

i RECUERDE

El que resulte sencillo programar Arduino no significa que este usando una tecnología simple, el Software está ocultando la complejidad del Hardware para hacerle el trabajo más simple pero no puede hacer magia y resolver un cableado desprolijo o fuentes de alimentación defectuosas.

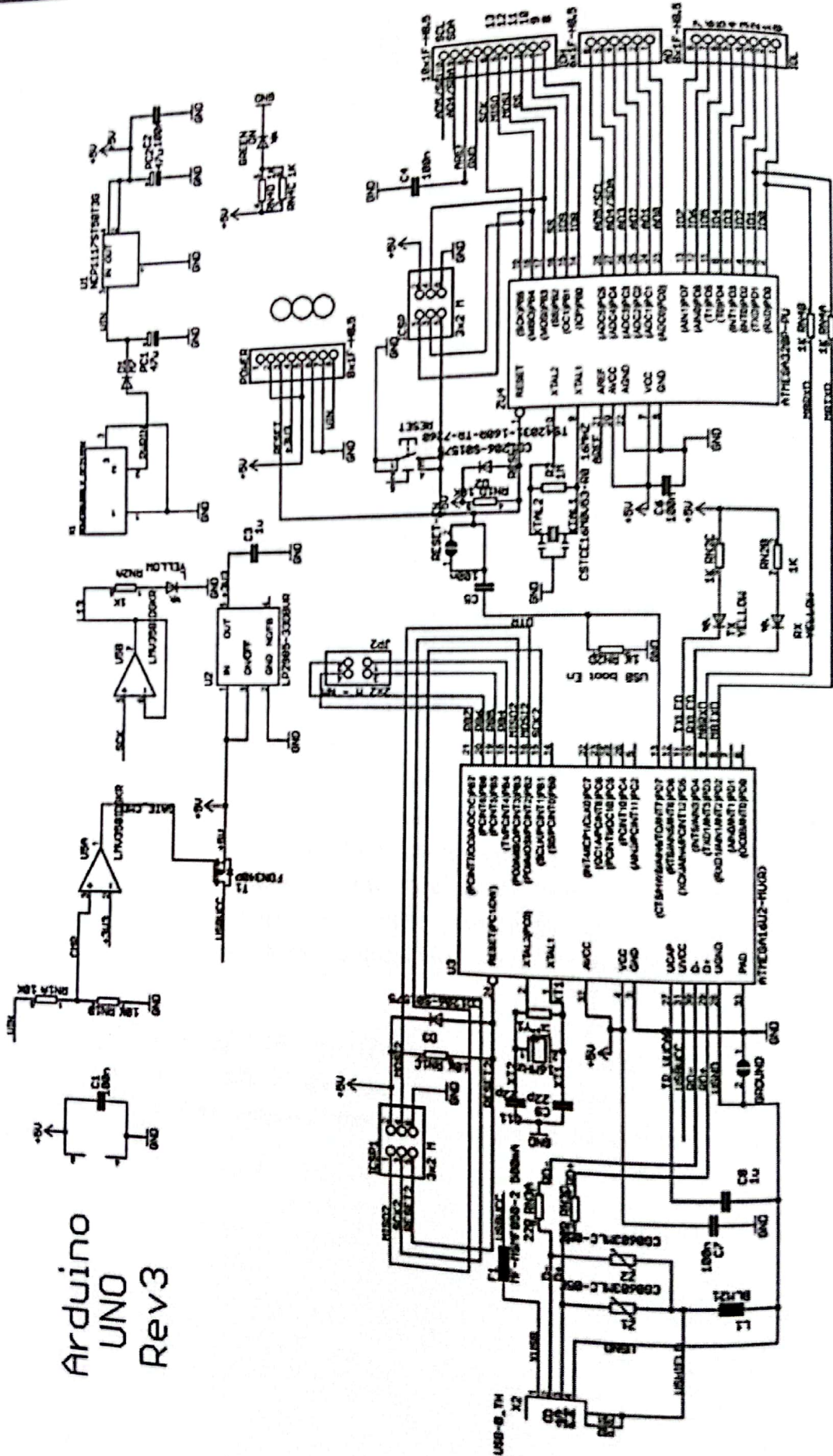
3.10 CIRCUITO ELECTRÓNICO DE LA PLACA ARDUINO UNO

El siguiente es el circuito completo de la placa Arduino UNO. La placa se puede alimentar directamente desde el USB o con una fuente externa de 9V, uno de los operacionales de circuito integrado LM358 es el encargado de “ver” desde donde estamos alimentando la placa y seleccionar solo una fuente de alimentación, el otro operacional del LM358 (tiene dos operaciones internos) se encarga de manejar el LED conectado al pin 13 de tal forma que el encendido del LED no cargue a salida.

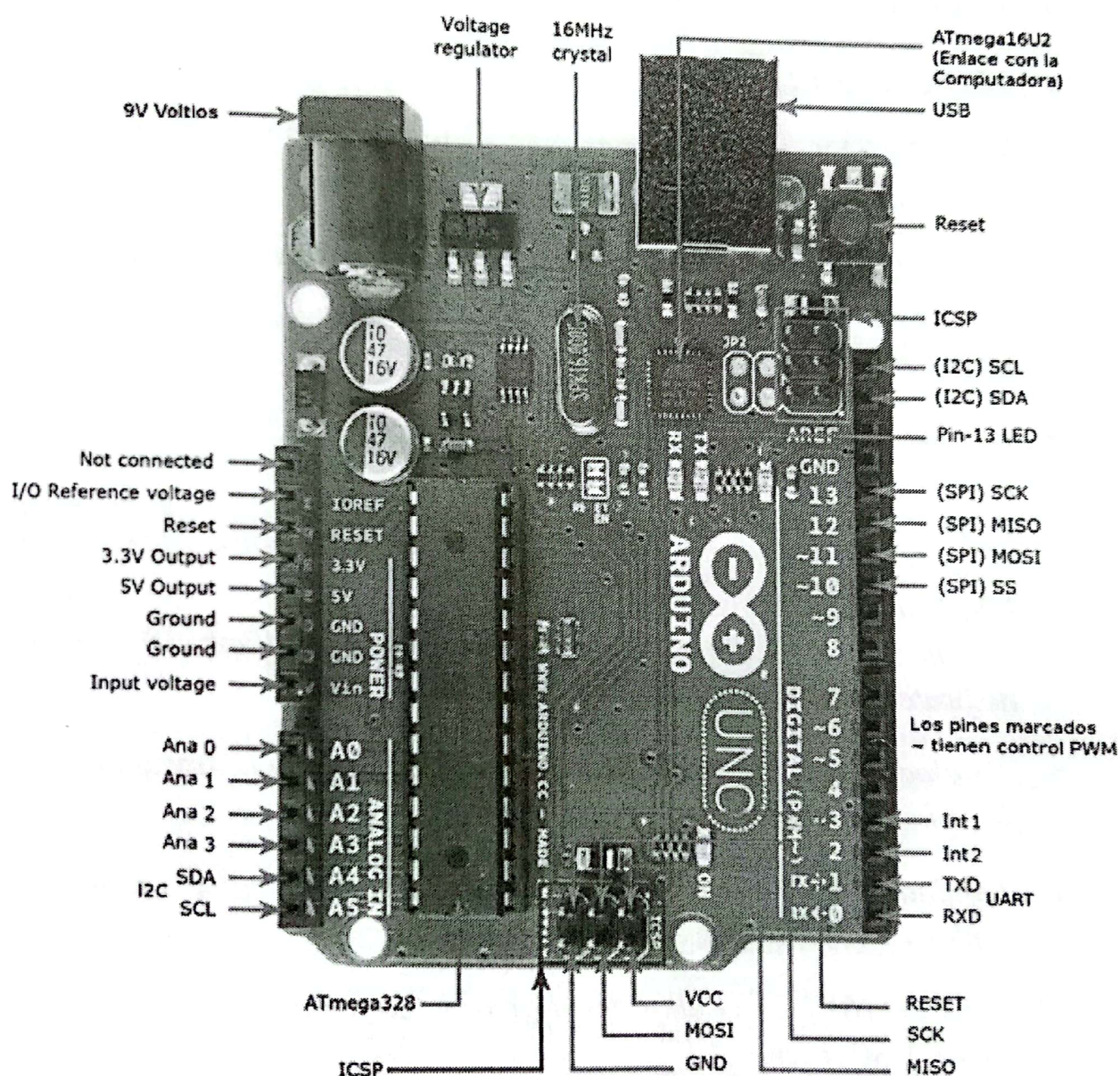
Dos reguladores se encuentran a bordo de la placa Arduino, un LPC2985 es el encargado de generar los 3.3 Voltios y el NCP1117 suministra los 5Voltios.

Hay que remarcar que el regulador de 3.3 Voltios saca energía del regulador de 5 Voltios por lo tanto la carga de 3,3 V se suma a la carga total del regulador de 5Voltios, si excede el Amper de corriente el regulador se destruirá. **Vigile la temperatura de este regulador.**

Arduino UNO Rev3



4.1 LA PLACA ARDUINO UNO



En la placa se pueden ver los pines que corresponden a los puertos, el cristal de 16 Mhz que fija la velocidad de operación, una pequeña fuente de alimentación, un botón de reset y algunos componentes periféricos necesarios para el correcto funcionamiento del microcontrolador.

4.2 COMPONENTES DE LA PLACA ARDUINO UNO

- ▼ *No necesita ningún puente USB para conectarse al MCU, en su lugar usa un MCU ATMEGA16U2 especialmente programado para trabajar como convertidor de USB a serie.*
- ▼ *La alimentación puede ser 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.*
- ▼ *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 programador 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. (No exagere su confianza en este sistema de protección)*
- ▼ *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.*
- ▼ *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 reprogramarlo con el bootloader, hacer algunas modificaciones, 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.*

Un comentario especial merece el pequeño integrado que hay en la placa de Arduino UNO junto al conector USB, se trata de un ATmega16u2 cuya misión es crear una interfaz USB para el Arduino UNO y comunicar los datos con el ATmega328p mediante el puerto serie.

Se podría usar como microcontrolador completamente funcional y no solo un convertor de USB a puerto serie, con algunas modificaciones podríamos usar ambos microcontroladores en la misma placa. Pero para eso debemos usar otra forma de conexión a la placa Arduino ya que el puente USB/Serie se perdería, realmente habría que evaluar si esto realmente vale la pena.

El microcontrolador de Arduino UNO es de 8 bits, la diferencia entre los microcontroladores de 8, 16, y 32 bits, es tamaño de palabra o datos que manejan e