

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

IE0624 – Laboratorio III

III ciclo 2023

Título: GPIO, ADC y comunicaciones

Estudiantes:

Kevin Campos Campos

Josué Salmerón Córdoba

Grupo 1

Profesor: Marco Villalta

28 de enero de 2024

# Índice

1. Resumen	1
2. Nota teórica	2
3. Desarrollo/Análisis	12
4. Conclusiones y recomendaciones	19
5. Anexos	21

# Índice de figuras

1.	Diagrama de bloques de ATmega328P. Tomado de [1]. . . . .	3
2.	Diagrama de pines de ATmega328P. Tomado de [1]. . . . .	4
3.	Clasificación de pines de ATmega328P. Tomado de [2]. . . . .	4
4.	Amplificador inversor. . . . .	6
5.	Diseño antirebote. . . . .	7
6.	Circuito del voltímetro . . . . .	8
7.	Tensión de salida a medir. . . . .	8
8.	Funcionamiento correcto de la pantalla PCD8544. . . . .	9
9.	Conexión de los LEDs al arduino. . . . .	10
10.	Canales del voltímetro. . . . .	10
11.	Introducción del puerto serial y el switch para realizar la comunicación. . . . .	11
12.	Diagrama de bloques general . . . . .	12
13.	Diagrama de flujo función mide_voltaje . . . . .	13
14.	Diagrama de flujo alarma AC/DC. . . . .	14
15.	Medición de tensión DC de los 4 canales. . . . .	15
16.	Activación de alarma de los LEDs. . . . .	15
17.	Medición de tensión AC de los 4 canales. . . . .	16
18.	Verificación de la comunicación entre el arduino y el puerto serial. . . . .	16
19.	Verificación de la comunicación entre el puerto serial y el puerto virtual. . . . .	17
20.	Medición y comunicación serial tensión DC. . . . .	17
21.	Medición y comunicación serial tensión AC. . . . .	18
22.	Archivo csv generado. . . . .	18

# Índice de tablas

1.	Lista de equipos . . . . .	7
----	----------------------------	---

# 1. Resumen

En este trabajo se demuestra el funcionamiento de un voltímetro que opera bajo 4 canales con fuentes AC o DC. Inicialmente se muestran las magnitudes en DC, sin embargo, al presionar un botón el voltímetro es capaz de realizar la medición en AC. Ambos modos tienen un umbral que respetar, de no ser así, se encenderá el LED correspondiente al canal que sobrepase el valor de tensión establecido, de lo contrario se mantendrá apagado. Todas estas cuatro mediciones se reflejan en una pantalla PCD8544 que va refrescando los valores dependiendo si el usuario modifica las fuentes de tensión o cambia de un modo a otro. Además, este voltímetro por medio de comunicación SPI tiene la capacidad de guardar las tensiones eléctricas en archivo .csv, por tanto, queda demostrado que el voltímetro cumple con todas las funciones solicitadas.

## 2. Nota teórica

En esta sección del laboratorio, se exponen los principales componentes que usarán para el proyecto a realizar: un voltímetro.

### Arduino UNO

Se usará la placa de Arduino UNO que posee el microcontrolador ATmega328P.

#### Características generales

Sus detalles se describen a continuación:

- Es un MCU de 8 bits.
- Posee arquitectura RISC/Harvard.
- 4/8/16/64 kb memoria flash.
- 512b/1/2kb de memoria SRAM.
- 1/2kb de EEPROM.
- 23 GPIOs.
- Timer/Counters de 8 y 16 bits.
- Posee interrupciones.
- 8 canales PWM y comparador analógico.
- 6 canales 10-bit ADC.
- Posee protocolo SPI y USART (Universal Synchronous/Asynchronous Receiver/Transmitter) I2C.

#### Diagrama de bloques y pines

El diagrama de bloques de este MCU se muestra en la figura 1.

Fig. 1. Block Diagram

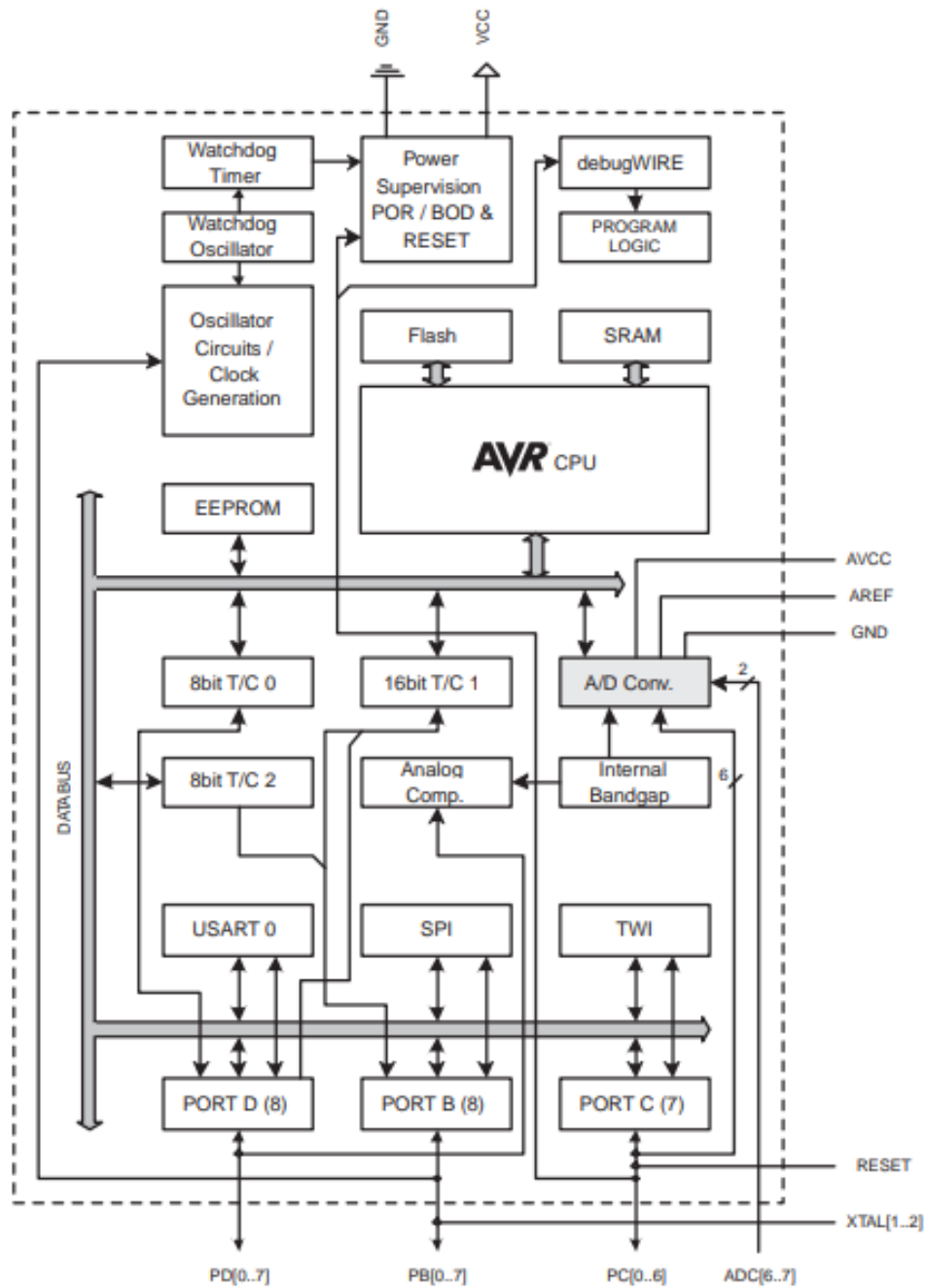


Figura 1: Diagrama de bloques de ATmega328P. Tomado de [1].

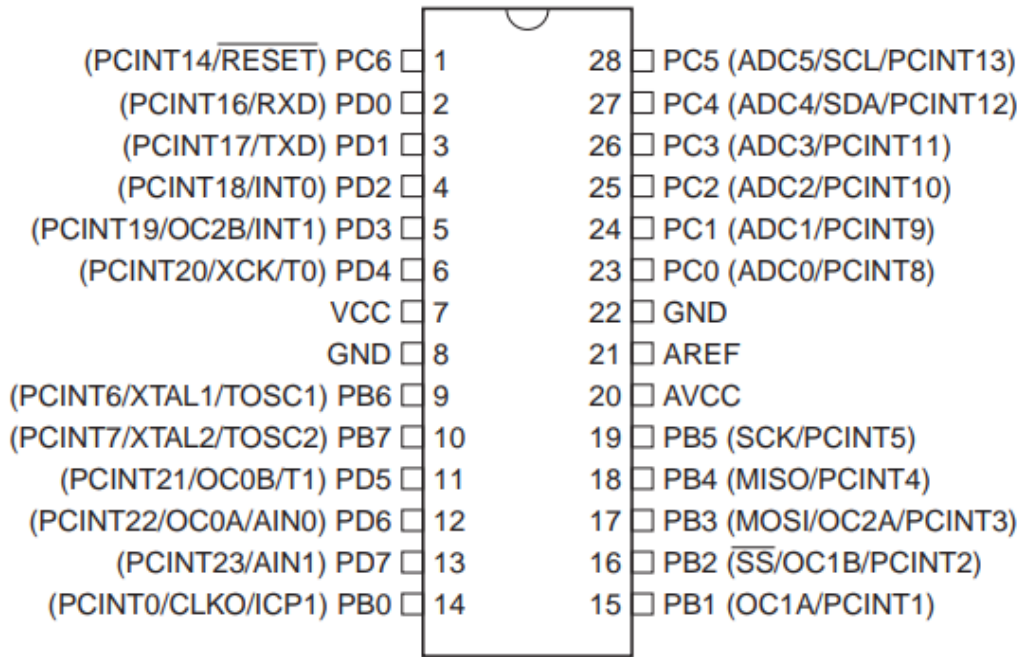


Figura 2: Diagrama de pines de ATMega328P. Tomado de [1].

Dado que será necesario saber a profundidad la clasificación de estos pines la figura 3 brinda una mejor clasificación de éstos.

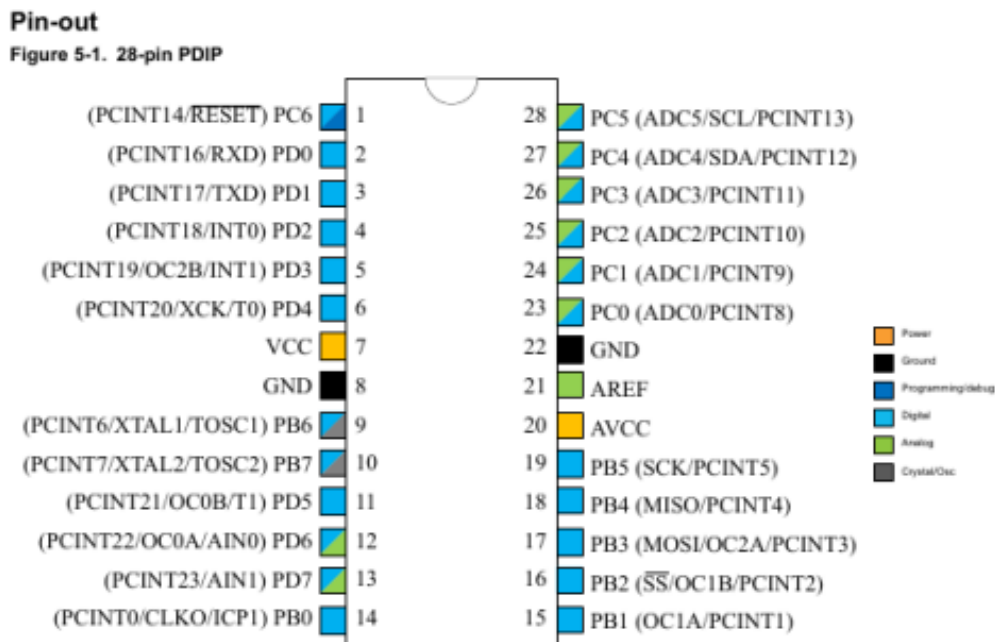


Figura 3: Clasificación de pines de ATMega328P. Tomado de [2].

## Características eléctricas

La siguiente lista describe estos detalles.



- Voltaje: 1.8-5,5 V.
- Velocidad: 0-4 MHz, 1.8-5,5 V. 0-10 MHz, 2.7-5,5 V. 0-20 MHz, 4.5-5,5 V
- Modo activo: 0,3 mA
- Temperatura de funcionamiento:  $-55^{\circ}$  a  $+125^{\circ}$ .
- Temperatura de almacenamiento:  $-65^{\circ}$  a  $+125^{\circ}$ .
- Temperatura pin RESET y GND: -0,5 V a 13 V.
- Temperatura en los demás pines: -0,5 V a VCC0,5 V.
- Corriente por pin I/O: 40 mA.

Sabiendo lo anterior, resulta pertinente mencionar la descripción de los pines usados en este laboratorio. Para la conexión de las fuentes de DC y AC, se usaron los pines A5, A4, A3, A2. Que se caracterizan por ser entre analógicos y digitales, esto permite tener más facilidad en cuanto a funciones. Mismo escenario para el botón de modo AC, solo que en este caso está conectado al pin A0. En cuanto a los pines que activan las alarmas de los LEDs y los que se conectaron a la pantalla PCD8544 se han usado los que son totalmente digitales (11-3), es decir, estos pines solo pueden funcionar como salidas o entradas, por lo que se declararon como INPUT o OUTPUT. Por último, para la comunicación serial se usaron los pines RX y TX, ya que son los encargados de la comunicación.

## Periféricos utilizados

Algunos periféricos usados del microcontrolador así como la pantalla PCD8544 fueron los siguientes:

- **pinMode:** esto se usó para cual pin era una entrada o salida, y esto ayuda para que el microcontrolador entienda todas las acciones que se desea realizar.
- **digitalWrite:** básicamente su función es para establecer el estado de una variable después de una acción, típicamente es para poner en alto o en bajo una señal.
- **analogRead:** esto devuelve el valor leído del pin de entrada analógico, el valor es proporcional a la entrada analógica tomando como base una tensión de referencia: 0-1023.
- **display.begin:** inicia/enciende la pantalla.
- **display.setContrast:** configuración del contraste de la pantalla.
- **display.clearDisplay:** limpia la pantalla.
- **display.setTextSize:** configura la posición del texto.
- **display.setTextColor:** establece el color de las letras.
- **display.setCursor:** este parámetro sirve para posicionar el texto en la pantalla.
- **display.println:** imprime el contenido deseado.
- **display.display:** imprime el logo de Adafruit y se coloca después de cada texto, de lo contrario no se mostrará el contenido.

Si bien, se debe mencionar que la pantalla PCD8544 desempeña un papel importante en este trabajo, porque permite mostrar las magnitudes de las tensiones eléctricas en las fuentes de alimentación, tanto para modo DC como AC. Por tanto, hay que conocer los pines que se usaron para darle sentido a los periféricos mencionados anteriormente. En el simulador se tienen 5 pines que se explican a continuación [3].

- **RST:** esta señal reiniciará el dispositivo y debe ser aplicado adecuadamente al chip. La señal activa está en bajo.
- **CS:** habilita la pantalla con el que se está comunicando en un bus SPI. Cuando esta señal está activa, el PCD8544 está habilitado y listo para recibir comandos o datos a través del bus SPI. Cuando la señal CS está inactiva, la pantalla no responde y no acepta datos.
- **D/C:** selecciona el modo de operación, alto o bajo.
- **DIN:** es una entrada para la línea de datos.
- **CLK:** es la señal de reloj que va de 0.0 a 4.0 Mbit/s.

Por lo que, los datos que se muestran en la pantalla PCD8544 es porque se usa el modelo de comunicaciones SPI: **S**erial **P**eripheral **I**nterface, es un bus de interfaz comúnmente utilizado para enviar datos entre microcontroladores y pequeños periféricos como registros de desplazamiento, sensores y tarjetas SD [4].

## Componentes electrónicos complementarios

Algunos de los componentes extra que fueron de gran ayuda para el diseño del voltímetro fueron los siguientes:

- **Amplificadores inversores:** se optó por esta opción, ya que el voltaje de salida de este tipo de amplificadores tienen la siguiente expresión:

$$v_o = -v_i \cdot \frac{R_2}{R_1} \quad (1)$$

Esto sirvió para diseñar que los rangos de tensión eléctrica no fueron mayores a 5 V. De esa manera, la base para lograr esto fue con valores de  $R_1 = R_2 = 2,5 \Omega$  y  $v_i = -2,5 \text{ V}$ ,

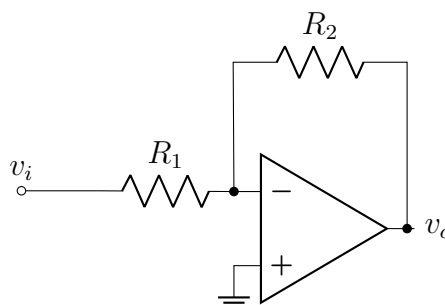


Figura 4: Amplificador inversor.

A partir, del esquemático anterior y la fórmula 1 fue posible regular las fuentes de tensión AC/DC para los 4 canales con el objetivo de obtener las mejores aproximaciones con respecto al voltímetro y lo mostrado en la pantalla.

- Relay: en este componente se conectan las 8 entradas de las fuentes AC/DC y salen 4 que están conectadas a los amplificadores inversores, la idea es determinar cuando las fuentes AC y DC estén normalmente cerradas o normalmente abiertas.
- Switches: son de gran ayuda para definir en que estado opera el voltímetro, si está realizando mediciones en AC o en DC, para realizar correctas lecturas de este cambio se debe tener en cuenta el diseño previo de valores de resistencia y capacitancia para evitar falsas lecturas.

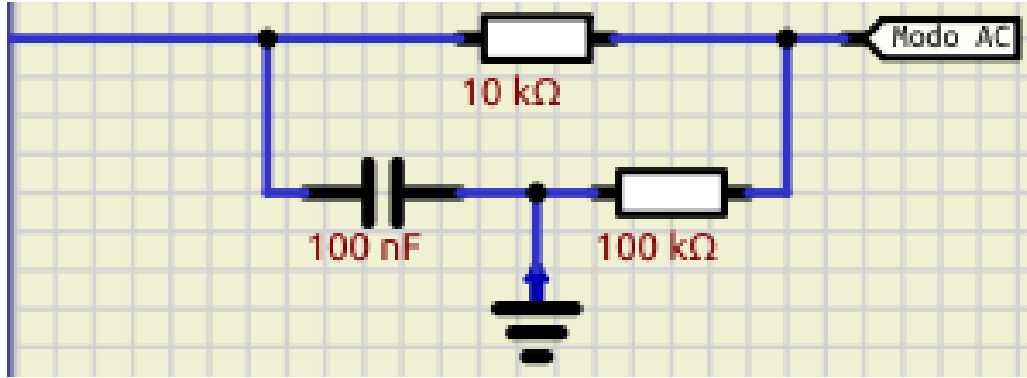


Figura 5: Diseño antirebote.

El diseño anterior, ayuda a que la transición de DC a AC sea inmediata.

## Lista de componentes

La siguiente tabla resume los componentes (con sus precios) utilizados para este trabajo. La lista de precios mostrados en la tabla 1 se tomaron de [5].

Tabla 1: Lista de equipos

Componente	Cantidad	Precio
Arduino UNO	1	17\$
Kit de resistencias	1	9.99\$
Capacitancias	1	0.2\$
LEDs	4	2.2\$
Pantalla PCD8544	1	5.85\$
Botón	9	5.85\$
Amplificadores	4	3.8\$
RelaySPST	2	29\$
Fuente de voltaje AC/DC	1	48.95\$
<b>Total</b>		122.84

## Diseño del circuito

El circuito que simula el voltímetro se muestra en la figura 6.

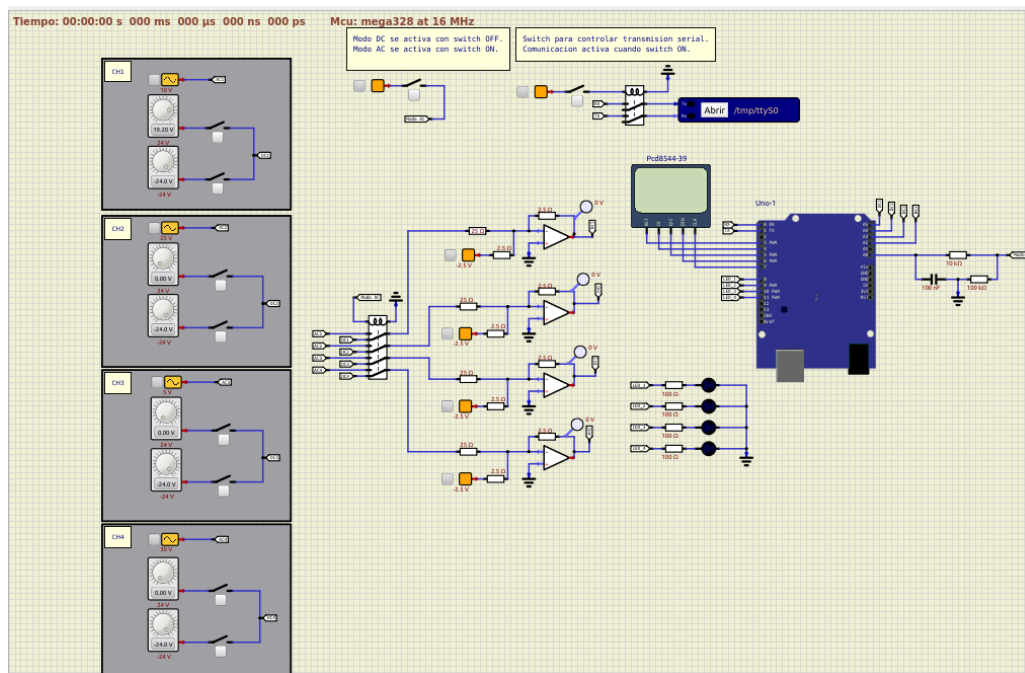


Figura 6: Circuito del voltímetro

Inicialmente, se hizo uso del amplificador inversor para lograr obtener una magnitud de tensión eléctrica adecuada y así no sobrepasar el umbral del arduino. Tal como se mostró en la sección de componentes complementarios, tomar esos valores de resistencias y elegir un voltaje de magnitud negativa dará un valor adecuado que permita colocar fuentes de tensión para el buen funcionamiento del voltímetro.

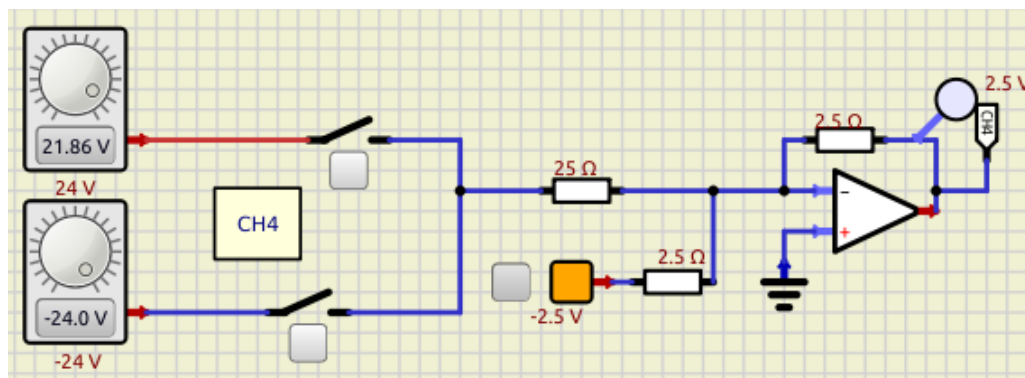


Figura 7: Tensión de salida a medir.

Esto se realizó para los 4 canales que se van usar, los cuales se conectaron como entradas al arduino en los pines A5-A2. Luego, en los pines 7-3 se conectaron las señales CLK, DIN, D/C, CS, RST respectivamente, esto para realizar unas pequeñas pruebas de funcionamiento de la pantalla mostrando un pequeño mensaje para comprobar su correcta conexión. En la sección de resultados se darán más detalles de este diseño.

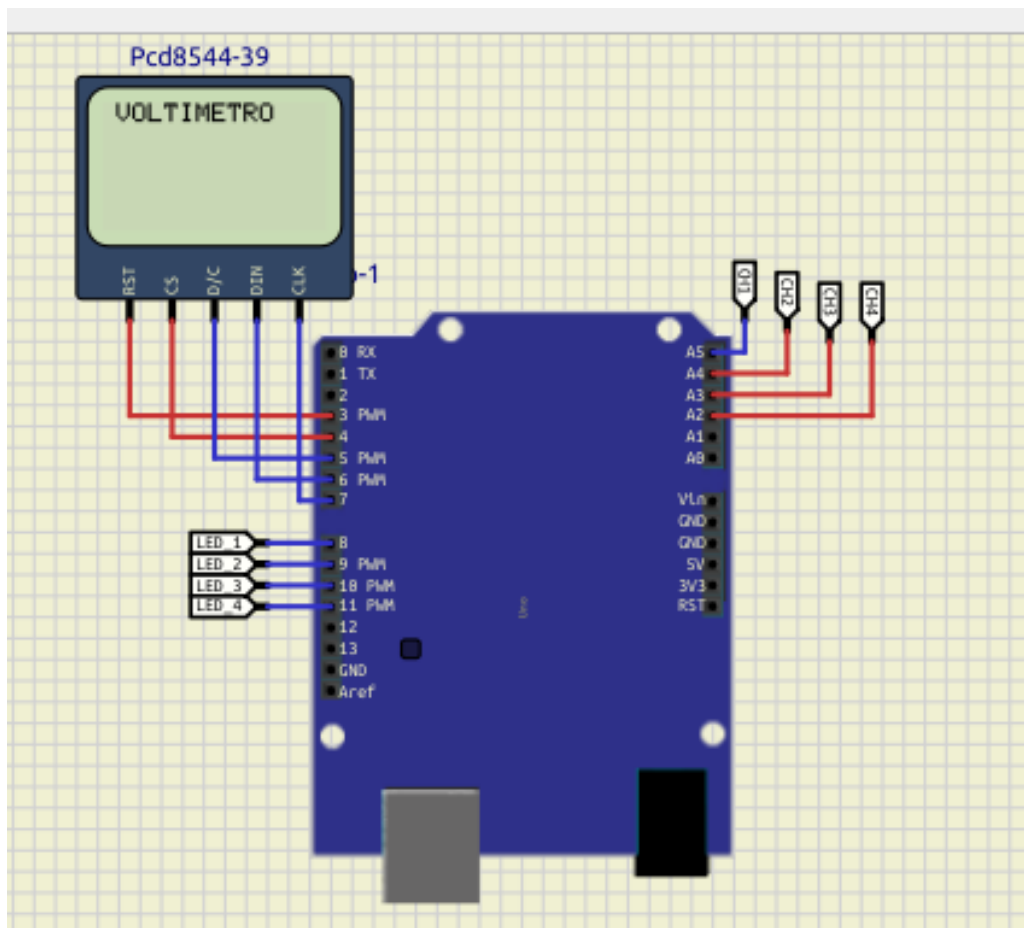


Figura 8: Funcionamiento correcto de la pantalla PCD8544.

Después, la conexión de los LEDs de alarma al arduino en los pines 8-11 que fueron configurados como salidas, se van a encender cuando se supere el rango de -20 a 20 en DC, y -14.14 a 14.14 en AC.

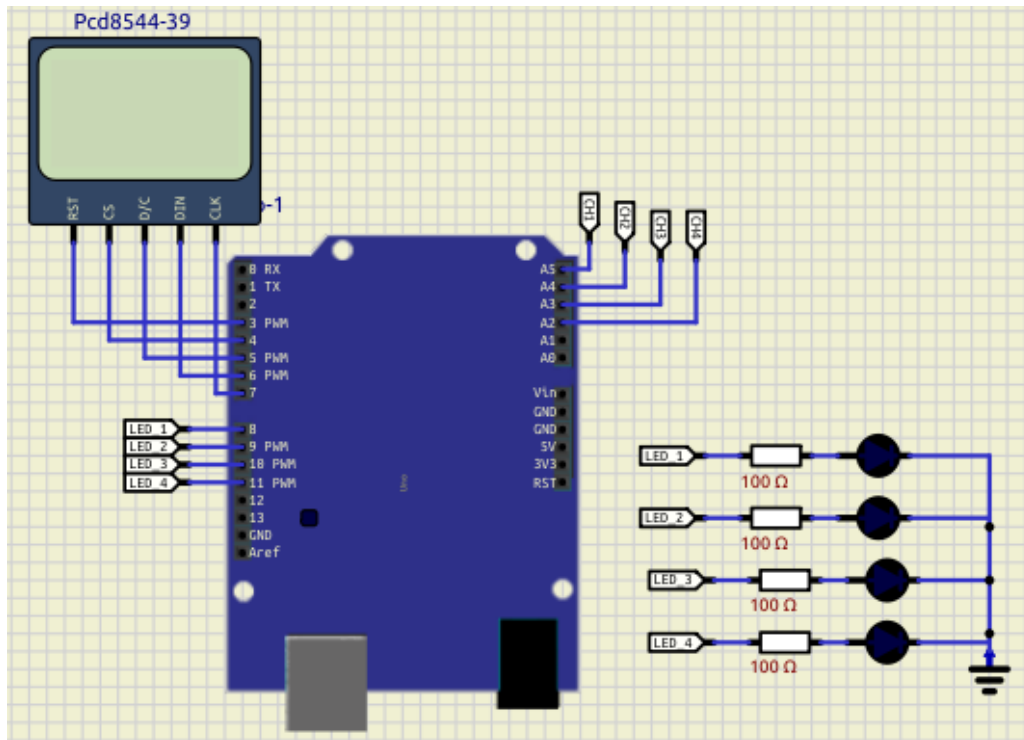


Figura 9: Conexión de los LEDs al arduino.

Otro detalle, para ordenar el voltímetro se decidió usar un relay que sirve de puente entre las fuentes AC/DC y la salida de los amplificadores inversores que se comunicará con la PCD8544, la cual mostrará las tensiones eléctricas en AC (si el botón está en alto) o en DC, enviará señales a los LEDs y estos se pondrán en alto o no dependiendo de las magnitudes que tengan las fuentes variables de tensión y reflejadas en la PCD8544. Esto se ve de la siguiente manera <sup>1</sup>.

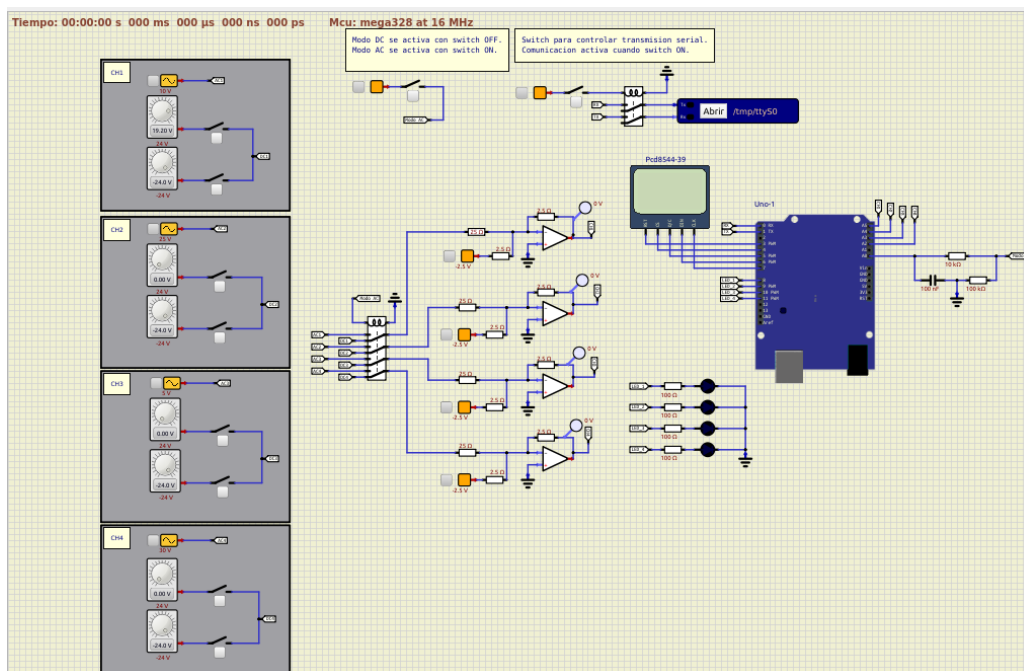


Figura 10: Canales del voltímetro.

<sup>1</sup>Por resolución de la imagen solo se mostrarán 3, en realidad posee 4.

Ahora, para la comunicación serial se introduce un puerto serial encargado de realizar la comunicación entre el arduino y el puerto serial, también se hace uso de un relé para hacer cumplir con lo solicitado de que la transmisión se pueda controlar por medio de un switch, el resultado final se observa en la siguiente figura:

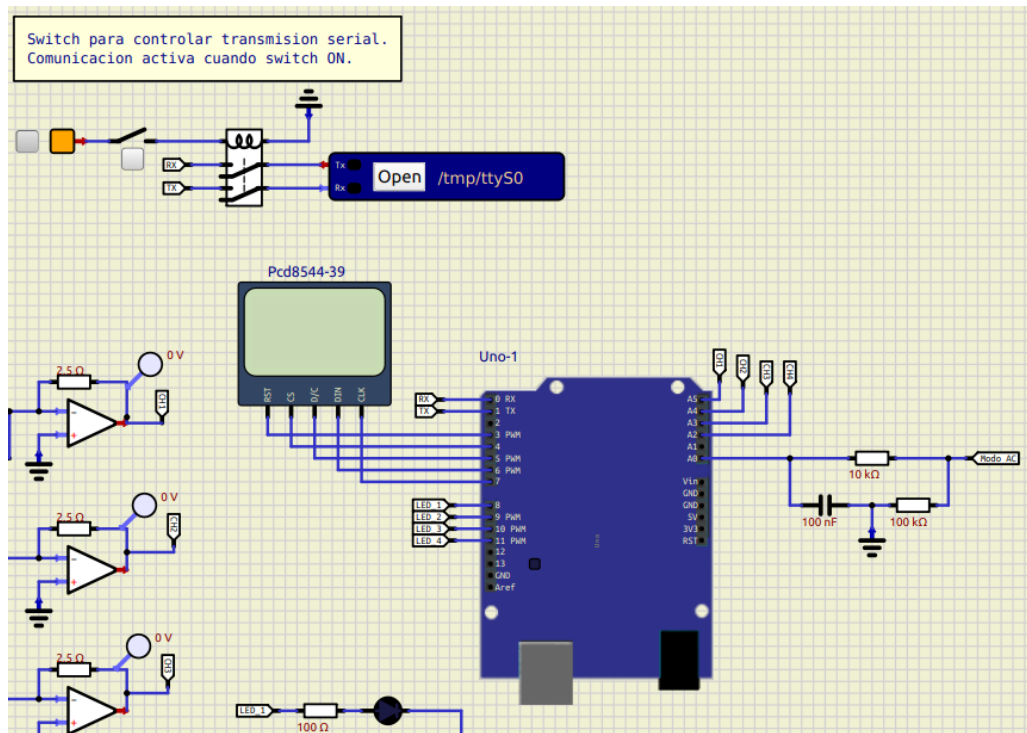


Figura 11: Introducción del puerto serial y el switch para realizar la comunicación.

Es importante cuando se hace la conexión ver que se toma el RX de un puerto y se conecta con el puerto TX para que se pueda dar la comunicación como se vio en clases.

### 3. Desarrollo/Análisis

El diagrama de bloques generales para el funcionamiento del voltímetro es el siguiente.

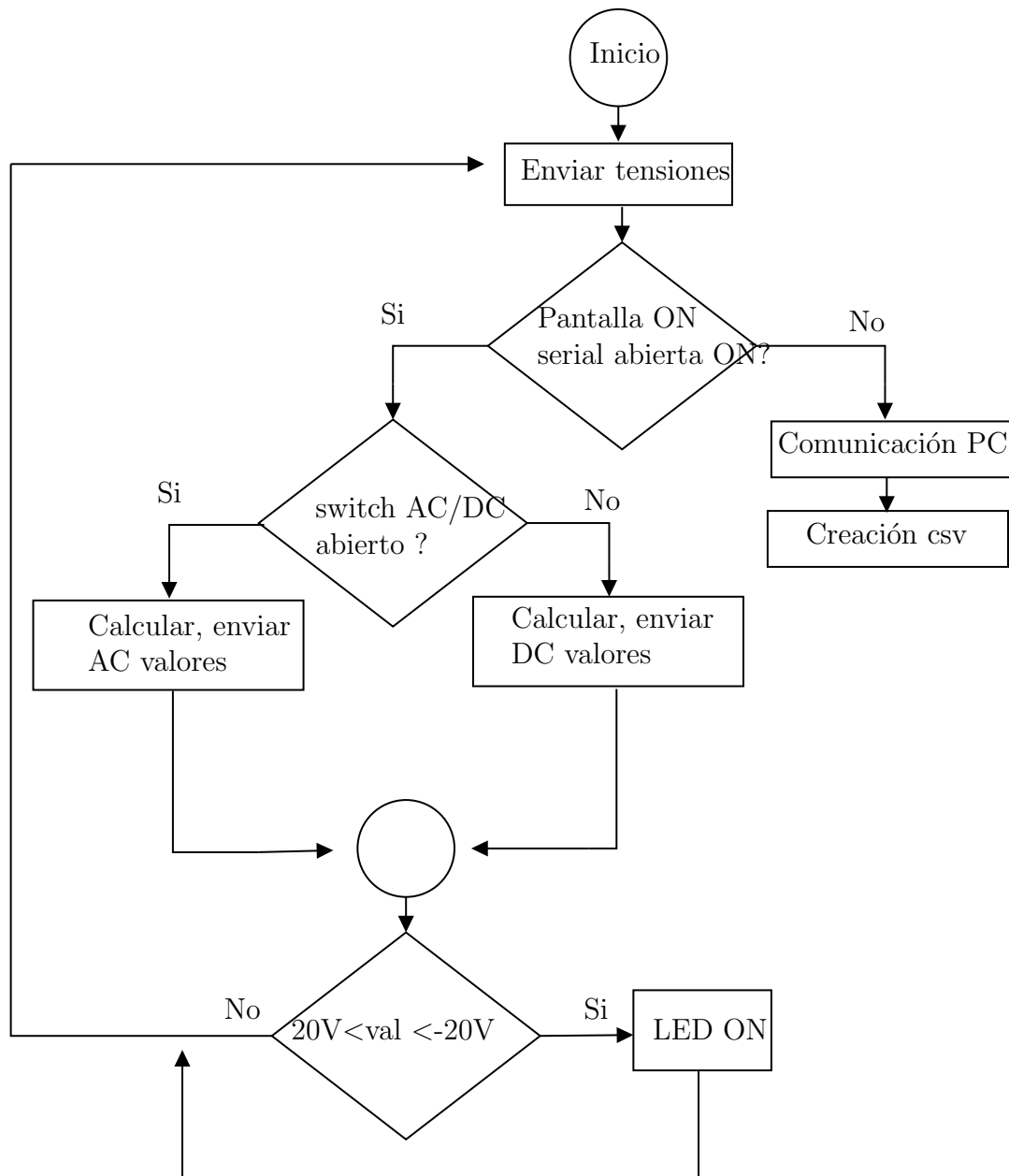


Figura 12: Diagrama de bloques general

Una vez que se activen alguna de las fuentes, en la pantalla PCD8544 se mostrará las magnitudes (despreciables, 0,05 V) por default en cada canal, y al estar el botón de modo AC/DC en bajo por tanto, en esta pantalla se mostrarán los valores en DC, si éste se pone en alto entonces en la pantalla se mostrarán los valores rms en AC, ahora, dependiendo de las magnitudes que se ajusten en ambos modos los LEDs de alarma se encenderán o no, que al mismo tiempo se verán en la pantalla. Por último, al presionar el botón de comunicación serial, se establece la comunicación adecuada para generar el informe del archivo csv.

Luego, para realizar las medidas adecuadas se tuvo que realizar una función capaz de normalizar y escalar en el intervalo de rango solicitado:  $[-24, 24]$ .



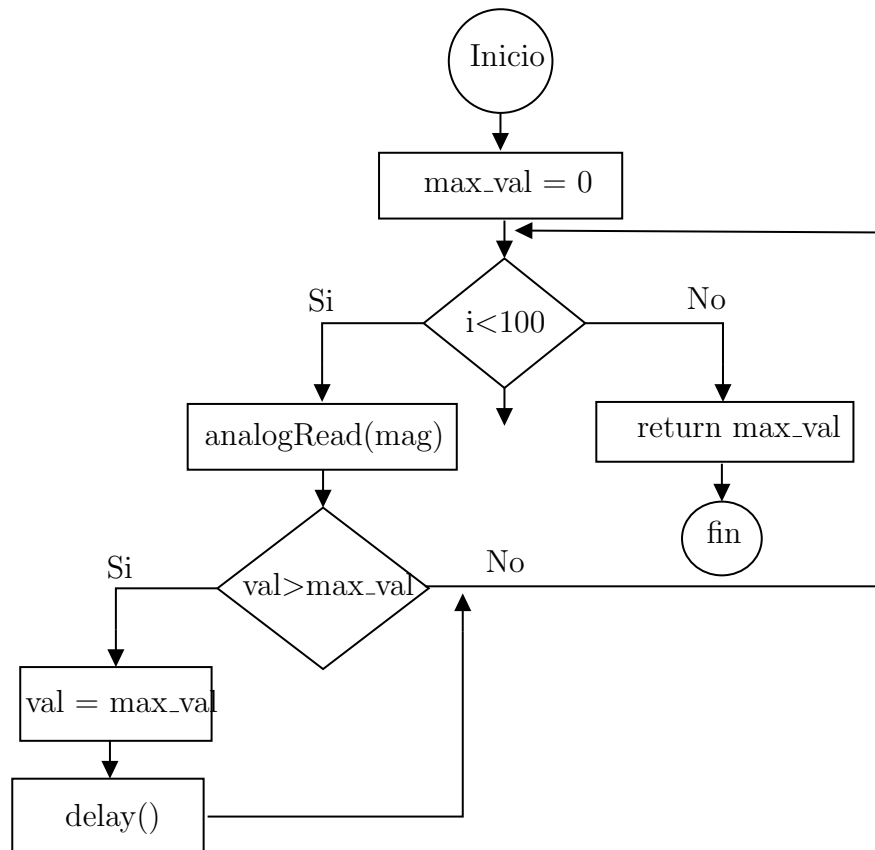


Figura 13: Diagrama de flujo función mide\_voltaje

Esta función se compone de un ciclo `for` que va iterando los valores que poseen los 4 canales y realiza una resta de 511 (la mitad de 1023) y voltaje medido, luego se hace la escala con el producto de esta subtracción y la división de 48 entre 1023, todo esto para lograr manipular voltajes de 0 V-5 V con el objetivo de normalizar y escalar el valor máximo medido.

La función para generar las alarmas tanto en DC como en AC son muy similares, por lo que el siguiente diagrama de bloques es equivalente para ambos.

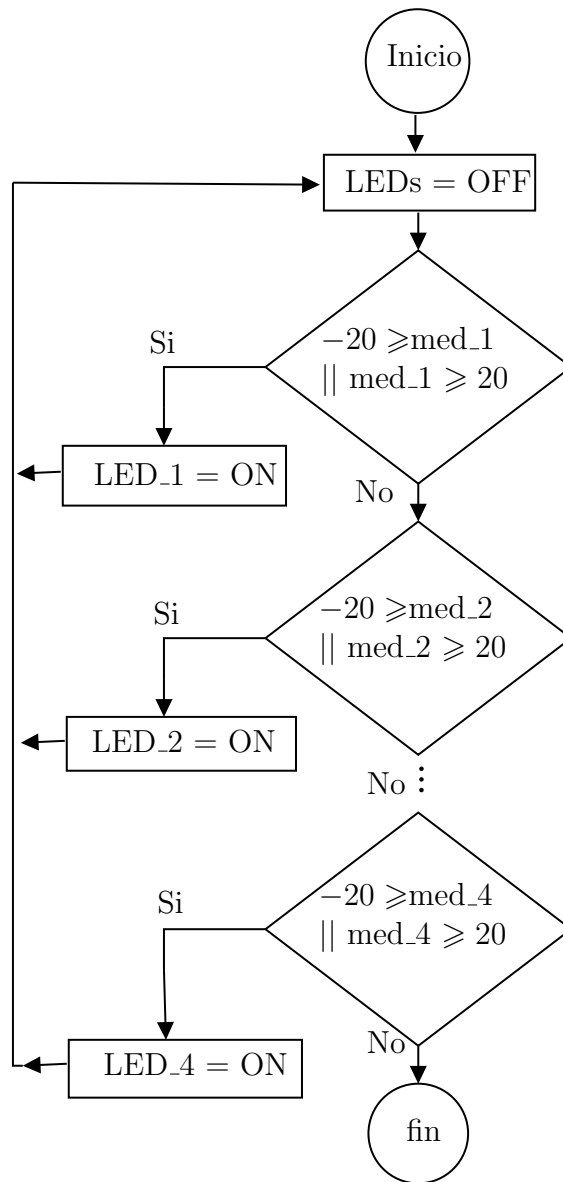


Figura 14: Diagrama de flujo alarma AC/DC.

Hasta este punto es posible mostrar el correcto funcionamiento del voltímetro con todos sus componentes. Primero se comprueba la medición de voltajes en DC.

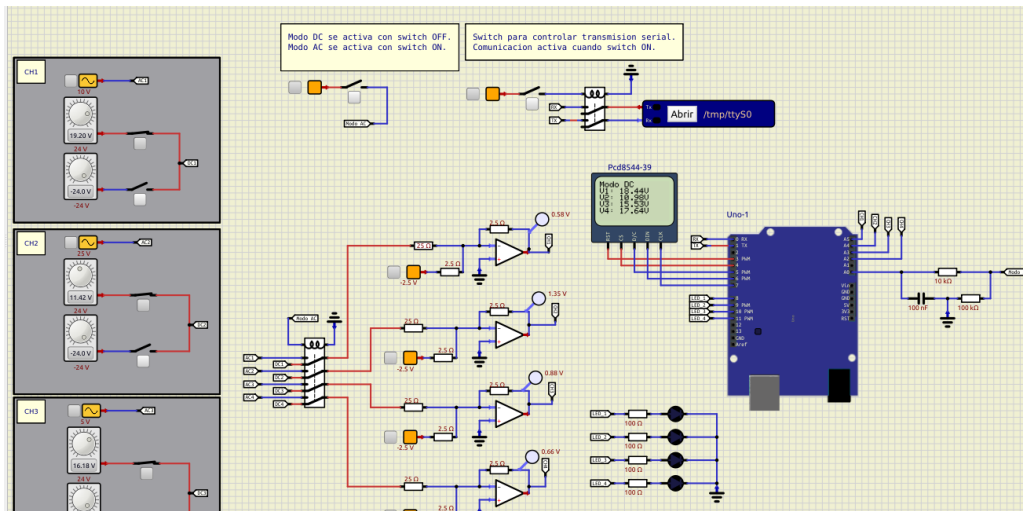


Figura 15: Medición de tensión DC de los 4 canales.

Se puede observar que la pantalla muestra las tensiones eléctricas en DC correctamente, es decir, el voltímetro es capaz de medir cuatro canales apropiadamente. Si se deseara ver la activación de los LEDs cuando se superan los umbrales de  $[-20, 20]$  basta con cambiar esto desde las fuentes. En la figura 16 se demuestra el resultado esperado.

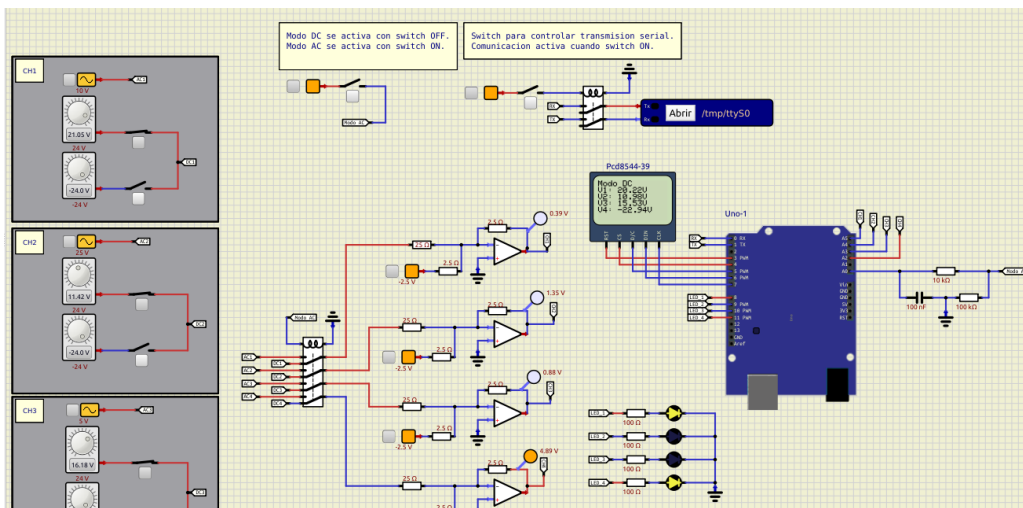


Figura 16: Activación de alarma de los LEDs.

Hecho esto, ahora se muestran las pruebas con tensión eléctrica en modo AC. Para realizar esta acción, se debe presionar el botón conectado al pin A0 para ver los siguientes resultados.

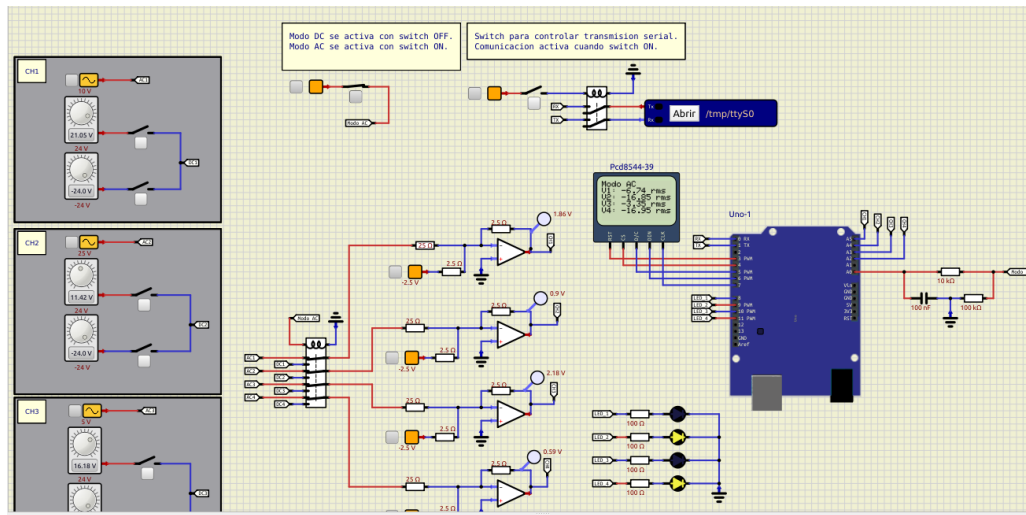


Figura 17: Medición de tensión AC de los 4 canales.

De la misma manera, note que en la figura 17 el modo AC también posee el sistema de alarmas para cuando los LEDs están fuera de los rangos permitidos:  $[-14.14, 14.14]$ .

Para la comunicación serial se realizaron los pasos correspondientes vistos en clases, para ello primeramente se configura el script con los comando para asignar los puertos, el puerto S0 es el que realiza la comunicación con el arduino y el puerto S1 es el que se diseña para poder hacer la comunicación con el script de python y guardar los datos medidos en el archivo csv. Primero se intentó establecer la comunicación entre el arduino y el puerto S0, de ahí que en la imagen 11 la dirección `/tmp/ttyS0`, la comprobación correcta del funcionamiento se verifica en la siguiente imagen:

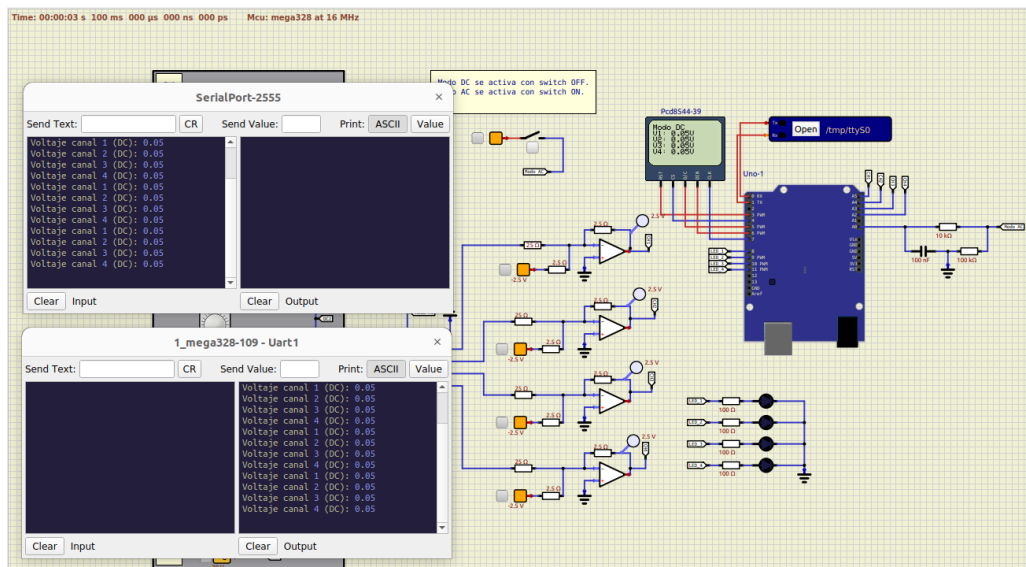


Figura 18: Verificación de la comunicación entre el arduino y el puerto serial.

Una vez comprobado el funcionamiento correcto de la comunicación anterior se procede a crear un puerto virtual para que pueda haber intercambio de información entre el puerto serial y el puerto serial virtual, S1, para ello se debe correr el bash script con el siguiente comando:

`socat PTY,link=/tmp/ttyS0,raw,echo=0 PTY,link=/tmp/ttyS1,raw,echo`

Con este comando se crea el puerto virtual necesario, por lo que finalmente se procede a escribir un programa sencillo en python encargado de abrir un archivo csv y escribir en ello los datos correspondientes del puerto serial.

Los pasos para correr todo entonces es, primero correr el bash script, seguidamente abrir la comunicación en el simulador haciendo click en la opción de "Open", correr el script de python y finalmente correr la simulación, basado en este proceso se obtienen los siguientes resultados:

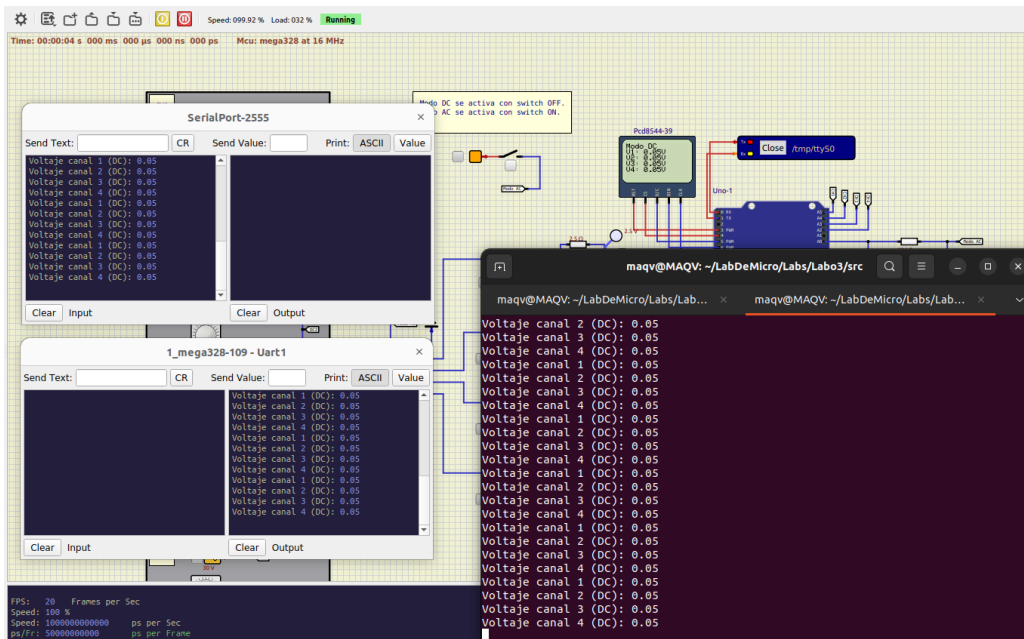


Figura 19: Verificación de la comunicación entre el puerto serial y el puerto virtual.

Realizando unos cambios en las fuentes de alimentación, se demuestra que por consola se obtienen los resultados esperados. Haciendo la medición en DC.

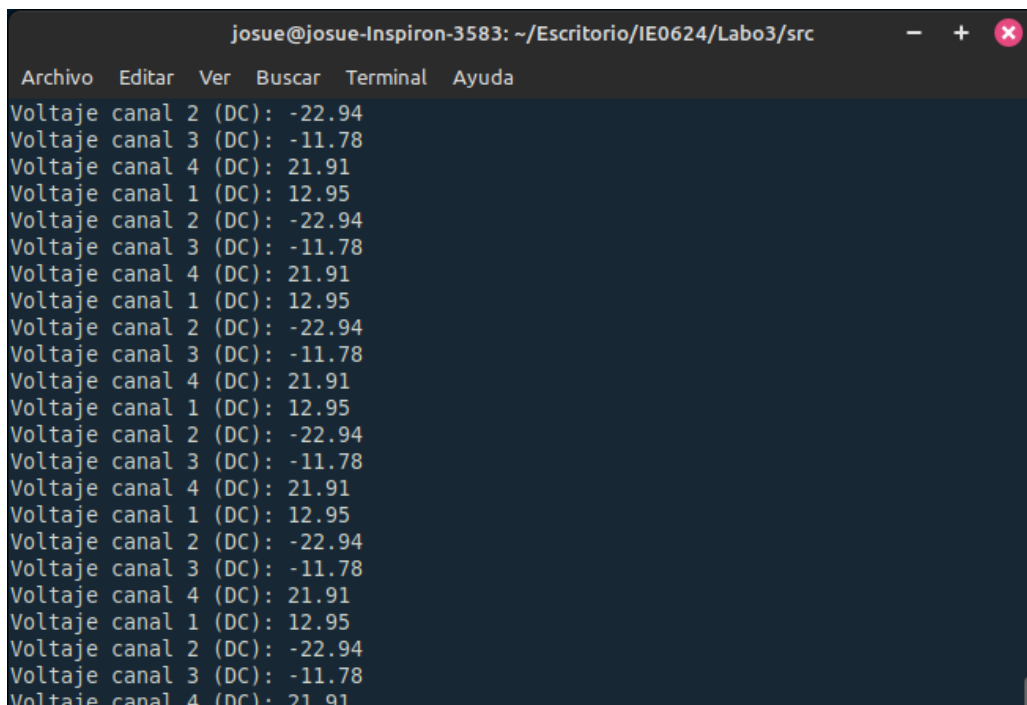


Figura 20: Medición y comunicación serial tensión DC.

Luego, la medición en AC.

```

josue@josue-Inspiron-3583: ~/Escritorio/IE0624/Labo3/src
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
Voltaje canal 3 (AC): -3.35
Voltaje canal 4 (AC): -16.95
Voltaje canal 1 (AC): -6.74
Voltaje canal 2 (AC): -16.85
Voltaje canal 3 (AC): -3.35
Voltaje canal 4 (AC): -16.95
Voltaje canal 1 (AC): -6.74
Voltaje canal 2 (AC): -16.89
Voltaje canal 3 (AC): -3.35
Voltaje canal 4 (AC): -16.95
Voltaje canal 1 (AC): -6.74
Voltaje canal 2 (AC): -16.89
Voltaje canal 3 (AC): -3.35
Voltaje canal 4 (AC): -16.95
Voltaje canal 1 (AC): -6.74
Voltaje canal 2 (AC): -16.85
Voltaje canal 3 (AC): -3.35
Voltaje canal 4 (AC): -16.95
Voltaje canal 1 (AC): -6.74

```

Figura 21: Medición y comunicación serial tensión AC.

Finalmente, se obtiene el archivo .csv generado a partir de la comunicación establecida.

	A	B	C	D	E	F	G	H	I	J	K	L
21	Voltaje canal 1 (DC): 12.95	Voltaje canal 2 (DC): -22.94	Voltaje canal 3 (DC): -11.78	Voltaje canal 4 (DC): 21.91								
22	Voltaje canal 1 (DC): 12.95	Voltaje canal 2 (DC): -22.94	Voltaje canal 3 (DC): -11.78	Voltaje canal 4 (DC): 21.91								
23	Voltaje canal 1 (DC): 12.95	Voltaje canal 2 (DC): -22.94	Voltaje canal 3 (DC): -11.78	Voltaje canal 4 (DC): 21.91								
24	Voltaje canal 1 (DC): 12.95	Voltaje canal 2 (DC): -22.94	Voltaje canal 3 (DC): -11.78	Voltaje canal 4 (DC): 21.91								
25	Voltaje canal 1 (DC): 12.95	Voltaje canal 2 (DC): -22.94	Voltaje canal 3 (DC): -11.78	Voltaje canal 4 (DC): 21.91								
26	Voltaje canal 1 (DC): 12.95	Voltaje canal 2 (DC): -22.94	Voltaje canal 3 (DC): -11.78	Voltaje canal 4 (DC): 21.91								
27	Voltaje canal 1 (DC): 12.95	Voltaje canal 2 (DC): -22.94	Voltaje canal 3 (DC): -11.78	Voltaje canal 4 (DC): 21.91								
28	Voltaje canal 1 (DC): 12.95	Voltaje canal 2 (DC): -22.94	Voltaje canal 3 (DC): -11.78	Voltaje canal 4 (DC): 21.91								
29	Voltaje canal 1 (DC): 12.95	Voltaje canal 2 (DC): -22.94	Voltaje canal 3 (DC): -11.78	Voltaje canal 4 (DC): 21.91								
30	Voltaje canal 1 (DC): 12.95	Voltaje canal 2 (DC): -22.94	Voltaje canal 3 (DC): -11.78	Voltaje canal 4 (DC): 21.91								
31	Voltaje canal 1 (DC): 12.95	Voltaje canal 2 (DC): -22.94	Voltaje canal 3 (DC): -11.78	Voltaje canal 4 (DC): 21.91								
32	Voltaje canal 1 (DC): 12.95	Voltaje canal 2 (DC): -22.94	Voltaje canal 3 (DC): -11.78	Voltaje canal 4 (DC): 21.91								
33	Voltaje canal 1 (DC): 12.95	Voltaje canal 2 (DC): -22.94	Voltaje canal 3 (DC): -11.78	Voltaje canal 4 (DC): 21.91								
34	Voltaje canal 1 (DC): 12.95	Voltaje canal 2 (DC): -22.94	Voltaje canal 3 (DC): -11.78	Voltaje canal 4 (DC): 21.91								
35	Voltaje canal 1 (DC): 12.95	Voltaje canal 2 (DC): -22.94	Voltaje canal 3 (DC): -11.78	Voltaje canal 4 (DC): 21.91								
36	Voltaje canal 1 (DC): 12.95	Voltaje canal 2 (DC): -22.94	Voltaje canal 3 (DC): -11.78	Voltaje canal 4 (DC): 21.91								
37	Voltaje canal 1 (DC): 12.95	Voltaje canal 2 (DC): -22.94	Voltaje canal 3 (DC): -11.78	Voltaje canal 4 (DC): 21.91								
38	Voltaje canal 1 (DC): 12.95	Voltaje canal 2 (DC): -22.94	Voltaje canal 3 (DC): -11.78	Voltaje canal 4 (DC): 21.91								
39	Voltaje canal 1 (AC): -6.74	Voltaje canal 2 (AC): -16.89	Voltaje canal 3 (AC): -3.35	Voltaje canal 4 (AC): -16.95								
40	Voltaje canal 1 (AC): -6.74	Voltaje canal 2 (AC): -16.89	Voltaje canal 3 (AC): -3.35	Voltaje canal 4 (AC): -16.95								
41	Voltaje canal 1 (AC): -6.74	Voltaje canal 2 (AC): -16.85	Voltaje canal 3 (AC): -3.35	Voltaje canal 4 (AC): -16.95								
42	Voltaje canal 1 (AC): -6.74	Voltaje canal 2 (AC): -16.89	Voltaje canal 3 (AC): -3.35	Voltaje canal 4 (AC): -16.95								
43	Voltaje canal 1 (AC): -6.74	Voltaje canal 2 (AC): -16.89	Voltaje canal 3 (AC): -3.35	Voltaje canal 4 (AC): -16.95								
44	Voltaje canal 1 (AC): -6.74	Voltaje canal 2 (AC): -16.89	Voltaje canal 3 (AC): -3.35	Voltaje canal 4 (AC): -16.95								
45	Voltaje canal 1 (AC): -6.74	Voltaje canal 2 (AC): -16.85	Voltaje canal 3 (AC): -3.35	Voltaje canal 4 (AC): -16.95								
46	Voltaje canal 1 (AC): -6.74	Voltaje canal 2 (AC): -16.85	Voltaje canal 3 (AC): -3.35	Voltaje canal 4 (AC): -16.95								
47	Voltaje canal 1 (AC): -6.74	Voltaje canal 2 (AC): -16.89	Voltaje canal 3 (AC): -3.35	Voltaje canal 4 (AC): -16.95								
48	Voltaje canal 1 (AC): -6.74	Voltaje canal 2 (AC): -16.89	Voltaje canal 3 (AC): -3.35	Voltaje canal 4 (AC): -16.95								

Figura 22: Archivo csv generado.

Ya se comprobó la generación correcta de los archivos csv, con sus respectivos valores, pero si se quisiera verificar se puede consultar el siguiente repositorio: <https://github.com/JosueC07183/Labo3.git> donde encontrará todos los detalles necesarios de su funcionamiento. De tal manera, que el voltímetro funciona correctamente.

## 4. Conclusiones y recomendaciones

Para concluir,

- Los puertos son de suma importancia, ya que permiten la comunicación entre otros dispositivos.
- Lo anterior permitió realizar el trabajo de una manera más sencilla porque las conexiones de los componentes electrónicos al mcu, así como el uso de los periféricos dieron sentido a lo que se estaba programando, desde lo más básico como la declaración de una pin como entrada o salida, como otras tareas, que suenan sencillas pero complejas si se desconoce el protocolo de comunicación de la pantalla PDC8544.
- El uso de funciones ayudó a que el código se viera más ordenado, por lo que en el `loop` principal es más claro lo que está realizando. Esto es muy importante porque permitió tener un funcionamiento claro y correcto de la función del voltímetro, ya que desde las magnitudes (con cierto grado de error) asignadas en las fuentes de tensión, la normalización y el escalamiento de estos valores se mostraron correctamente en la pantalla PCD8544, y cuando se irrespetaban los umbrales de  $[-20,20]$ DC y  $[-14.14,14.14]$ AC los LEDs se encendieron, y se apagan cuando se respetaba este rango. Ahora, con respecto a la comunicación serial del arduino con la PC fue exitosa, porque se estableció la correcta comunicación ya que lo mostrado en la pantalla se fue reflejando por consola y se guardó un archivo `.csv` en el directorio del proyecto. Por lo tanto, el proyecto cumple todos los requisitos solicitados.

Como recomendación se debe entender lo básico de como funciona la librería que se vaya a utilizar para poder realizar la comunicación y estar haciendo pruebas constantemente para verificar de que se esté trabajando de la forma que se espera, esto fue notable en la parte de generación del csv al igual que observar la comunicación dentro del mismo simulador porque hubieron momentos donde los datos se estaban imprimiendo de forma incorrecta.

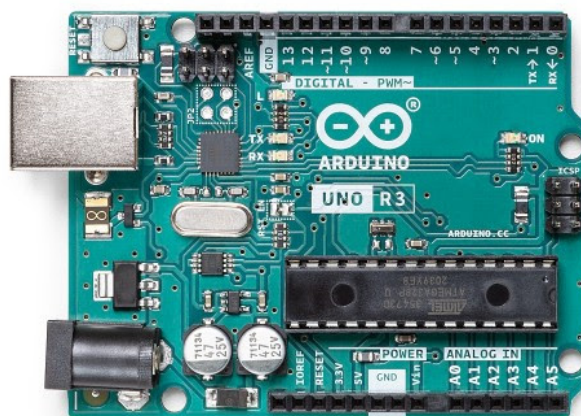
## Referencias

- [1] Atmel. Atmega328p datasheet (pdf) - atmel corporation. Atmel, <https://pdf1.alldatasheet.com/datasheet-pdf/view/241077/ATMEL/ATMEGA328P.html>, julio 2023. Accedido en enero de 2024.
- [2] Atmel. 8-bit avr microcontrollers. Atmel, <https://datasheet.octopart.com/ATMEGA328P-MU-Microchip-datasheet-65729177.pdf>, junio 2023. Accedido en enero de 2024.
- [3] Sparkfun. Datasheet pcd8544. SparkfunO, <https://www.sparkfun.com/datasheets/LCD/Monochrome/Nokia5110.pdf>, noviembre 2008. Accedido en enero de 2024.
- [4] Sparkfun. Serial peripheral interface (spi). SparkfunO, <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>, mayo 2018. Accedido en enero de 2024.
- [5] MicroJpm. Componentes disponibles. MicroJpm, <https://www.microjpm.com/>, mayo 2022. Accedido en enero de 2024.



## 5. Anexos

Aquí van las hojas del fabricante de los componentes usados para este laboratorio.



## Description

The Arduino UNO R3 is the perfect board to get familiar with electronics and coding. This versatile development board 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



# CONTENTS

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



# 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, the UNO 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; 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 R3 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 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

- 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.

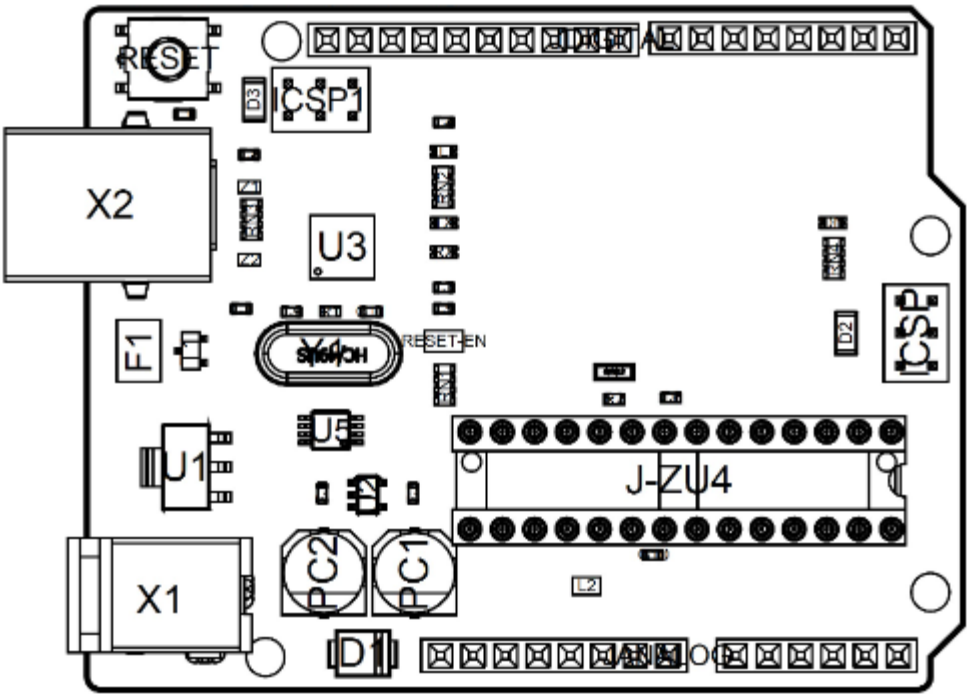
### 2.2 Power Consumption

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



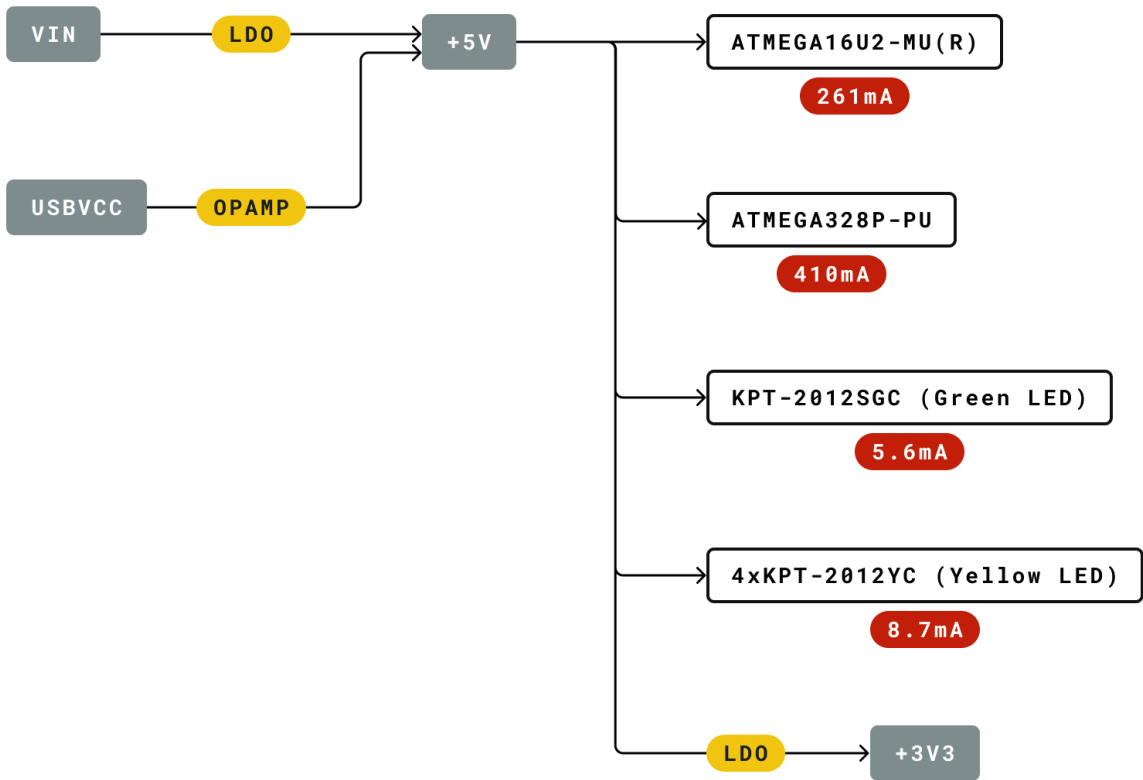
Ref.	Description	Ref.	Description
X1	Power jack 2.1x5.5mm	U1	SPX1117M3-L-5 Regulator
X2	USB B Connector	U3	ATMEGA16U2 Module
PC1	EEE-1EA470WP 25V SMD Capacitor	U5	LMV358LIST-A.9 IC
PC2	EEE-1EA470WP 25V SMD Capacitor	F1	Chip Capacitor, High Density
D1	CGRA4007-G Rectifier	ICSP	Pin header connector (through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

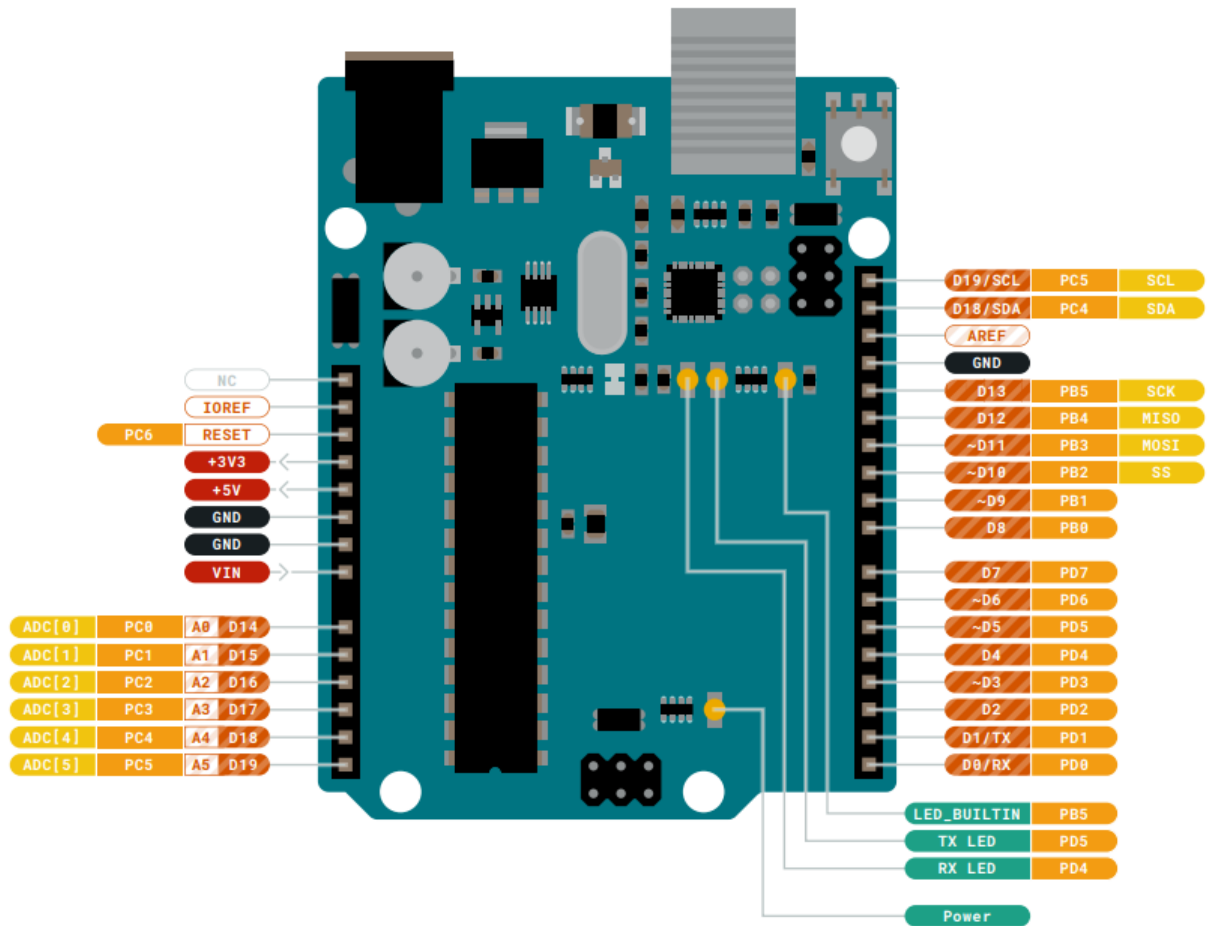
### 4.3 Sample Sketches

Sample sketches for the Arduino 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

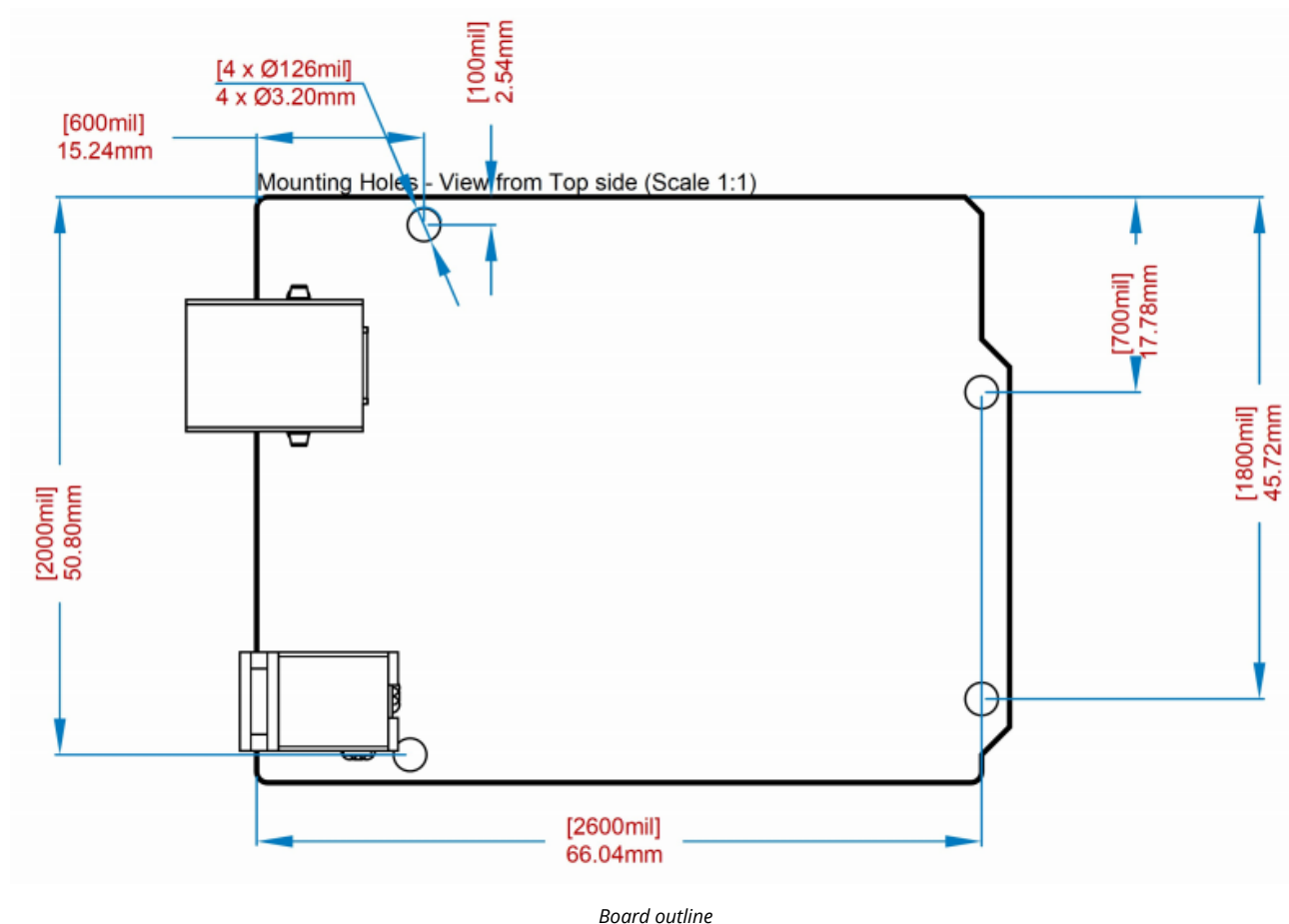
Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	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 IDE (Cloud)	<a href="https://create.arduino.cc/editor">https://create.arduino.cc/editor</a>
Cloud IDE Getting Started	<a href="https://create.arduino.cc/projecthub/Arduino_Genuino/getting-started-with-arduino-web-editor-4b3e4a">https://create.arduino.cc/projecthub/Arduino_Genuino/getting-started-with-arduino-web-editor-4b3e4a</a>
Arduino Website	<a href="https://www.arduino.cc/">https://www.arduino.cc/</a>
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>
Online Store	<a href="https://store.arduino.cc/">https://store.arduino.cc/</a>

## 10 Revision History

Date	Revision	Changes
26/07/2023	2	General Update
06/2021	1	Datasheet release

## Features

- High Performance, Low Power AVR<sup>®</sup> 8-Bit Microcontroller
- 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 20 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory (ATmega48P/88P/168P/328P)
  - 256/512/512/1K Bytes EEPROM (ATmega48P/88P/168P/328P)
  - 512/1K/1K/2K Bytes Internal SRAM (ATmega48P/88P/168P/328P)
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - 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
- 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
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Byte-oriented 2-wire Serial Interface (Philips I<sup>2</sup>C compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - 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 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)



**8-bit AVR<sup>®</sup>  
Microcontroller  
with 4/8/16/32K  
Bytes In-System  
Programmable  
Flash**

**ATmega48P/V  
ATmega88P/V  
ATmega168P/V  
ATmega328P**

**Preliminary**

**Summary**

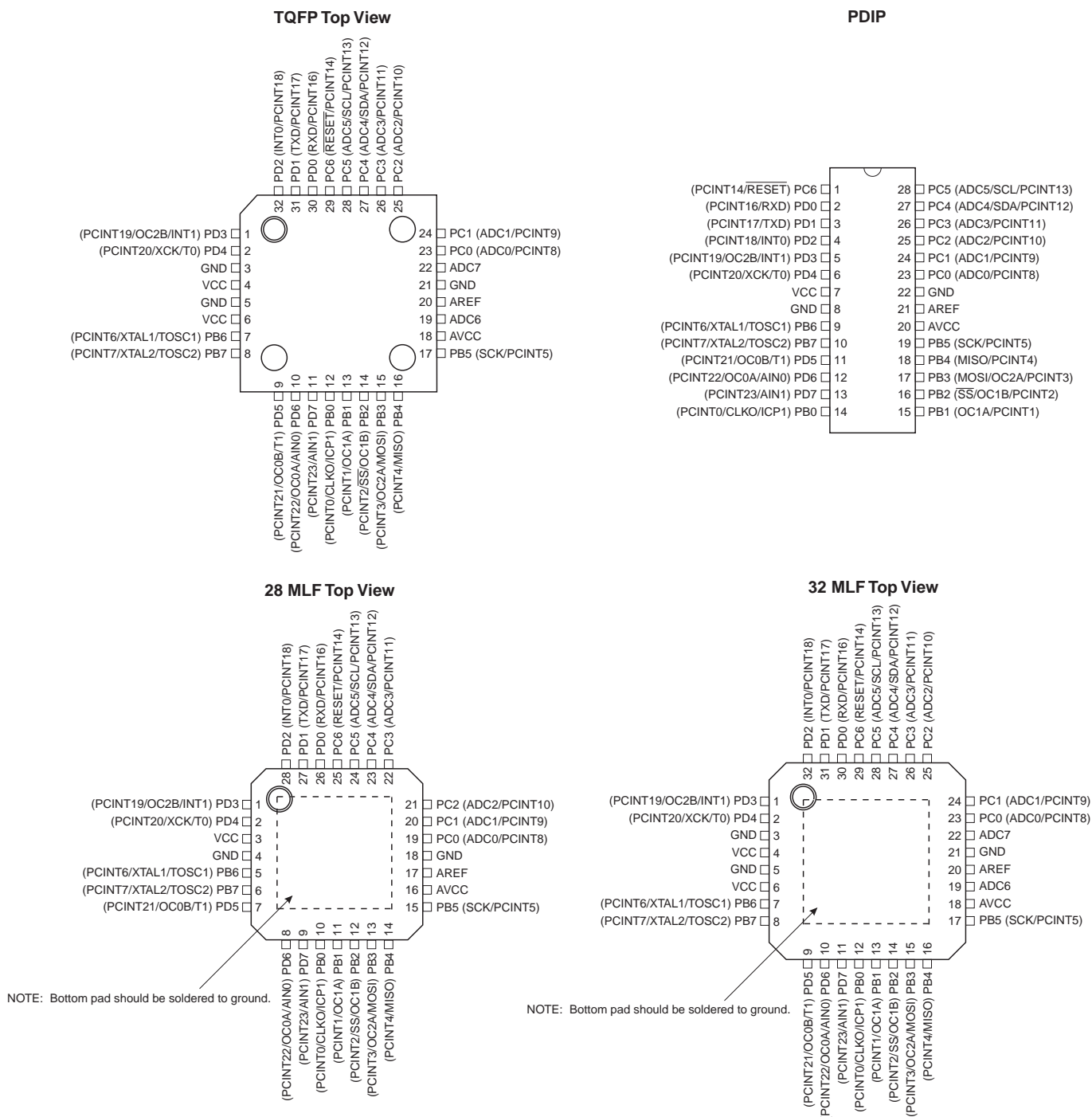
Rev. 8025FS-AVR-08/08





## 1. Pin Configurations

Figure 1-1. Pinout ATmega48P/88P/168P/328P



## 1.1 Pin Descriptions

### 1.1.1 VCC

Digital supply voltage.

### 1.1.2 GND

Ground.

### 1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in ["Alternate Functions of Port B" on page 82](#) and ["System Clock and Clock Options" on page 26](#).

### 1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

### 1.1.5 PC6/ $\overline{\text{RESET}}$

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in [Table 28-3 on page 320](#). Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in ["Alternate Functions of Port C" on page 85](#).

### 1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.



The various special features of Port D are elaborated in ["Alternate Functions of Port D" on page 88](#).

#### **1.1.7 $AV_{CC}$**

$AV_{CC}$  is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to  $V_{CC}$ , even if the ADC is not used. If the ADC is used, it should be connected to  $V_{CC}$  through a low-pass filter. Note that PC6..4 use digital supply voltage,  $V_{CC}$ .

#### **1.1.8 AREF**

AREF is the analog reference pin for the A/D Converter.

#### **1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)**

In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

### **1.2 Disclaimer**

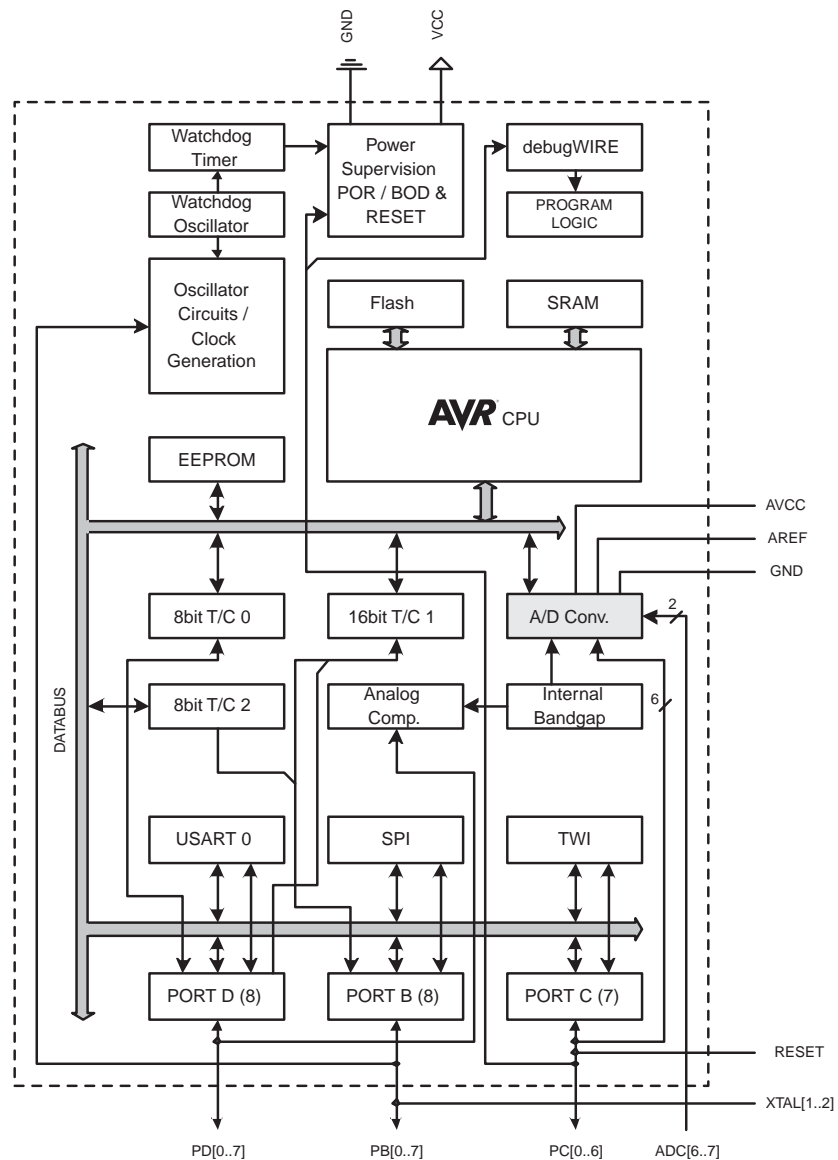
Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

## 2. Overview

The ATmega48P/88P/168P/328P 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 ATmega48P/88P/168P/328P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

### 2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting



architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega48P/88P/168P/328P provides the following features: 4K/8K/16K/32K bytes of In-System Programmable Flash with Read-While-Write capabilities, 256/512/512/1K bytes EEPROM, 512/1K/1K/2K bytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte-oriented 2-wire Serial Interface, an SPI serial port, a 6-channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages), a programmable Watchdog Timer with internal Oscillator, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, USART, 2-wire Serial Interface, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption.

The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega48P/88P/168P/328P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega48P/88P/168P/328P AVR is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits.

## 2.2 Comparison Between ATmega48P, ATmega88P, ATmega168P, and ATmega328P

The ATmega48P, ATmega88P, ATmega168P, and ATmega328P differ only in memory sizes, boot loader support, and interrupt vector sizes. [Table 2-1](#) summarizes the different memory and interrupt vector sizes for the three devices.

**Table 2-1.** Memory Size Summary

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48P	4K Bytes	256 Bytes	512 Bytes	1 instruction word/vector
ATmega88P	8K Bytes	512 Bytes	1K Bytes	1 instruction word/vector
ATmega168P	16K Bytes	512 Bytes	1K Bytes	2 instruction words/vector
ATmega328P	32K Bytes	1K Bytes	2K Bytes	2 instructions words/vector

ATmega88P, ATmega168P, and ATmega328P support a real Read-While-Write Self-Programming mechanism. There is a separate Boot Loader Section, and the SPM instruction can only execute from there. In ATmega48P, there is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash.