



使用手册

IR 遥控器开发平台使用手册

目录

1. 简介	3
2. 开发平台软件	3
2.1 一般型遥控器开发	5
2.2 自定义协议开发	6
2.3 学码 / 解码挂件	10
2.4 按键与驱动	11
3. 开发平台硬件	14
3.1 遥控器开发板	14
3.2 解码板	15
4. 其他功能说明	16
4.1 F/W 功耗控制	16
4.2 解码识别	16
5. 附录 & FAQ	16
5.1 特殊调制发码说明及应用	16
5.2 MARK 与 SPACE 设计误差说明	17

1. 简介

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 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。

常用一般型红外线遥控器主要应用功能分为以下几类：

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. 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。

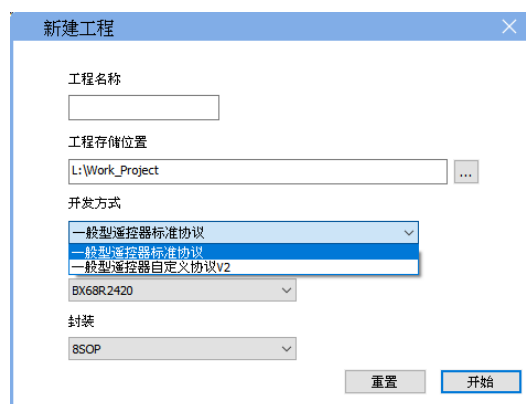


工程管理

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) 按键及驱动配置
- c) 产出 Project (可再做二次开发)，转到 BX-IDE 编译产出烧录档

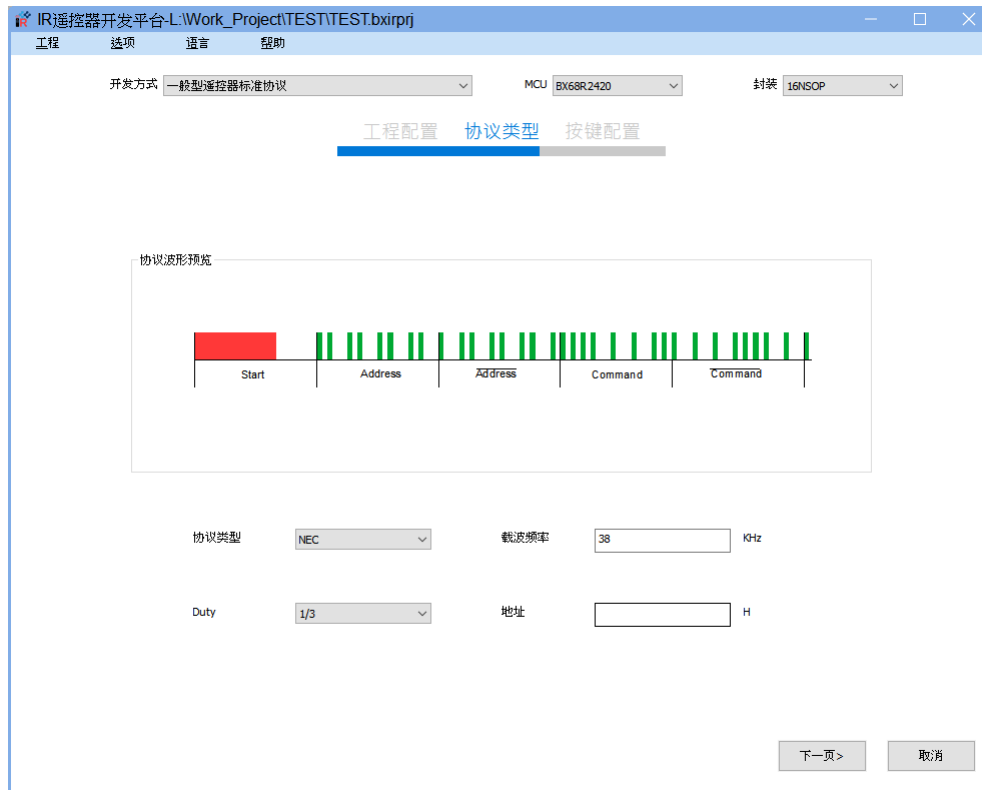


工程建立

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



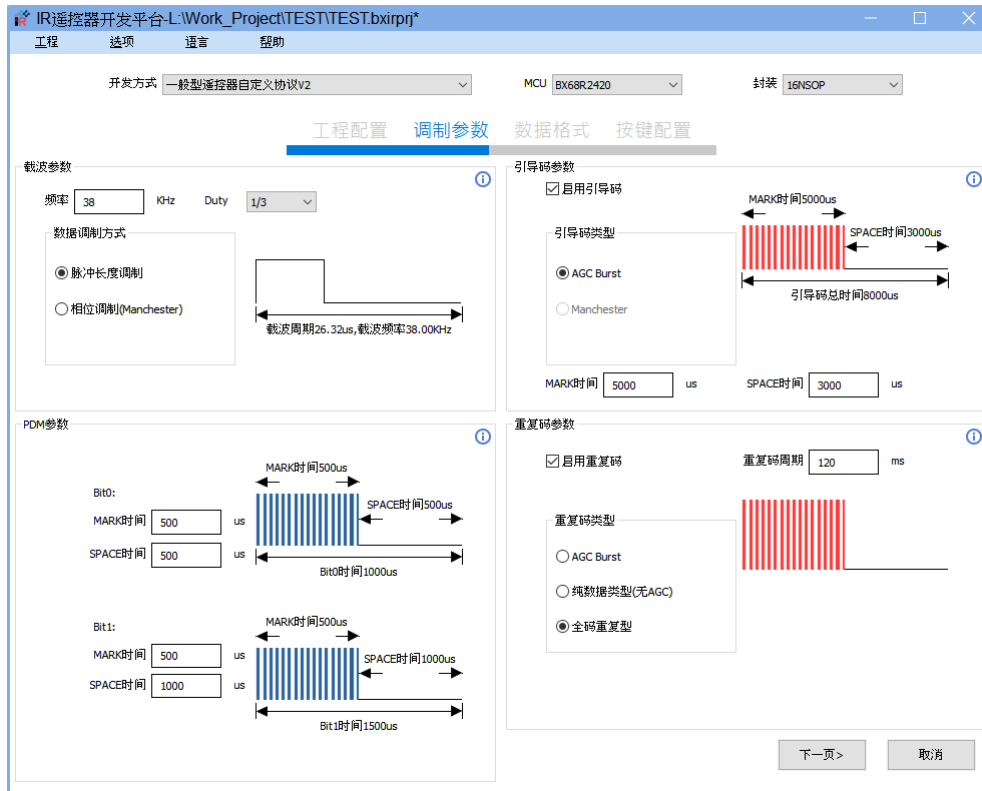
标准协议配置界面

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。在选择了使用的协议之后，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。

2.2 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。

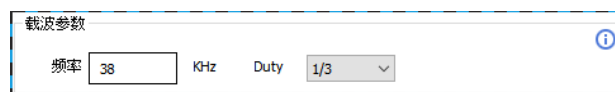
第一个步是设定调制参数。在调制参数页可以自定义载波参数，数字 1 和 0 的调制参数，引导码参数和重复码参数。



自定义协议配置界面

具体的配置操作方式如下：

载波

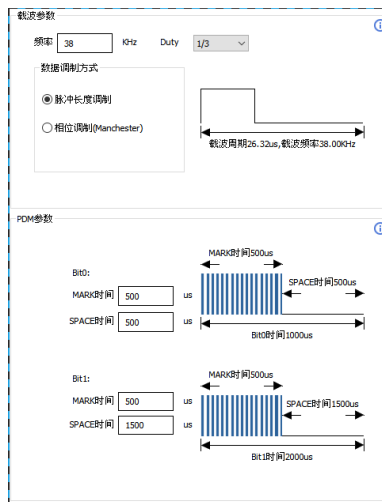


载波参数

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。

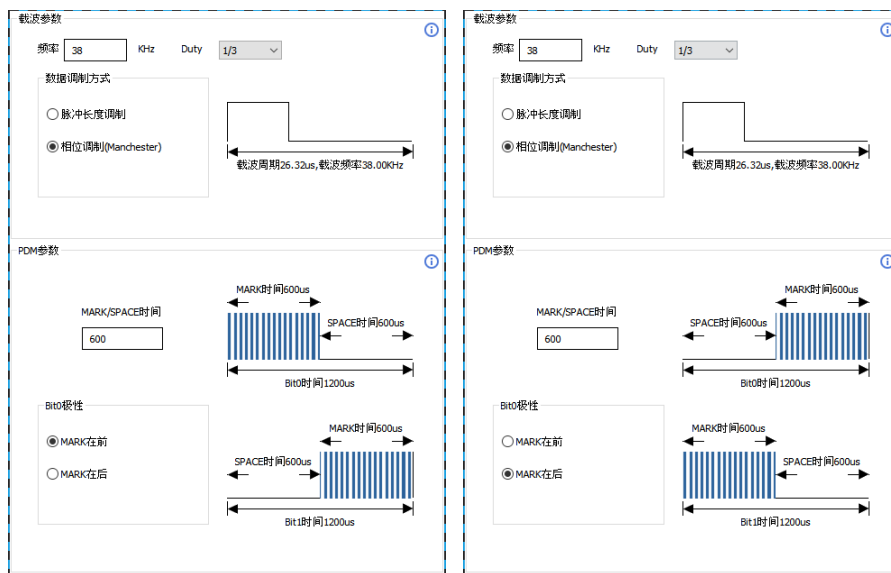
数据调制方式

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

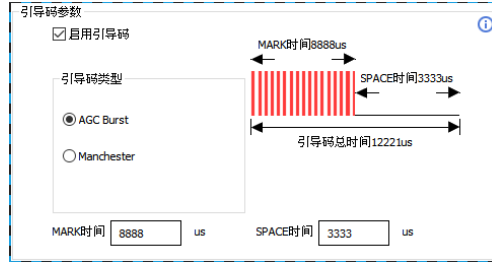
上图是相位调制参数的配置图。在软件上可以配置出 Bit0 和 Bit1 的 MARK/SPACE 时间, 相位调制只需要定义一个时间参数, 再通过 Bit0 极性定义的方式选择为 MARK 在前或是 MARK 在后来设定 Bit 极性。参数右方以图形的方式分别标示 MARK、SPACE 和总时间几项参数。

Bit 的 MARK 和 SPACE 的参数范围可配置在 $300\mu\text{s}$ ~ $2000\mu\text{s}$ 之间。

引导码 / 头码参数

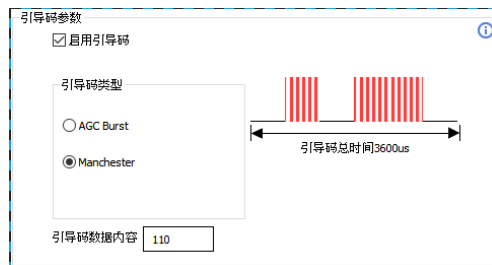
引导码也称为头码，主要用于遥控器向接收端发出控制信号前先发送一个初始化和校准的信号。

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。



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

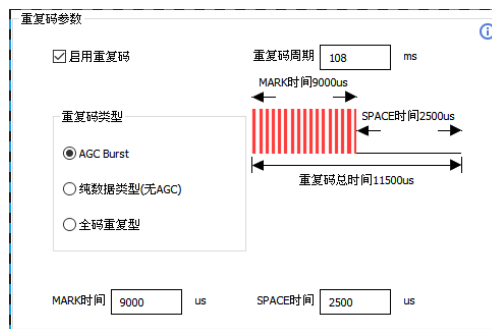
如上图，当数据使用 Manchester 调制方式时，则可使用类似 Philips RC-5 协议的数字形式引导码，引导码在软件中最多可配置 3 位，软件以图形的方式分别标示 MARK、SPACE 的图形位置，还有总时间参数。

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。

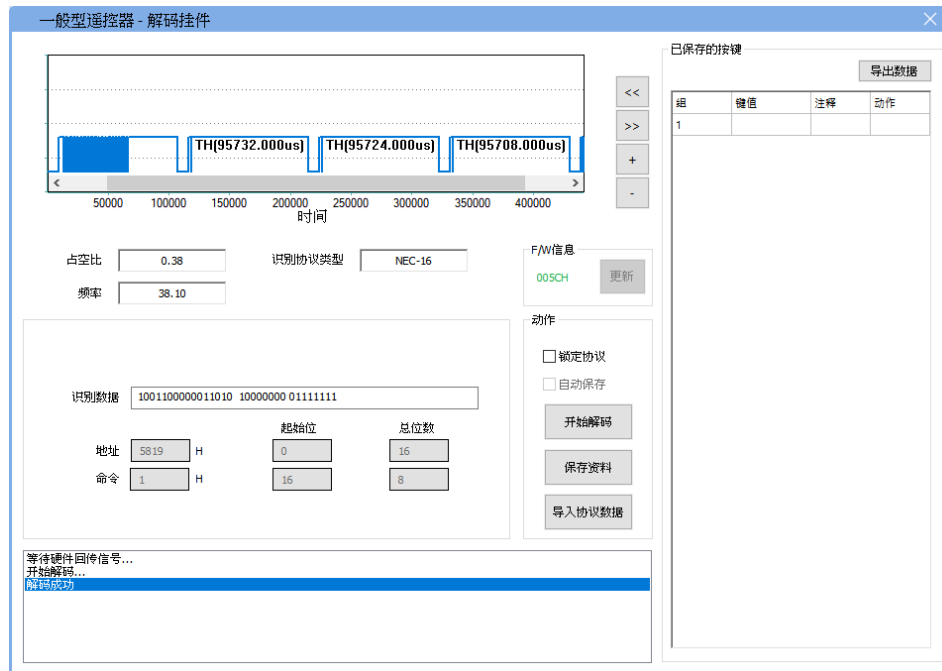
Toggle 位: 用于插入一个变化位，区分全码重复的遥控器所发出的两组码是保持按住的重复发码还是按键连续两次被按下发出的两组码。

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。

2.3 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 BX-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。

2.4 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。

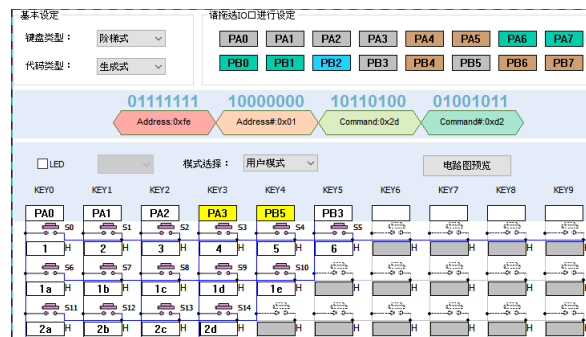


配置按键页

2.4.1 按键与 I/O

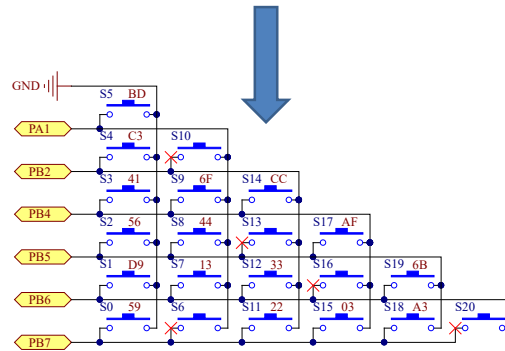
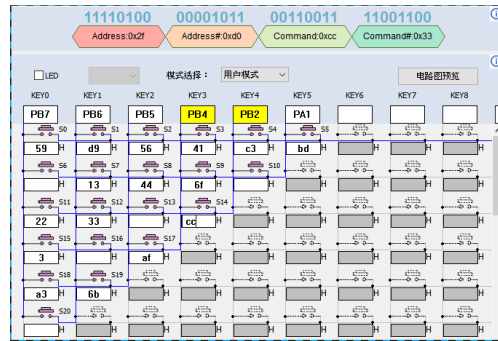
配置页面左上方 A 区可选择 MCU 的封装类型，B 区的基本设定可以选择不同的键盘类型和工程产出代码的方式，待选 I/O 区蓝色引脚图标为所有封装都共有的引脚，黄色引脚图标为 16NSOP 和 20SSOP/20NSOP 可使用的引脚，绿色引脚图标为 20SSOP/20NSOP 可使用的引脚。灰色代表已被使用或者当前选择的封装不包含的引脚。用拖选的方式将 I/O 图标拖入下方 E 区的按键引脚位置。

2.4.2 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。



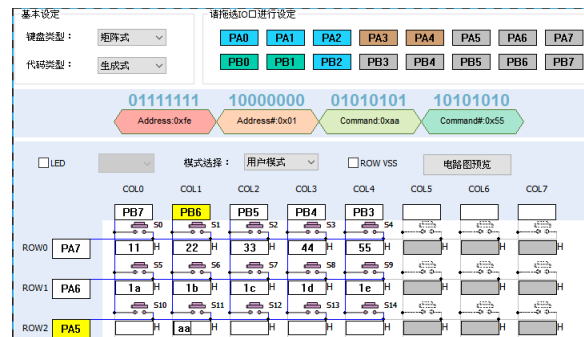
阶梯键盘按键配置与原理图关系

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。

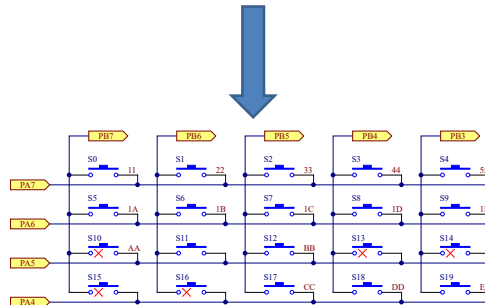
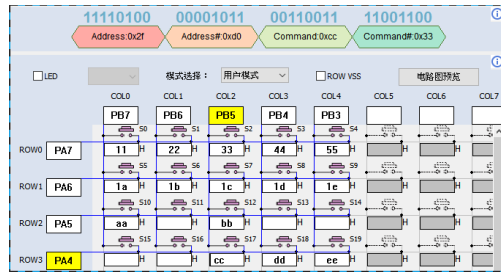
上图是阶梯按键表对应为下方的阶梯键盘原理图。未配置键值的按键在原理图上以打叉表示。图中 S6、S10、S13、S16、S20 即使未配置键值或者在实体电路未焊按键，但在软件生成配置时也会列入扫描的码表中。未配置按键在软件生成的键值表默认填充 00H。

2.4.3 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

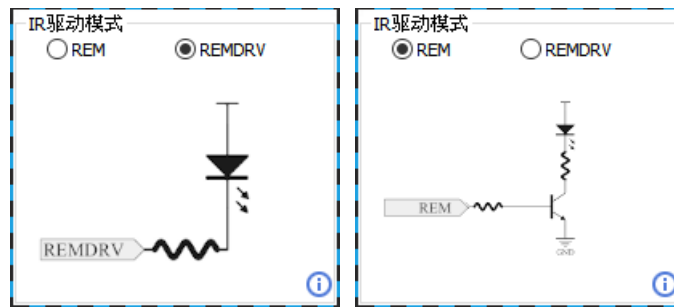
I/O 在软件中配置到按键区之后，可在对应的按键码表中输入命令码，鼠标点击任意一个可配置的按键码表格，与此命令码表格对应的两个 I/O 则会有黄色高亮提示，如上图上半部分按钮 S17 的连线连接到 PB5 和 PA4 两个 I/O。在对应的按键码表中输入命令码之后，发码预览区同样提示二进制码。

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。

2.4.4 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”。

3. 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
BX-IRRC-TMT00	BX68R2420	带 IR 发射模块化开发板主板
BX-IRRC-TK00	—	24 键矩阵式按键模块化开发板
BX-IRRC-TK01	—	136 键阶梯式按键模块化开发板

3.1 遥控器开发板

IR 遥控器开发平台遥控器的开发板为模块式开发板。模块式开发板对 MCU 的主控，发射，按键，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 模块式开发板

模块式的开发板由母板，搭配发射和按键这几个功能组合成一套完整的遥控器方案。

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。

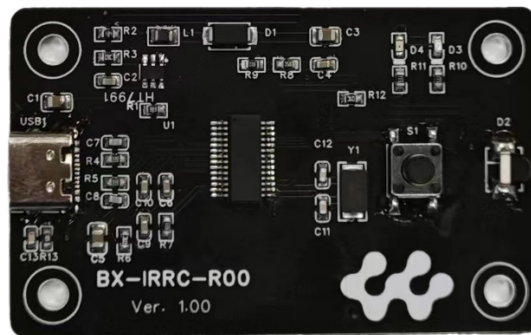
一般型遥控器不含显示，只需要把主控板和按键板组合起来之后即可做一般型遥控器的开发。



模块式一般型遥控器开发板

3.2 解码板

解码板 BX-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。BX-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。

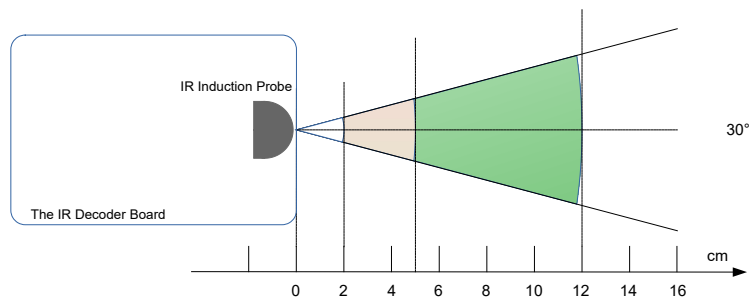


BX-IRRC-R00 解码板硬件图

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。

4. 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。

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。

5. 附录 & FAQ

5.1 特殊调制发码说明及应用

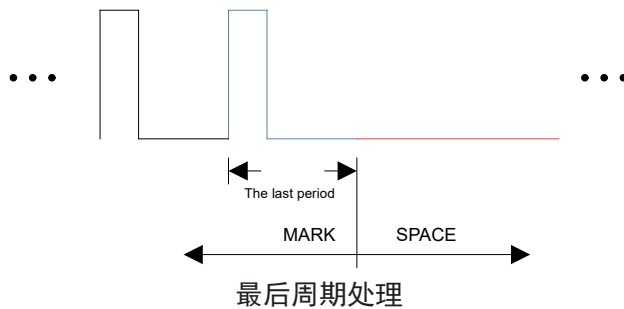
在红外遥控器的调制中，有部分红外发码本身符合特定的协议，且可用对应协议指定的解码程序进行解码，但是使用平台的解码板，对这种未能预知协议的解码设备做解码识别仍会出现一些识别问题。具体情况为以下几种。

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 协议应避开使用高 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。

调制方式	特殊情况	解码异常说明	解决方式
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。

5.2 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。

本文件出版时芯群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。芯群不承担任何明示、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。芯群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，芯群并不推荐将芯群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。芯群特此声明，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芯群产品的风险完全由买方承担，如因该等使用导致芯群遭受损害、索赔、诉讼或产生费用，买方同意出面进行辩护、赔偿并使芯群免受损害。芯群 (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。芯群在此并未明示或暗示授予任何知识产权。芯群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。