

ia-cloud・Node-RED

実践IoTオンラインハンズオンセミナー

リアルタイムデータの取得

ia-cloud・Node-REDを使った実践的IoT活用

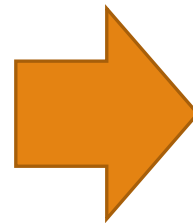
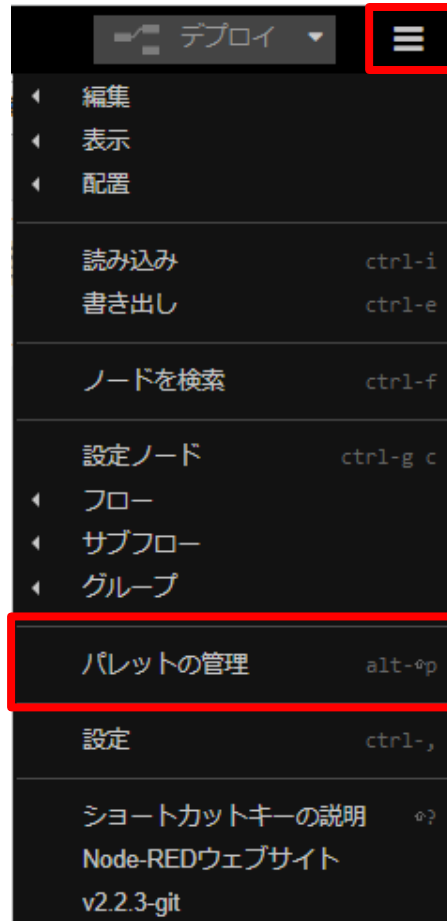
オンラインでのハンズオンセミナー

- 氏名：山田 純平
- 所属：株式会社ケー・ティー・システム 開発部
 - ◆ 製造業向けのシステム開発を中心
 - ◆ 参加・関係団体
 - ✓ 製造科学技術センター
 - ✓ ia-cloudプロジェクトグループ
 - ✓ Industrial Value Chain Initiative (IVI)
 - ✓ 日本ロボット工業会
 - ✓ ORiN協議会
 - ✓ 機械振興協会
 - etc...
- 主にEnOcean系のセンサーノードの開発や、セミナー講師を担当



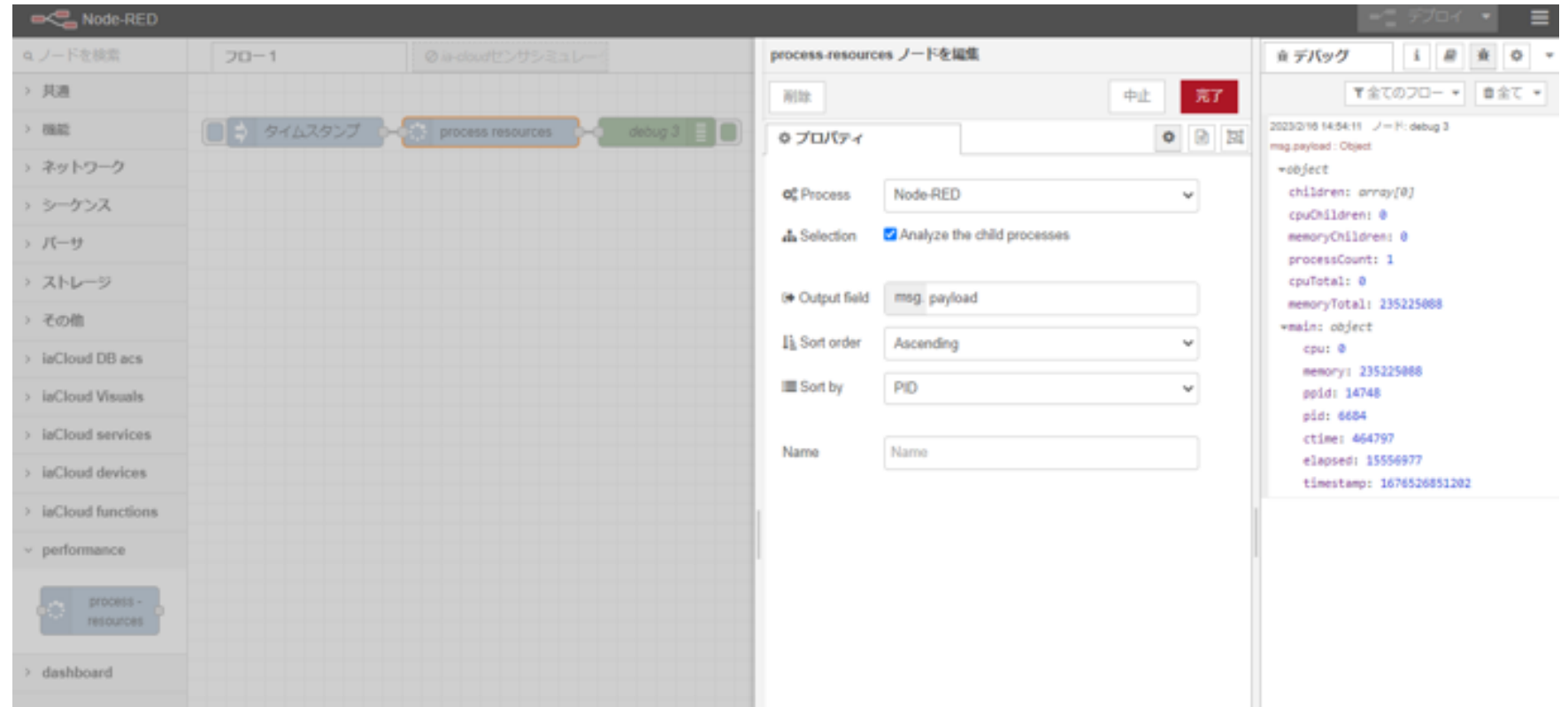
- CPU/メモリの使用状態の取得 (Raspberry Pi 以外)
 - ◆ process-resourcesノードの追加
 - ◆ データの取得
 - ◆ クラウドへの送信
- CPU/メモリの使用状態の取得 (Raspberry Pi)
 - ◆ ia-cloud-fds-raspberry-piノードの追加
 - ◆ データの取得とクラウドへの送信
- GrovePi+センサーを使用したデータの取得(Raspberry Pi)
 - ◆ GrovePiとは
 - ◆ データの取得
 - ◆ グラフ表示

■ process-resourcesノードの追加



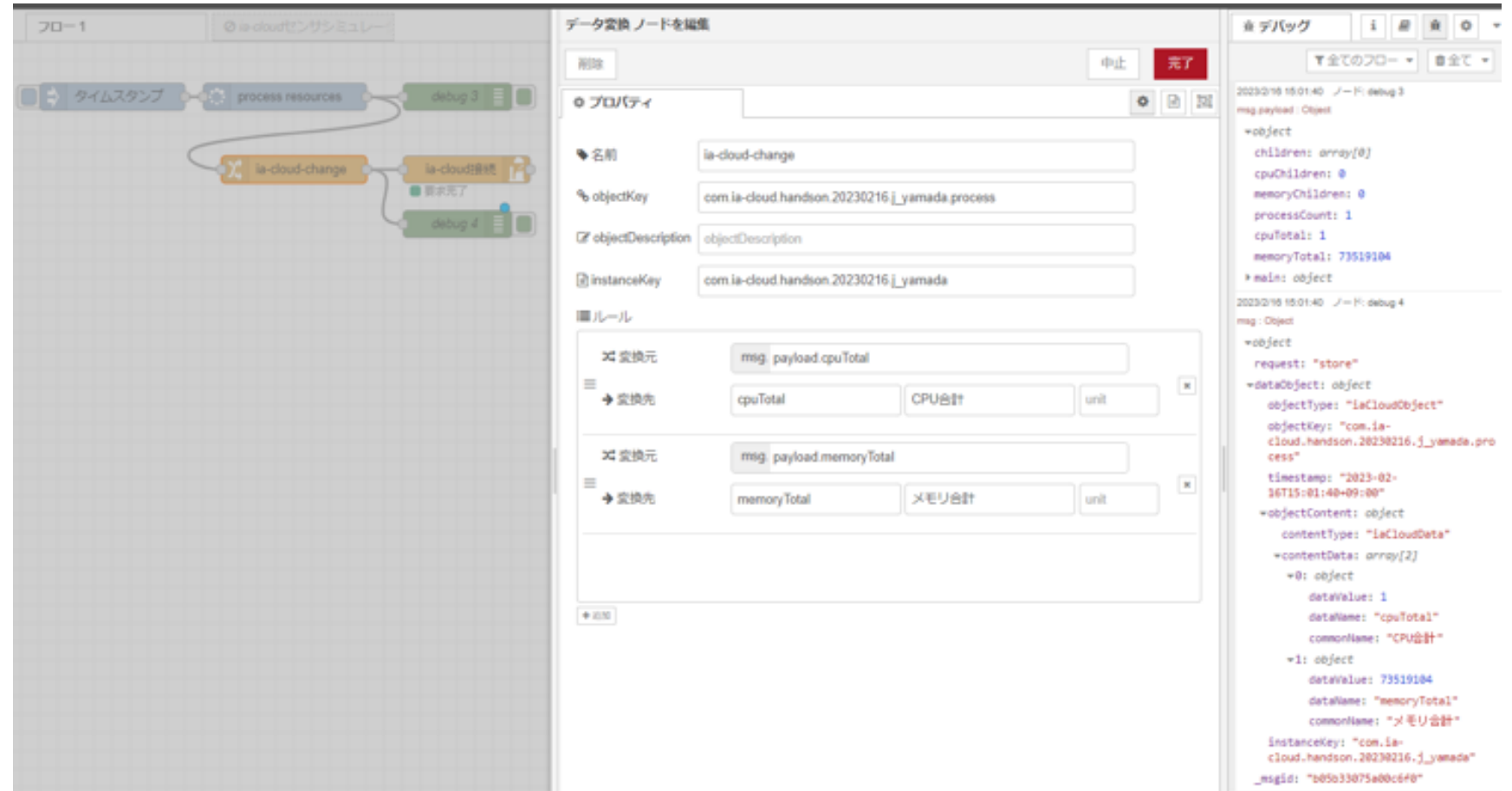
CPU/メモリの使用状態の取得 (Raspberry Pi 以外)

- データの取得
 - ◆ 以下のノードを配置
 - ✓ inject
 - ✓ process-resources
 - ✓ debug



※GitPodおよびRaspberry Piの環境で、process-resourcesノードの[Process]に「All」を選択すると、エラーが発生することを確認しています。Node-REDの再起動が必要となるためご注意ください。

- クラウドへの送信
 - ◆ 以下のノードを配置
 - ✓ データ変換
 - ✓ Ia-cloud接続
 - ◆ データ変換ノードで以下の2項目を抽出
 - ✓ cpuTotal
 - ✓ memoryTotal



The screenshot displays the 'ia-cloud' simulator interface. On the left, a flowchart shows a sequence of nodes: 'タイムスタンプ' (Timestamp), 'process resources', 'ia-cloud-change', 'ia-cloud接続' (Ia-cloud connection), and two 'debug' nodes. The 'ia-cloud-change' node is highlighted.

The central panel shows the configuration for the 'データ変換 ノードを編集' (Edit Data Transformation Node). It includes the following settings:

- 名前 (Name):** ia-cloud-change
- objectKey:** com.ia-cloud.handson.20230216.j.yamada.process
- objectDescription:** objectDescription
- InstanceKey:** com.ia-cloud.handson.20230216.j.yamada

The 'ルール (Rules)' section defines two transformation rules:

- Rule 1:**
 - 変換元 (Source): msg.payload.cpuTotal
 - 変換先 (Destination): cpuTotal
 - 単位 (Unit): CPU合計
- Rule 2:**
 - 変換元 (Source): msg.payload.memoryTotal
 - 変換先 (Destination): memoryTotal
 - 単位 (Unit): メモリ合計

The right panel shows the 'デバッグ (Debug)' console with two log entries:

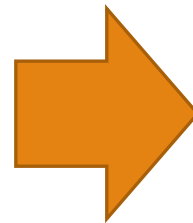
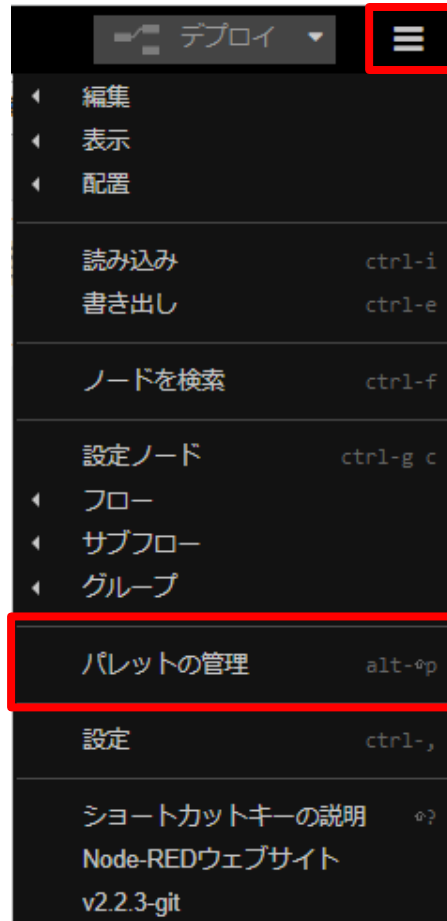
```

2023/02/16 15:01:40 ノード: debug 3
msg: Object
  *object
    children: array(0)
    cpuChildren: 0
    memoryChildren: 0
    processCount: 1
    cpuTotal: 1
    memoryTotal: 73519104
  *main: object

2023/02/16 15:01:40 ノード: debug 4
msg: Object
  *object
    request: "store"
  *dataObject: object
    objectType: "IaCloudObject"
    objectKey: "com.ia-cloud.handson.20230216.j.yamada.process"
    timestamp: "2023-02-16T15:01:40+09:00"
  *objectContent: object
    contentType: "IaCloudData"
    *contentData: array(2)
    *0: object
      dataValue: 1
      dataName: "cpuTotal"
      commonName: "CPU合計"
    *1: object
      dataValue: 73519104
      dataName: "memoryTotal"
      commonName: "メモリ合計"
    InstanceKey: "com.ia-cloud.handson.20230216.j.yamada"
    _msgid: "b05033075a00c4f0"
  
```

CPU/メモリの使用状態の取得 (Raspberry Pi)

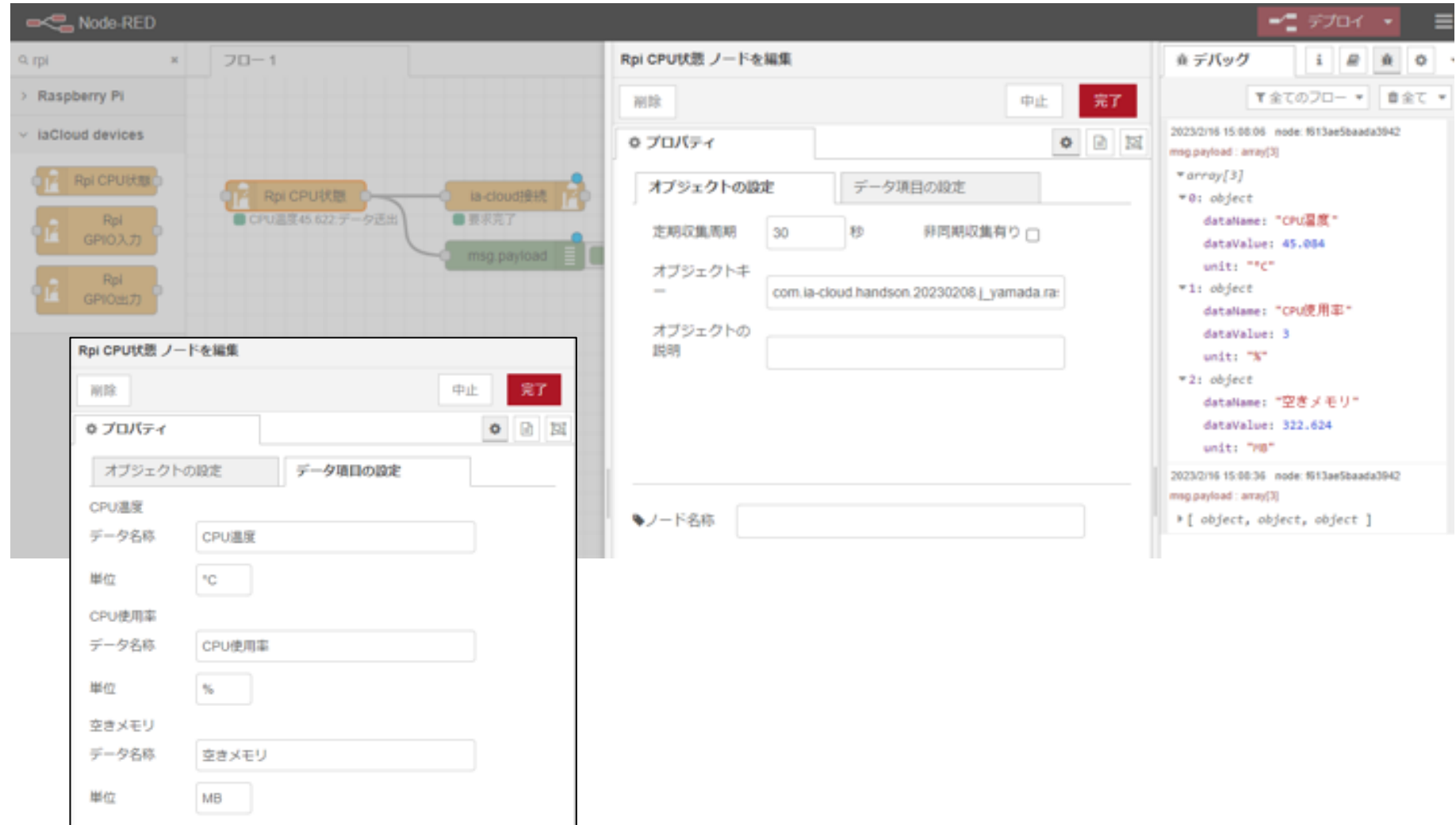
■ @ia-cloud/node-red-contrib-ia-cloud-fds-raspberry-piノードの追加



CPU/メモリの使用状態の取得 (Raspberry Pi)

- データの取得
 - ◆ 以下のノードを配置
 - ✓ Rpi CPU状態

- クラウドへの送信
 - ◆ 以下のノードを配置
 - ✓ Ia-cloud接続



The screenshot displays the Node-RED web interface. On the left, a flow named 'フロー 1' is shown with three nodes: 'Rpi CPU状態', 'Ia-cloud接続', and 'msg payload'. The 'Rpi CPU状態' node is connected to the 'Ia-cloud接続' node, which is then connected to the 'msg payload' node. The 'Rpi CPU状態' node has a status indicator showing 'CPU温度45.622データ送込'.

Two configuration panels are overlaid on the interface:

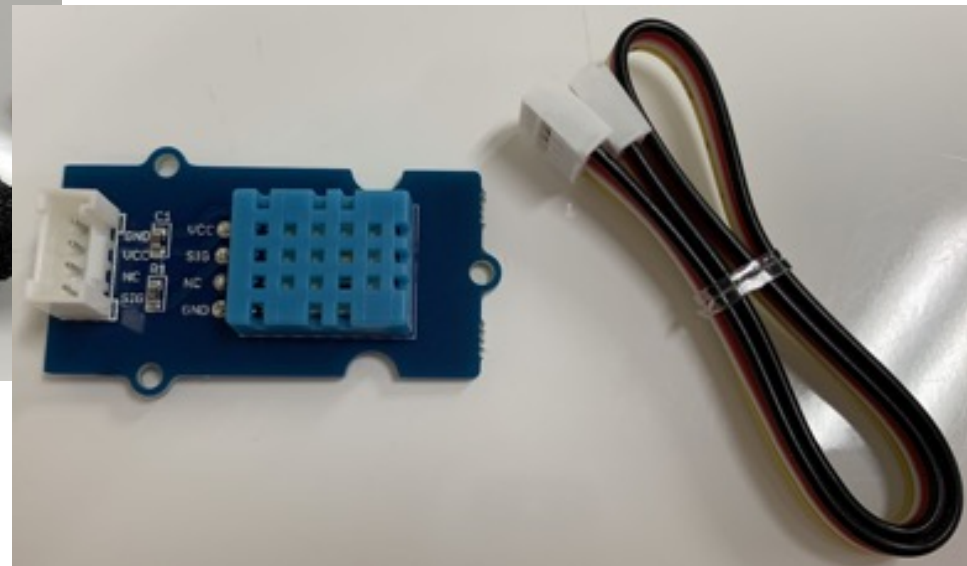
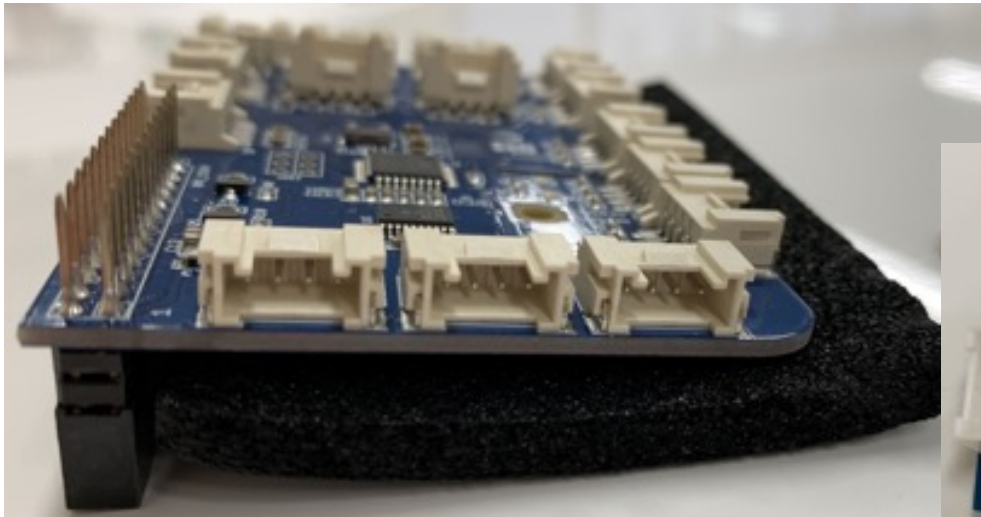
- Rpi CPU状態 ノードを編集:** This panel shows the configuration for the 'Rpi CPU状態' node. It has tabs for 'オブジェクトの設定' and 'データ項目の設定'. Under 'データ項目の設定', three data items are listed:
 - CPU温度: データ名称: CPU温度, 単位: °C
 - CPU使用率: データ名称: CPU使用率, 単位: %
 - 空きメモリ: データ名称: 空きメモリ, 単位: MB
- Ia-cloud接続 ノードを編集:** This panel shows the configuration for the 'Ia-cloud接続' node. It has tabs for 'オブジェクトの設定' and 'データ項目の設定'. Under 'オブジェクトの設定', the '定期収集周期' is set to 30 seconds, and the 'オブジェクトキー' is set to 'com.ia-cloud.handson.20230208_l.yamada.ra'. The 'オブジェクトの説明' field is empty.

On the right side of the interface, the 'デバッグ' (Debug) console shows the output of the flow. It displays a 'msg payload' object containing an array of three objects, each representing a data point from the Rpi CPU status node:

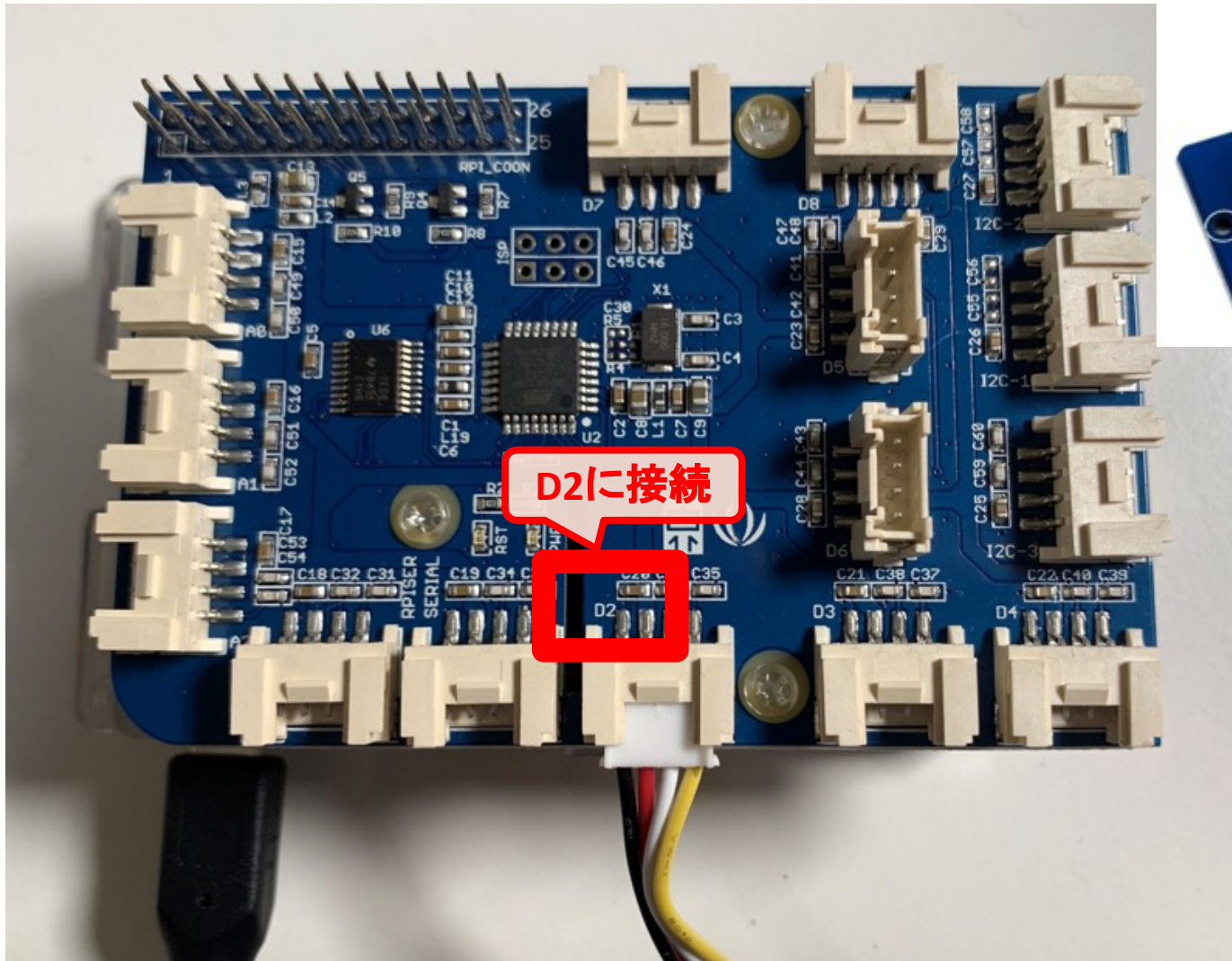
```

2023/2/16 15:08:06 node: 613ae5baada3942
msg payload: array[3]
  array[3]
    0: object
      dataName: "CPU温度"
      dataValue: 45.624
      unit: "°C"
    1: object
      dataName: "CPU使用率"
      dataValue: 3
      unit: "%"
    2: object
      dataName: "空きメモリ"
      dataValue: 322.624
      unit: "MB"
2023/2/16 15:08:06 node: 613ae5baada3942
msg payload: array[3]
  [ object, object, object ]
  
```


GrovePi+とは

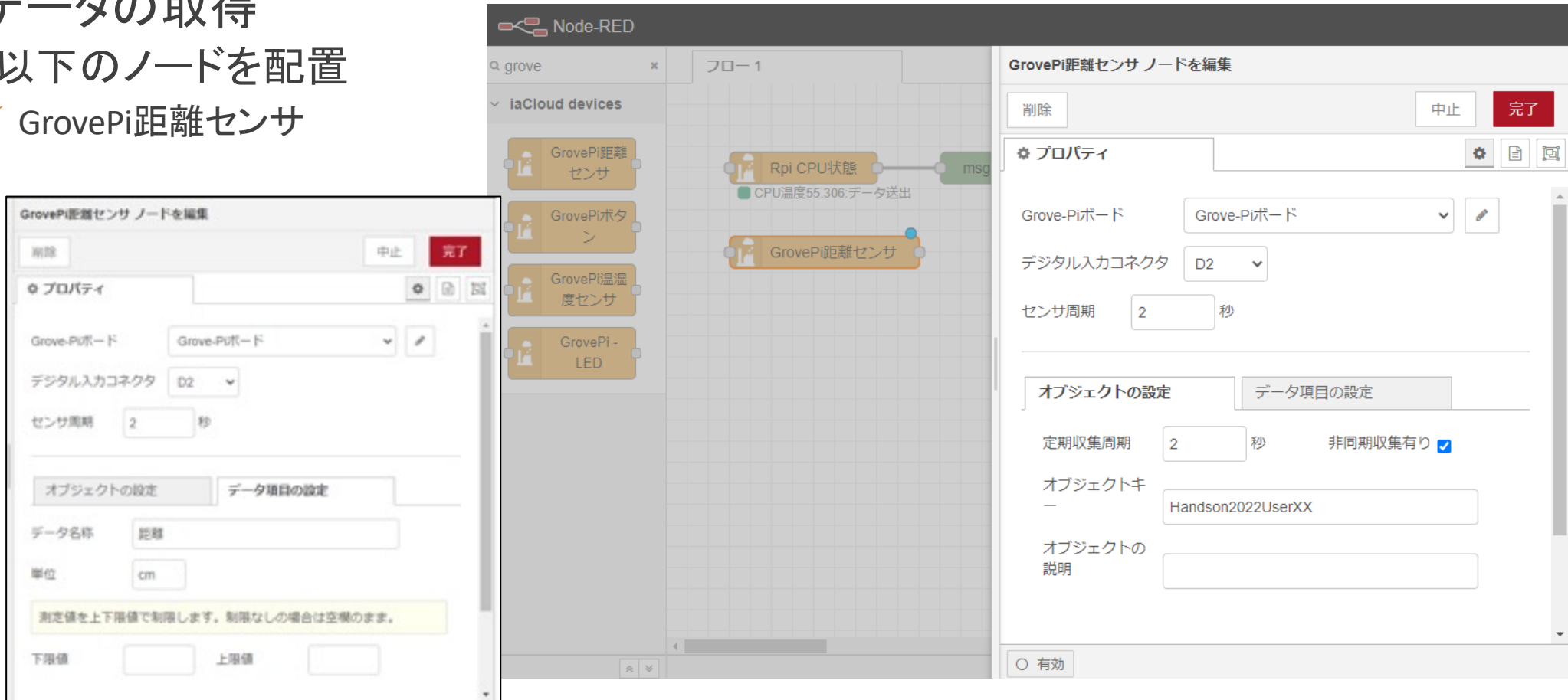


ラズパイに超音波距離センサを接続



GrovePi+センサーを使用したデータの取得(Raspberry Pi)

- データの取得
 - ◆ 以下のノードを配置
 - ✓ GrovePi距離センサ

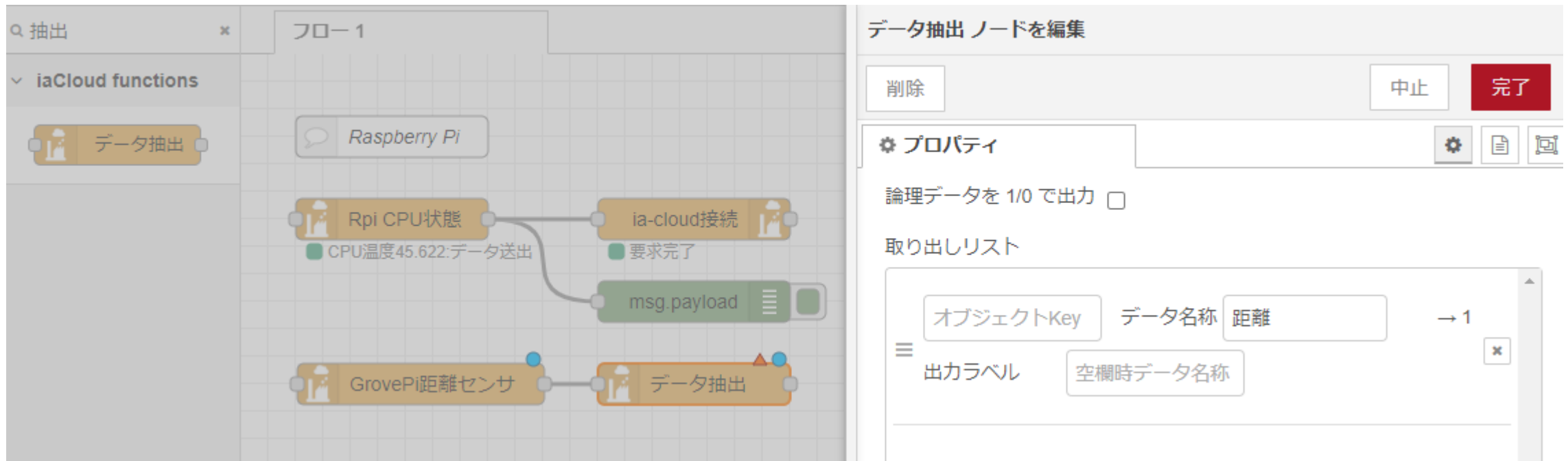


The image displays the Node-RED interface for configuring a GrovePi distance sensor. The main workspace shows a flow with the following nodes: 'GrovePi距離センサ', 'GrovePiボタン', 'GrovePi温度センサ', 'GrovePi - LED', 'Rpi CPU状態', and 'msg'. The 'Rpi CPU状態' node is connected to the 'msg' node, and the 'GrovePi距離センサ' node is connected to the 'Rpi CPU状態' node. The 'Rpi CPU状態' node has a label 'CPU温度55.306:データ送'.

Two configuration panels are overlaid on the workspace:

- Left Panel: GrovePi距離センサ ノードを編集**
 - Buttons: 削除, 中止, 完了
 - セクション: プロパティ
 - Grove-Piボード: Grove-Piボード
 - デジタル入力コネクタ: D2
 - センサー周期: 2 秒
 - タブ: オブジェクトの設定, データ項目の設定
 - データ名称: 距離
 - 単位: cm
 - メッセージ: 測定値を上下限値で制限します。制限なしの場合は空欄のまま。
 - 下限値: [] 上限値: []
- Right Panel: GrovePi距離センサ ノードを編集**
 - Buttons: 削除, 中止, 完了
 - セクション: プロパティ
 - Grove-Piボード: Grove-Piボード
 - デジタル入力コネクタ: D2
 - センサー周期: 2 秒
 - タブ: オブジェクトの設定, データ項目の設定
 - 定期収集周期: 2 秒 非同期収集有り
 - オブジェクトキー: Handson2022UserXX
 - オブジェクトの説明: []
 - 有効:

- データの取得
 - ◆ 以下のノードを配置
 - ✓ データ抽出



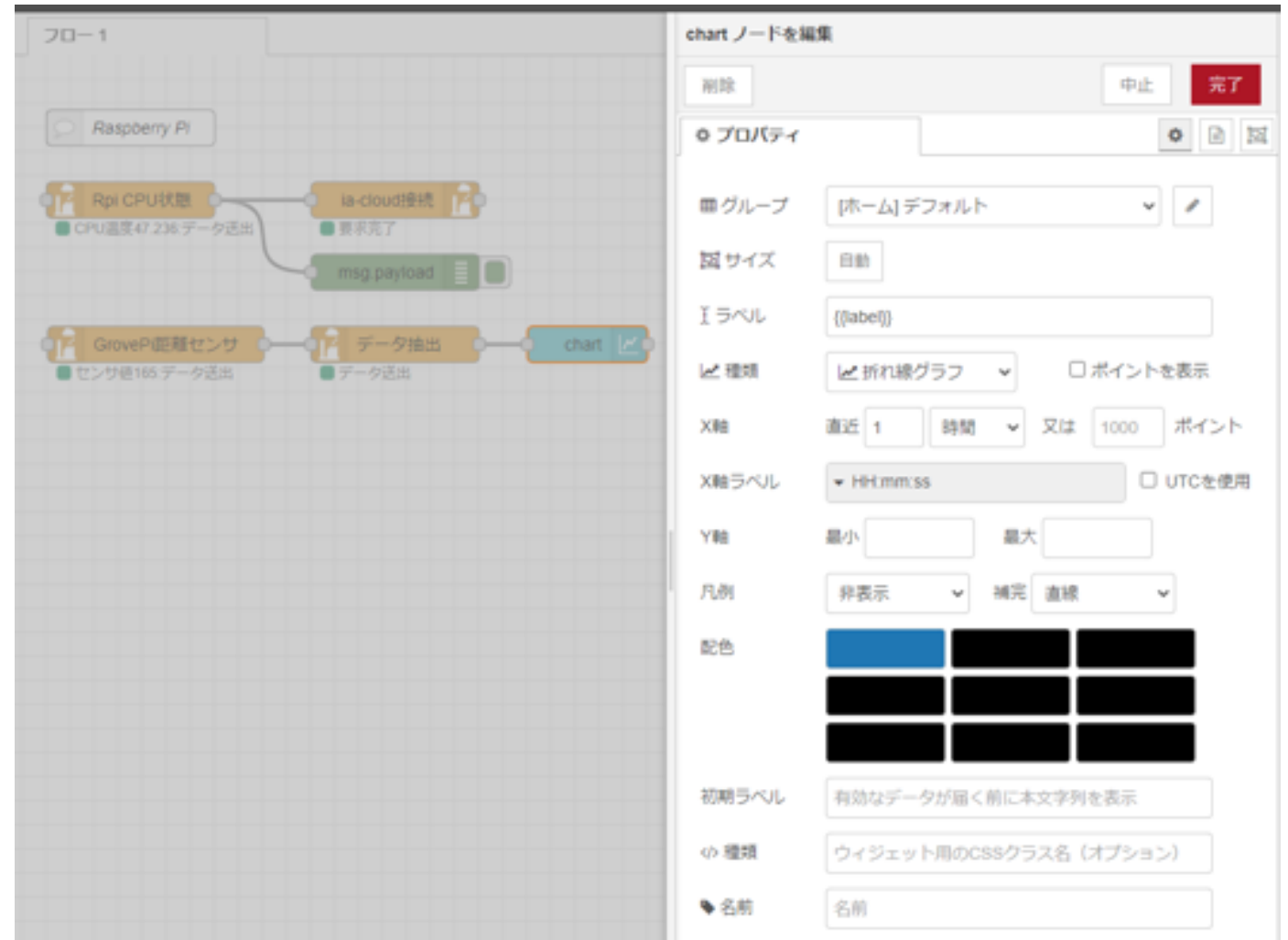
The screenshot displays the ia-cloud IDE interface. On the left, a search bar shows '抽出' and a list of 'iaCloud functions' including 'データ抽出'. The main workspace shows a workflow titled 'フロー 1' with a 'Raspberry Pi' node. The workflow consists of several nodes: 'Rpi CPU状態' (outputting 'CPU温度45.622:データ送出'), 'ia-cloud接続' (outputting '要求完了'), 'msg.payload', 'GrovePi距離センサ', and another 'データ抽出' node. On the right, the 'データ抽出 ノードを編集' (Edit Data Extraction Node) panel is open. It features buttons for '削除', '中止', and '完了'. Under 'プロパティ', there is a checkbox for '論理データを 1/0 で出力'. The '取り出しリスト' (Output List) section shows a table with columns for 'オブジェクトKey', 'データ名称', and '距離', with a value of '→ 1' in the '距離' column. Below this, there are fields for '出力ラベル' and '空欄時データ名称'.

■ グラフの表示

◆ 以下のノードを配置

✓ Chart

✓ ラベルに「`{{label}}`」と記入



The screenshot displays a Node-RED workflow and the configuration for a 'chart' node. The workflow includes:

- Raspberry Pi** (parent node)
- Rpi CPU状態** (Raspberry Pi CPU Status) node, outputting CPU temperature (47.236) and data.
- ia-cloud接続** (ia-cloud connection) node, outputting connection status (要求完了).
- msg.payload** (message payload) node.
- GrovePi距離センサ** (GrovePi distance sensor) node, outputting sensor distance (165) and data.
- データ抽出** (data extraction) node.
- chart** (chart) node, which is the focus of the configuration panel.

The **chart** node configuration panel shows the following settings:

- 削除** (Delete) button
- 中止** (Cancel) button
- 完了** (Done) button
- プロパティ** (Properties) section:
 - グループ** (Group): [ホーム] デフォルト
 - サイズ** (Size): 自動
 - ラベル** (Label): {{label}}
 - 種類** (Type): 折れ線グラフ (Line graph), with ポイントを表示 (Show points)
 - X軸** (X-axis): 直近 1 時間 (Last 1 hour), with または 1000 ポイント (or 1000 points)
 - X軸ラベル** (X-axis label): HH:mm:ss, with UTCを使用 (Use UTC)
 - Y軸** (Y-axis): 最小 (Minimum) and 最大 (Maximum) input fields
 - 凡例** (Legend): 非表示 (None), with 補完 (Completion) set to 直線 (Line)
 - 配色** (Color scheme): A color palette with a blue highlight on the first color.
 - 初期ラベル** (Initial label): 有効なデータが届く前に本文字列を表示 (Show this text before valid data arrives)
 - 種類** (Class): ウィジェット用のCSSクラス名 (オプション) (CSS class name for widget, optional)
 - 名前** (Name): 名前 (Name)

■ グラフの表示

- ◆ ダッシュボードに折れ線グラフが表示される

