UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

ELO 312
Laboratorio de Estructura de Computadores
Segundo semestre 2019

DEPARTAMENTO DE ELECTRONICA

Profesor P1: Mauricio Solís
Profesor P2: Ioannis Vourkas (coordinador)
Profesor P3: Alejandro Alviña

Laboratorio 0

Sin evaluación

Conceptos básicos de C

Resumen

En este laboratorio se familiarizaran con el lenguaje de programación C y el IDE ***Atollic trueSTUDIO*** for STM32. Se analizaran y desarrollarán programas sencillos para adquirir los conceptos básicos. Se usara directamente la tarjeta NUCLEO-L476RG. Como bibliografía para realizar el laboratorio, se sugiere el material presentado en la clase previa como también los siguientes recursos:

`http://www.cprogramming.com/tutorial/c-tutorial.html`
`https://www.tutorialspoint.com/cprogramming/`
`http://ramos.elo.utfsm.cl/~lsb/elo311/clases/apuntes_c/cbreve.pdf`
`http://ramos.elo.utfsm.cl/~lsb/elo311/clases/apuntes_c/Funciones.pdf`

`http://es.wikipedia.org/wiki/C`
`http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado`

Objetivos

Algunos objetivos para este laboratorio son:

- Estudiar algunos aspectos básicos de la programación en lenguaje C.
- Comprender las etapas en la compilación de programas en alto nivel.
- Familiarizarse con ambientes de programación IDE (Integrated Development Environment).
- Observar las instrucciones en Assembler (directivas) para crear espacios para las variables.
- Estudiar los modos de direccionamiento para tratar arreglos y estructuras de datos.
- Analizar la generación de código en Assembler.

Resultado de Aprendizaje

Desarrolla y evalúa código de programación en C, utilizando estructuras y elementos básicos del lenguaje, en el entorno de desarrollo integrado ***Atollic trueSTUDIO for STM32***.

0. Parte previa

Identificar las características principales de la tarjeta utilizada STM32.

- ➔ ¿Qué significa el número XX en la representación Nucleo-XX?
- ➔ ¿Por qué cree que el reloj de la CPU es tan bajo cuando el Raspberry PI funciona a frecuencias de GHz?
- ➔ ¿Qué arquitectura usa el procesador ARM Cortex M-4?
- ➔ Estudie el layout de su tarjeta: ¿Cuál es el rol de los resistores de 0-Ohm en la parte inferior?
- ➔ ¿Cuál es el rol de los LEDs y de los BOTONES en la tarjeta?

1. En el Laboratorio

1.1. Lenguaje C y uso de Atollic trueSTUDIO for STM32

1.1.1. Variables y Punteros

Utilizando el ambiente de trabajo de ***Atollic trueSTUDIO for STM32***, cree un nuevo proyecto y configure el proyecto cambiando las opciones que sean necesarias (lo que Ud. hará con cada nuevo proyecto) y luego compile y ejecute el programa paso a paso (modo Debug/Depuración).

Abra todas las ventanas de interés en ***trueSTUDIO*** (es decir, *Watch*, *Registers*, *Memory*, *I/O Terminal*, etc.). Con la opción Watch se puede observar el contenido de variables a medida que se ejecuta un programa. Con la ayuda del profesor, **explique el rol de todas estas ventanas del IDE**, explique **todas las sentencias de su código** y los resultados obtenidos. Fíjense en los **comentarios** en el código y trate de responder, entre otros:

- ¿Dónde están almacenadas las variables?
- ¿Cómo definimos un puntero?
- ¿Cómo le asignamos a un puntero la dirección en la memoria de una variable?
- ¿Cuál es la diferencia entre p1 y p2?
- Investigue qué hace la función sizeof() del lenguaje C.
- ¿Qué devuelve sizeof() cuando pasamos como parámetro p2?
- ¿Cómo se inicializa cte?
- Mirar e identificar operaciones básicas y cómo ellas se realizan **en Assembler**

1.1.2. Operaciones al Bit

Unas de las características más sobresalientes del lenguaje C son sus **operaciones al bit**. Existen dos tipos: corrimiento left y right (shift) y operadores lógicos (por ejemplo AND, OR, NOT, XOR).

En el mismo workspace, cree ahora un nuevo proyecto y use el código en el archivo *main1.c*. No olvide fijar este proyecto como activo. Dicho programa contiene la función **prt()** que imprime en la salida estándar la representación binaria de un número.

Compile y ejecute el código paso a paso. Estudie las salidas que se producen. Abra de nuevo todas las ventanas de interés en el IDE. Con la ayuda del profesor, explique todas las sentencias del código y los resultados de las operaciones. Fíjense en los comentarios en el código y trate de responder, entre otros:

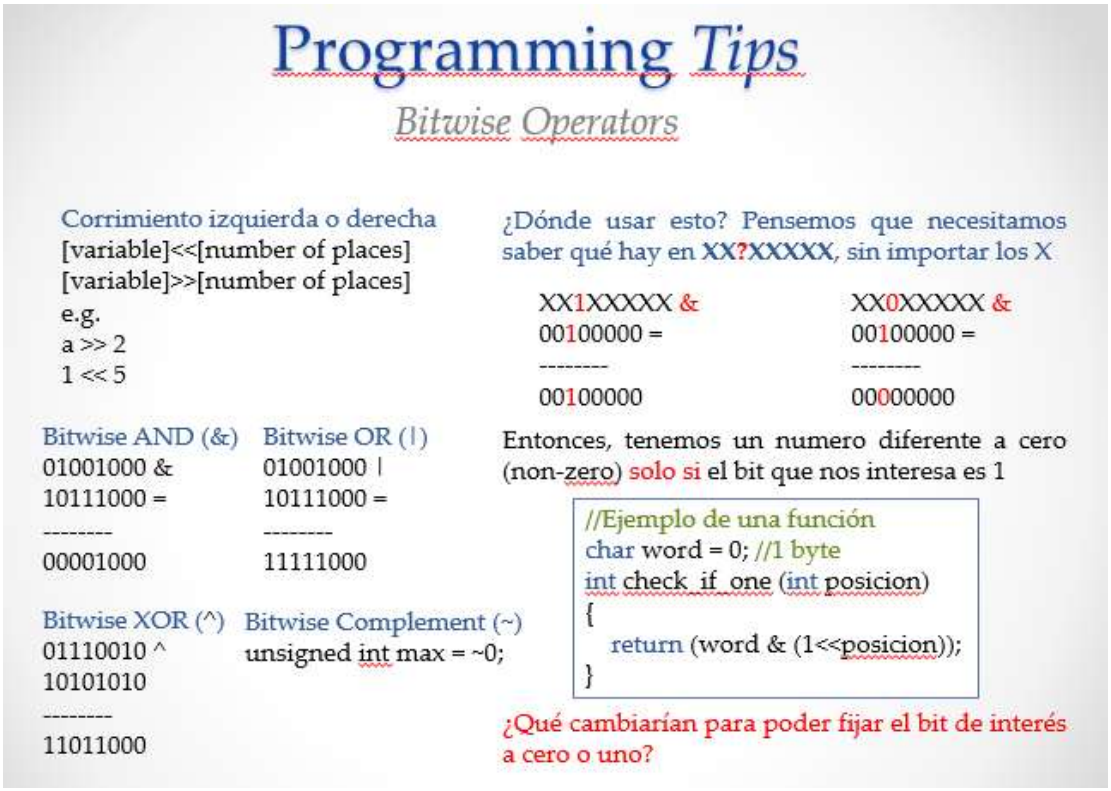
- ¿Dónde están almacenadas las variables *x*, *y*?
- ¿Dónde está almacenada la variable *j*?
- Observar la ejecución de **FOR** dentro de la función **prt** paso a paso.
- ¿Qué hace la sentencia: ***((1 < < j) & i)*** dentro del operador ternario?
- Mientras están en main(), ¿pueden visualizar en la ventana Watch la variable *j*?

Se recomienda practicar el **uso de Breakpoints** a fin de interrumpir la ejecución del programa en algunos puntos determinados, e incluso dentro de la función prt().

En el mismo archivo, con la ayuda del profesor, cree una nueva función cuyo prototipo es:

```
int check_if_one (int number, int position)
```

Dicha función examinará el estado del bit en la posición 0 dentro del número *number* y retornará 0 si es 0 y 1 si es 1. Ver figura siguiente para mayor información:



Modifique el código para llamar esta función y comprobar su funcionalidad.

Por último, con la ayuda del profesor, aprenda a ordenar su código. Para este proyecto, cree un *header file lab0_functions.h* que tendrá la declaración de todas las funciones. Por lo mismo, cree un archivo *lab0_functions.c* donde será su implementación. Debe incluir el nuevo header file a su código y agregar el archivo *lab0_functions.c* a los source files de su proyecto. Compruebe de nuevo la funcionalidad.