DRPC100-OLED UART PROTOCOL SPECIFICATION

Programmer Reference Manual

Version 1.4

Revision History

Revision	Issue Date	Comments
V1.0	2012/5/7	Initial release
V1.3	2012/9/18	Release
V1.4	2012/10/25	Add Switch Protocol

Agenda

- **Communication Protocol**
- **Command List**
- **COM port configuration**
- **Example**
- **Attachment A (Image Data Mapping)**
- **Attachment B (Some example code)**
- **Attachment C (Character layout of Text mode)**

Communication Protocol

- Background : OLED module is a 128x64 dots matrix display, black and white module for simply display messages from system, in text mode, user can print ascii character on screen by 8 rows vs. 21 columns, or in still picture mode, user can store 2 bank multiply 256 pages (total 512 page) in OLED module and call out to show on panel by simple command

- DRPC100 is presetting communication to OLED module by COM5 1200 N 8 1, all defined command are initialized by system/host/DRPC100, and response generated might by OLED module for system reference.

- Commands to MCU and data returned from MCU are conveyed in the following communication format :

Communication Protocol

Command	**Data**

Response	**Data**

Command List

- You can use following pre-defined commands on the application to communicate with OLED Module and to receive data returned from it.

Command List:

Command	Data	Description	Response bytes
0x04	[row][column][length][Ascii-0]….[Ascii-N] Length=1..21 N=length-1 Row=0..7 Column=0..20 Ascii-n=0x20 .. 0x7F	Display string on OLED at position [row.col]	0x5E
0x05	[bank][page][byte0]….[byte1023] Bank=0..1 Page=0..255 Byte0..byte1023 please reference attachment A	Burn a one-page image to Flash memory which is on OLEM module at address bank.page	0x5E
0x06	[bank][page] Bank=0..1 Page=0..255	Display the image at bank.page on OLED Remark: OLED module will default display page0 (in bank0) on screen when power on automatically, so we are recommend user to put company Logo at the page0, then it will first be seen when system be power on.	0x5E
0x07	[row][col]character] Row=0..7 Col=0..20 Character=0x00 (degree) =0x01 (Big Dot) =0x02 (Empty)	Put a special character at position row.col	0x5E
0x08	[bank][page] Bank=0..1 Page=0..255	Erase the image at address bank.page in Flash memory, This command need 25ms. System should wait over 25ms or get response 0x5E then to go next burning command (0x05).	0x5E
0xEE		Echo	0xEE
0xFC		Clear Screen	0x5E

0xFD		Get ID	0x44

Button Information

- OLEM Display module is implemented four digital buttons for user to control the display, there are UP, DOWN, ENTER and ESC buttons, if any button is pressed or released, module will response the button status in one byte data within 20ms through UART, user should check UART every 10ms or longer if there are buffer built and to decode each bit of response data to know the key is pressed (read as 0) or released (read as 1).

Command		Description	Response bytes
NONE		ESC Key Pressed	0xFE (11111110)
		ENTER Key Pressed	0xFD (11111101)
		LEFT/DOWN Key Pressed	0xFB (11111011)
		RIGHT/UP Key Pressed	0xF7 (11110111)
		All Key un-pressed	0xFF (11111111)

COM port configuration

Com port configuration setting	
Com Port	5
Baud rate	1200
Data bit	8
Parity check bit	None
Stop bit	1

Example

System SEND **OLED Module <Echo>** Command:

System->Module:
Command：<0xEE>
Module->System
Response : <0xEE>

System SEND **OLED Module <Get ID>** Command:

System->Module:
Command：<0xFD>
Module->System
Response : <0x44>

System SEND **OLED Module <Display String>** Command:

System->Module:
Command：<0x04><0x00><0x00><0x04><0x41><0x42><0x43><0x44>
Module->System
Response : <0x5E>

This command will show a string “ABCD”=” <0x41><0x42><0x43><0x44>” on screen from position [row, col]=[0, 0].

Remark : The ASCII code table please reference Attachment D.

System SEND **OLED Module <Burn Page>** Command:

System->Module:
Command：<0x05><0x00><0x00><0x55>……<0x55> (total 3+1024 bytes)
Module->System
Response : <0x5E>

This command will burn a simple 32 lines image at bank0, page0, where is the default power on display page.

Remark A: this function will take 10 sec to send data which was under 1200 buad rate translation , system should wait more 10 sec and check the 0x5E response from module to make sure burning is complete then leave for next action.

Remark B: Before burn a new image page, system should erase the page via erase command (0x08).

System SEND **OLED Module <Erase Page>** Command:

System->Module:
Command：<0x08><0x00><0x00>
Module->System

DRPC100-OLED

Response : <0x5E>

This command will erase the content at bank0, page0, all data will get back to 0xFF in Flash memory chip.

Remark : this function will take 25ms long, system should wait 25ms more or idle until get the complete response (0x5E) the go next action.

System SEND **OLED Module <Display Page>** Command:

System->Module:

Command：<0x06><0x00><0x01>

Module->System

Response : <0x5E>

This command will display bank0, page1 on the OLED module.

Attachment A

The data sequence of burning page command [0x05][bank][page][byte0][byte1]…..[byte1023] mapping to image are like below:

● (Left-up Corner of screen)						
Bit0	Bit0	Bit0		Bit0	Bit0	Bit0
(Row 0) Byte0	Byte1	Byte2	……	Byte125	Byte126	Byte127
Bit7	Bit7	Bit7		Bit7	Bit7	Bit7
Bit0	Bit0	Bit0		Bit0	Bit0	Bit0
(Row 1) Byte128	Byte129	Byte130	……	Byte253	Byte254	Byte255
Bit7	Bit7	Bit7		Bit7	Bit7	Bit7
……	……		……			
Bit0	Bit0	Bit0		Bit0	Bit0	Bit0
(Row 6) Byte768	Byte769	Byte770	……	Byte893	Byte894	Byte895
Bit7	Bit7	Bit7		Bit7	Bit7	Bit7
Bit0	Bit0	Bit0		Bit0	Bit0	Bit0
(Row 7) Byte896	Byte897	Byte898	……	Byte1021	Byte1022	Byte1023
Bit7	Bit7	Bit7		Bit7	Bit7	Bit7
				● (Right-Down corner of screen)		

Remark A : Totally 128x64 pixels or 8 rows by 128 bytes
Remark B : if bit=1, according position pixel is present white dot, if bit=0, black dot.

Attachment B

Some example C code

```
//clear screen
        SendByte(0xFC);
        Sleep(100); if(GetByte()==0x5E)printf("Clear Screen Done\n");

//display page
        printf("Page number(0..511)=");
        scanf("%d", &page);
        bank=(page>>8)&0x00ff;
        page=(page)&0x00ff;
        SendByte(0x06);
        SendByte(bank);
        SendByte(page);
        Sleep(100); if(GetByte()=='^')printf("Display page Done\n");

//Burn page
        printf("Page number(0..511)=");
        scanf("%d", &page);
        bank=(page>>8)&0x00ff;
        page=(page)&0x00ff;
        printf("image filename =");
        scanf("%s", filename);
        SendByte(0x08); //erase page
        SendByte(bank);
        SendByte(page);
        Sleep(1000); if(GetByte()=='^')printf("Erase page Done\n");
        SendByte(0x05);
        SendByte(bank);
        SendByte(page);
        f=fopen(filename, "rb");

        if(f!=NULL)
        {
            for(i=0; i<1024; i++)
            {
                c=fgetc(f);
                SendByte(c);
            }
        }
        else printf("fopen fail\n");
        fclose(f);
        Sleep(1000); if(GetByte()=='^')printf("Write page Done\n");
```

Attachment C

In text mode, the OLED panel provide a 8row X 21col display text characters, each character are 5x7 and occupy a 6x8 square of pixel block, OLED module use ASCII code to store each character font in internal ROM, and use Display String command [0x04] to put characters on screen.

Col 0 Col20

Row 0			
Row 1			
Row 2			
Row 3			
Row 4			
Row 5			
Row 6			
Row 7			

Attachment D

Character Set

0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27
space	!	“	#	$	%	&	‘

0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F
(	)	*	+	,	-	.	/

0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37
0	1	2	3	4	5	6	7

0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F
8	9	:	;	<	=	>	?

0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47
@	A	B	C	D	E	F	G

0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F
H	I	J	K	L	M	N	O

0x50	0x51	0x52	0x53	0x54	0x55	0x56	0x57
P	Q	R	S	T	U	V	W

0x58	0x59	0x5A	0x5B	0x5C	0x5D	0x5E	0x5F
X	Y	Z	]	¥	[	^	_

0x60	0x61	0x62	0x63	0x64	0x65	0x66	0x67
`	a	b	c	d	e	f	g

0x68	0x69	0x6A	0x6B	0x6C	0x6D	0x6E	0x6F
h	i	j	k	l	m	n	o

0x70	0x71	072	0x73	0x74	0x75	0x76	0x77
p	q	r	s	t	u	v	w

0x78	0x79	0x7A	0x7B	0x7C	0x7D	0x7E	0x7F
x	y	z	{	|	}	->	<-