北東北地域企業等へのIoT導入強化に関する研究

北東北つながる工場システム

構築マニュアル

令和５年３月

北東北技術連携推進会議IoT技術分野研究会

目次

1. システムの全体像、設計思想

本事業は、国立研究開発法人産業技術総合研究所（以下「産総研」という。）と公設試験研究機関（以下「公設試」という。）との間に、共同研究に利用可能な試験環境（テストベッド）として、「つながる工場テストベッド」を構築し、地域企業の IoT 活用促進と、IoT に関する地域課題の解決を図るための方法論を検討するものである。

北東北の公設試ではこれまで、地方独立行政法人青森県産業技術センター（以下「青森産技」という。）、秋田県産業技術センター（以下「秋田産技」という。）、及び地方独立行政法人岩手県工業技術センター（以下「岩手工技」という。）が参画する「北東北公設試 技術連携推進会議IoT技術分野研究会」（以下「北東北研究会」という。）による情報交換を行っており、本事業における実施体制の基盤となっている。本事業では、北東北の取組として、複数拠点の機器稼働状況などの情報を一元化し、仮想的に同一拠点のようにふるまう「仮想広域工場」のデモンストレーション環境を構築することで、拠点間の工程管理や受発注の融通にIoTが活用できる可能性を示し、関連するIoT技術を活用した地域企業での課題解決を目指す。図 1-1に仮想広域工場の概念図を、図 1-2にシステム全体構成を示す。

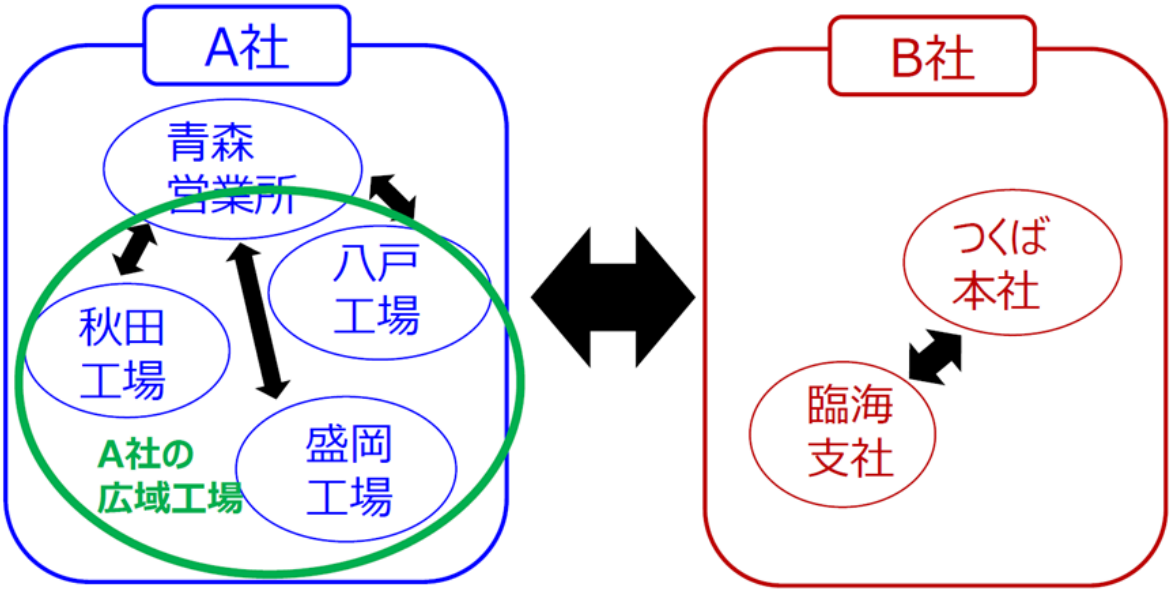


図 1-1　仮想広域工場の概念図

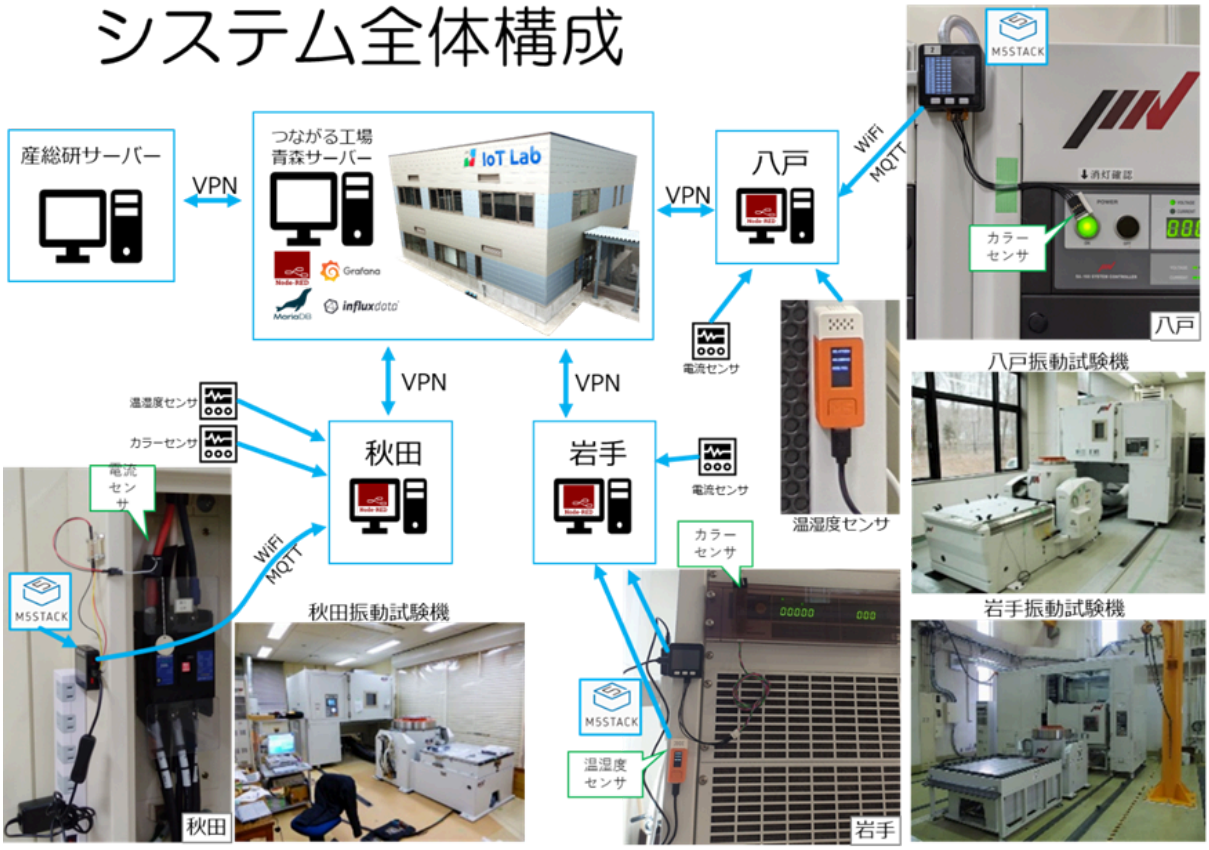


図 1-2　システム全体構成

2.　構築手順

今回構築したシステムについて、エッジデバイスのプログラミングからデータサーバに各種ソフトウェアをインストールしデータ収集及び可視化するまでの実手順を説明する。

なお、RaspberryPiとM5Stackでセンサ情報の収集を行いたい場合は付録9簡易システム構築例を参照されたし。各種ソフトウェアはWindowsPCでもインストール可能であるため、WindowsPCとM5Stackでセンサ情報の収集を行うことも可能である。

また、セキュリティ等の対策のため、インターネットを使用しないでローカルネットワーク内だけでデータ収集を行うことも可能である。

2.1.　ネットワーク

2.1.1.　VPN

- 開発に使用するWindowsPCを用意する。
- インターネット接続を用意する。インターネット経由でIoTデータの受信を行う場合にはルーターでアドレス変換やポート変換が可能な環境を用意する。
- 拠点間を接続するVPNルーターCisco Meraki MX64は産総研が設定・設置を行ったため、配布されたVPN接続用IDとパスワードを使用してVPN接続を設定する。

- 「リモートネットワークでデフォルトゲートウェイを使う」とインターネット接続が遅くなるため、チェックを外す設定を行い、VPN経由でサーバへの接続が可能なよう、ルーティングを設定する。
- M5Stack用にWiFiルーターを用意し、ネットワークに接続する
- 各拠点が発した、インターネット側から送信されてくるMQTTデータをLinuxサーバで受信するため、ポート変換設定を行う。

2.1.2. SoftEther

- WindowsPCにSoftetherをインストールし、softetherのDNSの設定を行う。

2.2. データサーバ

2.2.1. Linuxサーバ

- サーバ用に使用するPCを用意して、産総研が作成した「サーバー設定手順書」に従い、OSなどをインストールする。
- Node-red、Grafana、Influxdb、をインストールする。

2.3. エッジデバイス

2.3.1. M5Stack

- M5StackやM5Stick、電流センサ、温湿度センサ等を購入する。

  調達物品リスト - Google スプレッドシート

- Arduino1.8 開発環境を開発用PCにインストール
- Arduino開発環境にM5Stack,M5Stick用ボードマネージャー、ライブラリをインストール
- UIFLOW、M5Bunerを開発用PCをインストール
- プログラムのIoTデータ送信先サーバやWiFi設定を修正し、プログラムを書き込む

2.3.2. GM3環境センサ

- GM3環境センサ、受信機を購入し使用するネットワークに合わせてセットアップを行う。セットアップ方法はMZプラットフォームの「MZ Platform IoT Toolkit (スマート製造ツールキット/MZ-EX/IoT化用コンテンツ集)」内の「MZIoTtoolkit説明資料(市販機器).pdf」を参考
- Node-Redでデータ受信の確認を行う。

2.4. データフロー・データベース

2.4.1. Node-RED

- Node-redに追加プラグインをインストール
- Node-Redのフロー(プログラム)のインポートを行う
- ネットワークに合わせてフローの接続先IPアドレス等を修正する

2.4.2. InfluxDB

- インストール後、influxコマンドで対話モードに入り、
- create database aomori を実行

2.4.3. MariaDB

- MZプラットフォームの「MZ Platform IoT Toolkit (スマート製造ツールキット/MZ-EX/IoT化用コンテンツ集)」内の「MZ講習試料IoTサーバ編.pdf」を参考にテーブル等の初期設定を行う。

2.5. 可視化システム

2.5.1. Grafana

- defaultデータベースはinfluxdbをアクセスするよう設定を行う
- Dashboard（可視化画面）の作成を行う

2.5.2. MZプラットフォーム

- MZプラットフォームユーザー会にユーザー登録を行い、開発環境をセットアップする
- MZプラットフォームマニュアル等を参照しプログラムを開発する

3. ネットワーク

北東北研究会の青森産技、秋田産技、岩手工技と、産総研の、独立した機関の間でデータをリアルタイムで共有するためには、情報セキュリティを確保可能なネットワークが必要になる。また、青森産技では青森市の工業総合研究所（青森市）（以下「工総研」という。）と八戸市の八戸工業研究所（以下「八工研」という。）に、測定対象とする装置と研究員が分かれている。そのため、青森市、八戸市、秋田市、盛岡市、つくば市の拠点間を接続するためのVPNネットワークシステムを産総研側で構築した。

ネットワークの構成を図 3-1に示す。青森市にVPNルーター(Meraki MX64)とサーバを設置し、各拠点のPCからVPN接続することで安全なネットワークを構築した。

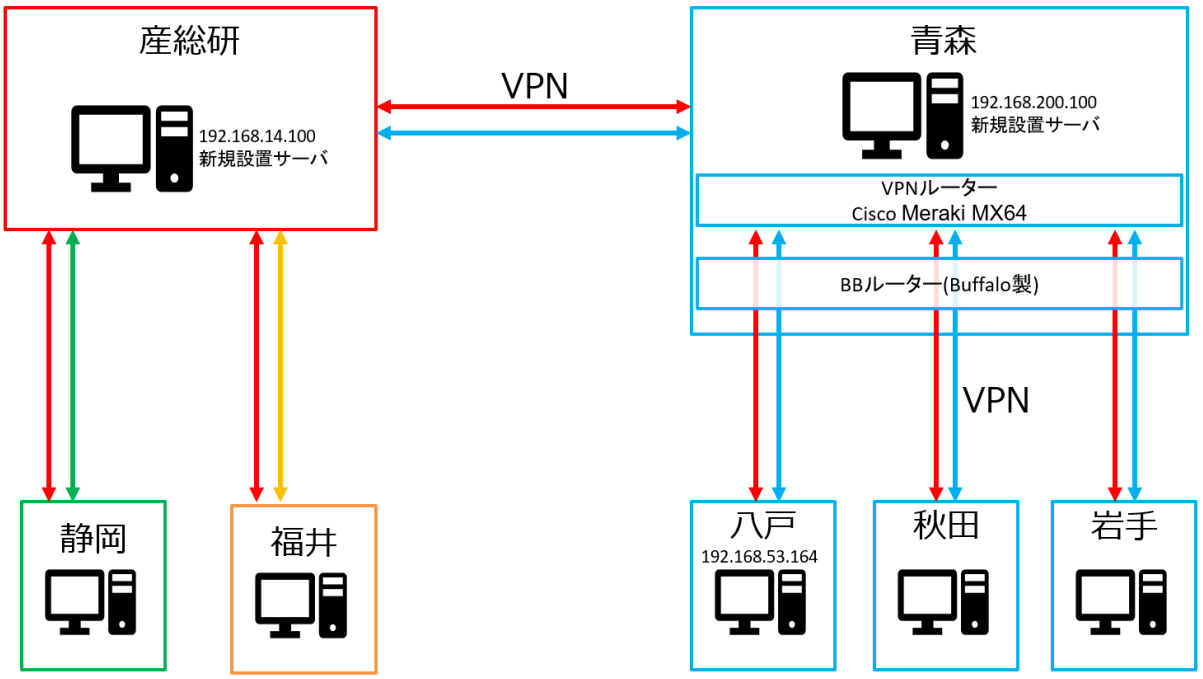


図 3-1　VPNネットワークシステム構成図（データ表示時）

3.1. VPN接続

「VPN」とは「Virtual Private Network」の略称で、「仮想プライベートネットワーク」や「仮想専用通信網」と訳される。VPNでインターネットに接続することをVPN接続といい、コストを要する物理的な専用回線を用いずに、共用回線を活用しながら仮想的に専用回線のように扱うことで、コストを抑えつつ安全性を確保することが可能である。

本事業では、独立した複数の機関による共同研究の中でリアルタイムデータの共有を行うため、情報セキュリティーを確保する必要があったことから、VPN接続としたものである。

VPNルーターは既存のインターネット回線に接続した。青森の工総研には業務用のネットワークとは別に試験研究･実験用のインターネット回線としてフレッツ光を契約しているためこれを使用した。

ブロードバンドルーター(Buffalo社,BHR-4GRV製)にはVPN用に500,4500番のポート変換設定を行った。ポート変換設定は各メーカーによりアドレス変換、ポート開放など設定画面での呼び方が異なっているため各自が使用しているルーターの取扱説明書等を確認してほしい。参考として青森のエ総研で使用しているBuffalo社製のブロードバンドルーターのポート変換設定画面を図3.1-1に示す。VPNルーターはNAT（Network Address Translation：ネットワークアドレス変換）の背後に設置したVPNサーバのため、WindowsPCからこのVPNサーバに接続するためにはレジストリの変更が必要になる。この手順については下記サイトを参考にして、「AssumeUDPEncapsulationContextOnSendRule 」をDWORD (32 ビット) の値「2」に設定すると良い。

NAT-T デバイスの背後に L2TP/IPsec サーバを構成する - Windows Server | Microsoft Learn

WIndows10でのVPN設定方法は下記サイトを参考にすると良い。

クライアントVPNのOS別設定 - Cisco Meraki

Windows10において、VPNの常時接続を実現するためには下記リンクの手法があるようで試したが、うまくいかない場合が多々あった。

Winodws10でVPN（L2TP/IPsec）の自動接続設定 - 株式会社ネディア ｜ネットワークの明日を創る｜群馬

とても簡単にVPN接続する方法

[VPN] Windows起動時に自動接続する - プロプログラマ -Flex,Air,C#,Oracle,HTML5+JS-

タスクスケジューラーでは「次のネットワーク接続が使用可能な場合のみタスクを開始する(Y)」等のオプションがあり、試行錯誤を繰り返したがうまくいかなかった。

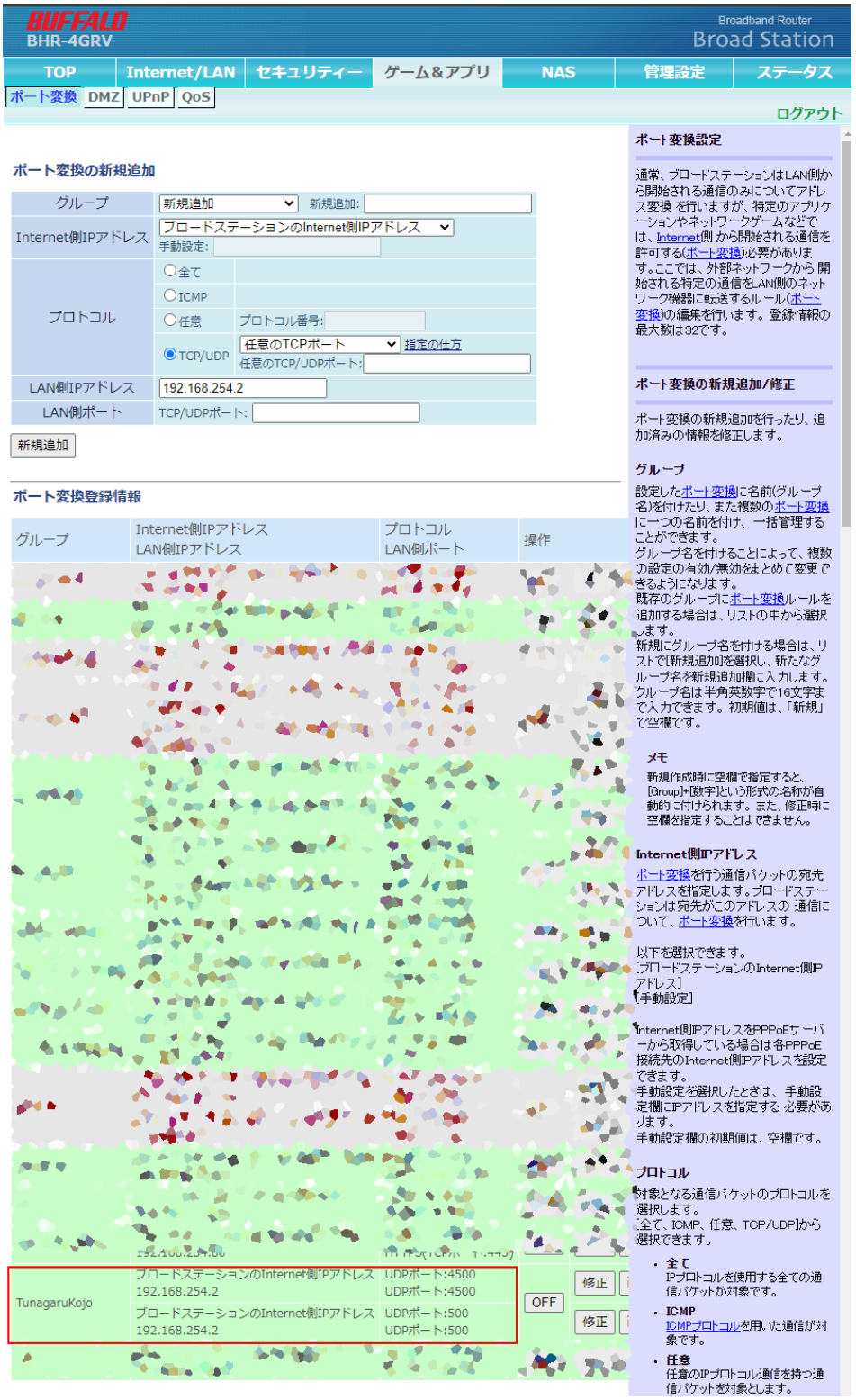


図3.1-1

VPNルーター（Cisco Meraki MX64）へのVPNの常時接続が各県でも実現できなかったため、各種センサデバイスのデータを青森に設置したDBサーバにVPN経由で収集することができなかった。そのため、IoTデータ収集にはインターネットを経由してデータ保存することとした（図 3.1-2）。青森のエ総研では研究用にインターネット側からMQTT接続を受け付けるようネットワーク及び中継サーバが構成されており、本事業用に受信したデータをDBサーバに書き込むよう設定した。これにより、

MQTT送信側であるIoTデバイスはインターネット上のサーバに向けてデータ送信するようなプログラムで済む。

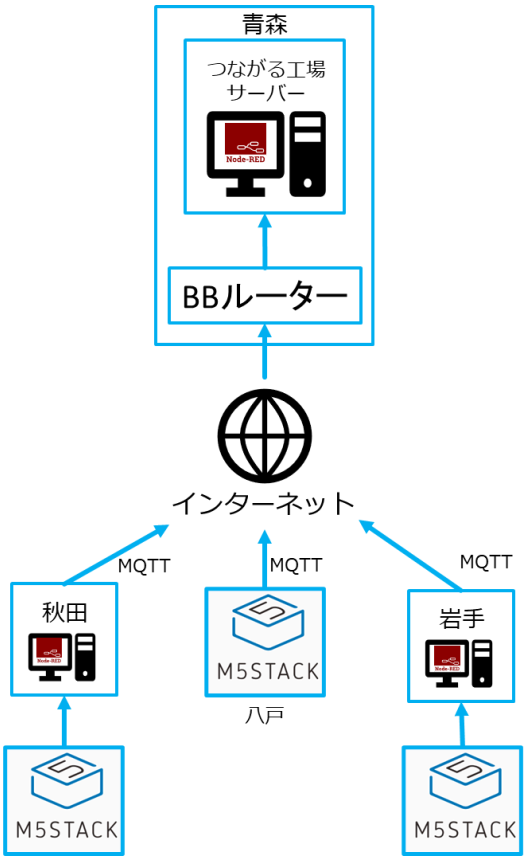


図 3.1-2 データ収集時のデータの流れ

データ参照時にはVPNを接続してMZアプリやブラウザからデータ表示を行う。しかし、VPN接続中にインターネット検索等を使用すると、表示が遅く感じられる時がある。これはVPN経由でインターネット接続を行うためである。Windwosでは「リモートネットワークでデフォルトゲートウェイを使う」オプションをOFFにすることで、回避することができる。

本事業で北東北のVPN接続端末は192.168.53.xxのIPアドレスが割り当てられる。青森のサーバは192.168.200.10が設定されており、産総研のサーバには192.168.14.100が設定されている。上記のリモートネットワークでデフォルトゲートウェイを使う」オプションをOFFにした場合、PC同士のサブネットが異なるため、ルーティング情報を正しく設定しなければアクセスすることができない。

Wndowsではコマンドプロンプトから、「route print」を実行することで、現在のルーティング情報を表示することができる。

本事業では青森サーバ(192.168.200.10)や臨海サーバ(192.168.14.100)へのアクセスにはVPN接続のネットワークアダプタ(192.168.53.164)からアクセスしたいため、下記コマンドにてルーティング情報を追加する。ルーティング情報を追加するには管理者権限が必要であるため、管理者権限でコマンドプロンプトを開き(図 3.1-3)、下記コマンドを実行する。

図3.1-3 ルーティング追加のため管理者権限でコマンドプロンプトを起動

route add 192.168.200.0 mask 255.255.255.0 192.168.53.1

route add 192.168.14.0 mask 255.255.255.0 192.168.53.1

なお、このコマンドではPCの再起動時に登録したルーティング情報が消えてしまうので、恒久的に設定を行いたい場合は、下記のように「-p」オプションを使用すると良い。

route -p add 192.168.200.0 mask 255.255.255.0 192.168.53.1

route -p add 192.168.14.0 mask 255.255.255.0 192.168.53.1

routeコマンドの詳しい使用方法ついては下記リンクやインターネット検索を参照されたし。

「route」コマンドでWindowsのルーティングテーブルを操作する：Tech TIPS - @IT

ネットワーク接続を行い、サーバに接続可能かどうか調べる際には「tracert」コマンドや「ping」コマンドを使い、相手からの応答が返ってくるか調べると良い。tracertコマンドについては下記リンクやインターネット検索を参照されたし。

tracertコマンドでネットワークの経路を調査する：Tech TIPS - @IT

3.2. SoftEther

VPN接続やIoTデバイスからのデータ受信をインターネット上から受け付けるには、固定IPアドレスやDNS名が必要となる。今回使用している青森の工総研の研究用ネットワークは一般家庭でも契約するようなフレッツ光であり、固定IPではないため、DNS名が必須であった。

過去には無料でDNSを登録し維持可能なサービスもあったが、システム構築時には見つからなかったため、実験用途で稼働させていたSoftetherにおいて、DNS名をIoTの用途で使えないか検証を行った。その結果、M5StackやESPマイコンからSoftetherのDNS名に対してデータ送信を行い、ブロードバンドルーターでポート変換後、Node-Redサーバでデータ受信が可能であった。このことから本事業ではDNSによる名前解決のためだけにSoftetherを使用している。

Softether運用中に一度、SoftetherVPNサーバをインストールしたPCのSSDが故障し、データを全て失った事があった。別PCで同一のDNS名を登録したい場合、「鍵(Key値)」が必要とのことであったが、SSD故障により「鍵」の復旧が困難であるため、以後別のDNS名を新たに設定し、全てのIoTデ

バイスのプログラム修正を行った。softehter.netのDNS名を他サーバに移行する場合は下記情報を参考にされたし。また、「鍵」情報のバックアップを推奨する。

softether.net のダイナミック DNS のホスト名を他のサーバに移行する方法

SoftEtherの導入方法については、以下の「SoftEther VPN プロジェクト」のサイトから詳細な情報が得られるので参照されたい。

SoftEther VPN プロジェクト

4. サーバ

装置の状態をセンシングした結果について、後に蓄積したデータを分析することで製造現場における生産性向上等に活用することを視野に入れ、リアルタイムモニタのほか、時系列データベース内にデータを蓄積できるよう、3県でセンシングしたデータを格納するデータサーバを次の手順により構築する。

4.1. Linuxサーバ（産総研より貸与）

データサーバはCentOS8で構築した。具体的な構築方法は、産総研が作成した「サーバー設定手順書」を参照した。

また、以下の記事が参考となるので併せてリンクを記載する。なお、LinuxOSに限定するものではない。

・産総研作成「サーバー設定手順書」

・サーバ構築の基本 CentOS Linux 8のインストール後に設定する12の項目 - レムシステム エンジニアブログ (rem-system.com)

北東北システム構築時には産総研でOSをセットアップしていただいたPCに、追加で今回使用するソフトウェアをインストールした。

- Node-Red　v1.2.5
- InfluxDB　v1.8.10
- MariaDB　v10.3.35
- Grafana　v7.3.1

4.1.1.　中継サーバ（仮想環境）

3.1 VPN接続に記載したとおり、北東北システムではVPNの常時接続において課題があったため、青森のDBサーバ（前記のLinuxサーバ）の手前に、インターネット側からMQTTを受信するための中継サーバを設け、青森のDBサーバに書き込む仕組みを構築した。

中継サーバはWindows10のPCの上に仮想マシンソフトウェアのVirtualBox6.1を用いてubuntu20.04.3LTSの仮想マシンを構築した。本仮想マシンはIoTの試験研究用に構築してあったもので所内の施錠システムなど他の用途でも使用している。仮想マシンの自動起動については、バッチファイル等で起動する手法がインターネット検索で得られるが、うまくいかなかったため、VBoxHeadlessTrayというソフトを用いて仮想マシンの自動起動・自動シャットダウンを行えるようにした。また、タスクトレイ常駐型であり、VirtualBoxのウィンドウを非表示状態で仮想マシンを実行できるため、WindowsPCを研修会で使う際に邪魔にならない。VBoxHeadlessTrayについては、下記リンクで概要を得ることができる。

PCシャットダウン時にVirtualBoxを自動でサスペンドしてくれるツール”VBoxHeadlessTray”

VBoxHeadlessTrayは下記リンクからダウンロード可能である。

VBoxHeadlessTray - Topten Software

中継サーバ（ubuntu）にも同様に今回使用するソフトウェアをインストールした。

- Node-Red　　v1.2.5
- InfluxDB　　v1.8.10
- MariaDB　　v10.3.35
- Grafana　　v7.3.1

4.1.2.　Node-Redのインストールと設定

- Node-Redのインストール方法は下記サイトを参照してほしい。

  Running Node-RED locally

- 初期設定

  初期設定は無くても使用可能であるが、誰でもフロー(プログラム)の編集が可能であるため、セキュリティ向上のためにはユーザーアカウントとパスワードの設定等を推奨する。

  Node-Redのセキュリティ対策は下記リンクを参照されたし。パスワードはハッシュ値生成して設定ファイルにコピペするため、マニュアルを良く読んだほうが良い。

  セキュリティ : Node-RED日本ユーザ会

- 起動、自動起動設定

ターミナルで、「node-red」 を実行する。

システム起動時に自動起動するには下記リンクを参考に設定が必要である。インストールスクリプトを使用した場合は自動起動の設定も行ってくれるようだが対応OSが書かれているので注意されたし。

ブート時にNode-REDを起動する

- 編集画面の表示

  WEBブラウザで http://localhost:1880 にアクセスする。

- 追加ノードのインストール

  パレットの管理－ノードを追加 から下記ノードを検索して追加する。

○ node-red-contrib-influxdb	*0.5.4*	*influxDB*用
○ node-red-node-mysql	*0.1.1*	*MariaDB*用
○ *node-red-contrib-aedes*	*0.5.1*	*MQTT*用

- 北東北システムのインポート

  参考として、本事業の中継サーバのNode-Redフローを添付する。下記リンクからダウンロード可能である。

  https://drive.google.com/file/d/1_Ngh3D_RiRVLla7VMx-Oj2Yp_VI23ojH/view?usp=share_link

4.1.3. Grafanaのインストールと設定

- Grafanaのインストール方法は下記サイトを参照してほしい。

  Install on Debian or Ubuntu | Grafana documentation

  To install the latest release: 配下のコマンドを実行
  Add this repository for stable releases: 配下のコマンドを実行
  After you add the repository: 配下のコマンド（OSS releaseまで）を実行
  To start the service and verify that the service has started: 配下のコマンドを実行
  Configure the Grafana server to start at boot: 配下のコマンドを実行

- ログインやダッシュボードの作成については以下のリンクを参照してほしい。

  Build your first dashboard | Grafana documentation

- ダッシュボード画面の表示

  WEBブラウザで　http://localhost:3000　にアクセスする。

- 参考に今回構築したダッシュボードデータを置く。（ダウンロード・インポート可能）

  https://drive.google.com/file/d/1-SJ__7Owl4C4-HrilkGJzWazJZn-pMv8/view?usp=share_link

4.2.　データベース

　データベースはMariaDBとinfluxDBを併用している。データ表示を行う際、「MZプラットフォーム」と「Grafana」を使用しており、MZプラットフォームでinfluxDBにアクセスすることが困難であったためMariaDBも併用することになった。influxDBは時系列データベースであり、ルールを設定することで古いデータを自動的にダウンサンプリングすることが可能である。

- InfluxDBのインストール方法は下記サイトを参照してほしい。

  GitHub - influxdata/influxdb: Scalable datastore for metrics, events, and real-time analytics

  上記サイトにも書かれているが、下記リンクからDL可能である。

  Downloads

  北東北では2020年度のシステム構築時にはversion 1.6.4を使用した。

- 起動、自動起動設定

  インストール時に自動起動設定も行ってくれる。

- 初期設定

  influxdbはサーバ機能を起動するコマンド「influxd」と、コマンドラインから対話的に操作可能な「influx」コマンドがある。

  初期設定として、データを保存するためのデータベースを1個作成する必要がある。データベースは目的や事業ごとに分けておくことも可能である。コマンドの使用方法は下記サイト等を参考にされたし。

  https://www.magata.net/memo/index.php?InfluxDB%C6%FE%CC%E7

  「influx」コマンドで対話モードに入り、「**create database**」コマンドでデータベースを作成する。参考にターミナルのコピペを下記に示す。

```
－－－－－－－－－－－－－－－－－－－－－－－－
[aitc@aitc ~]$ influx
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
> create database tsunagarudb
> show databases
name: databases
name
----
_internal
test
tsunagarudb
>
－－－－－－－－－－－－－－－－－－－－－－－－
```

ユーザー認証等の設定をしていないため誰でもデータの書き込み等が可能である。セキュリティに気をつける場合には、正しく設定されたし。

- MariaDBのセットアップ方法は産総研の「サーバ設定手順書.docx」p.37を参照のこと。

5. エッジデバイス

5.1. M5Stackを用いたセンサの試作

対象装置の電流や電源ランプの照度、周囲の温湿度などを取得してサーバに送信するエッジデバイスには、M5Stackシリーズを使用した。本製品は中国深圳に拠点を置く同名のスタートアップ企業が開発したマイコンモジュールであり、Wi-FiとBluetoothによる無線通信機能を備えたCPU（ESP32）をはじめ、液晶ディスプレイやボタン、microSDカードスロットなどがひとつのモジュールにまとまっている。シリーズによっては加速度など各種センサ類も搭載されているほか、オプション品を追加することで回路設計をすることなく簡単にセンサ類と接続することができる。シリーズには主流である54mm角の正方形のM5Stackのほか、細長な形状のM5StickC、さらに小型のM5Atomなどがある。今回は電流およびランプ照度の測定にM5Stack(Basic)を、温湿度の測定にM5StickCを用いた。

M5Stack　HP：https://m5stack.com/

5.1.1. 電流センサ

電流の測定には、M5Stackのアナログ入力ポートを使用し、ユーアールディ社の小型クランプ式交流電流センサ（CTL-16-CLS）を接続した。本センサは交流電流測定時に0Vを中心にプラスとマイナスの電圧が出力され、M5Stackではマイナス電圧のアナログ入力・測定ができないため、抵抗を用いた回路を設計した。

振動試験機の定格容量57kVA、今回の回路の測定電圧範囲が0~3.3Vで、電圧1.65V分オフセットし、1.0Vppの範囲で電流波形を測定したいため、電流センサの負荷抵抗は、

$$\frac{1[V]}{\frac{57000[VA]}{200[V]} \div 3000} = 10.5[\Omega]$$

$$\frac{1[V]}{\frac{57000[VA]}{200[V]} \div 3000} = 10.5[\Omega]$$

となり、10[Ω]を使用した。

M5Stackと電流センサの接続回路図を図 5-1に示す。接続時の写真を図 5-2に示す。

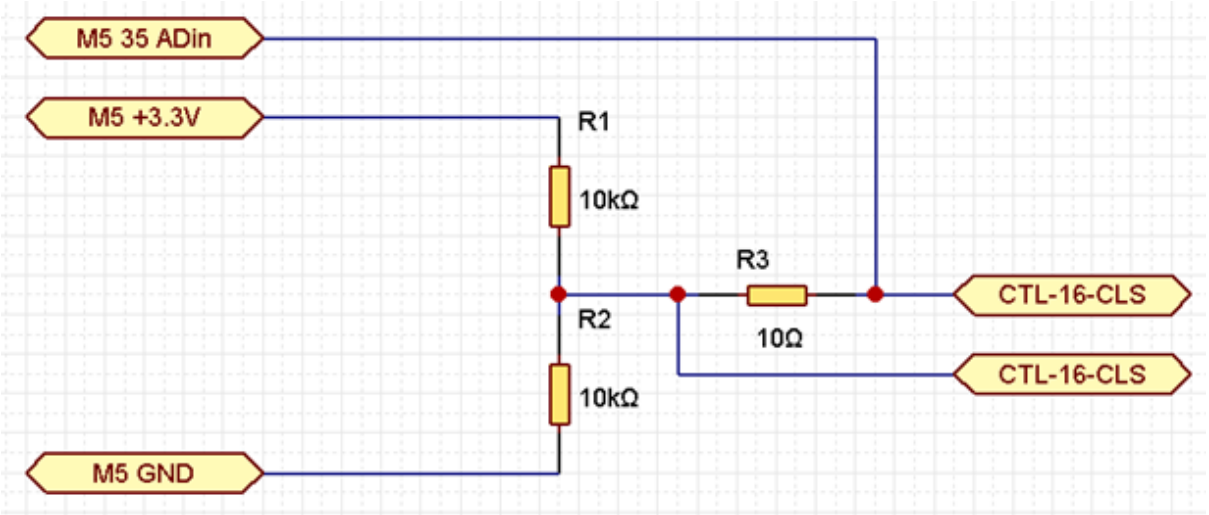


図 5-1　M5Stackと小型クランプ式交流電流センサの接続回路

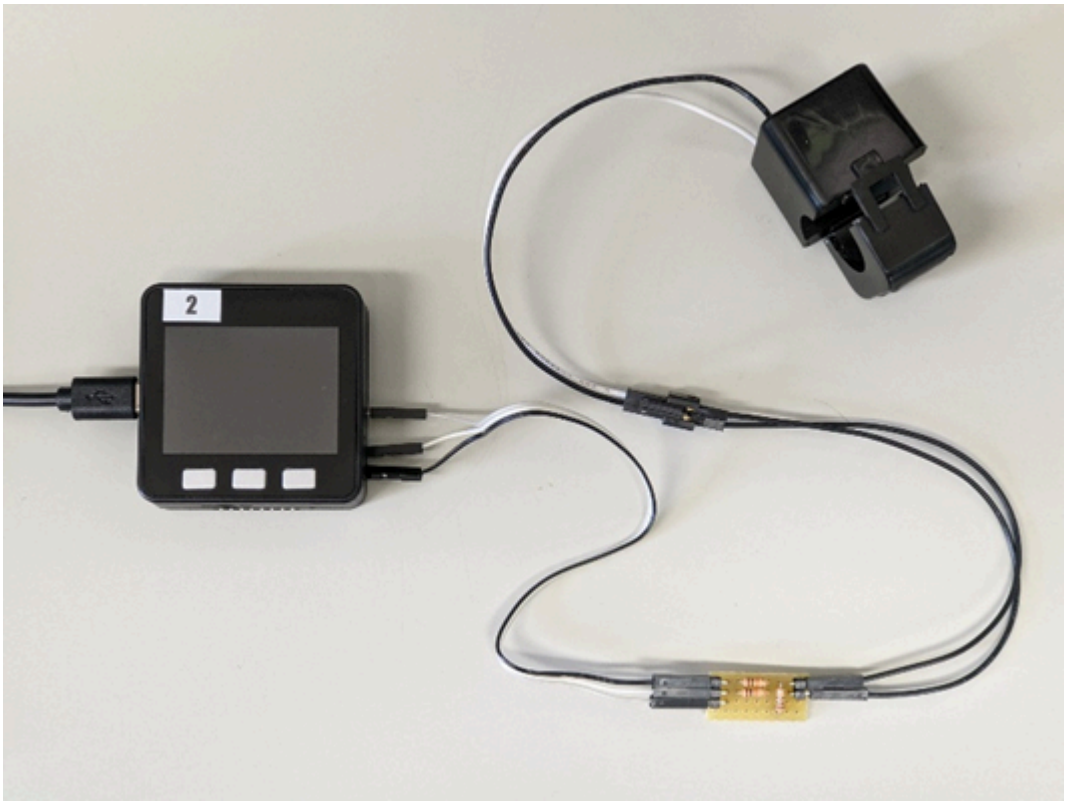


図 5-2　M5Stackと小型クランプ式交流電流センサの接続写真

M5Stackシリーズは、イタリアのArduino財団によって開発されたArduinoボード用のプログラム開発環境であるArduino-IDEによってプログラムを開発することができる。また、同社が提供するWEBブラウザ上の開発環境であるUI-FLOWでは、ブロックを組み合わせることによる視覚的なプログラム、いわゆるノーコード開発も可能である。それぞれの開発画面を図 5-3に示す。

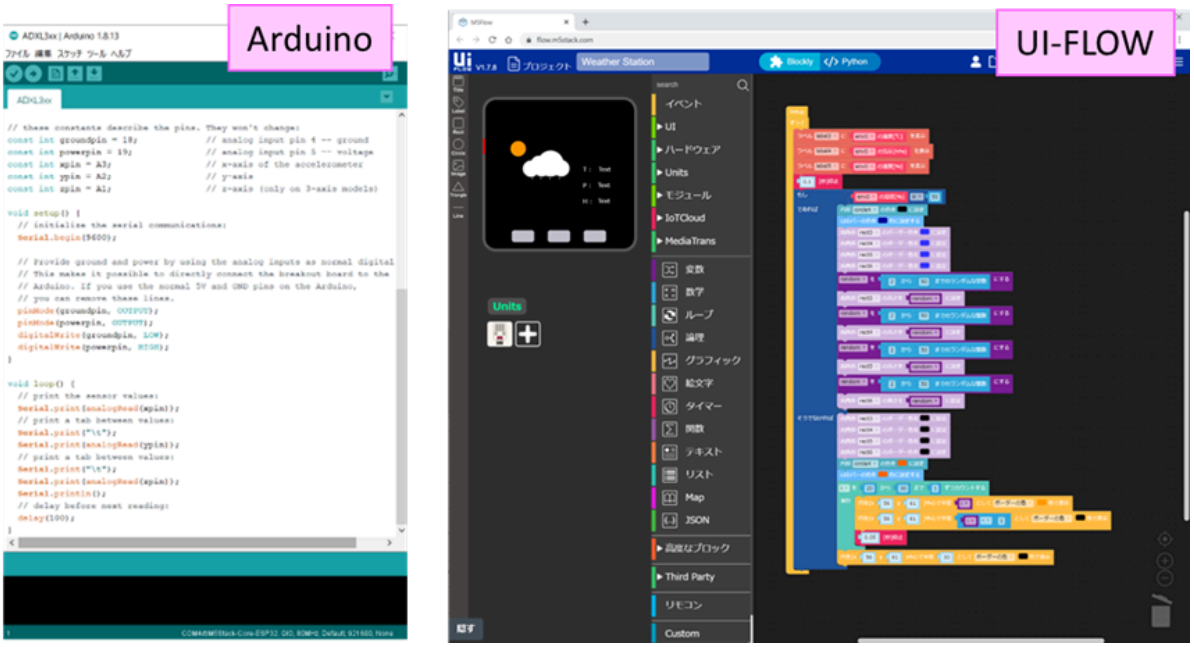


図 5-3　Arduino IDE と UI-FLOWの開発画面

電流測定のプログラムは以下の内容をArduino-IDEで作成した。

電流センサでは電流値を測定し、有効電力を計算した。電流波形に歪みがあると有効電力値がずれるため、有効電力の計算には電流波形に対して50Hzの基本波成分を計算後、電圧値を乗することで有効電力を算出した。測定結果はI2Cにより取得するカラーセンサの値とあわせ、MQTTプロトコルによるWiFi通信にてサーバへと送信した。詳細については「5.3 処理方式」を参照されたい。なお、今回使用したプログラムは、以下からダウンロード可能である。

https://drive.google.com/file/d/14OxXjvDKLb8wv_dstxJQt_Q0_xjajX8L/view?usp=share_link

5.1.2. カラーセンサ

振動試験機の運転状態を示すLED表示機の読み取りには浜松ホトニクス社のカラーセンサ（S11059－02DT）を使用した。M5Stackとの接続にはI2Cポートを使用した。接続についてはVpp(+3.3V),SCL,SDA,GNDをそれぞれ接続し、カラーセンサのデータシート記載のとおり、センサ側のVpp-GND間に0.1μFのコンデンサを追加した。接続した写真を図 5-4に示す。

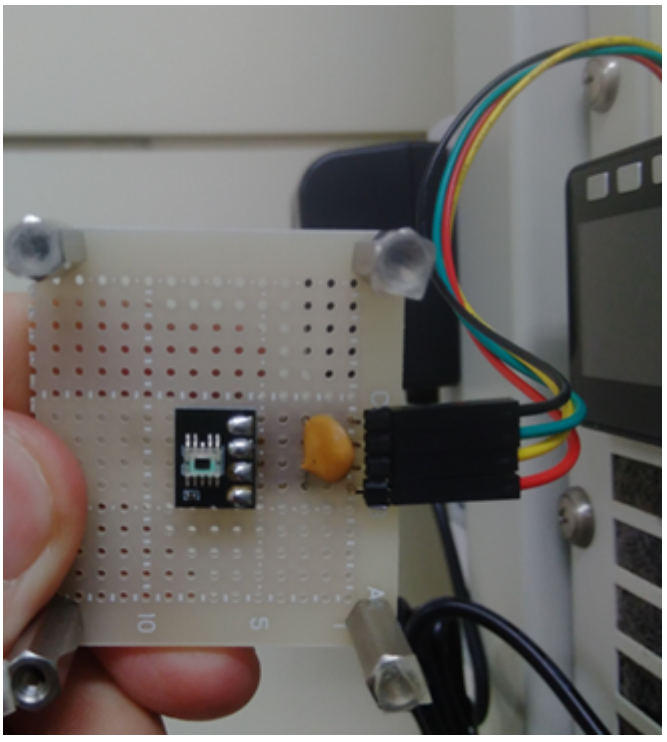


図 5-4 I2C通信のカラーセンサを接続した写真

I2C接続のセンサ値の取得からMQTT送信までを行うプログラムは、電流センサ同様にArduino-IDEで作成した。以下からダウンロード可能である。

https://drive.google.com/file/d/1V9S2ThVBVaX2sOh16nUpJXFEF0cBYAN5/view?usp=sharing

5.1.3.　温湿度センサ

温湿度の測定にはM5StickC用にオプションとして提供されているセンサデバイスENV II HATを用いた。上述の2種類のセンサとは異なり、市販されているものをそのまま使えるため、追加回路の自作は不要である。M5StickCにENV II HATを取り付けた状態を図 5-5に示す。ENV II HATについては下記リンクを参照されたい。

https://docs.m5stack.com/ja/hat/hat_envII

図 5-5　ENV II HATを取付けたM5StickC

温湿度測定のために作成したUI-FLOWのプログラムを図5-6に示す。内容は、大きく分けて「一度だけ処理される部分」と「繰り返し処理される部分」に分かれる。前者は初期設定のようなもので、WiFIやMQTT(5.3に記載)の設定などが行われ、後者ではセンサ値の取得や表示、サーバへの送信など、時々刻々と変化する数値についての処理が行われる。なお、使用したセンサデバイスENV II HATには気圧センサも搭載されているため、温湿度のほかに気圧も送信することとした。

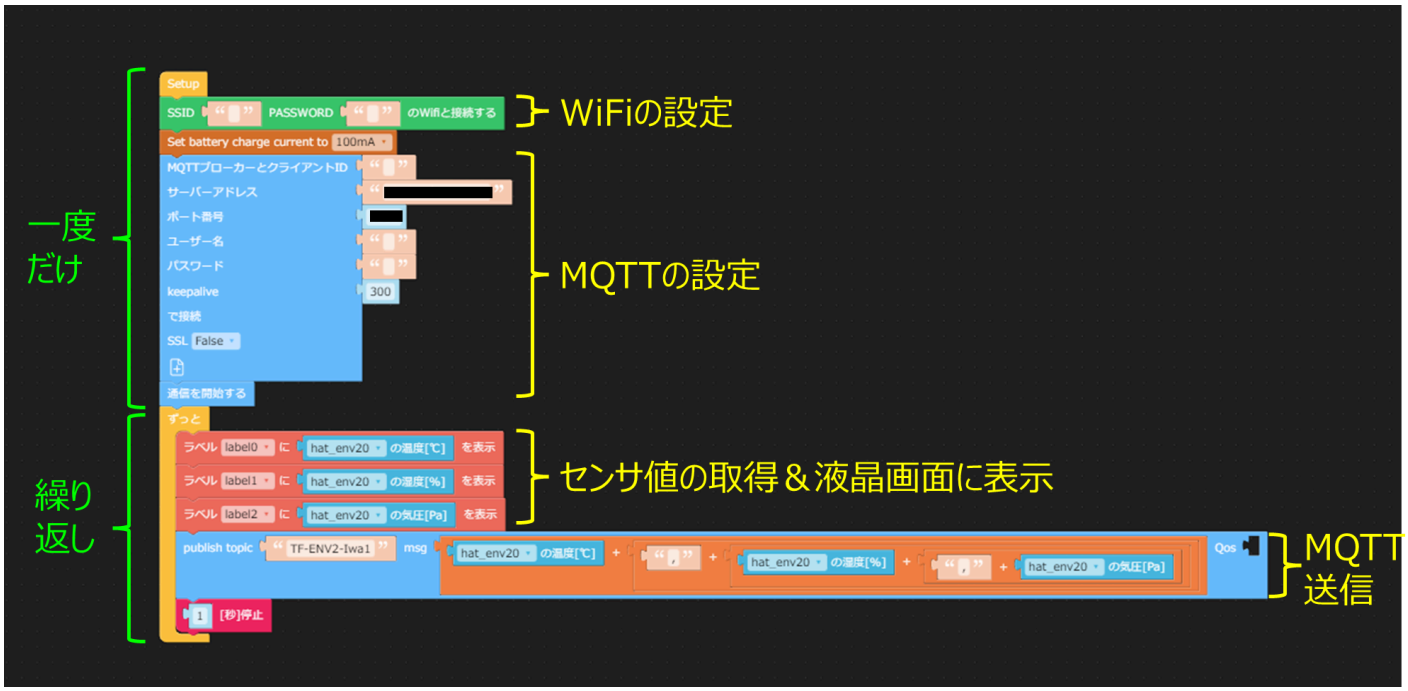


図 5-6　UI-FLOWによる温湿度センサプログラム

UI-FLOWの実際の操作については、「付録2」にも記載する以下の操作動画を参照されたい。

動画：　温度センサの値を画面に表示

動画：　温度センサの値をサーバにMQTT送信

研究所に寄せられるIoT関連の相談のなかには、「工場等の温湿度管理(記録)をIoT技術により自動化することで見回る手間を減らしたい」という要望が多い。そのような要望を迅速に実現して効果を体験するには、今回のように市販品のみを使用し、プログラムも理解しやすい構成は有効である。しかしMQTT送信を含むUI-FLOWのプログラムは一定の時間経過でデータの送信が停止してしまうことが確認されたため、長期間の実運用のためには「WiFI接続が切れている場合は再起動する」や、後述の「ウォッチドックタイマによる再起動」などの追加処理の記載が必要である。

5.2.　無線通信

本システムでは、通信プロトコルとしてMQTTを用いている。MQTTは、軽量なプロトコルであるため、小メモリやネットワーク帯域幅が限られる環境でも活用できるため、IoT分野で注目されている。MQTTはパブリッシャ（送信者）、ブローカ（中継者）、サブスクライバ（受信者）から構成され、パブリッシャから送信されたメッセージは、ブローカが管理するトピックに届けられ、トピックのメッセージは、配信をリクエストしているサブスクライバに配信される。

参考文献：インタフェース2022年11月号　第48巻 第11号 通巻545号　CQ出版社

岩手県工業技術センターでは、セキュリティの観点から所内ネットワークとは別途契約している回線（以下、Wimax回線という）を使用している。しかし、Wimax回線のモバイルルータを設置している部屋と監視対象装置が離れており、直接接続が困難であった。そのため、中継器（WEX-733DHPS、BUFFALO）を用いて接続している。実際の構成を図 5-7 に示す。監視対象装置のある試験室とルータのある事務所は同じ建物にあるが、それぞれ1階と3階である。そのため、中継器を2台設置し、M5Stackとモバイルルータを接続している。この2点間は直線距離としてはそれほど離れていないが、階がことなることと複数の壁に隔てられることから、電波強度が弱くなり接続が途切れやすい。そのため、図 5-8 に示すスペクトラムアナライザWi-Spyと解析ソフトChanalyzerを用いて電波強度を確認し、設置位置の微調整を行った。

図 5-7　中継器を用いたネットワーク接続状況

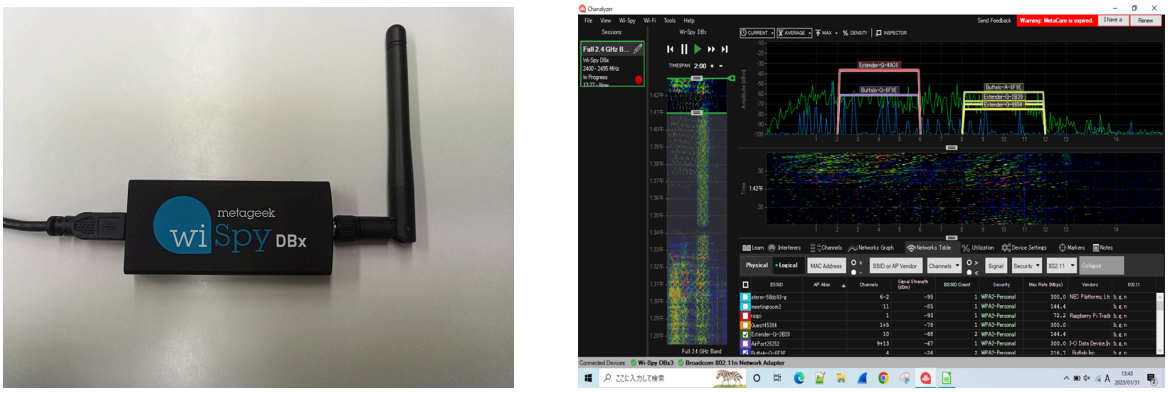


図 5-8　USB接続のスペクトラムアナライザ Wi-Spyと解析ソフトChanalyzer

5.3.　処理方式

5.3.1.　Arduino-IDEを用いたM5Stack用プログラム解説

5.3.1.1.　Arduino-IDEについて

M5Stack用のプログラム開発方法は大きく3つある。

1. UI-Flowを用いた開発
2. MicroPythonを用いた開発
3. Arduino-IDEを用いた開発

ここでは、上記3.のArduino-IDEを用いた開発について説明し、作成したプログラムを解説する。

Arduino-IDEは、イタリアのArduino財団によって開発されたArduinoボード用のプログラム開発環境である。プログラミング言語はC++をベースとしている。Arduino-IDEの最大の特徴は、マイコンを起動するプログラムを作成する必要がなく、簡単に目的の処理を記述するだけで動作可能な環境が提供されていることである。この容易な開発スタイルは、世界中のホビー向けマイコンに取り入れられ、M5StackのマイコンであるESP32のソフトウエア開発環境としてArduino-IDE版が使用可能になっている。Arduino-IDEを用いることで、ESP32用のプログラムであっても、Arduino共通の関数によって記述が可能である。現在、Arduino-IDEにはver2もあるが今回の開発ではver1.86を用いた。

5.3.1.2. プログラムの構成

プログラムは大きく3つの処理部からなる。

- 初期設定部
- 主処理部
- 割り込み処理部

初期設定部は、`setup()`関数に記述し、主に次の処理を行っている。

- 不揮発メモリ読み込み
- 初期値設定
- 入出力ピン設定
- 割り込みタイマー設定
- Wi-Fi設定
- MQTT設定

これらの処理は、回路上のセンサと通信する機能を定義し、割り込みタイマーによる定期処理の準備、Wi-Fi通信の準備などを行っている。Wi-Fiの接続情報は不揮発メモリに予め書き込んでおき、起動毎に読み込んで使う。

主処理部は、`loop()`関数に記述し、主に次の処理を行っている。

- センサ値読み込み（カラーセンサ）
- 正常フラグ設定
- センサ値文字変換
- センサ値転送
- センサ値表示

ほぼセンサに関する処理であるが、カラーセンサの読み込みは主処理部で取得する。一方、電流センサ（CTセンサ）の場合は別の処理で値を取得するため、ここでは取得済みのデータを扱い計算する。センサ値は、伝送およびM5Stackのディスプレイに表示する。

割り込み処理部は、`timerAttachInterrupt()`にポインタで渡された関数で次の処理を行っている。

- センサ値読み込み（CTセンサ）
- 異常時間検出
- 再起動命令

電流センサの場合は定刻かつ多くの数値を必要とするため、正確な時間間隔を保てるタイマー割り込みによって、センサ値取得を行う。異常時間検出と再起動命令は、主処理部がフリーズした場合に、それを検出し再起動を行う。

5.3.2. 各センサの通信方式（アナログ、デジタル）

今回用いたセンサに、CTセンサとカラーセンサがある。CTセンサは電流を電圧変換しESP32内蔵のADコンバータによって電圧を読み取り、電流値に変換する。カラーセンサは、ADコンバータがセンサ側に内蔵されており、センサ側とI2C通信によりセンサ値を取得する。I2C通信部分はArduino-IDEの共通関数があり容易に組み込むことができる。

5.3.3. 電流データの分析（フーリエ変換、リアルタイム処理）

CTセンサの測定対象は交流電源の電流である。交流の電流変化から、実効値に変換する必要がある。実効値は一定時間に取得した数値で積分したものであるため、先述した通り、タイマー割り込みによって一定間隔で、交流周期20ms以上の数値列を取得する。

有効電力を求めたい場合は、電流の基本波成分を用いることで近い値を求めることができる。基本波成分は50Hzの複素三角関数を乗じて2乗平均平方根を求めることで得られる。

5.3.4. 送信するデータ

センサデータは数値で表現される。これを文字列に変換し、サーバ側に送信する。数値データの送信は大きく2つ考えられる。

1. データの種類を付与して数値データと一緒に送信
2. 数値データだけ送信

今回は、MQTTによって、トピックスを切り替えることができるため、受信側は何の数値であるかは判っている。よって、送信側ではあえて名称を付けずに、数値データだけを送信している。

5.3.5. ウォッチドッグタイマによる再起動

ウォッチドッグタイマとは、一定の期間内に、機能がきちんと動いているかどうかを確認するタイマーである。この機能はタイマー割り込みのカウンタで実現している。一定期間は、割り込み時間間隔とカウンタ上限値とで設定する。

例えば、主たる処理が60秒以上かかる場合は異常であり、異常時は再起動する、という設計であれば、次のようになる。

- 主処理
    - 処理（この処理は60秒以内に終わるのが通常。それ以上は異常。）
    - ウォッチドッグ用カウンタをクリア（0にする）
    - （先頭に戻る）
- 割り込み処理（10秒毎）
    - ウォッチドッグ用カウンタを加算（10秒に1加算）
    - ウォッチドッグ用カウンタが上限値ならば再起動：（6以上つまり60秒間クリアされてなければ再起動）

割り込み処理は、主処理を行っている最中も実行されるため、主処理で無限ループなどに陥っていても、割り込みによって再起動がなされる。これにより止まってしまうことを防ぐことができる。

6. データフロー

6.1. フロー概要

Node-Redはフローベースのビジュアルプログラミング言語であり、C言語やJava言語と比較すると簡単にプログラムを作成することができる（図 6-1）。

センサからのデータはMQTTプロトコルで送られてくる。センサデータとして振動試験装置の消費電流、消費電力、電源スイッチランプのカラーセンサ情報、振動試験装置が設置されている部屋の室温・湿度、その他の装置の消費電力情報がある。

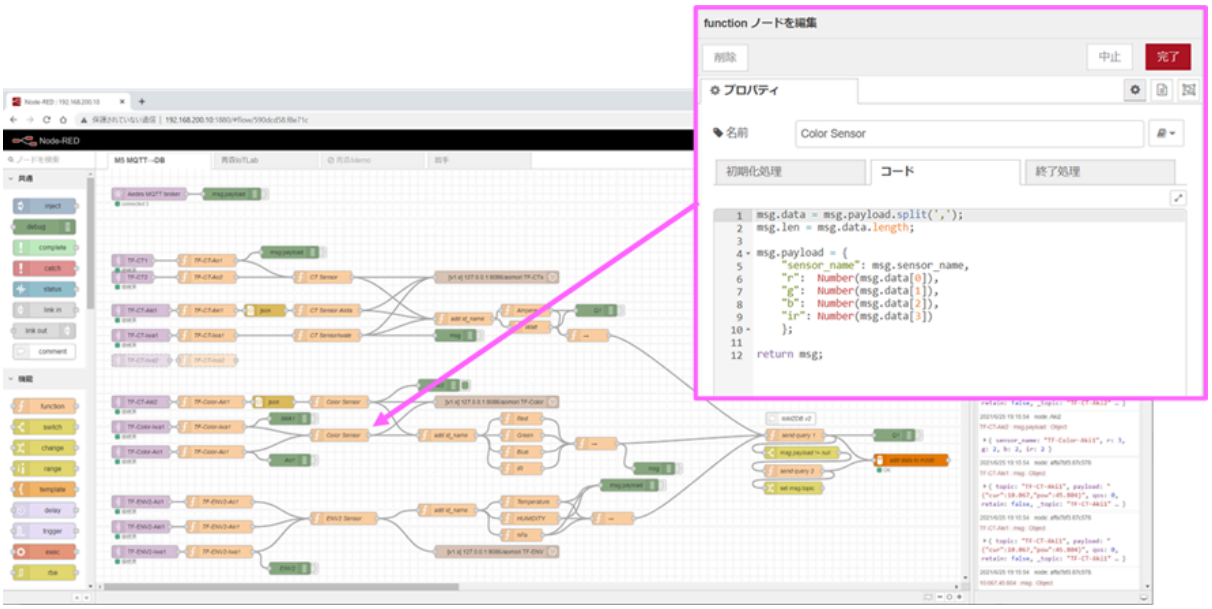


図 6-1　Node-Redの開発画面

青森に設置したサーバのNode-redにフローを作成し、無線センサ端末として用いたM5StackからのMQTT通信を受信し、データベースに保存した。フロー図を図 6-2 に示す。

各所に設置したPCにはNode-redをインストールし、各所に設置したM5StackからのMQTT通信を青森サーバに転送するフローを作成した。フロー図を図 6-3 に示す。

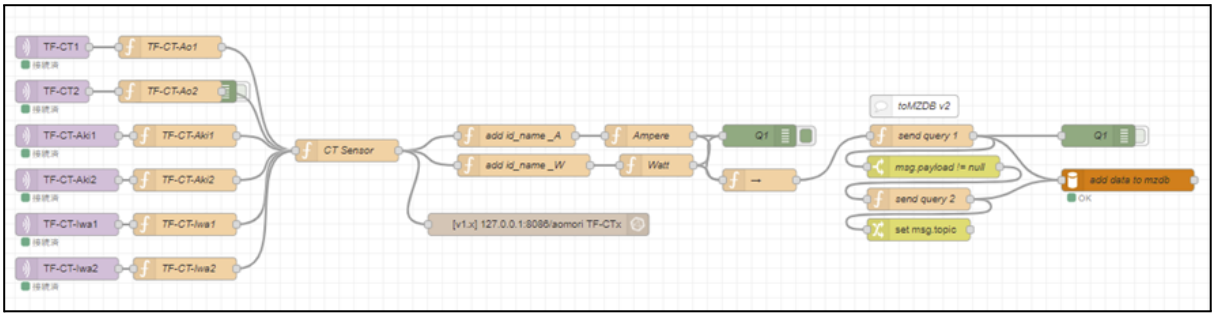


図 6-2　M5StackからのMQTT通信をデータベースに保存するフロー

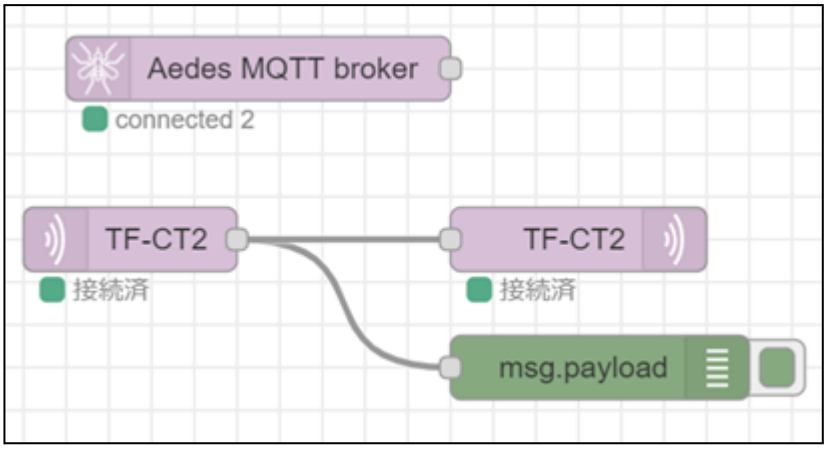


図 6-3　各所PCでM5StackからのMQTT通信を青森サーバに転送するフロー

6.2.　演算処理

振動試験装置に電流センサとして取り付けたM5Stackからは電流値と電力値がMQTTプロトコルによって送られてくる。

青森のDBサーバ手前に設置した、インターネット側からMQTTを受け付ける中継サーバには、下記のようなフローを設定しMQTTデータを転送している。

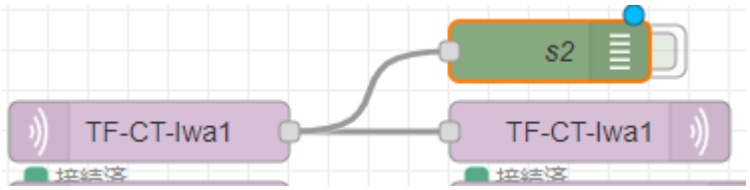


図 6-4　インターネット側からDBサーバへのMQTT転送フロー

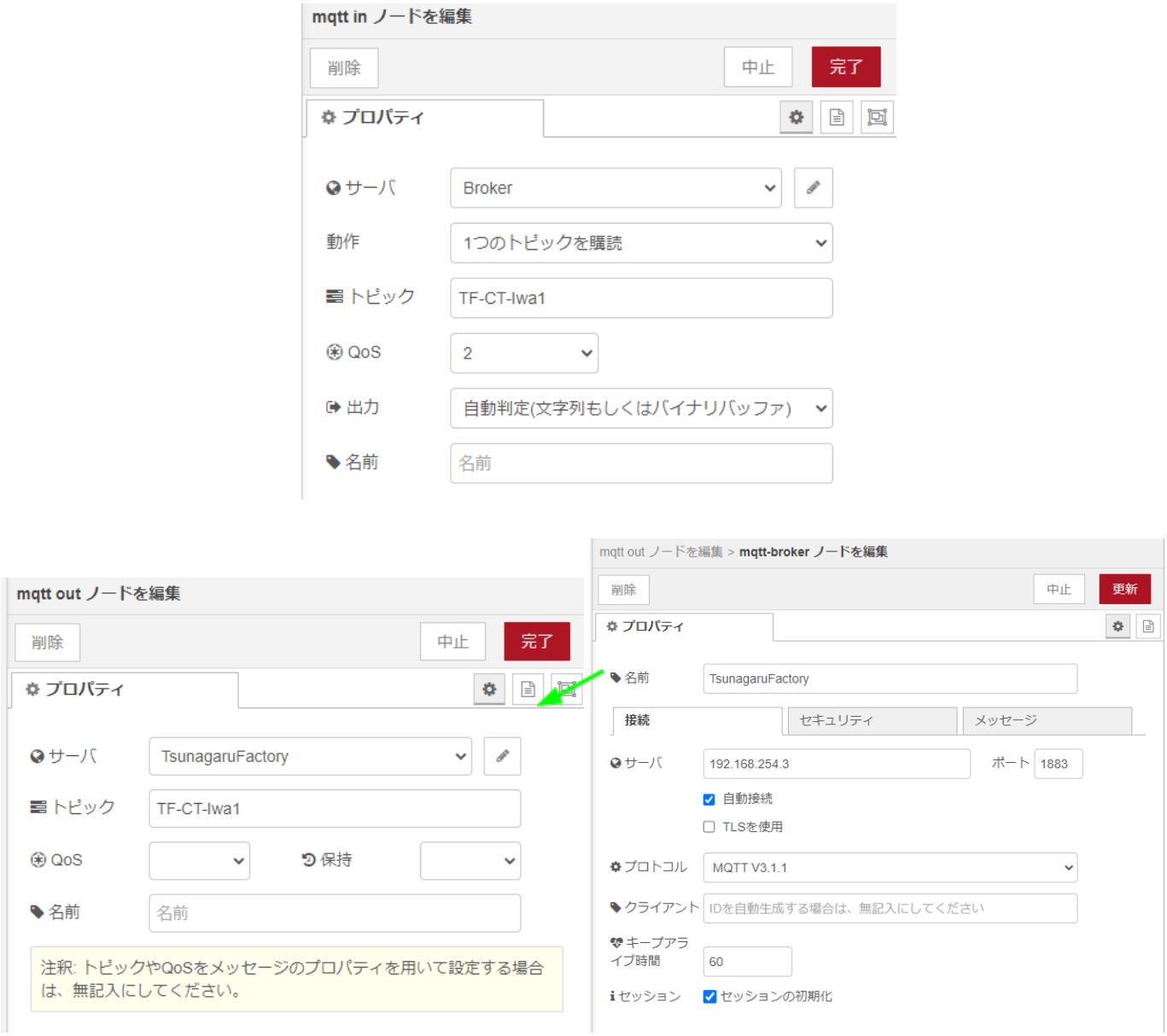


図 6-5　MQTTノードの設定

MQTT outノードの送信先が192.168.254.3となっているが、これはVPNルーターのIPアドレスである。VPNルーターでは中継用サーバ(192.168.254.118)からの通信は全て青森サーバ(192.168.200.10)に転送するよう設定を行っている（図 6-5）。

複数のMQTT通信を転送しているため、フローは図 6-6 のようになっている。

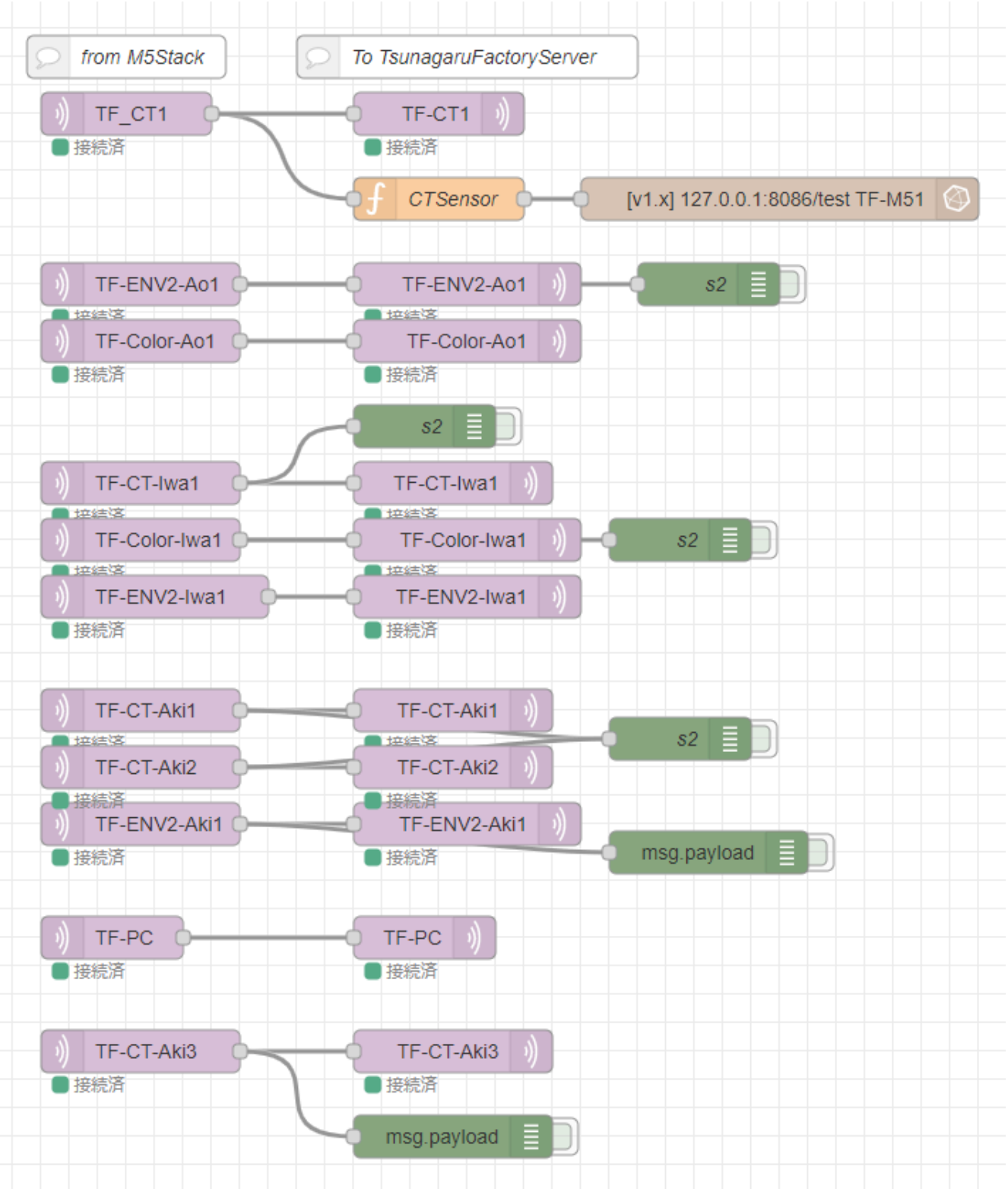
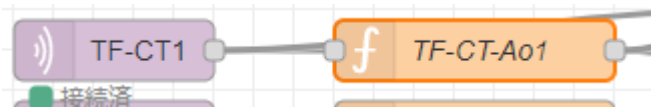
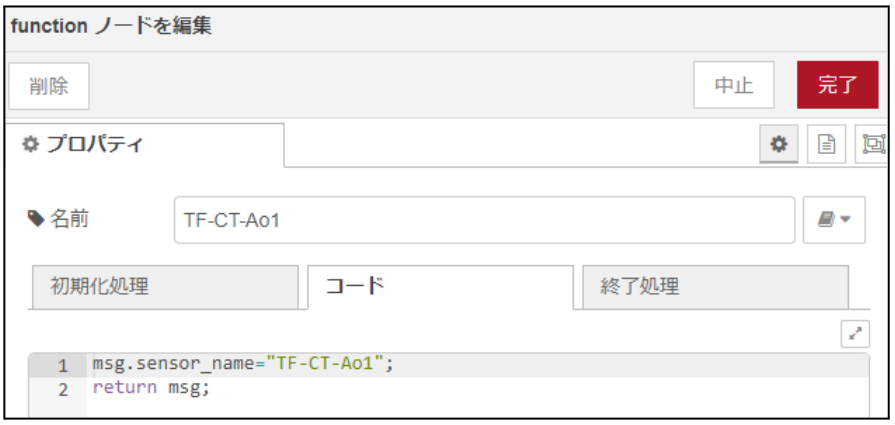


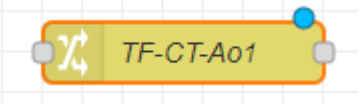
図 6-6　MQTT通信転送フロー部

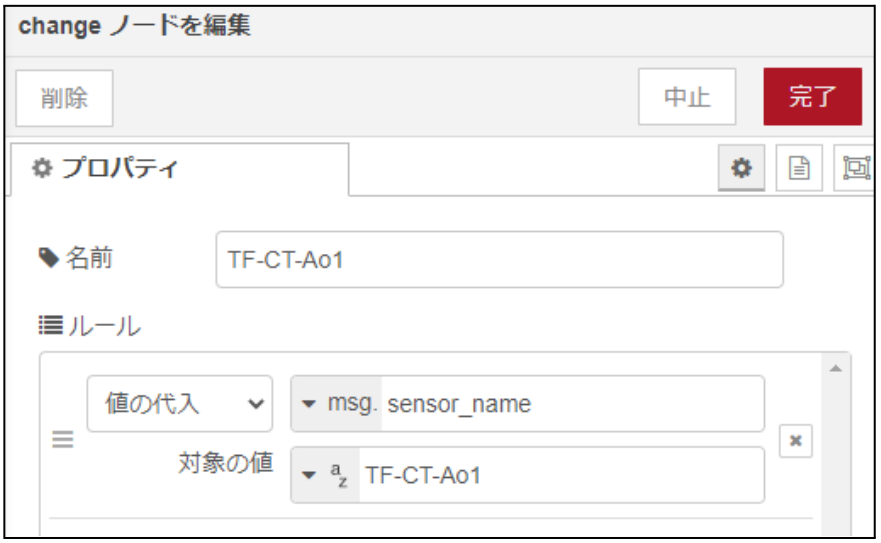
電流センサは青森、秋田、岩手にそれぞれ取り付けているため、データ表示時にわかりやすくするためMQTTのトピック名ごとにセンサ名の名付けを行った。システム構築時はNode-Redの使用方法に慣れておらずFunctionノードを使用して、ラベル名：sensor_nameとして青森のCTセンサ1番目には"TF-CT-Ao1"と名前をつけた。先頭のTFはTsunagaruFactoryの略である。





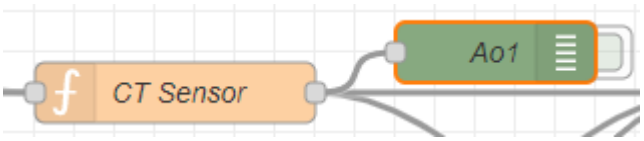
プログラムコードに慣れていないのであれば、同等の機能をchangeノードを使うことでも実現でき、表示もわかりやすいため、こちらをおすすめする。





MQTTで受信したセンサデータは、msg.payload内に「"19.099 , 3144.725"」と、文字列としてデータが格納されている。データベースには小数点を含む数値として保存したいため、文字列を数値に変換する必要がある。Node-Redでは「Number()」関数を使用することで数値に変換可能である。

また、influxdbにデータ登録する際、payloadを「項目名:数値」という書式でinfluxdbノードに渡すと正しくデータ登録可能であるため、下図17～19行目でpayloadの書式を変更している。



function ノードを編集

削除　中止　完了

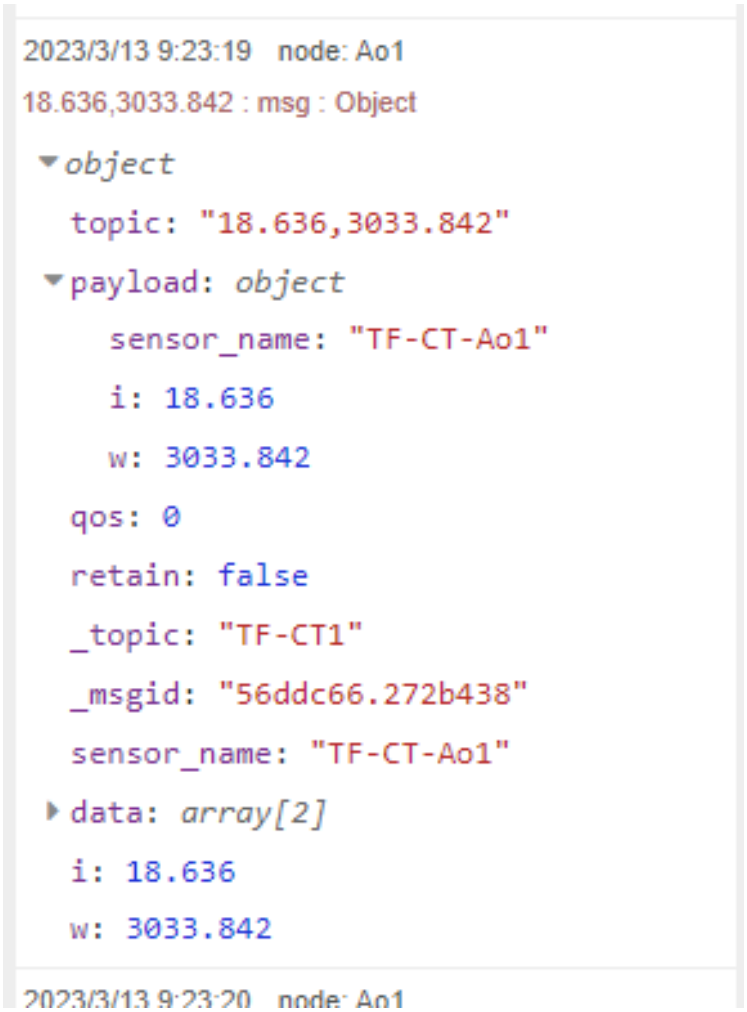
プロパティ

名前　CT Sensor

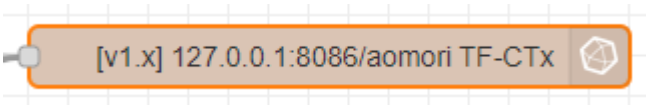
初期化処理　コード　終了処理

```
1  msg.data = msg.payload.split(',');
2  //sprintf(cValue, "%3.3f , %3.3f", fRms*FACT_CUR, fPow );
3  msg.i=Number(msg.data[0]);
4  msg.w=Number(msg.data[1]);
5  msg.topic=String(msg.i)+","+String(msg.w);
6  /*
7  //msg.ss = "M54";
8  msg.ss = msg.topic.slice(0,-1);
9  msg.t = Number(msg.payload[msg.ss + "t"]);
10 msg.s = Number(msg.payload[msg.ss + "s"]);
11 msg.k = Number(msg.payload[msg.ss + "k"]);
12 msg.b = Number(msg.payload[msg.ss + "b"]);
13
14 if( Number.isFinite(msg.t) && Number.isFinite(msg.s) && Number.isFinite
15 */
16 msg.payload = {
17     "sensor_name": msg.sensor_name,
18     "i": msg.i,
19     "w": msg.w
20 };
21
22 return msg;
23
```

デバッグタブでmsgオブジェクト全体を出力した図を下図に示す。



influxDBに書き込むため、payloadの書式を整えた後はinfluxdbにデータ登録を行う。初期状態のNode-Redではinfluxdbのサーバ情報を登録していないため、Serverの欄が「新規にinfluxdbを追加」表示になっている。

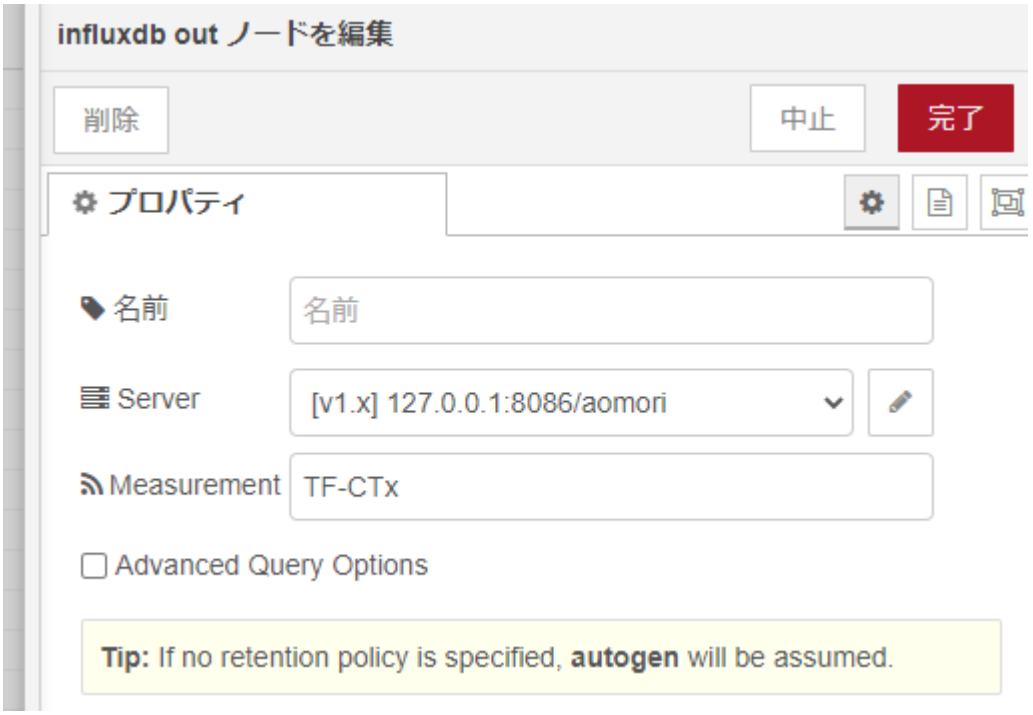


「新規にinfluxdbを追加」の右隣の アイコンをクリックし、サーバ情報を登録する。

Database名はinfluxdbインストール後、初期設定として作成したデータベース名に合わせて、更新ボタンを押す。

正しくデータベースが登録できると下図のようになり、Measurementにはセンサグループ名を設定した。センサ毎にMeasurement名を設定しても良いし、同一種類のセンサは同じMeasurementにしても良いが、データ登録時にセンサ名やセンサ番号なども一緒に登録しないと、データ検索時にどのセンサのデータなのか見分けがつかなくなる恐れがある。

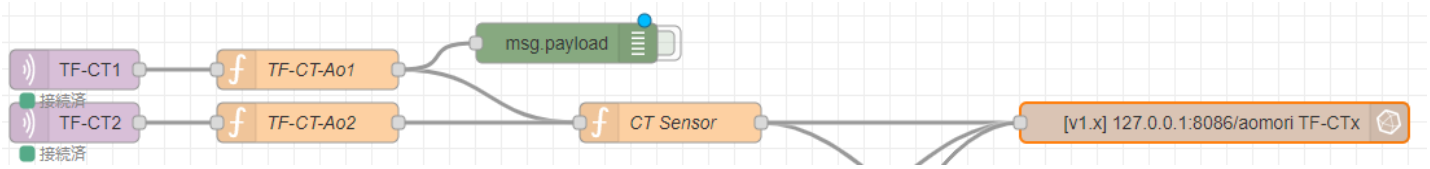
MeasurementはGrafana での可視化時に候補として表示される機能がある。influxdbでデータの削除を行う場合、「drop Measurement名」コマンドで簡単にMeasurementを削除することも可能である。



ここまでのフロー(下図)でM5Stackから送られてくるMQTTデータをinfluxdbデータベースに書込み可能である。

influxdbでは項目名やデータタイプを事前に定義しておかなくても自動で認識し格納することが可能であるため、MariaDBと比較すると簡単にデータ蓄積が可能である。

しかし、データの「型」を間違えて文字列としてデータベースに登録してしまったり、温度センサの数値としてのデータなのに文字列と数値のデータが混在してデータ登録してしまったりする場合もあるため、慣れていない場合は注意が必要である。数値を文字列としてデータベースに登録しても一見問題ないように表示はされるが、Grafanaでの可視化時に他の数値との比較や算術演算が使えないといったトラブルが発生する。



次にMZPlatformでの処理・表示用にMariaDBにデータを書き込むフローを説明する。

MariaDBはMySQLから派生したデータベースであるため、MySQLと共通点が多い。 influxdbではそれほど気にしていなかった「SQL文、テーブル、カラム、レコード、フィールド」を取り扱うため、用語の意味や機能がわからなければ、ネット検索等を参照してほしい。また、データベース設計も必要である。今回は産総研が開発したMZプラットフォームの「MZ講習試料IoTサーバ編.pdf」を参考にシステム構築を行ったので、詳しくはそちらを参照されたし。当該資料は「MZ Platform IoT Toolkit (スマート製造ツールキット/MZ-EX/IoT化用コンテンツ集)」の中の「server」フォルダ内に格納されており、この資料のダウンロードには無料のユーザー登録が必要である。

MZプラットフォーム　https://ssl.monozukuri.org/mzplatform/

MariaDBではセンサデータを蓄積するテーブル「iot_events」と最後のセンサデータのみを蓄積するデーブル「iot_events_latest」を主に使用した。参考にiot_events_latestのテーブルは下記のようになっている。

- time　　:センサデータのサーバ受信時刻
- source　　:センサ名や設置場所名
- type　　:センサ種別
- value　　:測定値

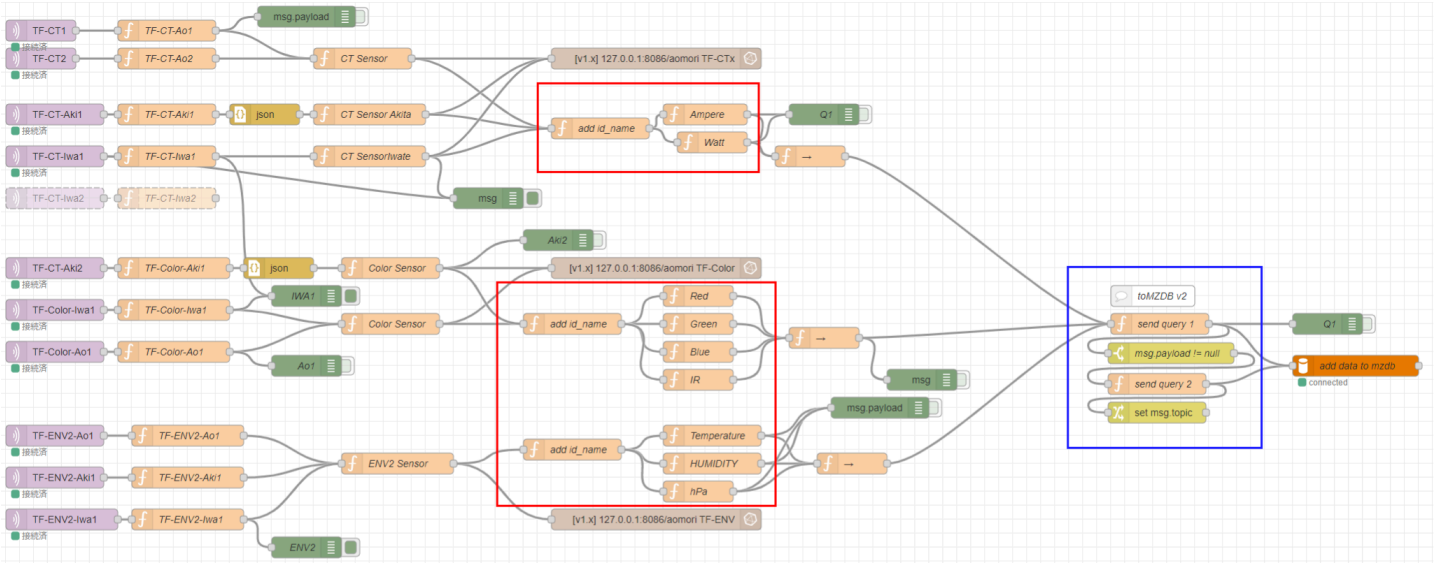
「iot_events」と「iot_events_latest」テーブルを初期登録するSQL文は下記の通りである。

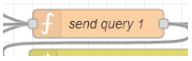
```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";
CREATE TABLE `iot_events` (
  `id` bigint(20) NOT NULL,
  `time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  `source` varchar(50) DEFAULT NULL,
  `type` varchar(20) DEFAULT NULL,
  `value` varchar(20) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

CREATE TABLE `iot_events_latest` (
  `time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  `source` varchar(50) NOT NULL,
  `type` varchar(20) NOT NULL,
  `value` varchar(20) DEFAULT NULL
) ENGINE=MEMORY DEFAULT CHARSET=utf8;
```

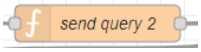
Node-RedからMariaDBにデータを登録するにはSQL文を生成する必要があり、テーブルに必要なセンサ名やセンサ種別等を下図赤枠内のフローで追加している。

SQL文の生成は下図青枠内で行っている。





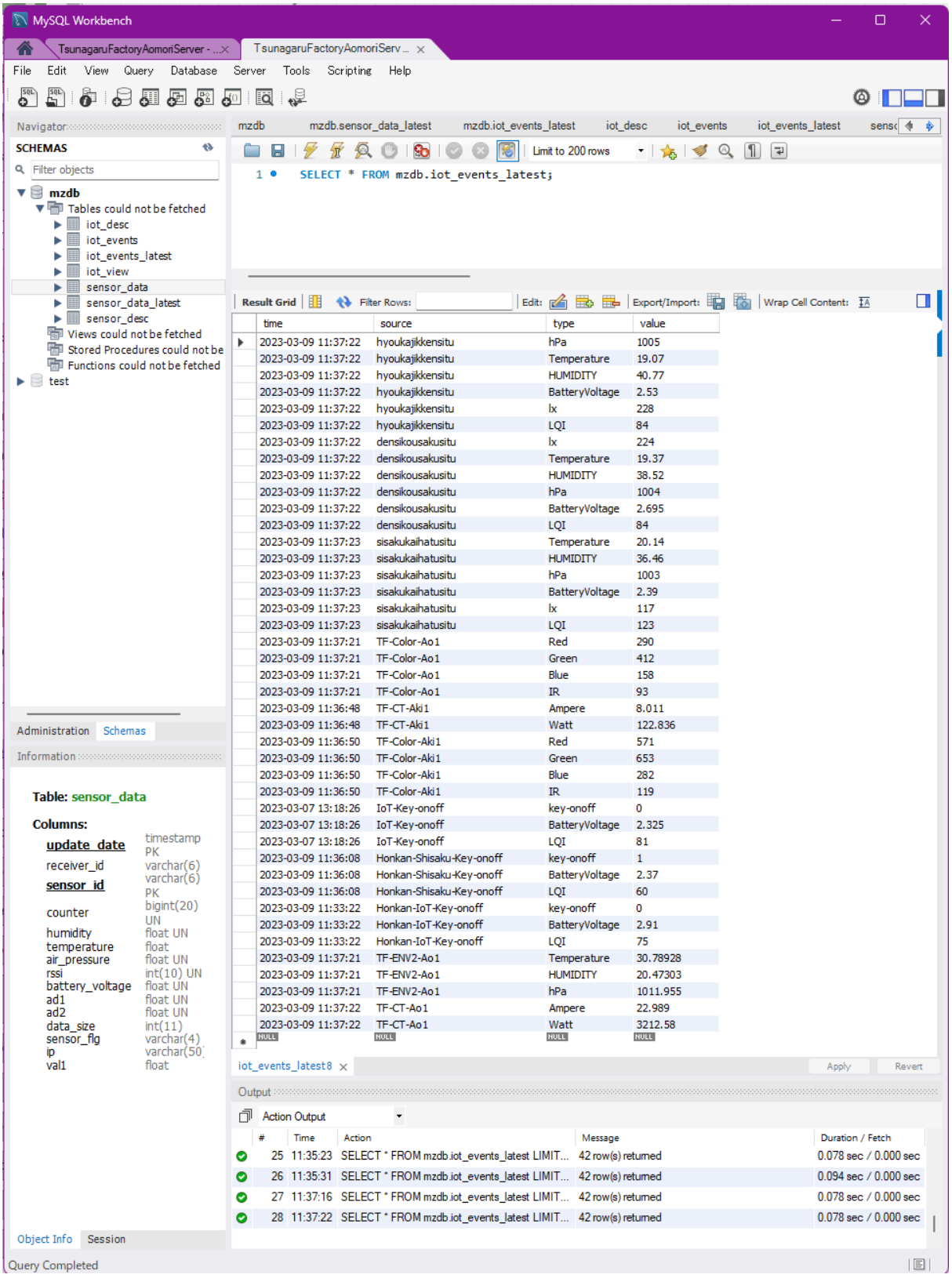
は「iot_events」テーブルにデータ追加するためのSQL分を生成し、



は「iot_events_latest」テーブルにデータ追加するためのSQL文を生成している。

参考に「send query 2」ノードの内容を下図に示す。

参考にMariaDB内のiot_events_latestのレコードをMySQL Workbenchソフトウェアを使用して表示したものを下記に示す。



6.3. アラートシステム

Node-Redからメール送信を行い、通知することも可能である。

パレットの管理からメールに関する機能のノードを追加して使用するが、古いノードでは機能しない場合もある。下記情報の動作は未確認であるため他のインターネット検索情報等も参考にされたし。

https://digital-light.jp/2022/12/29/how-to-send-email-from-nodered/

参考にGrafanaによるアラート実装例を7.2に記載する。

7. 可視化システム

7.1. MZプラットフォーム

産総研が開発したMZプラットフォームを使用して下記のような可視化システムを作成した。

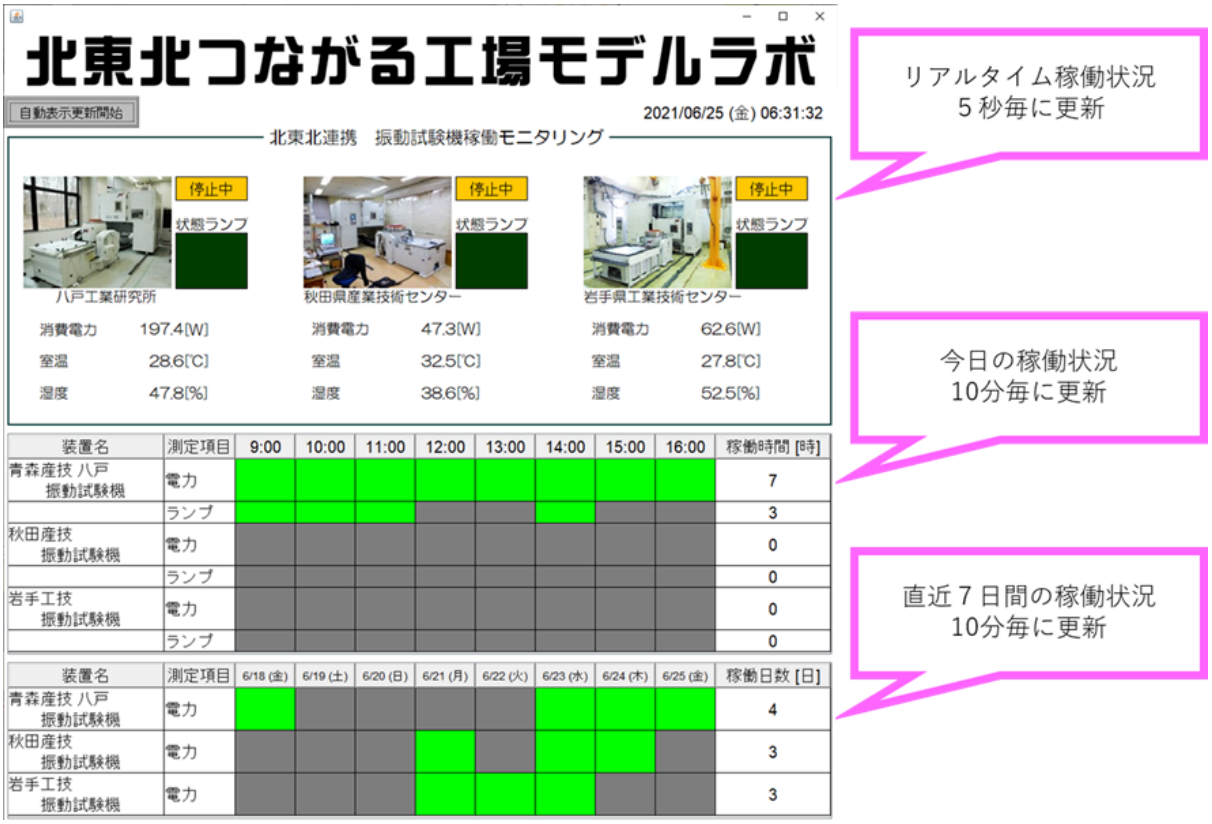


図 13　稼働モニタ画面

上半分には各拠点の振動試験機写真と稼働状態がひと目で分かるモニタリング表示、中央部には今日1日の時間別稼働状況表示、下部には直近7日間の稼働状況表示を設けた。

MZプラットフォームは下図のような「アプリケーションビルダー」を用いて開発を行う。アプリ開発マニュアルについてはMZプラットフォームユーザー会のHPを参照されたし。

https://ssl.monozukuri.org/mzplatform/download/

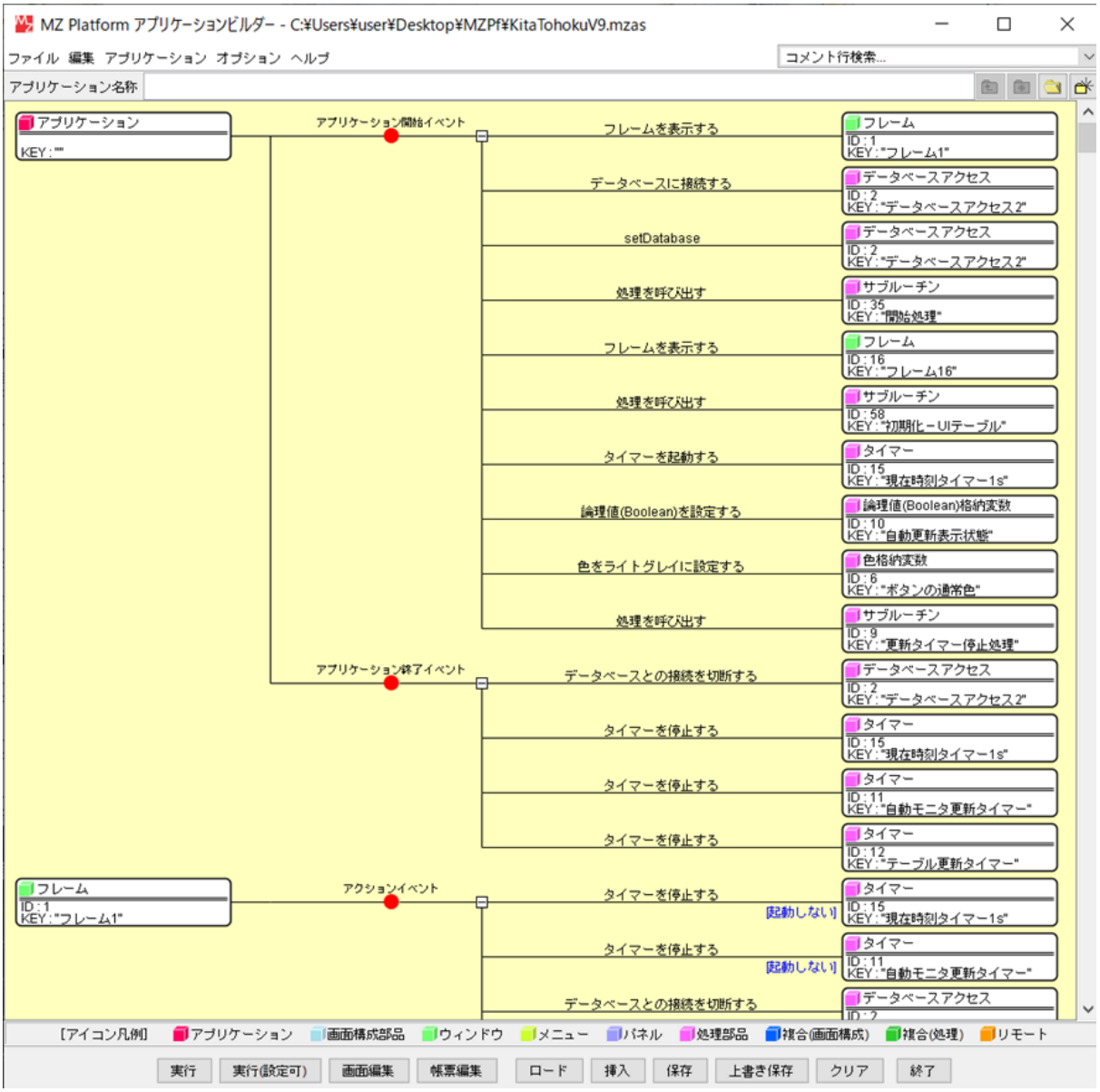


図 12　アプリケーションビルダー

北東北のVPN接続のネットワーク内でなければ動作しないが、参考として作成したプログラムは下記リンクからダウンロード可能である。

KitaTohokuV15 passblank.mzax　KitaTohokuV15 passblank.mzas

7.2. Grafana

オープンソースのデータ可視化ツールであるGrafanaを使って作成した可視化画面を図○○に示す。3拠点の振動試験機について、電流や電源ランプの明るさ（色）、周囲温度の変化がリアルタイムでグラフに描画される。

図　計測データの可視化

Grafanaは、InfluxDBやMySQLなど、多くのデータソースをサポートしているほか、ダッシュボードの操作に柔軟性があり、さまざまな種類のグラフ、テキスト、画像、およびテーブルなどを自由に配置できるため、独自のダッシュボードをカスタマイズできる。

- 参考に今回構築したダッシュボードデータを置く。（インポートにより再現可能）

https://drive.google.com/file/d/1-SJ__7Owl4C4-HrilkGJzWazJZn-pMv8/view?usp=share_link

ダッシュボードの作成方法など、実際の操作については「付録2」にも記載した以下の動画を参照されたい。ここでは、Node-RedによってInfluxDBに登録されたデータをグラフ表示するまでの手順を記録している。

動画：　受信データをGrafanaで確認

また補足として、Grafanaは各種データに閾値を設けるなどしてアラートを設定できる。設定したアラートは、Eメール、Slackなどの方法で通知を受けることができる。他の事業ではビジネスチャットツールであるSlackへの通知を試行したことがある。おおまかな手順は以下の通りである。

1．Webhook URLの取得

2. Grafanaでアラートの設定

3. Dashboardでアラートの設定

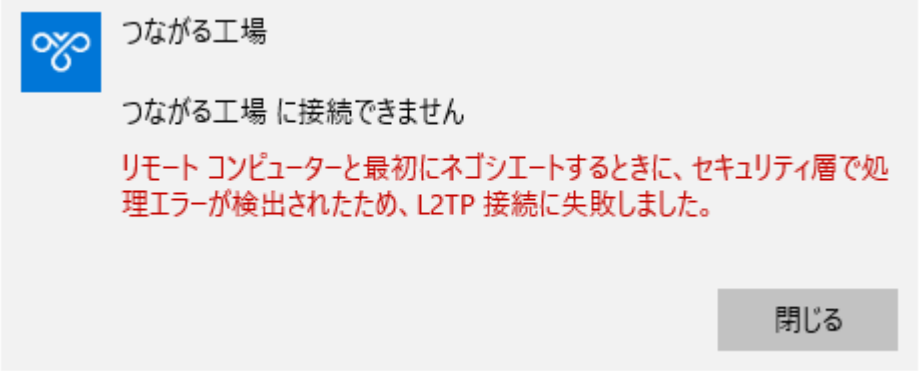
詳細については以下のサイトを参考にした。

参考サイト：https://www.n-novice.com/entry/2019/01/16/000000

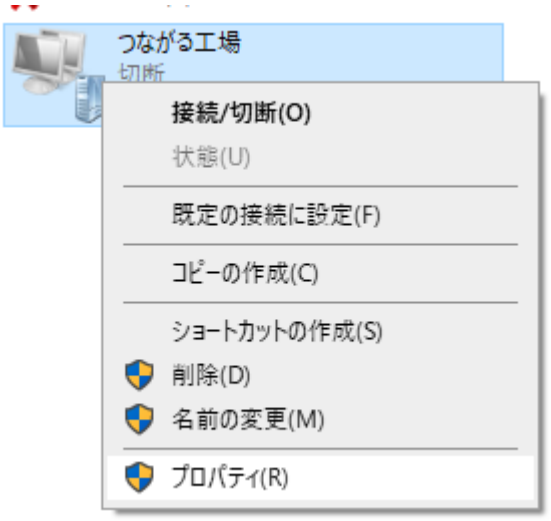
8. 付録1：失敗事例

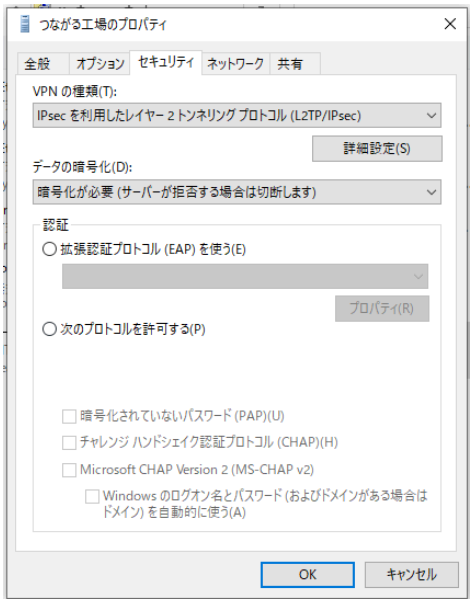
8.1. VPN構築の落とし穴

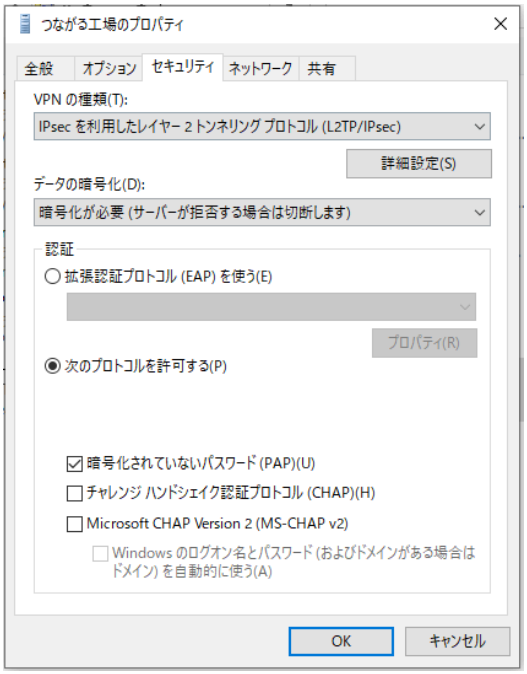
・2023/03/08　マニュアル作成のために環境の確認等をしていたが、VPNへの接続がうまくいかない。



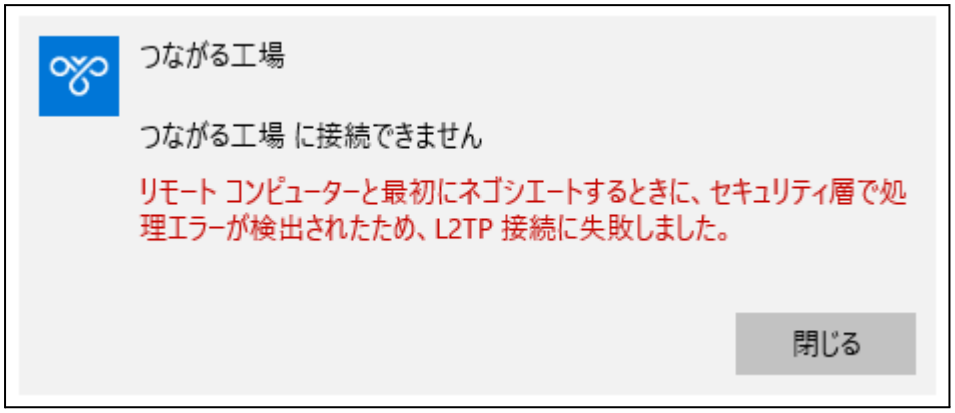
とのことなので、設定の「アダプターのオプションを変更する」→VPNのネットワーク接続のネットワークアダプタのプロパティから「セキュリティ」タブを確認すると、「次のプロトコルを許可する」が空欄になっている。そのため、再度設定、OKボタンを押す。



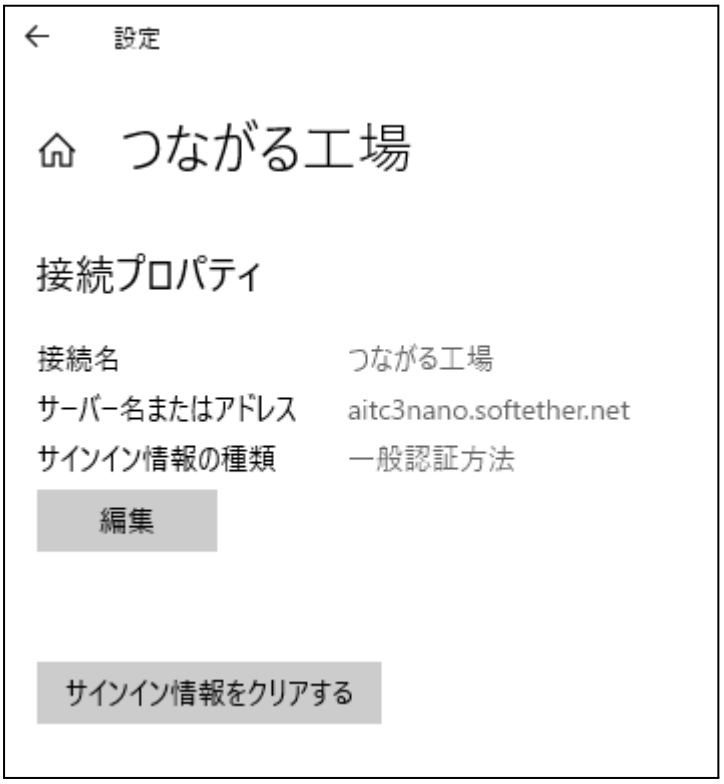




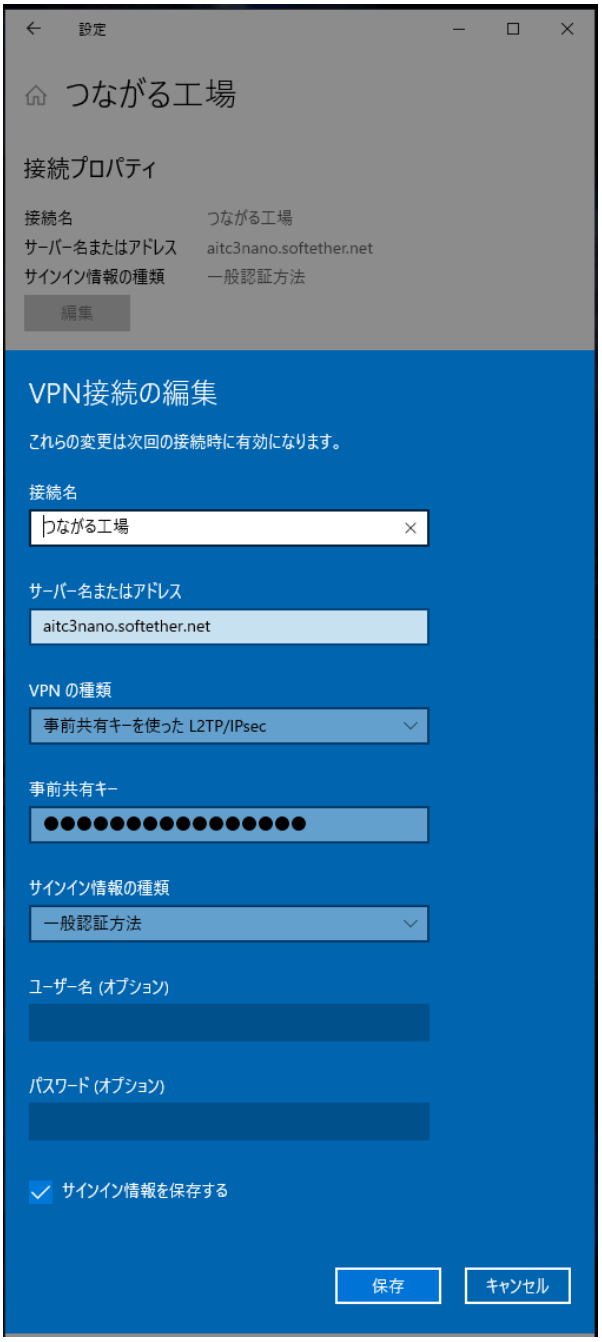
そして、VPNの接続を行うと、先ほどと同じエラーが出た。

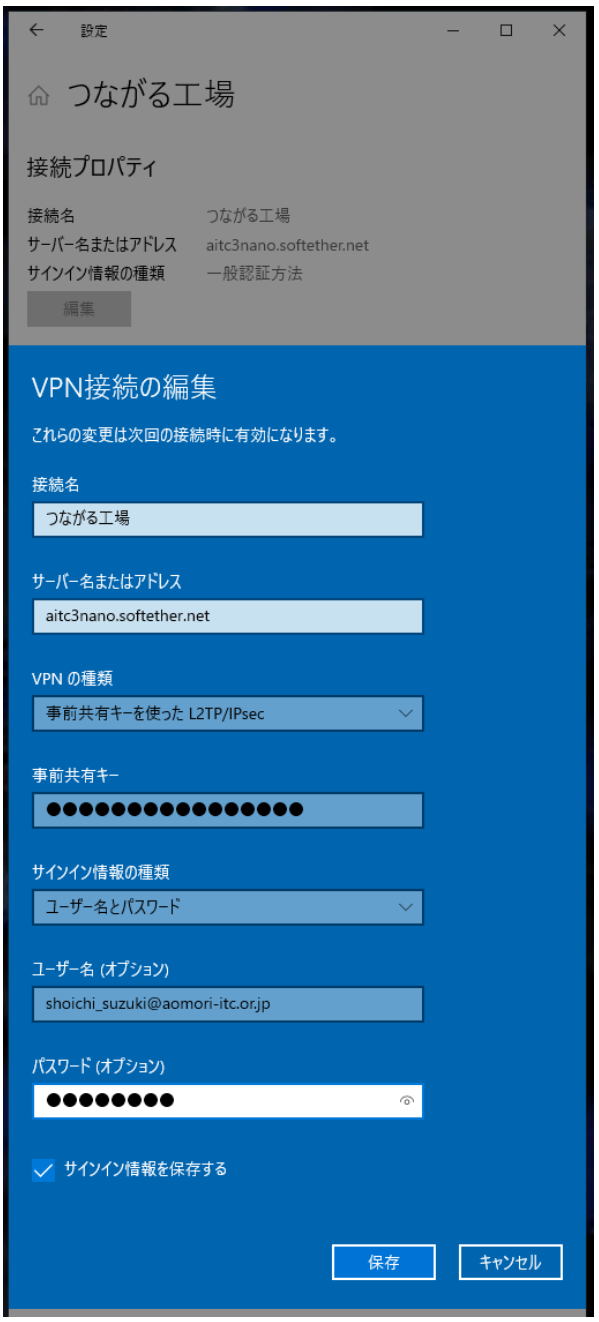


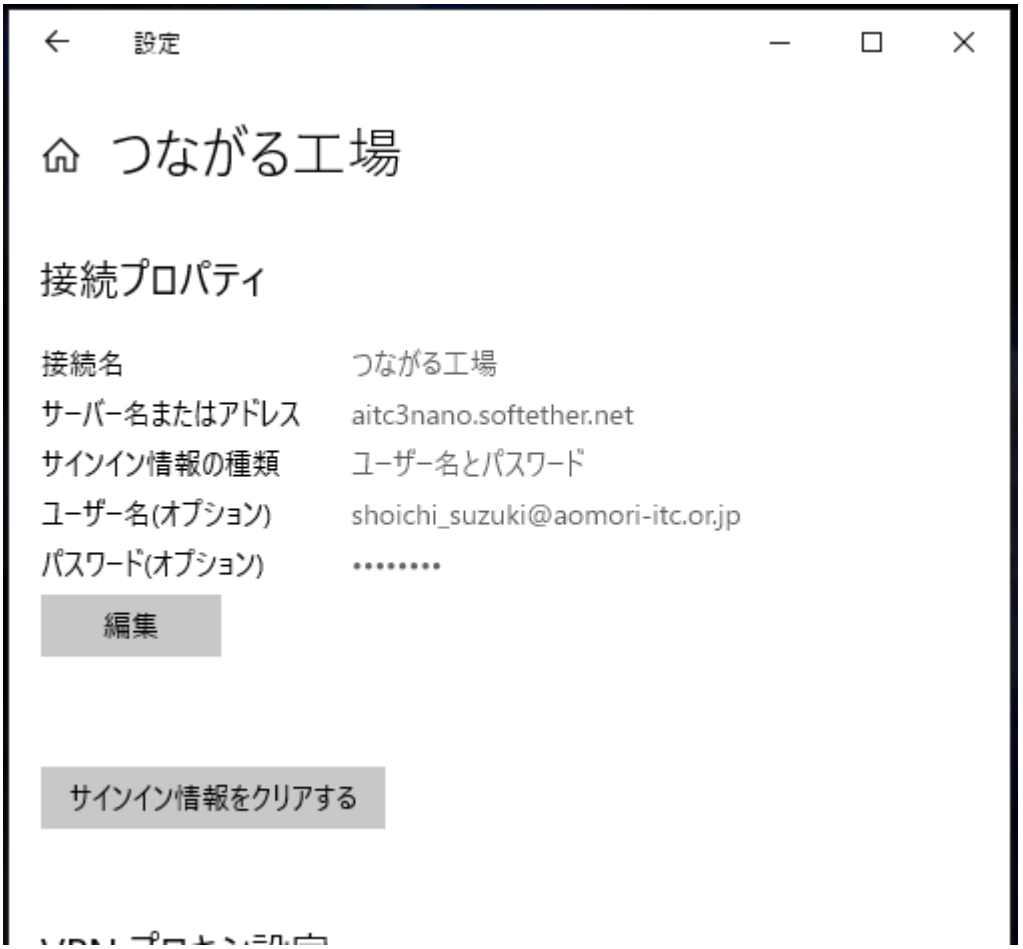
詳細オプションで設定内容を確認すると、サインイン情報の種類が一般認証方式に変わっている。



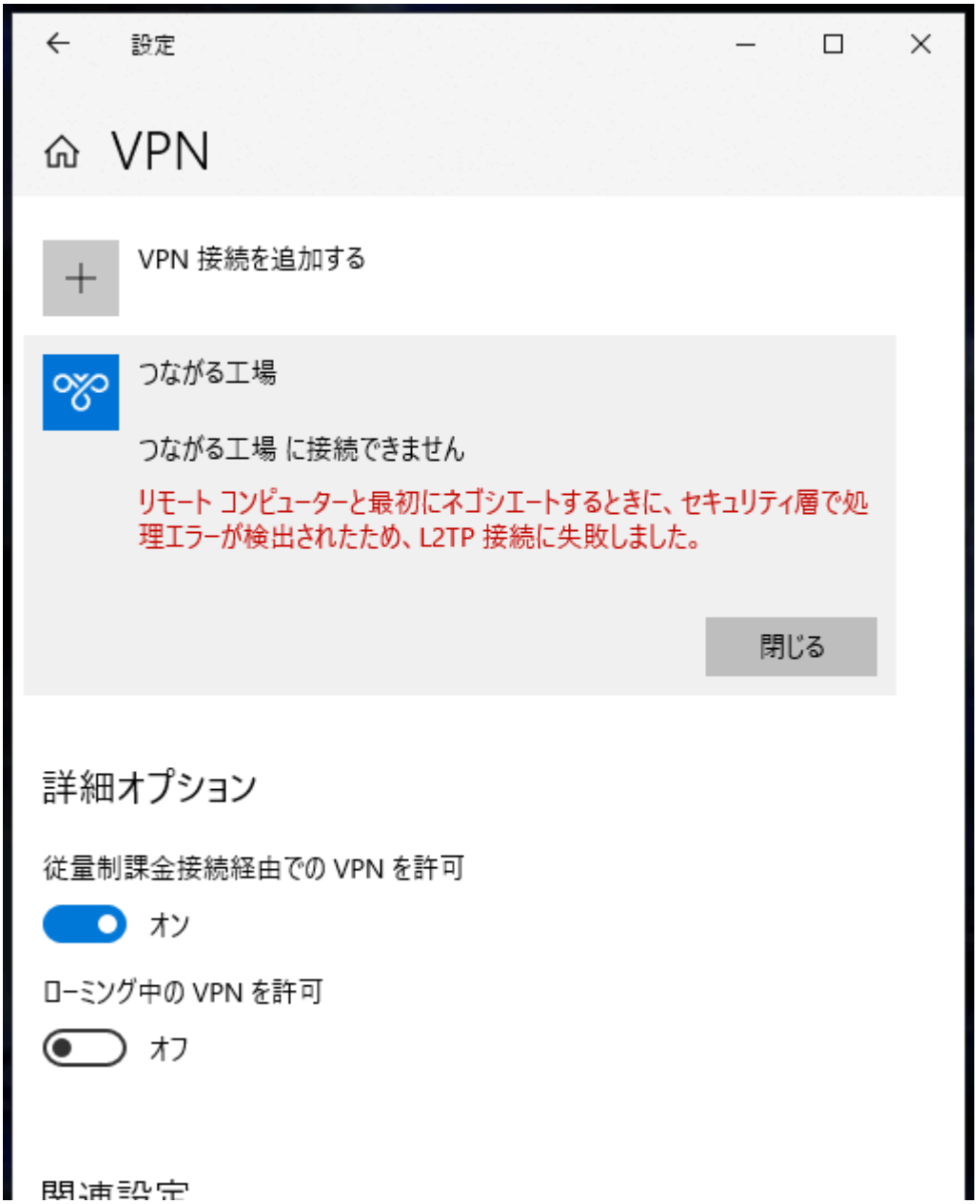
なので、編集ボタンでサインイン情報の種類をユーザー名とパスワードに変更し、保存ボタンを押して正しく設定を行う。


設定
つながる工場
接続プロパティ
接続名 つながる工場
サーバー名またはアドレス aitc3nano.softether.net
サインイン情報の種類 一般認証方法
編集
VPN接続の編集
これらの変更は次回の接続時に有効になります。
接続名
つながる工場
サーバー名またはアドレス
aitc3nano.softether.net
VPN の種類
事前共有キーを使った L2TP/IPsec
事前共有キー
●●●●●●●●●●●●●●●●
サインイン情報の種類
一般認証方法
ユーザー名 (オプション)
パスワード (オプション)
サインイン情報を保存する
保存
キャンセル


設定
つながる工場
接続プロパティ
接続名 つながる工場
サーバー名またはアドレス aitc3nano.softether.net
サインイン情報の種類 一般認証方法
編集
VPN接続の編集
これらの変更は次回の接続時に有効になります。
接続名
つながる工場
サーバー名またはアドレス
aitc3nano.softether.net
VPN の種類
事前共有キーを使った L2TP/IPsec
事前共有キー
●●●●●●●●●●●●●●●●
サインイン情報の種類
ユーザー名とパスワード
ユーザー名 (オプション)
shoichi_suzuki@aomori-itc.or.jp
パスワード (オプション)
●●●●●●●●
サインイン情報を保存する
保存
キャンセル

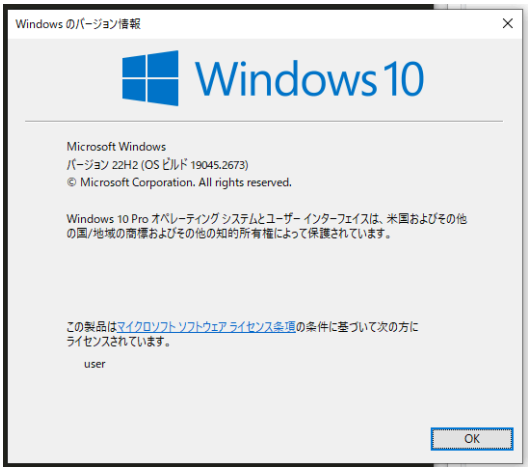


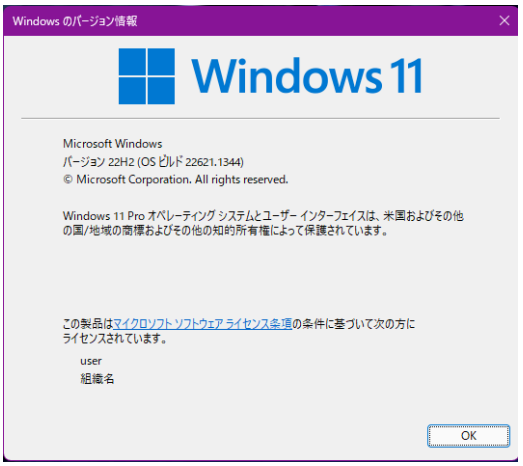
VPN接続を行ってみると、同じエラーが表示される。



設定とネットワーク接続のプロパティでどちらを設定しても八戸Windows10ではVPN接続ができなかった。別のWindows11PCでは接続エラーのたびにどちらかを修正して接続を試すと繋がることがわかった。

八戸Windows10と八戸Windwos11のバージョン情報





エ総研Windows11PCでも接続できないものがあった。

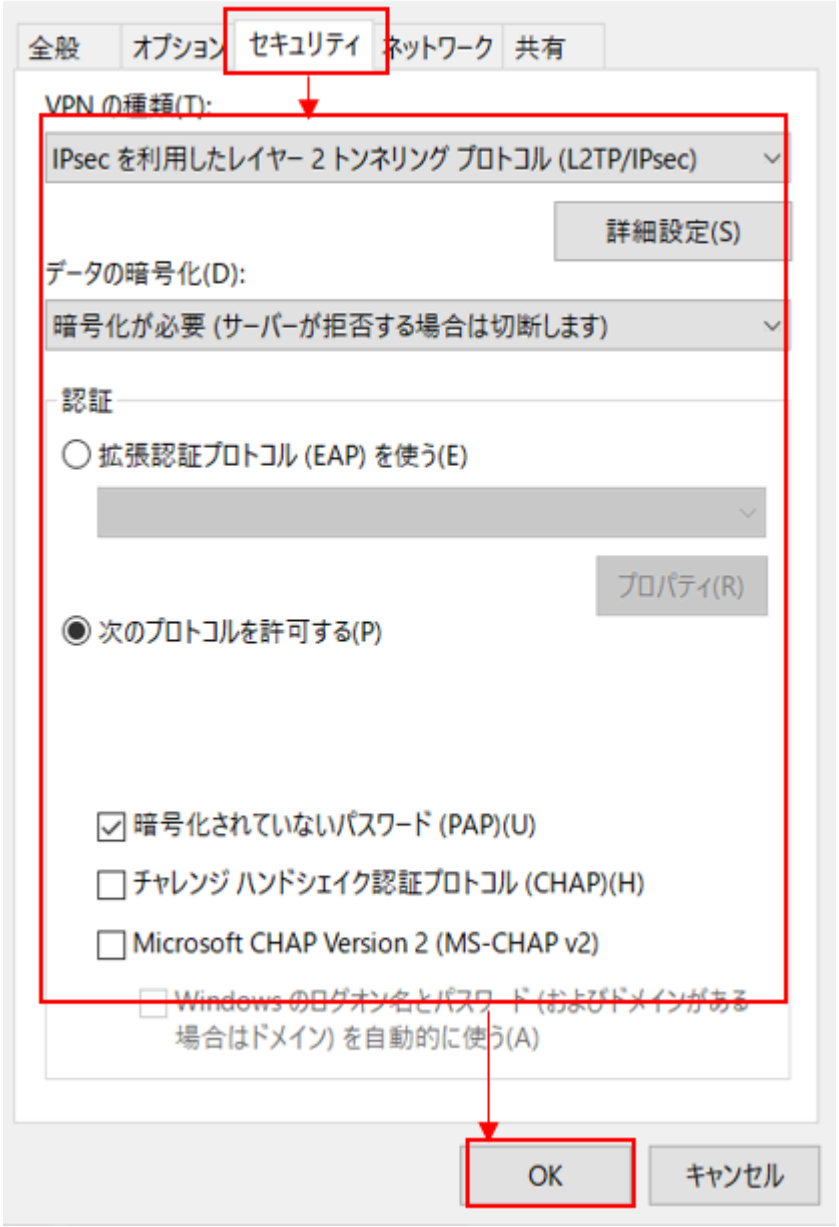
⇒アダプターのオプションを変更する⇒セキュリティ

において以下のとおり設定することで接続することができた。

なお、アダプターのオプション変更画面表示方法は下記サイトを参照されたい。

Windows 11 でのアダプターオプションの詳細（ネットワーク接続）の出し方 (zero.jp)

セキュリティ設定は今回のVPN接続では以下のとおりとした。



●但し、運用時は常時接続する必要があるが、常時接続するためには追加の設定を要すため紹介する。

Windwos11 22H2ではVPN接続時の項目に「自動的に接続」チェックボックスがあるが、チェックを入れ接続後、PCを再起動すると接続されていない状態であった。(2023/03/13 確認)

この方法ではPC再起動後、自動的に接続はできなかった。

●自動接続については下記の方法を試した。システム構築時にも色々試してうまく行かなかったが、今回、再度実験を行った結果、うまく設定できたので、気をつける点を記す。やり方については下記リンク参照のこと。

[VPN] Windows起動時に自動接続する - プロプログラマ -Flex,Air,C#,Oracle,HTML5+JS-

とても簡単にVPN接続する方法

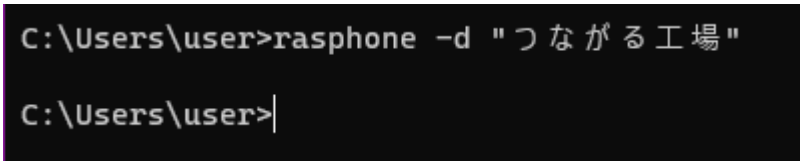
まず、下記ファイルをテキストエディタで書き換え、コマンド実行時に余計な画面が表示されないように設定するが、VPN接続が複数設定されていると、「PreviewUserPw=0」項目が複数あることに前回は気づかなかった。VPN接続設定1つごとに約150行の設定があるため縦に長く、接続名ごとに区切られていることに気づけなかった。

%APPDATA%\Microsoft\Network\Connections\Pbk\rasphone.pbk

["VPN接続名"] の書式で設定が区切られているので、自動接続したいVPN接続名の「PreviewUserPw=0」を書き換えると良い。

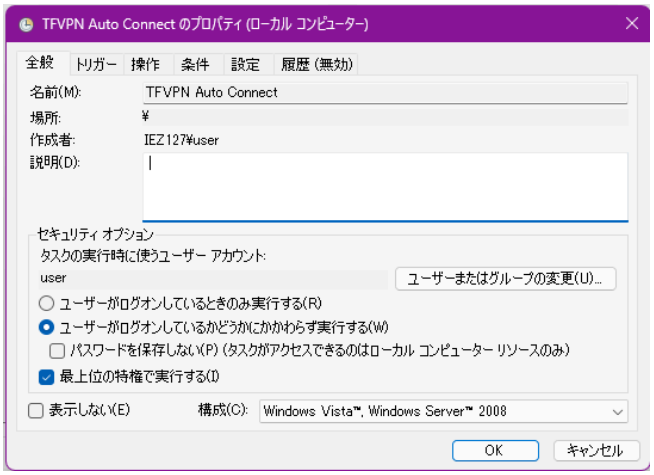
タスクスケジューラーで自動化する前に、下記コマンドでVPN接続が確立できるか確認すると良い。このコマンドでVPN接続ウィンドウが表示され、クリックしないと先に進まない場合は「PreviewUserPw=0」やパスワードの記憶等の設定を見直してほしい。

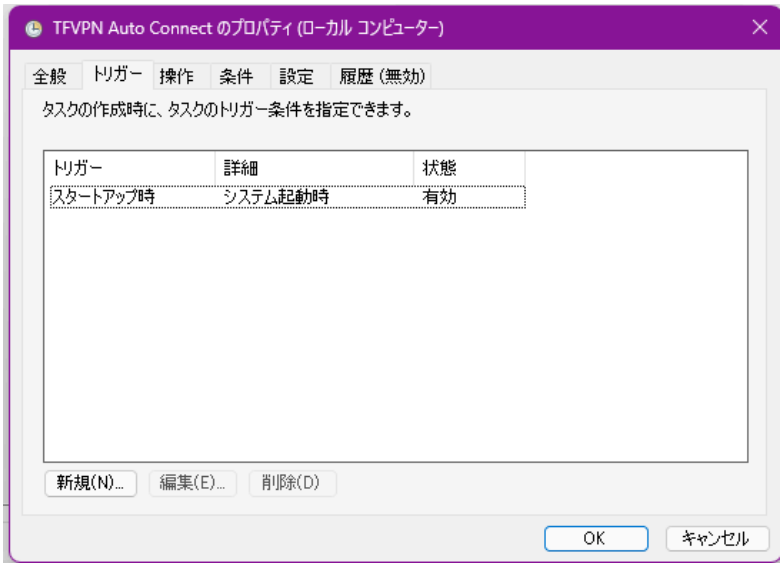
rasphone -d "「VPN接続名」"

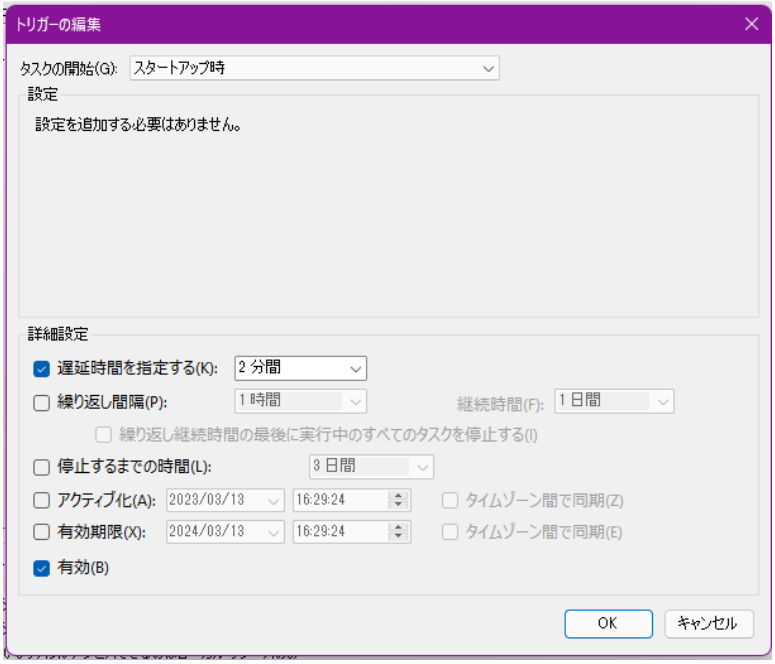


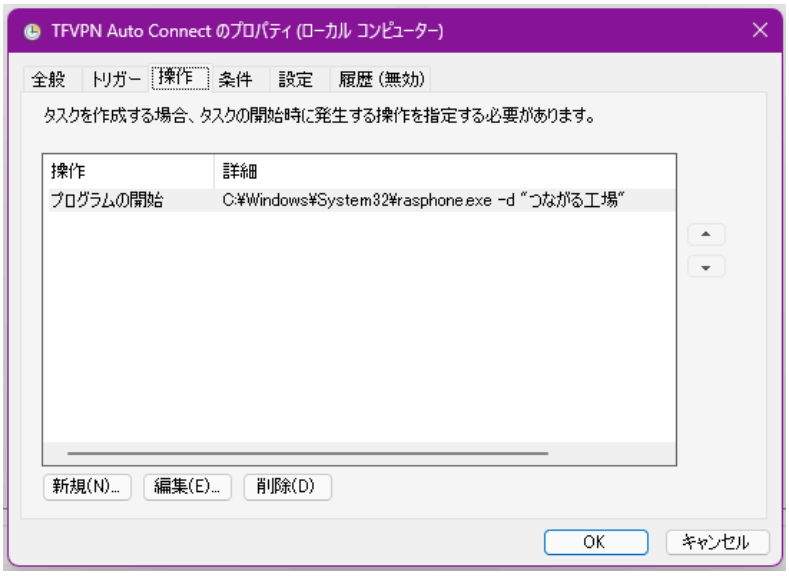
次にタスクスケジューラーに設定を下図のように設定することで、PC再起動時に自動的にVPN接続可能であることがわかった。

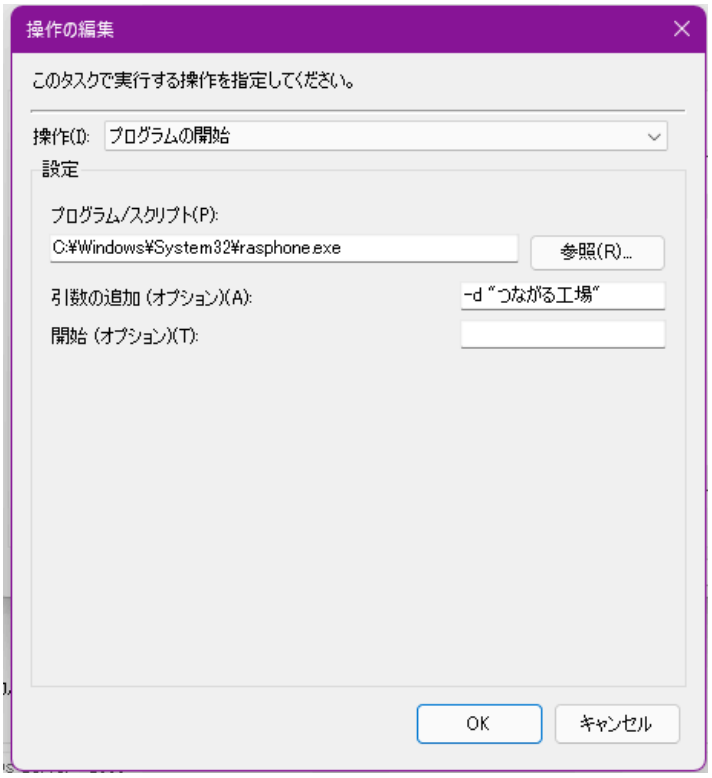
※ただし、VPNが切断された場合やWiFiが切断された場合のリダイヤル設定については別の設定が必要である。


TFVPN Auto Connect のプロパティ (ローカル コンピューター)
全般
トリガー
操作
条件
設定
履歴 (無効)
名前(M):
TFVPN Auto Connect
場所:
¥
作成者:
IEZ127¥user
説明(D):
セキュリティ オプション
タスクの実行時に使うユーザー アカウント:
user
ユーザーまたはグループの変更(U)...
ユーザーがログオンしているときのみ実行する(R)
ユーザーがログオンしているかどうかにかかわらず実行する(W)
パスワードを保存しない(P) (タスクがアクセスできるのはローカル コンピューター リソースのみ)
最上位の特権で実行する(I)
表示しない(E)
構成(C):
Windows Vista™, Windows Server™ 2008
OK
キャンセル


TFVPN Auto Connect のプロパティ (ローカル コンピューター)
全般
トリガー
操作
条件
設定
履歴 (無効)
タスクの作成時に、タスクのトリガー条件を指定できます。
トリガー
詳細
状態
スタートアップ時
システム起動時
有効
新規(N)...
編集(E)...
削除(D)
OK
キャンセル

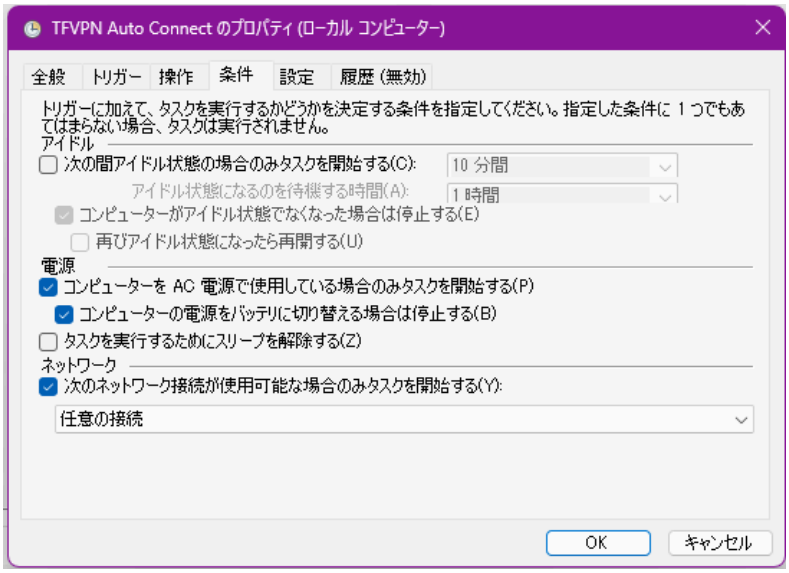

トリガーの編集
タスクの開始(G):
スタートアップ時
設定
設定を追加する必要はありません。
詳細設定
遅延時間を指定する(K):
2 分間
繰り返し間隔(P):
1 時間
継続時間(F):
1 日間
繰り返し継続時間の最後に実行中のすべてのタスクを停止する(I)
停止するまでの時間(L):
3 日間
アクティブ化(A):
2023/03/13
16:29:24
タイムゾーン間で同期(Z)
有効期限(X):
2024/03/13
16:29:24
タイムゾーン間で同期(E)
有効(B)
OK
キャンセル

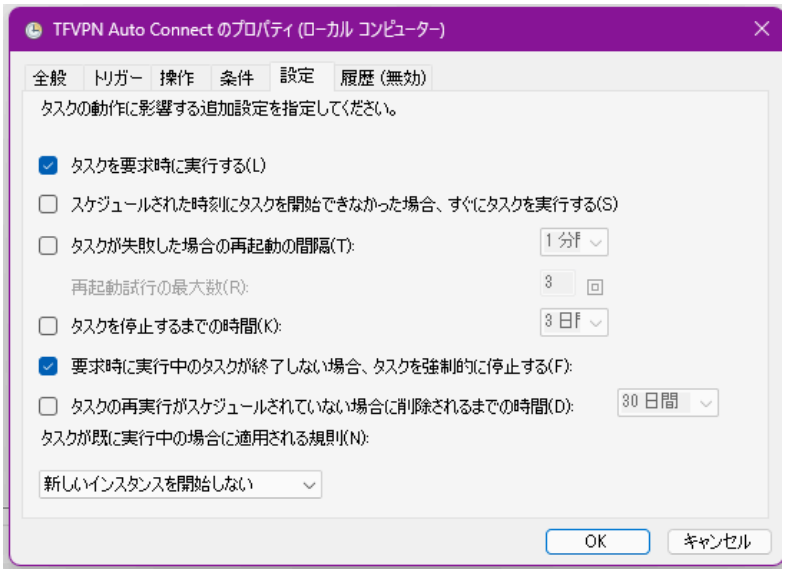




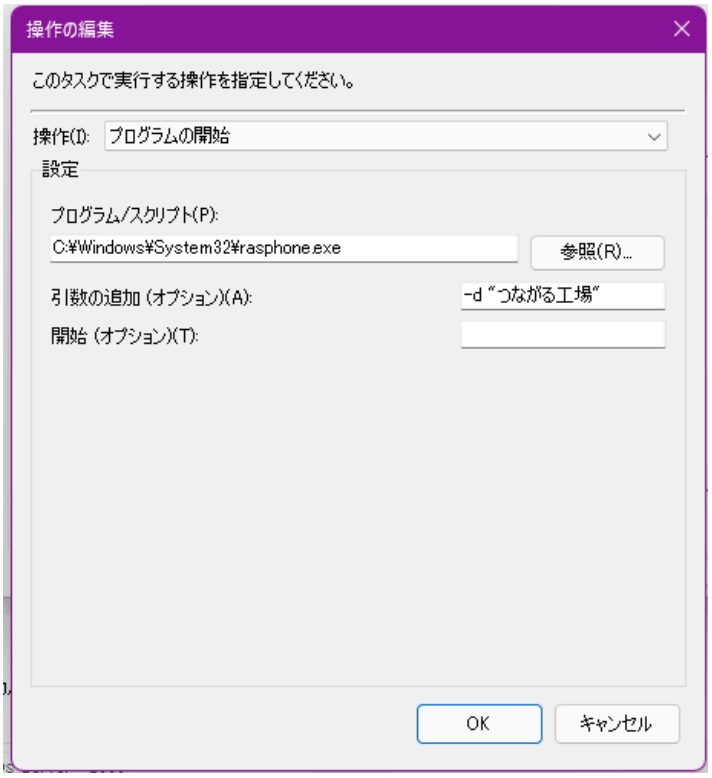
上図のプログラム/スクリプト欄に入力する文字列を間違えたこともあった。

正　C:\Windows\System32\rasphone.exe





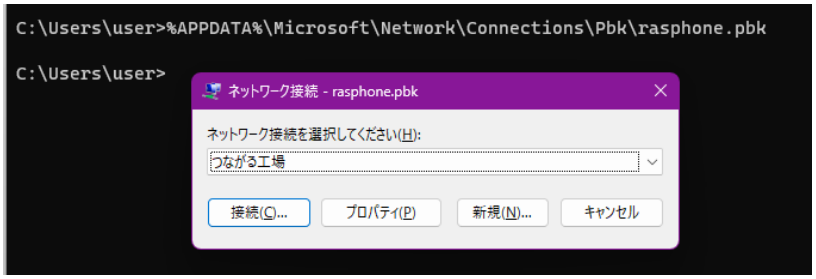
ーーーー以下　失敗談の捕捉ーーーー



上図のプログラム/スクリプト欄に入力する文字列を間違えたこともあった。

正　C:\Windows\System32\rasphone.exe

誤 rasphone.pbk　へのパス。「テキストエディタで編集するファイル」と「実行ファイル」は異なるが、rasphone.pbkをダブルクリック等(下図)で実行してもVPN接続するための画面が表示されるため、間違いに気づけなかった。



●上記タスクスケジューラーでのVPN自動接続中にWiFi接続中のWiFiルーターの電源を切ったところ、WiFiは別のルーターに自動的に再接続されたが、VPNは切断状態のままであったため、VPN切断時には再接続するような設定が必要である。

8.2.　M5シリーズでAD変換を行うときの留意点

M5Stackの内蔵AD変換器を用いてアナログ電圧を取得した際に、誤差が大きくなる事例があった。安定した測定を行うために、内蔵ADCについて調査した。

〇実験1

M5Stackの内蔵AD変換器とマイクロチップ社のAD変換IC（MCP3208）を使いそれぞれ電圧を計測し比較した。供給電圧は安定化電源を用いて、0.15～3.3Vまで0.05V刻みで計測。

・計測は各電圧に対して10回実施し平均値を求めた。

・AD変換で得られた12bitの値（0～4,095）を「AD変換値」とする。

実験結果を図8-1に示す。内蔵ADCは2.5Vを超えた辺りから線形性が悪くなることを確認した。

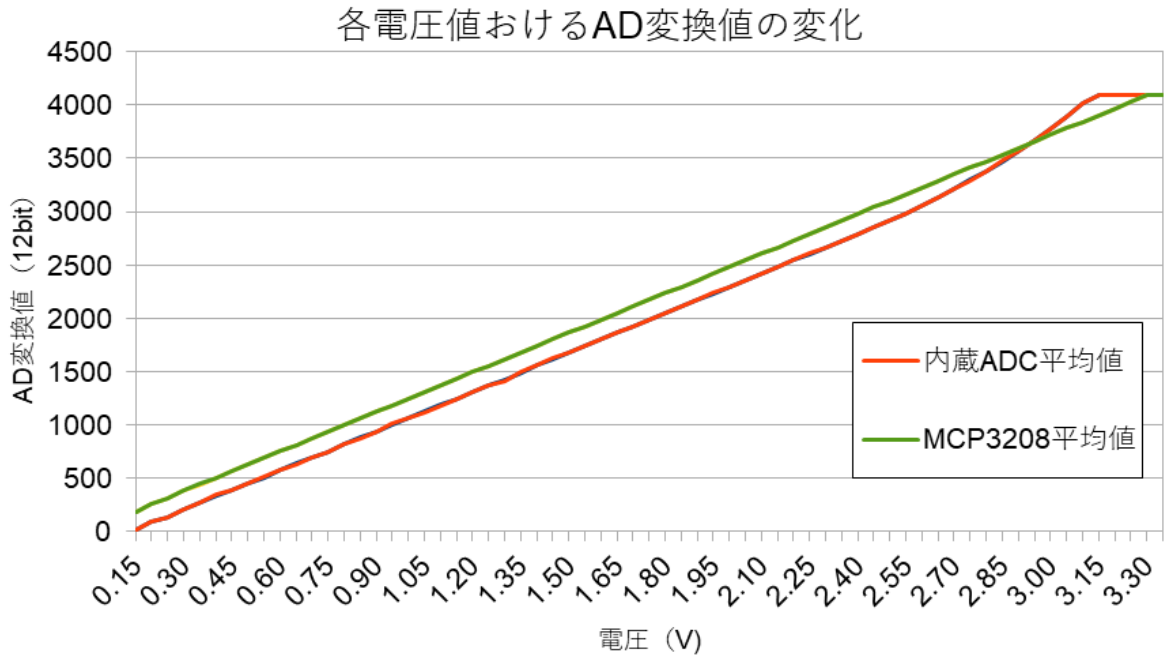


図 8-1　M5StackとMCP3208のAD変換器の誤差について

〇実験2

M5Stackの測定におけるばらつきについて調査するため、エッジデバイスとして同様によく使われるArduinoとの比較を行った。それぞれ内蔵AD変換器とマイクロチップ社のAD変換IC（MCP3204）を使いそれぞれ電圧を計測し比較した。供給電圧は安定化電源を用いて、0.0～3.3Vまで0.1V刻みで計測を行った。

・計測は各電圧に対して10回実施し最小、最大、平均値を求めた。

・AD変換器の分解能はArduinoの内蔵ADCは10bit、M5StackおよびMCP3204は12bitである。そのためAD変換で得られた値を電圧値に変換して比較している。

実験結果を図8-2に示す。M5Stackは内蔵ADC、MCP3204いずれを用いた場合もArduinoを用いた場合と比較して、ばらつきが大きいことを確認した。特に、内蔵ADCについては最大値と最小値の差が最大でで0.47Vと大きかった。

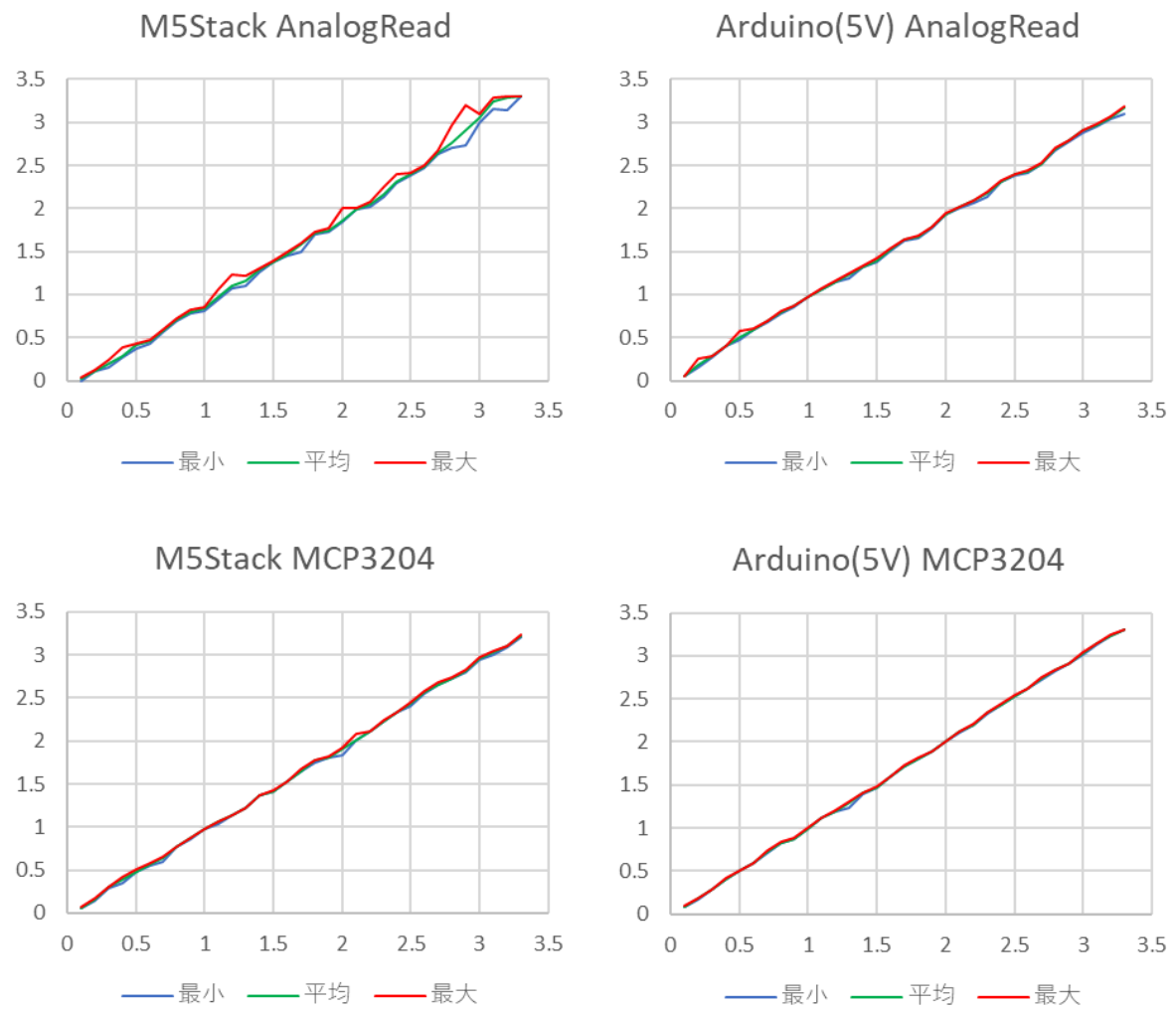


図 8-2　各電圧における誤差

〇結果、考察

・M5Stack内蔵のAD変換器を用いる場合、2.5V以上は線形性が悪くなるため、センサの出力値について留意する必要がある。

・M5Stackを用いたAD変換を行う場合、ばらつきが大きくなる傾向がある。特に内蔵ADCを使う場合はばらつきが大きくなるため、複数回測定し平均値や中央値をとるなどの対策を推奨する。

・M5Stackの測定がばらつく原因としては、電源ノイズの影響を受けやすい可能性が考えられる。AD変換以外の処理においてもM5Stackの動作が不安定になる事例を確認している。これの対策として安定化電源を用いることで一部改善した事例もあるため、ノイズ対策が有効である可能性がある。

8.3. UIFLOWプログラミング固定IPアドレス設定できない問題

青森県の振動試験機は八戸市の八戸工業研究所に導入されており、システム構築時にIoTデータ収集方法としてプリペイドSIMとLTE-USBドングル（ピクセラ社PIX-MT100）を使用していたが、設置した建物のLTE電波状況が悪いため、社内ネットワークにIoTセンサを接続するよう変更を行った。社内ネットワークは手動IP設定でPC等を接続しているため、M5Stack用にUIFlowで作成したプログラムを動作させる際にIPアドレスを取得できず、動作できない状態が発生した。また、無線LAN装置もメッシュWiFi対応のTPLink社製DecoX90を複数台導入したが、こちらもIPアドレスの手動設定ができなかった。そのためWindows10PCにOpenDHCPサーバソフトウェアをインストールし、設定を行った。OpenDHCPサーバについては下記サイトを参考にされたし。

https://qiita.com/tmiki/items/d5965ee767378cf7f13f

OpenDHCPサーバHP

https://dhcpserver.sourceforge.net/

DHCPサーバはMACアドレスに対して個別にIPアドレスを配布するよう設定を行った。設定ファイルに登録されていないMACアドレスにはIPアドレスを配布しないよう設定を行った。

8.4. GM3環境センサ落とし穴

センサノードからLANへセンサ値を伝送する場合、LAN接続可能なワンボードコンピュータやマイコンなどが考えられる。通信手段として挙げられるのは、Wi-FiやBluetoothなどであるが、消費電力が大きいことや接続が複雑になることが課題である。これらの課題を回避する方法の1つに、920MHz帯を用いた無線によるデータ伝送がある。
今回利用したGM3は、920MHz帯の送受信機を利用し、センサノードが小型低消費電力化され電池駆動が可能となっている。
GM3は、920MHzの送受信機を利用し、センサ側に送信機を組込み、有線LANに接続できるLANTRONIX社のXPORTにシリアル接続で受信機が組み込まれている。

詳しいセットアップ方法はMZプラットフォームの「MZ Platform IoT Toolkit (スマート製造ツールキット/MZ-EX/IoT化用コンテンツ集)」内の「MZIoTtoolkit説明資料(市販機器).pdf」を参考のこと。

もしトラブルがあった場合、下記方法でXPORTの設定を試してほしい。

8.4.1. XPORTとの接続

XPORTは、httpサーバを備えており、PCからXPORTにブラウザでhttp接続することで、動作設定を行うことができる。また、Telnet接続で設定条件の吸い上げも可能である。

XPORTの初期状態はLANに接続するとDHCPサーバからIPアドレスを付与されて稼働するようになっている。そのため、XPORTのIPアドレスは簡単には分からない。そこで、付録のプログラムを動かすか、次のpythonスクリプトを動かすことにより、IPアドレスを知る事が可能である。

```
#!python3
import socket
from contextlib import closing
import struct
import time

myself = '192.168.2.3'
broadcastIP = '255.255.255.255'

portDst = 30718
portSrs = 28672
bufsize = 4096
timeoutsec = 5.0

mysocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
mysocket.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)
mysocket.bind((myself, portSrs))
mysocket.settimeout(timeoutsec)
with closing(mysocket):
    for k in range(0,1):
        datTx = 0xf6000000
        byteTxBroad = struct.pack('I',datTx)
        mysocket.sendto(byteTxBroad, (broadcastIP, portDst))
        buf, addr = mysocket.recvfrom(bufsize)
        print('addr:', addr)
        print('buf:', buf)
        time.sleep(5)
```

8.4.2. XPORTの設定

IPアドレスが判明したら、webブラウザにて接続する。ここで重要なのが、ブラウザの種類である。
FireFoxでは設定できない。google chromeブラウザであれば設定が可能である。
設定の可否は、設定ボタンを押下した際に、404エラーが出るか否かで分かる。
ここで、最も重要なのは、工場設定に初期化をするボタンを押下しない点である。
もし、誤って初期化した場合は、BAUDレートが9600bpsとなっているのでそれを115200bpsに設定し直すことで、正常に戻すことが可能である。

8.4.3. おまけ：正体不明のネットワーク接続型機器の解析

正体不明のネットワーク接続型機器を解析する場合は、閉鎖的なLANを構築し、その中に流れるパケットを解析することで、その機器の理解が進む場合がある。

DHCPサーバおよびパケットキャプチャを行うデバイスとしてLANポートが内蔵されたRaspberry pi1B以降（以後ラズパイと記す）を用い、対象とする機器とLANケーブルで接続する。
ラズパイには、次をインストールする。

```
apt install isc-dhcp-server tshark lrzsz telnet
```

/etc/dhcp/dhcpd.confに追加

```
authoritative;
log-facility local7;
subnet 192.168.2.0 netmask 255.255.255.0 {
  range 192.168.2.4 192.168.2.254;
  default-lease-time 600;
  max-lease-time 7200;
  option routers 192.168.2.3;
  option domain-name "local";
  option domain-name-servers 192.168.2.3;
  option subnet-mask 255.255.255.0;
  option broadcast-address 192.168.2.255;
}
```

/etc/default/isc-dhcp-server にて次に変更

```
INTERFACESv4="eth0"
```

/etc/init.d/isc-dhcp-server にて 実行ウエイトを追加

```
sleep 10
```

dhcpサーバを再起動

```
systemctl restart isc-dhcp-server
```

接続機器に払い出したIPアドレスが記載されたファイル

```
/var/lib/dhcp/dhcpd.leases
```

パケットキャプチャ

```
tshark -i eth0 -w filrname.pcapng
```

(取得状況を見ながらctrl + cで止める。10秒程度。)
取得したパケットキャプチャデータの転送
- シリアル経由のzmodemを用いる
- もしくはwifiを接続しscpを用いる

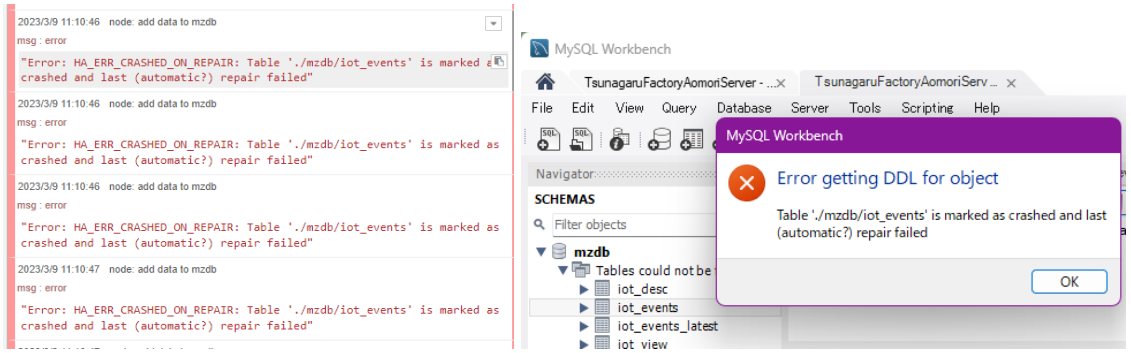
解析はwiresharkがインストールされているPCで行うと良い。
`DHCP 342 DHCP Request`とあれば、それがIPアドレスの払い出しを行っている様子である。
その他に、接続機器がパケットを投げているかどうかを確認する。

8.5.　UIFLOWでのフリーズ対策

M5Stackがフリーズし、センサデータが途切れることが多々ある。WiFiの接続に失敗や接続のロストでフリーズしている場合が多いように感じる。フリーズした際にはウォッチドッグタイマでリセットするプログラムの追加が必要である。

8.6.　MariaDBのテーブル破損

Node-RedやMySqlWorkbenchで下記エラーが表示され、テーブルが破損した。MySQLの操作になれていないため、まだ復旧はできていない。



下記コマンドを実行したが、1時間30分ほど応答なし。

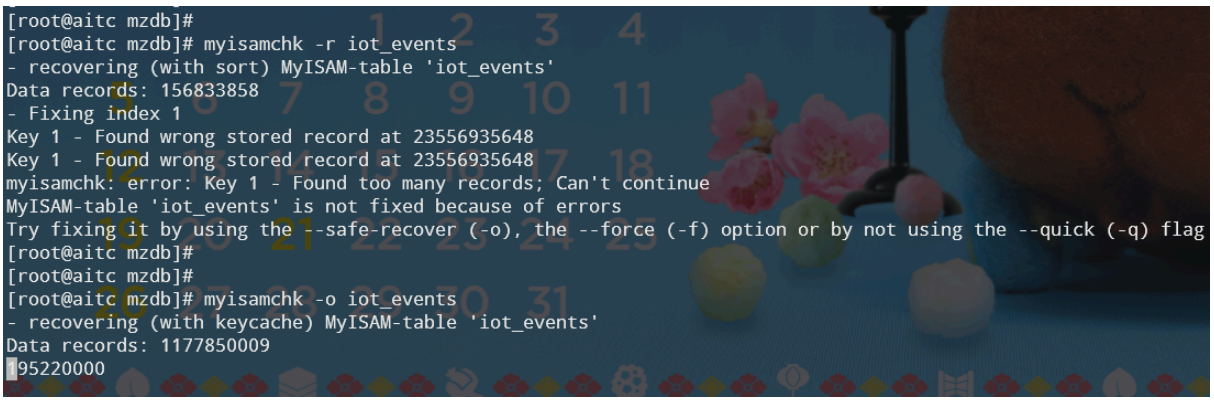
```
repair table iot_events;
```

次に下記を実施。

[MySQL table is marked as crashed and last (automatic?) repair failed - Stack Overflow]

myisamchk -o ‘iot_events’

16時間ほど実行したが、翌朝ターミナルが落ちていて失敗。



HA_ERR_CRASHED_ON_REPAIRのエラー内容はリペア中にインデックスファイルが壊れたものと思われる。

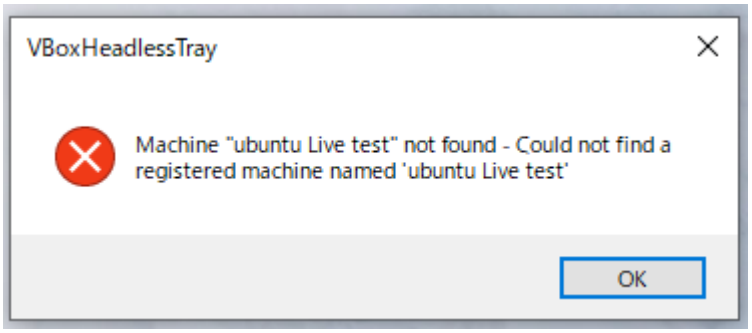
エラー情報：MySQL: include/my_base.h File Reference

iot_eventsテーブルが約22GBと巨大なことからか、「myisamchk -r」は対応するレコード数の制限のため実行途中で失敗し、「myisamchk -o」は実行から約16時間が経過しても完了しなかった。

後日、mariaDBを停止後に「myisamchk -o iot_events」を再実行し、約2週間ほど経過後にmariaDBを起動した。iot_events tableを確認したがエラーは解消されていなかった。
以上のことから「myisamchk」コマンドによる当該tableの復旧は困難だと思われる。

8.7. データ中継用仮想マシンの自動起動時のエラー

WindowsPCにVirtualBox仮想マシンとして構築したUbuntuサーバで、Windows起動時にVBoxHeadlessソフトを用いて自動起動していたが、ある時から下記エラーが表示されるようになった。おそらく仮想マシンのバックアップ等メンテナンスを行った際に仮想マシン名の変更が原因と思われる。

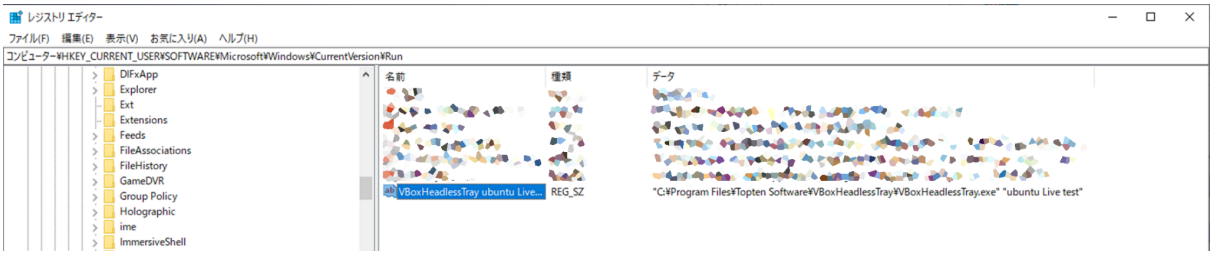


Windowsの起動時にVBoxHeadlessの自動起動設定をどのようにしたか覚えておらず再度調査した。サービスやタスクスケジューラー、スタートアップフォルダにはショートカット等登録されていなかった。

WIndowsの自動起動するソフトがあり、どこから起動しているのか分からない場合、下記サイトが参考となった。

Windows 10 の自動起動の設定は5つの場所を要チェック

下図のようにレジストリからの自動起動に設定が残っており、これを削除することで上記エラーは出なくなり、解決した。VBoxHeadlessはインストールを行い、仮想マシンを起動し、そのままWindowsを再起動すると、Windows起動時に仮想マシンを起動するか尋ねられるダイアログが表示され、OKすると、以後Windows起動時に仮想マシンも自動起動される。



9. 付録2：簡易システム構築例

9.1. M5GOの取得データをGrafanaで可視化

青森県産業技術センター工業総合研究所では、令和2年度より研究所独自のIoT技術研修会を実施している。本研修ではM5GO（M5Stackの一種）で取得したデータをラズベリーパイ（シングルボードコンピュータの一種）上のGrafanaで可視化するまでをハンズオン形式で実施している。上述のつながる工場システムの簡易版として、以下に本研修の操作内容を動画で示す。

9.2. ラズベリーパイのセットアップ

簡易システムに用いるラズベリーパイのセットアップは以下を参考とされたい。

IoT開発環境2020年度版ラズパイ用設定 - Qiita

2021年度IoT研修用ラズパイの設定 - Qiita

9.3. M5StickCを用いたロボットアームの動作見える化

温湿度測定のために用いたM5StickCの本体には加速度センサ（MPU6886）も内蔵されている。ここでは参考に振動測定による装置稼働状況の可視化の例を紹介する。

図 9-1 に小型ロボットアームの吸引ポンプの振動を測定した結果を示す。このロボットアームは吸引ポンプによって空気を吸うことで、アーム先端でキューブをつかむことができる。M5StickCの画面にはFFT解析された振動測定結果が表示され、アームの先でキューブを掴んでいるときと、掴んでいないときとで周波数のピークが異なることがわかる。

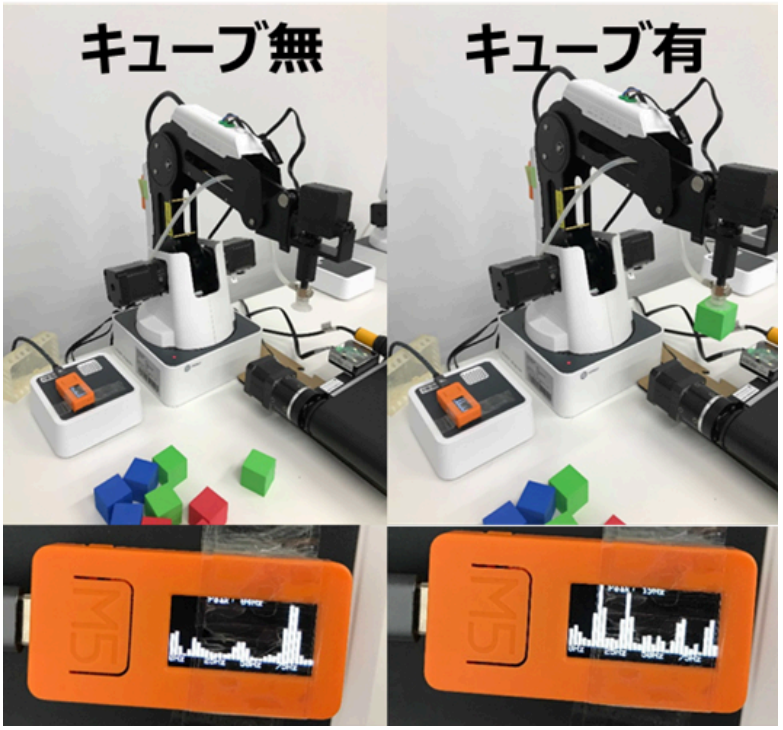


図 9-1 M5StickCによるロボットアームの振動測定

得られたデータはM5StickCからWiFIによってサーバに送信され、他のデバイスからアクセスすることで遠隔地でも確認可能とした。図 9-2 に周波数ピークの違いによってアームの状態を分類した結果と、子機(M5StickC)とは別のM5Stackによって分類結果(アームの状態)を表示している画面を示す。ここではアームを掴んでいる状態を「Grab」と表示している。

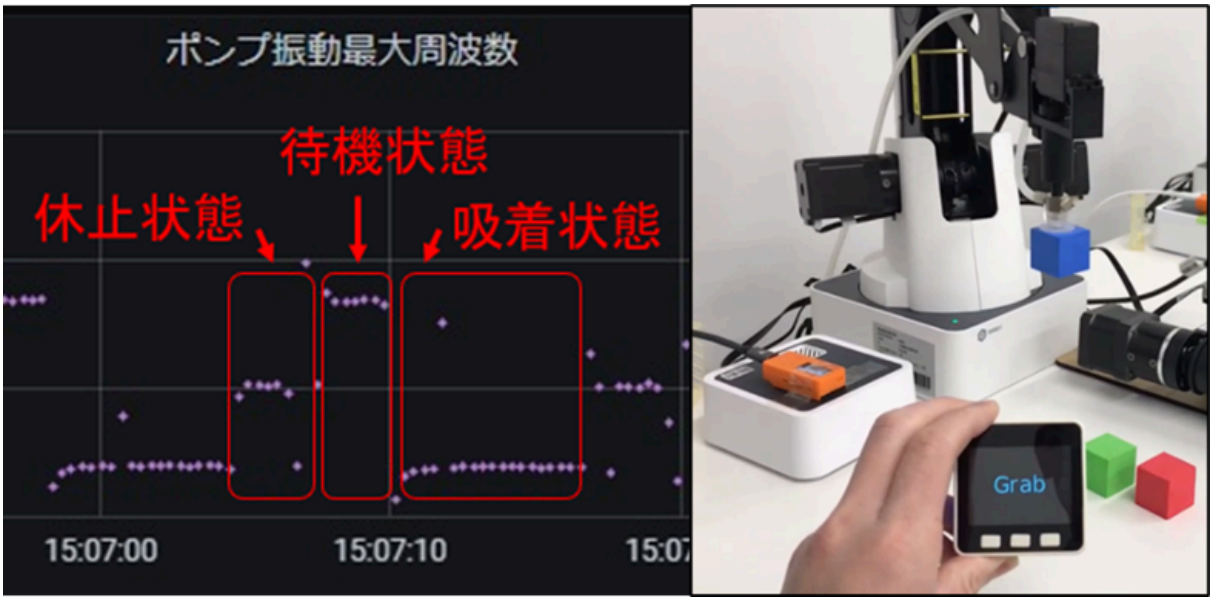


図 9-2 アーム状態の分類と状態表示画面

振動データを取得し、FFT解析をした結果、ピークの発生する周波数をUDP方式で送信するM5StickCのプログラムはArduino-IDEで作成した。以下からダウンロード可能である。

https://drive.google.com/file/d/1nnFesCfZxxyo3OfgvL4lsnExUGzR7_Jv/view?usp=share_link

受信したピーク周波数の値（ここでは”f”）によって、ポンプが停止している「休止」、ポンプが動いているがキューブはつかんでいない「待機」、ポンプ稼働によってキューブをつかんでいる「吸着」の3つの状態を表す数値を返す処理を、Node-Redのfunctionノードで行った。図 9-3 にその部分の記載を示す。なおコメントアウトされている部分は、UDPで受信したデータのなかから目的の子機(ここではM5StickC)を抽出するために設けた上位ノードの処理を転載したものであり、受信した文字列を「;」区切りで分割し、Arduino-IDE上で”HOSTNAME”とした部分を抽出している。返された数値(0,1,2)は下位のノードでInfluxDBに登録している。

図 9-3 アーム状態の分類と状態表示画面

上述の処理によってInfluxDBに格納されたデータを参照し、画面にその結果を表示するM5Stackのプログラムは UI-FLOWで作成した。図 9-4 に示す。

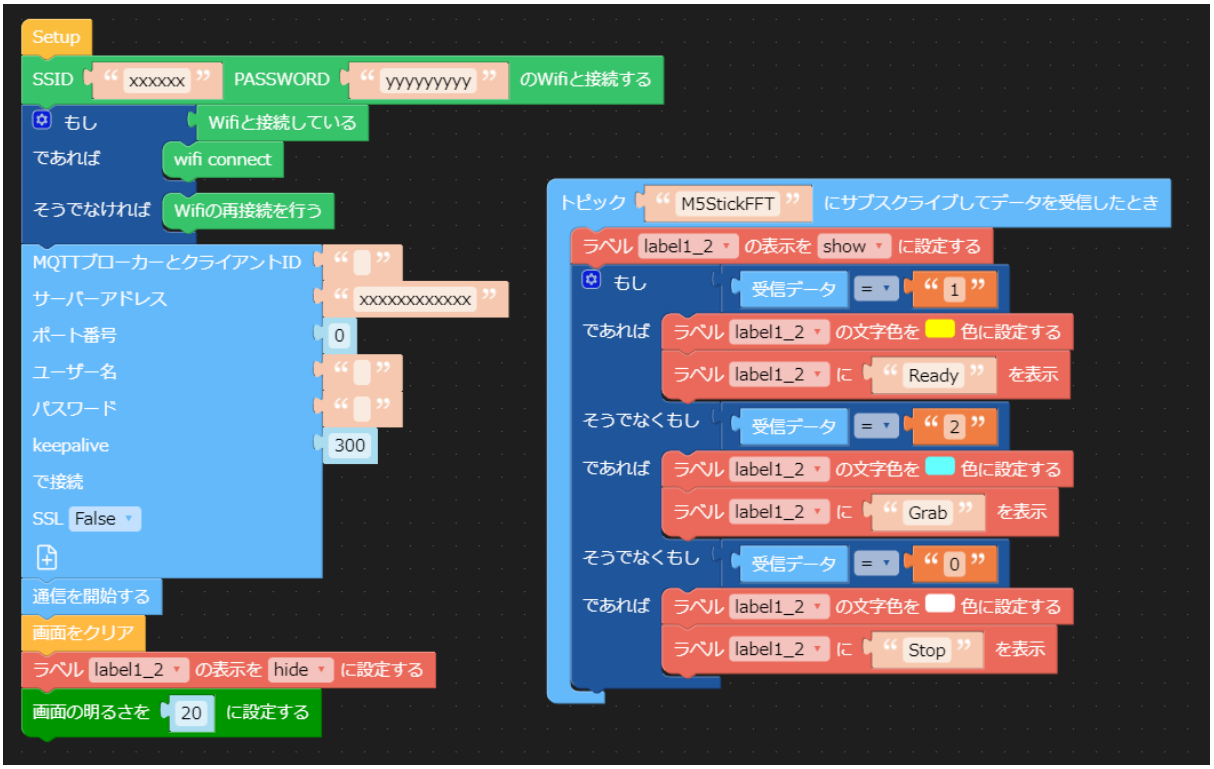


図 9-4　稼働状態表示プログラム

なお、機器の状態監視は、その対象によって取得すべきパラメータが変わるため、ロボットアームの状態監視に振動測定を限定して推奨するものではない。

●付録 小ネタ　Windowsリモートデスクトップ(RDP)を縦長などカスタム画面サイズで接続

4Kディスプレイを半分に分割してRDPしたいときに便利である。

リモートデスクトップ接続の画面サイズを変更する：Tech TIPS - @IT

```
3 desktopwidth:i:1920
4 desktopheight:i:2103
```

10. マニュアルの維持管理

10.1.　マニュアルの取り扱い・公開条件等について

IoT技術の初学者向けにWEB公開し、本研究会が維持管理する。

問い合わせ先：青森県産業技術センター工業総合研究所電子情報技術部（研究会事務局）

[お問い合わせフォーム（工業総合研究所）| 地方独立行政法人 青森県産業技術センター]

10.2.　Qiitaの活用

IT関連の技術革新は、他の分野に比べて非常に早く、この変化は情報を書籍化する期間よりも早いことが多く、書籍は豊富とはいえない。そして、メーカがwebに上げた技術情報は専門的であることが多く、一人で知識修得するのは時間がかかることが少なくない。この問題に対する解決策の1つとなりえるのがコミニティ(有志による共同体)が発信する技術系ブログサイトである。

2022年現在、日本の主要な技術系ブログサイトは、Qiita、Zenn、はてなブログ などがある。これらは、登録も閲覧も無料であり、googleなどのweb検索によっても記事を見つけることができる。

例えば、マイコンで家電の電力測定する回路を構築したい場合、「電力 計測」というキーワードで検索しても、目的に合った結果が得られないことが多い。そこで、キーワードに工夫が必要となる。もし、具体的なキーワードを知っていれば「電力 測定 マイコン クランプ式」を検索することで、目的に近いものになる。

ここで便利なのが、技術系ブログサイト名を検索キーワードに入れる方法である。例えば、「電力 計測 qiita」を検索すると、結果の殆どがQiitaになるものの、Qiitaは開発を行う有志によるブログサイトなので、製作方法についての記事が得られることが多い。また、技術系ブログ内には、「やってみた」という単語が多い。それを利用して、「電力 計測 やってみた」という検索方法も有効である。

なお、検索サイトを利用しなくとも、技術系ブログサイトを開き、そこで検索するのも手っ取り早い方法である。

図 10.2.1 電力と計測をキーワードに検索した例

11. 謝辞

本事業は国立研究開発法人 産業技術総合研究所インダストリアルCPS研究センター 様の支援を得て「つながる工場テストベッド事業」により実施しました。関係者各位へ深く感謝申し上げます。