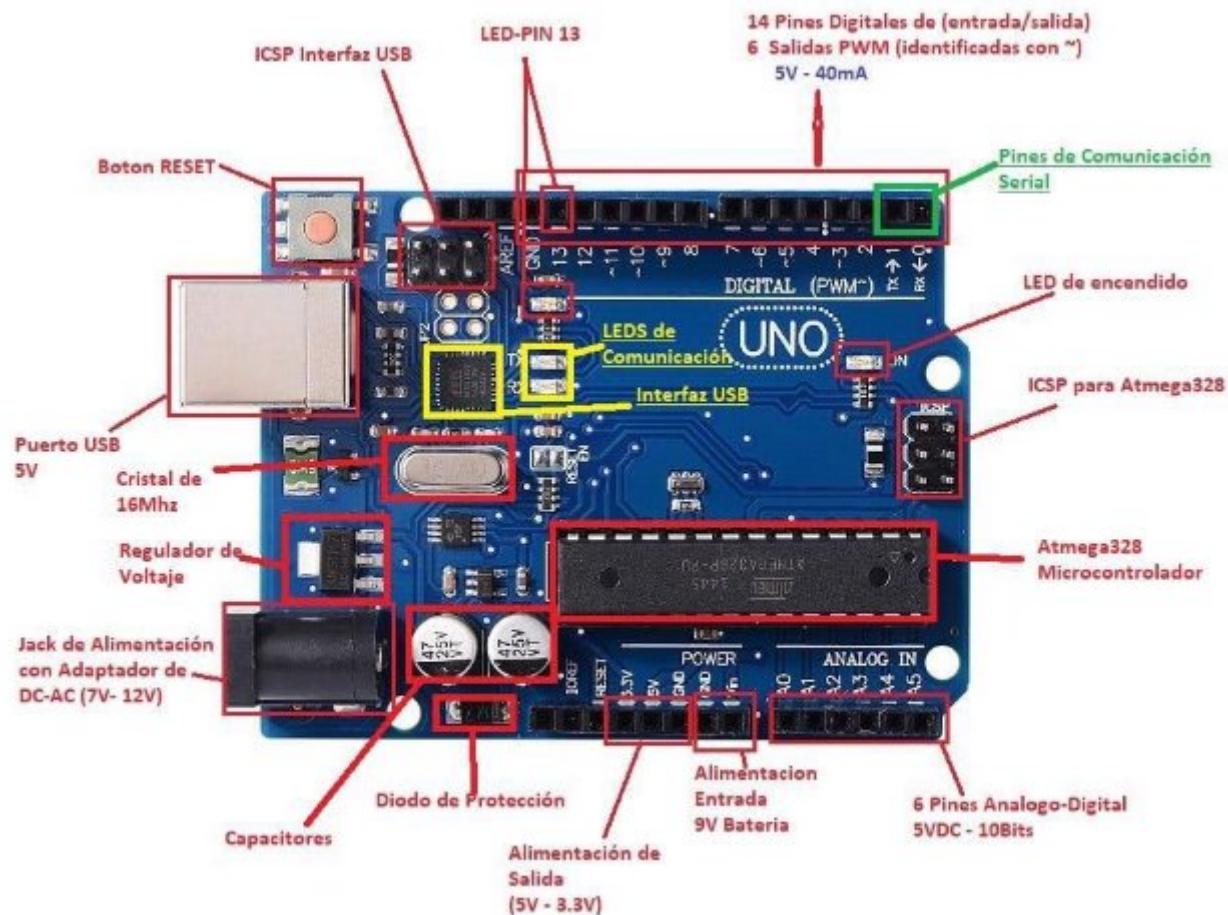



## 4. Placa de control

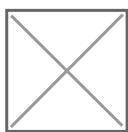
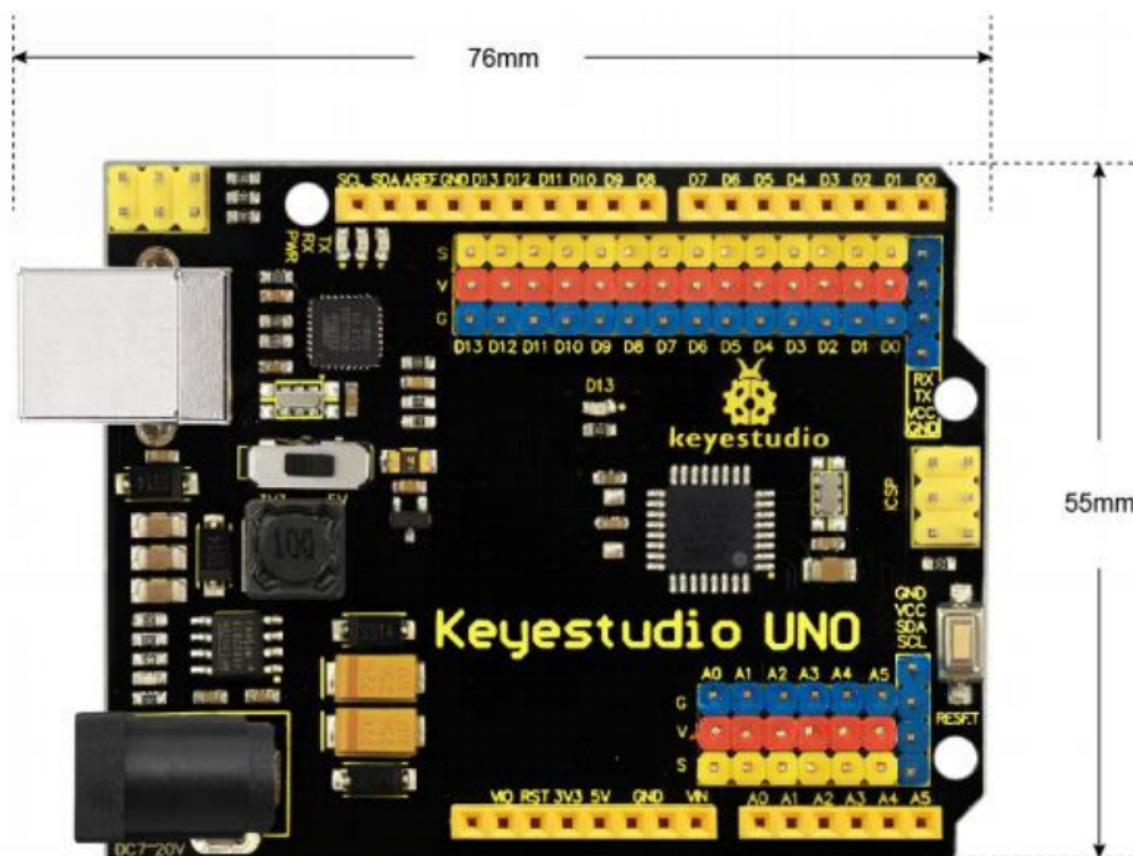
# Keyestudio UNO (compatible con Arduino UNO)

La placa de control que gestiona la placa Imagina TDR STEAM está basada en una placa **Arduino UNO**. Arduino es una plataforma de prototipos electrónicos de código abierto (Open-Source) basada en hardware y software libre, flexible y fácil de usar, con una comunidad muy grande que genera muchísima información. Esta plataforma permite crear diferentes tipos de sistema para diferentes usos. Puede ser empleada por *artistas, diseñadores, ingenieros, profesores, alumnos, etc.*, y en general, cualquier persona que esté interesada en crear objetos o entornos interactivos, prototipos, sistemas robóticos, etc.

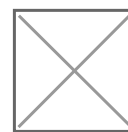




Al ser hardware libre existen multitud de fabricantes que han desarrollado  versiones basadas en Arduino. Uno de esos fabricantes es **Keyestudio**.



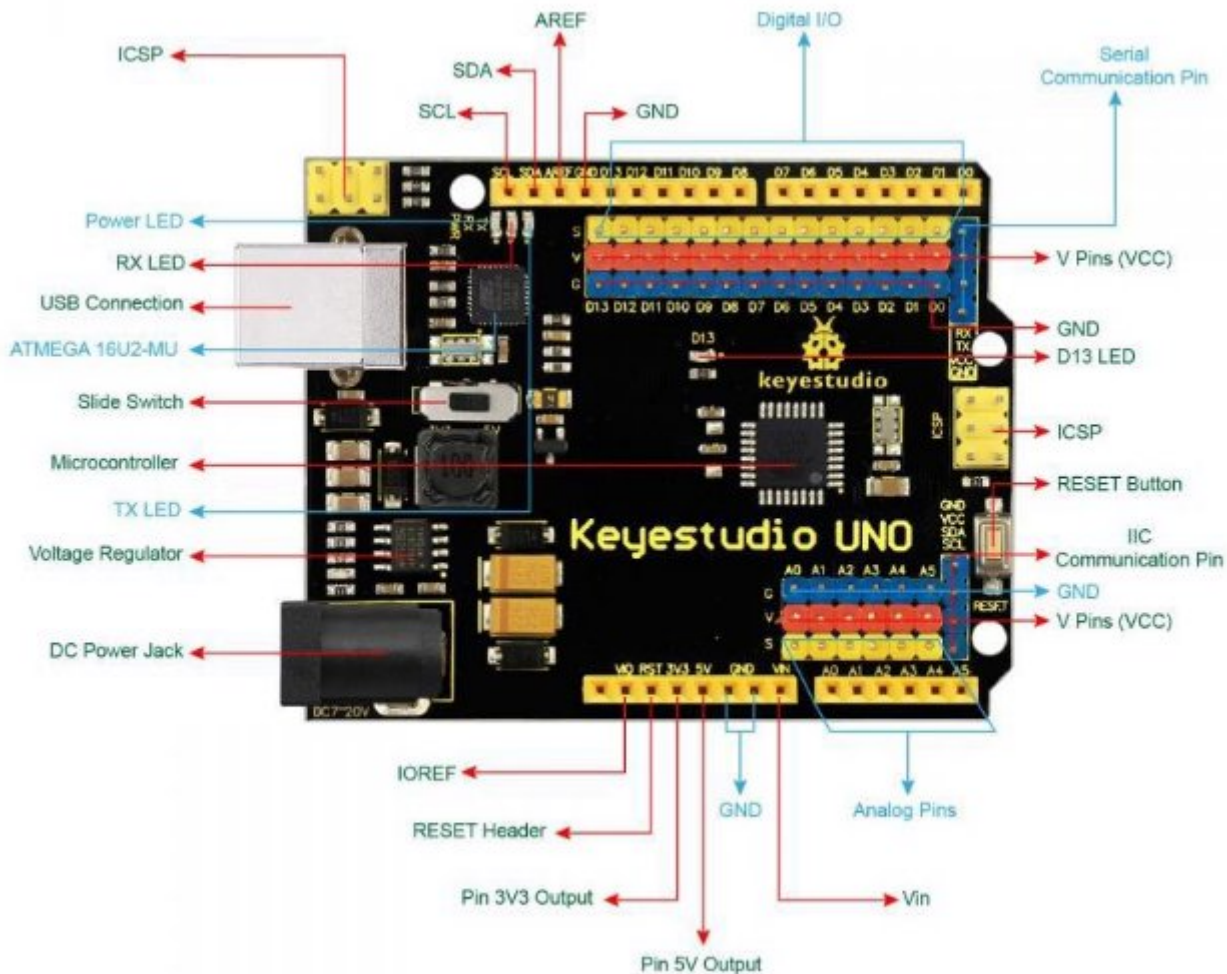
La placa **Keyestudio UNO** lleva un microcontrolador que está basado en el



ATmega328. Tiene 14 pines digitales de entrada / salida (de los cuales 6 se pueden usar como salidas PWM), 6 entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, un

conector de alimentación, un conector ICSP y un botón de reinicio (reset).





### Especificaciones:

- Voltaje de funcionamiento: + 5V.
- Voltaje de entrada externo: + 7V ~ + 12V. (Límite: +6 V. <+ 20 V).
- Corriente de interfaz DCI / O: 20mA.
- Flash Memory: 32 KB (ATmega328) de los cuales 0,5 KB utilizados por el gestor de arranque.
- Capacidad de almacenamiento EEPROM: 1KB.

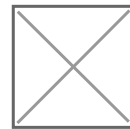


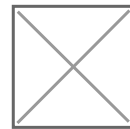
- Frecuencia del reloj: 16MHZ.
- Microcontrolador ATmega328P-PU
- Pines de E / S digital 14 (de los cuales 6 proporcionan salida PWM).
- Pines de E / S digitales PWM 6 (D3, D5, D6, D9, D10, D11).
- Pines de entrada analógica 6 (A0-A5).
- LED\_BUILTIN D13.



Las conexiones de la placa TDR STEAM con la placa Keyestudio UNO son las siguientes:

| Conexión | Descripción                             | Tipo              |
|----------|---|-------------------|
| D0       | Rx                                      | Comunicación      |
| D1       | Tx                                      | PC/Bluetooth/Wifi |
| D2       | Pulsador SW1                            | Entrada digital   |
| D3       | Libre                                   | E/S digital       |
| D4       | DHT11 (sensor de humedad y temperatura) | Entrada digital   |
| D5       | Libre                                   | E/S digital       |
| D6       | RGB (rojo)                              | Salida digital    |
| D7       | Pulsador SW2                            | Entrada digital   |
| D8       | Zumbador                                | Salida digital    |
| D9       | RGB (verde)                             | Salida digital    |
| D10      | RGB (azul)                              | Salida digital    |
| D11      | Sensor infrarrojos                      | Entrada digital   |
| D12      | Led rojo                                | Salida digital    |
| D13      | Led azul                                | Salida digital    |
| A0       | Potenciómetro                           | Entrada analógica |
| A1       | LDR (sensor de luz)                     | Entrada analógica |
| A2       | LM35 (sensor de temperatura)            | Entrada analógica |
| A3       | Libre                                   | Entrada analógica |
| A4       | SDA                                     | I2C               |
| A5       | SCL                                     |                   |



Para realizar la programación la podemos hacer mediante la IDE de Arduino  o mediante ArduinoBlocks. Como podemos ver son dos sistemas diferentes. En la IDE de Arduino la programación se realiza mediante instrucciones (derecha de la imagen) y en ArduinoBlocks se realiza mediante bloques (izquierda de la imagen). En la siguiente imagen se hace una comparación de código entre ArduinoBlocks y Arduino IDE.

Copyright 2025

```

1 void setup()
2 {
3   pinMode(13, OUTPUT);
4 }
5
6
7 void loop()
8 {
9   digitalWrite(13,HIGH);
10  delay(500);
11  digitalWrite(13,LOW);
12  delay(500);
13 }
  
```



Utilizar ArduinoBlocks simplifica y hace más inteligible el código, lo que permite iniciarse en el mundo de la programación de un modo más amigable. ArduinoBlocks también permite programar de diversas formas, ya que tiene bloques específicos que realizan las mismas funciones pero que se pueden entender de forma más sencilla.

---

Revision #2

Created 30 May 2022 10:09:56 by Equipo CATEDU

Updated 1 June 2022 10:20:02 by Equipo CATEDU