GM03-16bit Library and Sample Code User's Manual

V2.0 - May. 5, 09

3F, No. 8, Dusing Rd., Hsinchu Science Park, Hsinchu City 30078, Taiwan, R.O.C

Tel: 886-3-6662118 Fax: 886-3-6662117 Web: www.generalplus.com

Important Notice

GENERALPLUS TECHNOLOGY INC. reserves the right to change this documentation without prior notice. Information provided by GENERALPLUS TECHNOLOGY INC. is believed to be accurate and reliable. However, GENERALPLUS TECHNOLOGY INC. makes no warranty for any errors which may appear in this document. Contact GENERALPLUS TECHNOLOGY INC. to obtain the latest version of device specifications before placing your order. No responsibility is assumed by GENERALPLUS TECHNOLOGY INC. for any infringement of patent or other rights of third parties which may result from its use. In addition, GENERALPLUS products are not authorized for use as critical components in life support systems or aviation systems, where a malfunction or failure of the product may reasonably be expected to result in significant injury to the user, without the express written approval of Generalplus.

Generalplus Documentation

The following documents are available from Generalplus. These documents provide useful information regarding the software programming and application designing.

- *xxx Instruction set user's manual* -This document explains the xxx instruction sets.
- *xxx confirmation sheet - A* check list for releasing code.

Table of Content

PAGE

1 INTRODUCTION 5

1.1 GENERAL DESCRIPTION 5

1.2 STRUCTURE DIAGRAM 5

2 RESOURCE ALLOCATION 6

2.1 RESOURCE ROM/RAM SIZE 6

2.2 EXAMPLE 6

3 API FUNCTIONS DESCRIPTION 7

3.1 API FUNCTION LIST OF LIBRARY 7

3.2 API FUNCTION DESCRIPTION 7

4 VARIABLES DESCRIPTION 8

4.1 VARIABLE LIST OF LIBRARY 8

4.2 VARIABLE DESCRIPTION 8

5 GM03 USER'S FUNCTION DESCRIPTION 9

5.1 FUNCTION LIST OF GM03_USER.ASM 9

5.2 USER'S FUNCTION DESCRIPTION 9

6 GM03 USER'S FUNCTION DESCRIPTION 10

6.1 EXAMPLES 10

Revision History

Revision	Date	By	Remark
2.0	2009/05/05	Wilson Wu	1. Add Mono animation GM03 Decode function
1.0	2009/02/25	Wilson Wu	First edition.

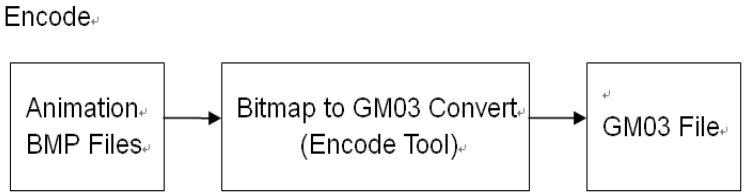
1 Introduction

1.1 General Description

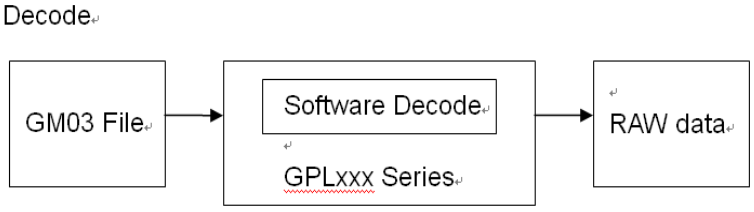
GM03 is a decoding algorithm for 4-Gray and Mono Image. We offer users the easy-to-use API and library to develop applications. Only the header file (GC03.h) and LIB need to be added and GM03_User.asm can be modified to meet various system requirements.

1.2 Structure Diagram

Structure diagram for encode



Structure diagram for decode



2 Resource Allocation

2.1 Resource ROM/RAM Size

The resources taken by GM03 Decoder are as follows:

RAM: 13 + Decode-Out buffer Size (for 4-gray GM03 decode only) words

ROM: 358 words

In GPL16 series, the LCD controller only supports 1bpp and 4 bpp formats. Thus, users should reserve one decode buffer to store 4-gray GM03 decode-out data and convert decode-out data type form 4-gray (2bpp) to 16-gray (4bpp). For Mono GM03 decode, this buffer can be assigned to LCD buffer.

2.2 Example

The decoding time of the corresponding frame size is as follows:

4-Gray animation:

Size: 32*64 Rate: 32.11%

Decode Time (Frame/second): AUTO mode: 2.5 ms (max)@48MHz SYSCLK

*MANUAL mode: 2.86 ms (max) @48MHz SYSCLK

Mono animation:

Size: 32*64 Rate: 32.11%

Decode Time (Frame/second): AUTO mode: 1.16 ms (max)@48MHz SYSCLK

*MANUAL mode: 1.4 ms (max) @48MHz SYSCLK

*Manual mode does not include the time accessing external memory (SPI Flash).

3 API Functions Description

3.1 API Function List of Library

Index	syntax
1	F_GM03_Initial
2	F_GM03_SetMode
3	F_GM03_Decode_Start
4	F_GM03_Get_RemainFrame_Num
5	F_GM03_Decode_ServiceLoop

3.2 API Function Description

Void F_GM03_Initial(int isGrayorMono);

Description
Desc: This function is to set decode format to Gray or Mon
Input: isGrayorMono: 0: is Mono, 1: is Gray
Output: Null

void F_GM03_SetMode(int Mode);

Description
Desc: This function is to set decode format to Auto mod or Manual mode
Input: Mode: 0: Auto mode, -1: Manual mode
Output: Null

void GM03_Decode_Start(int Index);

Description
Desc: If Auto mode is selected, users should assign GM03 resource index. After this function is completed, first frame will be decoded.
Input: Index: The GM03 resource index of GM03_Resource_Table
Output: Null

int GM03_Decode_ServiceLoop(void);

Description
Desc: GM03 library will decode next frame when this function is called.
Input: Null
Output: return = 1: GM03 decode is completed. return = 0: GM03 decode is not completed.

int F_GM03_Get_RemainFrame_Num(void);

Description
Desc: GM03 library will return the remaining frame number when this function is called.
Input: Null
Output: Remaining frame number

4 Variables Description

4.1 Variable List of Library

Index	syntax
1	R_Frame_Data_Size
2	R_GM03_Frame_Width
3	R_GrayorMono_Flag
4	R_Dec_In_Length
5	R_GM03_DEC_Out_Buffer
6	R_GM03_DEC_In_Buffer

4.2 Variable Description

R_Frame_Data_Size

This variable is public from GM03 Library. It will show the byte number of GM03 Decode out raw data size.

R_GM03_Frame_Width

This variable is public from GM03 Library. It will show the pixel number of frame width.

R_GrayorMono_Flag

This variable is public from GM03 Library. It will show the decoding GM03 type is Gray or Mono.

R_Dec_In_Length

This variable is public from GM03 Library. Users should write decode in buffer length to this variable when selecting Manual mode.

R_GM03_DEC_Out_Buffer

This variable is public from GM03_User.asm. Users should designate one decode out buffer for GM03 library to store decode out data.

R_GM03_DEC_In_Buffer

This variable is public from GM03_User.asm. Users should designate one decode in buffer for GM03 library to read Decode in data. This variable is necessary when selecting manual mode.

5 GM03 User's Function Description

5.1 Function List of GM03_User.asm

Index	syntax
1	F_GM03_User_Initial
2	F_GM03_SetBufferAddr_User
3	F_GM03_User_Get_Data
4	F_GM03_User_Process

5.2 User's Function Description

F_GM03_User_Initial

Description
Desc: sets decode in length or initializes LCD settings in this function.
Input: R_Frame_Data_Size

F_GM03_SetBufferAddr_User

Description
Desc: sets manual mode decode in buffer address in this function.
Input: Decode in buffer address

F_GM03_User_Get_Data

Description
Desc: GM03 library will call this function to read decode in data.
Input: R1: Write GM03 decode-in buffer start point R2: Write GM03 decode-in buffer data size

F_GM03_User_Process

Description
Desc: For GPL16, LCD controller only supports 4bpp and 1bpp format; it allows to process data format from 2bpp to 4bpp in this function. Users can also refer to R_GrayorMono_Flag to identify Decode out data format at 1bpp or 2bpp.
Input: R_GrayorMono_Flag

6 GM03 User's Function Description

6.1 Examples

Manual mode example code:

```
Int ret;
System_Initial();
TimeBase_Flag = 0;
FrameCnt = 0;
LCD_Enable ();
EnableTimeBase();
while(1)
{
    F_GM03_Initial(C_Mono);                          //Set decode format is Mono
    F_GM03_SetMode(C_Manual_Mode);                   //Set decode mode is manual mode
    F_GM03_SetBufferAddr_User(DecodeInBuffer);       //Set decode in buffer address
    F_GM03_Decode_Start(0);                          //Start decode
    FrameCnt = F_GM03_Get_RemainFrame_Num();         //Get remain Frame number
    while(ret != 1)                                  //Decode GM03 until return value = 1
    {
        if(TimeBase_Flag == 1)
        {
            ret = F_GM03_Decode_ServiceLoop();
            TimeBase_Flag = 0;
        }
    }
        ret = 0;
}
```

Auto mode example code:

```
Int ret;
System_Initial();
TimeBase_Flag = 0;
FrameCnt = 0;
LCD_Enable ();
EnableTimeBase();
while(1)
```

```
{
        F_GM03_Initial(C_Gray);                          //Set decode format is Gray
        F_GM03_SetMode(C_Auto_Mode);                     //Set decode mode is Auto mode
        F_GM03_Decode_Start(0);                          //Set GM03 decode the index 0 of
                                                         //GM03_Resource_Table. Start Decode
        FrameCnt = F_GM03_Get_RemainFrame_Num();         //Get remain Frame number
        while(ret != 1)                                  //Decode GM03 until return value = 1
        {
                if(TimeBase_Flag == 1)
                {
                        ret = F_GM03_Decode_ServiceLoop();
                        TimeBase_Flag = 0;
                }
        }
                ret = 0;
}
```