



Audio Workshop V1.0 使用手册

版本：V1.20 日期：2022-08-10

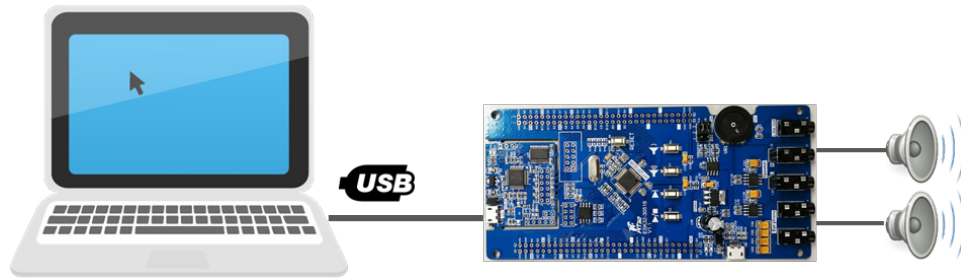
www.holtek.com

目录

一、开发环境	3
1.1 整体环境	3
1.2 软件	3
1.3 硬件	3
二、Audio Workshop 主界面	6
2.1 软件功能介绍	6
三、新建专案	7
3.1 新建专案	7
3.2 MIDI 配置	8
3.3 语音配置	10
3.4 音效配置	11
3.5 语句配置	12
3.6 基本设定	15
3.7 产出程序与 Audio 数据	16
3.8 下载功能	17
3.9 开启专案	17
四、音色编辑功能	18
4.1 新建专案	18
4.2 选择 midiOut 设备	20
4.3 音色参数调整	20
4.4 打击乐参数调整	24
4.5 音色替换	25
4.6 开启专案	29
五、函式库说明	30
5.1 播放函数	30
5.2 音效函数	31
5.3 系统函数	33
5.4 其它说明	35
六、附录	36
6.1 程序流程	36
6.2 e-link32 Pro 连接方式	40
6.3 Data Flash ROM 烧录	41
6.4 开发板原理图	44

一、开发环境

1.1 整体环境



I. Audio Workshop

开发板

喇叭

II. Keil™ MDK-ARM

1.2 软件

包括 Audio Workshop、Keil™ MDK-ARM μ Vision5

1.2.1 Audio Workshop 界面

- 选择 MCU、外挂 SPI Flash 容量选择、加载与编排 MIDI 曲 / 语音 WAV / 音效 WAV、语句编排 (由语音 / 音效组合)、音频输出配置、MIDI 接口配置等功能。
- MCU 程序产出、SPI Flash 音色数据生成与数据下载功能 (MCU/SPI Flash)。
- 音色编辑功能：包含音色库 (SF2) 加载、编辑音色参数与替换音色功能。

1.2.2 Keil™ MDK-ARM μ Vision5

- 用于编辑与查看源程序，可直接下载程序到开发板中。
- Audio Workshop 需搭配 Keil™ MDK-ARM μ Vision5 使用。使用 Audio Workshop 前请先确认已安装 Keil™ MDK-ARM μ Vision5 与 Keil™ HT32 支持的封装。

1.3 硬件

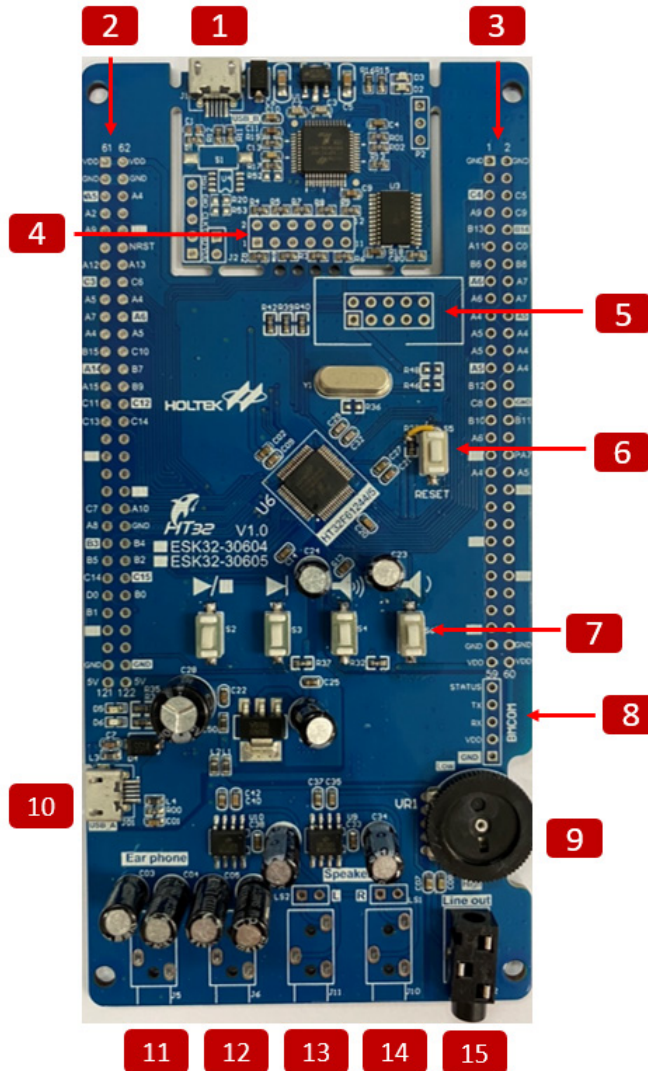
开发板：搭配 Audio Workshop 开发使用。

注意：HT32F0006 专案利用 ESK32-30617 开发板进行仿真开发，实际 HT32F0006 应用开发请洽本公司技术部门。

- ESK32-30615：HT32F61355 的开发板
 - ◆ MCU 含 32Mbits Data Flash ROM 供储存数据使用。
- ESK32-30616：HT32F61356 的开发板
 - ◆ MCU 含 64Mbits Data Flash ROM 供储存数据使用。
- ESK32-30617：HT32F61357/HT32F0006 的开发板
 - ◆ MCU 含 128Mbits Data Flash ROM 供储存数据使用。
- ESK32-30605：HT32F61244/HT32F61245 的开发板
 - ◆ MCU 含 16/32Mbits Data Flash ROM 供储存数据使用。

1.3.1 开发板介绍

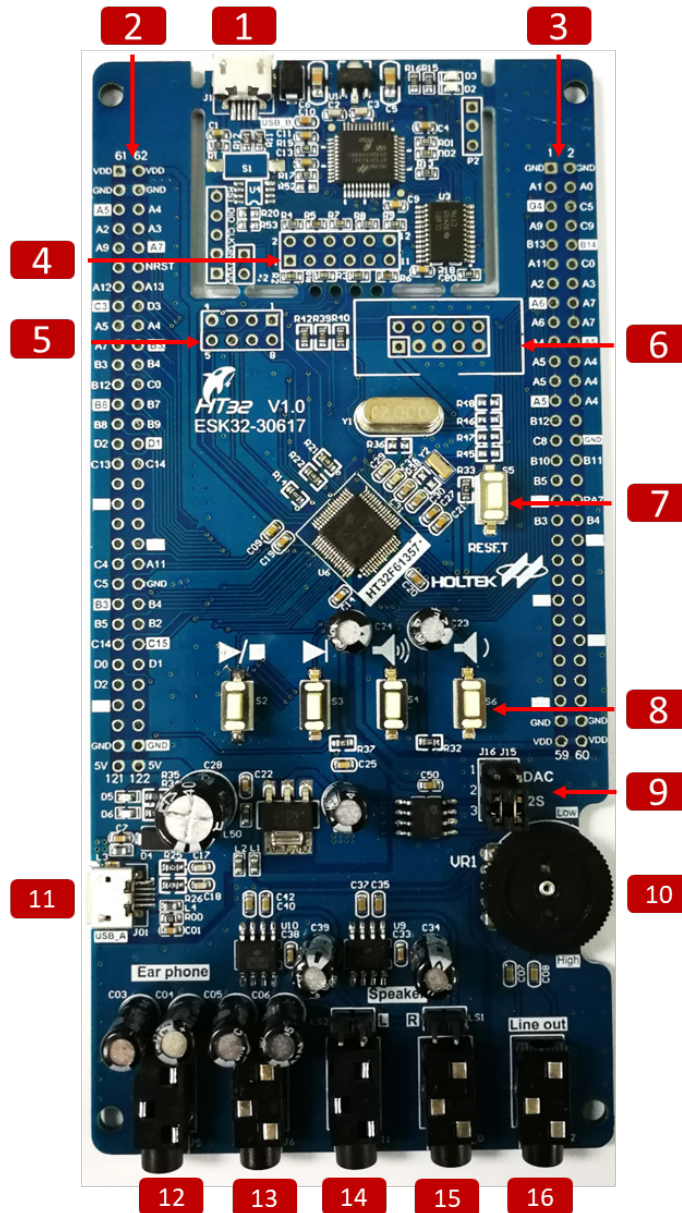
ESK32-30605



1. USB_B: e-Link32 Lite (下载程序 / 数据, 仿真)。
2. GPIO / 扩展连接器。
3. GPIO / 扩展连接器。
4. 下载连接器。
5. SWD-10P 连接器。
6. Reset 按键。
7. 功能按键:
 - S2: 播放 / 停止。
 - S3: 下一首。
 - S4: 音量大。
 - S6: 音量小。

8. UART 连接器。
9. 音量调整：顺时针大 / 逆时针小。
10. USB_A: HT32F6124x USB 供电。
11. 耳机接口。
12. 耳机接口。
13. 左声道喇叭接口。
14. 右声道喇叭接口。
15. Line out 接口。

ESK32-30617 (与 ESK32-30615/ESK32-30616 配置相同)



1. USB_B: e-Link32 Lite (下载程序 / 数据, 仿真)。
2. GPIO / 扩展连接器。
3. GPIO / 扩展连接器。
4. 下载连接器。
5. SPI Flash 连接器。
6. SWD-10P 连接器。
7. Reset 按键。
8. 功能按键:
 - S2: 播放 / 停止。
 - S3: 下一首。
 - S4: 音量大。
 - S6: 音量小。
9. DAC/I2S DAC 跳线。
10. 音量调整: 顺时针大 / 逆时针小。
11. USB_A: HT32F6135x USB 接口。
12. 耳机接口。
13. 耳机接口。
14. 左声道喇叭接口。
15. 右声道喇叭接口。
16. Line out 接口。

二、Audio Workshop 主界面

2.1 软件功能介绍



1. 新建专案 / 开启专案 / 储存专案。
2. 基本设定: 设定 MIDI 相关功能参数。
3. 编辑功能 (MIDI / 语音 / 音效 / 语句)。
4. 生成 Audio 数据 (.bin 档)。
5. 产出程序文件 (.hex)。
6. 下载功能:
 - 提供下载专案程序文件 (.hex) 与 Audio 数据 (.bin)。
7. SPI Flash 容量选择:
 - 显示 SPI Flash 的容量和使用率。
8. 开启音色调整功能。
9. 快速开启外部软件选单: 。
 - WAV 编辑软件: 预设开启 Audacity®。

MIDI 编辑软件：预设开启 Cakewalk by BandLab®。

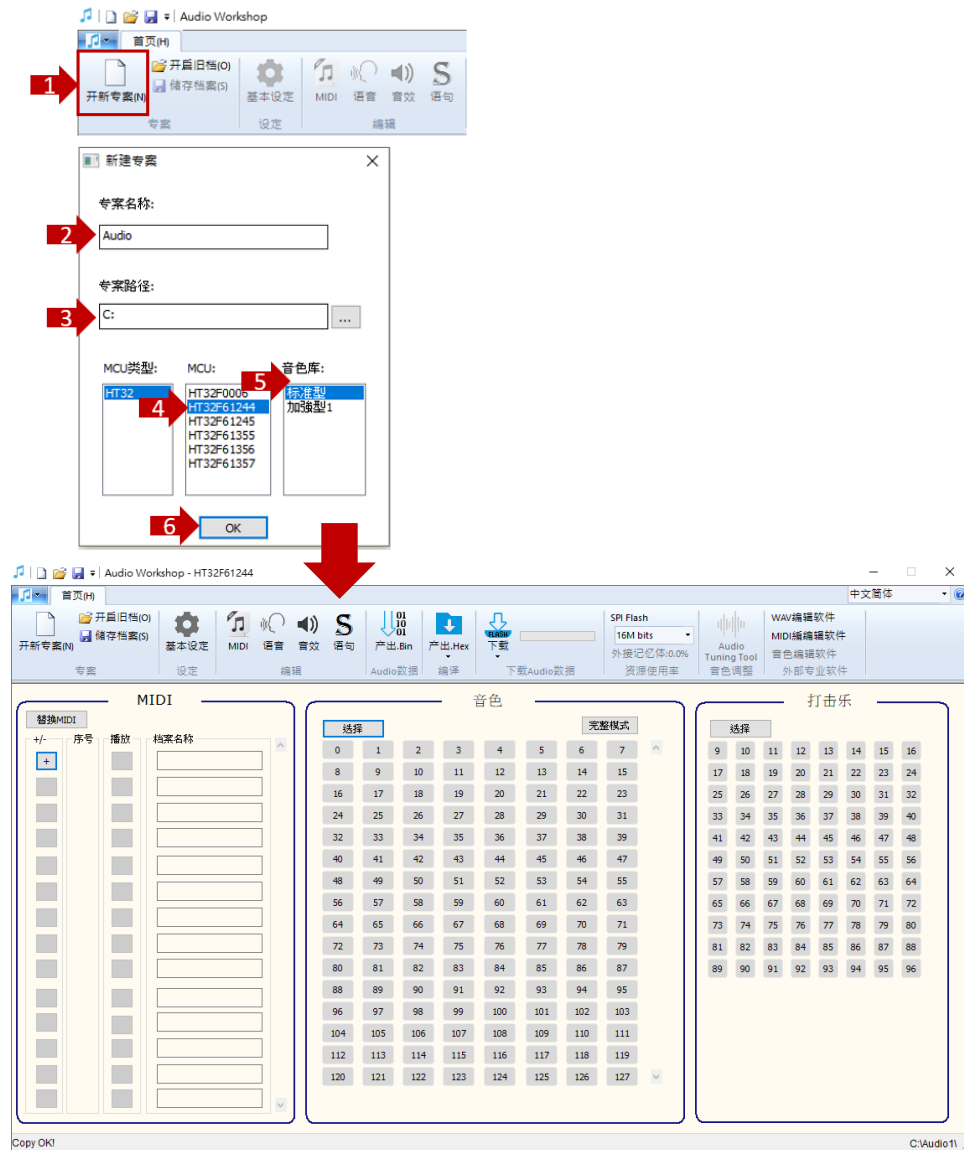
音色编辑软件：预设开启 Polyphone®。

10. 界面语系切换：提供繁体 / 简体 / 英文。

11. “关于”：显示软件版本信息。

三、新建专案

3.1 新建专案

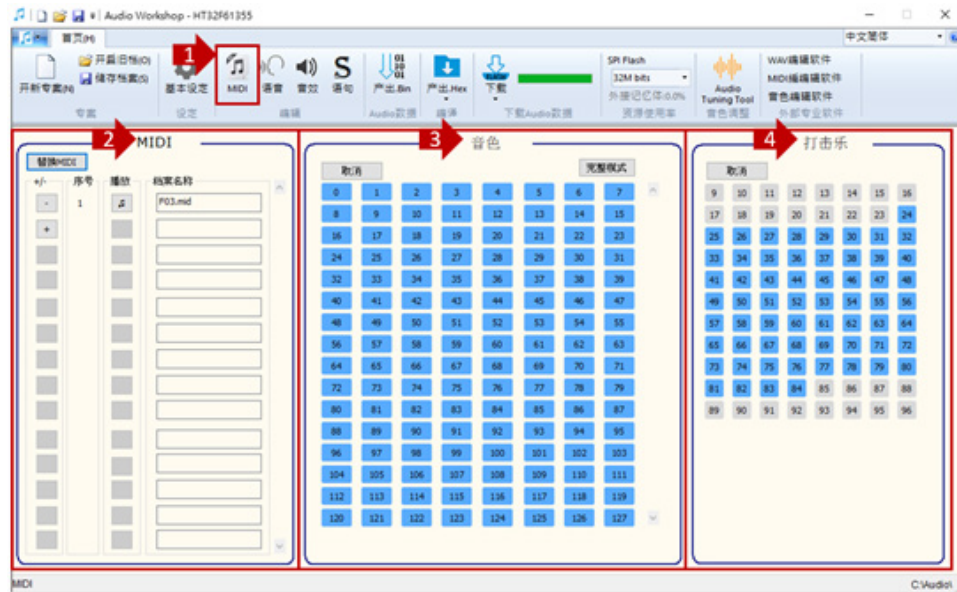


- 步骤 1：点击“开新专案”。
- 步骤 2：设定专案名称。
- 步骤 3：设定专案路径。
- 步骤 4：选择 MCU 型号。

- 步骤 5: 选择音色库。
- 步骤 6: 点击“OK”，进入 Audio 编辑主界面。

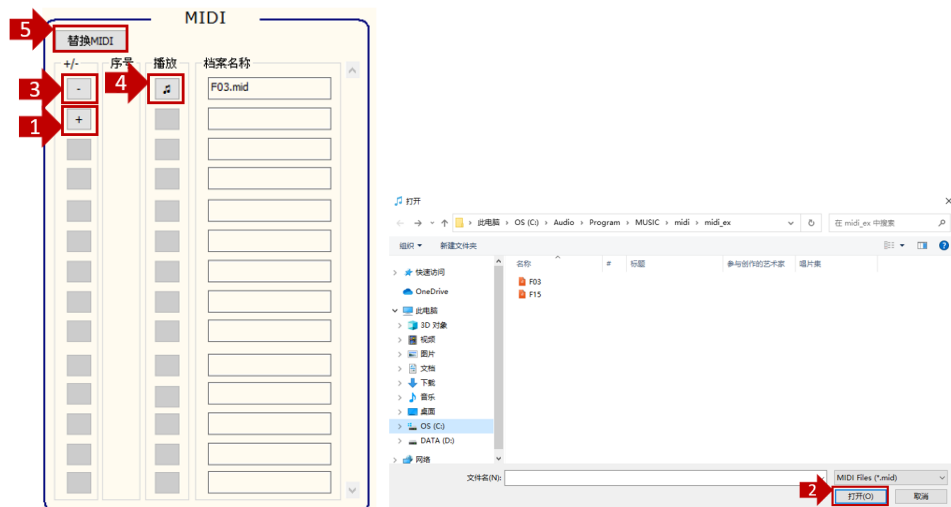
3.2 MIDI 配置

3.2.1 操作流程



- 步骤 1: 点击“MIDI”图示。
- 步骤 2: 加载 / 编排 MIDI。
- 步骤 3: 设定音色。
- 步骤 4: 设定打击乐。

3.2.2 功能介绍



- 步骤 1: 点击图示，可加载 MIDI。
- 步骤 2: 选择 MIDI 后，点击“开启”，MIDI 就会被加载至专案中。

- 步骤 3: 点击图示, 可删除 MIDI。
- 步骤 4: 点击可通过计算机上播放 MIDI。
- 步骤 5: 点击“替换 MIDI”, 可替换专案内的 MIDI 档案。



- 步骤 1: 点击“选择”, 可将音色 (0~127) 全选, 如上图右。
- 步骤 2: 点击“取消”, 可取消所有已经选择的音色, 如上图左。



- 步骤 3: 点击音色号, 可选择与取消。
此例: 选择 16/17/18/19/20 号音色。
- 步骤 4: 点击“完整模式”, 可显示出音色对应完整乐器名称, 如上图右。
- 步骤 5: 可选择乐器种类, 分类显示。
- 步骤 6: 可切换成“简易模式”界面, 如上图左。



- 步骤 1: 点击“选择”，可将打击乐 (24~84) 全选。
- 步骤 2: 点击“取消”，可取消所有已经选择的打击乐。
- 步骤 3: 点击打击乐号，可选择与取消。
此例：选择 29/30/31/32 号打击乐。

3.3 语音配置

3.3.1 功能介绍

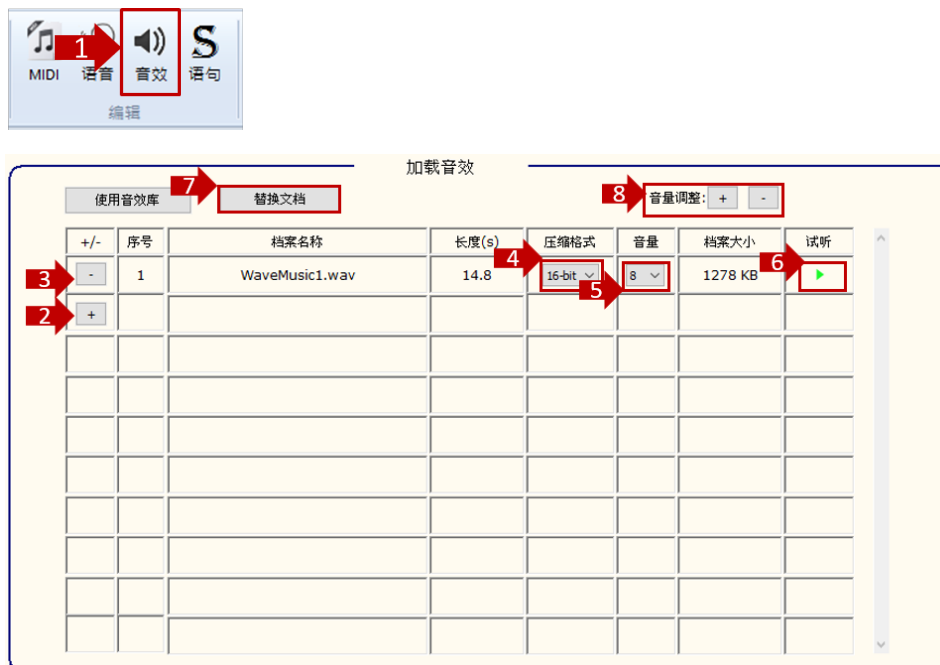


- 步骤 1: 点击“语音”图标。
- 步骤 2: 点击图示，可加载 WAV 到专案中。
- 步骤 3: 点击图示，可删除 WAV。
- 步骤 4: 设定压缩格式 (PCM 8-bit/12-bit/16-bit)。
- 步骤 5: 设定音量 (1~15)。
- 步骤 6: 点击可通过计算机播放语音 WAV。

- 步骤 7: 点击“替换文档”，可替换专案内的 WAV 档案。
- 步骤 8: 点击可调整所有 WAV 档案音量。

3.4 音效配置

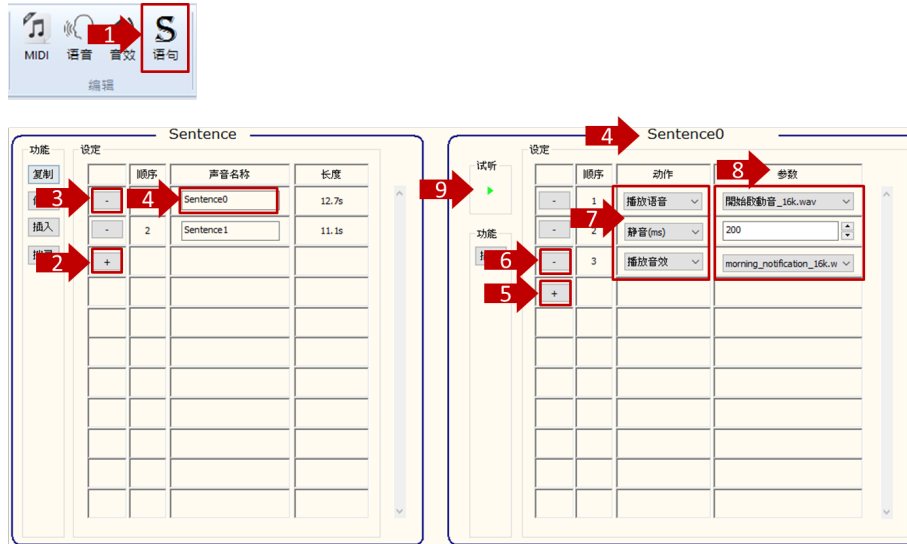
3.4.1 功能介绍



- 步骤 1: 点击“音效”图标。
- 步骤 2: 点击图示，可加载 WAV 到专案中。
- 步骤 3: 点击图示，可删除 WAV。
- 步骤 4: 设定压缩格式 (PCM 8-bit/12-bit/16-bit)。
- 步骤 5: 设定音量 (1~15)。
- 步骤 6: 点击可通过计算机播放音效 WAV。
- 步骤 7: 点击“替换文档”，可替换专案内的 WAV 档案。
- 步骤 8: 点击可调整所有 WAV 档案音量。

3.5 语句配置

3.5.1 功能介绍 1

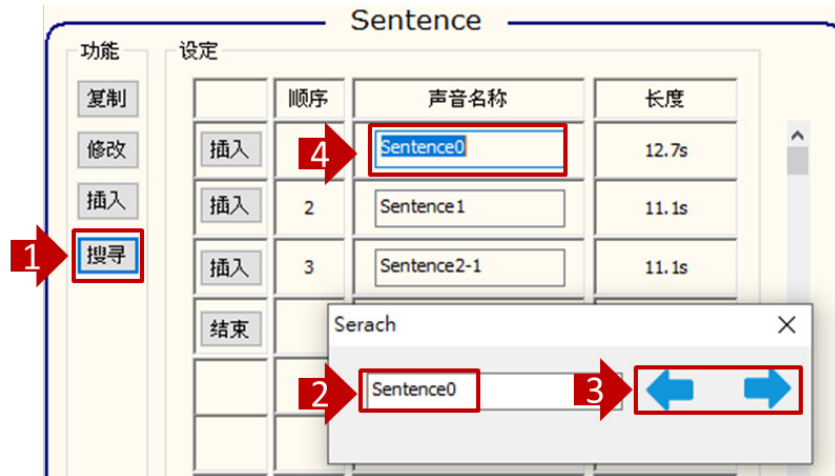


- 步骤 1: 点击“语句”图示。
- 步骤 2: 点击图示，可新增语句。
- 步骤 3: 点击图示，可删除语句。
- 步骤 4: 选择语句 sentence0，可编辑 sentence0 播放内容。
- 步骤 5: 新增动作。
- 步骤 6: 删除动作。
- 步骤 7: 编辑动作，可选择“播放语音 / 播放音效 / 静音”。
- 步骤 8: 设定语音，音效声音或静音时间。
- 步骤 9: 点击可通过计算机播放语句声音。

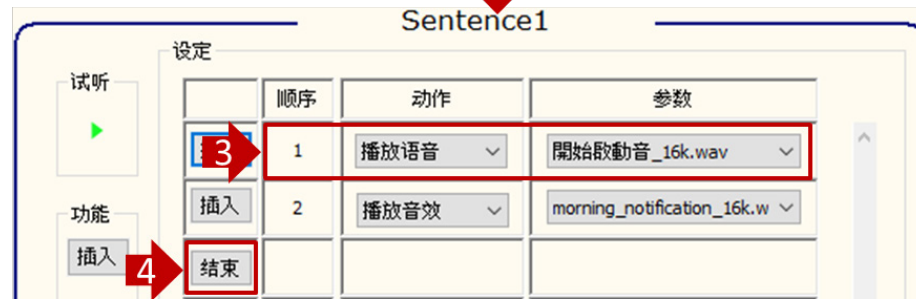
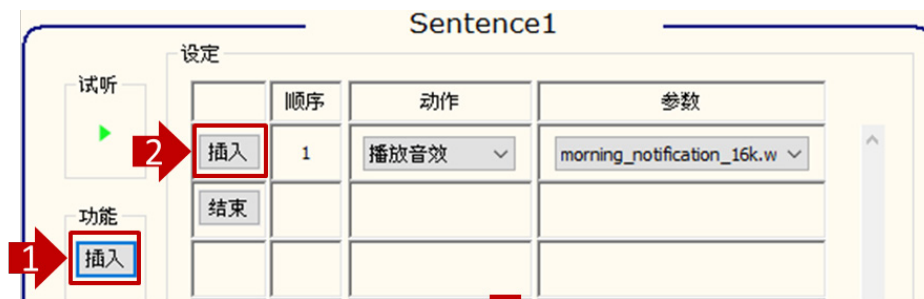
3.5.2 功能介绍 2



- 步骤 1: 复制语句功能。
- 步骤 2: 点击图示，可复制该语句 (Sentence1)。
- 步骤 3: 自动插入复制的语句 (Sentence1(1))。
- 步骤 4: 点击“结束”，完成编辑。



- 步骤 1: 搜寻语句功能。
- 步骤 2: 输入语句名称。
- 步骤 3: 点击即可搜寻语句 (Sentence0)。
- 步骤 4: 搜寻到语句 Sentence0。

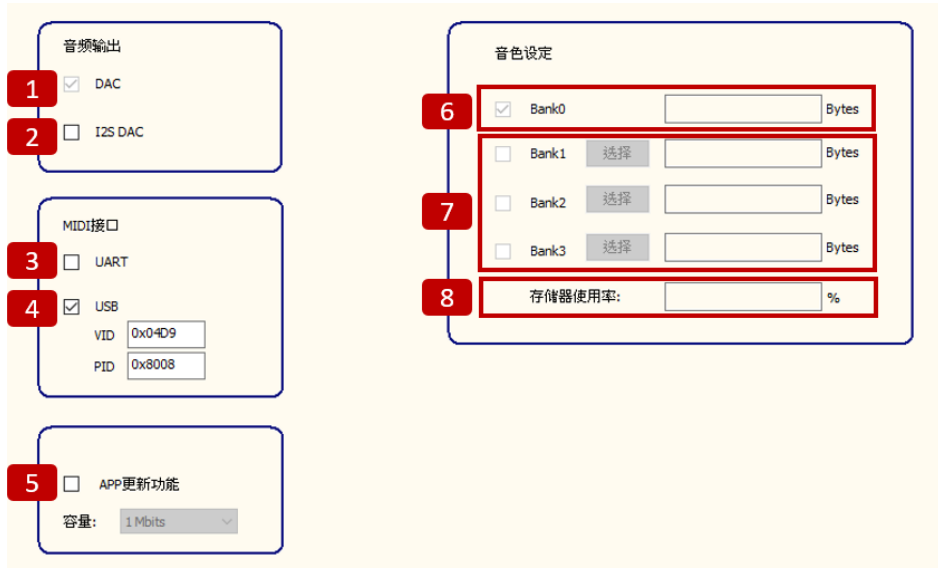


- 步骤 1: 插入语句动作功能。
- 步骤 2: 点击图示，可插入语句动作。
- 步骤 3: 自动插入一播放语音动作。
- 步骤 4: 点击“结束”，完成编辑。

3.6 基本设定

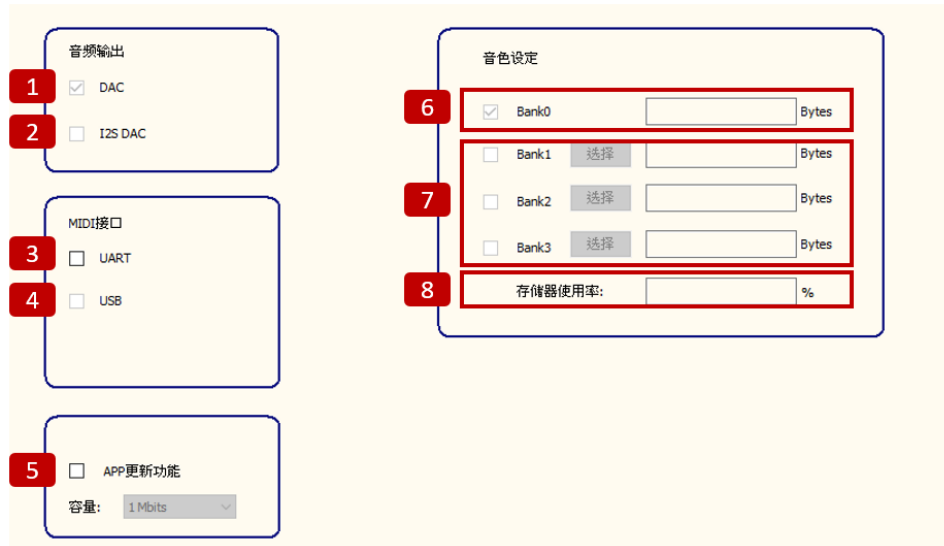


专案 MCU 型号为 HT32F0006/HT32F61355/HT32F61356/HT32F61357 时，基本设定页面如下：



1. DAC：默认功能。
2. I2S DAC 勾选可开启 I2S DAC 功能；
搭配开发板 Jumper 切换至 I2S，即可通过 I2S DAC 播放。
3. UART：勾选可通过 UART 接收 MIDI 信息，取消勾选则关闭功能。
4. USB：勾选可通过 USB 接收 MIDI 信息，取消勾选则关闭功能。
提供 VID 与 PID 可供用户修改。
5. APP 更新功能：勾选可开启功能，取消勾选则关闭功能。
容量：设置 APP 可更新用户的 MIDI 曲容量的大小。
6. 音色设定：Bank0 音色为默认设定。
7. Bank1-3：勾选可扩增音色数据，由程序档内 `__R_Bank` 数值决定使用的音色来源。
注：需先产出 Audio 数据 (.bin) 才可开启此功能，详细产出 Audio 数据步骤可参照 3.7 小节。
8. 存储器使用率：显示已使用多少 Data Flash ROM 空间。

专案 MCU 型号为 HT32F61244 / HT32F61245 时，基本设定页面如下：



1. DAC：默认功能。
2. I2S DAC：默认不勾选。
3. UART：勾选可通过 UART 接收 MIDI 信息，取消勾选则关闭功能。
4. USB：默认不勾选。
5. APP 更新功能：勾选可开启功能，取消勾选则关闭功能。
容量：设置 APP 可更新用户的 MIDI 曲容量的大小。
6. 音色设定：Bank0 音色为默认设定。
7. Bank1-3：勾选可扩增音色数据，由程序档内 __R_Bank 数值决定使用的音色来源。
注：需先产出 Audio 数据 (.bin) 才可开启此功能，详细产出 Audio 数据步骤可参照 3.7 小节。
8. 存储器使用率：显示已使用多少 Data Flash ROM 空间。

3.7 产出程序与 Audio 数据



- 步骤 1：当“基本设定 / MIDI / 语音 / 音效 / 语句”已配置完成。
- 步骤 2：点击图标，产出 Audio 数据 (.bin)。
- 步骤 3：点击图标，编译产出程序文件 (.hex) 与 Keil 专案工程。

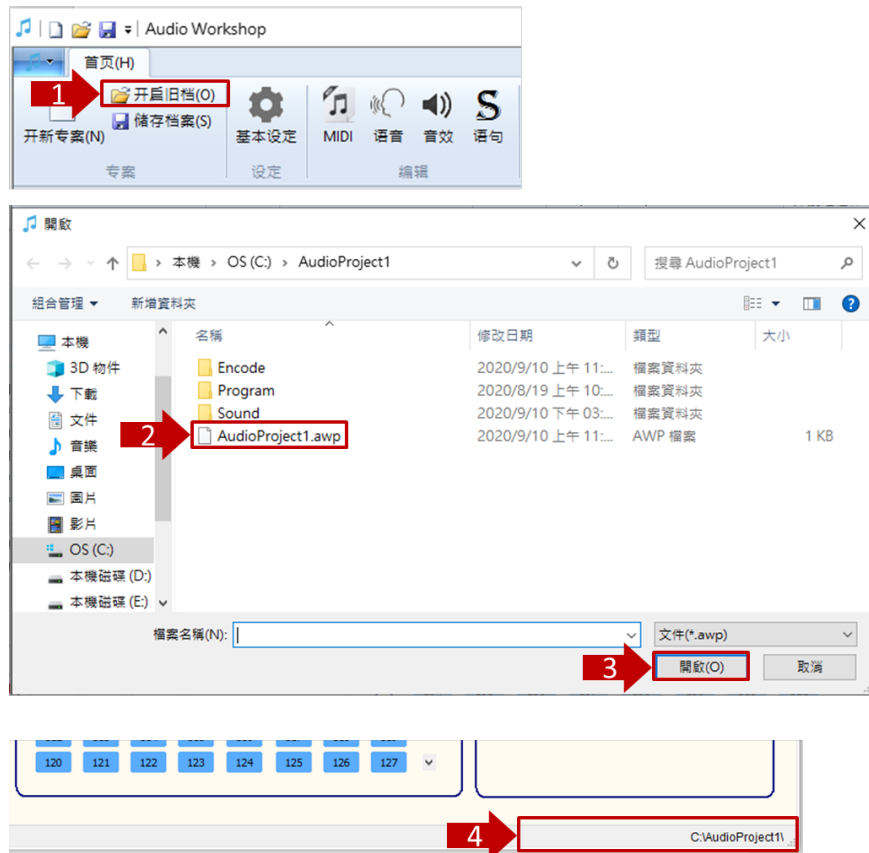
3.8 下载功能



- 步骤 1: 点击图示。
- 步骤 2: 选择下载动作。
 - 1: 下载专案的 bin 档。
 - 2: 下载专案的 hex 档。
 - 3: 选择 bin 档并下载。
 - 4: 选择 hex 档并下载。
 - 5: 使用 e-link32 Pro 下载专案的 bin 档。
 - 6: 使用 e-link32 Pro 下载专案的 hex 档。
 注意: e-link32 Pro 连接方式参照附录 6.2 小节。
- 步骤 3: 下载进度至 100%, 即下载完成。

3.9 开启专案

用户可通过“开启旧档”，开启 Audio 专案或切换 Audio 专案。



- 步骤 1: 点击“开启旧档”。
- 步骤 2: 选择欲开启的专案“xxxx.awp”(会自动默认上一次储存的专案路径)。此例为 AudioProject1.awp。
- 步骤 3: 点击“开启”，进入操作界面，即完成开启专案。
- 步骤 4: 开启专案后，专案的路径会显示于下方信息列。

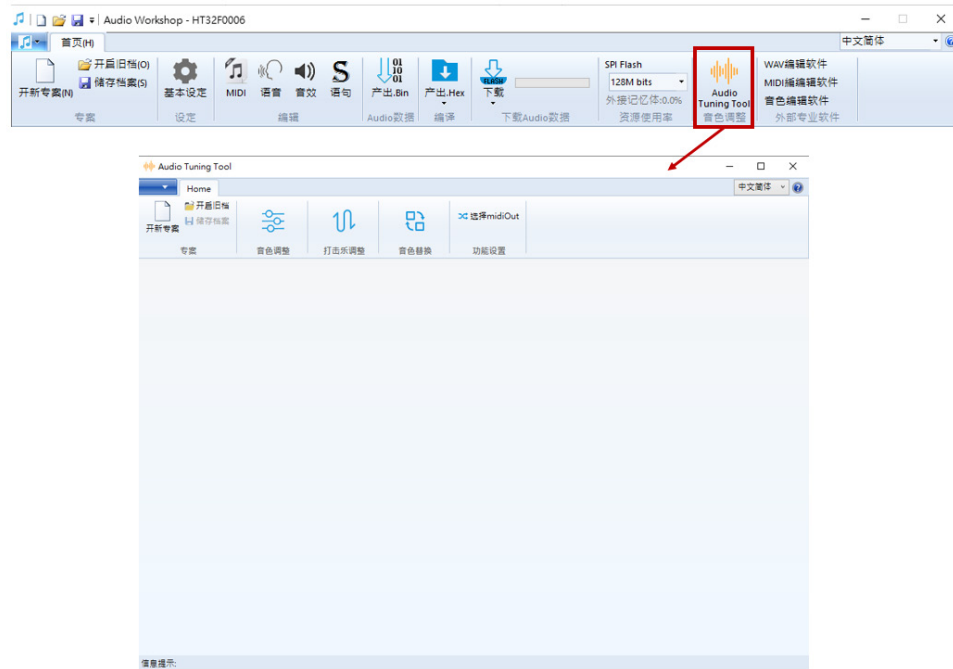
四、音色编辑功能

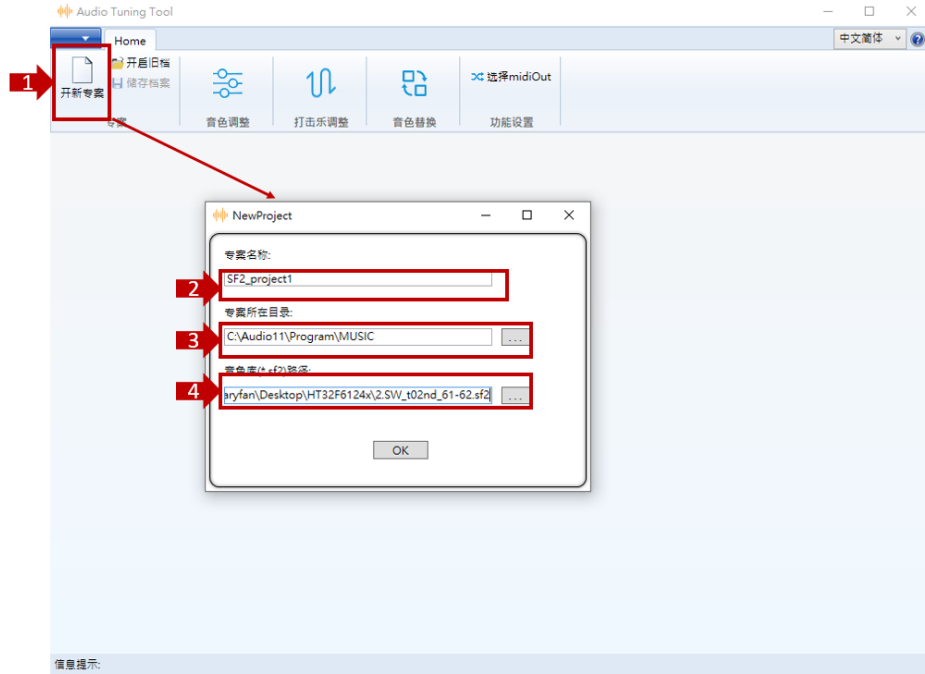
对于 HT32F61355/HT32F61356/HT32F61357 以及 HT32F0006，Audio Workshop 提供音色编辑功能，通过此功能可修改音色参数或替换为自定义音色。

一个 Audio Workshop 专案下可建立多个音色专案；以下介绍使用步骤。

4.1 新建专案

Audio Workshop 主界面上点击“Audio Tuning Tool”图示，即可开启音色编辑功能界面，如下图：





- 步骤 1: 新建音色专案。
- 步骤 2: 设定音色专案名称。
- 步骤 3: 设定音色专案路径, 使用默认路径即可。
- 步骤 4: 设定加载的音色库 (*.sf2) 路径。

如下图, 已完成加载音色库, 进入到音色编辑界面。



4.2 选择 midiOut 设备



- 步骤 1：选择 midiOut 设备。
- 步骤 2：选择“USB DEVICE”；
需先确认 PC USB 连接至开发板 USB_A 接口。
- 步骤 3：设定完成，显示 MidiOut USB DEVICE Connect，点选“确定”。

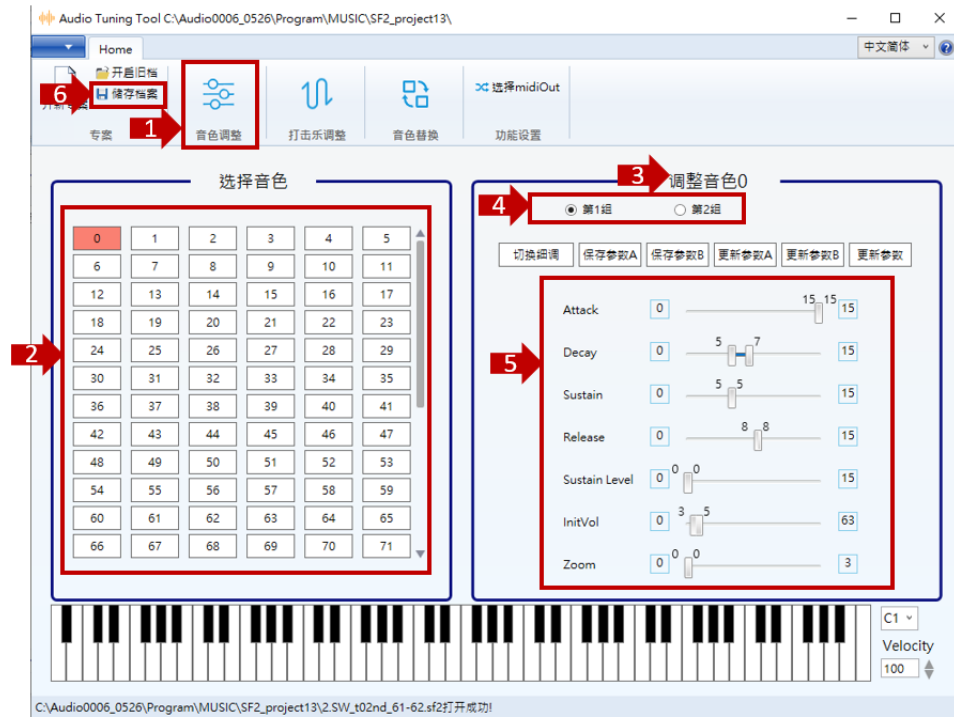


4.3 音色参数调整

提供音色参数“粗调”与“细调”模式，并且无须进行烧录，即可将正在调整的音色参数传送至开发板，实时听到调整后的效果。

- 音色粗调：对音色内所有的音色样本的参数一起调整。
- 音色细调：可对音色内的个别音色样本调整参数。

4.3.1 音色粗调



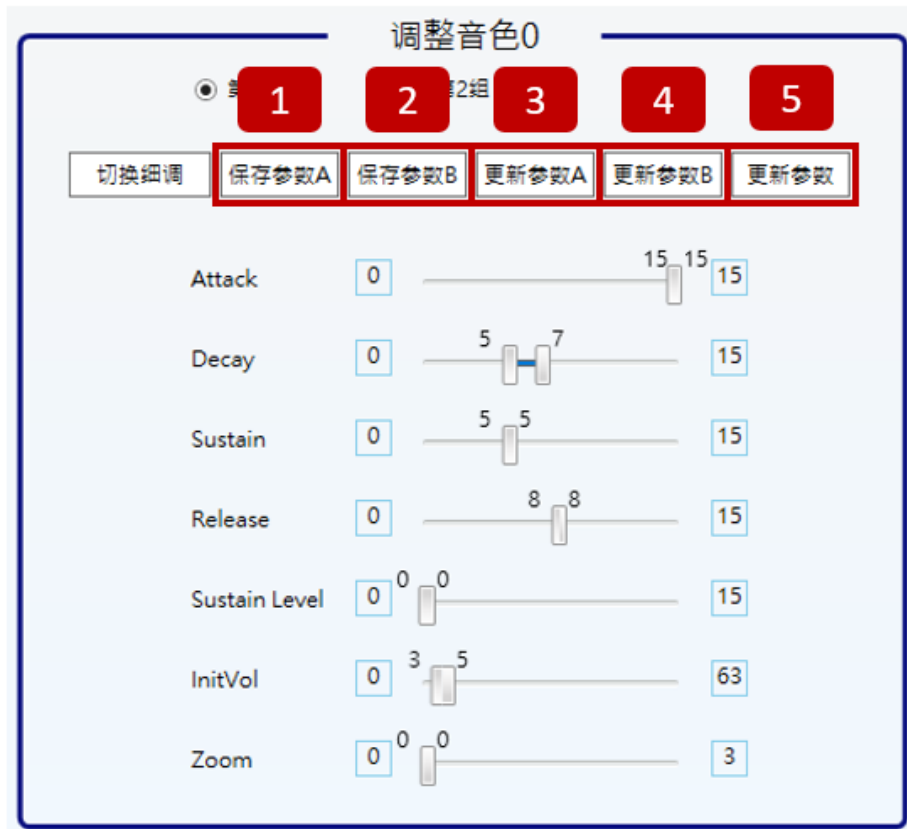
- 步骤 1: 点击“音色调整”图标。
- 步骤 2: 选择音色号 (范围 0~127)。
- 步骤 3: 界面显示所选择的音色号。
- 步骤 4: 选择第 1 或 2 组音色;
加载的音色库 (*.sf2) 须有第 2 组音色存在, 方可點選第 2 组音色进行参数修改; 若不存在, 第 2 组音色将无法點選。
- 步骤 5: 设定音色参数;
提供参数 (Attack/Decay/Sustain/Release/Sustain Level/InitVol/Zoom) 供调整。
- 步骤 6: 储存音色参数。

4.3.2 音色细调



- 步骤 1: 切换至音色细调模式。
- 步骤 2: 显示与选择音色内的音色样本 (wav)。此例: 选择音色 0 内“C0-D3”样本。
- 步骤 3: 设定音色参数 (Attack/Decay/Sustain/Release/Sustain Level/InitVol/Zoom)。此例: 设定音色 0 “C0-D3”样本的音色参数。
- 步骤 4: 可切换回音色粗调模式。

4.3.3 调试 ADSR 参数

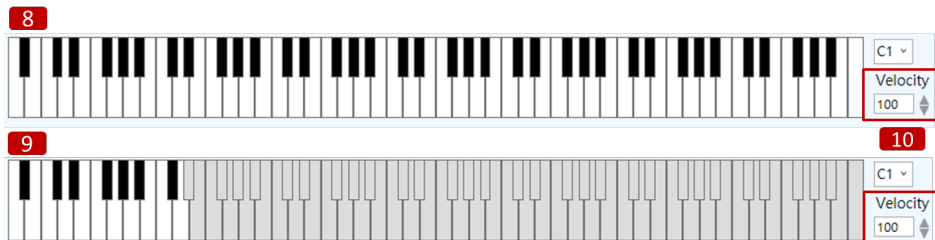




1. 保存参数 A: 储存当前 ADSR 参数至参数 A。
2. 保存参数 B: 储存当前 ADSR 参数至参数 B。
保存参数 A/B 可用于快速比对参数差异的效果。
3. 更新参数 A: 发送参数 A 至开发板上。
4. 更新参数 B: 发送参数 B 至开发板上。
5. 更新参数: 发送当前设定 ADSR 参数至开发板上。
6. 显示参数 A/B 所保存的参数 (鼠标指到“保存参数 A/B”上)。



- 7: 当“更新参数 / 更新参数 A / 更新参数 B”更新成功。
- 注意: 更新参数前需先确认 PC USB 连接至开发板 USB_A, 才能更新参数。



- 8: 键盘 (音色粗调); 显示 88Key 范围, 点击键盘可测试音色声音效果。

9: 键盘 (音色细调): 显示出音色内 wav 样本的音符范围, 点击画面上键盘可测试音色声音效果。

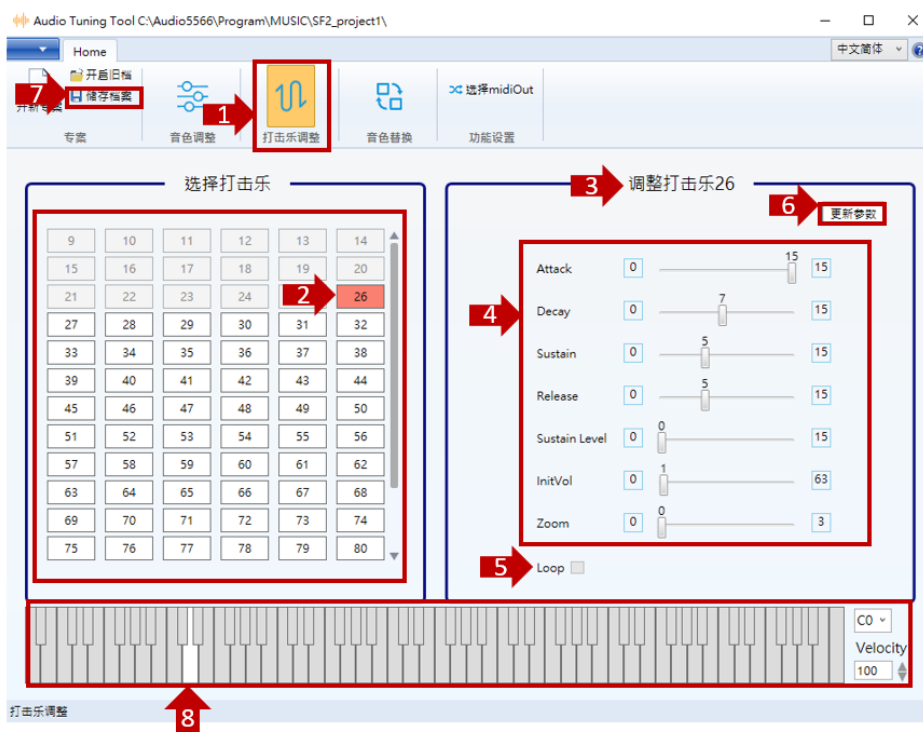
10: 调整键盘发音力度 (范围 0~127)。

注意: 1. 调试音色参数效果后, 需将调整完毕的音色通过“替换”功能, 替换到系统音色库中; 通过 Audio Workshop 进行“产出 Bin”“产出 Hex”并将产出的 bin 与 hex 通过“下载”功能, 下载至开发板上, 才能在系统上调用调整后的音色。

2. 更新参数前需先确认 PC USB 连接至开发板 USB_A 接口, 才能更新参数。

4.4 打击乐参数调整

4.4.1 调试 ADSR 参数



- 步骤 1: 界面上点击“打击乐调整”图示。
- 步骤 2: 选择打击乐号 (范围 9~96)。
- 步骤 3: 界面显示所选择的打击乐号。
- 步骤 4: 设定打击乐参数;
提供参数 (Attack/Decay/Sustain/Release/Sustain Level/InitVol/Zoom) 供调整。
- 步骤 5: 显示该打击乐是否有 Loop;
打勾表示有 Loop, 此例无 Loop。
- 步骤 6: 发送当前设定 ADSR 参数至开发板上。
- 步骤 7: 储存打击乐参数。
- 步骤 8: 点击画面上键盘可测试当前打击乐声音效果。

注意：

1. 调试打击乐参数效果后，需将调整完毕的打击乐音色通过“替换”功能，替换到系统音色库中；通过 Audio Workshop 进行“产出 Bin”“产出 Hex”并将产生的 bin 与 hex 通过“下载”功能，下载至开发板上，才能在系统上调用调整后的打击乐音色。
2. 更新参数前需先确认 PC USB 连接至开发板 USB_A 接口，才能更新参数。

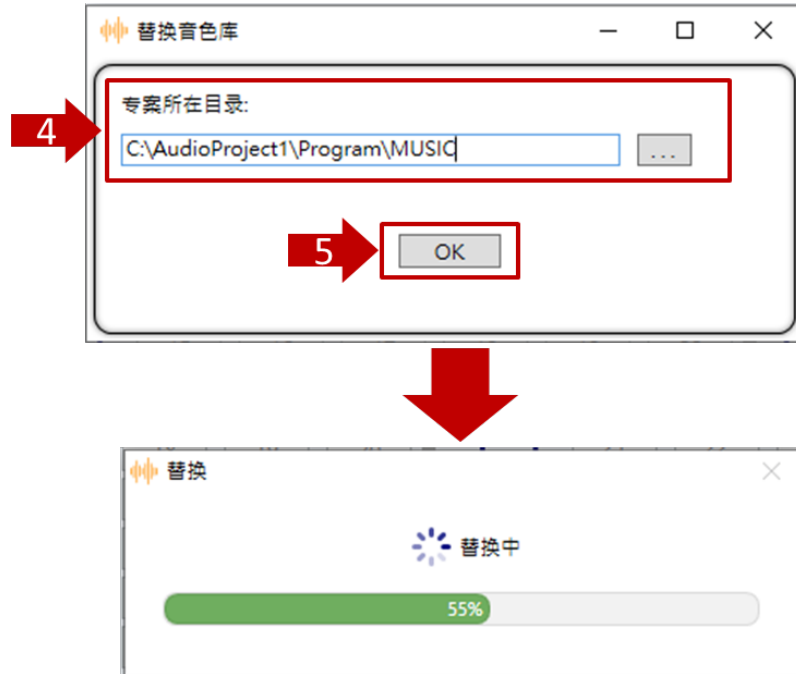
4.5 音色替换

- 音色库替换：提供将系统音色库(所有音色)替换成用户音色库(所有音色)。
- 个别音色替换：提供用户替换个别的音色或打击乐。

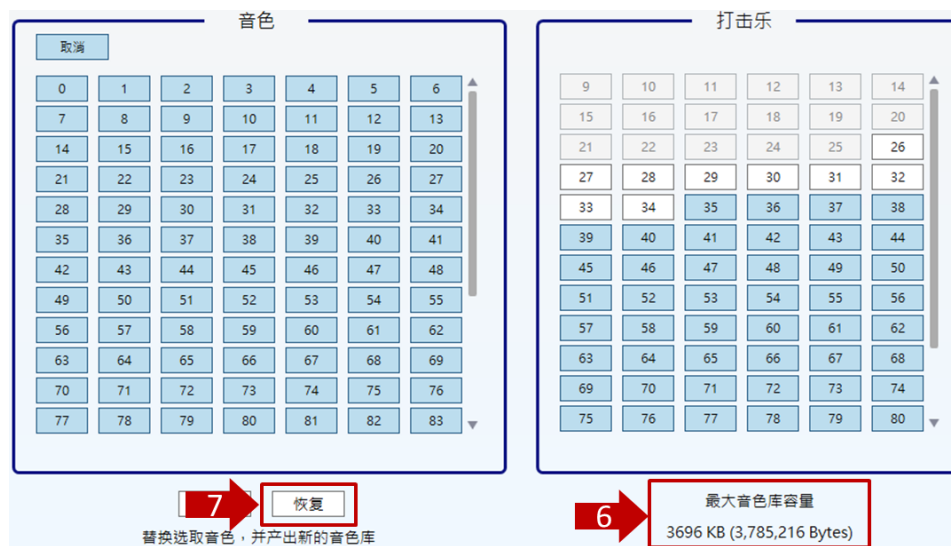
4.5.1 音色库替换



- 步骤 1: 界面上点击“音色替换”图标。
- 步骤 2: 点击“一键全选”按钮, 此时会将所有音色与打击乐选取 (显示为蓝色)。
- 步骤 3: 点击“替换”按钮后, 如下图。



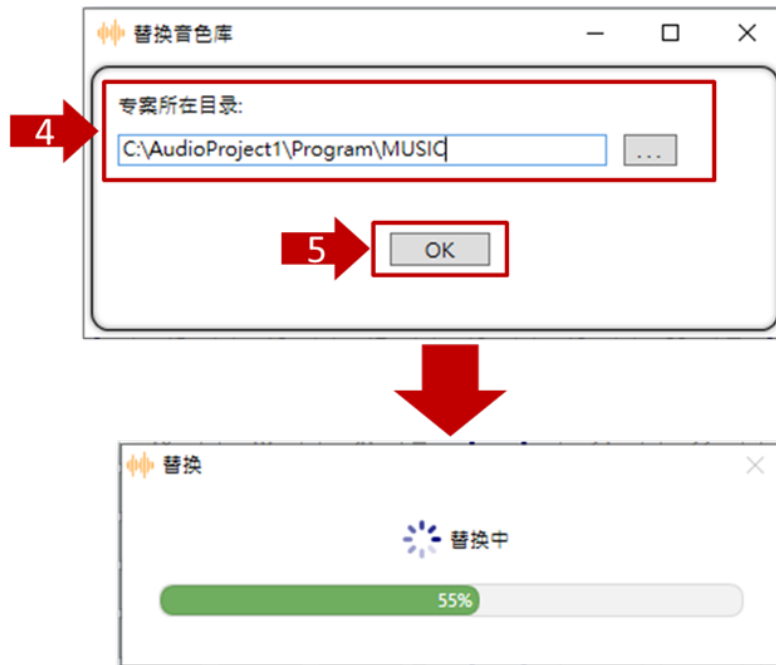
- 步骤 4: 设定替换的 Audio Workshop 专案路径, 使用默认路径即可。
- 步骤 5: 点击“OK”按钮, 则开始进行音色与打击乐数据替换, 进度 100% 及代表替换完成。
- 步骤 6: 替换完成 界面上会估算出替换音色后占用的最大资源 (SPI Flash ROM 容量)。
- 步骤 7: 点击“恢复”按钮, 可恢复系统默认音色库。



4.5.2 个别音色替换



- 步骤 1: 界面上点击“音色替换”图标。
- 步骤 2: 选取欲替换的音色与打击乐号 (选取显示为蓝色)。
此例: 选择音色 0/1/2/3/4/5/6/11/12/13/14/15/16/17, 选择打击乐 26/27/28。
除鼠标单击音色号选择外, 可搭配“shift”键一次选择多个;
此例: 音色(0 “shift” 6, 11 “shift” 13, 14 “shift” 17), 打击乐(26 “shift” 28)
- 步骤 3: 点击“替换”按钮后, 如下图。



- 步骤 4: 设定替换的 Audio Workshop 专案路径, 使用默认路径即可。
- 步骤 5: 点击“OK”按钮, 则开始进行音色与打击乐数据替换; 进度 100% 即代表替换完成。
- 步骤 6: 替换完成 界面上会估算出替换音色后占用的最大资源 (SPI Flash ROM 容量)。
- 步骤 7: 点击“恢复”按钮, 可恢复系统默认音色库。

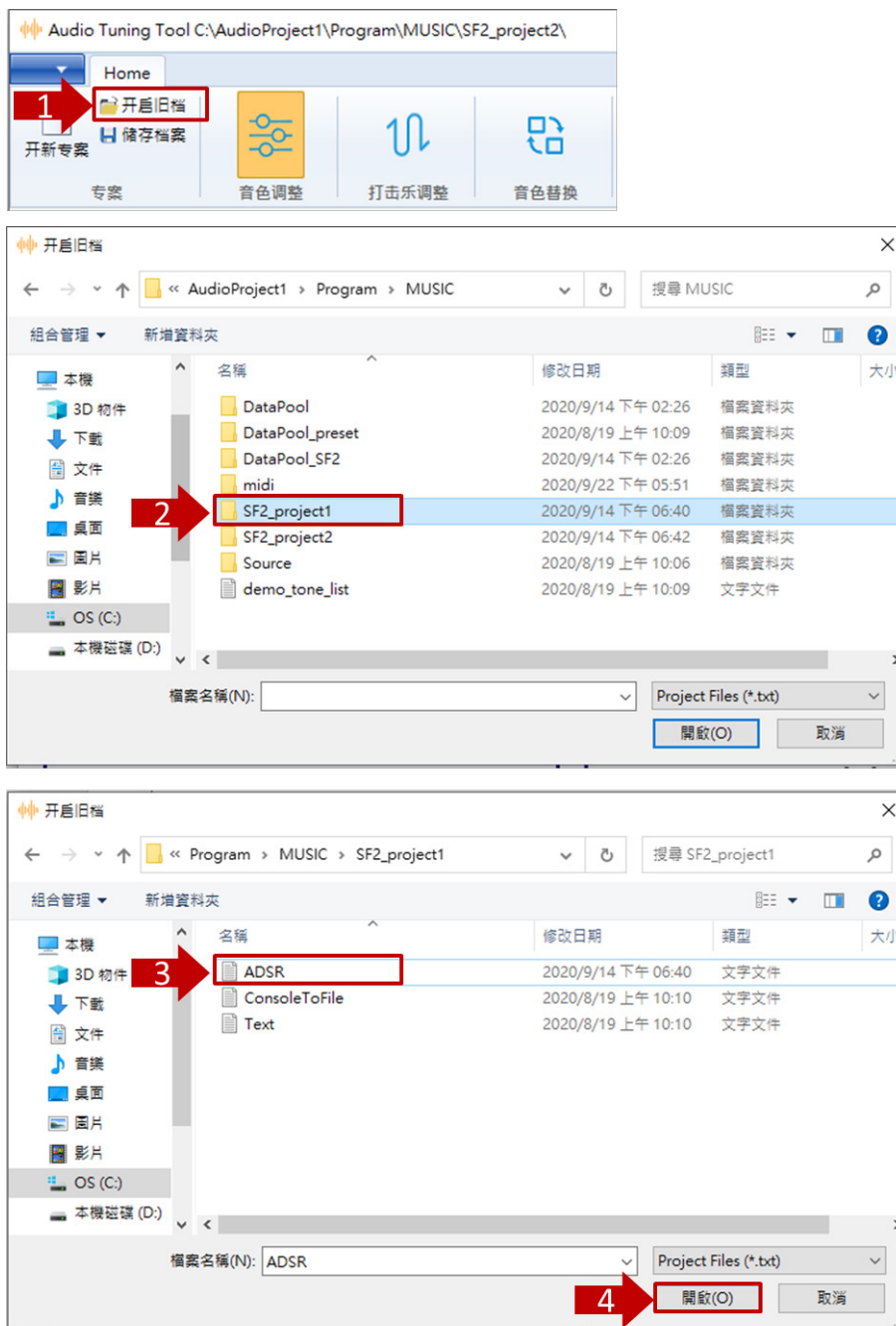


- 步骤 8: 替换完成的音色, 会同步显示到“音色调整”与“打击乐调整”界面上。

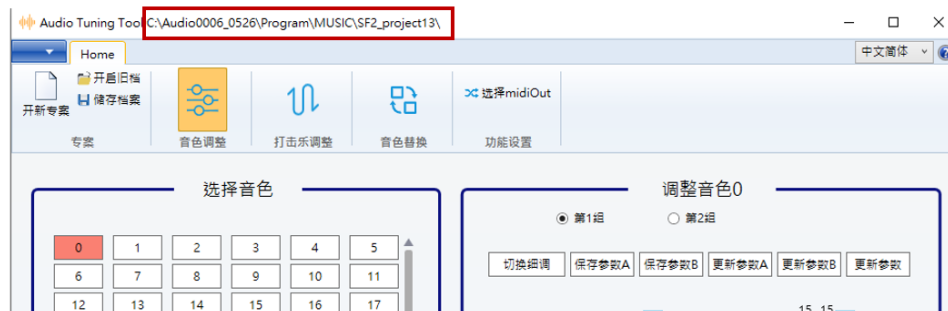


4.6 开启专案

用户可通过“开启旧档”，开启音色专案或切换音色专案。



- 步骤 1：界面上点击“开启旧档”。
- 步骤 2：选择欲开启的音色专案。
- 步骤 3：选择目录下“ADSR.txt”。
- 步骤 4：点击“开启”，进入操作界面，即完成“开启旧档”如下图。



- 开启专案后，专案的路径会显示于标题栏，如上图。
- 操作界面会显示出：音色参数 / 打击乐参数 / 上次替换的音色信息。

五、函式库说明

5.1 播放函数

函数	说明
void L_Audio_Init(void)	初始化设定
void L_Audio_Play(u8 audio_type, u32 index)	播放 (MIDI / 语音 / 音效)
void L_Audio_Stop(void)	停止播放所有声音
void L_Midi_Stop(void)	停止播放 MIDI
void L_Sntnc_Main_Loop(void)	主循环语句处理
void L_Sntnc_Play(u32 index)	播放语句
void L_Sntnc_Stop(void)	停止播放语句
EventStatus L_Midi_Finish(void)	判断 MIDI 是否播放完成
EventStatus L_Audio_Finish(void)	判断语音 / 音效是否播放完成
void L_Audio_Vol(u8 volume)	音量设定

5.1.1 初始化函数

L_Audio_Init()

说明：对 Clock/Timer/QSPI/MIDI Engine/DAC / 音量...等初始化。

5.1.2 播放函数 (User function)

L_Audio_Play(audio_type, index)

说明：播放 MIDI 或语音或音效。

参数：

audio type:

c_midi_type : MIDI

c_voice_type : 语音

c_sound_type : 音效

index: 第几首 (从 1 开始)

范例：

```
L_Audio_Play(c_midi_type, 2); // 播放第二首 midi
```

L_Audio_Stop()

说明：停止 MIDI 播放，并停止正在播放中的所有声音。

L_Midi_Stop()

说明：停止 MIDI 播放，并停止正在播放中的 midi 声音。

L_Sntnc_Main_Loop()

说明：使用播放语句功能，需于主循环 main() 中的 while(1) 中加入此函数。

L_Sntnc_Play(index)

说明：播放语句。

参数：

Index：第几句语句（从 1 开始）。

范例：

```
L_Sntnc_Play(2); // 播放第二首语句
```

L_Sntnc_Stop()

说明：停止播放语句。

5.1.3 音量设定函数 (User function)**L_Audio_Vol(volume)**

参数：

Volume：0~15。

0 表示静音，1~15 表示音量由小到大。

范例：

```
L_Audio_Vol(8); // 设定音量 8
```

5.1.4 播放状态函数**EventStatus L_Audio_Finish()**

返回值：

ENABLE：语音 / 音效已播放完。

DISABLE：语音 / 音效未播放完。

EventStatus L_Midi_Finish()

返回值：

ENABLE：MIDI 已播放完。

DISABLE：MIDI 未播放完。

5.2 音效函数

函数	说明
void L_Set_PadelSlop(u8 sloppadel)	延音踏板
void __L_Set_Pitch_Bend(u8 r_level)	滑音轮
void __L_Set_Vib(u8 r_level)	颤音
void __L_dec_Tempo()	MIDI 速度减少
void __L_inc_Tempo()	MIDI 速度增加
void __L_normal_Tempo()	MIDI 速度复原
void __L_Set_Tempo1(u16 r_tempo)	设定 MIDI 速度

函数	说明
void <code>__L_Limit_Release(u8 level)</code>	快速关闭声音

5.2.1 音效函数

`__L_Set_PadelSlop(u8 sloppadel)`

参数:

`sloppadel`: 0~8。

8 关闭延音踏板; 0~7 数值越小, 延音越长。

范例:

```
__L_Set_PadelSlop(8); // 关闭延音踏板
```

`__L_Set_Pitch_Bend(r_level)`

参数:

`r_level`: 0~255。

0~100: 音调升高, 100 对应 2 个半音。

255~156: 音调降低, 156 对应低 2 个半音。

范例:

```
__L_Set_Pitch_Bend(-50); // 降低一个半音
```

```
__L_Set_Pitch_Bend(50); // 升高一个半音
```

`__L_Set_Vib(r_level)`

参数:

`r_level`: 0~4。

颤音设置 `r_level` 的值控制抖动范围, `r_level` 以 5 音分为单位。

例如: 设置 `r_level=1`; 颤音抖动 5 音分, `r_level=2` 颤音抖动 10 音分。

范例:

```
__L_Set_Vib(4); // 颤音抖动 20 音分
```

`__L_dec_Tempo()`

说明: MIDI 速度减少 1 拍; 范围: 每分钟 30~280 拍。

`__L_inc_Tempo()`

说明: MIDI 速度增加 1 拍; 范围: 每分钟 30~280 拍。

`__L_normal_Tempo()`

说明: MIDI 速度复原; 范围: 每分钟 30~280 拍。

`__L_Set_Tempo1(u16 r_tempo)`

说明: 设定 MIDI 速度; `r_tempo` 范围 30~280。

范例:

```
__L_Set_Tempo1(80); // MIDI 速度 80
```

`__L_Limit_Release(u8 level)`

说明: 快速关闭声音, 对正在播放的声音有效。

参数:

`level`: 1~7 其中 7 表示余音关闭最快, 1 表示余音关闭最慢。

范例:

```
__L_Limit_Release(4);
```


5.3 系统函数

函数	说明
void __L_Func_Midi_Mute_Control(bool onoff)	MIDI 播放静音控制
void __L_Func_Midi_Pause_Control(bool onoff)	MIDI 播放暂停控制
void __L_User_Measure_Finish(void)	小节子函数
void __L_User_Beat_Finish(void)	节拍子函数
void __L_User_Half_Beat_Finish(void)	半拍子函数
void __L_Note_On(u8 r_type, u8 r_def_num, u8 r_keycode, u8 r_volume, u16 r_id_delay, u8 r_pan, u8 r_track)	音符发音
void __L_Note_Off(u8 r_id)	关闭音符发音
void __L_Midi_NRPN(u32 r_NRPN)	NRPN 事件输出
bool __L_Check_Wave_End()	检查通道的声音是否播放完毕
void __L_Play_Midi(u16 number, E_Midi_Play_Begin_Middle_Typedf_e_play_begin_middle, E_Midi_Pause_Typedf_r_e_pause)	播放 MIDI

5.3.1 系统函数

__L_Func_Midi_Mute_Control(bool onoff)

说明：MIDI 播放静音控制；MIDI 事件时间继续。

参数：

Bool type:

TRUE mute
FALSE 正常

范例：

```
__L_Func_Midi_Mute(TRUE); // 播放静音
```

__L_Func_Midi_Pause_Control (bool onoff)

说明：MIDI 播放暂停控制；midi 事件时间暂停，并结束目前正在发的音。

参数：

Bool type:

TRUE Pause
FALSE 正常

范例：

```
__L_Func_Midi_Pause_Control(TRUE); // 播放暂停
```

__L_User_Measure_Finish ()

说明：功能调用界面。小节结束，开发者可以此知道目前已经是小节结束。

__L_User_Beat_Finish ()

说明：功能调用界面。一拍结束，开发者可以此知道目前已经是一拍结束。

__L_User_Half_Beat_Finish(void)

说明：功能调用界面。半拍结束，开发者可以此知道目前已经是半拍结束。

__L_Note_On(u8 r_type, u8 r_def_num, u8 r_keycode, u8 r_volume, u16 r_id_delay, u8 r_pan, u8 r_track)

说明：音符发音。

参数：

r_type: 发音种类 (c_notekey_play、c_midi_play)
 若是 c_notekey_play 发音，则需要对应的 __L_Note_Off 去关音
 若是 c_midi_play 发音则，发 r_id_delay×1/48beat 长度
 r_def_num: 音色号
 r_keycode: 音高 (0~127)
 r_volume: 音量 (0~63)
 r_id_delay: 发音 id 及时长
 r_pan=0; 默认
 r_track=0; 默认

举例：

c_notekey_play 表示按键发音，还需要对应的关闭音符发音。如果是 c_midi_play，只需要给发音时长就好了。

`__L_Note_On(c_notekey_play, 20, 12, 0, 1, 0, 0)`

此函数表明是按键发音，音色号 20，发音值 keycode 为 12，音量为最大值 0 (0 为最大音量)，r_id_delay 值为 1 (用于关闭音符发音时使用)。如果要关闭这个发音则用此函数 `__L_Note_Off(1)` 就可以关闭 r_id_delay=1 的按键发音。

`__L_Note_On(c_midi_play, 20, 12, 0, 1, 0, 0)`

此函数表明是此发音发 1 个 1/48 拍就进入释放了，r_id_delay=1 发音的长度为 1/48 拍，若是 48 则发音长度为 1 拍。不需要关音函数。

__L_Note_Off(u8 r_id)

说明：关闭音符发音。

参数：

r_id: 对应 __L_Note_On() 里面的 r_id_delay。

__L_Midi_NRPN(u32 r_NRPN)

说明：功能调用界面。传输转换程序转换的 NRPN。

完整的 midi NRPN event 应该含有 0x6/x26/0x62/0x63。

__L_Check_Wave_End()

说明：检查通道的声音是否播放完毕。

返回值：

TRUE: 播放完毕。

FALSE: 还有声音在通道中。

void __L_Play_Midi(u16 number, E_Midi_Play_Begin_Middle_Typedf r_e_play_begin_middle, E_Midi_Pause_Typedf r_e_pause)

说明：播放 midi

参数：

Number: midi 序号

r_e_play_begin_middle=c_enum_midi_play_begin; 默认

r_e_pause=c_enum_midi_not_pause; 默认返回值

范例：

`__L_Play_Midi(0, c_enum_midi_play_begin, c_enum_midi_not_pause);`

此语句播放平台加入的第一个 MIDI。

5.4 其它说明

- MIDI IN 事件

通过平台选择，默认 USB MIDI，可同时选择

```
#define USB_MIDI_ENABLE (1)
```

支持 usb midi standard input, USB midi in 支持命令和转换

支援命令行表如下

		USB midi in	Convert
Note off(8X)		V	V
Note on(9X)		V	V
Aftertouch(AX)		X	X
Controller(BX)	Volume(7)	V	V
	PAN(A)	V	V
	Padel(40)	V	V
	NRPN(6/26/62/63)	X	V
	All soundoff(78)	V	V
Program Change(Cx)		V	V
Channel Pressure(Dx)		X	X
Pitch Wheel(Ex)		V	V
Meta message(Fx)	End of Track(2F)	X	V
	Time Signature(58)	X	V
	Tempo(51)	X	V

- MIDI 播放总 tick 及当前播放 tick。

__midi_total_tick: 总 tick

__midi_current_tick: 当前 tick

- 使用剩下空间时

当用户需要用到外部 Flash 剩余的容量时，请勾选 APP 更新功能，容量默认为 1Mbits 是用户存放用户 MIDI 曲的用量。

```
unsigned long const c_addr_user_midi_table=0xcf000;
```

// 用户 MIDI 曲存放的起始地址，以 Word 为单位。

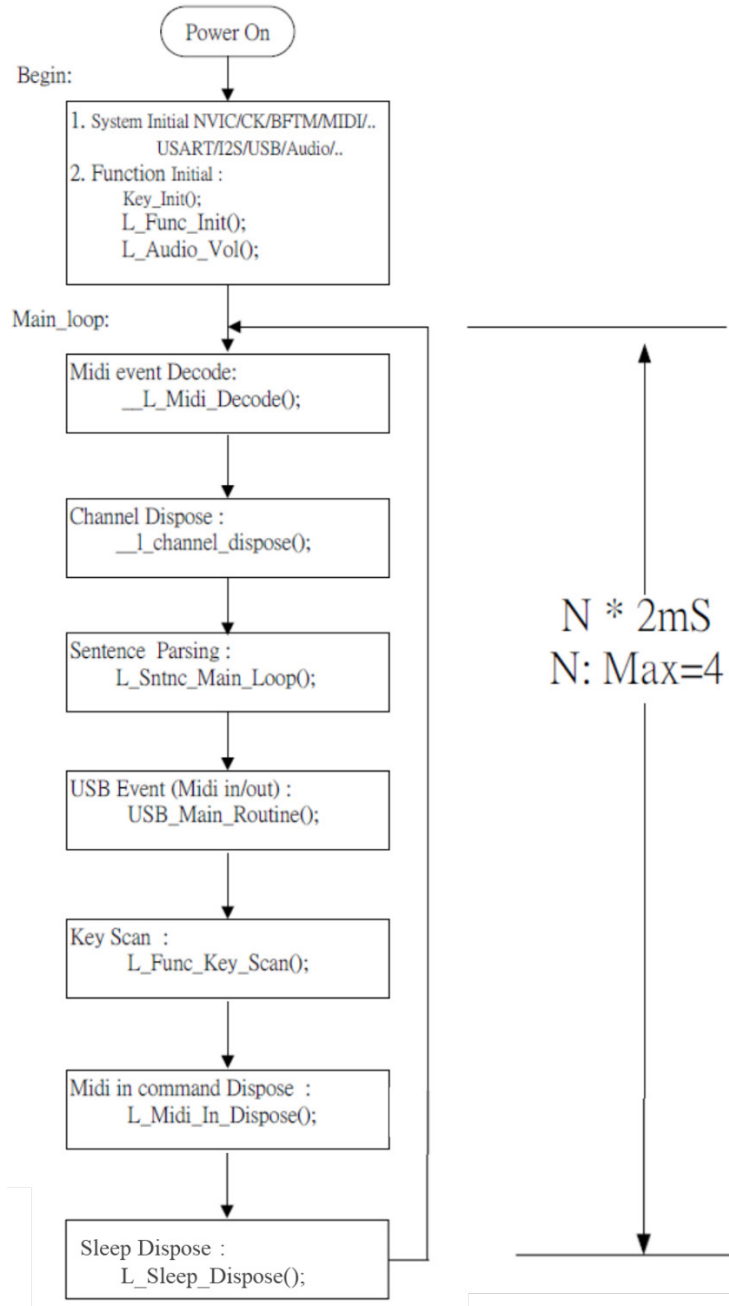
```
unsigned long const c_addr_user_flash=0xdf000;
```

// 用户数据存放的起始地址，以 Word 为单位。

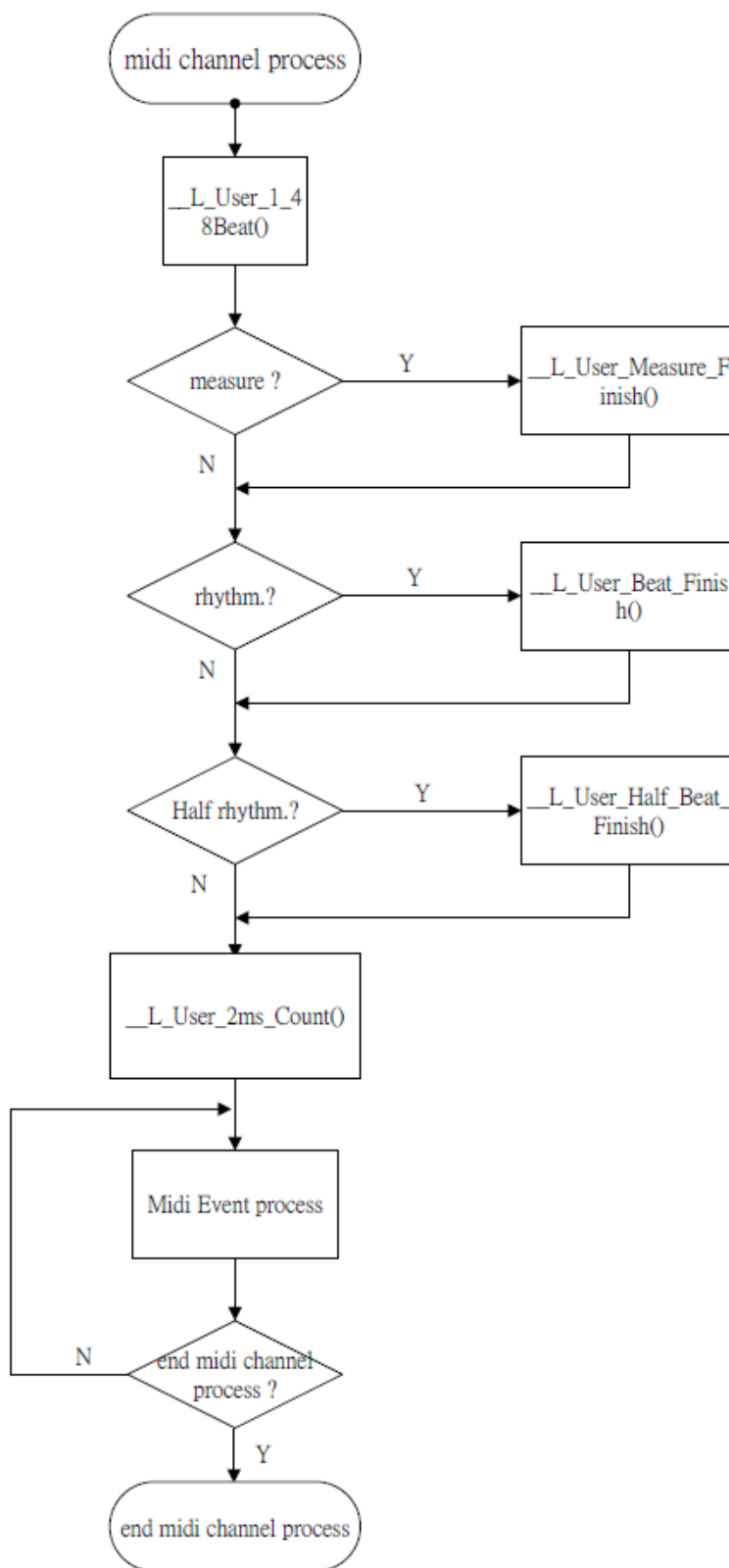
六、附录

6.1 程序流程

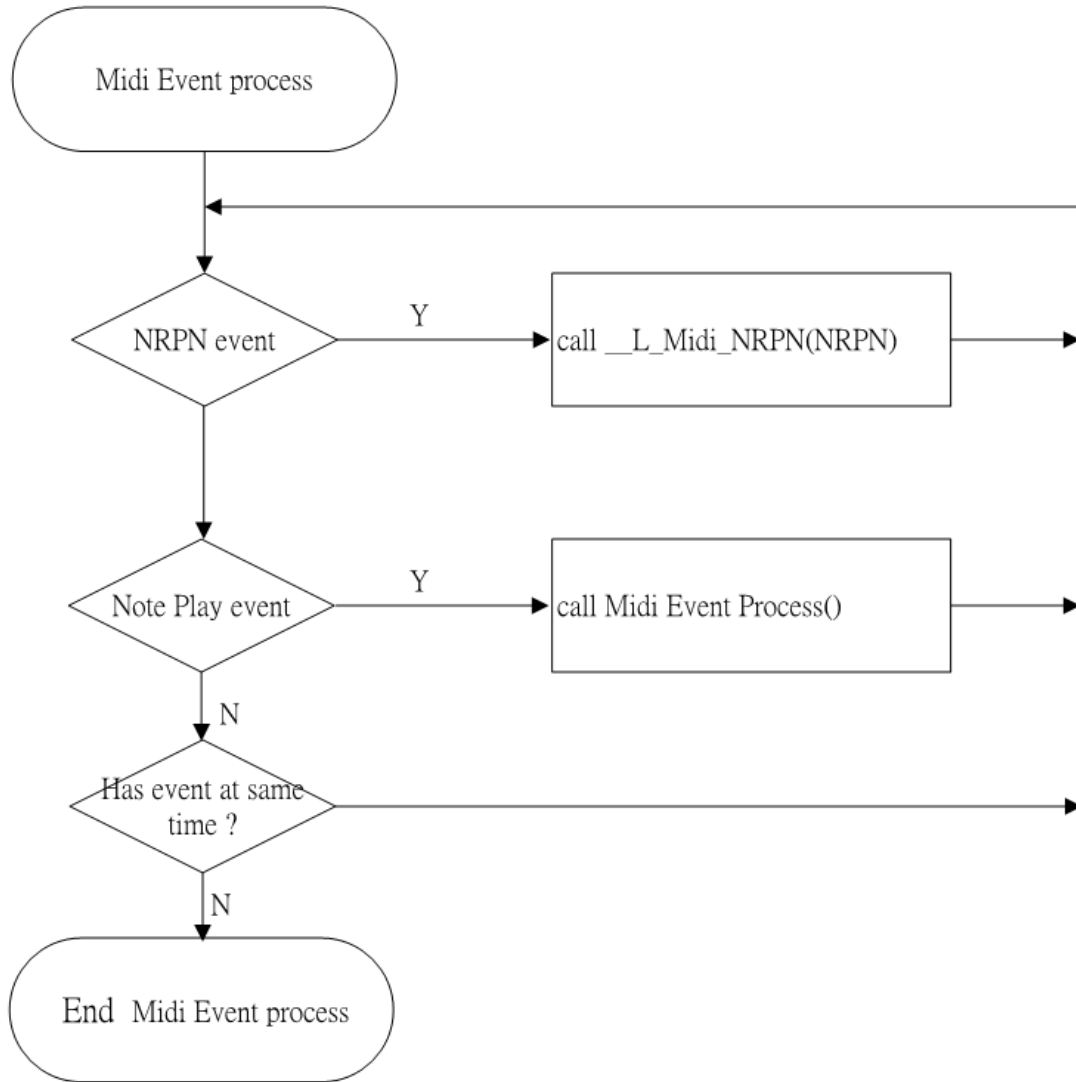
程序主流程

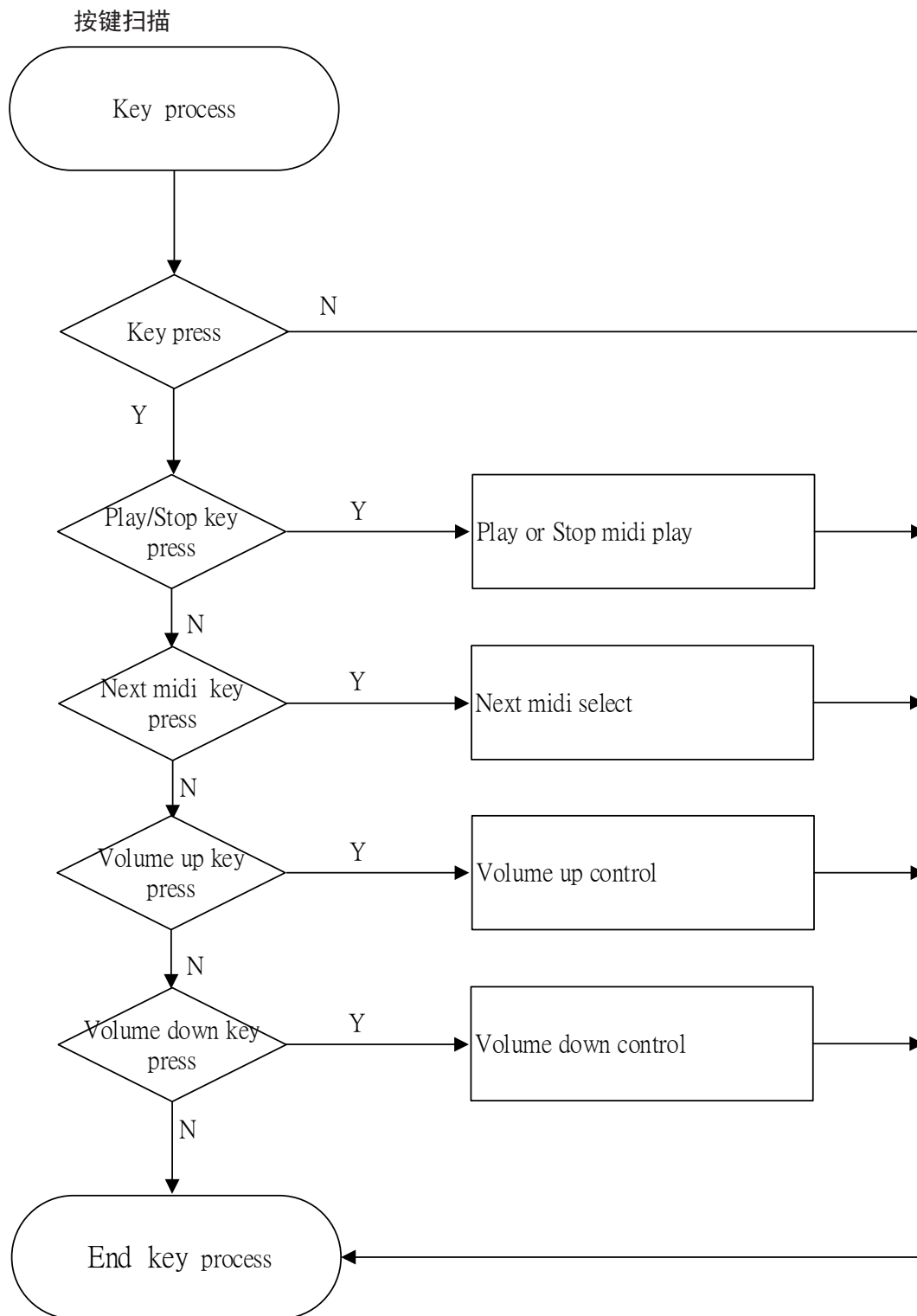


Midi 解码

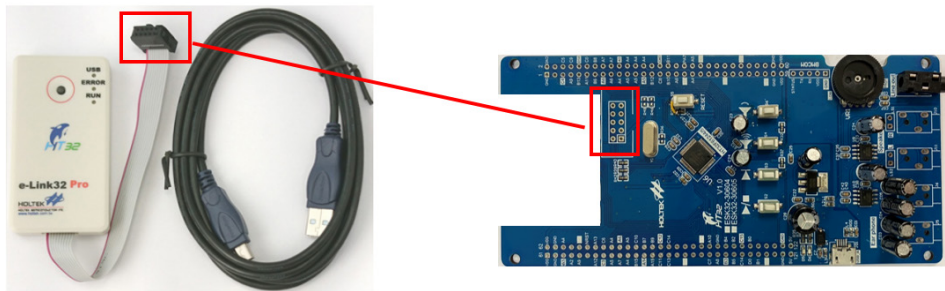


Midi 事件处理

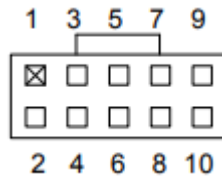




6.2 e-link32 Pro 连接方式



根据 1.3.1 小节所列开发板上 SWD-10P 连接器的引脚 (详细引脚说明可参照附录 6.4 小节), 对应 e-link32 Pro 的接口进行连接, e-link32 Pro 的接口说明如下所示。



Pin#	Description	Pin#	Description
1	3.3V	2	SWDIO
3	GND	4	SWCLK
5	GND	6	Reserved
7	NC (VCOM_RXD ^(Note))	8	NC (VCOM_TXD ^(Note))
9	GND	10	Reset

注意: 在 e-link32 Pro 接口中, 引脚 7 以及引脚 8 为 VCOM_RXD 以及 VCOM_TXD; 在 e-link32 接口中, 引脚 7 以及引脚 8 为 NC。

6.3 Data Flash ROM 烧录

6.3.1 开启烧录专案

1

名称	修改日期	类型	大小
Encode	2022/6/17 15:37	文件夹	
Program	2022/6/17 15:37	文件夹	
Sound	2022/6/17 15:37	文件夹	
Audio5566	2022/6/17 15:37	AWP 文件	1 KB

2

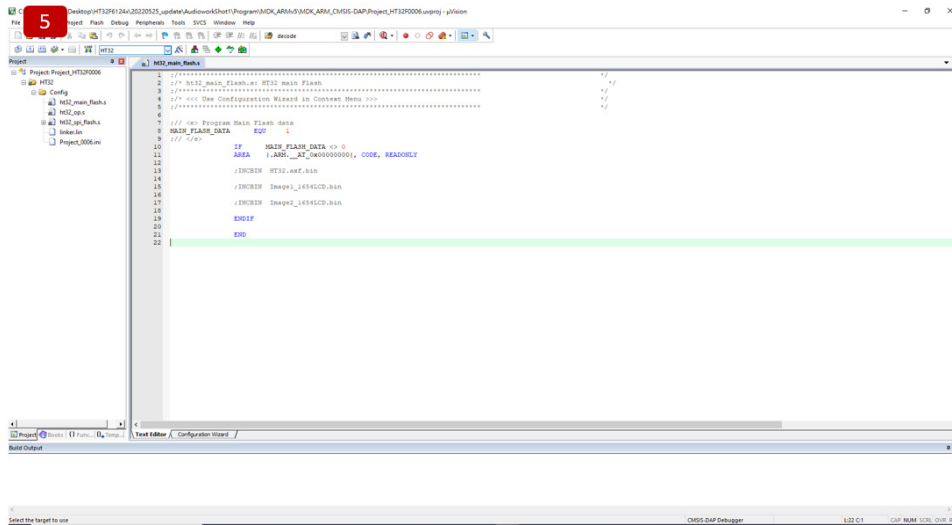
名称	修改日期	类型	大小
LEVI_LIB	2022/6/17 14:25	文件夹	
MDK_ARMv5	2022/6/17 15:37	文件夹	
MUSIC	2022/6/17 15:37	文件夹	
SYS	2022/6/17 14:25	文件夹	
USER	2022/6/17 14:25	文件夹	
ht32_usbd_descriptor.c	2022/6/17 15:37	C Source	21 KB
ht32_usbd_descriptor.h	2020/7/31 16:18	C/C++ Header	3 KB
ht32_usbd_descriptor_c	2020/7/31 16:18	C Source	21 KB
ht32_usbd_descriptor_h	2020/7/31 16:18	C/C++ Header	3 KB
ht32f5xxx_01_it.c	2022/5/31 14:31	C Source	11 KB

3

名称	修改日期	类型	大小
flash_main	2022/6/17 14:25	文件夹	
flash_spi	2022/6/17 14:25	文件夹	
HT32	2022/6/17 14:25	文件夹	
MDK_ARM_CMSIS-DAP	2022/6/17 14:25	文件夹	
build_log	2022/6/17 15:37	文本文档	4 KB
Error	2022/6/17 15:37	Windows 批处理...	1 KB
fromelf	2020/7/31 16:18	文本文档	2 KB
ht32_op.s	2020/7/31 16:18	Assembler Source	19 KB
HT32F5xxx_01_DebugSupport	2020/7/31 16:18	配置设置	6 KB

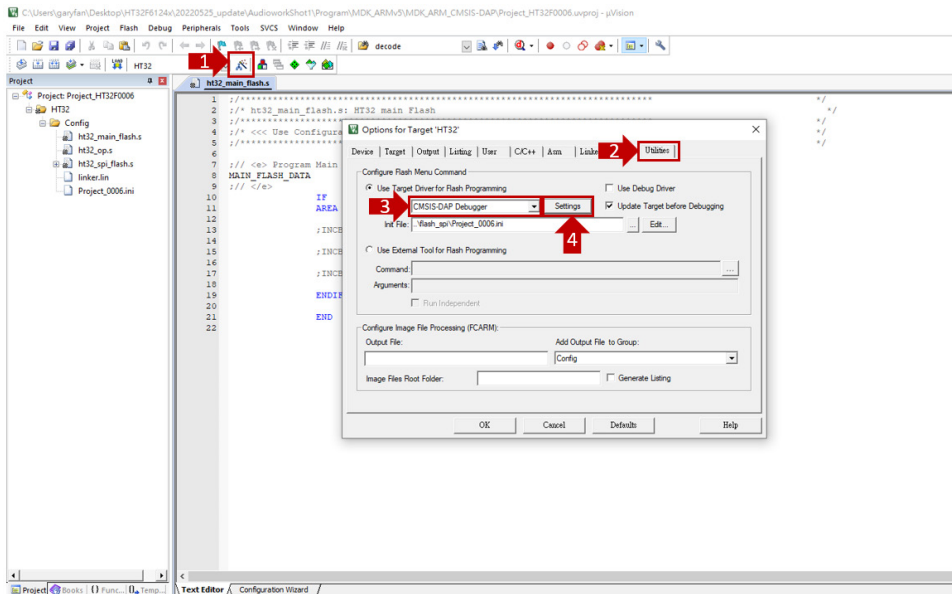
4

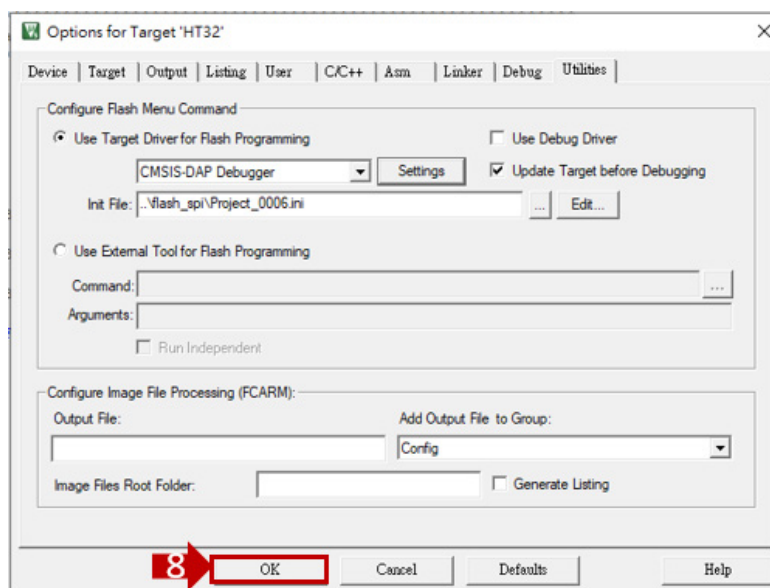
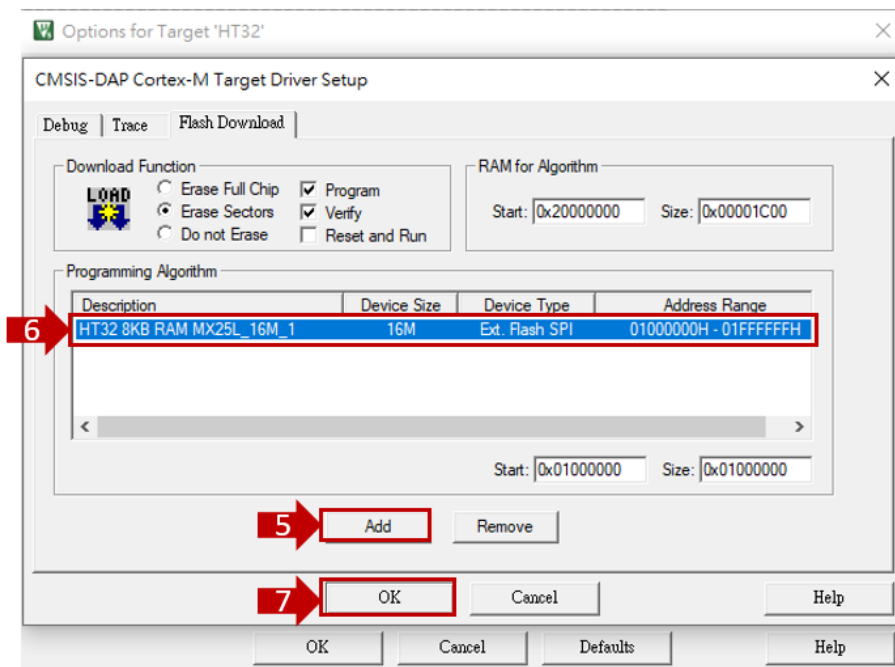
名称	修改日期	类型	大小
HT32	2022/6/17 14:25	文件夹	
download_log	2020/12/15 14:01	文本文档	1 KB
Error	2020/12/15 13:59	Windows 批处理...	1 KB
log	2020/11/24 8:50	文本文档	3 KB
Project_HT32F0006.uvopt	2022/4/18 16:39	UVOPT 文件	8 KB
Project_HT32F0006	2022/1/25 18:21	獼ision4 Project	20 KB
Project_HT32F0006_HT32.dep	2022/5/26 16:11	DEP 文件	2 KB
Project_HT32F61240.uvopt	2022/6/6 11:21	UVOPT 文件	8 KB
Project_HT32F61240	2022/3/16 13:15	獼ision4 Project	20 KB
Project_HT32F61240_HT32.dep	2022/5/10 10:51	DEP 文件	2 KB
Setting_16MFlashDownload	2020/10/28 14:26	PNG 文件	13 KB
Setting_32M_64MFlashDownload	2020/10/28 14:26	PNG 文件	15 KB
Setting_Debug	2020/10/28 14:26	PNG 文件	18 KB



- 步骤 1: 打开专案文件夹内的 Program 文件夹。
- 步骤 2: 选择 MDK_ARMv5 文件夹并开启。
- 步骤 3: 选择 MDK_ARM_CMSIS-DAP 文件夹并开启。
- 步骤 4: 开启烧录程序专案。
HT32F0006 与 HT32F61355/HT32F61356/HT32F61357 选用 Project_HT32F0006 烧录程序、HT32F61244/HT32F61245 选用 Project_HT32F61240 烧录程序。
- 步骤 5: 进入 Keil 主画面。

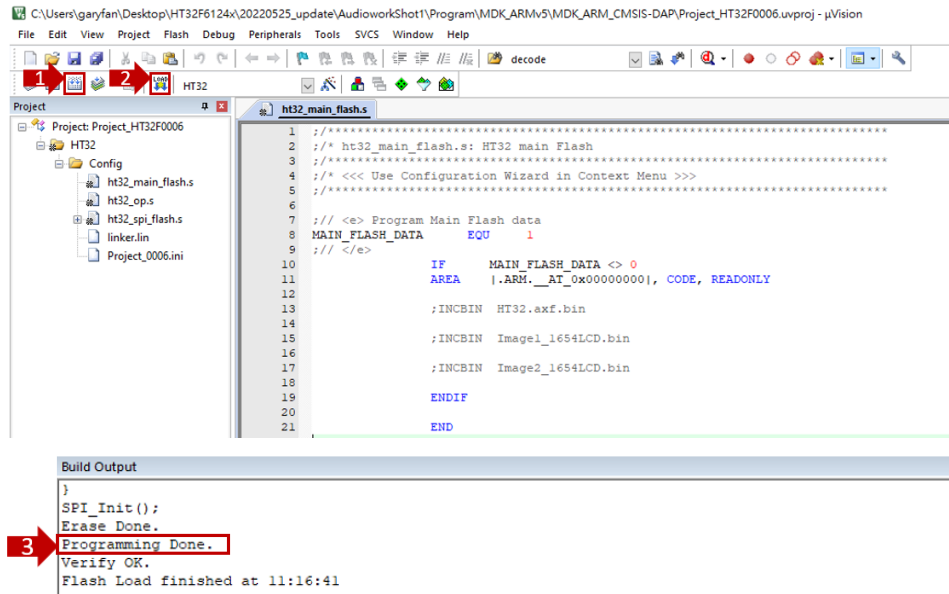
6.3.2 设定 Loader





- 步骤 1: 点击 Options for Target “HT32”。
- 步骤 2: 选择 Utilities。
- 步骤 3: 点选 Use Target Driver for Flash Programming 选择 CMSIS-DAP Debugger。
- 步骤 4: 点击 Settings 设定 Loader。
- 步骤 5: 点选 Add, 加入欲使用的 Loader。
- 步骤 6: 选择需要使用的 Loader。
- 步骤 7: 点选 “OK” 设定完成, 并离开 Settings 画面。
- 步骤 8: 点选 “OK” 设定完成, 并离开 Options for Target “HT32” 画面。

6.3.3 烧录音色库数据 (.bin)



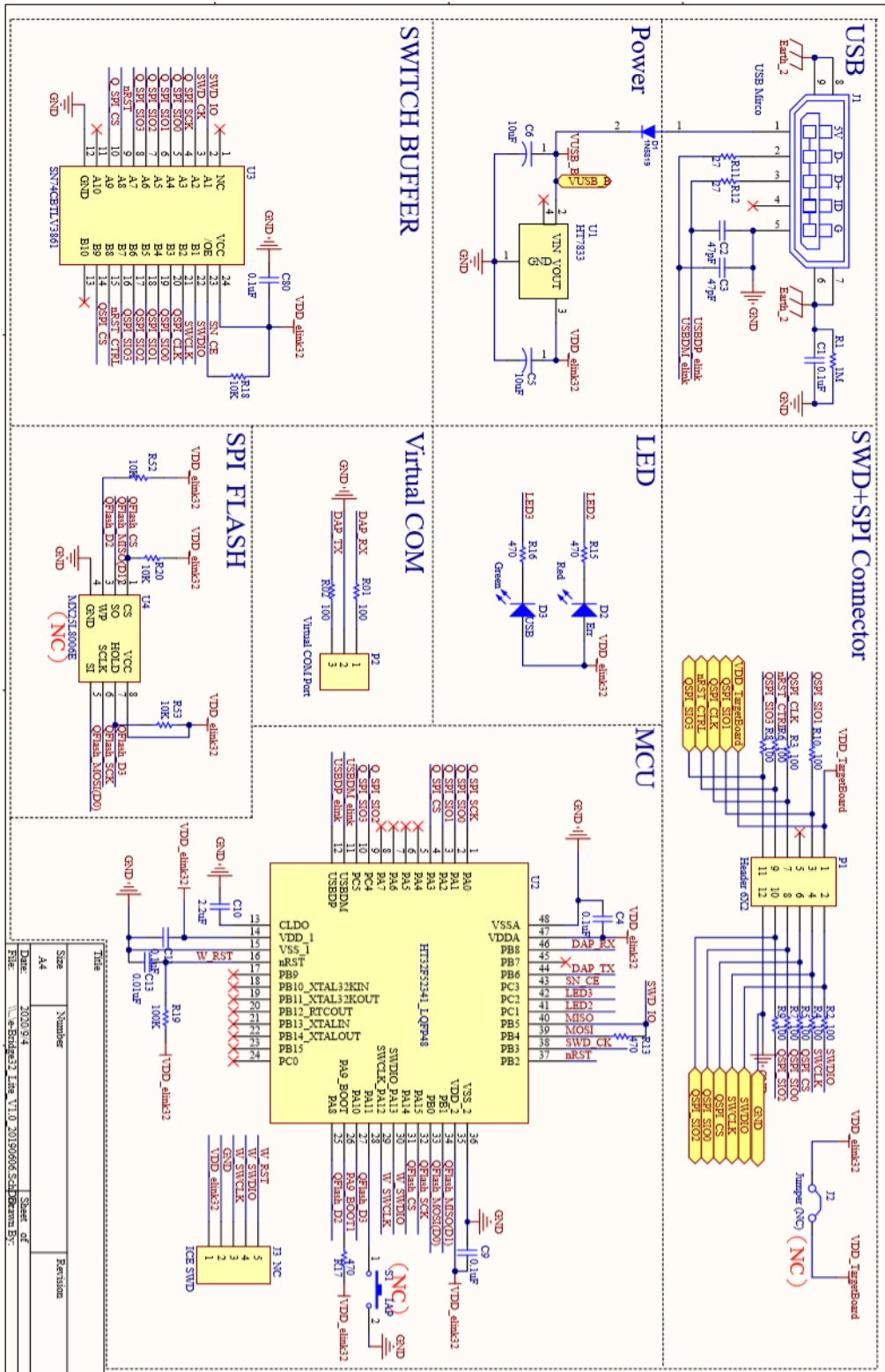
- 步骤 1: 点击 Rebuild 进行编译。
- 步骤 2: 点击 Download 开始进行烧录。
- 步骤 3: 于 Build Output Window 确认 Programming Done 表示烧录成功。

6.4 开发板原理图

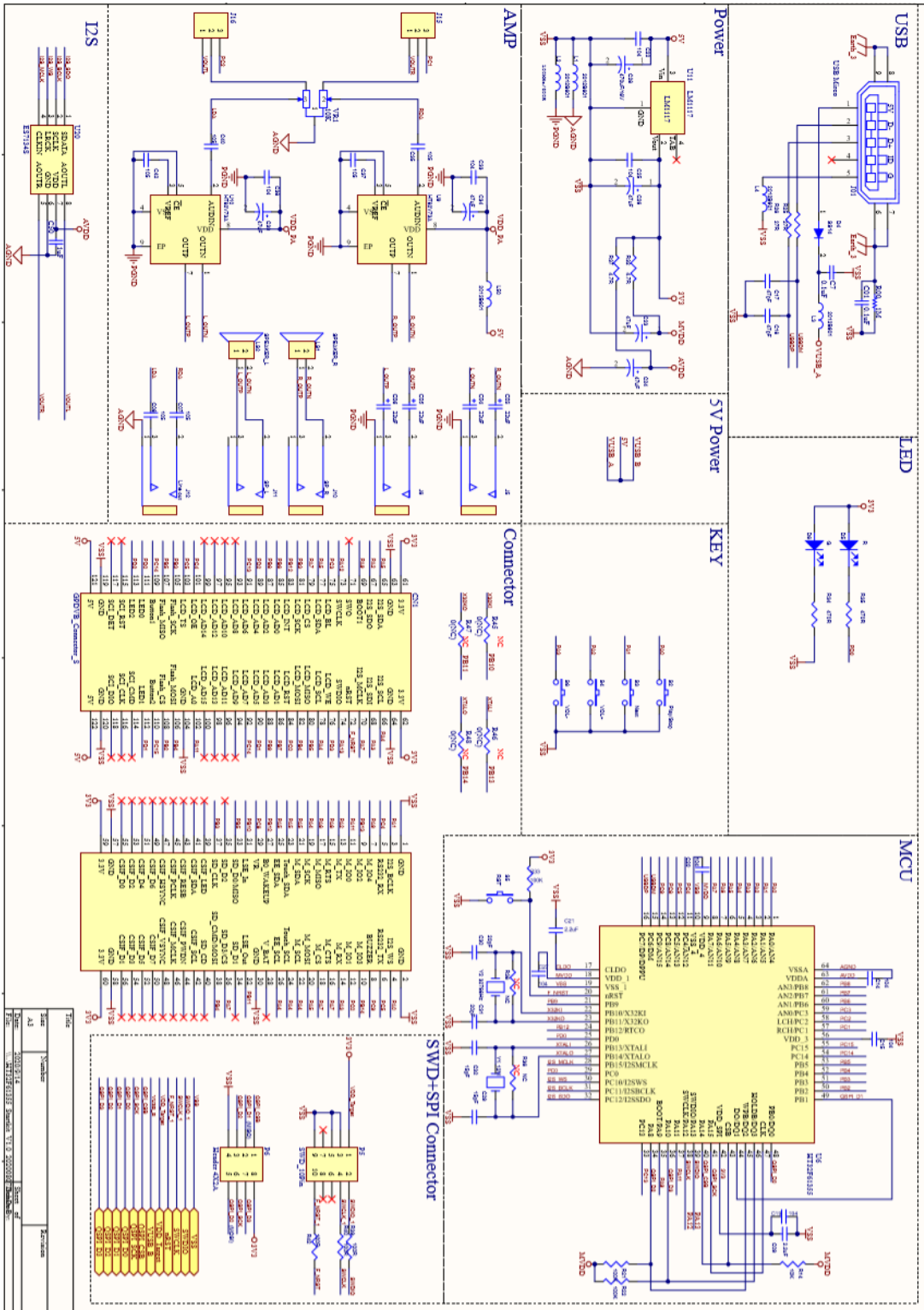
- ESK32-30615: HT32F61355 的开发板
- ESK32-30616: HT32F61356 的开发板
- ESK32-30617: HT32F61357/HT32F0006 的开发板
- ESK32-30605: HT32F61244/ HT32F61245 的开发板

ESK32-30615: HT32F61355 的开发板

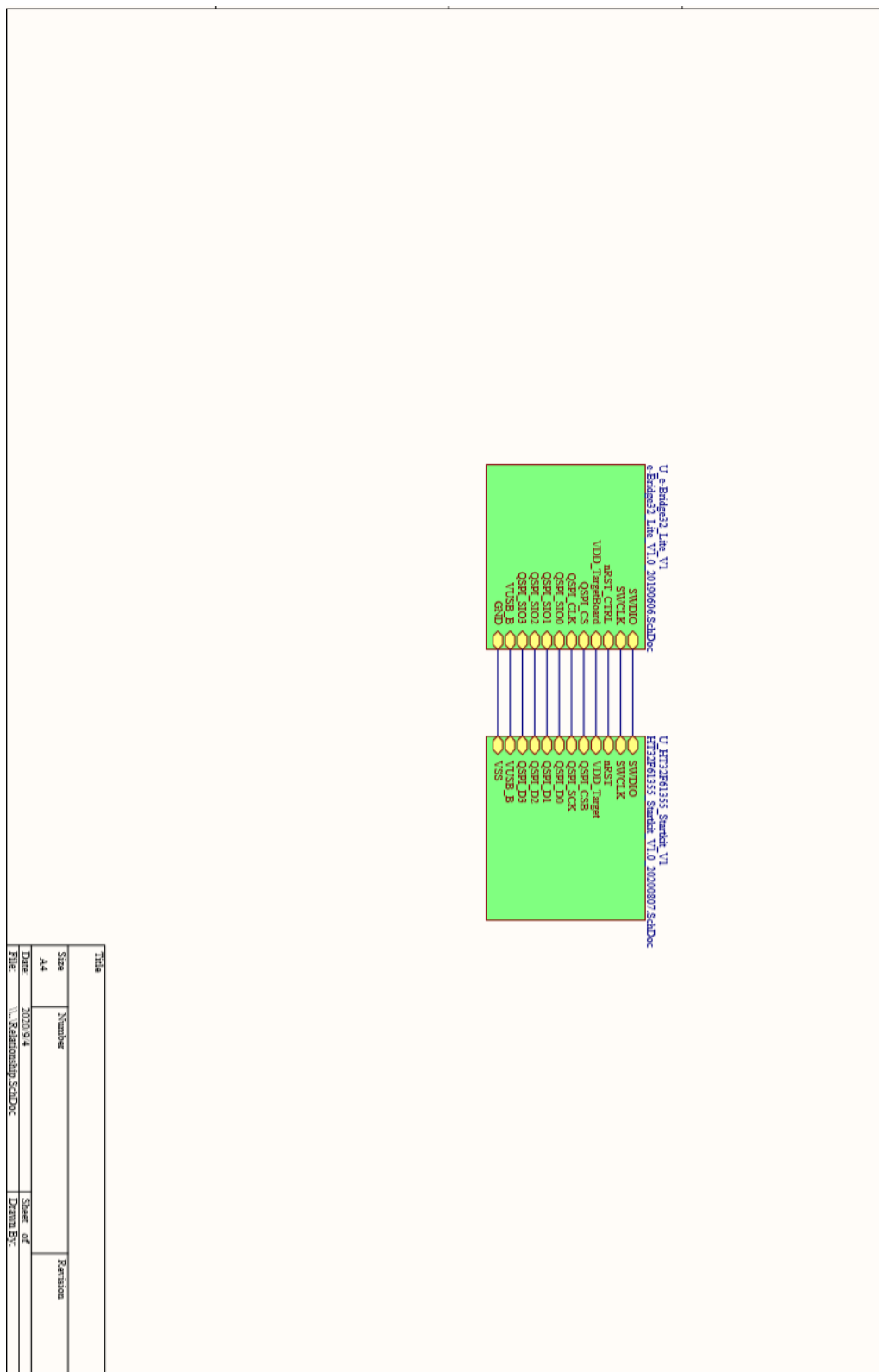
ESK32-30615 (1/3)



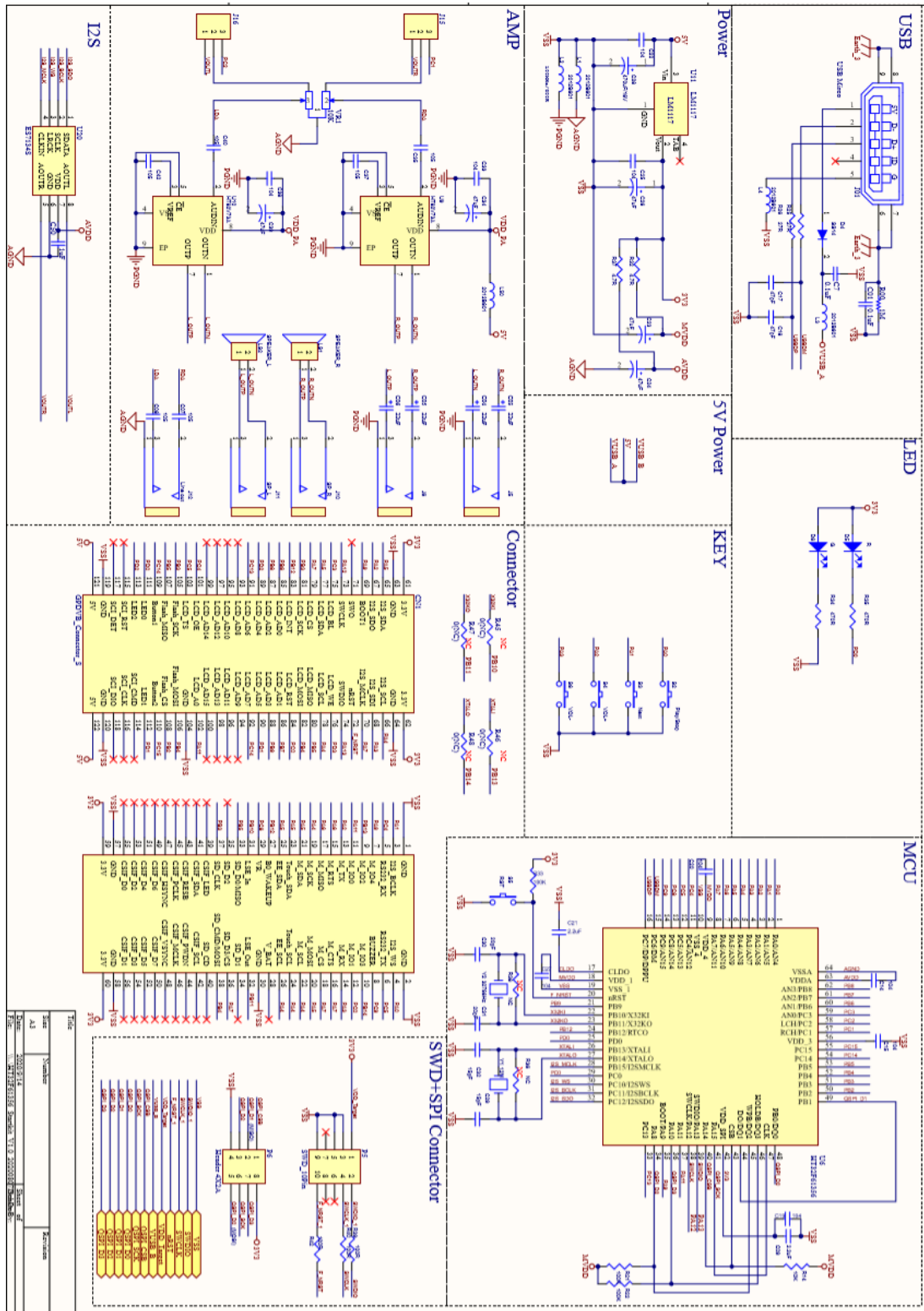
ESK32-30615 (2/3)



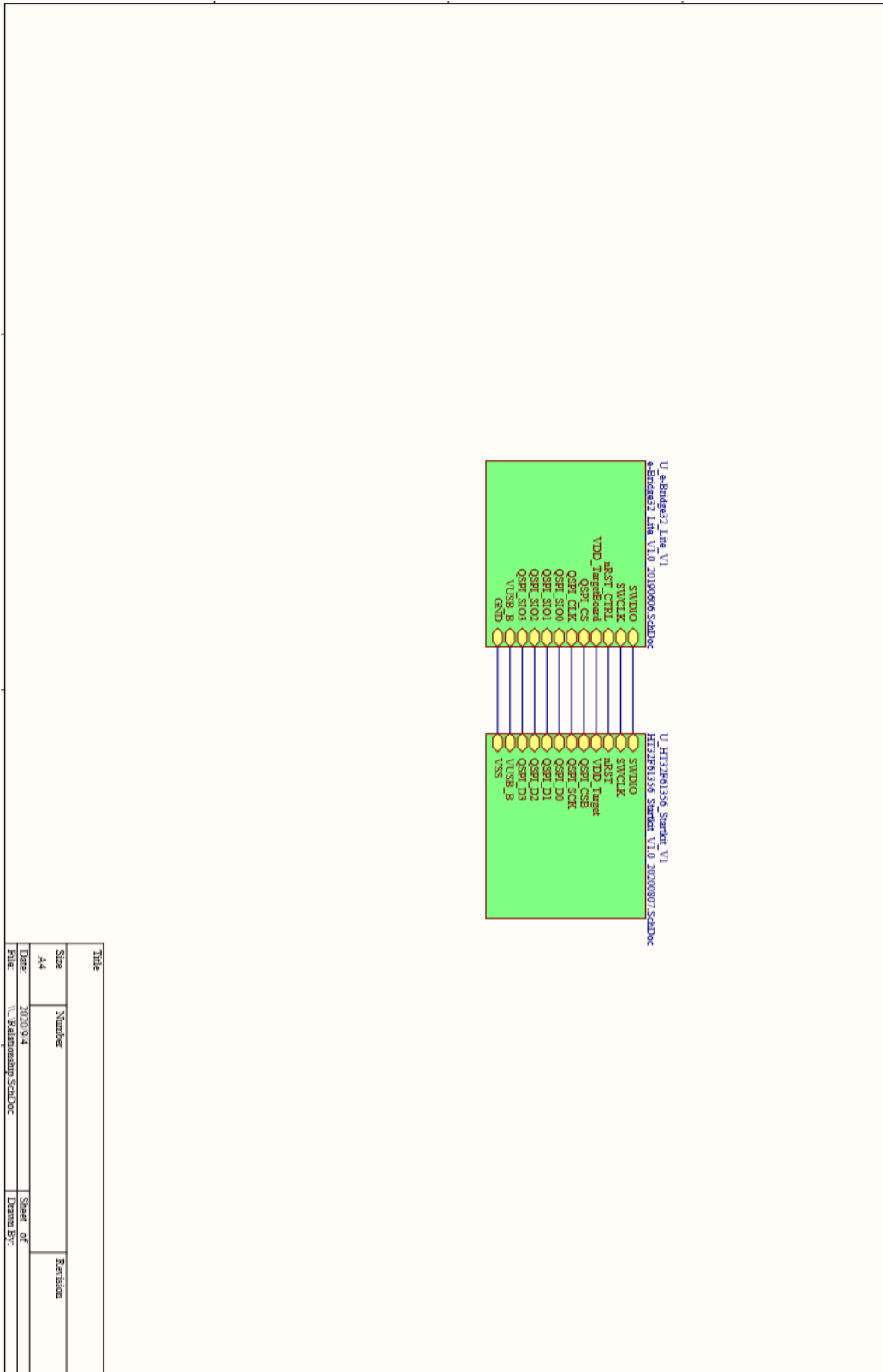
ESK32-30615 (3/3)



ESK32-30616 (2/3)

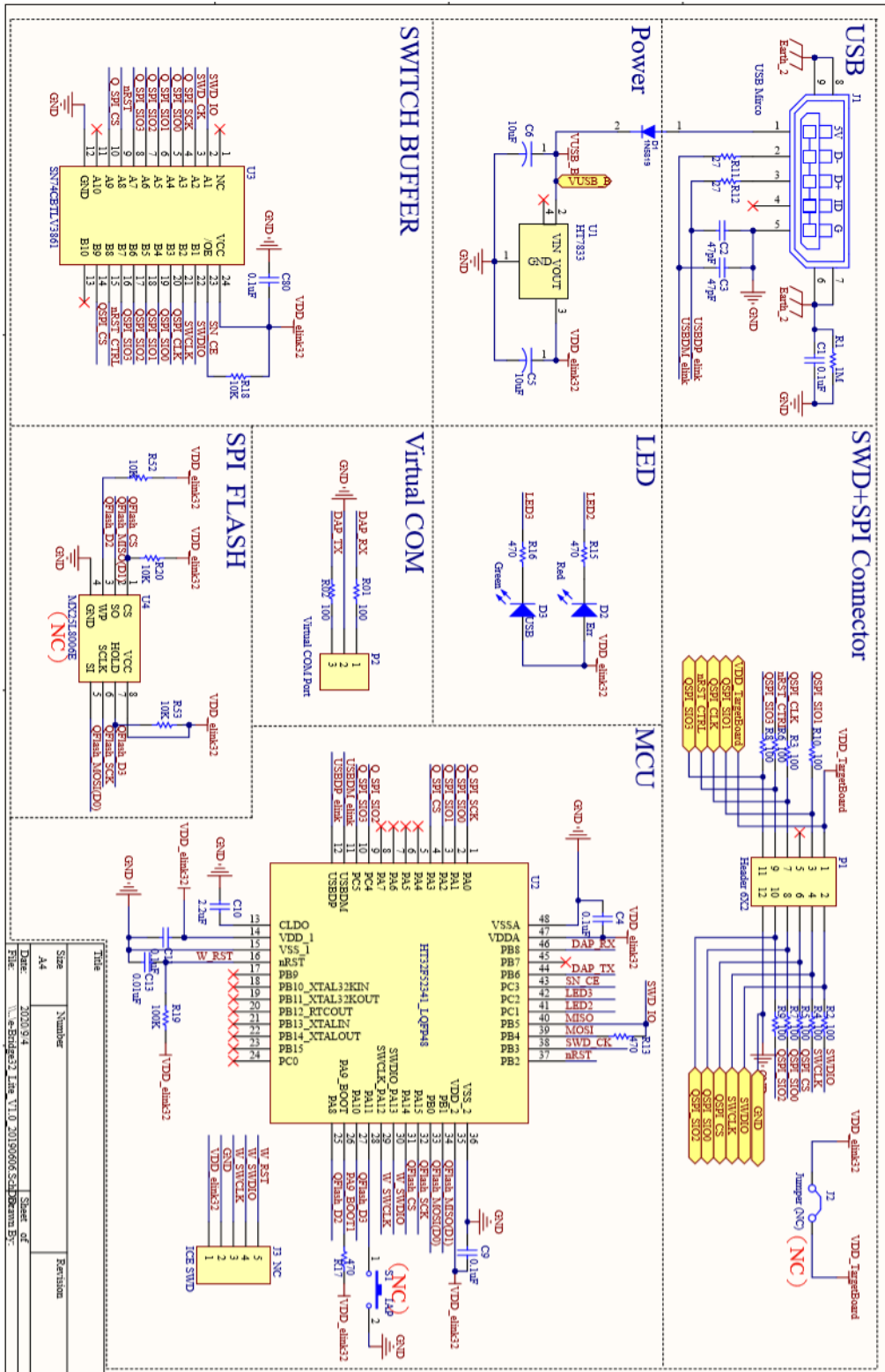


ESK32-30616 (3/3)

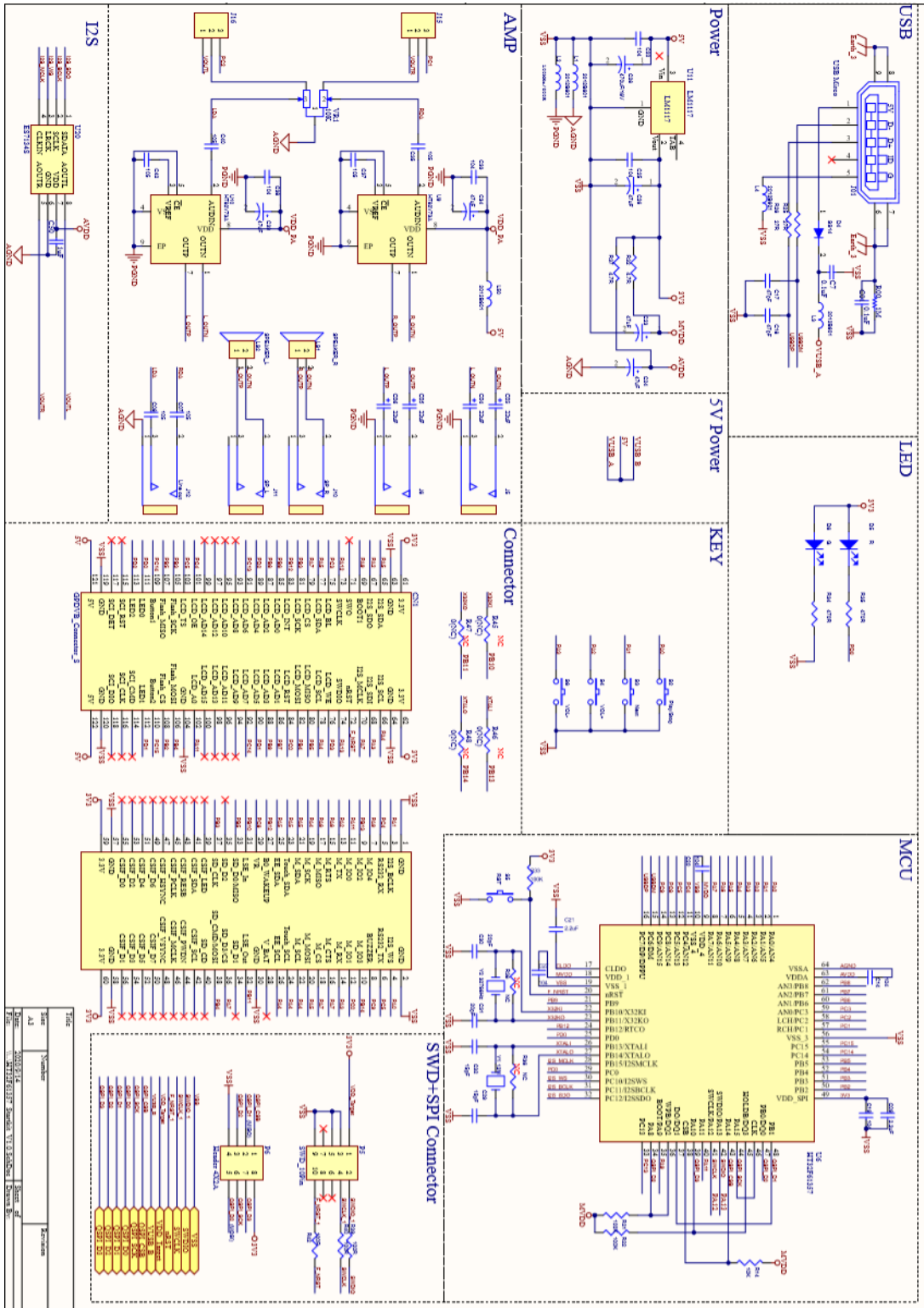


ESK32-30617: HT32F61357/HT32F0006 的开发板

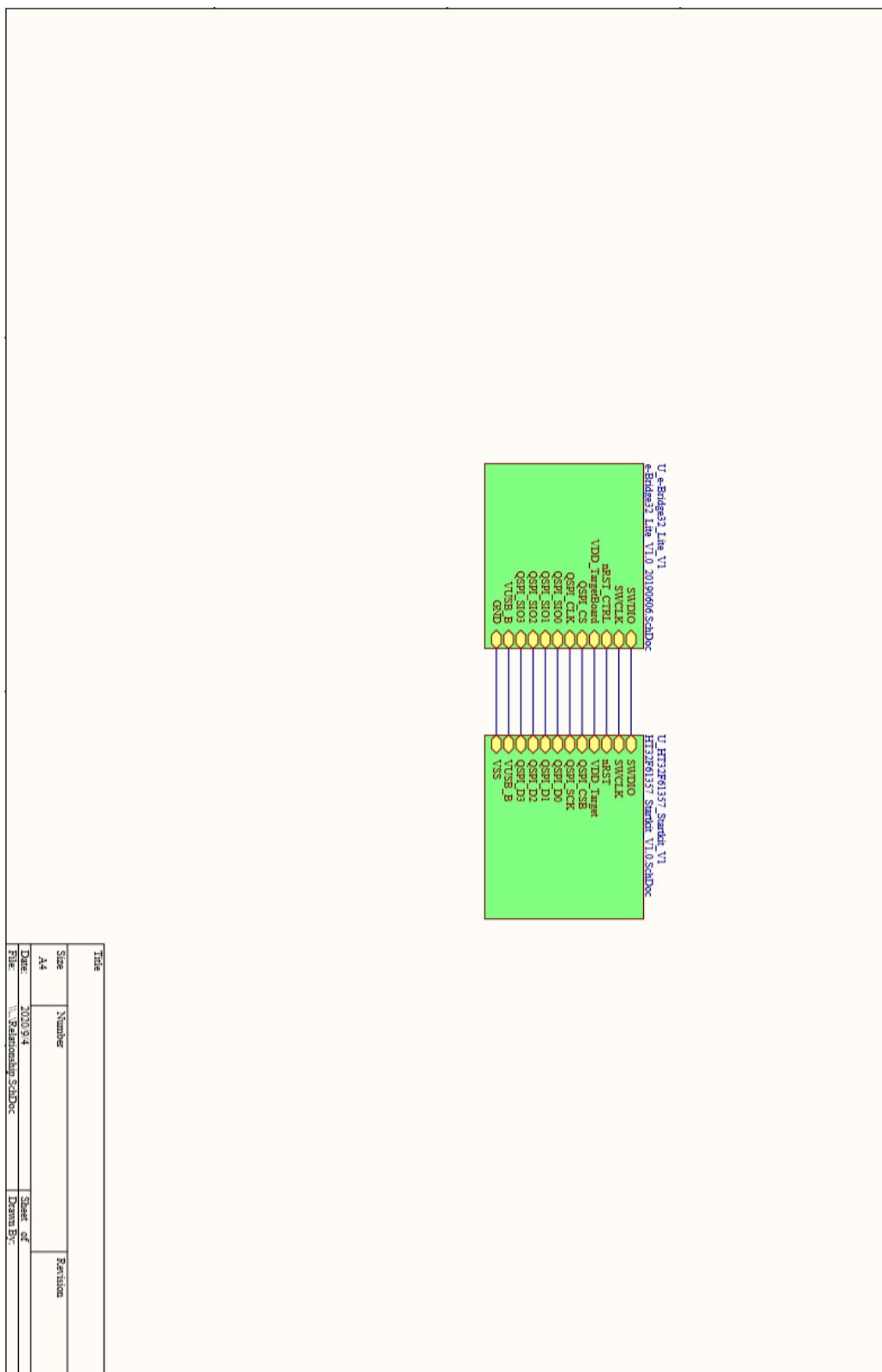
ESK32-30617 (1/3)



ESK32-30617 (2/3)

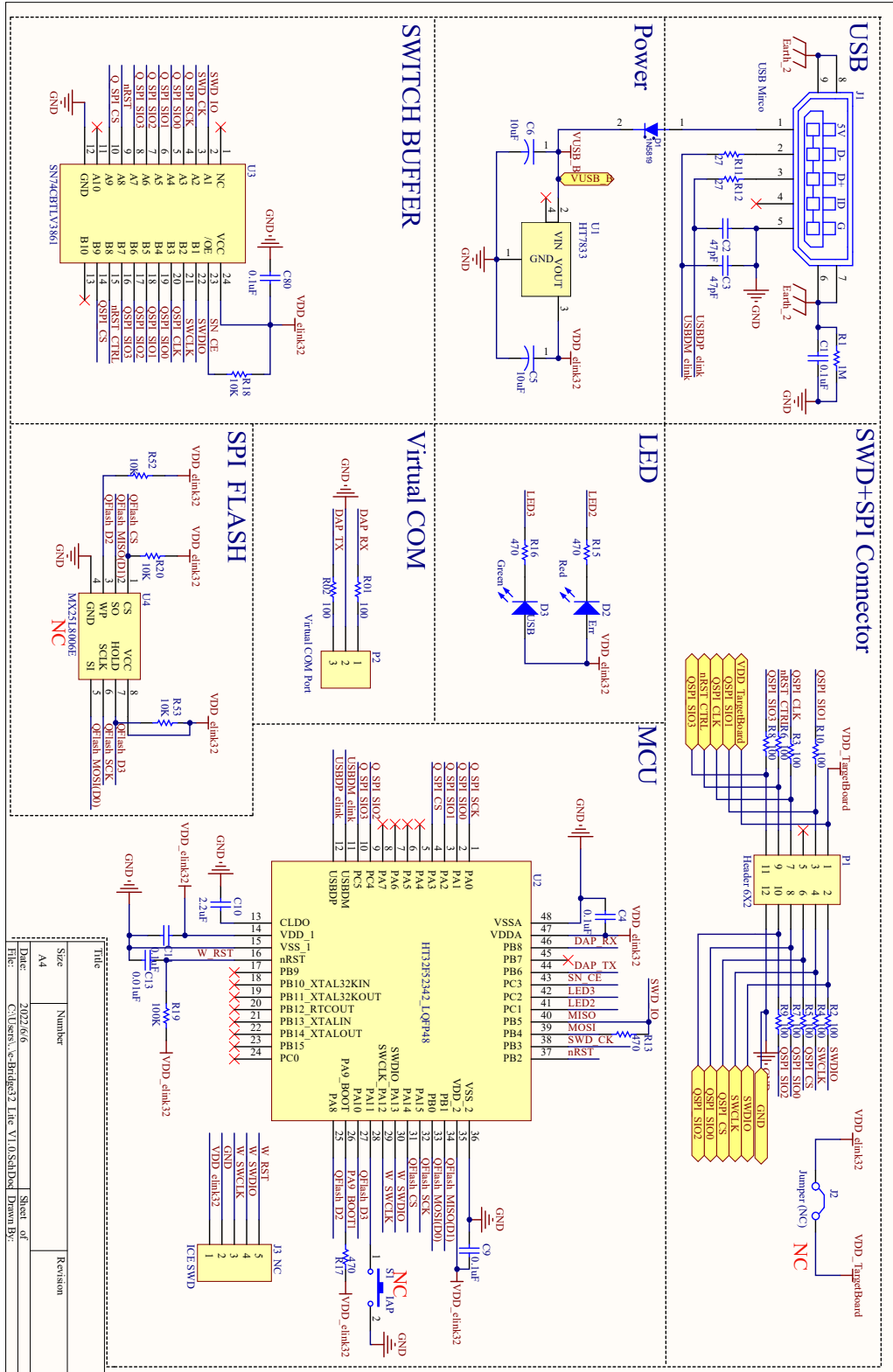


ESK32-30617 (3/3)

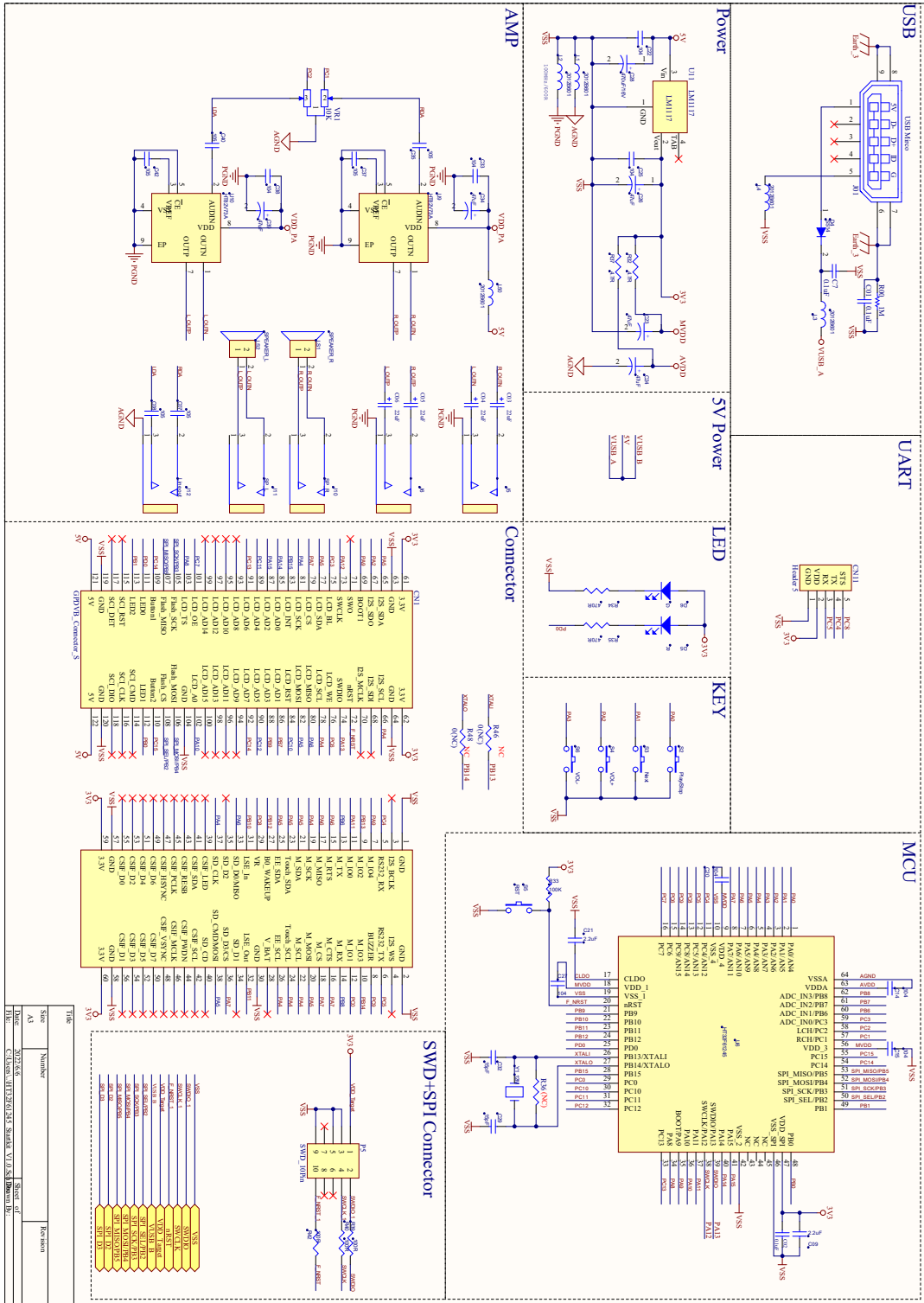


ESK32-30605: HT32F61244/ HT32F61245 的开发板

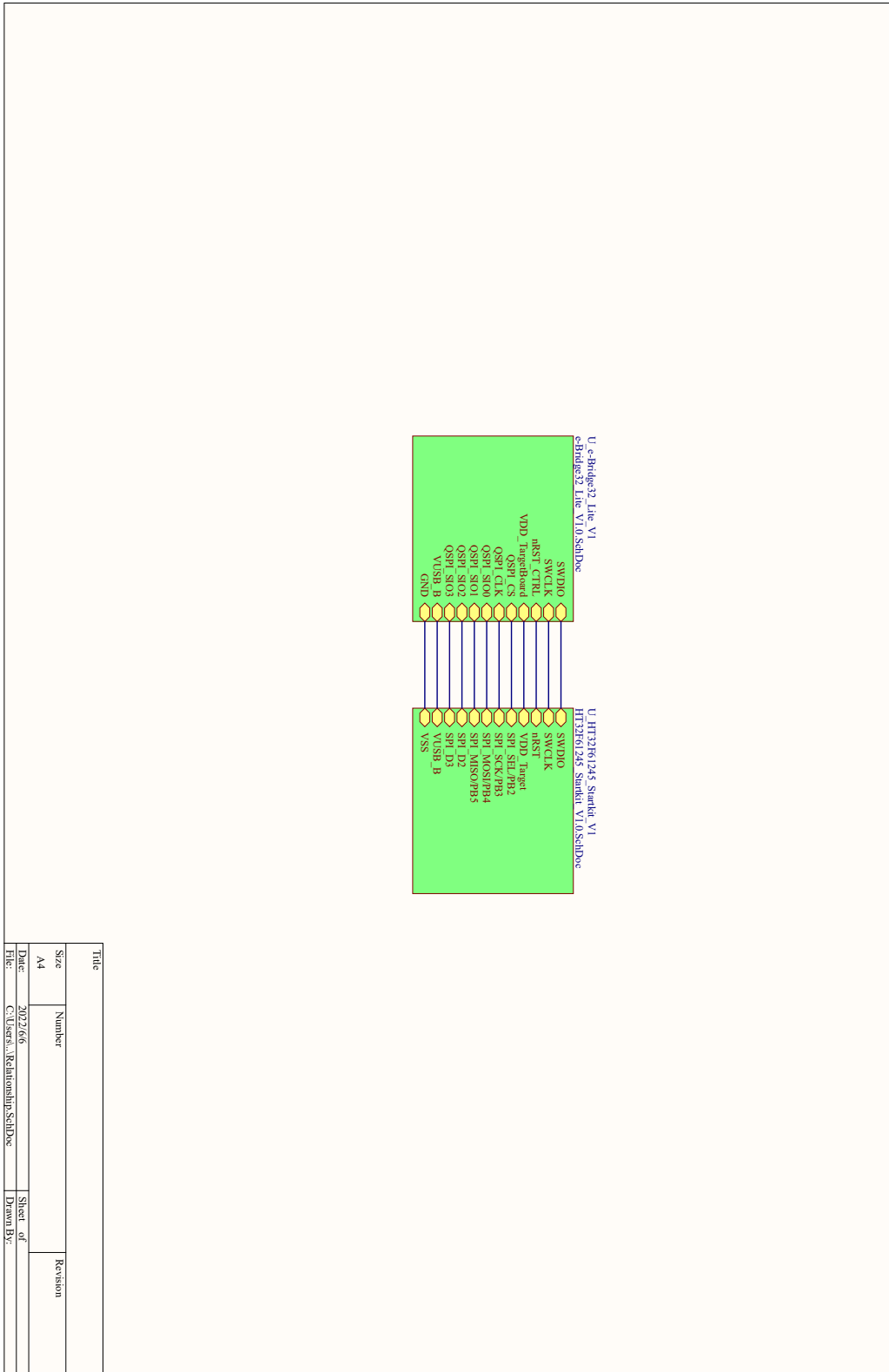
ESK32-30605 (1/3)



ESK32-30605 (2/3)



ESK32-30605 (3/3)



Title		Revision	
Size	Number	Sheet of	
A4		1 of	
Date:	2022/6/6	Drawn By:	
File:	C:\Users\... \Relationship_SchDoc		

Copyright© 2022 by HOLTEK SEMICONDUCTOR INC. All Rights Reserved.

本文件出版时 HOLTEK 已针对所载信息为合理注意，但不保证信息准确无误。文中提到的信息仅是提供作为参考，且可能被更新取代。HOLTEK 不担保任何明示、默示或法定的，包括但不限于适合商品化、令人满意的质量、规格、特性、功能与特定用途、不侵害第三方权利等保证责任。HOLTEK 就文中提到的信息及该信息之应用，不承担任何法律责任。此外，HOLTEK 并不推荐将 HOLTEK 的产品使用在会由于故障或其他原因而可能会对人身安全造成危害的地方。HOLTEK 特此声明，不授权将产品使用于救生、维生或安全关键零部件。在救生 / 维生或安全应用中使用 HOLTEK 产品的风险完全由买方承担，如因该等使用导致 HOLTEK 遭受损害、索赔、诉讼或产生费用，买方同意出面进行辩护、赔偿并使 HOLTEK 免受损害。HOLTEK (及其授权方，如适用) 拥有本文件所提供信息 (包括但不限于内容、数据、示例、材料、图形、商标) 的知识产权，且该信息受著作权法和其他知识产权法的保护。HOLTEK 在此并未明示或暗示授予任何知识产权。HOLTEK 拥有不事先通知而修改本文件所载信息的权利。如欲取得最新的信息，请与我们联系。