GUI Swipe Transition Design by AppWizard

Example Code Introduction for 64/32-bit NuMicro® Family

Document Information

Application	This example code demonstrates how to use the SEGGER emWin AppWizard tool to design a graphical user interface (GUI) featuring swiping transitions, and utilizes the ordering machine interface to illustrate the functional design of dragging and swiping the screen to change pages.
BSP Version	emWin Software Package for MA35 Family
Hardware	NuMaker-HMI-MA35H0-A1 V1.0

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1. Overview

This example code will demonstrate how to use the SEGGER emWin AppWizard tool and the NuMaker-HMI-MA35H0-A1 evaluation board to complete the GUI (Graphical User Interface) design. The ordering machine system interface, with the interface design and human-computer interaction functions completed step by step, is used as an example to explain how to optimize performance and enhance the user experience.

1.1 UI Design

This example code uses the Nuvoton GUI development platform NuMaker-HMI-MA35H0-A1 evaluation board, which uses a 7" TFT-LCD resistive touch panel with a resolution of 1024x600.

In the project, the graphical user interface design uses the SEGGER emWin AppWizard tool, version V1.40f_6.34g. For the AppWizard *User Manual*, please refer to the official website documentation.

The IDE development is divided into different parts: the Objects, Hierarchic Tree, Resource and Properties, as shown in Figure 1-1. Users can design the UI layout by dragging and dropping components from the Objects onto the Editor window.

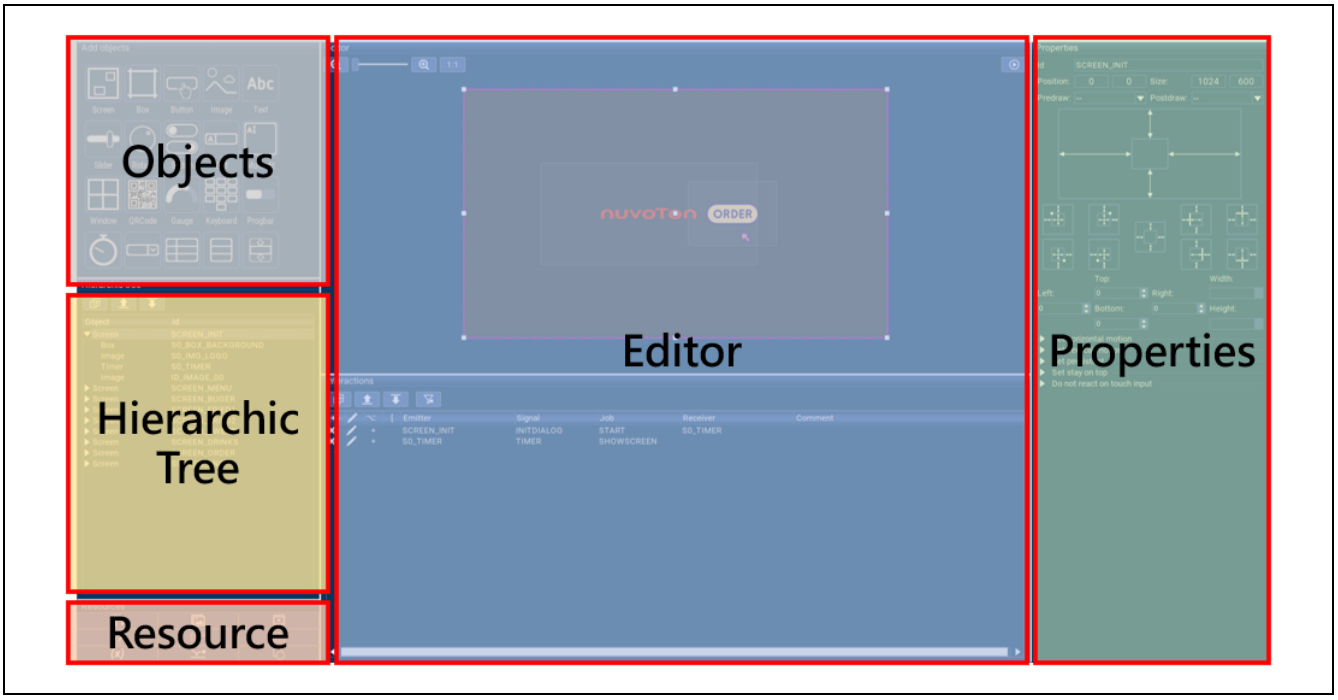

Figure 1-1 SEGGER emWin AppWizard IDE

For setting up the Swipe Transition project, user can refer to its configuration settings as shown in Figure 1-2. With the NuMaker-HMI-MA35H0-A1 evaluation board, set the Display size to 1024x600 pixels. Set the Color format to “32bit, GUICC_M8888I” and enable “Enable Multibuffering”.

The architecture diagram is shown in Figure 1-3. The GUI design divided into three pages: Home page, Order page, and Confirm Order page.

Home page	Display the home page and set the timer to jump to the next page after 3 seconds.
Order page	Displays a drag-and-drop window, allowing you to change pages by clicking on the menu or swiping. Click the "arrow" to jump page.
Confirm Order page	Confirm the meal page and click the “Order” button to return to the Order page.

Table 1-1 GUI Architecture Table

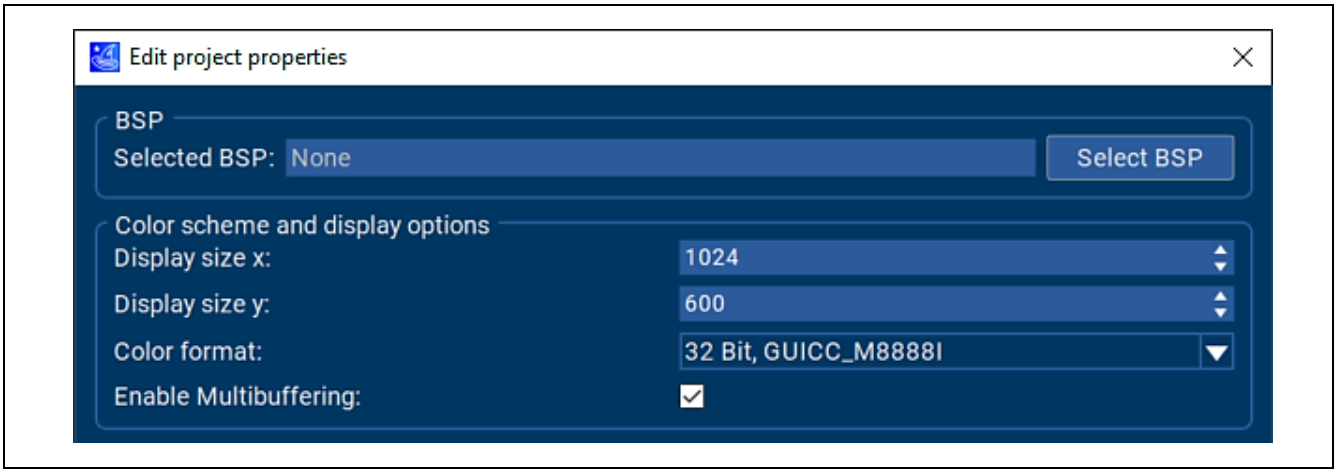

Figure 1-2 Project’s Properties

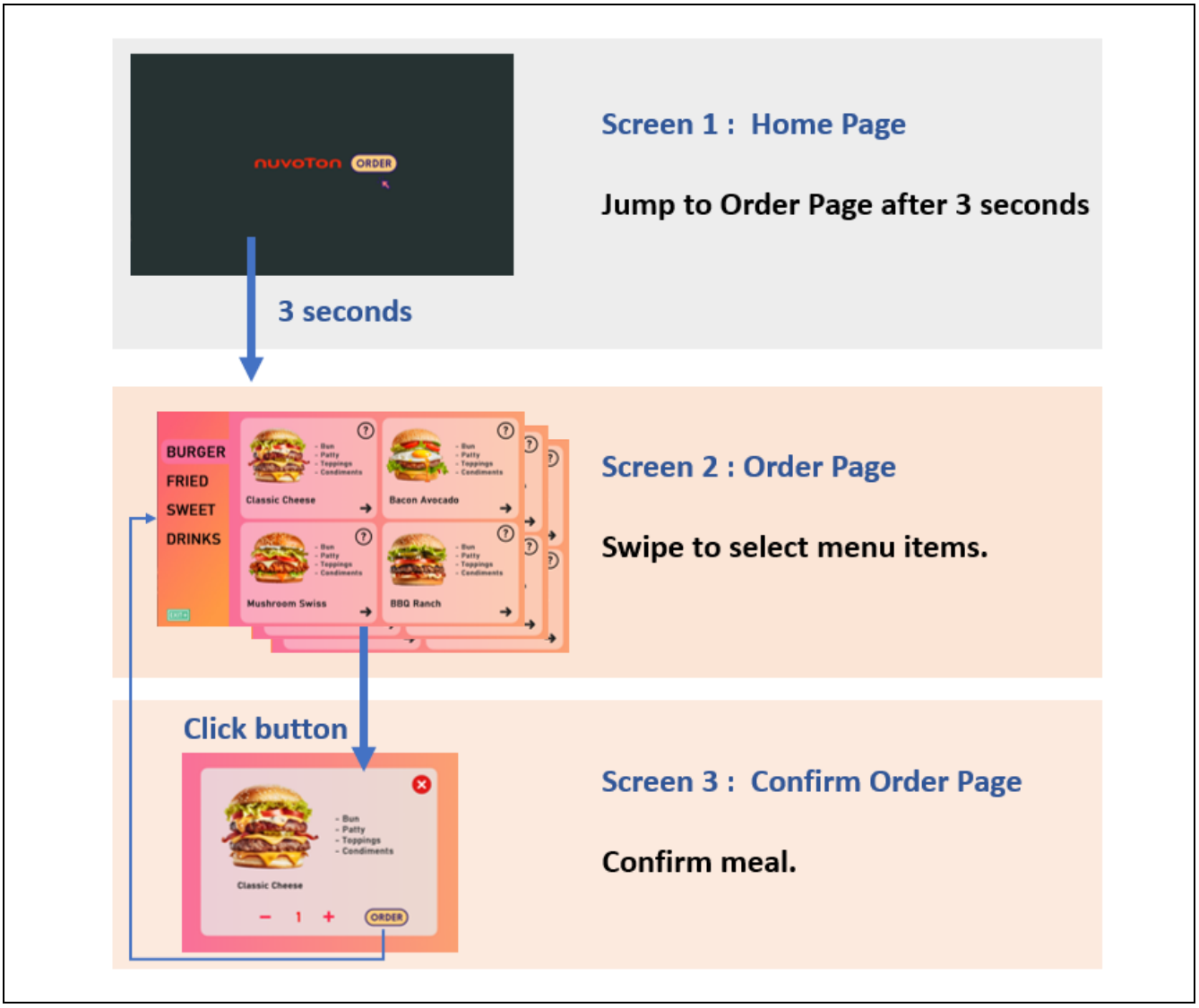

Figure 1-3 Flowchart of the GUI Swipe Transition Project

On the Home page design, the configuration is shown in Figure 1-4. Screen object, Box object and Image object are used in this page. The Screen object size is 1024x600 pixels and the background color is Color=ff263232 with the Box object.

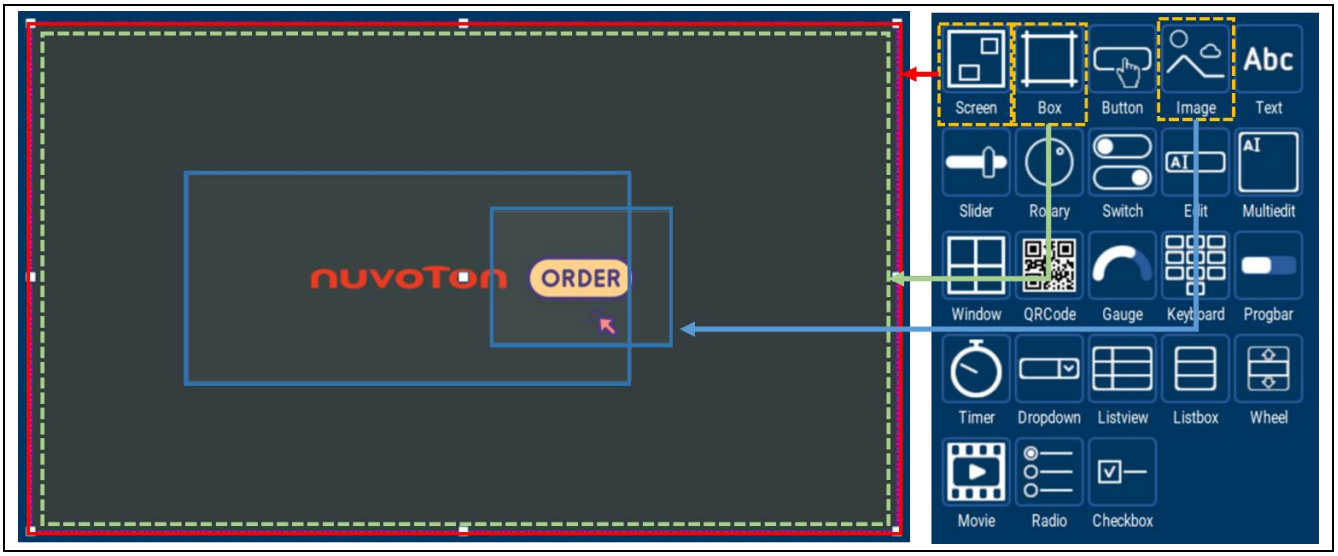

Figure 1-4 Home Page Design

On the Order page, the design is divided into left-side menu buttons and right-side meal items.

The menu buttons are arranged with Button objects. Both the Screen and Image size are set to 200x600 pixels, with the remaining areas configured as transparent. The meal item sets the Screen and Image size to 824x600 pixels. This page uses an Image object as the background and configures ordering buttons with a sliding setting: up and down sliding.

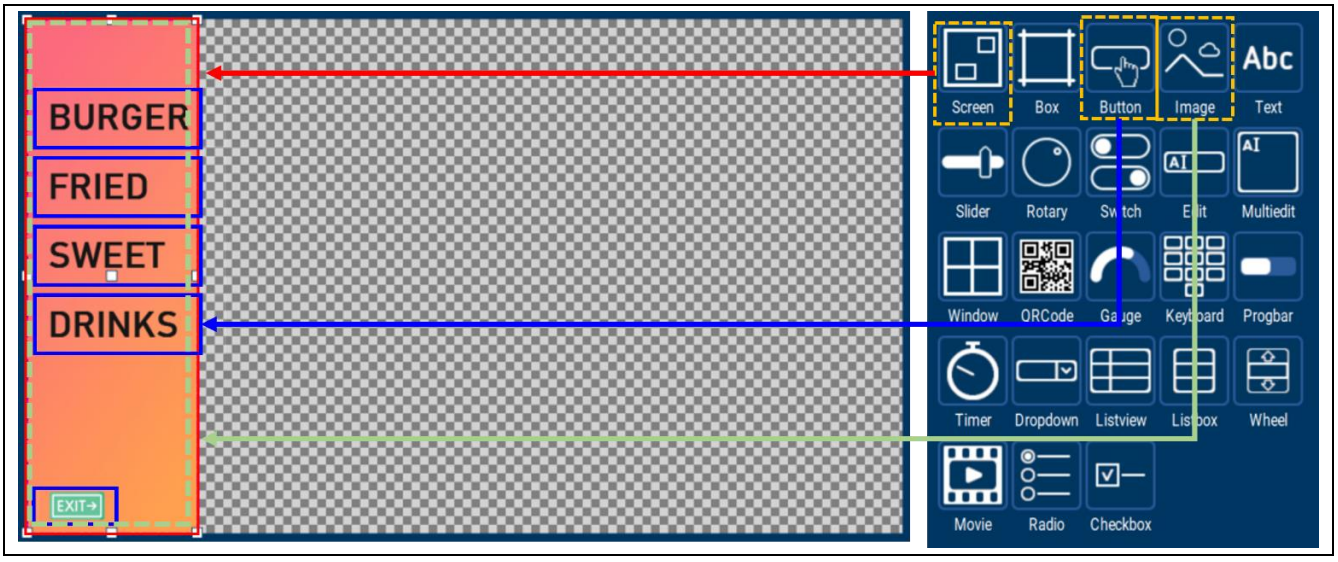

Figure 1-5 Order Page Menu Design

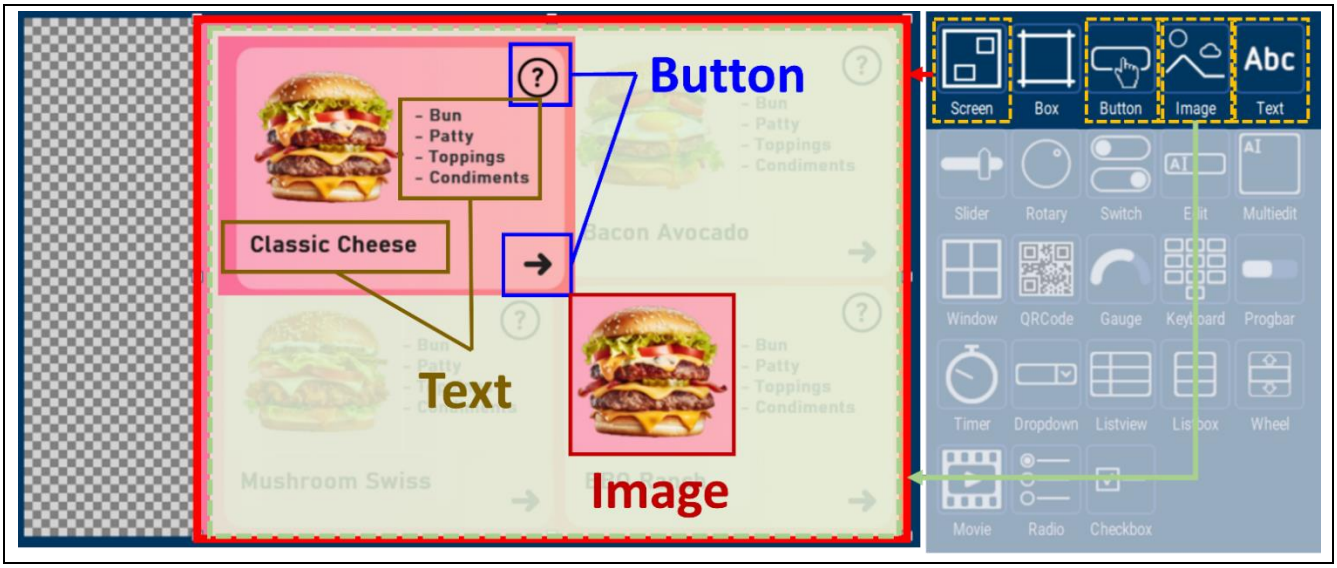

Figure 1-6 Order Page Meal Design

On the Confirm Order page, which serves as the meal confirmation page, users can add the number of meals by the button, as shown in Figure 1-7.

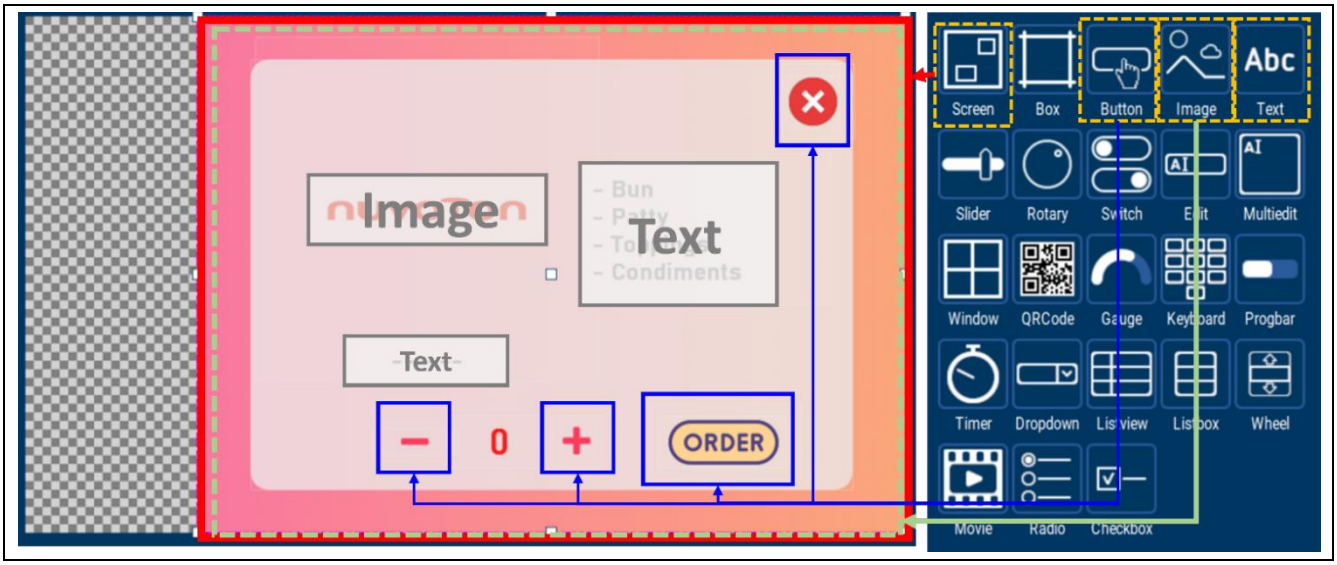

Figure 1-7 Confirm Order Page Design

1.2 UI Feature Design

The emWin AppWizard provides the "**Interactions**" function. Interactions can define user behavior operations and establish links between objects, such as automatic animation design or sliding transitions to switch between screens. The design interaction of this project is as follows:

Set timer=3000 on the Home page, jump to the Order page after three seconds, and set Interactions as shown in Figure 1-8. After the Screen initialization is completed, the timer will be started counting. When the three-second countdown is completed, the SHOWSCREEN Job will be triggered to switch to the Order page.

Figure 1-8 Home Page Interactions Settings

When the Order Page is created, the meal items need to be called at the same time, and the Interactions are set as shown in Figure 1-9. When the Screen is created, set SHOWSCREEN to display the meal items. Set the button to display different meal items when the button on the menu page is released.

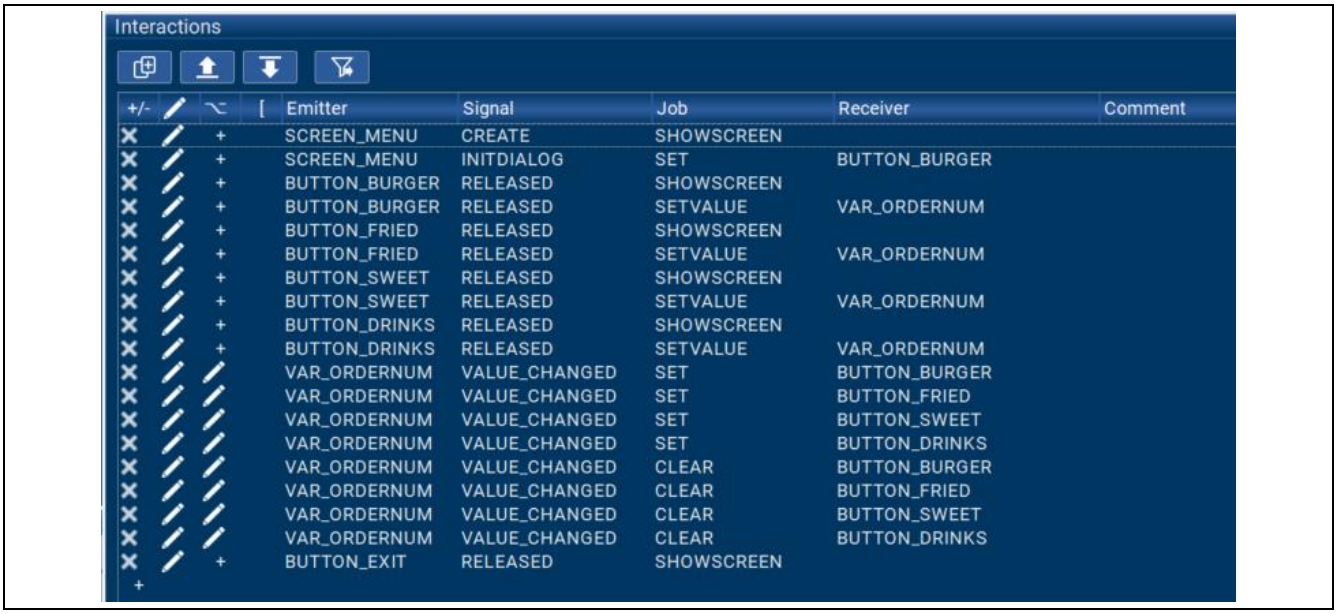

Figure 1-9 Order Page Interactions Settings

For the Screen properties settings, users can select vertical motion to enable up and down of swipe function and set the display settings for the Upper partner and Lower partner to swipe the page. The Properties settings are shown in Figure 1-10.

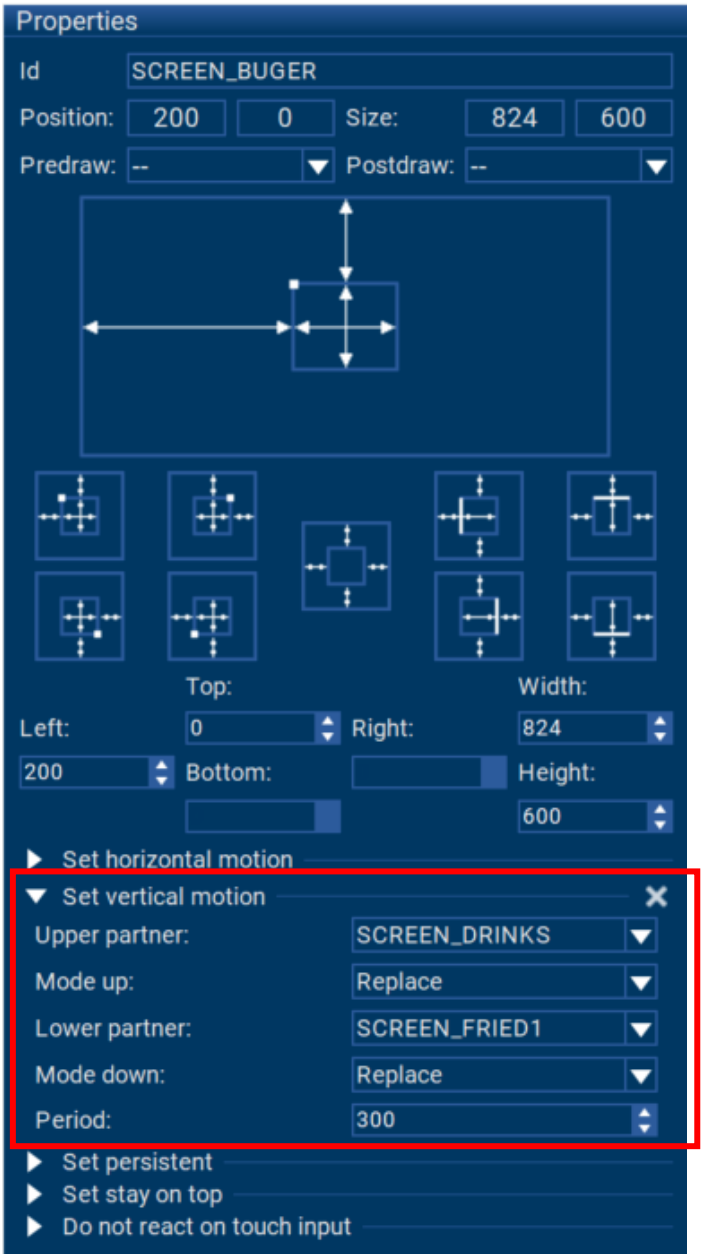

Figure 1-10 Screen Object Properties Settings

After the setting is completed, users can confirm the page settings for up and down sliding through PlayMode windows. The sliding area and result display are shown in Figure 1-11. The parameters of swiping function cannot be adjusted. The **SHIFTSCREEN** function needs to exceed half of the screen or the swiping speed is greater than a certain amount of change to trigger the page change condition. The user can adjust the page change animation speed, such as adjusting the **Period** option.

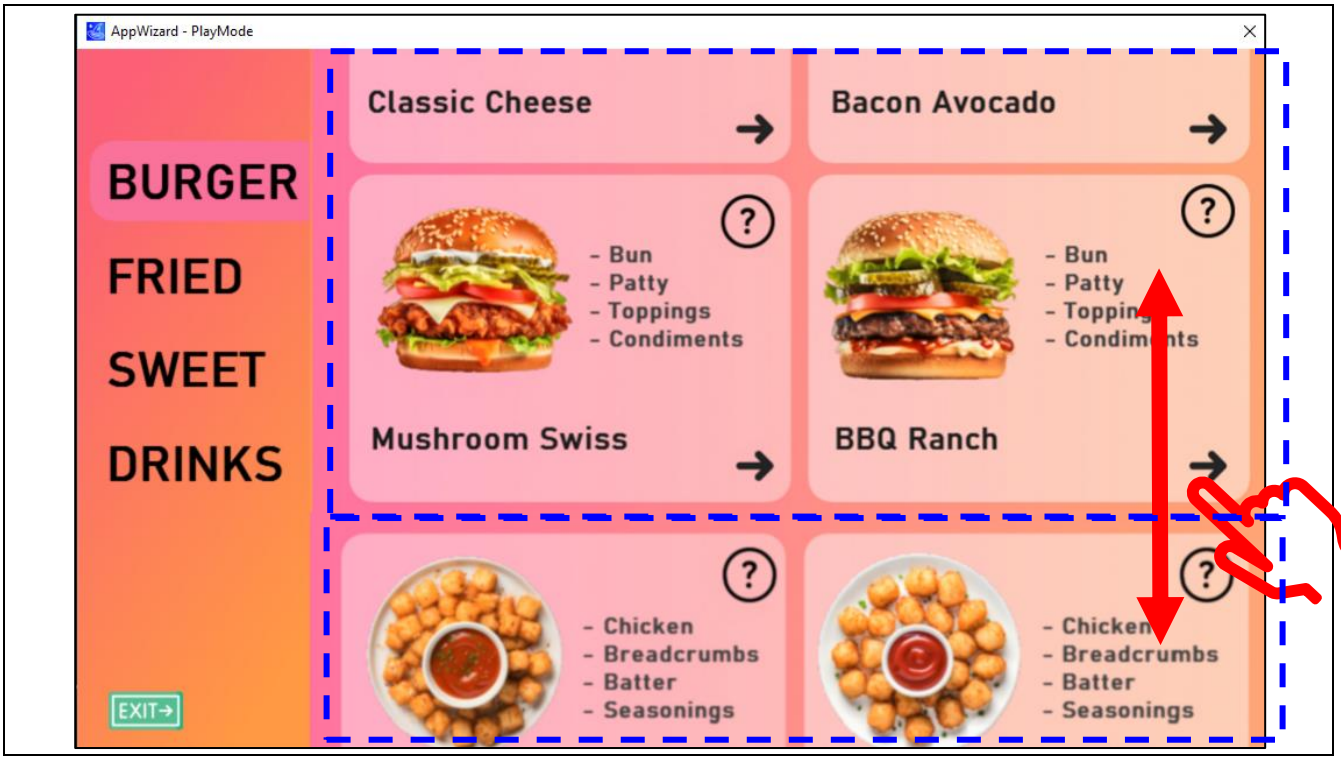

Figure 1-11 PlayMode Simulation Window

1.3 Swipe Transition Optimization

To reduce CPU usage and avoid additional calculations during touch control, images are first cropped to the expected size before loading. This prevents the CPU from having to perform scaling operations during loading picture. In this project, the Color format is GUICC_M8888I, and images are pre-selected as "**True color with alpha, RB swap, alpha inverted**", as shown in Figure 1-2.

The purpose of the above settings is to reduce CPU loading and avoid unnecessary calculations on the UI design and enhance system stability and improve user operating experience. The corresponding download location of the compilation environment development package is: *MA35H0 Industrial HMI Series - Nuvoton*.

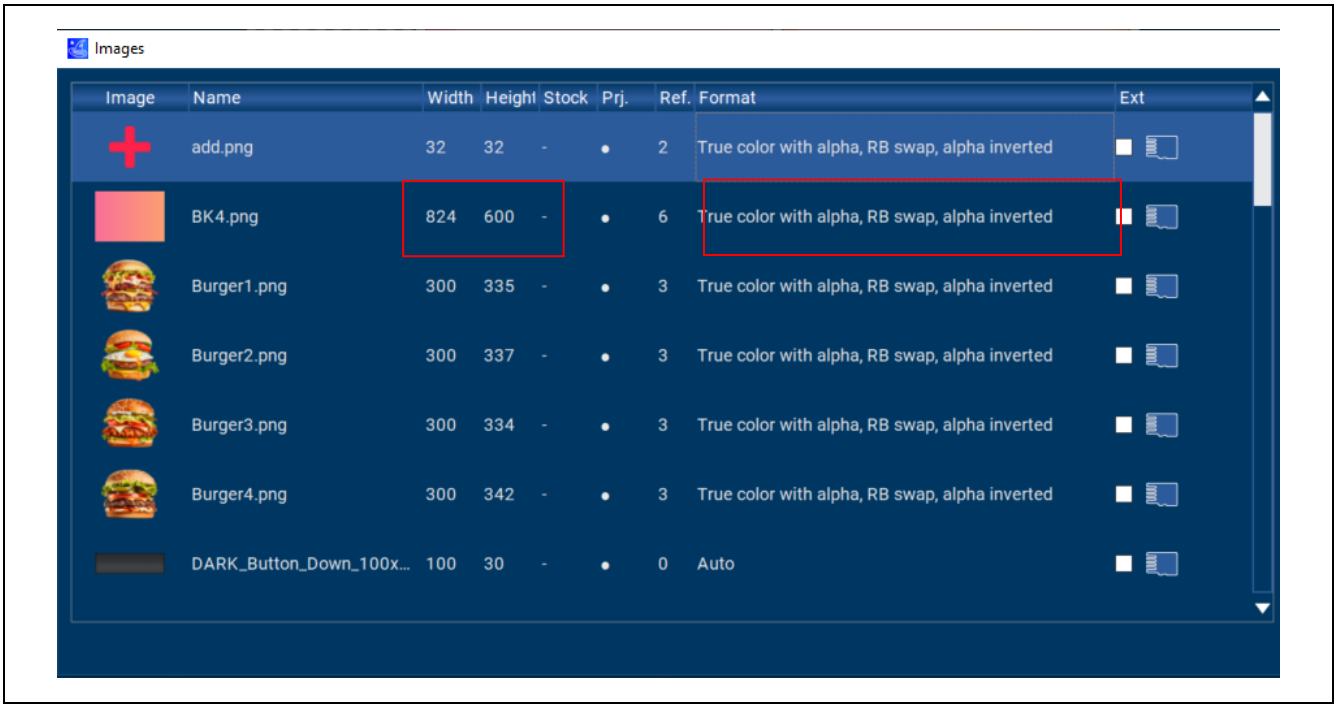

Figure 1-12 Image Object Properties Setting

Extract the *Linux emWin package*.
In the *A35_Family_Linux_emWin_64bit_Package/Sample/SimpleDemoAppWizard/LCDConf2.c* file, the **LCD_X_Config** defines the settings for the LCD. Within this function, the **GUI_DEVICE_CreateAndLink** function sets the color format processing. For this project, the Color Format is selected as "32bit, GUICC_M8888I", and modified to **GUI_DEVICE_CreateAndLink(GUIDRV_LIN_32, GUICC_M8888I, 0, 0);**.

```
void LCD_X_Config(void)
{
  … …
    //
    // Set display driver and color conversion for 1st layer
    //
    if (g_bits_per_pixel == 32)
         GUI_DEVICE_CreateAndLink(GUIDRV_LIN_32, GUICC_M8888I, 0, 0); // <Color Format
    else
         GUI_DEVICE_CreateAndLink(GUIDRV_LIN_16, GUICC_M565, 0, 0);
    if (LCD_GetSwapXY())
    {
        LCD_SetSizeEx (0, g_yres, g_xres);
        LCD_SetVSizeEx(0, g_yres * NUM_VSCREENS, g_xres);
    }
    else
    {
        LCD_SetSizeEx (0, g_xres, g_yres);
        LCD_SetVSizeEx(0, g_xres, g_yres * NUM_VSCREENS);
    }
    … …
}
```

2. Code Description

After completing the GUI design, the AppWizard tool will automatically convert the GUI design into program code. The user clicks the **File** > **Export & Save** button to generate the program code, as shown in Figure 2-1.

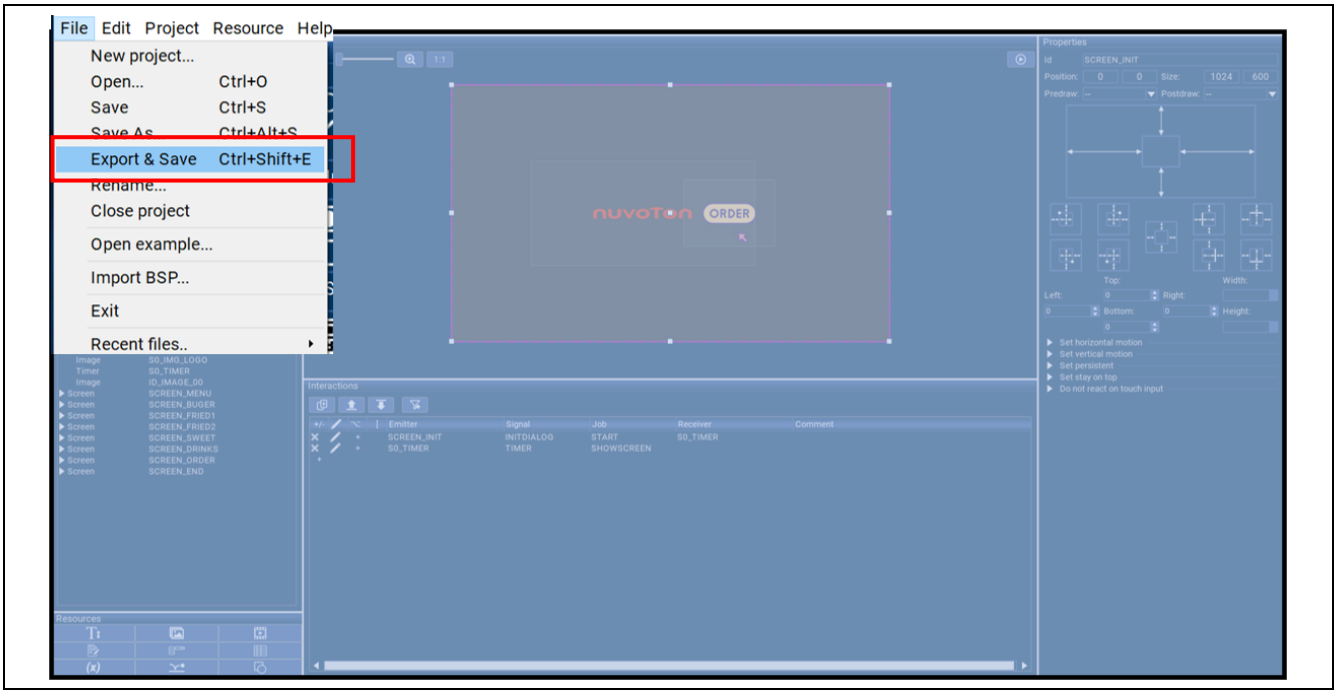

Figure 2-1 Export & Save of AppWizard Tool

In the Linux emWin package, unzip the file.

Program code *main.c* is used as the execution entry point of the code. Two threads will be used. One thread handles touch calculations, and the other is the main execution program and handles events such as refreshing the LCD panel. **MainTask2()** will execute the GUI code:

```
int main()
{
    … …

    pthread_create(&tid1, NULL, MainTask2_ISR, (void *)"MainTask2");
    pthread_create(&tid2, NULL, TouchTask_ISR, (void *)"TouchTask");

    pthread_join(tid1, NULL);
    pthread_join(tid2, NULL);

    /* Close LCD */
    munmap(pVideoBuffer, uVideoSize);  //return memory
    close(vpost_fd);
    return 0;
}
```

```
void MainTask2(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) {
    GUI_Exec1();
    APPW_Exec();
    GUI_X_Delay(1);
  }
}
```

Users only need to export the AppWizard project and overwrite all GUI code to the application's *Source* and *Resource* under *MA35_Family_Linux_emWin_64bit_Package*.

The cross-compilation process compiles the GUI code with the project into an executable file. For this cross-compilation process, please refer to the *MA35H0 Buildroot Project User Manual*.

3. Software and Hardware Requirements

3.1 Software Requirements

- emWin AppWizard IDE version
  - AppWizard_V1.40f_6.34g

3.2 Hardware Requirements

- Circuit components
  - NuMaker-HMI-MA35H0-A1 V1.0

4. Directory Information

EC_MA35_AppWizard_Swipe_Transition_V1.00	
AppWizard_Project	
Resource	Image files used for this example
Source	Source file of example code
Simulation	AppWizard simulation file

Table 4-1 Directory Structure

5. Example Code Execution

1. Download *MA35H0 Industrial HMI Series - Nuvoton* from Nuvoton official website.
2. Cover the files in the "*Resource*" and "*Source*" folders generated by the AppWizard project with those of the same names under the *Linux emWin package*.
   The path: */Sample/SimpleDemoAppwizard/Application*
3. Move the compiled execution file *SimpleDemoAppWizard* to the NuMaker-HMI-MA35H0-A1 evaluation board to execute.It is recommended to move the execution file through a USB storge or network. For how to move the executable file, please refer to *MA35H0 Buildroot Project User Manual*.
4. Execute the compiled execution file: *SimpleDemoAppWizard* or *EC_MA35_SwipeTransition*.

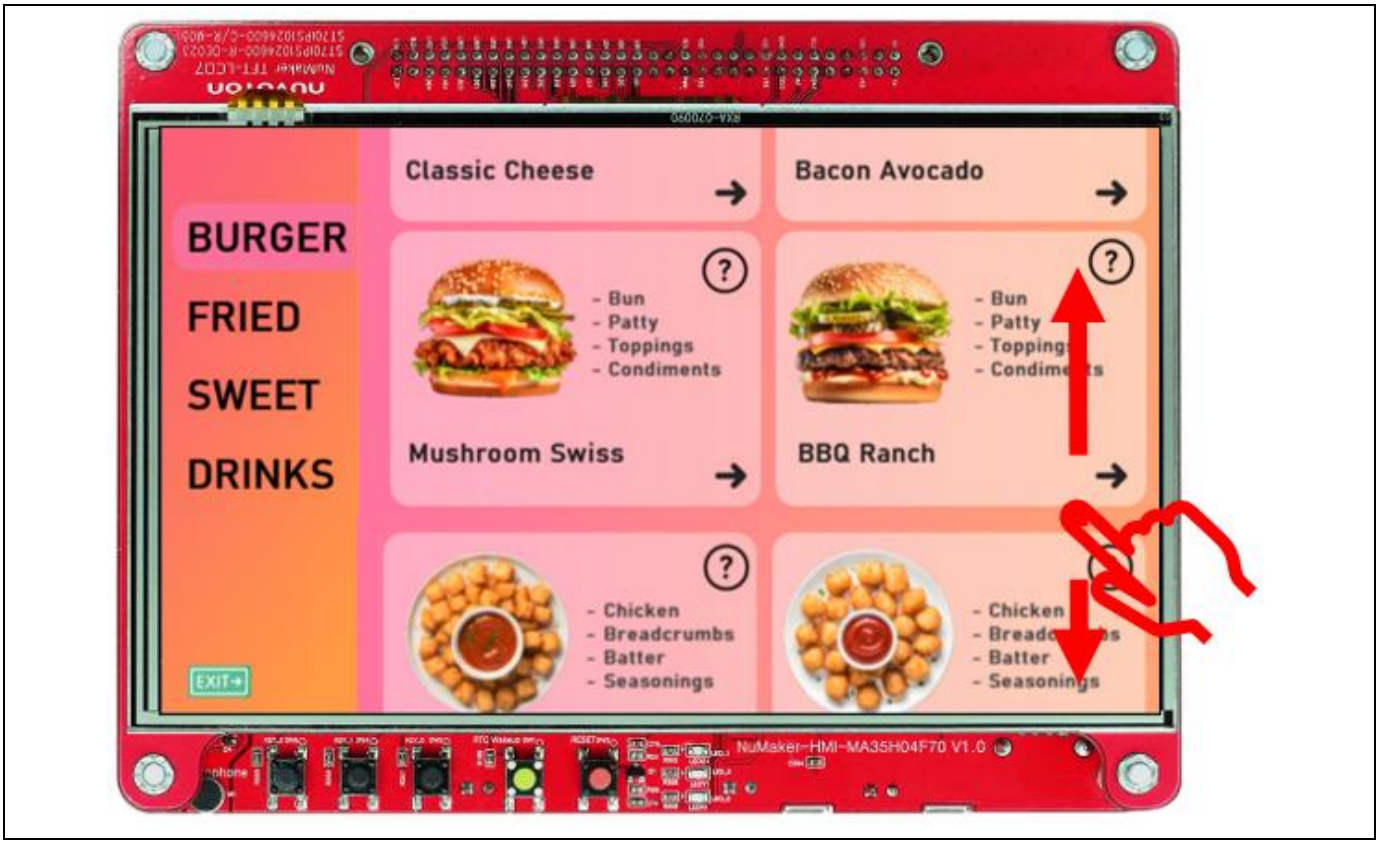

Figure 5-1 Execute the EC_MA35_SwipeTransition Result

6. Revision History

Date	Revision	Description
2024.05.10	1.00	Initial version.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.