

# Armadillo-IoT ゲートウェイ A6E Node-RED 開発ガイド

Version 1.0.1  
2023/12/26

株式会社アットマークテクノ [<https://www.atmark-techno.com>]

Armadillo サイト [<https://armadillo.atmark-techno.com>]

---

# Armadillo-IoT ゲートウェイ A6E Node-RED 開発ガイド

株式会社アットマークテクノ

製作著作 © 2023 Atmark Techno, Inc.

Version 1.0.1  
2023/12/26

# 目次

- 1. はじめに ..... 7
  - 1.1. Node-RED について ..... 7
  - 1.2. 本書について ..... 7
    - 1.2.1. 本書で扱うこと ..... 7
    - 1.2.2. 本書で扱わないこと ..... 7
    - 1.2.3. 本書で必要となる知識と想定する読者 ..... 7
    - 1.2.4. フォント ..... 8
    - 1.2.5. コマンド入力例 ..... 8
    - 1.2.6. アイコン ..... 8
    - 1.2.7. ユーザー限定コンテンツ ..... 9
    - 1.2.8. 本書および関連ファイルのバージョンについて ..... 9
- 2. ユーザー登録 ..... 10
  - 2.1. 購入製品登録 ..... 10
- 3. Armadillo のセットアップ ..... 11
  - 3.1. 作業の前に ..... 11
    - 3.1.1. 開発セット内容物の確認 ..... 11
    - 3.1.2. 開発に必要なもの ..... 11
    - 3.1.3. 接続方法 ..... 11
    - 3.1.4. 起動デバイス設定スイッチについて ..... 13
  - 3.2. インストールディスクを使用してセットアップを行う ..... 13
    - 3.2.1. Windows PC 上でインストールディスクを作成する ..... 14
    - 3.2.2. Linux PC 上でインストールディスクを作成する ..... 15
    - 3.2.3. 作成したインストールディスクを使用してセットアップする ..... 15
  - 3.3. インターフェースレイアウト ..... 16
  - 3.4. ケースの組み立てと分解方法 ..... 18
    - 3.4.1. ケースの組み立て手順 ..... 21
    - 3.4.2. ケースの分解 ..... 21
  - 3.5. microSD カードの挿抜方法 ..... 23
  - 3.6. ネットワークに接続する ..... 26
    - 3.6.1. Armadillo の有線 LAN に固定 IP アドレスを設定する ..... 29
  - 3.7. Node-RED に接続する ..... 31
- 4. 開発を行う ..... 32
  - 4.1. Node-RED に接続する ..... 32
  - 4.2. Node-RED コンテナのログを表示する ..... 33
  - 4.3. Node-RED で利用可能なノードの一覧 ..... 33
  - 4.4. フローを作成する ..... 34
    - 4.4.1. LED を制御する ..... 34
    - 4.4.2. CPU の測定温度を取得する ..... 38
    - 4.4.3. 接点入力の入力レベルを取得する ..... 40
    - 4.4.4. 接点出力の出力レベルを制御する ..... 45
    - 4.4.5. 接点入力の入力レベルの立ち上がりエッジを検出する ..... 50
    - 4.4.6. RS-485 modbus RTU 読み出しを行う ..... 59
    - 4.4.7. CPU の測定温度のグラフをダッシュボードに表示する ..... 62
    - 4.4.8. Node-RED を終了する ..... 71
  - 4.5. ユーザーデータを削除する ..... 74

# 目次

3.1. 初回起動時の Node-RED 画面 .....	11
3.2. Armadillo-IoT ゲートウェイ A6E と周辺デバイスの接続例 .....	12
3.3. 起動デバイス設定スイッチの操作 .....	13
3.4. Win32 Disk Imager Renewal 設定画面 .....	15
3.5. Armadillo-IoT ゲートウェイ A6E インターフェースレイアウト .....	17
3.6. ケースモデル展開図 .....	19
3.7. フック取り付け 1 .....	20
3.8. フック取り付け 2 .....	21
3.9. フックのツメ .....	22
3.10. ケースボトム部のツメ .....	23
3.11. カバーのツメ .....	23
3.12. カバーのロックを解除する .....	24
3.13. カバーを開ける .....	24
3.14. microSD カードの挿抜 .....	24
3.15. カードマークの確認 .....	25
3.16. カバーを閉める .....	25
3.17. カバーをロックする .....	25
3.18. パスワード登録画面 .....	27
3.19. パスワード登録完了画面 .....	28
3.20. ログイン画面 .....	28
3.21. 状態一覧を選択 .....	29
3.22. LAN 情報 .....	29
3.23. 初回起動時の Node-RED 画面 .....	31
4.1. 初回起動時の Node-RED 画面 .....	32
4.2. Node-RED の画面領域 .....	33
4.3. [inject] ノードのプロパティ内容 .....	35
4.4. [trigger] ノードのプロパティ内容 .....	36
4.5. [write file] ノードのプロパティ内容 .....	37
4.6. LED を 1 秒間隔で点滅するフロー .....	37
4.7. [inject] ノードのプロパティ内容 .....	38
4.8. [read file] ノードのプロパティ内容 .....	39
4.9. [function] ノードのプロパティ内容 .....	40
4.10. CPU の測定温度を 1 秒間隔で取得するフロー .....	40
4.11. 接点入力の接続方法(DI1 ピン) .....	42
4.12. [inject] ノードのプロパティ内容 .....	43
4.13. [GPIO in] ノードのプロパティ内容 .....	44
4.14. 接点入力の入力レベルを 1 秒間隔で取得するフロー .....	44
4.15. 接点出力の接続方法(制御対象がプラスコモンの場合の DO1 ピン) .....	46
4.16. 接点出力の接続方法(制御対象がマイナスコモンの場合の DO1 ピン) .....	46
4.17. [inject] ノードのプロパティ内容 .....	47
4.18. [trigger] ノードのプロパティ内容 .....	48
4.19. [GPIO out] ノードのプロパティ内容 .....	49
4.20. 接点出力の出力レベルを 1 秒毎に HIGH レベルと LOW レベルを切り替えるフロー .....	49
4.21. DI2 ピンの立ち上がりエッジを取得するための接続図 .....	51
4.22. [inject] ノードのプロパティ内容 .....	52
4.23. [GPIO in] ノードのプロパティ内容 .....	53
4.24. [rising edge] ノードのプロパティ内容 .....	54
4.25. [change] ノードのプロパティ内容 .....	55
4.26. [inject] ノードのプロパティ内容 .....	56
4.27. [inject] ノードのプロパティ内容 .....	57

---

4.28. [GPIO out] ノードのプロパティ内容 .....	58
4.29. 接点入力の入力レベルの立ち上がりエッジを検出するフローとデバッグメッセージ .....	58
4.30. スイッチの状態と終端抵抗の ON/OFF .....	59
4.31. RS-485 通信対応デバイスとの接続方法 .....	60
4.32. [modbus-client] ノードのプロパティ内容 .....	61
4.33. [Modbus-Read] ノードのプロパティ内容 .....	62
4.34. RS-485 を使用した Modbus RTU 読み出し用フロー .....	62
4.35. [ダッシュボード]を選択する .....	63
4.36. ダッシュボード編集画面 .....	63
4.37. ダッシュボードに [Tab1] を追加 .....	64
4.38. [Tab1] プロパティ内容 .....	65
4.39. ダッシュボード編集画面に [Armadillo] タブが追加 .....	66
4.40. ダッシュボード編集画面に [Group1] グループが追加 .....	67
4.41. [Group1] プロパティ内容 .....	68
4.42. ダッシュボード編集画面に [chart] ノードが追加 .....	69
4.43. [inject] ノードのプロパティ内容 .....	70
4.44. CPU の測定温度のグラフをダッシュボードに表示するフロー .....	70
4.45. CPU の測定温度のグラフのダッシュボード .....	71
4.46. [inject] ノードのプロパティ内容 .....	72
4.47. [exit] ノードのプロパティ内容 .....	73
4.48. Node-RED を任意のタイミングで終了するフロー .....	73

# 表目次

- 1.1. 使用しているフォント ..... 8
- 1.2. 表示プロンプトと実行環境の関係 ..... 8
- 1.3. コマンド入力例での省略表記 ..... 8
- 3.1. 開発に必要なもの ..... 11
- 3.2. 各部名称と機能 ..... 17
- 3.3. ケースモデル展開図パーツ一覧 ..... 19
- 3.4. シリアル通信設定 ..... 30
- 4.1. LED 信号配列 ..... 34
- 4.2. CON6 信号配列(接点入力関連) ..... 41
- 4.3. 接点入力に対応する CON6 ピン番号 ..... 41
- 4.4. CON6 接続可能な電線 ..... 41
- 4.5. CON6 信号配列(接点出力関連) ..... 45
- 4.6. 接点出力に対応する CON6 ピン番号 ..... 45
- 4.7. CON6 接続可能な電線 ..... 45
- 4.8. CON6 接続可能な電線 ..... 50
- 4.9. CON6 信号配列(接点入力、出力関連) ..... 50
- 4.10. 接点入力、出力に対応する CON6 ピン番号 ..... 50
- 4.11. CON6 信号配列(RS-485 関連) ..... 59
- 4.12. CON6 接続可能な電線 ..... 59

# 1. はじめに

このたびは Armadillo をご利用いただき、ありがとうございます。

本書では、Armadillo-IoT ゲートウェイ A6E 上での Node-RED を用いた開発手法を説明します。

Armadillo-IoT ゲートウェイ A6E の詳細な説明に関しては、製品マニュアルに記載しておりますので、以下の URL よりダウンロードしてください。

Armadillo サイト - Armadillo-IoT ゲートウェイ A6E ドキュメントダウンロード

<https://armadillo.atmark-techno.com/armadillo-iot-a6e/resources/documents>

## 1.1. Node-RED について

Node-RED は、IoT アプリケーション開発に適した、オープンソースのローコード開発ツールです。「ノード」と呼ばれる機能ブロックを繋ぎ合わせてアプリケーションを作ります。ビジュアルプログラミング環境のため、Python や JavaScript といったプログラミングの知識がなくてもアプリケーションを作れます。PLC やリモート I/O などの産業機器からのデータ収集や、それらの制御を行うアプリケーション用のノードもあり、製造業 DX の内製ツールとしても注目されています。Armadillo-IoT ゲートウェイ A6E ですぐに使えるようにした Node-RED コンテナを提供します。



Node-RED は、OpenJS Foundation の米国およびその他の国における登録商標または商標です。

## 1.2. 本書について

### 1.2.1. 本書で扱うこと

本書では Armadillo-IoT ゲートウェイ A6E へ Node-RED コンテナをインストールし、Armadillo-IoT ゲートウェイ A6E 上で Node-RED を操作し開発する方法を説明します。

### 1.2.2. 本書で扱わないこと

Node-RED の基本的な説明は記載しておりません。Node-RED の使用方法は各種書籍・Web サイトを参照ください。

また、Armadillo をご利用になられる際の注意事項は、Armadillo 製品マニュアル「注意事項」の章をご一読ください。

### 1.2.3. 本書で必要となる知識と想定する読者

Node-RED を使用したことがあり、使い方を把握しているエンジニアを対象読者として想定しております。

## 1.2.4. フォント

本書では以下のような意味でフォントを使いわけています。

表 1.1 使用しているフォント

フォント例	説明
本文中のフォント	本文
[PC ~]\$ ls	プロンプトとユーザ入力文字列
text	編集する文字列や出力される文字列。またはコメント

## 1.2.5. コマンド入力例

本書に記載されているコマンドの入力例は、表示されているプロンプトによって、それぞれに対応した実行環境を想定して書かれています。「/」の部分はカレントディレクトリによって異なります。各ユーザのホームディレクトリは「~」で表します。

表 1.2 表示プロンプトと実行環境の関係

プロンプト	コマンドの実行環境
[PC /]#	作業用 PC の root ユーザで実行
[PC /]\$	作業用 PC の一般ユーザで実行
[ATDE ~]#	ATDE 上の root ユーザで実行
[ATDE ~]\$	ATDE 上の一般ユーザで実行
[armadillo /]#	Armadillo 上 Linux の root ユーザで実行
[armadillo /]\$	Armadillo 上 Linux の一般ユーザで実行
[container /]#	Podman コンテナ内で実行
⇒	Armadillo 上 U-Boot の保守モードで実行

コマンド中で、変更の可能性のあるものや、環境により異なるものに関しては以下のように表記します。適宜読み替えて入力してください。

表 1.3 コマンド入力例での省略表記


表記	説明
[VERSION]	ファイルのバージョン番号

## 1.2.6. アイコン

本書では以下のようにアイコンを使用しています。



注意事項を記載します。



役に立つ情報を記載します。





用語の説明や補足的な説明を記載します。

### 1.2.7. ユーザー限定コンテンツ

アットマークテクノ Armadillo サイトで購入製品登録を行うと、製品をご購入いただいたユーザーに限定して公開している限定コンテンツにアクセスできるようになります。主な限定コンテンツには、下記のものがあります。

- ・ 各種信頼性試験データ・納入仕様書等製造関連情報

限定コンテンツを取得するには、「2. ユーザー登録」を参照してください。

### 1.2.8. 本書および関連ファイルのバージョンについて

本書を含めた関連マニュアル、ソースファイルやイメージファイルなどの関連ファイルは最新版を使用することをおすすめいたします。本書を読み始める前に、Armadillo サイトで最新版の情報をご確認ください。

Armadillo サイト - Armadillo-IoT ゲートウェイ A6E ドキュメントダウンロード

<https://armadillo.atmark-techno.com/armadillo-iot-a6e/resources/documents>

Armadillo サイト - Armadillo-IoT ゲートウェイ A6E ソフトウェアダウンロード

<https://armadillo.atmark-techno.com/armadillo-iot-a6e/resources/software>

## 2. ユーザー登録

アットマークテクノ製品をご利用のユーザーに対して、購入者向けの限定公開データの提供や大切なお知らせをお届けするサービスなど、ユーザー登録すると様々なサービスを受けることができます。サービスを受けるためには、「アットマークテクノ Armadillo サイト」にユーザー登録をする必要があります。

ユーザー登録すると次のようなサービスを受けることができます。

- ・ 製品仕様や部品などの変更通知の閲覧・配信
- ・ 購入者向けの限定公開データのダウンロード
- ・ 該当製品のバージョンアップに伴う優待販売のお知らせ配信
- ・ 該当製品に関する開発セミナーやイベント等のお知らせ配信

詳しくは、「アットマークテクノ Armadillo サイト」をご覧ください。

アットマークテクノ Armadillo サイト

<https://armadillo.atmark-techno.com/>

### 2.1. 購入製品登録

ユーザー登録完了後に、購入製品登録することで、「購入者向けの限定公開データ」をダウンロードすることができるようになります。

購入製品登録の詳しい手順は以下の URL をご参照ください。

Armadillo-IoT ゲートウェイ A6E 購入製品登録

<https://armadillo.atmark-techno.com/armadillo-iot-a6e/register>

# 3. Armadillo のセットアップ

Armadillo-IoT ゲートウェイ A6E 上で Node-RED を使用するためのセットアップ方法を説明します。Armadillo-IoT ゲートウェイ A6E に Node-RED 対応のインストールディスクをインストールし、開発用パソコンと Armadillo-IoT ゲートウェイ A6E を有線 LAN で繋がる様にします。セットアップが完了すると Node-RED の起動画面が表示されます。

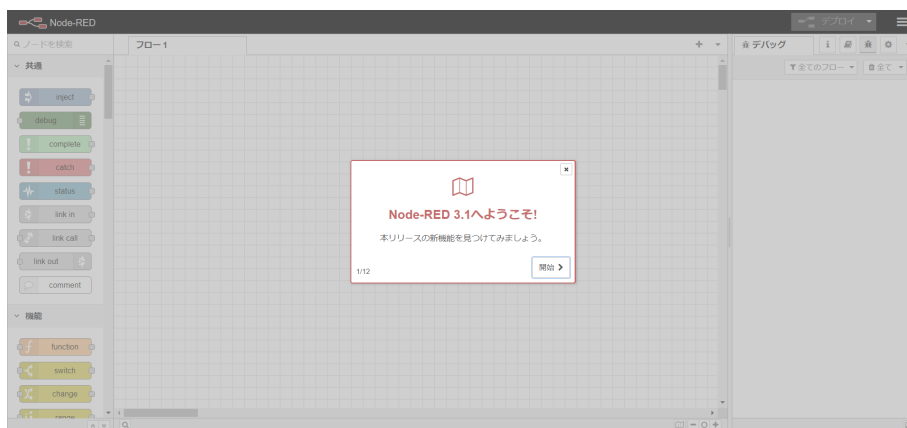


図 3.1 初回起動時の Node-RED 画面

## 3.1. 作業の前に

### 3.1.1. 開発セット内容物の確認

開発セットに同梱されている内容物は、同梱の内容物一覧でご確認いただけます。お使いになる前に、内容物がすべて揃っていることをご確認ください。万一、内容物の不足または部品の破損等がございましたら、ご購入の販売代理店までご連絡ください。

### 3.1.2. 開発に必要なもの

Armadillo-IoT ゲートウェイ A6E 上で Node-RED を使用するためには以下のものがが必要です。

表 3.1 開発に必要なもの

品目	説明
Armadillo-IoT ゲートウェイ A6E 開発セット	本製品です。
開発用パソコン	Linux または Windows が動作し、ネットワークインターフェースと 1 つ以上の USB ポート、microSD ポートを持つパソコンです。
有線 LAN ケーブル	本ガイドでは Armadillo-IoT ゲートウェイ A6E と開発用パソコンを有線 LAN で接続する前提で記載しています。
512MB 以上の microSD カード	Node-RED の開発環境をインストールするために使用します。開発用パソコンの SD ポートが microSD ではなく、SD/miniSD ポートの場合、microSD へ変換するアダプターも必要となります。

### 3.1.3. 接続方法

Armadillo-IoT ゲートウェイ A6E と周辺装置の接続例を「図 3.2. Armadillo-IoT ゲートウェイ A6E と周辺デバイスの接続例」に示します。

開発用パソコンとのファイル転送やインターネットへの接続は、LAN を介して行いますので LAN ケーブルを接続してください。

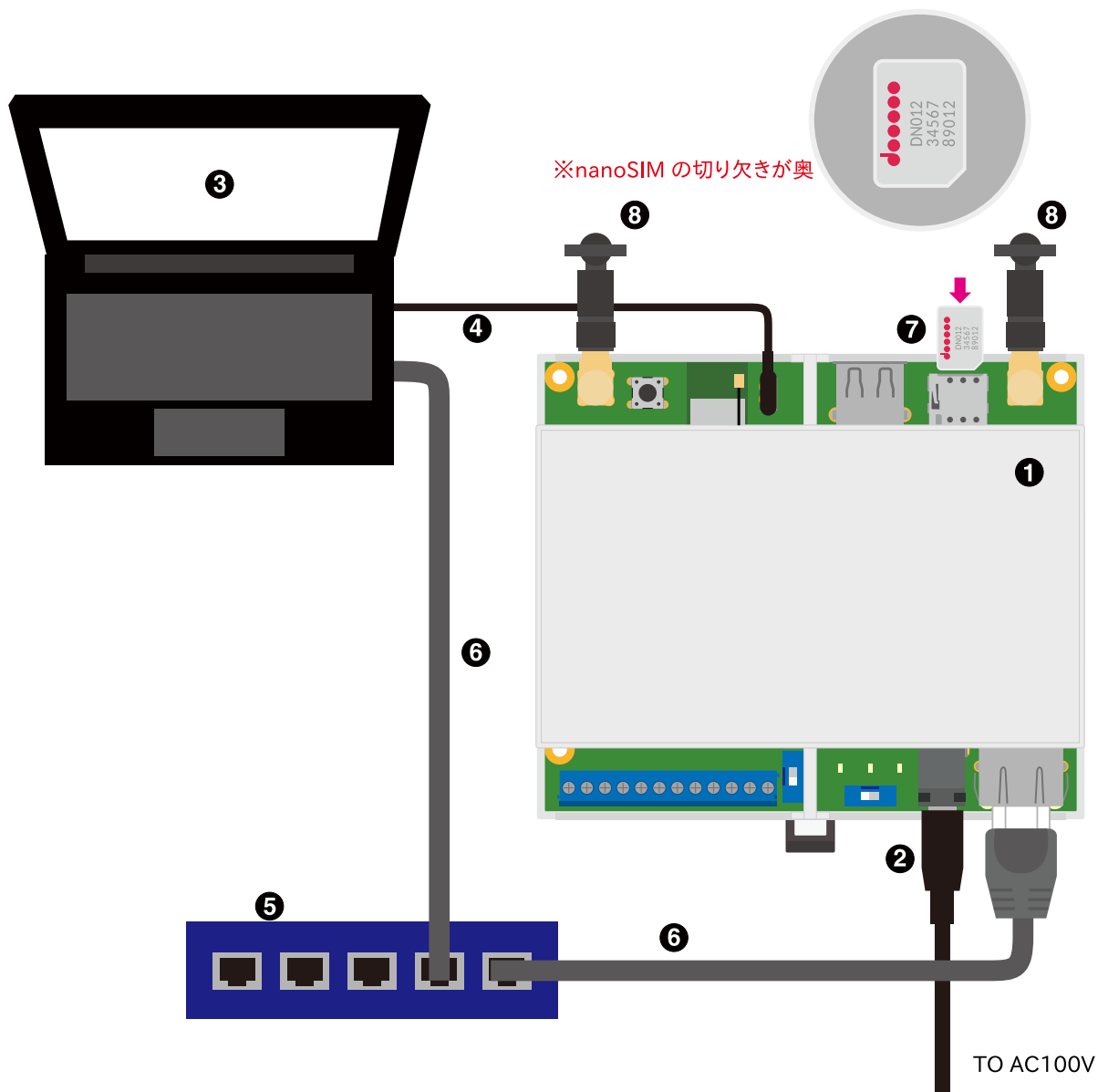


図 3.2 Armadillo-IoT ゲートウェイ A6E と周辺デバイスの接続例

- ① Armadillo-IoT ゲートウェイ A6E
- ② AC アダプタ(12V/2A)
- ③ 作業用 PC
- ④ シリアル通信用 USB ケーブル(A-microB)
- ⑤ LAN HUB
- ⑥ Ethernet ケーブル

- ⑦ nanoSIM カード
- ⑧ LTE 用外付けアンテナ

### 3.1.4. 起動デバイス設定スイッチについて

起動デバイス設定スイッチを操作することで、起動デバイスを設定することができます。

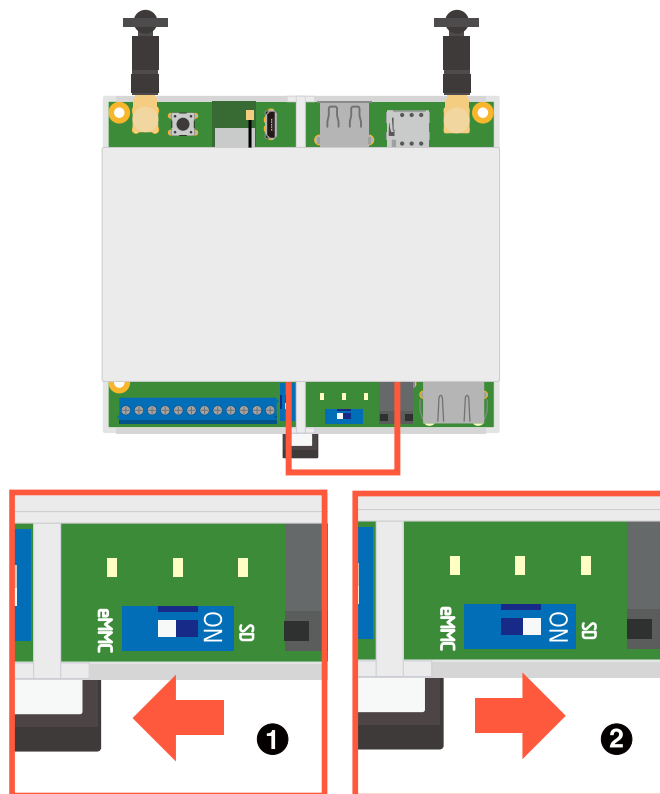


図 3.3 起動デバイス設定スイッチの操作

- ① 起動デバイスは eMMC になります。
- ② 起動デバイスは microSD になります。



起動デバイス設定スイッチの両脇の基板の上に、白い文字で eMMC/SD とシルク記載しているので、操作の目印にご利用ください。

## 3.2. インストールディスクを使用してセットアップを行う

microSD カードを用いてインストールディスクを作成して Node-RED コンテナと最新の Armadillo Base OS をインストールする手段を説明します。インストールディスクイメージについては以下から取得可能です。

- ・ Armadillo Base OS インストールディスクイメージ (Node-RED 対応)

Armadillo-IoT ゲートウェイ A6E インストールディスクイメージ [<https://armadillo.atmark-techno.com/resources/software/armadillo-iot-a6e/disc-image>] から「Armadillo Base OS インストールディスクイメージ (Node-RED 対応)」をダウンロードしてください。

作業用 PC 上でダウンロードした Zip ファイルにはインストールディスクイメージ (baseos-6e-installer-[VERSION]-node-red.img) のファイルが含まれています。このファイルを microSD カードに書き込みます。

### 3.2.1. Windows PC 上でインストールディスクを作成する

Windows PC の場合、[Win32 Disk Imager Renewal] を使用して img ファイルを microSD カードに書き込みます。[Win32 Disk Imager Renewal] は以下からダウンロードしてください。

- ・ Win32 Disk Imager Renewal リリースページ

Win32 Disk Imager Renewal リリースページ [<https://github.com/dnobori/DN-Win32DiskImagerRenewal/releases/>]

最新版の EXE ファイルをダウンロードして起動します。以下に img ファイルの書き込み手順について示します。

1. ダウンロードしてきた Zip ファイルを解凍して書き込み用の img ファイルを取得します。
2. [image File] に先ほど展開した img ファイルを指定します。
3. [Device] には microSD が挿入されているスロットを指定します。

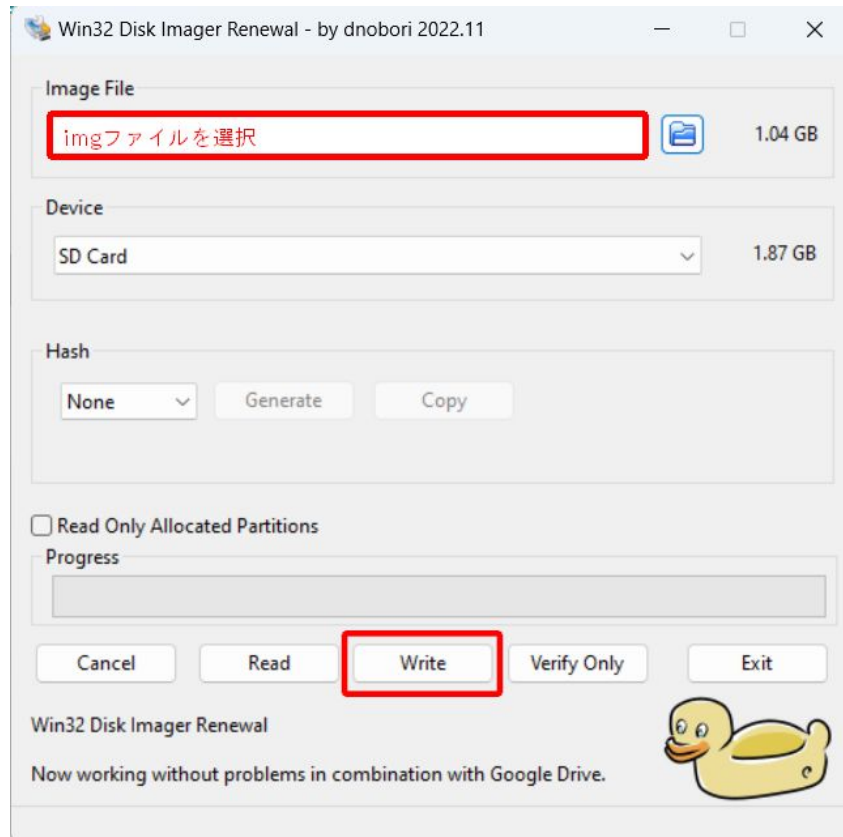


図 3.4 Win32 Disk Imager Renewal 設定画面

4. [write] ボタンを押すと警告が出ますので、問題なければ[はい]を選択して書き込みます。
5. 書き込み終了ダイアログが表示されたらインストールディスクの作成は完了です。

### 3.2.2. Linux PC 上でインストールディスクを作成する

Linux PC の場合、以下のように microSD カードに書き込むことができます。sd[X] の [X] は microSD のドライブを指定してください。

インストールディスクを作成する。

```
[PC ~]$ unzip baseos-6e-installer-[VERSION]-node-red.zip  
[PC ~]$ sudo dd if=baseos-6e-installer-[VERSION]-node-red.img of=/dev/sd[X] bs=1M oflag=direct  
status=progress
```

↵

書き込みが終了したらインストールディスクの作成は完了です。

### 3.2.3. 作成したインストールディスクを使用してセットアップする

1. Armadillo-IoT ゲートウェイ A6E のケースを開けます。ケースの分解手順は「3.4.2. ケースの分解」を参照してください。
2. インストールディスクを Armadillo-IoT ゲートウェイ A6E の microSD に挿入します。microSD の挿抜方法は「3.5. microSD カードの挿抜方法」を参照してください。

3. Armadillo-IoT ゲートウェイ A6E の起動デバイス設定スイッチを ON にし、起動デバイスを microSD に設定します。起動デバイス設定スイッチの場所は、「3.3. インターフェースレイアウト」の「**17** 起動デバイス設定スイッチ」です。
4. Armadillo-IoT ゲートウェイ A6E の AC アダプターをコンセントに繋ぎ電源を投入すると、20 分程度でソフトウェアの初期化が完了します。
5. インストールが完了すると電源が切れます(SYS(システム LED)が消灯します)。LED の位置は「3.3. インターフェースレイアウト」の「**13**システム LED」です。
6. Armadillo-IoT ゲートウェイ A6E の AC アダプターをコンセントから抜き、続いての起動デバイス設定スイッチを OFF に設定し、microSD カードを外してください。
7. Armadillo-IoT ゲートウェイ A6E のケースを組み立ててください。組み立て手順は「3.4.1. ケースの組み立て手順」を参照してください。
8. 10 秒以上待ってから再び電源を投入すると Node-RED が入った Armadillo-IoT ゲートウェイ A6E が起動します。

### 3.3. インターフェースレイアウト

Armadillo-IoT ゲートウェイ A6E のインターフェースレイアウトです。一部のインターフェースを使用する際には、ケースを開ける必要があります。



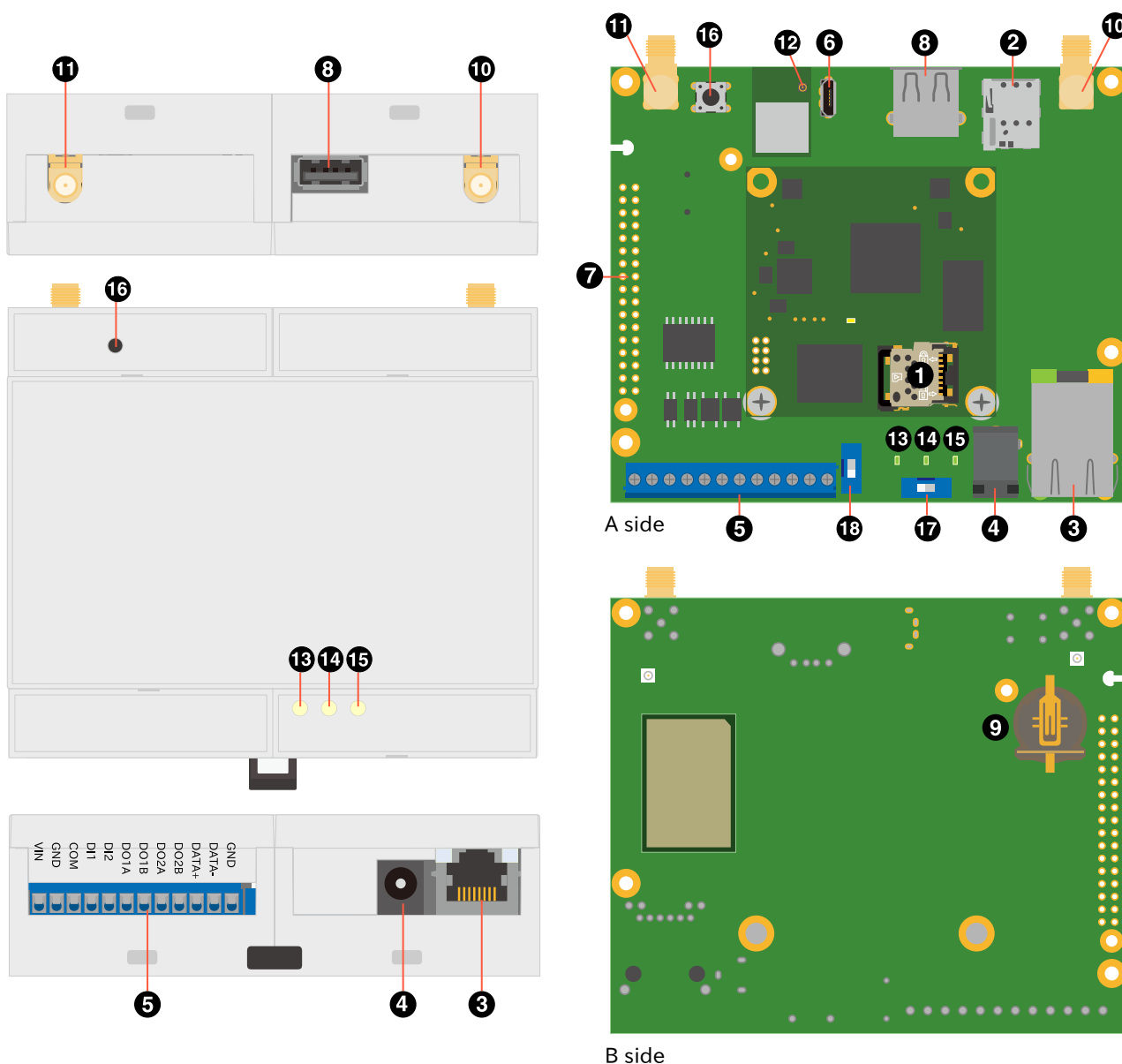


図 3.5 Armadillo-IoT ゲートウェイ A6E インターフェースレイアウト

表 3.2 各部名称と機能

番号	名称	形状	説明
1	SD インターフェース	microSD スロット	外部ストレージが必要な場合 <sup>[a]</sup> や、ブートローダーを破壊してしまった時の復旧等で使用します。microSD カードを挿入します。
2	nanoSIM インターフェース	nanoSIM スロット	LTE データ通信を利用する場合に使用します。nanoSIM カードを挿入します。
3	LAN インターフェース	RJ-45 コネクタ	有線 LAN を利用する場合に使用します。LAN ケーブルを接続します。
4	電源入力インターフェース	DC ジャック	Armadillo-IoT ゲートウェイ A6E への電源供給で使用します <sup>[b]</sup> 。付属の AC アダプタ (12V/2A) を接続します。対応プラグ:内径 2.1mm、外形 5.5mm

番号	名称	形状	説明
5	入出力インターフェース	端子台	絶縁入出力、RS485 通信する場合に使用します。Armadillo-IoT ゲートウェイ A6E への電源供給も可能です <sup>[b]</sup> 。
6	USB コンソールインターフェース	USB micro B コネクタ	コンソール入出力を利用する場合に使用します。USB micro B ケーブルを接続します。
7	拡張インターフェース	ピンヘッダ 34 ピン(2.54mm ピッチ)	機能拡張する場合に使用します。2.54mm ピッチのピンヘッダを実装することができます。
8	USB インターフェース	USB 2.0 Type-A コネクタ	外部ストレージが必要な場合等に使用します。USB メモリ等を接続します。
9	RTC バックアップインターフェース	電池ボックス	リアルタイムクロックのバックアップ給電が必要な場合に使用します。対応電池: CR1220 等
10	LTE アンテナインターフェース	SMA コネクタ	LTE データ通信を利用する場合に使用します。付属のアンテナを接続します。
11	LTE アンテナインターフェース	SMA コネクタ	LTE データ通信を利用する場合に使用します。付属のアンテナを接続します。
12	WLAN/BT アンテナインターフェース	MHF4 コネクタ	WLAN/BT データ通信を利用する場合に使用します。付属の WLAN/BT 用アンテナを接続します。
13	システム LED	LED(緑色、面実装)	電源の入力状態を表示する緑色 LED です。
14	アプリケーション LED	LED(緑色、面実装)	アプリケーションの状態を表示する緑色 LED です。
15	ワイヤレス WAN	LED(緑色、面実装)	LTE 通信の状態を表示する緑色 LED です。
16	ユーザースイッチ	タクトスイッチ	ユーザーが利用可能なタクトスイッチです。
17	起動デバイス設定スイッチ	DIP スイッチ	起動デバイスを設定する時に使用します。
18	RS485 終端抵抗設定スイッチ	DIP スイッチ	RS485 通信の終端抵抗を設定する時に使用します。

<sup>[a]</sup>WLAN 搭載モデルでは SD をストレージとして使用できません。

<sup>[b]</sup>DC ジャックと端子台の両方から同時に電源供給することはできません。

### 3.4. ケースの組み立てと分解方法

本製品はねじを使用しないスナップフィット方式を採用しており、容易に組み立てと分解が可能です。分解する際には手のけがやパーツの破損を防止するためマイナスドライバーなどの工具を使用してください。

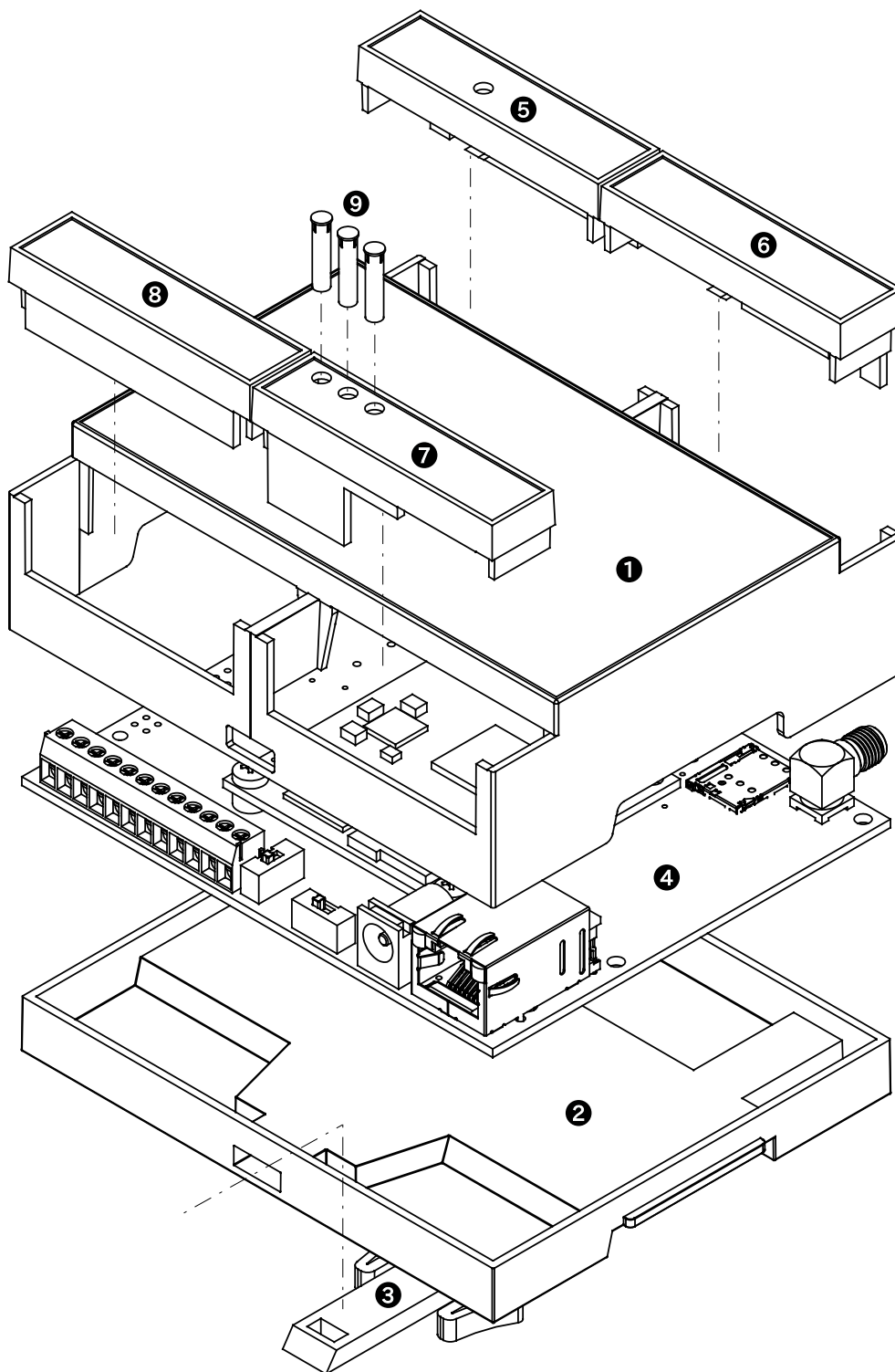


図 3.6 ケースモデル展開図

表 3.3 ケースモデル展開図パーツ一覧

番号	名称	説明
1	ケーストップ	ケース上側のパーツです。ケースボトムとは 4 か所のツメで固定されます。ケースを分解する際は、マイナスドライバーを使用してツメを破損させないよう慎重に取り外してください。

番号	名称	説明
2	ケースボトム	ケース下側のパーツです。
3	フック	ケースを DIN レールに固定するためのパーツです。
4	基板	
5	カバーパーツ A	ケース開口部のカバーです。ケースストップとは 1 か所のツメで固定されます。
6	カバーパーツ B	ケース開口部のカバーです。ケースストップとは 1 か所のツメで固定されます。
7	カバーパーツ C	ケース開口部のカバーです。ケースストップとは 1 か所のツメで固定されます。
8	カバーパーツ D	ケース開口部のカバーです。ケースストップとは 1 か所のツメで固定されます。
9	LED ライトパイプ	カバーパーツ C に装着する LED のライトパイプです。強い衝撃を加えた場合、ライトパイプが外れる場合がありますので、「図 3.6. ケースモデル展開図」を参考にカバーパーツ C の丸穴に差し込んでください。

フックは以下の図を参考に取り付けてください。

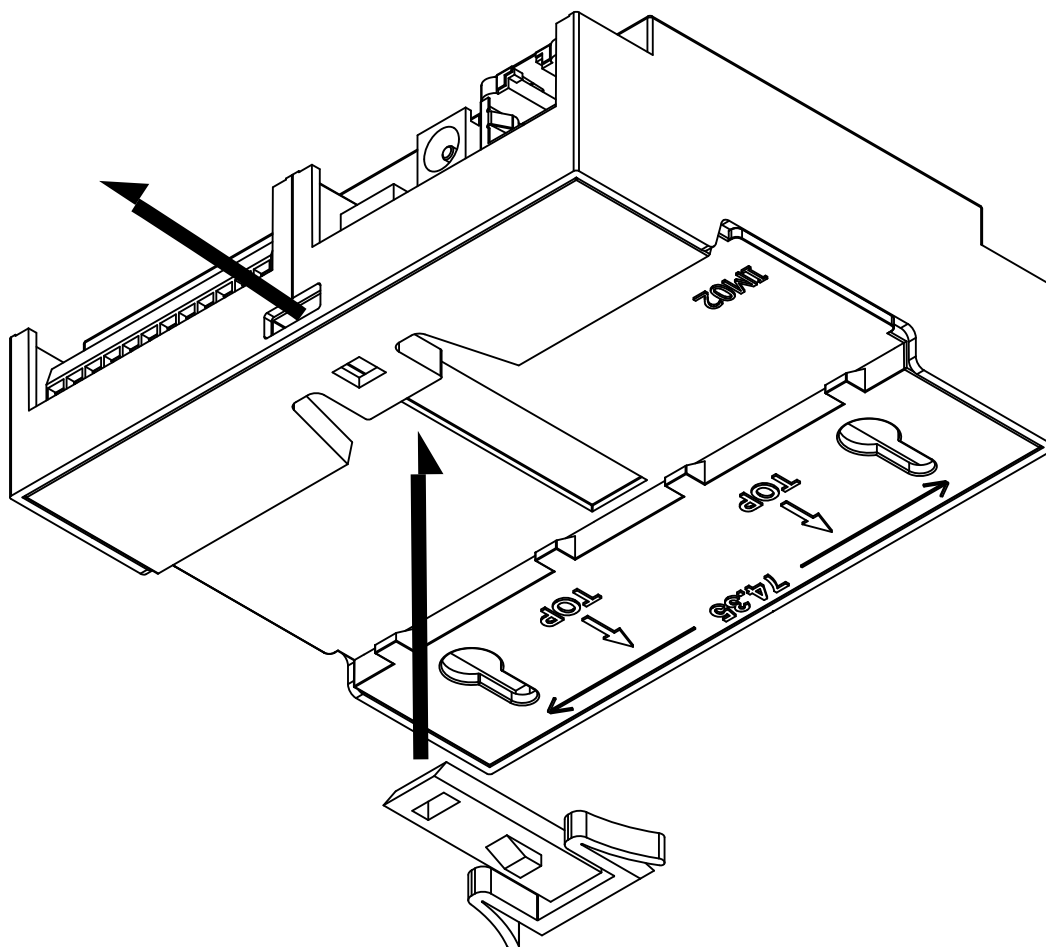


図 3.7 フック取り付け 1

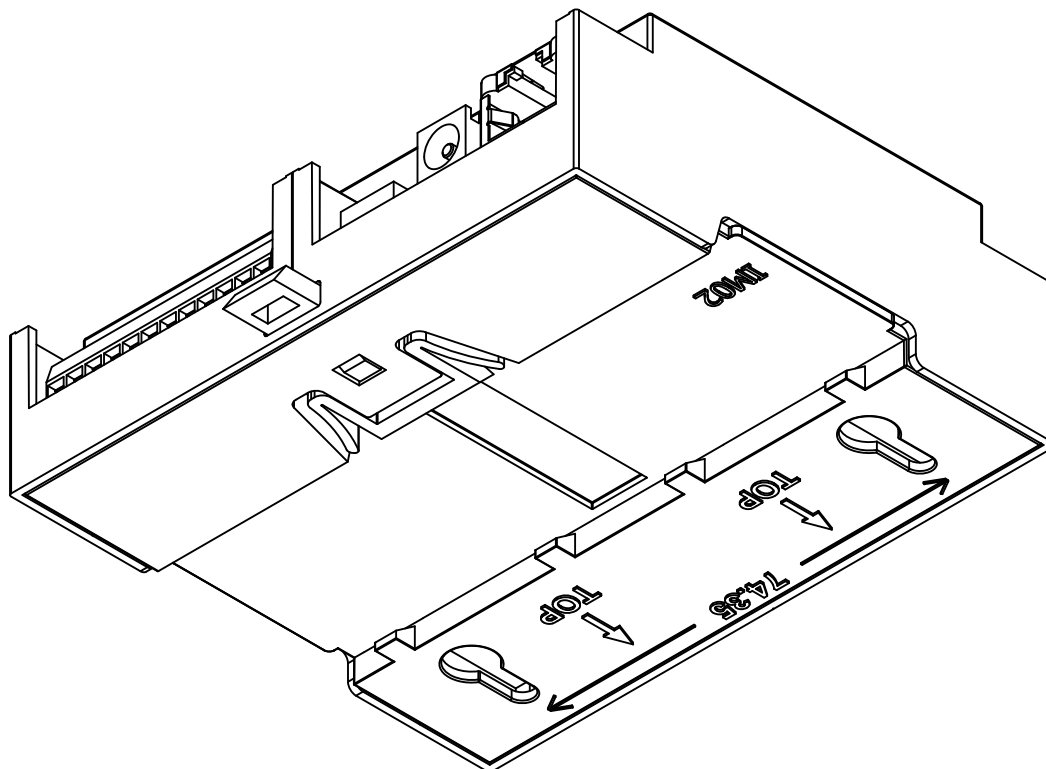



図 3.8 フック取り付け 2

### 3.4.1. ケースの組み立て手順




microSD カードの取り付けは、ケースの組み立て前に行う必要があります。取り付け手順については、を参照してください。

以下の手順に従い、ケースを組み立ててください。

1. 基板をケーストップに入れる
2. ケースボトムをケーストップにはめ込み、基板を固定する
3. フックをケースボトムにはめ込む
4. カバーパーツをケーストップにはめ込む

### 3.4.2. ケースの分解



WLAN+BT コンポモジュールを搭載した製品におきましては、ケーストップに貼り付けられている WLAN 基板アンテナのケーブルが製品基板の ANT3 と接続しています。ケースを分解する際は、ANT3 に無理な力が加わらないよう慎重に作業してください。



ツメに強い力を加えますと破損する恐れがありますので、十分ご注意ください。

マイナスドライバーなどの工具を用意してください。以下の手順に従い、ケースを分解してください。

1. フックをケースボトムから取り外す
2. ケースボトムを取り外す
3. 基板を取り外す
4. カバーパーツを取り外す

フックはツメで固定されていますので、「図 3.9. フックのツメ」を参考にツメを押しながらフックを引き出してください。

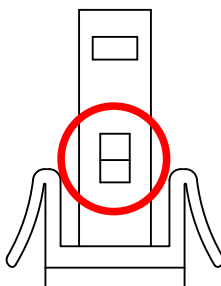


図 3.9 フックのツメ

ケースボトムはツメ 4 か所で固定されていますので、「図 3.10. ケースボトムのツメ」を参考にマイナスドライバーをケースの隙間に差し込み順に外してください。

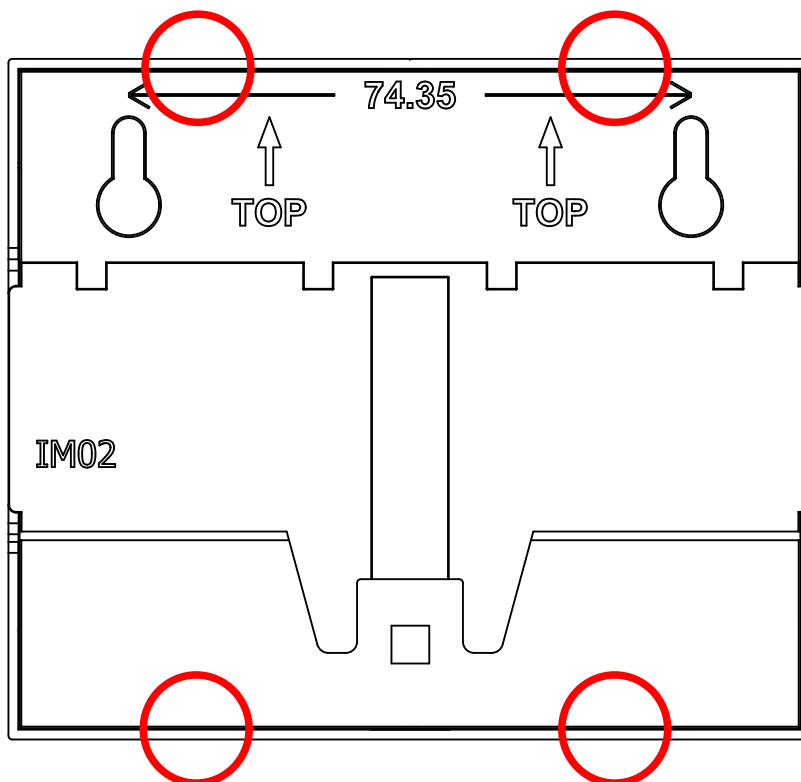


図 3.10 ケースボトムのカギ

Cat.1 モデル, Cat.M1 モデルではアンテナコネクタがケース開口部より飛び出しているため、反対側の LAN コネクタ側から先にケーストップから出すようにしてください。基板を取り外す際、LAN コネクタの突起部がケーストップに当たらないよう、ケースを広げながら基板を取り外すようにしてください。

カバーはツメ 1 か所でケーストップに固定されています。「図 3.11. カバーのカギ」を参考にマイナスドライバーをケースの隙間に差し込み外してください。

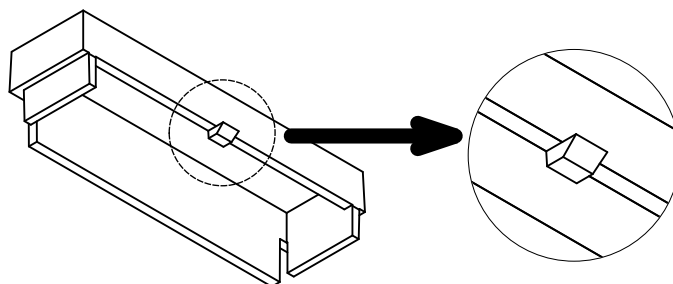


図 3.11 カバーのカギ

### 3.5. microSD カードの挿抜方法

1. 上からカバーを軽く押し、約 1.2mm スライドさせて、ロックを解除します。

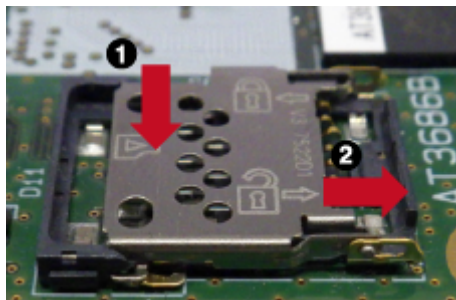


図 3.12 カバーのロックを解除する

2. カバーを開けます。

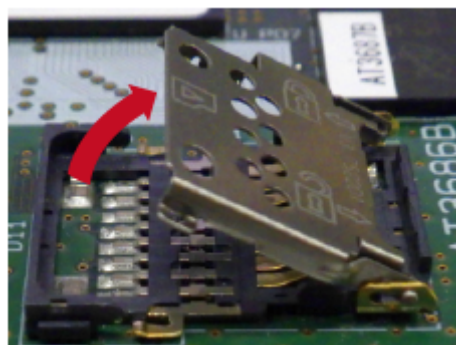


図 3.13 カバーを開ける



カバーは過度な力で回転させたり、回転方向以外の方向へ力を加えると、破損の原因となりますので、ご注意ください。

3. 任意の角度までトレイを開いた状態で、microSD カードを挿抜します。



図 3.14 microSD カードの挿抜





microSD カード挿入方向については、カバーに刻印されているカードマークを目安にしてください。



図 3.15 カードマークの確認

4. カバーを閉めます。

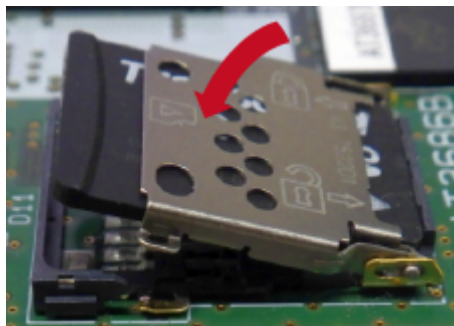


図 3.16 カバーを閉める

5. 上からカバーを軽く押し、約 1.2mm スライドさせて、ロックします。

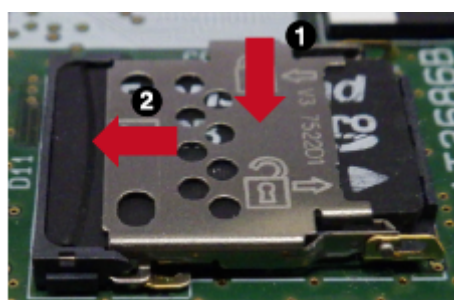


図 3.17 カバーをロックする



microSD カード装着後のカードの抜き取り手順は挿入時と同じです。

## 3.6. ネットワークに接続する

Node-RED の開発をするパソコンと Armadillo のネットワークを接続するために Armadillo-IoT ゲートウェイ A6E の IP アドレスを取得または設定します。

Armadillo の有線 LAN は、初期状態で DHCP の設定となっておりますので、DHCP が稼働している有線 LAN に接続した場合は、なんらかの IP アドレスが付与された状態になっています。

DHCP で IP アドレスが付与される環境の場合、ABOS Web を使用することで IP アドレスの確認が可能です。

1. Armadillo-IoT ゲートウェイ A6E を「3.2. インストールディスクを使用してセットアップを行う」でセットアップを済ませた場合は ABOS Web が起動しています。再起動がまだの場合は電源を再投入します。
2. 開発用パソコンで Web ブラウザーを起動し <https://armadillo.local:58080> にアクセスします。



ABOS Web が動作する Armadillo が、同じ LAN 上に複数ある場合、ABOS Web に接続する URL のホスト名部分 (armadillo.local) は、2 台目は armadillo-2.local、3 台目は armadillo-3.local と、違うものが自動的に割り当てられます。この場合どの URL がどの Armadillo かを判別するのは難しいので固定 IP アドレスを設定し、IP アドレスで指定できるようにする方法があります。固定 IP アドレスの設定方法は「3.6.1. Armadillo の有線 LAN に固定 IP アドレスを設定する」を参照ください。

3. 初回接続時は、ABOS Web のパスワードを設定する必要がありますので、設定します。



図 3.18 パスワード登録画面

4. "初回ログイン"のパスワード登録画面で、"パスワード" フィールドと "パスワード(確認)" フィールドに、登録したいパスワードを入力してから、"登録" ボタンをクリックしてください。パスワード登録完了画面が表示されたら、パスワード登録の完了です。



図 3.19 パスワード登録完了画面

5. ログインします。



図 3.20 ログイン画面

6. 左のメニューから「状態一覧」を選択します。



図 3.21 状態一覧を選択

7. 「LAN 情報」の「IP アドレス」の欄に有線 LAN の IP アドレスが表示されていますのでメモしておいてください。



図 3.22 LAN 情報

### 3.6.1. Armadillo の有線 LAN に固定 IP アドレスを設定する

DHCP サーバーが存在しない環境で開発を行う場合、Armadillo-IoT ゲートウェイ A6E に固定の IP アドレスを付与して開発を行います。

Armadillo にシリアルコンソールを接続してコマンドを入力する必要があります。シリアルコンソールの接続方法はを参照ください。

1. Armadillo-IoT ゲートウェイ A6E と開発用パソコンを開発セットに同梱されています USB(A オス-microB)ケーブルで接続します。Armadillo-IoT ゲートウェイ A6E 側は「図 3.5. Armadillo-IoT ゲートウェイ A6E インターフェースレイアウト」に示す「③USB コンソールインターフェース」に microB を、開発用パソコン側は Type-A を接続してください。
2. 開発用パソコンでシリアルコンソールを開きます。対応しているソフトウェアは、Windows OS であれば TeraTerm、Linux であれば minicom などがありますので、インストールしてご利用ください。
3. ソフトウェアのシリアル設定を「表 3.4. シリアル通信設定」に示します。

表 3.4 シリアル通信設定

項目	設定
転送レート	115,200bps
データ長	8bit
ストップビット	1bit
パリティ	なし
フロー制御	なし

1. Armadillo-IoT ゲートウェイ A6E の AC アダプターをコンセントに繋ぎ電源を投入します。
2. シリアルコンソールに以下のようにログインプロンプトが表示されましたら、root ユーザーでログインします。特に設定していない場合の root ユーザーの初期パスワードは root です。

```

Welcome to Alpine Linux 3.18
Kernel 5.10.197-0-at on an armv7l (/dev/ttyxc2)

armadillo login:
    
```

3. "Wired connection 1" に固定 IP アドレスを設定します。

```

[armadillo ~]# nmcli connection modify "Wired connection 1" \
  ipv4.method manual ipv4.addresses 192.0.2.10/24 ipv4.gateway 192.0.2.1
    
```

4. 有線 LAN eth0 を再起動します。

```

[armadillo ~]# nmcli connectio down "Wired connection 1"
[armadillo ~]# nmcli connectio up "Wired connection 1"
    
```

5. 設定された IP アドレスは ip addr コマンドで確認できます。

```

[armadillo ~]# ip addr show eth0
2: eth0: ... 中略...
   inet 192.0.2.10/24 brd 192.0.2.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
   ... 後略 ...
    
```

6. 設定を永続化するために、persist\_file コマンドを実行してください。

```
[armadillo ~]# persist_file /etc/NetworkManager/system-connections/"Wired connection
1.nmconnection"
```



### 3.7. Node-RED に接続する

パソコンの Web ブラウザから、「3.6. ネットワークに接続する」 で取得した IP アドレスを使用し、`http://<ip アドレス>:1880/` にアクセスしてください。

Node-RED の起動画面が表示されたらセットアップは終了です。

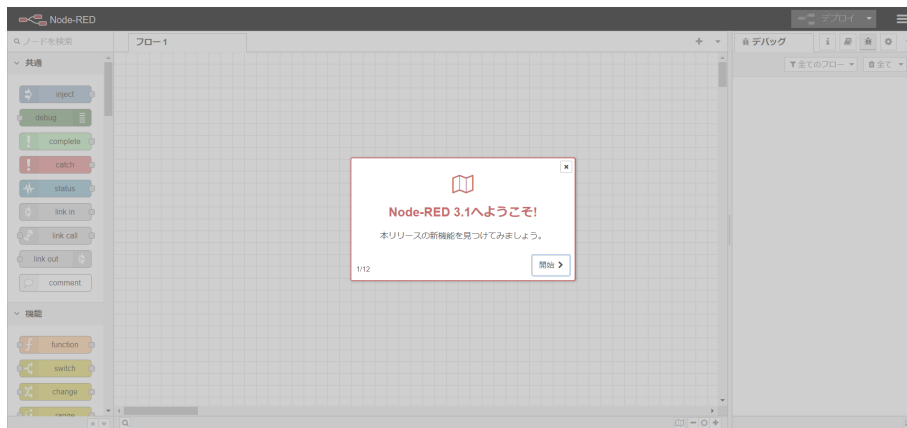


図 3.23 初回起動時の Node-RED 画面

## 4. 開発を行う

### 4.1. Node-RED に接続する

「3. Armadillo のセットアップ」を完了している場合は Armadillo を起動すると、自動的に Node-RED コンテナが起動します。Node-RED への接続には「3.6. ネットワークに接続する」で取得した IP アドレスを使用します。

パソコンの Web ブラウザから、`http://<ip アドレス>:1880/` にアクセスしてください。

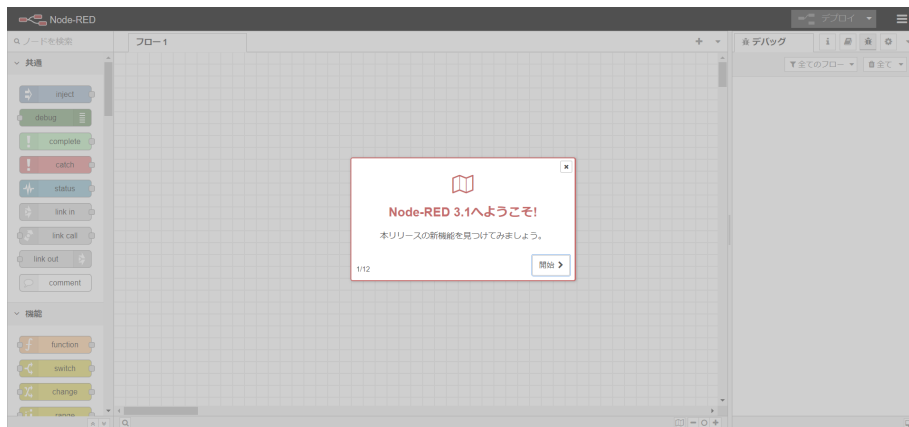


図 4.1 初回起動時の Node-RED 画面

Node-RED には大きく 3 つの領域があります。

- ・ パレット

使用可能なコアノード、カスタムノードの一覧を表示します。

- ・ ワークスペース

Node-RED では一つのアプリケーションの流れをフローという単位で表します。ワークスペース上にノードを繋げてフローを作成します。パレットから必要なノードをワークスペースにドラッグ、ドロップで配置します。

- ・ サイドバー

選択したノードの情報や、デバッグメッセージを表示します。





図 4.2 Node-RED の画面領域

## 4.2. Node-RED コンテナのログを表示する

パソコンの Web ブラウザから、<http://<ip アドレス>:1880/node-red.log> にアクセスすると Node-RED コンテナのログが表示されます。

このログファイルは `/var/app/rollback/volumes/node-red/log` に保存されています。ログのサイズが 5MB を超えると新しいログファイルに移行します。古いログファイルを表示したい場合は、<http://<ip アドレス>:1880/node-red.log.1> にアクセスしてください。

## 4.3. Node-RED で利用可能なノードの一覧

「3. Armadillo のセットアップ」でインストールした Node-RED コンテナには Node-RED のコアノードの他にインストールされているカスタムノードがあります。インストール済のカスタムノードは以下になります。

- ・ Dashboard ノード

Dashboard を表示するノードです。

- ・ exit ノード

Node-RED を終了するためのノードです。

- ・ GPIO in, GPIO out ノード

GPIO 入力、出力を行うノードです。

- ・ falling edge, rising edge, hysteresis ノード

GPIO 入力の立ち上がりエッジ、立ち下がりエッジ、もしくは両方の検出を行うノードです。

- ・ modbus ノード

RS485 を用いた Modbus 通信用のノードです。以下のノードが使用可能です。ただし、`modbus-flex-server` ノードは使用不可となっています。`modbus-response`, `modbus-read`, `modbus-getter`, `modbus-flex-getter`, `modbus-write`, `modbus-flex-write`, `modbus-server`, `modbus-queue-info`, `modbus-flex-connector`, `modbus-io-config`, `modbus-response-filter`, `modbus-flex-sequencer` ノード

## 4.4. フローを作成する

Node-RED では一つのアプリケーションの流れをフローという単位で表します。各ノードを使用したフローの作成方法について紹介します。

### 4.4.1. LED を制御する

ここでは、LED を 1 秒間隔で点滅するフローを作成します。すべて Node-RED のコアノードを使用します。Armadillo-IoT ゲートウェイ A6E の LED は以下がありますが、今回対象にするのはアプリケーション LED です。アプリケーション LED の場所については「3.3. インターフェースレイアウト」をご確認ください。

アプリケーション LED は /sys/class/leds/app/brightness ファイルへ値を書き込むことによって、LED の点灯/消灯を行うことができます。0 を書き込むと消灯、0 以外の値 (1~255) を書き込むと点灯します。

表 4.1 LED 信号配列

部品番号	名称(色)	説明
SYS	システム LED(緑)	電源(VCC_3.3V)の入力状態を表示、i.MX6ULL の UART2_CTS_B ピン (GPIO1_IO22)に接続 (Low: 消灯、High: 点灯)
APP	アプリケーション LED(緑)	アプリケーションの状態を表示、i.MX6ULL の UART2_RTS_B ピン(GPIO1_IO23)に接続 (Low: 消灯、High: 点灯)
WWAN	ワイヤレス WAN LED(緑)	LTE 通信の状態を表示、i.MX6ULL の UART1_RX_DATA ピン(GPIO1_IO17)に接続 (Low: 消灯、High: 点灯)

1. パレットから [inject] ノードをワークスペースにドラッグ、ドロップします。
2. [inject] ノードのプロパティを編集します。[inject] は 2 秒間隔で実行します。

名前: LED Blink  
 繰り返し: 指定した時間間隔  
 時間間隔: 2 秒



図 4.3 [inject] ノードのプロパティ内容

3. [trigger] ノードをドラッグ、ドロップします。
4. [trigger] ノードのプロパティを編集します。[trigger] は 1 を送信して 1 秒待機後、0 を送信します。これにより点滅動作を実現します。

送信データ : 1  
 送信後の処理 : 指定した時間待機  
 1 秒  
 再送信データ : 0



図 4.4 [trigger] ノードのプロパティ内容

5. [LED Blink] ノードの右側にある端子をクリックし、[trigger] ノードの左側の端子を選択して放します。
6. [write file] ノードをドラッグ、ドロップします。
7. [write file] ノードのプロパティを編集します。/sys/class/leds/app/brightness ファイルへ [trigger] からの送信データを書き込みます。

ファイル名: /sys/class/leds/app/brightness  
 動作: ファイルを上書き  
 文字コード: デフォルト  
 名前: APP LED



図 4.5 [write file] ノードのプロパティ内容

8. [trigger] ノードの右側にある端子をクリックし、[APP LED] ノードの左側の端子を選択して放します。
9. [debug] ノードをドラッグ、ドロップします。
10. [APP LED] ノードの右側にある端子をクリックし、[debug] ノードの左側の端子を選択して放します。

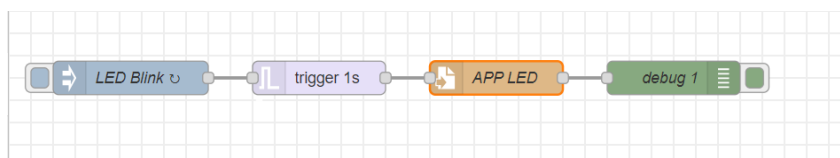


図 4.6 LED を 1 秒間隔で点滅するフロー

11. 画面右上の [デプロイ] を押します。アプリケーション LED が 1 秒毎に点滅を繰り返す動きをすれば成功です。

## 4.4.2. CPU の測定温度を取得する

ここでは、Armadillo-IoT ゲートウェイ A6E の CPU の測定温度を 1 秒間隔で取得するフローを作成します。全て Node-RED のコアノードを使用します。/sys/class/thermal/thermal\_zone0/temp ファイルの値を読み出すことによって測定温度を取得することができます。温度はミリ°C の単位で表示されるため、°C 単位への変換も行います。

1. パレットから [inject] ノードをワークスペースにドラッグ、ドロップします。
2. [inject] ノードのプロパティを編集します。[inject] は 1 秒間隔で実行します。

繰り返し: 指定した時間間隔  
時間間隔: 1 秒



図 4.7 [inject] ノードのプロパティ内容

3. [read file] ノードをドラッグ、ドロップします。
4. [read file] ノードのプロパティを編集します。/sys/class/thermal/thermal\_zone0/temp ファイルの値を読み出します。

ファイル名: /sys/class/thermal/thermal\_zone0/temp  
 出力形式: 文字列  
 文字コード: デフォルト  
 名前: CPU temp



図 4.8 [read file] ノードのプロパティ内容

5. [Get CPU temp] ノードの右側にある端子をクリックし、[CPU temp] ノードの左側の端子を選択して放します。
6. [function] ノードをドラッグ、ドロップします。
7. [function] ノードのプロパティを編集します。CPU の測定温度がミリ°C の単位のため、°C の単位へ変換します。msg.payload で渡された値を 1000 で割り、msg.payload に戻します。

名前: CPU temp calc  
 コード:

```
msg.payload = msg.payload / 1000;
return msg;
```

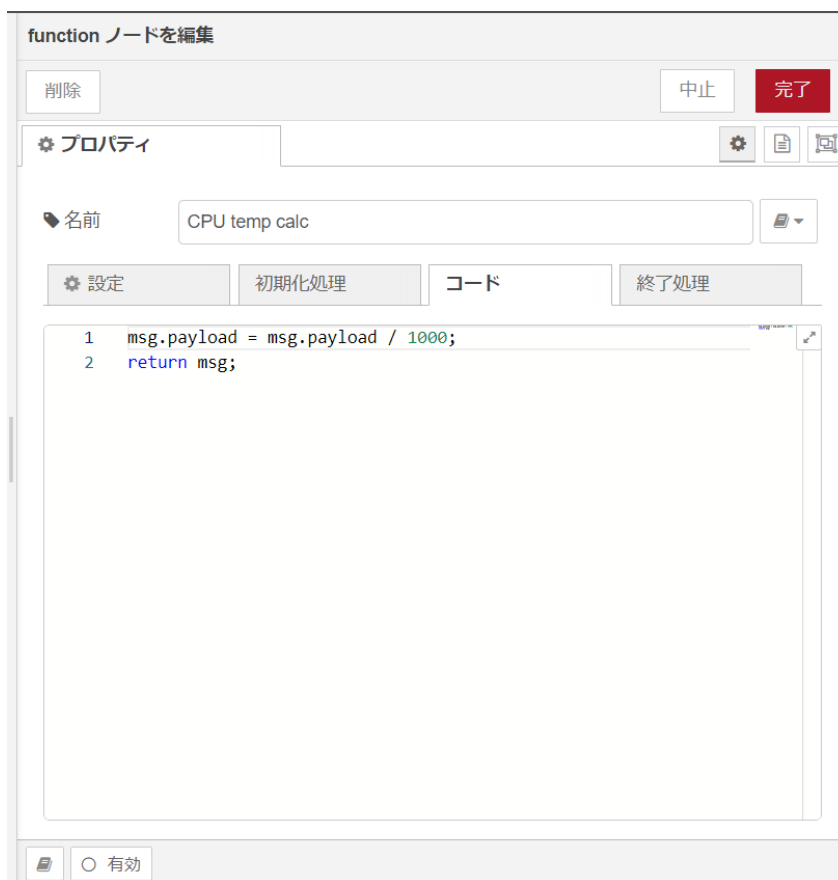


図 4.9 [function] ノードのプロパティ内容

8. [CPU temp] ノードの右側にある端子をクリックし、[CPU temp calc] ノードの左側の端子を選択して放します。
9. [debug] ノードをドラッグ、ドロップします。
10. [CPU temp calc] ノードの右側にある端子をクリックし、[debug] ノードの左側の端子を選択して放します。

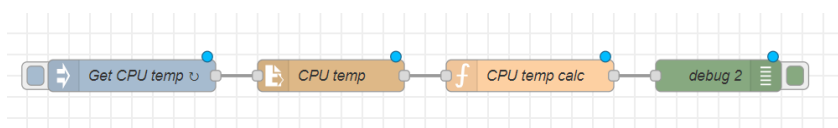


図 4.10 CPU の測定温度を 1 秒間隔で取得するフロー

11. 画面右上の [デプロイ] を押します。デバッグメッセージに 1 秒毎に CPU の測定温度が表示されます。

### 4.4.3. 接点入力の入力レベルを取得する

ここでは、接点入力の入力レベルを 1 秒間隔で取得するフローを作成します。Node-RED のコアノードのほかに GPIO in ノードを使用します。Armadillo-IoT ゲートウェイ A6E の CON6 にある接点入力



を使用します。入力レベル 0 は LOW レベル、1 は HIGH レベルを表わします。接点入力の場所については「3.3. インターフェースレイアウト」をご確認ください。

表 4.2 CON6 信号配列(接点入力関連)

ピン番号	ピン名	I/O	説明
3	COM	In	接点入力プラスコモン
4	DI1	In	接点入力 1
5	DI2	In	接点入力 2

DI ピンは GPIO として制御可能であり、対応する GPIO 番号を次に示します。

表 4.3 接点入力に対応する CON6 ピン番号

ピン番号	ピン名	GPIO チップ	GPIO 番号
4	DI1	gpiochip5	0
5	DI2	gpiochip5	1



接点入力に何も接続していない(開放状態)場合、取得できる入力レベルは 1 (HIGH レベル)となります。

接続に関する注意事項は以下です。

表 4.4 CON6 接続可能な電線

規格	UL	IEC
電線範囲	26~18 AWG	0.12~0.9mm <sup>2</sup>
被覆剥き長さ	5~6mm	
使用可能フェルル端子	型番 : MFL25-5BE メーカー : ミスミ	
推奨ねじ締めトルク	0.28Nm	



電線の先端に予備半田しないでください。正しい接続ができなくなります。



端子台に電線を接続する際、端子台に過度な力を加えないでください。端子台が破損する恐れがあります。

Armadillo-IoT ゲートウェイ A6E の接点入力インターフェースはフォトカプラで絶縁されています。そのため、フォトカプラを駆動させるために 5V 以上の電圧を印加する必要があります。外部電源を用いてフォトカプラを駆動させる場合の接続方法について以下に示します。

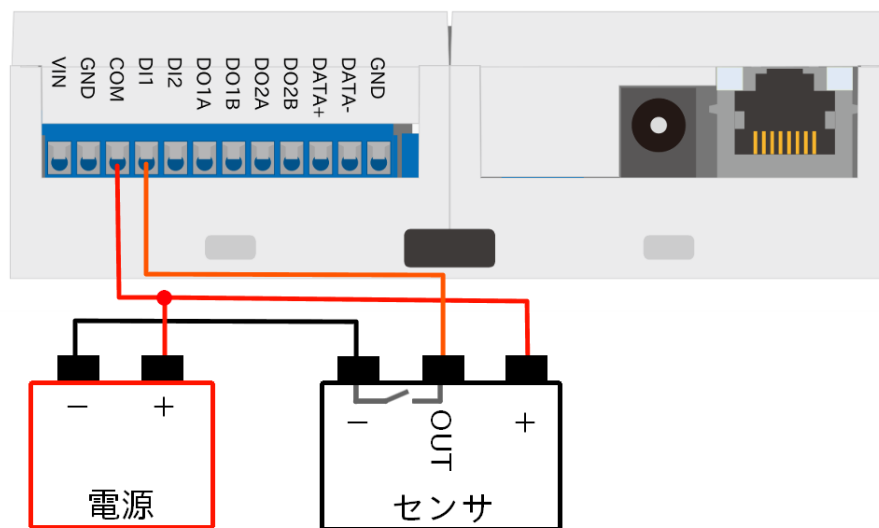


図 4.11 接点入力の接続方法(DI1 ピン)

例として DI1 ピンからの入力レベルを取得するフローを作成します。

1. パレットから [inject] ノードをワークスペースにドラッグ、ドロップします。
2. [inject] ノードのプロパティを編集します。[inject] は 2 秒間隔で実行します。

名前: Get DI1  
 繰り返し: 指定した時間間隔  
 時間間隔: 1 秒



図 4.12 [inject] ノードのプロパティ内容

3. [GPIO in] ノードをドラッグ、ドロップします。
4. [GPIO in] ノードのプロパティを編集します。

```

Device : gpiochip5
Type : Digital pin
Pin : 0
Name : DI1
    
```

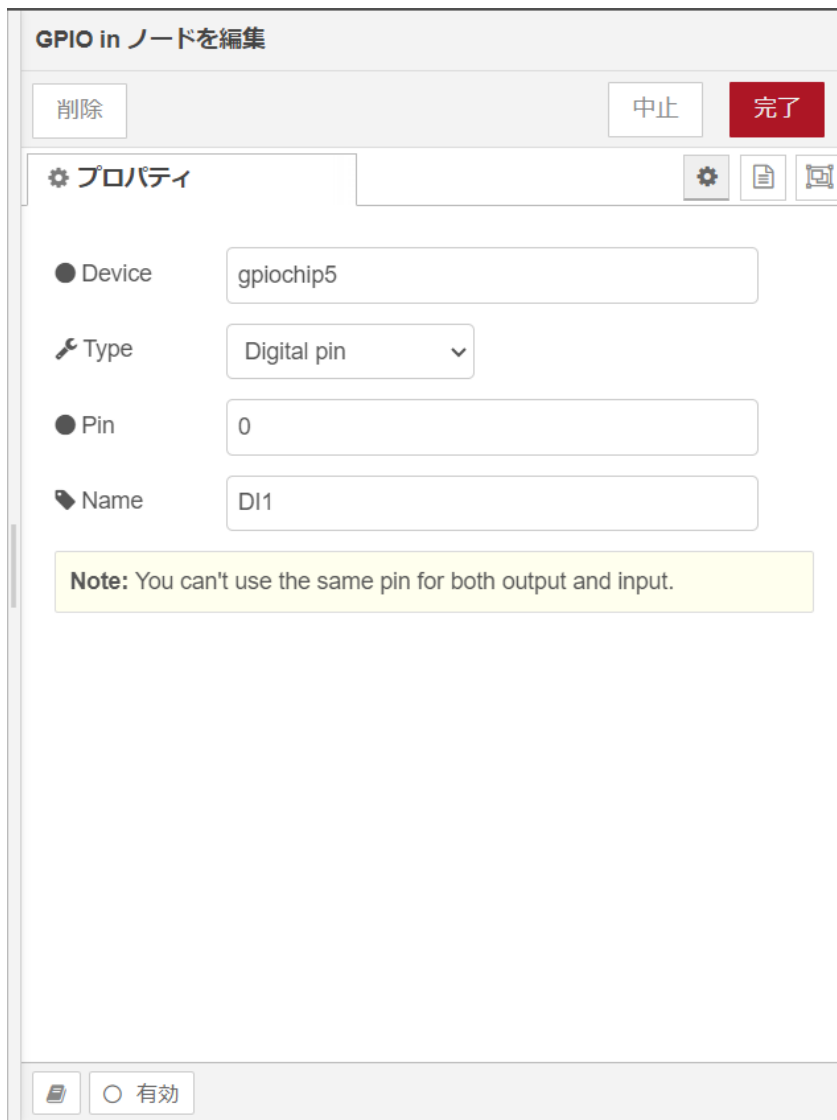


図 4.13 [GPIO in] ノードのプロパティ内容

5. [Get DI1] ノードの右側にある端子をクリックし、[DI1] ノードの左側の端子を選択して放します。
6. [debug] ノードをドラッグ、ドロップします。
7. [DI1] ノードの右側にある端子をクリックし、[debug] ノードの左側の端子を選択して放します。

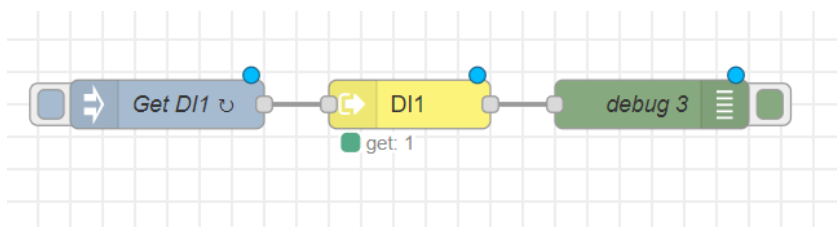


図 4.14 接点入力の入力レベルを 1 秒間隔で取得するフロー

8. 画面右上の [デプロイ] を押します。デバッグメッセージに 1 秒毎に DI1 ピンの入力レベルが表示されます。何も接続していない場合であれば常に 1 が表示されます。

### 4.4.4. 接点出力の出力レベルを制御する

ここでは、接点出力の出力レベルを1秒毎に HIGH レベルと LOW レベルを切り替えるフローを作成します。Node-RED のコアノードのほかに GPIO out ノードを使用します。Armadillo-IoT ゲートウェイ A6E の CON6 にある接点出力を使用します。出力レベル 0 は LOW レベル、1 は HIGH レベルを表わします。接点出力の場所については「3.3. インターフェースレイアウト」をご確認ください。

表 4.5 CON6 信号配列(接点出力関連)

ピン番号	ピン名	I/O	説明
6	DO1A	-	接点出力 1A
7	DO1B	-	接点出力 1B
8	DO2A	-	接点出力 2A
9	DO2B	-	接点出力 2B

DO ピンは GPIO として制御可能であり、対応する GPIO 番号を次に示します。

表 4.6 接点出力に対応する CON6 ピン番号

ピン番号	ピン名	GPIO チップ	GPIO 番号
6	DO1A	gpiochip5	2
7	DO1B	gpiochip5	2
8	DO2A	gpiochip5	3
9	DO2B	gpiochip5	3

接続に関する注意事項は以下です。

表 4.7 CON6 接続可能な電線

規格	UL	IEC
電線範囲	26~18 AWG	0.12~0.9mm <sup>2</sup>
被覆剥き長さ	5~6mm	
使用可能フェルール端子	型番 : MFL25-5BE メーカー : ミスミ	
推奨ねじ締めトルク	0.28Nm	



電線の先端に予備半田しないでください。正しい接続ができなくなります。



端子台に電線を接続する際、端子台に過度な力を加えないでください。端子台が破損する恐れがあります。

Armadillo-IoT ゲートウェイ A6E の接点出力インターフェースはフォトカプラで絶縁されています。そのため、フォトカプラを駆動させるために 5V 以上の電圧を印加する必要があります。外部電源を用いてフォトカプラを駆動させる場合の接続方法について以下に示します。

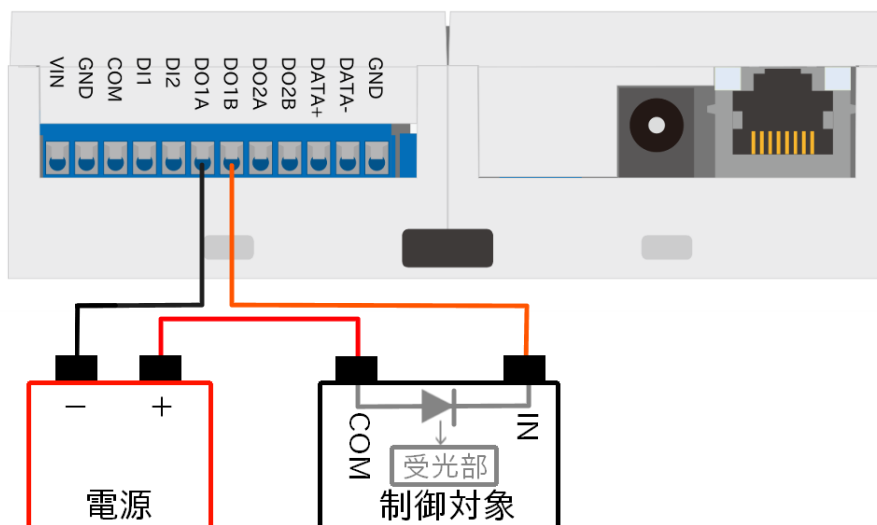


図 4.15 接点出力の接続方法(制御対象がプラスコモンの場合の DO1 ピン)

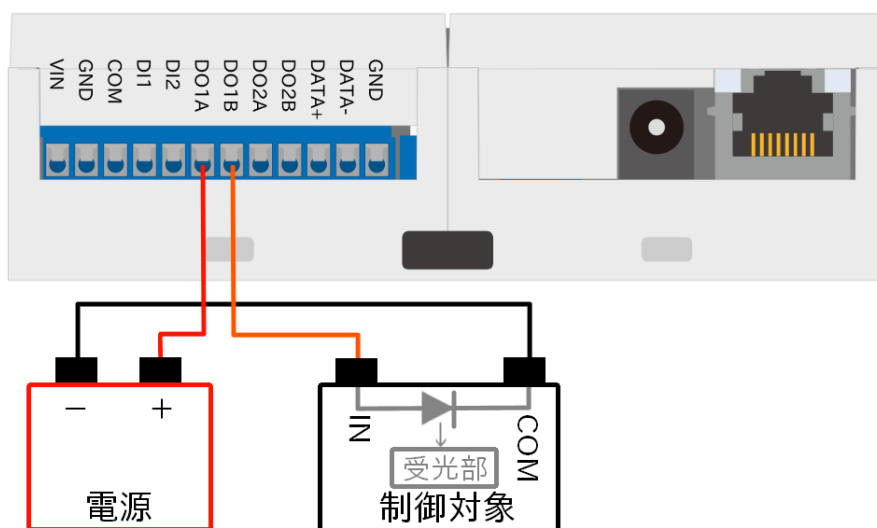


図 4.16 接点出力の接続方法(制御対象がマイナスコモンの場合の DO1 ピン)

例として DO1 ピンの出力レベルを制御するフローを作成します。

1. パレットから [inject] ノードをワークスペースにドラッグ、ドロップします。
2. [inject] ノードのプロパティを編集します。[inject] は 2 秒間隔で実行します。

```

名前: Set D01
繰り返し: 指定した時間間隔
時間間隔: 2 秒
    
```

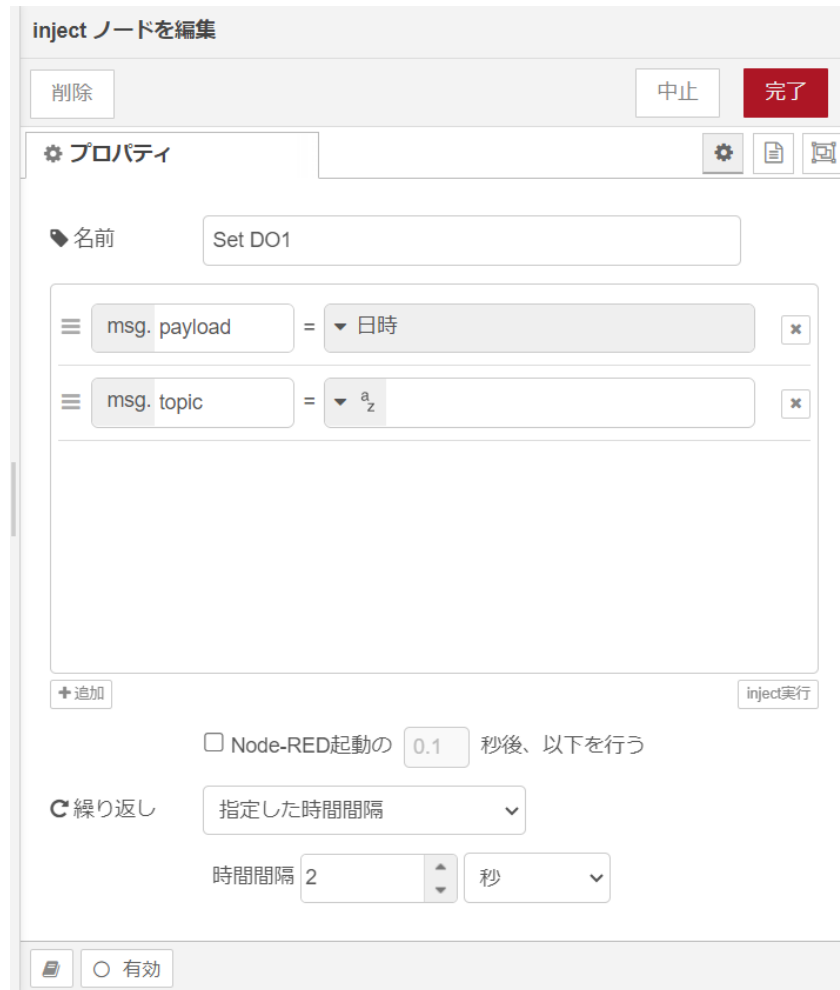


図 4.17 [inject] ノードのプロパティ内容

3. [trigger] ノードをドラッグ、ドロップします。
4. [trigger] ノードのプロパティを編集します。

送信データ : 1  
 送信後の処理 : 指定した時間待機  
 1 秒  
 再送信データ : 0



図 4.18 [trigger] ノードのプロパティ内容

5. [GPIO out] ノードをドラッグ、ドロップします。
6. [GPIO out] ノードのプロパティを編集します。

```
Device : gpiochip5
Type : Digital (0/1)
Pin: 2
Name : D01
```



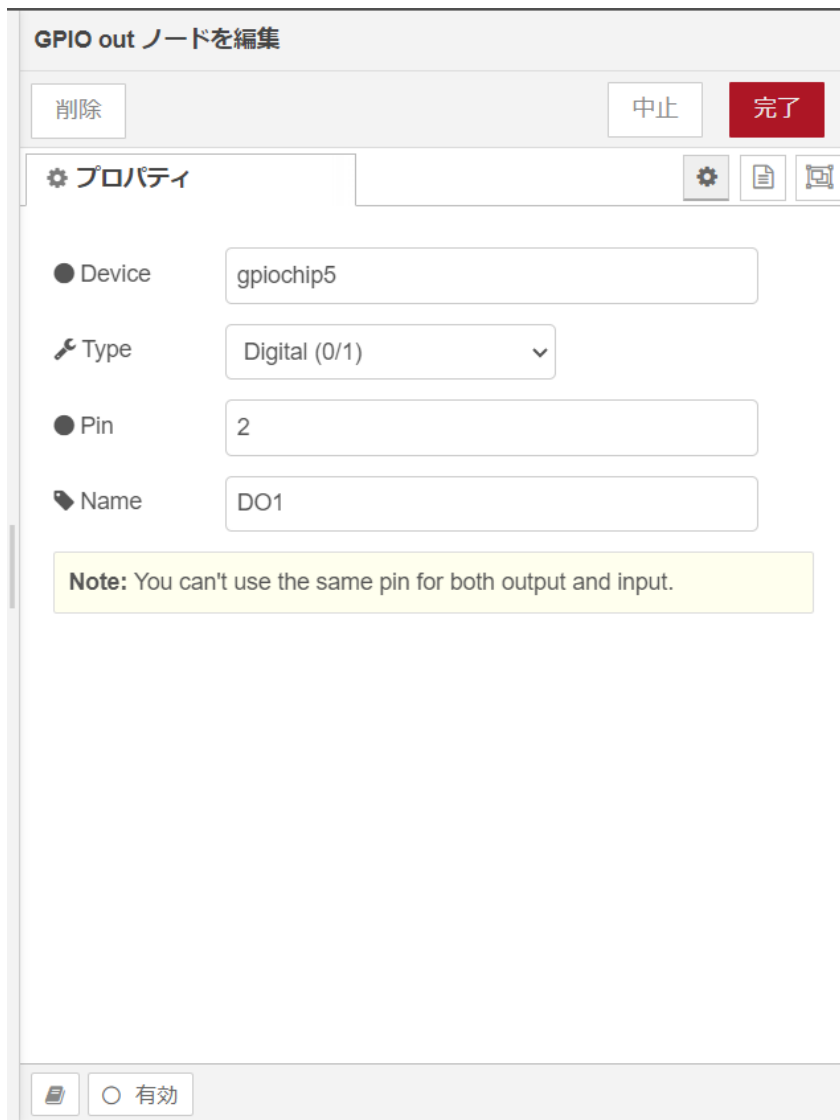


図 4.19 [GPIO out] ノードのプロパティ内容

7. [Set DO1] ノードの右側にある端子をクリックし、[DO1] ノードの左側の端子を選択して放します。
8. [debug] ノードをドラッグ、ドロップします。
9. [DO1] ノードの右側にある端子をクリックし、[debug] ノードの左側の端子を選択して放します。

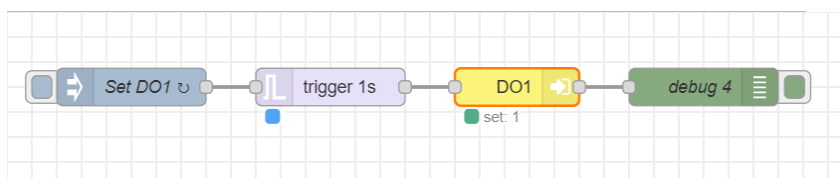


図 4.20 接点出力の出力レベルを 1 秒毎に HIGH レベルと LOW レベルを切り替えるフロー

10. 画面右上の [デプロイ] を押します。1 秒毎に DO1 ピンの出力レベルが HIGH レベルと LOW レベルで切り替わります。


### 4.4.5. 接点入力の入力レベルの立ち上がりエッジを検出する

ここでは、接点入力の入力レベルの立ち上がりエッジを検出するフローを作成します。Node-RED のコアノードのほかに rising edge ノードを使用します。Armadillo-IoT ゲートウェイ A6E の CON6 にある接点入力と接点出力を使用します。入力レベル、出力レベルの 0 は LOW レベル、1 は HIGH レベルを表わします。接点入力、接点出力の場所については「3.3. インターフェースレイアウト」をご確認ください。


例として DI2 ピンの立ち上がりエッジを取得するフローを作成します。また、立ち上がりエッジ検出のためにループバック接続を行います。接続に関する注意事項は以下です。

表 4.8 CON6 接続可能な電線

規格	UL	IEC
電線範囲	26~18 AWG	0.12~0.9mm <sup>2</sup>
被覆剥き長さ	5~6mm	
使用可能フェルルール端子	型番 : MFL25-5BE メーカー : ミスミ	
推奨ねじ締めトルク	0.28Nm	



電線の先端に予備半田しないでください。正しい接続ができなくなります。



端子台に電線を接続する際、端子台に過度な力を加えないでください。端子台が破損する恐れがあります。

CON6 にある接点入力、接点出力のピンを次に示します。

表 4.9 CON6 信号配列(接点入力、出力関連)

ピン番号	ピン名	I/O	説明
1	VIN	In	入力電圧
2	GND	In	GND
3	COM	In	接点入力プラスコモン
4	DI1	In	接点入力 1
5	DI2	In	接点入力 2
6	DO1A	-	接点出力 1A
7	DO1B	-	接点出力 1B
8	DO2A	-	接点出力 2A
9	DO2B	-	接点出力 2B

DI ピンと DO ピンは GPIO として制御可能であり、対応する GPIO 番号を次に示します。

表 4.10 接点入力、出力に対応する CON6 ピン番号

ピン番号	ピン名	GPIO チップ	GPIO 番号
4	DI1	gpiochip5	0
5	DI2	gpiochip5	1

ピン番号	ピン名	GPIO チップ	GPIO 番号
6	DO1A	gpiochip5	2
7	DO1B	gpiochip5	2
8	DO2A	gpiochip5	3
9	DO2B	gpiochip5	3

Armadillo-IoT ゲートウェイ A6E の接点入力、出力インターフェースはフォトカプラで絶縁されています。そのため、フォトカプラを駆動させるために 5V 以上の電圧を印加する必要があります。

ループバック接続を用いて DO2 ピンの出力レベルを変更して、DI2 ピンで立ち上がりエッジを検出します。必要なピン接続は以下です。

- ・ VIN ピン - COM ピン
- ・ GND ピン - DO2B ピン
- ・ DI2 ピン - DO2A ピン

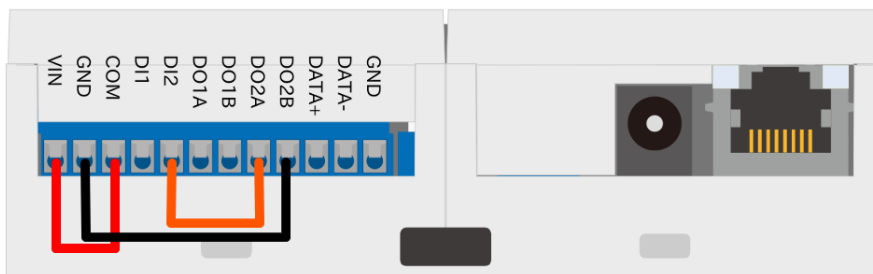


図 4.21 DI2 ピンの立ち上がりエッジを取得するための接続図

1. はじめに DI2 ピンの立ち上がりエッジを検出するフローを作成します。パレットから [inject] ノードをワークスペースにドラッグ、ドロップします。
2. [inject] ノードのプロパティを編集します。

名前: Get DI2  
 繰り返し: 指定した時間間隔  
 時間間隔: 0.1 秒



図 4.22 [inject] ノードのプロパティ内容

3. [GPIO in] ノードをドラッグ、ドロップします。
4. [GPIO in] ノードのプロパティを編集します。

```

Device : gpiochip5
Type : Digital pin
Pin: 1
Name : DI2
    
```

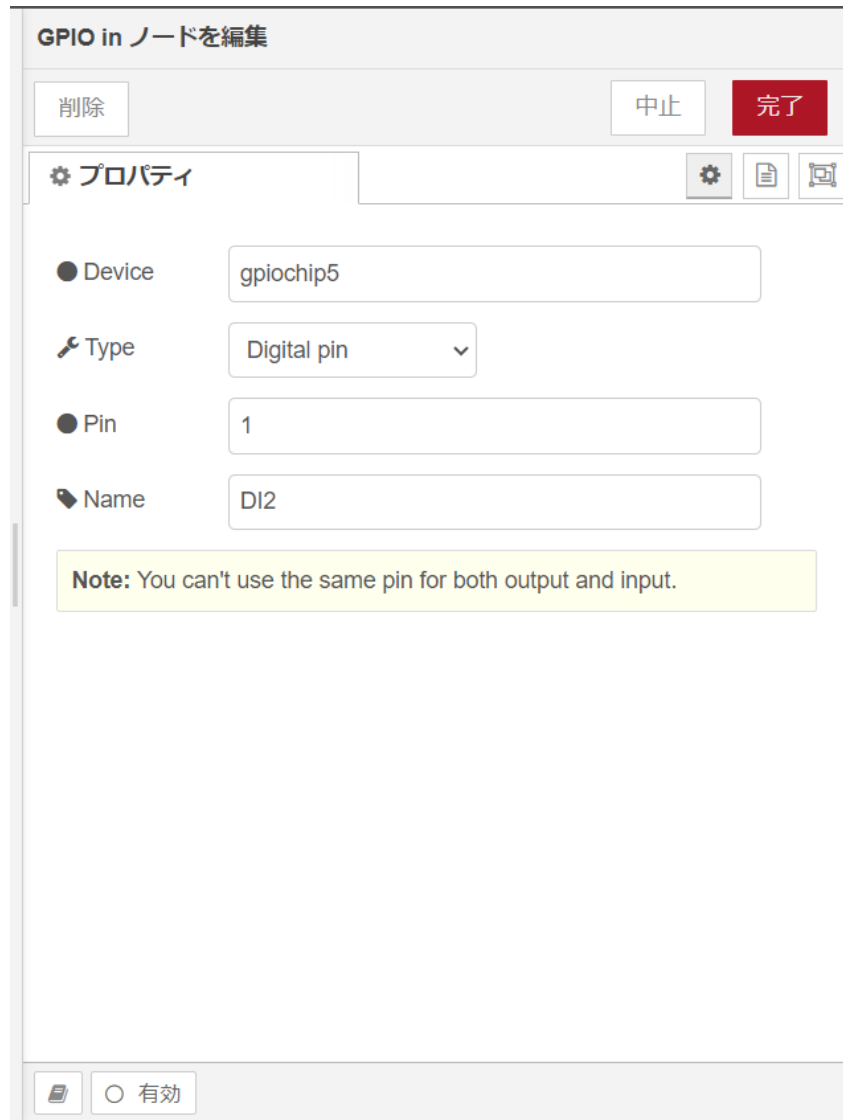


図 4.23 [GPIO in] ノードのプロパティ内容

5. [Get DI2] ノードの右側にある端子をクリックし、[DI2] ノードの左側の端子を選択して放します。
6. [rising edge] ノードをドラッグ、ドロップします。
7. [rising edge] ノードのプロパティを編集します。

```
Name: notify rising edge
Threshold: 0
```

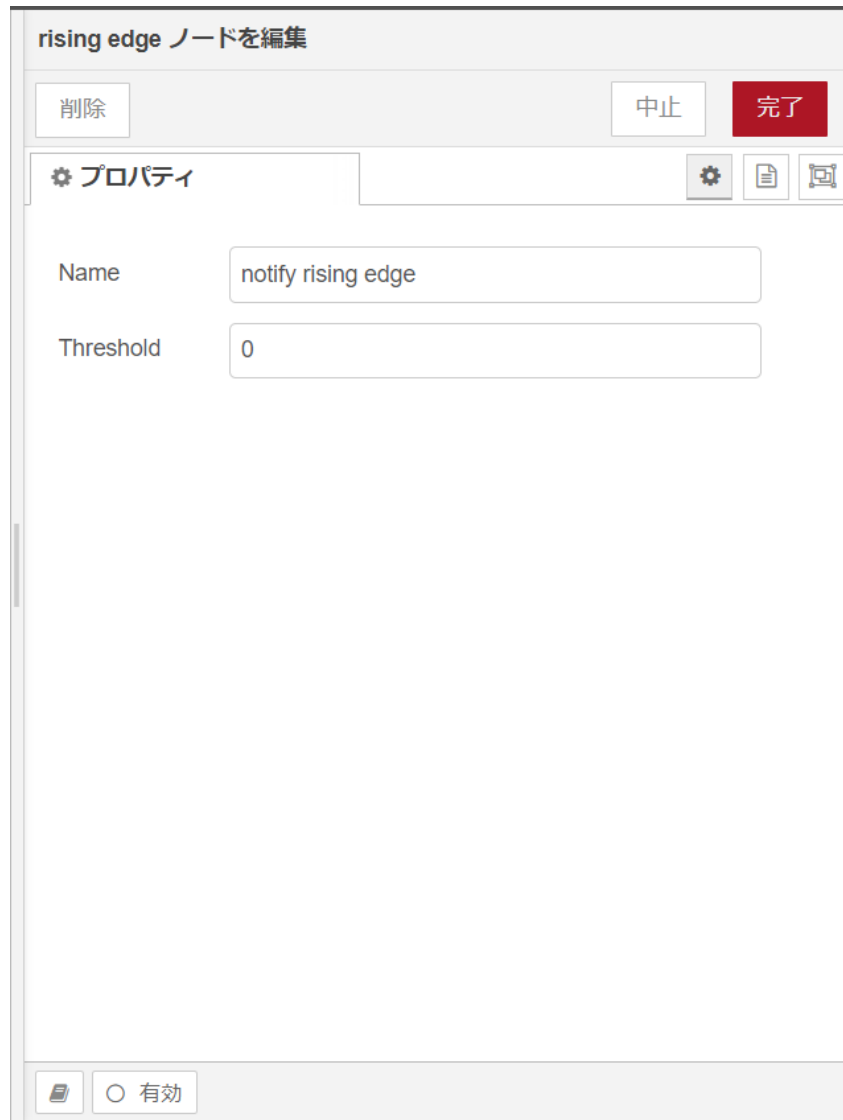


図 4.24 [rising edge] ノードのプロパティ内容

8. [DI2] ノードの右側にある端子をクリックし、[notify rising edge] ノードの左側の端子を選択して放します。
9. [change] ノードをドラッグ、ドロップします。
10. [change] ノードのプロパティを編集します。

名前: notification  
 ルール: 値の置換 msg.payload  
 検索する文字列: 1  
 置換後の文字列: notify!

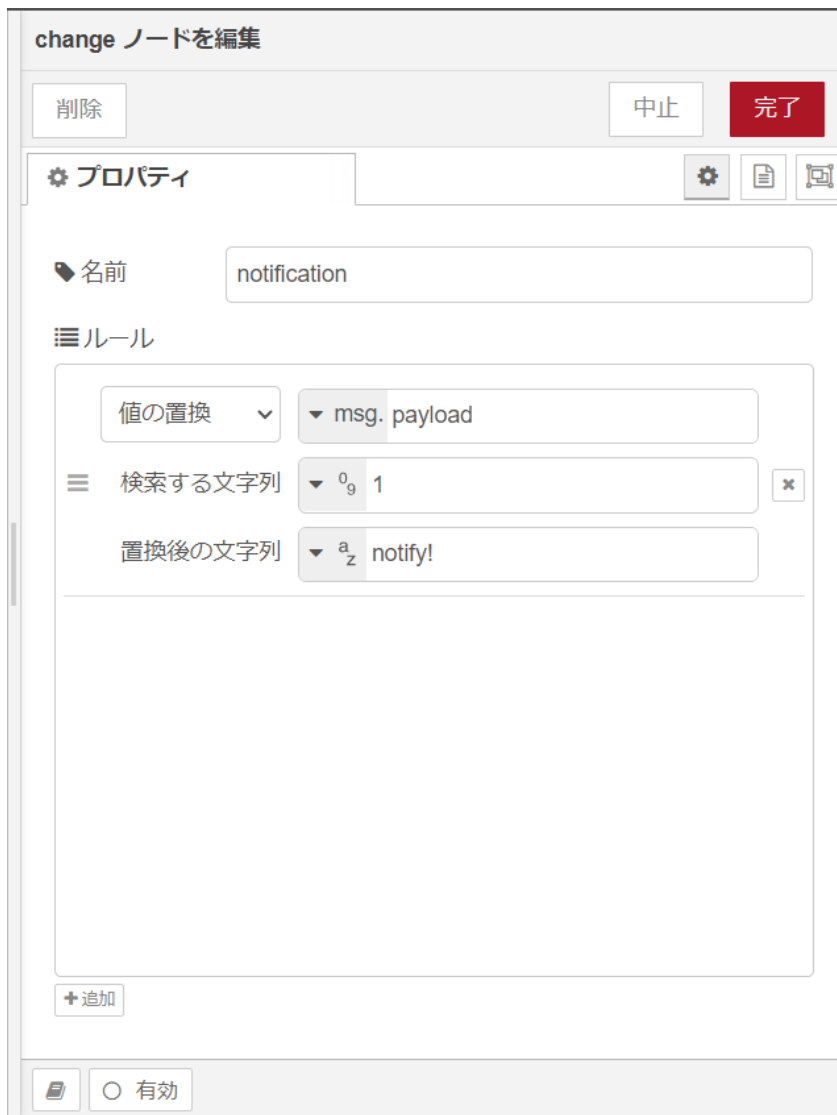


図 4.25 [change] ノードのプロパティ内容

11. [notify rising edge] ノードの右側にある端子をクリックし、[notification] ノードの左側の端子を選択して放します。
12. [debug] ノードをドラッグ、ドロップします。
13. [notification] ノードの右側にある端子をクリックし、[debug] ノードの左側の端子を選択して放します。
14. 次に DO2 ピンの出力レベルを変更するフローを追加します。[Get DI2] の下に [inject] ノードをドラッグ、ドロップします。
15. [inject] ノードのプロパティを編集します。

名前: Output LOW  
 msg.payload: 0  
 繰り返し: なし



図 4.26 [inject] ノードのプロパティ内容

16. [Output LOW] の下に [inject] ノードをドラッグ、ドロップします。
17. [inject] ノードのプロパティを編集します。

名前: Output HIGH  
 msg.payload: 1  
 繰り返し: なし





図 4.27 [inject] ノードのプロパティ内容

18. [GPIO out] ノードをドラッグ、ドロップします。
19. [GPIO out] ノードのプロパティを編集します。

```

Device : gpiochip5
Type : Digital (0/1)
Pin : 3
Name : D02
    
```

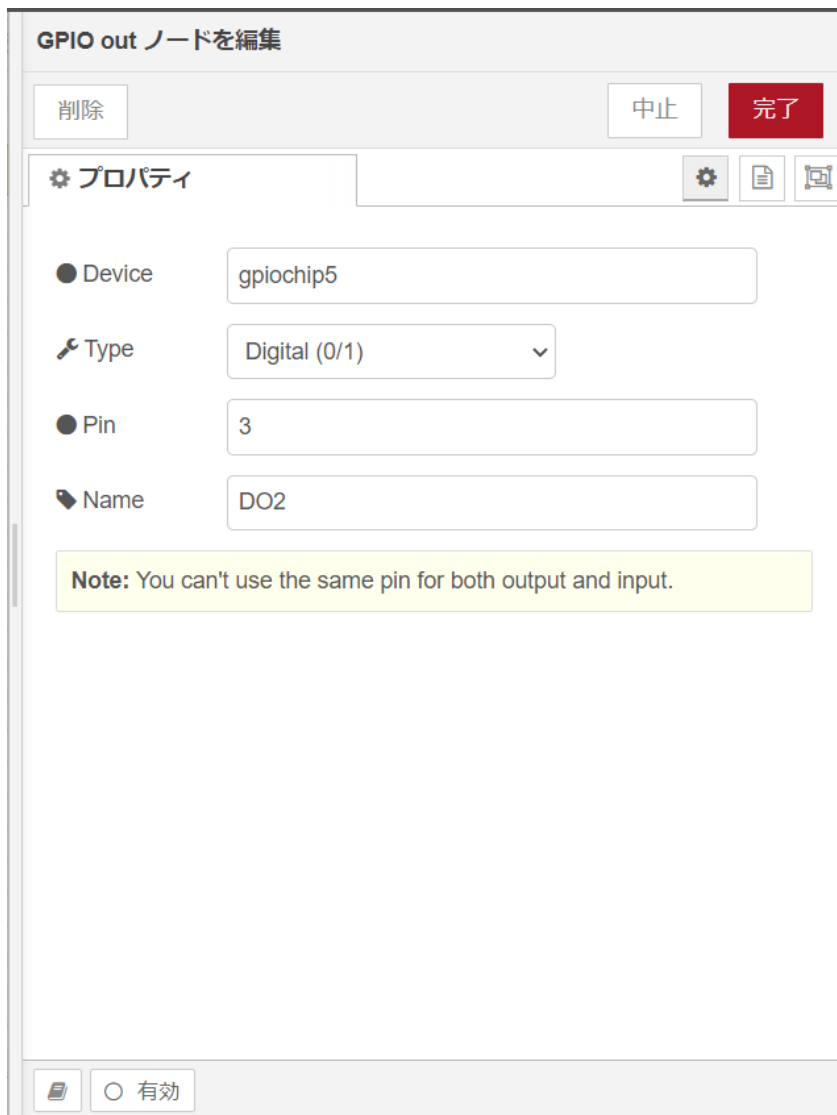


図 4.28 [GPIO out] ノードのプロパティ内容

20. [Output LOW] ノードの右側にある端子をクリックし、[DO2] ノードの左側の端子を選択して放します。
21. [Output HIGH] ノードの右側にある端子をクリックし、[DO2] ノードの左側の端子を選択して放します。[DO2] ノードの左側は二つのノードが繋がっている状態になります。

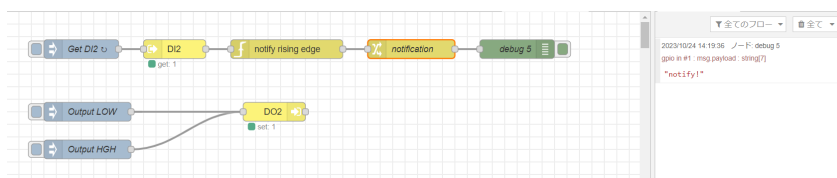


図 4.29 接点入力の入力レベルの立ち上がりエッジを検出するフローとデバッグメッセージ

22. 画面右上の [デプロイ] を押します。[Output LOW] を押した後に [Output HIGH] を押した場合に、立ち上がりエッジを検出して "notify!" がサイドバーのデバッグメッセージに表示されます。

### 4.4.6. RS-485 modbus RTU 読み出しを行う

ここでは、RS-485 を使用した Modbus RTU 読み出し用フローを作成します。Node-RED のコアノードのほかに modbus-client ノード、Modbus-Read ノード、 modbus-response ノードを使用します。Armadillo-IoT ゲートウェイ A6E の CON6 にある RS485 を使用します。RS-485 シリアルインターフェースのデバイスファイルは、 /dev/ttyxc4 を使用します。RS-485 の場所については「3.3. インターフェースレイアウト」をご確認ください。

表 4.11 CON6 信号配列(RS-485 関連)

ピン番号	ピン名
10	DATA+
11	DATA-
12	GND

- 機能
- ・ 最大データ転送レート：5Mbps
  - ・ 半二重対応
  - ・ RS-485 シリアルインターフェースのデバイスファイルは、 /dev/ttyxc4 を使用します。

終端抵抗 120Ω の ON/OFF をスイッチ (SW3) で切り替えることができます、設定方法は以下をご確認ください。

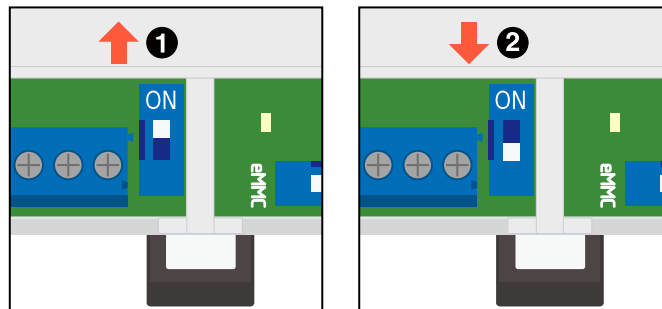


図 4.30 スイッチの状態と終端抵抗の ON/OFF

- ① 終端抵抗 120Ω が ON になります。
- ② 終端抵抗 120Ω が OFF になります。


終端は RS485 の信号線の最遠端で行います。Armadillo-IoT ゲートウェイ A6E が最遠端になる場合は終端抵抗を ON にしてください。

接続に関する注意事項は以下です。


表 4.12 CON6 接続可能な電線

規格	UL	IEC
電線範囲	26~18 AWG	0.12~0.9mm <sup>2</sup>

被覆剥き長さ	5~6mm
使用可能フェール端子	型番：MFL25-5BE メーカー：ミスミ
推奨ねじ締めトルク	0.28Nm



電線の先端に予備半田しないでください。正しい接続ができなくなります。



端子台に電線を接続する際、端子台に過度な力を加えないでください。端子台が破損する恐れがあります。

RS-485 通信対応デバイスとの接続方法について以下に示します。

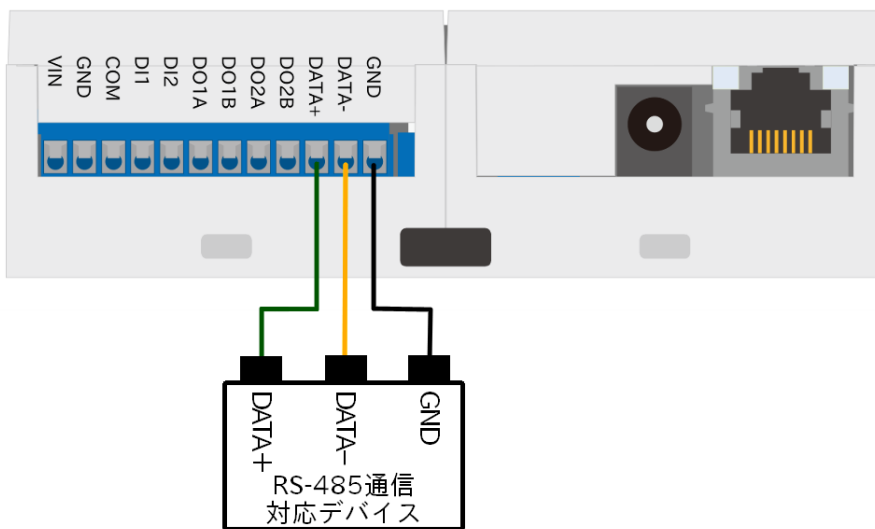


図 4.31 RS-485 通信対応デバイスとの接続方法

今回は以下の RS-485 通信対応デバイスを想定した場合の設定内容となります。実際に動作確認する場合は、使用する RS-485 通信対応デバイスの設定に変更してください。

```

ユニット ID: 1
通信プロトコル: Modbus RTU
ボーレート: 9600
読み出しアドレス: 0x00
ファンクションコード: 1
    
```

1. パレットから [Modbus-Read] ノードをワークスペースにドラッグ、ドロップします。
2. [Modbus-Read] ノードのプロパティを編集します。

- 先に Server を設定する必要があります。[新規に modbus-client を追加] の右隣の編集ボタンを押して [modbus-client] を作成します。

```

名前: RS485 slave device
Type: Serial
Serial port: /dev/ttyxc4
Serial type: RTU
Baud rate: 9600
Unit-Id: 1
    
```

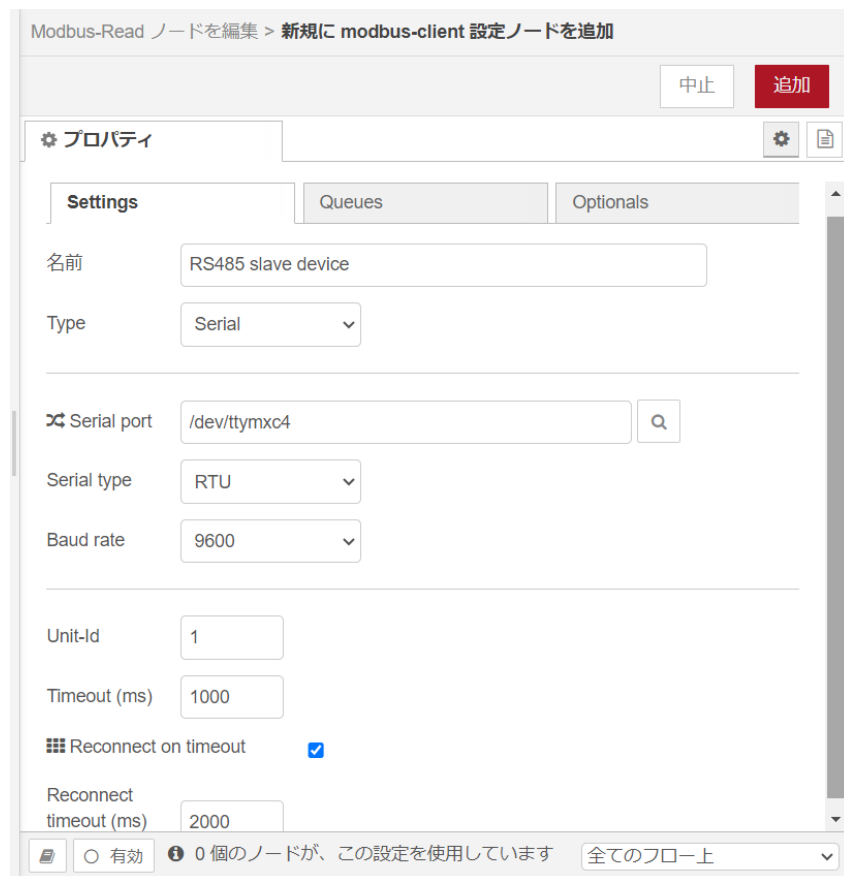


図 4.32 [modbus-client] ノードのプロパティ内容

- [Modbus-Read] ノードのプロパティを編集します。

```

名前: slave device
トピック:
Unit-Id:
FC: FC 1:Read Coil Status
Address: 0
Quantity: 1
Poll Rate: 1 Second(s)
Server: RS485 slave device
    
```

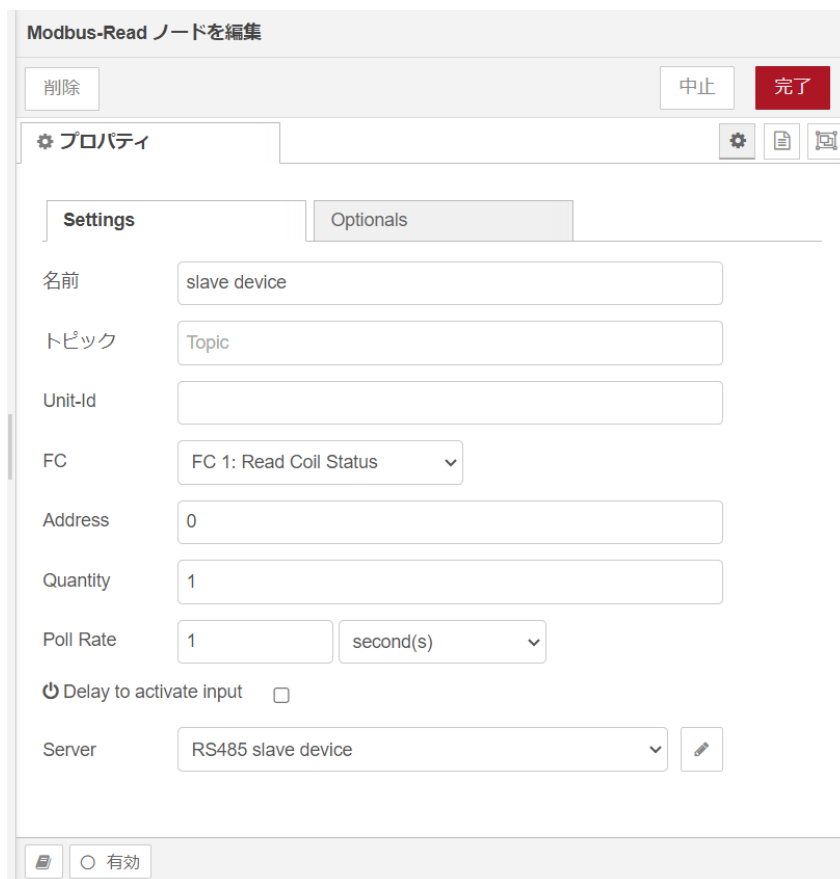


図 4.33 [Modbus-Read] ノードのプロパティ内容

5. [Modbus-Response] ノードをドラッグ、ドロップします。
6. [slave device] ノードの右側にある端子をクリックし、[Modbus-Response] ノードの左側の端子を選択して放します。

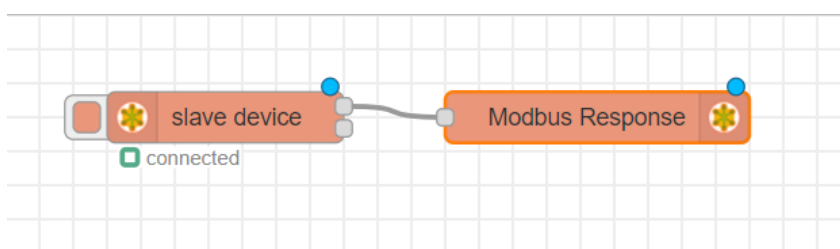


図 4.34 RS-485 を使用した Modbus RTU 読み出し用フロー

7. 画面右上の [デプロイ] を押します。読み出しに成功すると 1 秒毎に読みだした値が更新され [Modbus-Response] ノードの下に表示されます。

#### 4.4.7. CPU の測定温度のグラフをダッシュボードに表示する

「4.4.2. CPU の測定温度を取得する」 で取得した CPU 温度をダッシュボードにグラフとして表示するフローを作成します。

1. サイドバーの右端にある三角形のボタンを押し、[ダッシュボード] を選択します。

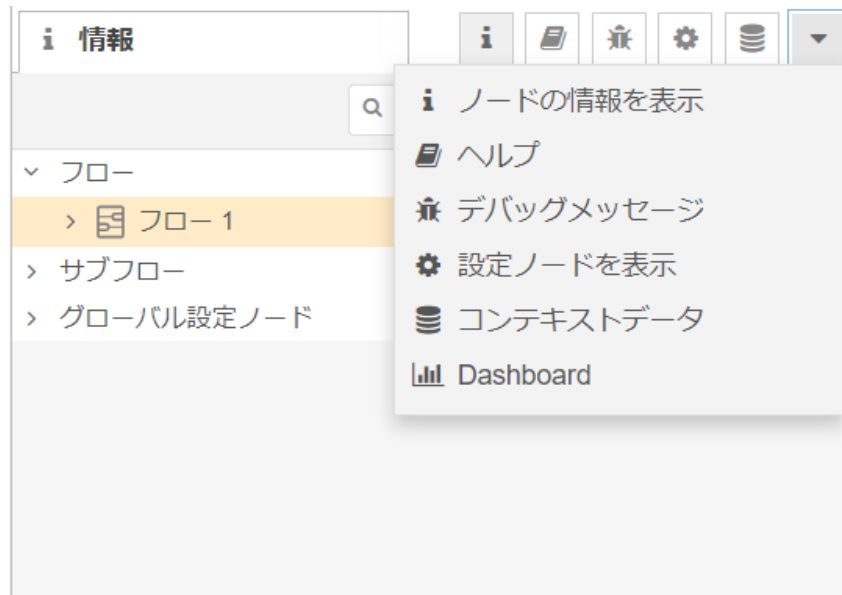


図 4.35 [ダッシュボード]を選択する

2. [ダッシュボード] 編集画面が表示されることを確認します。



図 4.36 ダッシュボード編集画面

3. [+タブ] ボタンを押して [Tab1] を追加します。



図 4.37 ダッシュボードに [Tab1] を追加

4. [Tab1] プロパティを編集します。

名前: Armadillo アイコン: dashboard
----------------------------------





図 4.38 [Tab1] プロパティ内容

5. [Armadillo] タブが追加されました。

名前: Armadillo  
 アイコン: dashboard



図 4.39 ダッシュボード編集画面に [Armadillo] タブが追加

6. [Armadillo] 横の [+グループ] ボタンを押して [Group1] を追加します。

名前: Armadillo  
 アイコン: dashboard



図 4.40 ダッシュボード編集画面に [Group1] グループが追加

7. [Group1] プロパティを編集します。

名前: Group 1 タブ: Armadillo 種類: 幅: 15 <input type="checkbox"/> グループ名を表示する
---



dashboard group ノードを編集

削除 中止 更新

プロパティ

名前 Group 1

タブ Armadillo

種類 ウィジェット用のCSSクラス名 (オプション)

幅 15

グループ名を表示する

1 個のノードが、この設定を使用しています

図 4.41 [Group1] プロパティ内容

- [chart] ノードをワークスペースにドラッグ、ドロップします。ダッシュボード編集画面の [Group1] グループに追加されたことを確認します。



図 4.42 ダッシュボード編集画面に [chart] ノードが追加

9. [chart] ノードのプロパティを編集します。

グループ: [Armadillo] Group 1  
サイズ: 自動  
ラベル: CPU temp  
種類: 折れ線グラフ  
X 軸: 直近 1 時間  
X 軸ラベル: HH:mm:ss  
Y 軸: 最小 20 最大 50  
凡例: 非表示 補完: 直線



図 4.43 [inject] ノードのプロパティ内容

10. [CPU temp calc] ノードの右側にある端子をクリックし、[CPU temp] ノードの左側の端子を選択して放します。

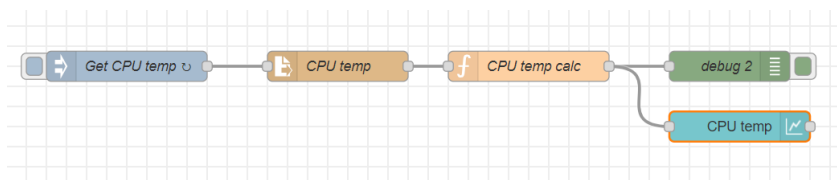


図 4.44 CPU の測定温度のグラフをダッシュボードに表示するフロー

11. 画面右上の [デプロイ] を押します。ダッシュボード編集画面の [テーマ] タブの右側にある四角に矢印が重なったボタンを選択すると、ダッシュボードが表示されます。

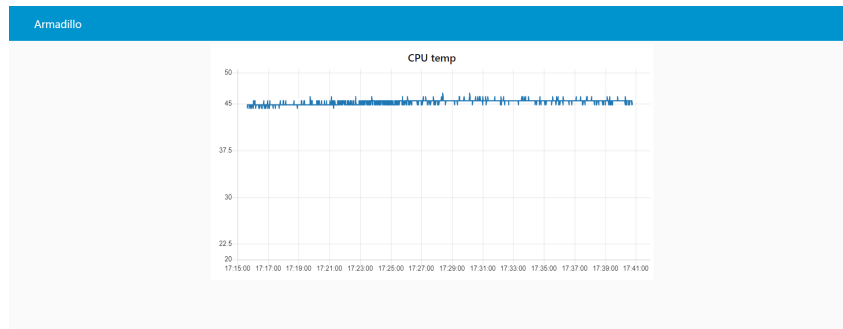


図 4.45 CPU の測定温度のグラフのダッシュボード

12. ダッシュボードの CPU 温度のグラフは一秒毎に更新されます。

### 4.4.8. Node-RED を終了する

ここでは、Node-RED を任意のタイミングで終了するフローを作成します。Node-RED のコアノードのほかに exit ノードを使用します。



Node-RED 終了後、Node-RED を再起動するためには Armadillo-IoT ゲートウェイ A6E の電源を再投入する必要があります。

1. パレットから [inject] ノードをワークスペースにドラッグ、ドロップします
2. [inject] ノードのプロパティを編集します。

名前: Finish Node-RED  
 繰り返し: なし



図 4.46 [inject] ノードのプロパティ内容

3. [exit] ノードをドラッグ、ドロップします。
4. [exit] ノードのプロパティを編集します。

```
Name: exit
Exit code: 0
```



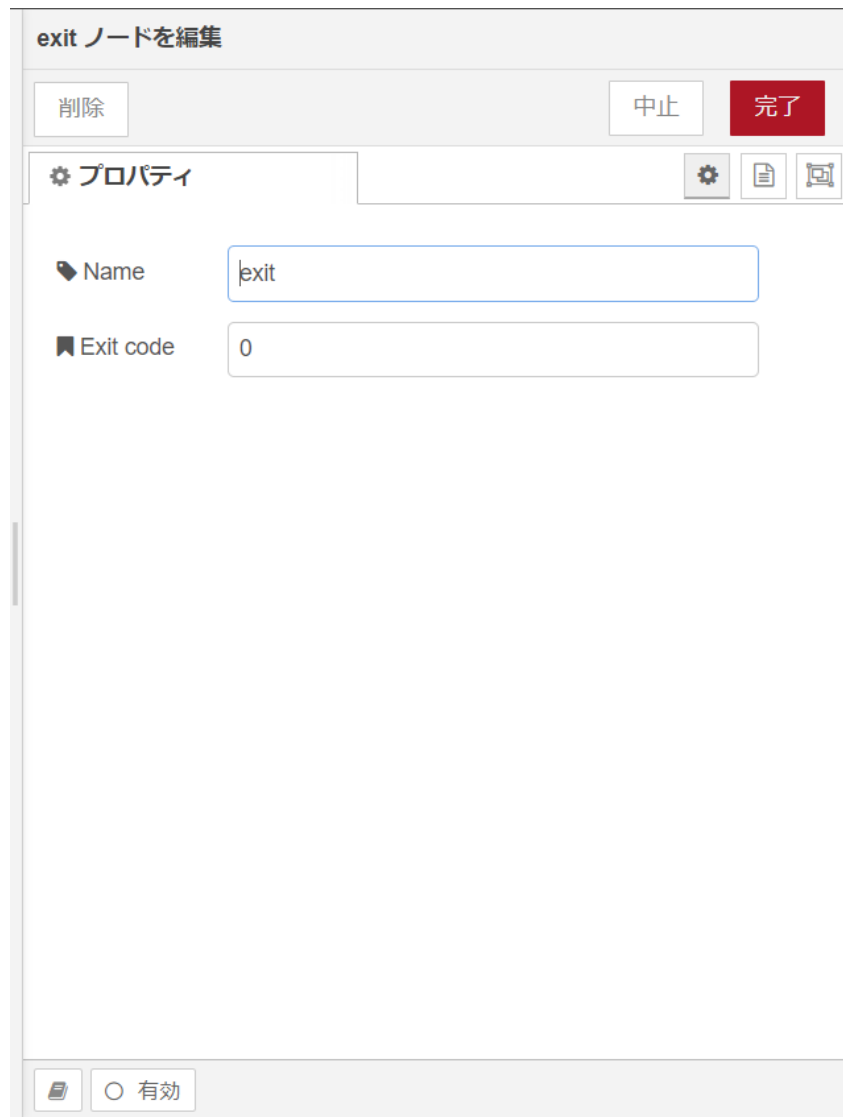


図 4.47 [exit] ノードのプロパティ内容

5. [Finish Node-RED] ノードの右側にある端子をクリックし、[exit] ノードの左側の端子を選択して放します。

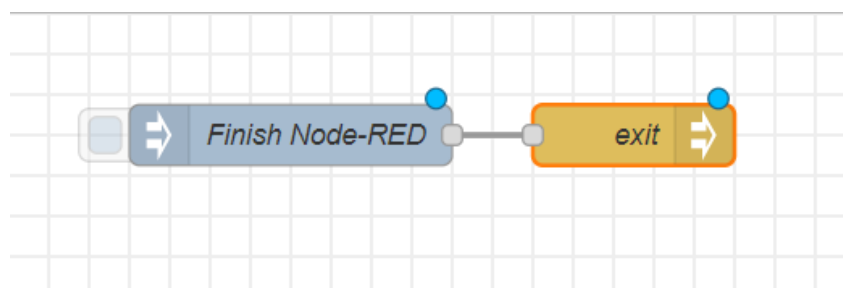


図 4.48 Node-RED を任意のタイミングで終了するフロー

6. 画面右上の [デプロイ] を押します。[Finish Node-RED] ノードの左側にあるボタンを押すと Node-RED が終了します。

## 4.5. ユーザデータを削除する

Node-RED に関するユーザデータは `/var/app/rollback/volumes/node-red/root` に保存されます。当該ディレクトリを削除した場合はユーザデータは全て削除されます。この場合 Node-RED のブラウザからインストールしたカスタムノードは削除されます。

## 改訂履歴

バージョン	年月日	改訂内容
1.0.0	2023/10/26	・ 初版発行
1.0.1	2023/12/26	・ 「4.4.7. CPU の測定温度のグラフをダッシュボードに表示する」を追加

