

emWin for RX GUI開発トレーニング (実践2)



Using AppWizard
RX72N Envision 評価ボード用



Embedded Software / Hardware Solutions

試作から量産ツールまで

「emWin for RX」について

RXグループMCUに対応するSEGGER社製「**emWin for RX**」GUIソフトウェアは、**RXマイコングループ（RXv1、RXv2、RXv3）** システム環境で無償で商用利用し、開発することが可能です。1999年から販売しており、emWinはワールドワイドで産業制御、スマートホーム・省エネ対応IoT機器、テレコム、自動車産業、家電、医療器械等、航空電子機器様々なアプリケーションで使用されています。emWin GUIはRXマイコンデバイスの内蔵RAMだけで使用出来ますので、外付けRAMが不要です。emWin GUIは使用中のOS（RX用embOS、μITRON等）で対応可、OS無しも使用可能です。**ロイヤリティー費用、量産制限等はありません。**

GUI開発トレーニング（実践2）

デモサンプル内容：写真アルバム

（オンボードSW2プッシュボタン及びスワイプ操作で動かします）

emWin AppWizardツールから画面UIをデザインして写真アルバムの実装方法を解説します。



実行手順

TASK-1

1

事前準備

開発環境 e2Studio、QE for Display [RX] をインストールします。

QE for Display [RX] のSmart Configuratorを使用してRX72N Envisionボード用BSPパッケージを作成します。

TASK-2

2

画面作成

emWin AppWizardツールから画面UIをデザインして、オンボードSW2 プッシュボタンから操作できるようにイベントジョブを設定します。

TASK-3

3

ビルド & 動作確認

e2Studioプロジェクト環境でBSP+UIデザインアプリケーションコードをビルドして、RX72N Envisionボード環境で動作を確認します。

TASK-1

事前準備

- ① 開発環境 e2Studio、QE for Display [RX] をインストールします。
- ② QE for Display [RX] のSmart Configuratorを使用してRX72N Envisionボード用BSPパッケージを作成します。
- ③ アプリケーションで使用するフォントをパソコンにインストールします。

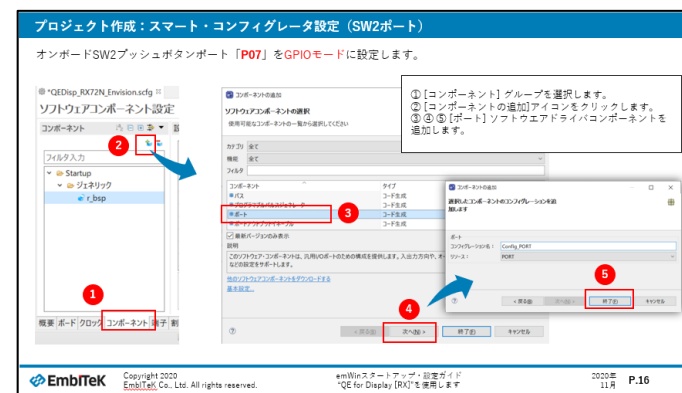
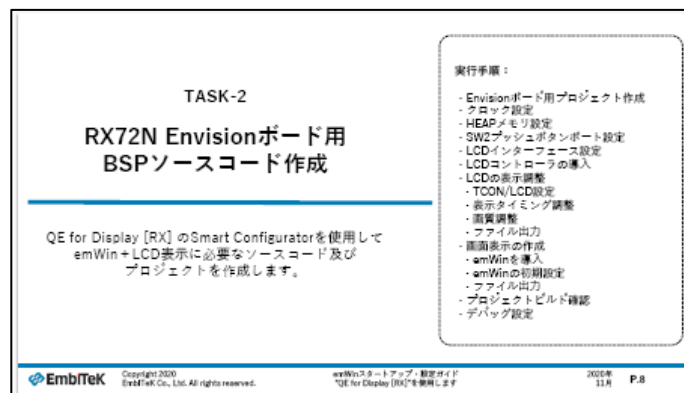
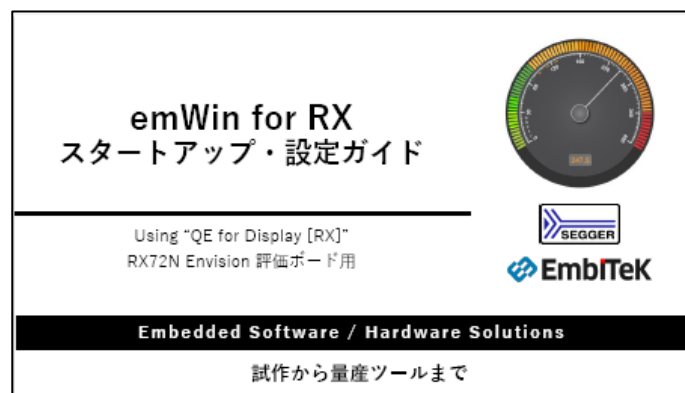
本セッションでは以下の開発環境を使用します。

- Host PC: Windows10 / Windows11
- 統合開発環境 e2 studio 2023-10
- ディスプレイ対応開発支援ツール パッケージ QE for Display V3.1.1

開発環境 e2Studio、QE for Display [RX] のインストール及びSmart Configuratorを使用してRX72N Envisionボード用BSPパッケージの作成手順は別資料「**emWin for RXスタートアップ・設定ガイド**」（ファイル名：emWin_for_RX72N_qedisp_startup.pdf）をご参照ください。

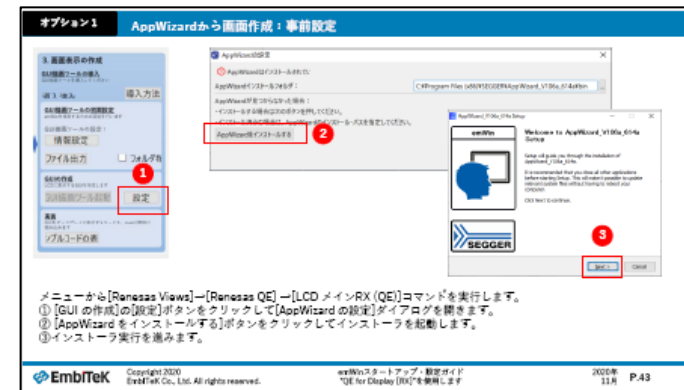
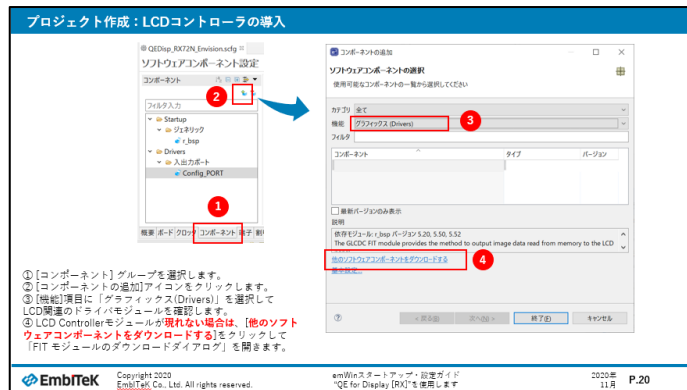
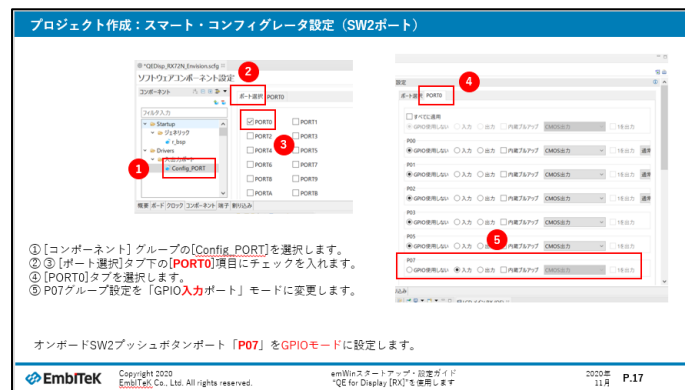
https://www.embitek.co.jp/technote/emwin/emWin_for_RX72N_qedisp_startup.pdf

Smart ConfiguratorでRX72N Envisionボード用BSPパッケージの作成の時に、オンボードSW2プッシュボタン関連の設定も行ってください。



事前準備：BSP+ベースプロジェクトの作成

オンボードSW2プッシュボタンポート「P07」をGPIOモードに設定します。



関連ドキュメント・参考資料：

- RXファミリ QE for Display [RX] アプリケーションノート

<https://www.renesas.com/jp/ja/document/apn/rx-family-qe-display-rx-application-note>

本セッションではグーグル社・アドビ社が共同開発した無償提供の「Noto Sans JP」フォントを使用します。

Noto Sans JP (Google)

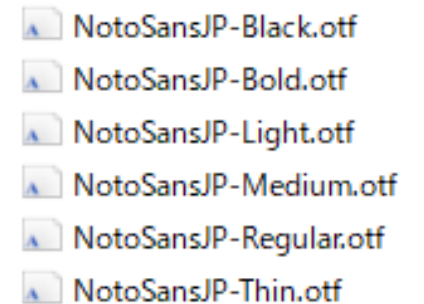
<https://fonts.google.com/noto/specimen/Noto+Sans+JP>

他、商用ライセンスフォントのご利用が可能です。

① 以下URLから「Noto Sans JP」フォントパッケージをダウンロードします。

<https://fonts.google.com/download?family=Noto%20Sans%20JP>

② ZIPパッケージの (*.ttf) フォントをマウス右クリックメニューからパソコンにインストールします。

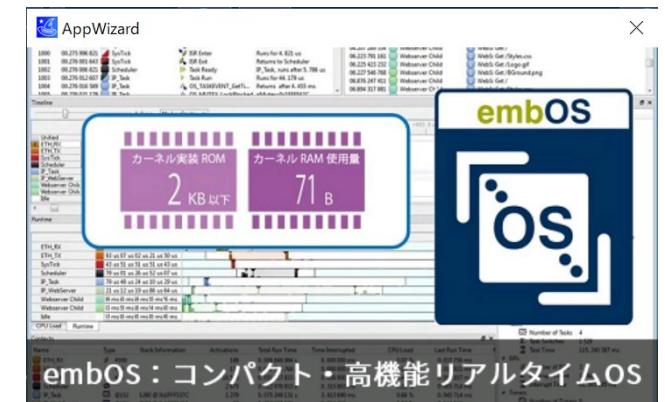


上記フォントの御社製品利用については、各ライセンスの利用条件をご確認の上、ご利用ください。

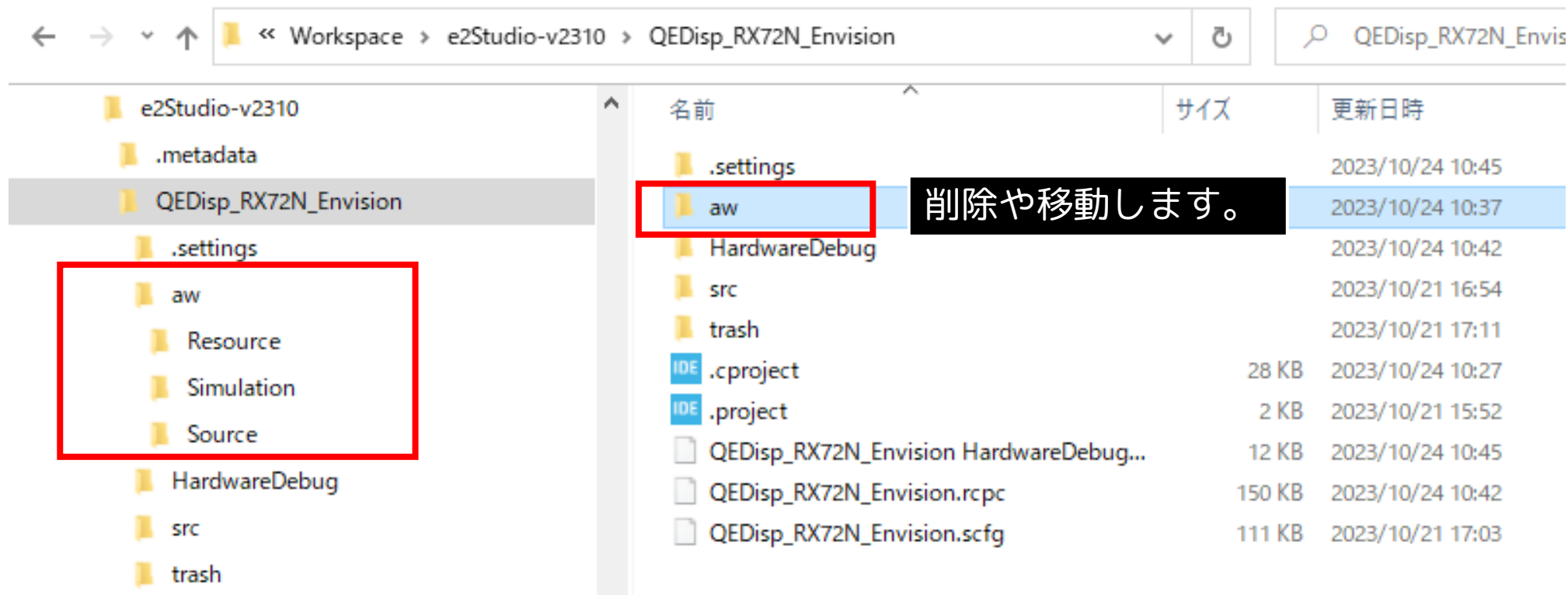
TASK-2

画面作成（UIデザイン）

emWin AppWizardツールから画面UIをデザインして必要なリソース
及びアプリケーションコードを作成します

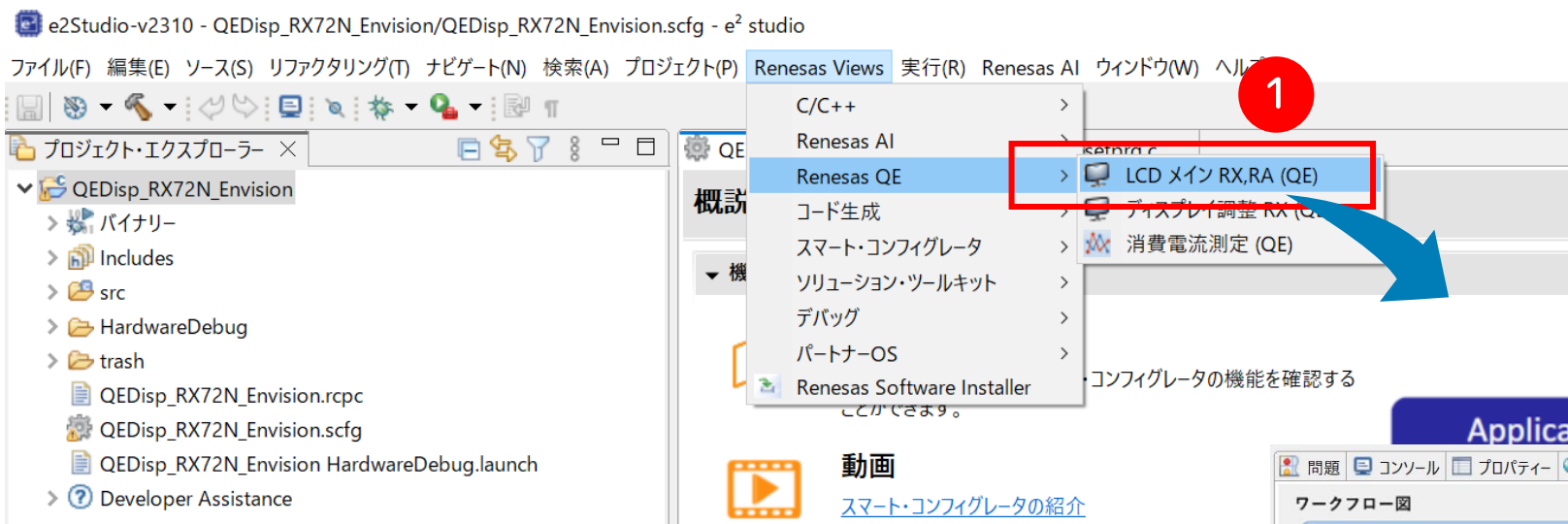


e2Studio QE環境でのデフォルト「aw」AppWizardプロジェクトを使用しますので、
e2Studioワークスペースの既存の「aw」フォルダを削除するか別のローカルディレクトリに移動してください。



既存の「aw」フォルダを削除や移動してからe2Studioツールを起動します。

画面作成：AppWizardツールを起動します



- ① e2StudioからRX72N Envisionボード用プロジェクトを起動し、メニューから[Renesas Views]→[Renesas QE] → [LCD メインRX (QE)]コマンドを実行します。
- ② [GUI 描画ツールの起動]ボタンをクリックしてAppWizardツールを起動します。

画面作成：画像リソースを追加

File Edit Project Resource Help

Add objects

Screen Box Button Image

Text Slider Rotary Switch

Edit Multiedit Window QRCode

Hierarchic tree

Object Id

Resources

1

2

Images

Image	Name	Width	Height	Stock	Prj.	Ref.	Format
-------	------	-------	--------	-------	------	------	--------

Import to project

Cleanup

Close

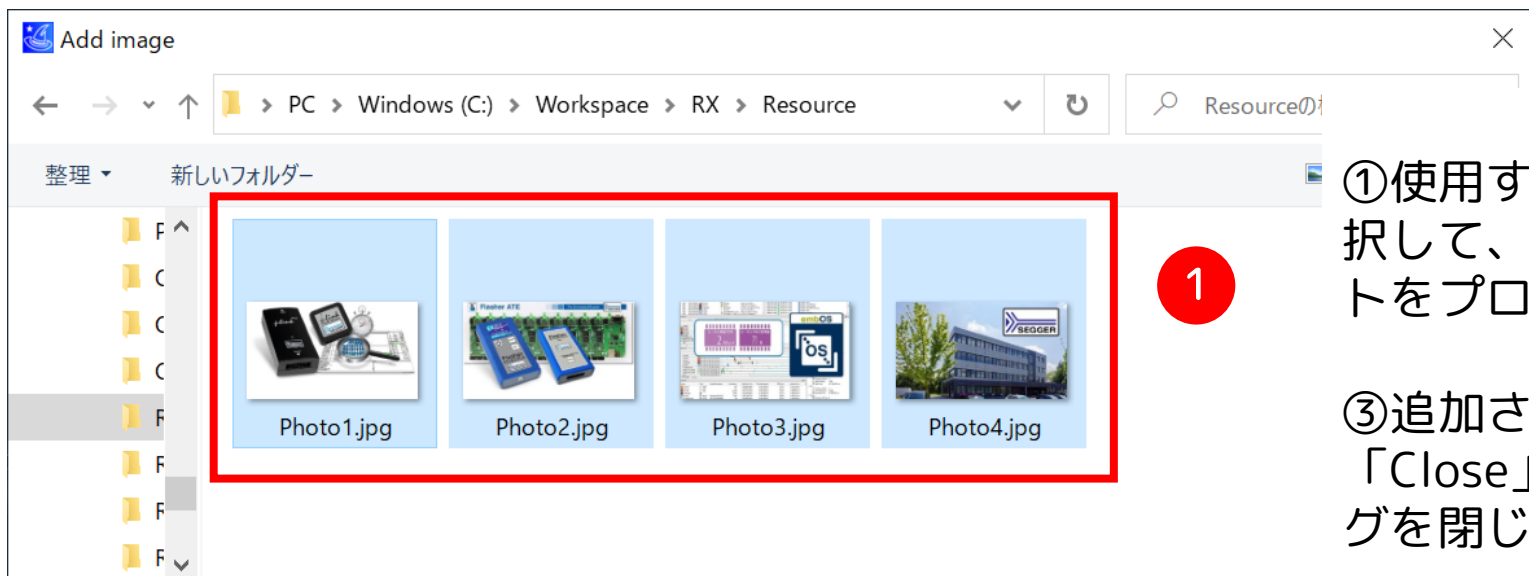
Interaction

+/- [Emitter Signal Job Receiver Comment

① 「Resource → Edit Images」 コマンドでイメージリソースダイアログを開きます。

② 「Import to project」 ボタンを押してAdd Imageダイアログからデモサンプルに使用する画像オブジェクト（Photo1～Photo4）を追加します

画面作成：画像リソースを追加



①使用する画像ファイル（Photo1～Photo4）を選択して、②「開く」ボタンを押して画像オブジェクトをプロジェクトリソースに追加します。

③追加されたイメージリソース一覧を確認し、「Close」ボタンを押してイメージリソースダイアログを閉じます。



画面作成：スクリーン作成

The screenshot displays the emWin GUI development software interface, which is divided into several panels:

- Add objects:** A panel on the left containing various GUI object icons. The 'Screen' icon, which shows a rectangle with a smaller rectangle inside, is highlighted with a yellow box and a red circle with the number '1'. A yellow arrow points from this icon to the 'Editor' panel.
- Editor:** The central workspace where the GUI is designed. It shows a large rectangular area with a gray and white checkerboard pattern, representing a transparent screen. The area is outlined with a dashed magenta border and has small square handles at the corners for resizing.
- Properties:** A panel on the right that displays the properties of the selected object. It includes fields for 'Id' (ID_SCREEN_00), 'Position' (0, 0), 'Size' (480, 272), and 'Predraw'/'Postdraw' settings. Below these are visual representations of motion and persistence settings.
- Hierarchic tree:** A panel at the bottom left showing a list of objects in the project. The 'Screen' object with ID 'ID_SCREEN_00' is highlighted with a yellow box.
- Interactions:** A panel at the bottom center for defining object interactions, with columns for '+/-', 'Emitter', 'Signal', 'Job', 'Receiver', and 'Comment'.
- Resources:** A panel at the bottom left showing available fonts, images, and other resources.

A text box with a white background and black text is overlaid on the bottom right of the Editor panel, containing the following Japanese text:

Screenボタンを押して画面にスクリーンオブジェクトを追加します

画面作成：イメージウィジェットを作成

The screenshot displays the emWin GUI development tool interface with three numbered steps highlighted:

- ①** In the **Hierarchic tree** on the left, the **Image** object under **ID_SCREEN_00** is selected.
- ②** In the **Add objects** panel, the **Image** button is highlighted with a red box and a red arrow points to the center of the **Editor** canvas.
- ③** In the **Properties** panel on the right, the **Position** settings (Left: 0, Top: 0, Right: 0, Bottom: 0) are highlighted with a red box.

A text box at the bottom left contains the following instructions:

①オブジェクトウィンドウから「ID_SCREEN_00」を選択します。
② Imageボタンを押してオブジェクトを追加します。
③ 画面中心に設定します。
TOP = 0, BOTTOM = 0, LEFT = 0, RIGHT = 0

画面作成：タイトルウィンドウを作成

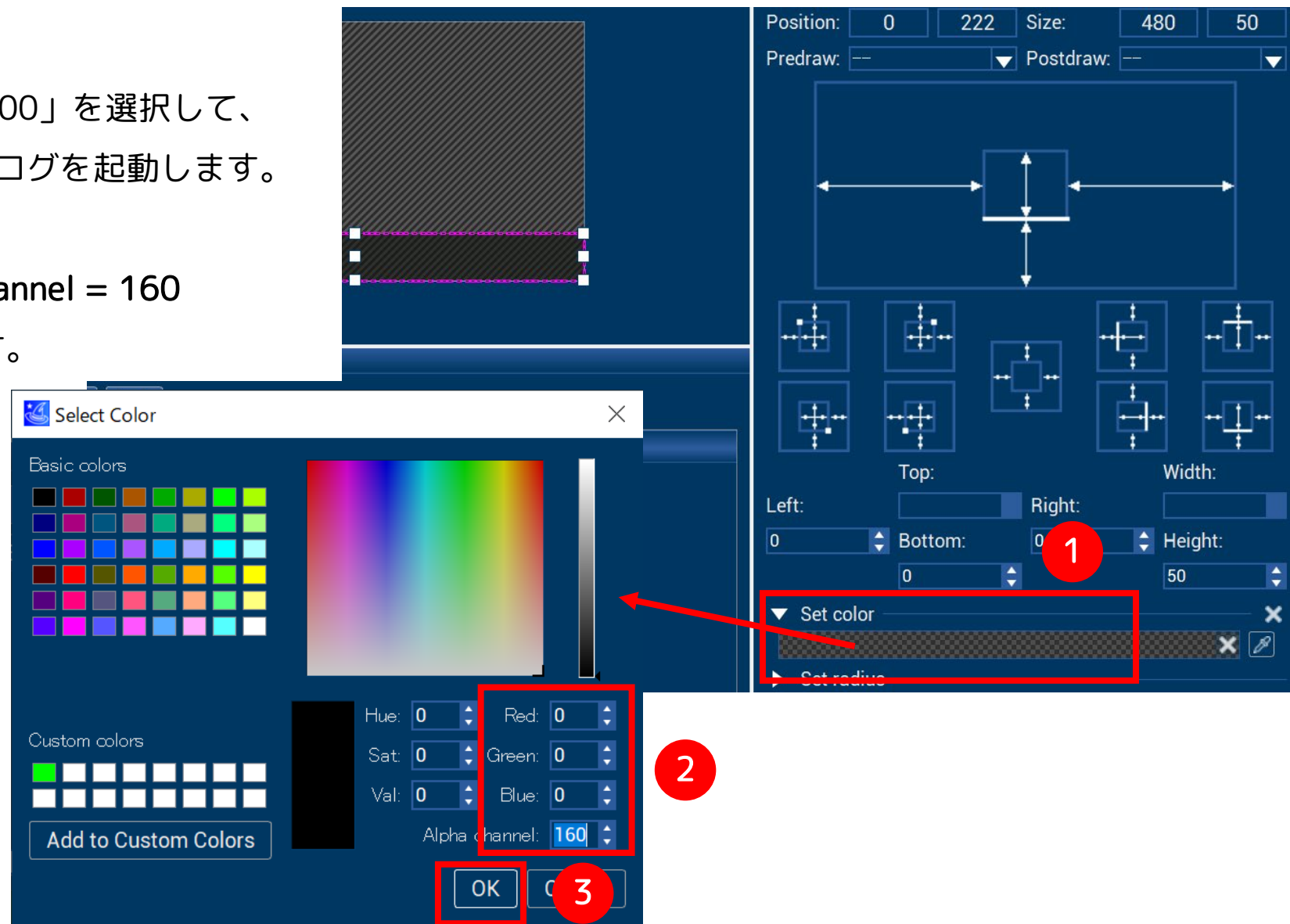
① オブジェクトウィンドウから「ID_SCREEN_00」を選択します。

② Boxボタンを押してオブジェクトを追加します。

③ 画面下側に配置します。
HEIGHT = 50, BOTTOM = 0, LEFT = 0, RIGHT = 0

画面作成：タイトルウィンドウの設定

- ①オブジェクトウィンドウから「ID_BOX_00」を選択して、Propertyウィンドウからカラー設定ダイアログを起動します。
- ② カラー設定は以下の通りです。
RED = 0, GREEN = 0, BLUE = 0, Alpha channel = 160
- ③ 「OK」ボタンを押して設定を保存します。



画面作成：テキストウィンドウを作成

File Edit Project Resource Help

Add objects

Screen Box Button Image

Text Slider Rotary Switch

Edit Multiedit Window QRCode

Hierarchical tree

Object	Id
Screen	ID_SCREEN_00
Image	ID_IMAGE_00
Box	ID_BOX_00
Text	ID_TEXT_00

Editor

①オブジェクトウィンドウから「ID_BOX_00」を選択します。
② Textボタンを押してオブジェクトを追加します。
③ 画面下側に配置します。
HEIGHT = 50, BOTTOM = 0, LEFT = 0, RIGHT = 0

Properties

Size: 480 50

Postdraw: ---

Interactions

Emitter	Signal	Job	Receiver	Comment
+				

Left: 0 Top: 0 Right: 0 Width: 480 Height: 50

Set text color

Set background color

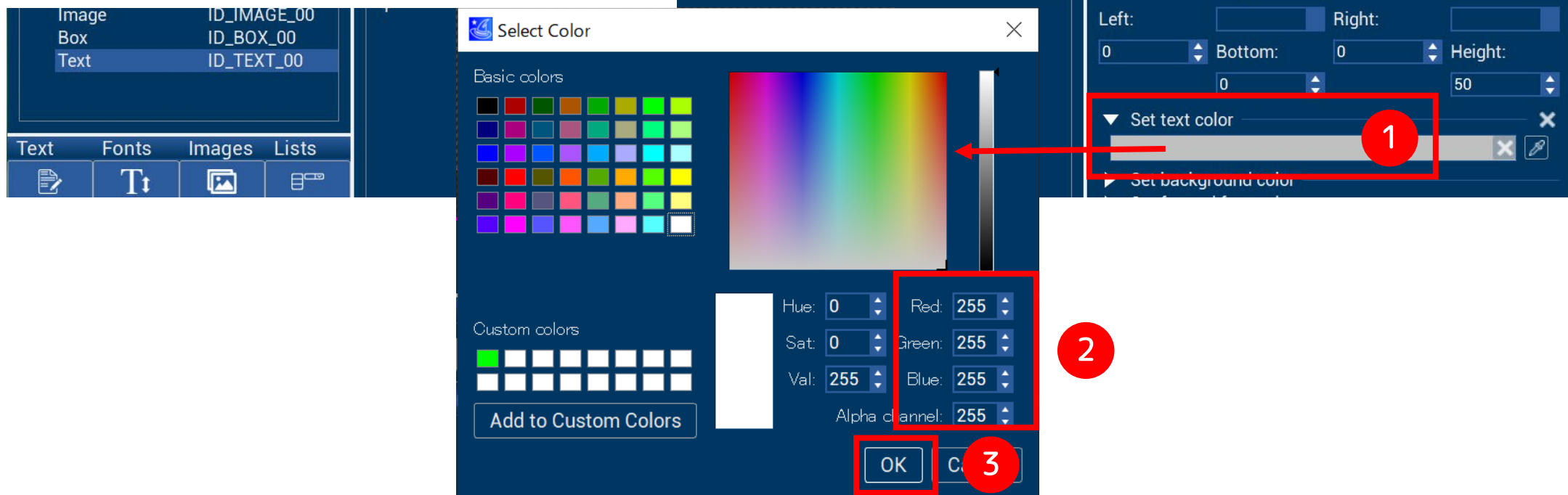
画面作成：テキストウィンドウの設定

①オブジェクトウィンドウから「ID_TEXT_00」を選択して、Propertyウィンドウからカラー設定ダイアログを起動します。

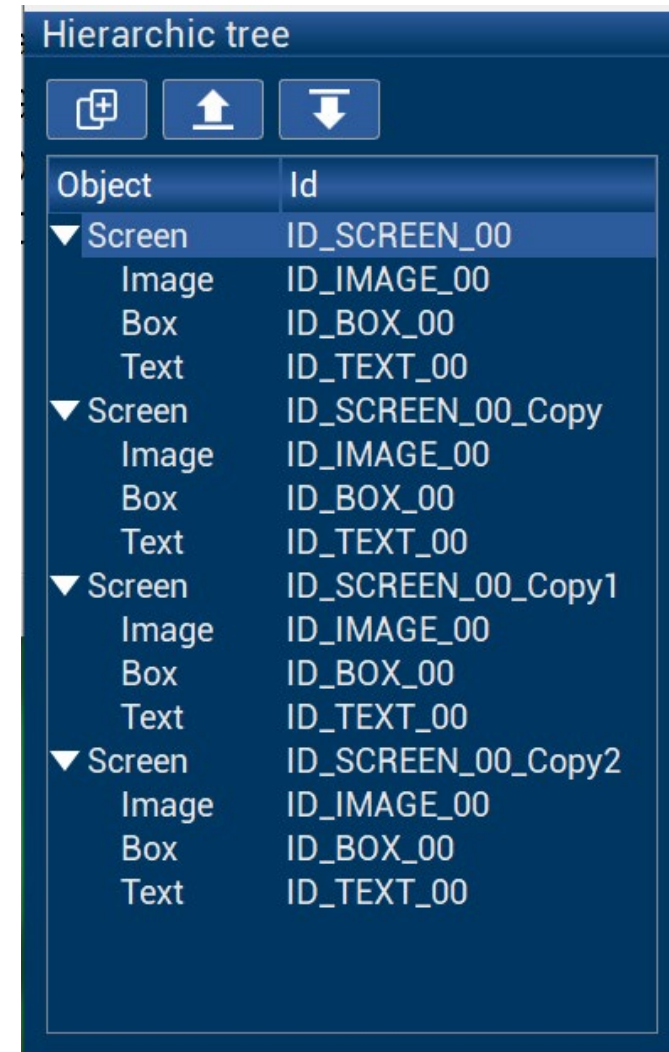
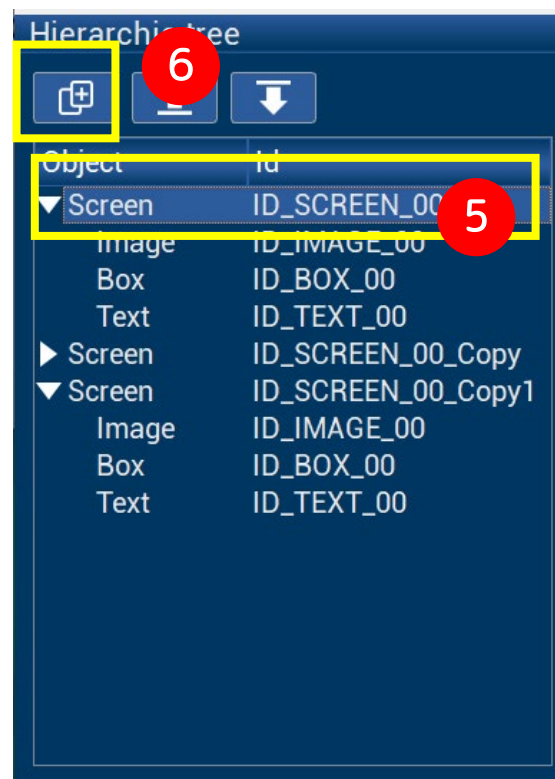
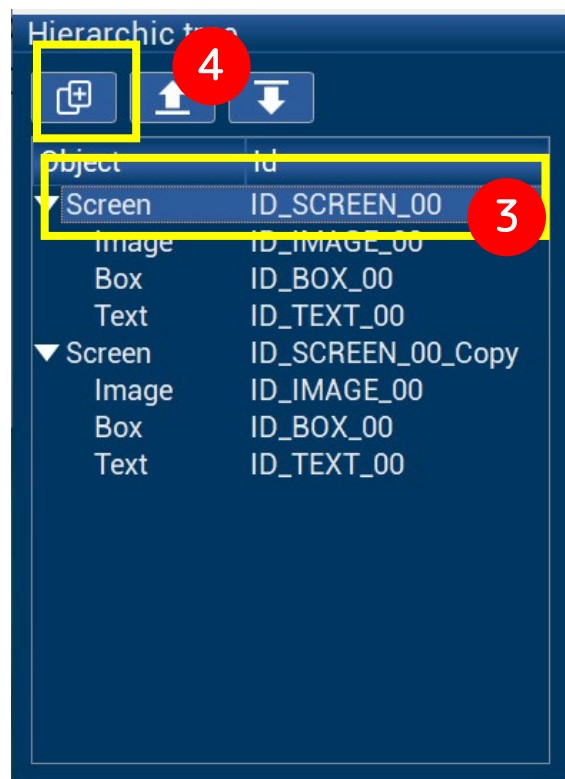
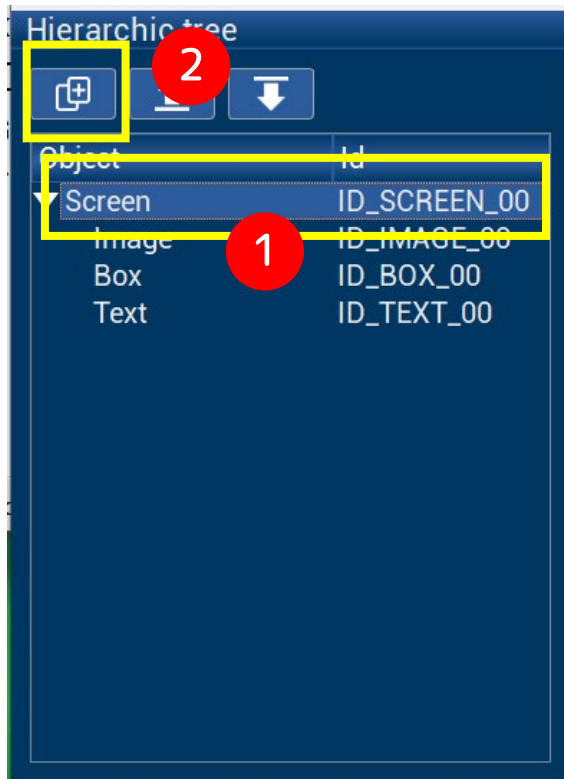
② 白色を設定します。カラー設定は以下の通りです。


RED = 255, GREEN = 255, BLUE = 255

③ 「OK」ボタンを押して設定を保存します。



画面作成：スクリーンテンプレートをコピー



- ①③⑤ オブジェクトウィンドウから「ID_SCREEN_00」を選択して、
②④⑥  アイコンをクリックしてスクリーン及びウィジェット内容の
コピーを作成します。最終的に4つのスクリーンセットを作成します。

画面作成：スクリーンIDを設定



① ② ③ Screen Idを選択して、マウスダブルクリックでIDテキストを編集します。

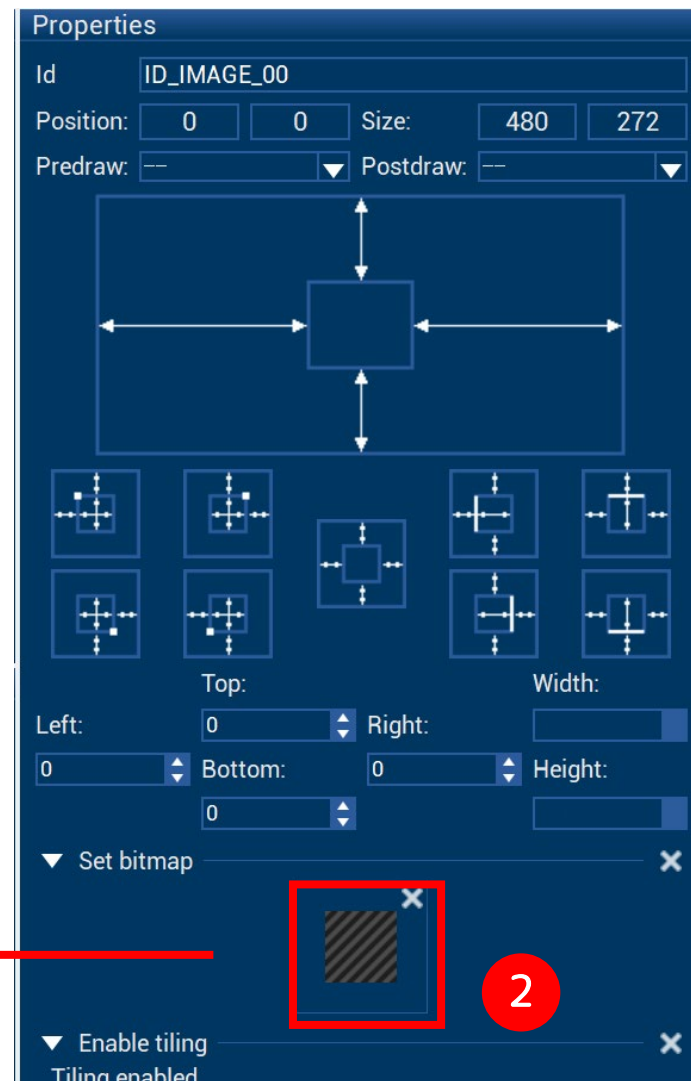
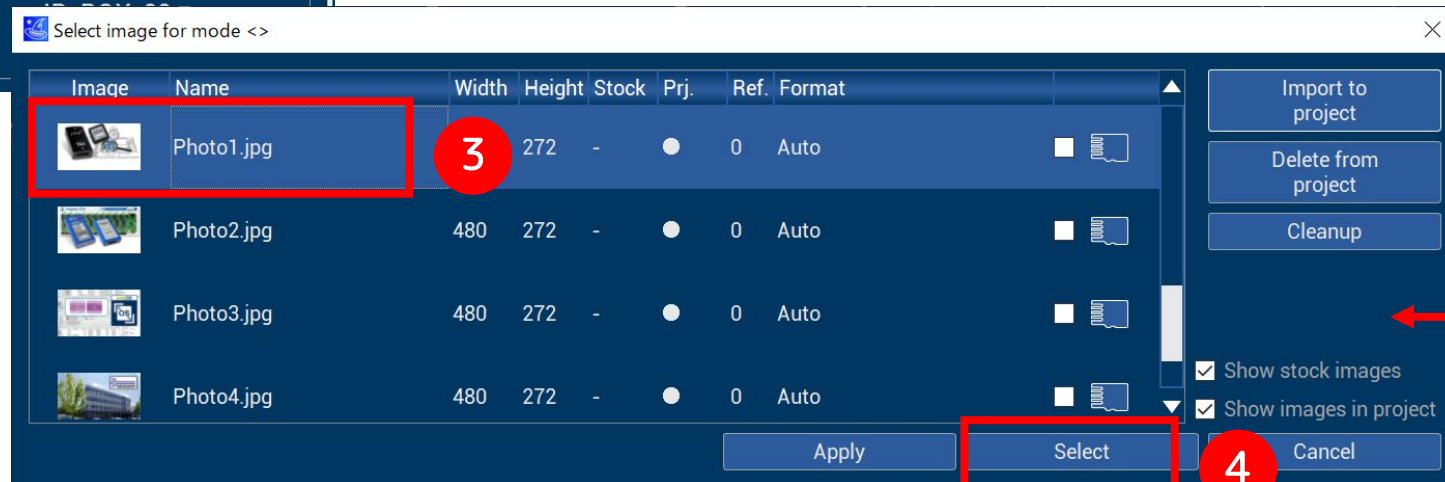
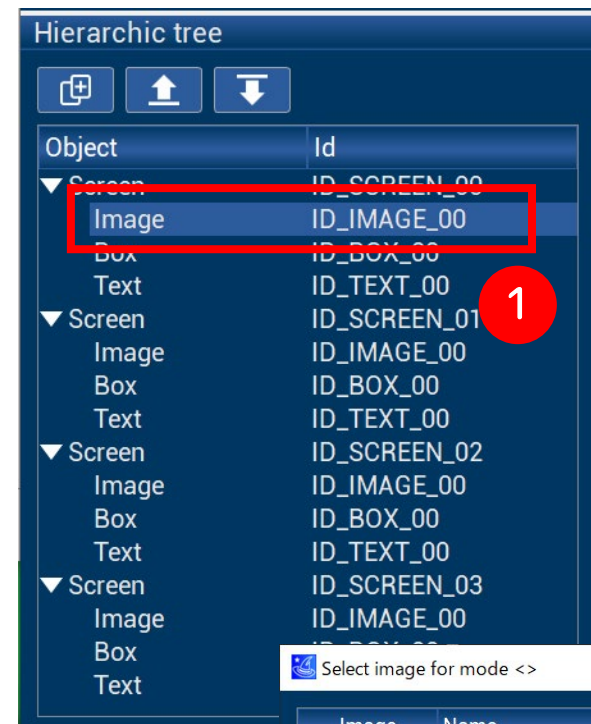
ID_SCREEN_00_Copy → ID_SCREEN_01

ID_SCREEN_00_Copy1 → ID_SCREEN_02

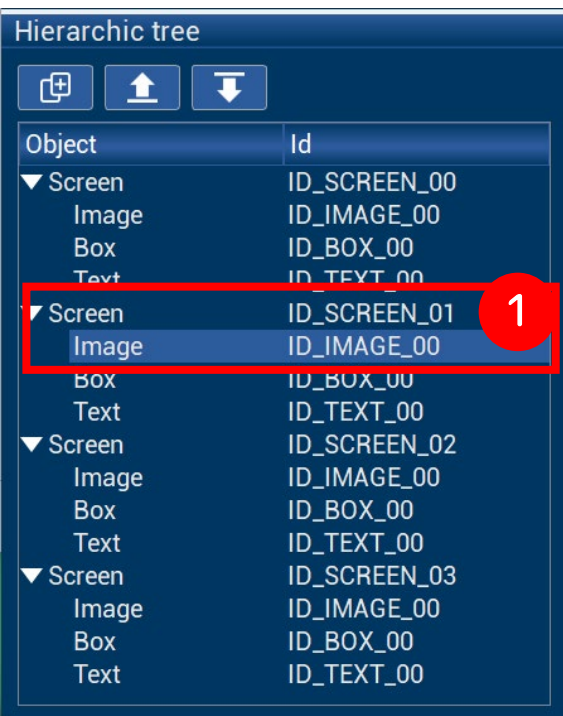
ID_SCREEN_00_Copy2 → ID_SCREEN_03

画面作成：イメージリソースの設定（写真1）

- ① オブジェクトウィンドウから「ID_SCREEN_00」の「ID_IMAGE_00」を選択して、
- ② Propertyウィンドウから「Set bitmap」設定ダイアログを起動します。
- ③ Select Imageダイアログから「Photo1.jpg」を選択します。
- ④ 「Select」ボタンを押して設定を保存します。



画面作成：イメージリソースの設定（写真2）

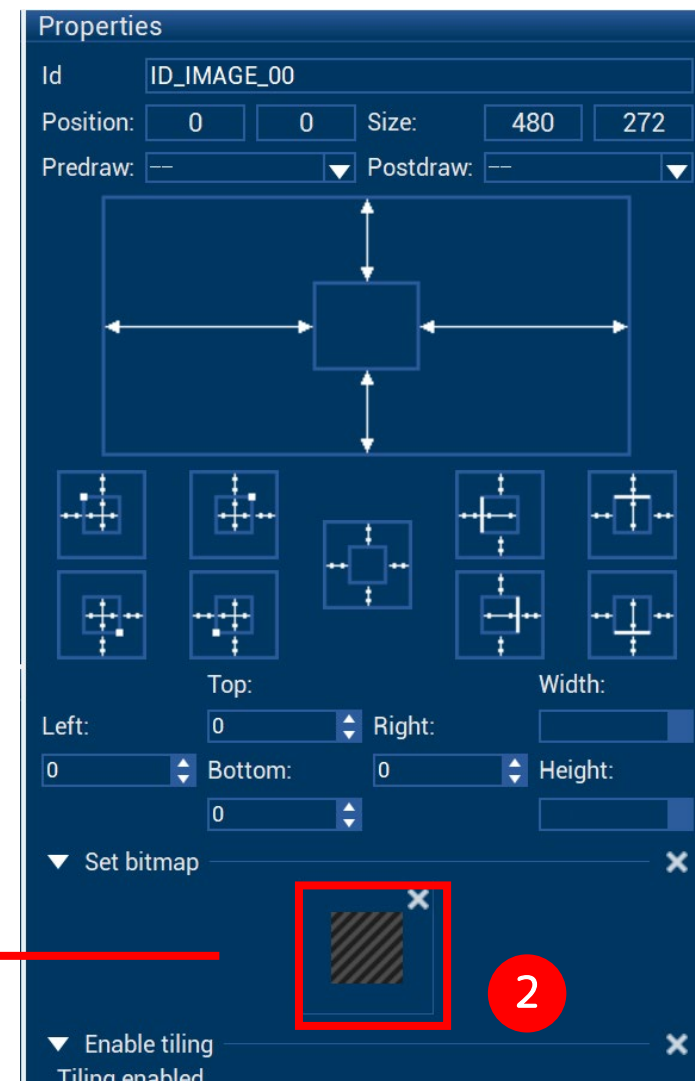
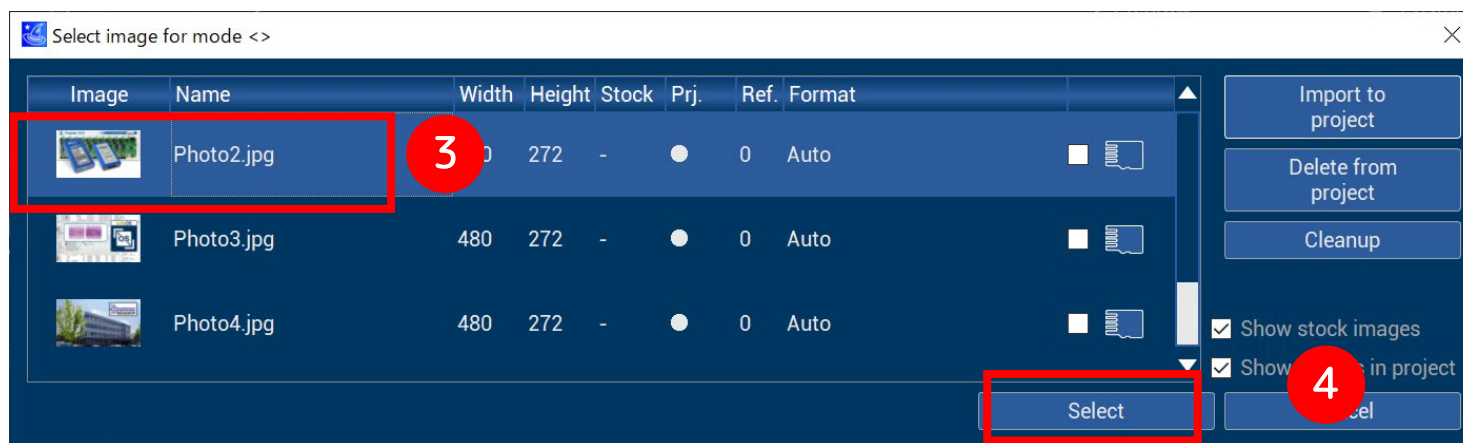


①オブジェクトウィンドウから「ID_SCREEN_01」の「ID_IMAGE_00」を選択して、

② Propertyウィンドウから「Set bitmap」設定ダイアログを起動します。

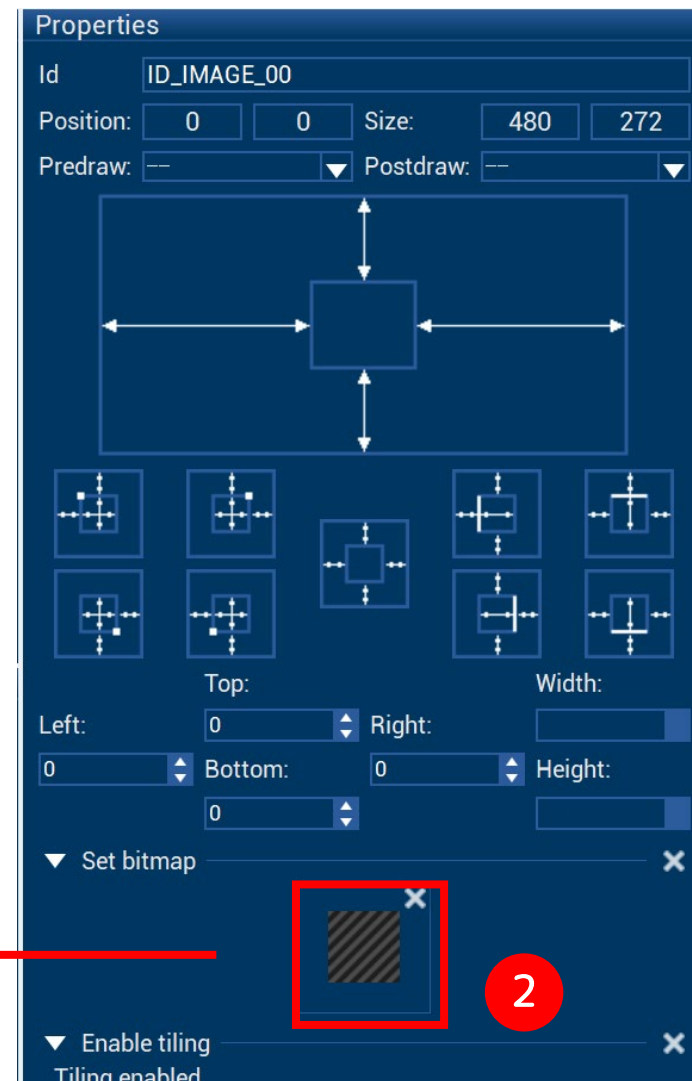
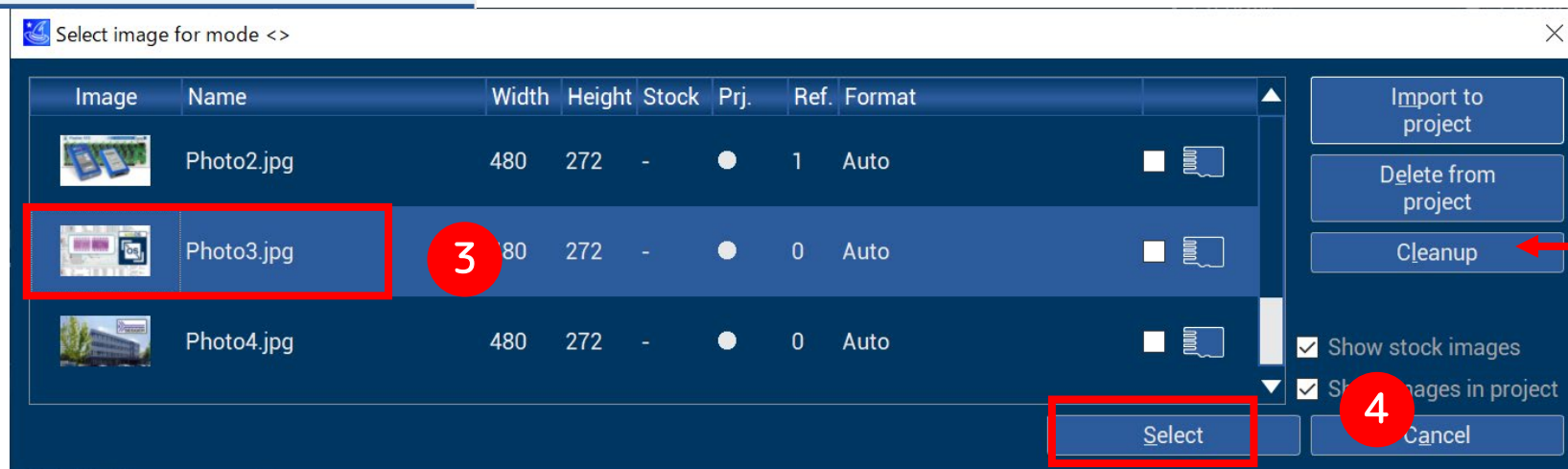
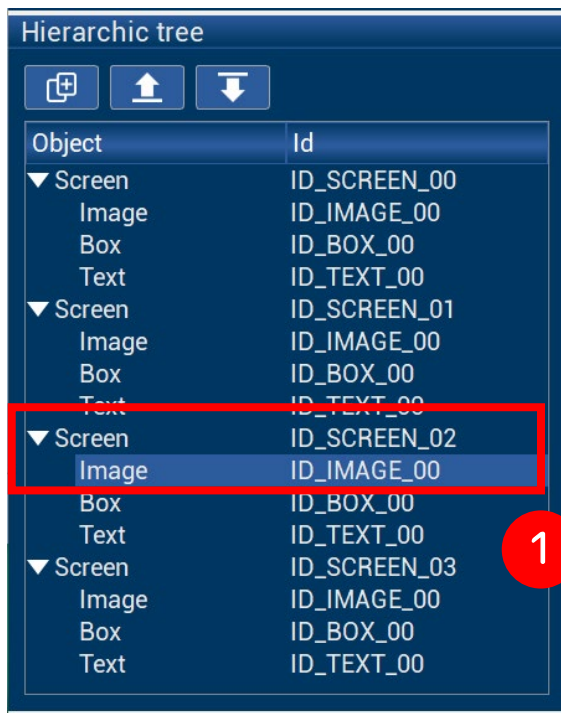
③ Select Imageダイアログから「Photo2.jpg」を選択します。

④ 「Select」ボタンを押して設定を保存します。

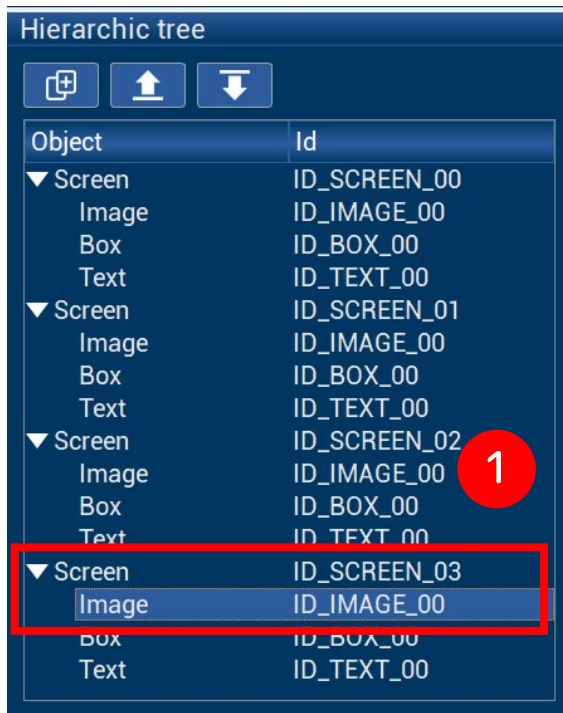


画面作成：イメージリソースの設定（写真3）

- ①オブジェクトウィンドウから「ID_SCREEN_02」の「ID_IMAGE_00」を選択して、
- ② Propertyウィンドウから「Set bitmap」設定ダイアログを起動します。
- ③ Select Imageダイアログから「Photo3.jpg」を選択します。
- ④ 「Select」ボタンを押して設定を保存します。



画面作成：イメージリソースの設定（写真4）

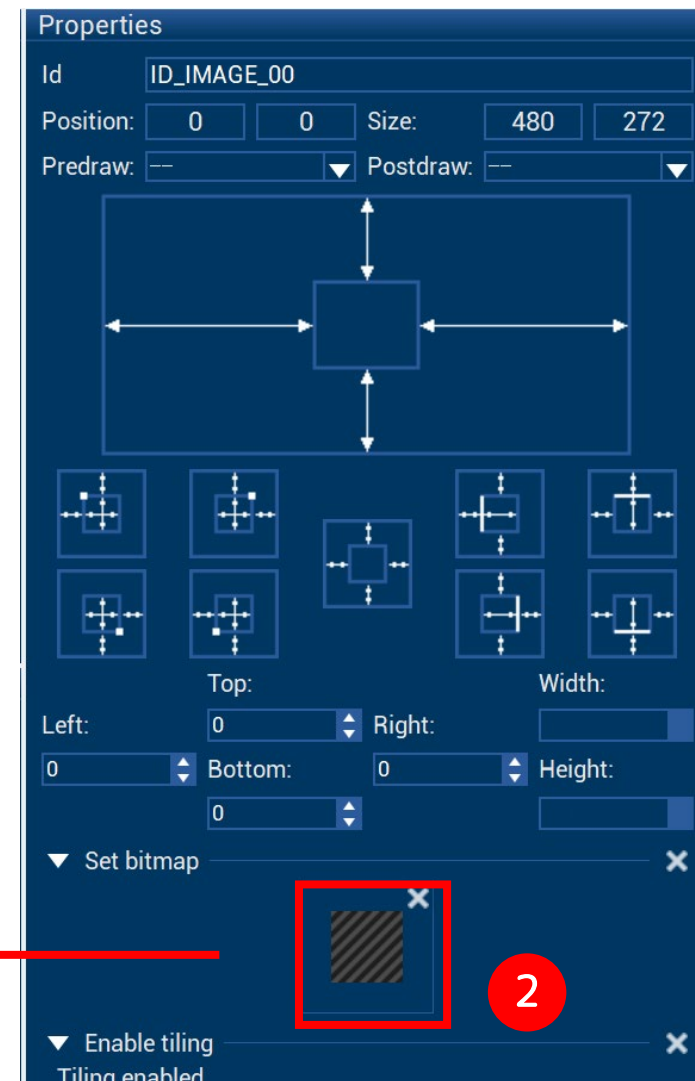
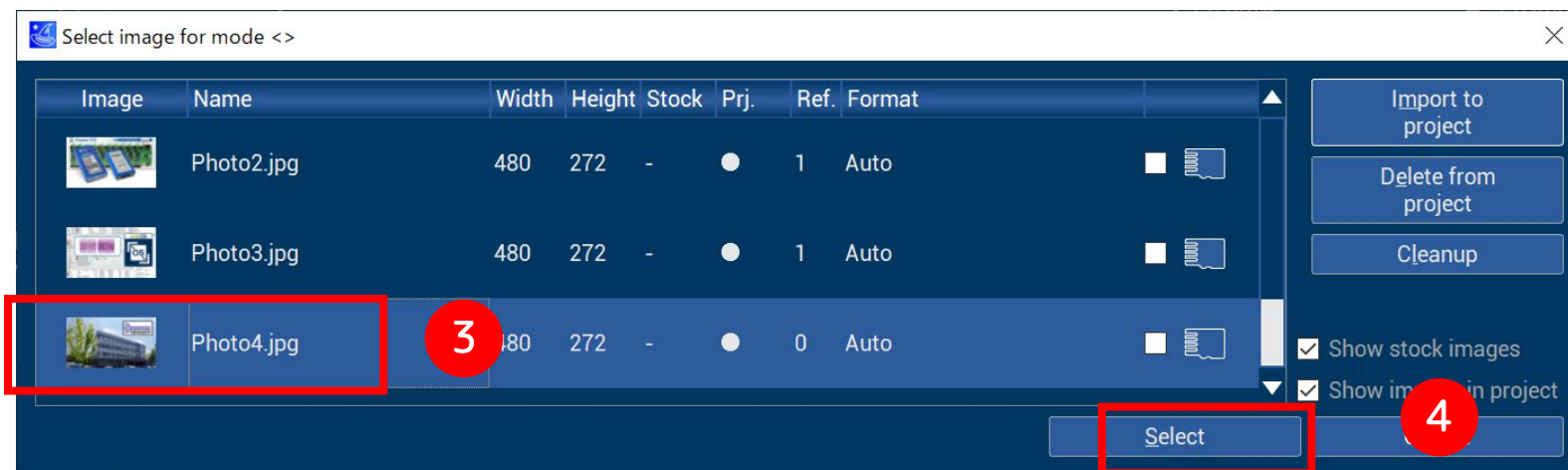


①オブジェクトウィンドウから「ID_SCREEN_03」の「ID_IMAGE_00」を選択して、

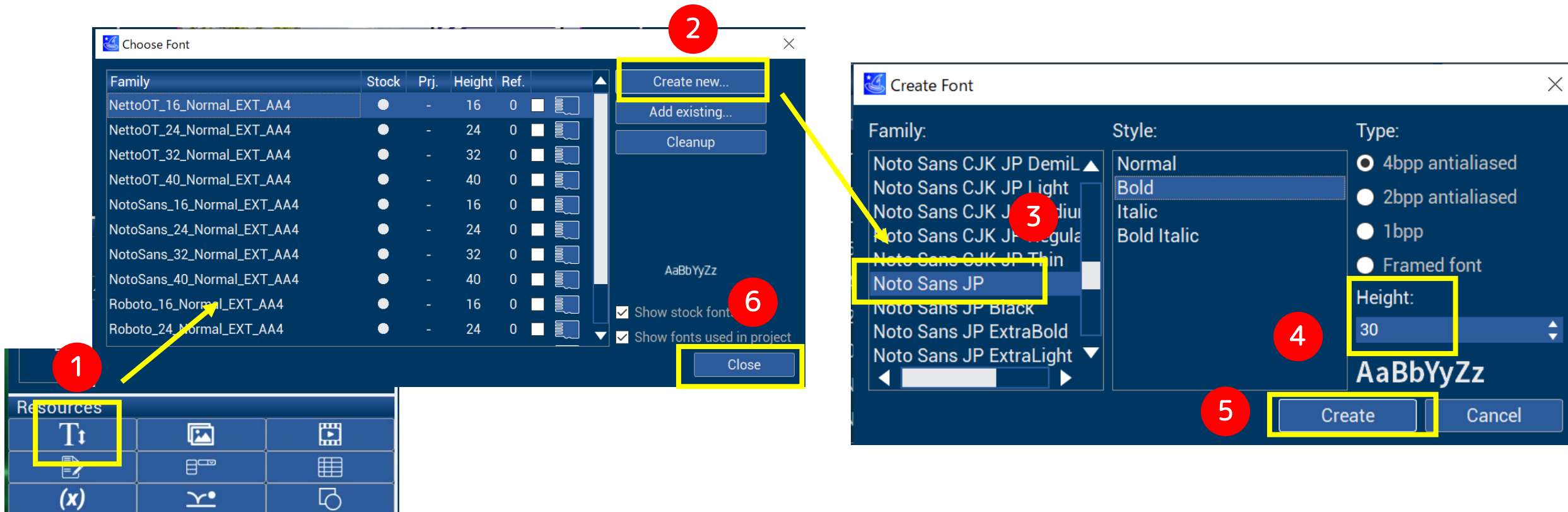
② Propertyウィンドウから「Set bitmap」設定ダイアログを起動します。

③ Select Imageダイアログから「Photo4.jpg」を選択します。

④ 「Select」ボタンを押して設定を保存します。

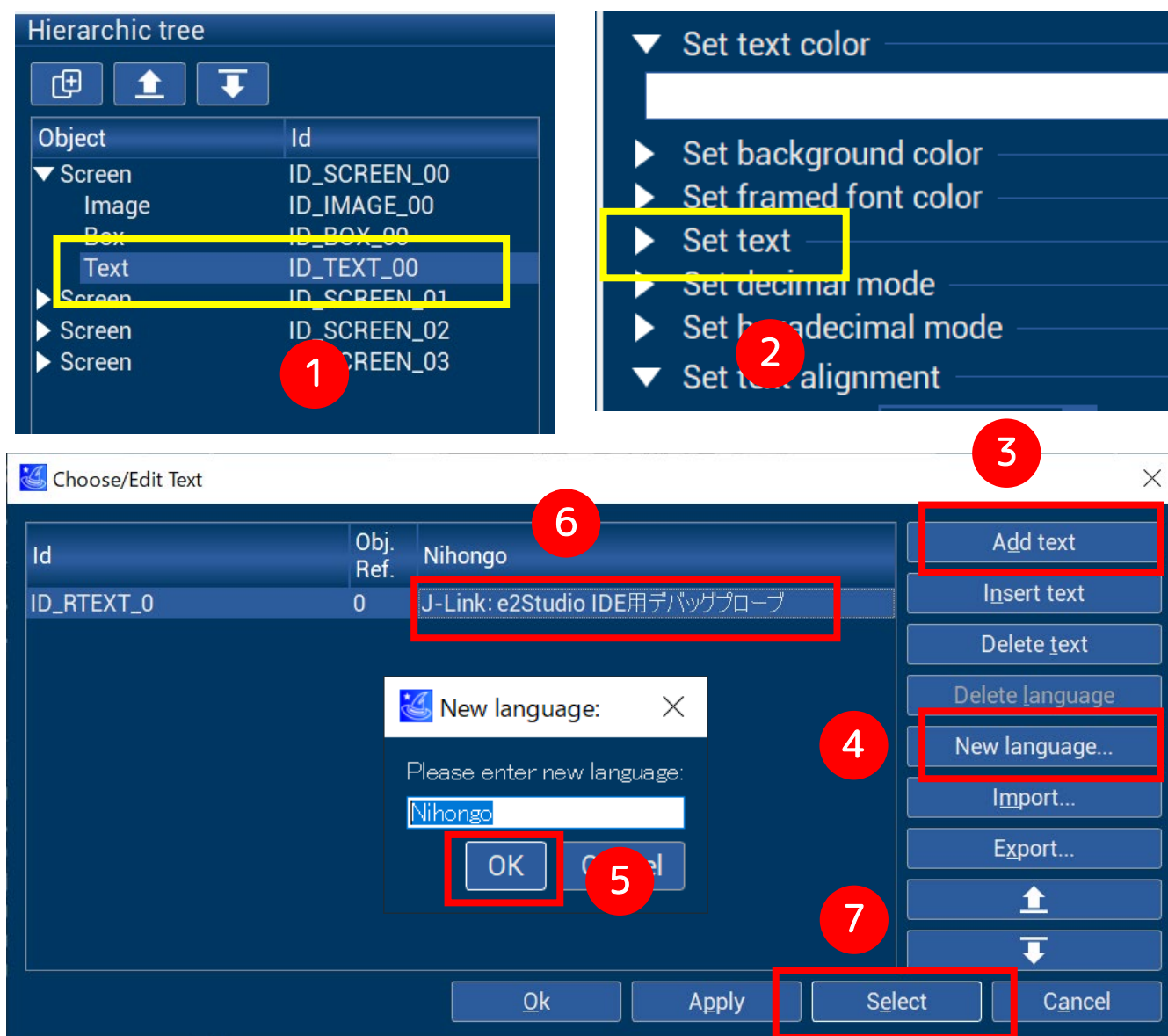


画面作成：フォントリソースを追加



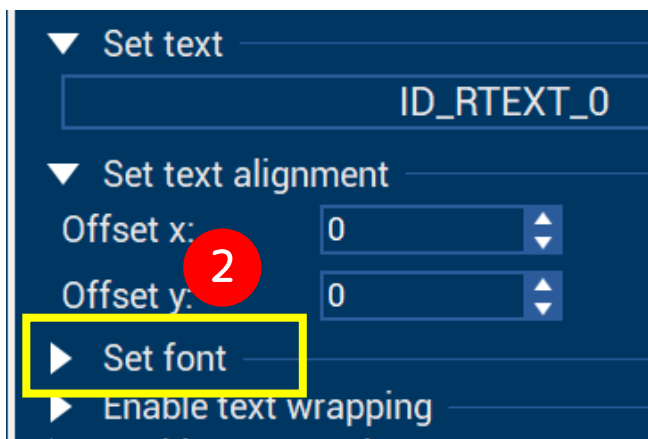
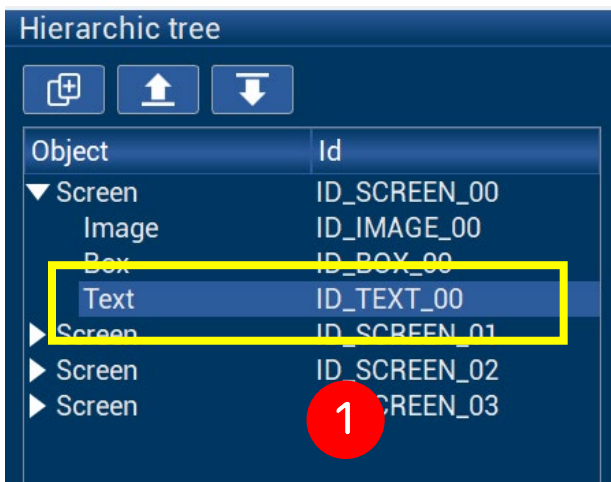
- ① Propertyウィンドウから「Set font」アイコンをクリックしてフォント設定ダイアログを開きます。
- ② 「Create New」ボタンを押してフォント作成ダイアログを開きます。
- ③ 「Noto Sans JP / Bold」フォントを選択
- ④ フォントHeight=30を設定します。
- ⑤ 「Create」ボタンを押してダイアログを閉じます。
- ⑥ 「Close」ボタンを押してフォント作成ダイアログを閉じます

画面作成：テキストタイトルの設定（写真1）

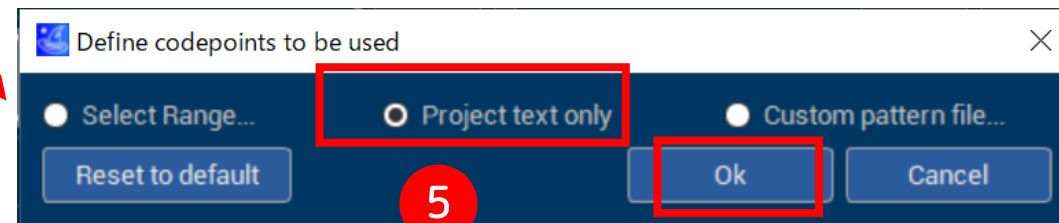
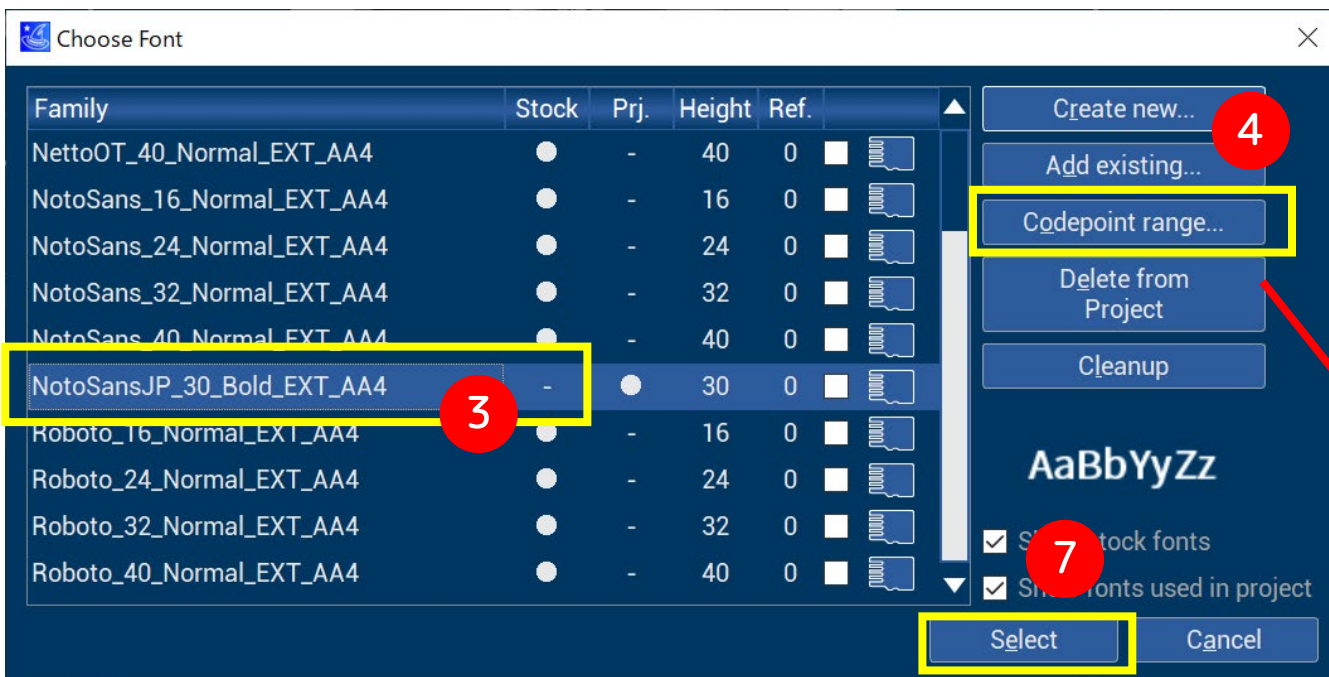


- ① 「ID_SCREEN_00」の「ID_TEXT_00」を選択します。
- ② Propertyウィンドウから「Set Text」アイコンをクリックText設定ダイアログを開きます。
- ③ 「Add text」ボタンを押して「ID_RTEXT_0」オブジェクトを追加します。
- ④ Textダイアログから「New Language」ボタンを押して、
- ⑤ 言語名「Nihongo」を追加します
- ⑥ 「Nihongo」言語項目下の「…」をマウスダブルクリックして「J-Link：e2Studio IDE用デバッグプローブ」文字列を設定します。
- ⑦ 「ID_RTEXT_0」を選択し「Select」ボタンを押してダイアログを閉じます。

画面作成：テキストタイトルの設定（写真1）



- ① 「ID_SCREEN_00」の「ID_TEXT_00」を選択します。
- ② Propertyウィンドウから「Set Font」アイコンをクリックしてFont設定ダイアログを開きます。
- ③ 「NotoSansJP_30_Bold..」フォントを選択し、
- ④ 「Codepoint range」ボタンを押して、
- ⑤ 「Project text only」オプションを選択してプロジェクトで使用する文字分のみフォントデータを作成するように設定します。
- ⑥ 「Ok」ボタンを押して設定を保存
- ⑦ 「Select」ボタンを押してダイアログを閉じます。



画面作成：テキストタイトルの設定（写真1）

File Edit Project Resource Help

Add objects

Screen Box Button Image

Text Slider Rotary Switch

Edit Multiedit Window QRCode

Hierarchic tree

Object	Id
Screen	ID_SCREEN_00
Image	ID_IMAGE_00
Box	ID_BOX_00
Text	ID_TEXT_00
Screen	ID_SCREEN_01
Screen	ID_SCREEN_02
Screen	ID_SCREEN_03
Image	ID_IMAGE_00

Resources

Editor

1:1

J-Link : e2Studio IDE用デバッグプローブ

Interactions

	Emitter	Signal	Job	Receiver	Comment
+					

Properties

Top: Width: Left: Right: Bottom: Height:

Set text color

Set background color

Set framed font color

Set text

Set text alignment

Offset x: Offset y: Alignment:

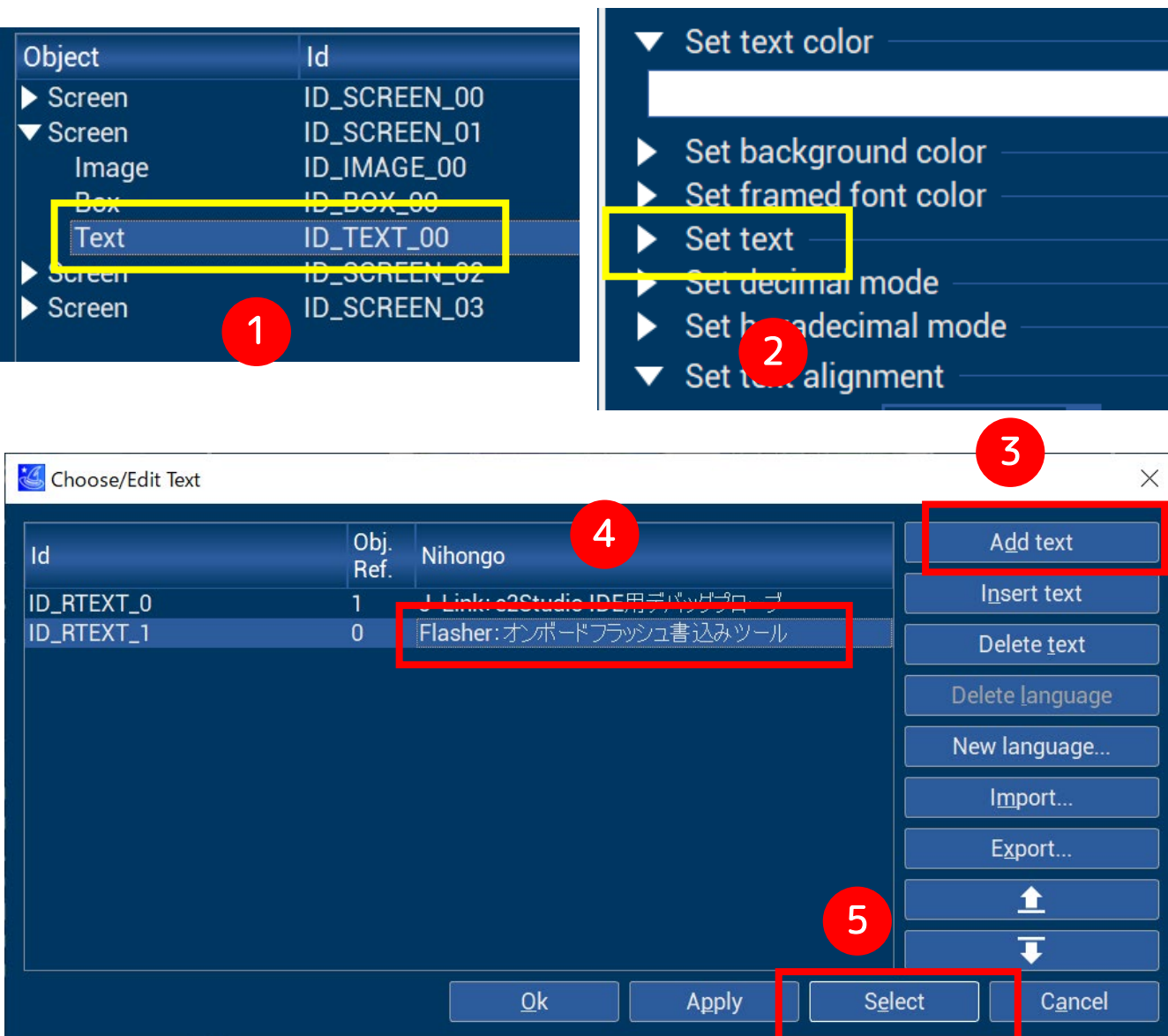
Set font

NotoSansJP_30_Bold_EXT_AA4.xbf

Enable text wrapping

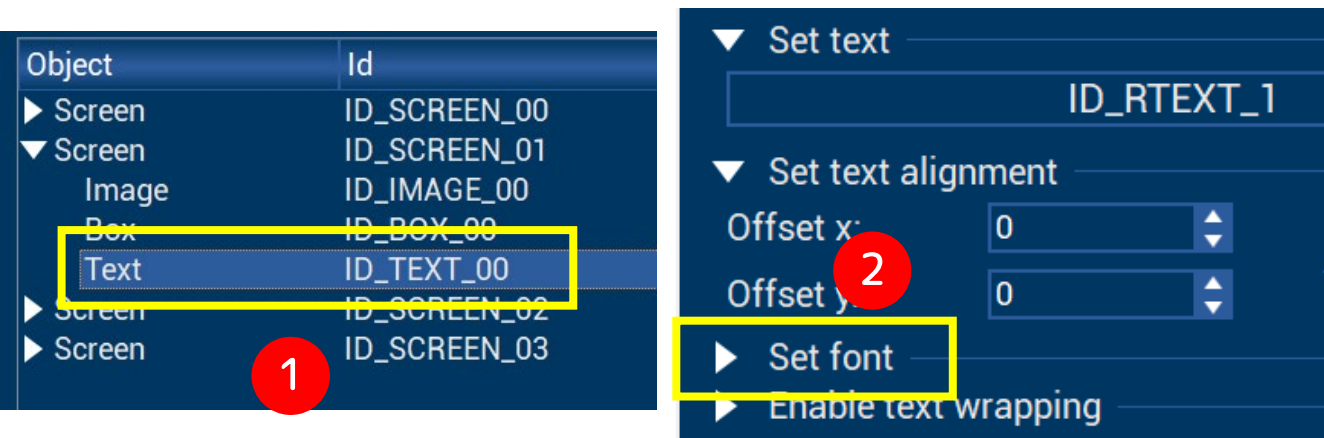
Enable text rotation

画面作成：テキストタイトルの設定（写真2）

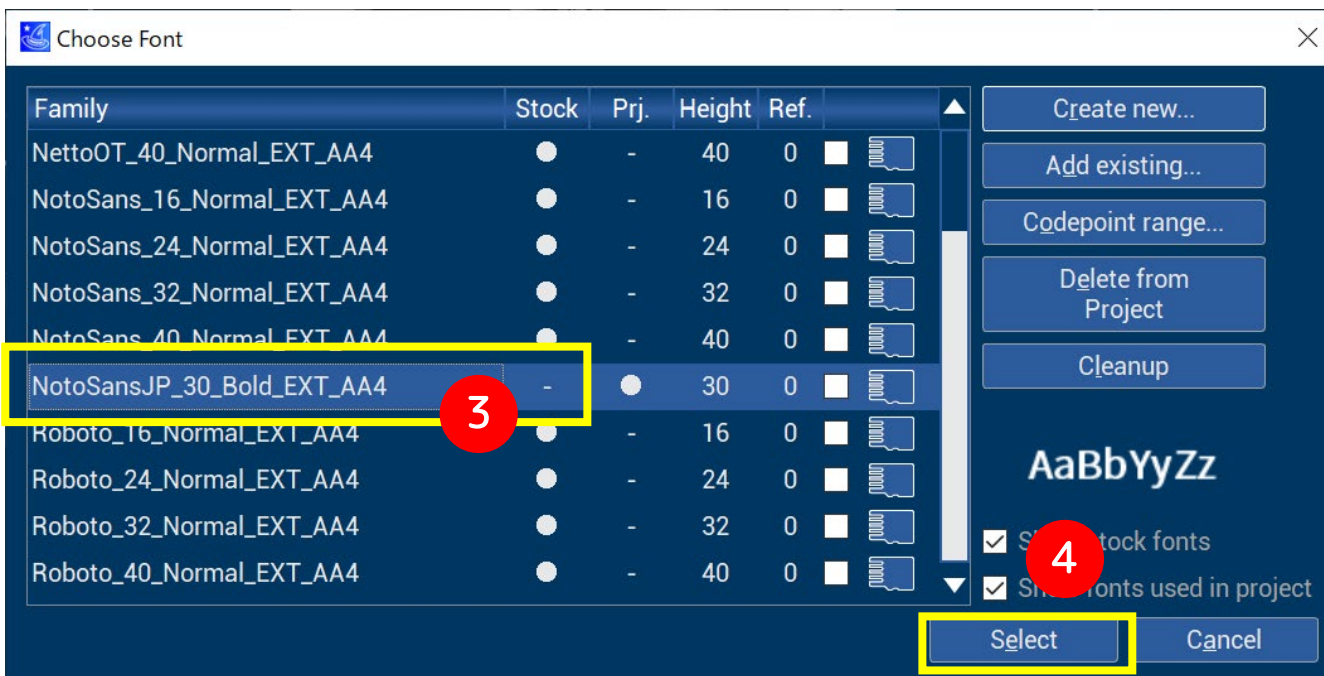


- ① 「ID_SCREEN_01」の「ID_TEXT_00」を選択します。
- ② Propertyウィンドウから「Set Text」アイコンをクリックしてText設定ダイアログを開きます。
- ③ 「Add text」ボタンを押して「ID_RTEXT_1」オブジェクトを追加します。
- ④ 「Nihongo」言語項目下の「…」をマウスダブルクリックして「Flasher：オンボードフラッシュ書き込みツール」文字列を設定します。
- ⑤ 「ID_RTEXT_1」を選択し「Select」ボタンを押してダイアログを閉じます。

画面作成：テキストタイトルの設定（写真2）



- ① 「ID_SCREEN_01」の「ID_TEXT_00」を選択します。
- ② Propertyウィンドウから「Set Font」アイコンをクリックしてFont設定ダイアログを開きます。
- ③ 「NotoSansJP_30_Bold..」フォントを選択し、
- ④ 「Select」ボタンを押してダイアログを閉じます。



画面作成：テキストタイトルの設定（写真2）

File Edit Project Resource Help

Add objects

Screen Box Button Image

Text Slider Rotary Switch

Edit Multiedit Window QRCode

Hierarchic tree

Object	Id
Screen	ID_SCREEN_00
Screen	ID_SCREEN_01
Image	ID_IMAGE_00
Box	ID_BOX_00
Text	ID_TEXT_00
Screen	ID_SCREEN_02
Screen	ID_SCREEN_03
Image	ID_IMAGE_00

Resources

Editor

Flasher ATE

The Embedded Experts

Flasher Portable PLUS

Flasher

Flasher : オンボードフラッシュ書き込みツール

Properties

Top: Width: Left: Right: Bottom: Height:

Set text color

Set background color

Set framed font color

Set text

Set text alignment

Offset x: Offset y: Alignment:

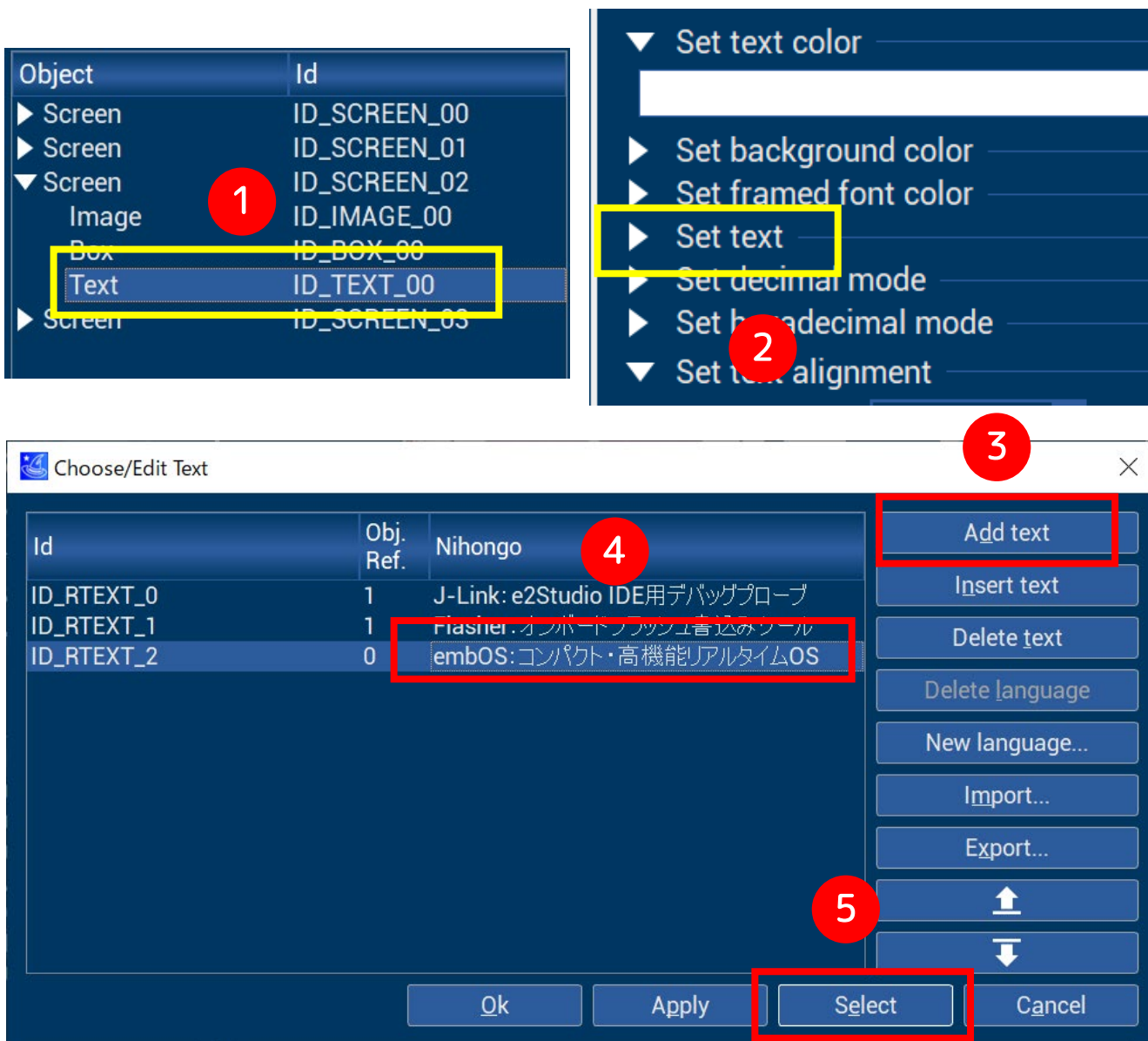
Set font

NotoSansJP_30_Bold_EXT_AA4.xbf

Enable text wrapping

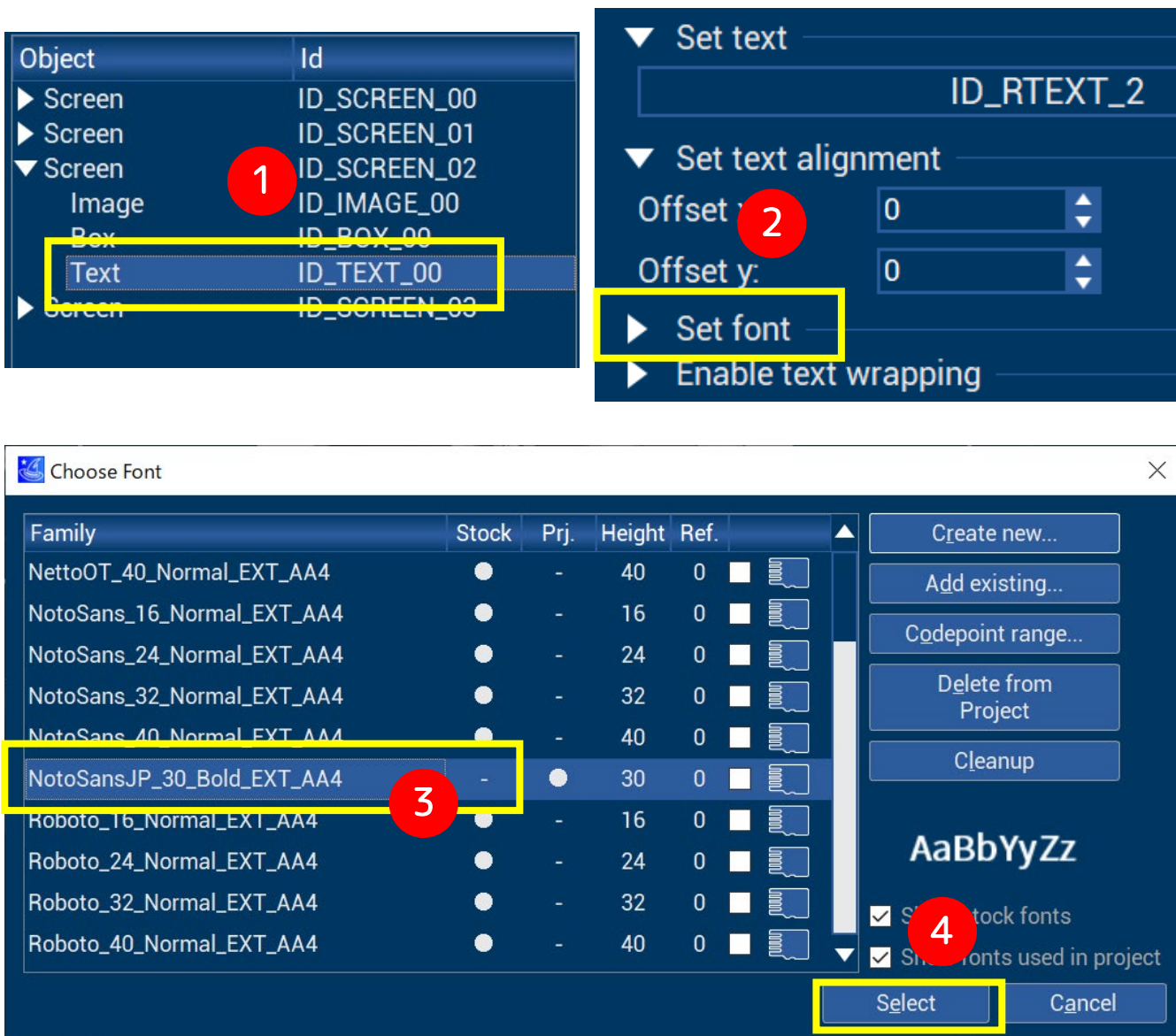
Enable text rotation

画面作成：テキストタイトルの設定（写真3）



- ① 「ID_SCREEN_02」の「ID_TEXT_00」を選択します。
- ② Propertyウィンドウから「Set Text」アイコンをクリックしてText設定ダイアログを開きます。
- ③ 「Add text」ボタンを押して「ID_RTEXT_2」オブジェクトを追加します。
- ④ 「Nihongo」言語項目下の「…」をマウスダブルクリックして「embOS：コンパクト・高機能リアルタイムOS」文字列を設定します。
- ⑤ 「ID_RTEXT_2」を選択し「Select」ボタンを押してダイアログを閉じます。

画面作成：テキストタイトルの設定（写真3）



- ① 「ID_SCREEN_02」の「ID_TEXT_00」を選択します。
- ② Propertyウィンドウから「Set Font」アイコンをクリックしてFont設定ダイアログを開きます。
- ③ 「NotoSansJP_30_Bold..」フォントを選択し、
- ④ 「Select」ボタンを押してダイアログを閉じます。

画面作成：テキストタイトルの設定（写真3）

File Edit Project Resource Help

Add objects

Screen Box Button Image

Text Slider Rotary Switch

Edit Multiedit Window QRCode

Hierarchic tree

Object	Id
Screen	ID_SCREEN_00
Screen	ID_SCREEN_01
Screen	ID_SCREEN_02
Image	ID_IMAGE_00
Box	ID_BOX_00
Text	ID_TEXT_00
Screen	ID_SCREEN_03

Resources

Editor

embOS

embOS : コンパクト・高機能リアルタイムOS

Interactions

Properties

Left: 0 Right: 0 Bottom: 0 Height: 50

Set text color

Set background color

Set framed font color

Set text

Set text alignment

Offset x: 0 Offset y: 0 Alignment:

Set font

NotoSansJP_30_Bold_EXT_AA4.xbf

Enable text wrapping

Enable text rotation

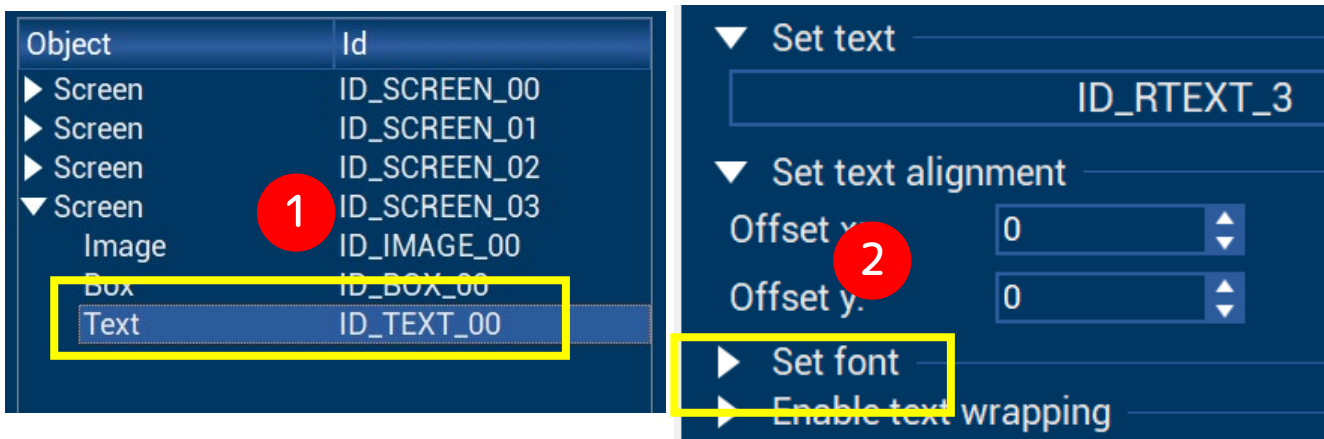
画面作成：テキストタイトルの設定（写真4）

The screenshot shows the AppWizard GUI with the following elements highlighted by numbered red circles:

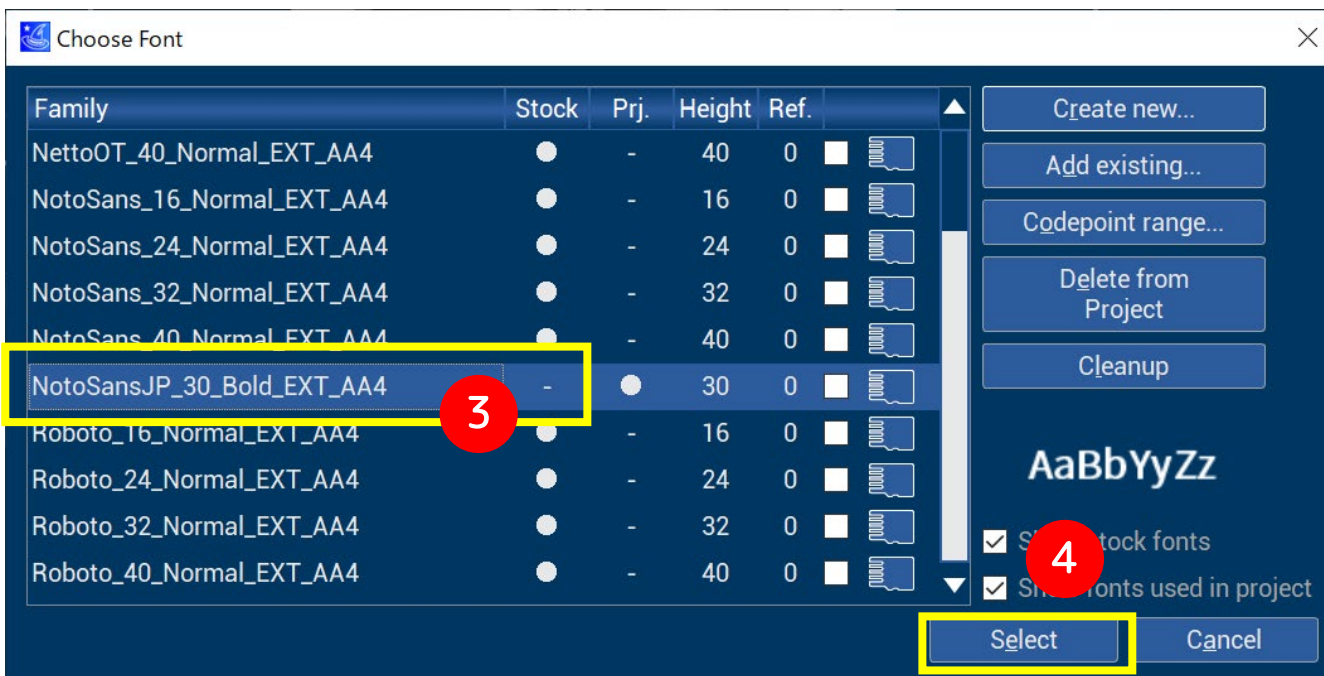
- ① In the Object list, the 'Text' object (ID_TEXT_00) is selected.
- ② In the property window, the 'Set text' option is selected.
- ③ In the 'Choose/Edit Text' dialog, the 'Add text' button is clicked.
- ④ In the 'Choose/Edit Text' dialog, the text 'SEGGER本社(ドイツ:開発センター)' is selected from the list.
- ⑤ In the 'Choose/Edit Text' dialog, the 'Select' button is clicked.

- ① 「ID_SCREEN_03」の「ID_TEXT_00」を選択します。
- ② Propertyウィンドウから「Set Text」アイコンをクリックしてText設定ダイアログを開きます。
- ③ 「Add text」ボタンを押して「ID_RTEXT_3」オブジェクトを追加します。
- ④ 「Nihongo」言語項目下の「…」をマウスダブルクリックして「SEGGER本社（ドイツ：開発センター）」文字列を設定します。
- ⑤ 「ID_RTEXT_3」を選択し「Select」ボタンを押してダイアログを閉じます。

画面作成：テキストタイトルの設定（写真4）



- ① 「ID_SCREEN_03」の「ID_TEXT_00」を選択します。
- ② Propertyウィンドウから「Set Font」アイコンをクリックしてFont設定ダイアログを開きます。
- ③ 「NotoSansJP_30_Bold..」フォントを選択し、
- ④ 「Select」ボタンを押してダイアログを閉じます。



画面作成：テキストタイトルの設定（写真4）

File Edit Project Resource Help

Add objects

Screen Box Button Image

Text Slider Rotary Switch

Edit Multiedit Window QRCode

Hierarchic tree

Object	Id
▶ Screen	ID_SCREEN_00
▶ Screen	ID_SCREEN_01
▶ Screen	ID_SCREEN_02
▼ Screen	ID_SCREEN_03
Image	ID_IMAGE_00
Box	ID_BOX_00
Text	ID_TEXT_00

Resources

Tt

(x)

Editor

1:1

SEGGER

SEGGER本社（ドイツ：開発センター）

Interactions

+/-		Emitter	Signal	Job	Receiver	Comment
+						

Properties

Top: Width: Left: Right: Bottom: Height:

Set text color

Set background color

Set framed font color

Set text

Set text alignment

Offset x: 0 Offset y: 0 Alignment:

Set font

NotoSansJP_30_Bold_EXT_AA4.xbf

Enable text wrapping

Enable text rotation

画面作成：スクリーンズライド設定

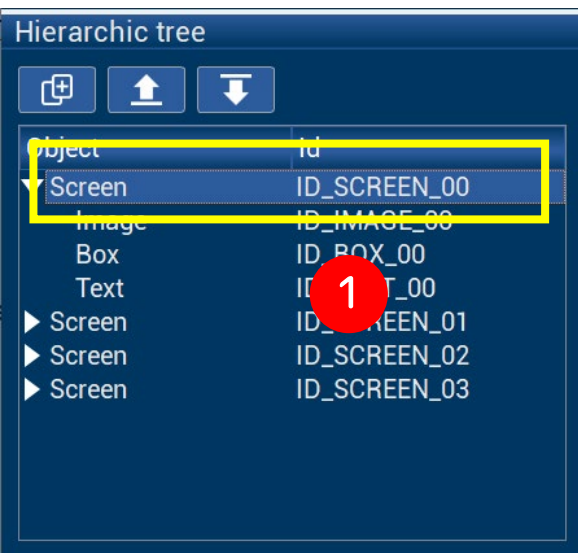


写真1のスワイプモードを設定します。
左側のスワイプに「Photo4」と左側のスワイプに
「Photo2」を設定します。

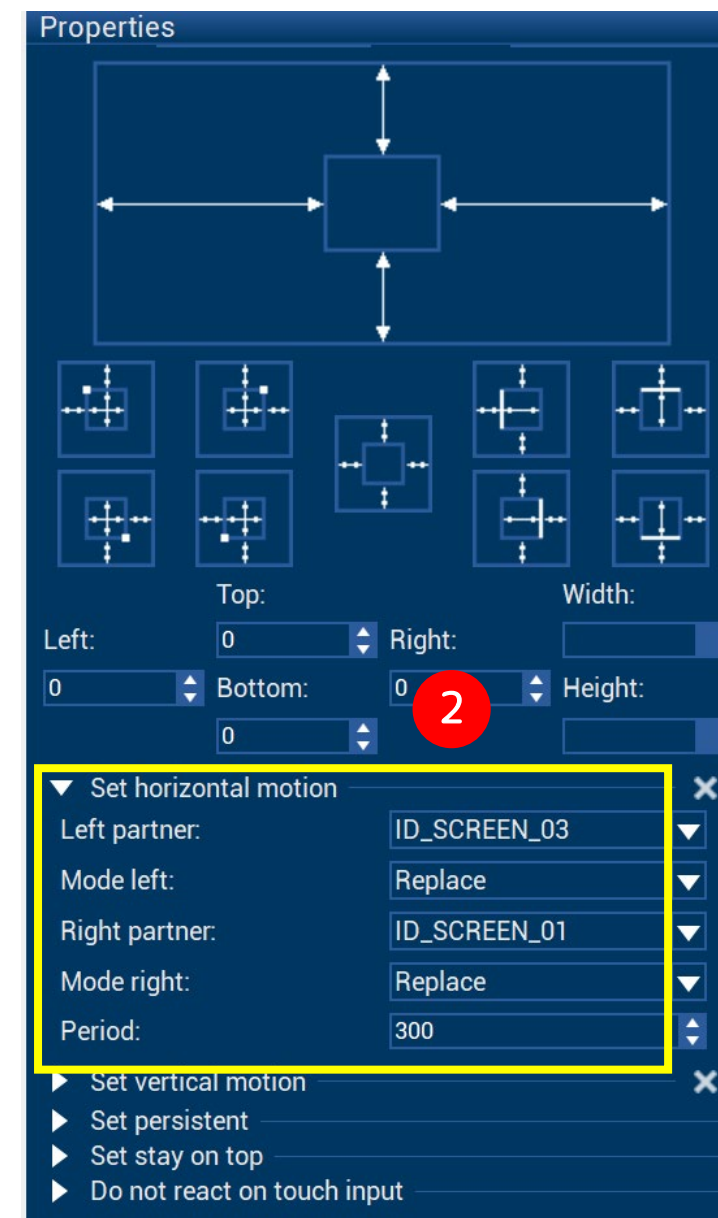
- ① 「ID_SCREEN_00」を選択します。
- ② Propertyウィンドウから「Set Horizontal motion」アイコンをクリックして、
以下のように左と右側のスライドスクリーンを設定します。

Left partnerに「ID_SCREEN_03」を選択します。

Right partnerに「ID_SCREEN_01」を選択します。

Mode left = Replace

Mode right = Replace



画面作成：スクリーンズライド設定

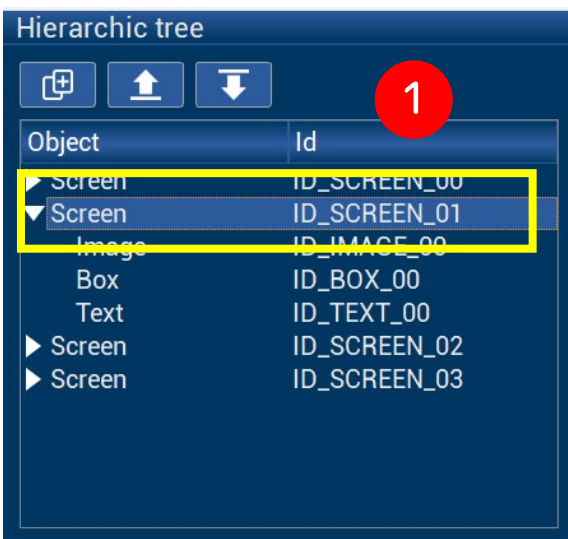


写真2のスワイプモードを設定します。
左側のスワイプに「Photo1」と左側のスワイプに
「Photo3」を設定します。

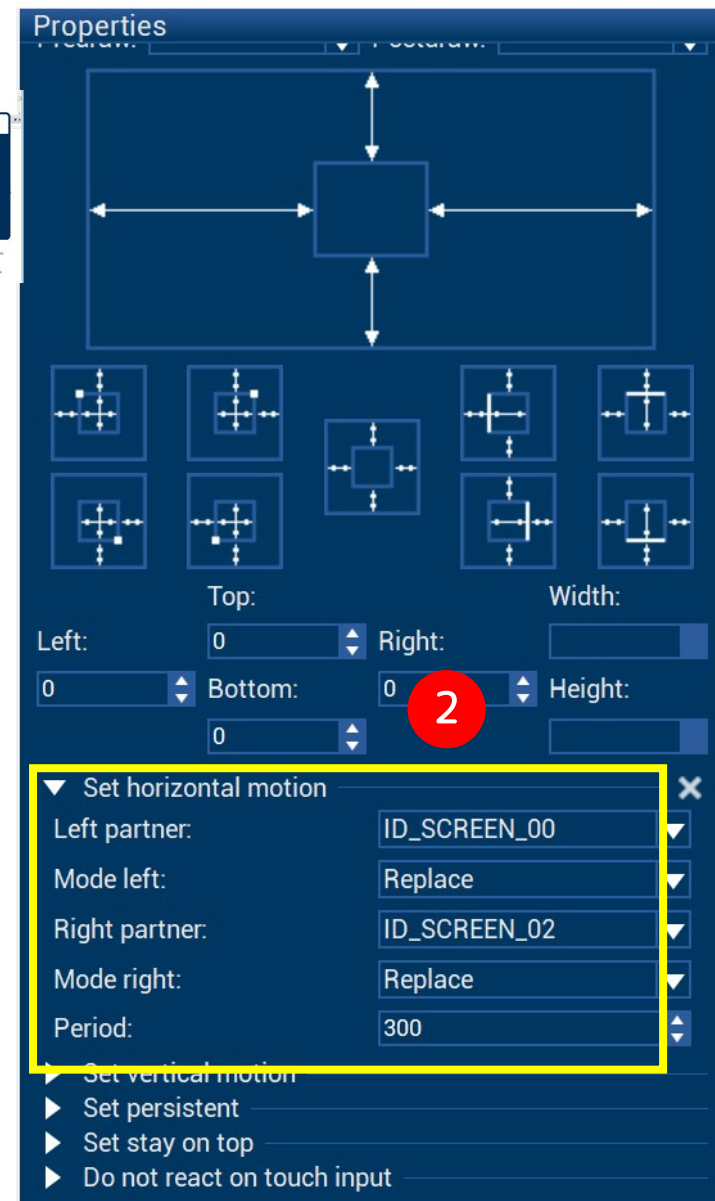
- ① 「ID_SCREEN_01」を選択します。
- ② Propertyウィンドウから「Set Horizontal motion」アイコンをクリックして、
以下のように左と右側のスライドスクリーンを設定します。

Left partnerに「ID_SCREEN_00」を選択します。

Right partnerに「ID_SCREEN_02」を選択します。

Mode left = Replace

Mode right = Replace



画面作成：スクリーンズライド設定

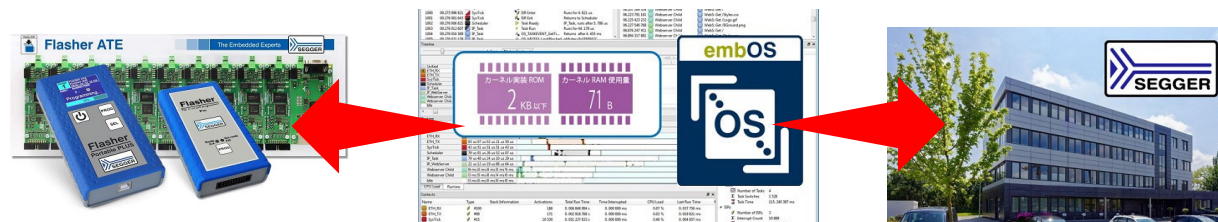
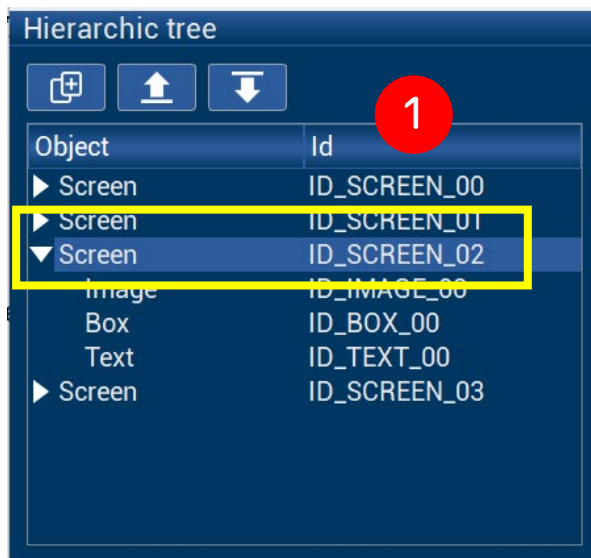


写真3のスワイプモードを設定します。
左側のスワイプに「Photo2」と左側のスワイプに
「Photo4」を設定します。

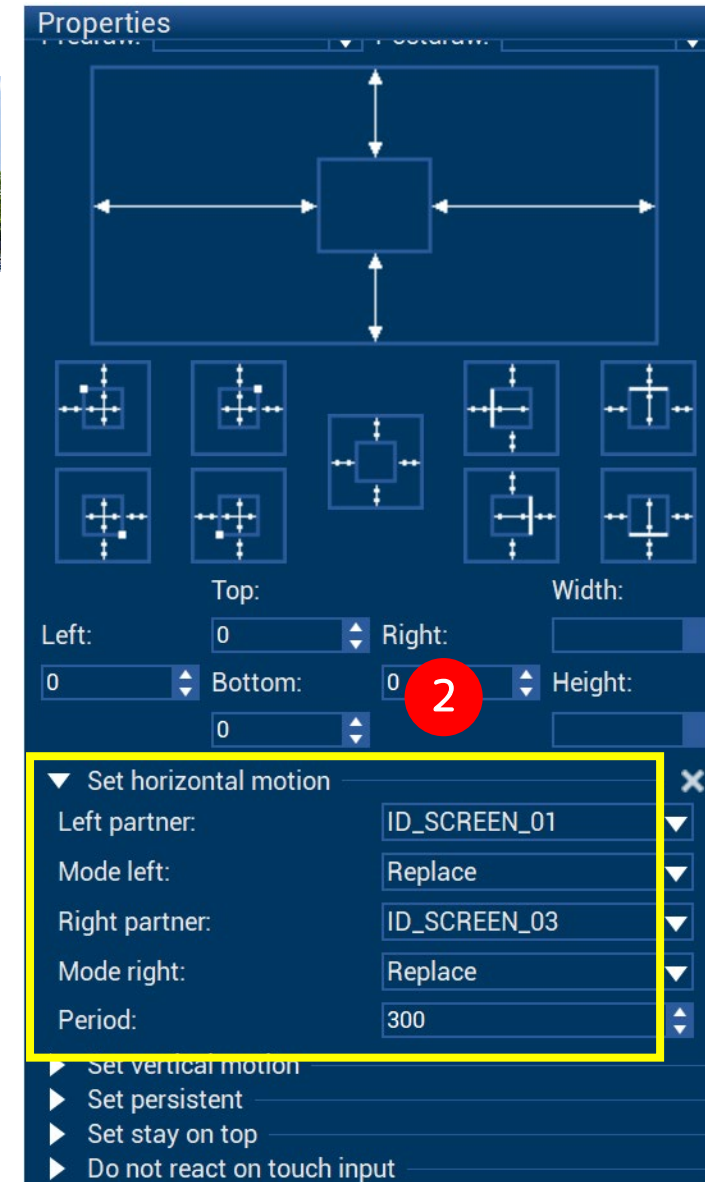
- ① 「ID_SCREEN_02」を選択します。
- ② Propertyウィンドウから「Set Horizontal motion」アイコンをクリックして、
以下のように左と右側のスライドスクリーンを設定します。

Left partnerに「ID_SCREEN_01」を選択します。

Right partnerに「ID_SCREEN_03」を選択します。

Mode left = Replace

Mode right = Replace



画面作成：スクリーンズライド設定

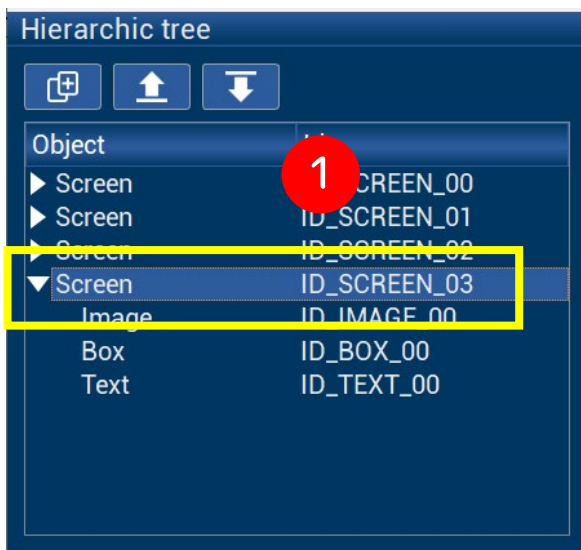


写真4のスワイプモードを設定します。
左側のスワイプに「Photo3」と左側のスワイプに
「Photo1」を設定します。

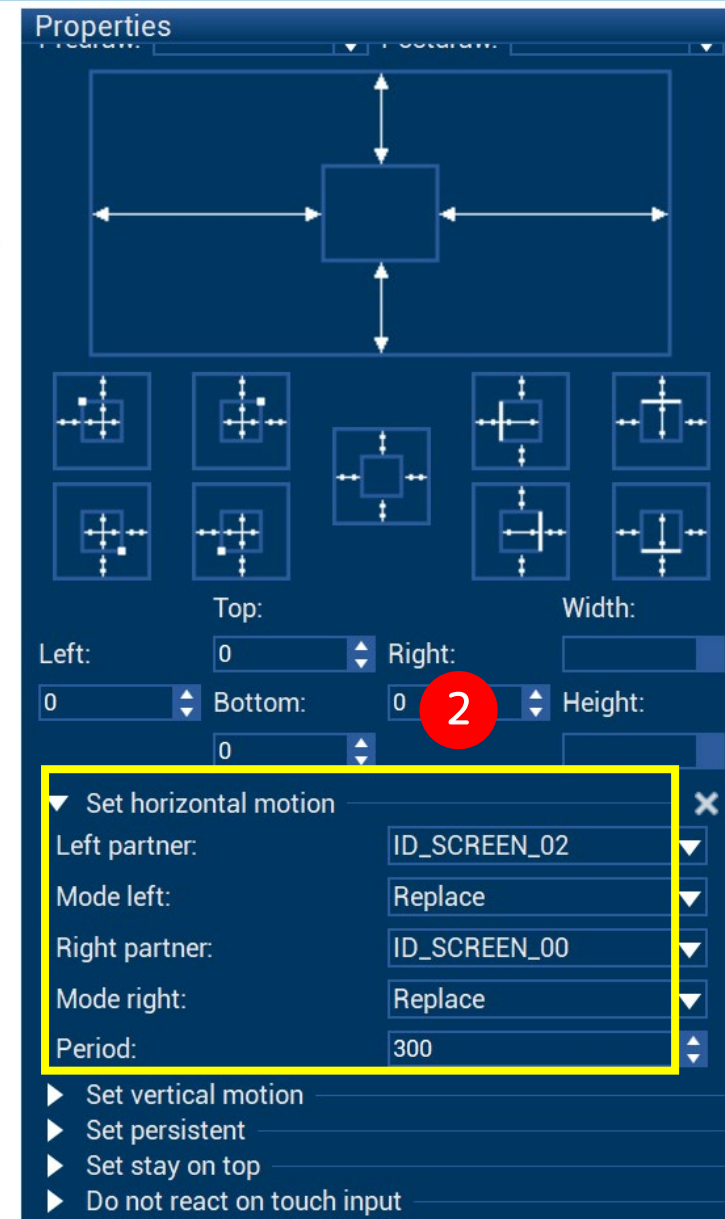
- ① 「ID_SCREEN_03」を選択します。
- ② Propertyウィンドウから「Set Horizontal motion」アイコンをクリックして、
以下のように左と右側のスライドスクリーンを設定します。

Left partnerに「ID_SCREEN_02」を選択します。

Right partnerに「ID_SCREEN_00」を選択します。

Mode left = Replace

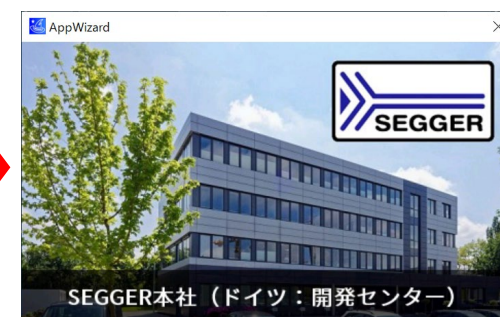
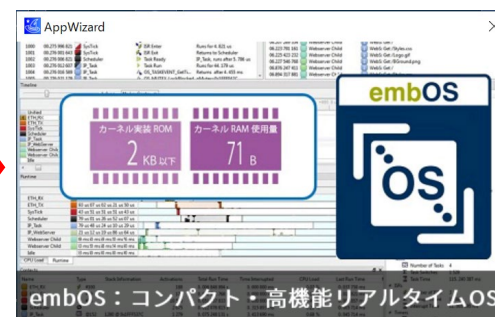
Mode right = Replace



画面作成：スクリーンライド設定の動作確認



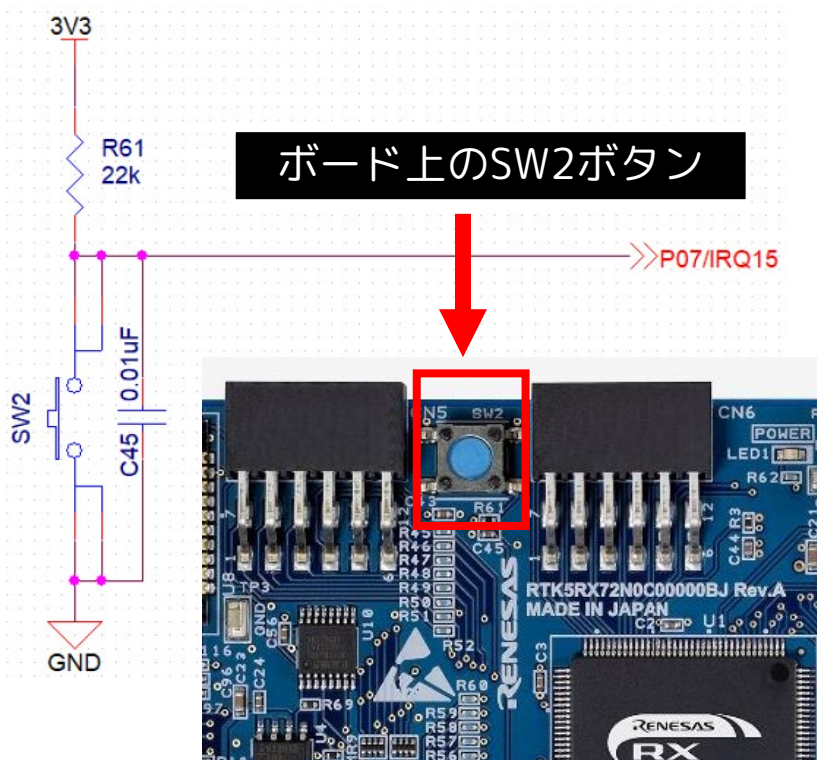
- ① AppWizardのPlayモードシミュレーションでマウススワイプ操作で画面移動（左、右）動作を確認します。
- ② 確認後にPlayモードを終了します。



カスタムイベント設定：周期ハンドラ処理を追加

実装したい処理：

周期ハンドラでオンボードSW2ボタン（GPIOポートP07）のステータスを確認、SW2ボタンを押してAppWizard変数（ID_VAR_00）の値を更新します。



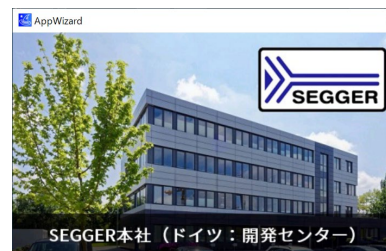
SW2ボタン
イベント



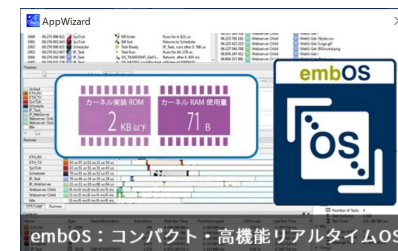
SW2ボタン
イベント



SW2ボタン
イベント

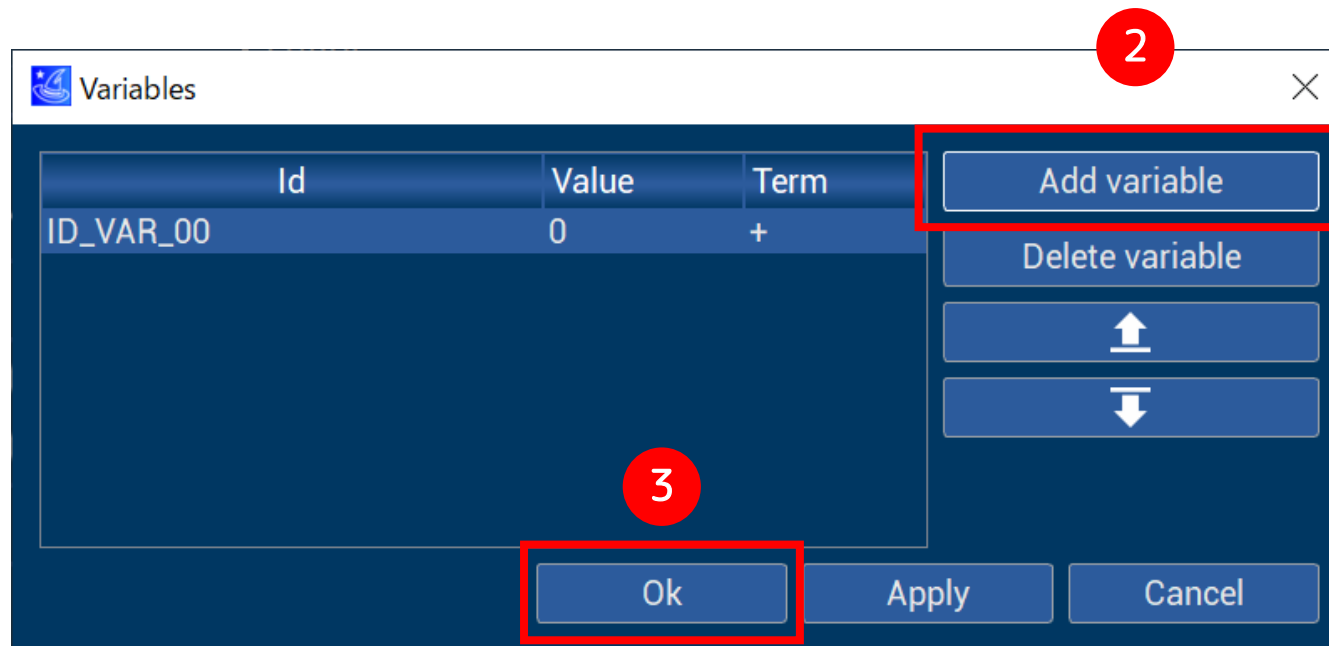
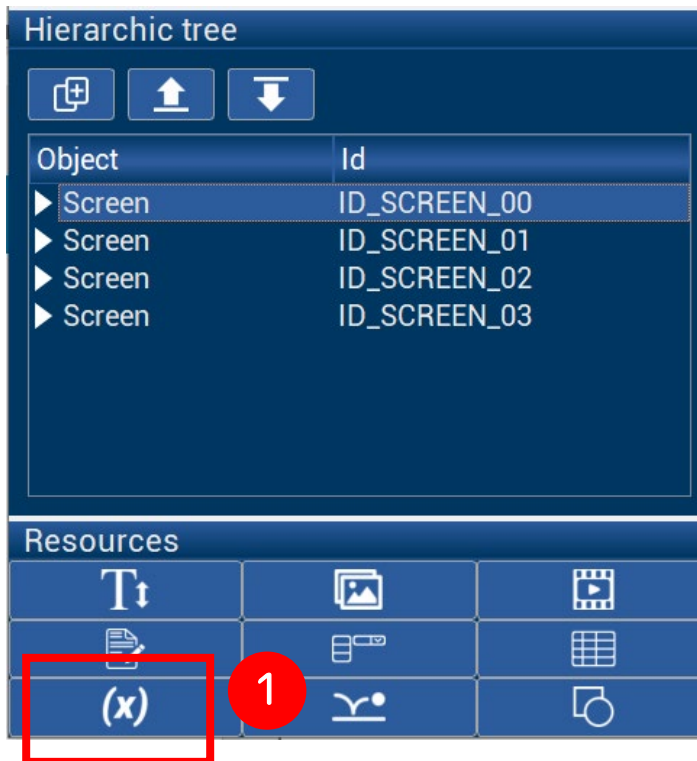


SW2ボタン
イベント



画面イベント設定：変数を作成

AppWizardプロジェクトで画像切り替え処理用変数を作成します。

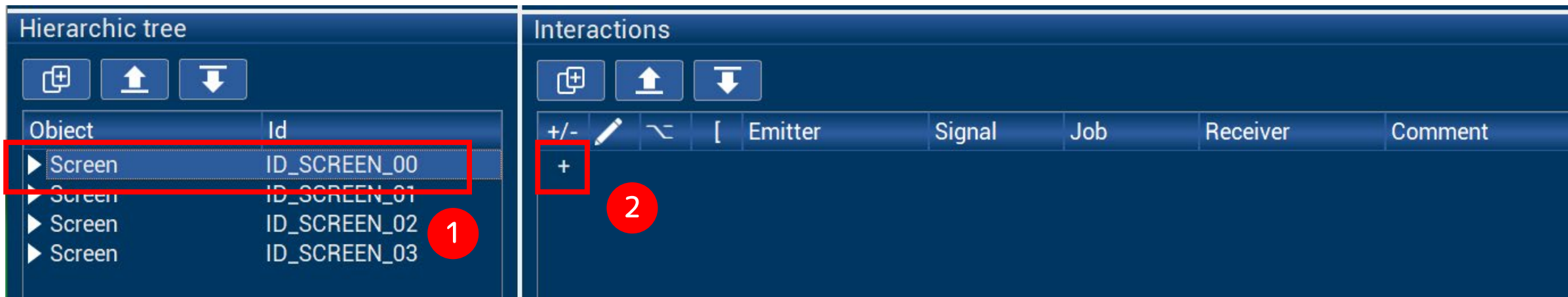


- ① AppWizard画面左下の「Variables」リソースボタンを押して変数設定ダイアログを起動します。
- ② 変数設定ダイアログから「Add variables」ボタンを押して新しい変数（ID_VAR_00）を追加します。
- ③ 「Ok」ボタンを押して設定を保存します。

画面イベント設定：変数のイベント設定

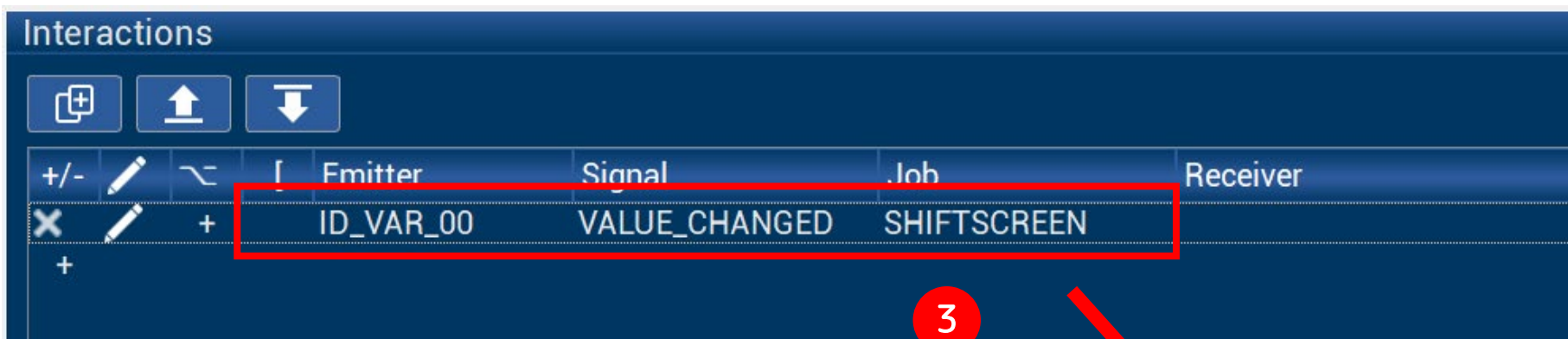
「ID_VAR_00」変数の値が変更されましたScreenオブジェクトに通知イベントが発生されるようにAppWizardのインタラクション（Interactions）・ジョブを作成します。

イベントを送信するオブジェクト（Emitter）	→ 変数（ID_VAR_00）
イベントタイプ（Signal）	→ 値が変更した時（VALUE_CHANGED）
実行処理（Job）	→ スクリーン移動（SHIFTSCREEN）
イベントを受信するオブジェクト（Receiver）	→ スクリーンオブジェクト



- ① オブジェクトウィンドウから「ID_SCREEN_00」を選択します。
- ② 「Interactions」ウィンドウから「+」アイコンボタンをクリックして新しいジョブイベントを追加します。

画面イベント設定：変数のイベント設定（写真1）



③ ID_VAR_00 変数の値が変更されたイベントにSHIFTSCREENインタラクションコマンドを設定します。

Emitter = ID_VAR_00, Signal = VALUE_CHANGED,
Job = SHIFTSCREEN

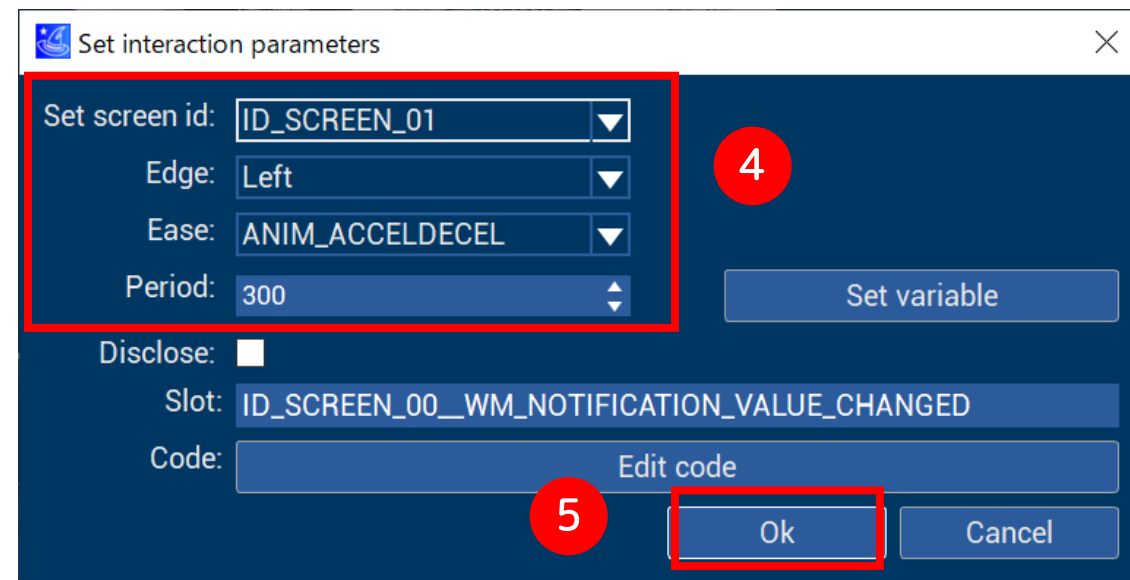
④ インタラクションパラメータ設定：

Set screen id: ID_SCREEN_01

Edge: Left

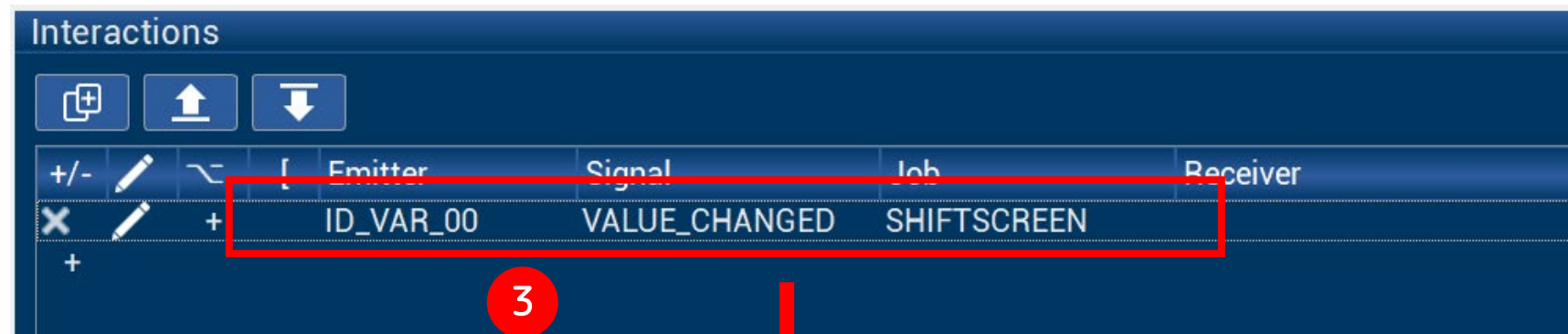
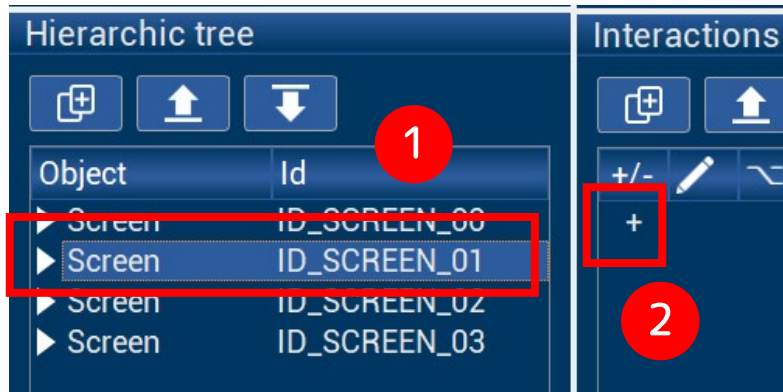
Ease: ANIM_ACCELDECEL

Period: 300



⑤ 「Ok」ボタンを押して設定を保存します。

画面イベント設定：変数のイベント設定（写真2）



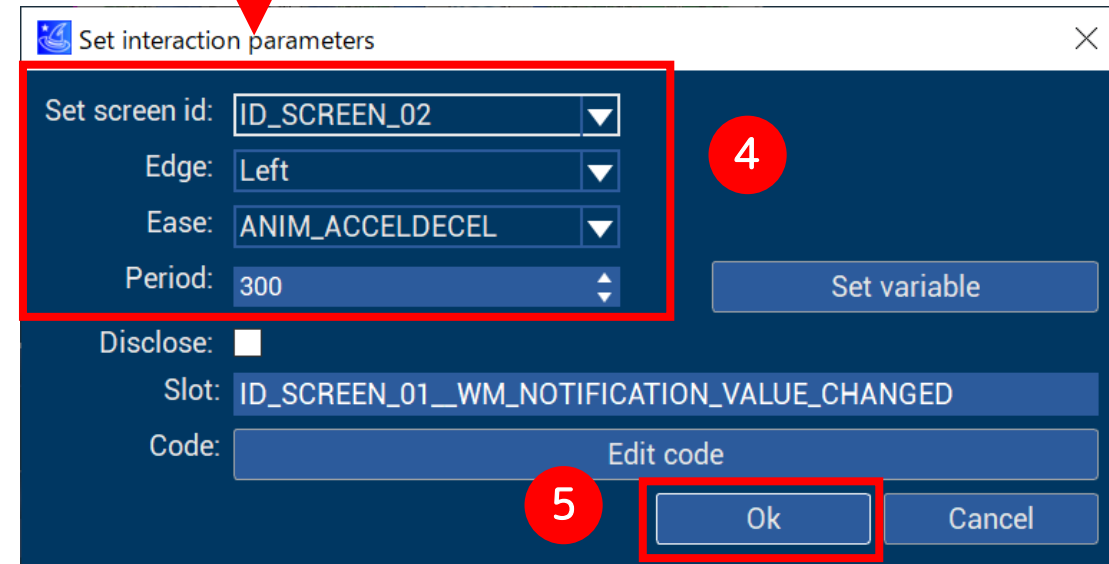
① オブジェクトウィンドウから「ID_SCREEN_01」を選択します。

② 「Interactions」ウィンドウから「+」アイコンボタンをクリックして新しいジョブイベントを追加します。

③ ID_VAR_00 変数の値が変更されたイベントにSHIFTSCREENインタラクションコマンドを設定します。
Emitter = ID_VAR_00, Signal = VALUE_CHANGED,
Job = SHIFTSCREEN

④ インタラクションパラメータ設定：
Set screen id: ID_SCREEN_02 Edge: Left
Ease: ANIM_ACCELDECCEL Period: 300

⑤ 「Ok」ボタンを押して設定を保存します。



画面イベント設定：変数のイベント設定（写真3）



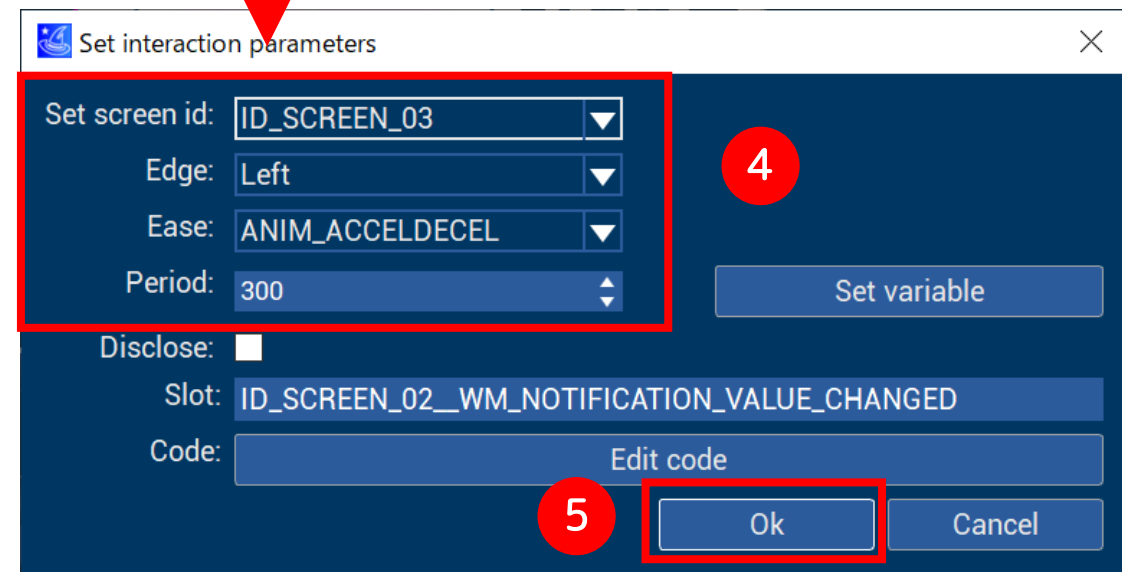
① オブジェクトウィンドウから「ID_SCREEN_02」を選択します。

② 「Interactions」ウィンドウから「+」アイコンボタンをクリックして新しいジョブイベントを追加します。

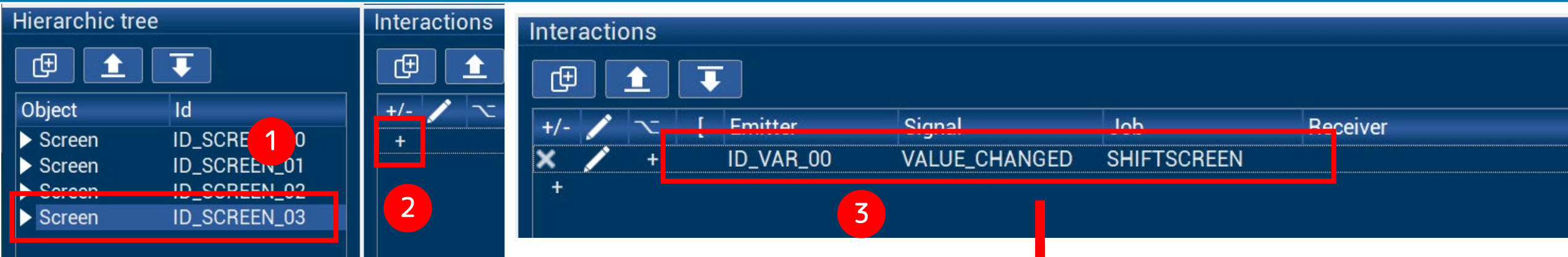
③ ID_VAR_00 変数の値が変更されたイベントにSHIFTSCREENインタラクションコマンドを設定します。
Emitter = ID_VAR_00, Signal = VALUE_CHANGED,
Job = SHIFTSCREEN

④ インタラクションパラメータ設定：
Set screen id: ID_SCREEN_03 Edge: Left
Ease: ANIM_ACCELDECEL Period: 300

⑤ 「Ok」ボタンを押して設定を保存します。



画面イベント設定：変数のイベント設定（写真4）



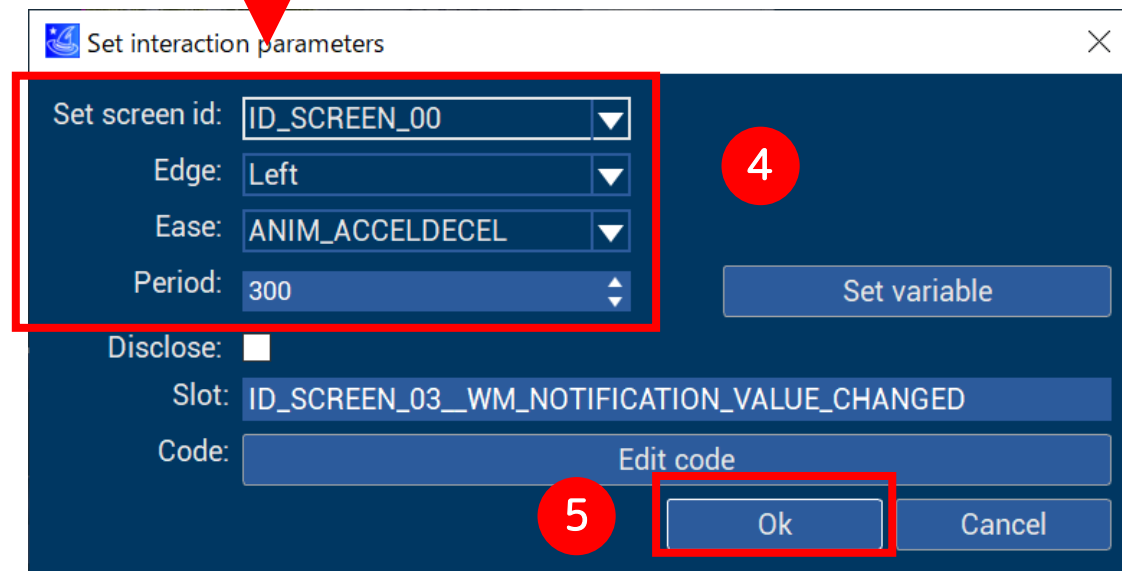
① オブジェクトウィンドウから「ID_SCREEN_03」を選択します。

② 「Interactions」ウィンドウから「+」アイコンボタンをクリックして新しいジョブイベントを追加します。

③ ID_VAR_00 変数の値が変更されたイベントにSHIFTSCREENインタラクションコマンドを設定します。
Emitter = ID_VAR_00, Signal = VALUE_CHANGED,
Job = SHIFTSCREEN

④ インタラクションパラメータ設定：
Set screen id: ID_SCREEN_00 Edge: Left
Ease: ANIM_ACCELDECCEL Period: 300

⑤ 「Ok」ボタンを押して設定を保存します。



画面デザインのソースコードを作成

The screenshot shows the e2Studio interface. On the left, the 'File' menu is open, and 'Export & Save' is highlighted with a red box and a red circle with the number '1'. A blue arrow points from this menu item to the 'Generated' folder in the project explorer on the right. The project explorer shows a tree structure with folders like 'CustomCode' and 'Generated'. The 'Generated' folder is expanded, showing files like 'APPWConf.c', 'ID_SCREEN_00.c', etc. The code editor on the right shows the content of 'APPW_MainTask.c', which includes a 'MainTask' function. To the right of the code editor, the text 'emWinスタートアップ 起動処理' is displayed.

File Edit Project Resource Help

- New project...
- Open... Ctrl+O
- Save Ctrl+S
- Save As... Ctrl+Alt+S
- Rename...
- Close project
- Export & Save Ctrl+Shift+E**
- Export external data Ctrl+D
- Open example...
- Import BSP...
- Exit
- Recent files..

プロジェクト・エクスプローラー

- QEDisp_RX72N_Envision
 - バイナリー
 - Includes
 - aw/Resource
 - aw/Source
 - CustomCode
 - Application.c
 - Application.h
 - ID_SCREEN_00_Slots.c
 - ID_SCREEN_01_Slots.c
 - ID_SCREEN_02_Slots.c
 - ID_SCREEN_03_Slots.c
 - Generated
 - APPWConf.c
 - ID_SCREEN_00.c
 - ID_SCREEN_00.h
 - ID_SCREEN_01.c
 - ID_SCREEN_01.h
 - ID_SCREEN_02.c
 - ID_SCREEN_02.h
 - ID_SCREEN_03.c
 - ID_SCREEN_03.h
 - Resource.c
 - Resource.h
 - APPW_MainTask.c

```
#include "Generated/Resource.h"

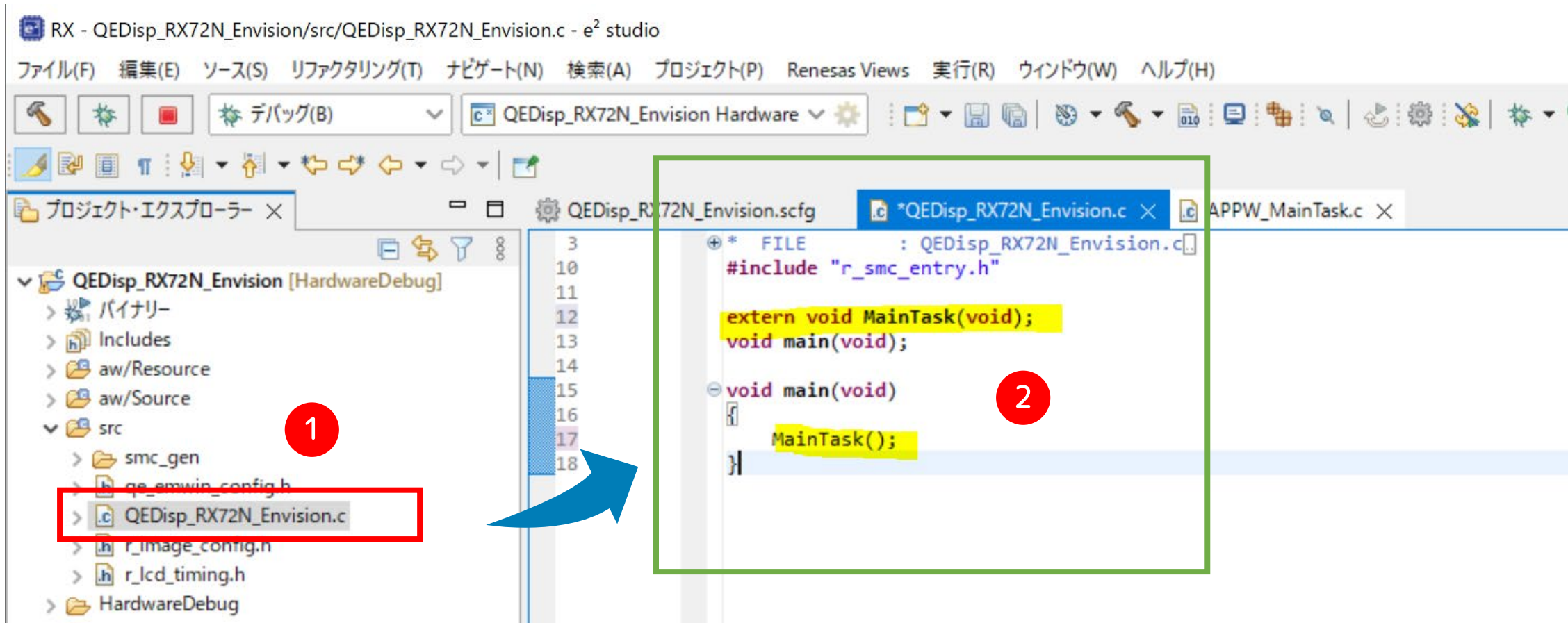
void MainTask(void) {
    //
    // Setup configuration dependent pointers
    //
    APPW_X_Setup();
    //
    // Initialize AppWizard
    //
    APPW_Init(APPW_PROJECT_PATH);
    //
    // Create all persistent screens except initial screen
    //
    APPW_CreatePersistentScreens();
    //
    // Create initial screen...
    //
    APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);
    //
    // ...and keep it alive
    //
    while (1) {
        while (GUI_Exec1()) {
            APPW_Exec();
        }
        APPW_Exec();
        GUI_X_Delay(5);
    }
}

/***** End of file *****/
```

emWinスタートアップ
起動処理

① AppWizardの「File → Export & Save」メニューコマンドでGUIコード（リソース、アプリケーション）を作成します。e2Studio側に「aw」フォルダ下にコードが表示されます。

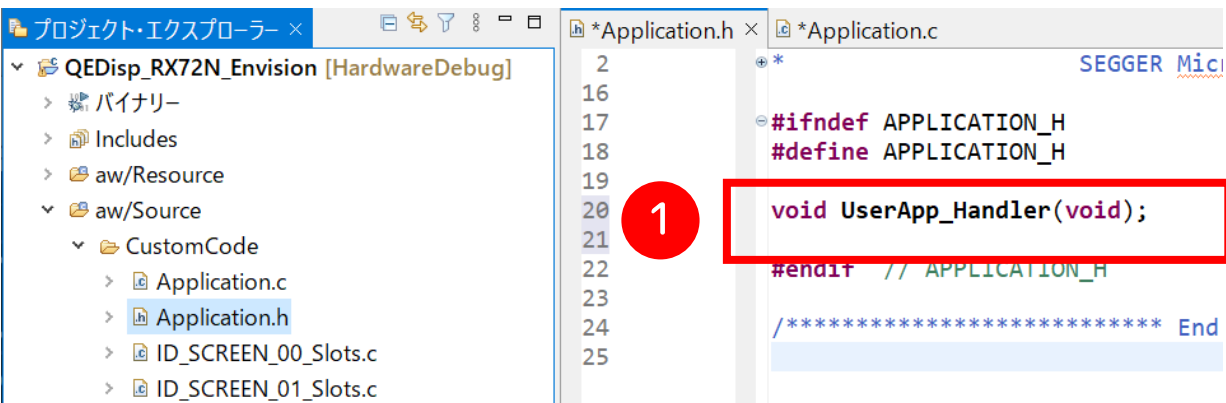
スタートアップコードを作成



- ① AppWizardのプロジェクトエクスプローラーウィンドウから「QEDisp_RX72N_Envision.c」スタートアップエントリファイルを開きます。
- ② 「QEDisp_RX72N_Envision.c」ファイルにemWinスタートアップエントリ関数 MainTask() を追加します。

カスタムイベント設定：周期ハンドラを登録

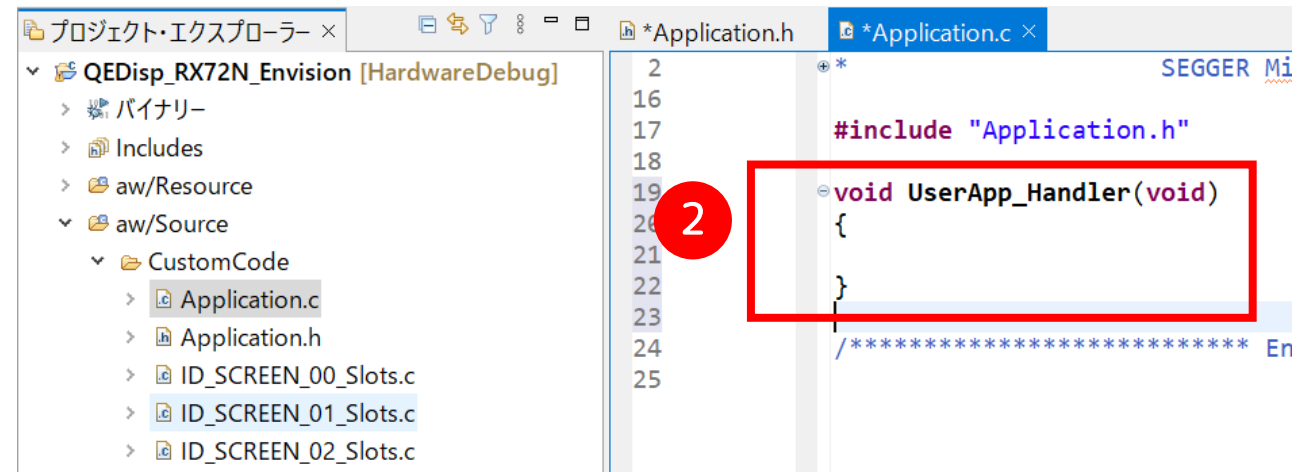
カスタム周期ハンドラ関数UserApp_Handler()の実態を「Application.c/h」ファイルに定義します。



¥aw¥Source¥CustomCode フォルダ下の Application.h ファイルに以下の関数宣言（赤い色コード）を追加します。

```
#ifndef APPLICATION_H  
#define APPLICATION_H
```

```
void UserApp_Handler(void);
```



¥aw¥Source¥CustomCode フォルダ下の Application.c ファイルに以下の関数定義を追加します。

```
#include "Application.h"
```

```
void UserApp_Handler(void)  
{  
  
}
```


カスタムイベント設定：SW2ボタンのクリックイベント処理を追加

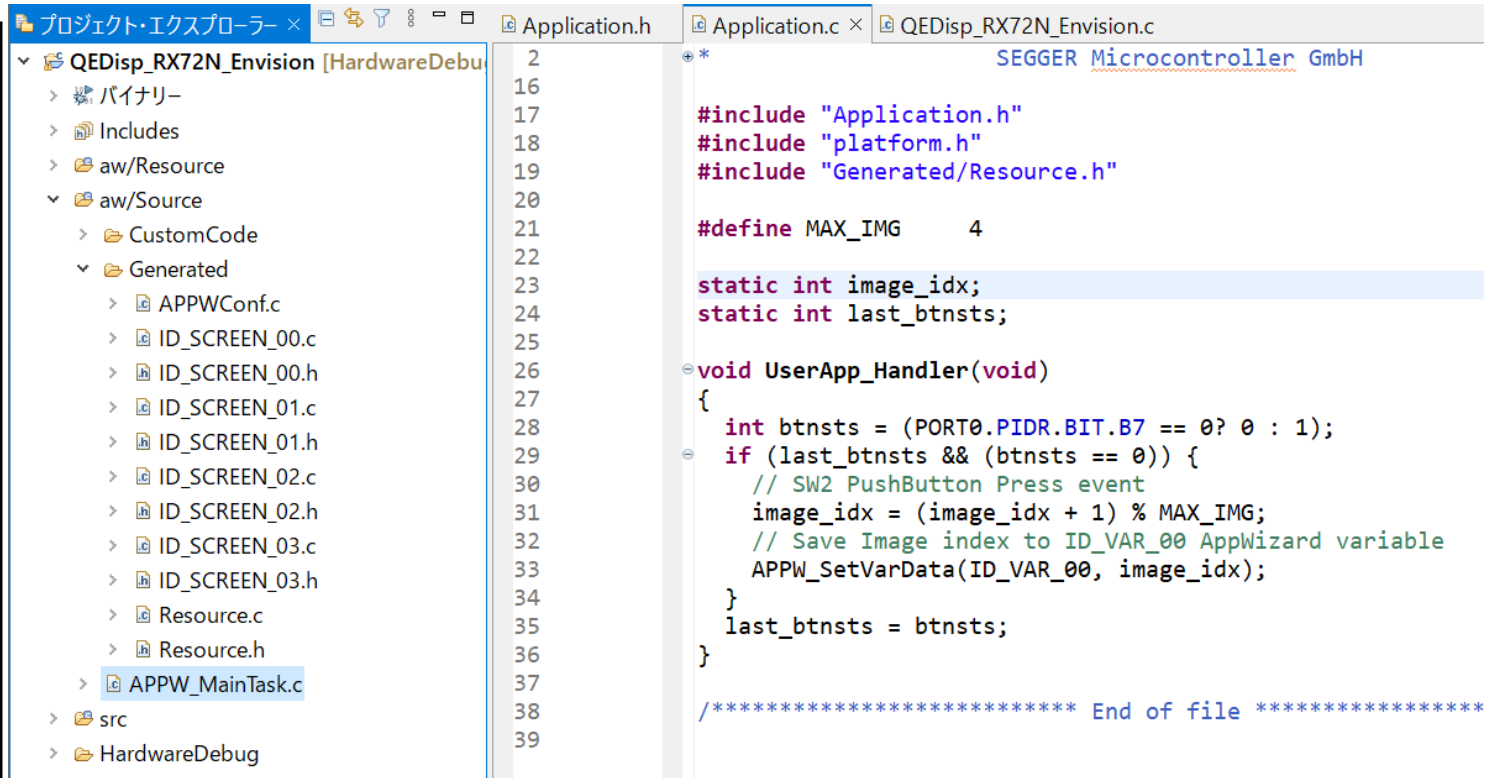
ボタンハンドラ処理コード（以下の赤い色コード）を「Application.c」ファイルに追加します。

```
#include "Application.h"
#include "platform.h"
#include "Generated/Resource.h"

#define MAX_IMG 4

static int image_idx;
static int last_btnsts;

void UserApp_Handler(void)
{
    int btnsts = (PORT0.PIDR.BIT.B7 == 0? 0 : 1);
    if (last_btnsts && (btnsts == 0)) {
        // SW2 PushButton Press event
        image_idx = (image_idx + 1) % MAX_IMG;
        // Save Image index to ID_VAR_00 AppWizard
        variable
        APPW_SetVarData(ID_VAR_00, image_idx);
    }
    last_btnsts = btnsts;
}
```



The screenshot shows the IDE interface. On the left, the 'プロジェクト・エクスプローラー' (Project Explorer) displays the project structure for 'QEDisp_RX72N_Envision'. The 'Generated' folder is expanded, showing files like 'APPWConf.c', 'ID_SCREEN_00.c', 'ID_SCREEN_01.c', 'ID_SCREEN_02.c', 'ID_SCREEN_03.c', 'Resource.c', and 'Resource.h'. The 'APPW_MainTask.c' file is selected. On the right, the 'Application.c' file is open, showing the code for the 'UserApp_Handler' function. The code includes headers, defines 'MAX_IMG' as 4, and declares static variables 'image_idx' and 'last_btnsts'. The function 'UserApp_Handler' is defined, which checks for a button press event and updates the image index and last button state. The code is color-coded: includes are blue, defines are purple, static variables are blue, and the function body is black. The function body includes comments in green and red. The red comments correspond to the code in the left box.

```
SEGGER Microcontroller GmbH
#include "Application.h"
#include "platform.h"
#include "Generated/Resource.h"

#define MAX_IMG 4

static int image_idx;
static int last_btnsts;

void UserApp_Handler(void)
{
    int btnsts = (PORT0.PIDR.BIT.B7 == 0? 0 : 1);
    if (last_btnsts && (btnsts == 0)) {
        // SW2 PushButton Press event
        image_idx = (image_idx + 1) % MAX_IMG;
        // Save Image index to ID_VAR_00 AppWizard variable
        APPW_SetVarData(ID_VAR_00, image_idx);
    }
    last_btnsts = btnsts;
}

/***** End of file *****/
```

APPW_SetCustCallback()関数APIでカスタム周期ハンドラをemWinに登録します。

src フォルダ下の QEDisp_RX72N_Envision.cファイルに赤色文字のコードを追加します。

```
#include "r_smc_entry.h"
#include "AppWizard.h"
#include "CustomCode/Application.h"

extern void MainTask(void);

void main(void) {
    // your codes...

    // Set custom periodic handler
    APPW_SetCustCallback(UserApp_Handler);
    MainTask();

    // your codes...
}
```

APPW_SetCustCallback()

ユーザマニュアル

Description

Sets a function pointer for a function which is executed at the end of `APPW_Exec()`.

Prototype

```
void APPW_SetCustCallback(void ( *pFunc) ());
```

Parameters

Parameter	Description
pFunc	Pointer to the function which should be called.

Additional information

This function allows the user to set a function pointer which is being called from `APPW_Exec()`. This allows the user to execute his own code periodically.

TASK-3

ビルド & 動作確認

e2Studioプロジェクト環境でBSP+UIデザインアプリケーションコードをビルドして、RX72N Envisionボード環境で動作を確認します。



ビルド & 動作確認：プロジェクトのビルド

e2Studio-v2310 - QEDisp_RX72N_Envision/src/QEDisp_RX72N_Envision.c - e2 studio

ファイル(F) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A) プロジェクト(P) Renesas Views 実行(R) Renesas AI ウィンドウ(W) ヘルプ(H)

プロジェクトを開く(E)
プロジェクトを閉じる(S)
すべてビルド(A) Ctrl+Alt+B
ビルド構成
プロジェクトのビルド(B) Ctrl+B
ランタイムセットのビルド(W)
クリーン(N)...
自動的にビルド(M)
ビルド・ターゲット
C/C++ インデックス(i)
すべての依存関係を更新 Alt+D
Change Device
Change Toolchain Version
C/C++ Project Settings Ctrl+Alt+P
プロパティ(P)

1

2

CDT ビルド・コンソール [QEDisp_RX72N_Envision]
Scanning and building file: .../src/QEDisp_RX72N_Envision.c
Invoking: Linker
Building target:
Renesas Optimizing Linker Completed
Finished building target:
Loading input file QEDisp_RX72N_Envision.abs
Parsing the ELF input file....
49 segments required LMA fixes
Converting the DWARF information....
Constructing the output ELF image....
Saving the ELF output file QEDisp_RX72N_Envision.x
Build complete.
10:14:28 Build Finished. 0 errors, 0 warnings. (took 2s.231ms)

①プロジェクトを選択して、
②ビルドコマンドを実行します。

更新が利用できます
ソフトウェアで更新が利用できます。
更新をレビューしてインストールします。

ビルド & 動作確認：評価ボード環境でデバッグ及び動作確認

デバッグ操作を開始します。

ボード上のSW2ボタンを押して差審切り替えを確認します

更新をレビューしてインストールします。

- RX72N Envision Kitの詳細

<https://www.renesas.com/jp/ja/products/software-tools/boards-and-kits/eval-kits/rx72n-envision-kit.html>

- RX65N / RX72N Envision Kit用emWinスタートアップガイド (PDF)

https://www.embitek.co.jp/technote/emwin/StartupGuide_emWin_EnvisionKit.pdf

- RXファミリ QE for Display [RX] アプリケーションノート

<https://www.renesas.com/jp/ja/document/apn/rx-family-qe-display-rx-application-note>

- AppWizardユーザマニュアル

https://www.segger.com/doc/UM03003_AppWizard.html

- emWinユーザマニュアル

https://www.segger.com/doc/UM03001_emWin.html

ソースパッケージアップグレードのご案内

OEM版emWinライブラリパッケージをご利用のユーザー様がソース付きのemWin PROパッケージへ通常価格の **割引** 価格でアップグレード可能です。

※ ライセンス製品価格・割引率は、サポート窓口対応や諸条件、為替相場により提示価格が変動する場合があります。

ソースアップグレードのメリット：

- ソース付きプロジェクトでGUIアプリケーションのデバッグ操作が簡単
- ソースを参考に新しいウィジェット作成可能
- emWinライブラリの最適化はさらにチューニング可能
- お困りの際に技術サポートへの問い合わせ可能

お気軽に以下窓口へお問い合わせください。

株式会社エンビテック

代理店販売（デバッグ、RTOS、GUI、ミドルウェア）

TEL: 03-6240-2655 / FAX: 03-6240-2656

E-mail : sales@embitek.co.jp

<https://www.embitek.co.jp>

