

CE labs literature Doc#: DVP44rs232may2014

## DVPS44



### The Command Protocol Format (RS-232 Serial Port)

#### Serial Port Setting:

Baud rate : 115200 bps  
Data bit : 8 bits  
Parity : None

#### Check Sum

The sum of Byte( 0) to Byte( n-1) and don't care carry number.

Below implement is written in C language

```
unsigned char CalculateChecksum(unsigned char* Array, unsigned int ArrayNum)
{
    unsigned char CheckSum;
    unsigned int loop;
    for (loop=0; loop<ArrayNum; loop++)
        CheckSum += Array[loop];
    return ChekcSum;
}
```

<b>Function: Set Output Ports Mapping Single or All outputs at one's</b>			
Command			
Byte	Name	Value	Comment
0	Header 1	0x7a	
1	Header 2	0xc1	
2	Header 3	0x5f	
3	Command Length	0x07	Length of byte3~byte9
4	Check Code	0x02	
5	Command ID	0x06	
6	Operation	0x01	0x01:set single output port 0x00:set all output ports
7	Output port number	0x01	Output port number 1~4 Mute output:0xc1~0xc4 <b>(This field is discarded if operation is 0x00)</b>
8	Input port number	0x03	Input port number
9	Check Sum	0x14	BYTE CheckSum = 0; for(int i =3; i<n; i++ ) CheckSum += Byte(i);
Acknowledgement character			
0	ACK/NACK	0x03	
1	ACK/NACK	0x81	
2	ACK/NACK	0x84	
Example: single input to single output and all output to single input			
<b>Select: Single output to single input.</b>			
Out #1 to In #1:	0x7a 0xc1 0x5f 0x07 0x02 0x06 0x01 0x01 0x01 0x12		
Out #1 to In #4:	0x7a 0xc1 0x5f 0x07 0x02 0x06 0x01 0x01 0x04 0x15		
Out #2 to In #1:	0x7a 0xc1 0x5f 0x07 0x02 0x06 0x01 0x02 0x01 0x13		
Out #2 to In #4:	0x7a 0xc1 0x5f 0x07 0x02 0x06 0x01 0x02 0x04 0x16		
Out #3 to In #1:	0x7a 0xc1 0x5f 0x07 0x02 0x06 0x01 0x03 0x01 0x14		
Out #3 to In #4:	0x7a 0xc1 0x5f 0x07 0x02 0x06 0x01 0x03 0x04 0x17		
Out #4 to In #1:	0x7a 0xc1 0x5f 0x07 0x02 0x06 0x01 0x04 0x01 0x15		
Out #4 to In #4:	0x7a 0xc1 0x5f 0x07 0x02 0x06 0x01 0x04 0x04 0x18		
<b>Select: All outputs to single input.</b>			
All out to In #1:	0x7a 0xc1 0x5f 0x06 0x02 0x06 0x00 0x01 0x0f		
All out to In #2:	0x7a 0xc1 0x5f 0x06 0x02 0x06 0x00 0x02 0x10		
All out to In #3:	0x7a 0xc1 0x5f 0x06 0x02 0x06 0x00 0x03 0x11		
All out to In #4:	0x7a 0xc1 0x5f 0x06 0x02 0x06 0x00 0x04 0x12		

<b>Function Description: Get routing Status</b>			
Command			
Byte	Name	Value	Comment
0	Header 1	0x7a	
1	Header 2	0xc1	
2	Header 3	0x5f	
3	Command Length	0x04	Length of byte3~byte6
4	Check Code	0x02	
5	Command ID	0x01	
6	Check Sum	0x07	BYTE CheckSum = 0; for(int i =3; i<n; i++ ) CheckSum += Byte(i);
Acknowledgement character			
0	ACK/NACK	0x03	
1	ACK/NACK	0x81	
2	ACK/NACK	0x84	
Receive			
0	Receive Parameter 0	0x01	Output port "1" is mapping to Input port 1
1	Receive Parameter 1	0x01	Output port "2" is mapping to Input port 1
2	Receive Parameter 2	0x01	Output port "3" is mapping to Input port 1
3	Receive Parameter 3	0x01	Output port "4" is mapping to Input port 1
4	Receive Parameter 4	0x00	1:There is a signal on input1 0:There isn't a signal on input1
5	Receive Parameter 5	0x00	1:There is a signal on input2 0:There isn't a signal on input2
6	Receive Parameter 6	0x00	1:There is a signal on input3 0:There isn't a signal on input3
7	Receive Parameter 7	0x00	1:There is a signal on input4 0:There isn't a signal on input4
Command:			
0x7a 0xc1 0x5f 0x04 0x02 0x01 0x07			

**Function Description: Save and Preset Mapping Status**

--	--	--	--

**Command**

Byte	Name	Value	Comment
0	Header 1	0x7a	
1	Header 2	0xc1	
2	Header 3	0x5f	
3	Command Length	0x06	Length of byte3~byte8
4	Check Code	0x02	
5	Command ID	0x05	
6	Operation	0x00	0x01:preset mapping 0x00:save mapping
7	block	0x00	Block number (0~7)
8	Check Sum	0x0d	BYTE CheckSum = 0; for(int i =3; i<n; i++ ) CheckSum += Byte(i);

**Acknowledgement character**

0	ACK/NACK	0x03	
1	ACK/NACK	0x81	
2	ACK/NACK	0x84	

**Example**

**Save mapping:**  
 Save mapping on block 0: 0x7a 0xc1 0x5f 0x06 0x02 0x05 0x00 0x00 0x0d  
 Save mapping on block 1: 0x7a 0xc1 0x5f 0x06 0x02 0x05 0x00 0x01 0x0e  
 Save mapping on block 2: 0x7a 0xc1 0x5f 0x06 0x02 0x05 0x00 0x02 0x0f  
 Save mapping on block 3: 0x7a 0xc1 0x5f 0x06 0x02 0x05 0x00 0x03 0x10  
 Save mapping on block 4: 0x7a 0xc1 0x5f 0x06 0x02 0x05 0x00 0x04 0x11  
 Save mapping on block 5: 0x7a 0xc1 0x5f 0x06 0x02 0x05 0x00 0x05 0x12  
 Save mapping on block 6: 0x7a 0xc1 0x5f 0x06 0x02 0x05 0x00 0x06 0x13  
 Save mapping on block 7: 0x7a 0xc1 0x5f 0x06 0x02 0x05 0x00 0x07 0x14

**Preset mapping:**  
 Preset mapping on block 0: 0x7a 0xc1 0x5f 0x06 0x02 0x05 0x01 0x00 0x0e  
 Preset mapping on block 1: 0x7a 0xc1 0x5f 0x06 0x02 0x05 0x01 0x01 0x0f  
 Preset mapping on block 2: 0x7a 0xc1 0x5f 0x06 0x02 0x05 0x01 0x02 0x10  
 Preset mapping on block 3: 0x7a 0xc1 0x5f 0x06 0x02 0x05 0x01 0x03 0x11  
 Preset mapping on block 5: 0x7a 0xc1 0x5f 0x06 0x02 0x05 0x01 0x05 0x13  
 Preset mapping on block 7 : 0x7a 0xc1 0x5f 0x06 0x02 0x05 0x01 0x07 0x15

**Function Description : Get TV Wall Parameter**

For all received values please take a reference in software TV wall GUI.

**Command**

Byte	Name	Value	Comment
0	Header 1	0x7a	
1	Header 2	0xc1	
2	Header 3	0x5f	
3	Command Length	0x05	Length of byte3~byte7
4	Check Code	0x02	
5	Command ID	0x0c	
6	Output port number-1	0x00	value 0~3 represents output 1~4
7	Check Sum	0x13	BYTE CheckSum = 0; for(int i =3; i<n; i++ ) CheckSum += Byte(i);

**Acknowledgement character**

0	ACK/NACK	0x03	
1	ACK/NACK	0x81	
2	ACK/NACK	0x84	

**Receive**

0	Receive Parameter 0	0x01	Select
1	Receive Parameter 1	0x01	H divide
2	Receive Parameter 2	0x01	V divide
3	left high byte	0x01	masking value must be minus 500 and the range is -500~500 for example: left masking=0x01*256+0xf4-500=0
4	left low byte	0xf4	
5	right high byte	0x01	
6	right low byte	0xf4	
7	upper high byte	0x01	
8	upper low byte	0xf4	
9	bottom high byte	0x01	
10	bottom low byte	0xf4	

**Example**

TV wall parameter Output#1 : 0x7a 0xc1 0x5f 0x05 0x02 0x0c 0x00 0x13  
 TV wall parameter Output#4 : 0x7a 0xc1 0x5f 0x05 0x02 0x0c 0x03 0x13

### Set TV Wall Parameter

Function Description: For all received values please take a reference in software TV wall GUI.

#### Command

Byte	Name	Value	Comment
0	Header 1	0x7a	
1	Header 2	0xc1	
2	Header 3	0x5f	
3	Command Length	0x10	Length of byte3~byte18
4	Check Code	0x02	
5	Command ID	0x00	
6	Output port number-1	0x00	value 0~3 represents output 1~4
7		0x01	Select
8		0x01	H divide
9		0x01	V divide
10	left high byte	0x01	masking value must be add 500 and the range is -500~500 for example:set left masking to 0 high byte = 500/256=1 low byte = 500-256=244
11	left low byte	0xf4	
12	right high byte	0x01	
13	right low byte	0xf4	
14	upper high byte	0x01	
15	upper low byte	0xf4	
16	bottom high byte	0x01	
17	bottom low byte	0xf4	
18	Check Sum	0xe9	BYTE CheckSum = 0; for(int i =3; i<n; i++) CheckSum += Byte(i);

#### Acknowledgement character

0	ACK/NACK	0x03	
1	ACK/NACK	0x81	
2	ACK/NACK	0x84	

Example:

Set output#1 to full screen :

**0x7a 0xc1 0x5f 0x10 0x02 0x00 0x00 0x01 0x01 0x01 0x01 0xf4 0x01 0xf4 0x01 0xf4 0xe9**

Set output#1 to 2x2 wall processor and select part 1 for this output (on 2x2 are 4 parts):

**0x7a 0xc1 0x5f 0x10 0x02 0x00 0x00 0x01 0x02 0x02 0x01 0xf4 0x01 0xf4 0x01 0xf4 0xeb**

Set output#2 to 2x2 wall processor and select part 2 for this output (on 2x2 are 4 parts):

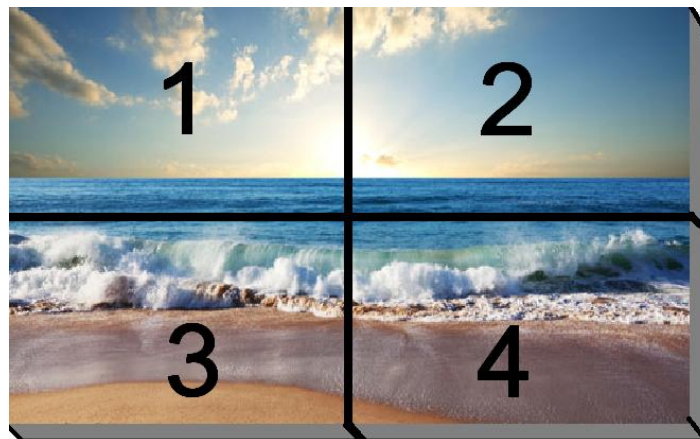
**0x7a 0xc1 0x5f 0x10 0x02 0x00 0x01 0x02 0x02 0x02 0x01 0xf4 0x01 0xf4 0x01 0xf4 0xed**

Set output#3 to 2x2 wall processor and select part 3 for this output (on 2x2 are 4 parts):

**0x7a 0xc1 0x5f 0x10 0x02 0x00 0x02 0x03 0x02 0x02 0x01 0xf4 0x01 0xf4 0x01 0xf4 0xef**

Set output#4 to 2x2 wall processor and select part 4 for this output (on 2x2 are 4 parts):

**0x7a 0xc1 0x5f 0x10 0x02 0x00 0x03 0x04 0x02 0x02 0x01 0xf4 0x01 0xf4 0x01 0xf4 0xf1**



Get Output Resolution			
Function Description:			
Command			
Byte	Name	Value	Comment
0	Header 1	0x7a	
1	Header 2	0xc1	
2	Header 3	0x5f	
3	Command Length	0x06	Length of byte3~byte8
4	Check Code	0x02	
5	Command ID	0x0f	
6	Operation	0x00	
7	input port number-1	0x00	value 0~3 represents input 1~4
8	Check Sum	0x17	BYTE CheckSum = 0; for(int i=3; i<n; i++) CheckSum += Byte(i);
Acknowledgement character			
0	ACK/NACK	0x03	
1	ACK/NACK	0x81	
2	ACK/NACK	0x84	
Receive			
0	Receive Parameter 0	0x0c	0x00:640x480@60 0x01:480p 0x02:576p 0x03:800x600@60 0x04:1024x768@60 0x05:720p@60 0x06:1280x768@60 0x07:1280x960@60 0x08:1280x1024@60 0x09:1366x768@60 0x0a:1440x900@60 0x0b:1680x1050@60 0x0c:1080p@60 0x0d:1920x1200@60 0x0e:1600x1200@60
Example			
Get output #1 resolution : 0x7a 0xc1 0x5f 0x06 0x02 0x0f 0x00 0x00 0x17			
Get output#4 resolution : 0x7a 0xc1 0x5f 0x06 0x02 0x0f 0x00 0x03 0x1a			



Set Output Resolution			
Function Description: set Output resolution			
Command			
Byte	Name	Value	Comment
0	Header 1	0x7a	
1	Header 2	0xc1	
2	Header 3	0x5f	
3	Command Length	0x07	Length of byte3~byte9
4	Check Code	0x02	
5	Command ID	0x0f	
6	Operation	0x01	
7	input port number-1	0x00	value 0~3 represents input 1~4
8	resolution	0x0c	0x00:640x480@60 0x01:480p 0x02:576p 0x03:800x600@60 0x04:1024x768@60 0x05:720p@60 0x06:1280x768@60 0x07:1280x960@60 0x08:1280x1024@60 0x09:1366x768@60 0x0a:1440x900@60 0x0b:1680x1050@60 0x0c:1080p@60 0x0d:1920x1200@60 0x0e:1600x1200@60
9	Check Sum	0x25	BYTE CheckSum = 0; for(int i =3; i<n; i++ ) CheckSum += Byte(i);
Acknowledgement character			
0	ACK/NACK	0x03	
1	ACK/NACK	0x81	
2	ACK/NACK	0x84	

Example set output resolution:	
Set Output#1 to 1080P :	0x7a 0xc1 0x5f 0x07 0x02 0x0f 0x01 0x00 0x0c 0x25
Set Output#2 to 480P :	0x7a 0xc1 0x5f 0x07 0x02 0x0f 0x01 0x01 0x01 0x1b
Set Output#3 to 720P :	0x7a 0xc1 0x5f 0x07 0x02 0x0f 0x01 0x02 0x05 0x20
Set Output#4 to 576P :	0x7a 0xc1 0x5f 0x07 0x02 0x0f 0x01 0x03 0x02 0x1e

Set TV Wall Group settings			
Function Description:			
Command			
Byte	Name	Value	Comment
0	Header 1	0x7a	
1	Header 2	0xc1	
2	Header 3	0x5f	
3	Command Length	0x05	Length of byte3~byte7
4	Check Code	0x02	
5	Command ID	0x16	
6	Operation	0x01	1~4: Group1~Group4
7	Check Sum	0x1e	BYTE CheckSum = 0; for(int i =3; i<n; i++ ) CheckSum += Byte(i);
Acknowledgement character			
0	ACK/NACK	0x03	
1	ACK/NACK	0x81	
2	ACK/NACK	0x84	
Command:			
Recall group setting #1		0x7a 0xc1 0x5f 0x05 0x02 0x16 0x01 0x1e	
Recall group setting #2		0x7a 0xc1 0x5f 0x05 0x02 0x16 0x02 0x1f	
Recall group setting #3		0x7a 0xc1 0x5f 0x05 0x02 0x16 0x03 0x20	
Recall group setting #4		0x7a 0xc1 0x5f 0x05 0x02 0x16 0x04 0x21	