

ユーザーズマニュアル

AI-HMI

目 次

第 1 章 AI-HMIについて

1-1 概要	1-1
1-2 特徴	1-2

第 2 章 インストール手順

2-1 開発環境	2-1
2-2 AI-HMI開発環境のインストール	2-6

第 3 章 AI-HMI開発環境を使用する前に

3-1 コンポーネントの組み込み	3-1
------------------	-----

第 4 章 AI-HMIの構成について

4-1 AI-HMIのページ構造について	4-1
4-2 カスタムコンポーネント	4-2
4-3 AI-HMI用INtime共有メモリ	4-4

第 5 章 AI-HMI画面アプリケーションの作成

5-1 プロジェクトの新規作成	5-2
5-2 ページの作成	5-5
5-3 フレームの設定	5-22
5-4 アプリケーションのコンパイルとデバッグ	5-26
5-5 ダイアログの追加	5-29
5-6 ラベルによる共有メモリのデータの表示	5-46

第6章 作成したアプリの実行環境PCでの動作

6-1 Releaseモードでのコンパイル	6-1
-----------------------	-----

第7章 AI-HMIデバッグツール

7-1 デバッグツールの概要	7-1
7-2 デバッグツールの使用方法	7-2
7-3 画面アプリケーションとデバッグツールの連動	7-5

第8章 プロジェクト構成

第9章 その他の機能の使用方法

9-1 アラーム機能	9-1
9-2 操作ログ機能	9-2 2
9-3 ユーザログイン機能	9-2 5
9-4 サンプリング機能	9-3 5
9-5 スケジューラ機能	9-4 4
9-6 トリガアクション機能	9-5 0
9-7 レシピ切替機能	9-5 7
9-8 統計グラフ機能	9-6 1
9-9 カスタマイズ ソフトキー	9-6 8
9-10 タッチ時のBeep音機能	9-7 6
9-11 外部アプリケーションの実行	9-8 2
9-12 コンポーネントの位置変更	9-8 6

第10章 多言語切替機能

10-1 言語設定ファイルの作成	10-1
10-2 言語切替画面の作成	10-5

第 1 1 章 サンプルプログラムについて

第 1 2 章 AI-PLC開発環境での使い方 (OPC変数)

1 2 - 1 OPC変数の登録	1 2 - 1
------------------	---------

第 1 3 章 AI-PLC開発環境での使い方 (共有メモリ)

1 3 - 1 I/Oグループの設定	1 3 - 1
1 3 - 2 変数のアドレス指定	1 3 - 4

第 1 4 章 ユーザ用システム変数

第 1 5 章 システム内システム変数

第 1 6 章 RAM Backup領域

第 1 7 章 Tips集

1 7 - 1 一度設定したプロパティを初期化したい	1 7 - 1
1 7 - 2 オブジェクトの画像を変更したい	1 7 - 2
1 7 - 3 画像ファイルを登録したい	1 7 - 6
1 7 - 4 ページ名がわかりにくい/ページ名を変更したい	1 7 - 1 0
1 7 - 5 複数の画像ファイルをアニメーション表示したい	1 7 - 1 2
1 7 - 6 画面に配置した部品をダブルクリックしてしまった	1 7 - 1 7

17-7 音声ファイルをすべての画面で再生したい	17-22
17-8 Windows起動時にAI-HMIアプリケーションを起動するようにしたい	17-24
17-9 タッチパネルでボタンをタッチしたときにマウスと同じ挙動にしたい	17-28
17-10 Mouse Modeで右クリック操作をしたい	17-30
17-11 タッチポインタを非表示にしたい	17-32
17-12 AI-HMIから共有メモリへSTRING型で値を書き込めなくなった	17-34
17-13 CSVファイルでSTRING型の値を使用したい	17-35

はじめに

AI-HMI は Microsoft 社製 Visual Studio を開発環境として利用できる HMI 画面作成ソフトです。
AI-HMI-SDK は VisualStudio のアドオンとして機能するため、下記の対象バージョンの Visual Studio を用意してください。

■動作対象の Visual Studio

Visual Studio 2013

Visual Studio 2015

Visual Studio 2017

Visual Studio 2019

Visual Studio 2022

※ 上記以外では動作確認していません。

※ Visual Studio はお客様とマイクロソフト社との直接契約をしていただき、ご利用ください。

AI-HMI で作成したアプリケーションは AI-HMI ランタイム搭載製品でのみ動作できます。

AI-HMI ランタイム搭載製品は以下の通りです。

○高機能表示機 (AI-HMI 版)

G-AP4A-xxx シリーズ

G-EC4A-xxx シリーズ

○オールインワンコントローラ (AI-HMI 版)

C-AP4A-xxx シリーズ

C-EC4A-xxx シリーズ

C-NP7A-xxx シリーズ

C-NP6A-xxx シリーズ

C-NP4C-xxx シリーズ

μFIX コントローラ

これら以外の実行環境で実行した場合や開発環境にてデバッグ実行した場合は 60 分毎にライセンスを
求めるダイアログが開くようになります。

Visual Studio2022 対応について

Visual Studio2022 に対しては AI-HMI-SDK ver1.3.2.2 以降のバージョンで対応しています。

Ver1.3.2.1 以前の AI-HMI-SDK をご利用の方はアップデートパッチを使用してアップデートをしてください。

・開発環境アップデート

弊社ホームページからアップデートパッチをダウンロードできます。

AI-HMI-SDK をセットアップした PC で使用することで VisualStudio2022 に対応できるようになります。

・プロジェクトコンバート

既に Ver1.3.2.1 以前の AI-HMI-SDK で作成しているプロジェクトについては AI-HMI-SDK の CD-ROM に
含まれている AI-HMI VS2022 Converter を使用することで Visual Studio2022 に対応できます。

AI-HMI VS2022 Converter についての詳細は CD-ROM 同梱の『AI-HMI VS2022 Converter 取扱説明書』を
ご参照ください。

AI-HMI VS2022 Converter は弊社ホームページからもダウンロードできます。

第 1 章 AI-HMI について

1-1 概要

AI-HMI は、Microsoft 社製 VisualStudio を開発環境として利用できる、HMI 画面作成ソフトです。AI-HMI は Visual Studio のプロジェクト管理機能、コンパイル機能、デバッグ機能、デザイン機能を利用しています。

Visual Studio で開発する Windows アプリケーションでは、画面デザインと一緒に、その機能をコーディングする必要がありますが、AI-HMI で用意したカスタムコンポーネントを使用することで、プロパティ設定のみで HMI 画面を構築することが可能です。

AI-HMI は Intime 上で動作するソフト PLC (AI-PLC) や、ユーザで作成したリアルタイムアプリケーションと共有メモリを介して接続することで、制御側のパラメータ設定や、読み出し、実行トリガのアクセスを行うことが可能です。

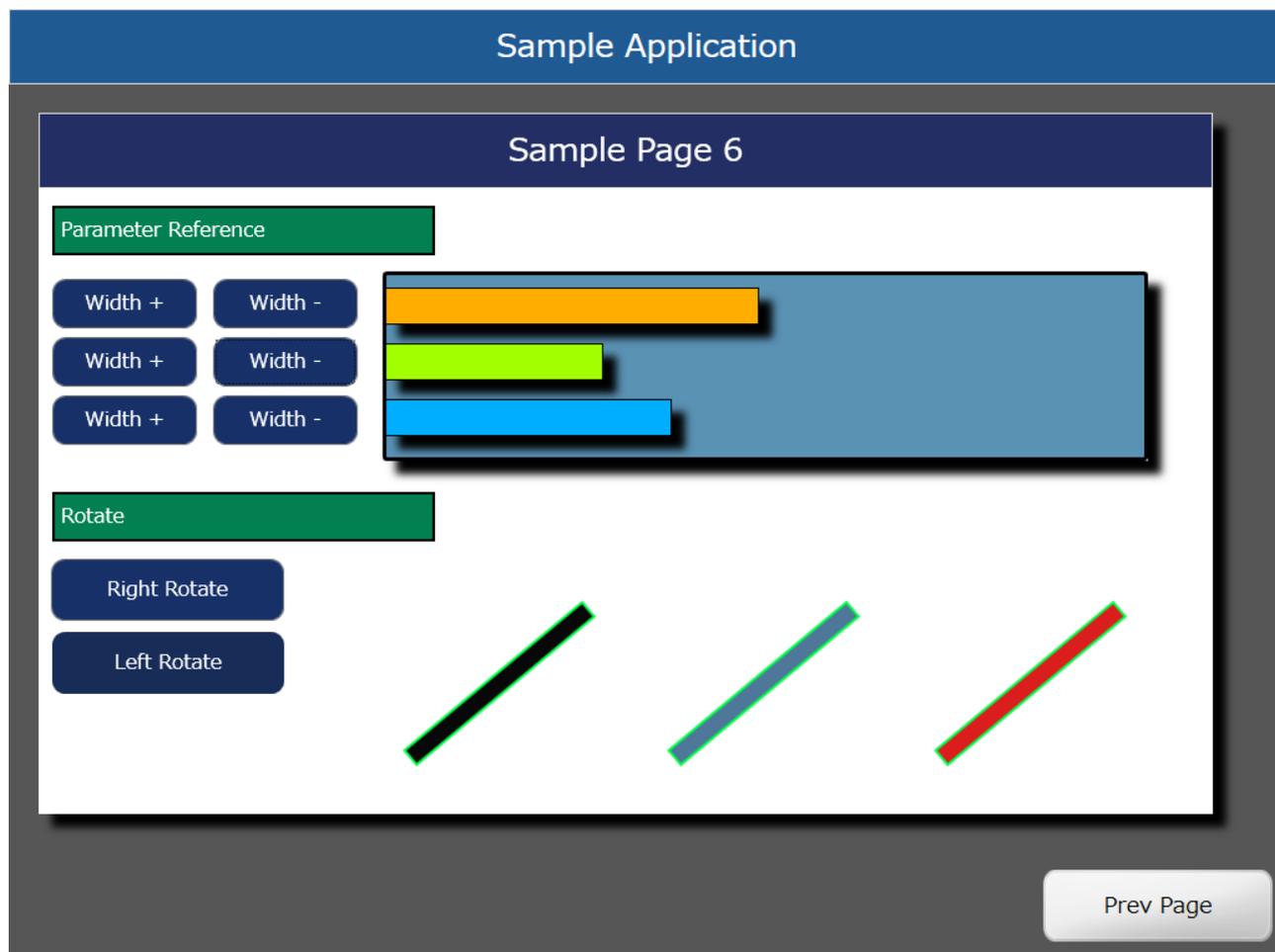


図 1-1-1 AI-HMI によって作成した画面の例

1-2 特徴

1-2-1 Microsoft 社製 Visual Studio で開発可能

AI-HMI の開発プラットフォームは、Microsoft 社の Visual Studio 統合開発環境を採用し、アドオンとして動作しています。

AI-HMI は HMI 画面作成に必要なボタンやラベル等のオブジェクトをカスタムコンポーネント化し、HMI 画面の開発は Visual Studio のデザイン機能を使用します。また、Visual Studio のプロジェクト管理機能と、コンパイル、デバッグ機能をそのまま使用するため、高機能かつ容易に HMI 画面を開発することが可能です。

AI-HMI は Microsoft 社より公開されている Visual Studio の無償版でも開発可能となっていますので、HMI 画面の開発コストを低くすることができます。

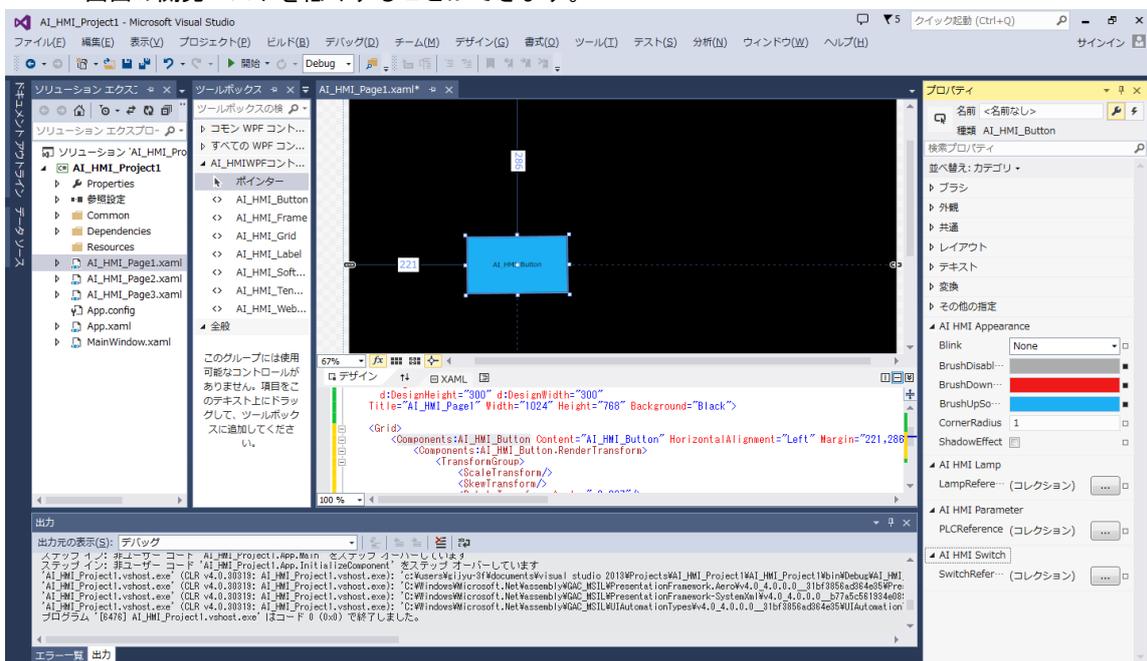


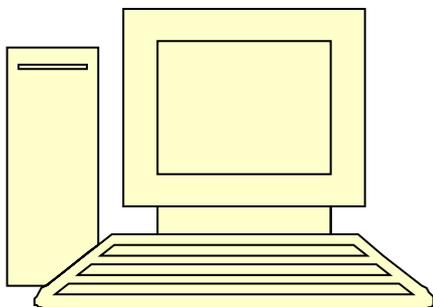
図 1-2-1-1 Visual Studio 上にアドオンされた AI-HMI 開発環境

- ※ Visual Studio は Microsoft 社の製品です。
Visual Studio そのものの挙動については、弊社が動作を保証するものではありません。
Visual Studio の不具合について弊社は責任を負うものではありませんので予めご了承ください。

1-2-2 AI-HMI の構成図

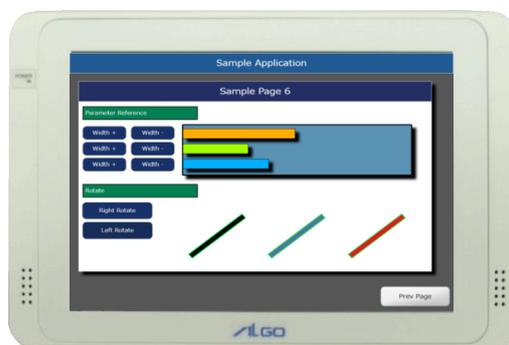
AI-HMI は、INtime 上で動作するソフト PLC (AI-PLC) または外部 PLC との連携を前提に開発されています。INtime 上で動作するソフト PLC とは、INtime 共有メモリまたは OPC 変数によりデータを参照します。INtime 共有メモリは通信によるデータのやり取りではなく、直接メモリを参照するためデータの反映を高速に行うことができます。OPC 変数は MULTIPROG 上の変数を設定することで、同じ変数名で AI-HMI 上で使用・通信することが可能です。外部 PLC とは LAN ケーブルを接続して通信し、その値を読み書きすることができます。

開発環境 PC
(一般的な Windows パソコン)



アプリケーション画面の開発

実行環境 PC
(弊社販売のパネル型コンピュータ等)



実際に現場で画面として表示

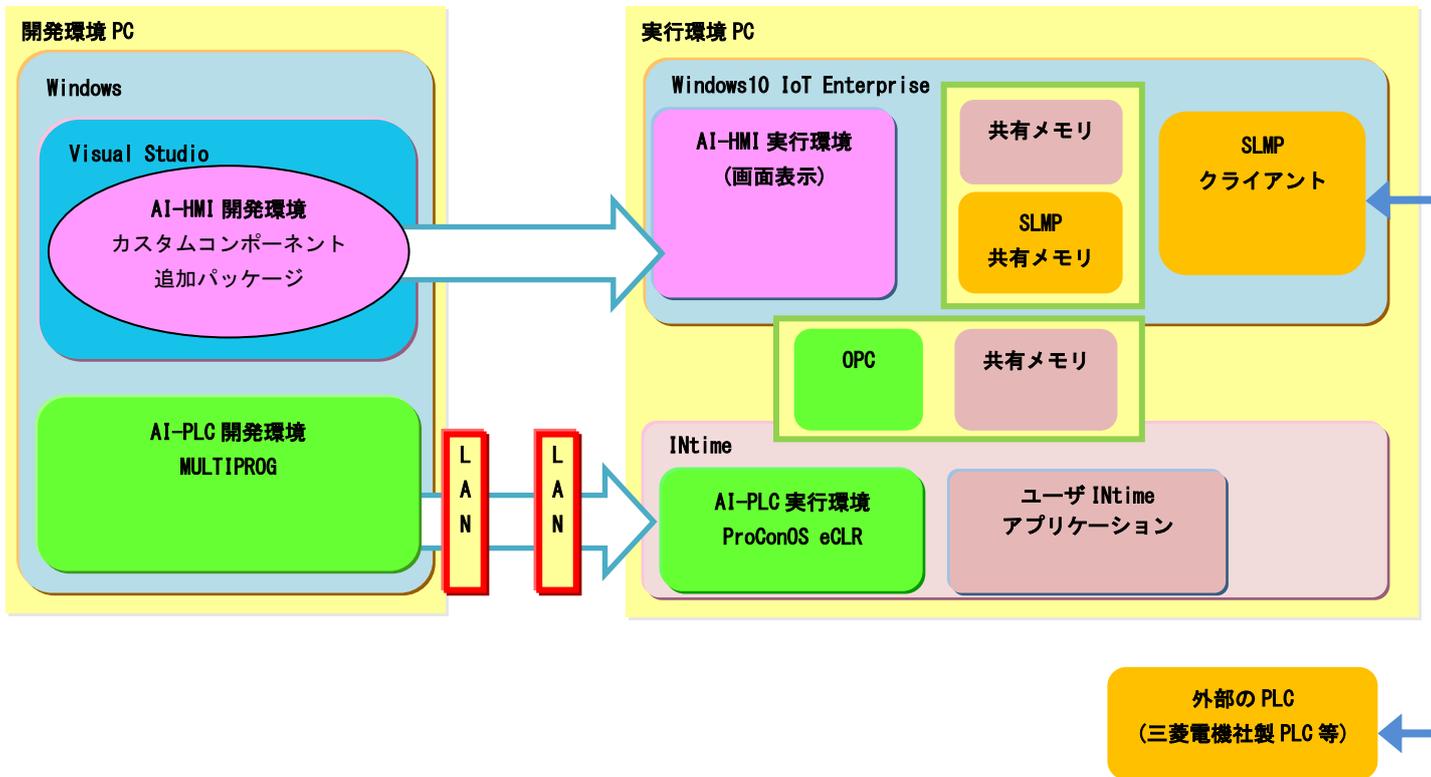


図 1-2-2-1 AI-HMI 開発環境、実行環境構成図

第2章 インストール手順

2-1 開発環境

AI-HMI で画面アプリケーションを開発するにあたって、事前に作業すべき内容について手順を解説します。AI-HMI を使用するには Visual Studio 2022, 2019, 2017, 2015 または 2013 を利用しますので用意してください。

- ※ 既に Visual Studio2022, 2019, 2017, 2015, 2013 をお持ちの場合はそのままご使用いただけます。
- ※ 開発環境 PC の OS は Windows10 以降のバージョンがインストールされている必要があります。

・旧バージョンとの互換性について

AI-HMI ver1.3.2.1 以前のバージョンで作成したプロジェクトは Visual Studio 2022 で編集しようとする正常に表示されません。

この場合は AIHMI_VS2022_Converter ツールをご使用ください。

AIHMI_VS2022_Converter ツールについての詳細は別紙『AIHMI_VS2022_Converter ツールについて』をご参照ください。

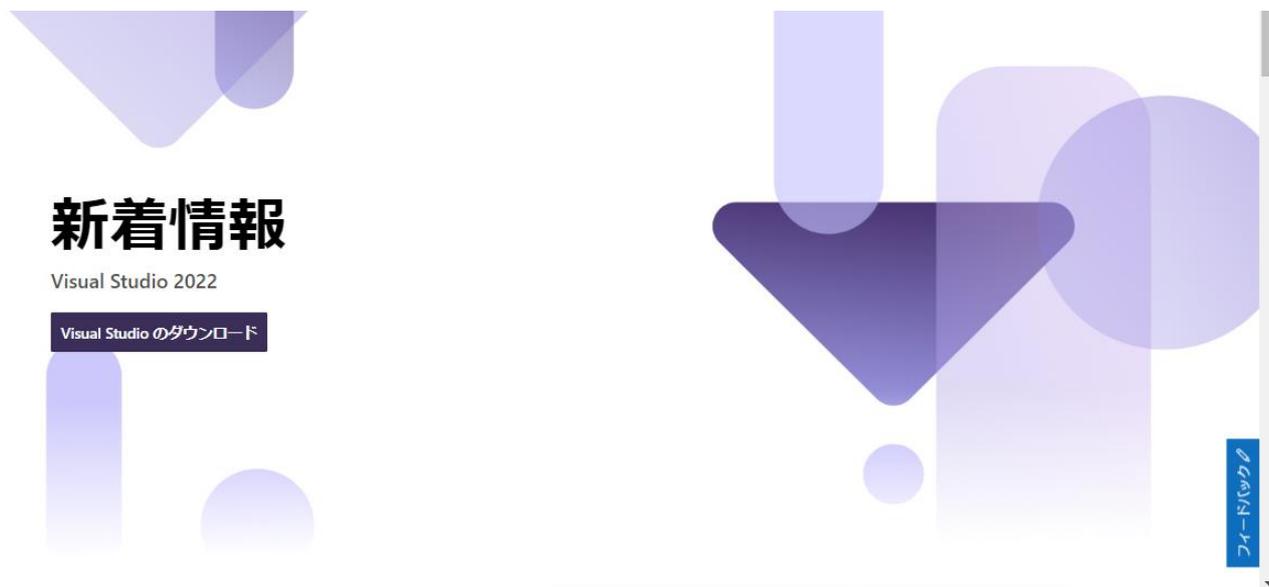
■ 参考

製品版の Visual Studio をお持ちではない場合は無償版の Visual Studio で使用することも可能です。参考として 2023 年 5 月時点の入手方法と使用方法を記載します。

- ※ Visual Studio の入手と使用についてはお客様とマイクロソフト社との契約となります。

2023 年 5 月時点では無償版 Visual Studio は Visual Studio 2022, 2019 の Community 版を利用できます。Visual Studio 2022 Community は Microsoft ホームページからインストーラを入手できます。

<https://visualstudio.microsoft.com/ja/vs/whatsnew/>



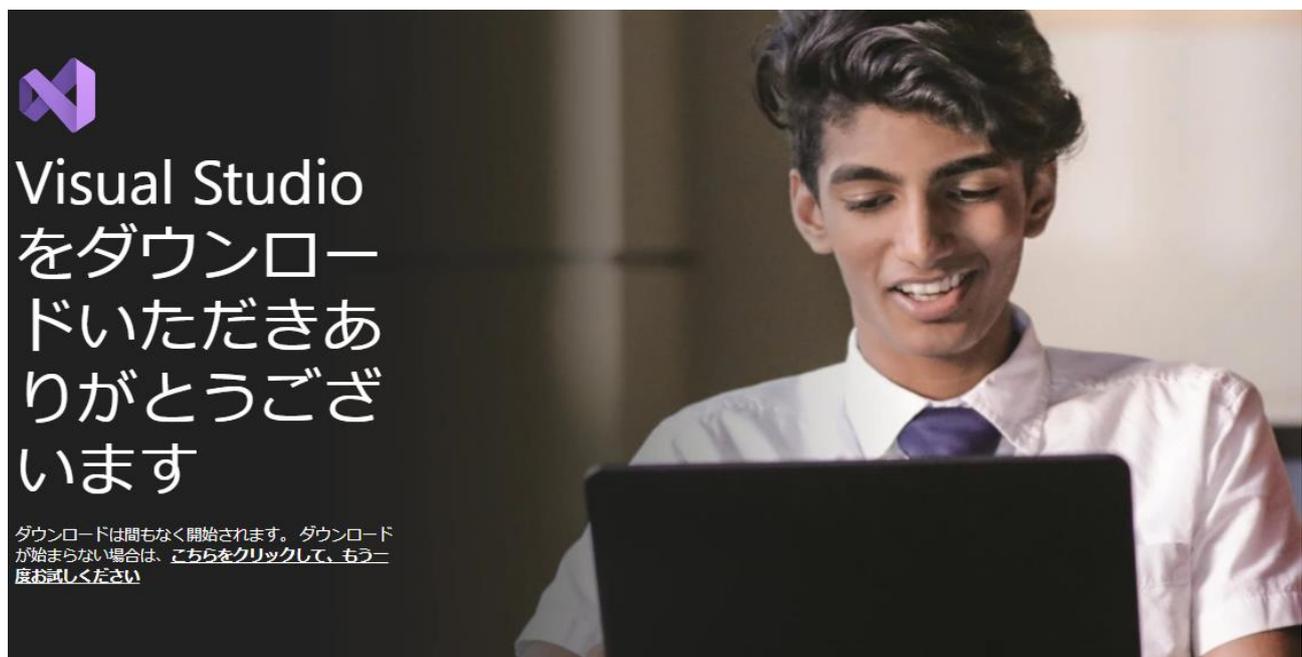
参考 図1 Visual Studio2022 インストーラのダウンロード

「Visual Studio のダウンロード」をクリックし、「Community 2022」を選択してください。



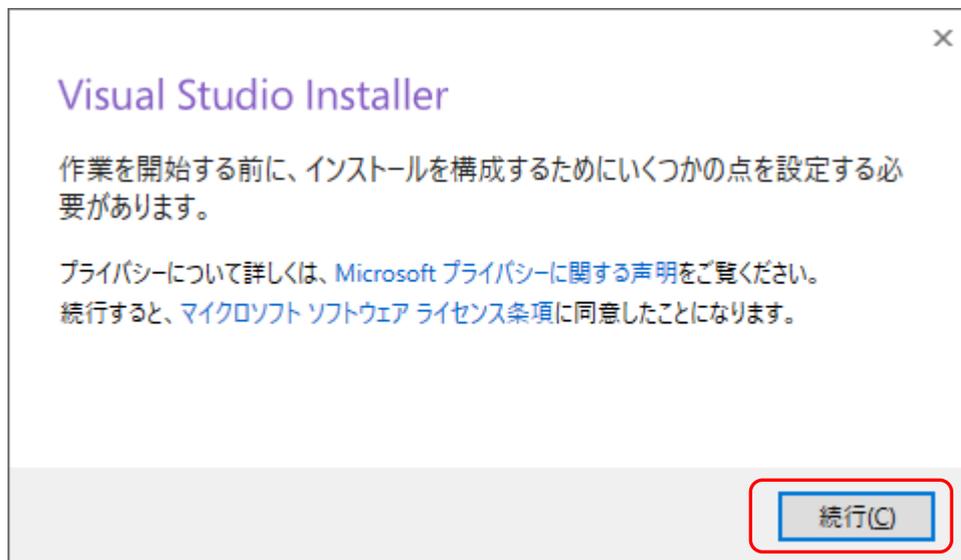
参考 図2 Visual Studio 2022 Community

インストーラのダウンロードが始まるので、そのままお待ちください。



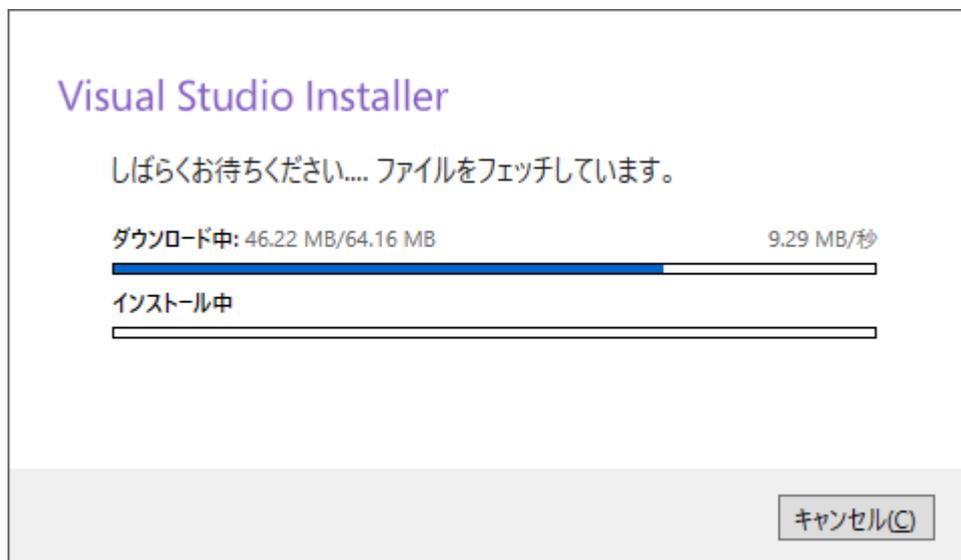
参考 図4 ダウンロード開始

インストーラダウンロードが完了したらインストーラを実行してください。参考 図5の画面が表示されます。「続行」ボタンをクリックしてください。



参考 図5 Visual Studio インストールの実行

Visual Studio のインストールが始まります。参考 図6の画面が消えるまでお待ちください。



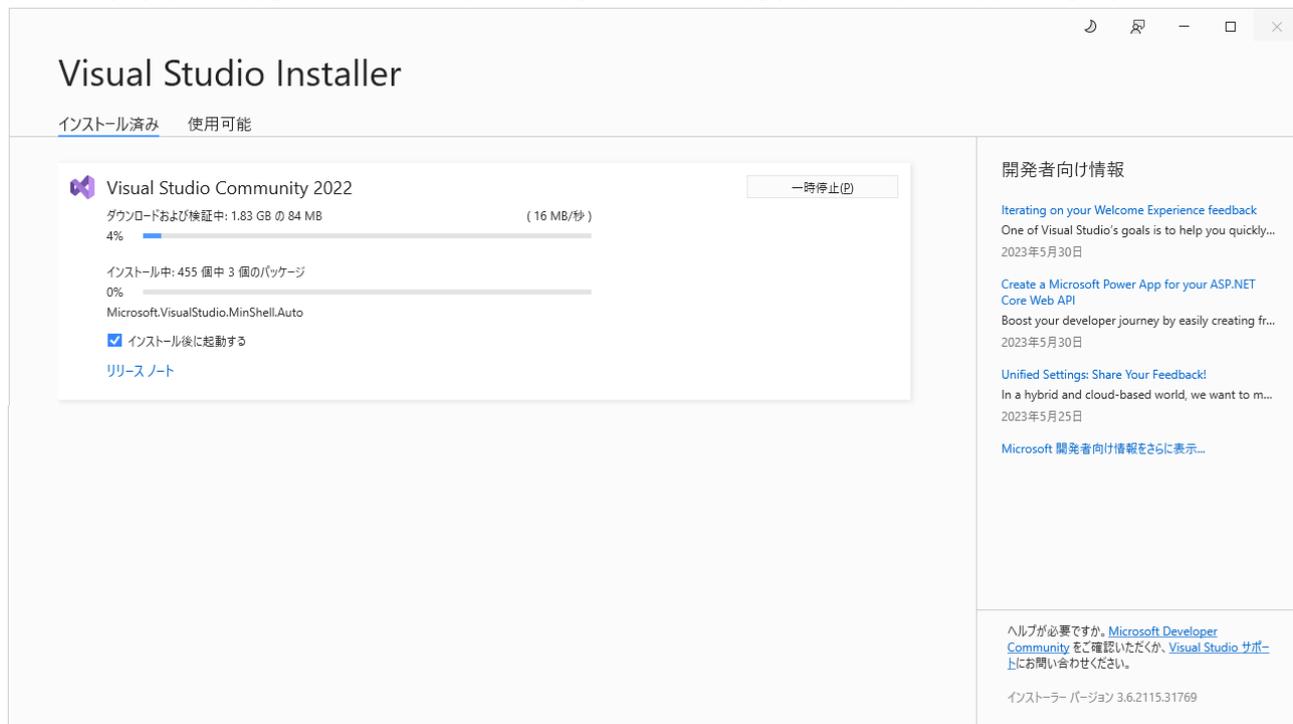
参考 図6 Visual Studio インストール準備

参考 図 7 が開きます。「.NET デスクトップ開発」にチェックを入れ、「インストール」ボタンをくりっくしてください。



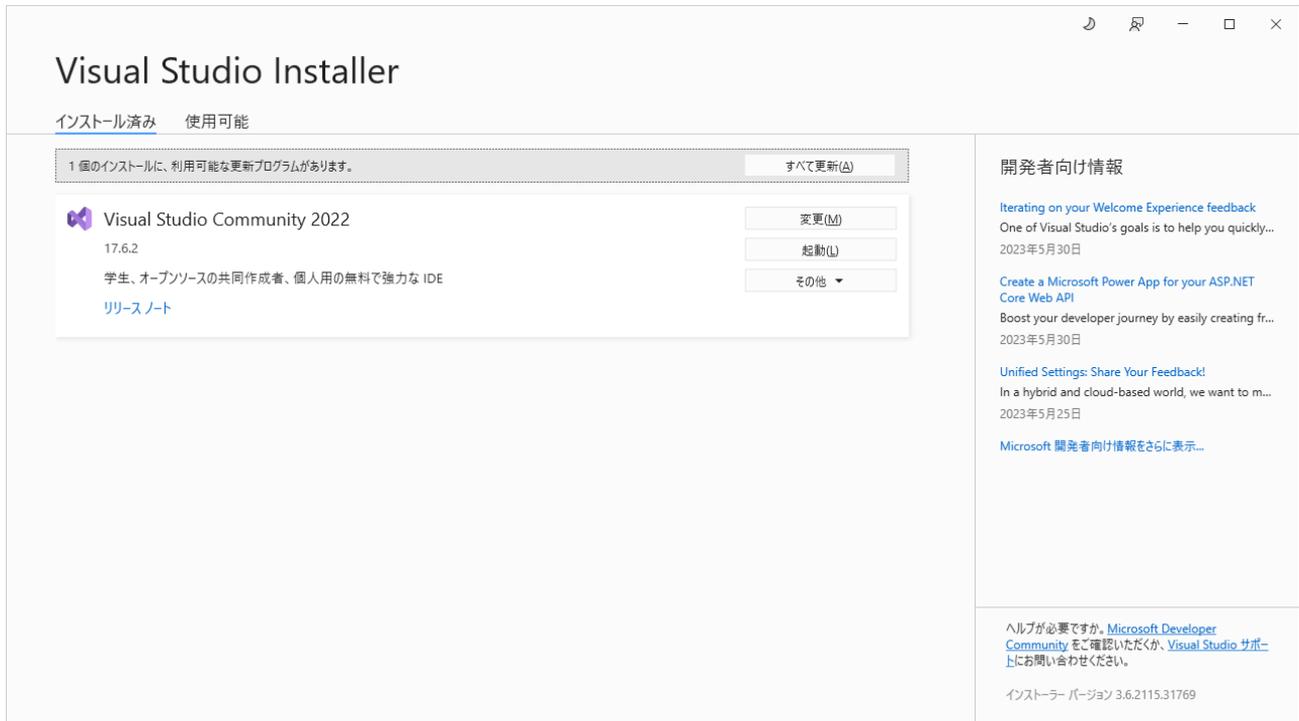
参考 図 7 追加コンポーネントの設定

参考 図 8 が開きインストールが始まります。インストール完了までしばらくお待ちください。



参考 図 8 インストール開始

インストールが完了すると参考 図 9 が開きます。



参考 図 9 インストール完了

2-2 AI-HMI 開発環境のインストール

- ① VisualStudio に AI-HMI 開発環境を組み込みます。
添付の CD-R の以下のファイルを実行してください。

[CD-R]¥SDK¥setup.exe

setup.exe を実行するとインストールが開始されます。

図 2-2-1 の画面が表示されるまでしばらくお待ちください。

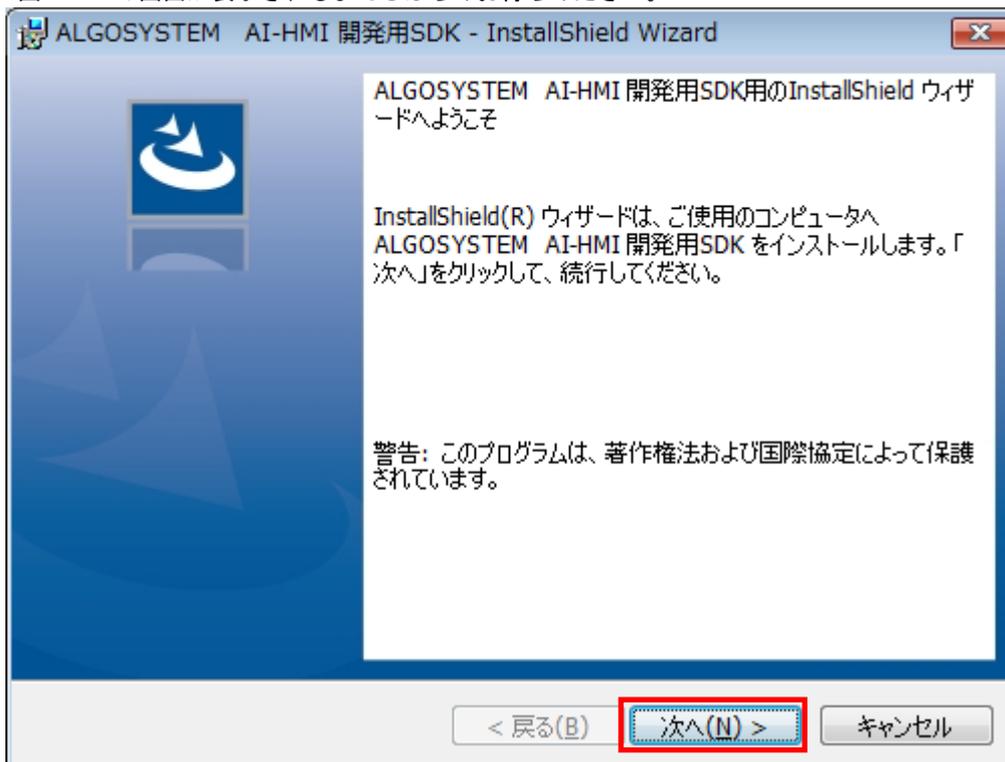


図 2-2-1 AI-HMI 開発環境インストール画面

図 2-2-1 が表示されたら「次へ」ボタンをクリックしてください。

- ② インストール先のフォルダの確認画面が開きます。
インストール先が以下のフォルダになっていることを確認してください。

C:¥ALGOSYSTEM¥AI-HMI 開発環境用 SDK

これ以外のフォルダが指定されている場合は「変更」ボタンをクリックし、インストール先のフォルダを上記フォルダに変更してください。

※ **必ずCドライブの上記フォルダを指定してください。**

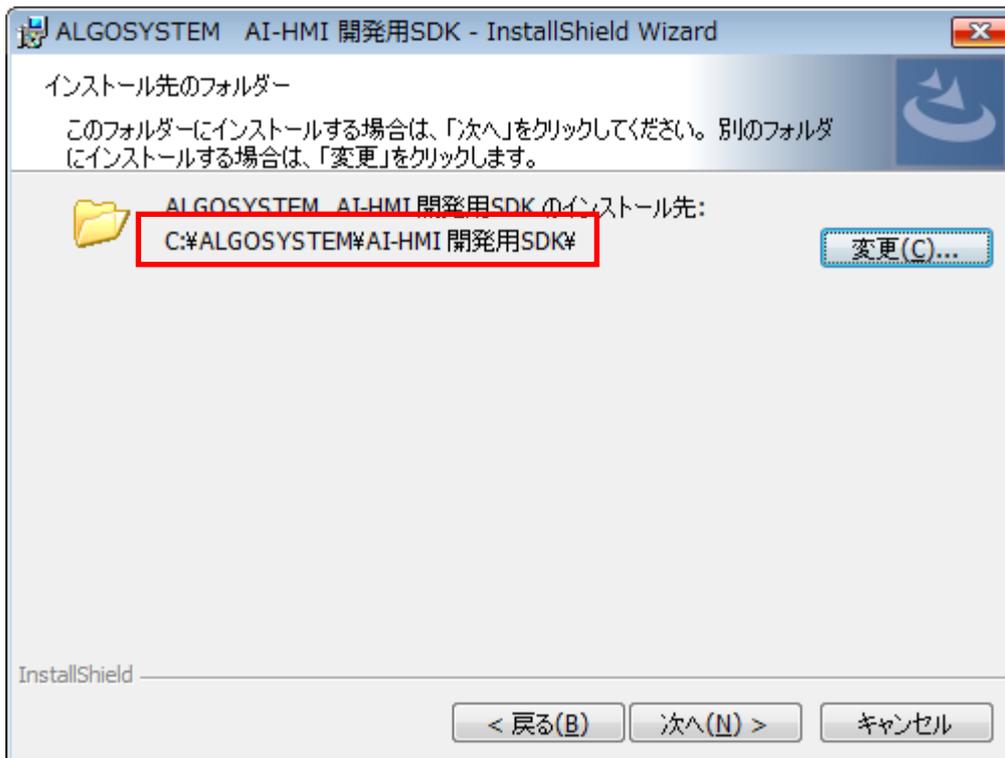


図 2-2-2 インストール先フォルダの確認画面

インストール先フォルダに問題がなければ「次へ」をクリックしてください。

- ③ 図 2-2-3 のインストール情報画面が表示されます。
「インストール」ボタンをクリックしてください。

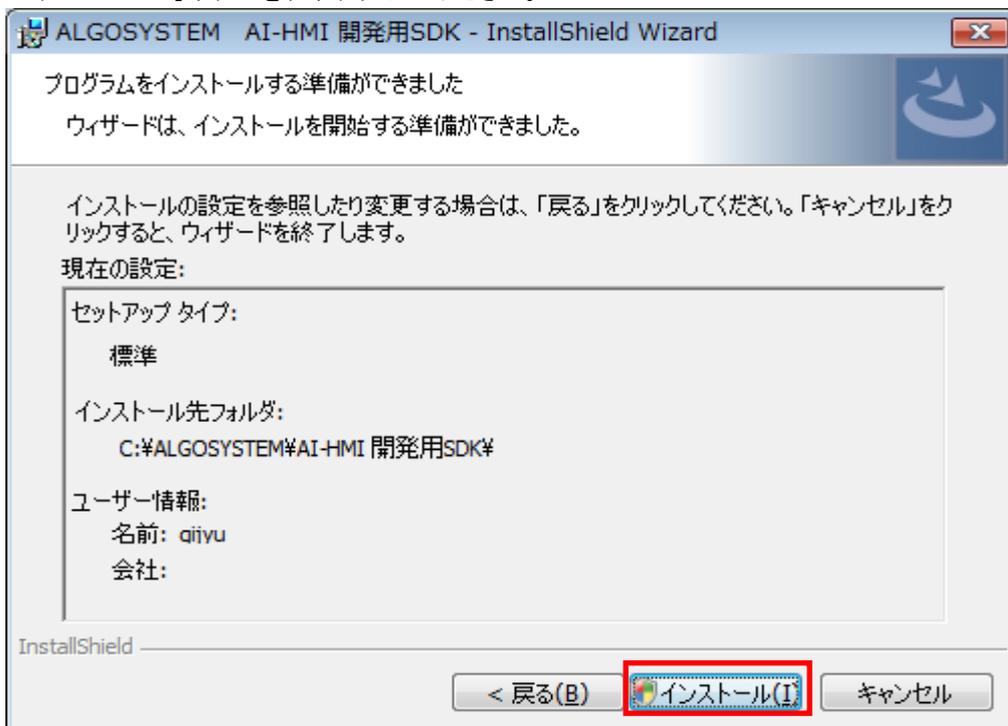


図 2-2-3 インストール情報画面

- ④ 図 2-2-4 の画面が表示され、インストールが開始されます。
インストールが完了するまでしばらくお待ちください。

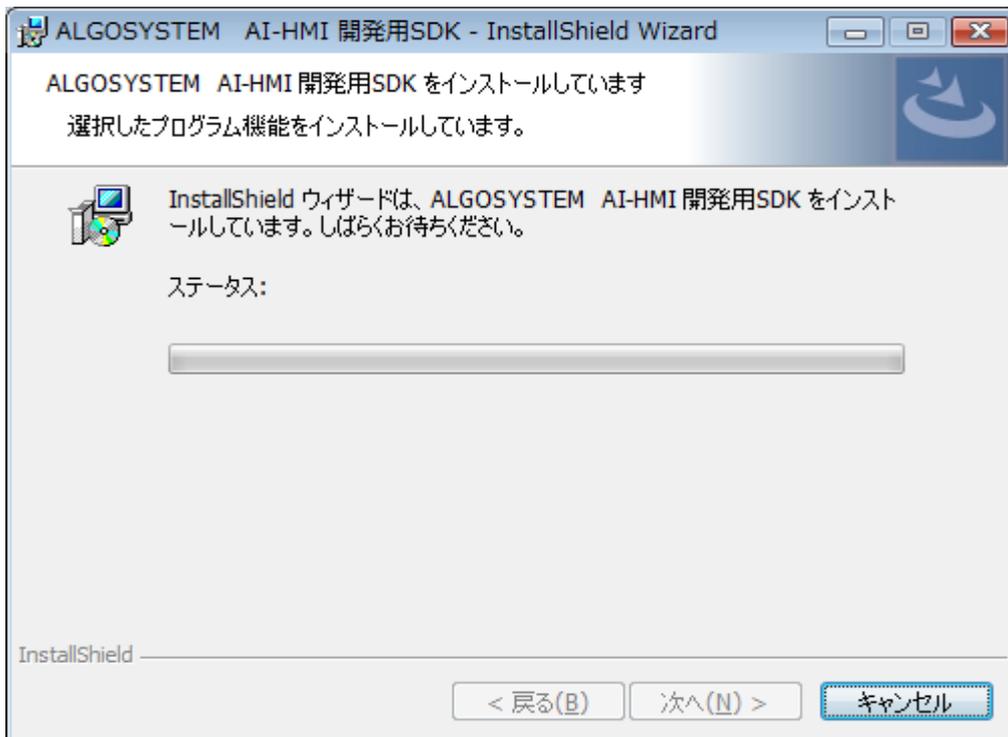


図 2-2-4 インストール中画面

- ⑤ インストールが完了すると図 2-2-5 の画面が表示されます。
「完了」ボタンをクリックし、インストーラを終了してください。

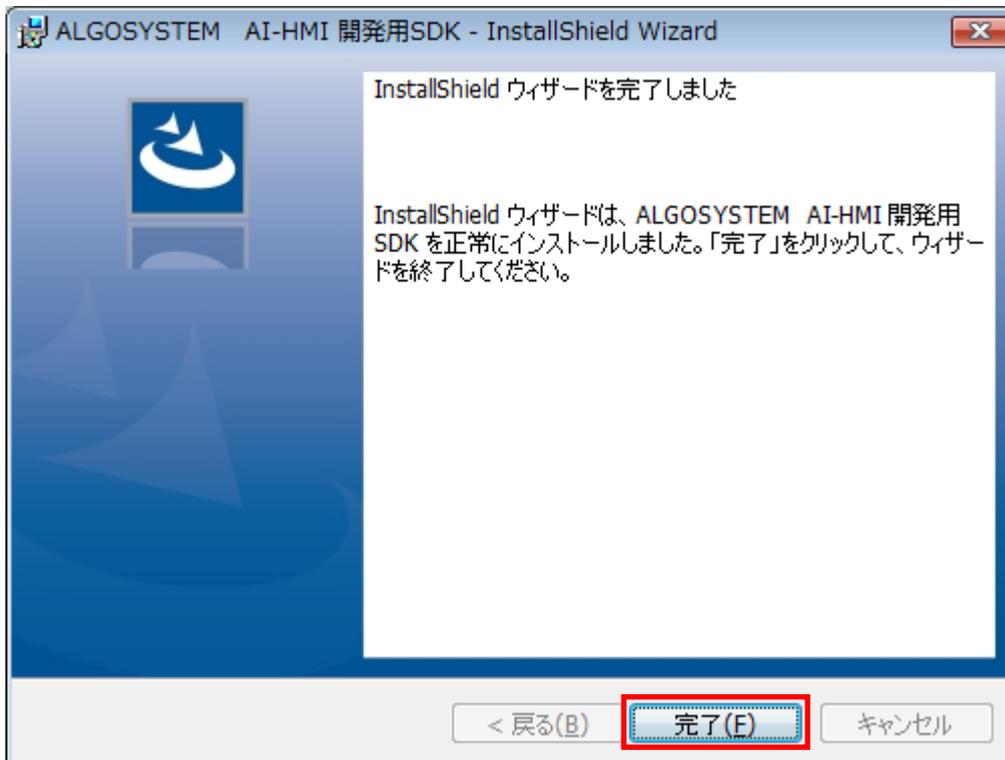


図 2-2-5 インストール完了画面

第3章 AI-HMI 開発環境を使用する前に

3-1 コンポーネントの組み込み

AI-HMI 開発環境でアプリケーションの開発を始める前に AI-HMI コンポーネントを Visual Studio に組み込む必要があります。

この作業は VisualStudio インストール後の初回起動時のみ必要です。
一度組み込みが完了したら、次回以降は作業の必要はありません。

本章では Visual Studio 2015 Express 版で操作した際の手順を記載します。
Visual Studio のバージョンに合わせた操作をしてください。

- ① Windows のスタートメニューから「すべてのプログラム – VS Express for Desktop」を実行してください。

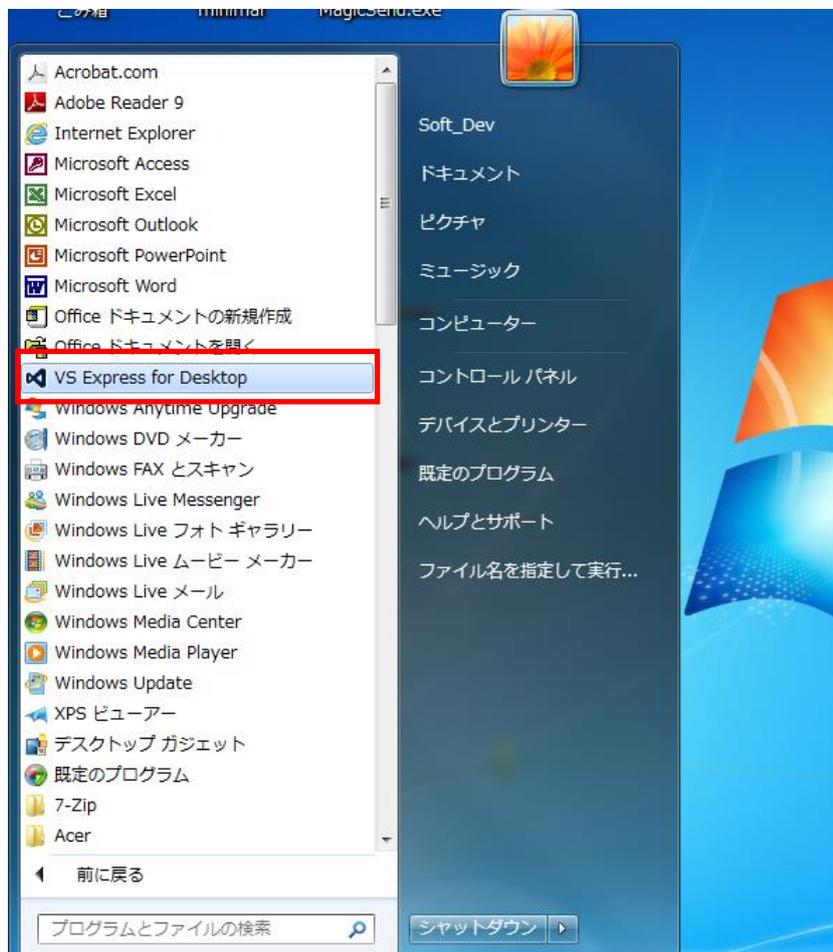


図 3-1-1 VisualStudioExpress の起動

- ② VisualStudioExpress が起動します。
「新しいプロジェクト」を選択してください。



図 3-1-2 新しいプロジェクトの作成

- ③ 左ペインから「テンプレート – Visual C#」の項目を選択してください。
中央ペインにテンプレート一覧が表示されます。「AI_HMI_Project」を選択してください。
プロジェクト名、作業場所、ソリューション名を入力したあと「OK」ボタンをクリックしてください。

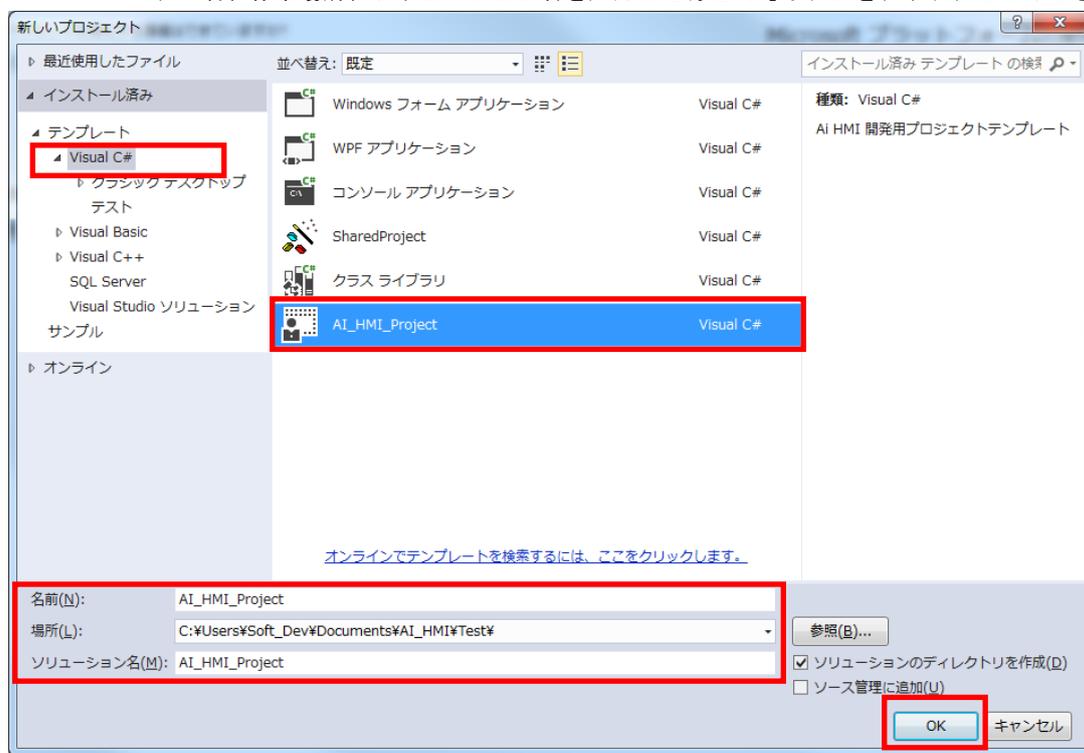


図 3-1-3 AI-HMI プロジェクトの作成

- ④ 新規のプロジェクト画面が開きます。
「ソリューションエクスプローラー」を開き、「MainWindow.xaml」をダブルクリックしてください。

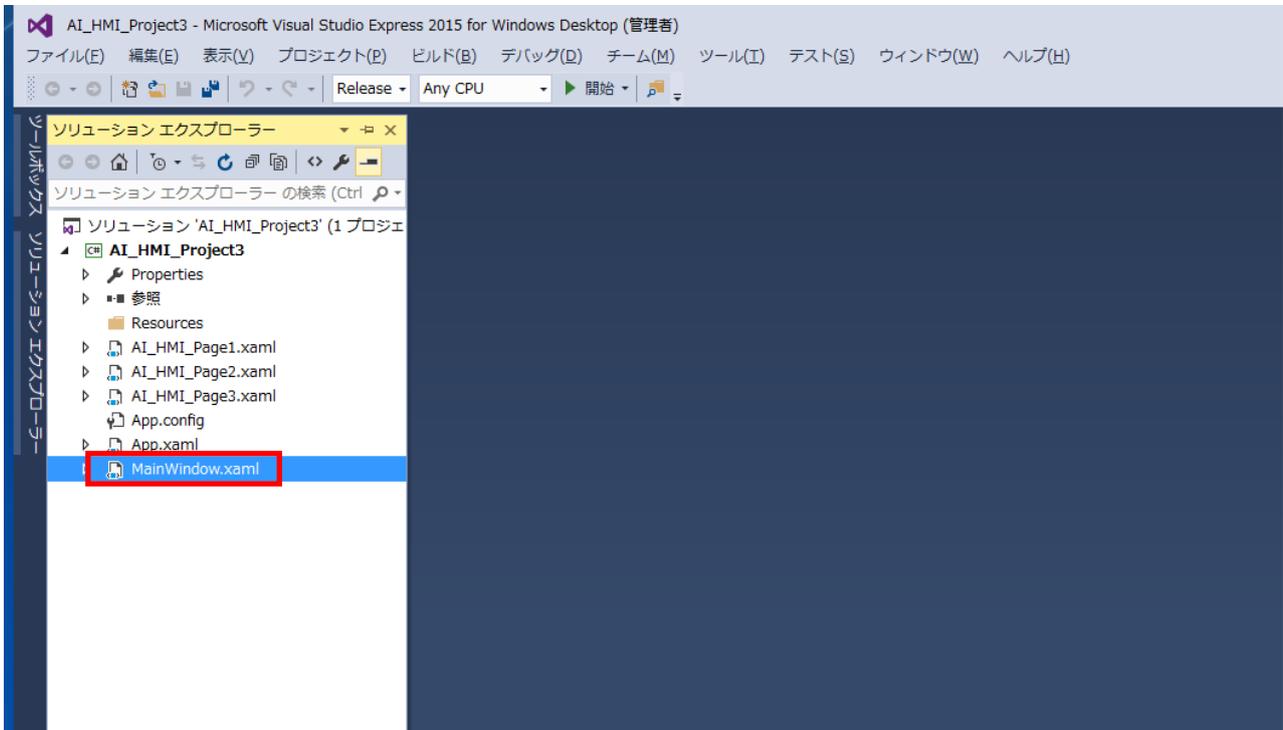


図 3-1-4 ソリューションエクスプローラー

- ⑤ 「ツールボックス」を開き、右クリックメニューから「タブの追加」を選択してください。

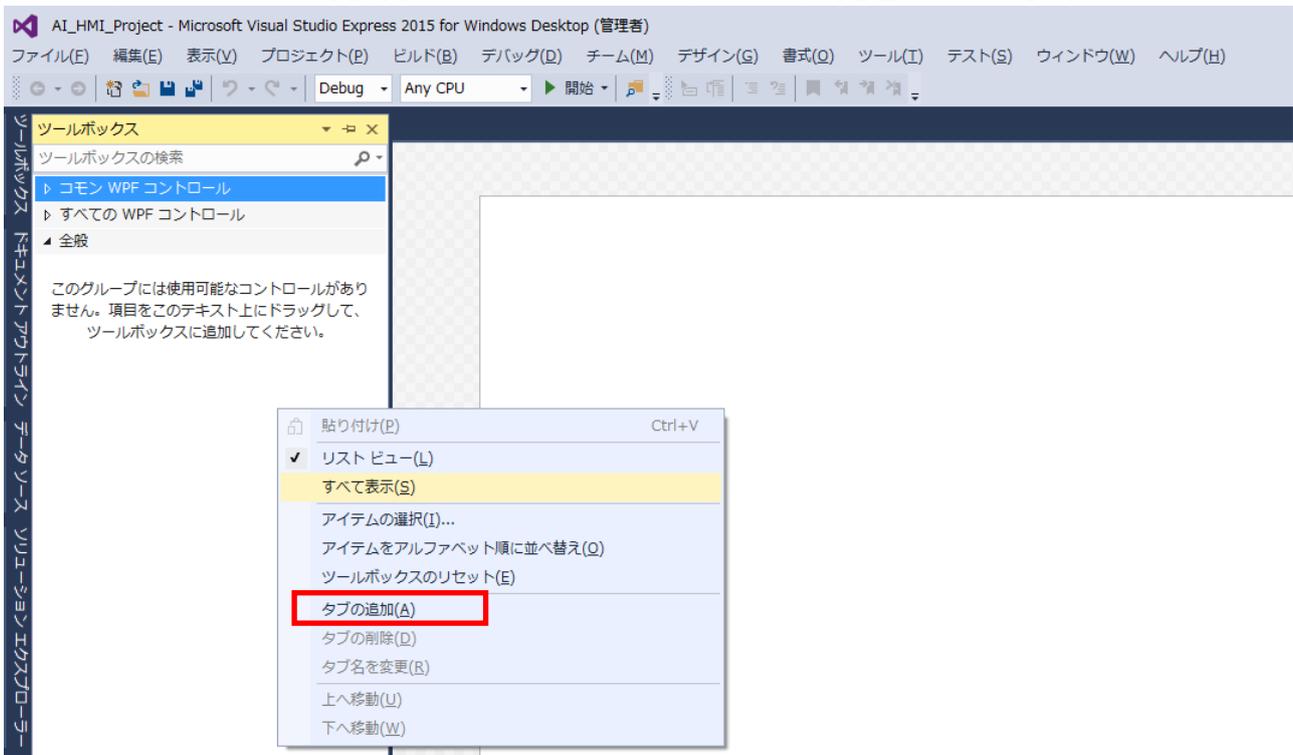


図 3-1-5 タブの追加

- ⑥ タブが追加され、名前を入力ができるようになるので「AI-HMI コントロール」という名称で登録してください。

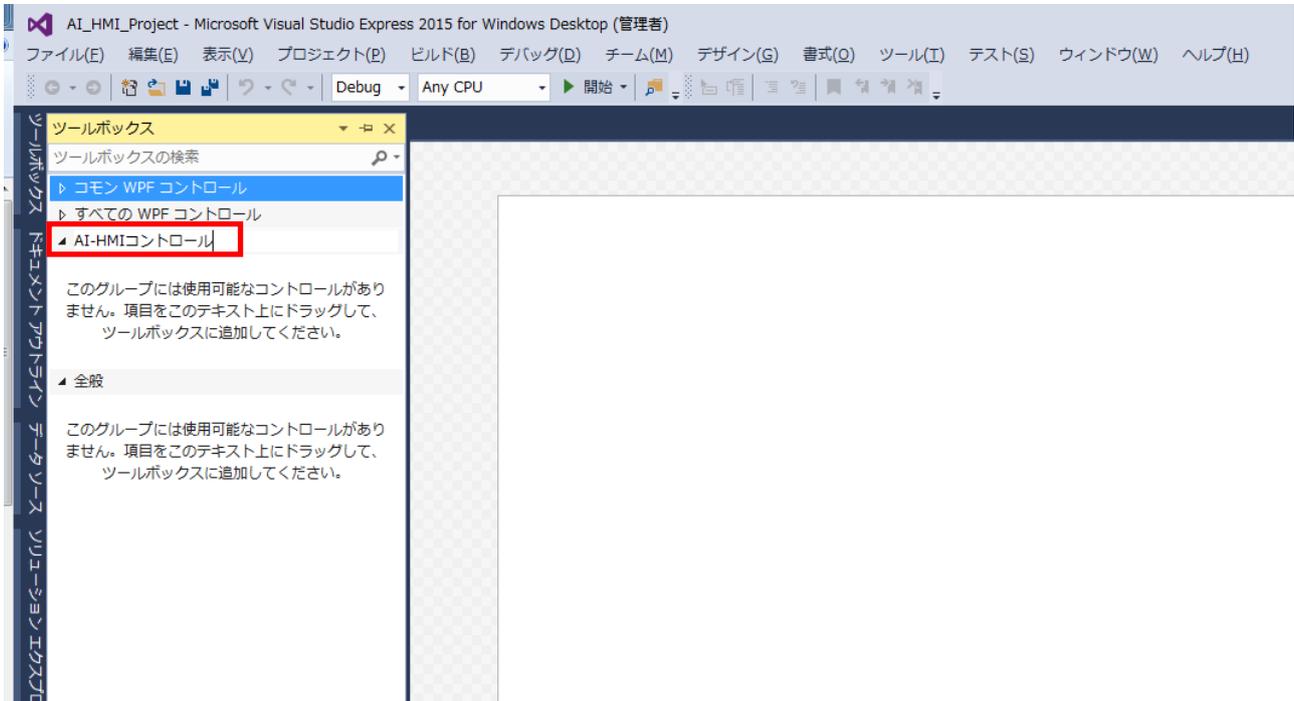


図 3-1-6 タブの名称入力

- ⑦ 「AI-HMI コントロール」タブを右クリックし、「アイテムの選択」を選択してください。

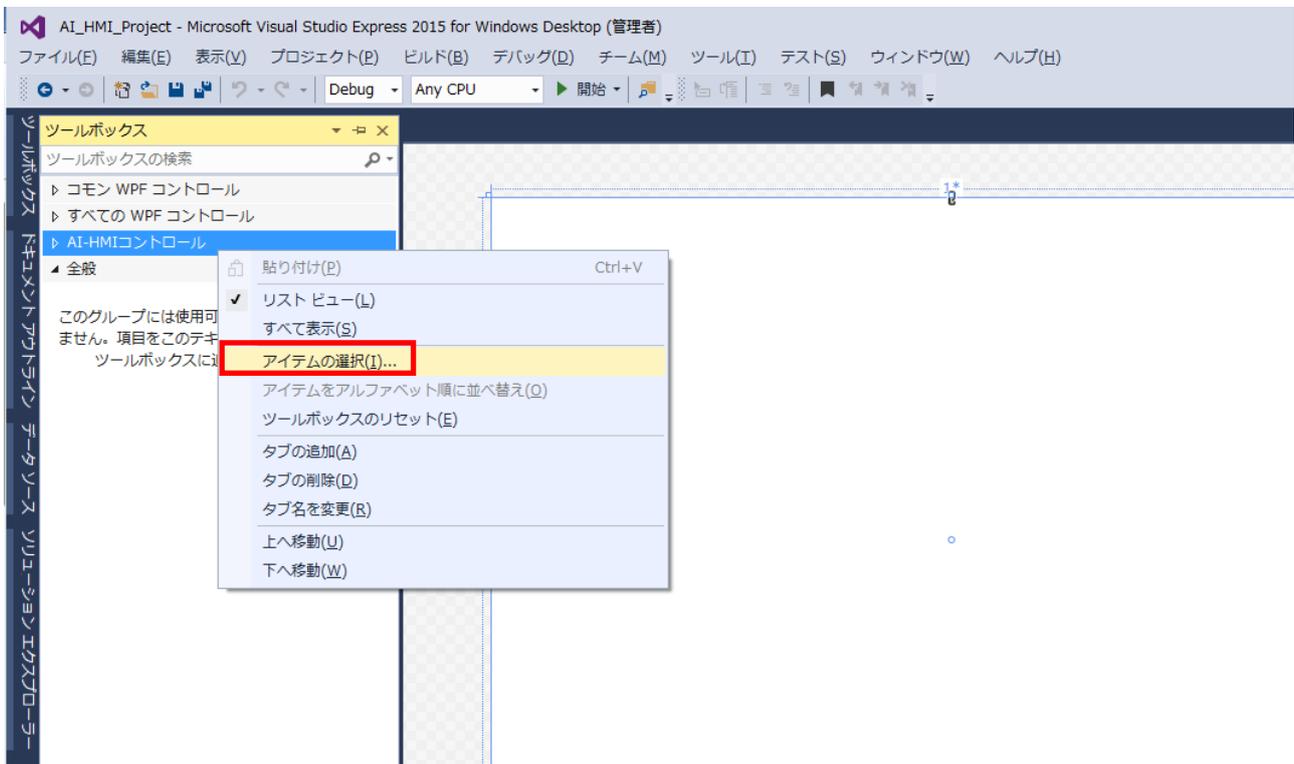


図 3-1-7 アイテムの追加

- ⑧ ツールボックスアイテムの選択画面が開きます。
データの読み込みが完了するまでしばらくお待ちください。(初回は数分程度かかることがあります)

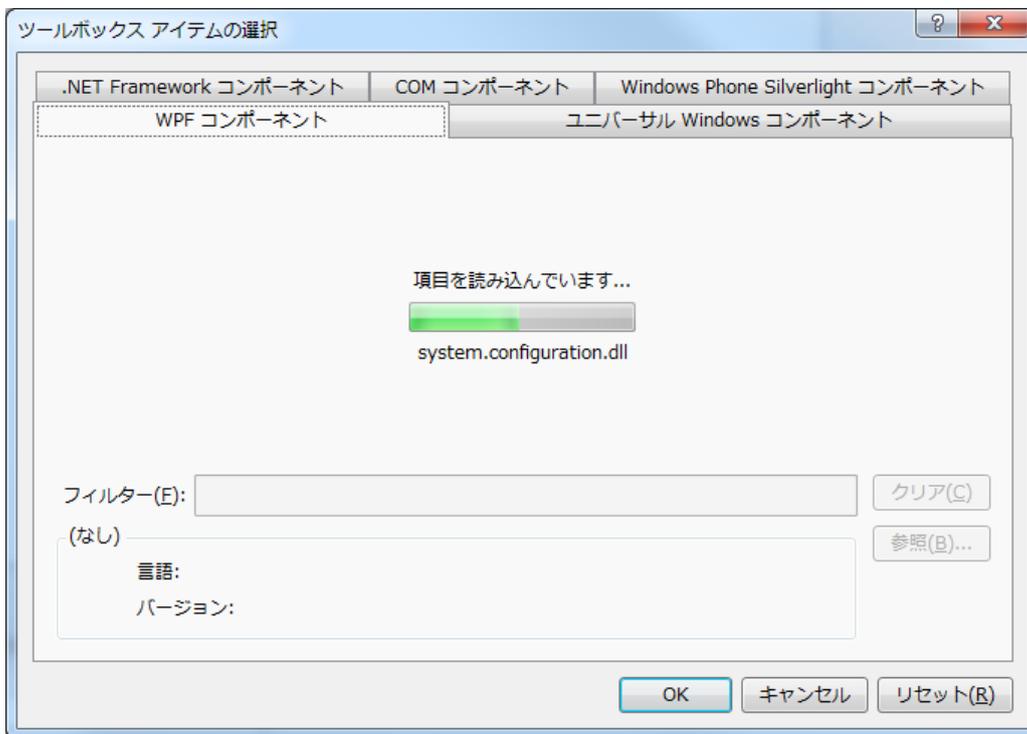


図 3-1-8 ツールボックスアイテムの選択画面

- ⑨ ツールボックスが表示されます。
「参照」ボタンをクリックしてください。

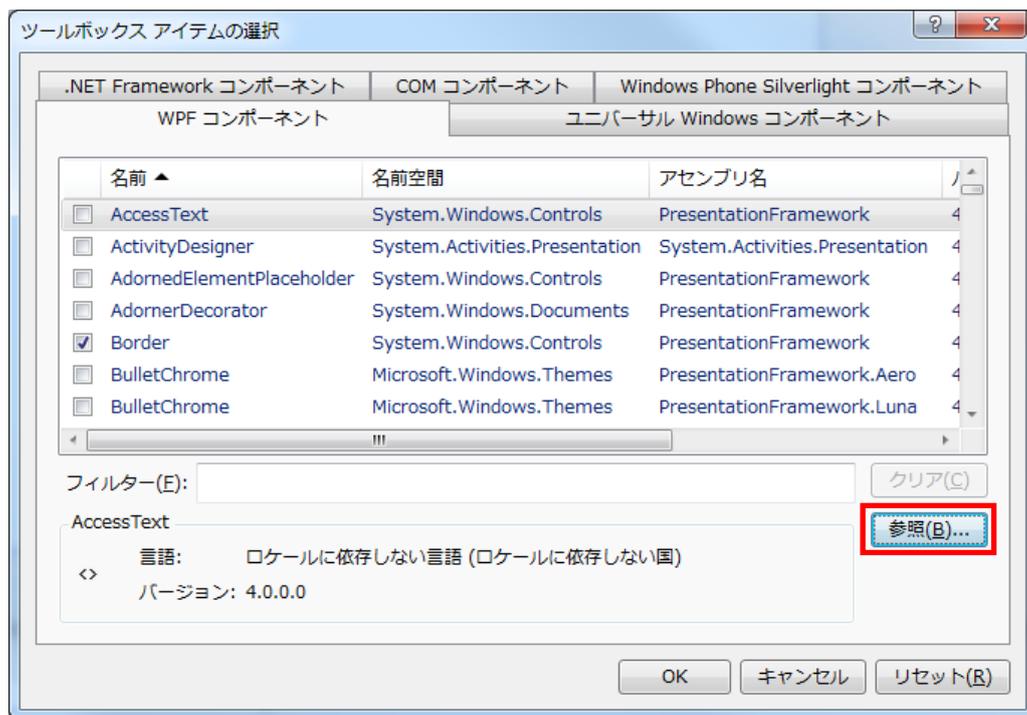


図 3-1-9 ツールボックスアイテムの追加

- ⑩ ファイルの選択画面が開きます。
以下のファイルを選択し、「開く」ボタンをクリックしてください。
C:\¥ALGOSYSTEM¥AI-HMI 開発環境用 SDK¥DLL¥AI_HMI_CTRL.dll

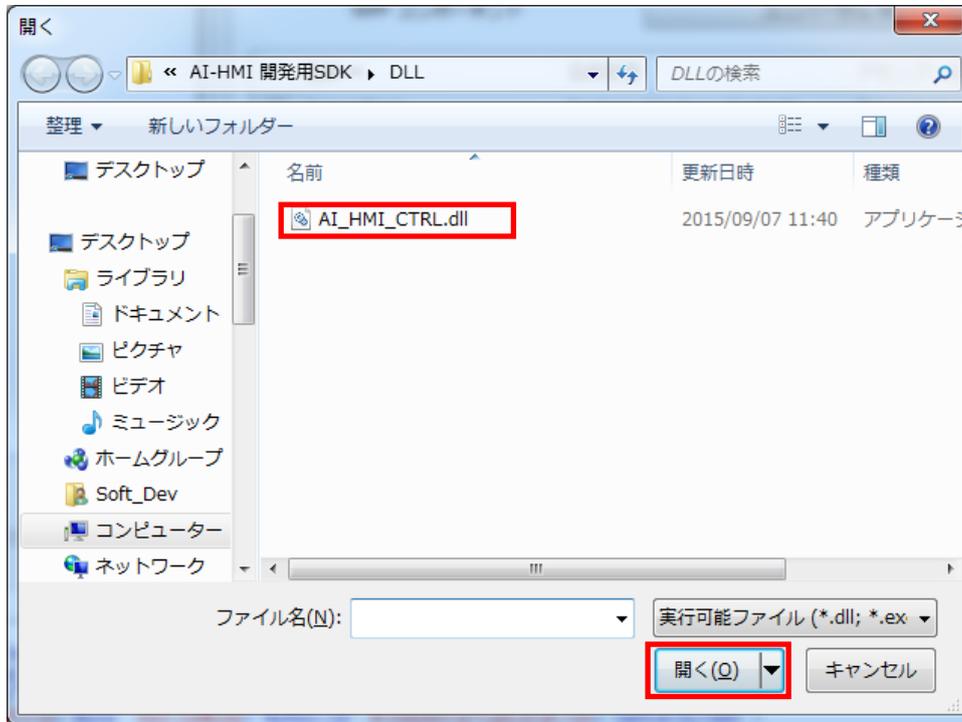


図 3-1-10 AI_HMI_CTRL.dll の追加

- ⑪ ツールボックスに AI-HMI のコンポーネントが追加されます。
各コンポーネントにチェックが入っていることを確認してから「OK」ボタンをクリックしてください。

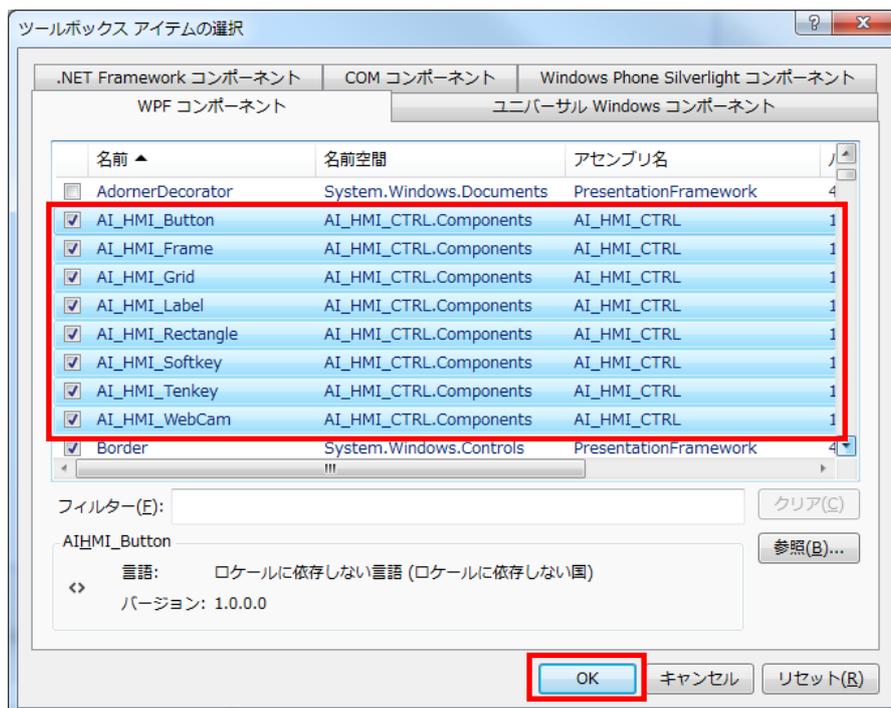


図 3-1-11 追加されたコンポーネントの確認

⑫ AI-HMI コントロールに各種コンポーネントが追加されていることを確認してください。

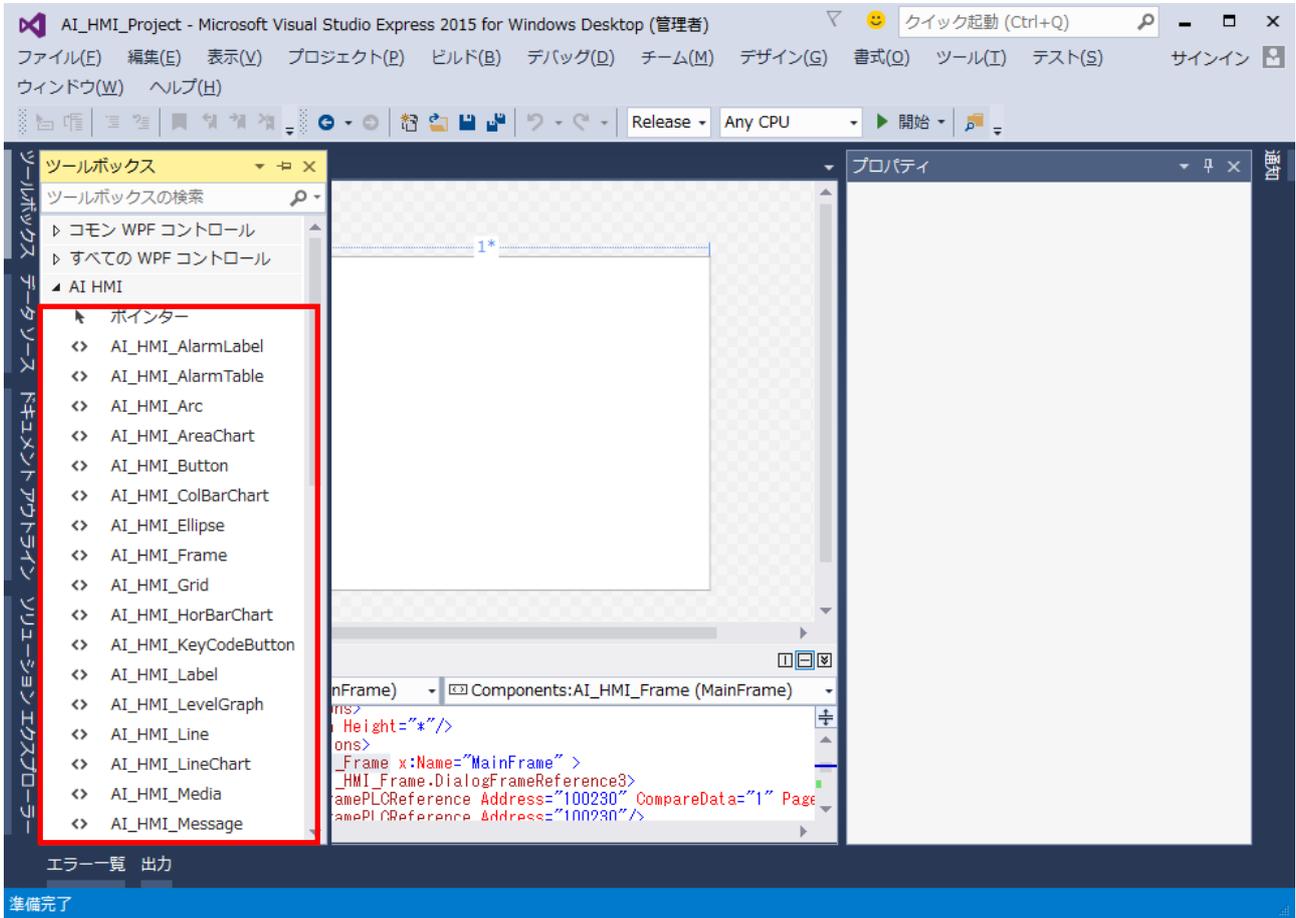


図 3-1-12 追加されたコンポーネントの確認

以上で開発環境へコンポーネントの組み込みを行う作業は完了です。

第4章 AI-HMIの構成について

AI-HMIはVisualStudio上で固有のコンポーネントを導入することで、容易に画面作成を行うことができます。

本章ではそれらのAI-HMI固有の構成について概略を説明します。

4-1 AI-HMIのページ構造について

AI_HMIでは、図4-1-1に示したような構造により、画面の切り替えおよびダイアログ表示を実現します。

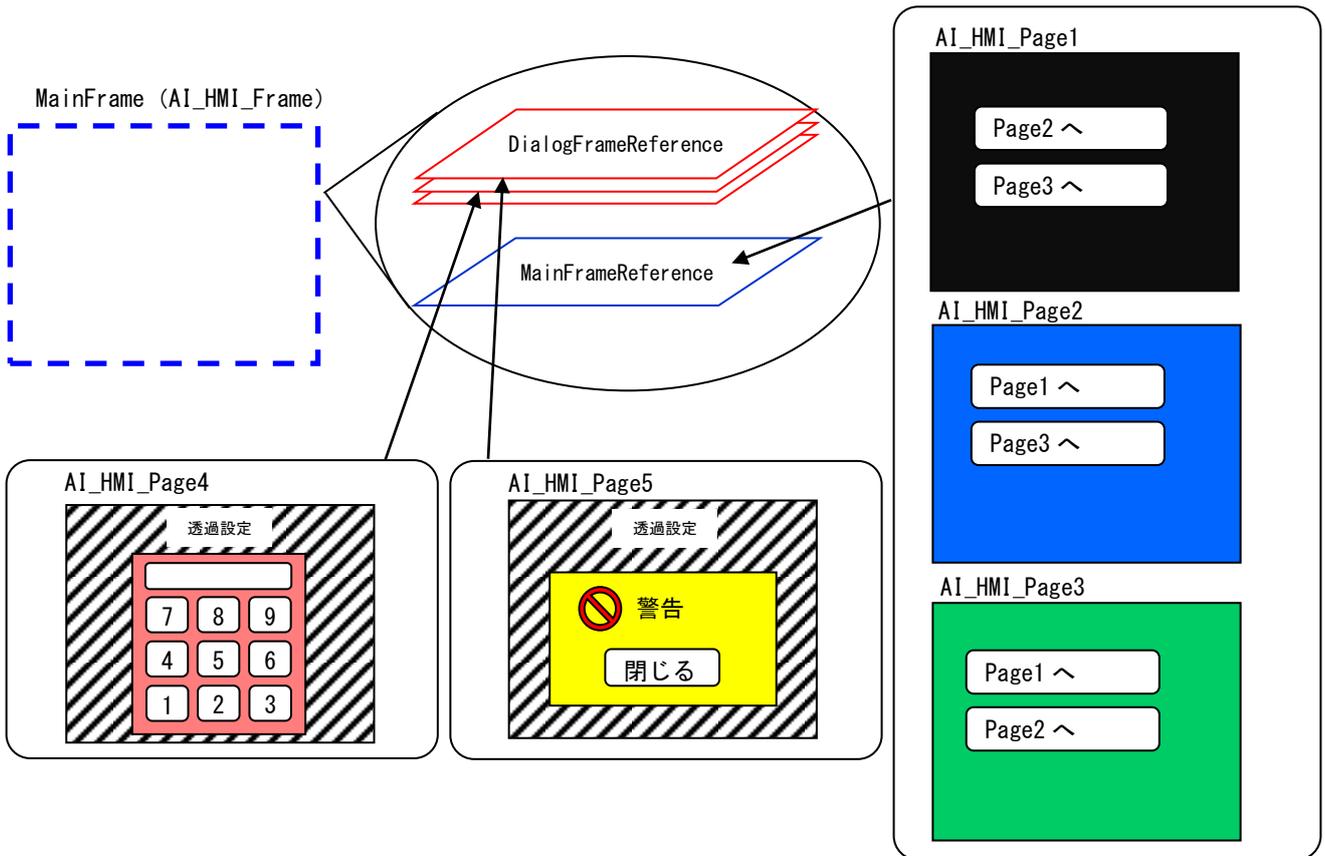


図 4-1-1 AI_HMI フレームとページの関係について

AI-HMIで画面を表示するためには、AI_HMI_Pageと呼ばれる画面を作成する必要があります。作成した画面は、AI_HMI_Frameと呼ばれるコンポーネント上に表示させることができます。

プロジェクト新規作成時に、MainFrame という名前の AI_HMI_Frame コンポーネントが配置されます。

AI_HMI_Frame コンポーネントはMainFrameReference プロパティと DialogFrameReference1~8 プロパティを持ちます。

MainFrameReference にページを登録し、表示する Page 番号を切り替える形で、画面を遷移させます。

DialogFrameReference にページを登録して表示することで、MainFrameReference で表示しているページの上に重ねてページを表示することができます。この時、MainFrameReference のページは見えなくなりますが、図 4-1-1 のように AI_HMI_Page4 を透過設定することで、MainFrameReference のページの上に警告ウインドウがポップアップしたようなイメージにすることが可能です。DialogFrameReference は一つの AI_HMI_Frame コンポーネントにつき 8 層まで設定できますので、図 4-1-1 のように AI_HMI_Page4 を表示しながら AI_HMI_Page5 を表示することも可能です。

4-2 カスタムコンポーネント

現在 AI-HMI で利用できるコンポーネントの一覧を表 4-2-1 に示します。

表 4-2-1 AI-HMI のコンポーネント一覧(1/2)

機能	コントロール名	主な機能
ボタン	AI HMI Button	クリックすることで共有メモリへの書き込みを行います。 また、共有メモリを参照して点灯/消灯の切り替えをしたり、共有メモリの値や文字列を表示することも可能です。
ボタン (ランプ機能拡張版)	AI HMI MultiLampButton	AI HMI Button の機能拡張版です。 点灯/消灯の切り替え条件を複数指定できます。
表示器	AI HMI Label	共有メモリの値や文字列を詳細な条件付で表示します。
テキストボックス	AI HMI TextBox	文字列を入力できます。 共有メモリの値や文字列を表示したり、ユーザからの入力を共有メモリに書き込んだりすることが可能です。
画面呼出し	AI HMI Frame	作成したページ画面を表示します。 複数のページ画面を共有メモリの値によって切り替え表示することも可能です。
テンキー	AI HMI Tenkey	数値入力、パスワード入力を行います。
ソフト キーボード	AI HMI Softkey	英数字記号の入力を行います。
カスタムソフト キーボード用 ボタン	AI HMI KeyCodeButton	クリックすることで設定したキーコードを送信します。 AI HMI TextBox と組み合わせることで、オリジナルのソフトウェアキーボードを作成することが可能です。
イメージ 表示器	AI HMI Grid	画像ファイルを表示します。 複数の画像ファイルを共有メモリの値によって切り替え表示したり、アニメーションとして連続表示することが可能です。
カメラ キャプチャ	AI HMI WebCam	USB カメラでキャプチャした画像をリアルタイムで表示します。
描写部品	AI HMI Rectangle	線、矩形、円、楕円を描写します。
線描写	AI HMI Line	線を描写します。
多角線描写	AI HMI Polyline	多角線を描写します。
多角形描写	AI HMI Polygon	多角形を描写します。
円描写	AI HMI Elipse	円、楕円を描写します。
円弧	AI HMI Arc	円弧を描写します。
目盛り	AI HMI Scale	グラフなどに取り付ける目盛り線を描写します。
レシピ設定	AI HMI Recipe	レシピの読み込みと書き込みをします。
再生	AI HMI Media	動画ファイルや音声ファイルなどのメディアファイルを再生します。
メッセージ表示	AI HMI Message	テキストファイルの内容を表示します。
スライダー	AI HMI Slider	スライダーにより、値を変更することが可能です。
セクター	AI HMI Selector	ドロップダウンリストから値を選択することが可能です。
表	AI HMI Table	複数のデータを表形式で表示します。
レベルグラフ (メータ)	AI HMI LevelGraph	レベルグラフを表示します。
メータグラフ (メータ)	AI HMI MeterGraph	メータグラフを表示します。
アラームラベル	AI HMI AlarmLabel	アラームが発生したとき、メッセージを表示します。
アラームテーブル	AI HMI AlarmTable	発生したアラームに関する情報を表示します。
サンプリング データ表示	AI HMI SamplingTable	サンプリングしたデータを表形式で表示します。

表 4-2-1 AI-HMI のコンポーネント一覧(2/2)

機能	コントロール名	主な機能
トレンドグラフ	AI HMI TrendGraph	サンプリングしたデータをトレンドグラフ形式で表示します。
縦棒グラフ	AI HMI ColBarChart	縦棒グラフを表示します。
横棒グラフ	AI HMI HorBarChart	横棒グラフを表示します。
エリアグラフ	AI HMI AreaChart	エリアグラフを表示します。
折れ線グラフ	AI HMI LineChart	折れ線グラフを表示します。
散布グラフ	AI HMI ScatterChart	散布グラフを表示します。
円グラフ	AI HMI PaiChart	円グラフを表示します。
積立 縦棒グラフ	AI HMI StkColBarChart	縦に積み上がった縦棒グラフを表示します
積立 横棒グラフ	AI HMI StkHorBarChart	横に積み上がった横棒グラフを表示します
積立 100% 縦棒グラフ	AI HMI Stk100ColBarChart	全体を 100%とした割合の縦棒グラフを表示します。
積立 100% 横棒グラフ	AI HMI Stk100HorBarChart	全体を 100%とした割合の横棒グラフを表示します。
トリガー アクション	AI HMI Window	特定の変数の変化を検知して他の変数へ書き込みを行います。
アラーム設定	AI HMI Window	特定の変数の bit 変化を検知してアラームとして記録、表示します。
サンプリング 設定	AI HMI Window	特定の変数の値を一定周期毎に記録、表示します。
パスワード設定	AI HMI Window	ユーザ名とパスワードを入力することでユーザログインをできるようにします。
操作ログ	AI HMI Window	AI HMI Button の操作をログとして記録します。
タイム スケジュール	AI HMI Window	特定の時間に変数へ書き込みを行います。
座標取得	AI HMI Position	次にクリックする座標位置を取得します。

4-3 AI-HMI 用 INtime 共有メモリ

AI-HMI で変数としてデータを共有メモリまたは OPC 変数から参照することで、表示や設定、画面の切り替え、コンポーネントの移動等を行います。

共有メモリの使用方法は『第13章 AI-PLC開発環境での使い方(共有メモリ)』を参照ください。

OPC 変数の使用方法は『第12章 AI-PLC開発環境での使い方(OPC変数)』を参照ください。

本章では共有メモリに関する詳細を解説します。

変数には BOOL データ、INT データ、UINT データ、DINT データ、UDINT データ、REAL データ、LREAL データ、文字列データがあります。データタイプの説明を表 4-3-1 に示します。

表 4-3-1 データタイプ

分類	型名	サイズ【バイト】	値の範囲	開始アドレス
バイナリ	BOOL	1	0 or 1	70000
整数	INT	2	-32768~32767	100000
	UINT	2	0~65535	
	DINT	4	-2147483648~2147483647	
	UDINT	4	0~4294967294	
実数	REAL	4	-3.40282347e+38 ~ -1.40129846e-45、 0、 1.40129846e-45 ~ 3.40282347e38、 +∞ / -∞	100000
	LREAL	8	-1.79769313486231e+308 ~ -2.22507385850720e-308、 0、 2.22507385850720e-308 ~ 1.79769313486231e+308、 +∞ / -∞	
文字列	STRING	1~32766 (標準 80※1)	-	140000

※1: AI-PLC の STRING 型のデフォルト文字数が 80 バイトとなります。詳細は後述します。

表 4-3-1 に記述されたデータタイプ毎に共有メモリは区切られます。共有メモリのマッピングを図 4-3-1 に示します。

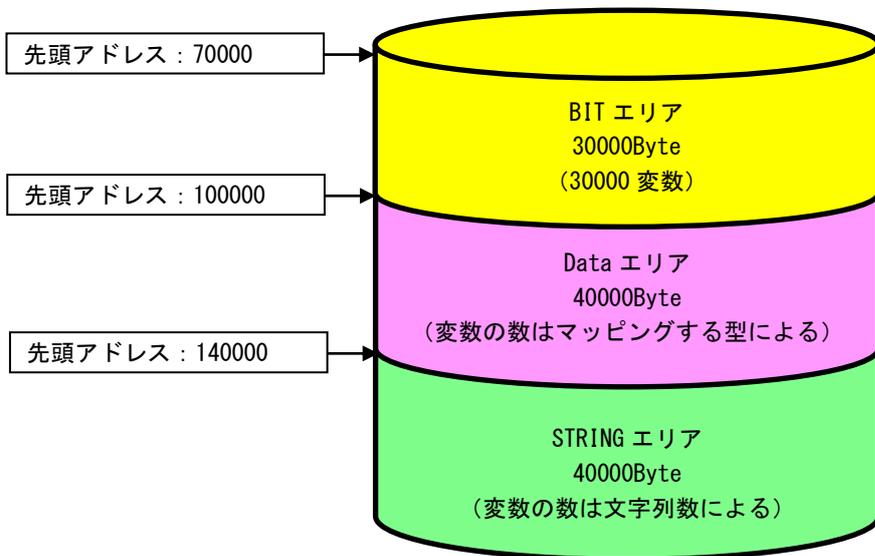


図 4-3-1 AI-HMI 共有メモリマッピング

AI-PLCの共有メモリの仕様上AI_HMI用共有メモリの先頭アドレスは70000からスタートされます。

各コンポーネントから、データを参照または設定するための変数の指定は、データタイプ+アドレスによって決定されます。アドレスはオフセット形式です。AI-PLC側のアドレス設定と、AI-HMI側のアドレス設定の対応を図4-3-2に示します。

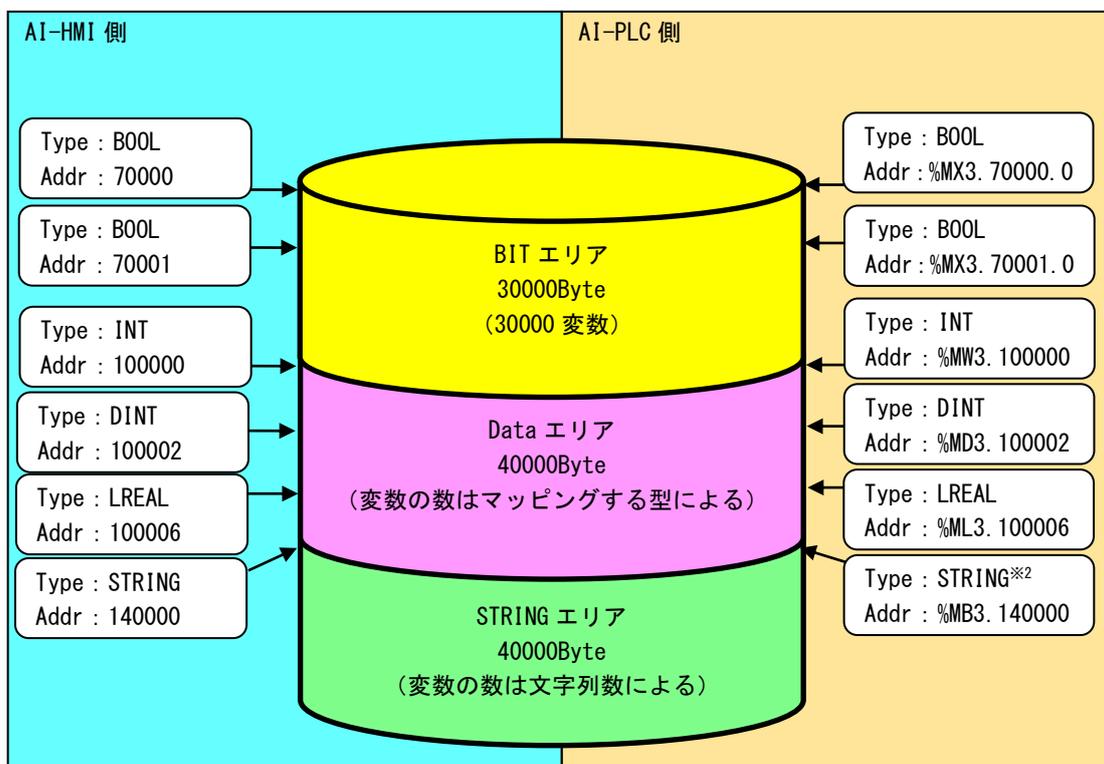


図4-3-2 共有メモリデータ型とアドレス指定対応図

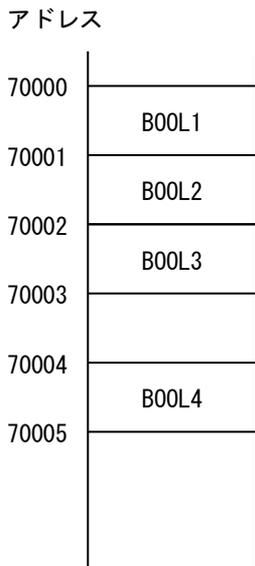
※2: AI-PLCで採用しているMULTIPROGのSTRING型はヘッダが4バイト先頭に付くほか、NULL文字1バイトが最後尾につくため、STRING型のサイズは最大文字列格納サイズ(デフォルトでは80Byte)+5となります。

2Byte 文字列最大長 (80)	2Byte 現在の文字列長 (5)	1~32766Byte (標準 80Byte) 文字列データ (AIUEO)	1Byte NULL文字 (0)
-------------------------	-------------------------	--	------------------------

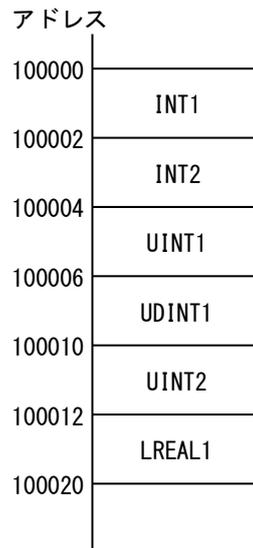
AI-HMI の共有メモリは INTtime の共有メモリが展開されている場合はその領域を使用します。
AI-HMI が INTtime の共有メモリが見つけれなかった場合は Windows の共有メモリを使用します。

実際に共有メモリに変数を登録する場合の例を以下に挙げます。

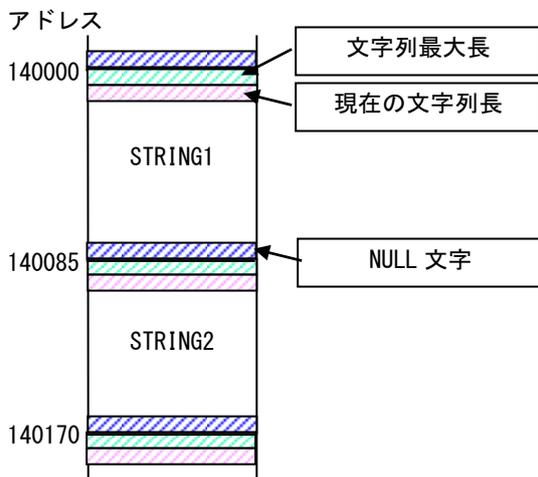
例1) Bit エリア



例2) Data エリア



例3) STRING エリア※3



※3 STRING 型は標準の 80Byte サイズの場合を示しています。

第5章 AI-HMI 画面アプリケーションの作成

完成した AI-HMI 開発環境を用いて実際にサンプルアプリケーションを作成します。

AI-HMI では変数として共有メモリの変数と OPC 変数を使用することができます。

本章では共有メモリの変数を使用した画面切り替えおよびテンキー表示アプリケーションの作成をします。

完成アプリケーション画面は下記になります。それぞれのページのボタンを押すことでページが切り替わります。また、テンキー表示ボタンを押すことで、テンキーが表示され値を入力できます。

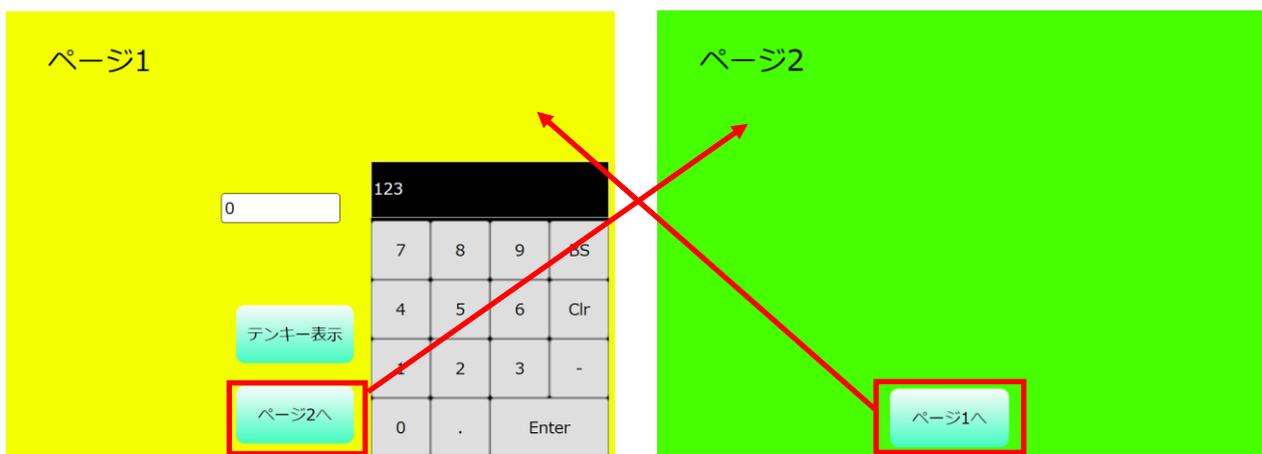


図 5-1. 画面切り替えアプリケーション

使用するコンポーネントは下記のとおりです。

AI_HMI_Button
AI_HMI_TenKey
AI_HMI_Label

使用する共有メモリのアドレスとデータ型、用途を表 5-1 に示します。

表 5-1 使用する共有メモリアドレス

アドレス	データ型	用途
100000	UINT 型	ページの切り替え変数
100002	UINT 型	ダイアログ表示切り替え変数
100008	UDINT 型	テンキー入力先変数
100012	UDINT 型	テンキー入力時のデータ型番号
140000	STRING 型	テンキー入力先の共有メモリアドレスを格納する変数

5-1 プロジェクトの新規作成

- ① 最初にサンプルアプリケーションのプロジェクトを新規に作成します。
Windows のスタートメニューから「すべてのプログラム – VS Express for Desktop」を実行してください。

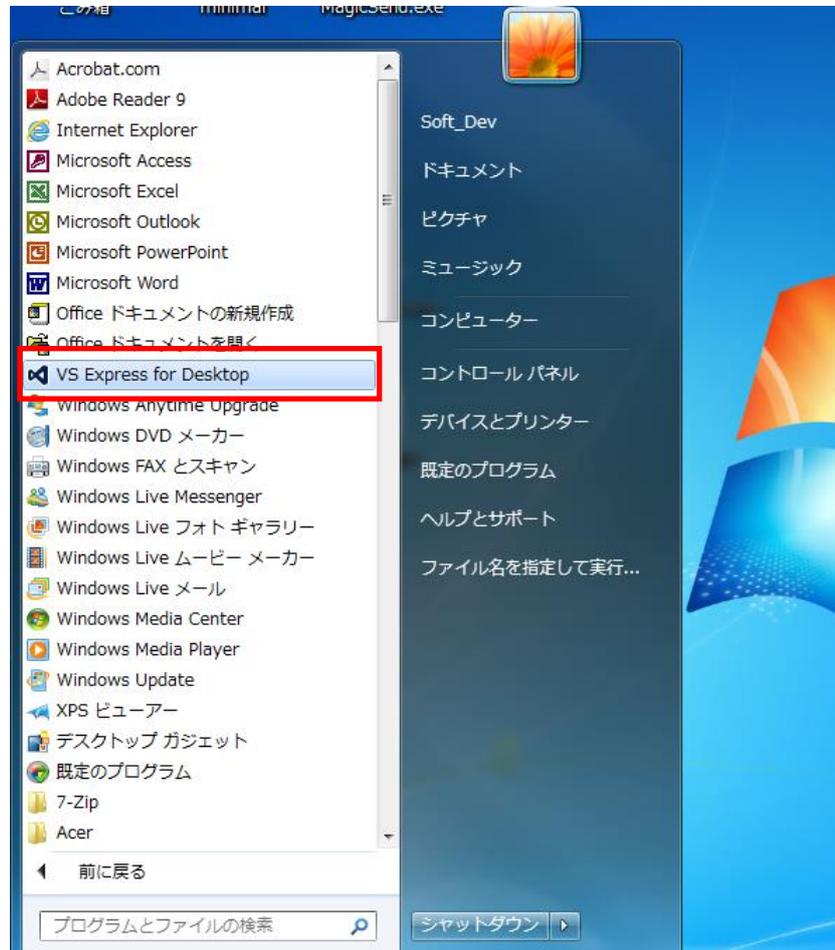


図 5-1-1 VisualStudioExpress の起動

- ② VisualStudioExpress が起動します。
「新しいプロジェクト」を選択してください。

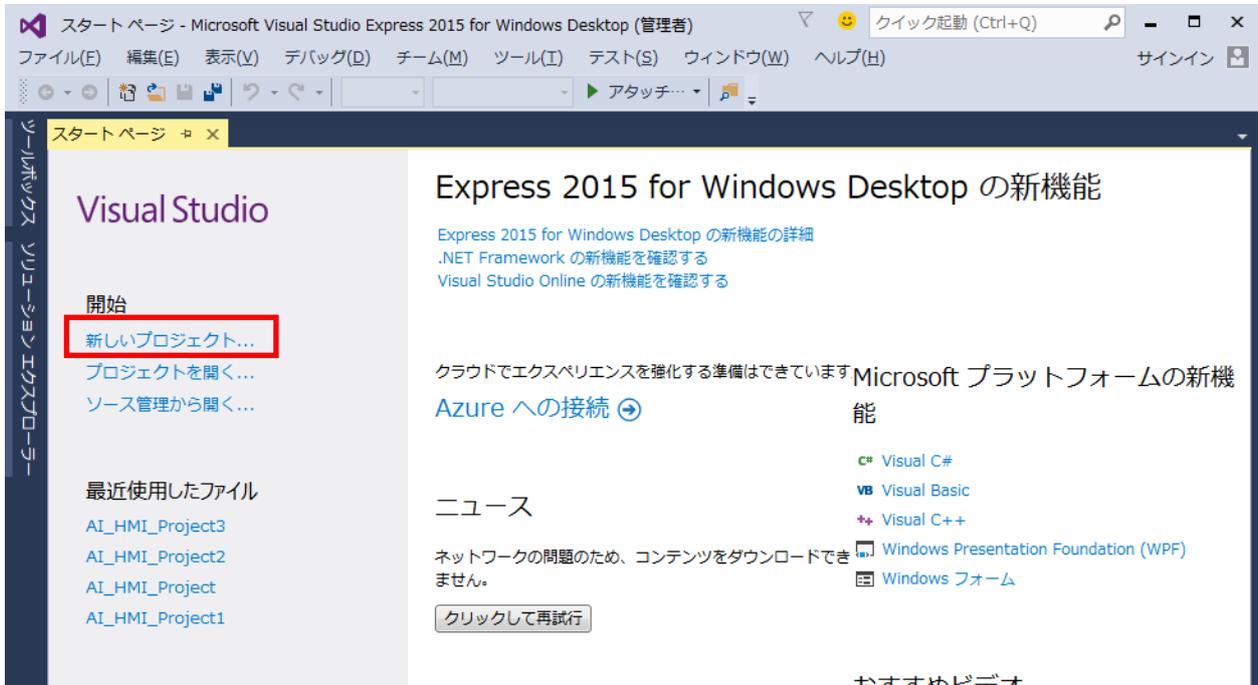


図 5-1-2 新しいプロジェクトの作成

- ③ 左ペインから「テンプレート – Visual C#」の項目を選択してください。
中央ペインにテンプレート一覧が表示されます。「AI_HMI_Project」を選択してください。
プロジェクト名、作業場所、ソリューション名を入力したあと「OK」ボタンをクリックしてください。

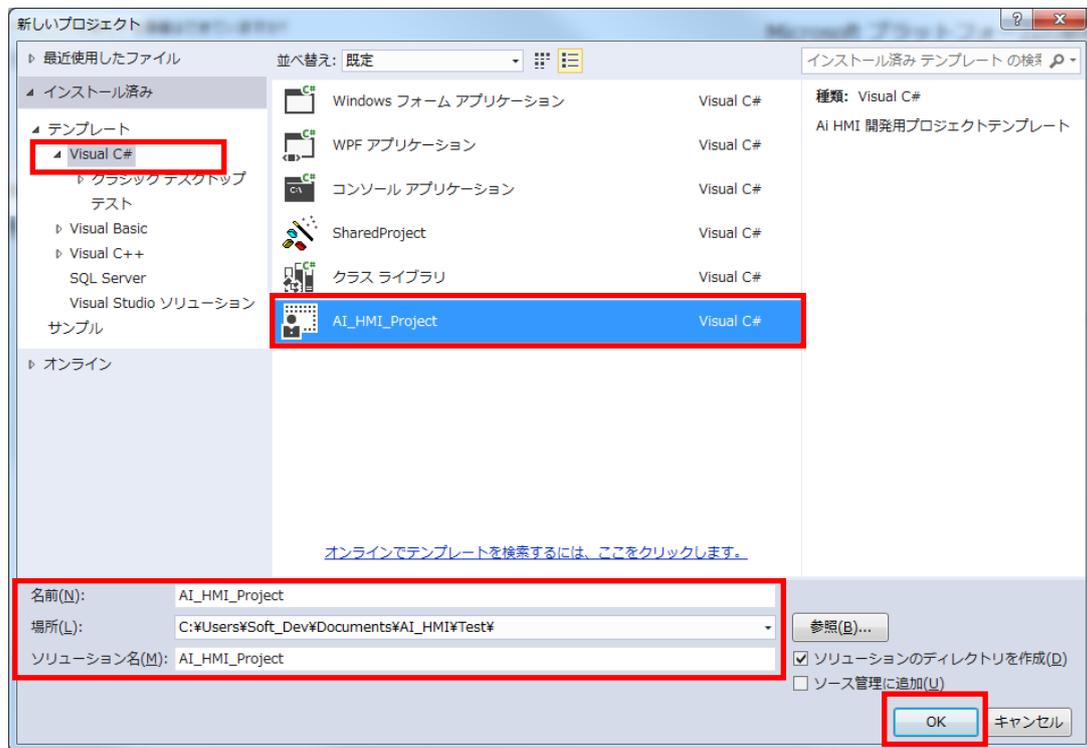


図 5-1-3 AI-HMI プロジェクトの作成

④ 新規のプロジェクト画面が開きます。

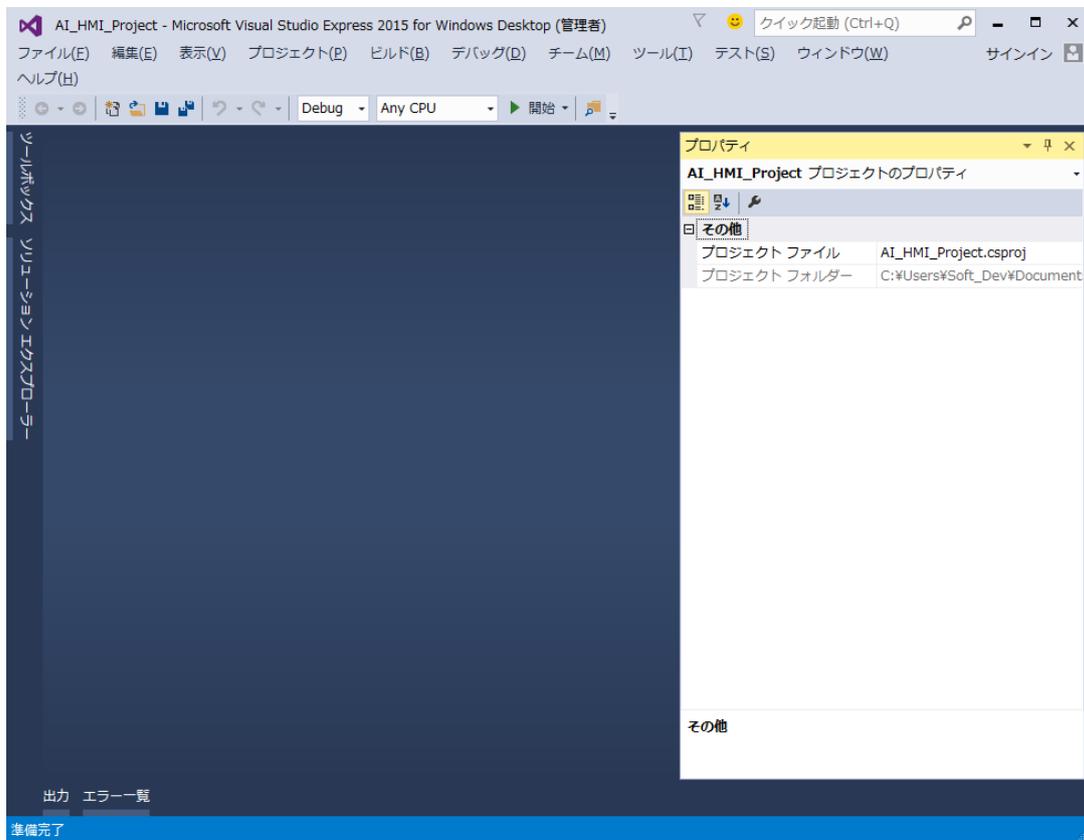


図 5-1-4 新規のプロジェクト画面

以上で新規プロジェクトの作成は完了です。

5-2 ページの作成

実際にページの作成をしていきます。

作成に当たって、本章では共有メモリを表 5-2-1 のような割り当てで使用します。

表 5-2-1 使用する共有メモリアドレス

アドレス	データ型	用途
100000	UINT 型	ページの切り替え変数
100002	UINT 型	ダイアログ表示切り替え変数
100008	UDINT 型	テンキー入力先変数
100012	UDINT 型	テンキー入力時のデータ型番号
140000	STRING 型	テンキー入力先の共有メモリアドレスを格納する変数

※ VisualStudio では画面作成時にデフォルトで画面下にソースが表示されていますが、AI-HMI では User 側は編集することなく画面の作成ができるよう設計されています。

これらのソースを編集することでより詳細な設計をすることは可能ですが、お客様がソースを直接編集した場合、AI-HMI は動作を保障できません。

基本的にソースの編集は行わないようにし、ソースの編集をする場合はお客様の責任で編集をしていただきますようお願いいたします。

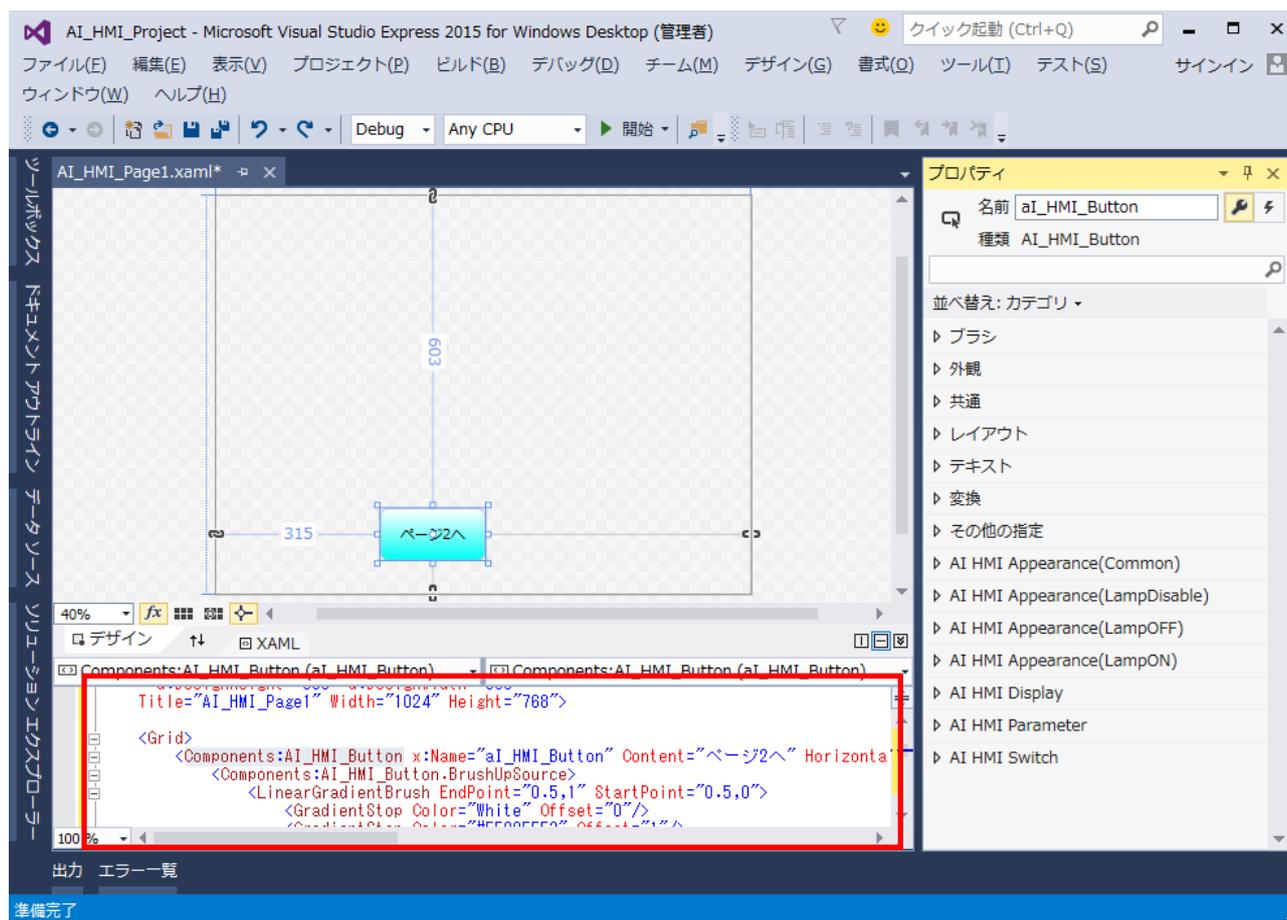


図 5-2-1 画面作成中のソース

- ① 各ページにボタンを配置し、それぞれのページに表示切替するサンプルアプリを作成します。
「ソリューションエクスプローラー」を開き、「AI_HMI_Page1.xaml」をダブルクリックしてください。

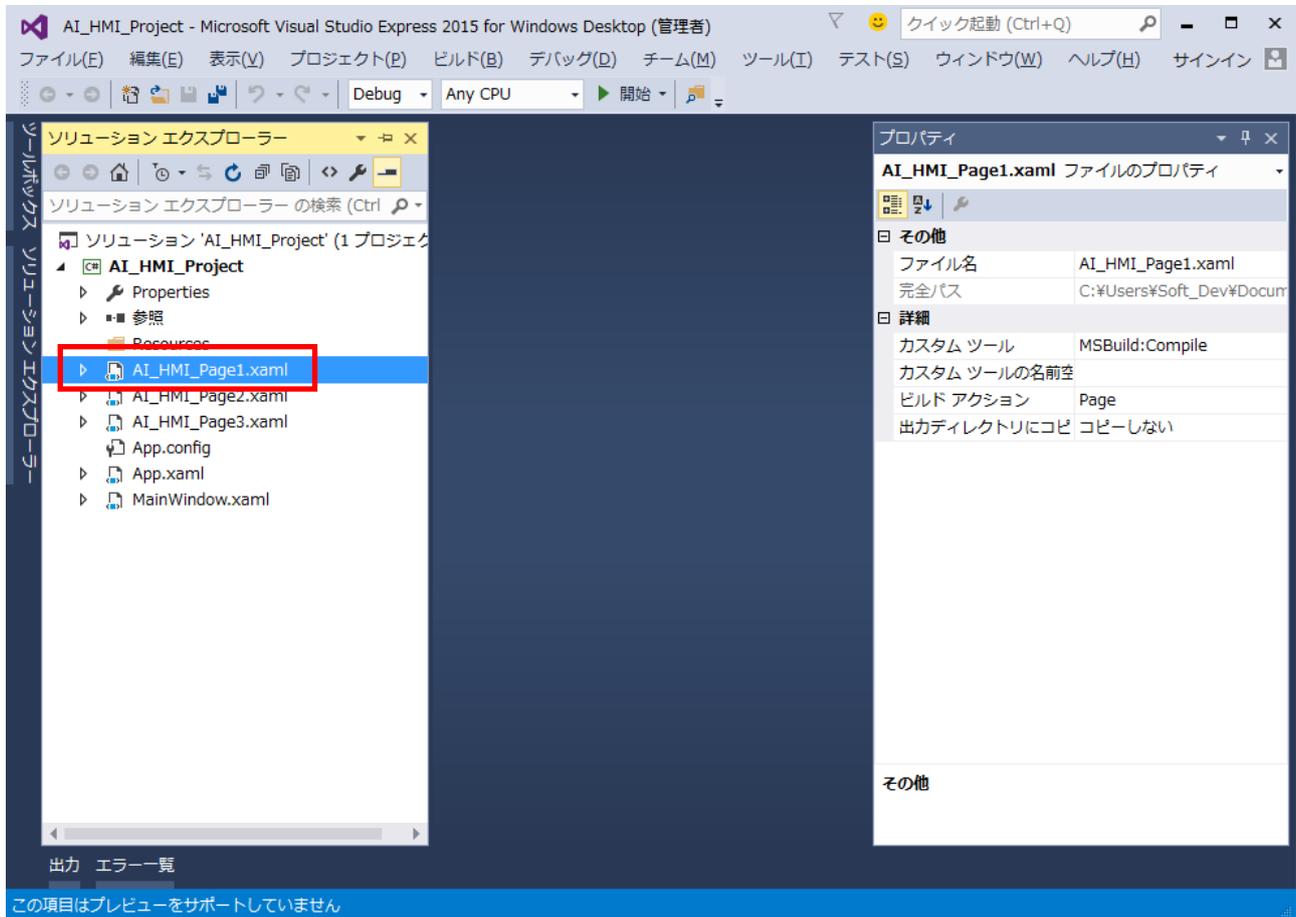


図 5-2-2 AI_HMI_Page1 の表示

② AI_HMI_Page1 画面が開きます。

「ツールボックス」から「AI_HMI_Button」を選択し、AI_HMI_Page1 の画面上に配置してください。

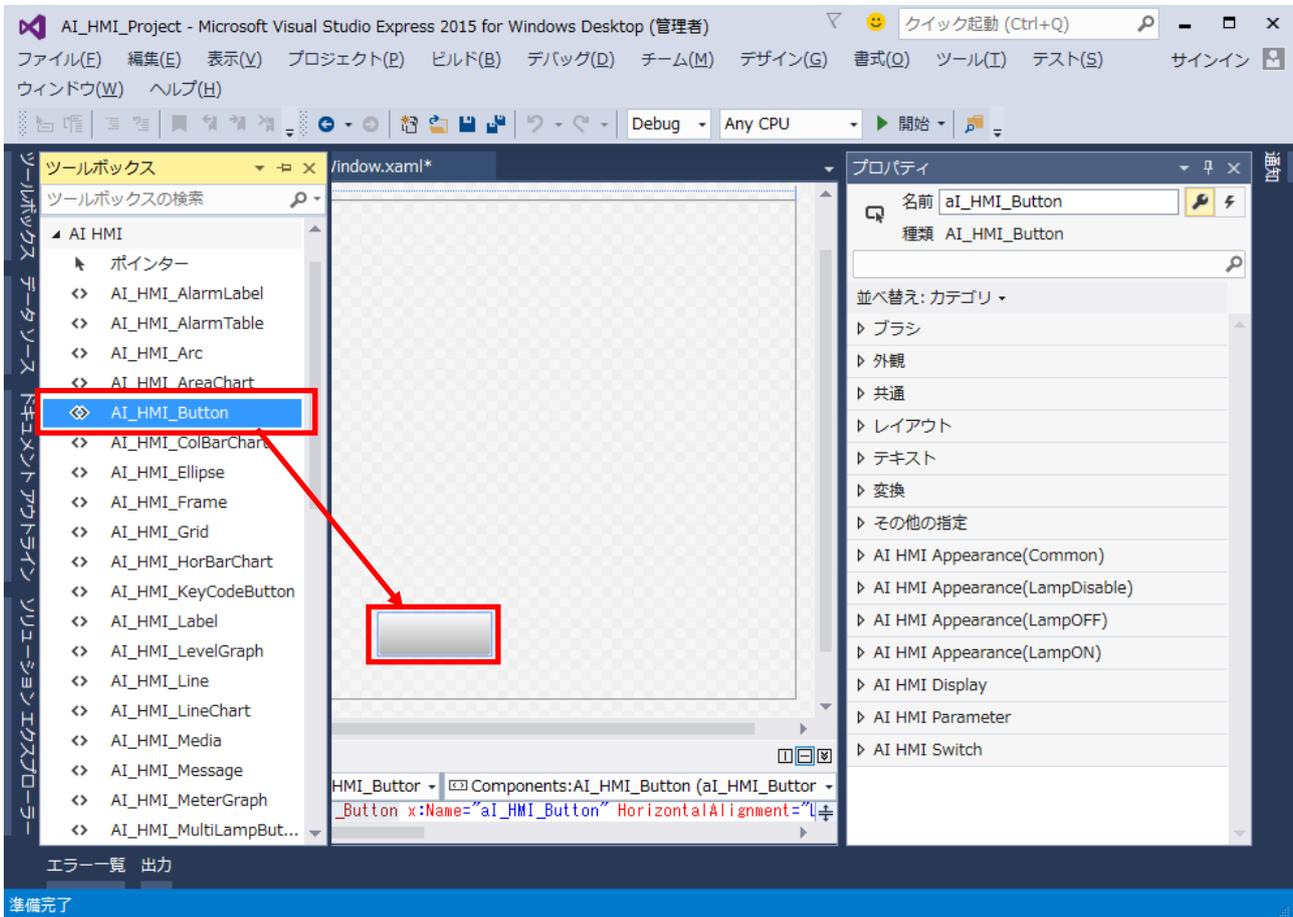


図 5-2-3 AI_HMI_Button の配置

各オブジェクトはツールボックスからAI_HMI_Buttonを選択した状態でAI_HMI_Page1上で矩形にドラッグすることで配置できます。

③ ボタンオブジェクトが配置されるので、プロパティを表 5-2-2 のように設定してください。

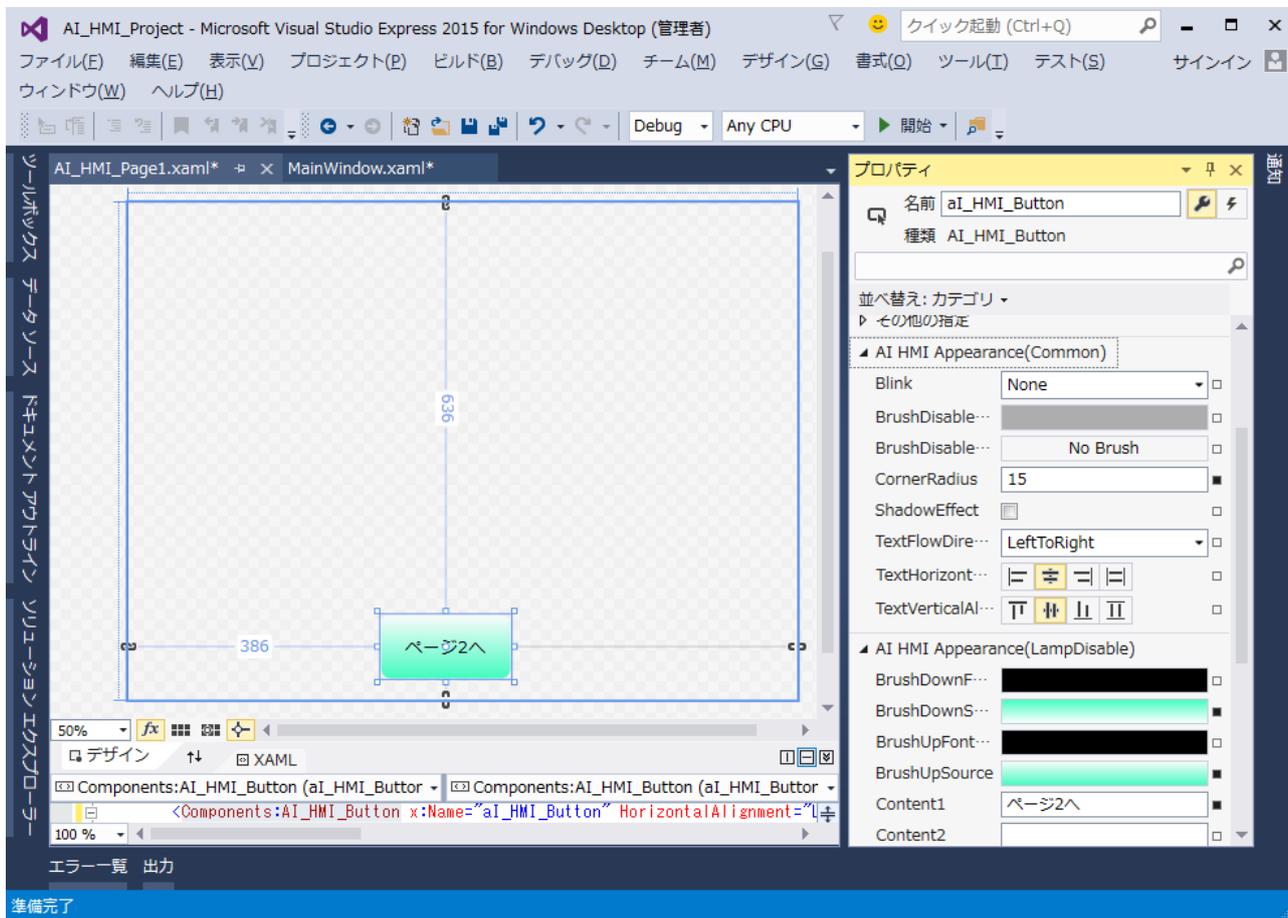


図 5-2-4 AI_HMI_Button のプロパティの変更

表 5-2-2 AI_HMI_Button のプロパティ

項目		設定値	備考
レイアウト	Width	200	ボタンの幅
	Height	100	ボタンの高さ
テキスト	(フォントサイズ)	20pt	表示する文字の大きさ
AI HMI Appearance (Common)	CornerRadius	15	ボタンの角の丸み
AI HMI Appearance (LampDisable)	BrushDownSource	任意の色	ボタンを押した時の画像
	BrushUpSource	任意の色	ボタンを離れた時の画像
	Content1	ページ 2 へ	ボタンに表示する文字列

- ④ ボタンが押されたときの挙動の設定をします。
ボタンプロパティの「AI HMI Switch – SwitchReference」のボタンをクリックしてください。

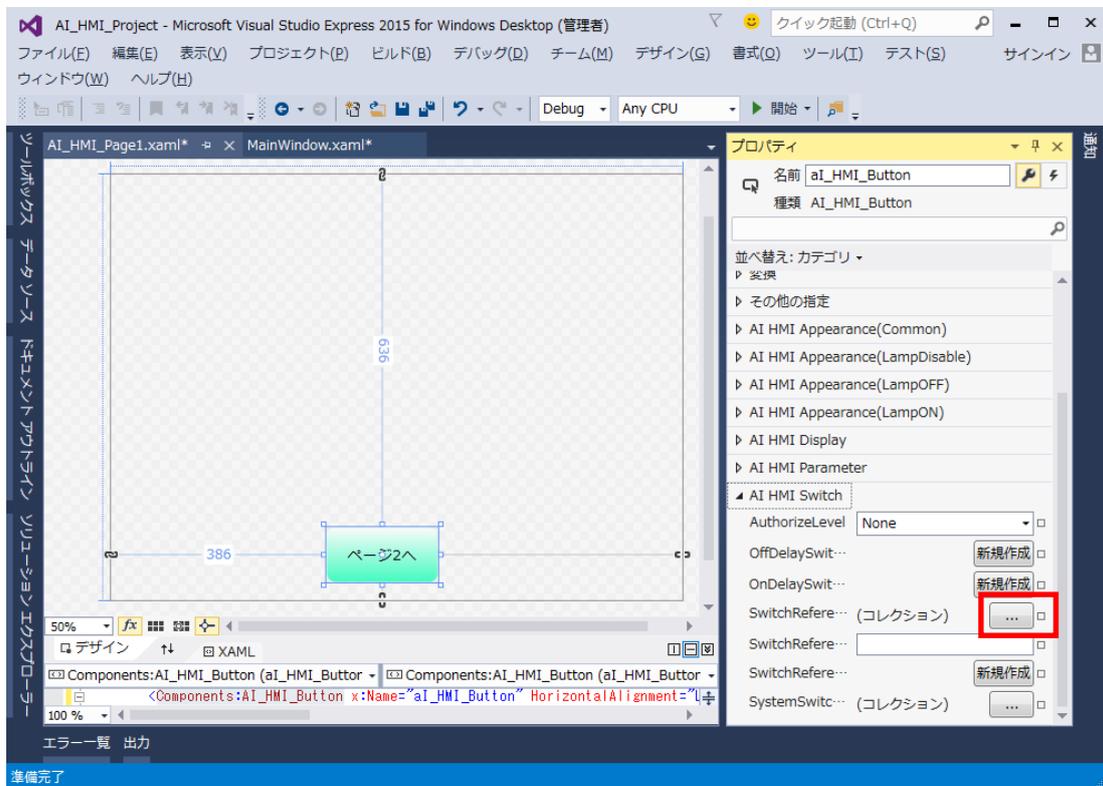


図 5-2-5 SwitchReference の変更

- ⑤ ボタンのコレクションエディタが開きます。「追加」ボタンをクリックしてください。

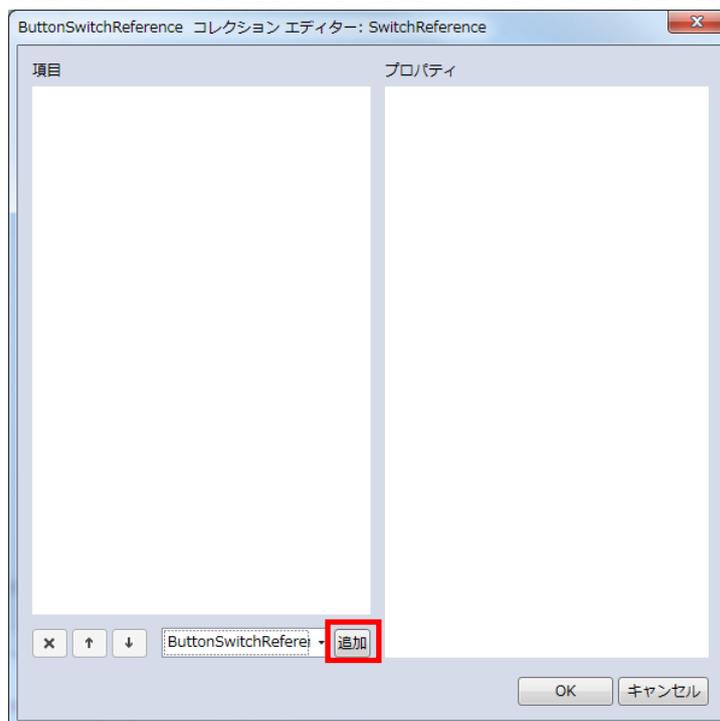


図 5-2-6 SwitchReference の変更

- ⑥ ButtonSwitchReference のプロパティが 1 項目追加されます。
表 5-2-2 のように設定をしてください。

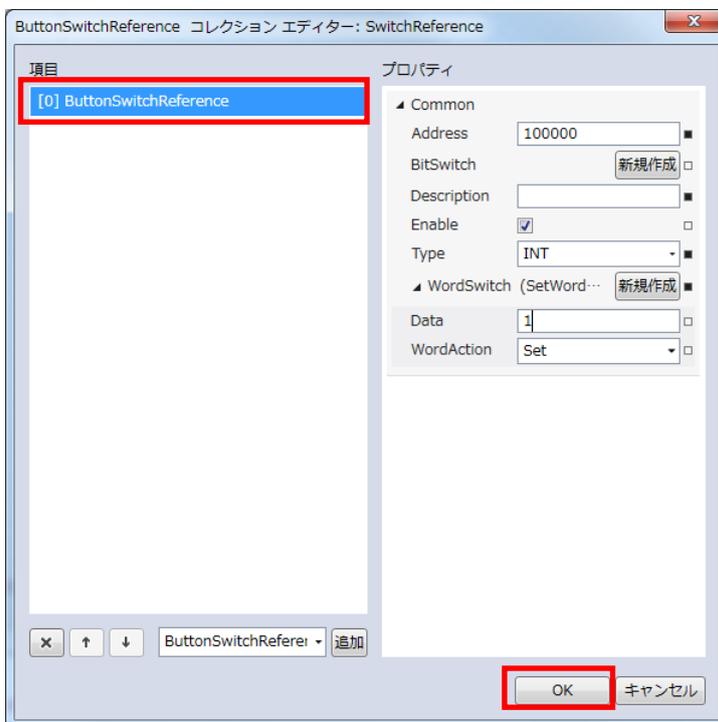


図 5-2-7 ButtonSwitchReference の変更

表 5-2-3 ButtonSwitchReference のプロパティ

項目	設定値	備考
Address	100000	共有メモリ上のアドレス
Type	INT	データタイプ
Data ※	1	ボタンが押されたときに書き込む値
WordAction ※	Set	ボタンが押されたときの挙動

※ WordSwitch の行の「新規作成」をクリックすることで表示します。

ボタンが押されたときの挙動は ButtonSwitchReference を設定することで制御できます。
上記の例の場合、「共有メモリアドレス 100000」に対して「INT 型変数」として「数値 1」を「書き込み(セット)」します。
これらの設定を変更することでボタンが押されたときの挙動として様々な動作を設定することができます。

また、ButtonSwitchReference の行を複数追加することで、一回のボタン押下に対して複数の挙動を設定することができます。

- ⑦ 先ほどのボタンとは別に文字列を表示するためのオブジェクトを配置します。
「ツールボックス」から「AI_HMI_Button」を選択し、AI_HMI_Page1 の画面上に配置してください。

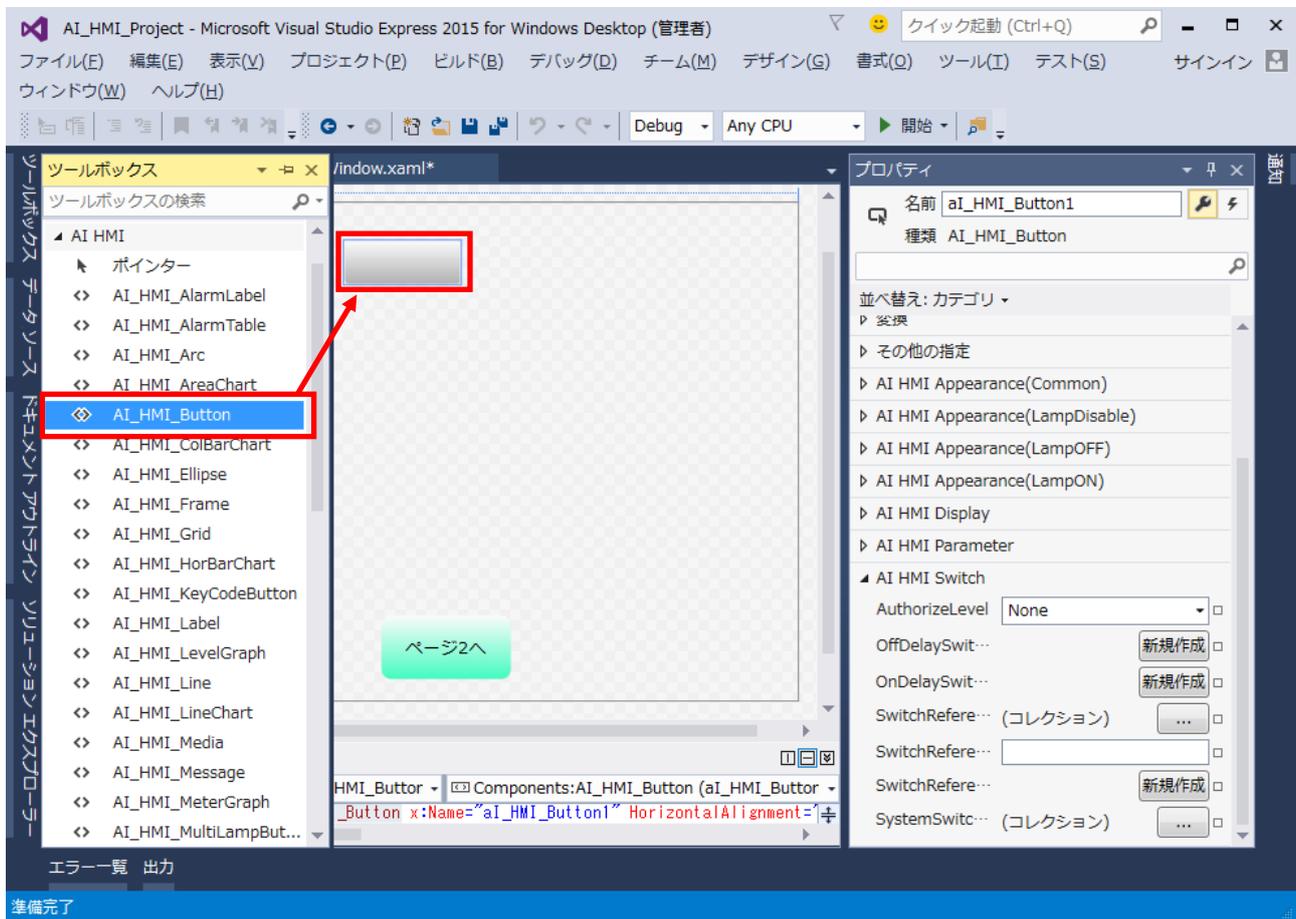


図 5-2-8 AI_HMI_Button の配置 (2 個目)

⑧ ボタンオブジェクトが配置されるので、プロパティを表 5-2-4 のように設定してください。

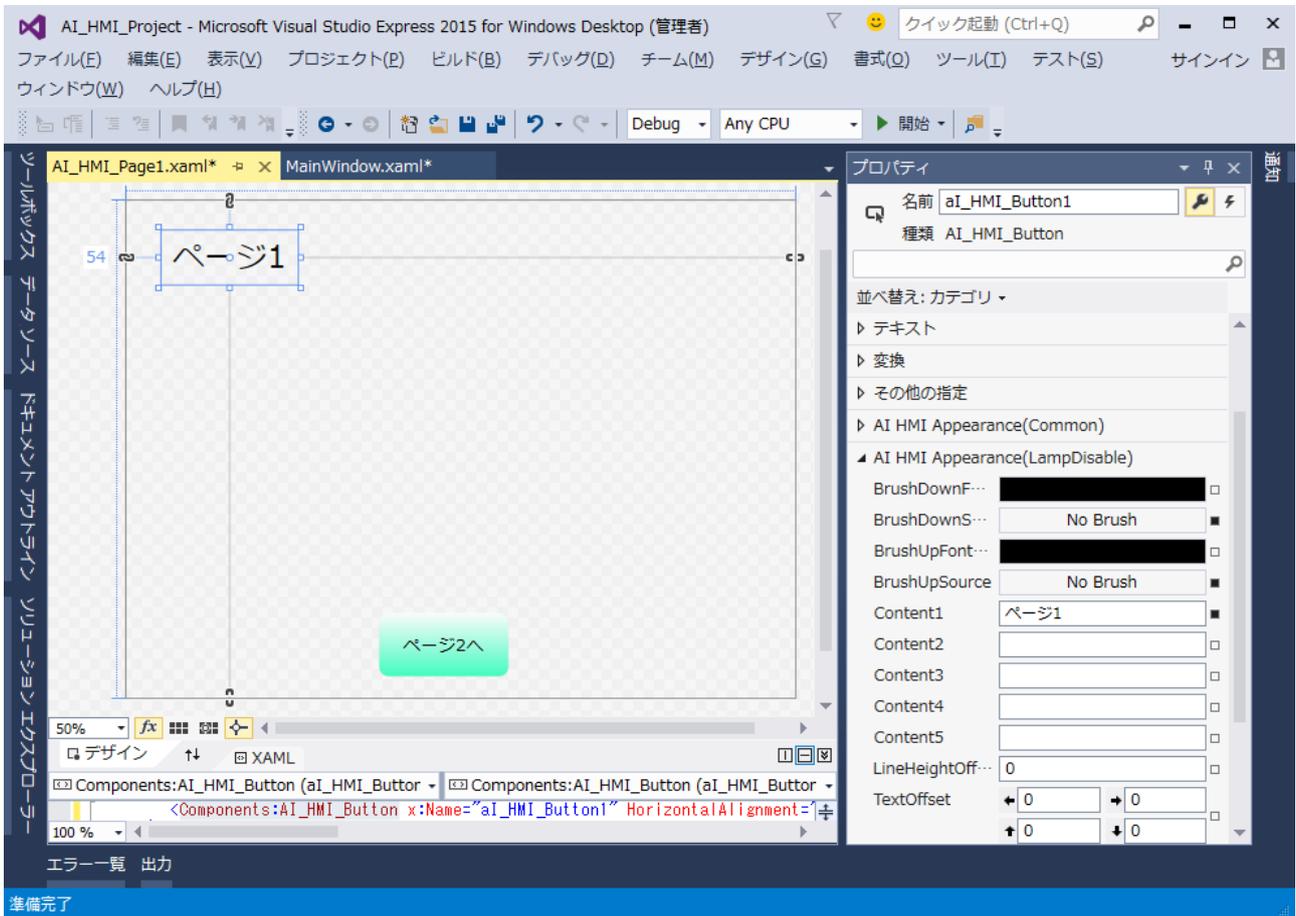


図 5-2-9 AI_HMI_Button(2 個目)のプロパティの変更

表 5-2-4 AI_HMI_Button(2 個目)のプロパティ

項目	設定値	備考
テキスト	(フォントサイズ)	36pt
AI HMI Appearance(LampDisable)	BrushDownSource	No Brush
	BrushUpSource	No Brush
	Content1	ページ 1

- ⑨ 背景画像の色を変更します。
背景部分をクリックし、表 5-2-5 のようにプロパティを変更してください。

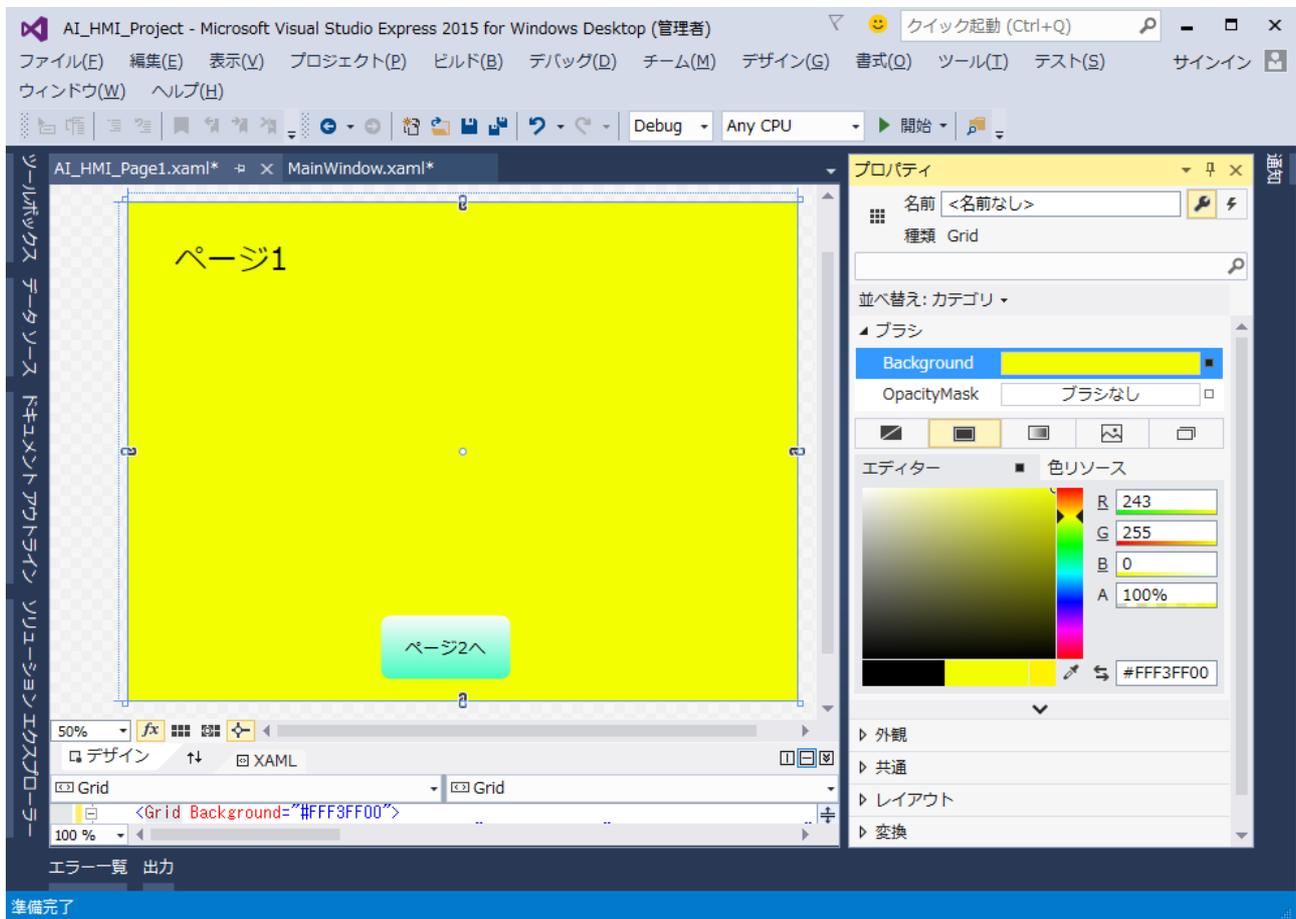


図 5-2-10 背景のプロパティの変更

表 5-2-5 背景のプロパティ

項目	設定値	備考
ブラシ	Background	任意の色 背景画像

- ⑩ 同様にして AI_HMI_Page2 を作成します。
「ソリューションエクスプローラー」を開き、「AI_HMI_Page2.xaml」をダブルクリックしてください。

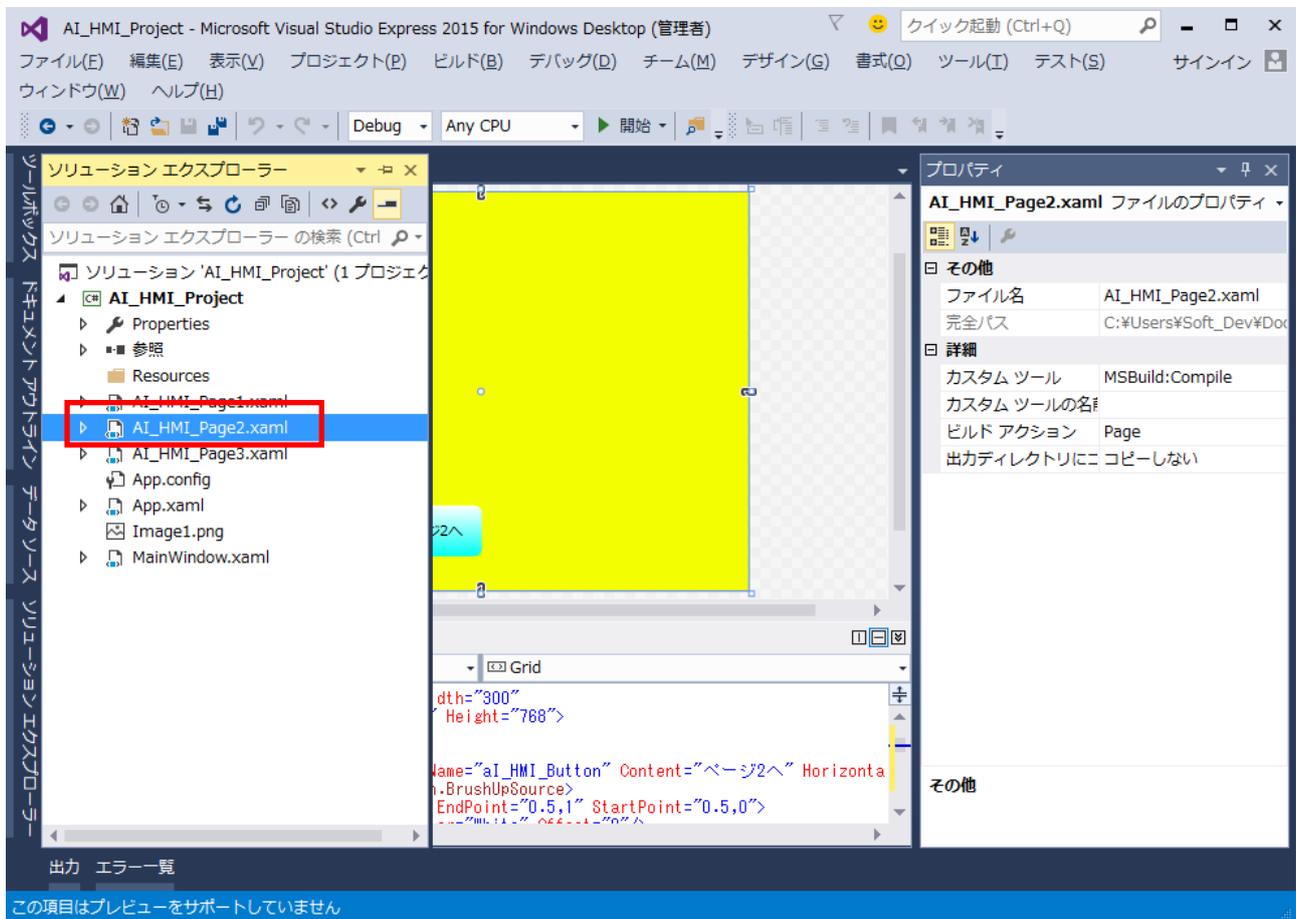


図 5-2-11 AI_HMI_Page2 の表示

① AI_HMI_Page2 画面が開きます。

「ツールボックス」から「AI_HMI_Button」を選択し、AI_HMI_Page2 の画面上に配置してください。

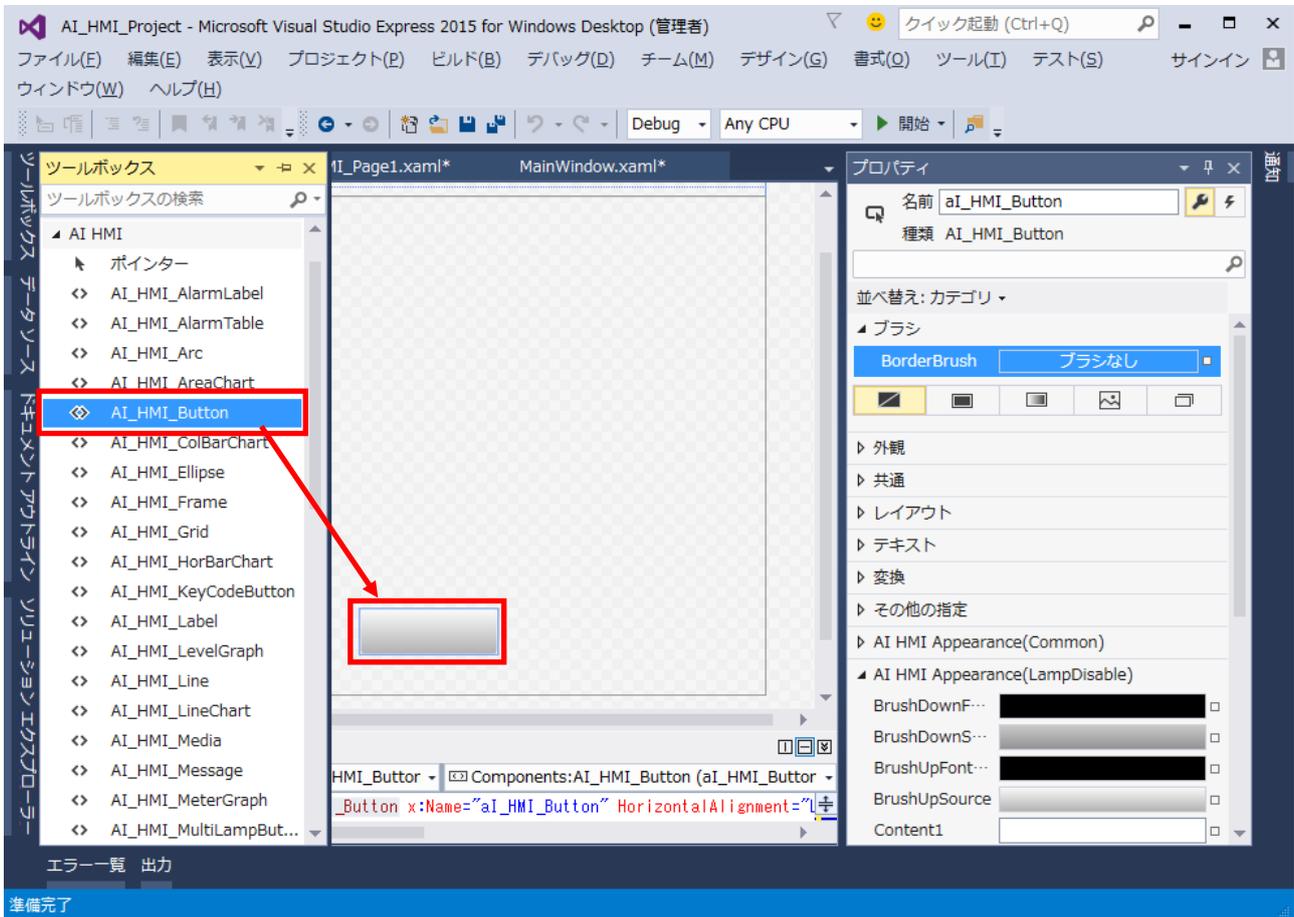


図 5-2-12 AI_HMI_Button の配置

⑫ ボタンオブジェクトが配置されるので、プロパティを表 5-2-6 のように設定してください。

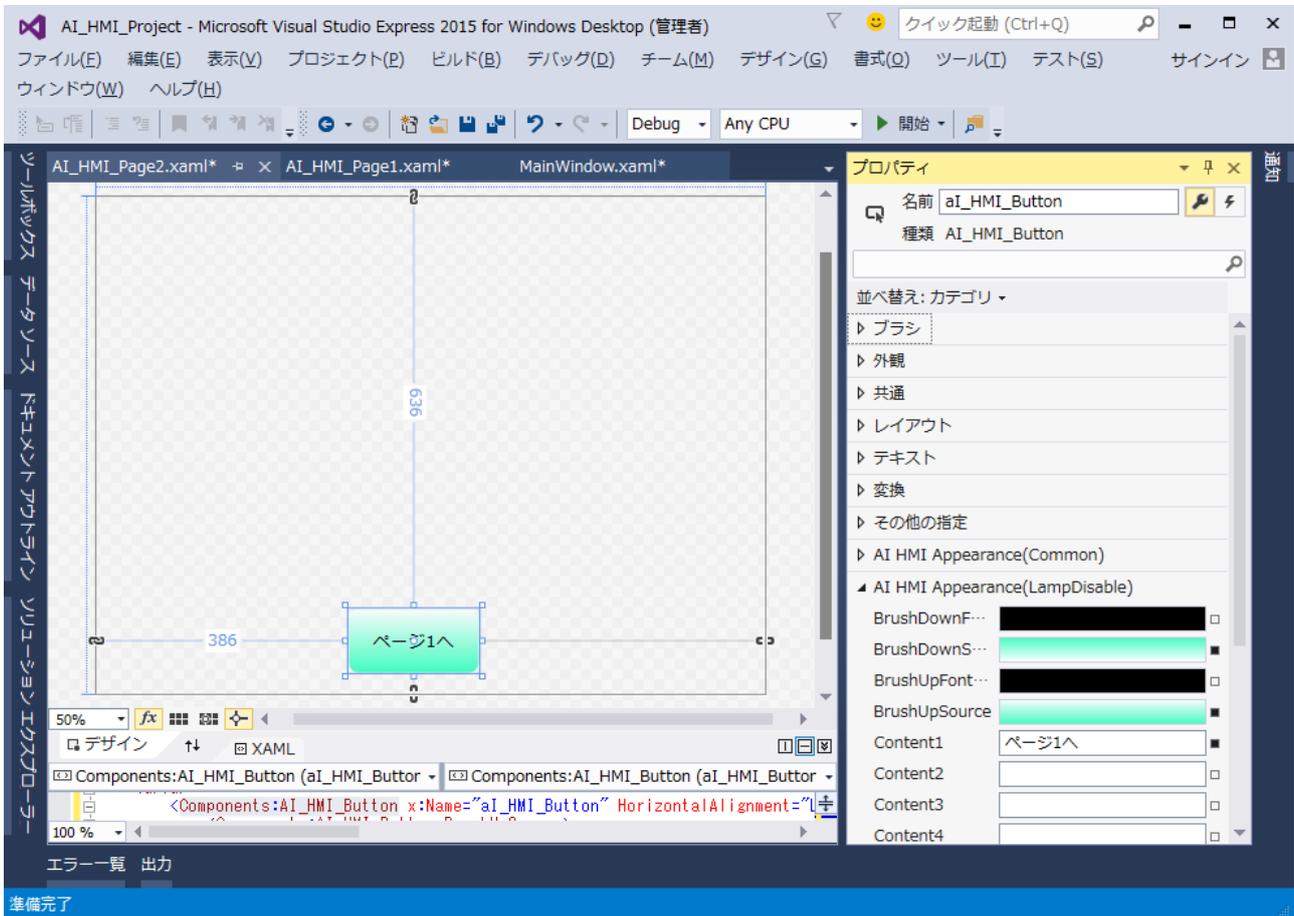


図 5-2-13 AI_HMI_Button のプロパティの変更

表 5-2-6 AI_HMI_Button のプロパティ

項目	設定値	備考
レイアウト	Width	200 ボタンの幅
	Height	100 ボタンの高さ
テキスト	(フォントサイズ)	20pt 表示する文字の大きさ
AI HMI Appearance (Common)	CornerRadius	15 ボタンの角の丸み
AI HMI Appearance (LampDisable)	BrushDownSource	任意の色 ボタンを押した時の画像
	BrushUpSource	任意の色 ボタンを離れた時の画像
	Content1	ページ1へ ボタンに表示する文字列

- ⑬ ボタンが押されたときの挙動の設定をします。
ボタンプロパティの「AI HMI Switch – SwitchReference」のボタンをクリックしてください。

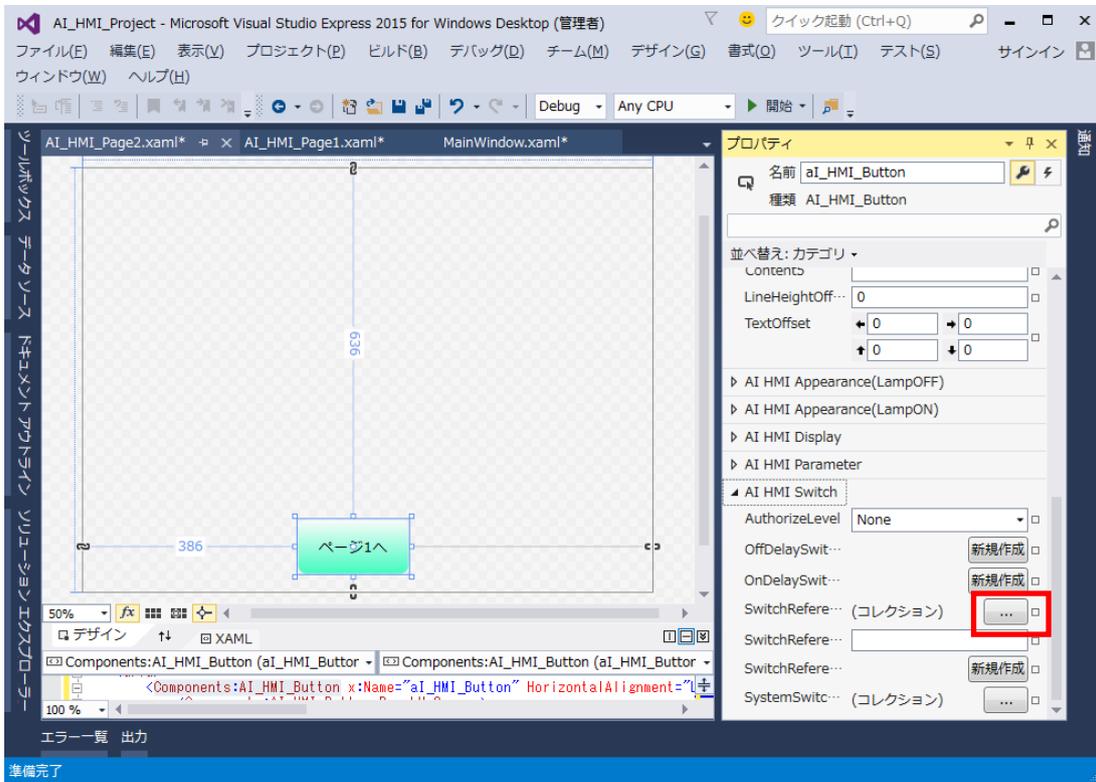


図 5-2-14 SwitchReference の変更

- ⑭ ボタンのコレクションエディタが開きます。「追加」ボタンをクリックしてください。

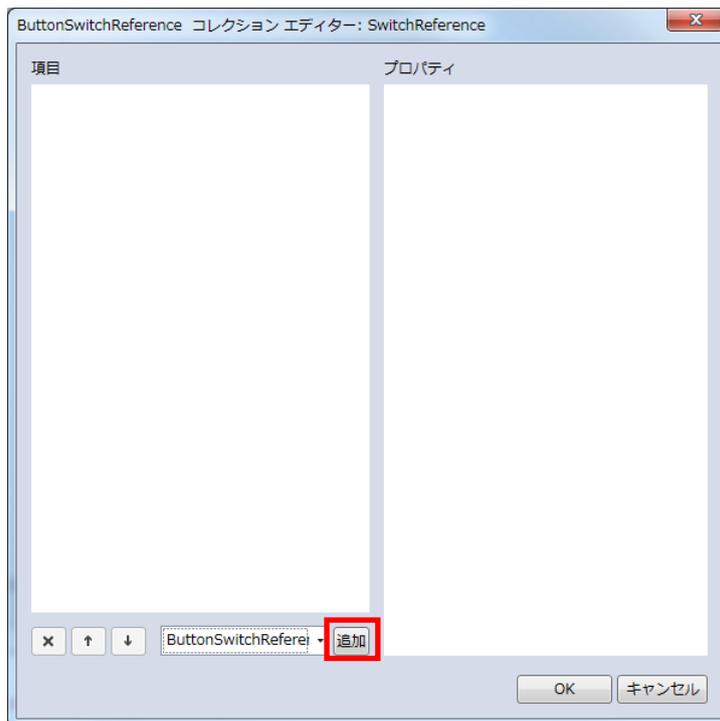


図 5-2-15 SwitchReference の変更

- ⑮ ButtonSwitchReference のプロパティが1項目追加されます。
表 5-2-6 のように設定をしてください。

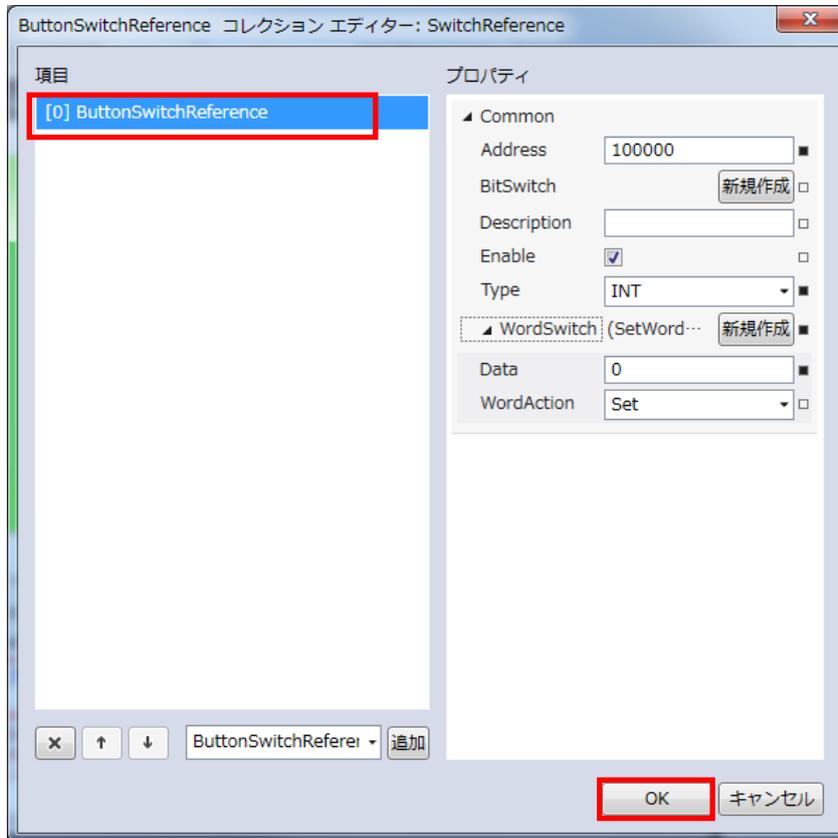


図 5-2-16 ButtonSwitchReference の変更

表 5-2-7 ButtonSwitchReference のプロパティ

項目	設定値	備考
Address	100000	共有メモリ上のアドレス
Type	INT	データタイプ
Data ※	0	ボタンが押されたときに書き込む値
WordAction ※	Set	ボタンが押されたときの挙動

※ WordSwitch の行の「新規作成」をクリックすることで表示します。

- ⑩ 先ほどのボタンとは別に文字列を表示するためのオブジェクトを配置します。
「ツールボックス」から「AI_HMI_Button」を選択し、AI_HMI_Page2 の画面上に配置してください。

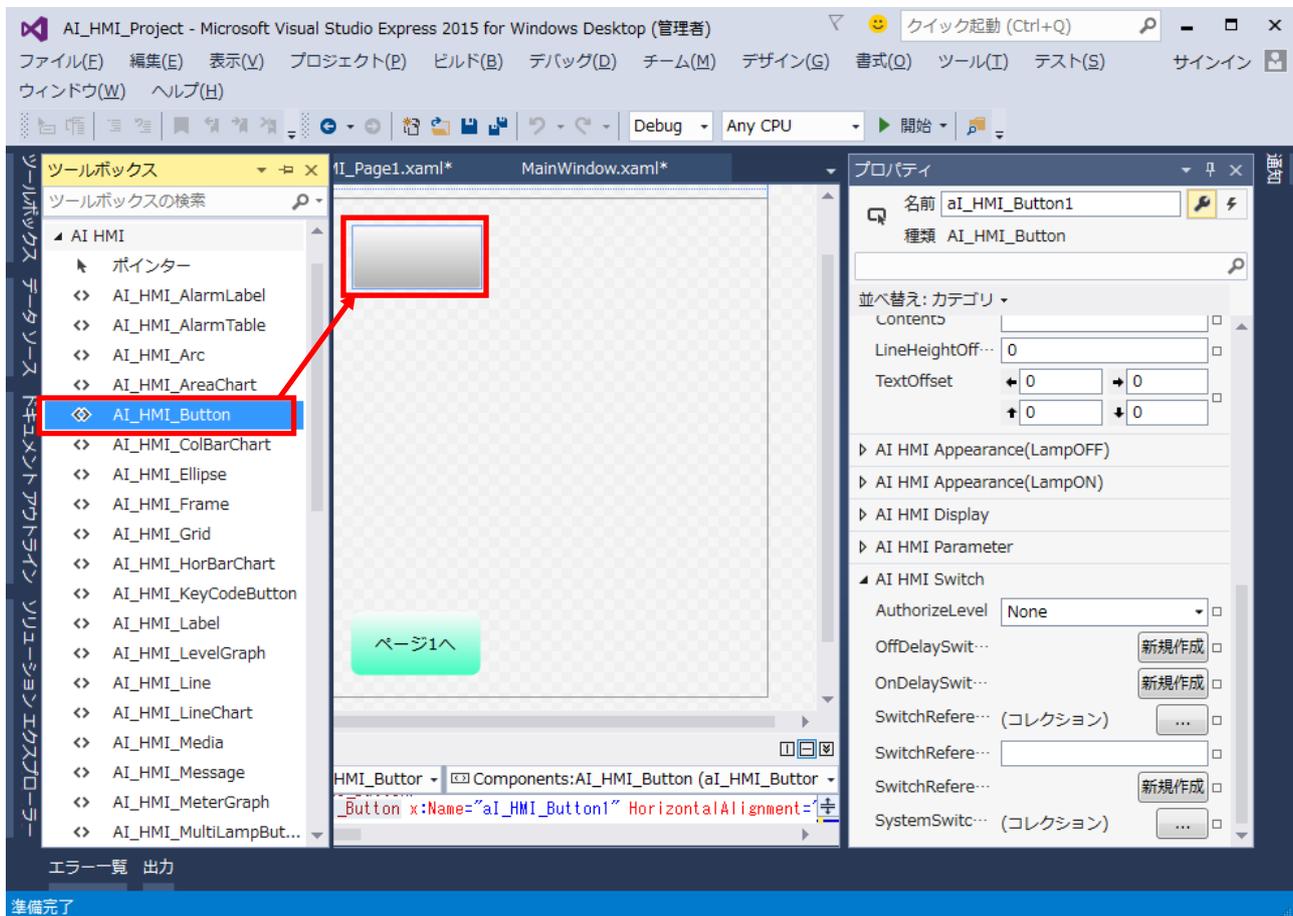


図 5-2-17 AI_HMI_Button の配置(2 個目)

- ⑰ ボタンオブジェクトが配置されるので、プロパティを表 5-2-8 のように設定してください。

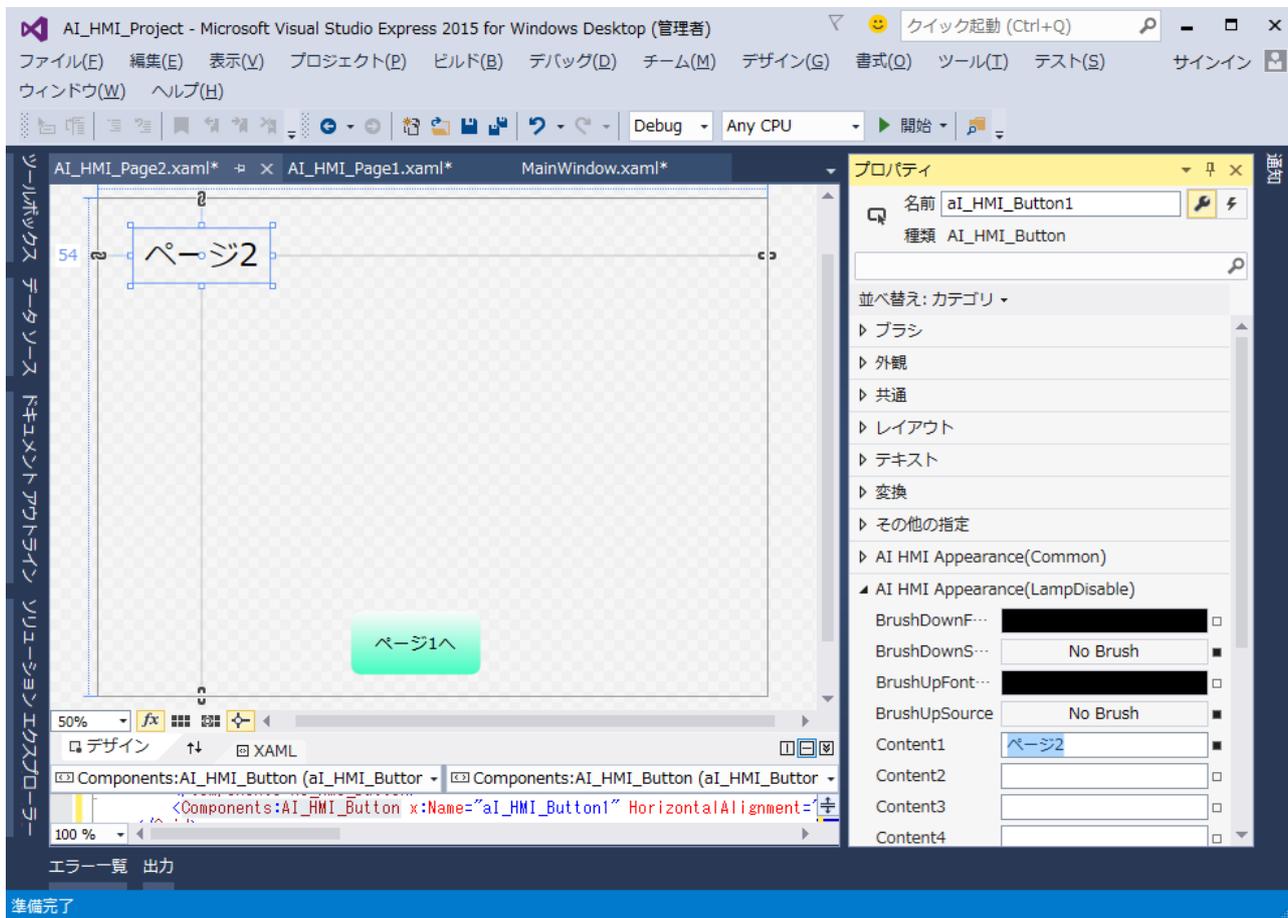


図 5-2-18 AI_HMI_Button(2 個目)のプロパティの変更

表 5-2-8 AI_HMI_Button(2 個目)のプロパティ

項目	設定値	備考
テキスト	(フォントサイズ)	36pt
AI HMI Appearance(LampDisable)	BrushDownSource	No Brush
	BrushUpSource	No Brush
	Content1	ページ 2
		表示する文字の大きさ
		ボタンの色を透明にする
		ボタンに表示する文字列

- ⑱ 背景画像の色を変更します。
背景部分をクリックし、表 5-2-9 のようにプロパティを変更してください。

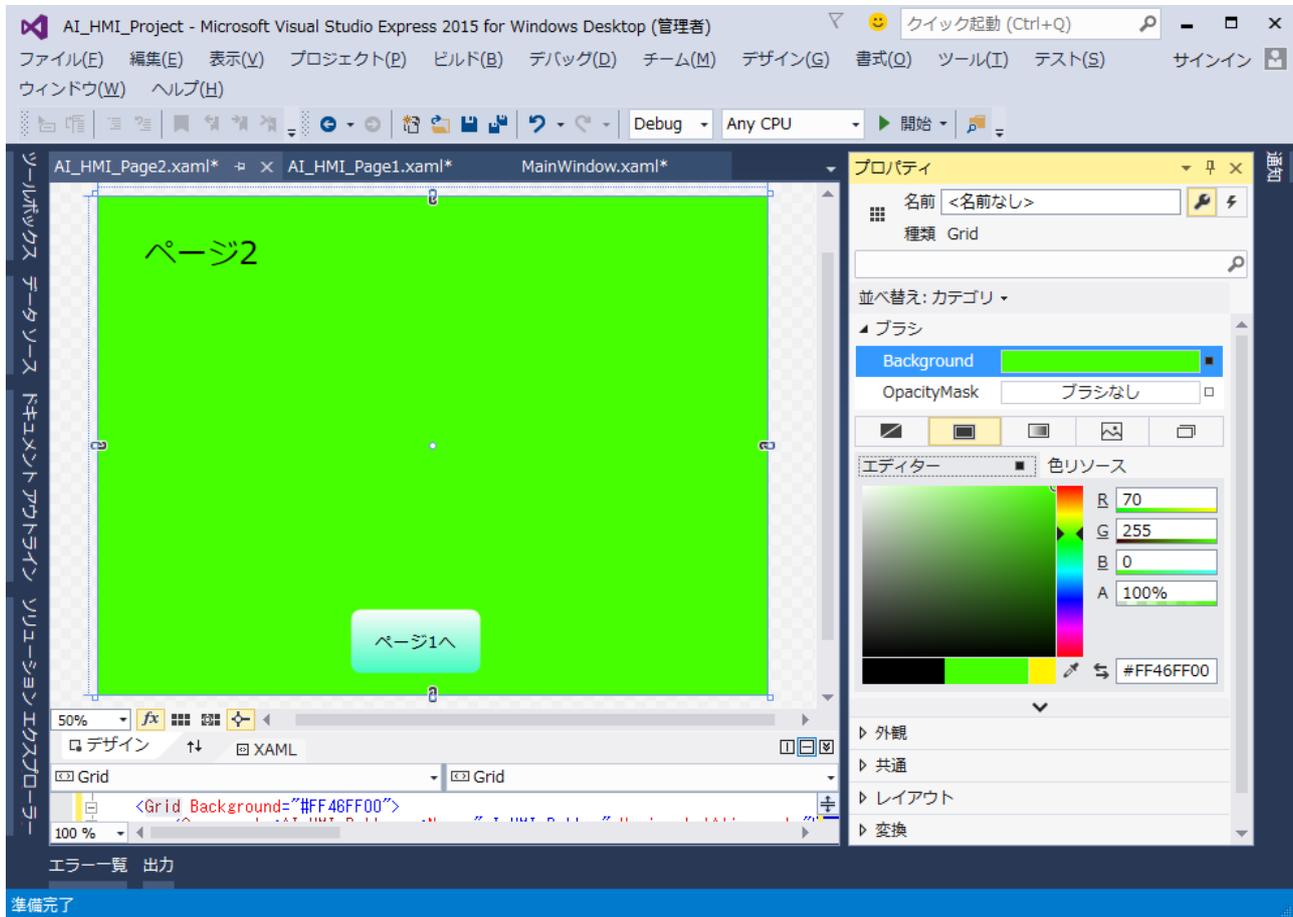


図 5-2-19 背景のプロパティの変更

表 5-2-9 背景のプロパティ

項目	設定値	備考
ブラシ	Background 任意の色	背景画像

5-3 フレームの設定

ページを作成しただけでは実行時に表示することはできません。
設定したページをフレーム上で表示できるように設定する必要があります。

- ① 「ソリューションエクスプローラー」を開き、「MainWindow.xaml」をダブルクリックしてください。

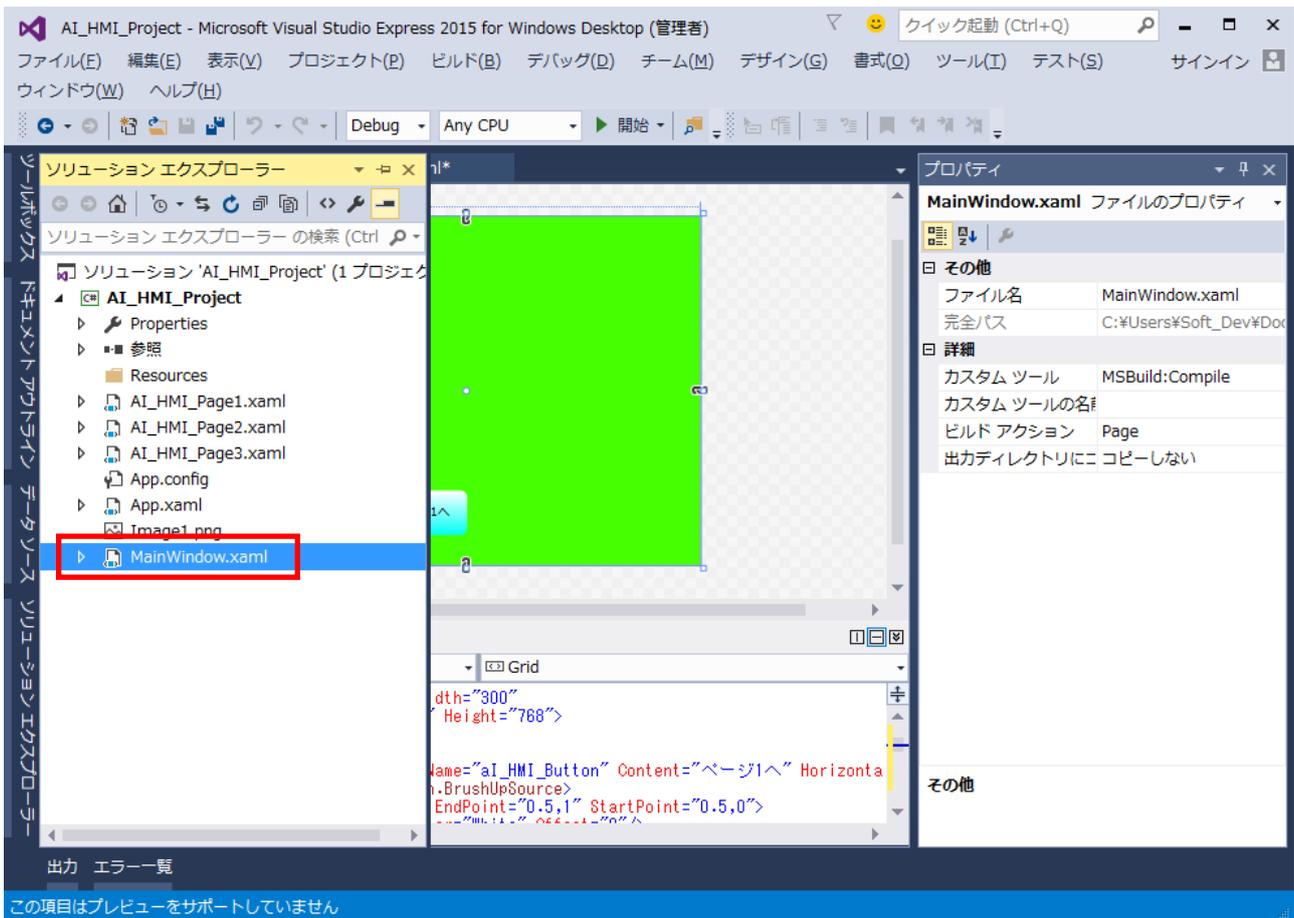


図 5-3-1 MainWindow の表示

- ② MainWindow 上には最初から AI_HMI_Frame というオブジェクトが設置されています。
AI_HMI_Frame - MainFrameReference のボタンをクリックしてください。

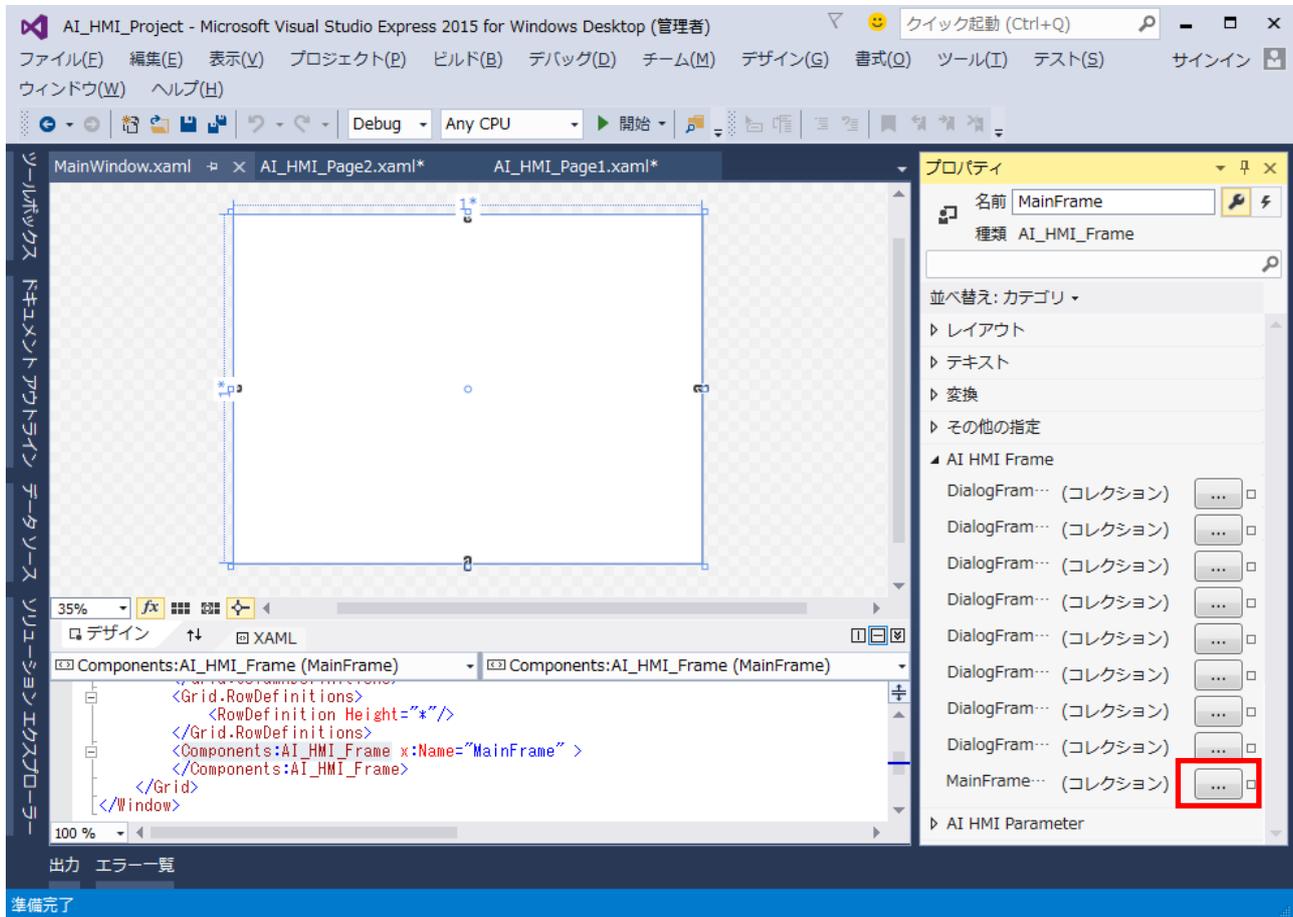


図 5-3-2 MainWindow の表示

- ③ MainFrameReference 設定画面が開きます。
「追加」ボタンをクリックして FramePLCReference」を 1 行追加してください。
追加された行のプロパティを表 5-3-1 のように変更してください。

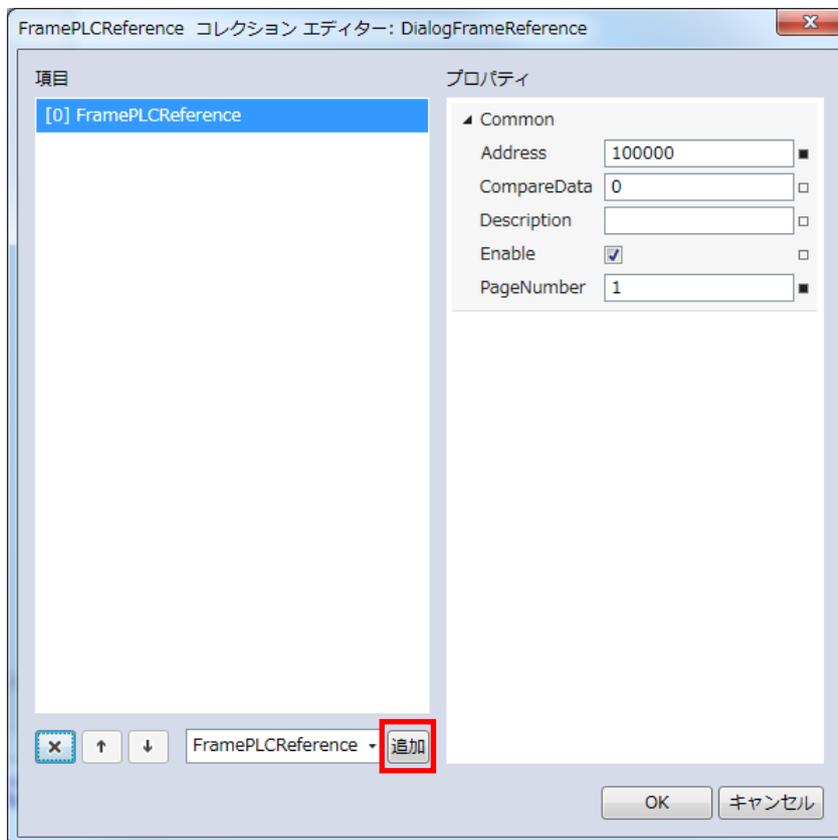


図 5-3-3 [0]FramePLCReference の設定

表 5-3-1 [0]FramePLCReference1 のプロパティ

項目	設定値	備考
Address	100000	参照する共有メモリ上のアドレス
CompareData	0	比較対象の数値
PageNumber	1	比較一致時に表示するページ番号

FramePLCReference ではアプリケーションで表示する画面(ページ)を表示する条件を設定することができます。

上記の例の場合、「共有メモリアドレス 100000」に格納されている値と「数値 0」とを比較して一致する場合に「ページ番号 1」のページを表示します。

ページ番号は AI_HMI_PageX の番号 X になります。

ページ番号を 0 にすることでページを表示しない状態になります。

FramePLCReference の行を追加することで、条件毎に表示する画面を切り替えることができます。

- ④ 同様の手順で AI_HMI_Page2 も表示できるようにします。
「追加」ボタンをクリックして FramePLCReference」を更に1行追加してください。
追加された行のプロパティを表 5-3-2 のように変更してください。

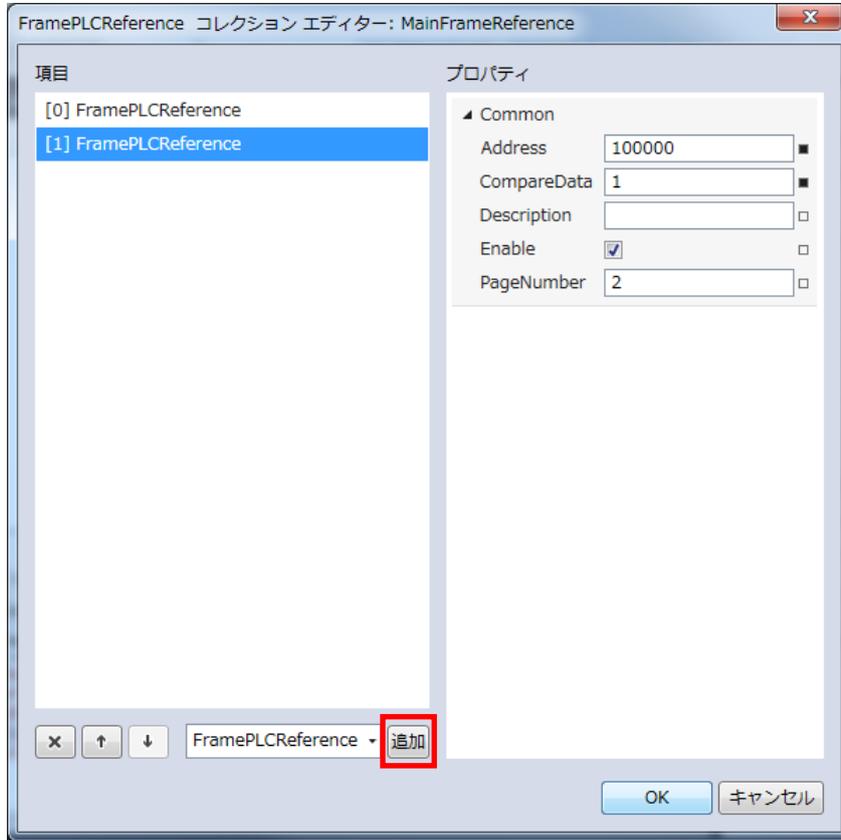


図 5-3-4 [1]FramePLCReference の設定

表 5-3-2 [1]FramePLCReference のプロパティ

項目	設定値	備考
Address	100000	参照する共有メモリ上のアドレス
CompareData	1	比較対象の数値
PageNumber	2	比較一致時に表示するページ番号

5-4 アプリケーションのコンパイルとデバッグ

VisualStudio のデバッグ機能を利用してデバッグを行います。
デバッグを実施する前にアプリケーションをコンパイルする必要があります。

- ① 上部メニューの「ビルド - ソリューションのリビルド」を選択してください。

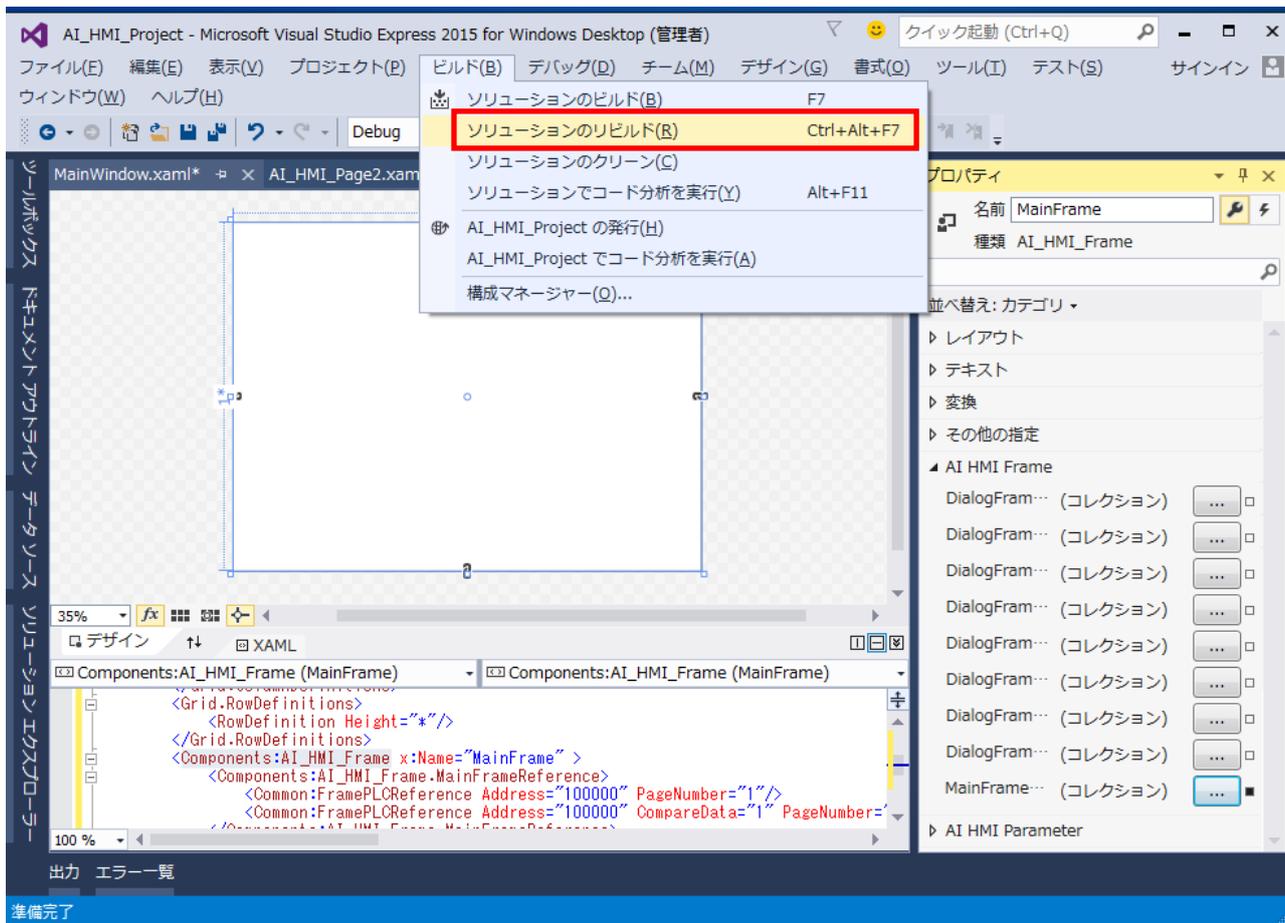


図 5-4-1 プロジェクトのコンパイル

アプリケーションのコンパイルが開始されます。
コンパイルが完了するまでしばらくお待ちください。

- ② コンパイルが完了したら「開始」をクリックしてください。
デバッグが開始され、作成した画面が表示されます。

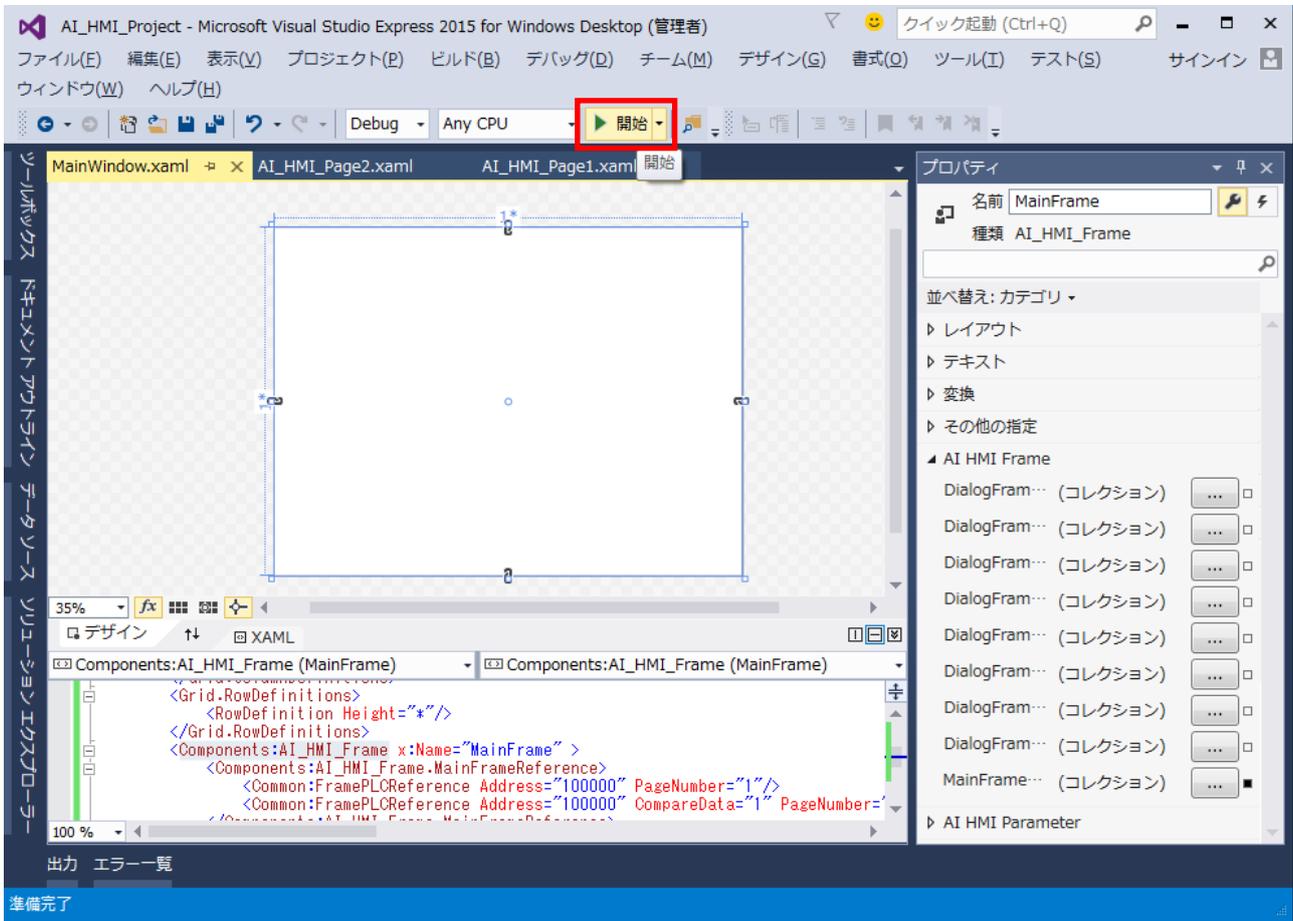


図 5-4-2 デバッグの実行

設定に問題がなかった場合、以下のようにボタンを押すことでページ 1 とページ 2 を行き来する動作を確認できます。

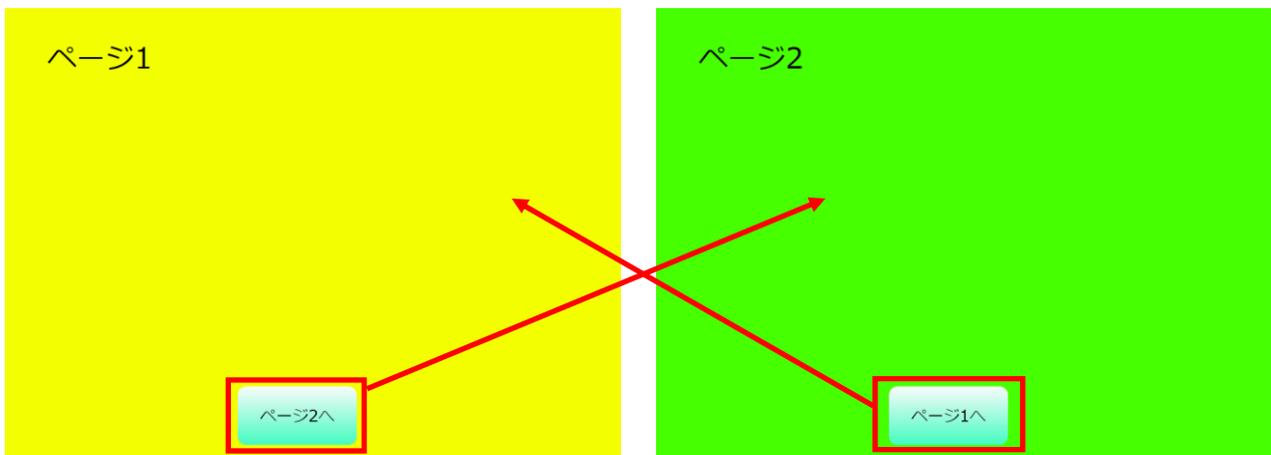


図 5-4-3 デバッグの実行

- ※ デバッグモードで実行時、デバッグを開始してから1時間経過するとライセンスエラーのメッセージが表示されます。(図 5-4-4)
この場合は「OK」ボタンをクリックすることでメッセージを閉じることができますが、デバッグ中は1時間毎に表示されます。
このメッセージはAI-HMI 実行環境 PC で実行する場合は表示されません。

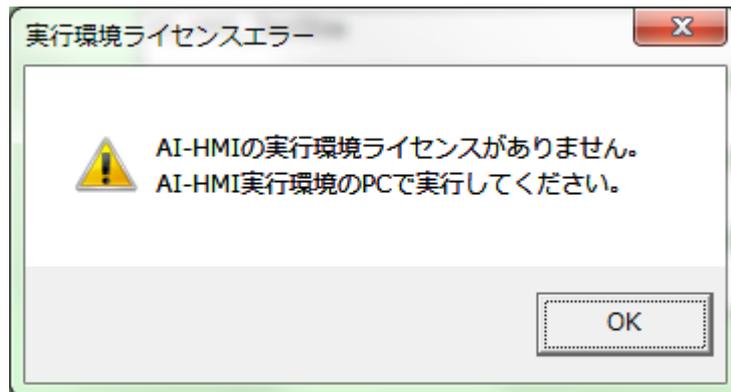


図 5-4-4 実行環境ライセンスエラー

5-5 ダイアログの追加

AI_HMI_Frame の DialogFrameReference を使用してダイアログをポップアップさせる機能を追加します。ダイアログの作成は基本的にはページの作成と同じです。

今回は新しく AI_HMI_Page を追加してダイアログとして表示させます。

- ① 「ソリューションエクスプローラー」の「AI_HMI_Project」を右クリックし、「追加 - 新しい項目」を選択してください。

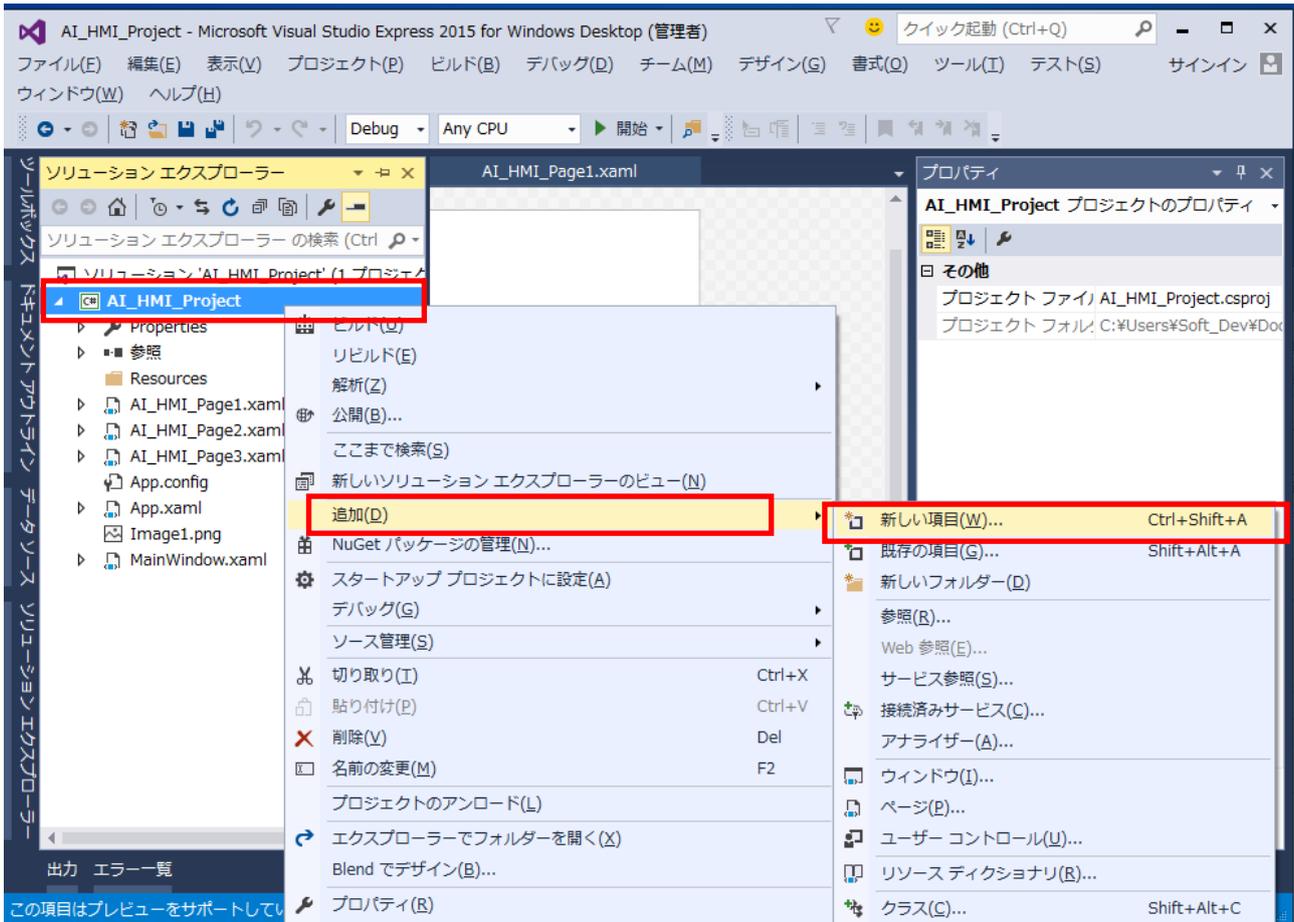


図 5-5-1 新規のページの追加

- ② 新しい項目の追加画面が開きます。
AI_HMI_Page を選択し、「追加」ボタンをクリックしてください。

※ AI_HMI_Page の名前は変更しないでください

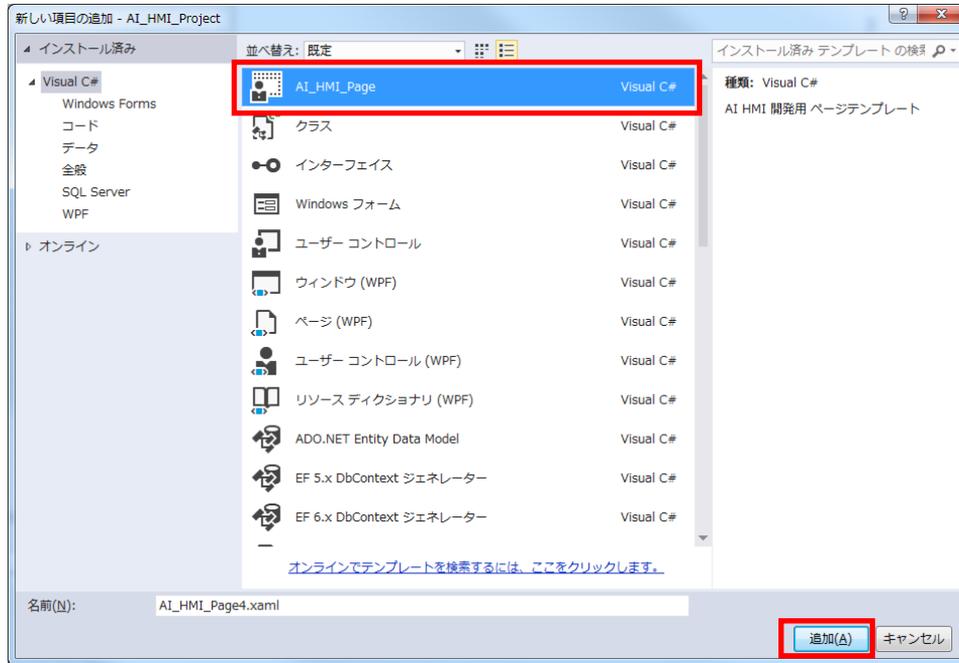


図 5-5-2 AI_HMI_Page4 の追加

- ③ AI_HMI_Page4.xaml がソリューションエクスプローラーに追加されたことを確認してください。
AI_HMI_Page4.xaml をダブルクリックしてください。

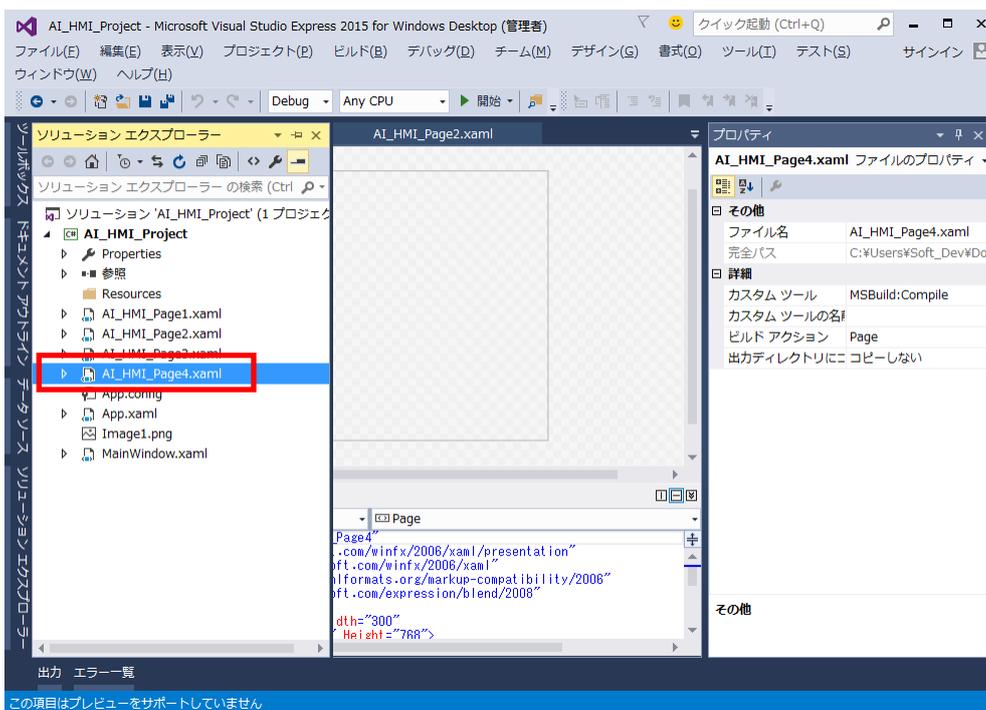


図 5-5-3 AI_HMI_Page4 の追加

- ④ AI_HMI_Page4 にテンキーを配置します。
「ツールボックス」の「AI_HMI_TenKey」を AI_HMI_Page4 に配置してください。

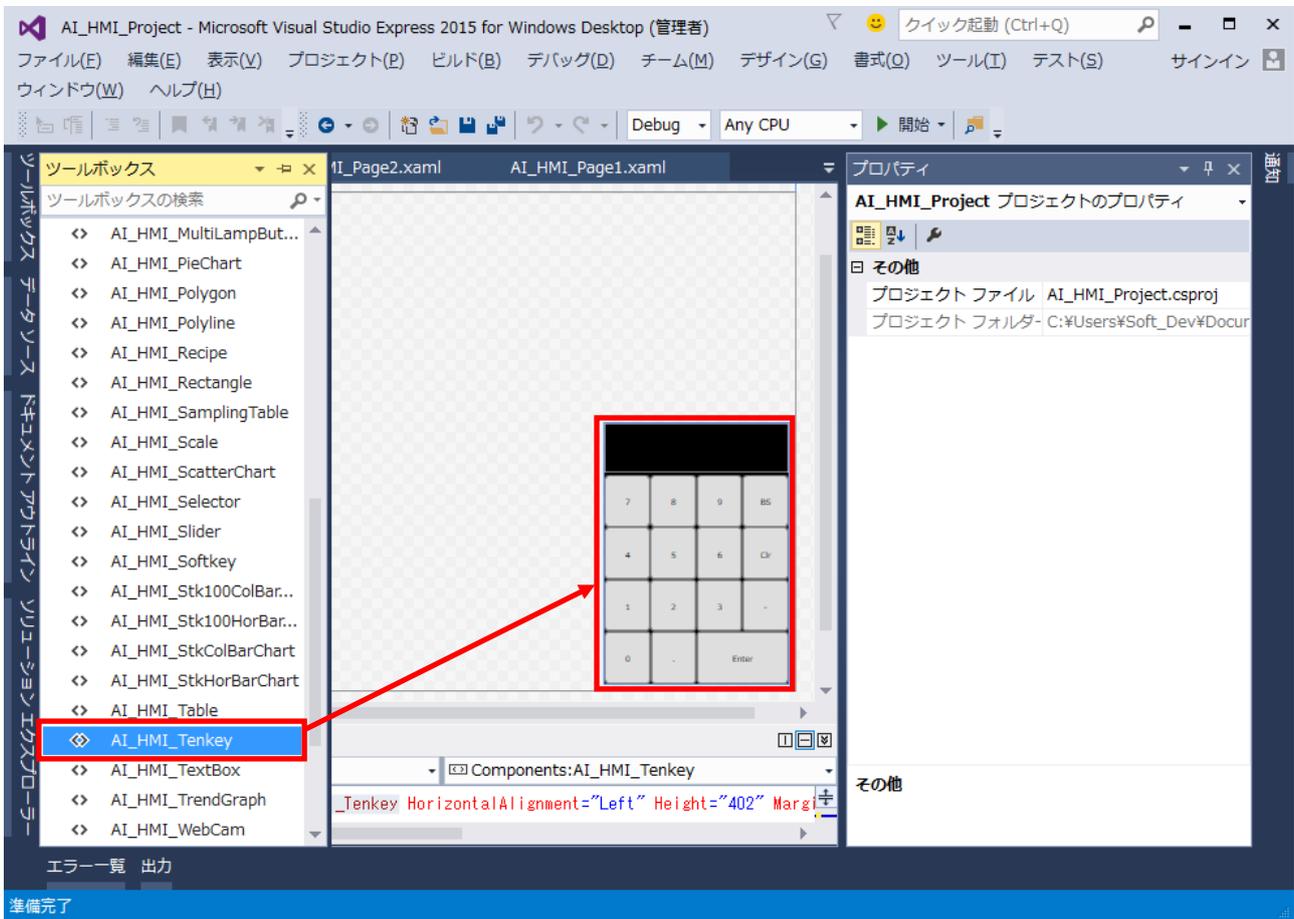


図 5-5-4 AI_HMI_TenKey の追加

- ⑤ テンキーオブジェクトが配置されるので、プロパティを表 5-5-1 のように設定してください。

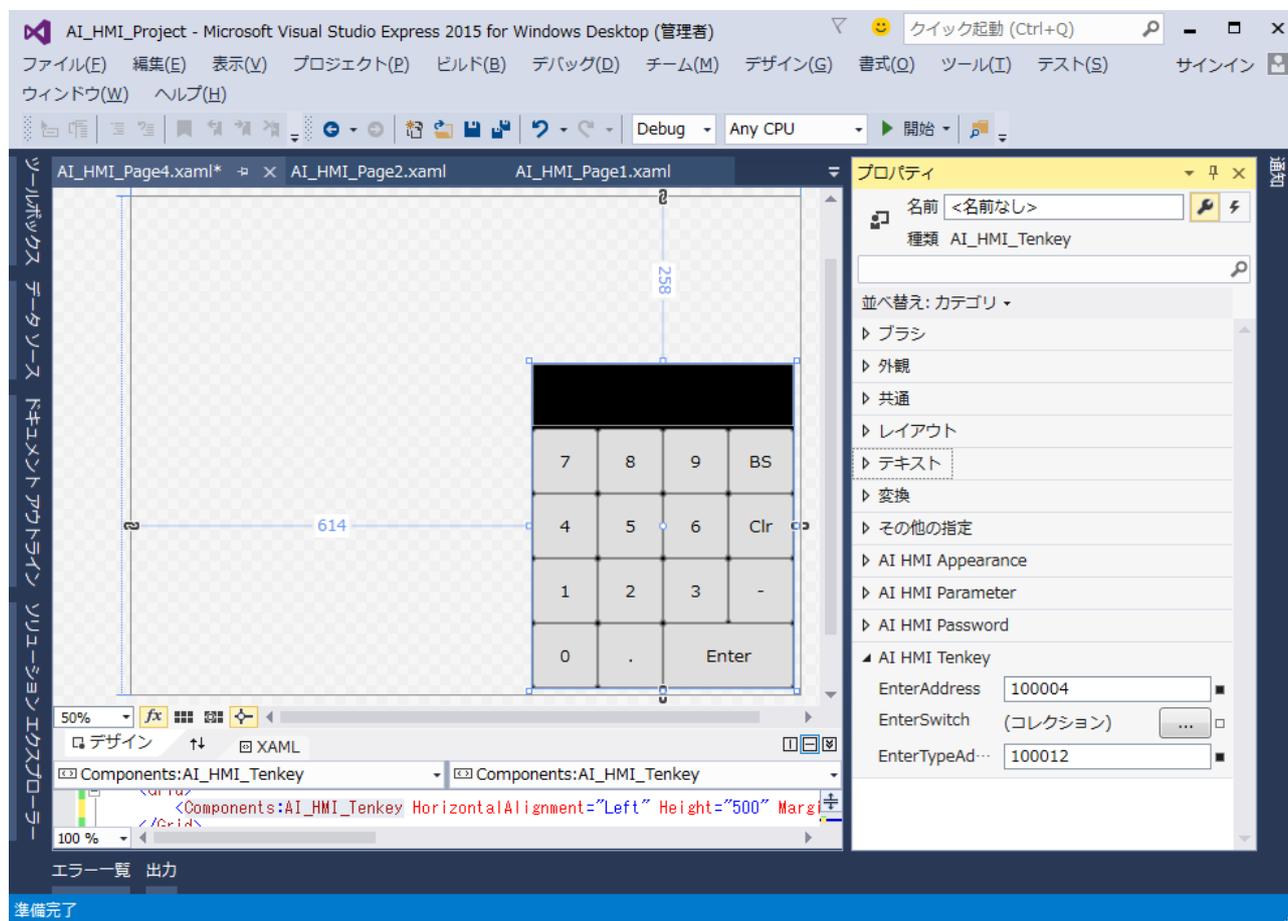


図 5-5-5 AI_HMI_Tenkey の設定

表 5-5-1 AI_HMI_Button1 のプロパティ

項目		設定値	備考
レイアウト	Width	400	テンキーの幅
	Height	500	テンキーの高さ
テキスト	(フォントサイズ)	20pt	表示する文字の大きさ
AI HMI Tenkey	EnterAddress	140000	テンキー入力先の共有メモリアドレスを格納する変数
	EnterTypeAddress	100012	テンキー入力時のデータ型番号

テンキーオブジェクトはテンキー入力を行うパラメータのアドレスと型指定のアドレスを共有メモリから取得することで、入力するパラメータ変数を自由に選択することができます。

テンキーは EnterAddress に入力したアドレス (今回は 140000) に格納されている値へテンキー入力の値を記述します。

また、EnterTypeAddress に入力したアドレス (今回は 100012) に格納されている値 (0 : BOOL、1 : INT、2 : UINT、3 : DINT、4 : UDINT、5 : REAL、6 : LREAL) に従って、テンキーが書き込むときのデータ型を指定します。

- ⑥ テンキー画面で数値の入力が完了したときの挙動を設定します。
テンキープロパティの「AI HMI Tenkey – EnterSwitch」のボタンをクリックしてください。

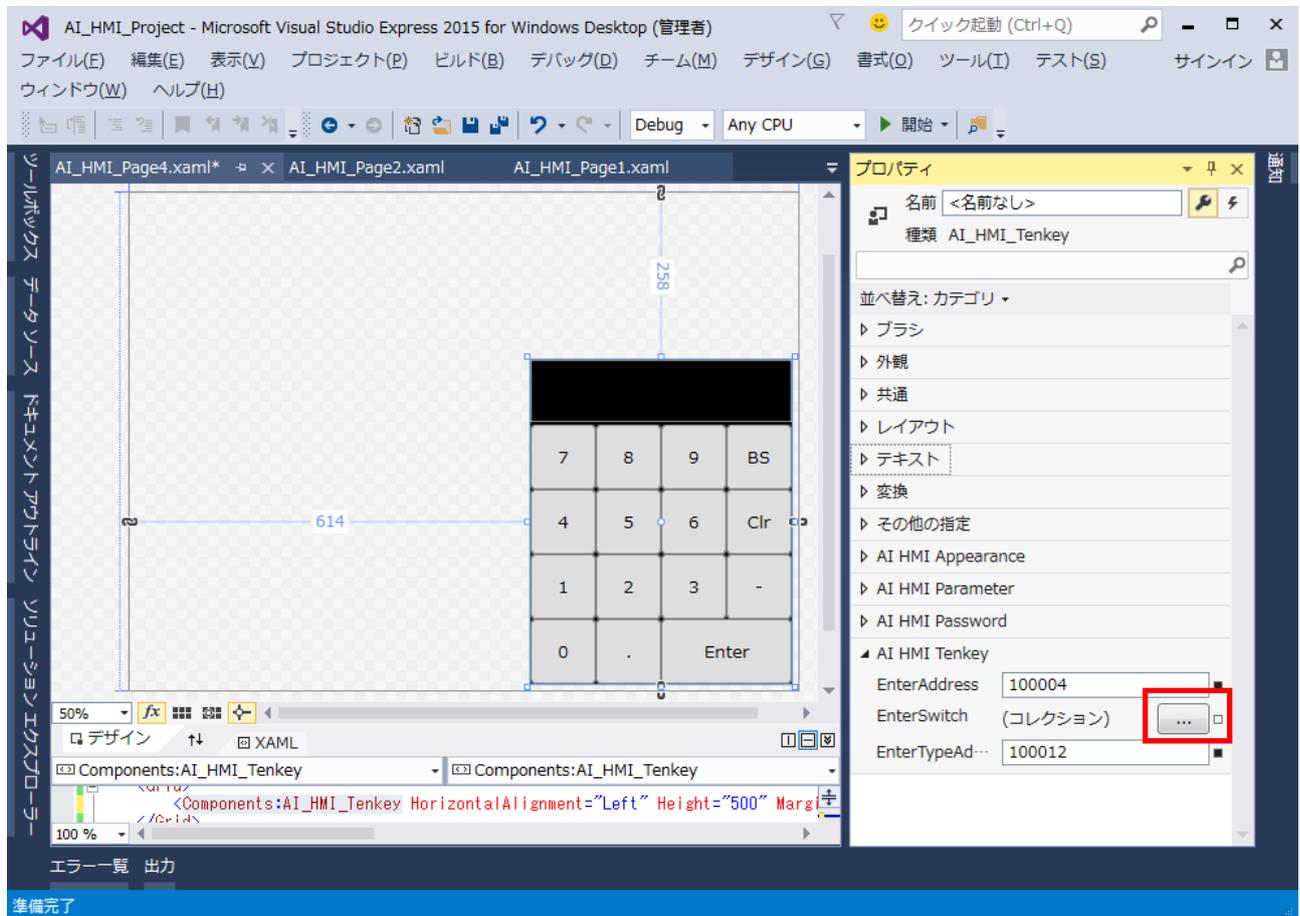


図 5-5-6 EnterSwitch の設定

- ⑦ テンキーのコレクションエディタが開きます。「追加」ボタンをクリックして CommonSwitchReference を 1 行追加し、表 5-5-2 のように設定をしてください。

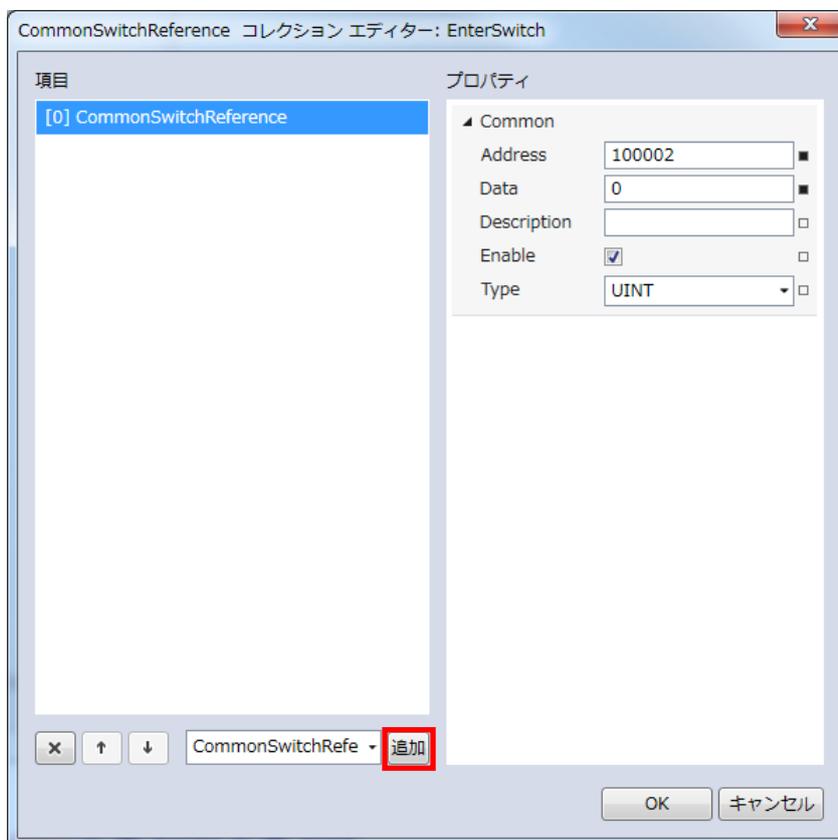


図 5-5-7 EnterSwitch の設定

表 5-5-2 CommonEnterSwitch のプロパティ

項目	設定値	備考
Address	100002	共有メモリ上のアドレス
Data	0	テンキー入力完了時に書き込む値
Type	UINT	テンキー入力完了時に書き込む値のデータ型

テンキー入力が完了したとき（テンキー上の ENTER ボタンが押されたとき）の挙動は CommonEnterSwitch を設定することで制御できます。

上記の例の場合、「共有メモリアドレス 100002」に対して「UINT 型変数」として「数値 0」を書き込みます。このサンプルではテンキーを表示しているダイアログを閉じるための挙動になります。

ButtonSwitchReference と同様に CommonSwitchReference の行を追加することで、一回のテンキー入力に対して複数の挙動を設定することができます。

- ⑧ ダイアログ上にテンキーを表示することができるようになりましたが、現状のままではダイアログ表示中に後ろのMainFrameの操作ができてしまいます。

今回はテンキー表示中は他のボタンが見えるようにしつつも操作できないようにするため、AI_HMI_Page4の背景色を透明色に指定します。

AI_HMI_Page4の背景部分を選択し、プロパティを表5-5-3のように設定してください。

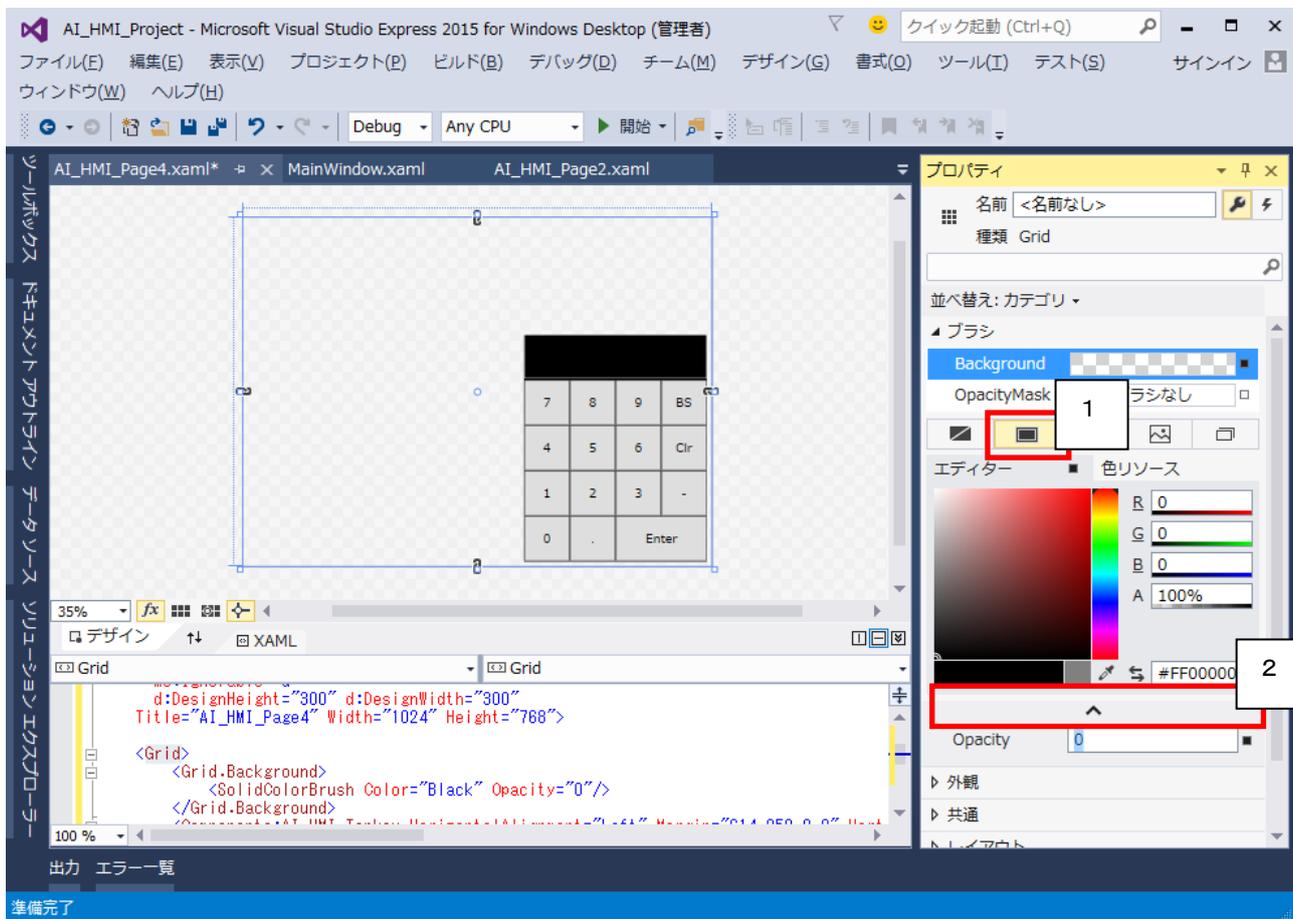


図 5-5-8 ダイアログ背景の設定

表 5-5-3 背景のプロパティ

項目	設定値	備考
ブラシ Background	Opacity ※	0 背景画像の透過度

※ BackgroundのOpacityの項目を表示するには以下の手順を踏んでください

1. 単色ブラシを選択する
2. 「V」マークをクリックしてオプションを展開する

- ⑨ 作成したダイアログ画面を AI_HMI_Page1 から呼び出せるように設定します。
AI_HMI_Page1 に新たに AI_HMI_Button を配置し、表 5-5-4 のようにプロパティを設定してください。

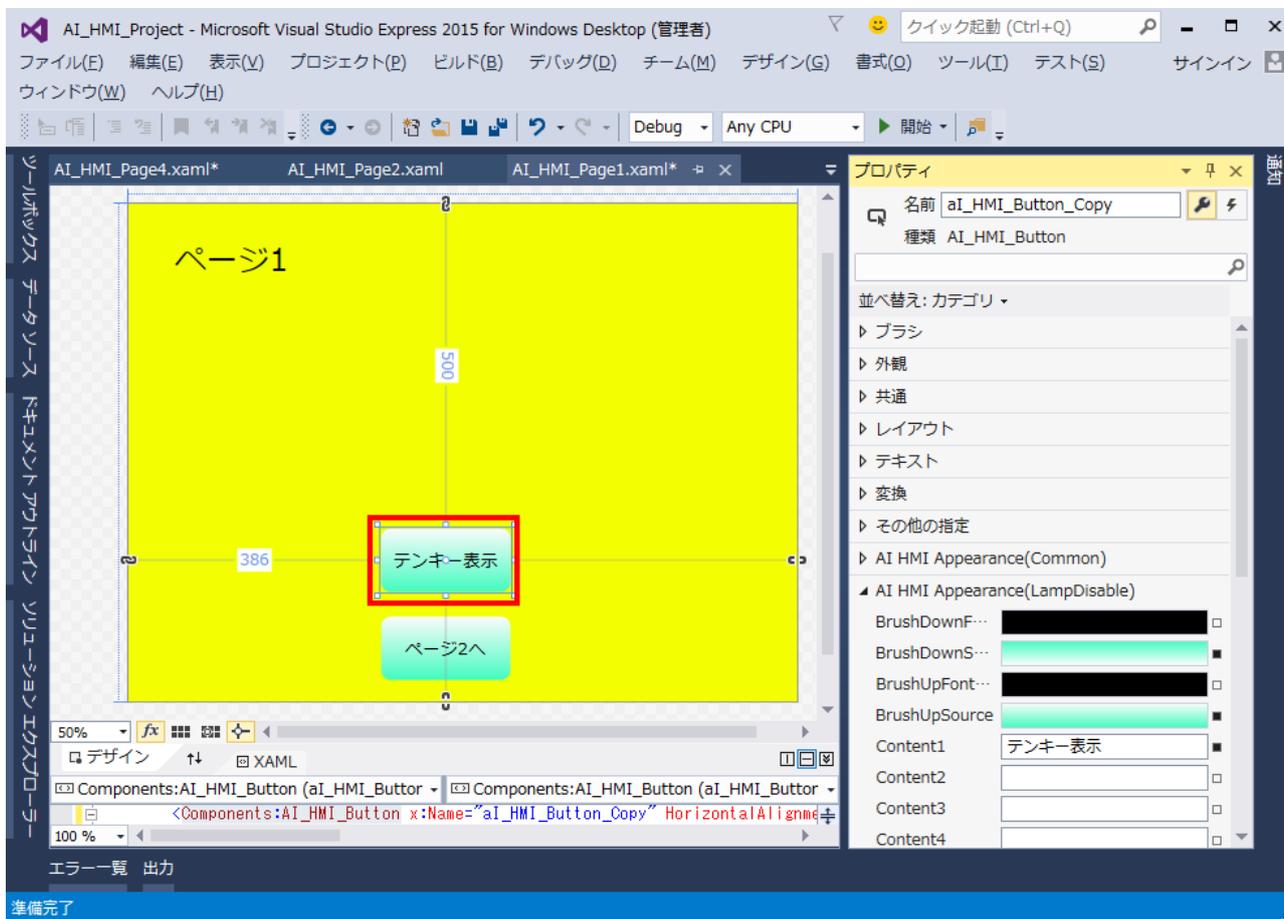


図 5-5-9 ダイアログ呼び出しボタンの配置

表 5-5-4 AI_HMI_Button(ダイアログ呼び出し用)のプロパティ

項目	設定値	備考
レイアウト	Width	200 ボタンの幅
	Height	100 ボタンの高さ
テキスト	(フォントサイズ)	20pt 表示する文字の大きさ
AI HMI Appearance (Common)	CornerRadius	15 ボタンの角の丸み
AI HMI Appearance (LampDisable)	BrushDownSource	任意の色 ボタンを押した時の画像
	BrushUpSource	任意の色 ボタンを離れた時の画像
	Content1	テンキー表示 ボタンに表示する文字列

- ⑩ 配置したボタンを押した時の挙動を設定します。
「AI HMI Switch – SwitchReference」のボタンをクリックして SwitchReference 設定画面を開いてください。

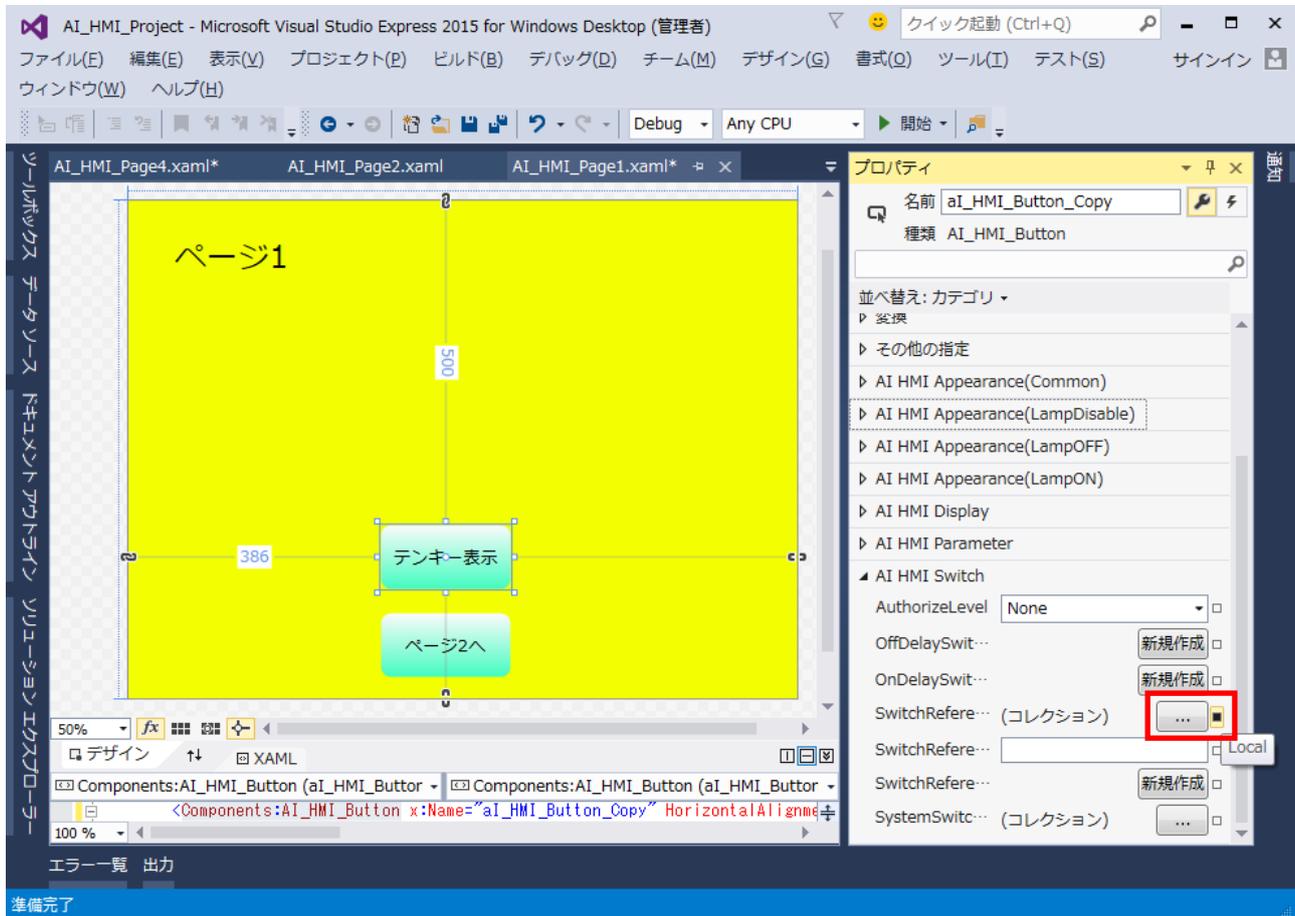


図 5-5-10 SwitchReference の設定

- ⑪ 「追加」ボタンをクリックしてください。
 [0]ButtonSwitchReference の行が追加されるので、表 5-5-5 のように設定してください。

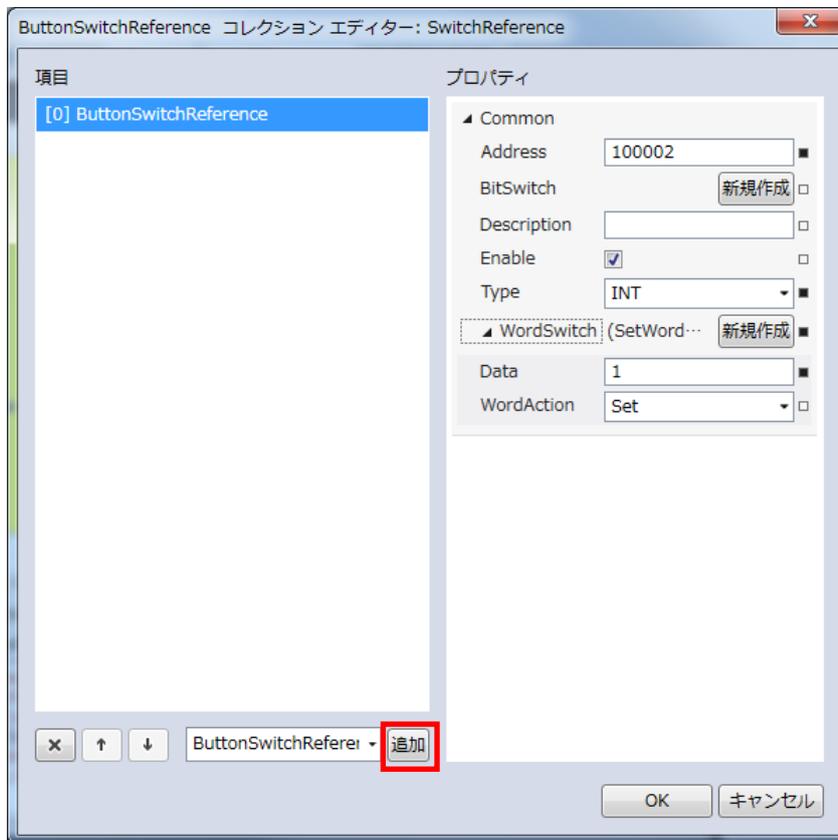


図 5-5-11 [0]ButtonSwitchReference の設定

表 5-5-5 [0]ButtonSwitchReference のプロパティ

項目	設定値	備考
Address	100002	共有メモリ上のアドレス
Type	INT	データタイプ
Data	1	ボタンが押されたときに書き込む値
WordAction	Set	ボタンが押されたときの挙動

[0]ButtonSwitchReference ではダイアログを表示するためのデータの入力をします。

- ⑫ もう一度「追加」ボタンをクリックしてください。
 [1]ButtonSwitchReference の行が追加されるので、表 5-5-6 のように設定してください。

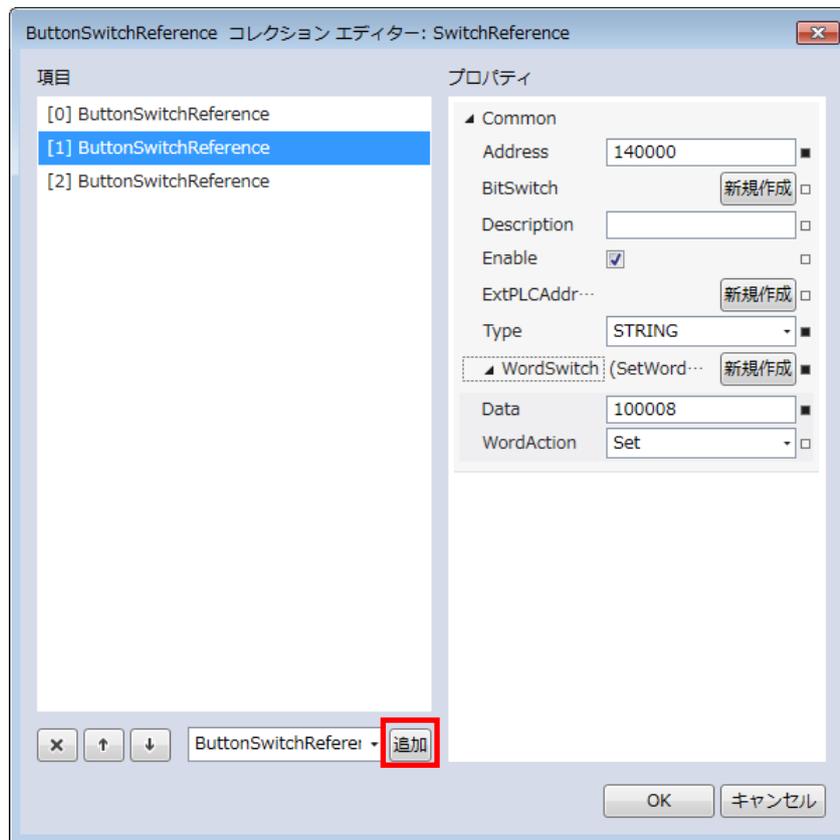


図 5-5-12 [1]ButtonSwitchReference の設定

表 5-5-6 [1]ButtonSwitchReference のプロパティ

項目	設定値	備考
Address	140000	共有メモリ上のアドレス
Type	STRING	データタイプ
Data	100008	ボタンが押されたときに書き込む値
WordAction	Set	ボタンが押されたときの挙動

[1]ButtonSwitchReference ではテンキーの EnterAddress で参照しにいく先のアドレスを入力します。

- ⑬ もう一度「追加」ボタンをクリックしてください。
 [2]ButtonSwitchReference の行が追加されるので、表 5-5-7 のように設定してください。

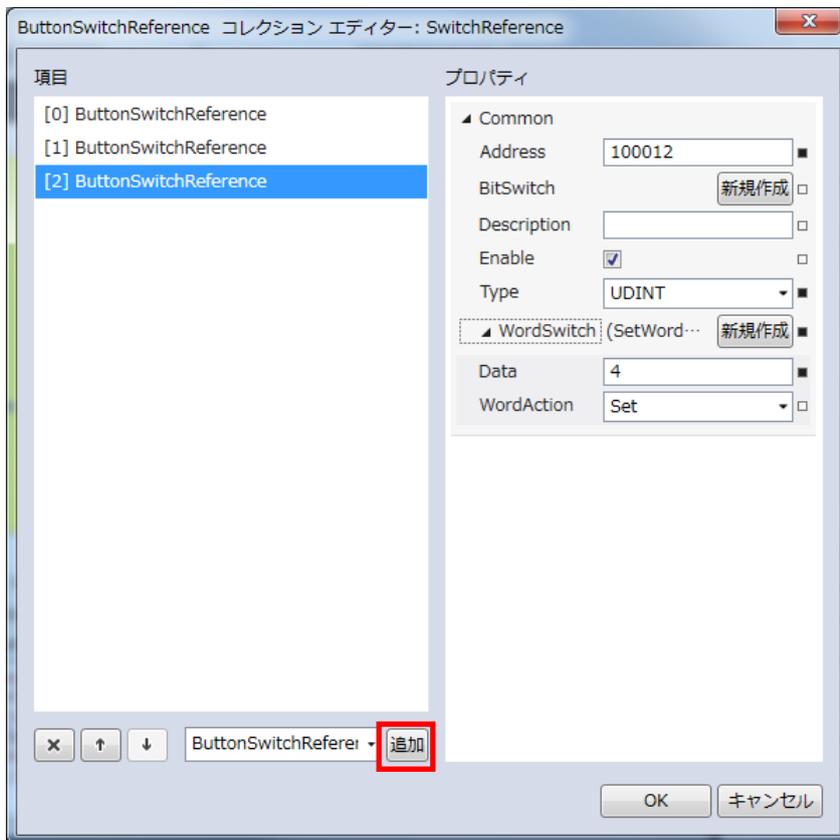


図 5-5-13 [2]ButtonSwitchReference の設定

表 5-5-7 [2]ButtonSwitchReference のプロパティ

項目	設定値	備考
Address	100012	共有メモリ上のアドレス
Type	UDINT	データタイプ
Data	4	ボタンが押されたときに書き込む値
WordAction	Set	ボタンが押されたときの挙動

[2]ButtonSwitchReference ではテンキーの EnterTypeAddress で取得するデータ型の番号を入力します。

- ⑭ 作成したダイアログをフレーム上で表示できるように設定します。
MainWindowを開いてください。

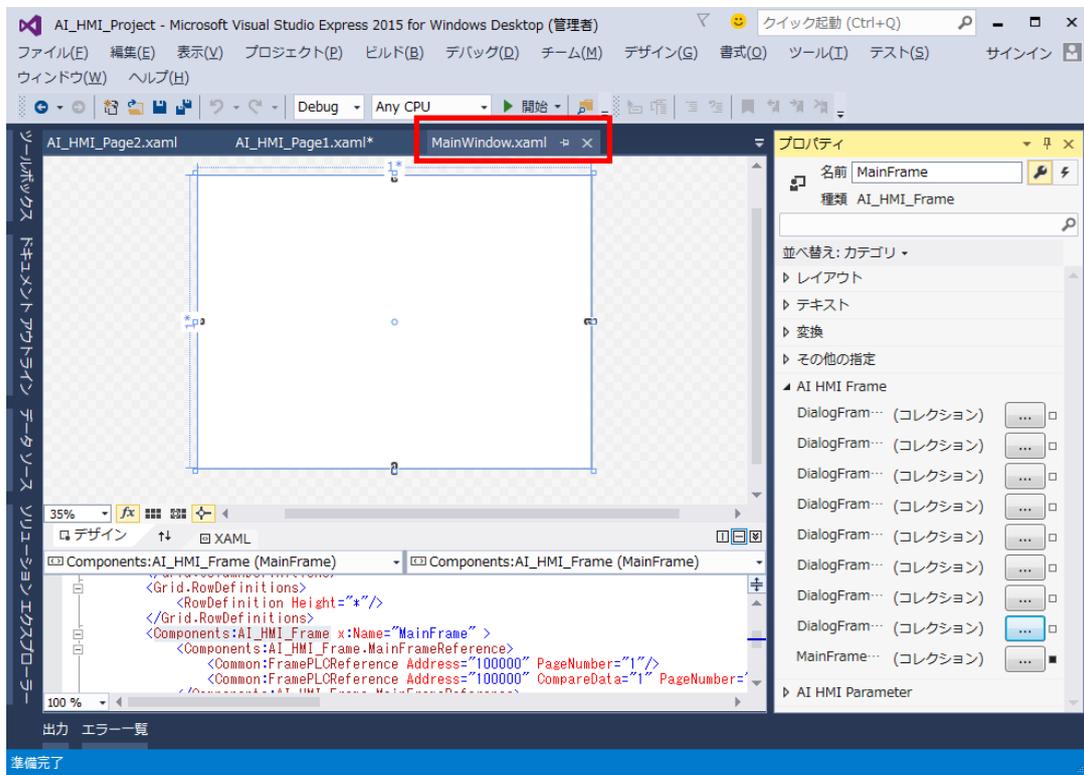


図 5-5-14 ダイアログ呼び出しボタンの配置

- ⑮ 「AI_HMI_Frame – DialogFrameReference8」のボタンをクリックしてください。
DialogFrameReference8 設定画面が開きます。

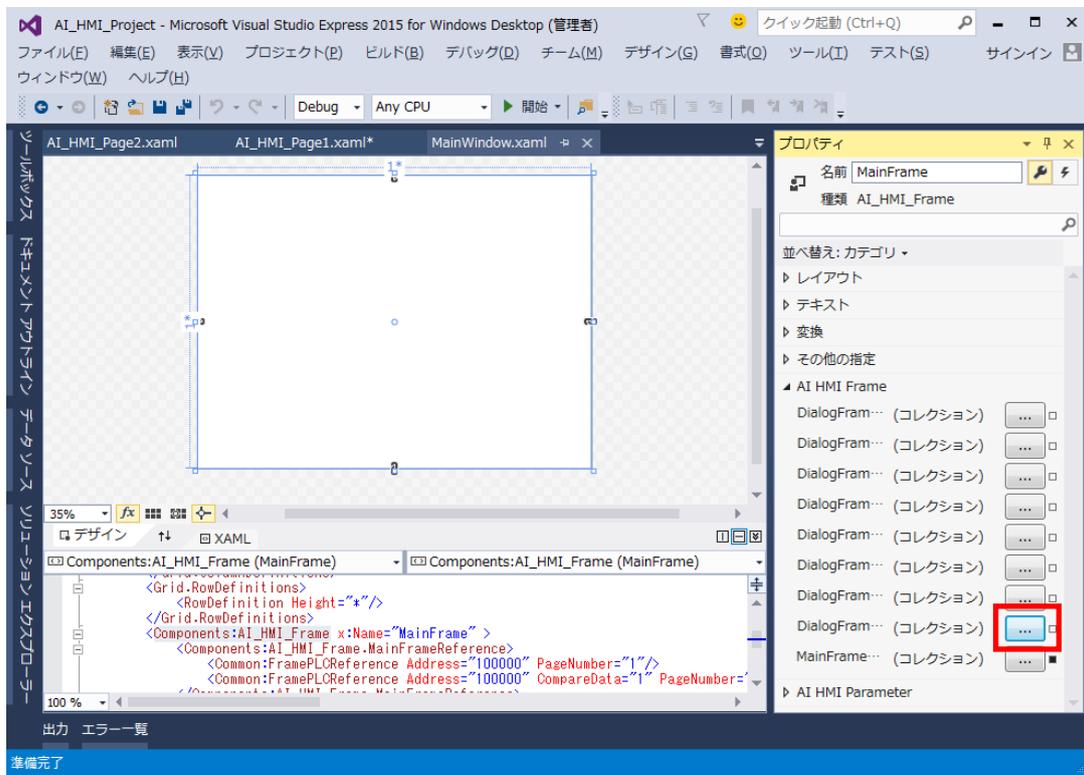


図 5-5-15 DialogReference の設定

⑩ 「追加」ボタンをクリックしてください。

[0]FramePLCReference の行が追加されるので表 5-5-8 のように設定してください。

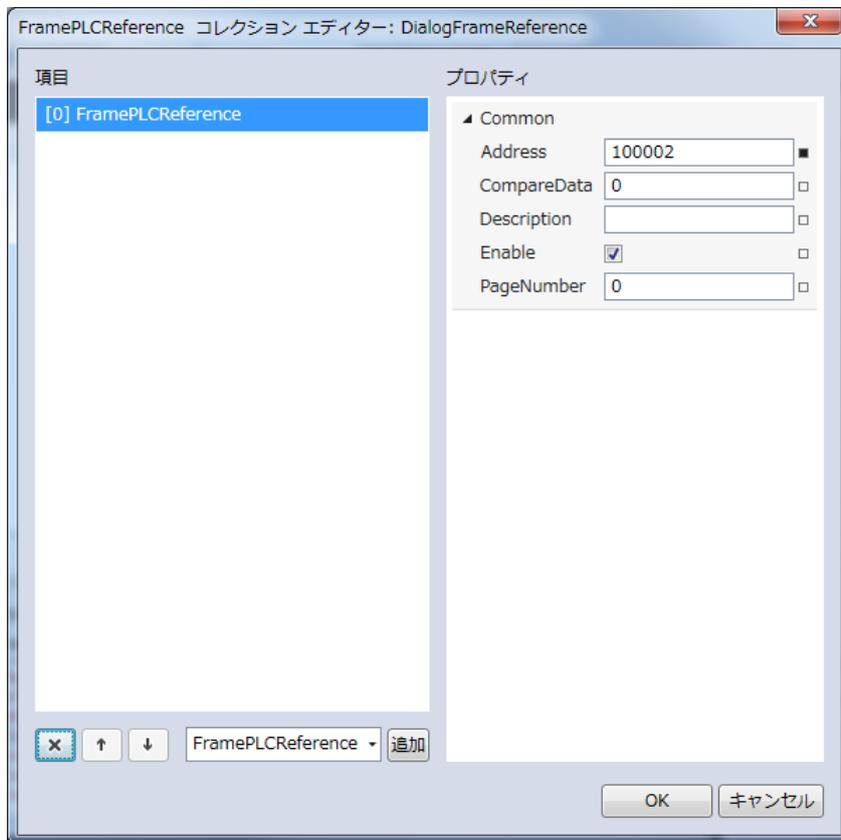


図 5-5-16 [0]FramePLCReference の設定

表 5-5-8 [0]FramePLCReference のプロパティ

項目	設定値	備考
Address	100002	参照する共有メモリ上のアドレス
Data	0	比較対象の数値
PageNumber	0	比較一致時に表示するページ番号

[0]FramePLCReference では PageNumber が 0 なので、ダイアログを表示しません。

テンキーの入力完了時に「共有メモリアドレス 100002」に 0 を入力するため、「テンキーの入力が完了したときにダイアログ(テンキー)を閉じる」という挙動を実現しています。

⑰ 「追加」ボタンをクリックしてください。

[1]FramePLCReference の行が追加されるので表 5-5-9 のように設定してください。

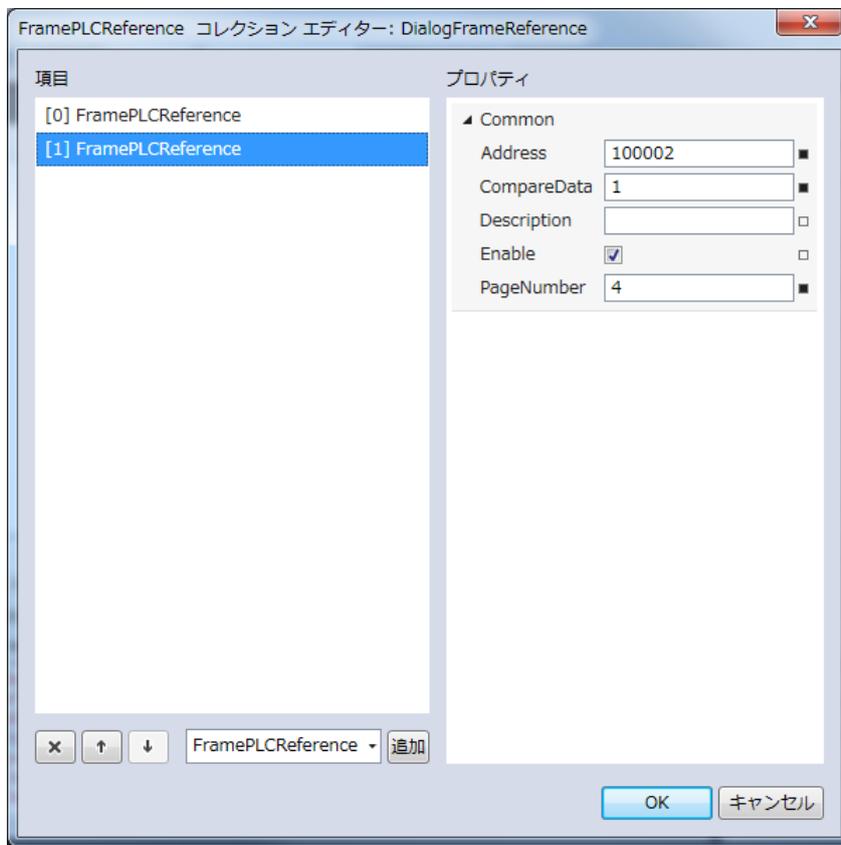


図 5-5-17 [1]FramePLCReference の設定

表 5-5-9 [1]FramePLCReference のプロパティ

項目	設定値	備考
Address	100002	参照する共有メモリ上のアドレス
Type	1	比較対象の数値
PageNumber	4	比較一致時に表示するページ番号

[1]FramePLCReference は AI_HMI_Page1 の「テンキー表示」ボタンがクリックされたとき、「共有メモリアドレス 100002」に 1 を入力するため、「ボタンがクリックされたときにダイアログ(テンキー)を表示する」という挙動を実現しています。

ここまでの作業でダイアログの作成は完了しました。

『5-4 アプリケーションのコンパイルとデバッグ』の手順に従い、作成したプログラムをコンパイルしてアプリケーションを実行してください。

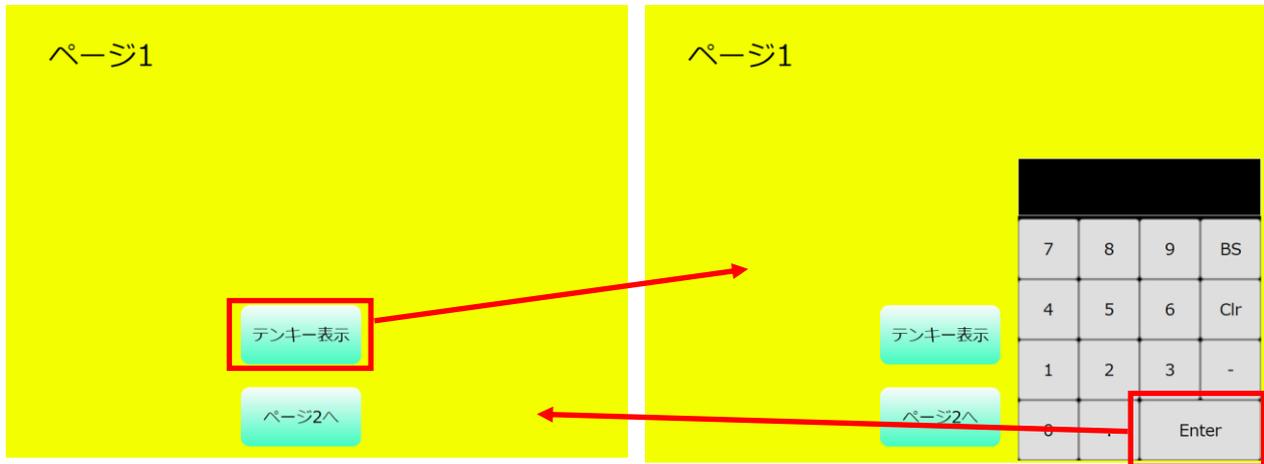


図 5-5-18 デバッグの実行

実際に「テンキー表示」ボタンをクリックするとテンキーが表示されること、テンキー表示中は他のボタンの操作ができなくなっていることを確認してください。

テンキー表示中に他のボタンの操作が無効化されていない場合は手順⑧を確認してください。

ボタンをクリックしてもテンキーが表示されない場合は手順⑨～⑬を確認してください。

テンキー入力をしてもテンキーが閉じない場合は手順⑮～⑰を確認してください。

5-6 ラベルによる共有メモリのデータの表示

前項でテンキーから共有メモリヘータを入力する仕組みを作成しました。
本項ではテンキーで入力した共有メモリのデータを画面上で表示できるように設定をします。

- ① AI_HMI_Page1 にデータ表示用のラベルを配置します。
「ツールボックス」の「AI_HMI_Label」をドラッグして配置してください。

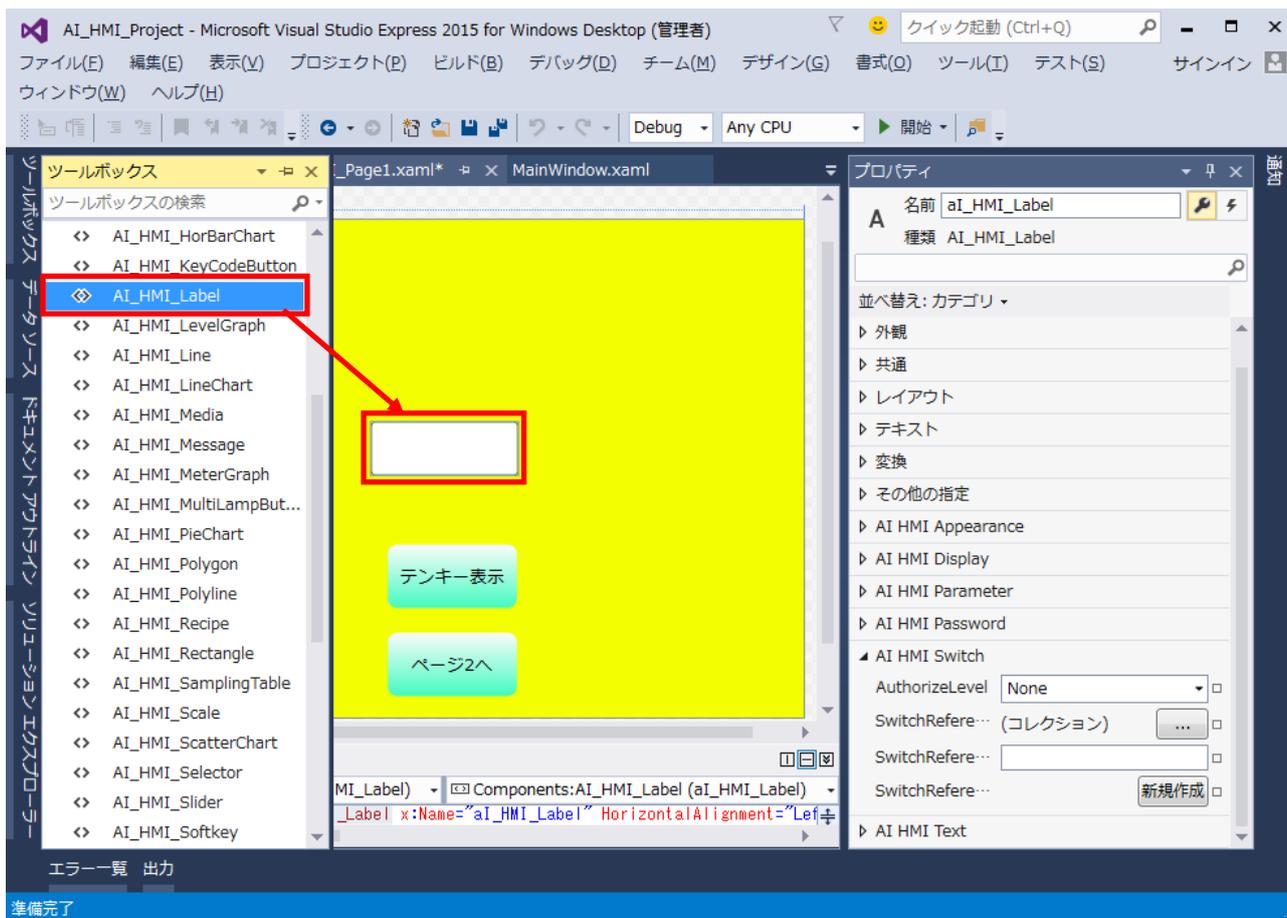


図 5-6-1 AI_HMI_Label の配置

② ラベルオブジェクトが配置されるので、プロパティを表 5-6-1 のように設定してください。

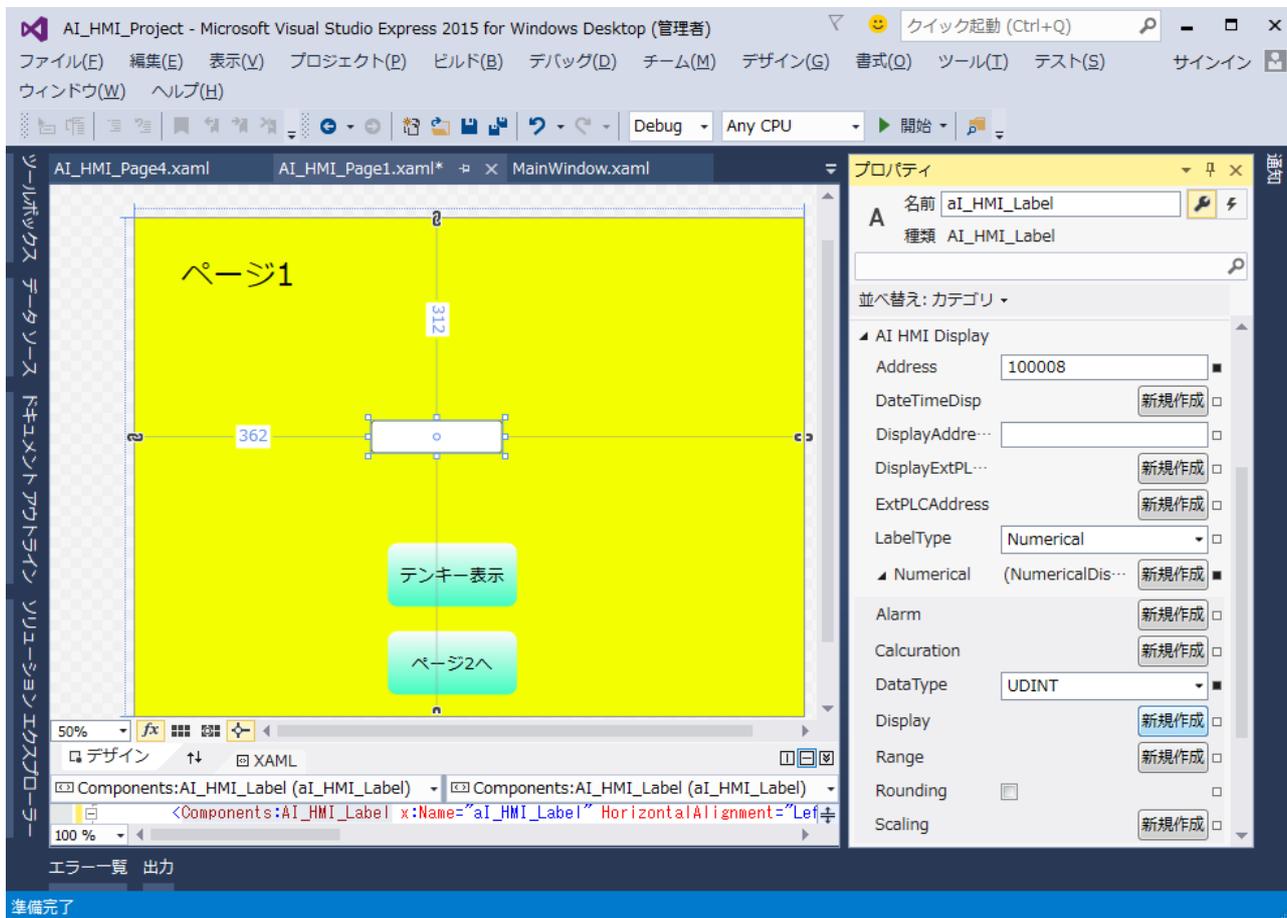


図 5-6-2 AI_HMI_Label の設定

表 5-6-1 AI_HMI_Label のプロパティ

項目	設定値	備考	
ブラシ	Background	任意の色	ラベルの背景色
レイアウト	Width	200	ラベルの幅
	Height	50	ラベルの高さ
テキスト	(フォントサイズ)	20pt	表示する文字の大きさ
AI HMI Display	Address	100008	ラベルで表示するデータのアドレス
	LabelType	Numerical	ラベルで表示するデータの種類
	DataType ※	UDINT	ラベルで表示するデータのデータ型

※ AI HMI Display – Numerical の「新規作成」ボタンをクリックすると表示されます。

- ③ ここまでの作業でデータ表示用のラベルの作成は完了しました。
『5-4 アプリケーションのコンパイルとデバッグ』の手順に従い、作成したプログラムをコンパイルしてアプリケーションを実行してください。

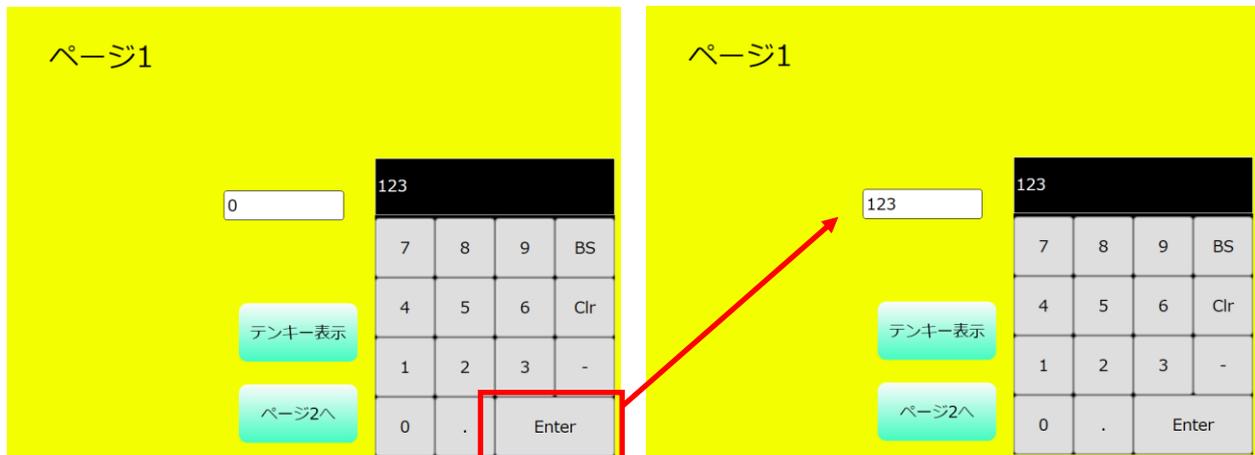


図 5-6-3 デバッグの実行

テンキーでデータ入力し「Enter」キーをクリックすると、データ表示用のラベルに入力したデータが表示されることを確認してください。

テンキーから入力してもラベルにデータが表示されない場合は手順②を確認してください。

第6章 作成したアプリの実行環境 PC での動作

前章で作成したサンプルアプリケーションを実際に実行環境 PC で動作させます。

6-1 Release モードでのコンパイル

VisualStudio のコンパイルモードを Release モードにしてからコンパイルを実行します。

- ① 上部メニューのコンパイルモードメニューから「Release」を選択してください。

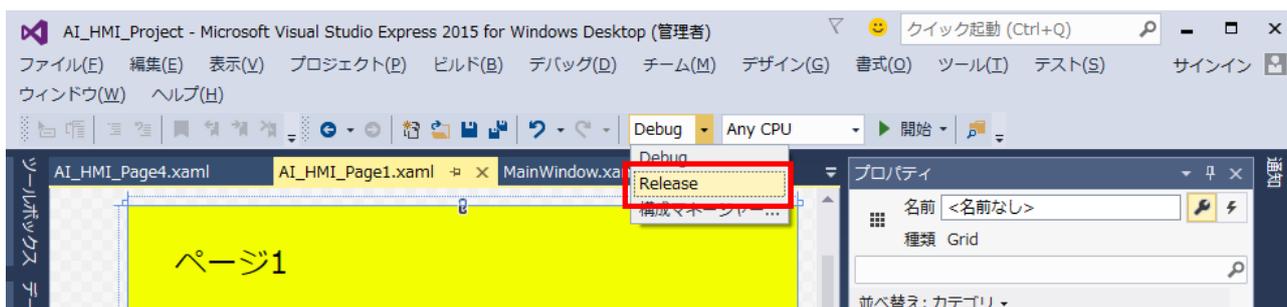


図 6-1-1 Release モードに変更

- ② 上部メニューの「ビルド - ソリューションのリビルド」を選択してください。
コンパイルが開始するので、コンパイルが完了するまでしばらくお待ちください。

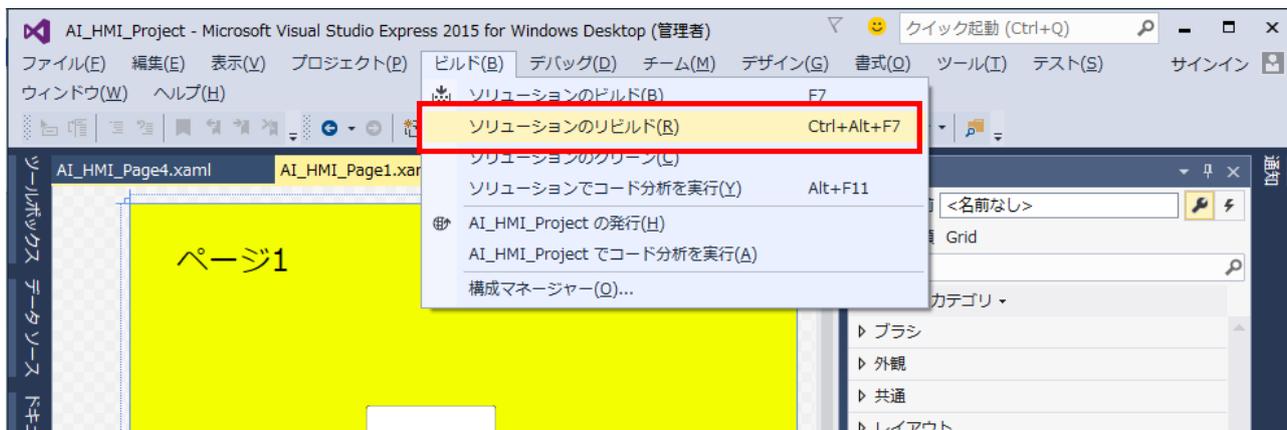


図 6-1-2 コンパイルの実行

- ③ コンパイルが完了すると以下のフォルダに実行ファイルが生成されます。

[プロジェクトのフォルダ]¥bin¥Release

生成された Release フォルダの中身全てを USB メモリなどにコピーするなどして実行環境 PC へコピーしてください。

名前	更新日時	種類	サイズ
Alarm	2017/07/14 10:08	ファイル フォル...	
en-US	2017/07/14 10:08	ファイル フォル...	
PageList	2017/07/14 10:08	ファイル フォル...	
Sampling	2017/07/14 10:08	ファイル フォル...	
User	2017/07/14 10:08	ファイル フォル...	
AI_HMI_CTRL.dll	2017/07/05 14:51	アプリケーション...	1,292 KB
AI_HMI_Project1.exe	2017/07/14 10:08	アプリケーション	16 KB
AI_HMI_Project1.exe.config	2017/07/14 10:07	XML Configurati...	1 KB
AI_HMI_Project1.pdb	2017/07/14 10:08	Program Debug...	50 KB
AI_HMI_Project1.vshost.exe	2017/07/14 10:07	アプリケーション	23 KB
AI_HMI_Project1.vshost.exe.config	2017/07/14 10:07	XML Configurati...	1 KB
AI_HMI_Project1.vshost.exe.manifest	2015/10/22 9:54	MANIFEST ファ...	1 KB
AsdSram.dll	2015/10/26 13:10	アプリケーション...	8 KB
IFSLMPCConnect.dll	2016/08/26 13:17	アプリケーション...	45 KB
System.Windows.Controls.DataVisu...	2016/01/05 11:40	アプリケーション...	344 KB
System.Windows.Controls.DataVisu...	2016/01/05 11:40	XML ファイル	616 KB

図 6-1-3 実行ファイルの生成

- ④ 実行環境 PC 上の任意のフォルダに先ほどのファイルとフォルダをコピーしてください。
コピー後、AI_HMI_Project.exe を実行してください。

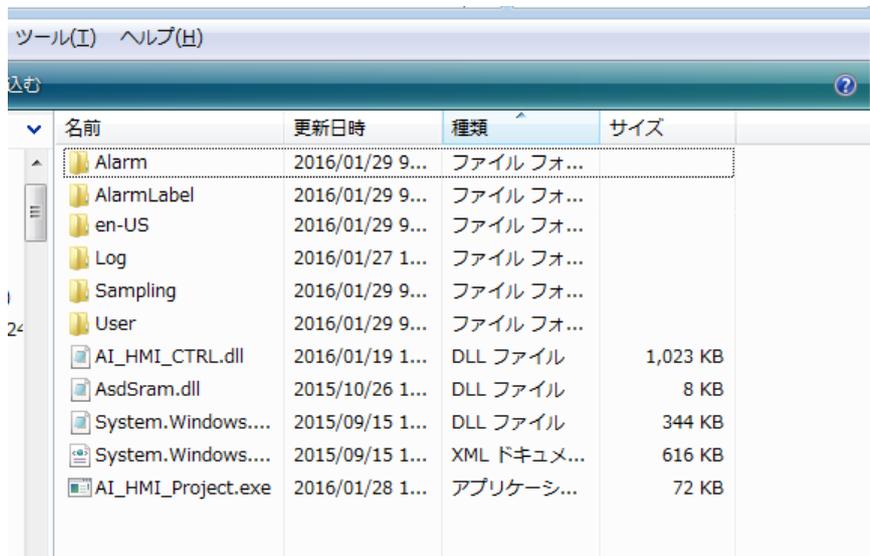


図 6-1-4 実行ファイルのコピー

実行環境 PC 上で作成したアプリケーションが動作することを確認してください。



図 6-1-5 実行環境 PC でのアプリケーション実行

第7章 AI-HMI デバッグツール

AI-HMI では実際にアプリケーションの開発を行うにあたって、共有メモリ上のデータを参照するためのデバッグツールを標準で提供しています。

AI-HMI デバッグツールは添付の CD-R の以下のフォルダ内に格納されています。

[CD-R]¥Tool¥SHMEM¥ShareMemDump. exe

AI-HMI デバッグツールは開発環境 PC、実行環境 PC のどちら上でも使用できます。

使用する場合は ShareMemDump. exe をダブルクリックして実行してください。

7-1 デバッグツールの概要

表示先頭アドレス、表示サイズ、表示データタイプを指定し、更新にチェックを入れることで現在の共有メモリのデータを参照することができます。



図 7-1-1 AI-HMI デバッグツール

表示先頭アドレス : 参照したいメモリの先頭アドレスを入力してください。

表示サイズ : 参照したいメモリのサイズを入力してください。

表示データタイプ : 参照したいメモリのデータタイプを選択してください。

(BOOL、INT、UINT、DINT、UDINT、REAL、LREAL)

更新 : チェックを入れることで共有メモリのデータを「データ」欄に周期的に取得します。

書込み : チェックを入れることで「書込みデータ」欄のデータを共有メモリへ周期的に書込みます。

データ表示形式 : 取得したデータの表示形式を Decimal (10 進数) か Hex (16 進数) かを指定します。

7-2 デバッグツールの使用方法

AI-HMI デバッグツールを使用するには以下の手順で操作してください。
今回は D100100 から 100 バイト分 UDINT 形式で取得する場合を例にとります。

- ① ShareMemDump.exe をダブルクリックし、AI-HMI デバッグツールを起動してください。

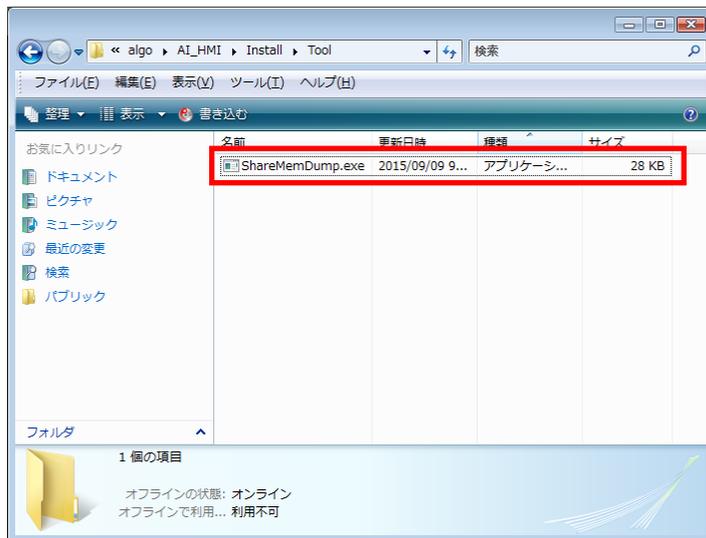


図 7-2-1 AI-HMI デバッグツールの起動

- ② AI-HMI デバッグツールが起動します。
設定欄に以下の値を入力してください。
先頭表示アドレス : 100100
表示サイズ : 100

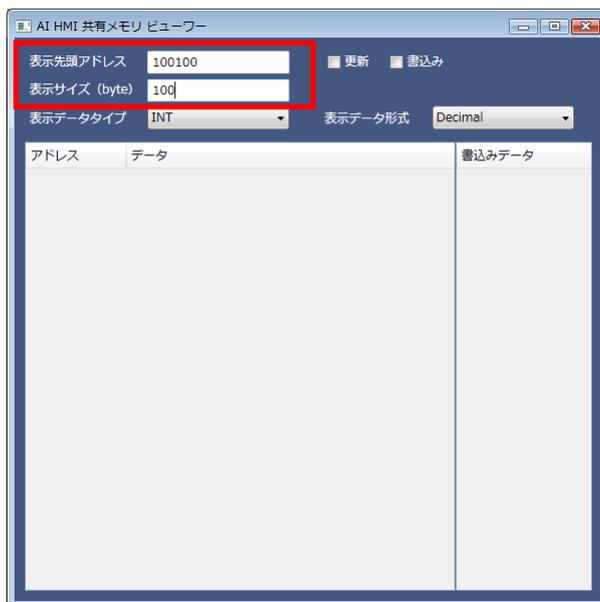


図 7-2-2 参照アドレスの設定

- ③ 「表示データタイプ」から UDINT を選択してください。

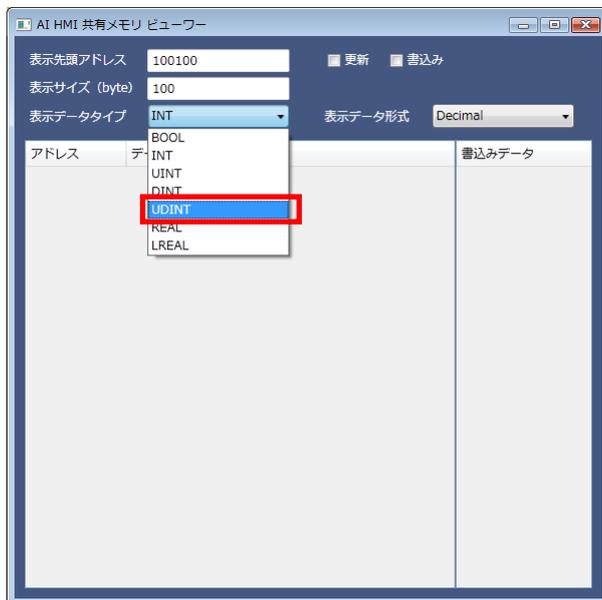


図 7-2-3 データ型の設定

- ④ 「更新」にチェックを入れてください。
各アドレスに格納されているデータが画面下部の「データ」欄に表示されます。



図 7-2-4 データの読み込み

- ⑤ 共有メモリに値を入力したい場合は「書き込み」にチェックを入れてください。チェックを入れた状態で画面下部の「書き込みデータ」に数値を入力すると値が共有メモリに書き込まれます。

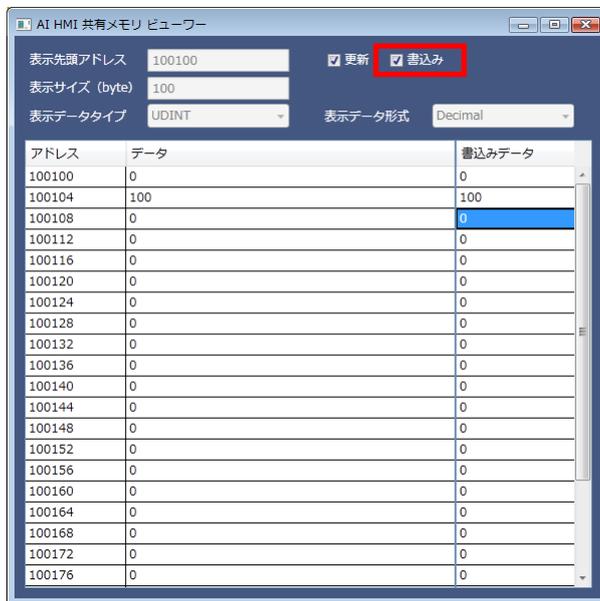


図 7-2-5 データの書き込み

- ⑥ 「表示データ形式」を Hex にすることで表示する値を 16 進数表示することができます。



図 7-2-6 表示形式の変更

7-3 画面アプリケーションとデバッグツールの連動

AI-HMI デバッグツールで共有メモリの値を変更することで、同じ共有メモリを参照しているアプリケーションの画面を変化させることができます。

前章までで作成したアプリケーションが参照する共有メモリのアドレスは表 7-3-1 のようになっています。

表 7-3-1 使用している共有メモリアドレス

アドレス	データ型	用途
100000	UINT 型	ページの切り替え変数
100002	UINT 型	ダイアログ表示切り替え変数
100008	UDINT 型	テンキー入力先変数
100012	UDINT 型	テンキー入力時のデータ型番号
140000	UDINT 型	テンキー入力先の共有メモリアドレスを格納する変数

これらの共有メモリの値を AI-HMI デバッグツールで書き換えることで、画面の切り替えやダイアログの表示、表示する値の変更ができることを確認します。

- ① AI-HMI デバッグツールと作成したアプリケーションを平行して起動してください。
AI-HMI デバッグツールから共有メモリアドレス 100000 を 0⇔1 と変化させることで、ページ1 とページ2 が切り替わることを確認してください。

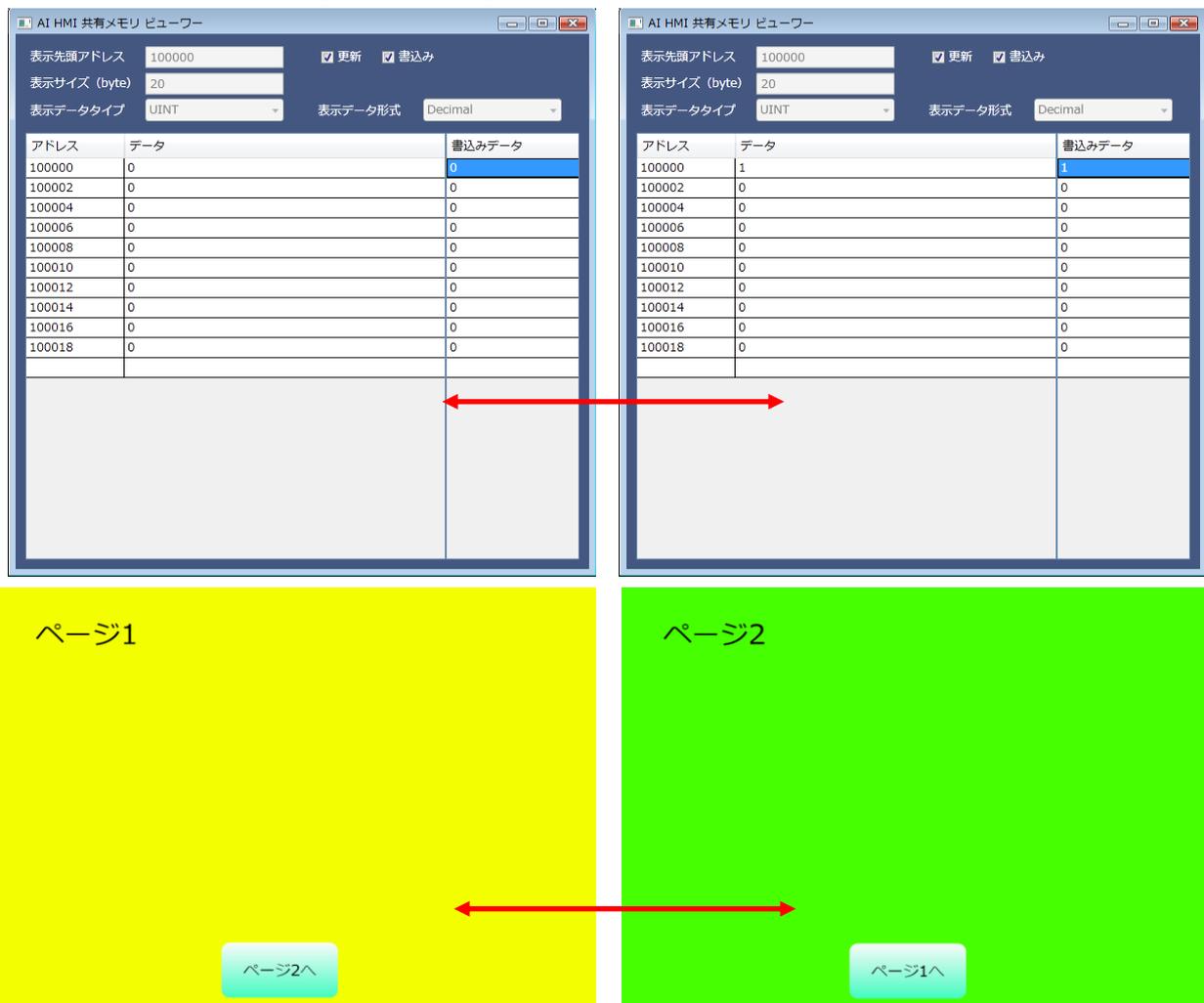


図 7-3-1 アプリケーションとの連動(画面切り替え)

- ② AI-HMI デバッグツールから共有メモリアドレス 100002 を 0⇔1 と変化させることで、ダイアログ(テンキー)の非表示⇔表示が切り替わることを確認してください。

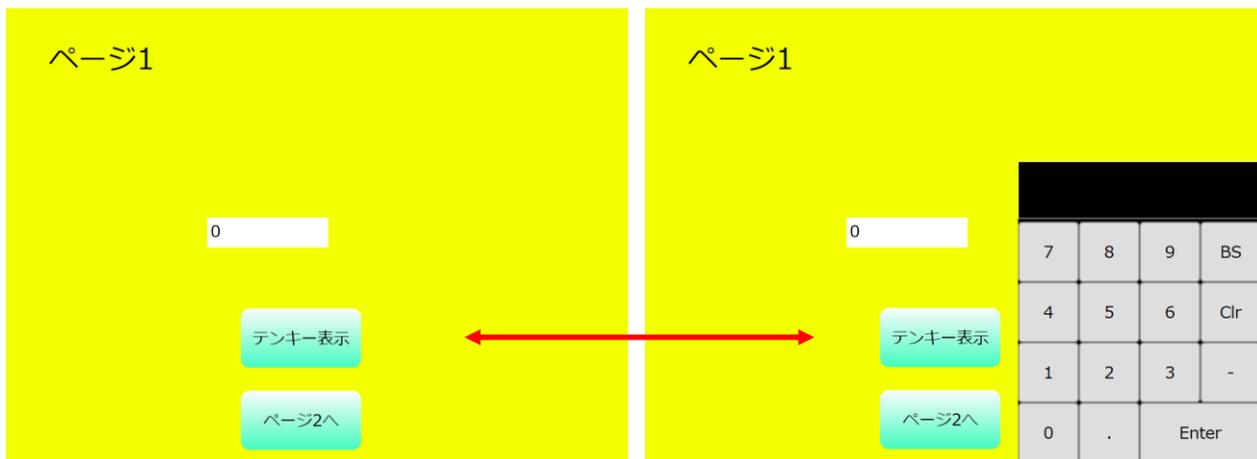
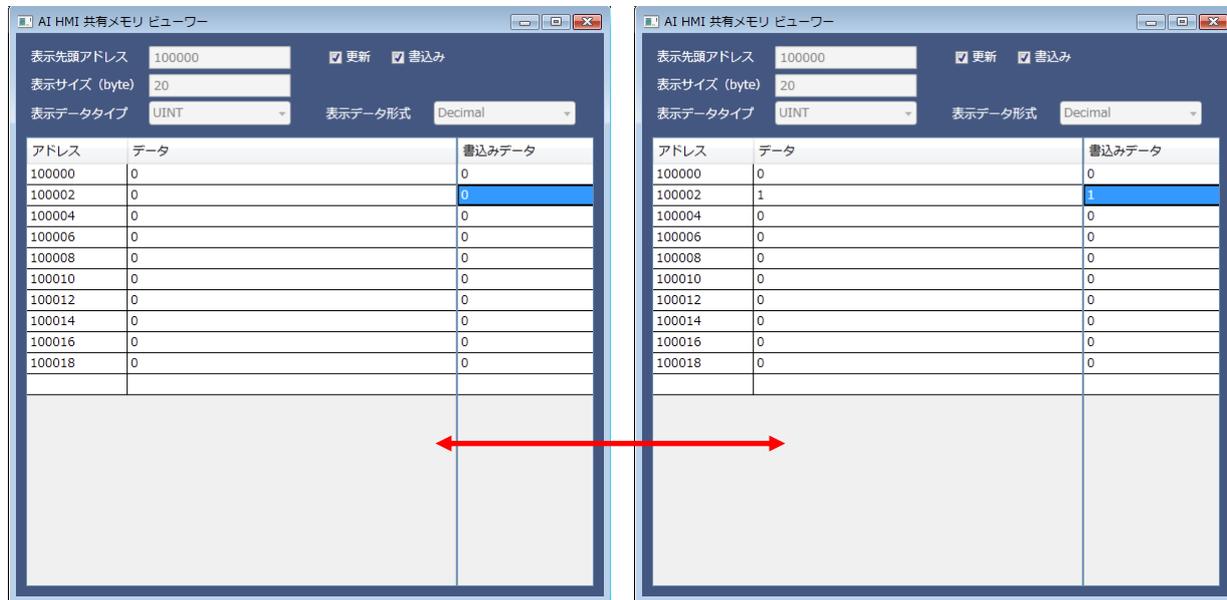


図 7-3-2 アプリケーションとの連動(ダイアログ表示)

- ③ AI-HMI デバッグツールから共有メモリアドレス 100008 の値を UDINT アクセスで変化させることで、ページ1のラベルの値が変化することを確認して下さい。

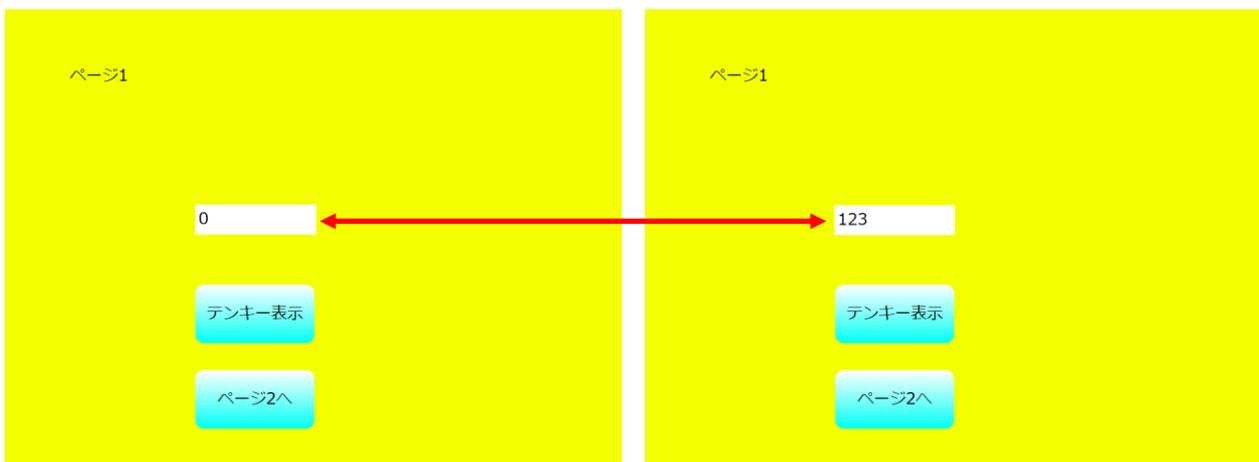
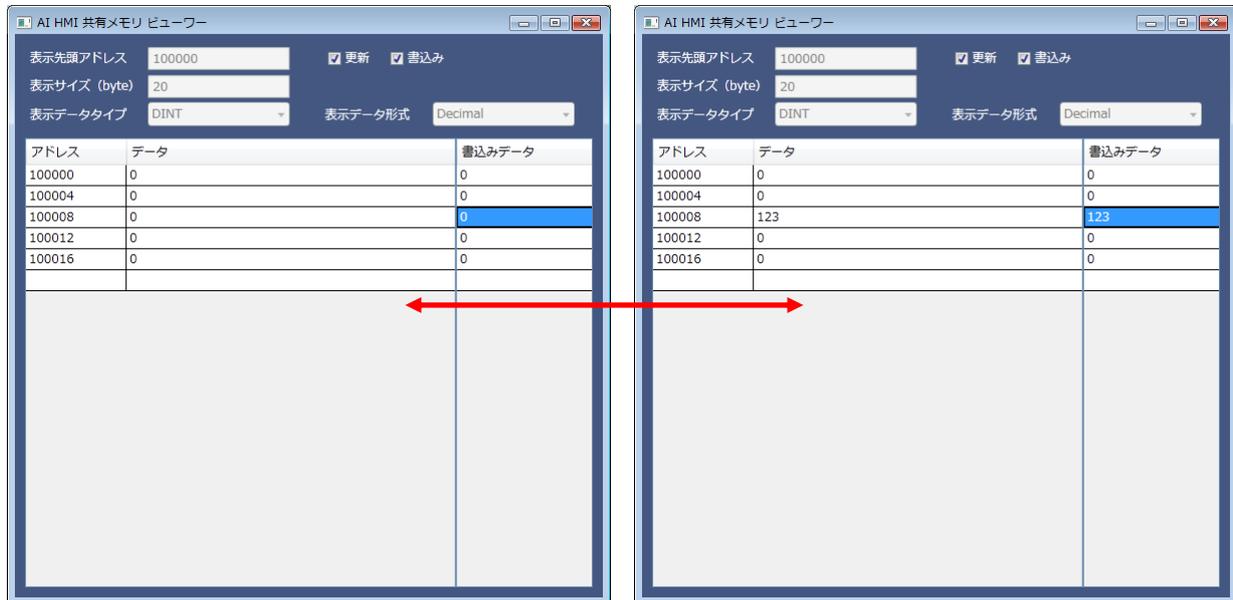


図 7-3-3 アプリケーションとの連動(表示する値の変更)

このように AI-HMI デバッグツールと画面アプリを連動して確認することで、AI-PLC の動作している環境上でもデバッグを行うことが可能です。

第8章 プロジェクト構成

実行環境上で AI-HMI を動作させるために必要なファイル構成を図 8-1 に示します。
これらのファイルは実行環境上の同じ任意のディレクトリ上に配置する必要があります。

```
└ (任意のディレクトリ)
  └ Alarm
    └ AI_HMI_AlarmGroupConfig.csv
    └ AI_HMI_AlarmKindConfig.csv
    └ AI_HMI_AlarmList.csv
  └ AlarmLabel
    └ AI_HMI_AlarmLabelList.csv
  └ Sampling
    └ AI_HMI_SamplingSetting.csv
  └ User
    └ AI_HMI_UserList.csv
  └ en-US
    └ XXXXX.resources.dll
  └ (その他の言語ファイル)
  └ PageList
    └ AI_HMI_PageList.csv
  └ AI_HMI_CTRL.dll
  └ AsdSram.dll
  └ System.Windows.Controls.DataVisualization.Toolkit.dll
  └ System.Windows.Controls.DataVisualization.Toolkit.xml
  └ XXXXXX.exe
    (XXXXはプロジェクト名)
```

図 8-1. ファイル構成

これらのファイルは開発環境上では実行ファイルの出力先フォルダに生成されます。
AI_HMI_CTRL.dll および exe ファイルはプロジェクトのビルド毎に生成されます。

それ以外のファイルはプロジェクトのリビルド毎に生成されます。

開発環境の実行ファイル出力先フォルダにある csv ファイルはリビルドの際に上書きされるのでご注意ください。

開発環境上で csv ファイルの編集を行う場合、プロジェクトフォルダ (AI_HMI_Page1 等のソースが格納されているフォルダ) 内の Alarm、AlarmLabel、Sampling、User 内にある csv ファイルを編集してください。プロジェクトのリビルドを実行するとこのファイルが実行ファイルの出力先フォルダにコピーされる形になります。

第9章 その他の機能の使用方法

第5章で紹介した機能以外にも AI-HMI は様々な機能があります。
本章ではそれぞれの機能について説明をしていきます。

9-1 アラーム機能

アラーム機能は、装置で発生したアラーム情報を表示させ、発生した日時をログとして保存することが出来ます。本項では、アラーム表示画面を作成します。

発生日時	復旧日時	回数	アラーム名	表示種別	グループ	優先順位	アラーム番
2016/07/06 19:03	-	2	アラーム1	軽故障	グループ	1	100
2016/07/06 19:03	2016/07/06 19:03	1	アラーム1	軽故障	グループ	1	100
2016/07/06 19:03	2016/07/06 19:03	1	アラーム2	軽故障	グループ	1	101

図 9-1-1. アラーム表示画面

9-1-1 アラームリスト

特定の Bit の変化を検知してアラームとして表示、記録する機能です。

- ① MainWindow.xaml を開き、「ドキュメントアウトライン - [AI_HMI_Window]」を選択してください。
プロパティ欄に AI HMI Window のプロパティが表示されます。

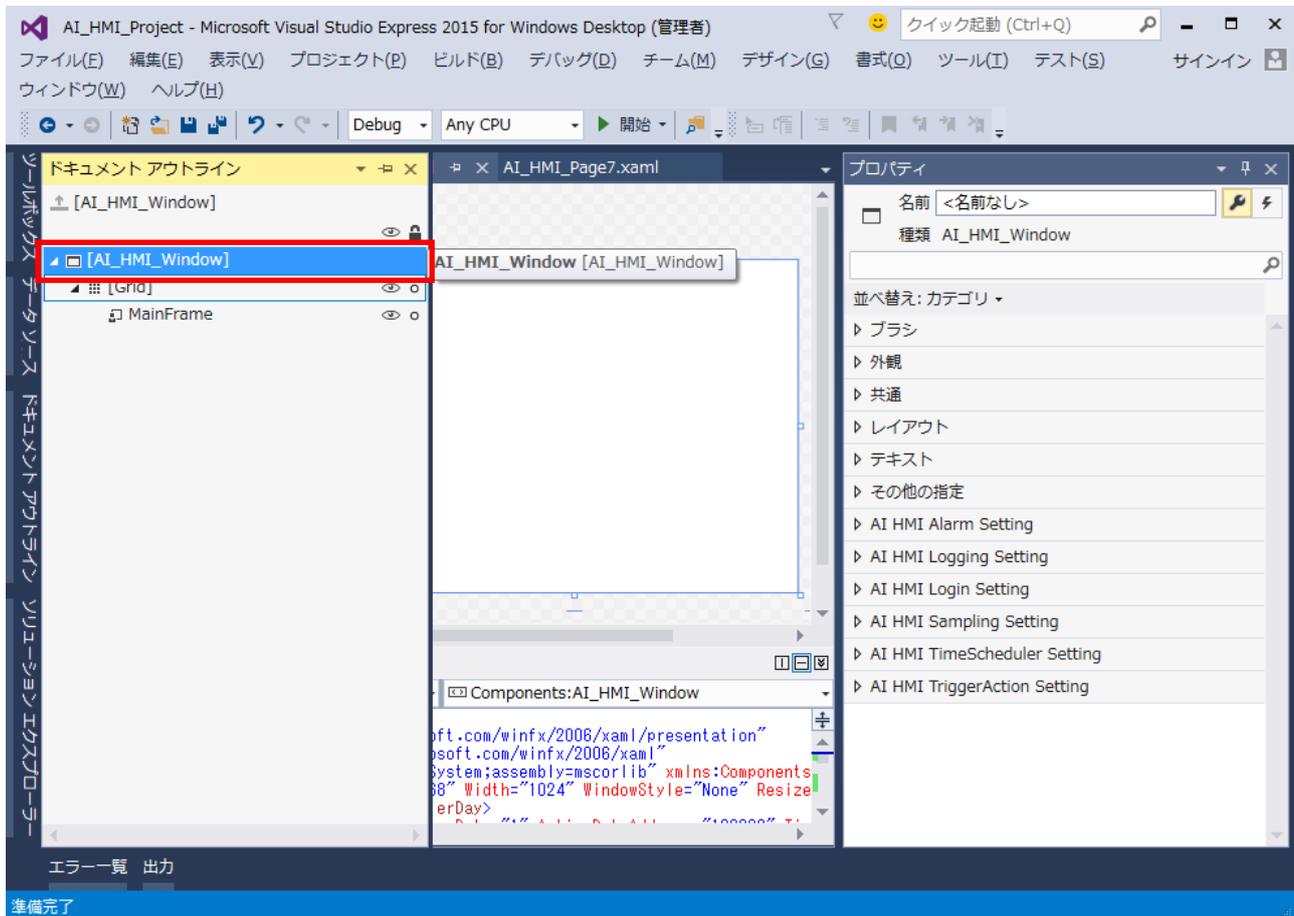


図 9-1-1-1 AI HMI Window プロパティ

- ② プロパティから「AI HMI Alarm Setting」を選択してください。
AI HMI Alarm Setting プロパティが開きます。

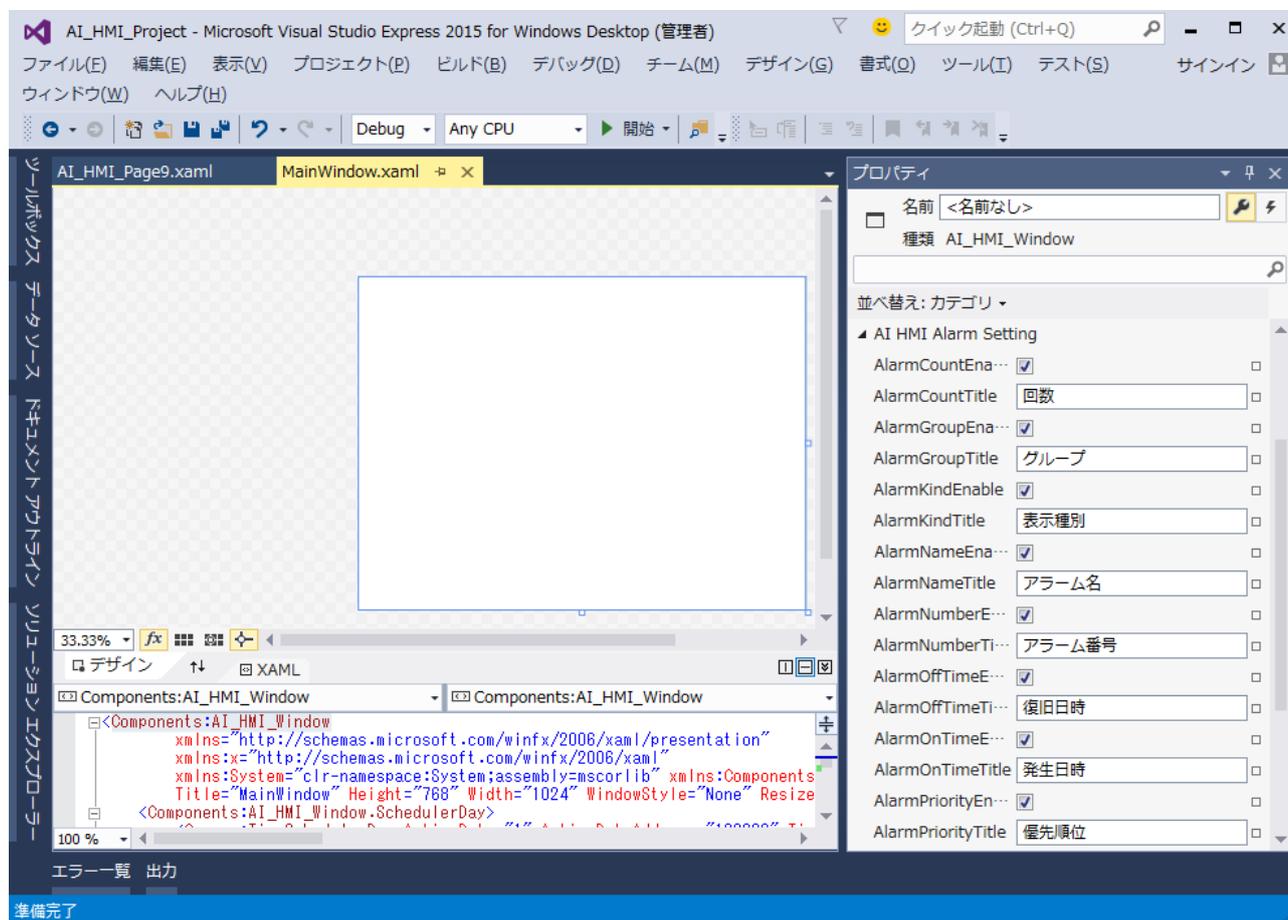


図 9-1-1-2 AI HMI Alarm Setting プロパティ

AI HMI Alarm Setting プロパティのそれぞれの意味は以下の通りです。

表 9-1-1-1 AI HMI Alarm Setting (1/2)

項目	データ型	初期値	用途
AlarmCountEnable	チェックボックス	チェックあり	アラーム発生時に「回数」項目を記録するかを設定します。
AlarmCountTitle	String 型	回数	アラーム発生時に CSV に保存する際の「回数」項目のタイトル文字列を設定します。
AlarmGroupEnable	チェックボックス	チェックあり	アラーム発生時に「グループ」項目を記録するかを設定します。
AlarmGroupTitle	String 型	グループ	アラーム発生時に CSV に保存する際の「グループ」項目のタイトル文字列を設定します。
AlarmKindEnable	チェックボックス	チェックあり	アラーム発生時に「表示種別」項目を記録するかを設定します。
AlarmKindTitle	String 型	表示種別	アラーム発生時に CSV に保存する際の「表示種別」項目のタイトル文字列を設定します。

表 9-1-1-1 AI HMI Alarm Setting(2/2)

項目	データ型	初期値	用途
AlarmNameEnable	チェックボックス	チェックあり	アラーム発生時に「アラーム名」項目を記録するかを設定します。
AlarmNameTitle	String 型	アラーム名	アラーム発生時に CSV に保存する際の「アラーム名」項目のタイトル文字列を設定します。
AlarmNumberEnable	チェックボックス	チェックあり	アラーム発生時に「アラーム番号」項目を記録するかを設定します。
AlarmNumberTitle	String 型	アラーム番号	アラーム発生時に CSV に保存する際の「アラーム番号」項目のタイトル文字列を設定します。
AlarmOffTimeEnable	チェックボックス	チェックあり	アラーム発生時に「復旧日時」項目を記録するかを設定します。
AlarmOffTimeTitle	String 型	復旧日時	アラーム発生時に CSV に保存する際の「復旧日時」項目のタイトル文字列を設定します。
AlarmOnTimeEnable	チェックボックス	チェックあり	アラーム発生時に「発生日時」項目を記録するかを設定します。
AlarmOnTimeTitle	String 型	発生日時	アラーム発生時に CSV に保存する際の「発生日時」項目のタイトル文字列を設定します。
AlarmPriorityEnable	チェックボックス	チェックあり	アラーム発生時に「優先順位」項目を記録するかを設定します。
AlarmPriorityTitle	String 型	優先順位	アラーム発生時に CSV に保存する際の「優先順位」項目のタイトル文字列を設定します。
AlarmSaveCsvPath	String 型	初期値なし	アラームデータの CSV ファイルの保存先フォルダを設定します。設定は相対パス、絶対パス共に可能です。相対パスの場合、画面アプリの exe ファイルからみた相対パスとなります。
IsAlarmEnable	チェックボックス	チェックなし	アラームデータの CSV ファイル保存機能の有効無効を切り替えます。チェックを入れると操作ログを保存するようになります。

- ③ アラームリストに使用するアラーム項目は AI_HMI_AlarmList.csv に記述します。
AI_HMI_AlarmList.csv は画面アプリの exe ファイルと同じフォルダ内の「Alarm」フォルダに格納します。
AI_HMI_AlarmList.csv の内容の詳細は以下のようになります。(エンコード形式 S-JIS)

リスト 9-1-1-1 AI_HMI_AlarmList.csv の内容

アラーム番号	アラーム名	変数名	検知パターン	表示種別	優先順位	グループ	発生文字色	復旧文字色
1,	異常 1,	80000,	1,	H,	1,	0,	FFFF00,	00FFFF
2,	異常 2,	80001,	1,	H,	1,	0,	FF00FF,	00FF00
3,	異常 3,	80002,	2,	H,	1,	1,	FF0000,	0000FF
4,	イベント,	80003,	1,	E,	2,	1,	FF0000,	0000FF

表 9-1-1-2 AI_HMI_AlarmList.csv の内容詳細(1/2)

項目	内容
アラーム番号	各アラームの識別番号です。 「アラーム番号」は重複しない1~5000までの整数値を設定する必要があります。 (連続する整数である必要はありません) 「アラーム番号」で設定したアラーム番号はアラーム発生時にアラーム保存ファイルのアラーム番号の列に記録されます。
アラーム名	各アラームの名称です。 「アラーム名」は文字列として扱います。 「アラーム名」で設定した文字列はアラーム発生時にアラーム保存ファイルのアラーム名の列に記録されます。
変数名	各アラームを発生させるトリガです。 「変数名」は変数または共有メモリアドレスを設定します。 「変数名」で設定した変数が「検知パターン」の条件を満たしたとき、そのアラームが発生したと判断します。
検知パターン	各アラームを発生させるトリガの検知パターンです。 「検知パターン」は1または2を設定する必要があります。 ・1 : 立上り検知 ・2 : 立下り検知 「変数名」で設定した変数が「検知パターン」の条件を満たしたとき、そのアラームが発生したと判断します。
表示種別	各アラームの表示種別です。 「表示種別」は AI_HMI_AlarmKindConfig.csv で設定するアルファベット1文字を設定する必要があります。 「表示種別」で設定した文字はアラーム発生時にアラーム保存ファイルの表示種別の列に AI_HMI_AlarmKindConfig.csv で設定する名称に置き換えて記録されます。
優先順位	各アラームの優先順位です。 「優先順位」は整数値を設定する必要があります。 「優先順位」で設定した値はアラーム発生時にアラーム保存ファイルの優先順位の列に記録されます。
グループ	各アラームのグループ名です。 「グループ」は AI_HMI_AlarmGroupConfig.csv で設定する整数値を設定する必要があります。 「グループ」で設定した値はアラーム発生時にアラーム保存ファイルのグループの列に AI_HMI_AlarmGroupConfig.csv で設定する名称に置き換えて記録されます。

表 9-1-1-2 AI_HMI_AlarmList.csv の内容詳細 (2/2)

項目	内容
発生文字色	各アラームの発生時の文字色です。 「発生文字色」は 6 ケタの RGB コードを設定する必要があります。 「発生文字色」で設定した RGB コードは AI HMI AlarmTable で表示する際の文字色になります。
復旧文字色	各アラームの復旧時の文字色です。 「復旧文字色」は 6 ケタの RGB コードを設定する必要があります。 「復旧文字色」で設定した RGB コードは AI HMI AlarmTable で表示する際の文字色になります。

AI_HMI_AlarmList.csv の表示種別に対応する文字列は AI_HMI_AlarmKindConfig.csv に記述します。

AI_HMI_AlarmKindConfig.csv は画面アプリの exe ファイルと同じフォルダ内の「Alarm」フォルダに格納します。

AI_HMI_AlarmKindConfig.csv の内容の詳細は以下のようになります。(エンコード形式 S-JIS)

リスト 9-1-1-2 AI_HMI_AlarmKindConfig.csv の内容

表示種別,	名称
H,	重故障
M,	中故障
L,	軽故障
E,	イベント

表 9-1-1-3 AI_HMI_AlarmKindConfig.csv の内容詳細

項目	内容
表示種別	AI_HMI_AlarmList.csv の表示種別に対応する文字です。 「表示種別」は重複しないアルファベット 1 文字である必要があります。
名称	アラーム発生時に表示種別の列に表示する文字列です。 「名称」は文字列として扱います。 「表示種別」の文字に一致する AI_HMI_AlarmList.csv の表示種別がある場合、「名称」の文字列を表示します。

AI_HMI_AlarmList.csv のグループに対応する文字列は AI_HMI_AlarmGroupConfig.csv に記述します。
AI_HMI_AlarmGroupConfig.csv は画面アプリの exe ファイルと同じフォルダ内の「Alarm」フォルダに格納します。

AI_HMI_AlarmGroupConfig.csv の内容の詳細は以下のようになります。(エンコード形式 S-JIS)

リスト 9-1-1-3 AI_HMI_AlarmGroupConfig.csv の内容

グループ,	名称
0,	グループ 1
1,	グループ 2
2,	グループ 3
3,	グループ 4

表 9-1-1-4 AI_HMI_AlarmGroupConfig.csv の内容詳細

項目	内容
グループ	AI_HMI_AlarmList.csv のグループに対応する文字です。 「グループ」は重複しない整数値である必要があります。
名称	アラーム発生時にグループの列に表示する文字列です。 「名称」は文字列として扱います。 「グループ」の文字に一致する AI_HMI_AlarmList.csv のグループがある場合、「名称」の文字列を表示します。

- ④ アラームリストではアラームの保存、削除をする際にシステム変数を使用します。
使用するシステム変数は表 9-1-1-5 の通りです。

表 9-1-1-5 アラーム機能で使用するシステム変数

システム変数名	アドレス	内容
ALARM_CSV_SAVE	68504	1 を入力すると SRAM 領域のアラームデータを CSV ファイルに保存します。 保存が完了すると ALARM_CSV_SAVE は自動的に 0 に戻ります。
ALARM_ALL_CLEAR	68506	1 を入力すると SRAM 領域のアラームデータをすべてクリアします。 クリアが完了すると ALARM_CSV_SAVE は自動的に 0 に戻ります。
ALARM_CSV_ERROR	68508	アラームの CSV ファイルへの保存に失敗すると 1 が入ります。 ALARM_CSV_ERROR は次回アラームデータの保存時に 0 になります。

- ⑤ 実際に操作ログを保存できるように設定します。
AI HMI Alarm Setting プロパティ表 9-1-1-6 のように設定してください。

表 9-1-1-6 AI HMI Alarm Setting

項目	設定内容	備考
AlarmCountEnable	チェックあり	
AlarmCountTitle	回数	
AlarmGroupEnable	チェックあり	
AlarmGroupTitle	グループ	
AlarmKindEnable	チェックあり	
AlarmKindTitle	表示種別	
AlarmNameEnable	チェックあり	
AlarmNameTitle	アラーム名	
AlarmNumberEnable	チェックあり	
AlarmNumberTitle	アラーム番号	
AlarmOffTimeEnable	チェックあり	
AlarmOffTimeTitle	復旧日時	
AlarmOnTimeEnable	チェックあり	
AlarmOnTimeTitle	発生日時	
AlarmPriorityEnable	チェックあり	
AlarmPriorityTitle	優先順位	
AlarmSaveCsvPath	AlarmFile	画面アプリの exe ファイルと同じ階層に AlarmFile というフォルダを作成し、アラーム保存ファイルを保存します。
IsAlarmEnable	チェックあり	アラーム機能を有効にします。

- ⑥ アラームリストが動作することを確認します。
図 9-1-1-3 のように AI HMI AlarmTable と AI HMI Button を配置してください。
AI HMI AlarmTable の詳細については AI HMI リファレンスマニュアル『第 25 章 AI HMI AlarmTable』を参照ください。

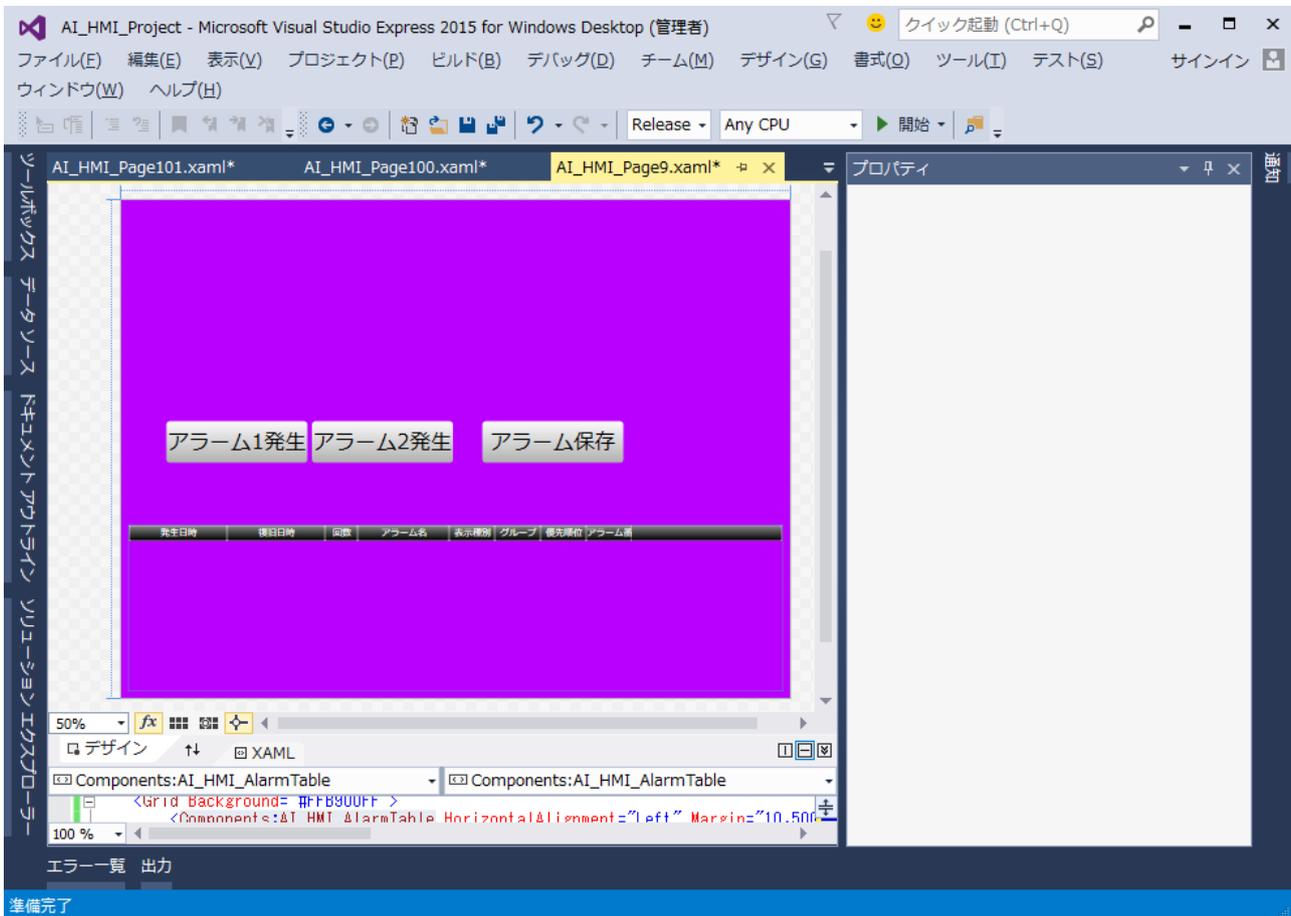


図 9-1-1-3 アラーム動作確認画面

AI_HMI_AlarmTableのプロパティは変更する必要はありません。
AI_HMI_Buttonのプロパティを以下のように変更してください。

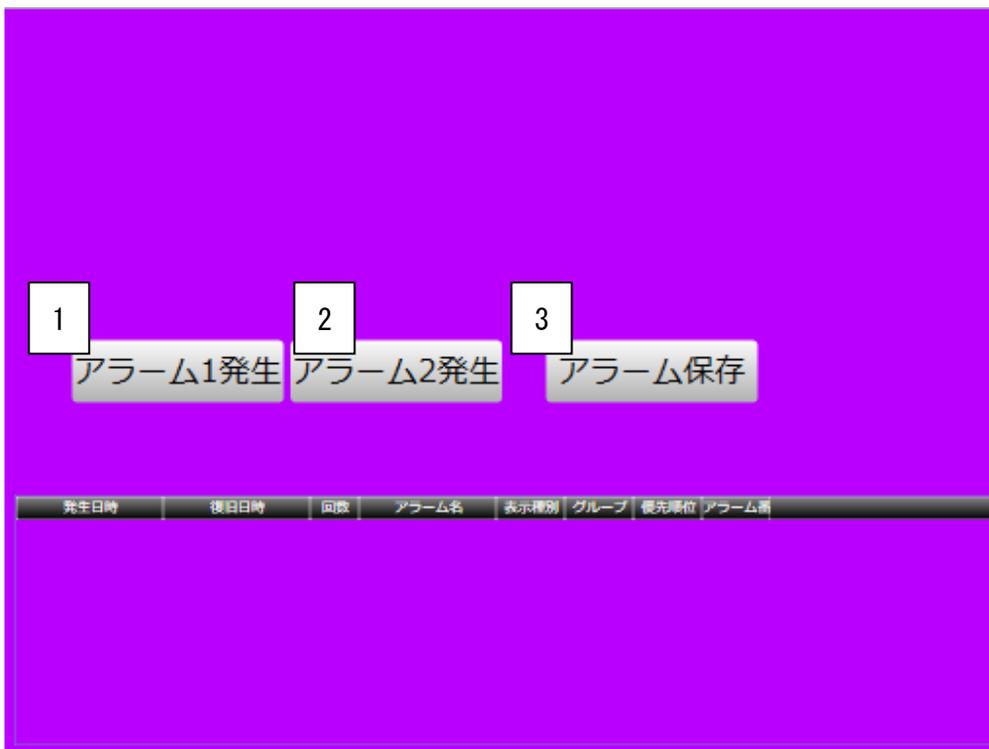


図 9-1-1-4 アラーム動作確認画面詳細

1. アラーム1を発生させるボタンです。AI HMI Buttonを使用します。

表 9-1-1-7 アラーム1発生ボタンプロパティ

項目		設定値		備考
AI HMI Switch	OffDelaySwitch	Address	80100	OffDelaySwitchReference の右側の新規作成ボタンを押すことで表示されます。
	Reference	OffDelayTimer	1000	

2. アラーム2を発生させるボタンです。AI HMI Buttonを使用します。

表 9-1-1-8 アラーム2発生ボタンプロパティ

項目		設定値		備考
AI HMI Switch	OffDelaySwitch	Address	80101	OffDelaySwitchReference の右側の新規作成ボタンを押すことで表示されます。
	Reference	OffDelayTimer	1000	

3. アラームを保存させるボタンです。AI HMI Buttonを使用します。

表 9-1-1-9 アラーム保存ボタンプロパティ

項目		設定値		備考
AI HMI Switch	Switch Reference	Address	68504	SwitchReference の右側のボタンを押すと表示されるButtonSwitchReference ウィンドウ内で設定します。
		Data	1	

- ⑦ アラームリストの確認用に AI_HMI_AlarmList に以下の 2 行を追加します。

リスト 9-1-1-4 アラーム確認用アラームリストの追加

アラーム番号	アラーム名	変数名	検知パターン	表示種別	優先順位	グループ	発生文字色	復旧文字色
100,	アラーム 1,	80100,	1,	L,	1,	2,	FF0000,	0000FF
101,	アラーム 2,	80101,	1,	L,	1,	2,	FFFF00,	00FFFF

- ⑧ 作成した画面を実際に動作させてアラームリストが動作することを確認します。

『第6章 作成したアプリの実行環境PCでの動作』を参照し、作成した画面アプリを動作させてください。

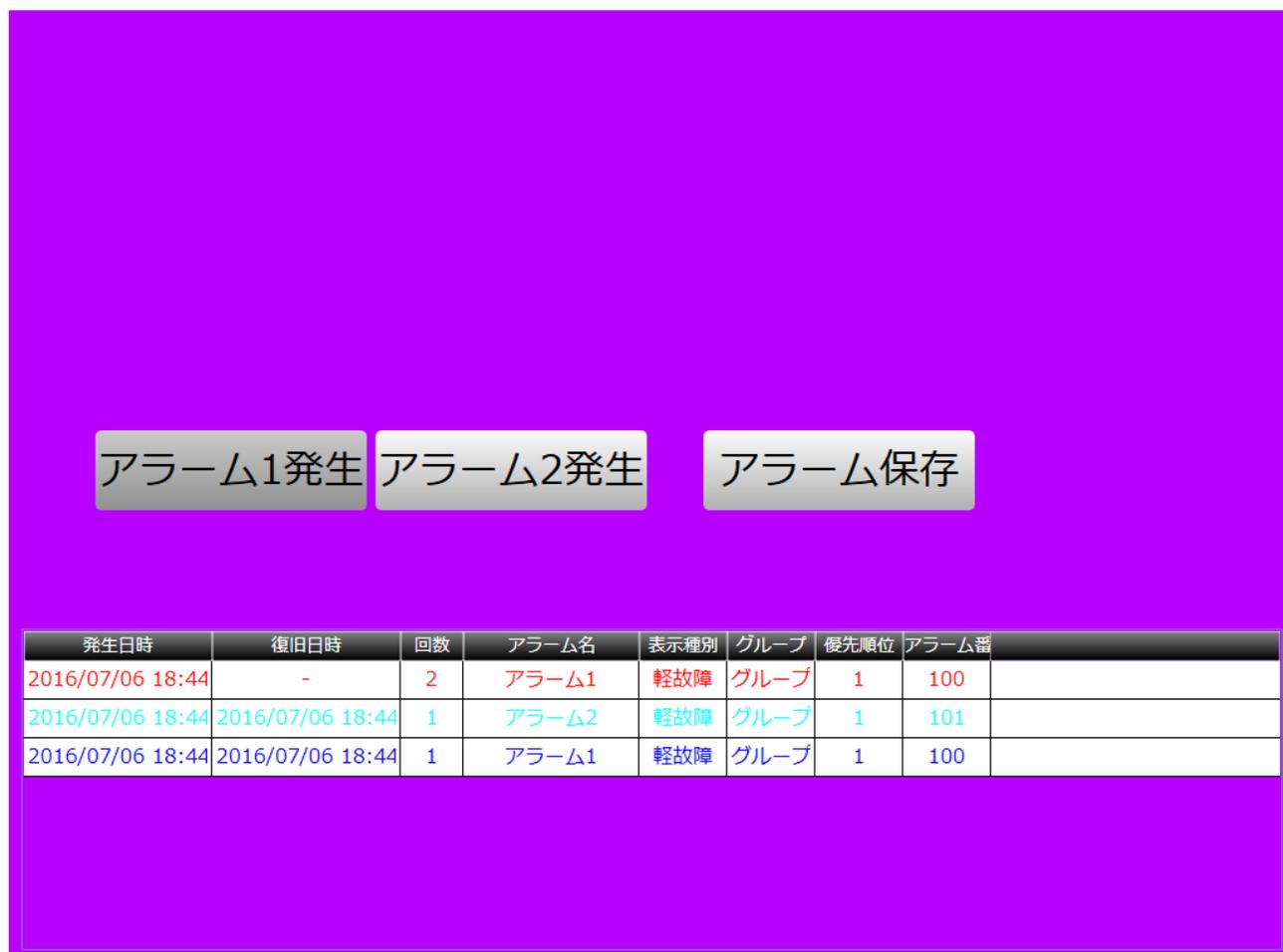


図 9-1-1-5 アラーム確認画面の実行

アラーム発生ボタンを押すことで AI HMI AlarmTable にアラームが 1 行追加されることを確認してください。

また、アラーム保存ボタンを押すことで exe ファイルと同じフォルダに AlarmFile フォルダが作成され、その中にアラーム保存ファイルが作成されることを確認してください。

9-1-2 アラームラベル

特定の Bit の変化を検知してアラームとしてメッセージを表示する機能です。

- ① アラームラベルに表示するメッセージは AI_HMI_AlarmLabelList.csv に記述します。
AI_HMI_AlarmLabelList.csv は画面アプリの exe ファイルと同じフォルダ内の「AlarmLabel」フォルダに格納します。

AI_HMI_AlarmLabelList.csv の内容の詳細は以下のようになります。(エンコード形式 S-JIS)

リスト 9-1-2-1 AI_HMI_AlarmLabelList.csv の内容

アラーム番号	アラーム名
1.	異常 1
2.	異常 2
3.	異常 3
4.	イベント

表 9-1-2-1 AI_HMI_AlarmLabelList.csv の内容詳細

項目	内容
アラーム番号	各アラームの識別番号です。 「アラーム番号」は重複しない 1~5000 までの整数値を設定する必要があります。 (連続する整数である必要はありません) 「アラーム番号」で設定したアラーム番号はアラーム発生時に AI HMI AlarmLabel に対応する文字列を表示させます。
アラーム名	各アラームの内容です。 「アラーム名」は文字列として扱います。 「アラーム名」で設定した文字列はアラーム発生時に AI HMI AlarmLabel 上に表示します。

- ② アラームラベルが機能することを確認します。
AI HMI Page100.xaml を作成し、図 9-1-2-1 のように AI HMI AlarmLabel と AI HMI Button および AI HMI Grid を配置してください。
AI HMI AlarmLabel の詳細に関しては AI HMI リファレンスマニュアル『第 28 章 AI HMI AlarmLabel』を参照ください。

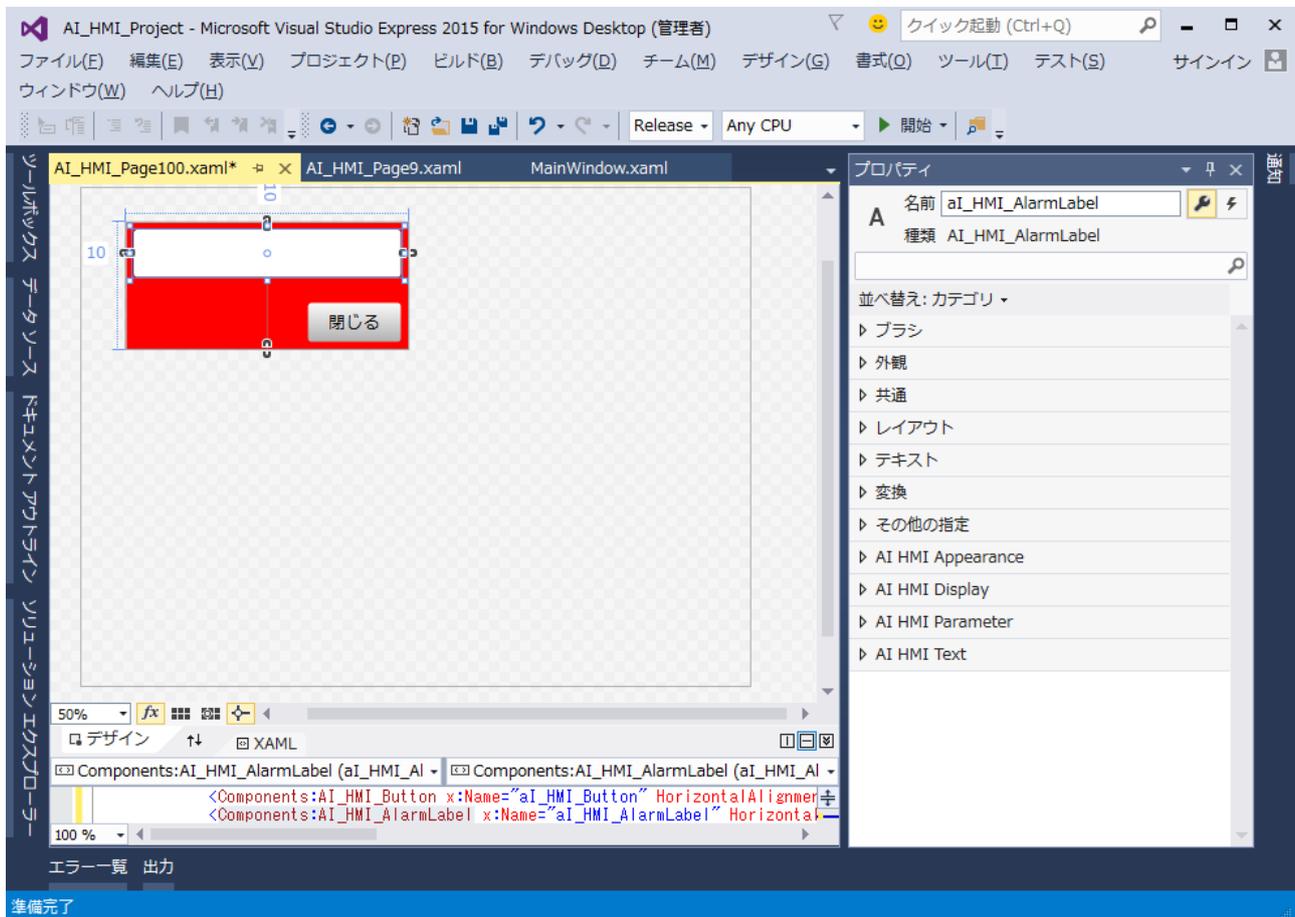


図 9-1-2-1 アラームラベル動作確認用画面 1

AI_HMI_AlarmLabel と AI_HMI_Button のプロパティを以下のように変更してください。

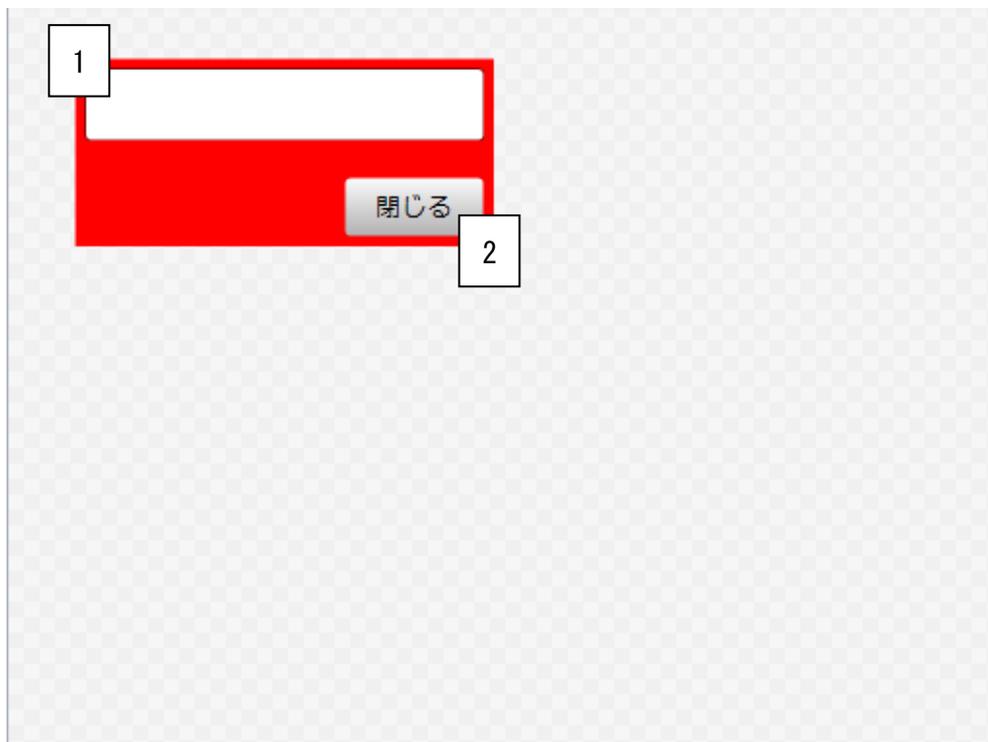


図 9-1-2-2 アラームラベル動作確認用画面詳細 1

1. アラーム内容を表示するラベルです。AI HMI AlarmLabel を使用します。

表 9-1-2-2 アラーム内容表示ラベルプロパティ

項目		設定値	備考
AI HMI Display	Address	100200	AlarmLabelList.csv のアラーム番号と比較する値が格納されたアドレスを指定します。

2. アラームラベルを閉じるためのボタンです。AI HMI Button を使用します。

表 9-1-2-3 閉じるボタンプロパティ

項目		設定値	備考
AI HMI Switch	Switch Reference	Address	100220
		Type	UINT
		Data	0
			アラームラベルを表示させる AI HMI Dialog を非表示にします。

- ③ 同様にもう1画面作成します。
AI HMI Page101.xaml を作成し、図 9-1-2-3 のように AI HMI AlarmLabel と AI HMI Button
および AI HMI Grid を配置してください。

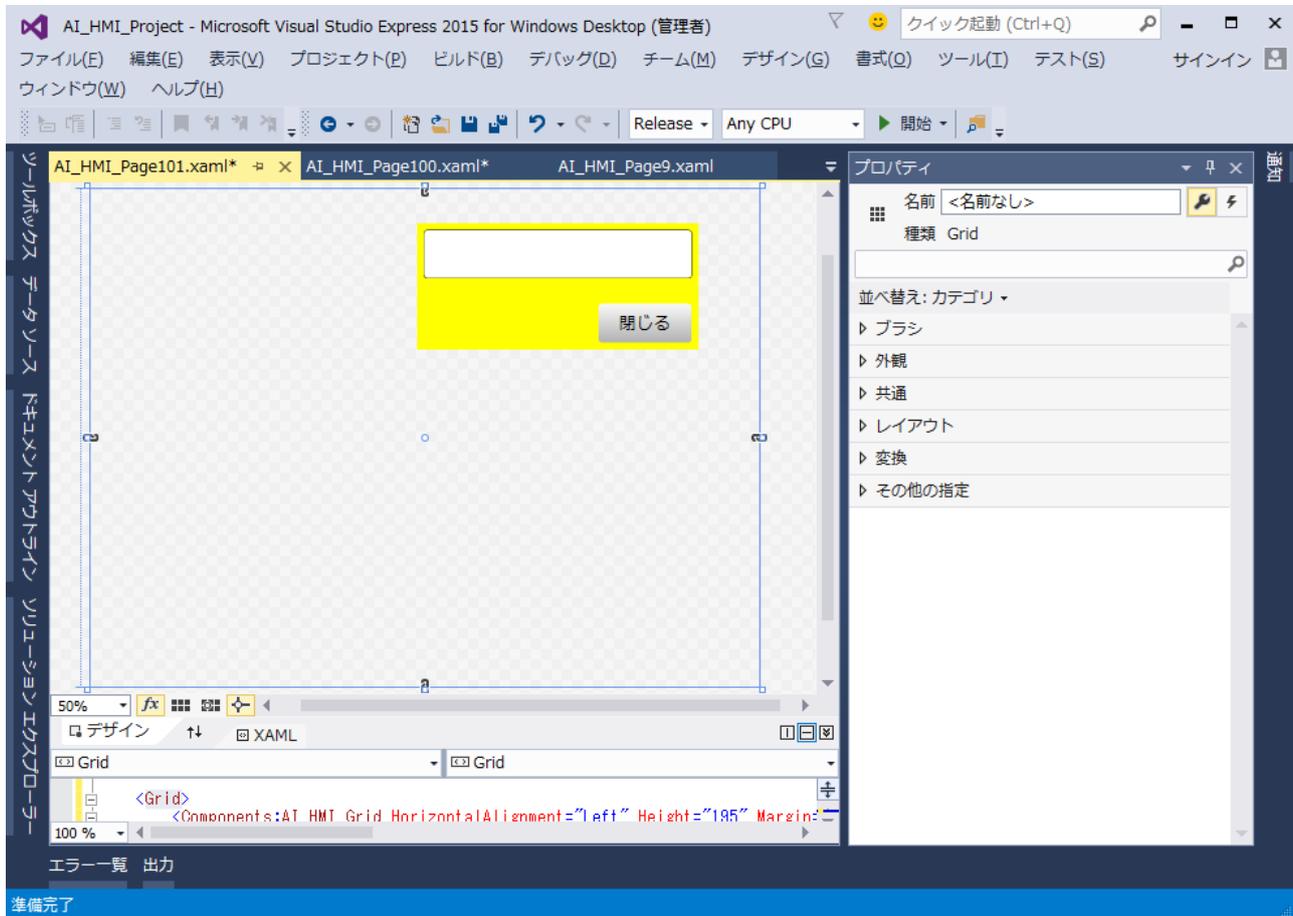


図 9-1-2-3 アラームラベル動作確認用画面 2

AI_HMI_AlarmLabel と AI_HMI_Button のプロパティを以下のように変更してください。

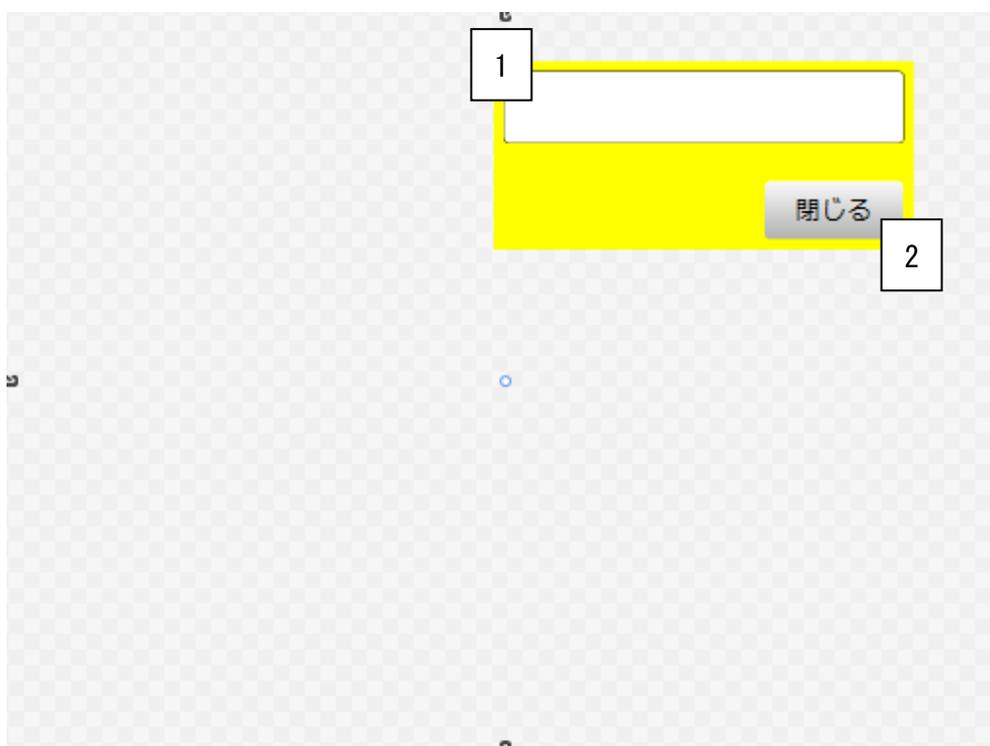


図 9-1-2-4 アラームラベル動作確認用画面詳細 2

1. アラーム内容を表示するラベルです。AI HMI AlarmLabel を使用します。

表 9-1-2-4 アラーム内容表示ラベルプロパティ

項目		設定値	備考
AI HMI Display	Address	100210	AlarmLabelList.csv のアラーム番号と比較する値が格納されたアドレスを指定します。

2. アラームラベルを閉じるためのボタンです。AI HMI Button を使用します。

表 9-1-2-5 閉じるボタンプロパティ

項目		設定値	備考
AI HMI Switch	Switch Reference	Address	100230
		Type	UINT
		Data	0
			アラームラベルを表示させる AI HMI Dialog を非表示にします。

- ④ 作成したポップアップ画面を AI HMI Dialog に表示させます。
MainWindow.xaml の DialogFrameReference2 を以下のように設定してください。

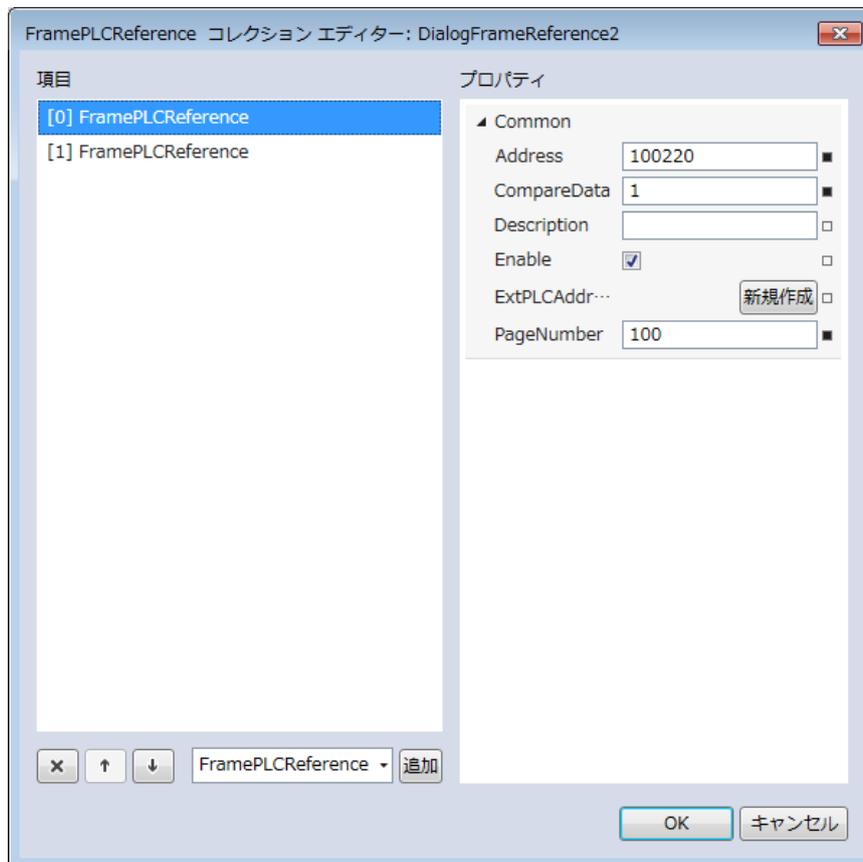


図 9-1-2-5 アラームラベルポップアップ表示設定 1

表 9-1-2-6 アラーム内容 1 表示用 Dialog プロパティ 1

項目		設定値	備考
AI HMI Frame	DialogFrame Reference2	Address	100220
		CompareData	1
		PageNumber	100
			アラームの内容をポップアップ表示します。

表 9-1-2-7 アラーム内容 1 表示用 Dialog プロパティ 2

項目		設定値	備考
AI HMI Frame	DialogFrame Reference2	Address	100220
		CompareData	0
		PageNumber	0
			アラームの内容のポップアップを非表示にします。

- ⑤ 同様に MainWindow.xaml の DialogFrameReference3 を以下のように設定してください。

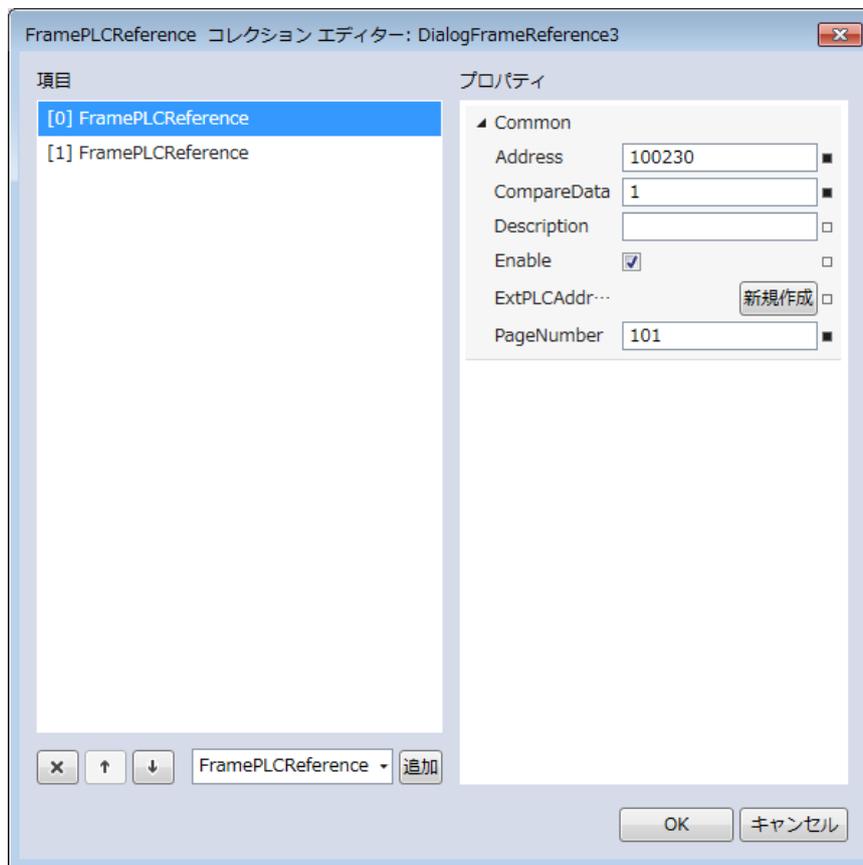


図 9-1-2-6 アラームラベルポップアップ表示設定 2

表 9-1-2-8 アラーム内容 2 表示用 Dialog プロパティ 1

項目		設定値	備考
AI HMI Frame	DialogFrame Reference2	Address	100230
		CompareData	1
		PageNumber	101
			アラームの内容をポップアップ表示します。

表 9-1-2-9 アラーム内容 2 表示用 Dialog プロパティ 2

項目		設定値	備考
AI HMI Frame	DialogFrame Reference2	Address	100230
		CompareData	0
		PageNumber	0
			アラームの内容のポップアップを非表示にします。

- ⑥ 『9-1-1 アラームリスト』で作成した画面からアラーム内容のポップアップを表示できるようにします。アラーム発生ボタンのプロパティに以下のように設定を追加してください。

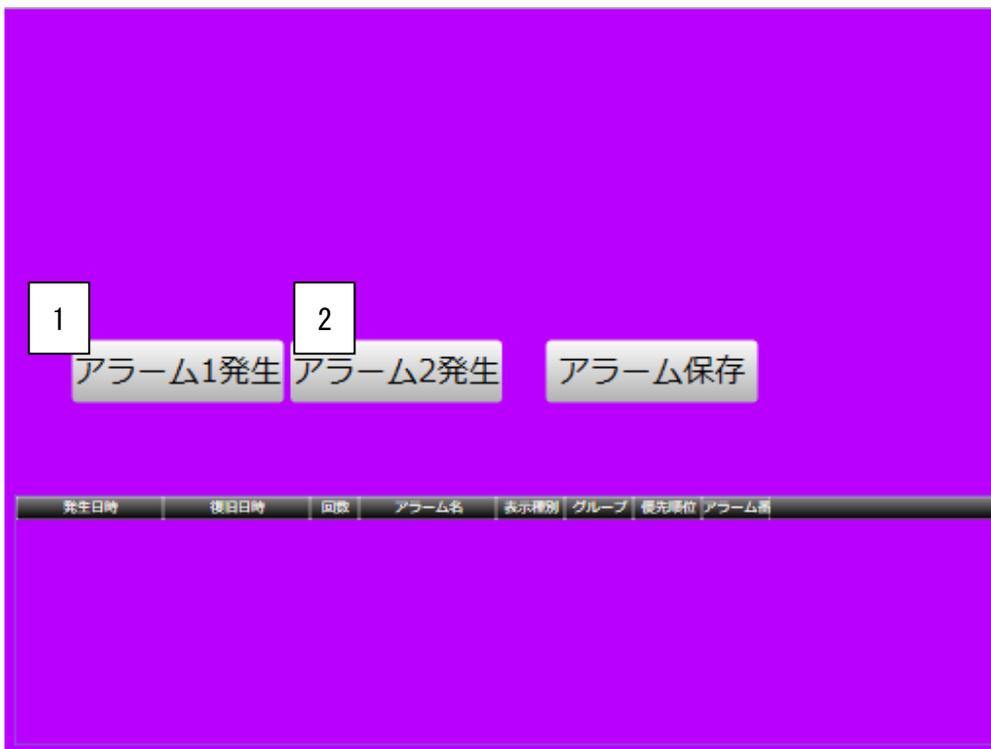


図 9-1-2-7 アラームラベル動作確認画面詳細

- アラーム1発生ボタンにポップアップを表示させる設定を追加します。
SwitchReference の ButtonSwitchReference に以下の2行を追加してください。

表 9-1-1-10 アラーム1発生ボタンプロパティ1

項目		設定値	備考
AI HMI Switch	Switch Reference	Address	100200
		Type	UINT
		Data	100

表 9-1-1-11 アラーム1発生ボタンプロパティ2

項目		設定値	備考
AI HMI Switch	Switch Reference	Address	100220
		Type	UINT
		Data	1

2. アラーム2発生ボタンにポップアップを表示させる設定を追加します。
SwitchReference の ButtonSwitchReference に以下の2行を追加してください。

表 9-1-1-12 アラーム2発生ボタンプロパティ1

項目		設定値	備考
AI HMI Switch	Switch Reference	Address	100210
		Type	UINT
		Data	101

表 9-1-1-13 アラーム2発生ボタンプロパティ2

項目		設定値	備考
AI HMI Switch	Switch Reference	Address	100230
		Type	UINT
		Data	1

- ⑦ AI_HMI_AlarmLabelList に以下の2行を追加します。

リスト 9-1-2-2 アラームラベル確認用 AlarmLabelList の追加

アラーム番号,	アラーム名
100,	アラーム1が発生しました。
101,	アラーム2が発生しました。

- ⑧ 作成した画面を実際に動作させてアラームラベルが動作することを確認します。
『第6章 作成したアプリの実行環境PCでの動作』を参照し、作成した画面アプリを動作させてください。



図 9-1-2-8 アラームラベル確認画面の実行

アラーム発生ボタンを押すことでポップアップ画面が表示され、AI_HMI_AlarmLabelList.csv に記述したアラーム内容の文字列が表示されることを確認してください。

9-2 操作ログ機能

作成した画面アプリ上の AI HMI Button コンポーネントを操作したログを保存することができます。

- ① MainWindow.xaml を開き、「ドキュメントアウトライン - [AI_HMI_Window]」を選択してください。
プロパティ欄に AI HMI Window のプロパティが表示されます。

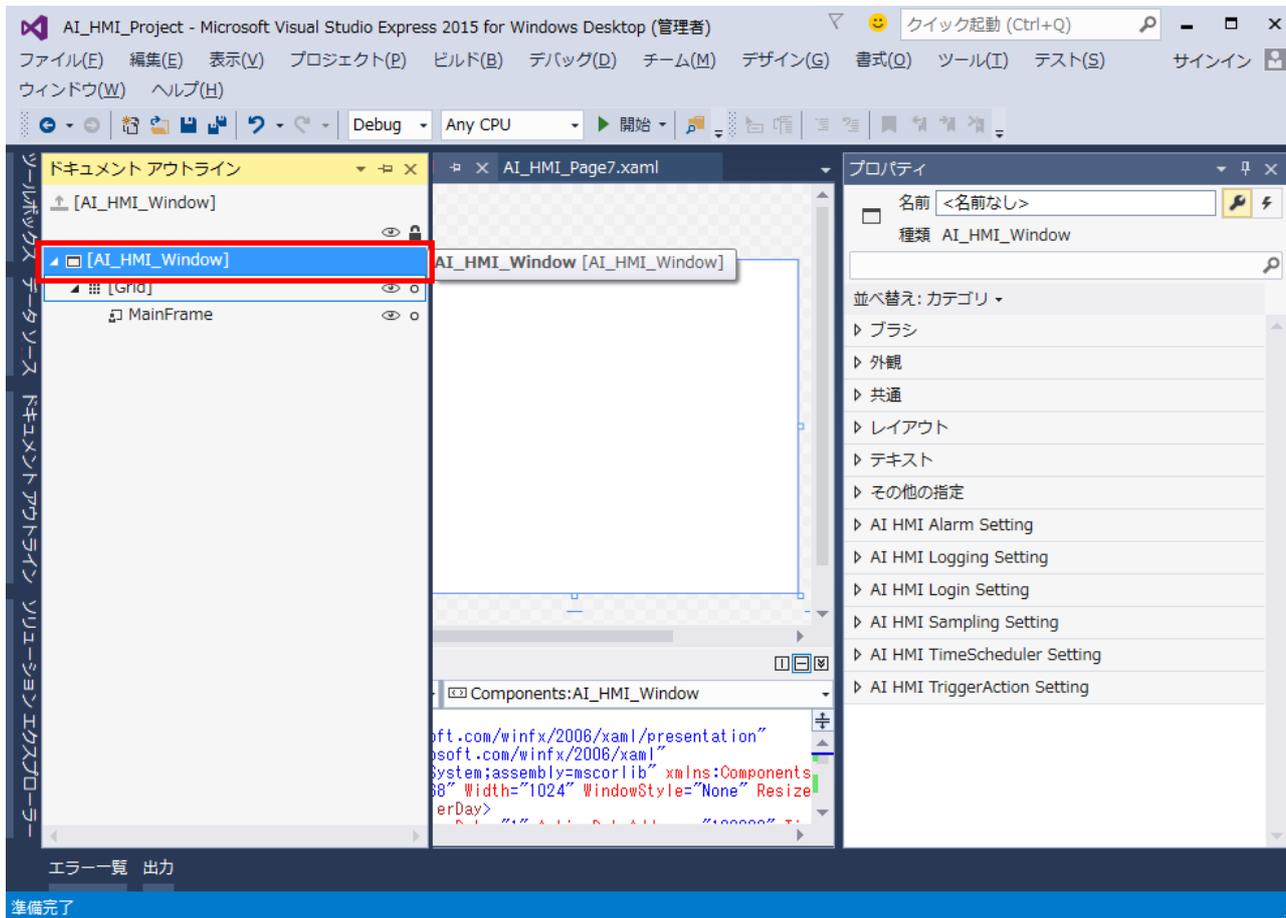


図 9-2-1 AI HMI Window プロパティ

- ② プロパティから「AI HMI Logging Setting」を選択してください。
AI HMI Logging Setting プロパティが開きます。

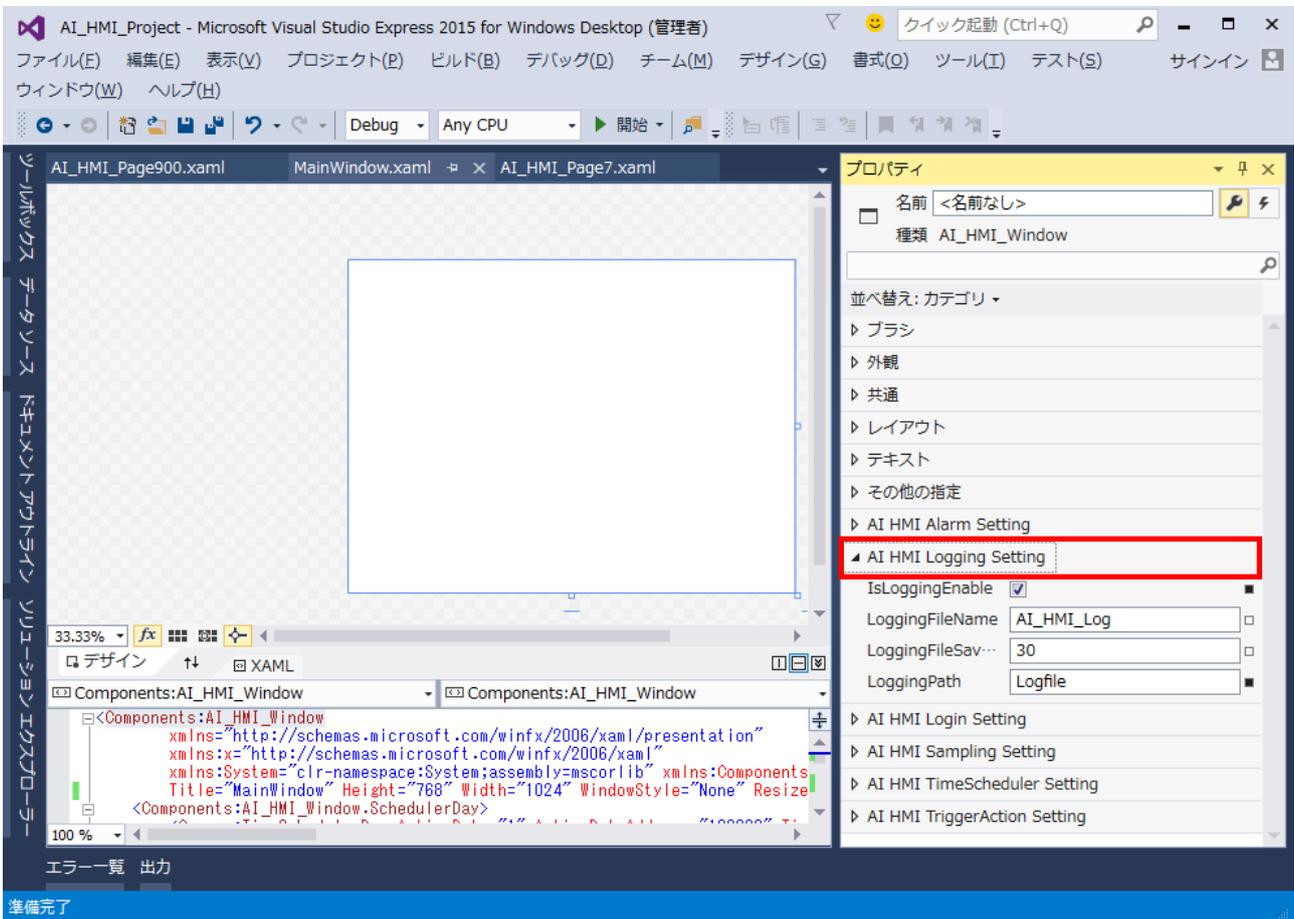


図 9-2-2 AI HMI Logging Setting プロパティ

AI HMI Logging Setting プロパティのそれぞれの意味は以下の通りです。

表 9-2-1 AI HMI Logging Setting

項目	データ型	初期値	用途
IsLoggingEnable	チェックボックス	チェックなし	操作ログ保存機能の有効無効を切り替えます。チェックを入れると操作ログを保存ようになります。
LoggingFileName	String 型	AI_HMI_Log	操作ログを保存するファイル名を設定します。実際に保存するファイルは[LoggingFileName で設定した名前)+年月日]という名前になります。
LoggingFileSaveDays	DINT 型	30	ログファイルを保存する日数を設定します。設定した日数を越えたファイルは自動的に削除されます。必要に応じてファイルのバックアップを取るなどしてください。
LoggingPath	String 型	初期値なし	操作ログを保存するファイルの保存先フォルダを設定します。設定は相対パス、絶対パス共に可能です。相対パスの場合、画面アプリの exe ファイルからみた相対パスとなります。

- ③ 実際に操作ログを保存できるように設定します。
AI HMI Logging Setting プロパティ表 9-2-2 のように設定してください。

表 9-2-2 AI HMI Logging Setting

項目	設定内容	備考
IsLoggingEnable	チェックあり	操作ログ保存機能を有効にします。
LoggingFileName	AI_HMI_Log	操作ログを AI_HMI_Log_(日付) というファイル名で保存します。
LoggingFileSaveDays	30	30 日より以前に保存したログファイルを自動で削除するようにします。
LoggingPath	LogFile	画面アプリの exe ファイルと同じ階層に LogFile というフォルダを作成し、操作ログを保存します。

- ④ 実際に画面アプリを操作し、ログが保存されることを確認してください。
操作ログが保存されるのは AI HMI Button コンポーネントの操作のみです。
操作ログファイルは以下のような形式で保存されます。(エンコード形式 S-JIS)

リスト 9-2-1 操作ログファイルの内容

2016/01/26 14:25:49	Object=aI_HMI_Button Action=OnClick
2016/01/26 14:26:16	Object=aI_HMI_Button Action=OnClick

① ②

表 9-2-3 操作ログファイルの内容詳細

No.	内容	備考
①	操作日時	操作のあった年月日時分秒を記録します。
②	操作のあったボタン	操作のあったボタンの名前を記録します。

※ 操作ログに保存されるボタン名は開発環境上でのオブジェクト名になります。
各ボタンのオブジェクト名はそれぞれのプロパティの「名前」項目から個別に設定することができます。

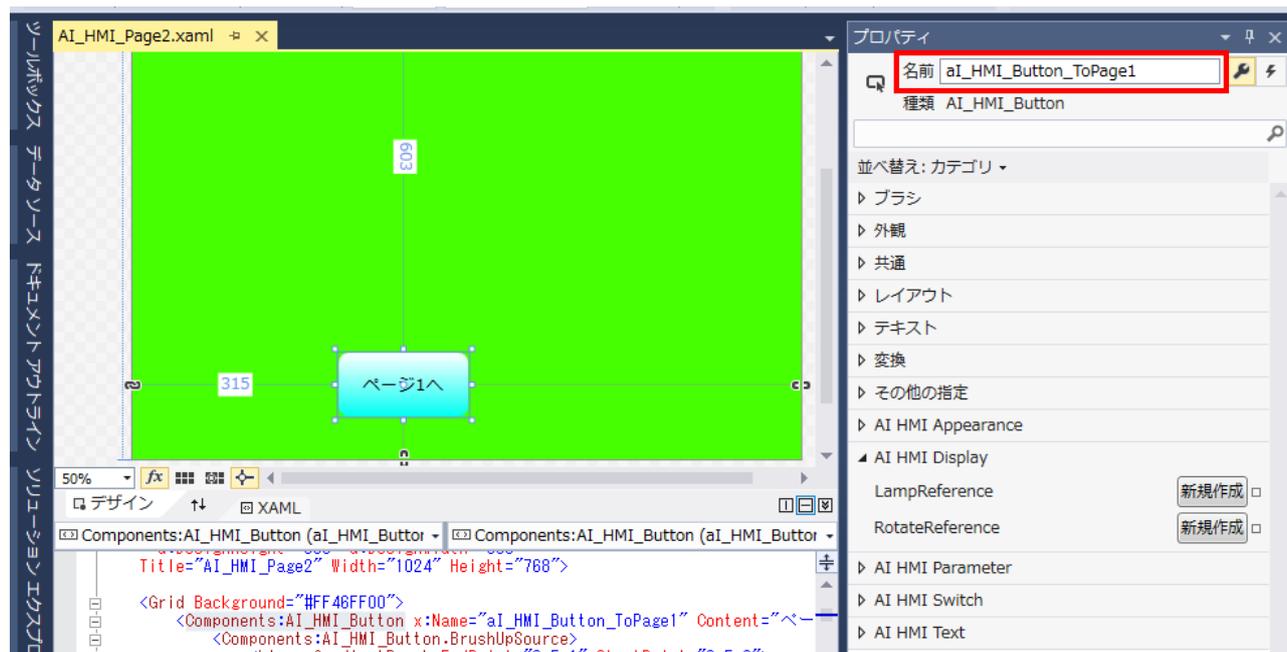


図 9-2-3 オブジェクト名の設定

9-3 ユーザログイン機能

特定のユーザとしてログインしている状態でのみボタン操作をすることができるようになります。

- ① MainWindow.xaml を開き、「ドキュメントアウトライン - [AI_HMI_Window]」を選択してください。
プロパティ欄に AI HMI Window のプロパティが表示されます。

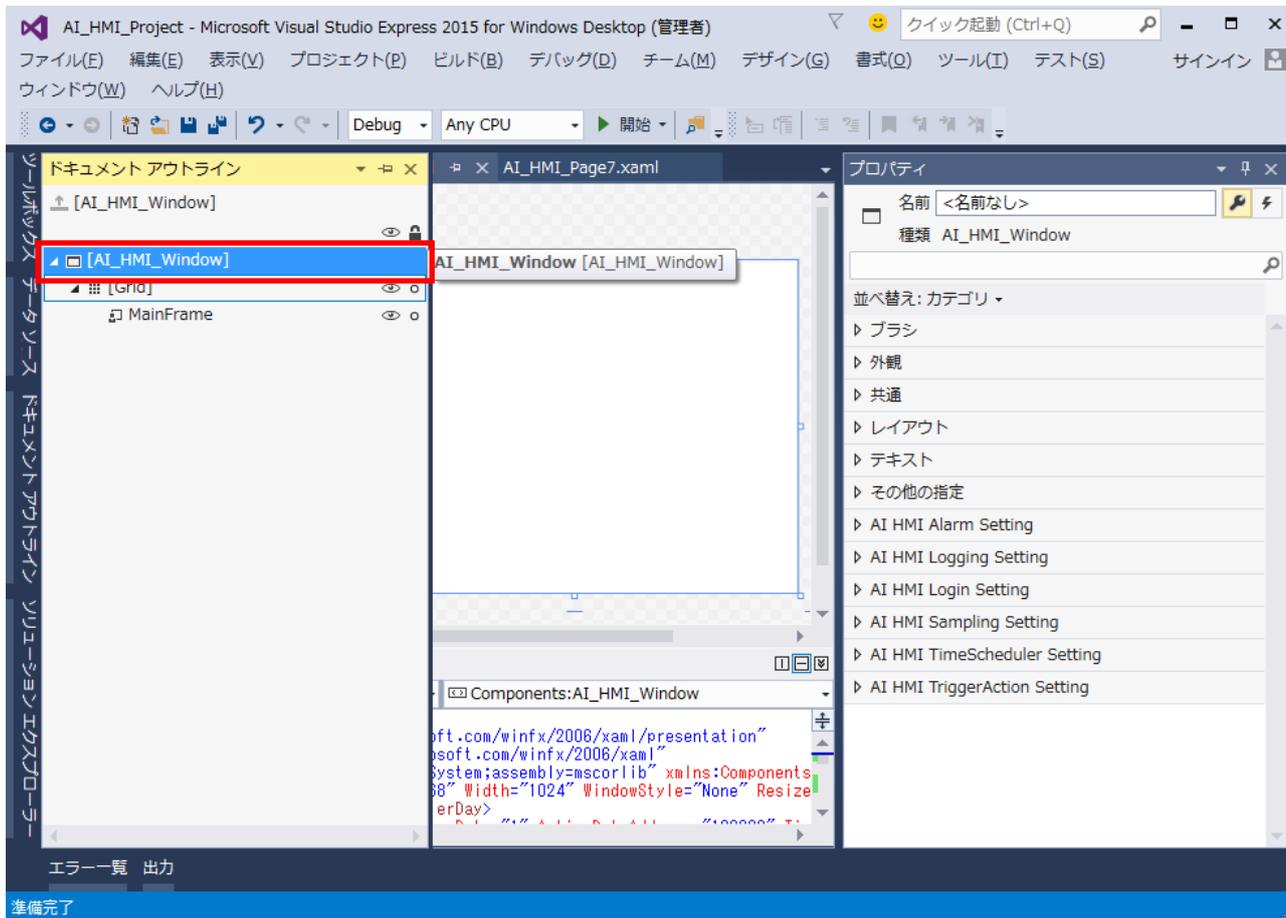


図 9-3-1 AI HMI Window プロパティ

- ② プロパティから「AI HMI Login Setting」を選択してください。
AI HMI Login Setting プロパティが開きます。

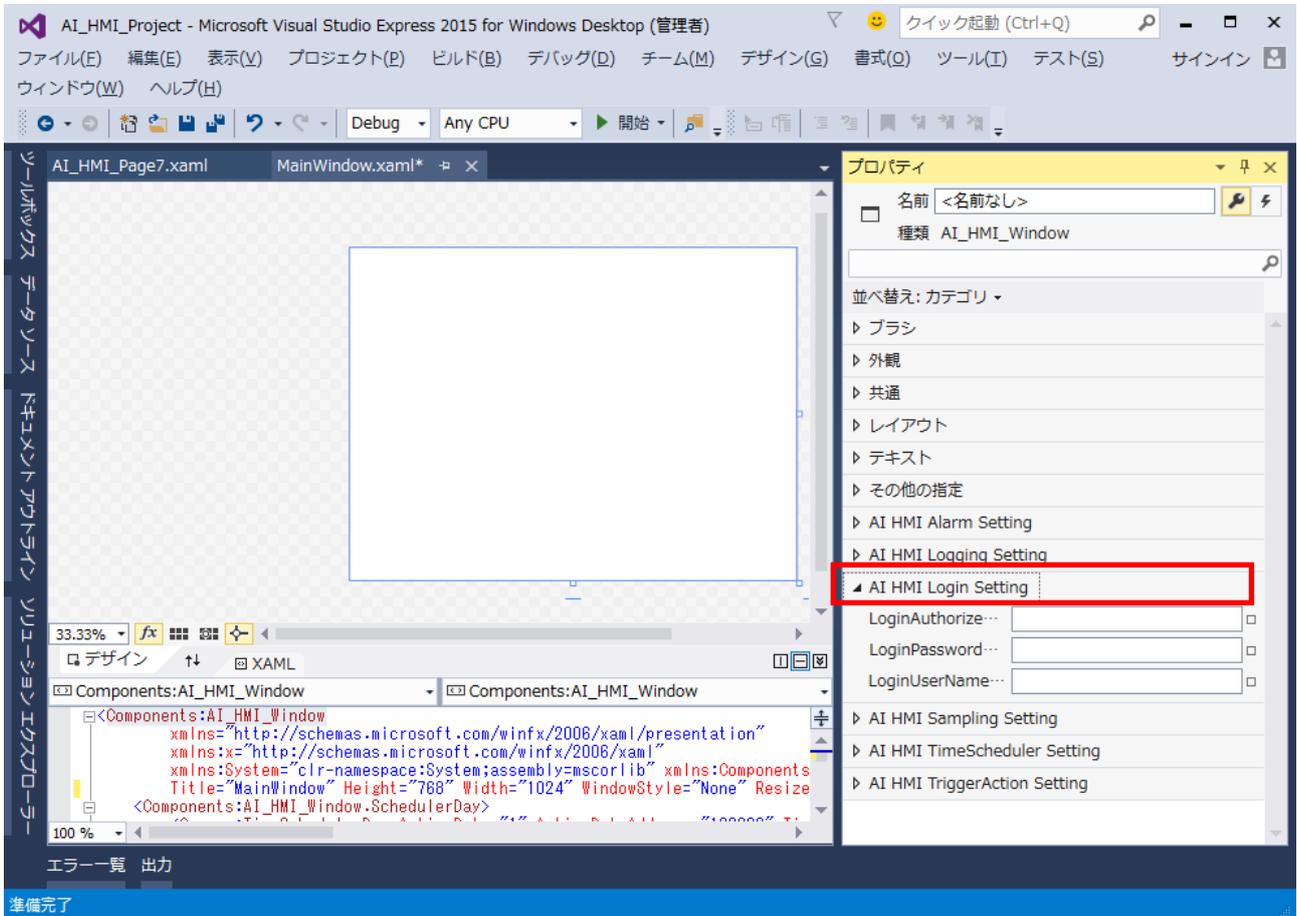


図 9-3-2 AI HMI Login Setting プロパティ

AI HMI Login Setting プロパティのそれぞれの意味は以下の通りです。

表 9-3-1 AI HMI Logging Setting

項目	データ型	初期値	用途
LoginAuthorizeAddress	String 型	初期値 なし	現在ログインしているユーザのログイン権限情報を格納する変数名または共有メモリアドレスを設定します。 格納されるデータは INT 型です。 ログインに成功するとユーザ権限番号が格納されます。 ログオフすると 0 が格納されます。 ログインに失敗すると -1 が格納されます。
LoginPasswordAddress	String 型	初期値 なし	ログインパスワードを入力する変数名または共有メモリアドレスを設定します。 ログインパスワードは String 型で設定されている必要があります。
LoginUserNameAddress	String 型	初期値 なし	ログインユーザ名を入力する変数名または共有メモリアドレスを設定します。 ログインユーザ名は String 型で設定されている必要があります。

- ③ ユーザログインに使用するユーザ名とパスワード、ログイン権限番号は AI_HMI_UserList.csv に記述します。
AI_HMI_UserList.csv は画面アプリの exe ファイルと同じフォルダ内の「User」フォルダに格納します。
AI_HMI_UserList.csv の内容の詳細は以下のようになります。(エンコード形式 S-JIS)

リスト 9-3-1 AI_HMI_UserList.csv の内容

ユーザ名,	パスワード,	権限
Maker,	Make,	10
Administrator,	Admin,	5
Maintenance,	Mainte,	3
Operator,	Ope,	1

※ パスワードには「,」や「,」などの特殊文字は使用しないでください。

表 9-3-2 AI_HMI_UserList.csv の内容詳細

項目	内容
ユーザ名	ログインユーザ名です。 LoginUserNameAddress の指定先の値との比較に使用します。
パスワード	ログインパスワードです。 LoginPasswordAddress の指定先の値との比較に使用します。
権限	ログイン権限番号です。 LoginAuthorizeAddress に格納される値です。

- ④ ユーザログイン機能ではログインする際にシステム変数を使用します。
使用するシステム変数は表 9-3-3 の通りです。

表 9-3-3 ユーザログイン機能で使用するシステム変数

システム変数名	内容
USER_LOGIN_ACTION	1 を入力するとログインユーザ名とログインパスワードの比較を行います。 比較に成功した場合、LoginAuthorizeAddress にログイン権限番号を格納します。 比較が完了すると USER_LOGIN_ACTION は自動的に 0 に戻ります。
USER_LOGOFF_ACTION	1 を入力するとログオフをします。 ログオフすると LoginAuthorizeAddress に 0 を格納します。 ログオフが完了すると USER_LOGOFF_ACTION は自動的に 0 に戻ります。

- ⑤ 実際にユーザログインができるように設定します。
AI HMI Logging Setting プロパティ表 9-3-4 のように設定してください。

表 9-3-4 AI HMI Login Setting

項目	設定内容	備考
LoginAuthorizeAddress	100300	共有メモリのアドレス 100300 に INT 型でログイン権限情報を格納するように設定します。 ログインに成功すると 100300 にユーザ権限番号が格納されるようになります。
LoginPasswordAddress	140300	ログインパスワードを入力する共有メモリのアドレスを 140300 に設定します。
LoginUserNameAddress	140400	ログインユーザ名を入力する共有メモリのアドレスを 140400 に設定します。

表 9-3-4 のように設定した場合、ユーザログインをする手順は以下のようになります。

1. アプリケーション上でユーザ名を 140400 に、パスワードを 140300 に入力する。
2. アプリケーション上でシステム変数「USER_LOGIN_ACTION」に 1 を入力する。
3. 100300 にログイン権限情報が格納される。

- ⑥ ユーザログイン機能を使用した画面を作成します。
図 9-3-3 のように部品を配置してください。

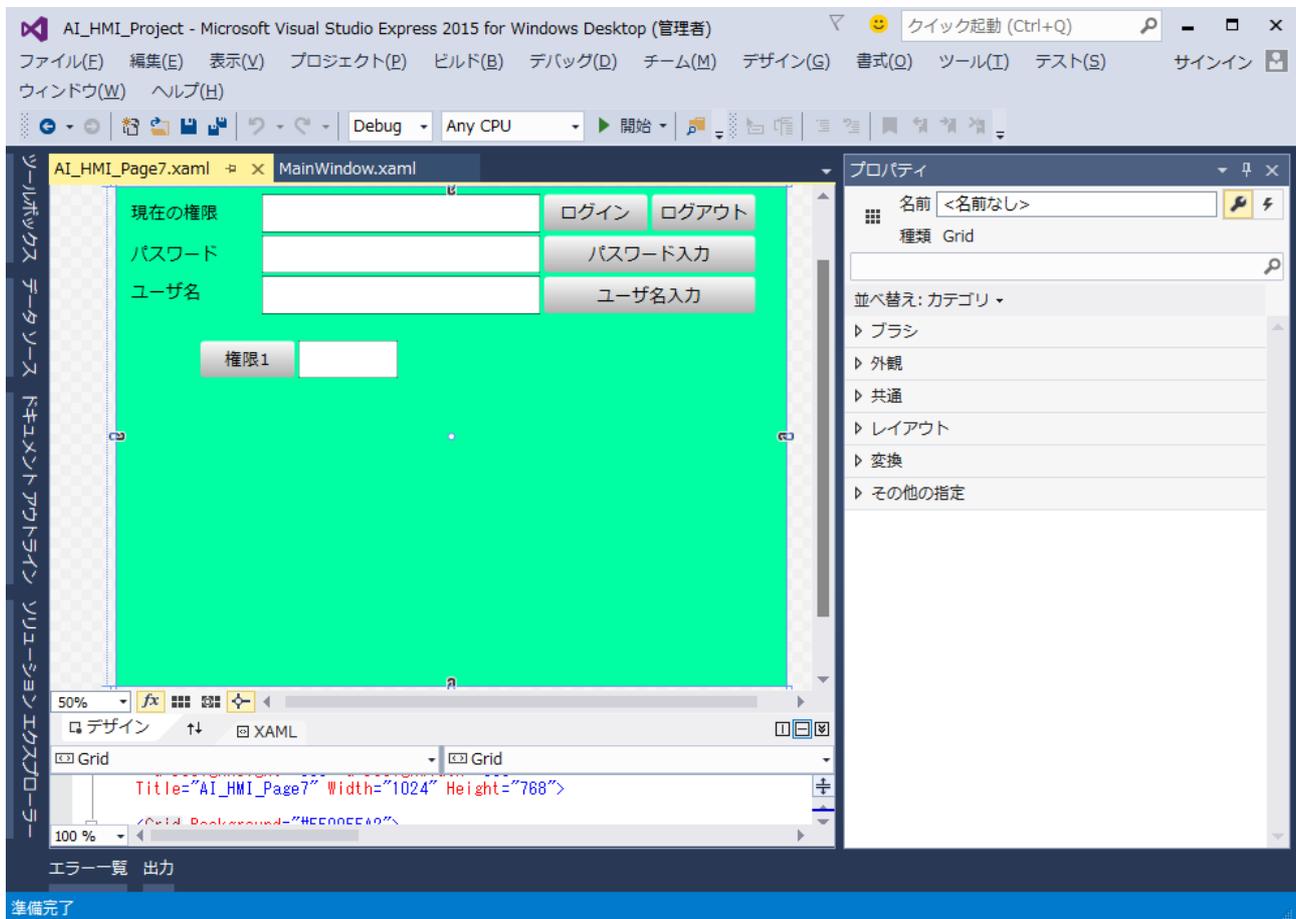


図 9-3-3 ユーザログイン画面の作成

それぞれの部品のプロパティは以下のように設定してください。

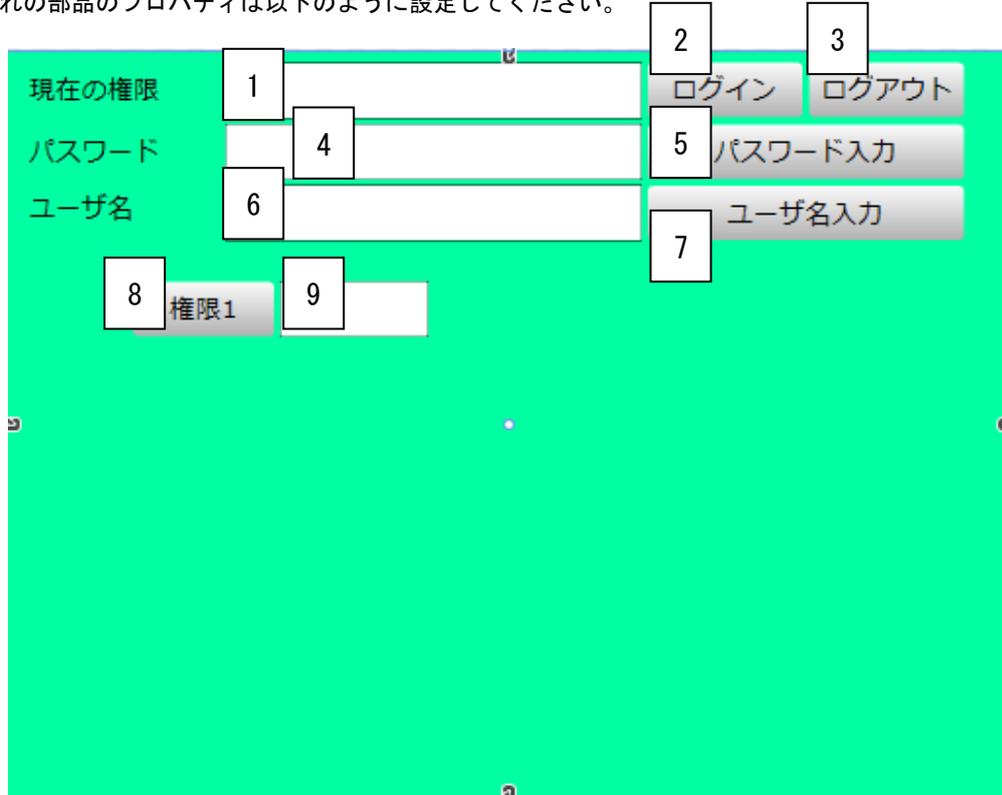


図 9-3-4 ユーザログイン画面詳細

1. ユーザ権限を表示する欄です。AI HMI Label を使用します。

表 9-3-5 ユーザ権限表示欄プロパティ

項目		設定値	備考
AI HMI Display	Address	100300	
	LabelType	Numerical	
	Numerical	DataType	INT

2. ログインを実行するボタンです。AI HMI Button を使用します。

表 9-3-6 ログインボタンプロパティ

項目		設定値	備考
AI HMI Switch	SystemSwitch Reference	Address	USER_LOGIN_ACTION
		Data	1

SystemSwitchReference の右側のボタンを押すと表示される ButtonSystemSwitchReference ウィンドウ内で設定します。

3. ログオフを実行するボタンです。AI HMI Button を使用します。

表 9-3-7 ログオフボタンプロパティ

項目		設定値	備考
AI HMI Switch	SystemSwitch Reference	Address	USER_LOGOFF_ACTION
		Data	1
SystemSwitchReference の右側のボタンを押すと表示される ButtonSystemSwitchReference ウィンドウ内で設定します。			

4. 入力したパスワードを表示する欄です。AI HMI Label を使用します。

表 9-3-8 パスワード表示欄プロパティ

項目		設定値	備考
AI HMI Display	Address	140300	LoginPasswordAddress と同じアドレスを設定します。
	LabelType	String	
AI HMI Password	AlternatePasswordChar	チェックあり	パスワード表示欄に表示する文字列を「●」で置き換えるよう設定します。
	PasswordChar	●	

5. 入力したパスワード入力するためのソフトキーを表示するボタンです。AI HMI Button を使用します。

表 9-3-9 パスワード入力ソフトキー呼び出しボタンプロパティ

項目		設定値	備考	
AI HMI Switch	Switch Reference	Address	100002	
		Type	INT	
		Data	10	
	Reference[1]	Address	141000	呼び出したソフトキーが LoginPasswordAddress に書き込みをするよう設定します。
		Type	STRING	
		Data	140300	
ソフトキーを配置したダイアログを呼び出す変数を入力します。				

6. 入力したユーザ名を表示する欄です。AI HMI Label を使用します。

表 9-3-10 パスワード表示欄プロパティ

項目		設定値	備考
AI HMI Display	Address	140400	LoginUserNameAddress と同じアドレスを設定します。
	LabelType	String	

7. 入力したパスワード入力するためのソフトキーを表示するボタンです。AI HMI Button を使用します。

表 9-3-11 パスワード入力ソフトキー呼び出しボタンプロパティ

項目		設定値	備考	
AI HMI Switch	Switch Reference	Address	100002	
		Type	INT	
		Data	10	
	Reference[1]	Address	141000	呼び出したソフトキーが LoginUserNameAddress に書き込みをする よう設定します。
		Type	STRING	
		Data	140400	

8. ユーザログインの確認のため、ユーザ権限1でログインしているときに動作するボタンです。
AI HMI Button を使用します。

表 9-3-12 ログイン確認用ボタンプロパティ

項目		設定値	備考
AI HMI Switch	AuthorizeLevel	Level1	ユーザ権限1以上でログインしているときに動作できるようにします。
	SwitchReference	Address	100310
		BitAction	Reversal

9. ログイン確認用ボタンの挙動を確認するための表示欄です。AI HMI Label を使用します。

表 9-3-13 ログイン確認用表示欄プロパティ

項目		設定値	備考
AI HMI Display	Address	100310	ログイン確認用ボタンを押したときに変化する値を表示します。
	LabelType	INT	

また、ログインパスワードとログインユーザ名を入力するためのソフトキーを作成します。
AI HMI Page10 を新たに作成し、図 9-3-5 のように AI HMI Softkey を配置してください。

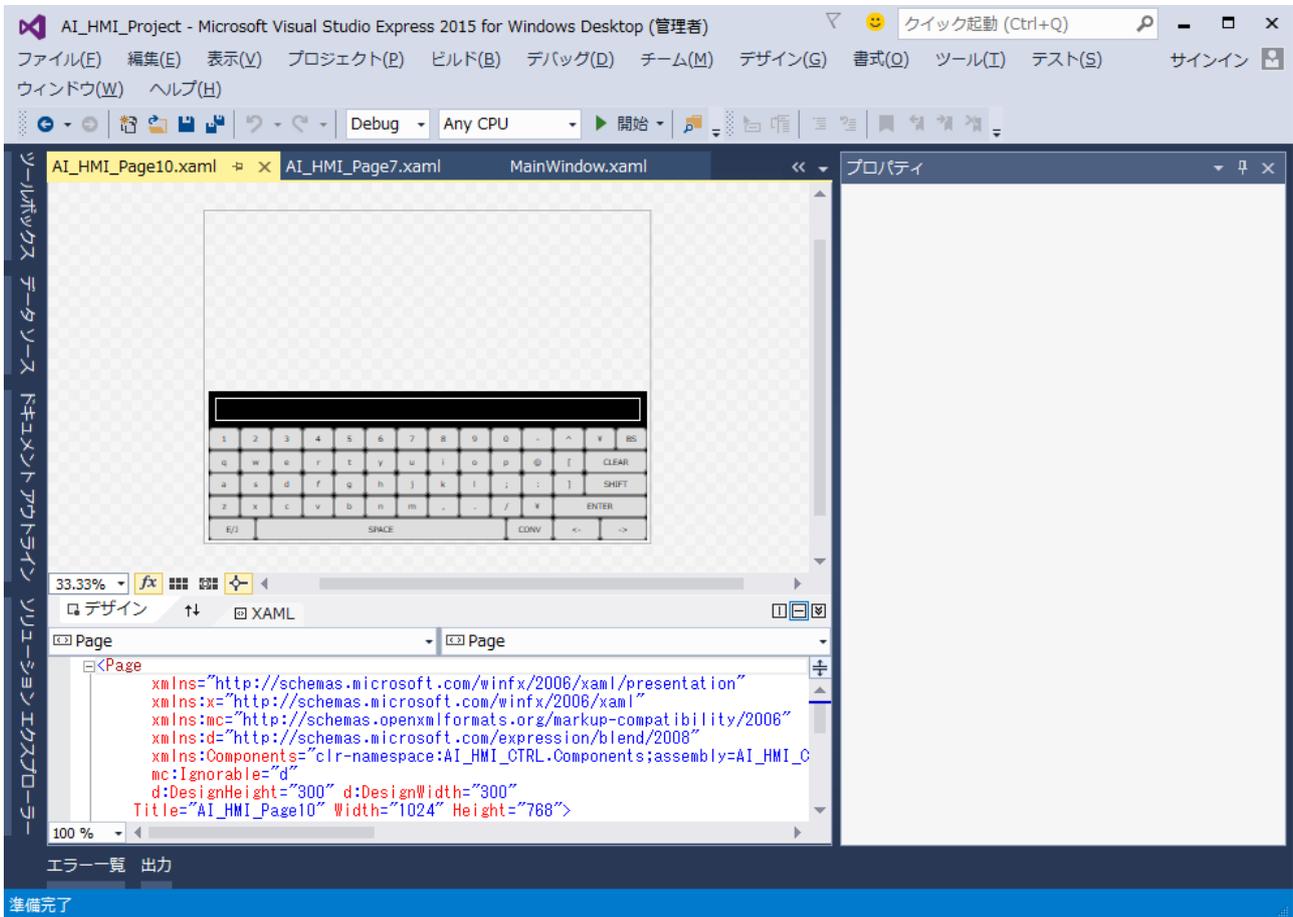


図 9-3-5 ログイン情報入力ソフトキーダイアログ作成

配置した AI HMI Softkey のプロパティは以下のように設定してください。

表 9-3-14 ログイン情報入力ソフトキープロパティ

項目		設定値	備考	
AI HMI Switch	EnterAddress	141000	ソフトキーの入力が完了したときにダイアログを閉じるようにします。	
	SwitchReference	Address		100002
		Data		0

作成したダイアログをフレーム上で表示できるように設定します。
MainWindow を開き、AI_HMI_Frame – DialogFrameReference で以下の設定を追加してください。

表 9-3-15 ダイアログ表示のプロパティ

項目	設定値	備考
Address	100002	ソフトキーのダイアログを表示します。
Data	10	
PageNumber	10	

- ⑦ 作成した画面を実際に動作させてユーザログインが成功することを確認します。
『第6章 作成したアプリの実行環境PCでの動作』を参照し、作成した画面アプリを動作させてください。

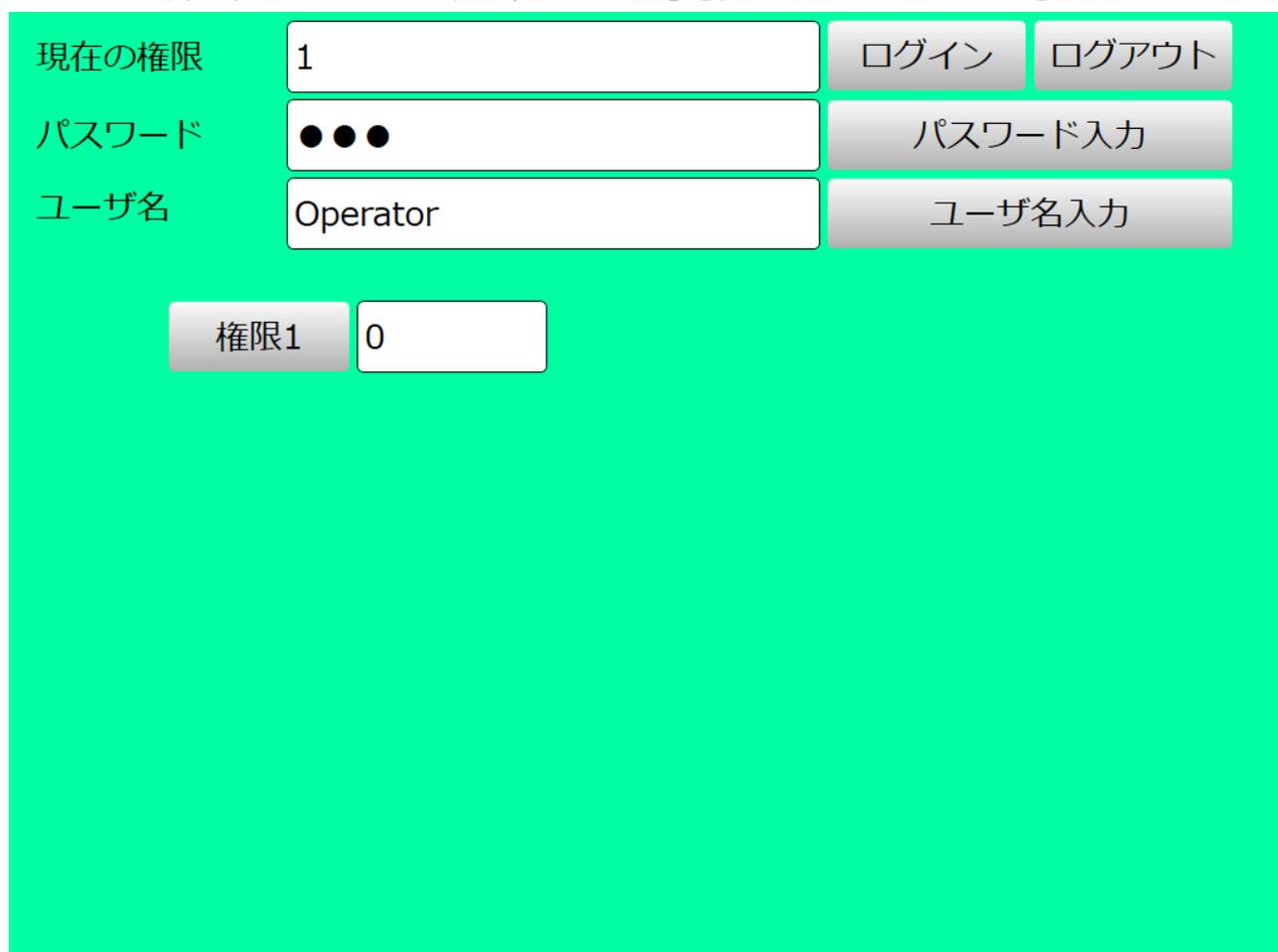


図 9-3-6 ログイン画面の実行

AI_HMI_UserList.csv で設定したパスワードとユーザ名を入力した状態でログインボタンを押したとき、現在の権限表示欄にログイン権限番号が表示されることを確認してください。
また、ログイン状態のとき権限 1 ボタンを押すことで確認用表示欄の値が 0⇔1 と表示が切り替わることを確認してください。

9-4 サンプリング機能

特定の OPC 変数または共有メモリアドレスの値をサンプリングし、記録する機能です。サンプリングできるデータ件数は、30 グループ×256 点 (32bit データ) になります。各グループ毎に別々の周期でサンプリングすることが可能です。

- ① MainWindow.xaml を開き、「ドキュメントアウトライン - [AI_HMI_Window]」を選択してください。プロパティ欄に AI HMI Window のプロパティが表示されます。

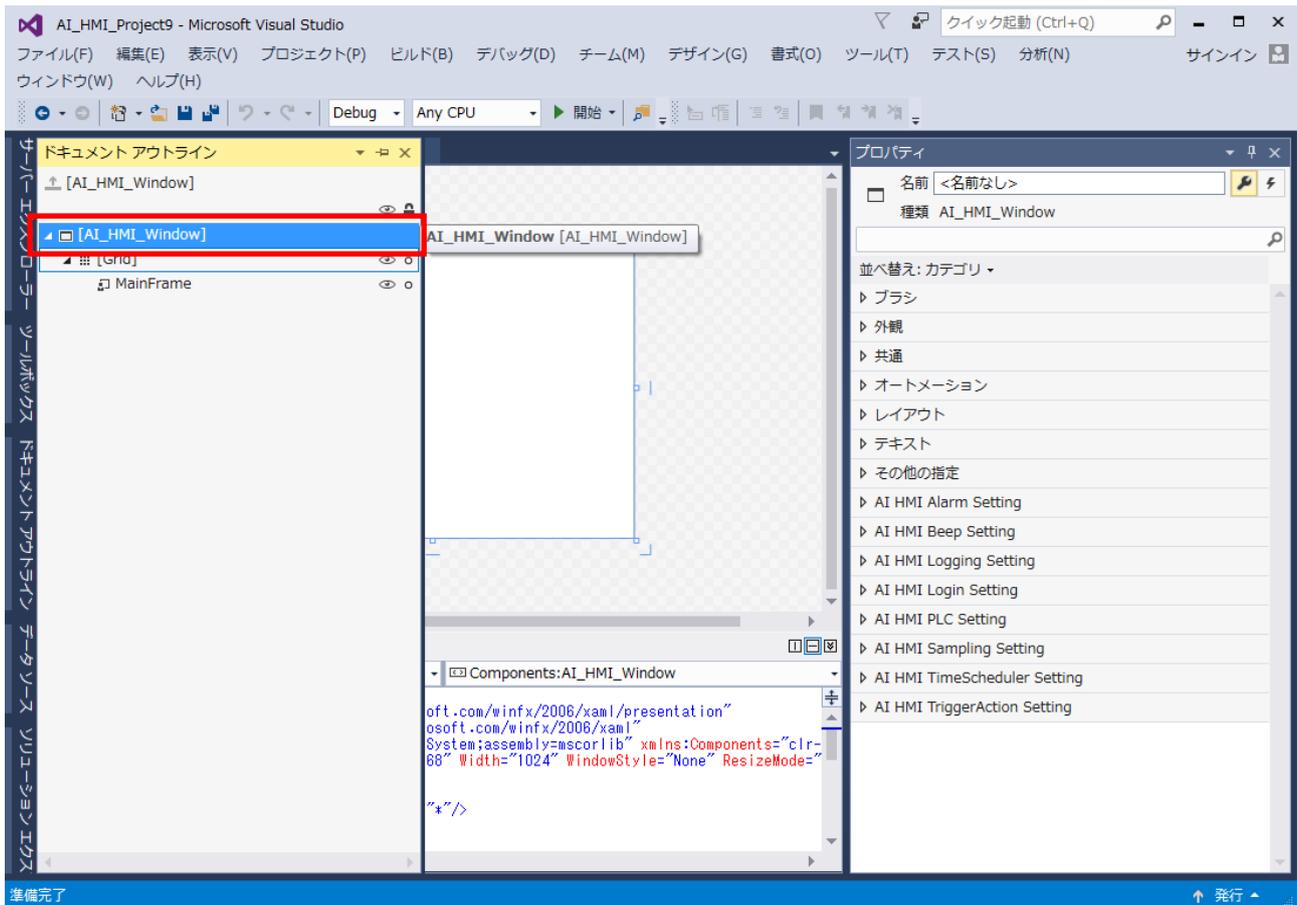


図 9-4-1 AI HMI Window プロパティ

- ② プロパティから「AI HMI Sampling Setting」を選択してください。
AI HMI Sampling Setting プロパティが開きます。

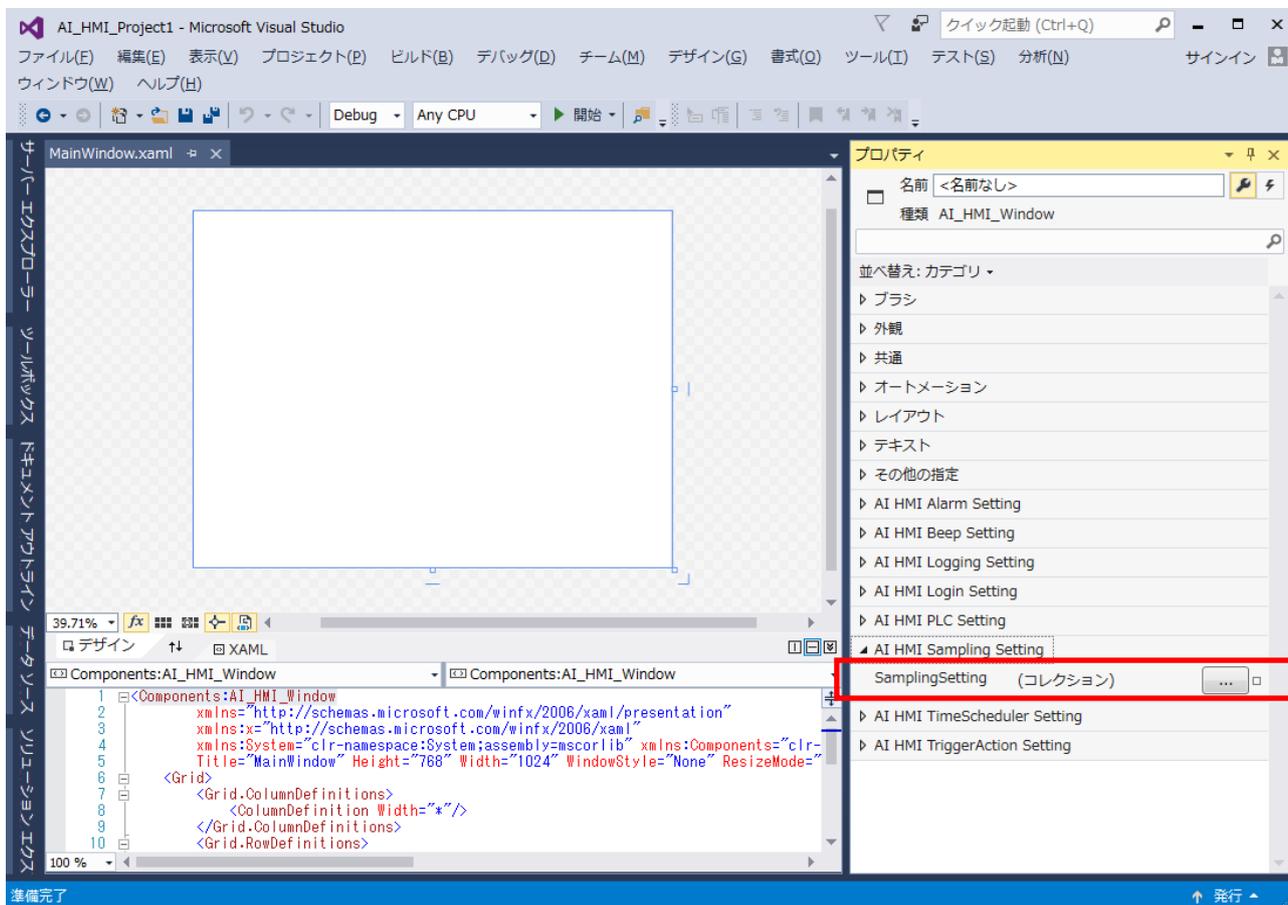


図 9-4-2 AI HMI Sampling Setting プロパティ

AI HMI Sampling Setting プロパティのそれぞれの意味は以下の通りです。

表 9-4-1 AI HMI Sampling Setting

項目	内容
SamplingSetting	サンプリング機能の挙動を設定します。

SamplingSetting の右のボタンを押すことで SamplingReference の設定ができます。
SamplingReference の詳細は以下の通りです。

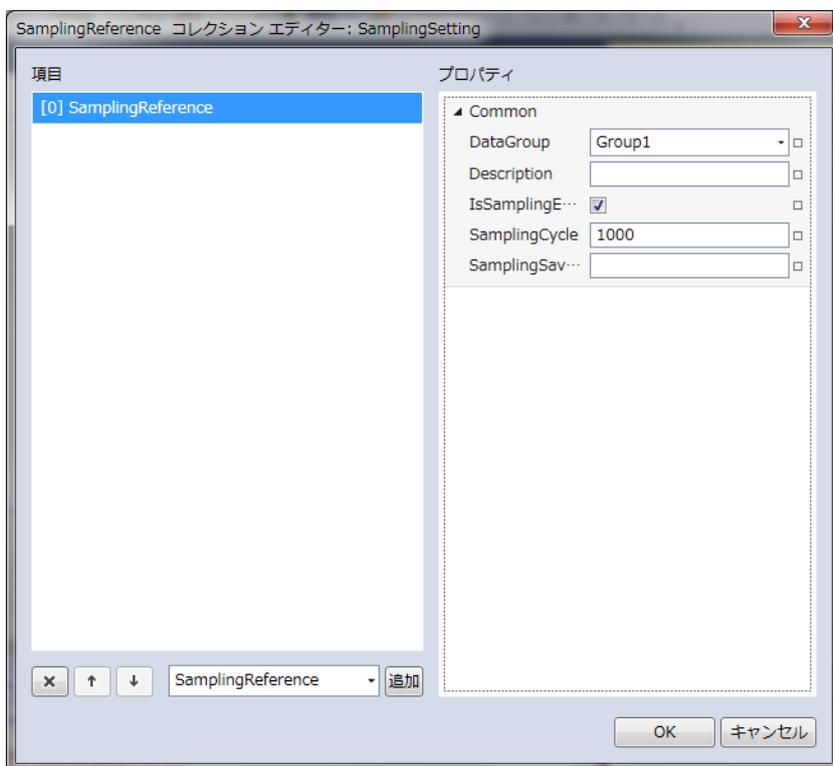


図 9-4-3 SamplingReference のプロパティ

表 9-4-2 SamplingReference のプロパティ詳細

項目	データ型	初期値	内容
DataGroup	プルダウンメニュー	Group1	サンプリングするデータのグループ番号を設定します。
Description	String 型	初期値なし	コメント欄として利用できます。オブジェクトの動作に影響は与えません。
IsSamplingEnable	チェックボックス	チェックあり	サンプリング機能の有効と無効を入力します。チェックが入っていると選択している SamplingReference のサンプリングを行います。
SamplingCycle	UDINT 型	1000	サンプリングの周期をミリ秒単位で設定します。設定が有効な値は 100 以上です。100 以下の値を設定した場合、強制的に 100msec で動作します。
SamplingSaveCsvPath	String 型	初期値なし	サンプリングデータの CSV ファイルの保存先フォルダを設定します。設定は相対パス、絶対パス共に可能です。相対パスの場合、画面アプリの exe ファイルからみた相対パスとなります。

- ③ サンプリング機能に使用する各グループ毎のサンプリング項目は AI_HMI_SamplingSetting_XX.csv に記述します。(XX=グループ番号 01~30)

AI_HMI_SamplingSetting_XX.csv は画面アプリの exe ファイルと同じフォルダ内の「Sampling」フォルダに格納します。

AI_HMI_SamplingSetting_XX.csv の内容の詳細は以下のようになります。(エンコード形式 S-JIS)

リスト 9-4-1 AI_HMI_SamplingSetting_XX.csv の内容

系列名,	変数名,	小数部桁数
Data1,	135000,	2
Data2,	135008,	3
Data3,	135016,	2
Data4,	135024,	2

表 9-4-3 AI_HMI_SamplingSetting_XX.csv の内容詳細

項目	内容
系列名	各サンプリング項目の系列名です。 「系列名」は文字列として扱います。 「系列名」で設定した文字列はサンプリング結果保存ファイルの系列名に使用されます。
変数名	各サンプリング項目でサンプリングするアドレスです。 「変数名」は変数または共有メモリアドレスを設定します。 「変数名」で設定した変数をサンプリング周期毎に SRAM のサンプリング領域に保存します。
小数部桁数	各サンプリング項目のサンプリングした値の小数点以下桁数です。 「小数部桁数」は整数値を設定します。 「小数部桁数」で設定した整数はサンプリングデータの小数点の位置をとして扱います。

- ④ サンプリング機能ではサンプリングの保存、削除をする際にユーザ用システム変数を使用します。使用するユーザ用システム変数は表 9-4-4 の通りです。

表 9-4-4 サンプリング機能で使用するユーザ用システム変数

システム変数	アドレス	内容
SAMPLING_CSV_SAVE	68514+Offset (※1)	サンプリングデータの保存をする際に使用します。 1 を入力すると SRAM のサンプリングデータ領域の内容を CSV ファイルに保存します。 保存が完了すると自動的に 0 が入力されます。
SAMPLING_CSV_ERROR	68516+Offset (※1)	サンプリングデータの保存結果を自動的に入力します。 保存成功時は 1、保存失敗時は -1 が入力されます。 SAMPLING_CSV_ERROR は次回サンプリングデータ保存時まで値は変化しません。
SAMPLING_ALL_CLEAR	68518+Offset (※1)	サンプリングデータのクリアをする際に使用します。 1 を入力すると SRAM のサンプリングデータ領域の内容をクリアします。 クリアが完了すると自動的に 0 が入力されます。

※1 Offset は「(グループ番号 - 1) × 10」になります。

- ⑤ 実際にサンプリングを実行するように設定します。
SamplingSetting を表 9-4-5 のように設定してください。

表 9-4-5 [0]SamplingReference の設定

項目	設定内容	備考
DataGroup	Group1	Group1 のサンプリングを、1000msec 周期で行います。 CSV が出力されるパスは相対パスの「Sampling」フォルダになります。
Description		
IsSamplingEnable	チェックあり	
SamplingCycle	1000	
SamplingSaveCsvPath	SamplingFile	

AI_HMI_SamplingSetting_01.csv をリスト 9-4-2 の内容で用意し、画面アプリの exe ファイルと同じフォルダ内の「Sampling」フォルダに格納します。

リスト 9-4-2 AI_HMI_SamplingSetting_01.csv の内容

系列名,	変数名,	小数部桁数
Data1,	135000,	2
Data2,	135008,	3
Data3,	135016,	2
Data4,	135024,	2

- ⑥ サンプル機能動作を確認します。

図 9-4-4 のように AI HMI TrendGraph と AI HMI Slider、AI HMI Button を配置してください。

AI HMI TrendGraph についての詳細は AI HMI リファレンスマニュアルの『第 31 章 AI HMI TrendGraph』を参照ください。

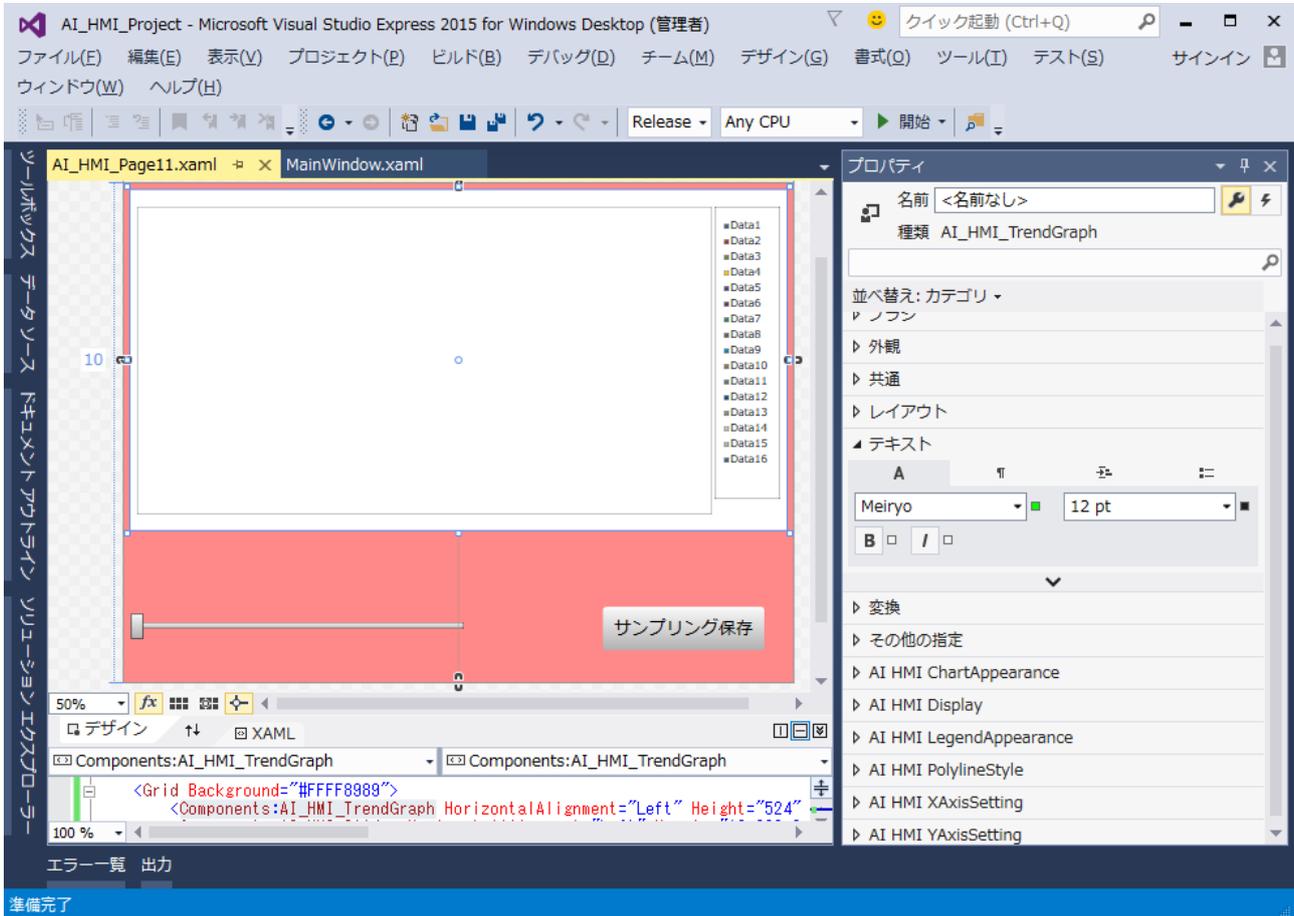


図 9-4-4 サンプル機能動作確認画面

それぞれの部品のプロパティを以下のように変更してください。

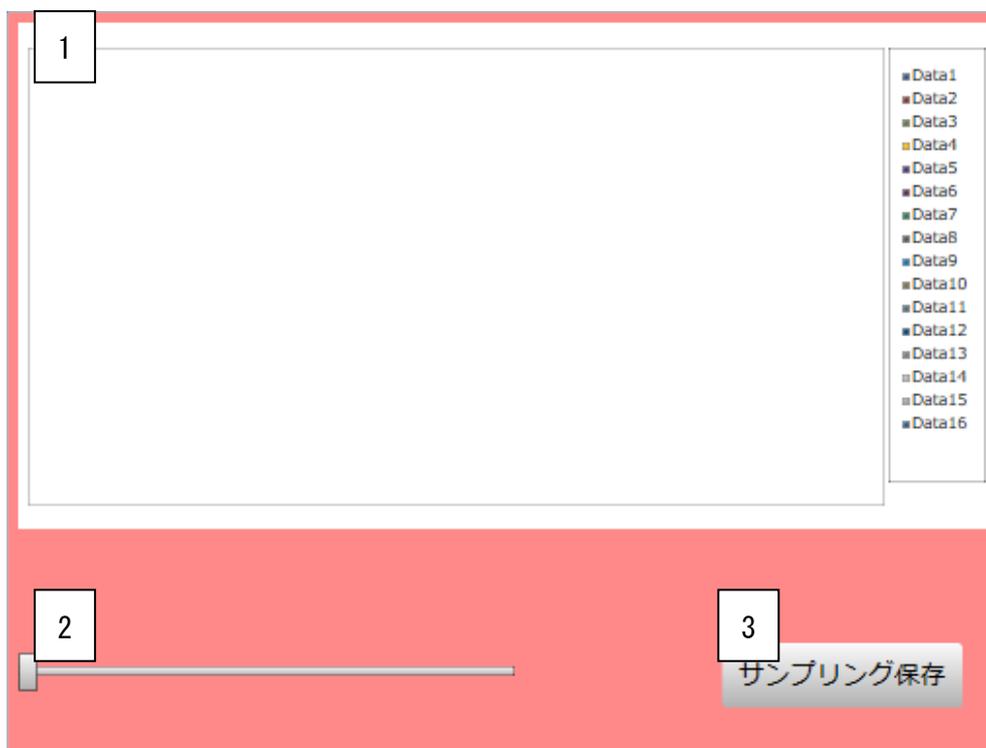


図 9-4-5 サンプリング動作確認画面詳細

1. サンプリングデータを表示するトレンドグラフです。AI HMI TrendGraph を使用します。

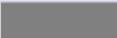
表 9-4-6 トレンジグラフ AI HMI YAxisSetting プロパティ

項目	設定値	備考	
AI HMI YAxisSetting	YAxisInterval	20	トレンドグラフの Y 軸のインターバルを 20 に設定します。
	YAxisMaxValue	100	トレンドグラフの Y 軸の最大値を 100 に設定します。

表 9-4-7 トレンジグラフ AI HMI SeriesData プロパティ (1/2)

項目	設定値	備考	
AI HMI SeriesData	DataGroup	Group1	サンプリングするデータのグループを Group1 に設定します。

表 9-4-8 トレンドグラフ AI HMI SeriesData プロパティ (2/2)

項目		設定値	備考
AI HMI SeriesData	[0]SeriesDataReference	DataColor	 データの色を赤色に設定します。
		DataNumber	1 参照するデータ番号を1に設定します。
		Visible	Visible データを表示するように設定します。
AI HMI SeriesData	[1]SeriesDataReference	DataColor	 データの色を灰色に設定します。
		DataNumber	2 参照するデータ番号を2に設定します。
		Visible	Visible データを表示するように設定します。
AI HMI SeriesData	[2]SeriesDataReference	DataColor	 データの色を緑色に設定します。
		DataNumber	3 参照するデータ番号を3に設定します。
		Visible	Visible データを表示するように設定します。
AI HMI SeriesData	[3]SeriesDataReference	DataColor	 データの色を青色に設定します。
		DataNumber	4 参照するデータ番号を4に設定します。
		Visible	Visible データを表示するように設定します。

2. サンプリングの確認のためにデータを増減させるスライダーです。AI HMI Slider を使用します。

表 9-4-9 データ変更用スライダープロパティ

項目		設定値	備考
AI HMI Slider	Address	135000	サンプリングするデータ (Data1) を増減できるよう設定します。

3. サンプリングデータを保存するボタンです。AI HMI Button を使用します。

表 9-4-10 サンプリングデータ保存ボタンプロパティ

項目		設定値	備考
AI HMI Switch	SwitchReference	Address	68514
		Enable	チェックあり
		Type	INT
		WordSwitch	Data
			SwitchReference の右側のボタンを押すと表示される ButtonSwitchReference ウィンドウ内で設定します。

- ⑦ 作成した画面を実際に動作させてサンプリング機能が動作することを確認します。
『第6章 作成したアプリの実行環境PCでの動作』を参照し、作成した画面アプリを動作させてください。

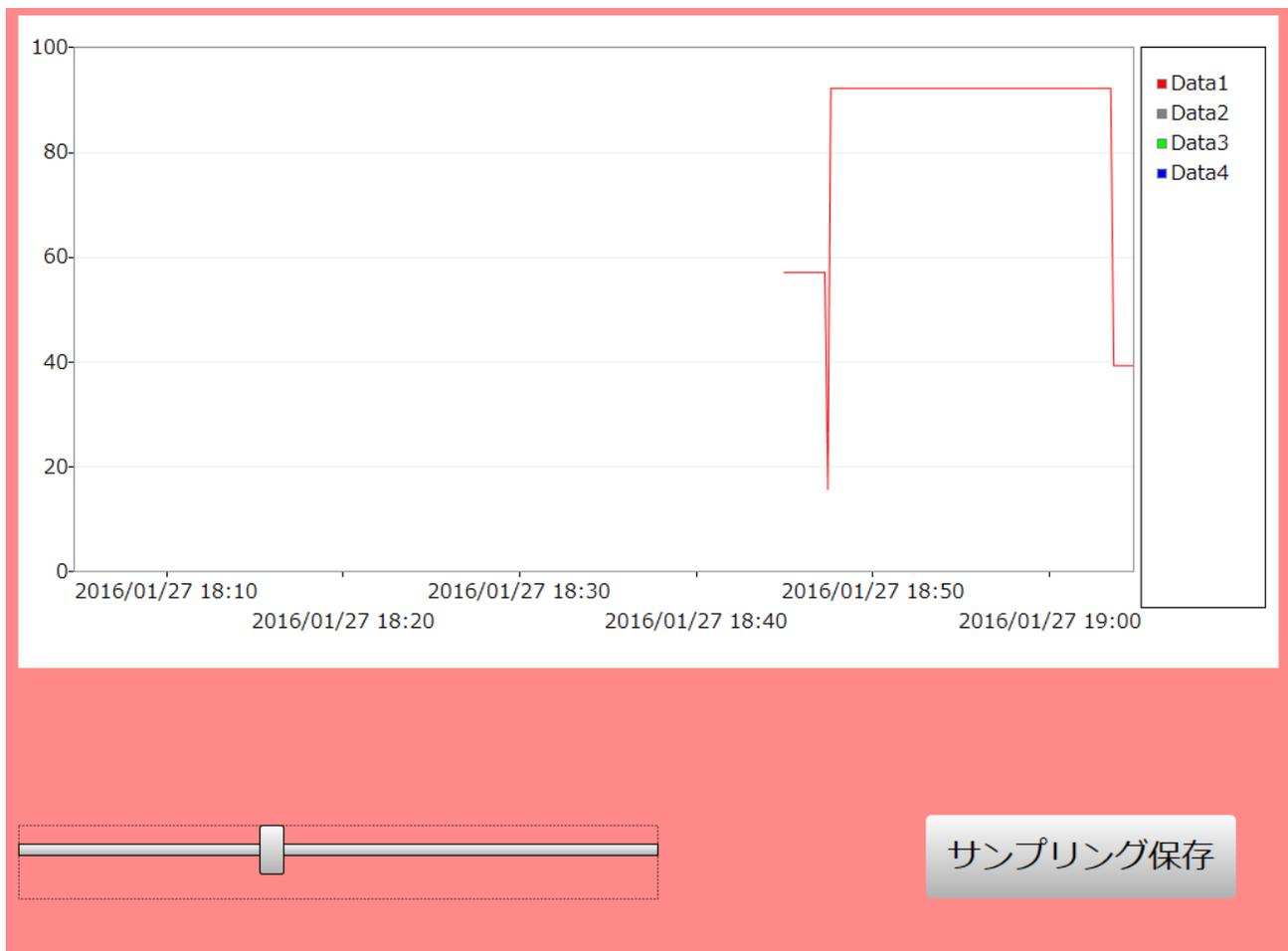


図 9-4-6 サンプリング確認画面の実行

スライダーを左右に動かすことでトレンドグラフの Data1 の値が上下することを確認してください。
また、サンプリング保存ボタンを押すことで exe ファイルと同じフォルダに SamplingFile フォルダが作成され、その中にサンプリング保存ファイルが作成されることを確認してください。

9-5 スケジューラ機能

設定した時刻に値の書き込みをする機能です。

- ① MainWindow.xaml を開き、「ドキュメントアウトライン - [AI_HMI_Window]」を選択してください。
プロパティ欄に AI HMI Window のプロパティが表示されます。

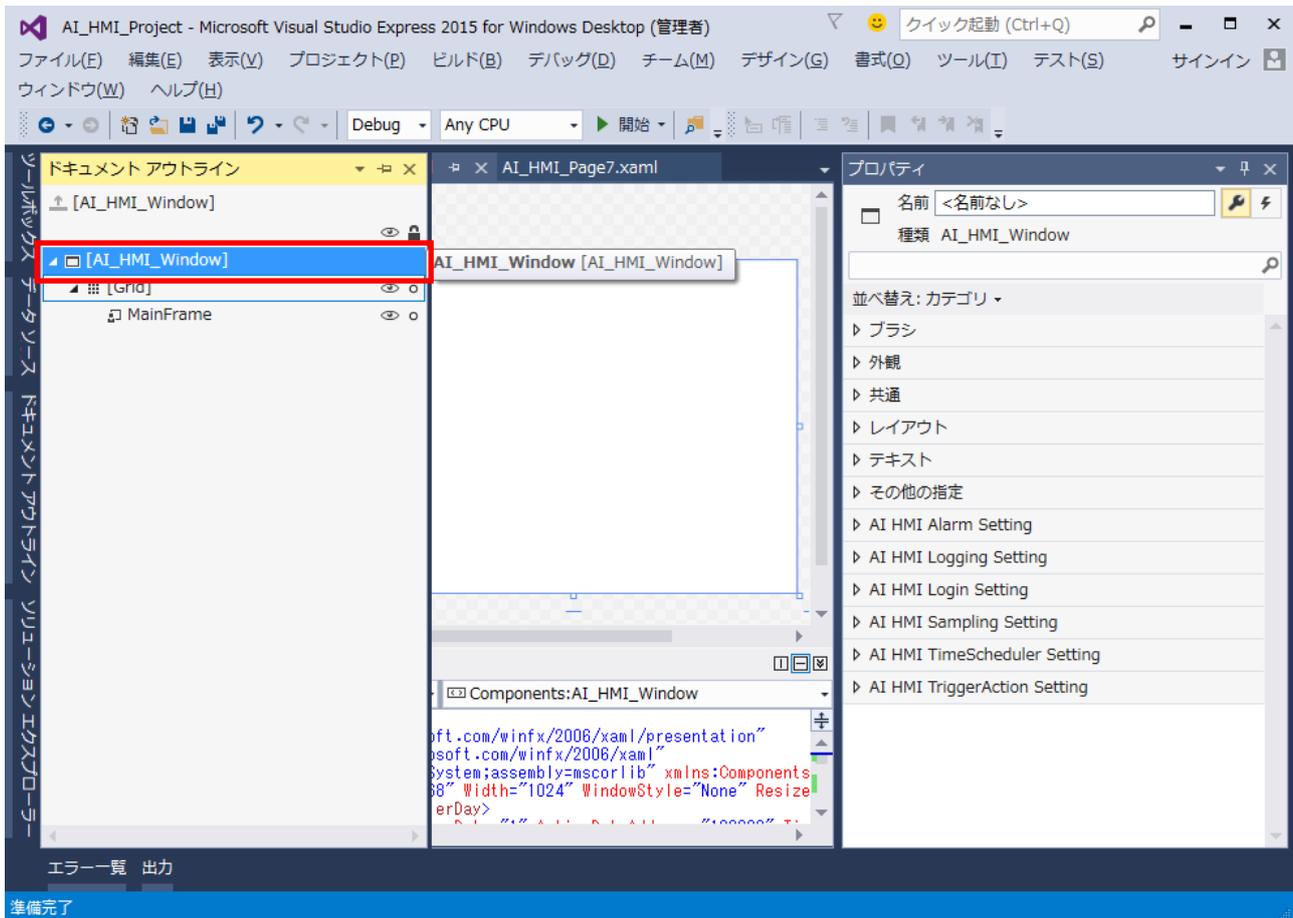


図 9-5-1 AI HMI Window プロパティ

- ② プロパティから「AI HMI TimeScheduler Setting」を選択してください。
AI HMI TimeScheduler Setting プロパティが開きます。

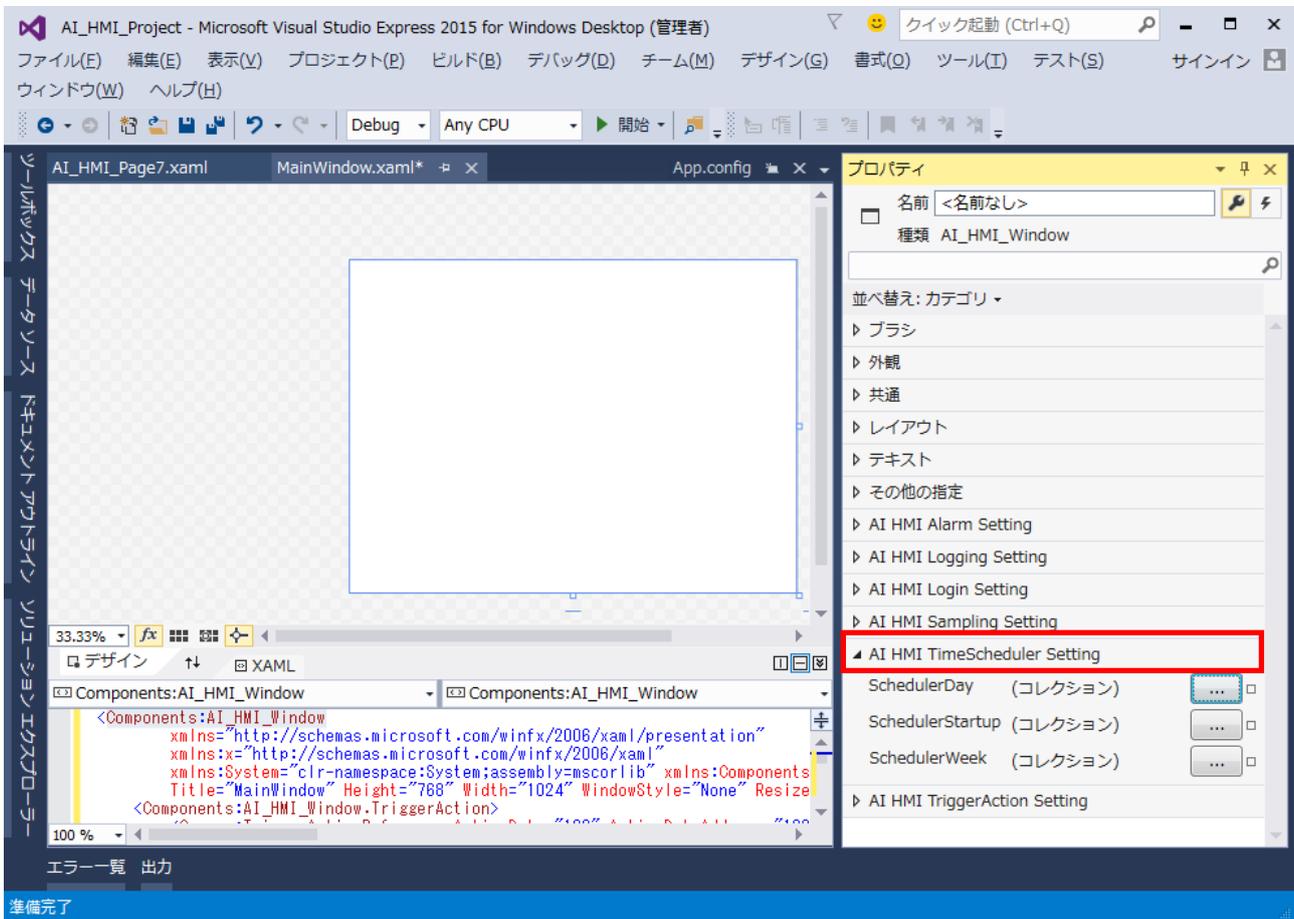


図 9-5-2 AI HMI TimeScheduler Setting プロパティ

AI HMI TimeScheduler Setting プロパティのそれぞれの意味は以下の通りです。

表 9-5-1 AI HMI TimeScheduler Setting

項目	内容
SchedulerDay	指定した日、時、分に値の書き込みを行う機能を設定します。
SchedulerStartup	アプリケーション起動時に値の書き込みを行う機能を設定します。
SchedulerWeek	指定した曜日、時、分に値の書き込みを行う機能を設定します。

AI HMI TimeScheduler Setting プロパティのそれぞれの項目は以下のような設定が可能です。

1. SchedulerDay

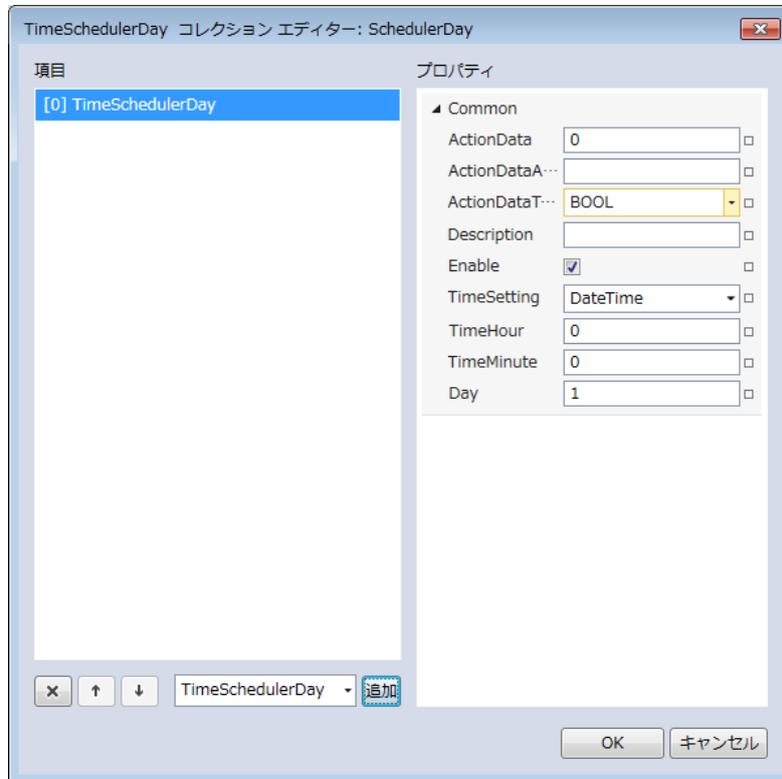


図 9-5-3 SchedulerDay のプロパティ

表 9-5-2 SchedulerDay のプロパティ詳細

項目	データ型	初期値	内容
ActionData	String 型	0	指定した時刻に書き込む値を設定します。
ActionDataAddress	String 型	初期値 なし	指定した時刻に値を書き込む先の変数名または共有メモリアドレスを設定します。
ActionDataType	プルダウン メニュー	BOOL	指定した時刻に書き込む値のデータ型を設定します。
Description	String 型	初期値 なし	コメント欄として利用できます。 オブジェクトの動作に影響は与えません。
Enable	チェック ボックス	チェック あり	TimeSchedulerDay の有効と無効を入力します。 チェックが入っていると選択している TimeSchedulerDay の判定を行います。
TimeSetting	プルダウン メニュー	DateTime	指定する時刻の形式を選択します。 ・ DateTime : TimeHour、TimeMinute、Day で指定した時刻に 値の書き込みを行います。 ・ Hour : TimeHour、Day で指定した時刻に 値の書き込みを行います。 ・ Minute : TimeMinute、Day で指定した時刻に 値の書き込みを行います。
TimeHour	DINT 型	0	値を書き込む時刻の時を設定します。
TimeMinute	DINT 型	0	値を書き込む時刻の分を設定します。
Day	DINT 型	1	値を書き込む時刻の日を設定します。

2. SchedulerStartup

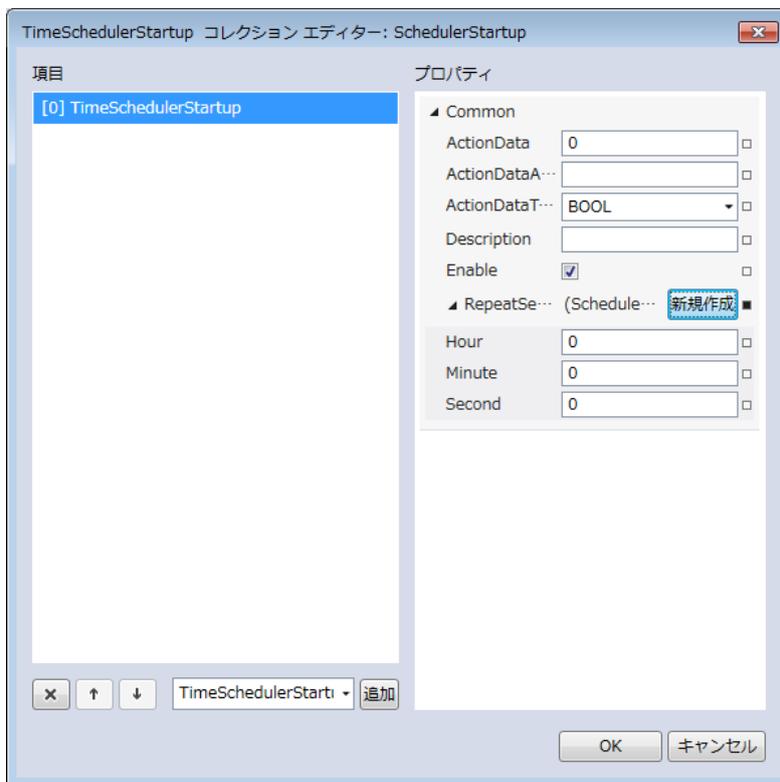


図 9-5-4 SchedulerStartup のプロパティ

表 9-5-3 SchedulerStartup のプロパティ詳細

項目	データ型	初期値	内容
ActionData	String 型	0	アプリケーション起動時に書き込む値を設定します。
ActionDataAddress	String 型	初期値 なし	アプリケーション起動時に値を書き込む先の変数名または共有メモリアドレスを設定します。
ActionDataType	プルダウン メニュー	BOOL	アプリケーション起動時に書き込む値のデータ型を設定します。
Description	String 型	初期値 なし	コメント欄として利用できます。 オブジェクトの動作に影響は与えません。
Enable	チェック ボックス	チェック あり	TimeSchedulerStartup の有効と無効を入力します。 チェックが入っていると選択している TimeSchedulerStartup の 判定を行います。
Repeat Setting	Hour	DINT 型	繰り返し実行する周期(時、分、秒)を設定します。 起動時に 1 度実行した後、周期毎に繰り返し実行するよう になります。 3 項目とも 0 を設定していると繰り返し実行をしません。
	Minute	DINT 型	
	Second	DINT 型	

3. SchedulerWeek

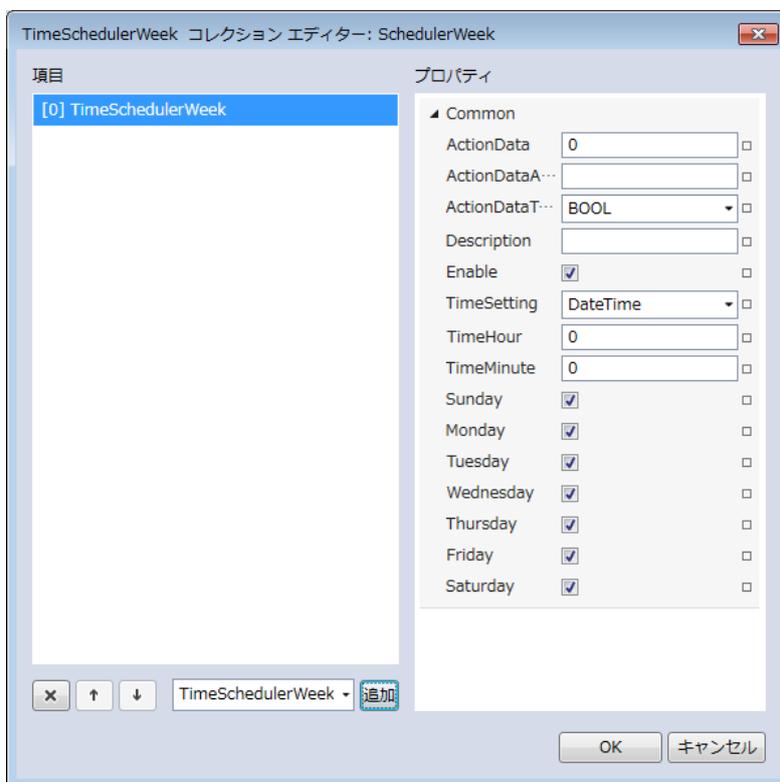


図 9-5-5 SchedulerWeek のプロパティ

表 9-5-4 SchedulerWeek のプロパティ詳細

項目	データ型	初期値	内容
ActionData	String 型	0	指定した時刻に書き込む値を設定します。
ActionDataAddress	String 型	初期値 なし	指定した時刻に値を書き込む先の変数名または共有メモリアドレスを設定します。
ActionDataType	プルダウン メニュー	BOOL	指定した時刻に書き込む値のデータ型を設定します。
Description	String 型	初期値 なし	コメント欄として利用できます。 オブジェクトの動作に影響は与えません。
Enable	チェック ボックス	チェック あり	TimeSchedulerWeek の有効と無効を入力します。 チェックが入っていると選択している TimeSchedulerWeek の判定を行います。
TimeSetting	プルダウン メニュー	DateTime	指定する時刻の形式を選択します。 ・ DateTime : TimeHour、TimeMinute、曜日で指定した時刻に 値の書き込みを行います。 ・ Hour : TimeHour、曜日で指定した時刻に 値の書き込みを行います。 ・ Minute : TimeMinute、曜日で指定した時刻に 値の書き込みを行います。
TimeHour	DINT 型	0	値を書き込む時刻の時を設定します。
TimeMinute	DINT 型	0	値を書き込む時刻の分を設定します。
Sunday ~ Saturday	チェック ボックス	チェック あり	値を書き込む時刻の曜日を設定します。

- ③ 実際に指定した時刻に値を書き込むように設定します。
SchedulerDay のプロパティ表 9-5-5 のように設定してください。

表 9-5-5 SchedulerDay の設定

項目	設定内容	備考
ActionData	1	共有メモリの 100200 に指定した時刻に 1 を UINT 型で書き込みます。
ActionDataAddress	100200	
ActionDataType	UINT	
Description		
Enable	チェックあり	
TimeSetting	DateTime	指定した日、時、分に値を書き込みます。
TimeHour	任意の値	
TimeMinute		
Day		

- ④ 実際に画面アプリを起動し、TimeHour、TimeMinute、Day で指定した時刻に値が書き込まれることを確認してください。

- ※ SchedulerDay で設定した内容は毎月指定した日付、時刻に実行されます。
- ※ SchedulerWeek で設定した内容は毎週指定した曜日、時刻に実行されます。

9-6 トリガアクション機能

特定の変数の挙動を参照して他の変数へ値を書き込む機能です。

- ① MainWindow.xaml を開き、「ドキュメントアウトライン - [AI_HMI_Window]」を選択してください。
プロパティ欄に AI HMI Window のプロパティが表示されます。

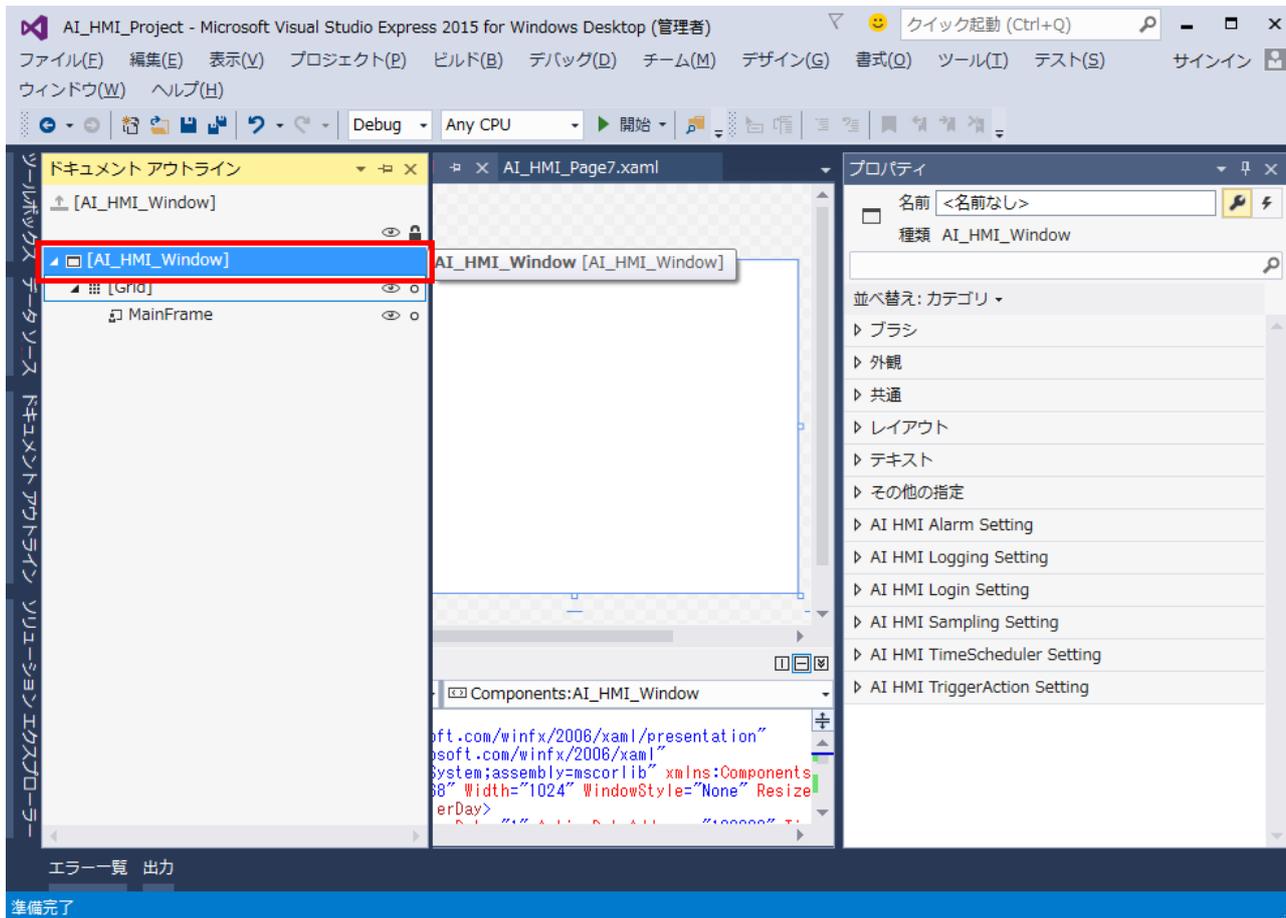


図 9-6-1 AI HMI Window プロパティ

- ② プロパティから「AI HMI TriggerAction Setting」を選択してください。
AI HMI TriggerAction Setting プロパティが開きます。

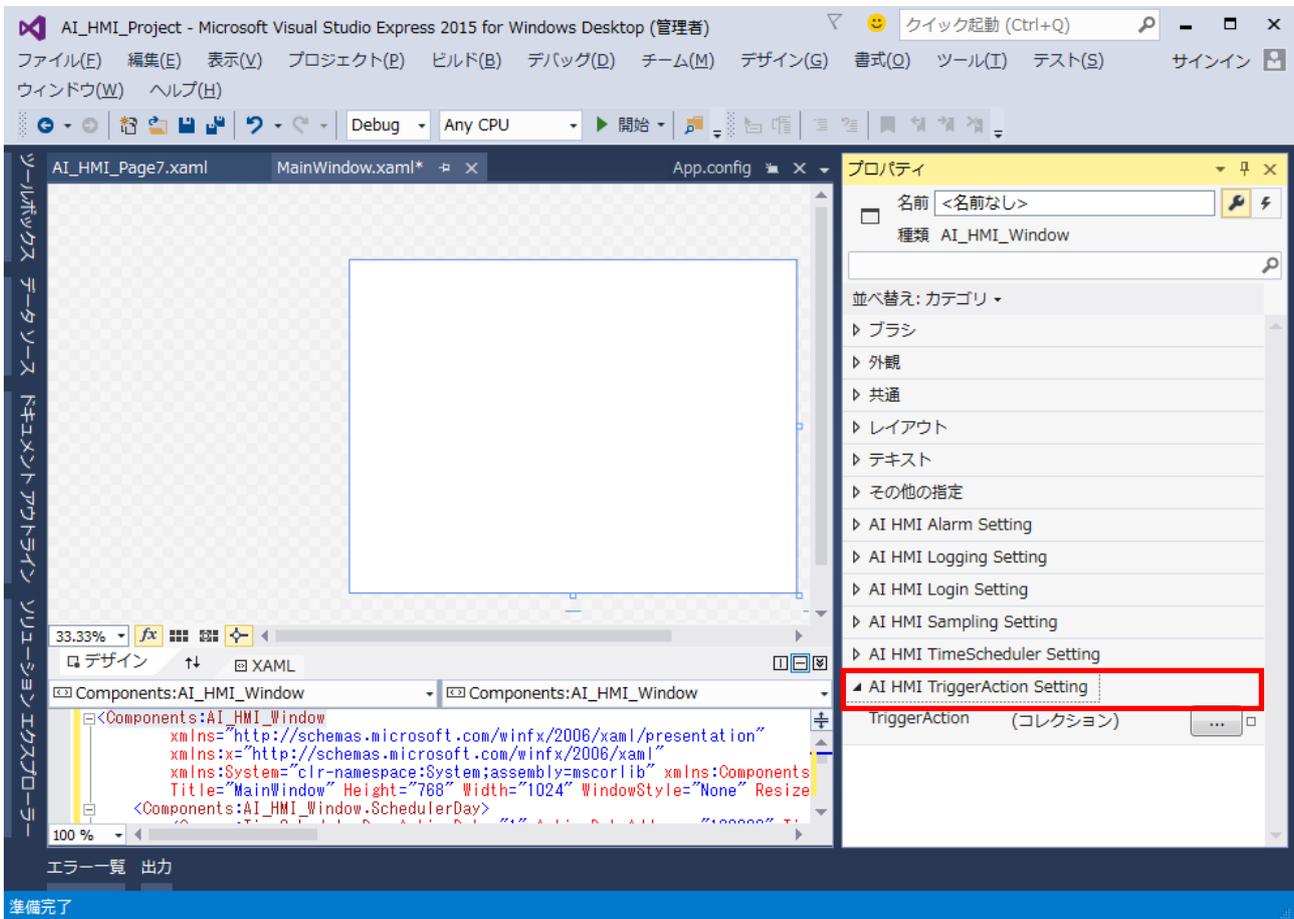


図 9-6-2 AI HMI TriggerAction Setting プロパティ

AI HMI TimeScheduler Setting プロパティのそれぞれの意味は以下の通りです。

表 9-6-1 AI HMI TriggerAction Setting

項目	内容
TriggerAction	特定の値が設定した条件に一致したときに値の書き込みを行う機能を設定します。

TriggerAction の右のボタンを押すことで TriggerActionReference の設定ができます。
TriggerActionReference の詳細は以下の通りです。

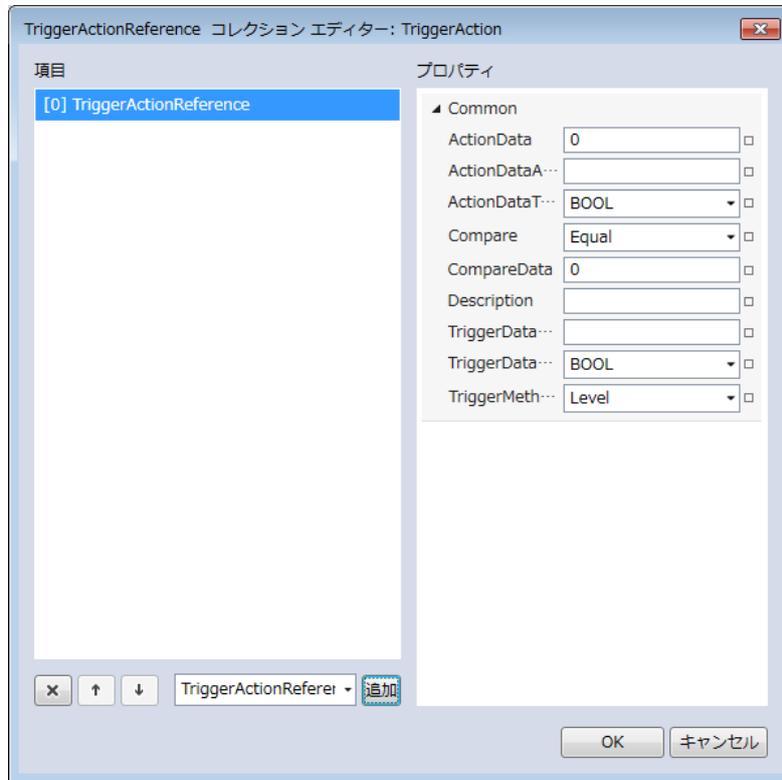


図 9-6-3 TriggerActionReference のプロパティ

表 9-6-2 TriggerActionReference のプロパティ詳細

項目	データ型	初期値	内容
ActionData	String 型	0	TriggerDataAddress に格納されている値が CompareData と Compare の条件に一致したときに書き込む値を設定します。
ActionDataAddress	String 型	初期値 なし	TriggerDataAddress に格納されている値が CompareData と Compare の条件に一致したときに書き込む先の変数名または共有メモリアドレスを設定します。
ActionDataType	プルダウン メニュー	BOOL	TriggerDataAddress に格納されている値が CompareData と Compare の条件に一致したときに書き込む値のデータ型を設定します。
Compare	プルダウン メニュー	Equal	TriggerDataAddress に格納されている値と CompareData を比較する際の条件を設定します。
CompareData	String 型	0	TriggerDataAddress に格納されている値と比較する値を設定します。
Description	String 型	初期値 なし	コメント欄として利用できます。 オブジェクトの動作に影響は与えません。
TriggerDataAddress	String 型	0	トリガとして参照する変数名または共有メモリアドレスを設定します。
TriggerDataType	プルダウン メニュー	BOOL	トリガとして参照する値のデータ型を設定します。
TriggerMethod	プルダウン メニュー	Level	トリガの方式を設定します。 <ul style="list-style-type: none"> ・ Level : レベルトリガ ・ RisingEdge : 立上りエッジ ・ FallingEdge : 立下りエッジ

- ③ 実際にトリガアクション機能で値を書き込むように設定します。
TriggerActionReference を 2 件登録してそれぞれのプロパティを表 9-6-3、9-6-4 のように設定してください。

表 9-6-3 [0]TriggerActionReference の設定

項目	設定内容	備考
ActionData	100	共有メモリの 100402 の値が 0 になったとき、共有メモリの 100400 に 100 を UINT 型で書き込みます。
ActionDataAddress	100400	
ActionDataType	UINT	
Compare	Equal	
CompareData	0	
Description		
TriggerDataAddress	100402	
TriggerDataType	UINT	
TriggerMethod	RigingEdge	

表 9-6-4 [1]TriggerActionReference の設定

項目	設定内容	備考
ActionData	200	共有メモリの 100402 の値が 1 になったとき、共有メモリの 100400 に 200 を UINT 型で書き込みます。
ActionDataAddress	100400	
ActionDataType	UINT	
Compare	Equal	
CompareData	1	
Description		
TriggerDataAddress	100402	
TriggerDataType	UINT	
TriggerMethod	RigingEdge	

- ④ 設定したトリガアクションの動作を確認する画面を作成します。
図 9-6-4 のように部品を配置してください。

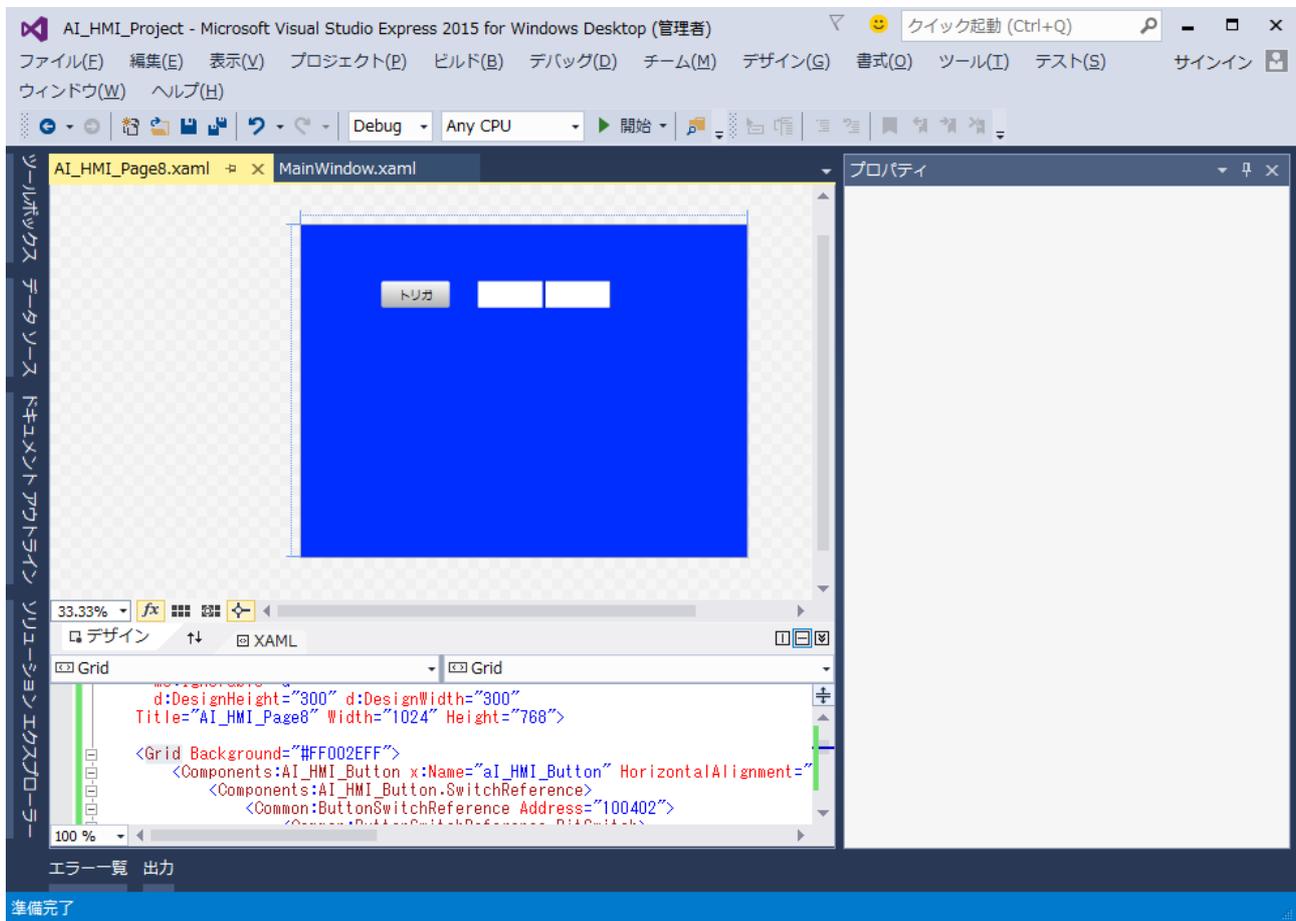


図 9-6-4 トリガアクション確認用画面の作成

それぞれの部品のプロパティは以下のように設定してください。

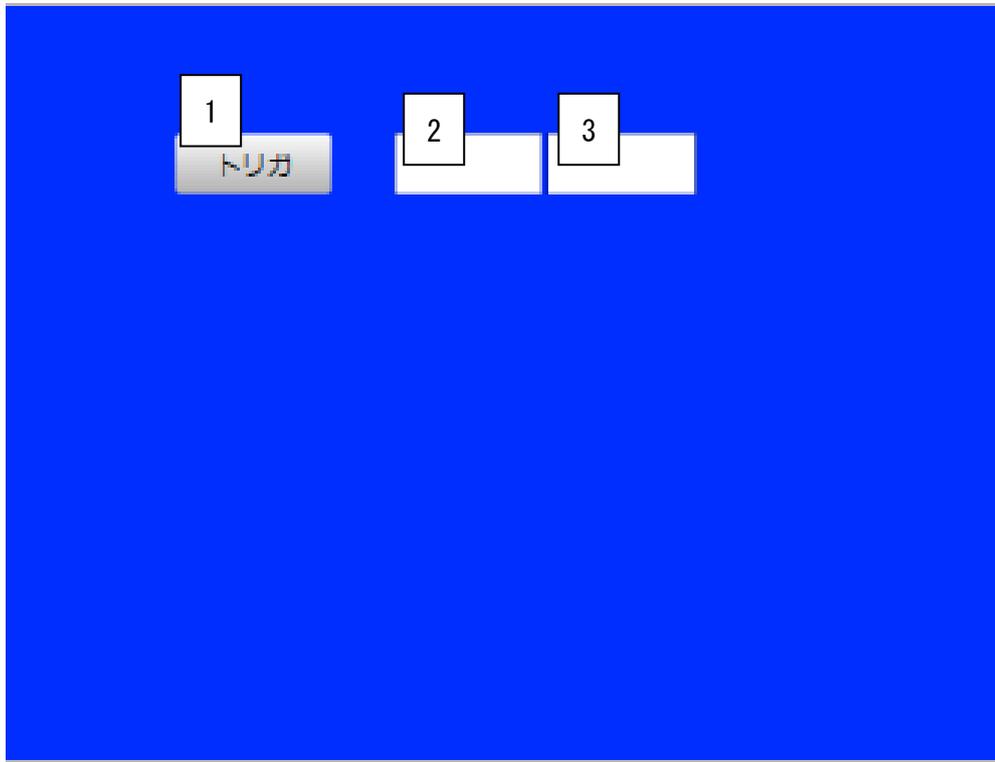


図 9-6-5 トリガアクション確認用画面の詳細

1. トリガとなる変数を ON/OFF するボタンです。AI HMI Button を使用します。

表 9-6-5 トリガ変数切替ボタンプロパティ

項目		設定値	備考
AI HMI Switch	SwitchReference	Address	100402
		BitSwitch BitAction	Reversal
SwitchReference の右側のボタンを押すと表示される ButtonSwitchReference ウィンドウ内で設定します。			

2. トリガとなる変数を表示する欄です。AI HMI Label を使用します。

表 9-6-6 トリガ変数表示欄プロパティ

項目		設定値	備考
AI HMI Display	Address	100402	
	LabelType	Numerical	
	Numerical	DataType	UINT

3. トリガアクションで書き込まれた値を表示する欄です。AI HMI Label を使用します。

表 9-6-7 トリガアクション書き込み先表示欄プロパティ

項目		設定値	備考
AI HMI Display	Address	100400	
	LabelType	Numerical	
	Numerical	DataType	UINT

- ⑤ 実際に画面アプリを起動し、トリガとなる変数を変化させることでトリガアクション書き込み先の値も変化することを確認してください。

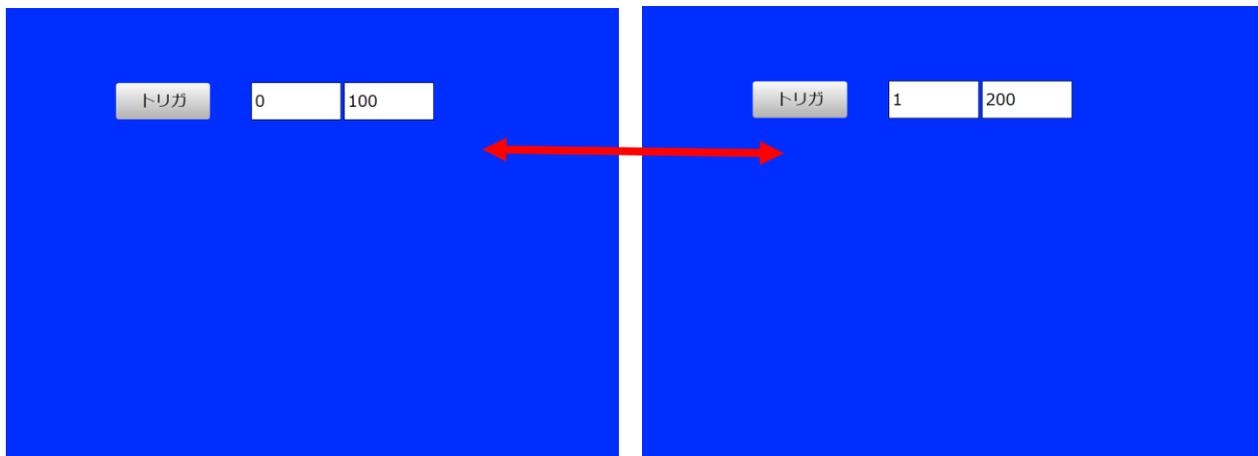


図 9-6-6 トリガアクションの確認

9-7 レシピ切替機能

複数の変数の対して同時に値を書き込んだり、現在の値をレシピとして保存したりする機能です。

- ① ツールボックスから AI HMI Recipe を AI HMI の画面に配置してください。

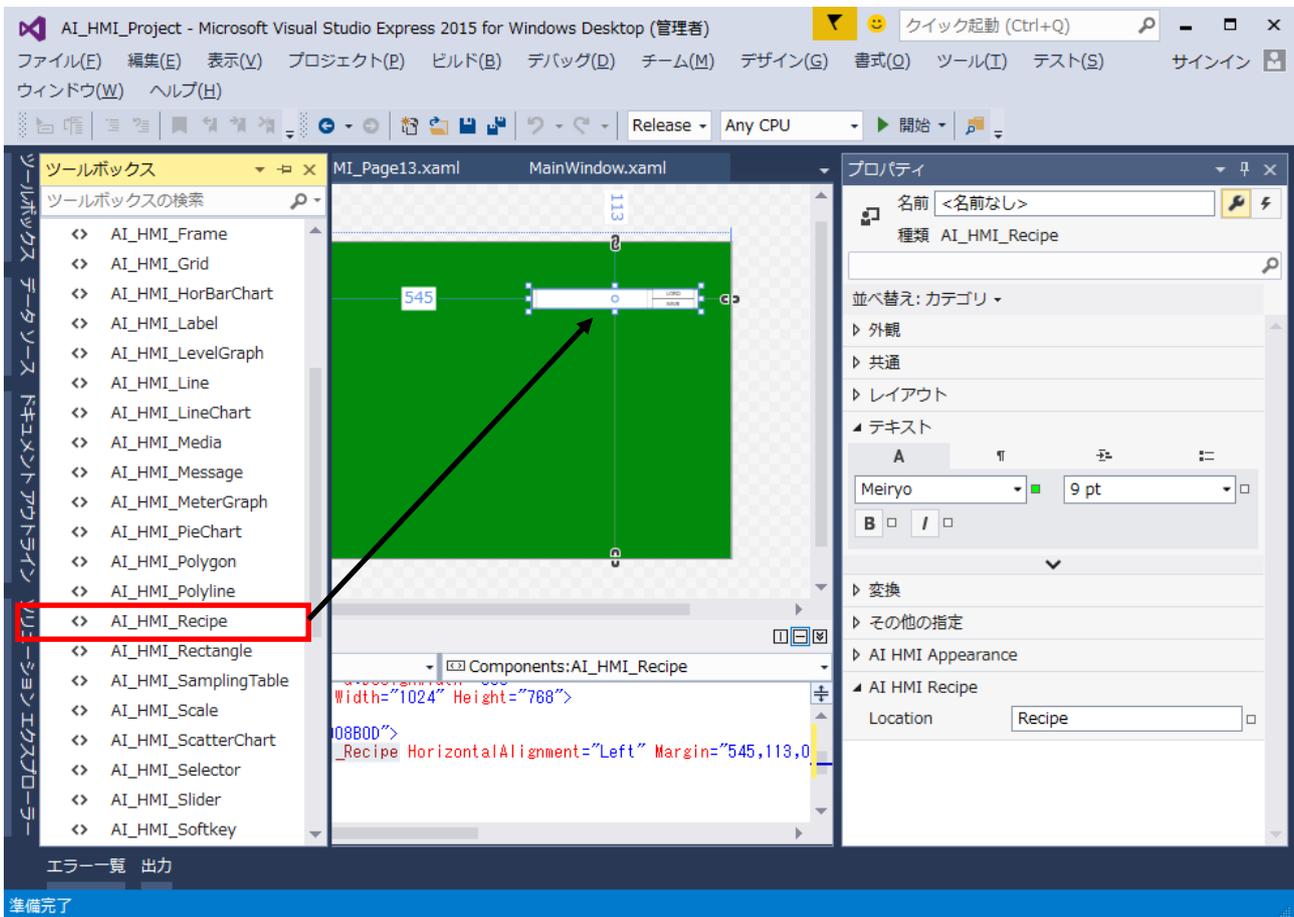


図 9-7-1 AI HMI Recipe

- ② 配置された AI HMI Recipe のプロパティを表 9-7-1 のように設定してください。

表 9-7-1 AI HMI Recipe 設定内容

項目		設定内容	備考
AI HMI Recipe	Location	RecipeFile	レシピファイルを格納するフォルダを設定します。 exe ファイルと同じフォルダ内の RecipeFile フォルダ内のファイルを読み書きするようにします。

- ③ AI HMI Recipe は設定したフォルダ内の csv ファイルをレシピファイルとして参照します。レシピファイルの形式はリスト 9-7-1 のようになります。(エンコード形式 S-JIS)

リスト 9-7-1 レシピファイルの内容

Address,	DataType,	Data
120000,	INT,	100
120002,	UINT,	356
120004,	UINT,	65535
120006,	DINT,	100000
120010,	BOOL,	1
120011,	REAL,	3204.24
120015,	LREAL,	5600.03
140000,	STRING,	RecipeDataA

動作の確認のため、RecipeA.csv、RecipeB.csv という 2 つのファイルを作成し、リスト 9-7-2、9-7-3 のように編集してください。

リスト 9-7-2 RecipeA.csv の内容

Address,	DataType,	Data
120000,	INT,	100
120002,	UINT,	65535
120004,	UDINT,	300

リスト 9-7-3 RecipeB.csv の内容

Address,	DataType,	Data
120000,	INT,	
120002,	UINT,	
120004,	UDINT,	

※ CSV ファイルに STRING 型データ記述する場合、記述ルールに従っていただく必要があります。詳細は『17-13 CSV ファイルで STRING 型の値を使用したい』をご参照ください。

- ④ レシピの動作を確認するために AI HMI Label を配置します。
図 9-7-2 のように AI HMI Label を配置してください。

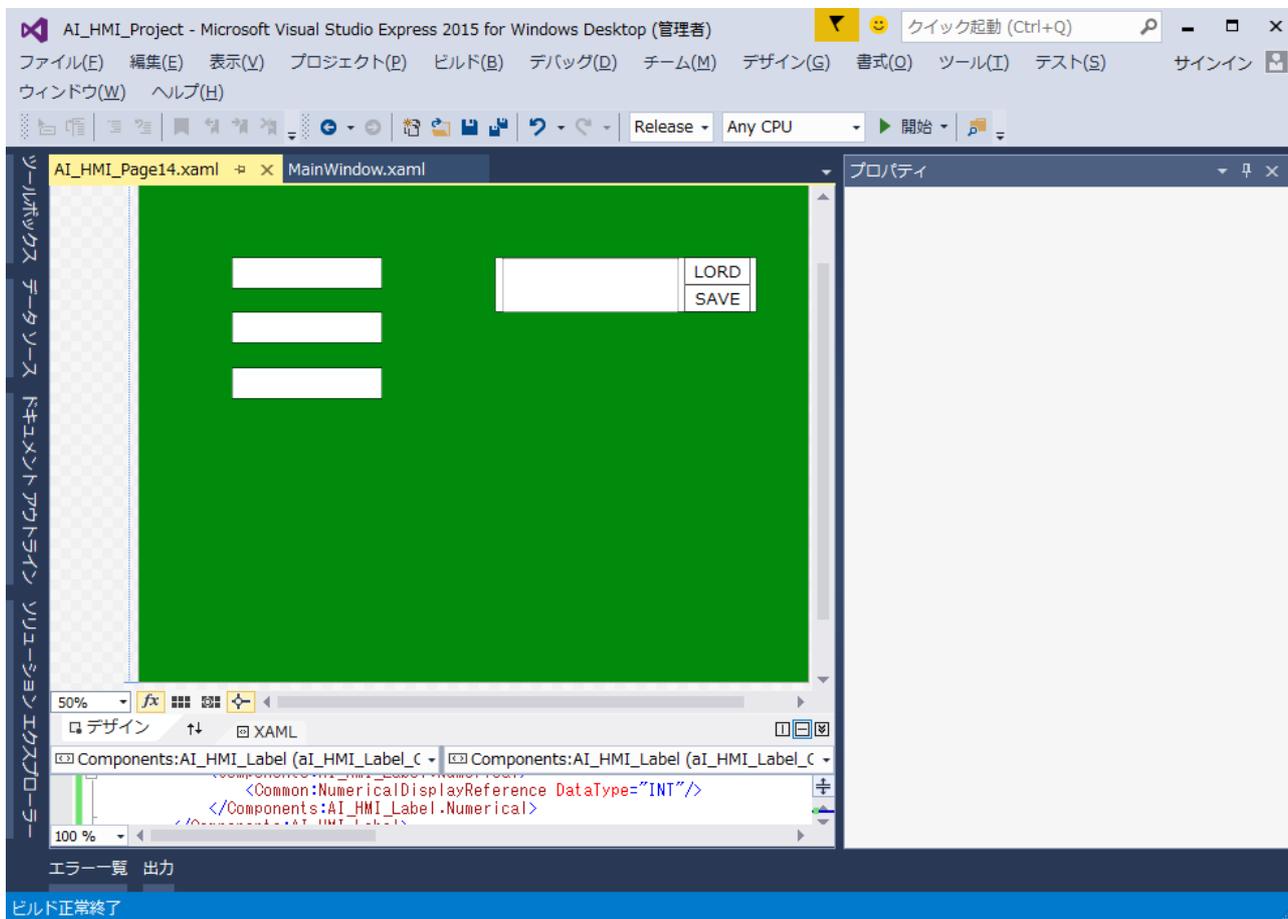


図 9-7-2 動作確認用 AI HMI Label

それぞれの AI HMI Label は以下のように設定してください。

表 9-7-2 レシピ内容表示欄 1

項目		設定値	備考
AI HMI Display	Address	120000	
	LabelType	Numerical	
	Numerical	DataType	INT

表 9-7-3 レシピ内容表示欄 2

項目		設定値	備考
AI HMI Display	Address	120002	
	LabelType	Numerical	
	Numerical	DataType	UINT

表 9-7-4 レシピ内容表示欄 3

項目		設定値	備考
AI HMI Display	Address	120004	
	LabelType	Numerical	
	Numerical	DataType	UDINT

- ⑤ 作成した画面を実際に動作させてレシピ機能が動作することを確認します。

『第6章 作成したアプリの実行環境PCでの動作』を参照し、作成した画面アプリを動作させてください。

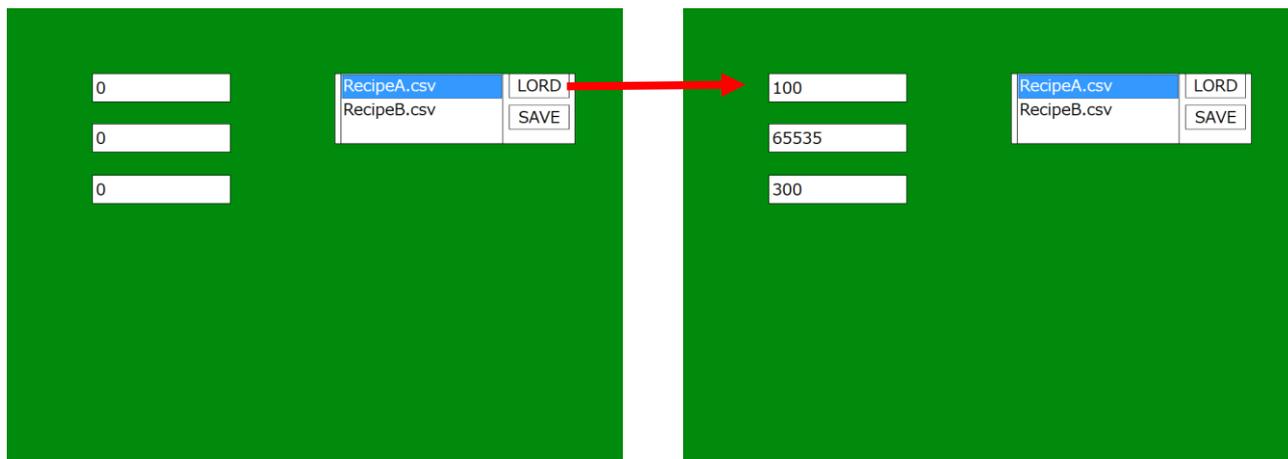


図 9-7-3 レシピ確認画面の実行

RecipeA.csv を選択している状態で「LOAD」を押すことで RecipeA.csv の内容を読み取ることを確認してください。

また、RecipeB.csv を選択している状態で「SAVE」を押すことで現在の値を RecipeB.csv に書き込むことを確認してください。

9-8 統計グラフ機能

AI HMI では複数の変数または共有メモリアドレスを参照し、グラフとして表示することができます。これらグラフは見た目により数種類用意されていますが、基本的に同様の設定でデータを表示することが可能です。

本章では AI_HMI_ColBarChart (棒グラフ) を設定する方法について解説します。

詳しい設定方法やその他の種類のグラフについては AI HMI リファレンスマニュアル『第 29 章 AI HMI Chart』を参照ください。

- ① ツールボックスから AI HMI ColBarChart を AI HMI の画面に配置してください。

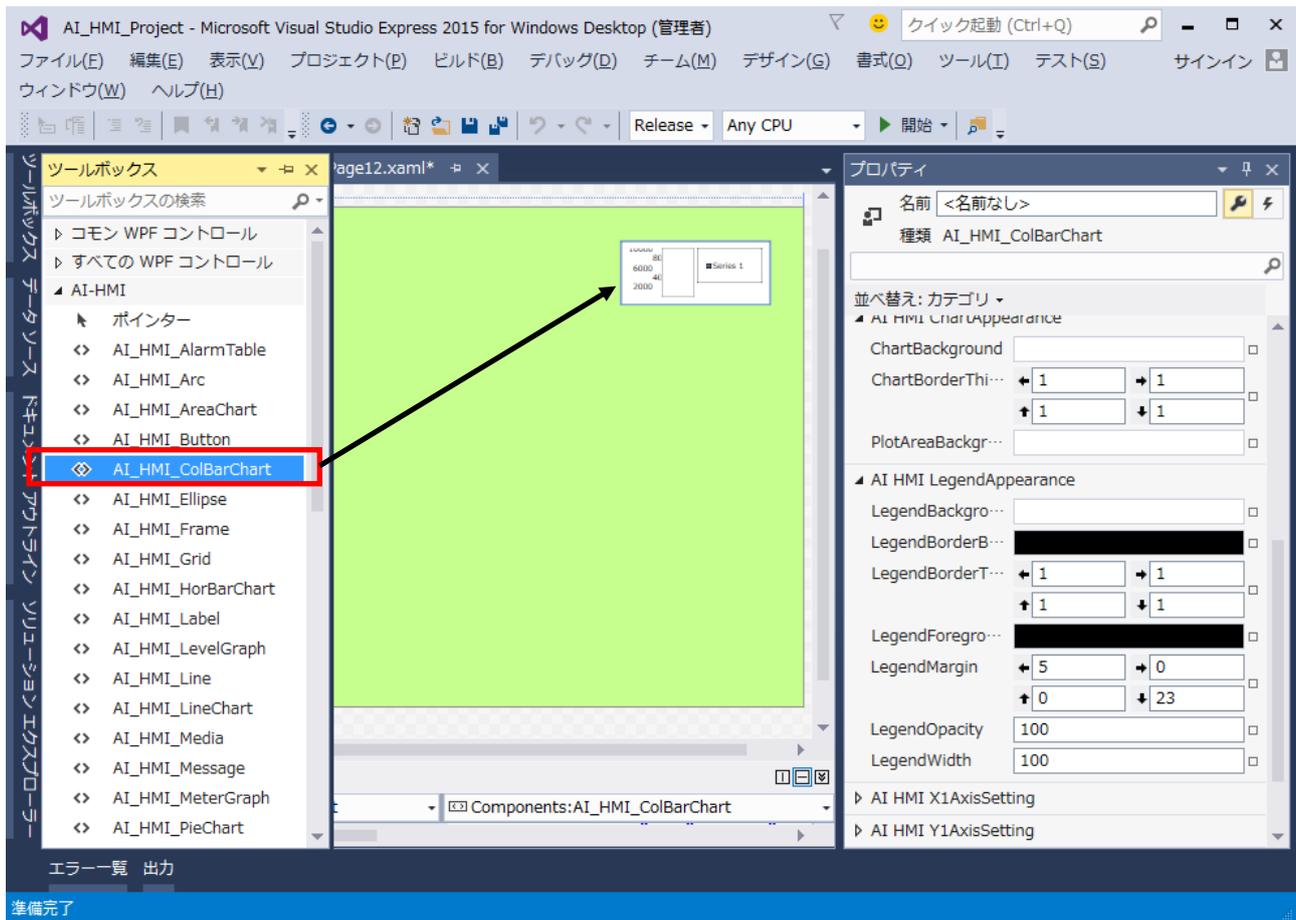


図 9-8-1 AI HMI ColBarChart

- ② 配置された AI HMI ColBarChart を任意の大きさに変更し、プロパティの AI HMI Chart Data – Series の右側にあるボタンを押してください。

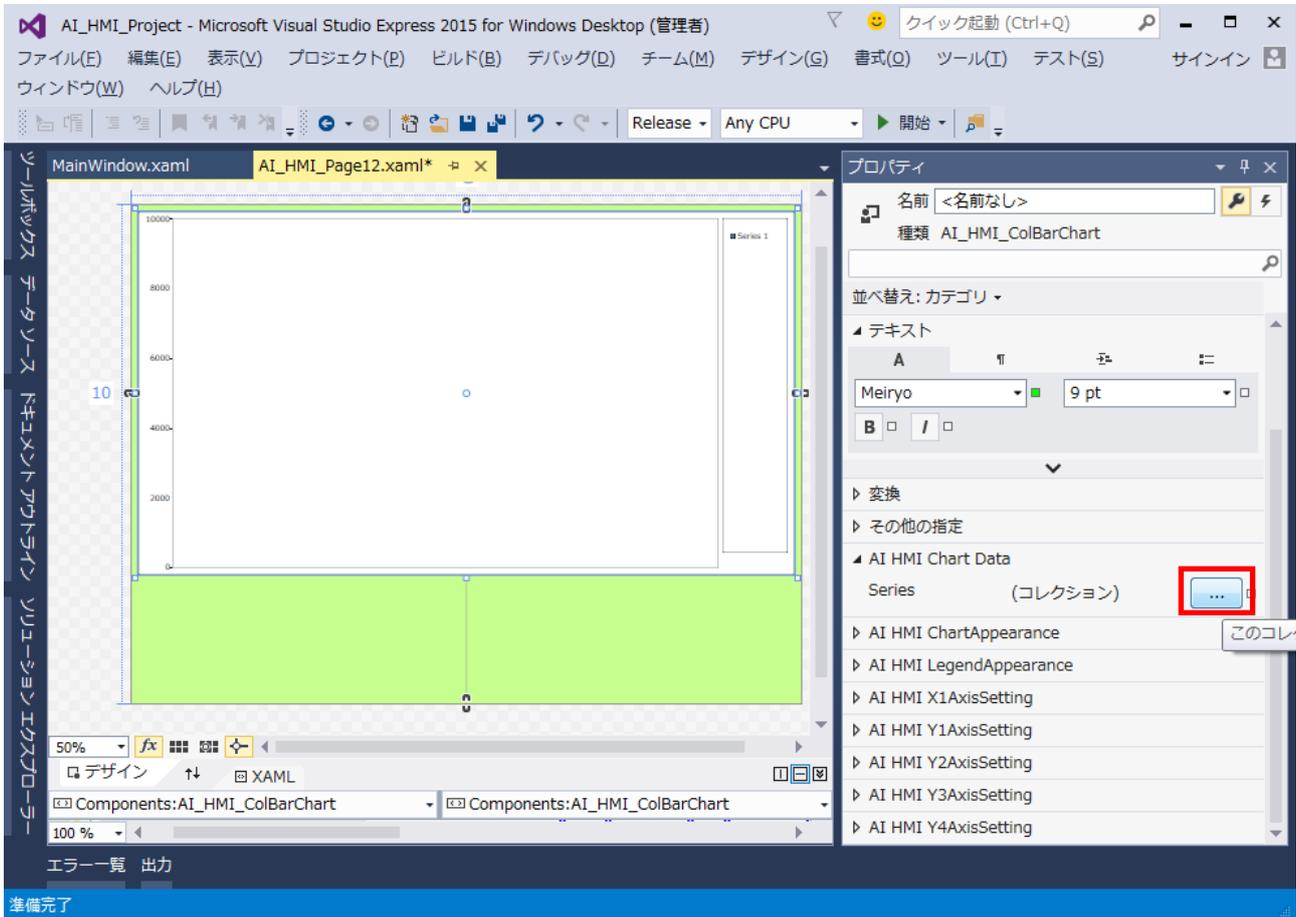


図 9-8-2 AI HMI ColBarChart の設定

- ③ ColBarChartSeriesReference のコレクションエディターが開きます。
「追加」ボタンを2回押し、ColBarChartSeriesReference を2行追加してください。

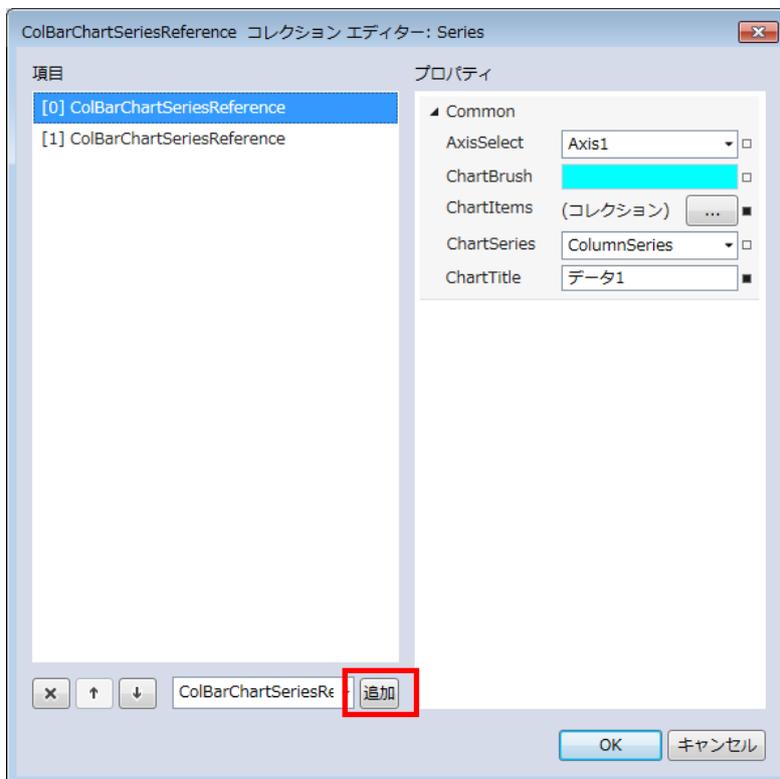


図 9-8-3 ColBarChartSeriesReference

それぞれの ColBarChartSeriesReference は以下のように設定してください。

表 9-8-1 [0]ColBarChartSeriesReference 設定内容

項目	設定内容	備考
ChartBrush	青	グラフの色です。
ChartItems	後述	グラフに表示するデータの詳細を設定します。
ChartTitle	データ 1	グラフの凡例に表示する文字列です。

表 9-8-2 [1]ColBarChartSeriesReference 設定内容

項目	設定内容	備考
ChartBrush	赤	グラフの色です。
ChartItems	後述	グラフに表示するデータの詳細を設定します。
ChartTitle	データ 2	グラフの凡例に表示する文字列です。

それぞれの ColBarChartSeriesReference の設定で ChartItem の右側のボタンを押すと ChartSeriesItemReference のコレクションエディタが開きます。
それぞれで「追加」ボタンを2回押し ChartSeriesItemReference を2行追加してください。

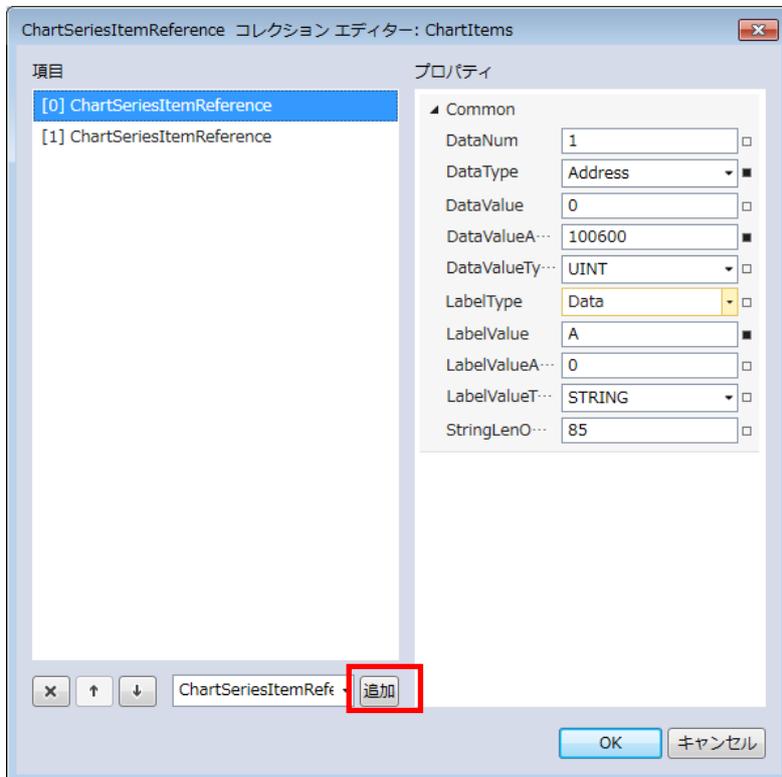


図 9-8-4 ChartSeriesItemReference

それぞれの ChartSeriesItemReference は以下のように設定してください。

表 9-8-3 [0]ColBarChartSeriesReference の[0]ChartSeriesItemReference 設定内容

項目	設定内容	備考
DataType	Address	共有メモリアドレス 100600 の値をグラフで表示するようにします。
DataValueAddress	100600	
LabelValue	A	グラフの X 軸方向の値を設定します。

表 9-8-4 [0]ColBarChartSeriesReference の[1]ChartSeriesItemReference 設定内容

項目	設定内容	備考
DataType	Address	共有メモリアドレス 100602 の値をグラフで表示するようにします。
DataValueAddress	100602	
LabelValue	B	グラフの X 軸方向の値を設定します。

表 9-8-5 [1]ColBarChartSeriesReference の[0]ChartSeriesItemReference 設定内容

項目	設定内容	備考
DataType	Address	共有メモリアドレス 100604 の値をグラフで表示するようにします。
DataValueAddress	100604	
LabelValue	A	グラフの X 軸方向の値を設定します。

表 9-8-6 [1]ColBarChartSeriesReference の[1]ChartSeriesItemReference 設定内容

項目	設定内容	備考
DataType	Address	共有メモリアドレス 100606 の値をグラフで表示するようにします。
DataValueAddress	100606	
LabelValue	B	グラフの X 軸方向の値を設定します。

- ④ グラフの動作を確認するために AI HMI Slider を配置します。

図 9-9-5 のように AI HMI Slider を配置してください。

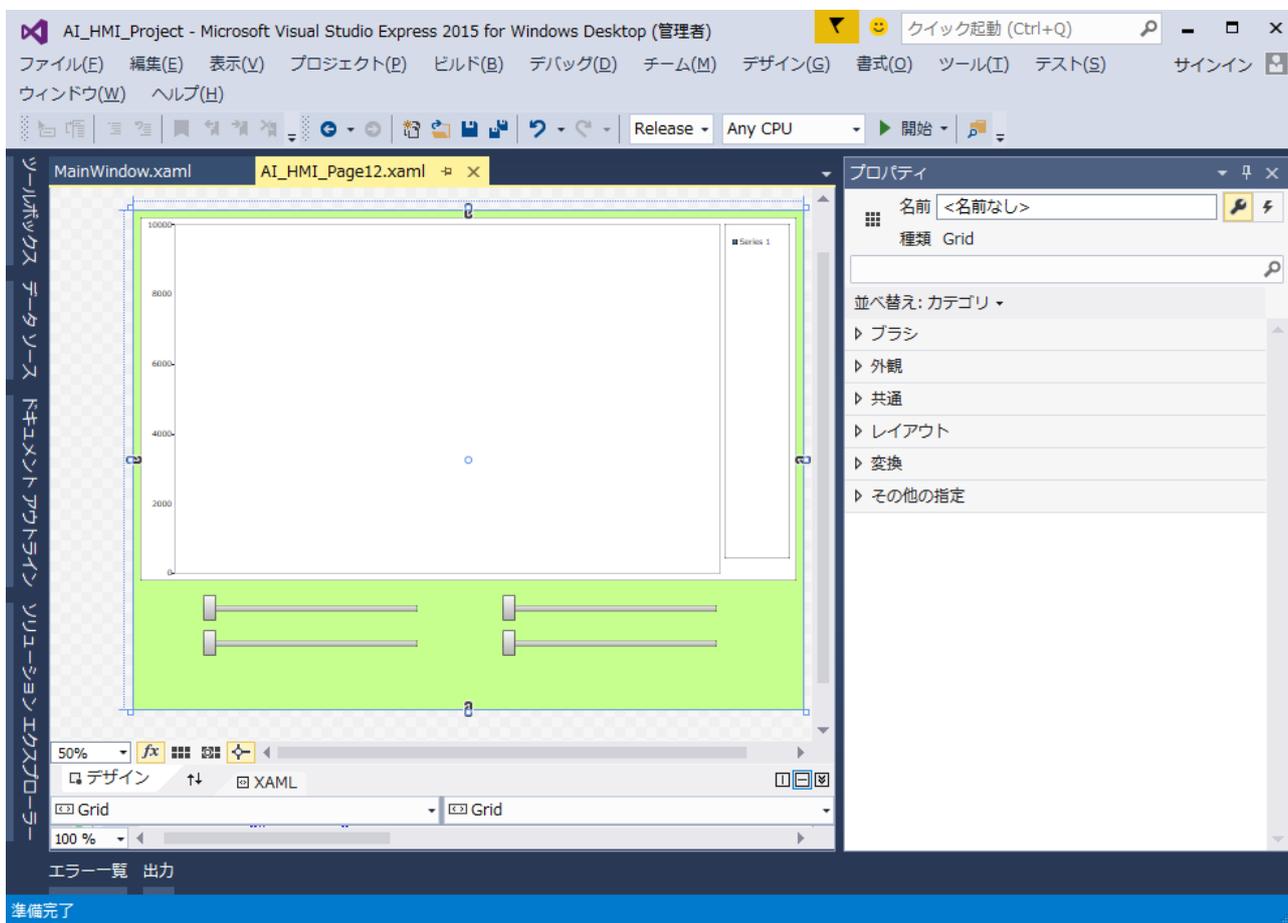


図 9-8-5 動作確認用 AI HMI Slider

表 9-8-7 動作確認用スライダー1 プロパティ

項目		設定値	備考
AI HMI Slider	Address	100600	グラフ表示するデータ 1 を UINT 型で 0~10000 の範囲で増減できるようにします。
	DataType	UINT	
	Maximum	10000	

表 9-8-8 動作確認用スライダー2 プロパティ

項目		設定値	備考
AI HMI Slider	Address	100602	グラフ表示するデータ 2 を UINT 型で 0~10000 の範囲で増減できるようにします。
	DataType	UINT	
	Maximum	10000	

表 9-8-9 動作確認用スライダー3 プロパティ

項目		設定値	備考
AI HMI Slider	Address	100604	グラフ表示するデータ 3 を UINT 型で 0~10000 の範囲で増減できるようにします。
	DataType	UINT	
	Maximum	10000	

表 9-8-10 動作確認用スライダー4 プロパティ

項目		設定値	備考
AI HMI Slider	Address	100606	グラフ表示するデータ 3 を UINT 型で 0~10000 の範囲で増減できるようにします。
	DataType	UINT	
	Maximum	10000	

- ⑤ 作成した画面を実際に動作させてグラフ機能が動作することを確認します。
『第6章 作成したアプリの実行環境PCでの動作』を参照し、作成した画面アプリを動作させてください。

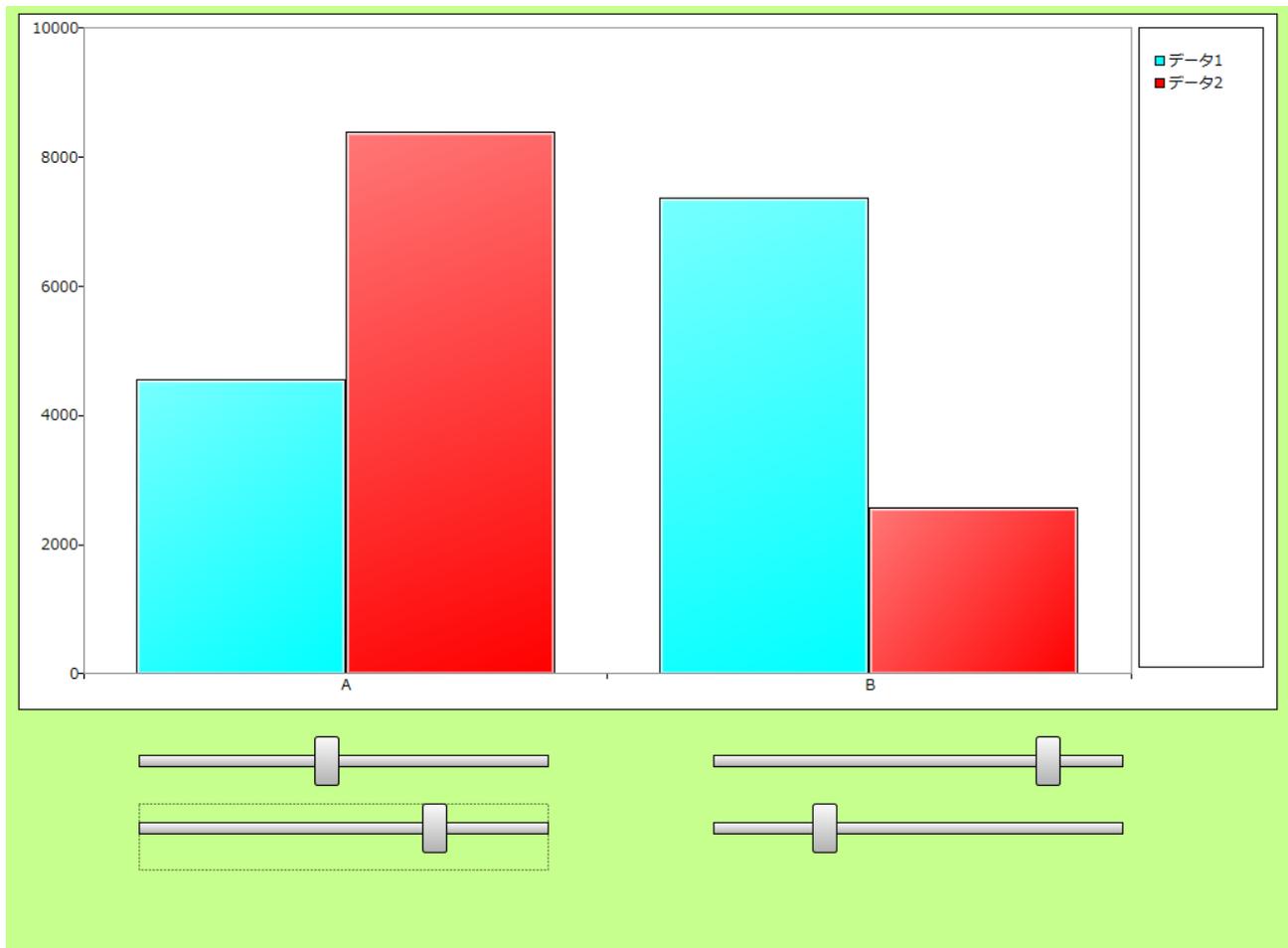


図 9-8-6 グラフ機能確認画面の実行

スライダーを左右に動かすことでグラフの表示が変化することを確認してください。

9-9 カスタマイズ ソフトキー

AI HMI ではデフォルト機能としてソフトウェアキーボードコンポーネント (AI HMI Softkey) を実装していますが、ユーザが自由にキー配置をカスタマイズしたソフトウェアキーボードを作成することもできます。

- ① ツールボックスから AI HMI KeyCodeButton を AI HMI の画面に配置してください。

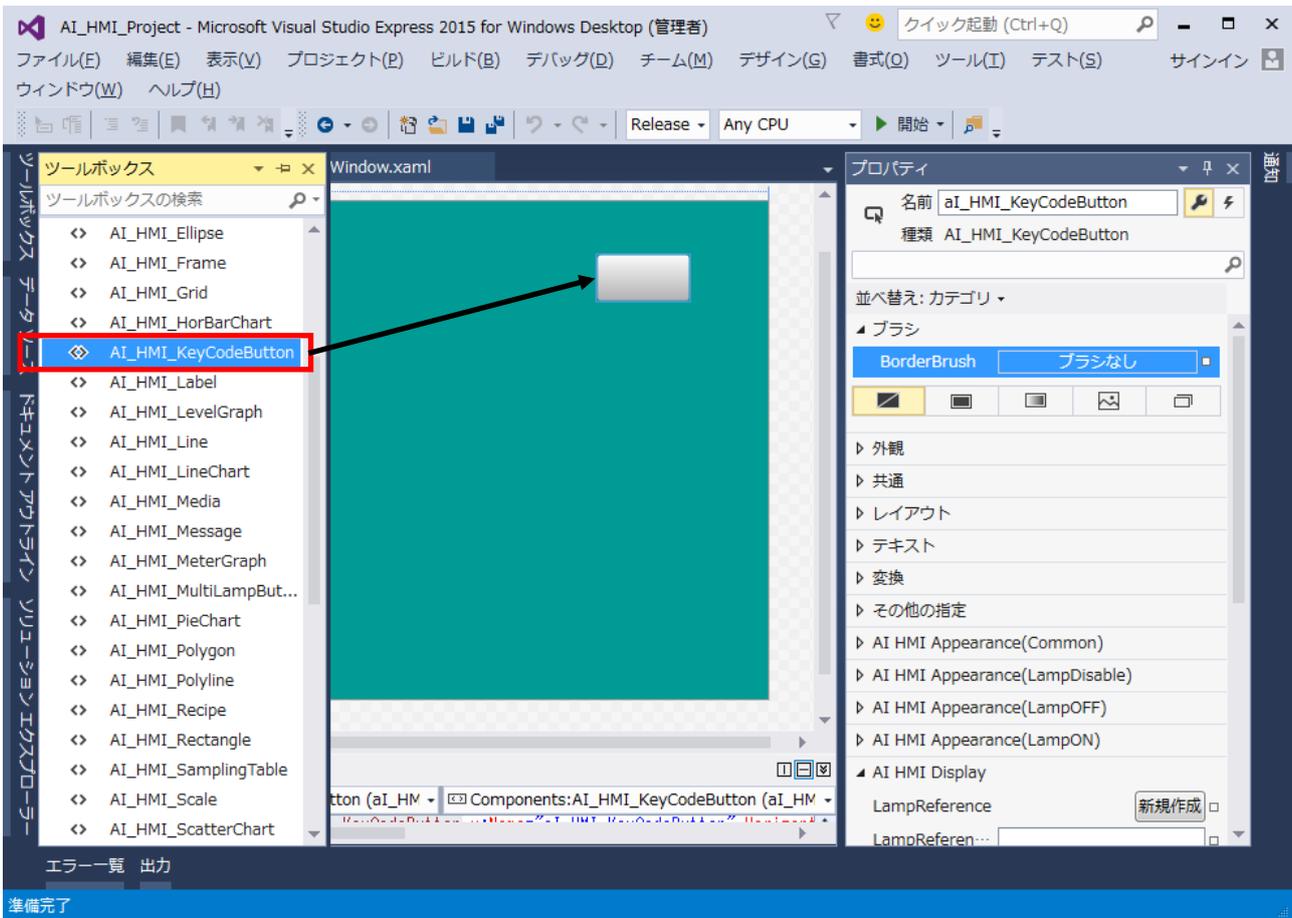


図 9-9-1 AI HMI KeyCodeButton

- ② プロパティから「AI HMI KeyCode」を選択してください。
AI HMI KeyCode プロパティが開きます。

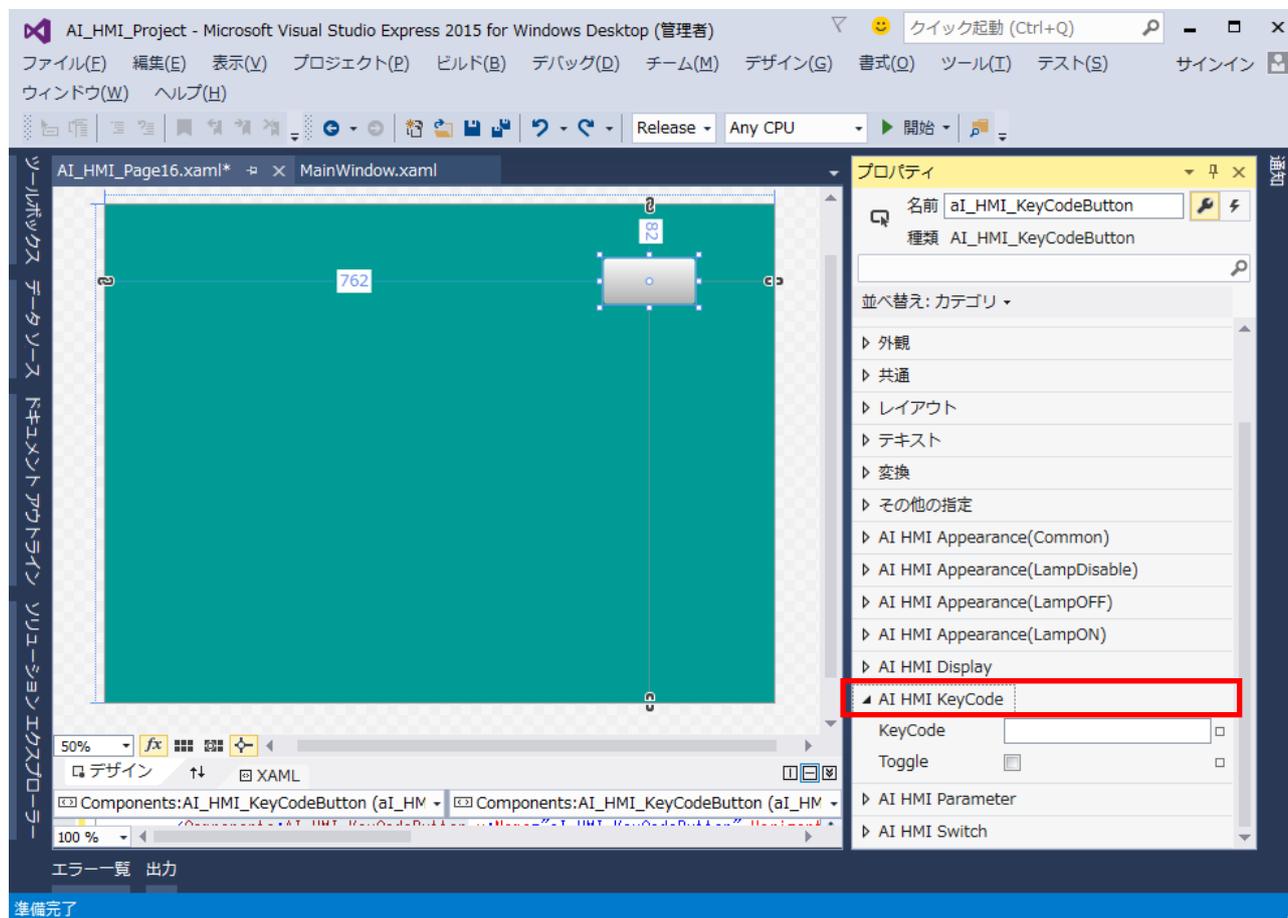


図 9-9-2 AI HMI KeyCode プロパティ

AI HMI KeyCode プロパティのそれぞれの意味は以下の通りです。

表 9-9-1 AI HMI KeyCode

項目	データ型	初期値	用途
KeyCode	String 型	初期値 なし	ボタンを押したときに送信するキーコードを 16 進数表記で設定します。 格納されるデータは String 型です。 キーコードは AI HMI 上の現在フォーカスのあるオブジェクトに送信します。
Toggle	チェック ボックス	チェック なし	ボタンをトグルスイッチとして使うかどうかを設定します。 トグルスイッチとして扱うボタンは一度押すとそのボタンが押されっぱなしの状態になります。 もう一度同じボタンを押すと押されっぱなし状態は解除されます。

- ③ 実際にボタンを押したときにソフトキーとして動作するように設定します。
AI HMI KeyCodeButton を 6 個配置してそれぞれのプロパティを以下のように設定してください。

表 9-9-2 AI HMI KeyCodeButton1 プロパティ

項目		設定値	備考
AI HMI Appearance (LampDisable)	Content1	A	ボタンに表示する文字列の設定です。 ボタン上に A と表示します。
AI HMI KeyCode	KeyCode	41	ボタンを押したときに「A」のキーコードを送信します。
	Toggle	チェックなし	

表 9-9-3 AI HMI KeyCodeButton2 プロパティ

項目		設定値	備考
AI HMI Appearance (LampDisable)	Content1	L	ボタンに表示する文字列の設定です。 ボタン上に L と表示します。
AI HMI KeyCode	KeyCode	4C	ボタンを押したときに「L」のキーコードを送信します。
	Toggle	チェックなし	

表 9-9-4 AI HMI KeyCodeButton3 プロパティ

項目		設定値	備考
AI HMI Appearance (LampDisable)	Content1	G	ボタンに表示する文字列の設定です。 ボタン上に G と表示します。
AI HMI KeyCode	KeyCode	47	ボタンを押したときに「G」のキーコードを送信します。
	Toggle	チェックなし	

表 9-9-5 AI HMI KeyCodeButton4 プロパティ

項目		設定値	備考
AI HMI Appearance (LampDisable)	Content1	0	ボタンに表示する文字列の設定です。 ボタン上に 0 と表示します。
AI HMI KeyCode	KeyCode	4F	ボタンを押したときに「0」のキーコードを送信します。
	Toggle	チェックなし	

表 9-9-6 AI HMI KeyCodeButton5 プロパティ

項目		設定値	備考
AI HMI Appearance (LampDisable)	Content1	SHIFT	ボタンに表示する文字列の設定です。 ボタン上に SHIFT と表示します。
AI HMI KeyCode	KeyCode	10	ボタンを押したときに「SHIFT」のキーコードを送信し続けるようにします。
	Toggle	チェックあり	

表 9-9-7 AI HMI KeyCodeButton6 プロパティ

項目		設定値	備考
AI HMI Appearance (LampDisable)	Content1	ENTER	ボタンに表示する文字列の設定です。 ボタン上に ENTER と表示します。
AI HMI KeyCode	KeyCode	D	ボタンを押したときに「ENTER」のキーコードを送信します。
	Toggle	チェックなし	

- ④ キーコードの送信先テキストボックスを配置します。
ツールボックスから AI HMI TextBox を AI HMI の画面に配置してください。

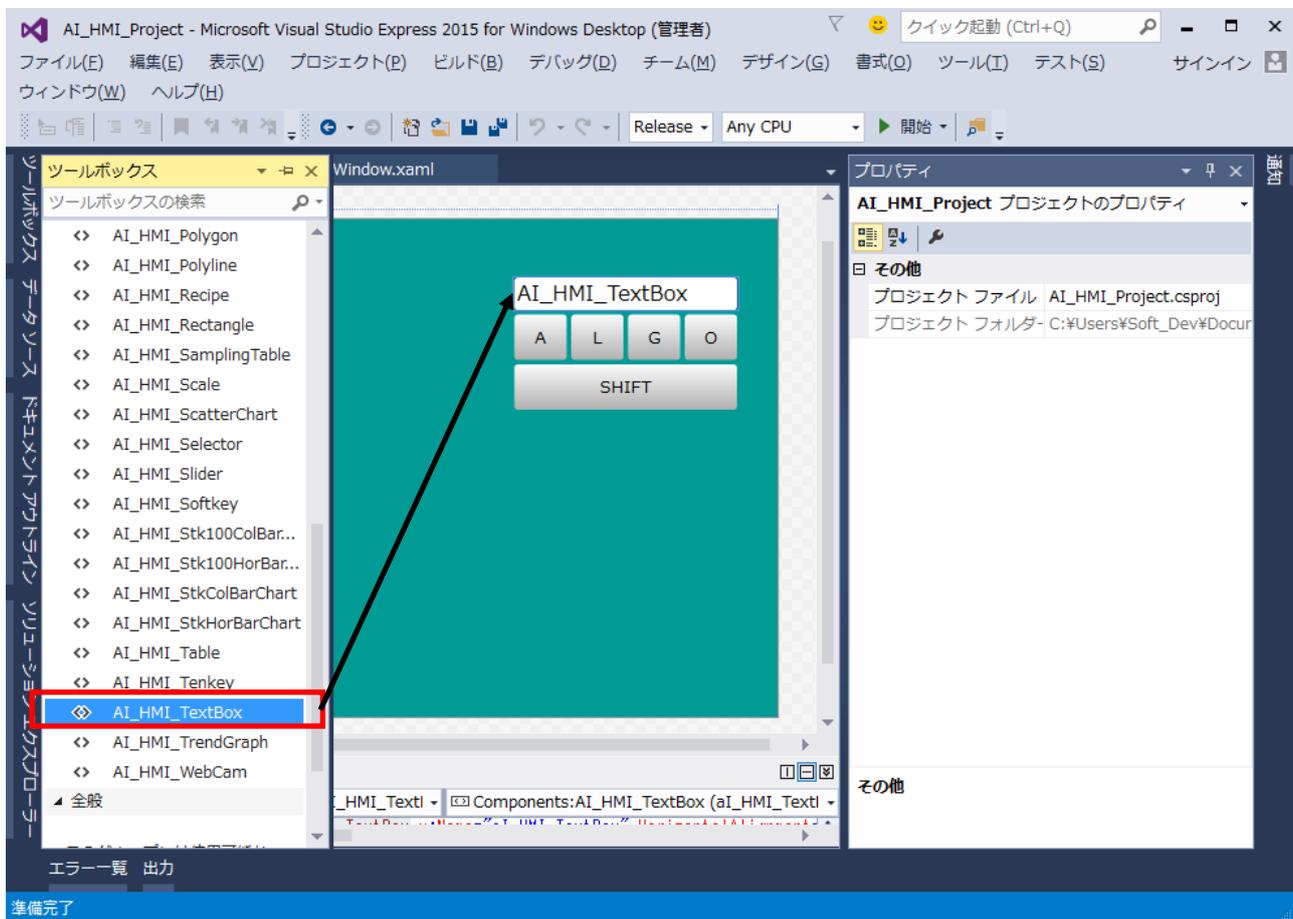


図 9-9-3 AI HMI TextBox

- ⑤ プロパティから「AI HMI Display」を選択してください。
AI HMI Display プロパティが開きます。

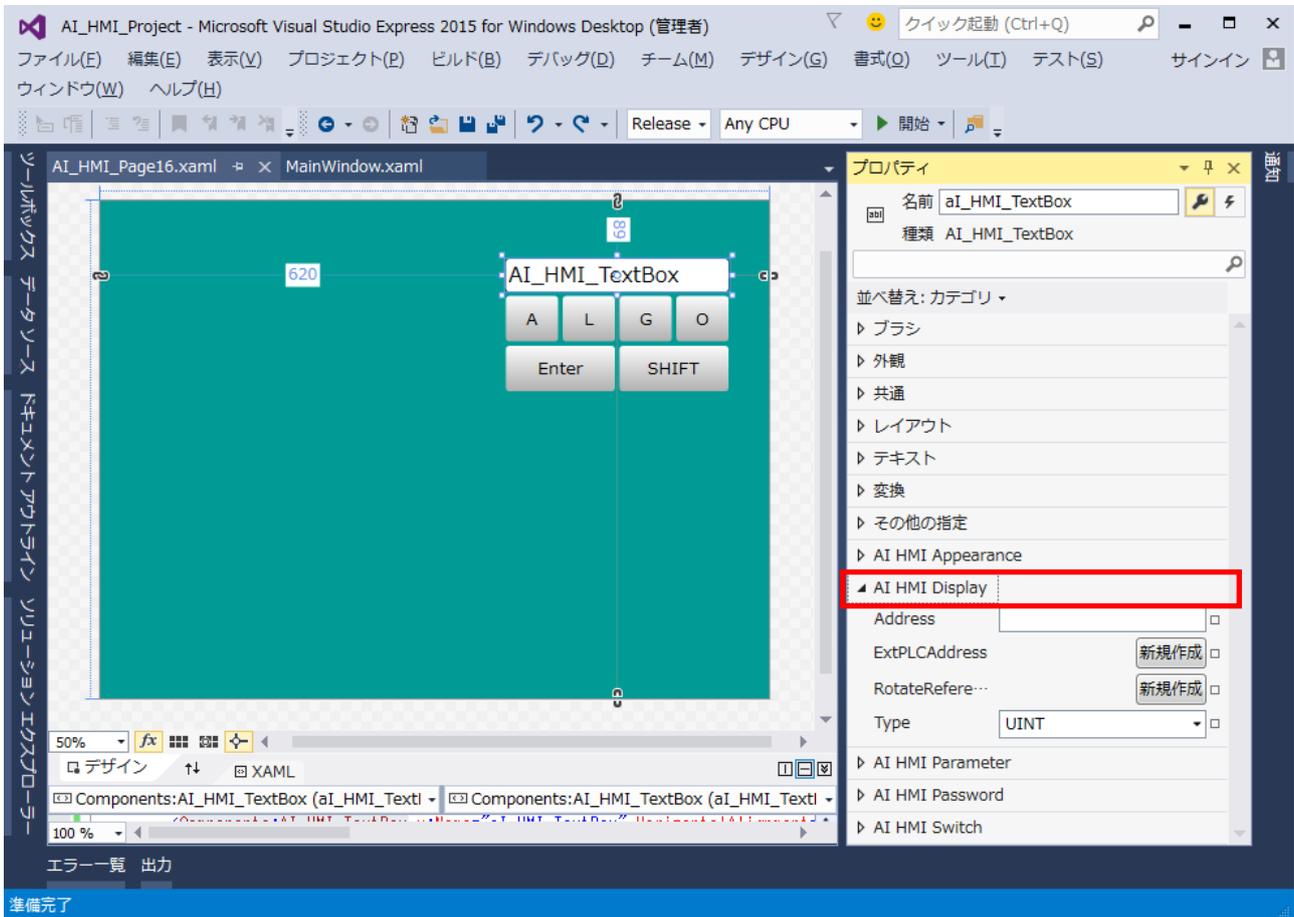


図 9-9-4 AI HMI Display プロパティ

AI HMI Display プロパティのそれぞれの意味は以下の通りです。

表 9-9-8 AI HMI Display

項目	データ型	初期値	用途
Address	String 型	初期値 なし	AI HMI TextBox に対して「Enter」のキーコードが送信されたとき、現在の AI HMI TextBox 上に現在表示している文字列を格納する先のアドレスを設定します。
Type	プルダウン メニュー	UINT	AI HMI TextBox に対して「Enter」のキーコードが送信されたとき、現在の AI HMI TextBox 上に現在表示している文字列を格納するデータ型を設定します。

- ⑥ 実際に文字列が入力されて Enter キーが押されたときに AI HMI TextBox 上の文字列を指定したアドレスに格納するように設定します。

AI HMI KeyCodeButton を 6 個配置してそれぞれのプロパティを以下のように設定してください。

表 9-9-9 AI HMI TextBox プロパティ

項目	設定値	備考
AI HMI Display	Address	145000
	Type	STRING

入力された文字列を共有メモリのアドレス 145000 に STRING 型で格納するように設定します。

- ⑦ 設定したソフトウェアキーボードの動作を確認する画面を作成します。

図 9-9-5 のように AI HMI Label を配置してください。

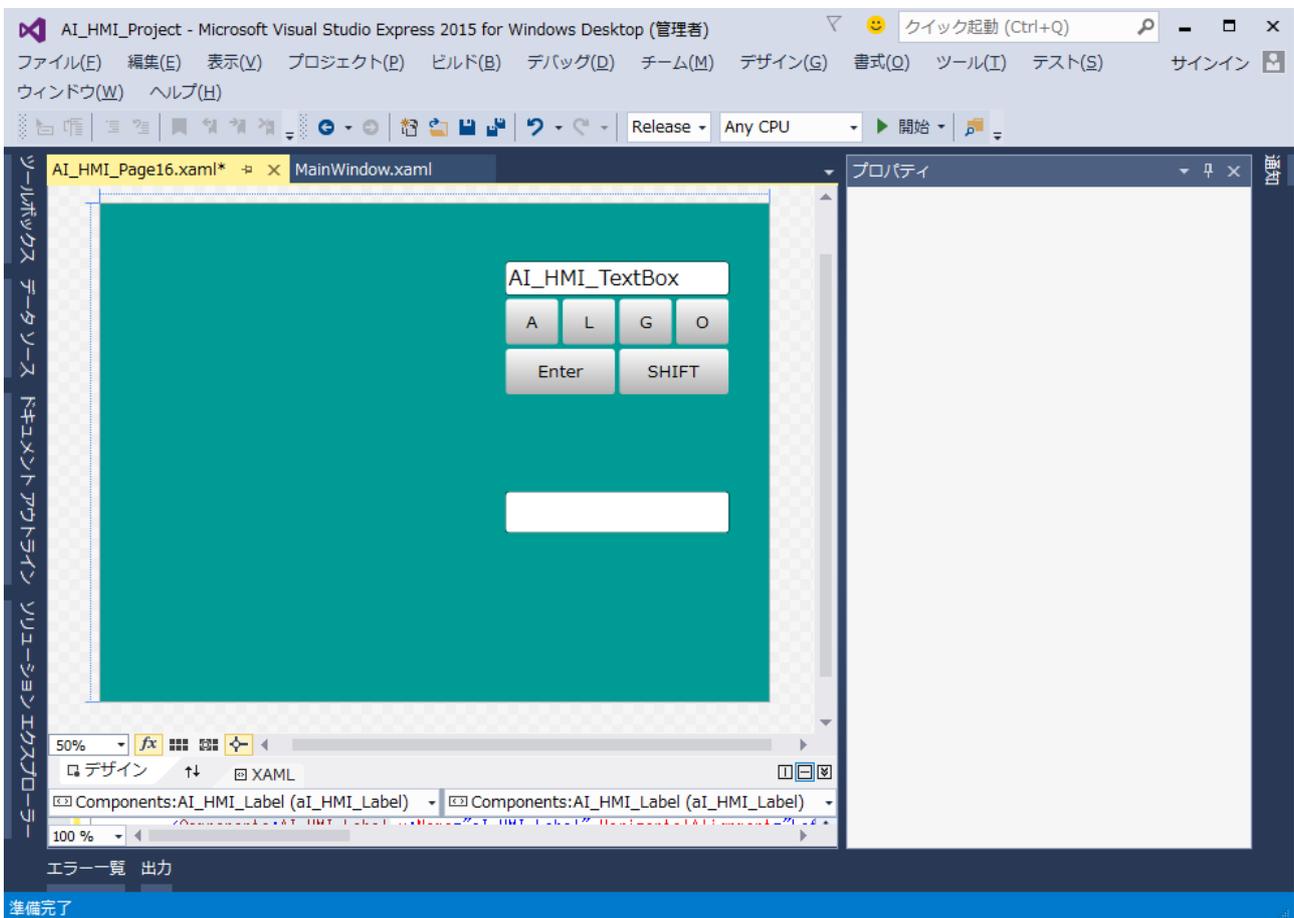


図 9-9-5 ソフトウェアキーボード確認用画面の作成

配置した AI HMI Label のプロパティは以下のように設定してください。



図 9-9-6 ソフトウェアキーボード確認用画面の詳細

表 9-9-10 AI HMI Label プロパティ

項目	設定値	備考
AI HMI Display	Address	145000
	LabeyType	STRING
		共有メモリのアドレス 145000 に格納されているデータを STRING 型で表示するように設定します。

- ⑧ 作成した画面を実際に動作させてソフトウェアキーボードが動作することを確認します。
『第6章 作成したアプリの実行環境PCでの動作』を参照し、作成した画面アプリを動作させてください。

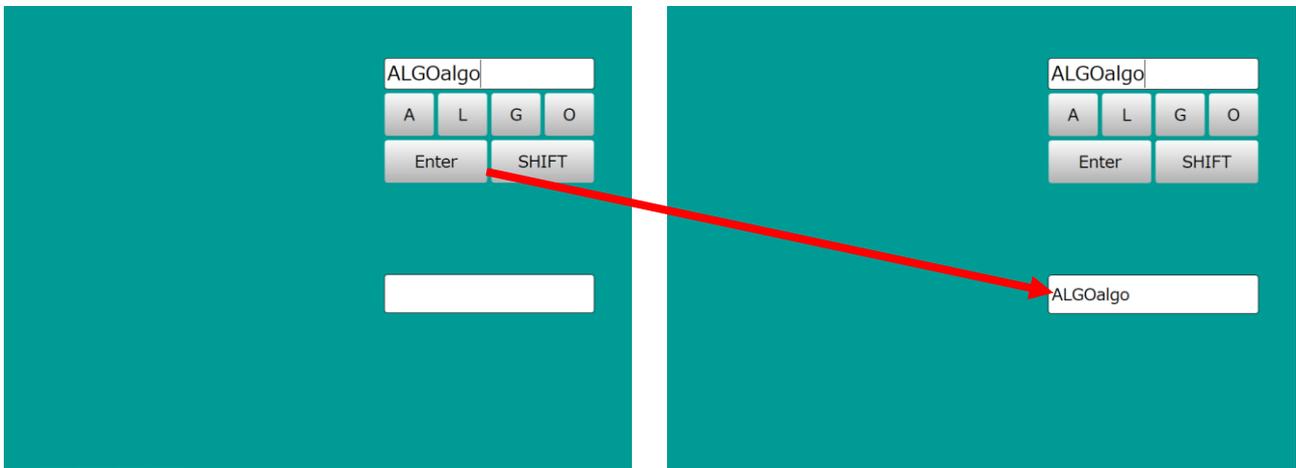


図 9-9-7 ソフトウェアキーボード確認画面の実行

テキストボックスを選択した状態でキーボタンを押したときにテキストボックスに文字列が入力されることを確認してください。
また、Enter ボタンを押したときにテキストボックス上の文字列がラベルに表示されることを確認してください。

9-10 タッチ時の Beep 音機能

特定の部品をタッチしたときに Beep 音やサウンドファイルを鳴らす機能です。

- ① MainWindow.xaml を開き、「ドキュメントアウトライン - [AI_HMI_Window]」を選択してください。
プロパティ欄に AI HMI Window のプロパティが表示されます。

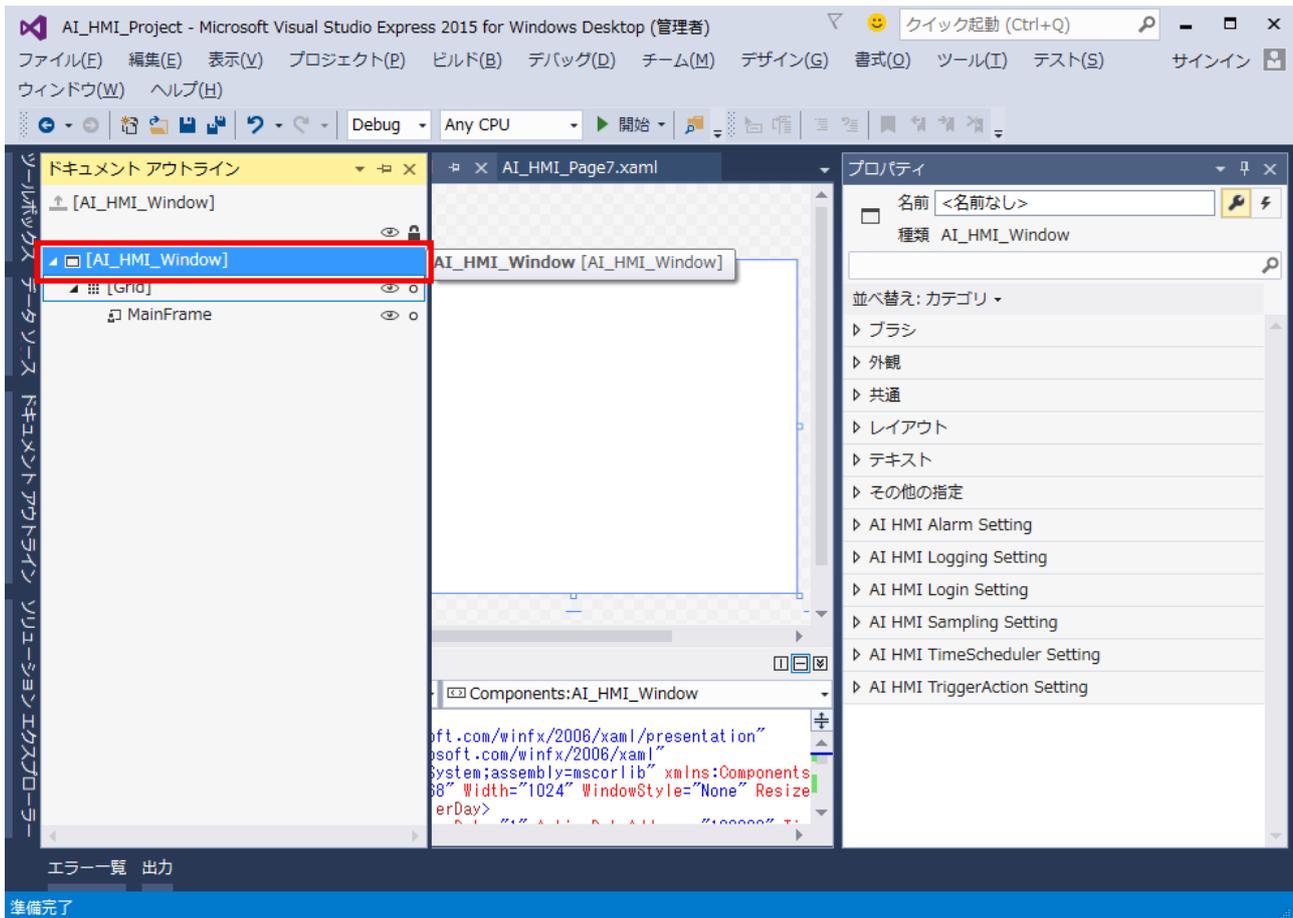


図 9-10-1 AI HMI Window プロパティ

- ② プロパティから「AI HMI Beep Setting」を選択してください。
AI HMI Beep Setting プロパティが開きます。

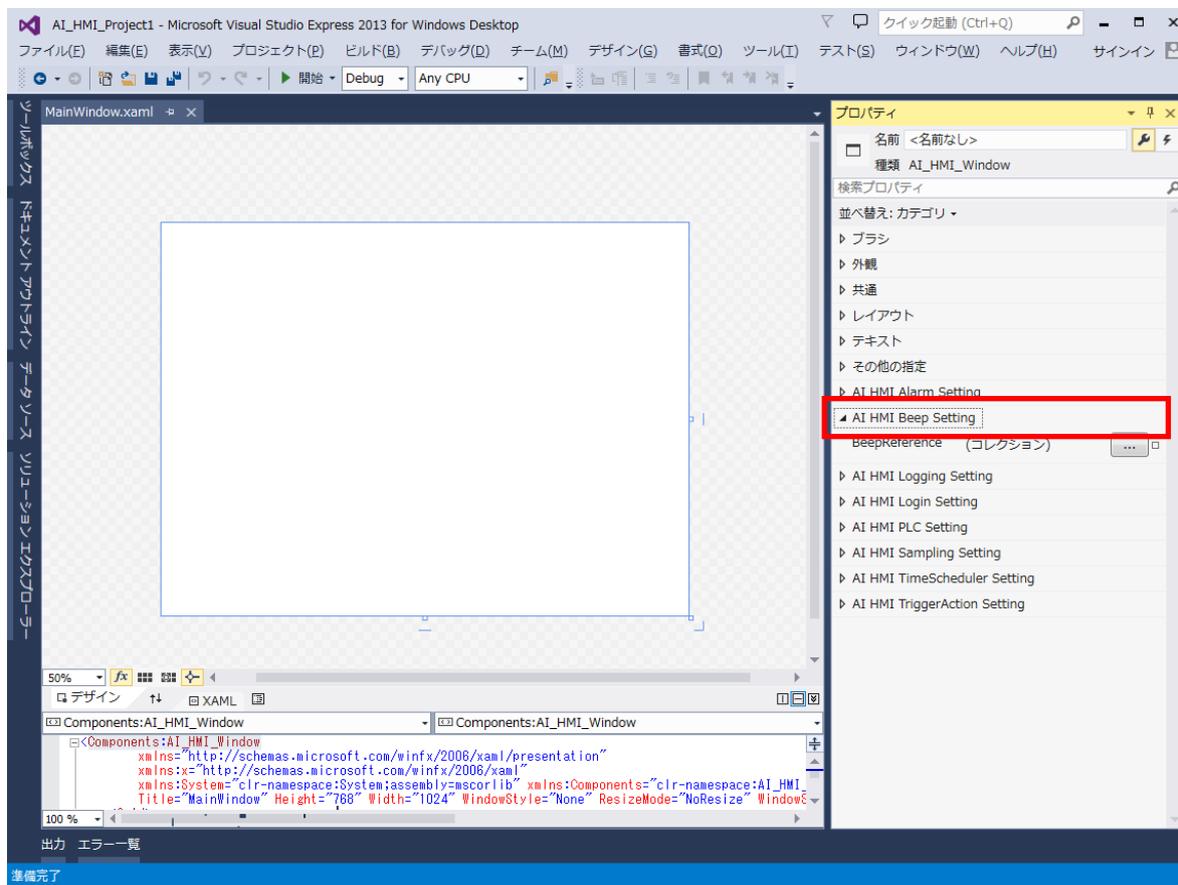


図 9-10-2 AI HMI Beep Setting プロパティ

AI HMI Beep Setting プロパティのそれぞれの意味は以下の通りです。

表 9-10-1 AI HMI Beep Setting

項目	内容
BeepReference	タッチしたときに鳴らす Beep 音やサウンドファイルの設定を行います。 Beep 音やサウンドは最大 3 種類まで登録することができます。

BeepReference では以下のような設定が可能です。

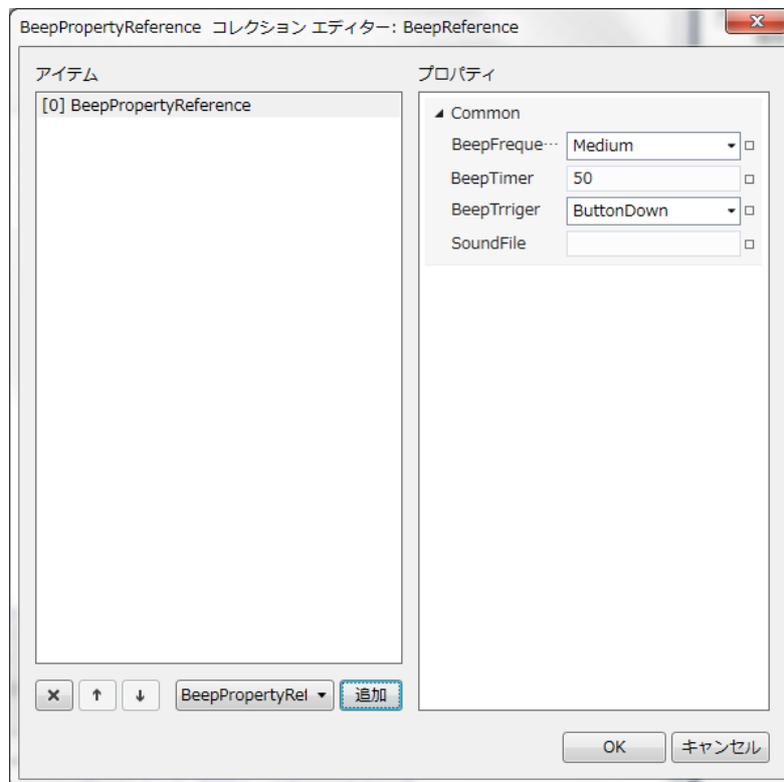


図 9-10-3 BeepReference のプロパティ

表 9-10-2 BeepReference のプロパティ詳細

項目	データ型	初期値	内容
BeepFrequency	プルダウン メニュー	Medium	タッチしたときに鳴らす Beep 音の周波数を設定します。
BeepTimer	UINT 型	50	タッチしたときに Beep 音を鳴らす時間を設定します。
BeepTrigger	プルダウン メニュー	ButtonDown	タッチしたときに Beep 音を鳴らすタイミングを設定します。
SoundFile	String 型	初期値 なし	タッチしたときに鳴らすサウンドファイルを設定します。 この項目に入力がある場合、Beep 音は鳴らしません。

※ BeepReference では複数の BeepPropertyReference を登録できますが、実際に使用ができるのは3件までです。

- ③ 実際にボタンをタッチした際に Beep 音やサウンドファイルを鳴らすように設定します。BeepReference のコレクションエディタ画面上で「追加」ボタンを2回押して BeepPropertyReference を2行追加し、それぞれ以下のように設定してください。

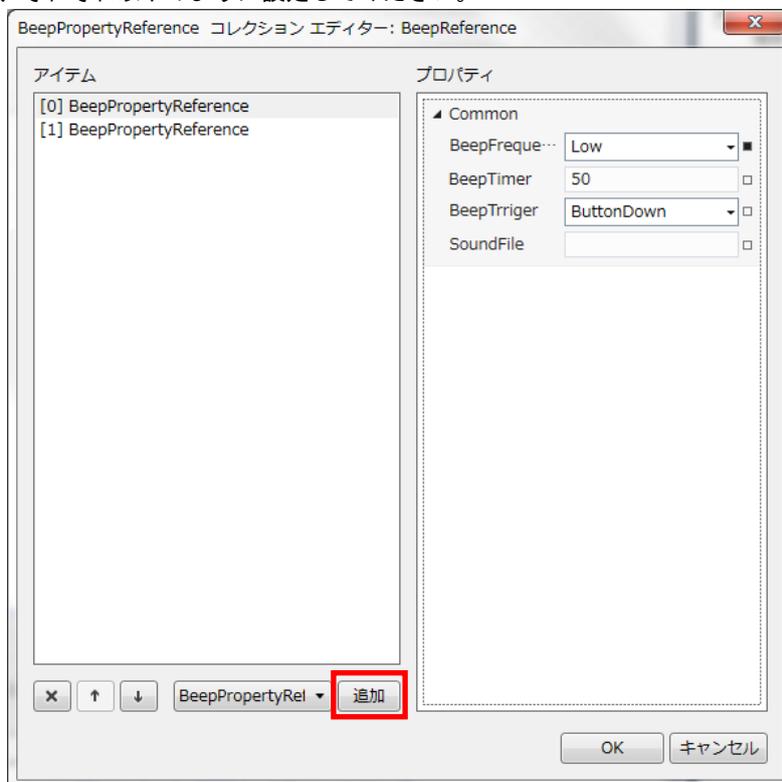


図 9-10-4 BeepReference のプロパティ

表 9-10-3 [0]BeepPropertyReference 設定内容

項目	設定内容	備考
BeepFrequency	Low	Beep 音の周波数を 1000Hz に設定します。
BeepTimer	50	Beep 音を鳴らす時間を 50msec に設定します。
BeepTrigger	ButtonDown	ボタンをタッチしたタイミングで Beep 音を鳴らすように設定します。
SoundFile	空白	サウンドファイルを再生しません。

表 9-10-4 [1] BeepPropertyReference 設定内容

項目	設定内容	備考
BeepFrequency	Medium	サウンドファイルを設定するため、この項目の設定は無効になります。
BeepTimer	50	
BeepTrigger	ButtonUp	ボタンをタッチして離れたタイミングでサウンドファイルを鳴らすように設定します。
SoundFile	.¥xxx.wav	再生するサウンドファイルを設定します。※

※ サウンドファイルは実行ファイルと同じフォルダ内に用意し、SoundFile と同じファイル名にしてください。

- ④ Beep 機能の動作を確認するために AI HMI Button を配置します。
 図 9-10-5 のように AI HMI Button を配置してください。

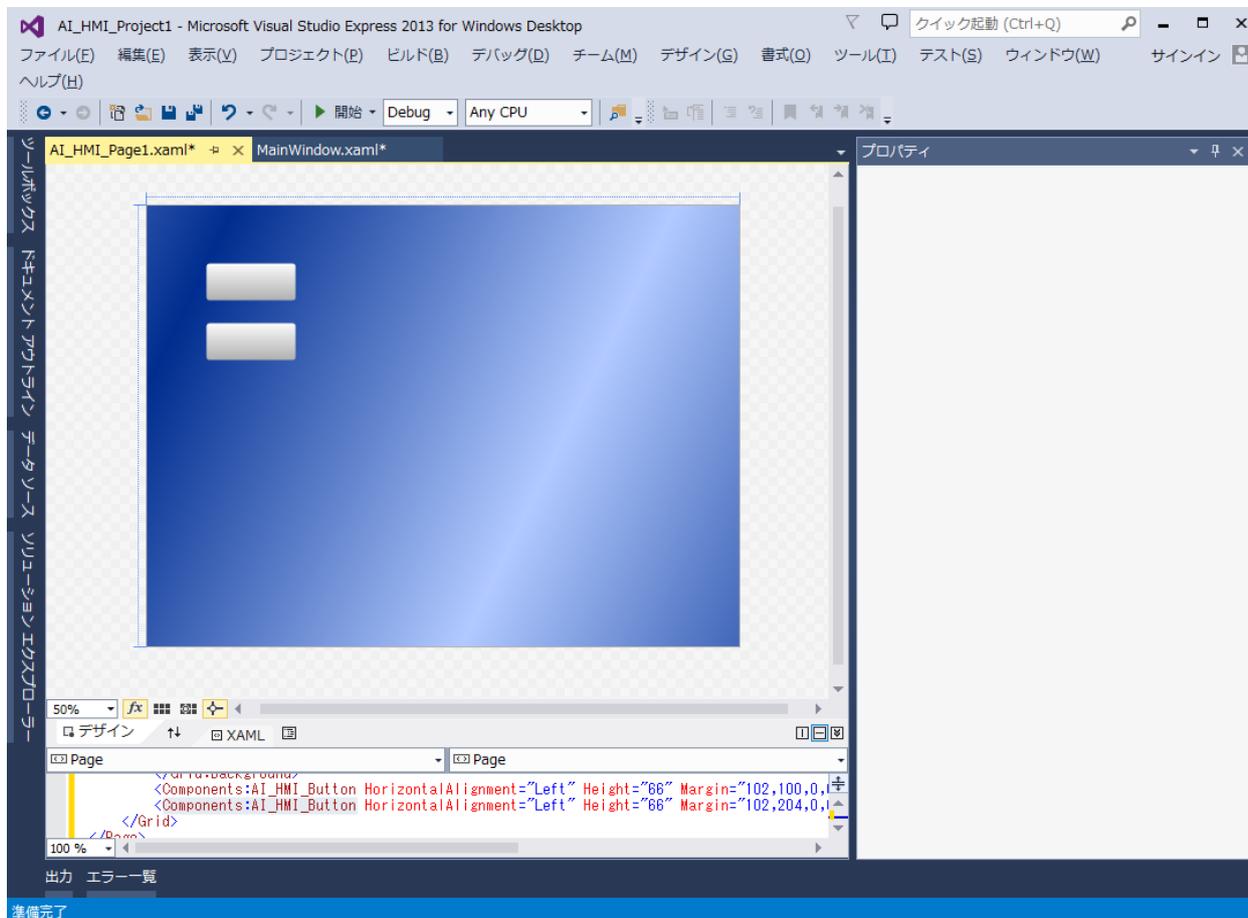


図 9-10-5 動作確認用 AI HMI Button

それぞれの AI HMI Button は以下のように設定してください。

表 9-10-5 Beep1 鳴動確認用ボタン

項目	設定値	備考
AI HMI Beep	BeepEnable	Beep1 を鳴らすように設定します。

表 9-10-6 Beep2 鳴動確認用ボタン

項目	設定値	備考
AI HMI Beep	BeepEnable	Beep2 を鳴らすように設定します。

- ⑤ 作成した画面を実際に動作させて Beep 音機能が動作することを確認します。
『第6章 作成したアプリの実行環境PCでの動作』を参照し、作成した画面アプリを動作させてください。

ボタン1をタッチすると、タッチした瞬間に Beep 音が鳴ることを確認してください。

また、ボタン2をタッチすると、タッチを離れた瞬間に設定したサウンドファイルが再生されることを確認してください。

※ Beep 音設定は以下のコンポーネントでも設定が可能です。

- ・ AI HMI Button
- ・ AI HMI MultiLampButton
- ・ AI HMI KeyCodeButton
- ・ AI HMI Label
- ・ AI HMI TextBox
- ・ AI HMI Tenkey
- ・ AI HMI Sound
- ・ AI HMI Grid
- ・ AI HMI Recipe
- ・ AI HMI Slider
- ・ AI HMI Selector
- ・ AI HMI Position

9-11 外部アプリケーションの実行

AI-HMI ではボタン操作で外部アプリケーションを実行することができます。
実行するファイルを指定することで、そのファイルに関連付けされているアプリケーションで実行します。
また、複数の引数を設定することも可能です。

- ① ツールボックスから AI HMI Button を AI HMI の画面に配置してください。

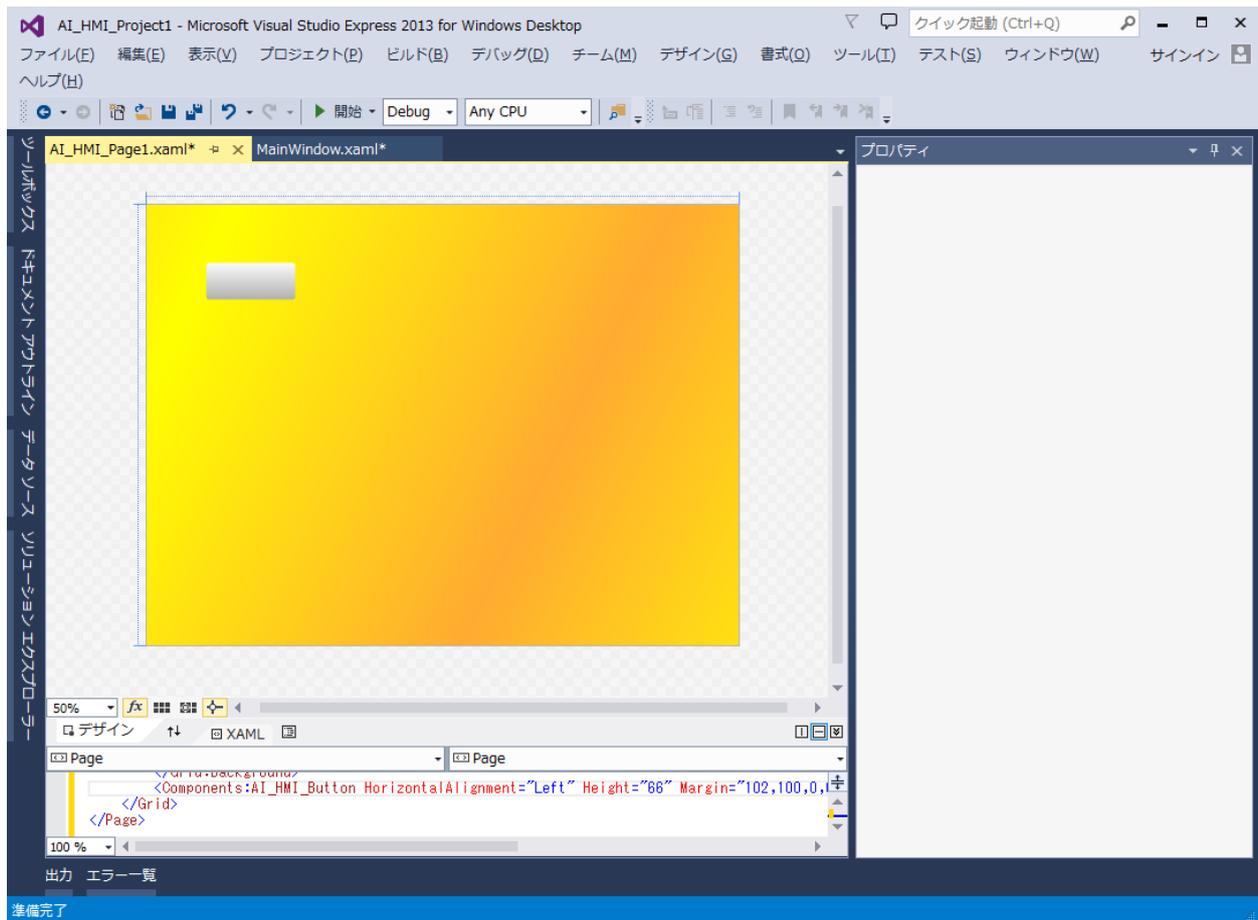


図 9-11-1 AI HMI Button

- ② 配置された AI HMI Button のプロパティの AI HMI Switch – SwitchReference の右側にあるボタンを押してください。

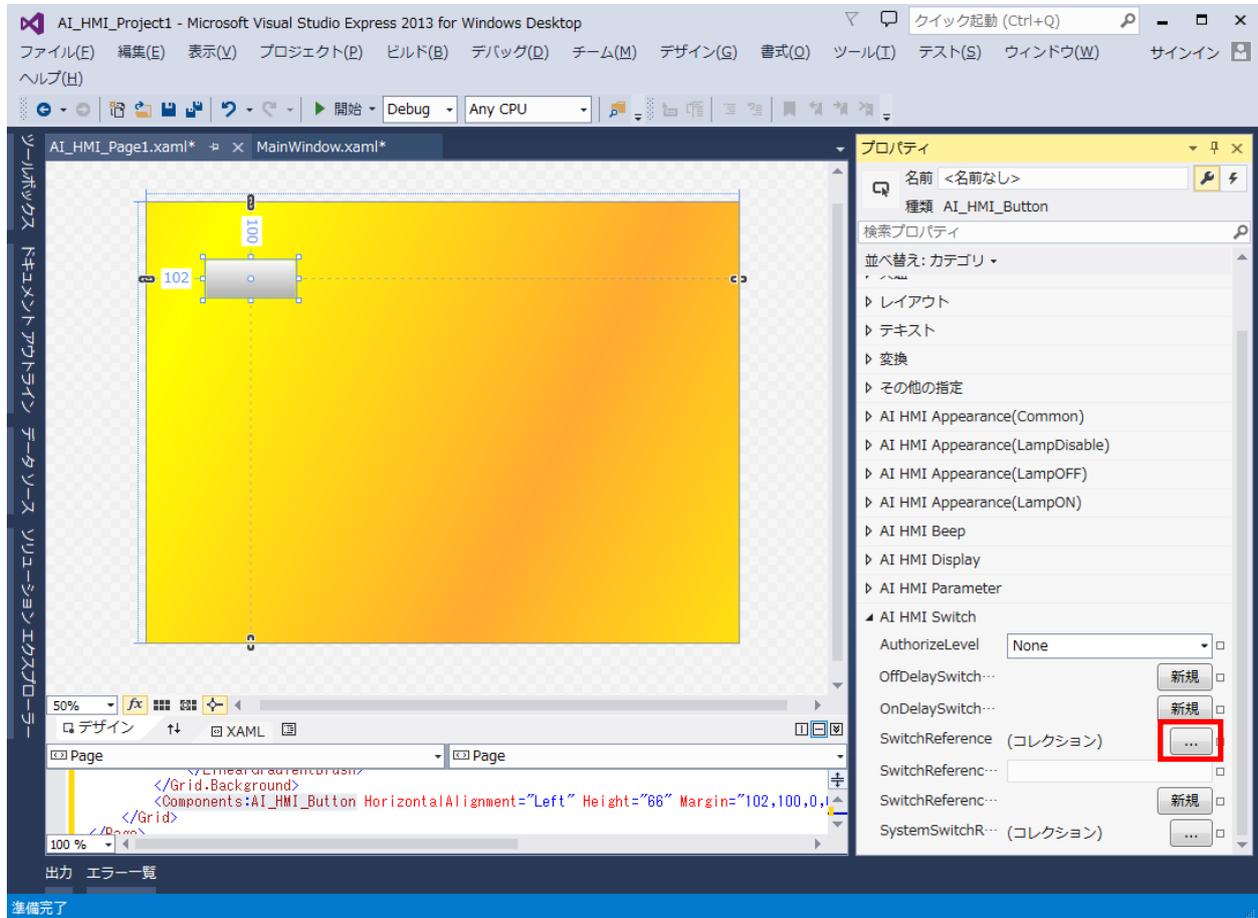


図 9-11-2 AI HMI Button の設定

- ③ ButtonSwitchReference のコレクションエディターが開きます。
「追加」ボタンを1回押し、ButtonSwitchReference を1行追加してください。

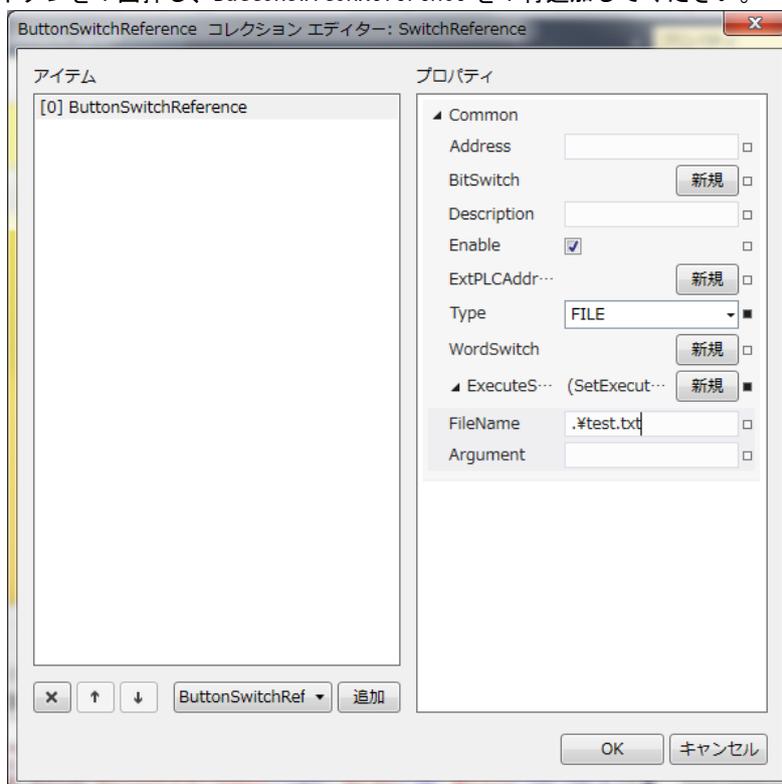


図 9-11-3 ButtonSwitchReference

ButtonSwitchReference は以下のように設定してください。

表 9-11-1 [0]ButtonSwitchReference 設定内容

項目	設定内容	備考
Type	FILE	ボタンを押したときに外部アプリを実行するように設定します。
FileName	.#xxx.txt	ExecuteSwitch の項目で「新規作成」ボタンを押すと項目が表示されます。※
Argument	(空白)	

※ アプリケーションと同じフォルダ内に確認用のテキストファイルを作成し、FileName の設定と同じファイル名にしてください。

- ④ 作成した画面を実際に動作させて外部アプリ実行機能が動作することを確認します。
『第6章 作成したアプリの実行環境PCでの動作』を参照し、作成した画面アプリを動作させてください。

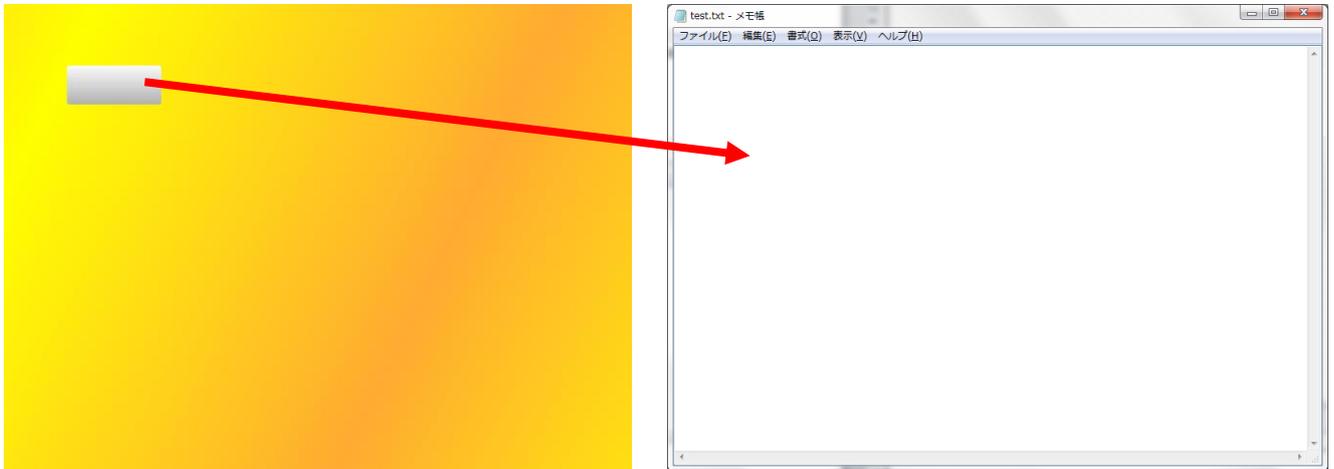


図 9-11-4 外部アプリ実行確認画面の実行

ボタンを押したときにあらかじめ作成していたテキストファイルが開くことを確認してください。

9-12 コンポーネントの位置変更

特定の部品の位置を変更することができる機能です。
この機能を使用してダイアログの位置を変更することも可能です。

- ① 図9-12-1のようにAI_HMI_Gridを配置し、その上にAI_HMI_Positionを配置してください。

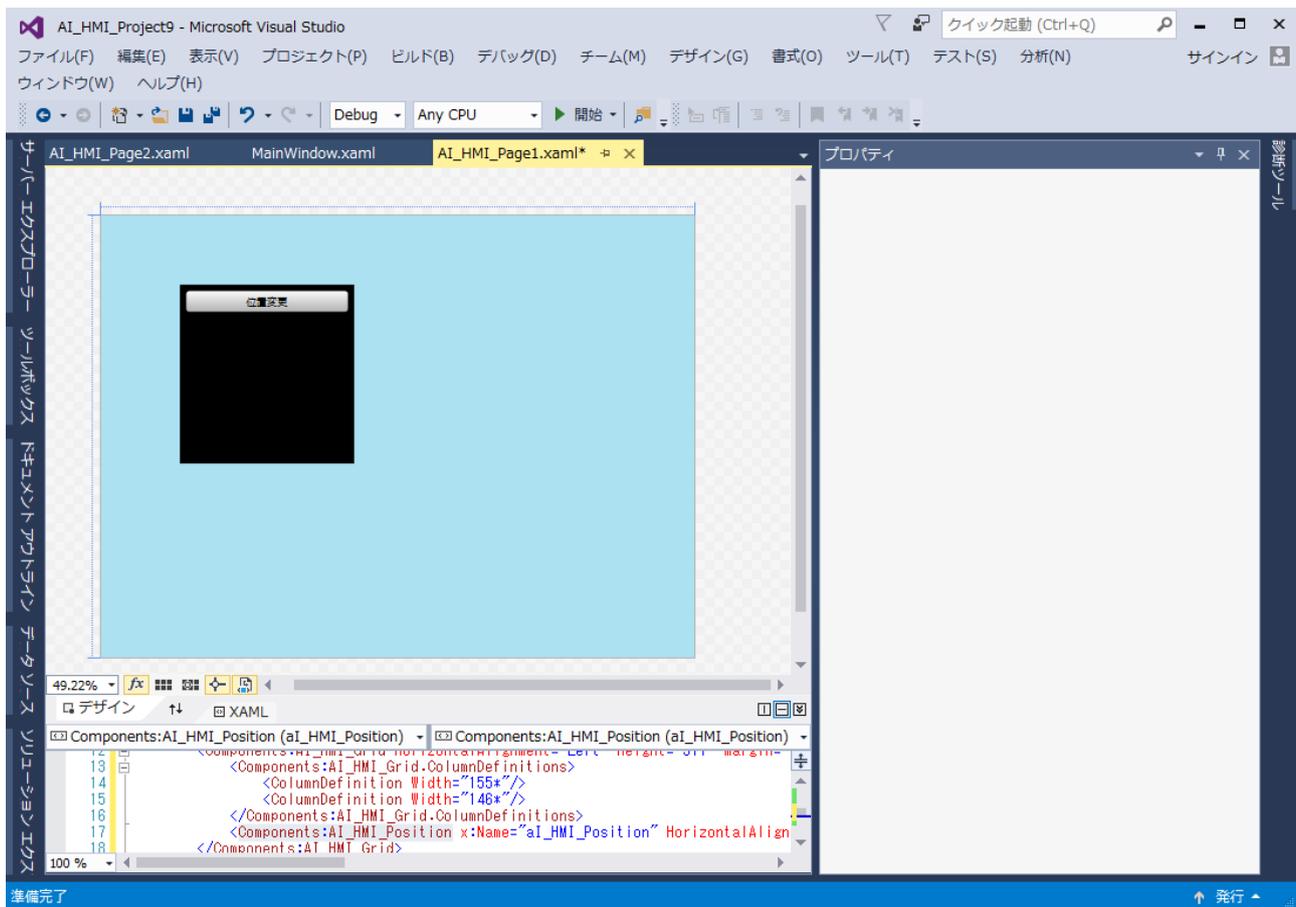


図 9-12-1 コントロール位置変更画面の作成

それぞれの部品のプロパティは以下のように設定してください。



図 9-12-2 コントロール位置変更画面詳細

1. 位置変更の対象となるコンポーネントです。ここでは AI HMI Grid を使用します。

表 9-12-1 位置変更対象コンポーネントプロパティ

項目			設定値	備考	
ブラシ	Background				
AI HMI Parameter	PLCReference	XPos	Address	110000	X 座標の値を参照するアドレスです。
		YPos	Address	110004	Y 座標の値を参照するアドレスです。

2. 座標位置を取得するボタンです。AI HMI Position を使用します。

表 9-12-2 座標位置取得ボタンプロパティ

項目			設定値	備考
AI HMI Position	XPosAddress	Address	110000	XPosAddress の項目で「新規作成」ボタンを押すことで項目が表示されます。 取得した X 座標の書き込み先アドレスです。
	YPosAddress	Address	110004	YPosAddress の項目で「新規作成」ボタンを押すことで項目が表示されます。 取得した Y 座標の書き込み先アドレスです。

- ② アプリケーション起動時に初期の座標位置を書き込むように設定します。
『9-5 スケジューラ機能』を参照し、SchedulerStartupのプロパティを表 9-12-3、表 9-12-4 のように設定してください。

表 9-12-3 [0]TimeSchedulerStartup の設定

項目	設定内容	備考
ActionData	100	共有メモリの 110000 に 100 を INT 型で書き込みます。
ActionDataAddress	110000	
ActionDataType	INT	
Description		
Enable	チェックあり	

表 9-12-4 [1]TimeSchedulerStartup の設定

項目	設定内容	備考
ActionData	100	共有メモリの 110004 に 100 を INT 型で書き込みます。
ActionDataAddress	110004	
ActionDataType	INT	
Description		
Enable	チェックあり	

- ③ 作成した画面を実際に動作させてコンポーネントの位置移動が動作することを確認します。
『第6章 作成したアプリの実行環境PCでの動作』を参照し、作成した画面アプリを動作させてください。

位置変更のボタンをクリックし、次に移動先の場所をクリックするとAI HMI Gridが移動することを確認してください。

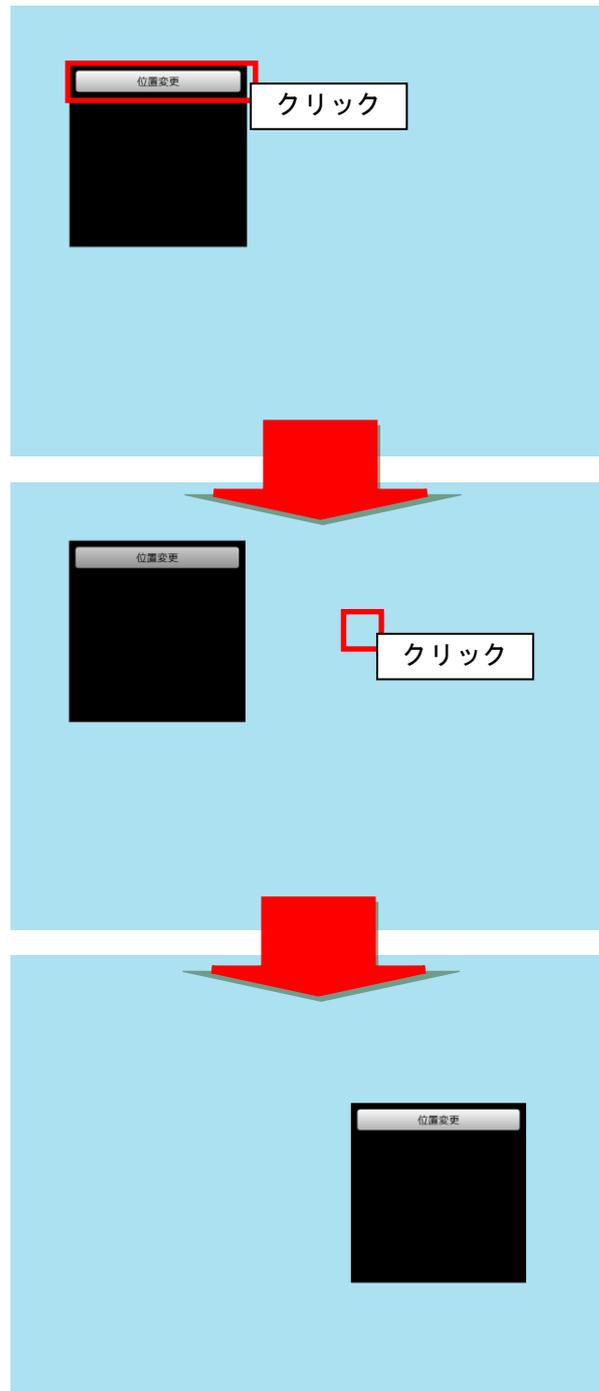


図 9-12-3 外部アプリ実行確認画面の実行

第 10 章 多言語切替機能

AI HMI Button などのコンポーネント上の表示文字列を言語切替する機能です。
本章ではボタン上の表示を日本語、英語、中国語の 3 言語で切り替える方法を解説します。

10-1 言語設定ファイルの作成

多言語切替機能を使用するために、最初に使用する言語の言語設定ファイルを作成する必要があります。
AI HMI ではデフォルトで日本語と英語の設定ファイル (Resources.resx と Resources.en-US.resx) が用意されているため、今回は中国語の設定ファイルを作成します。

- ① ソリューションエクスプローラーを開き、「(プロジェクト名) - Properties」をクリックしてください。
言語設定ファイルの一覧が表示されます。

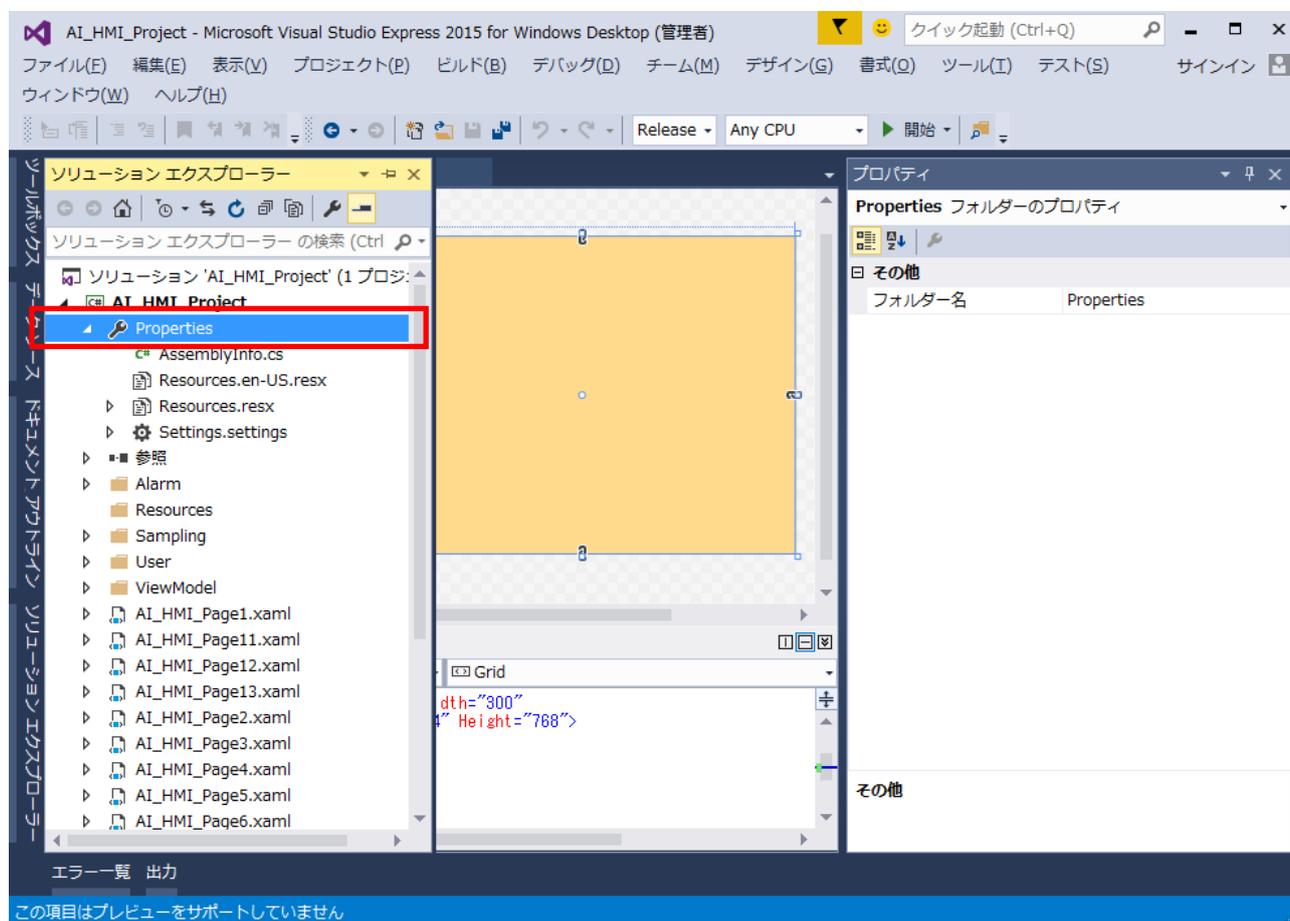


図 10-1-1 言語設定ファイル一覧

- ② Resources.en-US.resx を選択し、キーボードの「CTRL」キーと「C」キーを同時押ししてください。その後そのままキーボードの「CTRL」キーと「V」キーを同時押ししてください。「Resources.en-US - コピー.resx」という項目が新たに追加されるので、「Resources.zh-CN.resx」に名前を変更してください。

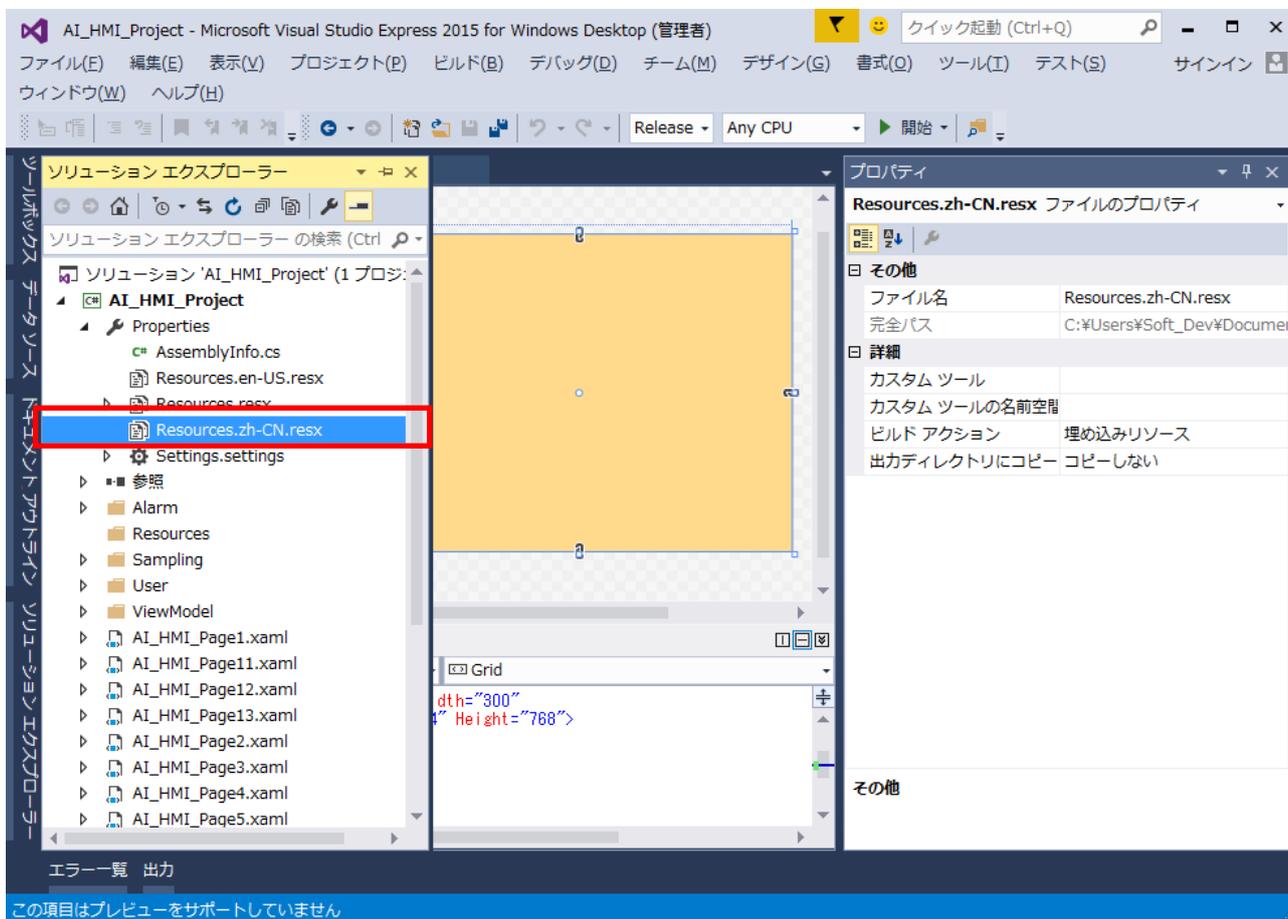


図 10-1-2 Resources.zh-CN.resx の作成

※ Resources.XXXX.resx の名前を変更するとき、XXXX 部分は正しい多言語コードを入力する必要があります。主な多言語コードの例は表 10-1-1 を参照してください。

表 10-1-1 主な多言語コード

項目	多言語コード	項目	多言語コード
英語 (アメリカ)	en-US	韓国語	ko
英語 (イギリス)	en-GB	ポルトガル語	pt
アラビア語	ar	ロシア語	ru
ドイツ語	de	中国語	zh
スペイン語	es	中国語 (簡体字)	zh-CN
フランス語	fr	中国語 (繁体字)	zh-TW
イタリア語	it		

これ以外の多言語コードは同梱の「多言語コード.txt」を参照してください。

- ③ 言語設定ファイルの編集を行います。
Resources.resx をダブルクリックしてください。
Resources.resx の言語表が開きます。

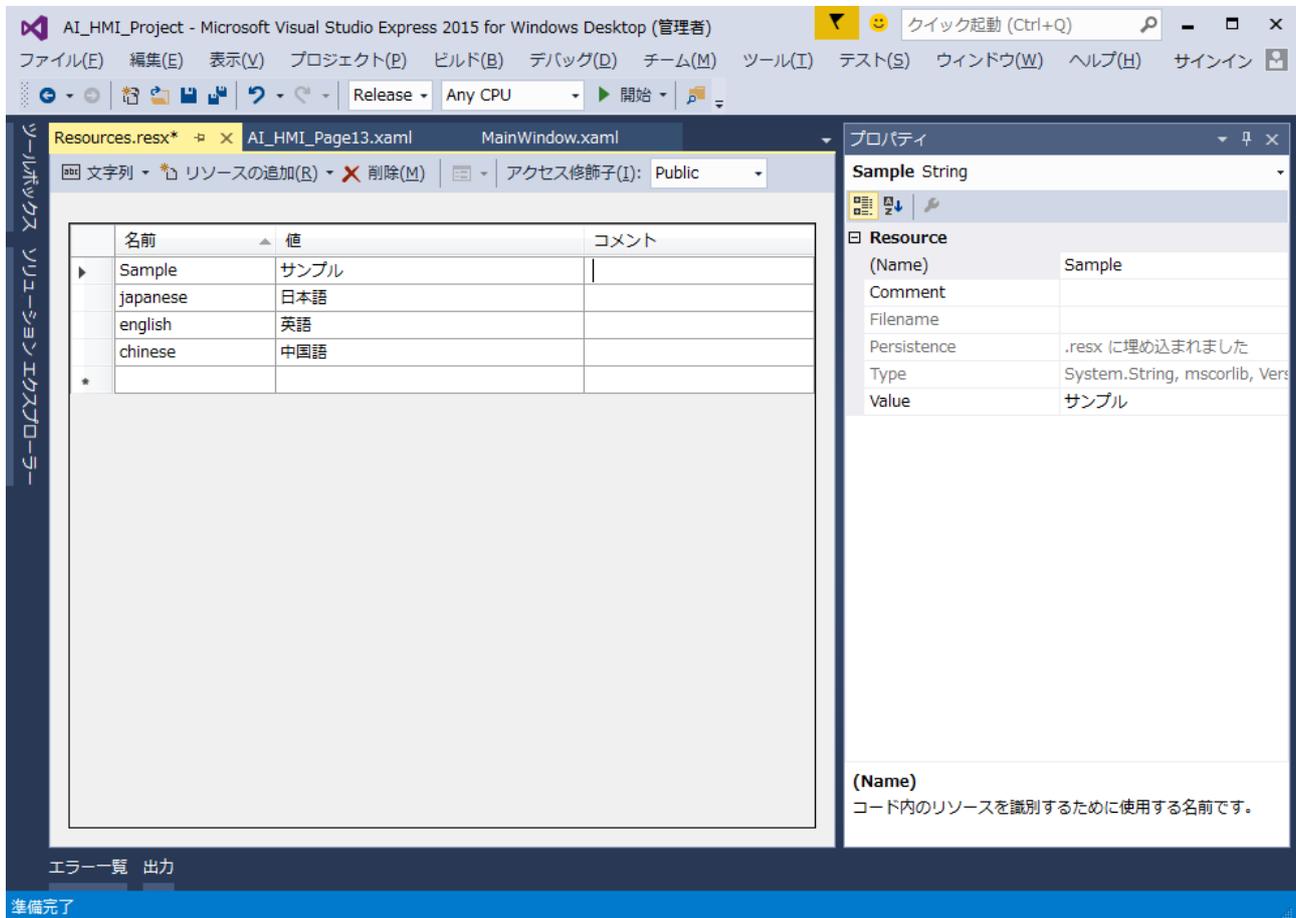


図 10-1-3 Resources.resx の編集

表内の名前と値の項目を以下のように追加してください。

表 10-1-2 Resources.resx の編集内容

名前	値
japanese	日本語
english	英語
chinese	中国語

同様にして Resources.en-US.resx および Resources.zh-CN.resx の表内の名前と値の項目を以下のように追加してください。

表 10-1-3 Resources.en-US.resx の編集内容

名前	値
japanese	Japanese
english	English
chinese	Chinese

表 10-1-4 Resources.zh-CN.resx の編集内容

名前	値
japanese	日文
english	英語
chinese	中文

10-2 言語切替画面の作成

作成した多言語設定ファイルを元に言語切替をすることができる画面を作成します。

- ① AI HMI Button コンポーネントを図 10-2-1 のように配置してください。

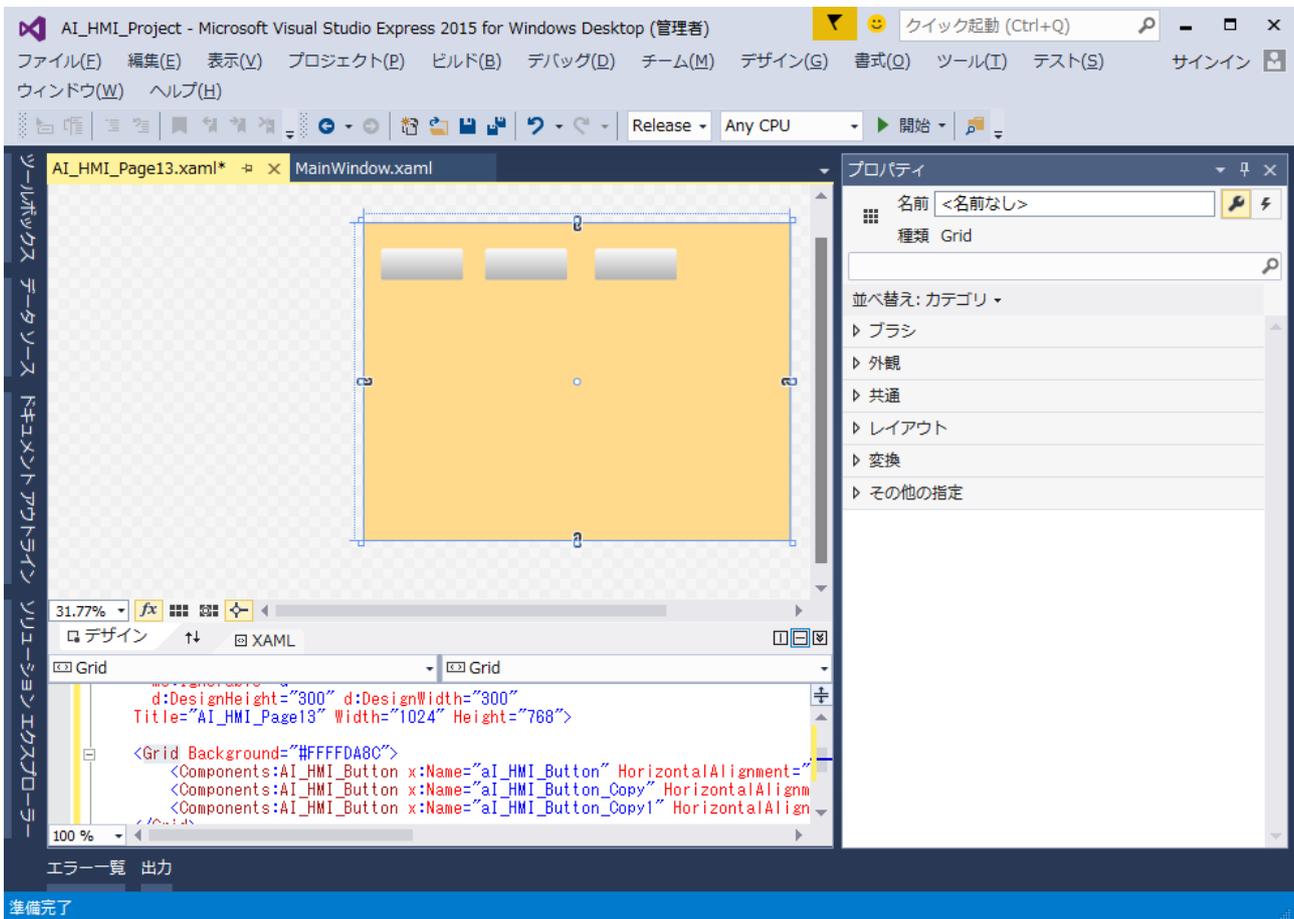


図 10-2-1 言語切替画面の作成

- ② AI HMI Button を選択し、プロパティの「AI HMI Text – Content1」の右側にある口をクリックしてください。

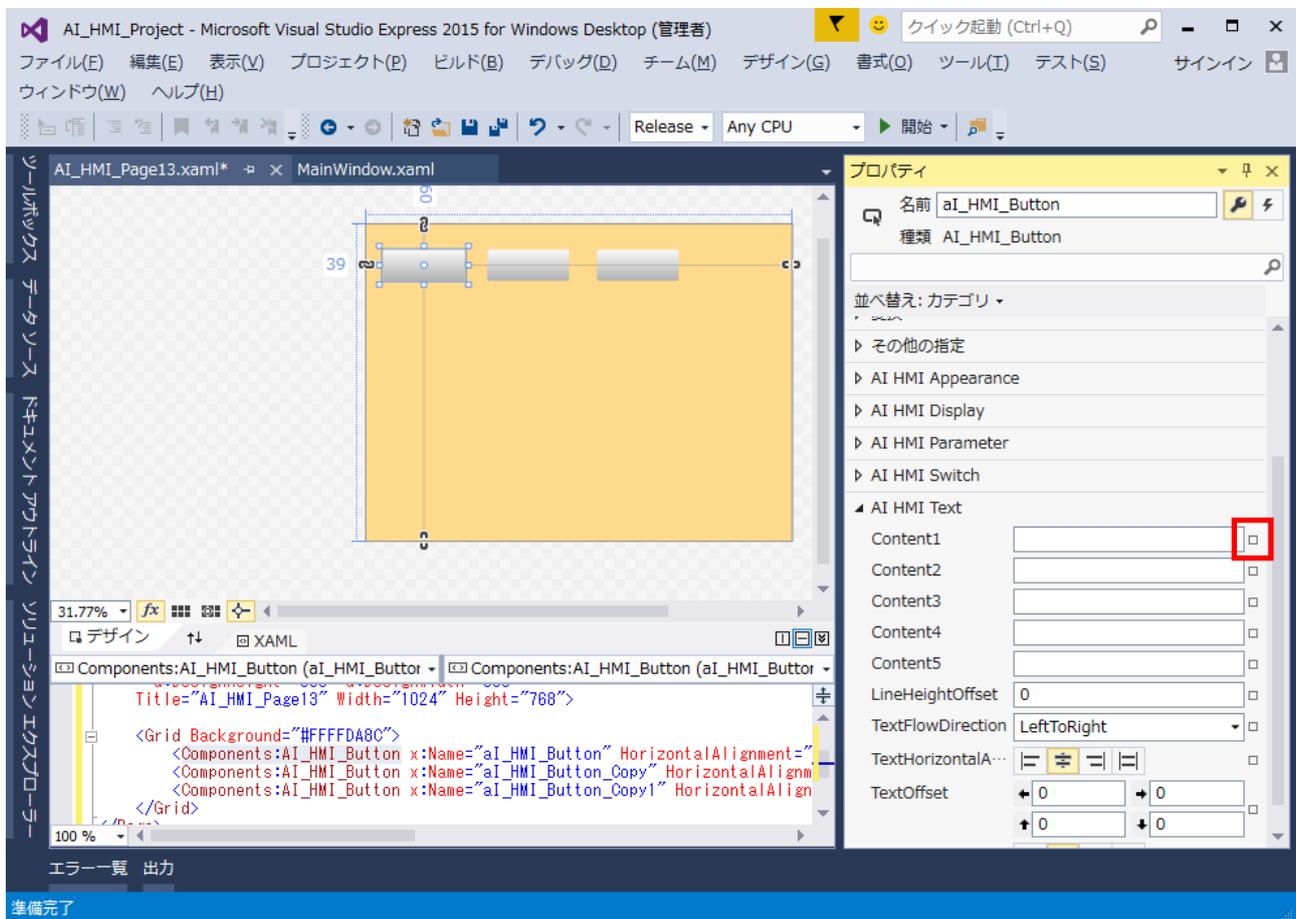


図 10-2-2 テキストプロパティ

③ ポップアップするメニューから「データバインディングの作成」を選択してください。

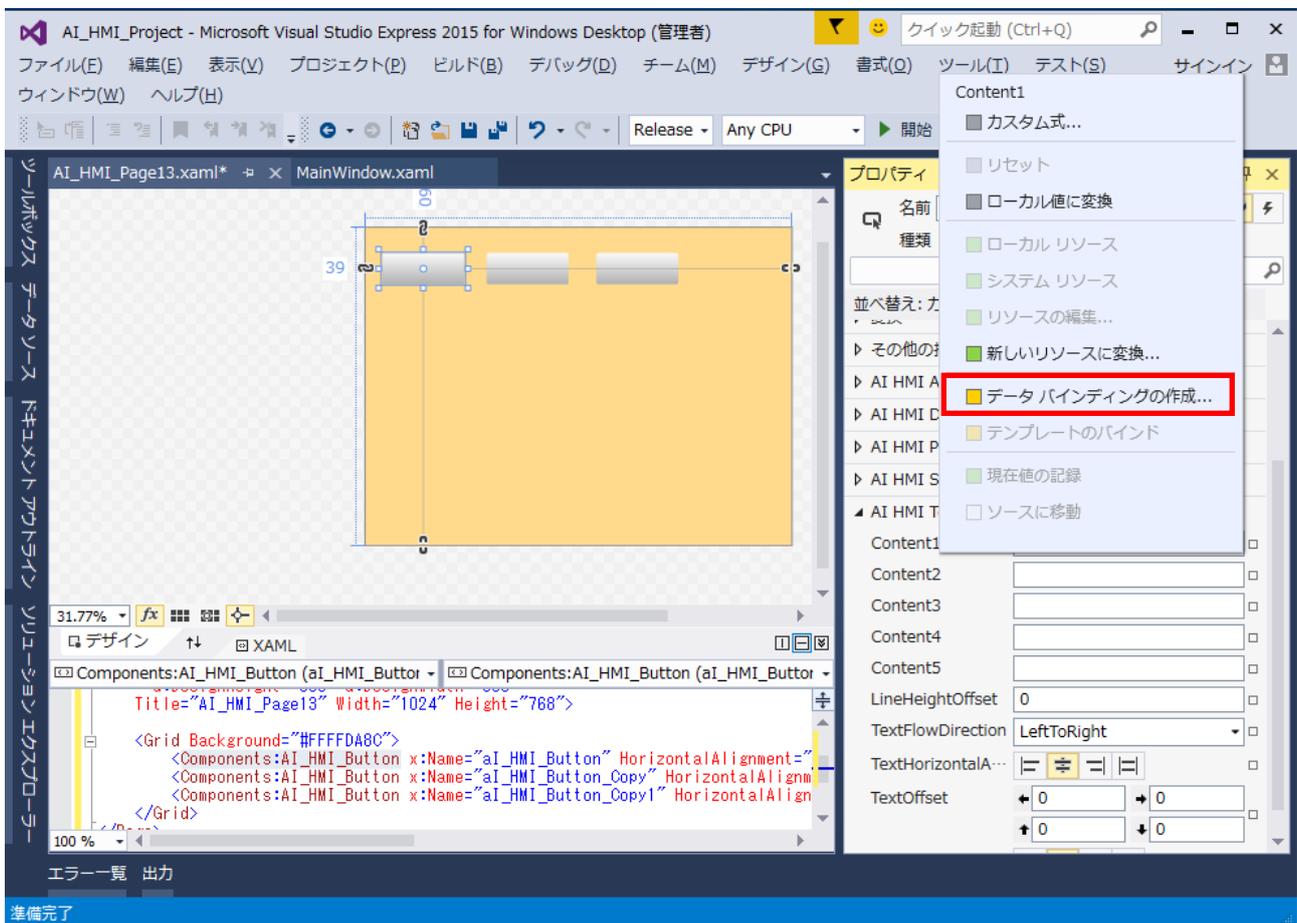


図 10-2-3 データバインディングの作成

- ④ データバインディングの作成画面が開きます。

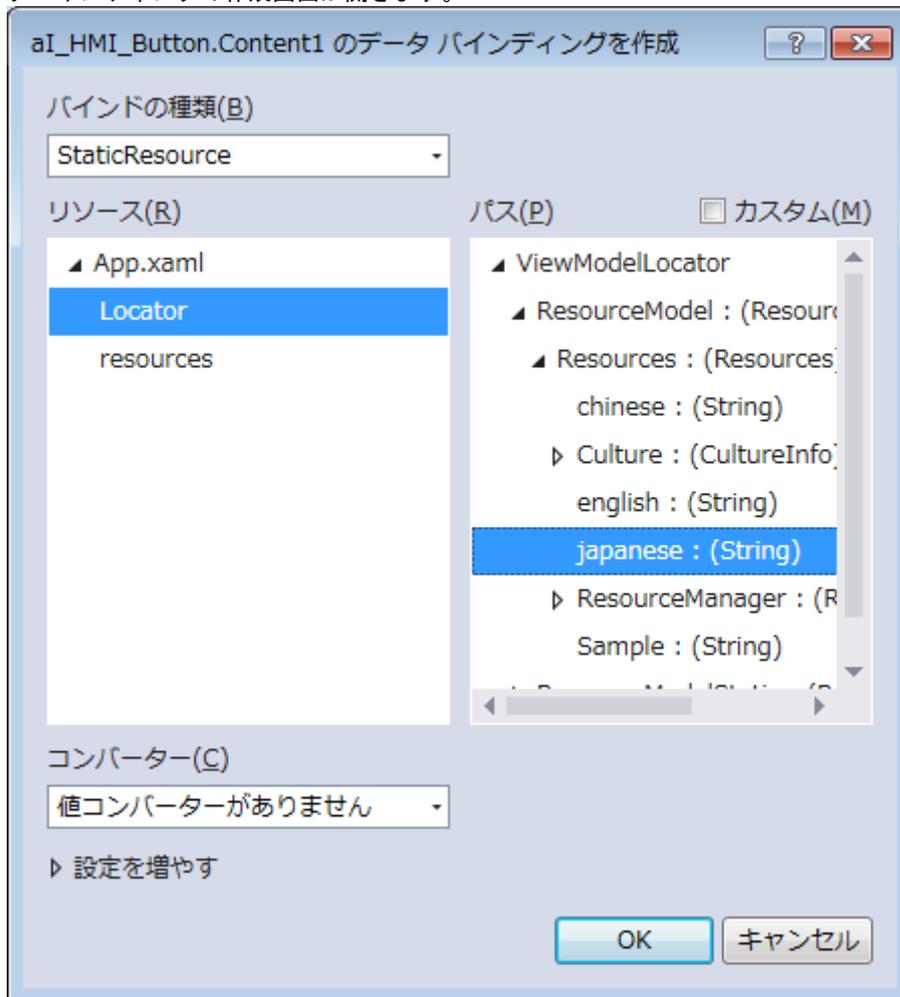


図 10-2-4 データバインディングの作成画面

以下の順番で設定をしてください。

- ・バインドの種類 : StaticResource
- ・リソース : Locator
- ・パス : ViewModelLocator – ResourceModel – Resources – japanese(String)

設定後、「OK」ボタンを押してください。

設定が完了するとプロパティの AI HMI Text – Content1 の右側の口が黄色に変化していることを確認してください。

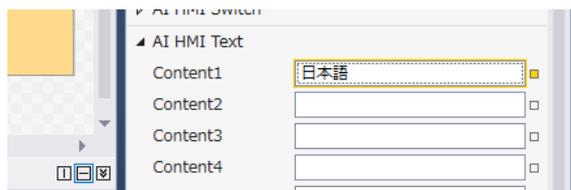


図 10-2-5 データバインディング完了

同様に他の 2 つの AI HMI Button も以下のように設定してください。

英語ボタン

- ・バインドの種類 : StaticResource
- ・リソース : Locator
- ・パス : ViewModelLocator – ResourceModel – Resources – english (String)

中国語ボタン

- ・バインドの種類 : StaticResource
- ・リソース : Locator
- ・パス : ViewModelLocator – ResourceModel – Resources – chinese (String)

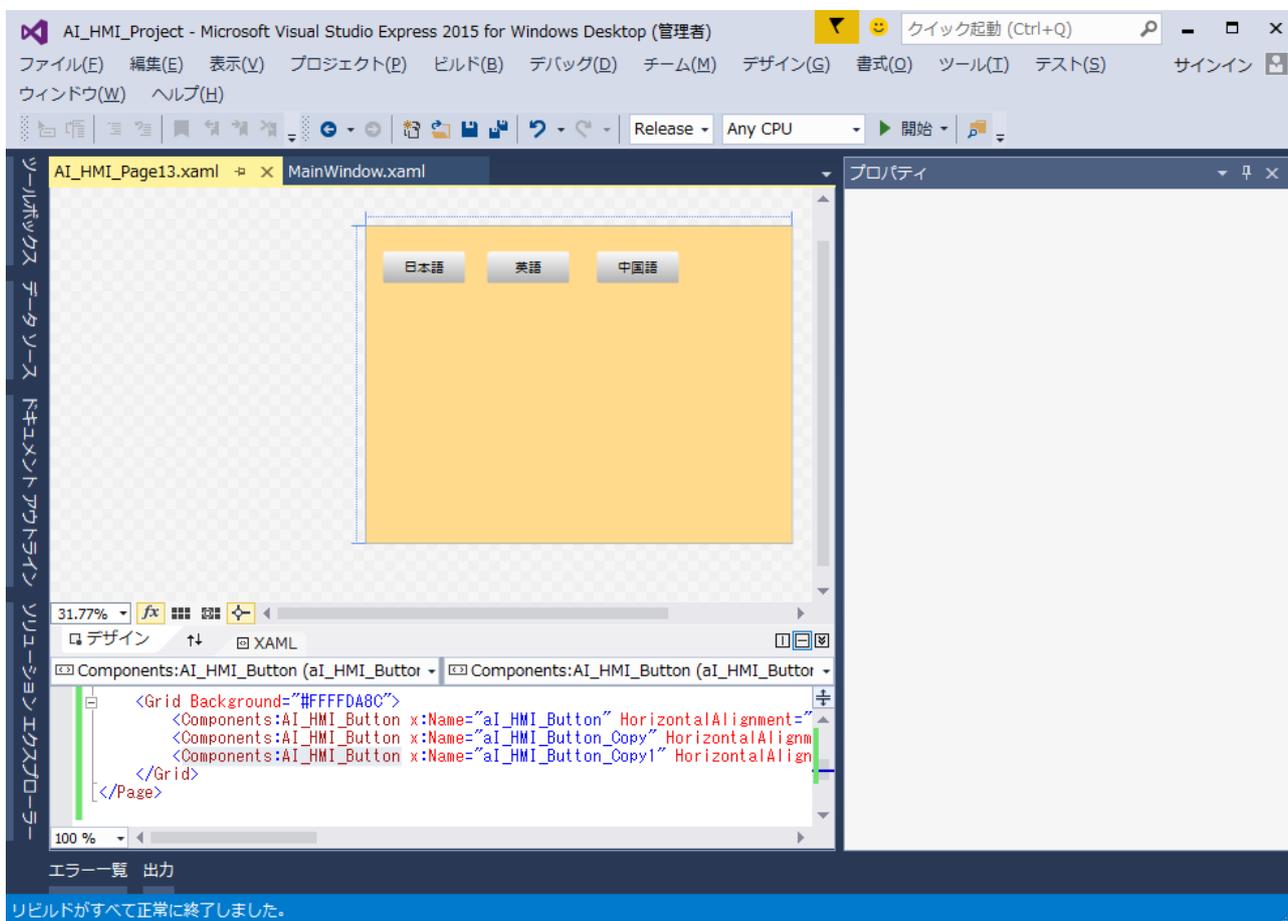


図 10-2-6 データバインディングの設定後

- ⑤ 作成したボタンの Switch の設定をします。
それぞれのボタンのプロパティを表 10-2-1 のように設定してください。

表 10-2-1 日本語ボタンプロパティ

項目		設定値	備考
AI HMI Switch	SwitchReference	Address	68000
		Type	STRING
		WordSwitch	Data
			共有メモリのユーザ用システム領域の LOCAL_DATA_CODE (多言語切替コード) に日本語コード「ja-JP」を書き込みます。

表 10-2-2 英語ボタンプロパティ

項目		設定値	備考
AI HMI Switch	SwitchReference	Address	68000
		Type	STRING
		WordSwitch	Data
			共有メモリのユーザ用システム領域の LOCAL_DATA_CODE (多言語切替コード) に英語コード「en-US」を書き込みます。

表 10-2-3 中国語ボタンプロパティ

項目		設定値	備考
AI HMI Switch	SwitchReference	Address	68000
		Type	STRING
		WordSwitch	Data
			共有メモリのユーザ用システム領域の LOCAL_DATA_CODE (多言語切替コード) に中国語コード「zh-CN」を書き込みます。

- ⑥ 作成した画面を実際に動作させて言語切替機能が動作することを確認します。
『第6章 作成したアプリの実行環境PCでの動作』を参照し、作成した画面アプリを動作させてください。

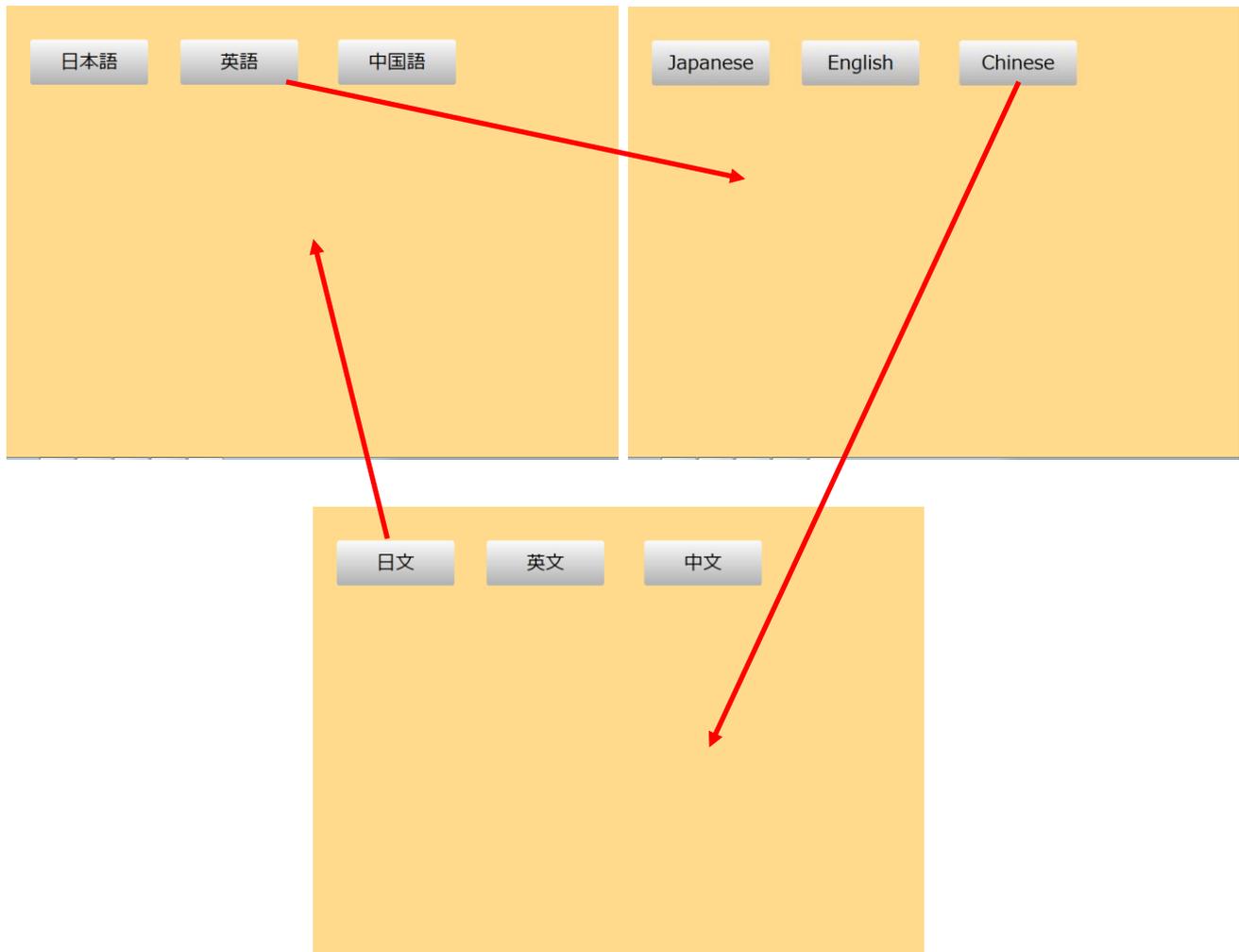


図 10-2-7 言語切替画面の実行

言語ボタンを押すことでそれぞれの言語に切り替わることを確認してください。

※ 多言語機能を使用するとき、実際にアプリケーションを実行しようとするときエラーが発生することがあります。その場合は以下の手順に従って再コンパイルを行ってください。

1. ソリューションエクスプローラの「Properties – Resource.resx – Resources.Designer.cs」を右クリックし、「コードの表示」を選択してください。

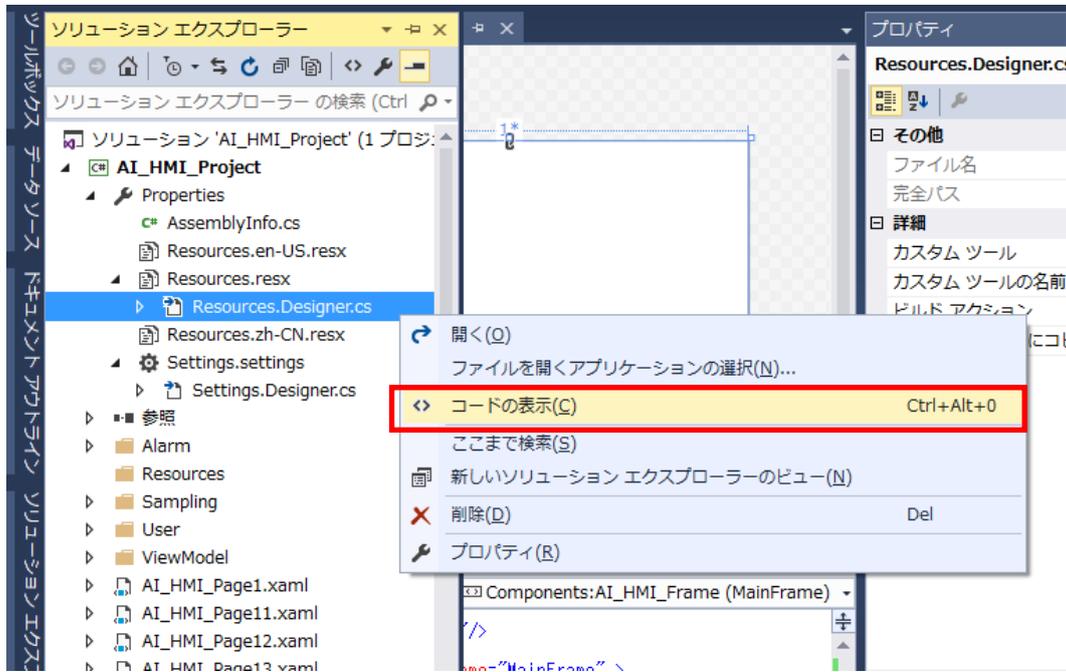


図 10-2-8 コードの修正

2. 表示されたコードの 31 行目の内容を以下のように修正してください。

変更前

```
internal Resources() {
```

↓

変更後

```
public Resources() {
```



図 10-2-9 修正するコード

3. 再度コンパイルを行ってください。

第 11 章 サンプルプログラムについて

AI-HMI はサンプルプログラムを同梱しています。

サンプルプログラムは AI-HMI の開発環境インストール時に以下のフォルダに展開されます。

C:\¥ALGOSYSTEM¥AI-HMI 開発用 SDK¥Sample¥V¥S¥AI-HMI_Sample

ソースのみの状態で格納されています。

AI_HMI_Sample.sln を実行し、VisualStudio 上で『5-4 アプリケーションのコンパイルとデバッグ』の手順に従いコンパイルしてアプリケーションを実行してください。



図 11-1 サンプルプログラム

第12章 AI-PLC 開発環境での使い方(OPC 変数)

AI-HMI は Intime 上で動作するソフトウェア PLC(AI-PLC)との連携を前提にしています。

AI-HMI と AI-PLC の相互の関係については『0

はじめに

AI-HMIはMicrosoft社製Visual Studioを開発環境として利用できるHMI画面作成ソフトです。
AI-HMI-SDKはVisualStudioのアドオンとして機能するため、下記の対象バージョンのVisual Studioを用意してください。

■動作対象のVisual Studio

Visual Studio 2013
Visual Studio 2015
Visual Studio 2017
Visual Studio 2019
Visual Studio 2022

※ 上記以外では動作確認していません。

※ **Visual Studioはお客様とマイクロソフト社との直接契約をしていただき、ご利用ください。**

AI-HMIで作成したアプリケーションはAI-HMIランタイム搭載製品でのみ動作できます。
AI-HMIランタイム搭載製品は以下の通りです。

- 高機能表示機(AI-HMI版)
 - G-AP4A-xxxシリーズ
 - G-EC4A-xxxシリーズ
- オールインワンコントローラ(AI-HMI版)
 - C-AP4A-xxxシリーズ
 - C-EC4A-xxxシリーズ
 - C-NP7A-xxxシリーズ
 - C-NP6A-xxxシリーズ
 - C-NP4C-xxxシリーズ
 - μFIXコントローラ

これら以外の実行環境で実行した場合や開発環境にてデバッグ実行した場合は60分毎にライセンスを求めるダイアログが開くようになります。

Visual Studio2022対応について

Visual Studio2022に対してはAI-HMI-SDK ver1.3.2以降のバージョンで対応しています。
Ver1.3.2.1以前のAI-HMI-SDKをご利用の方はアップデートパッチを使用してアップデートをしてください。

- ・開発環境アップデート
弊社ホームページからアップデートパッチをダウンロードできます。
AI-HMI-SDKをセットアップしたPCで使用することでVisualStudio2022に対応できるようになります。
- ・プロジェクトコンバート
既にVer1.3.2.1以前のAI-HMI-SDKで作成しているプロジェクトについてはAI-HMI-SDKのCD-ROMに含まれているAI-HMI VS2022 Converterを使用することでVisual Studio2022に対応できます。
AI-HMI VS2022 Converterについての詳細はCD-ROM同梱の『AI-HMI VS2022 Converter取扱説明書』を

ご参照ください。

AI-HMI VS2022 Converterは弊社ホームページからもダウンロードできます。

『AI-HMIについて』を参照してください。

実行環境用の AI-PLC のプロジェクトは開発環境の MULTIPROG で作成します。

開発環境の MULTIPROG でのプロジェクトの作成方法は「PLC アプリケーション 作成マニュアル」を参照してください。

AI-HMI と AI-PLC 間で通信をする方法としては共有メモリを使用する方法と OPC 変数を使用する方法があります。

本章では既に作成されている MULTIPROG のプロジェクトを OPC 変数を介して AI-HMI と連携できるようにするための手順について説明します。

1 2 - 1 OPC 変数の登録

OPC 変数を登録します。

本項で登録した OPC 変数はそのまま AI-HMI 上で変数として使用が可能です。

- ① MULTIPROG のプロジェクトから「Global Variables」をダブルクリックしてください。

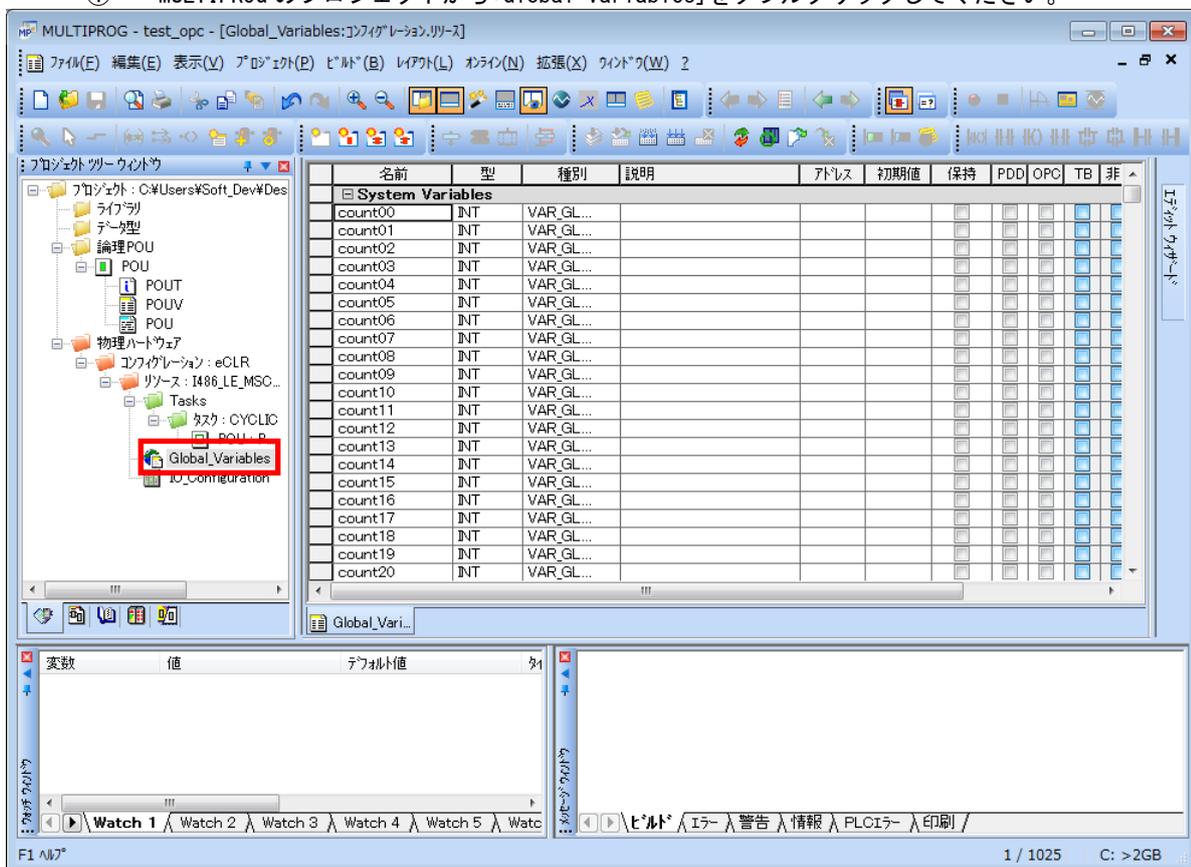


図 12-1-1 I/O Configuration

- ② グローバル変数一覧画面が開きます。
OPC 変数として扱いたい変数の「OPC」項目にチェックを入れてください。

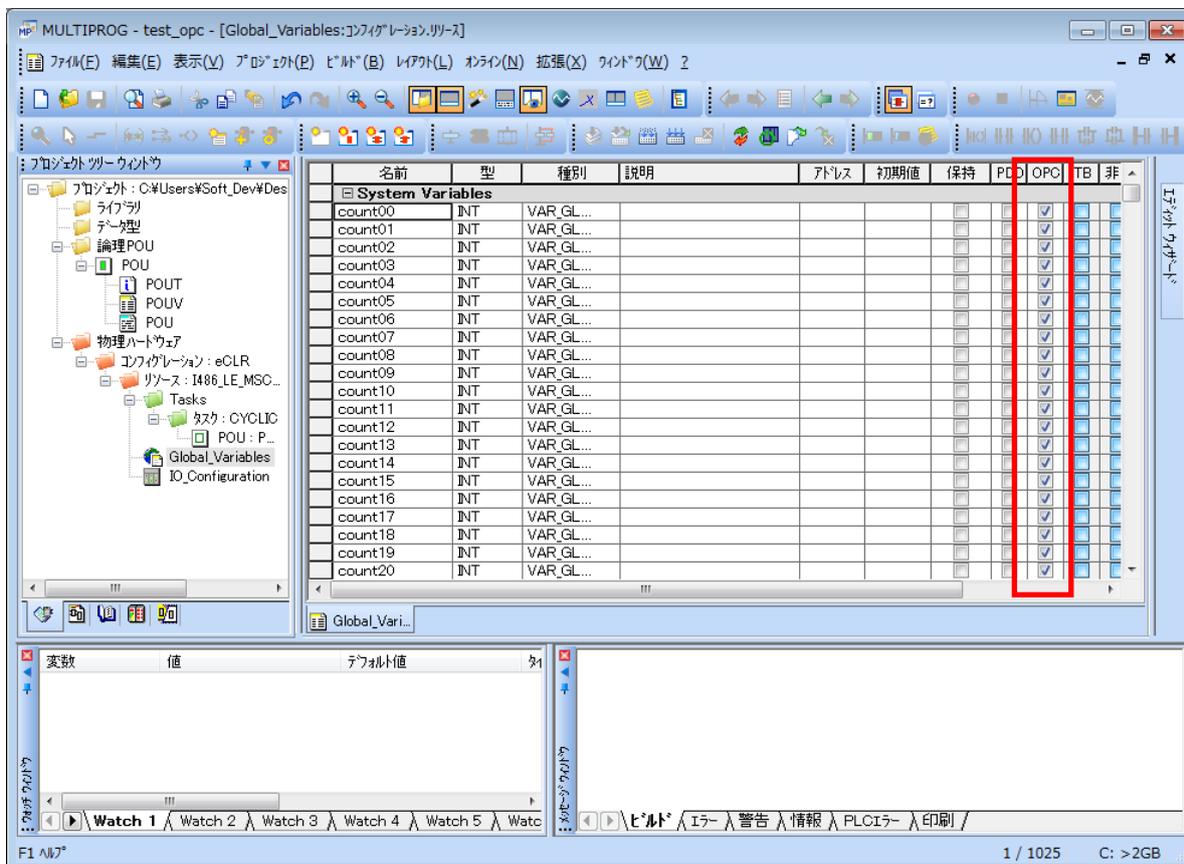


図 12-1-2 OPC 属性の付与

「OPC」にチェックを入れた変数は OPC 変数として AI-HMI 上で使用することができます。
AI-HMI 上で使用する場合、登録した OPC 変数名をそのまま変数名として使用してください。

OPC 変数として AI-HMI 上で使用する場合、以下の制限が付きます。

- ・ Global 変数以外は AI-HMI 上では使用できない。
グローバル変数以外のローカル変数は「OPC」にチェックを入れても
AI-HMI 上では使用することができません。
AI-HMI 上で使用する変数は Global Variables の変数一覧に登録してください。

第 13 章 AI-PLC 開発環境での使い方(共有メモリ)

AI-HMI は INtime 上で動作するソフトウェア PLC(AI-PLC)との連携を前提にしています。

AI-HMI と AI-PLC の相互の関係については『0 AI-HMI について』を参照してください。

実行環境用の AI-PLC のプロジェクトは開発環境の MULTIPROG で作成します。

開発環境の MULTIPROG でのプロジェクトの作成方法は「PLC アプリケーション 作成マニュアル」を参照してください。

AI-HMI と AI-PLC 間で通信をする方法としては共有メモリを使用する方法と OPC 変数を使用する方法があります。

本章では既に作成されている MULTIPROG のプロジェクトを共有メモリを介して AI-HMI と連携できるようにするための手順について説明します。

13-1 I/O グループの設定

ハードウェアの I/O グループの宣言を行います。

本項で設定した I/O のアドレスは変数に割り当てることで使用可能になります。

- ① MULTIPROG のプロジェクトから「I/O_Configuration」をダブルクリックしてください。

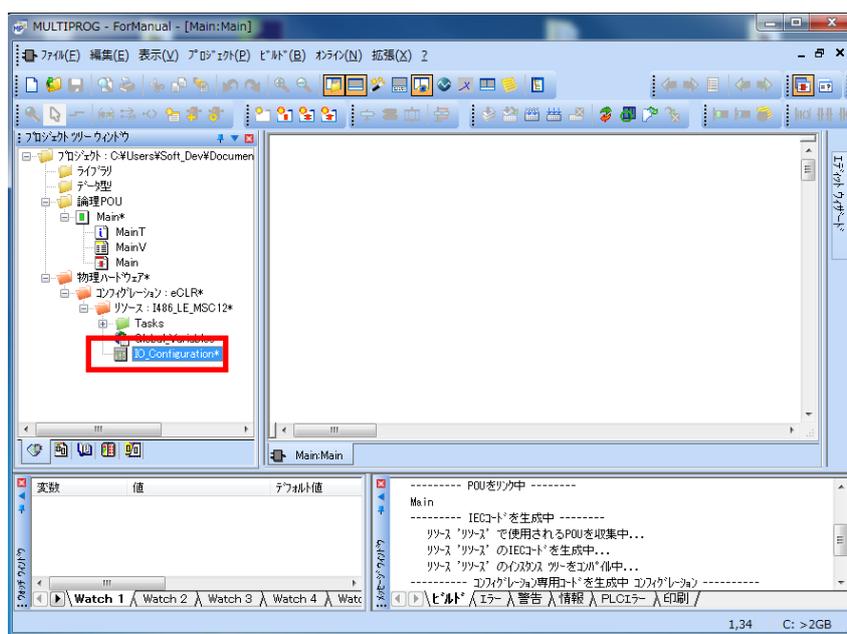


図 13-1-1 I/O_Configuration

- ② I/O コンフィグレーション画面が開きます。
「追加」ボタンをクリックしてください。



図 13-1-2 I/O グループを追加

③ I/O グループの追加画面が開きます。

名前、タスク、開始アドレス、長さ、リフレッシュ、デバイスを設定し、ボード I/O モジュールを選択してください。設定する値は表 13-1-1 を参照してください。

図 13-1-3 I/O グループの設定

表 13-1-1 I/O コンフィグレーションのプロパティ

パラメータ名	説明	設定値
名前	I/O グループ名称	AI_HMI
タスク	全入出力を行うタスク	任意
開始アドレス	IEC61131 規格の開始アドレス	任意
長さ	-	1
リフレッシュ	-	タスクによる
デバイス	-	ドライバ
ボード/I/O モジュール	接続モジュール名称	ユーザ定義

設定完了後、「ドライバパラメータ」ボタンをクリックしてください。

- ④ 標準デバイスのドライバ情報画面が開きます。
表 13-1-2 を参照してドライバパラメータを設定してください。



図 13-1-4 標準デバイスドライバ情報の設定

表 13-1-2 ドライバパラメータ

パラメータ名	説明	設定値
ドライバ名	I/O ドライバ名称	IoDriver_HMI
パラメータ 1	-	0
パラメータ 2	-	0
パラメータ 3	-	0
パラメータ 4	-	0
データ型	-	空白

設定完了後、「OK」ボタンをクリックしてください。

- ⑤ 以上で設定は完了です。
設定した値が表示されていることを確認し、「OK」ボタンを押してください。

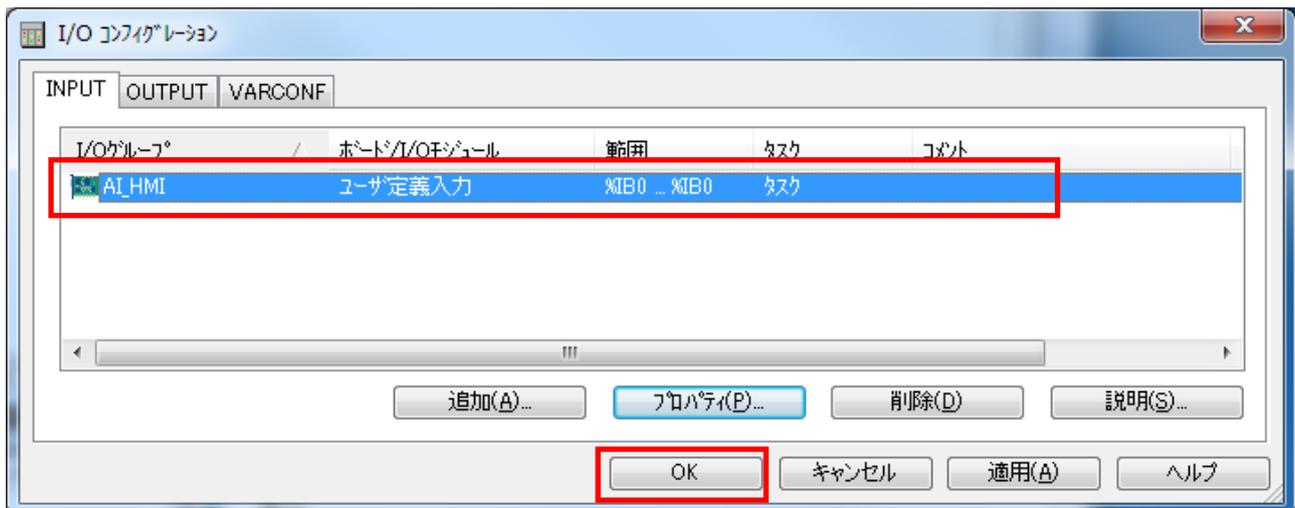


図 13-1-5 I/O コンフィグレーションの確認

※ Output の I/O グループは設定する必要はありません。

13-2 変数のアドレス指定

物理デバイスとして I/O ドライバで割り付けられた共有メモリにアクセスするためには MULTIPROG の変数にアドレスを割り付ける必要があります。

あらかじめ MULTIPROG で作成している変数にアドレスを記述することで変数と共有メモリを紐付けします。

MULTIPROG 上でのアドレスの表記方法については表 13-2-1 を参考にしてください。

表 13-2-1 アドレスの表記例

登録する変数型	記述例	備考
BOOL	%MX3. XXXXXX. 0	BOOL 型変数として共有メモリへアクセスします。 最後の桁は BIT 番号を意味しますが、AI-HMI では 0 で固定です。
INT	%MW3. XXXXXX	16 ビットの変数として共有メモリへアクセスします。 符号の有無は AI-HMI 側で扱います。
UINT		
DINT		
UDINT	%MD3. XXXXXX	32 ビットの変数として共有メモリへアクセスします。 符号や小数点の有無は AI-HMI 側で扱います。
REAL		
LREAL	%ML3. XXXXXX	64 ビットの変数として共有メモリへアクセスします。
STRING	%MD3. XXXXXX	文字列型変数の先頭アドレスへアクセスします。 デフォルトの最大サイズは 80byte です。※2

※1 XXXXXX は参照する共有メモリのアドレスを意味します。

共有メモリのアドレスの構成については『4-3 AI-HMI用Intime共有メモリ』を参照してください。

※2 デフォルトの文字列以外に任意の最大サイズの文字列型変数を追加することができます。

追加する場合は MULTIPROG 内のデータ型に以下のコードを追加してください。

TYPE

MYSTRING:STRING(YYYYY);

END_TYPE

(YYYYY は任意の 1~32766 の数値を入力できます。)

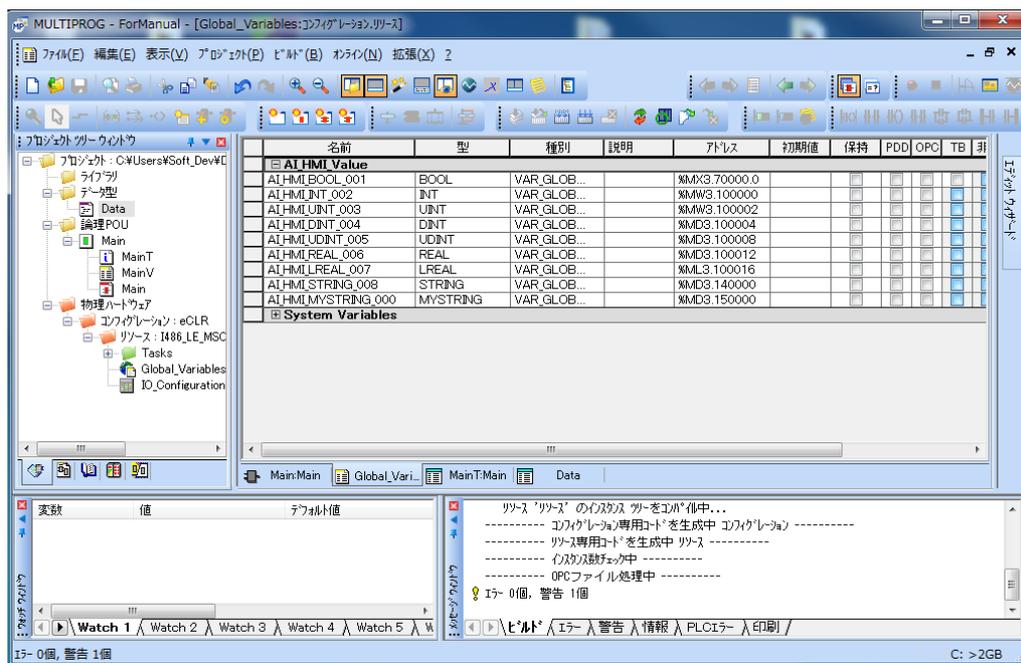


図 13-2-1 変数割付の例

アドレスの表現方法は IEC61131-3 に準拠しています。
 アドレスの表現方法の仕様は以下のとおりです。

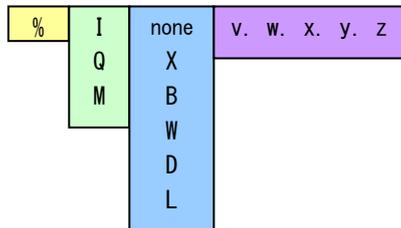


表 13-2-2 アドレスの構成要素の意味

記号	意味	備考
%	直接表現符号<%>ではじまる。	アドレスの先頭に付与します
I	入力	AI-HMI では使用しません
Q	出力	AI-HMI では使用しません
M	メモリ	AI-HMI では M 固定です
none	Bit	BOOL 変数としてアクセス
X	Bit	BOOL 変数としてアクセス
B	Byte(8 ビット)	AI-HMI では使用しません
W	Word(16 ビット)	INT、UINT 変数としてアクセス
D	Double Word(32 ビット)	DINT、UDINT、REAL 変数としてアクセス
L	Long Word(64 ビット)	LREAL 変数としてアクセス
v. w. x. y. z	複数桁からなら階層的アドレス。解釈は製造者に依存。	v : 3 固定 w : 未使用 x : 未使用 y : 共有メモリアドレス z : ビット番号

第 1 4 章 ユーザ用システム変数

AI-HMI では共有メモリ内にユーザアクセス可能なシステム変数を用意しています。

これらのシステム変数は他の共有メモリ変数と同様にアクセスが可能です。

ユーザ用システム変数の詳細は表 14-1 の通りです。

表 14-1 ユーザ用システム変数一覧(1/2)

システム変数	アドレス	データ型	内容
LOCAL_DATA_CODE	68000	STRING 型	多言語切替用の変数です。 言語コードを入力することで表示文字列の言語を切り替えます (※1)
RTC_SET_FLG	68100	UDINT 型	時刻を書き込むためのフラグです。 1 を入力すると RTC_DATA_YEAR~RTC_DATA_MILLISECONDS の値を WindowsSystem 時刻と SecondaryRTC に書き込みます。 時刻の書き込みが完了すると自動的に 0 が入力されます。
RTC_SET_MODE	68110	UDINT 型	時刻を書き込む際にローカル時刻または UTC のどちらとして書き 込むかを設定します。 ・ 0 : ローカル時刻 ・ 1 : UTC(協定世界時)
RTC_DATA_YEAR	68120	UINT 型	WindowsSystem 時刻および SecondaryRTC へ書き込む時刻データ です。
RTC_DATA_MONTH	68122	UINT 型	
RTC_DATA_DAYOFWEEK	68124	UINT 型	
RTC_DATA_DAY	68126	UINT 型	
RTC_DATA_HOUR	68128	UINT 型	
RTC_DATA_MINUTE	68130	UINT 型	
RTC_DATA_SECOND	68132	UINT 型	
RTC_DATA_MILLISECONDS	68134	UINT 型	
SYSTEM_DATETIME_SECOND	68200	UINT 型	現在の WindowsSystem 時刻を読み出すことができます。
SYSTEM_DATETIME_MINUTE	68202	UINT 型	
SYSTEM_DATETIME_HOUR	68204	UINT 型	
SYSTEM_DATETIME_DAY	68206	UINT 型	
SYSTEM_DATETIME_MONTH	68208	UINT 型	
SYSTEM_DATETIME_YEAR	68210	UINT 型	
SYSTEM_DATETIME_DAYOFWEEK	68212	UINT 型	
LCDBACKLIGHT_BRIGHTNESS	68300	UINT 型	LCD バックライトの輝度を設定できます。 設定値の範囲は以下のようになります。 ・ 255(暗い)~0(明るい)
LCDBACKLIGHT_POWER	68302	UINT 型	LCD バックライトの電源を ON/OFF できます。 ・ 0 : ON ・ 1 : OFF
WINDOW_CLOSE	68400	UINT 型	1 を書き込むことで、アプリケーションをクローズできます。

表 14-1 ユーザ用システム変数一覧(2/2)

システム変数	アドレス	データ型	内容
ALARM_CSV_SAVE	68504	UINT 型	アラームデータの保存をする際に使用します。 1 を入力すると SRAM のアラームデータ領域の内容を CSV ファイルに保存します。 保存が完了すると自動的に 0 が入力されます。(※2)
ALARM_CSV_ERROR	68506	UINT 型	アラームデータの保存結果を自動的に入力します。 保存成功時は 1、保存失敗時は-1 が入力されます。 ALARM_CSV_ERROR は次回サンプリングデータ保存時まで値は変化しません。(※2)
ALARM_ALL_CLEAR	68508	UINT 型	アラームデータのクリアをする際に使用します。 1 を入力すると SRAM のアラームデータ領域の内容をクリアします。 クリアが完了すると自動的に 0 が入力されます。(※2)
TREND_XAXIS_DATASPAN	68510+Offset (※4)	UINT 型	トレンドグラフの X 軸全体の表示区間です。 DINT 型で分単位の値を入力します。 値を変更することでトレンドグラフの表示が変化します。(※3)
TREND_XAXIS_INTERVAL	68512+Offset (※4)	UINT 型	トレンドグラフの X 軸の中間線の間隔です。 DINT 型で分単位の値を入力します。 値を変更することでトレンドグラフの表示が変化します。(※3)
SAMPLING_CSV_SAVE	68514+Offset (※4)	UINT 型	サンプリングデータの保存をする際に使用します。 1 を入力すると SRAM のサンプリングデータ領域の内容を CSV ファイルに保存します。 保存が完了すると自動的に 0 が入力されます。(※3)
SAMPLING_CSV_ERROR	68516+Offset (※4)	UINT 型	サンプリングデータの保存結果を自動的に入力します。 保存成功時は 1、保存失敗時は-1 が入力されます。 SAMPLING_CSV_ERROR は次回サンプリングデータ保存時まで値は変化しません。(※3)
SAMPLING_ALL_CLEAR	68518+Offset (※4)	UINT 型	サンプリングデータのクリアをする際に使用します。 1 を入力すると SRAM のサンプリングデータ領域の内容をクリアします。 クリアが完了すると自動的に 0 が入力されます。(※3)

※1 多言語切替についての詳細は『第 10 章 多言語切替機能』を参照ください。

※2 アラーム機能についての詳細は『9-1 アラーム機能』を参照ください。

※3 トレンドグラフ、サンプリング機能についての詳細は『9-4 サンプリング機能』を参照ください。

※4 Offset は「(グループ番号 - 1) × 10」になります。

第 15 章 システム内システム変数

AI-HMI では共有メモリ、OPC 変数の他にシステム内部で使用するシステム変数が用意されています。これらのシステム変数は AI HMI Button の SystemSwitchReference 機能からアクセスが可能です。システム内システム変数の詳細は表 15-1 の通りです。

表 15-1 システム内システム変数一覧

システム変数	属性	内容
USER_LOGIN_ACTION	書込専用	ユーザログインをする際に使用します。 1 を入力するとログインユーザ名とログインパスワードの比較を行います。 比較が完了すると自動的に 0 が入力されます。(※1)
USER_LOGOFF_ACTION	書込専用	ユーザログアウトをする際に使用します。 1 を入力するとログイン権限番号、ログインユーザ名、ログインパスワードをクリアします。 クリアが完了すると自動的に 0 が入力されます。(※1)

※1 ユーザログイン機能についての詳細は『9-3 ユーザログイン機能』を参照ください。

第 16 章 RAM Backup 領域

AI-HMI 対応機種は 150Mbyte の RAM Backup 領域を搭載しています。

AI-HMI ではこれらの領域のうち以下の図のような領域を使用しています。



図 16-1 SRAM 領域メモリマップ

第 17 章 Tips 集

17-1 一度設定したプロパティを初期化したい

VisualStudio のプロパティを初期化したい場合は以下の手順を実行してください。

- ① 削除したいプロパティの右端にある■をクリックしてください。

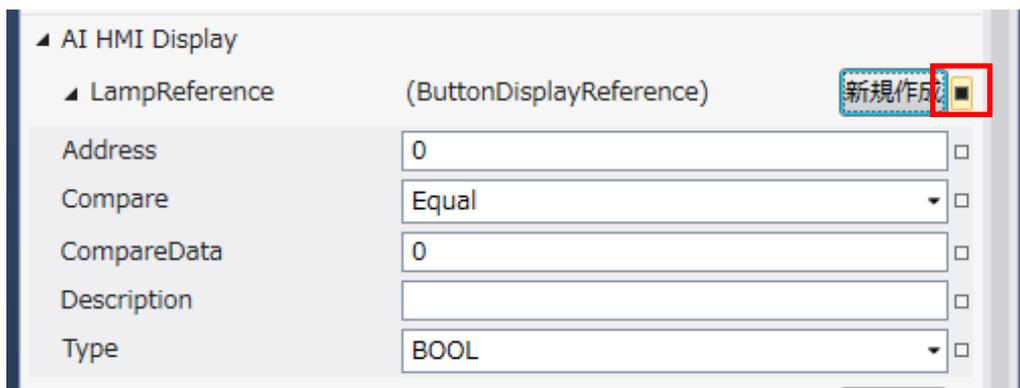


図 17-1-1 プロパティの初期化

- ② 表示されたメニューから「リセット」を選択してください。

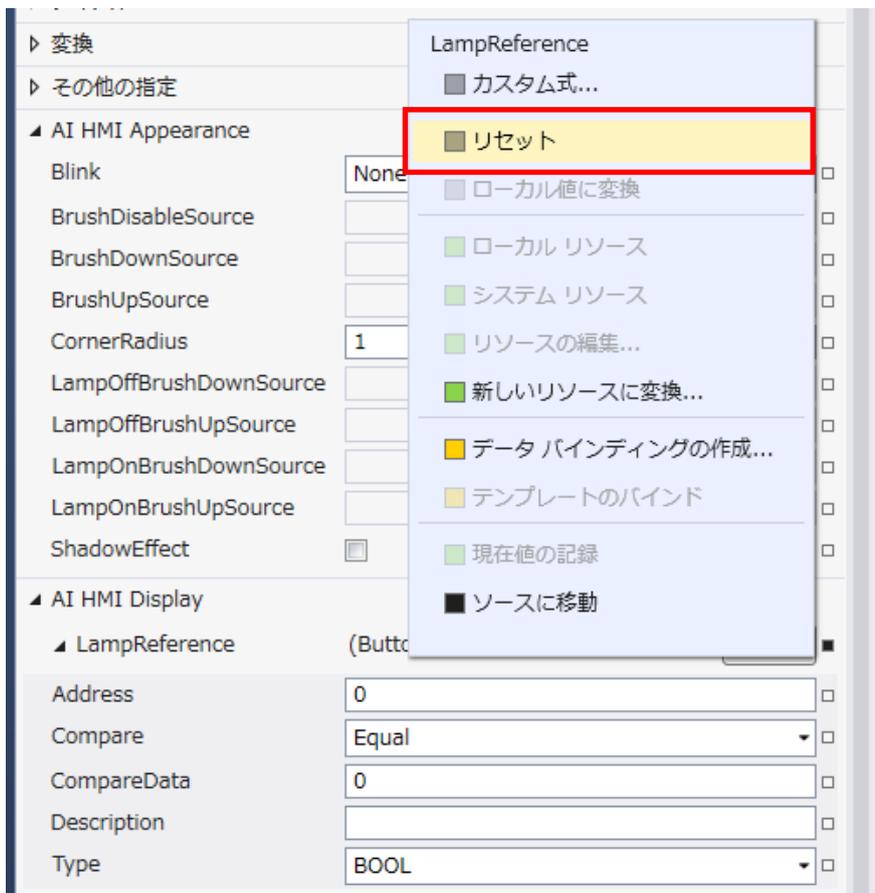


図 17-1-2 プロパティの初期化メニュー

- ③ 対象のプロパティが初期化され、■が□に変化することを確認してください。

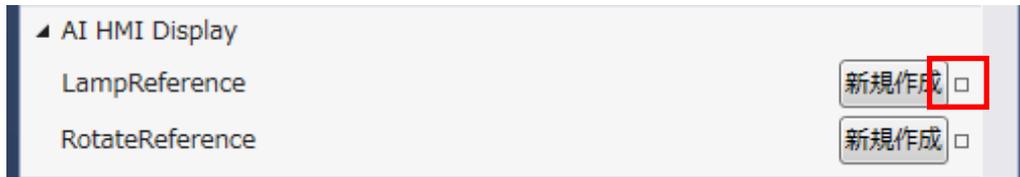


図 17-1-3 プロパティの初期化完了

- ※ プロパティの右端の口が黒塗りの場合、そのプロパティが変更されていることを意味します。白塗りの場合、そのプロパティが変更されていないことを意味します。

17-2 オブジェクトの画像を変更したい

VisualStudio のオブジェクトの画像を変更したい場合、ブラシプロパティを設定してください。以下にブラシプロパティの設定方法を示します。

- ① 変更したいブラシのプロパティをクリックしてください。



図 17-2-1 ブラシプロパティ

- ② ブラシメニューが表示されます。使用したいブラシの種類を選択してください。

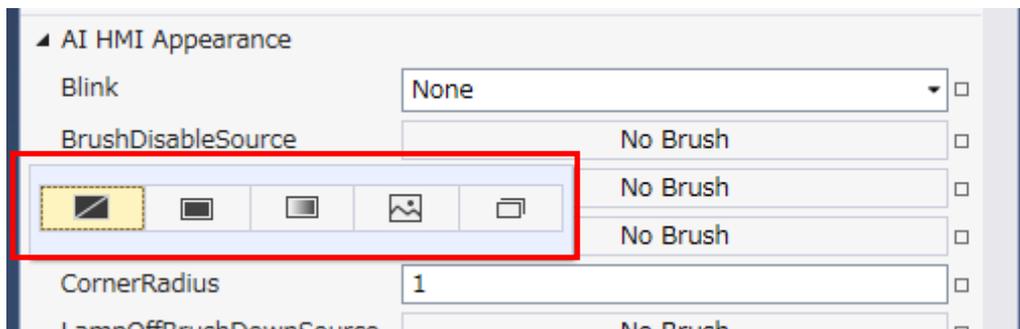


図 17-2-2 ブラシメニュー

- ③ ブラシを取り消したい場合。
ブラシメニューからブラシなしを選択してください。

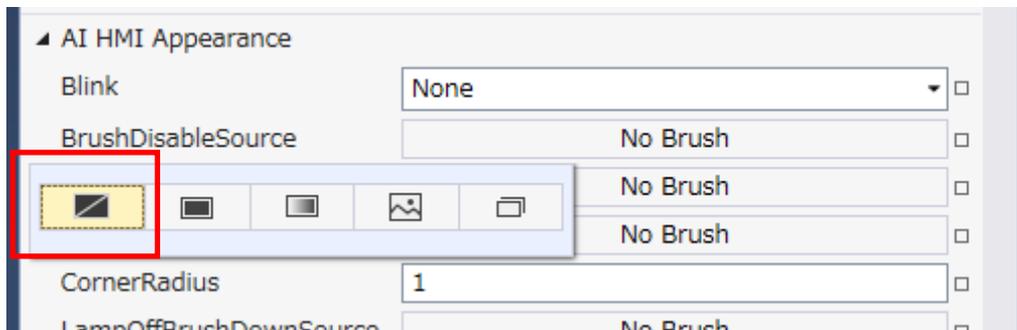


図 17-2-3 ブラシなし

- ④ 単色塗りつぶしをしたい場合。
ブラシメニューから単色ブラシを選択してください。
カラーパレットが表示されるので、使用したい色を選択してください。

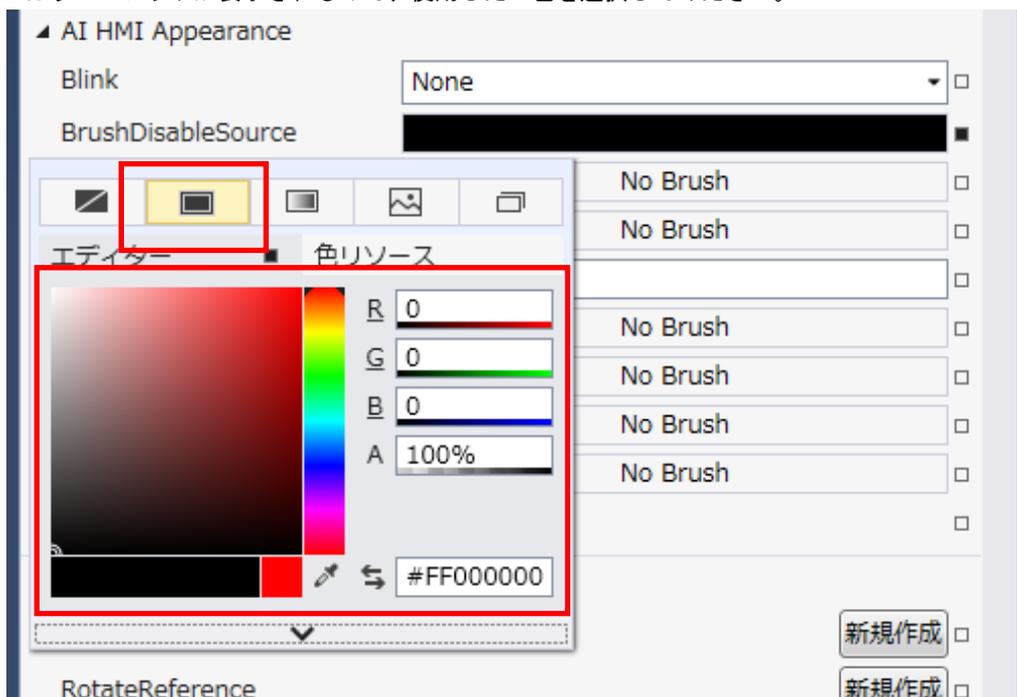


図 17-2-4 単色ブラシ

- ⑤ 単色ブラシの場合、画像の透過度の設定をすることが可能です。
カラーパレット下のオプションメニューをクリックし、表示される Opacity 設定を変更してください。

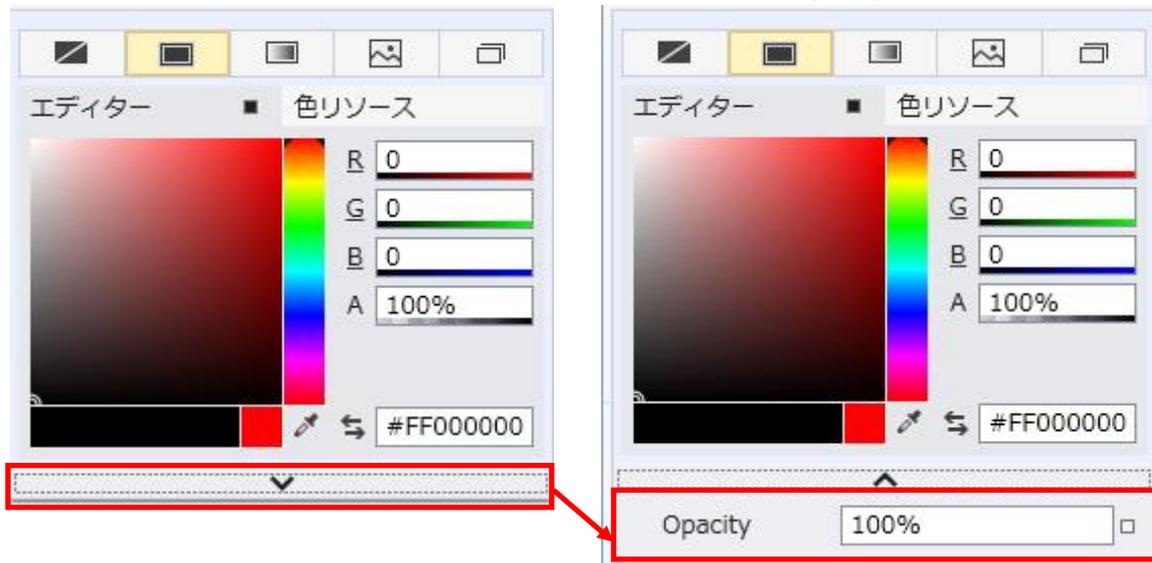


図 17-2-5 透過度設定

- ⑥ グラデーション塗りつぶしをしたい場合。
ブラシメニューからグラデーションブラシを選択してください。
カラーパレットが表示されるので、使用したいグラデーションを設定してください。

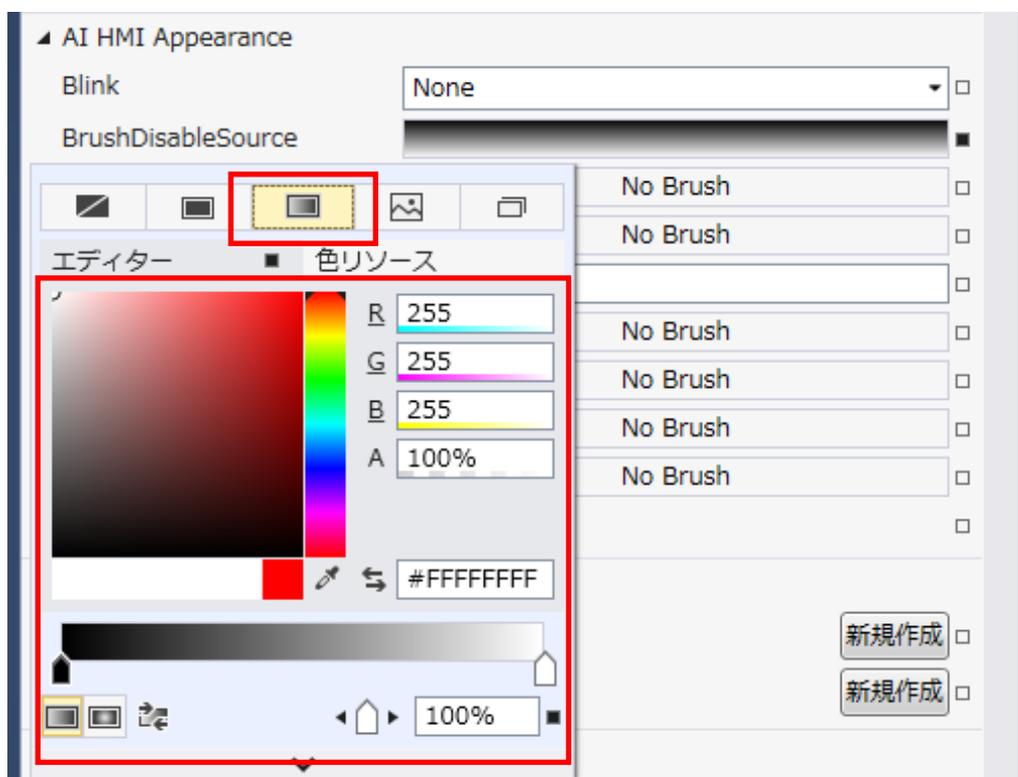


図 17-2-6 グラデーションブラシ

- ⑦ 画像ファイルを表示したい場合。
 ブラシメニューからタイルブラシを選択してください。
 タイル設定が表示されるので、使用したいファイルを選択してください。

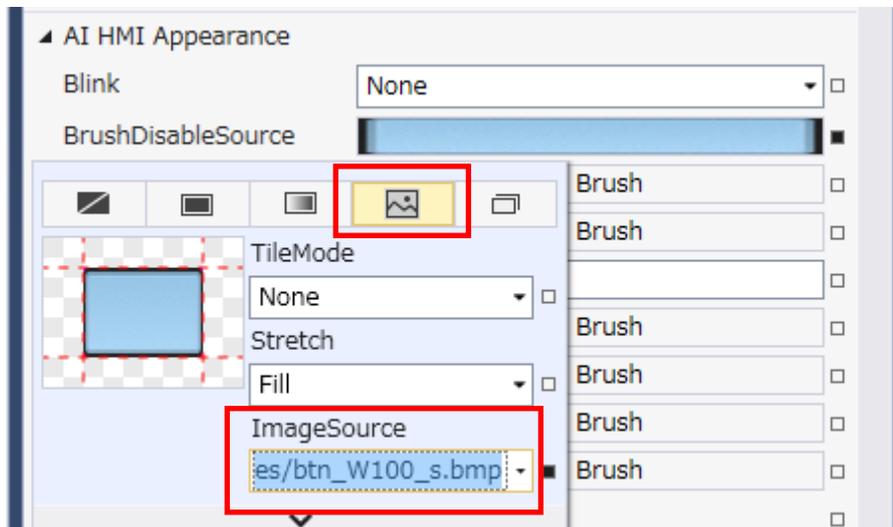


図 17-2-7 タイルブラシ

- ※ タイルブラシで画像ファイルを表示するためにはあらかじめ画像ファイルをリソースとして登録しておく必要があります。
 画像ファイルの登録方法については『17-3 画像ファイルを登録したい』を参照してください。

- ⑧ システムカラーで塗りつぶしたい場合。
 ブラシメニューからブラシリソースを選択してください。
 システム Brush リソースが表示されるので、色を選択してください。

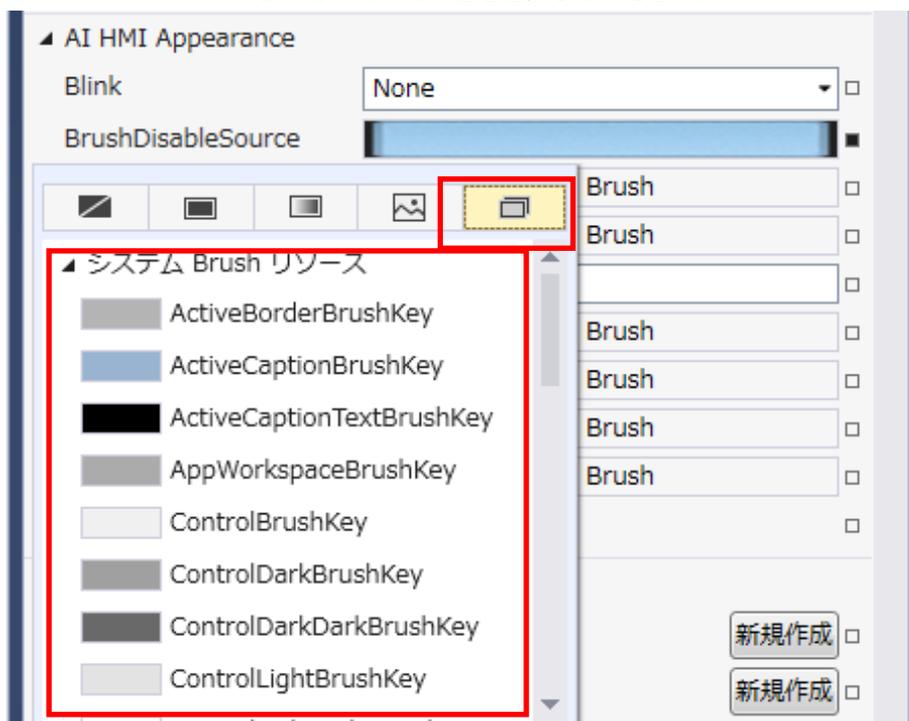


図 17-2-8 ブラシリソース

17-3 画像ファイルを登録したい

オブジェクト上に任意の画像ファイルを表示したい場合、あらかじめリソースとして画像ファイルを登録しておく必要があります。

以下に手順に従い、画像ファイルを登録してください。

- ① 登録したいファイルを Resource フォルダにコピーする。
登録したいファイルを現在編集中のプロジェクトの Resource フォルダへコピーしてください。
Resource フォルダは現在編集中のプロジェクトフォルダ直下にあります。

- ② ソリューションエクスプローラの Resource を右クリックしてください。

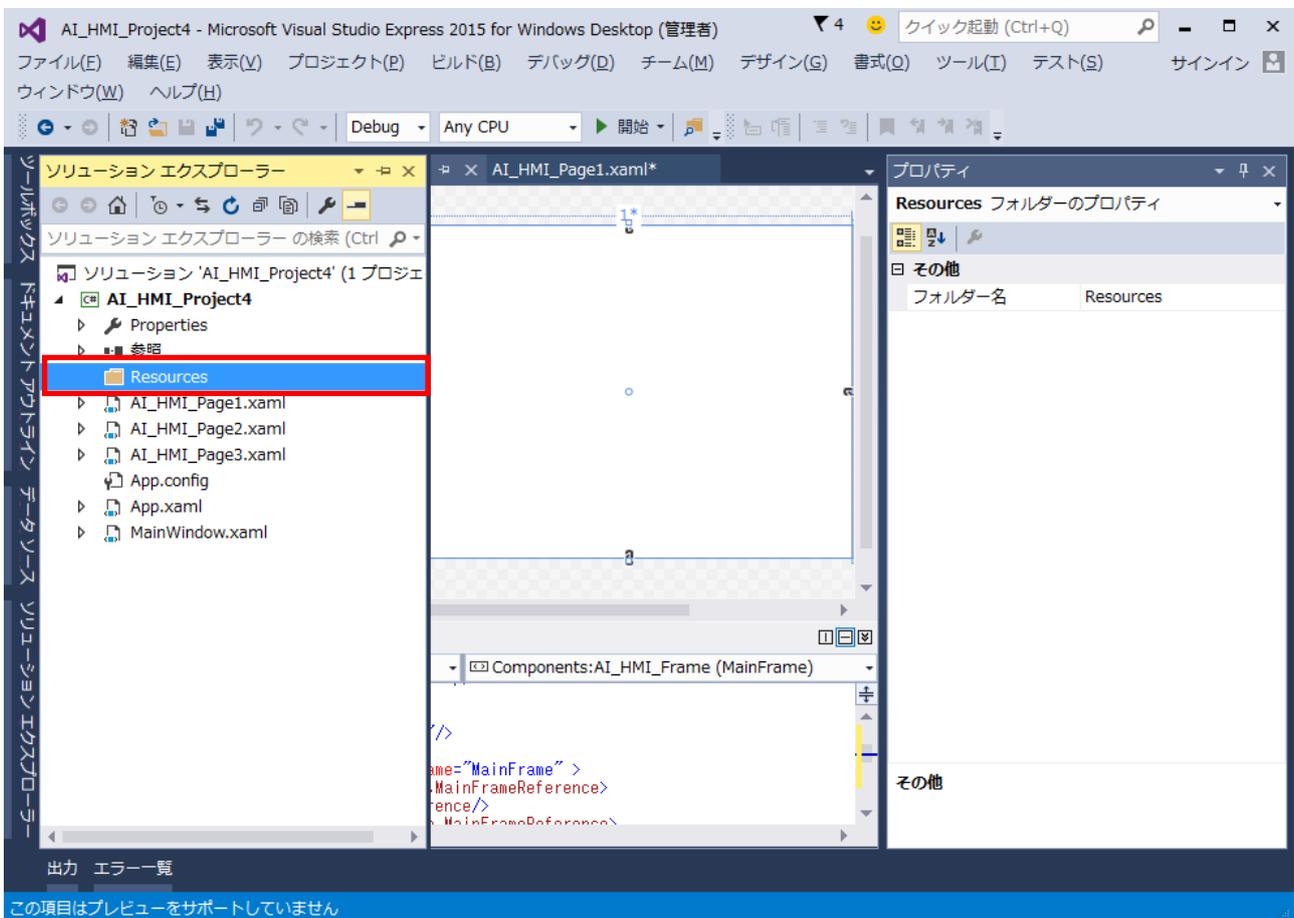


図 17-3-1 ソリューションエクスプローラの Resource

③ 右クリックメニューが表示されるので「追加 – 既存の項目」を選択してください。

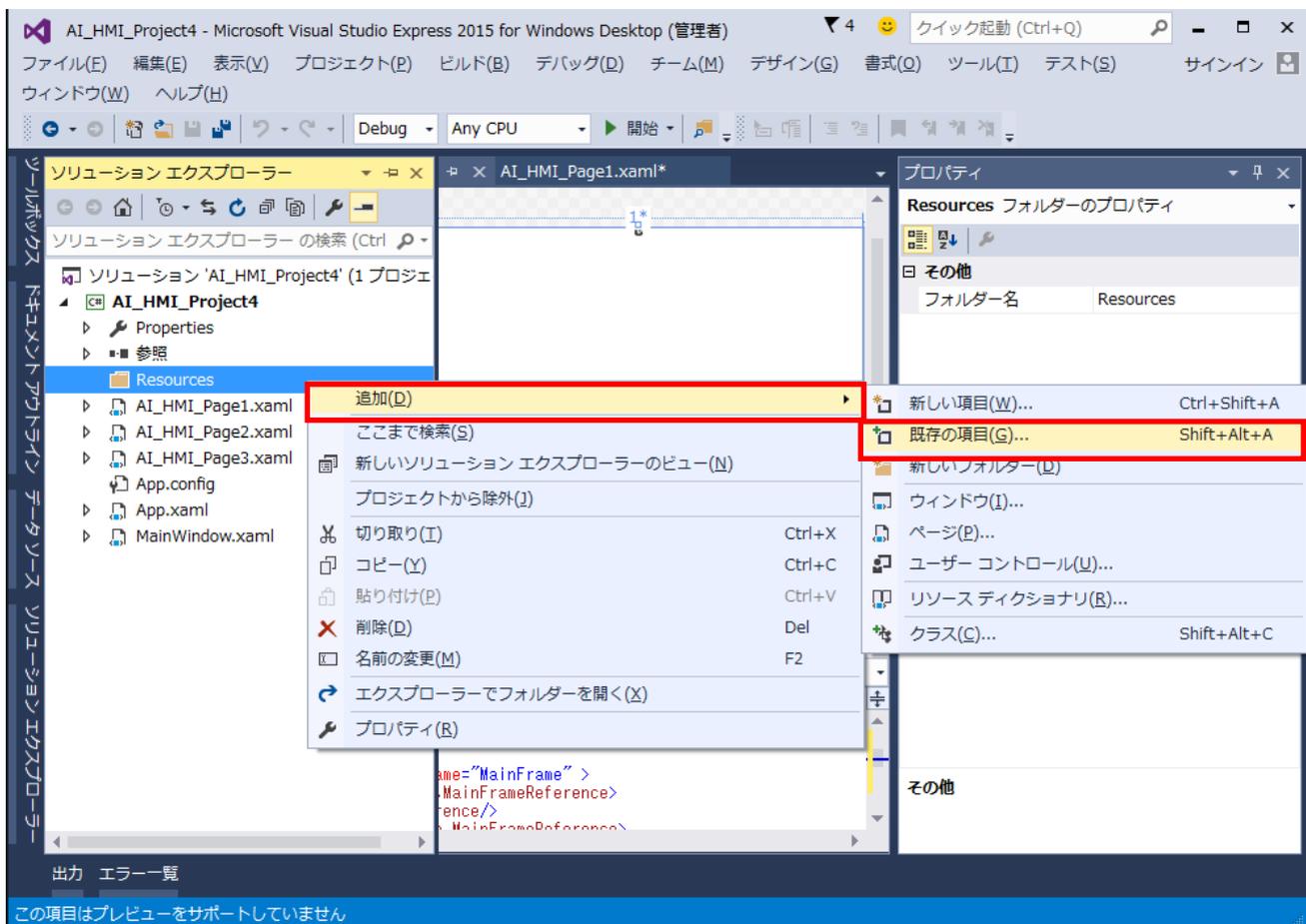


図 17-3-2 既存の項目の表示

- ④ 既存の項目の追加画面が表示されるので、表示するファイルの種類から「イメージファイル」を選択してください。

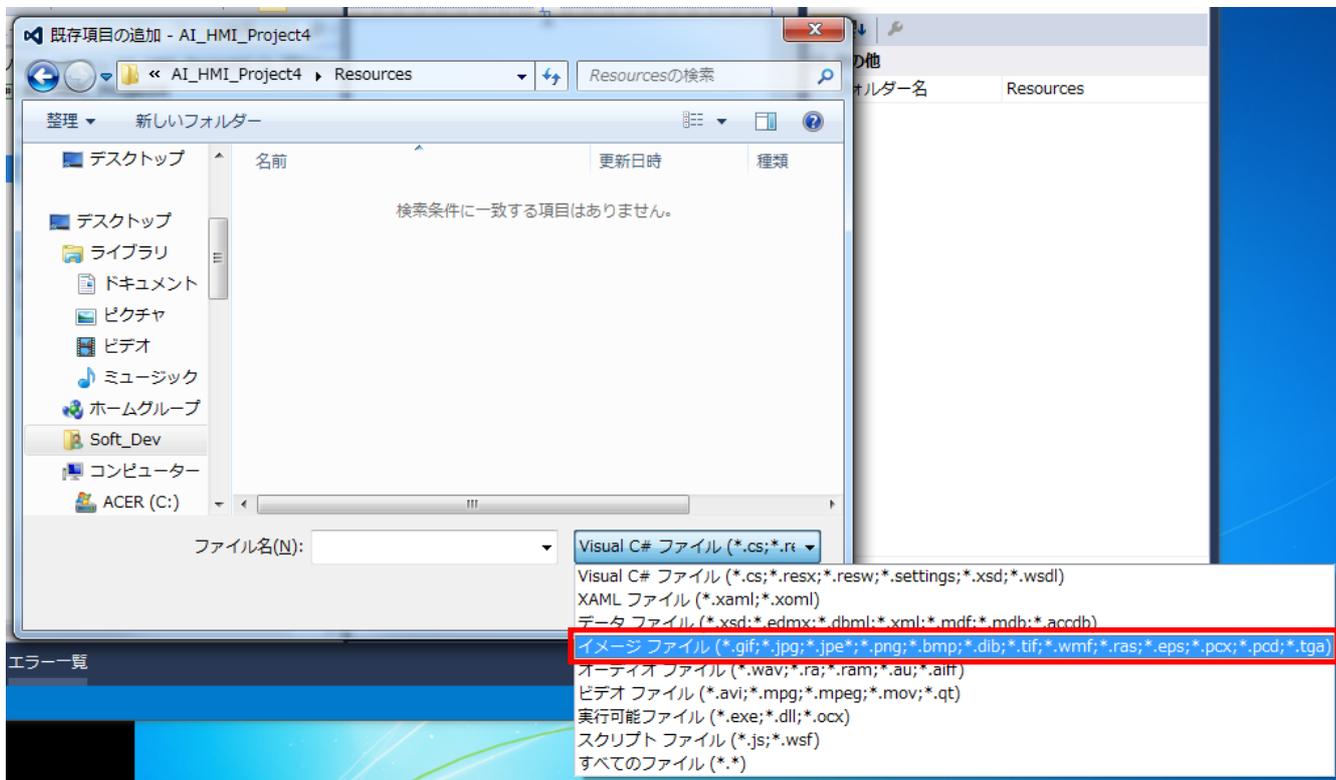


図 17-3-3 イメージファイルの表示

- ⑤ 画像ファイルが表示されるようになるので、登録したいファイルを選択し、「追加」ボタンをクリックしてください。

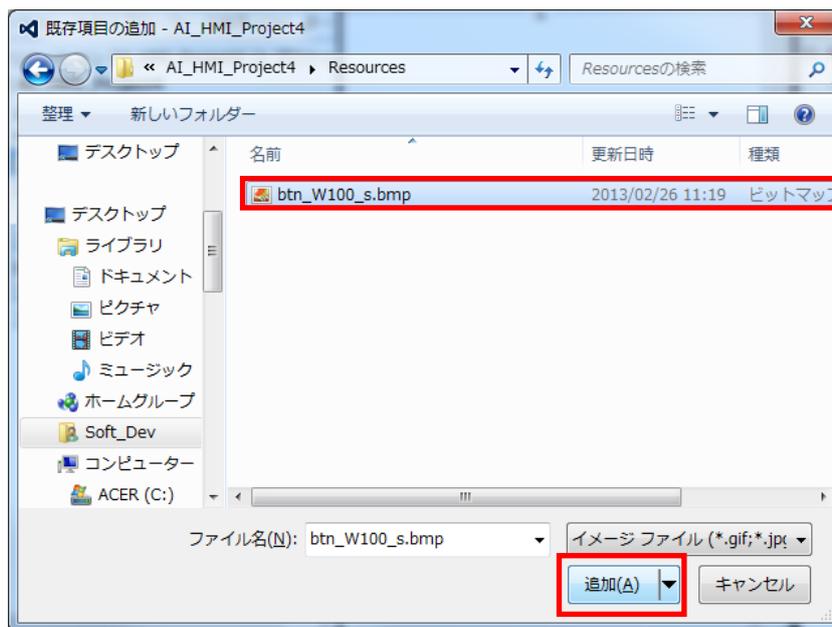


図 17-3-4 画像ファイルの選択

- ⑥ ソリューションエクスプローラの Resource の項目に先ほど選択したファイルが登録されていることを確認してください。

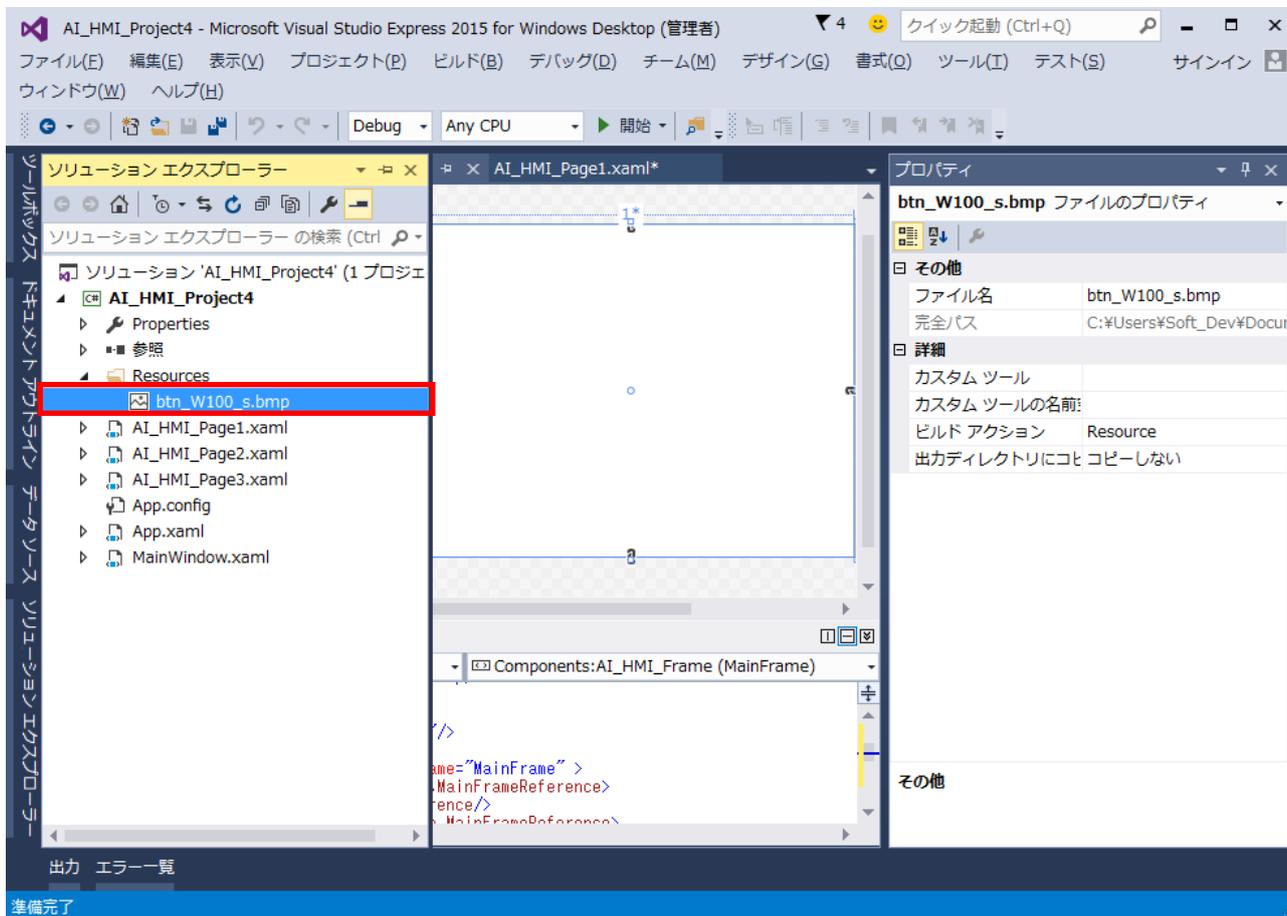


図 17-3-5 画像ファイルの登録の確認

以上で画像の登録は完了です。

『17-2 オブジェクトの画像を変更したい』の手順に従い、ブラシメニューのタイルブラシで登録した画像ファイルが使用できるようになっていることを確認してください。

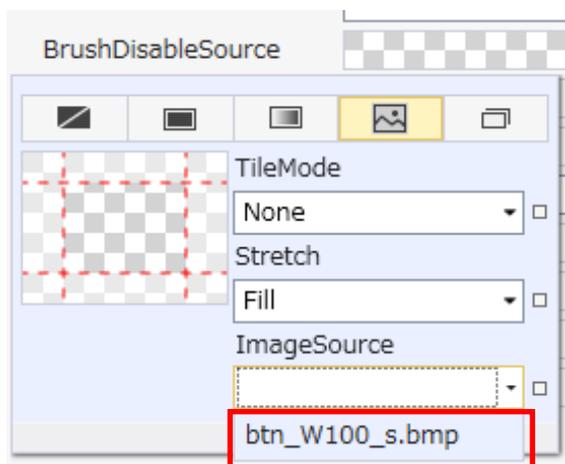


図 17-3-6 リソースの登録の確認

17-4 ページ名がわかりにくいページ名を変更したい

AI HMI Page を新たに登録するとき、デフォルト名として AI_HMI_PageXXX.xaml という名前が自動的に割り振られますが、この名称はユーザで変更することができます。

- ① AI HMI Page を新しく追加する際にファイル名を以下のルールに従って変更してください。
 - ・「3桁の任意の半角英数」+「_」+「任意の文字列」+「.xaml」または「AI_HMI_Page」+「3桁の任意の半角英数」+「.xaml」という構成の名前であること。
 - ・「3桁の任意の半角英数」は0埋めしていてもしていなくてもよい。
 - ・「3桁の任意の半角英数」は他のページのものと同様に重複してはいけない。

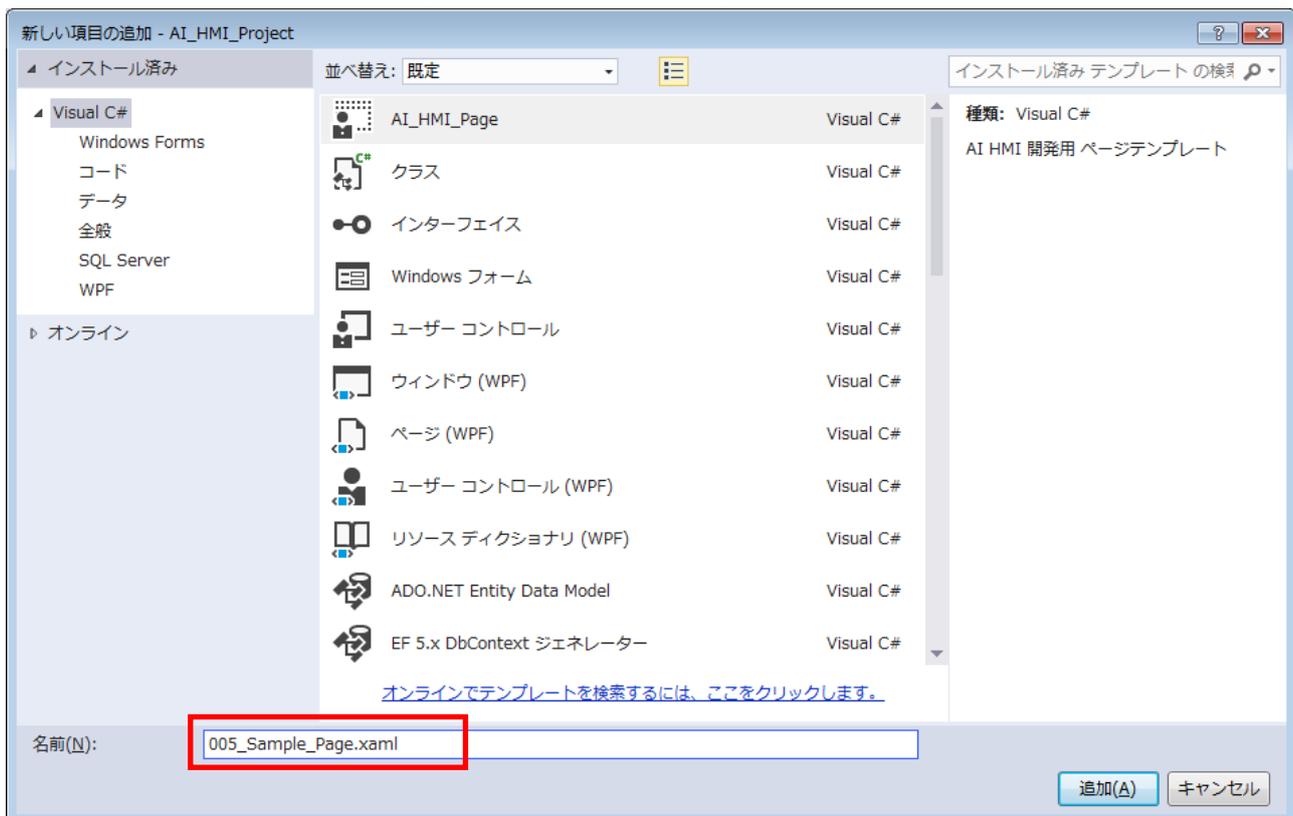


図 17-4-1 任意の名前の AI HMI Page を追加

**※ ページ名は必ずルールに従ったものにしてください。
ルールに従っていないページは AI HMI Page から呼び出し表示することができません。**

よい例

- ・ AI_HMI_Page004.xaml
- ・ 4_Main_Page.xaml
- ・ 004_メインページ.xaml

わるい例

- ・ AI_HMI_Page0004.xaml (任意の半角英数部分は4桁以上にできません)
- ・ O O 4_Main_Page.xaml (任意の半角英数部分は全角文字は使えません)
- ・ メインページ_004.xaml (任意の半角英数部分はページ名の先頭にしてください)

- ② ソリューションエクスプローラにページが新しく追加されていることを確認してください。
新しく追加したページもデフォルト名と同様に AI HMI Frame から呼び出して表示することができます。

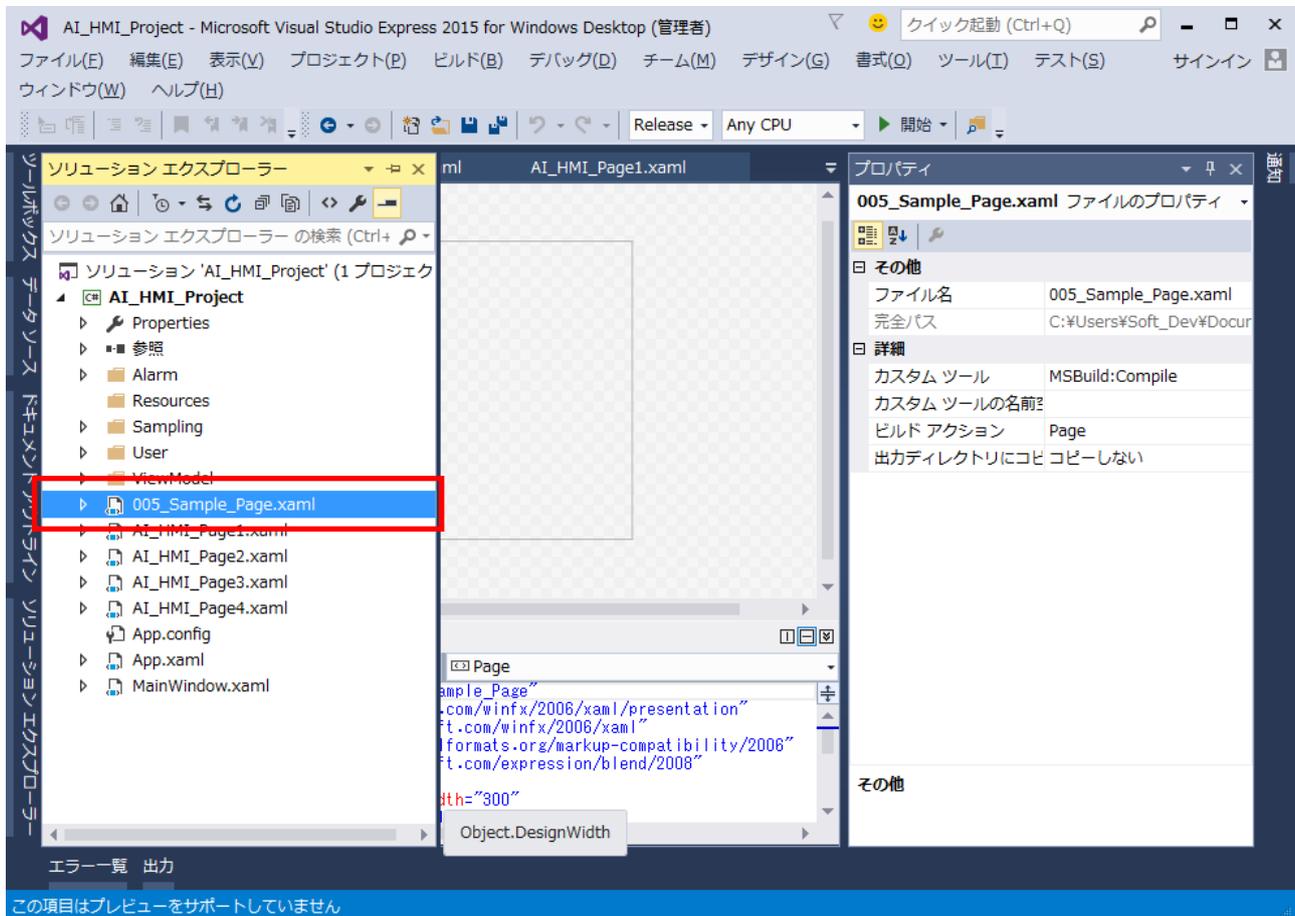


図 17-4-2 任意の名前の AI HMI Page

- ※ ソリューションエクスプローラ上では AI HMI Page の名前を変更しないでください。
変更したページが VisualStudio 上で編集できなくなる場合があります。

17-5 複数の画像ファイルをアニメーション表示したい

AI_HMI_Grid コントロールでは複数の画像ファイルを登録し、それらを周期的に連続して切り替え表示することでアニメーション表示することができます

アニメーション表示機能を使用したい場合、以下の手順に従って作業してください。

※ アニメーション機能で画像ファイルを使用したい場合、あらかじめリソースに登録しておく必要があります。

リソースの登録方法については『17-3 画像ファイルを登録したい』を参照してください。

① AI_HMI_Grid コントロールを配置して選択し、プロパティを表示してください。

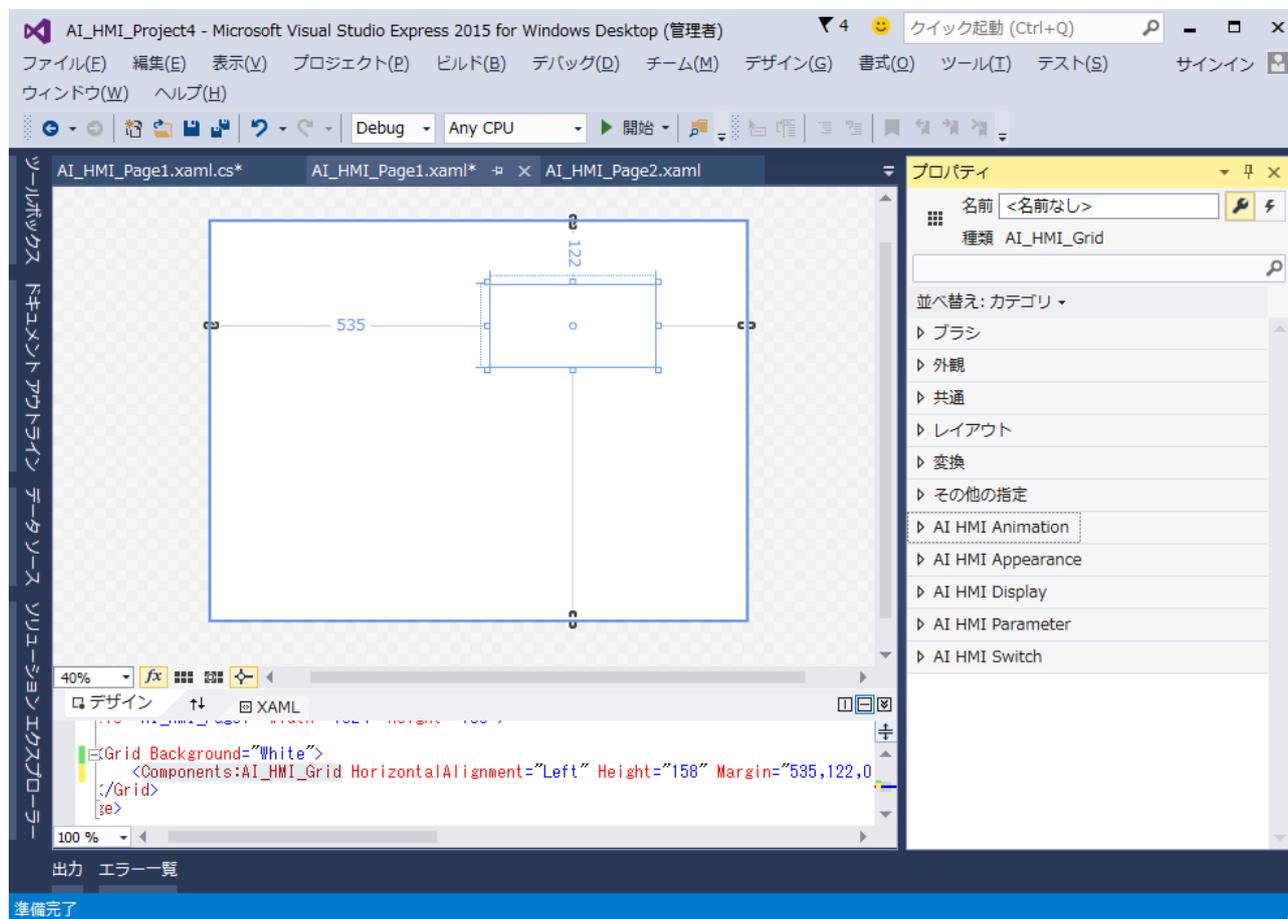


図 17-5-1 AI_HMI_Grid コントロールのプロパティ

- ② AI HMI Display の DisplayImage のコレクションを開きます。
DisplayImage 右側のボタンを押してください。

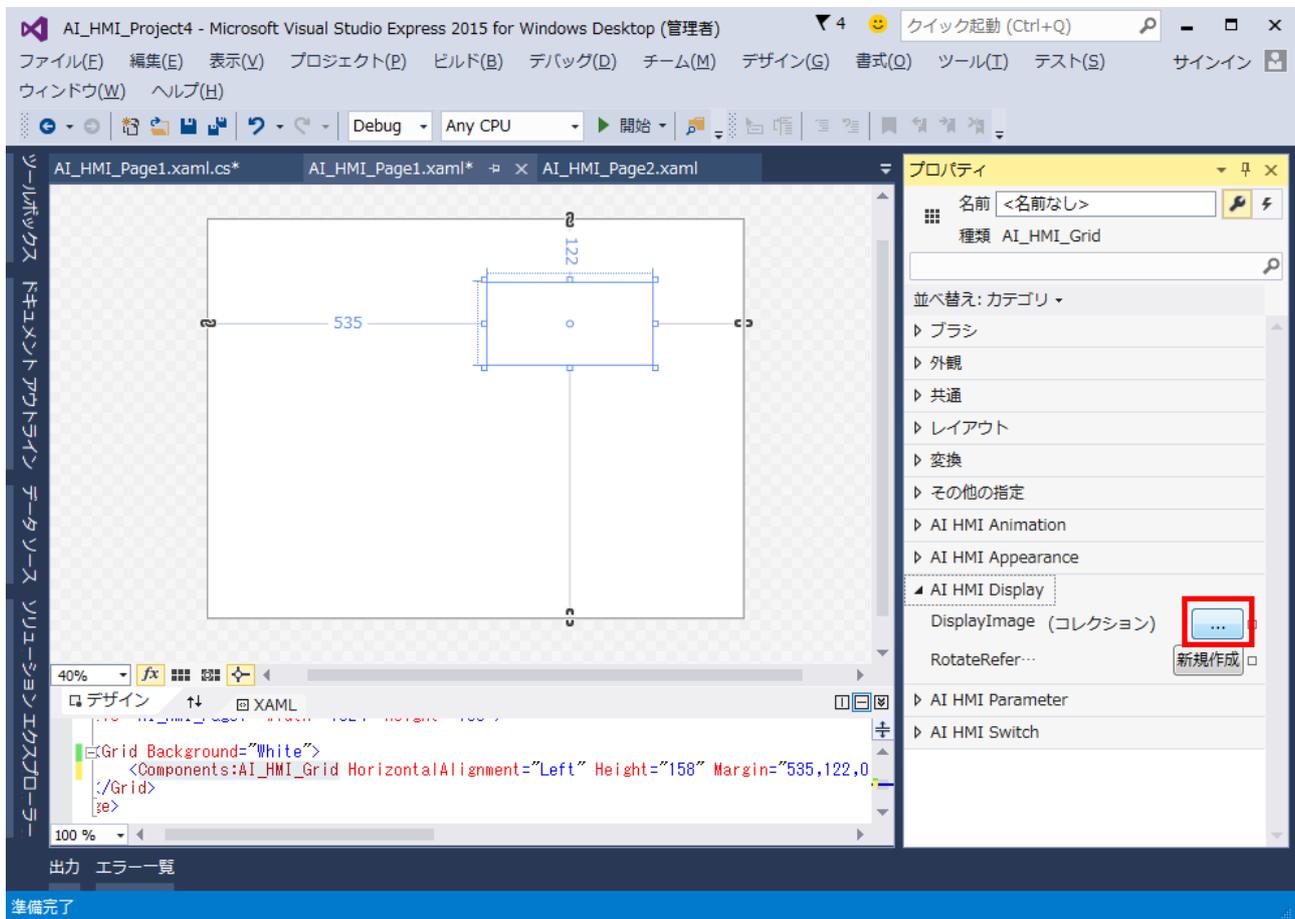


図 17-5-2 AI_HMI_Grid コントロールのプロパティ

- ③ ImageDisplayReference のコレクションエディタが開きます。
「追加」ボタンを押してください。

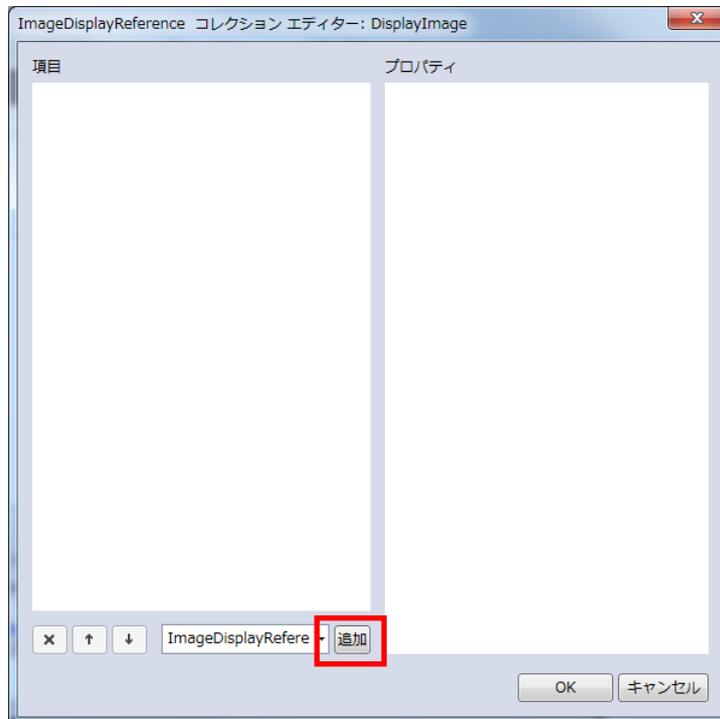


図 17-5-3 ImageDisplayReference コレクションエディタ

- ④ ImageDisplayReference が 1 行追加されます。
「Image」にアニメーションしたい画像の 1 枚目を設定してください。(他の設定をする必要ありません。)
画像の設定方法は『17-2 オブジェクトの画像を変更したい』を参照してください。

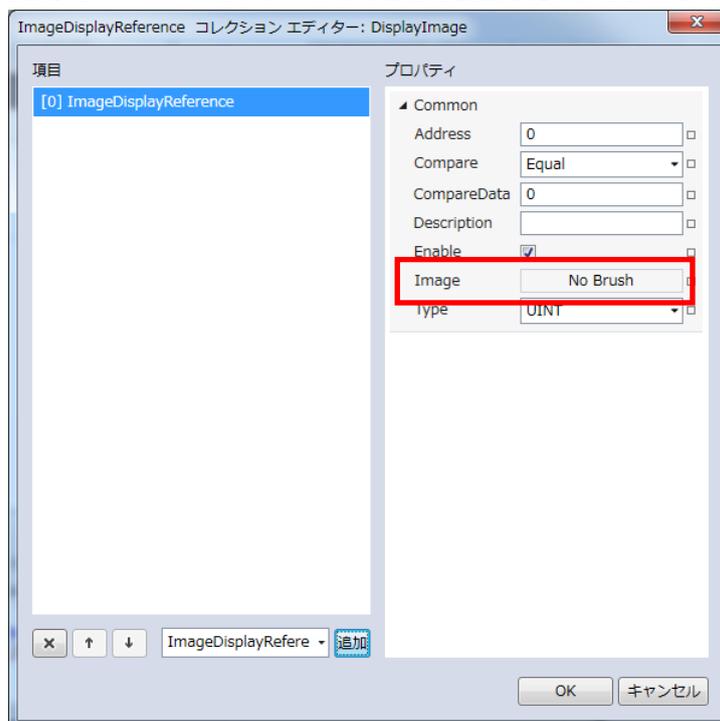


図 17-5-4 画像の設定

- ⑤ 画像が登録されます。
手順③と④を繰り返して、アニメーションに使用する画像を 1 行ずつ設定してください。

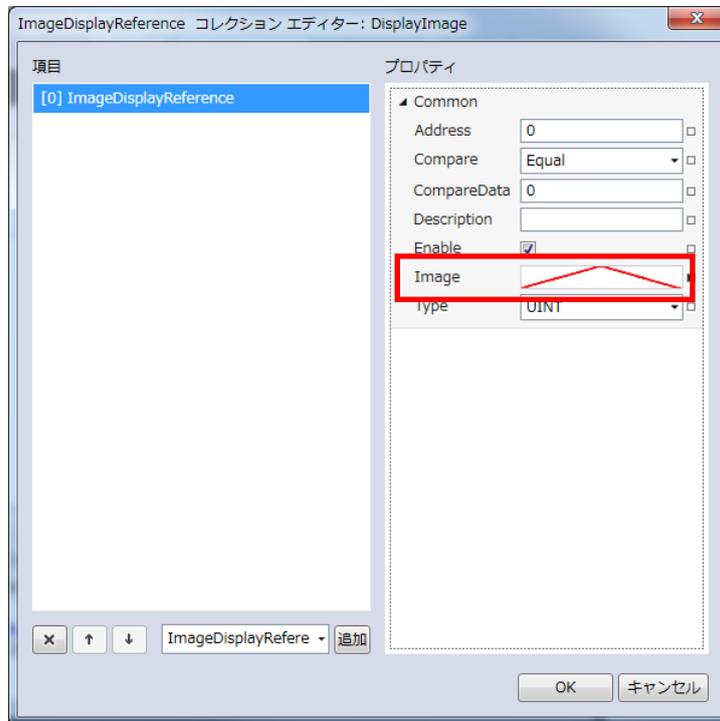


図 17-5-5 画像の設定

- ⑥ 全ての画像の設定が完了したら「OK」を押してください。

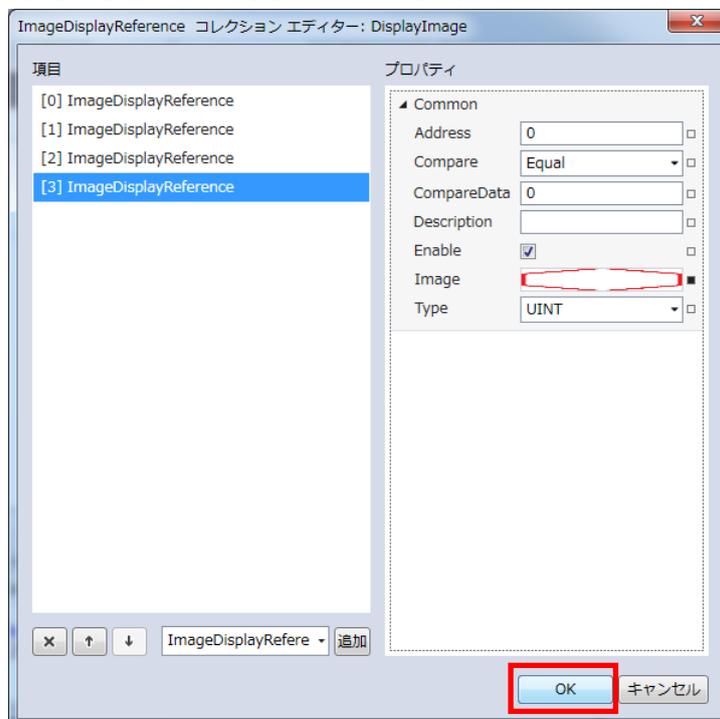


図 17-5-6 画像の設定の完了

- ⑦ 設定した画像をアニメーションさせます。
AI HMI Animation の Animation のプルダウンメニューから None 以外の項目を選択してください。
(Slow、Medium、High のいずれか)

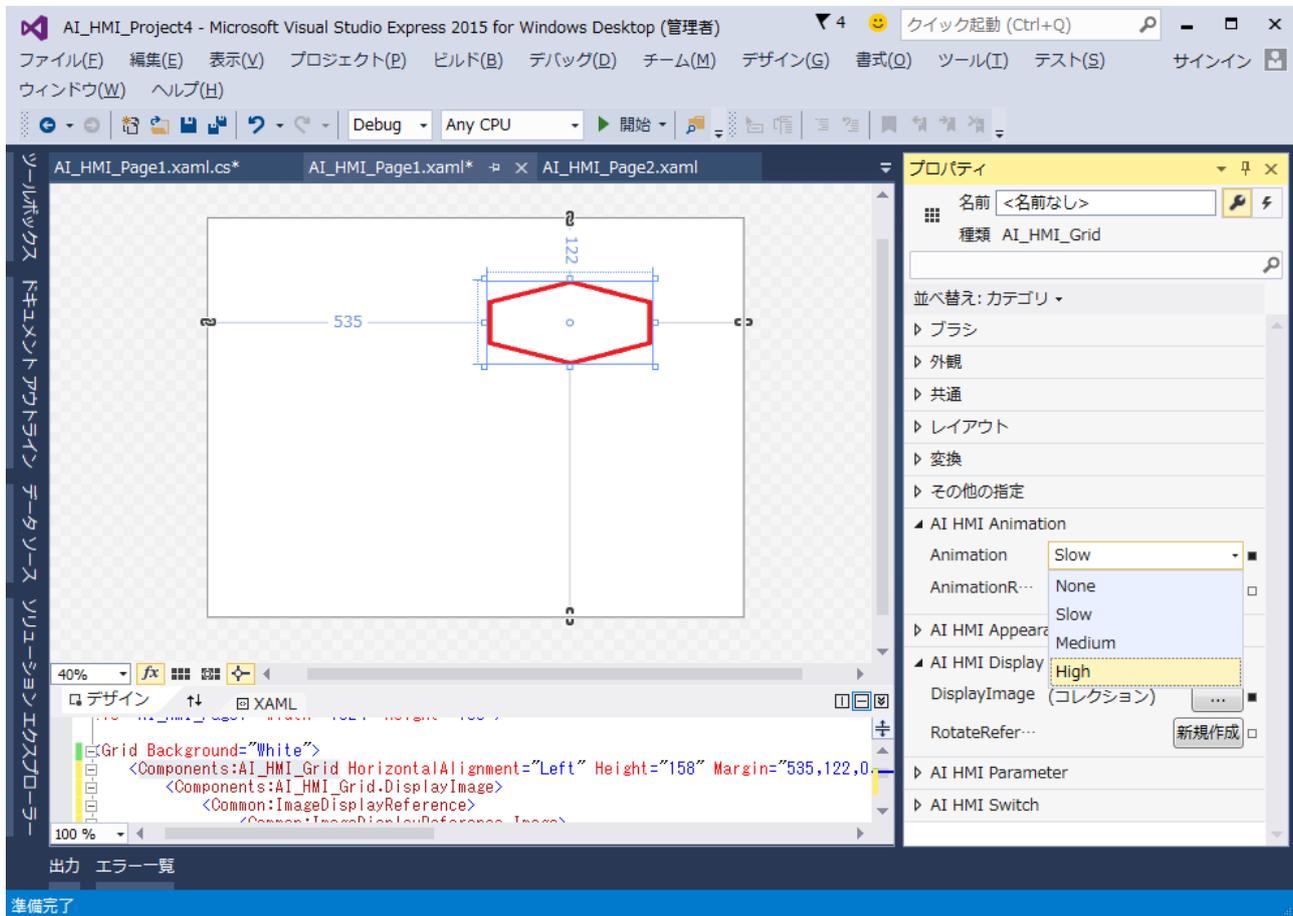


図 17-5-7 アニメーションの設定

以上でアニメーションの設定は完了です。
プロジェクトのコンパイルを行い、正常にアニメーションが表示されるか確認してください。

17-6 画面に配置した部品をダブルクリックしてしまった

VisualStudio のエディタ画面でコントロールをダブルクリックすると Click イベントが発行され、ソース上にイベントが残ってしまいます。

このイベントの記述が残っていても AI-HMI の動作自体に影響は起こりません。

発行された Click イベントを削除したい場合は以下の手順に従って作業してください。

ここでは AI_HMI_Page1.xaml 上で AI_HMI_Button をダブルクリックしてしまったものとして記述します。

- ① AI_HMI_Page1.xaml.cs を開いてください。

ai_HMI_Button_Click というイベント関数が作成されていることを確認してください。

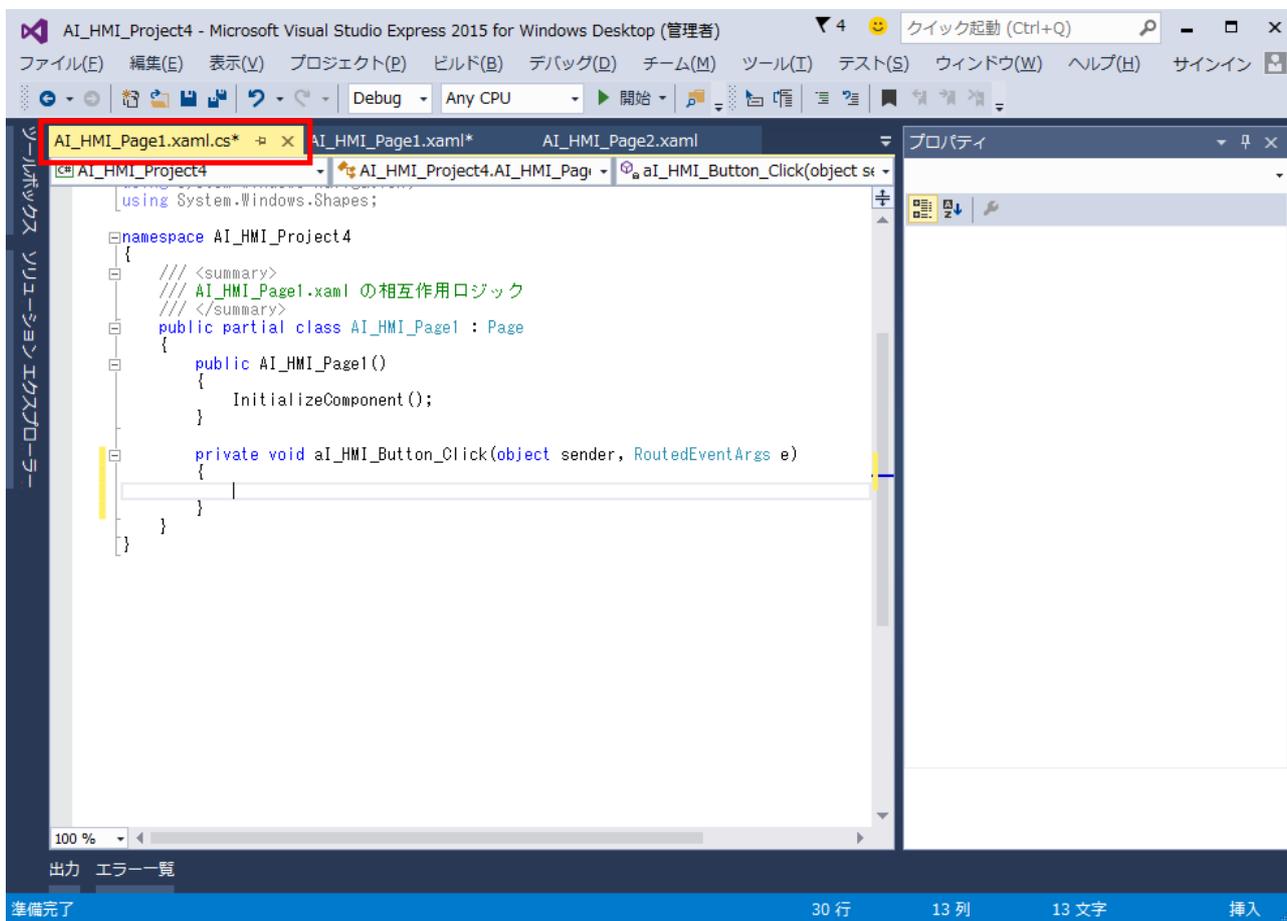


図 17-6-1 Click イベントのソースの表示

- ② ai_HMI_Button_Click を削除してください。
ai_HMI_Button_Click の関数部分(図 17-6-2 で選択している範囲)を削除してください。

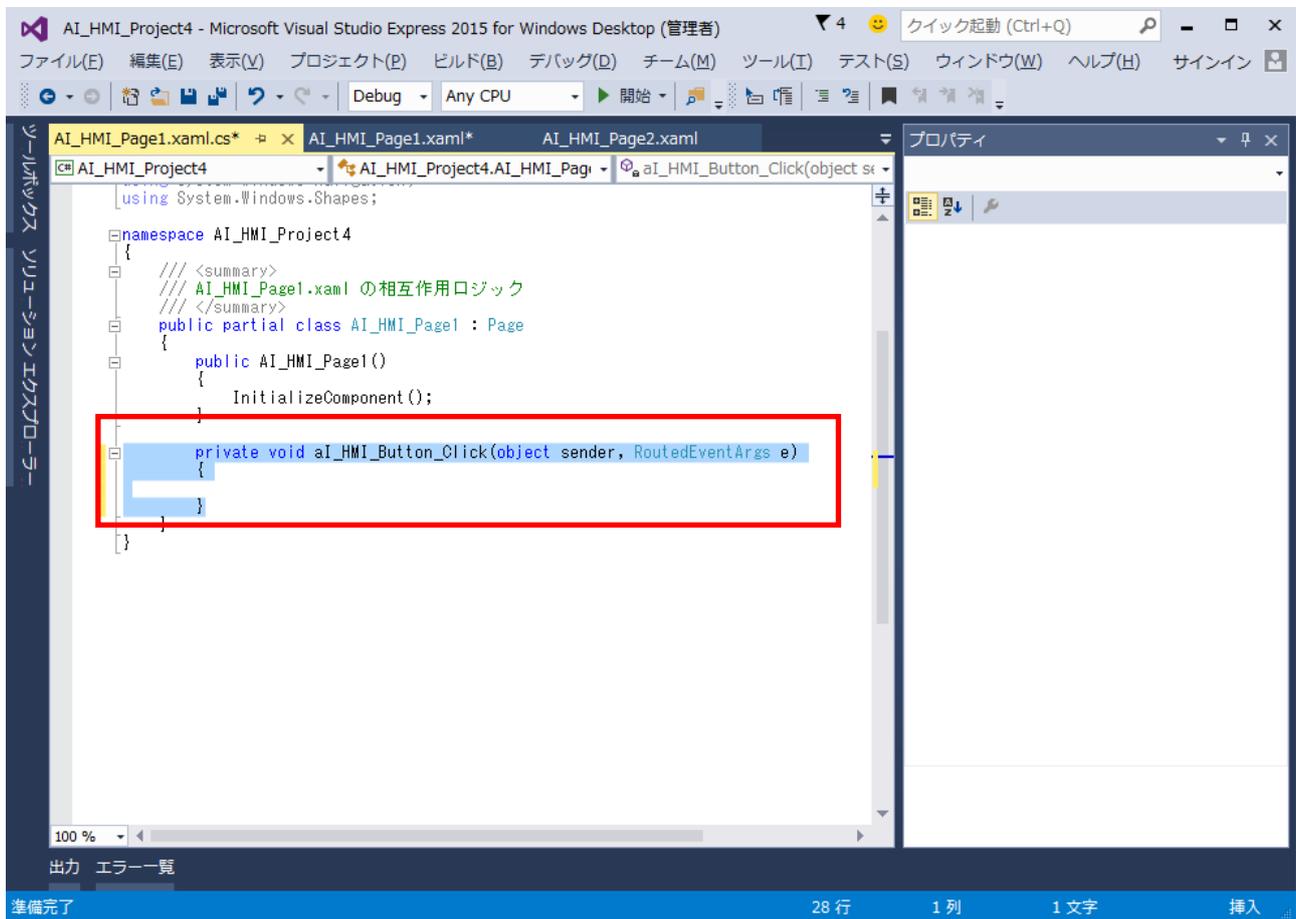


図 17-6-2 Click イベントのソースの表示

- ③ コントロール側の Click イベントを削除します。

AI_HMI_Page1.xaml を開き、AI_HMI_Button を選択した状態で「選択した要素のイベントハンドラー」を選択してください。

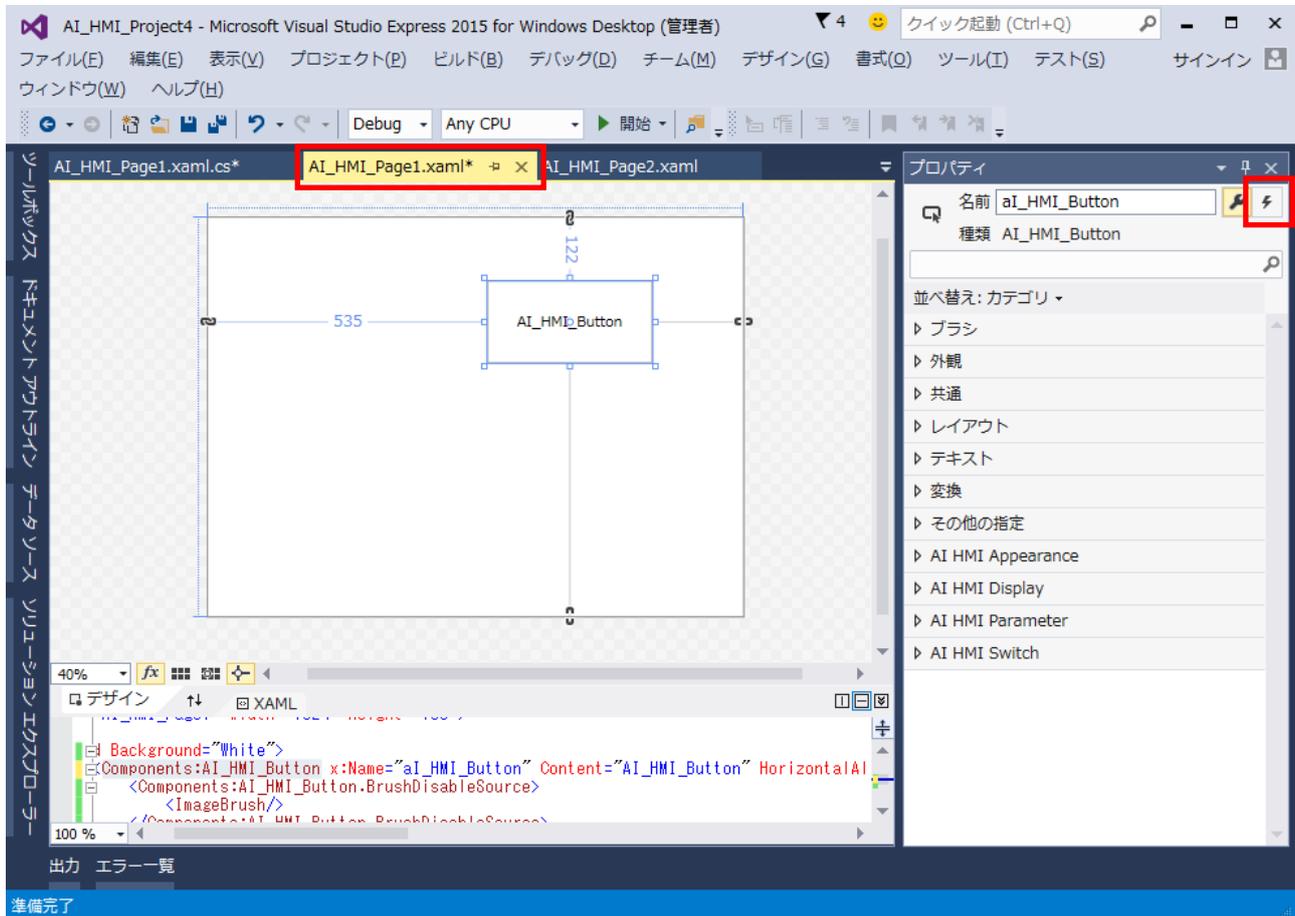


図 17-6-3 イベントハンドラーの表示

- ④ イベントハンドラ画面が開きます。
「Click」の項目の「ai_HMI_Button_Click」を削除してください。

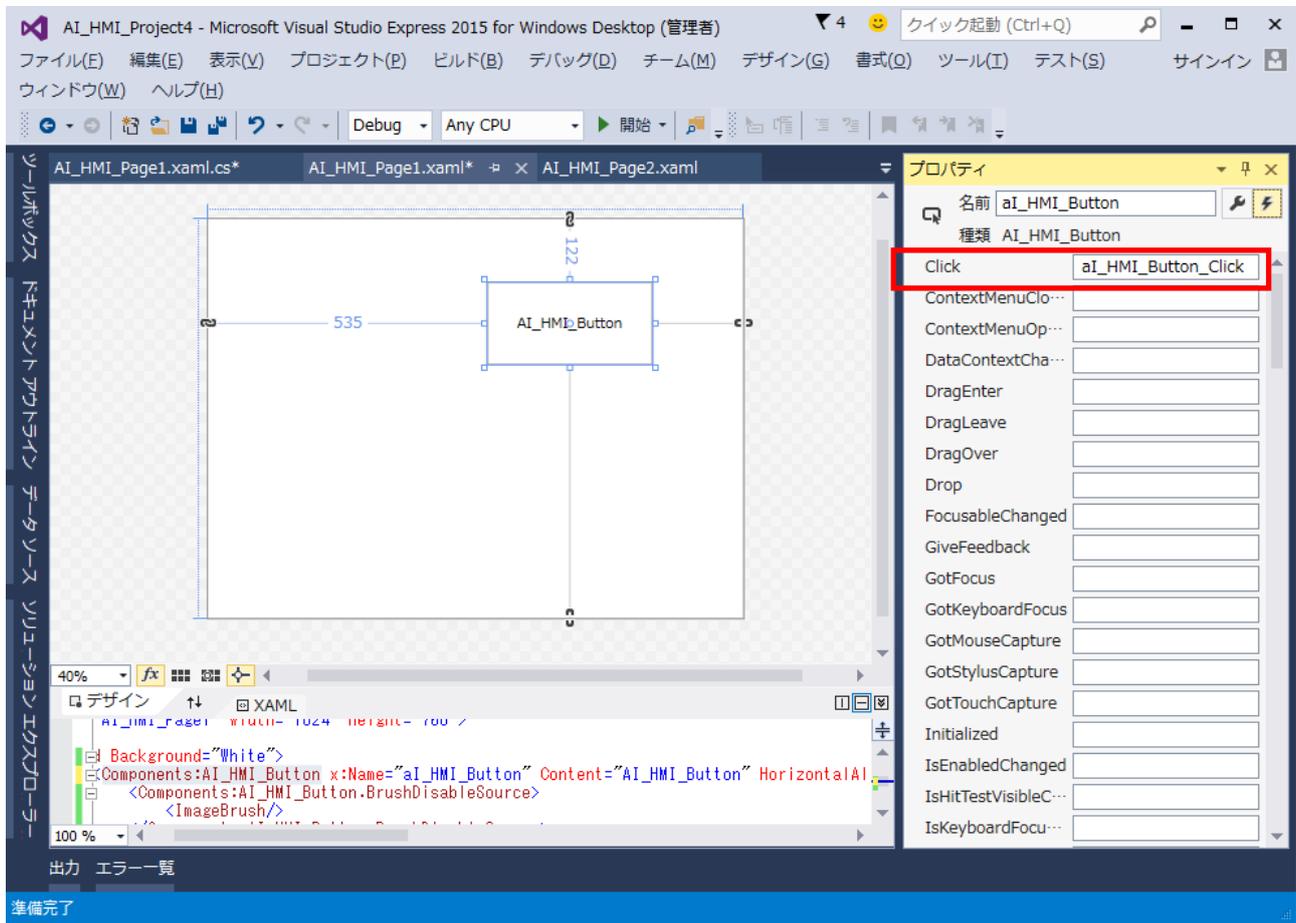


図 17-6-4 Click イベントの削除

- ⑤ プロジェクトをコンパイルします。
上部メニューの「ビルド - ソリューションのビルド」を選択してください。

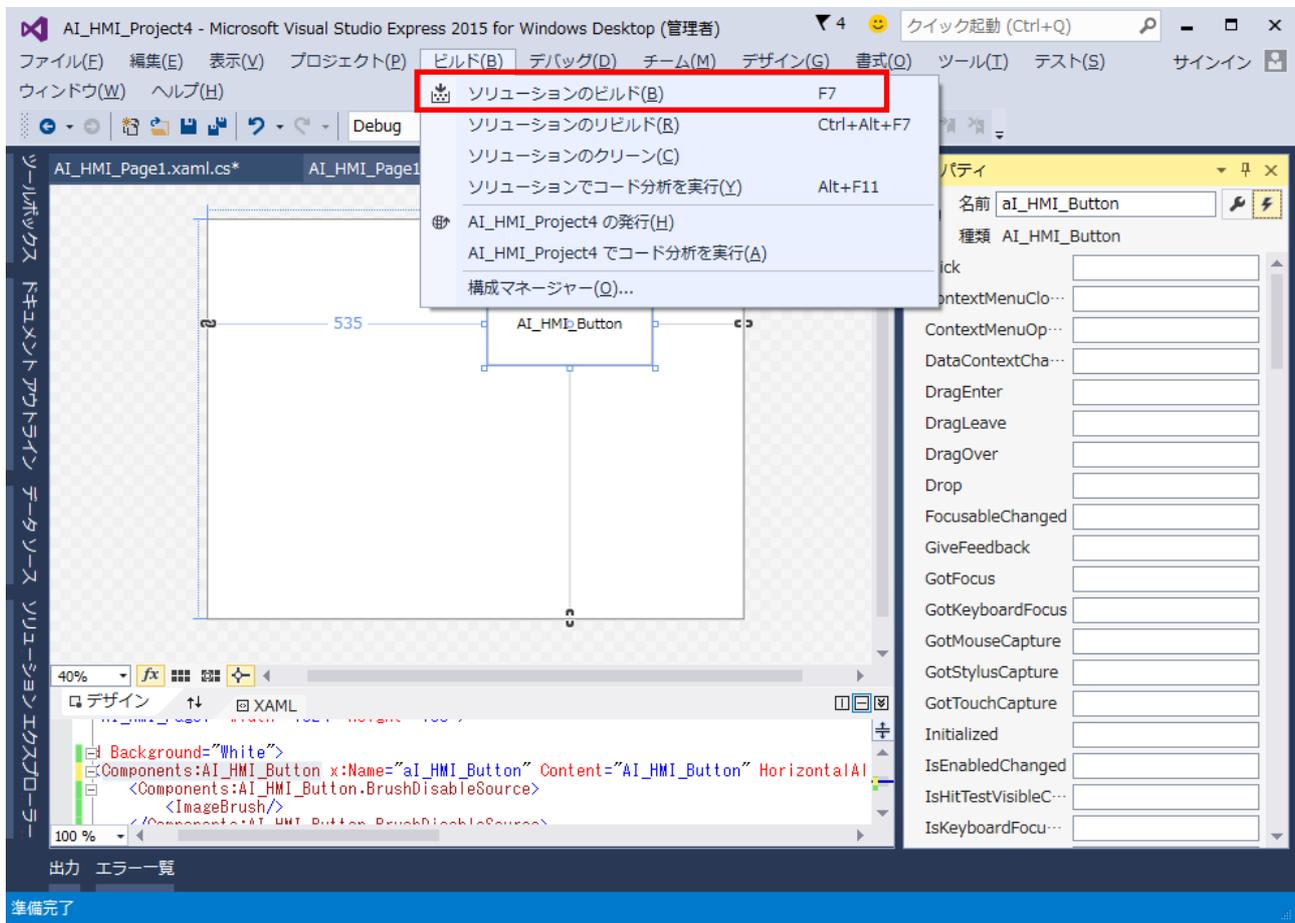


図 17-6-5 プロジェクトの再コンパイル

以上で修正作業は完了です。
コンパイルが完了したら、引き続き元の作業を続けることが可能です。

17-7 音声ファイルをすべての画面で再生したい

AI HMI Media コンポーネントを画面に配置することで動画ファイルや音声ファイルを再生することができますが、それらの再生は配置した画面上でのみ有効となります。

同一の音声ファイルをどの画面を表示していても再生するには以下の手順で登録してください。

- ① MainWindow.xaml を開く。

ソリューションエクスプローラから MainWindow.xaml をダブルクリックして開いてください。

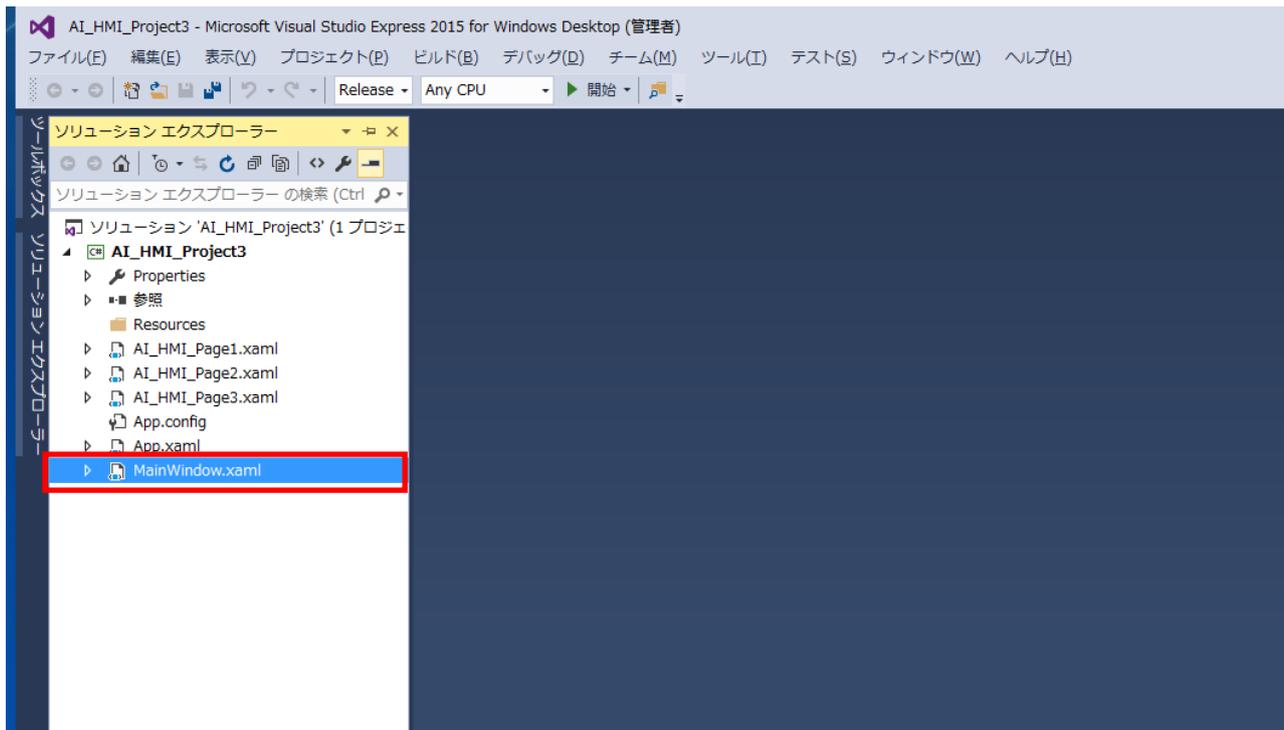


図 17-7-1 MainWindow.xaml

- ② AI_HMI_Media コンポーネントを配置し、音声ファイルを再生するように設定する。
AI HMI Media コンポーネントの設定方法は AI HMI リファレンスマニュアルの『第 18 章 AI HMI Media』を参照ください。

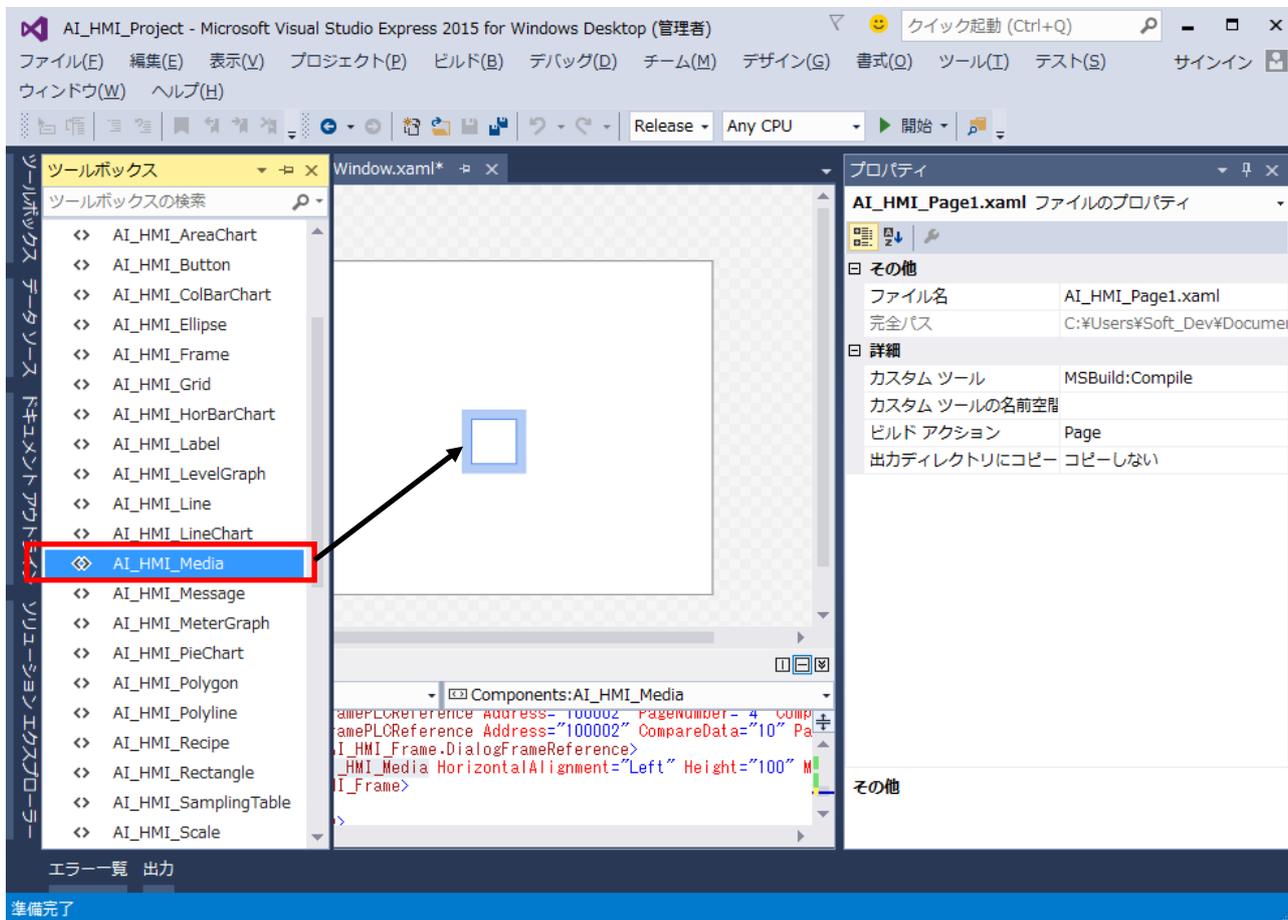


図 17-7-2 MainWindow.xaml

※ 同様の方法で動画ファイルも全画面再生にすることは可能ですが、動画の映像部分は他画面に隠れてしまうため、実質的に動画の音声のみが再生される形になります。

17-8 Windows 起動時に AI-HMI アプリケーションを起動するようにしたい

AI-HMI のアプリケーションを Windows 起動時に自動的に実行するための設定について説明します。

- ① AI-HMI アプリケーションの実行ファイル (XXXX.exe) を右クリックし、「ショートカットの作成」を選択してください。

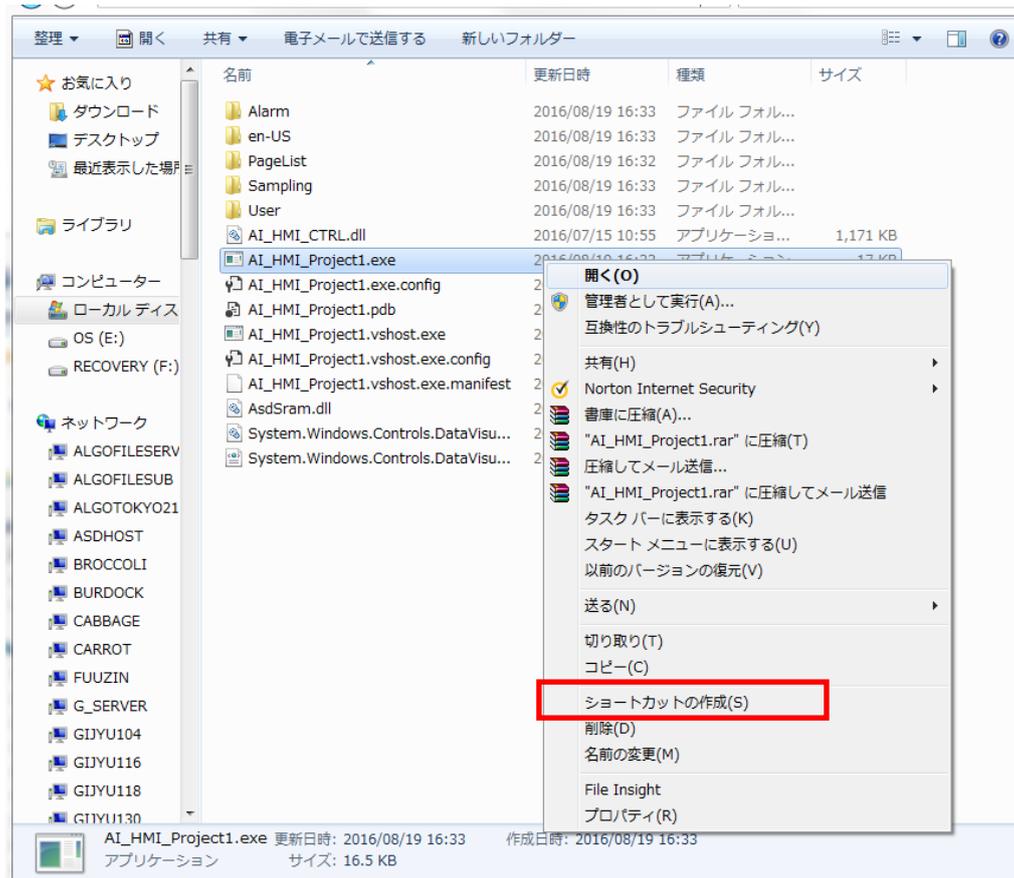


図 17-8-1 ショートカットの作成

- ② Windows のスタートメニューを開き、「すべてのプログラム – スタートアップ」を右クリックして「開く」を選択してください。
スタートアップフォルダが開くので、手順①で作成したショートカットをコピーしてください。

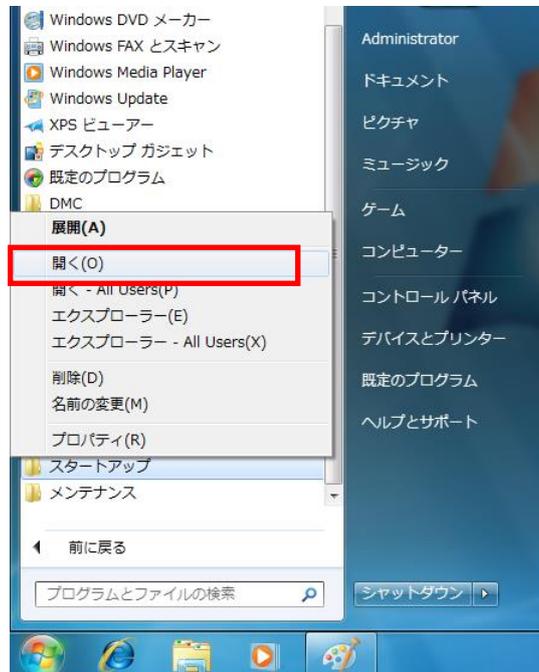


図 17-8-2 スタートアップフォルダを開く

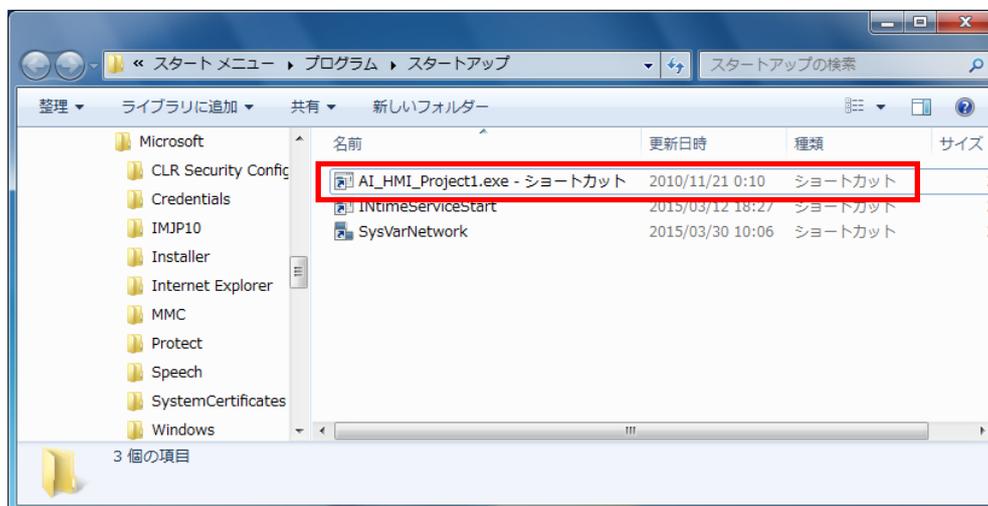


図 17-8-3 ショートカットの登録

- ③ IfSharedMemory.ini ファイルを編集します。
 スタートメニューにショートカットをコピーした時点で Windows 起動時にアプリケーションが起動するようになりますが、このままでは共有メモリが Intime と別々に生成されてしまうため、共有メモリの設定を変更する必要があります。
 以下の ini ファイルを編集します。
 C:¥Windows¥IfSharedMemory.ini
 IfSharedMemory.ini の内容の詳細は以下のようになります。(エンコード形式 S-JIS)

リスト 17-8-1 IfSharedMemory.ini の内容

```
[Config]
UsedMemName=HMIM
UsedMemType=1
UsedMemSize=112000
OpenSeqType=2
OpenTimeout=60
ProConOS_Offset=68000
```

表 17-8-1 IfSharedMemory.ini の内容詳細

項目	初期値	内容
UsedMemName	HMIM	共有メモリ名称です。 Intime の共有メモリ名に合わせる必要があります。 ※ 変更しないでください。
UsedMemType	1	使用する共有メモリ種別です。 1: Intime 共有メモリ (基本動作) 2: Windows メモリマップトファイル ※ 変更しないでください。
UsedMemSize	112000	共有メモリサイズです。 ※ 変更しないでください。
OpenSeqType	2	アプリケーション起動時に Intime 共有メモリが無かった場合の動作です。 (UsedMemType が 1 の場合のみ有効) 0: エラー応答 (基本動作) 1: 共有メモリが生成されるまで起動待機 2: 共有メモリに変わり、Windows マップトファイルを Open して応答
OpenTimeout	60	Open タイムアウト時間です。 (OpenSeqType=1 の場合のみ有効) 0: タイムアウト無し 10~: タイムアウト時間 (秒)
ProConOS_Offset	68000	MULTIPROG で指定するアドレスオフセットです。 ※ 変更しないでください。

アプリケーションを自動起動するため、以下のように設定を変更してください。

表 17-8-2 IfSharedMemory.ini 変更内容

項目	設定値	内容
OpenSeqType	1	Intime の共有メモリが生成されるまでアプリケーションの起動を待つように設定します。
OpenTimeout	120	Intime の共有メモリが生成されるまで待機するタイムアウト時間を 120 秒に設定します。

- ④ 正常に設定されたことを確認するため、Windows を再起動してください。
Windows の起動後、しばらくすると AI-HMI アプリケーションが自動で起動してくることを確認してください。
また、正常に共有メモリの読み書きができていることを確認してください。

17-9 タッチパネルでボタンをタッチしたときにマウスと同じ挙動にしたい

実行環境 PC は「Touch Mode」と「Mouse Mode」の 2 種類のタッチモードを使用できます。
それぞれの特徴は以下の通りです。

表 17-9-1 タッチモードの違い

	特徴	メリット	デメリット
Touch Mode (デフォルト)	タッチ操作がタブレット PC 等と同等の挙動になります。 ボタンなどはタッチした時点では動作せず、タッチを離れた時点で動作します。	タッチ長押しでマウスの右クリックと同等の挙動になるため、マウスなしでもデバッグをしやすい。	ボタンを押したときの描写がわかりにくい。
Mouse Mode	タッチ操作がマウスの左クリックとほぼ同等の挙動になります。 ボタンはタッチした時点で押した挙動になり、タッチを離れた時点で動作しません。	AH-HMI 等のボタンを押したとき、マウス操作と同様の感覚で行える。	右クリック操作ができないため、デバッグに支障が出る。(右クリック支援用のツールはある)

Touch Mode はボタン操作などにおいて通常のマウスクリックとは勝手が異なります。
Mouse Mode に変更することで通常のマウスクリックと同じ感覚で操作ができます。
Touch Mode と Mouse Mode の切り替えは以下の手順で行ってください。

1. コントロールパネルを開き、「ASD Config」を実行する。

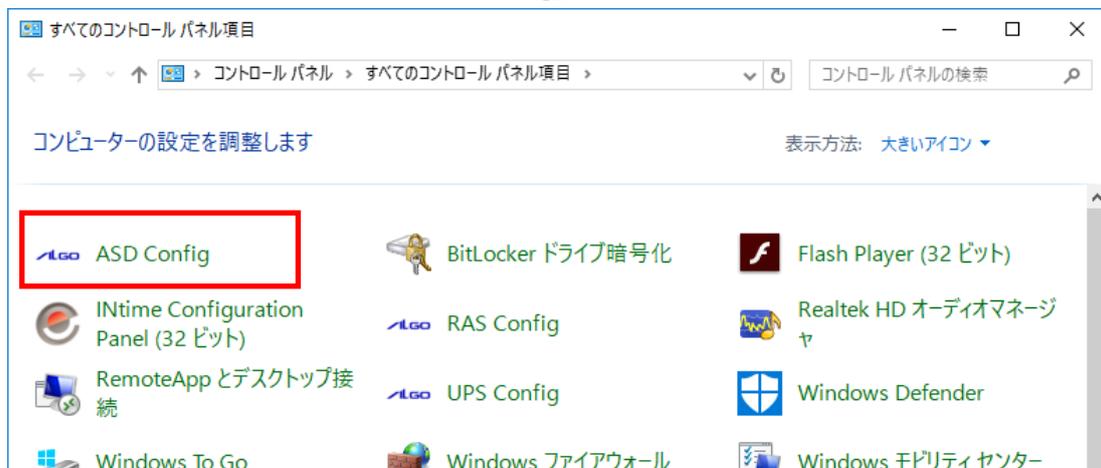


図 17-9-1 ASD Config の実行

2. 「Touch Panel Setting」タブを開く。
3. 「Touch Mode」または「Mouse Mode」を選択して「OK」ボタンをクリックする。

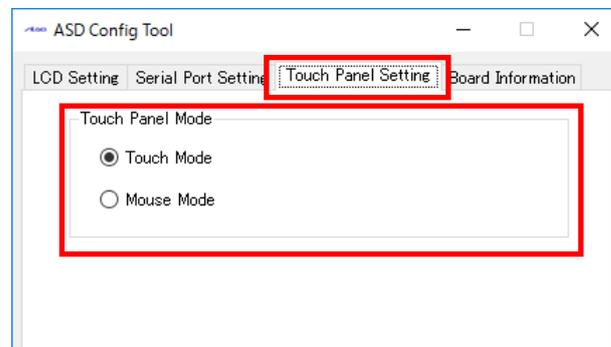


図 17-9-2 Touch Panel Mode の変更

デフォルト状態では Touch Mode に設定されていますが、この状態では AI HMI のボタン操作において、実際にボタンが押されているかの判断が難しい状態にあります。
そのため、AI-HMI で作成した画面アプリを運用する場合の設定は Mouse Mode を推奨させていただいております。

17-10 Mouse Mode で右クリック操作をしたい

「Mouse Mode」の状態では通常、マウスの右クリックと同様の操作を行うことができませんが、同様の操作をするための補助ツールが用意されています。以下の手順に従い、補助ツールを起動して使用してください。

1. Windows のスタートメニューから「すべてのプログラム - USB TouchPanel Setting」を実行する。

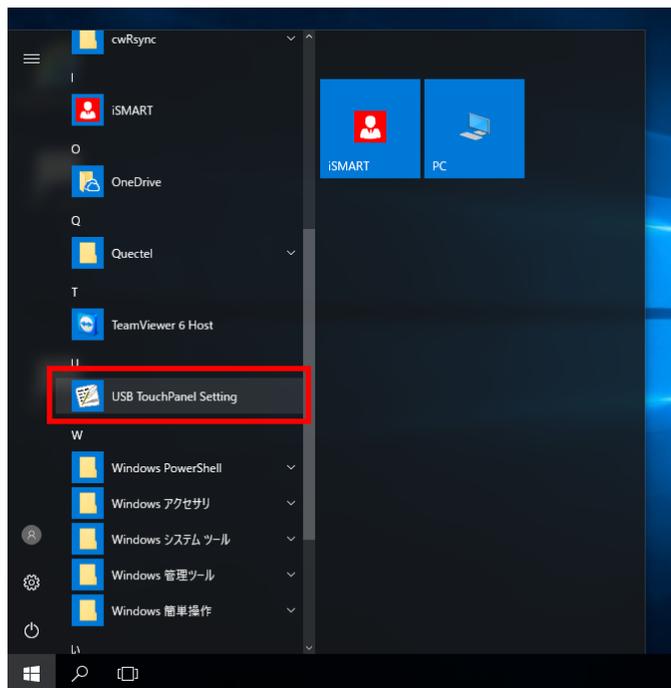


図 17-10-1 USB TouchPanel Setting の起動

2. 「右クリック設定」タブを開く。



図 17-10-2 USB TouchPanel Setting

3. 「」または「」にチェックを入れて「OK」ボタンをクリックする。

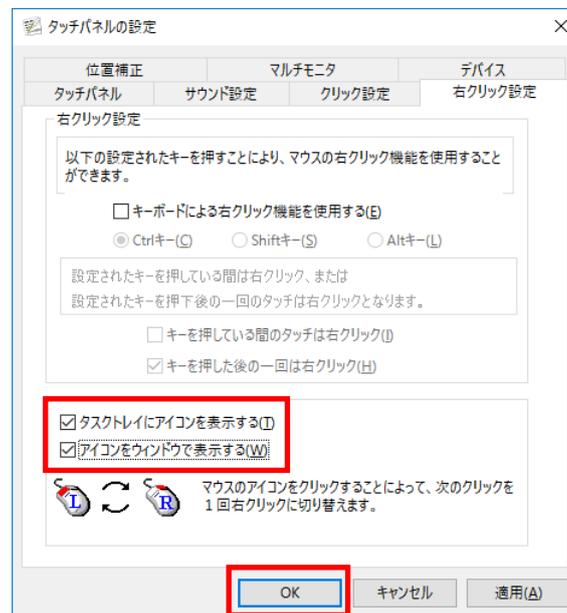


図 17-10-3 右クリック設定タブ

この操作を行うことで通常の画面または Windows タスクバー上に右クリック補助用のアイコンが表示されるようになります。

このアイコンをタッチすることで次のタッチを右クリックとして扱うようになります。

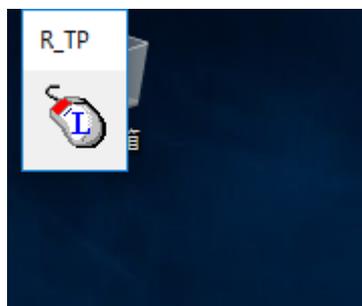


図 17-10-4 右クリック補助用アイコン

17-11 タッチポイントを非表示にしたい

タッチパネル画面をタッチしたとき、タッチポイントが画面に表示されます。

このタッチポイントを非表示にしたい場合は、以下の手順に従って AI-HMI 側の設定を変更してください。

- ① AI-HMI のプロジェクト上で「ソリューションエクスプローラー」を開き、「MainWindow.xaml」をダブルクリックしてください。

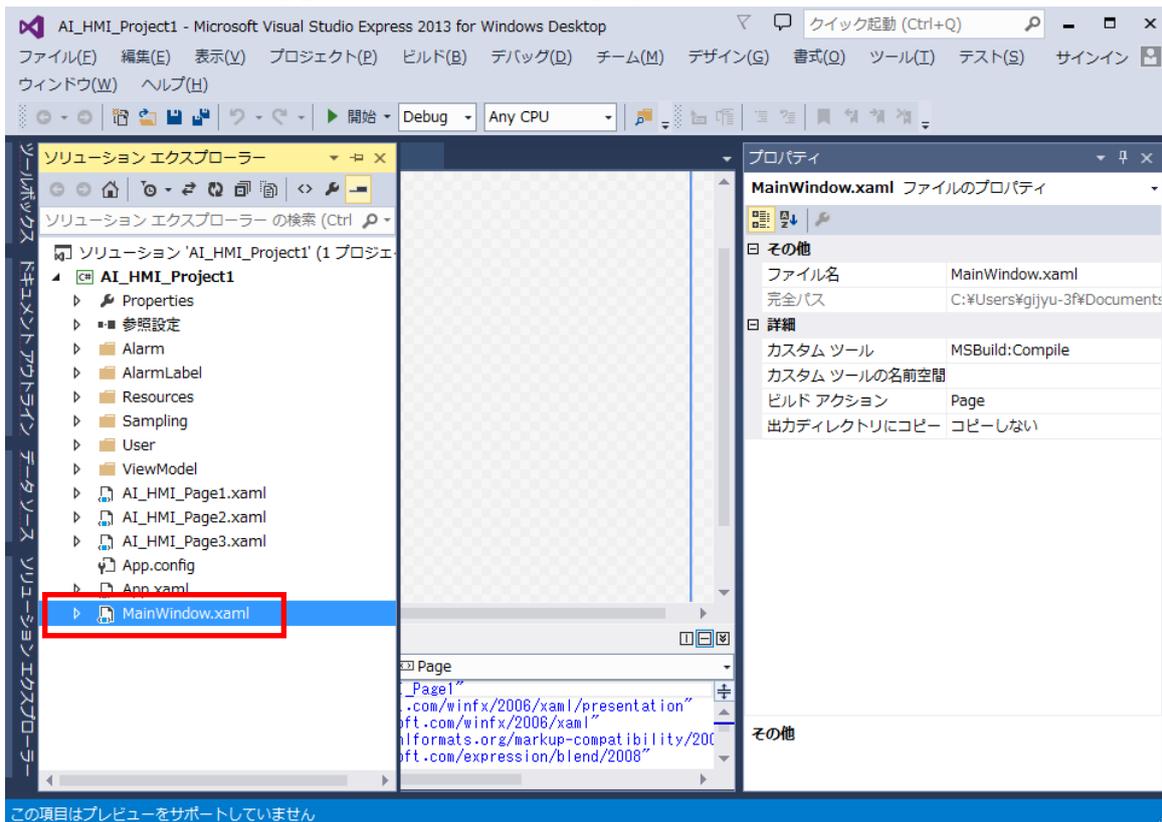


図 17-11-1 MainWindow の表示

② マウスの表示設定を変更します。

背景部分をクリックし、表 17-11-1 のようにプロパティを変更してください。

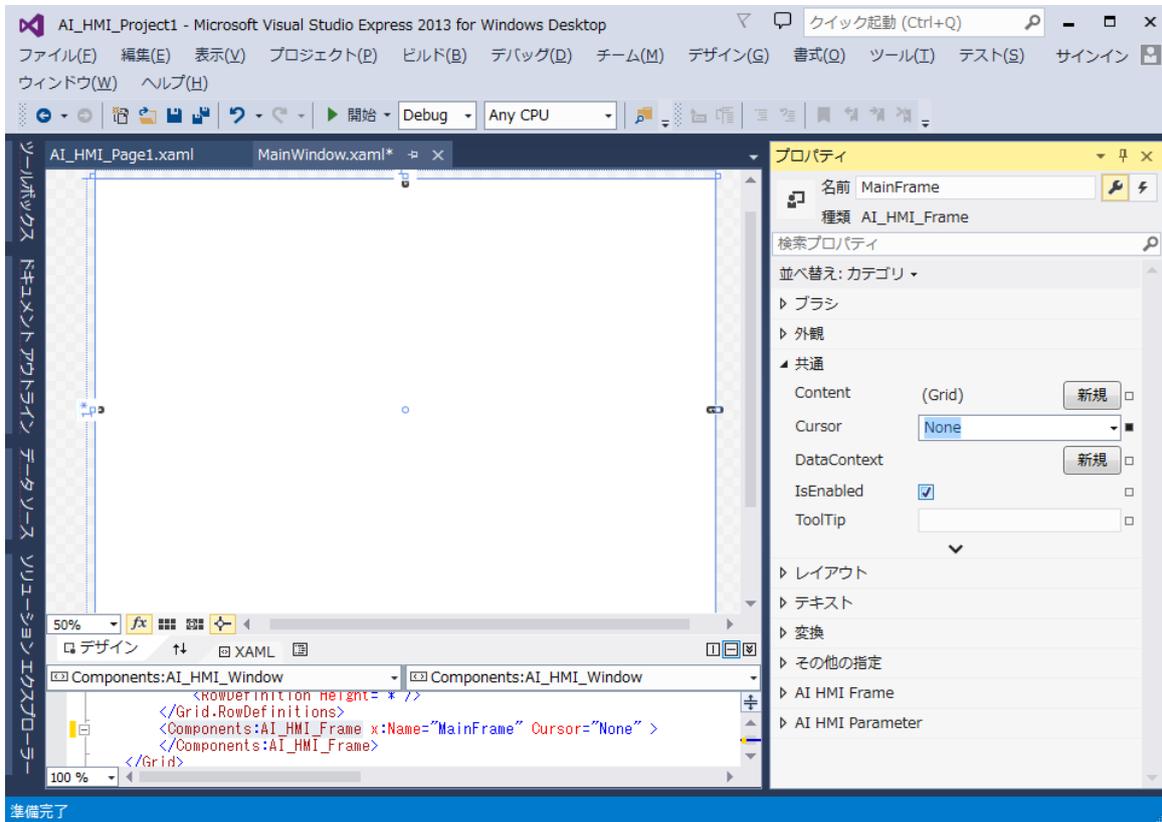


図 17-11-2 マウスカーソル表示のプロパティの変更

表 17-11-1 マウスカーソル表示のプロパティ

項目	設定値	備考	
共通	Cursor	None	マウスカーソルを非表示にします。
その他の指定	ForceCursor	チェックあり	子オブジェクトすべてにマウスカーソルが非表示になるよう設定します。

③ 作成した画面を実際に動作させてマウスカーソルが非表示になっていることを確認します。

『第 6 章 作成したアプリの実行環境PCでの動作』を参照し、作成した画面アプリを動作させてください。

17-12 AI-HMI から共有メモリへ STRING 型で値を書き込めなくなった

AI-HMI 単体で動作させているとき(開発環境 PC など)、共有メモリへは STRING 型変数を制約なしに書き込むことができますが、AI-PLC が動作している環境(実行環境 PC)では、共有メモリへ STRING 型変数を書き込むとき、AI-PLC 側で先に文字列変数を定義しておく必要があります。

(AI-PLC 側で STRING 型変数を定義していない共有メモリの領域へは AI-HMI は値を書き込むことができません。)

AI-HMI 側で STRING 型変数として使用したい共有メモリアドレスにはあらかじめ AI-PLC で STRING 変数を定義しておいてください。

17-13 CSV ファイルで STRING 型の値を使用したい

AI HMI Recipe などを使用する CSV ファイルには STRING 型の値を記述することがあります。
この場合、以下の記述ルールに従ってファイルを作成してください。

1. STRING 型データは前要素の「,」（カンマ）との間にスペース、タブの空白を入れないでください。

リスト 17-13-1 AI HMI Recipe 用 CSV ファイルの例 1

Address,	DataType,	Data
✗ 140000,	STRING,	TestData
○ 140100,	STRING,	TestData

2. STRING 型データに「,」（カンマ）を含む場合、STRING 型データの先頭と最後尾に「”」（ダブルクォーテーション）を記述してください。

リスト 17-13-2 AI HMI Recipe 用 CSV ファイルの例 2

Address,	DataType,	Data
✗ 140000,	STRING,Test,	Data
○ 140100,	STRING,"Test,	Data"

3. STRING 型データに「”」（ダブルクォーテーション）を含む場合、STRING 型データの先頭と最後尾に「”」（ダブルクォーテーション）を記述してください。
また、その上でデータ内の「”」（ダブルクォーテーション）を「” ”」（ダブルクォーテーション）に置き換えてください。

リスト 17-13-3 AI HMI Recipe 用 CSV ファイルの例 3

Address,	DataType,	Data
✗ 140000,	STRING,"Test"Check"Data"	
○ 140100,	STRING,"Test""Check""Data"	

このマニュアルについて

- (1) 本書の内容の一部または全部を当社からの事前の承諾を得ることなく、無断で複写、複製、掲載することは固くお断りします。
- (2) 本書の内容に関しては、製品改良のためお断りなく、仕様などを変更することがありますのでご了承下さい。
- (3) 本書の内容に関しては万全を期しておりますが、万一ご不審な点や誤りなどお気づきのことがございましたらお手数ですが巻末記載の弊社までご連絡下さい。その際、巻末記載の書籍番号も併せてお知らせ下さい。

77AH10001Q
77AH10001A

2023年 6月 第17版
2015年 9月 初版

 株式会社アルゴシステム

本社
〒587-0021 大阪府堺市美原区小平尾656番地

TEL (072) 362-5067
FAX (072) 362-4856

ホームページ <http://www.algosystem.co.jp/>