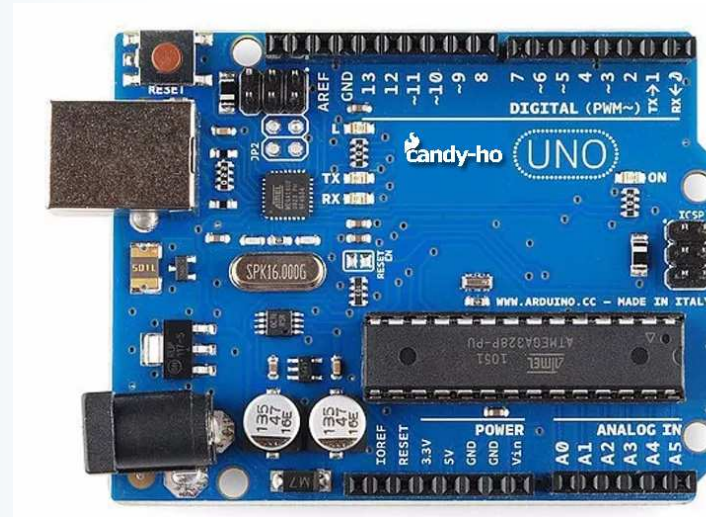


Descarga el manual

COMPLETO

Ingresando a:

www.candy-ho.com/manuales



Función pinMode(pin, mode)

Función usada en la función setup() para configurar un pin dado para comportarse como INPUT o OUTPUT. Ej. pinMode(pin, OUTPUT); configura el pin número 'pin' como de salida. Los pines de Arduino funcionan por defecto como entradas, de forma que no necesitan declararse explícitamente como entradas empleando pinMode().

Función digitalRead(pin)

Lee el valor desde un pin digital específico. Devuelve un valor HIGH o LOW. El pin puede ser especificado con una variable o una constante (0-13). Ej. v = digitalRead(Pin);

Función digitalWrite(pin, value)

Introduce un nivel alto (HIGH) o bajo (LOW) en el pin digital especificado. De nuevo, el pin puede ser especificado con una variable o una constante 0-13. Ej. digitalWrite(pin, HIGH);

Función analogRead(pin)

Lee el valor desde el pin analógico especificado con una resolución de 10 bits. Esta función solo funciona en los pines analógicos (0-5). El valor resultante es un entero de 0 a 1023. Los pines analógicos, a diferencia de los digitales no necesitan declararse previamente como INPUT o OUTPUT.

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 módulo 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).

Interrupciones externas: 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.

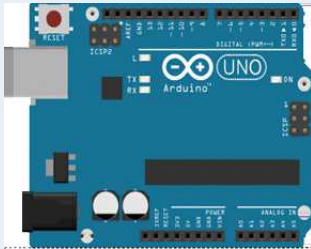
PWM: 3, 5, 6, 9, 10, y 11. Proporcionan salidas PWM de 8 bit con la función analogWrite(). En placas con ATmega8, las salidas PWM solo están disponibles en los pines 9, 10, y 11.

Reset BT: 7. (solo en Arduino BT) Conectado a la línea de reset del módulo bluetooth.

CANDY-HO ARDUINO

UNO R3

MANUAL DE INSTRUCCIONES



Características

Este es el nuevo Arduino Uno R3 utiliza el microcontrolador ATmega328. En adición a todas las características de las tarjetas anteriores, el Arduino Uno utiliza el ATmega16U2 para el manejo de USB en lugar del 8U2 (o del FTDI encontrado en generaciones previas). Esto permite ratios de transferencia más rápidos y más memoria. No se necesitan drivers para Linux o Mac (el archivo inf para Windows es necesario y está incluido en el IDE de Arduino).

La tarjeta Arduino Uno R3 incluso añade pins SDA y SCL cercanos al AREF. Es más, hay dos nuevos pines cerca del pin RESET. Uno es el IOREF, que permite a los shields adaptarse al voltaje brindado por la tarjeta. El otro pin no se encuentra conectado y está reservado para propósitos futuros. La tarjeta trabaja con todos los shields existentes y podrá adaptarse con los nuevos shields utilizando esos pines adicionales.

El Arduino es una plataforma computacional física open-source basada en una simple tarjeta de I/O y un entorno de desarrollo que implementa el lenguaje Processing/Wiring. El Arduino Uno R3 puede ser utilizado para desarrollar objetos interactivos o puede ser conectado a software de tu computadora (por ejemplo, Flash, Processing, MaxMSP). El IDE open-source puede ser descargado gratuitamente (actualmente para Mac OS X, Windows y Linux).

Nota: Esta plataforma requiere la carpeta de drivers Arduino 1.0 para poder instalarlo de forma apropiada en algunos computadores. Hemos testeado y confirmado que el Arduino Uno R3 puede ser programado en versiones anteriores del IDE. Sin embargo, la primera vez que uses el Arduino en una nueva computadora deberás tener el Arduino 1.0 instalado en la máquina. Si estás interesado en leer más acerca de los cambios en el IDE, revisa las notas oficiales de Arduino 1.0.

Las características principales son:

Microcontrolador: ATmega328
Voltage: 5V
Voltage entrada (recomendado): 7-12V
Voltage entrada (límites): 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 para 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

Datos técnicos

Alimentación

Las placas pueden ser alimentadas por medio de la conexión USB o con una fuente de alimentación externa de entre 6 a 20 V, aunque el rango recomendado es de 7 a 12 V. Las fuentes de alimentación externas (no-USB) pueden ser tanto un transformador como una batería. El transformador se puede conectar usando un conector macho de 2.1 mm con centro positivo en el conector hembra de la placa. Los cables de la batería pueden conectarse a los pines Gnd y Vin en los conectores de alimentación (POWER)

Entradas y Salidas

Cada uno de los 14 pines digitales (de 0 a 13) pueden utilizarse como entradas o como salidas usando las funciones pinMode(), digitalWrite() y digitalRead(). Las E/S operan a 5 V. Cada pin puede proporcionar o recibir una intensidad máxima de 40 mA. Los pines 3, 5, 6, 9, 10, y 11 proporcionan una salida PWM (modulación por anchura de pulsos) de 8 bits de resolución (valores de 0 a 255) mediante la función analogWrite(). El pin digital 13 lleva conectado un LED integrado en la propia placa. Se encenderá cuando dicho pin se configura como salida y adopte un valor HIGH.; con valor LOW se apaga. La placa tiene 6 entradas analógicas, y cada una de ellas proporciona una resolución de 10 bits (1024 valores).

Programación

La placa Arduino se puede programar a través del IDE (Entorno de Desarrollo Integrado) de Arduino. Primero hay que decirle al IDE nuestro tipo de placa. Para ello, selecciona bien "Arduino Duemilanove w/ ATmega328" o bien Arduino Uno del menu Herramientas > Tarjeta. También hay que decirle al IDE en qué puerto USB hemos conectado la placa. Lo haremos a través del menú Herramientas > Puerto Serial. Aparecen una serie de puertos denominados COMx, donde x es un número. Para saber cuál es, desconecta y conecta el cable USB del ordenador, y el que desaparezca y vuelva a aparecer es nuestro puerto.