

Aprendiendo Arduino

Aprendiendo a manejar Arduino en profundidad

Alimentación Arduino

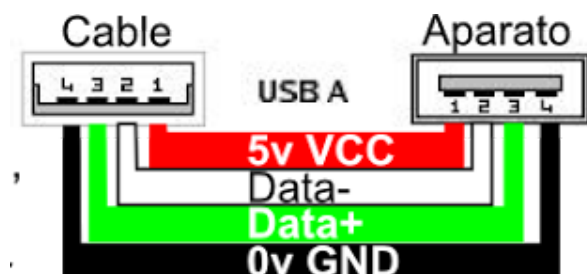
Una de la partes más importantes a la hora de afrontar un proyecto con Arduino o cualquier microcontrolador es como vamos a alimentarlo y en caso que haya que usar baterías que autonomía va a tener. Pero no solo es alimentar Arduino, sino que hay que tener en cuenta que además es posible que estemos alimentando a través del microcontrolador los sensores y actuadores y si no tenemos en cuenta las limitaciones eléctricas del microcontrolador, podemos encontrarnos que no funcione el proyecto.

Cuanto consume Arduino: <http://www.prometec.net/consumos-arduino/>

Métodos de Alimentar Arduino

USB

Cuando se trabaja con Arduino normalmente lo alimentas a través del cable USB que va al ordenador. Sin embargo, una vez programado, o bien se deja conectado al ordenador o bien se busca otra forma alimentar Arduino. Alimentar Arduino por USB tiene la desventaja de que solo es capaz de suministrar hasta 500 mA.



Adaptador de Corriente

El método más habitual de alimentar Arduino es mediante un **adaptador de corriente (AC DC)**. Este alimentador debe de dar un voltaje adecuado al arduino que conectemos y la potencia suficiente. Un voltaje adecuado para el Arduino UNO es un alimentador de 7 o 7,5V puesto que por debajo de ese voltaje es posible que la salida 5V de Arduino esté por debajo y por encima el regulador de tensión se calentará innecesariamente. A partir de 12V es desaconsejable Alimentar Arduino porque el regulador de tensión se calienta mucho y puede acabar dañándose. Esta opción es la mejor si el proyecto no necesita moverse y si hay una toma de corriente accesible cerca de Arduino.

Fuente de Alimentación Regulable

Para propósitos de testeo y desarrollo, podríamos usar una **fuentes de alimentación regulable** y usarla para alimentar Arduino, pero en un entorno de laboratorio.



Baterías

Otra opción para alimentar Arduino es mediante **baterías de 9V**, pero apenas tienen duración en un Arduino, puesto que el diseño de un Arduino no está pensado en un bajo consumo, por lo que salvo para una pequeña demostración no es una solución válida. Estas pilas tienen una capacidad entorno a 300 mAh y un proyecto sencillo con Arduino puede consumir 200 MAh, por lo que no llegará a una hora de autonomía.

Las típicas **pilas AA** proporcionan 1,5V. Se pueden poner varias pilas en serie hasta llegar al voltaje necesario (lo ideal es poner 5 o 6) y alimentar Arduino con ellas. La diferencia entre utilizar estas pilas y utilizar las de 9V es enorme. Una sola pila alcalina AA tiene entre 2700-2900 mAh (por los 300 mAh de una recargable de 9V), por lo que son una muy buena opción. La única pega que tiene esta opción es que como pilas que son se gastan, y tendrás que comprar pilas a menudo. Aun así una opción muy recomendable.

Las **baterías LiPo** (Polimero de Litio). Esta batería tiene mucha duración (tienen muchos mAh). Tienen una vida útil bastante larga, lo que hace que compense utilizar este recurso aunque sean más caras que las pilas AA. Sin lugar a dudas son la mejor opción para alimentar un coche a radiocontrol, dron o cualquier dispositivo móvil que se te ocurra. Existen baterías LiPo de distintos voltajes y capacidades. En el caso de alimentar Arduino, la mejor opción es utilizar una batería de 7,4V y, al menos, 1600 mAh.

Las baterías LiPo se componen de celdas y cada celda es de 3,7V, por lo tanto los voltajes que podemos encontrar de estas baterías son múltiplos de 3,7V.

Estas baterías son recargables y debemos cargarlas adecuadamente para alargar su vida, para ello disponemos de módulos cargadores de baterías LiPo.

Más información sobre baterías:

- <https://learn.sparkfun.com/tutorials/battery-technologies>
- <http://blog.bricogeeek.com/noticias/modelismo/todo-sobre-las-baterias-lipo/>

Otra forma de alimentar un Arduino es mediante una placa solar y un acumulador, pero este tema se tratará un poco después en este post.

Más información en:

- <https://learn.sparkfun.com/tutorials/how-to-power-a-project>
- <http://www.educachip.com/alimentar-arduino/>
- <http://soloarduino.blogspot.com.es/2014/12/regulacion-de-tension-explicacion.html>

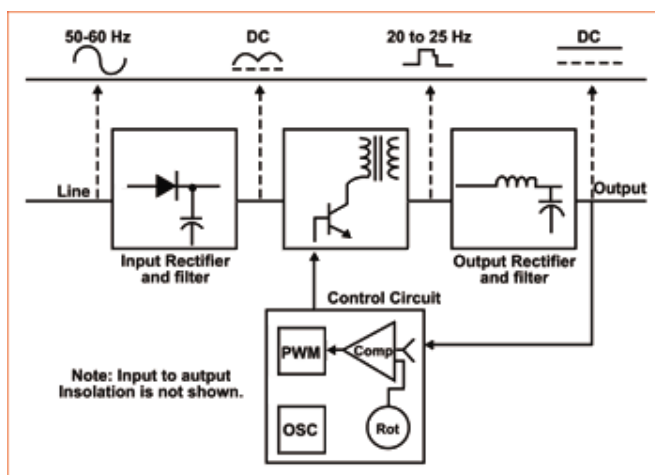
- <https://www.modmypi.com/blog/how-do-i-power-my-arduino>
- <http://txapuzas.blogspot.com.es/2009/12/alimentacion-para-paperduino-5v-12v.html>

Fuente conmutada

Otra forma de alimentar un Arduino es usar una **fuentes conmutada** como las fuentes ATX de los ordenadores. Una fuente conmutada es un dispositivo electrónico que transforma energía eléctrica mediante transistores en conmutación. Mientras que un regulador de tensión utiliza transistores polarizados en su región activa de amplificación, las fuentes conmutadas utilizan los mismos conmutándolos activamente a altas frecuencias (20-100 kilociclos típicamente) entre corte (abiertos) y saturación (cerrados). La forma de onda cuadrada resultante es aplicada a transformadores con núcleo de ferrita para obtener uno o varios voltajes de salida de corriente alterna (CA) que luego son rectificadas (con diodos rápidos) y filtrados (inductores y condensadores) para obtener los voltajes de salida de corriente continua (CC).

Las ventajas de este método incluyen menor tamaño y peso del núcleo, mayor eficiencia y por lo tanto menor calentamiento. Las desventajas comparándolas con fuentes lineales es que son más complejas y generan ruido eléctrico de alta frecuencia que debe ser cuidadosamente minimizado para no causar interferencias a equipos próximos a estas fuentes.

Además con esta fuentes tenemos suficiente potencia para alimentar a Arduino y los componentes que conectemos como LEDs, motores, etc...



Ejemplo de fuente de alimentación conmutada: <http://electronicaraya.com/?q=catalogo/753>



Se podría hacer una fuente conmutada controlada por Arduino: <http://es.slideshare.net/pablogindelfuente-conmutada-con-arduino>

Cómo gobernar una fuente ATX con Arduino: <http://www.prometec.net/fuentes-atx/>

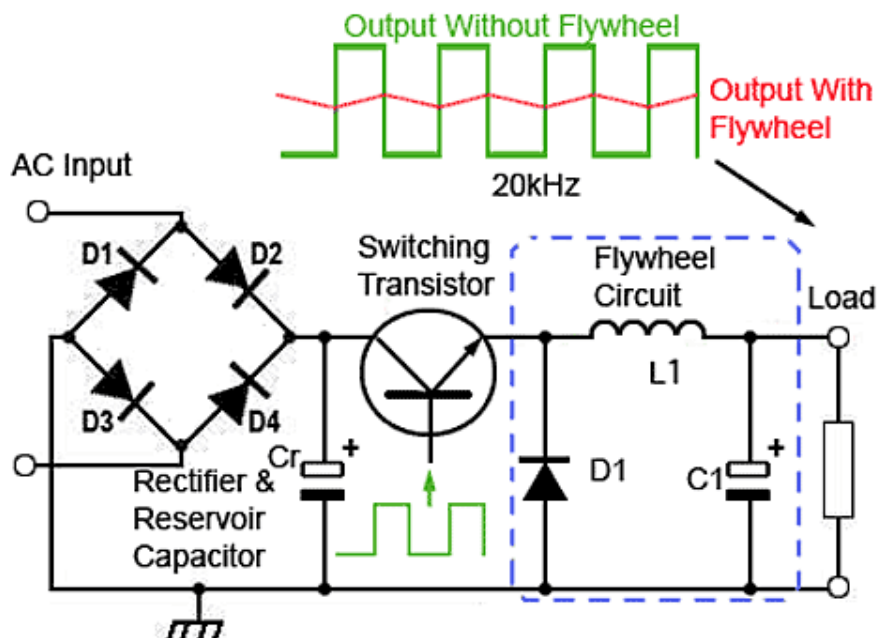
Fuente de alimentación reciclada para uso con Arduino: <https://sites.google.com/site/arduinoreciclar/fuente-de-energia>

Fuente de alimentación con Arduino: <http://www.urpmigeek.com/2012/06/diy-fuente-de-alimentacion-con-arduino.html>

Más información de fuentes conmutadas

- https://es.wikipedia.org/wiki/Fuente_conmutada
- Fuentes de alimentación: <http://www.prometec.net/fuentes-de-alimentacion/>
- <http://www.electan.com/fuente-alimentacion-conmutada-12v-21a-25w-p-2891.html>
- <http://www.prometec.net/fuentes-de-alimentacion-conmutadas/>
- Fuente de alimentación conmutada step-down o buck converters: <http://www.prometec.net/fuentes-step-down/>
- Buck converter: https://es.wikipedia.org/wiki/Convertidor_Buck
- Fuente de alimentación step-up: <http://www.prometec.net/boost-converters/>

Este tipo de circuitos reciben el nombre de fuentes conmutadas, convertidores Buck (Buck converters), o fuentes Step Down (De escalón de bajada) y son la base de las fuentes ATX.



La virtud de los buck converters o fuentes de Step Down es que proporcionan un buen rendimiento en la conversión de energía de una batería por ejemplo, sin desperdiciarla en forma de calor, frente a los reguladores de tensión lineales. Estas fuentes son ideales si estamos alimentando un equipo con baterías 12V, como por ejemplo en un coche, y deseamos alimentar nuestro Arduino para cualquier invento. Nos permiten bajar la tensión de 12V de la batería a los 5 V que necesitamos en nuestra entrada Vin con garantías de no desperdiciar energía de la batería en forma de calor, y por tanto extender sensiblemente su duración, al contrario de los reguladores lineales que veremos después.

En general el mercado nos provee de multitud de circuitos montados con fuentes conmutadas y aunque son algo más caras que los simples reguladores lineales, los precios han caído tanto que por unos pocos euros se consigue un Buck converter de calidad.

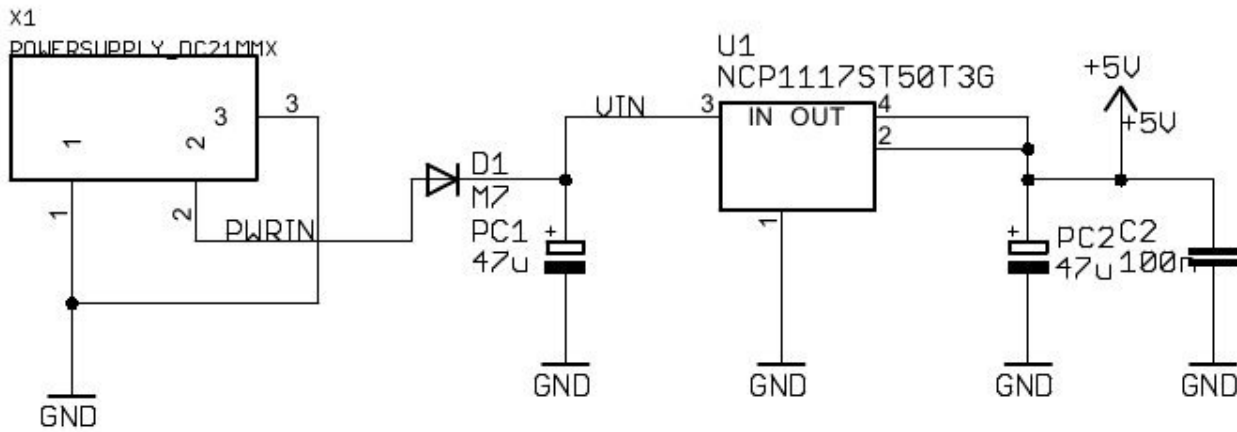
Ejemplo de fuentes step-down conmutadas:

- <http://www.prometec.net/producto/regulador-lm2596s-dc-dc-step-down/>
- <http://www.prometec.net/producto/fuente-conmutada-lm2596s-step-down/>
- <https://www.pololu.com/category/133/step-up-step-down-voltage-regulators>

Un ejemplo de un conversor DC/DC conmutado step-down es el TRACO TSR 1-2450. Datasheet: <https://cdn-shop.adafruit.com/datasheets/tsr1.pdf>. Consigue hasta un 96% de eficiencia sin necesidad de disipador y es el sustituto eficiente de un regulador lineal (LDO) como el LM7805. Además estos conversores tienen una salida más exacta ($\pm 2\%$), menor corriente en standby (2mA) y no requiere de condensadores externos al contrario que los reguladores lineales (LDO). Son ideales para aplicaciones alimentadas con baterías. El problema de estos conversores es que son más caros que los reguladores lineales.

Para alimentar un Arduino con un TSR 1-2450, habría que inyectar la corriente al pin de 5V pero no tendríamos de protección alguna.

Si lo alimentamos a Arduino a través del pin Vin, estaríamos pasando por el regulador NCP1117 del Arduino y perderíamos toda la eficiencia. Además en este caso necesitaríamos más de 6V y entonces habría que usar el TSR 1-2490.



Más información sobre TRACO power: <http://www.tracopower.com/home/>

También hay otros DC-DC converters pero aislados usando un transformador como el THD 12-2412WI: <http://docs-europe.electrocomponents.com/webdocs/0aca/0900766b80acafd9.pdf>

Application note muy interesante que compara y aconseja en caso usar entre los reguladores lineales y las fuentes conmutadas: <http://cds.linear.com/docs/en/application-note/AN140fa.pdf>

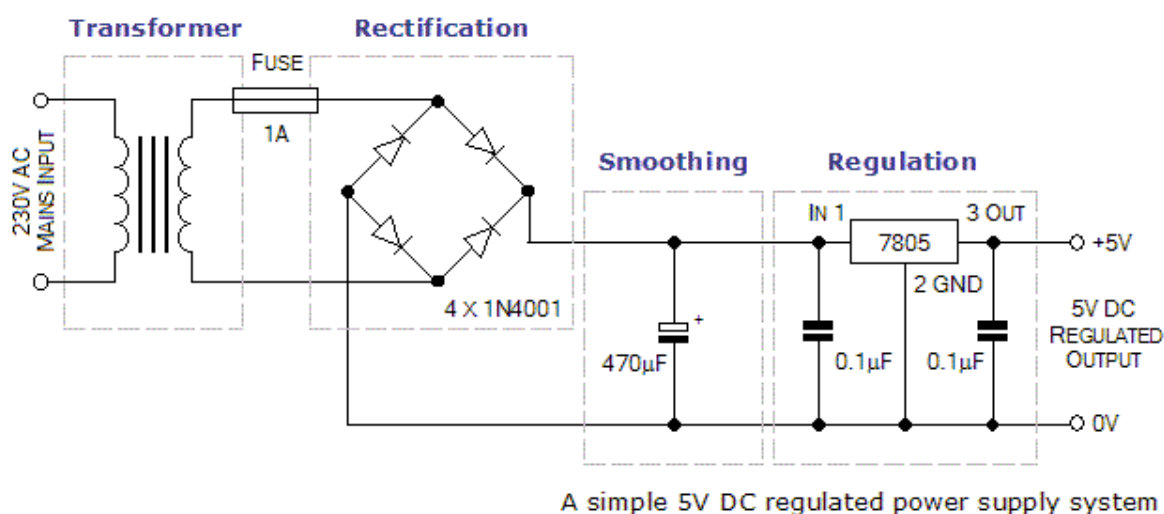
Video que compara buck converter frente a un linear regulator como el 7805: <https://www.youtube.com/watch?v=giGRrODKJSE>

Linear vs switching regulator en aplicaciones industriales de 24V: : <http://www.ti.com/lit/an/slyt527/slyt527.pdf>

Más información: <http://www.intersil.com/en/products/power-management/linear-vs-switching-regulators.html>

Se podría alimentar también un arduino con un regulador de tensión lineal, pero lo vamos a ver en el siguiente apartado y este ya viene integrado en Arduino.

Fuente con un regulador lineal:



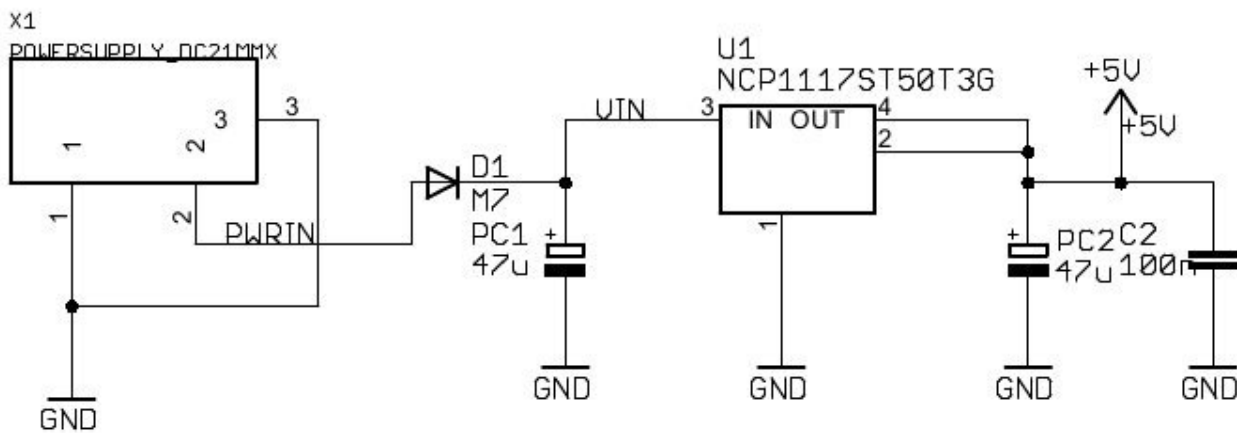
Ejemplo de alimentar un Arduino (ATmega328p) con un regulador lineal: <http://www.instructables.com/id/5V-voltage-regulator-with-7805/>

Regulador de Tensión

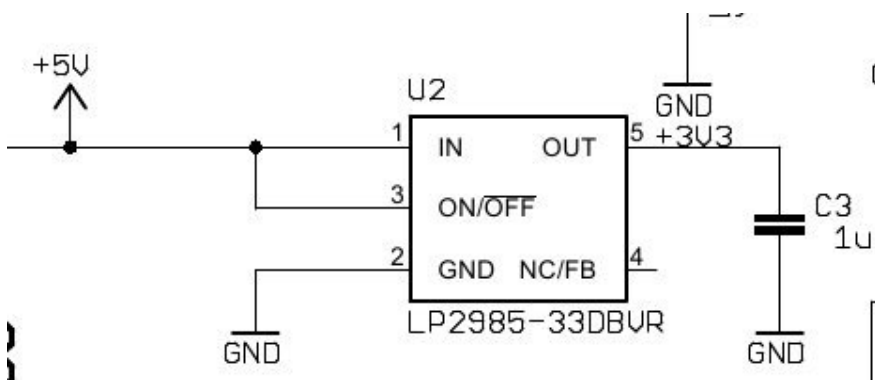
Los microcontroladores que hemos visto funcionan todos a 5V o a 3.3 V, pero como hemos visto anteriormente y según vemos en las especificaciones de Arduino UNO debemos alimentarlo al menos a 7V. Por este motivo, todos los Arduino tienen un regulador de tensión, cuya función es convertir el voltaje de alimentación al voltaje de los elementos electrónicos, principalmente al microcontrolador. Esta conversión tiene un gasto energético que se convierte en calor en el regulador de tensión.

El regulador de tensión necesita un voltaje mínimo para proporcionar 5V que está entorno 6.5-7V pero todo lo que esté por encima de ese valor se desperdicia y supone una ineficiencia energética.

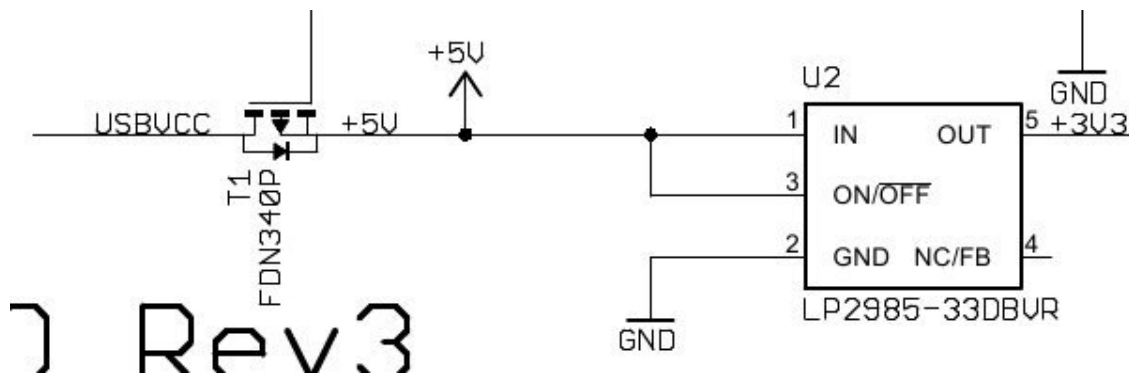
Como se puede ver en este esquema, Arduino es alimentado mediante un regulador de tensión, en este caso es un NCP1117 http://www.onsemi.com/pub_link/Collateral/NCP1117-D.PDF que alimenta al bus de 5V de Arduino.



A su vez el bus de 5V alimenta otro regulador de tensión LP2985-33 <http://www.ti.com/lit/ds/symlink/lp2985-50.pdf> del que se obtiene una salida de 3.3V



Además hay disponible una entrada al bus de 5V para la alimentación directa del cable USB.



Ver el esquema completo en:

- https://www.arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf
- <http://download.arduino.org/products/UNO/Arduino-UNO-SMD-Rev3e-SCH.pdf>

El límite máximo de alimentación de un Arduino es 20V y viene del límite de alimentación del regulador NCP1117.

Como se ha visto anteriormente los reguladores lineales de tensión son altamente ineficientes y el exceso de voltaje sobre la salida nominal se convierte en calor, por lo tanto a cuando mayor voltaje alimentemos Arduino más se calentará el regulador de tensión NCP1117.

Podemos ver cual es la tensión mínima de funcionamiento del Arduino a partir de los componentes implicados. Supongamos que pedimos a la placa una corriente de 200mA. Respecto al regulador, el dato más importante para el cálculo es la tensión de dropout, esto es, la caída entre entrada y salida cuando funciona. El dropout depende de la corriente y la temperatura. Para 200mA y 25°C, tenemos algo menos de 1V.

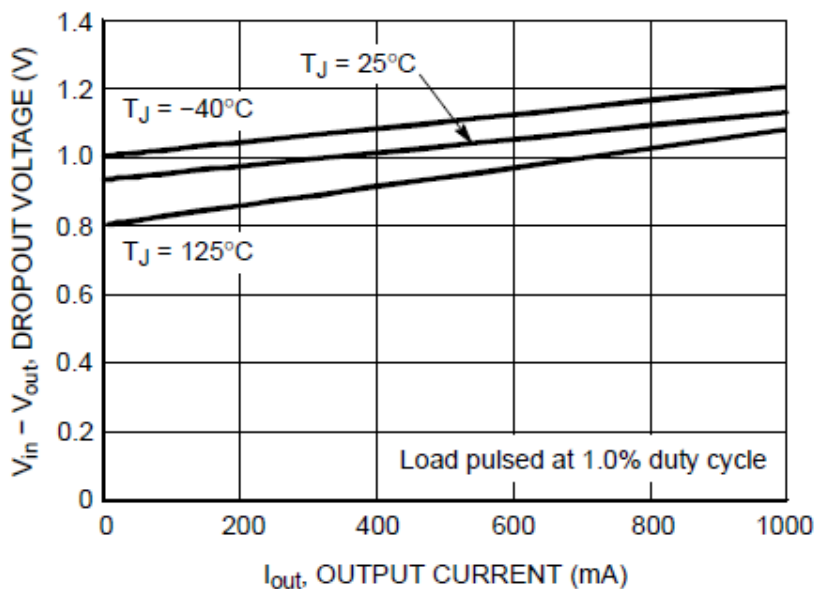


Figure 5. Dropout Voltage vs. Output Current

Ventajas y desventajas de los reguladores lineales frente a los convertidores conmutados:

<http://www.digikey.com/en/articles/techzone/2012/may/understanding-the-advantages-and-disadvantages-of-linear-regulators>. La principal ventaja de los reguladores lineales es su menor tamaño, menor complejidad de uso, bajo coste y menor ruido y ripple ([https://en.wikipedia.org/wiki/Ripple_\(electrical\)](https://en.wikipedia.org/wiki/Ripple_(electrical)))

Debido al uso de los reguladores de tensión y otras protecciones que tiene Arduino, hace que esta placa sea poco eficiente, haciendo que tenga un consumo elevado entorno a los 46mA en reposo para el Arduino UNO. Más información en <http://www.prometec.net/consumos-arduino/>

Los reguladores de tensión también se le denomina LDO (https://en.wikipedia.org/wiki/Low-dropout_regulator). Un LDO es un regulador de voltaje lineal de CC que puede regular la tensión de salida incluso cuando la tensión de alimentación está muy cerca de la tensión de salida. Las ventajas de un regulador de tensión sobre otros convertidores DC-DC conmutados es la ausencia de ruido de conmutación (se lleva a cabo conmutación), el tamaño del dispositivo más pequeño y la simplicidad diseño (por lo general consiste en una referencia, un amplificador, y un elemento de paso). Una desventaja importante es que, a diferencia de reguladores de conmutación, los reguladores de corriente continua lineal debe disipar toda la energía a través del dispositivo de regulación con el fin de regular la tensión de salida.

Un LDO, por su propia naturaleza siempre disminuye el nivel de tensión entrada y no puede usarse para que la tensión de salida sea mayor que la de entrada.

Más información:

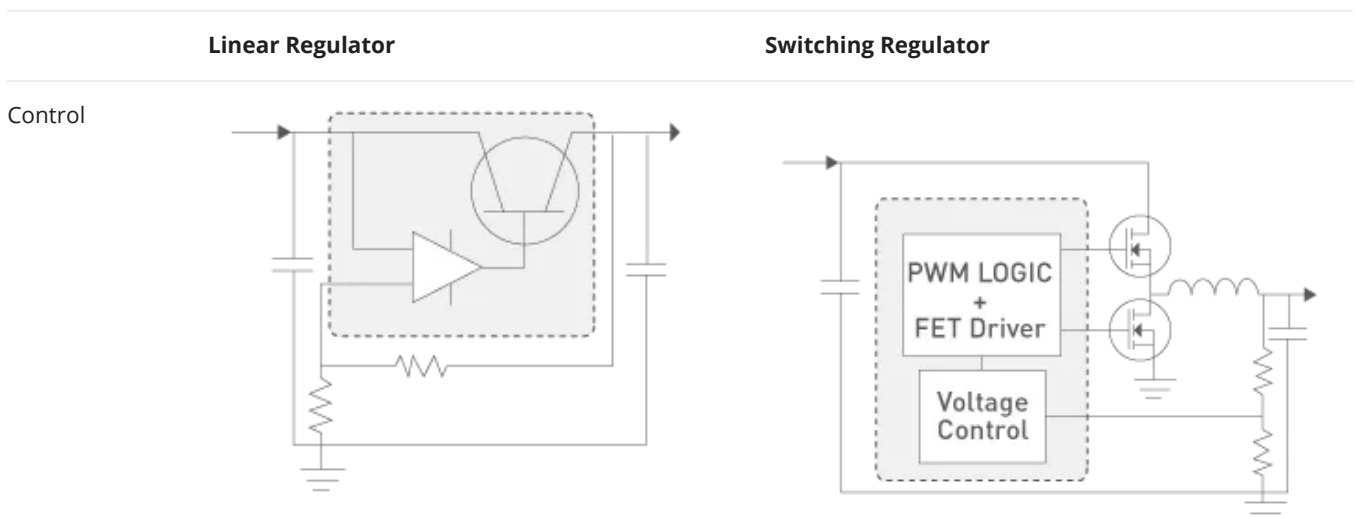
- https://en.wikipedia.org/wiki/Low-dropout_regulator
- http://www.linear.com/products/ldo_linear_regulators
- <http://www.ti.com/llds/ti/power-management/linear-regulators-ldo-overview.page>

Reguladores lineales VS fuentes conmutadas

Para decidir cómo alimentar los proyectos debes tener en cuenta un par de ideas básicas.

Es importante saber que la eficiencia típica de un regulador lineal suele ser de un 40% y puede caer hasta un 15% con facilidad, por eso nunca deberían usarse en proyectos que funcionan a baterías, frente a un 85% típico de una fuente conmutada. Las fuentes conmutadas son recomendables siempre que usemos circuitos alimentados por baterías, pero imprescindibles cuando el consumo aumenta por encima de más o menos medio amperio, porque el calor generado, y su evacuación, empezaran a dar problemas que irán complicándose cada vez más.

Una fuente conmutada DC que dé a su salida una tensión mayor que a la entrada, es una fuente del tipo Boost, Boost converter, o también una fuente Step-up y esto no es posible con una LDO.



Design Flexibility	Buck	Buck, Boost, Buck-Boost
Efficiency	Normally low to medium-high for low difference between VIN-VOUT	High
Complexity	Low	Medium to high
Size	Small to medium, larger at high power	Smaller at similar higher power (depending on the switching frequency)
Total Cost	Low	Medium to high – external components
Ripple/Noise/EMI	Low	Medium to high
VIN Range	Narrow (depending on power dissipation)	Wide

Otro regulador de voltaje muy utilizado es el LM7805:

<https://www.sparkfun.com/datasheets/Components/LM7805.pdf>:

- como funciona: <http://www.electronicshub.org/understanding-7805-ic-voltage-regulator/>
- familia de reguladores <https://es.wikipedia.org/wiki/78xx>
- <http://practica16c03.blogspot.com.es/2011/03/el-regulador-7805.html>

También podríamos hacer una versión de Arduino en protoboard con el LM7805:

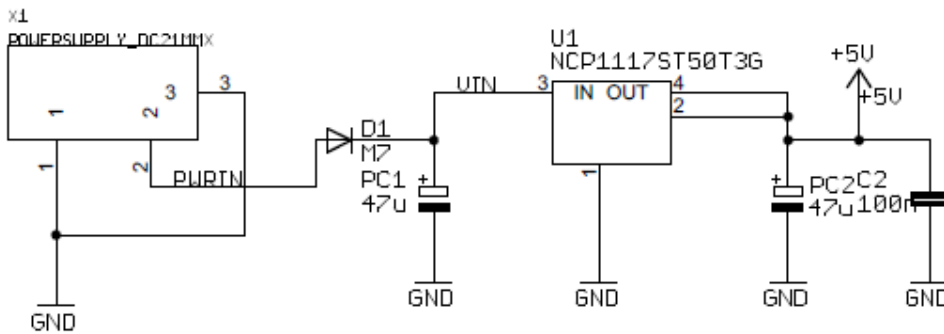
<http://www.prometec.net/arduino-en-protoboard/>

NerO es un Arduino UNO pero diseñado desde el punto de vista de la eficiencia energética,

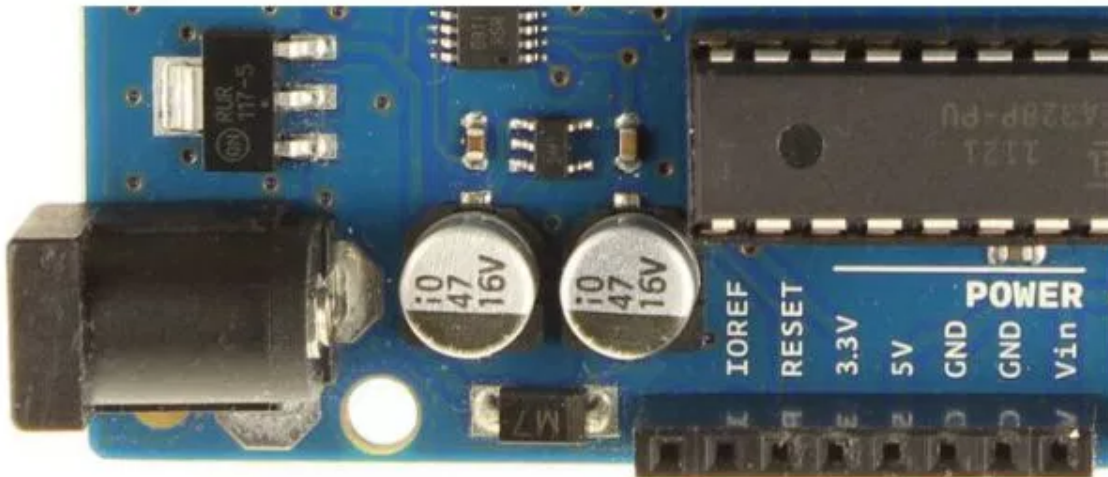
<https://www.kickstarter.com/projects/1218034479/nero-an-energy-efficient-arduino-uno-compatible-de>. NerO dispone de un regulador conmutado en lugar de un LDO, al igual que usa el Arduino Leonardo ETH el MPM3610 <http://www.monolithicpower.com/DesktopModules/DocumentManage/API/Document/getDocument?id=3301>

Cómo Funciona la Alimentación de Arduino

Este es el esquema de alimentación de un Arduino desde el jack de alimentación.

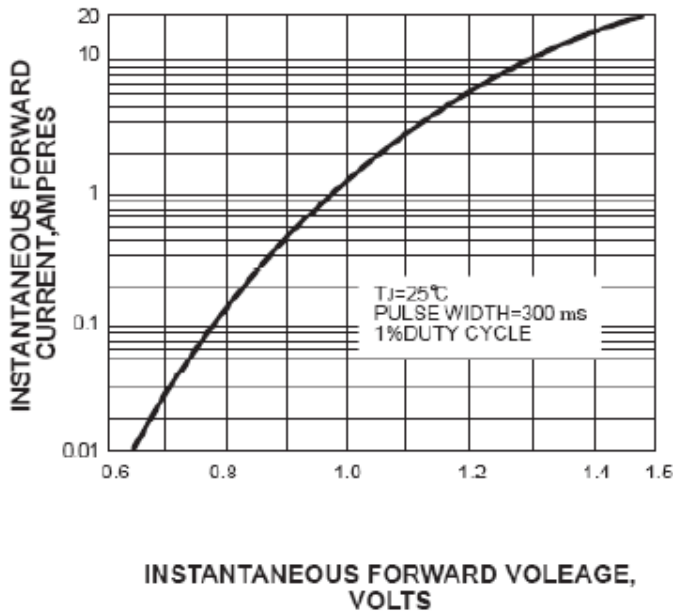


Esta es la parte del circuito que regula una salida a 5V desde el jack de alimentación.



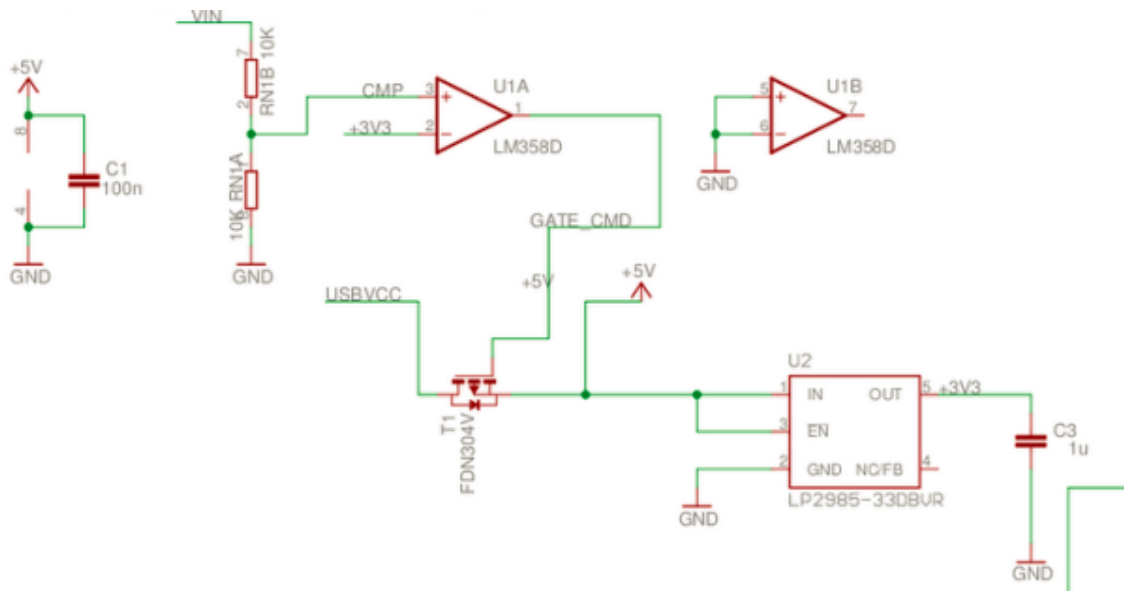
Abajo a la izquierda tenemos el conector de alimentación V_{in} . Esta entrada está también disponible al final de la tiras de pines que se halla abajo a la derecha en la figura si no se desea usar este voluminoso conector. A la derecha del conector tenemos los dos condensadores electrolíticos de 47uF y, debajo de éstos, el diodo D1 (M7). Encima del conector de alimentación tenemos el regulador de tensión NCP1117.

Para el diodo M7, el [datasheet](#) muestra la siguiente figura para la relación corriente-tensión, tenemos algo más de 0,9V a 0,2 A de de corriente. En total, juntando regulador y diodo tenemos una caída de 1,8V, por tanto, para una alimentación de 5V necesitamos un valor de V_{in} mínimo de 6,8V.

FIG. 3-TYPICAL INSTANTANEOUS FORWARD CHARACTERISTICS

La placa Arduino no funcionará fiablemente con menos de 7V, pero, dado que internamente funciona a 5V, la eficiencia máxima de la alimentación será del 71% ($=5V/7V$). Si alimentamos la placa a 9V la eficiencia cae hasta el 55%.

Adicionalmente la placa lleva otra circuitería paralela para gestionar tanto las tensiones de 3.3V como el caso en el que simultáneamente la placa esté alimentada vía jack y vía USB.



Cuando se enchufa la placa solo al cable USB, el micro y todo lo demás es alimentado por los 5V que suministra el conector USB. Si se conecta una alimentación en el rango de 7-12V al jack, el micro y todo lo demás es alimentado por el circuito de regulación.

Pero cuando se conecta la placa tanto vía USB como vía jack al mismo tiempo, la placa usa un transistor T1, para dar preferencia a la fuente de tensión regulada por el jack, anulando la suministrada vía USB. Este transistor es el que permite a la placa vigilar, en caso de que estando el Arduino alimentado de dos formas distintas, qué sistema se ha desconectado, y cambiar de uno a otro sin que notemos ninguna interrupción. En este caso si la tensión V_{in} es inferior a 6,6V se alimenta mediante el USB y si es superior, mediante el circuito de regulación.

Además como podría pasar que la alimentación a través del jack tuviera la polaridad cambiada (recordemos que para alimentar correctamente la placa el jack debe tener la tensión positiva en el centro), el sistema actúa como si la placa no estuviera conectada, protegiendo así al Arduino.

Más información:

- <https://soloarduino.blogspot.com.es/2014/12/regulacion-de-tension-explicacion.html>
- Fusibles reseteables: <https://soloarduino.blogspot.com.es/2014/12/fusibles-con-reset.html>
- <http://aim65.blogspot.com.es/2012/03/legobot-alimentando-el-arduino.html>
- <http://electronics.stackexchange.com/questions/26484/how-arduino-power-supply-works>

Esquemáticos de Arduino UNO:

- https://www.arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf
- <http://download.arduino.org/products/UNO/Arduino-UNO-SMD-Rev3e-SCH.pdf>

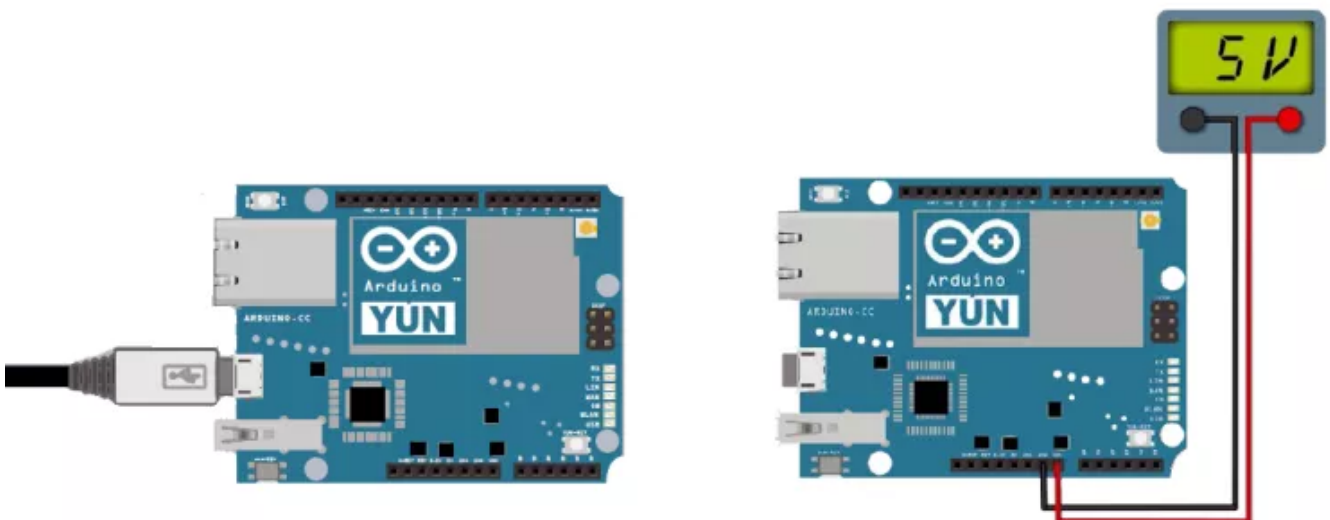
Reguladores de Tensión de Arduino

- Regulador de tensión en Arduino para bus 3,3V: http://www.onsemi.com/pub_link/Collateral/MC33269-D.PDF
- El regulador de tensión es el que marca la tensión que alimenta a la MCU:
http://www.onsemi.com/pub_link/Collateral/NCP1117-D.PDF

Alimentación Otros Arduinos

Hemos visto cómo funciona la alimentación en el Arduino UNO y en otros similares como el MEGA, pero otros Arduinos se alimentan de forma diferente y cuando nos toque usarlos habrá que ver cómo funcionan. Veamos unos ejemplos.

Para el caso del Arduino Yun al no disponer de un sistema de regulación de tensión, sólo puede alimentarse mediante un cable micro-USB o mediante el pin Vin donde debemos aplicar exactamente 5V. Arduino Yun no puede alimentarse por el pin de 5V porque en el esquema <https://www.arduino.cc/en/uploads/Main/arduino-Yun-schematic.pdf> se ve como hay un diodo que solo permite a la corriente salir.



El Arduino Yun usa el RT8010: http://www.richtek.com/assets/product_file/RT8010=RT8010A/DS8010A-10.pdf

Más información: <http://www.open-electronics.org/the-power-of-arduino-this-unknown/>

Arduino mejorado con selector de voltaje y un regulador más potente entre otras modificaciones:
<http://playground.arduino.cc/Main/Luigino328>

Más información para alimentar Arduino a 5V:

- <http://electronics.stackexchange.com/questions/65576/arduino-uno-r3-directly-supply-regulated-5v-to-5v-pin>
- <http://forum.arduino.cc/index.php?topic=18630.0>
- <http://forum.arduino.cc/index.php?topic=271158.0>

Las placas **Arduino Diecimila y Duemilanove** usan el MC33269 como regulador de tensión para conseguir los 5V necesarios para su funcionamiento. Este regulador tiene un "DropOut" de 1V, esto es, que para conseguir los 5V necesita estar alimentado con 6V, como mínimo.

Datasheet MC33269 http://www.onsemi.com/pub_link/Collateral/MC33269-D.PDF

Arduino Leonardo ETH http://download.arduino.org/products/LEONARDOETH/Arduino_Leonardo_Eth_V1_sch.pdf
usa un MPM3610 que es un Synchronous Step-Down Converter conmutado

Datasheet MPM3610:

<http://www.monolithicpower.com/DesktopModules/DocumentManage/API/Document/getDocument?id=3301>

Ejemplo: Problema de alimentación con Arduino motor shield

Veamos un ejemplo real de problema en un proyecto por culpa de no tener claro como se alimenta Arduino y los shields. Se trata de un Arduino UNO con un motor shield para mover un coche con dos motores DC de 12V.

El problema radica que al alimentar todo el bloque desde las baterías de 12V de usadas por lo motores, Arduino se queda colgado porque al arrancar los motores baja la tensión que llega al ATmega328p por debajo de 5V y deja de funcionar. Sin embargo en las pruebas con el Arduino conectado al USB para programar y hacer debug esto no pasaba.



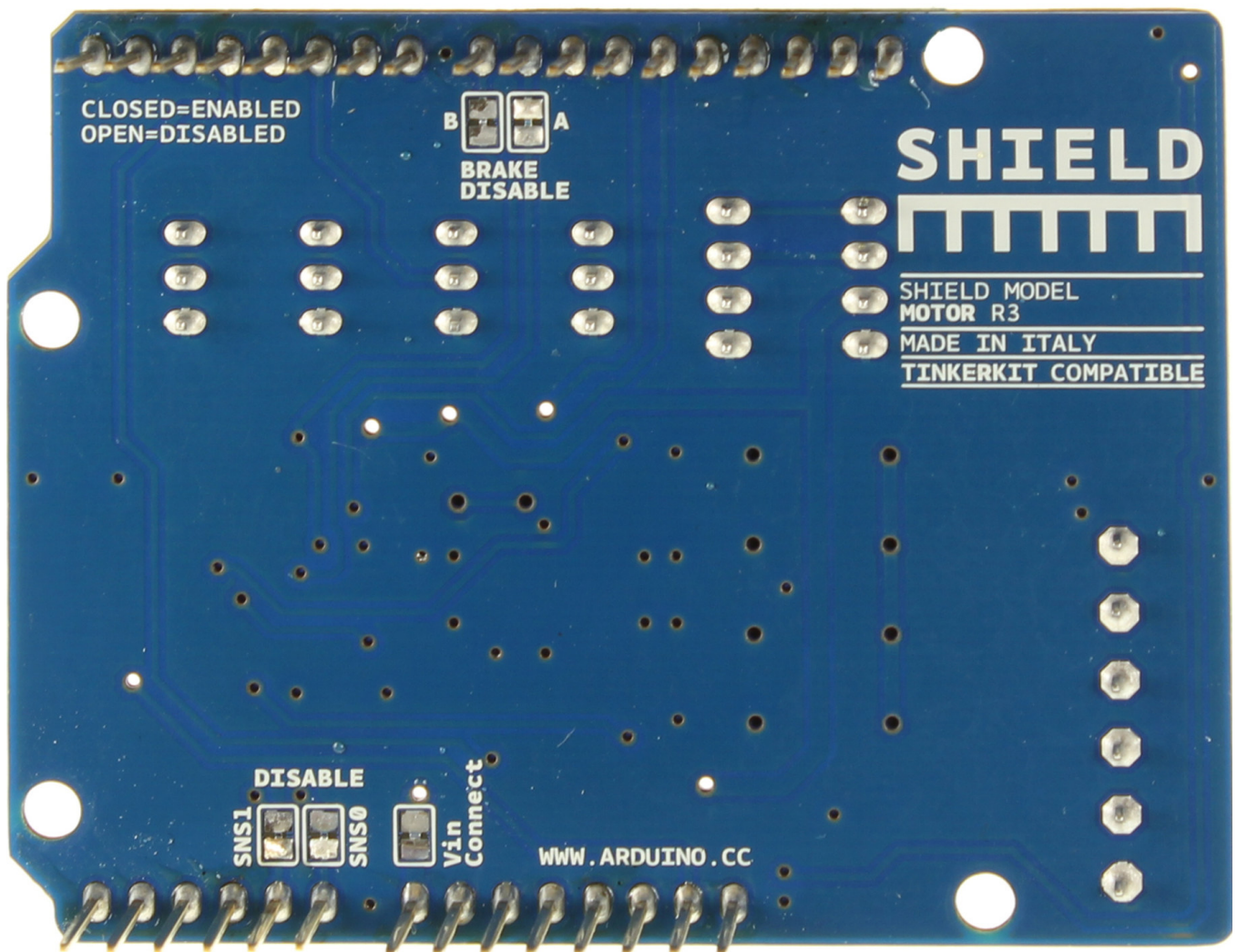
Esquema Motor Shield: https://www.arduino.cc/en/uploads/Main/arduino_MotorShield_Rev3-schematic.pdf

Esquema del Arduino UNO: <https://www.arduino.cc/en/uploads/Main/arduino-uno-schematic.pdf>

Al alimentar a 12V desde LSP5 (Borna marcada con Vin) en el motor shield, es el VMOT el que alimenta Vin del Arduino y al L298P (driver de motores). Lo que ocurre es que VMOT cae por debajo de los 6,5V necesarios para alimentar al Arduino e incluso es incapaz de alimentar al VSS (5V para la logica) L298P que lo recoge del regulador de tensión del Arduino a través del bus de 5V como se ve en el esquema.

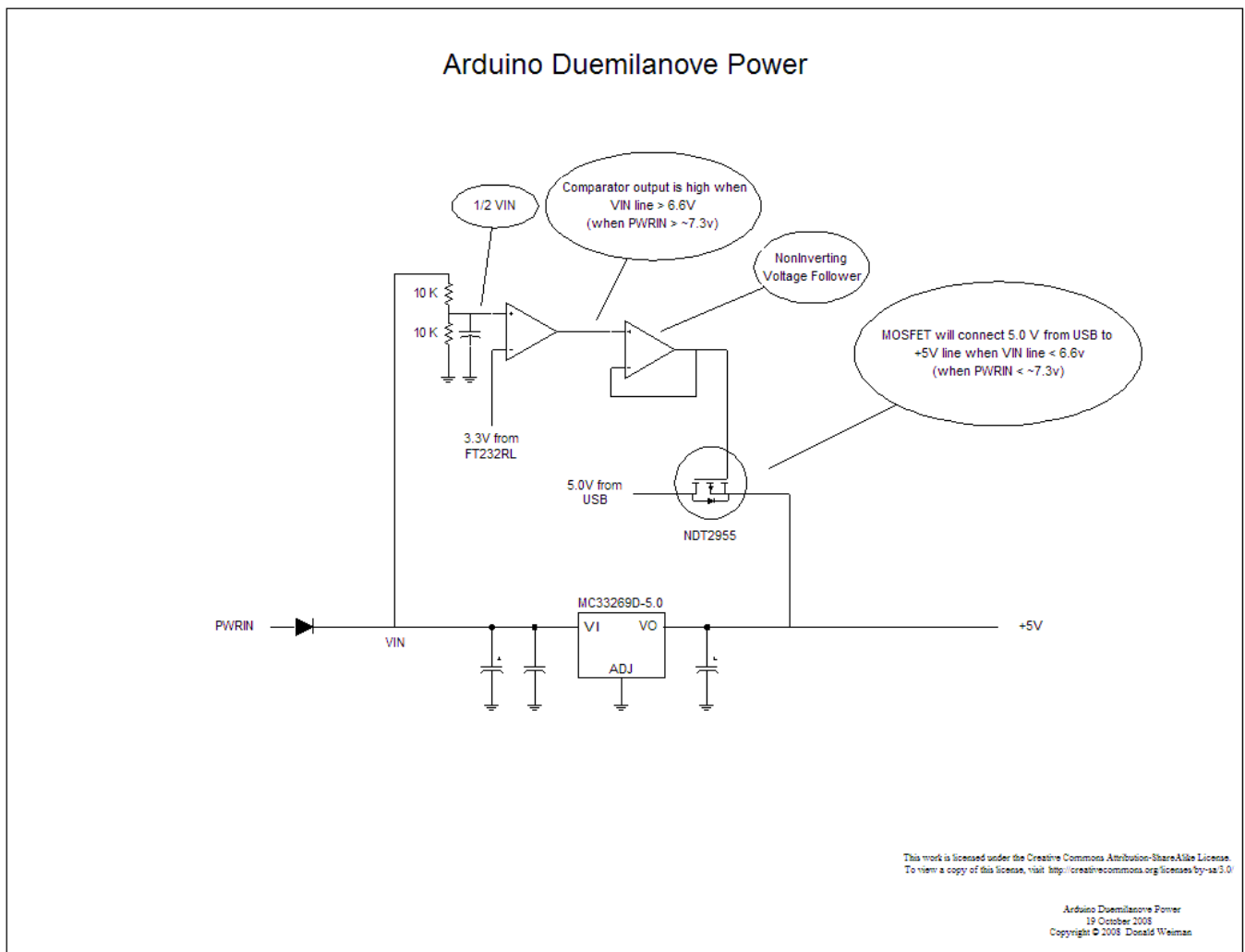
La causa de esa caída de tensión puede ser de los motores conectados al L298P están demandando mucho.

La solución es alimentar por separado los motores y el Arduino, que es precisamente lo que pasa cuando Arduino estaba conectado al USB, para ello hay que cortar el jumper en placa denominado Vin-Connect.



Cuando Arduino está alimentado por USB esto no pasa porque en ese caso te saltas la alimentación del regulador de tensión que tiene el Arduino. Es lo que se ve en el esquema de Arduino UNO como USBVCC y al ser Vin menor que 6.6V el mosfet FDN340P deja alimentar desde el USB en lugar de Vin.

Cómo decide la alimentación Arduino entre el USB o el Vin/adaptador



Eso sí, debemos tener en cuenta que al alimentar Arduino por el USB la corriente máxima será de 500 mA.

Mosfet:

- <https://es.wikipedia.org/wiki/MOSFET>
- <https://www.fairchildsemi.com/datasheets/FD/FDN340P.pdf>
- <http://es.rs-online.com/web/p/transistores-mosfet/6710435/>

Más información: <http://arduino.stackexchange.com/questions/893/arduino-what-happens-if-i-power-the-arduino-with-both-the-usb-and-external-pow>

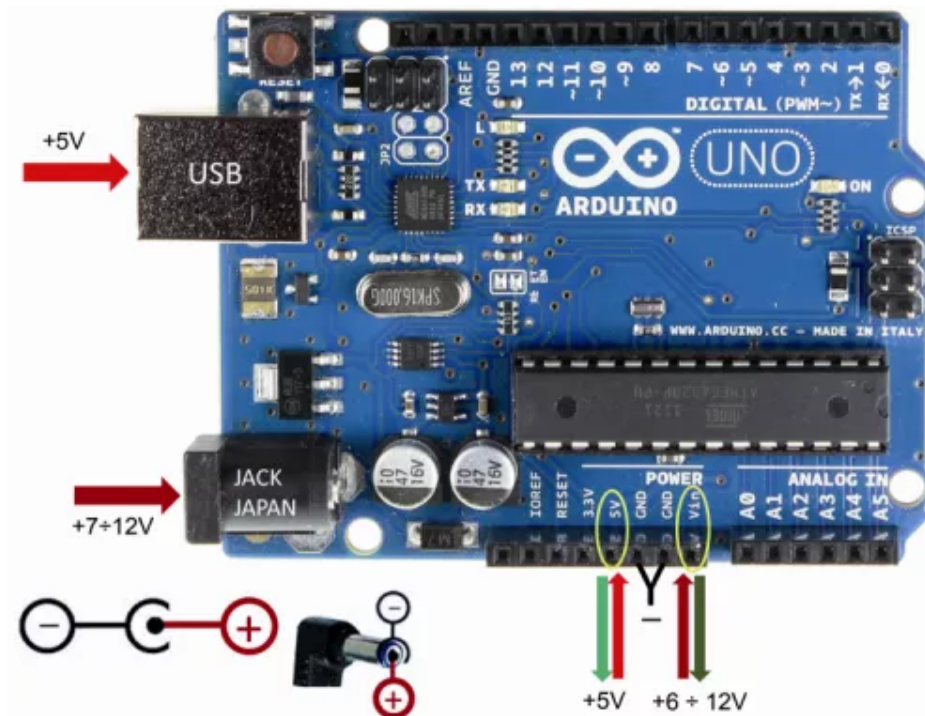
Como arduino selecciona automáticamente entre USB y Vin:

- <http://forum.arduino.cc/index.php?topic=141776.0>
- <http://forum.arduino.cc/index.php?topic=53627.0>
- <http://arduino.stackexchange.com/questions/893/arduino-what-happens-if-i-power-the-arduino-with-both-the-usb-and-external-pow>
- <http://electronics.stackexchange.com/questions/401/switch-between-5v-power-supplies>
- <https://forums.adafruit.com/viewtopic.php?f=25&t=12406>

Entradas de Alimentación de Arduino

Una vez visto cómo funciona la alimentación de Arduino, ya podemos entender las diferentes formas desde las que podemos alimentar la placa y los riesgos que tiene cada una de ellas.

Lugares desde donde se puede alimentar Arduino:



Arduino tiene 4 posibles entradas de alimentación:

- **Puerto USB:** llega al bus de 5V, procedente del puerto USB de un ordenador o de cualquier fuente de alimentación que se proporciona con un puerto USB como una batería. Si la alimentación proviene de un ordenador, hay una limitación de corriente de 500 mA, si por el contrario está utilizando una fuente de alimentación externa, la corriente máxima de salida (independientemente de la garantizada por la misma fuente de alimentación, que en general es de un máximo de 1 A o 2 A) está limitada a 500 mA por el fusible de protección auto-restaurable.
- **JACK:** una fuente de alimentación, por lo general, se debe conectar a esta toma, con el polo positivo en la parte central de la toma, y el valor debe que oscilar entre 6V y 20V, el recomendado es 7-12V, por lo que no es aconsejable utilizar tensiones inferiores a 7V o superiores a 12V.
- **Pin Vin:** este conector tiene una doble función.
 - Entrada para alimentación externa, no está protegida por el diodo ante inversiones de polaridad. La conexión va directamente a la entrada del regulador. Por supuesto no se debe aplicar tensión al jack, de lo contrario podrían surgir conflictos peligrosos
 - Salida de la que extraer el voltaje aplicado a la toma de JACK, pero sin la caída del diodo de protección. Podría resultar útil para alimentar cargas pequeñas que requieran una tensión superior a 5 V e igual a la aplicada a la toma JACK (siempre teniendo en cuenta la caída de tensión del diodo). En ambos casos el polo negativo de voltaje se puede encontrar en los pines de GND. Algunos shields usan este pin para alimentarse.
- **Pin 5V:** está conectado directamente a la salida del regulador y al bus de 5V que se usa para alimentar cargas externas a Arduino como los sensores. En el caso de no aplicar tensiones al puerto USB o al jack, la toma de 5V se puede utilizar para alimentar Arduino directamente, si es que tiene una fuente externa estabilizada de 5V. Uno tiene que tener en cuenta que, en general, los reguladores no les gustan los voltajes que se aplica a su salida, pero en este caso particular, esta situación resulta a ocurrir incluso cuando la alimentación de Arduino se hace desde el puerto USB, por lo tanto, se puede suponer que no es un gran problema. En este

caso no hay ninguna forma de protección, ya que tanto el diodo y el fusible se encuentran por encima de este bus y por lo tanto no tienen ninguna función activa. Al igual que en el caso de la toma de Vin, el polo negativo de la tensión se puede encontrar en los pines GND

NOTA: Arduino tiene un conector de salida de 3,3V para las cargas eléctricas que operan en esta tensión, mediante un segundo regulador que se conecta directamente a la V. 5 Este pin no se puede utilizar como entrada.

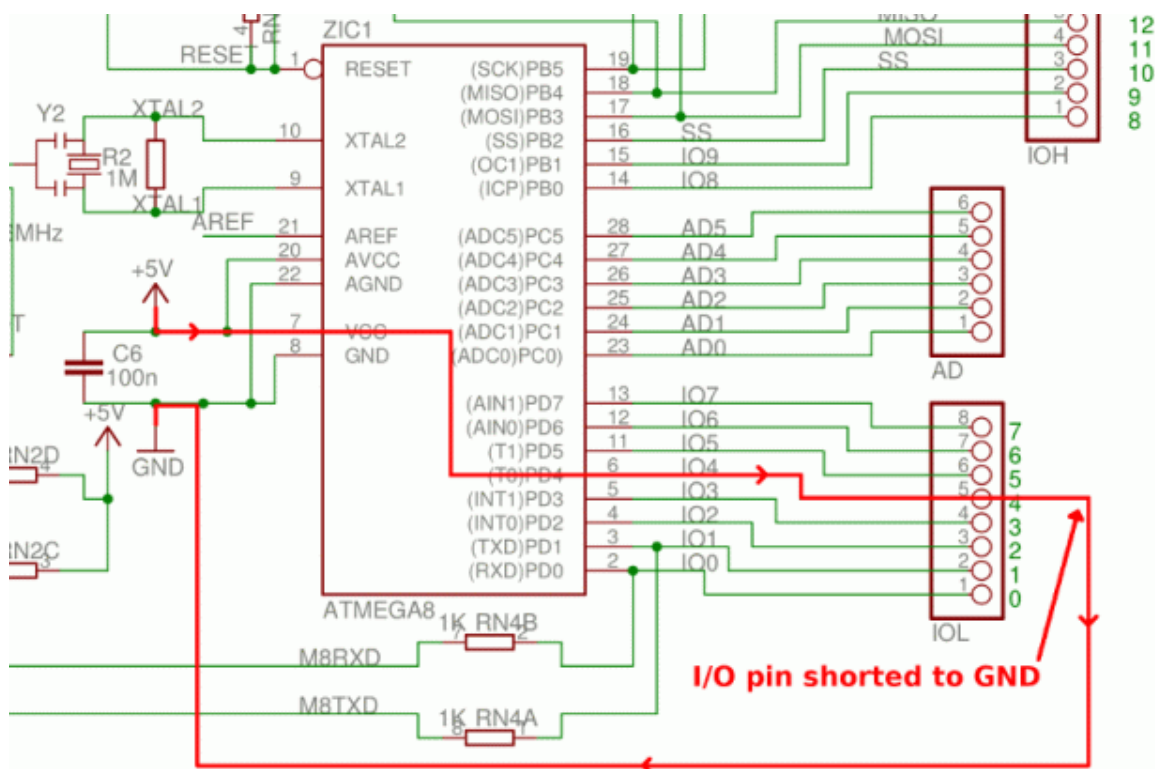
Como Destruir un Arduino

Ahora que ya conocemos bien cómo funciona la alimentación en Arduino, será fácil entender las malas prácticas que pueden dañar o destruir un Arduino.

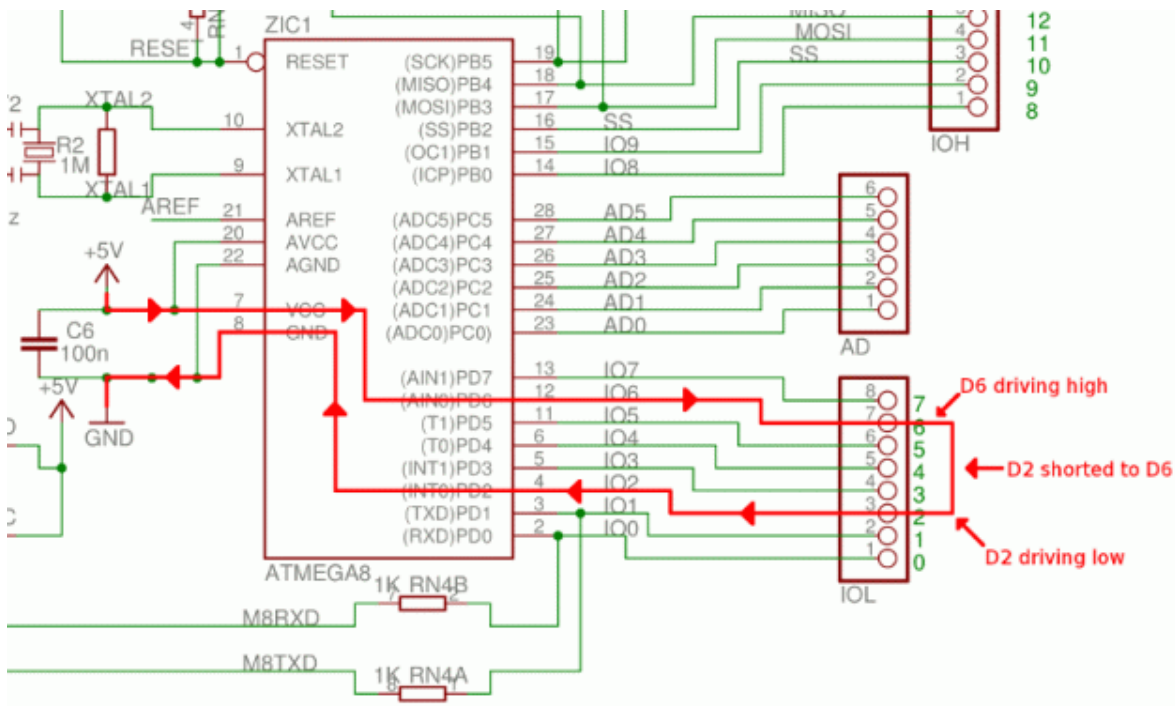
Veamos 10 formas de destruir un Arduino:

- Cortocircuitar un pin de entrada/salida con tierra. Configuramos uno de los pines como salida, lo establecemos en alto y este está directamente conectado a tierra, sin ningún componente que cree resistencia entre estos dos puntos (el pin de salida y el pin GND). El Pin de entrada/salida se sobrecarga y se destruye.

Según las especificaciones de Arduino, este puede entregar un máximo de 40mA por cada pin, pero sin nada que haga de resistencia entre el pin y tierra este entregará unos 200mA, suficiente para destruir el microcontrolador.

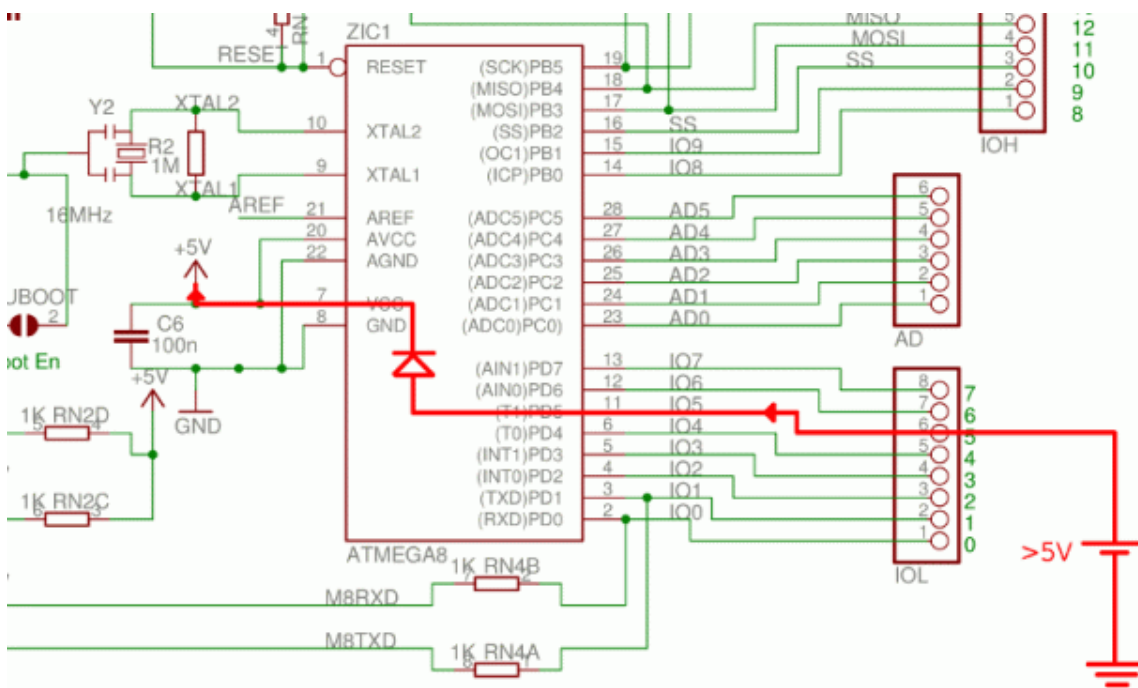


- Dos pines que se cortocircuitan mutuamente. Configuramos ambos pines como entradas, uno tiene un valor alto, el otro un valor bajo. Ambos son conectados uno con otro sin que exista ningún componente entre ellos. En este caso, ambos pines se sobrecargan mutuamente y son destruidos. Por un motivo similar al de la forma 1. El pin que está en baja en este caso haría las veces de tierra, y entregarían tensiones muy por encima del máximo del microcontrolador, lo que lo destruye.

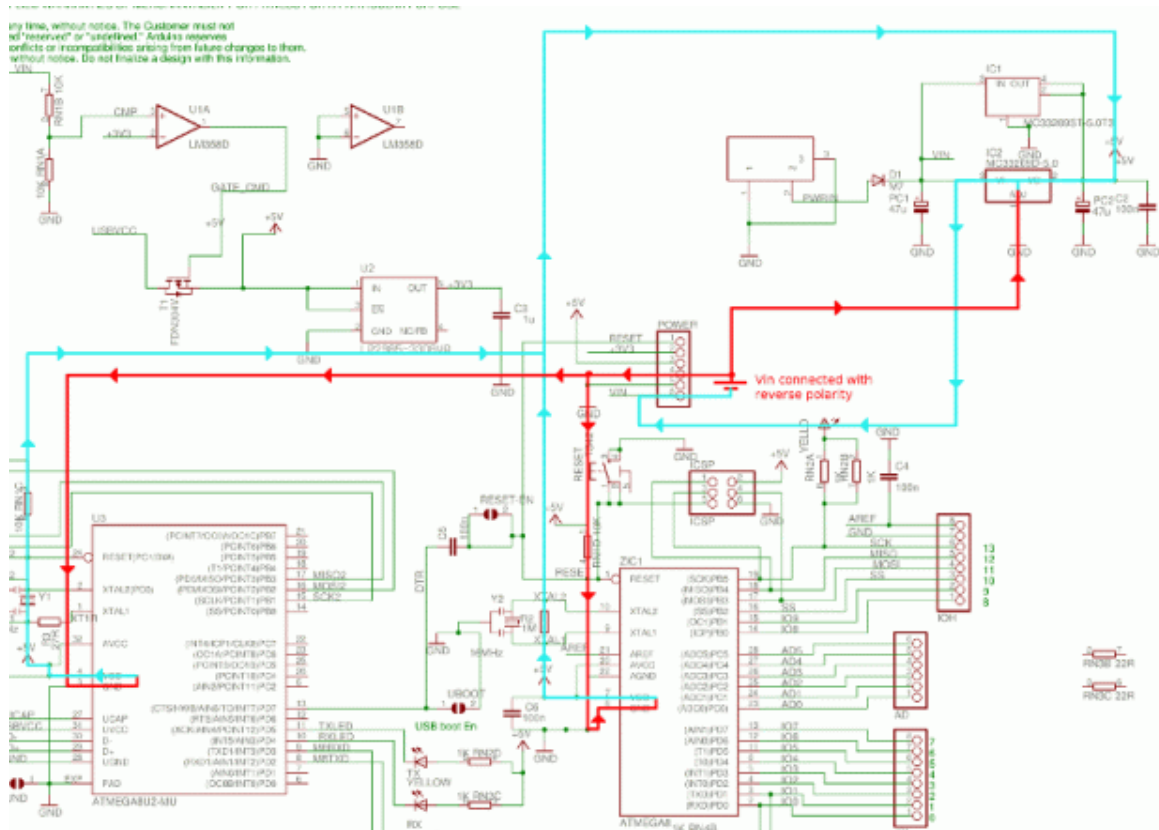


- Sobretensión en los pines de entrada/salida. El voltaje máximo recomendado para cualquier pin de entrada/salida es de 5V (Salvo las versiones que operan a 3.3V), si aplicamos cualquier tensión por encima de 5.5V destruiremos nuestro Arduino.

Los pines de Arduino cuentan con un diodo que protege el microcontrolador cuando se recibe una tensión superior a 5V, pero esta proyección esta pensada para descargas cortas, producidas por ejemplo por una descarga electrostática o un pequeño pico de tensión al arrancar, si esta sobre tensión dura, más que eso el diodo se quema, deja de proteger el pin del microcontrolador y este se quema.

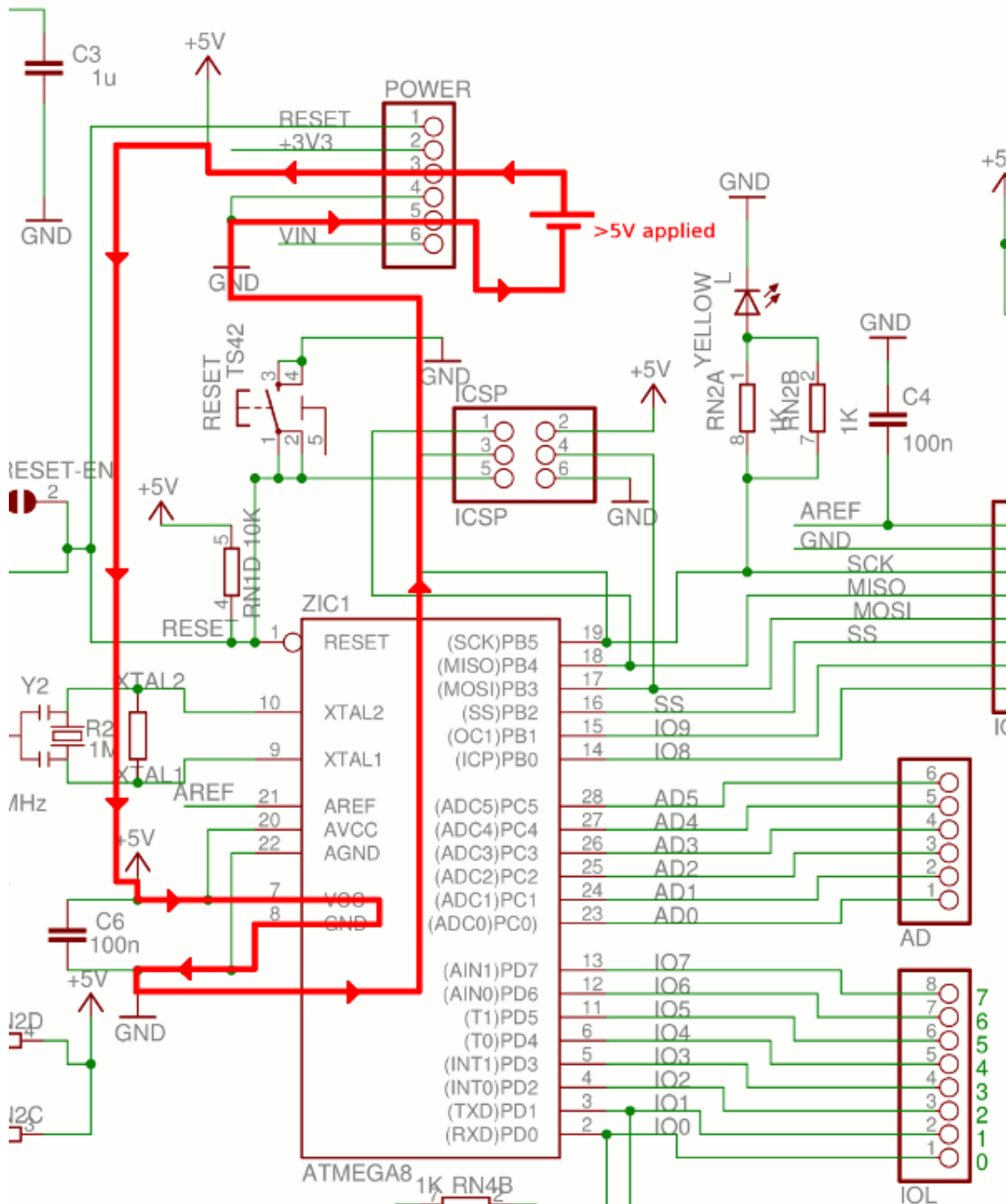


- Invertir la polaridad al alimentar Arduino por su patilla Vin. Cuando queremos alimentar nuestro Arduino mediante su patilla Vin y por error conectamos Vin a tierra y GND a 5V. Arduino no cuenta con protección en caso de invertir su polaridad de alimentación, por lo que la corriente avanzará "en dirección contraria" por todo el circuito de Arduino, destruyendo el regulador de voltaje y probablemente el microcontrolador.



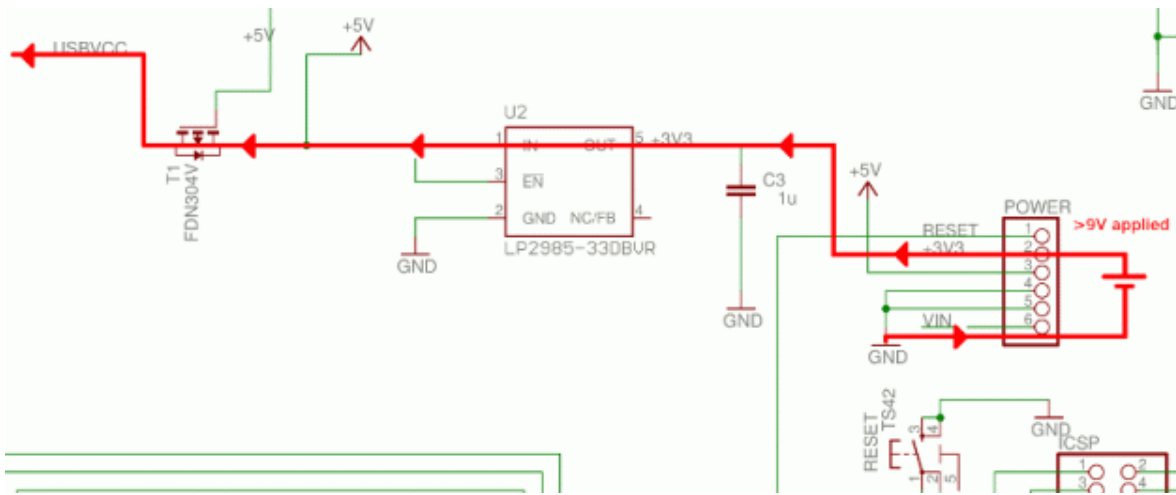
- Aplicar más de 5V al bus de 5V Al aplicar 6V o más al pin de 5V de Arduino se queman muchos de los componentes de esta placa, incluso podría llegar la corriente al puerto USB del ordenador si este está conectado.

No hay ningún tipo de protección ni regulador de voltaje en la pata 5V de Arduino, por lo que la electricidad correrá libremente por todos los componentes destruyéndolos a su paso.

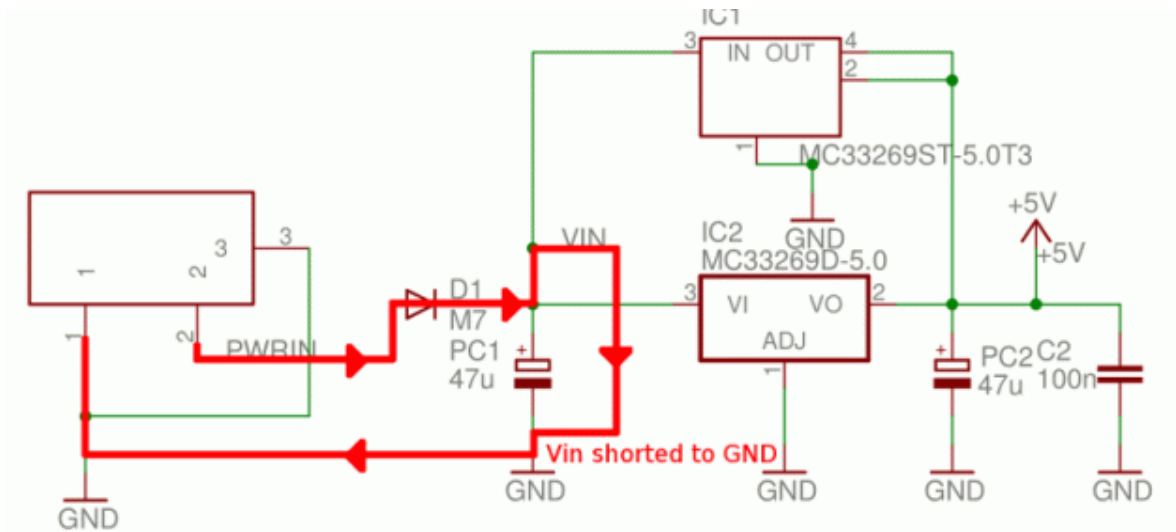


- Aplicar más de 3.3V en el pin de 3.3V de Arduino. Si aplicamos más de 3.6V en la patilla de 3.3V podría dañar varios componentes de Arduino y probablemente también cualquier Shield que este conectado en ese momento. Si llegasen a entrar más de 9V se destruiría también el regulador de 3.3V y la tensión podría causar daños en el puerto USB del ordenador si estuviese conectado en ese momento.

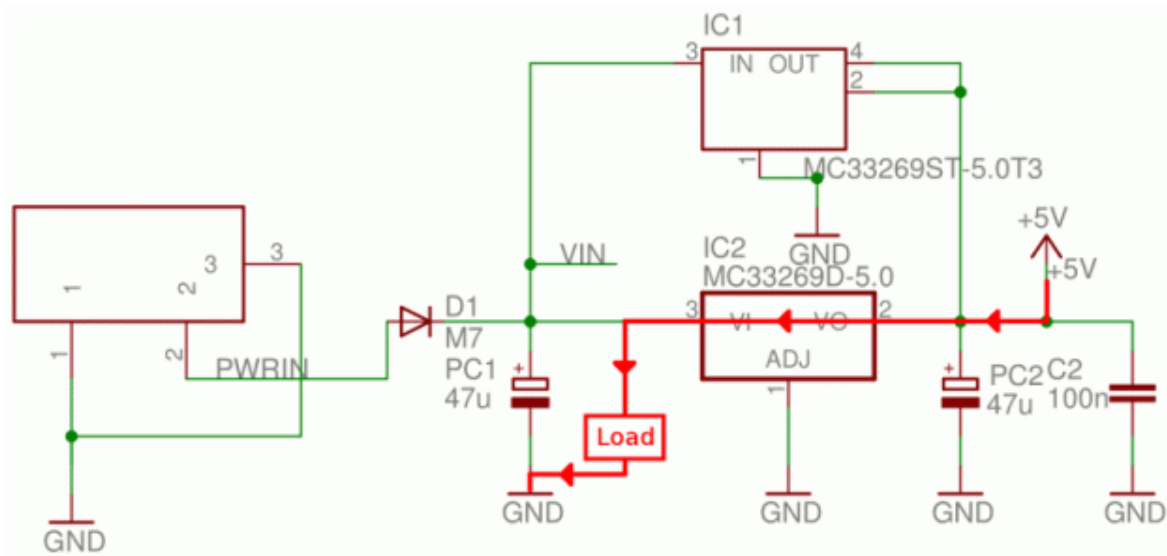
El pin de 3.3V no tiene circuitos de protección. La corriente entraría directamente al regulador de 3.3V y a cualquier componente conectado a esta tensión (Shield, módulos de bluetooth..etc). Si la tensión fuese de 9V o más el regulador sería destruido y la corriente llegaría también a los componentes alimentados normalmente por 5V.



- Cortocircuitar Vin y GND. Si conectamos la alimentación a Vin y este está cortocircuitado con GND. Si esto ocurre, el diodo de protección de Arduino sería destruido, la tensión aumentaría de forma brusca y el calor disipado sería suficiente para dañar las pistas.

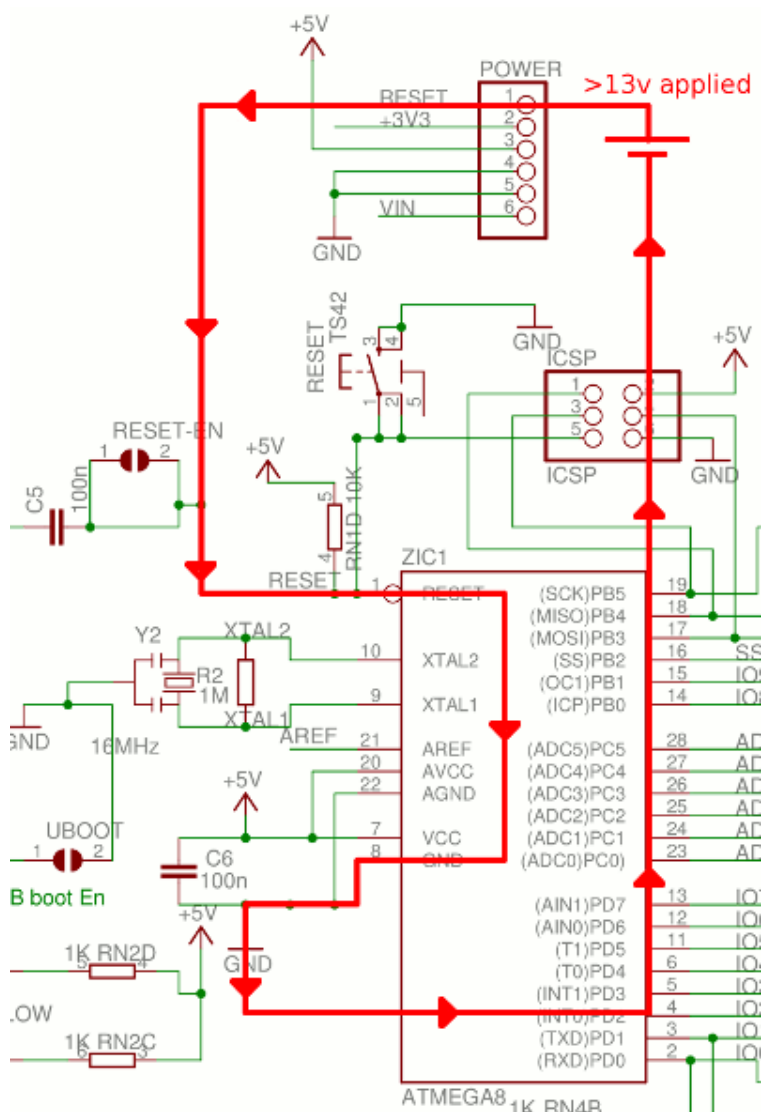


- Utilizar Vin como salida de corriente. Al alimentar la placa mediante la patilla 5V y utilizar Vin para alimentar componentes. El regulador no tiene protección para evitar que la corriente circule a la inversa. Al realizar esta conexión estamos haciendo que la corriente circule hacia atrás, lo que destruirá el regulador.



- Aplicar más de 13V en el reset. Si aplicamos más de 13 en la patilla de reset de Arduino dañaremos el microcontrolador.

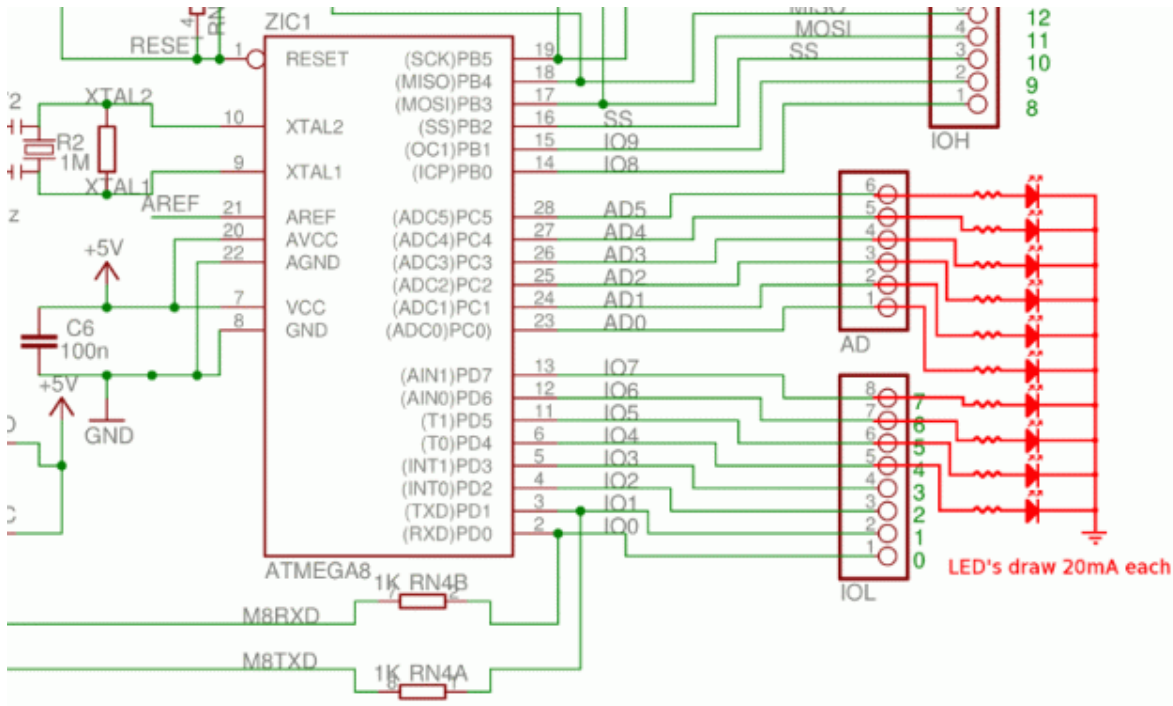
13V es el máximo que soporta la patilla de reset de microcontrolador de Arduino. Un voltaje superior daña el chip.



- Superar la tensión total que pueden entregar las salidas. Si el consumo de los pines es superior a 20mA (o la suma total de sus consumos supera los 200mA) se supera la carga máxima que es capaz de resistir el

microcontrolador y este sufre daños.

Se supera la tensión máxima que puede atravesar el microcontrolador y al no poder entregar más se daña.



Más información:

- <http://www.rugged-circuits.com/ruggeduino/>
- Artículo original <http://www.ruggedcircuits.com/10-ways-to-destroy-an-arduino/>
- Traducción: <http://www.trastejant.com/2013/07/03/10-formas-de-destruir-un-arduino/>

Clones de Arduino mejorados para ser más resistentes:

- Ruggeduino-SE "Special Edition":
 - <http://www.rugged-circuits.com/microcontroller-boards/ruggeduino-se-special-edition>
 - <http://www.rugged-circuits.com/ruggeduino-1>
- Rugged MEGA:
 - <http://www.rugged-circuits.com/microcontroller-boards/rugged-mega>
 - <http://www.rugged-circuits.com/mega-tech>

Videos de como destruir un Arduino:

- <https://www.youtube.com/watch?v=WmcMrKELkcs&feature=youtu.be>
- https://www.youtube.com/watch?v=P_fb6njcaoU

Limitaciones de Alimentación del Microcontrolador

Hemos visto cómo alimentar arduino y es la alimentación de todos los componente que la placa, incluido el microcontrolador. A la hora de hacer un proyecto, debemos tener en cuenta cómo se alimentan los elementos externos a la placa como sensores o actuadores y las limitaciones de corriente de la propia placa de Arduino y del microcontrolador.

Cuando alimentamos varios sensores o servos al bus de 5V, debemos tener en cuenta la limitación del regulador de tensión NCP1117 de 1A para saber el máximo número de sensores en función de su consumo. En el caso de un relé si lo alimentamos directamente del pin de Arduino, hay que tener en cuenta la limitación del pin.

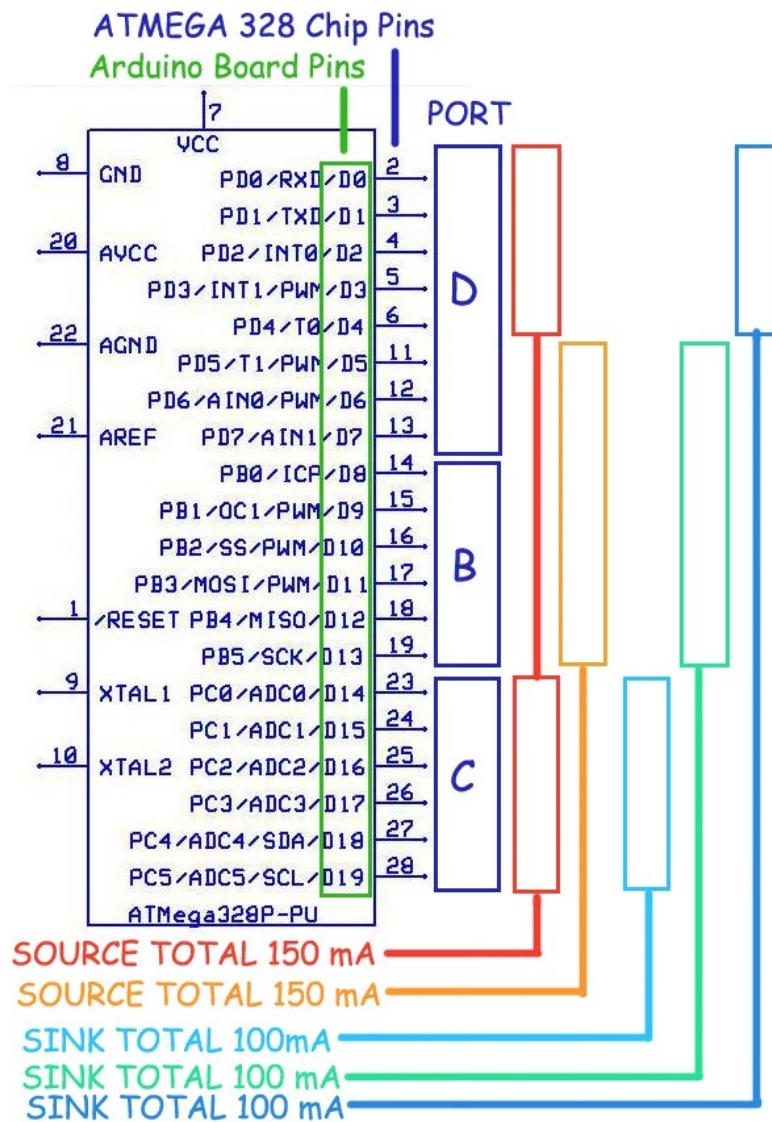
Para el microcontrolador ATmega328p los valores máximos absolutos son:

- DC Current per I/O Pin → 40.0 mA
- DC Current VCC and GND Pins → 200.0 mA

VCC current es la corriente que se puede dar (source) del pin Vcc del microcontrolador

GND current es la corriente que se puede obtener (sink) del pin GND del microcontrolador.

ARDUINO UNO / Duemilanove CURRENT SOURCE/SINK LIMITS



ABSOLUTE MAXIMUM PER PIN 40 mA - RECOMMENDED 20 mA

Aunque cada puerto I/O puede dar más corriente que en las condiciones normales (20 mA en VCC = 5 V, 10 mA en VCC = 3V) bajo condiciones de estado estable (no transitoria), hay que tener en cuenta lo siguiente.

- La suma de toda la corriente en HIGH (source) para los puertos C0 – C5, D0 – D4, ADC7, RESET no debe exceder los 150 mA
- La suma de toda la corriente en HIGH (source) para los puertos B0 – B5, D5 – D7, ADC6, XTAL1, XTAL2 no debe exceder los 150 mA.
- Si la corriente en HIGH superar los valores nominales, el voltaje en HIGH puede superar los valores nominales. No se garantiza que los pines puedan dar (source) más corriente que la de los valores de test.
- La suma de toda la corriente en LOW (sink) para los puertos C0 – C5, ADC7, ADC6 no debe exceder los 100 mA
- La suma de toda la corriente en LOW (sink) para los puertos B0 – B5, D5 – D7, XTAL1, XTAL2 no debe exceder los 100 mA
- La suma de toda la corriente en LOW (sink) para los puertos D0 – D4, RESET no debe exceder los 100 mA
- Si la corriente en LOW supera los valores de test, el voltaje de LOW puede exceder los valores nominales. No se garantiza que los pines puedan obtener (sink) más corriente que la de los valores de test.

Más información:

- <http://playground.arduino.cc/Main/ArduinoPinCurrentLimitations>
- <http://arduino-info.wikispaces.com/ArduinoPinCurrent>

Limitaciones de potencia, corriente y voltaje: <http://www.electricrcaircraftguy.com/2014/02/arduino-power-current-and-voltage.html>

Alimentar Arduino con Baterías

La capacidad de las baterías se mide en mAh o Ah. Un miliamperio hora es el término que se utiliza para determinar la duración de una batería. Si una batería o pila tienen 1000 mAh, podrás alimentar algo que consuma 1000 mA durante una hora (o 100 mA durante 10 horas). En realidad esto solo se cumple en la teoría. En la práctica, cuanto más rápido se descargue la batería, más potencia se estará disipando en la resistencia interna que tiene. Eso quiere decir que si realmente durase 10 horas alimentando un dispositivo que consume 100 mA, probablemente no duraría una hora alimentando algo con un consumo de 1000 mA.

¿Cuanto consume Arduino? Un pequeño circuito con una placa de Arduino y un Display ya suponen un consumo por encima de los 200 mAh, mientras que una pila recargable de 9V tiene entorno a los 300 mAh apenas podremos alimentar Arduino durante poco más de una hora. Alimentar Arduino con una pila de 9V es una mala opción.

Actualmente las baterías más usadas para alimentar Arduino son las LiPo, son las que se utilizan en radiocontrol y similares. Pero también pueden usarse otras como las AA, batería de coche, o las USB para los teléfonos.

Además de la capacidad, estas baterías LiPo vienen con lo que llaman la tasa de descarga, es decir lo rápido que se puede descargar la batería. Esto se muestra con la letra C. De manera que si tienes 10C y 5000mAh puede dar 50000 mA en una hora. Claro que no va a llegar pero se pueden utilizar 50A durante el tiempo que te dure la batería, que vendría a ser la décima parte de 1 hora. 60 minutos entre 10 son 6 minutos. A mayor C más cara va a resultar la batería, tal vez solamente necesites 2A para algún momento puntual así que seguramente sea necesario tanta tasa de descarga.

El único problema a estas LiPo es la carga, hay que ser cuidadosos porque tienen sus propios cargadores, si te pasas de carga se queman. No debe de pasar nada pero si no se pone el voltaje correcto en el cargador puede que se queme.

Cargador de batería LiPo: <https://www.adafruit.com/products/1904>

Más información sobre baterías LiPo: <http://www.educachip.com/cargar-baterias-lipo/>

A la hora de alimentar un Arduino con batería además de conocer los diferentes tipos de baterías, debemos conocer las técnicas a nivel de programación y electrónica para que Arduino consuma menor y pueda tener una larga vida con batería.

Para conseguir más autonomía en un Arduino alimentado con batería:

- **Apagando el circuito.** Seguramente no necesites que tu circuito esté todo el tiempo encendido. Si estás tomando datos de temperatura seguramente con tomar uno a la hora o a las dos horas te bastará. Si se trata de una alarma casera necesitarás encender el sistema una vez cada minuto o cada dos minutos para saber si hay alguien. Esto en Arduino se llama **modo Sleep** y lo que hace es hibernar y se queda en estado de baja energía hasta que lo enciendes. Este modo se verá más adelante.
- **Utilizando placas solares.** Las hay para Arduino y las hay pequeñas que puedes añadir a tu proyecto. La electrónica consume poco así que esto puede ser una carga para tu batería y un extra para utilizar menos energía.
- **Elegir una placa que consuma poco.** El Arduino UNO consume más que el Nano. Es normal, el UNO por ejemplo como acepta entre 7V y 12V, perdiendo energía en forma de calor. Y luego al ser más grande necesita más carga para alimentar a todos los elementos. Luego el Nano es tan poca cosa que consume muy poco, unos 15mA frente a los 46mA del UNO. Aún así, una placa que funcione a 3.3V aún consumirá menos. Hay varias así, de hecho ahora todas van en esa dirección ya que el consumo es algo que cada vez importa más.
- **Crear un código muy pulido.** Ser fino programando tiene su recompensa ya que a más líneas más consumo tenemos. Esto pasa también con el tiempo. A más líneas más tiempo se pierde y eso puede suponer más consumo. En este enlace <https://foro.hacklabalmeria.net/t/arduino-bajo-consumo/4230> e dedicaron a medir los mA utilizados en los sketches de Arduino y encontró una librería llamada JeeLib que lo que hace es reducir la corriente utilizada. Eso implica aumentar el número de horas de la batería.

Más información:

- <http://rufianenlared.com/bateria-arduino/>
- Consumos Arduino: <http://www.prometec.net/consumos-arduino/>

En realidad los Arduinos oficiales, son poco eficientes, pero disponemos de otros clones de Arduino fabricados para tener bajo consumo como los moteino: <https://lowpowerlab.com/guide/moteino/> que además incluyen comunicación inalámbrica y está basado en el ATmega328p.

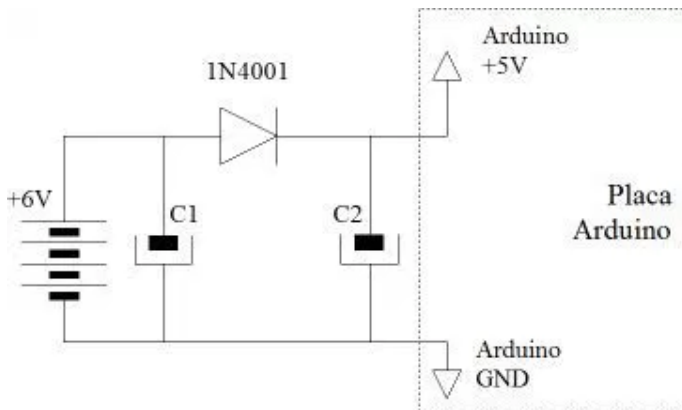
También es posible modificar un Arduino Mini para alimentarlo con batería durante más de un año, combinado con la librería low power: <http://www.home-automation-community.com/arduino-low-power-how-to-run-atmega328p-for-a-year-on-coin-cell-battery/>

Más enlaces con proyectos de Arduinos bajo consumo:

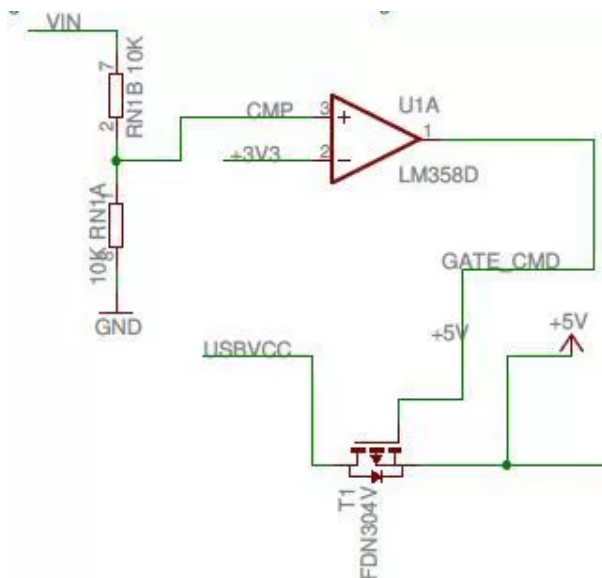
- <https://openhomeautomation.net/arduino-battery/>
- <https://github.com/petervojtek/diy/wiki/Arduino-with-Very-Low-Power-Consumption>
- <http://www.instructables.com/id/Arduino-low-Project-and-code/>

Otra opción interesante para alimentar directamente el microcontrolador de Arduino sin la placa con una batería, de forma que sea mucho más eficiente, es alimentar directamente mediante el bus de 5V de la placa. La propuesta

consiste en conectar el pack de 4 pilas a la placa Arduino usando un diodo en 1N4001 en serie. La caída de tensión en el diodo será de unos 0,7V por lo que la tensión que se verá en la línea interna de 5V de la placa Arduino será 5.3V. Dado que el micro [ATmega328](#) que es el corazón de Arduino UNO puede ser alimentado hasta con tensiones de 5.5V, nos hallamos dentro del margen correcto de funcionamiento. Los dos condensadores son para filtrar un poco el ruido y el valor dependerá del uso que se dé a la placa.



La realización de la solución anterior tiene un problema potencial si conectamos la placa Arduino al PC a través de un puerto USB cuando aún está alimentada por el pack de pilas.



Si conectamos las pilas con diodo, que da una tensión de 5.3V a la línea +5V de la placa. Dado que no hemos conectado nada en Vin, el transistor conducirá, por lo que al conectar la placa al puerto USB tendremos un conflicto entre los 5,3V de nuestras pilas y los 5V del puerto USB. En el mejor de los casos se dará una corriente en el sentido de las pilas hacia el PC que, cuando alcance 500mA, hará saltar el fusible que lleva la placa.

El uso de este tipo de alimentación requiere de desconectar el pack de pilas cuando se desee programar la placa Arduino usando el puerto USB.

Más información: <http://aim65.blogspot.com.es/2012/03/legobot-alimentando-el-arduino.html>

En este enlace lo explica muy bien la alimentación de Arduino para sistemas autónomos, resumiendo todo lo visto: <http://www.homautomation.org/2014/04/03/best-ways-to-power-a-arduino-according-to-your-need/>

Alimentación Arduino por energía solar:

- <http://tienda.bricogeek.com/shields-arduino/520-arduino-solar-charger-shield.html>
- http://wiki.seeedstudio.com/wiki/Solar_Charger_Shield_V2.2

Niveles de tensión 5V y 3.3V

Uno de los problemas principales que nos podemos encontrar cuando estás realizando un proyecto es que tus dispositivos no trabajan con el mismo voltaje. Un caso especialmente habitual como usuario de plataformas del tipo Arduino es que te encuentras con un dispositivo que trabaja a 3.3V teniendo una placa que trabaja a 5V, en cuyo caso te toca pasar de 5V a 3.3V.

Aunque en algunas ocasiones puedes conectar ambas cosas sin que nada se rompa no suele ser buena idea conectar los dispositivos directamente. Lo mejor que puedes hacer es utilizar algún tipo de interfaz o elemento que te permita realizar la conexión de forma segura.

5 formas de pasar de 5V a 3.3V para elegir la que mejor se adapte a tu proyecto. Son las siguientes:

- Utilizando un divisor de tensión.
- Añadiendo diodos 1N4148 (o similar) en serie.
- Mediante un buffer CD4050 (o similar) alimentado a 3.3V.
- Usando un regulador de voltaje variable LM317 (o similar).
- Con un regulador de tensión LM7833 de 3.3V (o similar).

Más información: <http://www.educachip.com/pasar-de-5v-a-3-3v/>

También nos podemos encontrar con el caso contrario, cada vez es más habitual que los Arduinos tengan entradas y salidas a 3,3V en lugar de a 5V, pero muchos elementos que conectados a Arduino funcionan con lógica de 5V.

Más información: <https://findeprehistoria.wordpress.com/2010/07/29/interfaz-de-3-3v-a-5v-y-visceversa/>

Y también tenemos los conversores de niveles lógicos como <http://tienda.bricogeek.com/herramientas-de-prototipado/82-conversor-de-niveles-logicos-3-3-5v.html>

Anuncios

Esta entrada se publicó en Arduino, Avanzado, Electronica, Hardware y está etiquetada con Arduino, Avanzado, Electronica, Hardware en 9 noviembre, 2016

[<https://aprendiendoarduino.wordpress.com/2016/11/09/alimentacion-arduino/>].

5 pensamientos en “Alimentación Arduino”



Maria G

3 julio, 2017 en 11:08 pm

Hola, que tal. Muy buena toda la informacion, yo quiero alimentar a mi arduino uno con una lipo de 7,4 v y del pin de 5 v alimentar otros componentes, tendria algun riesgo?



jecrespom

Autor de la entrada

3 julio, 2017 en 11:14 pm

No hay ningún problema si lo alimentas desde el jack de alimentación que admite de 7 a 12V. Además en caso que te equivoques con la polaridad de la alimentación el Arduino UNO está protegido por el diodo M7. El positivo va en el centro del jack y GND en el exterior.

Si tratas de alimentar el Arduino UNO a través del pin Vin, puedes hacerlo pero si te equivocas en la polaridad puedes dañarlo ya que desde este pin no está protegido por el diodo M7.



Ivan Ferrer

16 julio, 2017 en 5:31 pm

madre mía, esto es ¡La Biblia de la alimentación de Arduino! Maná de los cielos para inexpertos como yo! Además muy bien redactado y sin faltas (que en una jerga tan precisa a veces llevan a confusión). MIL gracias por el trabajazo!!

He llegado aquí buscando si se podría alimentar un Arduino usando el transformador que ya usa otro aparato. Digamos que tengo un teclado MIDI (alimentado con transformador AC/DC) y quiero añadirle un Arduino (u otro modelo) para ‘tunearlo’ y añadir algunos mensajes MIDI personalizados. Pensaba que podría alimentarlo directamente del mismo jack de corriente del teclado.

Imagino que tendría que medir el consumo de corriente de ambos (teclado y proyecto) y ver si el transformador puede con los dos. O en caso negativo usar uno de más mA.

En caso de poder, ¿como sería mejor ponerlo, en paralelo o en serie?

Pensaba en crear un chisme con el Arduino dentro y una entrada y salida de jack para conectarlo entre el teclado y el jack original.

Como digo, soy principiante total, si lo crees conveniente mándame al carajo con mi pregunta (a otro foro) 😊

Saludos y gracias de nuevo.