计算机劳动实践第二天日报

学号：222200408 姓名：连文桢

目录

[计算机劳动实践第二天日报 1](#_Toc155380320)

[一、 学习知识 2](#_Toc155380321)

[1. Sample 内三个文件 2](#_Toc155380322)

[1) 可执行文件 2](#_Toc155380323)

[2) C语言文件 2](#_Toc155380324)

[3) 文档 2](#_Toc155380325)

[2. 创建自己的第一个文件 2](#_Toc155380326)

[1) 创建文件夹 2](#_Toc155380327)

[2) 创建文件下的.c和bn文件 2](#_Toc155380328)

[3) 复制头文件到.c文件中 2](#_Toc155380329)

[4) 修改build 文件 2](#_Toc155380330)

[5) 修改函数 3](#_Toc155380331)

[6) 修改整的build文件 3](#_Toc155380332)

[7) 重新编译 4](#_Toc155380333)

[8) 导出bin文件 4](#_Toc155380334)

[9) 用hiburn重新烧写 5](#_Toc155380335)

[10) 检查烧写结果 6](#_Toc155380336)

[3. 学习烧录一个可手动控制亮度的程序 7](#_Toc155380337)

[1) 学习文档 7](#_Toc155380338)

[2) 复制，改写代码 7](#_Toc155380339)

[3) 烧入程序 9](#_Toc155380340)

[4) 硬件检测 10](#_Toc155380341)

[二、 学习收获 14](#_Toc155380342)

[1. 学习了vscode的sample文件 14](#_Toc155380343)

[2. 创建了自己的helloworld文件 15](#_Toc155380344)

[3. 实现了对小熊派硬件设备的控制 15](#_Toc155380345)

[4. 了解了示例文档的重要性 15](#_Toc155380346)

[5. 培养了条理性思维 15](#_Toc155380347)

[三、 待答疑的问题 16](#_Toc155380348)

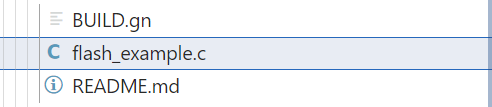
[1. 如何将多个程序融合成需要的程序？ 16](#_Toc155380349)

[2. 软件如何控制设备的根本原理是什么？ 16](#_Toc155380350)

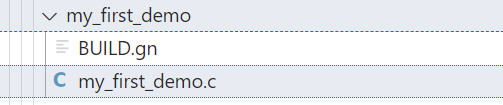
[3. 在build文件中，代码含义如何解释？例如头文件路径是什么意思？ 16](#_Toc155380351)

[4. 如何解决编译错误？有时会出现一些未知原因的错误。 16](#_Toc155380352)

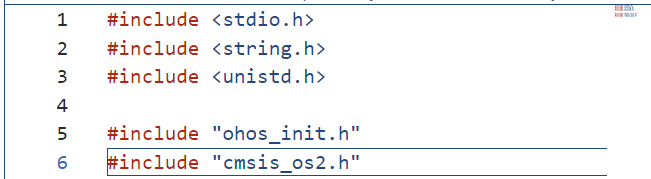
1. 学习知识
2. Sample 内三个文件
3. 可执行文件
4. C语言文件
5. 文档



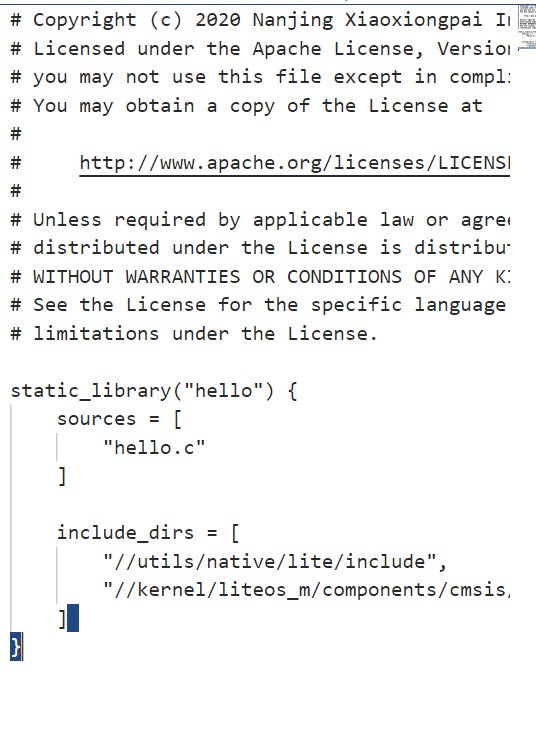
1. 创建自己的第一个文件
2. 创建文件夹



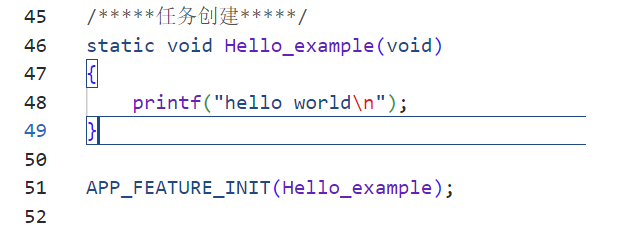
1. 创建文件下的.c和bn文件
2. 复制头文件到.c文件中



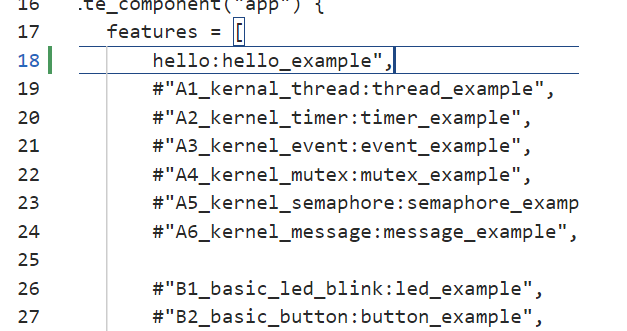
1. 修改build 文件



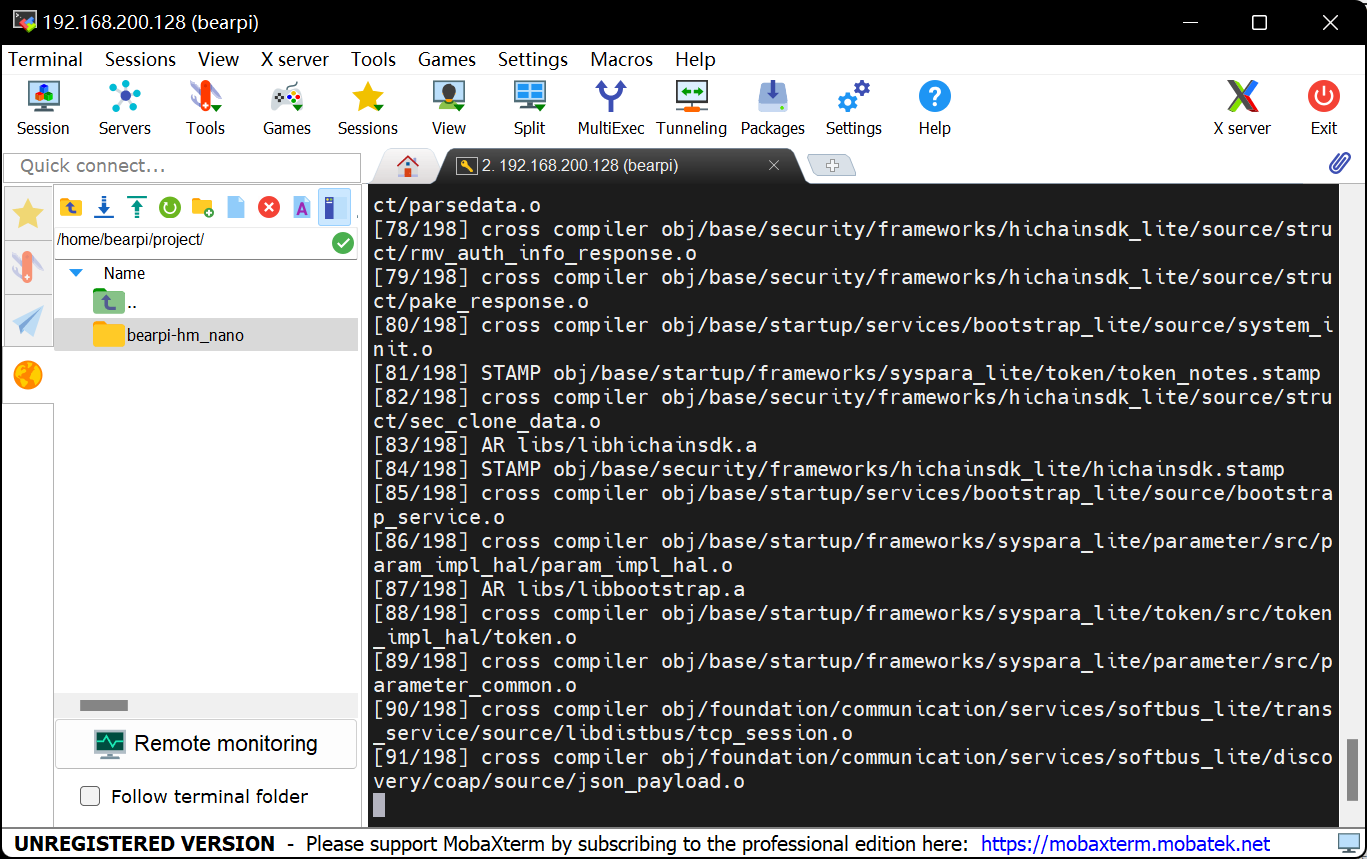
1. 修改函数



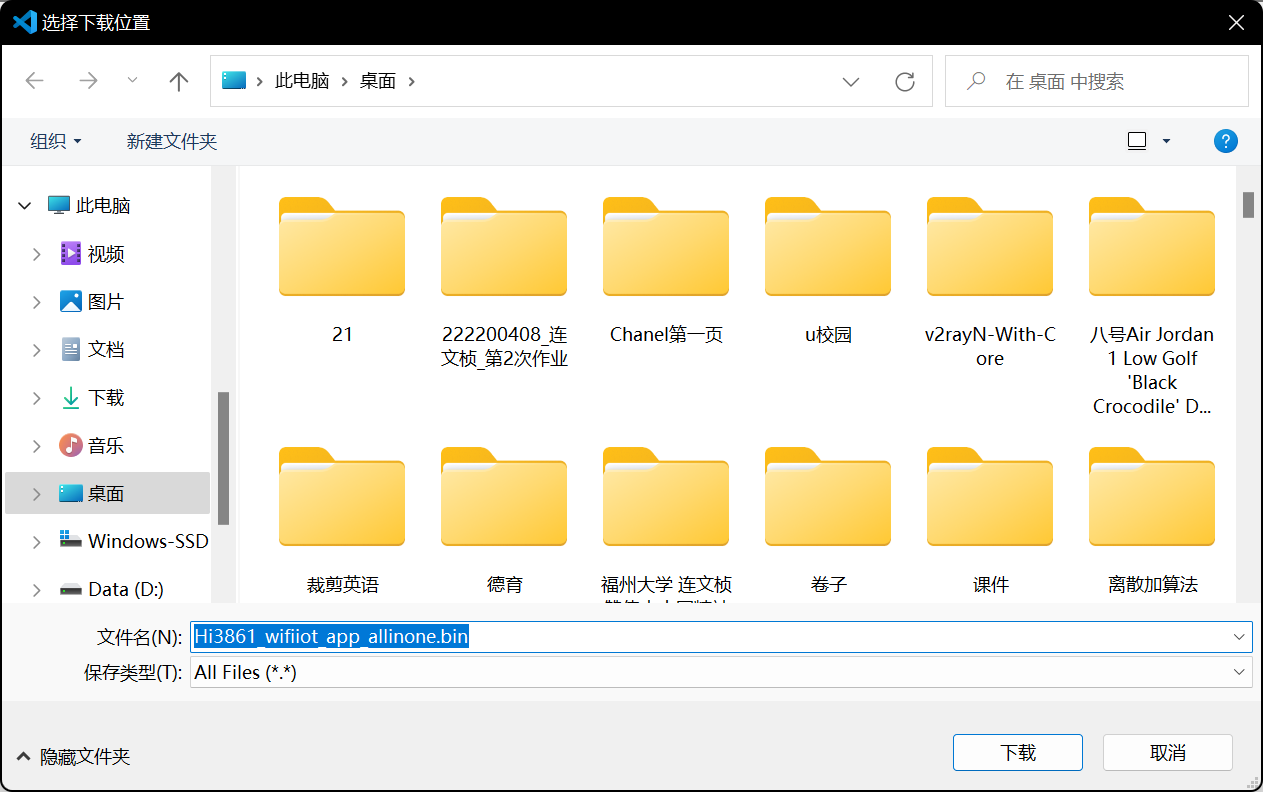
1. 修改整的build文件



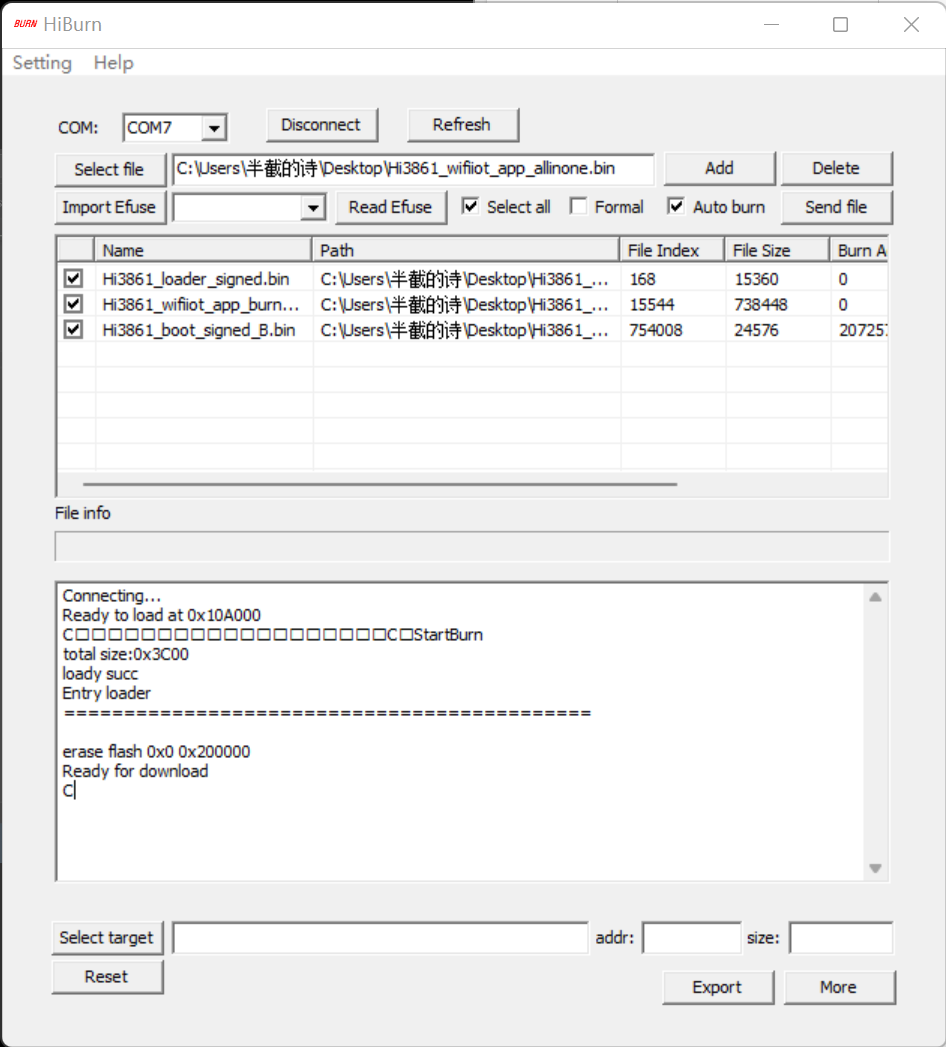
1. 重新编译



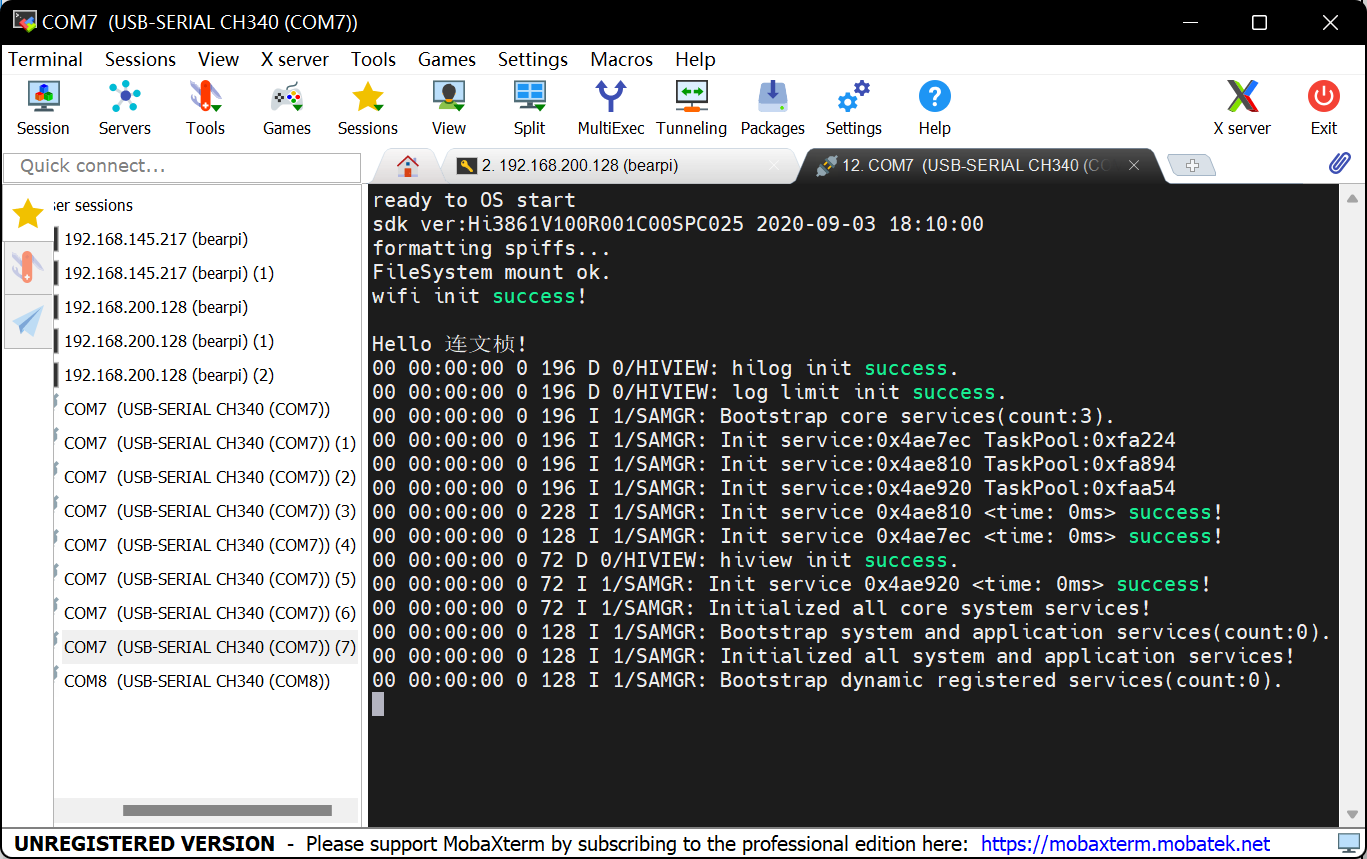
1. 导出bin文件



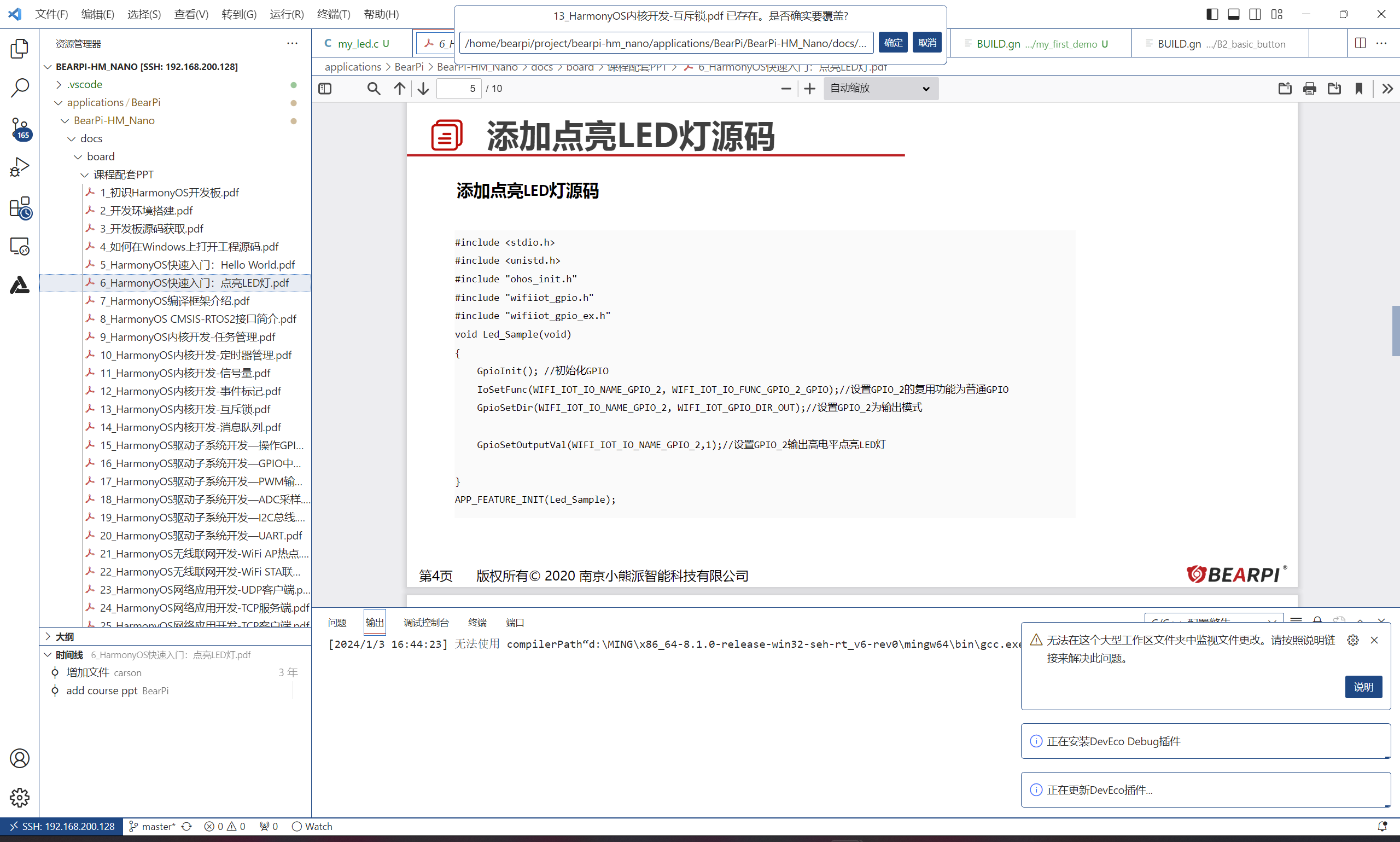
1. 用hiburn重新烧写



1. 检查烧写结果



1. 学习烧录一个可手动控制亮度的程序
2. 学习文档



1. 复制，改写代码

#include <stdio.h>

#include <unistd.h>

#include "ohos\_init.h"

#include "cmsis\_os2.h"

#include "wifiiot\_pwm.h"

#include "wifiiot\_gpio.h"

#include "wifiiot\_gpio\_ex.h"

#define PWM\_TASK\_STACK\_SIZE 512

#define PWM\_TASK\_PRIO 25

static  int i=20000;

static void  F1\_Pressed(char \*arg)

{

    (void)arg;

    i+=1000;

    printf("%d\n",i);

}

static void  F2\_Pressed(char \*arg)

{

    (void)arg;

    if(i-1000>=0)

    i-=1000;

    printf("%d\n",i);

}

static void ButtonExampleEntry(void)

{

    GpioInit();

    //初始化LED灯

    IoSetFunc(WIFI\_IOT\_IO\_NAME\_GPIO\_2, WIFI\_IOT\_IO\_FUNC\_GPIO\_2\_GPIO);

    GpioSetDir(WIFI\_IOT\_IO\_NAME\_GPIO\_2, WIFI\_IOT\_GPIO\_DIR\_OUT);

    //初始化F1按键，设置为下降沿触发中断

    IoSetFunc(WIFI\_IOT\_IO\_NAME\_GPIO\_11, WIFI\_IOT\_IO\_FUNC\_GPIO\_11\_GPIO);

    GpioSetDir(WIFI\_IOT\_IO\_NAME\_GPIO\_11, WIFI\_IOT\_GPIO\_DIR\_IN);

    IoSetPull(WIFI\_IOT\_IO\_NAME\_GPIO\_11, WIFI\_IOT\_IO\_PULL\_UP);

    GpioRegisterIsrFunc(WIFI\_IOT\_IO\_NAME\_GPIO\_11, WIFI\_IOT\_INT\_TYPE\_EDGE, WIFI\_IOT\_GPIO\_EDGE\_FALL\_LEVEL\_LOW, F1\_Pressed, NULL);

    //初始化F2按键，设置为下降沿触发中断

    IoSetFunc(WIFI\_IOT\_IO\_NAME\_GPIO\_12, WIFI\_IOT\_IO\_FUNC\_GPIO\_12\_GPIO);

    GpioSetDir(WIFI\_IOT\_IO\_NAME\_GPIO\_12, WIFI\_IOT\_GPIO\_DIR\_IN);

    IoSetPull(WIFI\_IOT\_IO\_NAME\_GPIO\_12, WIFI\_IOT\_IO\_PULL\_UP);

    GpioRegisterIsrFunc(WIFI\_IOT\_IO\_NAME\_GPIO\_12, WIFI\_IOT\_INT\_TYPE\_EDGE, WIFI\_IOT\_GPIO\_EDGE\_FALL\_LEVEL\_LOW, F2\_Pressed, NULL);

    //初始化GPIO

    GpioInit();

    //设置GPIO\_2引脚复用功能为PWM

    IoSetFunc(WIFI\_IOT\_IO\_NAME\_GPIO\_2, WIFI\_IOT\_IO\_FUNC\_GPIO\_2\_PWM2\_OUT);

    //设置GPIO\_2引脚为输出模式

    GpioSetDir(WIFI\_IOT\_IO\_NAME\_GPIO\_2, WIFI\_IOT\_GPIO\_DIR\_OUT);

    //初始化PWM2端口

    PwmInit(WIFI\_IOT\_PWM\_PORT\_PWM2);

    while(1)

    {

           PwmStart(WIFI\_IOT\_PWM\_PORT\_PWM2, i, 40000);

    }

}

static void PWMExampleEntry(void)

{

    osThreadAttr\_t attr;

    attr.name = "ButtonExampleEntry";

    attr.attr\_bits = 0U;

    attr.cb\_mem = NULL;

    attr.cb\_size = 0U;

    attr.stack\_mem = NULL;

    attr.stack\_size = 512;

    attr.priority = 25;

    if (osThreadNew((osThreadFunc\_t)ButtonExampleEntry, NULL, &attr) == NULL)

    {

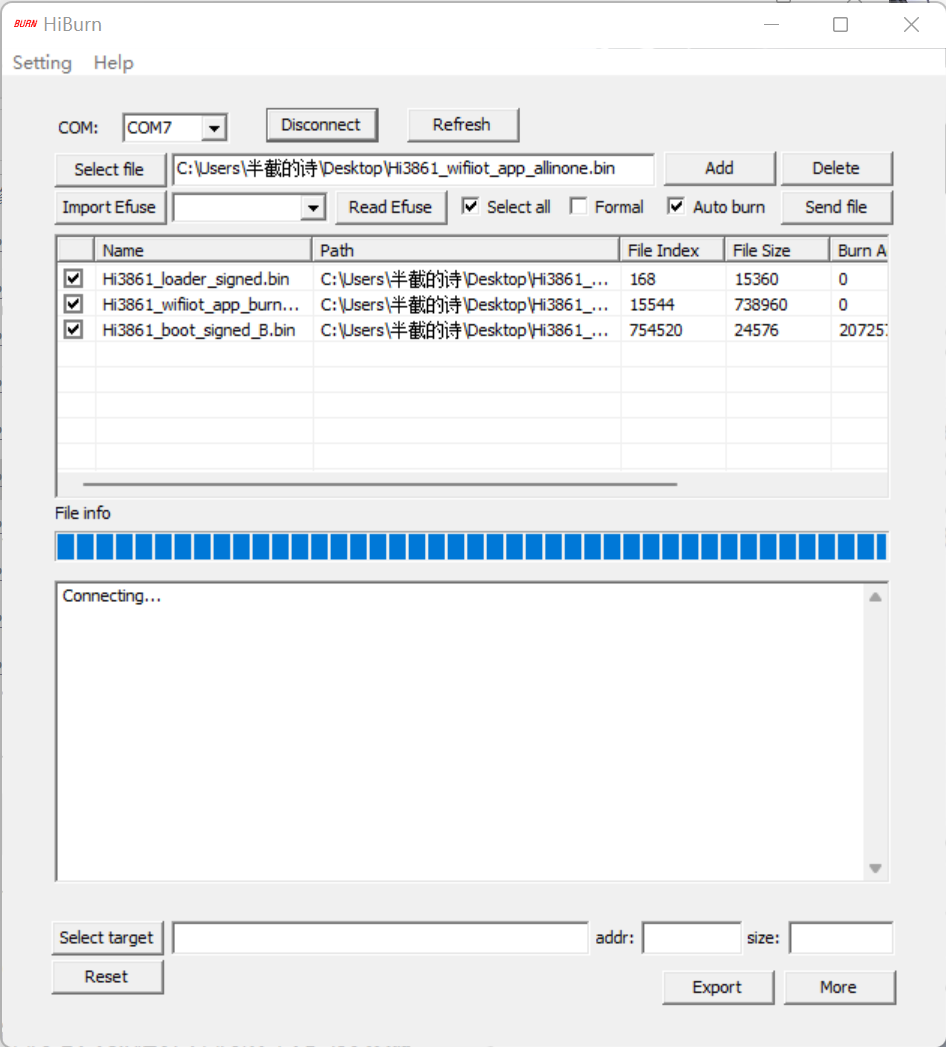
        printf("Falied to create ButtonExampleEntry!\n");

    }

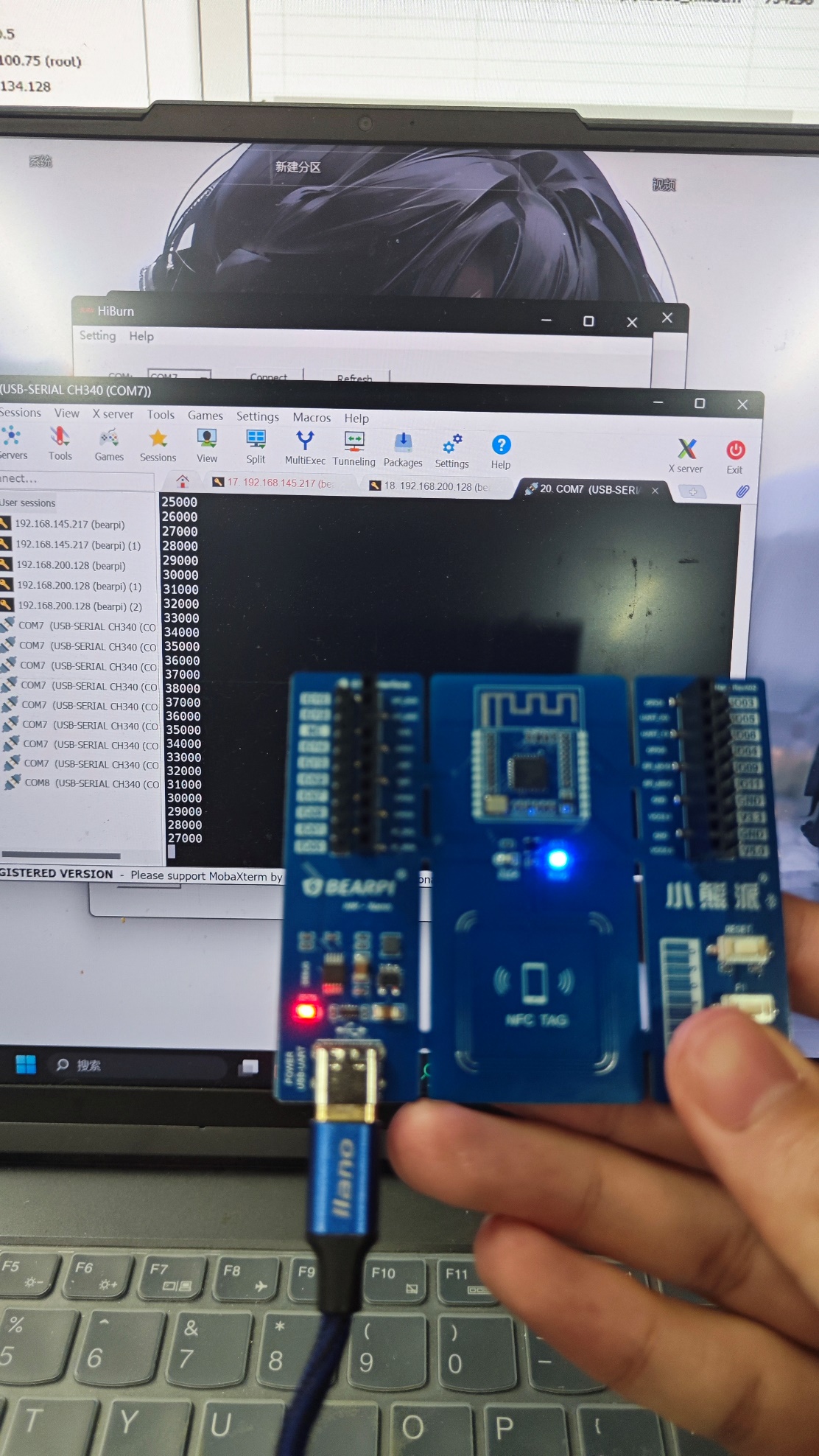
}

APP\_FEATURE\_INIT(PWMExampleEntry);

1. 烧入程序

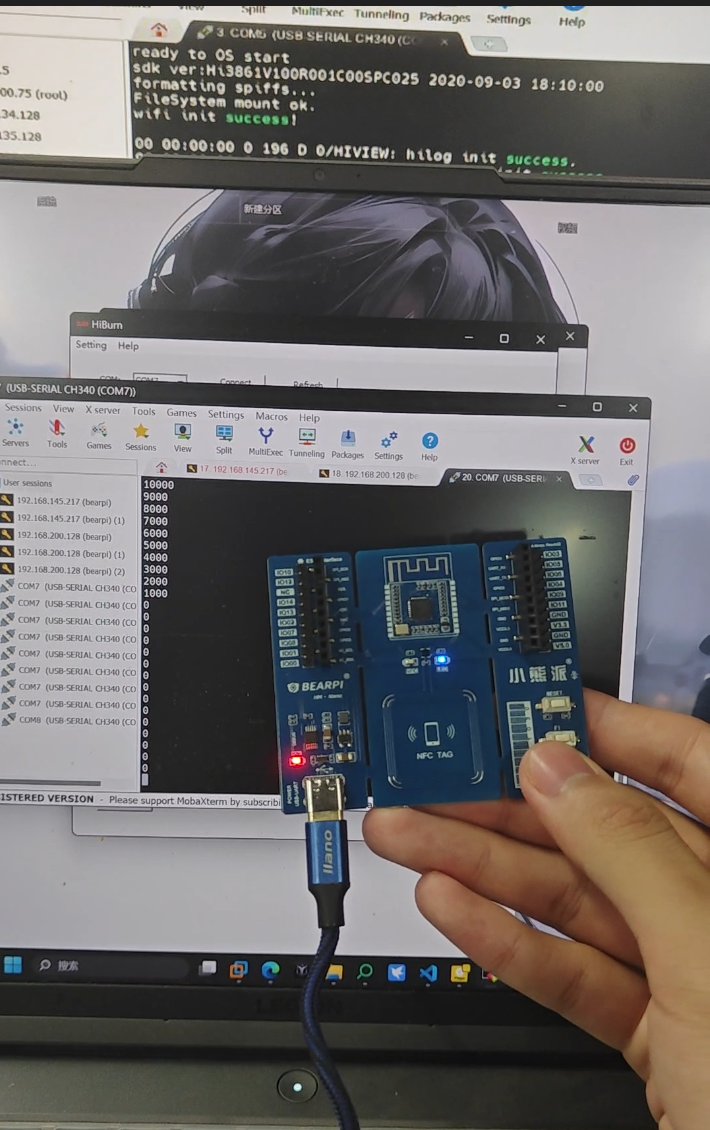


1. 硬件检测









1. 学习收获
2. 学习了vscode的sample文件

通过学习vscode的sample文件，我了解了如何配置开发环境和使用已有的代码模板。这些代码模板包含了一些常见的功能和库，方便我快速开发应用。

1. 创建了自己的helloworld文件

在使用sample文件的基础上，我创建了自己的helloworld文件，并成功编写出了第一个程序。这是硬件编程的第一步，也标志着我开始了自己的硬件编程之旅。

1. 实现了对小熊派硬件设备的控制

通过学习示例文档和使用模板，我掌握了如何自由控制小熊派硬件设备。这使得我可以使用代码实现各种功能，如LED灯、温度传感器等。

1. 了解了示例文档的重要性

学习示例文档可以帮助我快速上手，理解代码的工作原理和使用方法。示例文档中包含了常见的问题解答和最佳实践，让我更加深入地理解相关技术。

1. 培养了条理性思维

在学习过程中，我需要按照一定的顺序学习和实践，这培养了我的条理性思维。这将有助于我更加有效地学习和掌握技能。

总之，通过以上的学习，我已经掌握了硬件编程的基本知识和技能，并且学会了如何使用示例文档和代码模板来快速开发应用。这是一个非常好的开始，希望我可以继续努力学习，不断提升自己的技能水平。

1. 待答疑的问题
2. 如何将多个程序融合成需要的程序？
3. 软件如何控制设备的根本原理是什么？
4. 在build文件中，代码含义如何解释？例如头文件路径是什么意思？
5. 如何解决编译错误？有时会出现一些未知原因的错误。