[[1]](#footnote-1)

PROYECTO FINAL: Microcontrolador basado en una CPU 8-bit Atmel AVR usando

El simulador de circuitos ISIS Proteus.

Laura Elena Aristizabal Millán, Programa de Ingeniería Electrónica, Facultad de Ingeniería, Universidad del Quindío - Colombia.

*Resumen*— En el desarrollo de este proyecto final se mostrará la construcción de un Microcontrolador con CPU AVR a partir de las indicaciones y requerimientos dados como lo son: Puertos de entrada/salida con opción de configuración, Memoria de programa (Flash Program Memory), Memoria de datos (Data SRAM), Temporizador de 8 bits con opciones de configuración: Normal, CTC y Fast PWM. Con esto se logra simular todas sus partes internas las cuales serían: configuración de registros, instrucciones, saltos de líneas, operaciones lógicas, generador de pulsos, comparador lógico, entre otras. Para programar el microcontrolador se usó AVR Core de 8 bits.

***Abstract*—** In the development of this final Project Construction UN Microcontroller is displayed with CPU AVR From Indications and requirements given As son ports input / output option configuration , program memory ( Flash program memory ) Memory data (data SRAM) , 8-bit timer configuration options of normal, CTC and fast PWM . This is achieved with contradictory simulate all internal which would be: Setting records, INSTRUCTIONS, newlines, logic operations, pulse generator, logical comparison, among others. Program for the microcontroller Core AVR 8-bit used

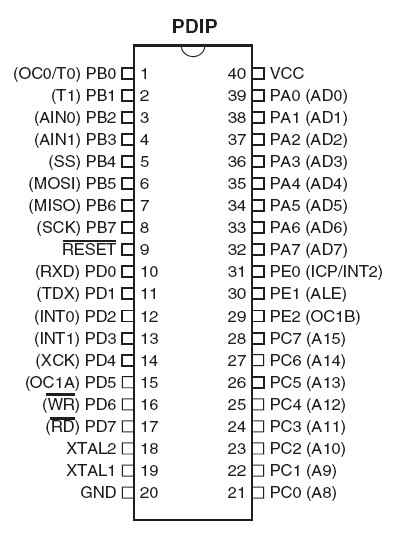
*Índice de Términos –* AVR, Microcontrolador, Registros, Puerto, Procesador, RAM, Temporizador.

***Index Terms —*** AVR,Microcontroller, Logs, Port, processor, RAM, Timer.

# **INTRODUCCIÓN**

Para que los microcontroladores puedan comunicarse con el mundo externo y poder interactuar con el ambiente físico, se dotan de PUERTOS o pines que se conectan a otros sistemas o componentes diversos. El Atmega16A cuenta con 4 puertos de 8 bits programables individualmente, ya sea como entrada o como salida. Estos puertos son PORTA, PORTB, PORTC y PORTD, que están distribuidos entre los pines del chip. El número de puertos varía de un modelo AVR a otro, dependiendo su tamaño y sus capacidades, pero en general todos los puertos de los AVR de 8 bits se programan de la misma manera.

En la siguiente imagen se ven los pines del chip asociados a cada puerto, cada pin, aparte de poder brindar una E/S de propósito general, puede brindar otra función o estar conectado a algún periférico interno.



***Figura 1.*** *Esquemático del ATmega 8515 (Pin Out)*

# **objetivos**

OBJETIVO GENERAL

* Implementar un sistema de configuración, registro y memoria para un microcontrolador, usando la herramienta ISIS PROTEUS para la parte del hardware y AVR de 8 bits(CODEBLOCKS) para la parte de software.

OBJETIVOS ESPECÍFICOS

* Valor obtenido en cada uno de los registros al configurar el puerto.
* Valor almacenado en el registro al capturar una dirección o un dato.
* Valor del contador del programa al ejecutar una instrucción desde el microcontrolador.
* Valor de las banderas almacenadas en el registro de estado al ejecutar una instrucción.
* Valor de salida en la ALU al ejecutar una operación lógica
* Valor de la unidad de control para activar cada uno de los elementos al capturar y ejecutar una instrucción.

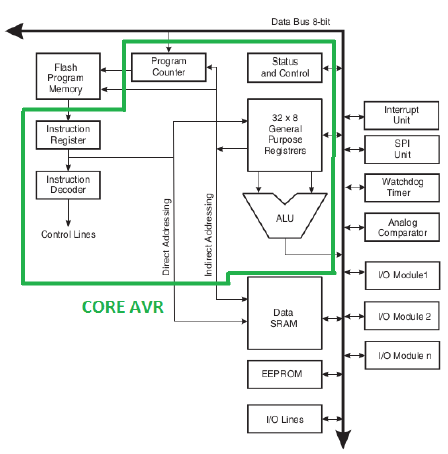
# **desarrollo**

Un Microprocesador es un circuito integrado, el cual es conocido como como el cerebro de un sistema informático, tales como computadores, celulares, tablet’s entre otros. Este está constituido básicamente por un banco de instrucciones, instrucción de registro, contador de programa, una unidad de control, un status register y una ALU.

Donde el banco de registro almacena los valores cargados a cada registro que el microprocesador tenga, permitiendo luego operar estos valores en la ALU; las instrucciones de registros permiten almacenar la dirección de la siguiente instrucción que se va a ejecutar, por su parte el contador de programa indica la posición de la secuencia de instrucciones que se está ejecutando.

De igual forma se tiene la unidad de control la cual es una de las más importantes que posee el microprocesador, puesta que esta se encarga en determinar qué operación debe hacer la ALU, ,mientras que el status register es el encargado de almacenar los estados de cada bandera que se halla activado de acuerdo a la operación hecha. Por ultimo tenemos la ALU se denomina unidad aritmético lógica, la cual permite realizar operación como suma, resta, multiplicación etc.

Para que un microprocesador ejecute una tarea dada se requiere que este tenga un conjunto de instrucciones que le indiquen que debe de hacer, estas instrucciones se encuentran programadas en un lenguaje de bajo nivel o de máquina, conocido como ensamblador. En la figura 2 se muestra la parte esquemática realizada en este proyecto.



***Figura 2.*** *Esquemático del procesador Core AVR.*

Básicamente la programación de microcontroladores se basa en la configuración de los registros que dispone junto con sus bits, y en la implementación de algoritmos y rutinas que interactúen con estos registros.

Pero, ¿qué es un registro?  
Un registro es un espacio de almacenamiento especial del microcontrolador que está relacionado (conectado) a algún periférico y que cada bit representa el funcionamiento o estado del mismo. Los registros tienen un nombre único de acuerdo a su funcionamiento, y los bits individuales del registro también suelen tener un nombre único y

Un propósito específico. El tamaño de los registros suele estar relacionada con el ancho de bus de cada arquitectura, para el caso de los AVR de 8 bits, la mayoría de sus registros son de 8 bits.

Por cada puerto de E/S existen 3 registros que lo controlan: DDRx, PORTx y PINx (la x hace referencia al nombre del puerto seleccionado: A, B, C, etc.). Cada bit de estos registros hace referencia a un pin físico del puerto en el chip.

1. **DDRx : Data Direccion Register**

Este registro es el Registro de Dirección de datos, y como su nombre lo indica establece la dirección de datos de cada pin del puerto, quiere decir que con este registro se programan los pines del puerto como entrada o como salida.

Cada pin en un puerto es independiente y por lo tanto la totalidad del puerto no tiene que ser configurado totalmente como un puerto de entrada o salida. Escribiendo un 1 en la posición del pin en el DDRx configurará ese pin como un pin de salida y escribiendo un 0 configurará el pin como un pin de entrada.

# **PORTx:**

Se trata un registro con una doble funcionalidad dependiendo de si el pin del puerto es de entrada o de salida.

Si el puerto está configurado como salida (DDRx = 1), el valor escrito en cualquiera de los bits de este registro será reflejado en el pin del microcontrolador, es decir un 1 pone la salida en Alto y un 0 pone la salida en Bajo.

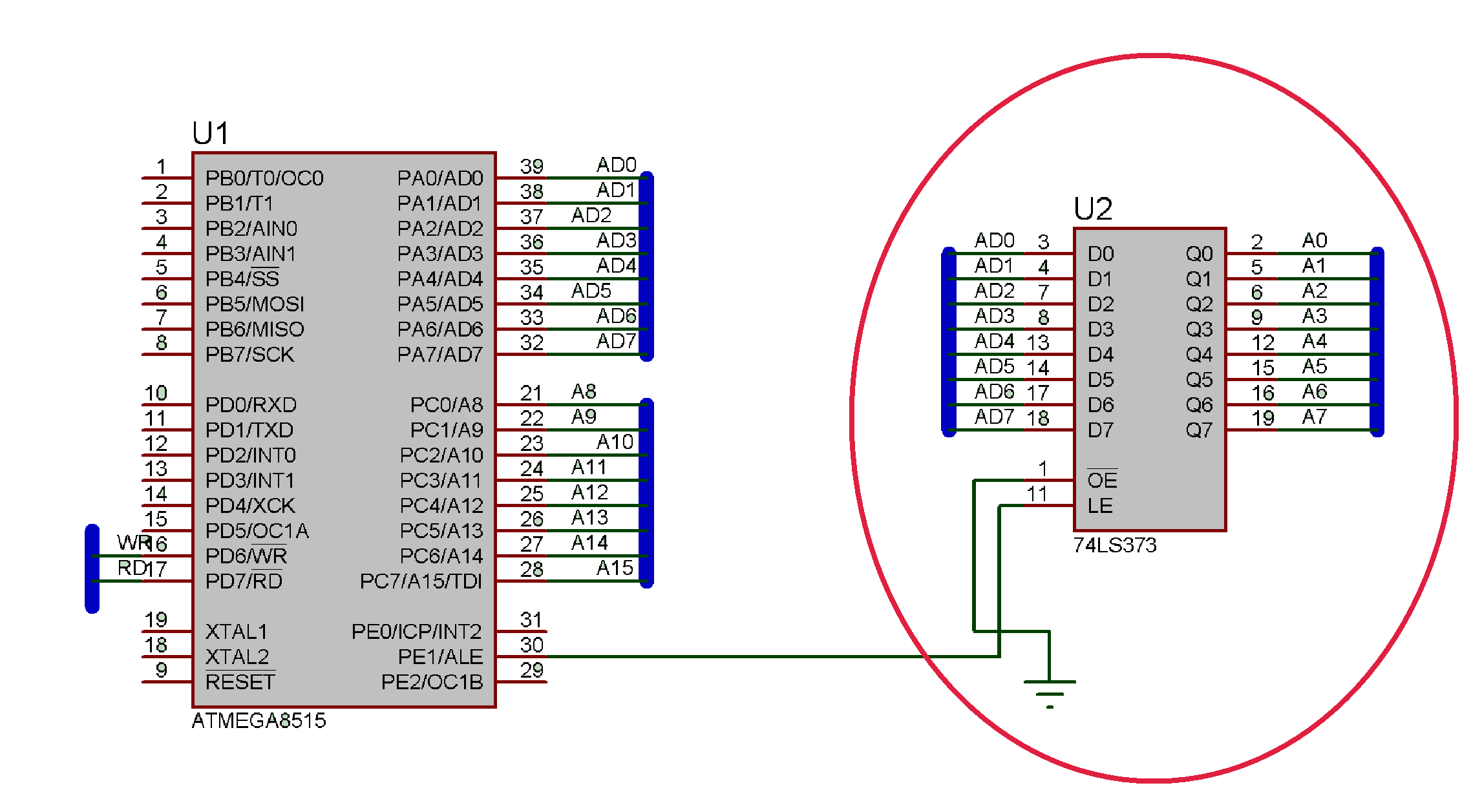
Por el contrario, si el puerto se configura como entrada (DDRx = 0), escribir un 1 en cualquier bit del registro habilitará la resistencia PULL-UP en el pin correspondiente al puerto. Si se escribe un 0 la resistencia PULL-UP queda desconectada.

# **PINX:**

Este registro es de solo lectura y sirve para leer el estado de los pines del puerto configurados como entrada. A vece se cae en el error de que si leemos el registro PORTx obtendremos el valor de los pines, y así es, pero solo leeremos el valor actual de los pines configurados como salida, para lectura de puertos se utiliza este registro.

En algunos modelos de AVR el registro PINx también es de escritura, con lo cual al escribir un 1 en cualquiera de sus bits, conmutara ese estado del pin correspondiente siempre y cuando el pin sea de salida.

El concepto de "latch", es importante en la creación de dispositivos de memoria. La función de éste circuito es "capturar" el valor creado por las señales de entrada al dispositivo y mantener ese valor hasta que lo cambie alguna otra señal o se muestre en la salida.

******

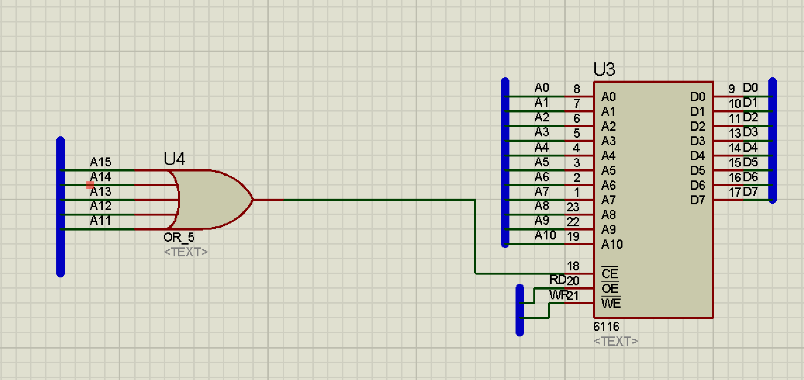
***Figura 3.*** *LATCH 74LS373 conectado al ATmega 8515.*

Un **latch** tiene una configuración biestable asíncrono usado para almacenar información en sistemas lógicos digitales. Un latch puede almacenar un bit de información, asimismo los

latches se pueden agrupar de tal manera que logren almacenar más de 1 bit, por Ejemplo el 'latch quad ' (capaz de almacenar cuatro bits) y el 'latch octal' (capaz de almacenar ocho bits).

Los latches son dispositivos biestables asíncronos que no tienen entrada de reloj y cuyo cambio en los estados de salida es función del estado presente en las entradas y de los estados previos en las salidas (retroalimentación). Los latches a diferencia de los flip-flops no necesitan una señal de reloj para su funcionamiento.

La memoria RAM está destinada al almacenamiento de información temporal que será utilizada por el procesador para realizar cálculos u otro tipo de operaciones lógicas. En el espacio de direcciones de memoria RAM se ubican además los registros de trabajo del procesador y los de configuración y trabajo de los distintos periféricos del microcontrolador. Es por ello que en la mayoría de los casos, aunque se tenga un espacio de direcciones de un tamaño determinado, la cantidad de memoria RAM de que dispone el programador para almacenar sus datos es menor que la que puede direccionar el procesador.



***Figura 4.*** *RAM 6116.*

**TEMPORIZADOR:**

Es un circuito sincrónico para el conteo de los pulsos que llegan a su poder para conseguir la entrada de reloj. Si la fuente de un gran conteo es el oscilador interno del microcontrolador es común que no tengan un pin asociado, y en este caso trabajan como temporizadores. Por otra parte, cuando la fuente de conteo es externa, entonces tienen asociado un pin configurado como entrada, este es el modo contador.

Los temporizadores son uno de los periféricos más habituales en los microcontroladores y se utilizan para muchas tareas, como por ejemplo, la medición de frecuencia, implementación de relojes, generador de pulsos PWM, comparadores, para el trabajo en conjunto con otros periféricos que requieren una base estable de tiempo, entre otras funcionalidades. Es frecuente que un microcontrolador típico incorpore más de un temporizador/contador e incluso algunos tienen arreglos de contadores. Los tamaños típicos de los registros de conteo son 8 y 16 bits.



***Figura 5.*** *Temporizador de 8 bits.*

El temporizador de 8 bits puede usar un clock externo o interno con la frecuencia de reloj dividida por 8, 64, 256, o 1024, tiene dos comparadores de 8 bits, los tipos de registro y su lógica de control se define de la siguiente manera

● **TCCRn:** Registro de Control del Temporizador. Permite modificar el funcionamiento del temporizador.

● **TCNTn:** Registro de datos que contiene la cuenta actual del temporizador.

● **OCRn:** Registro de datos usado para almacenar un valor de comparación. El valor almacenado en este registro es comparado con el valor actual del temporizador para generar interrupciones o modificar el ancho de pulso de una señal PWM.

● **TOVn:** Interrupción por sobreflujo. Se genera cuando el valor actual del temporizador ha llegado al valor máximo y se reinicia.

● **OCn:** Pin de salida del temporizador. Las señales PWM salen por este pin.

● **OCn (Int. Req.):** Interrupción por comparación. Es generada cuando el valor actual del temporizador es igual al valor en el registro OCRn.

● **BOTTOM:** Se coloca en uno cuando el valor del temporizador es cero, esto es, cuando llega al valor mínimo.

● **TOP:** Se coloca en uno cuando el valor del temporizador es **0xFF** o es igual al valor en OCRn, esto depende del modo de trabajo.

● **CLKTn:** Señal de reloj proveniente de un preescalador.

● **COUNT:** Incrementa o decrementa en uno el valor del temporizador.

Dependiendo de la velocidad de cambio en esta señal, el temporizador incrementa o decrementa más rápido o más lento.

● **CLEAR:** Inicializa el temporizador a cero.

# **Conclusiones**

* teniendo la construcción previa del puerto de 8 pines y sus respectivas pruebas a partir de compuertas lógicas se procede a configurar dos puertos de los creados, uno como entrada y el otro como salida.
* Es importante saber que para el correcto funcionamiento de nuestro puerto éste debe tener conectadas todas las direcciones de configuración (WDx, RDx, WPx, RPx, y RRx) de lectura, escritura y registro (WR, RD, A12, A1, A0) a partir de las compuertas lógicas, de ésta forma el PUERTO puede recibir datos y direcciones desde el procesador
* Se debe cagar dentro del ATmega un archivo .hex creado y compilado previamente en la plataforma AVR CODEBLOCKS en un nuevo proyecto que configure los puertos como entrada y salida de datos cada uno con las direcciones que el ATmega tiene por defecto.
* El preescalador tiene como función, dividir la frecuencia antes de cada incremento del COUNT.
* Un módulo contador/temporizador de 8 bits, que cuenta con un preescalador programable de 8 bits Puede funcionar como temporizador o como contador en modo temporizador el módulo Timer se incrementa con cada ciclo de instrucción.

*Referencias*

1. *Wikipedia, Microcontrolador, 2014. Disponible en:*

*http://es.wikipedia.org/wiki/Microcontrolador*

1. *Atmel® Corporation, ATmega16 8-bit AVR® microcontroller.*

*[3]* *Ing. Capacho, Luis Miguel, notas de clase.*

*[4] Ing. López, Gerardo, notas de laboratorio.*

1. [↑](#footnote-ref-1)