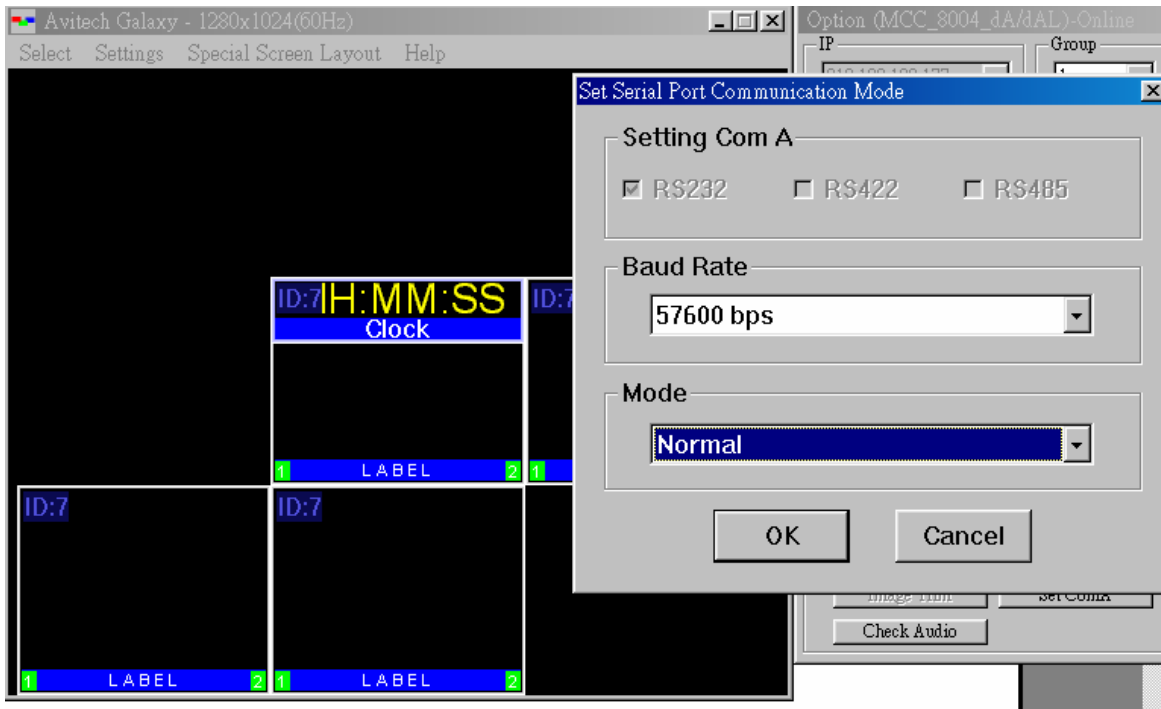


<<RS232 port Description>>

When module power on initial, RS232 default baud rate would be 115200 bps and it will send boot messages. After boot complete,(around 10 seconds), RS232 port baud rate will switch to what it set by PC's tool. The RS232 mode must set to "Normal" to support command prompt mode:



COM A Normal mode supports command prompt by using terminal like tool, the serial port configuration have to be : 8 data bits, no parity, 1 stop bit, no flow control.

After making connection with the modules and terminal, hit first "Enter" key to enable command prompt. You'll see the following message on the terminal:

```
port1 - 終端機
檔案(E) 編輯(E) 檢視(V) 呼叫(C) 傳送(T) 說明(H)
Welcome to AVITECH ASCII Command Shell. (Version 2.0)
Initializing.....O.K.!

>za 000000
>zb 000000 label 1
>zc 000000 label 000100000
>zw 000005 0 0 100 80
>_
>_
```

<<Z text commands>>

Common format

Member/Window assignment: GGMMPP

GG = Group "01", "02", ... "99"; "00" = ALL Groups

MM = Module "01", "02", ... "15"; "00" = ALL Modules

PP = Processor "01" ~ "04":Image windows; "98"=logo;"99"=clock; "00" = ALL image Processors

Color assignment: RRRGGGBBB

RRR = Red strength "000" ~ "255"

GGG = Green strength "000" ~ "255"

BBB = Blue strength "000" ~ "255"

ON/OFF switch: ON/OFF

ON/OFF "1":On, "0":Off

ZA = window auto Arrangement

Format: ZA GGMMPP(accept clock) [NByN(2,3,.....)] [Nth(1,2,.....)]

Example:

ZA 010900 2 1 → Set Group 1, Module 9, to 2x2 map position 1,2,3,4(quad).

ZA 010000 2 1 → Set Group 1, All modules to quad.

ZA 010202 3 2 → Place window 010202 to 3x3 map position 2.

ZA 010200 6 13 → Place group 1 module 2 to 6x6 map position 13,14,15,16.

ZA 000000 → Auto arrange all Groups' all windows to proper size and position.

ZB = set border or label Blinking

Format: ZB GGMMPP B[order]/L[abel] ON/OFF

Example:

ZB GGMMPP label 0 → GGMMPP label blink off.

ZB GGMMPP b 1 → GGMMPP border blink on

ZC = set border/clock/label background Color

Format: ZC GGMMPP(accept clock) B[order]/L[abel] RRRGGGBBB [NoDimColor]

Example:

ZC GGMMPP border 000255000 → Set GGMMPP border dim GREEN

ZC GGMMPP border 000255000 ndc → Set GGMMPP border pure GREEN

ZC GGMMPP border 000000000 → Turn GGMMPP border off

ZC GGMMPP label 255000000 → Set GGMMPP label background RED

ZC GGMM99 border 255000255 → Set clock color PINK

ZC GGMM99 label 255000255 → Set clock label background PINK

ZE = set Echo

Format: ZE GGMM ON/OFF

Description: Command response would be much faster when echo turned off.

Example:

ZE GGMM 0 → Set module GGMM RS232 echo off

ZF = set window Full screen

Format: ZF GGMMPP ON/OFF

Example:

ZF GGMMPP 1 → Set GGMMPP full screen
ZF 020000 0 → Set group 2 to not full screen status

ZI = set Input channel

Format: ZI GGMMPP(accept clock) channel

for video window:

channel : 1~4

for clock:

channel : 1=Internal, 2=DVI, 3=VITC-NTSC, 4=LTC, 5=VITC-PAL

Example:

ZI GGMM00 1 → Set GGMM all window source to channel 1

ZI GGMM99 2 → Set GGMM clock time sync to DVI.

ZK = set clock time & count way

Format for preset time: ZK GGMM P[reset] S[et]/L[oad]/[ID_of_analog_clock]

PresetID(1~8) HH MM SS

Example:

ZK GGMM preset S 1 11 22 33 → Set GGMM preset time ID1 11:22:33

ZK GGMM preset L 1 → Set GGMM time as preset time ID1

ZK GGMM preset id 2 → Using #2 analog clock

Format for control: ZK GGMM [HH MM SS] [count way]

Count way = 0: count down, 1: count up, 2: pause, 3: not pause, 4: invert pause status

Example:

ZK GGMM 11 22 33 → Set GGMM time 11:22:33

ZK GGMM 11 22 33 0 → Set GGMM time 11:22:33 and count down

ZK GGMM 1 → Set GGMM clock count up

ZL = set Label transparency, text color and text

Format: ZL GGMMPP(accept clock) transparency TextRRRGGG BBB LabelRRRGGG BBB

["TEXT"]

transparency "00" "255"

TextRRRGGG BBB&LabelRRRGGG BBB "000000000" = No Change

"TEXT" = label text string, 50 ASCII characters maximum.

Example:

ZL GGMMPP 0 255000000 000000255 " CNN News Station "

→ Set GGMMPP No Transparency, center justify, Text color red, Label color blue,
with text " CNN News Station "

Note: All windows will use the only and the same label transparency. That is, change one window's label transparency will affect all windows label transparency.

ZM = set display resolution Mode

Format: ZM GGMMPP mode [NoAutoArrangement]

Mode ID list :

1 : 800x600/60Hz

2 : 1024x768/60Hz

15 : 1280x720/60Hz

22 : 1280x768/60Hz

9 : 1280x1024/60Hz

23 : 1280x768/60Hz (SONY LMD230)

20 : 1360x768/60Hz
35 : 1400x1050/60Hz
27 : 1400x1050/60Hz (JVC)
10 : 1600x1200/60Hz
26 : 1920x1080/60Hz
36 : 1920x1200/60Hz

42 : 800x600/50Hz
31 : 1024x768/50Hz
30 : 1280x720/50Hz
32 : 1280x768/50Hz
29 : 1280x1024/50Hz
38 : 1360x768/50Hz
34 : 1400x1050/50Hz
33 : 1400x1050/50Hz (JVC)
39 : 1600x1200/50Hz
28 : 1920x1080/50Hz
37 : 1920x1200/50Hz
253 : color graphic card timing.
254 : frame lock timing.
255 : VESA timing.

Example:

ZM 010000 10 → Set Group 1 to 1600x1200/60Hz and auto arrange all windows to proper size and position.
ZM 000000 9 NA → Set All Groups to 1280x1024/60Hz, no auto arrangement.

ZN = on/off control

Format: ZN GGMMPP(accept clock) switch ON/OFF
switch can be :

A[larm]/B[order]/C[lockControl]/DisplayClosedCaption/F[PGA_selection]/L[abel]/M[eter]/N[eedle_of_clock]/R[atioAspect]/S[afeArea]/V[ideoFormatDisplay]/W[indow]

Example:

ZN GGMM99 W 0 → Turn GGMM clock off.
ZN GGMMPP Border 1 → Turn GGMMPP border on.

To select main board FPGA0 or main board FPGA1:

ZN GGMMPP FPGA 0/1 → Select FPGA
ZP GGMM Save → Save settings to flash

Reset machine manually.....

ZO = set audio Output

Format to initialize audio source : ZO GGMMPP I[nitalize] Lgroup Rgroup freq
Lgroup,Rgroup "1", "2", "3", "4" else AES
freq "60"=60Hz else 59.94Hz

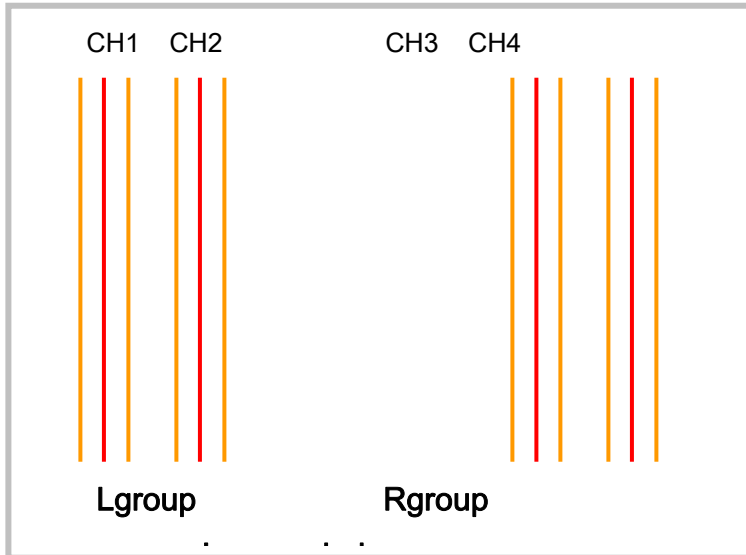
Example:

ZO GGMMPP Init 1 AES 60 → Set GGMMPP Lgroup=1 Rgroup=AES 60Hz

Format to set audio output : ZO GGMMPP ON/OFF [channel]

Example:

ZO 000000 0 → Turn audio off
ZO 020100 1 → Turn group 2 module 1 audio on
ZO 000203 1 4 → Turn all group module 2 processor 3 CH4 audio on



ZP = Preset load/save to RAM file. Save to flash.

Format for load : ZP GGMMPP L[oad] filename

Format for save : ZP GGMMPP S[ave] [filename]

Example:

ZP 000000 load stage1 → All modules load file "stage1"

ZP 020000 s file2.GP2 → Group 2 save current status "file2.GP2" to RAM.

ZP 000000 SaveToFlash → All modules save all configurations to flash.

ZP 000000 s " file name with spaces "

→ must quote the filename when there're spaces.

ZR = lock and adjust video Ratio

Format: ZR GGMMPP SDWidthRate SDHeightRate HDWidthRatio HDHeightRatio

Example:

ZR GGMM01 4 3 16 9 → Set GGMM01 SD video ratio as 4:3, HD video as 16:9

ZR 000000 16 9 4 3 → Set all windows' SD video ratio as 16:9, HD video as 4:3

ZR GGMMPP 7 12 7 12 → Set GGMMPP's SD HD video ratio as 7:12

ZR GGMMPP 0 0 0 0 → Disable the function by WidthRate or HeightRate=0

ZT = set Tally on/off and color

Format: ZT GGMMPP tally(1~3) ON/OFF [color]

tally = Tally Block "1", "2", "3"

color list:

1. null.

2. red

3. green

4. yellow

5. blue

6. pink
7. light blue
8. white

Example:

ZT 000000 2 1 6 → Set all window's tally block 2 to pink and ON.

ZT 010203 2 0 → Turn 010203 tally 2 off.

Note: All windows' tally 1 share the same color setting, so does tally 2, tally 3. That is, change one window's tally 1 color, all window's tally 1 color will be changed.

ZV = set analog audio Volume level

Format: ZV GGMMPP volume(default=32,range 0~127)

Example:

ZV GGMM01 0 → Set GGMM window 1 analog audio to mute.

ZW = set top Window position, size of a module

Format: ZW GGMMPP(accept clock) [X Y width height]

Example:

ZW GGMM01 → Set processor 1 as top window of GGMM.

ZW GGMMPP 100 200 300 400 → Set GGMMPP at (100,200)-(400,600)

ZW GGMMPP X Y 0 0 → Turn GGMMPP off by width or height=0

ZX = set label text and font size inside/outside video

Format: ZX GGMMPP(accept clock) ["TEXT"] [FontSize]

FontSize = 0, Current font size inside video

FontSize = 1~5, Label inside video

FontSize = 128, Current font size outside video

FontSize = 129~133, Label outside video

Example:

ZX 000000 " L A B E L " → Set all windows' label as " L A B E L ".

ZX 000000 3 → Set all window's label font size 3.

ZX 000000 " Outside video " 128

→ Set all windows' label as " Outside video " and outside video.

<<Serial port commands execution timing example>>

=====

HOST(CONTROLLER) sends MCC8004 sends

=====

/ Host sends the first ENTER to enable the service */*

First ENTER

/ After receiving First ENTER from host, MCC8004 is busying initialization, */*

/ When done, MCC8004 sends '>' character to indicate the completion. */*

Welcome to AVITECH ASCII Command Shell. (Version 2.0)

Initializing.....O.K.!

>

/ Host sends the following messages in a burst */*

Command1+ENTER

Command2+ENTER

Command3

Command1+ENTER

/ The message echoed back from MCC8004 seems like per-character basis. */*

/ MCC8004 is busying executing "Command1" now. */*

/ When done, MCC8004 sends '>' character to indicate the completion. */*

>

/ There is still Command2+ENTER queued in buffer */*

Command2+ENTER

/ MCC8004 is busying executing "Command2" now. */*

/ When done, MCC8004 sends '>' character to indicate the completion. */*

>

/ There is still Command3 queued in buffer */*

Command3

/ MCC8004 just echoes back Command3 due to the command haven't ended by ENTER */*

ENTER

ENTER

/ MCC8004 is busying executing "Command3" now. */*

/ When done, MCC8004 sends '>' character to indicate the completion. */*

>

=====

<<Flow control recommendation of serial port>>

It's recommended to wait module's ready for command character '>' after each command sent.

<<Execute Z text commands & TSL via network>>

It's capable to transmit and execute command prompt by formal connect method via network/RS232(not recommended).

By formal connect method, control application have to make first connection with master module:

<<Implement socket connection first>>

Network connection build at the moment control application make socket connection with module using port number 20036, module will send the following information immediately when socket connection complete:

```
unsigned char ModuleID; // !!! Keep this value if you need handshake procedure !!!
unsigned char ModuleTypeID;
unsigned char 0x00,0xFF,0x5A,0xA5,0x01;
```

Module will accept any other command after connection implement, control application can drive text command prompts by following packet after connection:

<<Command to transmit text command prompt>>

Command format :

```
unsigned char 0x55,0xAA; //command head
unsigned char 0xF8,~0xF8; //command ID
unsigned char strlen(prompt$)+2,~(strlen(prompt$)+2); //command length
unsigned char 0x07; //must be 0x07
unsigned char prompt$[]; //must end by 0
unsigned char 0; //command tail
```

Example :

```
0x55 0xAA 0xF8 0x07 0x0B 0xF4 0x07 "ZA 000000" 0x00
```

→ Execute "ZA 000000"=Arranges all windows to proper size and position

Note :

The quote "" of example string "ZA 000000" is just for string expression, there's not quote characters stays in command contents, actual memory dump of command

```
0x55 0xAA 0xF8 0x07 0x0B 0xF4 0x07 "ZA 000000" 0x00
```

Is 0x55 0xAA 0xF8 0x07 0x0B 0xF4 0x07

```
0x5A 0x41 0x20 0x30 0x30 0x30 0x30 0x30 0x30 0x00 0x00
```

<<Command to transmit TSL data>>

Command format :

```
unsigned char 0x55,0xAA; //command head
unsigned char 0xF8,~0xF8; //command ID
unsigned char 0x13,~0x13; //command length
unsigned char 0x0A; //must be 0x0A
unsigned char TSL[18]; //18 bytes TSL data
unsigned char 0; //command tail
```

<<Flow control recommendation of socket connection>>

It's recommended to execute handshake procedure if your application has to send lots of commands in a burst.

*** Execute handshake procedure per n(8 or 16 is recommended) commands sent. ***

//Handshake procedure

Send handshake packet for flow control :

0x55 0xAA ((ModuleID<<4)|0x09) ~ ((ModuleID<<4)|0x09) 0x00 0xFF 0x00

Where : ModuleID is the value module replied at the socket connection moment.

Wait 13 bytes reply from module after handshake packet sent :

0x55 0xAA xx ~xx yy ~yy zz zz 0x00 0xFF 0x5A 0xA5 0x01

<<Stop network socket connection before control application terminate>>

Before control application terminate, control application have to stop socket connection to maintain the module's link state normally.

