Arm® 926-EJS

32-bit Microcontroller

N9H20
Freezer HMI
User Manual

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

www.nuvoton.com

Table of Contents

1 OVERVIEW

Freezer HMI for N9H20 is a GUI reference implementation.

This document utilizes Nuvoton N9H20 series geneal-purpose microprcessor N9H20K5 (32MB DDR) to implement Freezer HMI with emWin GUI library. Nuvoton emWin GUI library supports hardware JPEG, BitBLT and OSD.

Figure 1-1 Freezer HMI Main Menu

2 FEATURES

2.1 Freezer HMI Features

- Support Nuvoton MPU N9H20K5
- Supports hardware JPEG decoder for baseline decoding
- Supports hardware BitBLT rotation and OSD overlapping
- Supports high quality and contrast LCD panel with resolution up to 480 x 272
- Supports SEGGER licensed emWin GUI library
- Supports many popular image formats, e. g., PNG, GIF, JPG and BMP
- Supports user defined image as icon source

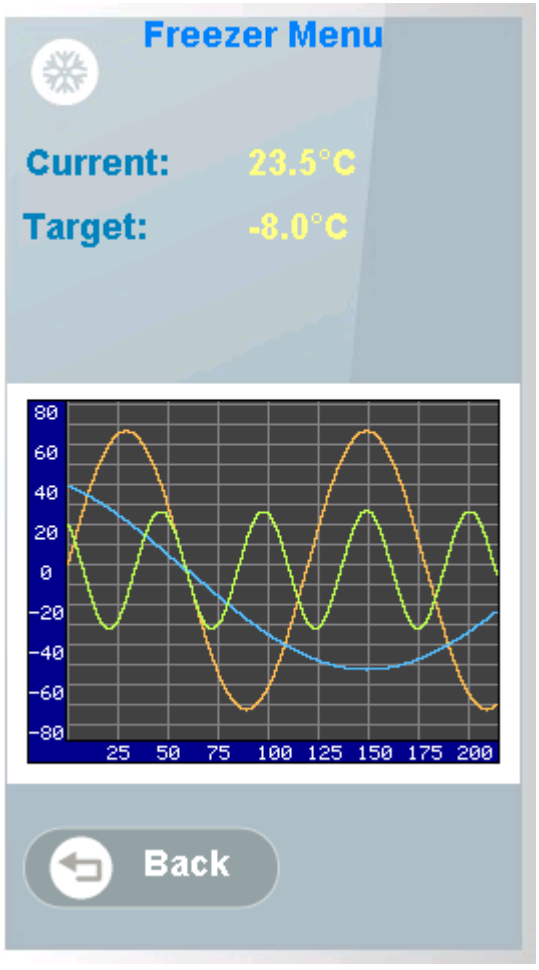

Figure 2-1 Freezer HMI Freezer Menu

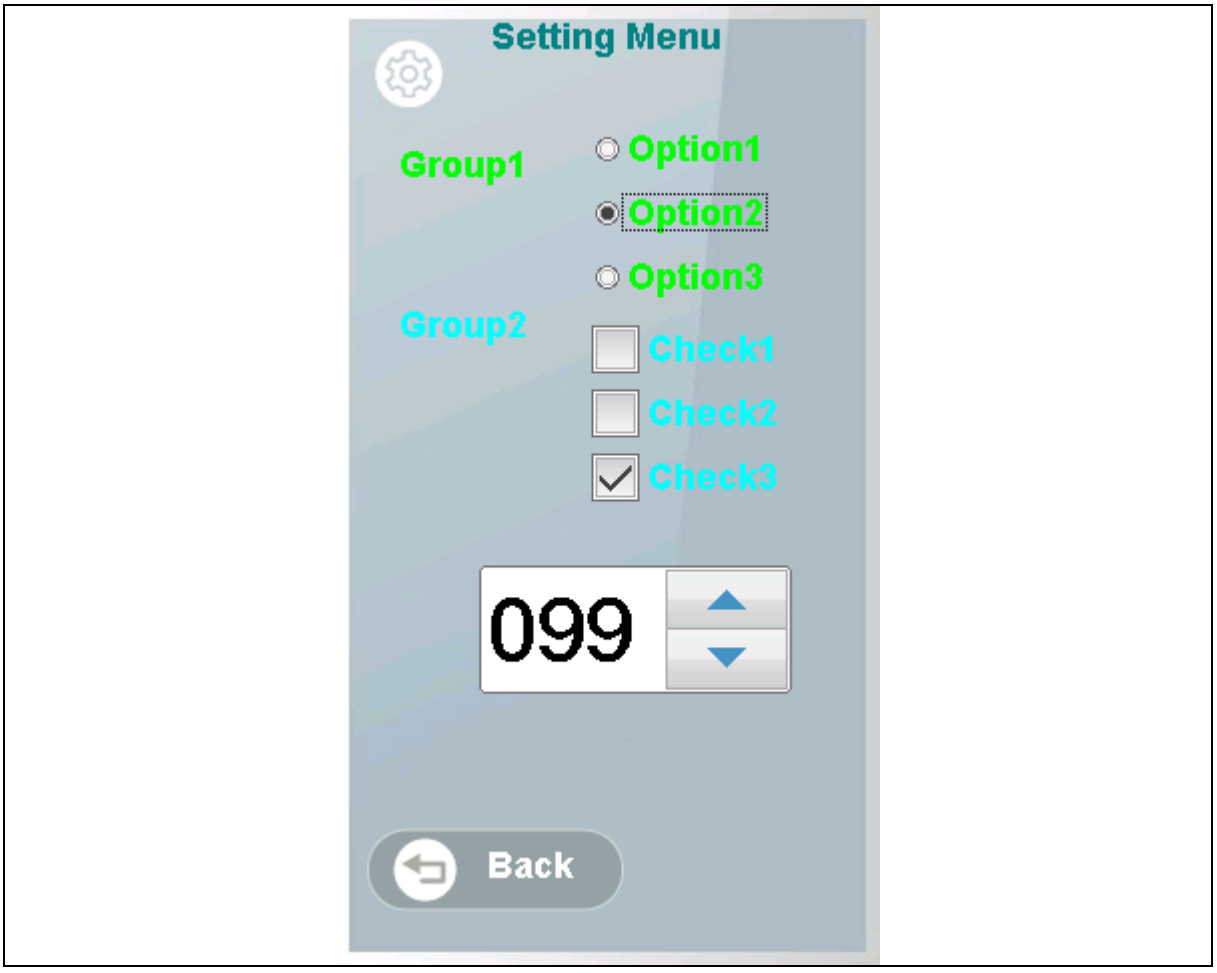

Figure 2-2 Freezer HMI Setting Menu

3 INSTALLATION AND ENVIRONMENT

3.1 Installing N9H20 Non-OS BSP

First, download the latest N9H20 Non-OS BSP from https://github.com/OpenNuvoton/N9H20_emWin_NonOS and unzip "*N9H20_emWin_NonOS-master.zip*" to a working folder, e. g., unzip it to the path "*C:\N9H20*", where "*N9H20*" is the working folder.

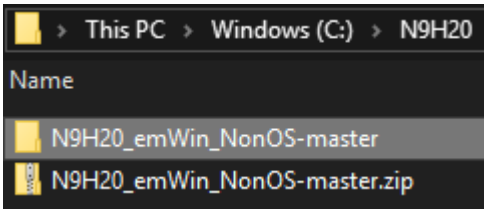

Figure 3-1 N9H20 Folder

The detailed information of N9H20 Non-OS BSP and emWin library can be found at "*N9H20_emWin_NonOS-master\N9H20 Readme.pdf*" and "*N9H20 emWin Quick Start Guide.pdf*" respectively.

3.2 Installing Freezer HMI

First, download and unzip the latest "*N9H_emWin_Template-master.zip*" from https://github.com/OpenNuvoton/N9H_emWin_Template and copy "*Freezer_N9H20_NonOS*" to the N9H20 sample path "*C:\N9H20\N9H20_emWin_NonOS-master\BSP\SampleCode\emWin*".

Then, open KEIL project file at "*C:\N9H20\N9H20_emWin_NonOS-master\BSP\SampleCode\emWin\Freezer_N9H20_NonOS\KEIL\SimpleDemo.uvproj*" and start compiling. The executable binary is in "*C:\N9H20\N9H20_emWin_NonOS-master\BSP\SampleCode\emWin\Freezer_N9H20_NonOS\Bin*", called "*conprog.bin*". Next, connect the USB cable between PC/NB and N9H20, then power on N9H20. Then copy "*conprog.bin*" to "*NAND1-1*" USB disk. Next, copy "*C:\N9H20\N9H20_emWin_NonOS-master\BSP\SampleCode\emWin\Freezer_N9H20_NonOS\Bin\NAND1-2*.**" to "*NAND1-2*" USB disk. Finally, remove the USB disk safely and reboot N9H20.

3.3 System Requirements

- KEIL IDE V5.xx and above with professional license
- Nuvoton N9H20K5 480 x 272 demo board (NuDesign HMI-N9H20 + NuDesign TFT-LCD4.3)

4 FOLDER STRUCTURE

4.1 Freezer HMI Folder Structure

The content of "*Freezer_N9H20_NonOS*" is described as follows.

Folder	Description
Freezer_N9H20_NonOS	Base folder ● Changelog.pdf is version history ● Freezer_Reference_Implementation.pdf is user manual ● main.c is for Freezer HMI main entry
Application	HMI folder ● GUIConf2.c is for emWin memory pool ● LCDConf2.c is for emWin multiple buffers ● NVT_Config.c is for Nuvoton platform ● Main1DLG.c is for main menu ● NVT_Main1.c is for main menu extension ● Freezer1DLG.c is for freezer menu ● NVT_Freezer1.c is for freezer menu extension ● Setting1DLG.c is for setting menu ● NVT_Setting1.c is for setting menu extension ● GUIDEMO_Graph.c is for freezer chart
Bin	Pre-built binaries folder ● conprog.bin is for Freezer HMI execution file
Bin / NAND1-1	Resource folder ● conprog.bin is for Freezer HMI execution file and identical with Bin's conprog.bin
Bin / NAND1-2	Dta resource folder ● main_background.dta is main menu background image ● freezer_background.jpg is freezer menu background image ● setting_background.jpg is setting menu background image ● main_freezer.dta is freezer menu button image ● main_setting.dta is setting menu button image ● freezer_back.dta is freezer menu back button image ● setting_back.dta is setting menu back button image ● logo.dta is company logo image
Bin / 9To565	Any image convert to dta file format

Bin / 28ToA565	Any image convert to dta file format with alpha-channel
Bin / LogGen	Any image convert to logo file format
KEIL	Arm Keil MDK project folder

Table 4-1 Freezer HMI Folder Structure

5 DESIGN GUIDE

Freezer HMI reference implementation guide assumes that you already have a mature knowledge of the following:

- IDE operation for editing and compiling
- The C programming language, how to use linker and C compiler
- The N9H20 Non-OS BSP programming knowledge
- The basic emWin programming knowledge

Note: the basic Freezer HMI utilizes SEGGER's emWin GUI library. About SEGGER's emWin GUI library user manual can be found at "*C:\N9H20\N9H20_emWin_NonOS-master\BSP\ThirdParty\emWin\Doc\UM03001_emWin.pdf*".

5.1 Multiple-Buffer Control

Freezer HMI utilizes multiple-buffer control to avoid tearing and flickering. You need to declare three frame buffers:

```
main.c


UINT8 u8FrameBuf[XSIZE_PHYS*YSIZE_PHYS*2*3] __attribute__((aligned(32)));


LCDConf2.c


//
// Buffers
//
#define NUM_BUFFERS  3 // Number of multiple buffers to be used
```

5.2 LCD Rotation Control

Freezer HMI utilizes LCD rotation control to use landscape LCD as portrait LCD:

```
LCDConf2.c


//
// Orientation
//
#define DISPLAY_ORIENTATION (GUI_MIRROR_Y | GUI_SWAP_XY)
```

```
GUI_SetOrientation(DISPLAY_ORIENTATION);
```

5.3 Main Menu Control

Main menu is generated from GUIBuilder, called "*Main1DLG.c*". You can open this file by GUIBuilder and re-arrange widget postion, size and property. It contains 2 IMAGEs, 1 TEXT and 2 BUTTONs.

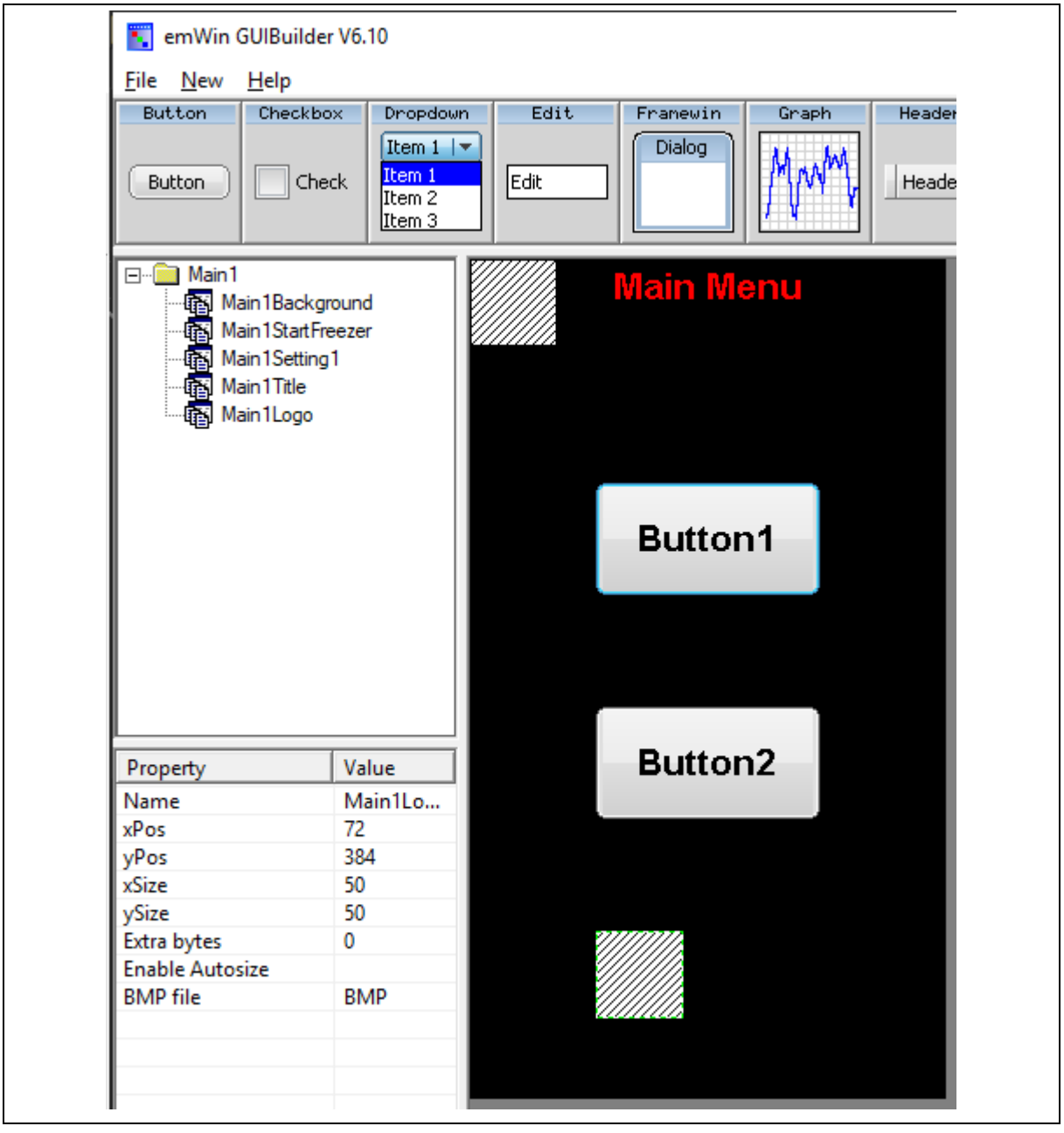

Figure 5-1 Freezer HMI Main Menu

By "*NVT_Main1.c*", you can assign user define dta image file to replace the original.

List of all main menu control definitions by the Freezer HMI.

```
NVT_Main1.c
```

```
//
// DTA image file full path (modify if needed)
//
// Logo
//
#define NVT_MAIN_LOGO "D:\\logo.dta"
//
// Freezer HMI main menu background
//
#define NVT_MAIN_BACKGROUND "D:\\main_background.dta"
//
// Main menu button image
//
#define NVT_MAIN_BUTTON1 "D:\\main_freezer.dta"
#define NVT_MAIN_BUTTON2 "D:\\main_setting.dta"
//
// Increase DTA buffer size if needed
//
static UINT8 s_u8BackgroundImageBuf[257 * 1024]
__attribute__((aligned(32)));
static UINT8 s_u8LogoImageBuf[26 * 1024] __attribute__((aligned(32)));
static UINT8 s_u8Button1ImageBuf[26 * 1024] __attribute__((aligned(32)));
static UINT8 s_u8Button2ImageBuf[26 * 1024] __attribute__((aligned(32)));
```

5.4 Freezer Menu Control

Freezer menu is generated from GUIBuilder, called "*Freezer1DLG.c*". You can open this file by GUIBuilder and re-arrange widget postion, size and property. It contains 1 IMAGE, 5 TEXTs, 1 BUTTON and 1 GRAPH.

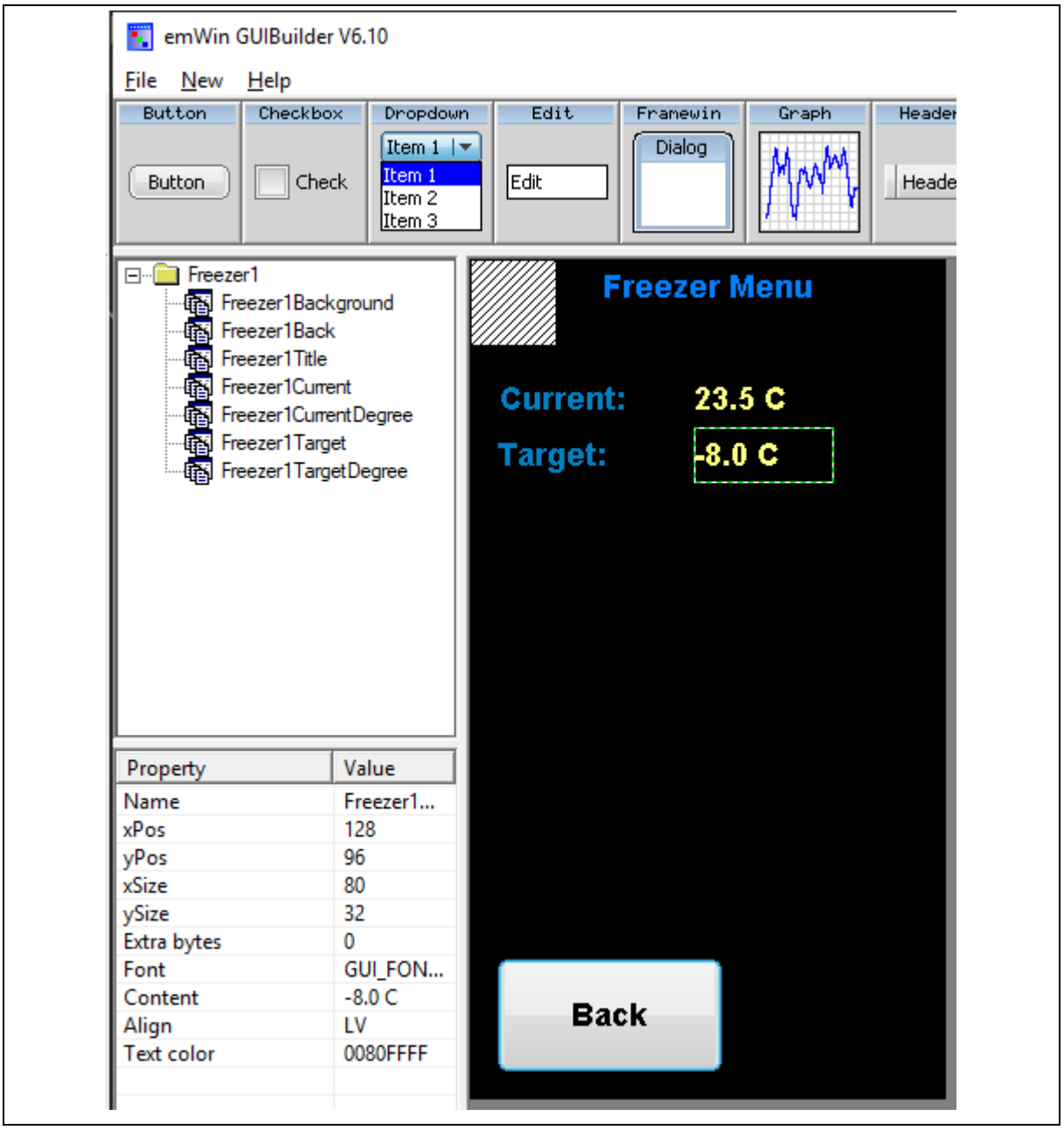

Figure 5-2 Freezer Menu Control

By "*NVT_Freezer1.c*", you can assign user define dta image file to replace the original.

List of all freezer menu control definitions by the Freezer HMI.

NVT_Freezer1.c

```
//
// Freezer HMI freezer menu background
//
#define NVT_FREEZER_BACKGROUND "D:\\freezer_background.jpg"
//
// Freezer menu button image
//
#define NVT_FREEZER_BUTTON1 "D:\\freezer_back.dta"
//
// Increase DTA buffer size if needed
//
static UINT8 s_u8BackgroundImageBuf[200 * 1024]
__attribute__((aligned(32)));
static UINT8 s_u8Button1ImageBuf[26 * 1024] __attribute__((aligned(32)));
```

5.5 Setting Menu Control

Setting menu is generated from GUIBuilder, called "*Setting1DLG.c*". You can open this file by GUIBuilder and re-arrange widget postion, size and property. It contains 1 IMAGE, 3 TEXTs, 1 BUTTON, 1 RADIO, 1 CHECKBOX and 1 SPINBOX.

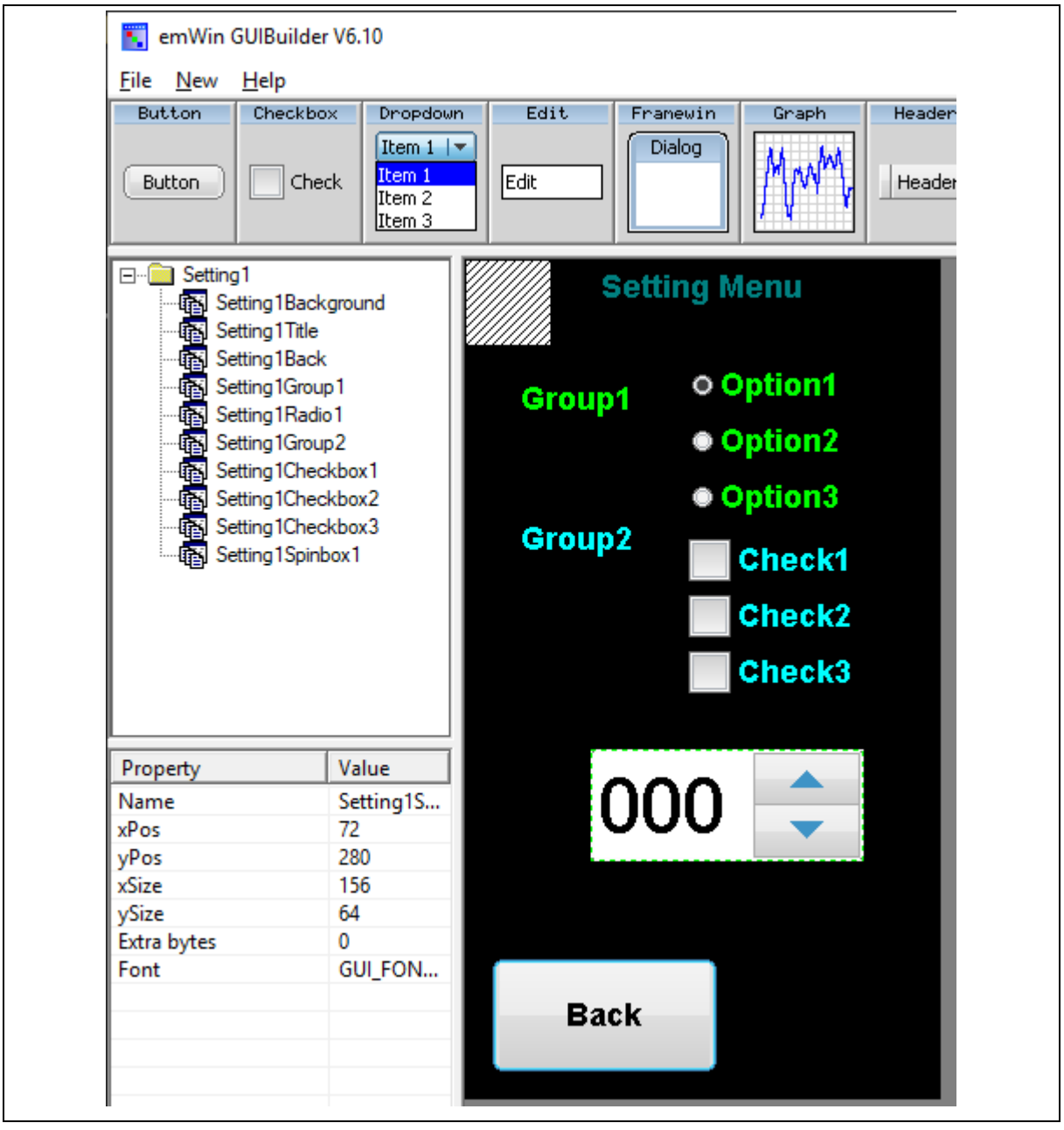

Figure 5-3 Setting Menu Control

List of all setting menu control definitions by the Freezer HMI.

By "*NVT_Setting1.c*", you can assign user define dta image file to replace the original.

```
NVT_Setting1.c
```

```
//
// Freezer HMI setting menu background
//
#define NVT_SETTING_BACKGROUND "D:\\setting_background.jpg"
//
// Setting menu button image
//
#define NVT_SETTING_BUTTON1 "D:\\setting_back.dta"
//
// Increase DTA buffer size if needed
//
static UINT8 s_u8BackgroundImageBuf[200 * 1024]
__attribute__((aligned(32)));
static UINT8 s_u8Button1ImageBuf[26 * 1024] __attribute__((aligned(32)));
```

6 FAQ

6.1 How to replace dta?

Use the same image filename, width and height and utilize 9To565 or 28ToA565 to convert to dta then copy to NAND1-2.

6.2 How to replace background jpeg?

Use the same jpeg filename, width and height, then copy to NAND1-2.

Note H/W JPEG decoder supports baseline profile only.

7 REVISION HISTORY

Date	Revision	Description
2020.11.30	1.00	1. Initially release.
2020.12.17	1.01	1. Updated images and increased images buffer size.

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.