



# OpenHarmony快速入门: Hello World

## 本节主要介绍:

- 如何编写Hello World程序
- 如何编译烧录Hello World程序
- 如何调试Hello World程序

# 三 目录

---

1. 添加Hello World源码文件
2. 编写Hello World业务代码
3. 编写编译构建文件BUILD.gn
4. 调试Hello World程序
5. 总结



# 添加Hello World源码文件

## 1、新增my\_app文件夹

在./applications/BearPi/BearPi-HM\_Nano/sample路径下新建一个my\_app目录，用于存放业务源码文件。

## 2、新增hello\_world.c文件

在./applications/BearPi/BearPi-HM\_Nano/sample/my\_app路径下新建一个hello\_world.c文件，该文件为业务源码文件。

## 3、新增BUILD.gn文件

在./applications/BearPi/BearPi-HM\_Nano/sample/my\_app路径下新建一个BUILD.gn文件，该文件为业务源码编译脚本。



# 编写Hello World业务代码

## 添加HelloWorld代码

在hello\_world.c中新建业务入口函数HelloWorld，并实现业务逻辑。并在代码最下方，使用HarmonyOS启动恢复模块接口APP\_FEATURE\_INIT()启动业务。（APP\_FEATURE\_INIT定义在ohos\_init.h文件中）

```
#include <stdio.h>
#include "ohos_init.h"

void Hello_World(void)
{
    printf("Hello World!\r\n");
}
APP_FEATURE_INIT(Hello_World);
```

# 编写业务编译构建文件BUILD.gn

## 编写用于将业务构建成静态库的BUILD.gn文件

在./applications/BearPi/BearPi-HM/sample/my\_app下的BUILD.gn文件中添加如下代码。

```
static_library("myapp") {  
    sources = [  
        "hello_world.c"  
    ]  
    include_dirs = [  
        "//utils/native/lite/include"  
    ]  
}
```

- static\_library中指定业务模块的编译结果，为静态库文件libmyapp.a，开发者根据实际情况完成填写。
- sources中指定静态库.a所依赖的.c文件及其路径，若路径中包含"//"则表示绝对路径（此处为代码根路径），若不包含"//"则表示相对路径。
- include\_dirs中指定source所需要依赖的.h文件路径。

# 编写模块编译构建文件BUILD.gn

**编写模块BUILD.gn文件，指定需参与构建的特性模块。**

在./applications/BearPi/BearPi-HM/sample下的BUILD.gn文件中添加如下代码。

```
import("//build/lite/config/component/lite_component.gni")
```

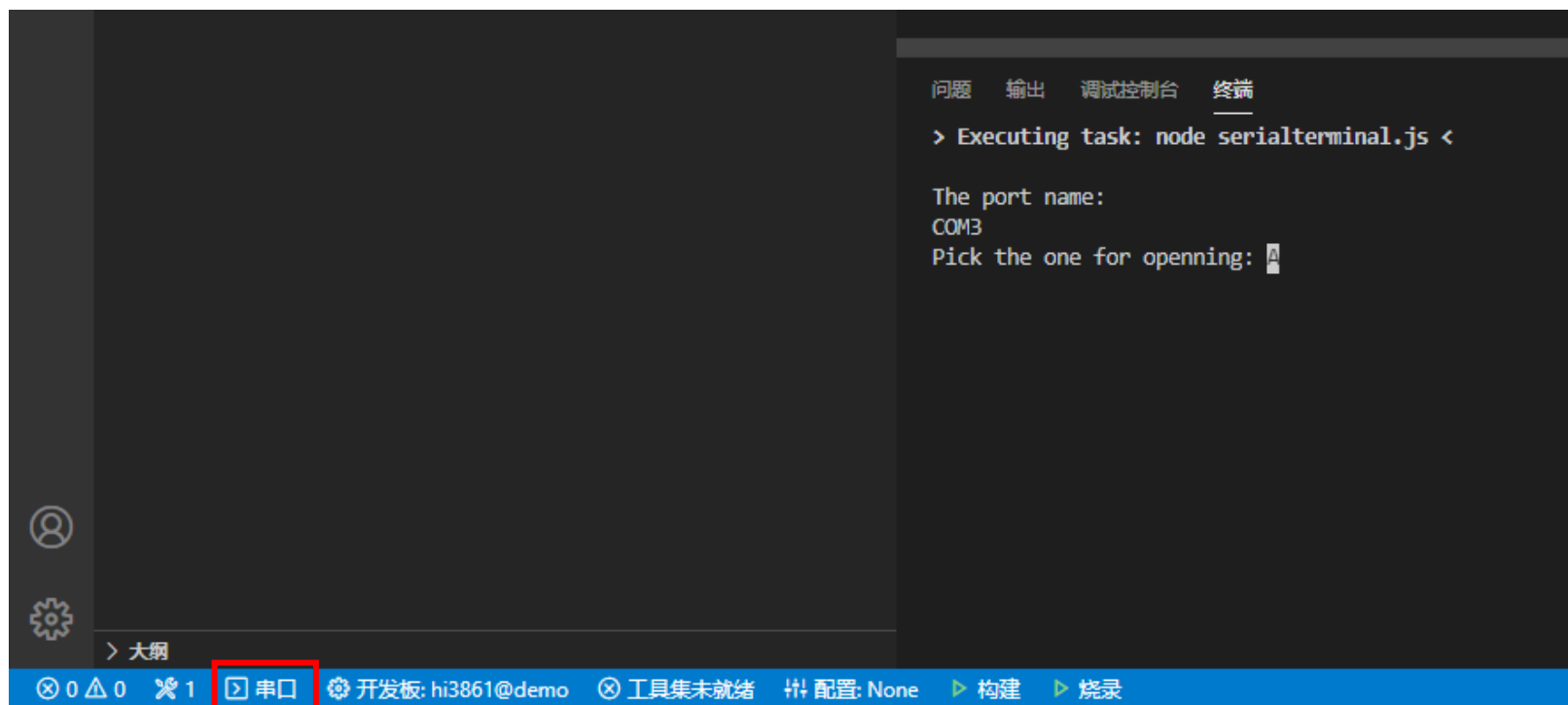
```
lite_component("app") {  
    features = [  
        "my_app:myapp",  
    ]  
}
```

- my\_app是相对路径，指向./applications/BearPi/BearPi-HM/sample/my\_app/BUILD.gn。
- myapp是目标，指向./applications/BearPi/BearPi-HM/sample/my\_app/BUILD.gn中的static\_library("myapp")。



# 调试Hello World程序

1. 在Visual Studio Code底部，点击“串口”按钮图标。
2. 输入端口号，其他参数保持默认配置。
3. 按下开发板RESET按键，即可输出日志信息。





## 本节小结

---

- 1、掌握如何创建业务源码文件
- 2、掌握如何编写业务源码
- 3、掌握如何编译业务源码
- 4、掌握串口工具的使用



谢谢观看

开源从小熊派开始

OPEN-SOURCE STARTED WITH THE BEARPI