

試験機運転状況の可視化に関する研究

小和瀬登・石黒 聡・黒岩広樹*

The study about visualizing operation conditions of testing equipments
Noboru KOWASE, Satoshi ISHIGURO, Hiroki KUROIWA

現在、IoT 技術は生産現場等において仕事の効率化に欠かせない存在になっており、群馬産業技術センター内においても機器運転状況の”見える化”による業務効率化が求められている。

本研究では、IoT を活用した業務の効率化と試験機のトラブル対応の迅速化を実現するために、離れた場所でも試験機の運転状況が把握できるようなシステムの構築を行った。

キーワード：IoT、見える化、センシング、ZABBIX、Raspberry Pi、
ネットワーク表示灯

Currently, IoT technology is indispensable for improving efficiency of work at production sites and the like. Even within the Gunma Industrial Technology Center, there is a need to improve operational efficiency by "visualizing" the operating status of equipments.

In this study, we constructed a system that can grasp the operating status of testing equipments even at a remote location in order to improve the efficiency of work using IoT and speed up the troubleshooting of the testing equipments.

Keywords: IoT, Visualizing, Sensing, ZABBIX, Raspberry Pi, Network indicator

1 はじめに

IoT 技術は生産現場等において仕事の効率化に欠かせない存在になっており、群馬産業技術センター（以下「センター」という）においても企業等への IoT 支援を進めているところだが、センター内では当技術の活用がほとんど進んでいない。2019年1月にセンター内で「停止したことを知りたい機器」についてアンケート調査をしたところ、20機以上の試験機器について要望があり、センター内においても機器運転状況の”見える化”による業務効率化が求められている。

また、センターの試験機は、職員室から離れている部屋に設置されているため、トラブル発生時に即時に状況把握ができない問題がある。

さらに、長期の連続試験の場合、度々不測の停止が発生してしまうと、発生タイミングによっては試験期間が2、3日後ろにずれてしまうことがある。そのため、試験依頼者の要望に応えられないことがあり、多くの機会損失が発生していると考えられる。

そこで、本研究では、IoT を活用した業務の効率化と、不測の停止時間を短くして機会損失を少なくするために、離れた場所でも試験機の運転状況が把握できるような仕組みを構築することを目的とする。

なお、本研究で構築するネットワーク構成については、依頼試験の試験条件等機密情報に関与するデータも取り扱うため、インターネット（外部）に接続せず、センターLAN のみの閉域網で構成するものとする。

2 研究内容

2.1 研究課題

センター内の試験機は新旧様々なメーカーによるものがあり、複数の機器の運転状況を管理しようとしても、機器ごとに表示が異なるため取り扱いが難しい問題がある。また、機器によっては知りたいデータを取得する機能がない場合やあったとしても使い方が難しいものもあり、情報そのものが取得できない問題もある。さらに、試験室にある試験機の異常を別室（職員室等）で確認する手段が確立されていない。以上の問題点を踏まえ、センターの様々な試験機に対応する対策を研究課題とし、各問題点に対してまとめたものを表1に示す。

表1 研究課題

問題	原因	対策 (研究課題)
機器ごとに表示が異なり使いづらい	各々異なるメーカーが作っているため	共通画面 (WEB画面) で可視化
知りたい情報が見られない	機器にデータ取得する機能がない あっても使い方が難しい	市販センサを使用したデータを取得するプログラムを作成
職員室から異常がすぐわからない	離れたところですぐにわかる手段がない	表示灯による警報発信により異常を早急に周知

2.2 状況把握試験機と取得データ

本研究で運転状況の把握を行う試験機は、オゾン環境試験機（サンシャインカーボンアーク灯試験機）とした。当試験機の運転状況を可視化することによって他の試験機にも同様の手段を展開できる事例につながる事が期待できる。

なお、当試験機は材料の耐候性を評価する環境試験機であり、試験条件や運転状況により試験機の出力電流や槽内の温度が変化するものである。今回はこの試験機の出力電流及び槽内温度、また試験室内温湿度を各測定用センサで取得することで、当試験機の運転状況を把握する。

2.3 機器構成

本研究で使用する機器の構成図を図1に示す。当試験機の温湿度等を測定器に接続したセンサにより取得し、センターLANを介してデータ収集サーバに各測定データを送信する。データ収集サーバには、WEBサーバとしての機能を持たせることで、LANに接続した端末のWEB画面より各測定データをグラフ表示するようにする。また、LANにネットワーク表示灯を設置し、測定値の任意の閾値を超過すると自動で警報を発報するシステムを構築する。

なお、今回は多数機器への汎用性を考慮し設置費用を抑えるため、測定器はシングルボードコンピュータのRaspberry Pi、データ収集サーバはオープンソース総合監視ツールであるZABBIXを使用した。

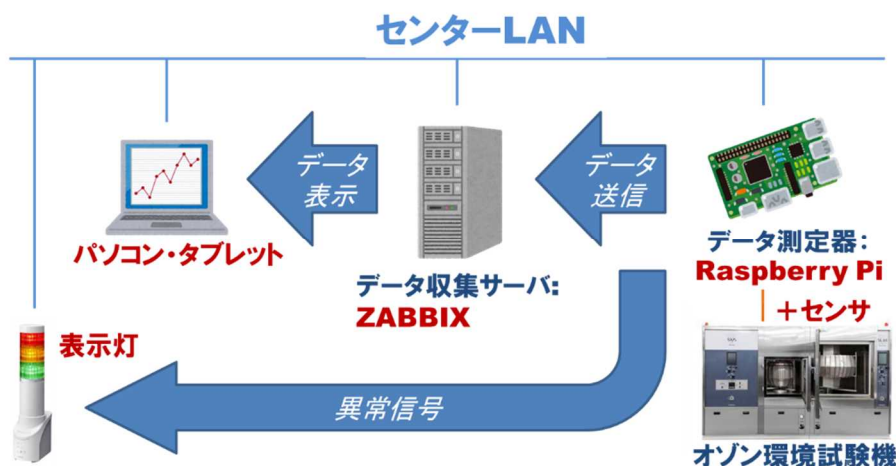


図1 機器構成

2. 4 使用機器

今回使用した主な機器について表 2 に示す。各センサ、モジュールはどれも安価で一般的に使用実績が多いものを選定した。Raspberry Pi と周辺機器の写真を図 2 に示す。

表 2 使用機器一覧

機能	使用機器
電流測定	CT センサ (クランプ) 100A AKW4802B
	A/D コンバータモジュール ADS1015
槽内温度測定	K 型熱電対モジュール MAX31855
室内温湿度測定	温湿度センサモジュール BME280
データと信号送信	Raspberry Pi 3 Model B+ OS:Raspbian Stretch
データ収集と表示	デスクトップパソコン OS:CentOS ソフト:ZABBIX
警報発報	ネットワーク表示灯 NHS-3FB1

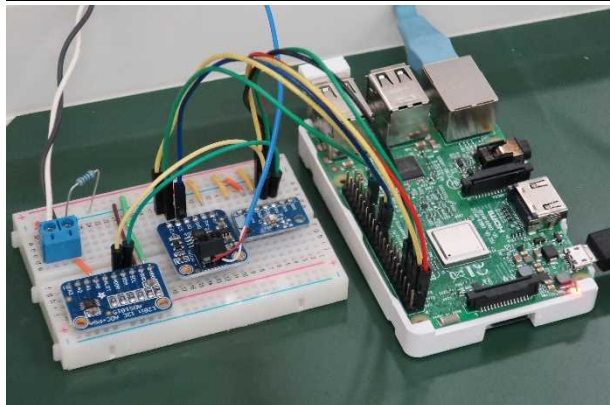


図 2 Raspberry Pi と周辺機器

2. 5 データ測定器の構築

2. 5. 1 Raspberry Pi の設定

本取り組みは長期的で複数のデータを測定するため、データ測定器は安定性の高い有線 LAN 接続と複数センサ接続が可能なインターフェースを装備する Raspberry Pi を使用した。以下の手順のとおり、データ測定器としてシステム構築した。

- 1) microSD に OS (Raspbian Stretch) を書き込み、Raspberry Pi を起動。
- 2) 固定 IP アドレスを設定。
- 3) SSH を有効化。
- 4) I2C、SPI を有効化。

2. 5. 2 電流測定の実装

電流測定で用いる CT センサの測定信号はアナログ信号であり、Raspberry Pi で測定するには A/D 変換が必要になるため、A/D 変換コンバータである ADS1015 モジュールを使用する。Raspberry Pi と ADS1015 モジュールは、表 3 に示すとおり Raspberry Pi 上の GPIO ピンと当モジュールピンを接続した。なお、当モジュールは I2C デバイスとして使用する。

表 3 Raspberry Pi と ADS1015 の接続

Raspberry Pi GPIO	ADS1015
01 3.3V	VDD
03 GPIO 2	SDA
05 GPIO 3	SCL
09 GND	GND
	ADDR

CT センサのクランプ部は図 3 に示すとおり、ランプ点灯回路のケーブルに取り付けた。

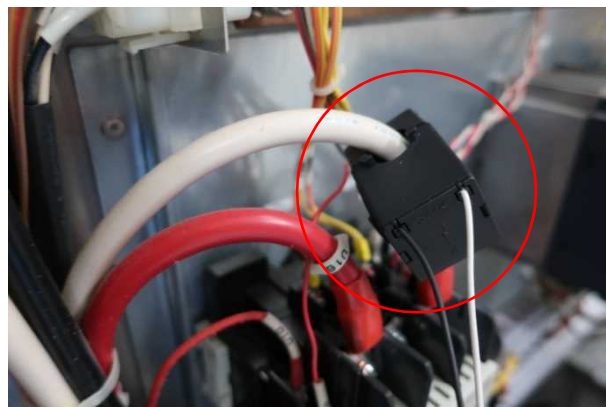


図 3 CT センサの取り付け

以上の配線後、ADS1015 のライブラリを使用し、Python による電流測定プログラムを作成した。

2. 5. 3 槽内温度測定の実装

槽内温度測定で用いる熱電対の測定信号はアナログ信号のため、K 型熱電対モジュール MAX31855 を使用する。Raspberry Pi と MAX31855 モジュールは、表 4 に示すとおり Raspberry Pi 上の GPIO ピンと当モジュールピンを接続した。なお、当モジュールは SPI デバイスとして使用する。

表 4 Raspberry Pi と MAX31855 の接続

Raspberry Pi GPIO	MAX31855
01 3.3V	Vin
09 GND	GND
21 GPIO 9	DO
23 GPIO11	CLK
29 GPIO 5	CS

熱電対の温度測定部は図 4 に示すとおり、槽内にくくりつける形で設置した。



図 4 熱電対の設置

以上の配線後、MAX31855 のライブラリを使用し、Python による槽内温度測定プログラムを作成した。

2. 5. 4 室内温湿度測定の実装

Raspberry Pi と温湿度測定で使用する BME280 モジュールは、表 5 に示すとおり Raspberry Pi 上の GPIO ピンと当モジュールピンを接続した。なお、当モジュールは I2C デバイスとして使用する。

表 5 Raspberry Pi と BME280 の接続

Raspberry Pi GPIO	BME280
01 3.3V	Vio, CSB
03 GPIO 2	SDI
05 GPIO 3	SCK
09 GND	GND, SDO

また、BME280 のセンサ部が室温測定に適した位置になるように配置した。

以上の配線後、BME280 のライブラリを使用し、Python による室内温湿度測定プログラムを作成した。

2. 6 データ収集サーバの構築

2. 6. 1 ZABBIX サーバの構築

データ収集サーバは、耐候性試験室に設置したデスクトップパソコンにサーバとしての環境を構築した。今回使用した ZABBIX は、データベース、WEB サーバなどの機能が含まれ、当サーバにデータを送るだけでグラフの表示や設定等が容易にできる特徴をもつ。ZABBIX で構築した環境を表 6 に示す。

表 6 ZABBIX の環境

機能	パッケージ
Web サーバ	Apache
開発言語	PHP
データベース	PostgreSQL
監視ソフト	Zabbix Server
監視用モジュール	Zabbix Agent2

2. 6. 2 ZABBIX でのグラフ表示

Raspberry Pi から送信されたデータを ZABBIX 上で表示するには、ZABBIX 上で Zabbix Agent としてホスト登録し、各測定データはホストのアイテムとして登録する必要がある。各センサのデータを ZABBIX のグラフ表示は以下の手順のとおり実施した。

- 1) Raspberry Pi に Zabbix Agent (測定データを ZABBIX で監視できるようにするためのソフト) をインストール及び設定。
- 2) センター LAN に接続した端末のブラウザから ZABBIX サーバの IP アドレスより管理画面を開く。
- 3) Raspberry Pi のホストを作成。
- 4) 各測定データのアイテムを登録。
- 5) 各グラフ表示を設定 (レンジ等)。(図 5 参照)



図 5 ZABBIX のグラフ設定画面

- 6) Raspberry Pi 上で zabbix sender を使用した各測定データを送信す

るシェルスクリプトを作成し、Python プログラムで得られた各データを zabbix sender の値に入れて送信する。

7) ZABBIX のグラフ画面で測定結果のグラフが表示されることを確認。

2. 7 ネットワーク表示灯の連携

データ測定器である Raspberry Pi とネットワーク表示灯を連携させ、測定値がある閾値を超えた際に表示灯を点灯及び鳴動（警報発報）させるシステムを構築する。今回は、表示灯の発報プログラムであるシェルスクリプトを Raspberry Pi から LAN 経由で実行させる rsh（リモートシェル）プログラムを用いて動作させた。システム構築は次の手順のとおり実施した。

1) 表示灯をセンターLANに接続し、IPアドレスよりWEBブラウザから管理画面にログインし、「送信元アドレス無効時ログイン名」を設定（シェルスクリプトプログラムに必要）。

- 2) Raspberry Pi に rsh-client パッケージをインストール。
- 3) Raspberry Pi に bc コマンド（測定値と閾値との比較に必要）をインストール。
- 4) 測定値が閾値を越えた時に表示灯を制御するスクリプトを作成し、実行。

3 結果および考察

3. 1 試験機運転状況の可視化

オゾン環境試験機（サンシャインカーボンアーク灯試験機）の運転時における、試験機の実出力電流、槽内温度、室内温湿度について4日間実施した。その測定結果を表すZABBIXのグラフを図6に示す。当グラフのCTセンサと熱電対の横軸は時間軸を表し、互いに一致している。なお、当試験の試験条件は以下のとおりである。

試験条件（空調機の室温設定 23℃）：
 サンシャインカーボンアーク灯試験機
 BPT：63℃
 水スプレー：60分中12分

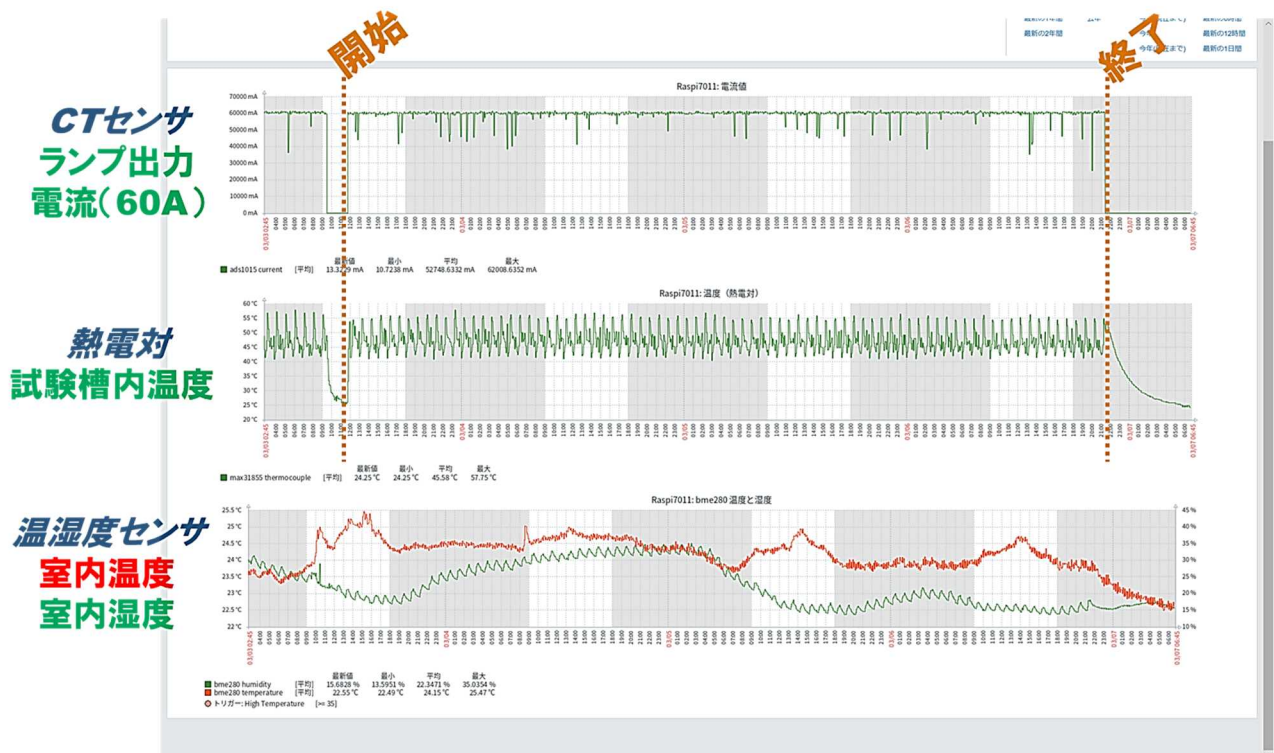


図6 オゾン環境試験機の運転状況

まず、CT センサと熱電対の挙動のうち、互いに同時に立ち下がっている箇所がわかる。この共通の立ち上がりから立下りの間は約 80 時間であり、これはカーボンの連続可能運転と一致する時間のため、当試験機の連続運転した時間がこのグラフにより可視化されている。

また、熱電対のグラフは運転時間帯において温度がある範囲で上下を繰り返している結果となっているが、これは周期的な水スプレーによる温度低下のタイミングと一致しているため、正常な運転をしていることが確認できる。

温湿度については、昼間の時間帯（非網掛け部）に一時的に上昇する挙動がわかることから、室内は常に空調機で温度管理されていても、日中の温度上昇に影響がでていることがわかる。

3. 2 表示灯による異常周知

室内温度が設定した閾値を超えた時、または出力電流がある値より小さくなった時に表示灯を点灯させるプログラムを作成したところ、いずれかの条件になると図 7 のように点灯とブザーが鳴動した。このように、ネットワークを介して表示灯で異常を通知することで、試験機から離れている場所でも、表示灯が視界に入る場所にあれば、試験機の異常状態をすぐに知ることができる。



図 7 ネットワーク表示灯の点灯

3. 3 考察

今回設置した CT センサや熱電対は非常に簡単に、メーカー保証範囲に影響もなく機器側へ設置ができることから、データ測定器の Raspberry Pi を Zabbix Agent としてセンター LAN に複数接続することで、

ZABBIX で監視できる対象機器を容易に増やすことが可能になる。また、ネットワーク表示灯についても同様のことがいえる。

しかし、今後このような監視対象が増加すると ZABBIX サーバの負荷が増え、ホストの管理やデータ容量増の対応、またセンター LAN 全体の負荷増の対応が今後の課題となる。また、ネットワーク表示灯は、今回は Raspberry Pi との連携で直接制御を行ったが、複数台を管理する場合、ZABBIX サーバで一括管理することが求められるため、この手法を確立することも今後の課題である。

4 まとめ

本研究により、センター LAN の環境からどこでも試験機の運転状況を把握することができ、異常を早急に周知可能にすることが実証できた。

今回の成果を、オゾン環境試験機以外の機器に展開、適用することで、センター全体の業務効率化や依頼試験等収入増が見込める。そのためには、ネットワークや ZABBIX サーバの管理に関する課題について、今後も実証を重ね、取り組むことが求められる。また、表示灯の接続が増えた場合に対しては、設置箇所や運用方法の検討が必要となる。

センター外に関しても、今回の研究実施内容は必要な機器を揃えることができれば容易に構築可能なシステムのため、様々な分野の企業への IoT 支援ツールとして展開することが期待できる。

参考文献

- 1) 「Zabbix 4.4 インストール手順(Cent OS8 / Apache2.4 / PHP7.2 / MySQL8.0)」, <<https://densan-hoshigumi.com/server/zabbix-4-4-apache-mysql-install>>
- 2) 「Raspberry pi+zabbix + BME280 で温度湿度気圧のロギングをする」, <<https://hotspring.hatenablog.jp/entry/2018/06/14/184408>>