

UNIVERSIDAD DE COSTA RICA
Escuela de Ingeniería Eléctrica
IE-0624 Laboratorio de Microcontroladores

Laboratorio 3

Arduino: PID,GPIO,ADC y comunicaciones

Dylan Agüero Carrillo B90083
Noel Blandón Saborío B61097

Grupo 01

Profesor: MSc. Marco Villalta Fallas

8 de mayo de 2024

1. Introducción

En el presente reporte de laboratorio se presentan los resultados relacionados a la aplicación de una incubadora de huevos, mediante el microcontrolador Arduino UNO y la visualización en una pantalla LCD PCD8544. Una vez realizado esto, se envían los datos hacia la computadora mediante un puerto serial con USART para posteriormente manipular los datos obtenidos mediante un código en Python.

De esta manera se pudo integrar los conceptos relacionados a control con un PID, y se pudo desarrollar una aplicación efectiva de una incubadora de huevos mediante las prestaciones que brinda el microcontrolador Arduino UNO.

Link del repositorio: <https://github.com/dagueroc/Laboratorio-de-Microcontroladores.git>

2. Nota teórica

2.1. Especificaciones de Arduino UNO

Arduino es una plataforma de hardware libre conformada por un microcontrolador, un entorno de desarrollo que permite desarrollar proyectos o prototipos en electrónica. Los modelos de las placas de Arduino han cambiado a lo largo de los años, en donde diferentes microcontroladores han sido empleados, y a su vez la memoria disponible se ha cambiado.[1]

Para el Arduino UNO se tiene la siguiente placa con sus respectivos puertos y periféricos.

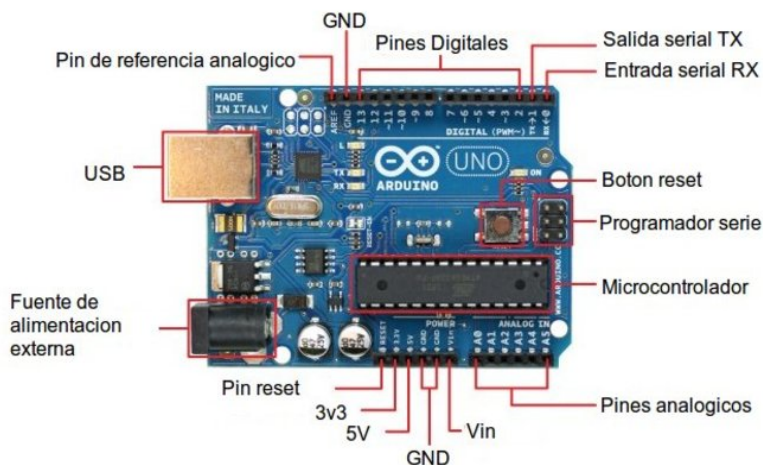


Figura 1: Placa Arduino UNO[1]

El Arduino presenta diferentes componentes y puertos con diferentes funciones para comunicaciones o instrucciones[1]:

- Puerto USB, que se utilizan para la energía de la placa
- Botón de reinicio, que tiene como función resetear el microcontrolador.
- LED TX y RX, es para verificar si existe comunicación entre la computadora y la placa.
- LED (Pin de referencia analógica), es un activador que se encuentra en forma predeterminada en la placa.
- Pines Digitales (13), se utilizan con entrada y salida en la placa de Arduino.
- LED de corriente, es para verificar que la placa reciba energía en forma correcta.
- Microcontrolador ATmega, en el circuito principal de la placa.

Un puerto de importancia relevante para las comunicaciones es el serial. Estos utilizan secuencias de bits para transmitir mensajes. Para hacer esto, necesita al menos dos conectores, RX (recepción) y TX (transmisión), para la transmisión de datos.[2]

El protocolo de comunicación serial de dispositivo a dispositivo es por medio de UART que son las siglas en inglés de *Universal Asynchronous Receiver-Transmitter (Transmisor-Receptor Asíncrono Universal)*. Por medio de este protocolo se puede enviar información a la computadora para posteriormente procesarla o manipularla. [3]

UART funciona enviando datos como una secuencia de bits, incluido un bit de inicio, un bit de datos, un bit de paridad opcional y bits de parada. A diferencia de la comunicación paralela, que envía varios bits a la vez, la UART envía datos en serie, un bit a la vez[3].

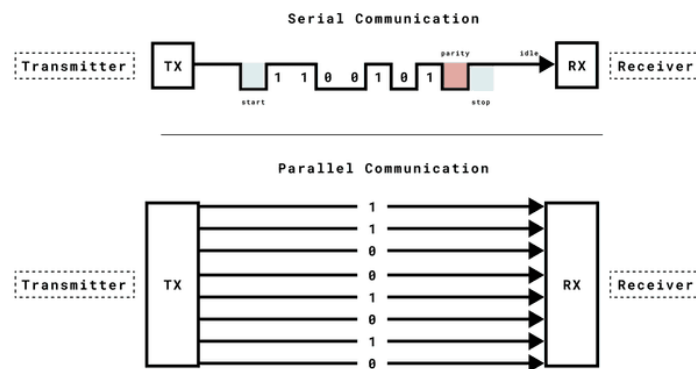


Figura 2: Transmisión serial/paralela UNO[3]

Tabla 1: Especificaciones Arduino UNO

Microcontrolador:	ATmega328P
Voltaje de operación:	5V
Entrada de alimentación:	7-12V
Pines digitales I/O:	14
Pines PWM:	6
Pines analógicos:	6
Corriente por pin:	20 mA
Corriente del pin 3.3.V:	50 mA max
Memoria Flash:	32 KB (0.5 KB usados por el bootloader)
SRAM:	2 KB
EEPROM:	1 KB
Velocidad de reloj:	16 MHz
LED programable integrado:	13
Dimensiones:	68.6 x 53.4 mm
Peso:	25 gramos

2.2. ATMEGA328P

El ATMEGA328P es un microcontrolador de 8 bits de alto rendimiento que combina memoria flash ISP de 32 kB con funciones de lectura y escritura, EEPROM 1024B, 23 líneas de E/S de uso general, 32 funciones de programa y 3 contadores. Incluye modo comparador, interrupciones internas y externas, USART programable, interfaz serial de bytes de 2 cables, puerto serial SPI, convertidor A/D de 10 bits y 6 conmutadores. [4]

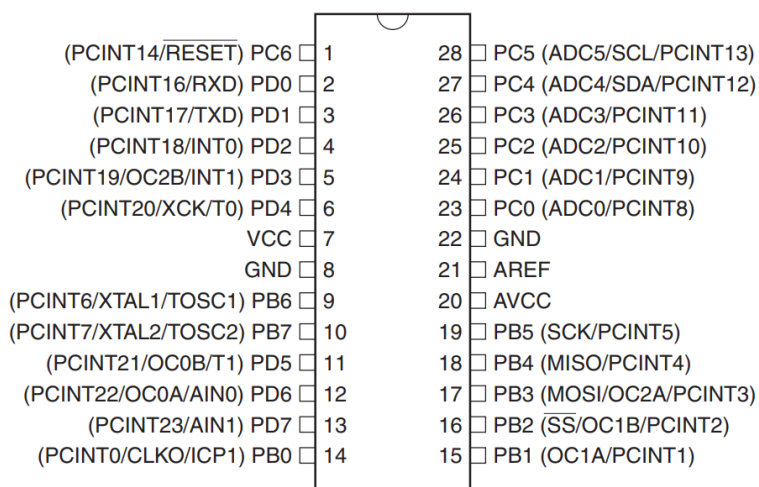


Figura 3: Diagrama del ATMEGA328P [4]

A continuación, se tienen las especificaciones técnicas del microcontrolador:

Tabla 2: Especificaciones ATMEGA328P

Fabricante	Atmel/Microchip
Voltaje de operación	1.8 V a 5.5 V
Frecuencia máxima de reloj	20 MHz
Tamaño de la memoria del programa	32 kB
Tamaño de la RAM de datos	2 kB
Resolución del ADC	10 bits
Número de canales ADC	6
Número de entradas/salidas	23
Cantidad de timers	2x8 bit, 1x16 bit
Temperatura de operación	- 40°C a 85°C
Interfaz de comunicación	I2C, SPI, USART
Altura	4,57 mm.
Longitud	34.8 mm
Ancho	7.49 mm
Peso unitario	4,19 gr
Encapsulado	De inserción (PDIP-28)

2.2.1. Diagrama

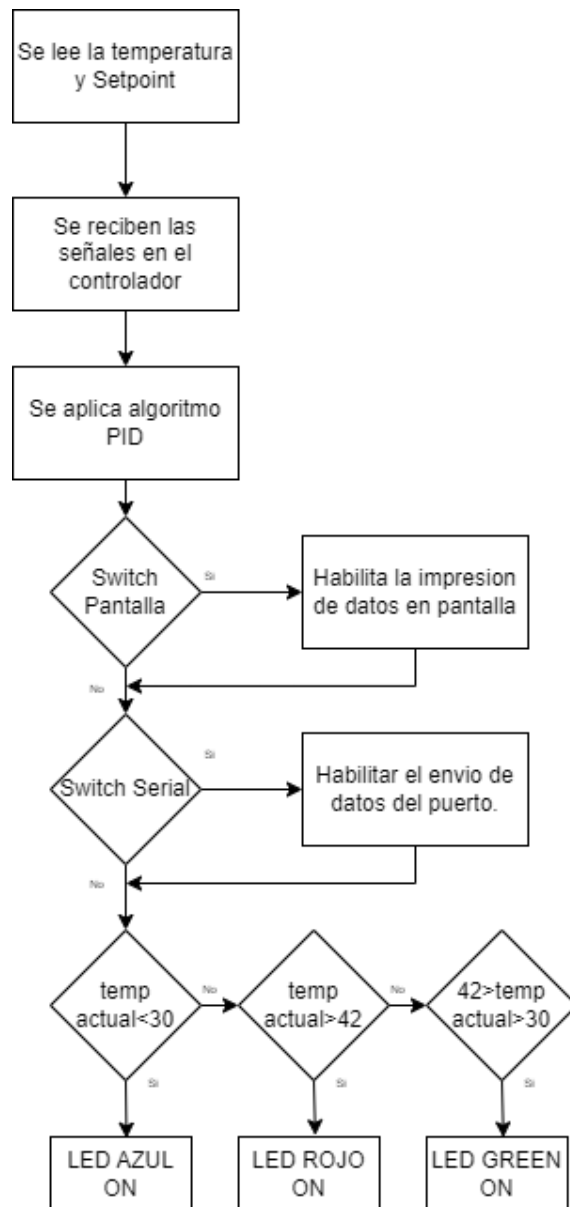


Figura 4: Diagrama

2.3. Sistema de control PID

Para el algoritmo de control PID, se inicia estudiando un lazo de control cerrado, como el siguiente:

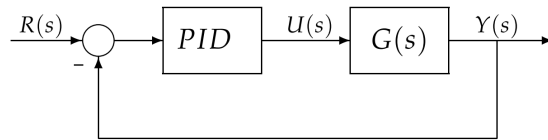


Figura 5: Lazo de control [5]

Los controladores PID, incluyen tres acciones: proporcional(P), integral (I) y derivativa (D). [5]

- **P: Acción de control proporcional.**

Salida del controlador que es proporcional al error.[5]

$$C_p(s) = K_p \quad (1)$$

- **I: Acción de control integral.**

Una salida del controlador que es proporcional al error acumulado, lo que implica que es un modo de controlar lento. La señal de control $u(t)$ tiene un valor distinto de cero cuando la señal de error $e(t)$ es cero. Por lo tanto, se puede concluir que el error de equilibrio es cero para un valor de referencia constante o para las perturbaciones. [5]

$$u(t) = K_i \int_0^t e(\tau) d\tau \quad (2)$$
$$C_i(s) = \frac{K_i}{s}$$

- **D: Acción de control derivativa.**

Esta función tiene propiedades predictivas, lo que hace que el control sea más rápido, aunque tiene la gran desventaja de amplificar señales ruidosas y puede causar saturación en el actuador. La función de control de derivadas nunca se utiliza sola porque sólo es efectiva durante los períodos transitorios. [5]

$$u(t) = K_p T_d \frac{de(t)}{dt} \quad (3)$$
$$C_d = s K_p T_d$$

Para el diseño del PID en el laboratorio se emplearon los siguientes valores:

- $P = 1$
- $I = 0.1$
- $D = 0.1$

2.4. Diseño de componentes del circuito para evitar rebotes

Al emplear un botón, éste provocan la presencia de rebotes indeseados a la hora de ejecutar el programa, es por ello que se emplea un circuito RC que permita eliminar los rebotes[6]. Se tiene la siguiente relación:

$$\tau = RC \quad (4)$$

Para un tiempo de aproximadamente 0.1 segundos, se tiene que fijar al menos una de las variables en la ecuación, en este caso se define el resistor en $1k\Omega$. Con esto se puede calcular el valor del capacitor:

$$\begin{aligned} 0,1s &= 1k\Omega \cdot C \\ C &= \frac{0,1s}{1k\Omega} = 100\mu C \end{aligned} \quad (5)$$

3. Desarrollo/Análisis de resultados

3.1. Análisis electrónico

Se tiene el siguiente diagrama de la incubadora construido en SimulIDE, en la cual se puede apreciar los 2 interruptores donde uno controlaría la conexión serial y el otro la comunicación de la pantalla. En los cuales se utilizaron capacitores para eliminar el efecto rebote produce el mecanismo al accionarse.

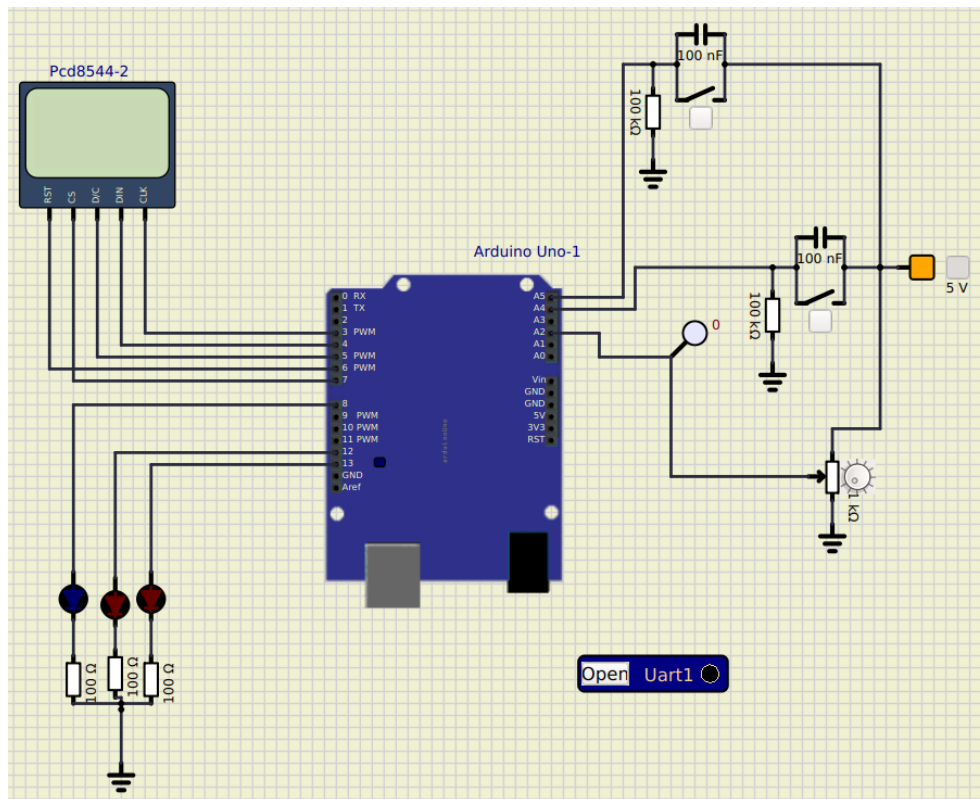


Figura 6: Esquemático de la Incubadora

Se empleó un potenciómetro para controlar la señal de punto de operación de la incubadora. La función de este es variar la tensión eléctrica entre $[0V-5V]$ en la entrada del pin A2.

3.2. Análisis del programa

3.2.1. Librerías empleadas

En el encabezado del script del Arduino se incluyeron las siguientes librerías para la ejecución del programa:

- PID v1: Permite emplear un control PID, la cual encapsula las funciones que ejecutan el

algoritmo PID. Esto facilita la comprensión del código y disminuye la complejidad que conlleva realizar de manera manual el controlador.

- SPI: Permite la comunicación entre un Arduino y otro dispositivo a través de SPI, con el Arduino como dispositivo controlador.
- Adafruit PCD8544: Se emplea para el display LCD monocromático del Nokia 5110. Se utiliza para comunicar de una manera más sencilla la información desde el Arduino al Display.
- Adafruit GFX: Es una dependencia que se encarga de la parte grafica de la librería de la PCD8544
- math: Se emplea para funciones matemáticas útiles.

3.2.2. Definición de Variables

En esta primera sección de código se instancia los nombres que van a identificar las variables y el pin correspondiente de ellas.

```
// Definición de pines
int CLK      = 3;
int DIN      = 4;
int DC       = 5;
int CS       = 7;
int RST      = 6;

int BLUE     = 12;
int GREEN    = 13;
int RED      = 8;

int SWITCHP  = A5;
int SWITCHU  = A4;

int POTENCIOMETRO = A2;
```

Seguidamente se instanciaron las siguientes variables y métodos:

```
// Definición de variables
int Temperatura_Operacion;
double temp_ac, setPoint, ctrl_sgn, output_sgn;

// Controlador PID
PID pid(&temp_ac, &ctrl_sgn, &setPoint, Kp, Ki, Kd, DIRECT);
```

```
// Objeto PCD
Adafruit_PCD8544 display = Adafruit_PCD8544(CLK,DIN,DC,CS,RST);
```

También se instancia en ésta sección el método de la planta proporcionado en el enunciado del laboratorio. El método de la pantalla recibe los valores correspondientes de los pines de conexión de la pantalla que tiene exactamente los mismos nombres de las variables nombradas para una fácil implementación. En el caso del método del algoritmo PID se referencia sus valores con punteros hacia su posición en memoria de las variables que necesita, además se establece la dirección en la cual va a actuar el controlador PID con DIRECT, significa que la salida aumentará cuando el error, entre temp ac y setPoint, sea positivo.

3.2.3. Configuración

El método void setup() en Arduino es una función crucial que se ejecuta una sola vez al inicio del programa, antes de que entre en el bucle principal. Su función principal es inicializar todos los componentes de hardware y software que se utilizarán en el proyecto. El caso más común es el pinMode, que indica si el pin es una salida o entrada. Se emplea de la siguiente manera:

```
void setup() {
  Serial.begin(9600);
  pinMode(CLK, OUTPUT);
  pinMode(DIN, OUTPUT);
  pinMode(DC, OUTPUT);
  pinMode(CS, OUTPUT);
  pinMode(RST, OUTPUT);
  pinMode(BLUE, OUTPUT);
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(SWITCHU, INPUT);
  pinMode(SWITCHP, INPUT);

  Temperatura_Operacion = analogRead(POTENCIOMETRO);
  setPoint = map(Temperatura_Operacion, 0, 1023, 20, 80);

  output_sgn = 0;

  pid.SetOutputLimits(-10, 80);
  pid.SetMode(AUTOMATIC);

  display.begin();
}
```

El método Serial.begin(9600); toma un argumento, que es la tasa de baudios. En este caso, 9600 indica que los datos se transferirán a 9600 bits por segundo. Establece una conexión y define la tasa de baudios, que es la velocidad a la que se transmiten los datos entre los dispositivos.

También se define el mapeo, como se le conoce popularmente, la asignación de los valores que va a obtener el pin analógico y la normalización que va a tomar el Arduino. Que este va a recibir interpreta la señal de entrada del potenciómetro en un rango de [0 - 1023] y los normaliza al rango de operación de la incubadora que es de [20 - 80].

También se hace uso del método SetOutputLimits de la biblioteca PID, el cual define la salida en un rango predeterminado. Para efectos de tener una menor oscilación al descender la temperatura se limitó la menor en -10.

3.2.4. Código principal

La función void loop() en Arduino es el núcleo de la ejecución continua del programa. Para determinar los valores de setPoint se sigue el mismo procedimiento implementado en la función setup(). Pero para la temperatura actual primero se pasa por el método que simula la planta:

```
temp_ac = simPlanta(ctrl_sgn);
```

Para actualizar la señal de control se debe ejecutar la función Compute() del objeto PID.

Seguidamente se muestra la sección del código que permite visualizar los datos de interés en la pantalla:

```
display.clearDisplay();
display.setCursor(0,0);
// Imprime temperatura de operacion en el display
display.print("Temp ST: ");
display.println(setPoint);

// Imprime salida del controlador en el display
display.print("PID sgn: ");
display.println(ctrl_sgn);

// Imprime temperatura actual en el display
display.print("Temp ac: ");
display.println(temp_ac);

if(digitalRead(SWITCHP) == HIGH){
    display.display();
}
else {
    display.clearDisplay();
    display.display();
}
```

Los condicionales de dicho código borra los datos en la pantalla si el switch se encuentra en bajo, si está en alto muestra todos los datos de interés.

Ya conociendo el valor de la temperatura actual de la incubadora se procede a programar los condicionales que van a controlar los LEDs:

```
// LEDs
if (temp_ac <= 30){
    digitalWrite(BLUE, HIGH);
}
else {
    digitalWrite(BLUE, LOW);
}
if (temp_ac >= 42){
    digitalWrite(RED, HIGH);
}
else {
    digitalWrite(RED, LOW);
}
if (temp_ac >= 30 && temp_ac <= 42){
    digitalWrite(GREEN, HIGH);
}
else{
    digitalWrite(GREEN, LOW);
}
```

Y finalmente, se envían los datos de interés a la PC, en cada ciclo, a través de comunicación serial. No obstante, esta acción esta condicionada al estado del pin A4 (SWITCH).

```
if(digitalRead(SWITCHU) == HIGH){
    Serial.print(temp_ac);
    Serial.print(",");
    Serial.print(setPoint);
    Serial.print(",");
    Serial.println(ctrl_sgn);
}
else {
    Serial.print("NULL");
    Serial.print(",");
    Serial.print("NULL");
    Serial.print(",");
    Serial.print("NULL");
    Serial.print(",");
    Serial.println("NULL");
}
```

Se empleó un script en Python para leer la información serial enviada por el Arduino a la PC (a través de una simulación) y la decodifica escribiéndola en un archivo .csv.

3.3. Muestra del funcionamiento

En la siguiente imagen se logra comprobar el funcionamiento de la incubadora. La cual al estar a una temperatura de 54°C se enciende el LED rojo:

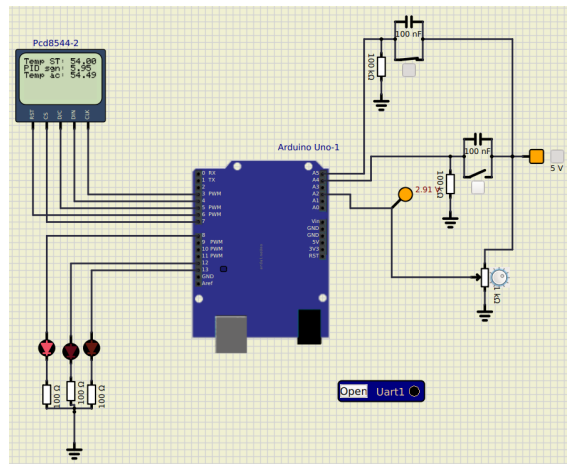


Figura 7: Led rojo

En la siguiente imagen se logra comprobar el funcionamiento de la incubadora. La cual al estar a una temperatura de 35°C se enciende el LED verde:

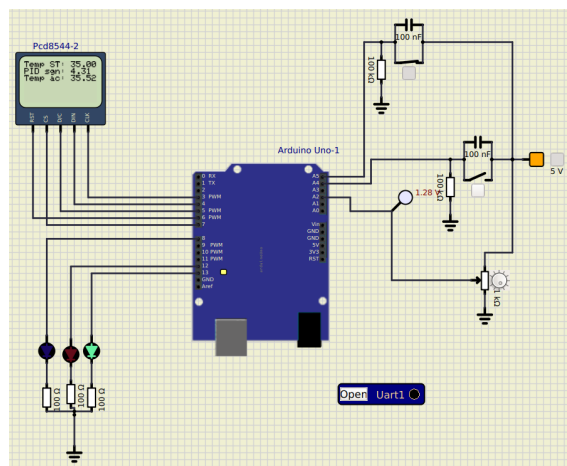


Figura 8: Led verde

En la siguiente imagen se logra comprobar el funcionamiento de la incubadora. La cual al estar a una temperatura de 24°C se enciende el LED rojo:

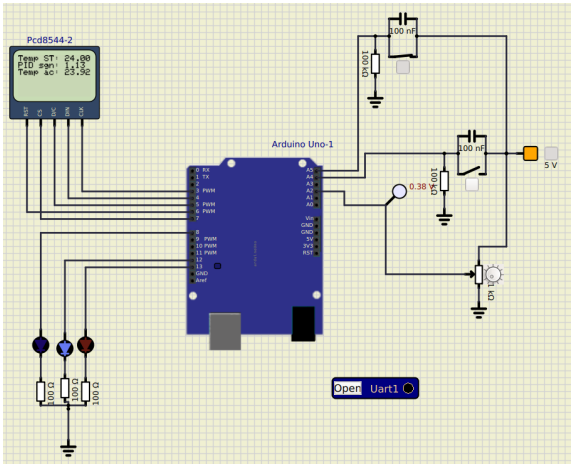


Figura 9: Led azul

Al graficar los datos obtenidos en el documento .csv, se obtiene la siguiente figura:

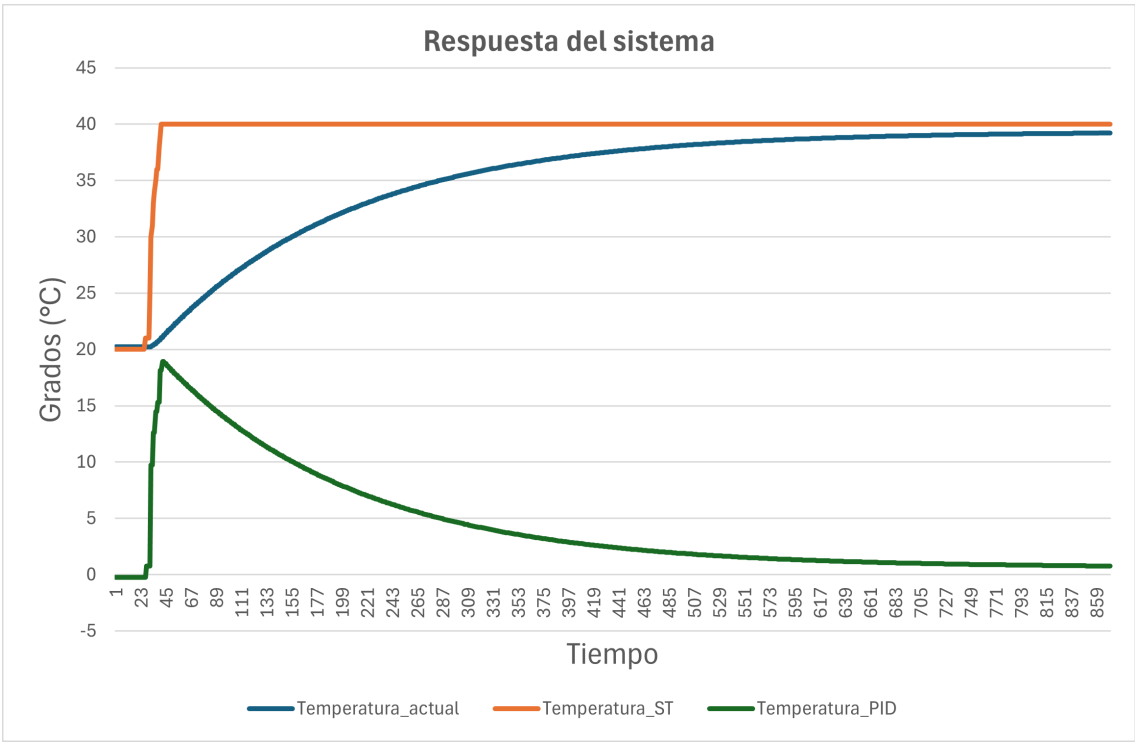


Figura 10: Gráfica del sistema

4. Conclusiones y recomendaciones

- Se pudo estudiar y aplicar los conceptos relacionados al Arduino Uno, esto con el fin de poder desarrollar la aplicación de una incubadora. Esto mediante una implementación de un algoritmo de control PID, para controlar la temperatura deseada.
- Se recomienda repasar el funcionamiento de controladores PID. Esto debido a que si no se escogen valores correctos de las constantes a emplear pueden generar oscilaciones en nuestro sistema.
- Al exportar los datos de csv a excel o algún otro programa de hoja de calculo hay que tener cuidado de como configurar la visualización de estos.
- La comunicación por medio del puerto serial puede resultar de muy buena ayuda en diferentes proyectos o usos en los cuales se necesite acceder a un registro de dichos datos medidos.

Referencias

- [1] M. V. Carrillo *et al.*, “Introducción de Arduino,” *Vida Científica Boletín Científico de la Escuela Preparatoria No. 4*, vol. 9, no. 17, pp. 4–8, 2021.
- [2] A. Álvarez Carulla, “MASB (Arduino): Comunicación serie I,” 2020.
- [3] H. Siebeneicher, “Universal asynchronous receiver-transmitter (uart),” 2023. [Online]. Available: <https://docs.arduino.cc/learn/communication/uart/>
- [4] “Microchip atmega328p-pn,” 2013. [Online]. Available: <https://mexico.newark.com/microchip/atmega328p-pn/microcontroller-mcu-8-bit-atmega/dp/68T2943>
- [5] V. Mazzone, “Controladores PID,” *Quilmes: Universidad Nacional de Quilmes*, 2002.
- [6] J. Christoffersen, “Switch bounce and how to deal with it,” 2015. [Online]. Available: <https://www.allaboutcircuits.com/technical-articles/switch-bounce-how-to-deal-with-it/>

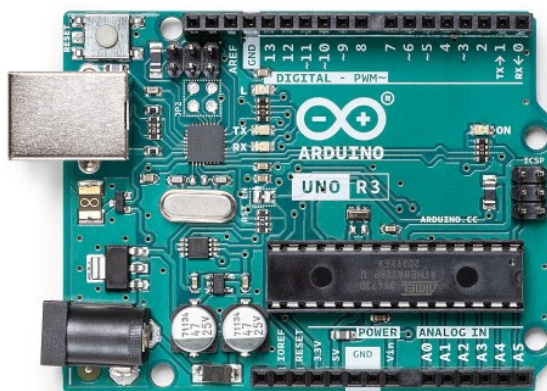
5. Anexos

5.1. Hoja de datos

En próximas páginas se adjuntan las hojas del fabricante cuya información fue necesaria para realizar el procedimiento

Product Reference Manual

SKU: A000066



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

1 The Board	4
1.1 Application Examples	4
1.2 Related Products	4
2 Ratings	5
2.1 Recommended Operating Conditions	5
2.2 Power Consumption	5
3 Functional Overview	5
3.1 Board Topology	5
3.2 Processor	6
3.3 Power Tree	7
4 Board Operation	8
4.1 Getting Started - IDE	8
4.2 Getting Started - Arduino Web Editor	8
4.3 Sample Sketches	8
4.4 Online Resources	8
5 Connector Pinouts	9
5.1 JANALOG	10
5.2 JDIGITAL	10
5.3 Mechanical Information	11
5.4 Board Outline & Mounting Holes	11
6 Certifications	12
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
7 FCC Caution	13
8 Company Information	14
9 Reference Documentation	14
10 Revision History	14



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.

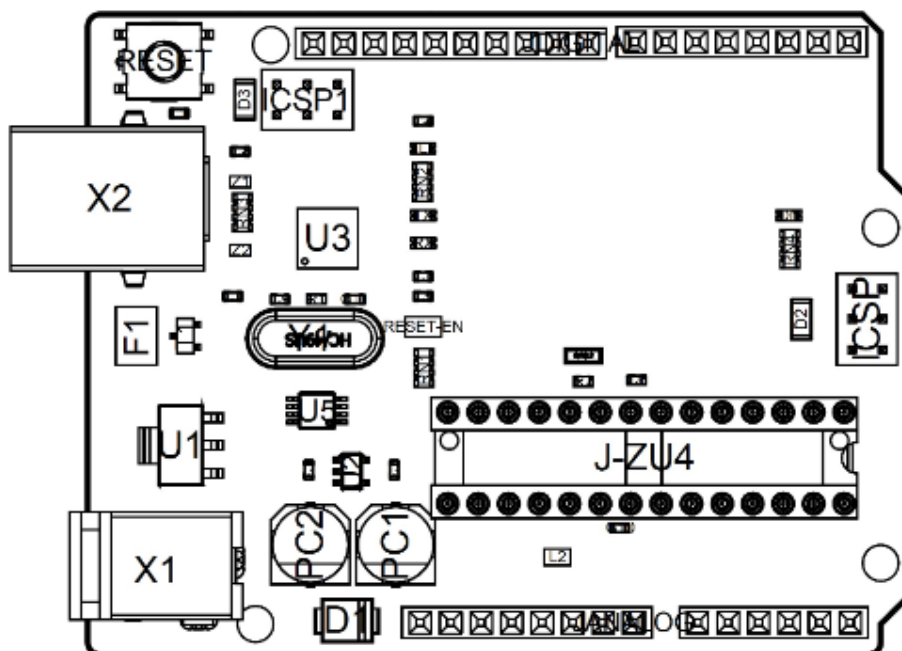
2.2 Power Consumption

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

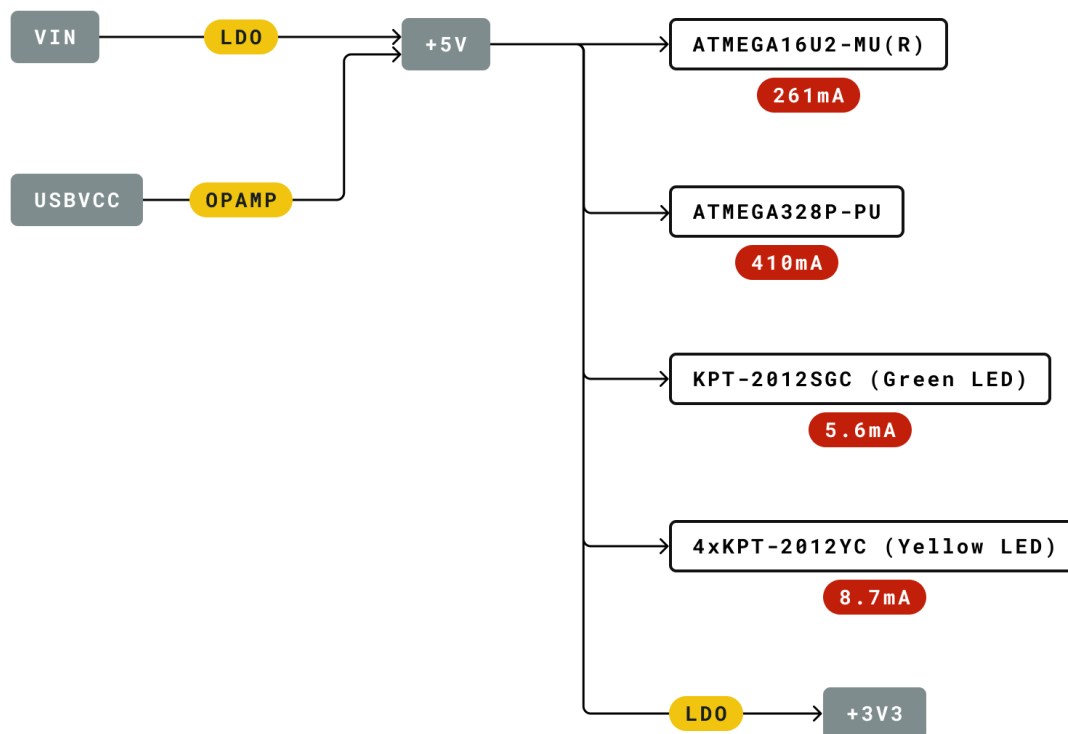


Ref.	Description	Ref.	Description
X1	Power jack 2.1x5.5mm	U1	SPX1117M3-L-5 Regulator
X2	USB B Connector	U3	ATMEGA16U2 Module
PC1	EEE-1EA470WP 25V SMD Capacitor	U5	LMV358LIST-A.9 IC
PC2	EEE-1EA470WP 25V SMD Capacitor	F1	Chip Capacitor, High Density
D1	CGRA4007-G Rectifier	ICSP	Pin header connector (through 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.

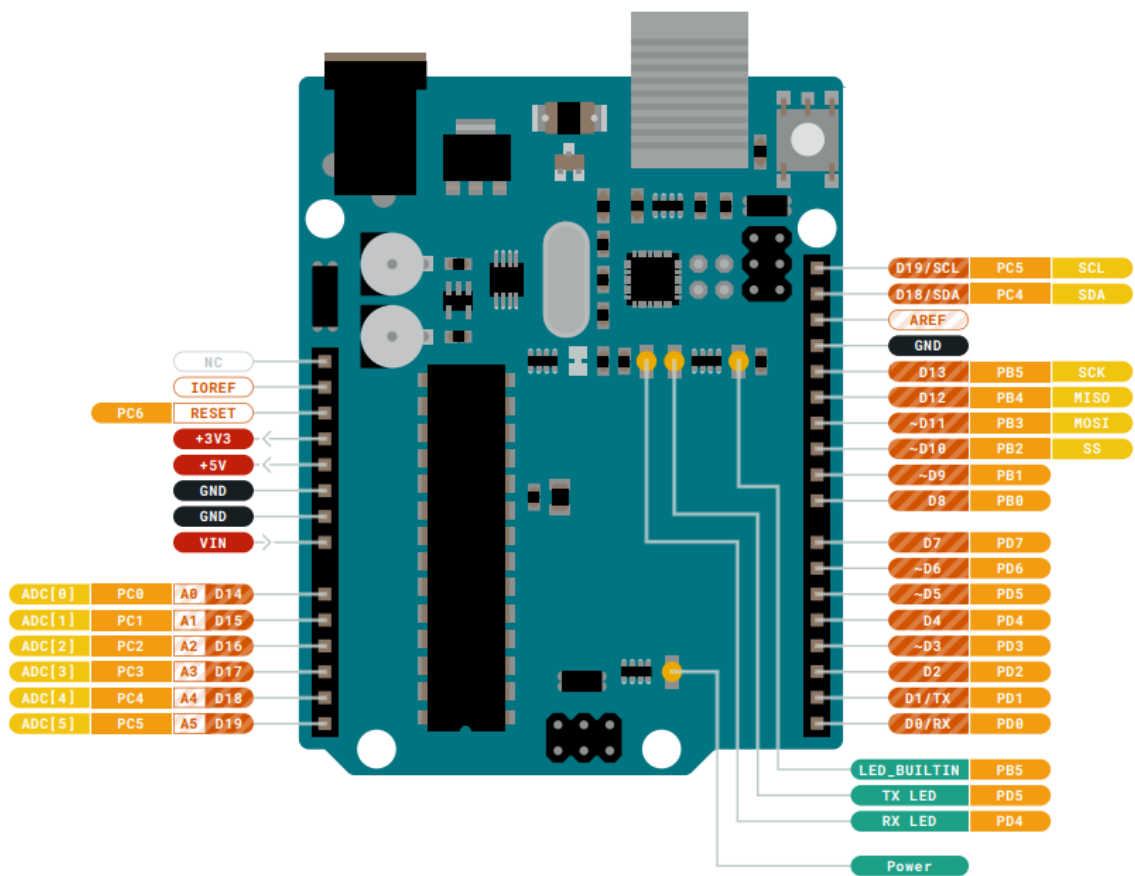
4.3 Sample Sketches

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

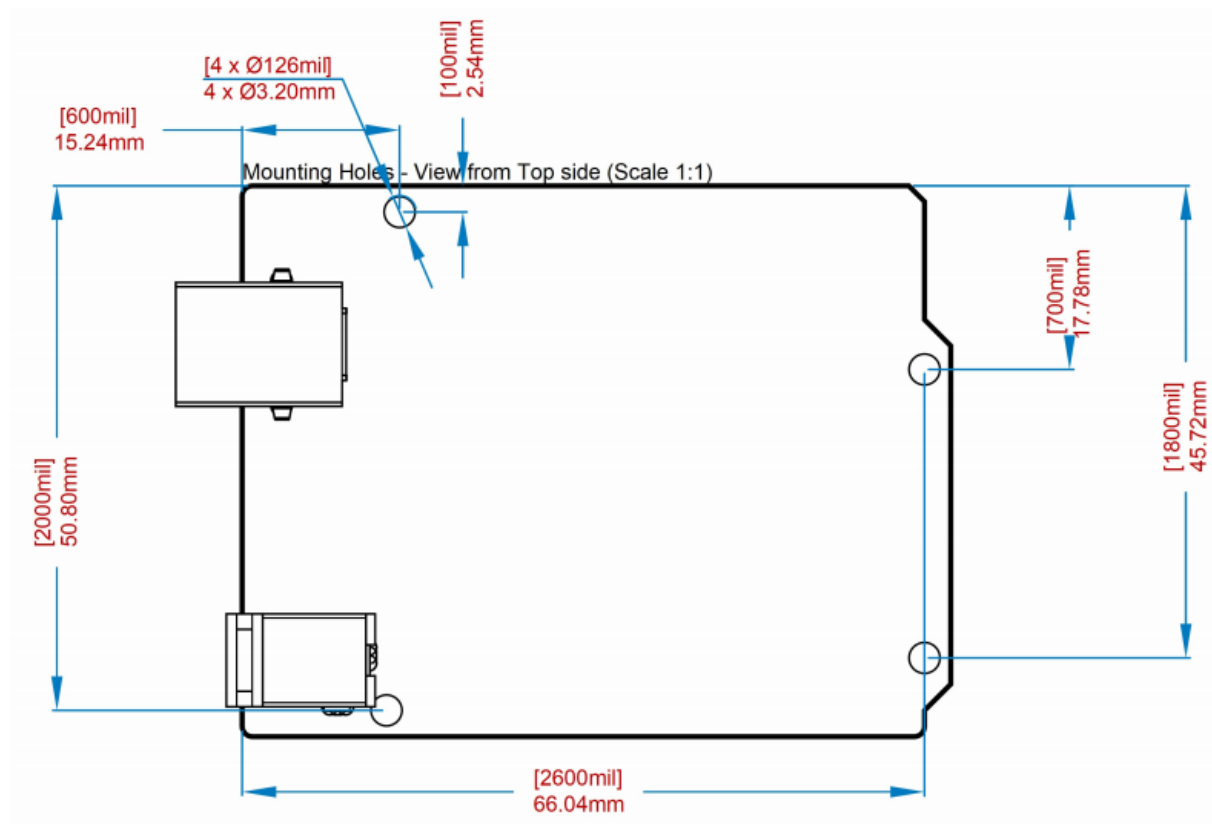
Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	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



Board outline

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).

ROHS 2 Directive 2011/65/EU	
Conforms to:	EN50581:2012
Directive 2014/35/EU. (LVD)	
Conforms to:	EN 60950-1:2006/A11:2009/A1:2010/A12:2011/AC:2011
Directive 2004/40/EC & 2008/46/EC & 2013/35/EU, EMF	
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)	https://www.arduino.cc/en/Main/Software
Arduino IDE (Cloud)	https://create.arduino.cc/editor
Cloud IDE Getting Started	https://create.arduino.cc/projecthub/Arduino_Genuino/getting-started-with-arduino-web-editor-4b3e4a
Arduino Website	https://www.arduino.cc/
Project Hub	https://create.arduino.cc/projecthub?by=part&part_id=11332&sort=trending
Library Reference	https://www.arduino.cc/reference/en/
Online Store	https://store.arduino.cc/

10 Revision History

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