

Quick Start for N32901-3 Non-OS IP V1.00.004

Publication Release Date: Dec. 2012

Support Chips:

W55VA/FA Series

Support Platforms:

Non-OS

The information in this document is subject to change without notice.

The Nuvoton Technology Corp. shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.

This documentation may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from the Nuvoton Technology Corp.

Nuvoton Technology Corp. All rights reserved.

Table of Contents

| | |
|---------------------------------|-----------|
| 1. Introduction | 4 |
| 1.1. What is the Document | 4 |
| 1.2. Limitation | 4 |
| 2. IP sample code | 8 |
| AVI..... | 8 |
| GPIO..... | 9 |
| I2C..... | 10 |
| JPEG..... | 11 |
| KPI | 14 |
| SIC..... | 14 |
| SIC_SECC..... | 15 |
| SPI..... | 15 |
| SPU..... | 16 |
| SYSLIB | 17 |
| UDC | 20 |
| USB Core | 21 |
| Video In..... | 23 |
| VPOST | 24 |
| ExtNandWriter | 24 |
| NandWriter | 25 |
| SDWriter | 25 |
| NVTFAT | 25 |
| 3. Revision History..... | 27 |

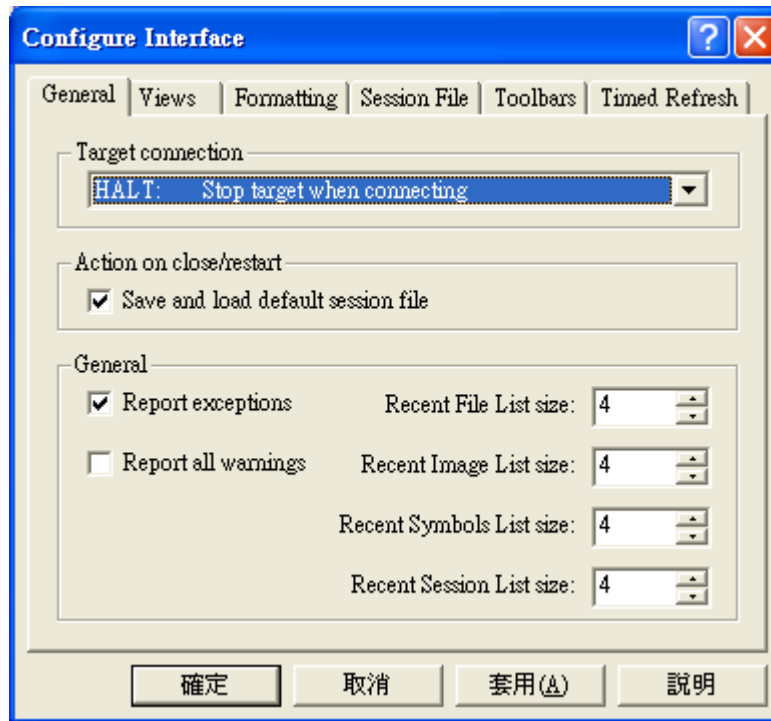
1. Introduction

1.1. What is the Document

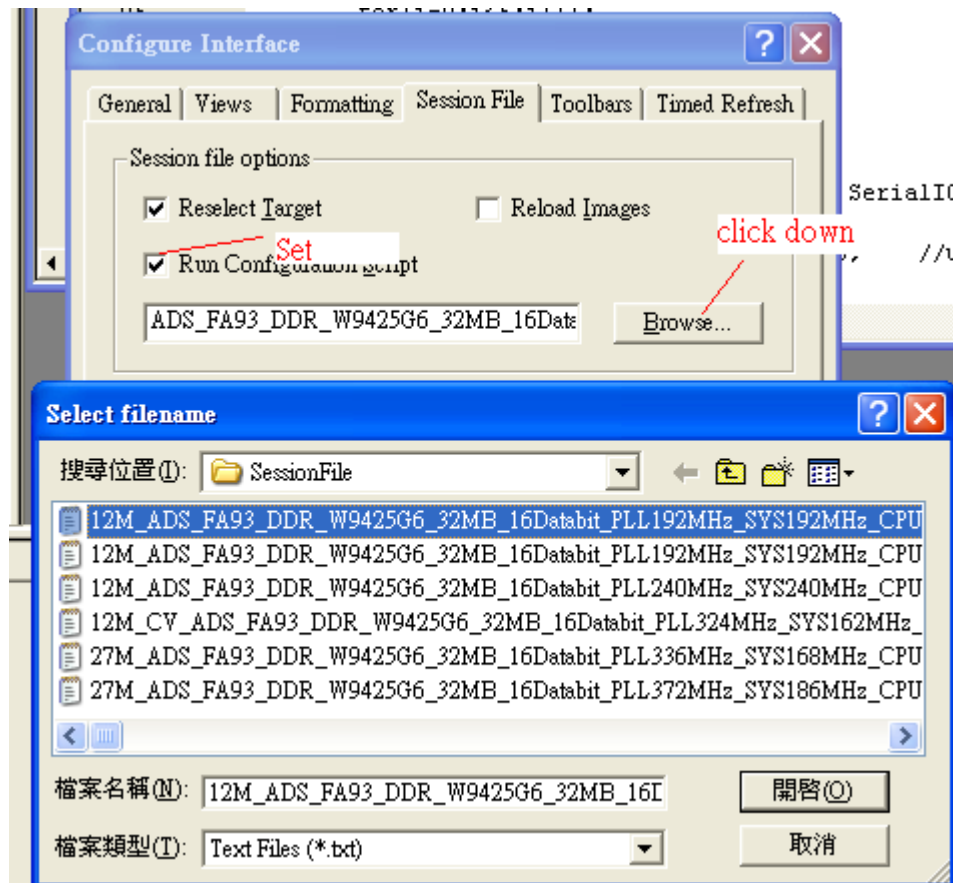
We use the sample code from Non-OS to test FA93 IP. There are some IP documents to introduce his function. But it is not enough to handle with running the sample code. There is no any document to introduce how to work the sample code. We will not know how to work. Therefore we will introduce the processing method for sample code in the document.

1.2. Limitation

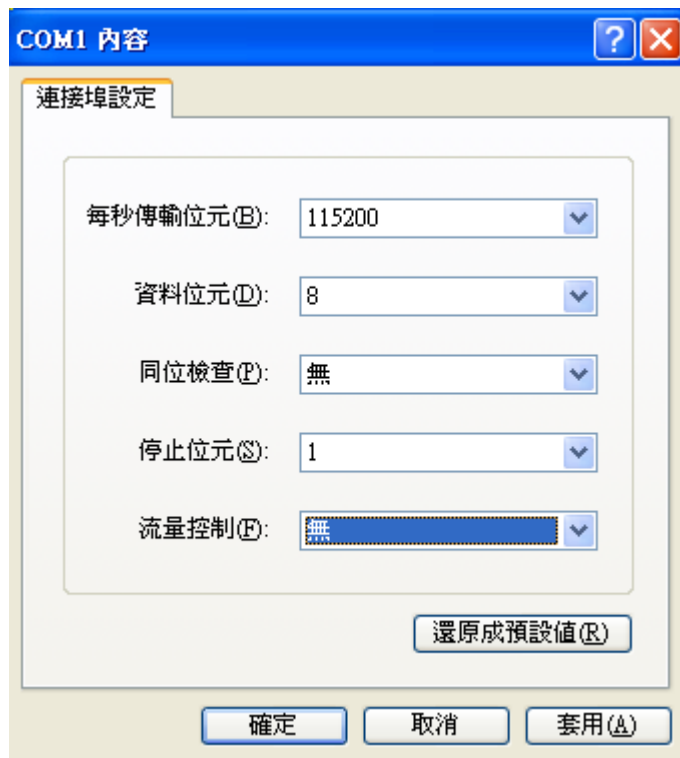
Current sample code from Non-OS uses ARM Developer Suite v1.2 to build. Users could use Multi-ICE or J-Link to work ICE emulation. Firstly running AxD, user should set the session file to press the menu Options -> Configure Interface..., the Dialog will show as follows.



User selects the item “Session File” to run the configuration script. For N32903, we could use the file “QDN_12M_ADS_FA93_DDR_W9425G6_SPG75_8MB_16DataBit_PLL192MHz_SYS192MHz_CP U192MHz_MCLK96MHz.txt” for demo board.



Later on user needs to create UART status by using the hyper terminal as follows. User could download the files hypertrm.exe and hypertrm.dll from the web and runs it.



The baud rate of UART is 115200. While any sample code is running, user could see the UART status for the execution status as follows.



The build options are N3290X_YYYYYYY, X is 1 or 3 . Currently we only support N32901 and N32903, so the build option has N32903_ and N32901_ to be the prefix.

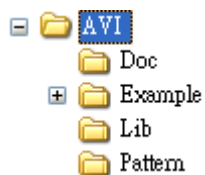
2. IP sample code

In this chapter, we will process the sample code of FA93 IP. For most of the sample code, we use the demo board to test as follows.



AVI

The folder structure is shown as follows



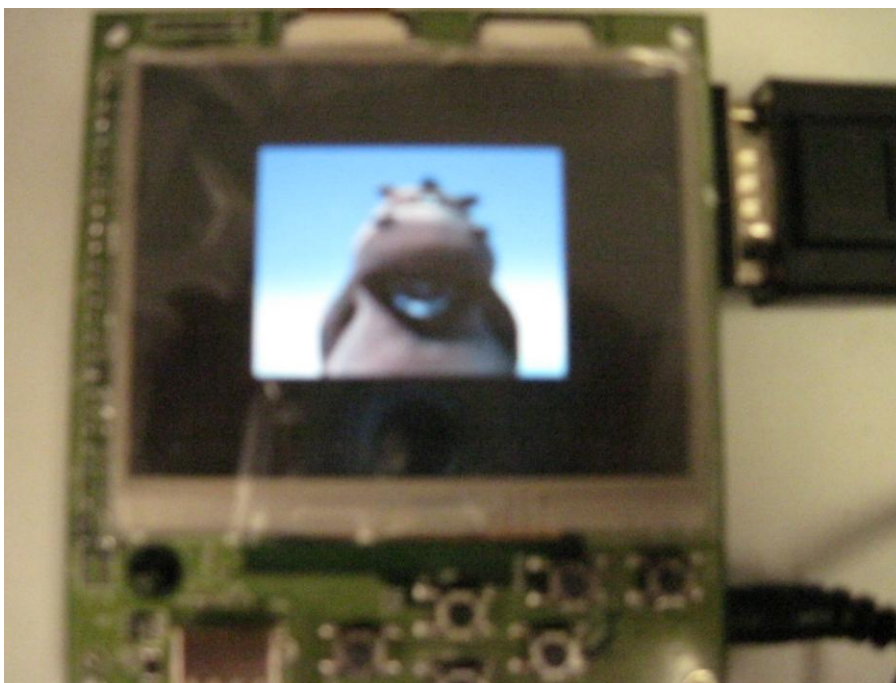
The function uses the demo board to test.

Firstly user makes sure the kind of VPOST. User could open the project W55FA93_VPOST.mcp within the subfolder src of the folder VPOST, and see the file W55FA93_VPOST.h within the subfolder lib. User should make sure the kind of LCD. Currently we use the specified LCD for N32905 demo board. So we open the definition “#define HAVE_GIANTPLUS_GPM1006D0”,

and comment the other definitions. And later build the project W55FA93_VPOST.mcp to obtain the new library W55FA93_VPOST_GPM1006D0.a.

User also opens the project AVIPlayer.mcp within the subfolder Example of folder AVI, and build in the N32903_Release mode. Before running it in the demo board, user should use the one AVI file named 480x272.avi within the subfolder pattern, including MJPEG and IMAADPCM. Put the file into SD card, and SD card plugs in SD card 0.

Running the built image, we could see the result as follows.



If no image displays, please confirm the library W55FA93_VPost_GPM1006D0.a to use the kind of LCD type and is included in build mode. User could open the project W55FA93_VPOST.MCP within the subfolder Src of folder VPOST, and change the LCD setting of the file W55FA93_VPOST.h within subfolder Lib. And rebuild to export the new library W55FA93_VPOST_GPM1006D0.a

AVI uses the standard library API malloc and free to work. If user would like to the user defined method, then we need to work the following. The file wb_mem.c of the project SYSLIB provides two user defined functions nv_malloc and nv_free. User could modify the functions to meet his request. The default of the function is malloc and free. Later on within the subfolder LIB of the project GNAND, rename the library W55FA93_gnand_nv.a into W33FA93_gnand.a to replace the old. Within the subfolder LIB of project NVT FAT, rename the library W55FA93_NVT FAT_nv.a into W55FA93_NVT FAT.a to replace the old. Within the subfolder LIB of project AVI, rename the library mp3lib_nv.a into mp3lib.a to replace the old. And then rebuild the project AVI.

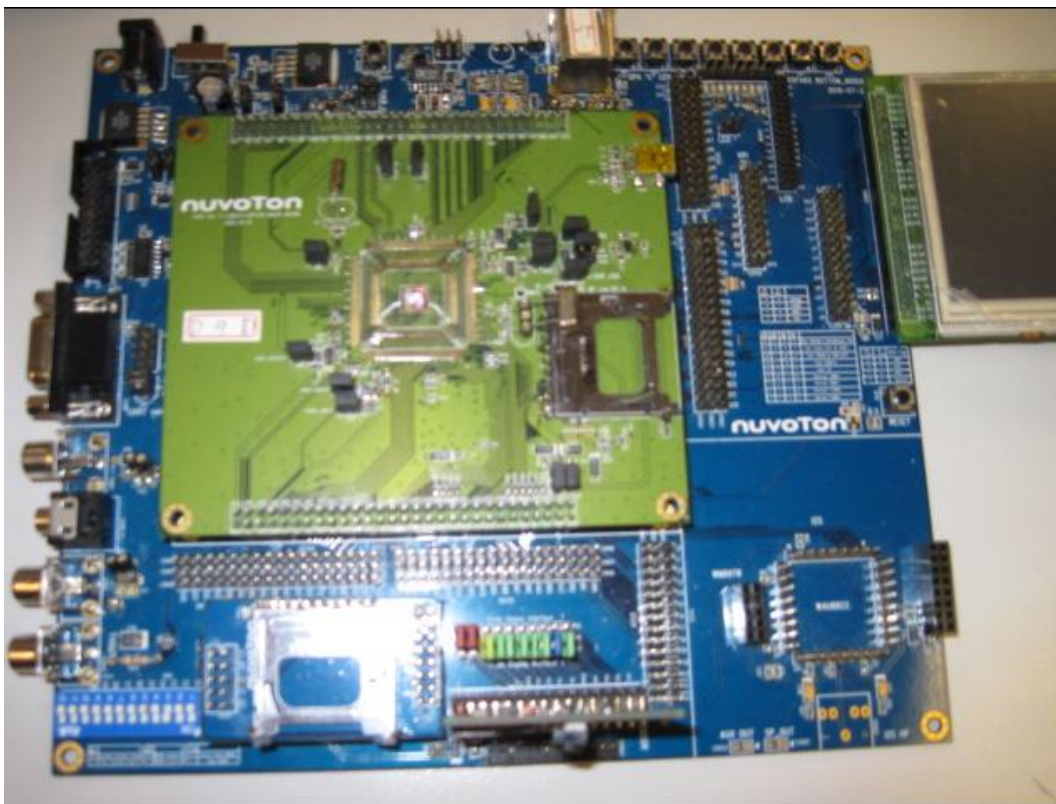
The two build modes N32901_ and N32903_ are supported.

GPIO

The folder structure is shown as follows.



The function uses EVB board to test, EVB board is shown as follows.

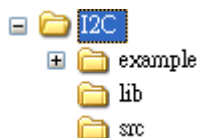


Firstly user should use GPIO board, and plugs it into GPIOB on EVB. User should build the project GPIO.mcp within the subfolder src to export one library. And then build the project gpio.mcp within the subfolder example. Later on running the built image, the LED of GPIO board will flush. The sample code works GPIOB for LED flush.

The two build modes N32901_ and N32903_ are supported.

I2C

The folder structure is shown as follows.



The function uses the demo board.

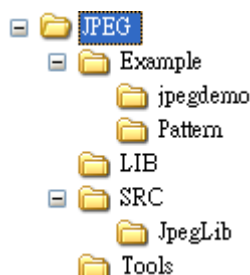
Firstly User should build the project i2c.mcp within the subfolder src to export one library. And then build the project demo.mcp within the subfolder example\demo. Later on running the built image, user could see UART status to check the result. For the sensor OV7725, the UART status should be shown as follows.

```
I2C demo read OV7670 sensor ID...
Sensor ID0 = 0x76
Sensor ID0 = 0x73
Sensor ID0 = 0x7F
Sensor ID0 = 0xA2
Sensor ID0 = 0xA2
Sensor ID0 = 0x80
```

The two build modes N32901_ and N32903_ are supported.

JPEG

The folder structure is shown as follows.



The function uses the demo board.

Firstly User puts the files within the subfolder example\pattern into one SD card, and put the SD card into SD card 0. Build the project JpegLib.mcp within the subfolder src\JpegLib to export one library. And then build the project fa93_jpegdemo.mcp within the subfolder example\jpegdemo. Later on running the built image, user could see UART status to check the result as follows.

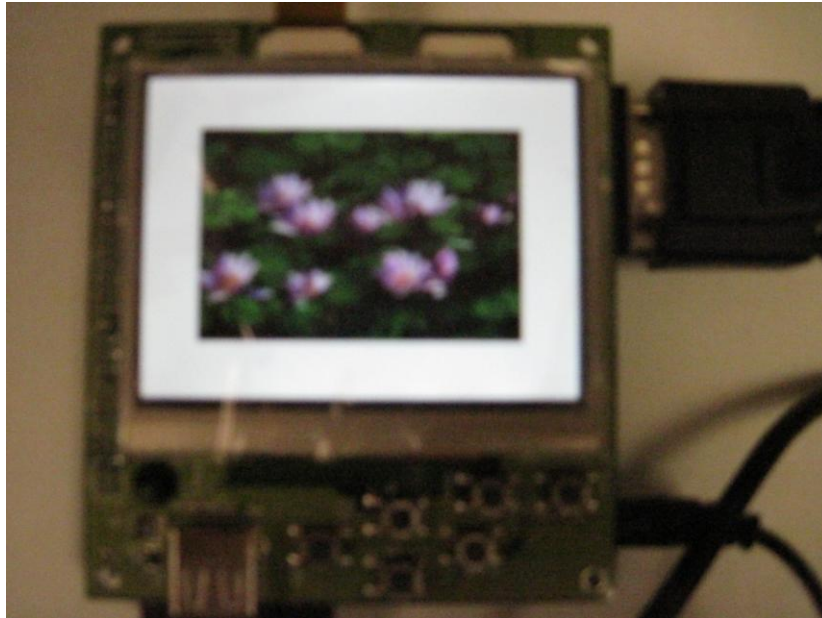
```

~~~~~
/*-----
/*  JPEG Demo code
/*-----
fsInitFileSyste

Please Select Test Item
[*]Decode Test
0 : Panel Test Enable
   -> Decode Downscale to QVGA
   -> Decode Stride is 320
   -> Output data size is 320x240
1 : Input Wait Enable
2 : Set Deocode output format
3 : Start to Decode
   -> Decode output format is PACKET YUV422
[*]Encode Test
4 : Upscale Enable
5 : Software Reserved Enable
6 : Set Encode Width & Height
7 : Start to Encode
   -> Encode Size 640x480
8 : Exit
>
    
```

For decoder, select the option 0 to enable the panel. And the decoded file will display one the panel. Select the option 2 to set the decode output format to be "PACKET RGB888", select the option 3 to decode the specified file jpegdec.jpg. It will export one DAT file on SD card, user

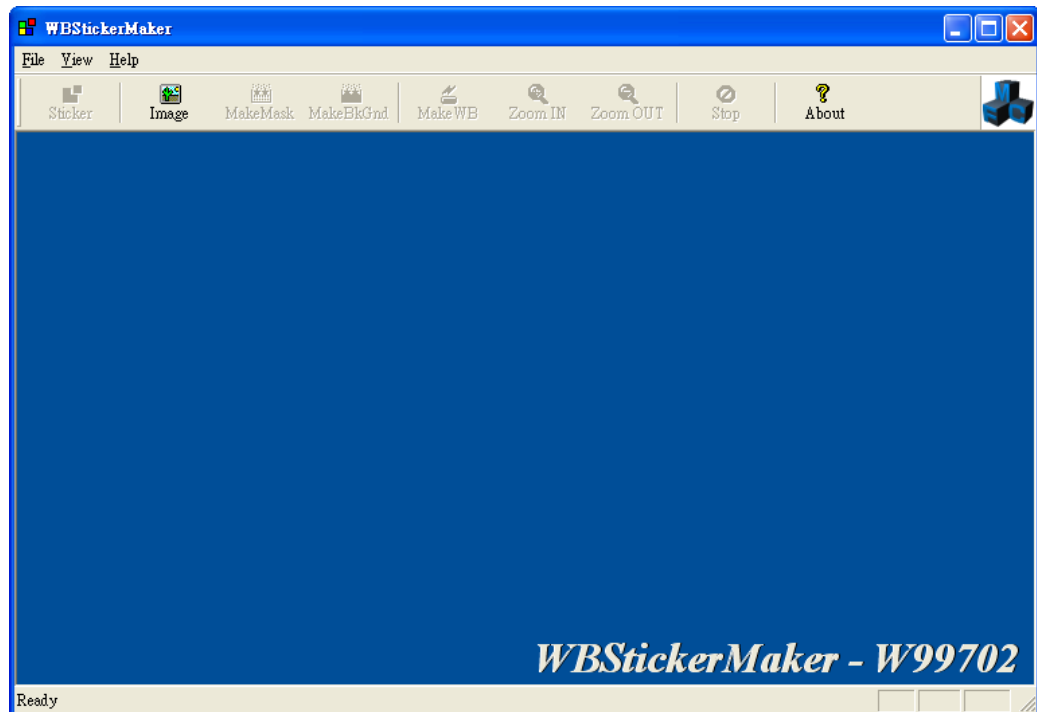
could use tool WBStickerMaker to see the result. Sometimes the JPEG file display on LCD of the demo board.



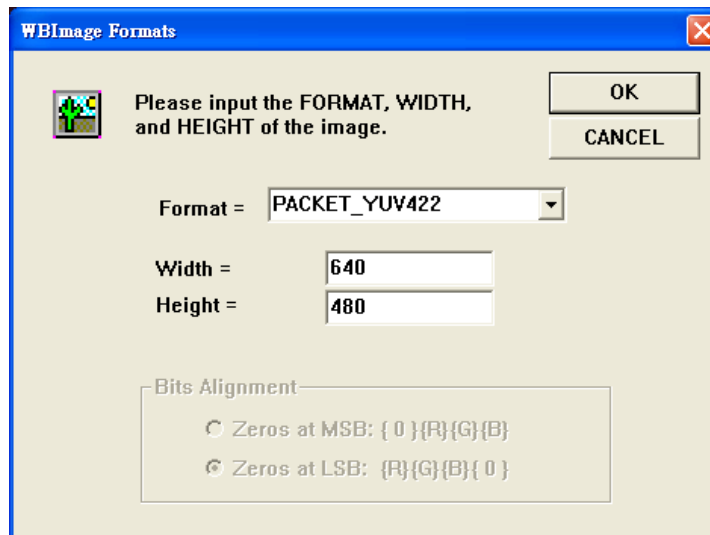
For encoder, user should input the one file with the YUV422 packet format. There is one pattern jpegenc.dat within the subfolder pattern. Select the option 7 to input the file to encode, and later one file jpegenc.jpg is created on SD card. User could use card reader to see the result.

If user would like to encode the other files, we propose one tool WBStickerMaker within the subfolder Tool to export one YUV422 packet format. We describe the method in the following.

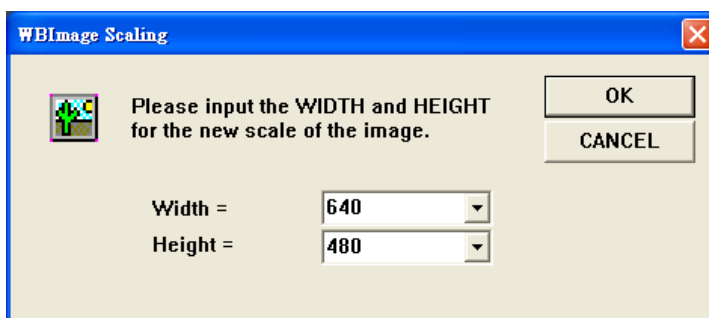
Run WBStickerMaker as follows,



And open one file by running the menu File -> Open Image File..., And the Click the menu Controls -> Color (Image) Format to set the image attribute as follows.



Click OK button to take effect. Click the menu Controls->Scaling to set the resolution if necessary as follows.

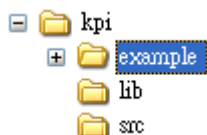


Run the menu Controls -> Make Sticker Background file (*.dat) to export the specified file with YUV422 packet format. Then the file could be encoded by the sample code.

The two build modes N32901_ and N32903_ are supported.

KPI

The folder is shown as follows



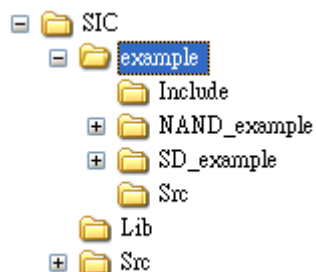
The function uses the demo board.

Firstly the demo board of N32903 has 6 keys. User should open the statement “#define CONFIG_FA93_2X3_KPI” within the file libkpi.c of subfolder src. And then build the kpi.mcp within the subfolder src to export one library. Build the project kpi.mcp within the subfolder example. Later on running the built image, pressing down/up the 6 keypads on the demo board. User could see UART status to check the result.

The two build modes N32901_ and N32903_ are supported.

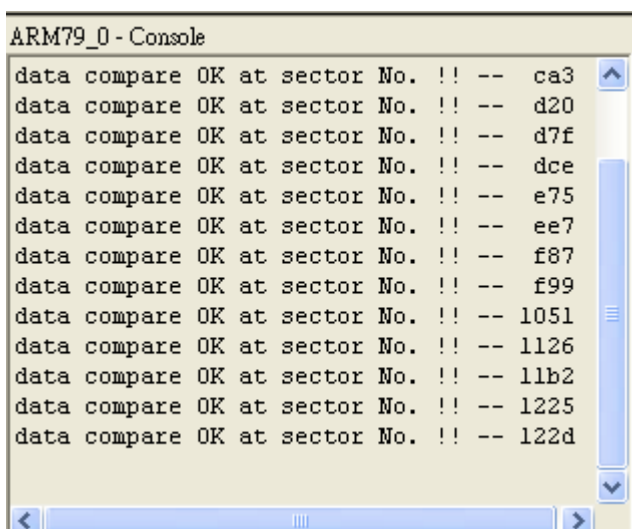
SIC

The folder is shown as follows



The function uses the demo board

Firstly user should build the dev_sic.mcp within the subfolder src to export one library. And then build the project NAND_example.mcp within the subfolder example\NAND_example. Later on running the built image, user could see the console status of AxD as follows.

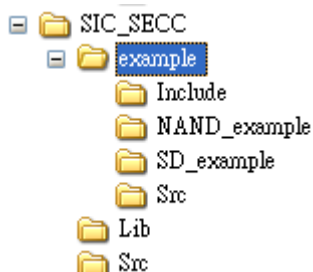


User also builds the project SD_example.mcp within the subfolder example\SD_example and runs the built image and sees the result on console of AxD.

The two build modes N32901_ and N32903_ are supported.

SIC_SECC

The folder is shown as follows.



The function uses the demo board.

Firstly user should build the project dev_sic.mcp within the subfolder src to export one library. And then build the project SD_example.mcp within the subfolder example\SD_example. Later on running the built image, user could see the console status of AxD. NAND_example can also be built, run, and see the console status of AxD.

The two build modes N32901_ and N32903_ are supported.

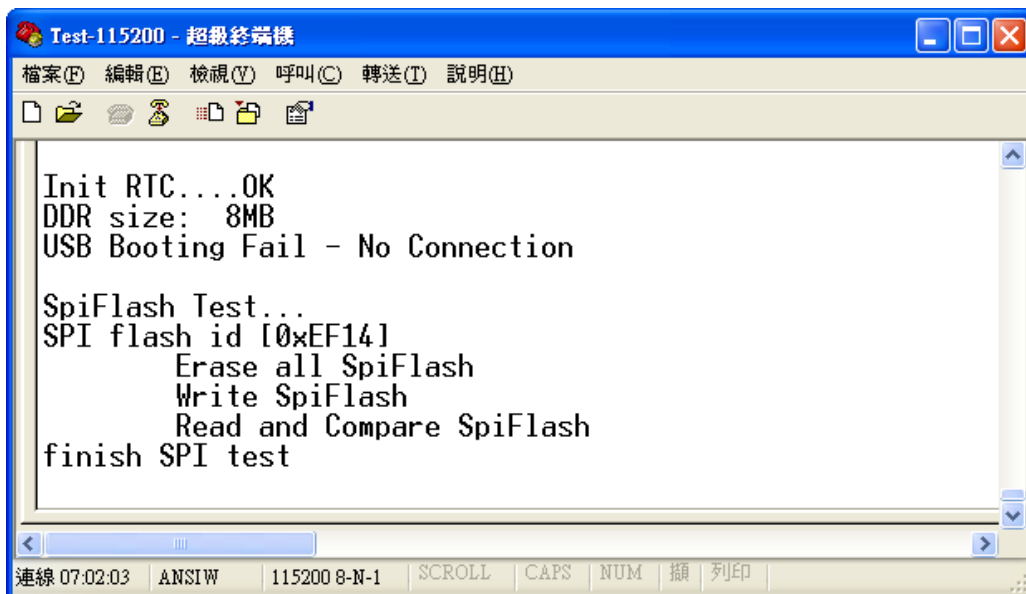
SPI

The folder is shown as follows.



The function uses the demo board.

Firstly user should build the project dev_spi.mcp within the subfolder Src to export one library. And then build the project SpiFlash.mcp within the subfolder example\SpiFlash. Running the built image, and user could see the UART status as follows.



```

Init RTC...OK
DDR size: 8MB
USB Booting Fail - No Connection

SpiFlash Test...
SPI flash id [0xEF14]
    Erase all SpiFlash
    Write SpiFlash
    Read and Compare SpiFlash
finish SPI test
  
```

The two build modes N32901_ and N32903_ are supported.

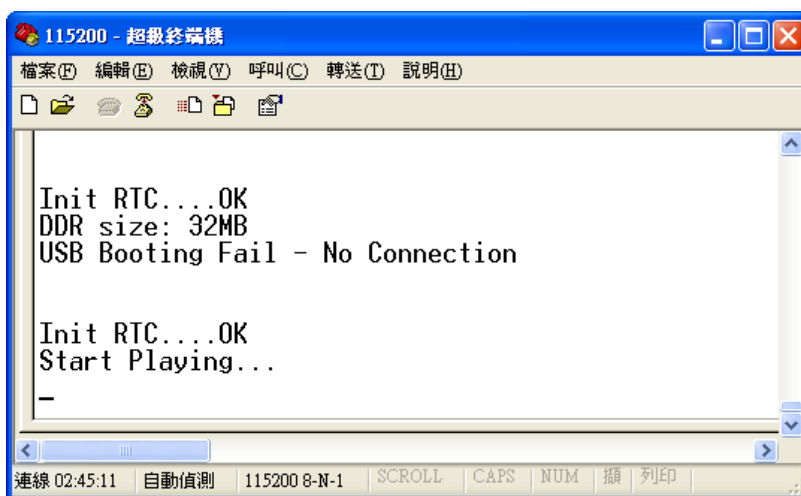
SPU

The folder is shown as follows.



The function uses the demo board.

Firstly user should build the project spu.mcp within the subfolder Src to export one library. And then build the project spudemo.mcp within the subfolder example. Later on running the built image, user could see the UART status as follows.

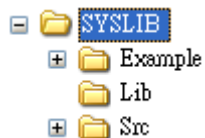


Then user could use speaker to plug in the demo board, and listen to the music from DAC of demo board.

The two build modes N32901_ and N32903_ are supported.

SYSLIB

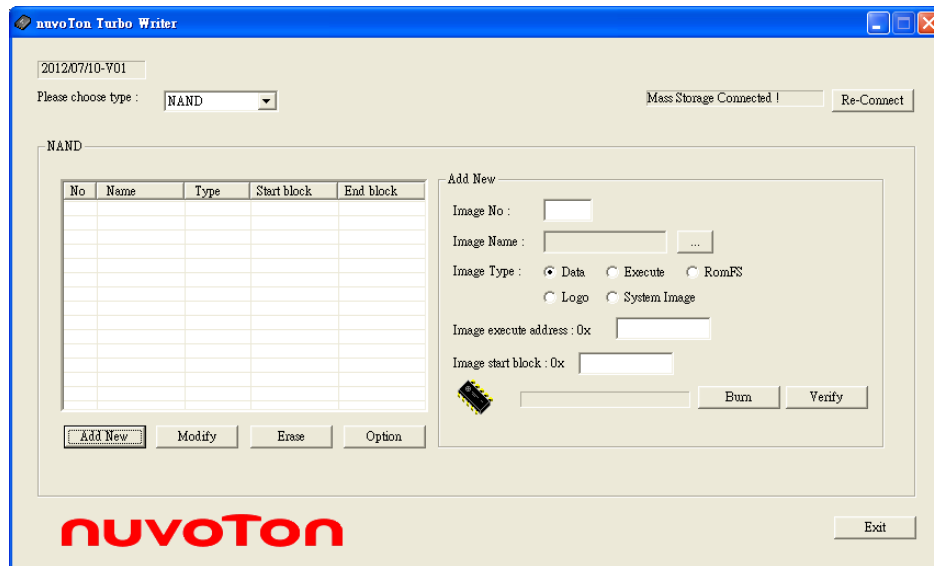
The folder is shown as follows



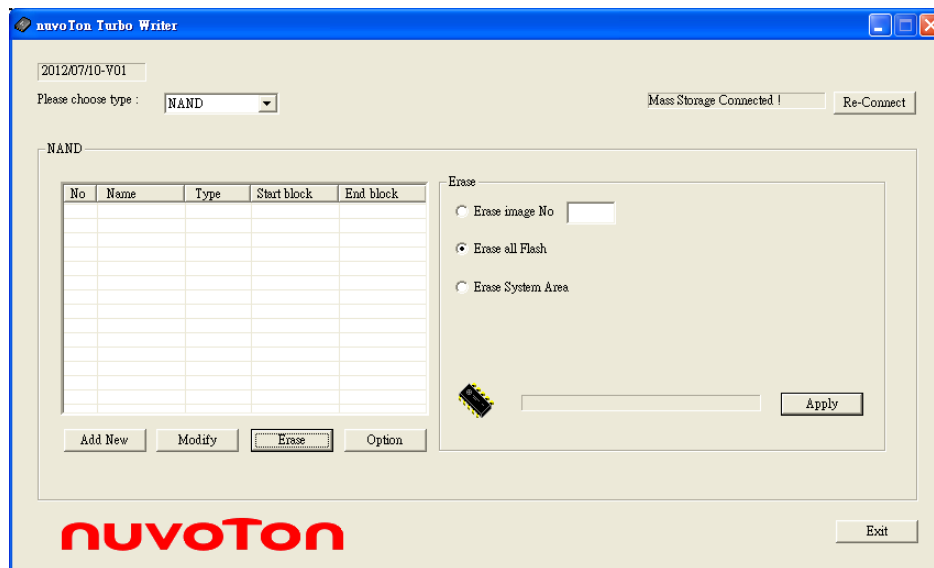
The function uses the demo board.

Firstly User should build the project syslib.mcp within the subfolder Src\syslib to export one library. And then build the project demo.mcp within the subfolder example\demo. The built binary file CK_Chg_D.bin could be burned into NAND by using TurboWriter.

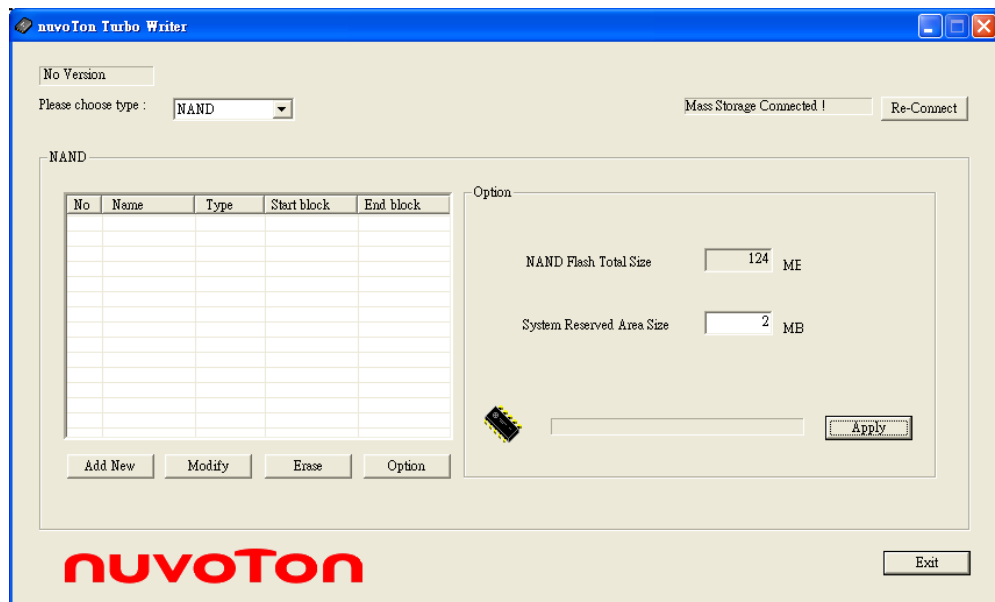
How to burn the binary file CK_Chg_D.bin into NAND of demo board? Firstly set the demo board into recovery mode. Run TurboWriter as follows.



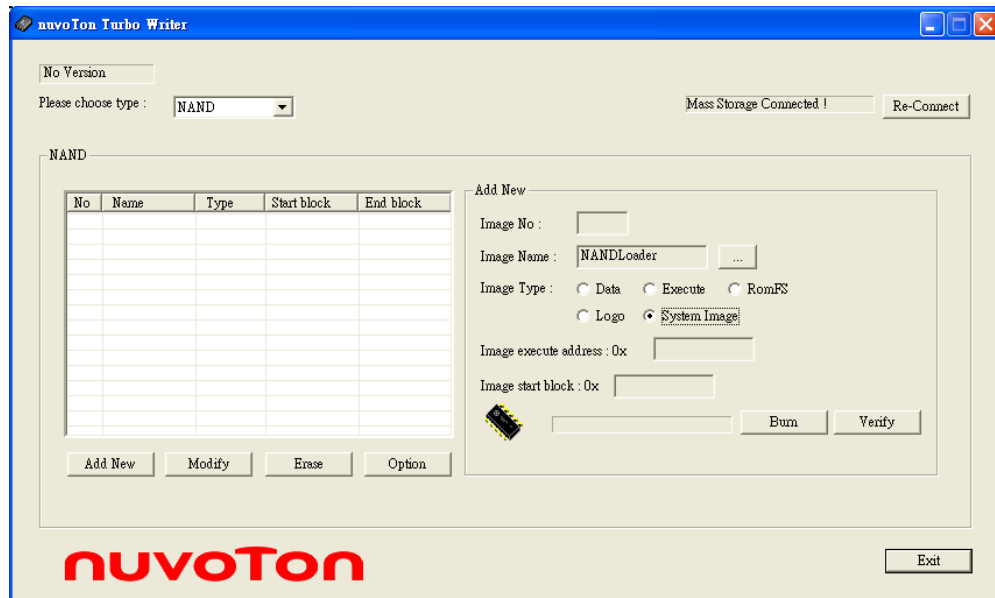
Click button Erase to run “Erase all Flash” as follows



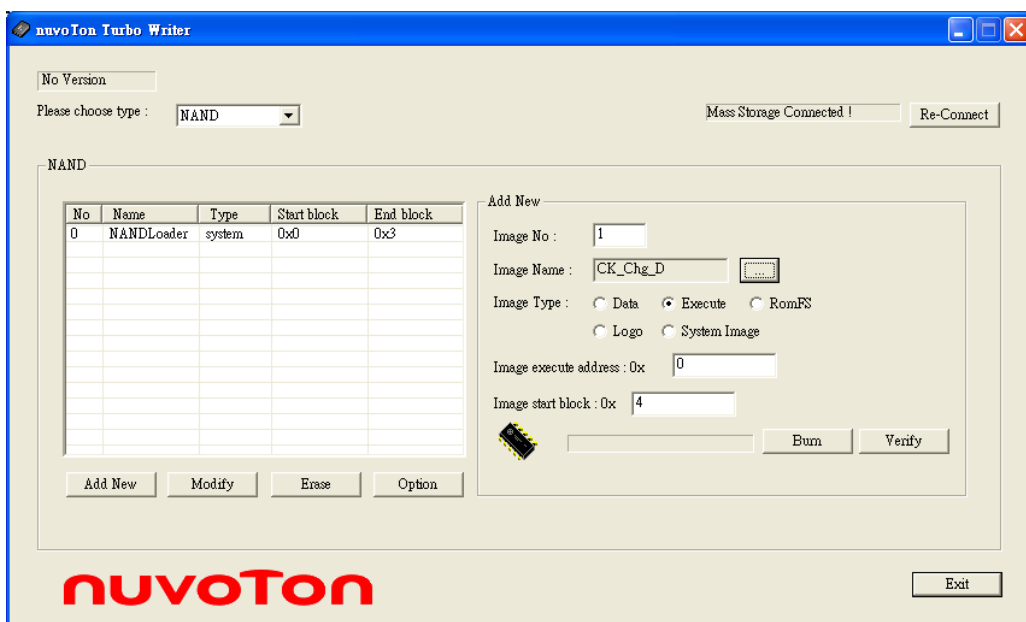
Click button Option to set “System Reserved Area Size” to be 2 MB and click button Apply as follows



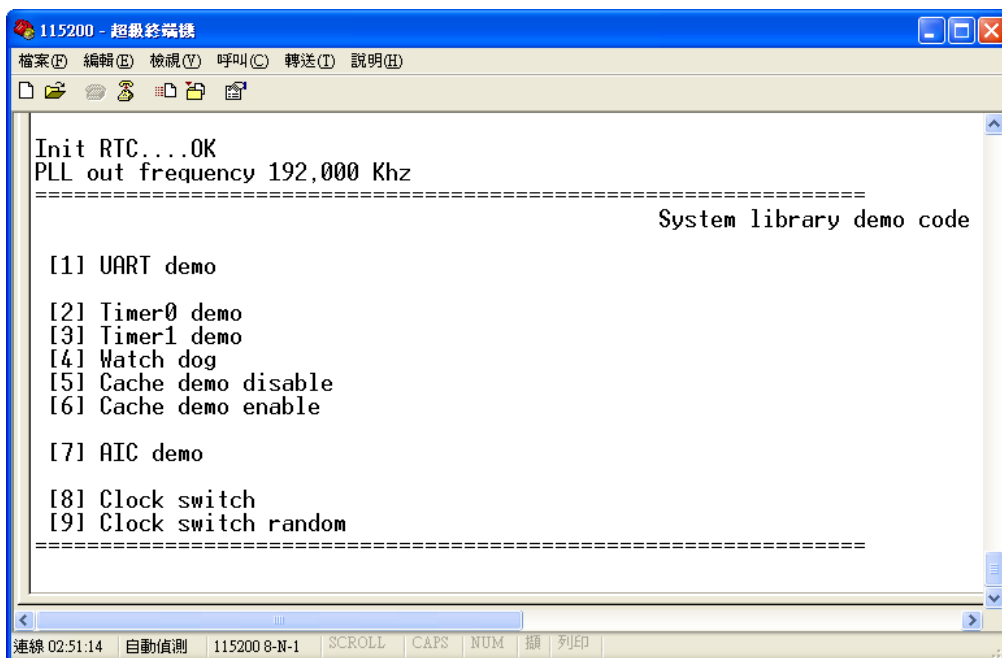
Click button “Add New” to burn the file “NandLoader.bin” as follows.



Burn the file Ck_Chg_D.bin as follows.



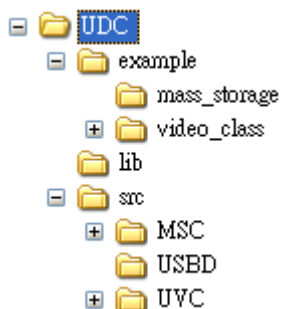
And then set the demo board into normal mode, and power on the demo board, user could see the UART status to execute the options as follows.



The two build modes N32901_ and N32903_ are supported.

UDC

The folder is shown as follows.



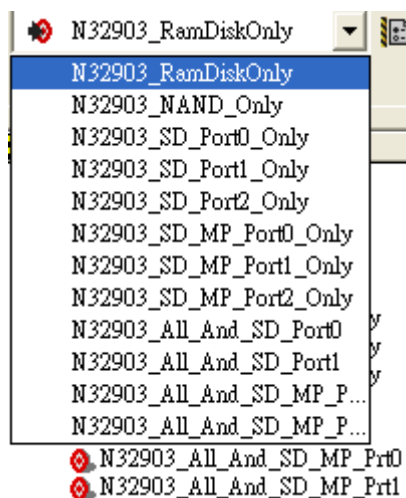
The folder includes the two USB class device USB mass storage and USB video class.

The function uses the demo board. Before running the AxD, firstly USB should plug in between the demo board and root hub of PC.

Firstly user should build the project MSC.mcp within the subfolder Src\MSC to export some USB mass storage libraries. Build the project USBD.mcp within the subfolder Src\USBD to export some USBD libraries, and build the project UVC.mcp within the subfolder\UVC to export some UVC libraries. Build the project mass_storage.mcp and video_class_demo.mcp within the subfolder example. Later on running the built images, user could see the UART status and Windows explorer to check the result. One video class device needs some DirectShow Applications to works, for example AMCAP.

About UVC build, it is for demo board after 2012.10. If user would use the demo board of old version, please contact with nuvoton.

Please note the build option of the project mass_storage.mcp as follows.

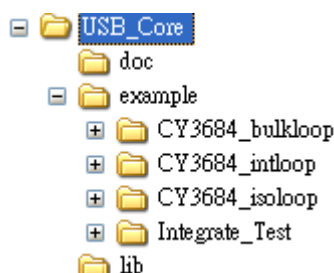


The current demo board has no SD port 1, it only has SD port 0 and port 2. The definition SD_MP means SD cards including multiple partitions. If SD has 2 partitions, select the options and user could see the 2 disks under Windows explorer. The definition All_Storage means to display the storage including SD and NAND.

The two build modes N32901_ and N32903_ are supported.

USB Core

The folder is shown as follows.



The function uses the demo board, and plug pen driver in the demo board.

For the example, we only use the subfolder Integrate_Test. We do not have the boards to test the others. In the example, build and run, the UART status will display as follows.

```
Init RTC...OK
PLL out frequency 192,000 Khz
fsInitFileSyste
=====
Please select the USB host port through Transceiver ports or GPIO
[1] Transceiver port 0
[2] Transceiver port 0 and 1
[3] Host-like port (GPB0 , GPB1)
[4] Host-like port (GPE3 , GPE4)
```

For the current N32903 demo board, user should set option 4 to run, and the following architecture will display.

```
USB host like port 1
act_altsetting is 0
id_index calculated to be: 17
Array length appears to be: 19
USB Mass Storage device detected
Endpoints: In: 0x800BB24C Out: 0x800BB260 Int: 0x0 (Period 0)
New GUID 090C620000000000000000000000
GetMaxLUN command result is 1, data is 0
Bulk max logical unit number: 0
Mass storage transport: Bulk-only
Mass storage protocol: Transparent SCSI
WARNING: USB Mass Storage data integrity not assured
USB Mass Storage device found at 2

***** Read lun 0 *****

INQUIRY ==>
Fixing INQUIRY data to show SCSI rev 2
Manufacture: Generic
Product: USB SD Reader
Serial number:
TEST UNIT READY ==>
REQUEST SENSE ==>
REQUEST_SENSE - no sense
READ CAPACITY ==>
USB disk found: size=971 MB, uTotalSectorN=1,990,655
USB Mass Storage support registered.
Disk size = 993,120 KB, Free speace = 952,960 KB
=====
Please select the item to verify
[1] Pen driver
[2] W99683 Camera + Pen driver + USB keyboard
```

User should select the option 1 to run (option 2 needs more devices to run), UART status will display as follows.

```

Opene file:IC:\Test\0000002.datl, file handle:30,008
262,144 bytes
Opene file:IC:\Test\0000002.datl, file handle:30,010
ImageSize: 262,144
Opene file:IC:\Test\0000003.datl, file handle:30,011
262,144 bytes
Opene file:IC:\Test\0000003.datl, file handle:30,013
ImageSize: 262,144
Opene file:IC:\Test\0000004.datl, file handle:30,014
262,144 bytes
Opene file:IC:\Test\0000004.datl, file handle:30,016
ImageSize: 262,144
Opene file:IC:\Test\0000005.datl, file handle:30,017
262,144 bytes
Opene file:IC:\Test\0000005.datl, file handle:30,019
ImageSize: 262,144
Opene file:IC:\Test\0000006.datl, file handle:30,020
262,144 bytes
Opene file:IC:\Test\0000006.datl, file handle:30,022
ImageSize: 262,144
Opene file:IC:\Test\0000007.datl, file handle:30,023
262,144 bytes
Opene file:IC:\Test\0000007.datl, file handle:30,025
ImageSize: 262,144
Opene file:IC:\Test\0000008.datl, file handle:30,026
262,144 bytes
Opene file:IC:\Test\0000008.datl, file handle:30,028
ImageSize: 262,144
Opene file:IC:\Test\0000009.datl, file handle:30,029
262,144 bytes
Opene file:IC:\Test\0000009.datl, file handle:30,031
Done

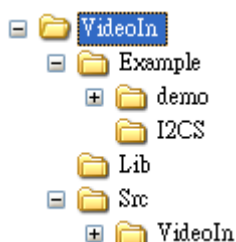
```

The 10 files will exists in pen driver.

The two build modes N32901_ and N32903_ are supported.

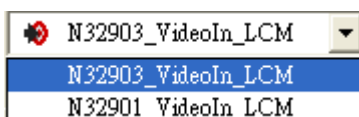
Video In

The folder is shown as follows



The function uses the demo board.

Firstly user should build the project VideoIn.mcp within the subfolder Src\VideoIn to export one library. And then build the project demo.mcp within the subfolder example\Demo by select the option N32903_VideoIn_LCM for N32903 or N32901_VideoIn_LCM for N32901. The build is for the demo board after 2012.10, with GiantPlus LCM. If user would like to use the setting of old version, please contact with nuvoton. Within the project demo.mcp, there is one option to build as follows



Please build the option, and run the built image by using ICE. After running and see the UART status .

Because the current demo board uses OV7725 sensor, user presses 3 to select OV7725. LCD panel will display some images from Video IN.

The two build modes N32901_ and N32903_ are supported.

VPOST

The folder is shown as follows.



The function uses the demo board.

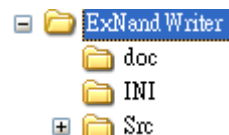
Firstly user should build the project W55FA93_VPOST.mcp within the subfolder Src to export one library. And then open the project VPOST_example.mcp within the subfolder example, and see the file main_vpost.c to check the statement #include "roof_320x240_rgbx888.dat" to be opened. If the statement is close, open the statement and close the other. And the build the project, later on running the built image to see LCD panel,

Open the project VPOST_OSD_example.mcp withn the subfolder example_OSD. And see the file main_vpost.c. Please confirm the arrayVpost_Frame, it is the background image. Please note the resolution of background image is equal to the resolution of current LCD, otherwise the error image will happen. The resolution of current LCD is 320 x240, so we select the file roof_320x240_yuv422.dat to be the background image.

The two build modes N32901_ and N32903_ are supported.

ExtNandWriter

The folder is shown as follows.



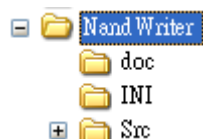
The function uses EVB, and now demo board cannot work it.

User could read the document ExNandWriter User Guide.pdf within the subfolder doc for more details.

The two build modes N32901_ and N32903_ are supported.

NandWriter

The folder is shown as follows.



The function uses the demo board.

User could read the document NandWriter User Guide.pdf within the subfolder doc for more details

The two build modes N32901_ and N32903_ are supported.

SDWriter

The folder is shown as follows.



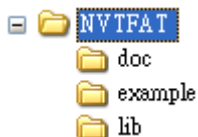
The function uses the demo board.

User could read the document SDWriter User Guide.pdf within the subfolder doc for more details. SDWriter could write two devices into SD card. There is one question to note, SD card only displays one device by using any SD card reader. If we want to see 2 devices within one SD card, plug the SD card into the SD card 0 of demo board. Build the project mass_storage.mcp within the subfolder UDC\example\mass_storage by selecting the option N32903_SD_MP_Port0_Only, and run it. The two devices will display in Windows explorer.

The two build modes N32901_ and N32903_ are supported.

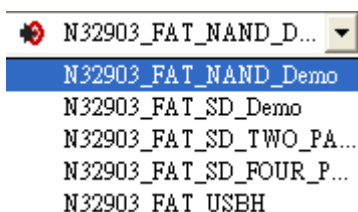
NVTFAT

The folder is shown as follows



The function uses the demo board.

Firstly User should build the project NVTFAT_demo.mcp within the subfolder example. Later on run the built image. User could see UART status to check the result. There are 5 options to build as follows.



N32903_FAT_NAND_demo will check the NAND on board, the result will display on UART status. N32903_FAT_SD_Demo will check SD card on board, so user must plug SD card in the demo board before running the built image. User could ignore the options N32903_FAT_SD_TWO_PART and N32903_FAT_SD_FOUR_PART. The UART status will display the result. For NAND and SD, user could use the command “dir” to display current file information, About the commands, please see the file Demo_CommandShell.c within the subfolder example, and check the variable _pcFileCommads and enumeration EnumerateCommandList for more details.

Before running N32903_FAT_USBH, user should plug the pen drive or card read including SD into the root hub of USB. The result will display on UART status.

The two build modes N32901_ and N32903_ are supported.

3. Revision History

| Version | Date | Description |
|-----------|--------------|--|
| V1.00.004 | Dec. 19 2012 | <ul style="list-style-type: none"> Integrate all N3290X products excluding N32905 |
| V1.00.003 | Nov. 16 2012 | <ul style="list-style-type: none"> Support Giantplus GPM1006D0 panel. |
| V1.00.002 | Aug. 20 2012 | <ul style="list-style-type: none"> Add nv_malloc and nv_free for user defined. |
| V1.00.001 | Jul. 12 2012 | <ul style="list-style-type: none"> Created |

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in equipment or systems intended for surgical implantation, atomic energy control instruments, aircraft or spacecraft instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for any other applications intended to support or sustain life. Furthermore, Nuvoton products are not intended for applications whereby failure could result or lead to personal injury, death or severe property or environmental damage.

Nuvoton customers using or selling these products for such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from their improper use or sales.