



LED Workshop 使用手册

版本 : V1.00 日期 : 2019-11-04

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、开启软件	5
三、语言选择	5
四、新建工程	6
4.1. 新建评估模式工程	6
4.2. 新建专业模式工程	15
五、库函数调用	18
5.1 HT16D31A/B & HT16D33A/B 调用方法	18
5.2 HT16D35A/B 调用方法	20

一、开发环境

1.1. 整体环境



1.2. 软件

包括 HT16D3xA/B LED Workshop、HT16D3xA/B LED Editor 和 IDE-3000。

1.2.1 HT16D3xA/B LED Workshop

用于实现主控 MCU 的选择、MCU 资源配置、按键配置、HT16D3xA/B 驱动引脚配置、外接闪存容量及引脚配置、排程设置、将 FLASHROM 数据烧录到开发板等功能。

1.2.2 HT16D3xA/B LED Editor

用于实现图形编辑和预览、HT16D3xA/B 自带的 MODE 设置等功能。

1.2.3 IDE-3000

用于编辑和查看源程序，通过 e-link 将程序和 PROM 数据下载到开发板中。

1.3. 硬件

包括 HOLTEK LED Workshop 配套开发板或用户自行开发的开发板、e-link (BICE000ELINK0B)。

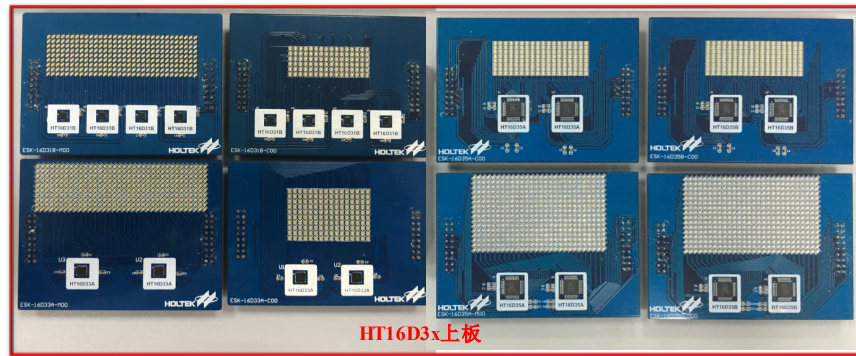
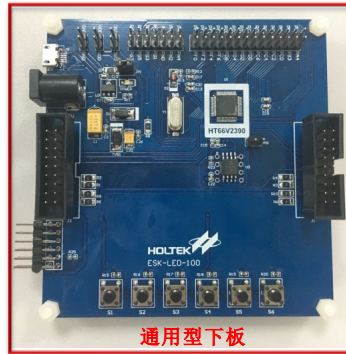
1.3.1 LED Workshop 配套开发板介绍

1.3.1.1 开发板类型

● 八块上板：

- ① ESK-16D31B-C00，选择 HT16D31B、COLOR 模式的时候使用；
- ② ESK-16D33A-C00，选择 HT16D33A、COLOR 模式的时候使用；
- ③ ESK-16D31B-M00，选择 HT16D31B、MONO 模式或者 GRAY 模式的时候使用；

- ④ ESK-16D33A-M00, 选择 HT16D33A、MONO 模式或者 GRAY 模式的时候使用;
- ⑤ ESK-16D35A-C00, 选择 HT16D35A、COLOR 模式的时候使用;
- ⑥ ESK-16D35B-C00, 选择 HT16D35B、COLOR 模式的时候使用;
- ⑦ ESK-16D35A-M00, 选择 HT16D35A、MONO 模式或者 GRAY 模式的时候使用;
- ⑧ ESK-16D35B-M00, 选择 HT16D35B、MONO 模式或者 GRAY 模式的时候使用;
- 一块通用型下板: ESK-LED-100

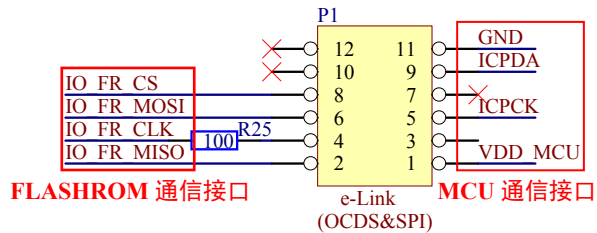


1.3.1.2 数据存储方式

- MCU 内置 PROM (主控 MCU: HT66F2390)
- 外接 FLASHROM (SST26VF032B)

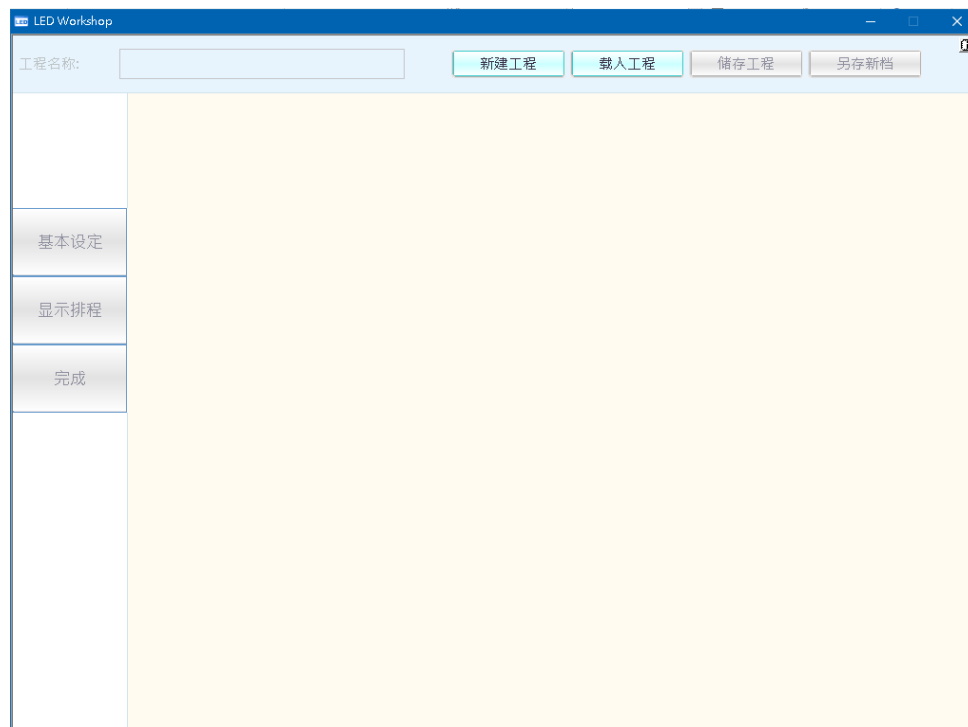
1.3.2 e-link 连接方式

用户可自行开发所需的开发板,但由于下载程序与数据的需要,与 e-link 连接的部分需要按照下图连接,才可以正确的进行烧录操作。



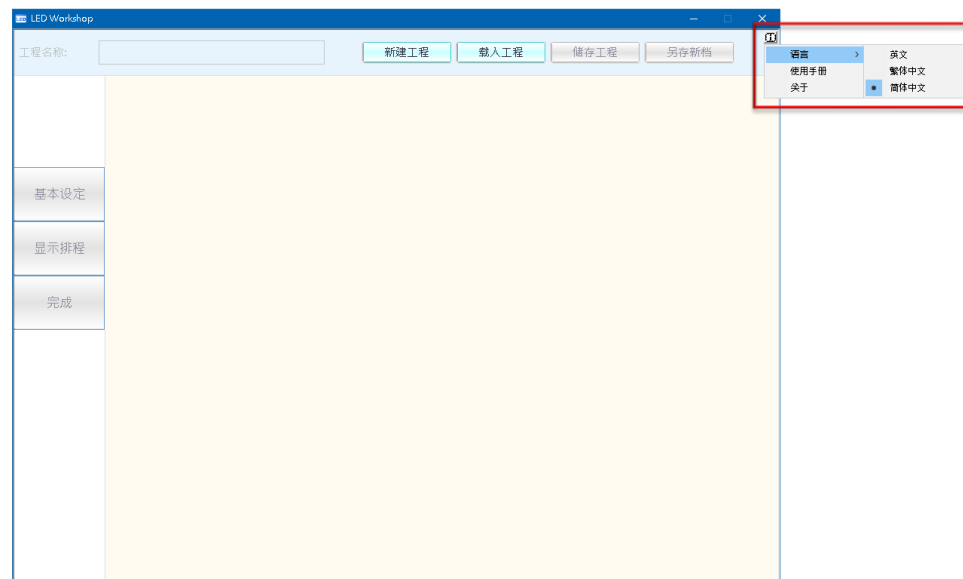
二、开启软件

双击图标  LED Workshop 可看到如下图所示的界面。



三、语言选择

单击界面右上角的图标 ，可以进行语言切换。



四、新建工程

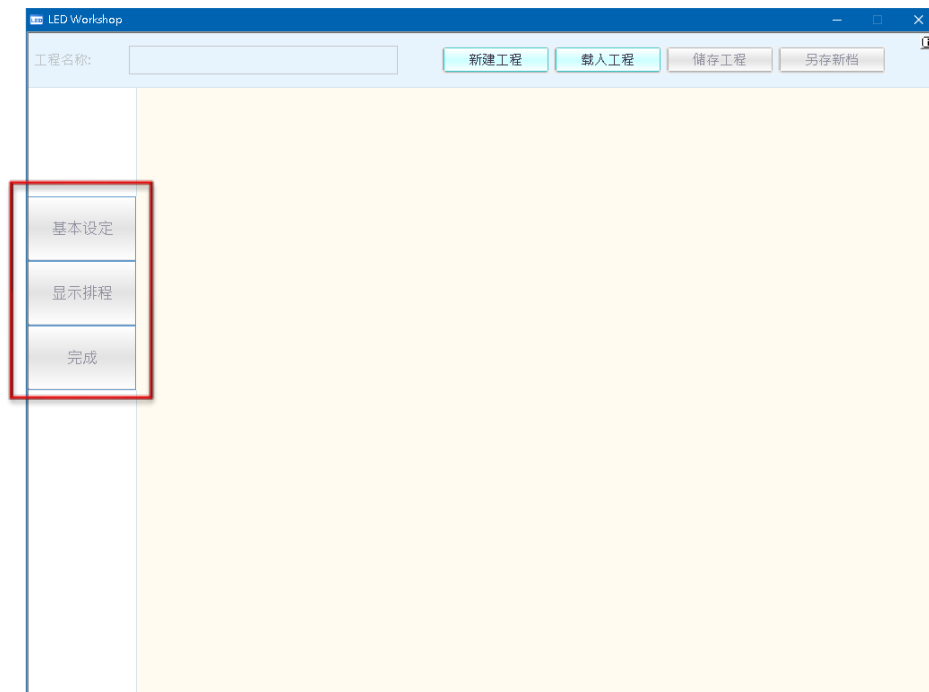
LED Workshop 有两种模式的工程：评估模式和专业模式。评估模式需搭配 Holtek 开发的 HT16D3xA/B EV Board 进行演示；专业模式客户可以自行开发电路板进行演示。下面介绍一下如何新建这两种模式。

4.1. 新建评估模式工程

连接好开发板和 e-link 后，打开 LED Workshop，在 LED Workshop 上单击“新建工程”图示，弹出新建工程对话框，根据接入的 EV Board 类型选择相应的内建模式，设置完成后单击“确定”按钮。



LED Workshop 主要分为三个操作步骤：基本设定、显示排程、完成。



4.1.1 基本设定

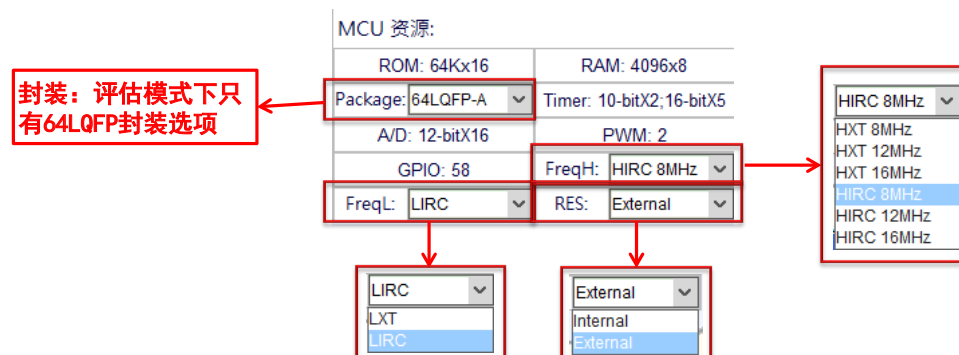
4.1.1.1 MCU 的选择

目前支持的 MCU 仅 HT66F2390 一颗。

可选用 MCU:			
HT66F2390			

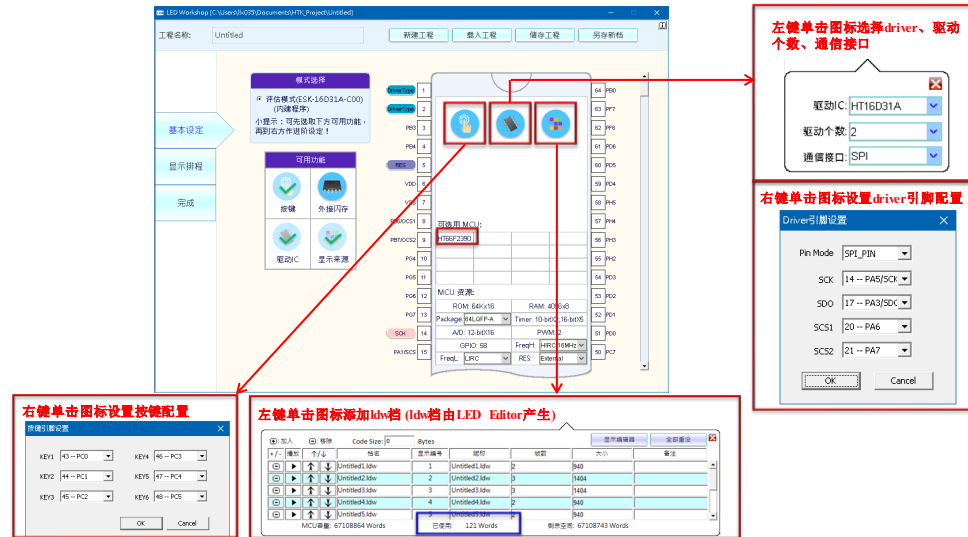
4.1.1.2 MCU 资源配置

MCU 资源可设置部分为封装 (Package)、高速振荡器 (FreqH)、低速振荡器 (FreqL) 和复位电路。



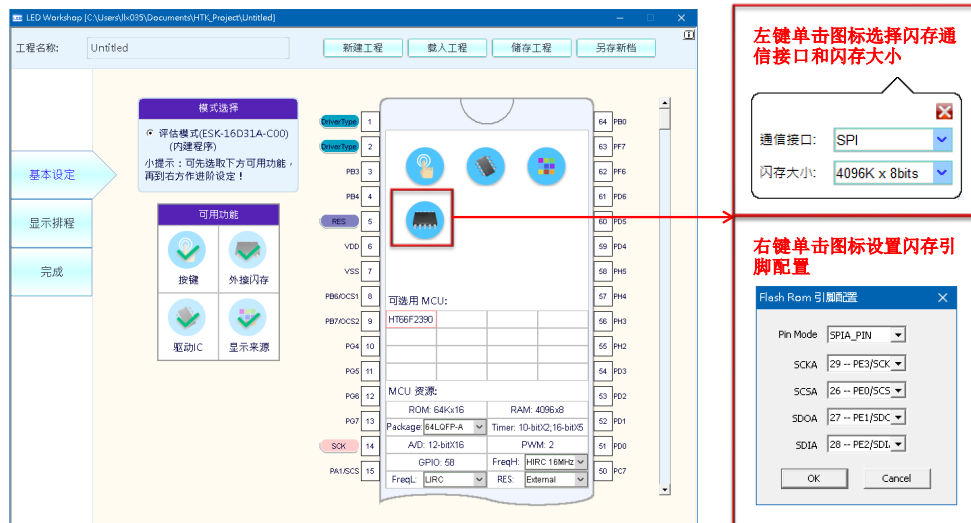
4.1.1.3 按键、驱动 IC、显示来源设置

点击“可选用 MCU”中的 HT66F2390，设置按键、驱动 IC、显示来源，这三项为必选项。当前接口为数据存放在 PROM 的模式，在这个模式下要注意空间已使用的情况（如下图蓝色框中所示），若超出只能选择外接闪存来存放显示数据。评估模式中选择开发板时已经固定按键引脚、driver 引脚及其通信接口引脚，所以无需重新选择。



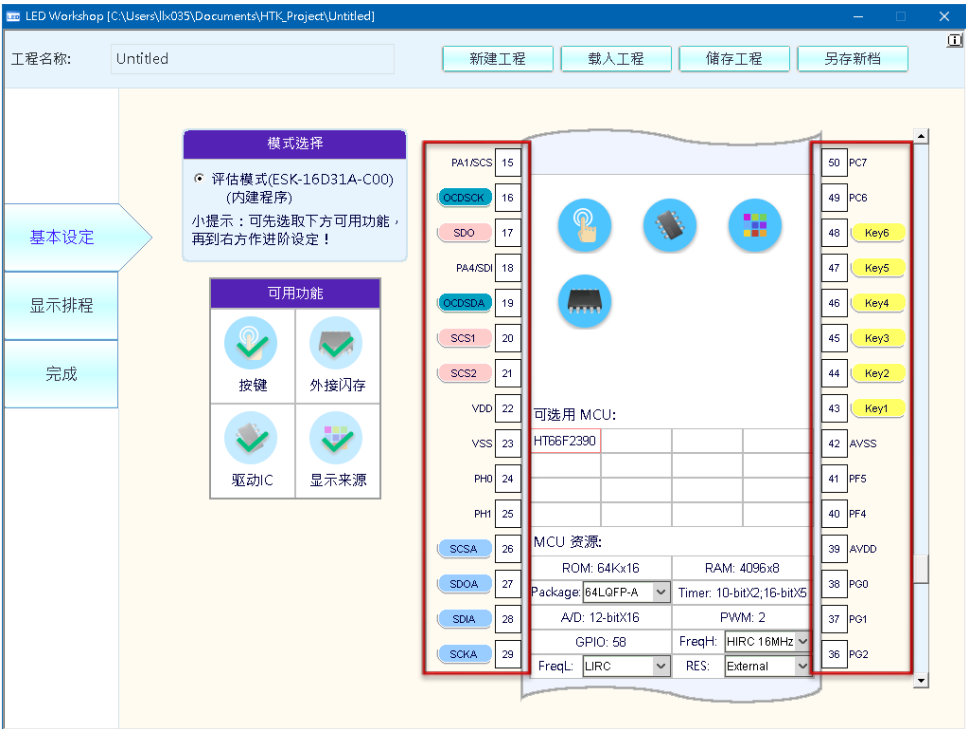
4.1.1.4 外接闪存设置

若要将数据存放在 FLASHROM 中，请点击“外接闪存”图标，并根据板子上焊接的 FLASHROM 大小选定闪存大小。



4.1.1.5 查看引脚配置

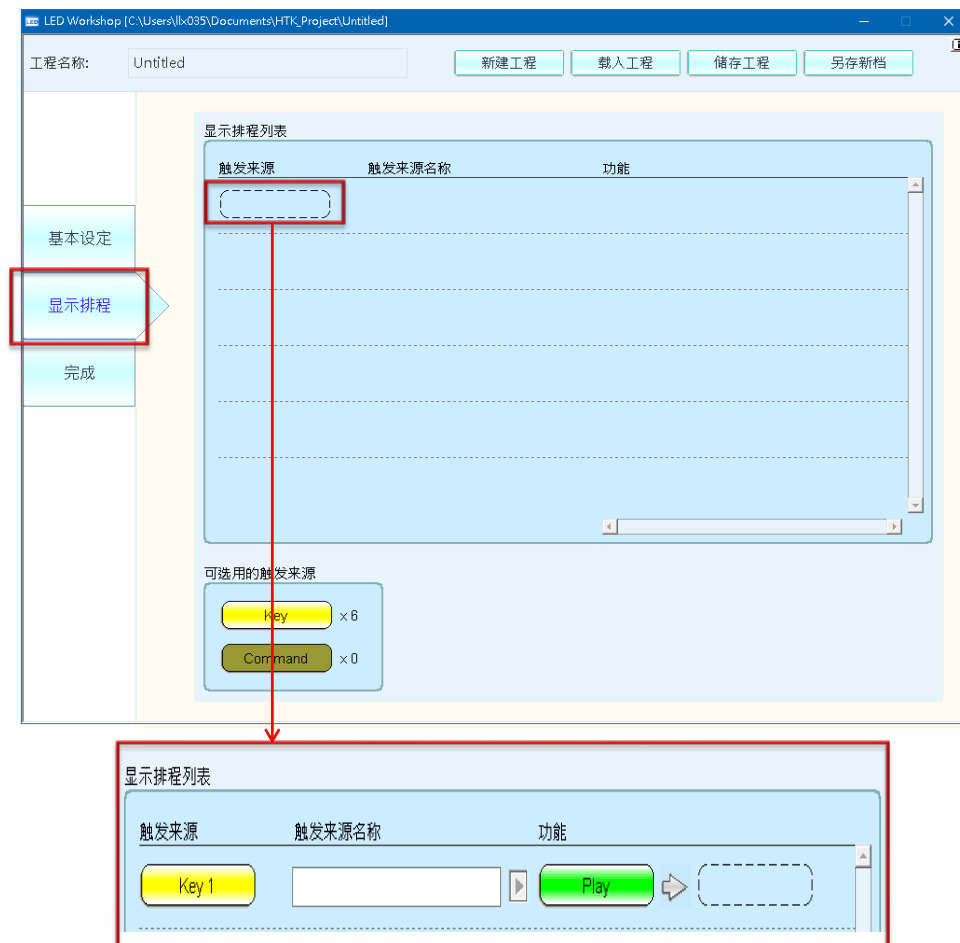
所有的引脚配置将体现在图示上，可通过上下拉动滑条进行查看。



4.1.2 显示排程

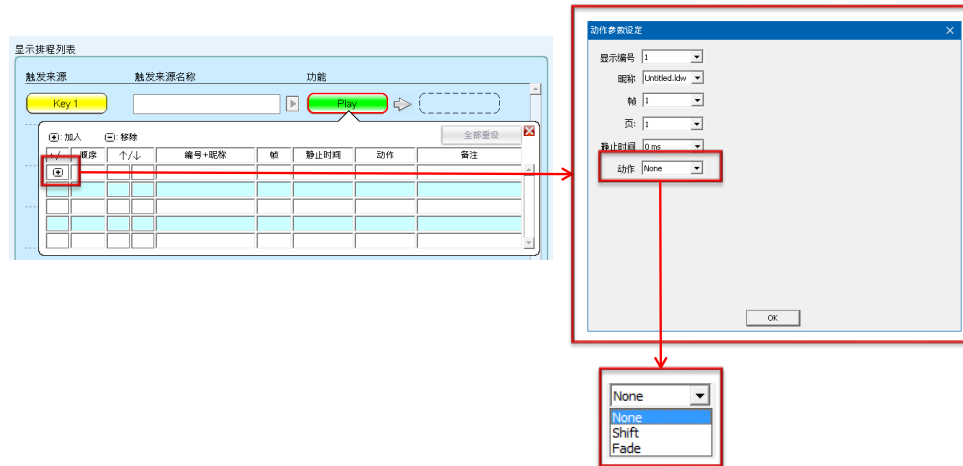
4.1.2.1 添加触发来源

点击左侧的“显示排程”进入排程设置。单击虚线框后，可用的触发来源会显示在虚线框中，评估模式中的触发来源为 Key。



4.1.2.2 添加显示及动作

点击“Play”按钮对功能进行设置，再点击“+”按钮添加显示及动作。

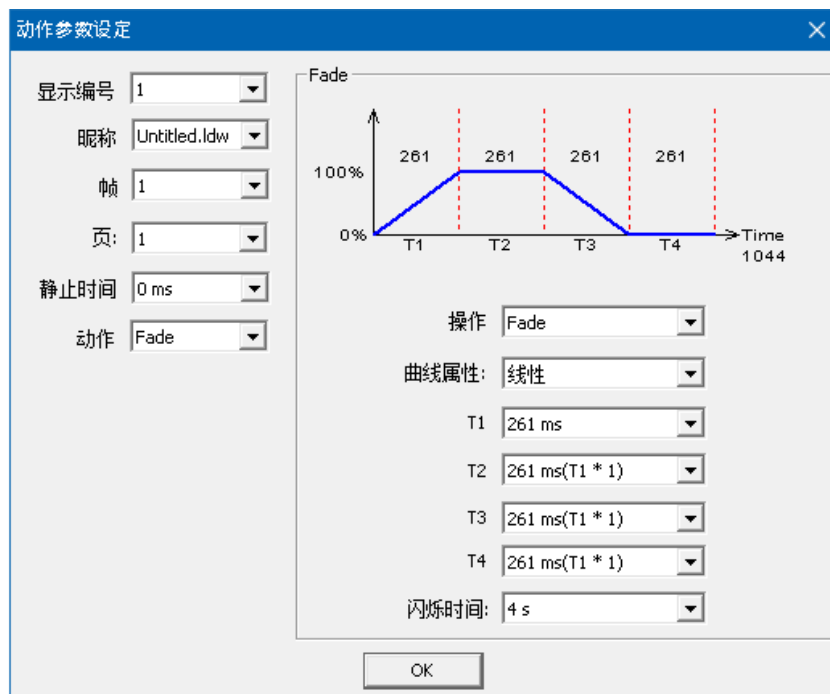
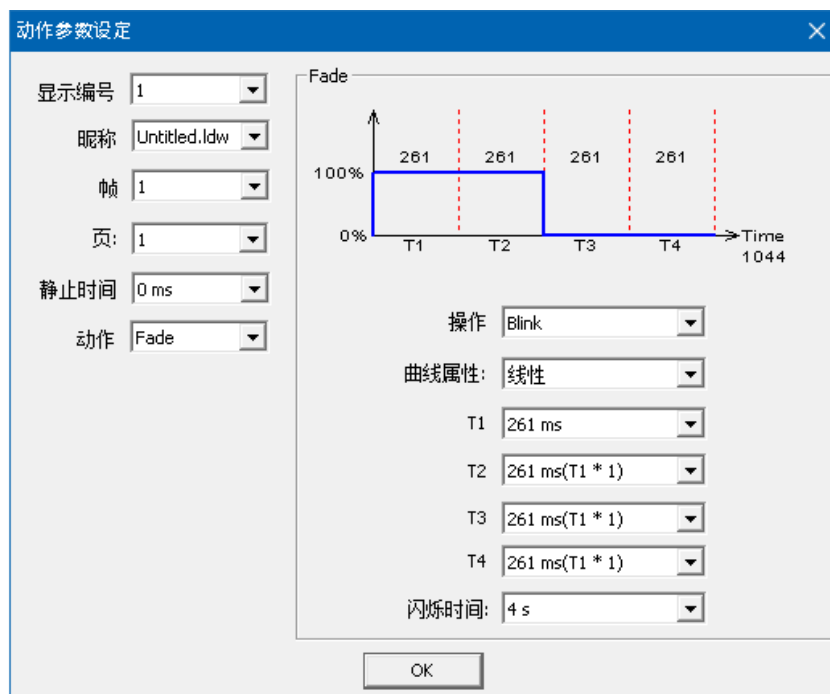


目前 workshop 支持三个动作：PLAY+NONE、PLAY+SHIFT、PLAY+FADE，分别由三个专用库函数控制。

- PLAY+NONE：作用为显示单一帧的画面及其 function。
- PLAY+SHIFT：作用为移动单一画面，支持左右方向的移动，还可以选择是否循环移动。（HT16D35A/B 支持上下左右移动）。

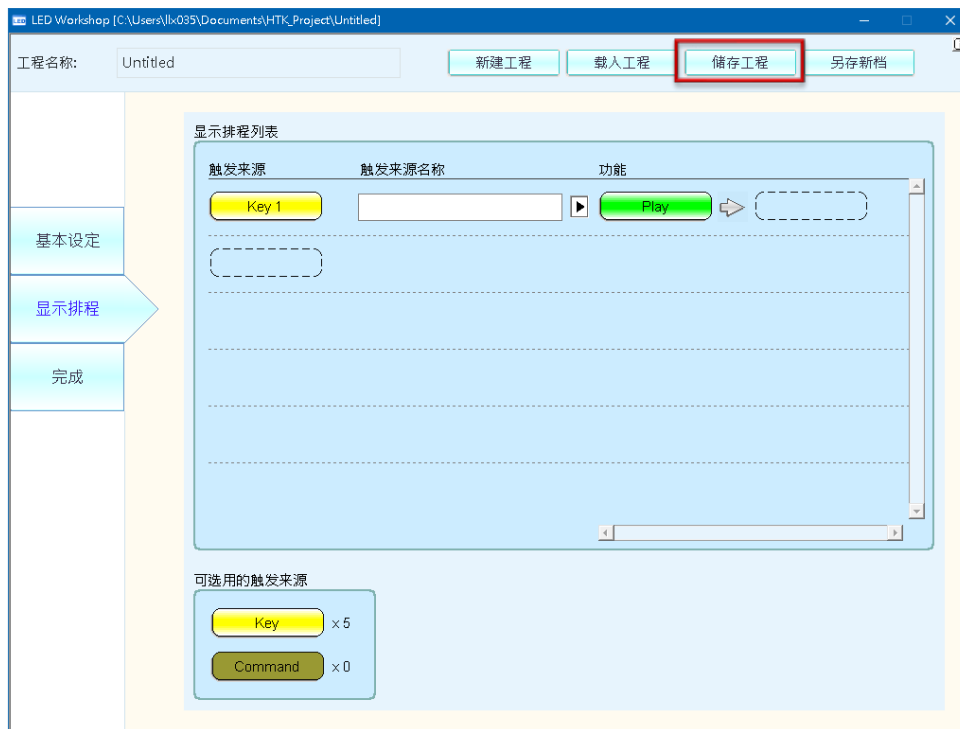


- PLAY+FADE: 支持两个操作，分别为 Blink 和 Fade; T1~T4 可自由设置。两个操作产生的波形如下图所示。



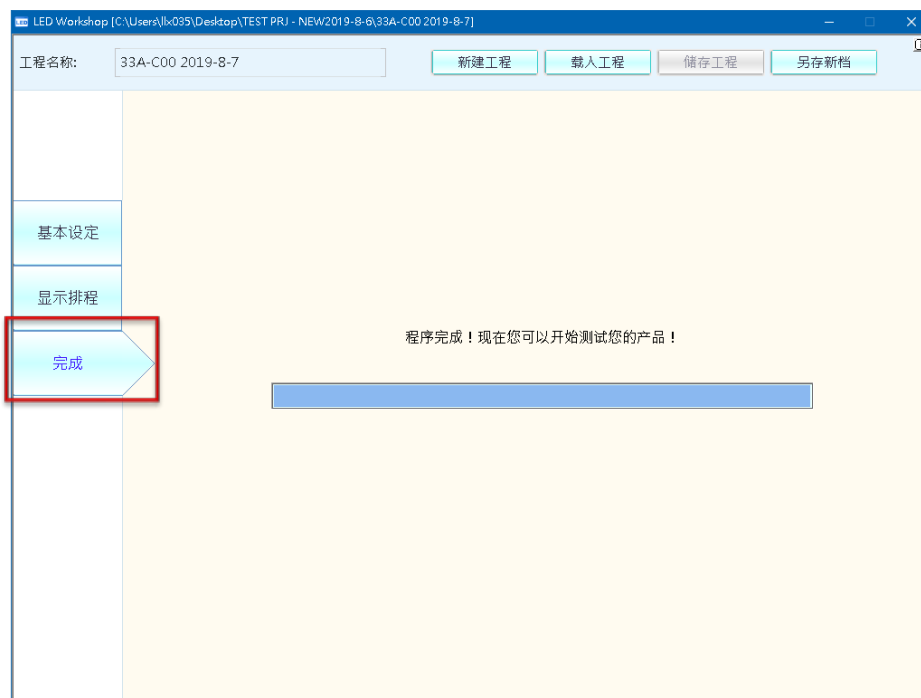
4.1.2.3 储存工程

设置完成后点击“储存工程”进行保存。保存的格式为 *.pjlw。



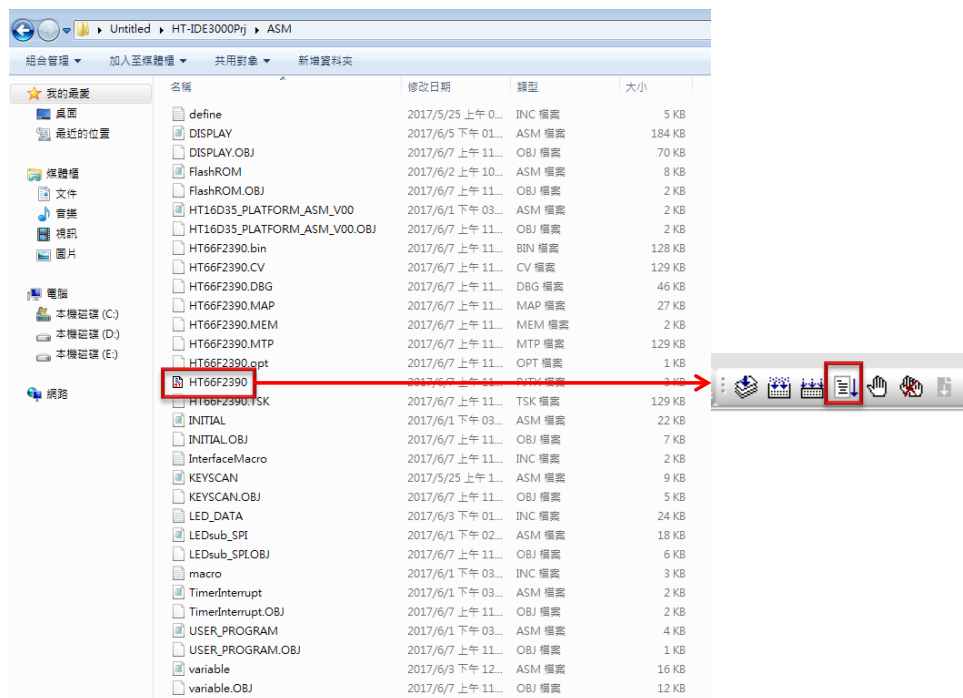
4.1.3 完成

点击左侧的“完成”进入生成显示数据、IDE 工程等步骤。

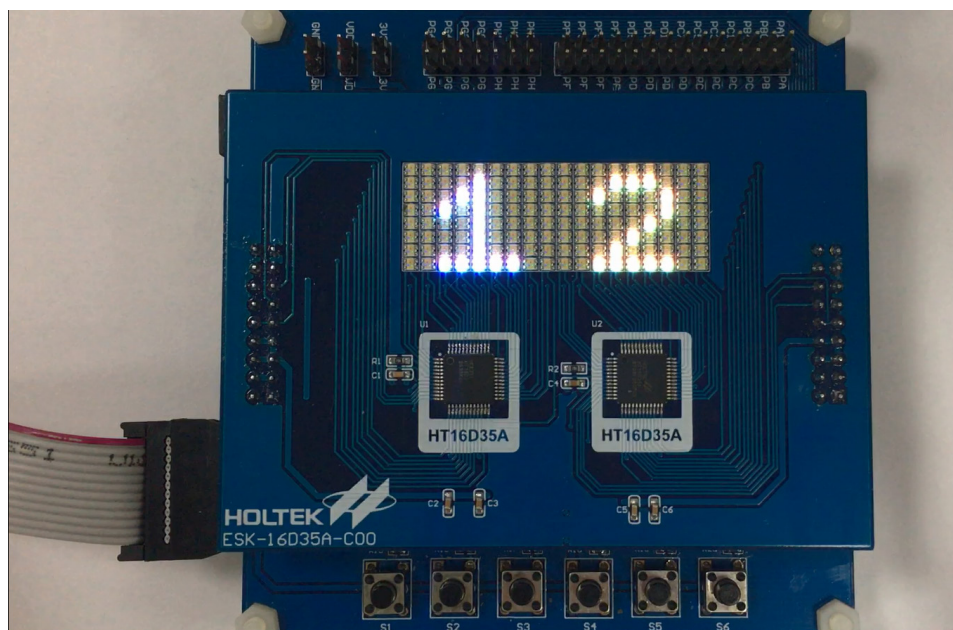


4.1.4 打开生成的 IDE 工程

在路径为：workshop 工程名 \ HT-IDE3000Prj 中的 C 或 ASM 文件夹中打开生成的 IDE 文件，使用 e-link 连接好开发板，点击“执行”图标，便可在开发板上观察到排程中设置的画面。

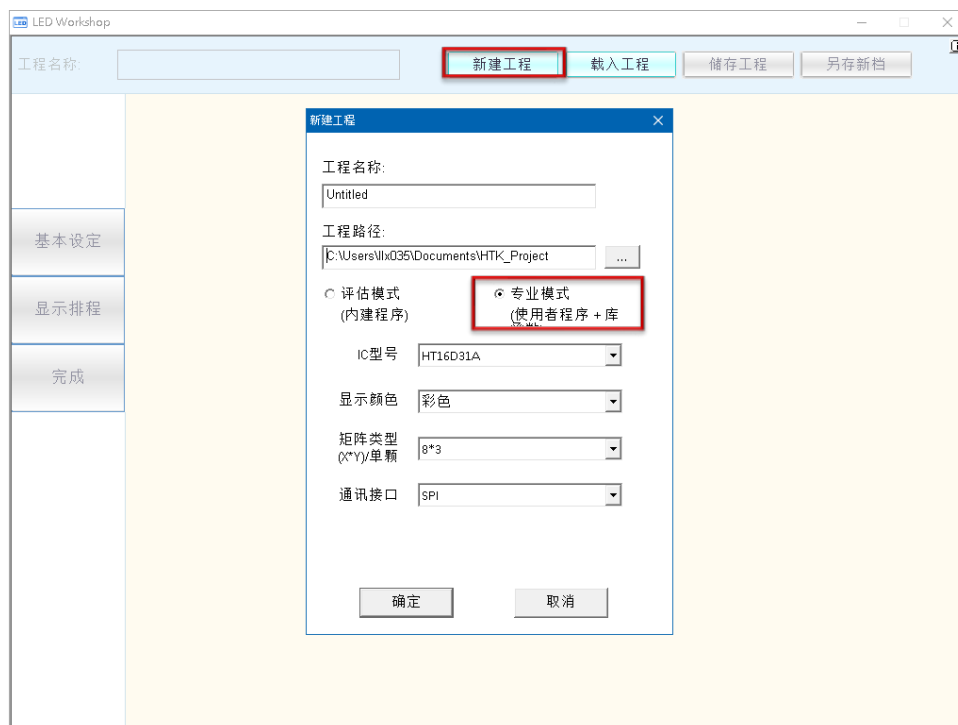


4.1.5 观察画面



4.2. 新建专业模式工程

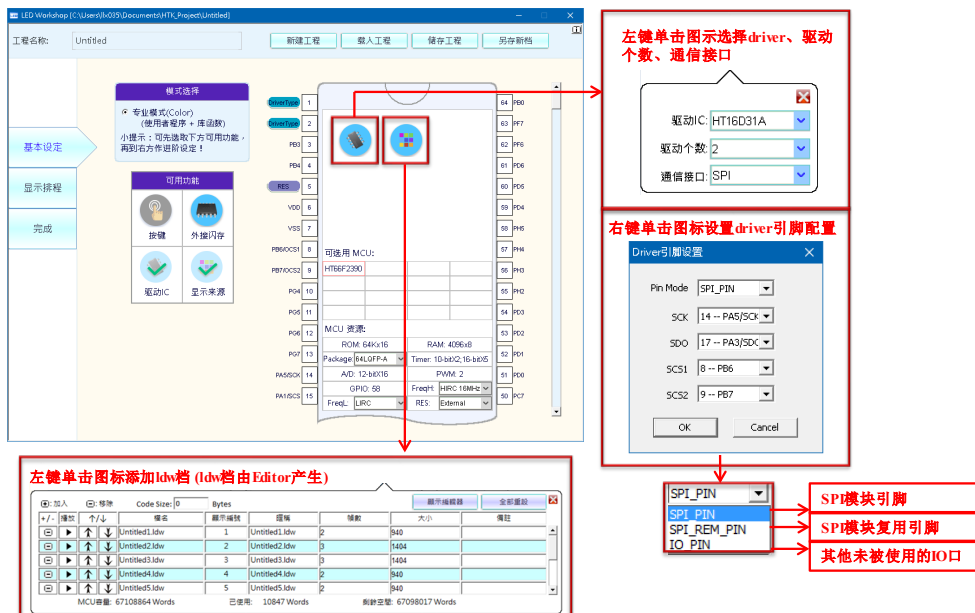
连接好开发板和 e-link 后，打开 LED Workshop，在 Workshop 上单击“新建工程”图标，弹出新建工程对话框，设置完成后单击“确定”按钮。



4.2.1 基本设定

4.2.1.1 驱动 IC、显示来源

点击“可选用 MCU”中的 HT66F2390，设置驱动 IC、显示来源，这两项为必选项。当前界面为数据存放在 PROM 的模式。



左键单击图示选择driver、驱动个数、通信接口

驱动 IC: HT16D31A
驱动个数: 2
通信接口: SPI

右键单击图示设置driver引脚配置

Driver引脚配置

Pin Mode: SPI_PIN
SCK: 14 -- PA5/SCK
SDO: 17 -- PA3/SDO
SCS1: 8 -- PB6
SCS2: 9 -- PB7

左键单击图标添加ldw档 (ldw档由Editor产生)

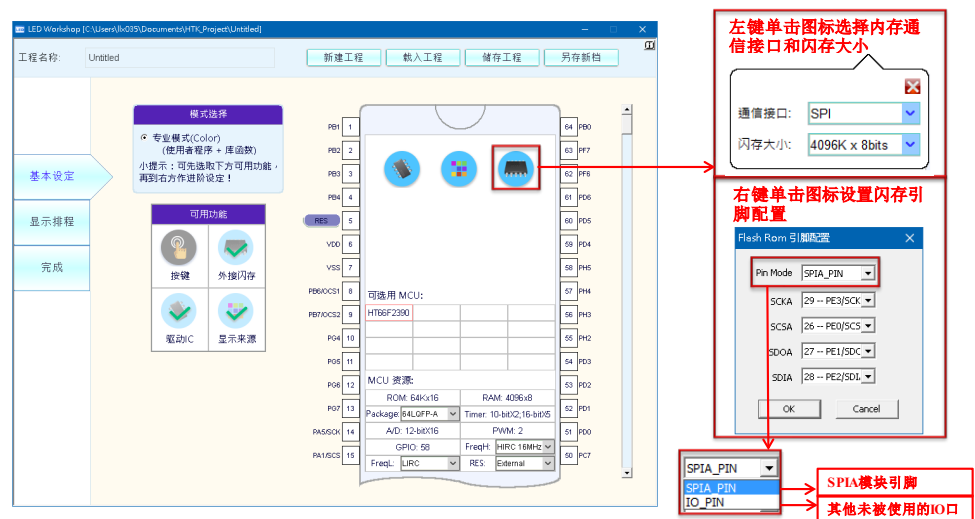
名称	Code Size	Bytes	地址	大小	备注
Untitled1.ldw	1	1	0x0000	1	
Untitled2.ldw	2	2	0x0001	2	
Untitled3.ldw	3	3	0x0002	3	
Untitled4.ldw	4	4	0x0003	4	
Untitled5.ldw	5	5	0x0004	5	

MCU选择: HT66F2390
ROM: 64Kx16
RAM: 4096x8
Package: B4LQFP-A
Timer: 10-bit/2.16-bit/5
A/D: 12-bit/16
PWM: 2
GPIO: 68
Freq: 16MHz
LIRC: External

SPi模块引脚
SPi模块复用引脚
其他未被使用的IO口

4.2.1.2 外接闪存设置

若要将数据存放在 FLASHROM 中，请点击“外接闪存”图标，并根据板子上焊接的 FLASHROM 大小选定闪存大小。



左键单击图标选择内存通信接口和闪存大小

通信接口: SPI
闪存大小: 4096K x 8bits

右键单击图标设置闪存引脚配置

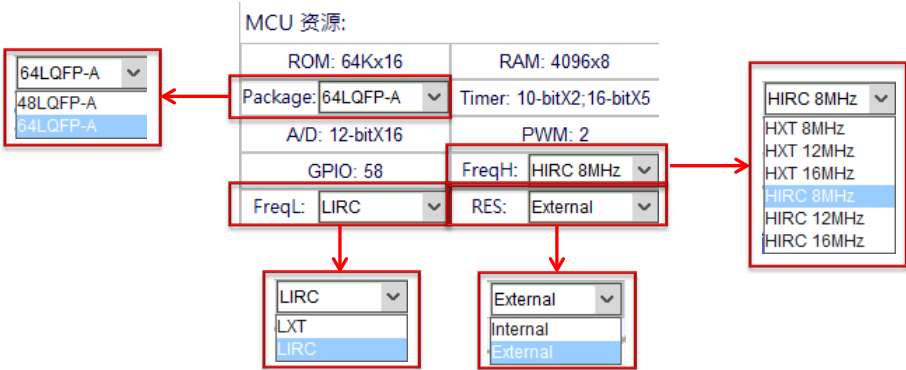
Flash Ram 引脚配置

Pin Mode: SPIA_PIN
SCKA: 29 -- PE3/SCK
SCSA: 26 -- PE0/SCS
SDOA: 27 -- PE1/SDO
SDIA: 28 -- PE2/SDI

SPiA模块引脚
其他未被使用的IO口

4.2.1.3 MCU 资源配置

MCU 资源可设置部分为封装 (Package)、高速振荡器 (FreqH)、低速振荡器 (FreqL) 和复位电路。



4.2.2 显示排程

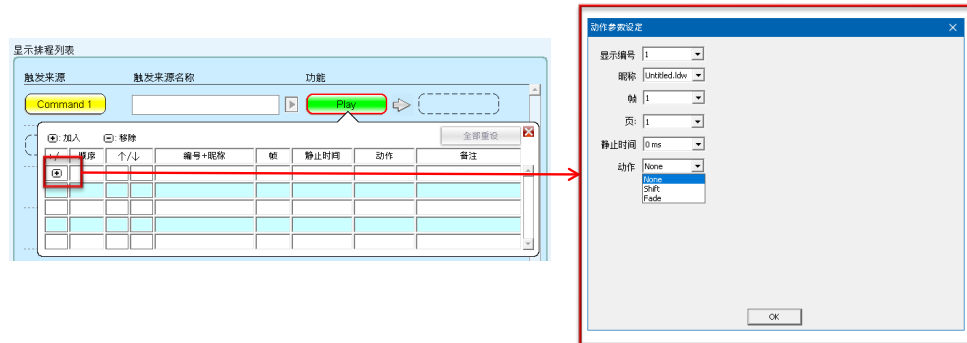
4.2.2.1 添加触发来源

点击左侧的“显示排程”进入排程设置。单击虚线框后，可用的触发来源会显示在虚线框中，专业模式中的触发来源为 Command。



4.2.2.2 添加显示及动作

点击“Play”按钮对功能进行设置，再点击“+”按钮添加显示及动作。专业模式同样支持 PLAY+NONE、PLAY+SHIFT、PLAY+FADE 这三个动作。后面的操作与评估模式一样，这里不再赘述。



五、库函数调用

前面提到 PLAY+NONE、PLAY+SHIFT、PLAY+FADE 这三个动作，分别由三个专用库函数控制，下面来介绍一下这三个库函数该如何调用。目前的库函数仅支持两颗 HT16D3xA/B 同时驱动的情况下使用。

5.1 HT16D31A/B & HT16D33A/B 调用方法

5.1.1 PLAY+NONE

```
void _PLAY_PICTURE_DISPLAY(INT8U LU8V_LDW_INDEX,INT8U LU8V_FRAME_INDEX,INT8U LU8V_PAGE_INDEX)
```

入口参数：

- ① LU8V_LDW_INDEX：在添加的 LDW 档中选择第几个 LDW 档；
- ② LU8V_FRAME_INDEX：在选择的 LDW 档中选择第几帧；
- ③ LU8V_PAGE_INDEX：在选择的帧中选择第几页显示。

调用方式：

```
_PLAY_PICTURE_DISPLAY(INT8U LU8V_LDW_INDEX,INT8U LU8V_FRAME_INDEX,INT8U LU8V_PAGE_INDEX)
```

说明：调用此函数时需对函数的入口参数进行赋值，如示例中的 LU8V_LDW_INDEX、LU8V_FRAME_INDEX 和 LU8V_PAGE_INDEX，需要先给这三个变量赋值。例如 LU8V_LDW_INDEX=2，LU8V_FRAME_INDEX=1，LU8V_PAGE_INDEX=1 时调用函数，实现第 2 个 LDW 档第 1 帧画面的第一页显示。

5.1.2 PLAY+SHIFT

```
void _PLAY_PICTURE_SHIFT_SETTING(INT8U DIRECTION,INT8U SPEED,INT8U SHIFT_NUM,INT8U SHIFT_LOOP,INT8U ROTATE)
```

入口参数:

- ① DIRECTION: 画面移动方向
=0, 右移
=1, 左移
- ② SPEED: 画面移动速度
=0~255, 实际为 (248ms+ 设定值 ×8ms) 时间内移动一次
- ③ SHIFT_NUM: 画面移动格数
=1~255, 每次移动多少列
- ④ SHIFT_LOOP: 画面移动次数
=1~255, 画面一共移动多少次
- ⑤ Rotate: 画面是否循环
=0, 不循环
=1, 循环

调用方式:

```
_PLAY_PICTURE_SHIFT_SETTING(INT8U DIRECTION,INT8U SPEED,INT8U SHIFT_NUM,INT8U SHIFT_LOOP,INT8U ROTATE)
```

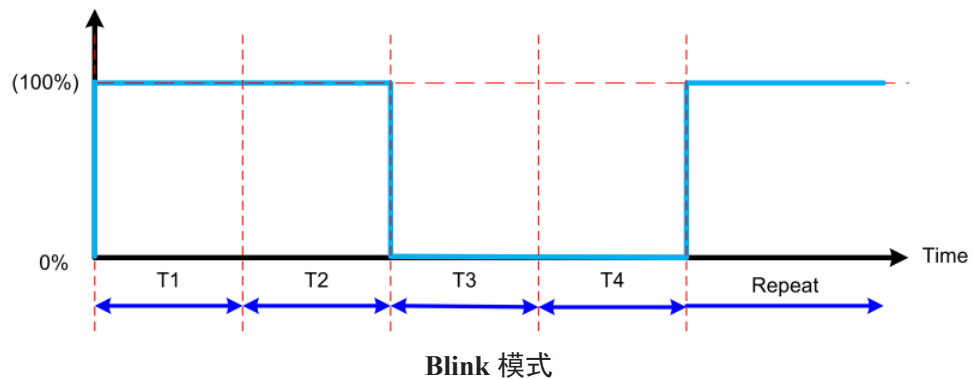
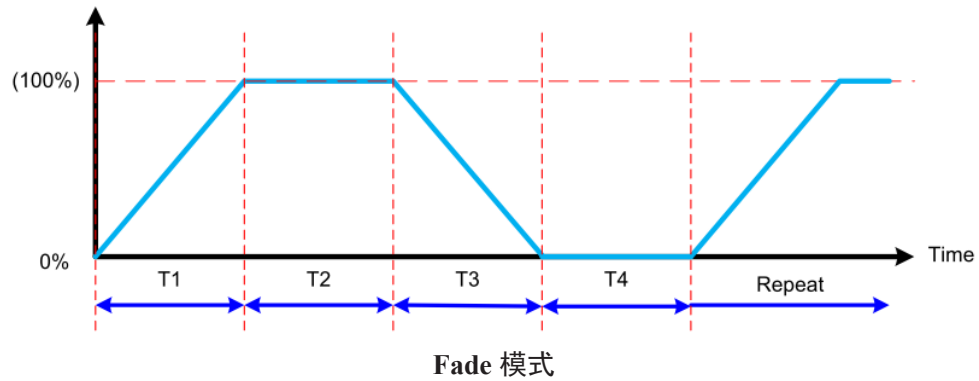
说明: 调用此函数时需对函数的入口参数进行赋值, 如示例中的 DIRECTION、SPEED、SHIFT_NUM、SHIFT_LOOP、ROTATE, 需要先给这五个变量赋值。例如 DIRECTION=0, SPEED=1, SHIFT_NUM=1, LOOP=2, ROTATE=1 时调用函数, 实现画面每隔 256ms 向右循环移动一格, 共移动 2 次的显示效果。

5.1.3 PLAY+FADE

```
void _PLAY_PICTURE_FIREFLY_SETTING(INT8U LGS,INT8U FBS,INT8U T1,INT8U T2,INT8U T3,INT8U T4)
```

入口参数:

- ① LGS: 线性 / 伽马显示效果选择
=0, 线性效果
=1, 伽马效果
- ② FBS: 渐变 / 闪烁显示效果选择
=0, 渐变效果
=1, 闪烁效果
- ③ T1: T1 持续时间
=0~7 (0: off; 1: 256; 2: 512; 3: 1024; 4: 1536; 5: 2048; 6: 2560; 7: 3072) 帧时间
- ④ T2: T2 持续时间
=0~3 (0: T1×0.25; 1: T1×0.5; 2: T1; 3: T1×2)
- ⑤ T3: T3 持续时间
=0~1 (0: T1; 1: T1×2)
- ⑥ T4: T4 持续时间
=0~3 (0: T1×0.25; 1: T1×0.5; 2: T1; 3: T1×2)



调用方式:

`_PLAY_PICTURE_FIREFLY_SETTING(INT8U LGS,INT8U FBS,INT8U T1,INT8U T2,INT8U T3,INT8U T4)`

说明: 调用此函数时需对函数的入口参数进行赋值, 如示例中的 LGS、FBS、T1、T2、T3、T4, 需要先给这六个变量赋值, 再进行库函数调用。例如 LGS=0, FBS=1, T1=1, T2=1, T3=1, T4=1 时调用函数, 实现 T1~T4 的持续时间均为 1 倍的 T1 时间, T1 时间具体参照规格书。

5.2 HT16D35A/B 调用方法

5.2.1 入口参数为变量

以下三个函数的入口参数必须为变量。

5.2.1.1 PLAY+NONE

`_PLAY_PICTURE_DISPLAY MACRO Play_Number,Frame`

入口参数:

- ① Play_Number: 在添加的 LDW 档中选择第几个 LDW 档;
- ② Frame: 在选择的 LDW 档中选择第几帧;

调用方式: `_PLAY_PICTURE_DISPLAY GU8V_LDW_INDEX,GU8V_FRAME_INDEX`

说明: 自行定义两个变量用于存放数据, 如示例中的 GU8V_LDW_INDEX 和 GU8V_FRAME_INDEX, 需要先给这两个变量赋值, 再进行库函数调用。例如 GU8V_LDW_INDEX=2, GU8V_FRAME_INDEX=1 时调用函数, 实现第 2 个 LDW 档第 1 帧画面的显示。

5.2.1.2 PLAY+SHIFT

`_PLAY_PICTURE_SHIFT` **MACRO** `COLOR_Direction,COLOR_Speed,COLOR_ShiftNum,COLOR_ShiftLoop,COLOR_Rotate`

入口参数:

- ① `COLOR_Direction`: 画面移动方向
=0~3 (0: 上移; 1: 下移; 2: 右移; 3: 左移)
- ② `COLOR_Speed`: 画面移动速度
=0~255, 实际为 (96ms + 设定值 × 8ms) 时间内移动一次
- ③ `COLOR_ShiftNum`: 画面移动格数
=1~255, 每次移动多少行 (上移或下移时) 或多少列 (左移或右移时)
- ④ `COLOR_ShiftLoop`: 画面移动次数
=1~255, 画面一共移动多少次
- ⑤ `COLOR_Rotate`: 画面是否循环
=0, 不循环
=1, 循环

调用方式:

`_PLAY_PICTURE_SHIFT` `GU8V_SHIFT_DIRECTION, GU8V_SHIFT_SPEED, GU8V_SHIFT_NUM, GU8V_SHIFT_LOOP, GU8V_SHIFT_ROTATE`

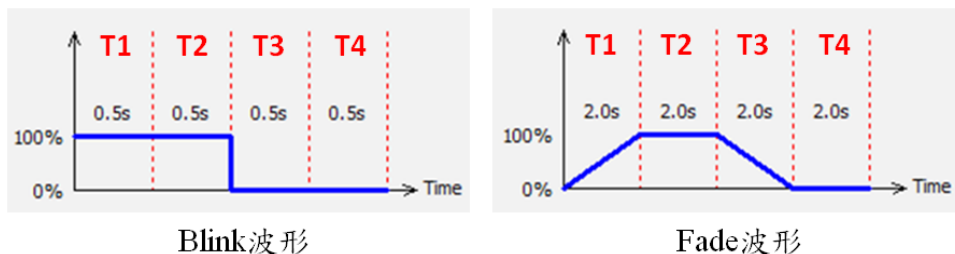
说明: 自行定义五个变量用于存放数据, 如示例中的 `GU8V_SHIFT_DIRECTION`、`GU8V_SHIFT_SPEED`、`GU8V_SHIFT_NUM`、`GU8V_SHIFT_LOOP`、`GU8V_SHIFT_ROTATE`, 需要先给这五个变量赋值, 再进行库函数调用。例如 `GU8V_SHIFT_DIRECTION=2`, `GU8V_SHIFT_SPEED=1`, `GU8V_SHIFT_NUM=1`, `GU8V_SHIFT_LOOP=2`, `GU8V_SHIFT_ROTATE=1` 时调用函数, 实现画面每隔 104ms 向右循环移动一格, 共移动 2 次的显示效果。

5.2.1.3 PLAY+FADE

`_PLAY_PICTURE_FIREFLY` **MACRO** `COLOR_BFS,COLOR_T1,COLOR_T2,COLOR_T3,COLOR_T4,COLOR_Times`

入口参数:

- ① `COLOR_BFS`: Blink/Fade 显示效果选择
=0, Blink
=1, Fade
- ② `COLOR_T1`: T1 持续时间
=0~3 (0: 0.5s; 1: 1.0s; 2: 1.5s; 3: 2.0s)
- ③ `COLOR_T2`: T2 持续时间
=0~3 (0: 0.5s; 1: 1.0s; 2: 1.5s; 3: 2.0s)
- ④ `COLOR_T3`: T3 持续时间
=0~3 (0: 0.5s; 1: 1.0s; 2: 1.5s; 3: 2.0s)
- ⑤ `COLOR_T4`: T4 持续时间
=0~3 (0: 0.5s; 1: 1.0s; 2: 1.5s; 3: 2.0s)
- ⑥ `COLOR_Times`:
=1~255, 画面一共闪烁多少次



调用方式:

```
_PLAY_PICTURE_FIREFLY GU8V_FIREFLY_FADE, GU8V_FIREFLY_T1, GU8V_FIREFLY_T2, GU8V_FIREFLY_T3, GU8V_FIREFLY_T4, GU8V_FIREFLY_CYCLETIME
```

说明: 自行定义六个变量用于存放数据, 如示例中的 GU8V_FIREFLY_FADE、GU8V_FIREFLY_T1、GU8V_FIREFLY_T2、GU8V_FIREFLY_T3、GU8V_FIREFLY_T4、GU8V_FIREFLY_CYCLETIME, 需要先给这六个变量赋值, 再进行库函数调用。例如 GU8V_FIREFLY_FADE=0, GU8V_FIREFLY_T1=1, GU8V_FIREFLY_T2=1, GU8V_FIREFLY_T3=1, GU8V_FIREFLY_T4=1, GU8V_FIREFLY_CYCLETIME=1 时调用函数, 实现 T1~T4 的持续时间均为 1.0s, 一共闪烁 1 次的 Blink 显示效果。

5.2.2 入口参数为立即数

以下三个函数与入口参数为变量的函数相同, 区别仅在于这三个函数的入口参数必须使用立即数, 故这里不再赘述这三个函数每个入口函数的定义。

5.2.2.1 PLAY+NONE

```
_PLAY_PICTURE_DISPLAY_IM MACRO Play_Number_IM, Frame_IM
```

调用方式: `_PLAY_PICTURE_DISPLAY_IM 2, 1`

说明: 实现第 2 个 LDW 档第 1 帧画面的显示。

5.2.2.2 PLAY+SHIFT

```
_PLAY_PICTURE_SHIFT_IM MACRO COLOR_Direction_IM, COLOR_Speed_IM, COLOR_ShiftNum_IM, COLOR_ShiftLoop_IM, COLOR_Rotate_IM
```

调用方式: `_PLAY_PICTURE_SHIFT_IM 2, 1, 1, 2, 1`

说明: 实现画面每隔 104ms 向右循环移动一格, 共移动 2 次的显示效果。

5.2.2.3 PLAY+FADE

```
_PLAY_PICTURE_FIREFLY_IM MACRO COLOR_BFS_IM, COLOR_T1_IM, COLOR_T2_IM, COLOR_T3_IM, COLOR_T4_IM, COLOR_Times_IM
```

调用方式: `_PLAY_PICTURE_FIREFLY_IM 0, 1, 1, 1, 1, 1`

说明: 实现 T1~T4 的持续时间均为 1.0s, 一共闪烁 1 次的 Blink 显示效果。

Copyright© 2019 by HOLTEK SEMICONDUCTOR INC.

使用指南中所出现的信息在出版当时相信是正确的，然而 Holtek 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，Holtek 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。Holtek 产品不授权用于救生、维生从机或系统中做为关键从机。Holtek 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 <http://www.holtek.com/zh/>。