请仔细阅读以下要求：

1、 在以下所有程序运行时，保持呼吸灯正常工作，以检测是否有阻塞。因此，需要先实现一个呼吸灯。呼吸灯参考代码及基本主程序结构如下：

#define ABS(\_\_N) ((\_\_N) < 0 ? -(\_\_N) : (\_\_N))

#define TOP (0x01FF)

/\*! \brief set the 16-level led gradation

\*! \param hwLevel gradation

\*! \return none

\*/

static void set\_led\_gradation(uint16\_t hwLevel)

{

static uint16\_t s\_hwCounter = 0;

if (hwLevel >= s\_hwCounter) {

LED1\_ON();

} else {

LED1\_OFF();

}

s\_hwCounter++;

s\_hwCounter &= TOP;

}

static void breath\_led(void)

{

static uint16\_t s\_hwCounter = 0;

static int16\_t s\_nGray = (TOP >> 1);

s\_hwCounter++;

if (!(s\_hwCounter & (\_BV(10)-1))) {

s\_nGray++;

if (s\_nGray == TOP) {

s\_nGray = 0;

}

}

set\_led\_gradation(ABS(s\_nGray - (TOP >> 1)));

}

int main(void)

{

system\_init();

while (1) {

breath\_led(); //!< 请不要移除该函数，用以检测代码是否存在阻塞

//! 在这里添加各test的任务函数，比如test1就写

//! Test1(); //!< test1的相关实现代码

}

}

2、 所有训练不允许使用任何形式的中断资源。所有训练仅允许通过规定好的接口函数来实现硬件外设的操作。他们分别是serial\_in和serial\_out实现串口的操作；LED1\_ON, LED1\_OFF，LED2\_ON和LED2\_OFF实现LED的控制。

3、 超级终端的基本功能：显示任意从串口发送出来的可见字符；将用户在键盘上按下的任何字符发送给串口。注意，超级终端显示的内容和用户在键盘上按下的内容无任何直接联系。

Test 1A “hello”

基本要求：

1、 按照下面的时序操作

a. 输出“hello\r\n”到控制台

b. 延时大约1s（不需要精确）

c. 重复以上步骤

说明: 输出字符串的时候不进行延时，延时的时候不进行字符串输出。

2、 呼吸灯效果必须不受影响。

3、 在主循环里面只能增加一个叫做test1()的函数。

4、 hello的输出采用最简单最直接的单一状态机，不要使用通用字符串输出的模式（指针法或者数组法输出）

Framework

static void test1(void)

{

…

}

void main(void)

{

system\_init();

while(1) {

test1();

breath\_led();

}

}

[FAQ]

1、如果常数不代表自己本身，则一定要用宏或者枚举（enum）为其赋予对应的意义。例如，用枚举替代数字作为状态码，可以为每个状态附加一个名字或者解释。不可用状态变量的算术运算来实现状态变化。

2、如果不是用于限定变量的作用范围，不要额外使用花括号，例如下面的方法就是不可取的：

switch (xxx) {

case xxx: {

…

}; break;

….

}

3、一般状态机一定要保留一个START，用于状态机的初始化，这是格式上的要求，必须保留。需要强调和注意的是START只在状态机复位后第一次运行的时候运行且仅运行一次，例如

switch (s\_chState) {

case START:

…在这里做一些初始化工作

s\_chState = XXXXX; //!< 跳转下一个状态

//break; //!< 这里的break是有意省略的

case XXXXX:

…

}

4、状态机在结束前，一定要通过宏XXXX\_RESET\_FSM() 来完成状态的复位，不能直接给状态变量赋值来实现复位，虽然行为上一致，但在逻辑上是不同的。直接给状态变量赋值，对应状态图上一个状态迁移，而不是复位。

任何情况下，不允许出现跳转到START，只允许使用XXXX\_RESET\_FSM()来达到相同的效果。

5、状态机管理自己的状态（状态变量），因此状态变量的作用范围应该限制在状态机内，一般用静态局部变量表示。