Generating PDF File in USB Disk on M258

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This sample code is a demonstration of how to generate the PDF by accesses the FAT file system on USB DISK using M258.
BSP Version	M251/M252/M254/M256/M258 BSP CMSIS V3.01.002
Hardware	NuMaker-M258KE Ver1.1

1 Overview

This sample code uses M258 to connects to the computer via a USB port to be implemented as a mass storage class device, and accesses the FAT file system on USB DISK. It also demonstrates how to write data to a PDF file on the USB DISK. The user can use the microcontroller as a general external USB flash drive to read PDF file on a personal computer. Please note that the sample code is also applicable to M252 series.

1.1 Demo Result

Connect the M258 USB to the computer, open the [device manager] console, you can see the Nuvoton DataLogger USB Device in the Disk drives area, as shown in Figure 1-1. Then go to [This PC] to open the disk drive of the external device. The PDF files can be created and stored as shown in Figure 1-2.

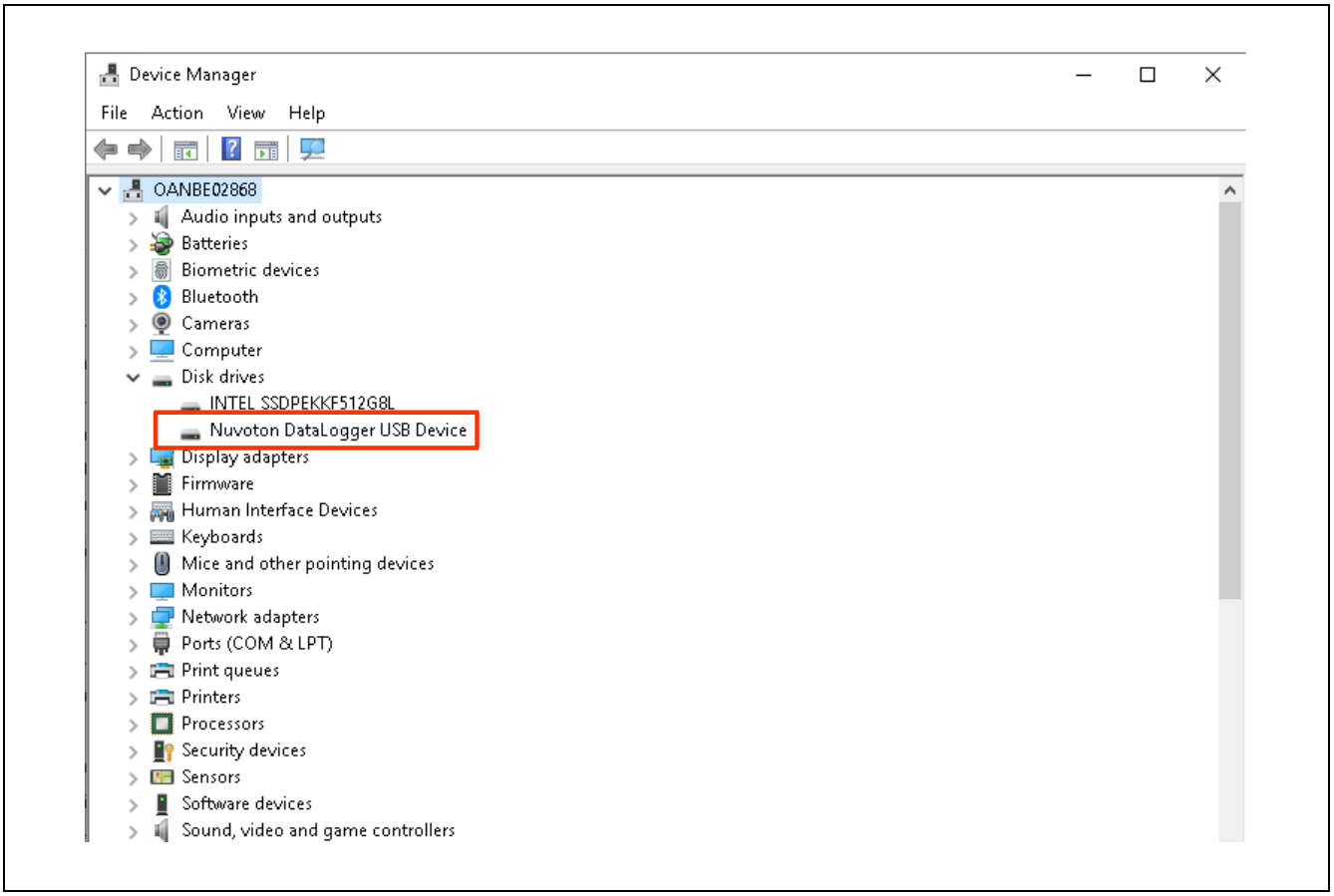

Figure 1-1 Device Manager Screenshot

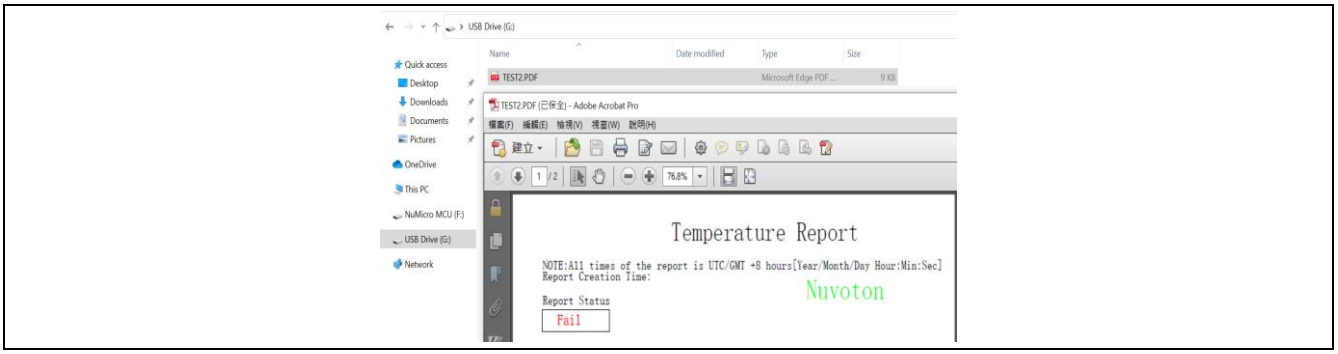

Figure 1-2 Screenshot of PDF File

2 Code Description

The user is mainly concerned with how to generate PDF file when using this code. The code in the USB part is basically standardized and does not need to be modified. If you want to create your own PDF page, you need to modify the following code.

```
int newpdf(char *name, uint32_t flag, fLoop func)
{
    uint32_t height, record_width;
    uint32_t width;
    uint32_t page_cnt;
    uint32_t     res;
    static axis_t axis_temp;
    static uint32_t total_time;
    static uint32_t total_lines, total_pages;
    NPDF_STATUS status;
    volatile uint32_t  start, end, sec;

    if (flag)
    {
        memset((char *)&warning_info, 0, sizeof(warning_info));

        fsize = g_total_len;
        getDeviceConfig();

        /* Parse record file to create PDF content(e.g date, time, temperature, and so on)
The content please refer to record_t structure */
        memset((void *)&axis_temp, 0, sizeof(axis_temp));
        total_lines = getFileInfo(&axis_temp, &axis_temp.setup, &total_time);

        if (total_lines == 0)
        {
            printf("total_lines == 0\n");
            goto out;
        }

        if (total_lines != (fsize / sizeof(record_t)))
        {
            printf("Flash record crash?(%d, %d)\n", fsize, total_lines);
            goto out;
        }
         /* Calculate total pages */
        total_pages = total_lines / LINES_PER_PAGE;

        if (total_lines % LINES_PER_PAGE)
            total_pages++;

        /* Include first page */
        total_pages++;
        printf("total_lines = %d, %d\n", total_lines, total_pages);

        NPDF_Pdf_SetGetSize();

    }
    else
        NPDF_Pdf_SetCallBack(func);

    height = NPDF_Page_GetHeight();
    width = NPDF_Page_GetWidth();

    record_width = width - PAGE_LEFT_SPACE * 2;

    page_cnt = NPDF_Pdf_GetPage();
```

```
    if (page_cnt > 1) /* begin from page */
    {
        printf("jump to page %d\n", page_cnt);
        page_cnt--;
        goto data_page2;
    }

    status = NPDF_Pdf_Open(name, total_pages, stream_buffer, NPDF_STREAM_BUF_SIZE,
owner_passwd, user_passwd);

    if (status != NPDF_OK)
    {
        printf("NPDF_Pdf_Open fail:%x\n", status);
        goto out;
    }

#ifdef ENABLE_ZIP
    /* Enable compress */
    NPDF_SetCompressionMode(NPDF_COMP_ALL);
#else
    NPDF_SetCompressionMode(0/*NPDF_COMP_ALL*/);
#endif

    /********************* page1 *********************/

    CreatePage1(&axis_temp, total_time, total_lines, total_pages);


    /**********************page 2..n*****************/
data_page2:
    page2_table[0].width = record_width - 4;

    res = NPDF_FOPEN();

    if (res != FR_OK)
    {
        printf("open file zhongji.txt fail, %x\n", res);
        goto out;
    }

    if (page_cnt != 1)
    {
        /* Jump to page_cnt */
        /* Seek ( page_cnt -1 ) * LINES_PER_PAGE *sizeof(record_t) data*/
        NPDF_FSEEK((page_cnt - 1)* LINES_PER_PAGE * sizeof(record_t));
    }

    while (page_cnt < total_pages)
    {
        CreateDataPage("SimSun", "Helvetica", ++page_cnt, total_pages);

        /* last line */
        NPDF_Page_SetRGBStroke(0.0, 0.0, 0.0);  //black
        NPDF_Page_SetLineWidth(1.0);
        NPDF_Page_MoveTo(PAGE_LEFT_SPACE, (tables[4].y - 400));
        NPDF_Page_LineTo(width - PAGE_LEFT_SPACE, tables[4].y - 400);
        NPDF_Page_Stroke();
    }

    NPDF_FCLOSE();

    NPDF_Pdf_Close();
out:
    return 0;
}
```

3 Software and Hardware Requirements

3.1 Software Requirements

- BSP version
  - M251/M252/M254/M256/M258 BSP CMSIS V3.01.002
- IDE version
  - Keil uVersion 5.26

3.2 Hardware Requirements

- Circuit components
  - NuMaker-M258KE Ver1.1

- Pin Connect

The pin connect is shown in Figure 3-1.

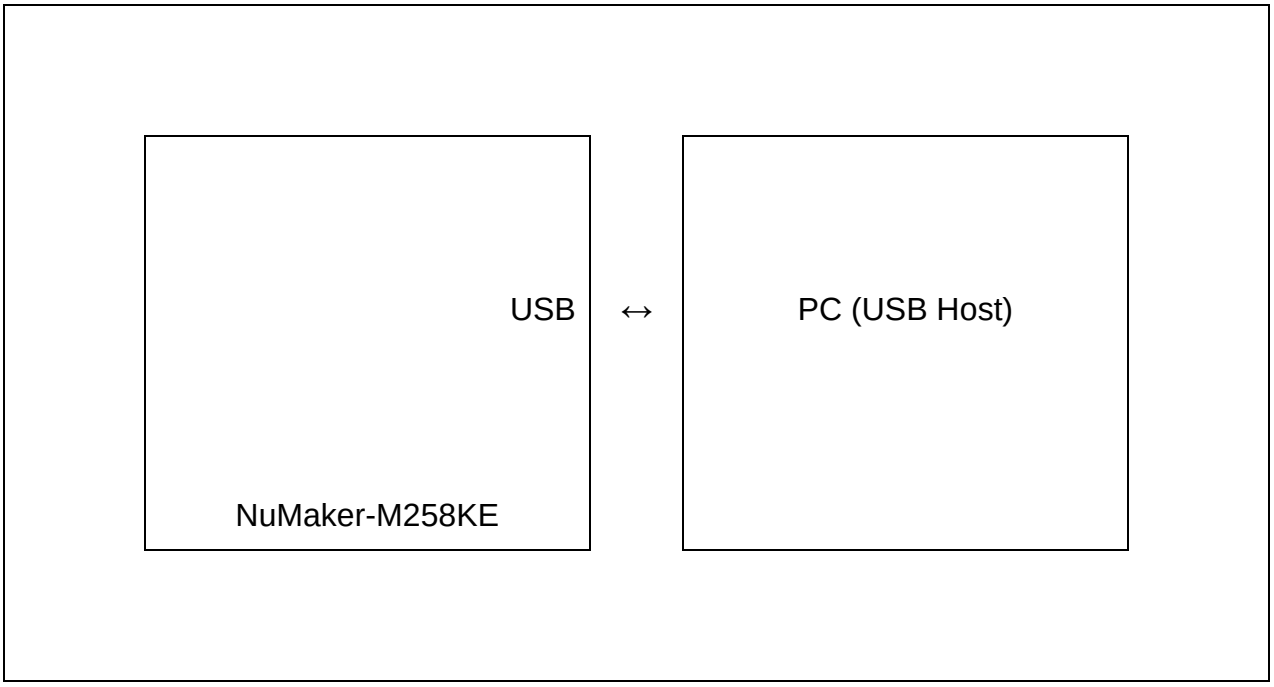

Figure 3-1 Pin Connect

4 Directory Information

The directory structure is shown in Figure 4-1.

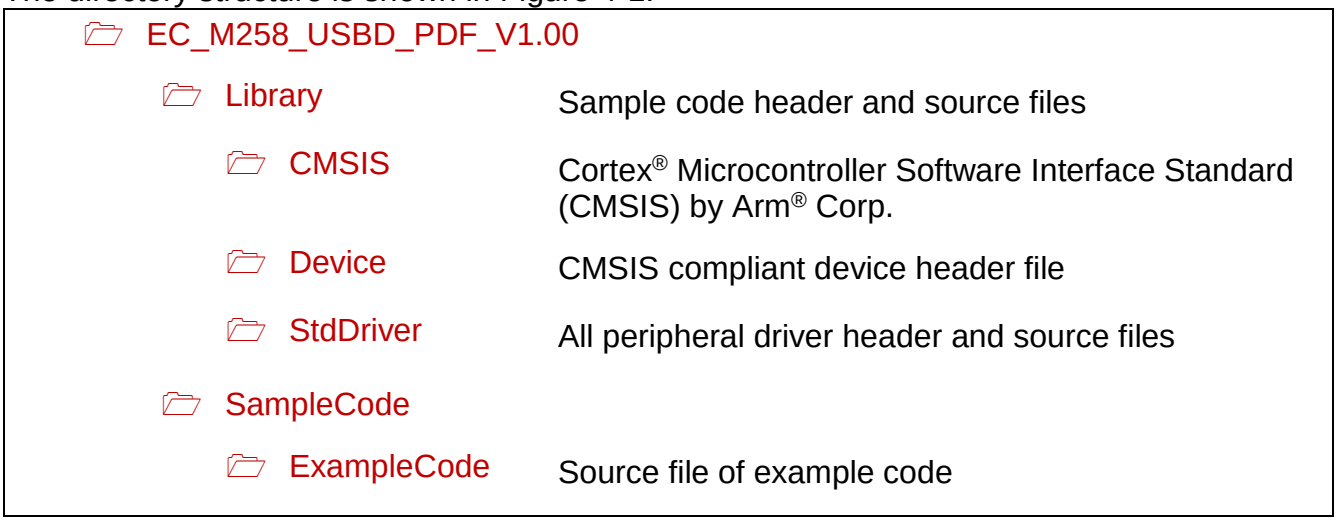

Figure 4-1 Directory Structure

5 Example Code Execution

1. Browse the sample code folder as described in the Directory Information section and double-click *M258_USBD_PDF.uvprojx.*
2. Enter Keil compile mode.
   - Build
   - Download
   - Start/Stop debug session
3. Enter debug mode.
   - Run

6 Revision History

Date	Revision	Description
2021.07.29	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.