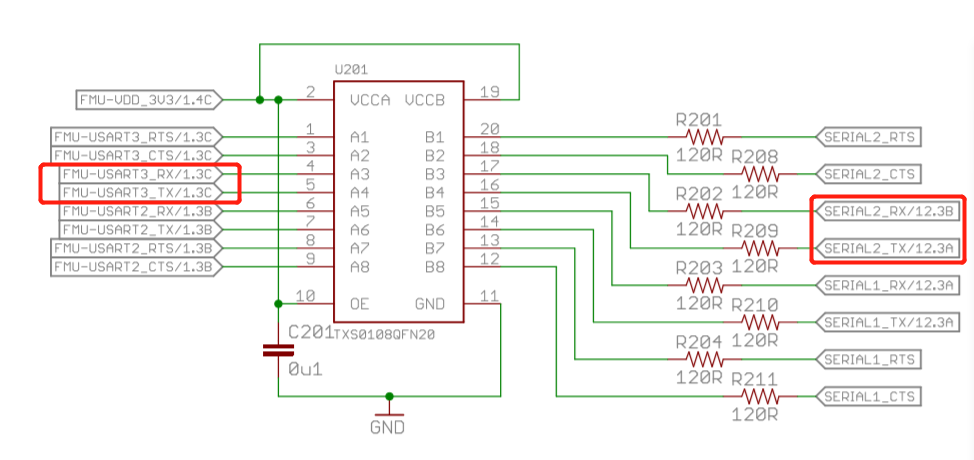
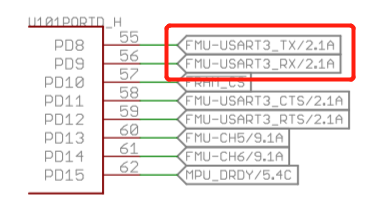
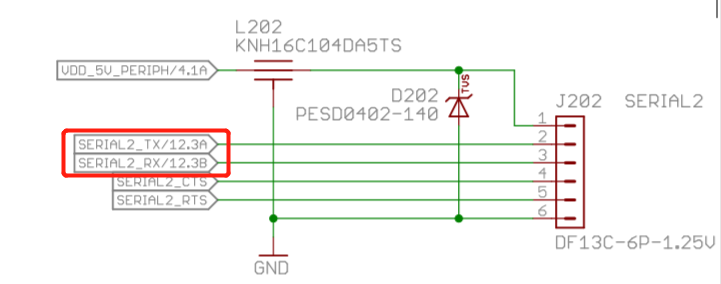
串口：打印Hello world实验

1. **硬件说明**

本实验基于主处理器（STM32F427）芯片进行，首先来看一下飞控板的相关硬件说明。如图1. 所示，飞控板串口3的相关硬件说明。



1. （2）



（3）

图1. 飞控板串口3的相关硬件说明

图1. 的三幅图可以看出，stm32F427芯片的串口3，对应的是pixhawk的SERIAL2。在（2）图中，芯片TXS0108是用来进行电平转换的芯片。在[pixhawk说明手册](https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk.html)上，J202对应的是SERIAL2，线序如图2.所示。

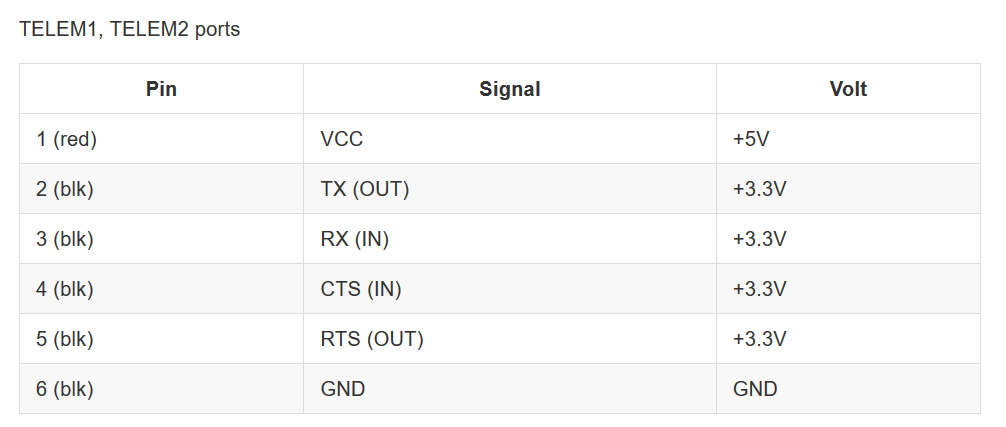


图2. Pixhawk串口线序

1. **硬件连接**

需要通过串口转USB电路将飞控板与上位机相连接起来。我们使用CH340串口转USB模块。模块如图3.所示。



图3. CH340串口转USB模块

按照模块上的丝印，按照表1. 的方式将串口转USB模块与飞控板相连接。

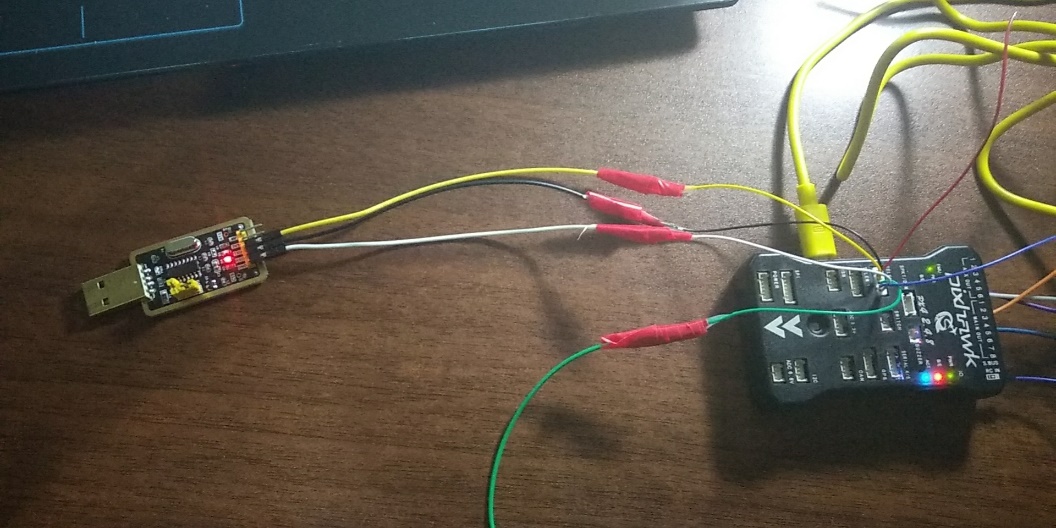
表1. 接线方式

|  |  |
| --- | --- |
| **CH340串口转USB** | **飞控板的串口** |
| VCC | VCC |
| GND | GND |
| TX | RX |
| RX | TX |

完成上述连接后，将CH340模块插到上位机上，并安装相应的硬件驱动程序。

CH340驱动：<http://www.opdown.com/soft/74004.html>

CH340与pixhawk的连接如下图所示，白线为GND，黄线对应pixhawk serial2的TX，黑线对应pixhawk serial2的RX。



1. **工程配置**

以STM32F429VIT6芯片建立工程文件。按照图4. 所示的步骤配置芯片的串口3.

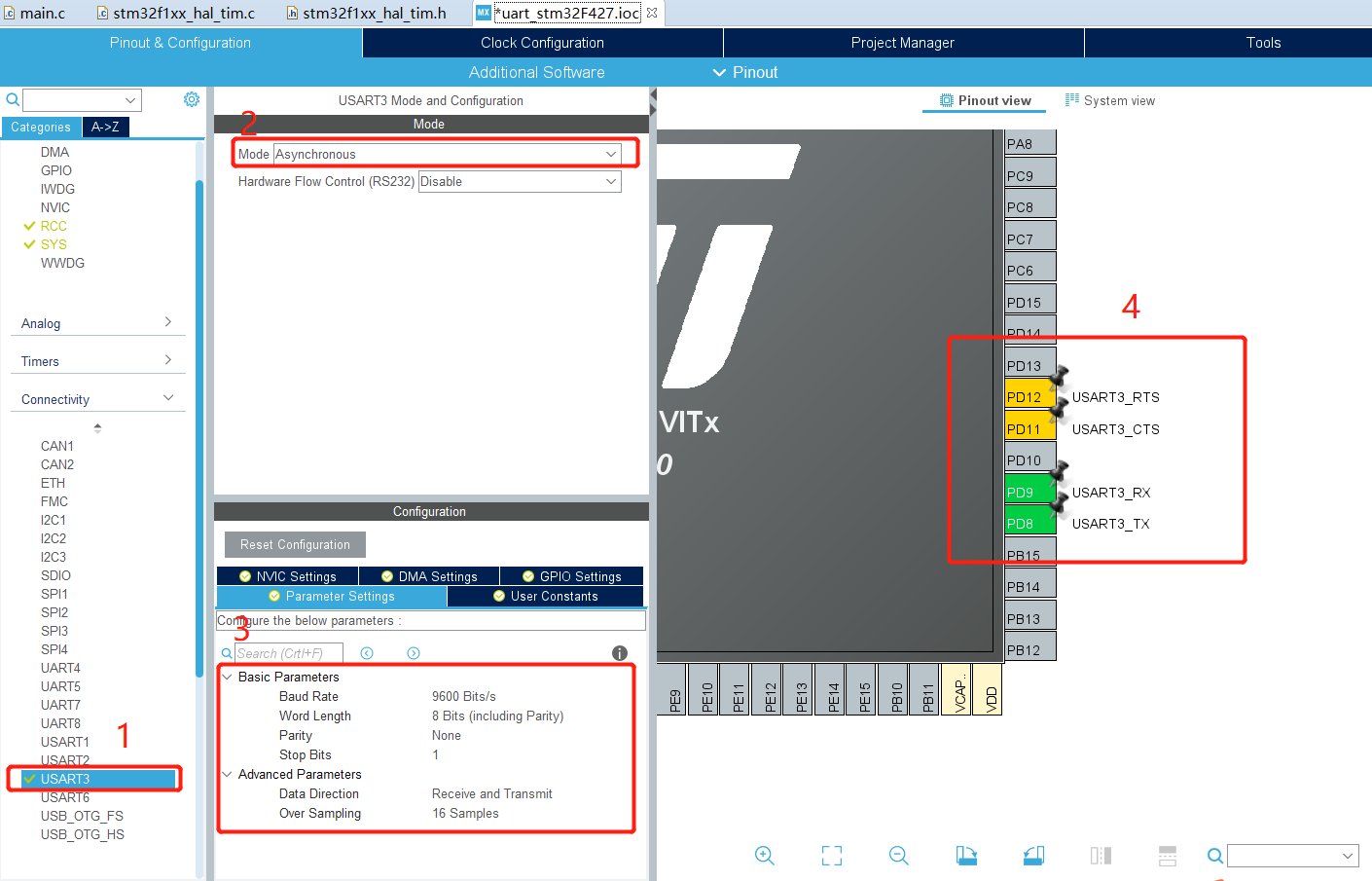


图3. 配置串口3

1）点击USATR3

2）设置MODE为异步通信(Asynchronous)

3）基础参数：波特率为9600 Bits/s。传输数据长度为8 Bit。奇偶检验无，停止位1 接收和发送都使能

4）GPIO引脚设置 USART3\_RX、USART3\_TX、USART3\_CTS、USART3\_RTS

配置完成之后，在main.c文件中，修改main函数如下所示：

|  |
| --- |
| **int** **main**(**void**)  {  /\* USER CODE BEGIN 1 \*/  /\* USER CODE END 1 \*/    /\* MCU Configuration--------------------------------------------------------\*/  /\* Reset of all peripherals, Initializes the Flash interface and the Systick. \*/  HAL\_Init();  /\* USER CODE BEGIN Init \*/  /\* USER CODE END Init \*/  /\* Configure the system clock \*/  SystemClock\_Config();  /\* USER CODE BEGIN SysInit \*/  /\* USER CODE END SysInit \*/  /\* Initialize all configured peripherals \*/  MX\_GPIO\_Init();  MX\_USART3\_UART\_Init();  /\* USER CODE BEGIN 2 \*/  uint8\_t pData[]="hello world";//通过字符串数组定义“hello world”字符串  /\* USER CODE END 2 \*/  /\* Infinite loop \*/  /\* USER CODE BEGIN WHILE \*/  **while** (1)  {  /\* USER CODE END WHILE \*/  //HAL\_UART\_Transmit(huart3,&pData,11,100);  HAL\_UART\_Transmit(&huart3,(uint8\_t\*)pData,**sizeof**(pData),100);//不断通过串口3发送，pData中的数据，sizeof计算pData中的字节数  /\* USER CODE BEGIN 3 \*/  }  /\* USER CODE END 3 \*/  } |

1. **重点代码解析**

USART3的初始化函数注解：

|  |
| --- |
| **static** **void** **MX\_USART3\_UART\_Init**(**void**)  {  /\* USER CODE BEGIN USART3\_Init 0 \*/  /\* USER CODE END USART3\_Init 0 \*/  /\* USER CODE BEGIN USART3\_Init 1 \*/  /\* USER CODE END USART3\_Init 1 \*/  huart3.Instance = USART3;  huart3.Init.BaudRate = 9600;//波特率设置为9600  huart3.Init.WordLength = UART\_WORDLENGTH\_8B;// 数据位的个数为8位  huart3.Init.StopBits = UART\_STOPBITS\_1;//停止位  huart3.Init.Parity = UART\_PARITY\_NONE;//无奇偶校验位  huart3.Init.Mode = UART\_MODE\_TX\_RX;//模式为收发模式  huart3.Init.HwFlowCtl = UART\_HWCONTROL\_NONE;  huart3.Init.OverSampling = UART\_OVERSAMPLING\_16;  **if** (HAL\_UART\_Init(&huart3) != *HAL\_OK*)//初始化串口3  {  Error\_Handler();  }  } |

|  |
| --- |
| HAL\_StatusTypeDef **HAL\_UART\_Transmit**(UART\_HandleTypeDef \*huart, uint8\_t \*pData, uint16\_t Size, uint32\_t Timeout) |

功能：串口发送指定长度的数据。如果超时没发送完成，则不再发送，返回超时标志（HAL\_TIMEOUT）。

参数：

UART\_HandleTypeDef \*huart uart的句柄名，huart3即为串口3

\*pData 需要发送的数据

Size 发送的字节数

Timeout 最大发送时间，发送数据超过该时间退出发送

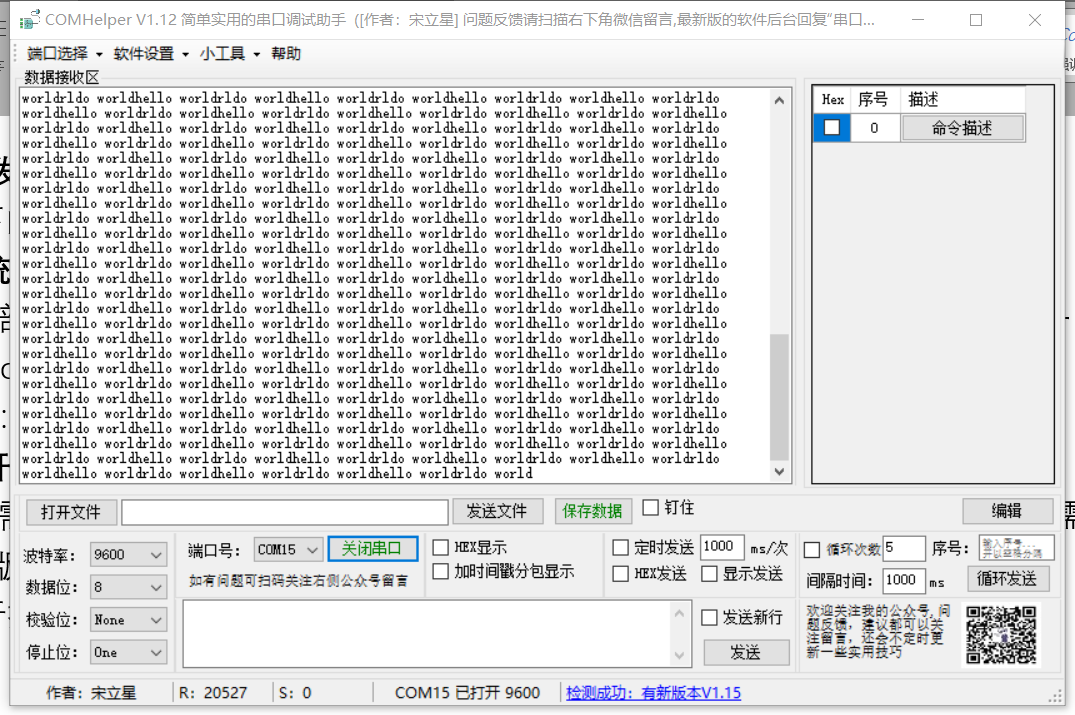
1. **串口调试助手的使用**

硬件连接完毕，且把程序烧写到芯片中之后，需要在上位机打开串口调试助手，观察实验效果。

串口调试助手的使用可参考：<https://jingyan.baidu.com/article/54b6b9c0b3c8c02d583b4707.html>

注意波特率的配置，要和程序中初始化的串口3的波特率相同，否则会出现通信错误。

1. **实验效果图**



1. **延伸**
2. 将定时器中断与串口发送结合在一起，实现每1s发送一遍“hello world”。
3. 示例程序中字符串没有换行显示，尝试换行显示hello world。