# 1. 准备工作

硬件准备

开发板

首先需要准备一个小熊派IoT开发板,并通过USB线与电脑连接。

mark

软件准备

需要安装好Keil - MDK及芯片对应的包，以便编译和下载生成的代码,可参考MDK安装教程

# 2.生成MDK工程

选择芯片型号

打开STM32CubeMX，打开MCU选择器：

mark

搜索并选中芯片STM32L431RCT6:

mark

配置时钟源

如果选择使用外部高速时钟（HSE），则需要在System Core中配置RCC；

如果使用默认内部时钟（HSI），这一步可以略过；

这里我都使用外部时钟：

mark

配置LED的GPIO引脚

查看小熊派开发板的原理图，如下：

mark

所以接下来我们选择配置PC13引脚：

mark

设置用户标签为LED：

mark

配置按键的GPIO引脚

查看原理图，确定控制按键引脚，这里我们的F1按键接的是PB2，F2按键接的是PB3。

image.png

所以接下来我们选择配置PB2引脚和PB3引脚：

mark

因为没有设置硬件上拉，所以我们配置开启上拉电阻，并设置用户标签为KEY1和KEY2：

mark

配置时钟树

STM32L4的最高主频到80M，所以配置PLL，最后使HCLK = 80Mhz即可：

mark

生成工程设置

mark

代码生成设置

最后设置生成独立的初始化文件：

mark

生成代码

点击GENERATE CODE即可生成MDK-V5工程：

mark

# 3. 在MDK中编写、编译、下载用户代码

编写用户代码

打开MDK工程，在main.c中的main函数中编写简单的用户代码，这里的代码意思为：一直循环读取按键引脚的电平，当读取到F1按键引脚为低电平时，LED灯引脚输出高电平；当读取到F2按键引脚为低电平时，LED灯引脚输出低电平，所以当按下KEY1，LED点亮，按下KEY2，LED熄灭。

int main(void)

{

HAL\_Init();

SystemClock\_Config();

MX\_GPIO\_Init();

while (1)

{

/\* USER CODE BEGIN 3 \*/

if(0 == HAL\_GPIO\_ReadPin(KEY1\_GPIO\_Port, KEY1\_Pin))

{

HAL\_GPIO\_WritePin(LED\_GPIO\_Port,LED\_Pin,GPIO\_PIN\_SET);

}

if(0 == HAL\_GPIO\_ReadPin(KEY2\_GPIO\_Port, KEY2\_Pin))

{

HAL\_GPIO\_WritePin(LED\_GPIO\_Port,LED\_Pin,GPIO\_PIN\_RESET);

}

}

/\* USER CODE END 3 \*/

}

编译代码

点击如图所示的按钮编译工程

image.png

编译成功

mark

设置下载器

点击如图所示按钮打开设置页面

image.png

进行下载设置，选择“ST-Link Debugger”，并点击“Settings”。

mark

在“Flash Download”菜单下勾选“Reset and Run”选项，已达到烧录程序后单片机自动复位并运行程序的目的。

mark

下载运行

点击"LOAD"按钮即可烧录代码到单片机中。

image.png

烧录成功

mark

实验现象

下载运行后，实验现象如下：

上电复位时LED处于熄灭状态；

按下KEY1，LED点亮；

按下KEY2，LED熄灭；

mark

至此，我们已经学会了如何使用STM32CubeMX快速生成MDK的工程，以及如何使用 STM32CubeMX初始化GPIO进行按键检测，下一节讲述如何配置NVIC使用外部中断检测按键。