

# Device driver (II)

## Purpose

Learn how to control GPIO(General Purpose Input Output)

## Steps

1. Write a C code to make the LED turn on and off overtime:

```
#include <stdio.h>
#include <unistd.h>

int main()
{
    FILE* p;
    p = fopen("/sys/class/gpio/export", "w");
    fprintf(p, "%d", 4);
    fclose(p);
    p = fopen("/sys/class/gpio/gpio4/direction", "w");
    fprintf(p, "out");
    fclose(p);
    int i = 1;
    while(++i)
    {
        if(i & 1)
        {
            p = fopen("/sys/class/gpio/gpio4/value", "w");
            fprintf(p, "%d", 1);
            fclose(p);
        }
        else
        {
            p = fopen("/sys/class/gpio/gpio4/value", "w");
            fprintf(p, "%d", 0);
            fclose(p);
        }
        sleep(1);
    }
    p = fopen("/sys/class/gpio/unexport", "w");
    fprintf(p, "%d", 4);
    fclose(p);
    return 0;
}
```

2. `export PATH=$PATH:$HOME/WORK/crossgcc2/bin`
3. `arm-linux-gnueabi-gcc -static -g driver.c -o driver`
4. Put the binary file on Raspberry Pi.

## Discussion

This experiment is fun and easy!