

UNIVERSIDAD DE COSTA RICA

IE-0624 LABORATORIO DE MICROCONTROLADORES

Laboratorio # 3

Freddy Zúñiga Cerdas

A45967

Profesor

MARCO VILLALTA FALLAS

May 6, 2023

1 Introducción

En el presente trabajo se desarrolló un circuito capaz de emular el comportamiento de un voltímetro de cuatro canales basado en el Arduino UNO, el cual es capaz de leer, presentar en una pantalla y enviar a la computadora datos sobre la tensión de cada uno de los canales, pudiendo manejar los canales corriente alterna o directa.

El repositorio de Github se puede consultar en la siguiente dirección:

https://github.com/JackTheKnife16/IE-0624_Laboratorio_de_Microcontroladores_I_2023

2 Nota Teórica

En este apartado se mostrarán algunas características del microcontrolador así como la justificación de la utilización de los componentes externos y el diagrama de flujo del firmware creado para el microcontrolador.

2.1 Microcontrolador ATmega328P

Se utilizó la placa Arduino UNO para el presente laboratorio

2.1.1 Características Generales

El microcontrolador ATmega328P cuenta con las siguientes características:

- Microcontrolador AVR de 8 bits.
- Arquitectura RISC/Harvard.
- 4/8/16/64Kb Flash, 512b/1/2k bytes de SRAM y 256/512/1k bytes de EEPROM.
- 23 GPIOs.
- Timer/Counter de 8 y 16 bits.
- 8 canales PWM y comparador analógico.
- 6 canales 10-bit ADC.
- SPI, USART, TWI, DBG, BTLDR.

2.1.2 Diagrama de Bloques

En la Figura 1 se muestra en detalle el diagrama de bloques del microcontrolador.

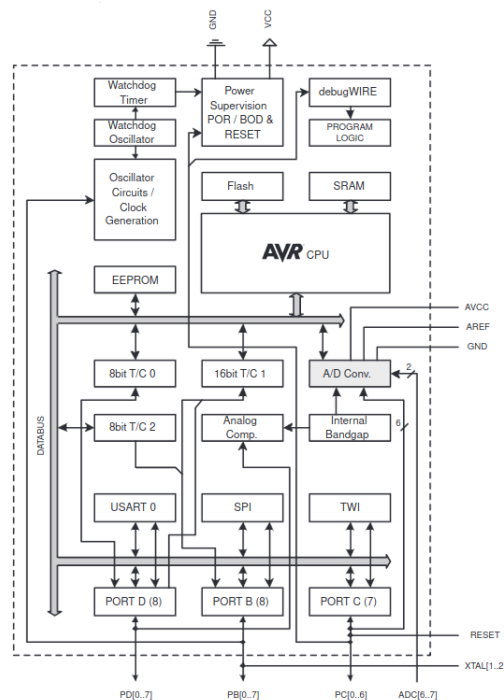


Figura 1: Diagrama de Bloques del ATmega328P [1]

2.1.3 Diagrama de Pines

En la Figura 2 se muestra el diagrama de pines del microcontrolador.

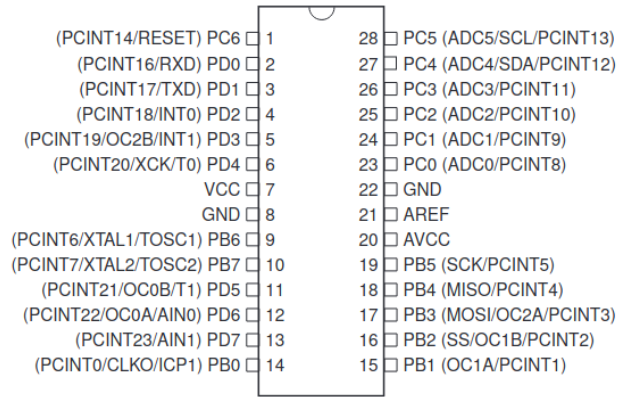


Figura 2: Diagrama de Pines del ATmega328P [1]

2.1.4 Características Eléctricas

En la Figura 3 se muestran las características eléctricas del microcontrolador.

- **Operating Voltage:**
 - 1.8 - 5.5V for ATmega48P/88P/168PV
 - 2.7 - 5.5V for ATmega48P/88P/168P
 - 1.8 - 5.5V for ATmega328P
- **Temperature Range:**
 - -40°C to 85°C
- **Speed Grade:**
 - ATmega48P/88P/168PV: 0 - 4 MHz @ 1.8 - 5.5V, 0 - 10 MHz @ 2.7 - 5.5V
 - ATmega48P/88P/168P: 0 - 10 MHz @ 2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V
 - ATmega328P: 0 - 4 MHz @ 1.8 - 5.5V, 0 - 10 MHz @ 2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V
- **Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48P/88P/168P:**
 - Active Mode: 0.3 mA
 - Power-down Mode: 0.1 µA
 - Power-save Mode: 0.8 µA (Including 32 kHz RTC)

Figura 3: Características Eléctricas del ATmega328P [1]

2.2 Diagrama Funcional del Circuito

En la Figura 4 se muestra el diagrama funcional para el laboratorio, se puede ver la representación de cada una de las partes del circuito, que consta de 3, circuito de entrada (botones y entradas para las cuatro tensiones), circuito de salida (pantalla lcd y leds indicadores) y el microcontrolador (en este caso la placa de desarrollo Arduino UNO).

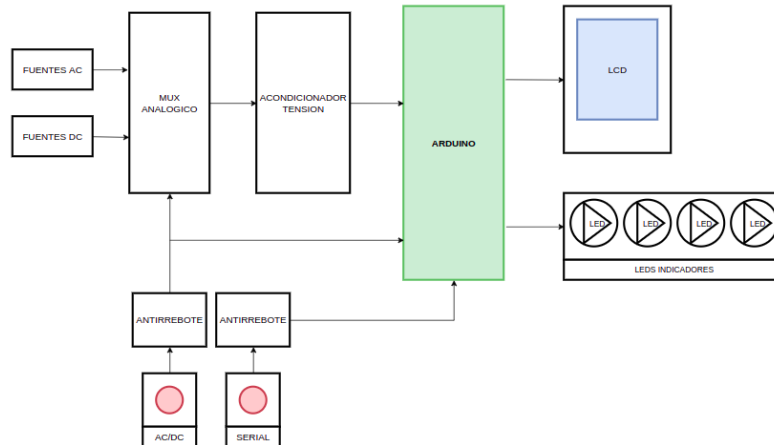


Figura 4: Diagrama funcional para el voltímetro de cuatro canales [imagen propia]

2.3 Pines Utilizados

En la Figura 5 se muestra las conexiones realizadas para el voltímetro de cuatro canales.

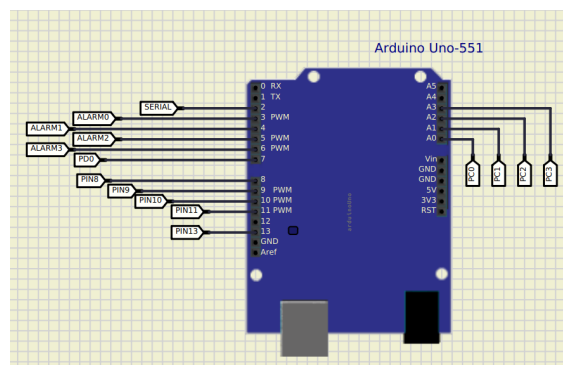


Figura 5: Conexiones de los pines [tomado de SimulIDE]

Pines de Entrada:

Tenemos como Pines Digitales de entrada al Pin 2 que controla la comunicación serial, y al Pin 7 que maneja la elección entre AC/DC.

También tenemos Pines Analógicos de entrada (A0, A1, A2, A3) que reciben las señales de las fuentes de tensión.

Pines de Salida:

Los Pines Digitales 3, 4, 5, 6 manejan los leds indicadores correspondientes a las fuentes 0, 1, 2 y 3, estos se activan cuando los voltajes son mayores a 20 o menores a -20 volts.

Los Pines Digitales 8, 9, 10, 11, 13 van a las entradas de la pantalla LCD: CS, Rst, D/C Mosi y SCK respectivamente.

2.4 Firmware del Circuito

Para el firmware se mostrará el diagrama de flujo para la función más importante del firmware, en el archivo de diagramas adjunto de este laboratorio se muestran otros diagramas para otras funciones del firmware, pero se omiten diagramas para funciones muy sencillas que no que se explican de manera autosuficiente en el código.

En la Figura 6 y 7 se muestra el diagrama para la función encargada de obtener el rms para las señales alternas.

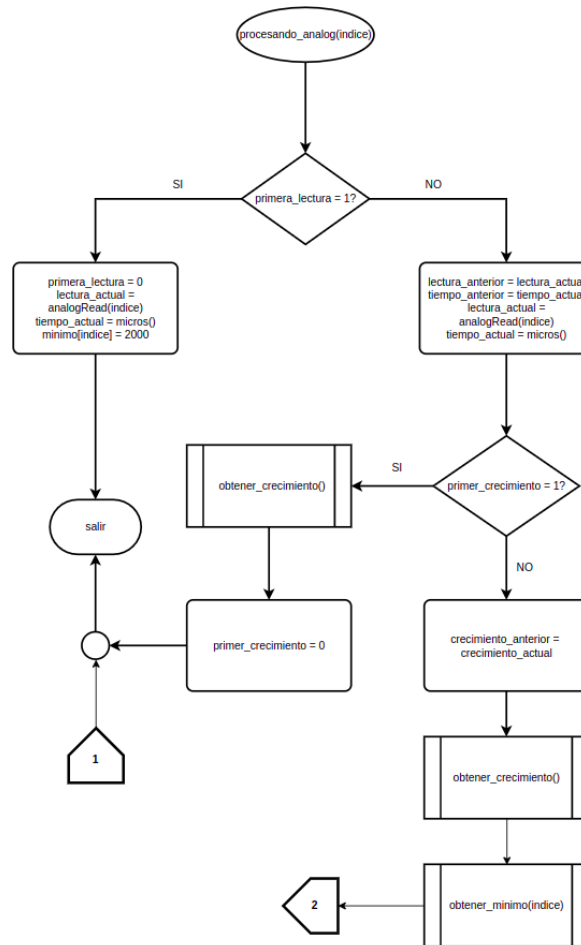


Figura 6: Diagrama del función `procesando_analog` [imagen propia]

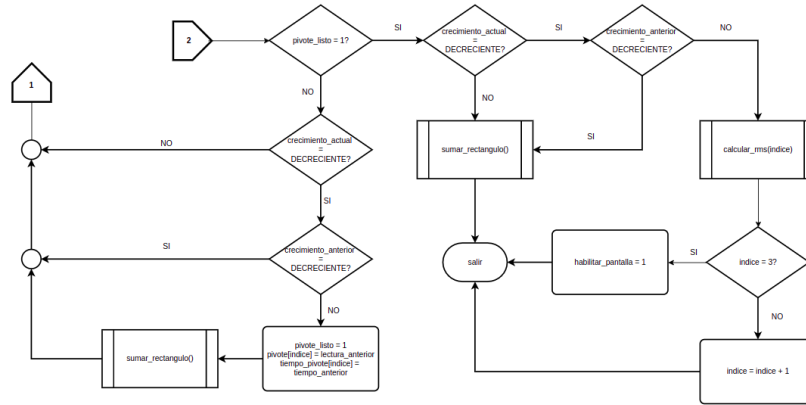


Figura 7: Diagrama del función procesando_analog [imagen propia]

Para los detalles de la implementación se recomienda ver el código hecho. Por otro lado lo que se intenta hacer con esta función es sacar el RMS utilizando la fórmula para valores discretos:

$$RMS \approx \sqrt{\frac{1}{T} \sum_{k=1}^n v^2(k(t_{k+1} - t_k)) \cdot (t_{k+1} - t_k)} \quad (1)$$

siendo que la fórmula exacta es:

$$RMS = \sqrt{\frac{1}{T} \int_{t_0}^{t_0+T} v^2(t) dt} \quad (2)$$

La idea de hacerlo de esta manera era dotar al programa de la capacidad de calcular las formas básicas de onda alterna: cuadrada, sinusoidal, sierra y triangular.

De esta fórmula se puede obtener lo que tenemos en la siguiente tabla:

tipo	RMS
sinusoidal	$\frac{amplitud}{\sqrt{2}}$
cuadrada	$amplitud$
triangular	$\frac{amplitud}{\sqrt{3}}$
sierra	$\frac{amplitud}{\sqrt{3}}$

Tabla 1: Tabla de RMS para ondas simples

2.5 Componentes Electrónicos Complementarios

El sistema que resuelve el problema planteado para este proyecto se diseñó con tres partes fundamentales, primero el circuito de entrada, segundo el microcontrolador y tercero el circuito de salida. En la Tabla se listan los componentes utilizados para cada uno de dichas partes, así como su precio en el mercado en colones.

Código	Tipo	Característica	Cantidad	Precio	Subsistema
Arduino UNO	Microcontrolador/Placa	-	1	15560	Microcontrolador
ILI9341	LCD	-	1	23600	Salida
-	Capacitor	1 μF	2	190	Entrada
-	diodo	8.2 V	2	150	Entrada
-	interruptor	-	2	99	Entrada
-	led	2.4 V , 30 mA	4	140	Salida
-	Resistencia	100 Ω	12	199	Entrada y Salida
-	Resistencia	480 Ω	8	199	Entrada
-	Resistencia	50 Ω	4	199	Entrada
-	Resistencia	82 k Ω	2	199	Entrada
-	Resistencia	18 k Ω	2	199	Entrada
-	Resistencia	10 k Ω	5	199	Salida
LM741 CN	aplicador operacional	-	10	680	Entrada
NTE4050B	BUFFER	5 V	2	591	Salida

Tabla 2: Información de los componentes utilizados

2.6 Diseño de los circuito Complementarios

Los circuitos complementarios serán explicados en detalle a continuación.

2.6.1 Circuito de Entrada

El circuito de entrada se compone de 2 botones y un circuito para controlar los picos de tensión al accionar los botones para cada uno.

En la Figura 8 se muestra el esquemático del circuito de entrada, para controlar los rebotes de los botones, esta compuesto por un circuito RC debouncer [2] y un amplificador operacional en modo seguidor de tensión [3].

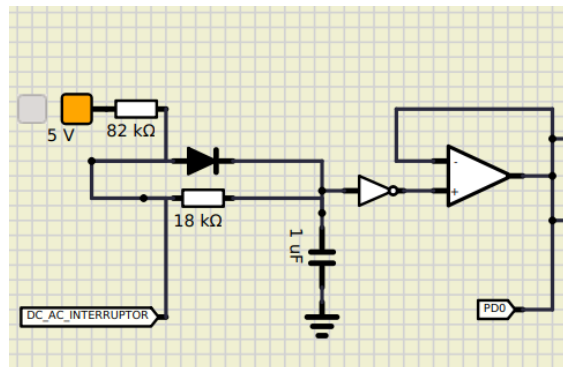


Figura 8: Circuito antirrebotes para botones [imagen propia]

El circuito para acondicionar los voltajes de entrada a los pines analógicos se encargan de convertir una tensión que puede estar entre los -24 y 24 volts a una que esté entre los 0 y 5 volts, para

esto se usan circuitos sumadores y circuitos inversores con amplificadores operacionales [3]. En la Figura 9 se muestra el esquemático del circuito.

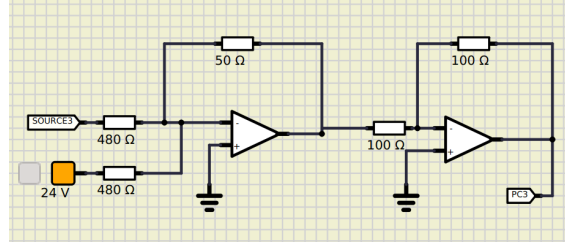


Figura 9: Circuito acondicionador de entradas analógicas [imagen propia]

Los valores precisos se consiguieron dadas las características de los valores máximos y mínimos permitidos para las fuentes $[-24, 24]$ y el rango que aceptan los pines analógicos $[0, 5]$ con estos valores límites se creo la siguiente función:

$$v_a = \frac{5}{48}(v_s + 24) \quad (3)$$

2.6.2 Circuito de Salida

El circuito de salida consta basicamente de 5 resistencias para las entradas de la pantalla LCD (valor recomendado por el fabricante) y 4 resistencias para los 4 leds indicadores, para este caso tenemos que satisfacer una tensión de entrada de 2.4 V y una corriente máxima de 30 mA. Por lo que necesitamos un valor mínimo de resistencia de:

$$R \approx \frac{5 - 2.4}{0.03} \approx 86 \Omega \quad (4)$$

Para este caso se eligieron resistencias de 100Ω y como se muestra la Figura 10 la corriente no sobrepasa el máximo especificado.

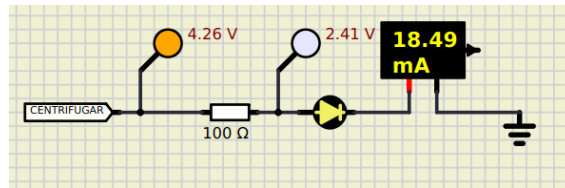


Figura 10: Medición de valores para resistencia de salida para el led normal de 100 ohms [imagen propia]

Para efectos reales siempre se debe empezar con los valores teóricos que son los seguros y luego ir calibrando según los valores experimentales. En la Figura 10 se muestra las mediciones para la resistencia de salida escogida.

3 Análisis de Resultados

En este apartado se mostrarán capturas de pantalla del funcionamiento del circuito en la simulación. Además se creó un video donde se ve el funcionamiento completo, el cual se puede consultar en esta dirección:

https://youtu.be/_wQ0oubGRaY

En la Figura 11 se muestra el funcionamiento para DC, como se puede ver en la figura tenemos valores DC en la pantalla muy cercanos a los valores DC de las fuentes, además se puede ver que los leds indicadores están encendidos pues V3 es mayor que 20 volts y V4 es menor que -20 volts.

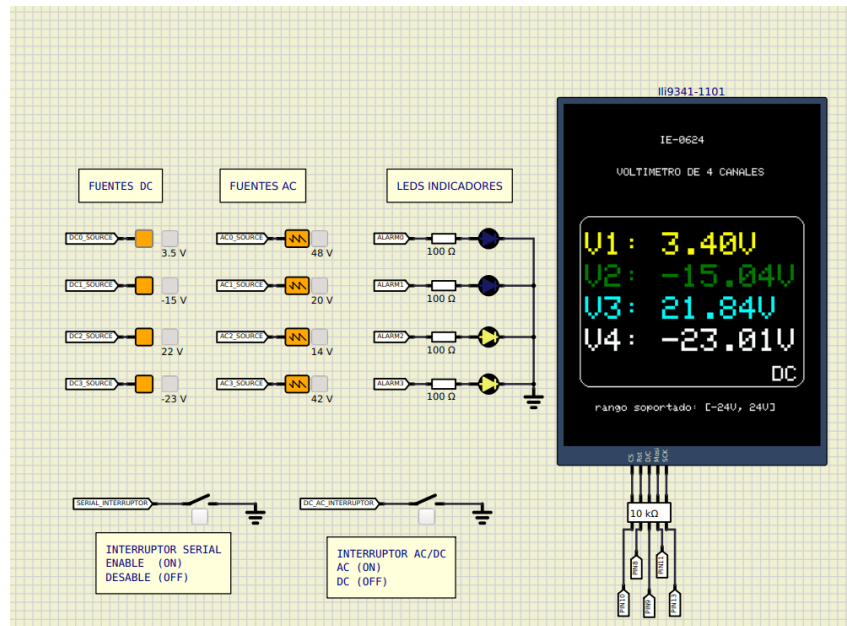


Figura 11: Prueba DC [imagen propia]

En la Figura 12 se muestra el funcionamiento del voltímetro para AC, en los osciloscopios se muestra el tipo de onda de cada fuente, donde tenemos sinusoidal, cuadrada, triangular y sierra. A partir de la imagen y usando la Tabla 1 podemos construir los resultados experimentales y compararlos con los teóricos para este caso.

Tipo	Amplitud	RMS teórico	RMS experimental	error
sinusoidal	24	16.97	16.80	1.00%
sierra	10	5.77	5.73	0.69%
triangular	7	4.04	4.02	0.50%
cuadrada	21	21	20.64	1.71%

Tabla 3: Comparación de tensiones teóricas y experimentales

Nótese que los leds de indicadores para V1 y V4 están encendidos y esto concuerda con los valores de amplitud en la Tabla 3.

Finalmente para el caso de la comunicación serial, se hizo el firmware pensando en que solo se envíe información por el puerto serial cuando la tensión de una fuente cambia, o bien cuando pasamos de AC a DC o viceversa, también se agregó una columna donde se muestra el tiempo en que el dato fue enviado a la computadora, esto en segundos (el script de python convierte los microsegundos enviados por el microcontrolador a segundos), en la Figura 13 se muestra esto. Finalmente en la

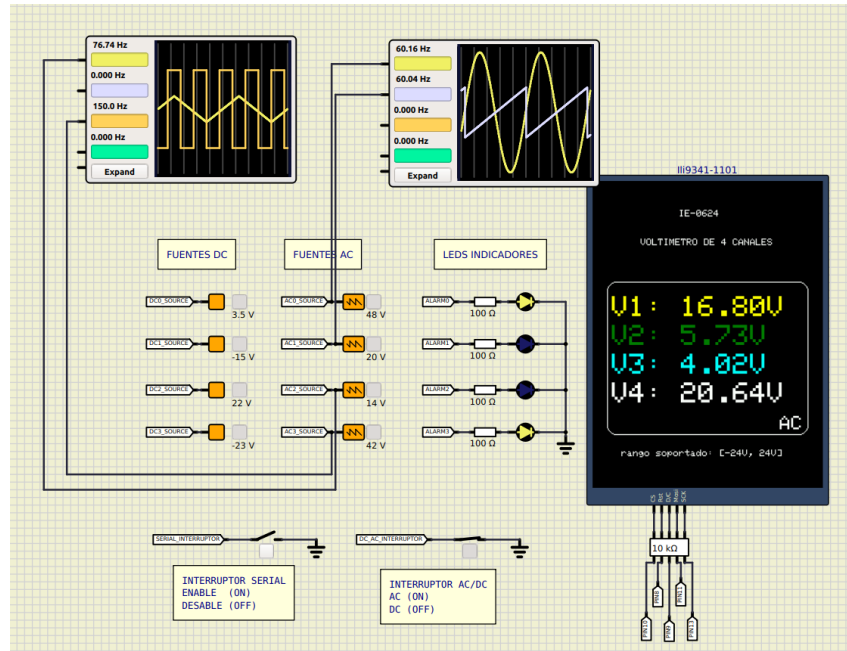


Figura 12: Prueba AC [imagen propia]

Figura 14 se muestra el archivo csv creado con los mismos datos que aparecen en la terminal de la Figura 13.

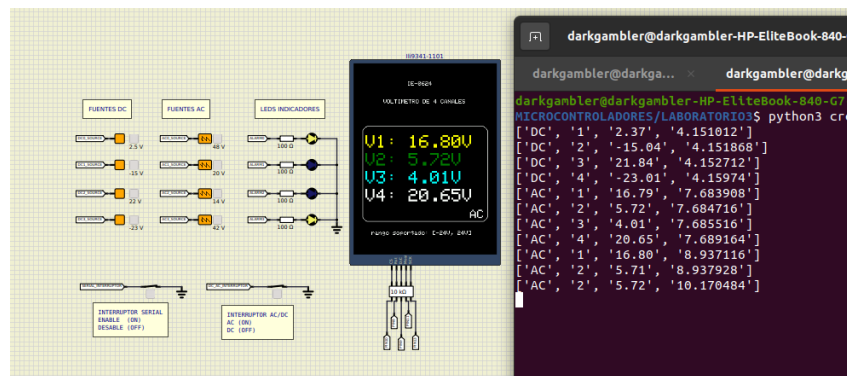


Figura 13: Prueba envio de datos seriales [imagen propia]

	A	B	C	D	
	TIPO (AC/DC)	FUENTE	TENSION (V)	TIEMPO (s)	
	DC	1	2.37	4.151012	
	DC	2	-15.04	4.151868	
	DC	3	21.84	4.152712	
	DC	4	-23.01	4.15974	
	AC	1	16.79	7.683908	
	AC	2	5.72	7.684716	
	AC	3	4.01	7.685516	
	AC	4	20.65	7.689164	
1	AC	1	16.8	8.937116	
2	AC	2	5.71	8.937928	
3	AC	2	5.72	10.170484	
4	AC	2	5.73	11.403772	
5					
6					

Figura 14: CSV con los datos enviados desde el Arduino [imagen propia]

4 Conclusiones y Recomendaciones

En el presente trabajo se utilizó el arduino UNO para realizar un voltímetro de cuatro canales, considero que se cumplieron todas las especificaciones solicitadas en el documento del laboratorio.

4.1 Recomendaciones

Es importante a medida que un proyecto es más grande tener un plan desde el principio y bien definido lo que el programa debe hacer, para esto es fundamental crear los diagramas de flujo antes de codificar. Otra recomendación importante es dividir el programa en funciones no demasiado grandes, me resultó muy provechoso combinar estas dos cosas para efectos de debugear fue mucho más sencillo que en otras ocasiones, guiarse con los diagramas y simplemente modificar las funciones que daban algún problema.

Otro detalle importante es que el simulador me ha funcionado algo extraño para este laboratorio, haciendo que las pruebas sean muy lentas (ver el video adjunto) por lo que para futuros laboratorios será bueno tener otras opciones del mismo simulador para evitar estos problemas.

Bibliografía

- [1] Atmel Corporation, *ATmega328P: 8-bit AVR Microcontroller with 4/8/16/32K Bytes In-System Programable Flash*, 2008.
- [2] J. Ganssle, *A guide to debouncing*. The Ganssle Group, 2008.
- [3] W. Jung, *Op Amp Applications Handbook*. Amsterdam: Newnes, 2006.