ADC：电压采集实验

1. **硬件说明**

参考电路原理图的相关部分，可以看到，pixhawk飞控板的主控芯片（stm32F427）引出了两路ADC，分别是PC3和PC4。具体情况如图1. 所示。

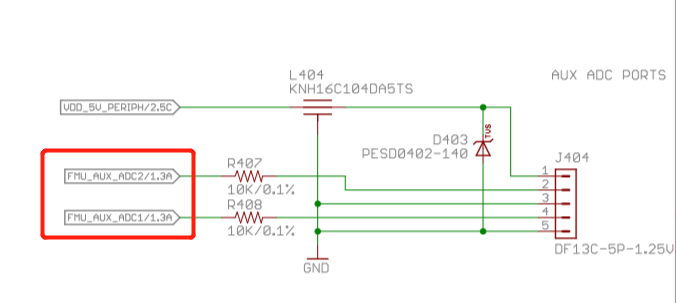
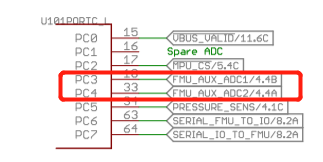


图1. ADC相关相关硬件

参考STM32F427的开发手册，STM32F4有三个12位的ADC，分别是ADC1、ADC2、ADC3，其中ADC1和ADC2均有16个输入通道，ADC3有若干个输入通道（根据芯片类型不同有差异）。PC3可以对应ADC1、ADC2、AD3的13通道，PC4可以对应ADC1、ADC2的14通道。

**12位**：也叫ADC分辨率、采样精度。先来看看二进制的12位可表示0-4095个数，也就是说转换器通过采集转换所得到的最大值是4095，stm32可以转换的电压范围是0v-3.3v，转换器就会把0v-3.3v平均分成4096份。设转换器所得到的值为x，所求电压值为y。

那么就有：

**16通道**：芯片上有16个引脚是可以接到模拟电压上进行电压值检测的。16个通道不是独立的分配给3个转换器（ADC1、ADC2、ADC3）使用，有些通道是被多个转换器共用的。例如PC3就是被ADC1、ADC2、ADC3公用的13通道。

STM32 ADC的一些概念（单次转换、连续转换、扫描模式）

**1）单次转换**：一次转换结束后，需要手动开启才能进行下一次转换。

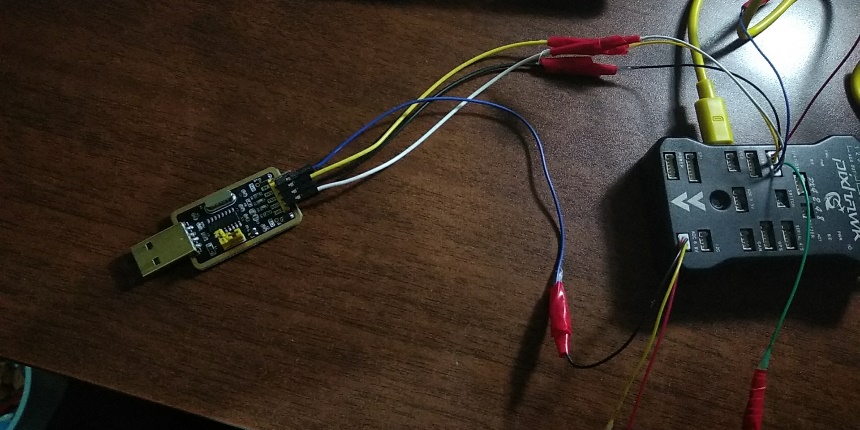
**2）连续转换**：一次转换结束后继续下一次转换。

**3）扫描模式**：对多通道的转换进行逐个转换。若此时开启单次转换模式，扫描完所有通道后需要手动启动下一次扫描转换；若此时开启连续转换模式，扫描完成后自动进入下一次扫描转换，扫描模式需要配合DMA才能读取数据。

1. **硬件连接**

将pixhawk上的adc接口线引出，中间的一条线为STM32F427的ADC1，对应的引脚为PC1，将该线与CH340板子上的3.3V电压输出相连接，即位ADC的输入电压为3.3v。

接线图如下所示：



1. **工程配置**
2. 配置好串口3相关的参数，参考前边的实验。
3. 配置好ADC相关参数，如图2. 所示：

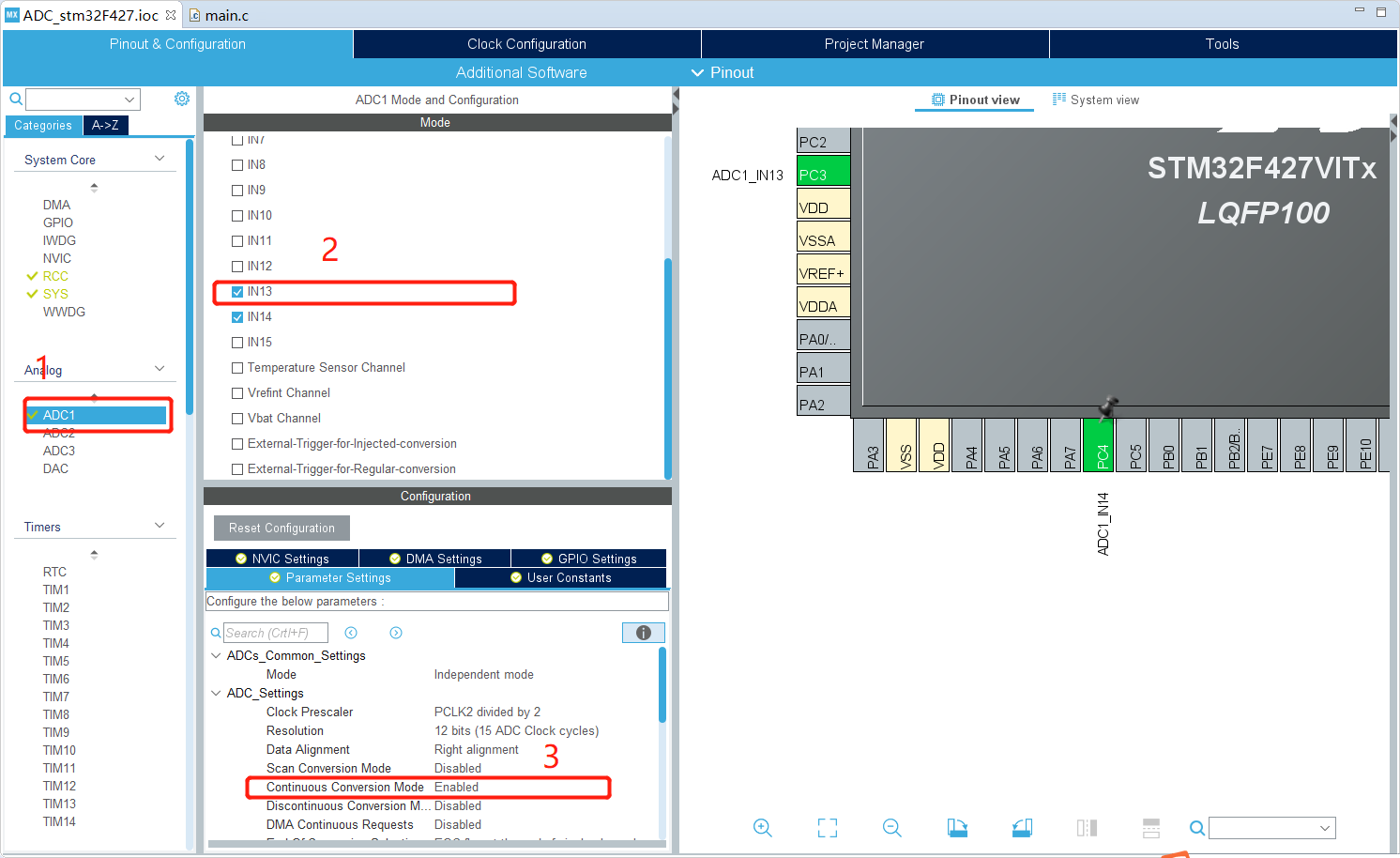


图2. ACD配置

1. 代码修改的内容

在main.c中加上。

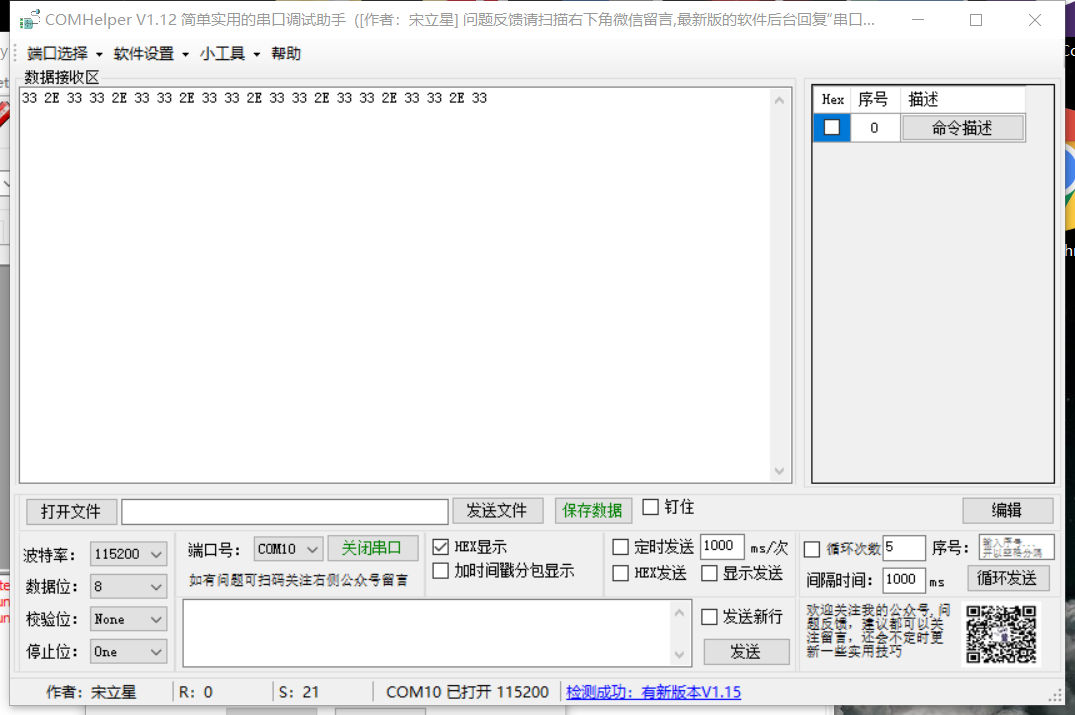
|  |
| --- |
| //在main函数之前，添加全局变量，用来存储ADC转换后的值  uint16\_t ADC\_Value; |

在while（1）中加上。

|  |
| --- |
| **while** (1)  {  /\* USER CODE END WHILE \*/  HAL\_ADC\_Start(&hadc1); //启动ADC转换  HAL\_ADC\_PollForConversion(&hadc1, 50); //等待转换完成，50为最大等待时间，单位为ms  **if**(HAL\_IS\_BIT\_SET(HAL\_ADC\_GetState(&hadc1), HAL\_ADC\_STATE\_REG\_EOC))  {  ADC\_Value = HAL\_ADC\_GetValue(&hadc1); //获取AD值  **float** vol =ADC\_Value\*3.3/4096;//计算得到电压值，变量类型为float  uint8\_t pData[4]="";  **sprintf**(pData,"%.1f",vol);//将浮点数，转化为字符串，以供串口输出  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  }  HAL\_Delay(1000);//延时1s的时间，即为每1s进行一次ADC转换  /\* USER CODE BEGIN 3 \*/  } |

1. **实验效果**

由串口显示出电压值，3.3，采用的ASCII的编码方式显示，33对应的’3’,2E对应的’.’

****

1. **延伸**

将ADC与定时器实验结合在一起，完成每1s，完成一次ADC转换的功能。