Universidad Autónoma de Baja California

Facultad de Ciencias Químicas e Ingeniería



**MICROCONTROLADORES**

**Práctica No. 4**

**Programación del uC en lenguaje C + ASM**

**Docente:** Castro Gonzalez Ricardo

**Alumno:** Gómez Cárdenas Emmanuel Alberto

**Matricula:** 01261509

# Objetivo:

Mediante esta práctica el alumno aprenderá el uso básico de la programación en lenguaje C + ASM con las herramientas AVR Studio y WinAVR. Para ello el alumno implementará los porcedimientos comunes para encender y apagar un LED a base de retardos en ensamblador

# Material:

* **Computadora presonal**

# Teoría:

* **Programación en lenguaje C en microcontroladores**

La programación en lenguaje C en microcontroladores AVR utilizando el compilador GCC (avr-gcc) implica la interacción entre código C y ensamblador. Para poder hacer esto posible se creó la convención de llamadas a funciones, ya sea desde C a ASM o viceversa.

* **Convención de llamadas a funciones en C en gcc para AVR (avr-gcc)**
* Una función en C debe estar definida como externa para que pueda ser vista por el ensamblador
  + **.extern my\_C\_function**
* Una rutina en ensamblador debe ser declarada como global para que pueda ser vista por el compilador C
  + **.global my\_assembly\_function**
* Un archivo C que pretenda llamar código ensamblador deberá tener la función prototipo declarada como externa
  + **extern unsigned char my\_assembly\_function(unsigned char, unsigned int)**

**Uso de registros**

* **Call saved:** Llamar estas funciones en C deja los registros sin modificar. Una rutina de ensamblador llamada desde C que use estos registros, deberá guardar y restaurar los contenidos de los registros que use.
* **Call used:** Significa que los registros son disponibles para cualquier uso. Un código ensamblador que llame una función en C deberá guardar cualquiera de los registros que use ya que el código compilado no los guardará.

| **Registro** | **Descripción** | **Ensamblador llamado desde C** | **Ensamblador que llama a C** |
| --- | --- | --- | --- |
| **r0** | Temporal | Guardar y restaurar si es usado | Guardar y restaurar si es usado |
| **r1** | Siempre es 0 | Debe ser borrado antes de retornar función | Debe ser borrado antes de retornar función |
| **r2-r17** | "Cal-saved” | Guardar y restaurar si es usado | Puede usarse libremente |
| **r28** | "Cal-saved” | Guardar y restaurar si es usado | Puede usarse libremente |
| **r29** | "Cal-saved” | Guardar y restaurar si es usado | Puede usarse libremente |
| **r18-r27** | "Call-used” | Puede usarse libremente | Guardar y restaurar si es usado |
| **r0** | "Call-used” | Puede usarse libremente | Guardar y restaurar si es usado |
| **r31** | "Call-used” | Puede usarse libremente | Guardar y restaurar si es usado |

**Paso de Parámetros**

| **Registro** | **r19** | **r18** | **r21** | **r20** | **r23** | **r22** | **r25** | **r24** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Order de los bytes** | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |

# Conclusiones y comentarios:

La combinación de lenguaje C y ensamblador en la programación de microcontroladores AVR ofrece una amplia gama de posibilidades para desarrolladores, y el conocimiento de las convenciones de llamadas a funciones es esencial para una comunicación efectiva entre ambas partes del código.