

# **BC7215 及 C 语言驱动库**

## **应用示例**

# 目录

学习型遥控器.....	3
示例程序功能描述.....	4
使用.....	5
2 路红外遥控电源插座.....	6
示例程序功能描述.....	7
使用.....	7
3. 红外数据传输.....	8
示例程序功能描述.....	9
使用.....	9

BC7215 C 语言驱动库例程，提供了三种应用的例子分别为：

- 学习型遥控器

一个具有 4 个按键的可分别学习不同品牌遥控器的学习型遥控器

- 2 路红外遥控电源插座

可以和任何遥控器“配对”，利用家中遥控器上的空闲按键遥控控制电源开关。

- 红外数据传输

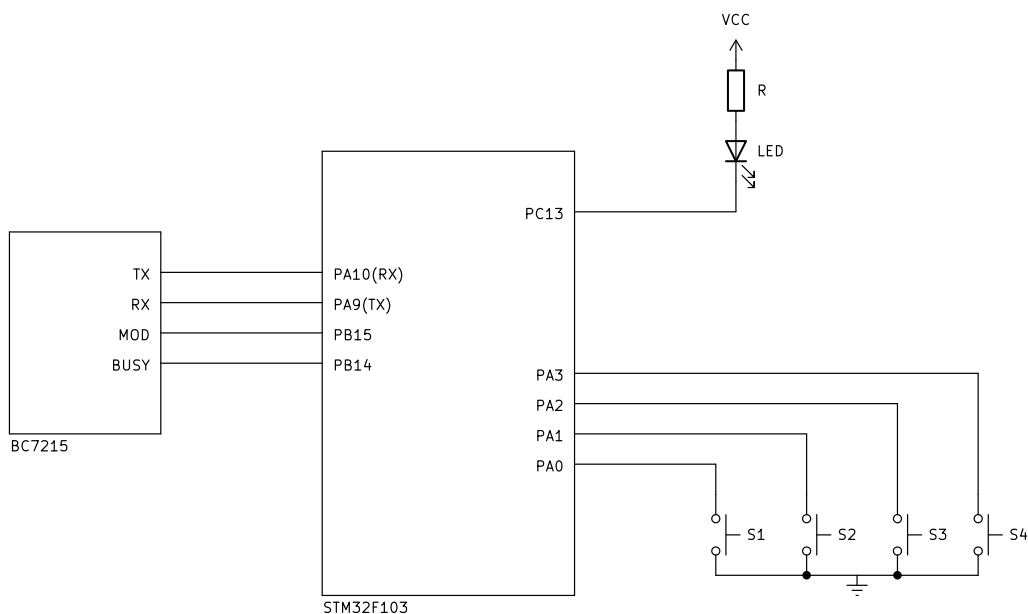
利用 BC7215 作为红外收发器件，用作数据传输。

## 学习型遥控器

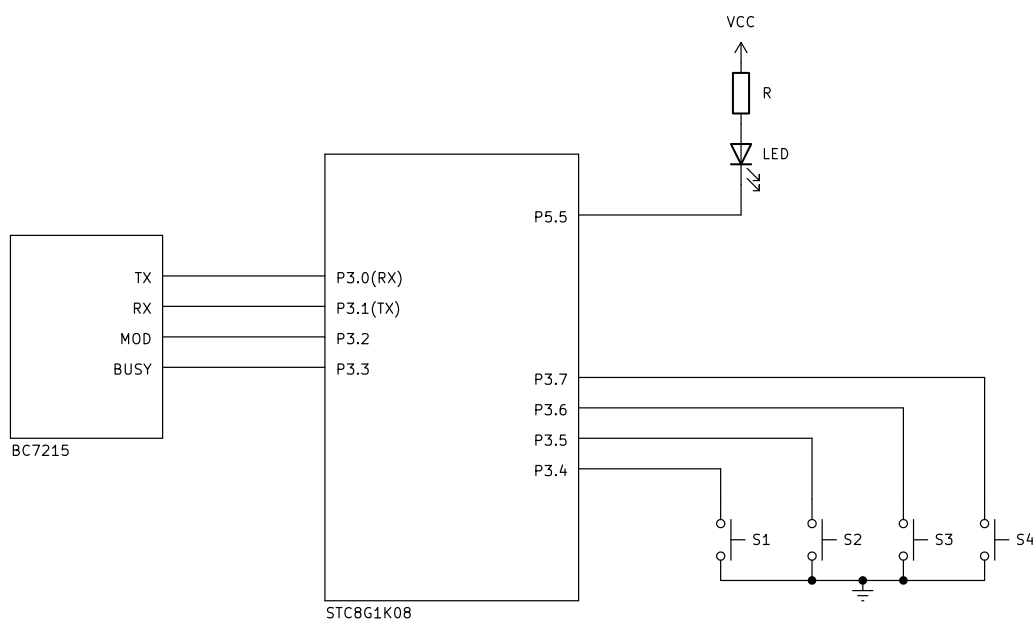
本应用示例使用单片机配合 BC7215 构成一个简单的具有 4 个按键的万能学习型红外遥控器，4 个按键可以分别学习为任意格式的遥控器按键，每个按键可以分别学习不同品牌型号的遥控器，控制不同的设备。

例子提供两种 MCU 的设计，分别为 STM32F103 和 STC8G1K08。

STM32F103 型单片机，使用片上 USART1（PA9, PA10）连接 BC7215，并使用 PB14, PB15 两个 GPIO 口连接 BUSY 和 MOD 两个 BC7215 上的控制引脚。同时，在 PA0-PA3 引脚对地间连接 4 个按键开关，作为按键输入；同时，PC13 引脚上连接一个 LED 作为工作状态显示。系统框图如下：



STC8G1K08 的硬件连接如下图：



## 示例程序功能描述

例子程序实现一个 4 键的学习型遥控器。程序有两个状态：发射模式和学习模式。上电后默认为发射模式，LED 约 1.6s 短暂闪烁一次，表示工作在发射模式。如果按键已经经过学习（上电后内容为空，按键无反应），按下按键，即发射所学习的遥控按键功能。

在发射模式下，同时按 S1 和 S2，则进入学习模式，此时程序控制 BC7215 进入接收状态，PC13 上的 LED 灯快速闪烁，提示用户此时处于学习状态。此时用户再选择按 S1-S4 中的一个键，表示学习内容将存入该键。选择按键后，LED 将变为长亮。在进入此状态后，用遥控器对准 BC7215 的红外接收头按下按键，如果 BC7215 接收解码成功，该红外遥控器的格式及所按按键的内容将被保存，演示程序重新回到发射模式，这时再按相同的按键，就会将所学习的按键复制发送出来。

演示程序设计为每个按键均对应有红外遥控格式和原始数据的存储，因此可以存储 4 个不同格式（不同厂家型号）的遥控信号。如果仅存储同一种格式的遥控器的不同按键，用户可以自行修改程序，改为仅保存一份格式信息，这样可以节省内存。同时，示例中为简单起见，原始数据使用了 4 个最大尺寸的原始数据包数组存储，如果用户希望节省内存，可以根据每个按键原始数据的长度，动态分配内存。

**注意：**某些格式的遥控器，如 RC5 格式和 RC6 格式，每次按键时，发出的数据中会有一个翻转位，每次按键该位均会翻转，而本学习遥控器程序为了简化起见，并未设置此功能，测试中对于 RC5/RC6 码的遥控器，如果接收设备会记忆上次接收到数据并期待翻转位的极性被翻转，使用本程序

复制发送原数据可能会被接收设备拒绝，这取决于接收设备的设计，如果用户需要完全实现原遥控器的功能，需再加上对翻转位的翻转功能。

此例子仅为演示 BC7215 功能用，因此并没有数据的掉电保持和低功耗功能。

## 使用

本例子的驱动设置如下：

```
#define TX_HW_FLOW_CONTROL 0
#define USE_UART_INTERRUPT 1
#define PROCESS_IN_DIFFERENT_THREAD 0
#define ENABLE_RECEIVING 1
#define ENABLE_TRANSMITTING 1
#define ENABLE_FORMAT 1
#define BC7215_MAX_RX_DATA_SIZE 32
```

本例子提供了三种不同的程序例子，功能完全一致，2 种为应用于 STM32F103C8 单片机的程序，另一种是应用于 STC8G1K08 的程序。

"stm32"目录中，提供了基于 ST 公司 HAL 硬件库的例程，使用 HAL 库函数完成设置及硬件操作，将打包文件导入 STM32 CubeIDE 后，即可直接使用；

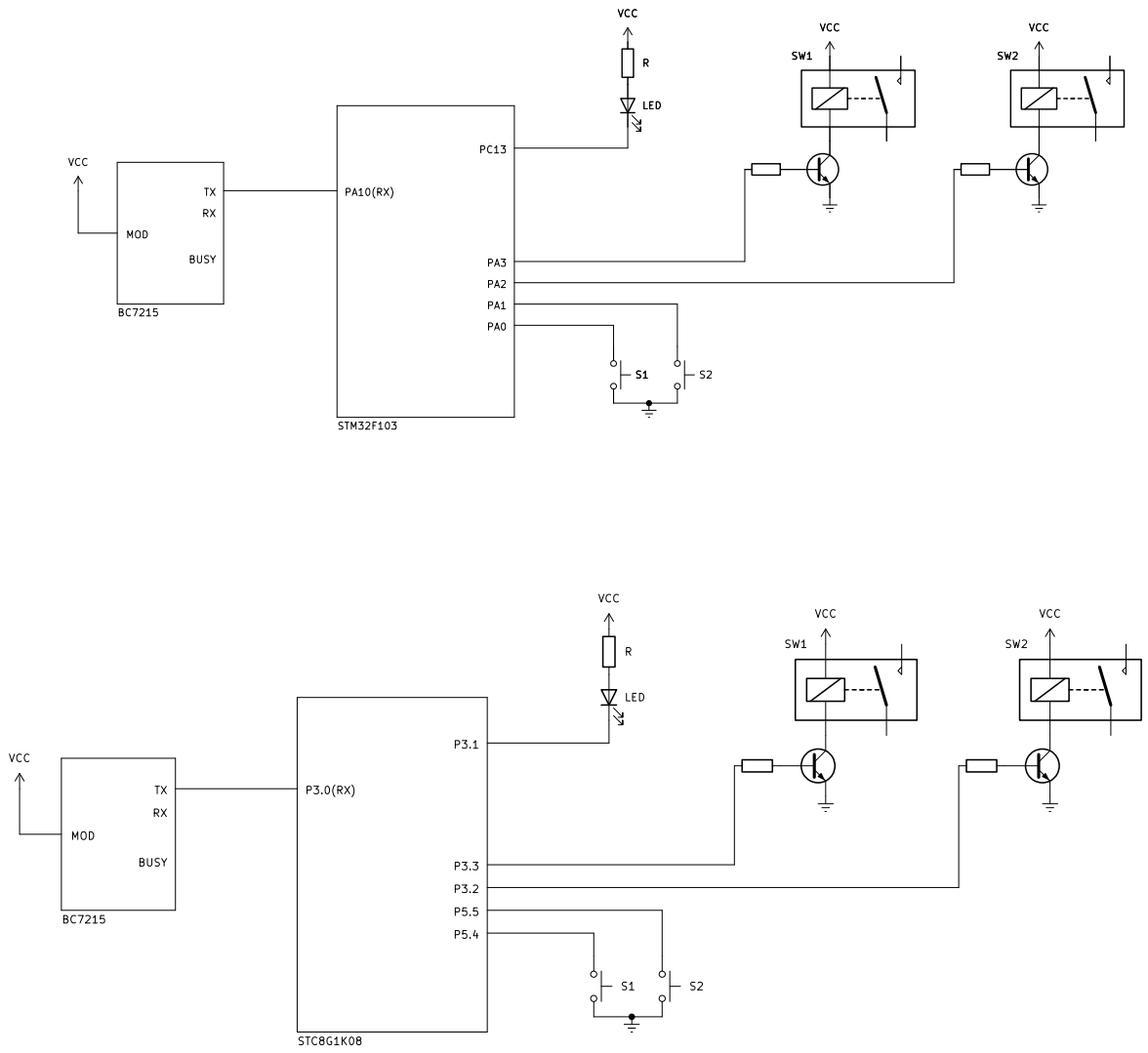
而 stm32\_reg 目录中，则提供了不使用驱动库，直接操作寄存器的版本。寄存器版本的程序，源程序仅需 main.c 单独一个文件，辅助文件仅需要 BC7215 驱动库本身和 ST 公司的寄存器定义头文件 stm32f10x.h，可以用于任何 STM32 开发环境。

C51 目录中为 STC8G1K08 的程序，包括一个 main.c 和一个定义特殊寄存器的头文件。此版本使用 SDCC 编译器编译，因为编译器的限制，BC7215 驱动库中 bc7215\_lib.c 文件做了一处必要的改动，在定义串口接收的缓冲区数组的语句前，增加了 \_\_idata 关键字，以便使用高 128 字节的内部 RAM。

# 2 路红外遥控电源插座

此应用演示了利用 BC7215 构成一个通用的红外控制开关。每个家中均有各种各样的红外遥控器，而每个遥控器上一般均会有一些从不使用的按键，而这个 BC7215 的应用，可以利用这些不使用的红外遥控器按钮，作为红外遥控开关使用。本遥控插座可以使用任何遥控器的空闲按键作为控制开关。

本例子提供了 STM32F103 和 STC8G1K08 两种芯片的应用实例，电路图示意图分别如下：



本例子仅使用 BC7215 的接收模式，所有发射有关的电路及函数均可省略，因此电路和程序均更简单。

## 示例程序功能描述

例子程序实现一个 2 路的通用红外遥控开关，电路本身具有 2 个按键，分别控制 2 路红外开关的学习功能。每一路均可以学习记忆任意遥控器的一个键，学习后当再收到这个键的红外信号时，即控制对应的继电器开关翻转，达到开关电器的目的。

程序有两个状态：普通模式和学习模式。上电后默认为普通模式，LED 约 3.2s 短暂闪烁一次，表示工作在普通模式。当用户按其中一个键后，LED 灯常亮，表示进入学习（配对）状态，此时用遥控器对本机发送红外信号，即可将该信号的信息保存，程序返回正常模式。此后如果再收到同一个按键的红外信号，则会控制对应的继电器翻转。

此例子仅为演示 BC7215 功能用，因此并没有数据的掉电保持和低功耗功能。

## 使用

这个应用的驱动库设置如下：

```
#define TX_HW_FLOW_CONTROL 0
#define USE_UART_INTERRUPT 1
#define PROCESS_IN_DIFFERENT_THREAD 0
#define ENABLE_RECEIVING 1
#define ENABLE_TRANSMITTING 0
#define ENABLE_FORMAT 0
#define BC7215_MAX_RX_DATA_SIZE 16
```

本例子提供了三种不同的程序例子，功能完全一致，2 种为应用于 STM32F103C8 单片机的程序，另一种是应用于 STC8G1K08 的程序。

"stm32"目录中，提供了基于 ST 公司 HAL 硬件库的例程，使用 HAL 库函数完成设置及硬件操作，将打包文件导入 STM32 CubeIDE 后，即可直接使用；

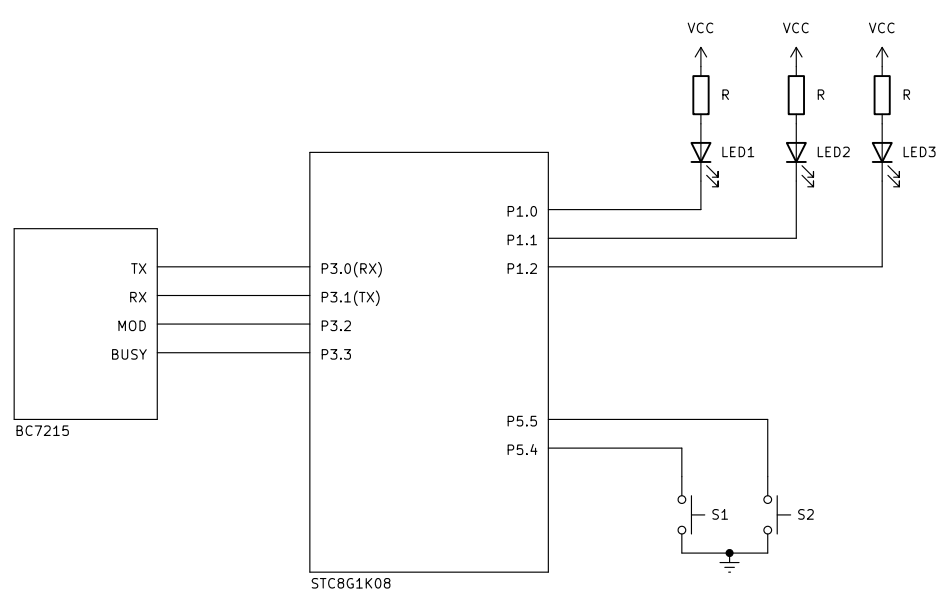
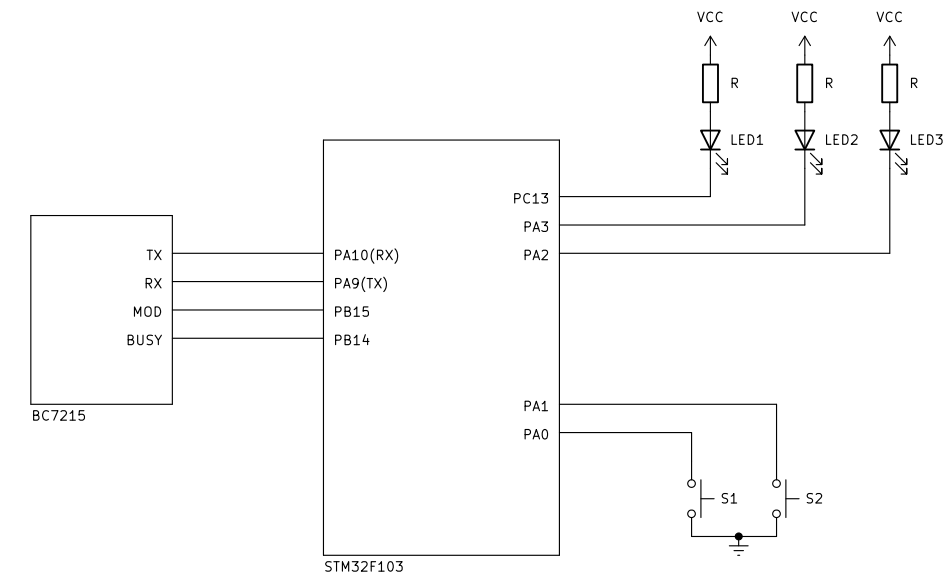
而 stm32\_reg 目录中，则提供了不使用驱动库，直接操作寄存器的版本。寄存器版本的程序，源程序仅需 main.c 单独一个文件，辅助文件仅需要 BC7215 驱动库本身和 ST 公司的寄存器定义头文件 stm32f10x.h，可以用于任何 STM32 开发环境。

C51 目录中为 STC8G1K08 的程序，包括一个 main.c 和一个定义特殊寄存器的头文件。此版本使用 SDCC 编译器编译。使用其他编译器有可能会需要做小的修改。

### 3. 红外数据传输

此应用演示了使用 BC7215 作为红外收发器件，将其应用于红外数据传输的用法。红外遥控器作为广泛使用的电子设备，其通讯可靠性已经过验证，但其通讯速率比较低，适宜作为少量数据对速度无要求时的传输，如传感器数据，指令数据等。例子中演示了使用 CRC 校验和不使用 CRC 校验两种方式。

本例子提供了 STM32F103 和 STC8G1K08 两种芯片的应用实例，电路图示意图分别如下：





本例子中使用了 NEC 编码作为红外传输的编码方式。

## 示例程序功能描述

本示例仅为演示数据传输功能，因此并无实际用途。系统由 2 个按键和 3 只 LED 组成。分别为

LED1——工作指示

LED2——接收数据 CRC 正确指示

LED3——接收数据 CRC 错误指示

上电后，LED1 每隔 3.2 秒闪烁一次，表明系统在工作，系统默认处在接收状态。

如果按下 S1 键，系统会从红外发送一个 21 个字节的数据（例子固定为 0-20 的连续数字），传送完数值数据后，在传送一段 ASCII 码数据，内容为“Hello World”，两段数据的末尾均添加了 CRC-8 校验值。在红外传输期间，LED1 会点亮。如果使用 Windows BC7215 演示软件，选择通讯演示功能，配合 BC7215 演示板，即可在电脑上看到输出的数据。当软件选择 TEXT 模式时，上面的 21 字节数据因为是不可显示字符，所以没有显示，而文字部分则可以看到还原的文字。当软件选择 HEX 模式时，则可以看到两条数据信息，分别为 22 个字节和 12 个字节，末尾最后一个字节是 CRC 字节。

如果按下 S2 键，系统发送同样的数据，但末尾不添加 CRC-8 校验值。在红外传输期间，LED1 会点亮。如果使用 Windows BC7215 演示软件，选择通讯演示功能，配合 BC7215 演示板，即可在电脑上看到输出的数据。当软件选择 TEXT 模式时，上面的 21 字节数据因为是不可显示字符，所以没有显示，而文字部分则可以看到还原的文字。当软件选择 HEX 模式时，则可以看到两条数据信息，分别为 22 个字节和 12 个字节，末尾最后一个字节是 CRC 字节。

PC 上软件如果打开了 CRC 校验功能，用户可以看到按 S1 键时，接收数据 CRC 正确，而 S2 键发送的数据，则 CRC 错误。

没有按键时，系统工作在接收模式。此时如果接收到红外信号，不管内容是什么，程序都会对其进行 CRC 校验，如果校验正确，会令 LED2 闪烁一下，如果校验不正确，会令 LED3 闪烁一下。用户可以用任何普通遥控器测试，因为普通遥控器不带 CRC 校验，因此将看到 LED3 闪烁，如果用 PC 上的 BC7215 演示软件，或者另一个同样的电路按 S1 键，则可以看到 LED2 闪烁。

源程序中已标记了数据接收完成的位置，用户亦可根据需要对所接收数据做任何处理。

## 使用

这个应用的驱动库设置如下：

```
#define TX_HW_FLOW_CONTROL 0
#define USE_UART_INTERRUPT 1
#define PROCESS_IN_DIFFERENT_THREAD 0
#define ENABLE_RECEIVING 1
#define ENABLE_TRANSMITTING 1
```

```
#define ENABLE_FORMAT 0
```

```
#define BC7215_MAX_RX_DATA_SIZE 32
```

本例子提供了三种不同的程序例子，功能完全一致，2种为应用于 STM32F103C8 单片机的程序，另一种是应用于 STC8G1K08 的程序。

"stm32"目录中，提供了基于 ST 公司 HAL 硬件库的例程，使用 HAL 库函数完成设置及硬件操作，将打包文件导入 STM32 CubeIDE 后，即可直接使用；

而 stm32\_reg 目录中，则提供了不使用驱动库，直接操作寄存器的版本。寄存器版本的程序，源程序仅需 main.c 单独一个文件，辅助文件仅需要 BC7215 驱动库本身和 ST 公司的寄存器定义头文件 stm32f10x.h，可以用于任何 STM32 开发环境。

C51 目录中为 STC8G1K08 的程序，包括一个 main.c 和一个定义特殊寄存器的头文件。此版本使用 SDCC 编译器编译。因为编译器的限制，BC7215 驱动库中 bc7215\_lib.c 文件做了一处必要的改动，在定义串口接收的缓冲区数组的语句前，增加了 \_\_idata 关键字，以便使用高 128 字节的内部 RAM。