

Aprendiendo Arduino

Aprendiendo a manejar Arduino en profundidad

HW Arduino a Fondo

Los Arduino y en general los microcontroladores tienen puertos de entrada y salida y de comunicación. En Arduino podemos acceder a esos puertos a través de los pines.

- **Pines digitales:** <http://www.arduino.cc/en/Tutorial/DigitalPins>, pueden configurarse como entrada (para leer, sensores) o como salida (para escribir, actuadores)
- **Pines analógicos de entrada:** <https://www.arduino.cc/en/Tutorial/AnalogInputPins>, usan un conversor analógico/digital y sirven para leer sensores analógicos como sondas de temperatura.
- **Pines analógicos de salida (PWM):** <https://www.arduino.cc/en/Tutorial/PWM>, la mayoría de Arduino no tienen conversor digital/analógico y para tener salidas analógicas se usa la técnica PWM. No todos los pines digitales soportan PWM.
- **Puertos de comunicación:** USB, serie, I2C y SPI

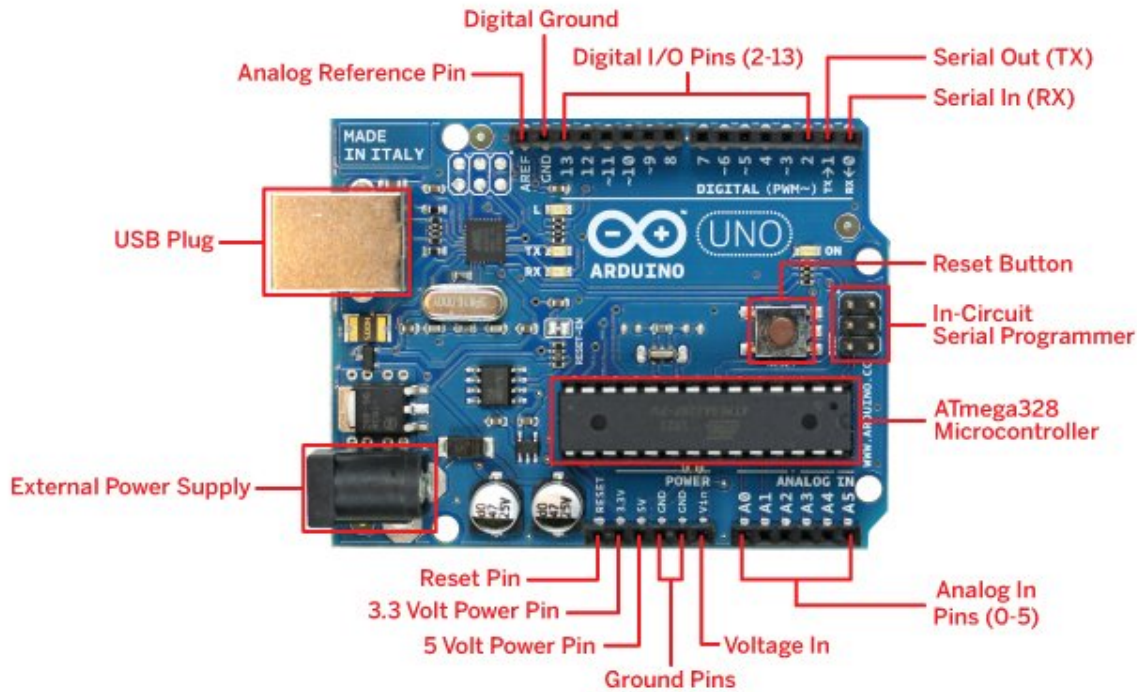
Otro aspecto importante es la memoria, Arduino tiene tres tipos de memoria:

- **SRAM:** donde Arduino crea y manipula las variables cuando se ejecuta. Es un recurso limitado y debemos supervisar su uso para evitar agotarlo.
- **EEPROM:** memoria no volátil para mantener datos después de un reset o apagado. Las EEPROMs tienen un número limitado de lecturas/escrituras, tener en cuenta a la hora de usarla.
- **Flash:** Memoria de programa. Usualmente desde 1 Kb a 4 Mb (controladores de familias grandes). Donde se guarda el sketch.

Más información en:

- <http://arduino.cc/en/Tutorial/Memory>
- <https://aprendiendoarduino.wordpress.com/2015/03/29/memoria-flash-sram-y-EEPROM/>

Placa Arduino Uno a fondo:

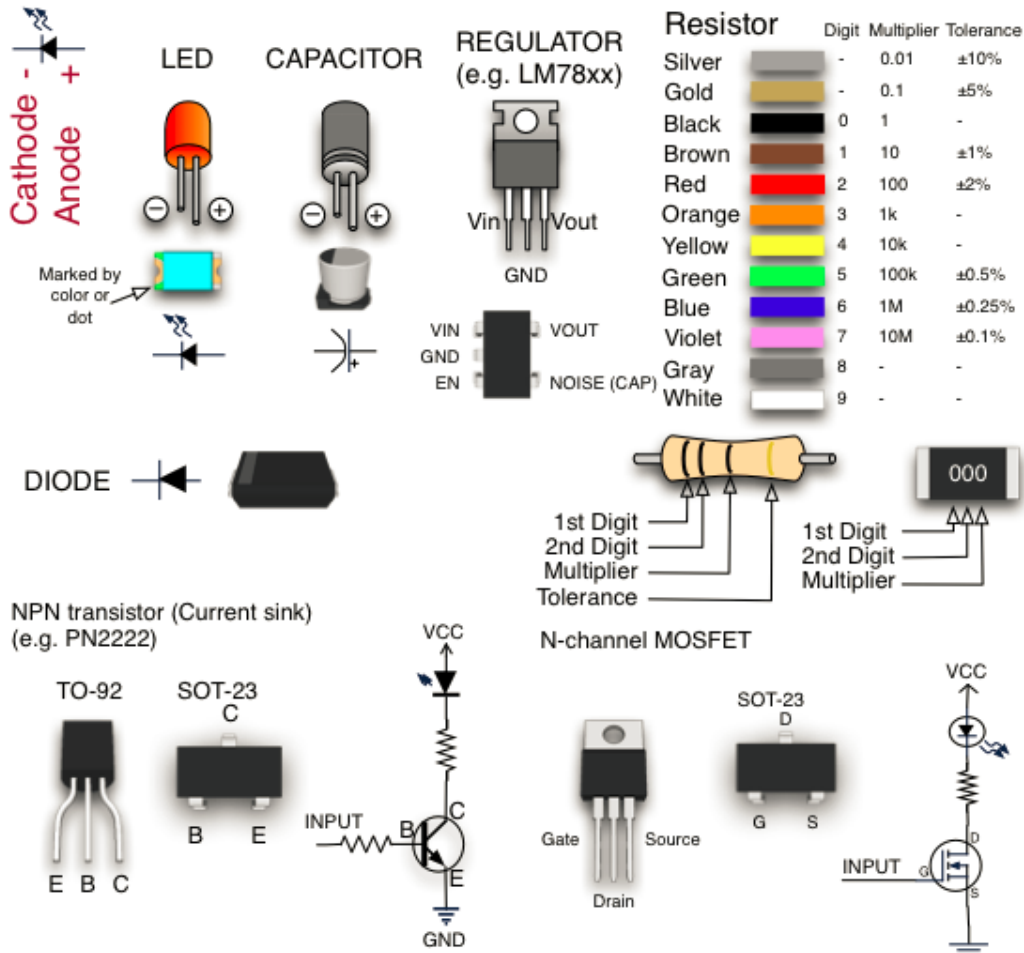


Especificaciones detalladas de Arduino UNO: <http://arduino.cc/en/Main/ArduinoBoardUno>

Microcontroller & USB-to-serial converter	ATmega328P & Atmega16U2
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Todos los componentes del Arduino UNO:





Más ping mapping y conexiones de HW en:

- <http://www.pighixxx.com/test/>
- <https://www.kickstarter.com/projects/pighixxx/abc-basic-connections-the-essential-book-for-maker>

HW de Arduino a fondo: <https://learn.adafruit.com/arduino-tips-tricks-and-techniques/arduino-uno-faq>

Componentes Arduino UNO:

- No necesita de un cable FTDI para conectarse al MCU, en su lugar uso una MCU ATMEGA16U2 especialmente programado para trabajar como conversor de USB a serie.
- Alimentación: vía USB, batería o adaptador AC/DC a 5V, seleccionado automáticamente. Arduino puede trabajar entre 6 y 20V, pero es recomendado trabajar entre 7 y 12V por las características del regulador de tensión.
- Puerto Serie en los pines 0 y 1.
- Interrupciones externas en los pines 2 y 3.
- Built-in LED en el pin 13.
- Bus TWI o I2C en los pines A4 y A5 etiquetados como SDA y SCL o pines específicos
- El MCU ATmega328P tiene un bootloader precargado que permite cargar en la memoria flash el nuevo programa o sketch sin necesidad de un HW externo.
- Fusible rearmable de intensidad máxima 500mA. Aunque la mayoría de pc's ya ofrecen protección interna se incorpora un fusible con la intención de proteger tanto la placa Arduino como el bus USB de sobrecargas y

cortocircuitos. Si circula una intensidad mayor a 500mA por el bus USB(Intensidad máxima de funcionamiento), el fusible salta rompiendo la conexión de la alimentación.

- Regulador de voltaje LP2985 de 5V a 3.3V que proporciona una corriente de alimentación máxima de 150 mA.
- Regulador de voltaje NCP1117 que proporciona un valor estable de 5V a la placa y soporta por encima de 1 A de corriente. Datasheet: http://www.onsemi.com/pub_link/Collateral/NCP1117-D.PDF
- ATMEGA16U2 => Es el chip encargado de convertir la comunicación del puerto USB a serie.
- Condensadores de 47µF de capacidad
- Diodo M7 en la entrada de alimentación de la placa. Con este diodo conseguimos establecer el sentido de circulación de la intensidad, de esta forma si se produce una contracorriente debido a la apertura de un relé u otros mecanismos eléctricos, el diodo bloquea dicha corriente impidiendo que afecte a la fuente de alimentación.
- DFU-ICSP. Puerto ICSP para el microcontrolador ATMEGA16U2, como en el caso del ATMEGA328P-PU se emplea para comunicarnos con el microcontrolador por el serial, para reflashearlo con el bootloader, hacer algunas modificaciones, ponerlo en modo DFU, etc..
- JP2. Pines libres del ATMEGA16U2, dos entradas y dos salidas para futuras ampliaciones.
- Encapsulados de resistencias.
- RESET-EN: Significa Reset enabled o reset habilitado. Está habilitado el auto-reset, para deshabilitar por cualquier tipo de seguridad (por ejemplo un proyecto que tenemos funcionando y no queremos que nadie lo reinicie al conectar un USB y detecte un stream de datos) debemos desoldar los pads RESET-EN y limpiarlos de forma que estén aislados el uno del otro.
- Cristal oscilador de 16MHz necesario para el funcionamiento del reloj del microcontrolador ATMEGA16U2.
- Resonador cerámico de 16 Mhz para el microcontrolador ATMEGA328P-PU. Los resonadores cerámicos son menos precisos que los cristales osciladores, pero para el caso hace perfectamente la función y ahorramos bastante espacio en la placa. Se trata del pequeño, porque el cristal grande es para el 16U2

Diferencias entre las diversas versiones de HW de los

Arduino: <http://startingelectronics.com/articles/arduino/uno-r3-r2-differences/>

Más información:

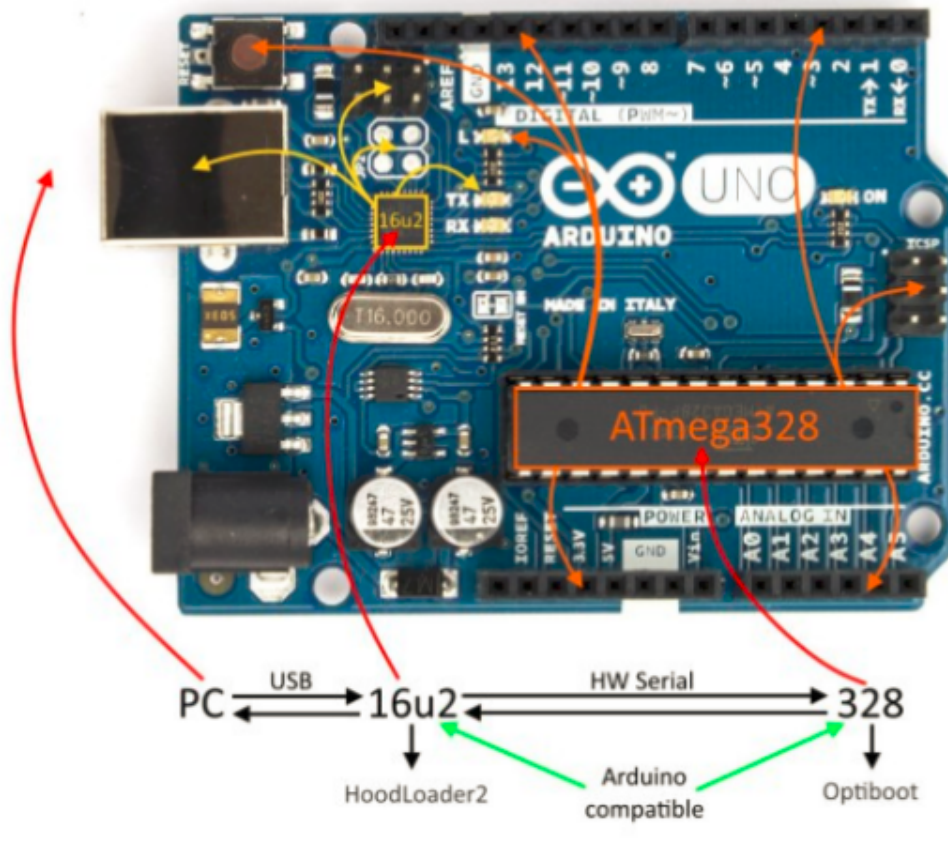
- <http://tallerarduino.com/2013/06/28/arduino-pinout-y-conexiones-basicas/>
- <https://learn.adafruit.com/arduino-tips-tricks-and-techniques/arduino-uno-faq>

MCU ATmega16u2 en Arduino

Si nos fijamos en el pequeño integrado que hay en la placa de Arduino UNO junto al conector USB, se trata de un **ATmega16u2** cuya misión es dar el interfaz USB al Arduino UNO y comunicar los datos con el **ATmega328p** mediante el puerto serie. Se podría usar como microcontrolador completamente funcional y no solo un conversor de USB a Serial con ciertas modificaciones. Podríamos usar ambas MCUs en la misma placa, pudiendo descargar trabajo de la MCU principal en la secundaria.

Para ello es posible usar el bootloader2 en el Atmega16U2 o Atmega8U2 dependiendo de la versión de Arduino Uno que tengamos y comunicamos ambas MCUs por HW serial

HoodLoader2 - The Power of 2 Microcontrollers

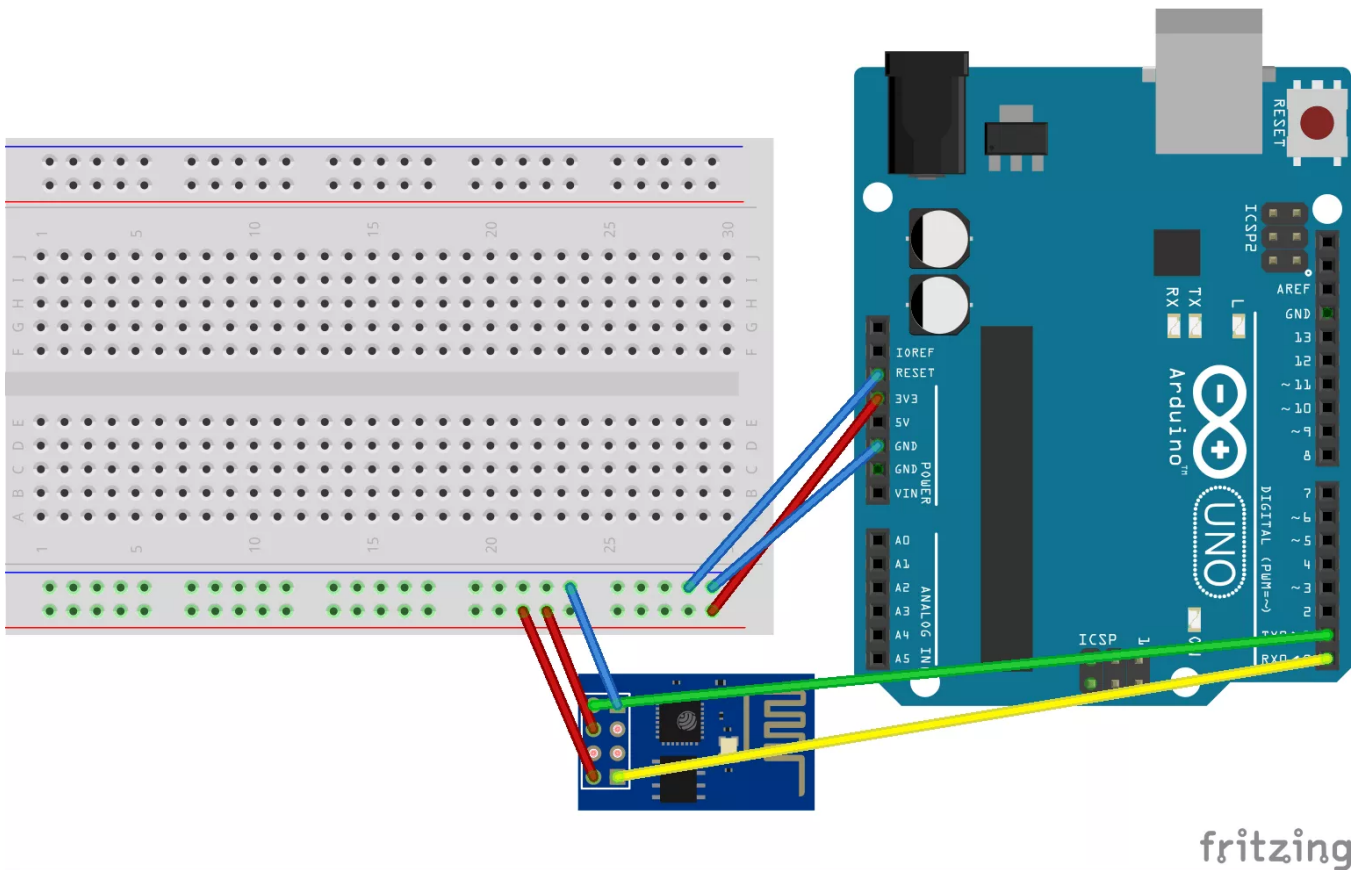


(c) 2014 NicoHood

Como usar el segundo MCU del Arduino UNO: <http://www.freetronics.com/blogs/news/16053025-using-both-microcontrollers-on-your-arduino-uno-compatible-board#.Vlg48zGG9B9>

HoodLoader2: <https://github.com/NicoHood/HoodLoader2>

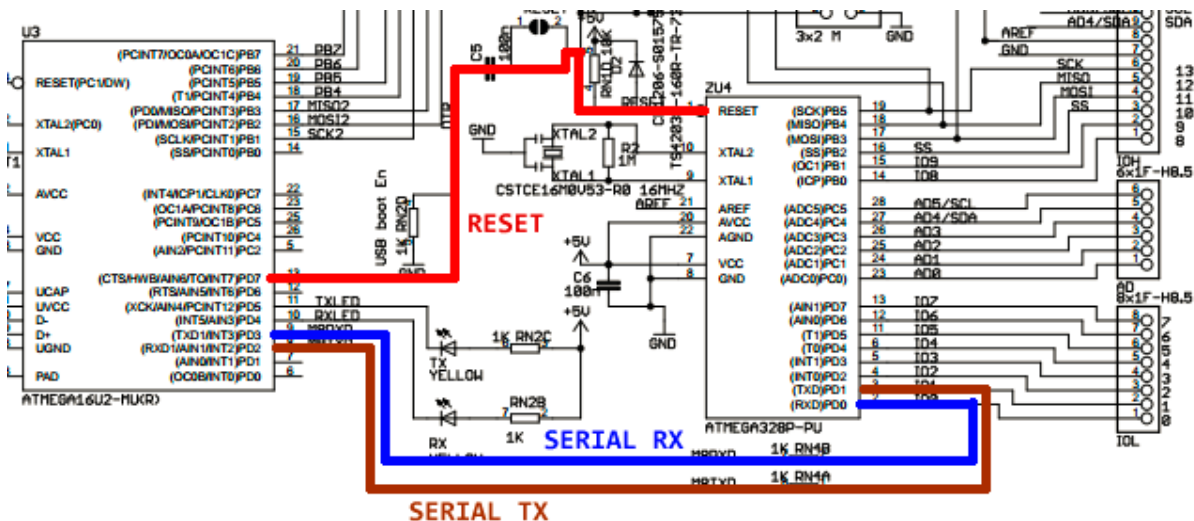
Además usaremos el microcontrolador ATmega16U2 para poder programar y conectar por USB el módulo ESP8266 ESP-01 que disponemos para prácticas.



fritzing

En este esquema estamos conectando directamente los puertos Tx y Rx del ATmega16u2 con los del ESP8266 (puertos serie), de forma que el ATmega16U2 hace de conversor USB a TTL para poder acceder al ESP8266 desde USB. Debe tenerse en cuenta que el ATmega328p de Arduino también está conectado a Tx y Rx, por lo que habrá que tener cargado algún programa que no use la UART de Atmega328p (cualquiera que con tenga Serial.begin).

Conexión de los puertos serie de las dos MCUs de un Arduino UNO:

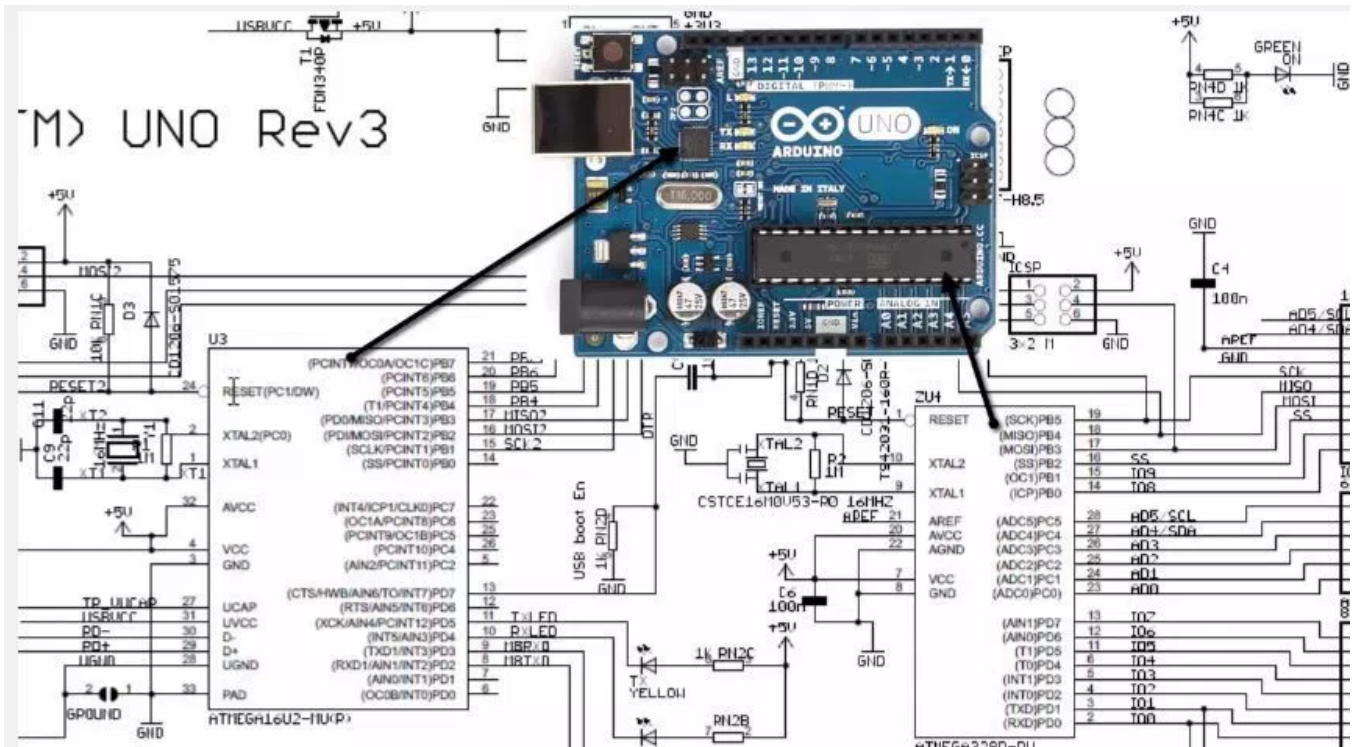


Mucha más información sobre el 16U2 en: <http://arduino.stackexchange.com/questions/13292/have-i-bricked-my-arduino-uno-problems-with-uploading-to-board>

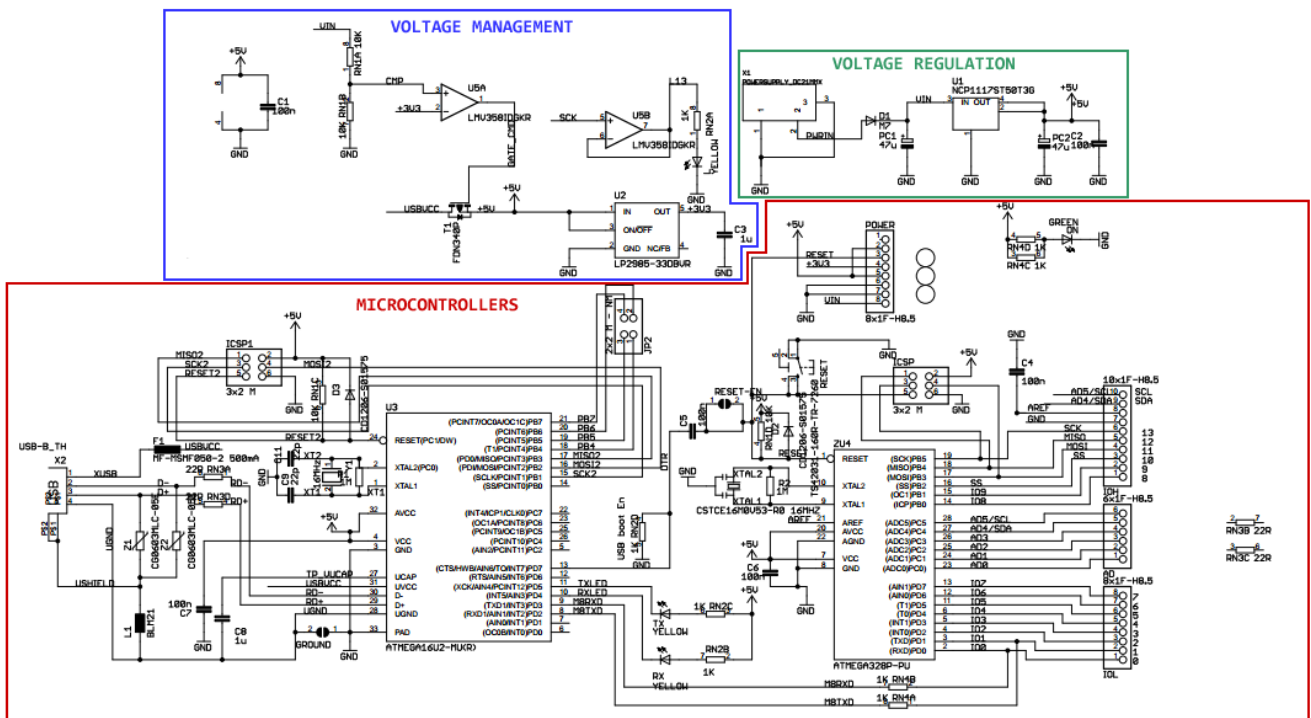
Esquemático Arduino UNO

En este pdf podemos ver el esquema de un Arduino UNO, muy importante conocerlo para evitar hacer maniobras que lo dañen: http://arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf

Los dos microcontroladores:



Partes del esquemático:



También es importante conocer cómo están distribuidos los pines del MCU en Arduino:

- Pin Mapping: <http://arduino.cc/en/Hacking/PinMapping168>
- http://www.umsats.ca/wp-content/uploads/2013/02/Arduino_uno_Pinout.pdf

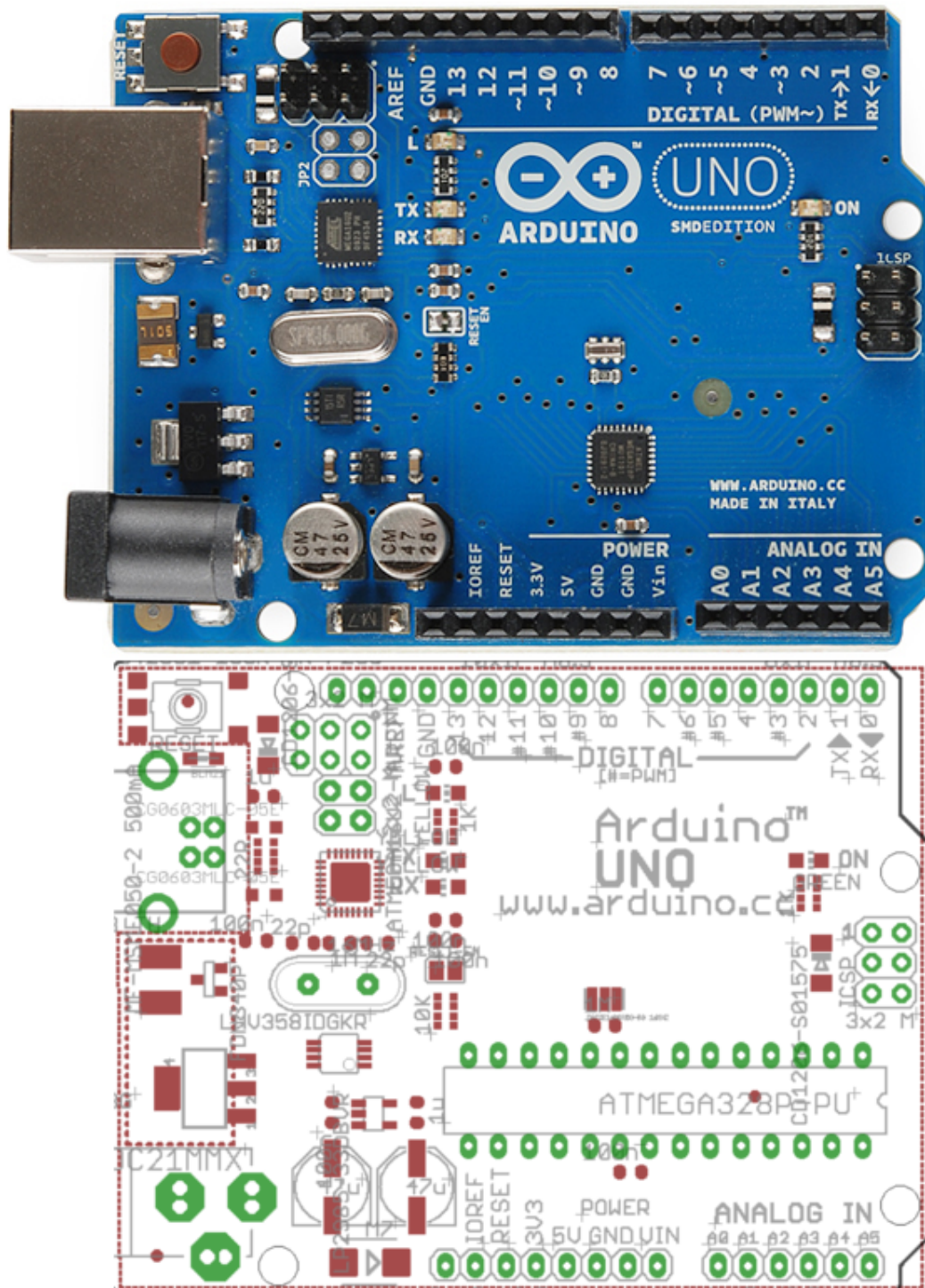
- http://pighixx.com/atmega328v3_0.pdf
- <http://pighixx.com/unov3pdf.pdf>
- Y más pinout de diversas placas, MCUs, etc... <http://www.pighixx.com/test/pinoutspg/>

Para saber todo sobre el HW de Arduino ver este tutorial donde desglosa todo el HW de Arduino para construir un Arduino UNO desde cero y crear tu propio clon:

<https://rheingoldheavy.com/category/education/fundamentals/arduino-from-scratch-series/>

Diseño PCB Arduino

La placa de Arduino:



Están disponible los esquemas y diseño en formato Eagle para Arduino UNO en http://arduino.cc/en/uploads/Main/arduino_Uno_Rev3-02-TH.zip. Por supuesto para el resto de Arduinos también disponemos de sus diseños de PCB.

Para ver los esquemas podemos usar Eagle, se trata de un programa de diseño de diagramas y PCBs con autoenrutador:

- <http://www.cadsoftusa.com/eagle-pcb-design-software/product-overview/>
- Versión Freeware <http://www.cadsoftusa.com/download-eagle/freeware/>

La versión freeware de Eagle es perfecta para diseños pequeños y sencillos, se trata de una licencia para uso no comercial y gratuita para todos. La versión freeware tienen todas las funcionalidades de la versión de pago pero tiene ciertas limitaciones:

- El área de la placa está limitada a 100 x 80 mm
- Solo pueden usarse dos capas (Top y Bottom)
- El editor del esquemático solo puede tener dos hojas.
- Soporte solo disponible vía email o foro
- Su uso está limitado a aplicaciones no comerciales o para evaluación
- Se puede cargar, ver e imprimir diseños que superen esos límites.

Una opción opensource para diseño de PCB es kicad: <http://kicad-pcb.org/>

PRÁCTICA: Instalar eagle y ver algunos de los planos de arduino que nos podemos descargar y ver algunos ejemplos. También probar a modificar algún elemento de los planos.

Webinar de Eagle: <https://www.youtube.com/watch?v=2aFNypxILu0>

Anuncios

Esta entrada se publicó en Arduino, Curso Avanzado 2017, Hardware y está etiquetada con Arduino, Curso Avanzado 2017, Hardware en 11 agosto, 2017 [<https://aprendiendoarduino.wordpress.com/2017/08/11/hw-arduino-a-fondo-2/>] .

