

Estudio detallado de la Conversión Analógica Digital empleada en los Microcontroladores de Atmel con la Implementación en Hardware de Ejercicios Prácticos

Grace Yagual⁽¹⁾ Miguel Rodríguez⁽²⁾ Danilo Riera⁽³⁾ Carlos Valdivieso⁽⁴⁾
Facultad de Ingeniería en Electricidad y Computación
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 Vía Perimetral
Apartado 09-01-5863. Guayaquil, Ecuador
gralyagu@espol.edu.ec⁽¹⁾ migparod@espol.edu.ec⁽²⁾ ariera@espol.edu.ec⁽³⁾
cvaldiv@fiee.espol.edu.ec⁽³⁾

Resumen

El proyecto tiene como finalidad el desarrollo de distintas aplicaciones del Convertidor Analógico Digital (ADC) en el microcontrolador Atmega169 con ayuda del kit AVR Butterfly y demás dispositivos electrónicos; consta de cinco ejercicios, tres de ellos realizados en lenguaje de programación C y los dos restantes en lenguaje de programación Assembler.

Ejercicios que mediante cierta regulación en la entrada del ADC del microcontrolador nosotros obtenemos una salida, visualizada ya sea en la pantalla LCD o en leds representados de forma binaria que indicarán los valores de temperatura, voltaje y luz; o representada por una señal de sonido mediante un speaker de la plataforma mencionada.

En cada uno tenemos su teoría respectiva, su implementación en software y hardware, su documentación y su toma de fotografía para que quede registrado.

Abstract

The purpose of this project is to develop various applications of Analog to Digital Converter (ADC) in the Atmega169 microcontroller using the AVR Butterfly and other electronic devices and consists of five exercises, three of them made in C language and the remaining two in Assembler.

Exercises that by some regulation in the microcontroller ADC input we get an output, displayed on the screen either LCD or LEDs represented in binary form to indicate the values of temperature, voltage and light, or represented by a sound signal by a speaker from the platform above.

In theory each have their respective implementation in software and hardware, documentation and photographs recorded.

1. Introducción

Desde la invención de los semiconductores, el desarrollo de la tecnología digital ha dado lugar a dispositivos cada vez más complejos y rápidos. Entre ellos los microprocesadores y los microcontroladores.

Estos dispositivos se encuentran en nuestro trabajo, en nuestra casa, etc. Controlan el funcionamiento de los teclados de las computadoras, están en los teléfonos celulares, en general, en todo aparato electrónico que posea un grado de automatismo.

Se dice que un controlador es un dispositivo que se emplea para manejar uno o varios procesos. Un microcontrolador es un chip que posee en su interior a un microprocesador, memoria de programa, memoria de datos y puertos para comunicarse con el exterior. Como es muy frecuente el trabajo con señales analógicas, éstas deben ser convertidas a digital y por ello muchos microcontroladores incorporan un conversor analógico-digital, el cual se utiliza para tomar datos de varias entradas diferentes que se seleccionan mediante un multiplexor.

Las resoluciones más frecuentes son 8 y 10 bits, que son suficientes para aplicaciones

sencillas. Para aplicaciones en control e instrumentación están disponibles resoluciones de 12bit, 16bit y 24bit.

2. Software utilizado

Hemos obtenido la realización de estas aplicaciones con ayuda de dos tipos de software: la interfaz de simulación Proteus y AVR Studio que utilizaremos para la programación del Atmega169.

2.1 AVR Studio 4

AVR Studio 4 es un entorno maduro de desarrollo integrado para el desarrollo de aplicaciones Atmel AVR 8-bits. [3] AVR Studio apoya al diseñador en el diseño, desarrollo, depuración y parte de la comprobación del proceso.

Provee herramientas de manejo, editor de código fuente, que puede ser en lenguaje ensamblador o lenguaje C. Una de las características principales de este producto tenemos es que se integra con compilador GCC.

El lenguaje ensamblador nos permite manipular con mayor detalle los valores de los registros que posee el microcontrolador que estemos utilizando.

De la misma manera las herramientas que posee nos permiten observar los valores que toman las diferentes variables que hayamos declarado a lo largo del desarrollo del programa.

El lenguaje GCC (GNU C Compiler) permite una programación de alto nivel de modo que consigamos desarrollar aplicaciones de mayor grado de desarrollo. El 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.

Este entorno consiste de varias ventanas y sub-módulos. Cada ventana apoya a las partes del trabajo que se intenta emprender.

2.2 Proteus.Isis

Proteus 7.7 es un software de simulación y diseño de PCB distribuido por Labcenter Electronics, que permite analizar el comportamiento de los circuitos, previo a la implementación física del circuito.

Es la única plataforma que nos ofrece la posibilidad de co-simular el código para microcontroladores en bajo y alto nivel.

Provee de herramientas de edición, a su vez de librerías que contienen componentes análogos y familias de microcontroladores entre ellas Microchip, Motorola, ARM, AVR, y microprocesadores de la familia ATMEL, Motorola entre otros.

Una de las ventajas para el desarrollo del proyecto es que se hizo uso de los ejemplos para ATMEL en el cual encontramos el Kit AVR Butterfly funcionando con su programa demo, con lo cual no fue necesario el diseño del esquemático correspondiente.

3.1 Hardware utilizado

Para el desarrollo de las diferentes aplicaciones que demuestra el uso y manejo del convertidor analógico digital del Atmega169 nos hemos apoyado en los recursos que provee la plataforma AVR Butterfly.

3.1 Kit AVR Butterfly

El kit de evaluación del AVR Butterfly está diseñado para mostrar las características y beneficios más importantes de los microcontroladores ATMEL. [4]



El microcontrolador usado por el AVR Butterfly es el ATmega169, que combina la tecnología Flash con el más avanzado microcontrolador de 8 bits disponible.

Los siguientes recursos están disponibles en el Kit AVR Butterfly: -

Microcontrolador ATmega169V.

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.

Termistor de Coeficiente de Temperatura Negativo – NTC, para obtener la temperatura.

Convertidor de nivel RS-232 e interfaz USART.

Resistencia Dependiente de Luz (LDR), para sensar y medir la intensidad luminosa.

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 a la plataforma AVR Butterfly.

4. Desarrollo de las aplicaciones

4.1.- Medidor de temperatura

Especificaciones [1]

En este ejemplo se utilizará el ADC para realizar mediciones de temperatura a través del Sensor de Temperatura del tipo Resistor NTC (Coeficiente Negativo de Temperatura), el cual viene incluido en el Kit AVR Butterfly.

Un termistor NTC se caracteriza por el hecho de que cuando la temperatura disminuye la resistencia aumenta y viceversa. La temperatura puede ser calculada, usando un divisor de voltaje para leer el voltaje sobre el termistor a través de los canales del ADC en el Atmega169.

La ecuación para el cálculo de la temperatura en grados Celsius es la siguiente:

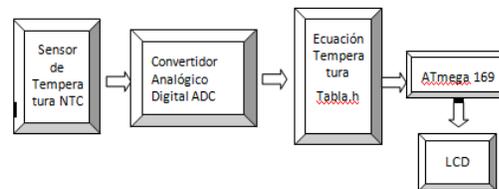
$$T_{\text{temperatura}} = \frac{\beta}{\left(\ln \left(\frac{ADC}{1024 - ADC} \right) + \frac{\beta}{T_{\text{amb}}} \right)} - T_0$$

La temperatura en Fahrenheit:

El AVR Butterfly es capaz de medir temperaturas desde -10°C/+14°F hasta 60°C/140°F con una precisión de ± 1°C.

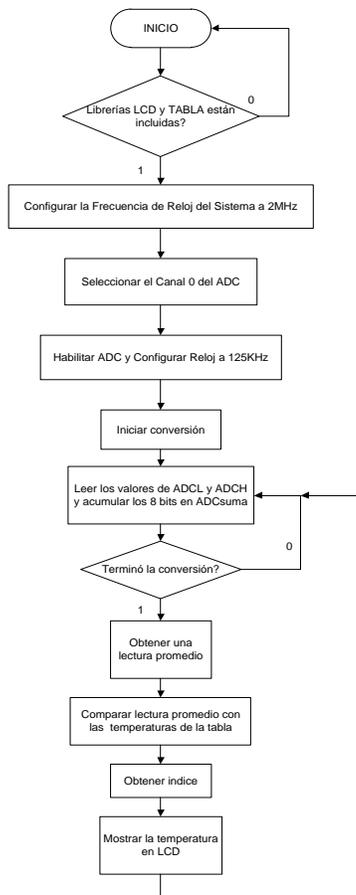
Diagrama de bloques

Los recursos del AVR Butterfly que se serán usados en el medidor de temperatura se muestran en el diagrama de bloques a

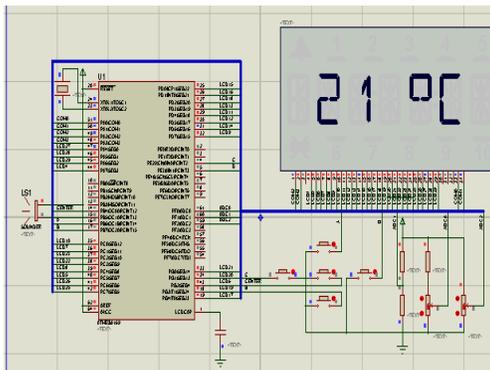


continuación.

Diagrama de flujo [5]



Circuito armado en Proteus



4.2.- Aplicación del ADC usando LDR

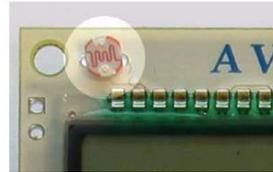
Especificaciones [2]

En este ejemplo aprenderemos sobre el convertidor analógico digital usando una resistencia que depende de la luz (LDR). Con el movimiento de la mano sobre la LDR, el altavoz (speaker) producirá un sonido parecido al de las antiguas películas de ciencia ficción de la década de 1950. Luego de producir el sonido en el auricular, habrá una pausa por un período de tiempo

que es determinado por el valor de la tensión / resistencia que se mide.

El resultado es un cambio de frecuencia del sonido sobre la base de nuestra aportación de la LDR.

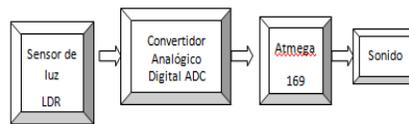
El sensor de luz será ubicado en la parte delantera del AVR Butterfly, sobre la pantalla del LCD.



Un LDR se utiliza para medir la luz y se caracteriza por el hecho de que cuando la luz disminuye la resistencia aumenta.

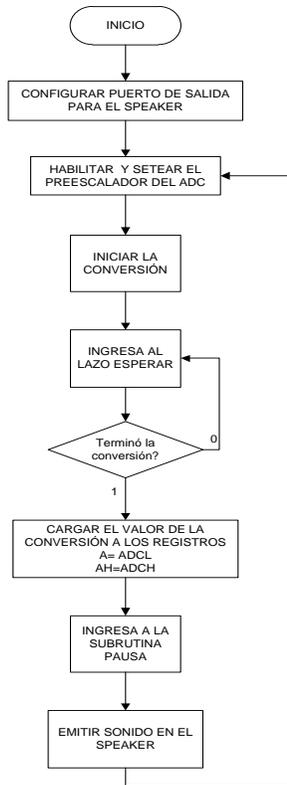
La luz puede ser calculada, usando un divisor de voltaje y leyendo el voltaje sobre el LDR a través de los canales del convertidor analógico digital ADC en el Atmega 169, existente en el AVR Butterfly.

Diagrama de bloques

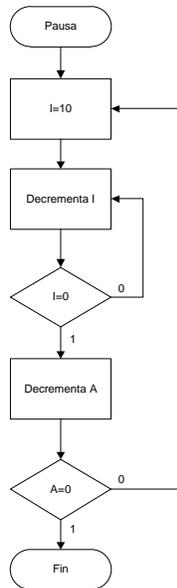


Diagramas de flujo

Diagrama de flujo principal. [5]



Subrutina PAUSA [5]



4.3.- ADC – Demostración en leds

Especificaciones

En este ejemplo usamos un potenciómetro

de 5kohmios que lo regulamos en la entrada del microcontrolador por ADC0 y mediante programación nos mostrará en sus salidas por el puerto B un número de forma binaria; cada vez que se varíe el potenciómetro lo relacionará con un voltaje que está en el rango de 0 a 5 voltios.

Diagrama de bloques

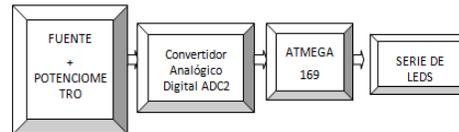
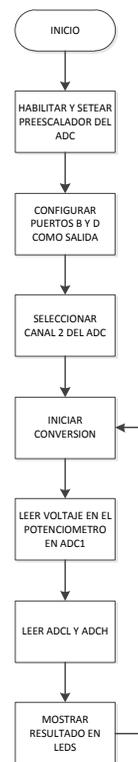
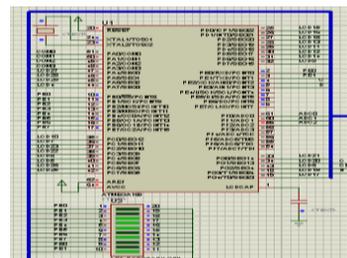


Diagrama de flujo [5]



Circuito armado en PROTEUS



4.4 Lectura de voltaje

Especificaciones

En el display del Avr Butterfly se mostrara un valor comprendido entre el 1 y el 9 y la letra V que representa el valor del voltaje el cual es regulado y fijado mediante un potenciómetro que esta ubicado en el ADC del microcontrolador.

Diagrama de bloques

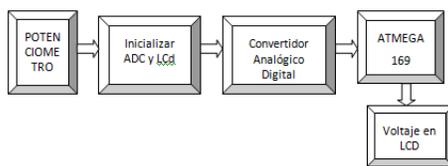
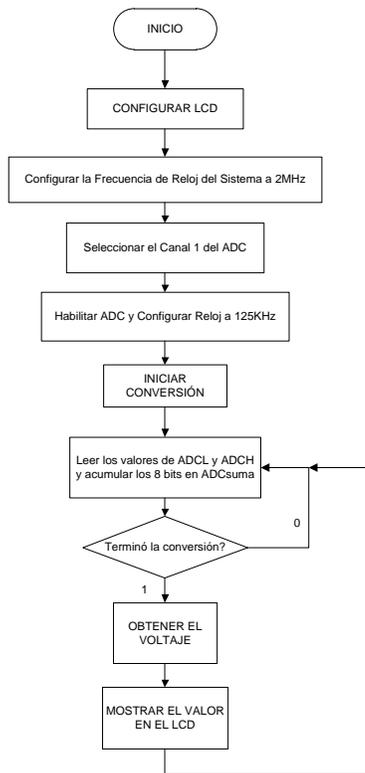
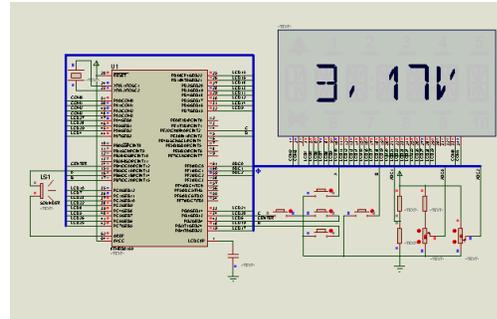


Diagrama de flujo [5]



Circuito armado en PROTEUS



4.5 Medidor de luz

Especificaciones

Variando la resistencia en el LDR, aumentando y disminuyendo su valor cambiara en la pantalla del LCD el valor de un numero hexadecimal; esto por medio de la entrada del ADC0 donde normalmente estamos programando nuestros programas.

Diagrama de bloques

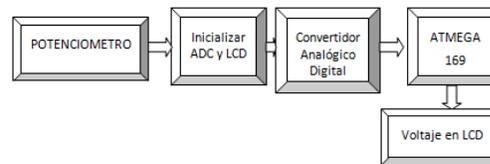
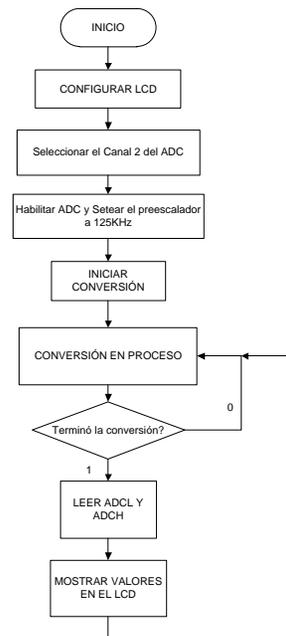
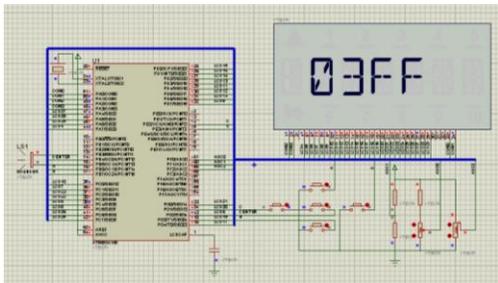
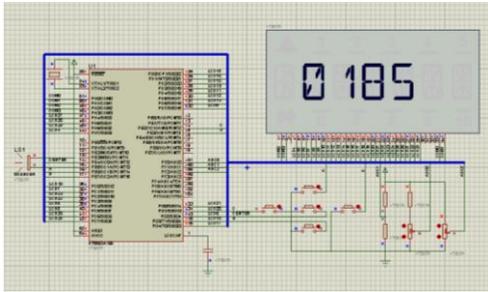


Diagrama de flujo [5]



Circuito armado en PROTEUS



5. Conclusiones

El Kit AVR Butterfly es una poderosa herramienta de aprendizaje, que dispone de los componentes necesarios para los ejemplos propuestos, entre ellos contiene en especial los sensores de temperatura (NTC) y de luz (LDR), que nos permitieron obtener resultados satisfactorios en el desarrollo de los diferentes ejercicios que mencionamos, con los cuales aprendimos el funcionamiento del ADC y descubrimos progresivamente las características del microcontrolador ATmega169.

Se consiguió un ahorro de tiempo y recursos gracias al entorno de programación AVR Studio y a la interfaz de simulación PROTEUS, que nos permitieron tener un enfoque de lo real antes de realizar la implementación física y las respectivas pruebas con el hardware, además de realizar los cambios que fueran necesarios para mejorar las aplicaciones.

El lenguaje C es óptimo en el desarrollo de software para microcontroladores, y nos ha facilitado la programación del módulo AVR Butterfly, ya que podemos segmentar el código fuente

en varias funciones especializadas, permitiéndonos volver a usar las mismas funciones en otras aplicaciones. Además, este lenguaje permite codificar más ordenadamente y al ser un lenguaje de medio nivel, es ideal para desarrollar aplicaciones de ingeniería.

6. Referencias

[1]. C Programming for Microcontrollers Featuring ATMEL's AVR Butterfly and the Free WinAVR Compiler

JoePardue

Edición 2005

Web: <http://www.smileymicros.com/>

Fecha de consulta: 22/11/2011

[2]. Beginners Introduction to the Assembly Language of ATMEL AVR Microprocessors

Gerhard Schmidt

Web: <http://www.avr-asm-tutorial.net>

Fecha de consulta: 20/12/2011

[3]. AVR Instruction Set

http://atmel.com/dyn/resources/prod_documents/doc0856.pdf

Fecha de consulta: 12/11/2011

[4]. AVR Butterfly User Guide

http://www.atmel.com/dyn/resources/prod_documents/doc4271.pdf

Fecha de consulta: 15/09/2011

[5]. Hoja de especificaciones del microcontrolador ATMEGA169.

http://www.atmel.com/dyn/resources/prod_documents/doc2514.pdf?&MMN_position=43:43

Fecha de consulta: 20/10/2011