14 遥控器控制小车行走

**本章概述**

RT-Thread的软件包提供了丰富的应用软件，在RT-Thread Studio 开发环境中，RT-Thread的软件包可以很方便的添加到项目中，大大的节省了产品的开发周期。

本章首先介绍红外接收的原理，然后通过使用RT-Thread的“infrared”软件包所提供的接口，进行远程红外信息的接收处理，接收遥控器发出的按健信息，并把信息显示到终端。通过本章的学习，读者可以了红外接收的原理，还可以进一步了解RT-Thread软件包的添加和使用流程，掌握RT-Thread软件包的使用方法。

**知识目标**

* 了解红外接收的通信原理；
* 掌握RT-Thread应用软件包的添加方法；
* 掌握RT-Thread应用软件包使用方法；

**技能目标**

* 能够根据项目需要查找RT-Thread相关应用软件；
* 能够把RT-Thread相关应用软件添加到项目中；
* 能够阅读RT-Thread相关应用软件的使用说明并根据说明进行编程；

**14.1 红外接收原理**

现在的家用电器中，很多都配置有红外接收器，可以接收红外遥控器发出的控制信号。这一节，我们先了解一下红外接收的工作原理。

**14.1.1 红外通信系统**

红外通信一般分为两部分，分别是发射部分和接收部分，发射部分的主要元件为红外发光二极管，而接收部分主要采用红外接收头。发送端对数据进行编码，然后调制成一系列的脉冲信号，最后通过带有红外发射管的发射电路发送脉冲信号，即我们常说的红外信号。接收端完成对脉冲信号的接收、放大、检波、整形，然后解调出编码信号，最后对其解码获取到发送的数据。具体的红外通信系统如图14-1所示。

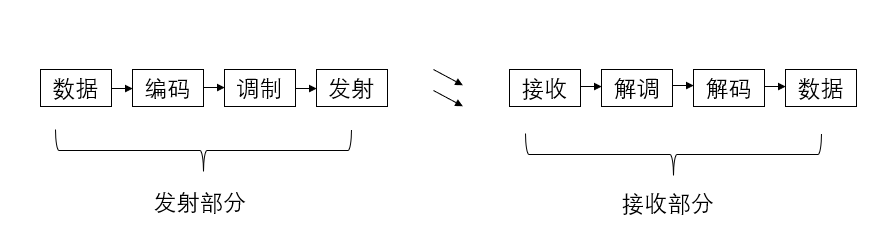


图14-1 红外通信系统

**14.1.2 认识红外接收头**

通常采用红外接收头接收红外信号，一般的红外接收头有三个引脚，分别是：电源正极VCC、电源负极GND、信号输出端OUT，如图14-2所示。

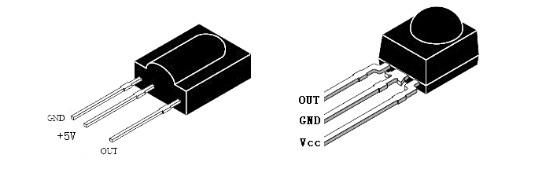


图14-2 红外接收头

封装一般有铁壳屏蔽封装和环氧树脂塑封封装，如图14-3所示，但是形状各种各样，有贴片型、插件型，鼻梁型、草帽型、圆柱形、半球型等，不同厂家的接收头引脚顺序以及外壳形状各异。

（a）铁壳屏蔽封装 （b）环氧树脂塑封封装

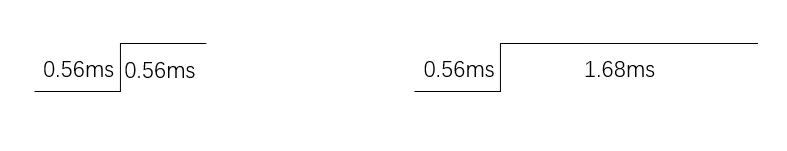
图14-3 不同封装的红外接收头

**14.1.3 红外遥控编码协议**

红外遥控编码协议目前应用比较广泛的有基于PWM（Pulse Width Modulation 脉宽调制）的NEC Protocol和基于PPM（Pulse Position Modulation 脉冲位置调制）的Philips RC-5 Protocol。这里我们主要介绍一下NEC编码原理。

（1）逻辑位规定

逻辑0：NEC规定用0.56ms低电平后面跟着0.56ms高电平来表示逻辑0，如图14-4（a）所示。



（a）逻辑0 （b）逻辑1

图14-4 逻辑位规定

逻辑1：NEC规定用0.56ms低电平后面跟着1.68ms高电平来表示逻辑1，如图14-4（b）所示。

（2）载波规定

NEC协议规定载波信号为频率为38KHz的方波信号。在发送端，发送38K载波信号来表示发送高电平，不发送载波信号表示发送低电平；而在接收端，接收到载波信号输出低电平，没接收到载波信号输出高电平。从这个定义可以看到，接收端和发送端逻辑正好相反。当发送端发送0.56ms低电平+0.56ms高电平时，接收到输出0.56ms高电平+0.56ms低电平，如图14-5所示。

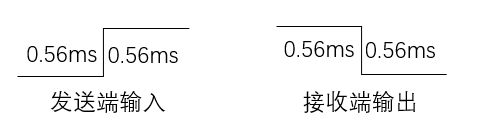


图14-5 发送端与接收端逻辑相反

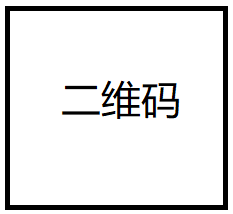
（3）遥控指令的数据格式

NEC 遥控指令的数据格式为：同步码头、地址码、地址反码、控制码、控制反码。同步码由一个 9ms 的低电平和一个 4.5ms 的高电平组成，地址码、地址反码、控制码、控制反码均是 8 位数据格式。按照低位在前，高位在后的顺序发送。采用反码是为了增加传输的可靠性（可用于校验）。

**14.2 任务14-1 识别红外遥控器按键信号**

任务描述：本任务通过检测红外接收头HS0038的输出信号，识别遥控器发出的信号，把识别结果打印到控制台。

RT-Thread提供了丰富的应用软件包，在实际项目开发中，可以方便地加载到项目中，从而大大缩短了产品的开发周期。本任务我们使用RT-Thread的应用软件包“infrared”进行设计开发。

扫描二维码观看任务实战演示

**14.2.1 硬件设计**

硬件设计如图14-6，采用HS0038作为远程红外接收头，其数据引脚接到STM32芯片的PA7引脚。

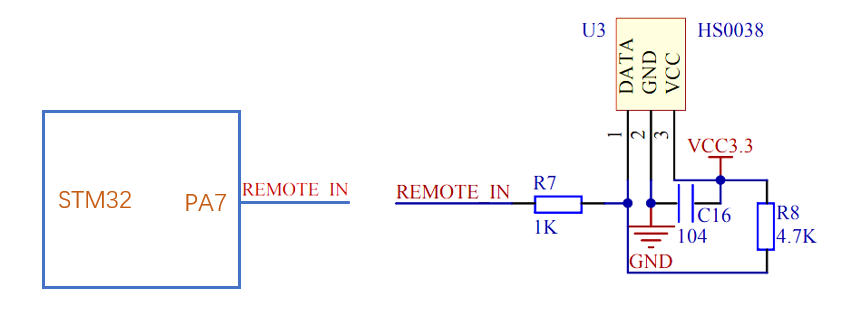


图14-6 红外接收硬件设计

**14.2.2创建项目及配置**

创建工程名称为“car\_Infrared”的RT-Thread项目，进行如下配置。

**（1）添加RT-Thread应用软件包。**

本项目我们使用RT-Thread自带的应用软件包“infrared”进行开发，该软件包实现了远程红外信号的接收和发送功能，这里我们使用其接收功能。设置方式如下。

1）在“RT-Thread Settings”配置界面点击“立即添加”添加软件包，如图14-7。

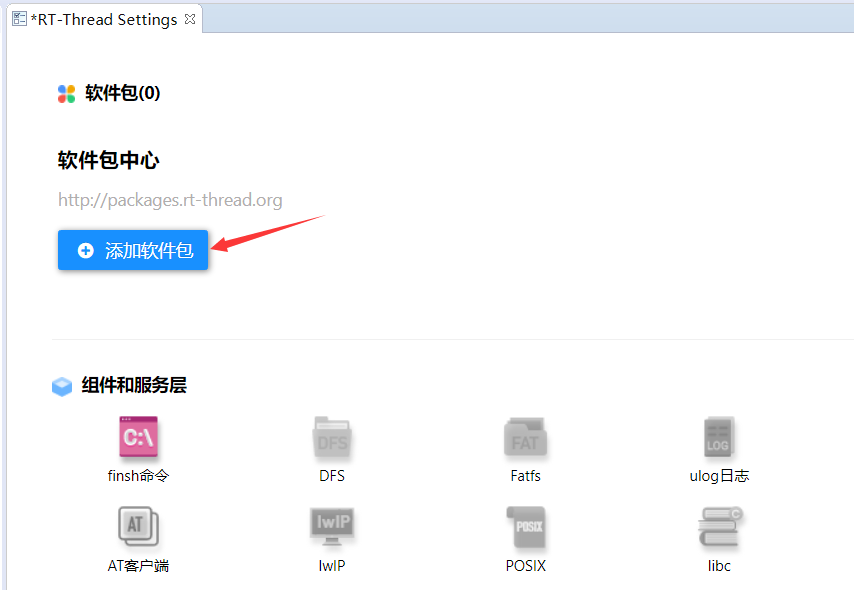


图14-7 添加软件包

2）如图14-8，在其中搜索“infrared”软件包。



图14-8 搜索软件包

3）在搜索结果中点击“添加”，把软件包添加到项目中，如图14-9所示。



图14-9 把软件包添加到项目

4）配置软件包参数。如图14-10，点击 “infrared”软件包的“配置项”，打开软件包配置界面，如图14-11配置软件包参数。

其中，主要配置两个地方，一个是红外接收引脚号，一个是用于计时的硬件定时器。本项目接收引脚编号设置为7（PA7引脚的编号为7），使用的硬件定时器名称为timer2。

引脚号可以通过drivers/drv\_gpio.c 的pins[]数组进行查看，如图14-12所示，PA7引脚对应的编号为7。

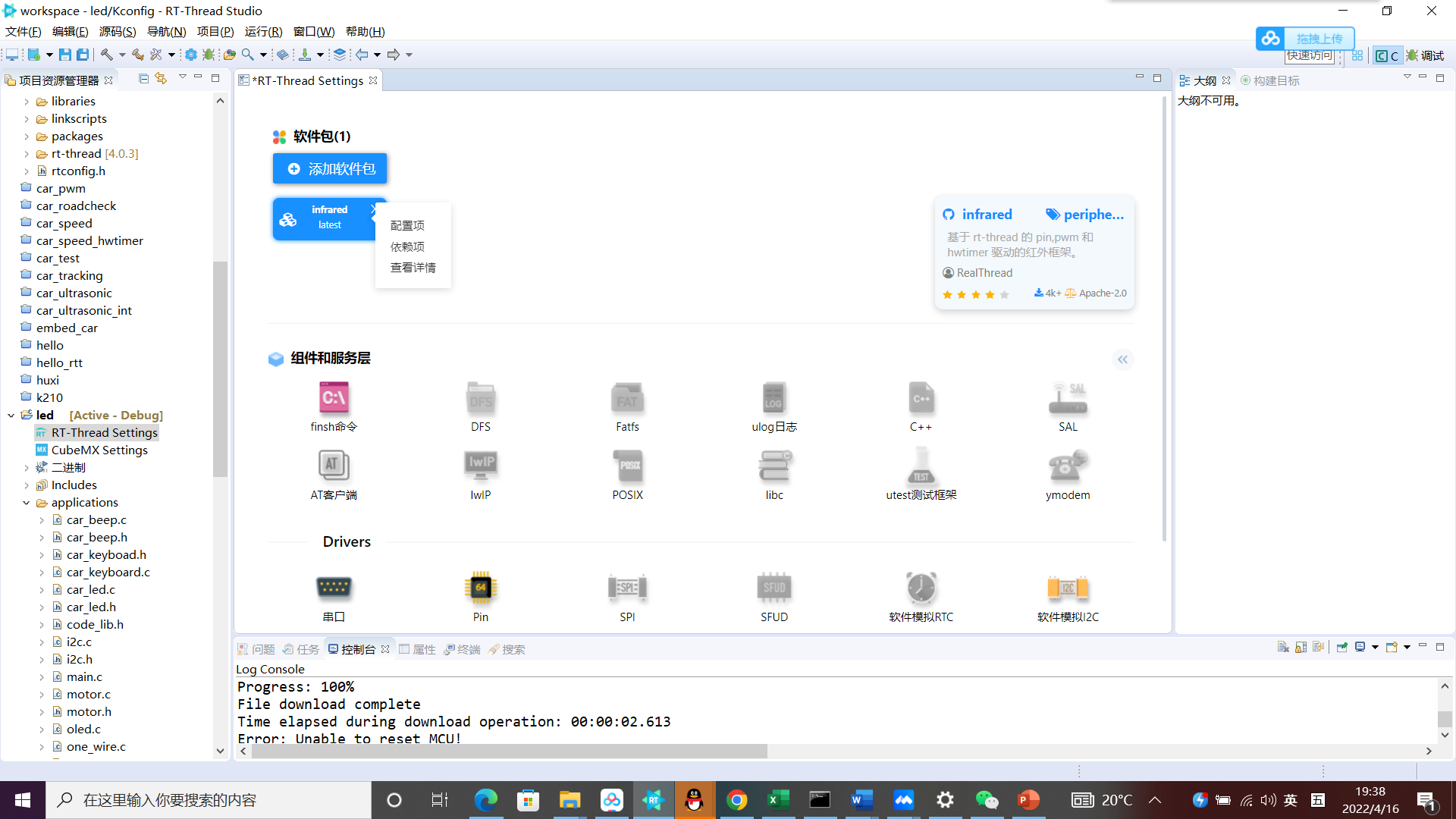


图14-10 打开软件包配置界面



图14-11 配置软件包参数

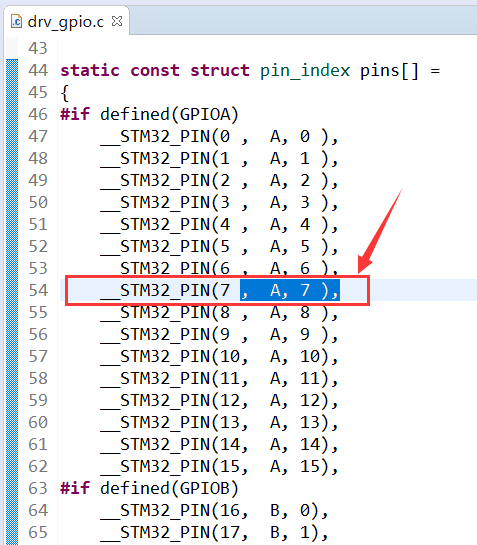
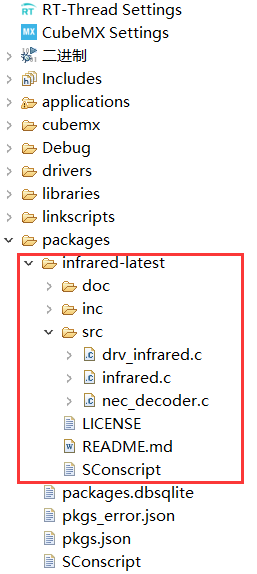
 

图14-12 查看引脚编号 图14-13项目中新增加的软件包代码目录树

5）保存配置。按ctrl+s保存后，可以看到项目树中新增加了一个软件包，如图14-13所示。其中文件drv\_infrared.c实现底层驱动，文件infrared.c实现中间层，文件nec\_decoder.c实现nec协议的解析。另外，软件包的用法可以参考软件包文件夹的README.md文件说明。

**（2）配置RT-Thread组件**

“infrared”应用软件中使用了硬件定时器，所以我们还需要在组件中开启“使用HWTIMER设备驱动程序”，如图14-14所示。

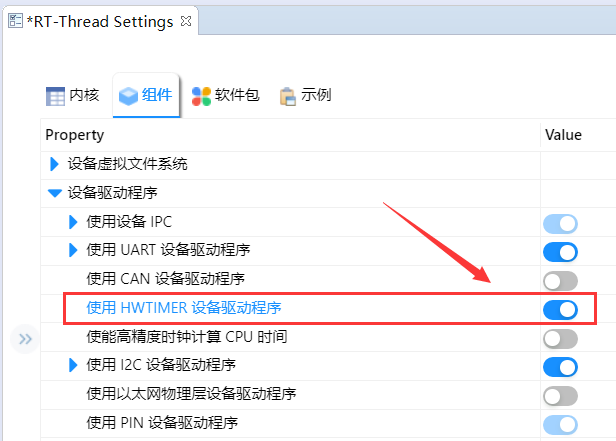


图14-14 使用HWTIMER设备驱动程序

**（3）使用CubMX配置STM32硬件外设**

本任务除了常规配置，如使能外部高速时钟、时钟树、使能串口1外，还必须配置“infrared”应用软件中使用到的硬件定时器（TIM2）,硬件定时器配置如图14-15所示。

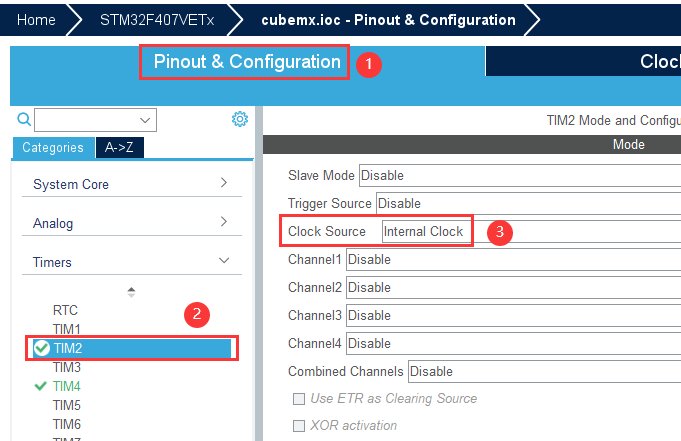
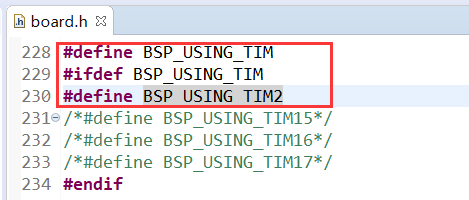
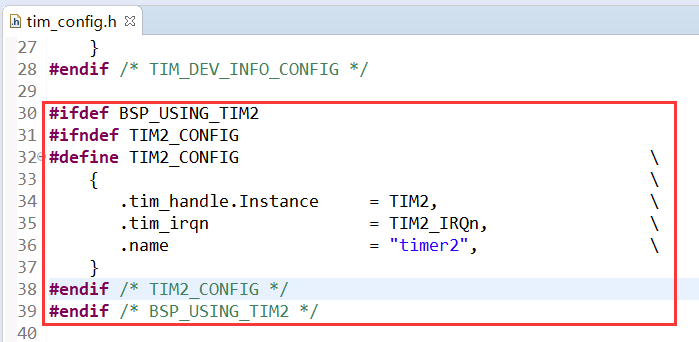


图14-15 硬件定时器配置

**（4） 打开硬件定时器宏定义**

如图14-16（a），在drivers/board.h文件中，打开定时器2的宏定义；如图14-16（b），在drivers/include/config/tim\_config.h文件中，定义定时器2的硬件参数。

（a） （b）

图14-16 TIM2宏定义

**14.2.3 程序设计**

新建源文件infrare\_sample.c，在infrare\_sample.c文件中进行程序设计。

infrare\_sample.c代码清单：

#include <infrared.h> //包含软件包的头文件

/\* 定义变量，用于保存读取到的数据 \*/

struct infrared\_decoder\_data infrared\_data;

/\* 从红外接收头读取数据 \*/

int infrared\_test(void)

{

ir\_select\_decoder("nec"); //设置使用NEC协议进行解调

while (1)

{

/\* 读取数据 \*/

if (infrared\_read("nec", &infrared\_data) == RT\_EOK)

{

if (infrared\_data.data.nec.repeat)

{

/\* 如果按键被长按，可以识别为重复按键 \*/

rt\_kprintf("repeat%d\n", infrared\_data.data.nec.repeat);

}

else

{

/\* 打印通信地址和按键的键值 \*/

rt\_kprintf("APP addr:0x%02X key:0x%02X\n", infrared\_data.data.nec.addr, infrared\_data.data.nec.key);

}

}

rt\_thread\_mdelay(10);

}

return RT\_EOK;

}

/\* 导出到 msh 命令列表中 \*/

MSH\_CMD\_EXPORT(infrared\_test, infrared receive test);

**14.2.4 测试**

把程序下载到开发板，启动系统后，首先在终端输入“infrared\_test”命令，然后随意按红外遥控器的按键，观察终端的输出如图14-17所示，说明程序已经接收到红外遥控器发出的信号并成功识别了按键的键值。



图14-17 红外遥控接收测试结果