

## Microcontroladores

# Laboratorio 2

### LED parpadeante

**Objetivo general:** Mediante la realización de un programa en ASSEMBLER, lograr que un LED de la placa de desarrollo utilizada en el curso se prenda de forma intermitente a una frecuencia de 2Hz.

**Materiales:** Placa de desarrollo ATmega328P Xplained Mini, módulo multifunción para Arduino, cable micro usb y herramienta de desarrollo Microchip Studio.

## Introducción

El ciclo máquina o de ejecución es un elemento muy importante a la hora programar microcontroladores ya que nos permiten determinar el tiempo exacto que demora en ejecutarse una instrucción (algunas instrucciones pueden llevar más de un ciclo máquina). Dependiendo del tipo de microcontrolador que se está utilizando el ciclo máquina no necesariamente es igual al ciclo de reloj del oscilador que se tiene en el sistema, puede ser mayor e incluso menor.

## Implementación

### Parte 1

**Analizado el HardWare:** Examine el HW disponible para la práctica y determine cuales son los puertos del microcontrolador ATmega328P que va a utilizar. Utilizando el esquemático del módulo multifunción determine el valor lógico necesario para prender el/los leds a utilizar (esquemático disponible en la webasignatura). Utilizando el manual [1] determine la frecuencia del reloj a la cual está trabajando el microcontrolador.

**Pseudo código o diagrama de bloques:** Realice un pseudo código o un diagrama de bloques que cumpla con el objetivo general, tome en cuenta los registros que debe configurar y observar, se recomienda ver el manual de microcontrolador ATmega328P [2] (disponible en la webasignatura) para determinar los valores necesarios para los registros involucrados. Tenga presente cuánto tiempo lleva la ejecución de un ciclo máquina y determine la cantidad necesaria de instrucción a realizar para llegar el tiempo deseado.

**Assembler:** Implemente un código en assembler considerando el pseudo código o diagrama de bloques de la sección anterior, se recomienda estudiar las instrucciones DEC, BRNE, RJMP, IN, SBRC y CPSE del AVR Instruction Set Manual [3], presten especial atención a cuantos ciclos máquina demora cada instrucción.

**Compilando y grabando:** Siguen los pasos de la sección 2.3.1-2 de [1].

**Breaking points:** Agregue un break point por cada línea de código que tenga, compile, programe y vaya ejecutando línea por línea, ¿Qué sucede con el valor en el registro utilizado?.

### Microcontroladores

**I/O Viewer:** Abra el I/O de Microchip Studio , vuelva a ejecutar instrucción por instrucción y observe el comportamiento del I/O Viewer

#### Parte 2

**Haciendo el auto increíble:** Modifique el código de Assembler para con los 4 leds del módulo, emular las luces del auto increíble (<https://www.youtube.com/watch?v=R5z7cGNwyek>)

**Haciendo Ruido:** Utilizando el esquemático del módulo arduino determine el pin que controla el Twitter disponible. Aprovechando el código del parte anterior e implemente una alarma sonora.

#### Parte 3

**Alto nivel:** Realice la parte 2 nuevamente, pero esta vez programando en alto nivel.

**Compilados generados:** Observe el código assembler generado por el compilador y compare contra el que usted realizó.

### Referencias

[1] Manual "ATmega328P Xplained Mini", pdf disponible en la webasignatura (ATmega328P-Xplained-Mini-UG-DS50002659B.pdf).

[2] Manual "ATmega328P", pdf disponible en la webasignatura (Atmel-7810-Automotive-Microcontrollers-ATmega328P\_Datasheet.pdf).

[3] Manual "del AVR Instruction Set Manual", pdf disponible en la webasignatura (atmel-0856-avr-instruction-set-manual.pdf).