



IR 遥控器开发平台使用手册

版本: V1.00 日期: 2024-01-19

www.holtek.com

目录

一、简介	3
二、开发平台软件	4
2.1. 一般型遥控器开发	6
2.2 LCD 型遥控器开发	16
三、开发平台硬件	33
3.1 遥控器开发板	33
3.2 解码板	35
四、其他功能说明	36
4.1 F/W 功耗控制	36
4.2 解码识别	36
五、附录 & FAQ	37
5.1 LCD 型遥控器功能的数据，按键，显示及函数关联表	37
5.2 特殊调制发码说明及应用	38
5.3 MARK 与 SPACE 设计误差说明	38

一、简介

Holtek IR 遥控器开发平台是一款用于快速设计出红外遥控器的设计软件。软件提供常规的标准协议，如 NEC、NEC-16、Philips RC-5、Philips RC-6、Sharp 等多种常见遥控器协议的遥控器软件开发。平台还提供自定义红外遥控发码参数的设计方式，可以设计出有别于其他标准协议的特殊参数。另外，配合 Holtek 提供的 IR 解码板，可以对已有遥控器做 IR 波形的解析，可用于开发与已有遥控器同协议的遥控器，或用于对正在开发中的遥控器做发码验证。

常用红外线遥控器主要应用功能分为以下几类：

1. 一般型遥控器

按键：识别用户操作。

调制发码：使用不同的发码协议调制信号，避免不同接收设备相互干扰。

指示灯：发码指示。

静态功耗控制：延长电池寿命。

2. LCD 型遥控器

按键：识别用户操作。

调制发码：使用不同的发码协议调制信号，避免不同接收设备相互干扰。

LCD 显示：显示遥控器目前向设备发送的各种工作状态信息。

背光灯：用于 LCD 的背光驱动。

静态功率控制：延长电池寿命。

二、开发平台软件



启动界面

软件启动之后如上图所示，左上方菜单栏分别为工程、选项、语言和帮助。

工程选项提供新建工程和打开工程选项。

选项选单提供学码模式挂件和复位动态图演示功能。

语言选单提供简体中文、繁体中文和英文三种语言切换。

帮助选单提供软件使用手册、开发板原理图、红外遥控器知识库和平台版本信息。

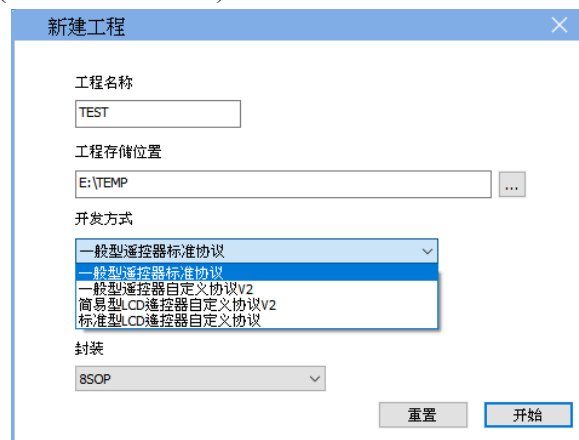


工程管理

在工程管理界面，可以建立新工程或者打开旧工程，在软件界面上还会提供几范例方案供用户参考。

在平台软件上的遥控器开发过程主要为以下几个配置步骤：

- a) 协议定义
- b) LCD 显示内容 (仅限 LCD 型遥控器)
- c) 按键及驱动配置
- d) 产出 Project (可再做二次开发) ，转到 IDE3000 编译产出烧录档



工程建立

2.1. 一般型遥控器开发

一般型遥控器开发支持标准协议开发和自定义协议开发两种开发模式。

2.1.1 标准协议开发



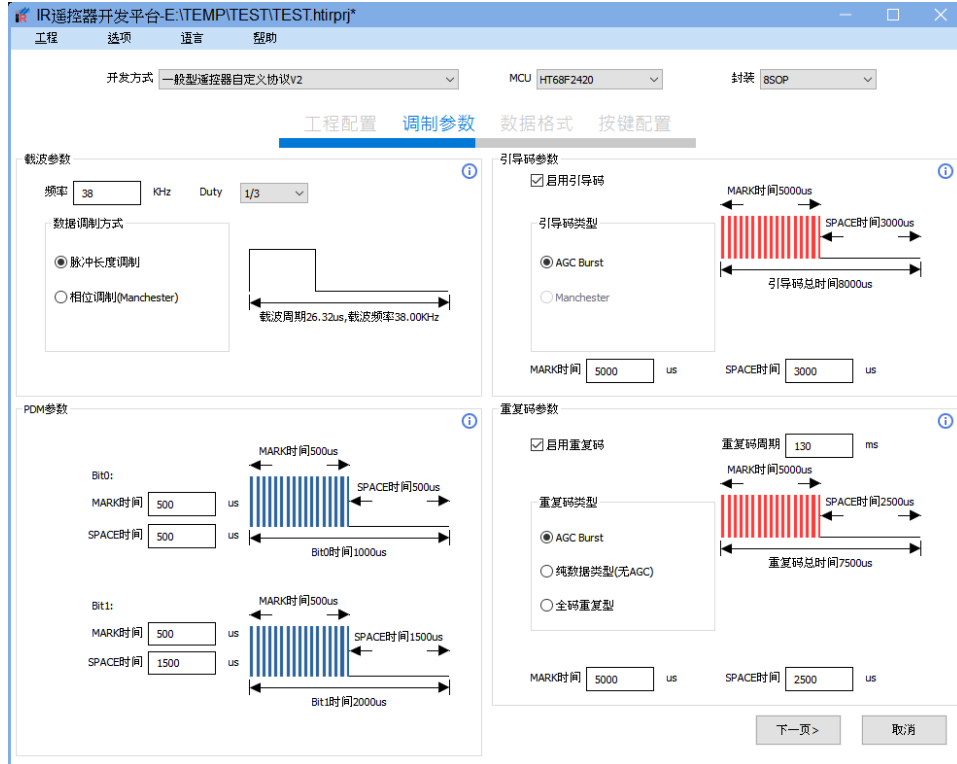
标准协议配置界面

如上图所示，在一般型遥控器的标准协议开发共分两步，在协议选择步骤可以选择 NEC、Philips RC-5、Philips RC-6、Sharp、JVC 等市场上流行的标准协议开发。使用标准协议不需对协议控制的参数做任何配置，软件界面提供波形图预览。在选择使用的协议之后，只需要配置合适的 Duty 和地址码即可完成对协议部分的配置。

2.1.2 自定义协议开发

一般型遥控器自定义协议开发分三个步骤。

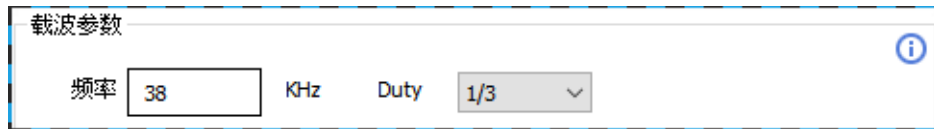
第一个步是设定调制参数。在调制参数页可以自定义载波参数，数字 1 和 0 的调制参数，引导码参数和重复码参数。



自定义协议配置界面

具体的配置操作方式如下：

载波

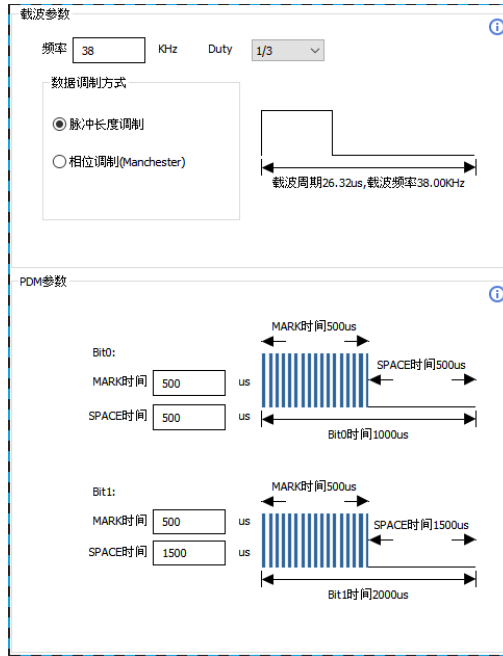


载波参数

如上图，在自定义协议模式中自定义载波参数，载波频率范围为 30kHz~58kHz，Duty 可选 1/2、1/3 和 1/4。

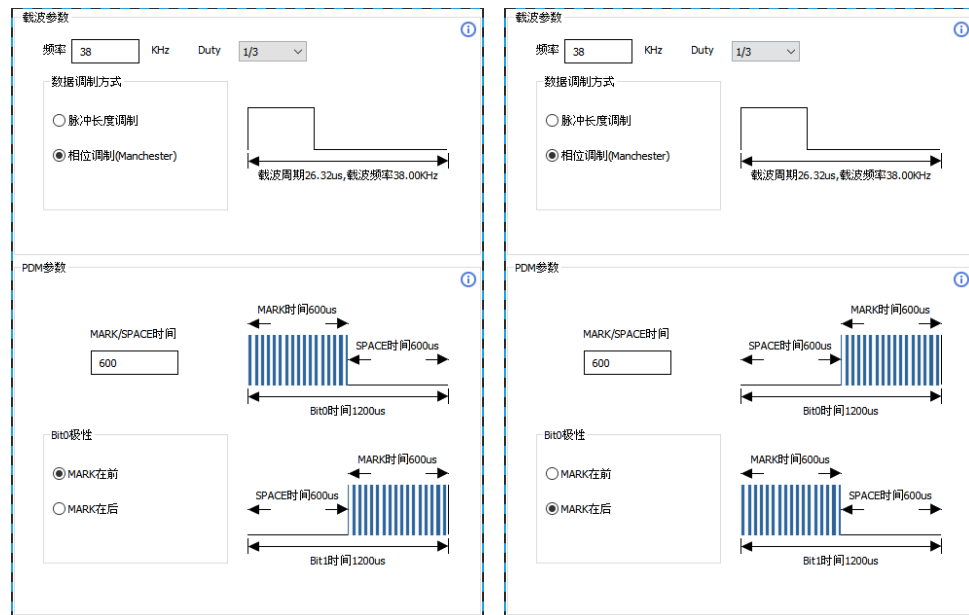
数据调制方式

数据的调制方式可以选择脉冲长度调制 (PDM) 或相位调制 (Manchester)。



脉冲长度调制 (PDM) 参数

Bit0 和 Bit1 的脉冲长度调制参数的配置图如上图。在软件上可以配置遥控器发送的调制信号 Bit0 和 Bit1 的 MARK (连续载波脉冲) 时间和 SPACE (无脉冲) 时间, 同时软件在参数右方以图形的方式标示发送 Bit 波形的 MARK、SPACE 和总时间的参数。



相位调制 (Manchester) 参数

上图是相位调制参数的配置图。在软件上可以配置出 Bit0 和 Bit1 的 MARK/SPACE 时间, 相位调制只需要定义一个时间参数, 再通过 Bit0 极性定义的方式

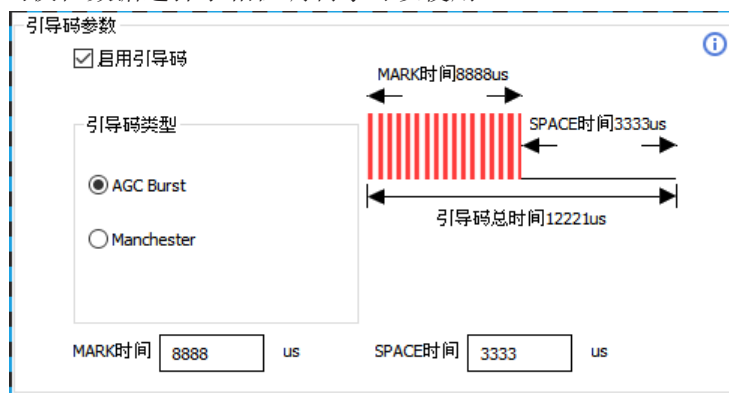
选择为 MARK 在前或是 MARK 在后来设定 Bit 极性。参数右方以图形的方式分别标示 MARK、SPACE 和总时间几项参数。

Bit 的 MARK 和 SPACE 的参数范围可配置在 300 μ s~2000 μ s 之间。

引导码 / 头码参数

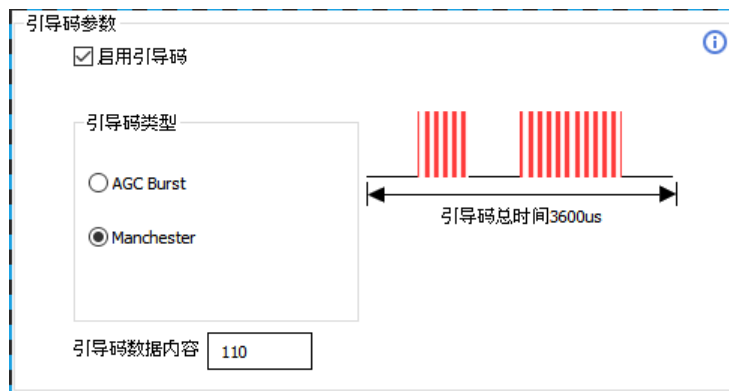
引导码也称为头码，主要用于遥控器向接收端发出控制信号前先发送一个初始化和校准的信号。

引导码支持使用 AGC Burst 和 Manchester 两种方式，Manchester (相位调制) 方式的引导码仅在数据选择了相位调制才可以使用。



AGC Burst 引导码参数

如上图，AGC Burst 类的引导码由一段比一般数据发码时间更长的 MARK 和 SPACE 组成。AGC Burst 引导码用于接收器初始化增益，但 MARK 的时长与功耗成正比，建议合理地配置此参数以控制遥控器功耗，可设定范围在 4ms~20ms。软件以图形的方式分别标示 MARK、SPACE 和总时间几项参数。



Manchester 引导码参数

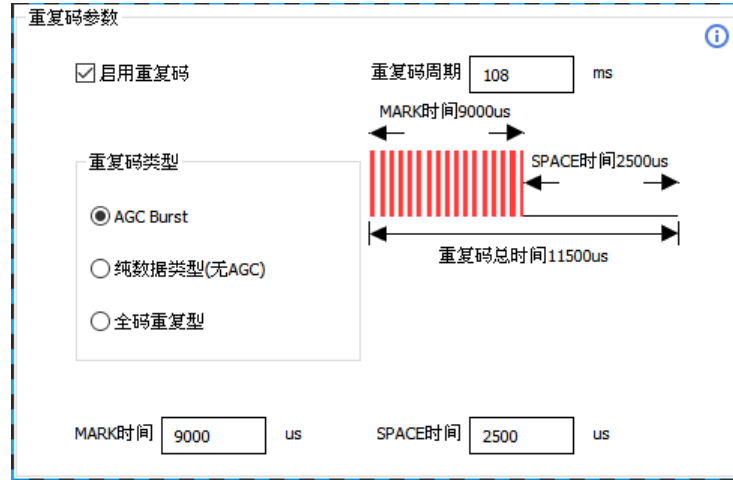
如上图，当数据使用 Manchester 调制方式时，则可使用类似 Philips RC-5 协议的数字形式引导码，引导码在软件中最多可配置 3 位，软件以图形的方式分别标示 MARK、SPACE 的图形位置，还有总时间参数。

重复码参数

重复码用于遥控器在按键被按下不放的状态发出重复信号。

如下图，当重复码启用之后，可对重复码参数进行配置。

重复码分为 AGC Burst、纯数据重复和全码重复三种类型。



重复码参数

重复码使用 AGC Burst 模式时需要对其参数进行配置，此参数可以与头码不一致，建议设定在 4ms~20ms 之间。

重复码的周期可以设定在 40ms~200ms 之间，重复周期的设定要保证重复的第二帧信号在第一帧信号结束之后至少 10ms 再发送。

数据格式

如下图，在第二页标签的数据格式中，可以设定固定码、命令码、发送次序、是否需要插入 Start 和 Toggle 位。



数据格式参数

固定码：可配置 1~4 组，默认强制按顺序启用，每组可任意配置为 1~8 位，可配置为正码或反码。

命令码：位数可配置范围为 1~8。

发送次序：可定义数据先发送高位或先发低位。

Start 位：用于在头码和数据码之间插入一个固定位。

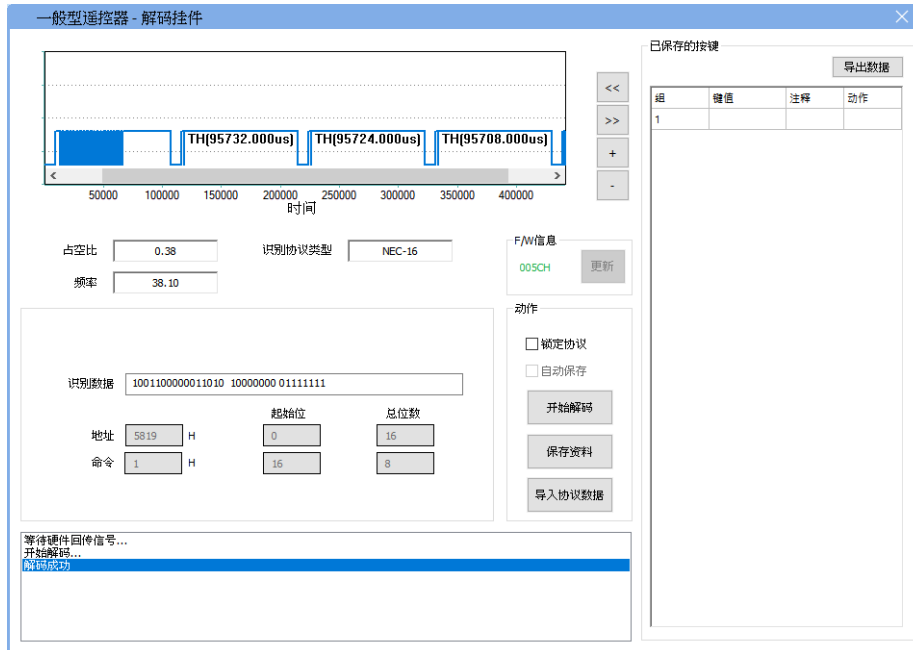
Toggle 位：用于插入一个变化位，区分全码重复的遥控器所发出的两组码是保持按住的重复发码还是按键连续两次被按下发出的两组码。

发送流程：以时间轴的图形展示发送流程，引导码之后可以选择发送固定码、固定反码、命令码、命令反码，以拖动的方式选择使用，前后顺序可任意调整。

2.1.3 学码 / 解码挂件

设计一般型遥控器，可以使用一般型遥控器解码挂件协助开发，解码功能需要 ESK-IRRC-R00 解码板连接电脑搭配使用。

如下图，软件会按照一般的红外接收头的输出方式绘出发码波形，有载波的 MARK 绘制为低电平，无载波的 SPACE 绘制为高电平。波形图下方为解析出来的发码参数，如载波的占空比、协议类型和频率。解码取得的 1/0 数字软件自动填入到识别数据中。识别数据的下方是解析地址和命令。仅当识别出标准协议时，软件会根据协议类型自动定位地址和命令所在的起始位和总位数。如果发码不符合标准协议，则软件会判定为自定义协议，需要手动输入地址和命令所在的起始位和总位数划分地址和命令。

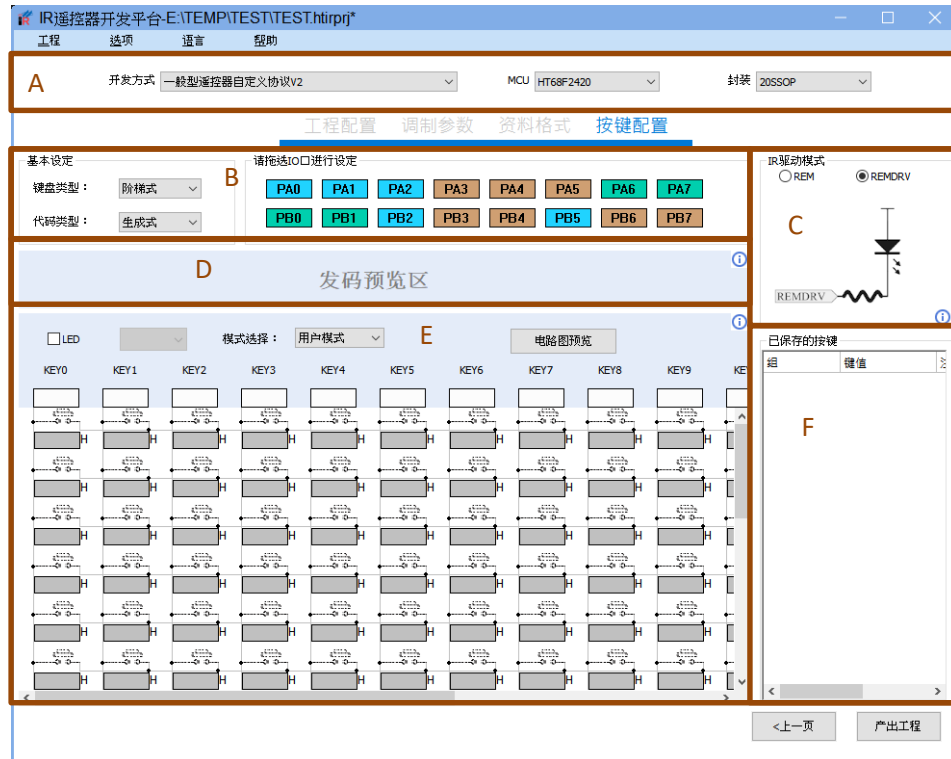


一般型遥控器解码挂件

识别出协议之后可以锁定协议并对遥控器的多个按键解码和保存，开发所需的按键都保存之后，按下导入协议数据即可将学习到的数据导入使正在开发的遥控器方案中。

2.1.4 按键与驱动

一般型遥控器标准协议或自定义协议开发到最后一步，是按键配置。如下图，A 区是方案类型和 MCU 类型配置区，B 区是基本设定和 I/O 待选区，C 区是 IR LED 驱动方式选择区，D 区可以预览发码，E 区是按键 I/O 配置及命令值写入区，F 区是学码开发已保存的按键区。

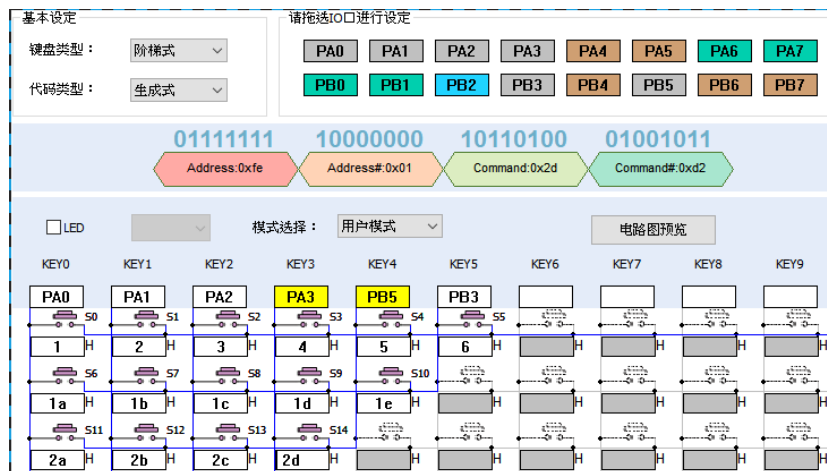


配置按键页

2.1.4.1 按键与 I/O

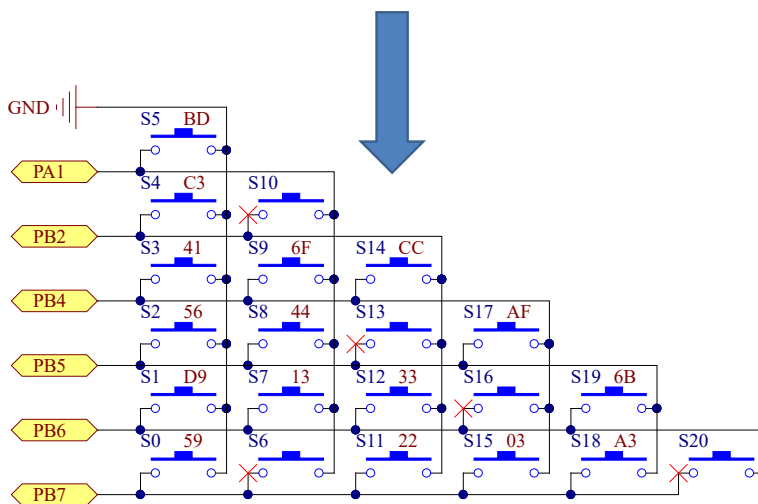
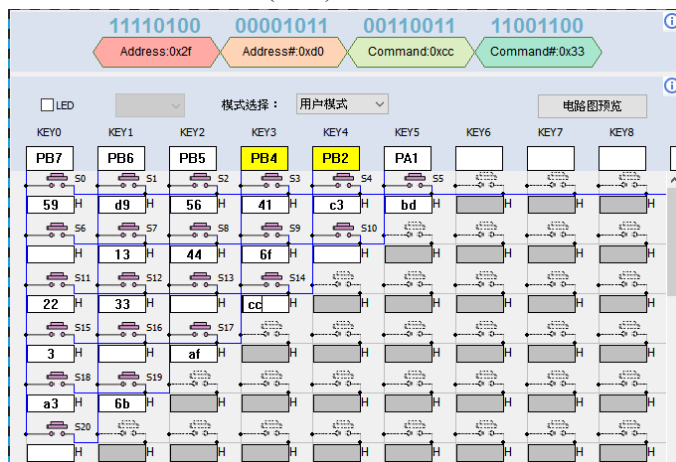
配置页面左上方 A 区可选择 MCU 的封装类型，B 区的基本设定可以选择不同的键盘类型和工程产出代码的方式，待选 I/O 区蓝色引脚图标为所有封装都共有的引脚，黄色引脚图标为 16NSOP 和 20SSOP/20NSOP 可使用的引脚，绿色引脚图标为 20SSOP/20NSOP 可使用的引脚。灰色代表已被使用或者当前选择的封装不包含的引脚。用拖选的方式将 I/O 图标拖入下方 E 区的按键引脚位置。

2.1.4.2 阶梯键盘



阶梯键盘配置按键图

如上图，当键盘类型选择了阶梯式，阶梯键盘表随 I/O 配置到按键自动生成。键盘最大键数计算公式为 $K = N \times (N + 1) / 2$ ，N 为已配置 I/O 的个数。



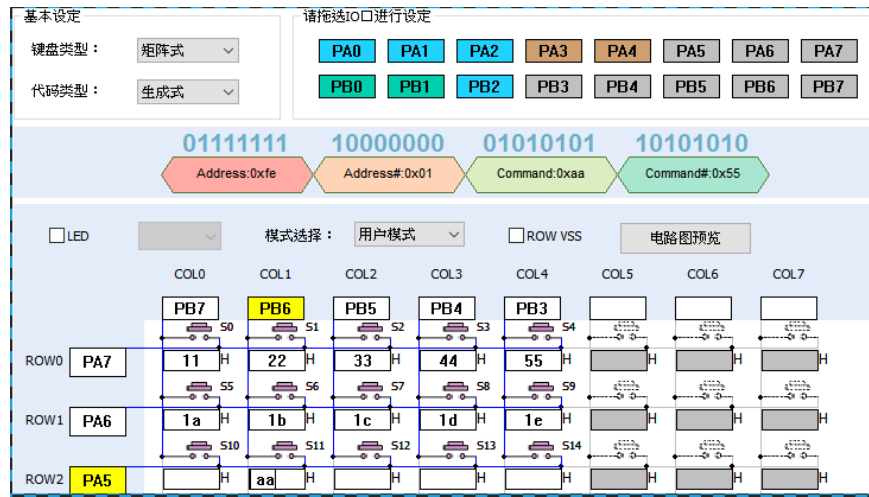
阶梯键盘按键配置与原理图关系

I/O 配置到阶梯按键区之后，可在对应的按键码表中输入命令码，鼠标点击任意一个可配置的按键码表格，与此命令码表格对应的两个 I/O 则会有黄色高亮提示，如上图上半部分按键 S14 的连线连接到 PB4 和 PB2 两个 I/O。在对应的按键码表中输入命令码之后，发码预览区会提示二进制发码形态，预览区二进制码是按时间关系从左到右列出，低位先发的协议所看到的二进制码是倒序的形式。

点击第一行的命令码仅有一个 I/O 亮起黄色，因为这组表格对应的按键是由 I/O 与 VSS 组成。

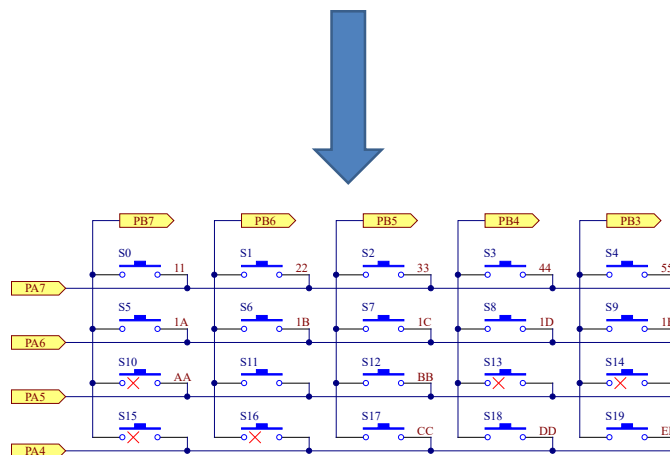
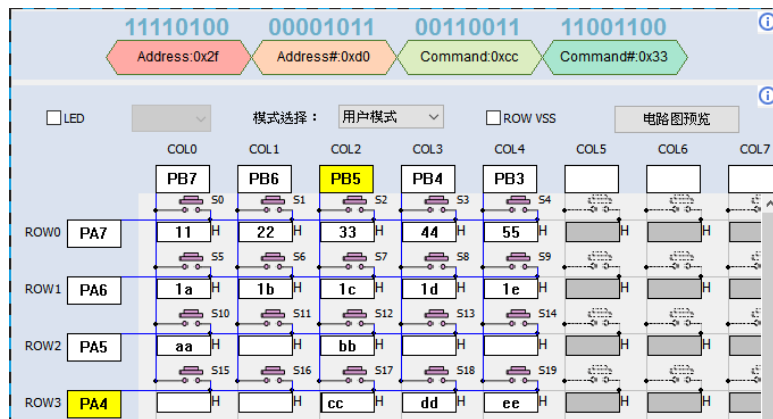
上图是阶梯按键表对应为下方的阶梯键盘原理图。未配置键值的按键在原理图上以打叉表示。图中 S6、S10、S13、S16、S20 即使未配置键值或者在实体电路未焊按键，但在软件生成配置时也会列入扫描的码表中。未配置按键在软件生成的键值表默认填充 00H。

2.1.4.3 矩阵键盘



矩阵键盘配置按键图

如上图，当键盘类型选择了矩阵式，I/O 配置到按键配置表 ROW 和 COL 之后，矩阵键盘表随即自动生成。可生成的最大键数计算公式为 $K=ROW \times COL$ ，8SOP 封装 5 个 I/O 最多可配置 6 个按键，16NSOP 封装 12 个 I/O 最多可配置 36 个按键，20NSOP/SSOP 封装 16 个 I/O 最多可配置 64 个按键。ROW VSS 可以额外增加一组 COL 数量的按键。



矩阵键盘按键配置与原理图关系

I/O 在软件中配置到按键区之后，可在对应的按键码表中输入命令码，鼠标点击任意一个可配置的按键码表格，与此命令码表格对应的两个 I/O 则会有黄色高亮提示，如上图上半部分按键 S17 的连线连接到 PB5 和 PA4 两个 I/O。在对应的按键码表中输入命令码之后，发码预览区同样提示二进制码。

在启用 ROW VSS 时，点击 ROW VSS 行的命令码仅有一个 I/O 亮起黄色，因为这组表格对应的按键是由 I/O 与 VSS 组成。

上图是矩阵按键表对应为下方的矩阵键盘原理图。未配置键值的按键在原理图上以打叉表示。即使未配置键值或者在实体电路未焊按键，但在软件生成配置时也会列入扫描的码表中。未配置按键在软件生成的键值表默认填充 00H。

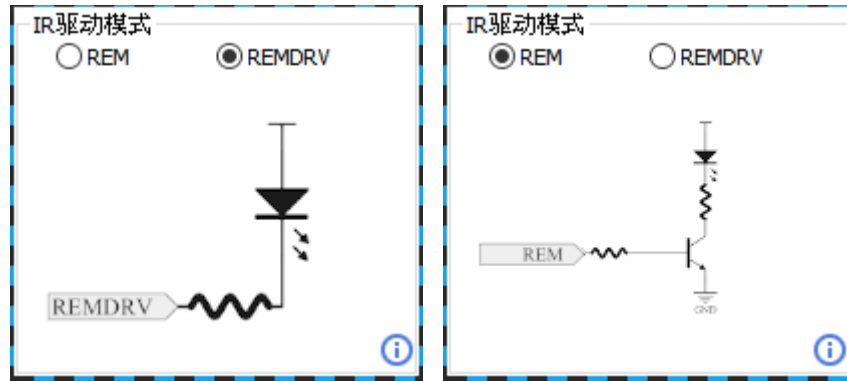
2.1.4.4 开发板模式

当 MCU 选择了 20SSOP 的 HT68F2420 时，键盘可以快捷选择配置为开发板模式，可以一键配置出可兼容开发板的按键，阶梯键盘可配置出 136 个按键，而矩阵键盘可配置出 72 个按键，由于 HT68F2420 开发板 ESK-IRRC-T00 是以阶梯键盘的方式画的 PCB 打上丝印，所以软件标示出的矩阵键盘按键位置并非按照自然顺序，而是按照 ESK-IRRC-T00 开发板上 I/O 配置对应的丝印的键位。

2.1.4.5 LED 驱动

LED 驱动部分包含 MCU 专用的驱动 IR LED 发射管和 I/O 驱动 LED 指示灯两部分。

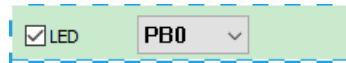
IR LED 驱动模式如下图。



IR LED 驱动配置

遥控器对发射红外线信号的 IR LED 的驱动是通过 MCU 的 REM/REMDRV 引脚输出，使用 REMDRV 驱动方式可以提供较大的驱动电流可以直驱 IR LED。使用 REM 的驱动方式需要外加三极管或场效应管以提高输出能力。

LED 指示灯驱动 I/O 口在 E 区配置，驱动 LED 指示灯可以使用任意独立的 I/O 引脚。阶梯键盘中还可以与 KEY0 复用，为了避免功耗增加和 LED 出现微亮，LED 指示灯与按键复用时只能复用在 KEY0。



指示灯配置

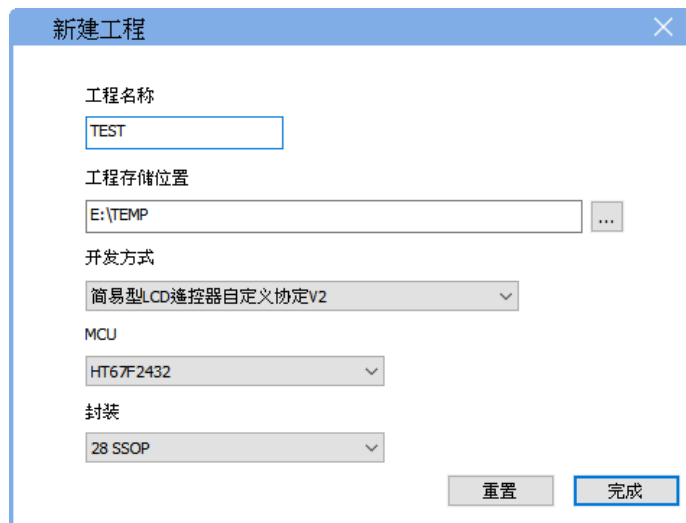
按键配置结束后，点击电路图预览可以导出遥控器的原理图。

完成配置之后可在 B 区在基本设定中选择生成代码的类型，可选择条件编译或者生成式的方式产出 IDE 工程，条件编译更便于二次开发，生成式程序可读性更好。

完成后点击产出工程即可生成开发方案的对应工程文件夹。

生成文件夹的自动命名规则为：“工程名称_年-月-日_时-分-秒”。

2.2 LCD 型遥控器开发



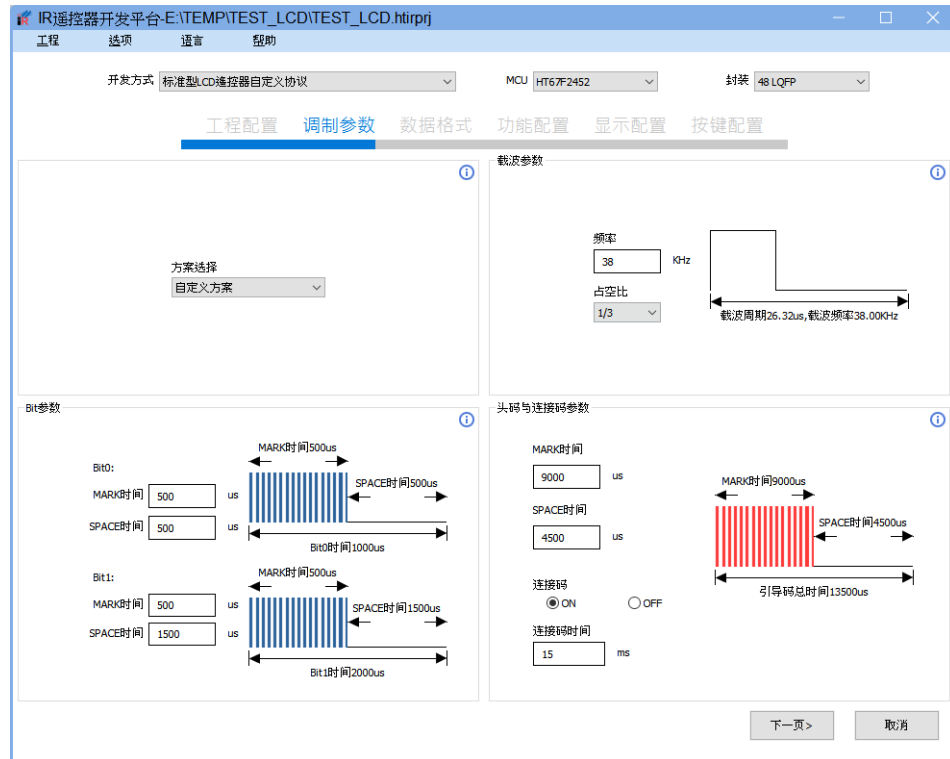
工程设置配置 LCD 型遥控器

开发平台根据 Holtek 不同的 LCD 遥控器 MCU 的资源，开发平台提供了简易型 LCD 型遥控器和标准型 LCD 型遥控器两套方案。两套方案都有基本的按键检测，LCD 显示和红外发射功能，标准型 LCD 型遥控器相比简易型 LCD 型遥控器有更多的可用资源，增加了几项可选配置功能，两套方案具体差别如下：

	简易型 LCD 型遥控器	标准型 LCD 型遥控器
发码模式	单组单 / 双帧	单组单 / 双帧 双组单 / 双帧
定时模式	2 种可选	3 种可选
双键同按	×	√
切换单位	×	√
锁键盘	×	√
时钟显示	×	√
ROM 适配	2K	4K 及以上

2.2.1 调制方式

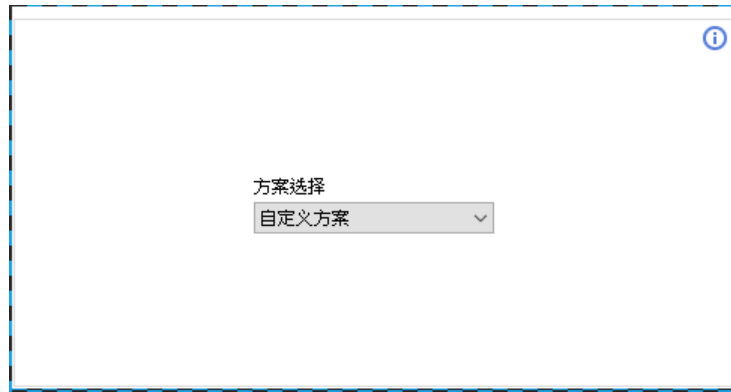
LCD 型遥控器在本平台软件上开发提供自定义协议的方式。开发者可以根据产品需求定制出参数和发码规则。



调制方式配置页面

配置发码首先是配置调制方式，LCD 型遥控器的编码仅限于 PDM 的调制方式，不支持相位方式调制。调制方式页包含 4 部分内容，分别是方案选择、载波参数、Bit 参数、头码 (引导码) 与连接码参数。

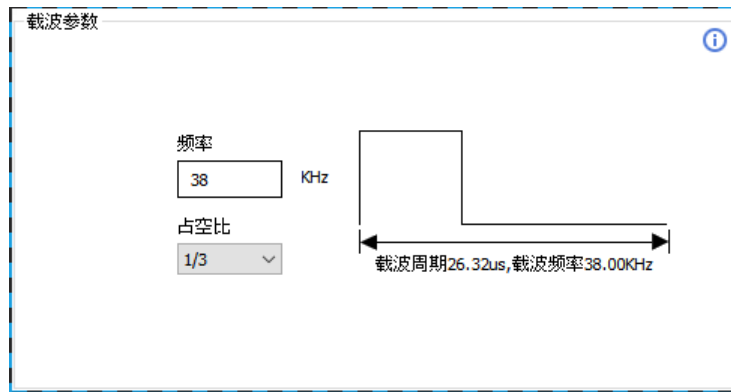
方案选择



方案选择

在方案选择中软件内置了两种设计方案，分别配置了不同的调制方式，调制参数和发码数据格式。

载波参数

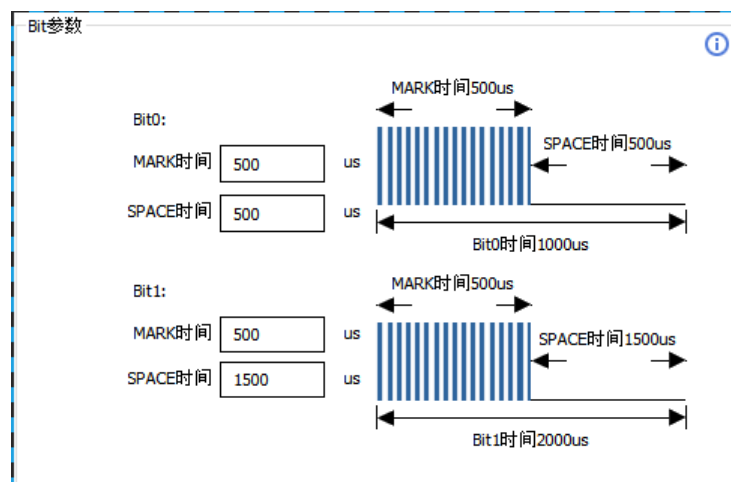


载波参数

载波参数可以设置载波的频率和占空比，载波频率范围为 30kHz~58kHz，Duty 可选 1/2、1/3 和 1/4。

Bit 参数

在 Bit 参数区可以配置 Bit0 和 Bit1 的 MARK (连续载波脉冲时间) 和 SPACE (无脉冲时间)，软件以图形的方式分别标示 1 个 Bit 的 MARK、SPACE 的对应参数。Bit 的 MARK 和 SPACE 的时间参数范围可配置在 300μs~2000μs 之间。

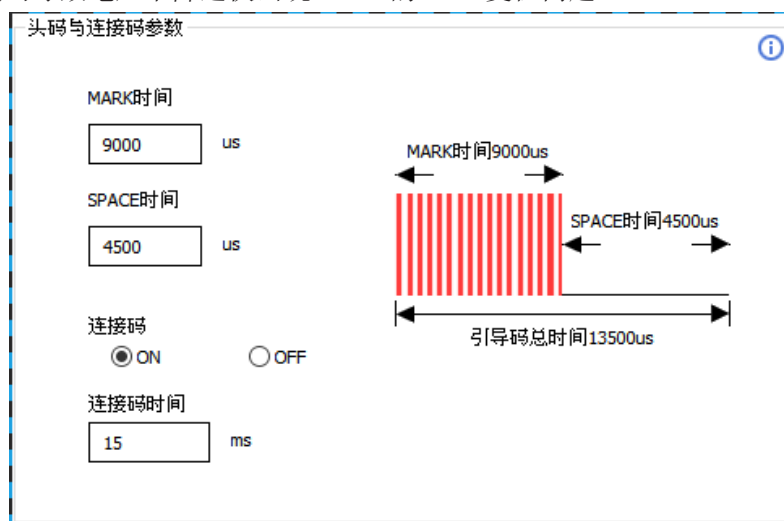


Bit 参数

头码与连接码参数

头码 (引导码) 主要用于遥控器对接收器发出一个初始化和校准的信号, LCD 型遥控器的引导码仅支持 AGC Burst 的方式。AGC Burst 引导码由一段比一般数据发码时间更长的 MARK 和 SPACE 组成。AGC Burst 引导码中时间较长的 MARK 有助于接收器初始化增益, 软件限制头码时间在 3ms~20ms 之间。软件以图形的方式分别标示 MARK、SPACE 和总时间几项参数。

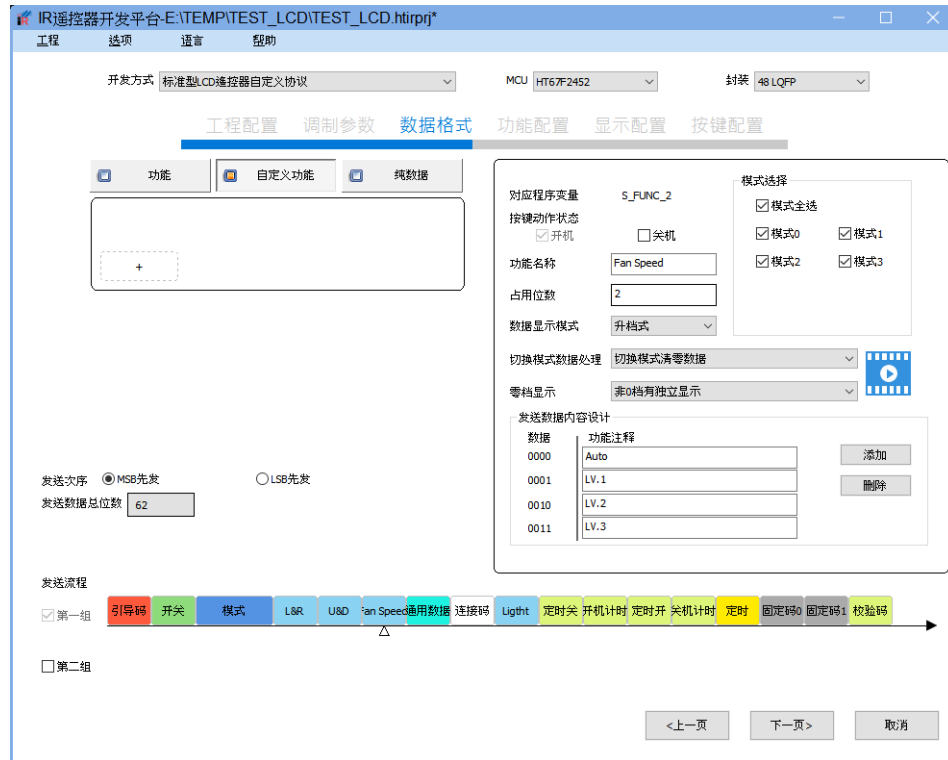
连接码是在 LCD 型遥控器的一长串 bit 发码中间插入一个较长的 SPACE。连接码既可用作解码软件辨识协议使用, 也可以用于避免电池在低电量时因连续不断的发码导致电压下降过快出现 MCU 的 LVR 复位问题。



头码与连接码参数

2.2.2 数据格式

配置发码的第二步是配置发送数据格式。常用 LCD 型遥控器所使用的数据格式，是将 LCD 的显示内容调制成发送数据。



数据格式页

数据格式页主要配置发送数据。

页面左方是发送数据可配置列表，右方是发送数据可配置内容。左下方发送次序可配置低位先发 (LSB) 或者高位先发 (MSB)。最下方是以时间轴的图形展示发送流程。简易型 LCD 型遥控器提供一组数据发送方式，标准型 LCD 型遥控器提供两组发送方式。双组模式应用可通过二次开发的方式使用按条件发送，典型应用为格力的空调遥控器仅在启用定时功能时发送两组，不启用定时功能则只发送一组，达到省电的目的。

左方的列表按照功能的差异划分为三类数据，分别是一般功能数据，自定义功能数据和纯数据。配置需要发送的数据，只需要选中对应的数据方块，拖到下方的发送流程中。

一般功能数据包含开关，模式，定时，其中开关和模式是必用功能，每个功能都有相应的 LCD 显示和对应的按键。

自定义数据功能是由用户自由配置的发送功能，可以由用户按需配置，每个功能都有相应的 LCD 显示和对应的按键。

纯数据包含通用数据，固定码，定时相关数据，校验码和自由数据。通用数据有对应的 LCD 数字显示，由“+”和“-”按键控制，其他纯数据没有对应的 LCD 显示或按键。

本页右半部分是用户可配置的参数。

对应程序变量：当前配置的数据对应产出工程程序中的变量；

按键动作状态：当前配置的按键可在 LCD 显示开机 / 关机的状态下响应按键动作；

功能名称：当前配置的默认名称，自定义功能和自由数据可以根据开发需求更改为相应的名称；

占用位数：当前配置的功能在发送流程中所占用的位数，默认按照档位数的二进制模式配置最低的占用位数（可在二次开发时修改）；

模式选择（自定义功能配置）：配置自定义功能在指定的某几个模式按键可用；

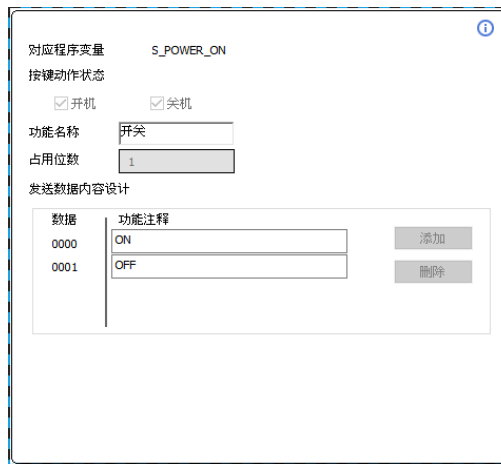
数据显示模式（自定义功能配置）：配置自定义功能所关联的 LCD 显示模式

切换模式数据处理（自定义功能配置）：当遥控器按下“模式”按键切换模式时，对当下自定义功能的数据处理方式。

零档显示（自定义功能配置）：此功能在 0 档和非 0 档时是否做显示的处理。

发送内容设计：配置自定义功能的档位数及档位功能注释。

开关



对程序变量 S_POWER_ON

按键动作状态

开机 关机

功能名称 开关

占用位数 1

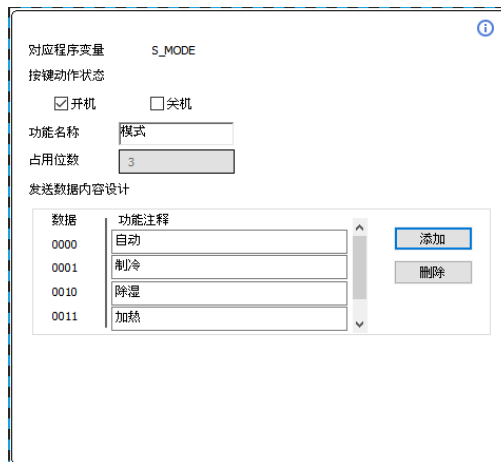
发送数据内容设计

数据	功能注释	
0000	ON	添加
0001	OFF	删除

开关按键配置

开关功能应该要配置为开机和关机状态都可以执行按键动作。产出的程序默认定义开机是 1，关机是 0。

模式



对程序变量 S_MODE

按键动作状态

开机 关机

功能名称 模式

占用位数 3

发送数据内容设计

数据	功能注释	
0000	自动	添加
0001	制冷	删除
0010	除湿	
0011	加热	

切换式模式按键配置

模式功能一般配置为开机状态执行按键动作。在发送数据内容设计中，可以通过添加和删除按钮配置工作模式，最多可以设置 8 种模式，在软件中可以对每种模式写入功能注释。模式在遥控器 LCD 显示的方式是切换式显示。

定时

定时按键配置

定时功能一般配置在开机和关机状态都执行按键动作。定时功能默认当遥控器处于开机状态时，定时按键启用定时关机；遥控器处于关机状态时，定时按键启用定时开机。在定时功能启用时，如果按下开关按键，则自动退出定时模式。24.5 模式最小操作数是半小时，24 模式最小操作数是 1 小时，88:88 模式最小操作数是 1 分钟。定时功能默认发送数据的位长用于存小时或半小时数，建议设定为 6~8 位长度。

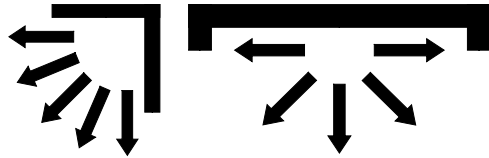
自定义功能

自定义功能配置

自定义功能可以根据用户需要的按键功能特性配置对应的功能。功能名称可以自由修改，后面配置 LCD 的和按键会相应的显示为已修改的自定义功能。

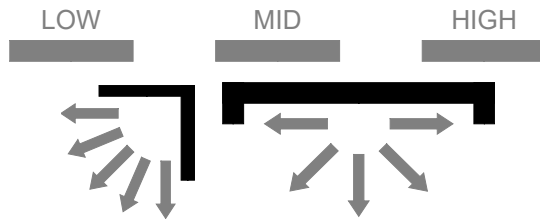
按遥控器 LCD 显示模式可将数据分为以下几种

- (1) 开关式：以开关的方式显示的数据，比如空调遥控器的上下扫风，左右扫风，风扇遥控器的摇摆等功能。数据定义启动是 1，关闭是 0，如下图中的黑色图标所示。



常用开关式显示示例

- (2) 切换式：切换模式显示的数据，比如空调遥控器扫风功能指定吹风风向，风扇遥控器风速的大、中、小；直吹风、自然风、睡眠风这种类型的显示模式，如下图中的灰色图标所示



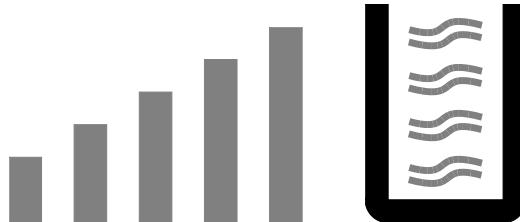
常用切换式显示示例

- (3) bit 式：用进位的模式显示需要简单相加的数据，可用于热水器遥控器的储水量控制，或杀菌时间类的显示控制，如下图中的灰色图标所示。



常用 bit 式显示示例

- (4) 升档式：每升一档增加一格的显示模式，一般用于空调遥控器的风速档位或热水器水位控制，如下图中的灰色图标所示。



常用升档式显示示例

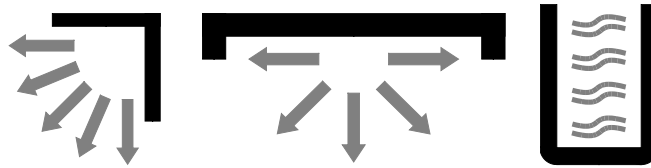
- (5) 数显式：以数字显示功能档位。

切换模式数据处理有三个方式分别是共享数据，独立数据和切换模式数据清零三种方式。

当按下模式按键执行切换模式操作时，三种数据处理方式在使用上的区别如下：

- (1) 共享数据会保持这个功能的档位大小或开关状态不变，比如空调遥控器在换气功能在切换模式后，仍保持换气功能状态。
- (2) 切换模式数据不清零：独立数据会为这个功能的每个模式分配独立缓存，切换模式自动加载上次在当前模式的对应档位，比如空调遥控器制冷模式的风速功能设置为 3 档，切换为其他模式变档后，切换到制冷模式仍可保持为 3 档。
- (3) 切换模式数据清零：会把这个功能的档位清零，比如空调遥控器在超强制冷功能，保证空调从制冷模式切换到非制冷模式时超强制冷功能立即关闭。

零档位在 LCD 上默认不做显示，有些设备需要把零档做特殊显示（比如“**AUTO**”），可在自定义功能配置上启用零档显示。有些设备需要非 0 状态显示正在工作（比如下图图标的黑色显示部分），则可在这个自定义功能配置上启用非零档显示。



常用非零档显示示例

发送内容设计可以自由增减当前设计功能的档位，开关式的数据最大 1 档，切换式最大 7 档，bit 式最大 15 档，升档式最大 7 档，数显式最大 9 档。

纯数据

- 通用数据

对应程序变量 S_Universal_Data

功能名称

占用位数

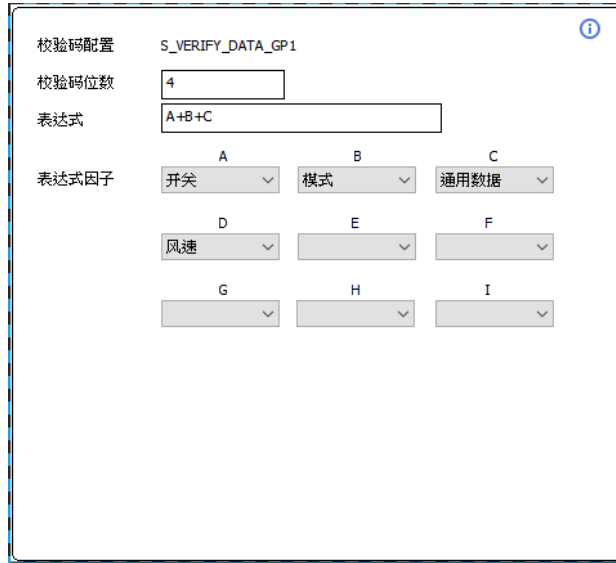
显示数据状态

开机 关机

通用数据配置

通用数据是 LCD 型遥控器用于全局数字显示的数据。常见的应用为空调遥控器的温度，无级风扇遥控器的转速档位，热水器 / 浴霸遥控器的温度，照明灯具的光强档位等。通用数据的变化由遥控器的“+”和“-”两个按键控制。

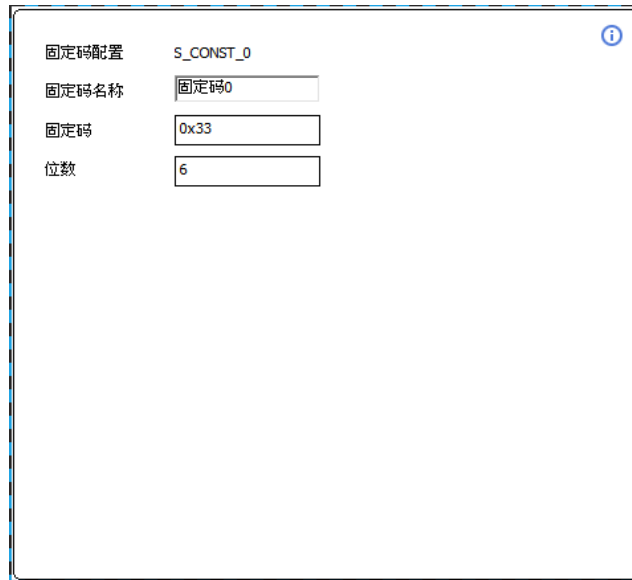
- 校验码，校验码 2



校验码配置

校验码在发码中用于校验使用。配置校验码可以自由选择已配置的发送数据做表达式因子，然后加入表达式计算校验码。表达式可以使用加减乘除做计算。校验码 2 仅用于启用第二组码后做第二组发码的校验计算。校验码可以设定为 1~8 位长度。简易型 LCD 型遥控器的 ROM 和 RAM 资源较少，不建议在校验码上做乘除法或其他复杂的运算。

- 固定码



固定码配置

固定码在发码中用于校验使用，是纯数据内容，最多可配置 10 组。配置固定码只需填入固定码内容和固定码位数。固定码可以设定为 1~8 位长度。

● 定时开，定时关

对应程序变量	S_COUNTDOWN_TIMER_ON
功能名称	定时开
占用位数	1

定时开关机关联数据配置

定时开和定时关在发码中用于对定时开机和定时关机做确认位操作，是纯数据内容。如果启用此数据功能，遥控器程序会判断定时开机和定时关机功能是否启用，对相应的发送位置 1。定时开和定时关可设定为 1~8 位长度。

● 开机计时，关机计时

对应程序变量	S_COUNTDOWN_TIMER_ON_COUNTER0, S_COUNTDOWN_TIMER_ON_COUNTER1
功能名称	开机计时
占用位数	16

定时开关机剩余计时数据配置

开机计时和关机计时在发码中用于向设备发送定时剩余时间，是纯数据内容。开机计时和关机计时功能分别对应两个 16-bit 变量。当定时功能的显示模式设置为 24.5 或 24 模式时，开机计时与关机计时寄存器存入的是剩余总分钟数，位长建议设定为 11~16 位。当定时功能的显示模式设置为模式 3 时，默认低 8 位为分钟数，高 8 位为小时数，位长固定为 16 位。

如果用户已经使用遥控器设定过开机或关机的定时后，半小时或 1 小时内再次使用遥控器发送其他功能（如升档或切换模式），那么两次发送的遥控数据中定时数据是不变的，这里在发送数据中加入开机计时与关机计时，把遥控器 MCU 计算的剩余的分钟数据，可以保证已设定了设备的开机或关机的定时时间不变。

● 自由数据

自由数据是发送流程中的一组可变数据，如格力的空调遥控器在开启上下扫风或左右扫风两者任意一个功能后，遥控器发码流程中有一位被置 1 做确认。

平台自由数据仅产生一个发送流程中的数据变量，数据默认状态是 0，数据的变换操作需要由用户在产出程序后用二次开发的方式实现。自由数据可设定为 1~8 位长度。

2.2.3 功能配置



发送数据配置

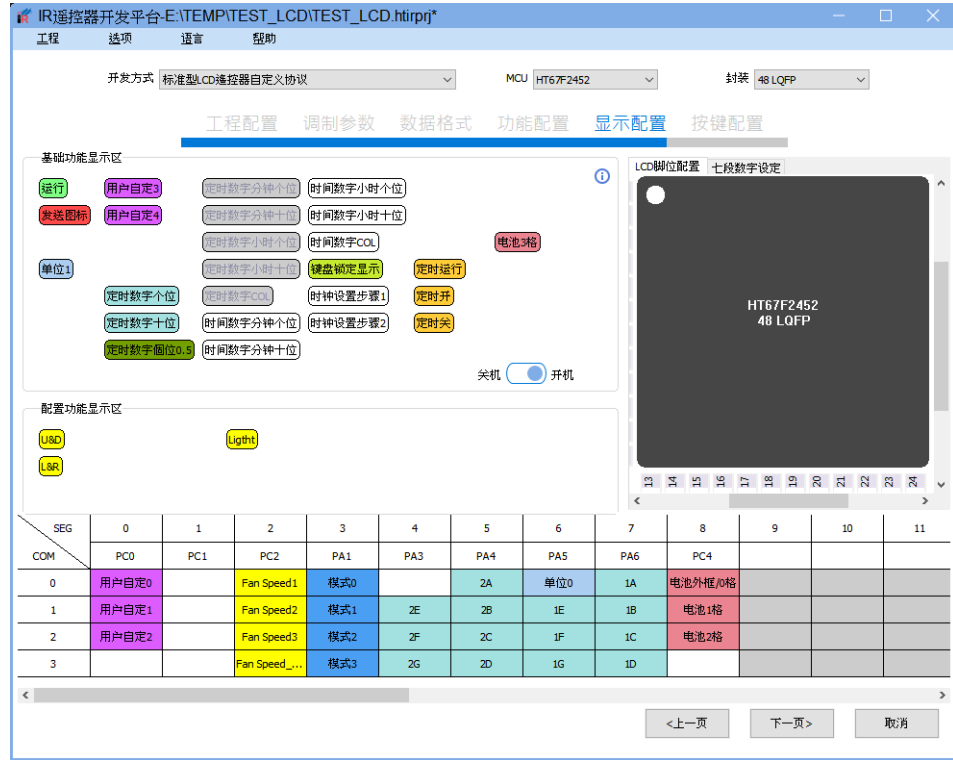
通用数据可显示的数据为 0~99，可在功能配置中限定范围，例如一般空调调温范围大多在 16°C~30°C，热水器大多在 30°C~80°C。简易型遥控器提供一个单位的显示数据，标准型遥控器提供两个单位的显示数据。

电量检测功能以检测电池电压的方式实现，可配置 2~4 档位，启用电压检测后，检测到电压低于最小档时，遥控器不做 IR 发射功能。

时钟显示可支持 12 小时和 24 小时可选的方式显示。

2.2.4 LCD 配置

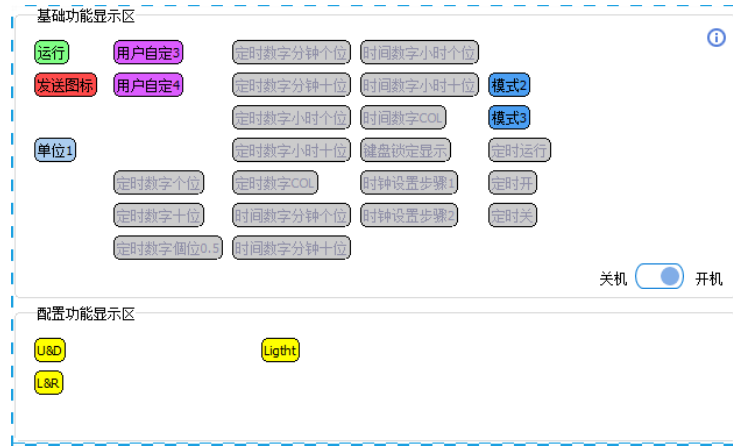
开发平台对 LCD 的显示功能以 LCD RAM 对应 LCD 面板图纸表格的方式帮助用户开发。LCD 型遥控器仅支持 1/4 duty, 1/3 bias 模式的 LCD 驱动。



LCD 显示整体配置

LCD 配置页面的左上方是 LCD 待配置内容区，待配置区分为基础功能配置区和自定义功能配置区两部分。

LCD 显示待配置区



LCD 待配置区

待配置区已启用相关功能的显示段绘制为彩色，可拖选到下方 COM/SEG 表格的配置区，未启用功能的显示段绘制为灰色，不可使用。用户自定义的方块默认是按下开机动作自动点亮，关机不亮，如果需根据其他功能的变化做关联的显

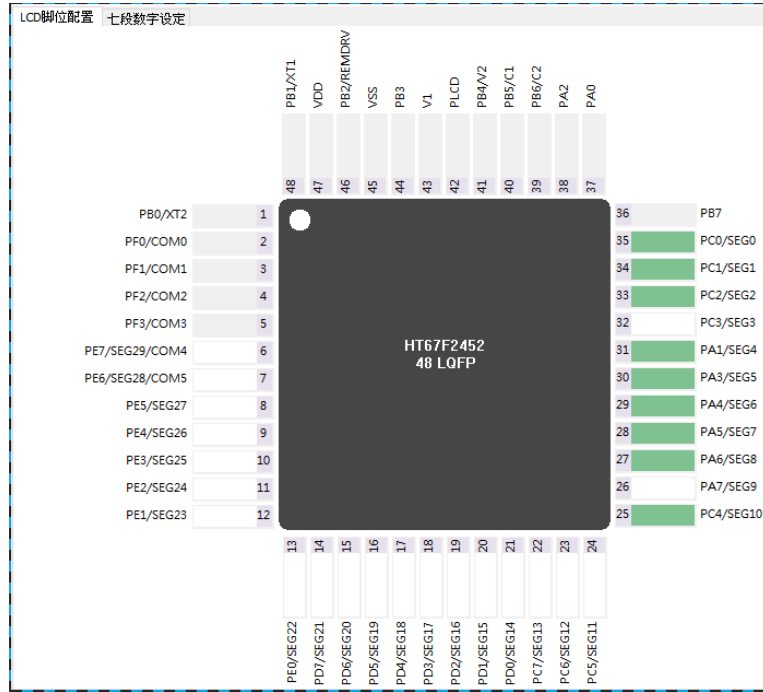
示动作，可在产出工程之后的二次开发里变更用户自定的显示操作。

自定义功能显示区由用户配置的自定义功能，并根据每个功能的数据变化格式产出相关可拖选的数据显示方块。

启用的显示方块拖到下方的 RAM 映像配置区即可在遥控器中启用对应的显示功能。

开机和关机状态的切换开关，可以查看在开机和关机状态显示的段点。

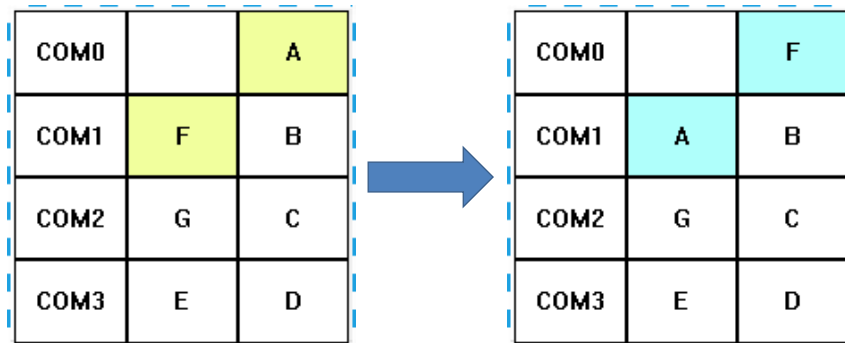
LCD 引脚与数字调整区



LCD 引脚与数字调整区

右方是 LCD 引脚配置和数字调整区。LCD 引脚配置页里可以配置 MCU 驱动 LCD 的 SEG PIN，在 LCD 引脚配置区默认 COM PIN 配置为 4 COM 模式，SEG PIN 由用户点击白色的引脚框即可启用，启用后的引脚变为绿色。

七段数字设定页里可以配置数字显示，软件支持 4 个 COM 和 2 个 SEG 组成一个 A~G 的七段数字，并且可以在数字调整区通过调整段排列方式兼容不同的 LCD 屏幕。操作方法如下图，鼠标先点击 A，再点击需要交换的 F，即可将 A、F 两段的位置互换，其他段位也是按此方式通过点击方式换到对应的驱动位置。



LCD 数字调整区变换方式

RAM 映射区

SEG \ COM	0	1	2	3	4	5	6	7	8	9	10	11
	PC0	PC1	PC2	PA1	PA3	PA4	PA5	PA6	PC4			
0	用户自定义0		Fan Speed1	模式0		2A	单位0	1A	电池外框/0格			
1	用户自定义1		Fan Speed2	模式1	2E	2B	1E	1B	电池1格			
2	用户自定义2		Fan Speed3	模式2	2F	2C	1F	1C	电池2格			
3			Fan Speed...	模式3	2G	2D	1G	1D				

LCD RAM 关联表

RAM 映射区的左方列是 4 个 COM。上面一排是 SEG 引脚的配置，启用为 SEG PIN 的引脚会从左到右自动配置到 SEG 栏中，已配置的 I/O 可以用拖动的方式调整配置。

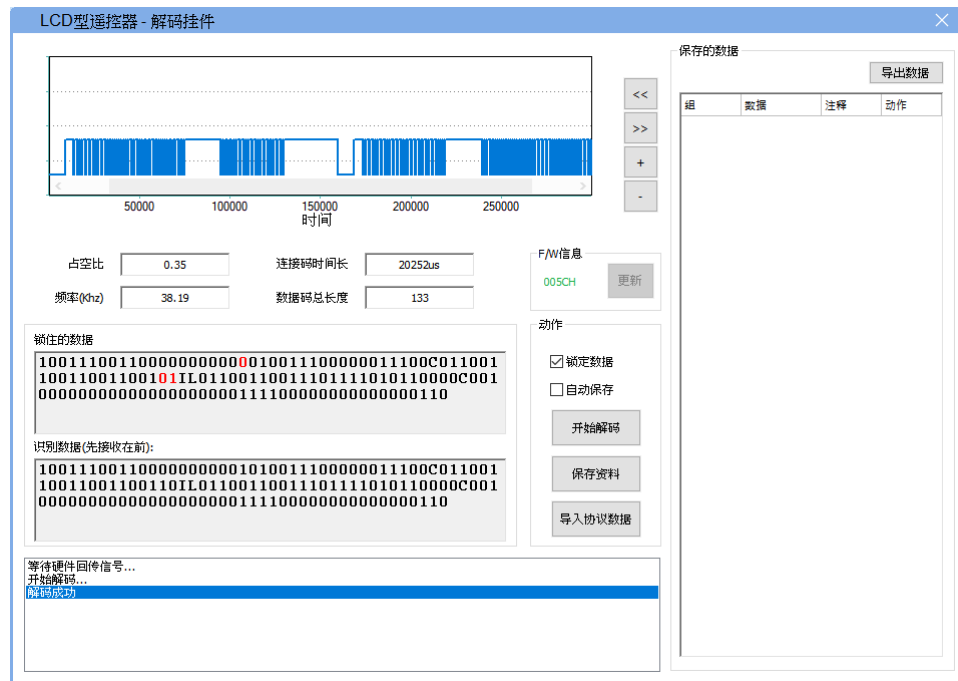
数字显示共有 7 段，拖动数字到 RAM 区软件自动分配两个连续的 SEG，并占用 LCD RAM 内 7 个 bit 的空间。已分配了 RAM 的数字，可以拖动显示数字占用的 RAM bit 到其他空置的 RAM 位置。

部分 LCD 玻璃有十位数字不需要做 F 段显示的，可以在已配置的 RAM 映像区右键选择移除。

2.2.5 学码 / 解码挂件

设计 LCD 型遥控器，可以搭配 LCD 型遥控器解码挂件协助开发，解码功能需要 ESK-IRRC-R00 解码板连接电脑使用。

如下图，软件会按照一般的红外接收头的输出方式绘出发码波形，有载波的 MARK 绘制为低电平，无载波的 SPACE 绘制为高电平。波形图下方为解析出来的发码参数，如载波的占空比、协议类型和频率。解析的 I/O 数字软件自动填入到识别数据中。动作区勾选锁定数据可以将解码的数据加入锁定区，再解码下一个按键时，识别的数据会自动与锁定的数据做对比，新数据与锁定数据有差别的地方软件会把对应的数据用红字提示变化。



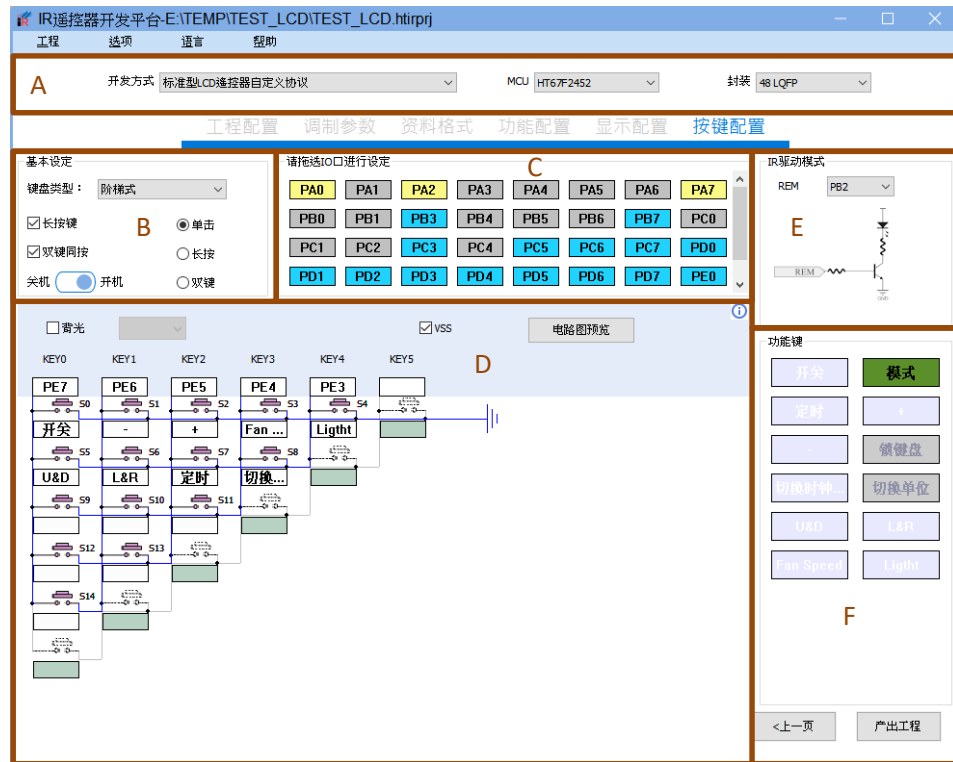
LCD 型遥控器学码模式

LCD 型遥控器的学码 / 解码仅用于学习 PDM 模式的 LCD 型遥控器发码，默认将 SPACE 较长的码设置为 1，SPACE 较短的码识别为 0。

导入协议数据功能仅作为学码的数据记录，为了方便用户做参数的记录和对比，不参与数据格式或按键配置。

2.2.6 按键与驱动

LCD 型遥控器开发到最后一步是按键配置。如下图，A 区是方案类型和 MCU 信息区，B 区是基本设定区，C 区是 I/O 待选区，D 区是按键动作配置区，E 区是 IR LED 驱动配置区，F 区是功能待选区。



按键配置区

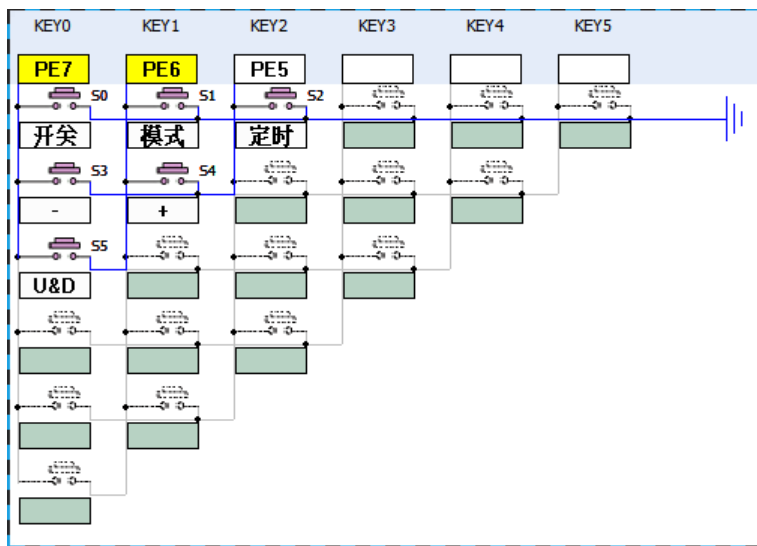
2.2.6.1 按键与功能

配置页面上方 A 区为 MCU 信息区可更改遥控器 MCU 的封装。B 区为基本设定区，可选择阶梯或者矩阵键盘，并且可以启配置键盘长按键或双键同按是否启用。C 区是 I/O 选择区，灰色为已使用或已被占用 I/O。D 区是 I/O 配置区，通过拖选的方式将 C 区 I/O 图标拖入 D 区的 I/O 配置位置。D 区的背光功能是键盘或显示屏的背光灯，任意按键按下后亮灯 4S。当 I/O 配置到键盘之后，即可在 E 区启用可配置的功能位，将 F 区按键功能用拖动的方式放到按键功能方格即可启用功能。

开机或关机状态切换的按钮，可以切换查看和配置开机或关机状态配置的功能。

遥控器的主要功能都是配置为短按键的方式操作，少部分功能 (如连续加减，设置时钟，锁键盘，切换单位) 则在长按键或双键同按操作中实现功能。

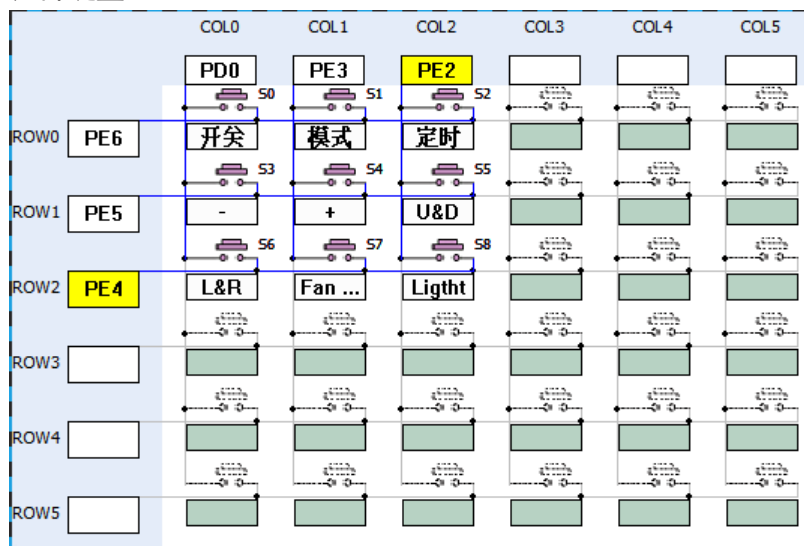
2.2.6.2 阶梯键盘



LCD 型遥控器阶梯按键

如上图，当键盘类型选择了阶梯式，I/O 配置到按键配置表之后，阶梯键盘表随即自动生成。勾选 VSS 加入电路时，按键最多，生成按键数计算公式为 $K = N \times (N+1) / 2$ ，在不勾选 VSS 加入电路时，可支持双键同按，生成按键数计算公式为 $K = N \times (N-1) / 2$ ，N 为已配置 I/O 的个数。

2.2.6.3 矩阵键盘



LCD 型遥控器矩阵按键

如上图，当键盘类型选择了矩阵式，I/O 配置到按键配置表 ROW 和 COL 之后，矩阵键盘表随即自动生成。可生成的最大键数计算公式为 $K = \text{ROW} \times \text{COL}$ 。

2.2.6.4 IR 驱动

IR 驱动根据 MCU 功能配置，如果 MCU 本身内置了 IR 专用驱动脚 REM/REMDRV，软件则强制使用专用的驱动脚驱动 IR LED。如果 MCU 不含 IR 专用驱动脚，则可选择任意一路 PTM 输出脚以 REM 方式输出 IR 驱动波形。

三、开发平台硬件

IR 遥控器开发平台配套的开发板现有可用组件如下：

开发板 / 开发板模块	支持 MCU	功能 / 应用	支持开发模式
*ESK-IRRC-T00	HT68R2420	136 键阶梯式一体式开发板	一般型遥控器
*ESK-IRRC-T01	HT67F2432	15 键阶梯式一体式开发板	简易型 LCD 型遥控器
ESK-IRRC-R00	—	IR 遥控器解码板	所有模式解码
ESK-IRRC-TMT00	HT68R2420	带 IR 发射模块化开发板主板	一般型遥控器
ESK-IRRC-TM00	HT67F370	模块化开发板主板	标准型 LCD 型遥控器
ESK-IRRC-TM01	HT69F340	模块化开发板主板	标准型 LCD 型遥控器
ESK-IRRC-TM02	HT69F350	模块化开发板主板	标准型 LCD 型遥控器
ESK-IRRC-TM03	HT69F360	模块化开发板主板	标准型 LCD 型遥控器
ESK-IRRC-TM04	HT67F2452	模块化开发板主板	标准型 LCD 型遥控器
ESK-IRRC-TMK00	BS67F2432	带触控按键模块化开发板主板	简易型 LCD 型遥控器
ESK-IRRC-TDT00	—	带显示和 IR 发射模块化开发板	简易型 LCD 型遥控器 标准型 LCD 型遥控器
ESK-IRRC-TK00	—	24 键矩阵式按键模块化开发板	一般型遥控器 简易型 LCD 型遥控器 标准型 LCD 型遥控器
ESK-IRRC-TK01	—	136 键阶梯式按键模块化开发板	一般型遥控器 简易型 LCD 型遥控器 标准型 LCD 型遥控器

Note: * 一体式开发板板上芯片使用实体 MCU，不支持仿真功能。

3.1 遥控器开发板

IR 遥控器开发平台遥控器的开发板有两种类型，分别是一体式开发板和模块式开发板。一体式开发板将 MCU，发射，按键，显示 (仅限于 LCD 型遥控器开发板) 集成在同一块 PCB 上的开发板。模块式开发板对每个 MCU 的主控，发射，按键，显示 (仅限于 LCD 型遥控器开发板) 功能拆分成几个不同功能的模块板，可以由用户按照开发项目的功能需求组装后再做开发。

3.1.1 一体式开发板

一体式开发板 ESK-IRRC-T00 是主控为 HT68F2420 的一般型遥控器开发板。用 16 个 I/O 引脚组成 136-key 阶梯矩阵键盘，可仿真任意一种 I/O 组合的按键。

一体式开发板 ESK-IRRC-T01 是主控为 HT67F2432 的 LCD 型遥控器开发板。5 个 I/O 引脚组成 15-key 的阶梯矩阵键盘。20 个 I/O 引脚配置成 4×16 的 LCD 驱动。

具体电路可在软件选项中打开。



一体式开发板

3.1.2 模块式开发板

模块式的开发板可以选择不同的母板，再搭配遥控器所需的发射，显示和按键这几个功能组合成一套完整的遥控器方案。具体的搭配方案如下：

开发类型	母板	开发搭配组件 1	开发搭配组件 2
一般型遥控器	ESK-IRRC-TMT00	ESK-IRRC-TK00 ESK-IRRC-TK01	
标准型 LCD 型遥控器	ESK-IRRC-TM00 ESK-IRRC-TM01 ESK-IRRC-TM02 ESK-IRRC-TM03 ESK-IRRC-TM04	ESK-IRRC-TK00 ESK-IRRC-TK01	ESK-IRRC-TDT00
简易型 LCD 型遥控器	ESK-IRRC-TMK00	ESK-IRRC-TDT00	

开发板具体的连接器连线可参考电路图。

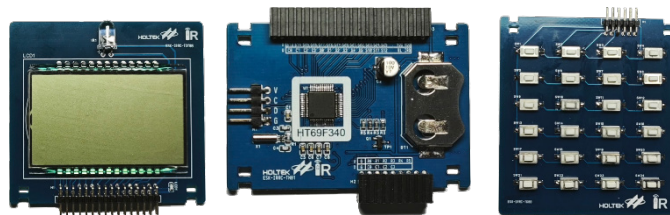
模块式开发板按照一般遥控器的应用将各个可替换功能分为显示，主控和按键三部分。

一般型遥控器不含显示，只需要把主控板和按键板组合起来之后即可做一般型遥控器的开发。



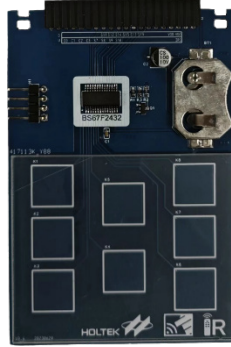
模块式一般型遥控器开发板

LCD 型遥控器的开发需要将显示板，主控板和按键板组合起来做 LCD 型遥控器开发。



模块式 LCD 型遥控器开发板

BS67F2432 是触控型的遥控器 MCU，开发板 ESK-IRRC-TMK00 为保证触控调试有更好效果，触控按键与主控集成在同一块板上。



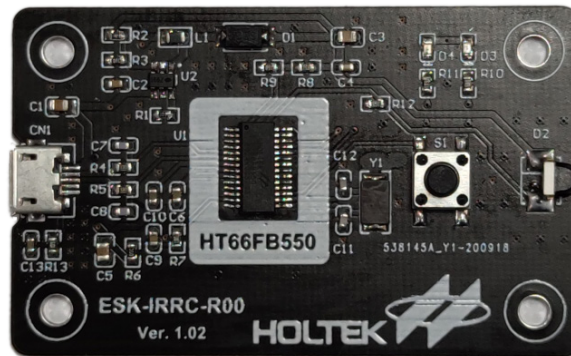
模块式 LCD 型遥控器开发板 ESK-IRRC-TMK00

一体式开发板使用的是不含仿真功能的 MCU 芯片，模块式开发板使用的是仿真芯片，需注意仿真芯片在做仿真时，无法使用 PA0 和 PA2 两个 I/O。

标准型 LCD 型遥控器系列开发板，电路默认使用 C type 驱动，修改开发板背面的短路跳线，可配置不同电压的 LCD 液晶屏。具体短路连线与 C 型偏压电压的关系请参考 MCU 规格书的 LCD 章节。

3.2 解码板

解码板 ESK-IRRC-R00 抓取红外载波信号并通过 USB 上传电脑，将解码内容显示在开发平台软件上。ESK-IRRC-R00 解码板为 HID 设备，在连接电脑使用时不需要安装任何驱动。

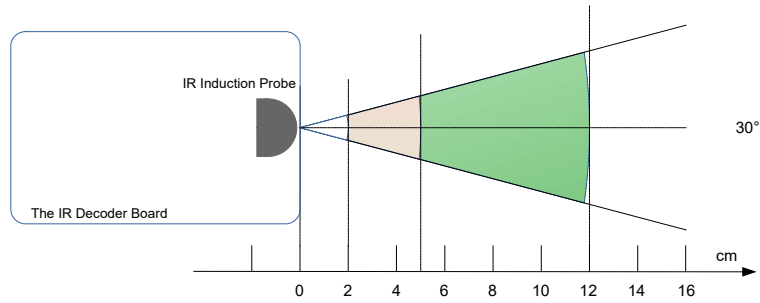


ESK-IRRC-R00 解码板硬件图

解码板接入电脑之后，红绿灯会同时亮起，当电脑识别 USB 设备成功之后，解码板每 2 秒一次绿灯闪烁。

在软件上打开解码挂件后，按下软件的解码按钮或者解码板上的按键，解码板显示灯改变为红灯常亮状态开始解码。

解码板右方为接收感应探头，用于感应红外遥控信号。



接收角度俯视图

为保证解码板能在较佳的工作状态，使用时建议遥控器的发射头距离解码板的感应头对射的角度上下不大于 ± 10 度，左右不大于 ± 15 度，一般的较佳接收距离在 5cm~12cm 之间（约 1 手掌宽）。对于一些使用了功率较低的发射头的遥控器，可以适当减少发码距离。2cm~4cm 之间虽然学码的识别能力较强，但此距离会令识别出的载波占空比会比实际偏大。

四、其他功能说明

4.1 F/W 功耗控制

IR 遥控器开发平台是专为遥控器而设计的软件，所以在遥控器的能耗上也做了特别的优化。

动态功耗控制

在市场上有部分遥控器，在 SPACE 状态的控制方式使用的是 IR LED 全亮的方式控制，这种应用方式不利于电池寿命的保持。本平台软件产生的遥控器程序，可做到仅在 MARK 对应的 Duty 时间驱动 IR LED，SPACE 的控制时间 IR LED 完全处于灭灯状态，可大大延长电池的使用时间。

一般型遥控器因为协议需要发送重复码，在按键被按下之后会持续发码，大部分遥控器一般不设定时间阈值，如果在遥控器意外被其他物体压到按键，则会导致遥控器一直在反复不停的发码最后将电池的电量用尽。本平台的 F/W 中内置了重复码发码次数的判断，当遥控器一直处于按键按下状态，重复码只重复 255 次，达到预设阈值程序会关闭 IR LED 的发射功能，直到等待按键松开才会启用发射功能，以降低功耗。

静态功耗控制

F/W 产生的遥控器程序，一般型遥控器默认 Halt 模式约每 125ms 一次 WDT 溢出唤醒扫描按键。LCD 型遥控器用 time base 唤醒，每隔约 125ms 检测一次按键，每 0.5s 变换一次定时 / 时钟数字闪烁动作，每 1s 一次背光和时钟计时，每 4 秒一次定时计时计算，16s 一次电池电压检测。

4.2 解码识别

解码板目前可识别长度最多 148 个 MARK+SPACE 组合，基本上可覆盖所有的一般型遥控器和大多数 LCD 型遥控器的发码。

因市场上遥控器协议及发码协议众多，而一些特殊且不符合一般编码规则的特殊协议，在平台上无法识别，所以无法保证 100% 的识别并还原出每一种协议。在以后升级版本的软件会继续升级学习功能，尽量满足更多的协议使用。

五、附录 & FAQ

5.1 LCD 型遥控器功能的数据，按键，显示及函数关联表

功能	数据变量	关联按键	关联显示变量	函数定义
开关	b_Power_ON S_POWER_ON	开关	DIS_POWER	FUNC_POWER_SWITCH
模式	S_MODE	模式	DIS_MODE_x(x=0~7)	FUNC_MODE_SWITCH
通用数据	B_Universal_Data S_Universal_Data	+ / -	NUMBER_UNIVERSAL_ DATA_(a)_RAM(b) (a = 0, 1; b = 0, 1)	FUNC_KEY_INC, FUNC_KEY_DEC, FUNC_MODE_SWITCH, FUNC_UNIT_SHIFT_ SWITCH
定时 (模式 1, 2)	REG_Countdown_Timer COUNTDOWN_TIMER_ SHOWONESTEP S_COUNTDOWN_TIMER S_COUNTDOWN_TIMER_OFF S_COUNTDOWN_TIMER_ON S_COUNTDOWN_TIMER_ON_ COUNTER(a) S_COUNTDOWN_TIMER_OFF_ COUNTER(a) (a = 0, 1)	定时 / 开关 / + / -	DIS_COUNTDOWN_TIMER_ PWRON DIS_COUNTDOWN_TIMER_ PWROFF NUMBER_TIMER_(a)_RAM(b) (a = 0, 1; b = 0, 1) DIS_TIMER_HALFHOUR	FUNC_COUNTDOWN_ TIMER_SWITCH FUNC_KEY_INC FUNC_KEY_DEC
定时 (模式 3)	REG_Countdown_Timer_MINUTE REG_Countdown_Timer_HOUR S_COUNTDOWN_TIMER S_COUNTDOWN_TIMER_OFF S_COUNTDOWN_TIMER_ON S_COUNTDOWN_TIMER_ON_ COUNTER(a) S_COUNTDOWN_TIMER_OFF_ COUNTER(a) (a = 0, 1)	定时 / 开关 / + / -	DIS_COUNTDOWN_TIMER_ PWRON DIS_COUNTDOWN_TIMER_ PWROFF NUMBER_TIMER_HOUR_(a)_ RAM(b) NUMBER_TIMER_MINUTE_ (a)_RAM(b) (a = 0, 1; b = 0, 1) DIS_TIMER_COL	FUNC_COUNTDOWN_ TIMER_SWITCH FUNC_KEY_INC FUNC_KEY_DEC
自定义 功能	ST_S_FUNC_x REG_S_FUNC_x S_FUNC_x (x = 0~19)	自定义 功能 / 模式	DIS_MODE_S_FUNC_x DIS_FUNC_x_z DIS_FUNC_x_nz (x = 0~19)	FUNC_x_SWITCH (x = 0~19)
设置时钟	B_TIME_SET_STEP B_TIME_Hour B_TIME_Minute B_TIME_SET_TEMP_HOUR B_TIME_SET_TEMP_MINUTE	设置 / 开关 / + / -	NUMBER_TIME_MINUTE_ (a)_RAM(b) NUMBER_TIME_HOUR_(a)_ RAM(b) (a = 0, 1; b = 0, 1) DIS_TIME_AM DIS_TIME_PM DIS_TIME_SETTING_STEP1 DIS_TIME_SETTING_STEP2	FUNC_TIME_SET_SWITCH ENTER_TIME_SET FUNC_KEY_INC FUNC_KEY_DEC
时钟	B_TIME_Hour B_TIME_Minute B_TIME_Second	无	NUMBER_TIME_MINUTE_ (a)_RAM(b) NUMBER_TIME_HOUR_(a)_ RAM(b) (a = 0, 1; b = 0, 1) DIS_TIME_AM DIS_TIME_PM DIS_TIME_COL	Time_Scond_plus
锁键	b_KEY_LOCK	锁键	DIS_LOCK	FUNC_LOCK_SWITCH
单位	B_UNITSHOW B_Universal_Data	单位	DIS_UNIT0 DIS_UNIT1	FUNC_UNIT_SHIFT_ SWITCH

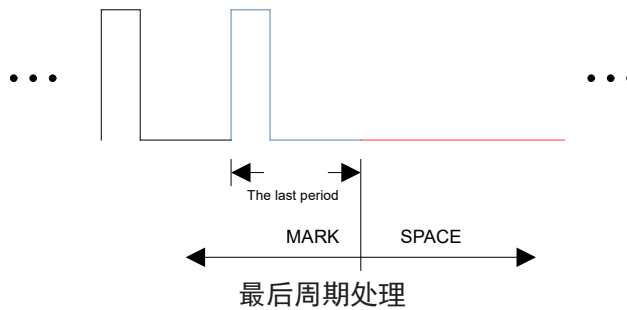
5.2 特殊调制发码说明及应用

在红外遥控器的调制中，有部分红外发码本身符合特定的协议，且可用对应协议指定的解码程序进行解码，但是使用平台的解码板，对这种未能预知协议的解码设备做解码识别仍会出现一些识别问题。具体情况为以下几种。

调制方式	特殊情况	解码异常说明	解决方式
PDM	所有 Bit 全 1 或全 0	因解码出来仅识别出一种位长形态，解码函数统一默认配置为 1。	地址和按键避开使用全 1 或全 0 的发码。
PDM NEC-16	地址配置为 55AAH 或 00FFH	因 55AAH, 00FFH 的高 8 位与低 8 位互为反码并符合 NEC 的地址编码方式，所以解码优先判定识别为 NEC 编码。	使用 NEC-16 协议应避免使用高 8 位与低 8 位互为反码的地址码。
Manchester	所有 Bit 全 1 或全 0	因解码出来仅识别出一种位长形态，解码函数统一默认配置为 1。	地址和按键避开使用全 1 或全 0 的配置；加入 Toggle。
Manchester	首 Bit MARK 在后，结束 Bit SPACE 在前，发码期间所有 Bit 每两位全 10 或全 01 组合	因解码出来仅识别出一种位长形态，解码函数统一默认配置为 1。	加入 Toggle；地址尽量和按键尽量避免 55H, AAH 的配置。
Manchester	自定义协议的头码使用 Manchester 方式调制在解码时无法被有效识别为头码	因 Manchester 方式的头码与 start、toggle 和数据位的码型都一样，所以在识别时是无法区分出此类头码。	此为自定义协议特征，不影响专门对此协议程序的解码。
LCD 型遥控器	所有功能关闭，发码全 0	配置为所有功能均为关闭且不含固定码或固定码全 0，导致解码板仅识别出一种位长。	协议配置多组非连续非 0 固定码。

5.3 MARK 与 SPACE 设计误差说明

接收设备对 MARK 与 SPACE 的解码从遥控器的发码就存在着最大 $\text{period} \times (1 - \text{duty})$ 的误差时间。这个误差是不同协议对于 MARK 的时间参数的不同理解和设计造成的。



例如协议设计载波为 38kHz 且 $\text{Duty}=1/3$ ，如果 MARK 时间设计在 $560\mu\text{s} \sim 579\mu\text{s}$ 的范围内，在示波器或者逻辑分析仪上面只测量到输出脉冲的起始和结束都是 22 个 IR 脉冲，这样就出现了不一样的 MARK 定义时间，却输出同样波形的情况，所以实际 MARK 波形时间的测量值往往都会偏小。

遥控器开发平台解码板对 MARK 最后一个脉冲的识别时间是以一个完整的载波时间记录的如上图，所以存在识别出来的 MARK 波形参数比设计参数偏大的情况，建议在学码完开发导入的 MARK 参数做适当调整再使用。

Copyright© 2024 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。