K210入门

如何配置、实验1.1、1.2

周钜宸

如何配置

参考：2 - PaddlePi-K210 开发环境搭建指南.pdf

一些需要用到的网址：

cmake下载：https://cmake.org/download/

Kendryte工具链：https://github.com/kendryte/kendryte-gnu-toolchain/releases

SDK：https://github.com/kendryte/kendryte-standalone-sdk

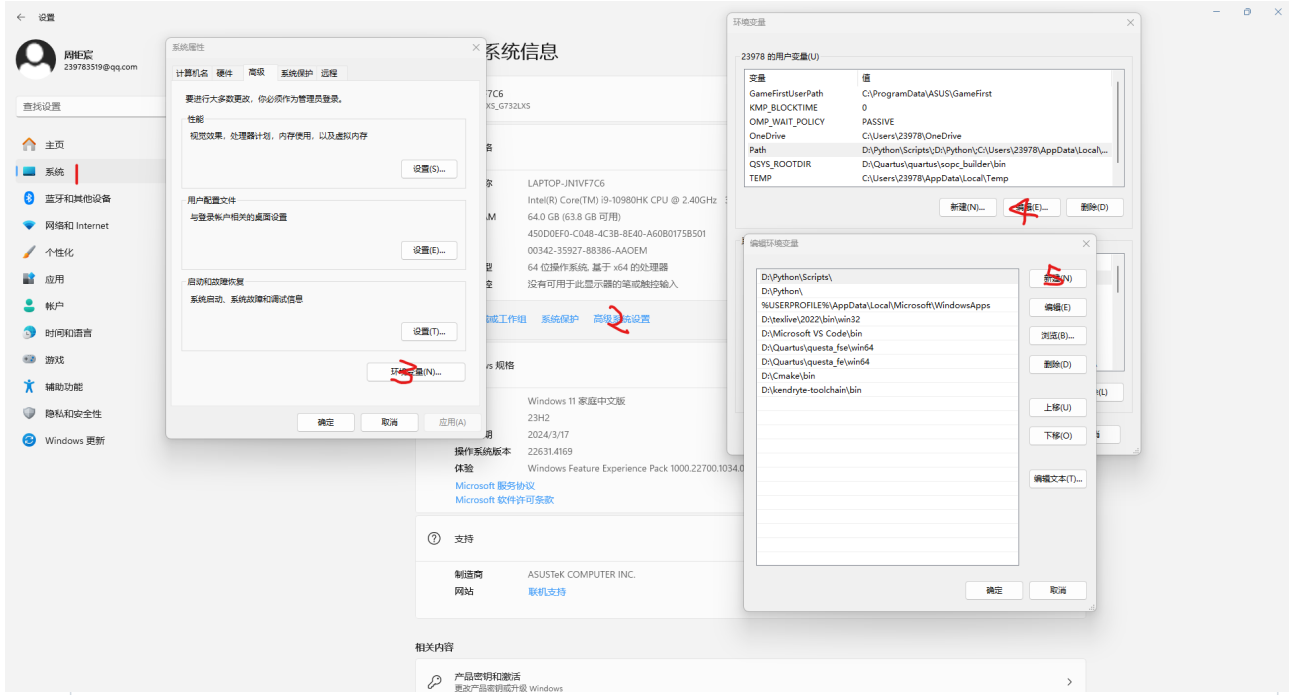
K-flash工具：https://github.com/kendryte/kendryte-flash-windows/releases

demos：https://github.com/kendryte/PaddlePi/tree/master

VScode：https://code.visualstudio.com/

如何配置

下载并安装VScode、cmake；
下载并解压Kendryte工具链；
记住cmake和Kendryte的路径，
将其中的bin加入Path环境变量。



▾ Assets 11

kendryte-toolchain-osx-darwin-m1-8.2.0-20210717.tar.xz	28.8 MB	Jul 19, 2021
kendryte-toolchain-osx-mojave-8.2.0-20190409.tar.bz2	46.3 MB	Apr 11, 2019
kendryte-toolchain-osx-mojave-8.2.0-20190409.tar.xz	21.4 MB	Apr 10, 2019
kendryte-toolchain-ubuntu-amd64-8.2.0-20190409.tar.bz2	45.6 MB	Apr 12, 2019
kendryte-toolchain-ubuntu-amd64-8.2.0-20190409.tar.xz	19.2 MB	Apr 12, 2019
kendryte-toolchain-win-amd64-8.2.0-20190409.tar.xz	18.2 MB	Apr 12, 2019
kendryte-toolchain-win-amd64-8.2.0-20190409.zip	51.4 MB	Apr 12, 2019
kendryte-toolchain-win-i386-8.2.0-20190409.tar.xz	16.8 MB	Apr 11, 2019
kendryte-toolchain-win-i386-8.2.0-20190409.zip	48.6 MB	Apr 11, 2019
Source code (zip)		Apr 9, 2019
Source code (tar.gz)		Apr 9, 2019

1 person reacted

如何配置

下载并解压SDK、demos；

将/standalone-demos/里的项目
develop/src中；

打开VScode，打开文件夹SDK；

在SDK中新建文件夹build；

打开CMakeLists.txt文件，添加语句并保存：set(PROJ xxx)，xxx为需要的文件夹；

选择RISCV的编译器，然后点击生成。应该在build文件夹中看到xxx.bin的文件。

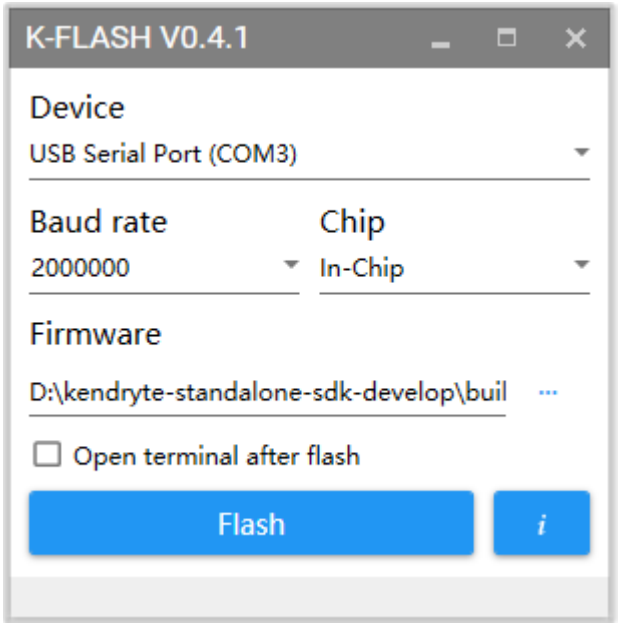
CMakeLists.txt

```
# DO NOT MODIFY THIS FILE, IT WILL BE OVERRIDE!!!

# set this will supress some warnings
set(BUILDING_SDK "yes" CACHE INTERNAL "")

# basic config
set(PROJ gpiohs_led)
if (NOT PROJ)
    get_filename_component(PROJ ${CMAKE_CURRENT_BINARY_DIR} DIRECTORY)
    get_filename_component(PROJ ${PROJ} NAME)
    string(REPLACE " " "_" PROJ ${PROJ})
    message(STATUS "PROJ not set, use ${PROJ} as PROJ. Also, you can set it manually. e.g. -DPROJ=hello_world")
else()
    message("PROJ = ${PROJ}")
endif ()
cmake_minimum_required(VERSION 3.0)
include(./cmake/common.cmake)
project(${PROJ} C CXX ASM)

# config self use headers
include(./cmake/macros.internal.cmake)
header_directories(${SDK_ROOT}/lib)
header_directories(src/${PROJ})
header_directories(kendryte-standalone-demo/${PROJ})
```

如何配置

下载并解压K-flash工具；

连接K210开发板，打开K-flash，选择适配的端口与生成的bin文件，flash至开发板上。



1.1 GPIO

打开CMakeLists.txt文件，修改语句并保存：set(PROJ gpio);
生成.bin文件;
使用K-flash软体将build中的gpio.bin烧录至开发板上;
应观察到LED灯以1秒的频率闪烁。

```c
int main(void)
{
    fpioa_set_function(10, FUNC_GPIO3);

    gpio_init();
    gpio_set_drive_mode(3, GPIO_DM_OUTPUT);
    gpio_pin_value_t value = GPIO_PV_HIGH;
    gpio_set_pin(3, value);
    while (1)
    {
        sleep(1);
        gpio_set_pin(3, value = !value);
    }
    return 0;
}

```

初始化GPIO

以1秒的频率
给GPIO赋值

1.2 GPIOHS_LED

可以观察到：按下按钮LED亮，松开按钮LED灭。

```c
int main(void)
{
    plic_init();
    sysctl_enable_irq();

    fpioa_set_function(PIN_LED, FUNC_GPIOHS3);
    gpiohs_set_drive_mode(GPIO_LED, GPIO_DM_OUTPUT);
    gpio_pin_value_t value = GPIO_PV_HIGH;
    gpiohs_set_pin(GPIO_LED, value);

    fpioa_set_function(PIN_KEY, FUNC_GPIOHS2);
    gpiohs_set_drive_mode(GPIO_KEY, GPIO_DM_INPUT_PULL_UP);
    gpiohs_set_pin_edge(GPIO_KEY, GPIO_PE_BOTH);

    gpiohs_irq_register(GPIO_KEY, 1, irq_gpiohs2, &g_count);

    while (1);
}
```

初始化GPIO

```c
uint32_t g_count;

int irq_gpiohs2(void* ctx)
{
    irq_flag = gpiohs_get_pin(GPIO_KEY);

    printf("IRQ The PIN is %d\n", irq_flag);

    uint32_t *tmp = (uint32_t *)(ctx);
    printf("count is %d\n", (*tmp)++);

    if (!irq_flag)
        gpiohs_set_pin(GPIO_LED, GPIO_PV_LOW);
    else
        gpiohs_set_pin(GPIO_LED, GPIO_PV_HIGH);
    return 0;
}
```

读取GPIO

赋值GPIO

敬请批评指正!