M451 Display BMP File From USB Disk

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This sample code reads the BMP file of USB DISK contents using M451
BSP Version	M451BSP_v3.01.003
Hardware	NuTiny-EVB-M451-LQFP100 V1.3

1 Function Description

1.1 Introduction

Using the M451 TINY BOARD, use the EBI interface to connect the 3.5-inch TFT screen and read out the BMP file of USB DISK contents.BMP format is 24 BIT, converted to RGB 565 format throw out.

2 Code Description

ILI9325 initialization：

```
void ILI9325_Initial(void)
{
    SET_RST;
    CLK_SysTickDelay(1000);  // Delay 1ms
    CLR_RST;
    CLK_SysTickDelay(10000); // DeLCD_CtrlWrite_ILI9325lay 10ms ,This delay time is necessary
    SET_RST;
    CLK_SysTickDelay(50000); // Delay 50 ms

    //************* Start Initial Sequence **********//

    LCD_WR_REG(0xCF);
    LCD_WR_DATA(0x00);
    LCD_WR_DATA(0xC1);
    LCD_WR_DATA(0X30);
    LCD_WR_REG(0xED);
    LCD_WR_DATA(0x64);
    LCD_WR_DATA(0x03);
    LCD_WR_DATA(0X12);
    LCD_WR_DATA(0X81);
    LCD_WR_REG(0xE8);
    LCD_WR_DATA(0x85);
    LCD_WR_DATA(0x10);
    LCD_WR_DATA(0x7A);
    LCD_WR_REG(0xCB);
    LCD_WR_DATA(0x39);
    LCD_WR_DATA(0x2C);
    LCD_WR_DATA(0x00);
    LCD_WR_DATA(0x34);
    LCD_WR_DATA(0x02);
    LCD_WR_REG(0xF7);
    LCD_WR_DATA(0x20);
    LCD_WR_REG(0xEA);
    LCD_WR_DATA(0x00);
    LCD_WR_DATA(0x00);
    LCD_WR_REG(0xC0);    //Power control
    LCD_WR_DATA(0x1B);   //VRH[5:0]
```

```
    LCD_WR_REG(0xC1);    //Power control
    LCD_WR_DATA(0x01);   //SAP[2:0];BT[3:0]
    LCD_WR_REG(0xC5);    //VCM control
    LCD_WR_DATA(0x30);   //3F
    LCD_WR_DATA(0x30);   //3C
    LCD_WR_REG(0xC7);    //VCM control2
    LCD_WR_DATA(0XB7);
    LCD_WR_REG(0x36);    // Memory Access Control
    LCD_WR_DATA(0x48);
    LCD_WR_REG(0x3A);
    LCD_WR_DATA(0x55);
    LCD_WR_REG(0xB1);
    LCD_WR_DATA(0x00);
    LCD_WR_DATA(0x1A);
    LCD_WR_REG(0xB6);    //Display Function Control
    LCD_WR_DATA(0x0A);
    LCD_WR_DATA(0xA2);
    LCD_WR_REG(0xF2);    //3Gamma Function Disable
    LCD_WR_DATA(0x00);
    LCD_WR_REG(0x26);    //Gamma curve selected
    LCD_WR_DATA(0x01);
    LCD_WR_REG(0xE0);    //Set Gamma
    LCD_WR_DATA(0x0F);
    LCD_WR_DATA(0x2A);
    LCD_WR_DATA(0x28);
    LCD_WR_DATA(0x08);
    LCD_WR_DATA(0x0E);
    LCD_WR_DATA(0x08);
    LCD_WR_DATA(0x54);
    LCD_WR_DATA(0XA9);
    LCD_WR_DATA(0x43);
    LCD_WR_DATA(0x0A);
    LCD_WR_DATA(0x0F);
    LCD_WR_DATA(0x00);
    LCD_WR_DATA(0x00);
    LCD_WR_DATA(0x00);
    LCD_WR_DATA(0x00);
    LCD_WR_REG(0XE1);    //Set Gamma
    LCD_WR_DATA(0x00);
    LCD_WR_DATA(0x15);
    LCD_WR_DATA(0x17);
```

```
    LCD_WR_DATA(0x07);
    LCD_WR_DATA(0x11);
    LCD_WR_DATA(0x06);
    LCD_WR_DATA(0x2B);
    LCD_WR_DATA(0x56);
    LCD_WR_DATA(0x3C);
    LCD_WR_DATA(0x05);
    LCD_WR_DATA(0x10);
    LCD_WR_DATA(0x0F);
    LCD_WR_DATA(0x3F);
    LCD_WR_DATA(0x3F);
    LCD_WR_DATA(0x0F);
    LCD_WR_REG(0x2B);
    LCD_WR_DATA(0x00);
    LCD_WR_DATA(0x00);
    LCD_WR_DATA(0x01);
    LCD_WR_DATA(0x3f);
    LCD_WR_REG(0x2A);
    LCD_WR_DATA(0x00);
    LCD_WR_DATA(0x00);
    LCD_WR_DATA(0x00);
    LCD_WR_DATA(0xef);
    LCD_WR_REG(0x11);   //Exit Sleep
    CLK_SysTickDelay(120000);
    LCD_WR_REG(0x29);   //display on

    //set xy, y
    LCD_WR_REG(0x36);
    LCD_WR_DATA(0x0);

    //set x, y memory access
    LCD_WR_REG(0x2a);
    LCD_WR_DATA(0 >> 8);
    LCD_WR_DATA(0 & 0XFF);
    LCD_WR_DATA(320 >> 8);
    LCD_WR_DATA(320 & 0XFF);
    LCD_WR_REG(0x2b);
    LCD_WR_DATA(0 >> 8);
    LCD_WR_DATA(0 & 0XFF);
    LCD_WR_DATA(480 >> 8);
    LCD_WR_DATA(480 & 0XFF);
```

```
}
```

Initialize USB Host Mass Storage：

```
int  USBH_MassInit(void)
{
    int i = 0;

    g_umas = NULL;

    for(i = 0; i < UMAS_MAX_DEV; i++)
        _umac_alloc_mark[i] = 0;
    /* register the driver, return -1 if error */
    if(USBH_RegisterDriver(&_UsbMassStorageDriver) < 0)
        return -1;
    return 0;
}
```

Read the BMP file of the contents of USB DISK in main：

```
while (1)
{
     LCD_SetCursor(0, p);
     LCD_WR_REG(0x2c);
     res = f_read(&file1, Buff, 320 * 3, &s1);

     for (i = 0; i < s1; i = i + 3)
     {
          b = Buff[i] >> 3;
          g = Buff[i + 1] >> 2;
          r = Buff[i + 2] >> 3;
          color = b | (g << 5) + (r << (5 + 6));
          LCD_WR_DATA(color);
     }

     p--;

     if (res || s1 == 0)
          break;
}
```

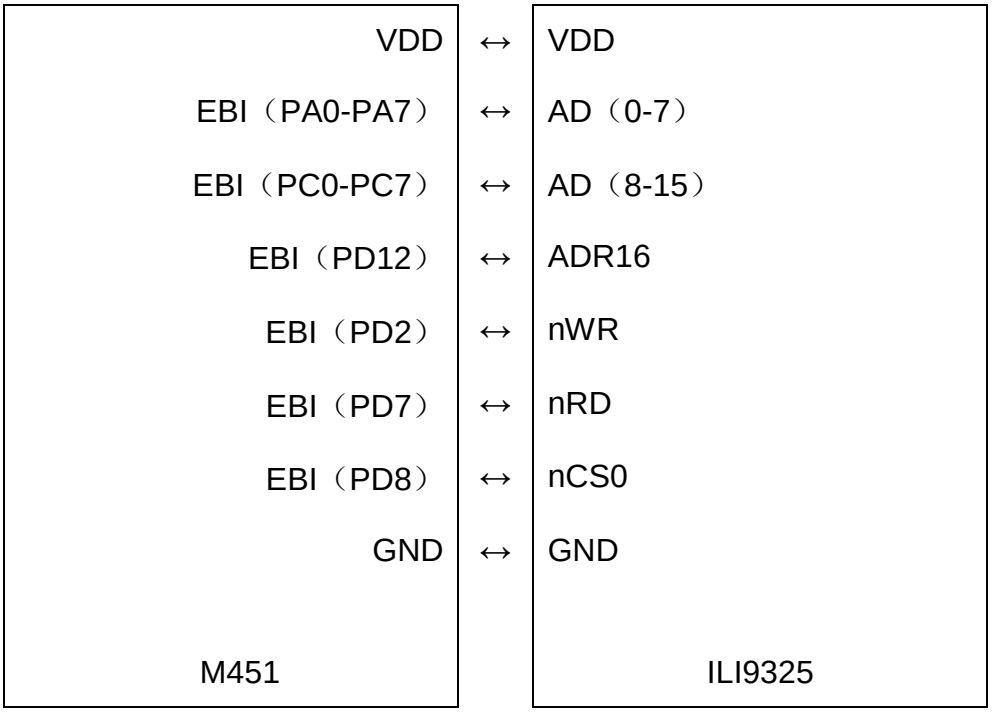
3 Software and Hardware Environment

Software Environment

- BSP version
  - M451BSP_v3.01.003
- IDE version
  - Keil uVersion5.25

Hardware Environment

- Circuit components
  - NuTiny-EVB-M451
  - ILI9325
- Diagram



4 Directory Information

- EC_M451_Display_BMP_File_From_USB_Disk_V1.00
  - Library — Sample code header and source files
    - CMSIS — Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
    - Device — CMSIS compliant device header file
    - StdDriver — All peripheral driver header and source files
    - UsbHostLib — USB host library source code
  - SampleCode
    - ExampleCode — Source file of example code
  - ThirdParty
    - FatFs — A generic FAT file system module for small embedded systems. Its official website is: http://elm-chan.org/fsw/ff/00index_e.html.

5 How to Execute Example Code

1. Browsing into sample code folder by Directory Information(section 4) and double click Display_BMP_File_From_USB_Disk.uvproj.
2. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
3. Enter debug mode
   a. Run

6 Revision History

Date	Revision	Description
Jul.24, 2019	1.00	1. Initially issued.

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.