DISCA
DEPARTAMENT D'INFORMÀTICA
DE SISTEMES I COMPUTADORS

Programación para sistemas embebidos

Ángel Perles

r02

Contenido

- Objetivos
- Punto de partida
- Guías de estilo
- Un buen módulo
  - main() de ejemplo
  - servo.c/servo.h

Objetivos

- Conocer técnicas típicas para calidad de código

Punto de partida

- Partimos de que tenéis conocimientos básicos de C/C++
  - Revisar borrador de libro “ARM Cortex-M práctico. 1 – Introducción a los microcontroladores STM32 de S” https://aperles.blogs.upv.es/2017/01/14/arm-cortex-m-practico-1-introduccion-a-los-microcontroladores-stm32-de-st/
  - La parte de C más general
- ¿Por qué C/C++?
  - Es el lenguaje rey para sistemas embebidos
  - Ver top programming languages
    - https://www.tiobe.com/tiobe-index/
    - https://spectrum.ieee.org/top-programming-languages-2021
- Pero es muy susceptible a meter la pata
  - Semi-solución: guías de estilo

Guías de estilo

- Ejemplo destacable
  - MISRA-C
    - Busca que es
    - Las HAL de St lo cumplen
    - Aquí pondré lo de MISRA-C y la guía Barr
  - Para nosotros:
    - BARR-C:2018 Embedded C Coding Standard
    - Léelo y úsalo
    - https://barrgroup.com/Embedded-Systems/Books/Embedded-C-Coding-Standard
  - Notas mías
    - https://aperles.blogs.upv.es/2015/08/28/tips-collection-for-good-quality-code-development/

Un buen módulo

- Partimos del siguiente main() … que no me gusta del todo

```
#include "stm32l4xx_hal.h"
#include "error.h"
#include "servo.h"
#include "main.h"

/* Private variables ---------------------------------------------------------*/
/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);

int main(void)
{

   error_t error_code;

   HAL_Init();
   SystemClock_Config();

   error_code = servo_Init();
   if (error_code != Error_NoError)
   {
     error_Handler(__FILE__,__LINE__, error_code);
   }
```

Un buen módulo

- Partimos del siguiente main() … que no me gusta del todo

```
for(;;)
{
    error_code = servo_SetPosition(900); // 90 degrees
    if (error_code != Error_NoError)
    {
        error_Handler(__FILE__,__LINE__, error_code);
    }
    HAL_Delay(3000);

    // NOT VALID anymore, please check always the error codes
    servo_SetPosition(0); // 0 degrees
    HAL_Delay(3000);

    servo_SetPosition(-900); // -90 degrees
    HAL_Delay(3000);

    HAL_Delay(3000);
}
```

Un buen módulo

- Gestionar los errores. error.h/error.c

```
#ifndef ERROR_H
#define ERROR_H

/* Includes ------------------------------------------------------------------*/

/* Exported types ------------------------------------------------------------*/
typedef uint8_t error_t;

/* Exported constants --------------------------------------------------------*/
#define Error_NoError                0x00    // All worked OK
#define Error_ServoInit              0x01    // Problems initiating servo
#define Error_ServoCommand           0x02    // Problems commanding servo
#define Error_UnimplementedError     0xFF    // Not yet specified
//IMPORTANT: enumerated-based error list is safer

/* Exported macro ------------------------------------------------------------*/

/* Exported functions --------------------------------------------------------*/
void error_Handler(char *file, int line, error_t error_code); // "int" OK here
const char* error_ToStr(error_t error_code);

#endif
```

Un buen módulo

- Gestionar los errores. error.h/error.c

```
/* Private define ------------------------------------------------------------*/
struct error_list_t error_list[] = {
    {Error_NoError,             "No error."},
    {Error_ServoInit,           "Servo can't be initiated."},
    {Error_ServoCommand,        "Servo command problems."},
    {Error_UnimplementedError,  "Error not implemented."} // end of list
};

/* Private macro -------------------------------------------------------------*/
/* Private variables ---------------------------------------------------------*/
/* Private function prototypes -----------------------------------------------*/
/* Exported functions --------------------------------------------------------*/

/******************************************************************************/
/**
  * @brief  This function is executed in case of error occurrence.
  * @param  file: The file name as string.
  * @param  line: The line in file as a number.
  * @retval None
  */
void error_Handler(char *file, int line, error_t error_code)
{

  printf("ERROR in file %s on line %d and reason %s\r\n", file, line, error_ToStr(error_code));

  for(;;)
  {
  }

}
```

Un buen módulo

- El módulo para el servo servo.h/servo.c

```
/**
    @file servo.h
    @brief Servo handling
*/

#ifndef SERVO_H
#define SERVO_H

error_t servo_Init(void);
error_t servo_SetPosition(int16_t tenth_degree);

#endif
/**** End of file ***/
```

Un buen módulo

- El módulo para el servo servo.h/servo.c

```
/**
    @file servo.c
    @brief Servo handling using bl, bla

    IMPORTANT: Used resources
      TIM2
      PA5     ------> TIM2_CH1
*/

#include <stdint.h>
#include "stm32l4xx_hal.h"
#include "error.h"
#include "servo.h"


// We use variables here for calibratiuon pourposes
static int16_t TENTH_DEGREES_MIN      = -900;
static int16_t TIMMING_MIN            = 1000;
static int16_t TENTH_DEGREES_MAX      = 900;
static int16_t TIMMING_MAX            = 2000;
static int16_t TIMMING_TOTAL          = 20000;

static TIM_HandleTypeDef htim;
```

Un buen módulo

- El módulo para el servo servo.h/servo.c

```
/**
  @brief Set-ups the underlaying hardware, bla, bla ...
  @param none
  Example:
  @verbatim
    servo_Init();
  @endverbatim
 */

error_t servo_Init(void)
{

   /* GPIO Ports Clock Enable */
   __HAL_RCC_GPIOA_CLK_ENABLE();

   /* Peripheral clock enable */
   __HAL_RCC_TIM2_CLK_ENABLE();


    TIM_ClockConfigTypeDef sClockSourceConfig;
    TIM_MasterConfigTypeDef sMasterConfig;
    TIM_OC_InitTypeDef sConfigOC;

    htim.Instance = TIM2;
    htim.Init.Prescaler = SystemCoreClock/1000000;   // 1 uS
    htim.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim.Init.Period = TIMMING_TOTAL;
    htim.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    if (HAL_TIM_Base_Init(&htim) != HAL_OK)
    {
        return Error_ServoInit;
    }
```

Un buen módulo

- El módulo para el servo servo.h/servo.c

```
/*****************************************************************************
/**
  @brief Sets the position of the servo
  @param tenth_degree tenth of dreeg for the position
  Example:
  @verbatim
    servo_SetPosition(456); // the position will be 45.6 degrees
  @endverbatim
 */
error_t servo_SetPosition(int16_t tenth_degree)
{
    // pa currar hoy
    htim.Instance->CCR1 = 1000;

    return Error_NoError;
}

/*** END OF FILE ***********************************************************
```

The End

(Cajón de sastre)

- Bip, bip