NDA102支持PUSH调光功能

NuMicro® 32位系列微控制器范例代码介绍

文件信息

应用简述	此范例代码在NDA102工程中加入PUSH功能，使得可以通过DALI线接入PUSH开关进行调光功能。
BSP 版本	102_207_NDA102EC1_v03_01_Lib
开发平台	DALI_SLAVE_NDA102EC1 V1.0

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1. 概述

此范例可以使 LED 照明电源可以通过 DALI (Digital Addressable Lighting Interface) 总线接口识别 PUSH 信号，从而实现 DALI 电源拥有 PUSH 调光功能，这样减少了 PUSH 专门接口及电路，降低了成本，并且可以在没有 DALI 主机时，提供了又一种组网调光方式 (如图 1-1) 。

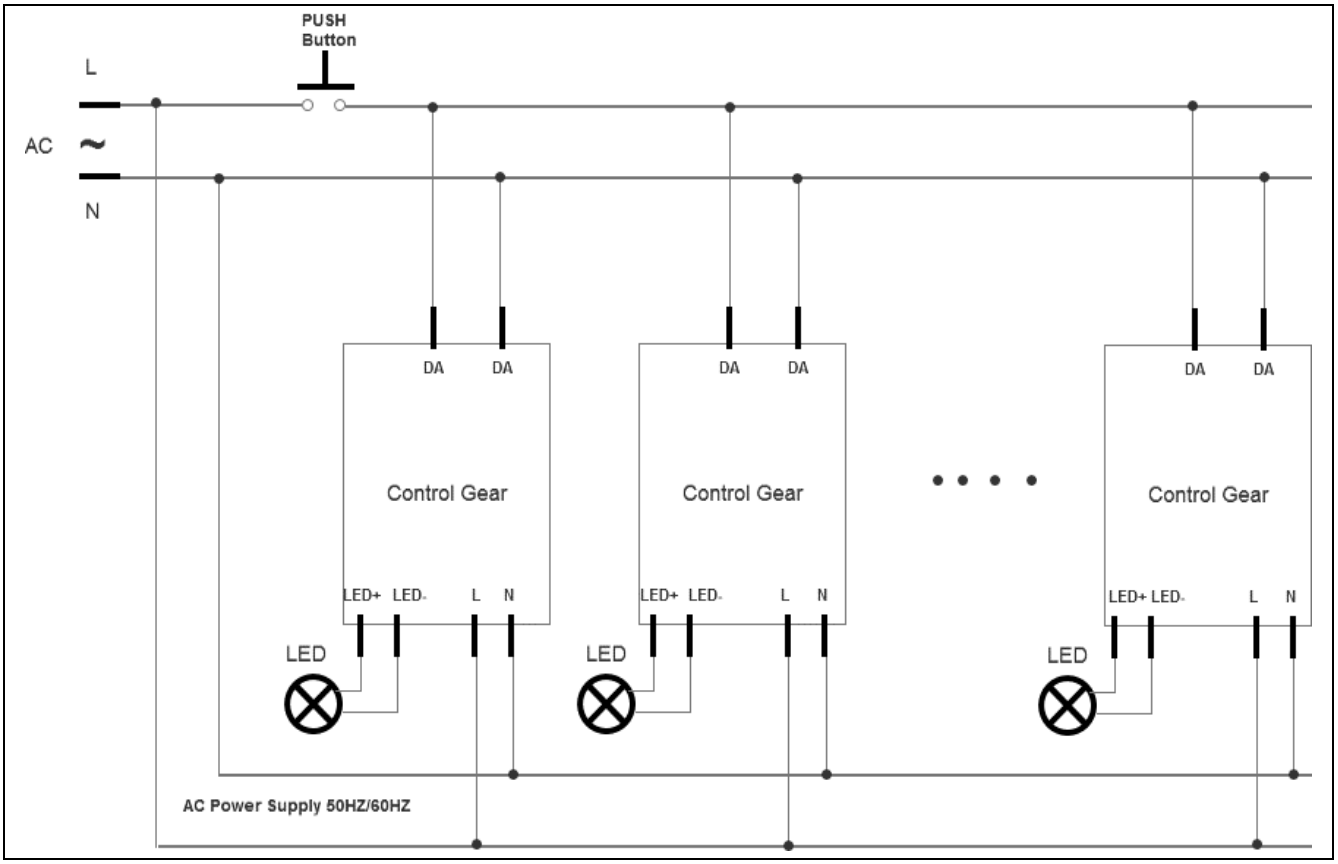

图 1-1 PUSH 多台连接示意图

1.1 原理

DALI 信号遵从 IEC-62386-101 2014 规定，为 1200bps 的曼彻斯特码波形，脉宽为 416us，直流 0~22.5V。由于 IEC-62386-102 2014 还要求 DALI 接口还要有 AC 信号盲接时有耐压 1 分钟以上的保护功能，所以 DALI 信号端硬件上面是可以支持输入 AC 信号作为一种调光信号的，这种利用 AC 信号作为一种调光功能业内一般称为 PUSH 调光功能。

PUSH 信号一般由 AC 市电供电直接到 PUSH 调光信号接口，这种信号由于直接使用市电 AC，有高压抗干扰能力强的特点；并且由于 DALI 总线规定单台设备总线电流小于 2mA，所以 PUSH 调光总线也有低功耗的特点。PUSH 调光信号特征为 50Hz/60Hz 正弦波，通过 DALI 总线电路

会被转变为 50Hz/60Hz 的 5V 方波。NDA102 会通过 IO 方波计数来判断有多少周期，通过周期数来计算按键时间的长短，再通过按键时间长短来定义调光功能，功能框图如下：

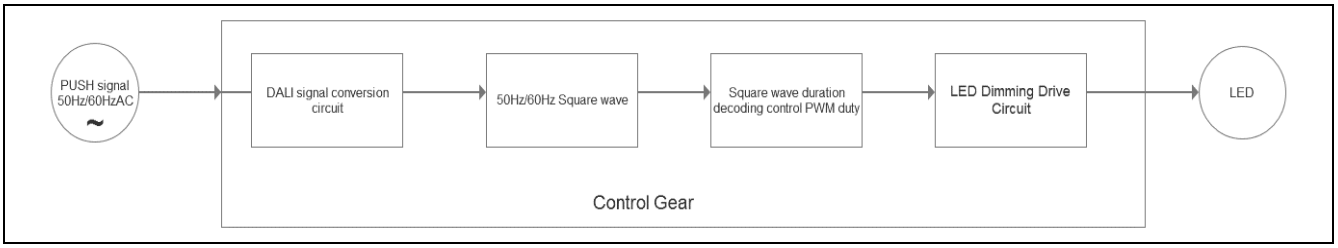

图 1-2 PUSH 功能框图

PUSH 按键功能定义如表 1-1：

	PUSH按键描述	按下至抬起时间Δ*t*	功能描述
PUSH模式	PUSH短按开关	0.1S <Δ*t*<0.6S	PUSH模式下，短按下开灯或关灯
	PUSH长按调光	Δ*t* >0.6S	PUSH模式下，长按调光，调光行程（0-100%亮度）
	PUSH长按复位	Δ*t* >9S	PUSH模式下，长按9S以上，亮度进入复位至一半亮度（50%）

表 1-1 PUSH 按键功能定义

1.2 ICP 烧写程序

连接 ICP 工具，烧写示例程序生成的 NDA102_PUSH_DIM_Control.bin 到一块开发板芯片 (下称 2#机) 。 配置如图 1-3 和图 1-4：

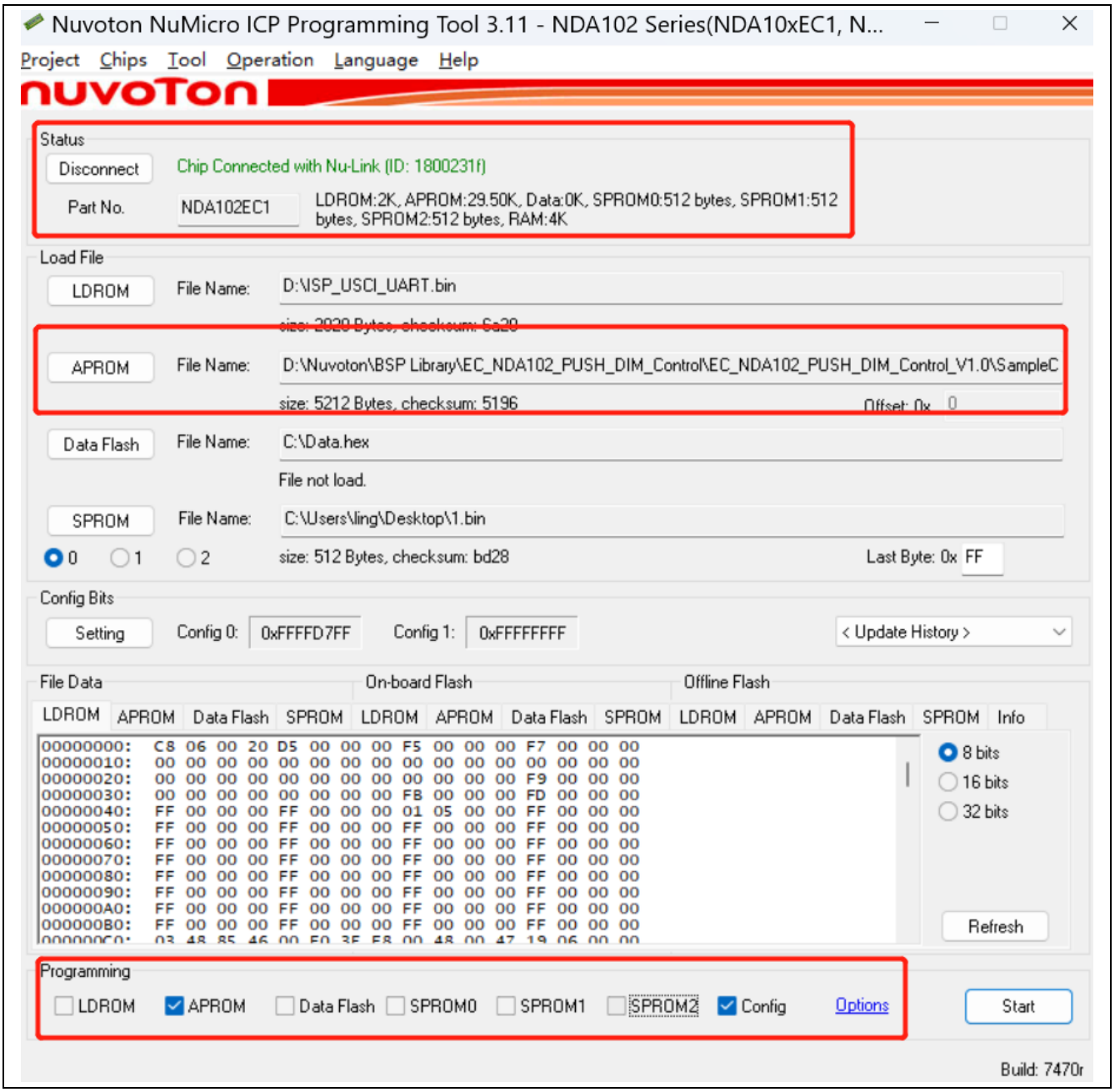

图 1-3 ICP 配置 (2#)

芯片配置为：

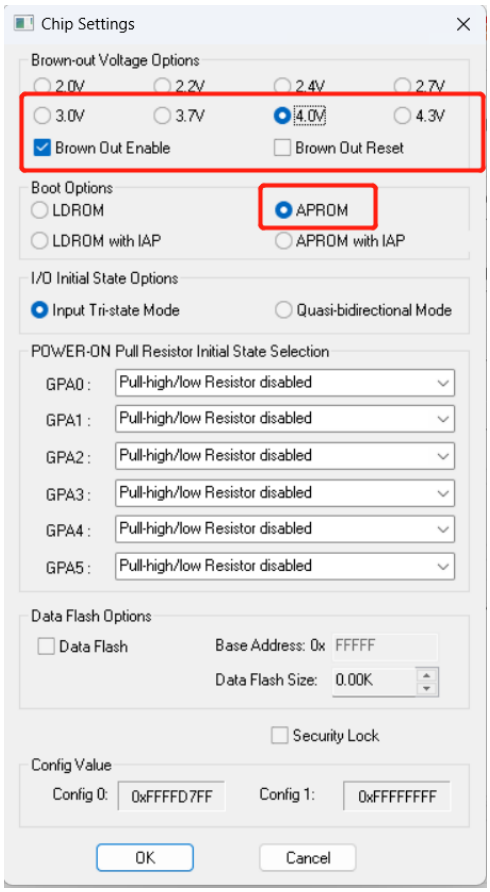

图 1-4 芯片配置 (2#)

连接 ICP 工具，烧写示例程序生成的 NDA102_PUSH_Wave_Out.bin 到另一块开发板芯片 (下称 1#机) 。 配置如图 1-5 和图 1-6：

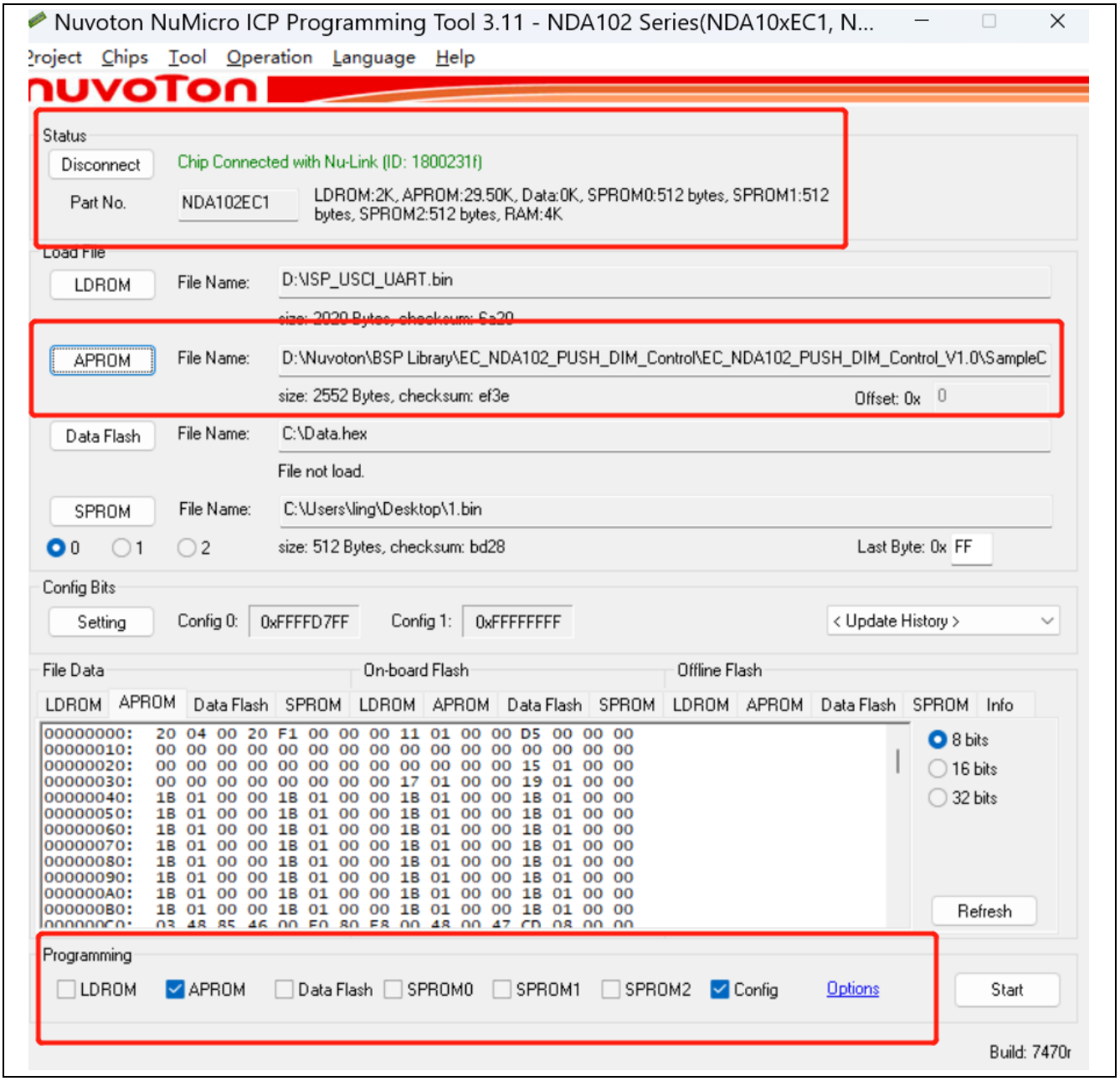
Nuvoton NuMicro ICP Programming Tool 3.11 - NDA102 Series(NDA10xEC1, N...
Project Chips Tool Operation Language Help
Status
Disconnect
Chip Connected with Nu-Link (ID: 1800231f)
Part No.
NDA102EC1
LDROM:2K, APROM:29.50K, Data:0K, SPROM0:512 bytes, SPROM1:512 bytes, SPROM2:512 bytes, RAM:4K
Load File
LDROM
File Name:
D:\ISP_USCI_UART.bin
APROM
File Name:
D:\Nuvoton\BSP Library\EC_NDA102_PUSH_DIM_Control\EC_NDA102_PUSH_DIM_Control_V1.0\SampleC
size: 2552 Bytes, checksum: ef3e
Offset: 0x
Data Flash
File Name:
C:\Data.hex
File not load.
SPROM
File Name:
C:\Users\ling\Desktop\1.bin
0 1 2
size: 512 Bytes, checksum: bd28
Last Byte: 0x FF
Config Bits
Setting
Config 0:
0xFFFFD7FF
Config 1:
0xFFFFFFFF
< Update History >
File Data
On-board Flash
Offline Flash
LDROM APROM Data Flash SPROM LDROM APROM Data Flash SPROM LDROM APROM Data Flash SPROM Info
00000000: 20 04 00 20 F1 00 00 00 11 01 00 00 D5 00 00 00
00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000020: 00 00 00 00 00 00 00 00 00 00 00 00 15 01 00 00
00000030: 00 00 00 00 00 00 00 00 17 01 00 00 19 01 00 00
00000040: 1B 01 00 00 1B 01 00 00 1B 01 00 00 1B 01 00 00
00000050: 1B 01 00 00 1B 01 00 00 1B 01 00 00 1B 01 00 00
00000060: 1B 01 00 00 1B 01 00 00 1B 01 00 00 1B 01 00 00
00000070: 1B 01 00 00 1B 01 00 00 1B 01 00 00 1B 01 00 00
00000080: 1B 01 00 00 1B 01 00 00 1B 01 00 00 1B 01 00 00
00000090: 1B 01 00 00 1B 01 00 00 1B 01 00 00 1B 01 00 00
000000A0: 1B 01 00 00 1B 01 00 00 1B 01 00 00 1B 01 00 00
000000B0: 1B 01 00 00 1B 01 00 00 1B 01 00 00 1B 01 00 00
8 bits
16 bits
32 bits
Refresh
Programming
LDROM APROM Data Flash SPROM0 SPROM1 SPROM2 Config Options
Start
Build: 7470r

图 1-5 ICP 配置 (1#)

芯片配置为：

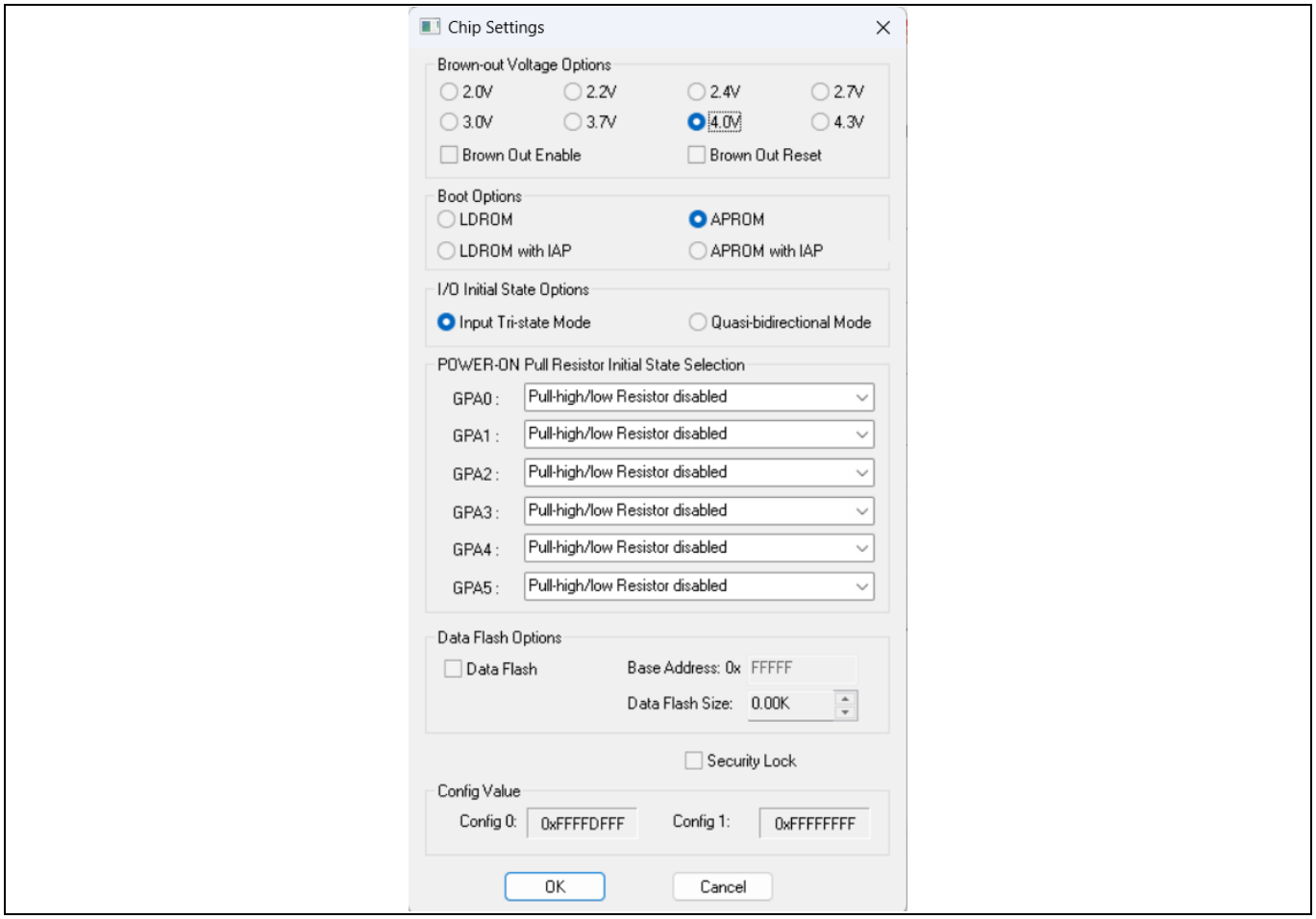

图 1-6 芯片配置 (1#)

1.3 执行结果

示例代码运行时， 开始测试 DALI 电路的 PUSH 功能。由于 DALI_SLAVE_NDA102EC1 V1.0 开发板 DALI 接口的耐压能力不够不能接市电，这里使用另外一块 DALI_SLAVE_NDA102EC1 V1.0 开发板输出模拟 PUSH 信号输入，接线图如下：

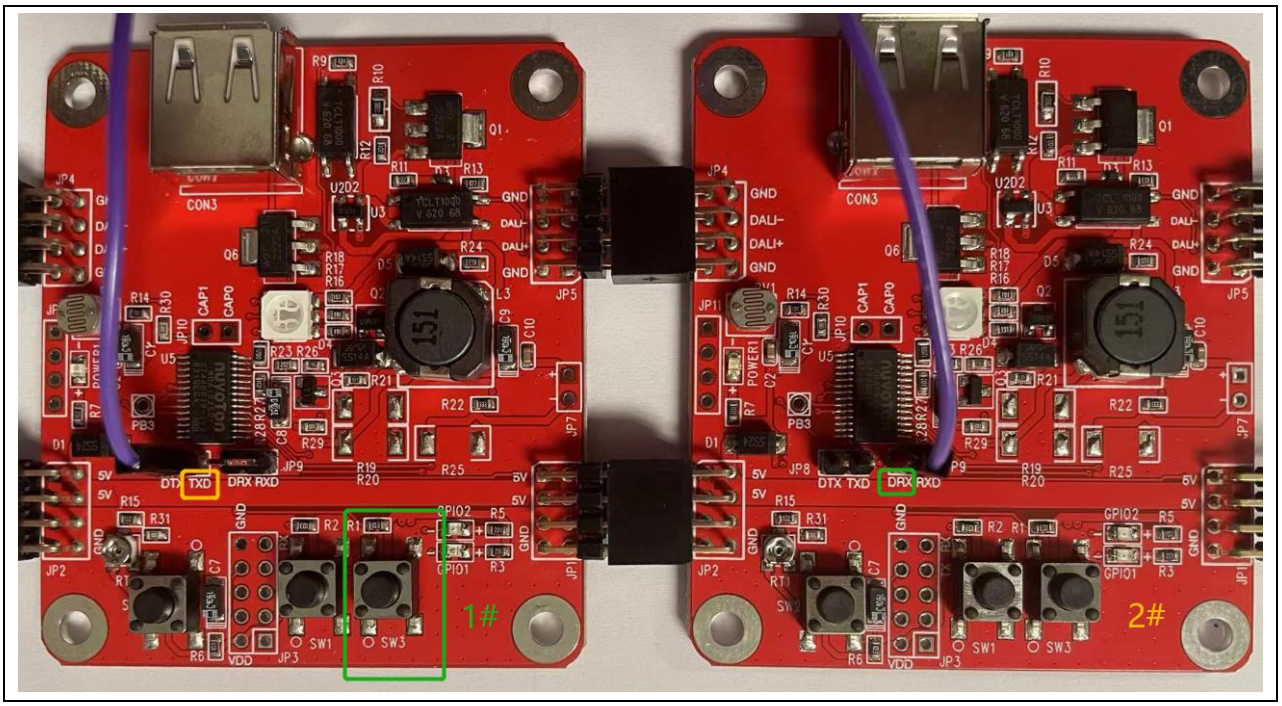

图 1-7 开发板测试 PUSH 功能实物接线图

图 1-7 中 1#机上，模拟 PUSH 按键使用 SW3 (PD.6) ，模拟 PUSH 波形输出信号使用 TXD (PD.3) ；1#机的 TXD (PD.3) 通过杜邦线连接到 2#机的 DRX (PB.0) 上。

第一次上电，2#机默认为 PUSH 模式，灯为最大亮度。

进入 PUSH 模式后，短按开灯：

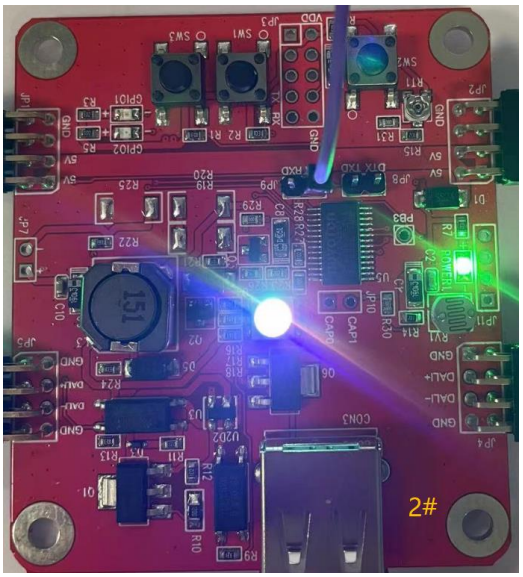

图 1-8 短按 PUSH 开灯

PUSH 模式下，短按关灯：

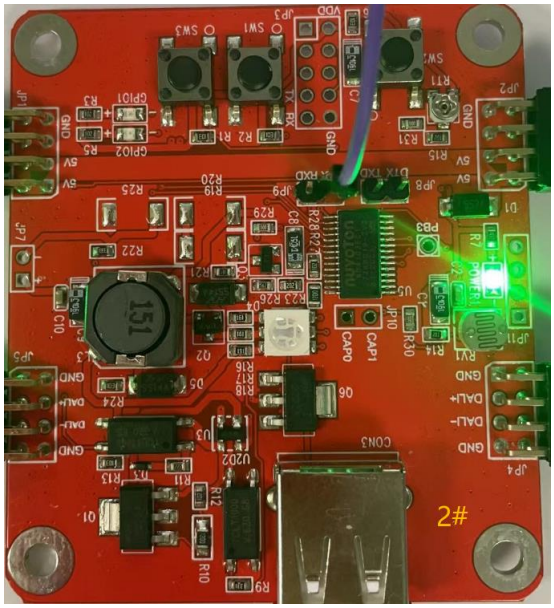

图 1-9 短按 PUSH 关灯

PUSH 模式下，长按 PUSH 按键，调光 0->100%亮度，可以看到灯亮度从 0->100%渐变大约 4S 时间。

PUSH 模式下，再长按 PUSH 按键，调光 100%->0.1%亮度，可以看到灯亮度从 100%->0.1%渐变大约 4S 时间。如图 1-10 调到 0.1%效果。

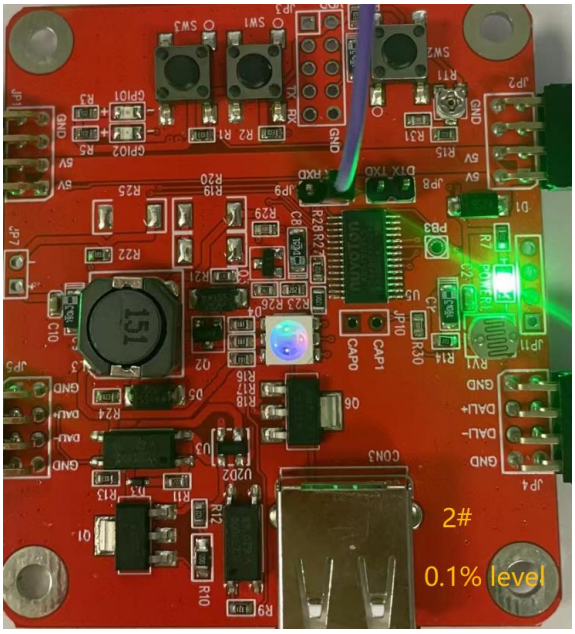

图 1-10 长按 PUSH 调光到最暗

PUSH 模式下，长按 9 秒以上，可以看到 LED 亮度从 0-100%渐变又回到 50%，复位成功。

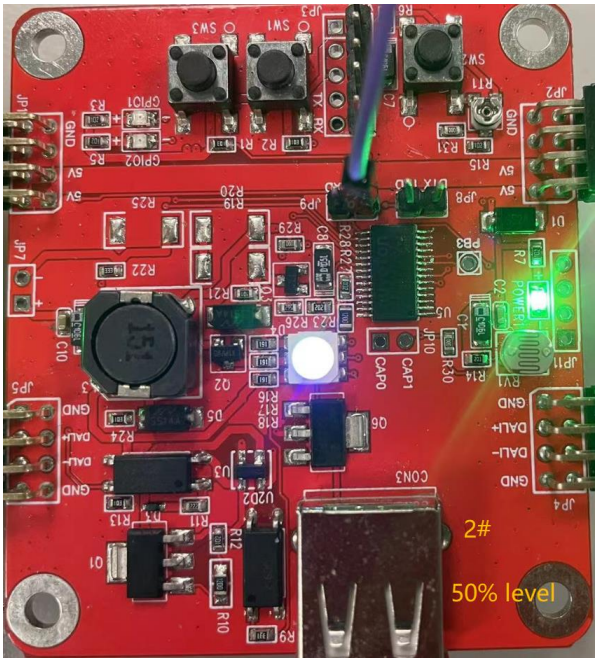

图 1-11 长按 PUSH 超过 9 秒复位

2. 代码介绍

在 main.c 中，在 GPIO 中断函数里面增加 PUSH 信号识别代码。

```
void GPABCD_IRQHandler(void)
{
    GPIO_CLR_INT_FLAG(DRX_PORT, DRX_BIT);

    /*------------- ---------------------------------------------------------------*/
    /* PUSH Decode                                                                  */
    /*-------- ---------------------------------------------------------------------*/
    /*push BUS IS HIGH,no push signal bus=low level so first signal change is high
level=io is low */
    if(DRX == HIGH_DRX)
    {
        /*Push signal recognition and Identify push mode */
        if((DRX_WAVE_Timer > 15) && (DRX_WAVE_Timer < 30)) /*Do not execute for the first
time*/
        {
            if(DimMode == 1) /*In push mode, in push dimming counter.*/
            {
                DRX_Push_Count++;/*counter*/

                if(65530 <= DRX_Push_Count)
                {
                    DRX_Push_Count = 65530;
                }
            }

          /*In non-push mode，Accumulate 10 consecutive pulses，then switch to PUSH mode.*/
            if(1 != DimMode)
            {
                DRX_toPUSH_Enable = 1;        /*Enable swtich mode*/

                DRX_toPUSH_Count++;

                if(DRX_toPUSH_Count == 1)
                    DRX_toPUSH_Timer = 0;      /*Clear and prepare the timing of 10
pulses*/
                else if(DRX_toPUSH_Count > 10)
                {

                    if((DRX_toPUSH_Timer > 600) && (DRX_toPUSH_Timer < 730))
                    {
                        DimMode = 1;           /*Confirm as PUSH signal*/
                        DRX_toPUSH_Enable = 0; /*Turn off to PUSH enable*/
                        DRX_toPUSH_Count = 0;      /*Clear, prepare for next count*/
                    }
                }
            }
        }

        DRX_WAVE_Timer = 40; /*Set the value to prepare for the next count*/
    }
    else
    {

    }
}
```

在 push_DT6.c 中，函数 DRX_PUSH_Count()在 1ms 任务中进行 PUSH 周期计数，变量 Push_Count 计算 PUSH 按键按下抬起的时间 Δt，再根据时间 Δt 判断长按短按来执行对应调光的动作。

```
void DRX_PUSH_Count(void)
{
    if(DimMode == 1) /*Push Mode*/
    {

        if(DRX_Push_Count_OLD == DRX_Push_Count) /*Trigger and trigger time interval:
about 20ms . Customer Define. */
        {
            PUSH_Status_Count++;/*Release key counter*/
        }
        else
        {
            Push_Count++;/*Press key count about 20ms cycle*/
            PUSH_Status_Count = 0;
            PUSH_IN_Flag = 1;
        }

        DRX_Push_Count_OLD = DRX_Push_Count;
        Push_Scan(); /*Change level in push mode*/
    }
}
```

3. 软件与硬件需求

3.1 软件需求

- BSP 版本
  - 102_207_NDA102EC1_v03_01_Lib
- IDE 版本
  - Keil uVersion 5.28

3.2 硬件需求

- 电路组件
  - DALI_SLAVE_NDA102EC1 V1.0 (2pcs)
- 线路示意图
  - 硬件电路使用 DALI_SLAVE_NDA102EC1 V1.0 开发板两套，其中 1#机模拟 PUSH 按键及 PUSH 信号输出，2#机用来作 PUSH 信号接收及 PUSH 调光显示，引脚连接如图 3-1：

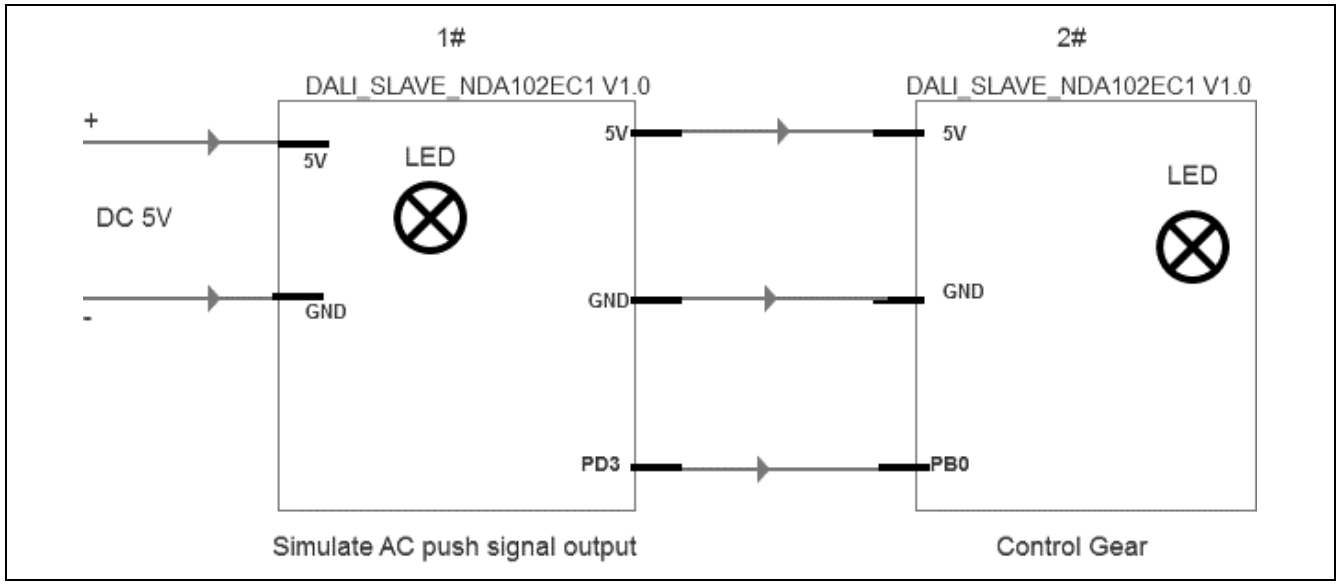

图 3-1 硬件连接示意图

4. 目录信息

目录结构如下图所示。

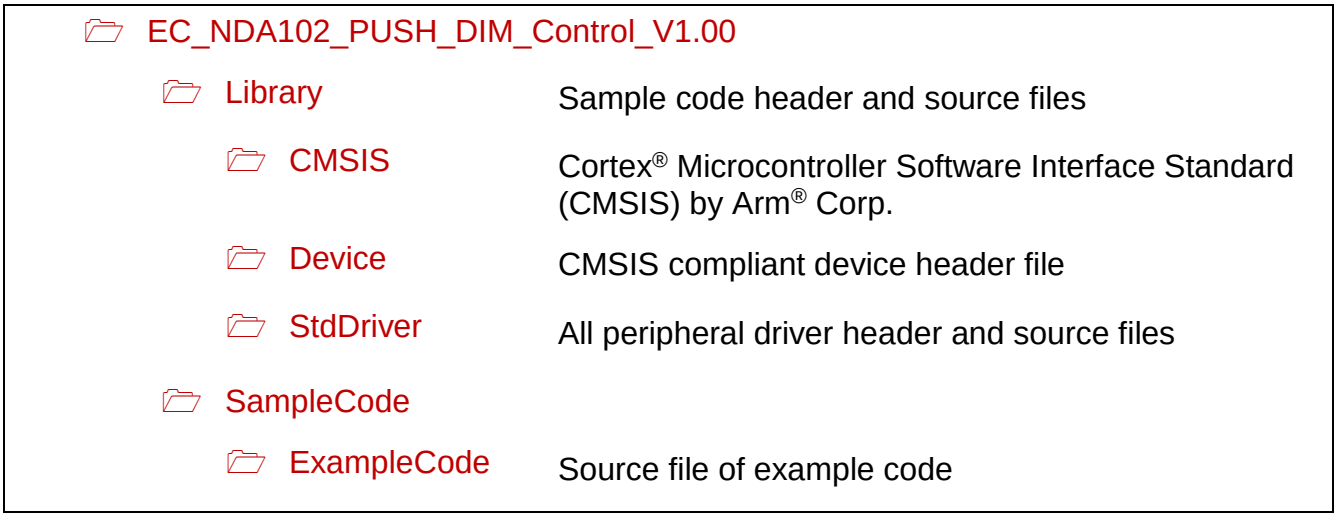

图 4-1 目录信息

5. 范例程序执行

1. 通过目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 *NDA102_PUSH_DIM_Control.uvprojx*(对应 2#机) 和 *NDA102_PUSH_Wave_Out.uvprojx*(对应 1#机)。
2. 进入编译模式界面.
   - 编译
   - 下载
   - 进入/离开仿真模式
3. 进入仿真模式界面.
   - 执行代码

6. 修订纪录

Date	Revision	Description
2023.03.29	1.00	初始发布.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.