



GPLB5X/3X GM03 Library user's manual

V2.0 – Apr.29,09'



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

	<u>PAGE</u>
1 INTRODUCTION	5
1.1 GENERAL DESCRIPTION	5
1.2 STRUCTURE DIAGRAM	5
2 RESOURCE OCCUPIED	6
2.1 RESOURCE ROM/RAM SIZE	6
2.2 EXAMPLES	6
3 API FUNCTIONS DESCRIPTION	7
3.1 API FUNCTION LIST OF LIBRARY	7
3.2 API FUNCTION DESCRIPTION	7
4 USER VARIABLES DESCRIPTION	錯誤! 尚未定義書籤。
4.1 VARIABLE LIST FOR USER	11
4.2 VARIABLE DESCRIPTION	11
5 GM03 USER'S FUNCTION DESCRIPTION	12
5.1 FUNCTION LIST OF GM03_USER.ASM	12
5.2 USER'S FUNCTION DESCRIPTION	12
6 GM03 USER'S FUNCTION DESCRIPTION	13
6.1 EXAMPLES	13

Revision History

Revision	Date	By	Remark
2.0	2009/4/29	dingdh	1. Add Mono animation GM03 Decode function

1 Introduction

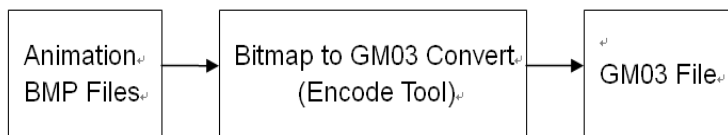
1.1 General Description

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.inc) and LIB need to be added and GM03_User.asm and GM03_User.inc can be modified to meet various system requirements.

1.2 Structure Diagram

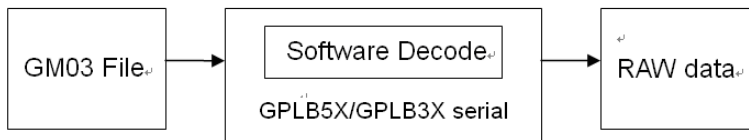
Structure Diagram for Encode

Encode



Structure Diagram for Decode

Decode



2 Resource Allocation

2.1 Resource ROM/RAM size

The resources used by GM03 Decoder are as follows:

RAM: 25 + 4(user RAM for manual mode only + Decode-In buffer Size (for manual mode only) bytes

ROM: 1249 bytes

2.2 Examples

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

4-Gray animation:



Size: 32*64 Rate: 32.11%

Decode Time (Frame/sec): AUTO mode: 7.6~9.2 ms (max)@6MHz SYSCLK on body GPLB5X

 *MANUAL mode: 8 ~12 ms (max) @6MHz SYSCLK on body GPLB5X

Mono animation:



Size: 32*64 Rate: 32.11%

Decode Time (Frame/sec): AUTO mode: 4.0~4.4 ms (max)@6MHz SYSCLK on body GPLB5X

 6.6~7.0 ms (max)@4MHz SYSCLK on body GPLB3X

 *MANUAL mode: 4.8~5.6 (max) @6MHz SYSCLK on body GPLB5X

 7.4~8.6 (max) @4MHz SYSCLK on body GPLB3X

*Manual mode does not include the time to access 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_SetBufferAddr_User
4	F_GM03_SetBufferFlag
5	F_GM03_SetLCDBuf_Interval
6	F_GM03_Decode_Start
7	F_GM03_Get_Frame_Num
8	F_GM03_Decode_ServiceLoop
9	F_GM03_GetWidth
10	F_GM03_GetDecodeOutLength

3.2 API Function Description

F_GM03_Initial:

Description
Desc: This function is to set decode format to Gray or Mono.
Input: A: 0: is Mono, 1: is Gray
Output: Null
Example: LDA #C_Mono_Type JSR F_GM03_Initial

F_GM03_SetMode:

Description
Desc: This function is to set decode format to Auto mod or Manual mode.
Input: A: 0: Auto mode, FFh: Manual mode
Output: Null
Example: LDA #C_Manual_Mode JSR F_GM03_SetMode

F_GM03_SetBufferAddr_User:

Description
Desc: This function is to set decode out buffer address.
Input: X: decode out buffer low address, Y: decode out buffer high address
Output: Null
Example: LDX #.LOW.C_Display_Buf LDY #.HIGH.C_Display_Buf JSR F_GM03_SetBufferAddr_User

F_GM03_SetBufferFlag:

Description
Desc: This function is to set decode out buffer attribute. if is set LCD buffer, then F_GM03_SetBufferAddr_User must set LCD buffer address;
Input: A: 0: LCD buffer, 1: user define ram buffer
Output: Null
Example: LDA # C_LCDBuffer JSR F_GM03_SetBufferFlag

F_GM03_SetLCDBuf_Interval:

Description
Desc: This function is to set the interval of LCD Buffer. If F_GM03_SetBufferFlag is set to LCD buffer, this function must be set.
Input: A: LCD buffer COM interval
Output: Null
Example: LDA #8 JSR F_GM03_SetLCDBuf_Interval

Note 1: If body is GPLB35/36, the actual address of COM0 is \$3D00, and actual address of COM1 is \$3D10; thus, input A is 10h (\$3D10-\$3D00=10h) of the function F_GM03_SetLCDBuf_Interval.

LCD Address Mapping for GPLB35/36: 32x100, 36x96, 48x84 (CPU Side)

	S0- S7	S8- S15	S16- S23	S24- S31	S32- S39	S40- S47	S48- S55	S56- S63	S64- S71	S72- S79	S80- S87	S88- S95	S96- S99
	b0-b7	b0-b7	b0-b7	b0-b7	b0-b7	b0-b7	b0-b7	B0-b7	b0-b7	b0-b7	b0-b7	b0-b7	b0-b3
Com0	\$3D00	\$3D01	\$3D02	\$3D03	\$3D04	\$3D05	\$3D06	\$3D07	\$3D08	\$3D09	\$3D0A	\$3D0B	\$3D0C
Com1	\$3D10	\$3D11	\$3D12	\$3D13	\$3D14	\$3D15	\$3D16	\$3D17	\$3D18	\$3D19	\$3D1A	\$3D1B	\$3D1C

Note 2: In GPLB33, the actual address of COM0 is \$3E00, and actual address of COM1 is \$3E10; thus, input A is 10h (\$3E10-\$3E00=10h) of the function F_GM03_SetLCDBuf_Interval.

LCD Address Mapping for SPLB33A: 16x52 (CPU Side)

	S0-S7	S8-S15	S16-S23	S24-S31	S32-S39	S40-S47	S48-S51
	b0-b7	b0-b7	b0-b7	b0-b7	b0-b7	b0-b7	b0-b3
Com0	\$3E00	\$3E01	\$3E02	\$3E03	\$3E04	\$3E05	\$3E06
Com1	\$3E10	\$3E11	\$3E12	\$3E13	\$3E14	\$3E15	\$3E16

Note 3: If body is GPLB39, the actual address of COM0 is \$3C00, and actual address of COM1 is \$3C10; thus, input A is 10h (\$3C10-\$3C00=10h) of the function F_GM03_SetLCDBuf_Interval.

LCD Address Mapping for SPLB39A: 112X8, 112X16, 112x32 , 112x48, 96x64 (CPU Side)

	S0-S7	S8-S15	S16-S23	S24-S31	S32-S39	S40-S47	S48-S55	S56-S63	S64-S71	S72-S79	S80-S87	S88-S95	S96-S103	S103-S111
	b0-b7	b0-b7	b0-b7	b0-b7	b0-b7	b0-b7	b0-b7	b0-b7	b0-b7	b0-b7	b0-b7	b0-b7	b0-b3	b0-b3
Com0	\$3C00	\$3C01	\$3C02	\$3C03	\$3C04	\$3C05	\$3C06	\$3C07	\$3C08	\$3C09	\$3C0A	\$3C0B	\$3C0C	\$3C0D
Com1	\$3C10	\$3C11	\$3C12	\$3C13	\$3C14	\$3C15	\$3C16	\$3C17	\$3C18	\$3C19	\$3C1A	\$3C1B	\$3C1C	\$3C1D

Note 4: If body is GPLB5X,

For Gray mode $A = (P_3045H_LCD_Common + 1) / 4$;

For Mono mode, $A = (P_3045H_LCD_Common + 1) / 8$;

F_GM03_Decode_Start

Description
Desc: User should assign GM03 resource index. After this function is finished, first frame will be decoded.
Input: X: The GM03 resource index of GM03_Resource_Table
Output: Null
Example: LDX #0 JSR F_GM03_Decode_Start

F_GM03_Decode_ServiceLoop

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

F_GM03_Get_Frame_Num:

Description
Desc: GM03 library will return remain frame number when this function is called
Input: Null
Output: A: Remain frame number

F_GM03_GetWidth:

Description
Desc: GM03 library will return the width of image
Input: Null
Output: A: the width of image

F_GM03_GetDecodeOutLength:

Description
Desc: GM03 library will return the length of decoded out data
Input: Null
Output: X: low byte of the length; Y: high byte of the length

4 Variable Description

4.1 Variable List for user

Index	syntax
1	R_GM03_DEC_In_Buffer
2	R_Manual_Size
3	R_GM03_Manual_AddrL
4	R_GM03_Manual_AddrH
5	R_GM03_Manual_AddrB
6	R_Dec_In_Length

4.2 Variable Description

R_GM03_DEC_In_Buffer

This variable is public from GM03_User.asm. It saves the decode-in data when selecting Manual mode.

R_Manual_Size

This variable is public from GM03_User.asm. It saves the length of read-data when selecting Manual mode.

R_GM03_Manual_AddrL, R_GM03_Manual_AddrH, R_GM03_Manual_AddrB

This variable is public from GM03_User.asm. Users should designate the address of the GM03 in manual mode.

R_Dec_In_Length

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

5 GM03 Function Description

5.1 Function List of GM03_User.asm

Index	syntax
1	F_GM03_Auto_Addr
2	F_GM03_Manual_Addr
3	F_GM03_GetData_User

5.2 User's Function Description

F_GM03_Auto_Addr

Description
Desc: Sets GM03 address from resource table in auto mode.
Input: x: index of GM03 resource table

F_GM03_Manual_Addr

Description
Desc: Sets GM03 address from resource table in manual mode.
Input: x: index of GM03 resource table

F_GM03_GetData_User

Description
Desc: GM03 library will call this function to read decode-in data in manual mode.
Input: X: data length of read-data

6 GM03 Example Code

6.1 Examples

6.1.1 Manual mode example code

```
LDA    #C_Mono_Type
JSR    F_GM03_Initial
LDA    #C_Manual_Mode
JSR    F_GM03_SetMode
LDX    #.LOW.C_Display_Buf
LDY    #.HIGH.C_Display_Buf
JSR    F_GM03_SetBufferAddr_User
LDA    #C_LCDBuffer
JSR    F_GM03_SetBufferFlag
LDA    #8                                ;Mono 64 segment/8, Gray 64 segment/4
JSR    F_GM03_SetLCDBuf_Interval
LDX    #0
STX    R_GM03Index
JSR    F_GM03_Decode_Start
```

L_SystemService:

```
%NeedLcdDisplay        ?L_CheckNeedKeyScan
JSR    F_GM03_Decode_ServiceLoop
BCC    ?L_CheckNeedKeyScan
LDX    R_GM03Index
JSR    F_GM03_Decode_Start
```

?L_CheckNeedKeyScan:

```
%NeedKeyScan          L_SystemService
%WaitDebounceOver    L_SystemService
;when debounce not over, program will not scan key
JSR    F_KeyboardScan
JSR    F_KeyService
JMP    L_SystemService
```

6.1.2 Auto mode example code

```

LDA    #C_Mono_Type
JSR    F_GM03_Initial
LDA    #C_Auto_Mode
JSR    F_GM03_SetMode
LDX    #.LOW.C_Display_Buf
LDY    #.HIGH.C_Display_Buf
JSR    F_GM03_SetBufferAddr_User
LDA    #C_LCDBuffer
JSR    F_GM03_SetBufferFlag
LDA    #8                                ;Mono 64 segment/8, Gray 64 segment/4
JSR    F_GM03_SetLCDBuf_Interval
LDX    #0
STX    R_GM03Index
JSR    F_GM03_Decode_Start

```

L_SystemService:

```

%NeedLcdDisplay    ?L_CheckNeedKeyScan
JSR    F_GM03_Decode_ServiceLoop
BCC    ?L_CheckNeedKeyScan
LDX    R_GM03Index
JSR    F_GM03_Decode_Start

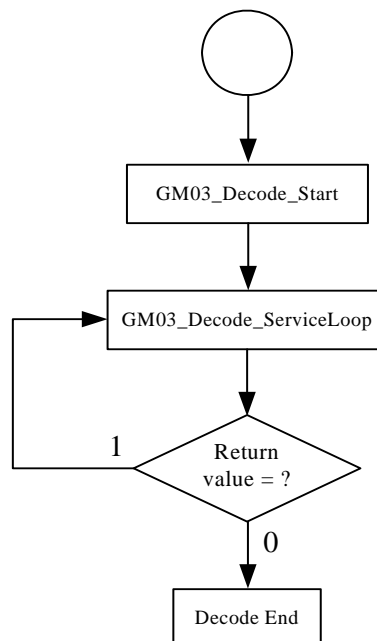
```

?L_CheckNeedKeyScan:

```

%NeedKeyScan      L_SystemService
%WaitDebounceOver L_SystemService
;when debounce not over, program will not scan key
JSR    F_KeyboardScan
JSR    F_KeyService
JMP    L_SystemService

```



Click below to find more

[Mipaper at www.lcis.com.tw](http://www.lcis.com.tw)

[Mipaper at www.lcis.com.tw](http://www.lcis.com.tw)