

AppWizard

Wizard for creating ready-to-use emWin applications

Quick Start Guide

Document: AN03003

Revision: 1

Date: July 2, 2020



A product of SEGGER Microcontroller GmbH

www.segger.com

Disclaimer

Specifications written in this document are believed to be accurate, but are not guaranteed to be entirely free of error. The information in this manual is subject to change for functional or performance improvements without notice. Please make sure your manual is the latest edition. While the information herein is assumed to be accurate, SEGGER Microcontroller GmbH (SEGGER) assumes no responsibility for any errors or omissions. SEGGER makes and you receive no warranties or conditions, express, implied, statutory or in any communication with you. SEGGER specifically disclaims any implied warranty of merchantability or fitness for a particular purpose.

Copyright notice

You may not extract portions of this manual or modify the PDF file in any way without the prior written permission of SEGGER. The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such a license.

© 2020 SEGGER Microcontroller GmbH, Monheim am Rhein / Germany

Trademarks

Names mentioned in this manual may be trademarks of their respective companies.

Brand and product names are trademarks or registered trademarks of their respective holders.

Contact address

SEGGER Microcontroller GmbH

Ecolab-Allee 5
D-40789 Monheim am Rhein

Germany

Tel. +49 2173-99312-0
Fax. +49 2173-99312-28
E-mail: support@segger.com*
Internet: www.segger.com

*By sending us an email your (personal) data will automatically be processed. For further information please refer to our privacy policy which is available at <https://www.segger.com/legal/privacy-policy/>.

Manual versions

This manual describes the current software version. If you find an error in the manual or a problem in the software, please inform us and we will try to assist you as soon as possible. Contact us for further information on topics or functions that are not yet documented.

Print date: July 2, 2020

Revision	Date	By	Description
1	200225	FO	Chapter 'Simple operations' added.
0	200102	FO	Initial version.

About this document

Assumptions

This document assumes that you already have a solid knowledge of the following:

- The software tools used for building your application (assembler, linker, C compiler).
- The C programming language.
- The target processor.
- DOS command line.

If you feel that your knowledge of C is not sufficient, we recommend *The C Programming Language* by Kernighan and Richie (ISBN 0--13--1103628), which describes the standard in C programming and, in newer editions, also covers the ANSI C standard.

How to use this manual

This manual explains all the functions and macros that the product offers. It assumes you have a working knowledge of the C language. Knowledge of assembly programming is not required.

Typographic conventions for syntax

This manual uses the following typographic conventions:

Style	Used for
Body	Body text.
Parameter	Parameters in API functions.
Sample	Sample code in program examples.
Sample comment	Comments in program examples.
User Input	Text entered at the keyboard by a user in a session transcript.
Secret Input	Text entered at the keyboard by a user, but not echoed (e.g. password entry), in a session transcript.
Reference	Reference to chapters, sections, tables and figures or other documents.
Emphasis	Very important sections.

Table of contents

1	Introduction	8
1.1	Purpose of this document	9
1.2	What is the AppWizard?	9
2	Getting started	10
2.1	Installation	11
3	Simple operations	12
3.1	Creating a new AppWizard project	13
3.2	Adding a Screen	14
3.3	Adding a Box	15
3.4	Adding a Button	16
3.5	Adding Interactions	17
3.6	Adding a Rotary	19
3.7	Adding a Screen transition	20
3.8	Adding a Slider	21
3.9	Adding Text	22
3.10	Adding an Edit object	24
3.11	Adding a Switch	25
4	The first project	26
4.1	Example: Simple counter application	27
4.1.1	Step 1: Create new project	28
4.1.2	Step 2: Add objects to the screen	28
4.1.3	Step 3: Adjust object properties	29
4.1.4	Step 4: Define the application's behavior	31
4.1.5	Step 5: Run the application	31
4.1.6	Step 6: Export the project	32
4.1.7	Step 7: Run on target	32

Chapter 1

Introduction

A short introduction about the AppWizard and the purpose of this guide.

1.1 Purpose of this document

The purpose of this document is to assist new users of the AppWizard to create and run their first project. Since this is a Quick Start Guide, it does not serve as a reference manual, nor does it document all of the AppWizard's features.

Note

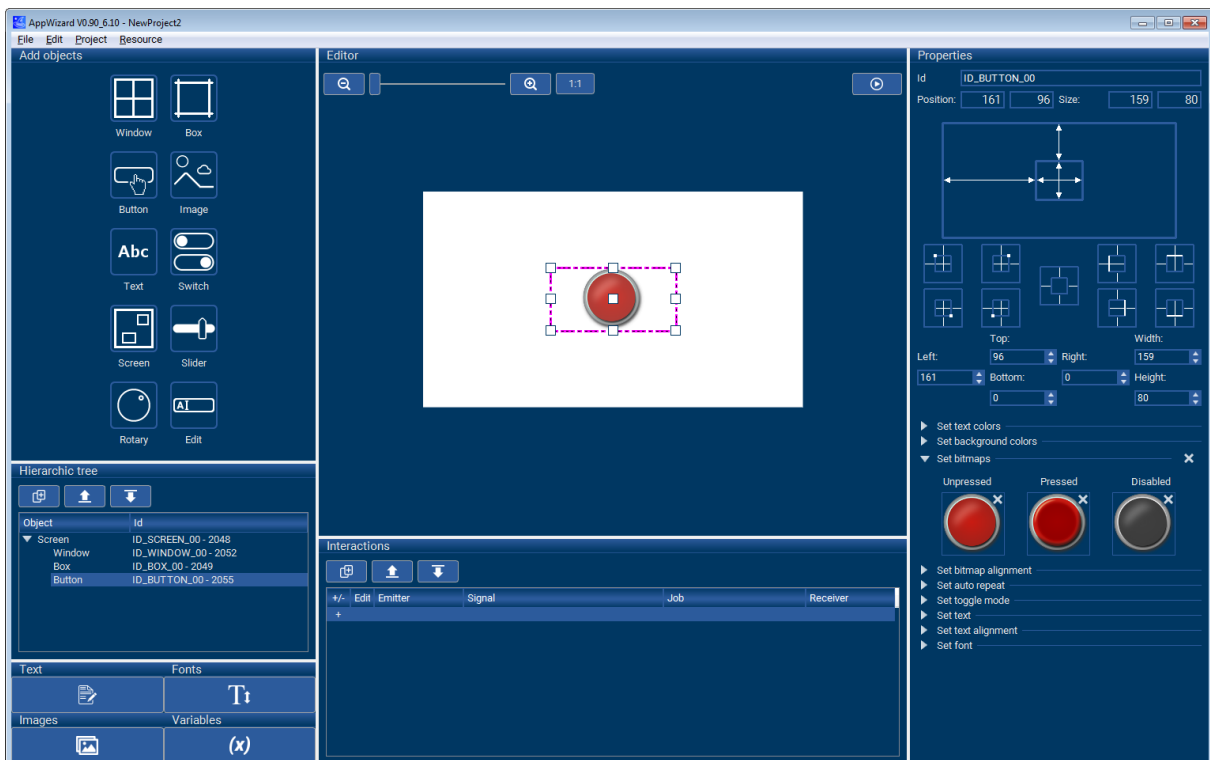
For detailed information about all of the AppWizard's features, refer to the document **UM03003 AppWizard User Guide & Reference manual**.

1.2 What is the AppWizard?

The AppWizard is a tool for creating complete and ready-to-use emWin applications. The tool makes it very easy to build an application, manage resources and even define the application's behavior.

The tool incorporates the idea of **WYSIWYG** ("What You See Is What You Get") which makes it very intuitive to use. Even testing the application is made easy with the AppWizard, it's only a press of the <F5> key.

Managing resources of an application is also made incredibly easy, since the user simply has to add their resources to the project. All of the rest is done by the AppWizard, this means the user doesn't have to fiddle around with dozens of resource files anymore.



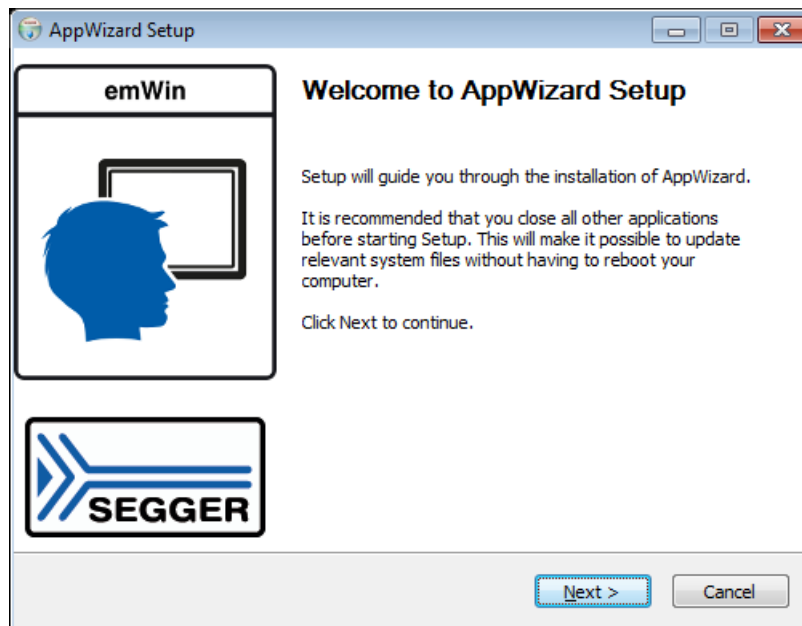
Chapter 2

Getting started

This chapter shows the user how to install the AppWizard.

2.1 Installation

To install the AppWizard, simply run the setup wizard which will guide you through the installation. It comes with all required components without having to download nor install further tools.



When the installation is finished the AppWizard may be opened by running **AppWizard.exe**, either from the Windows Start menu or the installation directory.

Chapter 3

Simple operations

This chapter demonstrates how simple operations within the AppWizard can be done, e.g. adding a button to the application.

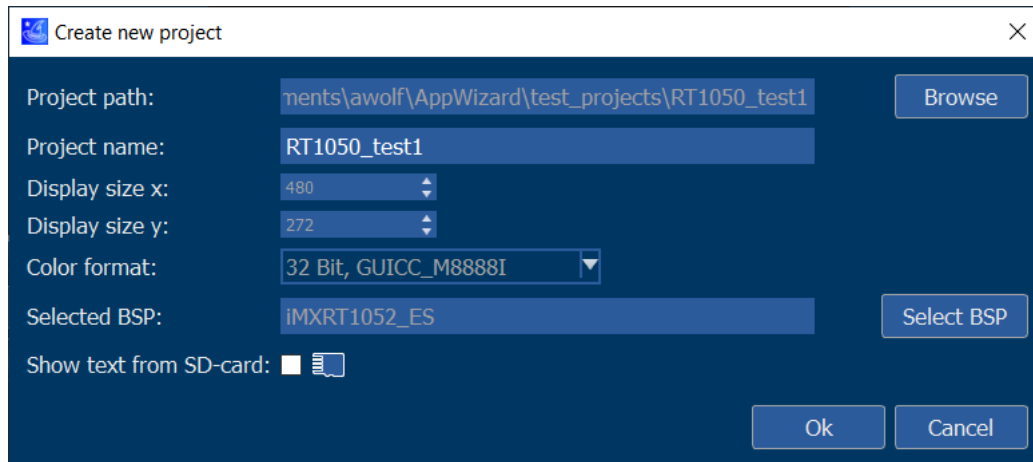
3.1 Creating a new AppWizard project

1. Run the SEGGER AppWizard from your desktop.



AppWizard

2. Click **Create new project**.
3. Choose a **Project path**. Give your AppWizard project a name in the Project name field. Select a **BSP** for this project. Then click **Ok**.



3.2 Adding a Screen

1. Click the **Screen** button to add a screen object. Notice an object called `ID_SCREEN_00` gets added.



3.3 Adding a Box

1. Click the **Box** button to add a screen-sized box (object ID_BOX_00) to your screen.



2. Give your box a different color by changing its color property on the right.



3.4 Adding a Button

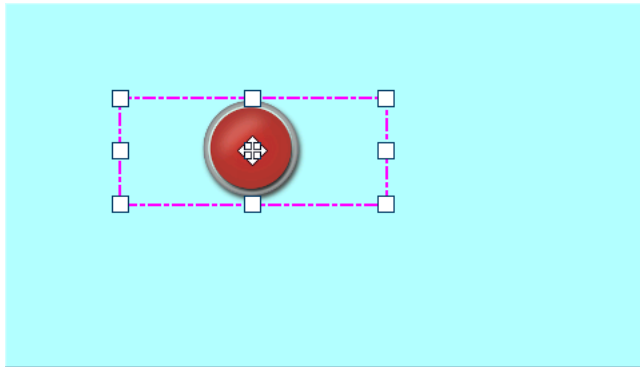
1. Click the **Button** button to add a button object.



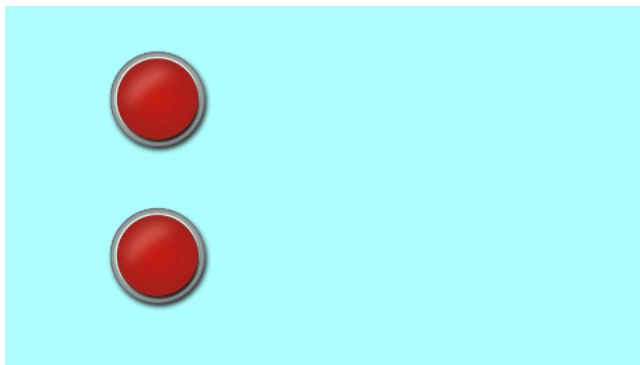
2. Assign a bitmap for the **unpressed** and **pressed** states to your button by changing the bitmap properties on the right.



3. Drag the button to position it where you want it on the screen.

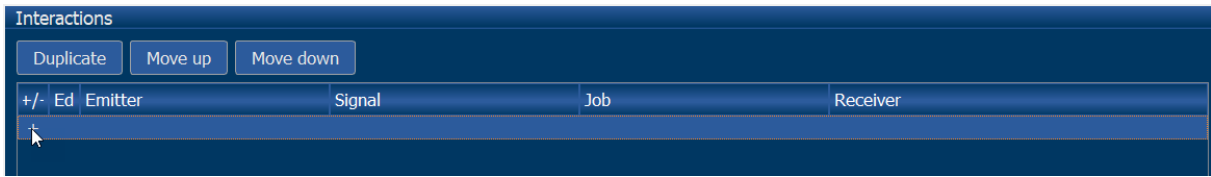


4. Add a second button the same way.

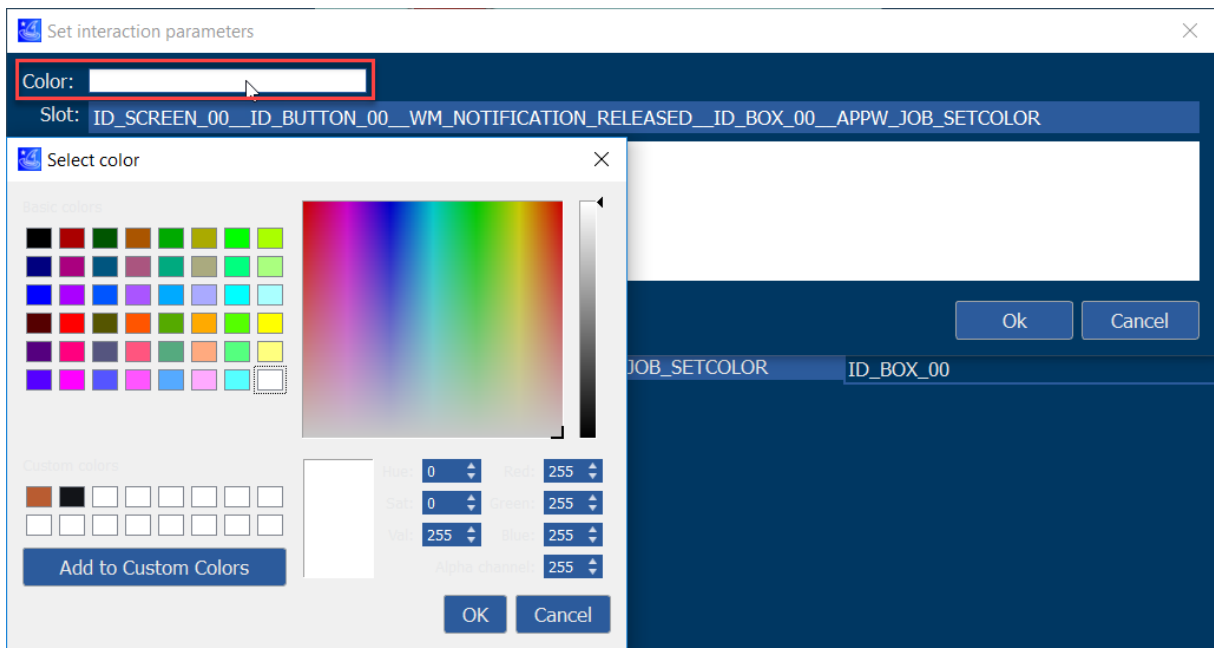


3.5 Adding Interactions

1. Select the first button (ID_BUTTON_00) and click on the '+' in the Interactions window to add an interaction.



2. As the **Emitter** (the object triggering the interaction), select **ID_BUTTON_00**.
3. As the **Signal**, select **WM_NOTIFICATION_RELEASED**.
4. As the **Job**, select **APPW_JOB_SETCOLOR**.
5. As the **Receiver**, select **ID_BOX_00**.
6. Set the color the box should change to when the button is released. Then click **Ok**.



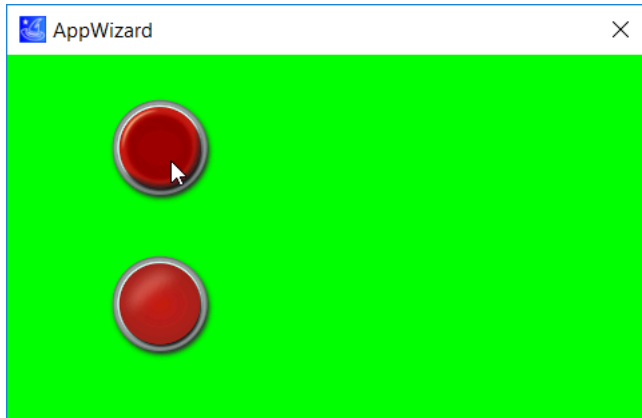
7. Add a second interaction that changes the color of ID_BOX_00 to a different color when the second button (ID_BUTTON_01) is released.



8. Check the behavior of your design by clicking the **Preview** button.



9. The background box should switch colors when you release the buttons. If things are not working as expected, fix your design.

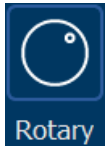


10. Close the preview window.

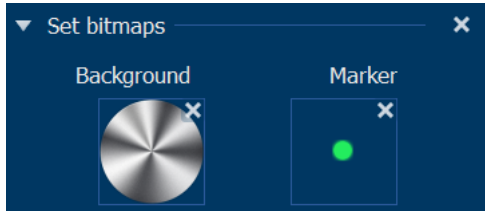


3.6 Adding a Rotary

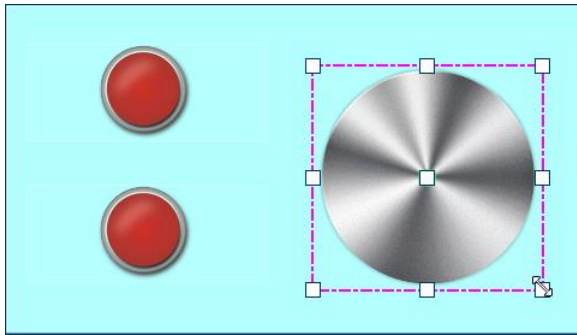
1. Click the **Rotary** button to add a rotary object to your design (ID_ROTARY_00).



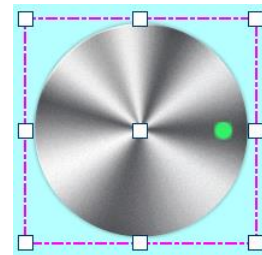
2. Pick the metal knob as the **bitmap** for your dial. Select the green dot as the **marker** for your dial.



3. Enlarge the object to fit the size of your dial's bitmap.



4. Increase the **Radius** property for the marker to fit your dial.



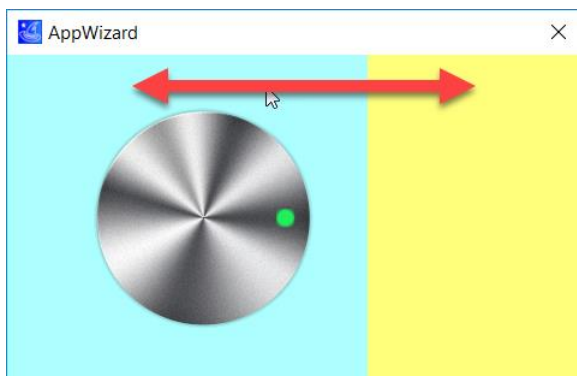
5. Increase the **Radius** property for the marker to fit your dial.
6. Check the behavior of your design by clicking the preview button.

3.7 Adding a Screen transition

1. Click the **Screen** button to add a second screen (`ID_SCREEN_01`).
2. Add a screen-sized box and give it a color.
3. Select **ID_SCREEN_01** in the **Hierarchic tree window**.

Hierarchic tree	
Object	Id
▼ Screen	ID_SCREEN_00
Box	ID_BOX_00
Button	ID_BUTTON_00
Button	ID_BUTTON_01
Rotary	ID_ROTARY_00
▼ Screen	ID_SCREEN_01
Box	ID_BOX_00

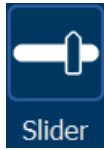
4. In the **Properties window**, configure horizontal motion for this screen:
Left partner: **ID_SCREEN_00** (your original screen)
5. Select **ID_SCREEN_00** in the Hierarchic tree window.
6. In the **Properties window**, configure horizontal motion for this screen:
Right partner: **ID_SCREEN_01** (your second screen)
7. Click the **Preview** button to start the Preview. You should now be able to swipe between the two screens using your mouse.



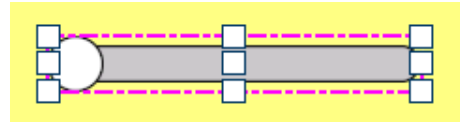
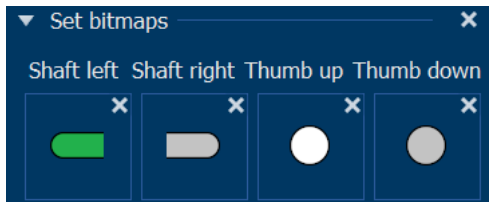
8. Close the preview.

3.8 Adding a Slider

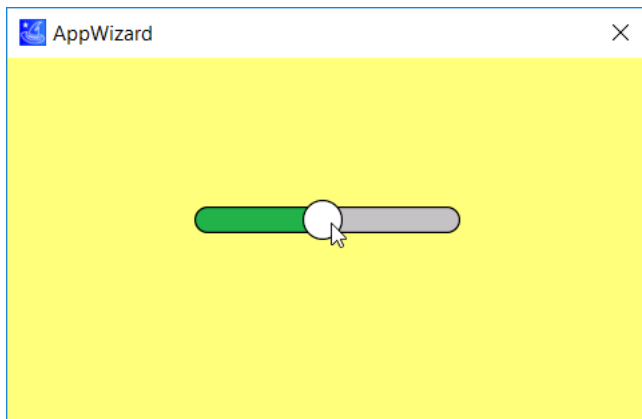
1. Select **ID_SCREEN_01** in the **Hierarchic tree** window.
2. Click the **Slider** button to add a slider to the screen.



3. Use the **Set Bitmaps** property to create a simple slider from the provided bitmaps.



4. Click the **Preview** button to test your slider.

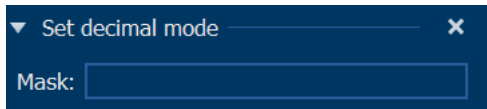


3.9 Adding Text

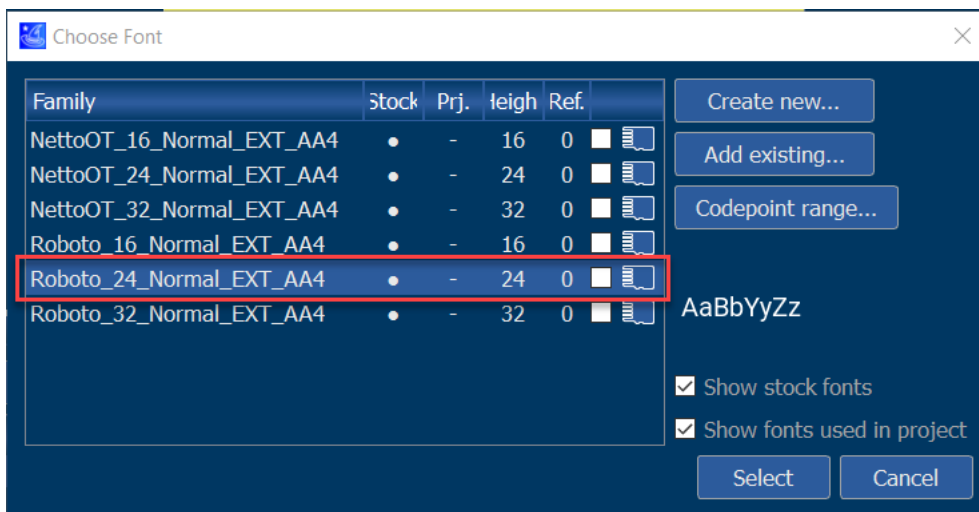
1. Click the **Text** button to add a text object.



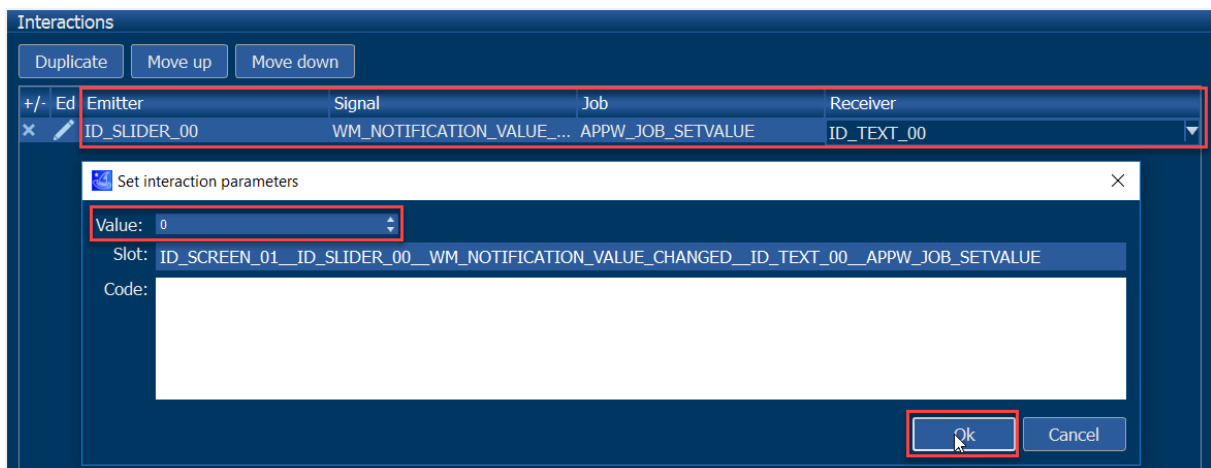
2. In the properties, set **decimal mode**.



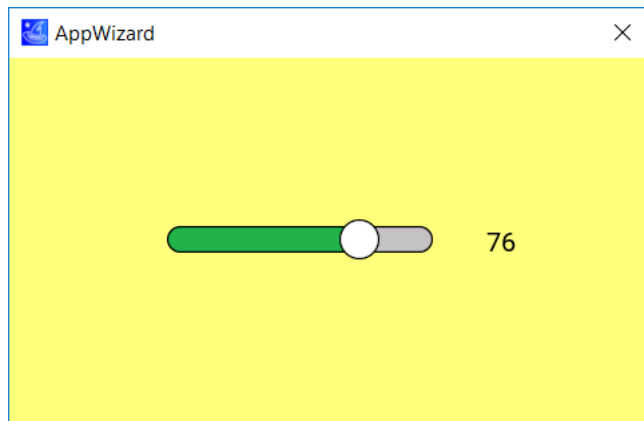
3. Set the font to use for the text. Pick **Roboto_24_Normal**.



4. Add an Interaction to display the slider value in the text box when the value changes:



5. Check the preview to confirm the expected result.



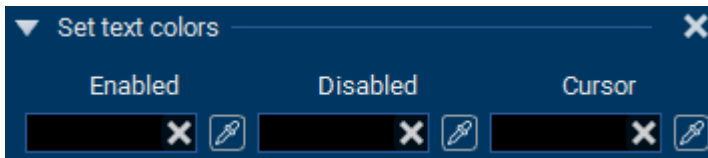
6. Close the Preview.

3.10 Adding an Edit object

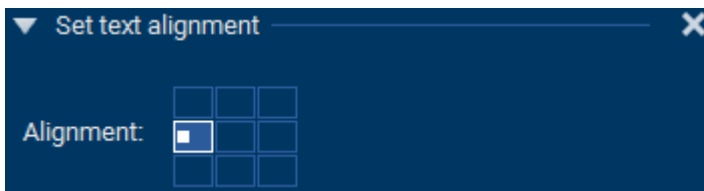
1. Click the **Edit** button to add a edit object.



2. Set the text color of the object by clicking on the **Set text colors** property and selecting a color for the different states and the cursor.



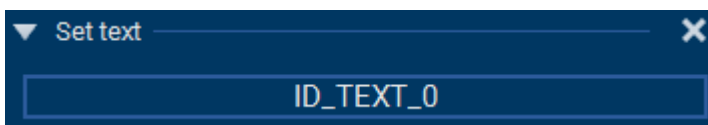
3. Set the font to be used for the text by clicking on the **Set font** property and clicking **Roboto_24_Normal**.
4. Define the text alignment by clicking on the **Set text alignment** property.



5. If needed, enable **decimal mode** and enter a mask (e.g. '0000' for four-digit values).



6. Alternatively, a text can be set for the object. To do this, select a text after clicking **Set text**.



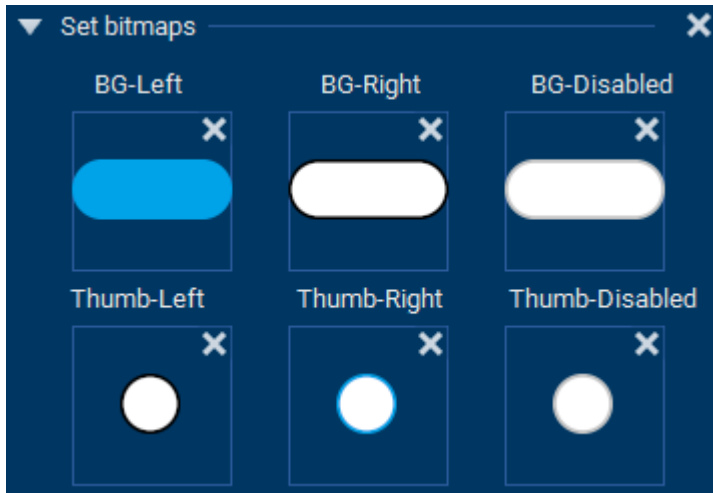
7. Check the behavior by clicking the preview button.

3.11 Adding a Switch

1. Click the **Switch** button to add a switch object.



2. Click **Set bitmaps** and select a fitting bitmaps for the background and thumb of the object for all states.



3. Select a font for the switch by clicking **Set font**.
4. Select text colors for both states of the object by clicking **Set text colors**.



5. Select a text for the left and right state of the object by clicking **Set left text** and **Set right text**.

Chapter 4

The first project

This chapter intends to guide new AppWizard users how to create a complete AppWizard application step-by-step and run it on a target.

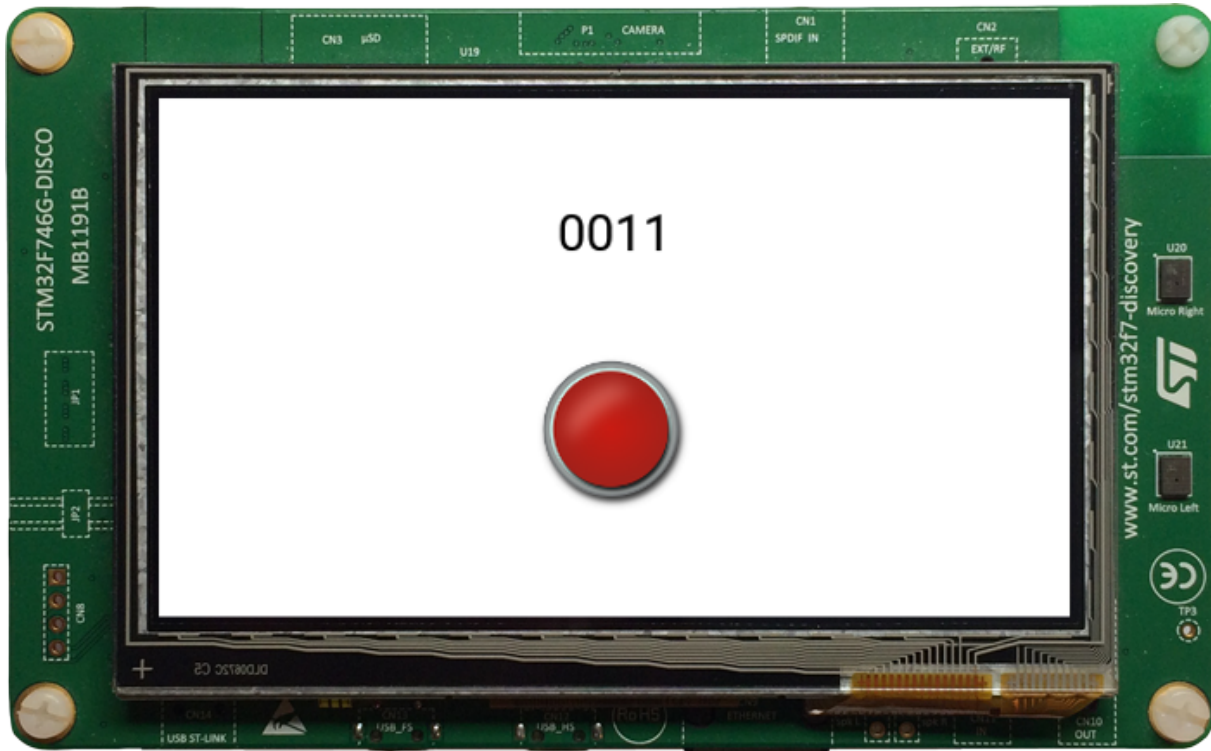
4.1 Example: Simple counter application

The following section provides a practical example for creating a project with the AppWizard. The example provides the exact steps that the user should perform.

The purpose of this example is to either be exactly followed by the new user to simply get an initial feel of how to work with the AppWizard, or to be inspired to create a different application.

Aim of this example

The final application will be a counter, that can be increased by clicking on a button. The sample is supposed to be run on an STM32F746G-Discovery board, but this hardware is not mandatory.



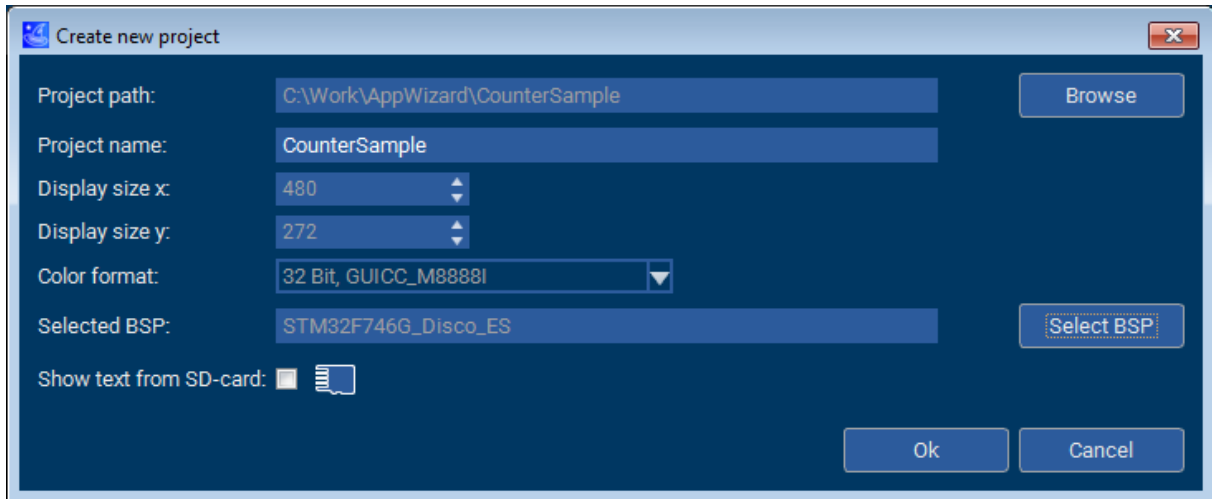
Note

If you're not able to use the STM32F746G-Discovery board for this example, you may select another one of the available BSPs or create a custom BSP if none of the BSPs shipped with the AppWizard fit your hardware. There are explanations on how this can be done in the AppWizard manual in the chapter "Board support packages (BSPs)".

4.1.1 Step 1: Create new project

When opening the AppWizard, the user is prompted to either load an existing project or to create a new one.

1. Click 'Create new project'.
2. Specify the project name, enter "CounterSample".
3. Click 'Select BSP' and select "STM32F746G-Discovery".



4.1.2 Step 2: Add objects to the screen

The next step is adding objects (similar to emWin's widgets) to the screen to create a structure for the application.

1. In the upper left corner of the AppWizard is the object section. Click 'Screen' to add a screen object.
2. To give the background a look, click 'Box' to add a box object.
3. Click 'Button' to add a button to the application.
4. Click 'Text' to add a text object to the application.



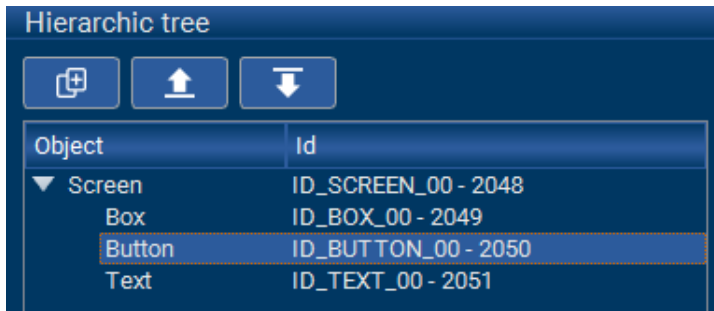
4.1.3 Step 3: Adjust object properties

Now, since all objects needed for this sample have been created, the object properties such as position and look need to be set.

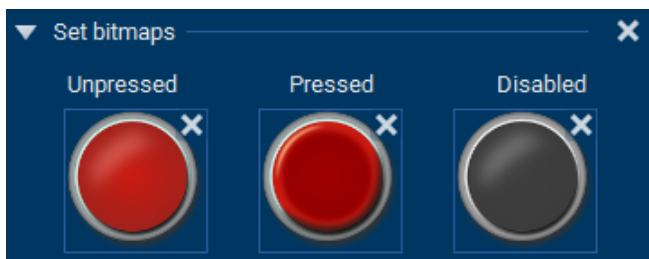
An object can be selected by clicking on it in the hierarchic tree view, seen on the left side of the tool. When it is selected, its properties such as size, position, color etc. may be edited in the properties section which is located on the right side of the tool.

Button object

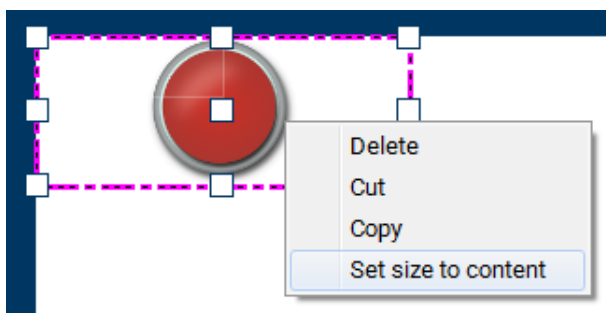
1. Click the button object within the hierarchic tree view to select it.



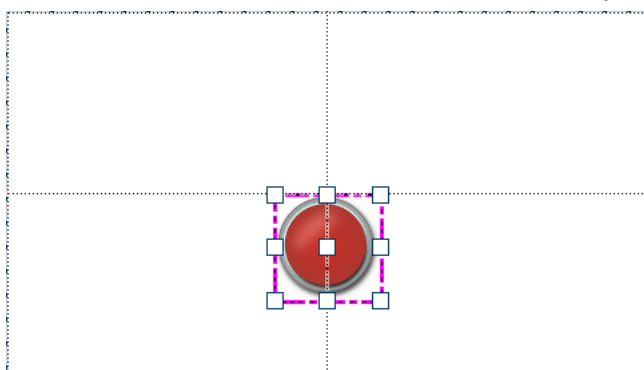
2. Click on the triangle next to 'Set bitmaps' in the 'Properties' section of the tool to set the bitmaps for the button.



3. Right-click the button object within the editor window and select "Set size to content". The editor window is located within the center of the AppWizard.

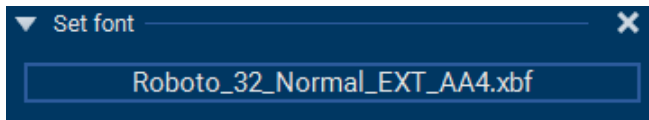


4. Move the button to the center of the screen by left-clicking and dragging it to the center.

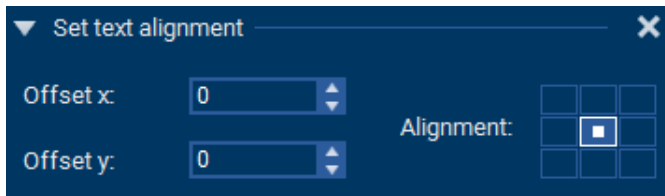


Text object

1. Click the text object within the hierarchic tree view to select it.
2. Set a font for the object by clicking the triangle next to 'Set font'.
3. Select the font `Roboto_32_Normal_EXT_AA4.xbf`.



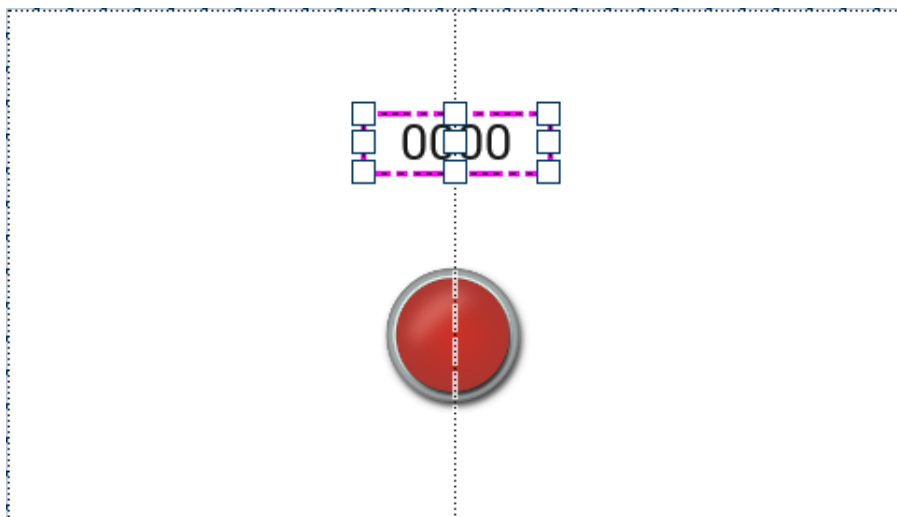
4. Set the text alignment to 'center' by clicking the center rectangle next to 'Alignment'.



5. Set the color of the text by clicking on the triangle next to 'Set text color'.
6. Select black, then click OK.
7. Enable decimal mode for the text object by clicking on the triangle next to 'Set decimal mode'.
8. Enter 0000 into the field that says 'Mask'.
9. Click the triangle next to 'Set range' and set it to Min: 0 and Max: 1000.



10. Move the item to the center of the screen, so it is above the button object.



Box object

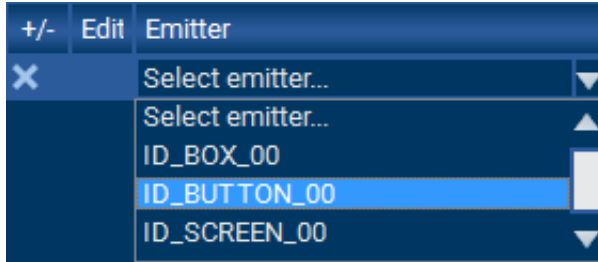
By default the box's color should be white. If, for some reason, it is not, follow the steps below.

1. Select the box object by clicking on it within the hierarchic tree view.
2. Click on the color property on the right side.
3. Select white, then click OK.

4.1.4 Step 4: Define the application's behavior

The user can define the behavior of the application by adjusting interactions between the objects.

1. Make sure one of the objects of the screen is selected within the hierarchic tree view.
2. Look for the interactions section of the tool, it is located in the lower part of the center, below the editor window.
3. Click into the empty field below the column header 'Emitter' and select the ID of the button, which should be ID_BUTTON_00.



4. Select WM_NOTIFICATION_RELEASED as the signal.
5. Select APPW_JOB_ADDVALUE as the job.
6. Select the ID of the text object as the receiver, which should be ID_TEXT_00.

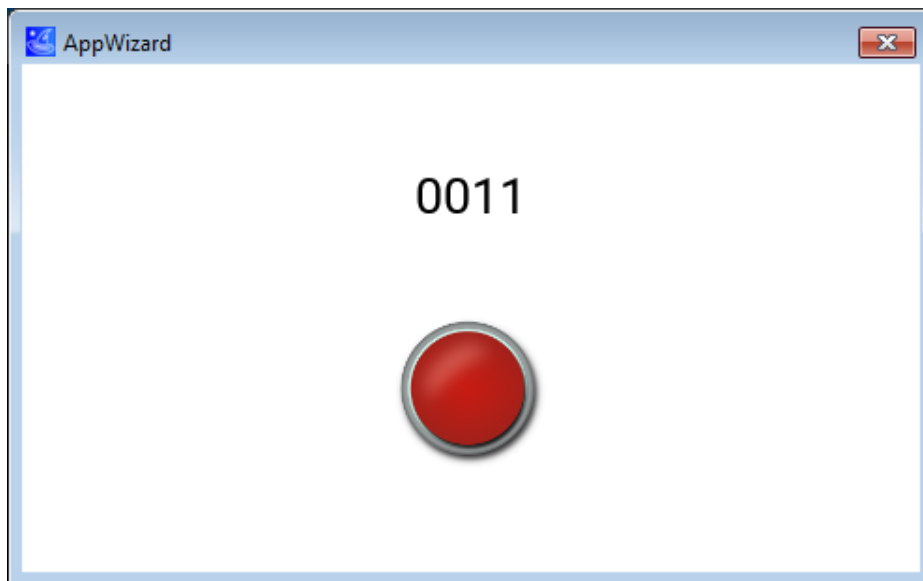


7. A window will open, asking for the interaction parameters. Enter 1 into the field that says 'Step'.



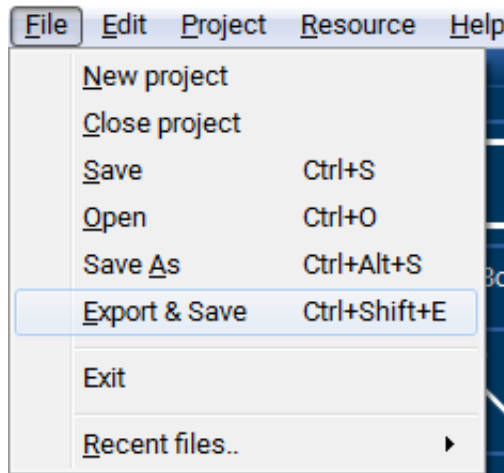
4.1.5 Step 5: Run the application

Now that the application is finished, it can be tested by clicking the play button or pressing the <F5> key. Each click on the button will increase the number the text object shows by one.



4.1.6 Step 6: Export the project

The next step is to export the AppWizard project. To do this, click **File → Export & Save**. The AppWizard generated a project for the SEGGER Embedded Studio IDE, as it specified for this BSP.



4.1.7 Step 7: Run on target

The .emProject file to run the exported project is located in the AppWizard project directory under Target/BSP/STM32F746_STM32F746G_Discovery/.

When compiling the project and flashing it onto the target, the application should run.

