Setting environmental

Download ODROID-SHOW source code first.
Then connect microUSB cable to the Host PC with ODROID-SHOW.
Finally check the serial port number and run bash shell script file.

Jumper must be not connected to ODROID-SHOW when you run sclript or echo

Download ODROID-SHOW source code

```
$ sudo apt-get install git
$ git clone https://github.com/hardkernel/ODROID-SHOW
```

Check the serial port number after connecting the micro-USB cable

```
$ ls /dev/ttyUSB*
```

Modify the serial port number

In the ODROID-SHOW/example/linux

```
#include <stdio.h>
#include <fcntl.h>
#include <termios.h>
#include <errno.h>

#define baudrate  B500000

const char serialPort[] = "/dev/ttyUSB0";

int main(void)
{
    int usbdev;
    struct termios options;

    usbdev = open(serialPort, O_RDWR | O_NOCTTY | O_NDELAY);

    if (usbdev == -1)
        perror("open_port : Unable to open:");

    tcgetattr(usbdev, &options);

    cfsetispeed(&options, baudrate);
    cfsetospeed(&options, baudrate);

    options.c_cflag |= CS8;
```

```
    options.c_iflag |= IGNBRK;
    options.c_iflag &= ~( BRKINT | ICRNL | IMAXBEL | IXON);
    options.c_oflag &= ~( OPOST | ONLCR );
    options.c_lflag &= ~( ISIG | ICANON | IEXTEN | ECHO | ECHOE | ECHOK |
ECHOCTL | ECHOKE);
    options.c_lflag |= NOFLSH;
    options.c_cflag &= ~CRTSCTS;

    tcsetattr(usbdev, TCSANOW, &options);

    while(1)
        sleep(0.2);

    return ;
}
```

Compile the port_open.c

```
$ gcc -o port_open port_open.c
```

Bash example

This bash script can display 2 text strings with different color and different font size.
Open the “/dev/ttyUSB0” port and sending VT100/ANSI commands with a couple of strings.
It also changes the color of strings.

Text output

In the ODROID-SHOW/example/linux/ODROID.sh

```
#!/bin/bash

flag=
serialPort="/dev/ttyUSB0"

trap "flag=1" SIGINT SIGKILL SIGTERM

./port_open &
subppid=$!

DATA[]="ODROID"
DATA[1]="SHOW"

echo -ne "\e[5s\e[0r" > $serialPort
sleep 0.1
```

```
echo -ne "\ec" > $serialPort
sleep 0.1

while true
do
    if [ $flag -ne  ] ; then
        echo -ne "\ec\e[2s\e[1r" > $serialPort
        kill $subppid
        exit
    fi
    for ((j=1; j<8; j++)); do
        echo -ne "\e[25;100f" > $serialPort
        for ((i=; i<6; i++)); do
            echo -ne "\e[3"$j"m\e[3"$j"m${DATA[0]:$i:1}" > $serialPort
            sleep 0.02
        done
        echo -ne "\eE\e[55;150f" > $serialPort
        for ((i=; i<4; i++)); do
            echo -ne "\e[3"$j"m\e[3"$j"m${DATA[1]:$i:1}" > $serialPort
            sleep 0.02
        done
    done
done
```

Show your ODROID's Stats

This bash script show the ODROID's stats and clock.
To run this script, you need to install the sysstat first.

```
$ sudo apt-get install sysstat
```

This script will show the clock, 4-core load status, CPU frequency and CPU temperature.

In the ODROID-SHOW/example/linux/status.sh

```
#!/bin/bash

flag=
serialPort="/dev/ttyUSB0"

trap "flag=1" SIGINT SIGKILL SIGTERM

./port_open &
subppid=$!

echo -ne "\ec\e[2s\e[3r" > $serialPort
sleep 0.1

function cpu_state {
    cpuFreqM=$(echo "scale=0; " `cat
/sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq` "/1000" | bc)
    cpuTempM=$(echo "scale=1; " `cat /sys/class/thermal/thermal_zone0/temp`
"/1000" | bc)
}

while true
do
    if [ $flag -ne  ] ; then
        echo -ne "\ec\e[2s\e[1r" > $serialPort
        sleep 0.1
        kill $subppid
        exit
    fi
        echo -ne "\e[H\e[35mTime : \e[36m" > $serialPort
        date +"%T" > $serialPort
        sleep 0.1
        echo -ne "\eE\eM\e[32mcore0 : \e[31m" > $serialPort
        sleep 0.1
        mpstat -P  | grep -A1 "usr" | grep -v "usr" | awk '{print ""$4"%
"}' > $serialPort
        sleep 0.1
        echo -ne "\eE\eM\e[32mcore1 : \e[31m" > $serialPort
        sleep 0.1
        mpstat -P 1 | grep -A1 "usr" | grep -v "usr" | awk '{print ""$4"%
"}' > $serialPort
        sleep 0.1
        echo -ne "\eE\eM\e[32mcore2 : \e[31m" > $serialPort
        sleep 0.1
        mpstat -P 2 | grep -A1 "usr" | grep -v "usr" | awk '{print ""$4"%
"}' > $serialPort
```

```
        sleep 0.1
        echo -ne "\eE\eM\e[32mcore3 : \e[31m" > $serialPort
        sleep 0.1
        mpstat -P 3 | grep -A1 "usr" | grep -v "usr" | awk '{print ""$4"%
"}' > $serialPort
        sleep 0.1
        cpu_state
        echo -ne "\eE\eM" > $serialPort
        sleep 0.1
        echo -ne "\e[33mCPU Freq: \e[37m"$cpuFreqM"MHz   \eE" > $serialPort
        echo -ne "\e[33mCPU Temp: \e[37m$cpuTempM\e   " > $serialPort
        sleep 0.1
done
```

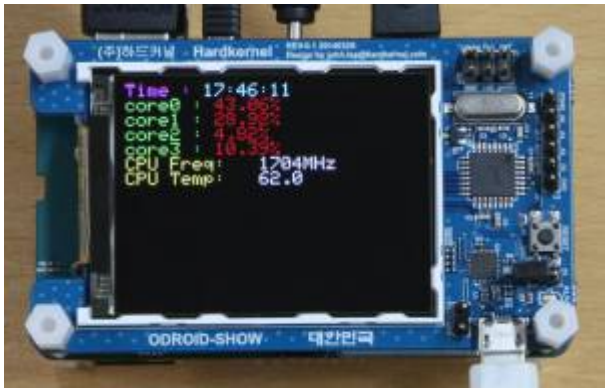


Image display

You can display a graphic image on the ODROID-SHOW.
It supports only raw RGB-565 format.

We used ffmpeg to convert a normal PNG file to raw RGB file. Note that you must resize the PNG file first.

```
$ sudo apt-get install ffmpeg
```

You can convert it with this command. If the conversion is success, you will have the penguin.raw file.

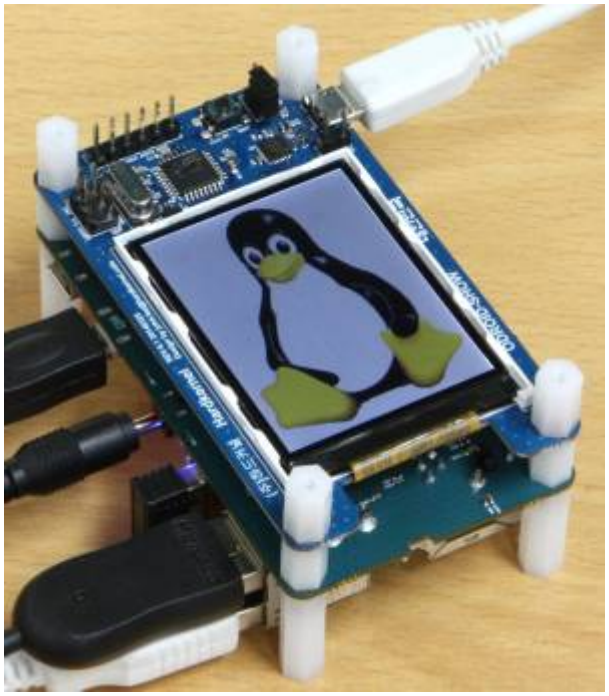
```
$ ffmpeg -vcodec png -i penguin.png -vcodec rawvideo -f rawvideo -pix_fmt
rgb565 penguin.raw
```

You can set the image load mode with the pixel coordination parameters.

In the ODROID-SHOW/example/linux/images.h

```
#!/bin/bash

flag=
serial="/dev/ttyUSB0"

trap "flag=1" SIGINT SIGKILL SIGTERM
```

```
./port_open &
subppid=$!

echo -ne "\ec\e[0r" > $serial
sleep 0.3

while true
do
    if [ $flag -ne  ] ; then
        echo -ne "\ec\e[1r" > $serial
        kill $subppid
        exit
    fi
        echo -ne "\e[0r" > $serial
        sleep 0.2
        echo -ne "\e[0;0,240;320i" > $serial
        cat penguin.raw > $serial
        sleep 0.1
        echo -ne "\e[1r" > $serial
        sleep 0.2
        echo -ne "\e[0;0,320;240i" > $serial
        cat butterfly.raw > $serial
        sleep 0.1
        echo -ne "\e[0r" > $serial
        sleep 0.2
        echo -ne "\e[0;0,240;320i" > $serial
        cat woof.raw > $serial
        sleep 0.1
        echo -ne "\ec\e[0r" > $serial
        sleep 0.3
        echo -ne "\e[40;10,220;200i" > $serial
        cat paint.raw > $serial
        sleep 0.1
        echo -ne "\ec\e[1r" > $serial
        sleep 0.3
        echo -ne "\e[10;10,190;200i" > $serial
        cat paint.raw > $serial
        sleep 0.1
done
```



Linux C example

In the ODROID-SHOW/example/linux/status.c

```
$ gcc -o status status.c
```

Show your x86 Stats

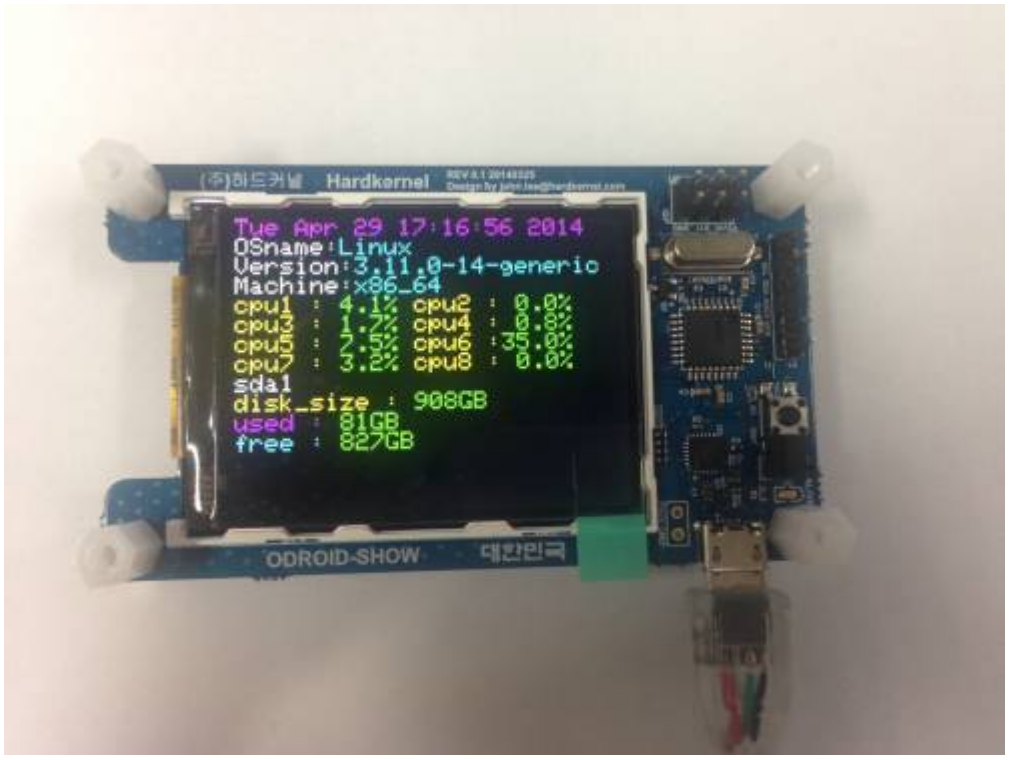
```
#include <stdio.h>
#include <string.h>
#include <assert.h>
#include <time.h>
#include <sys/utsname.h>
#include <sys/statvfs.h>

#include <math.h>

#include <fcntl.h>
#include <termios.h>
#include <errno.h>

#define baudrate  B500000

#define String_startsWith(s, match) (strstr((s), (match)) == (s))

const char procstat[] = "/proc/stat";
const char serialPort[] = "/dev/ttyUSB0";
const char mountPath[] = "/";
```

```
typedef struct CPUData_ {
    unsigned long long int totalTime;
    unsigned long long int userTime;

    unsigned long long int totalPeriod;
    unsigned long long int userPeriod;
} CPUData;

struct diskData_ {
    unsigned long disk_size;
    unsigned long used;
    unsigned long free;
};

int serialSetup(void)
{
    int usbdev;
    struct termios options;

    usbdev = open(serialPort, O_RDWR | O_NOCTTY | O_NDELAY);

    if (usbdev == -1)
        perror("open_port : Unable to open:");

    tcgetattr(usbdev, &options);

    cfsetispeed(&options, baudrate);
    cfsetospeed(&options, baudrate);

    options.c_cflag |= CS8;
    options.c_iflag |= IGNBRK;
    options.c_iflag &= ~( BRKINT | ICRNL | IMAXBEL | IXON);
    options.c_oflag &= ~( OPOST | ONLCR );
    options.c_lflag &= ~( ISIG | ICANON | IEXTEN | ECHO | ECHOE | ECHOK |
                          ECHOCTL | ECHOKE);
    options.c_lflag |= NOFLSH;
    options.c_cflag &= ~CRTSCTS;

    tcsetattr(usbdev, TCSANOW, &options);

    return usbdev;
}

struct diskData_ diskSpace(const char *Path)
{
    struct statvfs vfs;
    struct diskData_ diskData;
    if (!statvfs(Path, &vfs)) {
        diskData.disk_size = vfs.f_blocks * vfs.f_bsize;
```

```
        diskData.free = vfs.f_bfree * vfs.f_bsize;
        diskData.used = diskData.disk_size - diskData.free;
    }
    return diskData;
}

int cpuCount(char *buf, int cpus)
{
    FILE *file = fopen(procstat, "r");
    assert(file != NULL);

    do {
        cpus++;
        fgets(buf, 255, file);
    } while (String_startsWith(buf, "cpu"));

    fclose(file);

    return cpus;
}

void systemInfo(int fd, char *buf)
{
    time_t t;
    struct utsname uts;
    time(&t);
    sprintf(buf, "\e[35m%s\r", ctime(&t));
    write(fd, buf, strlen(buf) + 1);
    usleep(300000);
    uname(&uts);
    sprintf(buf, "\e[37mOSname:\e[36m%s\n\r", uts.sysname);
    write(fd, buf, strlen(buf) + 1);
    sprintf(buf, "\e[37mVersion:\e[36m%s\n\r", uts.release);
    write(fd, buf, strlen(buf) + 1);
    sprintf(buf, "\e[37mMachine:\e[36m%s\n\r", uts.machine);
    write(fd, buf, strlen(buf) + 1);
}

int main(void)
{
    unsigned long long int usertime, nicetime, systemtime, systemalltime,
                idlealltime, idletime, totaltime, virtalltime;
    double total = ;
    char buffer[256];
    int cpus = -1;
    int i;
    time_t t;
    struct utsname uts;
    int usbdev;
    usbdev = serialSetup();
    FILE *file;
```

```
    const unsigned int GB = 1024*1024*1024;
    struct diskData_ diskData;
    diskData = diskSpace(mountPath);
    diskData.disk_size /= GB;
    diskData.used /= GB;
    diskData.free /= GB;
//    printf("Disk usage : %lu \t Free space %lu,%lu\n",
diskData.disk_size/GB, diskData.free/GB, diskData.used/GB);
//    return 0;

    cpus = cpuCount(buffer, cpus);
    CPUData cpuData[cpus];

    for (i = ; i < cpus; i++) {
        cpuData[i].totalTime = 1;
        cpuData[i].userTime = ;
        cpuData[i].totalPeriod = 1;
        cpuData[i].userPeriod = ;
    }

    sprintf(buffer, "\ec\e[2s\e[1r");
    write(usbdev, buffer, strlen(buffer) + 1);
    fsync(usbdev);

    int cpuid;
    unsigned long long int ioWait, irq, softIrq, steal, guest;
    ioWait = irq = softIrq = steal = guest = ;

    while (1) {
        sprintf(buffer, "\e[H");
        write(usbdev, buffer, strlen(buffer) + 1);
        fsync(usbdev);
        usleep(300000);
        systemInfo(usbdev, buffer);
        fsync(usbdev);
        usleep(300000);
        file = fopen(procstat, "r");
        for (i = ; i < cpus; i++) {
            fgets(buffer, 255, file);
            if (i == ) {
                sscanf(buffer, "cpu %llu %llu %llu %llu %llu %llu %llu %llu
%llu",
                    &usertime, &nicetime, &systemtime, &idletime,
                    &ioWait, &irq, &softIrq, &steal, &guest);
            } else {
                sscanf(buffer, "cpu%d %llu %llu %llu %llu %llu %llu %llu
%llu %llu",
                    &cpuid, &usertime, &nicetime, &systemtime, &idletime,
                    &ioWait, &irq, &softIrq, &steal, &guest);
```

```
                assert(cpuid == i - 1);
            }

            idlealltime = idletime + ioWait;
            systemalltime = systemtime + irq + softIrq;
            virtalltime = steal + guest;
            totaltime = usertime + nicetime + systemalltime +
                        idlealltime + virtalltime;

            assert(usertime >= cpuData[i].userTime);
            assert(totaltime >= cpuData[i].totalTime);

            cpuData[i].userPeriod = usertime - cpuData[i].userTime;
            cpuData[i].totalPeriod = totaltime - cpuData[i].totalTime;

            cpuData[i].totalTime = totaltime;
            cpuData[i].userTime = usertime;

            total = (double)cpuData[i].totalPeriod;
            if ((i != ) && (i%2 == 1)) {
                sprintf(buffer, "\e[33mcpu%d :\e[32m%4.1f%% ",
                    i, cpuData[i].userPeriod/total*100.0);
                write(usbdev, buffer, strlen(buffer) + 1);
                fsync(usbdev);
            } else if ((i != ) && (i%2 == )) {
                sprintf(buffer, "\e[33mcpu%d :\e[32m%4.1f%%  \n\r",
                    i, cpuData[i].userPeriod/total*100.0);
                write(usbdev, buffer, strlen(buffer) + 1);
                fsync(usbdev);
            }
        }
        sprintf(buffer, "\e[37msda1\n\r\e[33mdisk_size : \e[32m%ldGB\n\r",
                            diskData.disk_size);
        write(usbdev, buffer, strlen(buffer) + 1);
        usleep(100000);
        sprintf(buffer, "\e[35mused : \e[32m%ldGB\n\r",
                            diskData.used);
        write(usbdev, buffer, strlen(buffer) + 1);
        usleep(100000);
        sprintf(buffer, "\e[36mfree : \e[32m%ldGB",
                            diskData.free);
        write(usbdev, buffer, strlen(buffer) + 1);
        usleep(100000);
        fclose(file);
    }

    return ;
}
```



From:
http://wiki.odroid.com/ - **ODROID Wiki**

Permanent link:
http://wiki.odroid.com/old_product/accessory/display/odroid-show/usage_examples

Last update: **2017/07/27 09:25**