

本資料は英語版を翻訳した参考資料です。内容に相違がある場合には英語版を優先します。資料によっては英語版のバージョンが更新され、内容が変わっている場合があります。日本語版は、参考用としてご使用のうえ、最新および正式な内容については英語版のドキュメントを参照ください。

要旨 (Introduction)

このアプリケーションノートでは、Express Logic GUIX™ Studio と Renesas Synergy™ GUIX モジュールの活用方法について説明しています。アプリケーションプロジェクトの完了後、ユーザは自身の設計に対して Synergy GUIX モジュールの追加やアプリケーションに合わせた正しい設定、そして付属するサンプルアプリケーションコードを参考にしてコード開発を始めることができます。API の詳細説明や他のアプリケーション例は、参考資料の章に記載された Renesas Synergy Knowledge Base に基本から高度な使用方法まで記されており、より複雑な設計を行うときの貴重な情報になります。

このアプリケーションノートでは、GUIX 機能を応用して、以下のような洗濯機アプリケーション (Washing Machine application) 向けの高機能 GUI を作成する方法を段階的に説明しています。

- ウィジェット (widget) の作成
- メイン画面内での複数の画面の作成
- 画面を切り替え時の子画面の接続と切離し画面乱れをおこさないためのダブルバッファトグル制御 (Double-buffer toggling control)
- 円形スライダ、垂直スライダ、水平スライダの作成
- アニメーションの表示

またこの資料では、Synergy GUIX アプリケーションを作成のための以下の設計ガイドラインや検討すべき事項も掲載しています。

- Synergy MCU 内に搭載されたハードウェアエンジン (JPEG コーデック、2D 描画エンジン) の活用方法
- Synergy GUIX に必要なフラッシュメモリと RAM のサイズ
- RAM/SDRAM が使用する画面サイズ依存のフレームバッファについて
- 内蔵フラッシュメモリと外部フラッシュメモリへの画像の保存方法
- タッチ制御フレームワーク (touch control framework) と I2C フレームワーク (I2C framework) の組み合わせによるタッチインタフェースについて

必須リソース (Required Resources)

Synergy GUIX のサンプルアプリケーションをビルドおよび実行するには、次のリソースが必要です。

開発ツールとソフトウェア (Development tools and software)

- e² studio ISDE v6.2.0 またはそれ以降 (renesas.com/e2studio)、あるいは IAR Embedded Workbench® for Renesas Synergy™ v8.2 またはそれ以降 (renesas.com/ewsyn)
- Synergy ソフトウェアパッケージ (SSP) 1.4.0 またはそれ以降 (renesas.com/ssp)、
- Synergy Standalone Configurator (SSC) 6.2.0 またはそれ以降 (renesas.com/ssc)
- SEGGER J-Link® および関連 USB ドライバ (renesas.com/jlink)
- GUIX Studio v5.4.0 またはそれ以降 (renesas.com/guix-studio)

ハードウェア (Hardware)

- Renesas Synergy™ PE-HMI1 キットバージョン 2.0 またはそれ以降 ([renessasynergy.com/pe-hmi1](https://renesas-synergy.com/pe-hmi1))
- Windows® 7 または 10 が動作している PC
- Mini USB ケーブル

前提条件と対象読者 (Prerequisites and Intended Audience)

このアプリケーションノートは、ユーザが Synergy e² studio ISDE か、Renesas Synergy に対応する IAR Embedded Workbench のいずれか、および Synergy ソフトウェアパッケージ (SSP) の使用経験があることを前提に説明しています。このアプリケーションノートの手順を実行する前に、『SSP ユーザマニュアル』(3.2 章 チュートリアル : Blinky の実行) の手順に従って **Blinky** プロジェクトをビルドし、実行してください。それにより、e² studio と SSP の使用に慣れ、ボードへのデバッグ接続が適切に機能していることを確認できます。

一般的な GUI インタフェース、そのコンポーネント、および設計の原理について理解している必要があります。配色、さまざまな画像形式、画像処理などに関するグラフィックスの基礎、および GUI の構成要素 (ウィンドウ、メニュー、アイコン、ウィジェットなど) に関する基礎知識も必要となります。

Synergy と GUIX を初めて使用する場合、次の資料に記載された GUIX 全般や、Synergy との統合方法について理解する必要があります。これらの資料へのリンクは、「参考資料」を参照してください。

- 『GUIX™ Synergy Port Framework のモジュールガイド』
英文タイトル : GUIX™ Synergy Port Framework Module Guide
- PE-HMI1 用 GUIX™ Hello World アプリケーションプロジェクト
英文タイトル : GUIX™ Hello World Application Project on PE-HMI1

以下のドキュメントを参考資料として使用することもできます。

- 『SSP ユーザマニュアル』
- 『GUIX Studio User's Guide』
- 『GUIX User's Guide』

目次

1. GUIX の導入と GUIX Studio の概要 (Getting started with GUIX and GUIX Studio Overview)	4
1.1 Synergy 向け GUIX アプリケーション開発検討 (GUIX application design consideration for Synergy)	4
2. GUIX モジュールの API 概要 (GUIX Module APIs Overview)	7
3. GUIX ベースの洗濯機アプリケーションの例 (GUIX Based Washing Machine Application Example)	8
3.1 アプリケーションの概要 (Application overview)	8
3.2 GUIX アプリケーションの作成手順 (Sequence of steps to create the GUIX application)	10
3.3 洗濯機アプリケーション向け GUI の概要 (Washing Machine application GUI overview)	10
3.4 GUIX Studio を使用した洗濯機サンプルプロジェクトの作成 (Washing machine example project creation using GUIX Studio)	12
3.4.1 GUIX プロジェクトの新規作成 (Create New GUIX project)	13
3.4.1.1 リソースファイルに対応するディレクトリの選択 (Selecting the directories for the resource files)	14
3.4.1.2 ターゲット CPU とツールチェーンの選択 (Selecting the target CPU and toolchain)	15
3.4.1.3 ディスプレイ設定 (Display configuration)	15
3.4.1.4 詳細設定 (Advanced settings)	16
3.4.1.5 GUIX ライブラリのバージョン (GUIX Library version)	16
3.4.1.6 設定完了後の GUIX プロジェクト (Configured project)	17
3.4.2 GUIX Studio リソースの追加 (Adding GUIX Studio resources)	17
3.4.2.1 ピクセルマップリソースの追加 (Adding Pixelmaps resources)	17
3.4.2.2 色リソースの追加 (Adding Colors resources)	20
3.4.3 Project View を使用したメイン画面の作成 (Creating the main screen under Project View)	21
3.4.3.1 プロパティビューを使用したウィンドウのプロパティの変更 (Changing the window properties using Properties View)	22
3.4.3.2 メイン画面への GUIX ロゴと Synergy ロゴの追加 (Adding GUIX and Synergy logo to the main screen)	24
3.4.3.3 メイン画面への時刻と日付情報の追加 (Adding Time and Date info to the main screen)	27
3.4.3.4 メイン画面へのホームボタンの追加 (Adding Home Button to the main screen)	30
3.4.3.5 メイン画面への複数のボタンの追加 (Adding Buttons to the main screen)	32
3.4.3.6 メイン画面への [Washer] (洗濯機の機能) ウィンドウの追加 (Adding Washer window to the main screen)	34
3.4.3.7 メイン画面へのスライダーバーの追加 (Adding Slider Bar to the main screen)	36
3.4.3.8 メイン画面への画面名の追加 (Adding the screen name to the main screen)	37
3.4.4 Garments (衣類) ウィンドウの作成 (Creating Garments Window)	39
3.4.5 [Temperature] (温度) ウィンドウの作成 (Creating Temperature Window)	40
3.5 [Water Level] (水位) ウィンドウの作成 (Creating Water Level Window)	42

3.5.1	作成したプロジェクトから生成するリソースと仕様 (Resource and Specification files from the created project)	43
3.6	アプリケーションでの GUIX モジュールの包含と設定 (Including and configuring the GUIX module in an application)	45
3.6.1	画面固有の関数 (Screen specific functions)	56
4.	Washing Machine (洗濯機) アプリケーションの実行 (Running the Washing Machine Application)	56
4.1	プロジェクトのインポート、ビルド、およびロード (Importing, building and loading the project)	56
4.2	アプリケーションのロードとデバッグ (Loading and Debugging the Application)	57
4.3	アプリケーションの検証 (Verifying the Application)	57
5.	次の手順 (Next Steps)	58
6.	参考資料 (References)	58
	ホームページとサポート窓口.....	59

1. GUIX の導入と GUIX Studio の概要 (Getting started with GUIX and GUIX Studio Overview)

GUIX Studio は、組み込みアプリケーション向け GUI の作成を目的とした、GUI アプリケーション設計ツールと環境です。GUIX は、ターゲット MCU で GUI アプリケーションを実行させるための、Express Logic 社が提供するミドルウェアソフトウェアライブラリです。Express Logic の GUIX Studio と GUIX Library の詳細は、本資料の「参考資料」に記載されている『GUIX Studio User's Guide』と『GUIX User's Guide』に記載されています。

1.1 Synergy 向け GUIX アプリケーション開発検討 (GUIX application design consideration for Synergy)

Renesas S7 と S5 の Synergy MCU グループシリーズは、以下の機能を持つグラフィックス LCD 機能を搭載しています。

- グラフィックス LCD コントローラ (GLCDC)
- 2D 描画エンジン (DRW)
- JPEG コード

1. メモリ要件に関する検討事項 (Memory requirement considerations)

- A. RAM/ROM 要求 (RAM/ROM requirement) : GUIX は、GUIX スレッド (GUIX thread) と他のグローバルな構造体データ (Global data structures) のために、約 80 K バイトの ROM (読み出し専用メモリ) と 10 K バイトの RAM を必要とします。これには、描画用のキャンバスメモリ (Canvas memory) やフレームバッファ (frame buffer) は含まれません。
- B. 内部フラッシュ / 外部フラッシュ (Internal Flash / External Flash) : 大部分のアプリケーションでは、GUIX のコアライブラリのストレージ要求 (core GUIX library storage requirements) に含まれないグラフィカルリソースも利用します。このリソースには、フォント、グラフィカルアイコン (ピクセルマップ pixelmaps)、固定の文字列があります。これらデータは、フラッシュ メモリの ROM 領域に配置します。コンパイルしたアプリケーションは、GUIX リソースイメージとともに内部フラッシュに配置します。コードサイズが大きい場合は、外部フラッシュに配置することもできます。メモリ領域のサイズは、使用するフォントの数やサイズ、グラフィカルアイコンの数やサイズ、出力カラーデータに依存します。また、RLE 圧縮 (Run Length Encoding compression) もサポートしているためフォントやピクセルマップデータの圧縮にも依存します。各リソースの

ストレージ要求は GUIX Studio アプリケーションの中で表示されるため、アプリケーションリソースが消費するフラッシュメモリの量の追跡と監視に役立ちます。

- C. **内部 RAM / 外部 RAM に関する検討事項 (Internal / External RAM considerations)** : キャンバスまたはフレームバッファが使用するメモリサイズは、LCD ディスプレイのサイズ、色の深度、画像の形式、画面の数などによって異なります。320 ピクセル * 256 ピクセルのような小型 LCD ディスプレイの場合、画像の形式や画面の色数にもよりますが、S7/S5 Synergy MCU グループシリーズの SRAM に格納できることがあります。800 ピクセル * 480 ピクセルのようなより大きい LCD ディスプレイの場合、「16 ビット RGB 565」(赤 5 ビット、緑 6 ビット、青 5 ビット) の画像形式を使用すると、S7/S5 Synergy MCU ボードの SRAM に格納することはできません。この場合、フレームバッファはボード上で利用可能な外部 RAM (SDRAM) 内に配置する必要があります。フレームバッファのメモリ要件は、フレームサイズと色深度の関数で表され、次の式で定義することができます。

フレームバッファの RAM (バイト) = $(x * y * (\text{bpp}/8))$

ここで、「x」と「y」は、キャンバス (ディスプレイ) の横と縦のピクセル数、bpp は 1 ピクセルのビット数です。

例えば、サイズが 800 * 480 で 16 ビット RGB 565 を使用しているディスプレイに対応するフレームバッファのサイズは、 $(800 * 480) * 16/8 = 768,000$ バイトと計算します。

- D. **JPEG ワークバッファに関する検討事項 (JPEG Work buffer considerations)** : JPEG のワークバッファに関して、JPEG のデコード速度とバッファサイズはトレードオフの関係にあります。画面のウィジェットが JPEG 形式になっている場合、JPEG のワークバッファは、デコード後の画像を作成するための一時的なストレージメモリとして使用されます。画像全体をデコードするうえでバッファサイズが十分な大きさではない場合、JPEG のデコードは出力バッファを使用したストリーミングモード (output buffer streaming mode) で実行されます。2D 描画エンジンが使用する BitBLT 演算は、バッファ内で JPEG ラスタ画像 (raster image) の一部をデコードし、次にフレームバッファに転送します。JPEG 作業バッファの最小サイズは {(水平ライン内のピクセル数) x (表示形式に対応する bpp (バイト数/ピクセル)) x 8 (ライン数)} です。例えば、デコード後の画像が水平ライン内で 800 ピクセルの幅で、RGB565 形式を使用している場合、この値は $800 \times 2 \times 8 = 12,800$ バイトになります。バッファサイズがこの値より小さい場合、JPEG のデコードは処理されません。スループット (throughput) を改善するには、**Size of the JPEG Work Buffer** (JPEG ワークバッファのサイズ) パラメータをより大きい値に設定する必要があります。その結果、JPEG デコードのスループットが向上します。JPEG 出力バッファのストリーミングモードを繰り返して、部分的な JPEG のデコードを繰り返すため、オーバーヘッドが発生します。
- E. **ソフトウェアレンダリングとハードウェアレンダリングの比較 (Software Rendering vs Hardware Rendering)** : ソフトウェアレンダリングと比較すると、ハードウェアレンダリングの方が高性能です。ただしこの性能を実現するためには、ハードウェアエンジンのために追加バッファが必要になり、メモリに制約のあるシステムではこの点を考慮する必要があります。
- F. **画面の回転 (Screen Rotation)** : GUIX モジュールは、90、180、270 度の画面の回転をサポートしています。画面の回転機能を使用するには、GUIX はフレームバッファ以外のメモリ (キャンバスバッファ Canvas buffer) を必要とします。

2. SSP フレームワークのインタフェース (SSP Framework interface)

SSP は、GUIX 用の ISDE 設定の一部として、以下のフレームワークと HAL ドライバを (HAL drivers) 提供しています。GUIX アプリケーションを設計する際に、以下の SSP コンポーネントを活用することができます。SSP タッチパネルフレームワークを使用する場合、メッセージングフレームワーク、I2C ドライバ、タッチドライバなどの関係するコンポーネントも含まれます。LCD のバックライトを駆動する PWM ドライバを使用することもできます。

- タッチパネルフレームワークインタフェース (Touch Panel Framework Interface)
- メッセージングフレームワーク (Messaging Framework)
- I2C ドライバ (I2C driver)
- タッチドライバ (Touch driver)
- PWM ドライバ (PWM driver)

3. ピン設定 (Pin configurations)

SSP は、GLCD 周辺回路ピン (GLCD Peripheral pins)、GPIO ピン (GPIO pins)、タイマ関連ピン (Timer related pins) など、各種ピンを設定するためのピンコンフィギュレータ (Pin configurator) を提供します。GUIX アプリケーションを設計する場合、ピンコンフィギュレータを使用してピンを設定してください。タッチインタフェースを搭載した LCD パネル (LCD panel with touch interface) を使用する場合、以下のピン設定が必要です。

- GLCDC 周辺回路ピン設定 (GLCDC Peripheral pin configuration)
- PWM ピン設定 (PWM Pin configuration)
- 割り込みピン設定 (Interrupt Pin configuration)
- I2C ピン設定 (I2C Pin configuration)
- リセットピンなどの LCD 制御ピン (LCD Control pins such as, Reset Pins)

4. 性能に関する検討事項 (Performance considerations)

- **外部フラッシュ / 内部フラッシュ (External Flash / Internal Flash)** : アプリケーションの性能は、アプリケーションや GUI のイメージを保存する場所に左右されます。アプリケーションと GUI のイメージは、内部フラッシュに保存する方が性能は良好です。
- **内部 RAM / 外部 RAM に関する検討事項 (Internal / External RAM considerations)** : 外部 RAM (SDRAM) と比較し、内部 RAM (SRAM) を使用の方が性能は良好です。メイン画面の中で画面を更新するアプリケーションや、それに対応するメモリレイアウトを SRAM に配置するように選択することができます。
- **色深度 (Color Depth)** : 色深度の高いアプリケーションより、色深度の低いアプリケーションの方が性能は良好です。24 bpp や 32 bpp の画像形式と比較し、8 bpp や 16 bpp の方が性能は良好です。
- **ディスプレイの解像度 (Display Resolution:)** : より高解像度の LCD ディスプレイの場合、ソフトウェアレンダリングの場合と比較し、ハードウェアエンジンを使用の方が高性能となります。
- **ソフトウェアレンダリングとハードウェアレンダリング (Software/Hardware rendering)** : GUIX は JPEG と 2D の描画エンジン (JPEG and 2D Drawing Engine) をサポートしており、グラフィックスのレンダリングとアクセラレーション機能を使用することで GUI の性能が向上します。これらのハードウェアモジュールは、CPU の介在なしにフレームバッファへの書き込み処理することができます。GUIX は、ハードウェアエンジンが使用できない場合に、ソフトウェアレンダリングをサポートしています。この場合、フレームバッファはソフトウェアによって全面的に制御されます。ソフトウェアレンダリングは、フレームバッファが小さい小型ディスプレイや、MCU のバンド幅利用率が高くないアプリケーションに適しています。ただし大型の LCD ディスプレイでソフトウェアレンダリングを使用すると、性能が問題になる可能性があります。

2. GUIX モジュールの API 概要 (GUIX Module APIs Overview)

以下に、GUIX モジュールで使用可能な API を示します。これらの API は、Synergy GUIX フレームワークモジュールの汎用 API (Synergy GUIX Framework Module Generic APIs) と、GUIX 固有 API (GUIX Specific APIs) に大別することができます。

GUIX フレームワークモジュールの API (GUIX Framework Module APIs)

以下の API は Synergy GUIX モジュールの一部であり、ロウレベルグラフィックスドライバの設定とキャンバスメモリの初期化に使用されます。これら API の詳細は、『SSP ユーザマニュアル』(5.1.27 章 GUIX Synergy Port Framework Module) を参照してください。

表 1 GUIX フレームワークモジュールの API (GUIX Framework Module APIs)

open	標準的な SSP に準拠する API であり、SF_EL_GX モジュールを開き、ロウレベルグラフィックスデバイスドライバとフレームバッファを設定する目的で使用します。
close	SF_EL_GX モジュールと、ロウレベルドライバを閉じます。
versionGet	モジュールのバージョンを返します。
setup	ロウレベルグラフィックスデバイスドライバを初期化するためのインタフェースであり、GUIX (Studio) サービスコール gx_studio_display_configure() を通じて、関数ポインタ (function pointer) の形で GUIX に渡す必要があります。次に GUIX は API をコールバックし、このときこの API は open によって渡された設定に基づいて SSP デバイスドライバを設定します。
canvasInit	GUIX のヘルパー API (helper API) であり、GUIX キャンバスのメモリアドレスを決定します。この API は、(GX_WINDOW_ROOT *) 型の引数を受け取り、GUIX に対してキャンバスメモリの開始アドレスを渡します。このアドレスは、低水準グラフィックスドライバが画像の描画や表示を行うために必要とするものです。

GUIX API : アプリケーションに対して、GUIX は豊富な API セットを提供します。ユーザが GUI アプリケーションの開発で利用できる API が 500 種類以上あります。GUIX API のグループを以下に示します。API の詳細な説明については、「GUIX User's Guide」 (日本語版 : Guix ユーザーズマニュアル) の 5 章を参照してください。

表 2 GUIX API の分類

アニメーション (Animation)	アニメーションコンポーネント、その関数、サービス。任意のウィジェットタイプ (widget type) に対する、フェードイン (fading in)、フェードアウト (fading out)、移動、またはスライド方式のアニメーション。
ユーティリティ関数 (Utility function)	GUIX で一般的に使用できるユーティリティ関数。例：整数を ASCII に変換する関数、平方根、サイン関数を計算する算術関数など。
ウィンドウ (Window)	ウィンドウ固有のコンポーネントとその関数。例：ウィンドウの作成、ウィンドウの描画、ウィンドウイベントの処理、ウィンドウスクロール情報の取得など。
描画 (Drawing)	GUIX が画面にあらゆる視覚的要素を描画するために必要とする基本的図形の描画。
ウィジェット関連 (Widget related)	ウィジェットの作成、ウィジェット境界の描画などを行う、ウィジェット関連コンポーネントとその関数。
ディスプレイ (Display)	ディスプレイの作成、ディスプレイテーブル内での色の置換などを行う、ディスプレイ関連コンポーネントとその関数。
キャンバス (Canvas)	キャンバスコンポーネントに関連する処理を行うこれらの API は、キャンバスの作成、キャンバスの非表示、キャンバスの表示などの処理に対応します。
GUIX システム (GUIX System)	システムの初期化、システム言語の設定、システムタイマの開始など、システム固有の機能。

GUIX API は、ヘッダファイル `gx_api.h` の中でも参照できます。

3. GUIX ベースの洗濯機アプリケーションの例 (GUIX Based Washing Machine Application Example)

3.1 アプリケーションの概要 (Application overview)

このアプリケーションについて理解するには、アプリケーションのアーキテクチャと Synergy との統合方法の理解から始める必要があります。また、このアプリケーションと SSP フレームワークやスレッドとの相互作用についても理解する必要があります。

図 1 に、SSP フレームワークと HAL ドライバに統合されている洗濯機アプリケーションのアーキテクチャの概要を示します。ここに示すアプリケーションスレッド (HMI スレッド HMI Thread) は、以下の機能を持ちます。

- GUIX とそのドライバの初期化
- ディスプレイの設定
- キャンバスの初期化
- バックライト用 PWM ドライバの初期化

タッチパネルフレームワークは、LCD 上のタッチコントローラを制御します。LCD 上のタッチパネルで発生したタッチイベントを、I2C インタフェース経由で受信します。タッチパネルフレームワークはタッチイベントを受信し、メッセージングフレームワークを使いそれに付随するスレッドにタッチイベントを送信します。タッチイベントに紐づいた HMI スレッド(HMI Thread)は、タッチパネルフレームワークからタッチイベントが返されるのを待ち、メッセージングフレームワークからイベントを取得します。

HMI スレッドは、(GUIX フレームワークの一部として生成される) GUIX スレッドによる処理を進めるためにタッチイベントを GUIX イベントに変換します。

SF_EL_GX は GUIX が Synergy MCU ボードにアクセス行うレイヤであり、グラフィクスエンジンである GLCDC、DRW (2DG エンジン)、JPEG デコードエンジンにもアクセスすることができます。

SF_EL_GX には、以下のような主要な関数があります。

- GUIX を SSP フレームワークに適応させます。
- SSP ディスプレイドライバインタフェースを GUIX ディスプレイドライバインタフェースに接続します。
- Synergy 2D 描画エンジン (DRW) によるアクセラレーションを使用し、GUIX がウィジェットを描画できるようにします。
- 画面のちらつきを発生させないためのダブルバッファをトグル制御します。

SSP、SF_EL_GX およびそれに関する HAL に対する GUIX の実装方法は、『SSP ユーザマニュアル』(SSP User's Manual) に掲載されています。

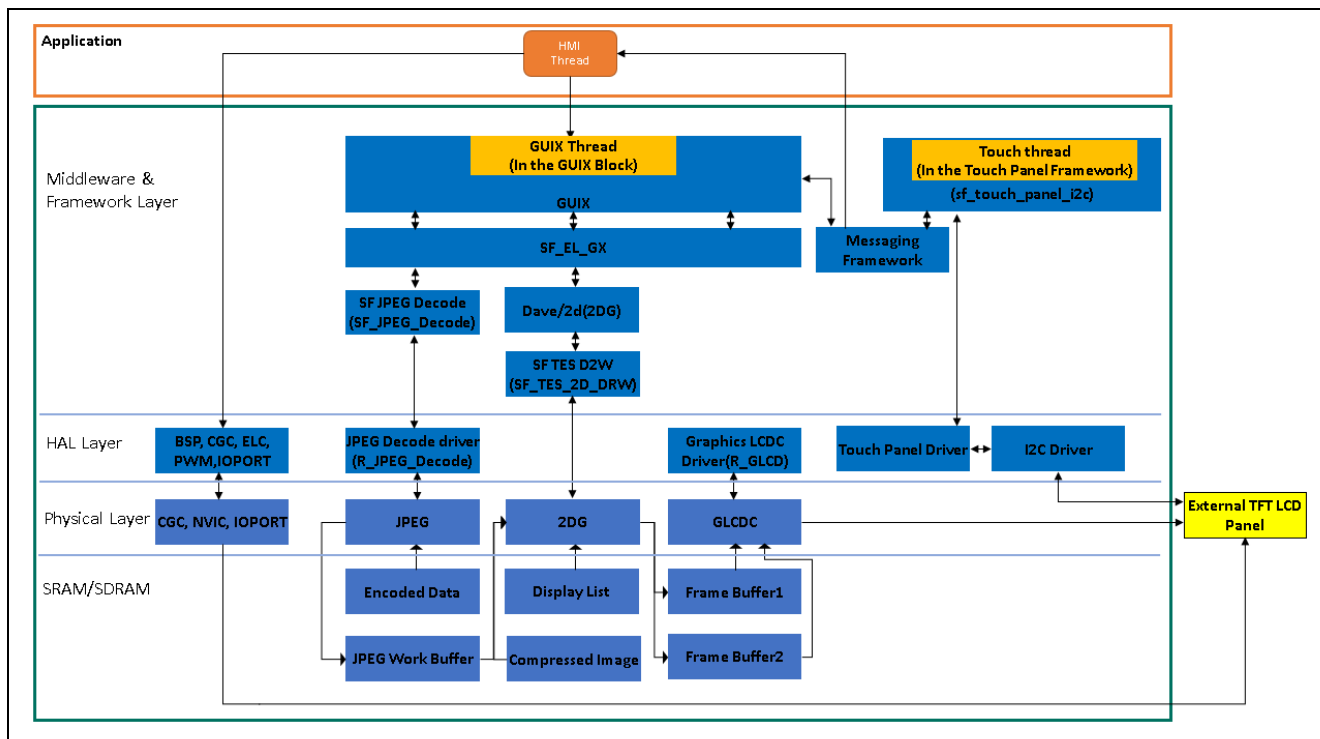


図 1 アーキテクチャの概要

3.2 GUIX アプリケーションの作成手順 (Sequence of steps to create the GUIX application)

以下に、GUIX アプリケーションを作成する際に必要となる開発のステップを示します。ステップ 1 と 4 は本資料で説明していません。それ以外のステップは、この後の 3.4 章と 4 章で説明します。

1. GUI 画面に対応するピクセルマップの画像を作成します。
一般的にこの作業は、グラフィックデザイナーが担当します。これらの画像を GUIX Studio に入力し、必要な複数の画面を作成します。
2. GUIX Studio で GUI を作成します。
GUIX Studio を起動し、プロジェクト用件に従ってさまざまな画面を作成します。
3. リソースファイルを生成します。
GUIX Studio を使用して画面を作成した後、ウィジェットとそのハンドラ関数は、アプリケーションコードとともにコンパイルできる形式 (C 言語プログラムコードや C データ構造) にする必要があります。
4. 画面イベントと画面描画のハンドラ関数に対応するコードを作成します。
各画面には、それぞれイベントハンドラと描画ハンドラが存在します。プロジェクト要求に基づいてイベントを処理し、画面を描画するコードを作成する必要があります。また他に、初期関数やユーティリティ関数の作成も必要です。
5. ISDE コンフィギュレータを使用して GUIX を SSP に実装します。
アプリケーションスレッドを作成します。GUIX は SSP に統合されます。このプロジェクトで使用する LCD のタイプ、タッチコントローラ、グラフィックスエンジンの仕様に基づいて、ISDE コンフィギュレータを使って GUIX コンポーネントの追加と設定を行う必要があります。
6. Synergy e² studio ISDE を使用して GUIX アプリケーションコードを統合します。
生成するリソースファイルに加えて、GUIX システムの初期化、GUIX ドライバの設定、キャンバスの初期化、ウィジェット作成 API (widget creation API) を使用した画面の作成、GUIX の起動、タッチドライバから受け取るタッチイベントの制御が必要になります。これらの作業はいずれも、アプリケーションスレッド (サンプルアプリケーションでは HMI スレッド) から実行します。
7. コードをビルドします。コードを生成し、ターゲット MCU 向けにビルドします。

3.3 洗濯機アプリケーション向け GUI の概要 (Washing Machine application GUI overview)

ここでは、洗濯機コントローラをアプリケーションの実例として説明しています。図 2 は、洗濯機コントローラを構成するグラフィックデザインを示しています。以下のように、4 種類の画面に分類することができます。

1. メイン画面 (洗濯機の設定画面 Washer Settings screen)
2. 衣類選択画面 (Garments selection screen)
3. 水位選択画面 (Water level selection screen)
4. 温度選択画面 (Temperature selection screen)

このアプリケーションは、GUI の視点から、洗濯機コントローラをシミュレーションします。

メイン画面 (Main screen) : メイン画面は洗濯機の制御機能を表示するもので、円形のスライダと、さまざまな画面の選択用ボタンや、画面の電源オン/オフ用ボタンがあります。

制御機能 (control configurations) に加えて、以下の情報が表示されます。

- Soak、Wash、Rinse、Spin (注水、洗濯、すすぎ、脱水) という状態を示すステータスバー。
- 洗濯サイクルの残り時間を示すステータスバー。
- 時刻と日付の情報。

Garment (衣類) 選択画面 : 衣類の種類を選択するための画面で、円形のスライダと、さまざまな画面の選択用ボタンや、画面の電源オン/オフ用ボタンがあります。

制御機能に加えて、以下の情報が表示されます。

- Soak、Wash、Rinse、Spin (注水、洗濯、すすぎ、脱水) という洗濯の状態を示すステータスバー。
- 時刻と日付の情報。

Water level (水位) 選択画面：水位を選択するための画面で、垂直スライダと、さまざまな画面の選択用ボタンや、画面の電源オン/オフ用ボタンがあります。

加えて、以下の情報が表示されます。

- 洗濯槽に対する水位をパーセントで表示。
- Soak、Wash、Rinse、Spin (注水、洗濯、すすぎ、脱水) という洗濯の状態を示すステータスバー。
- 時刻と日付の情報。

Temperature (温度) 選択画面：水の温度を制御するための円形のスライダがあります。また、さまざまな画面の選択用ボタンや、画面の電源オン/オフ用ボタンがあります。

制御機能に加えて、以下の情報が表示されます。

- 現在の水温を示すステータスバー。
- Soak、Wash、Rinse、Spin (注水、洗濯、すすぎ、脱水) という洗濯の状態を示すステータスバー。
- 時刻と日付の情報。

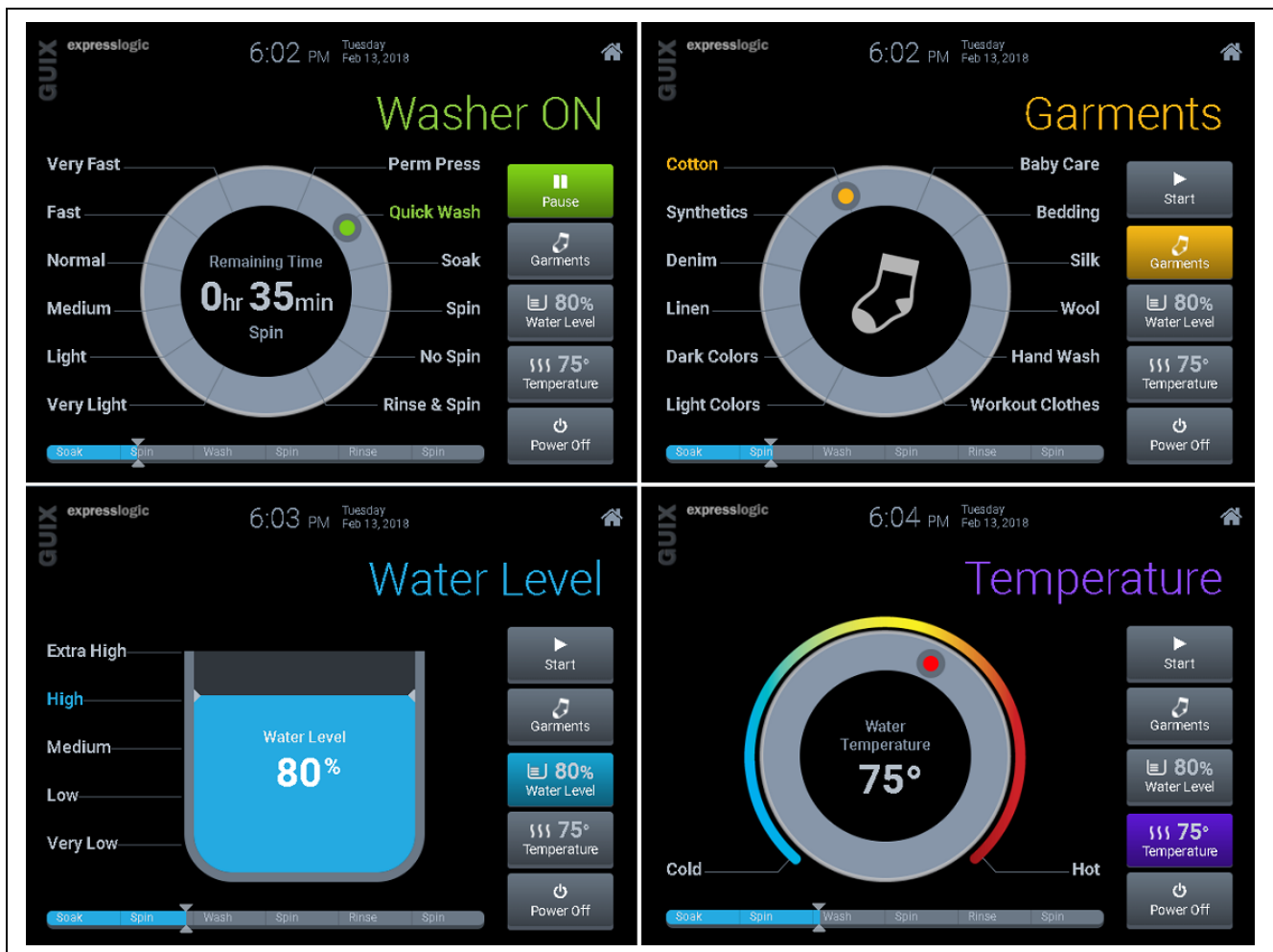


図 2 洗濯機コントローラで使用する GUI の概要

以下の図は、メイン画面の詳細を示しています。

1. Express Logic による、「GUIX」と「Express Logic」の各ロゴ
2. 時刻、日付、カレンダーの情報
3. ホームボタン
4. 洗濯機のステータス
5. さまざまな画面を選択するための制御ボタン
6. 洗濯サイクルを表すステータスバー
7. 洗濯機の制御ウィンドウ
8. 洗濯の状態に関する情報を示す円形のスライダ



図 3 洗濯機コントローラで使用する GUI の概要

以下の章でも、図 3 に記された番号(1~8)の順序で GUIX Studio を使用したウィジェットの作成方法を解説しています。

この章では、ユーザが GUIX や GUIX Studio の各ビューの基礎を理解し、本資料の巻末にある「参考資料」に掲載した『GUIX Module Guides』ドキュメントを閲覧した実績があることを前提に説明しています。まだこの参考資料を読んでいない場合、まずこれらのドキュメントを参照することを推奨します。

GUIX Studio を使用して GUI を作成する前に、グラフィックス画像と、.png (portable network graphics) 形式の画像ファイルを GUIX Studio で使用できるようにする必要があります。これらの画像ファイルは、グラフィックスリソースへの入力として使用します。グラフィックス画像の作成と画像ファイルへの変換は、このアプリケーションノートでは取り扱いません。グラフィックスデザイナーからこれらのファイルを受け取することを想定します。

3.4 GUIX Studio を使用した洗濯機サンプルプロジェクトの作成 (Washing machine example project creation using GUIX Studio)

この章では、GUIX Studio を使用して洗濯機 GUI の作成方法を段階的に説明します。サンプルプロジェクトを作成する前に、GUIX Studio の起動方法と、以下のようなさまざまなビュー(View)について理解しておくことを推奨します。ビューには、Project View (プロジェクトビュー)、Properties View (プロパティビュー)、

Target View (ターゲットビュー)、Resources View (リソースビュー) (Color (カラー)、Fonts (フォント)、Pixelmaps (ピクセルマップ)、文字列など)があります。これらは、本資料の「参考資料」に掲載されている『GUIX Studio User's Guide』の Chapter 3 と 4 で説明されています。これらドキュメントを参照しておくことで以下の説明に対する理解が容易になります。

3.4.1 GUIX プロジェクトの新規作成 (Create New GUIX project)

GUIX Studio で、[Project] タブ -> [New Project] を選択します。次の図に示す[Create New Project] ウィンドウがポップアップ表示されます。

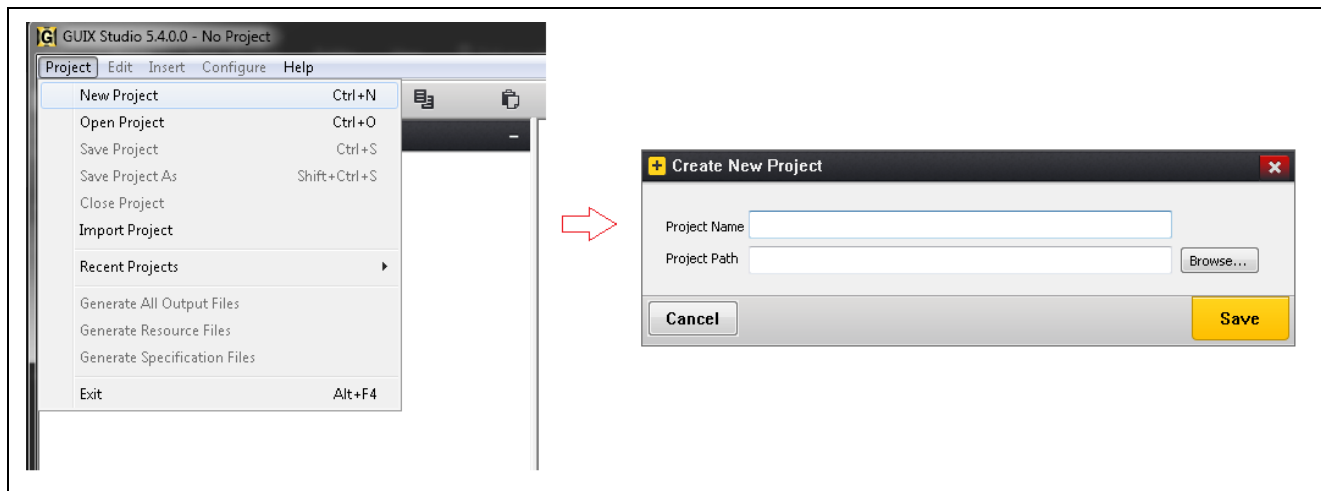


図 4 新しいプロジェクトの作成

[Project Name] と [Project Path] を入力します。Project Path は、プロジェクトファイル (project file) を配置するディレクトリ (directory) を意味します。入力後、[Save] ボタンをクリックします。

[Save] ボタンをクリックすると、[Configure Project] ウィンドウが表示されます。このウィンドウには、プロジェクトの初期設定のための多くの設定項目があります。以下の章で、これら設定項目の詳細と、変更方法を説明します。

注記：次の図で示すように、[Project Path] は GUIX Studio のインストール先と同じ場所を指定してください。

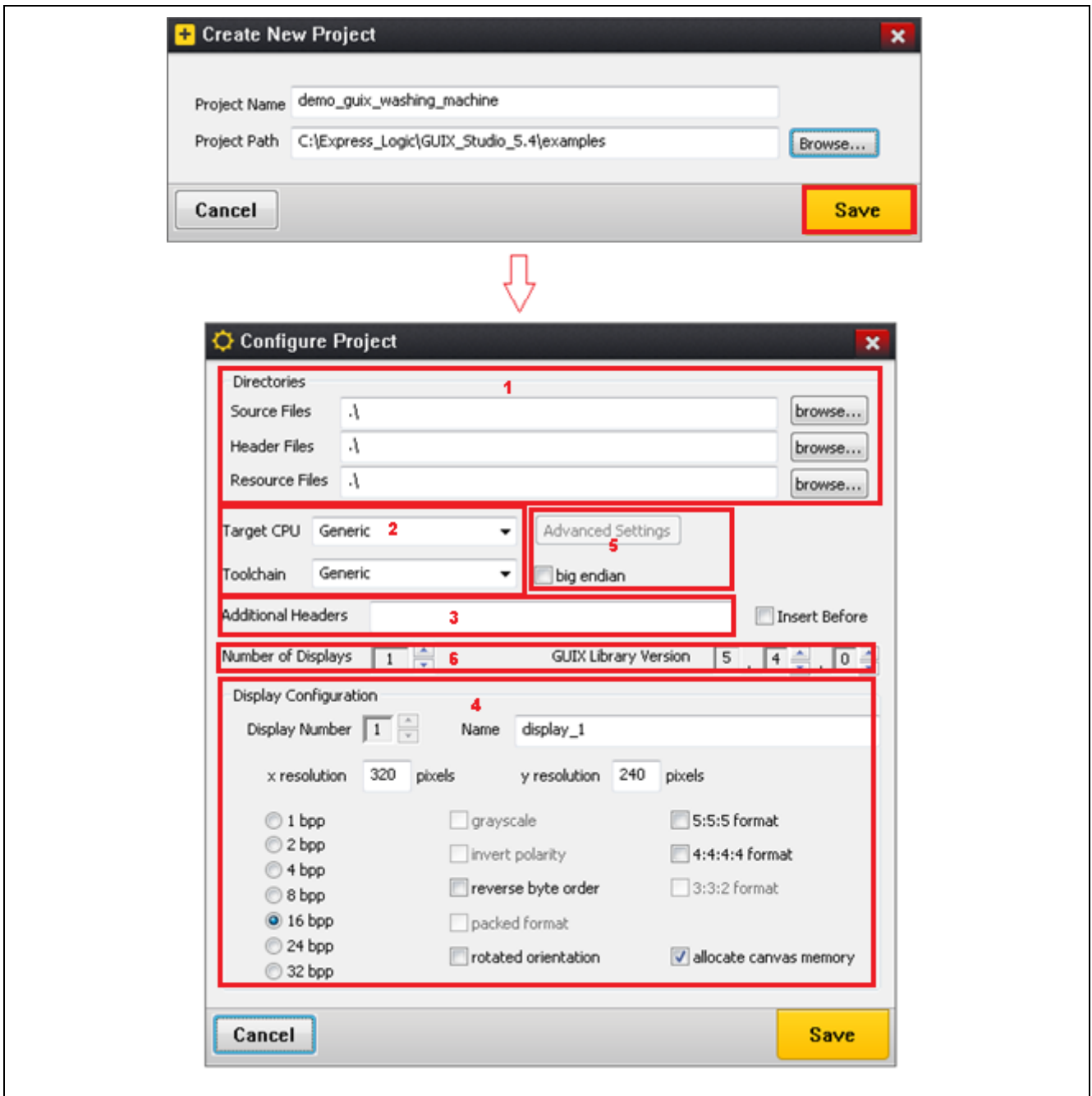


図 5 新しい [Configure Project] ウィンドウ

3.4.1.1 リソースファイルに対応するディレクトリの選択 (Selecting the directories for the resource files)

図 5 に示す [Configure Project] スナップショットの [Directories] 部 “1” では GUIX のソースファイルとヘッダファイルが配置される選択可能なディレクトリを示しています。[Resource Files] ディレクトリも、GUIX Studio がリソースファイル(仕様ファイルが配置される)を生成したディレクトリであり、選択可能です。

プロジェクトにヘッダファイルを追加する場合、ハイライトされている [Additional Headers] 部 “3” で設定することができます。

注記：ワークスペース内では、アプリケーションのソースコードを配置するディレクトリを指定することを推奨します。これにより、GUIX Studio で変更が行われたときに生成されたリソースファイルが、ビルド用のプロジェクトから使用できるようになります。リソースファイルを手動でコピーする必要がなくなります。

3.4.1.2 ターゲット CPU とツールチェーンの選択 (Selecting the target CPU and toolchain)

GUIX Studio はさまざまな種類のターゲット CPU とツールチェーン “2” をサポートしています。デフォルトで、Generic CPU (汎用 CPU) と Generic Tool chain (汎用ツールチェーン) をサポートします。加えて、Renesas Synergy CPU と GNU/IAR ツールチェーンの組み合わせもサポートします。Renesas Synergy でアプリケーションを開発するには、[CPU] として [Renesas Synergy] を選択し、[Toolchain] として [GNU] か [IAR] を選択してください。このサンプルアプリケーションでは、[GNU] ツールチェーンを選択しています。

3.4.1.3 ディスプレイ設定 (Display configuration)

GUIX プロジェクトのディスプレイ設定 “4” で、プロジェクトが必要とするディスプレイの数を指定します。[Display Number] を使用してこの値を設定できます。洗濯機アプリケーションの場合、ディスプレイの数は 1 です。ディスプレイの名前を指定できます。洗濯機アプリケーションの場合、この 1 つのディスプレイが、メインディスプレイ (**main display**) となります。

[GUIX Configuration] では、X と Y のピクセル解像度を設定します。ディスプレイの解像度には、800 * 480 (PE-HMI1 ボードに搭載されている LCD の解像度) を選択しています。

GUIX の設定で、ピクセルあたりのビット数 (bpp) を指定します。このサンプルでは、16 bpp を選択しました。

以下の項目の設定が可能です。

- [grayscale] (グレースケール)
- [invert polarity] (極性反転)
- [reverse byte order] (バイト逆順)
- [packed format] (パケットの形式)
- [rotated orientation] (回転方向)

また、プロジェクトの要件に応じて、さまざまな RGB 形式を選択できます。

GUIX の設定を使用して、[Allocate Canvas memory] (キャンバスメモリの割り当て) を指定することができます。この設定項目は、Windows 環境に効果的ですが、組み込みシステムの場合、このオプションは選択されていません。

3.4.1.4 詳細設定 (Advanced settings)

[Synergy Advanced Settings] を使用して、Synergy MCU グループの一部で使用できる機能を設定することができます。[Runtime Image Decoder] (実行時の画像デコーダ) に対応する設定と、[2D Drawing Engine] (2D 描画エンジン) を有効にするためのオプションを選択することができます。洗濯機アプリケーションでは、次の図に示すように、[2D Drawing Engine] (2D 描画エンジン) を有効にし、[Hardware JPEG Decoder] (ハードウェア JPEG デコーダ) を選択しています。

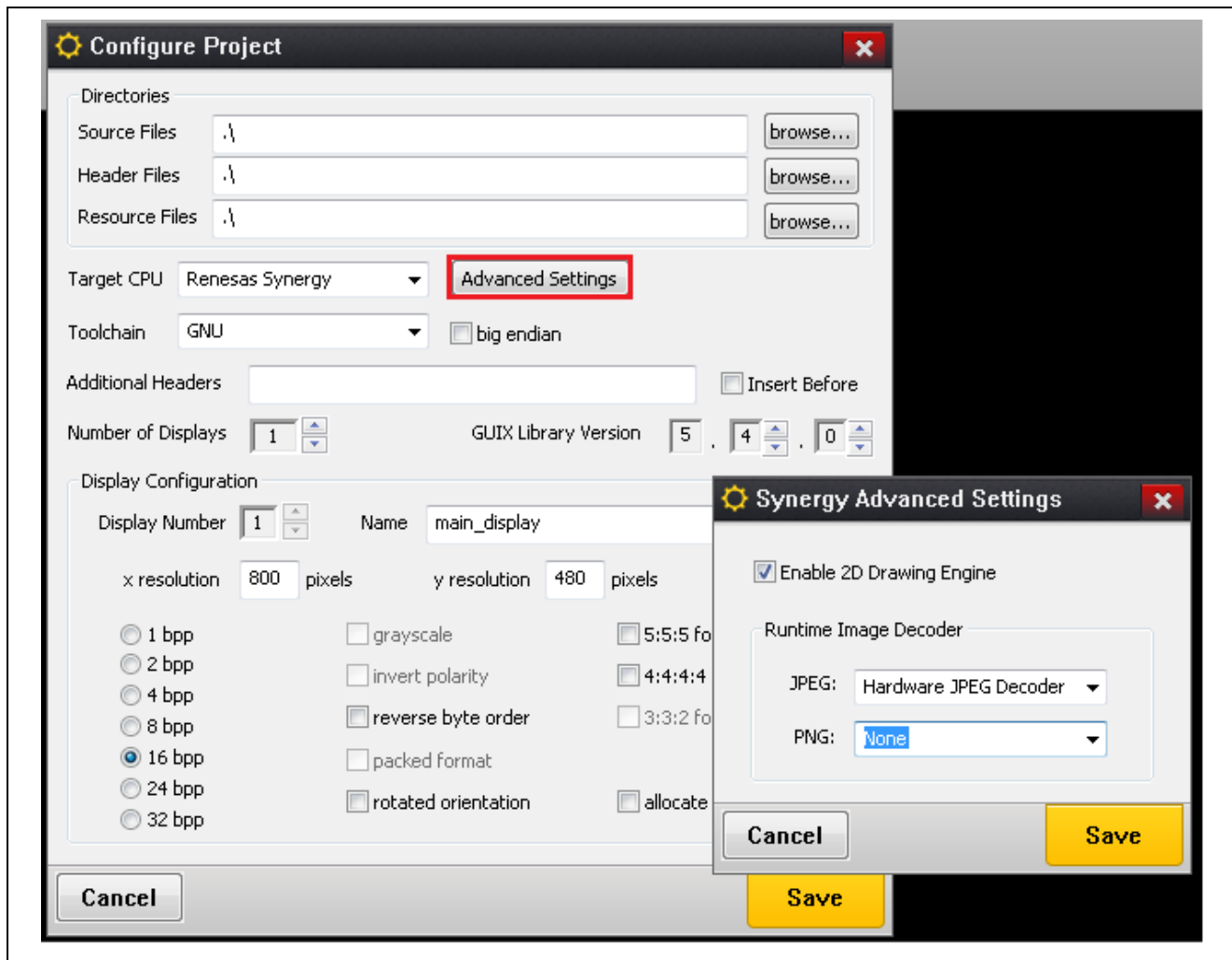


図 6 Synergy Advance Settings (Synergy の詳細設定)

3.4.1.5 GUIX ライブラリのバージョン (GUIX Library version)

この章で、GUIX ライブラリのバージョンを設定します。GUIX Studio と組み合わせて使用する SSP のバージョンに合わせることを推奨します。GUIX Studio バージョン 5.4.0.0 で使用しているライブラリのバージョンは 5.4.0 です。

3.4.1.6 設定完了後の GUIX プロジェクト (Configured project)

次の図に、洗濯機アプリケーションに必要な設定を行ったプロジェクトを示します。

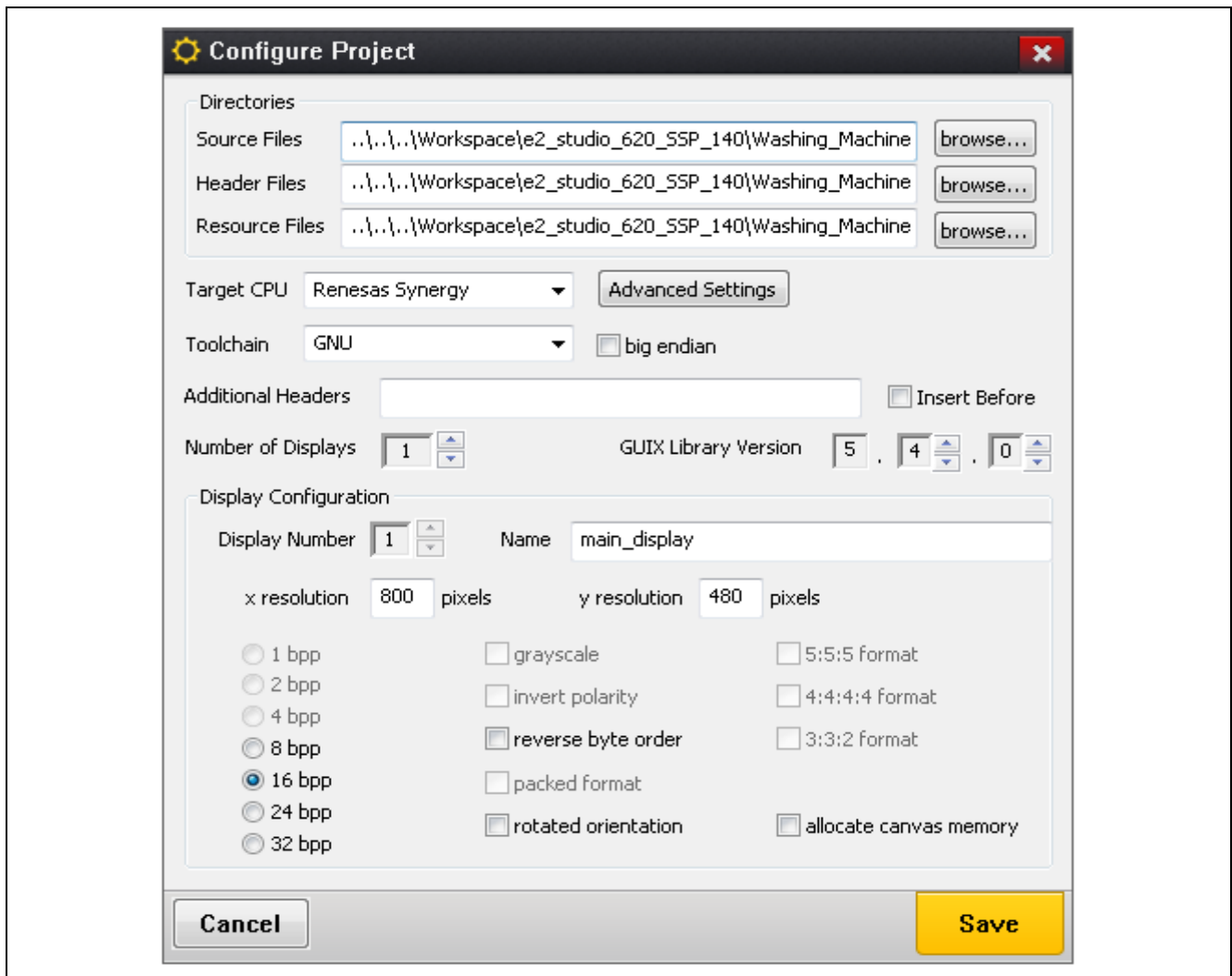


図 7 洗濯機プロジェクトの設定

3.4.2 GUIX Studio リソースの追加 (Adding GUIX Studio resources)

GUIX Studio は、色 (colors)、フォント (fonts)、ピクセルマップ (pixel maps)、文字列 (strings) のようなリソースに対するインタフェースを提供します。この章では、ピクセルマップと色の各リソースを追加する方法を説明しています。GUIX リソースの詳細については、本資料の「参考資料」に掲載されている『GUIX Studio User's Guide』の Chapter 4 「GUIX Studio Resources」を参照してください。

3.4.2.1 ピクセルマップリソースの追加 (Adding Pixelmaps resources)

ピクセルマップの追加とは、GUIX Studio を使用してプロジェクトにグラフィックスリソースを追加することを意味します。グラフィックスデザイナーから各画像に対応する個別の .png ファイルが提供されます。GUIX Studio で、使用するピクセルマップリソースに対してこれらの画像を追加する必要があります。ピクセルマップに関して、[System] ディレクトリと [Custom] ディレクトリを指定します。[System] ディレクトリは、標準的な画像の保存場所です。[Custom] ディレクトリは、カスタムデザインした画像の配置先であり、画像をこのディレクトリにコピーする必要があります。

GUIX Studio IDE の右側に、リソースセクションがあります。[Pixelmaps] をクリックして、次の図に示すように [Custom] フォルダを選択し、グレー表示されている [+ Add New Pixelmap] をクリックする方法により、[Pixelmaps] -> [Custom] の [+ Add New Pixelmap] を選択することができます。

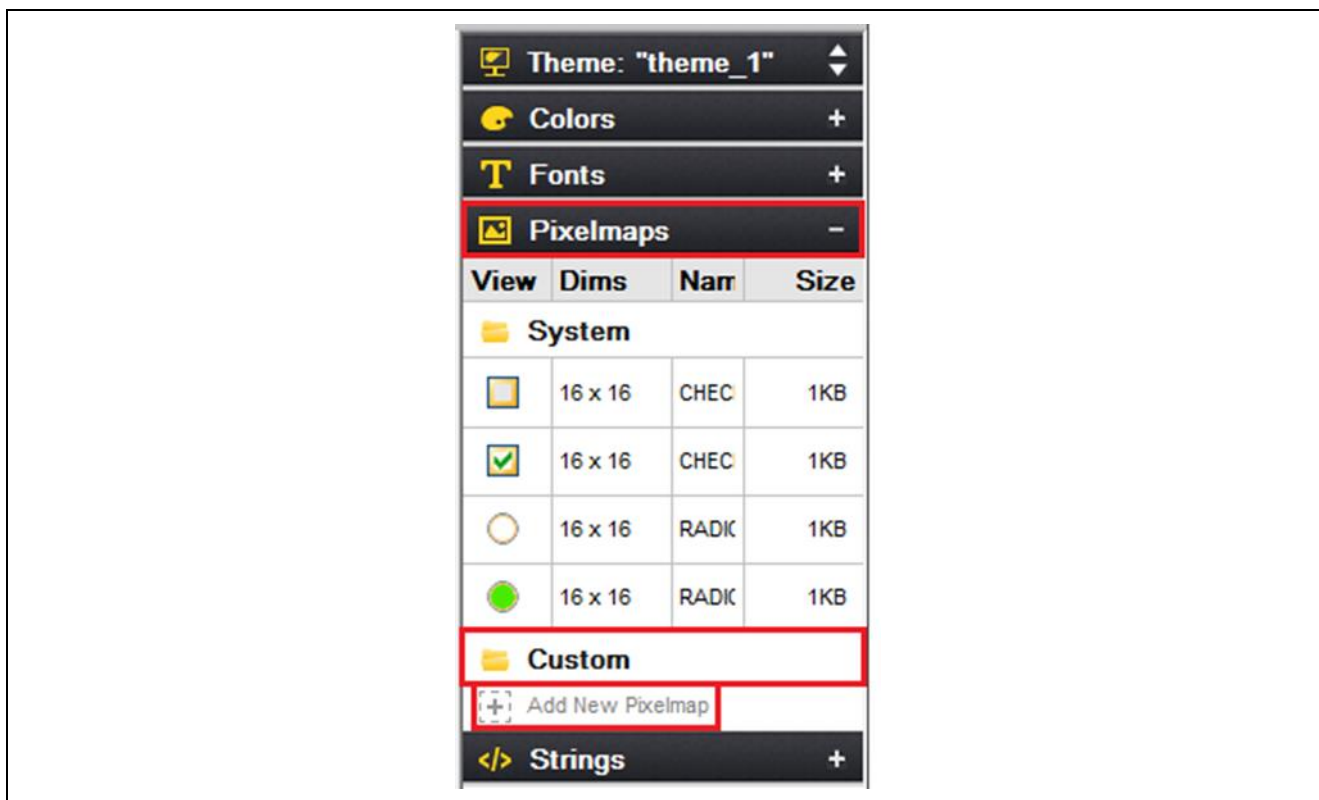


図 8 ピクセルマップリソースの追加

ここで、画像ファイルを選択するためのウィンドウが開きます。GUIX Studio に画像ファイルをインポートするためのディレクトリを指定します。以下の図は、ディレクトリを指定した後の表示の例です。

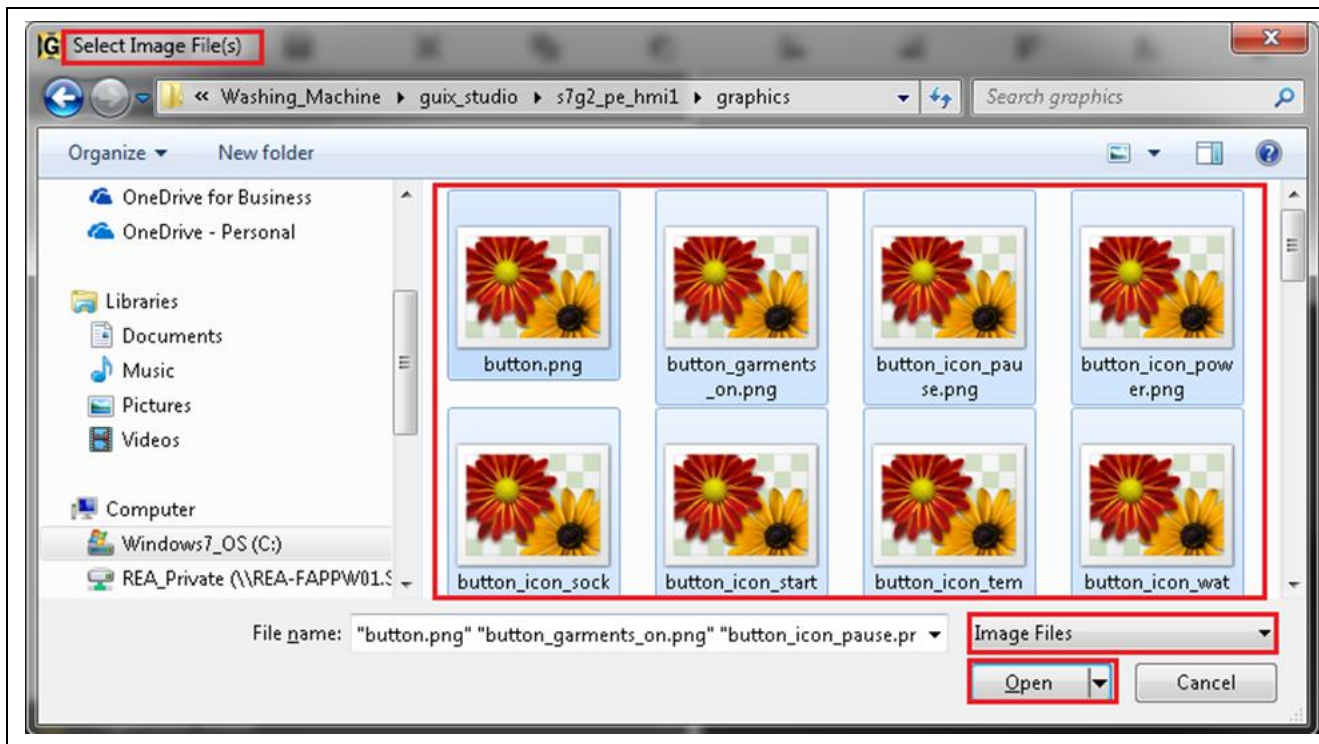


図 9 選択した画像ファイルをピクセルマップリソースに追加

画像ファイルを GUIX Studio にインポートした後、次の図のように、GUIX Studio のリソースとして、[Custom] ピクセルマップの下にそれらのファイルが表示されます。

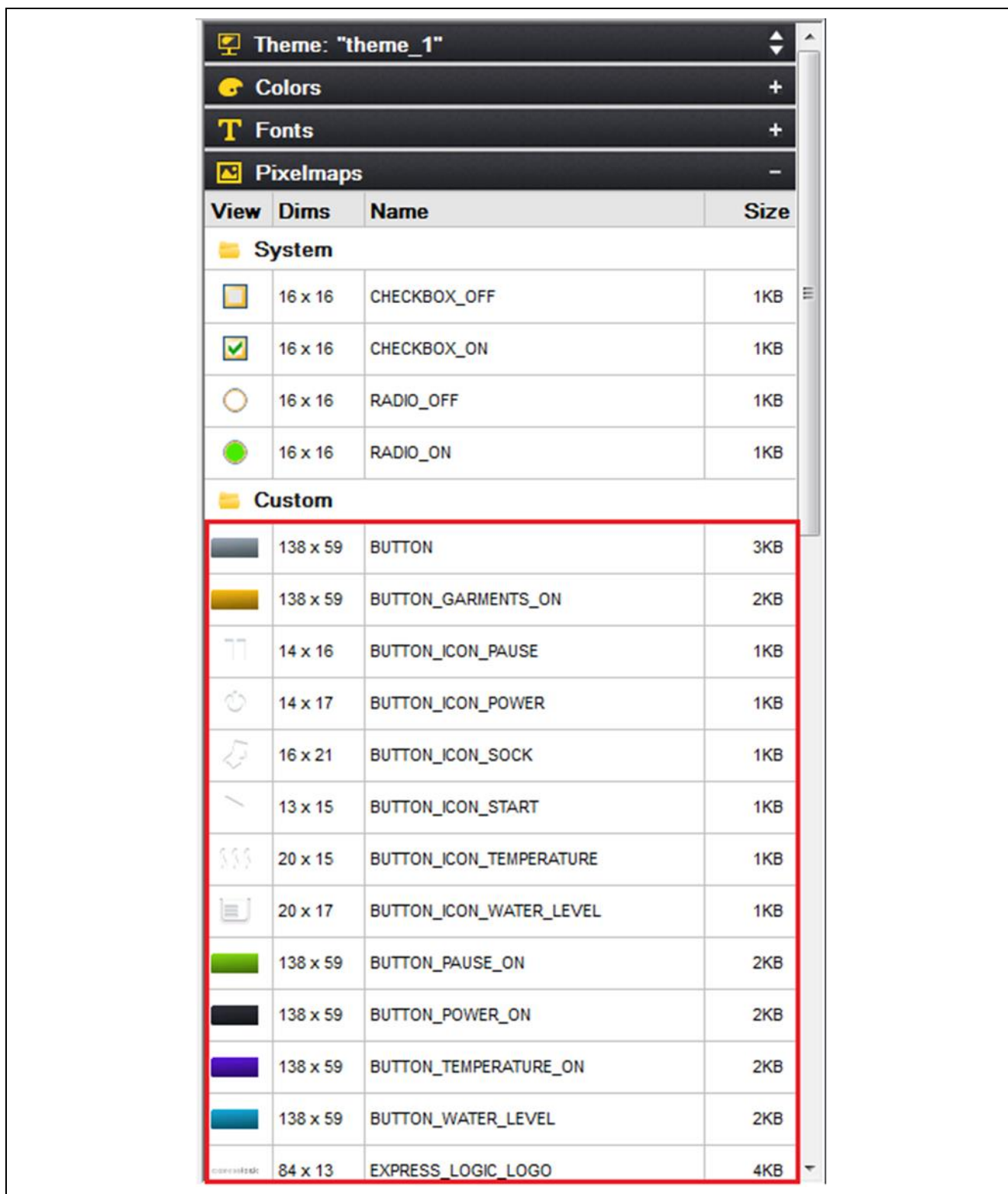


図 10 追加したピクセルマップリソース

3.4.2.2 色リソースの追加 (Adding Colors resources)

GUIX Studio が持つデフォルト色に加え、必要とする色をプロジェクトに追加することもできます。新しい色をプロジェクトに追加するには、[Colors] の横の[+] 記号をクリックし、[+] (Add New Color、新しい色の追加) を選択します。この結果、[Color Name] を選択するためのウィンドウがポップアップ表示され、RGB の値を設定して最適な色を決定します。必要な色を決定した後、その色を保存します。洗濯機アプリケーションでは、新しい色が必要です。新しい色 (SILVERY、銀白色) を追加する例を次の図に示します。色を追加した後は、GUIX Studio で設計する画面でそれらの色を使用できるようになります。

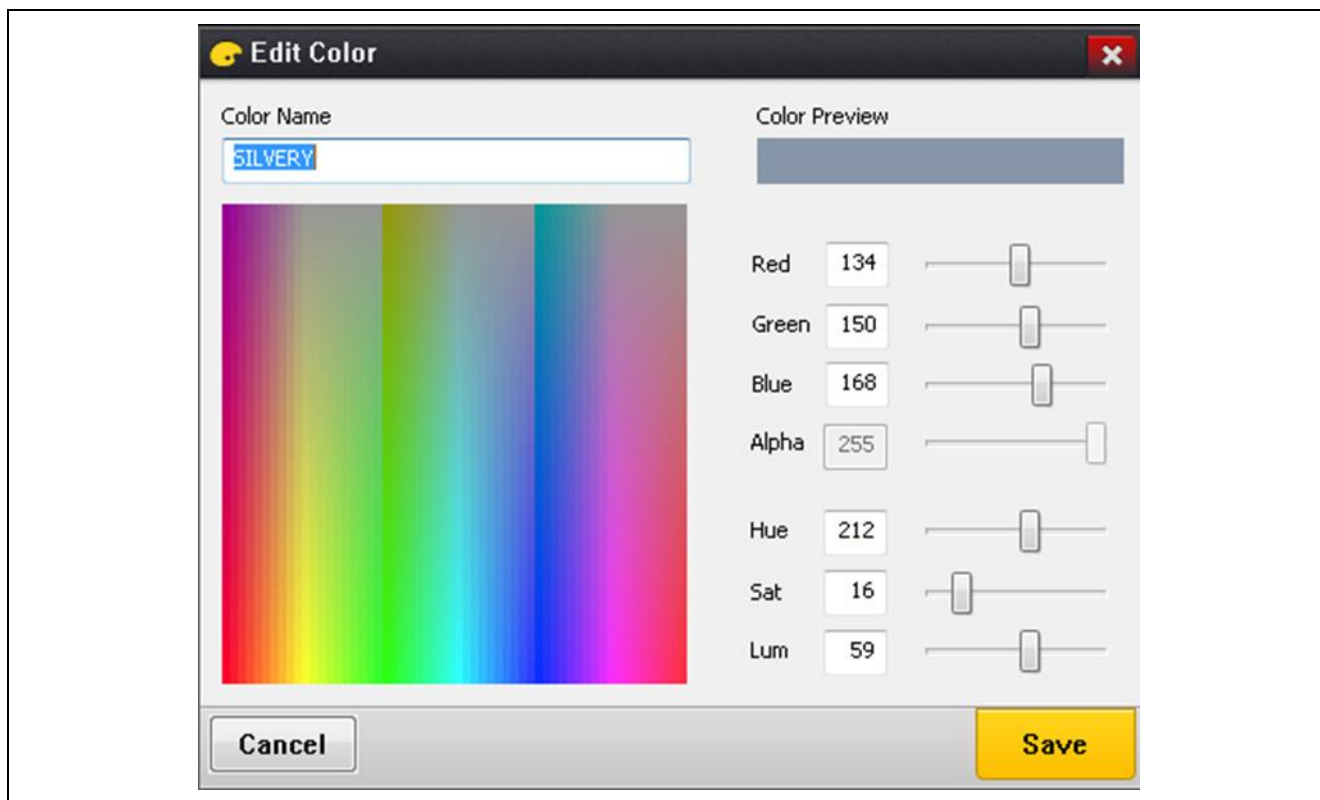


図 11 新しい色リソースの追加

3.4.3 Project View を使用したメイン画面の作成 (Creating the main screen under Project View)

初期のプロジェクト設定が完了し、リソースファイルを GUIX Studio に追加した後、プロジェクトに必要なさまざまなウィンドウ (画面) の作成が可能になります。次の図は、ディスプレイに新しいウィンドウを追加する際のスナップショットです。最初の項目を右クリックし、[main_display] -> [Insert] -> [Window] -> [Window] を選択します。この結果、ディスプレイに新しいウィンドウが追加されます。

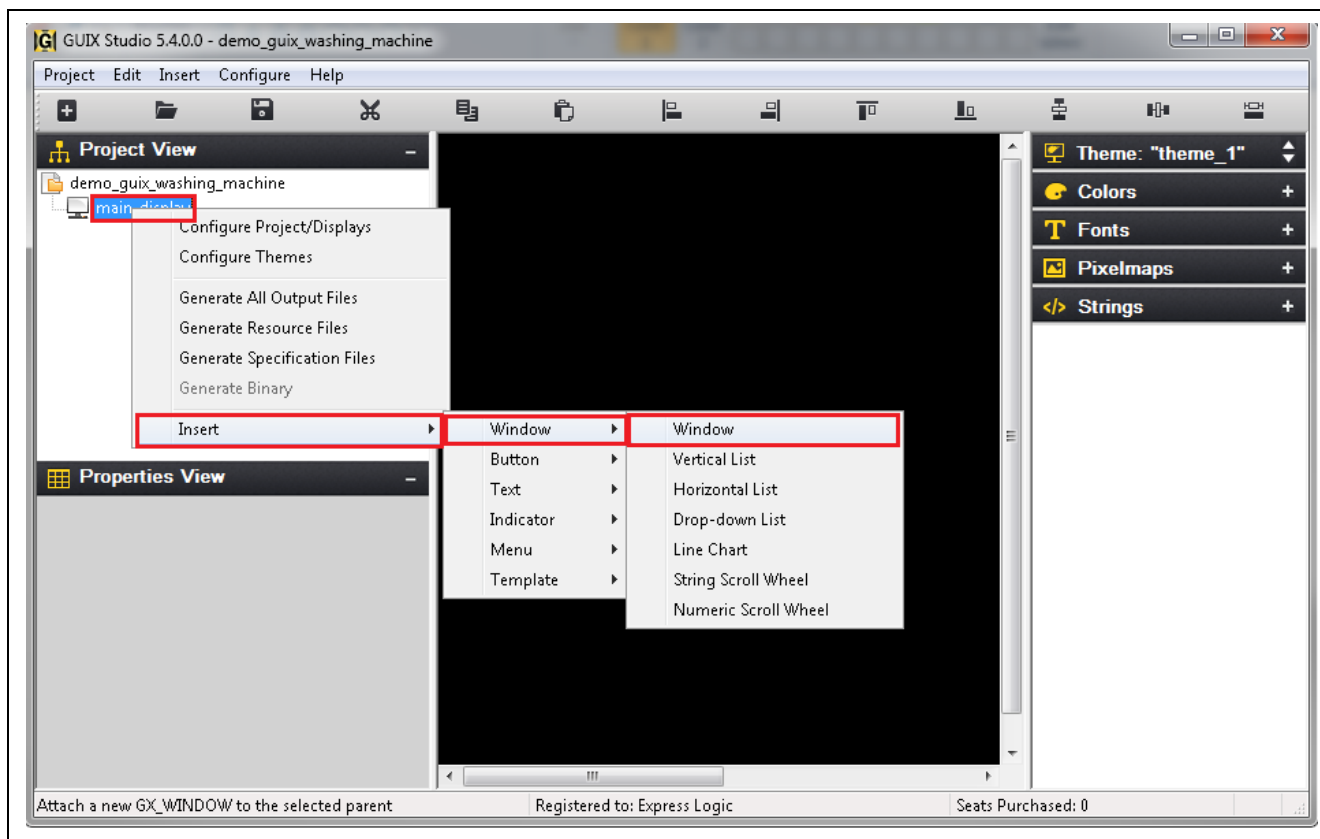


図 12 ディスプレイへの新しいウィンドウの追加

3.4.3.1 プロパティビューを使用したウィンドウのプロパティの変更 (Changing the window properties using Properties View)

追加したウィンドウに対し、プロパティビューを使用してウィンドウのプロパティを変更することができます。デフォルトで作成されるウィンドウは、次の図に示すように親ウィンドウの半分のサイズです。

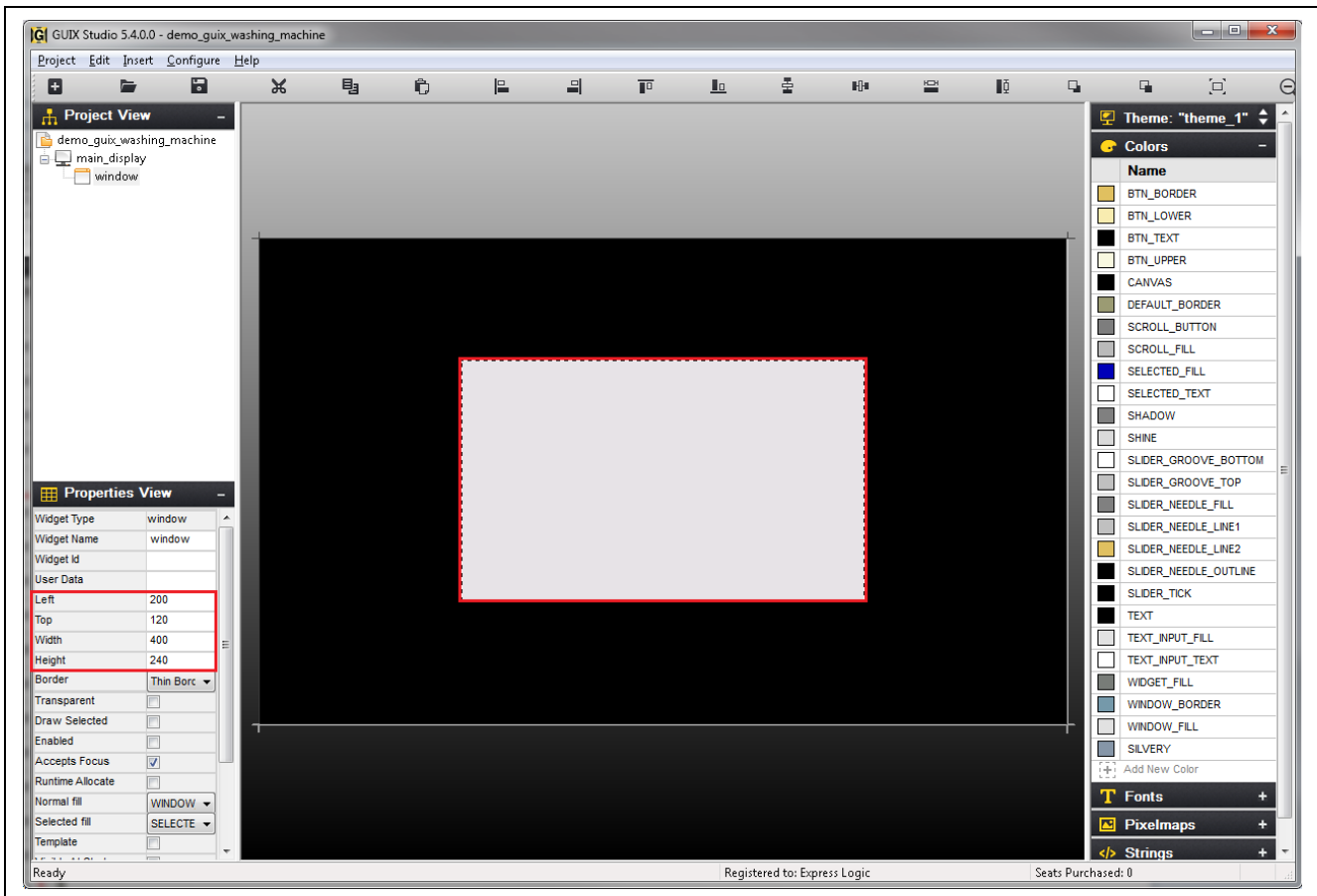


図 13 新しいウィンドウのプロパティの変更

洗濯機アプリケーションの場合、画面全体に表示されるようにウィンドウのサイズを変更する必要があります。[Height] パラメータを 480、[Width] を 800、[Left] を 0、[Top] を 0 に変更することで実現します。さらに、ウィンドウの色を変更する必要があります。洗濯機アプリケーションでは、ウィンドウの色を黒 (背景色) にする必要があります。この色を変更するには、GUIX Studio の [Resources] にある [Color] 部で、[WINDOW_FILL] (ウィンドウの塗りつぶし) 色を変更します。最初の項目を右クリック、[WINDOW_FILL] -> [Edit Color] を選択した後、[Red]、[Green]、[Blue] の各値すべてに「0」を指定すると黒色になります。[WINDOW_FILL] に変更を加えるサンプルのスナップショットを、次の図に示します。

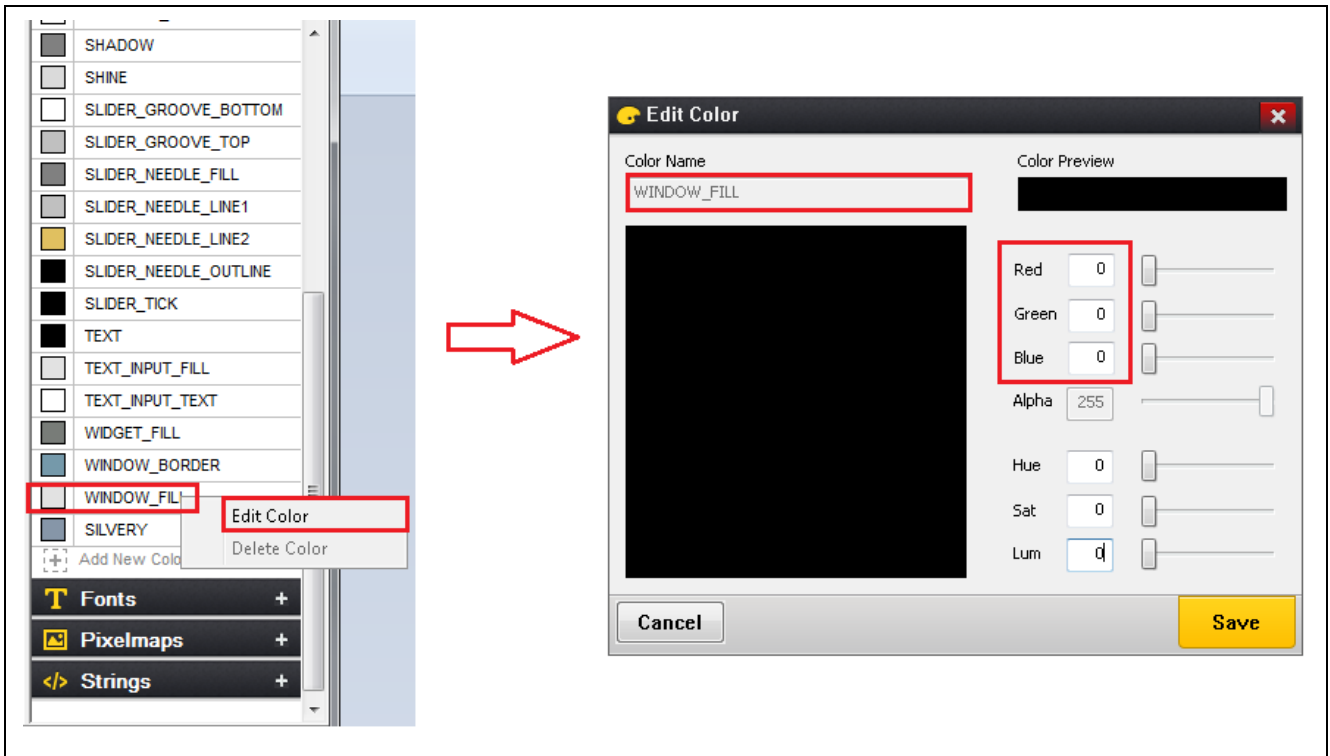


図 14 ウィンドウの塗りつぶしに使用する色の編集

次の図は、ウィンドウのプロパティである、[Left]、[Right]、[Top]、[Bottom] と [WINDOW_FILL] を変更した後の状態を示しています。

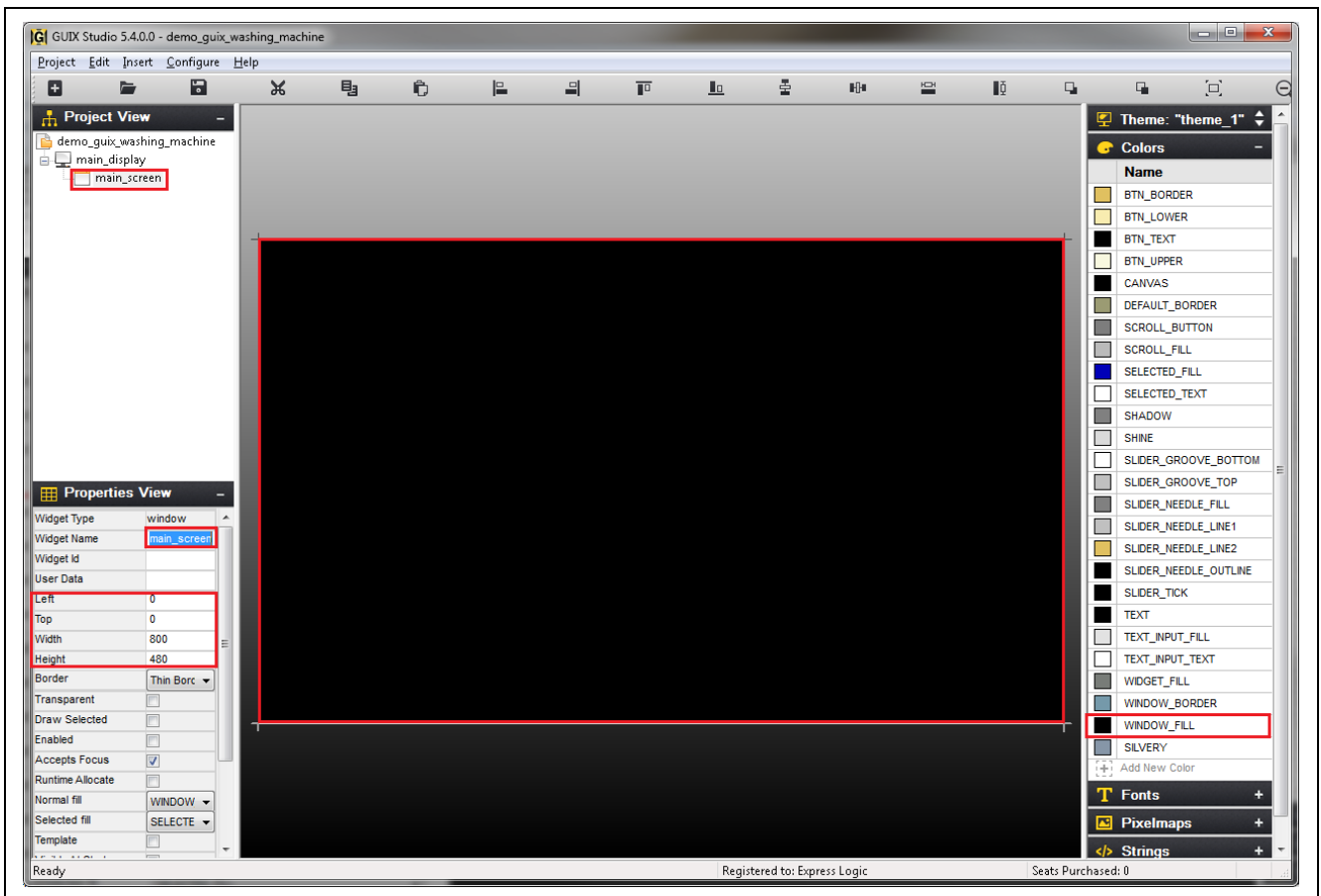


図 15 メイン画面の背景ウィンドウ

背景ウィンドウより手前に、洗濯機アプリケーションで使用するさまざまなウィジェットを追加する必要があります。

- GUIX のロゴ
- Express Logic のロゴ
- 時刻と日付の情報
- ホームボタン
- 制御ボタン

洗濯方法の選択肢を示す [Radial Slider] (円形のスライダ) ウィンドウと [Status] ウィンドウなども定義します。

3.4.3.2 メイン画面への GUIX ロゴと Synergy ロゴの追加 (Adding GUIX and Synergy logo to the main screen)

GUIX のロゴと Express Logic のロゴを画面に追加するには、ターゲットのウィンドウを右クリックし、[Insert] -> [Button] -> [Icon] を選択します。この結果、次の図に示すように、アイコンウィジェットが画面に追加されます。

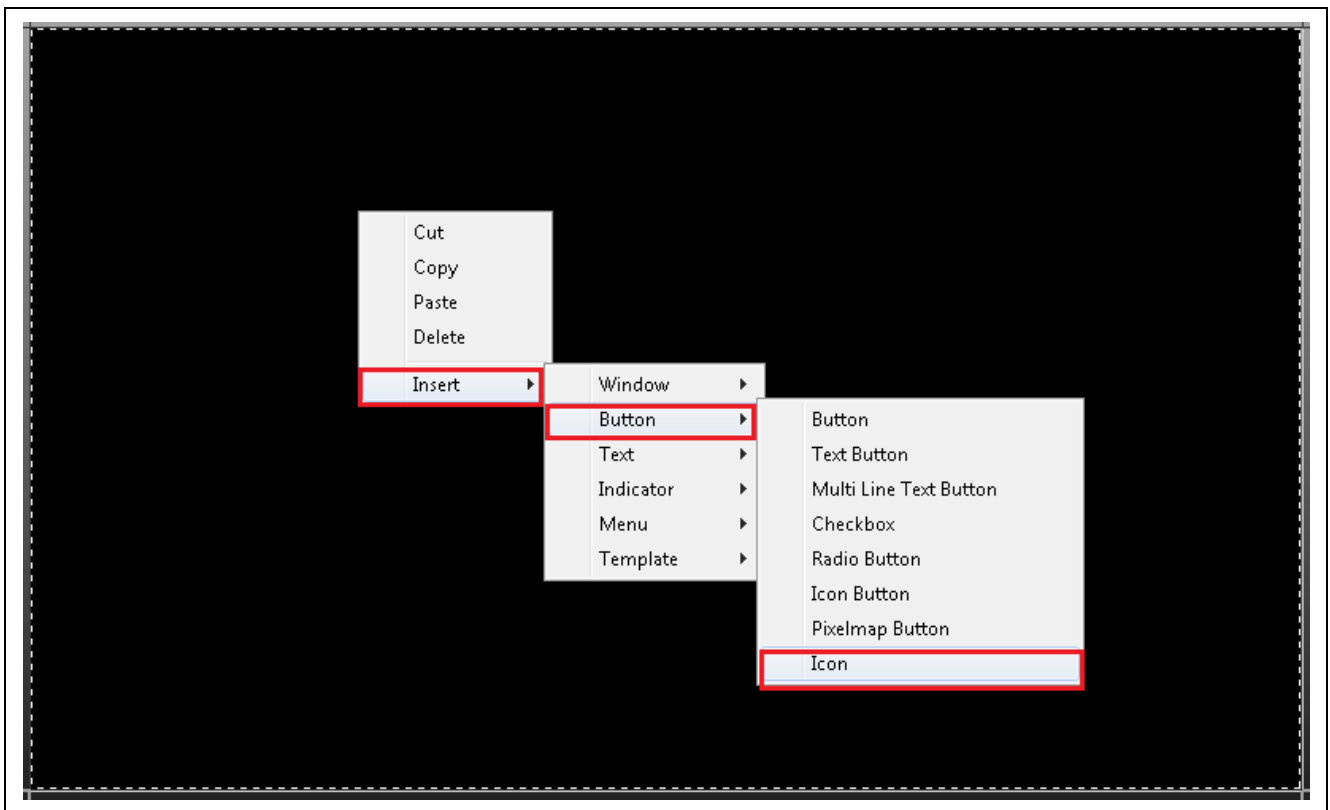


図 16 ウィンドウへのボタンアイコンの挿入

このアイコンは、GUIX のロゴのプレースホルダです。ここで、[Pixelmap] リソースから [GUIX_LOGO_VERTICAL] をターゲットウィンドウにドラッグし、次の図に示すように、新しく作成したアイコンに重なるように配置します。

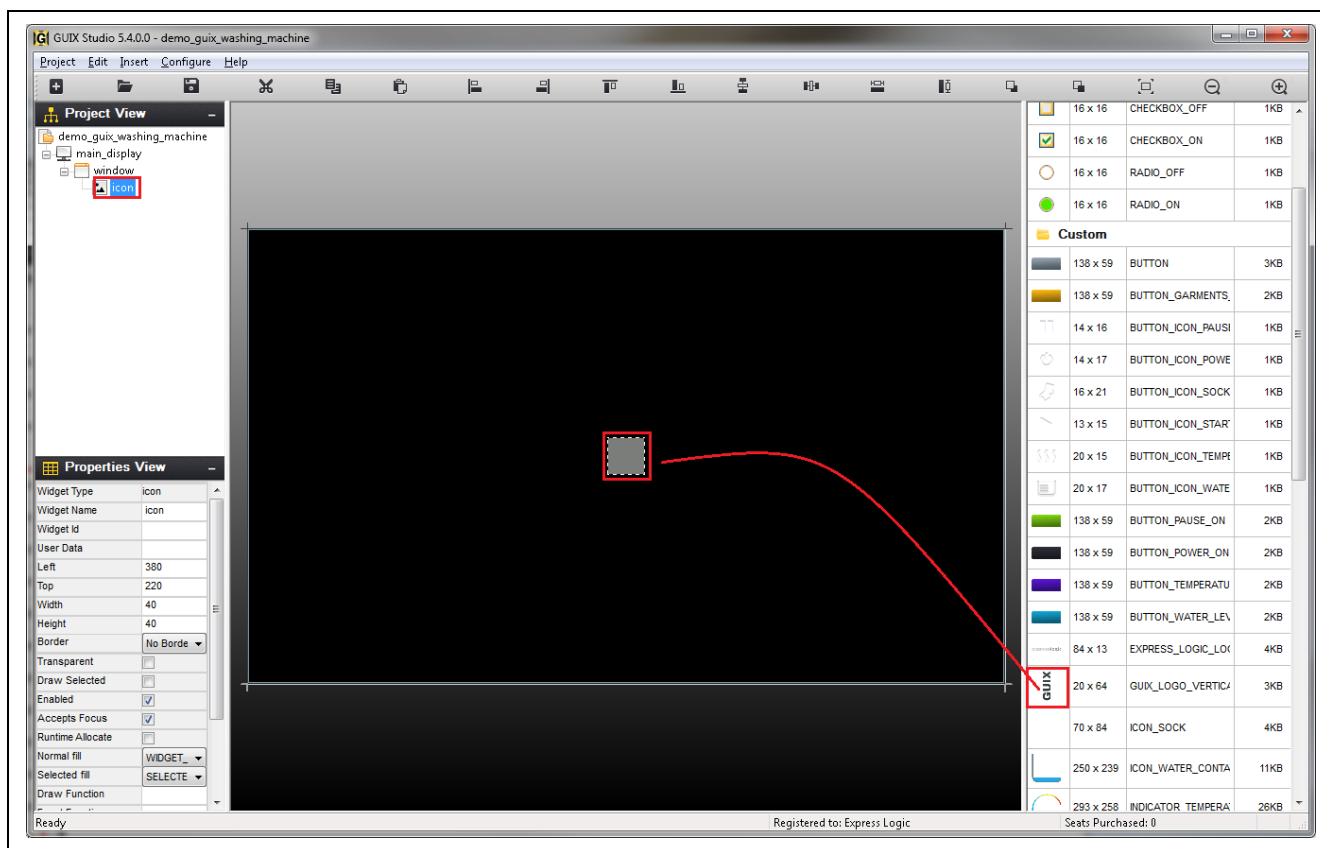


図 17 ボタンアイコンへの GUIX ロゴの追加

次の図に示すように、ターゲットウィンドウに GUIX のアイコンが表示されます。このアイコンを左上隅に移動し、アイコンウィジェットの名前を「guix」に変更する必要があります。ターゲットウィンドウで新しく作成したアイコンをクリックし、GUIX アイコンを左上隅にドラッグします。プロパティビューで、ウィジェットの名前を「icon」から「guix」に変更します。次の図に示すように、最終的なウィンドウが表示されます。

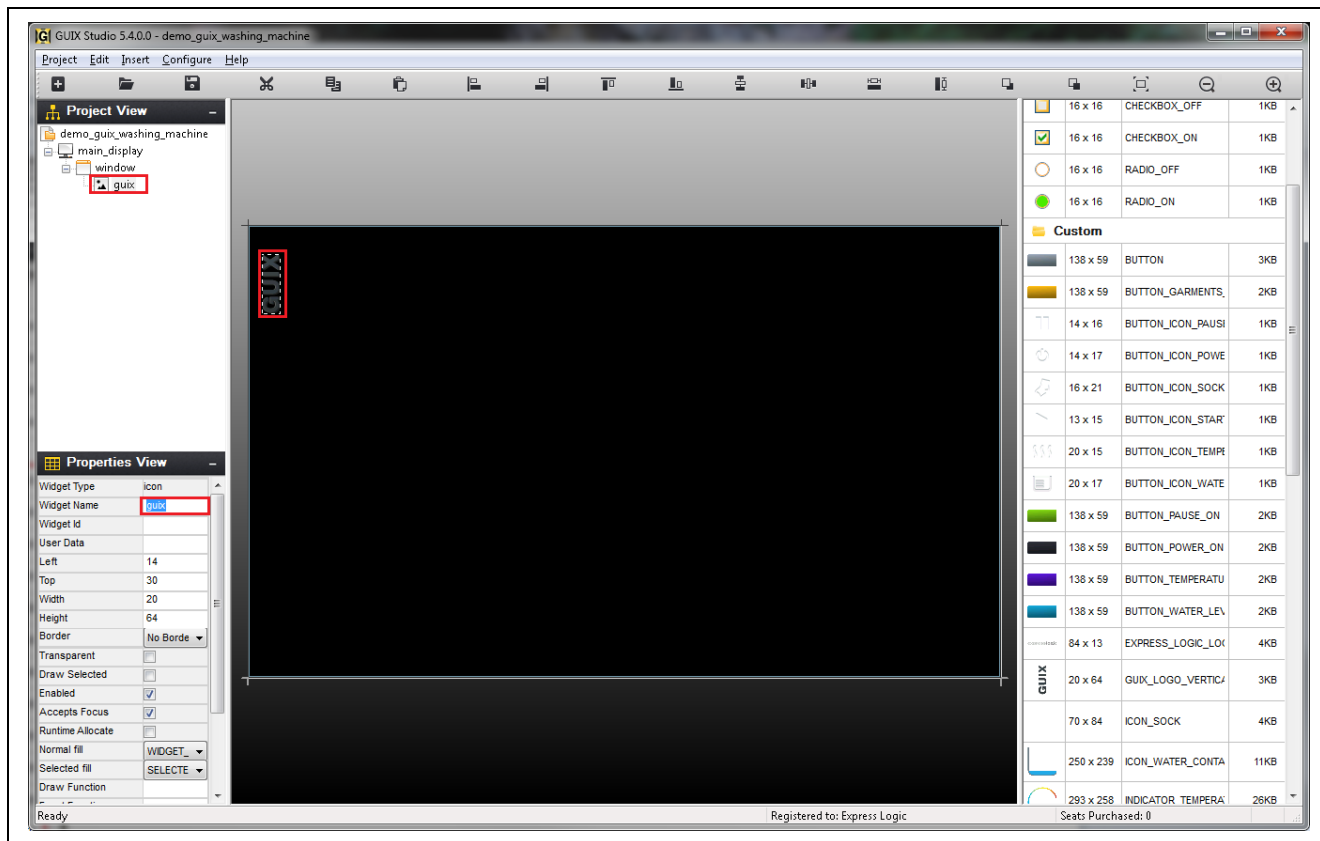


図 18 ボタンアイコンに追加した GUIX ロゴ

同様の方法により、Express Logic のロゴを画面に追加することができます。

3.4.3.3 メイン画面への時刻と日付情報の追加 (Adding Time and Date info to the main screen)

時刻と日付情報を、テキストウィジェット (text widget) としてメイン画面に追加します。テキストウィジェットを追加するには、ターゲットウィンドウを右クリックし、[insert] -> [text] -> [prompt] を選択します。この結果、次の図に示すように、プロンプト (prompt) が画面に追加されます。

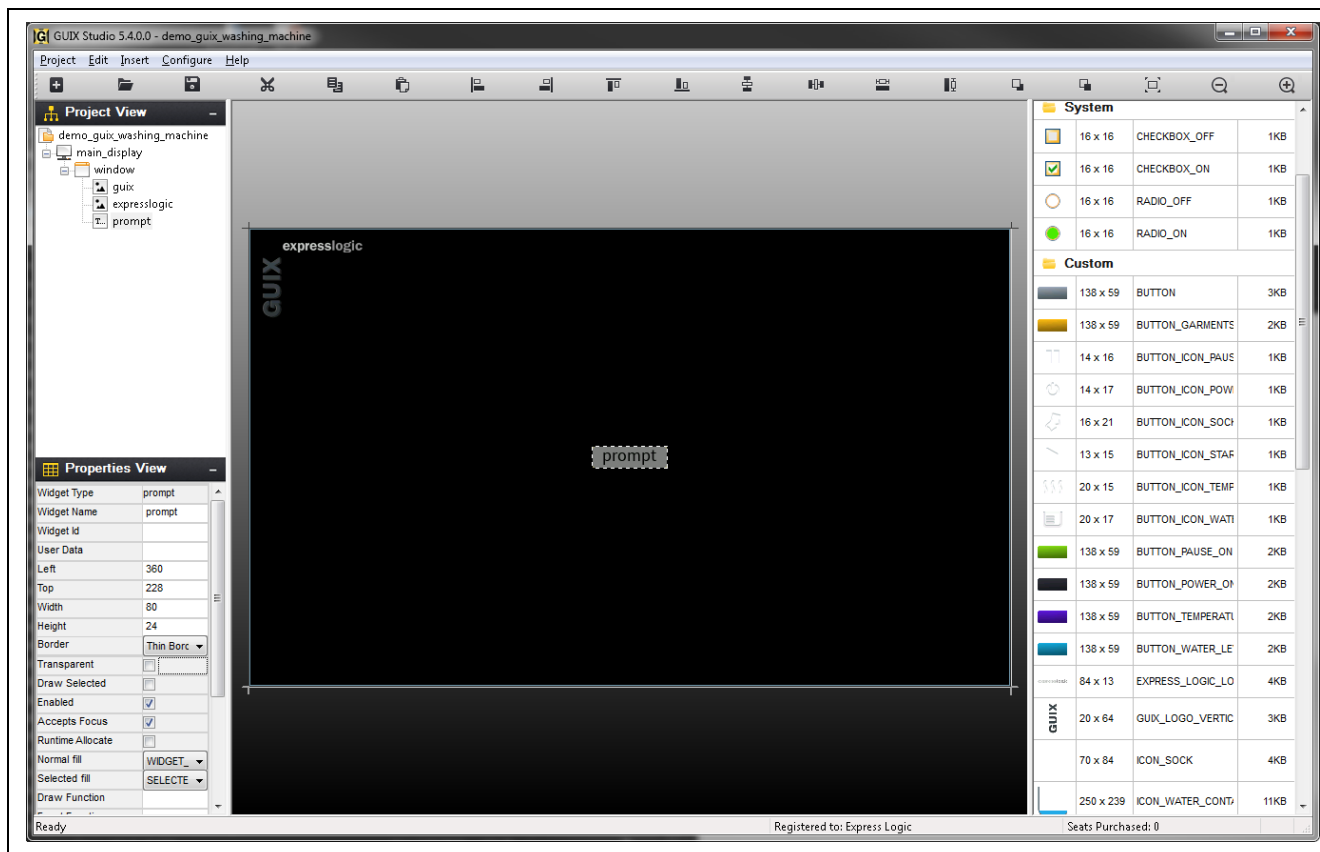


図 19 メイン画面へのテキストウィジェットの追加

洗濯機アプリケーションの場合、テキストプロンプト (時刻と日付) の背景色を透明、縁は色なし、テキストの色を銀色に設定します。時刻プロンプトの位置は、上端中央に調整します。次の図に示すように、プロパティビューで選択肢を変更することで設定できます。

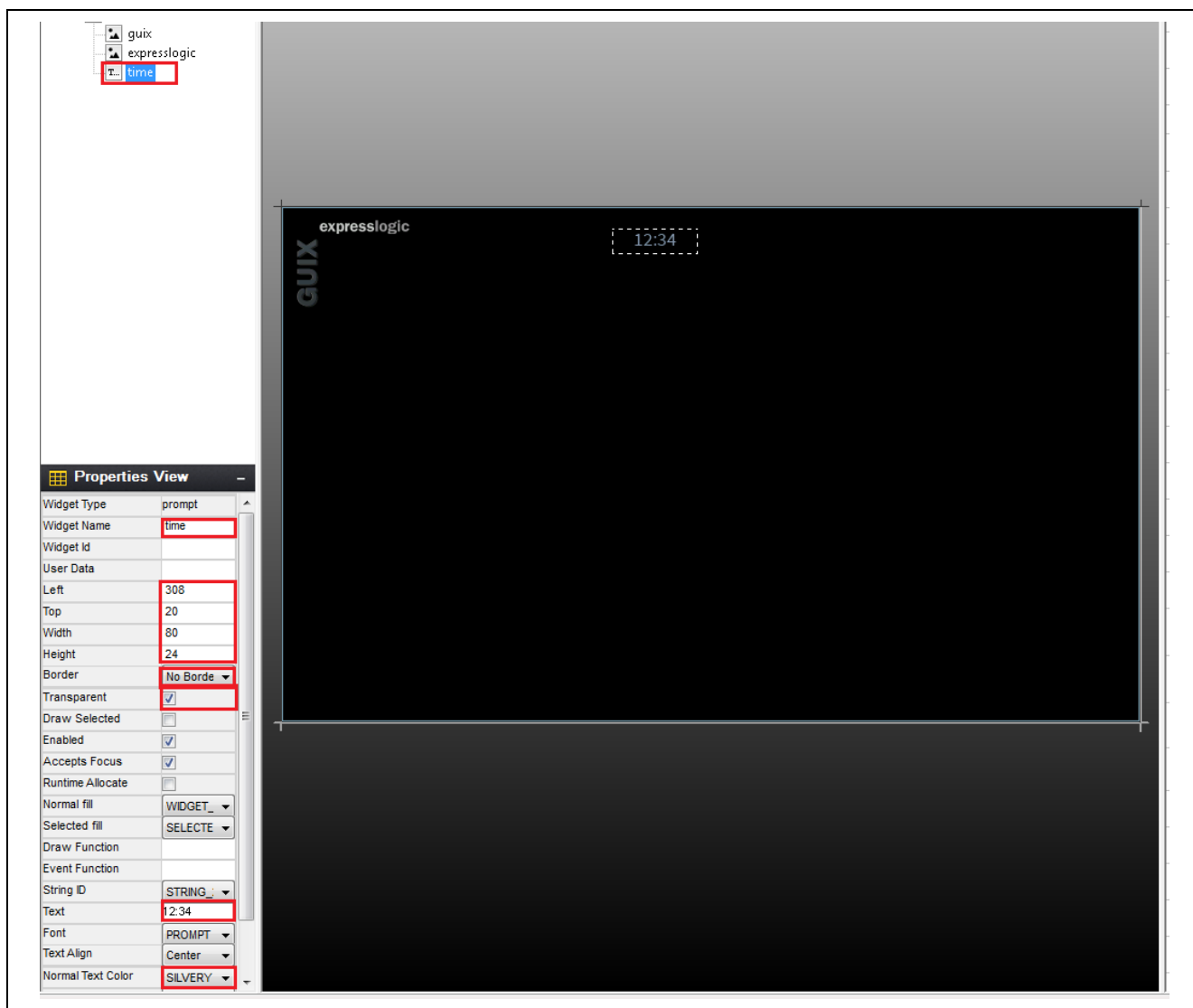


図 20 プロパティビューを使用してテキストウィジェットを変更

同様の方法で、AM/PM (午前/午後) の選択や、曜日、日付に関連する各種テキストウィジェットを追加することができます。これらのテキストウィジェットを追加した後、次の図の画面のようになります。詳細については、付属している GUIX Studio プロジェクトとそのプロパティビューを参照してください。

注記：他のテキストウィジェットの追加は、前述の時刻ウィジェットの追加と同様の方法で実現できます。

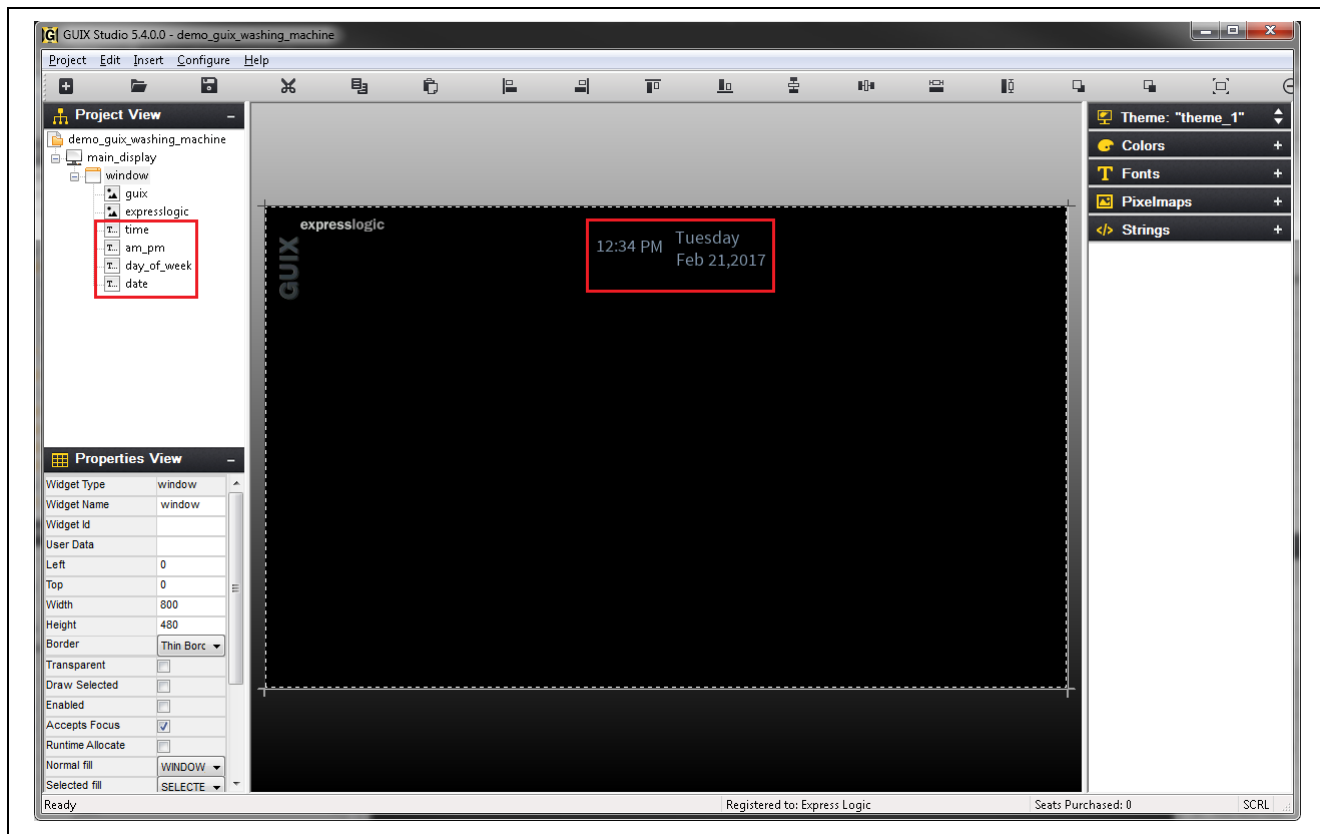


図 21 時刻と日付のテキストウィジェットをメイン画面に追加

3.4.3.4 メイン画面へのホームボタンの追加 (Adding Home Button to the main screen)

ホームアイコンは、ピクセルマップボタン (pixel map button) であり、ターゲットスクリーンを右クリックし、[Insert] -> [Button] -> [Pixelmap Button] を選択する方法で追加できます。

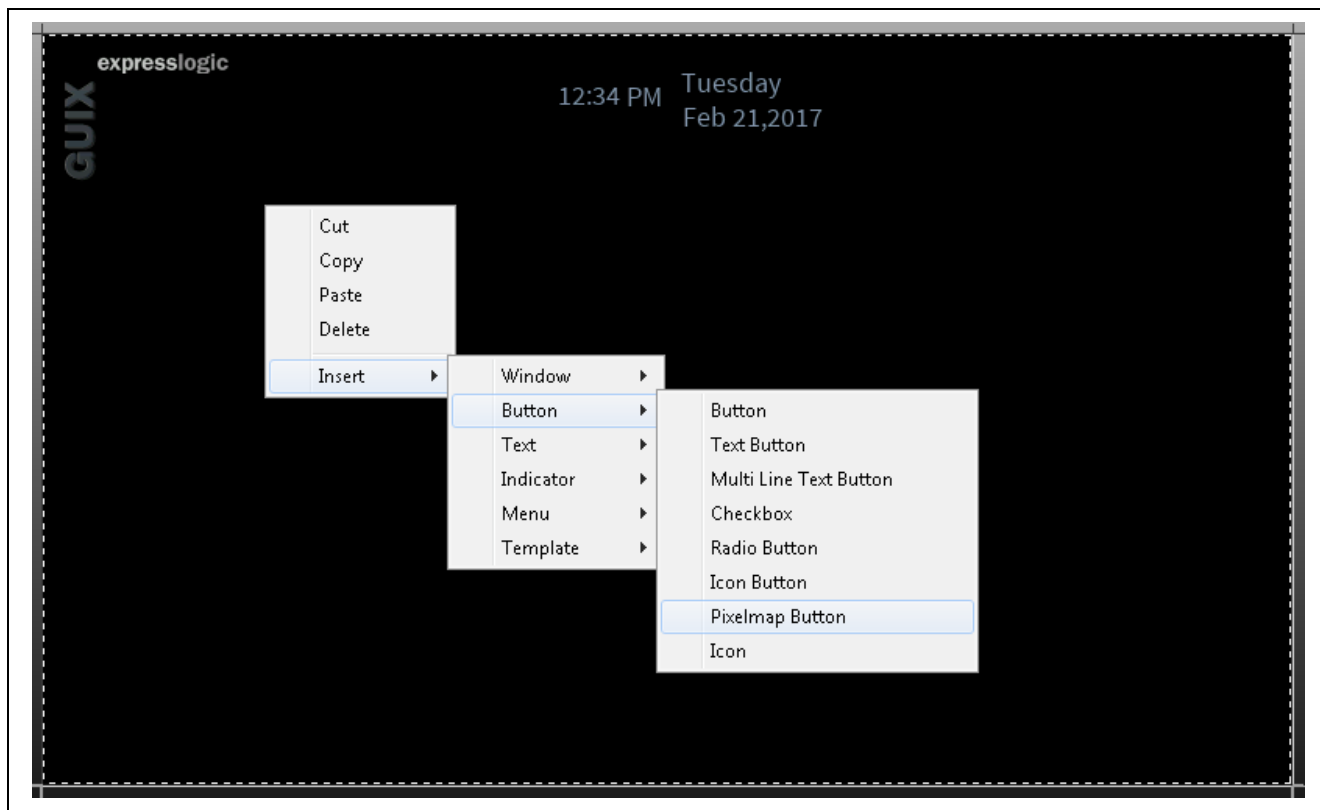


図 22 メイン画面へのピクセルマップボタンの追加

ピクセルマップリソースから [MENU_ICON_HOME] をドラッグし、ピクセルマップボタンに重なるように配置します。そのアイコンを画面の右上隅に移動します。この操作により、次の図に示す画面の状態になります。

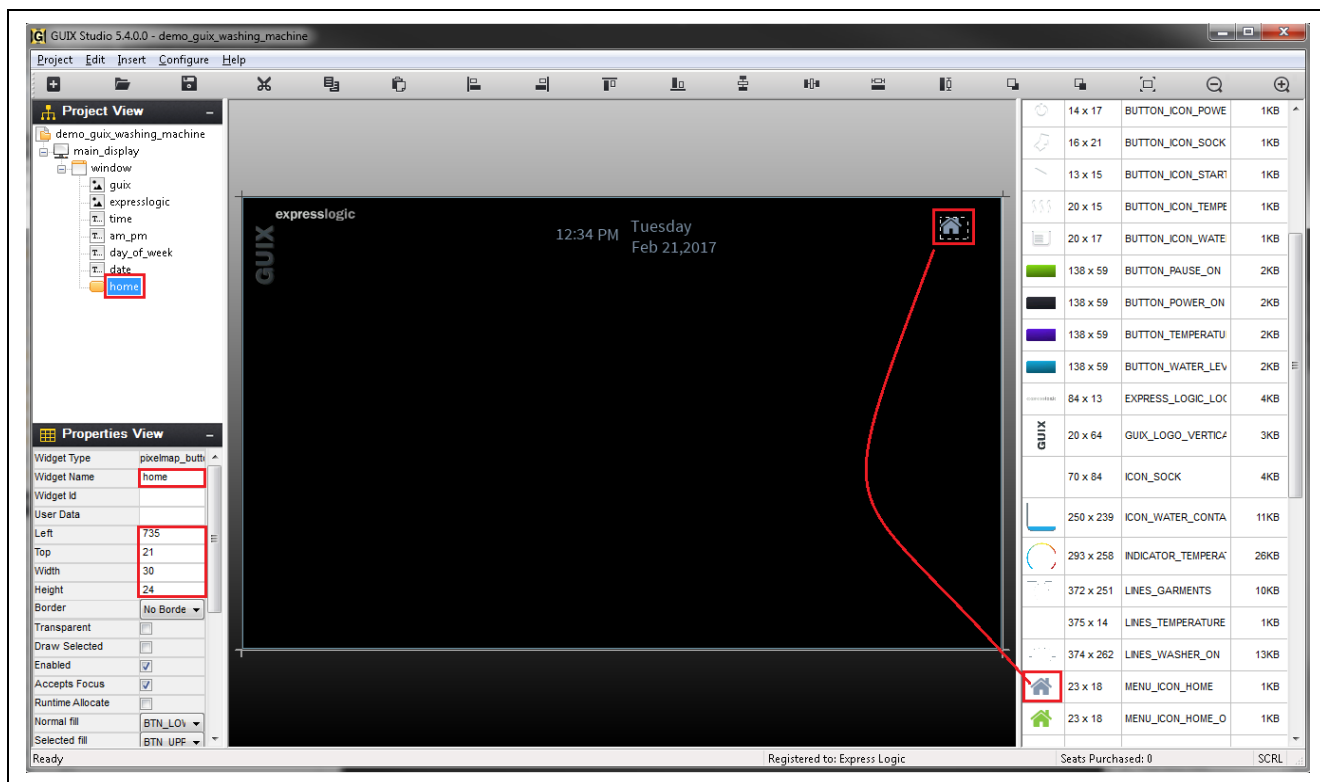


図 23 ピクセルマップボタンへのホームアイコンの追加

3.4.3.5 メイン画面への複数のボタンの追加 (Adding Buttons to the main screen)

この章では、洗濯機の機能 (washer)、衣類 (garments)、水位 (water level)、温度 (temperature)、電源オン/オフ (power on/off) を選択するためのボタンの追加を説明しています。

[Washer] (洗濯機) ボタンは、一時停止を意味するアイコンと「pause」(一時停止) という文字を含むピクセルマップボタンです。このボタンを追加するには、ターゲットウィンドウを右クリックし、[insert] -> [Button] -> [Pixelmap Button] を選択します。この結果、[Target View] に黄色いボタンが追加されます。[Properties View] に移動し、[Normal Pixelmap] フィールドを [BUTTON] に変更し、[Selected Pixelmap] (選択状態のピクセルマップ) フィールドを [BUTTON_PAUSE_ON] (一時停止状態のボタン) に変更します。また、[Pushed] と [Radio] の各フィールドに対応するチェックボックスをオンにします。通常のピクセルマップの代わりに、選択状態のピクセルマップを表示する方法で、このボタンはプッシュ状態になります。このボタンを [Radio] に設定したので、このボタンが選択されている状態では、他のいずれのボタンも選択解除されます。Studio のボタンバーで、[size to content] (コンテンツに合わせてサイズ調整) をクリックします。この結果、ボタンは指定したサイズで表示されるようになります。ボタンの配置先である画面の右上隅にこのボタンを移動します。作成済みのアイコンを右クリックし、[Insert] -> [Button] -> [Icon] を選択します。

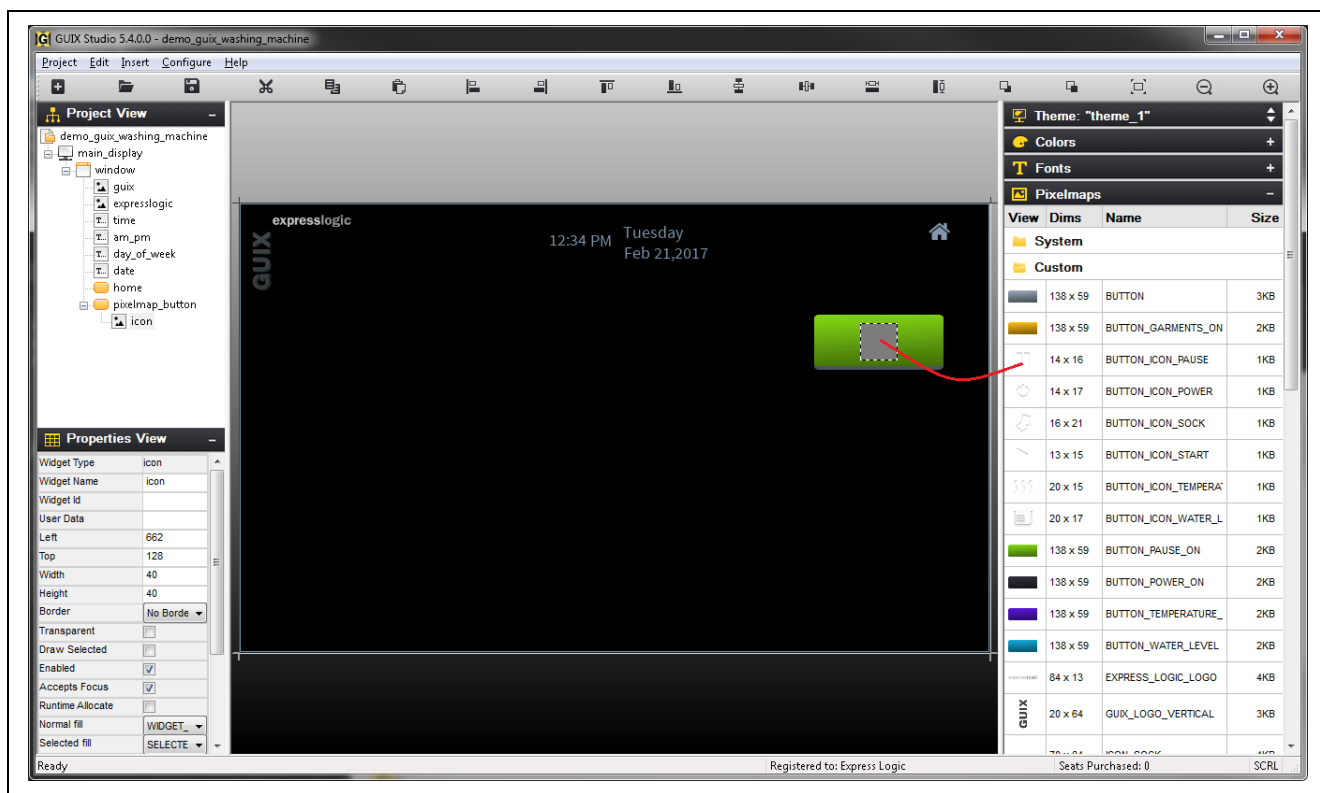


図 24 ピクセルマップボタンへの [Pause] (一時停止) アイコンの追加

[BUTTON_ICON_PAUSE] をドラッグし、上の図に示すように、作成したアイコンの手前に配置します。[pause] (一時停止) プロンプトをボタンに追加するために、ボタンを右クリックし、[insert] -> [text] -> [prompt] を選択します。

プロンプトの [Properties View] で、テキストフィールドを [PAUSE] (一時停止) に変更し、[Transparent] フィールドを選択し、[Border] フィールドを [No Border] (境界線なし) に変更します。[Normal Text Color] に [WHITE] を選択します (この色が見つからない場合、色を作成してください)。

次の図に示す [Properties View] のスナップショットは、[Washer Button] (洗濯機の機能ボタン)、[Pause Icon] (一時停止アイコン)、および [Pause Prompt] (一時停止プロンプト) に対応しています。

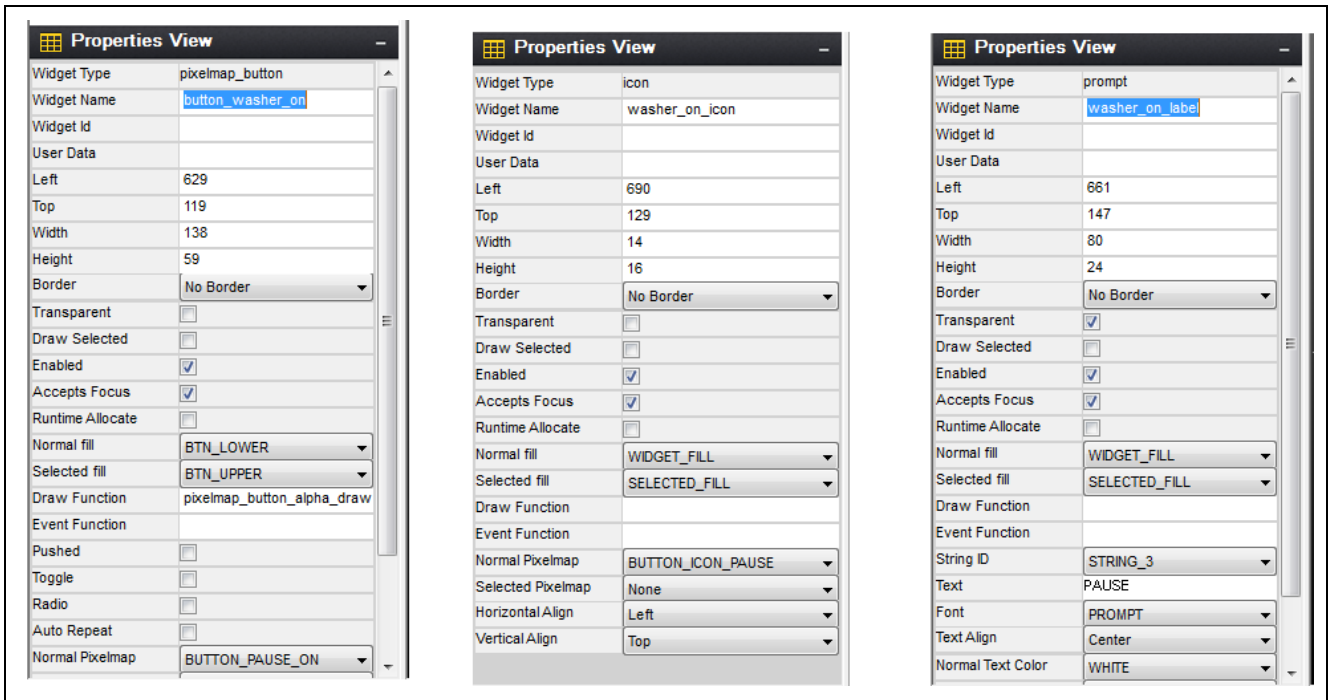


図 25 プロパティビュー - Washer ボタン、Pause アイコン、Washer プロンプト

[Washer] (洗濯機の機能) ボタンに対応するプロパティビューを変更した後、ターゲットウィンドウの [Washer] (洗濯機の機能) ボタンは、次の図のようになります。

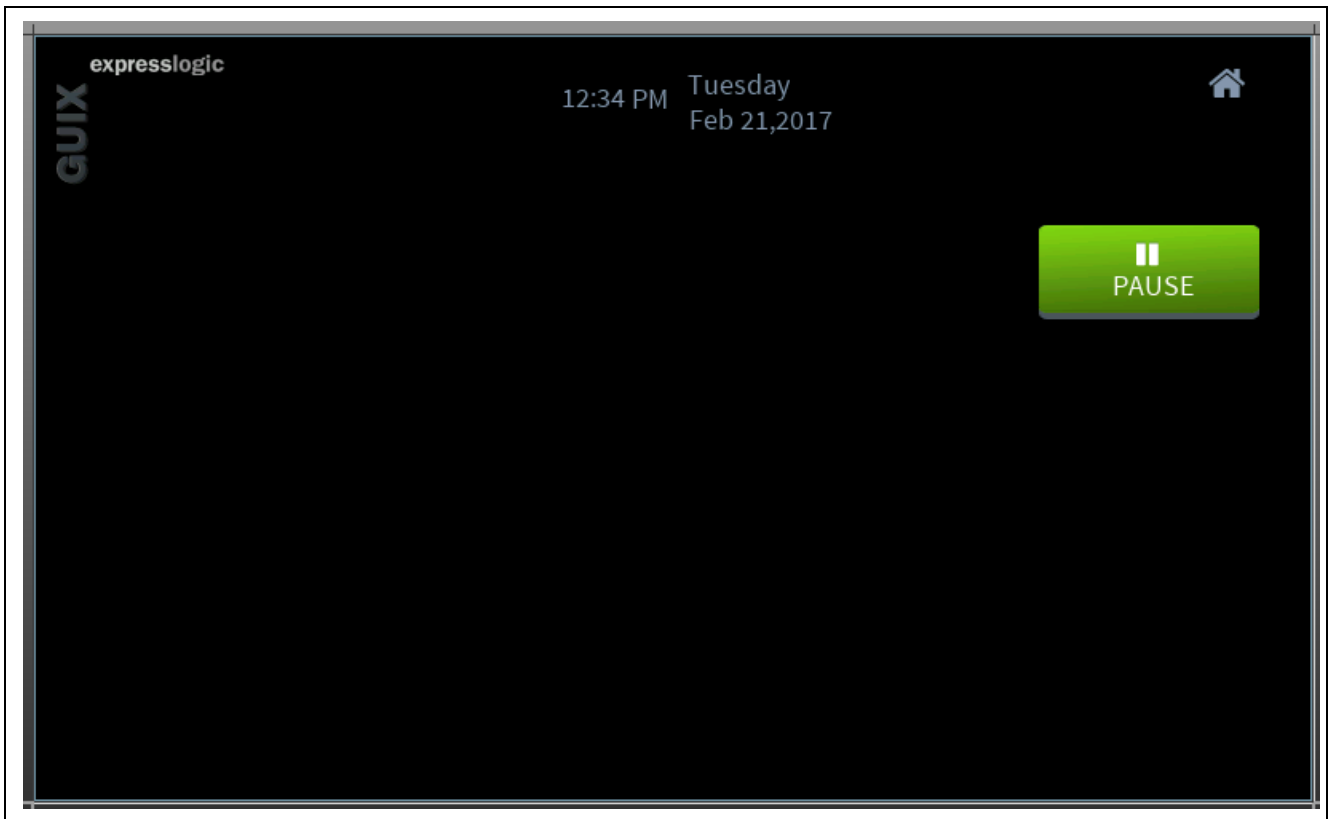


図 26 メイン画面への [Washer Button] (洗濯機の機能ボタン)の追加

同様の方法で、他のボタンをメイン画面に追加することができます。個別のボタンのプロパティビューに関する詳細については、付属している GUIX Studio プロジェクトを参照してください。

3.4.3.6 メイン画面への [Washer] (洗濯機の機能) ウィンドウの追加 (Adding Washer window to the main screen)

[Washer] (洗濯機の機能) ウィンドウは、メイン画面の子ウィンドウであり、円形のホイール (円形のスライダ) とプロンプトに対応する複数の線で構成しています。これら 2 つのアイコンが重ね合わせで表示され、[Washer] (洗濯機の機能) ウィンドウを形成しています。各プロンプト線の脇に、洗濯機のさまざまな設定を表すラベルを表示します。

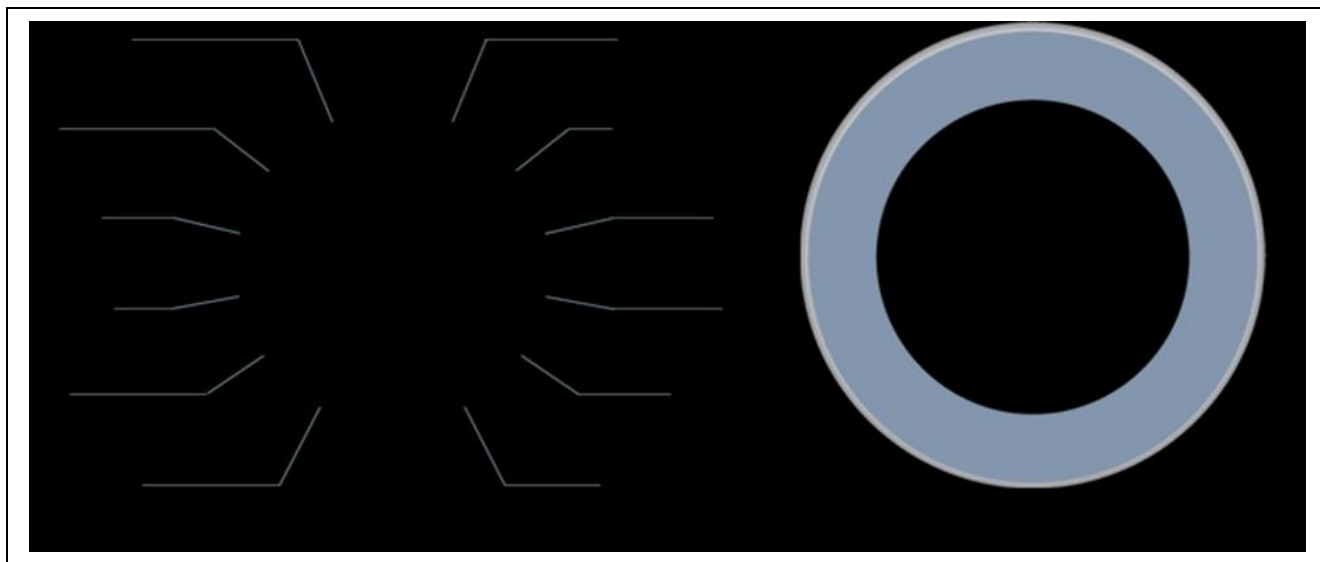


図 27 Radial Slider (円形のスライダ) コンポーネント

画面上のスナップショットに、複数のアイコンを表示します。これらは、[Washer] (洗濯機の機能) ウィンドウを作成するために、GUIX Studio により重ね合わせが行われます。

注記：子ウィンドウ ([Washer Window] (洗濯機の機能ウィンドウ)) は、衣類、水位、温度などのボタン選択状態に応じて着脱されます。洗濯機アプリケーションはこのような方法でアーキテクチャを決定していますが、最新バージョンの GUIX Studio のテンプレートを使用することで、さまざまな方法で実装が可能です。

[Washer Control] (洗濯機の機能制御) インタフェースを作成するには、[Washer] (洗濯機の機能) ウィンドウの下に 2 種類の異なるアイコンを追加し、子ウィジェットをテキストプロンプトとして追加します。これらのアイコンに対応する [Pixelmap] は、[Pixelmap] の下にある [Resource] 部の一部として使用できます。ここではピクセルマップを追加する詳細な方法は記載していません。ウィジェットを追加する方法については前の章を参照してください。まだ同時に、このアプリケーションノートに付属している GUIX プロジェクトもあわせて参照してください。

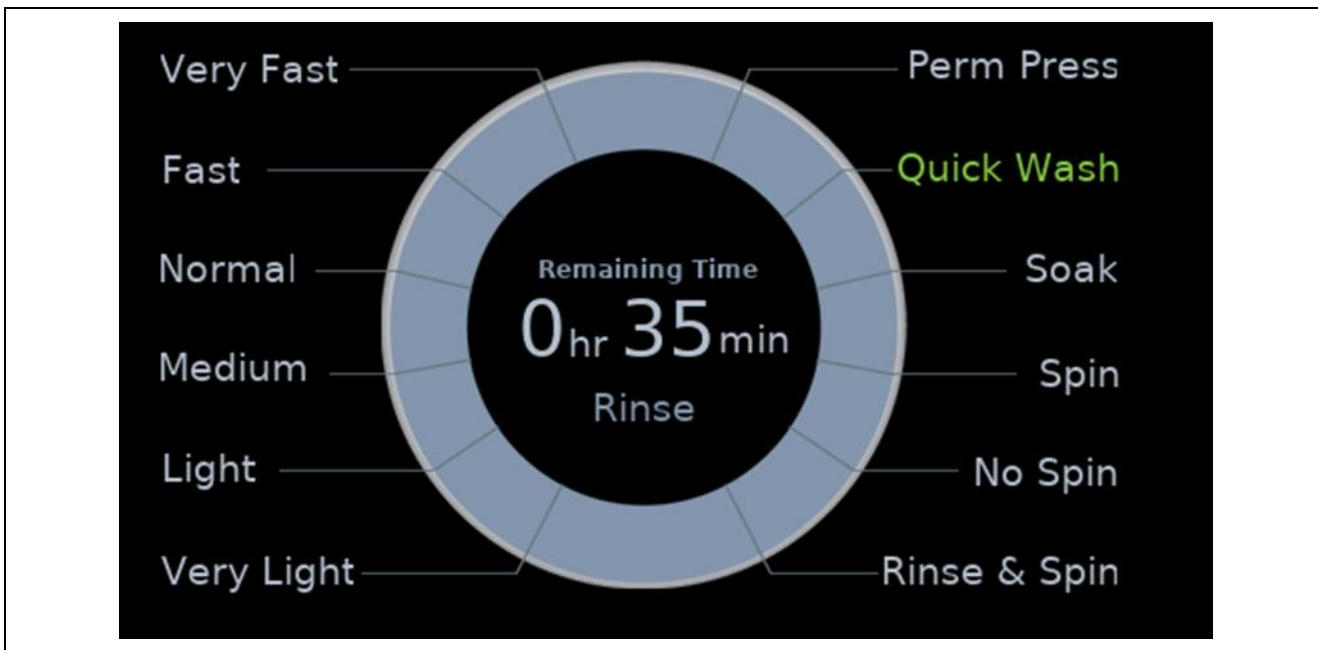


図 28 作成済みの円形のスライダウィンドウ

次の図に、Radial Slider アイコンウィジェット (WHEEL) とプロンプトラインアイコン (LINES_WASHER_ON) に対応した [Washer]ウィンドウの[Properties View]スナップショットを示します。

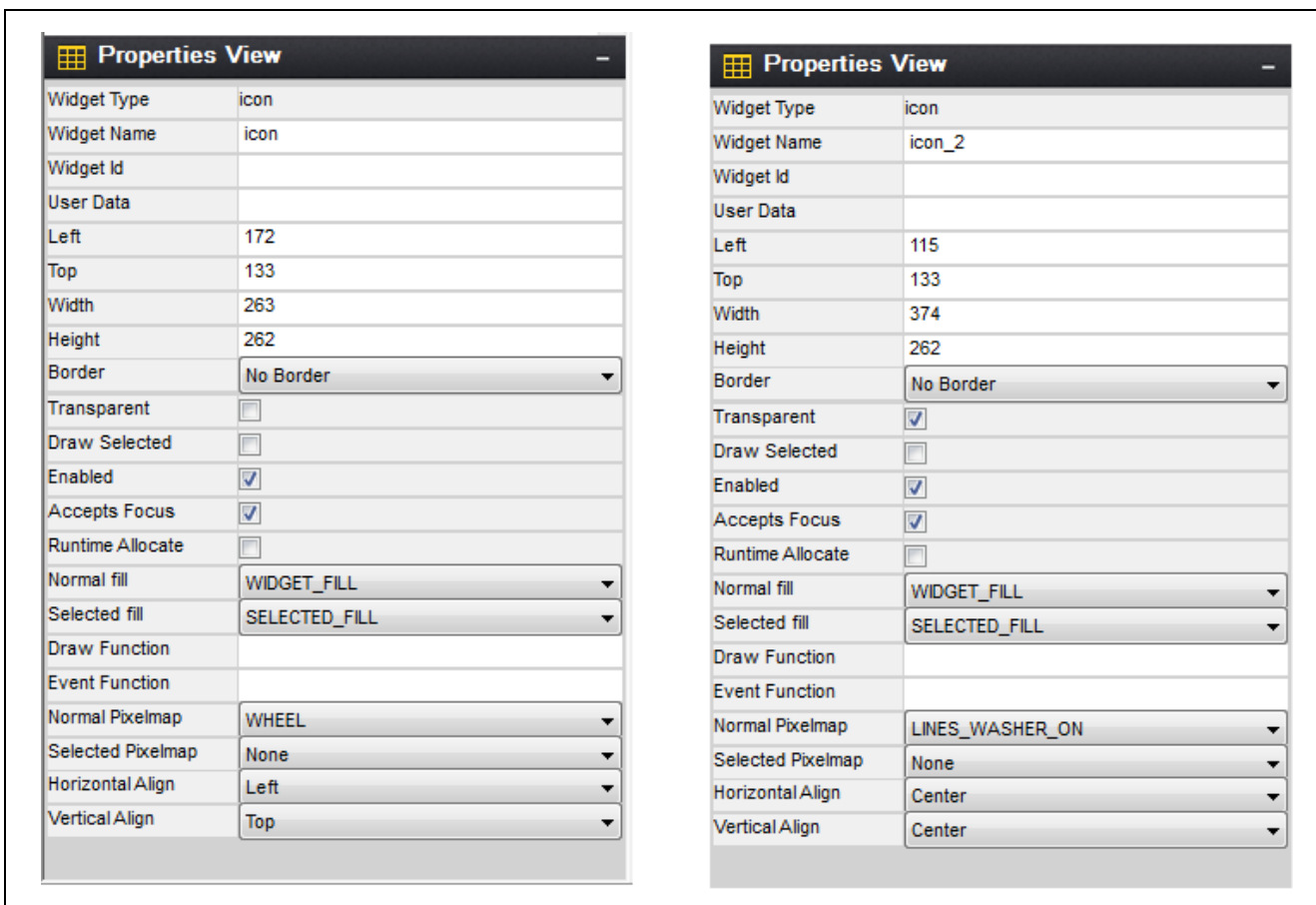


図 29 円形のスライダのプロパティビュー

3.4.3.7 メイン画面へのスライダバーの追加 (Adding Slider Bar to the main screen)

Slider Bar (スライダバー) は、洗濯の段階を示すためのステータスバーです。スライダバーをメイン画面に追加するには、[Pixmap_slider] ウィジェットと[Progress Bar Icon] を使用します。ステータスバーのステータス更新は、描画関数 `pixelmap_slider_alpha_draw` を使用して制御します。以下に、[Slider Bar] の個別コンポーネントを示します。

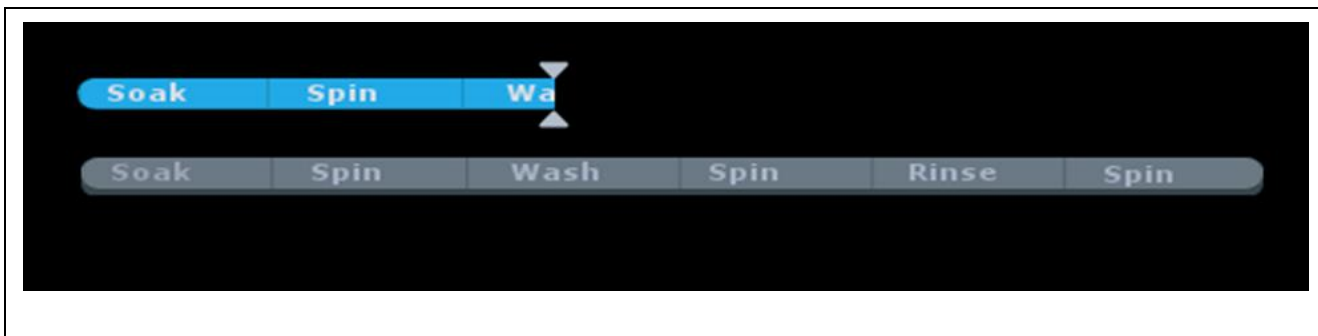


図 30 Slider Bar (スライダバー) コンポーネント

[Pixmap Slider] と [Status Bar Icon] それぞれのプロパティビューを次の図に示します。詳細については、GUIX Studio プロジェクトを参照してください。

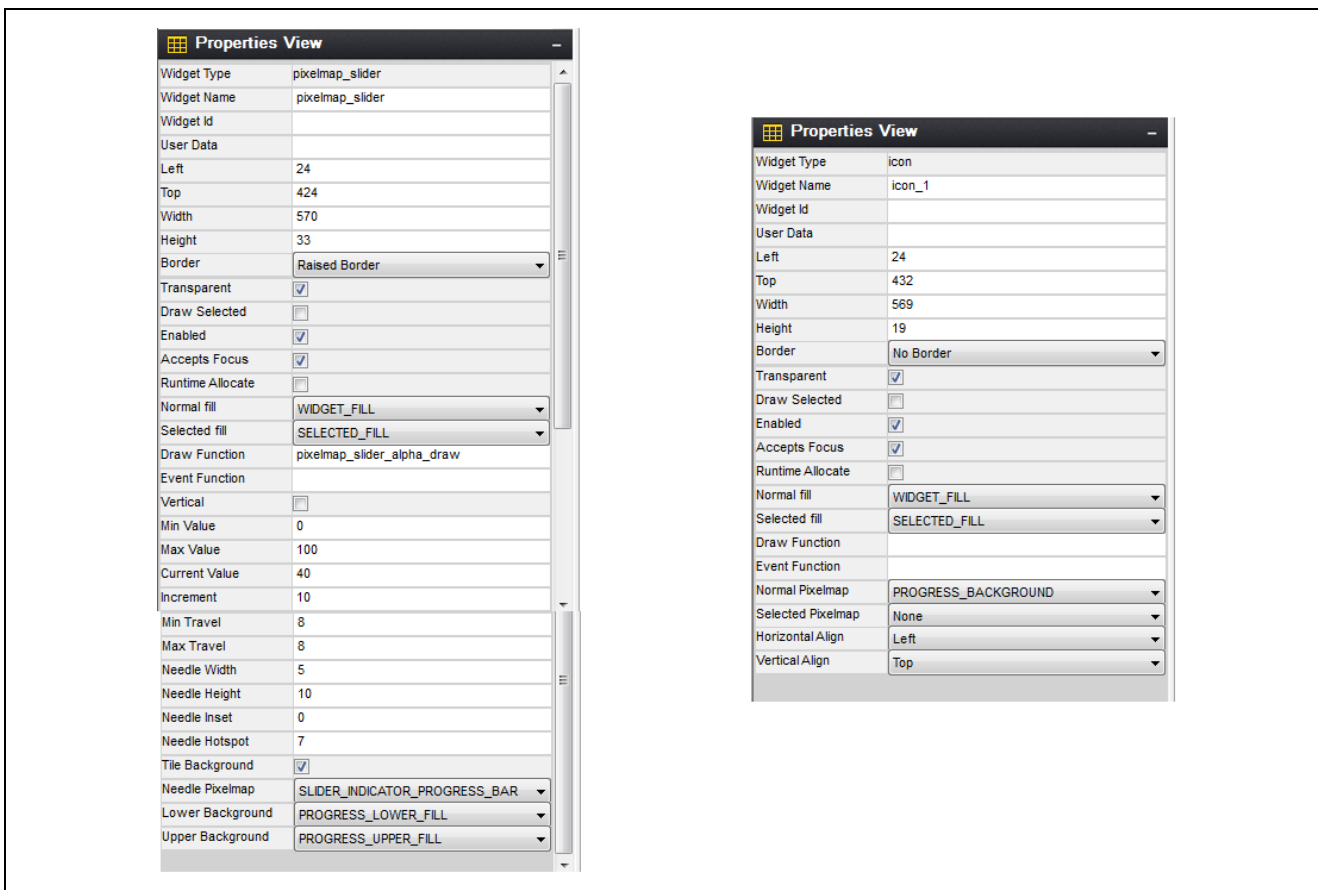


図 31 スライダバーのプロパティビュー

3.4.3.8 メイン画面への画面名の追加 (Adding the screen name to the main screen)

ページ名または画面名とは、描画関数 `prompt_alpha_draw` で作成されるウィジェットタイプのテキストプロンプトです。画面を選択したときにスクリーンを動的に更新します。メイン画面の場合、デフォルトのテキストとして「**Washer ON**」(洗濯機がオン)が表示されます。

画面名に対応するプロパティビューを次の図に示します。

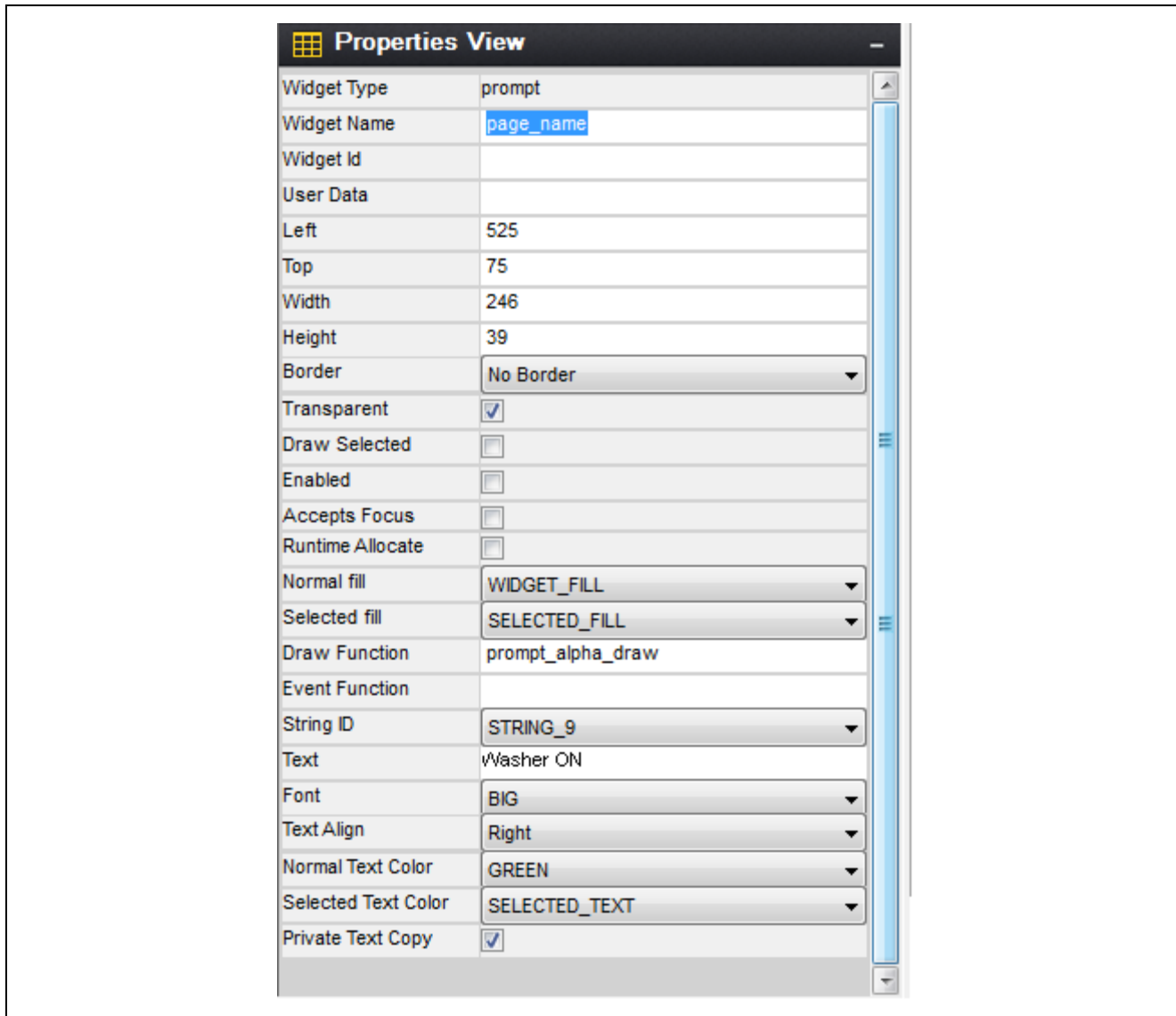


図 32 スライダーのプロパティビュー

これまでの章で説明した作成の手順とガイドラインでは、次の図のようなメイン画面を作成することを想定しています。詳細は、ここまでの説明と参考資料、および GUIX プロジェクトを参照してください。



図 33 ウィジェットを追加した後のメイン画面の外見

3.4.4 Garments (衣類) ウィンドウの作成 (Creating Garments Window)

[Garments] (衣類) ウィンドウはメイン画面の子ウィンドウで、[Garments] ボタンの選択/解除により、接続または接続解除されます。このウィンドウは、前の章で説明した [Washer] (洗濯機の機能) ウィンドウに似ています。このウィンドウには、状態を示すための靴下 (Sock) アイコンが内側にあり、コードを使用したアニメーションで制御されています。[Garments] (衣類) ウィンドウの作成では、付属の GUIX Studio プロジェクトの [Washer] (洗濯機の機能) ウィンドウが参考になります。

[Garments] (衣類) ウィンドウは、メイン画面の子ウィンドウであり、円形のホイール (円形のスライダ) と、プロンプトに対応する複数の線 (アイコン) で形成されています。これら 2 つのアイコンが重ね合わせで表示され、[Garments] (衣類) ウィンドウを形成しています。それぞれの線上では、[Garments] (衣類) ウィンドウの上に、[Washer] (洗濯機の機能) のラベルを表すテキストプロンプトが作成されます。次の図に、個別のアイコンを複数示します。これらは GUIX Studio でテキストプロンプトに重ね合わせ、最終的な [Garments] (衣類) ウィンドウを形成しています。



図 34 Radial Slider コンポーネント

[Garments] (衣類) ウィンドウを作成した後、次の図に示すスナップショットの画面になります。

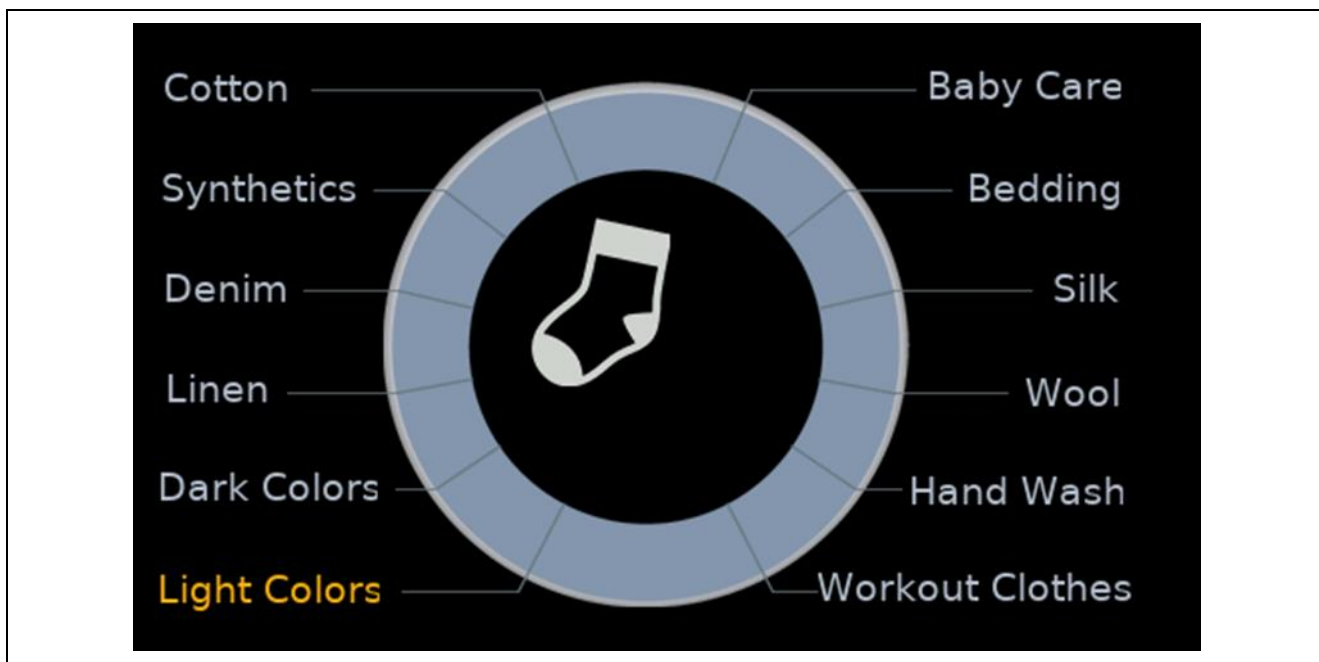


図 35 ウィジェットを追加した後の [Garments] (衣類) ウィンドウ

[Garments] (衣類) ウィンドウに対応するプロパティビューを次の図に示します。描画関数 `window_alpha_draw` とイベント関数 `garments_window_event_function` は、ウィンドウの描画と、[Garments] (衣類) ウィンドウ固有のイベント処理を行います。/src フォルダ内の `garments.c` を参照することにより、その機能をより理解することができます。

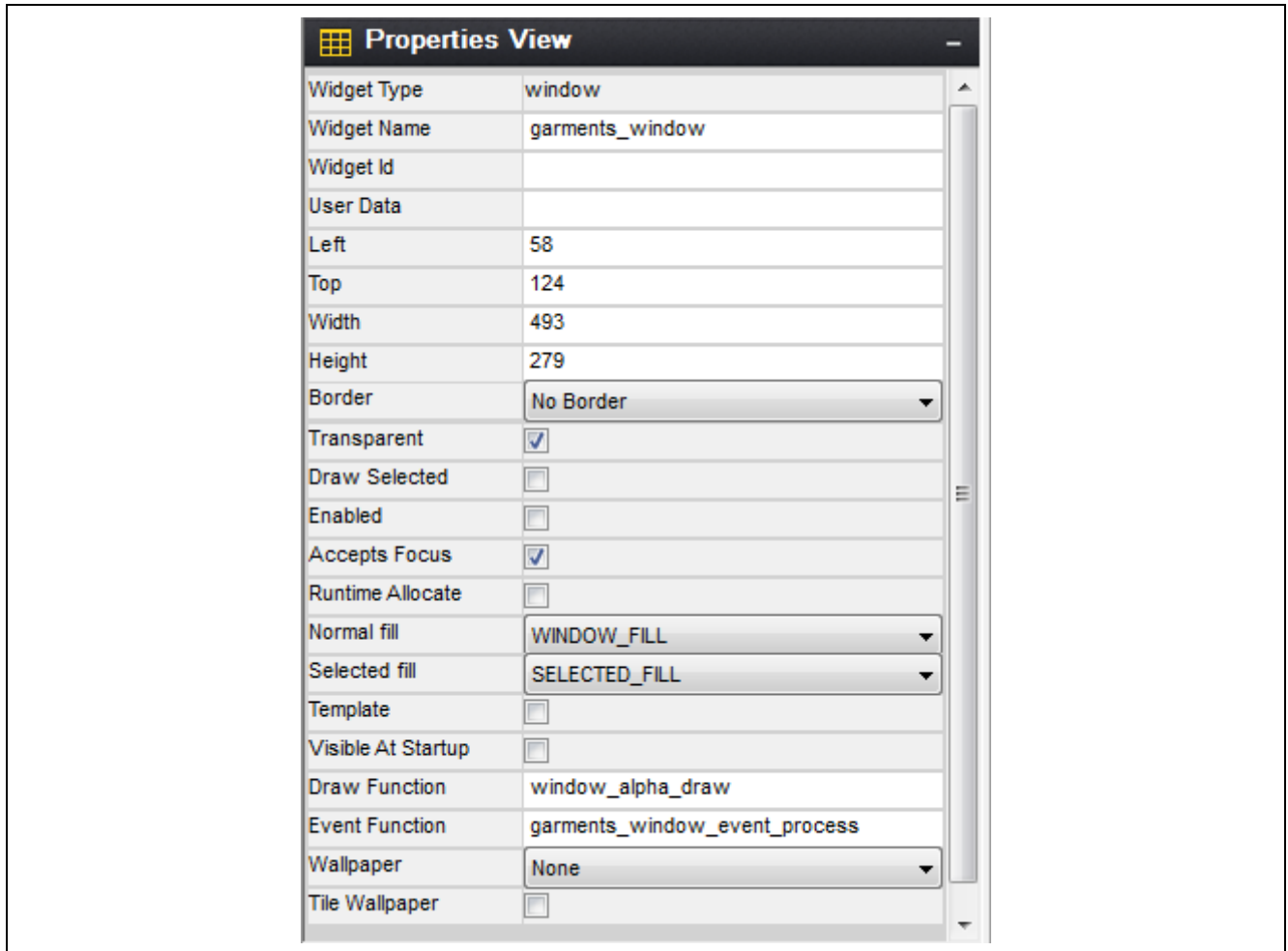


図 36 [Garments] (衣類) ウィンドウのプロパティビュー

3.4.5 [Temperature] (温度) ウィンドウの作成 (Creating Temperature Window)

[Temperature] (温度) ウィンドウは、円形のホイール (円形のスライダ radial slider) と、アーチ (青から赤に色が変わる円弧) を使用して、水の温度を表します。[Temperature] (温度) ウィンドウには、低温から高温までの水の温度を表すテキストウィジェットと、実際の水の温度を表す状態ウィジェットが内部にあります。次の図のように、[Temperature] (温度) ウィンドウを構成するアイコンが複数表示されます。

このウィンドウは子ウィンドウであり、[Temperature] (温度) ボタンが選択または解除を選択することにより、接続または接続解除が行われます。このウィンドウには、コードにより制御されるアニメーションのようなスライダペン (円形の形状) があります。このウィンドウは、2 個のアイコンを重ね合わせ、複数のテキストウィジェットを追加する形で作成します。

[Temperature] (温度) ウィンドウに対応するプロパティビューを次の図に示します。詳細については、付属している GUIX Studio プロジェクトを参照してください。描画関数 `temperature_window_draw` とイベント関数 `temperature_window_event_process` を理解するためのソースコードが含まれます。



図 37 [Temperature Window] (温度ウィンドウ) のコンポーネント

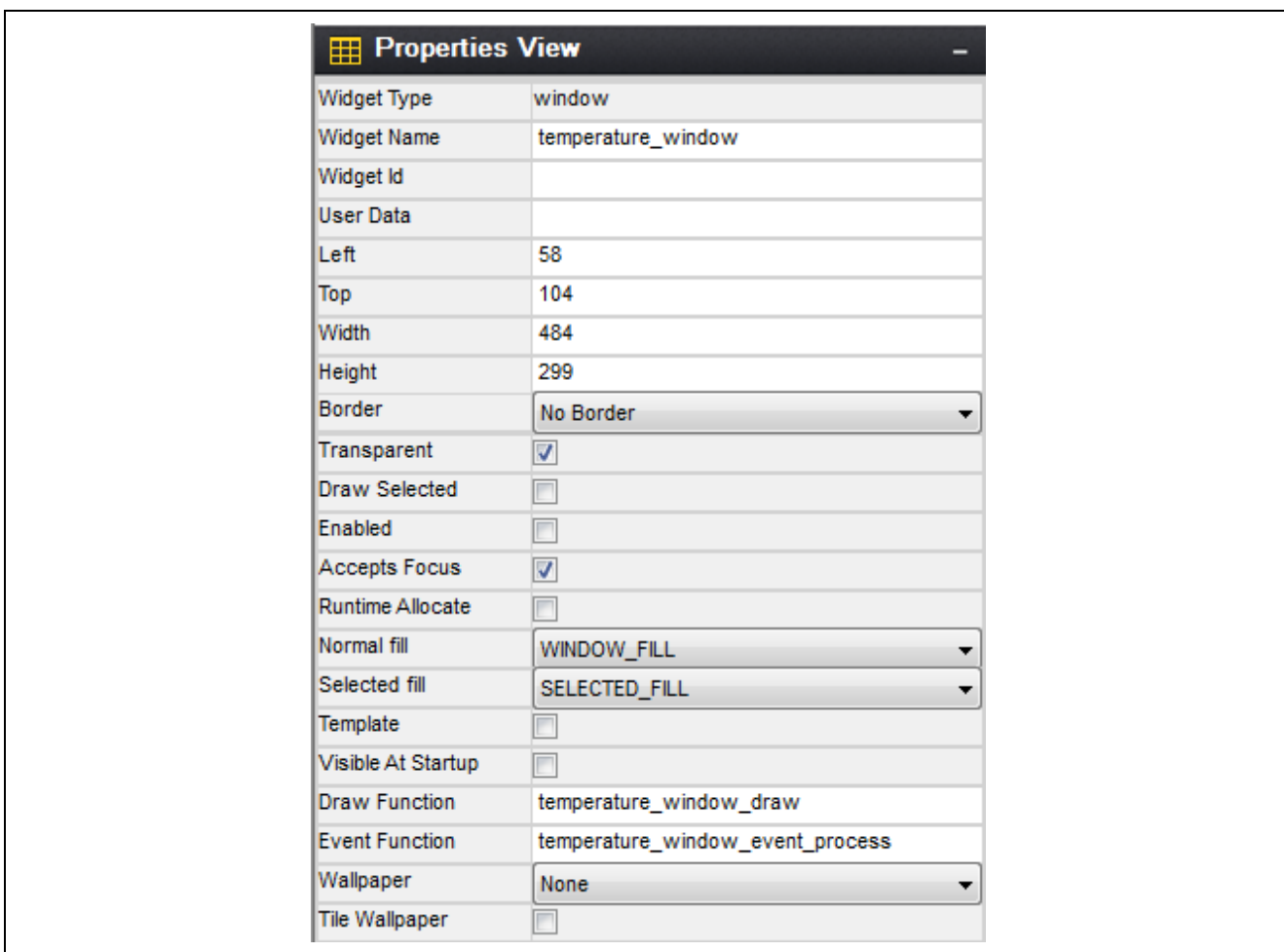


図 38 [Temperature] (温度) ウィンドウのプロパティビュー

[Temperature] (温度) ウィンドウは次の図に示すスナップショットのようになります。

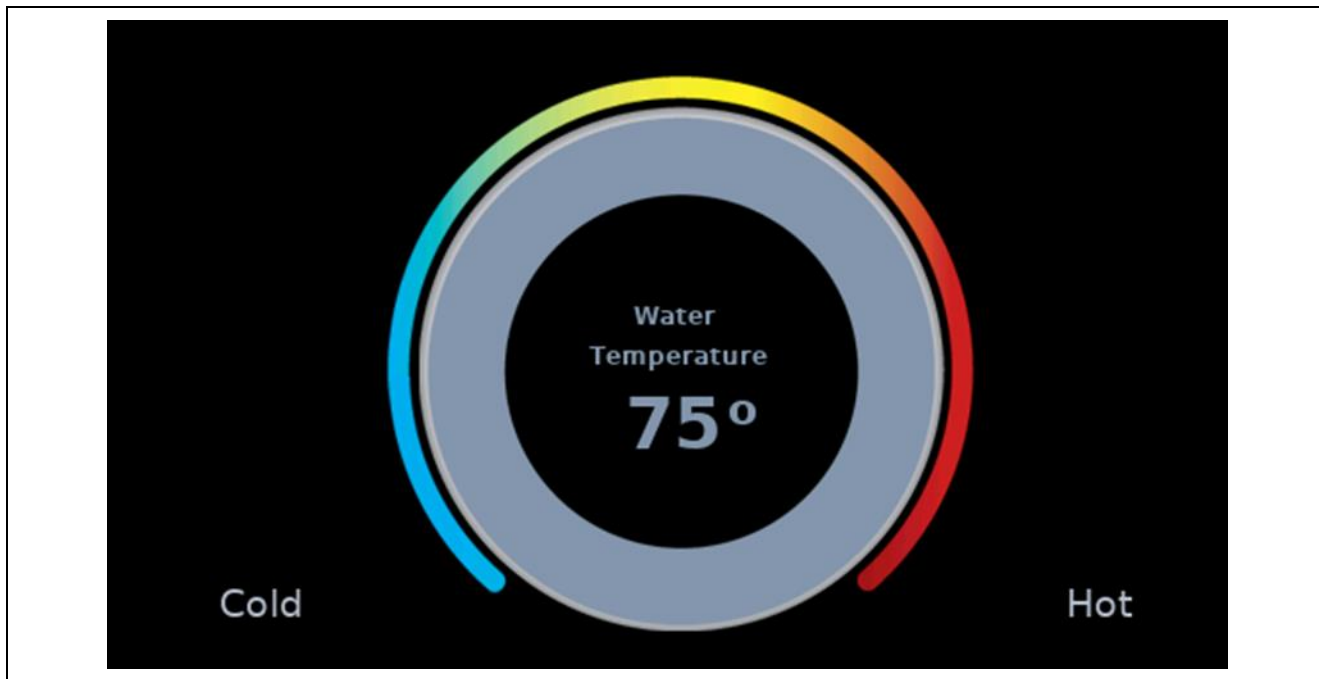


図 39 ウィジェットを追加した後の [Temperature] (温度) のウィンドウコンポーネント

3.5 [Water Level] (水位) ウィンドウの作成 (Creating Water Level Window)

[Water Level] (水位) ウィンドウは子ウィンドウであり、[Water Level Selection] (水位選択) ボタンの状態に基づいて、接続または解除することができます。[Washer] (洗濯機の状態) または [Temperature] (温度) ウィンドウと比較すると分かるように、このウィンドウの外見は異なっています。このウィンドウは、アイコンウィジェットとピクセルマップスライダ (Pixelmap Slider) を重ね合わせてアイコンを完成させる方法で実装します。また、このウィンドウには水位を示すためのテキストウィジェットもあります。

次の図は、個別のアイコンとピクセルスライダを示しています。

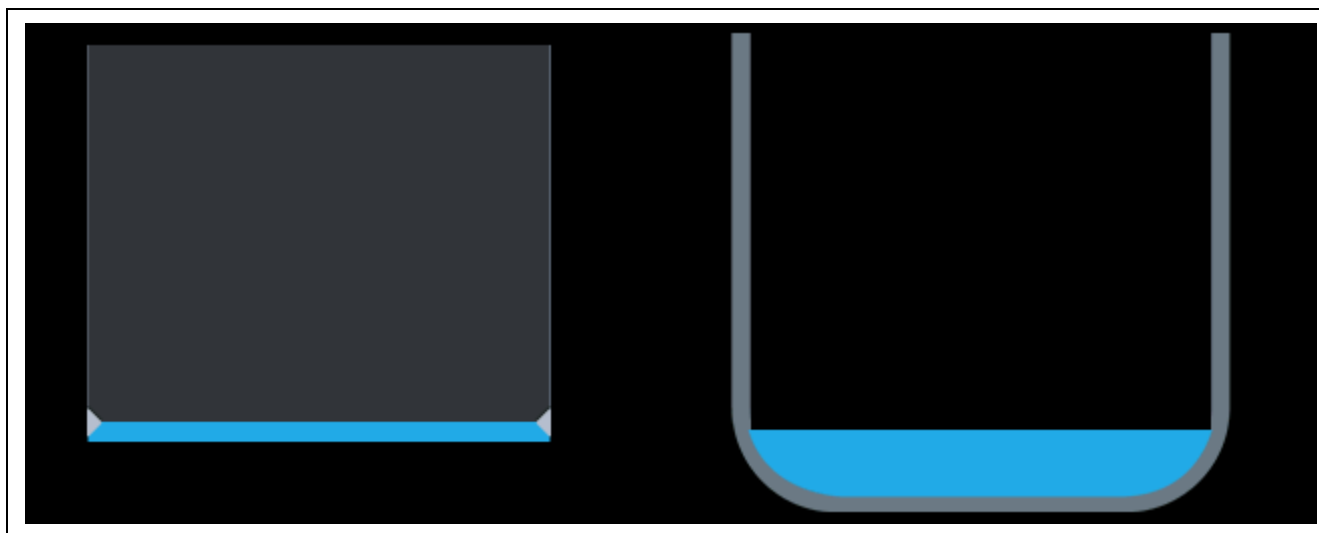


図 40 [Water Level] (水位) ウィンドウのコンポーネント

[Water Level] (水位) ウィンドウには、イベントを処理するイベント関数 `water_level_window_event_process` と、スライダの移動に応じてさまざまな水位を描画するイベント関数 `water_level_window_draw` があります。

この [Water Level] (水位) ウィンドウを作成する際には、個別のウィジェットに関する説明が記された付属している GUIX Studio プロジェクトで参照することもできます。[Water Level] (水位) ウィンドウに対応するプロパティビューを次の図に示します。

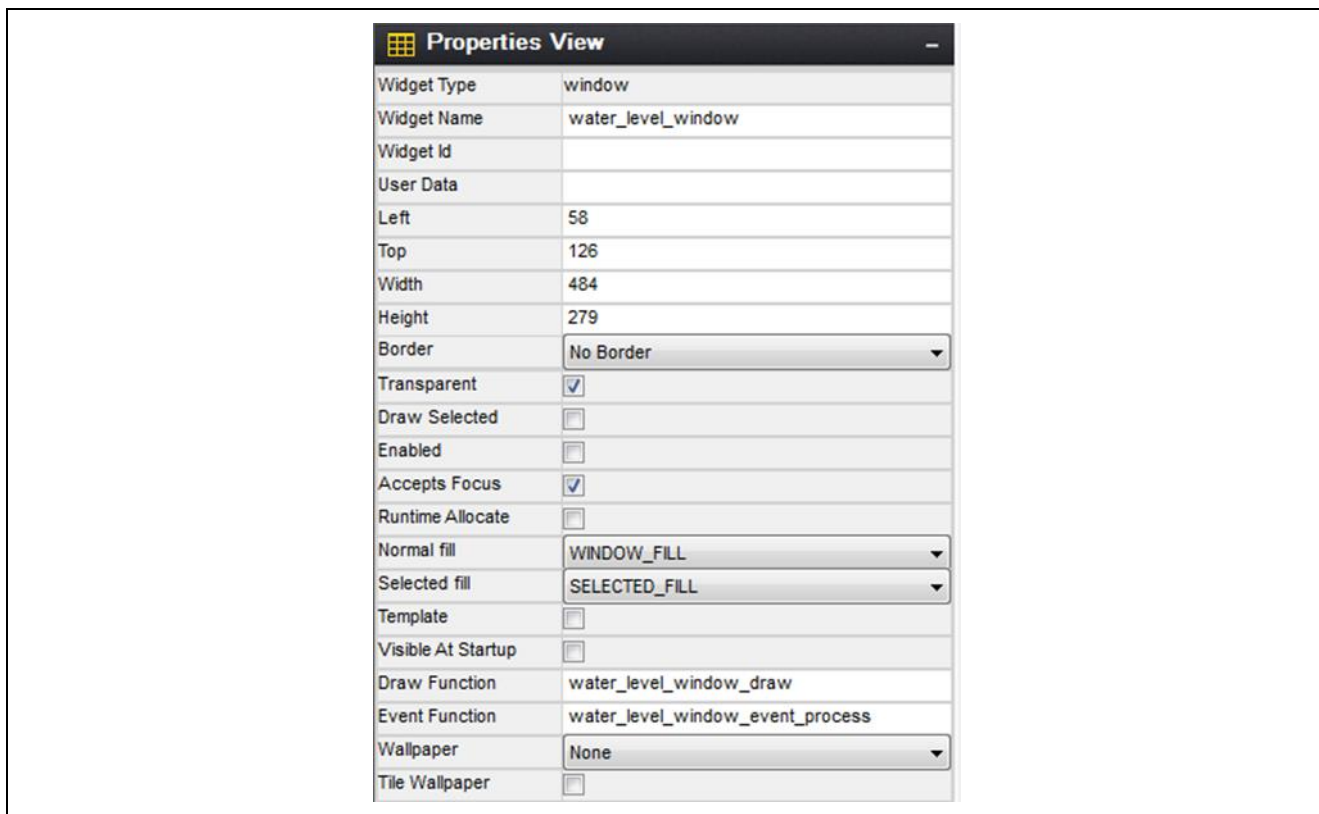


図 41 [Water Level] (水位) ウィンドウのプロパティビュー

[Water Level] (水位) ウィンドウのすべてのウィジェットを作成した後の最終的な画面を次の図に示します。

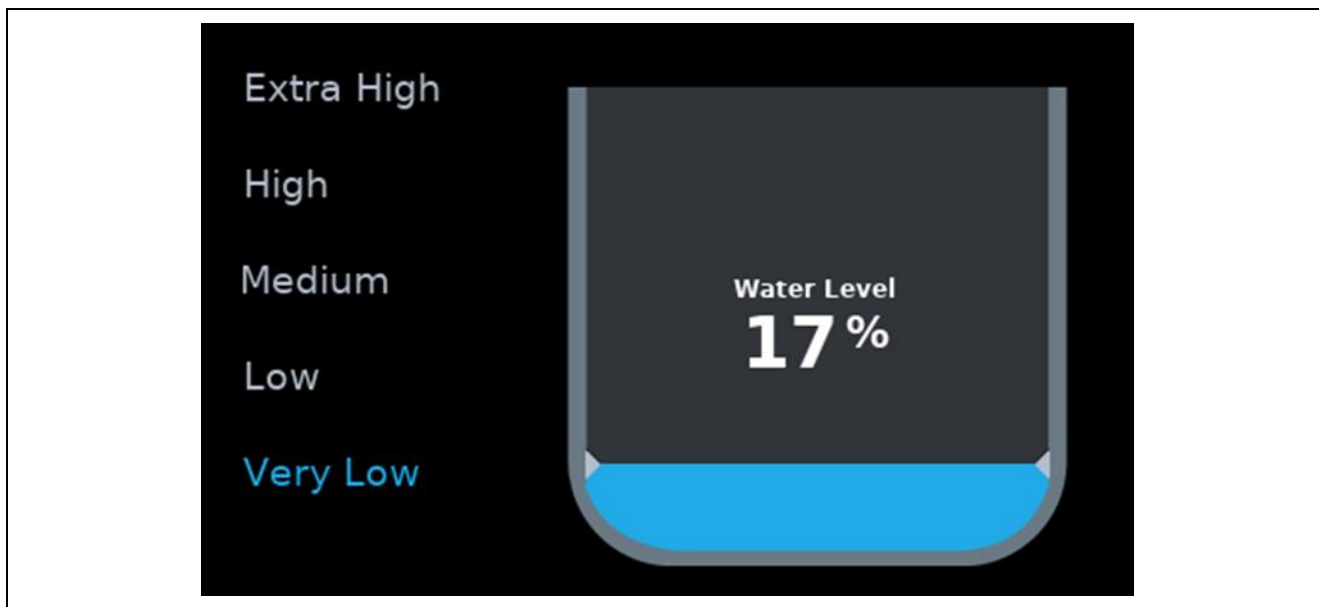


図 42 ウィジェットを追加した後の [Water Level] (水位) ウィンドウのコンポーネント

3.5.1 作成したプロジェクトから生成するリソースと仕様 (Resource and Specification files from the created project)

ここまで、プロジェクトの要求に基づき、GUIX Studio を使用してすべての画面を作成しました。GUIX Studio には、リソースファイルと仕様ファイルを生成するためのいくつかのオプションがあります。これらのリソースファイルは、新しく追加した画面、グラフィックス画像とそのデータ構造体 (data structure)、および関連する描画関数やイベントハンドラ関数があり、ANSI C 形式のコードで生成されます。これらのリソースファイルと仕様ファイルは、アプリケーションプロジェクトに入力します。リソースファイルと仕様ファイルは、次の表に示すように、GUIX Studio を使用して生成します。

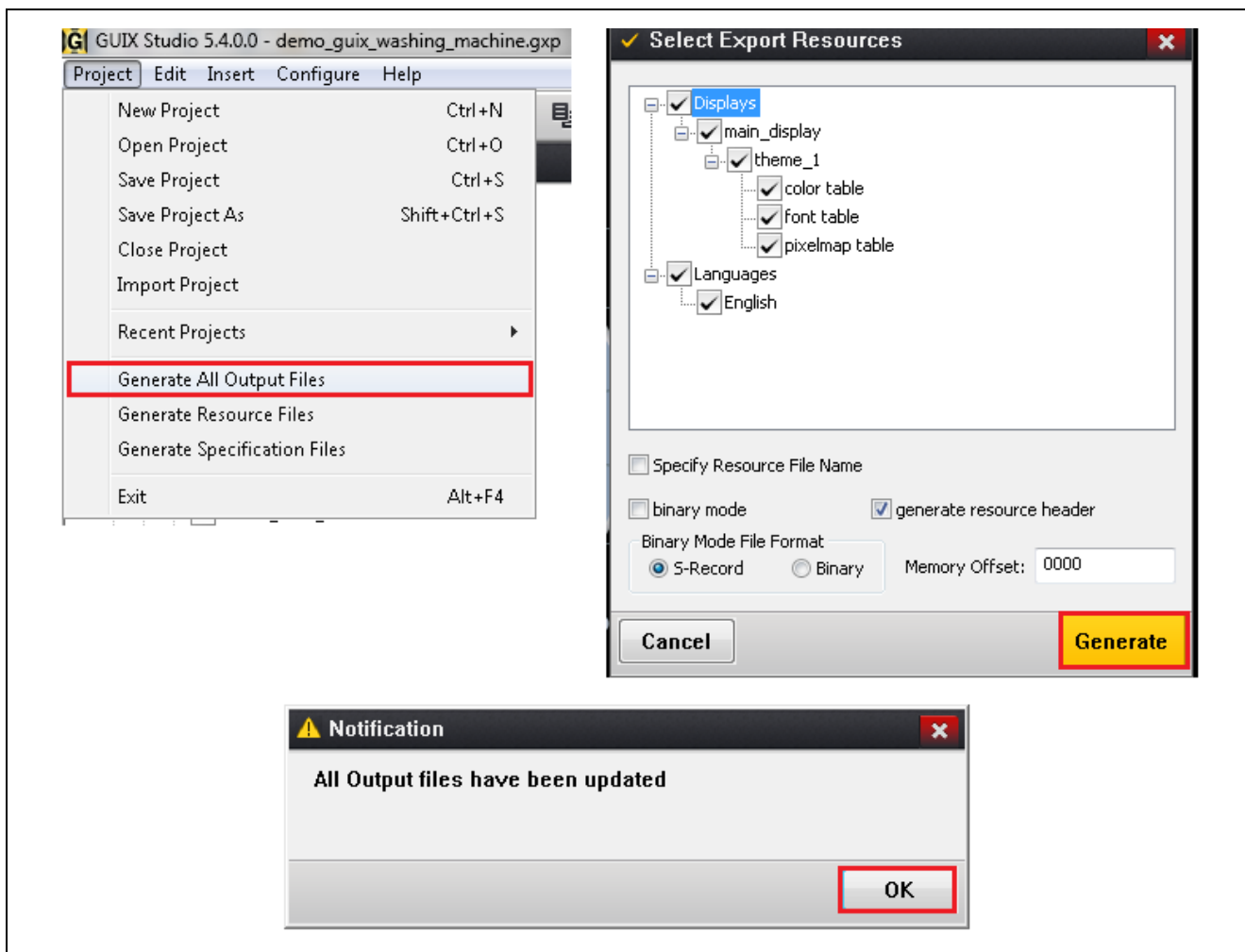


図 43 リソースファイルの生成

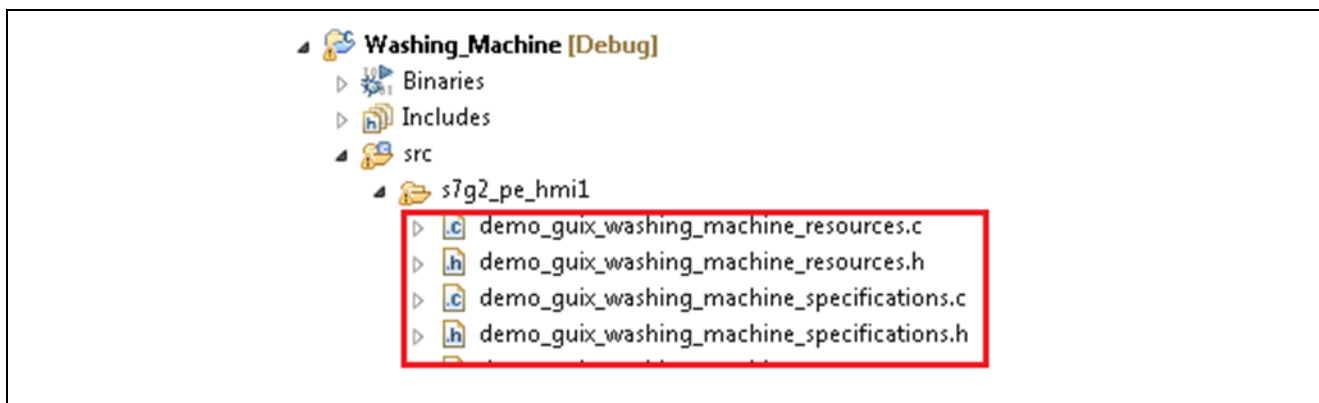


図 44 [Project View] (プロジェクトビュー) 内のリソースファイル

生成されたこれらのコードは、アプリケーションコードとともにコンパイルし、ターゲットの最終イメージを生成します。洗濯機アプリケーションプロジェクトで使用するリソースファイルと仕様ファイルに関するサンプルのスナップショットを、上の図に示しています。

3.6 アプリケーションでの GUIX モジュールの包含と設定 (Including and configuring the GUIX module in an application)

- [Threads] から [New Thread] をクリックして、新しいスレッドを作成します。

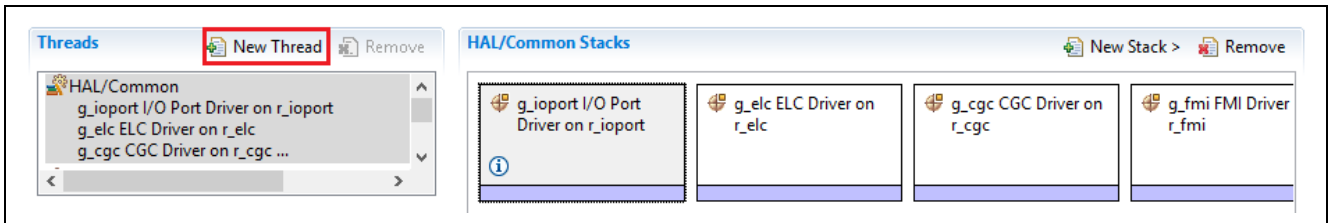


図 45 新しいスレッドの作成

- [New Thread] をクリックして、プロパティを表示します。
- [Properties] を以下の内容と一致するように編集します。

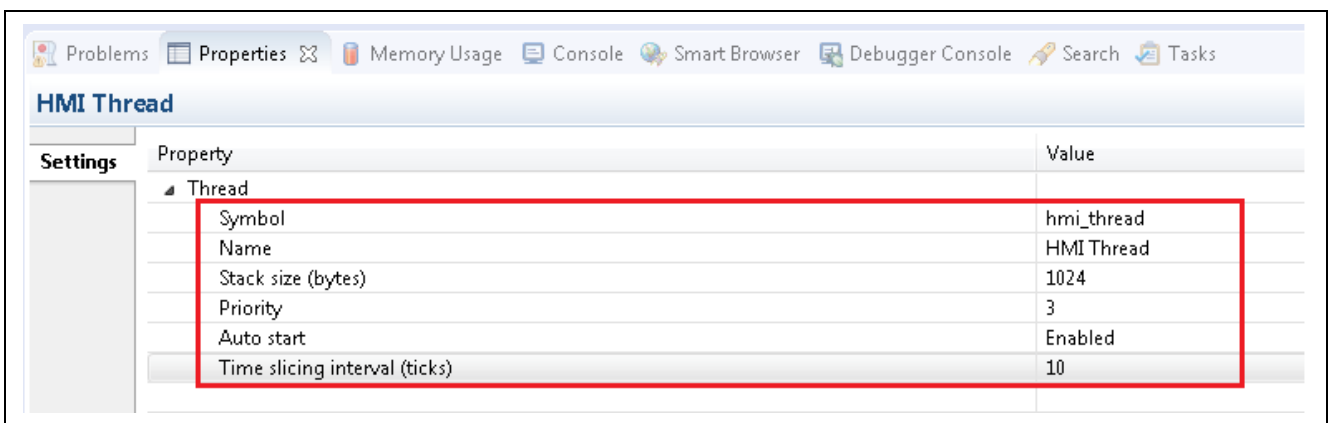


図 46 HMI スレッドのプロパティの設定

- [Synergy Configuration Window] > [Threads] タブ > [Main Thread Stacks] から、[New Stack] をクリックします。

注記：新しいモジュールを追加する前に、[HMI Thread] が選択されていることを確認してください。

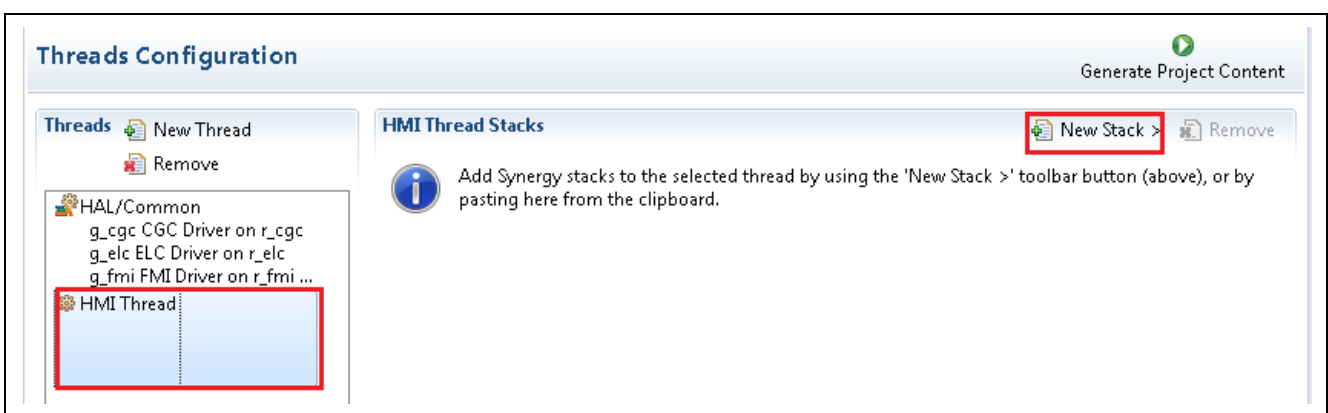


図 47 HMI スレッドのスタック

- [Synergy Configuration Window]>[Threads]>[HMI Thread Stacks] から、[New Stack] を選択し、タッチパネル用のフレームワーク `sf_touch_panel_i2c` を追加します。

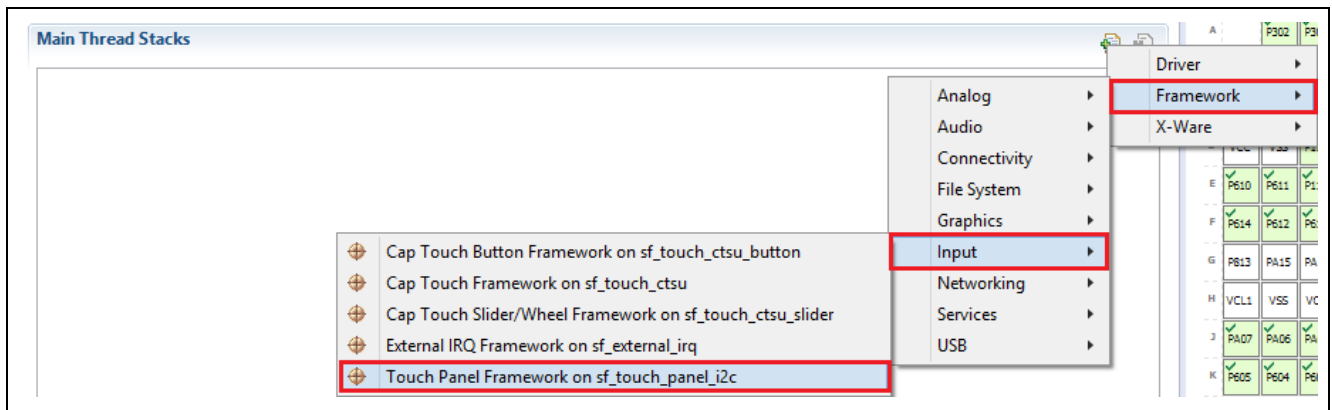


図 48 タッチパネルフレームワークの追加

- 以下のプロパティを設定します。

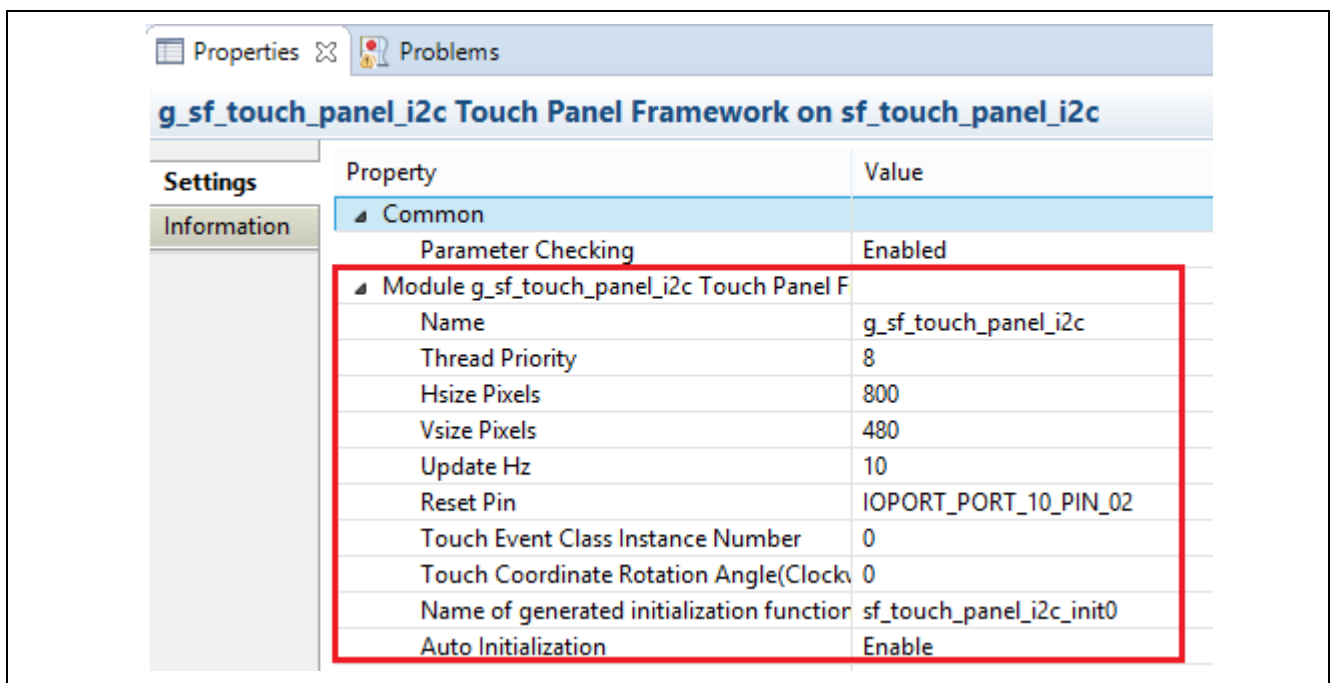


図 49 タッチパネルのプロパティの設定

この時点で、Synergy Configurator はメッセージフレームワーク (message framework) と外部 IRQ フレームワーク (external IRQ framework) を作成しています。以下の図に示す外部 IRQ と I2C ドライバスタック (I2C driver stack) のプレースホルダを持っていることに注意してください。

メッセージングフレームワークは、他のフレームワークレイヤやタスクがシステム内でメッセージを渡すために使用します。このシステムは、タッチ入力が発生したときに、タッチスクリーンドライバから HMI スレッドにデータを渡すために使用します。

SF 外部割り込み (SF External Interrupt) は、次の図に示すようにタッチコントローラドライバで使用されるフレームワークレイヤです。

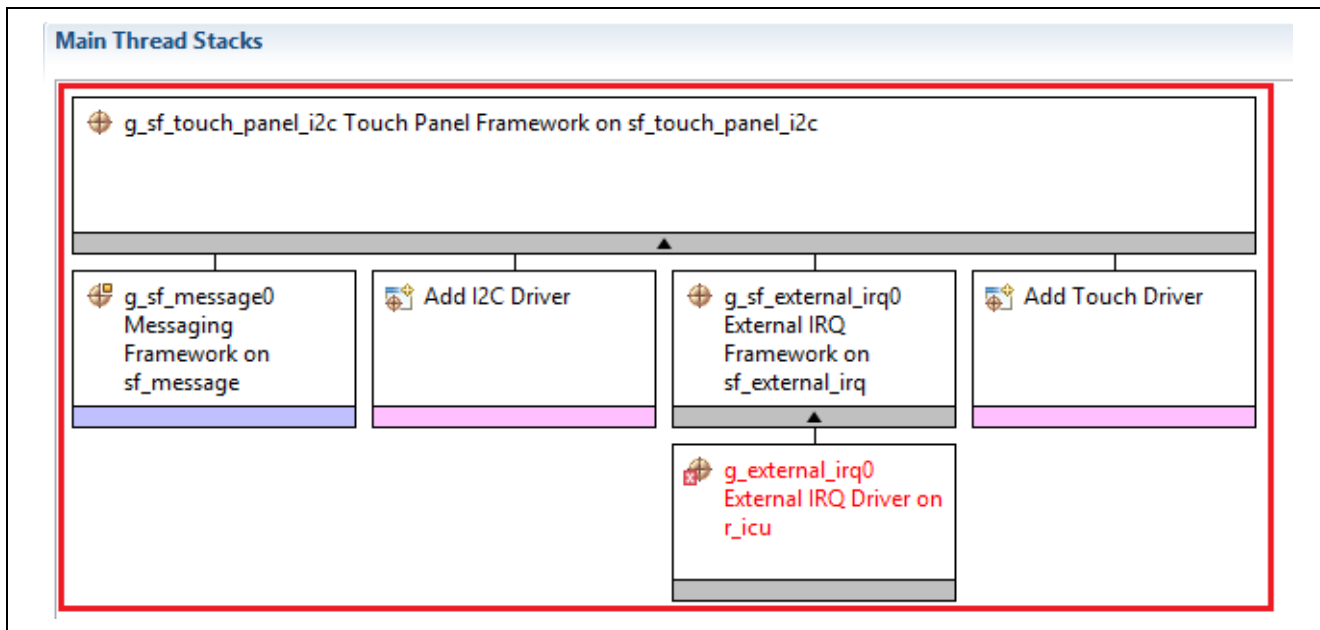


図 50 タッチパネルフレームワークのスタック

- [External IRQ Framework on sf_external_irq] を選択し、以下のプロパティを設定します。

Property	Value
Common	
Parameter Checking	Enabled
Module g_sf_touch_irq External IRQ Framework on sf_external_irq	
Name	g_sf_touch_irq
Event	Semaphore Put

図 51 外部割り込みのプロパティの設定

- [External IRQ Driver on r_icu] を選択し、新しいモジュールに以下のプロパティを設定します。ヒント: 最初にチャンネルを変更してください。

Property	Value
Common	
Parameter Checking	Default (BSP)
Module g_touch_irq External IRQ Driver on r_icu	
Name	g_touch_irq
Channel	12
Trigger	Falling
Digital Filtering	Enabled
Digital Filtering Sample Clock (Only valid when Digital Filtering is E	PCLK / 1
Interrupt enabled after initialization	True
Callback	NULL
Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)

図 52 タッチスクリーンの IRQ プロパティ

- [Synergy Configuration Window]>[Threads] > [Main Thread Stacks] からで、[Add I2C Driver] を右クリックした後、[New] -> [I2C Master Driver on r_riic] を選択し、I2C バスのドライバを追加します。

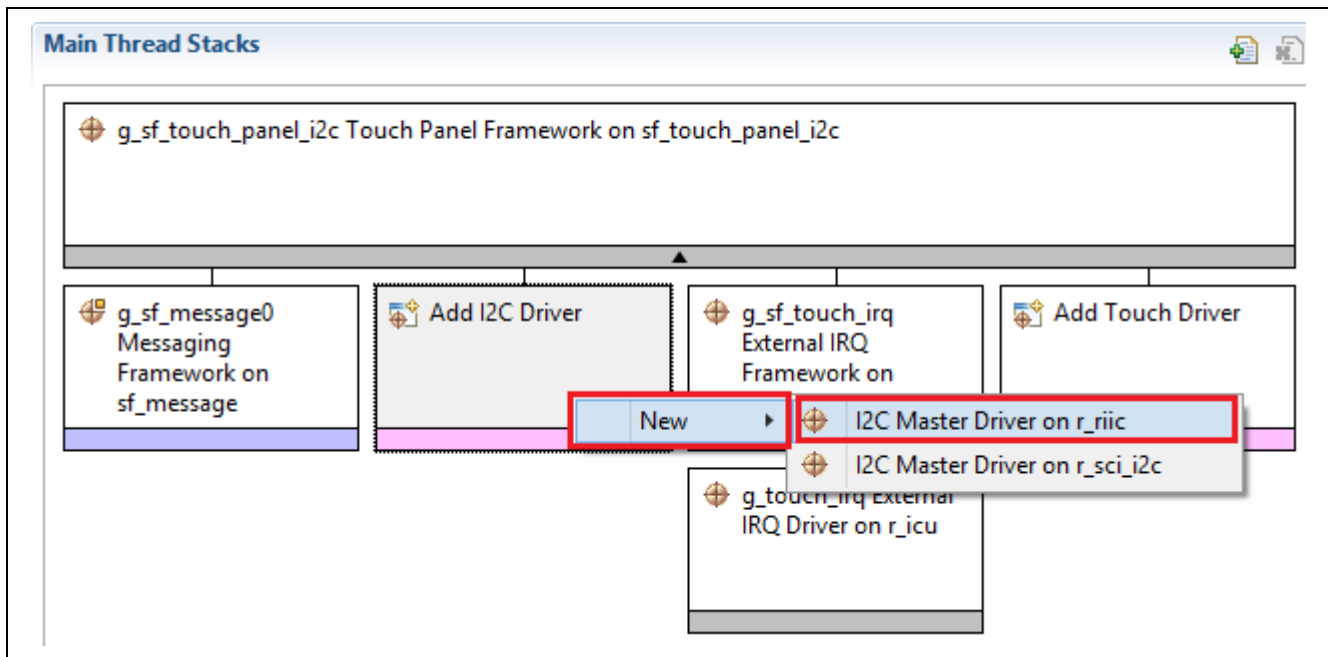


図 53 I2C ドライバの追加

- [I2C Master Driver on riic] で以下のプロパティを設定します。ヒント: 最初にオプションを変更してください。

Property	Value
▼ Common	
Parameter Checking	Default (BSP)
▼ Module g_i2c I2C Master Driver on r_riic	
Name	g_i2c
Channel	1
Rate	Fast-mode
Slave Address	0x38
Address Mode	7-Bit
Callback	NULL
Receive Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)
Transmit Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)
Transmit End Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)
Error Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)

図 54 I2C ドライバの設定

- Touch Driver の以下のプロパティを設定します。

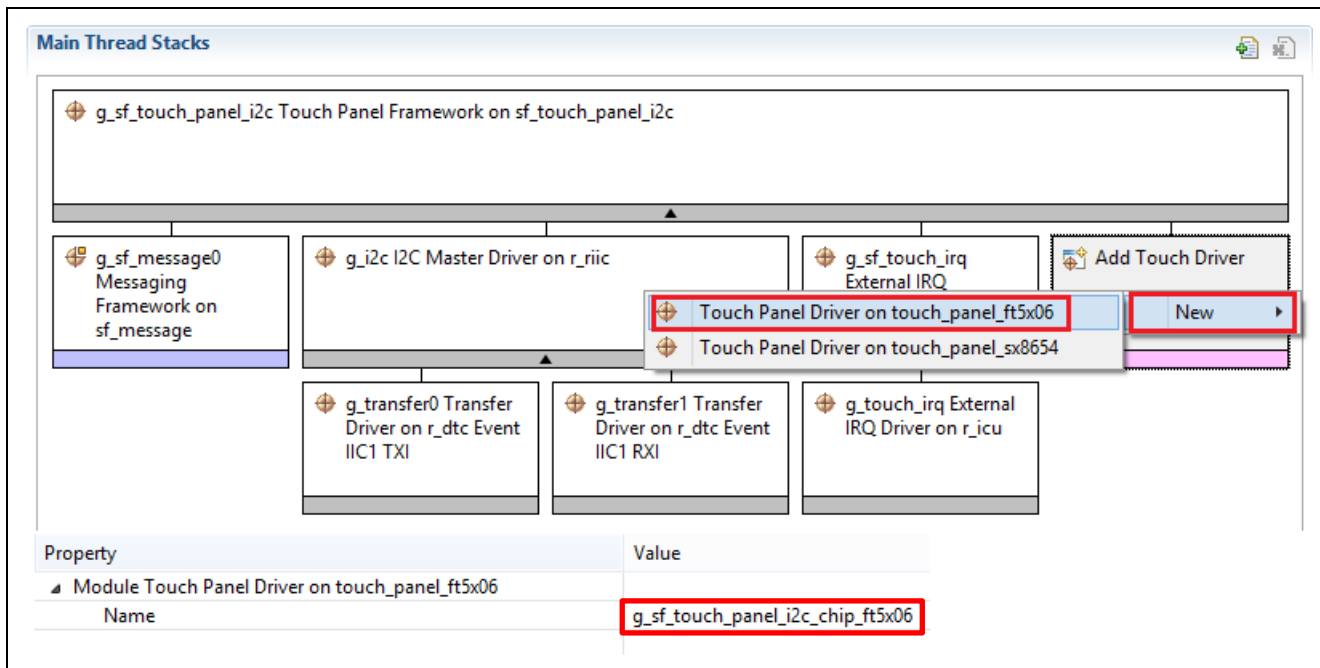


図 55 タッチドライバの設定

- [HMI Thread Stacks] で、[New Stack] > [X-Ware] > [GUIX] > [GUIX on gx] を選択します。

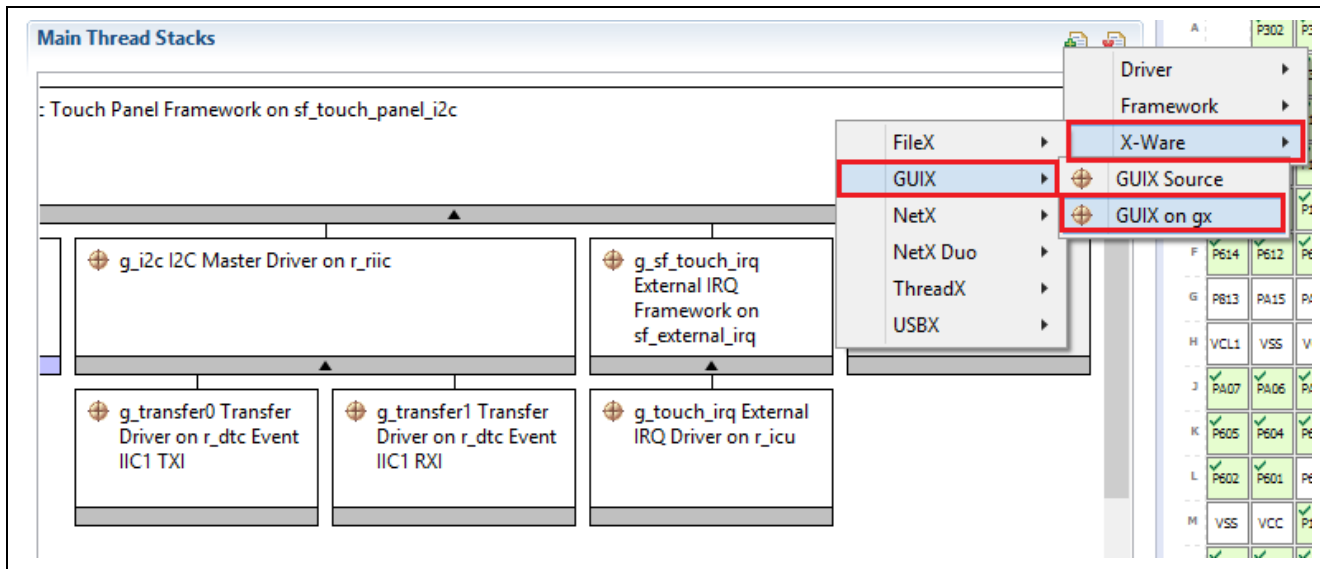


図 56 GUIX on gx

Synergy Configurator は既に GUIX Port on sf_el_gx フレームワーク と Display Driver を作成して、JPEG デコードと D/AVE ハードウェアアクセラレータに対応するプレースホルダが存在する状態にあります。

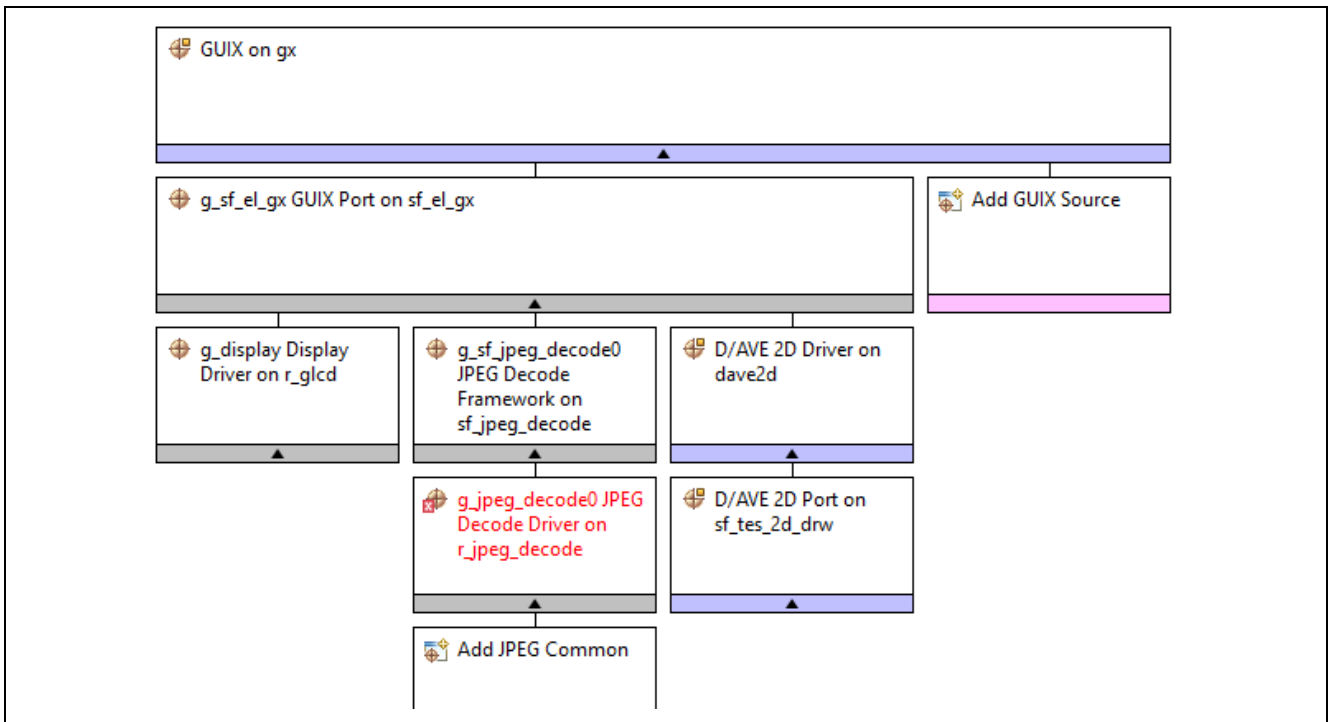


図 57 GUIX on gx

- [GUIX on gx] を選択し、以下の [Properties] を設定します。

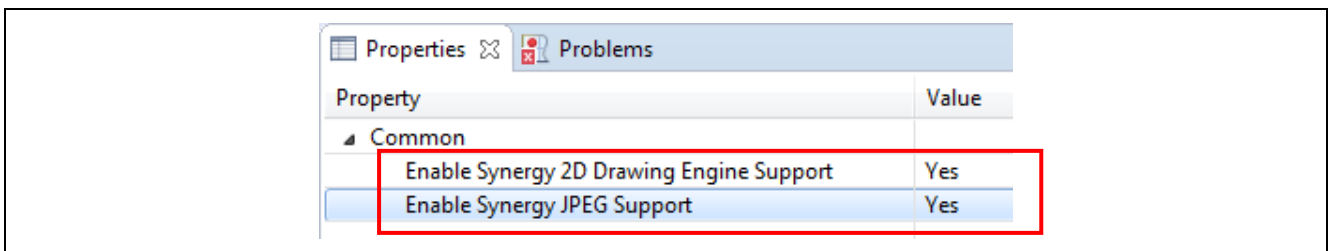


図 58 gx プロパティに対応する GUIX

- [JPEG Common] を [Decode Driver on r_jpeg_decode] に追加します。

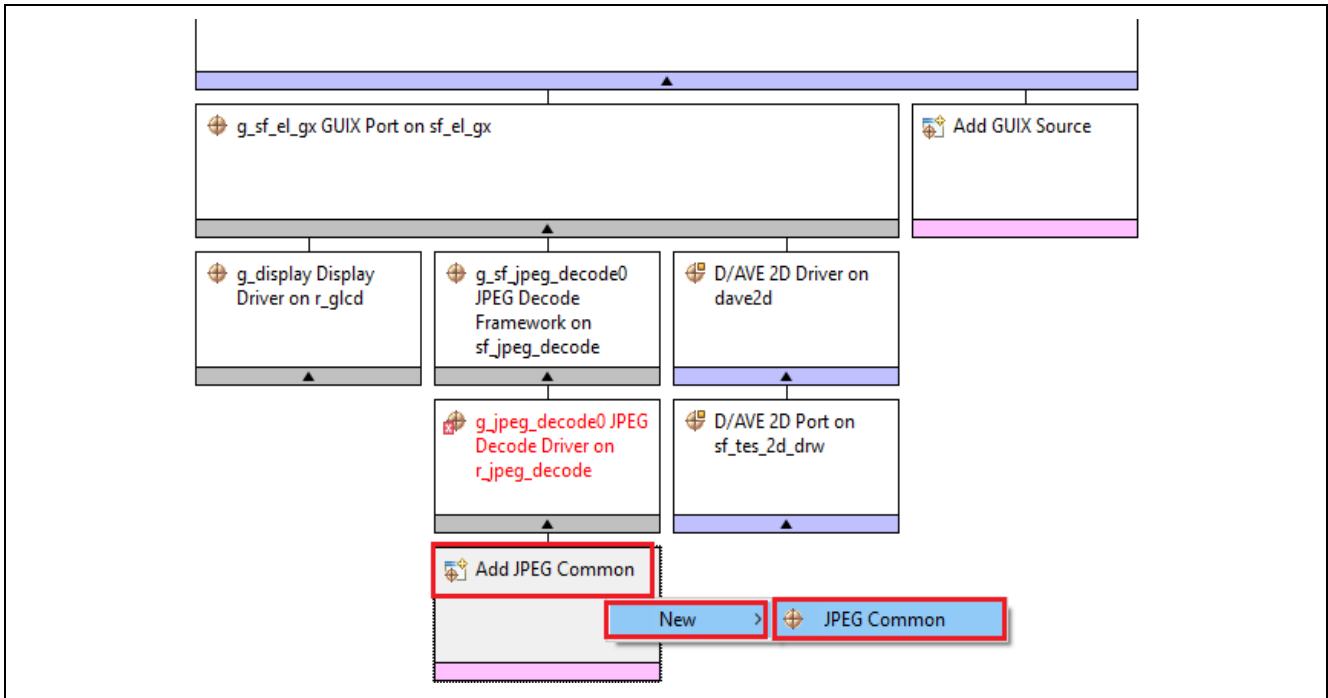


図 59 JPEG 共通モジュール

- [GUIX Port on sf_el_gx] を選択し、以下の [Properties] を設定します。

Property	Value
Common	
Parameter Checking	Enabled
Module g_sf_el_gx GUIX Port on sf_el_gx	
Name	g_sf_el_gx
Display Driver Configuration Inheritance	Inherit Graphics Screen 1
Name of User Callback function	NULL
Screen Rotation Angle(Clockwise)	0
GUIX Canvas Buffer (required if rotation angle is not zero)	Not used
Size of JPEG Work Buffer (valid if JPEG hardware acceleration is enabled)	1000
Memory section for GUIX Canvas Buffer	sdram
Memory section for JPEG Work Buffer	sdram

図 60 sf_el_gx プロパティに対応する GUIX ポート

- [JPEG Decode Driver on r_jpeg] を選択し、以下の割り込みプロパティを設定します。優先度 3 は任意であることに注意してください。

Property	Value
▼ Common	
Parameter Checking	Default (BSP)
▼ Module g_ipeg_decode0 JPEG Decode Driver on r_ipeg	
Name	g_jpeg_decode0
Byte Order for Input Data Format	Normal byte order (1)(2)(3)(4)(5)(6)(7)(8)
Byte Order for Output Data Format	Normal byte order (1)(2)(3)(4)(5)(6)(7)(8)
Output Data Color Format	Pixel Data RGB565 format
Alpha value to be applied to decoded pixel data(only valid for ARG	255
Name of user callback function	NULL
Decompression Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)
Data Transfer Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)

図 61 r_jpeg プロパティに対応する JPEG デコードドライバ

- [Main Thread Stacks] から [D/AVE 2D Port on sf_tes_2d_drw] を選択し、以下のプロパティを設定します。

Property	Value
▼ Common	
Work memory size for display lists in bytes	32768
DRW Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)

図 62 D/AVE 2D ポートのプロパティ

- [Main Thread Stacks] から [Display Driver on r_glcd] を選択し、以下の [Interrupt Properties] を設定します。

MISC - Correction Process Order	Brightness and Contrast then Gamma
Line Detect Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)
Underflow 1 Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)
Underflow 2 Interrupt Priority	Disabled

図 63 割り込みプロパティ

- 下にスクロールし、以下の [Graphics Screen 1 Properties] を表示します。

Module	
Name	g_display
Name of display callback function to be defined by user	NULL
Input - Panel clock source select	Internal clock(GLCDCLK)
Input - Graphics screen1	Used
Input - Graphics screen1 frame buffer name	fb_background
Input - Number of Graphics screen1 frame buffer	2
Input - Section where Graphics screen1 frame buffer allocated	sdram
Input - Graphics screen1 input horizontal size	800
Input - Graphics screen1 input vertical size	480
Input - Graphics screen1 input horizontal stride(not bytes but pixels)	800
Input - Graphics screen1 input format	16bits RGB565
Input - Graphics screen1 input line descending	Not used
Input - Graphics screen1 input lines repeat	Off
Input - Graphics screen1 input lines repeat times	0
Input - Graphics screen1 layer coordinate X	0
Input - Graphics screen1 layer coordinate Y	0
Input - Graphics screen1 layer background color alpha	255
Input - Graphics screen1 layer background color Red	255
Input - Graphics screen1 layer background color Green	255
Input - Graphics screen1 layer background color Blue	255
Input - Graphics screen1 layer fading control	None
Input - Graphics screen1 layer fade speed	0

図 64 グラフィックスクリーン 1 のプロパティ

以下の [Output properties] を設定します。

Output - Horizontal total cycles	1024
Output - Horizontal active video cycles	800
Output - Horizontal back porch cycles	46
Output - Horizontal sync signal cycles	20
Output - Horizontal sync signal polarity	Low active
Output - Vertical total lines	525
Output - Vertical active video lines	480
Output - Vertical back porch lines	23
Output - Vertical sync signal lines	10
Output - Vertical sync signal polarity	Low active
Output - Format	24bits RGB888
Output - Endian	Little endian
Output - Color order	RGB
Output - Data Enable Signal Polarity	High active
Output - Sync edge	Rising edge
Output - Background color alpha channel	255
Output - Background color R channel	0
Output - Background color G channel	0
Output - Background color B channel	0

図 65 出力スクリーン 2 のプロパティ

- 次の図に合わせて、以下の [TCON settings] を変更します。

TCON - Hsync pin select	LCD_TCON0
TCON - Vsync pin select	LCD_TCON1
TCON - DataEnable pin select	LCD_TCON2
TCON - Panel clock division ratio	1/8

図 66 TCON の設定

- [Synergy Configuration] ウィンドウで [Messaging] タブを選択します。以下のウィンドウが表示されます。

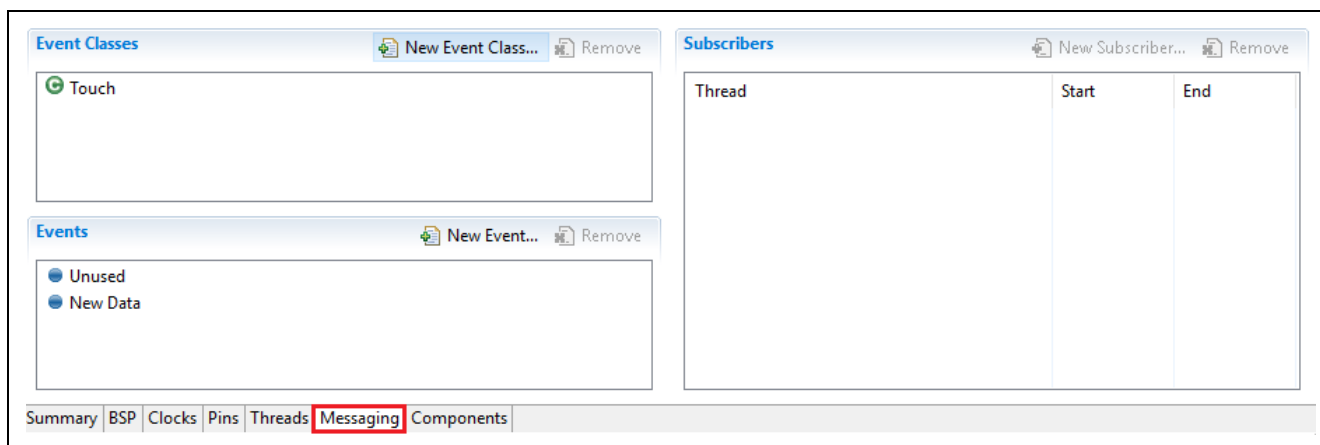


図 67 [Messaging] タブ

注記：このタブで、タッチスクリーンイベントのイベントクラス定義を設定するほか、イベントキュー（event queue）を初期化し、変数の関連付けを行います。[Threads] メニューで [Touch Panel Framework on sf_touch_panel_I2C] を追加した時点で、タッチイベントが自動的に生成されています。

- [Touch Event] クラスを選択します。
- [Touch Subscribers] メニューで、[New Subscriber] ボタンをクリックします。

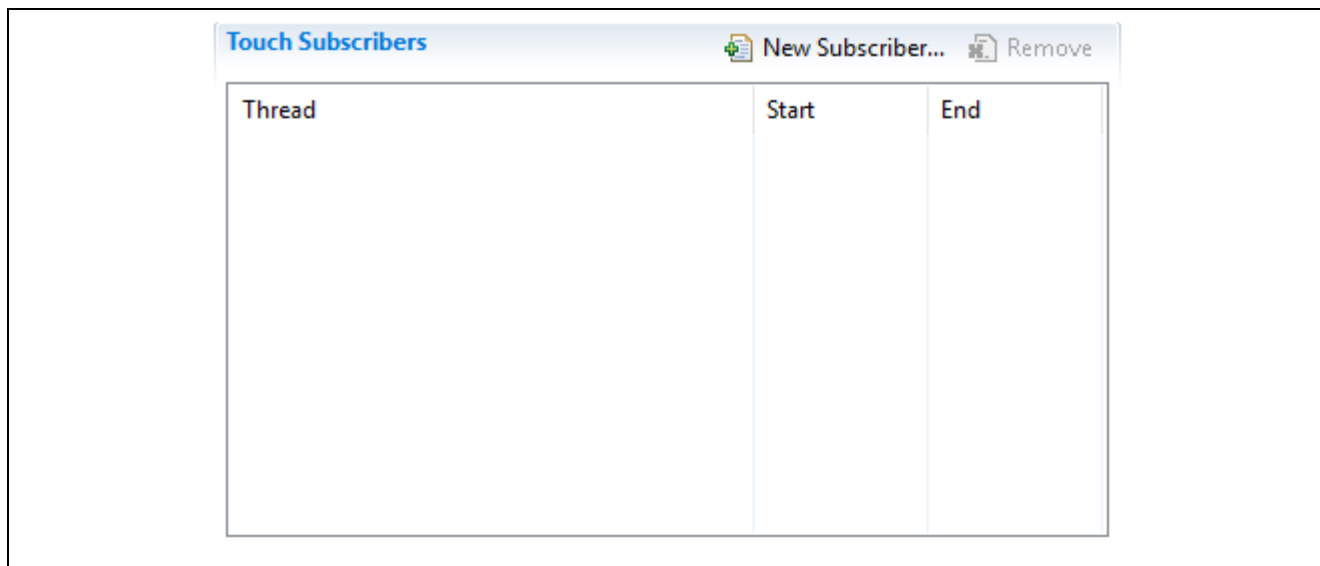


図 68 [Messaging] タブ

- [New Subscribers] ダイアログで、[HMI Thread] を選択します。

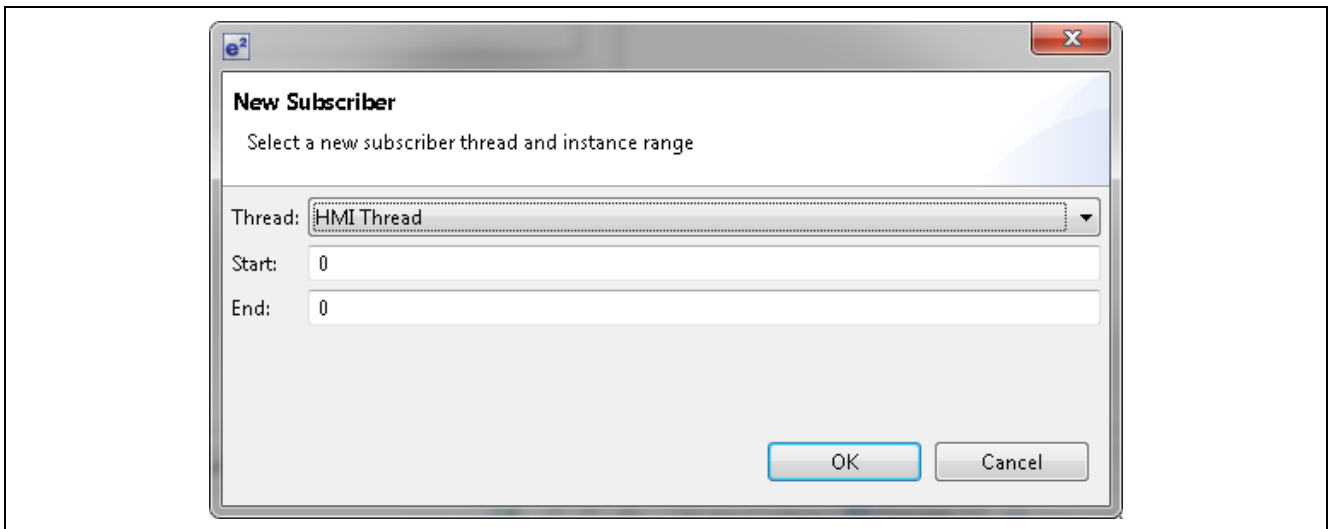


図 69 [New Subscriber] ダイアログ

- [OK] ボタンをクリックします。
- キーボードで、Ctrl+s によりプロジェクトを保存します。
- [Generate Project Content] ボタンをクリックしてプロジェクトファイルを更新します。

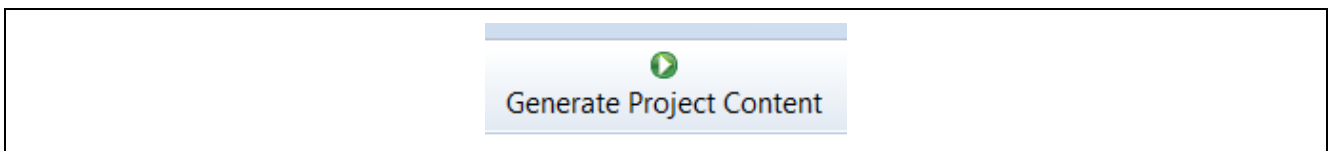


図 70 プロジェクトコンテンツの生成

/src フォルダの中に HMI_thread_entry.c ファイルが生成されます。3.6.1 章で説明したように、アプリケーションスレッドから GUIX システムを初期化する必要があります。

アプリケーションスレッド (HMI thread) による GUIX の初期化は、以下のシーケンスで実行します。

- gx_system_initialize 関数を使用して GUIX を初期化します。
- g_sf_el_gx.p_api > open を使用して、GUIX ドライバを初期化します。
- gx_studio_display_configure 関数により GUIX システムを設定します。
- canvasInit API g_sf_el_gx.p_api > canvasInit により、キャンバスのメモリアドレスを初期化します。
- gx_studio_named_widget_create 関数を使用してメイン画面を作成し、その画面をルートウィンドウ (root window) に接続します。
- Washer Mode (洗濯機の機能モード) という円形のスライダを作成します。
- Garments Mode (衣類モード) という円形のスライダを作成します。
- Water Level (水位) ウィンドウを作成します。
- Temperature (温度) ウィンドウを作成します。
- Temperature (温度) という円形のスライダを作成します。
- gx_widget_show 関数を使用してルートウィンドウを表示し、メイン画面を可視化します。
- gx_system_start 関数を使用して GUIX システムを起動します。
- LCD の GPIO 制御を初期化します。
- PWM ドライバを初期化して開き、TFT パネルのバックライトを制御します。
- GUI エンジンが制御を引き継ぎ、LCD 画面に画像をレンダリングします。

注記：詳細については、付属している Washing Machine (洗濯機) アプリケーションプロジェクトの HMI_thread_entry.c ファイルを参照してください。

3.6.1 画面固有の関数 (Screen specific functions)

各画面に対応するコードは、次の図に示すように、プロジェクトの `src/s7g2_pe_hmi1` ディレクトリにあります。個別の画面に対応する以下の関数を処理するコードです。

- ページの初期化
- 円形のスライダの作成
- ウィジェットラベルの更新
- ウィンドウに対応するイベントの処理
- アイコンの描画
- 特定の角度にアイコンを回転
- アニメーションの処理
- ウィンドウのステータス更新

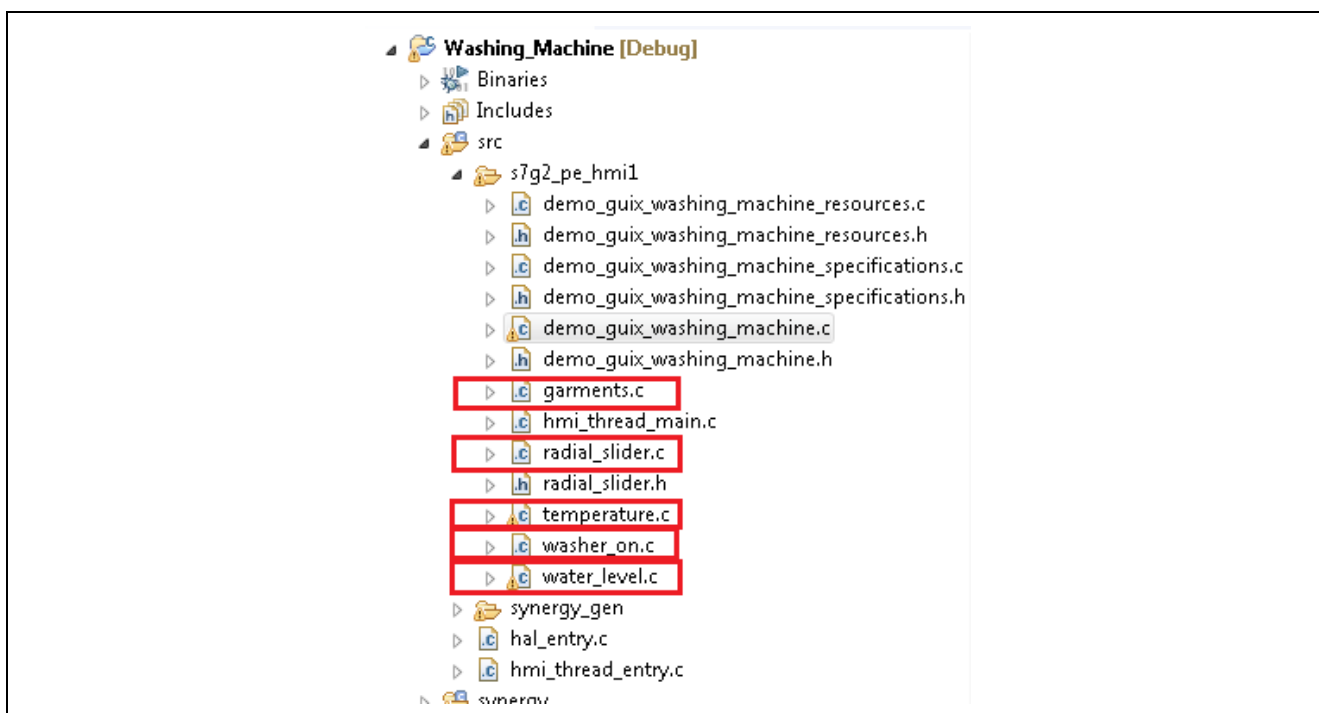


図 71 プロジェクトビューで表示した画面固有のファイル

注記：これらは自動的に生成されるコードではありません。Washing Machine (洗濯機) アプリケーションのウィンドウを制御するためのコードを作成する必要があります。

注記：プロジェクトに対応するすべてのコードを作成した時点でそれらのコードをコンパイルし、PE-HMI ボードでテストすることができます。

注記：以下の章で説明するように、付属の Washing Machine (洗濯機) アプリケーションプロジェクトをインポートし、ビルドして、PE-HMI ボードでテストすることができます。

4. Washing Machine (洗濯機) アプリケーションの実行 (Running the Washing Machine Application)

4.1 プロジェクトのインポート、ビルド、およびロード (Importing, building and loading the project)

このパッケージに付属する『Synergy プロジェクトインポートガイド』(r11an0023eu0119-synergy-ssp-import-guide.pdf)を参照してください。プロジェクトを e² studio にインポートしてビルドする手順が掲載されています。付属している Washing_Machine.zip ファイルには、完成したプロジェクトが含まれています。

4.2 アプリケーションのロードとデバッグ (Loading and Debugging the Application)

- PE-HMI1 の電源を入れ、J-Link Lite Cortex M debugger を PC と PE-HMI1 に接続します。
注記：この時点で、アプリケーションを対象ハードウェア上で実行する準備が整っていません。アプリケーションを実行するには、以下の手順が必要です。
- [Debug] ドロップダウンメニューをクリックし、デバッグオプションを表示します。

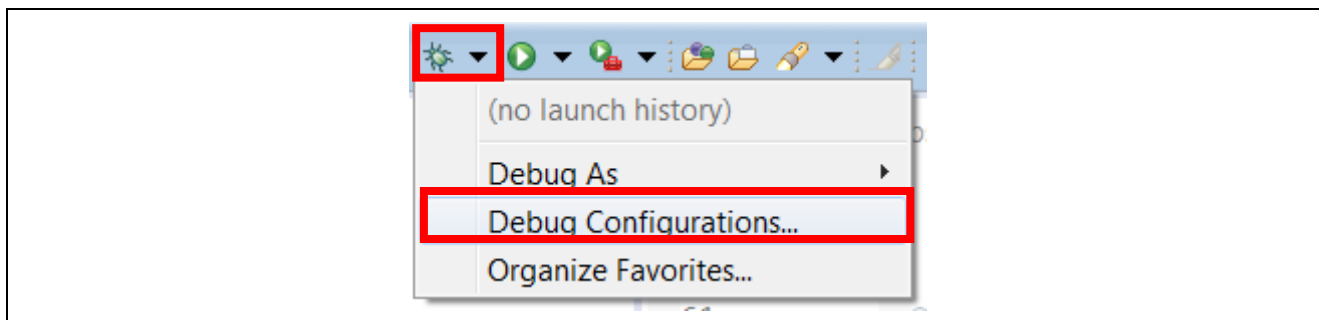


図 72 デバッグオプション

- [Debug Configurations...] オプションを選択します。
- [Renesas GDB Hardware Debugging] から、[Washing_Machine Debug] を選択します。
- [Debug] ボタンをクリックしてデバッグを開始します。

4.3 アプリケーションの検証 (Verifying the Application)

Washing_Machine (洗濯機) アプリケーションをダウンロードし、PE-HMI ボードで実行した時点で、図 3 に示すようなメイン画面が表示されます。画面の右側に表示されている [Garments] (衣類)、[Water Level] (水位)、[Temperature] (温度) の各ボタンを順にクリックします。図 2 に示す GUI 形式のさまざまな画面が表示されます。

画面の詳細は、3.3 章で説明しています。

[Washer] (洗濯機の機能) 画面を選択し、以下の機能を使用してさまざまな項目を設定することができます。

- 大きい円の中に、小さい円形のボール形状でアイコンが表示され、Washer (洗濯機の機能) 設定項目を [Very Light] (ごく少量) から [Rinse and Spin] (すすぎと脱水)、まで任意の項目に移動することができます。
- [Wash] (洗濯機の機能) 選択項目である [Medium] (中) または [Quick Wash] (短時間洗濯) にタッチすると、小さいアイコンが右側の [wash] (洗濯機の機能) 設定項目に移動します。

[Garments] (衣類) ボタンを選択し、[Garment] (衣類) 画面で以下の選択項目を使用して、さまざまな設定を行います。

- 大きい円の中の小さい円形のボール形状をしたアイコンが、[Light Colors] (明るい色) から [Workout Clothes] (体操着) まで、[Garments] (衣類) の各選択項目へと移動することができます。
- 希望する衣類の選択項目にタッチすると、小さい円形のボールも適切な選択項目に移動します。たとえば、[Denim] (デニム) または [Wool] (ウール) にタッチすると、小さいボールのアイコンが衣類の適切な選択項目に移動します。

[Water Level] (水位) 画面を選択し、以下の機能を使用してさまざまな項目を設定することができます。

- 大きな [Water Level] (水位) アイコンの中にある [Vertical Slider] (垂直スライダ) を、[Very Low] (ごく低位) から [Extra High] (ごく高位) までのさまざまな温度設定項目に移動します。
- [Water Temperature] (水温) の選択項目である、[Very Low] (低温) または [Extra High] (高温) にタッチします。垂直スライダが、適切な水温設定項目に移動します。

ステータスウィンドウでは、洗濯サイクルの残り時間 (remaining time wash cycle)、水温 (temperature of the water)、洗濯する衣類 (garments under wash)、シミュレーションによって求めた洗濯の残り時間 (remaining time for the simulated wash) などのステータスが表示されます。

5. 次の手順 (Next Steps)

1. renesassynergy.com/kits には、製品サンプルキットである PE-HMI1 に関する詳細や、その製品の『Quick Start Guide』(クイックスタートガイド)、設計データ、注文情報、他の役立つアプリケーションプロジェクトが掲載されています。
2. 次の各 URL では、Synergy プラットフォームの 3 つの主要要素に関する詳細を確認し、それらに関連するコンポーネントやドキュメントをダウンロードすることができます。
 - A. ソフトウェア : renesassynergy.com/software
 - B. ハードウェア : renesassynergy.com/hardware
 - C. ソリューションギャラリー : renesassynergy.com/solutionsgallery
3. 以下の各 URL にアクセスし、以下の要素の詳細を確認します。
 - A. マイクロコントローラ : renesassynergy.com/mcus
 - B. 開発ツール : renesassynergy.com/tools
 - C. キット : renesassynergy.com/kits

6. 参考資料 (References)

GUIX のリファレンスドキュメントとリンクを、以下に示します。

- 『GUIX Studio User's Guide』 : <https://rtos.com/solutions/guix-studio/embedded-ui-design-tool/>
- 『GUIX User's Guide』 : https://synergygallery.renesas.com/media/products/1/275/en-US/Synergy_X-Ware_Docs.zip
- 『GUIX™ Synergy Port Framework のモジュールガイド』 : <https://www.renesas.com/en-us/software/D6001976.html>

SSP とアプリケーションのリファレンスドキュメントとリンクを、以下に示します。

- PE-HMI1 用 GUIX Hello World アプリケーションプロジェクト
- → Renesas ホームページ (<https://www.renesas.com/ja-jp/>) → プラットフォーム → Synergy プラットフォームで、サンプルプログラム「[GUIX "Hello World" for PE-HMI1 - Application Project](#)」を検索してください。
- 『SSP ユーザマニュアル』 : <https://www.renesas.com/ja-jp/products/synergy/software/ssp.html>
- Renesas Synergy Knowledge Base : https://en-us.knowledgebase.renesas.com/English_Content/Renesas_Synergy%E2%84%A2_Platform/Renesas_Synergy_Knowledge_Base

日本語サイト : https://ja-jp.knowledgebase.renesas.com/Japanese_Content/Renesas_Synergy%E2%84%A2_%E3%83%97%E3%83%A9%E3%83%83%E3%83%88%E3%83%95%E3%82%A9%E3%83%BC%E3%83%A0

ホームページとサポート窓口

以下のヘルプとサポートをご活用ください。

ドキュメント : renesassynergy.com/docs

ナレッジベース : renesassynergy.com/knowledgebase

フォーラム : renesassynergy.com/forums

トレーニング : renesassynergy.com/training

ビデオ : renesassynergy.com/videos

すべての商標および登録商標はそれぞれの所有者に帰属します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2018年6月28日	-	第 1.00 版 発行 英文版（資料番号 r11an0299eu0100-synergy-guix-advanced-application-washing-machine、リビジョン Rev1.00、発効日 2018年3月27日）を翻訳

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
 2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
 3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
 4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
 5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、
家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、
金融端末基幹システム、各種安全制御装置等
当社製品は、データシート等により高信頼性、Harsh environment向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じて、当社は一切その責任を負いません。
 6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
 7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシートにおいて高信頼性、Harsh environment向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
 8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
 9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
 10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
 11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
 12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。
- 注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

(Rev.4.0-1 2017.11)



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<https://www.renesas.com/contact/>