The “VCOM Interface” is existed in the NS XpressROM BIOS for providing a convenience way to set/get the status of COP8. For more powerful and sophisticated usage, the extra commands had been supported to access system status..

Currently, we support 7 commands and two events in BIOS. The following is the command summary.

Command:

Draco access command

Command	Value	Informally Description
ENTER_COMMAND_PHASE	0XE0	Inform VCOM Interface ready to send command to COP8
EC_COMMAND_SEND	0XE1	Send command to COP8
ENTER_RECEIVE_PHASE	0XE2	Inform VCOM Interface ready to get data form COP8
EC_RECEIVE_DATA	0XE3	Receive data

LCD ON/OFF command

Command	Value	Informally Description
TURN_OFF_BACKLIGHT	0XB0	Turn the LCD panel backlight OFF
TURN_ON_BACKLIGHT	0XB1	Turn the LCD panel backlight ON

Native audio support command

Command	Value	Informally Description
NAUDIO_FLAG_SET	0XE5	Enable Native audio driver support

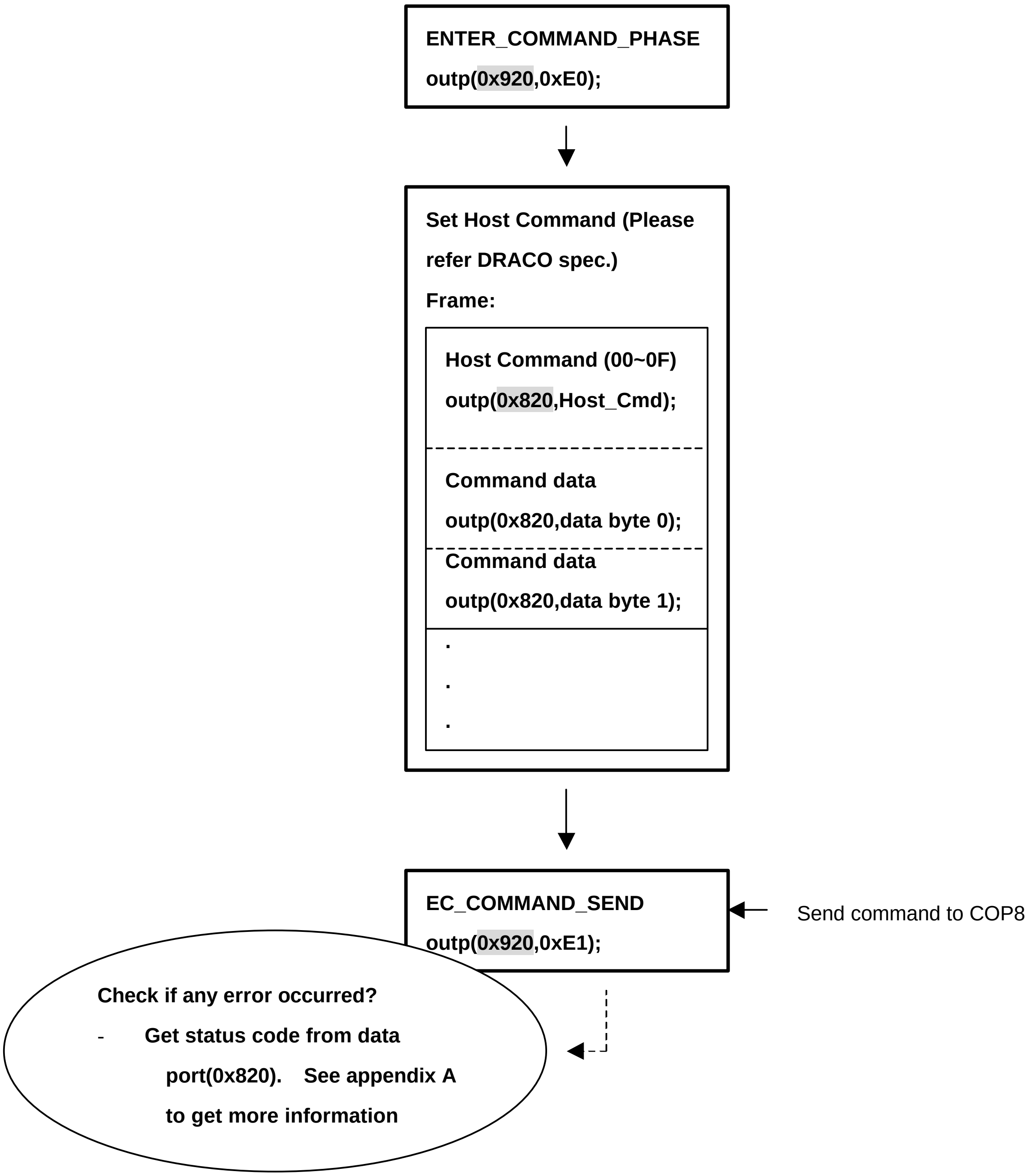
Event: (returned form VCOM2)

Event	Informally Description
Button event	Bit7 set to 1
Power event	0x3?

Command Usage:

Very Important: please mask the IRQ9 whenever you want to send any command to VCOM Interface.
Refer Appendix B to get detail description.

1. Draco (COP8) access command
   Send Command: 0x920 for command port,0x820 for data port



ENTER_COMMAND_PHASE

outp(0x920,0xE0);

Set Host Command (Please refer DRACO spec.)

Frame:

Host Command (00~0F)

outp(0x820,Host_Cmd);

Command data

outp(0x820,data byte 0);

Command data

outp(0x820,data byte 1);

.
.
.

EC_COMMAND_SEND

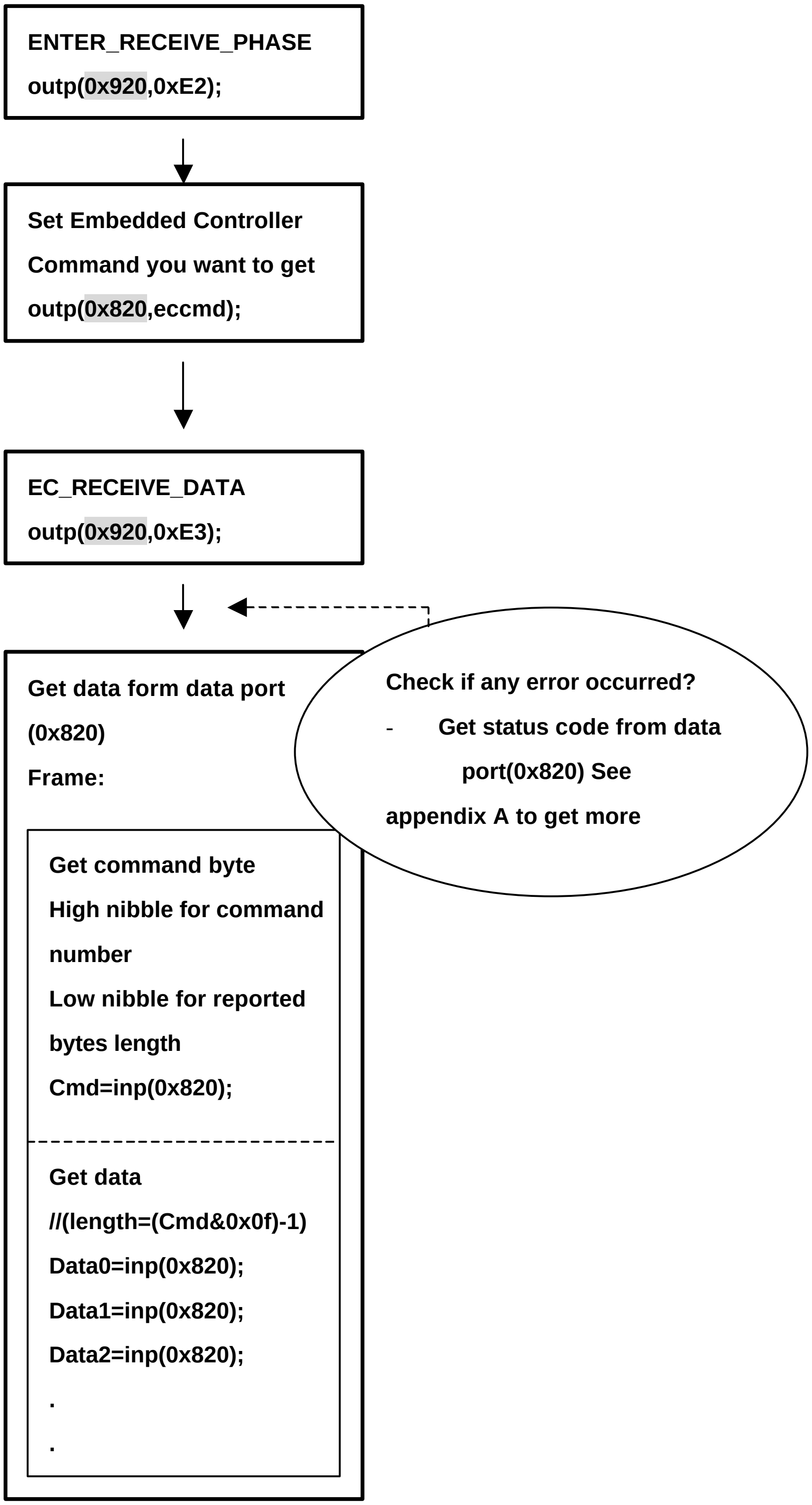
outp(0x920,0xE1);

Send command to COP8

Check if any error occurred?

- **Get status code from data port(0x820). See appendix A to get more information**

Receive Data: 0x920 for command port,0x820 for data port



ENTER_RECEIVE_PHASE

outp(0x920,0xE2);

Set Embedded Controller

Command you want to get

outp(0x820,eccmd);

EC_RECEIVE_DATA

outp(0x920,0xE3);

Get data form data port

(0x820)

Frame:

Get command byte

High nibble for command

number

Low nibble for reported

bytes length

Cmd=inp(0x820);

Get data

//(length=(Cmd&0x0f)-1)

Data0=inp(0x820);

Data1=inp(0x820);

Data2=inp(0x820);

.

.

Check if any error occurred?

- Get status code from data port(0x820) See

appendix A to get more

2. LCD Panel Backlight Control

Turn off backlight

```
TURN_OFF_BACKLIGHT
outp(0x920,0xb0);
```

Turn on backlight

```
TURN_ON_BACKLIGHT
outp(0x920,0xb1);
```

3. Native audio driver support

Turn on backlight

```
NAUDIO_FLAG_SET
outp(0x920,0xE5);
```

Event model:

The event will be report form data port(0x820) IRQ9. Please hook an interrupt service handler to IRQ9 to process event.

There are two kinds of event will be report that supported by the currently released Xpress BIOS from data port.

When any event occurred, you can get one byte form data port(0x820). They means:

1. Button event

   Format: bit 7 has been set to 1

   Any value greater or equal than 0x80, that's the button event.

   Bit0~5 means respective button has been pressed or released.

2. Power event

   Format: 0x3?

   If ? = 1, that's the AC event.

   If ? = 2, that's the battery event.

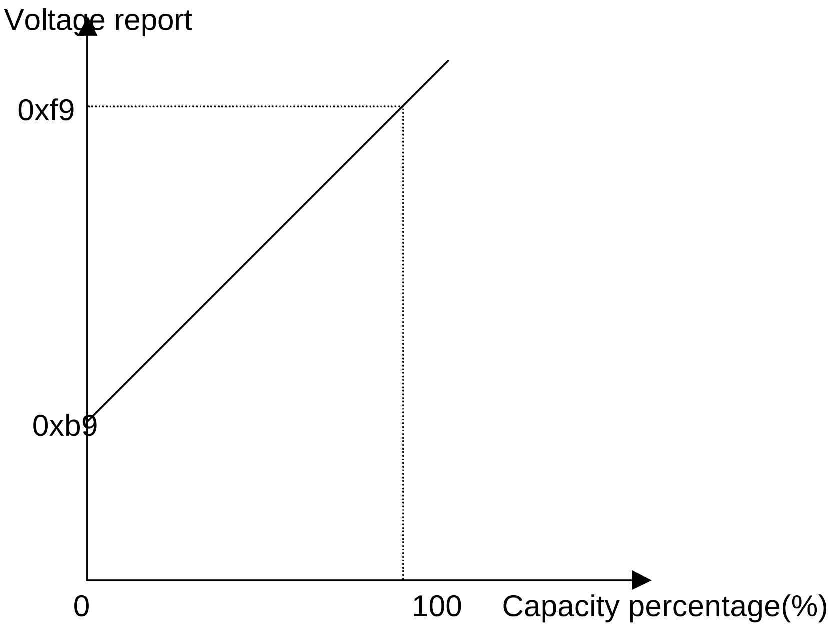
   You can get the detail information by receiving data form COP8.(Please refer the **Receive Data** place in this document.

Battery Gas Gauge report:

Currently, the COP8 support the battery status report only. You have no way to get the battery gas gauge value. We have to add the battery gas gauge report with our next version of COP8 firmware.

Actually, we will report the battery voltage potential value, which is the linear dependence with battery capacity. You can compute the percentage of battery remaining capacity with the simply linear equation.

It will be look like.



You can get the voltage report with the second byte of EC_BATT_STAT command. You need to update the COP8 firmware and new Xpress BIOS of course.

Appendix – A

Error code definition

Format: 0x1?

? means :

```
#define NO_ERROR        0x00
#define Unknow_COMMAND_Error0x01
#define VCOM_BUFFER_OVERFLOW     0x02
#define Data_Package_Format_Err 0x03
#define Unenter_Command_Phase0x04
#define Unenter_Receive_Phase    0x05
#define Rev_Command_Error 0x06
#define Reserved_Command_Error0x07
#define sendPacketToEC_Error     0x08
#define sendPackageretry_Error    0x09
```

Appendix B:

Programming Guide

1. If your ISR get any data sent by VCOM data port. Check the bit7 first to see if it's a button event.If not, you can check the PostCode nibble to see what happen. Actually, the POWER_NOTIFY the the only one we support now.

   EX:

```
if (VCOM2Data&0x80){
// button data handler
}
else{
    switch(VCOM2Data&0xf0){
        case POWER_NOTIFY:
        // get the AC and Battery status
            switch(VCOM2Data&0x0f){
                case ACIN_NOTIFY:
                    //get AC status(IN/OUT) –
                    //send EC_ACIN_STAT command to get AC status
                  break;
                case BATT_NOTIFY:
                    // get battery status - send EC_BATT_STAT command
                break;
            }
        .....
        break;
    }
}
```

2. If you want to read some data form Draco(Embedded Controller).

1). disable the data port buffer full interrupt

EX:

```
outp(data_port+1,inp(data_port+1)&(~1));
```

2). send ENTER_RECEIVE_PHASE command to command port. (You can get the errCode from data port now,but I don't think that's necessary at this mean time.)

EX:

```
outp(command_port,ENTER_RECEIVE_PHASE);
```

3). send the EC command which you need to data port.

EX: 0x06 for PWM (You can get the command list from NS draco Embedded controllor spec.)

```
outp(data_port,0x06);
```

4). send the EC_RECEIVE_DATA command to command port. And get the status report form data port.It's should be '2x' if it's successful('2' for RECEIVE_EC_DATA PostCode, 'x' is the byte count you should to read). If it's successful, you can get your data form data_port now. (Very important,you must read all of the data out.Don't leave any byte there to make a mess of that.)

EX:

```
outp(command_port,EC_RECEIVE_DATA);
status=inp(data_port);
if((status & 0xf0)!=RECEIVE_EC_DATA){
    //error handle
    if((status & 0xf0)==Error_Report)
        switchc(status&0x0f){
        case Unknow_COMMAND_Error:
        .....
        break;
        case VCOM_BUFFER_OVERFLOW:
        .....
        break;
        .}}
else{
// receive data here
for(i=1;i<status&0x0f;i++)
{buffer[i-1]=inp(data_port);}
// You have got the data that saved into buffer[] now.}
```

5). enable the command port buffer full interrupt

EX:

```
outp(data_port+1,inp(data_port+1)|1);
```

3. Send command to Draco

1). disable the command port buffer full interrupt

EX:

```
outp(data_port+1,inp(data_port+1)&(~1));
```

2). send ENTER_COMMAND_PHASE command to command port.

EX:

```
outp(command_port,ENTER_COMMAND_PHASE);
```

3). send the data you want to send to Draco.

EX:Set PWM (You can get the command list from NS draco Embedded controllor spec.)

```
outp(command_port,0x06); //(0x06 for PWM)
outp(data_port,brightness);
outp(data_port,contrast);
```

4). send EC_COMMAND_SEND to trigger command phase to write command into Draco.

EX:

```
outp(command_port,EC_COMMAND_SEND);
```

5). get the status report form data_port to see if any error occured.

EX:

```
status=inp(data_port);
    //error handle
    if((status & 0xf0)==Error_Report)
        switchc(status&0x0f){
        case NO_ERROR:
        break;
        case Unknow_COMMAND_Error:
        .....
        break;
        case VCOM_BUFFER_OVERFLOW:
        .....
        break;
        .
        .
        .
    }
```

6). enable the command port buffer full interrupt

EX:

```
outp(data_port+1,inp(data_port+1)|1);
```

4. Set system to native audio mode

simply send NAUDIO_FLAG_SET command to command_port

```
outp(command_port,NAUDIO_FLAG_SET);
```

5. Turn ON/OFF LCD backlight

simply send LCD backlight command to command_port

For turnning off LCD backlight:

1). disable the command port buffer full interrupt

EX:

```
outp(data_port+1,inp(data_port+1)&(~1));
```

2). For turnning off LCD backlight

EX:

```
outp(command_port,TURN_OFF_BACKLIGHT);
```

6). enable the command port buffer full interrupt

EX:

```
outp(data_port+1,inp(data_port+1)|1);
```

For turnning on LCD backlight

1). disable the command port buffer full interrupt

EX:

```
outp(data_port+1,inp(data_port+1)&(~1));
```

2). For turnning on LCD backlight

EX:

```
outp(command_port,TURN_ON_BACKLIGHT);
```

6). enable the command port buffer full interrupt

EX:

```
outp(data_port+1,inp(data_port+1)|1);
```

summary:

1. For supporting scale contrast/brightness,battery basic information, set system mode (EX: auto shutdown).

Please refer item 2,3 to access Draco command(Please refer Draco Embedded controllor spec to get the full command list).

2. For supporting LCD backlight ON/OFF function. Please refer item 5.