



红外线 (IR) 遥控器开发平台使用手册



版本：V1.00 日期：2021-06-03

www.holtek.com

目录

一、简介	3
二、开发平台软件	3
2.1 一般型遥控器开发	5
2.2 LCD 型遥控器开发	13
三、开发平台硬件	28
3.1 ESK-IRRC-T00	28
3.2 ESK-IRRC-T01	28
3.3 ESK-IRRC-R00	29
四、解码与学码开发	30
4.1 码形校验	30
4.2 学码开发	31
五、其他功能说明	35
5.1 F/W 功耗控制	35
5.2 解码识别	35
六、附录 & FAQ.....	36
6.1 LCD 型遥控器发码控制与数据表	36
6.2 特殊调制发码说明及应用	36
6.3 MARK 与 SPACE 设计误差说明	37

一、简介

HOLTEK 红外线 (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

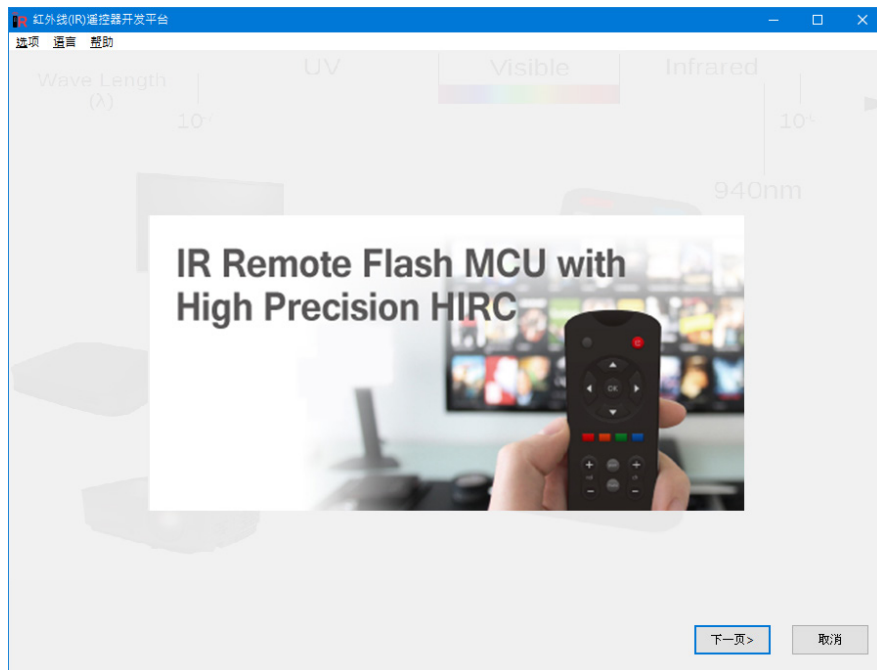


Fig.1 启动界面

软件启动之后如 Fig.1 所示，左上方菜单栏分别为选项、语言和帮助。

选项选单提供 F/W 更新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、芯片 Datasheet、开发板原理图、红外遥控器知识库和平台版本信息。



Fig.2 工程设置

在工程开发界面可以新建工程，开启旧工程和 Example。开发者可通过新建工程的一般型遥控器和 LCD 型遥控器做新遥控器的开发。一般型遥控器开发支持使用的 MCU 是 HT68F2420，支持标准协议开发、自定义协议开发和学码开发；LCD 型遥控器开发支持使用的 MCU 是 HT67F2432，支持自定义协议开发和学码开发。

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：

1. 协议定义
2. LCD 显示内容 (仅限 LCD 型遥控器)
3. 按键及驱动配置
4. 产出 Project (可再做二次开发)，编译产出烧录档

2.1 一般型遥控器开发

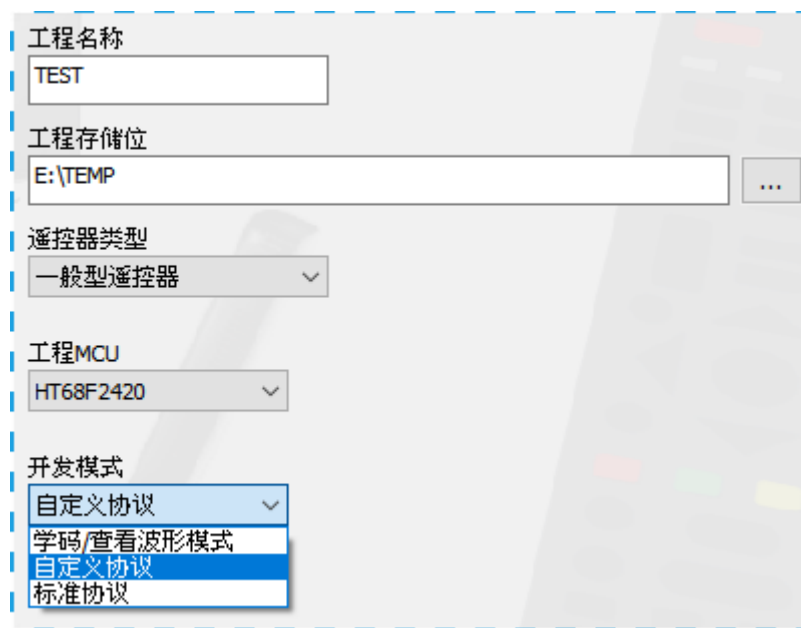


Fig.3 工程设置配置一般型遥控器

一般型遥控器开发支持标准协议开发、自定义协议开发和学码开发三种开发模式。在工程设置页面遥控器类型选择一般型遥控器，软件支持的 MCU 为 HT68F2420。

2.1.1 标准协议开发

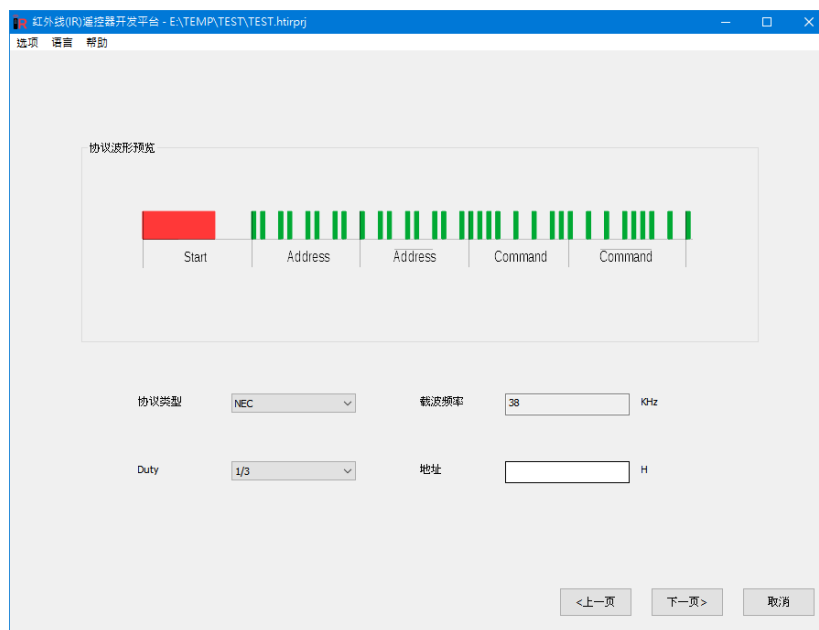


Fig.4 标准协议配置界面

如图 Fig.4 所示，在标准协议开发中，可以使用 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

自定义协议开发分两个标签页配置参数。

第一个参数页是调制方式页。在调制方式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。

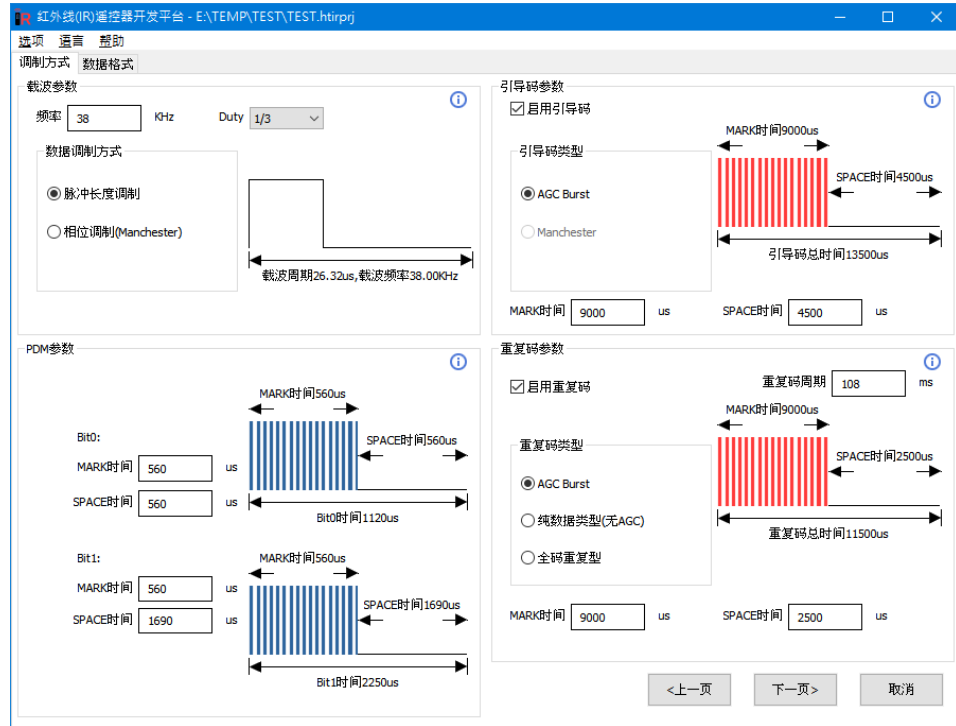


Fig.5 自定义协议配置界面

具体的配置操作方式如下：

载波

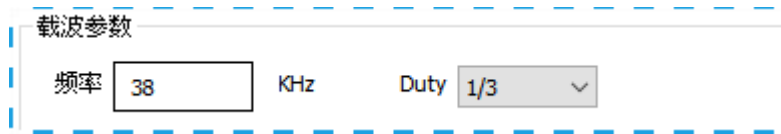


Fig.6 载波参数

如图 Fig.6，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)。

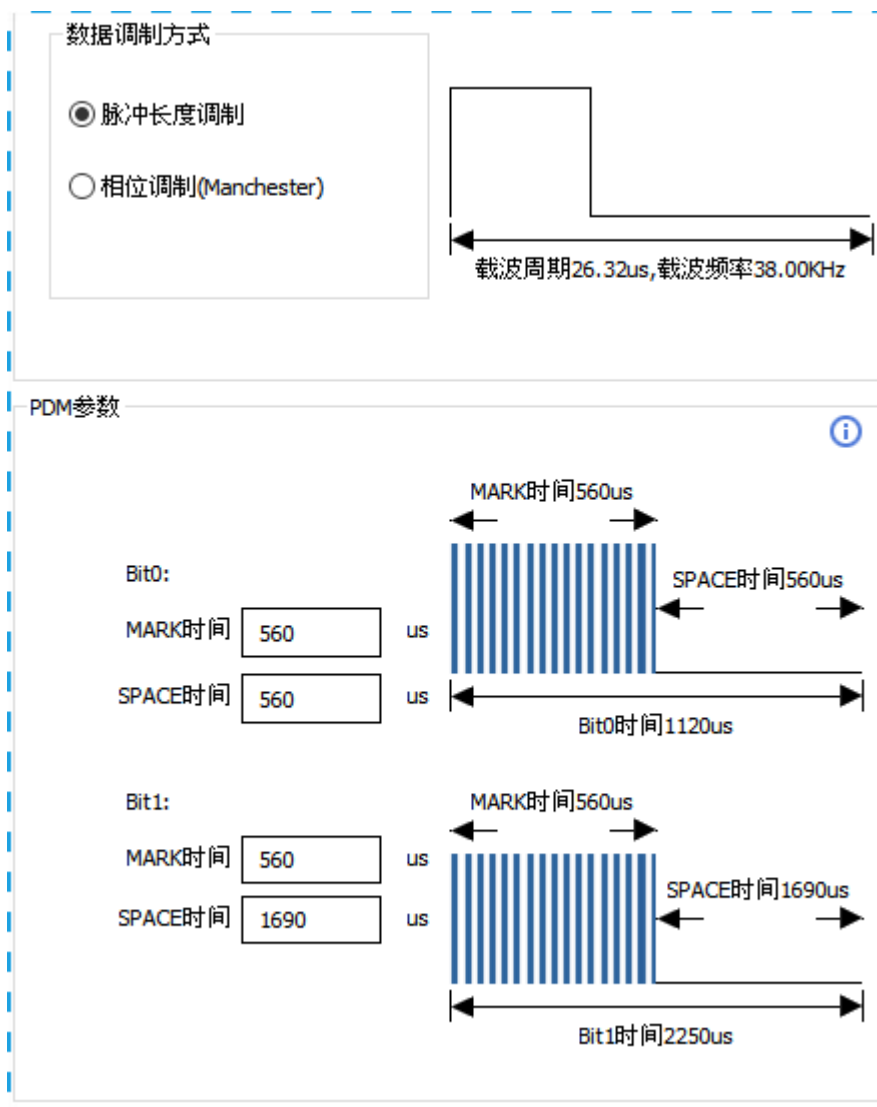


Fig.7 脉冲长度调制 (PDM) 参数

Bit0 和 Bit1 的脉冲长度调制参数的配置图如图 Fig.7。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。

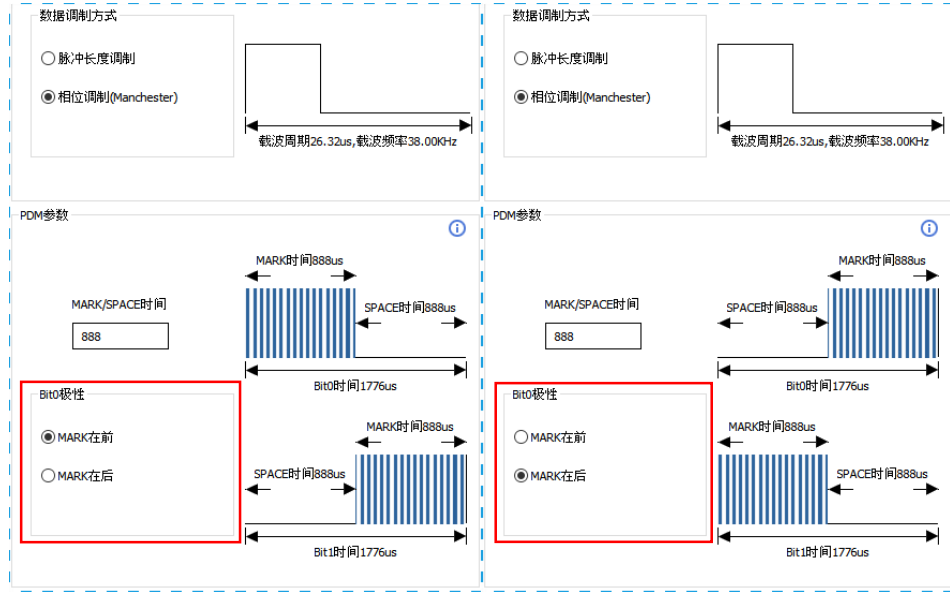


Fig.8 相位调制 (Manchester) 参数

图 Fig.8 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 $300\mu\text{s}$ ~ $2000\mu\text{s}$ 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。

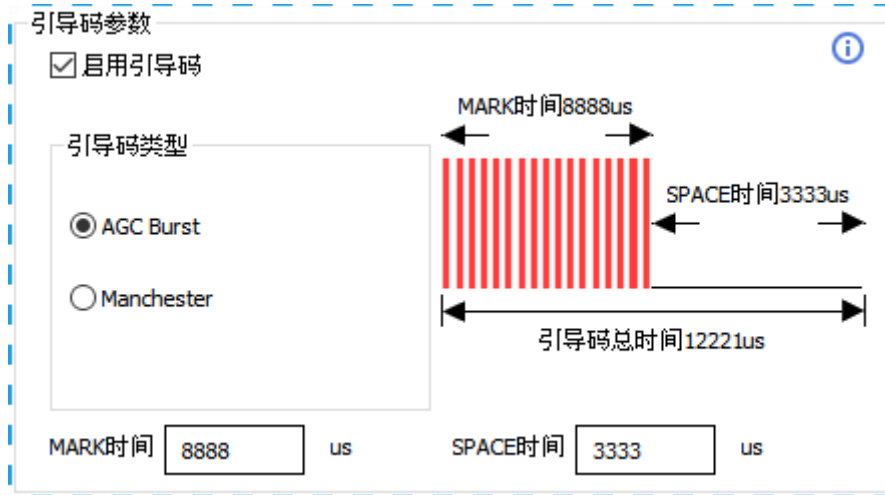


Fig.9 AGC Burst 引导码参数

如图 Fig.9，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。

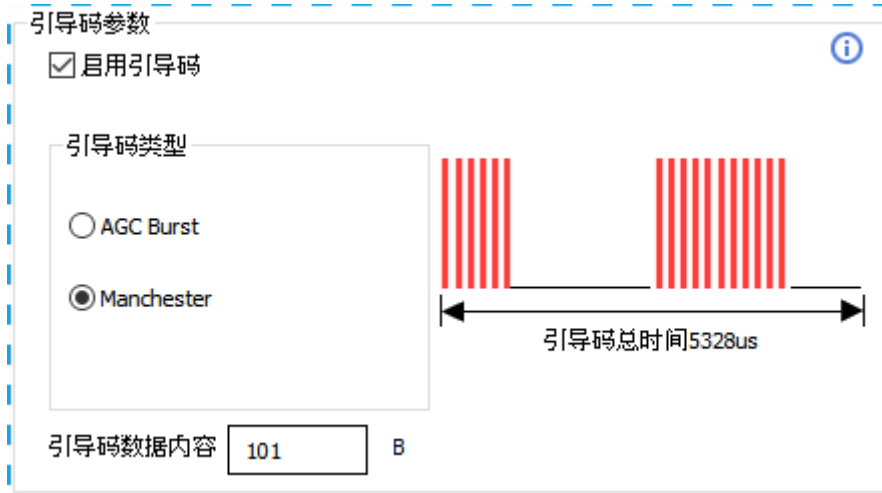


Fig.10 Manchester 引导码参数

如图 Fig.10，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。

如图 Fig.11，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。

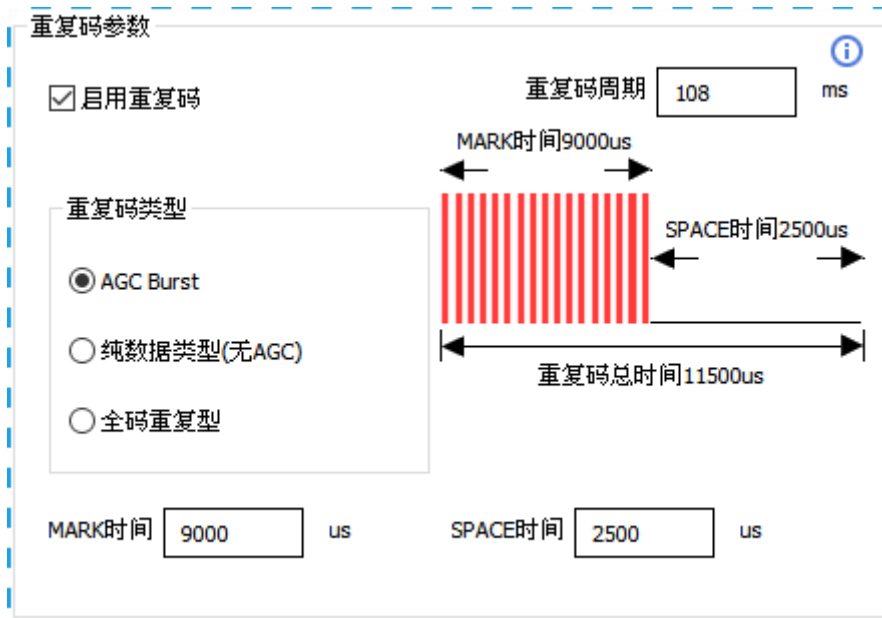


Fig.11 重复码参数

重复码使用 AGC Burst 模式时需要对参数进行配置，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

如图 Fig.12，在第二页标签的数据格式参数中，可以设定地址码的位数、命令码的位数、发送次序、是否需要插入 Start 和 Toggle 位。

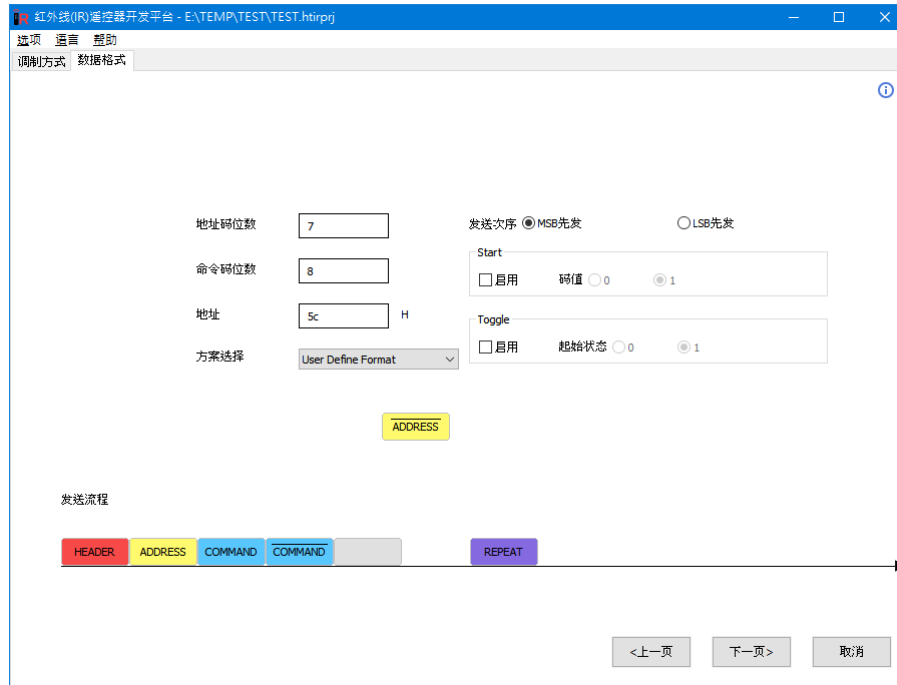


Fig.12 数据格式参数

地址码位数：位数可配置范围为 1~8。

命令码位数：位数可配置范围为 1~8。

地址码：地址码的最大值取决于设置的地址码位数。地址码以 16 进制方式输入。

发送次序：可定义数据先发送高位还是先发低位。

Start 位：用于在地址和命令码前面插入一个固定位。

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长按下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地址码、地址反码、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。当启用 Start 或 Toggle 位时，这两位则固定放在引导码之后，地址码和命令码之前。

2.1.3 阶梯矩阵按键与驱动

标准协议或自定义协议参数配置完成的下一步，是阶梯矩阵按键的配置页，如下图，A 区是 I/O 待选区，B 区是当前配置方案支持的 MCU 封装提示区，C 区是 IR LED 驱动方式选择区，D 区是按键 I/O 配置区，E 区是命令值写入区，F 区是学码开发已保存的按键区。

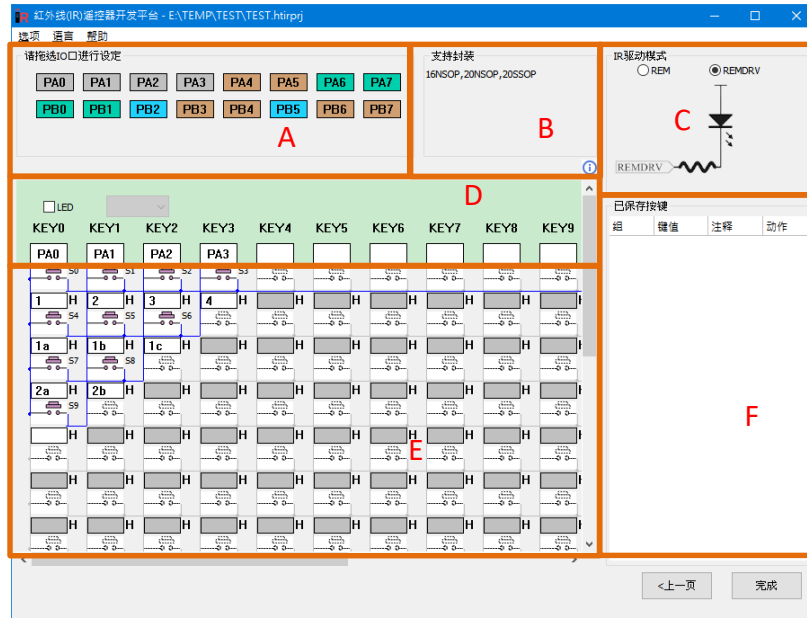


Fig.13 配置按键页

2.1.3.1 按键与 I/O

配置页面左上方 A 区为可使用的 I/O 引脚，其中蓝色引脚图标为 HT68F2420 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。通过拖选的方式将 I/O 图标拖入下方 D 区的按键位置即可完成对 I/O 的配置。B 区会根据目前按键配置显示可用的封装。

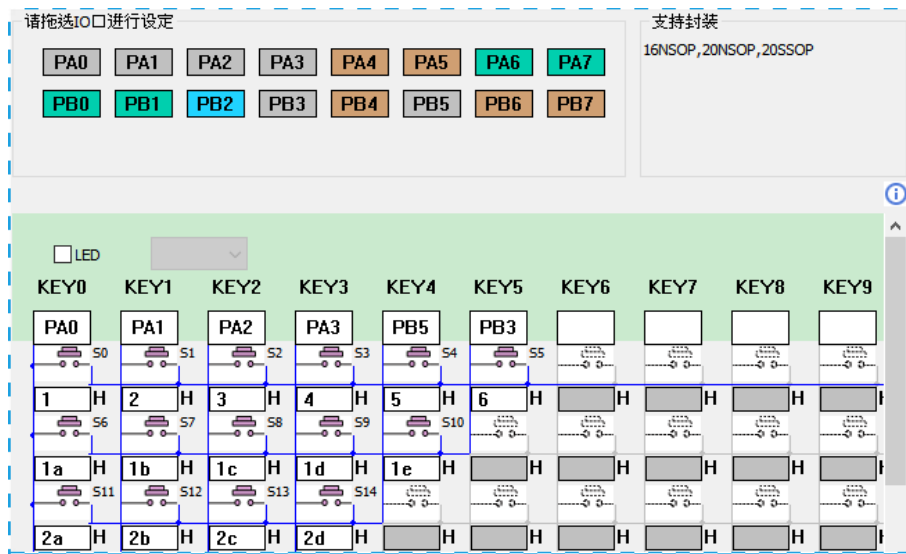


Fig.14 配置按键图

如 Fig.14，当 I/O 配置到按键配置表之后，阶梯矩阵键盘表随即自动生成。可生成的最大键数计算公式为 $K = 1+2+\dots+N$ ，N 为已配置 I/O 的个数，8-pin 最多可配置 15 个按键，16-pin 最多可配置 78 个按键，20-pin 最多可配置 136 个按键。

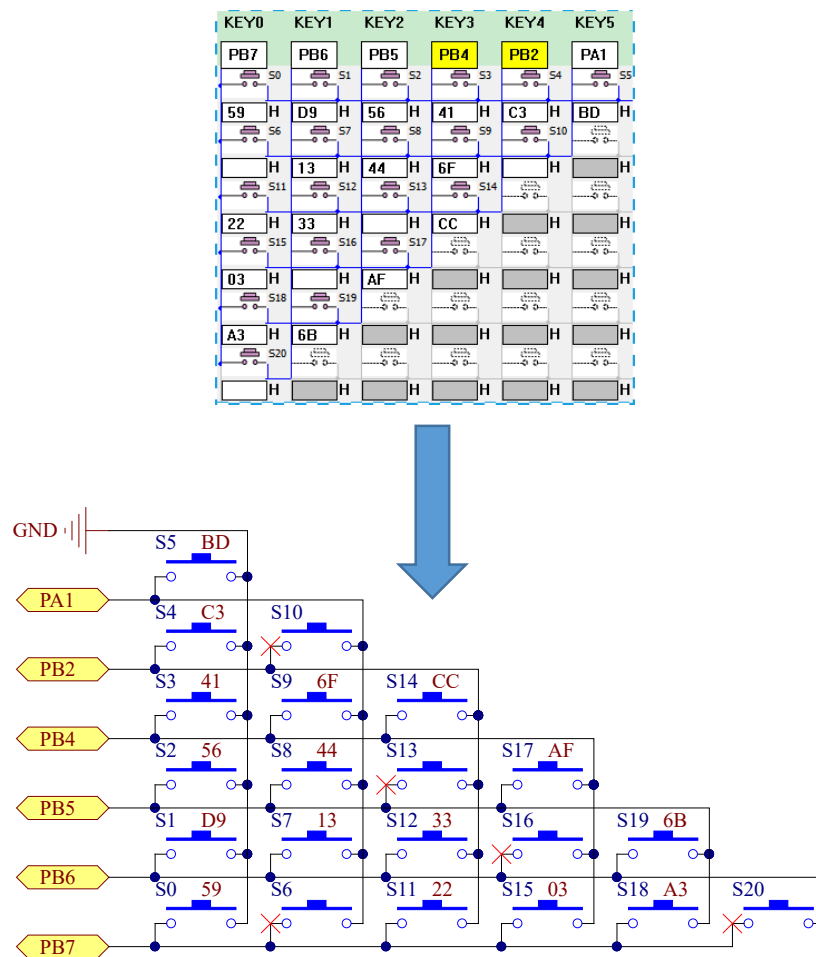


Fig.15 按键配置与原理图关系

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 Fig.15 上半部分按键 S14 的连线连接到 PB4 和 PB2 两个 I/O。

在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。

Fig.15 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 即使未配置键值或未在实体电路上焊按键，但已在生成配置时列入扫描的码表中。未配置按键在软件生成的键值表默认填充 00H。

2.1.3.2 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 Fig.16。

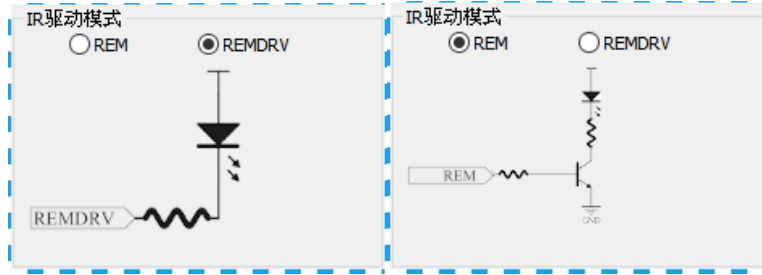


Fig.16 IR LED 驱动配置

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 驱动方式最大可提供 500mA@3V 的驱动电流。如需更大的驱动能力，可使用 REM 的驱动方式外加三极管或场效应管以提高输出能力。

LED 指示灯可以使用 I/O 口驱动，在 D 区配置，驱动 LED 指示灯可以使用任意独立的 I/O 引脚，也可以与按键引脚复用。因为阶梯键盘扫描按键程序会对 KEY0 以外的 I/O 输出低电平，所以为了避免功耗增加和 LED 出现微亮，LED 指示灯与按键复用只能与 KEY0 复用。



Fig.17 指示灯配置

按键配置结束后，点击完成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

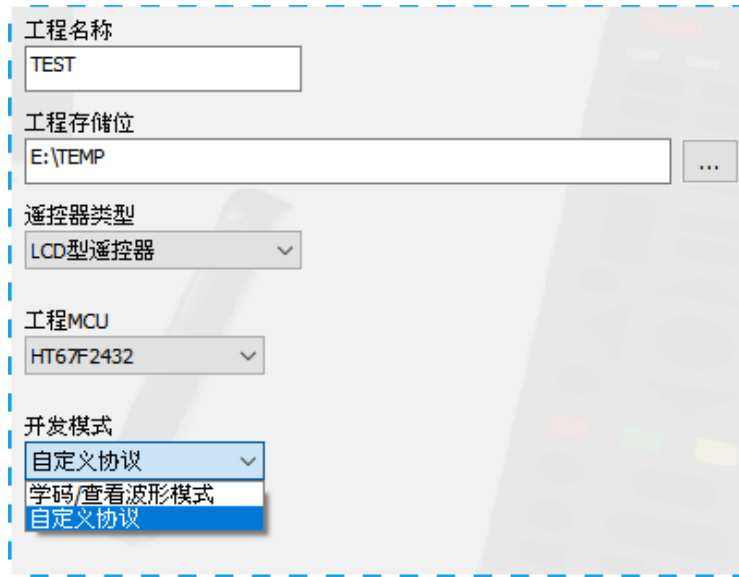


Fig.18 工程设置配置 LCD 型遥控器

LCD 型遥控器开发支持自定义开发和学码开发两种开发模式。在工程设置页面，遥控器类型选择 LCD 型遥控器，软件支持使用的 MCU 为 HT67F2432。LCD 型遥控器在空调遥控产品上应用较多，现软件以空调的功能做配置开发，后续软件升级将加入更多丰富的遥控产品开发类型。

2.2.1 协议定义

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。

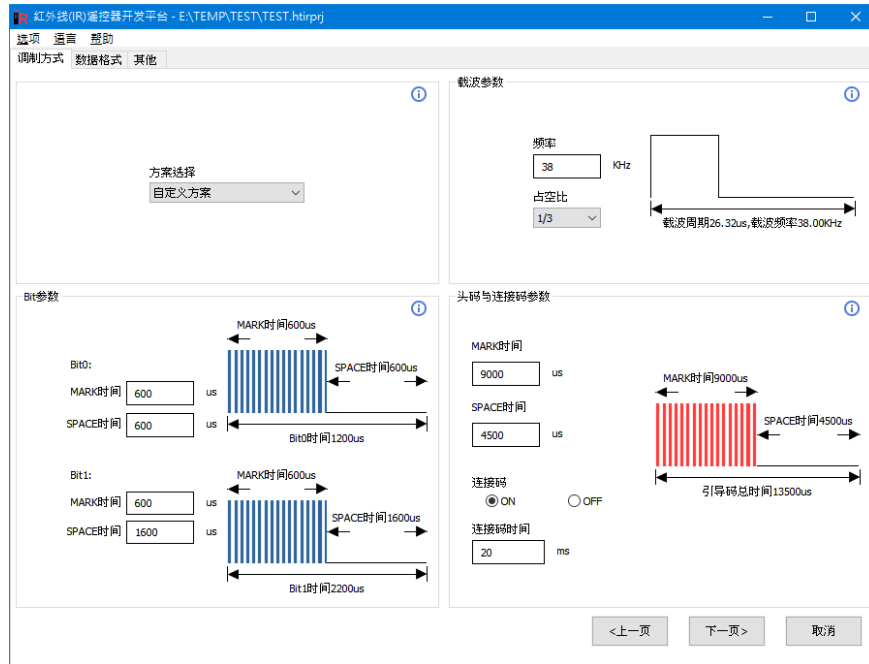


Fig.19 调制方式配置页面

2.2.1.1 调制方式

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

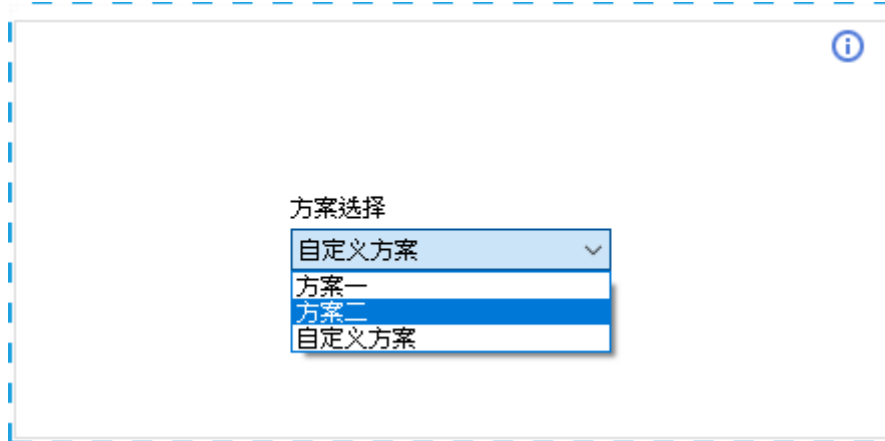


Fig.20 方案选择

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

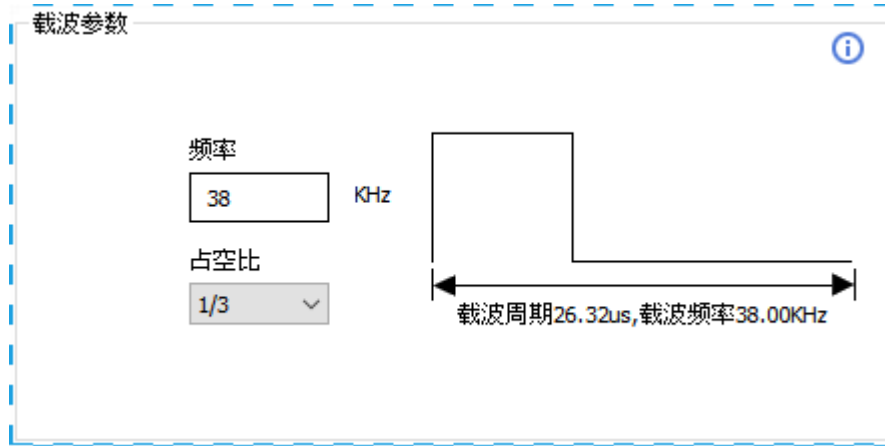


Fig.21 载波参数

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。

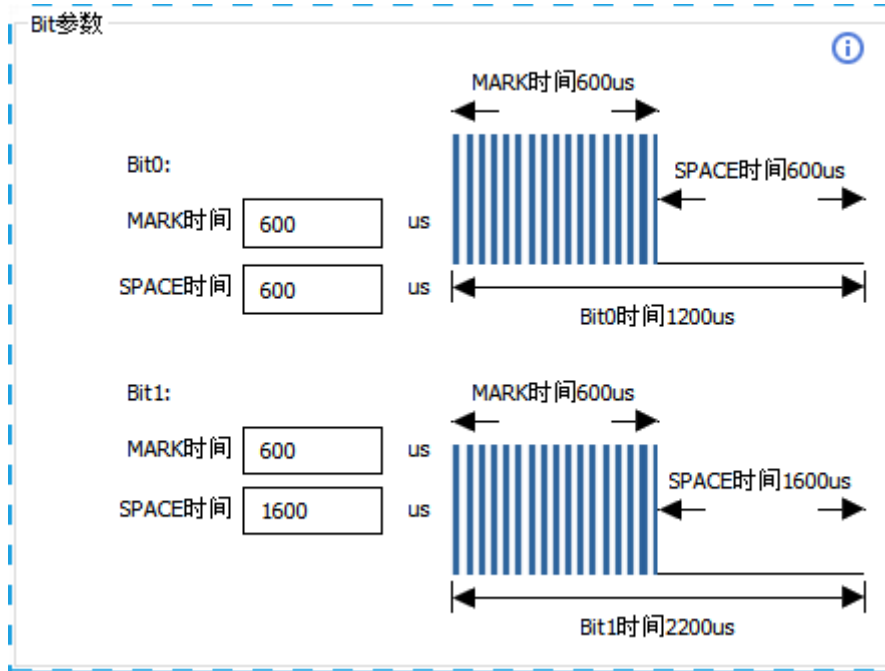


Fig.22 Bit 参数

- 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, 但 MARK 的时长与功耗成正比, 建议合理的配置此参数以保证遥控器电池有更长的使用时间, 设定范围在 4ms~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。连接码既可用于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容易出现 MCU 的 LVR 复位问题。

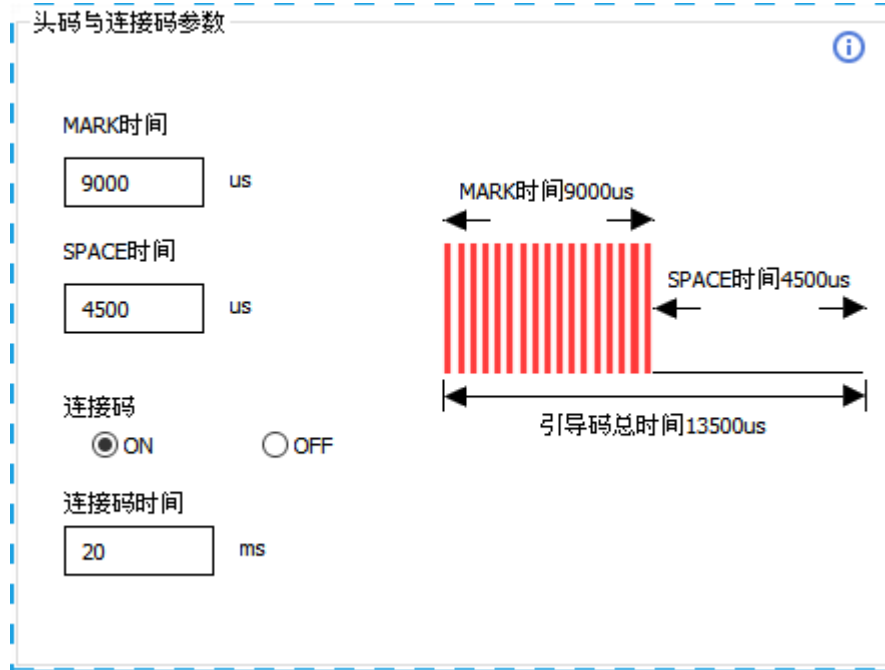


Fig.23 头码与连接码参数

2.2.1.2 数据格式

配置发码的第二步是配置数据格式。LCD 型遥控器所使用的协议, 大多是将 LCD 的显示内容调制成发送数据, 而不同的 LCD 型遥控器产品因显示的内容不同, 会调制成各种不同的发送数据 (参考软件内置红外线遥控器知识库说明)。

LCD 型遥控器在消费类产品中应用得较多的是空调遥控器, HT67F2432 的自定义中, 按照空调遥控器的常用功能设计了自定义数据配置。后续在有增加更多资源的 MCU 后, 将开放更多更自由的功能和配置。

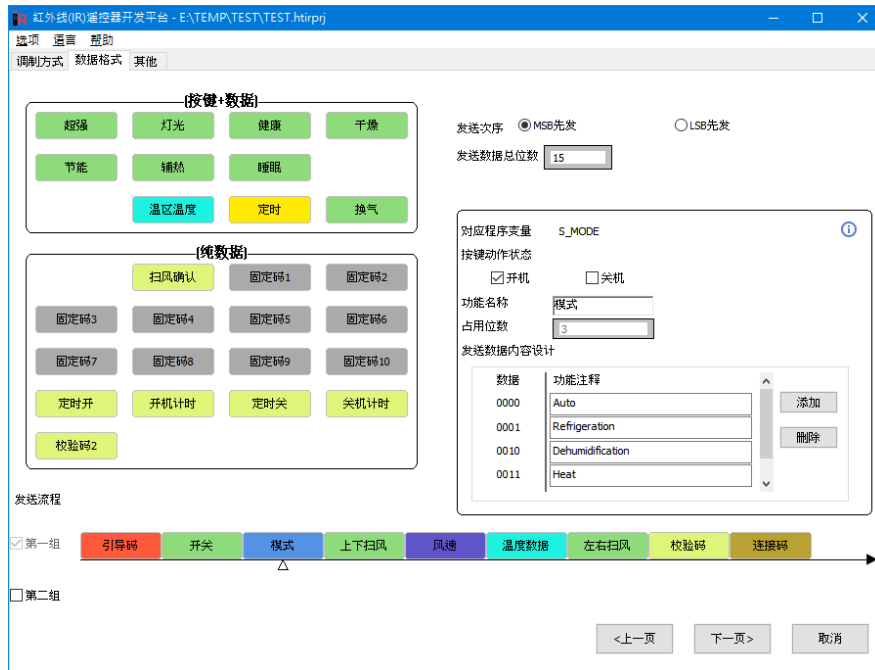


Fig.24 数据格式页

数据格式页主要是对功能数据做配置，在右上角是设置发码发送的次序，可选低位先发 (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，默认引导码以及模式、开关、温度数据、风速这 4 组发码数据固定使用在发送流程中。

左方的功能区按照不同的功能设计成可选用方块图标，遥控器有需要发送的功能数据，可将其拖到下方的发送流程中。左上方是按键加数据，是一组搭配按键功能的发送数据，左下方是纯数据，是一组固定或者随其他按键操作产生状态变化计算出的数据。

当点开任意一组数据，都可以在右方的数据配置栏中配置数据的功能。

右方展示的数据参数有：

对应程序变量：当前配置的数据对应产出工程中的变量；

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；

占用位数：当前配置在发送流程中所占用的位数，默认带按键的数据在配置时不可更改 (可在二次开发时修改)，纯数据可在配置时自定义位数；

模式选择：某些特定功能不需要在所有模式都做按键响应，可以在此项配置中设计成指定的某几个模式可用。可选择的模式根据模式配置中已启用的模式来确定；

发送内容设计：一般的功能在发送内容的配置上是 0 和 1。模式、风速和温区温度这几项功能数据可配置成多档模式。

- 开关

Fig.25 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

Fig.26 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。模式功能配合 LCD 显示的方式是切换段码的显示。

- 上下扫风，左右扫风，健康，换气，灯光



Fig.27 开关式按键配置

这几项功能的配置方式比较类似，一般配置在开机状态执行按键动作。产出的程序默认定义启动是 1，关闭是 0。上下扫风，左右扫风，健康，换气，灯光功能配合 LCD 显示的方式是段码的亮来显示。

- 风速



Fig.28 升档类按键配置

风速功能一般配置在开机状态执行按键动作。在发送数据内容设计中，可以通过添加按钮添加风速的不同档位，最多可以设置 8 档模式，默认 0 档是自动档。同时在右上方的工作模式中，指定模式支持变档动作。风速功能配合 LCD 显示时，做自动档的 0 档是切换显示，其他档做递增显示。

- 超强，节能，睡眠，辅热，干燥

Fig.29 指定工作模式开关类按键配置

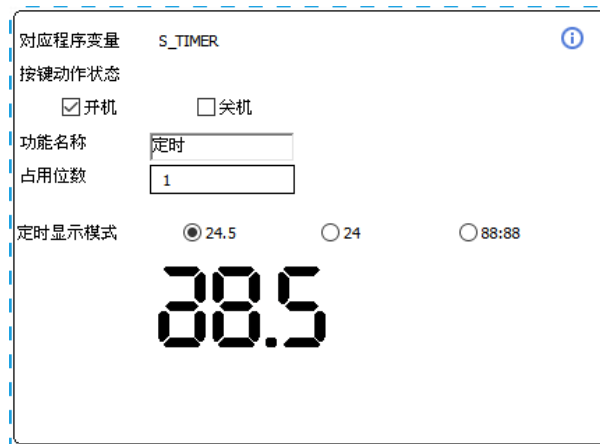
这几项功能的配置方式比较类似，一般配置在开机状态执行按键动作。产生的程序默认定义启动是 1，关闭是 0。同时在右上方的工作模式中，可以选择在何种模式支持变档动作。超强，节能，睡眠，辅热和干燥功能配合 LCD 显示的方式是段码的亮灭显示。

- 温区温度

Fig.30 变档式温区按键配置

温区温度功能一般配置在开机状态执行按键动作。在发送数据内容设计中，可以通过添加按钮添加温区的不同位置，最多可以设置 4 档模式。开发者可以对每个控制状态位的功能做注释。温区温度功能配合 LCD 显示时，0 档无显示，1~3 档位有一段是常亮显示。

- 定时



对应程序变量 S_TIMER

按键动作状态

开机 关机

功能名称 定时

占用位数 1

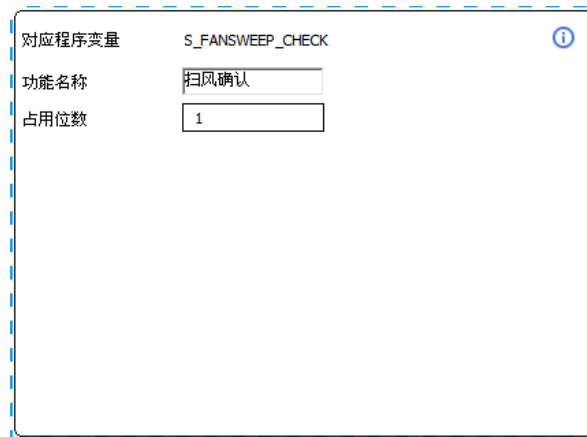
定时显示模式 24.5 24 88:88

28.5

Fig.31 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，定时按键启用定时关机。遥控器处于关机状态，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。定时的显示模式可以选择 24.5，24，88:88 三种显示模式，可以根据不同的 LCD 规格做出合适的选择。定时功能的位长用于存小时或半小时数，建议设定为 6~8 位长度。

- 扫风确认



对应程序变量 S_FANSWEEP_CHECK

功能名称 扫风确认

占用位数 1

Fig.32 扫风数据

扫风确认在发码中用于关联上下扫风和左右扫风两个功能做确认位操作，是纯数据内容。如果启用此数据功能，遥控器程序会判断当上下扫风或左右扫风开启时，将扫风确认发送位置 1。扫风确认可以设定为 1~8 位长度。

- 校验码，校验码 2

Fig.33 校验码配置

校验码在发码中用于校验使用，是纯数据内容。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，然后在表达式中加入计算式计算校验码使用。表达式可以使用加减乘除做计算。校验码 2 是用于启用第二组码时做第二组发码的校验计算。校验码可以设定为 1~8 位长度。

- 固定码 1~10

Fig.34 固定码配置

固定码在发码中用于校验使用，是纯数据内容。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_TIMER_ON
功能名称	定时开
占用位数	1

Fig.35 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。定时开和定时关可以设定为 1~8 位长度。

- 开机计时，关机计时

对应程序变量	S_TIMER_ON_COUNTER0,S_TIMER_ON_COUNTER1
功能名称	开机计时
占用位数	10

Fig.36 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对定时的开和关做确认位操作，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，对相应的变量写入剩余时间。开机计时和关机计时功能分别对应两个 8-bit 变量，可以设定为 9~16 位长度。

2.2.1.3 其他配置

Fig.37 发送温度调整数据配置

由于一般空调调温范围大多在 16°C~30°C 温度范围，内置的遥控器 F/W 在温度的设定上可以做最大值和最小值限定，并且可以通过设定温度扣减基数，来减少发送温度数字的长度，比如 30 度的二进制为 11110B 至少需 5 位表达，扣减掉 16 之后二进制为 1110B，可在发送时少发一位，有减小发送功耗的作用。

大多 LCD 型遥控器在关机状态仍会保持 LCD 处于显示状态，软件内置的遥控器方案在 LCD 显示状态待机可保持待机电流在 10μA 以内，如果在电流上有比较极端的控制需求，可以关闭“关机状态保持 LCD 显示”功能。

2.2.2 LCD 配置

LCD 玻璃是 LCD 型遥控器的零件之一，本软件在 LCD 配置页面集成了大部分 LCD 型遥控器在 LCD 空调应用显示的常用功能和显示方式。

COM0	E	A
COM1	F	B
COM2	G	C
COM3		D

SEG	0	1	2	3	4	5	6	7	8	9
COM 0	<input type="checkbox"/> PC0	<input type="checkbox"/> PC1	<input type="checkbox"/> PC2	<input type="checkbox"/> PC3	<input type="checkbox"/> PC4	<input type="checkbox"/> PC5	<input type="checkbox"/> PC6	<input type="checkbox"/> PC7	<input type="checkbox"/> PD1	<input type="checkbox"/> PD0
1	<input checked="" type="checkbox"/> PB4									
2	<input checked="" type="checkbox"/> PB5									
3	<input checked="" type="checkbox"/> PB6									
4	<input checked="" type="checkbox"/> PB7									

Fig.38 LCD 显示整体配置

LCD 配置页面的左上方是 LCD 待配置内容区，右方是数字调整区，下方是 LCD RAM 映射配置区。

LCD 显示待配置区



Fig.39 LCD 待配置区

在前一步有配置启用的功能和文字都会渲染上颜色表示为可用状态，如果未配置功能则为灰色被限制状态。已配置为启用的显示方块拖到下方的 RAM 映射配置区即可启用对应的显示功能。

数字调整区

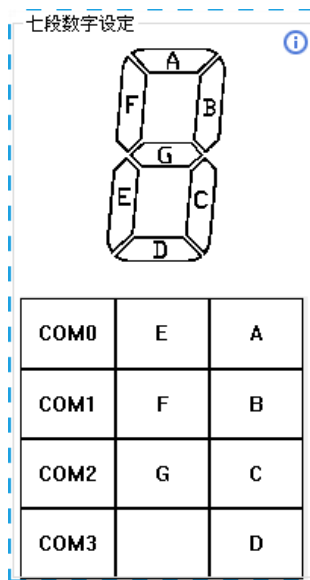


Fig.40 LCD 数字调整区

LCD 型遥控器一般少不了数字显示，LCD 的玻璃规格对 4 个 COM 引脚的设计，大多是把一个数字的 A~G 七段由两个 SEG 控制，而 A~G 七段可能因产品使用不同的 LCD 玻璃规格有不同排列，对于不同的数字显示控制，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。

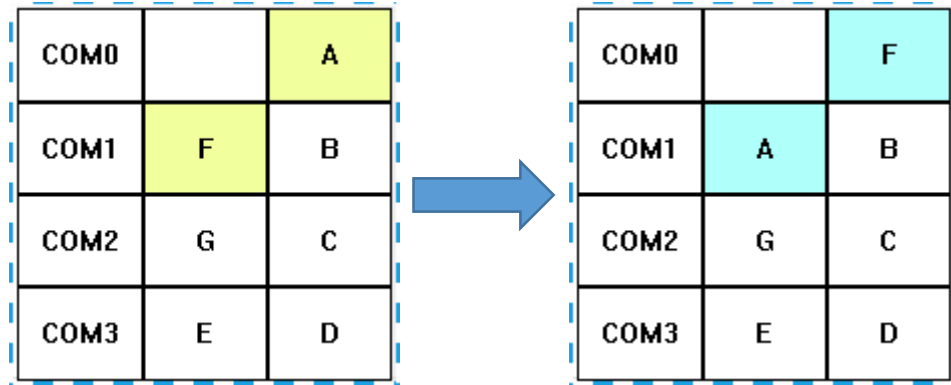


Fig.41 LCD 数字调整区变换方式

RAM 映射区

COM	SEG	0	1	2	3	4	5	6	7	8	9
		<input checked="" type="checkbox"/> PC0	<input checked="" type="checkbox"/> PC1	<input checked="" type="checkbox"/> PC2	<input checked="" type="checkbox"/> PC3	<input type="checkbox"/> PC4	<input checked="" type="checkbox"/> PC5	<input checked="" type="checkbox"/> PC6	<input checked="" type="checkbox"/> PC7	<input checked="" type="checkbox"/> PD1	<input type="checkbox"/> PD0
0	<input checked="" type="checkbox"/> PB4	模式0	上下扫风	风速自动	发送图标			1F	℃	2F	
1	<input checked="" type="checkbox"/> PB5	模式1	左右扫风	风速一档	睡眠		1A	1B	2A	2B	
2	<input checked="" type="checkbox"/> PB6	模式2	风速	风速二档	运行		1G	1C	2G	2C	
3	<input checked="" type="checkbox"/> PB7			风速三档			1E	1D	2E	2D	

Fig.42 LCD RAM 关联表

RAM 映射区的左方列是 COM 引脚的配置，COM 引脚按照 HT67F2432 的 COM0~COM3 放置，此功能为强制启用。上面一排是 SEG 引脚的配置，SEG 引脚按照 HT67F2432 的 SEG0~SEG19 的配置，SEG0~SEG19 分别映射到 HT67F2432 RAM Bank4 的 00H~13H 地址。SEG13 引脚 (PA4) 与 REM 引脚共用，所以 SEG13 的 LCD 驱动功能软件强制禁用。

数字显示共有 7 段，所以拖动数字到 RAM 区软件会自动分配两个连续的 SEG，并占用 RAM Bank4 内 7 个 bit 的空间。如果数字要不同的非连续的 SEG 引脚驱动，也可以再用拖动的方式将任意一组 SEG 拖到其他位置。

对于部分 LCD 显示玻璃有部分的十位数字不需要做 F 段显示的，可以在数字配置到 RAM 映射区之后右键选择移除 F 段。

每个 SEG 下方都有显示对应的 I/O 引脚名称，当显示的段拖到 RAM 配置区时，如有未使能的 SEG，软件会自动使能。如果有 I/O 不做 LCD 驱动使用，应将 LCD 驱动脚的使能关闭，否则此 I/O 在下一步的操作中，这个 I/O 无法用来配置按键。

2.2.3 阶梯矩阵按键与驱动

LCD 型遥控器的协议和 LCD 功能配置完成之后，软件下一步会跳转到阶梯矩阵按键的配置页，如下图，A 区是 I/O 待选区，B 区是当前配置方案支持的 MCU 封装提示区，C 区是 IR LED 驱动方式展示区，D 区是已配置的按键区，E 区是生成键盘按键区，F 功能按键待配置区。

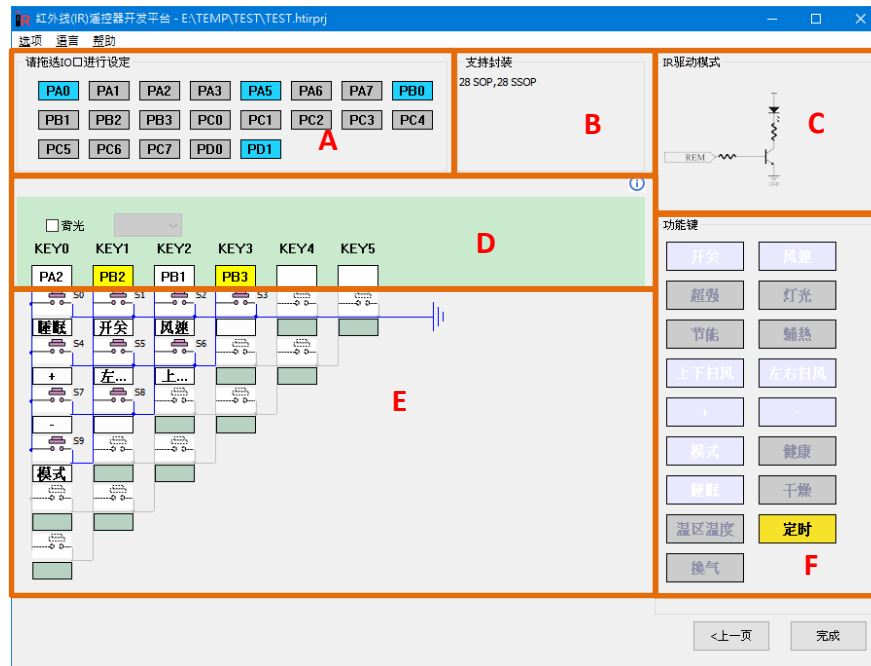


Fig.43 按键配置区

配置页面左上方 A 区为可使用的 I/O 引脚，其中蓝色引脚图标为 HT67F2432 24SOP/24SSOP 和 28SOP/28SSOP 封装都共有的引脚，绿色引脚图标为 28SOP/28SSOP 特有的引脚。通过拖选的方式将 I/O 图标拖入下方 D 区的按键位置即可完成对 I/O 的配置。按键页上方 B 区提示目前按键配置方式可使用的封装，如果在 LCD 配置页面将 PC0、PC1、PC2 和 PC3 这几个 I/O 配置为 LCD 驱动脚或在配置被拖入 D 区配置为按键脚，24SOP/24SSOP 封装则在 B 区被移除。

当 I/O 配置到按键配置表中，则阶梯矩阵键盘表随即自动生成。可生成的最大键数计算公式为 $K = 1+2+\dots+N$ ，N 为已配置 I/O 的个数，HT67F2432 的 LCD 型遥控器按键配置最多可用 6 个 I/O 配置 21 个按键。

同样，当鼠标点击任意一个可配置的按键表格，与此按键对应的两个 I/O 则会有黄色高亮提示。

点击第一行的按键表格仅有一个 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。

按键功能通过从 F 区把相应的功能方块拖动到左方的 E 区按键中即可完成按键最后的功能配置。

因 HT67F2432 仅支持 REM 方式驱动输出，所以 C 区只做驱动电路展示用，不可改变驱动方式。

三、开发平台硬件

红外线 (IR) 遥控器开发平台的配套解码硬件共有 3 部分，分别是开发板 ESK-IRRC-T00，ESK-IRRC-T01 和解码板 ESK-IRRC-R00。

3.1 ESK-IRRC-T00

ESK-IRRC-T00 是主控为 HT68F2420 20SSOP 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。1 个 REMDRV 引脚直驱 5mm IR LED。电源使用 CR2032 电池。电路如下图。

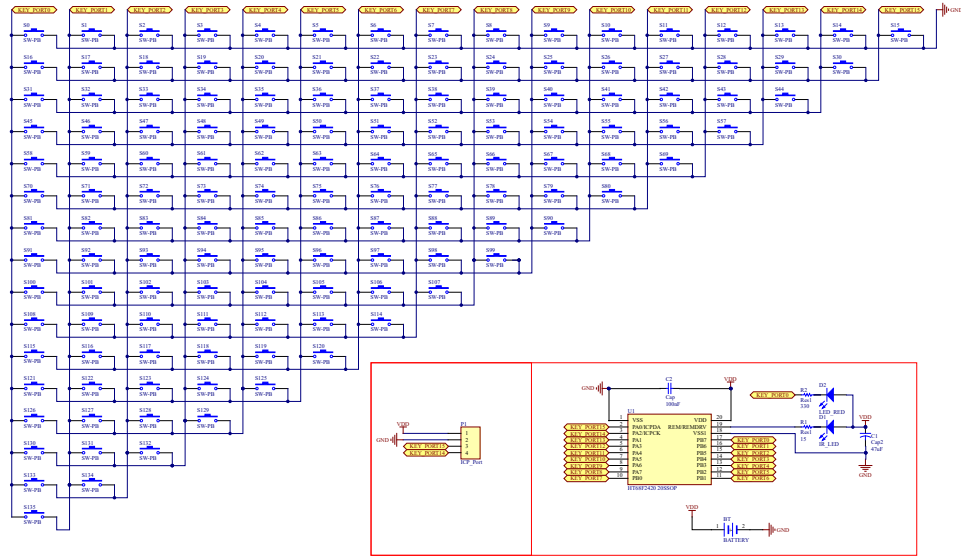


Fig.44 ESK-IRRC-T00 原理图

3.2 ESK-IRRC-T01

ESK-IRRC-T01 是主控为 HT67F2432 28SSOP 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。1 个 REM 引脚控制 MOS 驱动 5mm IR LED。电源使用 CR2032 电池。电路如下图。

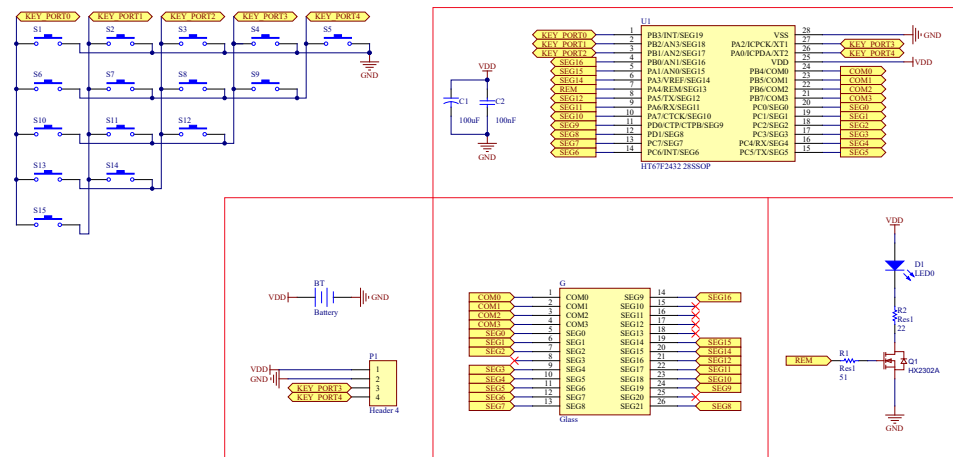


Fig.45 ESK-IRRC-T01 原理图

3.3 ESK-IRRC-R00

ESK-IRRC-R00 是主控为 HT66FB550 的解码板，以 16MHz 的速率抓取红外载波信号。抓取信号通过板上的 Micro USB 接口与电脑连线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。

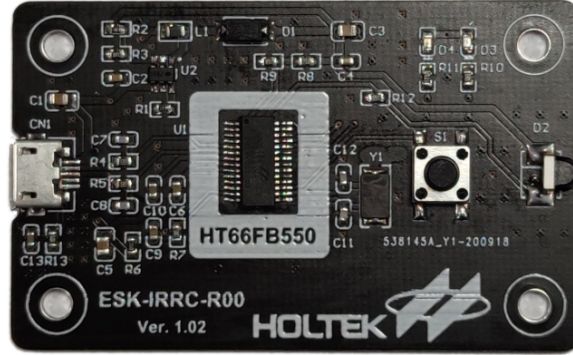


Fig.46 ESK-IRRC-R00 解码板硬件图

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。

解码板右方为接收感应探头，用于感应红外遥控信号。

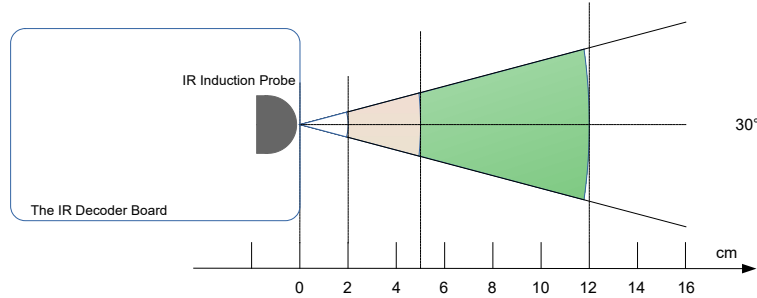


Fig.47 接收角度俯视图

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、解码与学码开发

4.1 码形校验

在一款标准协议或自定义协议遥控器开发完成之后，可以将生成的工程编译之后烧录到遥控器开发板中，用开发板向解码板发码，可对遥控器开发板的程序进行验证。

操作方式如下：

1. 将解码板连入电脑。
2. 由新建档案进入“学码 / 查看波形模式”，跳入 Fig.48 软件界面。
3. 按下软件的“开始解码”按键，此时解码板状态转入接收状态，红灯亮起，需在 6 秒内把待测的遥控器对准解码板的感应探头，并按下遥控器按键发射红外波形。
4. 当解码板绿灯亮起，则接收完成，解码板将发码数据上传到电脑，解码板绿灯快闪 4 次提示上传成功，同时软件解码完成后显示出发码参数。
5. 如果未收到发码或者数据异常，解码板红灯快闪 4 次提示上传失败。

一般型遥控器接收解码效果图如下：

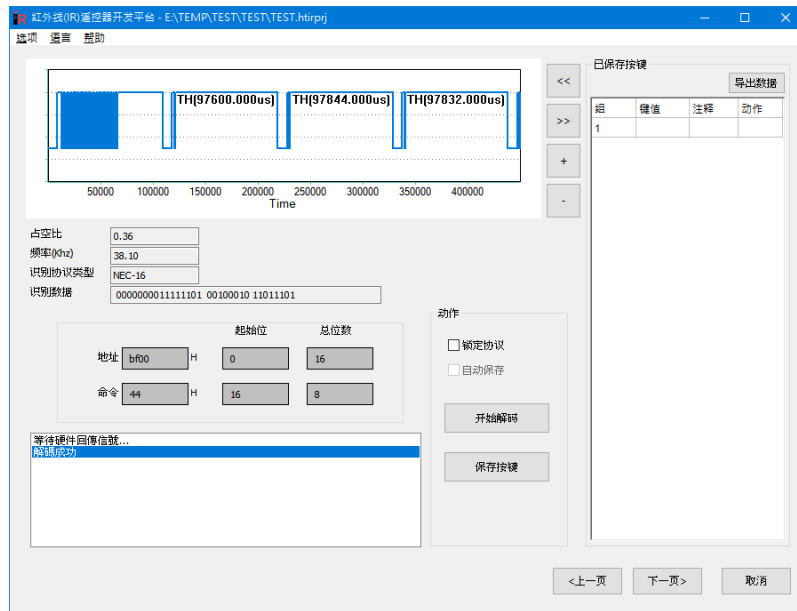


Fig.48 一般型遥控器接收校验

在 Fig.48 中，软件的上方还原出遥控器发码波形图，波形与 IR 接收器件 1838 输出方式类似，有载波的 MARK 用低电平显示，无载波的 SPACE 用高电平显示。波形图右方有放大缩小和左右移动的按钮。也可以用鼠标在波形图按住鼠标左键向右下方画出一个矩形，可将图形放大，按住鼠标左键向右上方画出一个矩形，可将图形缩小。按住鼠标右键左右拖动可以移动查看图形。

波形还原图下方显示的内容则是发码的具体参数，如载波的占空比、频率、识别出协议的类型和波形记录到的每个 bit。在这一串 bit 下方分别是识别到的地址和命令，以及地址和命令所在的起始位和总位数。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，需手动输入地址和命令所在的起始位和总位数。

LCD 型遥控器解码效果如下：

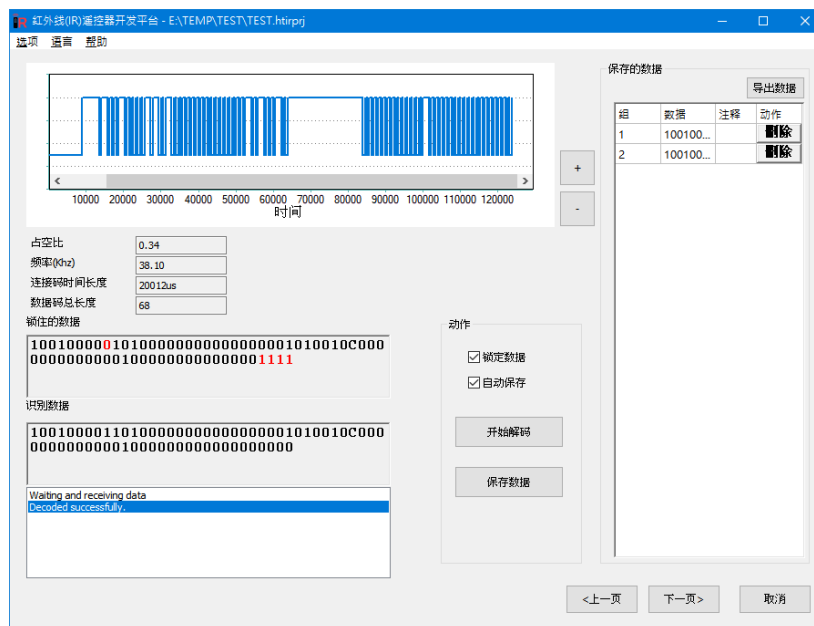


Fig.49 LCD 型遥控器接收校验

在 Fig.49 中，软件的上方还原出遥控器发码波形图，波形与 IR 接收器件 1838 输出方式类似，有载波的 MARK 用低电平显示，无载波的 SPACE 用高电平显示。波形图右方有放大缩小和左右移动的按钮。也可以用鼠标在波形图按住鼠标左键向右下方画出一个矩形，可将图形放大，按住鼠标左键向右上方画出一个矩形，可将图形缩小。按住鼠标右键左右拖动可以移动查看图形。

波形还原图下方显示的内容则是发码的具体参数，如载波的占空比、频率、连接码时间、数据码总长度。

识别数据区默认按 PDM 的调制方式识别，Bit 识别方式按 SPACE 时间比较长的为 1，较短的为 0。当锁定数据勾选后，则会将此串数据复制到锁定区，后面每次识别到的新的发码都会与锁定区进行比较，每个位与锁定区有区别时，则在锁定区的对应位的二进制数变红色显示。比如上图，新识别到的代码比锁定区加了温度 1 度，出现了两处变化，可由此确定温度所在的位还有温度数据加入了校验码计算。

识别数据区内容中，C 表示此处插入了连接码，L 表示两组码插入的间隔，E 表示发码结束。

4.2 学码开发

如果有一个已有的遥控器，因找不到原工程或源代码，但又需要开发出与此遥控器相同协议的方案，可以使用学码开发的方式，通过解码板读取此遥控器的协议、载波参数、地址和命令码等参数，并以此开发出协议和功能一致的遥控器产品。

4.2.1 一般型遥控器学码开发

在以上码形校验步骤，解析出了红外遥控器协议之后，点击保存按键，即可将识别到的命令码保存下来，用于新方案的按键使用。

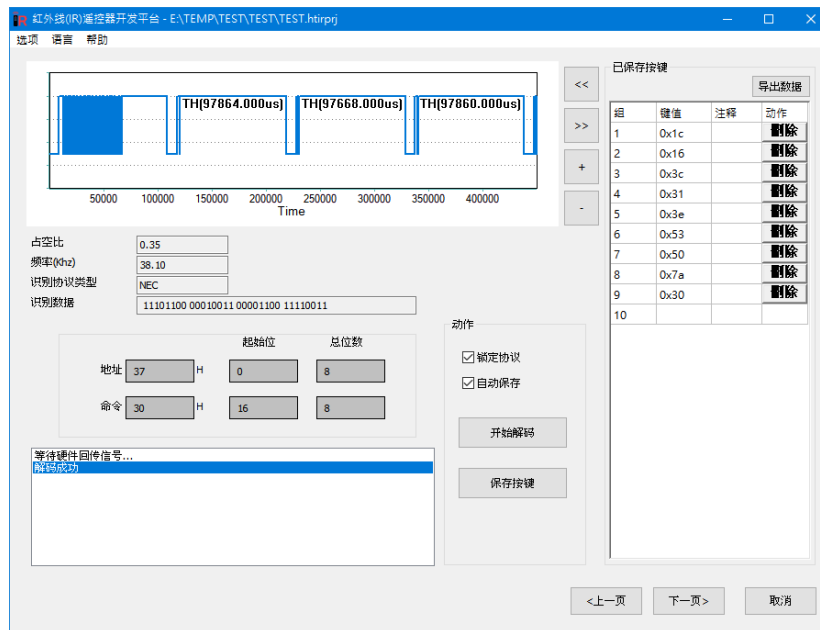


Fig.50 接收学码开发

对于非标准协议的学码，因为软件无法预判地址或命令的位数，需要开发者自行分析多组发码确定了地址和命令的起始位和总位数，手动输入配置栏中，由软件根据地址和命令码的定位识别出正确的发码。

如需对多个按键命令码进行保存，可以在确定了协议之后勾选锁定协议和自动保存，可以简化多键学习的操作。

在对多个按键的命令码成功保存之后，点击下一步，可以跳转到相应的标准协议或自定义协议中再做部分的参数修改。

1. 识别为标准协议，下一页会跳转到标准协议参数配置区，此时可更改的部分只有地址和 Duty。软件会预写入学习到的遥控器的地址码。

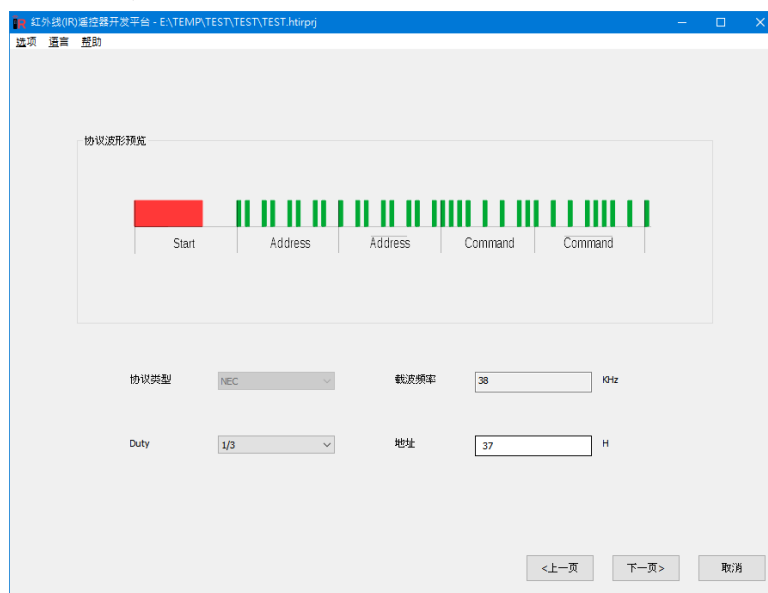


Fig.51 学码标准协议跳转

2. 识别为非标准协议，软件会把自动记录学的相关码型参数导入自定义协议的设计参数中。因为很多遥控器本身在发码受软件指令的影响，所以测量到的实际参数跟设计参数会有误差。因此，在做非标准协议转自定义参数的开发时，建议对软件导入的参数另做适当调整。

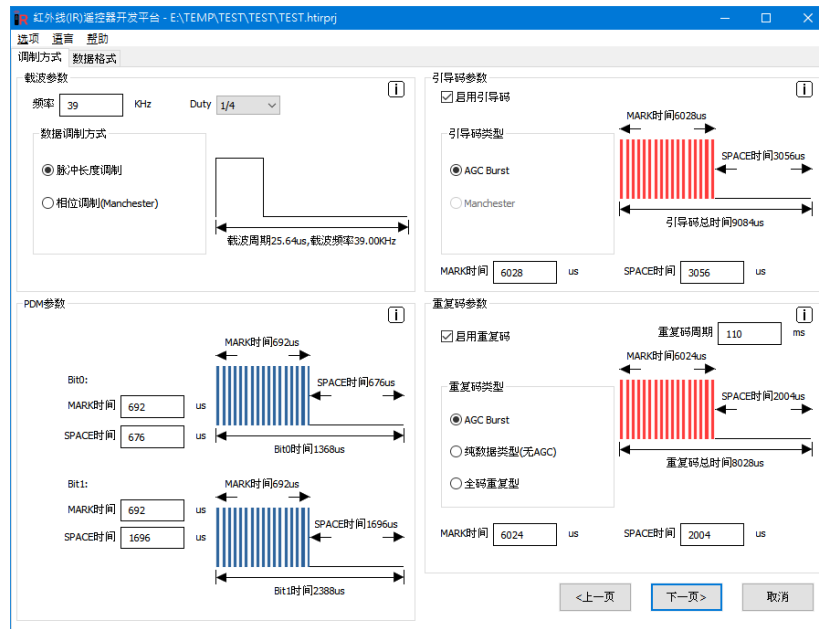


Fig.52 学码非标准协议跳转

4.2.2 LCD 型遥控器学码开发

识别操作按码形校验步骤，在解析出了红外遥控器发码参数和数据后，点击保存按键，即可将识别到的数据保存下来。

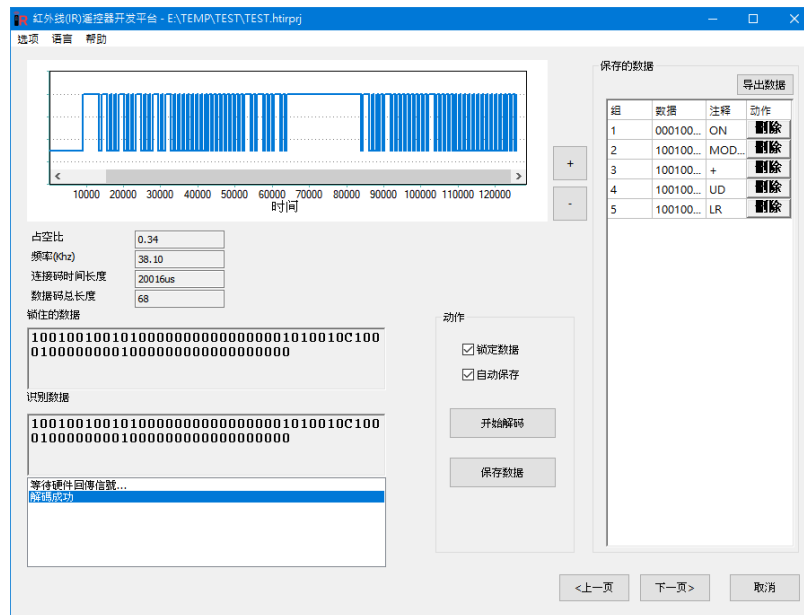


Fig.53 LCD 型遥控器解码

软件右上角有导出数据按键，可以将 LCD 型遥控器的发码参数和保存的数据码以 CSV 表格的方式保存出来。

档案创建时间:		"2021-04-27 16:13:13							
解码信号频率:		'38.10							
占空比:		'0.34							
数据码长度(不含引导码):		'68							
载波:		'							
	MARK	'9072							
	SPACE	'4476							
Bit0		'							
	MARK	'612							
	SPACE	'600							
Bit1		'							
	MARK	'612							
	SPACE	'1568							
连接码:		'1							
	连接码时间:	'20ms							
识别数据(先接收在前):		'							
	第1组:	'00010000100100000000000000001010010C0000000000001000000000000001011							ON
	第2组:	'10010000100100000000000000001010010C00000000000100000000000000111							MODE1
	第3组:	'10010000101000000000000000001010010C000000000001000000000000001111							+
	第4组:	'10010010010100000000000000001010010C1000000000001000000000000001111							UD
	第5组:	'10010010010100000000000000001010010C1000100000001000000000000000							LR

Fig.54 LCD 型遥控器解码导出参数与解码内容

因 LCD 型遥控器的发码数据内容比较复杂且难以预测，所以学码软件不对数据内容做分析。需要由开发者对比各个按键功能在发码上的差别。在学码完成之后点击下一页软件会弹出以下提示，LCD 型遥控器的学码只导入发码参数，不产生配置。

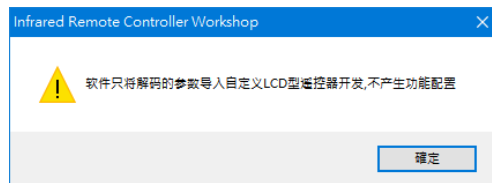


Fig.55 LCD 型遥控器解码提示

五、其他功能说明

5.1 F/W 功耗控制

红外线 (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尽量做到较佳的优化表现。

5.1.1 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。此平台软件产出的遥控器程序，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。

5.1.2 静态功耗控制

F/W 产出的遥控器程序，一般型遥控器默认 Halt 模式每 0.128s 一次 WDT 溢出唤醒扫描按键。LCD 型遥控器每 64ms 中断唤醒检查定时，每 0.128 秒扫描一次按键，每 0.512 秒可执行 LCD 闪烁，背光计时，电池电压检测（仅开放于二次开发）。一般型遥控器最多扫描 136 个按键，待机功耗可控制在 7 μ A 以内，LCD 型遥控器在 LCD 常亮显示可控制在 10 μ A 以内，LCD 不亮状态可控制在 5 μ A 以内。

5.2 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。

六、附录 & FAQ

6.1 LCD 型遥控器发码控制与数据表

功能	独立按键	数据	按键在开机 / 关机可动作	关联按键	功能特别说明
开关	√	√	√	开关	清除所有定时状态
模式	√	√	×	模式	切换模式自动清除超 强、睡眠、节能状态
风速	√	√	×	风速	默认 0 档为自动档
温度数据	×	√	×	[+], [-]	
上下扫风	√	√	×	上下扫风	
左右扫风	√	√	×	左右扫风	
扫风确认	×	√	×	上下扫风, 左右扫风	
超强	√	√	×	超强	
节能	√	√	×	节能	
健康	√	√	×	健康	
睡眠	√	√	×	睡眠	
换气	√	√	×	换气	
干燥	√	√	×	干燥	
辅热	√	√	×	辅热	
灯光	√	√	×	灯光	
温区温度	√	√	×	温区温度	
校验码	×	√	×	*	
定时	√	√	√	定时, [+], [-]	
定时开	×	√	×	定时	
定时关	×	√	×	定时	
开机计时	×	√	×	定时	
关机计时	×	√	×	定时	
固定码 1~10	×	√	×	N/A	

* 校验码可根据计算公式使用任意按键的发码值。

6.2 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，但是这种发码使用平台的解码板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 非 MARK 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开 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。

6.3 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。

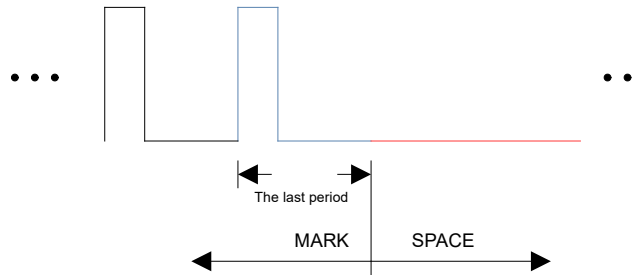


Fig.56 最后周期处理

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 Fig.56，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© 2021 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/>。