

Placa Arduino Uno



Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328. Cuenta con 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas. Además, incluye un resonador cerámico de 16 MHz, un conector USB, un conector de alimentación, una cabecera ICSP y un botón de reseteado. La placa incluye todo lo necesario para que el microcontrolador haga su trabajo, basta conectarla a un ordenador con un cable USB o a la corriente eléctrica a través de un transformador.

Con esta placa, Arduino conmemoraba el lanzamiento de Arduino 1.0., convirtiéndose en la versión de referencia de Arduino.

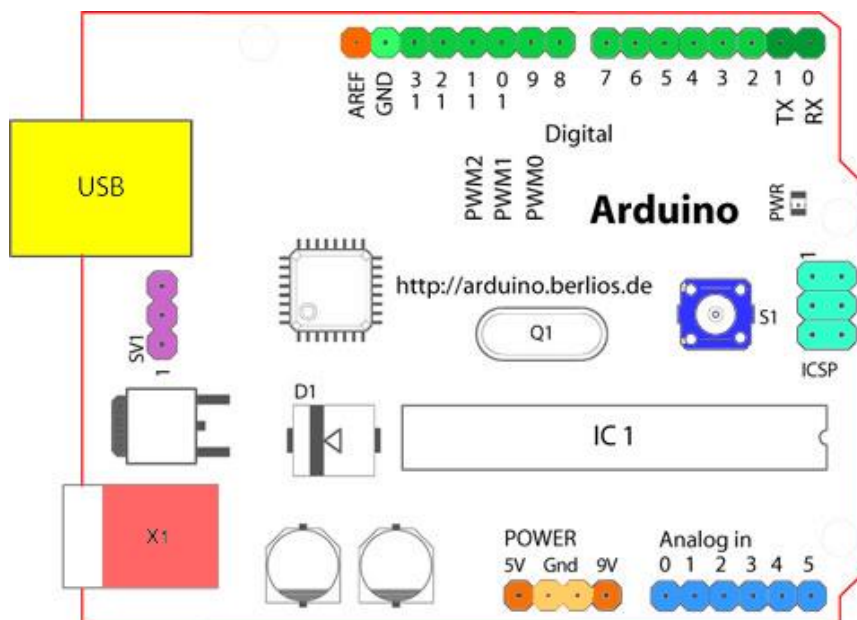
Características técnicas de Arduino Uno r3

- | | |
|---|-----------|
| ▶ Microcontrolador: | ATmega328 |
| ▶ Voltage: | 5V |
| ▶ Voltage entrada (recomendado): | 7-12V |
| ▶ Voltage entrada (limites): | 6-20V |
| ▶ Digital I/O Pins: 14 (de los cuales 6 son salida PWM) | |

- ▶ Entradas Analógicas: 6
- ▶ DC Current per I/O Pin: 40 mA
- ▶ DC Current parar 3.3V Pin: 50 mA
- ▶ Flash Memory: 32 KB (ATmega328) de los cuales 0.5 KB son utilizados para el arranque
- ▶ SRAM: 2 KB (ATmega328)
- ▶ EEPROM: 1 KB (ATmega328)
- ▶ Clock Speed: 16 MHz

Introducción general a una placa Arduino

Mirando a la placa desde la parte de arriba, este es el esquema de lo que puedes ver (los componentes de la placa con los que puedes interactuar en su uso normal están resaltados):



Empezando según las agujas del reloj:

- ▶ Terminal de referencia analógica (naranja)
- ▶ Tierra digital (verde claro)
- ▶ Terminales digitales 2-13 (verde oscuro)
- ▶ Terminales digitales 0-1/ E/S serie - TX/RX (verde oscuro) - Estos pines no se pueden utilizar como e/s digitales (digitalRead() y digitalWrite()) si estás utilizando comunicación serie (por ejemplo Serial.begin).
- ▶ Botón de reinicio - S1 (azul oscuro)
- ▶ Programador serie en circuito "In-circuit Serial Programmer" o "ICSP" (azul celeste).
- ▶ Terminales de entrada analógica 0-5 (azul claro)
- ▶ Terminales de alimentación y tierra (alimentación: naranja, tierras: naranja claro)
- ▶ Entrada de alimentación externa (9-12VDC) - X1 (rosa)
- ▶ Selector de alimentación externa o por USB (coloca un jumper en los dos pines mas

cercanos de la alimentación que quieras) - SV1 (púrpura). En las versiones nuevas de Arduino la selección de alimentación es automática por lo que puede que no tengas este selector.

► USB (utilizado para subir programas a la placa y para comunicaciones serie entre la placa y el ordenador; puede utilizarse como alimentación de la placa) (amarillo)

Entradas y salidas digitales/analógicas

Un sistema electrónico es cualquier disposición de componentes electrónicos con un conjunto definido de entradas y salidas. Una placa Arduino, por tanto, puede pensarse de forma simplificada como un sistema que acepta información en forma de señal de entrada, desarrolla ciertas operaciones sobre ésta y luego produce señales de salida.

Justamente, una de las opciones que hacen más potente a Arduino son sus entradas/salidas digitales. ¿Entonces por qué hablamos de analógicas?

En los sistemas electrónicos, una magnitud física variable se representa generalmente mediante una señal eléctrica que varía de manera tal que describe esa magnitud. Por lo general, se hace referencia a las señales continuas como señales analógicas, mientras que asociamos las señales discretas a señales digitales: el ejemplo más claro es el de las señales binarias, donde la señal sólo pueden tomar dos niveles, 0 o 1.

Arduino incorpora terminales digitales (señales discretas) pero de tal forma que tenemos un gran abanico de valores con los que trabajar (por ejemplo, 255 valores de luz en un fotosensor, siendo 0 ausencia de luz y 255 el máximo valor lumínico).

Terminales Digitales

Las terminales digitales de una placa Arduino pueden ser utilizadas para entradas o salidas de propósito general a través de los comandos de programación `pinMode()`, `digitalRead()`, y `digitalWrite()`. Cada terminal tiene una resistencia pull-up que puede activarse o desactivarse utilizando `DigitalWrite()` (con un valor de HIGH o LOW, respectivamente) cuando el pin esta configurado como entrada. La corriente máxima por salida es 40 mA.

► Serial: 0 (RX) y 1 (TX). Utilizado para recibir (RX) y transmitir (TX) datos serie TTL. En el Arduino Diacemila, estas terminales están conectadas a las correspondientes patas del circuito integrado conversor FTDI USB a TTL serie. En el Arduino BT, están conectados al las terminales correspondientes del modulo Bluetooth WT11. En el Arduino Mini y el Arduino LilyPad, están destinados para el uso de un módulo serie TTL externo (por ejemplo el adaptador Mini-USB).

- ▶ Interruptores externos: 2 y 3. Estas terminales pueden ser configuradas para disparar una interrupción con un valor bajo, un pulso de subida o bajada, o un cambio de valor. Mira la función `attachInterrupt()` para mas detalles.
- ▶ PWM: 3, 5, 6, 9, 10, y 11. Proporcionan salidas PWM de 8 bit con la función `analogWrite()`. En placas con ATmega8, las salidas PWM solo están disponibles en los pines 9, 10, y 11.
- ▶ Reset BT: 7. (solo en Arduino BT) Conectado a la línea de reset del módulo bluetooth.
- ▶ SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estas terminales soportan comunicación SPI. Aunque esta funcionalidad esta proporcionada por el hardware, no está incluida actualmente en el lenguaje Arduino.
- ▶ LED: 13. En el Diacemila y el LilyPad hay un led en placa conectado al pin digital 13. cuando el pin tiene valor HIGH, el LED está encendido, cuando el pin está en LOW, está apagado

Pines Analógicos

- ▶ Los pines de entrada analógicos soportan conversiones analógico-digital (ADC) de 10 bit utilizando la función `analogRead()`. Las entradas analógicas pueden ser también usadas como pines digitales: entrada analógica 0 como pin digital 14 hasta la entrada analógica 5 como pin digital 19. Las entradas analógicas 6 y 7 (presentes en el Mini y el BT) no pueden ser utilizadas como pines digitales.
- ▶ I2C: 4 (SDA) y 5 (SCL). Soportan comunicaciones I2C (TWI) utilizando la librería `Wire` (documentación en la página web de Wiring).

Pines de alimentación

- ▶ VIN (a veces marcada como "9V"). Es el voltaje de entrada a la placa Arduino cuando se está utilizando una fuente de alimentación externa (En comparación con los 5 voltios de la conexión USB o de otra fuente de alimentación regulada). Puedes proporcionar voltaje a través de este pin. Date cuenta que diferentes placas aceptan distintos rangos de voltaje de entrada, por favor, mira la documentación de tu placa. También date cuenta que el LilyPad no tiene pin VIN y acepta solo una entrada regulada.
- ▶ 5V. La alimentación regulada utilizada para alimentar el microcontrolador y otros componentes de la placa. Esta puede venir de VIN a través de un regulador en placa o ser proporcionada por USB u otra fuente regulada de 5V.

- ▶ 3V3. (solo en el Diacemila) Una fuente de 3.3 voltios generada por el chip FTDI de la placa.
- ▶ GND. Pines de tierra.

Otros Pines

- ▶ AREF. Referencia de voltaje para las entradas analógicas. Utilizada con la función `analogReference()`.
- ▶ Reset. (Solo en el Diacemila) Pon esta línea a LOW para resetear el microcontrolador. Utilizada típicamente para añadir un botón de reset a shields que bloquean el de la placa principal.

Más información en www.arduino.cc