

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

IE0624 – Laboratorio de Microcontroladores

II ciclo 2024

Laboratorio # 3

Arduino: GPIO,ADC y comunicaciones

Leonardo Serrano Arias C17484

Lorena Solís Extteny B97657

Profesor: Marco Villalta

6 de Octubre, 2024

# Índice

<b>Introducción</b>	<b>2</b>
<b>Nota teórica</b>	<b>2</b>
Arduino UNO . . . . .	2
GPIOs (General Purpose Input/Output) . . . . .	3
Convertidor ADC (Analog-to-Digital Converter) . . . . .	3
Protocolos de comunicación . . . . .	4
UART (Universal Asynchronous Receiver-Transmitter) . . . . .	4
Estructura del Paquete de Datos en UART . . . . .	4
LCD PCD8544 . . . . .	4
Diagrama de bloques del controlador LCD PCD8544 . . . . .	5
SPI en PCD8544 . . . . .	5
Hardware . . . . .	5
LEDs . . . . .	5
Diseño del Voltímetro y Conversión de Voltajes . . . . .	6
Selección del Circuito . . . . .	6
Aplicación del Teorema de Superposición . . . . .	6
Resolución de Ecuaciones . . . . .	6
Tabla de componentes electrónicos . . . . .	7
<b>Desarrollo</b>	<b>7</b>
Diagrama de flujo . . . . .	7
Funcionamiento del circuito . . . . .	8
Análisis electrónico . . . . .	12
<b>Conclusiones y recomendaciones</b>	<b>13</b>
<b>Apéndices</b>	<b>14</b>

## Índice de figuras

1.	Diagrama de pines del Arduino UNO [1] . . . . .	2
2.	Distribución de la corriente[1] . . . . .	3
3.	Estructura de datos de UART[3] . . . . .	4
4.	Diagrama de bloques PCD8544[2] . . . . .	5
5.	Diagrama de flujo del voltímetro . . . . .	7
6.	Circuito total para el funcionamiento del voltímetro . . . . .	8
7.	Fuentes de voltaje ajustables . . . . .	9
8.	LED de alarma en caso de voltajes fuera de los márgenes deseados . . . . .	10
9.	Valores medidos en modo AC . . . . .	10
10.	Se enciende el bloque USART y se cierra el switch que controla la transmisión serial . . . . .	11
11.	Datos mostrados en la terminal . . . . .	11
12.	Datos guardados en un csv . . . . .	12
13.	Circuito con voltímetros . . . . .	12

# Introducción

Para el desarrollo de este laboratorio se utiliza el microcontrolador Arduino UNO para elaborar el diseño de un voltímetro que sea capaz de leer tensiones desde -24 a 24. Particularmente, se va a trabajar con los GPIOs, ADC y protocolos de comunicación USART. Adicionalmente, se implementará una interfaz de usuario que mostrará los valores de voltaje en una pantalla LCD y permitirá la transmisión de datos a una PC a través del puerto serial, almacenando estos datos en un archivo CSV. El repositorio utilizado para el desarrollo de este laboratorio se encuentra en la siguiente dirección: [https://github.com/Leonardo-SA/IE-0624\\_II\\_2024\\_C17484\\_B97657.git](https://github.com/Leonardo-SA/IE-0624_II_2024_C17484_B97657.git)

## Nota teórica

### Arduino UNO

El Arduino UNO es uno de los microcontroladores más emplados para desarrollar proyectos, debido a su facilidad de uso y versatilidad. Este dispositivo cuenta con un **microprocesador ATmega328P** [1] y dispone de varios pines de entrada/salida digitales y analógicas. Entre sus principales características, destacan:

- 14 pines digitales de entrada/salida (6 pueden ser usados como salidas PWM).
- 6 entradas analógicas para leer voltajes entre 0 y 5V, con una resolución de 10 bits.
- Comunicación serial (UART), SPI e I2C para transmisión de datos.
- Memoria flash de 32 KB para almacenamiento de código.
- Alimentación a través de USB o de una fuente externa (hasta 12V).

El entorno de desarrollo **Arduino IDE** permite la programación en un lenguaje basado en C/C++, lo que facilita su integración en proyectos como el voltímetro de este laboratorio.

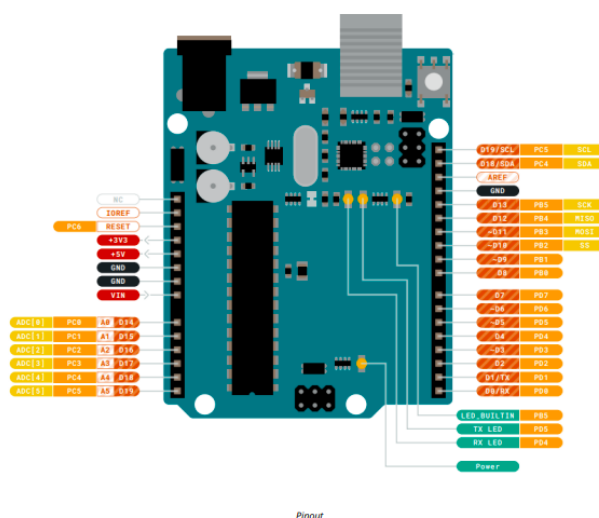


Figura 1: Diagrama de pines del Arduino UNO [1]

Además, este tiene la alimentación en el puerto de entrada  $V_{IN}$ . De acuerdo a la hoja de datos del microcontrolador la distribución de la corriente corresponde a la siguiente:

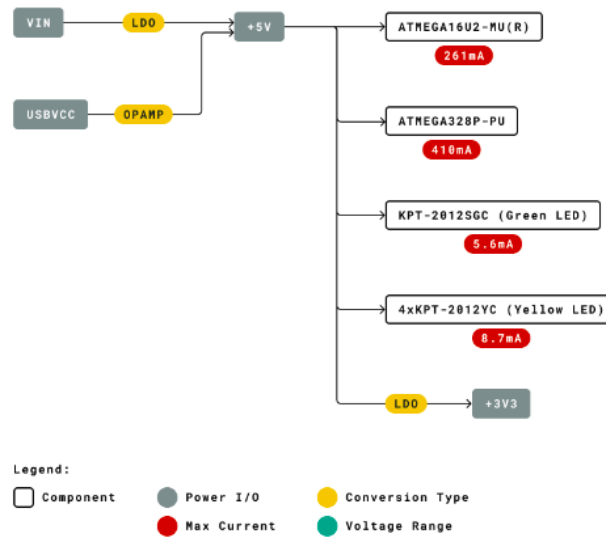


Figura 2: Distribución de la corriente[1]

Finalmente, se tienen las características del arduino que contienen los valores máximos y mínimos y el máximo consumo de potencia, como se observa en la siguiente tabla:

Símbolo	Descripción	Mín	Típ	Máx	Unidad
<b>VINMax</b>	Voltaje máximo de entrada desde el pin VIN	6	-	20	V
<b>VUSBMax</b>	Voltaje máximo de entrada desde el conector USB	-	-	5.5	V
<b>PMax</b>	Consumo máximo de potencia	-	-	xx	mA

Tabla 1: Tabla que muestra los límites de voltaje de entrada y consumo de potencia.

## GPIOs (General Purpose Input/Output)

Los pines GPIO son utilizados para la interacción con dispositivos externos, ya sea para recibir señales o para enviar instrucciones a través de ellos. Los GPIOs pueden configurarse como entradas o salidas mediante las siguientes funciones:

- `pinMode(pin, INPUT/OUTPUT)`: Define si el pin funciona como entrada o salida.
- `digitalRead(pin)`: Lee el estado de un pin configurado como entrada.
- `digitalWrite(pin, HIGH/LOW)`: Establece un valor lógico alto o bajo en un pin de salida.

En este laboratorio, los GPIOs se utilizan para controlar los LEDs de advertencia y para leer el estado de los interruptores que permiten cambiar entre los modos AC y DC.

## Convertidor ADC (Analog-to-Digital Converter)

Un convertidor analógico-digital (ADC) se encarga de transformar una señal de tensión en bits, permitiendo que el microcontrolador, en este caso el Arduino, procese estas lecturas. En el Arduino, los pines de entrada analógica pueden medir tensiones en el rango de 0V a 5V. La resolución del ADC del Arduino es de 10 bits, lo que significa que puede representar hasta

$$2^{10} = 1024$$

niveles de voltaje diferentes. Esto permite que el voltaje de entrada se convierta en un número digital que varía entre 0 y 1023, donde 0 representa 0V y 1023 representa 5V. Así, cada incremento de 1 en el valor digital corresponde aproximadamente a un incremento de 4.88 mV en el voltaje de entrada.

## Protocolos de comunicación

### UART (Universal Asynchronous Receiver-Transmitter)

UART es un protocolo de comunicación serial que se utiliza para transmitir datos entre el Arduino y otros dispositivos [3]. En este laboratorio, se emplea para enviar los valores de voltaje medidos por el Arduino a la computadora. El protocolo UART divide los datos en bytes, que se envían bit a bit a través de los pines RX y TX del Arduino. Los datos son enviados en formato CSV para su almacenamiento en la PC con ayuda de Virtual Serial Port Tool y Python. A diferencia de los protocolos síncronos, en UART no es necesario compartir una línea de reloj entre los dispositivos. En su lugar, se establece un acuerdo previo entre los dispositivos sobre la velocidad de transmisión, conocida como baud rate, que define la cantidad de bits transmitidos por segundo. El baud rate debe coincidir entre los dos dispositivos para que la comunicación sea efectiva.

### Estructura del Paquete de Datos en UART

La figura muestra un paquete de datos en UART, compuesto por los siguientes elementos:

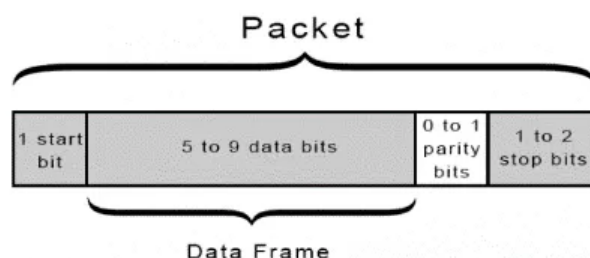


Figura 3: Estructura de datos de UART[3]

- **Bit de inicio:** Señal de nivel bajo (0) que indica el comienzo de la transmisión.
- **Bits de datos:** Contienen la información a transmitir, entre 5 y 9 bits.
- **Bit de paridad (opcional):** Utilizado para detectar errores, puede ser par, impar o no utilizado.
- **Bits de parada:** Uno o dos bits de nivel alto (1) que señalan el fin del paquete.

Esta estructura permite la sincronización y transmisión confiable de datos entre dispositivos sin necesidad de una señal de reloj compartida.

## LCD PCD8544

La pantalla **PCD8544** [2] es una pantalla gráfica monocromática de 48x84 píxeles. En este laboratorio, se utiliza para mostrar los valores de voltaje medidos por el Arduino de manera clara y organizada.

Características clave de la pantalla:

- Resolución de **48x84 píxeles**, adecuada para mostrar texto y gráficos sencillos.
- Bajo consumo de energía.
- Interfaz serial que simplifica la conexión con el Arduino.

El control de la pantalla se realiza mediante la librería `PCD8544.h`, que facilita la impresión de datos en la pantalla sin la necesidad de manejar manualmente los píxeles.

## Diagrama de bloques del controlador LCD PCD8544

El diagrama de bloques del controlador LCD PCD8544 se compone de los siguientes módulos principales:

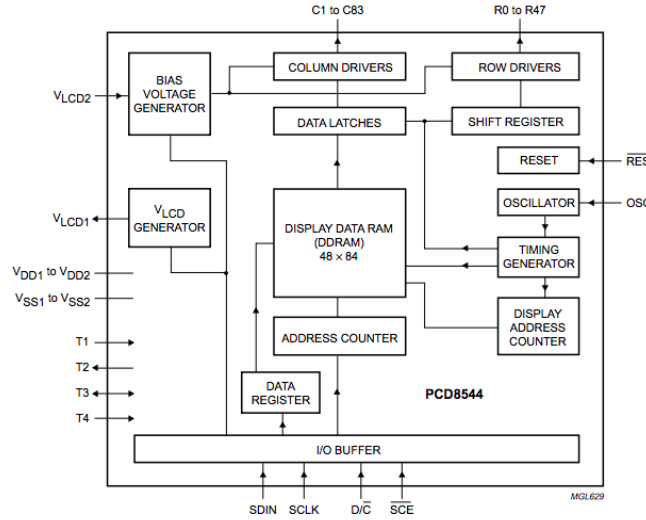


Figura 4: Diagrama de bloques PCD8544[2]

Este controlador está diseñado para manejar pantallas LCD gráficas, organizando la comunicación entre el microcontrolador y la pantalla a través de su memoria interna y sus drivers de columna y fila, permitiendo la visualización de gráficos en la matriz de 84x48 píxeles.

### SPI en PCD8544

La PCD8544 utiliza el protocolo SPI (Serial Peripheral Interface) para comunicarse con microcontroladores, permitiendo la transmisión rápida y eficiente de datos. MOSI (Master Out Slave In) para enviar datos, SCLK (Serial Clock) para sincronizar la transmisión, DC (Data/-Command) para indicar si se envían comandos o datos, y CS (Chip Select) para habilitar la comunicación con el display, como se observa en la figura 4. Durante la operación, el microcontrolador envía comandos iniciales para configurar el display y luego actualiza la memoria interna con los datos de imagen, permitiendo así la visualización de gráficos y texto en la pantalla monocromática.[6]

## Hardware

### LEDs

Los LEDs son diodos emisores de luz. Contienen dentro un semiconductor el cual, al recibir una diferencia de potencial continua, produce luz, fenómeno conocido como electroluminiscencia. Cuando se conecta un diodo LED, la tensión que soporta es normalmente de 2V; si la tensión es mayor, se debe colocar una resistencia en serie para no dañar el LED. [5] Dado que los pines del arduino entregan una tensión de aproximadamente 5V, la resistencia de protección necesaria se calcula de la siguiente manera:

$$R_{LED} = \frac{5 - 2}{20 \times 10^{-3}} = 150 \Omega \quad (1)$$

De esta manera se limita la corriente a 20mA y los LEDs funcionarían como alarma cuando el valor de la tensión sea menor a los 20V o mayor a los 20V.

## Diseño del Voltímetro y Conversión de Voltajes

El objetivo de este diseño es crear un voltímetro con cuatro canales capaces de medir un rango de voltaje de  $[-24, 24]$ V DC/AC, y visualizar estos valores en una pantalla LCD. Dado que el microcontrolador Arduino solo puede leer voltajes en el rango de  $[0, 5]$ V, es necesario implementar un circuito que convierta el voltaje de entrada en dicho rango.

### Selección del Circuito

Este circuito utiliza dos fuentes independientes y permite escalar la entrada de  $[-24, 24]$ V a  $[0, 5]$ V. El divisor de tensión está compuesto por tres resistencias:  $R_1$ ,  $R_2$  y  $R_3$ , y dos fuentes de voltaje  $V_s$  y  $V_{in}$ .

### Aplicación del Teorema de Superposición

El cálculo del voltaje de salida ( $V_o$ ) se realiza utilizando el **teorema de superposición**, que establece que la contribución de cada fuente puede calcularse por separado, y el valor total de  $V_o$  es la suma de estas contribuciones.

Para  $V_s$ , la contribución se describe como:

$$V_{o1} = \frac{R_1 \parallel R_3}{(R_1 \parallel R_3) + R_2} \cdot V_s \quad (2)$$

Para  $V_{in}$ , la expresión es:

$$V_{o2} = \frac{R_2 \parallel R_3}{(R_2 \parallel R_3) + R_1} \cdot V_{in} \quad (3)$$

Sumando ambas contribuciones, se obtiene:

$$V_o = V_{o1} + V_{o2} \quad (4)$$

$$V_o = \frac{R_1 \parallel R_3}{(R_1 \parallel R_3) + R_2} \cdot V_s + \frac{R_2 \parallel R_3}{(R_2 \parallel R_3) + R_1} \cdot V_{in} \quad (5)$$

### Resolución de Ecuaciones

Para ajustar los valores de  $R_1$ ,  $R_2$ , y  $R_3$ , se tienen las siguientes condiciones:

- Cuando  $V_{in} = 24$ V, se espera que  $V_{out} = 5$ V.
- Cuando  $V_{in} = -24$ V, se espera que  $V_{out} = 0$ V.

Asumiendo  $R_3 = 1,8k\Omega$  y  $V_s = 5$ V, las ecuaciones a resolver son las siguientes:

$$5 = \frac{R_1 \parallel 1800}{(R_1 \parallel 1800) + R_2} \cdot 5 + \frac{R_2 \parallel 1800}{(R_2 \parallel 1800) + R_1} \cdot 24 \quad (6)$$

$$0 = \frac{R_1 \parallel 1800}{(R_1 \parallel 1800) + R_2} \cdot 5 + \frac{R_2 \parallel 1800}{(R_2 \parallel 1800) + R_1} \cdot (-24) \quad (7)$$

Al resolver estas ecuaciones, se obtienen los valores óptimos para  $R_1$  y  $R_2$ . Por lo tanto,

$$\boxed{R_1 = 6,840k\Omega, R_2 = 1,425k\Omega} \quad (8)$$

## Tabla de componentes electrónicos

A continuación, se muestra una tabla con los componentes utilizados para el desarrollo del proyecto:

Componente	Descripción	Cantidad	Precio (€)
Arduino UNO	Microcontrolador	1	15999
PCD8544	Pantalla LCD	1	5800
LEDs	Indicadores de alerta	4	316
Botones	Switch AC/DC y Serial	2	2200
Potenciómetros	Reguladores de tensión	4	1000
Resistencia	150 $\Omega$	4	250
Resistencia	1,8k $\Omega$	4	250
Resistencia	1,425k $\Omega$	4	250
Resistencia	6,84k $\Omega$	4	250

Tabla 2: Componentes utilizados en el voltímetro

## Desarrollo

### Diagrama de flujo

Seguidamente, se muestra el diagrama de flujo que se desarrolla para el voltímetro:

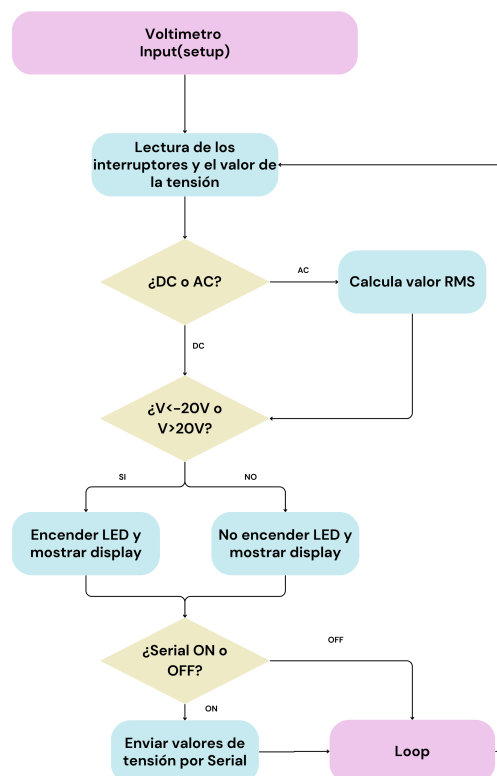


Figura 5: Diagrama de flujo del voltímetro

Entonces, más específicamente se tiene lo siguiente:



- **Input:** Es este se establecen las conexiones y se preparan los componentes para la medición.
- **Lectura de interruptores y voltajes:** Se leen los estado de los interruptores (switches) y las tensiones de los canales analógicos.
- **¿DC o AC?:** Se revisa el estado del switch para ver si funciona en modo DC o AC de manera que sea posible mostrar el display correcto.
- **AC:** En el este caso se calcula el valor eficaz RMS de las tensiones medidas. Y posteriormente se procede a revisar si el valor de las tensiones no supera los límites establecidos  $]-20V, 20V[$ .
- **DC:** Al ser el display por defecto, se procede automáticamente a chequear que el valor de las tensiones no supere los límites establecidos  $]-20V, 20V[$ .
- **Alarma de LEDs:** En ambos modos (AC y DC), si el voltaje está fuera de los límites seguros, los LEDs se encienden y se muestra el valor de tensión en el display. De lo contrario, se opta por no encender los LEDs, pero se sigue mostrando la lectura en el display.
- **¿Serial ON or OFF?:** Finalmente, se verifica el estado del interruptor de comunicación serial. Si el estado es ON, se envían los valores de tensión leídos a través de la conexión serial para su registro o análisis en la computadora. Si el estado es OFF, el sistema regresa al estado de Loop.

## Funcionamiento del circuito

El circuito se simuló en el programa SimulIDE de forma que se pudiera implementar la funcionalidad de un voltímetro a partir del uso de un microcontrolador Arduino Uno. Al comenzar la simulación se tiene el siguiente circuito:

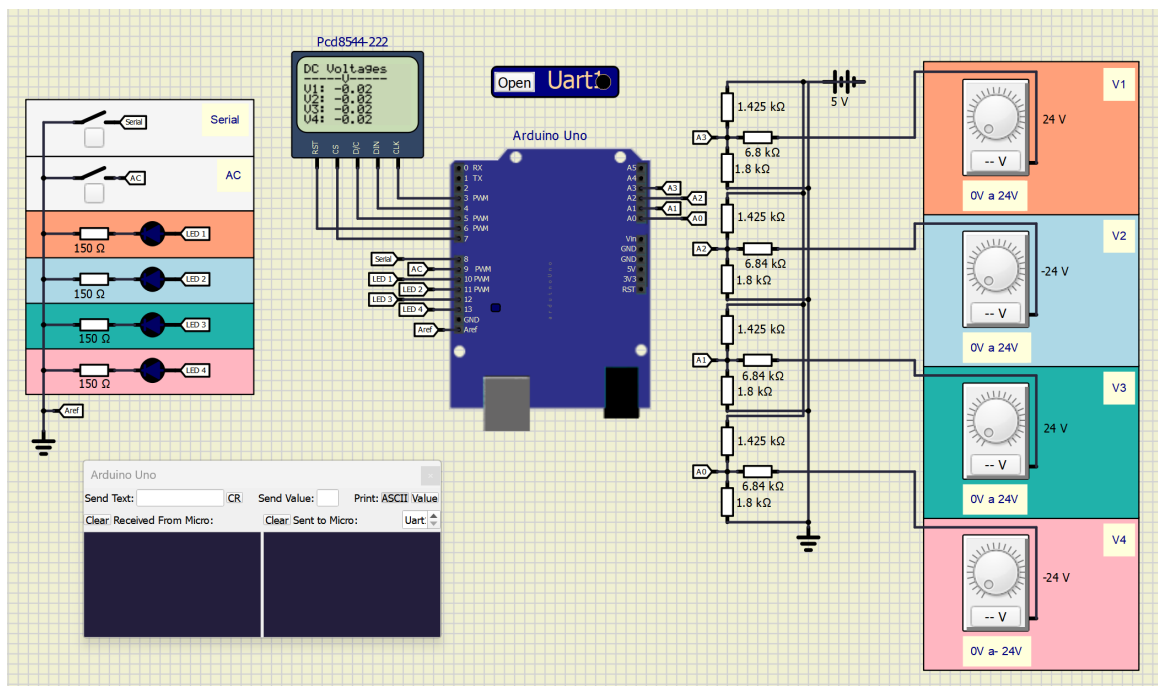


Figura 6: Circuito total para el funcionamiento del voltímetro

Para definir los voltajes de entrada se puede modificar el valor a partir de las fuentes ajustables. En total son 4 fuentes de voltaje, donde dos pueden tomar valores de -24V a 0V y las otras dos de 0V a 24V.

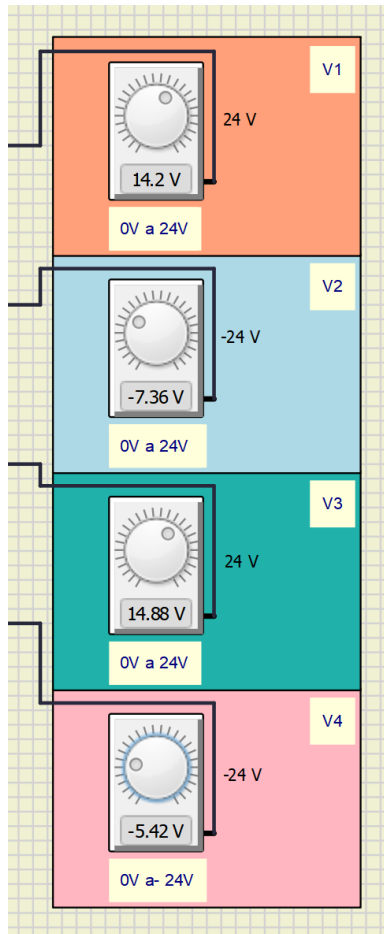


Figura 7: Fuentes de voltaje ajustables

Estos valores se muestran en la pantalla LCD con cierta tolerancia debido a ciertas pérdidas que sufren los datos a lo largo del circuito. Si se mide un valor de voltaje superior a 20V o menor a -20V se enciende un LED de alarma correspondiente a la entrada específica. Esto se puede observar a continuación, donde se tiene un voltaje de 21.27V y -23.29V en las entradas 1 y 4, respectivamente, por lo que se encienden los LEDs asociados a estas:

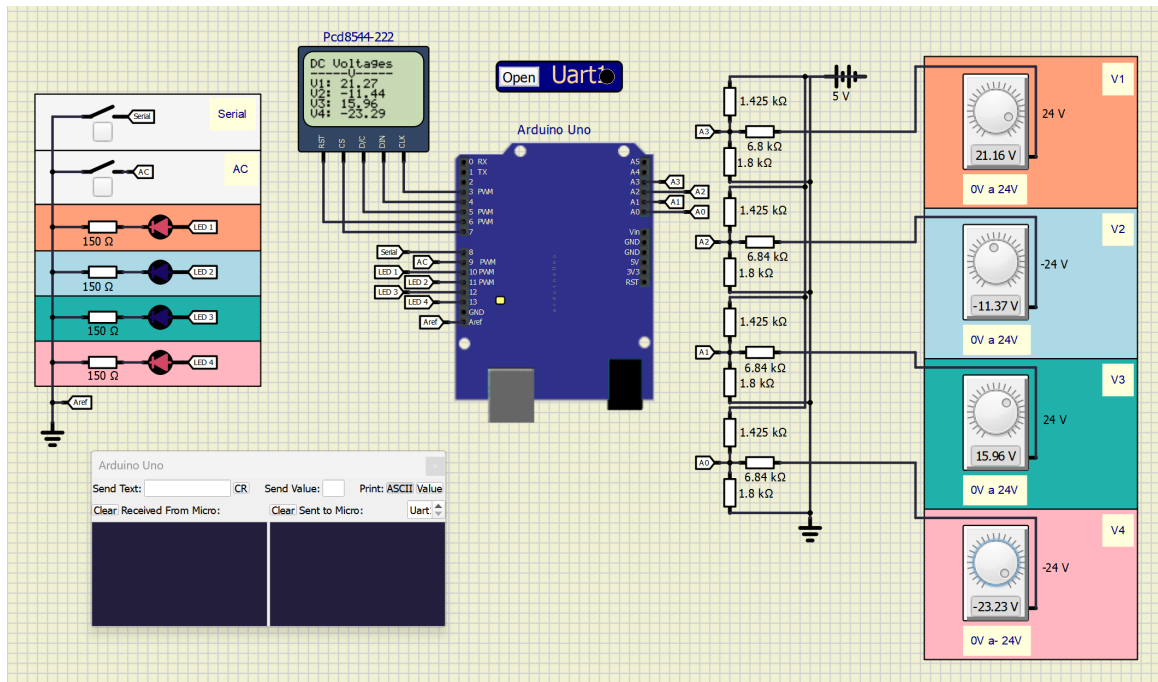


Figura 8: LED de alarma en caso de voltajes fuera de los márgenes deseados

Asimismo, se tiene un switch que controla el modo de medición DC o AC, midiendo como corriente directa cuando está abierto y como corriente alterna cuando se cierra. En la pantalla se muestran las mediciones. En el caso de medir voltaje AC se muestra en la pantalla el valor rms. En la siguiente figura se muestra en la pantalla el valor rms para un voltaje pico de 13.09V, 11.44V, 15.96V y 9.999V, provenientes de las entradas 1, 2, 3 y 4 respectivamente.

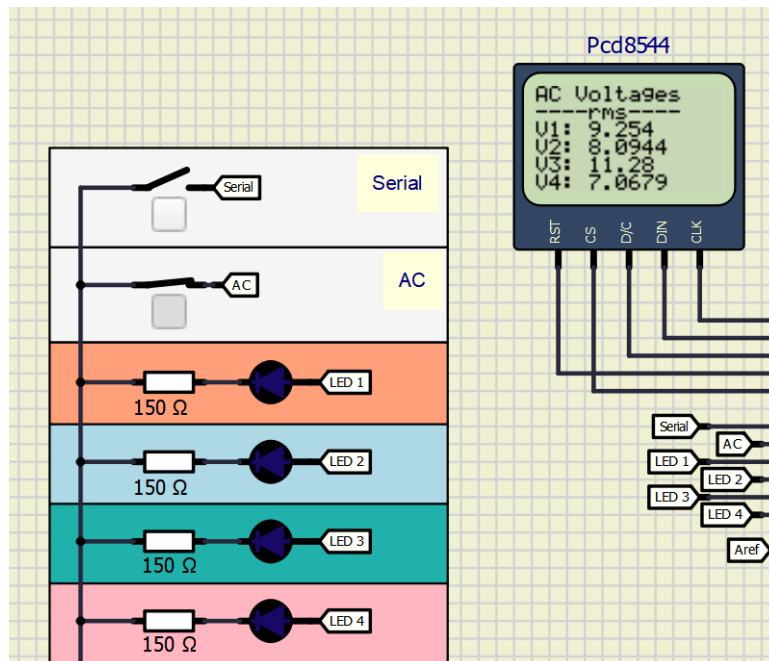


Figura 9: Valores medidos en modo AC

Si se desea enviar estos datos medidos por el puerto serial, se enciende el bloque USART (el cual en este caso está configurado en el puerto COM2) y se cierra también el switch que controla esta transmisión serial. Estos valores transmitidos se pueden apreciar también en el monitor serial.

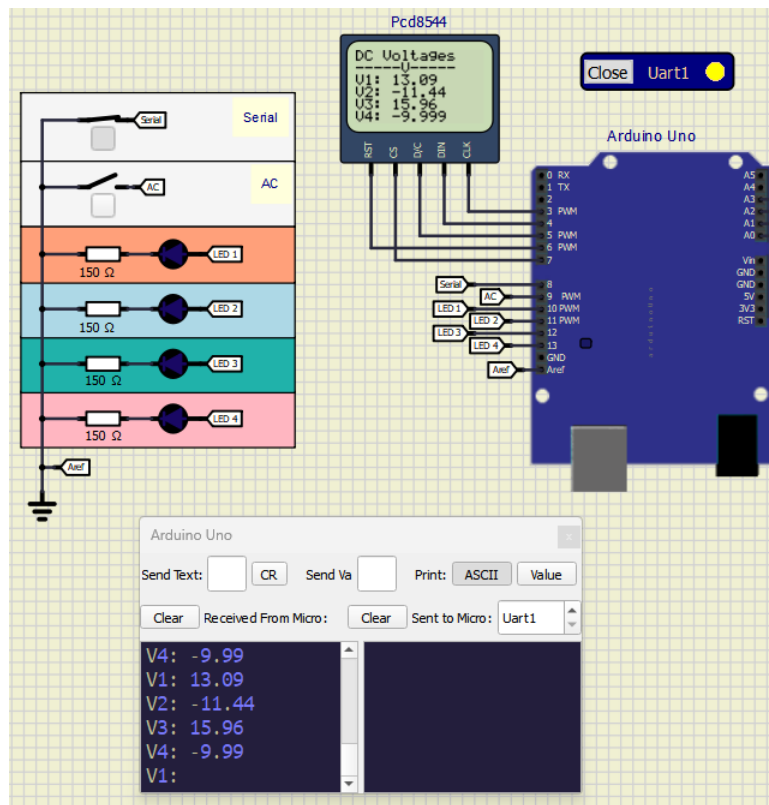


Figura 10: Se enciende el bloque USART y se cierra el switch que controla la transmisión serial

Posteriormente, se puede ejecutar el código de Python llamado "datasaver.py" para tomar los datos medidos y almacenarlos. La computadora lee el puerto serial mediante el programa de Python y guarda el registro de datos como un archivo CSV.

El programa se configuró para recolectar únicamente 200 datos, pero se puede modificar para guardar una mayor cantidad.

```

Windows PowerShell
PS C:\Users\leona\Desktop\Microcontroladores\Laboratorio3\src> python datasaver.py
Connected to: COM3
['V1: 16.14']
['V1: 16.14', 'V2: -15.30']
['V1: 16.14', 'V2: -15.30', 'V3: 6.93']
['V1: 16.14', 'V2: -15.30', 'V3: 6.93', 'V4: -14.50']
['V1: 16.14']
['V1: 16.14', 'V2: -15.30']
['V1: 16.14', 'V2: -15.30', 'V3: 6.93']
['V1: 16.14', 'V2: -15.30', 'V3: 6.93', 'V4: -14.50']
['V1: 16.14']
['V1: 16.14', 'V2: -15.30']
['V1: 16.14', 'V2: -15.30', 'V3: 6.93']
['V1: 16.14', 'V2: -15.30', 'V3: 6.93', 'V4: -14.50']
['V1: 16.14']
['V1: 16.14', 'V2: -15.30']
['V1: 16.14', 'V2: -15.30', 'V3: 6.93']
['V1: 16.14', 'V2: -15.30', 'V3: 6.93', 'V4: -14.50']
['V1: 16.14']
['V1: 16.14', 'V2: -15.30']
['V1: 16.14', 'V2: -15.30', 'V3: 6.93']
['V1: 16.14', 'V2: -15.30', 'V3: 6.93', 'V4: -14.50']
['V1: 16.14']
['V1: 16.14', 'V2: -15.30']
['V1: 16.14', 'V2: -15.30', 'V3: 6.93']

```

Figura 11: Datos mostrados en la terminal

	A
1	Volt1; Volt2; Volt3; Volt4
2	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
3	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
4	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
5	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
6	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
7	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
8	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
9	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
10	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
11	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
12	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
13	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
14	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
15	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
16	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
17	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
18	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
19	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
20	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
21	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
22	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
23	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
24	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
25	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
26	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
27	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
28	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
29	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
30	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
31	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
32	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
33	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
34	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
35	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50
36	V1: 16.14;V2: -15.30;V3: 6.93;V4: -14.50

Figura 12: Datos guardados en un csv

## Análisis electrónico

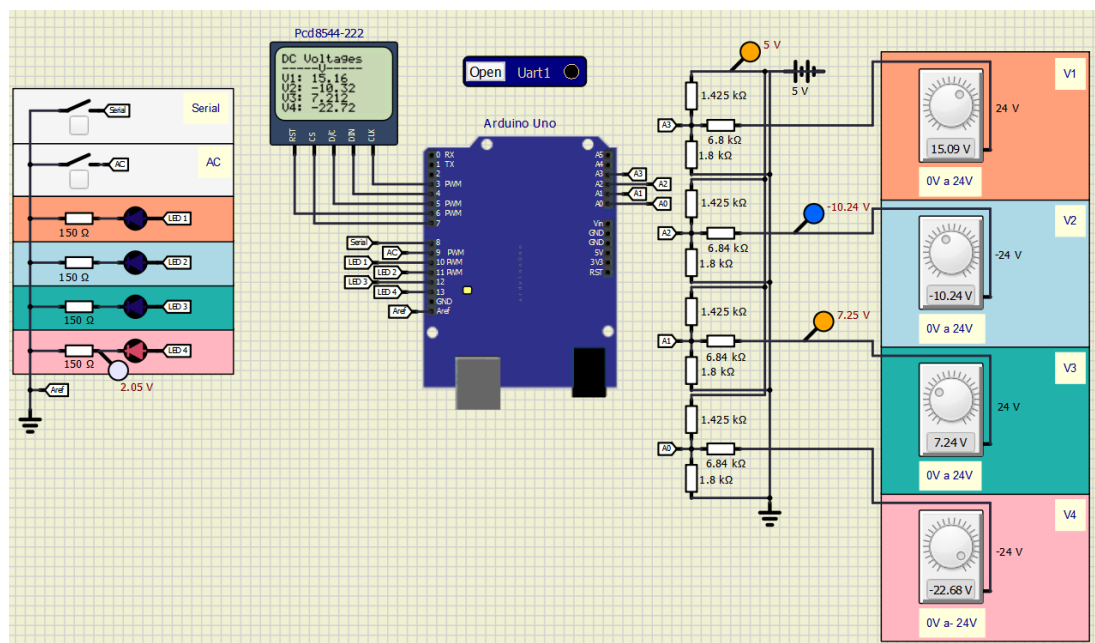


Figura 13: Circuito con voltímetros

El circuito utiliza un Arduino Uno para medir varios voltajes a través de sus entradas analógicas (A0, A1, A2 y A3), que provienen de diferentes fuentes de voltaje ajustables ( $V_1$ ,  $V_2$ ,  $V_3$ ,  $V_4$ ). Las mediciones se realizan utilizando divisores de voltaje que reducen los voltajes de las fuentes (hasta  $\pm 24$  V) a un rango adecuado para el Arduino (0 V a 5 V). Estos divisores están compuestos por resistencias de  $1,425\text{ k}\Omega$ ,  $6,8\text{ k}\Omega$  y  $1,8\text{ k}\Omega$ , lo que proporciona una proporción de reducción de aproximadamente  $1/5$ . El código del Arduino convierte las lecturas del ADC en valores de voltaje reales y los muestra en una pantalla PCD8544, indicando los valores de voltaje medidos en cada pin analógico:  $V_1$  (A3) muestra  $15,16\text{ V}$ ,  $V_2$  (A2) muestra  $-10,32\text{ V}$ ,  $V_3$  (A1) muestra  $7,21\text{ V}$ , y  $V_4$  (A0) muestra  $-22,72\text{ V}$ . Las mediciones realizadas por el Arduino son bastante precisas en comparación con los voltajes de referencia ajustados manualmente.

El sistema también cuenta con cuatro LEDs conectados a los pines digitales 10, 11, 12 y 13 del Arduino, los cuales se encienden cuando los voltajes leídos superan un umbral de  $\pm 20\text{ V}$ . En este caso, el LED correspondiente a A0 está encendido, ya que el voltaje medido ( $-22,72\text{ V}$ ) supera dicho umbral. Los otros LEDs permanecen apagados debido a que los voltajes en los otros pines no alcanzan los valores requeridos para activar los LEDs. Se mide una tensión de caída de  $2,05\text{ V}$  por el LED, teniendo en cuenta que debe pasar una corriente de aproximadamente  $20\text{ mA}$ .

Además, el circuito permite alternar entre los modos de medición de voltaje AC y DC, así como habilitar la impresión de los valores en el monitor serial, utilizando dos botones conectados a los pines 8 y 9.

## Conclusiones y recomendaciones

El voltímetro desarrollado con el microcontrolador Arduino Uno es capaz de medir con precisión voltajes en un rango de  $\pm 24\text{ V}$ , mostrando los resultados en una pantalla LCD y activando LEDs de advertencia cuando los valores superan los umbrales establecidos. La precisión de las mediciones es notablemente adecuada, ya que el sistema utiliza divisores de voltaje correctamente diseñados y ajustados para convertir los voltajes altos a un rango manejable para el ADC del Arduino. El cálculo de las resistencias es sumamente importante para lograr los resultados esperados.

El uso de LEDs para alertar sobre voltajes fuera de los márgenes seguros añade una capa de seguridad visual al sistema. Este diseño asegura que el usuario sea notificado de manera inmediata cuando los voltajes exceden los límites establecidos, mejorando la seguridad y confiabilidad del voltímetro. Además, se codificó el microcontrolador de manera efectiva para logrando medir tanto voltajes AC como DC, alternando entre ambos modos mediante un interruptor.

Por otro lado, se logró enviar y leer los datos del puerto serial correctamente. En algunas ocasiones se desconfigura el puerto del bloque USART, pero volviendo a hacer la conexión en "Virtual Serial Port Tools" se soluciona. Al ejecutar el programa de Python que lee y almacena los datos medidos del voltímetro, este funciona perfectamente, por lo que es posible cambiar los voltajes de entrada y registrar estos valores en un archivo CSV. No obstante, si se desea volver a leer y almacenar otro conjunto de datos, es necesario cerrar y abrir de nuevo SimulIDE, debido a un fallo de esta herramienta que no permite otra corrida del programa de manera seguida. Incluso, si se desea cerrar el puerto del bloque USART después de una ejecución, SimulIDE deja de funcionar.

## Referencias

- [1] Arduino UNO. A000066-Data Sheet. <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>
- [2] Texas Instruments, “PCD8544: 48 x 84 pixels matrix LCD controller/driver,” 1999. <https://www.alldatasheet.com/datasheet-pdf/pdf/18170/PHILIPS/PCD8544.html>
- [3] Arduino.cl, “Cómo configurar la comunicación UART en Arduino,”. <https://arduino.cl/como-configurar-la-comunicacion-uart-en-arduino/>.
- [4] Arduino, *analogRead()* - *Arduino Reference*. <https://www.arduino.cc/reference/es/language/functions/analog-io/analogread/>.
- [5] Gómez, R. ”Diodos LED y su resistencia en Arduino”. <https://robertogomez.info/diodos-led-y-su-resistencia-en-arduino/>.
- [6] E-Tinkers, “How to Use LCD 5110 PCD 8544 with Arduino,”. <https://www.e-tinkers.com/2017/11/how-to-use-lcd-5110-pcd-8544-with-arduino/>.

## Apéndices

## Features

- **ATMega328P Processor**
  - **Memory**
    - AVR CPU at up to 16 MHz
    - 32 kB Flash
    - 2 kB SRAM
    - 1 kB 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





# CONTENTS

<b>1 The Board</b>	<b>5</b>
1.1 Application Examples	5
1.2 Related Products	5
<b>2 Ratings</b>	<b>6</b>
2.1 Recommended Operating Conditions	6
2.2 Power Consumption	6
<b>3 Functional Overview</b>	<b>6</b>
3.1 Board Topology	6
3.2 Processor	7
3.3 Power Tree	8
<b>4 Board Operation</b>	<b>9</b>
4.1 Getting Started - IDE	9
4.2 Getting Started - Arduino Cloud Editor	9
4.3 Sample Sketches	9
4.4 Online Resources	9
<b>5 Connector Pinouts</b>	<b>10</b>
5.1 JANALOG	11
5.2 JDIGITAL	11
5.3 Mechanical Information	12
5.4 Board Outline & Mounting Holes	12
<b>6 Certifications</b>	<b>13</b>
6.1 Declaration of Conformity CE DoC (EU)	13
6.2 Declaration of Conformity to EU RoHS & REACH 211 01/19/2021	13
6.3 Conflict Minerals Declaration	14
<b>7 FCC Caution</b>	<b>14</b>
<b>8 Company Information</b>	<b>15</b>
<b>9 Reference Documentation</b>	<b>15</b>
<b>10 Revision History</b>	<b>15</b>
<b>11 电路板简介</b>	
11.1 应用示例	
11.2 相关产品	
<b>12 额定值</b>	
12.1 建议运行条件	



# 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 R3 as a tool for education purposes or industry-related tasks, the UNO R3 is likely to meet your needs.

**First entry to electronics:** If this is your first project within coding and electronics, get started with our most used and documented board; 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 UNO R3 board in industries, there are a range of companies using the UNO R3 board as the brain for their PLC's.

**Education purposes:** Although the UNO R3 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

- Arduino Starter Kit
- Arduino UNO R4 Minima
- Arduino UNO R4 WiFi
- Tinkerkit Braccio Robot



## 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.

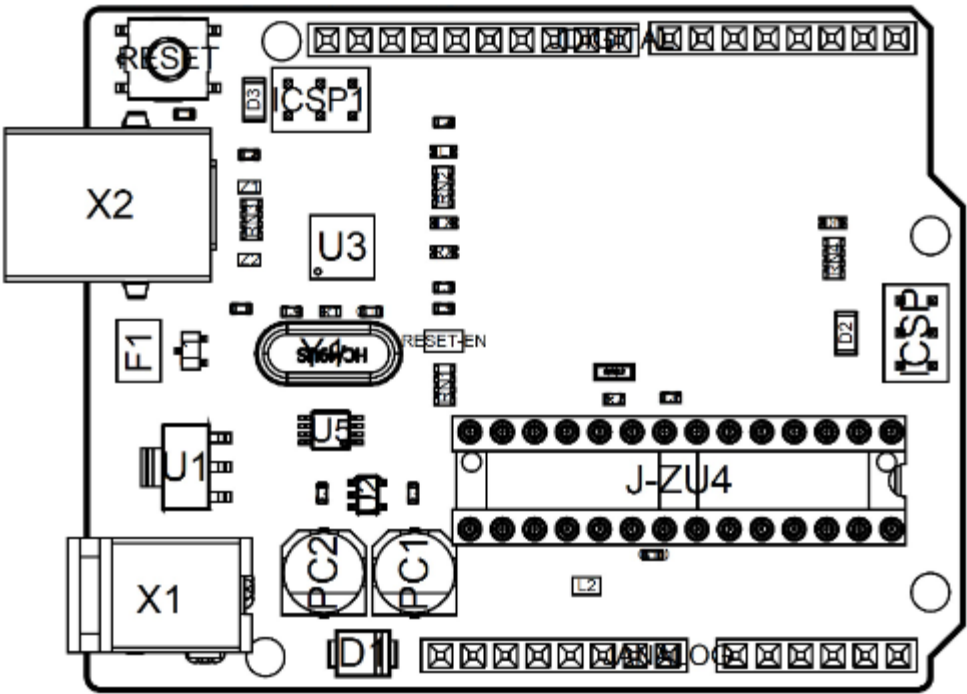
### 2.2 Power Consumption

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

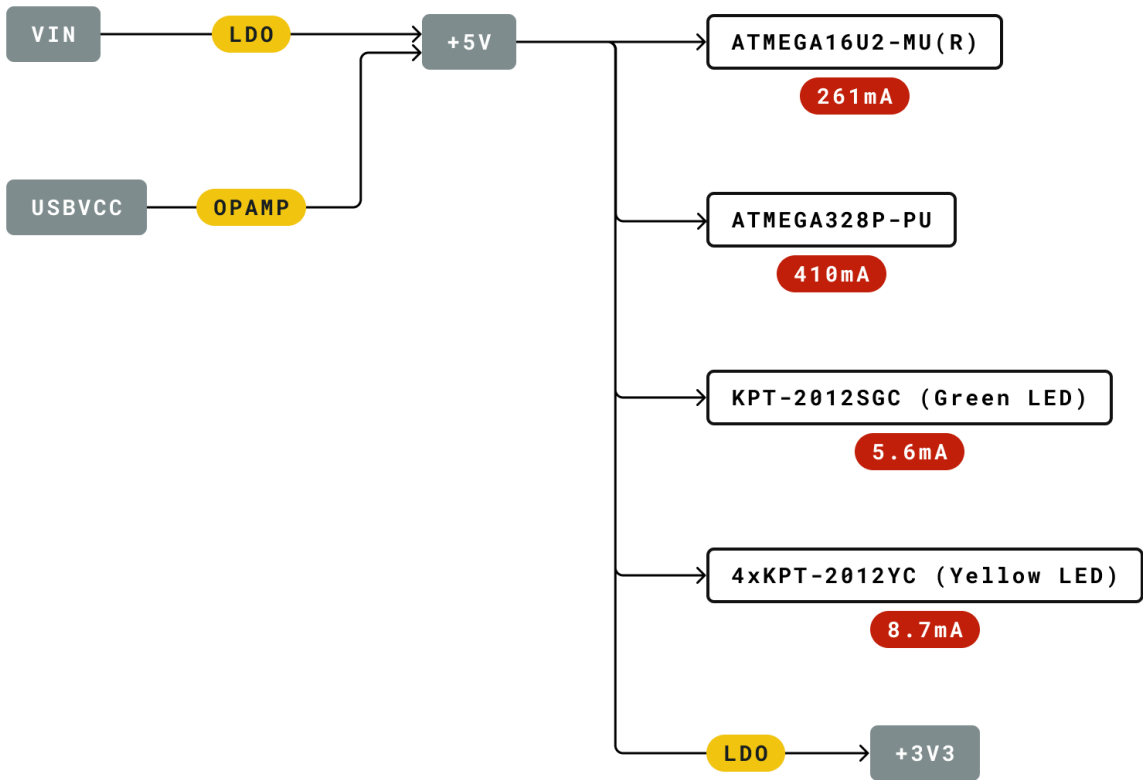


Ref.	Description	Ref.	Description
X1	Power jack 2.1x5.5mm	U1	SPX1117M3-L-5 Regulator
X2	USB B Connector	U3	ATMEGA16U2 Module
PC1	EEE-1EA470WP 25V SMD Capacitor	U5	LMV358LIST-A.9 IC
PC2	EEE-1EA470WP 25V SMD Capacitor	F1	Chip Capacitor, High Density
D1	CGRA4007-G Rectifier	ICSP	Pin header connector (through 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 UNO R3 while offline you need to install the Arduino Desktop IDE [1] To connect the UNO R3 to your computer, you'll need a USB-B cable. This also provides power to the board, as indicated by the LED.

### 4.2 Getting Started - Arduino Cloud Editor

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

The Arduino Cloud 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.

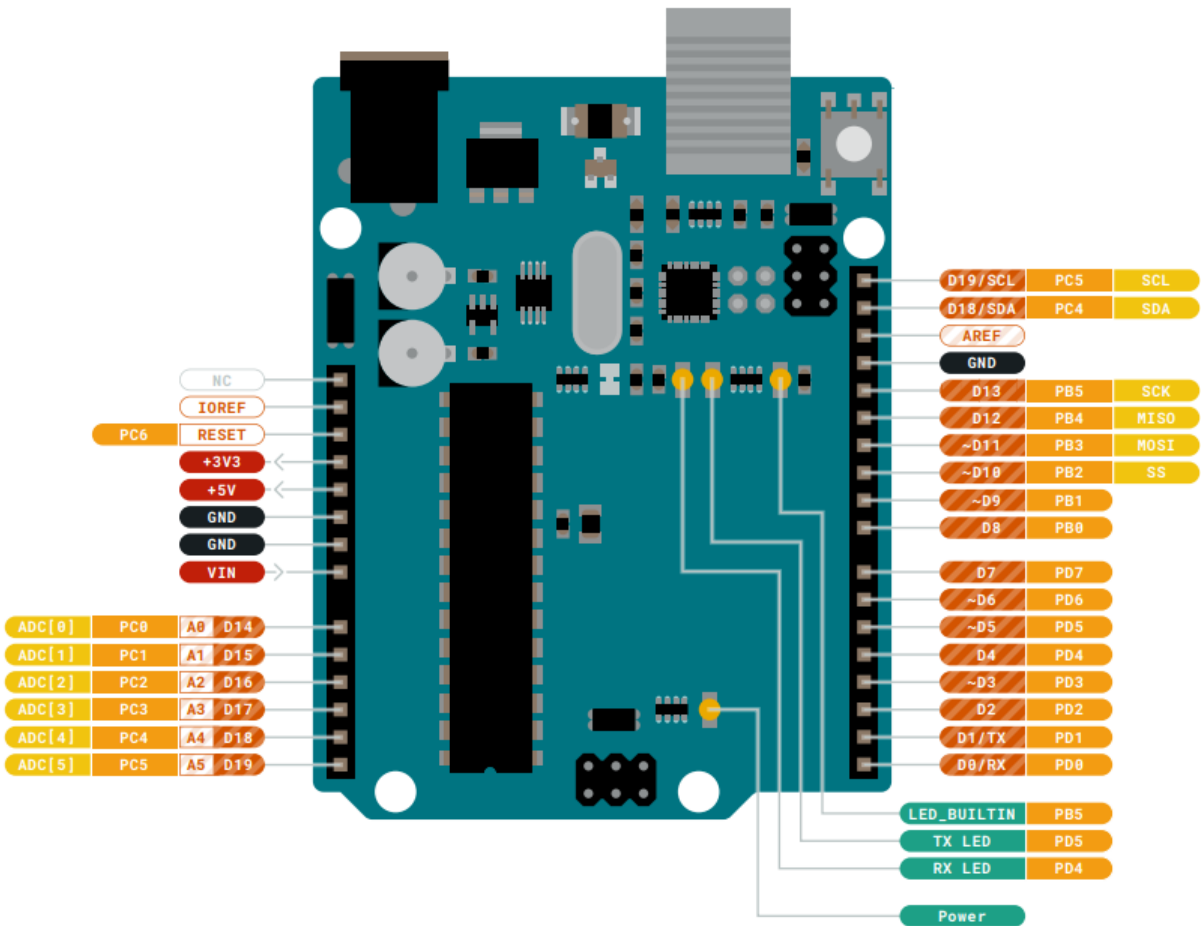
### 4.3 Sample Sketches

Sample sketches for the UNO R3 can be found either in the "Examples" menu in the Arduino IDE or in the "Documentation" section of the Arduino website [4].

### 4.4 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 Arduino Project Hub [5], the Arduino Library Reference [6] and the online Arduino store [7] where you will be able to complement your board with sensors, actuators and more.

# 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

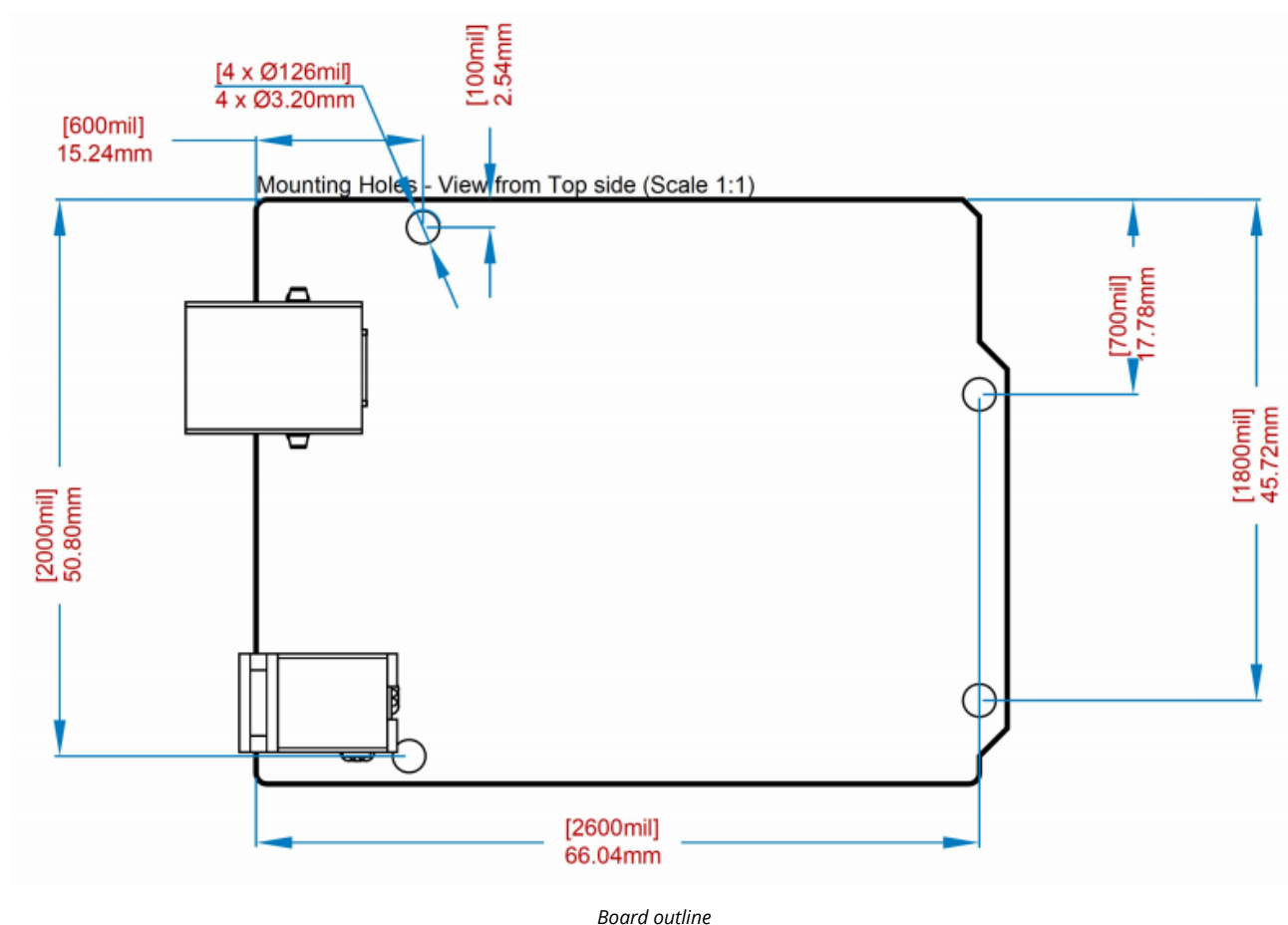
## 5.2 JDIGITAL

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)



## 5.3 Mechanical Information

## 5.4 Board Outline & Mounting Holes



## 6 Certifications

### 6.1 Declaration of Conformity CE DoC (EU)

We declare under our sole responsibility that the products above are in conformity with the essential requirements of the following EU Directives and therefore qualify for free movement within markets comprising the European Union (EU) and European Economic Area (EEA).

<b>ROHS 2 Directive 2011/65/EU</b>	
Conforms to:	EN50581:2012
<b>Directive 2014/35/EU. (LVD)</b>	
Conforms to:	EN 60950-1:2006/A11:2009/A1:2010/A12:2011/AC:2011
<b>Directive 2004/40/EC &amp; 2008/46/EC &amp; 2013/35/EU, EMF</b>	
Conforms to:	EN 62311:2008

### 6.2 Declaration of Conformity to EU RoHS & REACH 211 01/19/2021

Arduino boards are in compliance with RoHS 2 Directive 2011/65/EU of the European Parliament and RoHS 3 Directive 2015/863/EU of the Council of 4 June 2015 on the restriction of the use of certain hazardous substances in electrical and electronic equipment.

Substance	Maximum limit (ppm)
Lead (Pb)	1000
Cadmium (Cd)	100
Mercury (Hg)	1000
Hexavalent Chromium (Cr6+)	1000
Poly Brominated Biphenyls (PBB)	1000
Poly Brominated Diphenyl ethers (PBDE)	1000
Bis(2-Ethylhexyl} phthalate (DEHP)	1000
Benzyl butyl phthalate (BBP)	1000
Dibutyl phthalate (DBP)	1000
Diisobutyl phthalate (DIBP)	1000

Exemptions: No exemptions are claimed.

Arduino Boards are fully compliant with the related requirements of European Union Regulation (EC) 1907 /2006 concerning the Registration, Evaluation, Authorization and Restriction of Chemicals (REACH). We declare none of the SVHCs (<https://echa.europa.eu/web/guest/candidate-list-table>), the Candidate List of Substances of Very High Concern for authorization currently released by ECHA, is present in all products (and also package) in quantities totaling in a concentration equal or above 0.1%. To the best of our knowledge, we also declare that our products do not contain any of the substances listed on the "Authorization List" (Annex XIV of the REACH regulations) and Substances of Very High Concern (SVHC) in any significant amounts as specified by the Annex XVII of Candidate list published by ECHA (European Chemical Agency) 1907 /2006/EC.

### 6.3 Conflict Minerals Declaration

As a global supplier of electronic and electrical components, Arduino is aware of our obligations with regards to laws and regulations regarding Conflict Minerals, specifically the Dodd-Frank Wall Street Reform and Consumer Protection Act, Section 1502. Arduino does not directly source or process conflict minerals such as Tin, Tantalum, Tungsten, or Gold. Conflict minerals are contained in our products in the form of solder, or as a component in metal alloys. As part of our reasonable due diligence Arduino has contacted component suppliers within our supply chain to verify their continued compliance with the regulations. Based on the information received thus far we declare that our products contain Conflict Minerals sourced from conflict-free areas.

## 7 FCC Caution

Any Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

- (1) This device may not cause harmful interference
- (2) this device must accept any interference received, including interference that may cause undesired operation.

#### **FCC RF Radiation Exposure Statement:**

1. This Transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.
2. This equipment complies with RF radiation exposure limits set forth for an uncontrolled environment.
3. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.

English: User manuals for license-exempt radio apparatus shall contain the following or equivalent notice in a conspicuous location in the user manual or alternatively on the device or both. This device complies with Industry Canada license-exempt RSS standard(s). Operation is subject to the following two conditions:

- (1) this device may not cause interference
- (2) this device must accept any interference, including interference that may cause undesired operation of the device.

French: Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes :

- (1) l'appareil ne doit pas produire de brouillage
- (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

#### **IC SAR Warning:**

English This equipment should be installed and operated with minimum distance 20 cm between the radiator and your body.



French: Lors de l'installation et de l'exploitation de ce dispositif, la distance entre le radiateur et le corps est d'au moins 20 cm.

**Important:** The operating temperature of the EUT can't exceed 85°C and shouldn't be lower than -40°C.

Hereby, Arduino S.r.l. declares that this product is in compliance with essential requirements and other relevant provisions of Directive 2014/53/EU. This product is allowed to be used in all EU member states.

## 8 Company Information

Company name	Arduino S.r.l
Company Address	Via Andrea Appiani 25 20900 MONZA Italy

## 9 Reference Documentation

Reference	Link
Arduino IDE (Desktop)	<a href="https://www.arduino.cc/en/Main/Software">https://www.arduino.cc/en/Main/Software</a>
Arduino Cloud Editor	<a href="https://create.arduino.cc/editor">https://create.arduino.cc/editor</a>
Arduino Cloud Editor - Getting Started	<a href="https://docs.arduino.cc/arduino-cloud/guides/editor/">https://docs.arduino.cc/arduino-cloud/guides/editor/</a>
Arduino Website	<a href="https://www.arduino.cc/">https://www.arduino.cc/</a>
Arduino Project Hub	<a href="https://create.arduino.cc/projecthub?by=part&amp;part_id=11332&amp;sort=trending">https://create.arduino.cc/projecthub?by=part&amp;part_id=11332&amp;sort=trending</a>
Library Reference	<a href="https://www.arduino.cc/reference/en/">https://www.arduino.cc/reference/en/</a>
Arduino Store	<a href="https://store.arduino.cc/">https://store.arduino.cc/</a>

## 10 Revision History

Date	Revision	Changes
25/04/2024	3	Updated link to new Cloud Editor
26/07/2023	2	General Update
06/2021	1	Datasheet release