irBoard

ユーザーズマニュアル

ito-soft-design, Inc.

2026-10-06

irBoard 3.0 新機能

irBoard 3.0では、画面デザインの効率化と多言語対応を中心に多くの新機能を追加しました。

新しい部品

マトリックス

テーブル形式で複数のデバイス値を一覧表示・編集できる部品です。

- 行数・列数を自由に設定
- セルごとにデバイスを割り当て
- デザインテンプレートで見た目をカスタマイズ
- ボーダーの追加・色の変更が可能

チュートリアル動画を見る

AIによる画面デザイン

AIインポート

描画ソフトなどで作成した画面イメージから、AIが部品を自動検出して配置します。

- 画像から部品の種類・位置・サイズを自動認識
- 描画ソフトで作成したデザイン画像を取り込んで画面を再現

多言語対応の強化

翻訳管理インスペクター

翻訳可能なテキストをインスペクターから一括で管理できるようになりました。

- ページ部品、メッセージ部品、メッセージセットの翻訳テキストを一覧表示・編集

- 翻訳の漏れを簡単に確認
- YAMLエクスポート/インポートによるテキストエディタやAIツールでの一括翻訳編集

ローカライズ機能の詳細は ローカライズ（多言語対応） をご覧ください。

新しいインスペクター

デバイス管理

プロジェクトで使用しているデバイスを一覧で確認・管理できるインスペクターです。

- 使用中のデバイスを一覧表示
- デバイスの重複や未使用を確認
- YAMLエクスポート/インポートによるデバイス割り当ての一括編集

データログ

データログ

PLCのデバイス値をCSVファイルに自動記録する機能を追加しました。

- 複数の設定を作成し、同時に記録可能
- 時系列モードとテーブルモードに対応
- インターバル記録またはビットトリガー記録
- CSV / Excel 形式で自動書き出し
- 実行中のセッション閲覧・エクスポート
- YAML / CSV による設定のインポート・エクスポート

YAMLプロジェクト インポート/エクスポート

YAMLプロジェクト形式

人間にも読みやすいYAMLテキストファイルでプロジェクトを作成・編集できるようになりました。AIによるプロジェクト自動生成にも対応します。

- プロジェクトをYAMLとしてエクスポートし、確認・バージョン管理・編集が可能
- YAMLファイルをインポートしてプロジェクトを作成・更新

・AIフレンドリーな形式と包括的なドキュメントにより、自動プロジェクト作成に対応
・デバイス一覧、ローカライズ、メッセージセット、全アイテム設定を含む

操作性の改善

複数選択モード

部品の選択方法が改善されました。

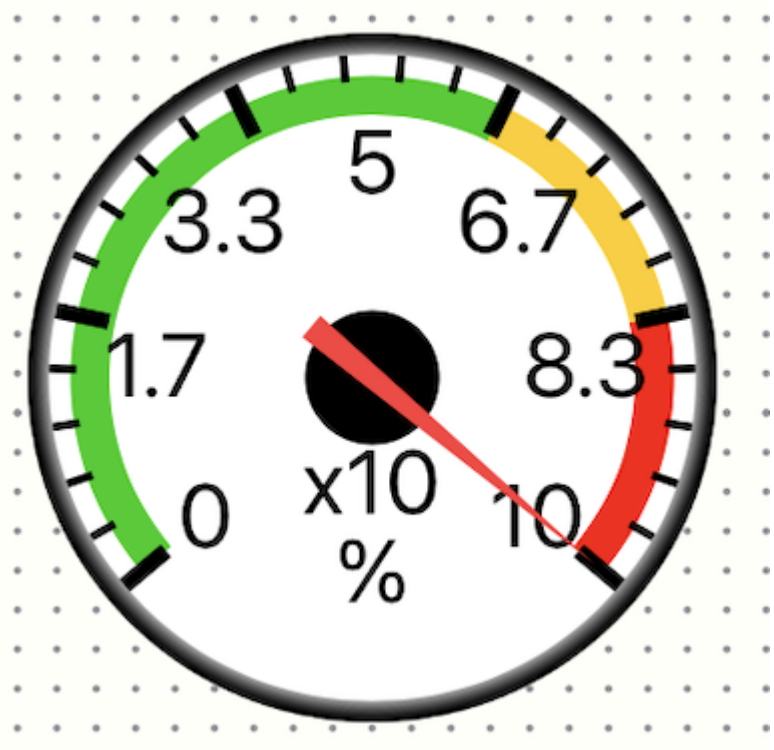
・通常モード: タップで単一選択
・マルチ選択モード: ダブルタップで複数選択モードに入り、タップで選択・解除
・空白エリアのタップで通常モードに戻る
・Cmd+Aで全選択（自動的にマルチ選択モードに切り替え）

部品の強化

メーターゾーン

メーターに色付きゾーンを設定して、値の範囲を視覚的に表示できるようになりました。

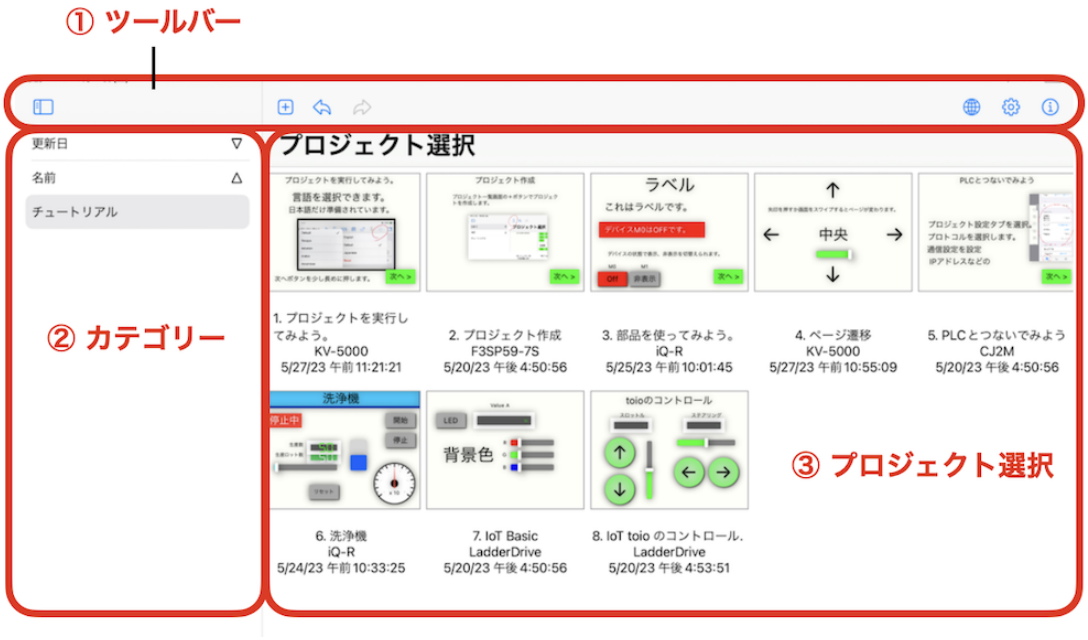
・複数のゾーンをカスタム色と値の範囲で定義
・メーター上に色付き円弧として表示
・個別追加または3ゾーンプリセット（緑/黄/赤）に対応
・YAMLインポート/エクスポートに完全対応


5
3.3
6.7
1.7
8.3
0
x10
%
10

プロジェクト一覧

irBoardを起動するとプロジェクト一覧画面が表示されます。
プロジェクト一覧画面は以下の部分に分かれます。

1. ツールバー
   画面上部にあるのがツールバーです。
2. カテゴリー
   画面左側にあるのがカテゴリー部です。
3. プロジェクト選択
   画面中央にあるのがプロジェクト選択部です。



ツールバー

- 
  カテゴリー表示を出したり隠したりします。
- 
  プロジェクトを 追加 します。
- 
  アンドゥ
  プロジェクトの 追加 、 削除 操作などを取り消します。

-  リドゥ
  アンドゥで取り消した操作をやり直します。
-  言語選択
  言語選択ができます。
  ローカライズ(多言語化)を参照
-  **irBoard**設定
  - 自動実行
    プロジェクトを指定すると、irBoard起動とともにプロジェクトが実行されます。
    ロック画面から戻った際も指定のプロジェクトが実行されます。
    ※Lite版では「接続の解除」アイテムを購入した場合に使用できるようになりました。(ver2.3.0より)
  - ブラウズモード
    ブラウズモードをONにすると一度プロジェクトを実行するともう編集画面に戻れなくなります。オペレーターが不用意に画面を編集してしまうのを防ぐことができます。
    編集画面に戻るためには 設定アプリでブラウズモードをOFFにします。
    ブラウズモードをONにし、iOSのアクセシビリティー機能のアクセスガイドを使用するとirBoardのみが動作できる状態になり、専用端末の様に使用できる様になります。
    ※Lite版では「接続の解除」アイテムを購入した場合に使用できるようになりました。(ver2.3.0より)
  - メッセージ表示 **(ver 2.0.3**以降**)**
    メッセージ表示では実行時下部に表示されるメッセージ表示の仕方を指定できます。
    - 表示

      メッセージ表示をします。
    - 非表示

      メッセージ表示しません。
    - 自動的に隠す

      一定時間表示したら消えます。
    - エラーのみ

      エラーのみ表示します。

-  情報
  - サポートサイト
    irBoardサポートサイトをプラウザーアプリで表示します。
  - ニュース
    irBoardに関するニュースをプラウザーアプリで表示します。
  - フィードバック
    アプリに対するフィードバックや問い合わせがありましたら選択してください。
    メールフォームが立ち上がりますので、内容を記入いただいて送信してください。
  - バグ報告
    設定アプリ で「バグ報告」をオンにするとこのメニューが表示されます。
    選択するとログファイルが添付されたメールフォームが表示されます。メールを送っていただけると不具合解析の手助けになります。
  - 謝辞
    irBoardではオープンソースのライブラリーやフォントを使わせていただいております。 各ライブラリーやフォントのライセンスを記載しています。
  - 利用規約
    irBoardの利用規約を記載しています。
  - バージョン
    irBoardのバージョンを記載しています。



irBoard設定



メッセージ表示の仕方



アクセスガイド開始



情報



カテゴリー

カテゴリーではプロジェクトの表示順の並べ替えを行うことができます。

- 更新日
  プロジェクトの更新日で並べ替えします。右側の▵▿を押すと昇順，降順を切り替えられます。
- 名前
  プロジェクトの名前で並べ替えします。右側の▵▿を押すと昇順，降順を切り替えられます。
- チュートリアル
  チュートリアルのプロジェクトが表示されます。バージョンアップで上書きされることがあります。その際変更した箇所が元に戻る場合があります。

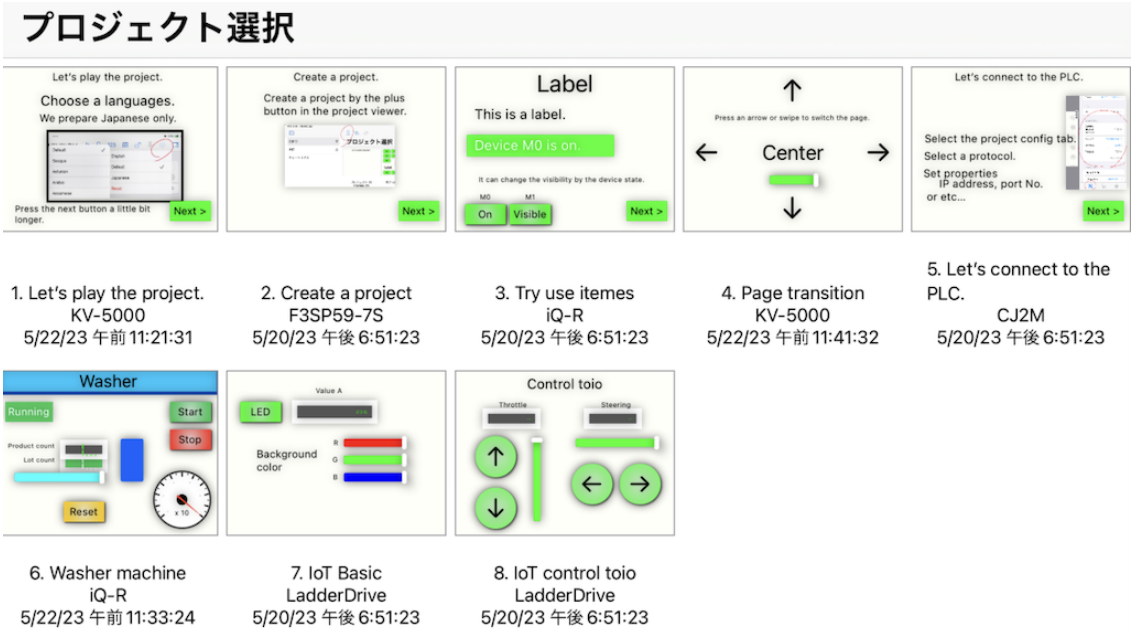


プロジェクト選択

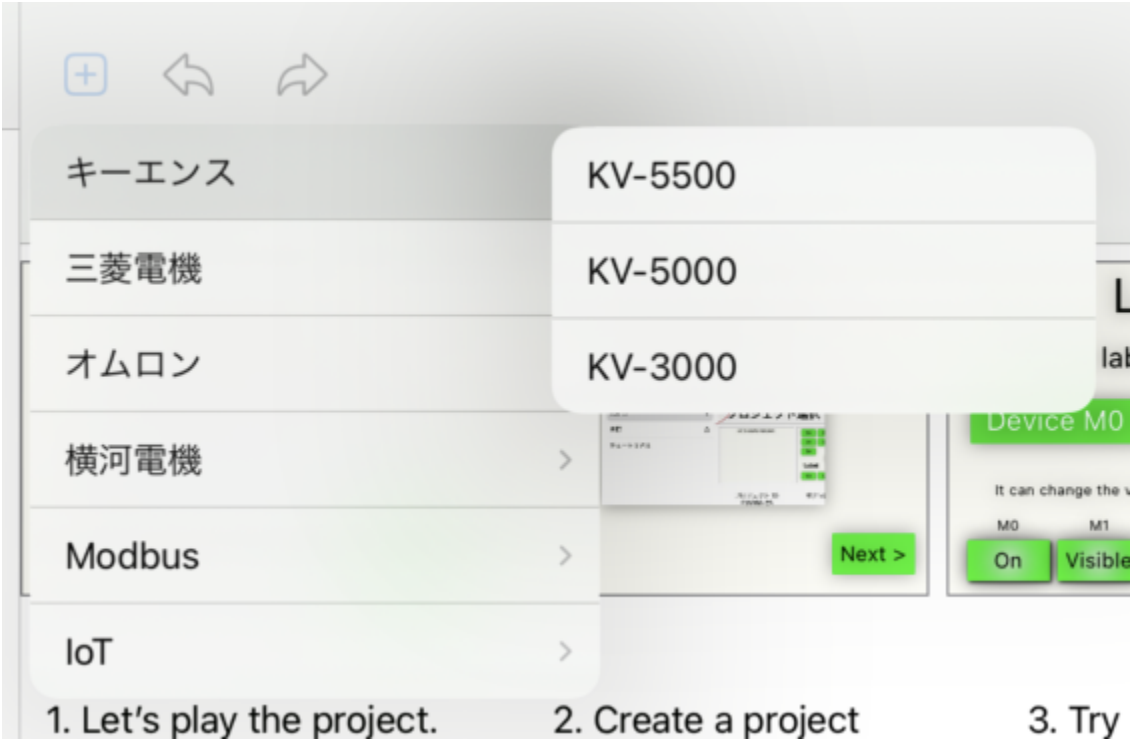
プロジェクトの一覧が表示されます。
プロジェクトをタッチし選択すると [編集画面](#) が表示されます。

プロジェクト追加

- ツールバーの  追加ボタンを押します。
- 表示されたメニューからPLCメーカーを選択します。
- 表示されたメニューからPLCの型式を選択します。



プロジェクト追加



プロジェクトエクスポート

- エクスポートしたいプロジェクトを長押しするとコンテキストメニューが表示されます。メニューの中からエクスポートを選択するとプロジェクトが書き出されます。

プロジェクト複製

- 複製したいプロジェクトを長押しするとコンテキストメニューが表示されます。メニューの中から複製を選択すると複製されます。

プロジェクトコンテキストメニュー



PLC変更

- PLCを変更したいプロジェクトを長押しするとコンテキストメニューが表示されます。メニューの中からPLC変更を選択しPLCの機種を選択するとPLCを変更することができます。 メーカーの違う**PLC**への変更もできます。
  使用しているデバイスは変更する前のままになっていますので再度設定してください。

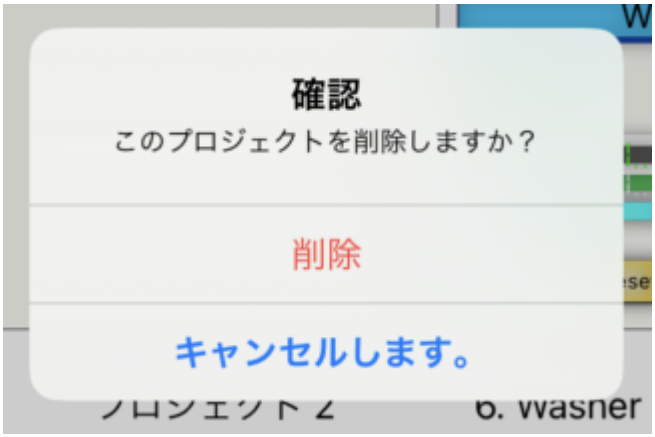
PLC変更



プロジェクト削除

- 削除したいプロジェクトを長押しするとコンテキストメニューが表示されます。メニューの中から削除を選択すると確認画面が表示されます。
- 確認画面で削除を選択すると削除されます。

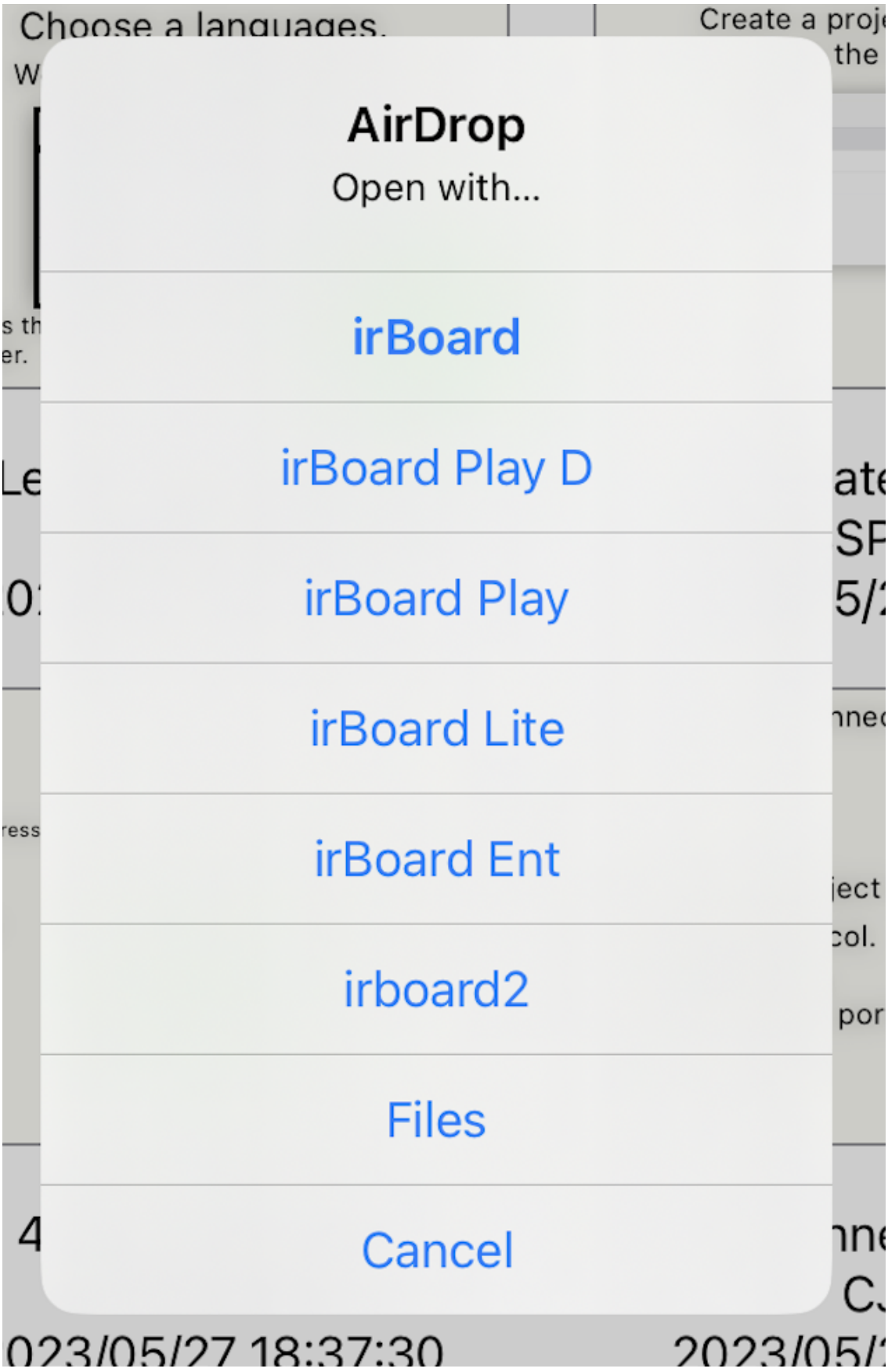
削除確認



プロジェクトインポート

- プロジェクトをインポートするにはAirDropやファイルアプリなどでから行います。
- AirDropでファイルが送られてくると確認画面が出ますのでirBoardやirBoard Liteなどインポートしたいアプリを選択します。

インポート


AirDrop
Open with...
irBoard
irBoard Play D
irBoard Play
irBoard Lite
irBoard Ent
irboard2
Files
Cancel

編集

編集画面は起動時に表示されるプロジェクト一覧画面で編集対象となるプロジェクトを選択すると表示されます。

編集画面では大きく4つの部分に分かれます。

1. ツールバー
   画面上部にあるのがツールバーです。
   ツールバーではプロジェクト一覧ページや実行画面に移動したり、コピーアンドペーストボタンなどの画面の移動や編集に必要なボタンが配置されています。
2. ページ選択
   画面左側にあるのがページ選択部です。
   ページの追加、削除や編集対象となるページを選択できます。
3. インスペクター
   画面右側にあるのがインスペクター部です。
   プロジェクトや選択中のページ、部品の設定、部品の一覧などが行えます。
4. プレビュー
   画面中央にあるのがプレビュー部です。
   画面の見た目を確認できます。



ツールバー

ツールバーでは編集で使う機能がボタンに割り当てられています。

左側アイコン

- 

プロジェクト一覧画面に戻ります。

- 

ページ選択表示を出したり隠したりします。

-  アンドゥ

操作を取り消します。
アンドゥ、リドゥはプロジェクト単位、ページ単位で行われます。どちらに対して行われるかはインスペクター部下部(右下)のタブ選択で変わります。
左側のプロジェクト設定が選選択されている場合はプロジェクトに対して行われます。
真ん中のオブジェクト設定または右側のオブジェクト一覧が選択されている場合は編集中のページに対して行われます。



-  リドゥ

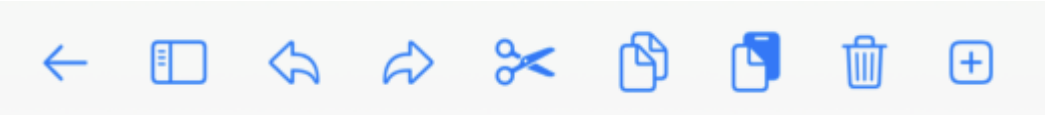
アンドゥで取り消した操作をやり直します。

-  カット

選択中の部品をクリップボードにコピーし、画面上からは削除します。

-  コピー
  選択中の部品をクリップボードにコピーします。
-  貼り付け
  クリップボードにコピーされた部品から複製し画面に追加します。
-  削除
  選択中の部品を削除します。
-  追加
  部品を追加します。部品は ラベル 、 ボタン 、 値 、 スライダー 、 進捗 、 メーター 、 Webページ 、 メッセージ 、 マトリックス があります。また、 AIインポート で画面イメージから部品を自動検出して配置できます。

ツールバー左側アイコン



部品の追加 

右側アイコン

-  プレイ
  実行画面に移行します。
-  電球
  押す毎に  デバイスがONの場合や  OFFの場合の見た目を確認できます。
- 
  プレビューに表示する値や文字を設定できます。
  [値の確認を参照]
-  グリッド
  部品を配置する際のガイドとなるグリッドを設定します。
  - 有効
    ONにするとグリッドが表示され部品がグリッドに沿って配置されます。部品のノブをドラッグしサイズを変更して離した時にグリッドの位置に調整されます。部品を移動した場合はセンター位置がグリッドに調整されます。この時、部品の幅や高さがグリットの奇数倍になっていると隅の位置がグリッドから半分ずれます。
  - スタイル
    グリッドのガイド表示をドットとラインから選択できます。
  - ピクセル
    ガイドの幅を指定します。この幅で位置が調整されます。
-  ロック
  選択中の部品の位置を  固定し移動できなくしたり  解除したりします。
-  ピン
  ページのスクロールを制御します。ページを選択した状態で操作できます。
   固定するとスクロールできなくなります。部品を操作する場合にスクロールが有効になっていると部品の操作をしようとしたのにページが移動してしまうとい

うのを防ぎます。


解除するとページのスクロールができます。拡大して確認したい時に移動しながら確認できる様になります。

- 
向き

画面の向きを切り替えます。
編集中のページの画面方向が両方になっている場合に切り替えができます。
設定は プロジェクト設定 または ページ設定 で行います。

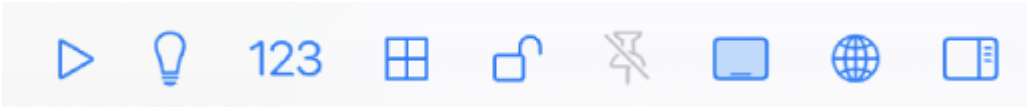
- 
地球

言語選択ができます。
ローカライズ(多言語化)を参照

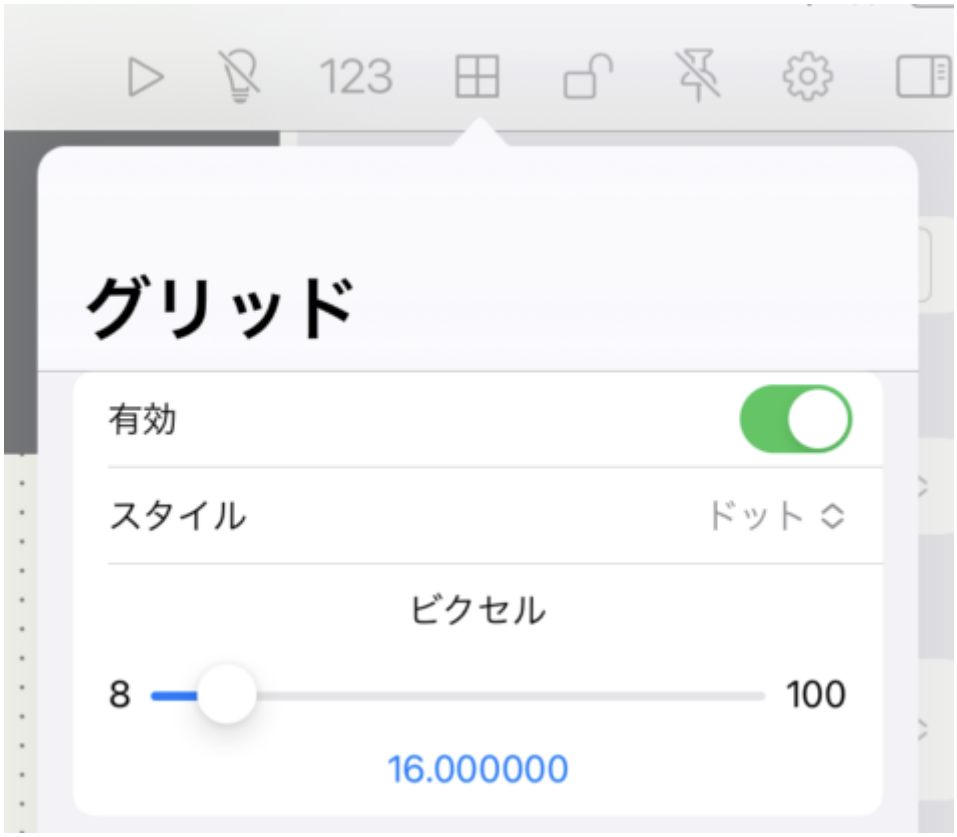
- 

インスペクター部を出したり隠したりできます。

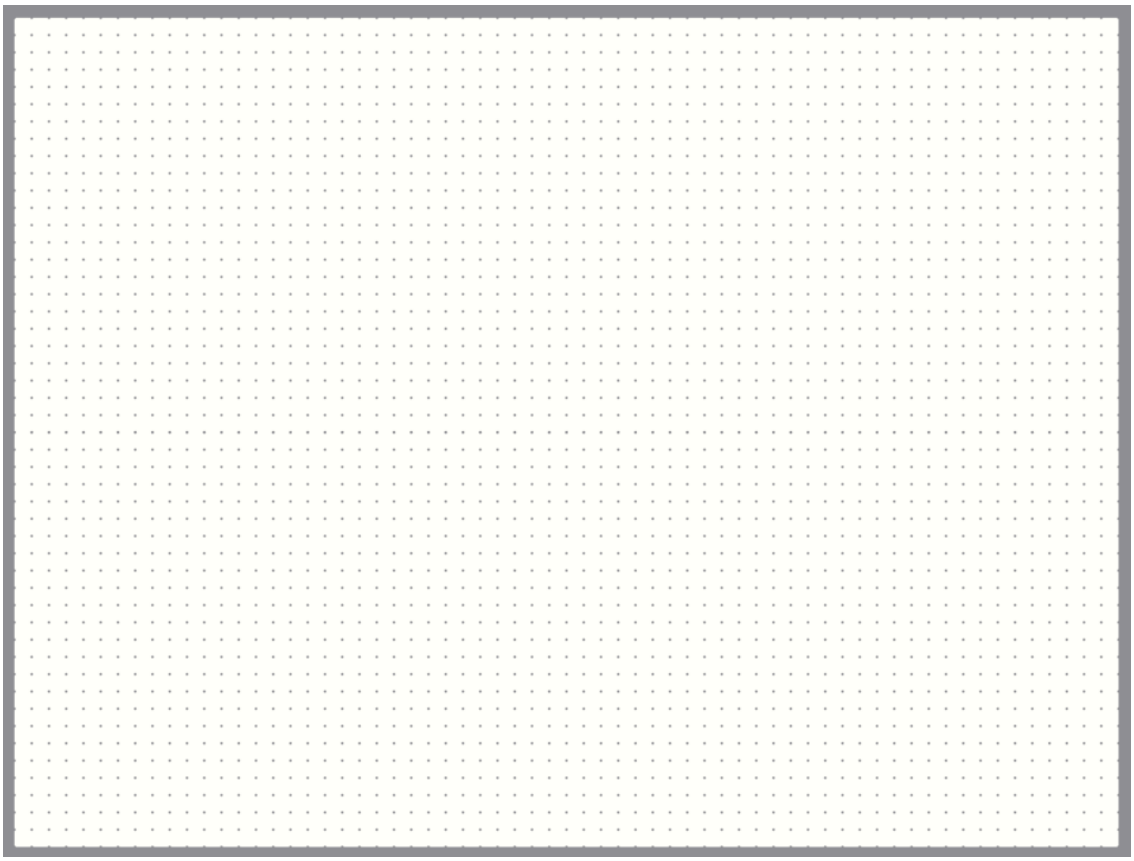
ツールバー右側アイコン



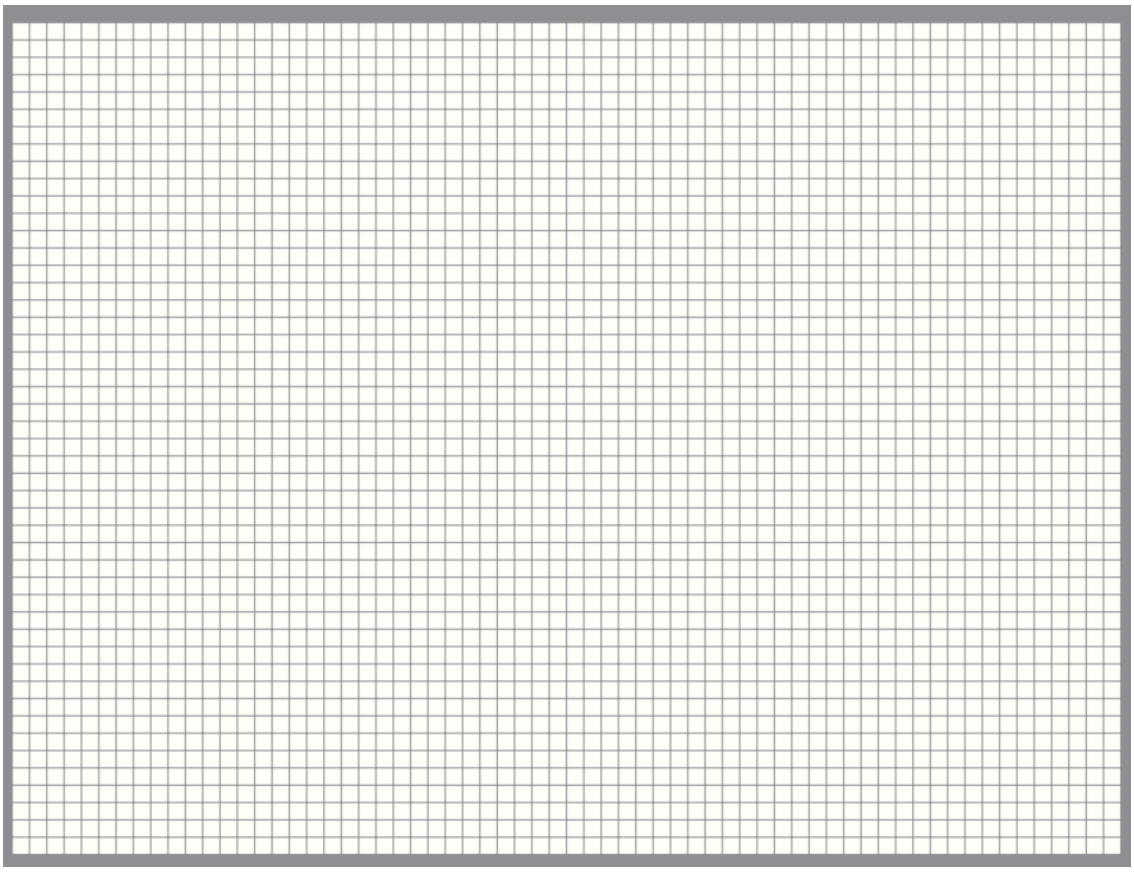
グリッド設定



ドットグリッド



ライングリッド



ページ選択

ページ選択ではページの一覧、選択、追加、削除ができます。
各ページ右隅の数字はページ番号になります。

選択

ページを選択するには該当ページをタッチします。
選択されたページの枠が青色になり、選択されていることを示します。
プレビュー画面に選択されたページが描画されます。

追加

ページを追加するには  追加アイコンをタッチします。
ページが一覧の末尾に追加され、プレビュー画面に描画されます。ページ番号は追加前最後のページの続き番号になります。
ページ番号を変更する場合はインスペクター部で行います。



削除

ページを削除するには削除したいページを押し続けると削除メニューが表示されます。
削除メニューを選択すると確認表示がでますので、削除を選択すると削除されます。
最後の1ページになった時は削除できません。



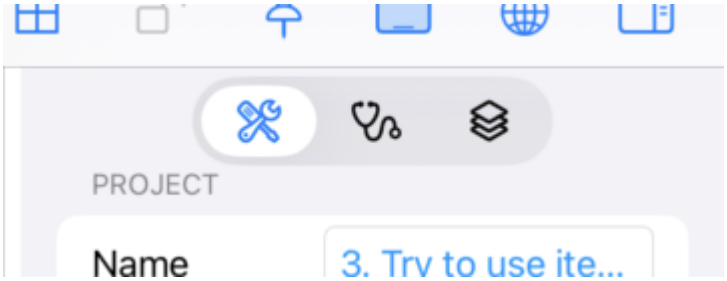
インスペクター

インスペクターではプロジェクトやページ、部品の設定などを行えます。

インスペクターでは5つの機能に別れていてタブボタンで切り替えます。

- プロジェクト設定
  プロジェクト設定ではプロジェクトに関する設定を行います。
- オブジェクト設定
  オブジェクト設定ではページや部品に関する設定を行います。
- 部品一覧
  部品一覧でページ内の部品の一覧、選択、並べ替えを行います。
- デバイス管理
  デバイス管理ではプロジェクト内で使用されているPLCデバイスの一覧表示、検索、一括置換を行います。
- 翻訳管理
  翻訳管理ではプロジェクト内のローカライズ可能な文字列の一覧表示、翻訳進捗確認、翻訳編集を行います。

インスペクタータブ切り替え 

iOS 18以降 

iOS 18からタブの位置が上に変更になりました。このドキュメントではタブの位置を下と表現しているところがありますが、上に読み替えてください。

プロジェクト設定

プロジェクト設定ではプロジェクトに関する設定を行います。

インスペクター下部の  ボタンを押すとプロジェクト設定を表示します。

プロジェクト

・名前
プロジェクトの名前を設定します。



PLC

・**Plc**
PLCの型式を選択します。
該当する型式がない場合はデバイスの構成が近いPLCを選択してください。選択したPLCで扱うデバイス範囲内で使うことができます。



接続

PLCとの接続の仕方を指定します。
PLC毎の設定については PLC接続 をご覧ください。

- 通信プロトコル
  PLCと通信する際に使用する通信プロトコルを選択します。
  シミュレートはPLCと接続せずirBoard単体でシミュレート動作させて確認したり、プレゼンテーションに用いたりする時に使います。
- **HOST/IP**
  PLCのホスト名かIPアドレスを設定します。
- ポート番号
  PLCと接続する際のポート番号を設定します。
- プロトコル
  Ethernetプロトコルを設定します。現在サポートされているのは**TCP**のみです。

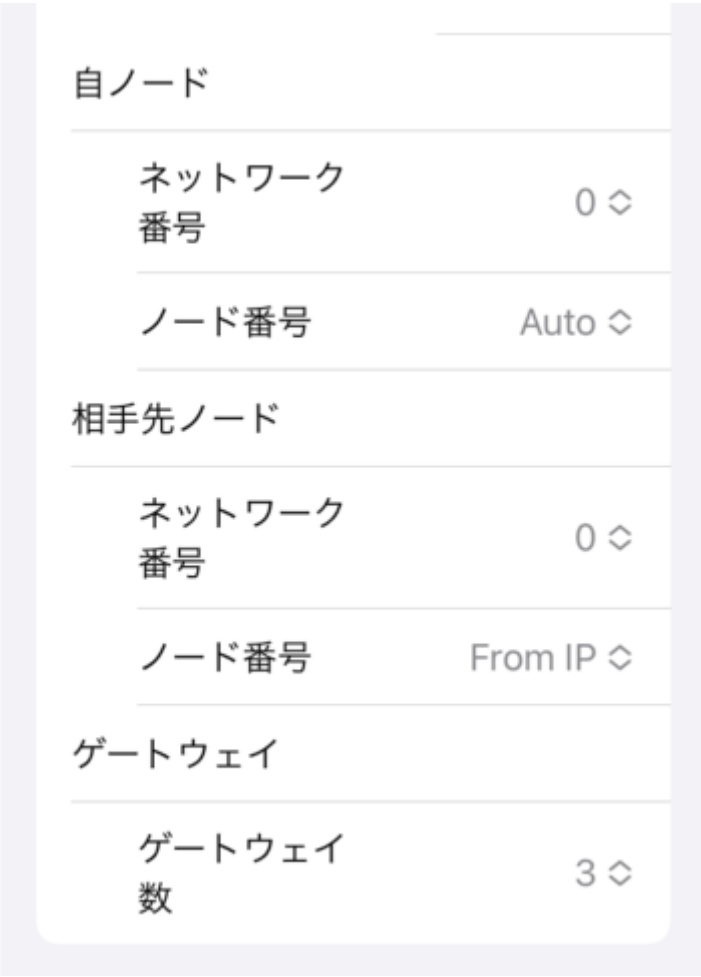
オムロンの場合は次の拡張設定ができます。

- 自ノード
  - ネットワーク番号
    自ノードのネットワーク番号を設定します。通常は0です。
  - ノード番号
    自ノードのノード番号を設定します。Autoを設定すると通信開始時に自動的に決定されます。
- 相手先ノード
  - ネットワーク番号
    相手先ノードのネットワーク番号を設定します。
  - ノード番号
    相手先ノードのノード番号を設定します。From IPを設定するとIPアドレスの末尾番号をノード番号とします。
- ゲートウェイ
- ネットワーク番号
  許容ゲートウェイ通過数を設定します。

横河電機の場合は次の設定があります。

- ネットワーク番号
  ネットワーク番号を指定します。





オムロンの場合



横河電機の場合

相互通信

irBoardとPLC間の制御をするためのデバイスを指定します。
右側のトグルスイッチをオフにするとその項目は相互通信として使用しないことを選択できます。

- **irBoard→PLC**
  irBoardからPLCに対する制御を行うデバイスです。
  - ページ番号
    irBoardで現在表示しているページ番号を書き込むPLCデバイスを指定します。
  - ステータス
    irBoardから通信状態を書き込むPLCデバイスを指定します。

BIT	内容
0	OFF: 切断中 ON: 通信中
1	ヘルシービット OFF / ONを繰り返します。 OFFまたはONのまま変化しない時は通信障害が発生しています。
2	ヘルシービット OFF / ONを繰り返します。 OFFまたはONのまま変化しない時は通信障害が発生しています。 弊社iOSアプリioTouch2、irPanelと互換性を保つためにbit1と同じ動作をします。
3 - 15	将来のために予約されています。

- **irBoard←PLC**
  PLCからirBoardに対する制御を行うデバイスです。
  - ページ番号
    PLCからページ切り替えを指示するデバイスを指定します。
  - ステータス
    PLCからirBoardへ制限を指示するデバイスを指定します。

BIT	内容
0	OFF: 制限なし ON: 書き込み禁止制限
1 - 15	将来のために予約されています。

・権限

複数デバイスでPLCに対して操作指示すると予想外の動作をする場合があり危険です。
そのため**irBoard**では操作(書き込み)できるデバイスを権限領域をつかつて**1**台に限定する様にしています。

権限で指定されたデバイスを先頭とする**10**ワード**(**権限領域**)**を使います。

以下は権限取得のルールです。

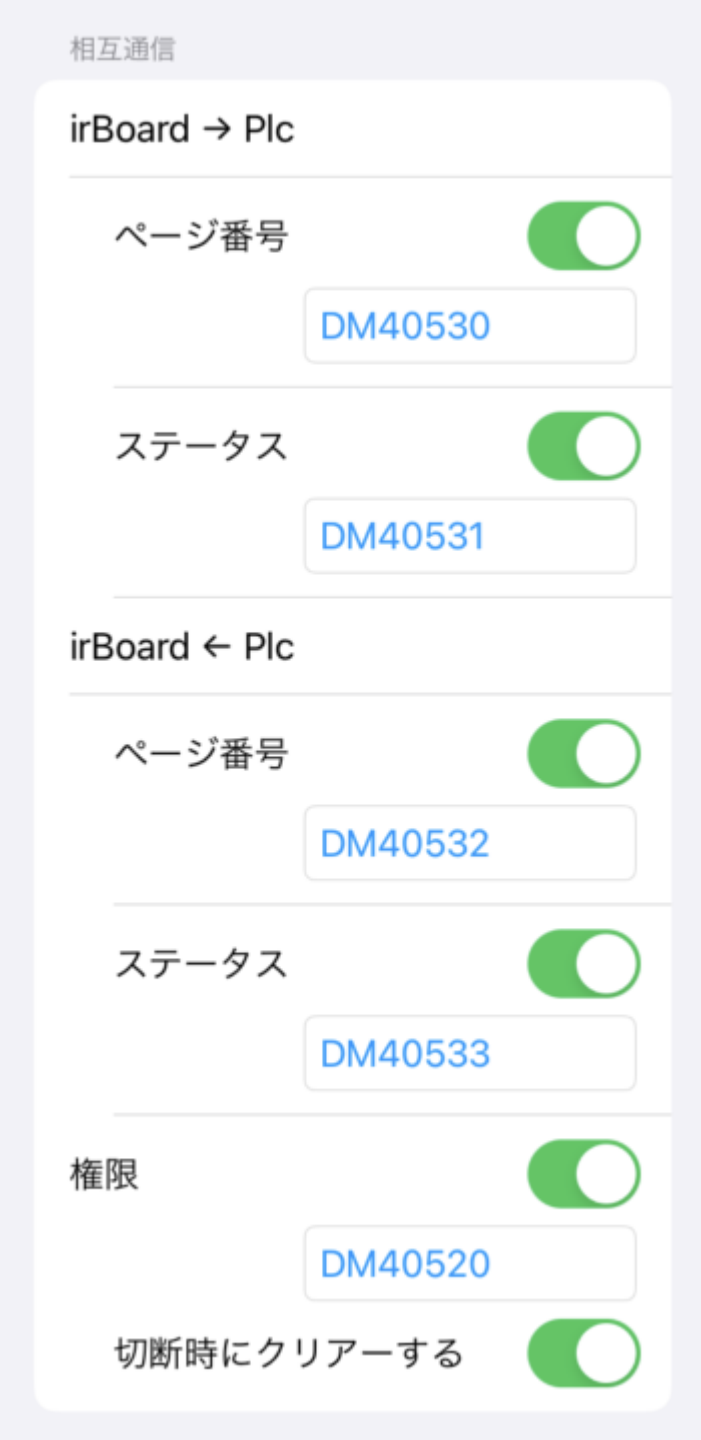
- 権限領域のデータが全て0の場合は自分のデータを書き込み権限を取得します。
- 権限領域のデータが自分のデータと一致する場合は既に権限を取得していいるのでそのま継続します。
- 権限領域のデータが自分のデータと一致しない場合は他のデバイスが既に権限を取得しているので、操作できず読み込みだけできます。
- 権限を取得後に切断する場合は権限データを0にクリアーし他のデバイスに権限を開放します。「切断時にクリアーする」で0クリアーしない選択ができます。

PLC側で必要となる処理

- 必要に応じてPLC起動時に権限データを0に初期化するルーチンを入れてください。
- 通信障害時はirBoardから権限データを0に戻せない場合がありその対策が必要になる場合があります。

・切断時にクリアーする **(ver 2.0.3**以降**)**

オンにすると切断時に権限領域全てを0で書き換え権限を解放し他のデバイスが接続できる様にします。 他のデバイスでの操作を禁止したい場合はオフにし権限を解放しない様にしてください。



画面サイズ

画面サイズを選択します。作画はiPadで行なって実行はiPhoneで行う時はiPhoneの画面サイズで作成するということができます。
指定した画面サイズとiOSデバイスの画面サイズが異なる場合は画面サイズに合わせて拡大または縮小されます。

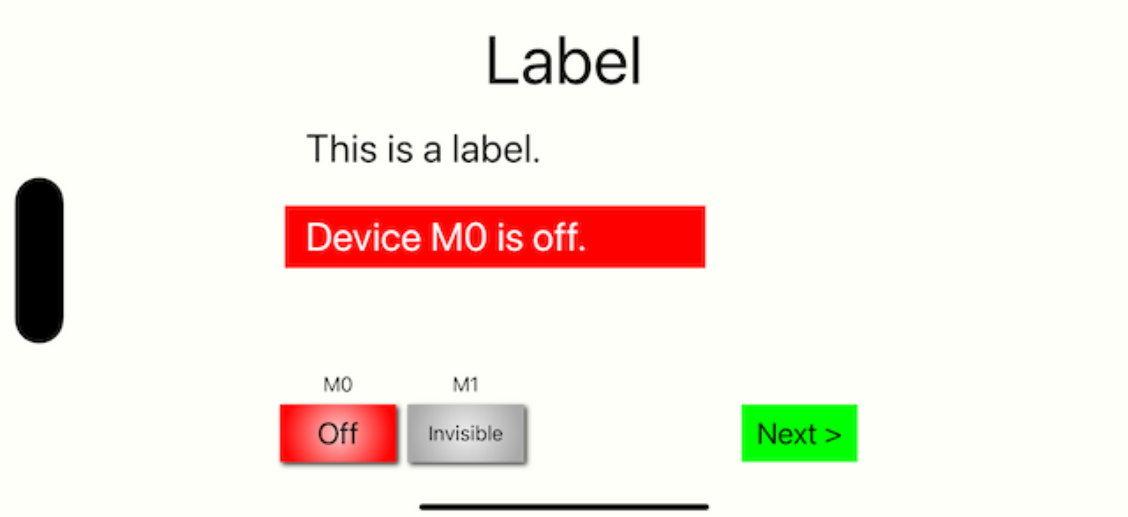
スケール

画面サイズで指定したサイズとデバイスの画面サイズが異なる際に、画面に合わせるための拡大縮小の仕方を選択します。

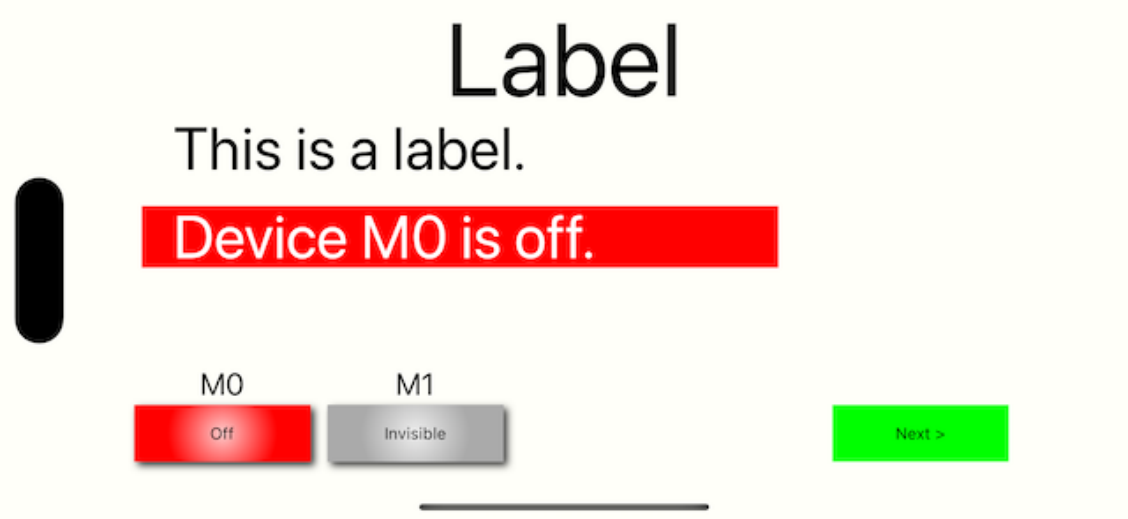
- 比率を保つ
    - 画面の比率を保つ様に拡大縮小されます。
- 画面に合わせる(デフォルト)
    - 画面のサイズに合わせて拡大縮小され、画面いっぱいに表示されます。



比率を保つ

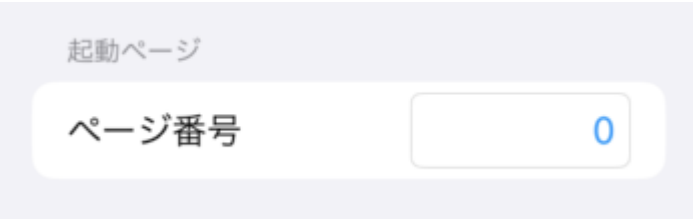


画面に合わせる



起動ページ

起動ページのページ番号を設定します。
実行すると指定されたページを表示します。
その後速やかに相互通信でページ番号に指定されているデバイスで指定されたページに移動します。



デフォルト

プロジェクト内のデフォルト値を設定します。
ページや部品が追加された際にデフォルト値が設定されます。

- 更新間隔
  PLCのデバイスを読み込むインターバルを設定します。通信に時間がかかる場合は実際の時間はこれより長くなることがあります。
- ボタン遅延時間
  ボタンを押してから反応するまでの時間を秒数で設定します。iOSデバイスは触るとすぐ反応してしまうため、不用意に触って誤操作してしまうのを防ぎます。反応を早めたい場合は短くすると早くなりますが、誤操作も増えることに注意してくだ

さい。
ボタン設定で遅延時間を設定した場合はボタンの設定が優先されます。
・スライダー遅延時間
スライダーを押してから反応するまでの時間を秒数で設定します。iOSデバイスは触るとすぐ反応してしまうため、不用意に触って誤操作してしまうのを防ぎます。反応を早めたい場合は短くすると早くなりますが、誤操作も増えることに注意してください。
スライダーで遅延時間を設定した場合はスライダーの設定が優先されます。
・向き
画面レイアウトの向きを指定します。
ページ設定で向きを指定している場合はページ設定が優先されます。
    ◦横向き
      横向きのレイアウトになります。
    ◦縦向き
      縦向きのレイアウトになります。
    ◦両方
      横向き、縦向き両方のレイアウトになります。
両方に設定した場合はデバイスの向きに合わせて縦と横のレイアウトが切り替わります。
横向きまたは縦向きにした場合はデバイスの向きが変わっても縮尺を変更してそのままのレイアウトで表示します。
・ボタンスタイル
ボタンを追加した時に適用されるスタイルを指定します。
スタイルは ボタンの項 を参照してください。
・背景色
ページの背景色を指定します。
全てのページに適用されますが、ページで指定した場合はページの指定が優先されます。
・背景画像
ページの背景画像を指定します。
全てのページに適用されますが、ページで指定した場合はページの指定が優先されます。
画像選択の仕方は 画像選択 をご覧ください。



デフォルトフォント

ラベル、ボタン、値、メーターなどで使用するデフォルトのフォントを指定します。

- テキスト
  テキストに使用するデフォルトのフォントを指定します。
- 値
  数値表示に使用するデフォルトのフォントを指定します。

フォントの指定の仕方は ラベルの項 を参照してください。



データログ

PLCのデバイス値をCSVファイルに記録する機能です。

- データログ
  トグルスイッチでデータログの有効･ 無効を切り替えます。
- 設定
  設定画面を開きます。

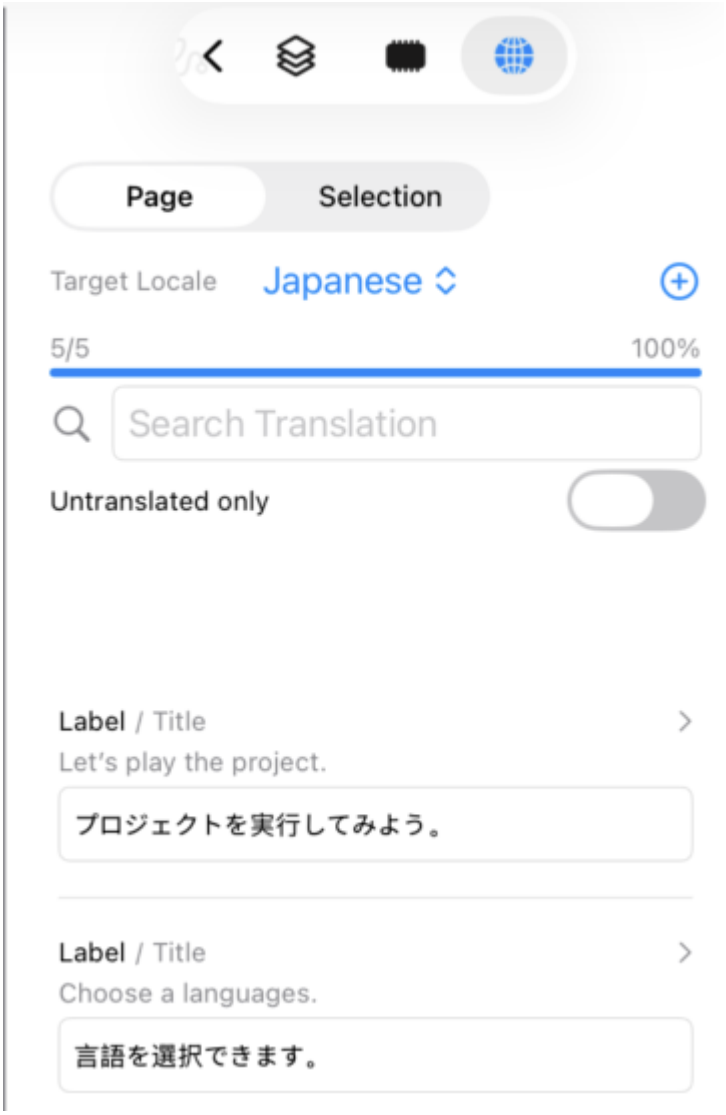
詳しくは データログ をご覧ください。



翻訳管理

右サイドバーの「翻訳管理」タブで、プロジェクト内のすべてのテキスト（タイトル、単位、メッセージなど）を一覧表示し、まとめて翻訳を管理できます。

詳しくは ローカライズ(多言語化) をご覧ください。



オブジェクト設定

オブジェクト設定ではページや部品の設定を行います。

インスペクター下部の  ボタンを押すとオブジェクト設定を表示します。

対象

オブジェクト設定ではプレビュー部で選択されているページまたは部品を対象として設定します。

ページ内の部品を押すと周りに青い丸のノブが8個表示され選択中であることを示します。

ページ内の部品が何もないところを押すとページの外周が青い線で表示されページが選択中であることを示します。

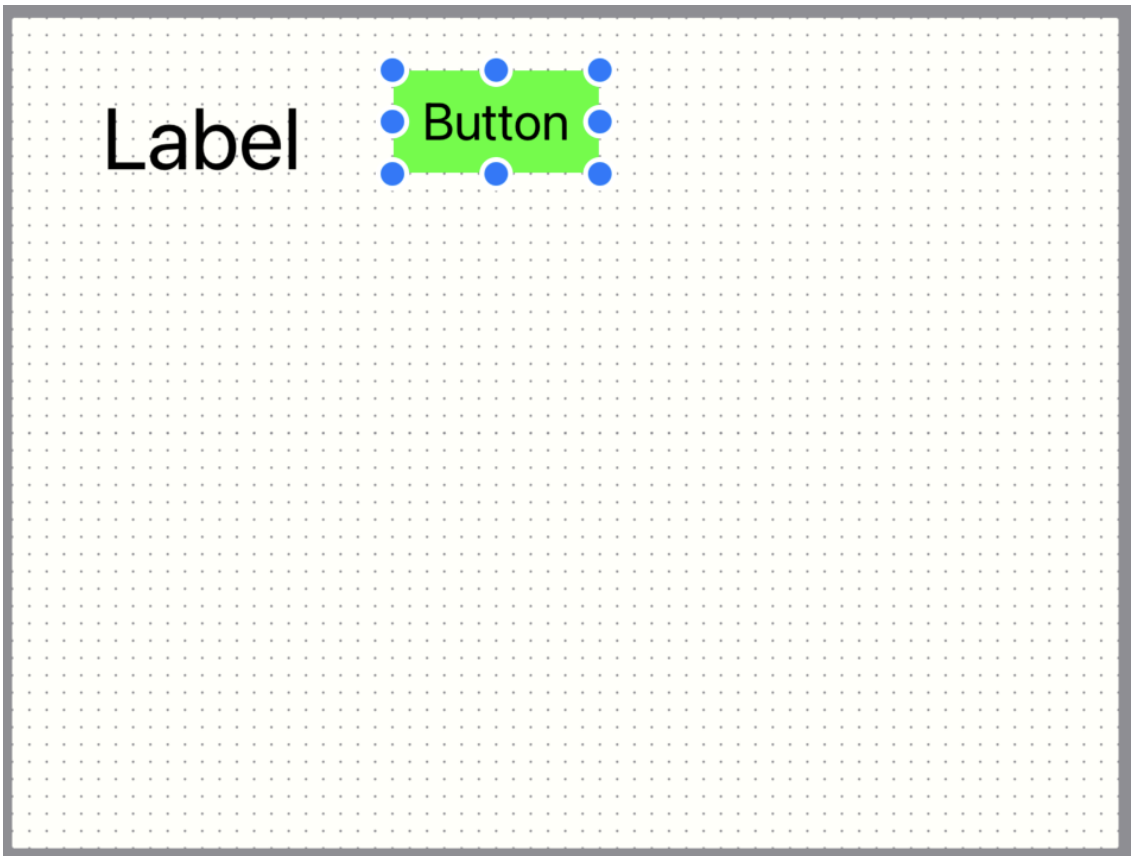
オブジェクト設定に選択中のページまたは部品の設定内容が表示されます。

設定

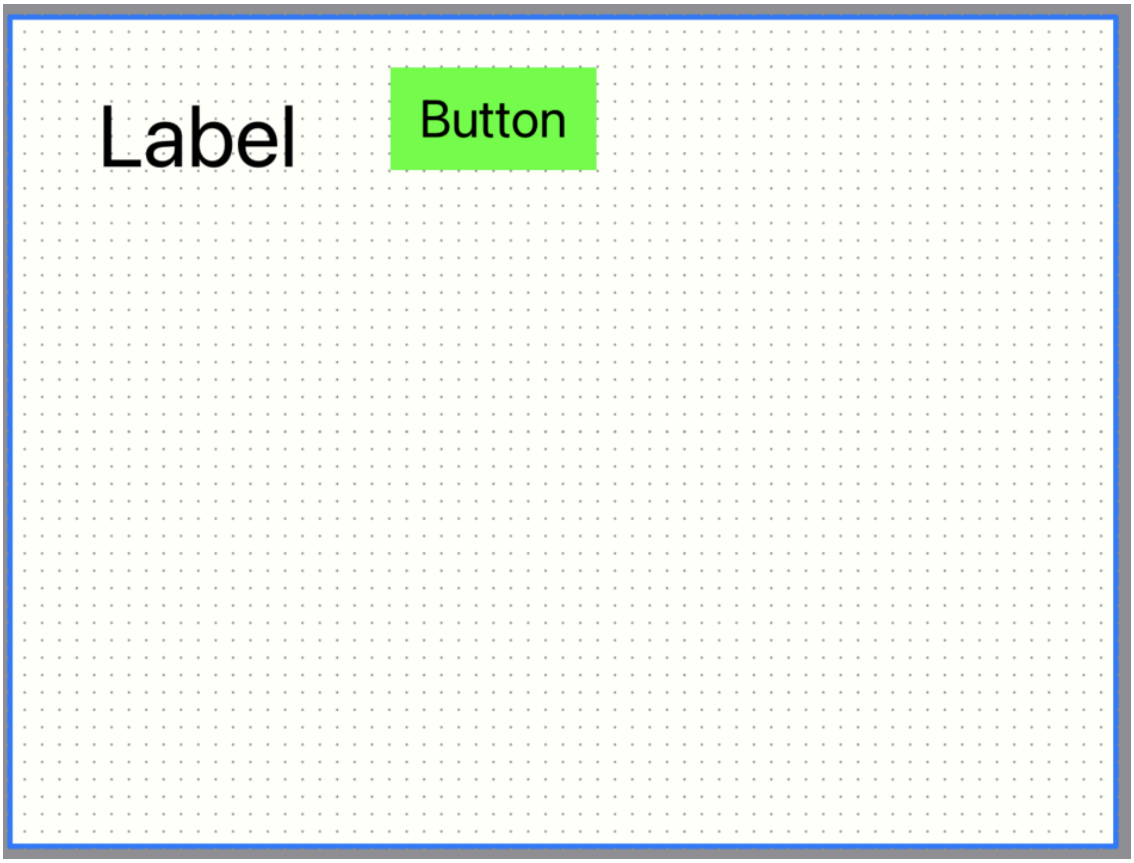
各設定についてはそれぞれの項をご覧ください。

- ページ
- 部品
  - 共通項目
    各部で共通している部分は共通項目をご覧ください。
  - ラベル
  - ボタン/ランプ
  - 値
  - スライダー/進捗
  - メーター
  - Webページ
  - メッセージ

ボタンが選択された状態



ページが選択された状態



オブジェクト設定


ボタンスタイル
スタイル
プレーン
デバイス
ボタン
デバイス名
モード
モーメンタリー
遅延
デフォルト
ページリンク
プロジェクトリンク
ランプ
デバイス名
インターロック
デバイス名
表示
表示
タイトル
配置
センター
通常
Button
フォント
デフォルトフォント
色

部品一覧

ページとページ内の部品が一覧されます。
部品が重なってなかなか選択できない場合にリストから選択することで簡単に選択できます。

選択されている部品には ✓ チェックマークが付きます。

 並べ替えアイコンをドラックすると重なりの順番を変更できます。一覧の上側が重なりの上になります。

マトリックスアイテム

マトリックスアイテムはツリー構造で表示されます。
マトリックス行の右側には生成された子アイテムの数が表示されます。
マトリックス名の左側の三角アイコンをタップすると、生成されたすべての子アイテム（ヘッダー、データセル）が表示されます。
子アイテムはインデントされて表示され、マトリックスに属していることが分かります。


ページ
ページ
0
アイテム
進捗
ラベル
矢印を押すか画面をスワイプするとページが変わります。
ボタン
↓
ボタン
↑
ボタン
→
ボタン
←
ラベル
中央

デバイス管理

デバイス管理ではプロジェクト内で使用されているPLCデバイスの一覧を表示します。

インスペクター下部の  ボタンを押すとデバイス管理を表示します。

機能

デバイス一覧

プロジェクト、ページ、または選択中のアイテムで使用されているすべてのデバイス名が一覧表示されます。
各デバイス名をタップすると、そのデバイスが使用されている箇所（ページ、部品、フィールド）の詳細が表示されます。
使用箇所をタップすると、その部品に直接ジャンプしてオブジェクト設定で編集できます。

範囲の選択

画面上部のセグメントコントロールで表示範囲を切り替えられます。

- 選択
  現在選択中のアイテムで使用されているデバイスのみ表示
- ページ
  現在のページで使用されているデバイスのみ表示
- プロジェクト
  プロジェクト全体で使用されているデバイスを表示

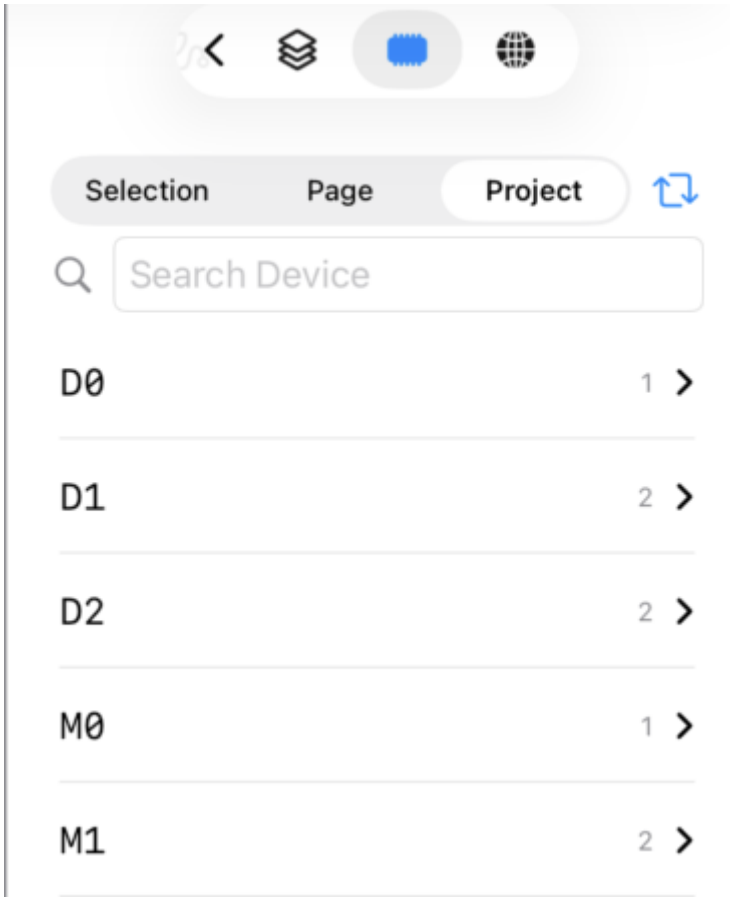
検索

検索フィールドでデバイス名を検索できます。
前方一致検索により、入力した文字で始まるデバイス名が絞り込まれます。

デバイス置換ウィザード

画面上部の  ボタンをタップするとデバイス置換ウィザードが開きます。
デバイス名のプレフィックスを一括で変更したり、複数のデバイスを一度に置換できま

す。
変更内容はプレビューで確認してから適用できるため、安全に一括変更を行えます。



デバイス管理

YAMLエクスポート/インポート

デバイス管理ではデバイス一覧をYAML形式でエクスポート・インポートできます。 テキストエディタやAIツールを使ったデバイス割り当ての一括編集に便利です。

エクスポート

共有ボタンをタップすると、現在のデバイス一覧をYAMLファイルとしてエクスポートします。 エクスポート範囲は現在の表示範囲（選択、ページ、プロジェクト）に従います。

インポート

インポートボタンをタップしてYAMLファイルを読み込み、プロジェクト内のデバイス割り当てを更新します。 UUIDが一致する既存アイテムのデバイス名とタイトルを上書き更新します。 UUIDが一致しないエントリはスキップされるため、新しいアイテムの追加は

できません。 アイテムの追加が必要な場合は YAMLプロジェクト形式 でのインポートを使用してください。 インポート後に更新されたアイテム数が表示されます。

YAML形式

エクスポートされるYAMLは以下の構造です。

```
# Device List - プロジェクト名

items:
 # ページ1
  - item_uuid: "C1325E46-..."
    page: "ページ1"
    item: Button
    title: スタート
    alternate_title: ""
    devices:
      - field: device
        device: D100
      - field: switch
        device: M0
      - field: lamp
        device: M0
      - field: visible
        device: ""
      - field: interlock
        device: ""
```

フィールド

フィールド	説明
`item_uuid`	ページアイテムのUUID (インポート時のマッチングに使用)
`page`	ページ名 (参考情報)
`item`	部品の種類 (Button, Label, ValueField 等)
`title`	部品のタイトル (インポート時に更新される)
`alternate_title`	代替タイトル (インポート時に更新される)
`devices`	デバイスフィールド割り当てのリスト

デバイスフィールドの種類

field	対象部品	説明
`device`	値、メーター	メインデバイス値
`switch`	ボタン	スイッチ (書き込み) デバイス
`lamp`	ボタン	ランプ (インジケーター) デバイス
`interlock`	ボタン	インターロックデバイス
`visible`	全部品	表示/非表示制御デバイス
`editable`	値	編集許可デバイス
`label`	ラベル	ラベル表示デバイス
`max_value`	値	最大値デバイス
`min_value`	値	最小値デバイス
`step_value`	ステップ値	ステップ値デバイス
`figures`	メーター	桁数指定デバイス
`major`	メーター	大目盛りデバイス
`minor`	メーター	小目盛りデバイス
`message`	メッセージ	メッセージ条件デバイス

活用のヒント

- デバイス一覧をエクスポートし、テキストエディタで一括編集してからインポートで変更を適用
- AIを使って仕様書からデバイス割り当てを生成し、YAMLとしてインポート
- プロジェクト全体のデバイス使用状況の確認用としてエクスポート

翻訳管理

翻訳管理ではプロジェクト内のローカライズ可能な文字列の一覧を表示し、翻訳の編集や進捗確認を行えます。

インスペクター下部の  ボタンを押すと翻訳管理を表示します。

機能

翻訳一覧

プロジェクト内のすべてのローカライズ可能な文字列が一覧表示されます。
各項目にはデフォルト値（基準となる文字列）と翻訳入力フィールドが表示されます。
翻訳フィールドに直接入力すると、自動的に保存されます。

対象ロケールの選択

画面上部のロケールピッカーで翻訳対象のロケールを選択します。
翻訳進捗（翻訳済み/全体の割合）がプログレスバーで表示されます。
ロケールメニューからロケールを追加してから使用してください。

範囲の選択

画面上部のセグメントコントロールで表示範囲を切り替えられます。

- 選択
  現在選択中のアイテムの翻訳可能な文字列のみ表示
- ページ
  現在のページの翻訳可能な文字列のみ表示
- プロジェクト
  プロジェクト全体の翻訳可能な文字列を表示

検索と絞り込み

検索フィールドで文字列を検索できます。
「未翻訳のみ」トグルをオンにすると、未翻訳の項目だけを表示できます。

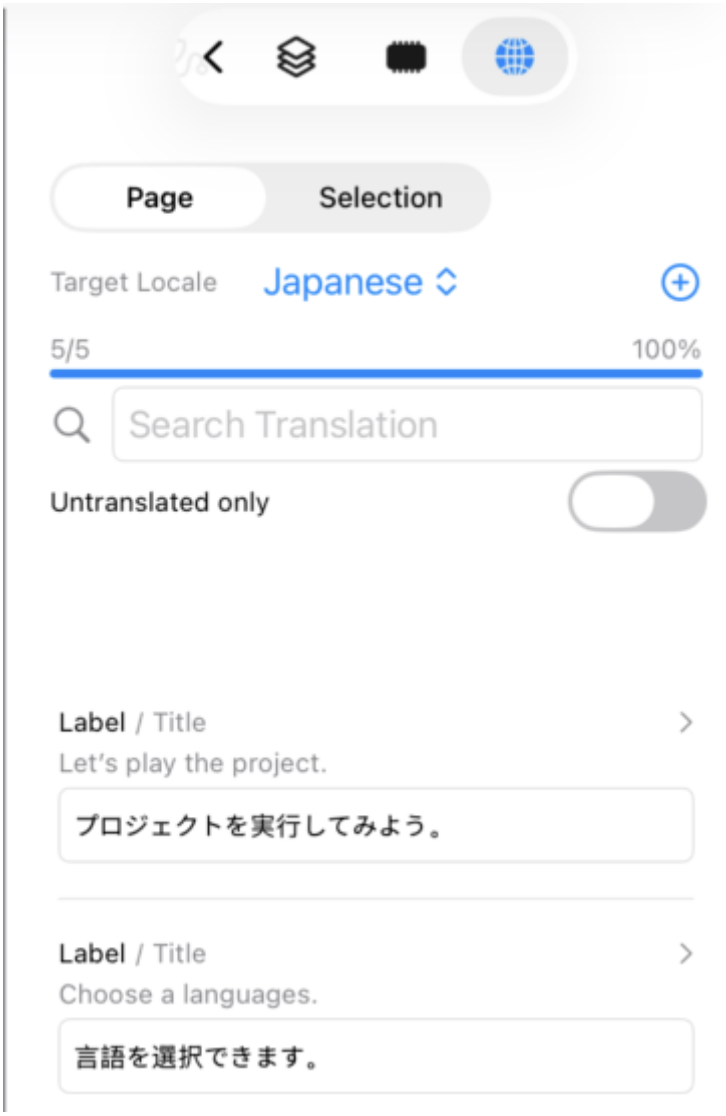
翻訳可能なフィールド

以下のフィールドが翻訳対象となります。

- 部品のタイトル
- 部品の代替タイトル
- 単位（値、メーター）
- メッセージ
- 詳細メッセージ
- メッセージセットのタイトル
- メッセージセットのサマリー

ナビゲーション

項目のヘッダー（部品名とフィールド名）をタップすると、その部品に直接ジャンプしてオブジェクト設定で編集できます。



翻訳管理

YAMLエクスポート/インポート

翻訳管理では翻訳データをYAML形式でエクスポート・インポートできます。 テキストエディタ、スプレッドシート、AI翻訳ツールを使った翻訳の一括編集に便利です。

エクスポート

共有ボタンをタップすると、翻訳データをYAMLファイルとしてエクスポートします。 設定済みのすべてのロケールがエクスポートに含まれます。 エクスポート範囲は現在の表示範囲（選択、ページ、プロジェクト）に従います。

インポート

インポートボタンをタップしてYAMLファイルを読み込み、プロジェクト内の翻訳を更新します。 UUIDが一致する既存アイテムの翻訳テキストを上書き更新します。 UUIDが一致しないエントリはスキップされるため、新しいアイテムの追加はできません。 インポート後に更新された翻訳エントリ数が表示されます。

YAML形式

エクスポートされるYAMLは以下の構造です。

```
# Localization - プロジェクト名

locales:
  - en
  - ja

pages:
  - page: "ページ1"
    items:
      - uuid: "C1325E46-..."
        type: Button
        title: スタート
        fields:
          title:
            en: Start
            ja: スタート
          alternate_title:
            en: Stop
            ja: ストップ

messages:
  - uuid: "FBB22E64-..."
    set: "アラーム"
```

```
    fields:
      title_:
        en: Alarms
        ja: アラーム
      summary_:
        en: Alarm list
        ja: アラーム一覧
    items:
      - device_name: M100
        value: 1
        default: モーター過負荷
        field: message_
        translations:
          en: Motor overload
          ja: モーター過負荷
```

構造

セクション	説明
`locales`	エクスポートに含まれるロケール識別子のリスト（例: en, ja, zh_CN）
`pages`	ページごとにグループ化された翻訳
`pages[].items[].uuid`	ページアイテムのUUID（インポート時のマッチングに使用）
`pages[].items[].fields`	翻訳フィールド（title, alternate_title, unit）と口ケール別の翻訳テキスト
`messages`	メッセージセットの翻訳
`messages[].fields`	メッセージセットフィールド（title_, summary_）と口ケール別の翻訳テキスト
`messages[].items`	個別メッセージアイテムの翻訳

翻訳可能なフィールド

フィールドキー	対象	説明
`title`	ページ部品	部品のタイトル
`alternate_title`	ページ部品	代替タイトル（ボタン等）
`unit`	ページ部品	単位ラベル（値、メーター）

フィールドキー	対象	説明
`title_`	メッセージセット	メッセージセットのタイトル
`summary_`	メッセージセット	メッセージセットのサマリー
`message_`	メッセージアイテム	メッセージテキスト
`detail_message`	メッセージアイテム	詳細メッセージテキスト

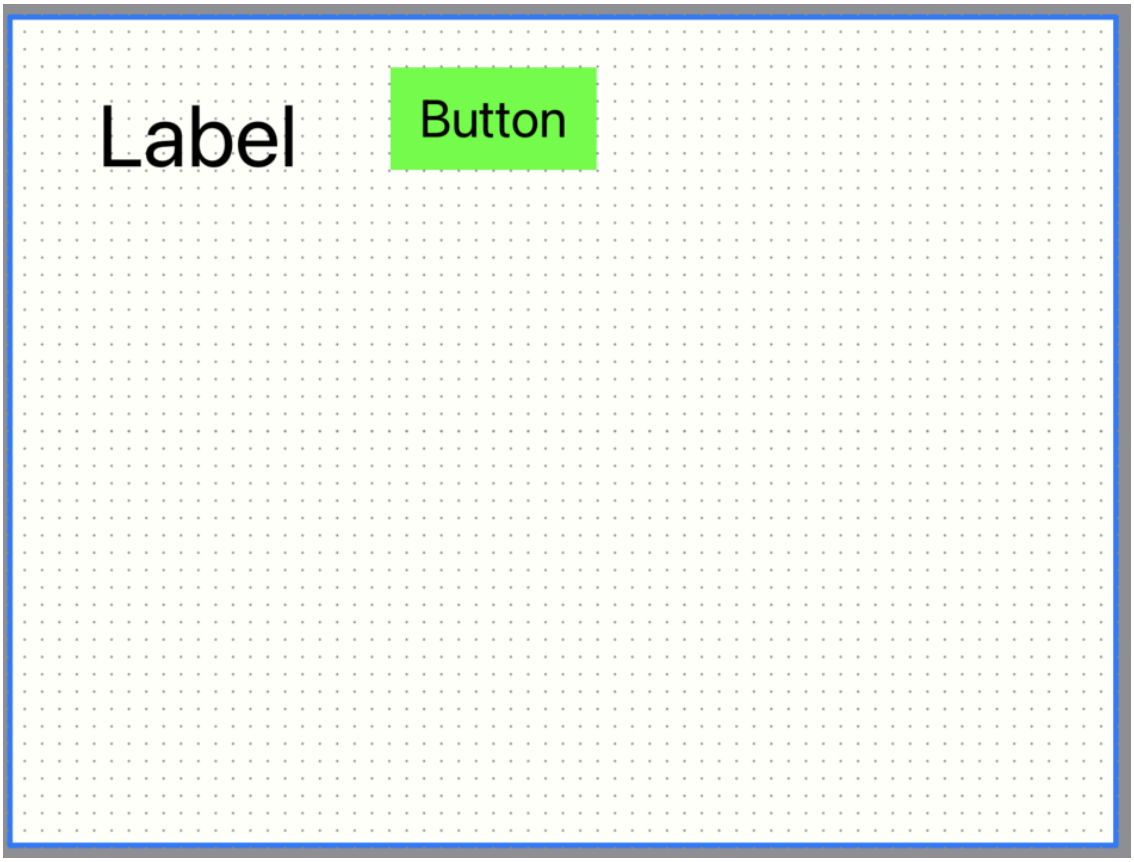
活用のヒント

- 翻訳データをエクスポートし、翻訳者やAIサービスに送信してから、完成した翻訳をインポート
- AIを使って全文字列を複数言語に一括翻訳
- プロジェクト全体の翻訳完了状況の確認用としてエクスポート

ページ設定

プレビュー部で部品が何もない箇所を押すとページが選択され、オプジェクト設定にページの設定項目が表示されます。

ページが選択された状態



一般

・ページ番号
ページのページ番号を設定します。
重複した場合はどちらが使用されるかは分かりません。重複時はページ一覧に注意マークが表示されます。
・タイトル
ページのタイトルを設定します。
現時点では登録できるだけで用途はありません。

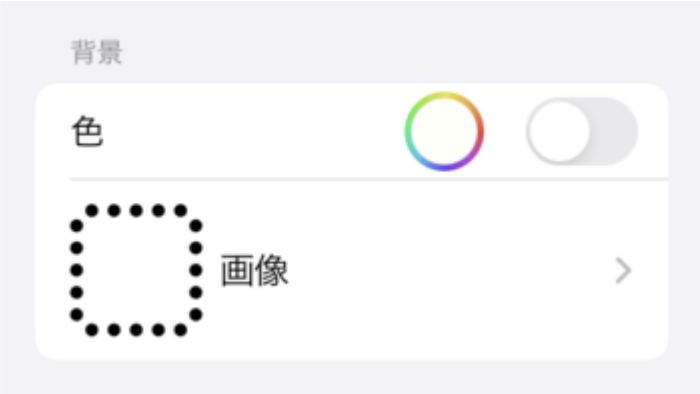
・向き
画面レイアウトの向きを指定します。
ページ設定で向きを指定している場合はページ設定が優先されます。
◦プロジェクト設定に従う
プロジェクトで設定した向きになります。
◦横向き
横向きのレイアウトになります。
◦縦向き
縦向きのレイアウトになります。
◦両方
横向き、縦向き両方のレイアウトになります。
両方に設定した場合はデバイスの向きに合わせて縦と横のレイアウトが切り替わります。
横向きまたは縦向きにした場合はデバイスの向きが変わっても縮尺を変更してそのままのレイアウトで表示します。





背景

・色
背景色を設定します。
色を選択すると右端のスイッチがオンになり色を背景に適用します。
解除したい場合は右端のスイッチをオフにします。オフの時は プロジェクトで設定した背景色 になります。
・背景画像
ページの背景画像を指定します。
なし(点線で囲まれた四角)を選択した場合は プロジェクトで設定した背景画像 になります。
画像選択の仕方は 画像選択 をご覧ください。



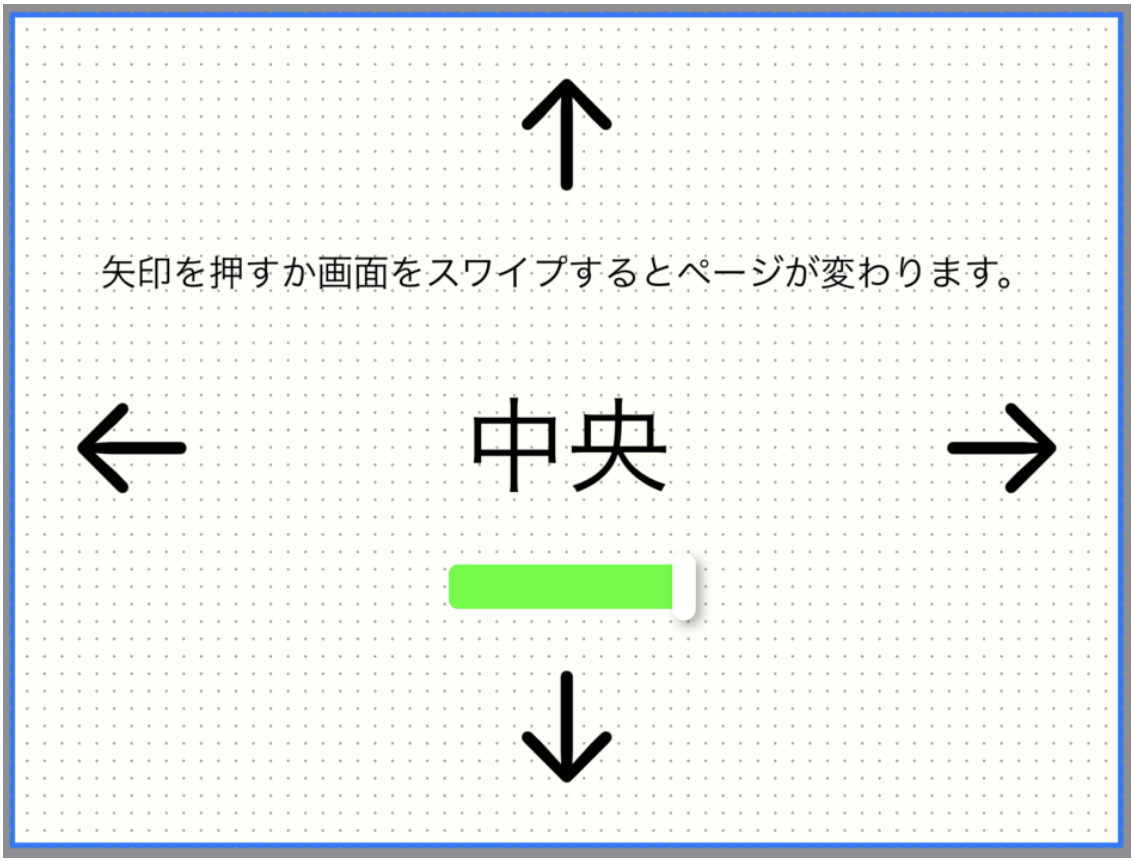
画面遷移

画面遷移を設定するとフリックでページ間を移動できます。
※Lite版で実機接続の制限がかかっている時は移動できません。

・上ページ
下にフリックした時に切り替わるページを指定します。設定したページの下ページが未設定の場合は選択中のページに設定され、相互移動ができる様になります。
・左ページ
右にフリックした時に切り替わるページを指定します。設定したページの右ページが未設定の場合は選択中のページに設定され、相互移動ができる様になります。
・右ページ
左にフリックした時に切り替わるページを指定します。設定したページの左ページが未設定の場合は選択中のページに設定され、相互移動ができる様になります。
・下ページ
上にフリックした時に切り替わるページを指定します。設定したページの上ページが未設定の場合は選択中のページに設定され、相互移動ができる様になります。



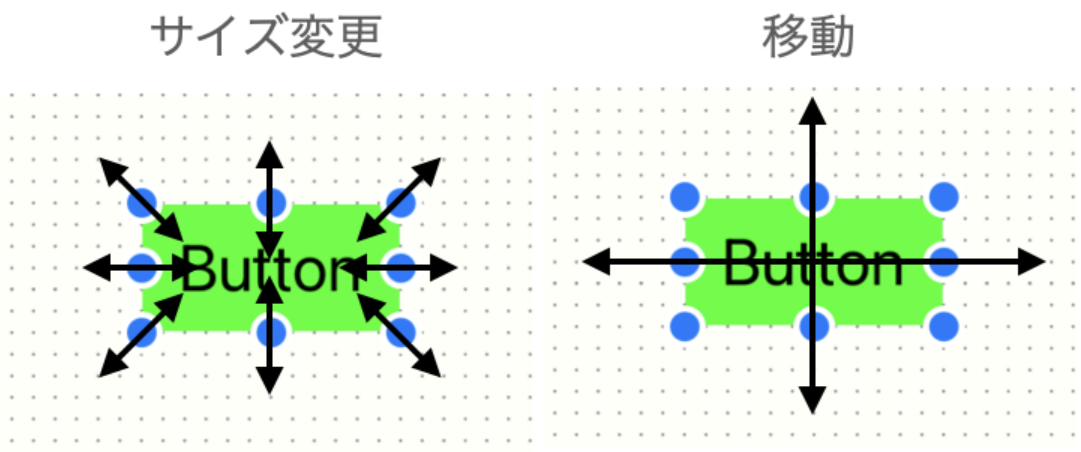
チュートリアルのページ遷移サンプルプロジェクト



部品

ページ内の部品をタッチすると部品が選択されます。
選択させた部品には青丸のノブが表示されます。ノブを移動すると部品のサイズを変更することができます。
部品の中央部をドラックすると部品の移動ができます。

- 共通項目
  各部で共通している部分は共通項目をご覧ください。
- ラベル
- ボタン/ランプ
- 値
- スライダー/進捗
- メーター
- Webページ
- メッセージ



部品共通項目

部品インスペクターの共通している項目について説明します。

表示/非表示

表示項目で部品の表示/非表示の設定ができます。

- 表示
    - 部品を表示します。
- 非表示
    - 部品を非表示にします。
- デバイス
    - 指定したデバイスの状態で表示、非表示を切り替えます。
    - 正論理の場合はデバイスがONの場合に表示されます。
    - 負論理の場合はデバイスがOFFの場合に表示されます。





タイトル

タイトルでは部品に表示するテキストを設定できます。

- 配置
  テキストの配置を選択します。
  - 左上
  - 上
  - 右上
  - 左
  - 中央
  - 右
  - 左下
  - 下
  - 右下
- アクティブ**(ON) -** 通常タイトル
  部品に表示するテキストを入力します。
- インアクティブ**(OFF) -** 代替タイトル
  ランプデバイスを持つ部品（ボタンなど）では、ランプがOFF（インアクティブ）のときに表示される代替タイトルを設定できます。

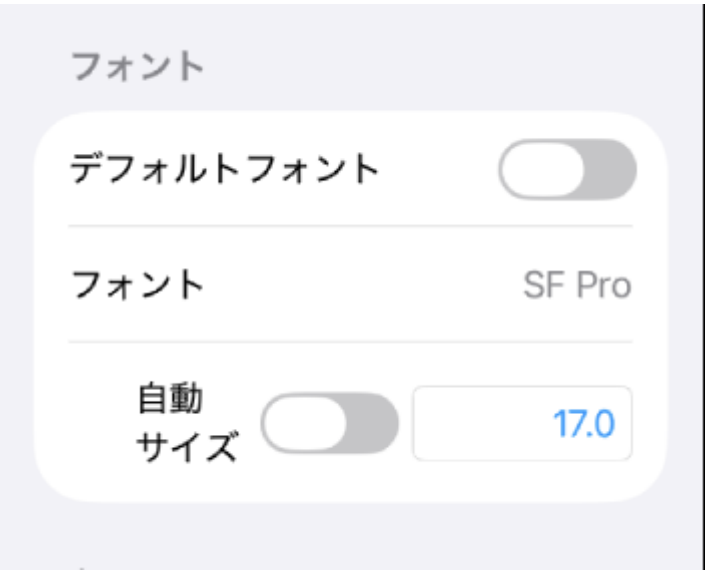
タイトルの多言語化については ローカライズ(多言語化) をご覧ください。



フォント

フォントでは部品のテキスト表示に使用するフォントを設定できます。

- デフォルトフォント
  ONにするとプロジェクトのデフォルトフォントが使用されます。個別にフォントを変更する場合はOFFにします。
- フォント
  デフォルトフォントがOFFの場合に表示されます。フォント名とサイズを指定できます。
- 自動サイズ
  ONにするとフォントサイズが部品に合わせて自動調整されます。



スタイル

スタイルでは部品のバックグラウンド表示を指定することができます。

- 形
  背景の形を選びます。
  - 長方形
  - 角丸長方形
  - 丸
  - カプセル
  - 楕円
- 線幅
  枠線の幅を指定します。
- 余白
  背景と部品との余白を指定します。

・角丸半径
角丸長方形の場合の角の丸みを指定します。
・線色
枠線の色を指定します。
・背景色
背景の色を指定します。
・画像
背景画像を指定します。 画像選択 をご覧ください。







画像選択

画像を選択する項目では、画像一覧画面が表示されます。 過去に選択したことがある画像が一覧表示されます。この中に使用する画像がある場合はタッチすることで選択されます。

- ライブラリーから画像を選択する場合は追加(+)ボタンを押して選択することができます。
  ※但し、現バージョンではHEICファイルを選択しても適用になりません。
- 画像を使用しない場合は、なし(点線で囲まれた四角)を選択します。

画像を選択すると、以下の詳細設定が表示されます。

- 表示モード
  画像の表示方法を選択します。
  - フィット — 画像のアスペクト比を維持し、部品内に収まるように表示します。
  - フィル — 画像のアスペクト比を維持し、部品全体を覆うように表示します。はみ出た部分はクリップされます。
  - 引き伸ばし — 画像を部品のサイズに合わせて引き伸ばします。アスペクト比は維持されません。
- 拡大率
  画像の拡大縮小倍率を指定します。1.0が等倍です。

・オフセット **X / Y**
画像の表示位置をずらします。
・不透明度
画像の不透明度を指定します。1.0で完全に不透明、0.0で完全に透明になります。
・画像の配置
画像の重なり順を指定します。
  ◦背景の後ろ — 画像は背景色の後ろに表示されます。背景色を半透明にすると画像が透けて見えます。
  ◦背景の前 — 画像は背景色の前、テキストの後ろに表示されます。
  ◦コンテンツの前 — 画像はテキストなどのコンテンツの前に表示されます。アイコンのオーバーレイなどに使えます。



追加



なし





位置

位置では部品の位置や大きさを指定することができます。

- X: 部品配置のX座標
- Y: 部品配置のY座標
- 幅: 部品の幅
- 高さ: 部品の高さ
- 正方形: ONにすると幅と高さが同じになります。
- ロック: ONにすると位置がロックされ移動できなくなります。



複数選択

複数の部品を同時に選択して操作できます。

選択モード

複数選択には2つのモードがあります。

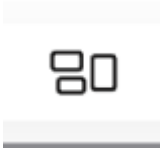
- 通常モード
  部品をタップすると、その部品だけが選択されます（既存の選択は解除されます）。
- マルチ選択モード
  部品をダブルタップすると、マルチ選択モードに入ります。このモードでは、部品をタップすると選択に追加され、既に選択されている部品をタップすると選択が解除されます。
  モード中はオブジェクトインスペクターの右上に「複数選択」バッジが表示されます。
- モード解除
  ページの空白部分をタップすると、マルチ選択モードが解除され通常モードに戻ります。
- 全選択
  キーボードの Cmd+A で現在のページの全部品を選択できます。自動的にマルチ選択モードに入ります。

複数選択時の操作

・移動
選択中の部品をドラッグすると、全ての選択部品が一緒に移動します。
・ロック
インスペクターでロックのON/OFFを切り替えると、全ての選択部品に適用されます。
・グループ化
ツールバーの「グループ」ボタンで、選択した部品をグループにまとめることができます。
・グループ解除
ツールバーの「グループ解除」ボタンで、グループを解除できます。
・コピー・削除
選択中の全部品に対してコピーや削除を実行できます。

グループ選択

グループ化されている部品をタップすると、グループ全体が選択されます。 グループ内の個別の部品を選択するには、部品をダブルタップしてグループ編集モードに入ります。



グループ化 / グループ解除ボタン

ラベル

ラベルは文字を表示します。

デバイス

・ランプ
　ここで指定したデバイスの状態で見た目が変化します。
・正論理
　デバイスがONの時にアクティブで指定した表示になります。OFFの場合はインアクティブで指定した表示になります。
・負論理
　デバイスがOFFの時にアクティブで指定した表示になります。ONの場合はインアクティブで指定した表示になります。



タイトル

ラベルに表示する文字を指定します。

・配置
　文字の位置寄せを指定します。
・通常
　表示させる文字を指定します。(ランプデバイスを指定しない場合)
・インアクティブ
　ランプで指定したデバイスがインアクティブの場合の文字を指定します。アクティブ、インアクティブで表示が共通の場合に入力を省略できます。
・アクティブ
　ランプで指定したデバイスがアクティブの場合の文字を指定します。





フォント

文字のフォントを指定します。

- デフォルトフォント
  ONにするとデフォルトのフォントを使用し、文字サイズは幅に合わせて調整されます。
- フォント
  フォントを指定します。
- 自動サイズ
  ONにすると文字サイズは幅に合わせて調整されます。
- 文字サイズ
  自動サイズをOFFにすると文字サイズを指定できます。文字サイズとラベル幅、高さを調整すると文字が折り返されたりします。



色

ラベル背景色、枠線の色を指定します。
インアクティブ時、アクティブ時の背景色、枠線色を指定します。



確認

上部のツールバーの　　アイコンで見た目を確認できます。



インアクティブ表示



アクティブ表示

ボタン/ランプ

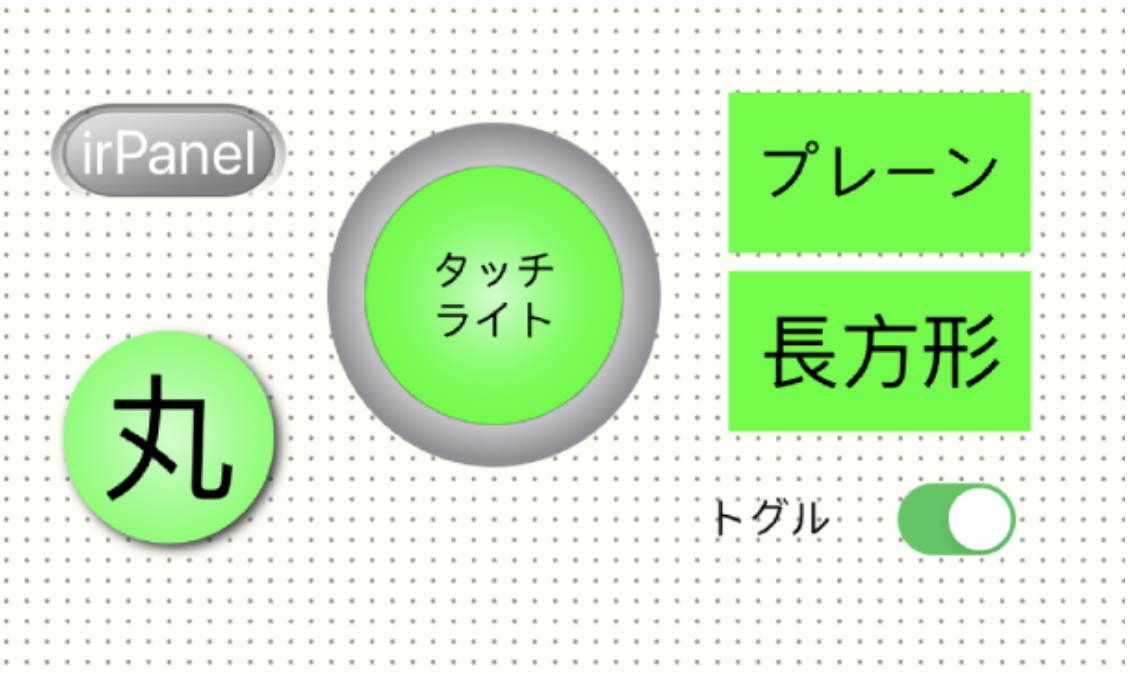
ボタンはビットデバイスをON/OFFさせたり、ビットデバイスの状態をランプとして表示することができます。
(ボタンデバイスを指定しない場合はランプとして振る舞います。)
ページ移動やプロジェクトの切り替えも行えます。

スタイル

スタイルではボタンのスタイル(形状)を指定します。

- **irPanel**
  弊社iOSアプリirPanelで使われているボタンスタイルです。
- 長方形
  長方形のスタイルで影付きです。
- 丸
  丸形のスタイルで影付きです。
- タッチライト
  センターにランプがあるタッチライト形式のスタイルで影付きです。
- プレーン
  長方形スタイルで影なしです。ボタンを透明にできます。ボタン追加時はプレーンで作成されます。
- トグル
  iOS標準のトグルボタンです。


デバイス
ボタン
MR00000
正論理
モード
モーメンタリー
遅延
デフォルト
ページリンク
プロジェクトリンク
ランプ
MR00001
正論理
インターロック
MR00002
正論理
表示
表示


irPanel
丸
タッチライト
プレーン
長方形
トグル

デバイス

デバイスではボタンを押した時にON/OFFさせるデバイスやランプ表示させるデバイスなどを設定します。

- ボタン
  ボタンを押した時にON/OFFさせるデバイスを指定します。デバイスを指定しない場合はランプとしてだけの振る舞いになります。
  - 正論理
    モードがモーメンタリーの場合ボタンを押した時に指定したデバイスをONにします。他のモードの場合はモードの説明をご覧ください。
  - 負論理
    モードがモーメンタリーの場合ボタンを押した時に指定したデバイスをOFFにします。他のモードの場合はモードの説明をご覧ください。
  - モード
    ボタンを押した時に指定したデバイスに対するON/OFFの動作を指定します。以下は正論理の場合の振る舞いで、負論理の場合はON/OFFが逆になります。
    - モーメンタリー
      ボタンを押した時に指定したデバイスをONにし、離すとOFFになります。
    - オルタネート
      ボタンを押す毎に指定したデバイスのON/OFFを反転します。
    - セット
      ボタンを押すとデバイスをONにします。
    - リセット
      ボタンを押すとデバイスをOFFにします。
  - 遅延
    ボタンを押してから反応するまでの時間を秒数で設定します。未設定の場合は プロジェクト設定のデフォルト値 が使用されます。iOSデバイスは触るとすぐ反応してしまうため、不用意に触って誤操作してしまうのを防ぐためプロジェクトのデフォルト設定で**0.3**秒遅延して反応する様にしています。
    反応を早くしたい場合は遅延時間を短めに設定してください。
  - ページリンク
    ボタンを押した時に遷移するページを選択します。選択するとページ一覧画面が表示されるので対象となるページを選択します。既に選択されているページを選択すると解除されます。Lite版で機能制限されている場合は機能しません。
  - プロジェクトリンク
    ボタンを押した時に遷移するプロジェクトを選択します。選択するとプロジェクト一覧画面が表示されるので対象となるプロジェクトを選択します。

既に選択されているプロジェクトを選択すると解除されます。Lite版で機能制限されている場合は機能しません。

- ランプ

  ボタンのランプをON/OFFさせるデバイスを指定します。

  - 正論理

    指定したデバイスがONの場合にアクティブで指定された文字や色で表示されます。OFFの場合はインアクティブの内容で表示されます。

  - 負論理

    指定したデバイスがOFFの場合にアクティブで指定された文字や色になります。ONの場合はインアクティブの内容で表示されます。

- インターロック

  指定したデバイスでボタンの有効/無効を切り替えます。

  - 正論理

    指定したデバイスがONの場合に有効になります。

  - 負論理

    指定したデバイスがOFFの場合に有効になります。

- **Visibility**

  Specify whether to show or hide the button. See Common Properties.


デバイス
ボタン
M0
正論理
モード
モーメンタリー
遅延
デフォルト
ページリンク
プロジェクトリンク
ランプ
M0
正論理
インターロック
デバイス名
表示
表示


モード
モーメンタリー
モーメンタリー
オルタネート
セット
リセット

タイトル

ボタンに表示する文字を指定します。

- 配置
  文字の位置寄せを指定します。
- 通常
  表示させる文字を指定します。(ランプデバイスを指定しない場合)
- インアクティブ
  ランプで指定したデバイスがインアクティブの場合の文字を指定します。アクティブ、インアクティブで表示が共通の場合に入力を省略できます。
- アクティブ
  ランプで指定したデバイスがアクティブの場合の文字を指定します。





フォント

文字のフォントを指定します。

- デフォルトフォント
  ONにするとデフォルトのフォントを使用し、文字サイズは幅に合わせて調整されます。

・フォント
フォントを指定します。
・自動サイズ
ONにすると文字サイズは幅に合わせて調整されます。
・文字サイズ
自動サイズをOFFにすると文字サイズを指定できます。文字サイズとボタン幅、高さを調整すると文字が折り返されたりします。



色

ボタン背景色、枠線の色を指定します。
インアクティブ時、アクティブ時の背景色、枠線色を指定します。



確認

上部のツールバーの　　アイコンで見た目を確認できます。



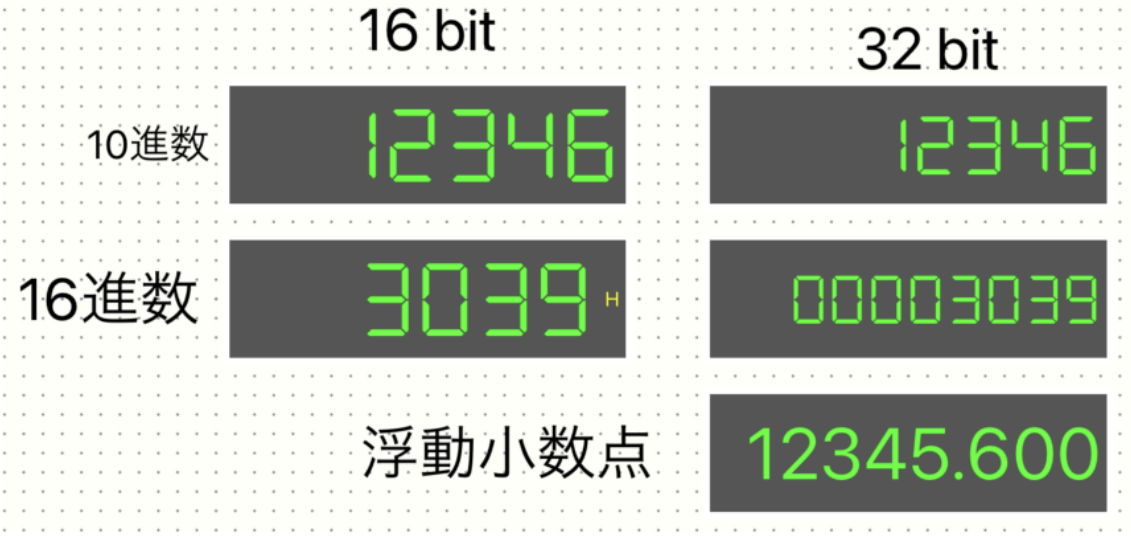
インアクティブ表示



アクティブ表示

値

値はデバイスと関連付けられた数値や文字の表示や入力ができます。



デバイス

デバイスでは数値や文字データを格納しているデバイスを設定します。

- 値
  対象となる数値や文字データを格納している先頭のデバイスを指定します。
  - 値の型
    デバイスで表現される数値や文字の型を指定します。
    - **Unsigned**
      16bit符号なし整数として扱います。
    - **Signed**
      16bit符号あり整数として扱います。
    - **HEX**
      16bit符号なし16進数として扱います。
    - **BCD**
      16bit符号なしBCD値として扱います。
    - **Unsigned(DWORD)**
      32bit符号なし整数として扱います。
    - **Signed(DWORD)**
      16bit符号あり整数として扱います。
    - **HEX(DWORD)**
      32bit符号なし16進数として扱います。

- **BCD(DWORD)**
  32bit符号なしBCD値として扱います。
- **Float**
  32bit単精度浮動小数点として扱います。
- **String**
  文字として扱います。

◦ 編集可/不可
数値や文字の編集(入力)が可能かどうかを指定します。

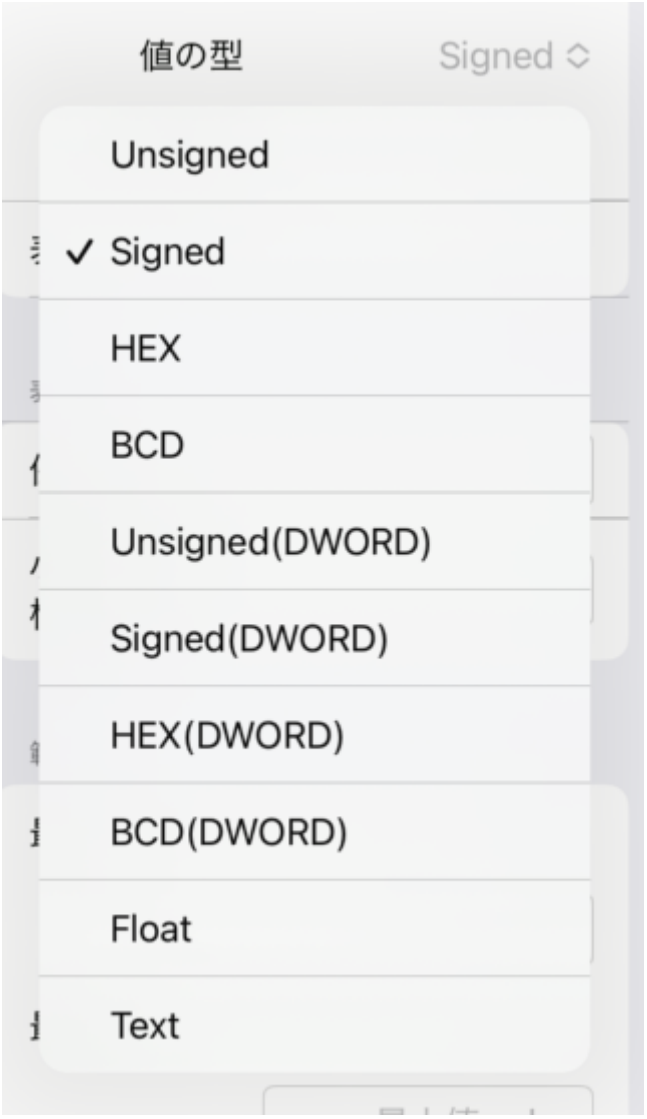
- 編集不可
  数値や文字の編集はできません。表示のみとなります。
- 編集可
  数値や文字の編集ができます。
- デバイス
  編集可/不可を切り替えるデバイスを指定します。
  - 正論理
    デバイスがONの場合に編集ができます。
  - 負論理
    デバイスがOFFの場合に編集ができます。
    ※実際は反転という表記になっていますが、バージョンアップ時に他に倣い正論理、負論理の表記に修正します。

・表示
ボタンの表示/非表示を指定します。 共通項目をご覧ください。



デバイスセクション

型の指定



編集可/不可設定



デバイスによる編集可/不可設定



表現

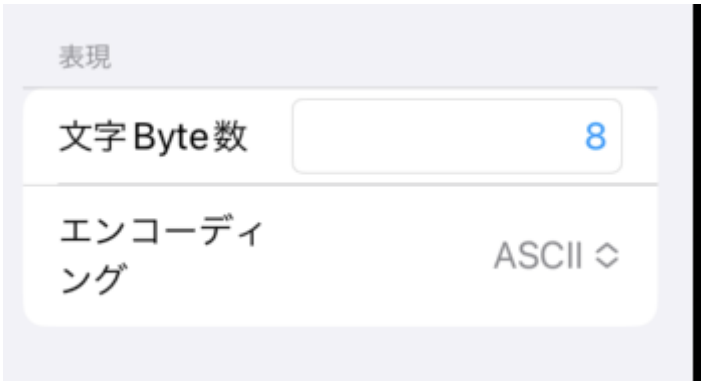
数値や文字の桁数など設定します。

- 係数、オフセット
  係数、オフセットを用いてデバイス値から表示する値にスケール変換ができます。
  一次関数の傾きと切片に相当します。
  係数をデバイスの値に掛けた後オフセットを加算した数値を表示します。実行時に入力された数値はオフセットを差し引き係数の逆数を掛けた結果をデバイスに書き込みます。
- 小数点以下桁数
  小数点以下の桁数を指定します。係数が未設定の場合、桁数に合わせた係数が自動的に適用されます。
- 秘密
  入力した値が"."で表示され分からなくします。パスワード入力する際に使用します。
- 文字**Byte**数
  文字の長さをByteで制限します。
- エンコーディング
  文字のエンコーディングを指定します。

数値の場合



文字の場合



バーコード

数値の編集が可能の場合にカメラを利用しバーコードを読み込み入力させる事ができます。

- バーコード使用
  バーコードの読み込みができる様になります。
- パターン
  設定したパターンと一致すると入力ができます。パターンは 正規表現 の形式で入力します。
  例えば \d は数値を表し + は1文字以上を表します。
  読み取ったコードが SN12345A の場合に1文字以上の数値に該当する12345が入力対象となります。

実行時

- 実行時に値部品の右側にQRコードマークが表示されます。



  マークをタッチするとカメラが起動します。
- カメラでコードを読み取りパターンと一致すると右上に一致した文字がリストアップされます。



コードは SN12345A ですがパターンに一致した 12345 がリストアップされています。

・リストアップされた文字が正しければ、その文字をタッチすると反映されます。





範囲

数値の場合に実行時に入力値の範囲を制限することができます。PLCから読み出した数値に対しては影響せずそのまま表示されます。

- 最小値
  直値を選択した場合、最小値を入力します。
  デバイスを選択した場合に指定したデバイスの値を最小値として扱います。
  実行時に指定された最小値より小さい数値を入力した場合は最小値に変更されます。
- 最大値
  直値を選択した場合、最大値を入力します。
  デバイスを選択した場合に指定したデバイスの値を最大値として扱います。
  実行時にこの値より大きい数値を入力した場合は最大値に変更されます。



直値での範囲設定



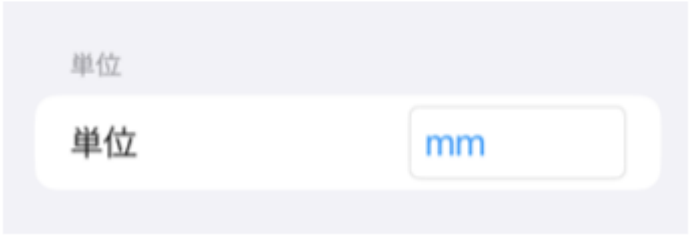
デバイスでの範囲設定

単位

数値の場合単位を付加することができます。

・単位
　単位となる記号や文字を入力します。





フォント

数値や文字、単位のフォントを指定できます。設定については ラベル や ボタン の項目をご覧ください。
ラベルやボタンと異なり表示が領域内に入り切らない場合は末尾が省略表示になります。





色

数値や文字の色、単位の色、背景色、枠線色を指定します。
インアクティブ時、アクティブ時の背景色、枠線色を指定します。



確認

上部のツールバーの　　アイコンを押すとプレビューに使用する値を変更できます。

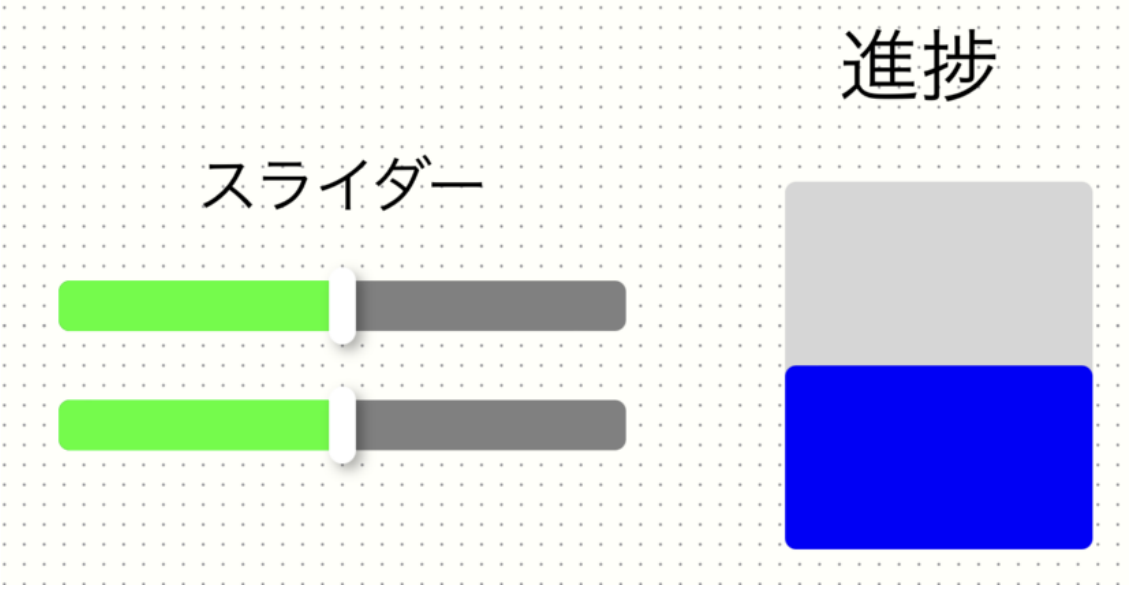
- 数値
  プレビューで数値に使用する値を設定します。デバイスの値として扱われ、この値に桁数設定など適用されて表示されます。
- テキスト
  プレビューで文字に使用する値を設定します。


123
プレビュー値
数値
12345
テキスト
ABCDEFGHIJK


1234.5
ABCDEFGHIJK

スライダー/進捗

スライダーはスライドさせて数値を変化させその値をデバイスに書き込みます。
プログレスはスライダーの数値を変更するノブがなくなったもので、数値をレベルで表現します。



デバイス

デバイスでは数値を格納しているデバイスを設定します。

- 値
  対象となる数値を格納している先頭のデバイスを指定します。
  - 遅延
    スライダーを押してから反応するまでの時間を秒数で設定します。未設定の場合は プロジェクト設定のデフォルト値 が使用されます。iOSデバイスは触るとすぐ反応してしまうため、不用意に触って誤操作してしまうのを防ぐためプロジェクトのデフォルト設定で**0.3**秒遅延して反応する様にしています。
  - 値の型
    デバイスで表現される数値や文字の型を指定します。
    - **Unsigned**
      16bit符号なし整数として扱います。
    - **Signed**
      16bit符号あり整数として扱います。
    - **HEX**
      16bit符号なし整数を16進数として扱います。

- **BCD**
  16bit符号なし整数を16進数として扱います。
- **Unsigned(DWORD)**
  32bit符号なし整数として扱います。
- **Signed(DWORD)**
  16bit符号あり整数として扱います。
- **HEX(DWORD)**
  32bit符号なし整数を16進数として扱います。
- **BCD(DWORD)**
  32bit符号なし整数を16進数として扱います。
- **Float**
  単精度浮動小数点(32bit)として扱います。

◦ 編集可/不可
数値の編集がが可能かどうかを指定します。編集可能だとノブが表示されスライダーとして振る舞います。編集不可の時はノブが非表示になり進捗表示になります。

- 編集不可
  スライダーで数値を変更できます。
- 編集可
  数値の変更ができず進捗表示のみとなります。
- デバイス
  編集可/不可を切り替えるデバイスを指定します。
  - 正論理
    デバイスがONの場合に編集ができます。
  - 負論理
    デバイスがOFFの場合に編集ができます。
    ※実際は反転という表記になっていますが、バージョンアップ時に他に倣い正論理、負論理の表記に修正します。

・連続
OFFの場合はスライダー操作が終わった時に数値をデバイスに書き込みます。
ONにするとスライダーを操作している途中でも連続的に数値をデバイスに書き込みます。通信回数が増えますので、反応が遅くなる場合はOFFにして使用してください。

・**Visibility**
Specify whether to show or hide the button. See Common Properties.



表現

数値の桁数など設定します。

・係数、オフセット
係数、オフセットを用いてデバイス値から表示する値にスケール変換ができます。一次関数の傾きと切片に相当します。
係数をデバイスの値に掛けた後オフセットを加算した数値を表示します。実行時に入力された数値はオフセットを差し引き係数の逆数を掛けた結果をデバイスに書き込みます。
・小数点以下桁数
小数点以下の桁数を指定します。係数が未設定の場合、桁数に合わせた係数が自動的に適用されます。



範囲

実行時に入力値の範囲を制限することができます。PLCから読み出した数値が範囲外の場合はスライダーや進捗の最小値、最大値として表示されます。

- 最小値
  直値を選択した場合、最小値を入力します。
  デバイスを選択した場合に指定したデバイスの値を最小値として扱います。
  実行時に指定された最小値より小さい数値を入力した場合は最小値に変更されます。
- 最大値
  直値を選択した場合、最大値を入力します。
  デバイスを選択した場合に指定したデバイスの値を最大値として扱います。
  実行時にこの値より大きい数値を入力した場合は最大値に変更されます。



直値での範囲設定

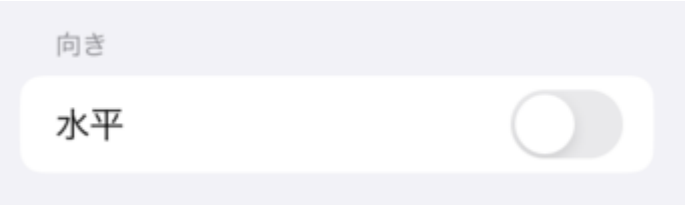


デバイスでの範囲設定

向き

スライダーや進捗の数値の増減する方向を設定します。

- 水平
  水平方向に増減します。
- 垂直
  垂直方向に増減します。



色

ノブや値、スライダー色を指定します。

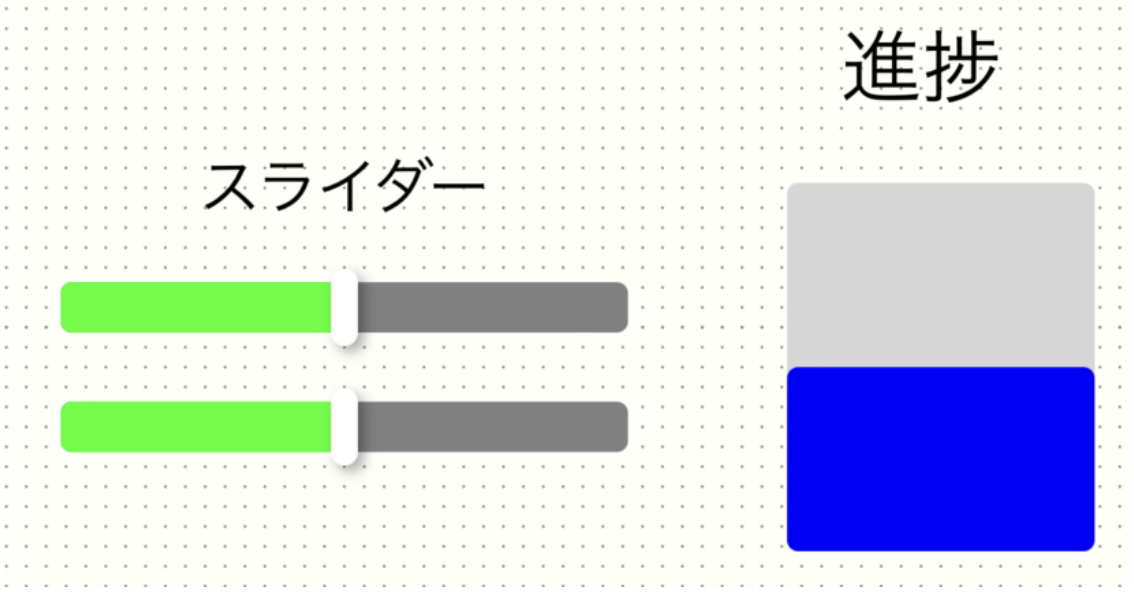


確認

上部のツールバーの　　アイコンを押すとプレビューに使用する値を変更できます。

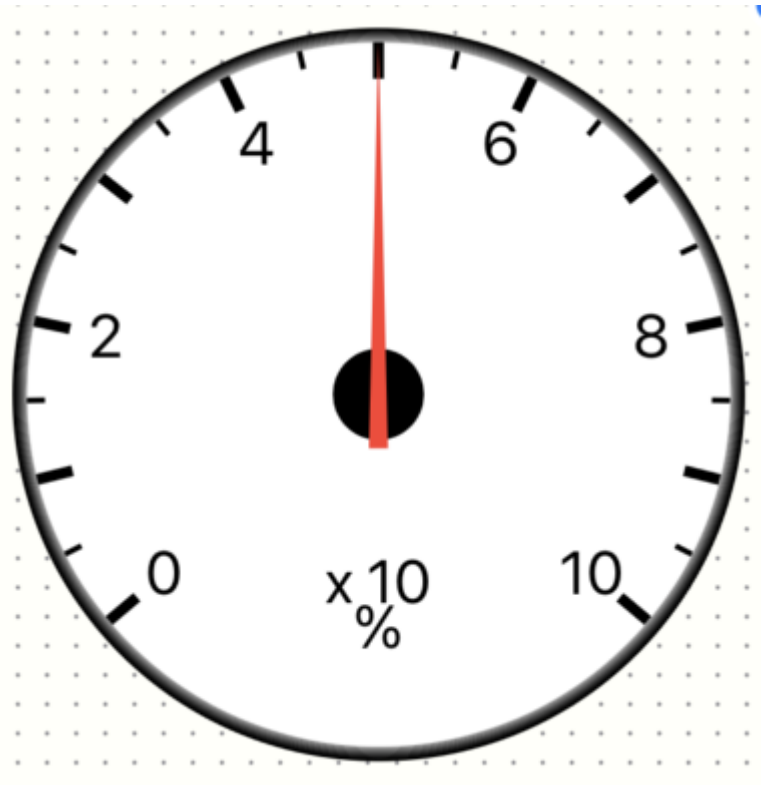
- 数値
  プレビューで数値に使用する値を設定します。デバイスの値として扱われ、この値に桁数設定など適用されて表示されます。


123
プレビュー値
数値
50


進捗
スライダー

メーター

数値をメーターで表現します。



デバイス

デバイスでは数値を格納しているデバイスを設定します。

- 値
  対象となる数値を格納している先頭のデバイスを指定します。
- 値の型
  デバイスで表現される数値や文字の型を指定します。
  - **Unsigned**
    16bit符号なし整数として扱います。
  - **Signed**
    16bit符号あり整数として扱います。
  - **HEX**
    16bit符号なし整数を16進数として扱います。
  - **BCD**
    16bit符号なし整数を16進数として扱います。

- **Unsigned(DWORD)**
  32bit符号なし整数として扱います。
- **Signed(DWORD)**
  16bit符号あり整数として扱います。
- **HEX(DWORD)**
  32bit符号なし整数を16進数として扱います。
- **BCD(DWORD)**
  32bit符号なし整数を16進数として扱います。
- **Float**
  単精度浮動小数点(32bit)として扱います。

・**Visibility**
Specify whether to show or hide the button. See Common Properties.



表現

数値の桁数など設定します。

・係数、オフセット
係数、オフセットを用いてデバイス値から表示する値にスケール変換ができます。
一次関数の傾きと切片に相当します。
係数をデバイスの値に掛けた後オフセットを加算した数値を表示します。実行時に入力された数値はオフセットを差し引き係数の逆数を掛けた結果をデバイスに書き込みます。

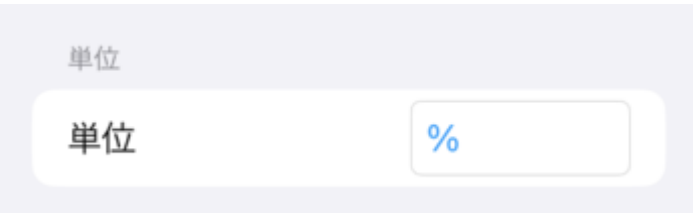
・小数点以下桁数
小数点以下の桁数を指定します。係数が未設定の場合、桁数に合わせた係数が自動的に適用されます。



単位

数値の場合単位を付加することができます。

・単位
単位となる記号や文字を入力します。



範囲

実行時に入力値の範囲を制限することができます。PLCから読み出した数値が範囲外の場合はスライダーや進捗の最小値、最大値として表示されます。

・最小値
直値を選択した場合、最小値を入力します。
デバイスを選択した場合に指定したデバイスの値を最小値として扱います。
実行時に指定された最小値より小さい数値を入力した場合は最小値に変更されます。
・最大値
直値を選択した場合、最大値を入力します。
デバイスを選択した場合に指定したデバイスの値を最大値として扱います。
実行時にこの値より大きい数値を入力した場合は最大値に変更されます。



直値での範囲設定



デバイスでの範囲設定

スケール

スケールではメーターの目盛り関する設定を行います。

- メジャー
  目盛り表示の最小値から最大値までの分割数を指定します。
  直値を選択した場合は入力した値が用いられ、デバイスを指定した場合はデバイスの値が用いられます。
- マイナー
  直値を選択した場合は入力した値が用いられ、デバイスを指定した場合はデバイスの値が用いられます。 メジャーで分割された一目盛り内の分割数を指定します。
- 文字
  直値を選択した場合は入力した値が用いられ、デバイスを指定した場合はデバイスの値が用いられます。 メジャーで分割された目盛りの何個置きに数値のラベルを描画するかを指定します。



直値での設定



デバイスでの設定

ゾーン

ゾーンを使用すると、メーター上に色付きの範囲を定義して、値の状態（正常・注意・危険など）を視覚的に表示できます。

- ゾーンの追加
  「+」ボタンでゾーンを追加します。各ゾーンには色、開始値（From）、終了値（To）を設定します。

・プリセット（3ゾーン）
メーターの範囲を3等分した緑（低）・ 黄（中）・ 赤（高）のゾーンを一括で追加します。
・色
カラーピッカーでゾーンの色を設定します。任意の色を選択できます。
・**From / To**
各ゾーンの値の範囲を設定します。ゾーンはメーター上に色付き円弧として表示されます。
・削除
ゾーンの行を左にスワイプして削除します。



フォント

数値や単位のフォント(共通)を指定できます。設定については ラベル や ボタン の項目をご覧ください。

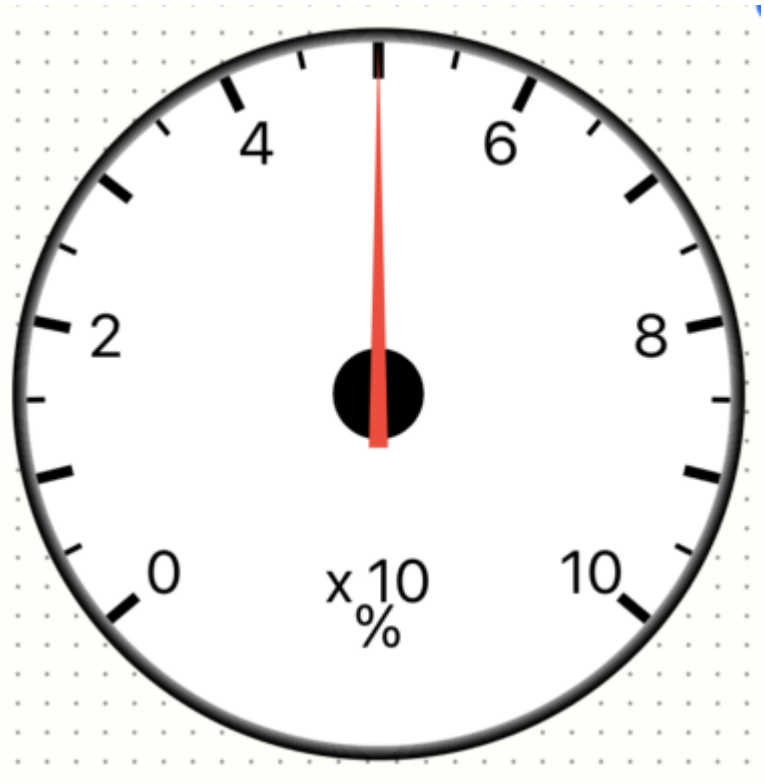


確認

上部のツールバーの　　アイコンを押すとプレビューに使用する値を変更できます。

・数値
プレビューで数値に使用する値を設定します。デバイスの値として扱われ、この値に桁数設定など適用されて表示されます。




0
2
4
6
8
10
x10
%

Webページ

Webページを表示できます。

URL

表示するページのURLを指定します。

USER AGENT

ページにアクセスする際にUSER AGENTを指定したい場合に設定します。

コントロール

・移動ボタン
ONにするとページの移動ボタンを表示します。

使用例

・Webカメラで現場状況をモニター



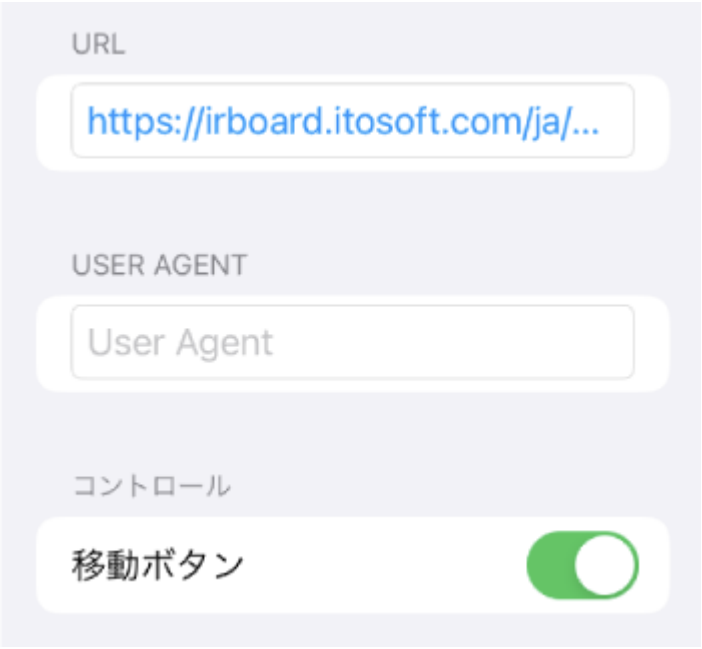
・ドキュメントの表示
使い方や手順などの説明を表示できます。



・データ表示
Ambient などのデータを蓄積できるサイトを利用してIoTデータの確認ができます。



・その他Webアプリケーションや Unitアプリなどを実行したり様々な応用ができます。


URL
https://irboard.itosoft.com/ja/...
USER AGENT
User Agent
コントロール
移動ボタン

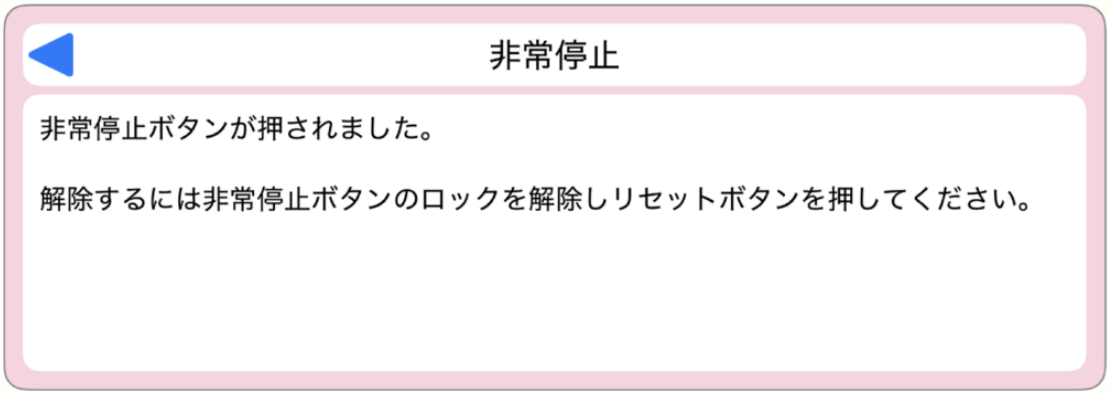
メッセージ

アラームや条件などのメッセージを表示できます。 また、詳細情報を表示することもできます。

発生しているアラームの表示



アラームを選択すると詳細を表示。



メッセージ部品設定

メッセージ

表示するメッセージの登録や選択を行います。

- メッセージリストの編集
  表示する メッセージリストの編集 をします。タッチするとメッセージリストの編集画面が表示されます。

・メッセージリスト
メッセージリストは複数作成することができます。メッセージ部品で表示対象となるメッセージリストを選択します。



メッセージリストの編集

メッセージリストの編集をタッチすると編集画面が開きます。左側にメッセージリストの一覧が表示され、右側で選択したメッセージリストの編集を行います。

・"+" ボタン
メッセージリストを追加します。
・タイトル
メッセージリストの内容が分かる様にタイトルを入力します。
・メッセージ追加、削除
メッセージリストで扱う メッセージの追加、削除 を行います。
・メッセージ
メッセージの一覧が表示されます。タッチすると メッセージの内容を編集 することができます。



メッセージ追加、削除

メッセージを表示するトリガーとなるデバイス(メッセージデバイス)の追加、削除を行います。開始デバイスとデバイスの増加数、繰り返し数(個数)を指定することで複数のメッセージデバイスの追加、削除ができます。

・開始ビットデバイス
メッセージを表示するトリガーとなる最初のビットデバイスを入力します。
・増加幅
デバイスの増加数を入力します。開始デバイスがM0で増加幅が1の場合は、M0,M1,M2,...が対象となります。増加幅が2の場合はM0,M2,M4,...と一つ跳びのデバイスが対象となります。
・繰り返し数
開始デバイスから増加幅で何回繰り返す(個数)かを入力します。
・追加
開始ビットデバイス、増加幅、繰り返し数で指定されたメッセージデバイスが追加されます。既に追加されているメッセージデバイスがある場合に、新たに追加する場合は未登録のもののみ追加されます。
・削除
開始ビットデバイス、増加幅、繰り返し数で指定されたメッセージデバイスが削除されます。
プレビュー
追加、削除の対象となるメッセージデバイスが表示されます。



メッセージ編集

・ビットデバイス名
トリガーとなるメッセージデバイスを表示します。このデバイスがONの時にオン状態の表示をします。
・メッセージ
メッセージ部品に表示される文字を入力します。
・詳細
メッセージに対する詳細を表示する場合に入力します。
例) アラームの対処方法など



行

メッセージを行にどの様に表示するかを指定します。

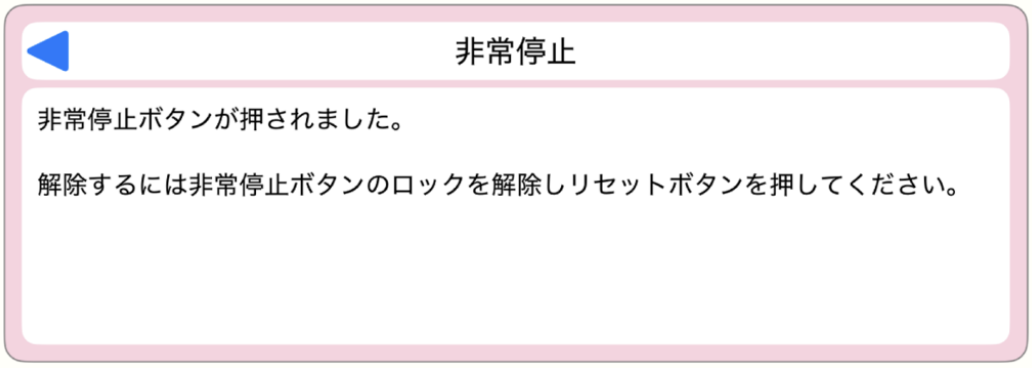
・オン状態のみ
オンにするとメッセージデバイスがオンになっているメッセージのみ表示されます。



オフの場合は全てのメッセージデバイスが表示されます。



・詳細表示
詳細を表示するかしないかを指定します。
オンにするとメッセージの右側に"?"マークが表示され、タッチすると詳細が表示されます。
オフの場合はマークが表示されずタッチしても何も起こりません。



・高さ
行の高さを指定します。



フォント

文字のフォントを指定します。

・デフォルトフォント
ONにするとデフォルトのフォントを使用し、文字サイズは幅に合わせて調整されます。
・フォント
フォントを指定します。
・自動サイズ
ONにすると文字サイズは幅に合わせて調整されます。
・文字サイズ
自動サイズをOFFにすると文字サイズを指定できます。文字サイズとラベル幅、高さを調整すると文字が折り返されたりします。



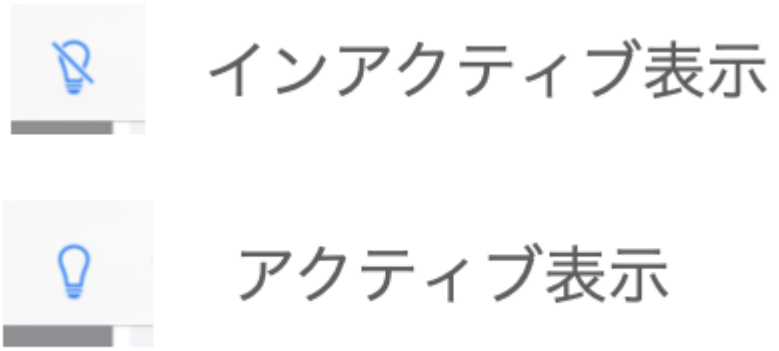
色

ラベル背景色、枠線の色を指定します。
メッセージデバイスがオン状態、オフ状態の時の文字色、背景色を指定します。



確認

上部のツールバーの　　アイコンで見た目を確認できます。



マトリックス

マトリックスはテンプレートから複数のアイテムを自動生成し、グリッド状に配置する部品です。 同じパターンのアイテムを大量に作成する場合に便利です。 テンプレートを編集すると全セルに反映され、デバイスオフセットを設定することでセルごとにデバイスアドレスを自動的にインクリメントできます。

グリッド設定

マトリックスの基本的なグリッド構成を設定します。

・行数
　グリッドの行数を指定します。
・列数
　グリッドの列数を指定します。

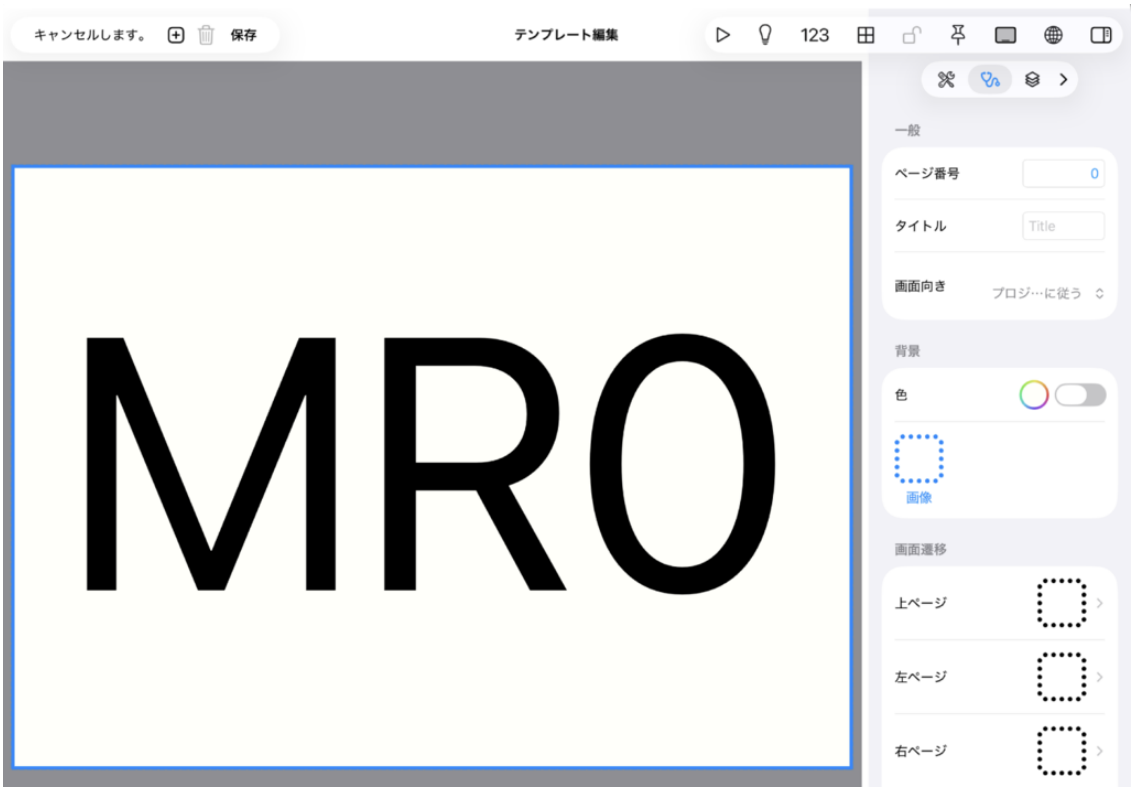


テンプレート編集

テンプレートはマトリックスの各セルの内容を定義します。 テンプレートを変更すると、全てのセルに反映されます。

・テンプレートをデザイン
　「テンプレートをデザイン」ボタンを押すと、テンプレート編集画面が開きます。この画面では1セル分の内容を編集できます。 ラベルやボタンなどの部品をテンプレート内に配置し、デバイスやタイトルを設定します。
・テンプレートの保存
　編集が完了したら「保存」ボタンを押してテンプレートを適用します。 保存するとマトリックスの全セルが再生成されます。

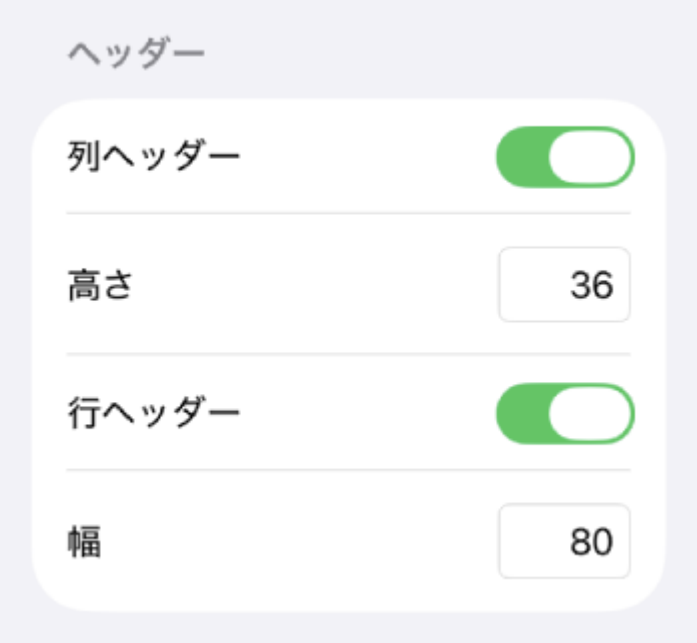

ウィザード
テンプレートをデザイン
列ヘッダーテンプレート
行ヘッダーテンプレート
デバイスオフセット
ウィザード


キャンセルします。
保存
テンプレート編集
123
MRO
一般
ページ番号
0
タイトル
Title
画面向き
プロジ…に従う
背景
色
画像
画面遷移
上ページ
左ページ
右ページ

ヘッダー

行ヘッダーと列ヘッダーを表示できます。ヘッダーにもテンプレートとデバイスオフセットを設定できます。

- 列ヘッダー
  グリッドの上部に列ヘッダーを表示します。
  - 高さ
    列ヘッダーの高さを指定します。
- 行ヘッダー
  グリッドの左側に行ヘッダーを表示します。
  - 幅
    行ヘッダーの幅を指定します。



デバイスオフセット

デバイスオフセットを設定すると、セルの位置（行・列）に応じてデバイスアドレスやタイトルテキストを自動的にインクリメントできます。「デバイスオフセット」ボタンを押すと、ルールベースのデバイスオフセット設定画面（ウィザード）が開きます。

ルールの構成

各ルールは以下の4つの条件でターゲットを指定します。

- アイテム
  ルールを適用するテンプレートアイテムを選択します。「全アイテム」を選ぶとテ

ンプレート内の全てのアイテムが対象になります。特定のアイテムだけにオフセットを設定したい場合は個別に選択します。

- プロパティ
  ルールを適用するデバイスプロパティを選択します。「全プロパティ」を選ぶと対象アイテムの全デバイスプロパティが対象になります。 プロパティには以下のものがあります (部品の種類によって異なります) :
  - device — デバイス名
  - visible — 表示/非表示デバイス
  - label — ラベルデバイス
  - lamp — ランプデバイス
  - switch — スイッチデバイス
  - interlock — インターロックデバイス
  - editable — 編集可能デバイス
  - max_value — 最大値デバイス
  - min_value — 最小値デバイス
  - Title — タイトルテキスト
  - Alt Title — 代替タイトルテキスト
- 行
  ルールを適用する行を指定します。「All」を選ぶと全行が対象になります。特定の行番号を指定すると、その行のみにルールが適用されます。
- 列
  ルールを適用する列を指定します。「All」を選ぶと全列が対象になります。特定の列番号を指定すると、その列のみにルールが適用されます。

オフセット値

- 通常モード (行または列が「All」の場合)
  行オフセットと列オフセットの2つの値を設定します。各セルのオフセットは以下の式で計算されます:
  `オフセット = 行番号 × 行オフセット + 列番号 × 列オフセット`
- セル指定モード (行と列の両方を特定の番号で指定した場合)
  オフセットは1つの値のみ設定します。指定した値がそのセルに直接適用されます。

オフセットの計算例

テンプレートにデバイス `D100` のアイテムがあり、3行4列のマトリックスの場合:

例**1:** 行オフセット=10, 列オフセット=1

	列**0**	列**1**	列**2**	列**3**
行**0**	D100	D101	D102	D103

	列**0**	列**1**	列**2**	列**3**
行**1**	D110	D111	D112	D113
行**2**	D120	D121	D122	D123

例**2:** 行オフセット=20, 列オフセット=2

	列**0**	列**1**	列**2**	列**3**
行**0**	D100	D102	D104	D106
行**1**	D120	D122	D124	D126
行**2**	D140	D142	D144	D146

ルールの優先順位

複数のルールがある場合、より具体的なルールが優先されます。具体性のスコアは以下の通りです:

- セル指定（行+列を指定）: 最も優先度が高い
- 行指定のみ: 高い
- 列指定のみ: 中程度
- 全セル対象: 低い
- 特定アイテム指定は「全アイテム」より優先
- 特定プロパティ指定は「全プロパティ」より優先

同じスコアのルールが複数ある場合は、ルールリストの上にあるものが優先されます。ルールはドラッグで並べ替えできます。

タイトルテキストのオフセット

デバイスアドレスだけでなく、タイトルや代替タイトルに含まれる数字やアルファベットも自動的にインクリメントできます。 テキスト中の数値は以下の形式を自動検出します:

- 半角数字 — 例: "CH1" → "CH2" → "CH3"
- 全角数字 — 例: "１号機" → "２号機" → "３号機"
- 漢数字 — 例: "第一" → "第二" → "第三"
- 半角英字 — 例: "Line A" → "Line B" → "Line C"
- 全角英字 — 例: "ラインＡ" → "ラインＢ" → "ラインＣ"

ターゲットの切り替え

ヘッダーが有効な場合、ウィザード上部のセグメントでルールの対象を切り替えできます:

- **Data** — データセル（メインのグリッド領域）
- **Column Header** — 列ヘッダー
- **Row Header** — 行ヘッダー

それぞれのターゲットに対して独立したルールセットを設定できます。

プレビュー

ウィザード画面の左下にプレビューグリッドが表示されます。 アイテムとプロパティを選択すると、各セルに適用されるオフセット値をリアルタイムで確認できます。 ルールを編集すると即座にプレビューに反映されます。



条件設定

LABEL [MR0]

device	R+10, C+1
lamp	R+10, C+1
title	R+10, C+1

ルール



プレビュー

	CH1	CH2	CH3	CH4
ROW1	MR0	MR1	MR2	MR3
ROW11	MR10	MR11	MR12	MR13
ROW21	MR20	MR21	MR22	MR23
ROW31	MR30	MR31	MR32	MR33
ROW41	MR40	MR41	MR42	MR43

表示

罫線

セル間に罫線を追加できます。

- 罫線を追加
  ONにするとセル間に罫線が表示されます。
- 罫線の太さ
  罫線の太さを指定します。
- 罫線のスタイル
  罫線のスタイルを選択します。
  - 実線
  - 破線



色

マトリックスの各部分の色を設定できます。

- 背景色
  ヘッダーも含めたマトリックス全体の背景色を指定します。
- セル背景色
  データセルの背景色を指定します。
- 交互パターン
  データセルの行または列の交互色を設定できます。行単位か列単位かを選択します。
- 交互色
  偶数行（または偶数列）の背景色を指定します。
- 罫線の色
  罫線の色を指定します。
- 列ヘッダー背景色
  列ヘッダーの背景色を指定します。
- 行ヘッダー背景色
  行ヘッダーの背景色を指定します。
- コーナー背景色
  行ヘッダーと列ヘッダーの交差部分（左上コーナー）の背景色を指定します。
- セル個別色
  特定のセルの背景色を個別に上書きできます。「セル個別色」ボタンを押すとセル色設定ウィザードが開きます。 行番号· 列番号付きのグリッドでセルを選択し、カラーピッカーで色を指定します。「色をクリア」選択セルの色を解除、「全ての色をクリア」で全セルの色を一括解除できます。

  色の優先順位: セル色 > 交互色 > セル背景色

色

背景色

セル背景色

交互パターン 行交互

交互色

罫線の色

列ヘッダー 背景色

行ヘッダー 背景色

コーナー背景色

セル個別色 (3)


キャンセルします。
セル個別色
完了
0
1
2
3
0
1
2
3
4
Cell (2, 1)
色
色をクリア
全ての色をクリア

AIインポート

AIインポートは制御パネルレイアウトの画面イメージを取り込み、AIが自動的に部品を検出してページに配置する機能です。 既存の制御パネルの画面キャプチャや写真から、ラベル、ボタン、値フィールド、メーターなどの部品を自動認識し、レイアウトを再現します。

はじめに

AIインポート機能を使用するには、Anthropic APIキーが必要です。 APIキーはAIインポート画面の「ヘルプ&設定」から登録できます。

- **APIキーの取得**
  Anthropic Consoleにアクセスし、アカウントを作成してAPIキーを取得します。
- **APIキーの登録**
  AIインポート画面上部の「ヘルプ&設定」ボタンをタップし、「Anthropic API Key」にAPIキーを入力します。

練習用モード**:** AIインポートの動作をお試しできる練習用モードが利用できます。 サンプルデータを使って部品検出の結果を確認できます。

使い方

1. AIインポートを開く

編集画面のツールバーにあるアイテム追加ボタン（「+」）をタップし、メニューから「AIインポート」を選択します。

2. 画像を選択

AIインポート画面で以下のいずれかの方法で画面イメージを取り込みます:

- **画像を選択**
  「画像を選択」ボタンをタップして写真ライブラリから画像を選択します。

- 練習用
  練習用セクションからサンプルをタイトルで選択してAIインポートの動作を試せます:
  - コントロールパネル (31アイテム) — 通常の部品（ラベル、ボタン、値フィールド、メーター）
  - マトリックス表 (9アイテム) — マトリックス（表形式のデータ）

3. 検出モードを選択

画像の内容に応じて検出モードを選択します:

- マトリックスなし
  通常の部品（ラベル、ボタン、値フィールド、メーターなど）を検出します。 表形式のレイアウトが含まれない画面に適しています。
- マトリックスのみ
  マトリックス（表形式のレイアウト）のみを検出します。 表形式のレイアウトが含まれる画面に適しています。

4. 解析実行

「解析」ボタンをタップすると、AIが画像を分析して部品を検出します。 検出が完了すると、検出された部品のリストが表示されます。

5. 検出結果の確認と編集

検出された部品のリストで、各部品の情報を確認・編集できます:

- 部品タイプ
  部品の種類（ラベル/ボタン/値/メーター/マトリックス）をPickerで変更できます。
- タイトル
  部品に表示されるテキストを編集できます。
- 単位（値/メーターのみ）
  AIが自動検出した単位をタップでON/OFF切り替えできます。 例:
  "°C"、"kg"、"個"、"%"など
- 編集可（値のみ）
  鉛筆アイコンをタップして、値フィールドの編集可/不可を切り替えます。
- 背景色とテキスト色
  色丸で部品の背景色とテキスト色を確認できます。
- 罫線（マトリックスのみ）
  AIが判定した罫線の有無を確認・変更できます。
- ヘッダー背景色（マトリックスのみ）
  "H"ラベル付き色丸でヘッダー背景色を確認できます。

6. ページに配置

「選択を配置」ボタンをタップすると、検出された部品が編集中のページに配置されます。 部品は画像内での位置とサイズに基づいて配置されます。

7. 再解析

検出結果が期待通りでない場合、「再解析」ボタンで再解析できます。

検出の詳細

検出可能な部品

- ラベル — テキスト表示
- ボタン — ボタン
- 値 — 数値表示・入力フィールド
- メーター — メーター（ゲージ）
- マトリックス — データ表（行・列のグリッド）

自動検出される情報

- 位置とサイズ
  画像内での部品の位置とサイズをパーセンテージで検出します。
- テキスト内容
  日本語・英語のテキストを認識します。
- 背景色とテキスト色
  部品の色を色名（Red / Blue / Green など）で検出します。
- 数値の単位
  値フィールドやメーターのタイトルから単位を自動抽出します。 例: "85.23 °C" → 単位:"°C"、小数点以下:2桁
- 小数点桁数
  表示されている数値から小数点以下の桁数を自動判定します。
- マトリックスの構造
  行数・列数、ヘッダーの有無、罫線の有無を検出します。

検出精度を高めるコツ

- 画質
  鮮明でピントの合った画像を使用します。
- 画像サイズ
  大きめの画像の方が精度が高くなります。

・余白
不要な部分を含まず、必要な部品のみが写るようにトリミングします。
・検出モード
画像の内容に応じて適切な検出モード（マトリックスなし / マトリックスのみ）を選択します。

キャッシュ

解析結果はデバイスにキャッシュされます。 同じ画像と指示（追加の指示）の組み合わせで解析する場合、APIを呼び出さずにキャッシュから即座に結果を読み込みます。

・キャッシュの確認
キャッシュからの結果の場合、検出結果のヘッダーに「(キャッシュ)」と表示されます。
・新しい結果の取得
キャッシュを使用せずに新しい解析結果を取得したい場合は、「再解析」ボタンをタップします。

キャッシュにより、同じ画像を繰り返し解析する際のAPI呼び出しコストと時間を節約できます。

ヒント

・デバイスの割り当て
AIインポートで配置された部品には自動的にデバイス名は割り当てられません。 配置後にインスペクターで各部品のデバイス名を設定してください。
・微調整
配置された部品の位置やサイズは、配置後に手動で調整できます。
・部品の削除
不要な部品は選択して削除できます。 配置した全アイテムを一括削除したい場合は、Undoを使用してください。
・色のカスタマイズ
配置後にインスペクターで色を変更できます。
・マトリックスの編集
配置されたマトリックスは、 マトリックスドキュメント を参照してテンプレートやデバイスオフセットを設定してください。
・コストについて
AIインポートはClaude Haiku 4.5を使用しており、低コストです。

プライバシー

AIインポート機能を使用する際、選択した画像はAnthropicのClaude APIに送信されて処理されます。 画像に機密情報が含まれる場合はご注意ください。

練習用モードでは画像がサーバーに送信されることはありません。 サンプルデータはアプリ内に保存されており、オフラインで動作します。

プロジェクト実行

実行方法

- 編集画面 のツールバーから  プレイ
ボタンを押すと実行画面に移行します。
- 初めての場合は実行を停止する方法が表示されます。次回も表示するかどうか聞いてきますので、どちらかを選びます。

終了方法

- プロジェクトの実行を停止したい場合は、部品が何もない箇所を **5**回 タップします。
- 

上部にナビゲーションバーが表示されます。その中から 終了 を選択します。
- 確認画面で終了を選択すると終了します。

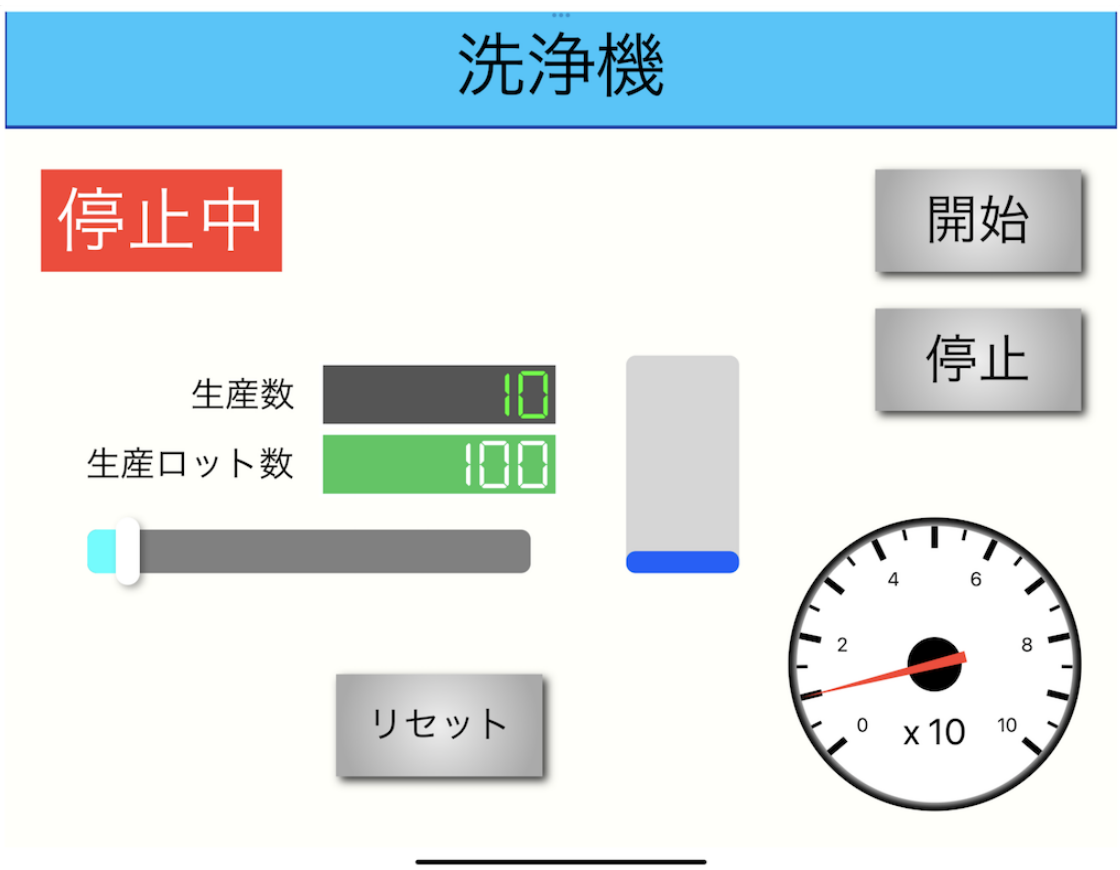
ナビゲーションバー

- 

ナビゲーションバーを隠します。
- 
言語選択ができます。
一度押して から、 もう一度長押し してメニューを表示させてください。
最初に一度押すのはナビゲーションバーが消えてメニューも消えてしまうのを防ぐためです。
ローカライズ(多言語化)を参照
- 

プロジェクトの 実行を終了 します。

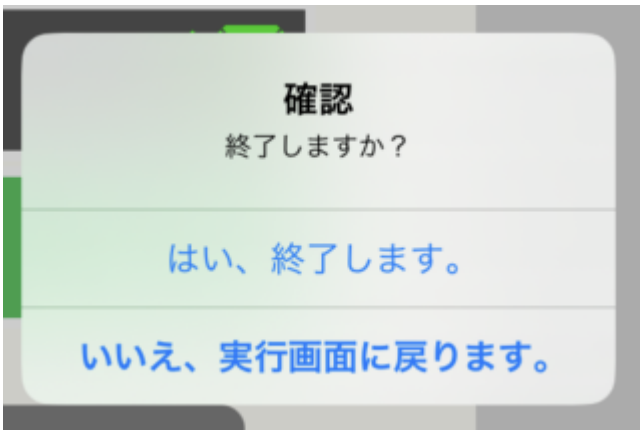
実行画面



実行を停止する方法



終了の確認



データログ

データログ機能は、PLCのデバイス値を定期的または条件に応じてCSVファイルに記録する機能です。
複数の設定を作成し、同時に記録することができます。

設定

プロジェクト設定の「データログ」セクションから設定画面を開きます。

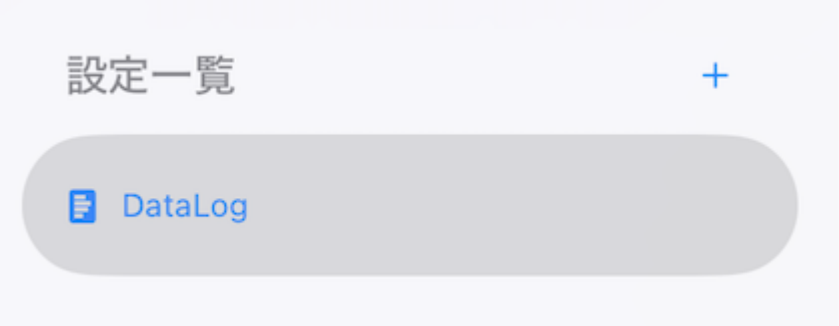
設定一覧

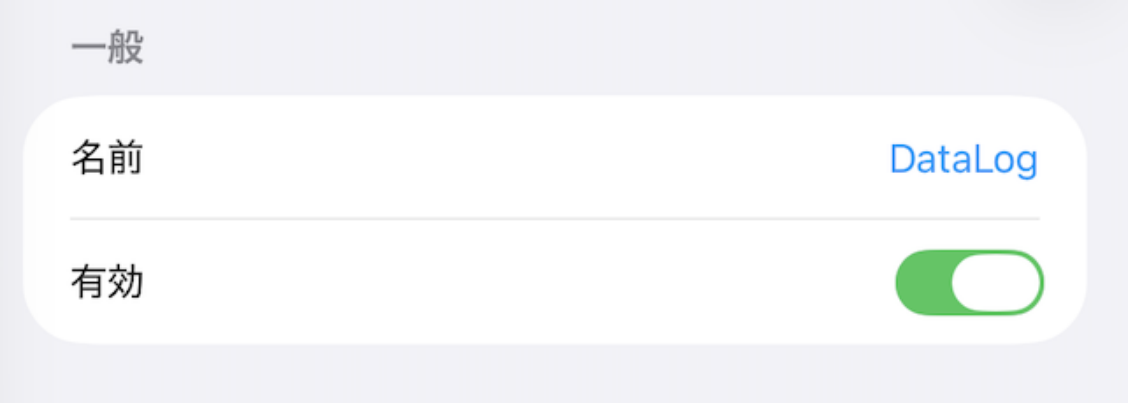
- 左側のサイドバーに設定一覧が表示されます。
- + ボタンで新しい設定を追加できます。
- 設定名をダブルタップすると名前を変更できます。
- スワイプまたはコンテキストメニューで削除できます。

全般

- 名前
  設定の名前を指定します。
- 有効
  この設定を有効にするかどうかを指定します。無効な設定はプロジェクト実行時にロギングが開始されません。







記録タイプ

時系列

デバイス値を時系列で記録します。各行にタイムスタンプと全デバイスの値が記録されます。

timestamp	速度	圧力	温度
2026-05-16 10:00:00.000	1500	80	25
2026-05-16 10:00:01.000	1520	81	26

テーブル

トリガーごとに1ファイルを作成し、テーブル形式でデバイス値を記録します。
設定した行数に応じて、各デバイスのアドレスをオフセットしながら展開します。

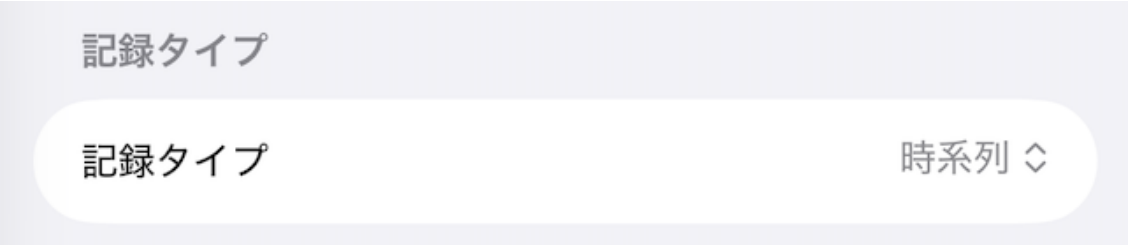
- 行数
  テーブルの行数を指定します。

例えば、デバイスに `D0000`（速度）と `D0100`（圧力）を設定し、行数を3にした場合:

	速度	圧力
1	D0000 の値	D0100 の値
2	D0001 の値	D0101 の値
3	D0002 の値	D0102 の値

このように、行ごとにデバイスアドレスが +1 ずつオフセットされます。
出力されるCSVは以下のようになります。

	速度	圧力
1	1500	80
2	1520	81
3	1480	79



出力

- 保存先
  書き出し先のフォルダを指定します。未指定の場合はDocuments/DataLogs に保存されます。
- サブフォルダ
  サブフォルダの作成方法を選択します。
  - なし - サブフォルダを作成しない
  - プロジェクト名 - プロジェクト名のサブフォルダを作成
  - プロジェクト**UUID** - プロジェクトUUIDのサブフォルダを作成
- ファイル名
  テンプレートタグを使用してファイル名を指定します。
  - `{project}` - プロジェクト名
  - `{title}` - タイトル行の値
  - `{date}` - 日付 (yyyyMMdd)
  - `{time}` - 時刻 (HHmmss)
  - `{datetime}` - 日時 (yyyyMMdd_HHmmss)

・エンコーディング
書き出し時のCSVファイルの文字コードを指定します（ASCII, Shift_JIS, Japanese EUC, UTF-8, ISO Latin 1, ISO Latin 2, Non-lossy ASCII）。
・**BOM**
UTF-8の場合、BOM（バイトオーダーマーク）を付加するかどうかを指定します。Excelで開く場合はBOMを付加することを推奨します。
・タイトル行
CSVの先頭行にタイトル行を追加します。空欄の場合は追加されません。
・自動エクスポート
プロジェクト停止時に自動的にファイルを書き出す形式を選択します。
    ◦なし - 自動書き出しなし（内部ファイルのみ保存）
    ◦**CSV** - CSV形式で書き出し
    ◦**Excel** - Excel (XLSX) 形式で書き出し
・**Excel**罫線
Excel形式の場合、セルに罫線を追加します。自動エクスポートがExcelの場合のみ表示されます。
・タイムスタンプを除外 / 行番号を除外
エクスポート時にタイムスタンプ列（時系列モード）または行番号列（テーブルモード）を除外します。自動エクスポートが有効な場合のみ表示されます。



トリガー

記録のタイミングを設定します。

- 間隔
  指定した秒数ごとに記録します（0.1秒〜3600秒）。
- ビット立ち上がり
  指定したビットデバイスが OFF→ON になった時に記録します。
- ビット立ち下がり
  指定したビットデバイスが ON→OFF になった時に記録します。
- ビット変化
  指定したビットデバイスの値が変化した時に記録します。

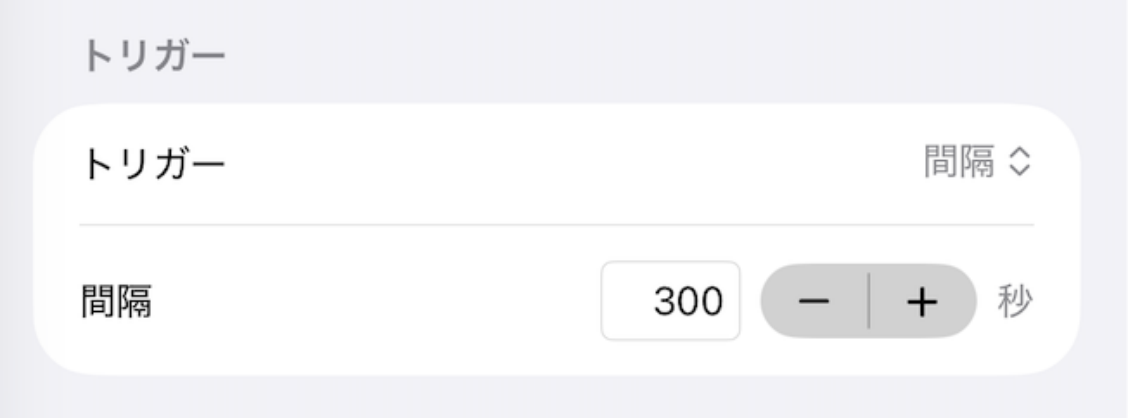
トリガーデバイス

ビットトリガーの場合、トリガーとなるビットデバイスを指定します。

後アクション

ビットトリガーの場合、記録後にトリガーデバイスを自動操作できます。

- なし - 何もしない
- セット - 記録後にトリガーデバイスをONにする
- リセット - 記録後にトリガーデバイスをOFFにする





デバイス設定

記録するデバイスを設定します。テーブルモードの場合は「列」と表示されます。

- デバイス
  PLCデバイス名を指定します（例: W0000, DM00000）。

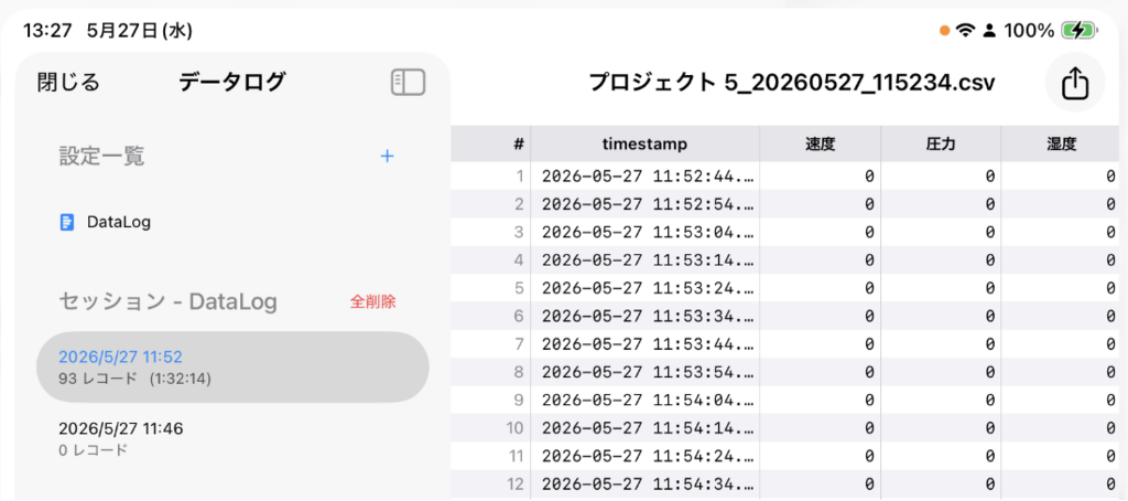
・ラベル
CSVのヘッダー列名として使用されます。
・表現形式
値の解釈方法を指定します。「自動」を選択するとデバイスの型に応じて自動判定されます。ビットデバイスの場合は自動的にBitになります。
  ◦ **Unsigned / Signed** - 16ビット 符号なし/符号あり整数
  ◦ **HEX** - 16ビット 16進数表示
  ◦ **BCD** - 16ビット BCD (Binary Coded Decimal)
  ◦ **Unsigned(DWORD) / Signed(DWORD)** - 32ビット 符号なし/符号あり整数
  ◦ **HEX(DWORD)** - 32ビット 16進数表示
  ◦ **BCD(DWORD)** - 32ビット BCD
  ◦ **Float** - 32ビット浮動小数点
  ◦ **Text** - 文字列（複数ワードを結合してデコード）。文字エンコーディングとバイト数を指定できます。
・連続数
連続するデバイスの数を指定します。1つのエントリで複数デバイスを一括登録できます。テーブルモードでは非表示です（行数で自動展開されるため）。Text型の場合は連続数の代わりにバイト数を指定します。



セッション

サイドバーの「セッション」セクションに記録されたセッションが一覧表示されます。

・セッションを選択するとCSVデータのプレビューが表示されます。
・プレビュー画面からCSVやExcel形式でエクスポートできます。
・スワイプで個別のセッションを削除できます（関連ファイルも削除されます）。
・「全削除」で全セッションを一括削除できます（確認ダイアログが表示されます）。

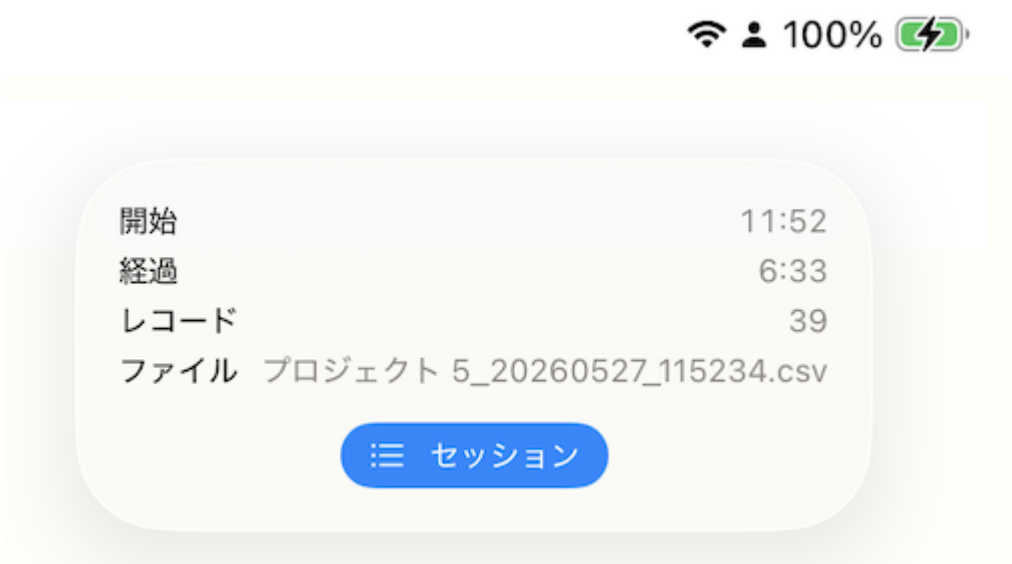


実行時の動作

プロジェクト実行時に、有効な設定のデータログが自動的に開始されます。

- 画面上部のナビゲーションバーに赤い点滅インジケータとレコード数が表示されます。
- インジケータをタップすると、各ロガーの開始時刻、経過時間、レコード数、ファイル名などが確認できます。
- 「セッション」ボタンからセッション一覧とプレビューを表示できます。
- 記録中のセッションには赤い丸が表示され、削除はできません。
- プロジェクトの実行を停止するとロギングが終了し、自動エクスポートが行われます。

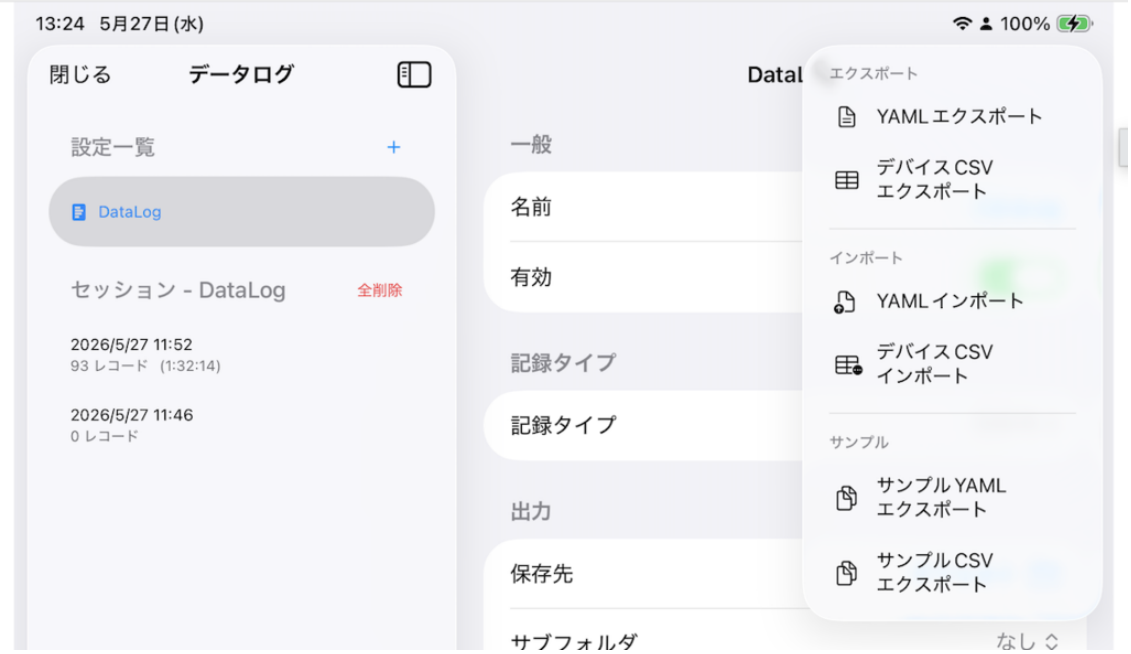




インポート / エクスポート

設定のインポート・エクスポートに対応しています。

- **YAML**
  設定全体（トリガー、出力設定、デバイス一覧）をYAML形式でエクスポート・インポートできます。
- デバイス**CSV**
  デバイス一覧のみをCSV形式でエクスポート・インポートできます。
  表計算ソフトでデバイス一覧を編集してからインポートする使い方ができます。
- サンプル
  サンプルのYAML・CSVをエクスポートして、インポート用のテンプレートとして使用できます。


13:24 5月27日 (水)
100%
閉じる
データログ
Datal
設定一覧
DataLog
セッション - DataLog
全削除
2026/5/27 11:52
93 レコード (1:32:14)
2026/5/27 11:46
0 レコード
一般
名前
有効
記録タイプ
記録タイプ
出力
保存先
サブフォルダ
なし
エクスポート
YAML エクスポート
デバイス CSV エクスポート
インポート
YAML インポート
デバイス CSV インポート
サンプル
サンプル YAML エクスポート
サンプル CSV エクスポート

YAMLプロジェクト形式

irBoardのプロジェクトはYAMLテキストファイルで作成・編集できます。 この形式は人間にもAIにも読みやすく設計されており、プロジェクトの自動生成が可能です。

プロジェクトは編集画面の プロジェクト設定 インスペクター、またはプロジェクト一覧画面のツールバーからインポートします。 「YAMLからインポート」を選択して、YAMLプロジェクトファイルを含むフォルダー、 または `project.yaml` ファイルを直接選択してインポートしてください。

YAMLの基本

YAML（YAML Ain't Markup Language）は構造化データを表現するテキスト形式です。インデント（スペース）で階層を表現するため、特別なツールなしで読み書きできます。

キーと値のペア

基本的なデータは `キー: 値` のペアで記述します:

```
name: "My Project"
rotation_type: 1
enabled: true
```

インデント = ネスト（入れ子）

ネスト構造はスペースによるインデントで表現します（1レベルにつき2スペース）:

```
plc:
  provider: KEYENCE
  config:
    ip_or_hostname: "192.168.0.10"
    port_no: 8501
```

リスト（配列）

リストの各要素は `-`（ハイフン + スペース）で始めます:

```
pages:
  - uuid: "page-001"
    page_no: 1
    title: Main
  - uuid: "page-002"
    page_no: 2
    title: Settings
```

データ型

型	例	備考
文字列	`"Hello"` または `Hello`	値に`:`、`#`、特殊文字を含む場合はクォートが必要。
整数	`42`	クォートなしで記述。
浮動小数点	`3.14`	クォートなしで記述。
真偽値	`true` / `false`	クォートなしで記述。

コメント

`#`で始まる行はコメントとして無視されます:

```
# これはコメントです
name: "My Project"   # インラインコメントも可能
```

フォルダー構成

YAMLプロジェクトは`project.yaml`ファイルとオプションの`images/`サブフォルダーを含むフォルダーです:

```
ProjectName/
  project.yaml       # プロジェクト設定（必須）
  images/             # 画像ファイル（オプション）
    {uuid}.png        # UUIDで参照される背景画像
```

`project.yaml` ファイルにはすべてのプロジェクト設定、ページ、アイテム、各種設定が含まれます。 プロジェクト、ページ、アイテムの背景画像は `images/` フォルダーにUUIDで命名されたPNGファイルとして格納されます。

最小構成の例

ラベルとボタンを1つずつ含む1ページの最小構成プロジェクトです:

```
uuid: "00000000-0000-0000-0000-000000000001"
name: "My Project"
screen_size_string: "768x1024"
rotation_type: 1

plc:
  model: KV-8000
  provider: KEYENCE
  plc_channel:
    config:
      ip_or_hostname: "192.168.0.10"
      port_no: 8501

pages:
  - uuid: "00000000-0000-0000-0000-100000000001"
    page_no: 1
    title: Main
    items:
      - uuid: "00000000-0000-0000-0000-200000000001"
        class: Label
        frame_string: "{{50, 50}, {200, 40}}"
        title: Temperature
        text_color_name: "RGBA,0.0,0.0,0.0,1.0"
      - uuid: "00000000-0000-0000-0000-200000000002"
        class: Button
        frame_string: "{{50, 120}, {120, 50}}"
        title: Start
        switch_device_name: MR100
        switch_mode: momentary
```

必須フィールド: `plc` と `pages` は必須です。 `plc` セクションがないとインポート時に工ラーになります。 `uuid` 、 `name` は省略可能で、省略時は自動生成されます。 各ページの `uuid` と `page_no` も省略時は自動補完されます。 各アイテムの `uuid` と `frame_string` も省略可能（デフォルト値が適用されます）。 `class` は必須ですが、小文字や別名（例: gauge, slider, alarm, grid等）も使用できます。

プロジェクトプロパティ

プロジェクト全体を構成するトップレベルのプロパティ:

プロパティ	型	説明
`uuid`	String	自動 一意の識別子（UUID形式）。省略時は自動生成。
`name`	String	自動 プロジェクト名。省略時は自動生成。
`plc`	Object	必須 PLC設定。省略するとインポートエラー。
`pages`	Array	必須 ページのリスト。
`screen_size_string`	String	画面サイズ「幅x高さ」（例: "768x1024"）。
`rotation_type`	Integer	画面回転。0=縦向き、1=横向き。
`style`	String	プロジェクトスタイル。"default"または"dark"。
`background_color_name`	String	デフォルト背景色。色形式のセクションを参照。
`launch_page_no`	Integer	起動時に表示するページ番号。
`page_scale_type`	Integer	画面スケーリング。0=中央、1=フィット、2=フィル。
`transition`	String	ページ遷移アニメーションスタイル。
`text_font_name`	String	デフォルトテキストフォント名。
`text_font_size`	Float	デフォルトテキストフォントサイズ。
`number_font_name`	String	数値表示のデフォルトフォント。
`number_font_size`	Float	数値表示のデフォルトフォントサイズ。

グリッド設定

```
grid:
  style: 1
  enabled: 1
  pixel: 10.0
```

プロパティ	型	説明
`style`	Integer	グリッドスタイル。0=ドット、1=ライン。
`enabled`	Integer	1=グリッド表示、0=非表示。
`pixel`	Float	グリッド間隔（ポイント）。

PLC設定

`plc` セクションでPLC接続を設定します。このセクションは必須です。省略するとインポートエラーになります。 対応PLCプロバイダーとモデルについては PLC接続 のドキュメントを参照してください。

KEYENCE の例

```
plc:
  model: KV-8000
  provider: KEYENCE
  plc_channel:
    config:
      ip_or_hostname: "192.168.0.10"
      port_no: 8501
```

三菱電機（MCプロトコル）の例

```
plc:
  model: iQ-R
  provider: MITSUBISHI
  plc_channel:
    config:
      ip_or_hostname: "192.168.0.10"
      port_no: 5000
```

オムロン（FINS）の例

```
plc:
  model: NJ
  provider: OMRON
  plc_channel:
    config:
```

```
      ip_or_hostname: "192.168.0.10"
      port_no: 9600
      self_node: 0
      destination_node: 0
```

MODBUS/TCPの例

```
plc:
  model: MODBUS
  provider: MODBUS
  plc_channel:
    config:
      ip_or_hostname: "192.168.0.10"
      port_no: 502
      unit_id: 1
```

PLCプロバイダー / モデル / チャンネル一覧

プロバイダー	モデル	デフォルトポート
キーエンス	KV-8000, KV-7500, KV-5500, KV-5000, KV-3000, KV-1000, KV-Nano	8501
三菱電機	iQ-R, iQ-F, Q, L, FX5, FX3	5000
オムロン	NJ, NX, CJ2, CJ1, CP1, CP2	9600
横河電機	FA-M3	12289
MODBUS	MODBUS	502

設定オプション

プロパティ	型	説明
`monitor_interval`	Float	ポーリング間隔（秒、デフォルト: 0.5）。
`button_default_delay`	Float	ボタン押下時のデフォルト遅延（秒）。
`slider_default_delay`	Float	スライダー操作時のデフォルト遅延（秒）。
`area_permission`	String	権限領域のデバイスアドレス。
`use_area_permission`	Boolean	

プロパティ	型	説明
		権限領域を有効にする。
`area_to_plc_screen_no`	String	現在の画面番号をPLCに送信するデバイスアドレス。
`use_area_to_plc_screen_no`	Boolean	画面番号のPLC通知を有効にする。
`area_to_plc_status`	String	irBoardのステータスをPLCに送信するデバイスアドレス。
`use_area_to_plc_status`	Boolean	ステータスのPLC送信を有効にする。
`area_from_plc_screen_no`	String	PLCから画面番号を受信するデバイスアドレス。
`use_area_from_plc_screen_no`	Boolean	PLCからの画面番号受信を有効にする。
`area_from_plc_status`	String	PLCからステータスを受信するデバイスアドレス。
`use_area_from_plc_status`	Boolean	PLCからのステータス受信を有効にする。

ページプロパティ

```
pages:
  - uuid: "page-uuid-1"
    page_no: 1
    title: Main
    background_color_name: "RGBA,0.95,0.95,0.95,1.0"
    items:
   - ...
```

プロパティ	型	説明
`uuid`	String	自動 一意の識別子。省略時は自動生成。
`page_no`	Integer	自動 ページ番号（1始まり）。省略時は自動補完。
`title`	String	ページタイトル。
`background_color_name`	String	ページ背景色。
`locked`	Integer	1=ロック（実行時編集不可）、0=アンロック。

プロパティ	型	説明
`rotation_type`	Integer	ページ固有の回転設定。
`background_image`	Object	背景画像参照 ({uuid, kind})。
`items`	Array	ページアイテムのリスト。

ページナビゲーション（スワイプ）

ページ間のスワイプナビゲーションをUUID参照で設定できます:

プロパティ	型	説明
`above_page_uuid`	String	上スワイプ時のページUUID。
`below_page_uuid`	String	下スワイプ時のページUUID。
`left_page_uuid`	String	左スワイプ時のページUUID。
`right_page_uuid`	String	右スワイプ時のページUUID。

アイテムタイプ

各アイテムには `class` フィールドでタイプを指定します:

class	説明	主要プロパティ
`Label`	テキスト表示	title, text_color_name
`Button`	ランプ付きボタン	switch_device_name, switch_mode, lamp_device_name
`Button`（スイッチなし）	状態表示インジケーター	lamp_device_name, lamp_color_name
`ValueField`	数値/テキストの表示・入力	device_name, editable, decimal_place, unit
`SteppedValueField`	+/-ボタン付き値（スライダー/進捗）	device_name, step_value, min_value, max_value
`Meter`	アナログゲージ表示	device_name, min_value, max_value, figures

class	説明	主要プロパティ
`Message`	メッセージリスト表示	message_set_uuid
`Matrix`	表/グリッドレイアウト	matrix_rows, matrix_columns, matrix_device_offset
`WebPage`	埋め込みWebブラウザ	url

アイテム共通プロパティ

すべてのアイテムに共通するプロパティ:

プロパティ	型	説明
`uuid`	String	自動 一意の識別子。省略時は自動生成。
`class`	String	必須 アイテムタイプ。アイテムタイプの表を参照。小文字や別名も使用可能（例: gauge→Meter、alarm→Message、grid→Matrix、slider→Slider、progress_bar→Progress、value_field→ValueField、web_page→WebPage）。
`frame_string`	String	自動 位置とサイズ。省略時はクラスに応じたデフォルトサイズ。形式: "{{x, y}, {width, height}}" または "x, y, width, height"
`title`	String	表示テキスト。
`device_name`	String	メインPLCデバイスアドレス。
`background_color_name`	String	背景色。
`line_color_name`	String	枠線の色。
`line_width`	Float	枠線の幅。
`corner_radius`	Integer	角丸の半径。
`style`	String	ビジュアルスタイル（アイテムタイプにより異なる）。
`font_name`	String	フォント名のオーバーライド。
`font_size`	Float	フォントサイズのオーバーライド。

プロパティ	型	説明
`alignment`	Integer	テキスト配置。0=左、1=中央、2=右。
`padding`	Integer	内部パディング。
`locked_frame`	Integer	1=フレームロック、0=編集可能。
`visible_device_name`	String	表示/非表示を制御するデバイス。
`visible_negative`	Integer	1=表示条件を反転。
`alternate_title`	String	代替タイトル（トグル状態用）。

ラベル

静的テキストまたはデバイス値に基づくテキストを表示します。

```
- uuid: "label-uuid"
  class: Label
  frame_string: "{{50, 50}, {200, 40}}"
  title: Temperature
  text_color_name: "RGBA,0.0,0.0,0.0,1.0"
  background_color_name: "RGBA,0.9,0.9,0.9,1.0"
  alignment: 1
  font_size: 18.0
```

プロパティ	型	説明
`text_color_name`	String	テキスト色。
`alternate_text_color_name`	String	代替状態でのテキスト色。
`label_device_name`	String	ラベル状態（ON/OFF外観）を制御するデバイス。
`label_negative`	Integer	1=ラベルデバイス条件を反転。

ボタン

PLCデバイスをON/OFFでき、ランプ状態を表示する対話型ボタンです。 スイッチデバイスを指定しない場合はランプインジケーターとして動作します。

```
- uuid: "button-uuid"
  class: Button
  frame_string: "{{50, 100}, {120, 50}}"
  title: Start
  alternate_title: Stop
  switch_device_name: MR100
  switch_mode: momentary
  lamp_device_name: MR100
  lamp_color_name: "RGBA,0.0,0.8,0.0,1.0"
  style: irPanel
```

スイッチプロパティ

プロパティ	型	説明
`switch_device_name`	String	押下時にON/OFFするビットデバイス。
`switch_mode`	String	"momentary"（押下中ON）または"alternate"（トグル）。
`switch_negative`	Integer	1=スイッチ出力を反転。
`delay_time`	Double	コマンド送信前の遅延（秒）。
`interlock_device_name`	String	ボタンを有効にするために必要なデバイス。
`interlock_negative`	Integer	1=インターロック条件を反転。

ランププロパティ

プロパティ	型	説明
`lamp_device_name`	String	ランプ状態を監視するビットデバイス。
`lamp_color_name`	String	ランプON時の色。
`alternate_lamp_color_name`	String	ランプOFF時の色。
`lamp_negative`	Integer	1=ランプ条件を反転。

ページナビゲーション

プロパティ	型	説明
`page_uuid`	String	押下時に遷移するページのUUID。
`page_no`	Integer	遷移先のページ番号。

ボタンスタイル

プロパティ	型	説明
`style`	String	"irPanel"、"Square"、"Circle"、"Touch"、"Plane"、"Toggle"。大文字小文字は自動補正されます。

値フィールド

PLCデバイスの数値やテキストを表示・編集するフィールドです。

```
- uuid: "value-uuid"
  class: ValueField
  frame_string: "{{300, 100}, {150, 40}}"
  title: Temperature
  device_name: DM100
  editable: 0
  decimal_place: 1
  unit: "\u00B0C"
  number_type: 2
  min_value: 0.0
  max_value: 999.0
  text_color_name: "RGBA,0.0,0.0,0.0,1.0"
  field_background_color_name: "RGBA,1.0,1.0,1.0,1.0"
```

プロパティ	型	説明
`device_name`	String	値を読み書きするワードデバイス。
`editable`	Integer	0=読み取り専用、1=実行時に編集可能。
`number_type`	Integer	0=符号なし16bit、1=符号付き16bit、2=符号なし32bit、3=符号付き32bit、4=float、5=double、6=BCD、10=文字列。

プロパティ	型	説明
`decimal_place`	Integer	小数点以下の桁数。
`unit`	String	単位（例: "\u00B0C"、"kg"、"%"）。
`coefficient`	Float	スケーリング係数（表示値 = 生値 * 係数 + オフセット）。
`offset`	Float	値オフセット。
`min_value`	Double	最小値。
`max_value`	Double	最大値。
`field_background_color_name`	String	入力フィールドの背景色。
`text_color_name`	String	表示値のテキスト色。
`unit_color_name`	String	単位テキストの色。
`border_style`	Integer	0=なし、1=ライン、2=ベゼル、3=丸角。
`encoding`	Integer	文字列型のエンコーディング。1=UTF-8。
`text_length`	Integer	文字列型の最大文字数。
`secret`	Boolean	true=入力を隠す（パスワードフィールド）。

ステップ値フィールド（スライダー/進捗）

+/-ボタン付きの値フィールド、またはスライダーやプログレスバーとして表示されます。

```
- uuid: "slider-uuid"
  class: SteppedValueField
  frame_string: "{{50, 200}, {300, 40}}"
  title: Speed
  device_name: DM200
  step_value: 1.0
  min_value: 0.0
  max_value: 100.0
  style: slider
```

プロパティ	型	説明
`step_value`	Double	1ステップあたりの増減量。
`style`	String	"default"（ボタン）、"slider"、"progress"。
`vertical`	Boolean	true=縦方向。
`continuous`	Boolean	true=ボタン長押し中に連続変更。

値フィールドのすべてのプロパティ（`device_name`、`min_value`、`max_value`、`number_type`など）を継承します。

メーター

PLCデバイスの値をアナログゲージで表示します。

```
- uuid: "meter-uuid"
  class: Meter
  frame_string: "{{50, 300}, {200, 200}}"
  title: Pressure
  device_name: DM300
  min_value: 0.0
  max_value: 100.0
  figures: 5
  major: 20
  minor: 10
  unit: MPa
  decimal_place: 1
```

プロパティ	型	説明
`min_value`	Double	スケールの最小値。
`max_value`	Double	スケールの最大値。
`figures`	Integer	スケールラベルの数。
`major`	Integer	主目盛りの間隔。
`minor`	Integer	副目盛りの間隔。
`meter_zones`	Array	メーター上のカラーゾーン。

メーターゾーン

```
meter_zones:
  - color_name: "RGBA,0.0,0.8,0.0,1.0"
    from_value: 0.0
    to_value: 60.0
  - color_name: "RGBA,1.0,0.8,0.0,1.0"
    from_value: 60.0
    to_value: 80.0
  - color_name: "RGBA,1.0,0.0,0.0,1.0"
    from_value: 80.0
    to_value: 100.0
```

ランプ (状態表示インジケーター)

スタンドアロンの状態表示インジケーター。ボタンと同じクラス (`Button`) で、スイッチデバイスを指定しないことでランプとして動作します。

```
- uuid: "lamp-uuid"
  class: Button
  frame_string: "{{300, 50}, {60, 60}}"
  title: Running
  lamp_device_name: MR200
  lamp_color_name: "RGBA,0.0,0.8,0.0,1.0"
  alternate_lamp_color_name: "RGBA,0.5,0.5,0.5,1.0"
  style: Circle
```

ボタンと同じランププロパティ (`lamp_device_name` 、 `lamp_color_name` 、 `lamp_negative` など) を使用します。 `switch_device_name` を省略すると、ランプインジケーターになります。

メッセージ

メッセージセットからのメッセージリストを表示します。メッセージはPLCデバイス値に基づいてアクティブになります。

```
- uuid: "message-uuid"
  class: Message
  frame_string: "{{50, 400}, {400, 200}}"
  message_set_uuid: "msgset-uuid-1"
  active_color_name: "RGBA,1.0,0.0,0.0,1.0"
```

```
  inactive_color_name: "RGBA,0.9,0.9,0.9,1.0"
  active_only: true
```

プロパティ	型	説明
`message_set_uuid`	String	表示するメッセージセットのUUID。
`active_color_name`	String	アクティブメッセージの背景色。
`inactive_color_name`	String	非アクティブメッセージの背景色。
`active_text_color_name`	String	アクティブメッセージのテキスト色。
`inactive_text_color_name`	String	非アクティブメッセージのテキスト色。
`active_only`	Boolean	true=アクティブメッセージのみ表示。
`row_height`	Integer	メッセージ行の高さ。
`show_detail`	Boolean	true=詳細テキストを表示。
`show_title`	Boolean	true=メッセージセットタイトルヘッダーを表示。

マトリックス

テンプレートアイテムを行列で繰り返し、デバイスアドレスオフセットで展開する表/グリッドレイアウトです。 詳細は マトリックス のドキュメントを参照してください。

```
- uuid: "matrix-uuid"
  class: Matrix
  frame_string: "{{50, 50}, {600, 400}}"
  matrix_rows: 5
  matrix_columns: 3
  matrix_device_offset: 10
  matrix_add_borders: true
  matrix_border_color_name: "RGBA,0.7,0.7,0.7,1.0"
  matrix_has_column_header: true
  matrix_has_row_header: true
```

プロパティ	型	説明
`matrix_rows`	Integer	データ行数。

プロパティ	型	説明
`matrix_columns`	Integer	データ列数。
`matrix_device_offset`	Integer	行間のデバイスアドレス増分。
`matrix_add_borders`	Boolean	true=セル周りに枠線を追加。
`matrix_border_color_name`	String	枠線の色。
`matrix_border_width`	Float	枠線の幅。
`matrix_has_column_header`	Boolean	true=列ヘッダーを含む。
`matrix_has_row_header`	Boolean	true=行ヘッダーを含む。
`matrix_cell_bg_color_name`	String	デフォルトセル背景色。
`matrix_alt_bg_color_name`	String	交互行の背景色。
`matrix_column_header_bg_color_name`	String	列ヘッダーの背景色。
`matrix_row_header_bg_color_name`	String	行ヘッダーの背景色。

マトリックステンプレート

マトリックスはセルテンプレートを定義でき、インポート時に全行列にわたって自動展開されます。 テンプレートは各セルに表示するアイテムを定義し、デバイスオフセットルールがグリッド全体のデバイスアドレスの増分を制御します。

テンプレートは `matrix_template_json` で定義します。 YAML形式でアイテム定義の配列として直接記述できます（JSON文字列でも可）。 座標はセル原点(0, 0)からの相対位置です。 テンプレートがあるが未展開のマトリックスは、YAMLインポート時に自動的に展開されます。

```
- uuid: "matrix-uuid"
  class: Matrix
  frame_string: "{{50, 50}, {600, 300}}"
  matrix_rows: 3
  matrix_columns: 4
  matrix_add_borders: true
  matrix_border_color_name: "RGBA,0.7,0.7,0.7,1.0"
  matrix_template_cell_width: 140.0
  matrix_template_cell_height: 60.0
  matrix_template_json:
    - class: Label
      uuid: "tpl-001"
```

```
      frame_string: "{{2, 2}, {60, 26}}"
      title: CH1
      font_size: 12.0
      text_color_name: "RGBA,0.3,0.3,0.3,1.0"
    - class: ValueField
      uuid: "tpl-002"
      frame_string: "{{2, 30}, {136, 28}}"
      device_name: DM100
      number_type: 2
      decimal_place: 1
      unit: "\u00B0C"
      text_color_name: "RGBA,0.0,0.0,0.0,1.0"
      field_background_color_name: "RGBA,1.0,1.0,1.0,1.0"
      border_style: 3
  matrix_device_offset_rules:
    - id: "rule-1"
      itemUUID: "*"
      property: device_name
      rowOffset: 10
      colOffset: 1
```

プロパティ	型	説明
`matrix_template_json`	配列	1セル分のアイテム定義の配列。通常のアイテムと同じプロパティを使用。
`matrix_template_cell_width`	Float	テンプレートアイテムの基準セル幅。
`matrix_template_cell_height`	Float	テンプレートアイテムの基準セル高さ。

デバイスオフセットルール

`matrix_device_offset_rules` は行・列間でデバイスアドレスをどのように変化させるかを制御するルールの配列です。

プロパティ	型	説明
`id`	String	ルールの一意の識別子。
`itemUUID`	String	適用先のテンプレートアイテムUUID。"*"は全アイテムに適用。
`property`	String	デバイスプロパティ名。"*"は全デバイスプロパティに適用。選択肢: device_name, switch_device_name, lamp_device_name, visible_device_nameなど。
`rowOffset`	Integer	行あたりのデバイスアドレス増分。

プロパティ	型	説明
`colOffset`	Integer	列あたりのデバイスアドレス増分。

各セルのデバイスアドレスは以下の式で計算されます:
`offset = row * rowOffset + col * colOffset`

テンプレート例の説明

上記の例は3x4の温度監視グリッドを作成します:

- 各セルにはラベル（"CH1"）とDM100を読み取る値フィールドが含まれます
- タイトルテキストは自動的にインクリメント: CH1, CH2, CH3, ...
- デバイスオフセットルール: `rowOffset: 10, colOffset: 1`
  セル(0,0)はDM100、セル(0,1)はDM101、セル(1,0)はDM110を読み取ります。

Webページ

ページ内にWebブラウザビューを埋め込みます。

```
- uuid: "web-uuid"
  class: WebPage
  frame_string: "{{50, 50}, {600, 400}}"
  url: "https://example.com"
```

プロパティ	型	説明
`url`	String	読み込むURL。
`user_agent`	String	カスタムUser-Agent文字列。

ローカライズ文字列

アイテムテキストの翻訳はプロジェクトレベルの `localizedStrings` 配列で定義します。各エントリはアイテムUUIDとフィールドキーを特定のロケールの翻訳文字列にマッピングします。

```
localizedStrings:
  - localeIdentifier: ja
    key: "item-uuid-1.title"
```

```
    text: "\u30B9\u30BF\u30FC\u30C8"
  - localeIdentifier: ja
    key: "item-uuid-1.alternate_title"
    text: "\u30B9\u30C8\u30C3\u30D7"
  - localeIdentifier: fr
    key: "item-uuid-1.title"
    text: "D\u00E9marrer"
```

プロパティ	型	説明
`localeIdentifier`	String	ロケールコード（例: "ja"、"en"、"fr"、"de"、"zh-Hans"）。
`key`	String	形式: "{item_uuid}.{field_name}"（例: "xxx.title"、"xxx.unit"）。
`text`	String	翻訳テキスト。

ローカライズ可能なフィールド

- `title` — アイテムタイトル
- `alternate_title` — 代替タイトル（ボタンのトグル状態）
- `unit` — 単位テキスト（ValueField、Meter）
- `message_` — メッセージテキスト（MessageItem）
- `detail_message` — 詳細メッセージテキスト（MessageItem）
- `title_` — メッセージセットタイトル

メッセージセット

メッセージセットはメッセージアイテムで表示されるメッセージのグループを定義します。 各メッセージはPLCデバイス値によってトリガーされます。

```
messageSets:
  - uuid: "msgset-uuid-1"
    no: 1
    title_: Alarms
    order_style: 0
    messageItems:
      - device_name: DM500
        value: 1.0
        number_type: 0
        message_: "Motor overload"
        detail_message: "Motor overload detected. Check motor wiring."
        disabled: false
```

```
        order_style: 0
      - device_name: DM500
        value: 2.0
        number_type: 0
        message_: "Temperature high"
        detail_message: "Temperature exceeds the limit."
        disabled: false
        order_style: 1
```

プロパティ	型	説明
`uuid`	String	一意の識別子。
`no`	Integer	メッセージセット番号。
`title_`	String	メッセージセットタイトル（末尾のアンダースコアは必須）。
`order_style`	Integer	ソート順。
`messageItems`	Array	メッセージアイテムのリスト。

メッセージアイテムのプロパティ

プロパティ	型	説明
`device_name`	String	メッセージをトリガーするデバイス。
`value`	Double	メッセージをトリガーする値。
`number_type`	Integer	データ型（ValueFieldと同じ）。
`message_`	String	メッセージテキスト（末尾のアンダースコアは必須）。
`detail_message`	String	詳細メッセージテキスト。
`disabled`	Boolean	true=メッセージを無効化。

色形式

色はRGBA形式の文字列で指定します:

```
"RGBA,R,G,B,A"
```

R、G、B、Aは0.0から1.0の浮動小数点値です。

例

色	値
黒	`"RGBA,0.0,0.0,0.0,1.0"`
白	`"RGBA,1.0,1.0,1.0,1.0"`
赤	`"RGBA,1.0,0.0,0.0,1.0"`
緑	`"RGBA,0.0,0.8,0.0,1.0"`
青	`"RGBA,0.0,0.0,1.0,1.0"`
黄	`"RGBA,1.0,1.0,0.0,1.0"`
灰	`"RGBA,0.5,0.5,0.5,1.0"`
薄灰	`"RGBA,0.9,0.9,0.9,1.0"`
透明	`"RGBA,0.0,0.0,0.0,0.0"`

フレーム形式

アイテムの位置とサイズはCGRect文字列形式で指定します:

```
"{{x, y}, {width, height}}"
```

値はポイント単位です。原点(0, 0)はページの左上隅です。 座標系は `screen_size_string` の寸法に対応します。

例

```
"{{0, 0}, {768, 1024}}"      # フルページ（縦向き）
"{{50, 50}, {200, 40}}"      # 左上付近の小さなアイテム
"{{384, 512}, {100, 50}}"    # 中央付近のアイテム（概算）
```

`screen_size_string: "768x1024"` 、 `rotation_type: 1` （横向き）のプロジェクトでは、 座標空間は幅1024ポイント、高さ768ポイントになります。

背景画像

プロジェクト、ページ、アイテムの背景画像を使用するには:

1. `images/` フォルダーにUUIDで命名されたPNG画像ファイルを配置します (例: `abc-123.png`) 。
2. YAMLで `background_image` オブジェクトを使って画像を参照します:

```
background_image:
  uuid: "abc-123"
  kind: page_background
```

画像の種類

種類	説明
`page_default_background`	全ページのデフォルト背景。
`page_background`	特定ページの背景。
`lamp_background`	ランプ/ボタン状態の背景画像。
`page_item_background`	ページアイテムの背景画像。

PLCデバイス名

デバイス名はPLCプロバイダーによって異なります。各プロバイダーの一般的なデバイス名は以下の通りです:

キーエンス (**KV**シリーズ)

デバイス	種類	説明
`DM0` 、 `DM100`	ワード	データメモリ
`MR0` 、 `MR100`	ビット	内部リレー
`R0` 、 `R100`	ビット	リレー
`FM0` 、 `FM100`	ワード	ファイルメモリ
`CM0` 、 `CM100`	ワード	カウンター

デバイス	種類	説明
`T0`、`T100`	ワード	タイマー

三菱電機（**MC**プロトコル）

デバイス	種類	説明
`D0`、`D100`	ワード	データレジスタ
`M0`、`M100`	ビット	内部リレー
`Y0`、`Y10`	ビット	出力（16進アドレス）
`X0`、`X10`	ビット	入力（16進アドレス）
`W0`、`W100`	ワード	リンクレジスタ（16進アドレス）
`R0`、`R100`	ワード	ファイルレジスタ

オムロン（**FINS**）

デバイス	種類	説明
`D0`、`D100`	ワード	DMエリア
`CIO0`、`CIO100`	ワード	CIOエリア
`W0`、`W100`	ワード	ワークエリア
`H0`、`H100`	ワード	保持エリア

MODBUS/TCP

デバイス	種類	説明
`HR0`、`HR100`	ワード	保持レジスタ (40001+)
`CO0`、`CO100`	ビット	コイル (00001+)
`IR0`、`IR100`	ワード	入力レジスタ (30001+)
`DI0`、`DI100`	ビット	ディスクリート入力 (10001+)

注意: ビットデバイス (MR、M、COなど) は `switch_device_name` 、 `lamp_device_name` 、 `visible_device_name` に使用します。 ワードデバイス (DM、D、HRなど) は `device_name` (値の読み書き) に使用します。

完全な例

KEYENCE KV-8000 PLCを使用した、2ページ、各種アイテム、メッセージセット、日本語ローカライズを含む完全なプロジェクトの例:

```
uuid: "a1b2c3d4-e5f6-7890-abcd-ef1234567890"
name: "Factory Monitor"
screen_size_string: "768x1024"
rotation_type: 1
style: default
background_color_name: "RGBA,0.95,0.95,0.95,1.0"

grid:
  style: 1
  enabled: 1
  pixel: 10.0

plc:
  model: KV-8000
  provider: KEYENCE
  plc_channel:
    config:
      ip_or_hostname: "192.168.0.10"
      port_no: 8501

pages:
  - uuid: "page-0001"
    page_no: 1
    title: Main
    right_page_uuid: "page-0002"
    items:
      - uuid: "item-0001"
        class: Label
        frame_string: "{{20, 20}, {300, 40}}"
        title: Factory Monitor
        text_color_name: "RGBA,0.0,0.0,0.0,1.0"
        font_size: 24.0
        alignment: 0
      - uuid: "item-0002"
        class: Button
        frame_string: "{{20, 80}, {150, 60}}"
        title: Start
```

```
      alternate_title: Stop
      switch_device_name: MR100
      switch_mode: alternate
      lamp_device_name: MR100
      lamp_color_name: "RGBA,0.0,0.8,0.0,1.0"
      style: irPanel
    - uuid: "item-0003"
      class: Button
      frame_string: "{{200, 80}, {60, 60}}"
      title: Run
      lamp_device_name: MR200
      lamp_color_name: "RGBA,0.0,0.8,0.0,1.0"
      alternate_lamp_color_name: "RGBA,0.5,0.5,0.5,1.0"
      style: Circle
    - uuid: "item-0004"
      class: Label
      frame_string: "{{20, 170}, {120, 30}}"
      title: Temperature
      text_color_name: "RGBA,0.3,0.3,0.3,1.0"
    - uuid: "item-0005"
      class: ValueField
      frame_string: "{{150, 160}, {150, 50}}"
      device_name: DM100
      number_type: 2
      decimal_place: 1
      unit: "\u00B0C"
      min_value: 0.0
      max_value: 999.0
      text_color_name: "RGBA,0.0,0.0,0.0,1.0"
      field_background_color_name: "RGBA,1.0,1.0,1.0,1.0"
      border_style: 3
    - uuid: "item-0006"
      class: Meter
      frame_string: "{{20, 230}, {280, 280}}"
      title: Pressure
      device_name: DM200
      number_type: 2
      min_value: 0.0
      max_value: 100.0
      figures: 6
      major: 20
      minor: 10
      unit: MPa
      decimal_place: 1
      meter_zones:
        - color_name: "RGBA,0.0,0.8,0.0,1.0"
          from_value: 0.0
          to_value: 60.0
        - color_name: "RGBA,1.0,0.8,0.0,1.0"
```

```
            from_value: 60.0
            to_value: 80.0
          - color_name: "RGBA,1.0,0.0,0.0,1.0"
            from_value: 80.0
            to_value: 100.0
      - uuid: "item-0007"
        class: SteppedValueField
        frame_string: "{{350, 160}, {300, 50}}"
        title: Speed
        device_name: DM300
        step_value: 5.0
        min_value: 0.0
        max_value: 100.0
        number_type: 2
        unit: "%"
        style: slider
      - uuid: "item-0008"
        class: Button
        frame_string: "{{850, 20}, {120, 40}}"
        title: "Alarms >>"
        page_uuid: "page-0002"

  - uuid: "page-0002"
    page_no: 2
    title: Alarms
    left_page_uuid: "page-0001"
    items:
      - uuid: "item-0009"
        class: Label
        frame_string: "{{20, 20}, {200, 40}}"
        title: Alarm List
        text_color_name: "RGBA,0.0,0.0,0.0,1.0"
        font_size: 24.0
      - uuid: "item-0010"
        class: Message
        frame_string: "{{20, 80}, {600, 400}}"
        message_set_uuid: "msgset-0001"
        active_color_name: "RGBA,1.0,0.9,0.9,1.0"
        inactive_color_name: "RGBA,0.95,0.95,0.95,1.0"
        active_text_color_name: "RGBA,0.8,0.0,0.0,1.0"
        inactive_text_color_name: "RGBA,0.3,0.3,0.3,1.0"
        show_detail: true
        show_title: true
      - uuid: "item-0011"
        class: Button
        frame_string: "{{850, 20}, {120, 40}}"
        title: "<< Main"
        page_uuid: "page-0001"
```

```
messageSets:
  - uuid: "msgset-0001"
    no: 1
    title_: Alarms
    order_style: 0
    messageItems:
      - device_name: DM500
        value: 1.0
        number_type: 0
        message_: "Motor overload"
        detail_message: "Motor overload detected. Check wiring."
        disabled: false
        order_style: 0
      - device_name: DM500
        value: 2.0
        number_type: 0
        message_: "Temperature high"
        detail_message: "Temperature exceeds limit."
        disabled: false
        order_style: 1
      - device_name: DM500
        value: 3.0
        number_type: 0
        message_: "Emergency stop"
        detail_message: "Emergency stop activated."
        disabled: false
        order_style: 2

localizedStrings:
  - localeIdentifier: ja
    key: "item-0001.title"
    text: "\u5DE5\u5834\u30E2\u30CB\u30BF\u30FC"
  - localeIdentifier: ja
    key: "item-0002.title"
    text: "\u30B9\u30BF\u30FC\u30C8"
  - localeIdentifier: ja
    key: "item-0002.alternate_title"
    text: "\u30B9\u30C8\u30C3\u30D7"
  - localeIdentifier: ja
    key: "item-0003.title"
    text: "\u904B\u8EE2\u4E2D"
  - localeIdentifier: ja
    key: "item-0004.title"
    text: "\u6E29\u5EA6"
  - localeIdentifier: ja
    key: "item-0006.title"
    text: "\u5727\u529B"
  - localeIdentifier: ja
    key: "item-0007.title"
```

```
    text: "\u901F\u5EA6"
  - localeIdentifier: ja
    key: "item-0009.title"
    text: "\u30A2\u30E9\u30FC\u30E0\u4E00\u89A7"
```

AIプロジェクト生成のヒント

- 自動補完: プロジェクト・ページ・アイテムの `uuid` は省略可能で、インポート時に自動生成されます。 `name` 、 `page_no` 、 `frame_string` も省略時にデフォルト値が自動補完されます。
- **PLC設定は必須:** `plc` セクションは必ず指定してください。省略するとインポートエラーになります。 `provider` と `model` を指定すれば、PLC内部のclass_nameは自動解決されます。
- クラス名の別名: `class` は大文字小文字を問いません。また以下の別名が使えます: gauge→Meter、alarm→Message、grid→Matrix、slider→Slider、progress_bar→Progress、value_field→ValueField、web_page→WebPage。
- **frame_stringの代替形式:** "{{x, y}, {width, height}}" の他に "x, y, width, height" 形式も使用できます。
- 座標系: 横向きプロジェクト（rotation_type: 1）でscreen_size_string "768x1024"の場合、使用可能な領域は幅1024 x 高さ768です。 この範囲を超えてアイテムを配置しないでください。
- デバイス名の一貫性: デバイス名がPLCプロバイダーの構文に合っていることを確認してください。スイッチ/ランプにはビットデバイス、値にはワードデバイスを使用します。
- 最小間隔: 読みやすさのため、アイテム間に少なくとも5〜10ポイントの間隔を空けてください。
- フォントサイズ: 推奨サイズ: タイトル18〜24pt、ラベル14〜16pt、値16〜20pt。
- メッセージセット: メッセージアイテムの `message_set_uuid` は `messageSets` 配列のUUIDと一致する必要があります。
- ローカライズ: localizedStringsの `key` は"{item_uuid}.{field_name}"形式です。 一般的なフィールド: "title"、"alternate_title"、"unit"。
- インポート動作: インポート時に同じUUIDのプロジェクトが存在する場合は上書きされます。 新しいコピーを作成するにはUUIDを変更してください。
- **Lite版の制限:** irBoard Liteでは、インポート時にプロジェクトが最大2ページ、1ページあたり最大10アイテムに切り詰められます。 超過分のページやアイテムはインポート時に自動的に削除されます。切り詰めが行われた場合はインポート後に通知が表示されます。

ページ単位インポート

プロジェクト全体のインポートに加えて、個別のページをYAMLファイルからインポートできます。 既存のプロジェクトにページを追加したり、エクスポートしたページを再インポートして更新するのに便利です。

インポート方法

プロジェクト編集画面のページリストにあるインポートボタン（ ）からYAMLファイルを選択します。

クリップボードからの貼り付け

ページリストのクリップボードボタン（ ）をタップすると、クリップボードにコピーされたYAMLテキストからページをインポートできます。 AIで生成したYAMLテキストをコピーして、そのまま貼り付けてインポートする使い方に便利です。 ファイルからのインポートと同じく、重複検出・上書き確認が動作します。

YAML形式

ページ単位のYAMLは2つの形式をサポートし、自動判定されます:

単一ページ（**items**トップレベル）

```
title: "操作パネル"
items:
  - class: Button
    frame_string: "{{35, 110}, {100, 100}}"
    title: 運転
    switch_device_name: M1000
    switch_mode: momentary
```

複数ページ（**pages**配列）

```
pages:
  - title: "Page 1"
    items:
      - class: Label
        frame_string: "{{10, 10}, {200, 50}}"
        title: "Status"
  - title: "Page 2"
    items:
      - class: Button
```

```
        frame_string: "{{10, 10}, {100, 100}}"
        title: "Start"
        switch_device_name: M0
```

エクスポート

ページリストでページを長押し（コンテキストメニュー）して「Export YAML」を選択すると、 そのページをYAMLファイルとしてエクスポートできます。 エクスポートされたファイルは `pages` 配列形式で、そのまま再インポートできます。

重複検出と上書き

インポート時、YAMLに含まれるページが既存のプロジェクト内のページと一致する場合は確認ダイアログが表示されます。 一致はUUID（優先）またはページ番号で判定されます。

- **上書き:** 既存ページを削除し、同じ位置（ページ番号）に新しい内容で置き換えます。
- **新規追加:** 新しいUUIDとページ番号を割り当て、末尾に追加します。
- **キャンセル:** インポートを中止します。

自動補完

プロジェクトインポートと同様に、ページ単位インポートでも以下が自動補完されます:

- `uuid` — 省略時は自動生成
- `page_no` — 省略時は既存ページの次の番号を割り当て
- `frame_string` — 省略時はクラスに応じたデフォルトサイズ
- クラス名の正規化（大文字小文字、別名の自動変換）
- ボタンの `switch_mode` デフォルト（momentary）
- `lamp_device_name` → `device_name` へのマッピング

注意: ページ単位のインポートでは `plc` 設定は不要です。 PLC設定はインポート先のプロジェクトのものが使用されます。

デモ動画

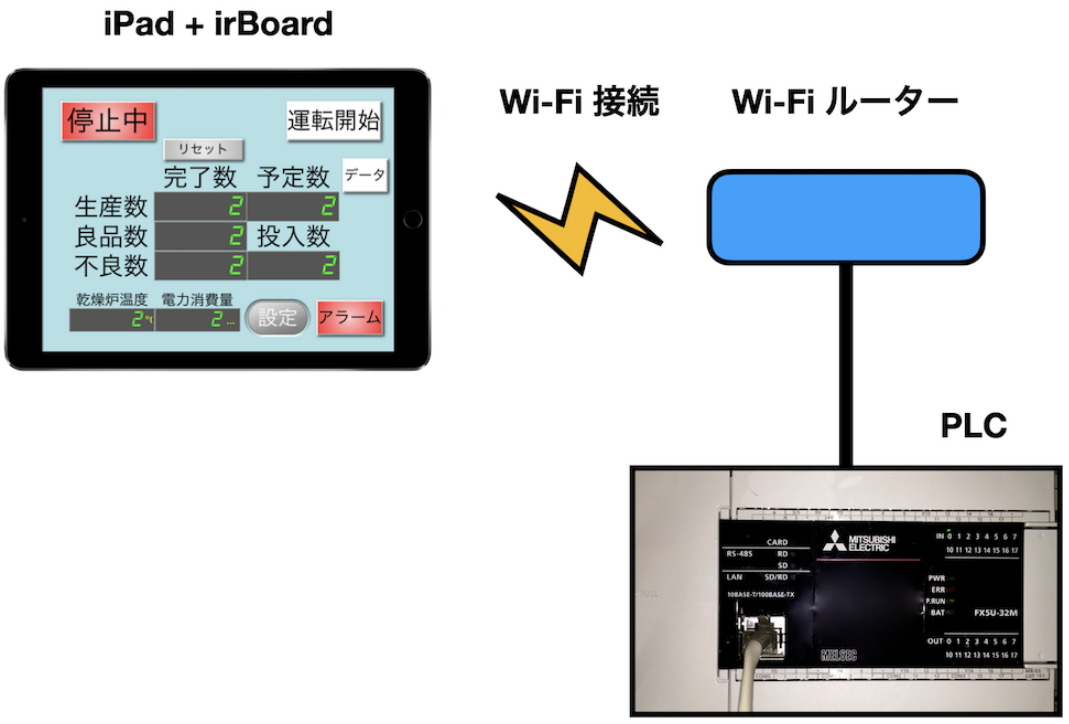
YAMLプロジェクト形式を使ったインポートのデモです。 AIにYAMLファイルを生成させ、irBoardにインポートして画面を構築する流れを紹介しています。

PLC 接続

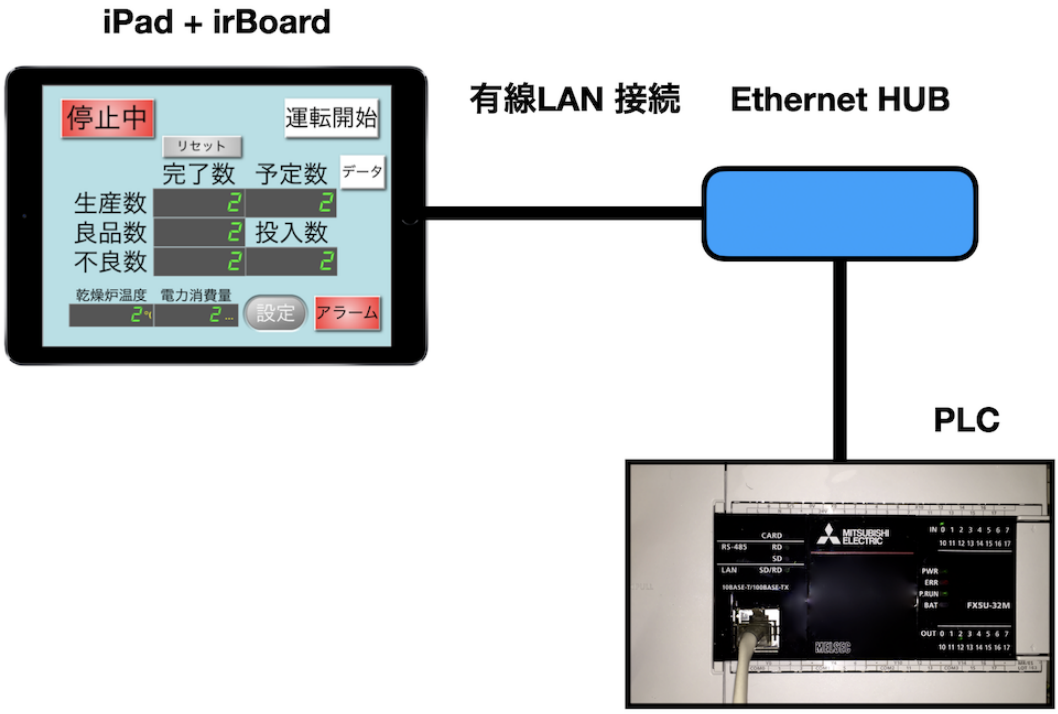
接続概要

irBoardとPLCを接続するには次の様な構成が一般的です。

- irBoardはiPadやiPhoneなどのiOSデバイスで動作するためWiFi接続が基本になります。
- WiFi接続のためにWiFiルーターを準備します。既にWiFi環境が整っている場合は不要です。
- PLCとWiFiルーター間は有線LANケーブルで接続します。
  途中HUBを経由することもあります。
- WiFiルーターの設定についてはメーカーの説明書を参考に設定してください。 **(多種多様な機種があるため弊社ではサポート致しかねます。)**
  基本的にはDHCPなどでIPアドレスを自動設定できる様にしてiOSデバイスと接続します。
  DHCPの自動割り当て範囲外のIPアドレスをPLCの固定アドレスとして使用します。
- ver 1.8.1以前はEhternet-Serialアダプターを介したシリアル接続が可能でしたがver 2.0以降は対応しておりません。



iPadと有線LANで接続することも可能です。
こちらの記事を参考にしてください。
有線LANでの接続



PLC設定

PLC設定は各メーカ毎のリンクをクリックして確認してください。

- 三菱電機: MCプロトコル(Binary)
- キーエンス: 上位リンク
- オムロン: FINS/TCP
- 横河電機: 上位リンク
- **Modbus:** Modbus/TCP
- **IoT:** LadderDrive

接続確認

設定が終わったら通信できるか確認します。

PLCとのIP接続確認

まずはPLCとIP接続できるか確認します。既にGX WorksやKV Studioなどの開発ツールをPLCとLANで接続している場合はここのIP接続確認の必要はありません。ここを飛ばしてポート番号の確認をしてください。

PCから確認

PCから ping コマンドを使用して接続確認します。

- PCとPLCを同一LAN上で接続します。
- 必要に応じてPC側のLAN設定をします。
- スタートボタン横の検索窓にcmdと入力すると検索結果にコマンドプロンプトが表示されますのでクリックし起動します。Windows Power Shelでも可能です。



・コマンドプロンプトが起動したら ping コマンドで接続を確認します。 ping コマンドは ping の後に PLCに割り付けたIPアドレスを入力します。

```
例) ping 192.168.1.10
```

- ここで接続が確認できない場合は **PLC**や**PC**設定、**WiFi**ルーターなどの**WiFi**環境を確認してください。
- 接続が確認できたらポート番号の確認をします。



PLCのポート番号接続確認

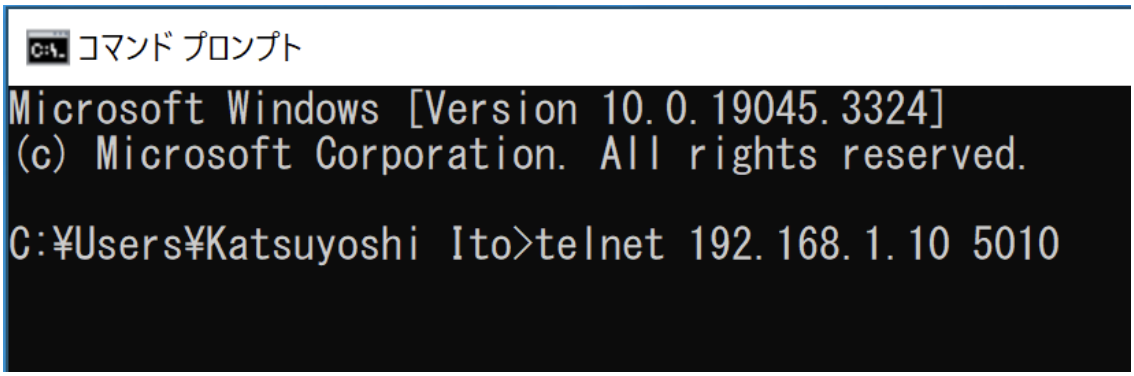
ポート番号との接続確認をするには telnet コマンドを使用します。

Windowsでは初期状態では telnet コマンドが有効になっていないため有効にする必要があります。
具体的な手順は [window telent 有効] で検索するとでてきますのでそちらをご覧ください。

- IP接続確認同様にコマンドプロンプトを用います。
- コマンドプロンプトが起動したら telnet コマンドで接続を確認します。 telnet コマンドは telnet の後に PLCに割り付けたIPアドレスとポート番号を入力します。

```
例) telnet 192.168.1.10 5010
```

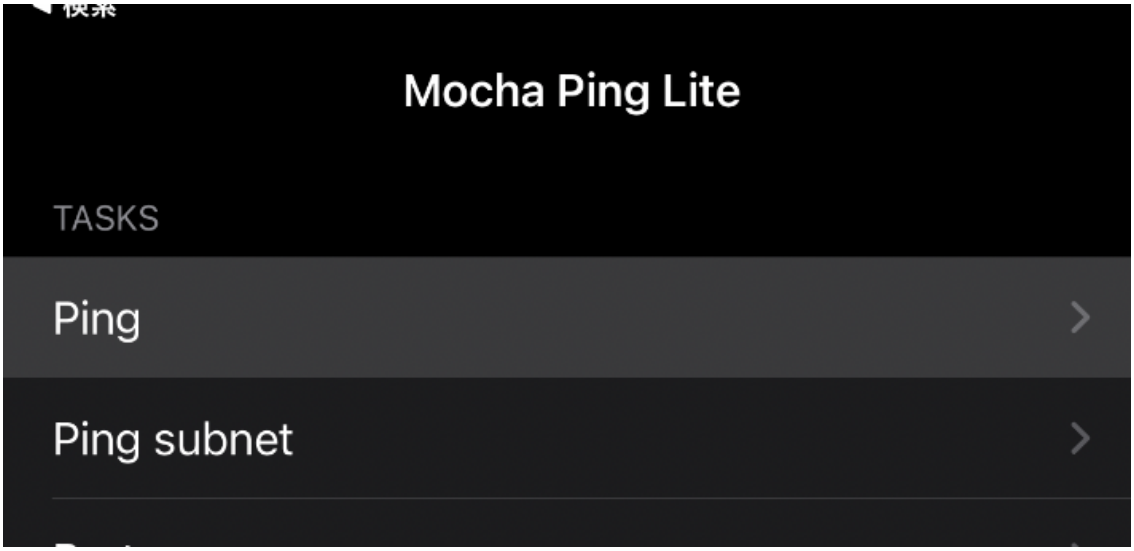
- 正常に接続すると一見画面に何も表示されていない様に見えますが、これが正常な状態です。CTRL +] で終了できますが、うまくいかない場合はコマンドプロンプトを閉じてください。
- 接続エラーが表示される場合は **PLC**のポート設定を確認してください。

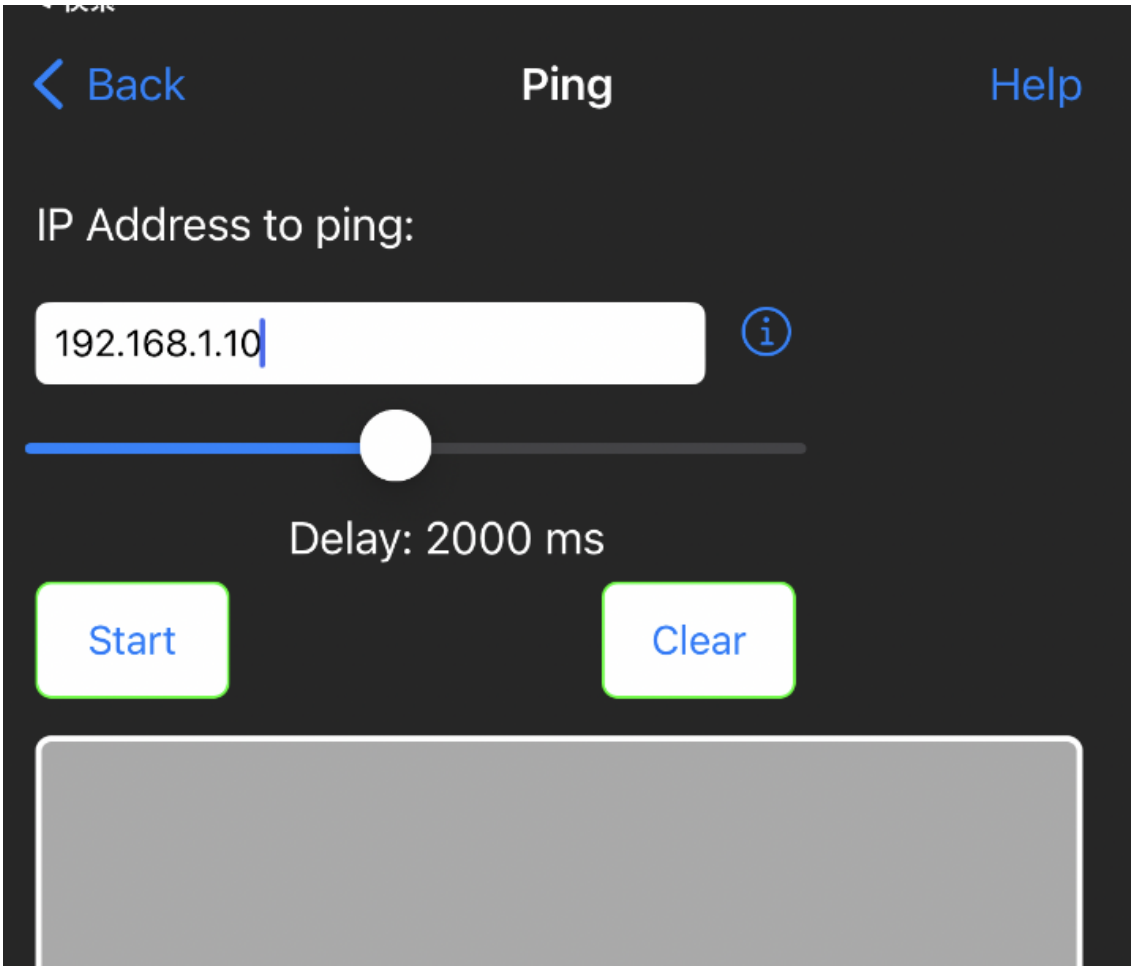


irBoardからの接続確認

irBoardから接続する際にPC同様に ping を使って接続確認するのが確実です。

- MochaSoft社の Network Ping Lite という素晴らしいアプリがありますので、iOSデバイスにインストールします。
- iOSデバイスのWiFi設定でWiFiルーターのSSIDと接続します。
- Network Ping Liteを起動します。
- TASKSからpingをタッチしてます。
- Ping画面でIP Address to ping:の欄にPLCのIPアドレスを入力します。
- Startボタンを押すと ping コマンドが発行され結果が下に表示されます。
- PLCとの接続が確認できない場合は **WiFi**ルータの設定や**WiFi**環境を確認してください。
  プレス機などのノイズを発生する装置がある箇所では 電波状況が悪く接続できない場合があります。
- 接続が確認できたらirBoardの プロジェクト設定 でPLC接続設定に合わせてIPアドレスやポート番号などを設定します。
- プロジェクトを実行 させて接続します。




Back
Ping
Help
IP Address to ping:
192.168.1.10
Delay: 2000 ms
Start
Clear

三菱電機: MC プロトコル(Binary)

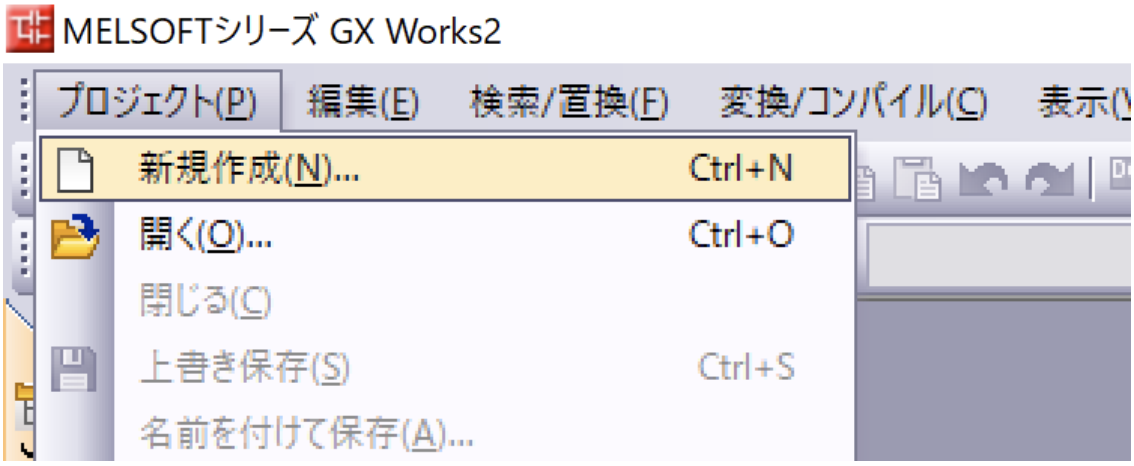
三菱電機製のPLCとはMC プロトコル(Binary)を通じて通信します。

- Q/L CPU の場合
  - PLC内蔵Ethernetの場合
  - QJ71E71-100の場合
- iQ-Rの場合
  - PLC内蔵Ethernetの場合

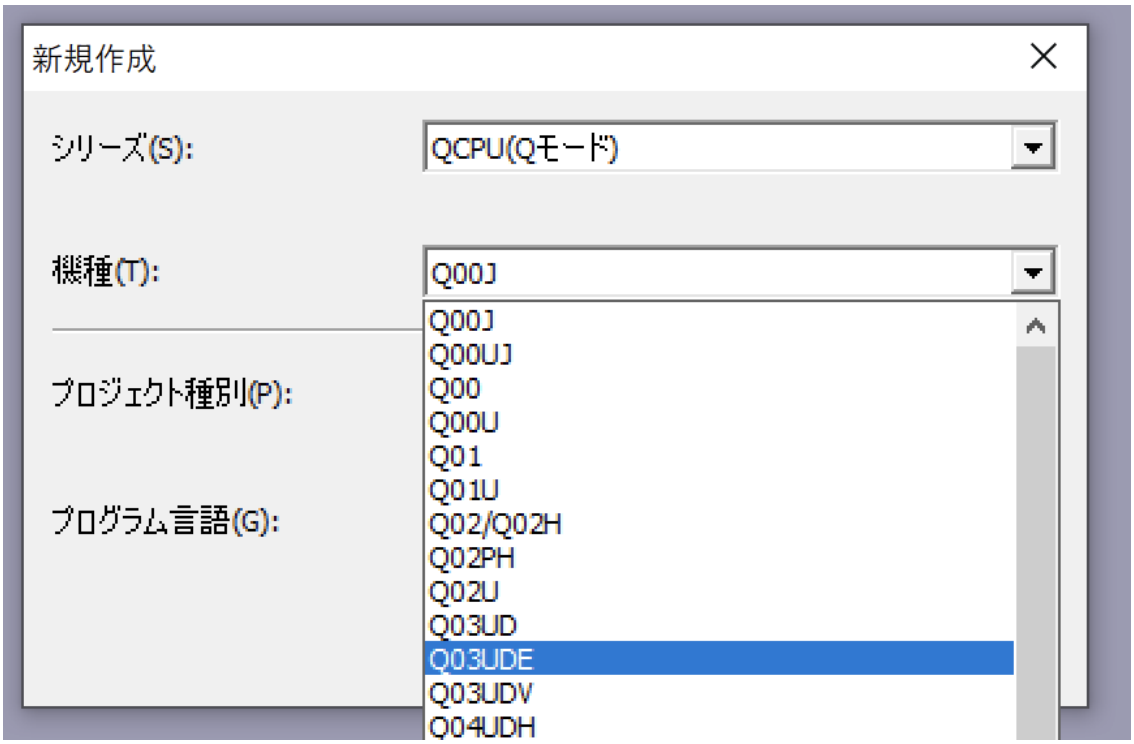
Q/L CPU の場合

PLCプロジェクト作成

- GX Works2で新規プロジェクトを作成します。
  ファイルメニューから新規作成...を選びます。

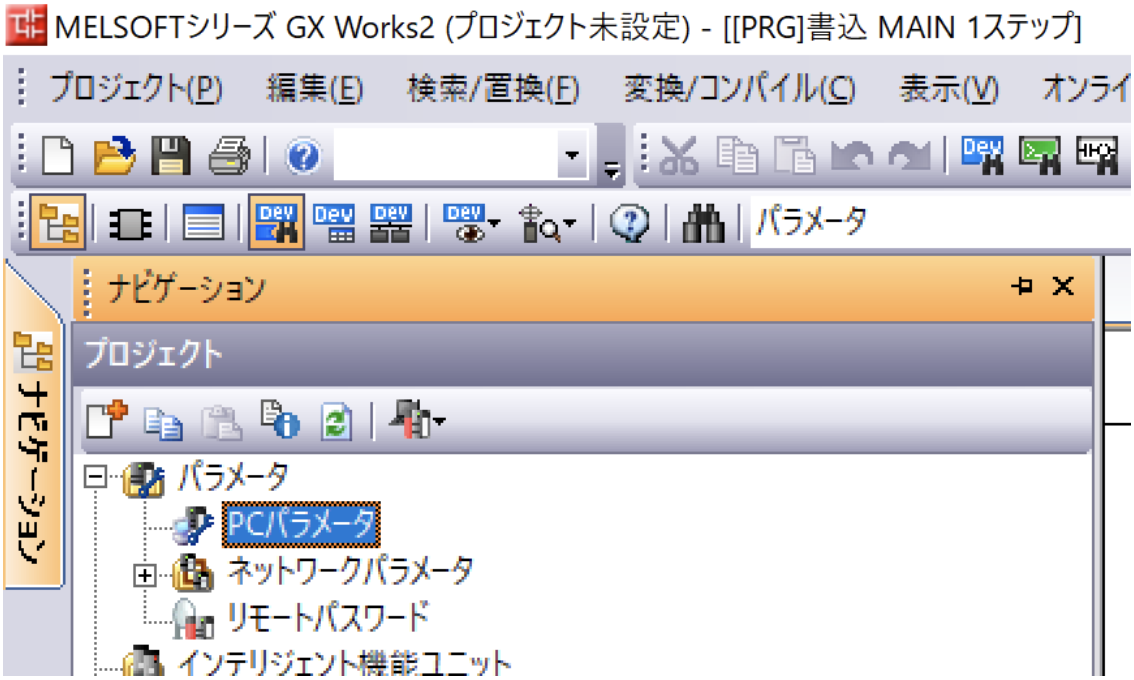


- ここではQ03UDE CPU を対象とします。
  機種からQ03UDEを選択しOKボタンを押してプロジェクトを作成します。



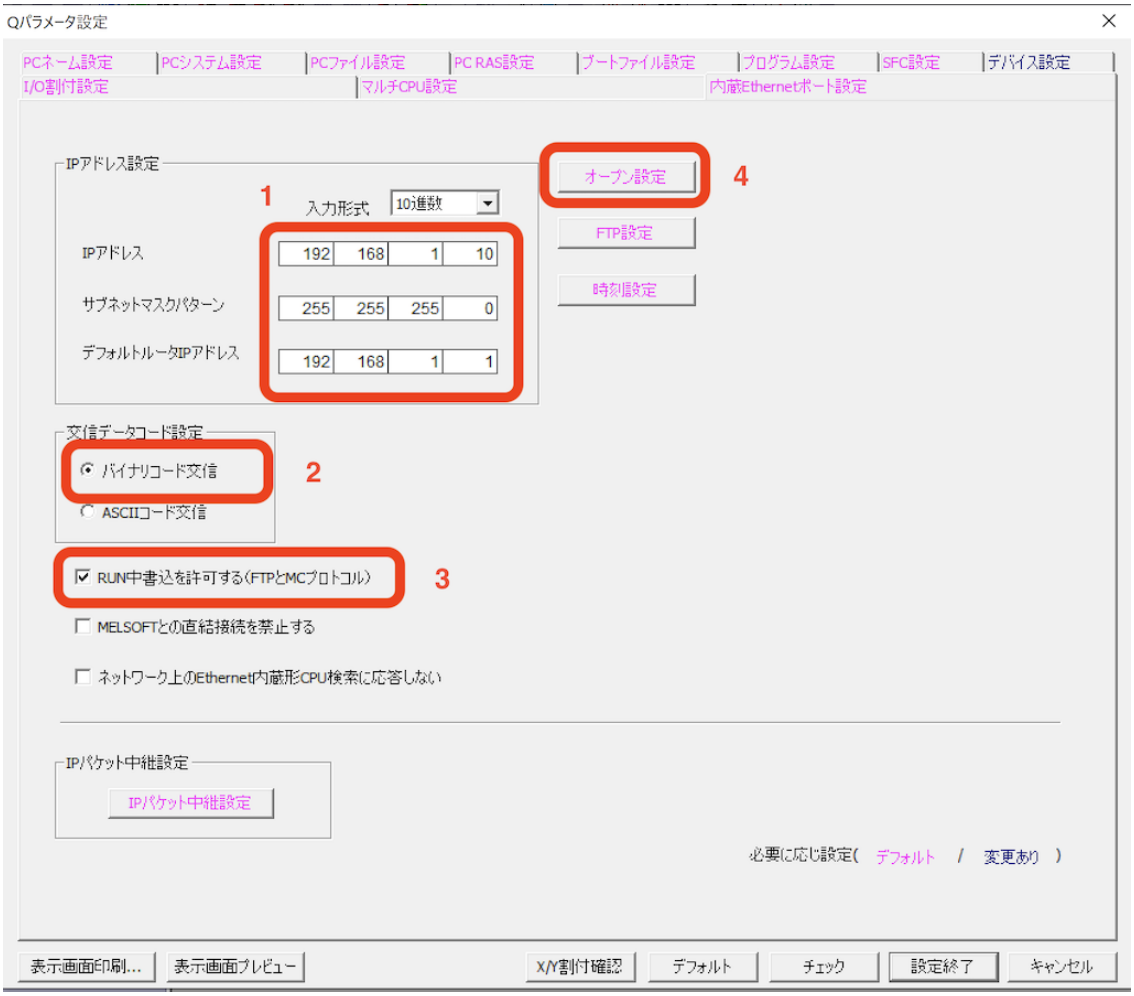
PLC内蔵Ethernetの場合

・ナビゲーションビューにあるパラメータグループのPCパラメータをダブルクリックで開きます。



1. PLCに割り当てるIPアドレス関係を設定します。
   irBoardで IPを設定 する時にここで指定したIPアドレスを指定します。
2. 交信データコード設定がバイナリコード交信になっていることを確認します。
3. RUN中書き込みを許可するにチェックを入れます。

4. オープン設定ボタンを押します。



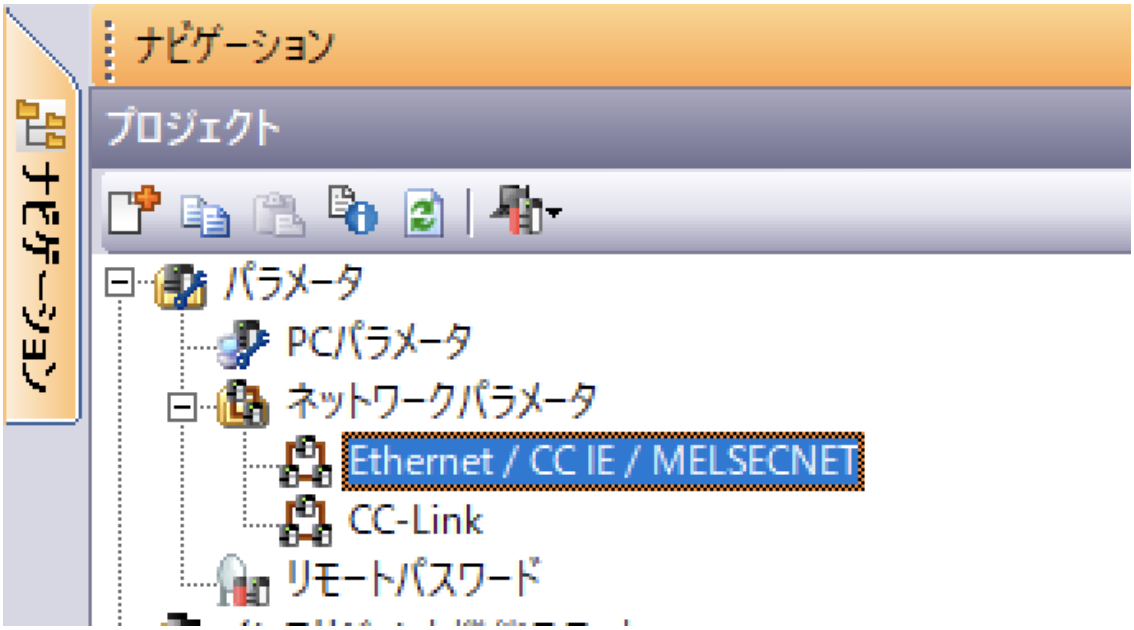
・空いている行に次の様に入力します。
  ◦プロトコル: TCP
  ◦オープン方式: MCプロトコル
  ◦自局ポート番号: 5010
  ここでは5010にしていますが、 irBoardで設定 する時にここで設定したポート番号を指定すればいいので空いている任意の番号を指定できます。
・入力が終わったら設定終了ボタンを押します。



・パラメータ設定画面でも設定終了ボタンを押して終了します。
・PLCにプロジェクトを転送し、RESETするか電源を投入し直します。

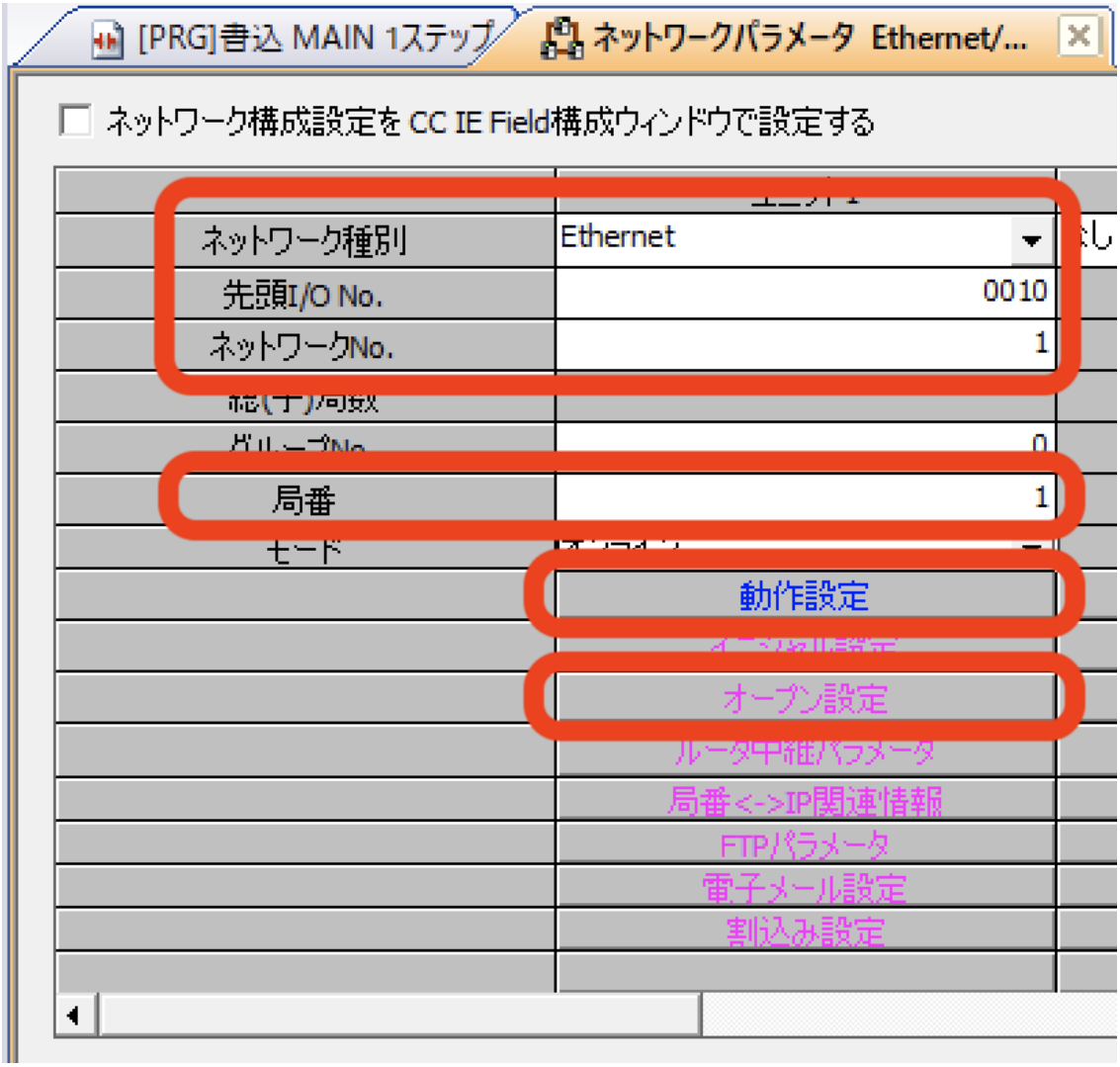
QJ71E71-100の場合

・ナビゲーションビューにあるパラメータグループのネットワークパラメータ内のEthernet/CC IE/MELSECNETをダブルクリックで開きます。

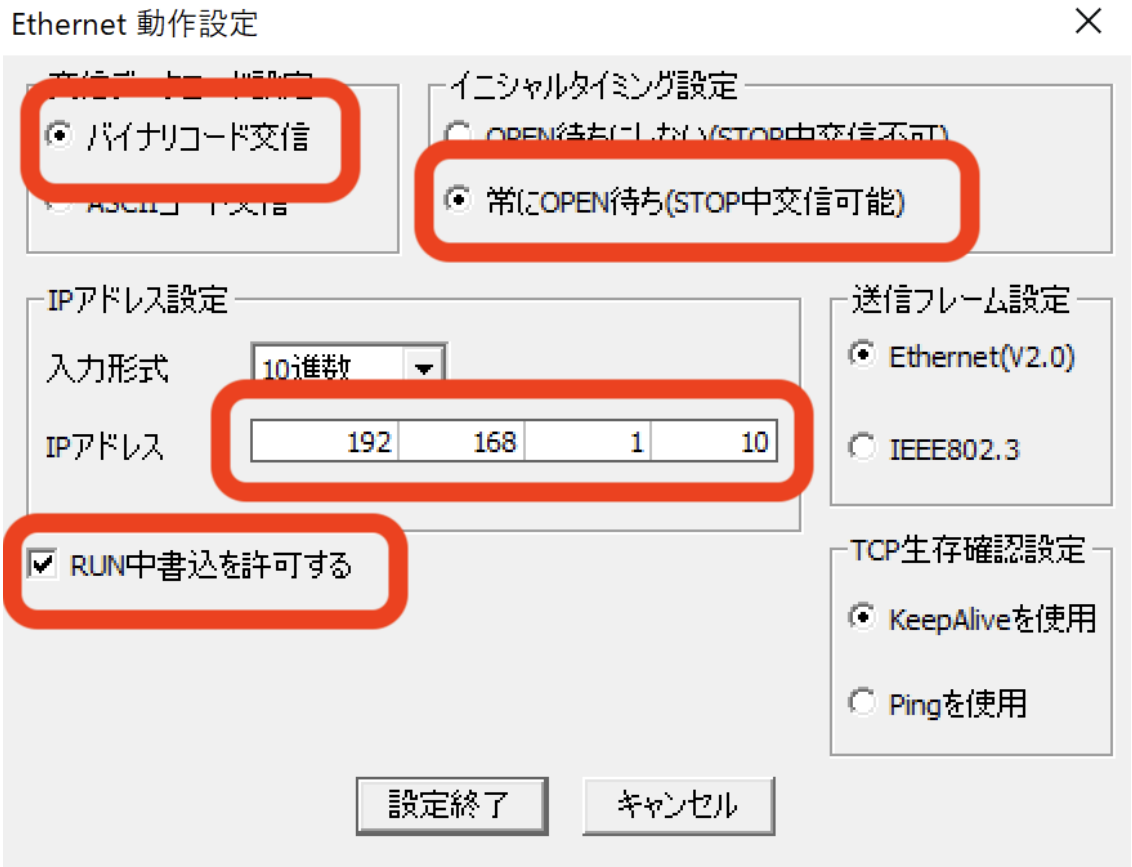


・該当ユニットのネットワーク種別でEthernetを選択します。
・先頭I/O No. にユニットの先頭I/O Noを入力します。
・ネットワーク番号を入力します。
・局番を入力します。

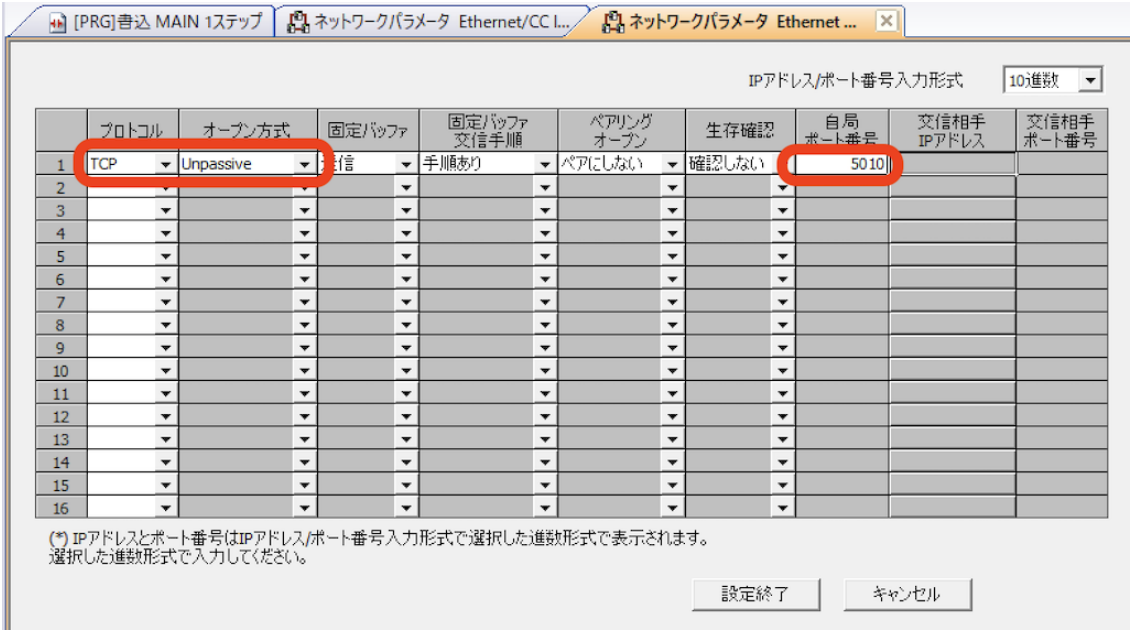
・動作設定の項目をクリックします。



・交信データコード設定がバイナリコード交信になっていることを確認します。
・イニシャルタイミング設定で 常にOPEN待ち を選択します。
・IPアドレスを設定します。
・RUN中書き込みを許可するにチェックを入れます。
・設定終了ボタンを押します。



・戻った画面で オープン設定 の項目をクリックします。゜
・空いている行に次の様に入力します。
  ◦プロトコル: TCP
  ◦オープン方式: Unpassive
  ◦自局ポート番号: 5010
  ここでは5010にしていますが、 irBoardで設定 する時にここで設定したポート番号を指定すればいいので空いている任意の番号を指定できます。
・入力が終わったら設定終了ボタンを押します。



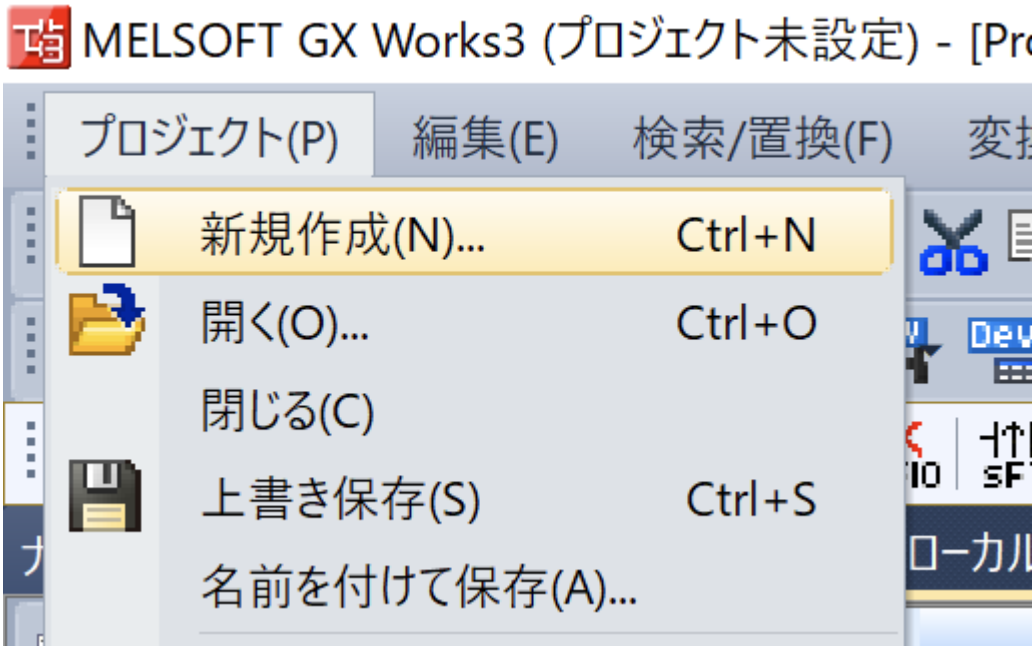
・戻った画面でも設定終了を押して設定を完了します。
・PLCにプロジェクトを転送し、RESETするか電源を投入し直します。

---

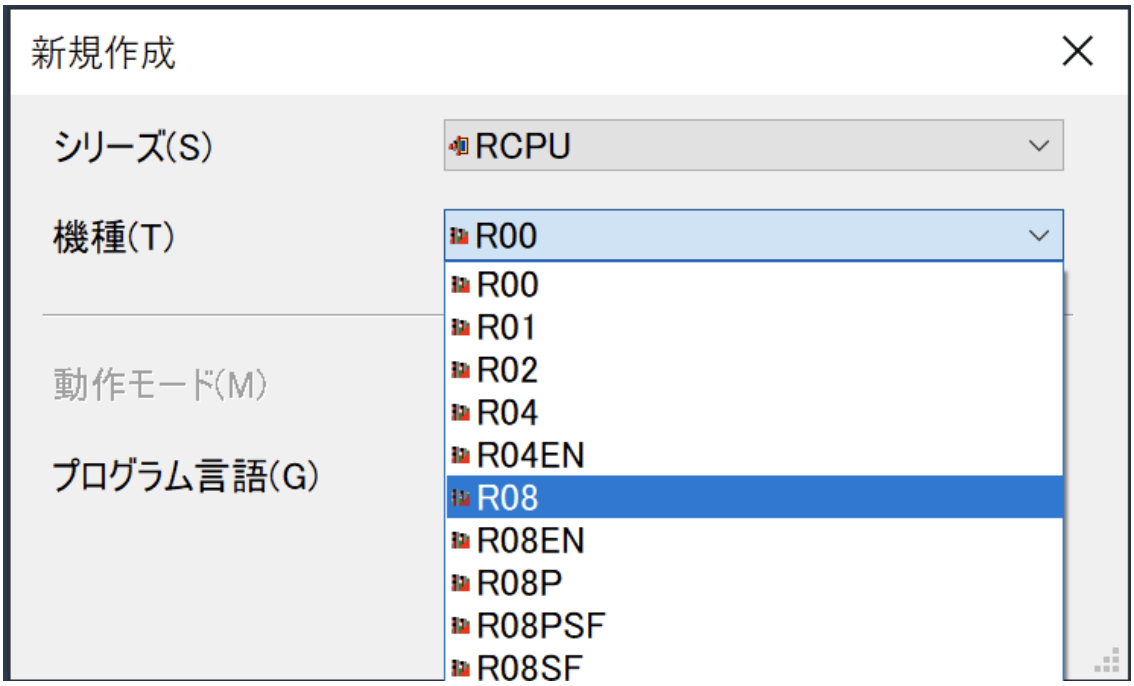
iQ-R CPU の場合

PLCプロジェクト作成

・GX Works3で新規プロジェクトを作成します。
ファイルメニューから新規作成...を選びます。

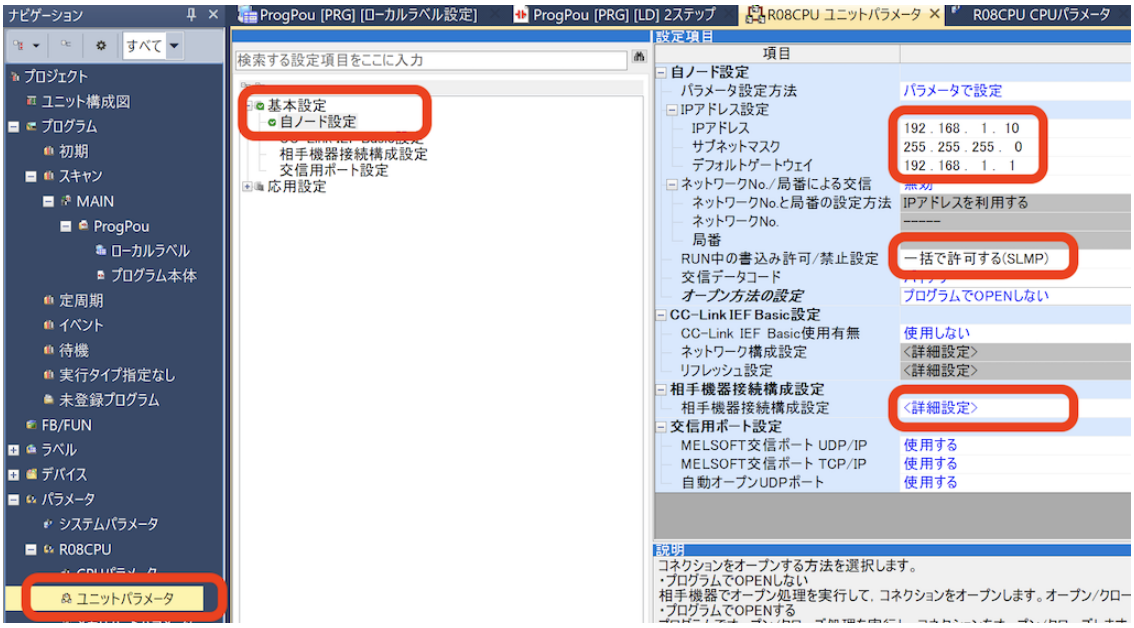


・ここではR08 CPU を対象とします。
機種からR08を選択しOKボタンを押してプロジェクトを作成します。

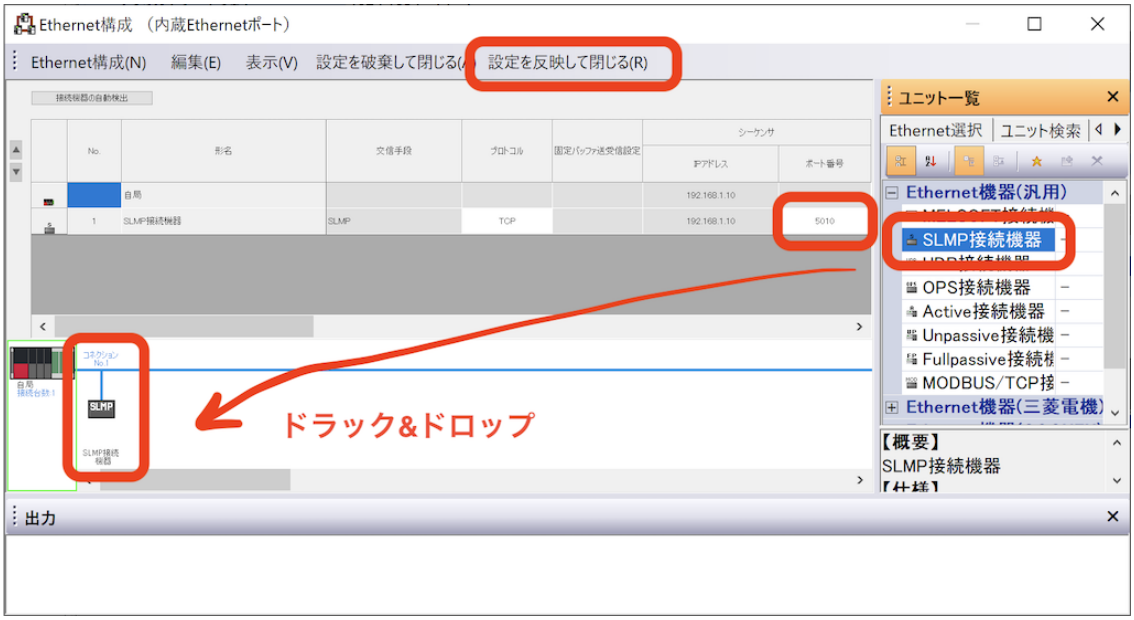


PLC内蔵Ethernetの場合

・ナビゲーションビューにあるパラメータグループを展開し、CPUグループ内のユニットパラメータをダブルクリックで開きます。
・開いた画面の左側から基本設定グループを展開し、自ノード設定を選択します。
・右側の画面のIPアドレス設定でPLCのアドレスを設定します。
・RUN中の書き込み許可/禁止設定で"一括で許可する(SLMP)"を選択します。
・相手機器接続構成設定の"<詳細設定>"をダブルクリックで開きます。



・右側のユニット一覧でEhternet機器(汎用)グループを展開し"SLMP接続機器"を画面下のPLCアイコン付近までドラッグ&ドロップします。
・上の欄に行が追加されますので、ポート番号を5010にします。 ここでは5010にしていますが、 irBoardで設定 する時にここで設定したポート番号を指定すればいいので空いている任意の番号を指定できます。
・設定を反映して閉じるを押します。



・戻った画面で適用ボタンを押します。
・PLCにプロジェクトを転送し、RESETするか電源を投入し直します。


設定項目
項目
設定
自ノード設定
パラメータ設定方法
パラメータで設定
IPアドレス設定
IPアドレス
10 . 0 . 1 . 202
サブネットマスク
255 . 255 . 255 . 0
デフォルトゲートウェイ
ネットワークNo./局番による交信
無効
ネットワークNo.と局番の設定方法
IPアドレスを利用する
ネットワークNo.
局番
RUN中の書込み許可/禁止設定
一括で許可する(SLMP)
交信データコード
バイナリ
オープン方法の設定
プログラムでOPENしない
CC-Link IEF Basic設定
CC-Link IEF Basic使用有無
使用しない
ネットワーク構成設定
<詳細設定>
リフレッシュ設定
<詳細設定>
相手機器接続構成設定
相手機器接続構成設定
<詳細設定>
交信用ポート設定
MELSOFT交信ポート UDP/IP
使用する
MELSOFT交信ポート TCP/IP
使用する
自動オープンUDPポート
使用する
説明
交信を行う相手機器に対する設定を行います。
チェック(K)
デフォルトに戻す(U
適用(A)

キーエンス: 上位リンク

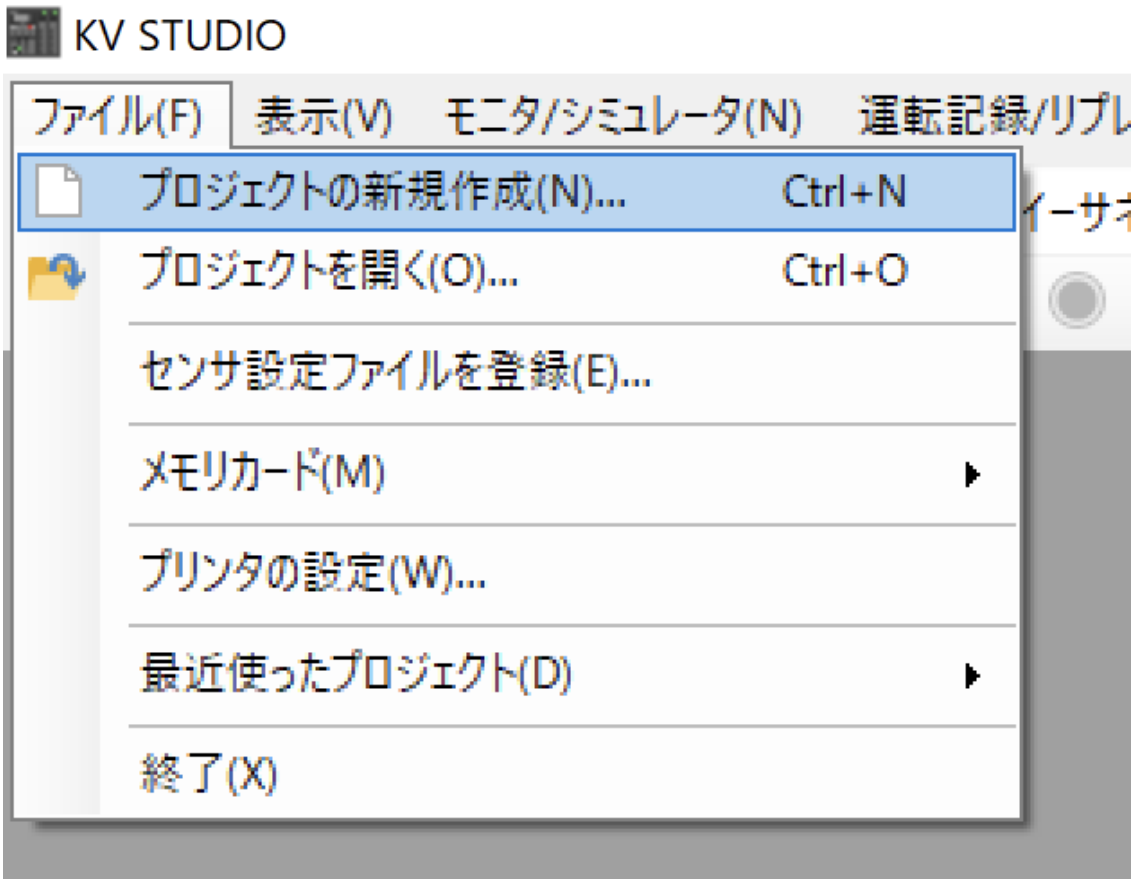
キーエンス製のPLCとは上位リンクを通じて通信します。

- KV-3000/5000/5500/7500/8000の場合
  - PLCプロジェクト作成
    - 内臓Ethernetの場合
    - EthernetモジュールKV-LE2*の場合
- KV-Nanoの場合
  - PLCプロジェクト作成
  - Ethernetユニット設定

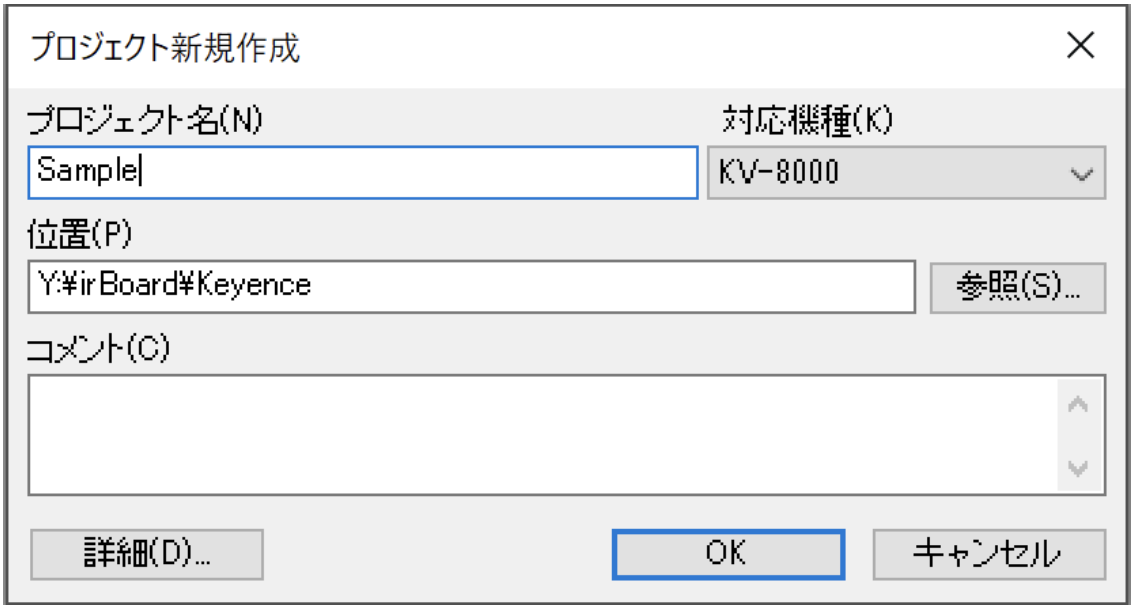
KV-3000/5000/5500/7500/8000

PLCプロジェクト作成

- KV Studioで新規プロジェクトを作成します。
  ファイルメニューからプロジェクトの新規作成...を選びます。

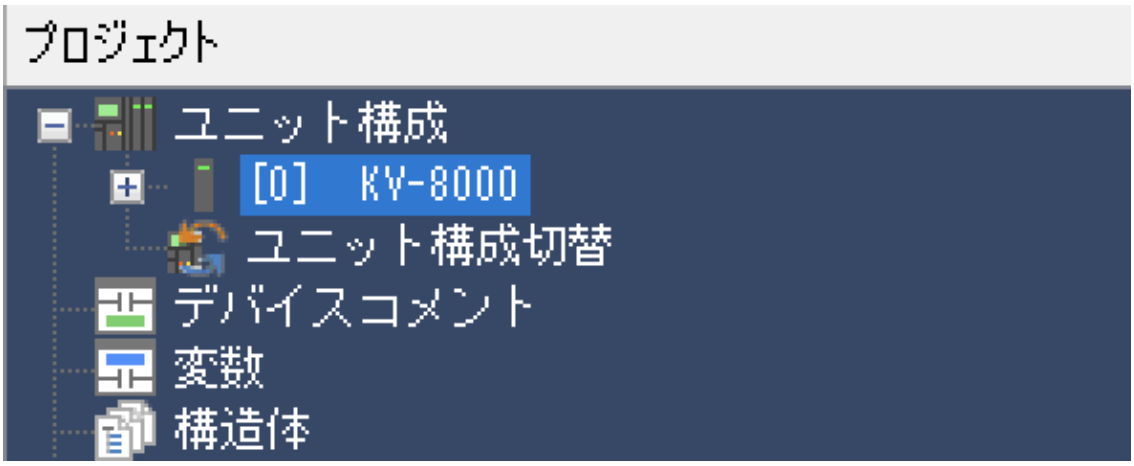


・プロジェクト名を入力し機種を選択します。ここではKV-8000を選択しています。



内臓**Ethernet**の場合

・プロジェクトビューのユニット構成からCPUユニットを選択しダブルクリックします。



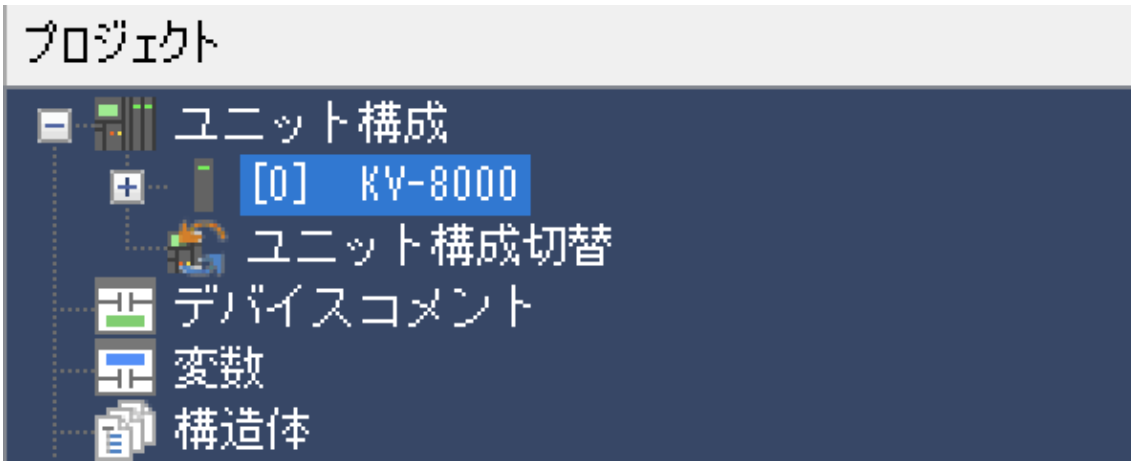
・PLCユニットを選択し、右側のユニットビューでユニット設定(2)タブを選択しIPアドレスを設定しOKまたは適用ボタンを押します。



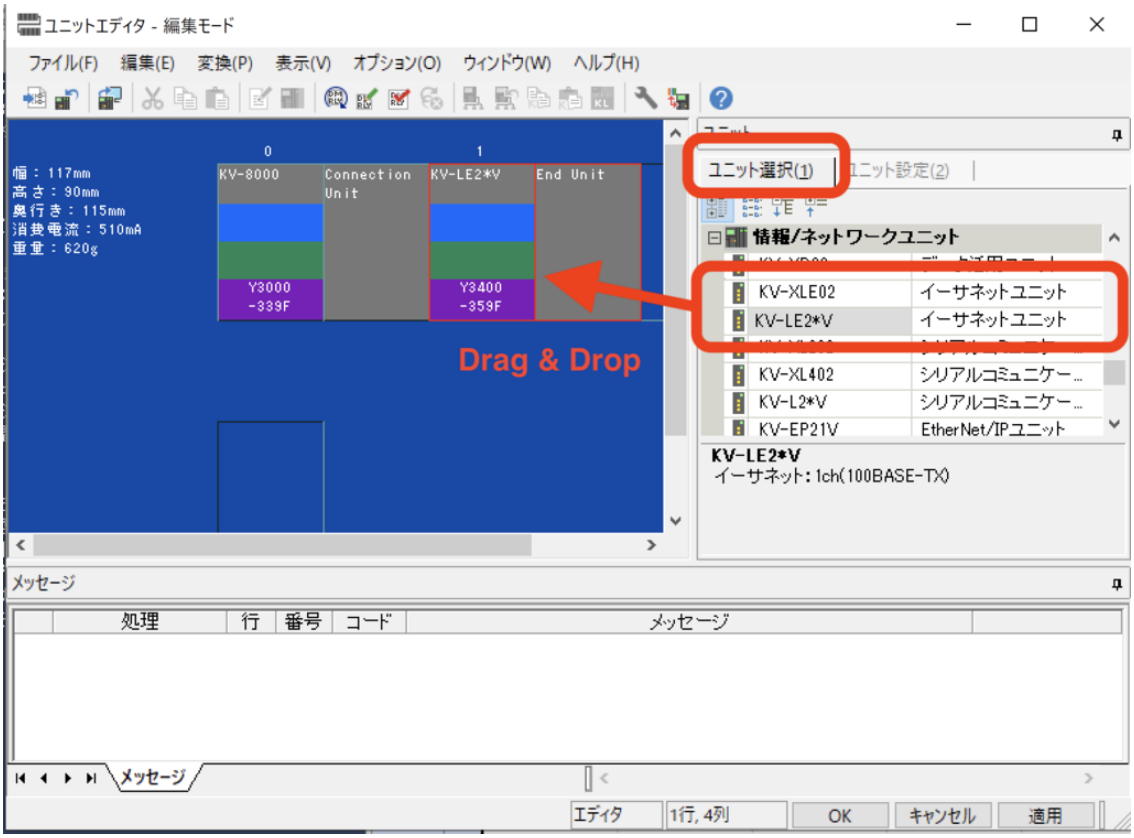
・設定が終了したらPLCにプロジェクトを転送します。

Ethernetモジュール**KV-LE2***の場合

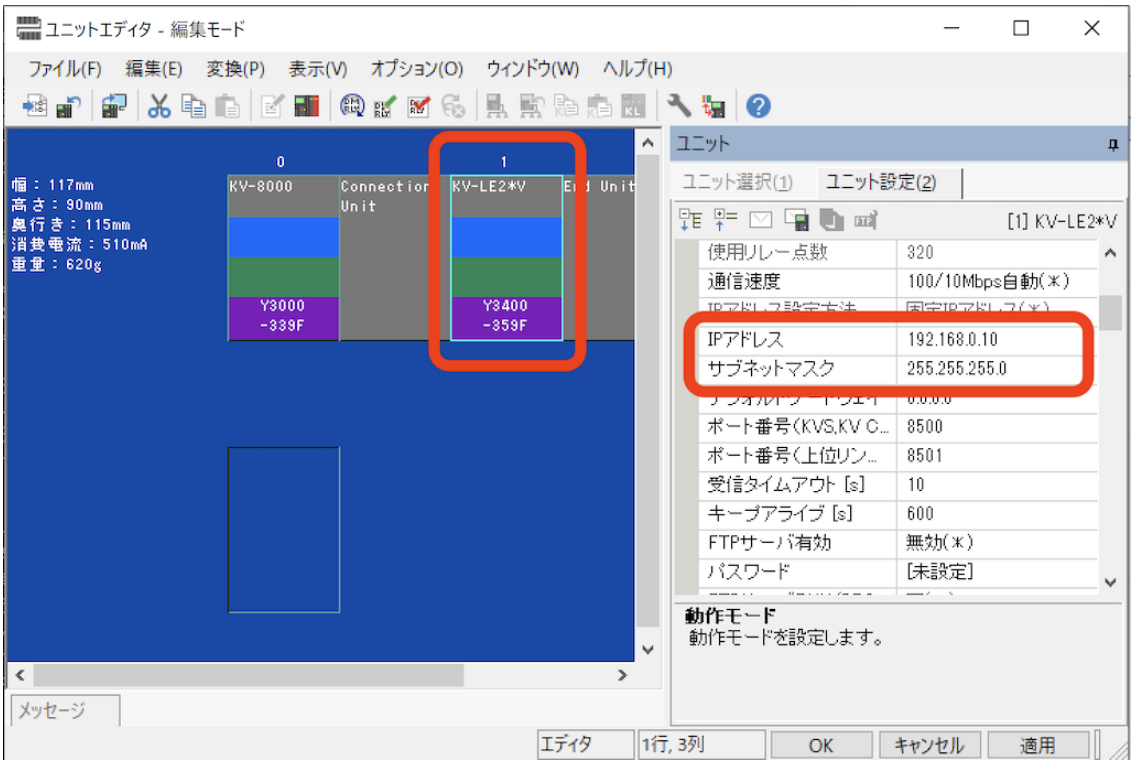
・プロジェクトビューのユニット構成からCPUユニットを選択しダブルクリックします。



・右側のユニットビューでユニット設定(1)タブを選択しネットワークユニットをCPUユニットの近くにドラックし離します。



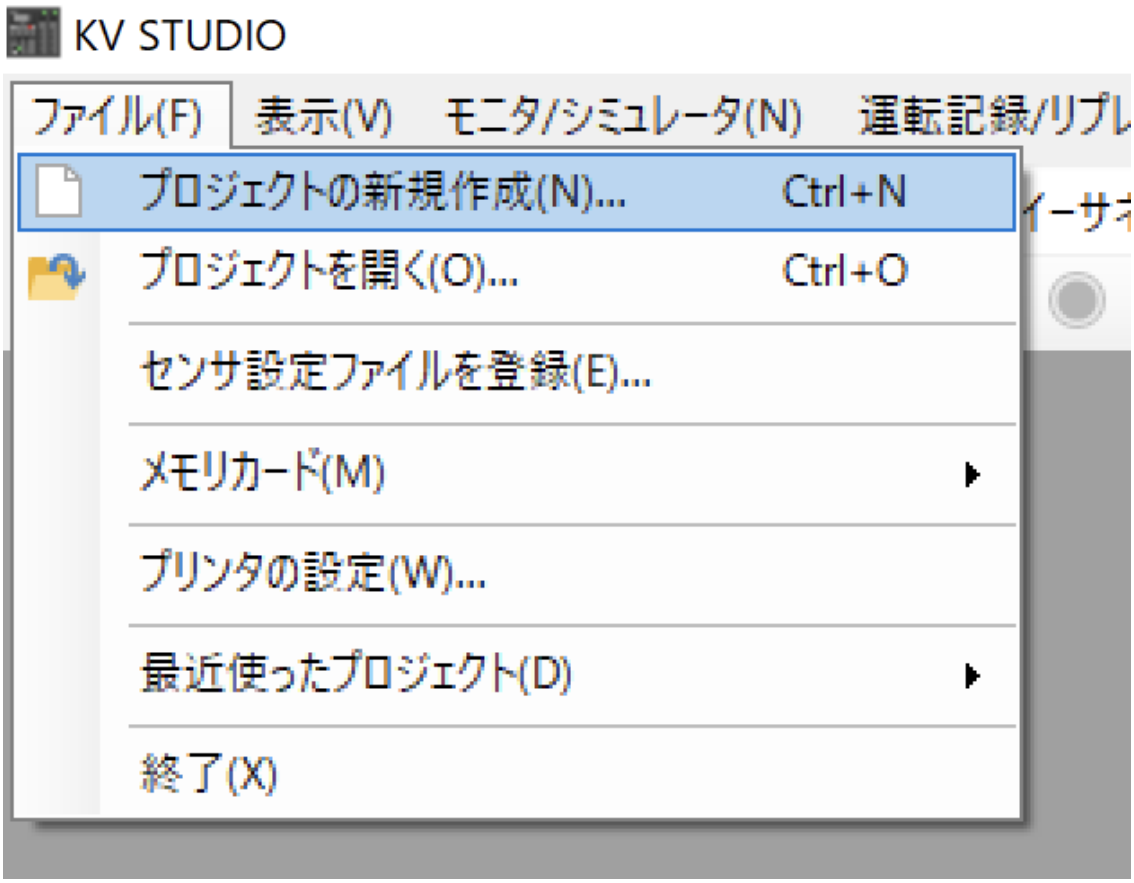
・Ethernetモジュールを選択し、右側のユニットビューでユニット設定(2)タブを選択しIPアドレスを設定しOKまたは適用ボタンを押します。



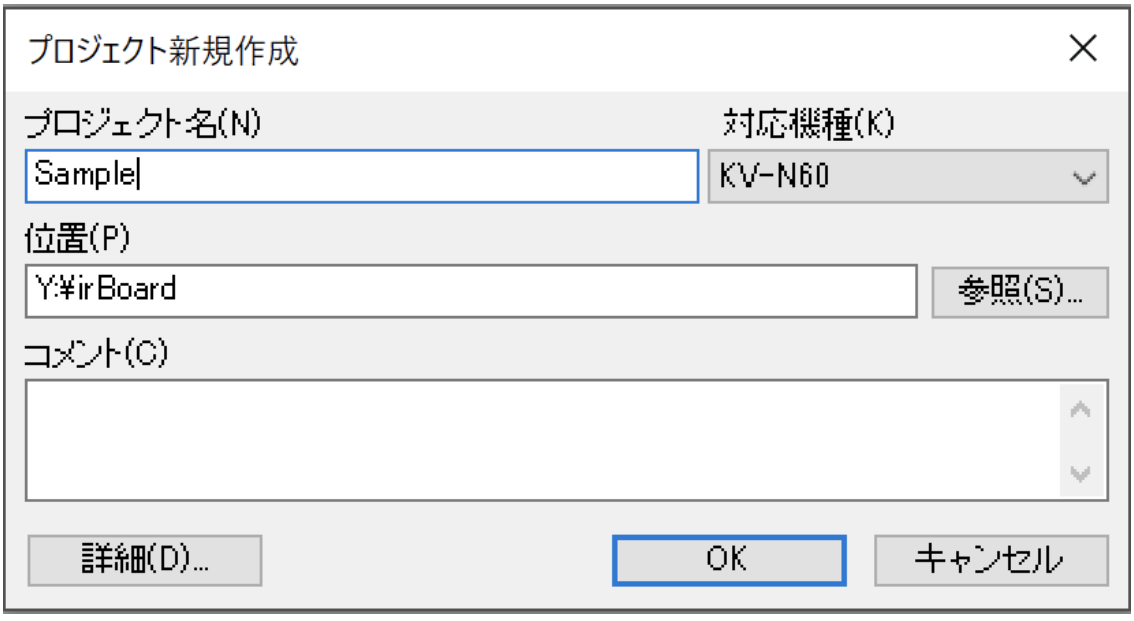
KV-Nano

PLCプロジェクト作成

・KV Studioで新規プロジェクトを作成します。
　ファイルメニューからプロジェクトの新規作成...を選びます。

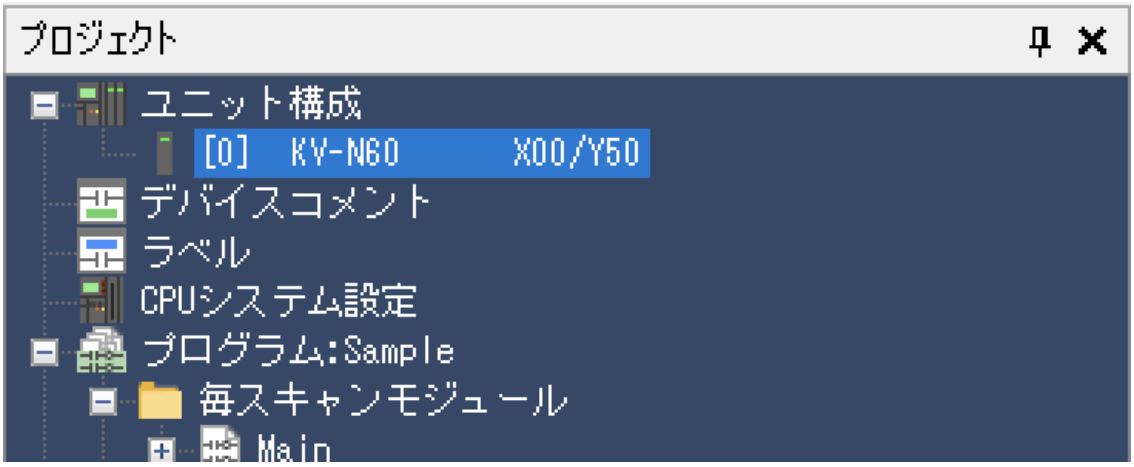


・プロジェクト名を入力し機種を選択します。ここではKV-N60を選択しています。

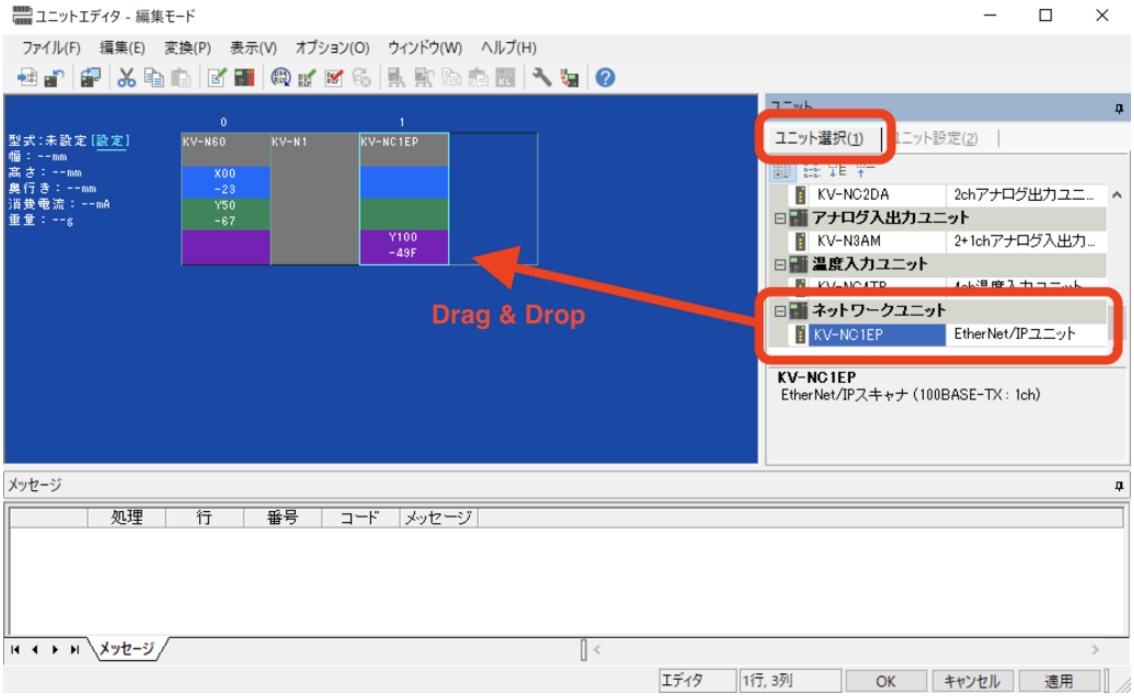


Ethernetユニット設定

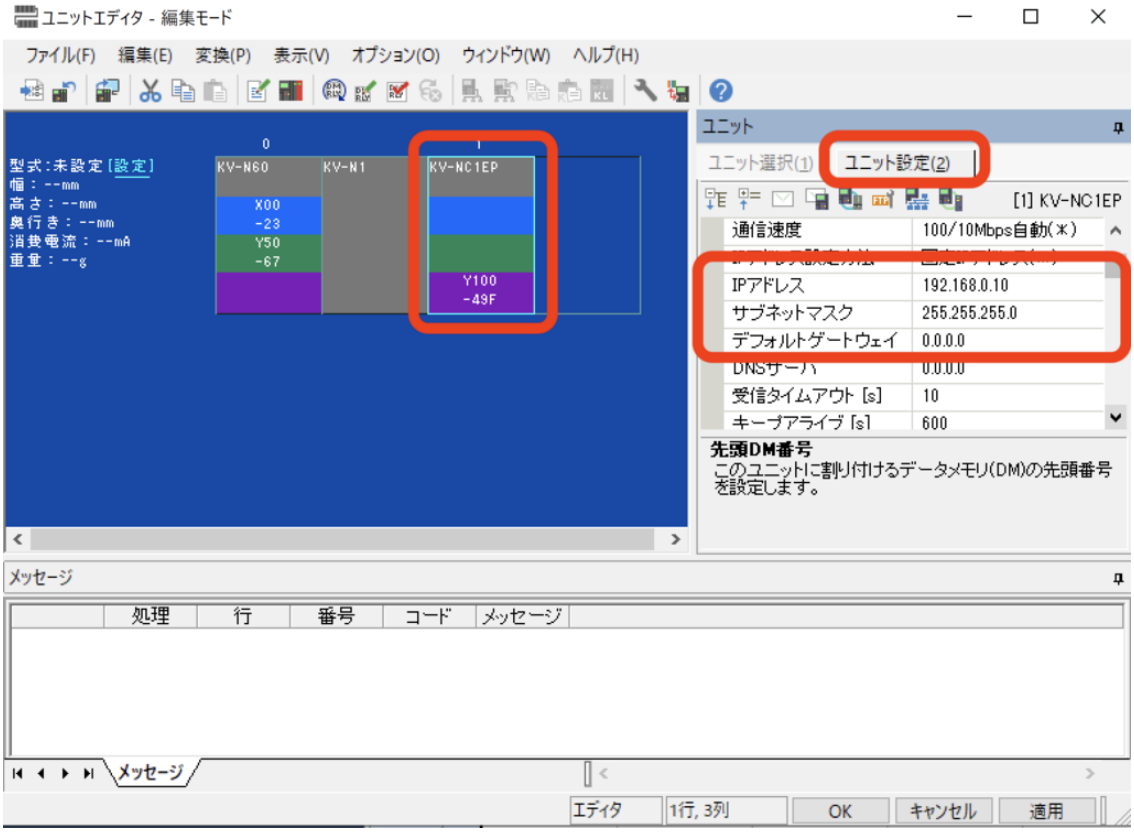
・プロジェクトビューのユニット構成からCPUユニットを選択しダブルクリックします。



・右側のユニットビューでユニット設定(1)タブを選択しネットワークユニットをCPUユニットの近くにドラックし離します。



・Ethernetモジュールを選択し、右側のユニットビューでユニット設定(2)タブを選択しIPアドレスを設定しOKまたは適用ボタンを押します。



・設定が終了したらPLCにプロジェクトを転送します。

オムロン: FINS/TCP

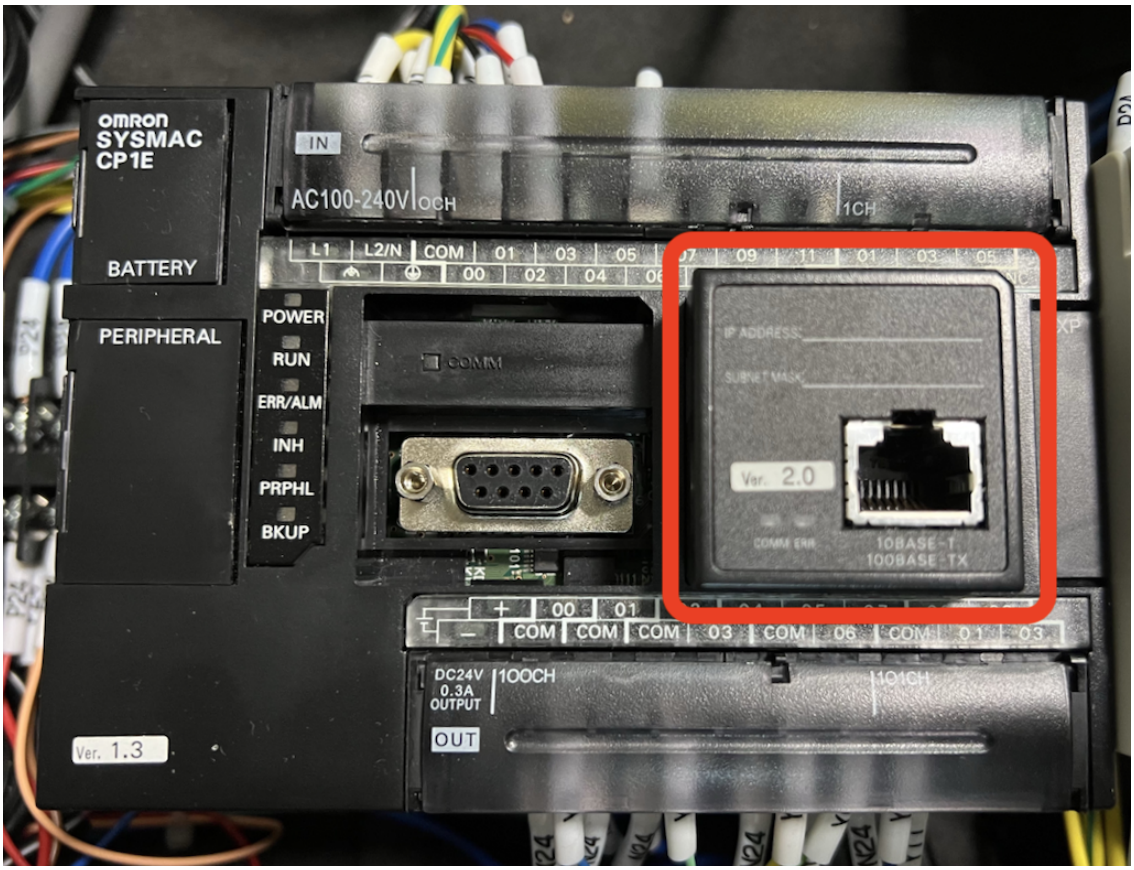
オムロン製のPLCとFINS/TCPを通じて通信します。

- CP1Eシリーズの場合
  - Ethernet オプションボード
  - 通信設定
  - ネットワーク設定
  - オプションボードの初期化
- CJ2Mシリーズの場合
  - PLC内蔵Ethernetの場合

CP1Eシリーズの場合

Ethernet オプションボード

- CP1Eシリーズの N/NA □□ CPU ユニットタイプ でirBoardと接続するにはEthernet オプションボード(形 CP1W-CIF41)が必要です。
  ※) N/NA □□ S CPU ユニットタイプ にはEthernet オプションボードはつきませんので注意してください。



通信設定

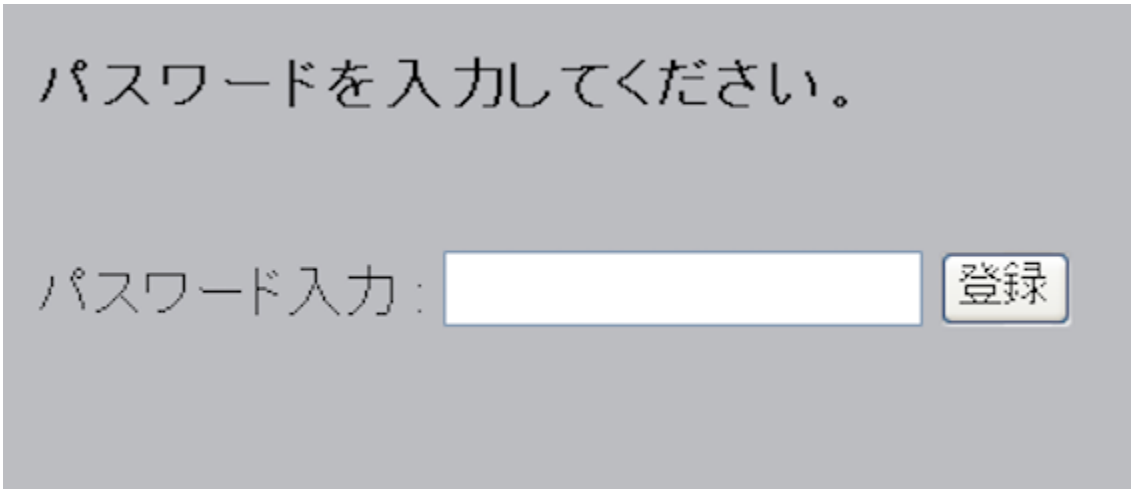
- PLCシステム設定の シリアルオプションボード タブで以下を設定します。
  - ユーザー設定をチェック
  - 通信速度で 115200 を選択
  - パラメータで 7,2,E を選択
  - モードで 上位リンク(デフォルト) を選択
  - 号機No.で 0 を選択
  - 設定したらPLCに転送後PLCの電源を再投入します。



ネットワーク設定

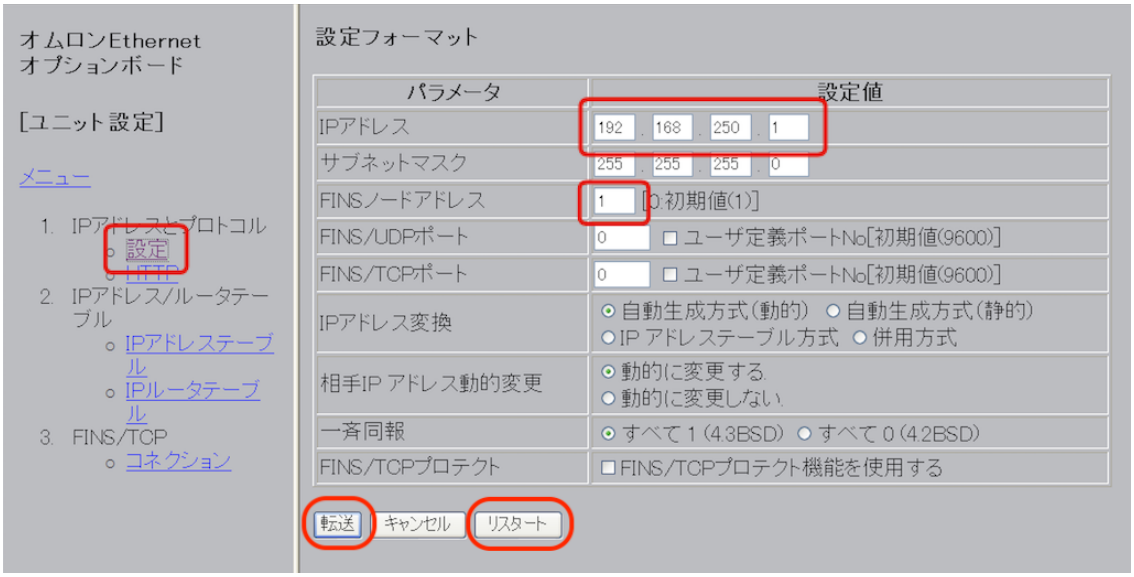
ネットワーク設定はWebブラウザーから行います。

- Ethernetケーブルを接続しPCと接続できる様に構成します。
- 購入直後Ethernet オプションボードは 192.168.250.1 になっています。PCのネットワーク設定でこのアドレスと接続できる様に設定します。
- Webブラウザーで http://192.168.250.1/J00.htm にアクセスします。
  パスワードの入力を求められますので ETHERNET を入力し登録ボタンを押します。



- 表示されたページ左側メニューの 1. IPアドレスとプロトコルから 設定 を選択します。

・IPアドレスに設定したいアドレスを設定します。 FINSノードアドレスにはIPアドレスの末尾(図の場合は1)を設定します。 必要があればサブネットマスクを設定します。
・設定後、転送ボタンを押して反映させます。
・IPアドレスを更新させるためにリスタートボタンを押します。
・設定後Webブラウザーでアクセスする場合は http://192.168.250.1/J00.htm のIPアドレスを設定したIPアドレスに置き換えてください。



Ethernet オプションボードの初期化

設定したIPアドレスがわからなくなった場合に初期化をするとIPアドレスを
192.168.250.1 に戻す事ができます。
初期化するには次の手順で初期化します。

・A525.01 をONにすると初期化を開始します。
・A525.09 をONにしEthernet オプションボートをリスタートします。
・リスタートが完了すると A525.09 がOFFになります。

CJ2Mシリーズの場合

PLC内蔵Ethernetの場合

(CJ2M CPU31、IPアドレス 192.168.250.2 の場合)



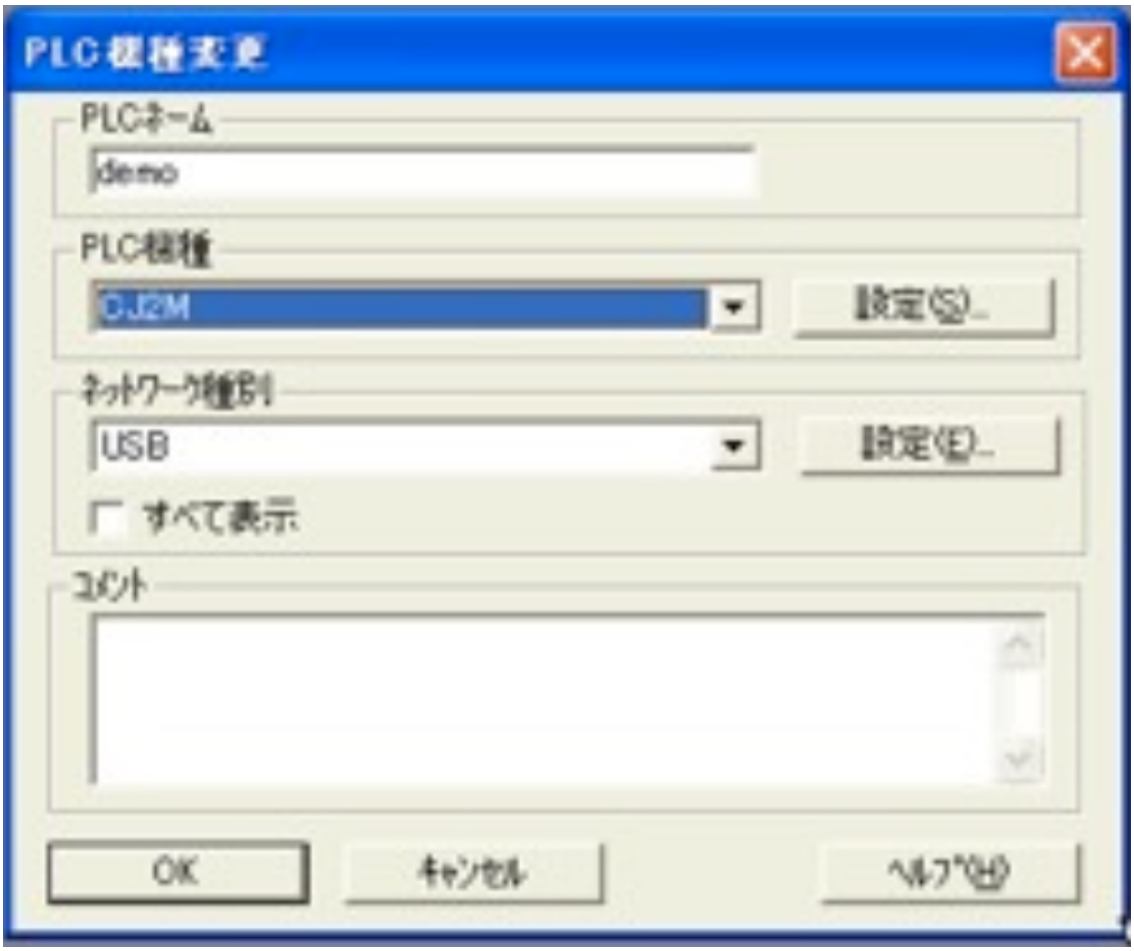
・電源を入れる前にPLCのノード番号を設定します。
ノード番号はIPアドレスの末尾の番号を指定します。192.168.250.2の場合は最後の2がノード番号になります。設定する際16進数に変換しPLCのノード設定デジスイッチで設定します。
・設定後電源を投入します。



・CX Programmerを立ち上げて、新規プロジェクトを作成します。



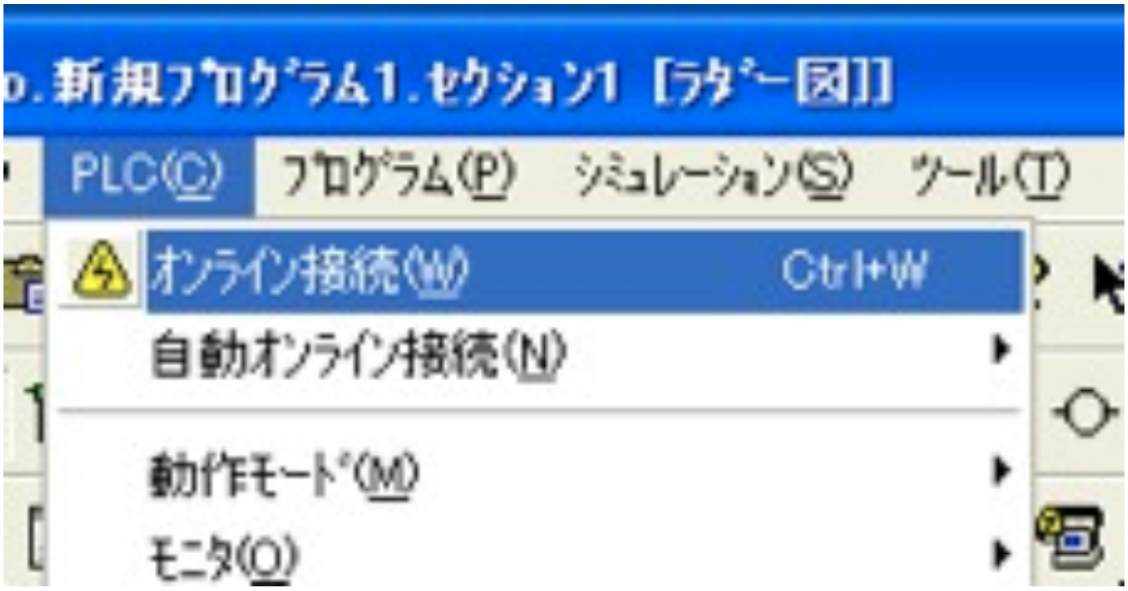
・PLC機種でCJ2Mを選択します。



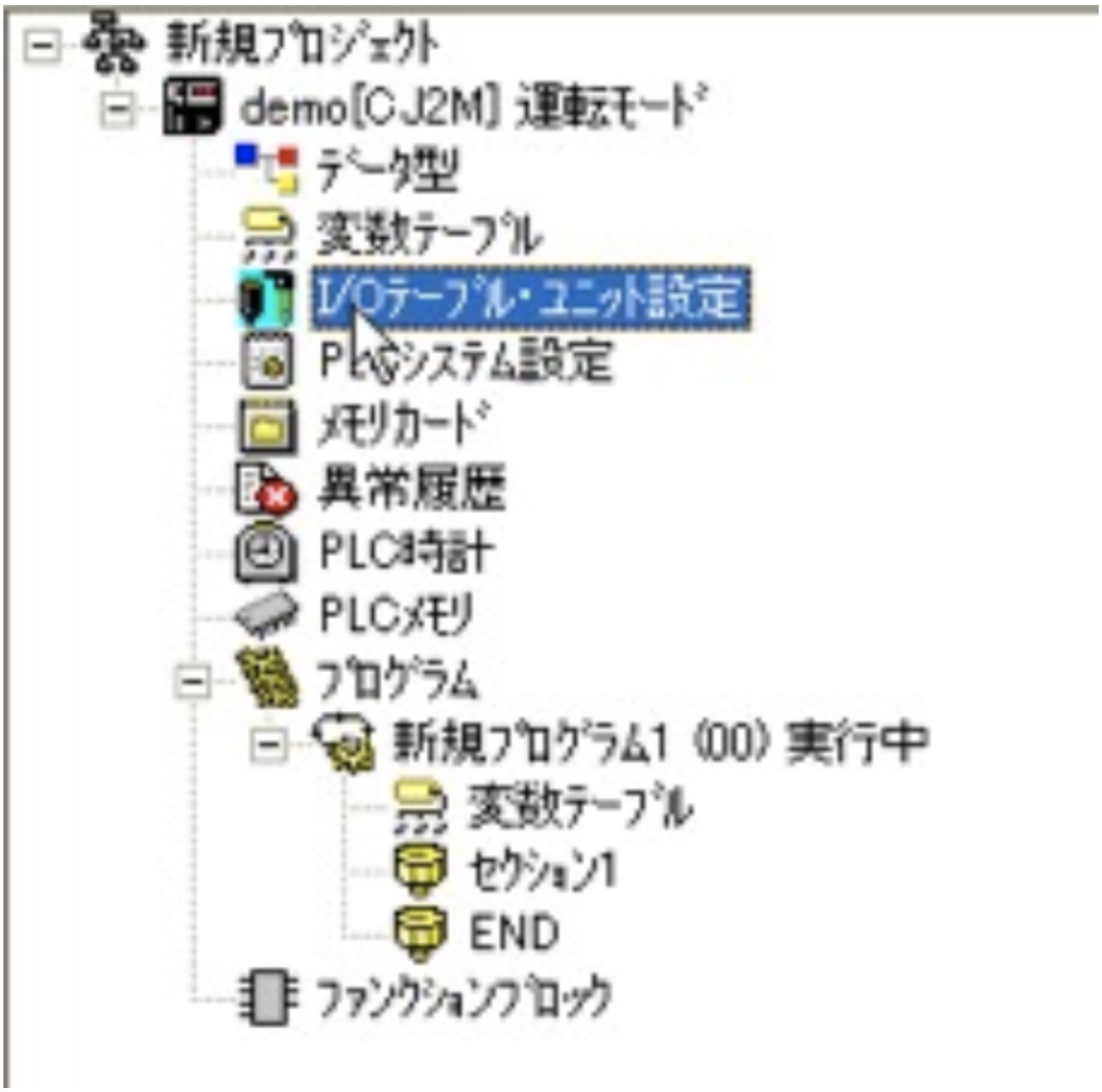
・設定ボタンを押して、表示されたダイアログでCPU31を選択しOKボタンを押します。PLC機種変更ダイアログでもOKを押してダイアログを閉じます。



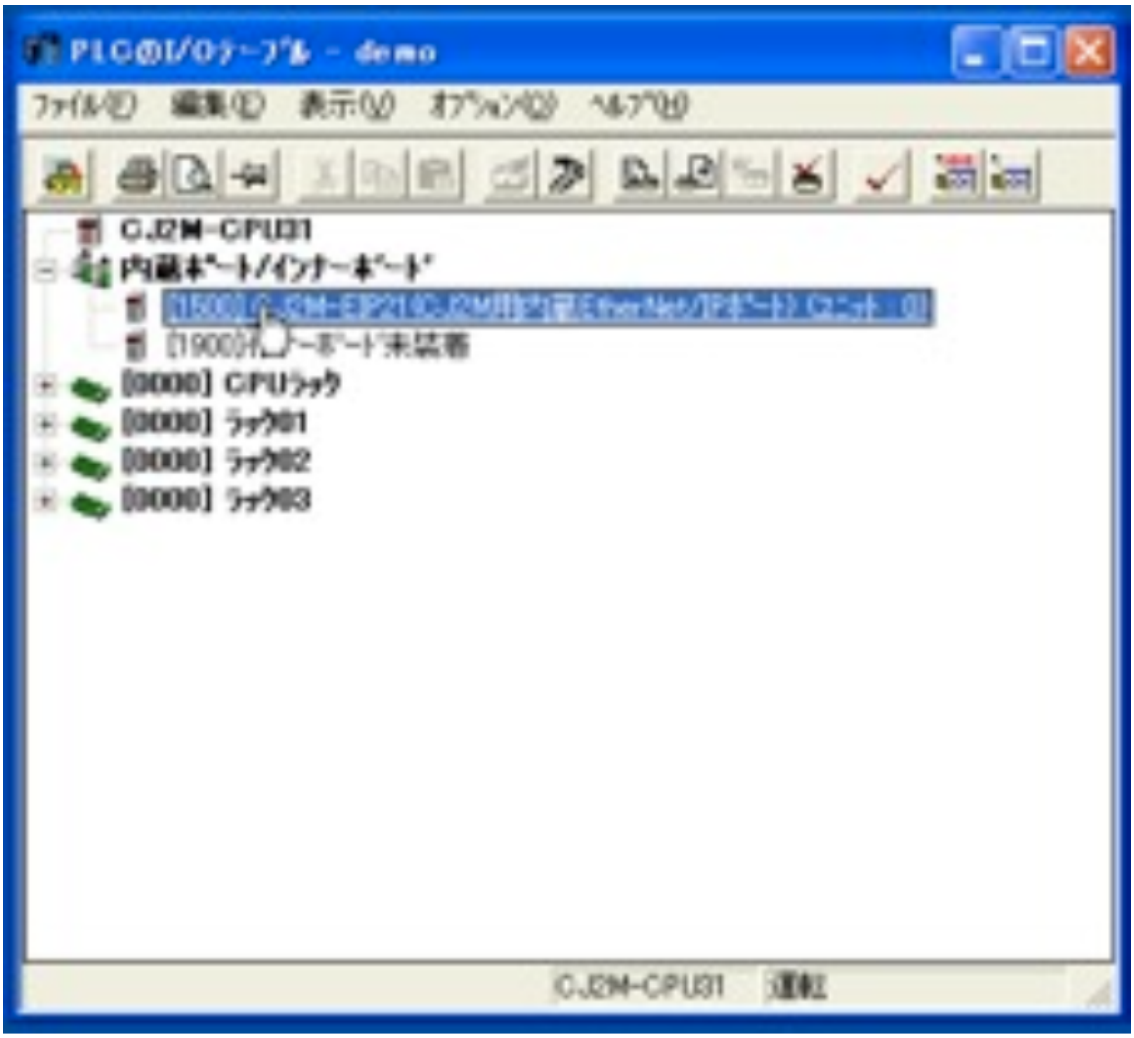
・続いてPLCメニューのオンライン接続を選択し、PLCと接続します。



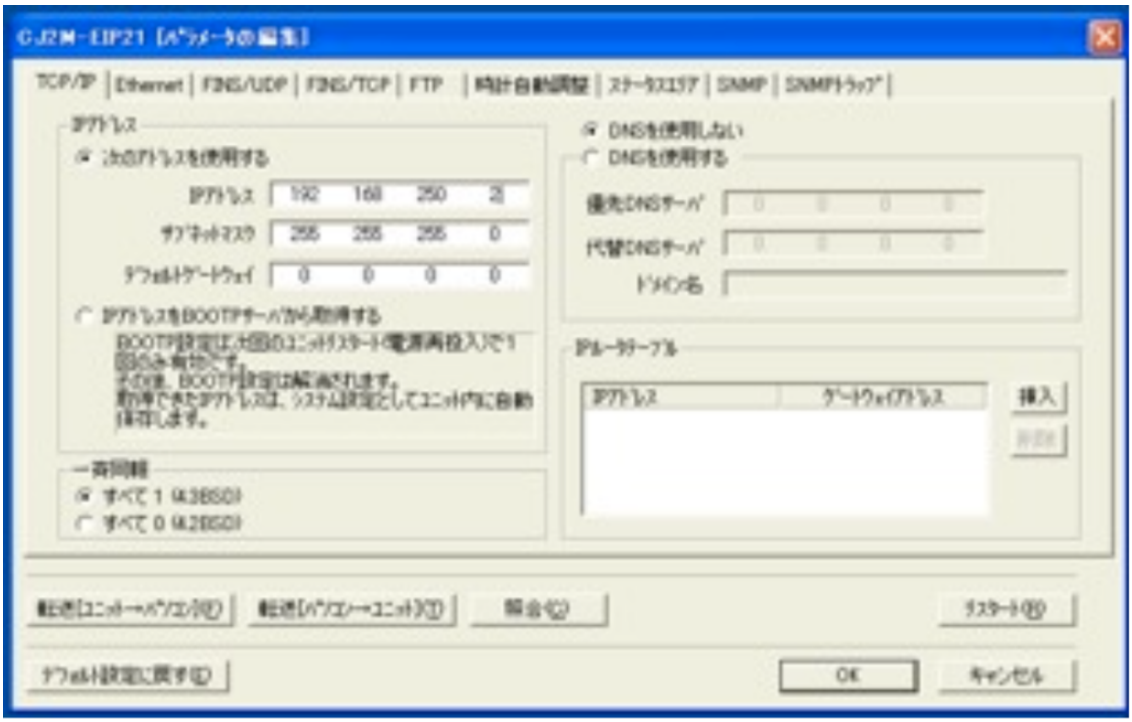
・左のツリービューの一覧から、I/Oデーブル・ユニット設定を開きます。



・内蔵ポート/インナーボードのCJ2M-EIP21(JC2M用内蔵EtherNet/IPポート)(ユ ニット:0)を開きます。



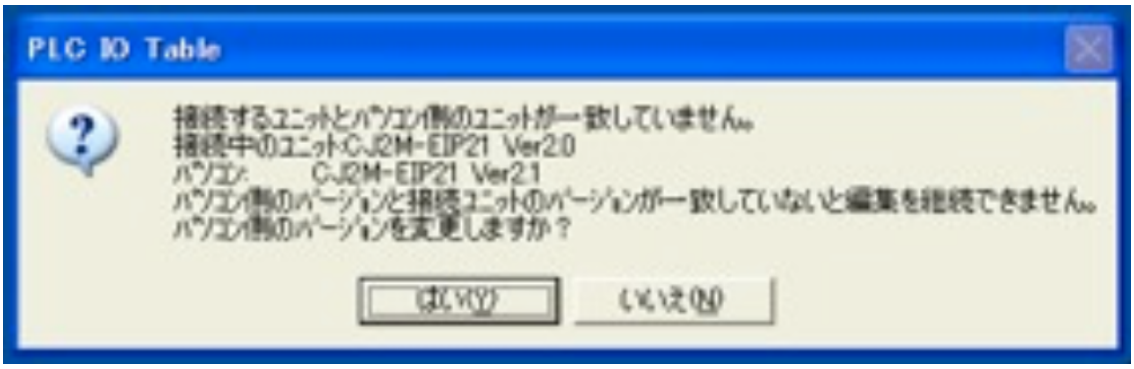
・TCP/IPタブを選択し、IPアドレスに192.168.250.2を、サブネットマスクに255.255.255.0を入力しOKボタンを押しダイアログを閉じます。
irBoardで IPを設定 する時にここで指定したIPアドレスを指定します。



・オプションメニューから転送[パソコン->PLC](T)を選択設定をPLCに転送します。



・途中でバージョン不一致のアラートが出た場合は、はいを押して変更して下さい。



横河電機: 上位リンク

横河電機製のPLCと上位リンクを通じて通信します。

- FA-M3"V"シリーズの場合
  - PLC内蔵Ethernetの場合
  - Ethernetインタフェースモジュール LE-11 の場合

FA-M3"V"シリーズの場合

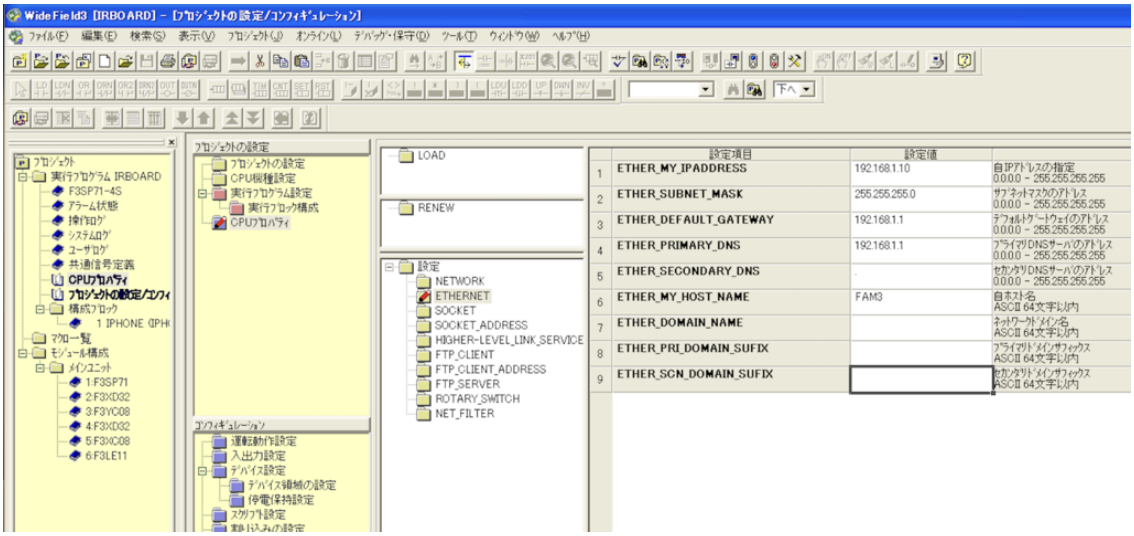


PLC内蔵Ethernetの場合

F3SP71-4S IPアドレス 192.168.1.10 の場合

- WideField3でPLCと接続します。
- CPUプロパティのETHERNT設定でIPアドレスを設定します。
  irBoardで IPを設定 する時にここで指定したIPアドレスを指定します。
- OKボタンを押してファイルに保存します。
- メニューから オンライン -> ダウンロード[PC-CPU] -> CPUプロパティ を選択し、
  保存したファイルをPLCにダウンロードします。

・電源を再投入します。



Ethernetインタフェースモジュール **LE-11** の場合

F3SP58-6S + LE11-OT、IPアドレス 192.168.1.10 の場合

・PLCの電源を入れる前にIPアドレスを設定します。
・IPアドレスはLE11-0Tモジュールのの側面にあるロータリースイッチで設定します。 192.168.1.10を16進数で表し C0 A8 01 0A を設定します。
irBoardで IPを設定 する時にここで指定したIPアドレスを指定します。
・設定したら電源を入れます。
・WideField2のプロジェクトでは特別な設定は必要有りません。



C A 0 0
0 8 1 A

MODBUS: MODBUS/TCP

MODBUS/TCPに対応しているPLCや機器はMODBUS/TCPを通じて通信します。

通信設定

通信設定についてはご使用の機器の取説をご覧ください。
基本的にはMODBUS/TCPでの通信を有効にし、IPアドレスを設定する事で接続できる様になります。
温調器やセンサー、I/Oなどの機器と接続する場合はirBoardの プロジェクト設定の相互通信を使用しない設定 にする事で接続できる様になります。

デバイスの指定

デバイスを設定する場合は、デバイスの種類によって次のサフィックスを付けます。

タイプ	プリフィックス	例
コイルデバイス	M	M0
入力デバイス	I	I0
レジスター	HR	HR0
入力レジスター	IR	IR0

ファイルデバイスは対応していません。

ローカライズ(多言語化)

ver 2.0以降テキストデータのローカライズ(多言語化)ができる様になりました。

言語切り替え

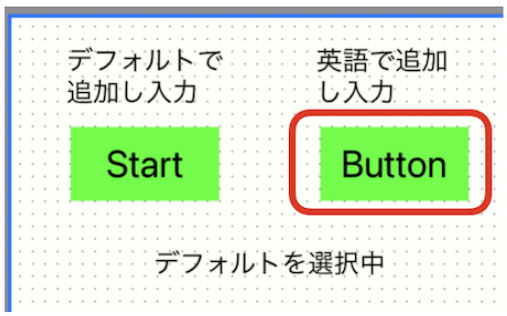
言語を切り替えるには上部ツールバーの地球アイコンから選択します。
リセット項目の上には最近使用した言語が表示されます。
言語を追加するにはリセット項目下の行から追加します。
リセットを選択すると最近使用した言語の項目が削除されます。

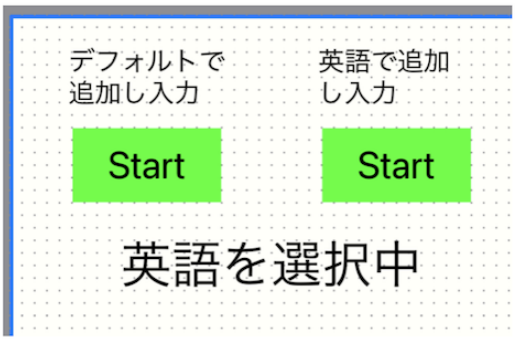
※ 編集画面と実行画面ではギヤアイコンから選択できます。

主となる言語はデフォルトを選択して入力してください。
追加する言語は言語を切り替えてから入力します。

例) 主言語を日本語で第二言語を英語にする場合

- デフォルトを選択して日本語で入力します。
- 英語を入力する場合は英語を選択してから入力します。
- 注意事項
    - デフォルトを選択した状態で部品を追加すると他の言語を選択した時にデフォルトで入力した文字が転用されます。
    - 英語を選択した場合は最初デフォルトで入力した文字が表示されます。
    - デフォルト以外(この場合は英語)を選択した状態で部品を追加するとデフォルトに転用されません。英数字など共通して使用できる文字の場合にデフォルトにすると再度入力する必要があります。
    - 部品を追加する場合はデフォルトで追加し文字入力する方が二度手間になりません。







翻訳管理（Localization Manager）

右サイドバーの「翻訳管理」タブで、プロジェクト内のすべてのテキストを一覧表示し、まとめて翻訳を管理できます。

基本機能

- スコープ選択: ページ単位または選択中のアイテムのみ表示
- ロケール選択: 翻訳する言語を選択（デフォルト = ベーステキスト）
- ロケール追加: +ボタンで新しい言語を追加
- 進捗表示: 翻訳済み項目数とパーセンテージを表示
- 検索・フィルター: テキスト検索と「未翻訳のみ」フィルター
- アイテムへジャンプ: 行の右端の>ボタンでアイテムに移動

翻訳の入力

各翻訳フィールドをタップして入力します。
フォーカスが外れると自動保存されます。

プロジェクトインポート時のロケール自動登録

.irboardファイルをインポートすると、プロジェクト内で使用されている言語が自動的に登録されます。
チュートリアルプロジェクトや他のユーザーから受け取ったプロジェクトの翻訳をすぐに確認できます。

翻訳可能なテキスト

対象	フィールド	説明
ページアイテム	タイトル	ラベル、ボタン、値フィールド、メーター等の表示テキスト
	代替タイトル	ランプデバイス有効時の代替テキスト
	単位	値フィールド、メーターの単位表示
メッセージアイテム	メッセージ	メッセージ本文
	詳細メッセージ	メッセージの詳細説明
メッセージセット	タイトル	メッセージセットのタイトル
	サマリー	メッセージセットの概要

設定アプリ

設定アプリでirBoardのチュートリアルを復元したり、設定を解除したりできます。

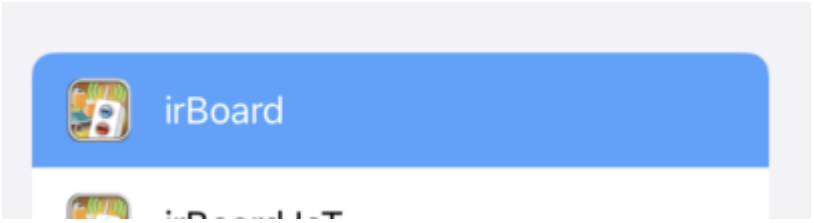
- 設定アプリを開きます。
- irBoardの項目を選択します。
  - 設定
    - ブラウズモード
      ブラウズモードを切り替えられます。
      irBoardアプリからは**OFF**にできないため設定アプリから**OFF**にします。
  - 一般
    - チュートリアルの復元
      チュートリアルを元に戻します。
  - メッセージ
    次回以降の説明を表示するかの確認で、表示しないを選択後に以下の設定をオンにすると表示する様に戻すことができます。
    - ブラウズモードの説明を表示する。
    - ナビゲーションバーの表示方法を表示する。
    - **Lite**版の説明を表示する。
  - 振る舞い
    - 値の入力を入力を前の方法に戻す
      値の入力を以前のインラインで入力する場合にオンにします。オフにすると入力するダイアログビューが表示されます。
  - レポート
    - データレポートファイル作成
      アプリ起動時にドキュメントフォルダーにプロジェクトデータをコピーします。不具合発生時にこのファイルを送っていただけると不具合箇所の解析に役立つ場合があります。
    - バグ報告
      不具合発生時にオンにするとログの記録を行います。
      オンにするとプロジェクト一覧のツールバーの 情報メニュー に「バグ報告」が表示される様になります。
  - メンテナンス
    - 次回起動時にウィンドウ状態をクリアする
      次回アプリ起動時に保存されたウィンドウ状態をすべてクリアします。ウィンドウ状態の破損によるクラッシュからの復旧に使用します。クリア後、この設定は自動的にオフになります。
    - 全データをリセット（注意して使用してください）
      次回アプリ起動時にすべてのプロジェクトデータを削除します。この

操作は元に戻せません。リセット後、この設定は自動的にオフになります。

設定アプリ

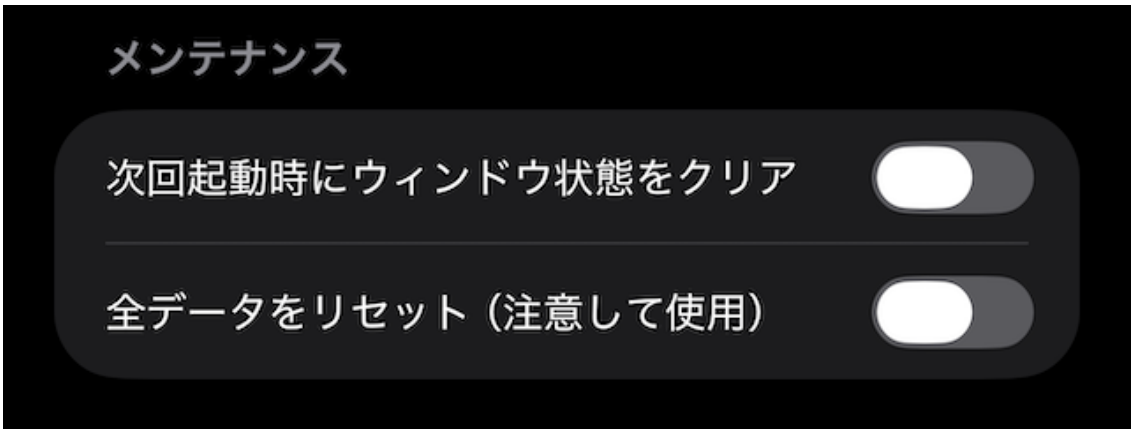


irBoard項目



設定項目


21:05
68
設定
irBoard
言語
日本語
IRBOARD設定
設定
ブラウズモード
一般
チュートリアルの復元
メッセージ
ブラウズモードの説明を表示する
ナビゲーションバーの表示方法を
表示する
振る舞い
値の入力を前の方法に戻す
レポート
データレポートファイル作成
バグ報告

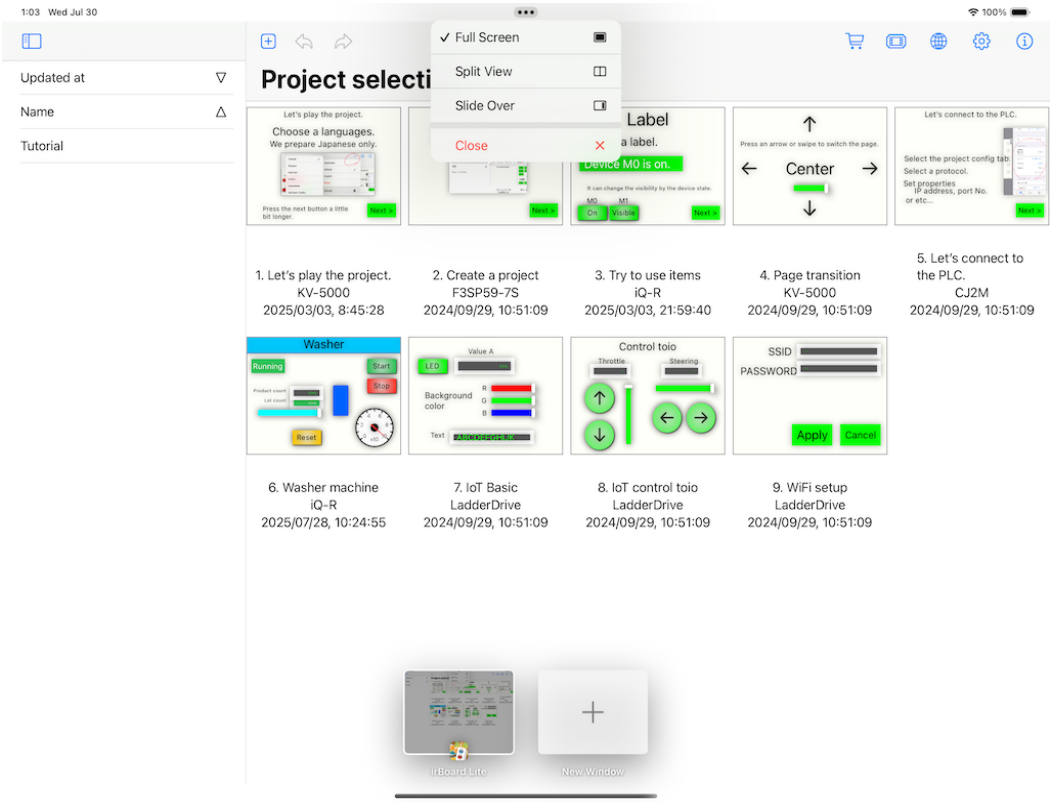

メンテナンス
次回起動時にウィンドウ状態をクリア
全データをリセット（注意して使用）

マルチタスク

irBoard ver. 2.8 から、マルチタスクに対応しました。
iPad でirBoardを使用する際、複数のプロジェクトを開いて作業できるようになりました。iPadOS 18の時点では、最大2つのプロジェクトを同時に実行できます。
これにより、 **PLC**で制御されている**2**つの装置を同時に確認したり、操作したり することが可能になります。

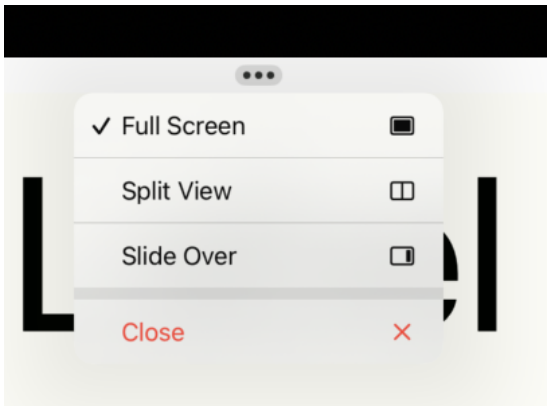
マルチタスクの使い方

・画面の上部のマルチタスクボタン(・・・ の箇所)をタッチするとマルチタスクメニューが表示されます。



下部に開いているプロジェクトの一覧や追加(+)ボタンが表示されます。追加(+)を選択すると新たなウインドウを開くことができます。また一覧からウインドウを切り替えることができます。

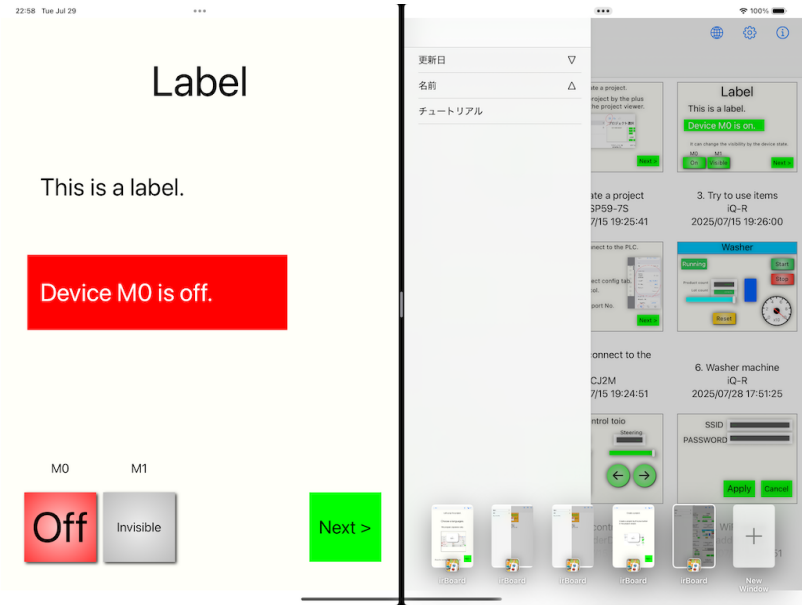
・マルチタスクメニューでSplit ViewかSlide Overを選ぶと対になるアプリを選択できる様になります。



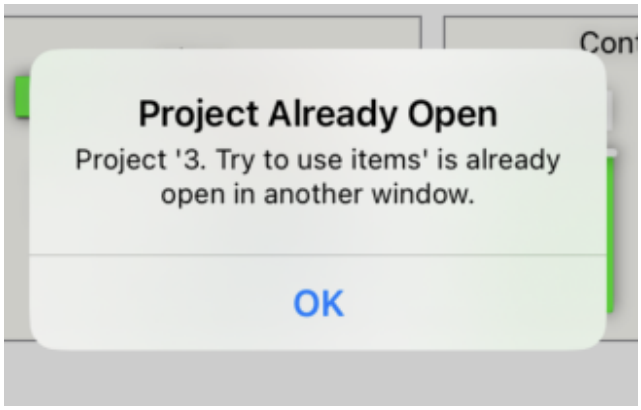
この状態でirBoardアイコンをタッチすると左右に2つのウインドウが表示されます。またはirBoardアイコンを軽く長押しし、表示されるメニューからShow All Windowsを選択することで開いているウインドウから対になるウインドウを選択することもできます。



・Split ViewやSlide Overを選択することで2つのプロジェクトを同時に実行することできます。



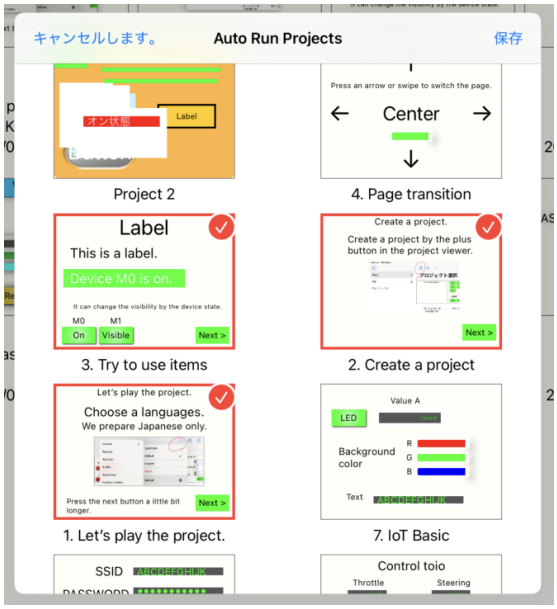
・同一のプロジェクトを複数のウインドウで開くことはできません。同一のプロジェクトを開こうとした場合に警告が出ます。この時は既に開いているウインドウを選択してください。



・irBoardが終了しても開いたウインドウの状態が記憶され次回起動時に再現されます。(場所は変わる場合があります)
・マルチタスクの使い方についてはAppleの公式サイトもご覧ください。
iPad でマルチタスク機能を使う

自動実行

・自動実行 では複数のプロジェクトを選択できる様になりました。



・プロジェクトが自動実行されるのは次の何れかの時に実行されます。
1. 自動実行として選択したプロジェクトのウインドウを選択した時
2. 自動実行として選択したプロジェクトのウインドウがない場合は、プロジェクト一覧画面のウインドウを選択した時
3. プロジェクト一覧画面のウインドウもない場合は、新たにウインドウが追加された時

irBoard起動時に複数のプロジェクトを自動実行させるには、ウインドウを予め必要数開いている必要があります。

Acknowledgements
This application makes use of the following third party libraries and fonts:

KeychainAccess

The MIT License (MIT)

Copyright (c) 2014 kishikawa katsumi

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

SSZipArchive

Copyright (c) 2013-2021, ZipArchive, https://github.com/ZipArchive

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE

```
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.


## DSEG

Copyright (c) 2020, keshikan (https://www.keshikan.net),
with Reserved Font Name "DSEG".


This Font Software is licensed under the SIL Open Font License, Version 1.1.
This license is copied below, and is also available with a FAQ at:
http://scripts.sil.org/OFL


-----------------------------------------------------------
SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007
-----------------------------------------------------------

PREAMBLE
The goals of the Open Font License (OFL) are to stimulate worldwide
development of collaborative font projects, to support the font creation
efforts of academic and linguistic communities, and to provide a free and
open framework in which fonts may be shared and improved in partnership
with others.

The OFL allows the licensed fonts to be used, studied, modified and
redistributed freely as long as they are not sold by themselves. The
fonts, including any derivative works, can be bundled, embedded,
redistributed and/or sold with any software provided that any reserved
names are not used by derivative works. The fonts and derivatives,
however, cannot be released under any other type of license. The
requirement for fonts to remain under this license does not apply
to any document created using the fonts or their derivatives.

DEFINITIONS
"Font Software" refers to the set of files released by the Copyright
Holder(s) under this license and clearly marked as such. This may
include source files, build scripts and documentation.

"Reserved Font Name" refers to any names specified as such after the
copyright statement(s).

"Original Version" refers to the collection of Font Software components as
distributed by the Copyright Holder(s).

"Modified Version" refers to any derivative made by adding to, deleting,
or substituting -- in part or in whole -- any of the components of the
```

Original Version, by changing formats or by porting the Font Software to a new environment.

"Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

PERMISSION & CONDITIONS
Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

1) Neither the Font Software nor any of its individual components, in Original or Modified Versions, may be sold by itself.

2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.

3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.

4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.

5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.

TERMINATION
This license becomes null and void if any of the above conditions are not met.

DISCLAIMER
THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,

```
INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL
DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM
OTHER DEALINGS IN THE FONT SOFTWARE.
```