RS-WSUHA

Node-redでBルートプログラミング

Rev.1.0
Date: 2023/05/09

RATOC Systems

● 準備するもの

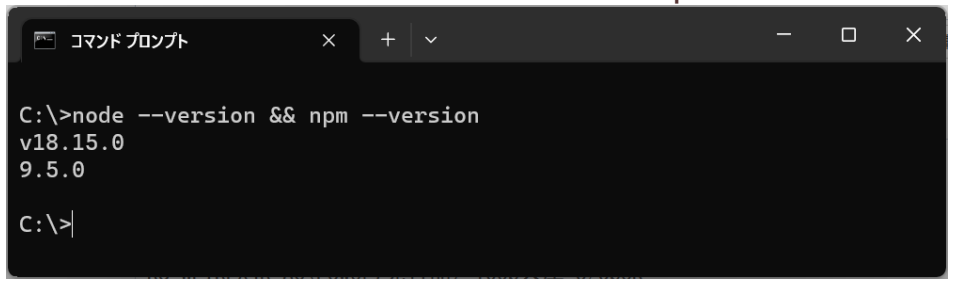
1.Windows PC、下記はWindows11での説明になります。

2.Node.jsのインストール

Node.jsの最新版: 18.15.0 LTSをNode.js公式ホームページからダウンロード

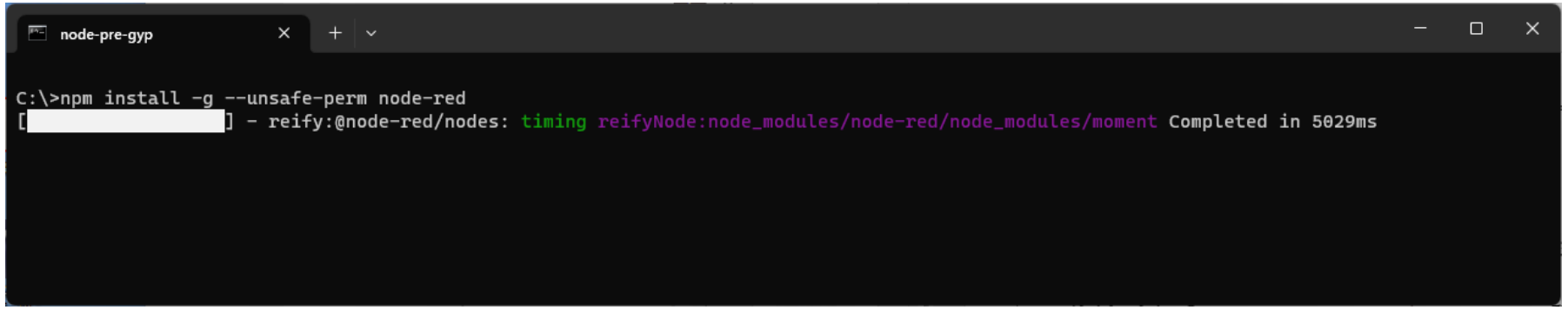
https://nodejs.org/ja

コマンドプロントから“node --version && npm --version”を起動して下記のように表示されることを確認してください。



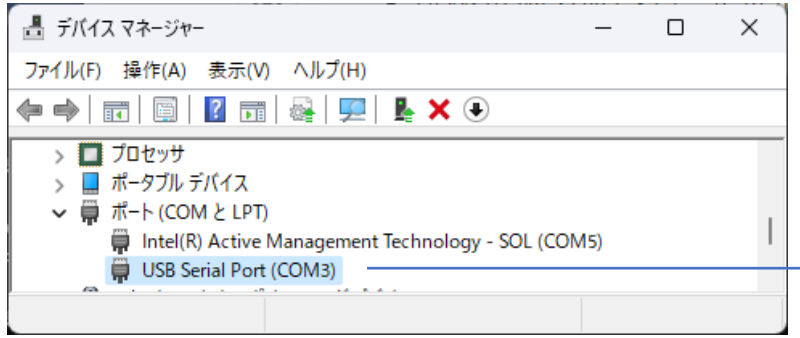
3.Node-REDのインストール

コマンドプロントから“npm install -g --unsafe-perm node-red”を起動します。



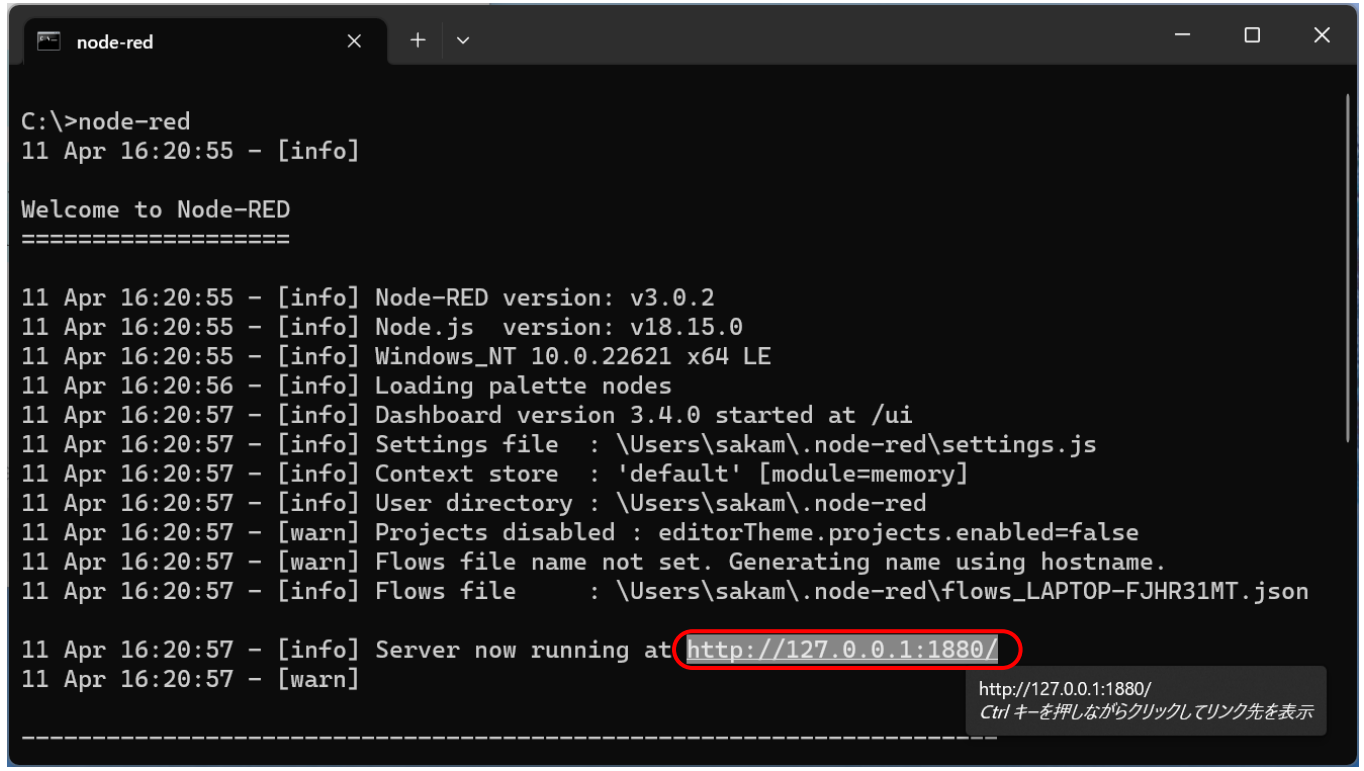
インストールが完了すれば、Node-REDを実行する準備が整いました。

4.WSUHAをPCに接続
USBポートにRS-WSUHA Wi-SUN USBアダプタ（以下、WSUHA）を接続するとインストールは自動的に完了します。デバイスマネージャを起動してアサインされたCOMポート番号を確認します。

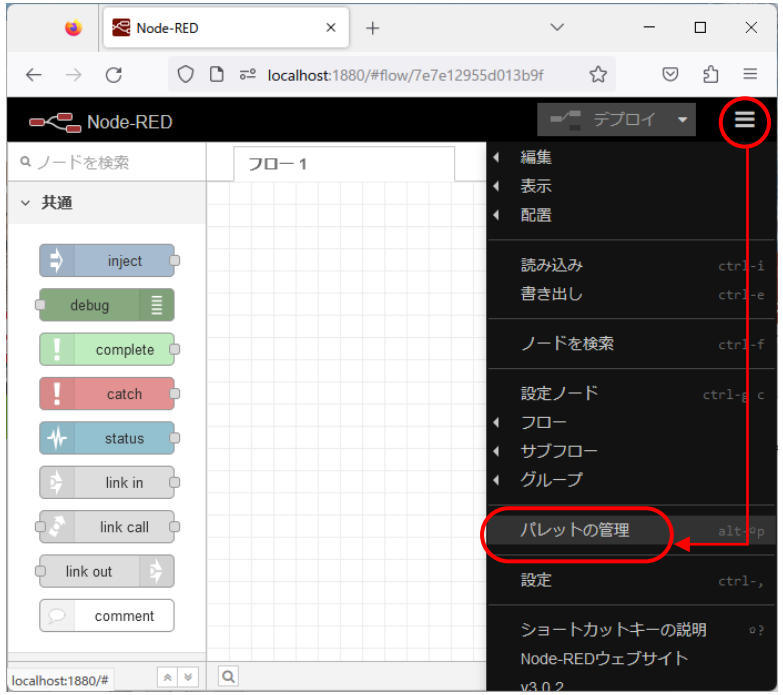


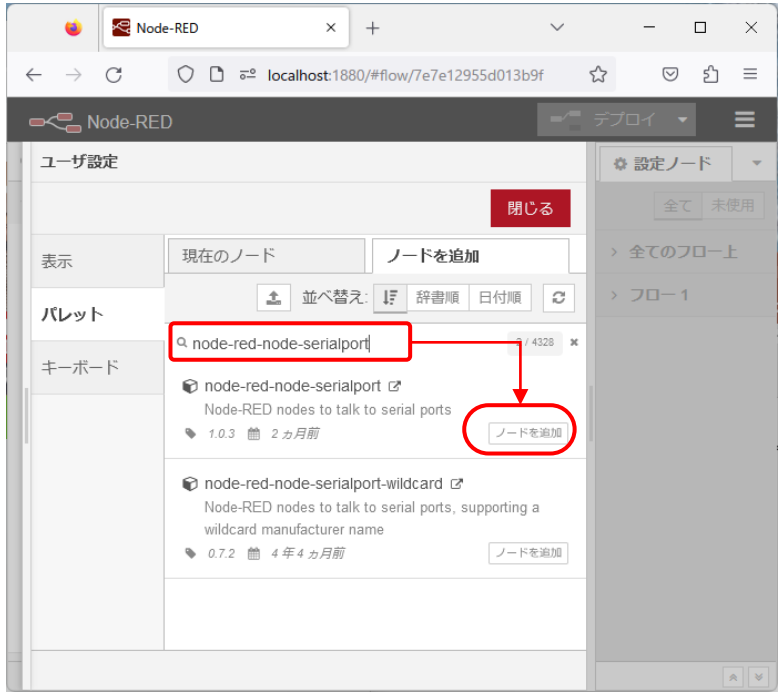
5.Node-redを起動
コマンドプロンプトから右図のように、”node-red”を起動します。

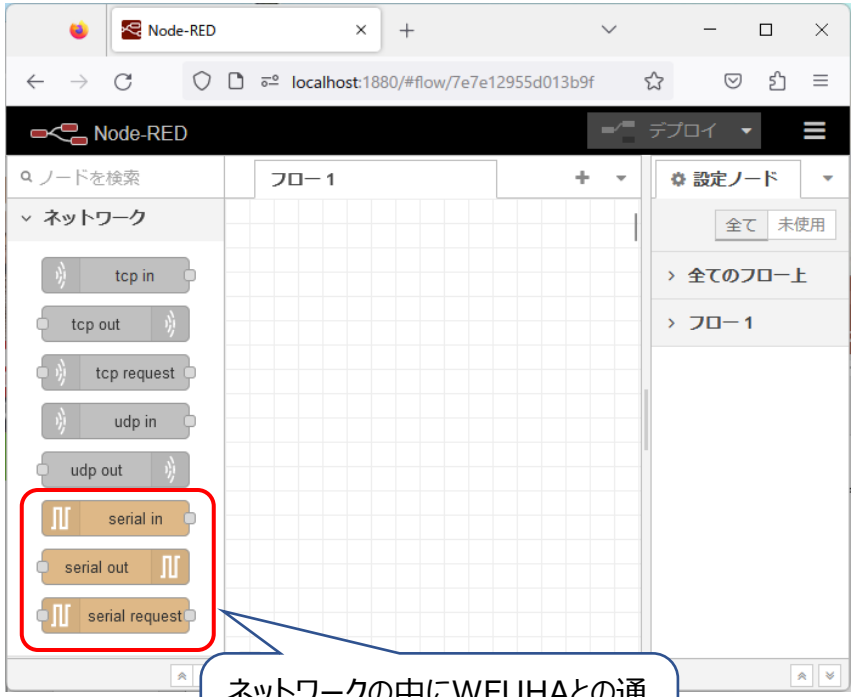
右図のhttp://127.0.0.1:1880/の部分をCtrlキーを押しながらクリックしてリンクを表示します。



6. serialportノードのインストール
右上のハンバーガーメニューより”パレットの管理”を選択します。
“ノードを追加”タグを開いて、”node-red-node-serialport”を追加します。

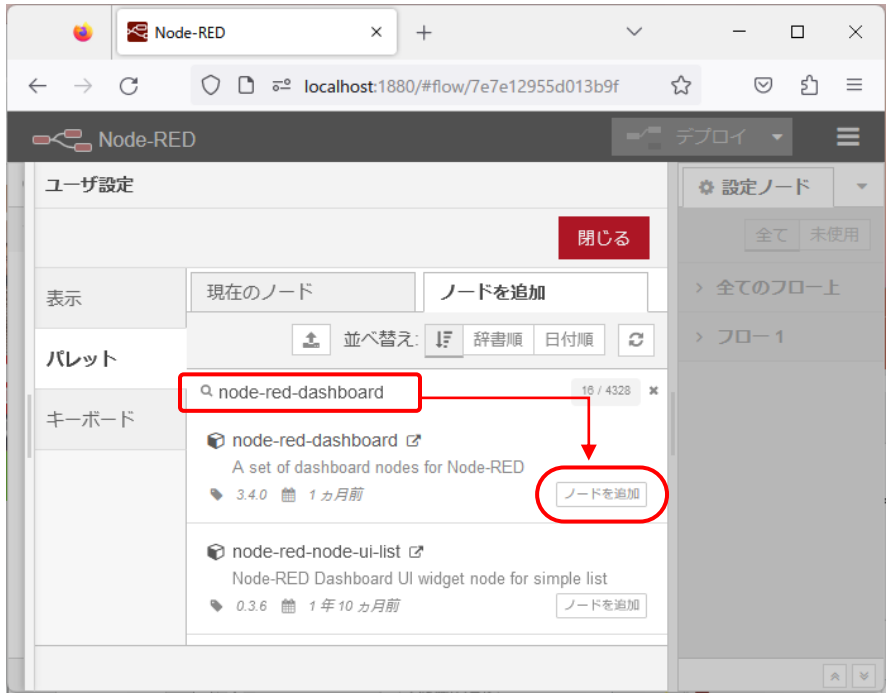


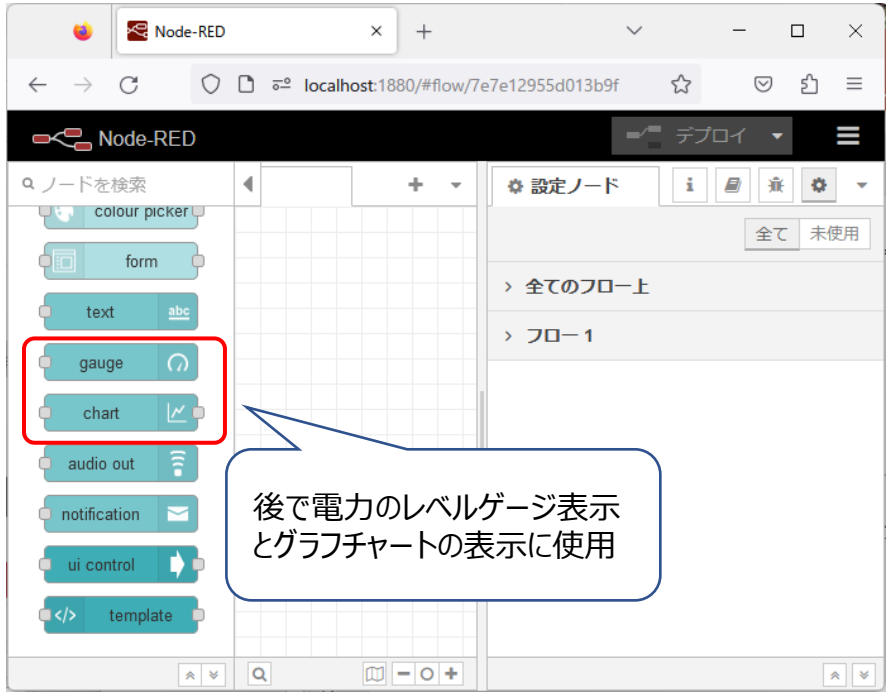




ネットワークの中にWFUHAとの通信に使用するシリアル出力とシリアル入力のノードが追加されます。

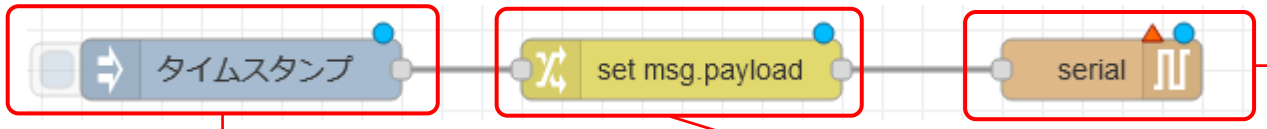
7. dashboardノードのインストール
Node-Redにグラフ表示のためにdashboardノードをインストールします。

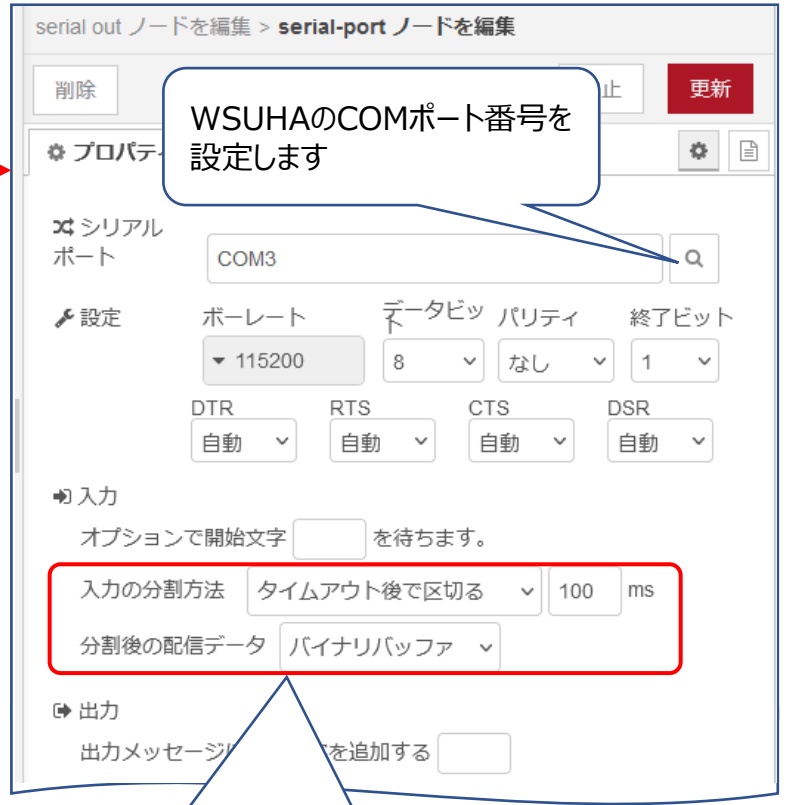




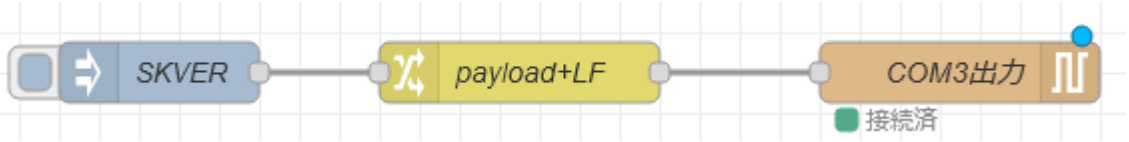
【1】ファームウェアバージョンを取得

WSUHAへのコマンド送信ノードを構築します。
injectノード、changeノード、serial outノードを配置して接続を行い、各ノードを編集します。
”SKVER”コマンドの後ろにLFを追加してCOMポートに出力します。

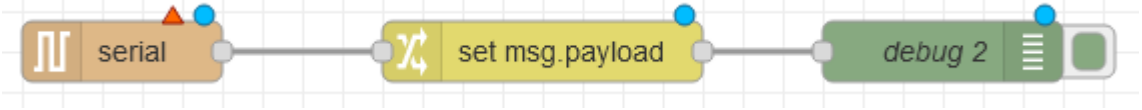


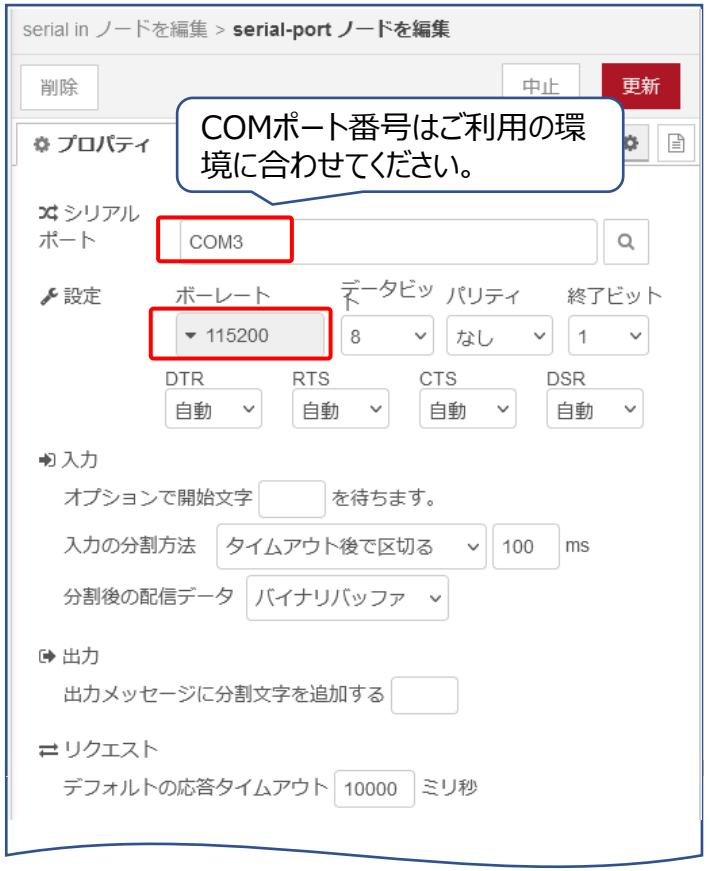


電力データを受信し取扱う都合上、バイナリーバッファとします。
CRLFは電力計測のバイナリーデータに含まれる可能性があるために、100ms受信得データが来ないことを受信の終了と判断することにします。



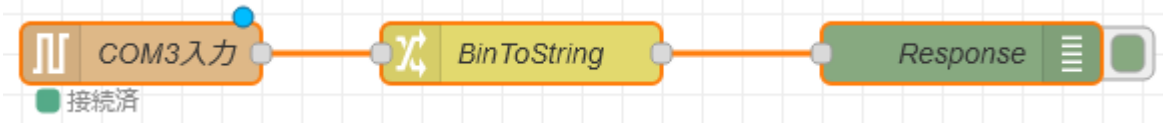
WSUHAへ送信したコマンドの応答データを受信するためのノードを構築します。
serial inノード、changeノード、debugノードを配置して接続を行い、各ノードを編集します。



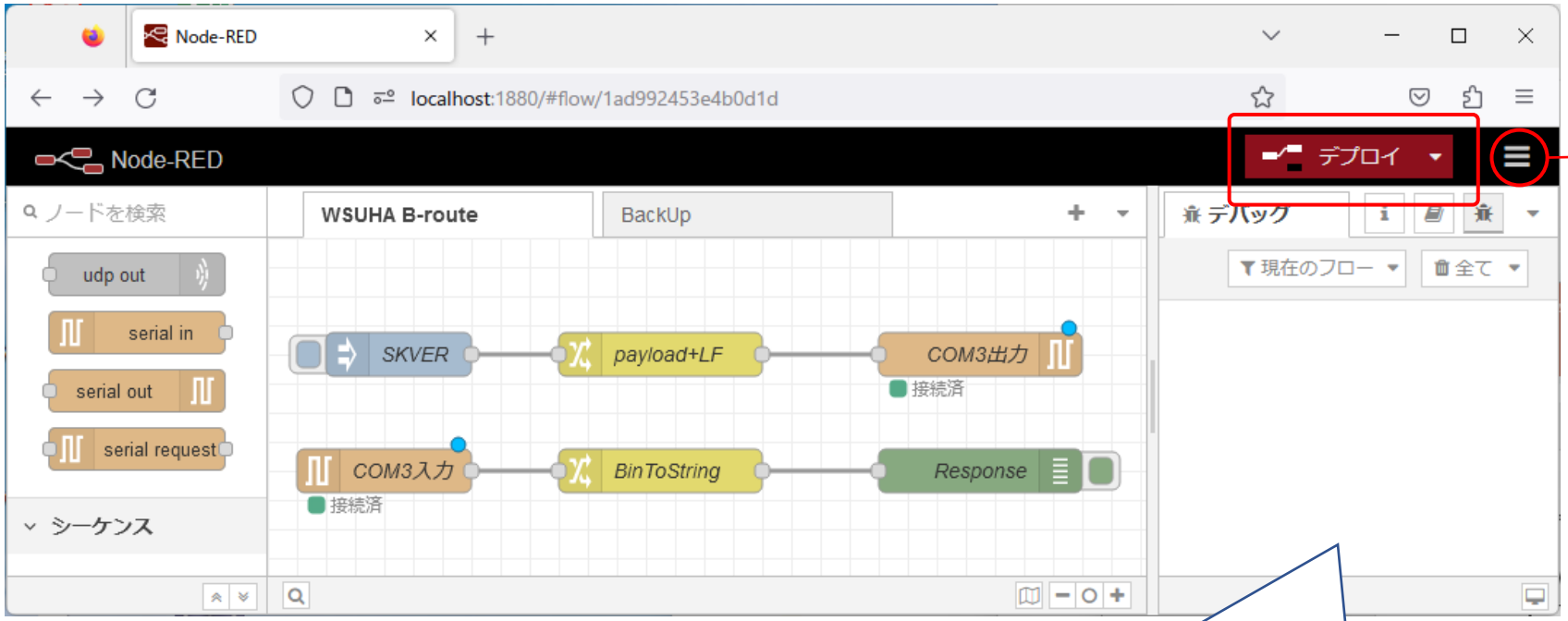


バイナリーデータをデバッグウィンドウに文字列で表示するために変換を行います。

※ JSONata式 に関する説明は下記を参照
https://docs.jsonata.org/overview.html



デプロイを行って、injectノードのボタンをクリックします。デバッグウィンドウに”SKVER”ファームウェアバージョン取得コマンドの応答データが表示されます。

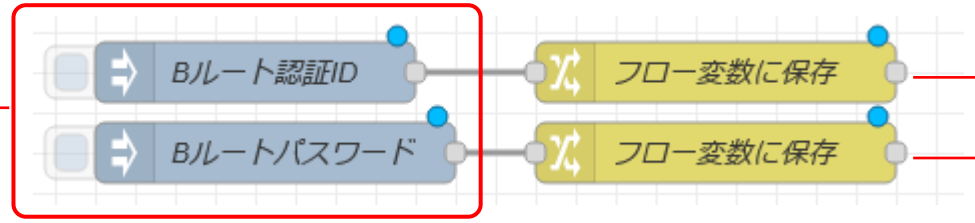


デバッグウィンドウに下記のようなバージョン情報が表示されます。

```
2023/4/12 11:21:08  ノード: Response
msg.payload : string[23]
▸ "SKVER↵EVER 1.5.2↵OK↵"
```

【2】Bルート認証IDとパスワード

電力会社が発行したBルート認証IDとパスワードをノード変数として保存します。



7．Bルート認証IDの定義

- Bルート認証IDは、お客様（需要家）のスマートメーター Bルート利用に当たって、電力会社等メーター設置者においてお客様（需要家）の確認、識別を行うために用いられるものであり、代表的な使い方は以下の通り。
- お客様（需要家）側は、EMS･アグリゲーションコントローラーにBルート認証IDとパスワードを入力することで正しいお客様（需要家）であることがスマートメーター側で証明されるとデータ通信が開始される。
- Bルート認証IDについては、お客様（需要家）が引越等で利用電力会社等変更する場合でも混乱をきたさないことを目的に全ての電力会社等メーター設置者が同一のフォーマットを採用する。
- スマートメーターを複数台設置する場合など、お客さま（需要家）がBルート認証IDとスマートメーターの関連が認識できるように、電力会社等のメーター設置者は、以下の項目を通知する。

通知項目：「種類（契約内容等）」「Bルート認証ID」「パスワード」「計器番号」「供給地点特定番号」

■Bルート認証ID：16進数[0～9、A～F] 32桁

0 0 -

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 29 30 31 32

8桁目迄：スマートメーター設置事業者特定領域

9桁目以降：自由領域

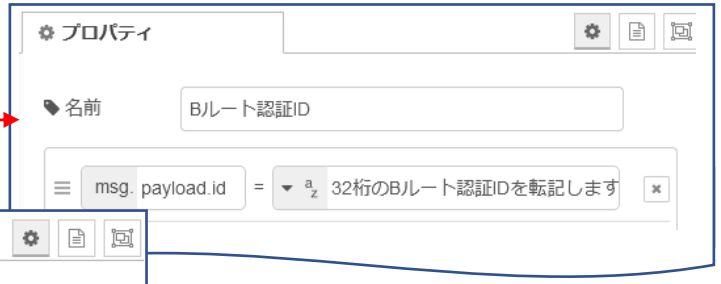
3～8桁目：エコーネットコンソーシアム会員メーカーコード

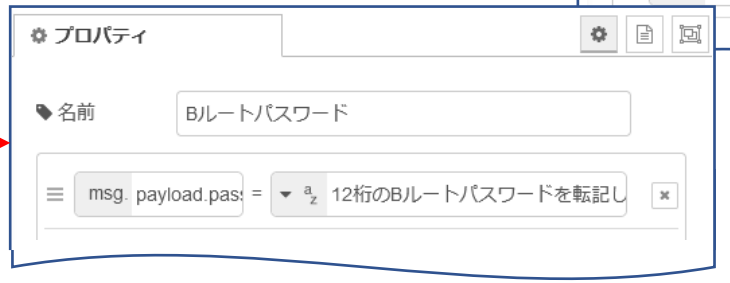
■パスワード：英数字［大文字･小文字区別なし］ 12桁

- - - - - - - - - - - -

1 2 3 4 5 6 7 8 9 10 11 12

18

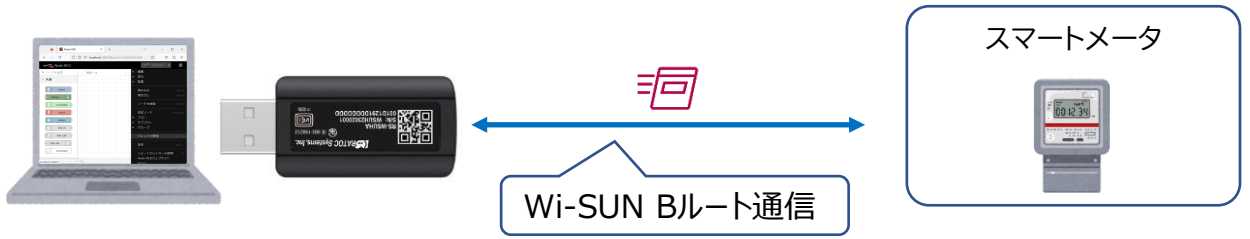




ここに電力会社から割り当てられたIDおよびパスワードを転記します。

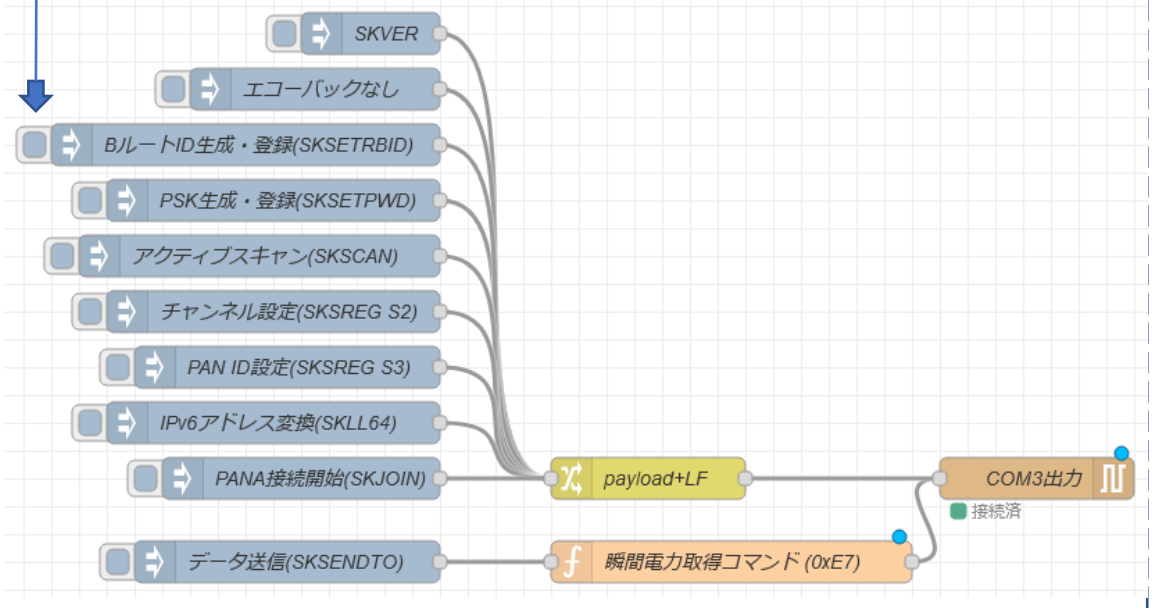
【3】スマートメータから電力測定値を取得するまで

スマートメータから電力測定値を取得するために下記のWi-SUNコマンドを順に送信します。次ページより、下記”PSK生成”以下のコマンドから順に、injectノードの追加とコマンド送信した時の受信処理を説明します。

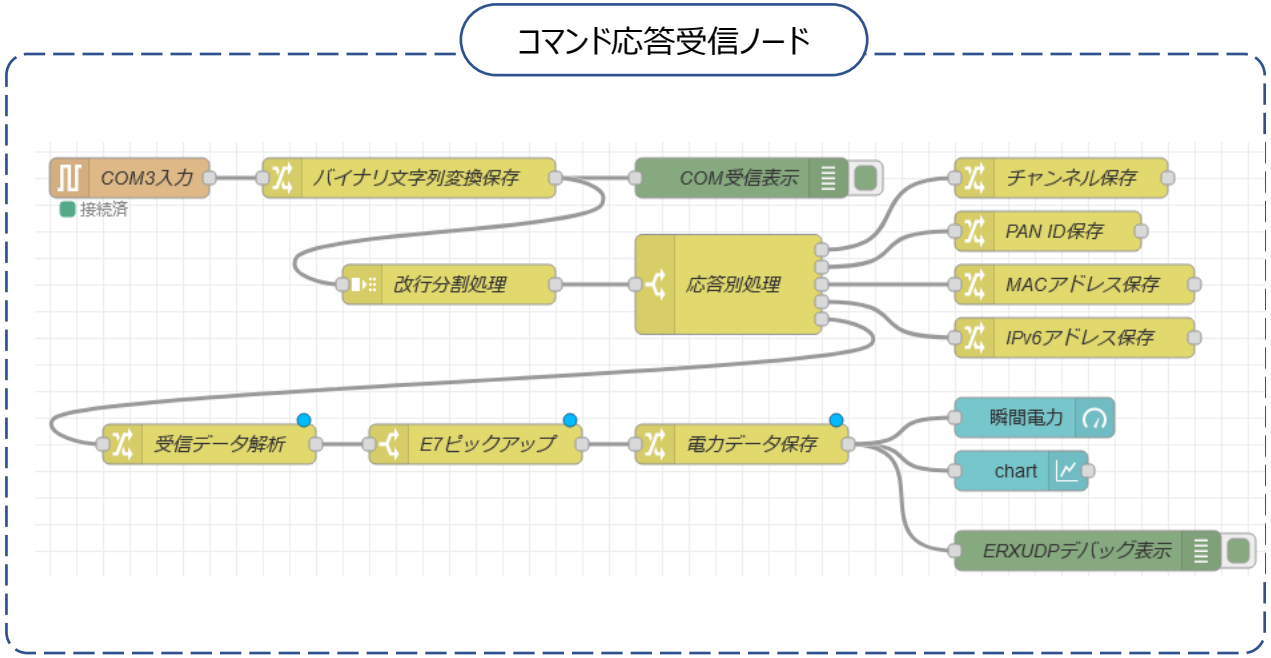


ここから順にinjectを起動します

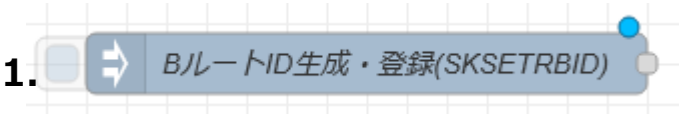
コマンド送信ノード



コマンド応答受信ノード



3-1.



● **受信処理**
“OK”のみなので省略

ROHMから提供されているBP35C2_コマンドリファレンス_HAN_DSE版_v1.0.0からの切り出しです。

4.19. SKSETRBID

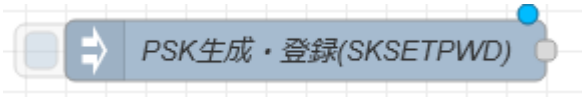
指定された<ID>から各 Route-B ID を生成して設定します。
Pairing ID (SA レジスタ)として<ID>の下位 8 文字が設定されます。

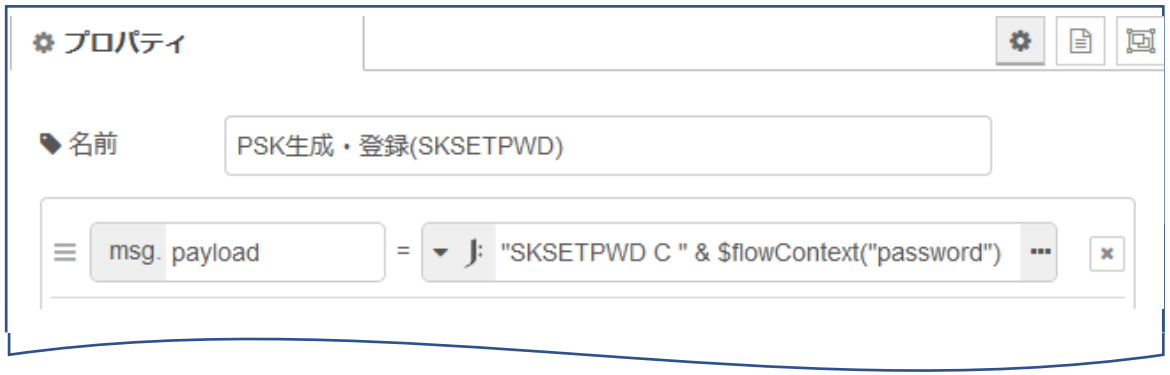
＊）<ID>は ASCII 32 文字必要で、足りない場合、不足分が不定値になります。

B 面（B ルート）側での実行になります。

Input		Response
SKSETRBID+ <ID> <CRLF>	→	
	←	OK<CRLF>

3-2.





● **受信処理**
“OK”のみなので省略

4.17. SKSETPWD

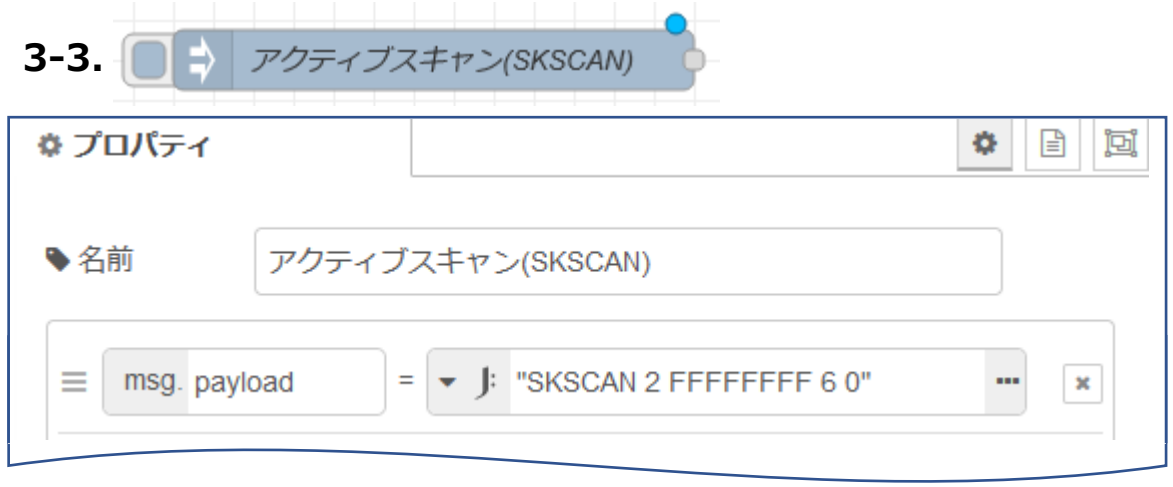
<PWD>で指定したパスワードから PSK を生成して登録します。
SKSETPSK による設定よりも本コマンドが優先され、PSK は本コマンドの内容で上書きされます。

＊）アルファベットの小文字はすべて大文字にして指定してください。
＊）<PWD>の文字数が指定した<LEN>に足りない場合、不足分は不定値になります。

B 面（B-ルート）側での実行になります。

Input		Response
SKSETPWD+ <LEN>+ <PWD><CRLF>	→	
	←	OK<CRLF>

3-3.



● **SKSCAN受信処理**

下記の受信処理行います。

1.EVENT 20:Beaconを受信した(直後にEPANDESCイベントが発生)

2.EPANDESC:アクティブスキャンで発見したPANの情報
　右受信データで後の処理で必要な”Channel番号”、”Pan ID”、”Addr”の情報
　を取り出してフロー変数に保存します。

3.EVENT 22:アクティブスキャンが完了した

4.11. SKSCAN

指定したチャンネルに対してアクティブスキャンまたは ED スキャンを実行します。

アクティブスキャンは、PAN を発見する度に 0x20 コードで EPANDESC イベントが発生して内容が通知されます。その後、指定したすべてのチャンネルのスキャンが完了すると EVENT イベントが 0x22 コードで発生して終了を通知します。

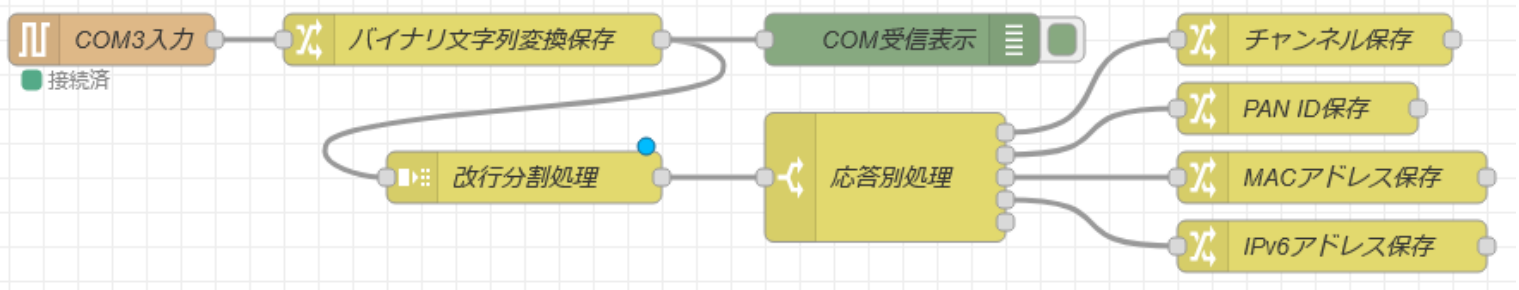
MODE に 2 を指定すると、拡張ビーコン要求の Payload IE に Pairing Sub-ID が付与されます。Pairing 値(8 バイト)は S0A で設定します。

Pairing ID が付与された拡張ビーコン要求を受信したコーディネータは、同じ Pairing 値が設定されている場合に、拡張ビーコンを応答します。

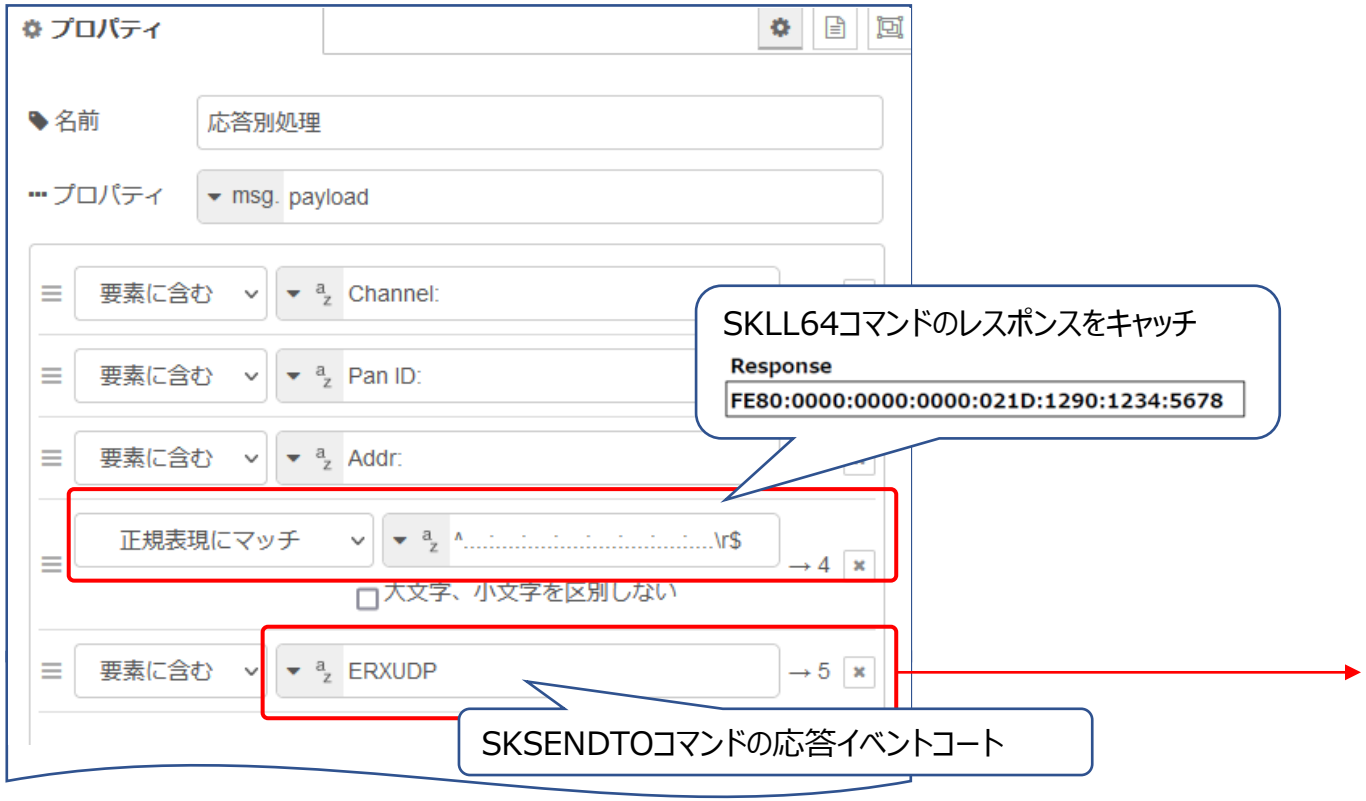
Input		Response
SKSCAN + <MODE>+ <CHANNEL_MASK>+ <DURATION>+ <SIDE><CRLF>	→	
	←	OK<CRLF>

Input Parameters

Name	Type	Description
<MODE>	UINT8	0 : ED スキャン 2:アクティブスキャン（IE あり） 3:アクティブスキャン（IE なし）
<CHANNEL_MASK>	UINT32	スキャンするチャンネルをビットマップフラグで指定します。 最下位ビットがチャンネル 33 に対応します。
<DURATION>	UINT8	各チャンネルのスキャン時間を指定します。 スキャン時間は以下の式で計算されます。 0.096 sec * (2^<DURATION> + 1) 値域：0-14
<SIDE>	UINT8	指定した MAC 面側で送信します 0:B ルート側へ送信 1:HAN 側へ送信



このフローでのポイントはEPANDESCイベント受信時に構文解析を行って”Channel番号”、”Pan ID”、”Addr”の情報を取り出してフロー変数に保存することです。



5.5. EPANDESC

アクティブスキャンを実行して発見した PAN を通知します。

Pair ID は、アクティブスキャンを<MODE>=2 で実行すると付与され、<MODE>=3 の場合は付与されません。

以下に一例を示します。

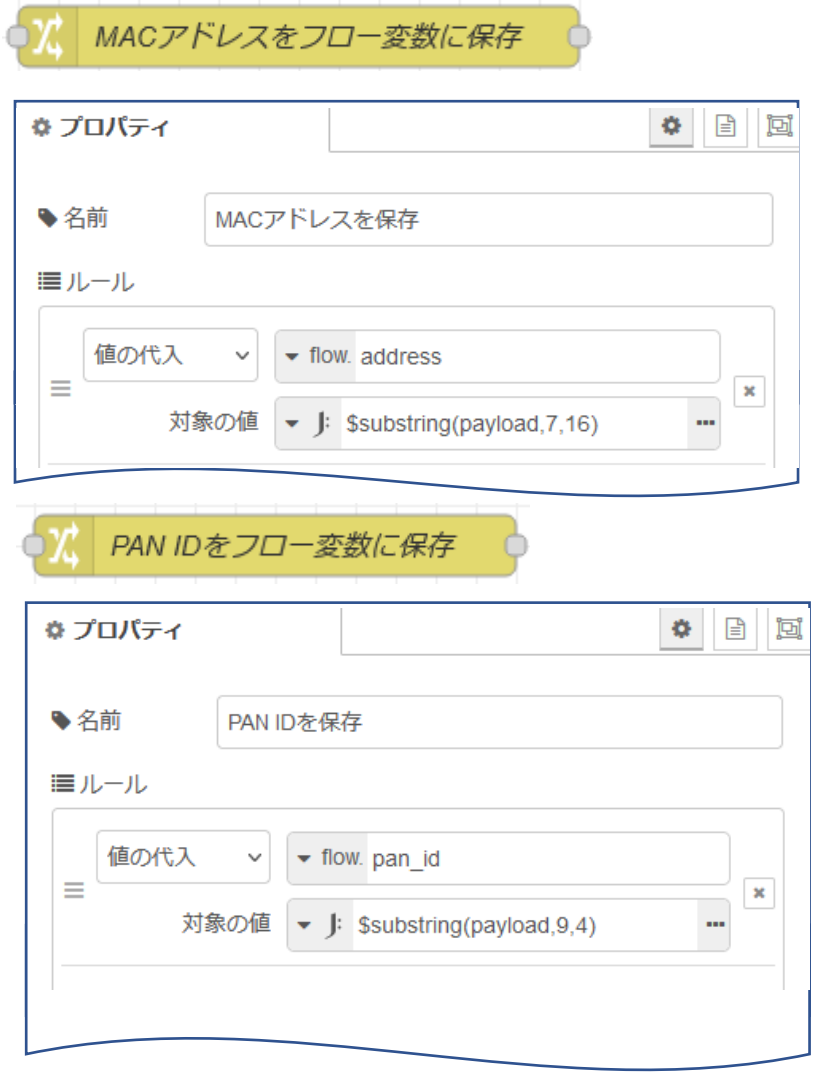
参照：SKSCAN

```
EPANDESC <CRLF>
  Channel:21 <CRLF>
  Channel Page:09 <CRLF>
  Pan ID:8888 <CRLF>
  Addr:12345678ABCDEF01 <CRLF>
  LQI:E1<CRLF>
  Side:0<CRLF>
  (PairID:AABBCCDD<CRLF>)
  (HEMS:12345678ABCDEF01<CRLF>)
  (Relay:0<CRLF>)
  (Relay Endpoint:1<CRLF>)
```

Name	Type	Description
Channel	UINT8	発見した PAN の周波数（論理チャンネル番号）
Channel Page	UINT8	発見した PAN のチャンネルページ
Pan ID	UINT16	発見した PAN の PAN ID
Addr	ADDR64	アクティブスキャン応答元のアドレス
LQI	UINT8	受信したビーコンの受信 RSSI
Side	UINT8	スキャンを実行した MAC 面(0 または 1)
PairID	CHAR[8]	（mode=2 でスキャンを実行した場合のみ） 相手から受信した Pairing ID
HEMS	CHAR[8]	（H 面で mode=2 でスキャンを実行した場合のみ） HEMS の 64bit アドレス
Relay	UINT8	（H 面で mode=2 でスキャンを実行した場合のみ） 1:リレーデバイスとして動作している 0:リレーデバイスでない


改行分割処理
プロパティ
型に基づいて msg.payload を分割:
文字列 / バッファ
分割
a z \n
メッセージのストリームとして処理
配列
分割
固定長 1
オブジェクト
各key/valueペアのメッセージを送信
keyのコピー先
msg.
名前
改行分割処理

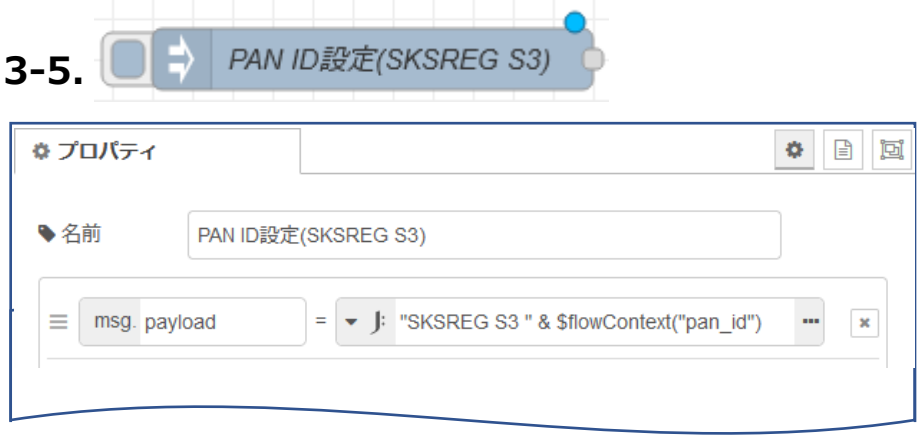

文字列とバイナリを別々に保存
プロパティ
名前
バイナリ文字列変換保存
ルール
値の代入
msg. payload_binary
対象の値
msg. payload
値のディープコピー
値の代入
msg. payload
対象の値
J: payload_binary.toString()
チャンネルをフロー変数に保存
プロパティ
名前
チャンネル保存
ルール
値の代入
flow. channel
対象の値
J: $substring(payload,10,2)


MACアドレスをフロー変数に保存
プロパティ
名前
MACアドレスを保存
ルール
値の代入
flow. address
対象の値
J: $substring(payload,7,16)
PAN IDをフロー変数に保存
プロパティ
名前
PAN IDを保存
ルール
値の代入
flow. pan_id
対象の値
J: $substring(payload,9,4)

3-4.

● **受信処理**
“OK”のみなので省略

3-5.



● **受信処理**
“OK”のみなので省略

4.1. SKSREG

仮想レジスタの内容を表示・設定します。
<SREG>に続けて<VAL>を指定すると値の設定、<VAL>を指定しないとそのレジスタの現在値を表示します。値の場合は ESREG イベントで通知されます。

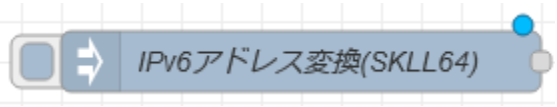
コマンド例：
PAN ID = 0x8888 を設定する場合

```
SKSREG S3 8888
```

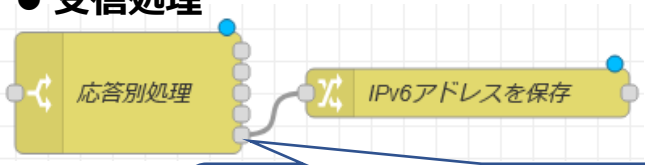
Input		Response
SKSREG+ <SREG>+ <VAL><CRLF>	→	
	←	設定の場合 OK<CRLF>

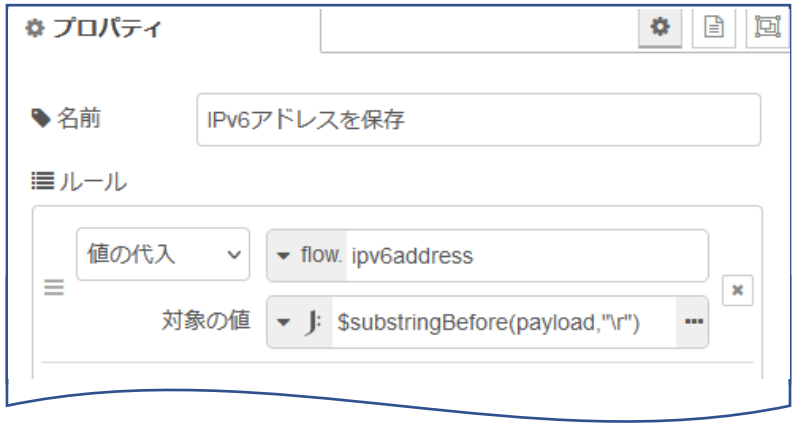
レジスタ番号	内容	属性	初期値	値域	保存	面対応
S01	自端末の IEEE 64bit（MAC アドレス） このレジスタを設定するとプロトコル・スタックが一度、リセットされます。	R/W		0x0 – 0xFFFFFFFFFFFFFFFF	○	×
S02	自端末が使用する周波数の論理チャンネル番号	R/W	0x21	0x21 – 0x3C	○	×
S03	自端末の PAN ID 0xFFFF を除いて、B, H 面で同じ PAN ID を設定することはできません。	R/W	0xFFFF	0x0000 – 0xFFFF	○	○

3-6.



● 受信処理





4.30. SKLL64

MAC アドレス(64bit)から IPv6 リンクローカルアドレスへ変換した結果を表示します。

コマンド例：

SKLL64 001D129012345678

Response

FE80:0000:0000:0000:021D:1290:1234:5678

Input		Response
SKLL64 + <ADDR64><CRLF>	→	
	←	<IPADDR><CRLF>

Input Parameters

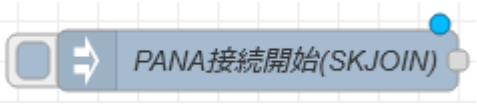
Name	Type	Description
<ADDR64>	UINT8[8]	端末の IEEE 64bit MAC アドレス

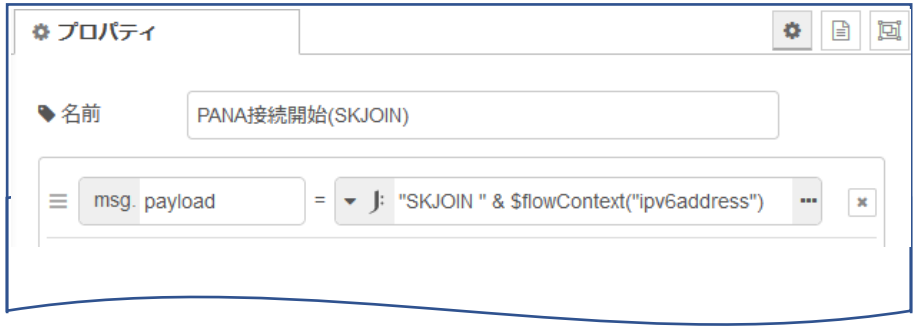
Response Parameters

Name	Type	Description
<IPADDR>	UINT8[16]	入力した MAC アドレスから変換したリンクローカルアドレスを表示します。

MACアドレスから変換したIPv6アドレスをフロー変数に保存します。SKSENDTOで必要になります。

3-7.





● **受信処理**

下記の受信処理行います。

1. EVENT 21, EVENT 02が発生し、で自動的に接続処理を行います
   EVENT 21: UDP送信処理が完了した
   EVENT 02: NAを受信した
2. EVENT 25: PANAによる接続が完了した

4.5. SKREJOIN

現在接続中の相手に対して再認証シーケンスを開始します。
再認証シーケンスの前に SKJOIN による接続が成功している必要があり、接続していないと ER10 になります。
再認証に成功すると、暗号キーと PANA セッション期限が更新されます。

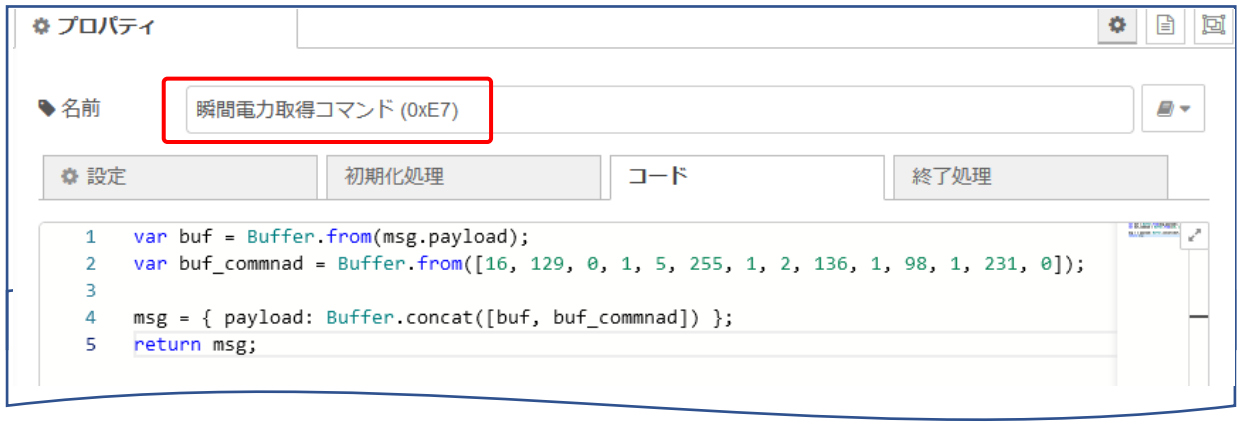
PaC は、PAA が指定したセッションライフタイムの 80%が経過した時点で、自動的に再認証シーケンスを実行します。このため SKREJOIN コマンドは基本的に発行する必要がありませんが、任意のタイミングで再認証したい場合には本コマンドを使います。

PAA は、セッションが更新されずにライフタイムが過ぎるとセッション終了要請を自動的に発行します。

B 面（B ルート）側での実行になります。

Input		Response
SKREJOIN<CRLF>	→	
	←	OK<CRLF>

3-8.



4.9. SKSENDTO

指定した宛先に UDP でデータを送信します。
SKSENDTO コマンドは以下の形式で正確に指定する必要があります。

1）アドレスは必ずコロン表記で指定してください。
2）ポート番号は必ず 4 文字指定してください。
3）データ長は必ず 4 文字指定してください。
4）セキュリティフラグは 1 文字で指定してください。
5）データは入力した内容がそのまま忠実にバイトデータとして扱われます。スペース、改行もそのままデータとして扱われます。
6）データは、データ長で指定したバイト数、必ず入力してください。サイズが足りないと、指定したバイト数揃うまでコマンド受け付け状態から抜けません。
7）データ部の入力はエコーバックされません。

Input		Response
SKSENDTO+ <HANDLE>+ <IPADDR>+ <PORT>+ <SEC> + <SIDE> + <DATALEN>+ <DATA>	→	
	←	<CRLF>OK<CRLF>

● ECHONET Lite フレーム電文構成

SKSENDTO 1 FE80:0000:0000:0000:XXXX:XXXX:XXXX:XXXX 0E1A 1 0 000E **[BINARY 0x:10 81 00 01 05 FF 01 02 88 01 62 01 E7 00]**

ECHONET Lite フレーム電文

SEOJ	DEOJ	ESV	OPC	EPC 1	PDC 1	EDT 1

SEOJ	送信元ECHONET Lite オブジェクト指定	05 FF 01	
DEOJ	相手先ECHONET Lite オブジェクト指定	02 88 01	
ESV	ECHONET Lite サービス	62	
OPC	処理プロパティ数	01（= n）	
		瞬時電力	積算電力
EPC n	ECHONET Lite プロパティ	E7	E0
PDC n	EDTのバイト数	00	00
EDT n	プロパティ値データ（PDCで指定）		

EHD1	EHD2	TID	EDATA

EHD1	ECHONET Lite 電文ヘッダー1	10（固定）
EHD2	ECHONET Lite 電文ヘッダー2	81（固定）
TID	トランザクションID（自由形式）	1234
EDATA	ECHONET Lite データ	

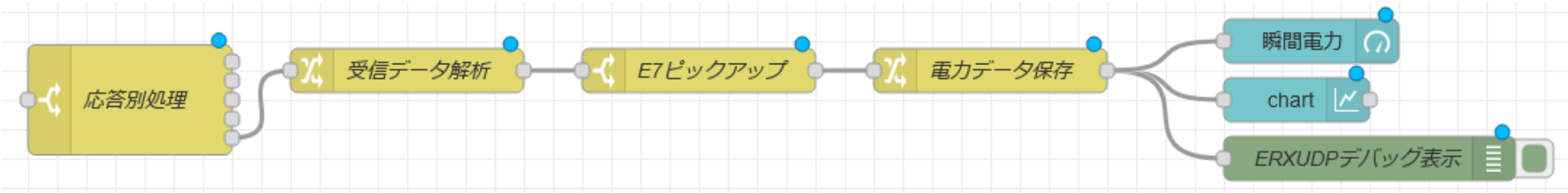
ECHONET 機器オブジェクト詳細規定

３．３．２５ 低圧スマート電力量メータクラス規定

クラスグループコード ： 0x02
クラスコード ： 0x88
インスタンスコード ： 0x01～0x7F (0x00：全インスタンス指定コード)

プロパティ名称	EPC	プロパティ内容 値域(10 進表記)	データ型	データサイズ	単位	アクセスルール	必須	状変時アナウンス	備考
積算電力量計測値(正方向計測値)	0xE0	積算電力量を 10 進表記において、最大 8 桁で示す。 0x00000000～0x05F5E0FF (0～99,999,999)	unsigned long	4 Byte	kWh	Get	○		
瞬時電力計測値	0xE7	電力実効値の瞬時値を 1W 単位で示す。 0x80000001～0x7FFFFFFD (-2,147,483,647～2,147,483,645)	signed long	4 Byte	W	Get	○		

● 受信処理



プロパティ

名前 受信データ解析

ルール

値の代入 msg. property
対象の値 J: payload_binary.slice(135,136).readUInt8(0)

値の代入 msg. length
対象の値 J: payload_binary.slice(136,137).readUInt8(0)

値の代入 msg. data
対象の値 J: payload_binary.slice(137,141).readUInt32BE(0)

プロパティ

名前 E7ピックアップ

プロパティ msg. property

== 09 231 → 1

プロパティ

名前 電力データ保存

ルール

値の代入 msg. payload
対象の値 msg. data
値のディープコピー

“ERXUDP FE80:0000:0000:0000:5AC2:32FF:FECB:B31F FE80:0000:0000:0000:021D:1291:0004:BE8A 0E1A 0E1A 58C232FFFECBB31F 1 0 0012 | 文字化けデータ部

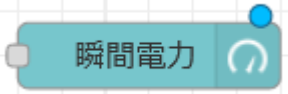
ECHONET Lite 受信データ部を16進に変換して再表示

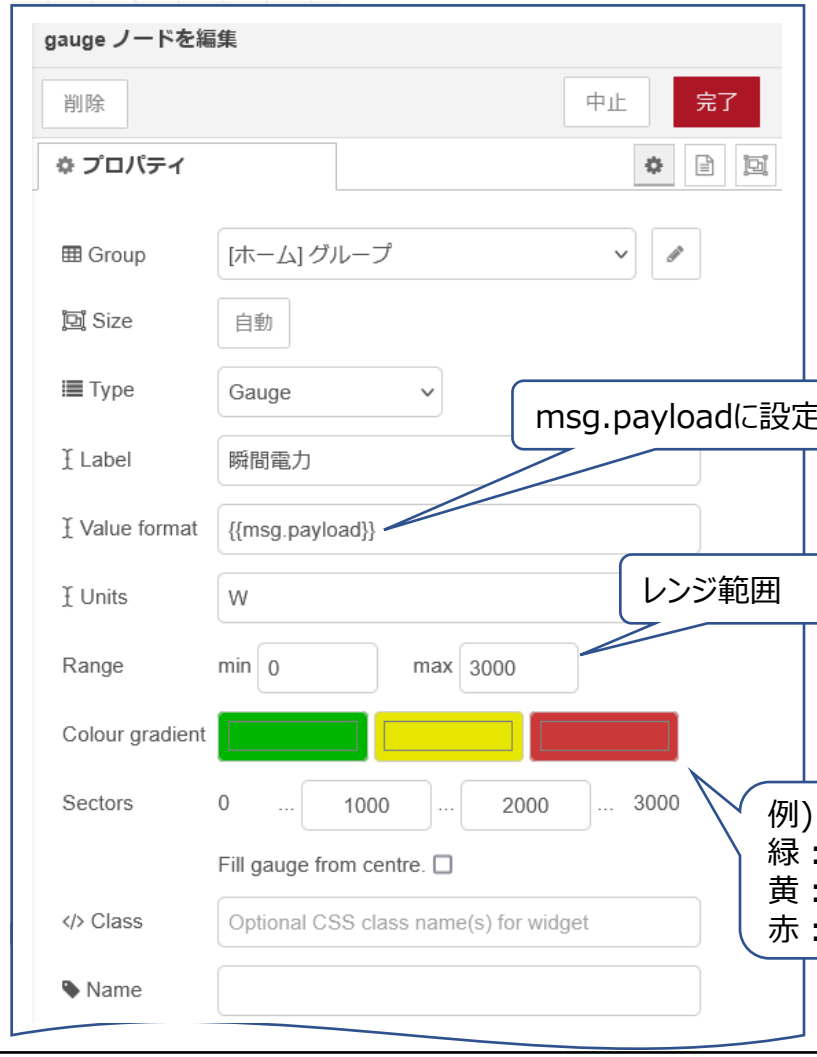
16,129,0,1,2,136,1,5,255,1,114,1,231,4,0,0,1,180

0x10, 0x81, 0x00, 0x01, 0x02, 0x88, 0x01, 0x05, 0xFF, 0x01, 0x72, 0x01, 0xE7, 0x04, 0x00, 0x00, 0x01, 0xB4

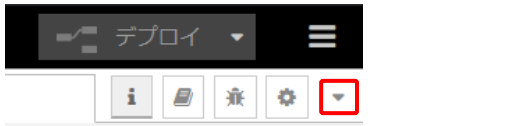
項目	ECHONET Lite 受信データ部	内容
ECHONET Liteヘッダ1	0x10	ECHONET Lite規格 (1xxx0000b:従来のECHONTE規格　00010000b:ECHONET Lite規格
ECHONET Liteヘッダ2	0x81	形式1　　(0x81:形式1 0x82:形式2)
Transaction ID	0x0001	送信受信を紐づけるためのパラメータ 対応する送受信で同じ値になる
SEOJ	0x02, 0x88, 0x01	相手先ECHONET Liteオブジェクト指定
	0x02	住宅・設備関連機器グループ　　(クラスグループコード)
	0x88	低電圧スマート電力メータ　　(クラスコード)
	0x01	インスタンスコード　同一クラスの識別コード
DEOJ	0x05, 0xFF, 0x01	送信元ECHONET Liteオブジェクト指定)
	0x05	管理・操作関連機器クラスグループ　(クラスグループコード)
	0xFF	コントローラ　　(クラスコード)
	0x01	インスタンスコード　同一クラスの識別コード
ESV	0x72	プロパティ値読み出し応答　(ECHONET Liteサービスコード)
OPC	0x01	処理プロパティ数
EPC1	0xE7	瞬時電力計測値　　(ECHONET Liteプロパティ)
PDC1	0x04	EDTのバイト数
EDT1	0x00, 0x00, 0x01, 0xB4	0x01B4 = 436[W]

3-9.



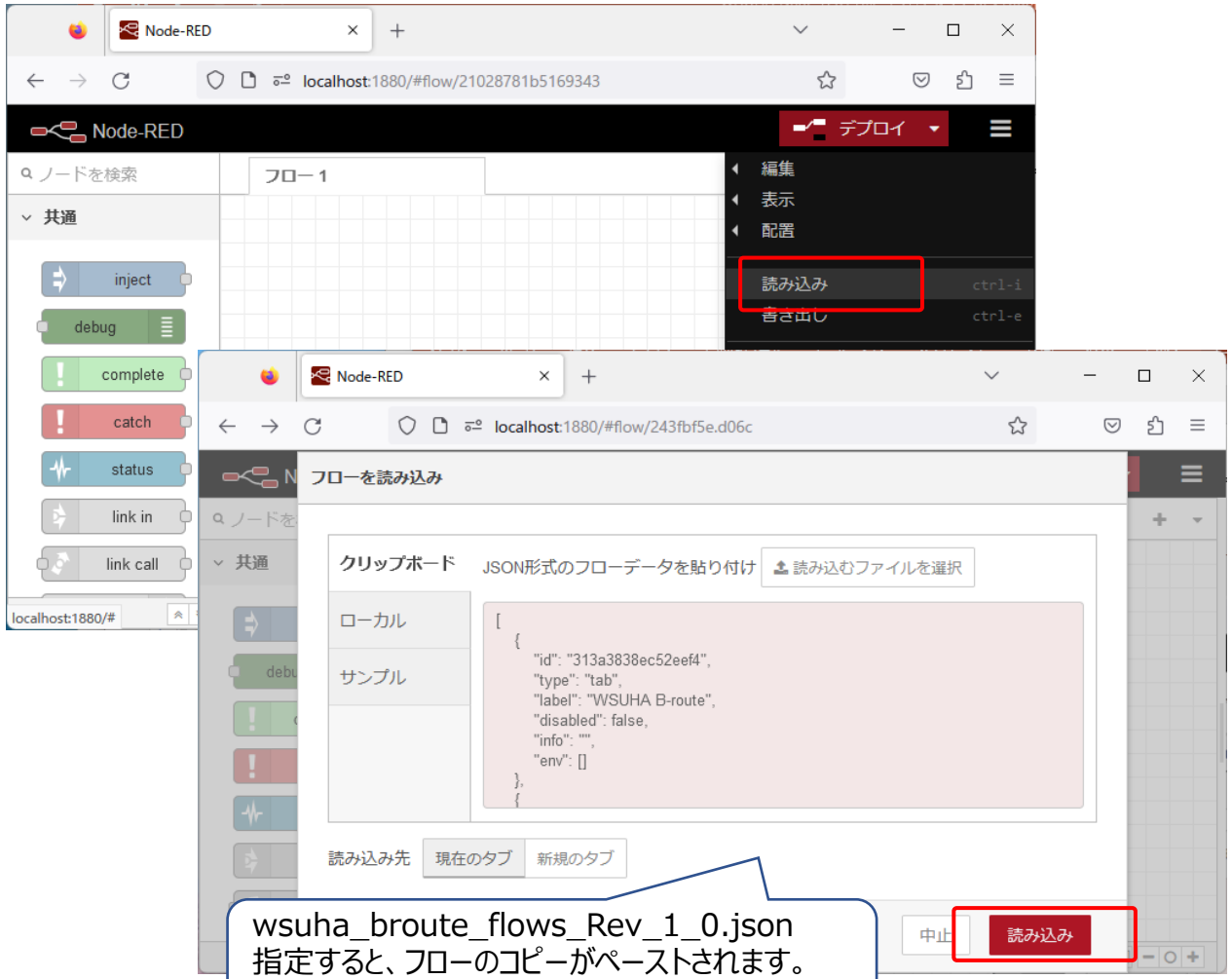


Dashboardの表示



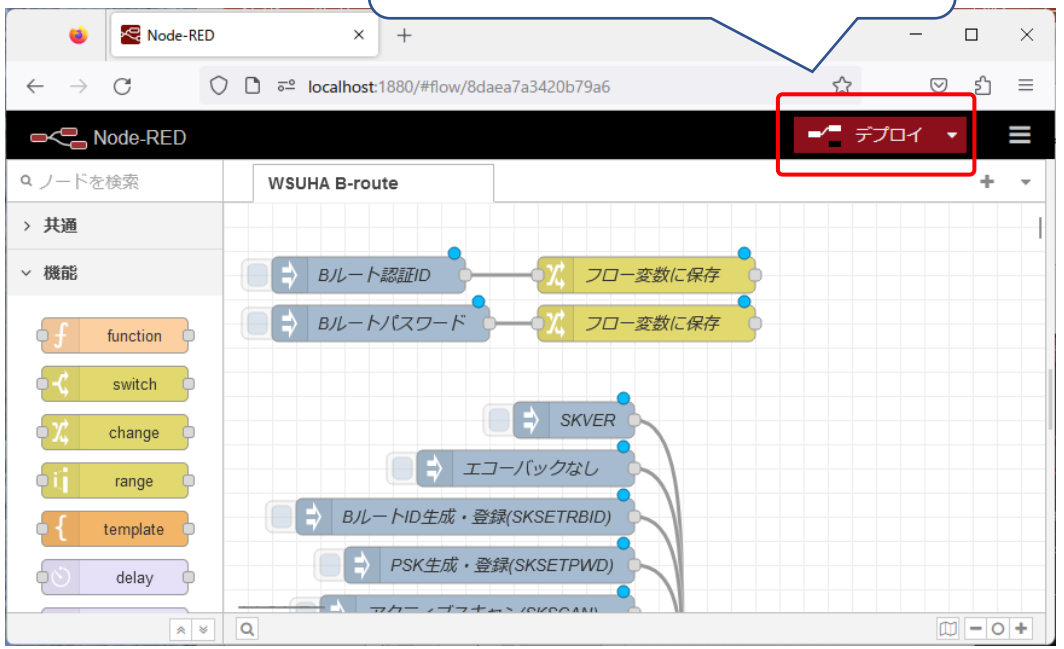
3-10.デモフローの読み込み

これまで説明を行った事例のフローを以下の手順で読み込みます。 ※WSUHAはパソコンのUSBポートに接続した状態で実行します。



wsuha_broute_flows_Rev_1_0.json
指定すると、フローのコピーがペーストされます。
現在のタブを指定して「読み込み」を行います

最後にデプロイを行うと次回より読み込んだフローが表示されます。



● 制限事項

太陽光発電装置やエネファーム等が導入されている家庭においては、発電量が消費電力を上回った状態において瞬時電力計測値は負の値になります。消費電力が発電量を上回った場合は、正の値になりますが、家庭での真の消費電流を表すものではありません。従って、発電装置が導入された家庭においては、スマートメータからエアコンなどを含めた全ても家電製品が消費している電流値を取得することはできませんので注意してください。発電量を考慮して見積る必要があります。

● 参照URL

1）Node.jsダウンロードサイト
https://nodejs.org/ja
2）Node-RED User Group Japan チュートリアル　はじめてフロー
https://nodered.jp/docs/tutorials/first-flow
3）JSONataドキュメント
https://docs.jsonata.org/overview.html
4）経済産業省「次世代スマートメーター制度検討会 とりまとめ」低圧スマートメーターbルート運用ガイドライン
https://www.meti.go.jp/shingikai/energy_environment/jisedai_smart_meter/20220531_report.html
5）ECHONET Lite規格書
https://echonet.jp/spec_g/#standard-01
6）ROHM BP35C0ドキュメントダウンロード
https://www.rohm.co.jp/products/wireless-communication/specified-low-power-radio-modules/bp35c0-product#designResources
7）ラトックシステム製品紹介
https://www.ratocsystems.com/products/wisun/usb-wisun/rs-wsuha/