

# **Diseño de plataforma de trabajo para microcontroladores Atmel con visualización de salidas en tercer estado (tri-state) para probar su arquitectura con elementos de entrada y salida**

Marlon Contreras Urgiles, Arturo Ayala Rocafuerte, Carlos Valdivieso  
Facultad de Ingeniería en Electricidad y Computación  
Escuela Superior Politécnica del Litoral (ESPOL)  
Campus Gustavo Galindo, Km 30.5 Vía Perimetral  
Apartado 09-01-5963. Guayaquil. Ecuador  
mgcontrerasur@hotmail.com, arthur\_niceboy@hotmail.com, cvaldiv@fiec.espol.edu.ec

## **Resumen**

*El presente proyecto tiene como finalidad la elaboración e implementación de una plataforma de trabajo con el KIT AVR Butterfly de Atmel, para que en ella podamos montar, probar y corregir una amplia variedad de ejercicios que cubran todas o la mayoría de las capacidades que poseen los microcontroladores de Atmel, en este caso el Atmega169 que es el microcontrolador en el cual se sustenta el AVR Butterfly. Nuestro proyecto se basa en los Puerto Entrada – Salida y Tri-state del Atmega169 por lo que se desarrollaron cinco ejercicios que muestran como podemos configurar y utilizar de diversas formas las capacidades de los Puertos I/O que posee Atmel con sus microcontroladores AVR. Además en los primeros capítulos realizamos una revisión de dos de las más importantes empresas fabricantes de microcontroladores como son Intel y Microchip, en lo concerniente a la arquitectura y manejo de sus puertos de Entrada – Salida. Posteriormente se presenta una completa guía de características y programación del Kit AVR Butterfly, así como también el manejo y utilización de las herramientas de software proporcionadas por Atmel en esta caso el IDE AVR Studio 4 y para la simulación de la circuitería electrónica y microcontroladores en software independiente de simulación PROTEUS.*

**Palabras Claves:** AVR Butterfly, Atmega169, Tri-state, AVR Studio 4, Proteus.

## **Abstract**

*The aim of the present project is to develop and implement a work platform with the AVR Butterfly kit of Atmel, allowing us to setup, test and debug a wide variety of exercises covering all or most of the capacities of Atmel microcontrollers, in this case Atmega169, the microcontroller on which the AVR Butterfly is built. Our project is based on the Input/Output and Tri-State Port of the Atmega169, for which five exercises were developed which show how we can configure and utilize in various ways the capacities of the I/O ports that Atmel offers with its AVR microcontrollers. Moreover, in the first chapters we carry out an analysis of two of the most important companies that manufacture microcontrollers, Intel and Microchip, with regard to the architecture and handling of their Input/Output ports, Then we present a complete guide to the characteristics and programming of the AVR Butterfly Kit, as well as the handling and correct use of the software supplied by Atmel, specifically the AVR Studio 4 IDE, and instructions for the simulation of the electronic circuitry and microcontrollers in PROTEUS independent simulation software.*

**Keywords:** AVR Butterfly, Atmega169, Tri-state, AVR Studio 4, Proteus.

## 1. Introducción

Una parte fundamental que caracteriza a los microcontroladores son los Puertos de Entrada – Salida que poseen los mismos, ya que esta es una de las características que los diferencian de los microprocesadores que para utilizarlos deben implementar circuitería adicional que comunique su bus de datos y control a dichos puertos externos. Además la gran importancia que tiene la característica de tener un estado en alta impedancia de los puertos de entrada y salida que permite que la comunicación entre diversos dispositivos y el microcontrolador sea más eficiente.

Durante el desarrollo de los microcontroladores esta característica se a mejorado desde los primeros modelos de Intel como son la serie MCS-51, que poseía una arquitectura básica y simple de manejo de Puertos de Entrada – Salida, con únicamente un Latch para manejar un pin de un puerto, hasta los modernos microcontroladores de Microchip y Atmel que poseen de dos a tres registros independientes para configurar dichos puertos.

Además que no solamente los Puertos de Entrada – Salida sirven para ese fin, sino que también poseen diversas funciones multiplexadas con la función principal de Puerto I/O lo que les da mayor importancia y funcionalidad además esta característica varía de fabricante y modelo brindándonos una amplia gama de posibilidades para realizar nuestros diseños y proyectos.

## 2. Software utilizado

Para desarrollar el proyecto hicimos uso de dos tipos de software: AVR Studio 4, cuyo fin es la programación del ATmega169 y Proteus que nos servirá para la simulación completa del proyecto.

### 2.1. AVR Studio 4

AVR Studio 4 es un Entorno de Desarrollo Integrado (IDE) para escribir y depurar aplicaciones AVR ya sea en lenguaje ASM o en lenguaje C.

En este software podemos realizar el diseño, desarrollo y depuración de nuestro código que posteriormente descargaremos al microcontrolador correspondiente.

Además posee arquitectura modular que permite interactuar con software de otros fabricantes. Está disponible para descargarlo desde la página web del autor.

AVR Studio 4 proporciona herramientas para la administración de proyectos, edición de archivo fuente, simulación del chip e interfaz para emulación In-circuit para la poderosa familia RISC de microcontroladores AVR de 8 bits. [1]

## 2.2. Proteus

Proteus es una aplicación CAD (Diseño Asistido por Computador), compuesta de tres módulos:

1. ISIS (Intelligent Schematic Input System): es el módulo de captura de esquemas.
2. VSM (Virtual System Modelling): es el módulo de simulación, incluyendo PROSPICE.
3. ARES (Advanced Routing Modelling): es el módulo para la realización de circuitos impresos (PCB).

La herramienta de Proteus que se a usado en este proyecto se denomina ISIS, esta aplicación nos permite colocar en el área de trabajo todos los elementos que conformaran nuestro circuito, ya que posee una extensa librería de la mayoría de elementos que se utilizan para la creación de un circuito electrónico y eléctrico además de que esta librería esta en constante actualización, este circuito posteriormente lo podremos emular para poder así comprobar el correcto funcionamiento del mismo y poder hacer así las correcciones o ajustes necesarios, la herramienta ARES de PROTEUS no lo hemos usado en este trabajo ya que la implementación se ha realizado sobre un Protoboard y no es necesaria el PCB de cada ejercicio. [2]

## 3. Hardware utilizado

Se detallan a continuación el hardware que sea utilizado para el desarrollo de la plataforma de trabajo para microcontroladores Atmel.

### 3.1. Kit AVR Butterfly

El presente proyecto se trabajó con el kit de desarrollo, entrenamiento y aprendizaje de Atmel “AVR Butterfly” el cual provee de una excelente conjunto de herramientas y facilidades para que el estudiante aprenda y evalúe el potencial que tienen los microcontroladores de Atmel.

Los siguientes recursos están disponibles en el Kit AVR Butterfly:

- Microcontrolador ATmega169V (en encapsulado tipo MLF).
- Pantalla tipo vidrio LCD de 120 segmentos, para demostrar las capacidades del controlador del LCD incluido dentro del ATmega169.
- Joystick de cinco direcciones, incluida la presión en el centro.
- Altavoz piezoeléctrico, para reproducir sonidos.
- Cristal de 32 KHz para el RTC.
- Memoria DataFlash de 4 Mbit, para almacenar datos.
- Convertidor de nivel RS-232 e interfaz USART, para comunicarse con unidades fuera del Kit sin la necesidad de hardware adicional.
- Termistor de Coeficiente de Temperatura Negativo (NTC), para sensar y medir temperatura.
- Acceso externo al canal 1 del ADC del ATmega169, para lectura de voltaje en el rango de 0 a 5 V.

- Emulación JTAG, para depuración.
- Interfaz USI, interfaz adicional de comunicación.
- Terminales externas con conectores tipo Header, para el acceso a periféricos.
- Batería de 3 V tipo botón (600mAh), para proveer de energía y permitir el funcionamiento del AVR Butterfly.
- Bootloader, para programación mediante la PC sin hardware especial. [3] [1]



Figura 1. Kit AVR Butterfly [4]

### 3.2. Plataforma de trabajo

Para montar la plataforma de trabajo se implemento la misma sobre una Protoboard de tamaño medio sobre la cual se monto dos porta pilas de tamaño doble AA y obviamente el Kit AVR Butterfly.

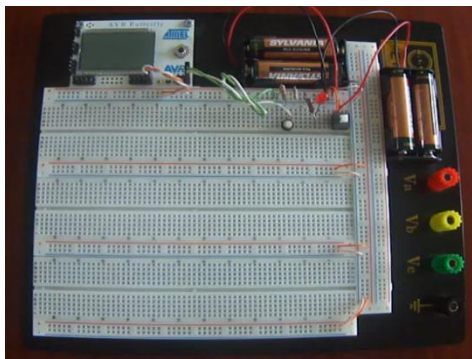


Figura 2. Plataforma implementada

Además de todo lo mostrado es necesario el cable de comunicación serial entre el Kit AVR Butterfly y la PC, mediante el cual cargaremos las aplicaciones desarrolladas. Debido a que los computadores actuales no poseen en su mayoría el puerto de comunicación serial se debió adquirir un cable USB-Serial que sirve de igual manera.

Para ejercicio se utilizan diferentes componentes y como esta es una plataforma multifuncional esta es su característica principal ya que podemos montar en ella lo que imaginemos en nuestros diseños.

## 4. Ejercicios Desarrollados

A continuación mostraremos tres de los cinco ejercicios desarrollados en esta tesina.

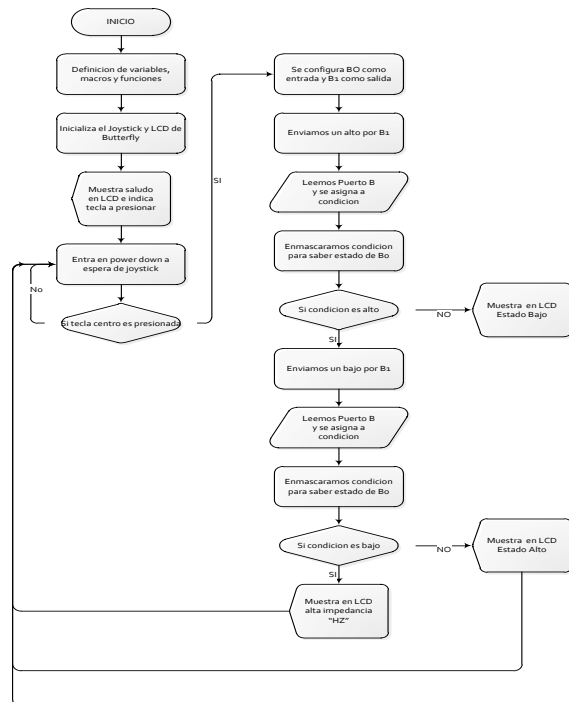
### 4.1. Punta Lógica

La Punta Lógica es una herramienta de mucha utilidad para cuando se trabaja con circuitos digitales y se quiere hacer un seguimiento de cómo están las señales en el mismo, así podemos saber si en algún punto del circuito se encuentra un estado lógico alto, bajo o en alta impedancia. El circuito trabaja de mediante dos pines los cuales están conectados entre sí por medio de una resistencia de 47 Kohm, el extremo del pin configurado como entrada es la que debemos utilizar como la punta lógica la cual está definida, luego de colocar la punta en lugar a testear presionamos en centro del joystick, para luego visualizar en el LCD el estado presente en la punta. Así cuando queramos realizar una prueba siempre lo haremos presionando el centro del joystick.



Figura 3. Diagrama de bloques Punta Lógica

#### 4.1.1. Diagrama de flujo de la Punta Lógica



### 4.1.2. Algoritmo de la Punta Lógica.

1. Definimos variables, funciones y macros, inicializamos joystick y LCD.
2. Mostramos saludo por LCD e indicaciones de teclado.
3. Ponemos al microcontrolador en sleep power down en donde esperara hasta que se presione una tecla del joystick, que causara una interrupción y lo sacara de ese estado.
4. Cuando presiones una tecla del joystick, la evaluaremos para saber si la que fue presionada fue la tecla del centro que es la que inicia la prueba caso contrario seguirá esperando.
5. Una vez presionada la tecla del centro, comenzamos con la evaluación del estado en la punta lógica.
6. Configuramos los pines B0 como entrada y B1 como salida, estos pines están conectados entre si con una resistencia de 47 k aproximadamente, B0 sera la que sea la punta lógica ya que es la entrada, ahora colocamos un alto en B1 y leemos el estado presente en B0 si en B0 esta cero sabremos que la punta detecto un bajo si no es así, procedemos a enviar por B1 uno un bajo, y nuevamente leemos el estado de B0, si en B0 esta presente un uno confirmaríamos el estado alto en la punta, pero si esta un cero, sabremos que en la punta existe un estado de alta impedancia HZ, luego de haber analizado el ejercicio queda listo para otra prueba.

En este este ejercicio se usa las librerías para manejar LCD que vienen con el AVR Butterfly.

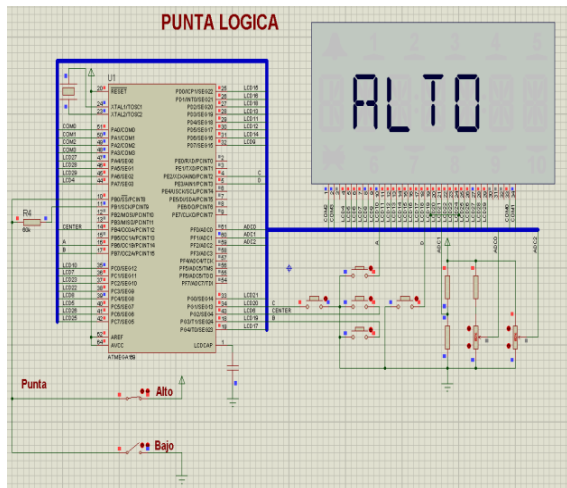


Figura 4. Esquemático de la Punta Lógica

### 4.2. Juego de Reacción

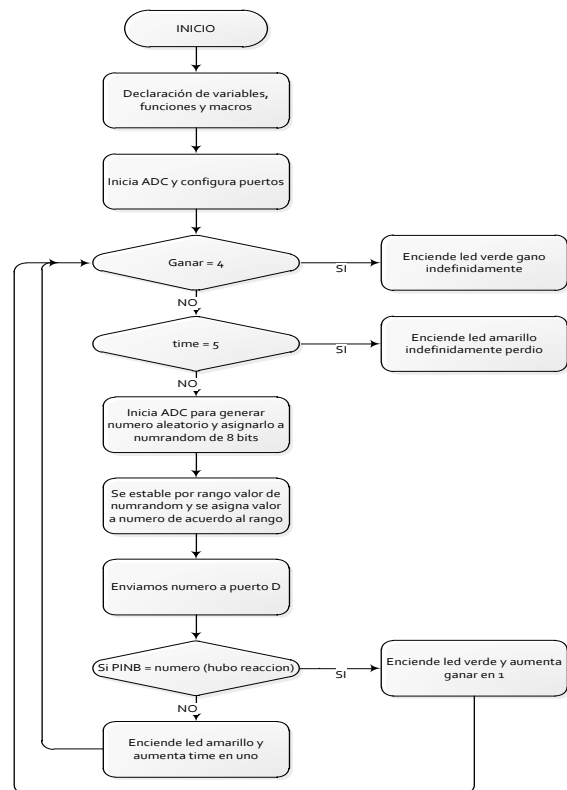
En este ejercicio podemos ver cómo utilizar tanto entradas como salidas para implementar un divertido juego.

En este ejercicio tenemos ocho Leds los cuales al iniciar el juego ya sea enciendo la plataforma o reseteando el Butterfly encenderá un Led aleatorio y deberemos antes de un tiempo presionar el Push botton correspondiente al Led que encendió si no lo hacemos se encenderá un Led Amarillo indicando que hemos fallado, si lo hacemos bien encenderá un Led verde indicando que lo hemos hecho bien, se tienen 5 oportunidades caso contrario se pierde el juego permaneciendo el led amarillo encendido, se gana acertando 4 veces y en ese caso permanecerá encendido el led verde.



Figura 5. Diagrama de bloques Juego de Reacción

#### 4.2.1. Diagrama de flujo de Juego de Reacción



#### 4.2.2. Algoritmo de Juego de Reacción.

1. Declaramos Variables, funciones y macros a utilizar.
2. Ya dentro de “main” iniciamos el ADC y configuramos el Puerto B como entrada, el Puerto D y E como salidas.
3. Lo primero que hace la aplicación es preguntar por el estado de las variable ganar y time que nos indicara si el juego continua o a terminado, ya que si se ha ganado o acertado 4 veces la variable ganar hará que el juego termine con victoria encendiendo un led verde indefinidamente hasta presionar reset para jugar nuevamente, o caso contrario si la variable time llega a 5 antes que ganar el juego terminara con perdida encendiendo un led amarillo indefinidamente hasta darle un reset para jugar nuevamente.
4. Si las variables no están en su tope el juego sigue, y lo hace primero encontrando un número aleatorio para comenzar a jugar, lo hace por medio del ADC que coge una muestra de voltaje externo en el pin F1, con esta muestra podemos asignar un valor a la variable numero que será la que se envíe al puerto D encendiendo un led.
5. El usuario deberá presionar el Push botton correspondiente al led encendido en un periodo de tiempo establecido para ganar caso contrario será un error.
6. Este proceso continua hasta que una de las variables ganar en caso de acertar 4 veces o perder cuando se ha errado 5 veces alcance estos valores y veamos que hemos perdido o ganado el juego.

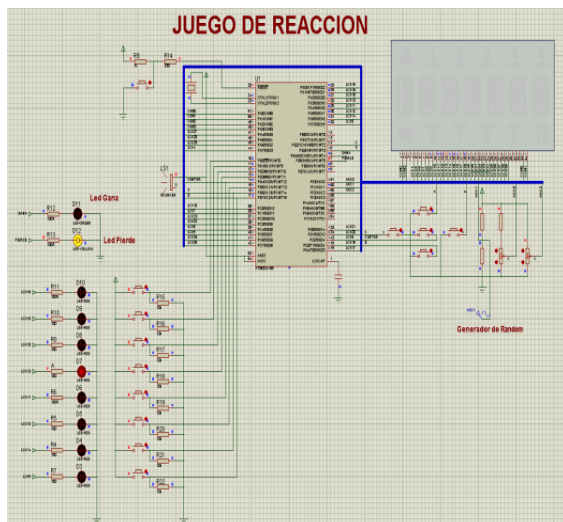


Figura 6. Esquemático del Juego de Reacción

#### 4.3. Menú de entradas, salidas y alta impedancia.

En este ejercicio de realizo un pequeño menú en LCD y por medio del joystick podemos elegir una de las opciones que son:

Joystick hacia arriba configuramos los pines B0 y B1 como salidas y podemos encender y apagar dos leds conectados a los mismos cada vez que presionamos hacia arriba.

Joystick hacia la derecha configuramos los mismos pines B0 y B1 ahora como entradas y se conectan a dos interruptores los cuales podemos manipular para visualizar en el LCD el estado lógico en el que se encuentran dichos interruptores.

Joystick hacia la izquierda aquí observamos que se enciende un led que está conectado en un pin el cual después de 1 segundo pasara a estar en alta impedancia por lo que dicho led se apagará y mostraremos en el LCD que ahora se encuentra en Alta Impedancia.

Para la parte de configuración de entrada y salida se utilizó un buffer de tres estados conectados entre los pines B0 y B1 hacia las circuiterías de entrada (interruptores) y salida (Leds) ya que si no se hacía esto se interferían entre si por estar conectados a los mismos pines del Atmega169. Todo esto se aprecia en la simulación.

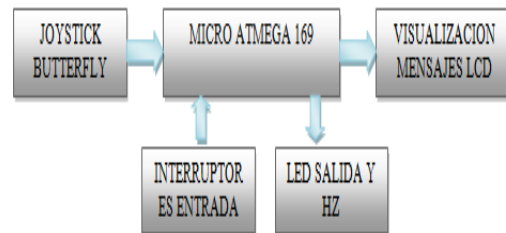
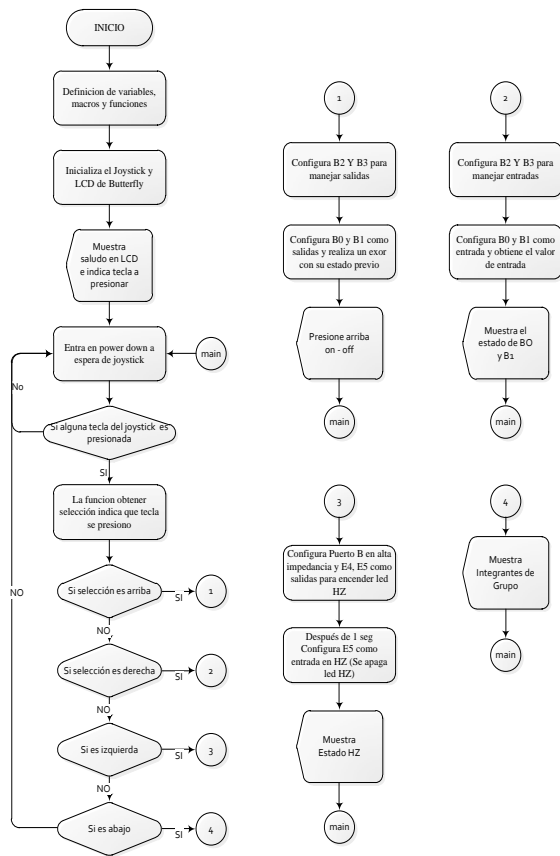


Figura 7. Diagrama de bloques Menú de entradas, salidas y alta impedancia

### 4.3.1. Diagrama de flujo Menú de entradas, salidas y alta impedancia



### 4.3.2. Algoritmo de Menú de entradas, salidas y alta impedancia.

1. Definimos variables, funciones y macros, inicializamos joystick y LCD.
2. Mostramos saludo por LCD e indicaciones de teclado.
3. Ponemos al microcontrolador en sleep power down en donde esperara hasta que se presione una tecla del joystick, que causara una interrupción y lo sacara de ese estado.
4. Cuando se presiona una tecla del joystick el microcontrolador recibirá una interrupción, la cual activara una función que verificara que tecla ha sido presionada.
5. Si se presiono la tecla arriba entrara en una subrutina en la que configuraremos los pines B0 y B1 como salidas y los pines B2 y B3 se configuran para manejar un buffer triestado externo que habilitara los led hacia B0 y B1, ahora cada vez que presiones hacia arriba en microcontrolador encenderá o apagara los led conectados hacia el.
6. Si se presiona la tecla derecha se configurara B0 y B1 como entradas y además se reconfigura B2 y B3

para que el buffer externo conecte los interruptores en las entradas BO y B1, luego leeremos el estado lógico en estos dos pines y mostraremos el resultado en el LCD.

7. Si presionamos la tecla izquierda por medio de dos pines E4 como salida en alto y E5 como salida en bajo encenderemos un led conectado entre ellos, que luego de 1 segundo se apagara ya que cambiaremos el estado de E5 a entrada en Hz y mostraremos en el LCD el mensaje de "HZ".

8. La tecla hacia abajo mostrara en el LCD el mensaje del numero de grupo y los integrantes de este proyecto.

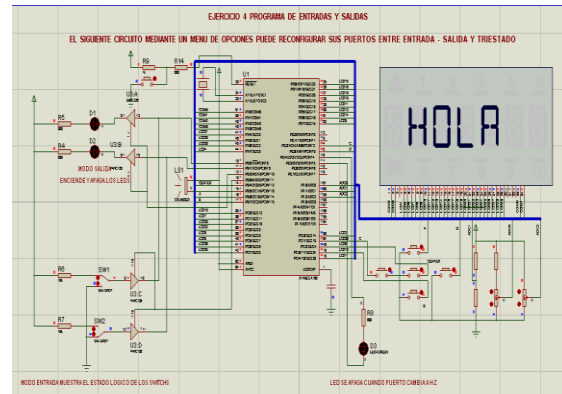


Figura 8. Esquemático de Menú de entradas, salidas y alta impedancia.

## 5. Conclusiones

Con el presente proyecto hemos mostrado como de manera simple y sin necesitar mucho tiempo, se puede implementar un sistema robusto de aprendizaje y prueba de buenas prestaciones, gracias a la utilización del Kit AVR Butterfly de Atmel, ya que este posee muchas características que nos facilitan su programación y utilización y en el cual podemos implementar un sin número de ejercicios limitados únicamente por la imaginación de quien los diseñe.

Además hemos comprobado las excelentes características de los microcontroladores Atmel, en este caso el ATmega169 que viene en el Kit AVR Butterfly que hemos utilizado, siendo este únicamente una muestra de lo es la familia completa de microcontroladores de Atmel, permitiéndonos de esta manera expandir nuestros conocimientos hacia otras arquitecturas de microcontroladores, y así tener un mayor rango de criterios a la hora de escoger un

dispositivo o integrarlos según las especificaciones que se necesiten.

Hemos podido analizar tres diferentes arquitectura de puertos de Entrada y Salida desde la más básica con Intel, pasando por la más comercial como Microchip, hasta otra con mayor complejidad pero más versátil como Atmel. De esta forma adquirimos mayores conocimientos para trabajar con uno u otro fabricante según las necesidades de nuestro trabajo o proyecto.

## 6. Recomendaciones

Debemos tener cuidado cuando conectamos cargas a los puertos de salida del Atmega169, y no exceder la capacidad de corriente de cada puerto, que ocasionaría el daño de dicho puerto o el microcontrolador.

Debemos tener siempre en cuenta que el AVR Butterfly ocupa los puertos B y D cuando utilizamos el joystick y la LCD para no implementar circuitería en dichos pines cuando estamos utilizando dichas herramientas del Kit.

Cuando se programa el AVR Butterfly a través del puerto serie y con el AVR Studio 4, en ocasiones el AVRProg no se conecta, entonces lo que debemos hacer es dar clic en aceptar en el mensaje de error de no conexión y luego sin presionar el joystick en al AVR Butterfly ejecutar nuevamente el AVRProg, veremos que la aplicación comenzara con normalidad.

Debemos tener siempre en cuenta al programar nuestras aplicaciones que el espacio de memoria a utilizar deberá ser el total de la flash menos el espacio del Bootloader que se encuentra programado en el microcontrolador.

Es recomendable manipular con mucho cuidado el Kit AVR Butterfly ya que es de un tamaño reducido, por lo que es fácil que se nos pueda escapar de las manos y por lo tanto estropearse.

## 7. Bibliografía

[1] Pardue Joe, C Programming for Microcontrollers, 2005.

[2] Brochure Proteus descargado de la Pagina Oficial LabCenter Electronics <http://downloads.labcenter.co.uk/proteus7brochure.pdf>

[3] AVR Butterfly Evaluation Kit - User Guide que se lo puede descargar de la pagina de Atmel <http://www.atmel.com/tools/AVRBUTTERFLY.aspx?tab=documents>

[4] Figura 1 obtenida de la pagina de Atmel <http://www.atmel.com/tools/AVRBUTTERFLY.aspx>