Arm® 926-EJS

32-bit Microcontroller

N9H26
Cooking HMI
User Manual

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 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

Table of Contents

1 OVERVIEW

Cooking HMI for N9H26 Linux is a GUI reference implementation.

This document utilizes Nuvoton N9H26 series geneal-purpose microprcessor N9H26K6 (64MB DDR2) to implement cooking HMI with emWin GUI library. Cooking HMI supports H.264 H/W decode.

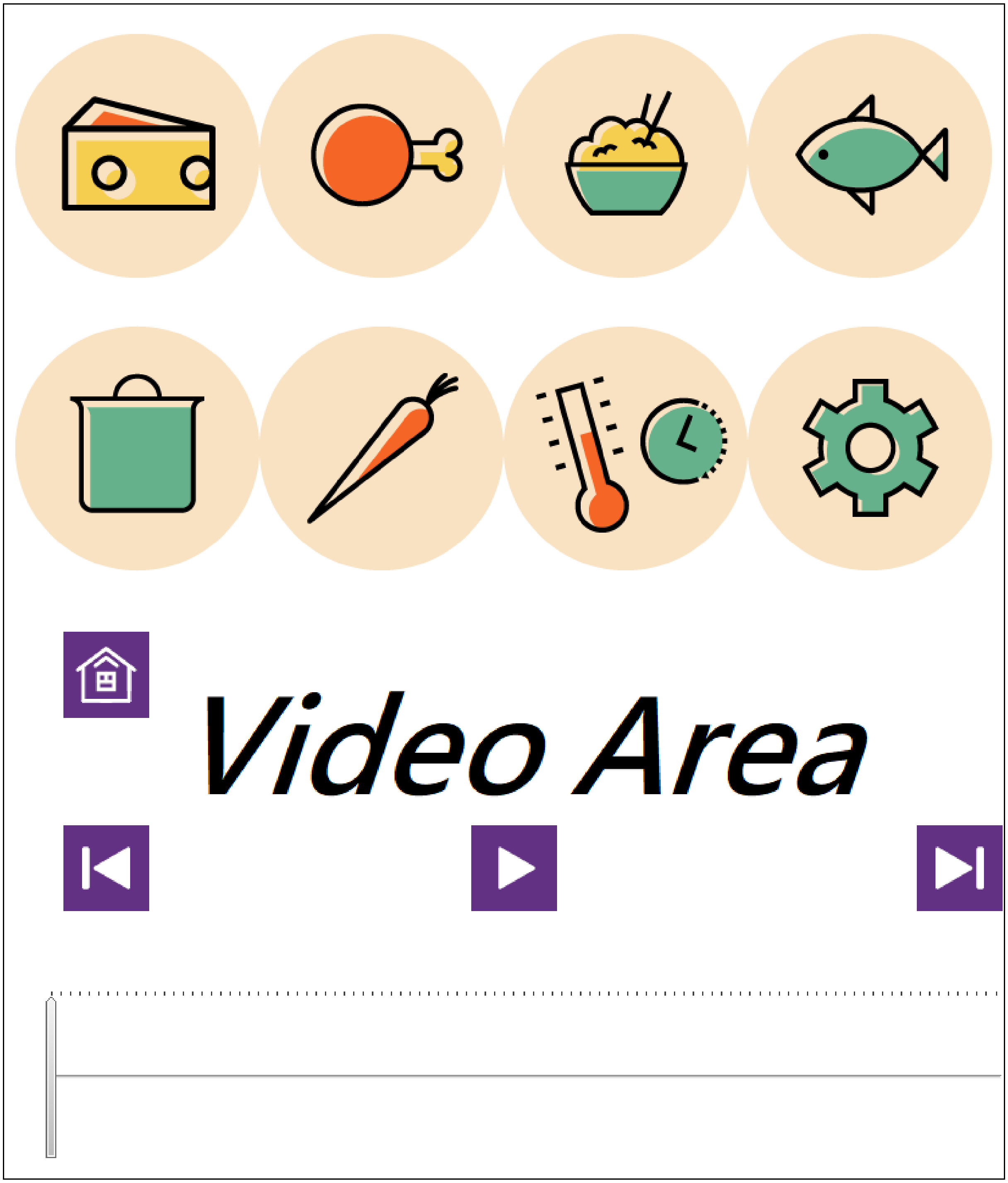
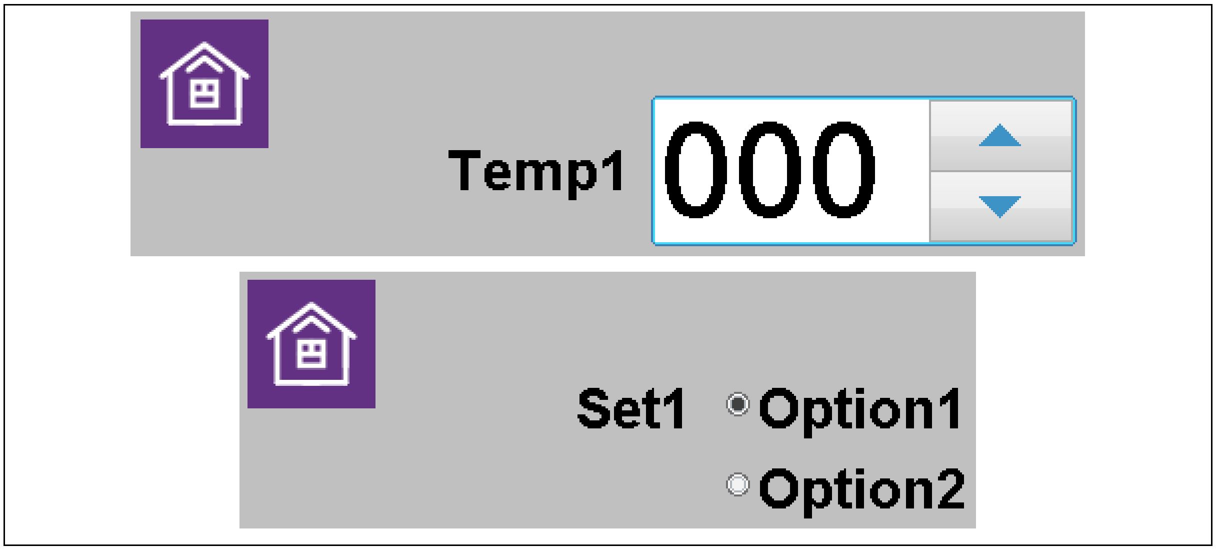


Figure 1-1 Cooking HMI, main menu and video area.

2 FEATURES

2.1 Cooking HMI Features

- Supports SEGGER licensed emWin GUI library
- Supports Linux for extension open sources
- Supports hardware H.264 decoder for baseline decoding
- Supports resistive touch up to 800 x 480 area with built-in touch ADC
- Supports high quality and contrast LCD panel with resolution up to 800 x 480
- Supports many popular image formats, e. g., PNG, GIF, JPG and BMP
- Supports various audio decoder and up to 48000Hz stereo 16-bit format



Temp1 000

Set1 ◉ Option1 ○ Option2

Figure 2-1 Typical setting for cooking (Temperature and other options)

3 INSTALLATION AND ENVIRONMENT

3.1 Installing N9H26

First, download the latest N9H26 Linux BSP from https://github.com/OpenNuvoton/N32926_Linux_BSP, and unzip "*N32926_Linux_BSP-master.zip*" to a working folder, e.g., unzip it to the path "*/home/n9h26*", where "*N9H26*" is the working folder.

The detailed information of N9H26 Linux BSP and emWin library can be found at "*N329 Series MCU Linux 2.6.35 BSP User's Manual.pdf*" and "*emWinStartGuide-N9H26-Series.pdf*" respectively.

3.2 Installing Cooking HMI

First, download the latest "*N9H_emWin_Template-master.zip*" from https://github.com/OpenNuvoton/N9H_emWin_Template and unzip and copy "*Cooking_N9H26_Linux*" to the Linux working folder.

Then, type "*make*" to build the application, called "*CookingDemo*". Next, connect the USB cable between PC/NB and N9H26 and keep pressing UP + DN key and power on. Then copy "*CookingDemo*" to "SD1-1" USB disk. Finally, remove the USB disk safely and reboot N9H26.

3.3 System Requirements

- Linux environment, please refer to N329 Series MCU Linux 2.6.35 BSP User's Manual.pdf
- Nuvoton N9H26K6 800 x 480 demo board (NuDesign HMI-N9H26 + NuDesign TFT-LCD5)

4 FOLDER STRUCTURE

4.1 Code Folder Structure

The content of “*Cooking_N9H26_Linux*” is described as follows.

Folder	Description
CookingDemo	Base folder ● main.c is cooking demo code and platform related initilizations ● Makefile contains the definition “*__GUI_NUMBYTES__*“ to define emWin memory pool size (suggest 2~4MB for the application) ● Changelog.pdf is version history ● Cooking_Reference_Implementation.pdf is user manual ● N329 Series MCU Linux 2.6.35 BSP User's Manual.pdf is introduction and installation of Linux environment and tool-chain
Application	emWin resource folder ● WindowDLG.c is video area control menu ● Window_CookingDLG.c is cooking main menu ● Window_TempDLG.c is temperature setting menu ● Window_SettingDLG.c is option setting menu
tslib	Touch folder ● Resistive touch panel ● Touch area is 800 x 480
Res / Image	Icon folder ● Cooking main menu and video control (didn’t convert to DTA format)
Res / SD1-1	SD card folder ● Default video is company advertisement at beginning playback ● Cooking main menu icons and video control icons ● conprog.bin is Linux 2.6.35.4 kernel ● mplayer is media player for video and audio playback ● CookingDemo is main application
Res / Windows_Tool	Windows toolfolder ● HMI_Image_20200715_CS.7z converts image to DTA file format ● HMI_Video_20200715_CS.7z converts video to H.264 MP4 ● HMI_Video_20200729_CS.7z converts video to MJPEG AVI

Table 4-1 Cooking HMI Folder Structure

4.2 Image and Video Resource Folder Structure

The “SD1-1” folder contains image and video.

Folder	Description
01_dessert.dta	Cooking main menu ● emWin bitmap RGB565 format ● 200 x 240 dimension
02_meat.dta	
03_rice.dta	
04_seafood.dta	
05_soup.dta	
06_vegetable.dta	
07_temp_time.dta	
08_setting.dta	
Home_01.dta	Video area control menu ● emWin bitmap RGB565 format ● 64 x 64 dimension
Play_01.dta / Pause_01.dta	
_movie.avi	Company advertisement video (playback at the begin stage)
DCIM / videos	Cooking videos mapping to 01_dessert ~ 06_vegerable topics

Table 4-2 Cooking HMI Images and Video Folder Structure

5 DESIGN GUIDE

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

- Linux programming
- The C programming language, how to use arm-linux compiler
- The N9H26 IPs programming knowledge
- The basic emWin programming knowledge

5.1 Video and emWin Display Buffer Control

The N9H26 utilizes hardware H.264 decoder to decode to frame buffer layer. emWin utilizes hardware OSD (On Screen Display) layer for GUI. Then, N9H26 combines these two layers and outputs to LCD dispaly.

```
// Open frame buffer layer for H.264 display

open("/dev/fb0", O_RDWR);



// Allocate frame buffer + OSD layer address

pVideoBuffer = mmap(NULL, uVideoSize, PROT_READ|PROT_WRITE, MAP_SHARED,
fd, 0);



// Get OSD layer address for emWin GUI display

ioctl(fd, IOCTL_GET_OSD_OFFSET, &osd_offset);

pVideoBuffer = pVideoBuffer + osd_offset;
```

5.2 Video Play Control

The N9H26 utilizes hardware video decoder to decode H.264 and MJPEG, respectively. First, to create fifo to communicate between mplayer and emWin. (larger FPS video may drop performance)

In cooking main menu, you can select up to 6 cooking video and playback.

```
#define    MPLAYER_FIFO_PATH       "/tmp/fifo_mplayer_cmd"

//

// Create FIFO

//

mkfifo( MPLAYER_FIFO_PATH, 0777 );



#define NVT_BEGIN_MOVIE "_movie.avi"

#define NVT_BEGIN_HIDE 5000 // millisecond
```

```
//

// sBuf contain necessary command to let mplayer play a video and process
fifo commands comes from emWin

//

snprintf(sBuf, sizeof(sBuf), "./mplayer -noaspect -loop 0 -idle -quiet -
osdlevel 0 -fs -vendor-osdcount 0 -vendor-osdshmid 12345 -vendor-
osdshmsize 1024 -slave -vendor-framedrop 1 -input
file=/tmp/fifo_mplayer_cmd %s &", NVT_BEGIN_MOVIE);

system(sBuf);
```

Note: Video controller can play, pause, change time position or back to cooking main menu.

```
void NVT_PlayFile(const char * FileName)

{

    snprintf(sBuf, sizeof(sBuf), "echo loadfile %s >
/tmp/fifo_mplayer_cmd &", FileName);

    s_stream = popen( sBuf, "rw" );

    pclose( s_stream );

}



void NVT_GetPos(U32 Pos)

{

    snprintf(sBuf, sizeof(sBuf), "echo pausing_keep seek %d 1 >
/tmp/fifo_mplayer_cmd", Pos);

    s_stream = popen( sBuf, "rw" );

    pclose( s_stream );

}



void NVT_Home(void)

{

    s_stream = popen( "echo stop > /tmp/fifo_mplayer_cmd", "rw" );

    pclose( s_stream );

}
```

N9H26K6 USER MANUAL

```
void NVT_PlayPause(void)
{
    s_stream = popen( "echo p > /tmp/fifo_mplayer_cmd", "rw" );
    pclose( s_stream );
}
```

5.3 Background Control

The N9H26 utilizes emWIn API to clear background after switching between menu.

```
//
// Enable memory device to try to avoid flicker
//
WM_SetCreateFlags(WM_CF_MEMDEV);
//
// emWin init
//
GUI_Init();
//
// Register a call back function to clear background if needed
//
WM_SetCallback(WM_HBKWIN, cbBackgroundWin);
```

Background callback function:

```
static void cbBackgroundWin(WM_MESSAGE *pMsg)
{
    switch (pMsg->MsgId)
    {
        case WM_TOUCH: // When touch, show video control
            t = GUI_GetTime();
            WM_ShowWin(g_hWinPlay);
            break;
```

N9H26K6 USER MANUAL

```
       case WM_PAINT: // Draw color key as transparent for OSD
          GUI_SetBkColor(GUI_MAGENTA);
          GUI_Clear();
          break;
   }
}
WM_DefaultProc(pMsg);
}
```

6 FAQ

6.1 How to replace image?

Use the same image filename, width and height, then copy to related folder, please see the chapter 4.2.

Note: To utilize DTA file format, you need to use HMI_Image.

6.2 How to replace video?

Use the same video filename, then copy to related folder, please see the chapter 4.2.

Note: To utilize video, you need to use HMI_H264 or HMI_MJPEG.

6.3 Why the video looks so slow and laggy?

It is caused by softwwre decoder for audio in AAC higher sampling rate or video hardware H.264 decode in higher FPS. You can use MP3 at 22050 mono channel or video FPS at 15~20, respectively.

7 REVISION HISTORY

Date	Revision	Description
2020.07.29	1.00	1. Initially release.

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.