



DEPARTAMENTO DE TECNOLOGÍA ELECTRÓNICA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Programación del microcontrolador ATMega328P

1. Introducción y objetivos

- Utilizar el entorno de programación y depuración de microcontroladores de ATMEL llamado AtmelStudio que ya se presentó en la práctica anterior.
- Simular y depurar programas escritos en lenguaje ensamblador para el microcontrolador ATMEGA328P que usen puertos de E/S.
- Programar realmente el microcontrolador ATMEGA328P que se encuentra en una placa de desarrollo llamada Arduino1 Duemilanove (Figura 1a). Para ello se utiliza la plataforma de depuración/ programación AVR-DRAGON (Figura 1b) también del fabricante ATMEL.

Las placas Arduino están diseñadas para ser programadas en un lenguaje de programación propio, transfiriéndose los programas a través de su puerto USB. En esta sesión de laboratorio no se utilizarán estas características, es decir, se programarán directamente en ensamblador. Por ello, se han realizado modificaciones en dichas placas. Aunque no es relevante para esta sesión de laboratorio, se puede consultar toda la información adicional sobre esta placa en: <http://www.arduino.cc>

Atmel-Studio puede descargarse gratuitamente desde las páginas de Microchip en: <https://www.microchip.com/mplab/avr-support/avr-and-sam-downloads-archive>

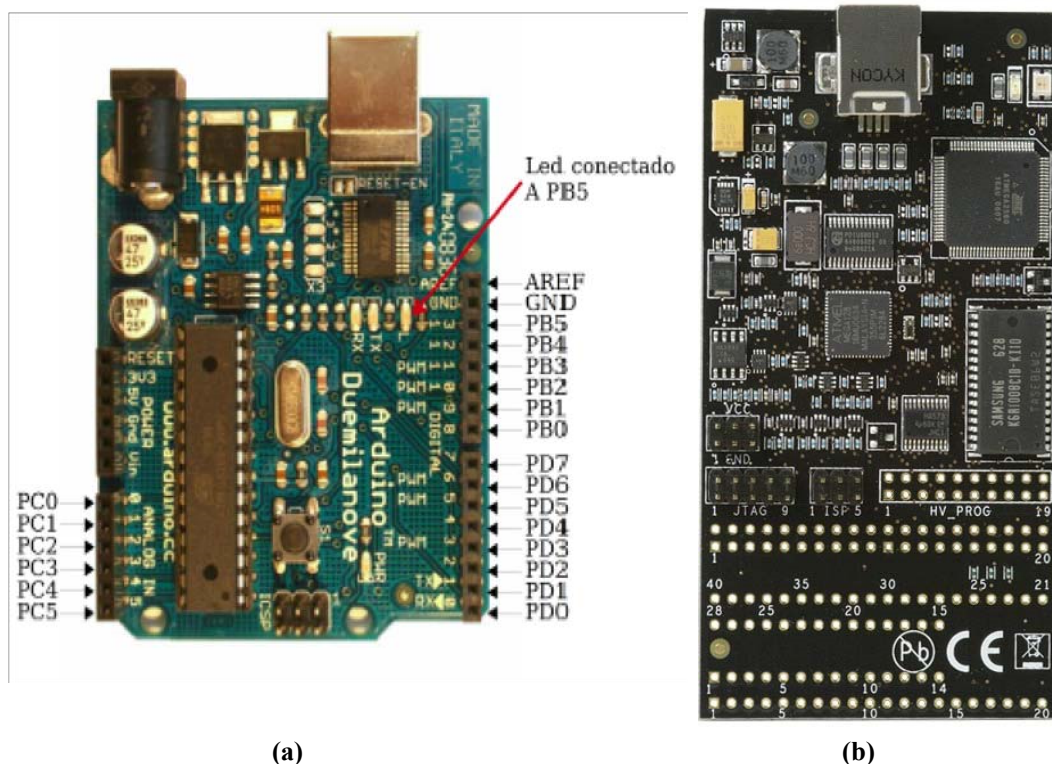


Figura 1: (a) Placa de desarrollo Arduino. (b) Programador/depurador AVR-Dragon.

Durante la sesión de laboratorio se debe disponer de los ficheros indicados en la tabla 1.

Nombre del fichero	Contenido
contador_0_10.asm	Programa contador de 0 a 10. A completar por el alumno.
contador_0_1000.asm	Programa contador de 0 a 1000. A completar por el alumno.
conmutadores.asm	Programa para controlar dos pulsadores y un led. A completar por el alumno.

Tabla 1: Ficheros necesarios para realizar la práctica

2. Estudio teórico

Se van a utilizar tres programas (Tabla 1) en lenguaje ensamblador. Estos están disponibles (sin completar) en ficheros que se pueden encontrar en la carpeta correspondiente a esta práctica en la web de la asignatura.

A continuación se presentan algunos detalles de los mismos:

1. Programa CONTADOR de 0 a 10:

Se trata de realizar un programa **contador_0_10.asm** en ensamblador que cuente de 0 a 10 utilizando un registro del microcontrolador. Cuando termine la cuenta el programa debe invertir el valor del pin PC0 y volver a empezar, es decir, volverá a contar de 0 a 10 e invertirá de nuevo el pin PC0. Así indefinidamente.

Para realizar el programa correctamente se debe configurar el puerto C como salida, para ello se propone comenzar el programa utilizando el siguiente fragmento de código (archivo contador_0_10.asm):

```

; Programa contador de 0 a 10
; Cada vez que se pase por 10 se debe invertir el pin PC0

start:
    sbi ddrC,0 ; Configura el pin PC0 como salida

```

2. Programa CONTADOR de 0 a 1000:

Se trata de realizar un programa **contador_0_1000.asm** similar al anterior pero que cuente de 0 a 1000 antes de invertir el PC0. Ha de tenerse en cuenta que los registros del microcontrolador son de 8 bits y, por tanto, la cuenta máxima que se puede realizar con un único registro es de 0 a 255.

3. Programa para controlar los CONMUTADORES:

Se trata de realizar un programa **conmutadores.asm** que permita manejar los puertos de entrada/salida para poder activar/apagar cierto led cuando se pulsa un conmutador.

En el esquema de la Figura 2 aparecen dos conmutadores (PC4 y PC5 - línea discontinua) y dos leds (PC2 y PC3 - línea continua). Lo que se pide es que opere con PC2, PC4 y PC5 de modo que:

- si se pulsa el conmutador conectado a PC4 debe encenderse el led conectado a PC2 y mantenerse encendido hasta que se pulse el conmutador conectado a PC5

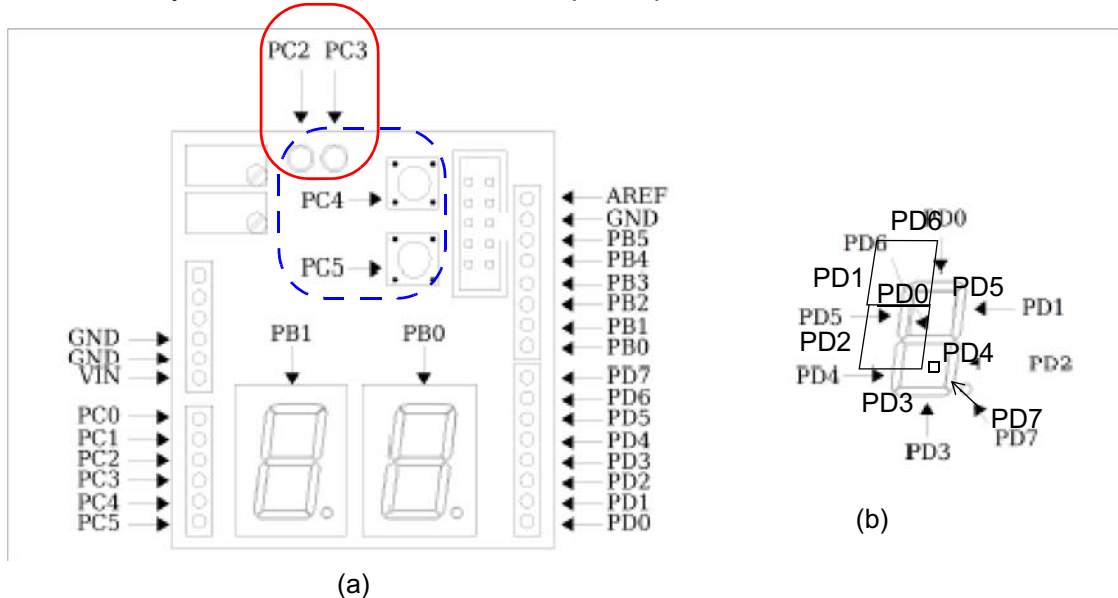


Figura 2: (a) Placa de expansión E/S para Arduino. (b) Detalle de conexión de los segmentos del display a los pines de E/S.

La Tabla 2 muestra los puertos y los bits asociados a los componentes así como la configuración necesaria para que operen correctamente. Téngalo en cuenta al realizar el programa.

Pin	Puerto	Componente	Configuración	Funcionamiento
PC2	C	Led	salida (DDRC2 = 1)	PINC2=0 apagado PINC2=1 encendido
PC3			salida (DDRC3 = 1)	PINC3=0 apagado PINC3=1 encendido
PC4		Conmutador	entrada (DDRC4 = 0)	PINC4=1 no pulsado PINC4=0 pulsado
PC5			entrada (DDRC5 = 0)	PINC5=1 no pulsado PINC5=0 pulsado

Tabla 2: Configuración del puerto de entrada/salida PORTC y funcionamiento de los leds y conmutadores.

3. Estudio experimental

El estudio experimental consistirá en **completar** los programas indicados en la Tabla 1 y **comprobar** su funcionamiento.

En primer lugar (apartado 3.1) usaremos el simulador para completar y probar el programa contador_0_10.asm. Después (apartado 3.2) simularemos y programaremos el microcontrolador con el programa contador_0_1000 y observaremos su funcionamiento sobre elementos de la placa, para lo que utilizaremos el osciloscopio y uno de los diodos. Finalmente (apartado 3.3) programaremos el microcontrolador con el programa conmutadores.asm para controlar los conmutadores y leds señalados en la Figura 2a.

En la guía de uso del programa Atmel-Studio que encontrará en la web junto con este enunciado, puede recordar aspectos esenciales como son la creación de proyectos o la compilación y depuración de programas.

3.1 Contador de 0 a 10

Complete el programa **contador_0_10.asm** y compruebe que su programa opera correctamente.

Para comprobar el funcionamiento del programa **se debe realizar la ejecución paso a paso observando cómo cambian los valores de los registros y el PC0. Hay que desplegar los registros en la ventana del procesador y el puerto C en la ventana de E/S para visualizar los registros del puerto.**

3.2 Contador de 0 a 1000

Complete el programa **contador_0_1000.asm** y compruebe que su programa opera correctamente.

- a. Simule el programa y cárguelo en la placa Arduino, para ello siga los pasos indicados en el apartado *Programación de la placa* de la guía de uso del programa Atmel Studio 7.0.

- b. Dado que el PC0 está conmutando su valor continuamente podemos asegurar que se está generando una señal cuadrada. Mida con el osciloscopio la frecuencia de dicha onda.
- c. Modifique el programa para que active y desactive el pin PC2 en lugar del pin PC0.
- d. Compruebe que fijando el valor de cuenta máximo aproximadamente a 20000 el PC2 se enciende alrededor de 5 veces por segundo.

3.3 Conmutadores

Cree un nuevo proyecto utilizando el código suministrado en el fichero **conmutadores.asm** y complete el programa comprobando que opera correctamente.

Debe conmutar manualmente los pines PC4 y PC5 desde el simulador. Esto se consigue desplegando el puerto C y pulsando el botón del ratón sobre el cuadro que representa el bit correspondiente. Cuando el cuadro está relleno de color negro significara que el bit está a 1, si está en blanco es 0.

Una vez comprobado en el simulador el correcto funcionamiento, repita los pasos realizados anteriormente para programar el microcontrolador con este nuevo programa. Compruebe que funciona correctamente pulsando los conmutadores.

3.4 Apartado opcional

Escriba un programa en ensamblador que permita encender y apagar el led conectado al pin PC2, utilizando solo el pulsador PC4.

Simule el programa, compruebe su correcto funcionamiento y posteriormente, programe el microcontrolador.