

Laboratorio #3: Arduino: GPIO, ADC y comunicaciones

Adrián Avilés Flores, B80835 y Steven Mora Barboza, B95109

I. INTRODUCCIÓN

En el presente informe se describe el desarrollo de un voltímetro de 4 canales basado en el microcontrolador Arduino UNO. El objetivo principal del laboratorio era desarrollar un sistema capaz de medir y visualizar voltajes en un rango de $[-24, 24]$ V DC o AC en 4 canales diferentes. Además, se requería que los datos pudieran ser enviados a una PC a través del puerto serial con el bloque USART y guardados en un archivo de texto plano. Para cumplir con los requerimientos se diseñó un circuito que permitiera condicionar los voltajes de entrada para ser manejados por el ADC del Arduino. Se incluyó también un switch para configurar el modo de medición (AC o DC) y LEDs de alarma para indicar las tensiones fuera del rango deseado. En cuanto al software, se utilizó la librería Liquid Crystal para procesar la señal de entrada y mostrar los valores en la pantalla LCD. También se incluyó código para enviar los datos a la PC a través del puerto serial con el protocolo USART. Se logró desarrollar un programa que cumplió con los requerimientos del laboratorio y se pudo visualizar y almacenar los voltajes en tiempo real en la pantalla LCD y en el archivo de texto plano en la PC, por lo que el desarrollo del voltímetro de 4 canales fue un éxito y se logró cumplir con los objetivos del laboratorio.

II. NOTA TEÓRICA

II-A. Arduino Uno (ATmega328p)

El Arduino UNO es una placa de desarrollo basada en el microcontrolador ATmega328P. Este microcontrolador es un microprocesador AVR de 8 bits desarrollado por Atmel. Este es un dispositivo con arquitectura RISC (Reduced Instruction Set Computing), lo que significa que tiene un conjunto reducido de instrucciones, pero altamente optimizadas para realizar tareas específicas. Esto le permite ser muy eficiente en términos de velocidad y consumo de energía. El ATmega328P cuenta con 32 KB de memoria flash para almacenar el programa, 2 KB de SRAM y 1 KB de EEPROM para almacenar datos. También cuenta con 14 pines de entrada/salida digitales, 6 pines de entrada analógica, una conexión USB y una conexión de alimentación de 5V. Además, el ATmega328P cuenta con un conversor analógico-digital (ADC) integrado, que permite medir tensiones en un rango de 0 a 5V. Esta función es crucial para el desarrollo del voltímetro de 4 canales que se desarrolló en este laboratorio.

5.1. Pin-out

Figure 5-1. 28-pin PDIP

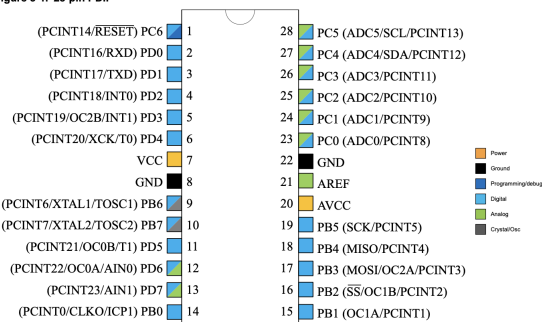


Figura 1. Diagrama de pines Arduino Uno [1].

En el caso del laboratorio, se utilizarán los pines analógicos para desarrollar el voltímetro como mencionado anteriormente. A continuación el diagrama de bloques del MCU:

Block Diagram

Figure 4-1. Block Diagram

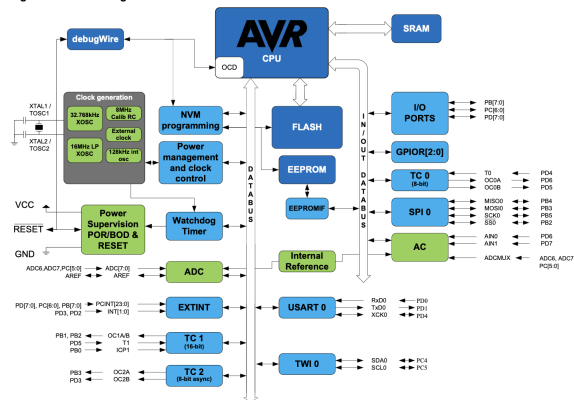


Figura 2. Diagrama de bloques del Arduino Uno [1].

Entre los componentes a usar según el diagrama de bloques, se utilizará el conector USB para programar y alimentar el Arduino, regulador de tensión, que convierte la alimentación externa de hasta 12V en una alimentación de 5V estable para el microcontrolador, puertos de entrada/salida (I/O), se utilizan para conectar sensores, actuadores y otros componentes externos al Arduino. ADC conversor analógico-digital que convierte la señal analógica en una señal digital para su procesamiento por el microcontrolador. USART: bloque de comunicación serial que permite la transmisión de datos entre el Arduino y una PC. Crystal Oscillator: proporciona una frecuencia de reloj para el microcontrolador.

32.1. Absolute Maximum Ratings

Table 32-1. Absolute Maximum Ratings

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except RESET with respect to Ground	-0.5V to $V_{CC}+0.5V$
Voltage on RESET with respect to Ground	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0mA
DC Current V_{CC} and GND Pins	200.0mA

Figura 3. Especificaciones eléctricas del ATtiny4313 [1].

Además, es importante tener en cuenta las especificaciones eléctricas mostradas en la Figura 3, donde se puede observar que la tensión máxima de operación es de 6 V y la corriente máxima por pin input/output es de 40 mA, esto se tomará en cuenta con la elaboración del circuito según los valores de resistencias y fuentes.

II-B. Periféricos utilizados

II-B1. Virtual Serial Port Tools: Virtual Serial Port Tools es un software que permite crear puertos serie virtuales en un ordenador, los cuales pueden ser usados por programas como si fueran puertos serie reales, en este caso, para hacer la comunicación entre COM1 (Arduino) y COM2 (PC)

II-C. Diseño del circuito

En la siguiente figura, se puede observar el diseño del circuito implementado para el voltímetro:

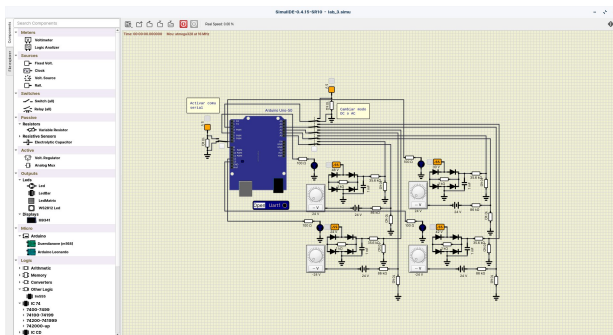


Figura 4. Esquemático del circuito completo.

Primeramente se puede apreciar el arduino uno y este posee el MCU ATmega328P en el circuito, este es el cerebro del circuito, y se encarga de ejecutar las secuencias lógicas programadas, luego, se utilizó un puente de diodos por medidor analógico debido que este protege de tensiones negativas al arduino, el puente permite que solo fluya la corriente en una dirección, limitando así la tensión de entrada a un rango seguro para el microcontrolador. De esta manera, se asegura la correcta operación y protección del dispositivo. y a continuación se hizo un divisor de tensión, compuesto por dos resistores, el cual se encarga de mantener el valor de la tensión entre 0 y 5V debido que esos son los valores máximos que permite el arduino, y la razón de por qué se utilizaron resistores de 35.6k Ω y 10k Ω , es debido a esa misma razón, al calcular el V_{out} se tiene que a una tensión máxima recibida por la fuente, este

daría un máximo de 5 V. De forma similar con la parte de medir DC del circuito, pero en este caso no existe un puente de diodos al ser una tensión continua, pero si el divisor de tensión para asegurarse mantener la tensión entre 0 y 5V de manera segura.

A continuación la lista de componentes utilizados con sus respectivos precios en dólares:

Componente	Cantidad	Precio \$
Arduino Uno	1	35
Diodo	16	3
Boton	1	5
Resistor	20	2
Capacitor	4	3

TABLA I
PRECIO DE LOS COMPONENTES

III. ANÁLISIS DE RESULTADOS

El software implementado en el arduino más el hardware añadido, permite medir la tensión ya sea AC o DC, en cuatro terminales distintas, para esto, se utilizó el switch con 5 polos, los cuales 4 son conectados a las entradas analógicas, y uno para detectar el cambio cuando se requiere usar AC o DC; de esta forma, cuando el switch es accionado, si está en alto se leerá los valores en AC y cuando está en bajo en DC, esto gracias a la forma como están conectados los distintos voltímetros. En la siguiente imagen se puede apreciar el circuito en funcionamiento:

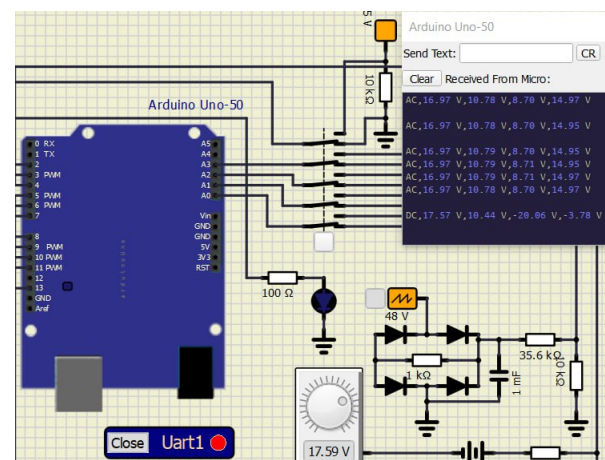


Figura 5. Obteniendo datos del voltímetro

Una vez comprobado el funcionamiento, se procede a probar la comunicación serial usando USART y la obtención de los datos en un archivo csv:

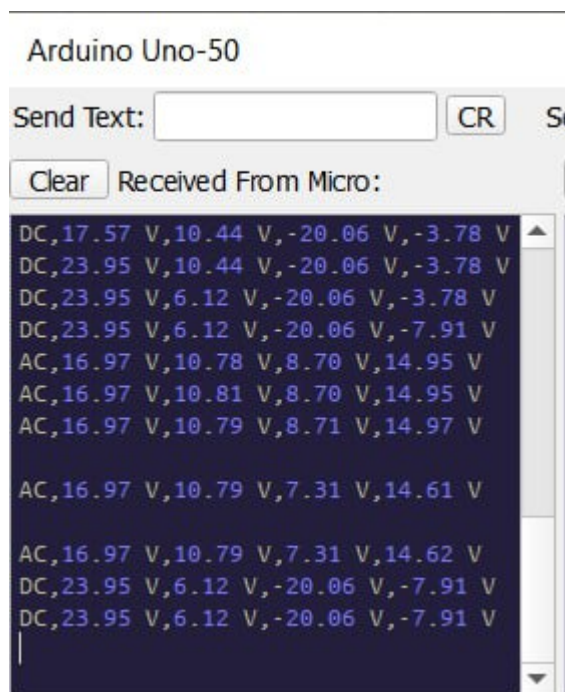


Figura 6. Datos enviados por el arduino

	A	B	C	D	E
1	Measure_mode	Channel_0	Channel_1	Channel_2	Channel_3
2	DC	17.57 V	10.44 V	-20.06 V	-3.78 V
3	DC	23.95 V	10.44 V	-20.06 V	-3.78 V
4	DC	23.95 V	6.12 V	-20.06 V	-3.78 V
5	DC	23.95 V	6.12 V	-20.06 V	-7.91 V
6	AC	16.97 V	10.78 V	8.70 V	14.95 V
7	AC	16.97 V	10.81 V	8.70 V	14.95 V
8	AC	16.97 V	10.79 V	8.71 V	14.97 V
9	AC	16.97 V	10.79 V	7.31 V	14.61 V
10	AC	16.97 V	10.79 V	7.31 V	14.62 V
11	DC	23.95 V	6.12 V	-20.06 V	-7.91 V
12	DC	23.95 V	6.12 V	-20.06 V	-7.91 V
13					

Figura 7. Datos obtenidos puestos en el archivo csv

Como se puede apreciar en la figura 6, el el voltímetro es capaz de medir tensiones en AC de hasta +24 V, a pesar de que el arduino como tal solo permite 5 V en sus puertos, además, para las mediciones en DC, se puede apreciar donde su valor máximo medido fue de 23,95 V, por lo tanto, el limite de 24 V funciona como lo esperado y el envío entre COM1 y COM2 son exitosos.

En la figura 7, se muestra como el script de python ingresa el título de cada columna para ser identificada, y en orden va registrando los valores de tensión leídos por el voltímetro.

Para observar la funcionalidad del programa, de forma clara y breve se presenta el siguiente diagrama de flujo:

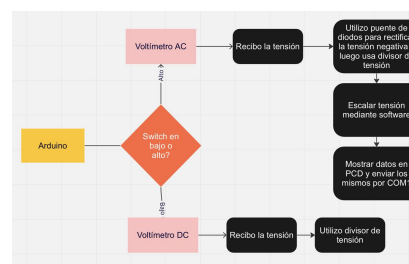


Figura 8. Diagrama de flujo del funcionamiento del programa

IV. CONCLUSIONES

- El código lee los voltajes de cuatro canales y los muestra en la pantalla LCD después de ser convertidos y escalados a valores reales.
- El código permite seleccionar entre dos modos de operación, DC y AC, a través del uso de un switch.
- El código incluye la funcionalidad de alerta que se activa mediante el encendido de cuatro LEDs cuando los voltajes en los canales superan ciertos valores límite.
- En resumen, el código desarrollado controla y visualiza la lectura de los voltajes en los canales a través de una pantalla LCD y proporciona una función de alerta para indicar cuando los voltajes superan ciertos valores límite y además, guarda los resultados en un archivo csv.

El repositorio con el desarrollo de este laboratorio se encuentra en: <https://github.com/Stevenlmb/Lab-Microcontroladores/tree/Lab3>

REFERENCIAS

- [1] Atmel. (2016) 8-bit avr microcontrollers atmega328/p. [Online]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/241077/ATMEL/ATMEGA328P.html>

Introduction

The Atmel[®] picoPower[®] ATmega328/P is a low-power CMOS 8-bit microcontroller based on the AVR[®] enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega328/P achieves throughputs close to 1MIPS per MHz. This empowers system designer to optimize the device for power consumption versus processing speed.

Feature

High Performance, Low Power Atmel[®]AVR[®] 8-Bit Microcontroller Family

- Advanced RISC Architecture
 - 131 Powerful Instructions
 - Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 32KBytes of In-System Self-Programmable Flash program Memory
 - 1KBytes EEPROM
 - 2KBytes Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data Retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Atmel[®] QTouch[®] Library Support
 - Capacitive Touch Buttons, Sliders and Wheels
 - QTouch and QMatrix[®] Acquisition
 - Up to 64 sense channels

a

- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Two Master/Slave SPI Serial Interface
 - One Programmable Serial USART
 - One Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - One On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
 - 1.8 - 5.5V
- Temperature Range:
 - -40°C to 105°C
- Speed Grade:
 - 0 - 4MHz @ 1.8 - 5.5V
 - 0 - 10MHz @ 2.7 - 5.5V
 - 0 - 20MHz @ 4.5 - 5.5V
- Power Consumption at 1MHz, 1.8V, 25°C
 - Active Mode: 0.2mA
 - Power-down Mode: 0.1µA
 - Power-save Mode: 0.75µA (Including 32kHz RTC)