の温湿度を一定に保っているが、その変化により群の状態を把握できると考えられ、データの解析を進めていきたい。また温湿度や重量は急激には変化しないが、MEMS マイクで巣箱内の音声を取得することで、急な状況変を検知できる可能性があり、これも今後の課題である。

本研究室ではビルの屋上等の空きスペースにセンサを自走した水耕栽培装置を設置し、緑のスペースを拓けるスマート都市農業の研究[18]を行っており、スマート都市養蜂との融合も進めていく予定である。

参考文献

- [1] A. Koning: IndusBee 4.0 – Integrated Intelligent Sensory Systems for Advanced Bee Hive Instrumentation and Hive Keepers' Assistance Systems, Sensors & Transducers Journal, Vol.273, No. 9-10, pp. 109-121, 2019.
- [2] S. Cecchi et al.: Multi-sensor platform for real time measurements of honey bee hive parameters," ICP Conf, 2019.
- [3] V. Kulyukin, et. al.: "Toward Audio Beehive Monitoring: Deep Learning vs. Standard Machine Learning in Classifying Beehive Audio Samples," Applied Sciences, Vol.8, No. 9, pp.1-33, 2018.16.
- [4] アドダイス: 養蜂業支援 AI – Bee Sensing, <https://ad-dice.com/solution_category/beesensing/> (参照 2021-05-01).
- [5] Espressif Systems: ESP32 Series of Modules, <<https://www.espressif.com/en/products/modules/esp32>> (参照 2021-05-01).
- [6] Home Assistant, <<https://www.home-assistant.io/>> (参照 2021-05-01).
- [7] Eclipse Mosquitto An open source MQTT broker, <<https://mosquitto.org/>> (参照 2021-05-01).
- [8] Influxdata: Act in Time. Build on InfluxDB., <<https://www.influxdata.com/>> (参照 2021-05-01).
- [9] Grafana Labs: Grafana The open observability platform, <<https://grafana.com/>> (参照 2021-05-01).
- [10] Grafana Labs: Grafana The open observability platform, <<https://grafana.com/>> (参照 2021-05-01).
- [11] JS Foundation: Node-RED Low-code programming for event-driven applications, <<https://nodered.org/>> (参照 2021-05-01).
- [12] TE Connectivity: HTU21D Digital High Accuracy RH/T Sensor,

<<https://www.te.com/usa-en/product-CAT-SC0004.datasheet.pdf>> (参照 2021-05-01).

- [13] AVIA Semiconductor: HX711, <<http://en.aviaic.com/detail/730856.html>> (参照 2021-05-01).
- [14] Advanced Monolithic Systems: AMS1117 1A LOW DROPOUT VOLTAGE REGULATOR, <<http://www.advanced-monolithic.com/pdf/ds1117.pdf>> (参照 2021-05-01).
- [15] TEXAS INSTRUMENTS: TLV755P 500-mA, Low I_Q, Small Size, Low Dropout Regulator <https://www.ti.com/lit/ds/symlink/tlv755p.pdf?ts=1619514599488&ref_url=https%253A%252F%252Fwww.google.com%252F> (参照 2021-05-01).
- [16] BEETOPIA はらじゅく, <<https://www.team-mitsubacheer.com/>> (参照 2021-05-01).
- [17] チームみつばちあ, <<https://www.team-mitsubacheer.com/>> (参照 2021-05-01).
- [18] 佐藤証: 都市型農業のサービス産業化を実現する水耕栽培システム, DICOMO2018, 5B-3, pp.925-928, 2018 年 7 月.

本研究の一部は、令和 2 年度日本中央競馬会畜産振興事業の支援を受けて実施された。