

IRIG-B 解码模块 数据手册

V1.09



成都云智优创科技有限公司

公司网站: www.cloudintel.cn

淘宝官网: cloudintel.taobao.com

目 录

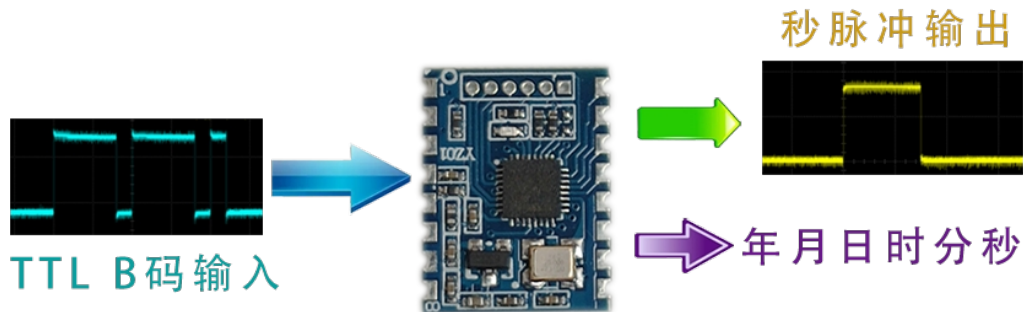
一、	IRIG-B 解码模块简介	3
二、	引脚定义	3
三、	性能参数	4
1.	DC Characteristics	4
2.	AC Characteristics	4
四、	关键时序	5
3.	B 码同步沿与 PPS_INT	5
4.	B 码同步沿与 LED_PPS	6
五、	串口通信	7
六、	IO 串行通信	7
1.	IO 时序	7
2.	数据格式	8
七、	参考电路	9
八、	用户编程	9
1.	概述	9
2.	使用串口方式与 B 码模块通信	9
3.	使用 IO 串行时序与 B 码通信	9
4.	相关定义：	10
九、	订购型号	11

一、 IRIG-B 解码模块简介

IRIG-B 解码模块是专门用于对 IRIG-B 码进行解码的模块，能解码出同步秒脉冲，同时通过串口或 IO 接口可以获取同步时间，用于对设备进行校时。

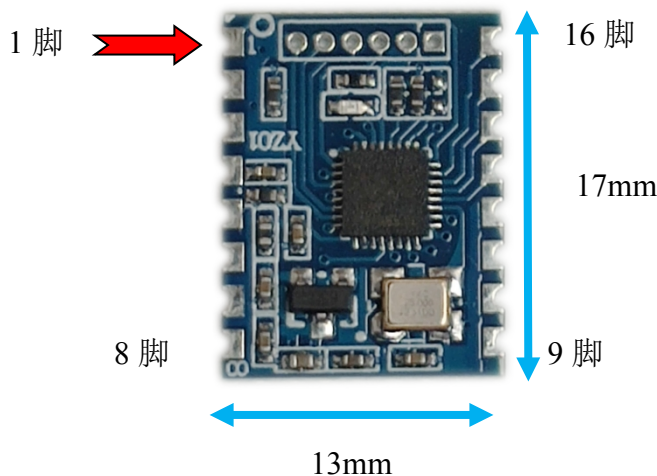
模块具有处理速度快，输出精度高的特点，秒脉冲误差在 10ns 左右，能满足设备对时间的高精度要求。模块使用简单，输入 TTL 电平的 B 码，就能自动解码出同步秒脉冲和时间，对单片机性能没有任何要求。

从模块获取时间有两种方式：1) 通过串口接收，模块每秒钟会发送一次；2) 通过 IO 口模拟串行时序获取时间，这种方式不占用串口，适合串口资源不够的应用。



二、 引脚定义

解码模块体积很小，可以直接通过邮票孔焊接到主板上。



引脚号	名称	方向	功能
1	IO2	--	预留，不连接
2	IO1	--	预留，不连接
3	IO0	--	预留，不连接
4	LED_UART	out	串口发送指示灯，当串口输出时间时闪烁，低电平有效
5	LED_SPI	out	串行通信指示灯，当串行通信时闪烁，低电平有效
6	LED_B	out	B 码接收指示灯，接收到 B 码后闪烁，低电平有效
7	GND *	in	电源负
8	3V3 *	in	电源正，输入 3.3V
9	RST_N *	in	模块复位信号，低有效。可以连接复位芯片，也可以用 RC 设计一个简单的自复位电路
10	B_IN *	in	IRIG-B 码输入引脚，TTL 电平
11	PPS_INT	out	秒中断信号，当解码出 B 码同步沿后输出方波，上升沿有效，脉宽 7.5ms，精度 10ns 左右
12	LED_PPS	out	秒脉冲指示灯，当解码出 B 码同步沿后闪烁，脉宽 107.5ms，低电平有效
13	UART_TX	out	串口发送引脚，波特率 9600
14	IO_CS	in	IO 串行通信片选，低有效
15	IO_SDA	out	IO 串行通信数据
16	IO_SCK	in	IO 串行通信时钟，上升沿出数据

（*表示必须使用的引脚，其它引脚根据需要选择使用）

三、性能参数

1. DC Characteristics

名称	描述	最小值	最大值
VCC	电源电压	2.5V	3.44V
VIO	IO 引脚电压	2.5V	3.44V
Ambient Temperature	工作环境温度(工业级)	-40℃	+85℃

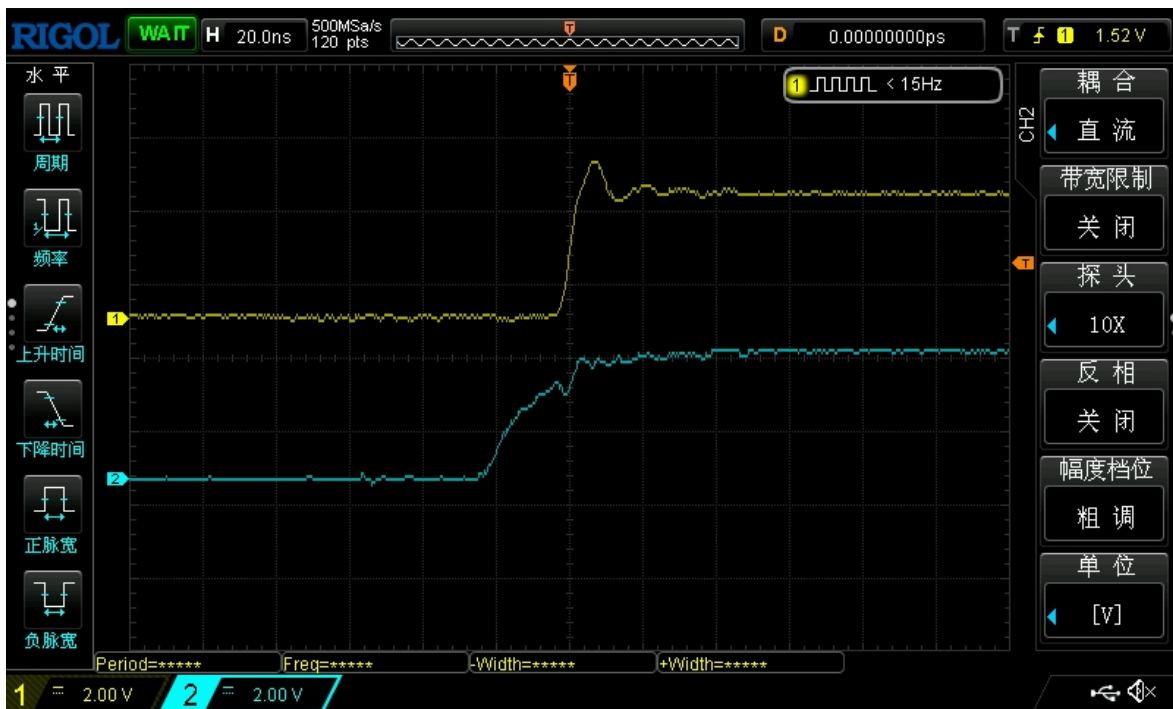
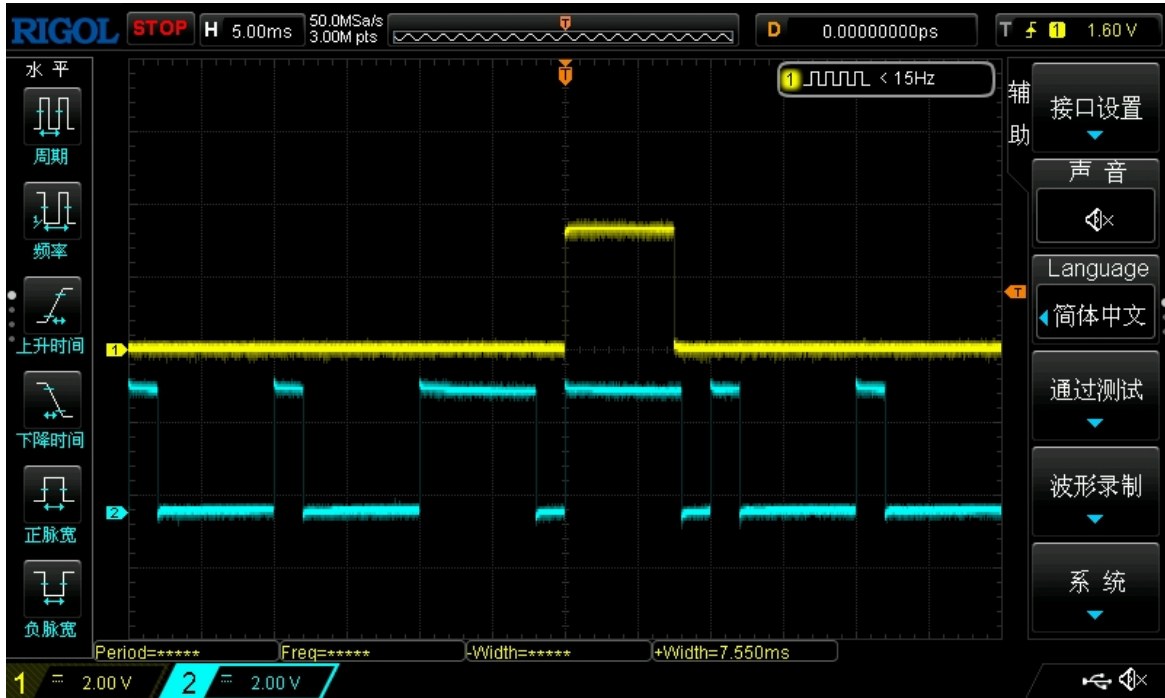
2. AC Characteristics

名称	描述	典型值
PPS_INT Time Delay	PPS_INT 秒中断脉冲相对 B 码同步脉冲时延	10ns (从 B 码同步沿 70%位置开始计算)
LED_PPS Time Delay	LED_PPS 秒脉冲相对 B 码同步脉冲时延	10ns (从 B 码同步沿 70%位置开始计算)
IO_SCK Speed	IO_SCK 最大时钟	50Mhz
UART Baudrate	串口波特率	9600bps
Frequency Tolerance(25℃)	频率稳定度(25℃)	±10ppm

四、 关键时序

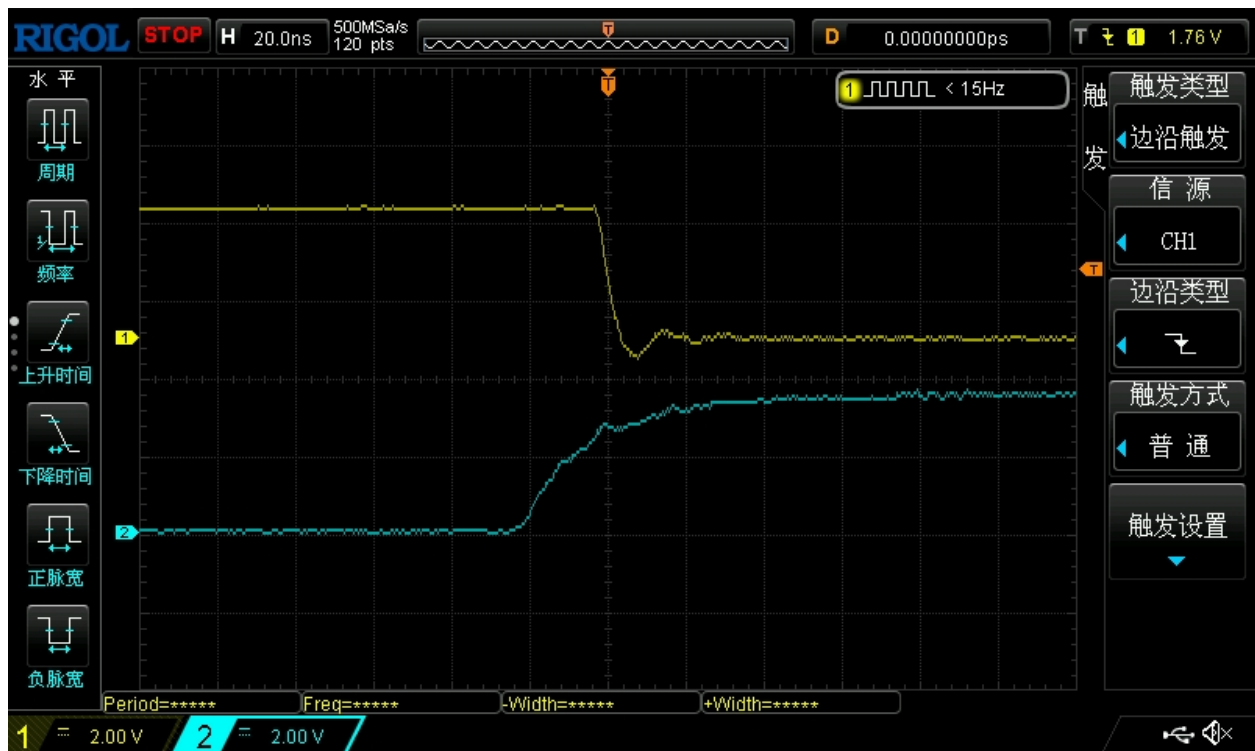
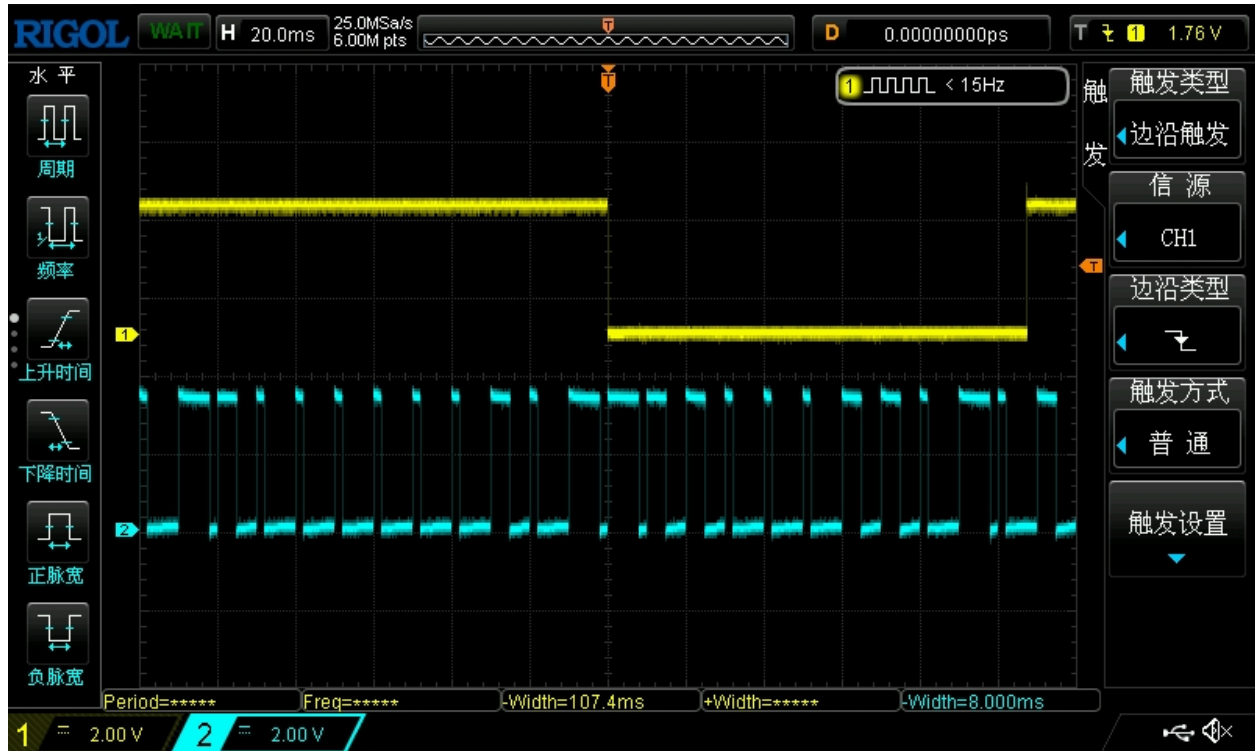
1. B 码同步沿与 PPS_INT

黄色为 PPS_INT，蓝色为 B 码。PPS_INT 滞后 B 码同步沿约 10ns（从 B 码同步沿 70%位置开始计算）。



2. B 码同步沿与 LED_PPS

黄色为 LED_PPS，蓝色为 B 码。LED_PPS 滞后 B 码同步沿约 10ns（从 B 码同步沿 70%开始计算）。



五、 串口通信

串口波特率 9600bps，1 位起始位，1 位停止位，无校验。数据一共 12 个字节，96bit。当模块的秒脉冲发出后，100us 以内就会发送串口数据。

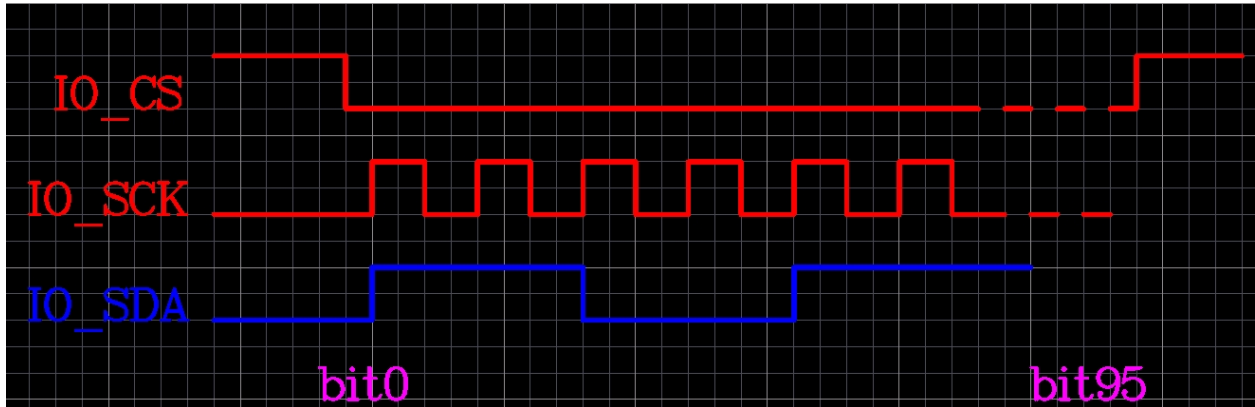
输出格式：

字节	说明
[7:0]	报文头 0x5A
[9:8]	bit8 为刷新标志，刷新一次会反转一次； bit9：保留
[16:10]	秒，为 BCD 码，低 4 位为个位，高 3 位为十位
[23:17]	分，为 BCD 码，低 4 位为个位，高 3 位为十位
[29:24]	时，为 BCD 码，低 4 位为个位，高 2 位为十位
[39:30]	一年中的天数，低 4 位为个位，中 4 位为十位，高 2 位为百位
[47:40]	年的十位和个位，低 4 位为个位，高 4 位为十位
[64:48]	一天中的秒数，17 位，为 16 进制
[79:65]	B 码标志位,一般情况下不需要使用 bit65: 闰秒预告 bit66: 闰秒标志 bit67: 夏时制预告 bit68: 夏时制标志 bit69: 时间偏移符号位 bit73:70: 时间偏移 bit74: 时间偏移 0.5h bit78:75: 时间质量 bit79: 校验位
[87:80]	异或校验位，bit8-bit79，9 个字节，按字节异或
[95:88]	结束符 0xA5

六、 IO 串行通信

1. IO 时序

CS 和 SCK 由处理器控制输出，解码模块在 SCK 的上升沿输出数据，处理器在 SCK 的下降沿读取数据。数据低位在前，一共 96 位。



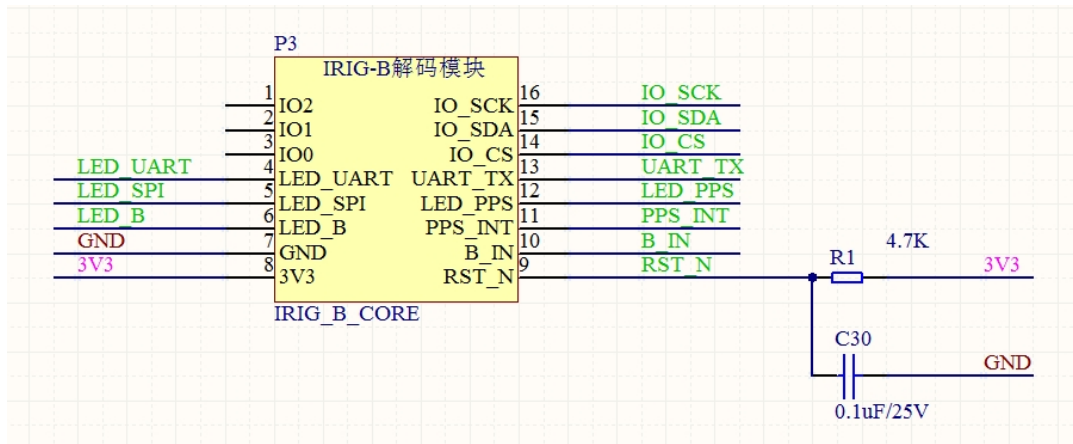
2. 数据格式

(和串口格式一样)

位号	说明
[7:0]	报文头 0x5A
[9:8]	bit8 为刷新标志, 刷新一次会反转一次; bit9: 保留
[16:10]	秒, 为 BCD 码, 低 4 位为个位, 高 3 位为十位
[23:17]	分, 为 BCD 码, 低 4 位为个位, 高 3 位为十位
[29:24]	时, 为 BCD 码, 低 4 位为个位, 高 2 位为十位
[39:30]	一年中的天数, 低 4 位为个位, 中 4 位为十位, 高 2 位为百位
[47:40]	年的十位和个位, 低 4 位为个位, 高 4 位为十位
[64:48]	一天中的秒数, 17 位, 为 16 进制
[79:65]	B 码标志位, 一般情况下不需要使用 bit65: 闰秒预告 bit66: 闰秒标志 bit67: 夏时制预告 bit68: 夏时制标志 bit69: 时间偏移符号位 bit73:70: 时间偏移 bit74: 时间偏移 0.5h bit78:75: 时间质量 bit79: 校验位
[87:80]	异或校验位, bit8-bit79, 9 个字节, 按字节异或
[95:88]	结束符 0xA5

七、 参考电路

详细电路请参考 IRIG-B 评估板原理图。



八、 用户编程

1. 概述

为了简化用户编程，我们提供了相关驱动文件 IRIG_B_CORE.c 和 IRIG_B_CORE.h。驱动使用 C 语言编写，开发板提供 STM32F107 的例程。

驱动通过宏定义选择使用串口或 IO 口通信，如果注释掉宏定义即为 IO 口通信。

```
#define USE_UART //使用串口通信
```

2. 使用串口方式与 B 码模块通信

这种方式最简单，只需要调用一个解析函数就可以得到精确时间。

首先需要在 IRIG_B_CORE.h 中定义 #define USE_UART。然后调用解析函数，解析串口接收到的数据。解析函数如下：

```
err_code Irig_Parse(unsigned char *val, sTime *time)
```

*val: 串口接收到的数据指针，大小为 12 字节；

*time: 解析后输出的时间结构体指针；

err_code: 解析结果，为枚举类型；

注：如果想提高授时精度，可以把模块输出的 PPS_INT 连接到单片机的外中断引脚，当中断触发后，首先判断毫秒时钟大于 900ms，如果是，毫秒时钟清 0，秒加一。如果不是，仅对毫秒时钟清 0。

3. 使用 IO 串行时序与 B 码模块通信

当单片机串口不够用时，可以使用这种方式，实现定时。驱动会使用到 3 个 IO，所以需要根据不同的硬件适配单片机的 IO 口。

1) 注释掉宏定义：

```
//#define USE_UART
```

2) 通过宏定义，适配 IO 口

修改 IRIG_B_CORE.h，红色部分需要根据不同的单片机进行修改；

```
#define IO_CS_L    GPIO_ResetBits(GPIOD,GPIO_Pin_15)    //CS=0
#define IO_CS_H    GPIO_SetBits(GPIOD,GPIO_Pin_15)      //CS=1
#define IO_SCK_L   GPIO_ResetBits(GPIOD,GPIO_Pin_14)    //SCK=0
#define IO_SCK_H   GPIO_SetBits(GPIOD,GPIO_Pin_14)      //SCK=1
#define IO_SDA     GPIO_ReadInputDataBit(GPIOD,GPIO_Pin_13) //SDA 输入
```

3) 调用初始化函数 Irig_Init()，设置 IO 的初始化状态

4) 调用 IO 串行读取函数，获取数据，

可以周期性调用读取，也可以在收到 PPS_INT 中断后读取，推荐后者。

```
void Get_Irig_Data(unsigned char *val)
```

*val: 接收数组指针，数组大小为 12 字节

5) 调用 B 码解析函数，获取时间

```
err_code Irig_Parse(unsigned char *val, sTime *time)
```

*val: 串口接收到的数据指针，大小为 12 字节；

*time: 解析后输出的时间结构体指针；

err_code: 解析结果，为枚举类型；

注：如果想提高授时精度，可以把模块输出的 PPS_INT 连接到单片机的外中断引脚，当中断触发后，首先判断毫秒时钟大于 900ms，如果是，毫秒时钟清 0，秒加一。如果不是，仅对毫秒时钟清 0。

4. 相关定义说明

1) sTime 时间结构体

```
typedef struct{
    unsigned char year;    //年
    unsigned char month;  //月
    unsigned char date;    //日
    unsigned char hour;    //时
    unsigned char minute;  //分
    unsigned char second;  //秒
}sTime;
```

2) err_code 枚举类型

```
typedef enum{
```

```
Parse_success=0,           //解析成功
No_update,                 //B 码没有更新
Head_error,                //报文头错误
Sec_check_error,           //秒校验失败
XOR_check_error,           //异或校验失败
End_error                  //结束符错误
}err_code;
```

九、 订购型号

Type Number	Temperature Range	Package Type
YZ-B-1S	-40℃~+85℃	邮票孔



云智优创
CLOUDINTEL

» 身边的工控专家，你信赖的合作伙伴
» 十年工控研发经验，支持企业级定制



成都云智优创科技有限公司

公司网站: www.cloudintel.cn

淘宝官网: cloudintel.taobao.com

2022 年 11 月