

使用 STM32CubeMX 实现 USB 虚拟串口的环回测试功能

前言

客户在 STM32F401RET6 中使用到了 USB 的虚拟串口功能。要求提供一个在 STM32Cube 中实现虚拟串口进行数据环回测试功能的范例程序。因为在我们目前所提供范例程序中，并没有适合客户需求的范例，所以我们在 STM32CubeMX 中创建一个范例程序给客户进行参考。

环境需求如下：

硬件环境:STM32F401C-Discovery

电脑操作系统: window 7.x

集成开发环境: MDK、IAR

驱动: ST VCP Driver

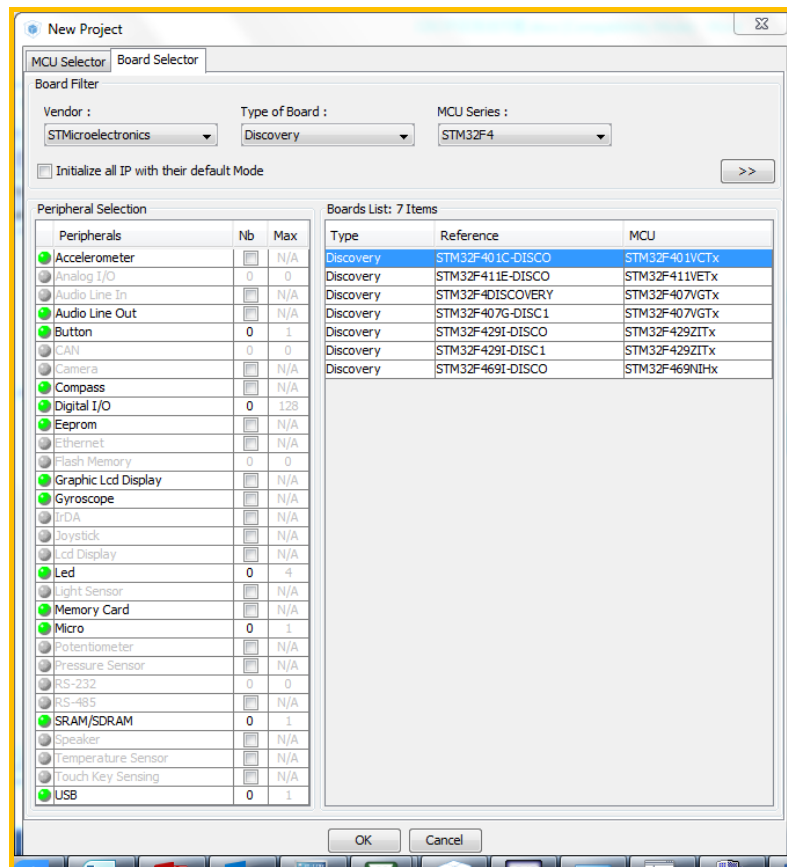
STM32CubeMX: 4.12.0

STM32CubeF4: 1.10.0

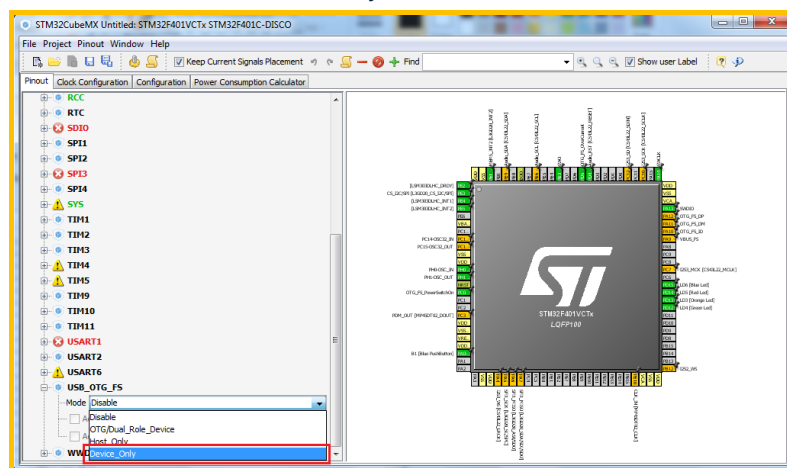
STM32CubeMX 中配置流程

实现步骤如下：

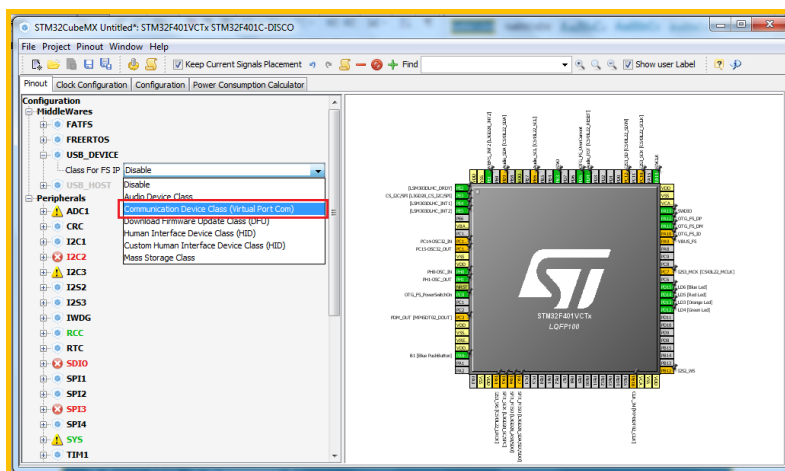
首先，打开 STM32CubeMX 软件，选择 STM32F401C-Discovery 板子。



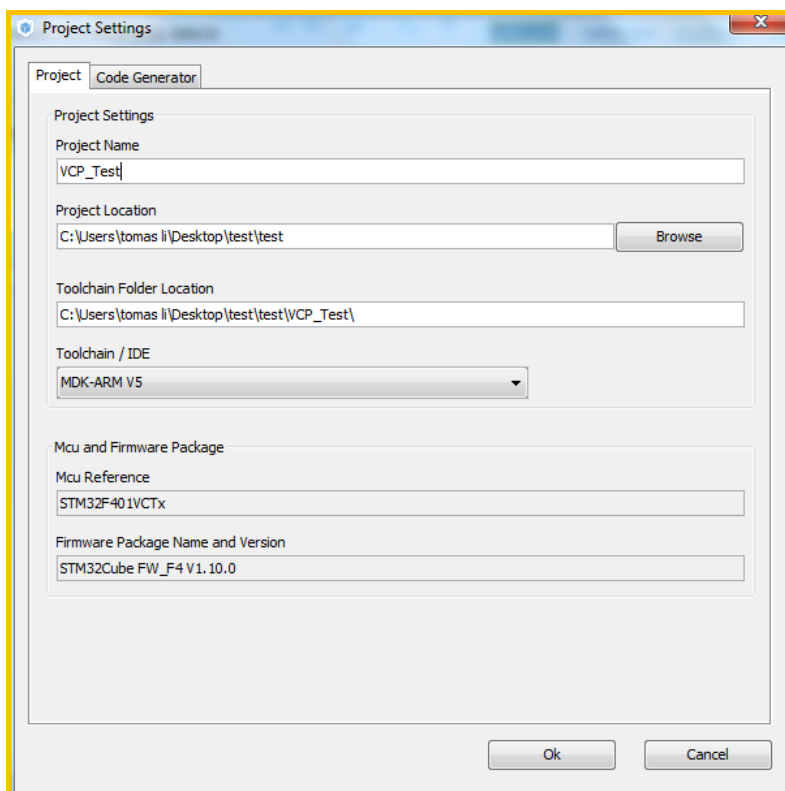
在 PinOut 列中选择 USB OTG FS 的 Device Only 选项



使能 USB 中间件的虚拟串口功能



直接生成代码，这里我们选择 MDK 的 IDE，工程名字叫 VCP_Test



在生成的工程代码中的 `usbdc_cdc_if.c` 文件中添加如下几行代码：

```
#define APP_RX_DATA_SIZE 1024
#define APP_TX_DATA_SIZE 1024
```

这个定义本来就有的，只是建议将定义的值修改为 1024，这样效果更好。这个值的默认值是 4，但是在实际的操作中发现，如果你上位机传输的数据大于 4，且并不是 4 的整数倍的时候，会出现丢数据或者数据不返回的问题。其根本原因是因为虚拟串口的数据是以数据流的方式发送出来的，接收数据不知道每次接收到的数据大小是多少，所以使用了循环队列，但是一旦你的循环队列很小，很容易出现溢出的问题。比如你设置这个值为 5，你发一个 123456 的数据给 MCU，那么 MCU 就会返回 123451 这六个数据。最后的一个数据就是溢出了，所以只能被第一个数据所代替。

```
/* USER CODE BEGIN PRIVATE_MACRO */

  USBD_CDC_LineCodingTypeDef LineCoding = {
    115200, /* baud rate */
    0x00, /* stop bits-1 */
    0x00, /* parity - none */
    0x08 /* nb. of bits 8 */
  };
```

```
uint32_t UserTxBufPtrIn = 0;
uint32_t UserTxBufPtrOut = 0;
/* USER CODE END PRIVATE_MACRO */
```

上边这部分代码是用来做串口参数设置的。下边的代码中会用到。

在 **CDC_Control_FS** 函数中的对应位置添加如下 code

```
case CDC_SET_LINE_CODING:
  pbuf[0] = (uint8_t)(LineCoding.bitrate);
  pbuf[1] = (uint8_t)(LineCoding.bitrate >> 8);
  pbuf[2] = (uint8_t)(LineCoding.bitrate >> 16);
  pbuf[3] = (uint8_t)(LineCoding.bitrate >> 24);
  pbuf[4] = LineCoding.format;
  pbuf[5] = LineCoding.paritytype;
  pbuf[6] = LineCoding.datatype;
  break;
```

在 **CDC_Receive_FS** 函数中的对应位置添加如下代码

```
/* USER CODE BEGIN 6 */
if(UserTxBufPtrIn+*Len<=APP_TX_DATA_SIZE){
    memcpy((uint8_t*)(UserTxBufferFS+UserTxBufPtrIn),Buf,*Len);
    UserTxBufPtrIn+=*Len;
}
else{
    memcpy((uint8_t*)(UserTxBufferFS),Buf,*Len);
    UserTxBufPtrIn=*Len;
    UserTxBufPtrOut=0;
}
return (USBD_OK);
/* USER CODE END 6 */
```

这是用来接收 **USB** 传输的数据，并缓存到 **UserTxBufferFS** 这个数组中。

在最后添加如下函数

```
/* USER CODE BEGIN PRIVATE_FUNCTIONS_IMPLEMENTATION */

//this function is check the application buffer whether has data
void Main_loop(void)
{
  uint32_t buffsize;

  if(UserTxBufPtrOut != UserTxBufPtrIn)
  {
```

```
if(UserTxBufPtrOut > UserTxBufPtrIn) /* Rollback */
{
    buffsize= UserTxBufPtrIn;
}
else
{
    buffsize = UserTxBufPtrIn - UserTxBufPtrOut;
}
```

```
    CDC_Transmit_FS((uint8_t*)(UserTxBufferFS+UserTxBufPtrOut),buffsize);
    UserTxBufPtrOut += buffsize;
    USB_D_CDC_ReceivePacket(hUsbDevice_0);
}
}
/* USER CODE END PRIVATE_FUNCTIONS_IMPLEMENTATION */
```

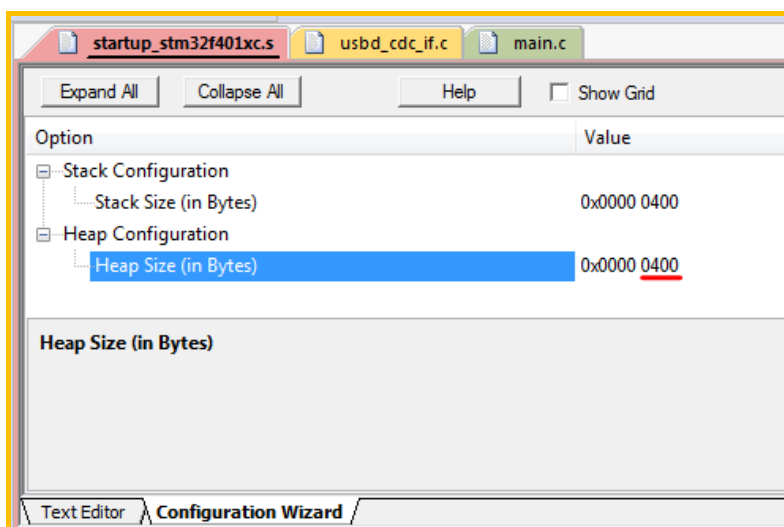
这个函数实现了一个简单的数据队列操作，一旦有数据收到，就将该数据发送到 PC 端，实现数据的环回功能。

在 main.c 的主循环中，调用 Main_loop 这个函数即可。

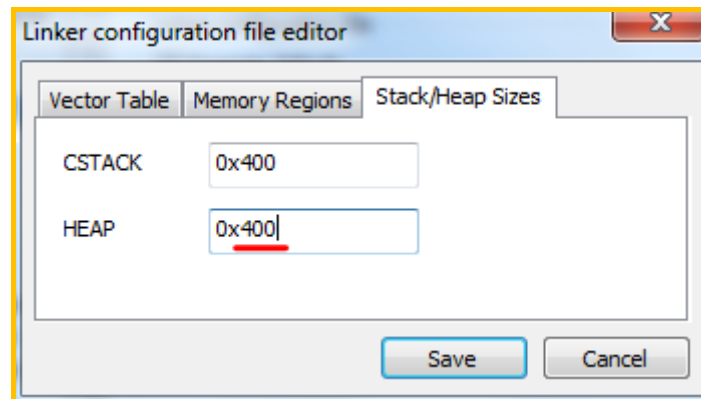
```
/* USER CODE BEGIN WHILE */
while (1)
{
    Main_loop();
}
/* USER CODE END WHILE */
```

最后修改一下堆栈的大小：

在 MDK 中修改堆栈：



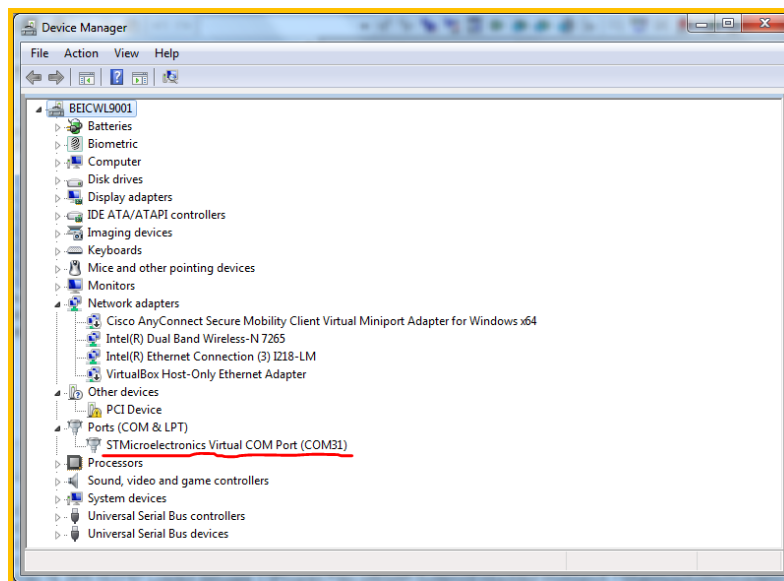
IAR 中配置堆栈大小



编译后直接下载到 MCU 中即可以运行查看结果。

注意：硬件需要一根 Micro USB 线来连接板子的 USB 端口和 PC 机。

PC 端需要已经安装好 ST 提供 VCP 驱动程序。





问题总结:

STM32CubeMX 会使 STM32 全系列中的软件移植工作变得更方便。在一些以前看起来比较复杂的程序，在 Cube 中可以很简单的完成。

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对ST 产品和/ 或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于ST 产品的最新信息。ST 产品的销售依照订单确认时的相关ST 销售条款。

买方自行负责对ST 产品的选择和使用， ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的ST 产品如有不同于此处提供的信息的规定，将导致ST 针对该产品授予的任何保证失效。

ST 和ST 徽标是ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。