

版本：V1.2



Nanochap

杭州暖芯迦电子科技有限公司

**ENS1EVKB**

可编程通用刺激全功能开发板

用户手册



# 目录

文档修订记录 .....	1
<b>1. 开发板图示及说明 .....</b>	<b>3</b>
1.1 适用范围 .....	3
1.2 参考资料 .....	3
1.3 ENS1EVKB全功能开发板图片及说明.....	3
<b>2. ENS1A芯片说明 .....</b>	<b>9</b>
2.1 ENS001芯片简介 .....	9
2.2 引脚说明 .....	10
2.2.1电源部分引脚说明 .....	10
2.2.2其他重要引脚说明 .....	12
<b>3. ENS1EVKB 全功能开发板操作说明 .....</b>	<b>13</b>
3.1 准备工具 .....	13
3.1.1 硬件工具及名称描述.....	13
3.1.2 软件工具及名称描述.....	13
3.2 开发板的供电说明.....	13
3.2.1开发板供电连接方式.....	13
3.2.2 刺激电流输出以及模拟负载说明.....	15
3.3 关键寄存器说明及例程的使用 .....	16
3.3.1 软件准备 .....	16
3.3.2 ENS001的波形发生原理及寄存器配置.....	17
3.3.3 ENS001的波形发生相关寄存器说明 .....	19
3.3.4 ENS001的波形发生例程 .....	21
3.4功耗测试说明 .....	23
<b>4. 联系方式.....</b>	<b>24</b>

# 1. 开发板图示及说明

编写该说明的主要目的是基于 ENS1EVKB全功能开发板使用需求，由本公司技术人员给与技术问题说明，并形成文字形式的报告，可用于客户研发人员自行了解与测试 ENS1EVKB全功能开发板。

## 1.1 适用范围

本文档适用于 ENS1EVKB全功能开发板操作参考。

## 1.2 参考资料

参考资料描述如表1.1。

序号	文档名称	作者	版本
1	NNC_ENS1QFN96L1010A_datasheet.pdf	暖芯迦	V1.2
2	ENS1_Drivers_Specs_User20221206.pdf	暖芯迦	R001
3	可编程通用刺激全功能开发板ENS1EVKB用户手册	暖芯迦	V1.0

表 1.1

## 1.3 ENS1EVKB全功能开发板图片及说明

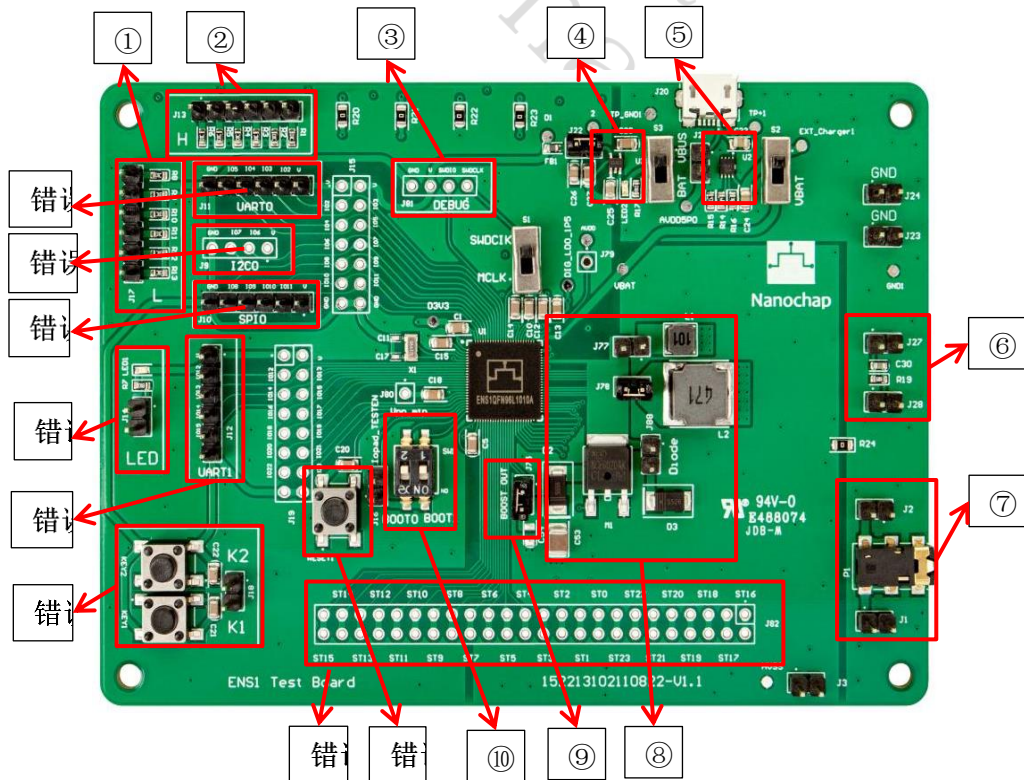


图1.1 ENS1EVKB全功能开发板TOP

- ① 下拉电位排针。
- ② 上拉电位排针。
- ③ SWD 调试接口，使用 J-Link 烧录器。
- ④ SGM2036-3.3 芯片，将输入电压稳压到3.3V供3.3V电源域使用。
- ⑤ SGM40561-4.2 锂电池充电管理芯片，可以用于给3.7V锂电池充电。
- ⑥ 模拟负载。

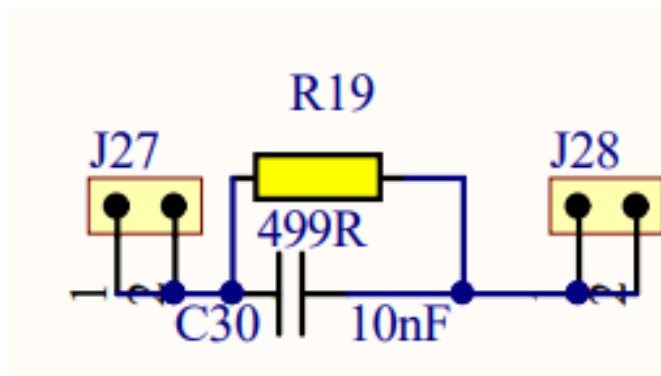


图1.2 模拟负载

- ⑦ 用于刺激电流的输出与测试。

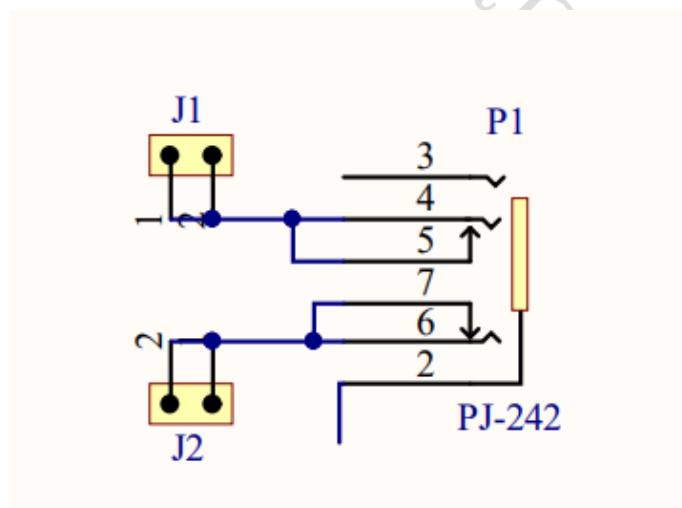


图1.3 2.5mm电刺激线接口

⑧ 为BOOST部分电路，包含电感，电容、肖特基二极管（与芯片内部的肖特基二极管并联），芯片内部集成了MOSFET，芯片的SW引脚连接内部MOS管的漏极D，实际使用时J77和J78选一个（电感二选一），如图为外部元器件的连接位置。

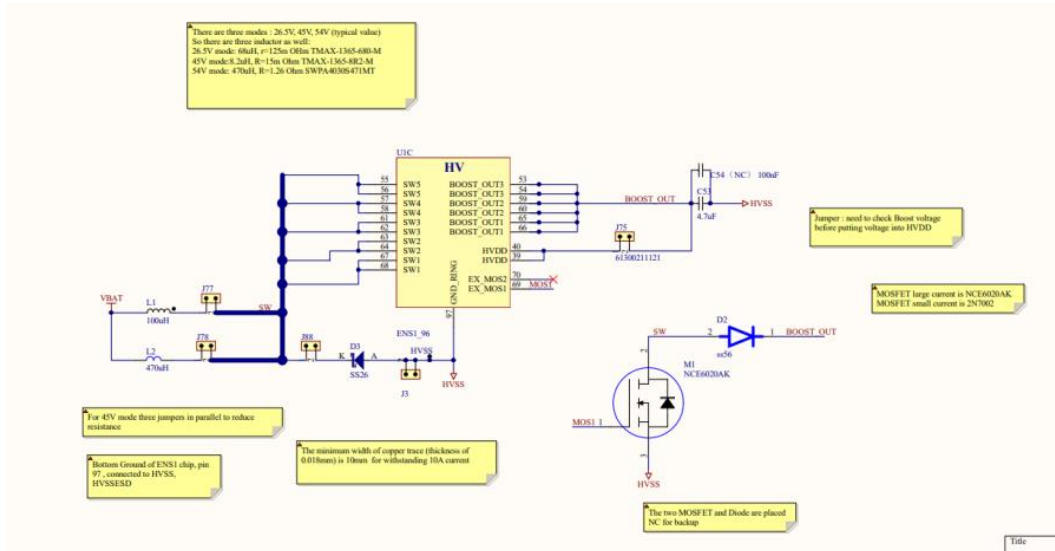


图1.4 BOOST部分电路

⑨ 75是芯片的BOOST\_OUT与HVDD引脚之间的跳帽，BOOST电路为内部的刺激驱动器供能，boost\_out引脚的输出电压通过寄存器设置来控制。

例如：程序中设置输出电压为45V，则使能后可以在boost\_out端测到45V左右的电压，将此电源输入到HVDD引脚后，再配置波形参数设置相关的寄存器就可以生成由恒流源控制的刺激驱动器输出相应的波形。

⑩ BOOT模式切换开关。

Boot mode selection		Boot mode	Aliasing
Boot1 pin	Boot0 pin		
X	0	MTP base	MTP main area from base is selected as boot area for application code usage
0	1	MTP high 4KByte	MTP high 4KByte area is selected as boot area for bootloader usage
1	1	Embedded SRAM	Embedded SRAM is selected as boot area for debug usage

图1.5 BOOT模式说明

11 RESET 复位按键。

12 电刺激引脚引出位置。

■ DRIVER A :

其中：ST0 - ST7 为driverA 的电极引脚，两两组成一组通道（通道1：ST0+ST1，通道2：ST2+ST3，通道3：ST4+ST5，通道4：ST6+ST7），内部电路简图如图：

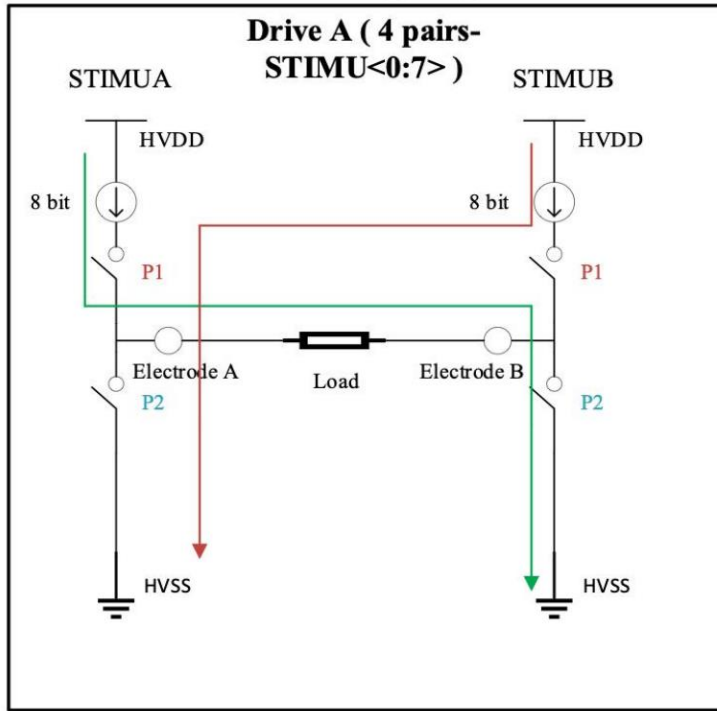


图1.6 DRIVER-A 电极工作原理示意图

■ DRIVER C :

ST8 - ST23 为DRIVER C的电极引脚，STIMU8 / 9 / 10 / 11 / 12 / 13 / 14 / 15 这8个电极是带有电流源的电极，STIMU 16 / 17 / 18 / 19 / 20 / 21 / 22 / 23 这8个电极是配合有源电极的开关电极，使用driverC时，在程序中将STIMU8-15中的一个电极与STIMU16-23中的一个switch 电极组合，形成一个双电极的通道使用。

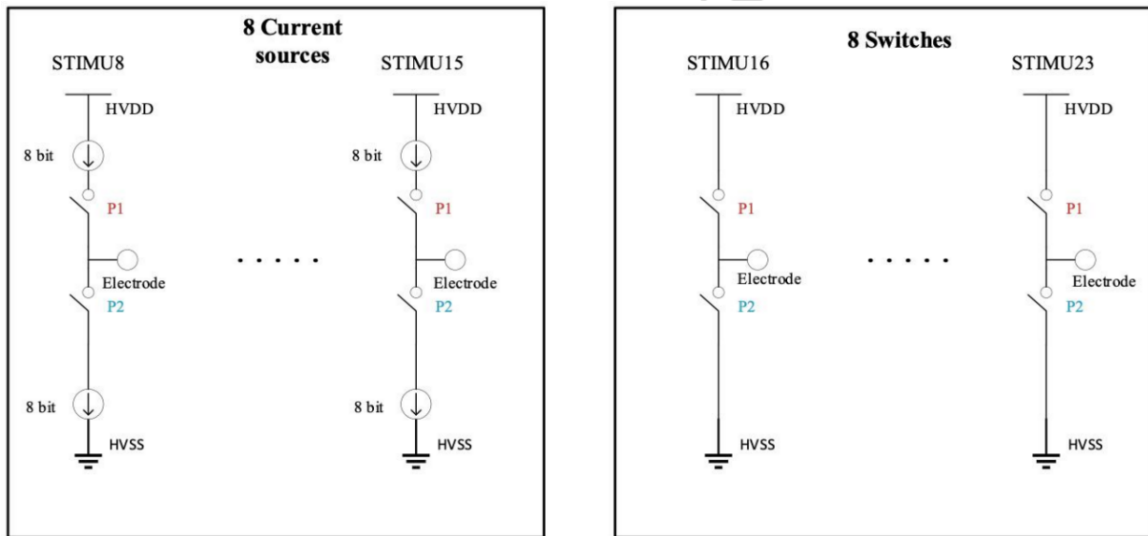


图1.7 DRIVER-C 电极工作原理示意图

■ DRIVER B:

ST0 - ST23 共23个电极又可以作为driverB 的电极使用，当使用driverB时， driverA 和C便暂时不能使用。

在driverB的使用中，所有的电极使用一个电流源，此时STIMU23是作为公共电极和其他电极组合使用，且每次同一时间只能有一对电极工作。

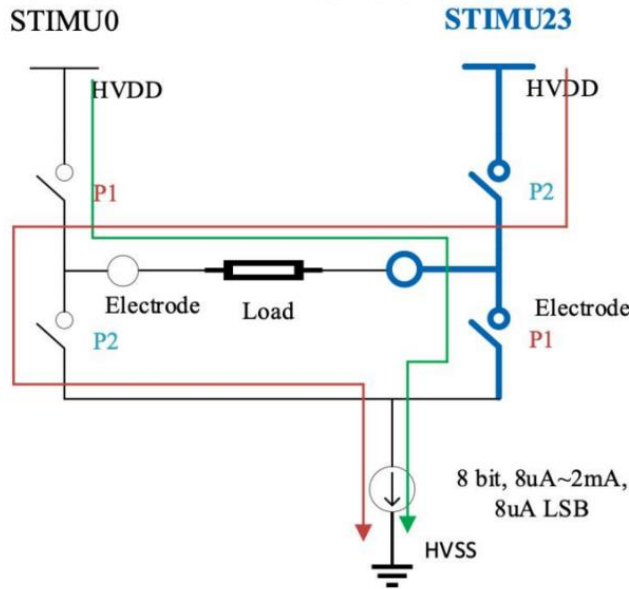


图1.8 DRIVER-B 电极工作原理示意图

13 按键实验电路。

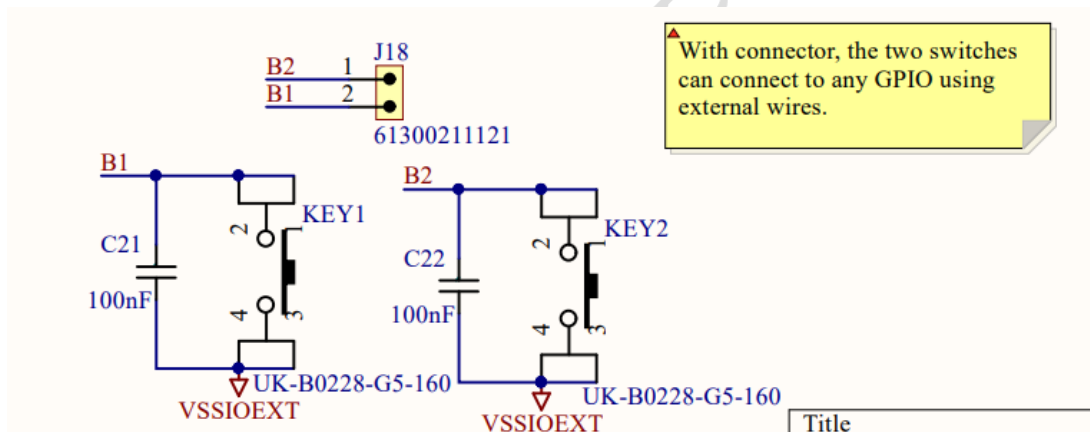


图1.9 按键电路

14 串口1引脚引出插座。



15 LED实验电路。

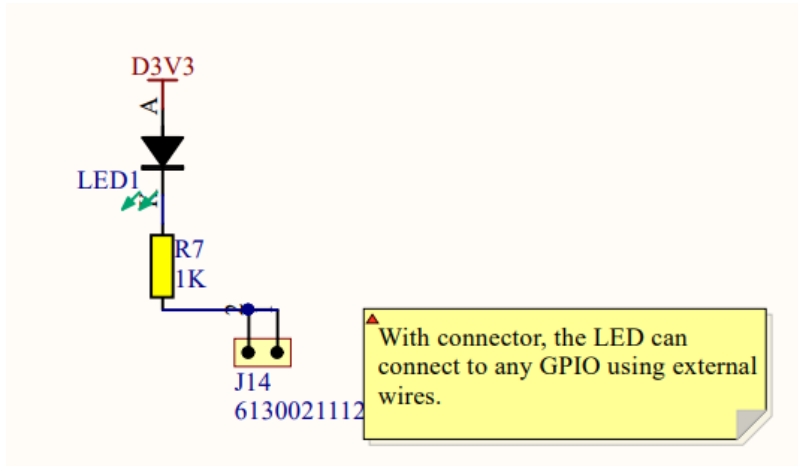


图1.10 LED电路

16 SPI0引脚引出插座。

17 IIC引脚引出插座。

18 串口0引脚引出插座。

# 2. ENS1A芯片说明

## 2.1 ENS001芯片简介

ENS001芯片有三种型号，分别是可编程通用刺激芯片ENS001-96L10x10-A-PFTS（全功能）、可编程通用刺激芯片ENS001-64L8x8-A3-PFTS（四通道）、可编程通用刺激芯片ENS001-64L8x8-A2-PFTS（双通道）。

ENS1A芯片集成了cortex-m0内核，32KB MTP存储，8KBSRAM，数字通信接口有UART\*1,SPI\*1,IIC\*1，1个12-bitADC，温度传感器，2路比较器，PGA，电池充电管理模块，低电压检测，12个GPIO（部分与数字通信接口复用），计时器部分有：实时时钟、定时器、看门狗、脉宽调制、32位或16位双定时器等，另外集成有专用的高压刺激驱动器。

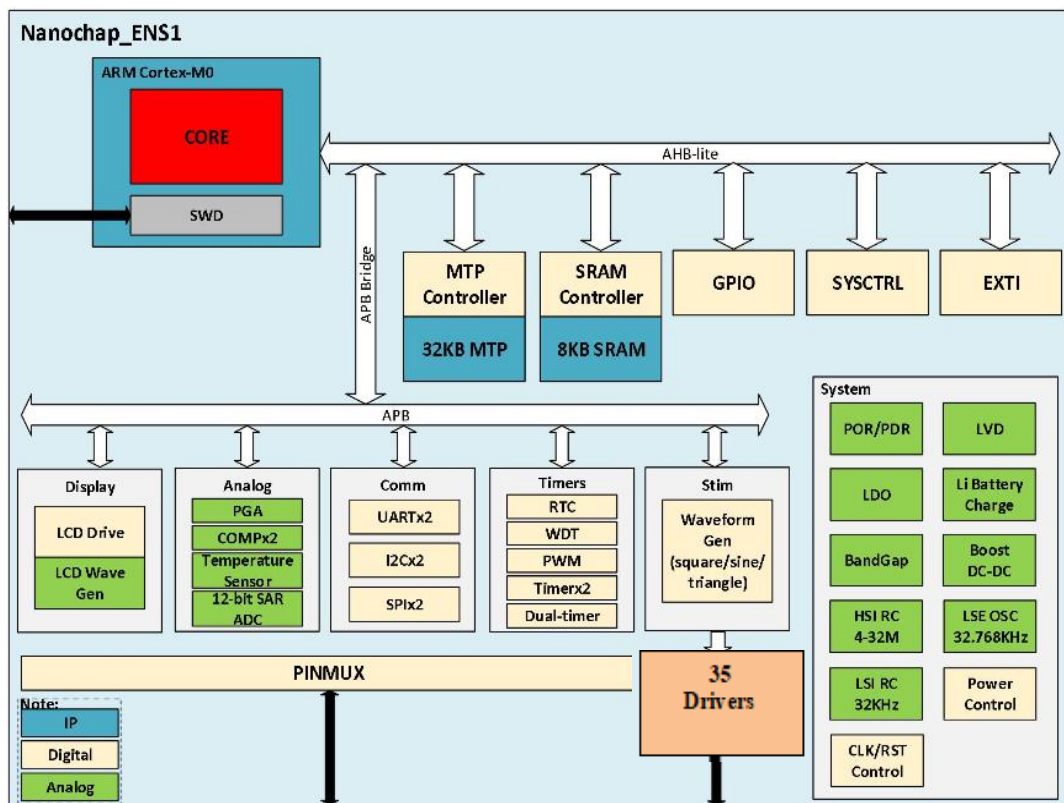


图2.1 ENS001 芯片框图

其中刺激驱动器部分：全功能芯片共有STMU0-STMU23共24个电刺激电极引脚，四通道芯片有STMU0-STMU7共8个电刺激引脚，2通道芯片有STMU0-STMU3共4个电刺激引脚。

型号	ENS001-96L10x10-A-PFTS (全功能)	ENS001-64L8x8-A3-PFTS (四通道)	ENS001-64L8x8-A2-PFTS (双通道)
封装	QFN96L/10*10mm	QFN64L/8*8mm	QFN64L/8*8mm
电刺激通道数量	DRIVER-A: STMU0-STM7 (8电极, 可组成4个独立双向通道, 最高60V) DRIVER-B: STMU0-STM23 (24电极, 可组成23个通道分时输出, 最高60V) DRIVER-C: STMU8-STM23 (16电极, 可组成8个独立双向通道, 最高60V)	DRIVER-A: STMU0-STM7 DRIVER-B: 无 DRIVER-C: 无	DRIVER-A: STMU0-STM3 DRIVER-B: 无 DRIVER-C: 无
输出电流能力/单元电流	DRIVER-A: 每通道33uA-120mA (255级), 单元电流33uA~264uA, 8档可调 (最大可输出120mA电流, 配置方法见芯片规格书) DRIVER-B: 每通道8uA-2mA, 8uA步进共255级 DRIVER-C: 每通道50uA-100mA (255级), 单元电流50uA, 72~204uA (8档可调)	DRIVER-A: 每通道0-120mA (最大可输出120mA电流, 配置方法见芯片规格书) 单元电33uA~264uA	DRIVER-A: 每通道0-120mA (最大可输出120mA电流, 配置方法见芯片规格书) 单元电33uA~264uA

表2.1 ENS001芯片三种型号区别

## 2.2 引脚说明

### 2.2.1 电源部分引脚说明

电源分为2个部分, 第一部分是5V及小于5V工作电压电源, 第二部分是BOOST升压及刺激驱动器部分的电源。

#### 1) 第一部分

- **AVDD5P0:** 需要给锂电池充电时此引脚输入5V, 并将电池正极与VBAT接通, 配置程序后可以给电池充电。
- **VBAT:** 芯片供电与锂电池充电引脚。
  - 与锂电池正极连接 (使用锂电池时)
  - 直接输入3.6V-5V直流电源 (不使用电池时)
- **VSYS\_LDO:** 内部4.23VLDO输出引脚, 仅需连接去耦电容 (此LDO的输出用于锂电池充电)。
- **VDD\_IO1/2/3:** 三个引脚均为3.3V电源输入引脚, 为3.3VIO部分电路供电。
- **VDD\_DIG:** 需要在外部连接DIG\_LDO引脚, 提供1.8V电源给内核工作使用。
- **DIG\_LDO:** 内部LDO生成的1.8V电源输出引脚。

## 2) 第二部分

- **BOOST\_OUT1/2/3:** ENS001集成了BOOST升压电路中的部分器件，最终电压输出引脚为 BOOST\_OUT1/2/3。
- **SW1/2/3/4/5:** BOOST升压路径有两种模式可供选择，即内部模式和外部模式，在使用内部模式下，使用芯片内集成的MOS管，肖特基二极管，电压反馈电路（过压保护），芯片的SW引脚连接内部MOS管的漏极D。设计电路时仅需要外加（图 2.2 红框中）功率电感、肖特基二极管（与内部并联）、电容组成完整的BOOST电路。

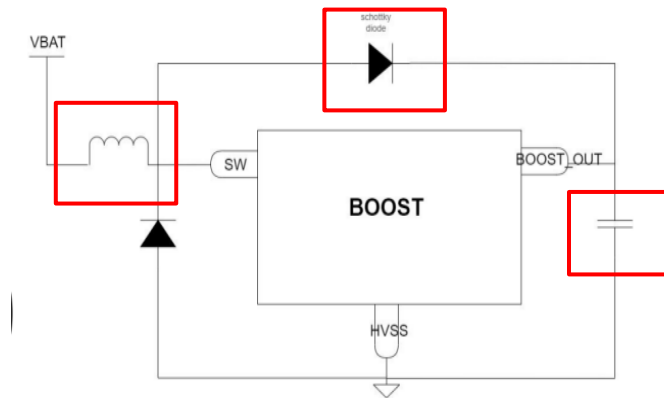


图2.2 ENS001 的BOOST电路示意图

如图2.3中，SW1/2/3/4/5引脚连接内部MOS管的D极，内部以并联方式存在，这样设计是为了提供较大的电流承受能力，D3可以忽略不使用。

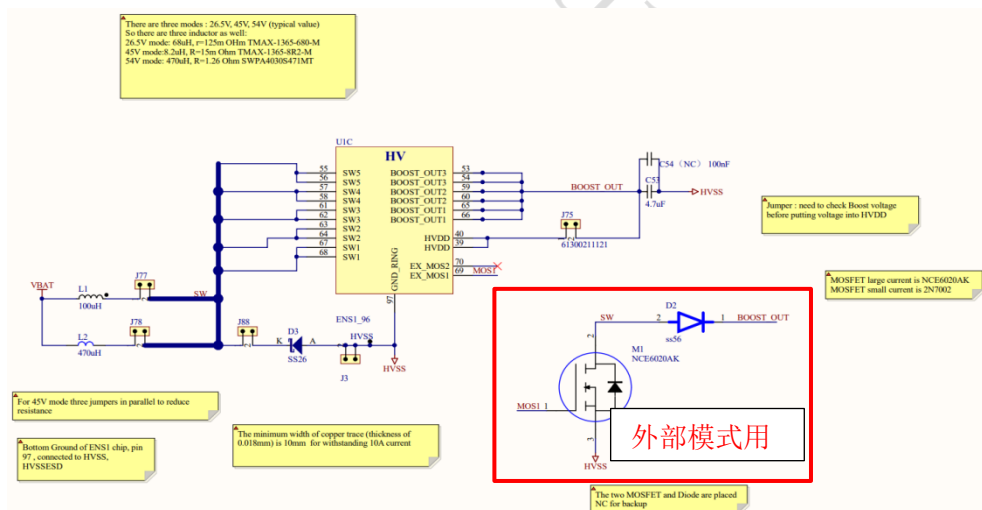


图2.3 ENS001 的BOOST电路示意图

在使用外部模式下，需要多使用一个外部的MOS管（如图2.3）。

- **HVDD:** ENS001集成了刺激驱动器，用于控制生成确定参数的刺激波形，且以恒流源的形式输出，可以通过寄存器配置输出电流大小、脉宽、频率等参数。刺激驱动器的电源输入（HVDD引脚）接BOOST\_OUT引脚。

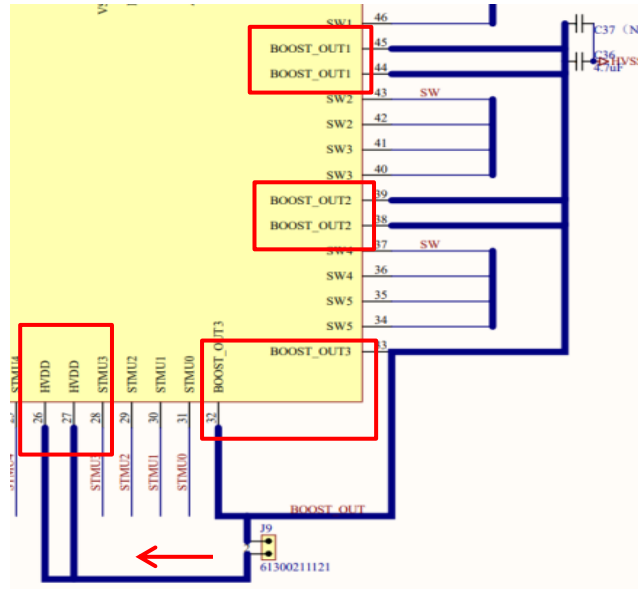


图2.4 ENS001 的boost\_out和HVDD引脚连接示意

## 2.2.2其他重要引脚说明

- **GPIO0:** 默认为SWCLK功能，即SWD的时钟接口，此引脚又可复用HSE\_CLK功能，即使用外部高速时钟作为主时钟输入，此时连接为外部有源晶振的OUT端（一般使用内部高速时钟）。
- **CLK32K\_OUT:** 接32K晶振引脚。
- **CLK32K\_IN:** 接32K晶振引脚。
- **BOOTSEL0:** BOOT模式选择引脚0。
- **BOOTSEL1:** BOOT模式选择引脚1。
- **nRST:** 芯片复位引脚，默认为高电平，拉低时触发复位。

# 3. ENS1EVKB 全功能开发板操作说明

## 3.1 准备工具

### 3.1.1 硬件工具及名称描述

- J-Link 仿真器（推荐V9版本，其它如ST Link等带有SWD接口的仿真器也可使用）
- ENS1EVKB 全功能开发板
- Micro USB 线
- 2.5mm一出二理疗仪导线和电极贴片（选配）
- OLED模块（选配）
- 示波器
- 500欧姆电阻
- 串口工具（TTL）

### 3.1.2 软件工具及名称描述

- keil5.23 开发工具
- ENS001芯片的烧录算法（位置：NNC\_ENS1QFN96L1010A--全功能ENS开发资料\5、例程\3、开发环境\xxx.FLM）

## 3.2 开发板的供电说明

### 3.2.1 开发板供电连接方式（此处命名使用原理图中的信号名称）

- ENS001芯片有3组电源输入引脚，分别是：AVDD5P0、D3V3、VBAT
- 1) AVDD5P0仅在需要给锂电池充电时使用，输入5V电源后，通过内部LDO生成4.2V电源通过VBAT引脚给锂电池充电。
  - 2) D3V3需要输入3.3V电源，为GPIO提供电源。
  - 3) VBAT连接3.7V锂电池时可以输出给电池充电，也可以作为输入为芯片供电。  
VBAT和D3V3必须有正常输入芯片才能正常工作。

■ 开发板供电方法:

- 1) 通过micro-USB接口供电，需要将S2开关的 1-3 飞线短接。

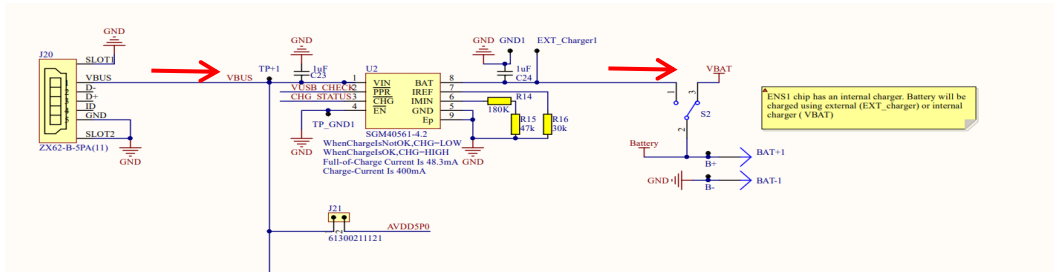


图3.1 Micro USB接口输入供电示意

- 2) 通过3.7V锂电池供电，需要先焊接锂电池到开发板背面的BAT+和BAT-焊点。将开关S2拨到2-3。

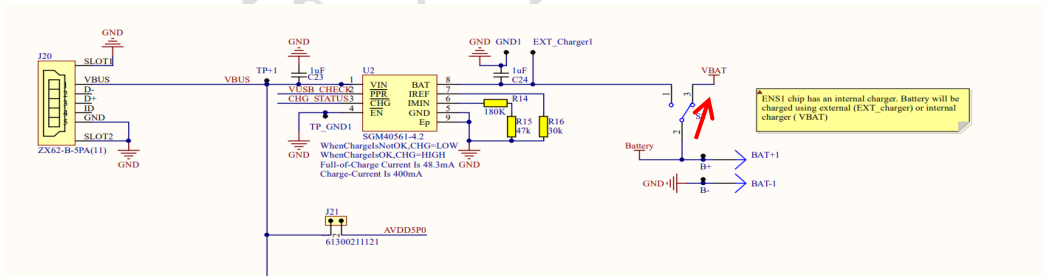


图3.2 锂电池供电示意

- 3) 通过ENS001芯片给电池充电，将J21短接给AVDD5P0引脚输入电源，将S2开关切换至2-3。

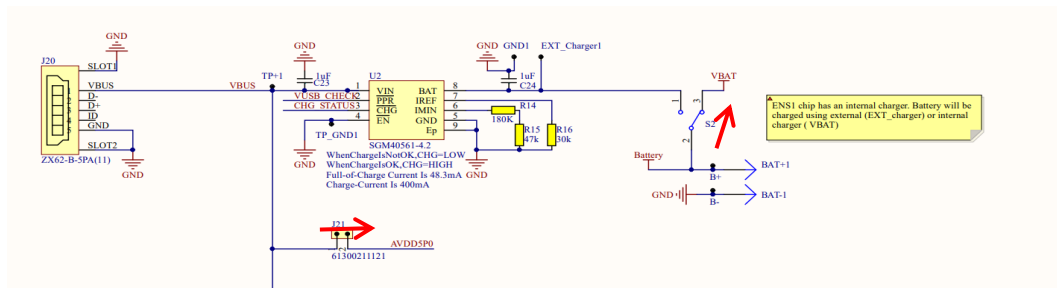


图3.3 ENS001为锂电池充电示意

4) 使用SGM40561-4.2芯片可提供4.2V电压为锂电池充电。

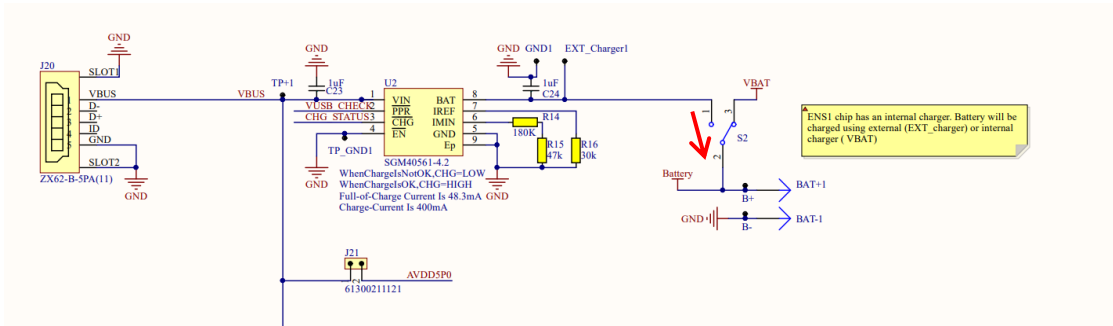


图3.4 SGM40561芯片为锂电池充电示意

### 3.2.2 刺激电流输出以及模拟负载说明

- 全功能电刺激输出引脚如图3.5，图3.6中的电容电阻模拟负载可用于波形测试，2.5mm耳机孔可连接一出二理疗仪导线和电极贴片用于体验不同刺激波形的实际体感。

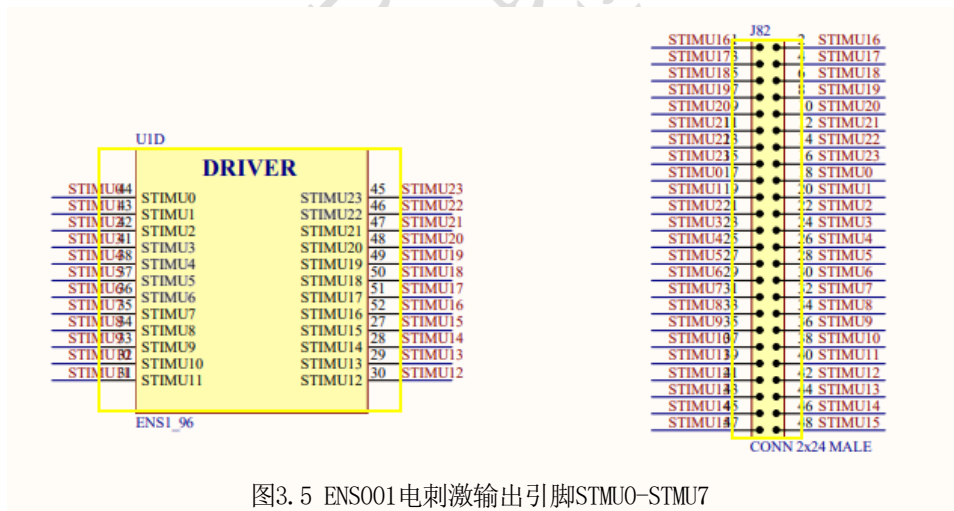


图3.5 ENS001电刺激输出引脚STIMU0-STIMU7

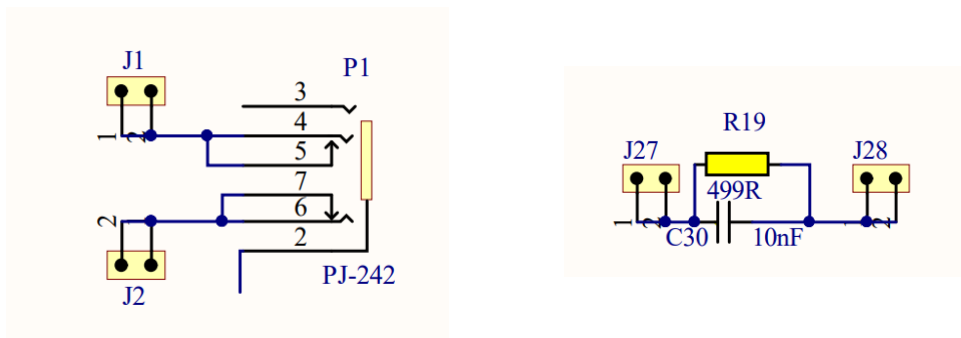


图3.6 模拟负载与电刺激线接口



## 3.3 关键寄存器说明及例程的使用

### 3.3.1 软件准备

- 1) 将 NNC\_ENS1QFN96L1010A--全功能ENS开发资料\5、例程\3、开发环境路径下的 XXXXX.FLM 文件复制到 KEIL安装路径下的 /ARM/Flash 文件夹下。
- 2) 打开\NNC\_ENS1QFN96L1010A--全功能ENS开发资料\5、例程\1、波形发生例程路径下的任一例程，打开配置窗口-> Debug 页面（图3）-> Setting 窗口-> Flash Download 页面 -> 添加XXXXX.FLM 包。

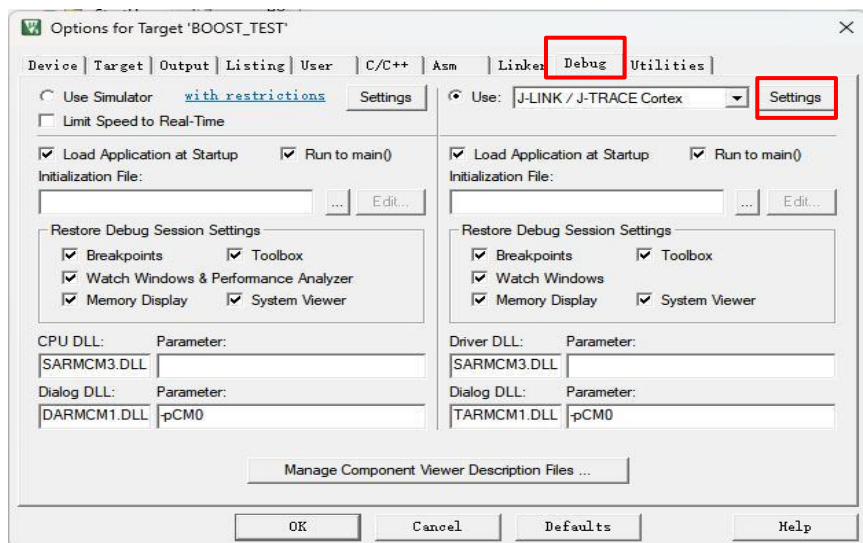


图3.7 KEIL的options界面

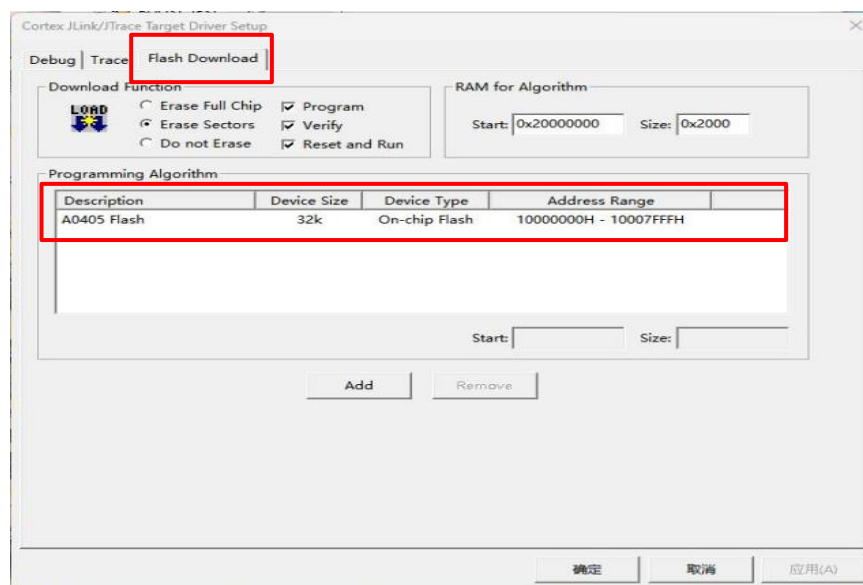


图3.8 KEIL的flash烧录文件选择界面

3) 选择编译器版本（选择为您软件现有的版本即可）

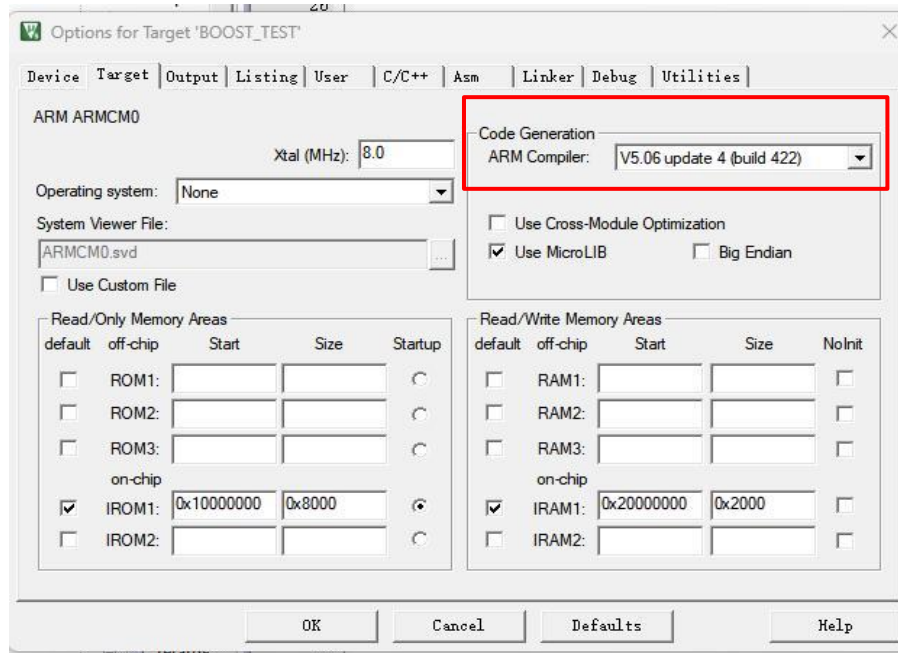


图3.9 编译器选择

### 3.3.2 ENS001的波形发生原理及寄存器配置

ENS001可以生成双向电流刺激波形，其生成波形的方式为“描点法”，每个周期的波形由128个点“描出”，负半周期的波形由正半周期镜像或者“拉伸”得到，因此在“描点”时，只需描出正半周期的64个点的电流值即可。

以ST0+ST1(DRIVERA - 通道1)生成双向方波步骤为例：

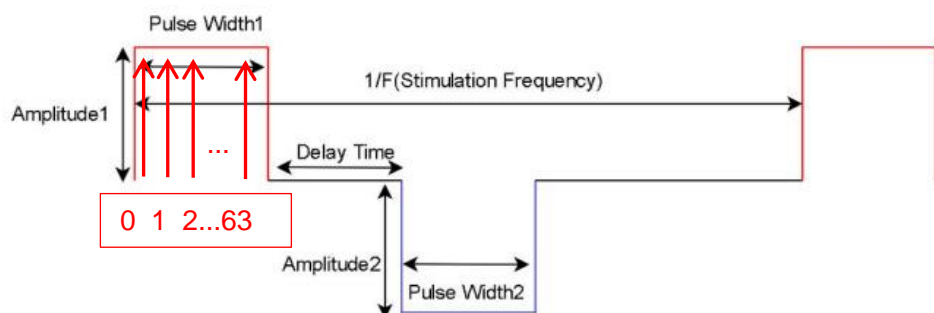


图3.10 双向方波

- 1) 在ST0和ST1中间连接一个500欧姆电阻；
- 2) 配置ADDR\_WG\_DRV\_CONFIG\_REG寄存器（datasheetP.182）：使能死区时间（bit0: rest enable），使能负半周期波形发生（bit1: negative enable），使能静默时间（bit2: silent enable），使能负半周电极的SOURCE开关（bit3: source B enable）；
- 3) 配置ADDR\_WG\_DRV\_HLF\_WAVE\_PRD\_REG寄存器，设置正半周期的脉宽时间（单位us），可以设置的范围是（2-99999us）；
- 4) 配置ADDR\_WG\_DRV\_REST\_T\_REG寄存器，设置死区时间（图3.7中的Delay Time），范围是（1-255us）；
- 5) 配置ADDR\_WG\_DRV\_NEG\_HLF\_WAVE\_PRD\_REG寄存器，设置负半周期的脉宽时间（单位us），可以设置的范围是（2-99999us）；
- 6) 配置ADDR\_WG\_DRV\_SILENT\_T\_REG寄存器，设置静默时间（图3.7中负半周结束后的一段时间）；
- 7) 设置电流：输出的电流大小由 ADDR\_WG\_DRV\_ISEL\_REG 和ADDR\_WG\_DRV\_INT\_REG 寄存器共同决定，输出电流大= $(ADDR\_WG\_DRV\_ISEL\_REG+1) * (ADDR\_WG\_DRV\_INT\_REG + 1) * 33\mu A$ ，其中33uA 是最小单元电流；
- 8) 启动刺激输出：将 ADDR\_WG\_DRV\_CTRL\_REG 赋1，启动输出；
- 9) 连接示波器，使用双探头分别测量电阻两端的输出电压，如图：

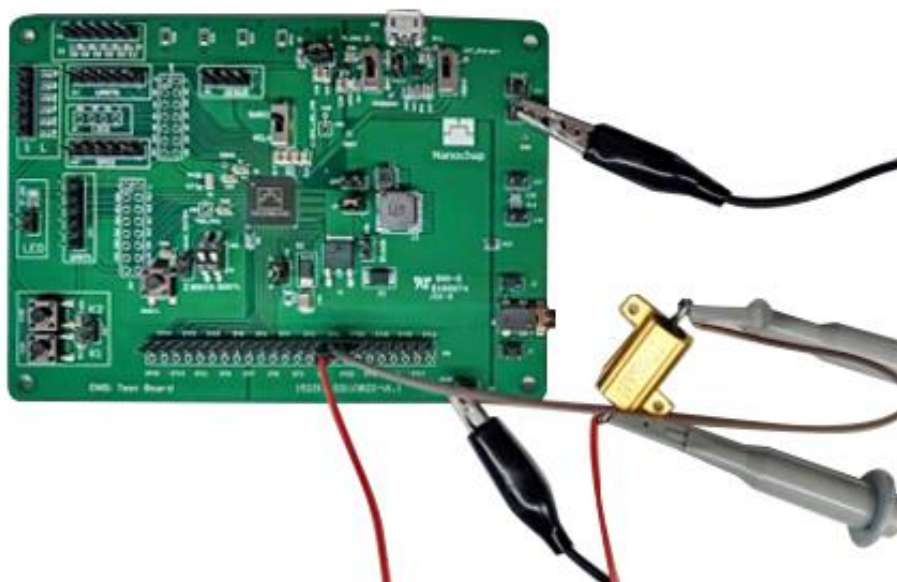


图3.11 示波器探头连接方法

10) 需要使用示波器的math功能，将CH1与CH2的波形做差，得到最终波形。

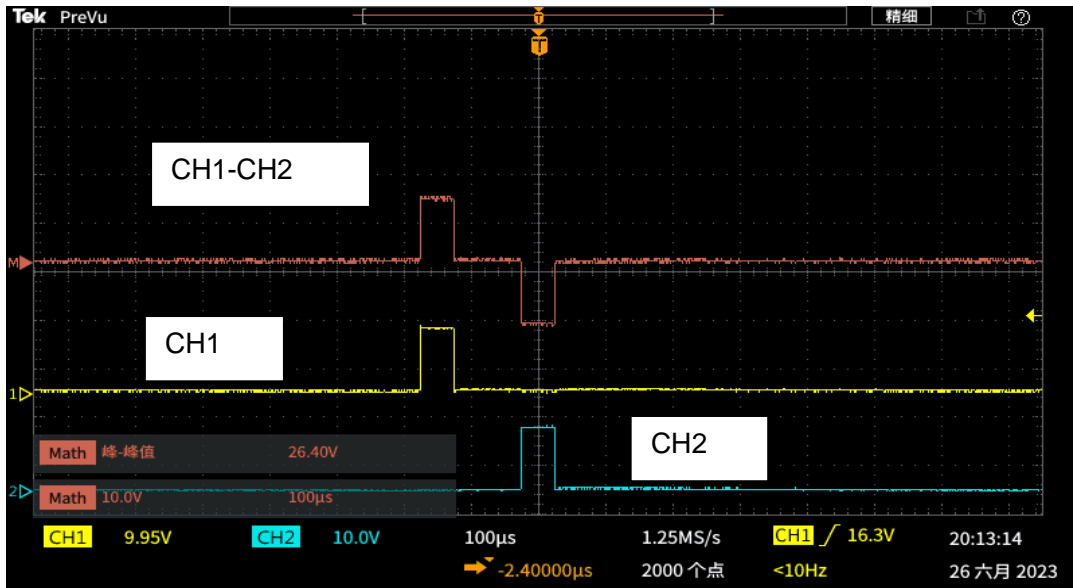


图3.12 示波器处理

### 3.3.3 ENS001的波形发生相关寄存器说明

输出电流公式：

$$\text{DRIVER-A} : (\text{ADDR\_WG\_DRV\_IN\_WAVE\_REG}+1) * (\text{ADDR\_WG\_DRV\_ISEL\_REG}+1) * 33\mu\text{A}$$

$$\text{DRIVER-B} : (\text{ADDR\_WG\_DRV\_ISEL\_REG}+1) * 8\mu\text{A}$$

$$\text{DRIVER-C} : (50\mu\text{A} + \text{ADDR\_WG\_DRV\_ISEL\_REG} * 22\mu\text{A}) * (\text{ADDR\_WG\_DRV\_IN\_WAVE\_REG}+1)$$

表3.1 部分寄存器说明

寄存器名	寄存器长度	读写	功能描述
ADDR_WG_DRV_CON FIG_REG	<7:0>	RD/W R	bit0: 使能死区时间，如果使能了这一位，则<ADDR_WG_DRV_REST_T_REG>寄存器不能赋值为0；如果失能了这一位，则需要赋值为0，否则无波形输出。 Bit1: 负半周期波形使能，如果需要输出负半周期波形，需要将这一位使能。 Bit2: 静默时间使能，静默时间是负半周期波形结束后的不输出时间。 Bit3: 负半周期的source开关，需要负半周期波形时，此位也需要置1。 Bit4: 交替模式使能。 bit5: 连续输出波形模式（无中断时）。 Bit6: 当使用多个通道时需要使能，如需要使用driverA的第二个通道，不使能此位，则第二个通道步输出波形。

ADDR_WG_DRV_CTRL_REG	<7:0>	RD/WR	仅bit0有效，控制是否使能波形发生时钟，即控制波形的产生和停止。
ADDR_WG_DRV_REST_T_REG	<7:0>	WR	死区时间时长设置（单位us，范围0-255），死区时间是图3.10的delay_time，如果需要使用到255us以上的时间，需要借用波形脉宽时间。
ADDR_WG_DRV_SILENT_T_REG	<31:0>	WR	静默时间时长设置（单位us，范围0-10秒+）
ADDR_WG_DRV_HLF_WAVE_PRD_REG	<31:0>	WR	正半周期的脉宽时间（单位us，范围2-99999us）
ADDR_WG_DRV_NEG_HLF_WAVE_PRD_REG	<31:0>	WR	负半周期的脉宽时间（单位us，范围2-99999us）
ADDR_WG_DRV_CLK_FREQ_REG	<7:0>	WR	波形发生器的时钟频率（单位 MHz）
ADDR_WG_DRV_IN_WAVE_ADDR_REG	<7:0>	WR	波形生成的方法是描点法，正半周期的波形由64个点的电流幅值组成，这个寄存器赋值范围为（0-63），代表64个地址。
ADDR_WG_DRV_IN_WAVE_REG	<7:0> <15:8>	WR	<7:0> 给上面对应的64个地址赋值，设置电流大小，每个地址都需要设置一个对应的电流大小，范围0-255。 <15:8> 在driverB的应用中，选择与公共电极配对的电极（范围0-23）。
ADDR_WG_DRV_DELAY_LIM_REG	<15:0>	WR	延迟输出时钟数，可以实现多通道错位输出，1us的时钟数公式： $\text{时钟频率}/1000000$ （us）。 例：延迟X us输出，此寄存器设置为 $\text{时钟频率}/1000000 * X$ （us）。
ADDR_WG_DRV_NEG_SCALE_REG	<7:0>	WR	倍乘系数： 负半周期波形的幅值 = ADDR_WG_DRV_NEG_SCALE_REG * 正半周期幅值 如果幅值大于255，结果会溢出，从0开始。
ADDR_WG_DRV_NEG_OFFSET_REG	<7:0>	WR	偏移量参数： 负半周期波形的幅值 = ADDR_WG_DRV_NEG_OFFSET_REG + 正半周期幅值， 如果幅值大于255，结果会溢出，幅值从0开始计算。 最终负半周期的电流幅值 = ADDR_WG_DRV_NEG_SCALE_REG * 正半周期幅值 + ADDR_WG_DRV_NEG_OFFSET_REG。
ADDR_WG_DRV_INT_REG	<31:0>	RD/WR	说明：一次可以设置两个中断地址，分别叫做第一地址，第二地址，波形每半个周期由64个点组成，每个点都有对应的地址（0-63），两个中断地址的实际设置值为64个波形地址中的2个。  写入访问： bit0：启用中断。 Bit1：启用时清除第一个地址中断。 Bit2：启用时清除第二个地址中断。

			<p>Bit&lt;15:8&gt;: 第一个中断地址。当波形发生器到达该地址时, 启用 APB 中断信号 (有64个点波形, 因此, 选择64个地址中的某一个地址用作第一个地址中断)。</p> <p>Bit&lt;23:16&gt;: 第二个中断地址。当波形发生器到达该地址时, 启用 APB 中断信号 (有64个点波形, 因此, 选择64个地址中的某一个地址用作第二个地址中断, 注意: 第二中断地址应该大于第一中断地址)。</p> <p>剩余位保留。</p> <p>读取访问:</p> <p>Bit&lt;7:0&gt;: 正在读取的波形发生器编号。</p> <p>Bit8: 中断已启用。</p> <p>Bit9: 发生第一次地址中断。当波形发生器到达第一个波形地址时为1。</p> <p>Bit10: 发生第二次地址中断。当波形发生器到达第二个波形地址时为1。</p> <p>Bit&lt;23:16&gt;: 报告中断的第一个地址。</p> <p>Bit&lt;32:24&gt;: 报告中断的第二个地址。</p>
ADDR_WG_DRV_ISEL_REG	<2:0>	WR	单元电流系数设置, 范围 (0-7), 实际单元电流 = (ADDR_WG_DRV_ISEL_REG + 1) * 33uA
ADDR_WG_DRV_SW_CONFIG_REG	<7:0>	WR	在driverC的应用中, 选择与某个有源电极配对的switch开关电极, 每一位代表一个电极, 如0x1表示使用STIMU16电极, 0x2表示使用STIMU17电极, 0x80表示使用STIMU23电极, 0xff表示所有从STIMU16到STIMU23这8个开关电极都使用。

### 3.3.4 ENS001的波形发生例程

- 波形发生例程的路径为: \NNC\_ENS1QFN96L1010A--全功能ENS开发资料-V1.0\5、例程\1、例程, 分别有DRIVER-A部分, DRIVER-B部分, DRIVER-C部分

DRIVER-A部分一共提供了5个示例程序, 分别是boost电压选择例程、方波例程、正弦波例程、EMS波形、干扰波例程。

以《2、ENS\_PULSE\_TEST》为例, 主函数设置步骤为:

- 1) 设置MTP时钟相关的参数, CMSDK\_MTPREG->MTP\_CR = 0x00000003;
- 2) 设置主频为32MHz;
- 3) 使能APB总线上的部分时钟如 UART1、波形发生器的时钟;
- 4) 选择boost参数;
- 5) 串口初始化;
- 6) 给多路波形发生器的寄存器赋值 (参考表3.1), 程序中四个通道对应的波形发生器硬件地址别名分别为WAVE\_GEN\_DRVA\_BLK0、WAVE\_GEN\_DRVA\_BLK1、WAVE\_GEN\_DRVA\_BLK2、WAVE\_GEN\_DRVA\_BLK3;

```

CMSDK_WAVEGEN_DRVA->WAVE_GEN_DRV_REST_T_REG = 0x00000064; //REST_T 死区时间 (0-255us)
CMSDK_WAVEGEN_DRVA->WAVE_GEN_DRV_SILENT_T_REG = 0x000003E8; //SILENT_T 静默时间
CMSDK_WAVEGEN_DRVA->WAVE_GEN_DRV_CLK_FREQ_REG = 0x00000020; //波形发生器时钟频率
CMSDK_WAVEGEN_DRVA->WAVE_GEN_DRV_HLF_WAVE_PRD_REG = 0x000000C8; //正半周期脉宽 (2us-100000us)
CMSDK_WAVEGEN_DRVA->WAVE_GEN_DRV_NEG_HLF_WAVE_PRD_REG = 0x000000C8; //负半周期脉宽 (2us-100000us)
CMSDK_WAVEGEN_DRVA->WAVE_GEN_DRV_CONFIG_REG = 0x0000004F; //bit0:死区时间使能, 当不使能时, REST_T必须等于0
//bit1:负半周期波形发生使能, 当使能后, 负半周期的波形才会发生
//bit2:静默时间使能, 当使能后, 设置的静默时间才生效
//bit3:使能第二个电流源 (driverA/C的应用中每一对电极都有2个电流源)
//bit4:交替模式使能
//bit5:连续模式
//bit6:多电极使能, 当需要使用2/3/4通道时需要使能
CMSDK_WAVEGEN_DRVA ->WAVE_GEN_DRV_ISEL_REG = 0x03; //范围 0x00 - 0x07, 这个寄存器设置单元电流
//单元电流 = (此寄存器值+1) * 33ua (范围: 33-264ua)

for(int i=0; i<64; i++){
    CMSDK_WAVEGEN_DRVA->WAVE_GEN_DRV_IN_WAVE_ADDR_REG = i;
    CMSDK_WAVEGEN_DRVA->WAVE_GEN_DRV_IN_WAVE_REG = 0xff; //这里设置正半周期的64个点中每个点的电流大小
//64个点各有一个地址, 这里给每一个点都设置了一样的电流大小, 因此是方波 (一条直线)
//负半周期的幅值由正半周期拷贝过去。
}

//延时生成波形设置, 当driverA的四个通道的延时时间 (WAVE_GEN_DRV_DELAY_LIM_REG) 设置不同时, 四个通道的输出便不同步

```

图3.13 波形参数赋值

7) 启动刺激输出（通道一）：

AVE\_GEN\_DRVA\_BLK0->WAVE\_GEN\_DRV\_CTRL\_REG = 0x00000001。

■ DRIVER-C 部分：

Driver-C的例程实现了有源电极和SWITCH开关电极的组合，并且实现了在一段时间的输出之后，切换电极组合并继续输出。

Driver-C测试程序的负载连接方法如图所示，例程中，STIMU15开始与STIMU17组合为一个通道，然后开始输出双向刺激电流，经过5秒之后再与STIMU23组合为一个通道。经过5秒之后再与STIMU15 / 17 / 23 三个通道组合。

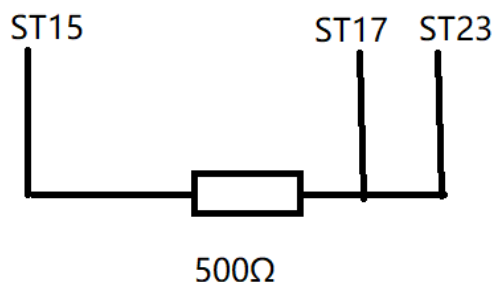


图3.14 DRIVER-C例程对应的负载连接方法

■ DRIVER-B 部分：

DRIVER-B例程是选择了STIMU0与STIMU23组合使用，测试时在STIMU0与STIMU23之间接500Ω电阻。

### 3.4 功耗测试说明

使用ENS1EVKB全功能开发板，去除R1、R2、R7电阻，断开J14跳帽，使用Micro USB线供电，将数字万用表串联到J13两端。修改主频等参数，记录如下功耗数据。

主频	APB时钟	输出通道	BOOST电压设置V	功耗 (mA)
内部低速时钟 32khz	×	×	×	1.15
内部高速时钟4mhz	×	×	×	1.98
内部高速时钟8mhz	×	×	×	2.8
内部高速时钟16mhz	×	×	×	4.5
内部高速时钟32mhz	×	×	×	8.01
内部高速时钟4mhz	×	×	11	2.23
内部高速时钟4mhz	×	×	15	2.23
内部高速时钟4mhz	×	×	26	2.23
内部高速时钟4mhz	×	×	45	2.23
内部高速时钟4mhz	WAVE_GEN_PCLK ANALOG_PCLK 其余关闭	1	45	2.605
内部高速时钟4mhz	WAVE_GEN_PCLK ANALOG_PCLK 其余关闭	2	45	2.85
内部高速时钟4mhz	WAVE_GEN_PCLK ANALOG_PCLK UART1_PCLK_EN 其余关闭	2	45	2.9



## 4. 联系方式

可通过以下方式了解更多产品详情：

1) 公司电话：4008605922； 180 9470 6680

2) 技术人员QQ：1708154204



3) 公众号：暖芯迦电子



Copyright© 2023by Hangzhou Nanochap Electronics Co.,Ltd.

用指南中所出现的信息在出版当时相信是正确的，然而暖芯迦对于说明书的使用不负任何责任。文中提到的应用目的仅仅是用来做说明，暖芯迦不保证或表示这些没有进一步修改的应用将是适当的，也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。暖芯迦产品不授权使用于救生、维生从机或系统中做为关键从机。暖芯迦拥有不事先通知而修改产品的权利，对于最新的信息，请参考我们的网址<https://www.nanochap.cn>或与我们直接联系（4008605922）。

## X-ON Electronics

Largest Supplier of Electrical and Electronic Components

*Click to view similar products for [Development Boards & Kits - ARM category](#):*

*Click to view products by [Nanochap manufacturer](#):*

Other Similar products are found below :

[CY4541](#) [OM13090UL](#) [Raspberry Pi 4 Model B,8GB](#) [YR0K77210B000BE](#) [B-U585I-IOT02A](#) [NUCLEO-C031C6](#) [NUCLEO-U5A5ZJ-Q](#)  
[NUCLEO-WL55JC1](#) [STM32MP135F-DK](#) [ZDSD-Pinboard](#) [081ZYKFB](#) [LKS32MC034DOF6Q8-k](#) [LKS32MC077MBS8-K](#)  
[LKS32MC038Y6P8B-K](#) [LKS32MC071DOC8T8-K](#) [LKS32MC074DOF8Q8-K](#) [LKS32MC038Y6P8-k](#) [Ai-WB2-32S-Kit](#) [GD32E103T-START](#)  
[GD32L233K-START](#) [RTK7F124FPC01000BJ](#) [XDS601](#) [RP2040-Tiny](#) [M6G2C-256LI](#) [YT37](#) [LKS32MC033H6P8B-K](#) [VC-02-Kit\\_EN](#) [Ra-](#)  
[08H-Kit](#) [Hi-12FL-Kit](#) [PB-03M-Kit](#) [Ai-WB2-13-Kit](#) [PB-03F-Kit](#) [Hi-07SL-Kit](#) [Hi-07S-Kit](#) [Ai-WB2-12F-Kit](#) [PB-03-Kit](#) [AT-START-F437](#)  
[AT-START-F407](#) [E104-BT40-TB](#) [FT8132Q-3HALL-FOC+EMF](#) [FU6832L-TGB-DEMO](#) [APM32F072VBT6](#) [APM32F091VC MINI](#)  
[APM32F407IG-MINIBOARD](#) [APM32F003F6P6-MINIBOARD](#) [APM32F051R8 MINI](#) [GD32EPRTV-START](#) [GD32FPRT-START](#)  
[GD32407H-START-1](#) [GD32E503V-EVAL](#)