CÓDIGO A

El siguiente código utiliza la configuración por defecto de CMSIS-RTOS.

El sistema operativo se está ejecutando y se ha llamado a la función Init_Thread.

```
#include "cmsis_os2.h"
#include "stm32f4xx_hal.h"                // CMSIS RTOS header file
#include <string.h>
#include <stdlib.h>
/*----------------------------------------------------------------------------
 *      Thread 1 'Thread_Name': Sample thread
 *---------------------------------------------------------------------------*/

uint32_t z=0;

osThreadId_t tid_Thread_producer;     // thread id
osThreadId_t tid_Thread_consumer;
int Init_Thread (void);
void Producer (void *argument);       // thread function producing data
void Consumer (void *argument);       // thread function consuming data
int qsize=0;
uint8_t a=0;
uint8_t b=0;


int Init_Thread (void) {
    tid_Thread_consumer = osThreadNew(Consumer, NULL, NULL);
  if (tid_Thread_consumer == NULL) {
    return(-1);
  }

  return(0);
}

void Producer (void *argument) {
  uint32_t status;
  while (1) {
         status= osThreadFlagsSet(tid_Thread_consumer,0x0001);
         osDelay(1000);
         status= osThreadFlagsSet(tid_Thread_consumer,0x0002);
         osDelay(1000);
    }
}
void Consumer (void *argument) {
    uint8_t val=0;
    uint32_t status;
    int errors=0;
    GPIO_InitTypeDef led_ld1 = {
        .Pin = GPIO_PIN_0,
        .Mode = GPIO_MODE_OUTPUT_PP,
        .Pull = GPIO_NOPULL,
        .Speed = GPIO_SPEED_FREQ_LOW
    };
    GPIO_InitTypeDef led_ld2 = {
        .Pin = GPIO_PIN_7,
        .Mode = GPIO_MODE_OUTPUT_PP,
        .Pull = GPIO_NOPULL,
        .Speed = GPIO_SPEED_FREQ_LOW
```

```
  };
  __HAL_RCC_GPIOB_CLK_ENABLE();

  HAL_GPIO_Init(GPIOB, &led_ld1);

  HAL_GPIO_Init(GPIOB, &led_ld2);


while (1) {
  status=osThreadFlagsWait(0x3,osFlagsWaitAny,100);
      switch (status){
          case 1:
              HAL_GPIO_TogglePin(GPIOB,led_ld1.Pin);
              a=!a;
              break;
      case 2:
              HAL_GPIO_TogglePin(GPIOB,led_ld2.Pin);
              b=!b;
              break;
      case osFlagsErrorTimeout:
              if (z==10){
                tid_Thread_producer = osThreadNew(Producer, NULL, NULL);
                z++;
              }
              else z++;
          break;
      default:
              errors++;
              break;
      }


  }
}
```

VAR A88A

0000 0111 1111 0000

1010 1000 1000 1010

1010 1000 1111 1010 >

2

CÓDIGO B

El siguiente código utiliza la configuración por defecto de CMSIS-RTOS.

El sistema operativo se está ejecutando y se ha llamado a la función Init_Thread.

```
#include "cmsis_os2.h"          // CMSIS RTOS header file
#include "stm32f4xx_hal.h"
#include <string.h>
#include <stdlib.h>


void Init_Threads (void);

void Timers (void*);

GPIO_InitTypeDef led_ld1 = {
        .Pin = GPIO_PIN_0,
        .Mode = GPIO_MODE_OUTPUT_PP,
        .Pull = GPIO_NOPULL,
        .Speed = GPIO_SPEED_FREQ_LOW
    };

GPIO_InitTypeDef led_ld2 = {
        .Pin = GPIO_PIN_7,
        .Mode = GPIO_MODE_OUTPUT_PP,
        .Pull = GPIO_NOPULL,
        .Speed = GPIO_SPEED_FREQ_LOW
    };

void Timer1_Callback_1(void *arg);
void Timer1_Callback_2(void *arg);
osTimerId_t timsoft1, timsoft2;

void Init_Threads(void){
     osThreadId_t tid_Thread = osThreadNew(Timers, NULL, NULL);
}

void Timers (void* arg) {
    __HAL_RCC_GPIOB_CLK_ENABLE();
    HAL_GPIO_Init(GPIOB, &led_ld1);
    HAL_GPIO_Init(GPIOB, &led_ld2);
    HAL_GPIO_WritePin(GPIOB, led_ld1.Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, led_ld2.Pin, GPIO_PIN_RESET);

    timsoft1 = osTimerNew(Timer1_Callback_1, osTimerPeriodic, NULL, NULL);

    osTimerStart(timsoft1,1000);


  while(1){
        osDelay(1000);
  }
}
```

```
void Timer1_Callback_1(void *arg){
      static uint8_t counter=0;
            if (counter==10){
                  HAL_GPIO_TogglePin(GPIOB,led_ld1.Pin);
                  timsoft2 = osTimerNew(Timer1_Callback_2, osTimerPeriodic, NULL,
NULL);

                  osTimerStart(timsoft2, 500);
                  osTimerStop(timsoft1);
            }
            counter++;

}

void Timer1_Callback_2(void *arg){

            HAL_GPIO_TogglePin(GPIOB,led_ld2.Pin);

}
```

A los 10s enciende Led 1

y 0'5s despues enciende Pin 2 y se queda parpadeando