

UNIVERSIDAD DE COSTA RICA

ESCUELA DE INGENIERÍA ELÉCTRICA

IE-0624: LABORATORIO DE MICROCONTROLADORES

---

# Laboratorio 3

## Arduino: PID, GPIO, ADC y comunicaciones

---

Prof. Marco Villalta

Estudiantes:  
Sofía Fonseca Muñoz, B42634

28 de enero de 2023

# Índice

<b>1. Introducción/Resumen</b>	<b>2</b>
<b>2. Nota teórica</b>	<b>3</b>
2.1. Microcontrolador . . . . .	3
2.1.1. Características eléctricas . . . . .	4
2.2. Bibliotecas . . . . .	5
2.3. Diseño . . . . .	5
2.4. Componentes electrónicos . . . . .	8
<b>3. Desarrollo/Análisis de resultados</b>	<b>9</b>
3.1. Lectura de los valores DC . . . . .	10
3.2. Lectura de valores en AC . . . . .	11
3.3. Envío de valores sensados a través de USART según la activación del switch. . . . .	12
3.4. Lectura de datos a través de script de Python. . . . .	13
<b>4. Conclusiones y recomendaciones</b>	<b>14</b>
<b>Referencias</b>	<b>15</b>
<b>5. Apéndice</b>	<b>16</b>

## 1. Introducción/Resumen

Este proyecto consiste en la creación de un voltímetro basada en Arduino UNO. El voltímetro cuenta con una pantalla LCD PCD8544 donde se muestra los resultados de las 4 posibles mediciones, teniendo la última la opción de desplegar la tensión en DC o en AC, en la cual presenta la RMS. Para poder leer tensiones entre -24V y 24 V, se utilizó un divisor de tensión que permite leer la tensión presente en los puertos sin dañar al Arduino, ya que este puede leer hasta 5V.

Además el voltímetro se conecta con la computadora a través del protocolo USART y se crea un archivo .csv que contiene la información recolectada en cada puerto.

El repositorio correspondiente a este laboratorio se puede encontrar en el enlace [https://github.com/sofifon/IE0624-2023/tree/main/Lab3\\_Voltimeter](https://github.com/sofifon/IE0624-2023/tree/main/Lab3_Voltimeter).

## 2. Nota teórica

### 2.1. Microcontrolador

Para este laboratorio se utilizó la placa Arduino UNO. Esta placa es muy popular debido a su bajo costo y su capacidad para resolver problemas en áreas educativas como industriales. Un ejemplo de ello es la capacidad de capturar información de sensores en tiempo real y enviar esta información a la computadora. [1]

La placa cuenta con un procesador ATmega328P, 14 pines de entrada y salida digital, 6 pines de entrada analógica y conexión USB. A nivel eléctrico, la placa cuenta con dos salidas fijas de 3V y 5V y con entrada de voltaje a través de un pin, el cual soporta de 6 a 20 V y a través del USB el cual por convención recibe 5.5V. Además el procesador puede correr a una frecuencia de hasta 20MHz.[1] En la siguiente imagen se puede observar el diagrama de bloques del procesador ATmega328P:

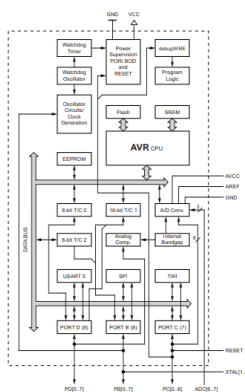


Figura 1: Diagrama de bloques del procesador ATmega328P. Tomada de [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)

Es importante destacar que este procesador se encuentra integrado en la placa Arduino UNO, por lo que no se realizan conexiones directas a los pines del microprocesador sino a los pines de la placa. La forma en que se integra el microprocesador en la placa, se puede observar en el siguiente diagrama:

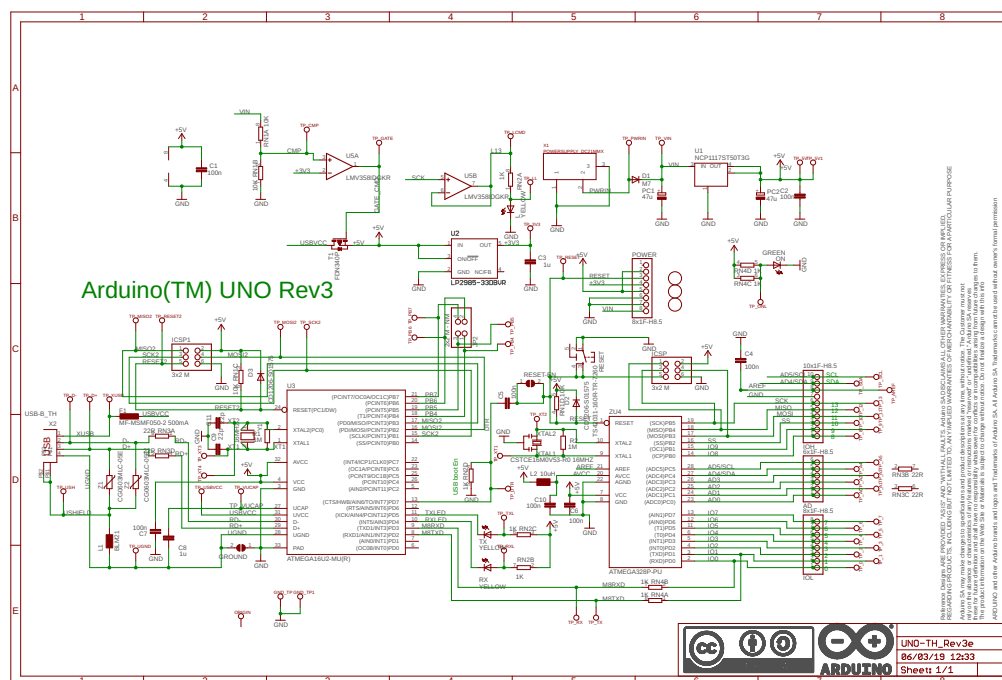


Figura 2: Diagrama de bloques de la placa Arduino UNO. Tomada de [https://content.arduino.cc/assets/UNO-TH\\_Rev3e\\_sch.pdf](https://content.arduino.cc/assets/UNO-TH_Rev3e_sch.pdf)

Por último, para simplificar el uso de la placa, la hoja del fabricante cuenta con un diagrama de pines de alto nivel que se utilizó como guía para realizar las conexiones en este laboratorio.

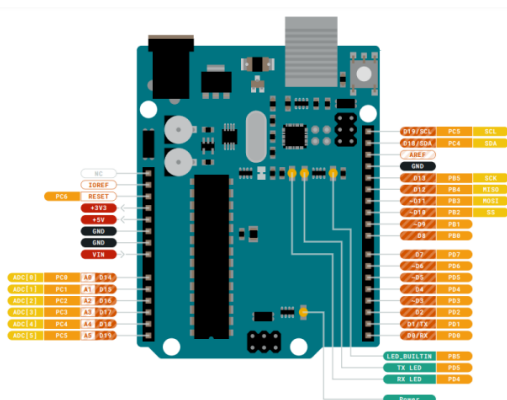


Figura 3: Conexiones de pines de la placa Arduino UNO. Tomada de <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

### 2.1.1. Características eléctricas

En la siguiente tabla se pueden encontrar las características físicas de este microcontrolador, las cuales se deben tomar en cuenta para la implemetación en la que se desea utilizar el microcontrolador.<sup>[1]</sup>

Especificaciones eléctricas	
Temperatura de operación	-40°C a 85 °C
Voltaje de entrada	6 a 20V
Voltaje entre los otros pines y Vss	0V a 5V
Máximo voltaje de operación	5.0V
Máxima corriente por cada pin	40.0 mA

## 2.2. Bibliotecas

A nivel de software, se utilizó el lenguaje propio de Arduino UNO el cual es basado en C/C++ y en esta ocasión sólo se necesitó la biblioteca **PCD8544.h** para controlar la pantalla LCD. Esta biblioteca se puede encontrar en <https://github.com/carlosefr/pcd8544>. Fue elegida porque cuenta con buena documentación de cómo conectar los pines de la pantalla para que funcione adecuadamente.

## 2.3. Diseño

El diseño completo del circuito eléctrico que se utilizó para el voltímetro se puede observar en la siguiente imagen:

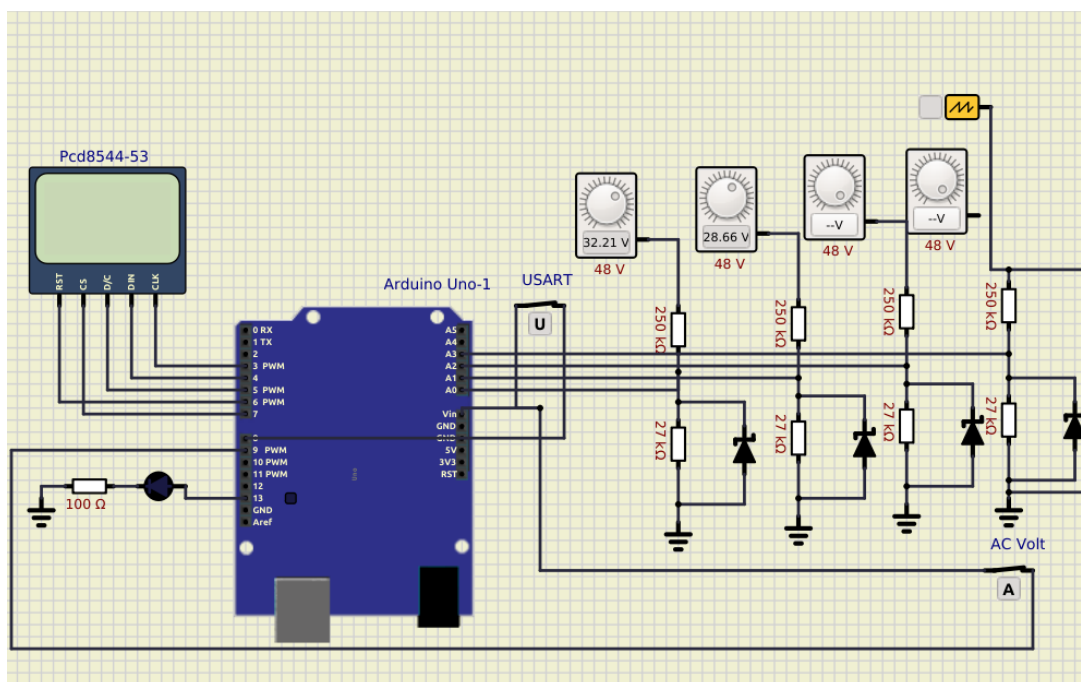


Figura 4: Diseño completo del circuito de la incubadora. Imagen propia.

De manera general, este diseño cuenta con la pantalla PCD8544 para desplegar los datos, un led para indicar si el circuito está sobrecargado, 2 switches, uno para la conexión USART y uno para activar la lectura AC y cuatro divisores de tensión. Los cuáles se encuentran conectados a fuentes de poder variables tanto de DC y AC para representar los valores que se leen por el voltímetro. A continuación se va a explicar cada uno con más detalle.

Para la pantalla, se utilizó la biblioteca PCD8544.h que sugiere conectar SCLK al **pin3**, SDIN al **pin4**, DC al **pin5**, SCE al **pin7** y RST al **pin6**. También existen otras conexiones físicas que le dan

poder a la pantalla, sin embargo a nivel de simulación estas no son necesarias. [2]

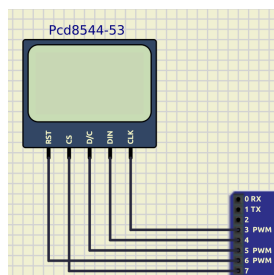


Figura 5: Conexión de la pantalla PCD8544. Imagen propia.

Para la conexión del switch se requiere conectar un lado al **pin 8** y **pin 9** que van a servir como entrada, y del otro lado a Vin, de esta manera cuando se cierra el switch, le llega los 5V que entrega este pin al pin de entrada.

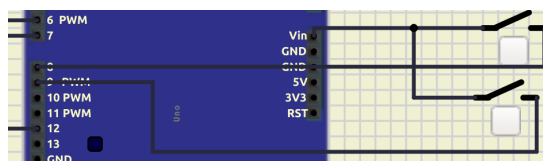


Figura 6: Conexión del switch. Imagen propia.

Para la conexión del LED se toma en cuenta la siguiente ecuación. Como el voltaje de entrega es de 5 V y la corriente máxima que pueden soportar los LEDs es de 50 mA, se utiliza una resistencia de 100  $\Omega$ .

$$V = IR \Rightarrow R = \frac{V}{I} = \frac{5V}{50mA} = 100 \Omega \quad (1)$$

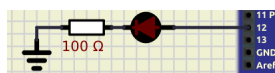


Figura 7: Conexión del LED. Imagen propia.

El Arduino sólo soporta 5 V como entrada en cada pin. Para poder leer las tensiones que son superiores a este valor, se utiliza un divisor de tensión resistivo. Este es una configuración de un circuito eléctrico que permite repartir la tensión entre varios puntos de conexión en serie. La tensión de entrada se conoce como  $V_{in}$  y la tensión de salida que nos interesa como  $V_{out}$  y se conecta como se muestra en la imagen a continuación. [3]

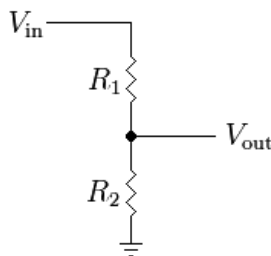


Figura 8: Conexión de un divisor de tensión. Imagen propia.

La idea del divisor es encontrar las resistencias adecuadas para que la salida tenga un valor de tensión adecuado para el Arduino. La ecuación que define a un divisor de tensión es:

$$V_{out} = \frac{R_2}{R_2 + R_1} V_{in} \quad (2)$$

Se buscan valores de resistencias para que la tensión sea 5 V en la salida, que sería el valor que lee la resistencia, tomando en cuenta que la entrada puede llegar a 48 V que es la diferencia de tensión entre -24 V y 24 V. Además, se escogen valores de resistencia bastante altos para que la corriente se mantenga dentro de los 40 mA que soporta el pin del microcontrolador. Tomando estos aspectos en cuenta se tiene  $R_1 = 250k \Omega$  y  $R_2 = 27k \Omega$ . Adicionalmente se agrega un diodo zenner que evita que tensiones negativas lleguen al Arduino y lo dañen. Por lo que el circuito queda así:

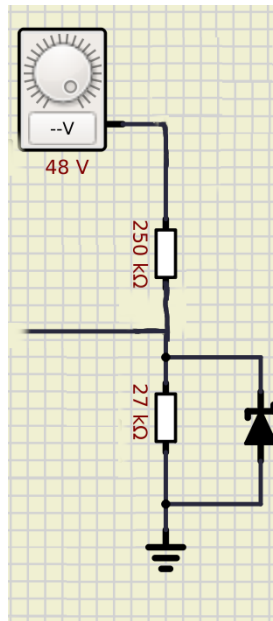


Figura 9: Conexión del divisor de tensión para lectura de tensiones. Imagen propia.

En resumen, las conexiones presentes en el circuito se pueden ver de la siguiente manera:



Pin	Config	Elemento	Funcion
PD3	Salida	PCD8544	Desplegar los datos de temperatura y humedad
PD4	Salida		
PD5	Salida		
PD6	Salida		
PD7	Salida		
PD8	Entrada	Switch	Controlar el envío de datos por medio de USARTS.
PD9	Entrada	Switch	Controlar la señal de tensión para la lectura AC.
PD13	Salida	LED	Indicar si el valor de tensión es menor a -20 V o mayor a 20 V.
A0	Entrada	Lector de voltaje 1	Lee un valor de tensión entre -24V y 24V.
A1	Entrada	Lector de voltaje 1	Lee un valor de tensión entre -24V y 24V.
A2	Entrada	Lector de voltaje 1	Lee un valor de tensión entre -24V y 24V.
A3	Entrada	Lector de voltaje 1	Lee un valor de tensión entre -24V y 24V. Incluyendo AC

## 2.4. Componentes electrónicos

Para la creación de la incubadora, se utilizó la lista de componentes que se encuentra en la siguiente tabla. Los precios de cada componente fueron tomados de <https://www.crcibernetica.com/> tomando en cuenta un tipo de cambio de 595 colones equivalente a 1 USD. Además los precios de las resistencias y los LEDs se calcularon a partir de paquetes grandes dividiendo su valor entre la cantidad, ya que por su valor no se consiguen individualmente.

Componente	Precio (colones)
PCD8544	990
2 x Switch	1130
2 x Resistor 100 ohm	65
LED	20
4 x Resistor 250k	2975
4 x Resistor 27k	890
Diodo Zener	595
Resistor 100 ohm	30
Arduino UNO	16300
<b>Precio total</b>	<b>22995</b>

El precio total de todos los componentes necesarios para construir el voltímetro sería 22995 colones.

### 3. Desarrollo/Análisis de resultados

Existe una serie de requerimientos que se necesitan para tener un funcionamiento adecuado del voltímetro y que se van a analizar a continuación:

1. Se leen los valores de cada uno de los pines analógicos conectados como lectores del voltímetro.
2. Se convierten esos valores a valores equivalentes que se desean ver como los del valor de entrada del divisor de tensión.
3. Si alguna de las tensiones es menor a -20 o mayor a 20 v se enciende el LED.
4. Existe un switch que se encarga de convertir la lectura de datos DC a AC, incluyendo la conversioón neceraria para recibir el dato RMS.
5. Si el switch está cerrado, se envían los valores leídos a la computadora través de USARTs.
6. La computadora crea un archivo .csv con los valores de temperatura y humedad leídos.

En el enlace <https://youtu.be/LkkVH1LVfaU> se puede ver una demostración de la simulación completa en SIMULIDE del voltímetro. Sin embargo, a continuación se hará una explicación del diseño escogido para cada uno de los componentes del voltímetro.

### 3.1. Lectura de los valores DC

Con la escogencia de las resistencias para que fueran valores completos y valor reales, la tensión de salida del divisor de tensión no llega a los 5 V si no que llega a 4.76 V. Por lo que si se quiere tener una lectura real del valor leído, al hacer la conversión no se puede tomar en cuenta los 5 V porque sino no se leerán el máximo de los 24 V.

El valor máximo que lee el pin es de 5 V, sin embargo ese valor no lo lee como 5 V sino como el valor analógico en bit. que el Arduino puede interpretar. Por lo que en realidad el pin va dar un valor máximo de 1023 en la lectura. Ahora, si el divisor de tensión llegara a ese valor, se tomara ese valor para hacer una conversión a los 48 V ( que van desde -24 V a 24 V. Pero como no es así, se toma el valor máximo real, es decir, los 4.76 V como el equivalente a los 24 V máximos. En la línea de 0 a 1023 ese valor sería 956. La conversión en el código entonces queda como:

```
map(analogRead(volt0Pin), 0, 956, -24, 24);
```

Esto se hace tomando en cuenta que volt0 es la lectura del pin analógico 0. El mismo proceso se hace para los 3 siguientes pines. Todos estos valores a su vez son procesados y como resultado se muestran en la pantalla como en la imagen a continuación:

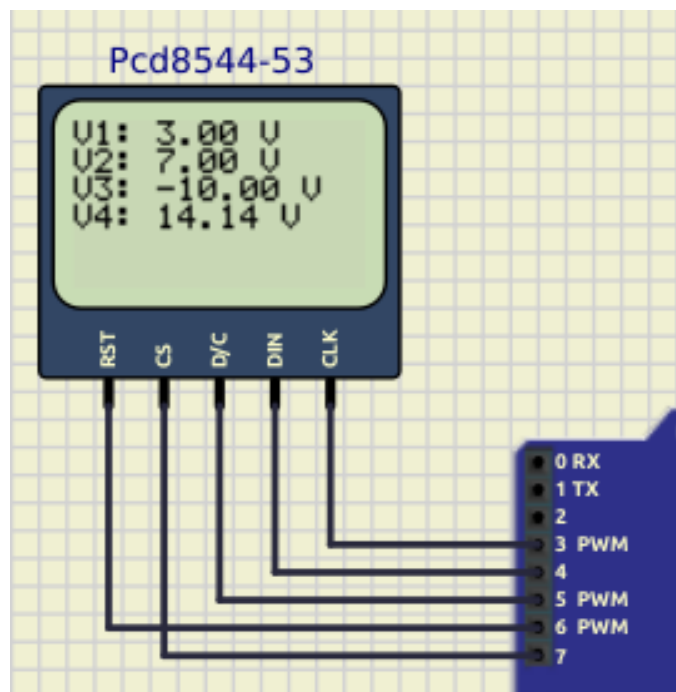


Figura 10: Funcionamiento de la pantalla LED. Imagen propia.

### 3.2. Lectura de valores en AC

Si bien es cierto los principios de conversión utilizados en la parte DC funcionan también para la parte AC, existen pasos adicionales. Lo primero es tomar en cuenta que la transmisión de datos se da a través de una onda que tiene una frecuencia muy alta. Si se analiza una onda sinusoidal, el valor máximo define el valor real que tiene la onda cuando está centrada en cero.

La primera parte para la lectura del valor AC es hacer una lectura de 100 tomas que se realizan cada 200 ms. Esto se asegura que haya leído una onda completa al tomar en consideración la frecuencia de la onda. DE ahí se toma valor más alto de ellas. Este valor se puede tomar como el valor pico de la onda. Ese valor, sigue leyéndose en rangos de 0 a 1023 como en el caso anterior por lo que hay que hacer la conversión. Una vez que la conversión está hecha se tiene el valor pico de la tensión AC. Para obtener el valor RMS se divide entre  $\sqrt{2}$ . En la siguiente imagen se muestran dos valores dados en valor pico de la onda como entrada y los valores que dio el voltímetro como resultado, los cuales corresponden adecuadamente al valor rms de cada uno de ellos.

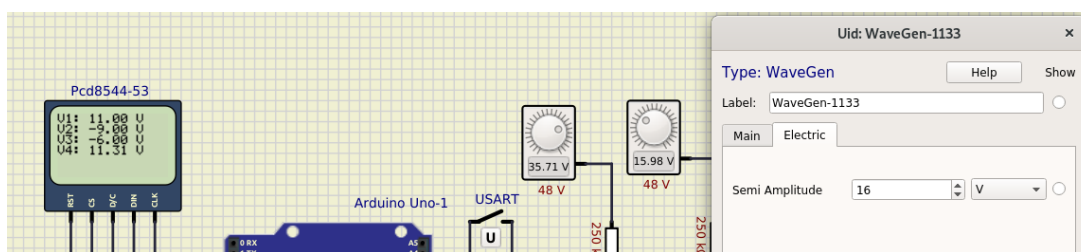


Figura 11: Señal sinusoidal de 16 V.

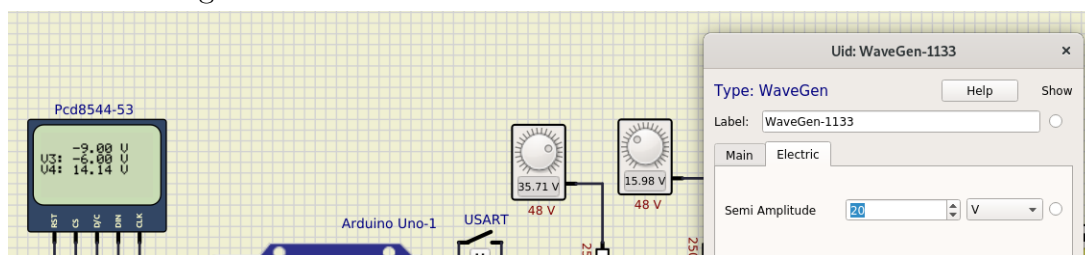


Figura 12: Señal sinusoidal de 20 V.

Figura 13: Funcionamiento de la conversión. Figura propia.

### 3.3. Envío de valores sensados a través de USART según la activación del switch.

En las siguientes figuras se muestran el funcionamiento del switch para la transmisión de datos a través del puerto serial. La primera imagen muestra la transmisión de valores cuando el switch se encuentra en alto y la segunda imagen muestra cómo no se transmiten datos cuando el switch está en bajo. Estos valores se envían sin ninguna unidad o valor extra para disminuir la cantidad de datos que se envían pero los datos son trabajados al ser leídos por el script.

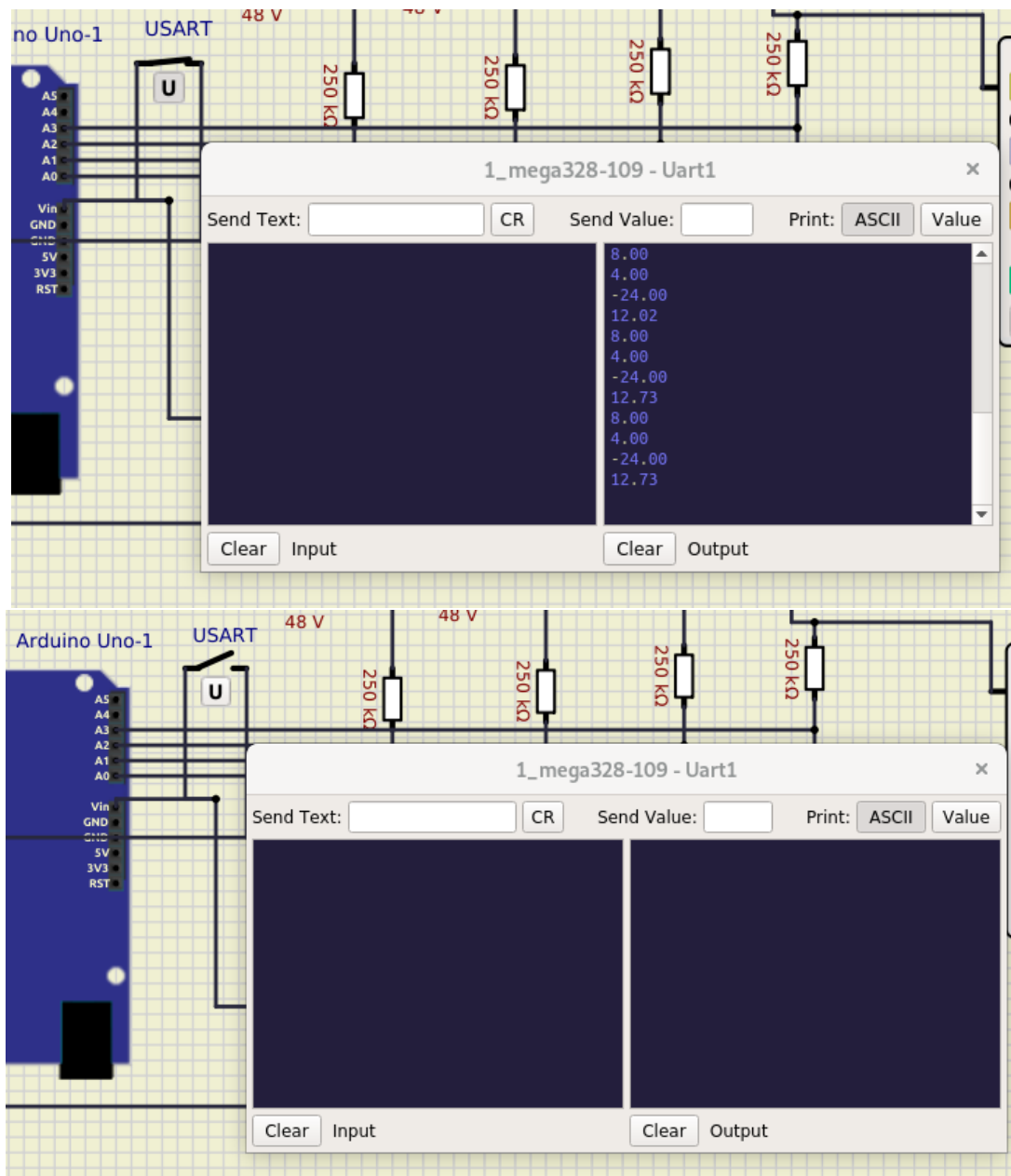


Figura 14: Funcionamiento del switch para la transmisión de datos.

### 3.4. Lectura de datos a través de script de Python.

En la siguiente figura se muestran los datos recibidos por el script de Python a través del puerto Serial y desplegados en el documento .csv. Para esto se incluyó solamente los números para disminuir el tránsito de datos y se agregan las indicaciones y unidades de medir después de recibidos los datos.

	A	B	C	D	E	F	G	H
1	V1:	8	V2:	-24	V3:	-13	V4:	11.31
2	V1:	8	V2:	10	V3:	-13	V4:	11.31
3	V1:	8	V2:	10	V3:	-13	V4:	11.31
4	V1:	8	V2:	10	V3:	-13	V4:	11.31
5	V1:	8	V2:	-16	V3:	-13	V4:	16.97
6	V1:	8	V2:	-16	V3:	-13	V4:	16.97
7	V1:	8	V2:	-16	V3:	-13	V4:	16.97
8	V1:	8	V2:	-16	V3:	-13	V4:	16.97
9	V1:	3	V2:	-16	V3:	-13	V4:	16.97
10	V1:	3	V2:	-16	V3:	-13	V4:	16.97
11	V1:	3	V2:	10	V3:	-13	V4:	16.97
12	V1:	3	V2:	10	V3:	-13	V4:	16.97
13	V1:	3	V2:	10	V3:	-13	V4:	12.73
14	V1:	3	V2:	10	V3:	-13	V4:	12.73
15	V1:	3	V2:	10	V3:	7	V4:	12.73
16	V1:	3	V2:	10	V3:	7	V4:	12.73
17	V1:	3	V2:	10	V3:	7	V4:	12.73
18	V1:	3	V2:	10	V3:	7	V4:	12.73

Figura 15: Archivo .csv generado por el script de Python. Imagen propia.

## 4. Conclusiones y recomendaciones

- El arduino UNO facilita el uso del microcontrolador ATmega328P, ya que al estar integrado, el usuario no tiene que enfocarse en darle la alimentación y los pines son mucho más fáciles de acceder.
- El protocolo USART permite que sólo un dispositivo esté escribiendo a un canal, sin embargo se puede realizar la lectura desde varios dispositivos. Si se quiere escribir desde varios, se recomienda utilizar un temporizador que permite acceder al canal en diferentes momentos.
- Los divisores de tensión permiten manejar tensiones muy altas en dispositivos de baja tensión como lo son microcontroladores.
- Los pines digitales del Arduino se pueden utilizar como entradas y salidas pero los analógicos sólo como entradas.
- Realizar aplicaciones a través del Arduino puede traer soluciones de bajo costo a distintos problemas, ya que como se puede ver, con un valor precio muy bajo se puede obtener un voltímetro bastante exacto.

## Referencias

1. Arduino (2022) Arduino Uno R3. Recuperado de: <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>
2. Rodrigues, C. (2020) PCD8544. Recuperado de: <https://github.com/carlosefr/pcd8544>
3. Khan Academy. Divisor de tensión. Recuperado de: <https://es.khanacademy.org/science/electrical-engineering/ee-circuit-analysis-topic/ee-resistor-circuits/a/ee-voltage-divider>



## 5. Apéndice



## Description

The Arduino UNO R3 is the perfect board to get familiar with electronics and coding. This versatile microcontroller is equipped with the well-known ATmega328P and the ATmega 16U2 Processor. This board will give you a great first experience within the world of Arduino.

## Target areas:

Maker, introduction, industries



## Features

- **ATMega328P Processor**
  - **Memory**
    - AVR CPU at up to 16 MHz
    - 32KB Flash
    - 2KB SRAM
    - 1KB EEPROM
  - **Security**
    - Power On Reset (POR)
    - Brown Out Detection (BOD)
  - **Peripherals**
    - 2x 8-bit Timer/Counter with a dedicated period register and compare channels
    - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
    - 1x USART with fractional baud rate generator and start-of-frame detection
    - 1x controller/peripheral Serial Peripheral Interface (SPI)
    - 1x Dual mode controller/peripheral I2C
    - 1x Analog Comparator (AC) with a scalable reference input
    - Watchdog Timer with separate on-chip oscillator
    - Six PWM channels
    - Interrupt and wake-up on pin change
- **ATMega16U2 Processor**
  - 8-bit AVR® RISC-based microcontroller
- **Memory**
  - 16 KB ISP Flash
  - 512B EEPROM
  - 512B SRAM
  - debugWIRE interface for on-chip debugging and programming
- **Power**
  - 2.7-5.5 volts



## 1 The Board

### 1.1 Application Examples

The UNO board is the flagship product of Arduino. Regardless if you are new to the world of electronics or will use the UNO as a tool for education purposes or industry-related tasks.

**First entry to electronics:** If this is your first project within coding and electronics, get started with our most used and documented board; Arduino UNO. It is equipped with the well-known ATmega328P processor, 14 digital input/output pins, 6 analog inputs, USB connections, ICSP header and reset button. This board includes everything you will need for a great first experience with Arduino.

**Industry-standard development board:** Using the Arduino UNO board in industries, there are a range of companies using the UNO board as the brain for their PLC's.

**Education purposes:** Although the UNO board has been with us for about ten years, it is still widely used for various education purposes and scientific projects. The board's high standard and top quality performance makes it a great resource to capture real time from sensors and to trigger complex laboratory equipment to mention a few examples.

### 1.2 Related Products

- Starter Kit
- Tinkerkit Braccio Robot
- Example

## 2 Ratings

### 2.1 Recommended Operating Conditions

Symbol	Description	Min	Max
	Conservative thermal limits for the whole board:	-40 °C (-40°F)	85 °C ( 185°F)

**NOTE:** In extreme temperatures, EEPROM, voltage regulator, and the crystal oscillator, might not work as expected due to the extreme temperature conditions



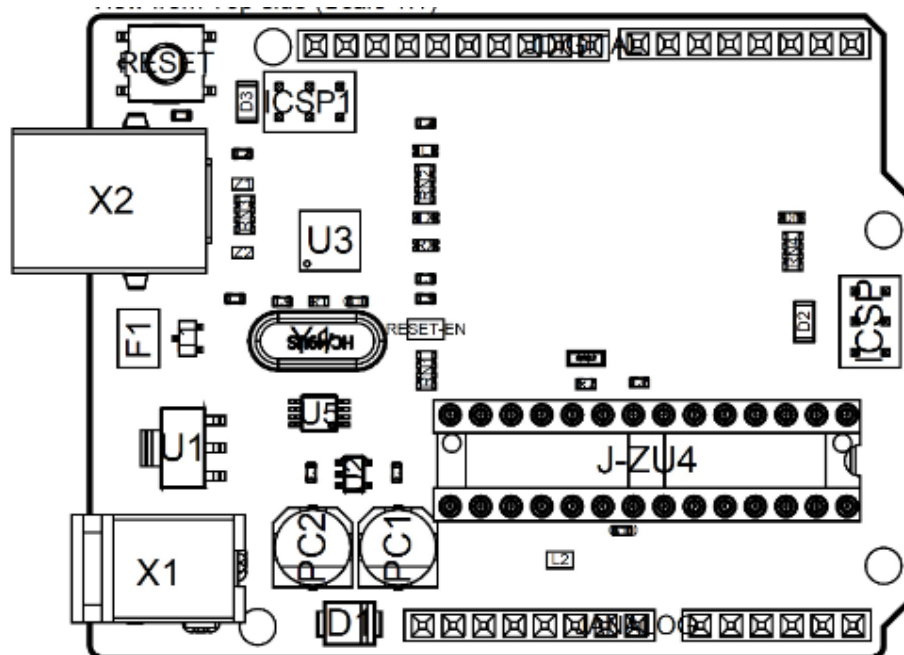
## 2.2 Power Consumption

Symbol	Description	Min	Typ	Max	Unit
VINMax	Maximum input voltage from VIN pad	6	-	20	V
VUSBMax	Maximum input voltage from USB connector		-	5.5	V
PMax	Maximum Power Consumption	-	-	xx	mA

## 3 Functional Overview

### 3.1 Board Topology

Top view



Board topology

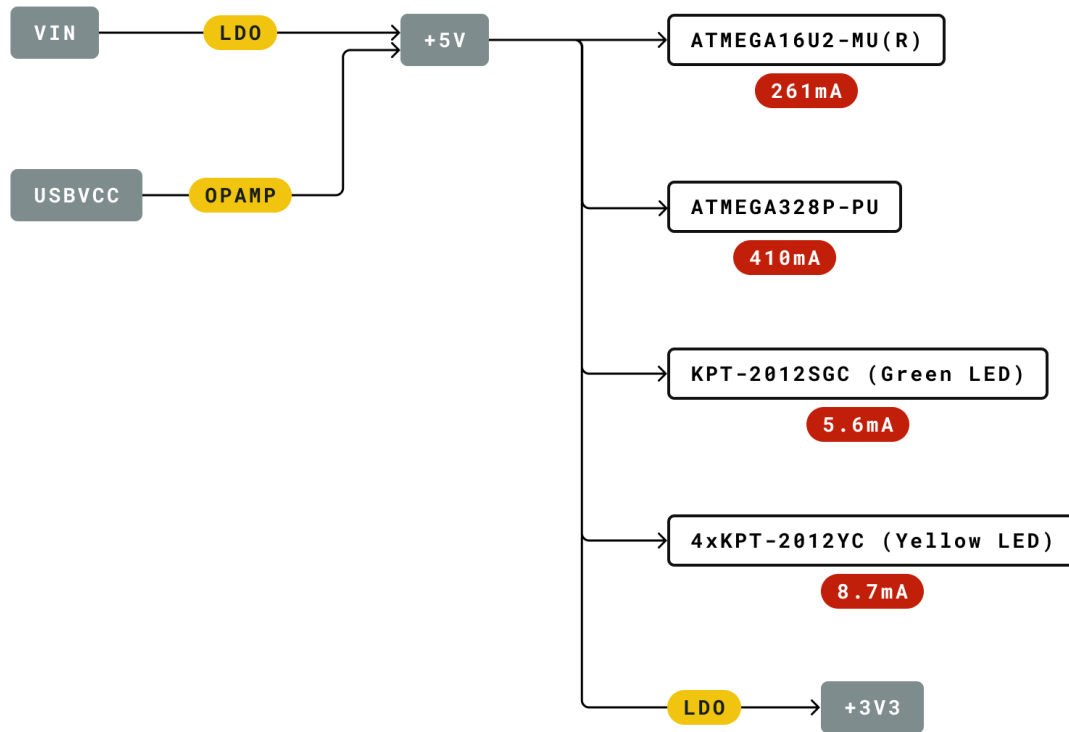
Ref.	Description	Ref.	Description
X1	Power jack 2.1x5.5mm	U1	SPX1117M3-L-5 Regulator
X2	USB B Connector	U3	ATMEGA16U2 Module
PC1	EEE-1EA470WP 25V SMD Capacitor	U5	LMV358LIST-A.9 IC
PC2	EEE-1EA470WP 25V SMD Capacitor	F1	Chip Capacitor, High Density
D1	CGRA4007-G Rectifier	ICSP	Pin header connector (through hole 6)
J-ZU4	ATMEGA328P Module	ICSP1	Pin header connector (through hole 6)
Y1	ECS-160-20-4X-DU Oscillator		



### 3.2 Processor

The Main Processor is a ATmega328P running at up to 20 MHz. Most of its pins are connected to the external headers, however some are reserved for internal communication with the USB Bridge coprocessor.

### 3.3 Power Tree



#### Legend:

- |           |             |                 |
|-----------|-------------|-----------------|
| Component | Power I/O   | Conversion Type |
|           | Max Current | Voltage Range   |

Power tree



## 4 Board Operation

### 4.1 Getting Started - IDE

If you want to program your Arduino UNO while offline you need to install the Arduino Desktop IDE [1] To connect the Arduino UNO to your computer, you'll need a Micro-B USB cable. This also provides power to the board, as indicated by the LED.

### 4.2 Getting Started - Arduino Web Editor

All Arduino boards, including this one, work out-of-the-box on the Arduino Web Editor [2], by just installing a simple plugin.

The Arduino Web Editor is hosted online, therefore it will always be up-to-date with the latest features and support for all boards. Follow [3] to start coding on the browser and upload your sketches onto your board.

### 4.3 Getting Started - Arduino IoT Cloud

All Arduino IoT enabled products are supported on Arduino IoT Cloud which allows you to Log, graph and analyze sensor data, trigger events, and automate your home or business.

### 4.4 Sample Sketches

Sample sketches for the Arduino XXX can be found either in the "Examples" menu in the Arduino IDE or in the "Documentation" section of the Arduino Pro website [4]

### 4.5 Online Resources

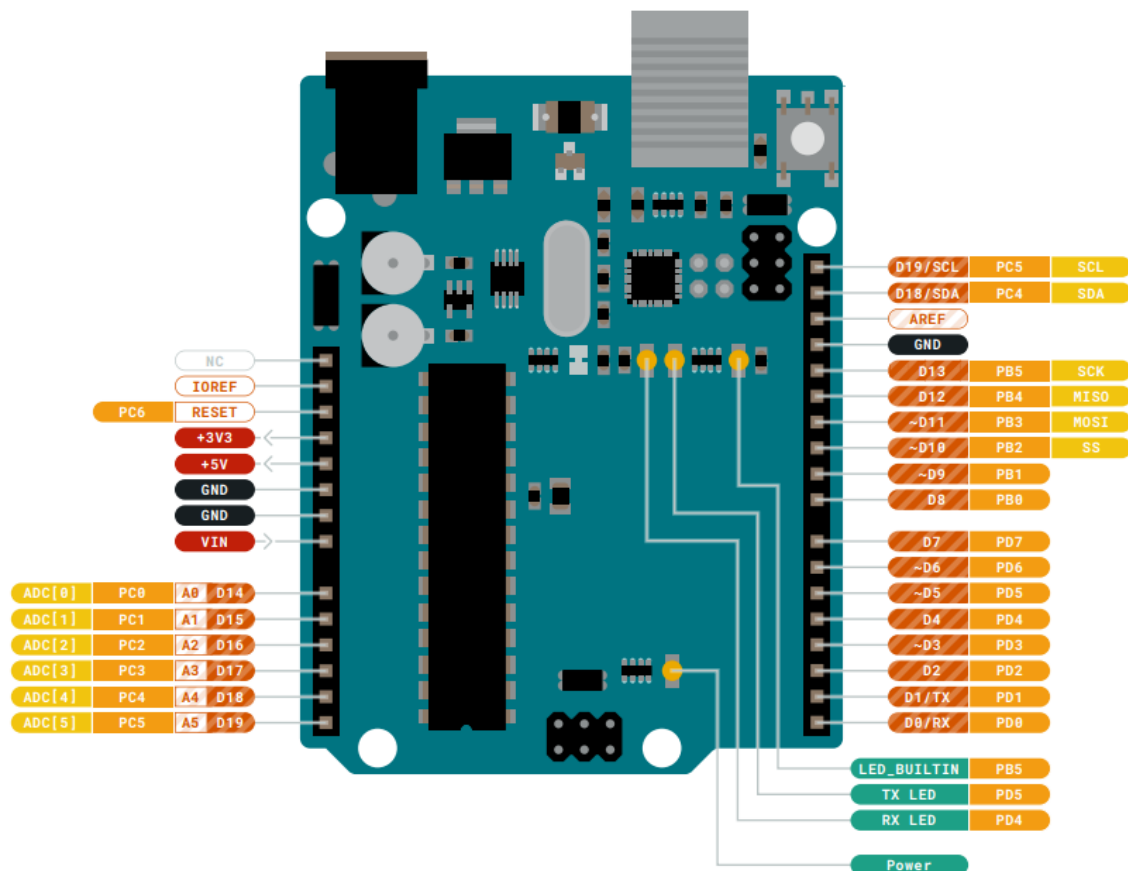
Now that you have gone through the basics of what you can do with the board you can explore the endless possibilities it provides by checking exciting projects on ProjectHub [5], the Arduino Library Reference [6] and the online store [7] where you will be able to complement your board with sensors, actuators and more



## 4.6 Board Recovery

All Arduino boards have a built-in bootloader which allows flashing the board via USB. In case a sketch locks up the processor and the board is not reachable anymore via USB it is possible to enter bootloader mode by double-tapping the reset button right after power up.

## 5 Connector Pinouts



Pinout





## 5.1 JANALOG

Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

## 5.2 JDIGITAL

Pin	Function	Type	Description
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	SPI1 Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)