

CARACTERÍSTICAS DE LAS PLACAS ARDUINO

Céspedes Machicao Marcelo

Universidad Autónoma Juan Misael Saracho
Tarija, Bolivia

Correo electrónico: marcelocespedes@yahoo.com

RESUMEN

Arduino es una placa de hardware open-source, que ha tenido una gran difusión desde su lanzamiento en 2005, su éxito y masificación de uso se debe, en parte, a su bajo costo, facilidad de uso, amplia documentación y gran asequibilidad; en este artículo se describen las características técnicas, desde un punto de vista más accesible al usuario común, sin ingresar en los detalles de las especificaciones técnicas formales.

PALABRAS CLAVE

Arduino, hardware, microcontroladores, código abierto.

INTRODUCCIÓN

Arduino es un proyecto de hardware y software que surgió en Italia en 2005 [Hughes, 2016,1] y ha tenido un gran impacto en la comunidad tecnológica debido a que es un proyecto de código abierto, de bajo costo y de múltiples posibilidades de aplicación.

La comunidad que usa esta tecnología y apoya a este proyecto ha tenido un crecimiento exponencial y comprende a personas prácticamente de todas las edades.

Gran parte del éxito de esta tecnología se debe a las extensiones de hardware que se fueron añadiendo a la placa Arduino, como variedad de sensores, actuadores, placas o tarjetas extras (shields), dispositivos de comunicación, conectividad a la internet mediante protocolos estándar y el desarrollo de diferentes placas Arduino con características reducidas y avanzadas, que satisfacen la mayoría de las necesidades y requerimientos

para la diversidad de proyectos donde se aplican.

Otro aspecto fundamental de Arduino, es la existencia de extensa documentación técnica disponible en la Internet, y cientos de manuales, revistas y libros disponibles en formato físico y electrónico.

Sin embargo, se ha observado, que la gran documentación existente trata de la descripción de proyectos concretos y particularmente los aspectos relativos a la programación, mientras que la documentación de especificaciones técnicas está más dirigida a personal especializado y profesionales del área; por ello, en este artículo, se realiza una síntesis de las características técnicas, sin entrar en los formalismos de ingeniería.

EL HARDWARE ARDUINO

Una placa Arduino está construida en base a un microcontrolador, denominado AVR, del fabricante americano ATMEL, que es una versión muy reducida de un microprocesador programable, y que contiene todos los elementos esenciales de una computadora, exceptuando los dispositivos periféricos, como disco duro, tarjeta de video, teclado, etc.

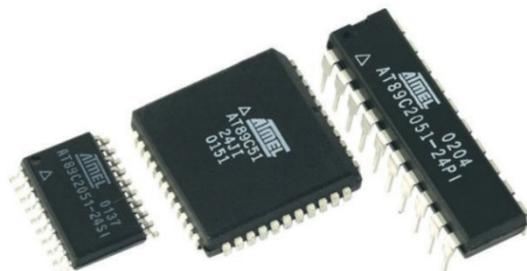


Figura 1: Microcontroladores ATMEL



El microcontrolador es el elemento central de la placa Arduino, que le proporciona su carácter altamente versátil, de forma similar al que le otorga un microprocesador a una computadora.

Entre las partes más importantes del microcontrolador se tiene:

- Un procesador programable que contiene una unidad lógica aritmética (ALU) y los registros necesarios para la ejecución de las operaciones, que soporta un conjunto de instrucciones reducido, optimizado y de alto rendimiento.
- Memoria flash (no volátil), para almacenar los programas del usuario.
- Memoria RAM para los datos del usuario.
- Memoria ROM para datos persistentes.
- Puertos de entradas/salidas digitales.
- Puertos de entrada analógicos.
- Salida analógica PWM.
- Temporizadores internos.
- Comunicación serial, I2C y SPI,
- Estado de bajo consumo.

El microcontrolador ejecuta las operaciones en sincronismo con una señal binaria de clock o reloj, a la velocidad de 8 a 32 MHz, dependiendo del modelo, que le provee un cristal de cuarzo.

La placa Arduino contiene diversos componentes para convertirla en una tarjeta autónoma y completamente funcional, como:

1. Puerto USB.
2. Terminales digitales de entrada/salida
3. Terminales para entrada de señales analógicas.
4. Botón RESET de reiniciación.
5. Conector de alimentación con regulador de voltaje.
6. Terminales de alimentación de energía para dispositivos externos.
7. Leds indicadores de transmisión de datos

8. Led indicador de encendido
9. Microcontrolador

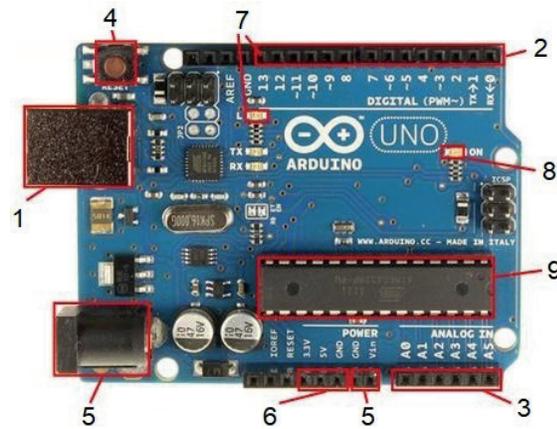


Figura 2: Tarjeta Arduino UNO

PUERTO USB

El puerto USB, cuenta con un conector estándar de tipo B o micro B, según el modelo de placa, para la transmisión serial de datos de forma bidireccional.

Permite la conexión de la placa a una computadora para la transferencia del programa compilado, lo que facilita su programación.

Cuenta con las terminales de energía de 5V, lo que posibilita la alimentación independiente de la placa por el puerto USB, por lo que no es necesario otra fuente de alimentación.

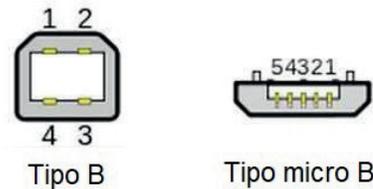


Figura 3: Conectores USB

TERMINALES DIGITALES

Arduino dispone de un conjunto de terminales digitales de entrada/salida para la conexión de cualquier dispositivo externo compatible como: sensores, motores, pantallas, teclados, tarjetas de expansión como microSD, reloj de tiempo real, tarjetas de red, Bluetooth, GSM, etc. El número de dispositivos externos digitales, compatibles con



Arduino es cada vez mayor

Las terminales se numeran a partir de 0 y cada una es configurable mediante software, de manera individual, como de entrada o de salida de datos, constituyendo el medio más empleado para la conexión de dispositivos externos.

Las terminales numeradas como 0 y 1 tienen predeterminadas la función de transmisión serial, sin embargo, también pueden ser reconfiguradas para otros usos, al igual que las demás terminales digitales.

Si en una terminal digital se dispone un 1 lógico, ello se traduce físicamente en una tensión o voltaje de aproximadamente 5V; por el contrario, si la terminal se presenta 0 lógico, ello equivale a una tensión o voltaje de aproximadamente 0V. Esta situación es válida tanto para la entrada de datos como para la salida.

Existe un límite máximo en la corriente que cada terminal puede suministrar al exterior cuando se configura como salida, de alrededor de 40mA, la cual no debe excederse; además se recomienda que la salida de corriente de todas las terminales, en conjunto, no exceda de 200 mA aproximadamente, aunque ello depende del modelo de placa.

MODULACIÓN POR ANCHO DE PULSO

Algunas terminales digitales tienen una marca o indicación junto a su numeración, lo que indica que, cuando son configuradas como de salida, soportan la modulación por ancho de pulso (PWM).

La modulación por ancho de pulso es una técnica que consiste en controlar el ancho del pulso de una señal digital de alta frecuencia, para obtener un valor promedio equivalente proporcional a la relación de los tiempos de duración del pulso alto entre la duración del pulso bajo, para cada ciclo de la señal digital.

La figura 4, muestra que cuando el ancho de pulso es alto, por ejemplo 90%, el nivel de tensión promedio es de 4,5 V; cuando es de 50% (denominado duty cycle) valor medio que se obtiene es 2,5V y cuando

es del 10%, el valor promedio de la tensión es de 0,5V.

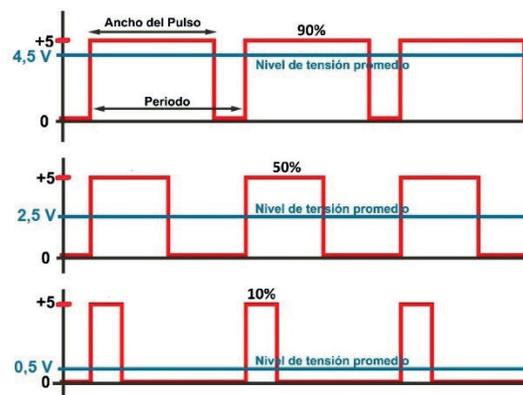


Figura 4: Modulación por ancho de pulso

El ancho de pulso puede ser regulado mediante instrucciones de software con muy alta precisión, para obtener cualquier valor comprendido entre 0 y 5V, lo cual equivale a un convertor de señal digital a señal analógica.

ENTRADAS ANALÓGICAS

Una característica de Arduino es el soporte que brinda para la adquisición de datos analógicos desde el exterior, para lo que dispone de un convertor analógico digital (CAD) que convierte las señales analógicas en valores binarios equivalentes, para poder procesarlos, transmitirlos o almacenarlos en forma digital, de forma similar a lo que se realiza con un archivo digital de audio o video, como MP3 o MP4.

El CAD incorporado en la placa Arduino tiene una resolución de 10 bits, lo que determina la subdivisión del rango de voltajes de 0 a 5 V en un conjunto de $2^{10} = 1024$ valores pudiendo discriminarse niveles de voltaje que se diferencian en $5/1024=0,0049$ V, que lo hace conveniente y propicio para una variedad de aplicaciones.

El rango de voltajes que se digitaliza es configurable por software, tanto en su valor máximo (no superior a 5V) y en su valor mínimo.

Arduino dispone también la entrada denominada Aref (referencia analógica), que permite establecer un nivel de referencia externa, la cual servirá de base para la conversión.



Los valores digitales que se obtienen del proceso de conversión se almacenan en variables estándar o arreglos dentro de la placa.

RESET

La placa Arduino dispone de un botón, tipo pulsador, para el reinicio de la misma, lo que provoca la ejecución del programa almacenado desde el principio.

LEDs INDICADORES

La placa contiene cuatro LEDs indicadores incorporados en la misma:

- Power ON, indicador de encendido
- TX, transmisión serial
- RX, recepción serial
- LED13, conectado en la salida digital 13.

COMUNICACIÓN SERIAL

La comunicación serial entre dispositivos es la más empleada actualmente en los sistemas de transmisión de datos digitales y es el sistema básico de comunicación de las placas Arduino. Está soportado por uno o más dispositivos UART o transmisor/receptor asincrónico serial, mediante los cuales se transmite cada byte de manera independiente de los demás y en cualquier instante.

El UART da el soporte básico para la comunicación por el puerto USB y para las terminales digitales denominadas Tx y Rx.

COMUNICACIÓN SPI

Arduino incluye el sistema de comunicación denominado SPI (Interface serial para periféricos), para la transferencia de datos, full dúplex (bidireccional simultáneo) de alta velocidad (hasta 10 MB) y distancias cortas (hasta 30 cm), entre diferentes dispositivos de hardware, como memorias, sensores, conversores, otras placas Arduino, etc., bajo el esquema: maestro/esclavo.

El dispositivo maestro inicia la comunicación, habilitando a un dispositivo esclavo, pudiendo intercambiar datos de forma sincrónica con el mismo.

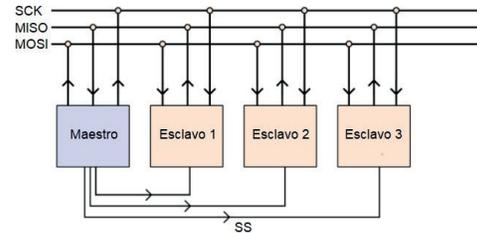


Figura 5: Comunicación SPI

El estándar utiliza cuatro cables, denominados:

1. MOSI, master out, slave in, transmisión del maestro al esclavo.
2. MISO, master in, slave out, transmisión del esclavo al maestro.
3. SCK, señal de clock de sincronización enviado por el maestro
4. SS, slave select, permite que el dispositivo maestro habilite a un esclavo, uno a la vez.

COMUNICACIÓN I²C

Arduino incorpora otro sistema de comunicación para el intercambio de datos, basado en el protocolo I²C que se caracteriza por emplear solo dos líneas denominadas SDA (serial data) y SCL (serial clock), y al igual que el protocolo SPI, usa un esquema maestro/esclavo, una velocidad de hasta 5 MB y un alcance de hasta 30 cm.

A diferencia del protocolo SPI, el I²C permite que cualquier dispositivo conectado sea maestro, lo que se define mediante un sistema de arbitraje, otorgando ese privilegio a un dispositivo a la vez. La transferencia de datos siempre lo inicia el dispositivo maestro, habilitando al esclavo específico mediante una dirección de 7 bits, lo que permite administrar hasta 128 dispositivos esclavos.



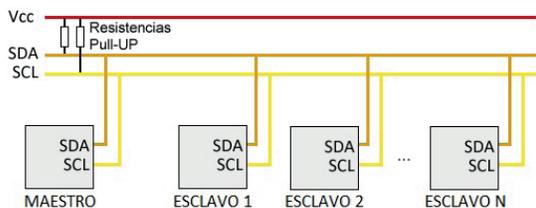


Figura 6: Comunicación I2C

FUENTE DE ALIMENTACIÓN

Los componentes electrónicos de la placa Arduino operan con 5 V y/o 3.3 V, dependiendo del modelo; estas tensiones se consiguen mediante uno o dos reguladores de tensión incorporados en la placa, en consecuencia, la alimentación de energía externa debe tener un voltaje mayor.

Arduino puede ser alimentada a través de tres medios distintos:

Fuente externa de corriente continua, con un valor comprendido entre 7 a 12 V CC, mediante un conector Jack, con la polaridad indicada en la figura 7. La fuente de suministro puede ser un adaptador universal de un mínimo de 500 mA, una batería de 9V, un juego de 4 pilas AA conectadas en serie o cualquier batería recargable que proporciones un voltaje en el rango establecido.



Figura 7: Conector Jack de alimentación

Un conector USB con alimentación de 5 V de cc, proporcionado por una computadora o un cargador de teléfono móvil.

Voltaje entre 7 y 9 V CC, directamente aplicado a la entrada Vin de la placa y negativo a la terminal GND. (<https://www.arduino.cc/en/Reference/Board>)

SHIELDS

Una shield, es una placa de expansión que puede conectarse a la placa Arduino para añadirle

funcionalidades extra.

Las placas shield están diseñadas para ser montadas directamente sobre la placa Arduino aprovechando las terminales de la misma y proveyendo nuevas terminales para otros dispositivos.

A la fecha, existe más de un centenar de shields, cuya descripción y conexión puede encontrarse en: <http://shieldlist.org/>



Figura 8: Montaje de placas Shield

SOFTWARE

El software del proyecto Arduino es open-source, desarrollado en Processing (<https://processing.org/>) y Java; su código fuente está disponible para su descarga en un repositorio de GitHub accesible desde el sitio oficial: (<https://www.arduino.cc/en/Main/Software>)

El software es un IDE (entorno de desarrollo integrado) libre, con funciones de edición, compilación y transferencia de programas a la placa, disponible en su versión 1.8.5, en el sitio indicado, disponible y actualizado al 13/10/17, para los sistemas operativos Windows, Linux y Mac, para 32 y 64 bits y ahora para Android.

El mismo sitio ofrece un editor Web, permanentemente actualizado con las últimas bibliotecas de los desarrolladores y con almacenamiento de programas en la nube o Internet.



El lenguaje de programación de la placa está basado en C++ estándar, y el compilador denominado avr-gcc, y las bibliotecas de funciones mínimas necesarias.

El software de Arduino llega con sus propias bibliotecas, algunas de las cuales son:

- Serial, para lectura y escritura por el puerto serie.
- Servo, para controlar servomotores.
- Stepper, para controlar motores paso a paso.
- Wire, para transmisión y recepción de datos I2C.
- LiquidCrystal, soporte para pantallas LCD.

La documentación del lenguaje se encuentra en: <http://arduino.cc/en/Reference/HomePage>

Diversos desarrolladores de hardware y software ponen a disposición bibliotecas para que algunos entornos, plataformas o diversos lenguajes de programación, interactúen con un programa que se ejecute en la placa Arduino, mediante una comunicación serial de datos. Entre los lenguajes que brindan ese soporte se encuentran:

C, C++, C#, Objective-C, Java, LabView, Mathematica, Matlab, PHP, Physical Etoys, Processing, Pure Data, Python, Ruby, Scratch, VBScript, Visual Basic .NET (<http://www.arduino.cc/playground/Main/InterfacingWithSoftware>)

Existen además diversas herramientas de desarrollo que dan soporte para la conectividad con Arduino, entre las más destacadas se encuentran:

- PlatformIO IDE
- Visual Studio Code Extension for Arduino
- Arduino for Visual Studio
- Programino IDE
- Deviot, para Sublime Text
- Sloeber, para Eclipse
- Eclipse AVR
- Biicode
- Pluto, para Python
- Embrio
- Zerynth

- Arduino for Atmel Studio (<https://playground.arduino.cc/Main/DevelopmentTools>)

CONCLUSIONES

El uso de una placa Arduino requiere un mínimo de conocimientos técnicos básicos, tanto de electrónica, como de programación; una vez superada esta etapa, su uso es simple, pudiéndose abordar proyectos de nivel básico e intermedio. Para encarar proyectos avanzados o más especializados, es necesario considerar otros aspectos técnicos a detalle.

La gran versatilidad de las placas Arduino, permiten aplicarlas en multitud de proyectos en diferentes áreas, por ejemplo: robótica móvil, robótica industrial, domótica, sistemas de vigilancia y seguridad, adquisición de datos, sistemas de control automáticos, inteligencia artificial, publicidad, tableros de control, etc., etc.; estando los campos de aplicación limitados solo por la creatividad de los usuarios.

Las posibilidades de aplicación de las placas Arduino crecen exponencialmente con la gran variedad de sensores, actuadores y placas shields existentes y la gran variedad de recursos de software de programación, plataformas y entornos de desarrollo disponibles.

REFERENCIAS

Hughes J. M. (2016) Arduino: A Technical Reference. USA: O'Reilly. 1ra Edición. 613 págs.

Evans M., Noble J., Hochenbaum J. (2013) Arduino in Action, Ed. Manning Publications Co.

Wheat D., (2011) Arduino Internals, New York, ed. Apress

Oxer J., Blemings H. (2009) Practical Arduino, Cool projects for open source hardware. New York, ed. Apress.

Página Web oficial de Arduino: <https://www.arduino.cc/> [consulta: 17/10/17].

