

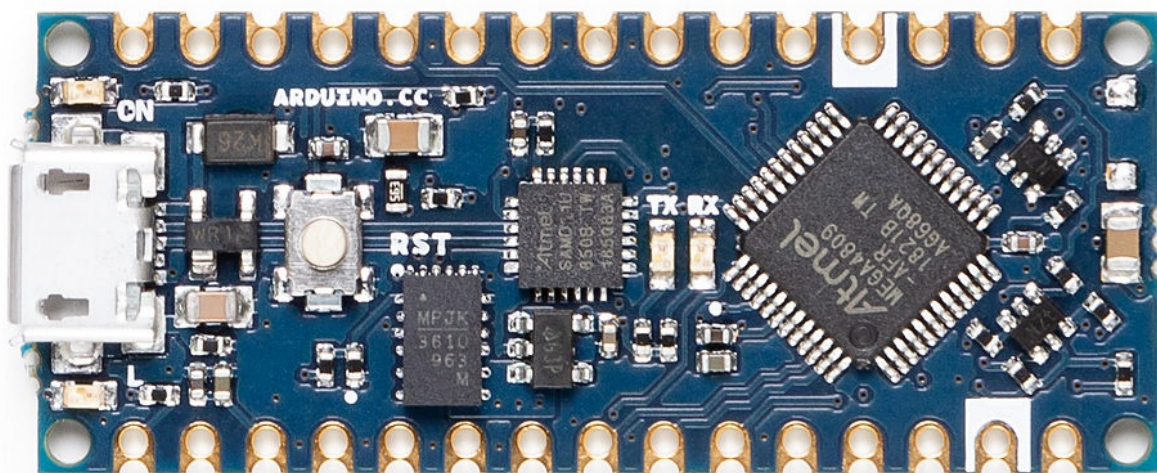
Resumen de la Sesión 1 del Taller de iniciación al diseño de PCBs de MakerAlc.

Objetivos de la sesión: conocernos, hablar sobre el desarrollo del taller, ver el esquema electrónico de una placa de Arduino explicando de una forma **sencilla** como funcionan las distintas partes del esquema, para poder ir introduciendo conceptos de electrónica.

El [proyecto Arduino](#) comenzó en el año 2005, consiste en unas herramientas hardware y software que permiten a una persona con conocimientos básicos de electrónica y de programación crear dispositivos electrónicos inteligentes, dispositivos que se programan para realizar la tarea deseada, siendo estos dispositivos por ejemplo capaces de leer sensores y mandar la información, controlar motores, escribir en una pantalla LCD, encender diodos LEDs, y un largo etc..

Por lo que Arduino se ha convertido en una de las herramientas principales con las que comienzan en el mundo de la electrónica las personas que quieren aprender. Como también es una de las herramientas principales para las personas que solo quieren crear de forma rápida un dispositivo electrónico que realice determinada función, adquiriendo en el proceso unos conocimientos mínimos de electrónica y abstrayéndose del funcionamiento de esta.

Para esta sesión se va a poner como ejemplo el esquema de la placa [Arduino Nano Every](#), que podemos ver en la siguiente imagen:

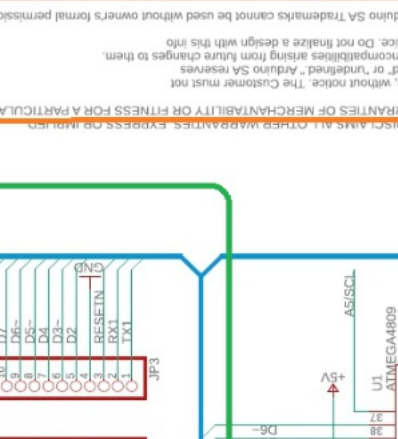
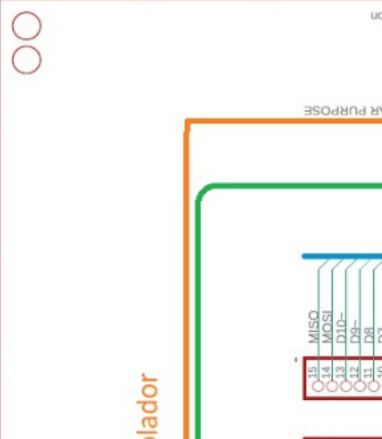
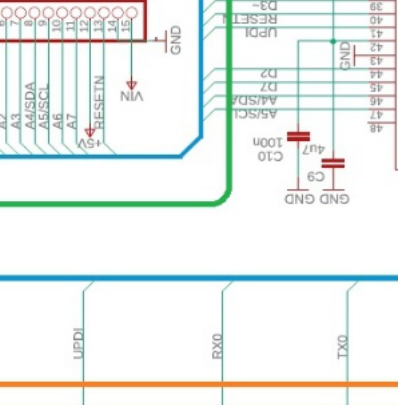
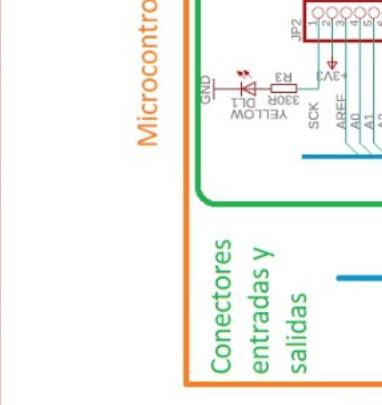
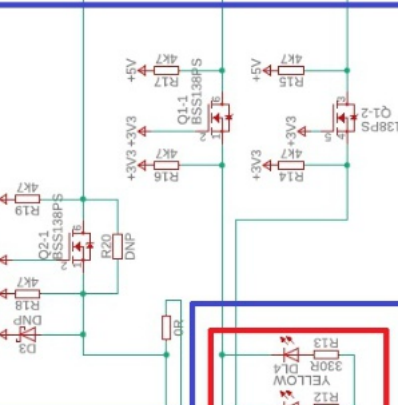
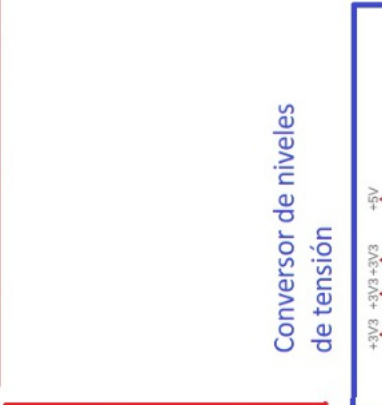
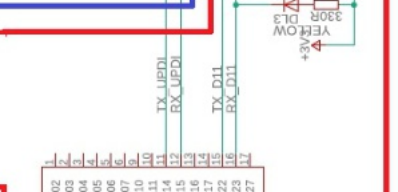
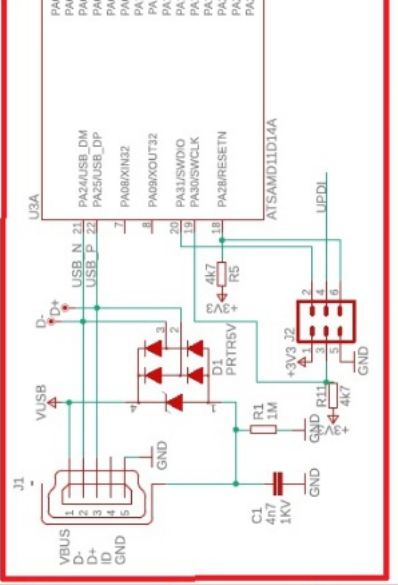
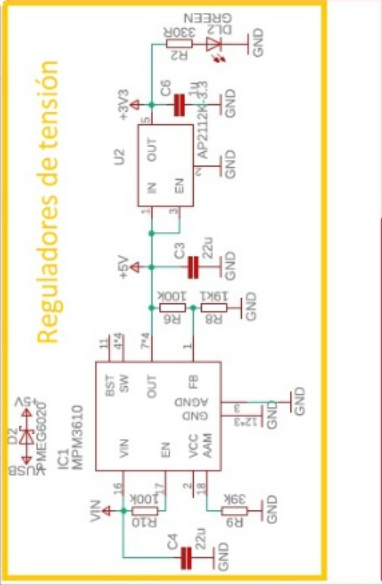


Por lo general los componentes principales de una placa de Arduino son: el microcontrolador que programamos para que a través de sus pines de entrada salida se comuniquen con otros componentes electrónicos y crear un dispositivo electrónico que realice la tarea deseada.

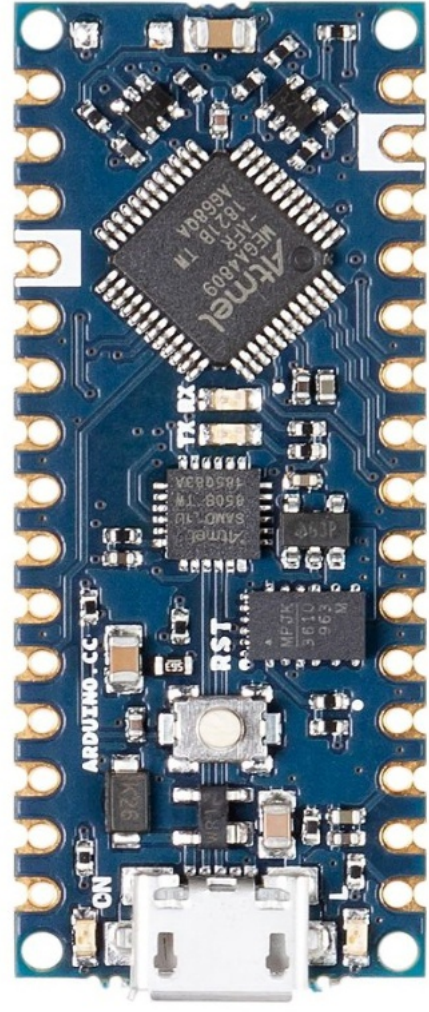
Un conector de programación o un puerto USB, que permite a la placa Arduino comunicarse con un ordenador (pc) para recibir el programa que tiene que ejecutar el microcontrolador, o para intercambiar información con el pc una vez programado.

Por último otro de los componentes principales es el regulador de tensión, el regulador de tensión convierte el voltaje de una batería o fuente de alimentación a un voltaje constante de 3.3 o 5 voltios, voltaje requerido por el microcontrolador y resto de componentes electrónicos para su correcto funcionamiento.

En la siguiente página podemos ver el esquema de la placa Arduino Nano Every, del que vamos a explicar el objeto de las distintas partes que lo componen.



Convertidor de protocolos USB a serie



CC BY SA

ARDUINO

Arturo Gundaupli

NANOeveryJ3.0

17/07/2019 11:09

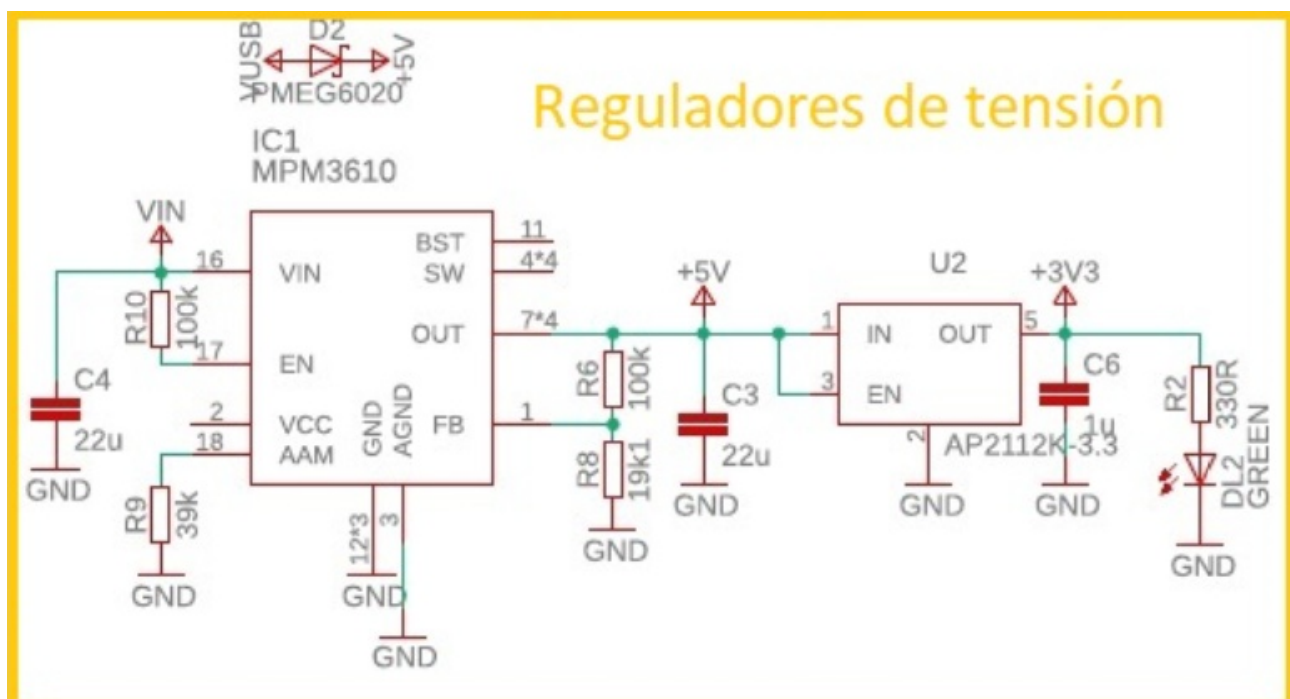
Sheet: 1/1

Reference Designs are provided "AS IS" WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. These designs are provided for informational purposes only. The Customer must not use these designs for any purpose other than that for which they are intended. The Customer must not use these designs for any purpose other than that for which they are intended. The Customer must not use these designs for any purpose other than that for which they are intended.

Como se puede ver en la imagen anterior, cada componente electrónico que encontramos en la placa tiene su representación en el esquema electrónico, las líneas verdes representan las conexiones entre los distintos pines (patas) de los componentes electrónicos que encontramos en la placa.

Todo dispositivo electrónico necesita una fuente de energía para funcionar, por lo general esta fuente de energía es una batería o fuente de tensión, que proporciona un voltaje al dispositivo electrónico. El valor del voltaje de la batería o fuente de energía muchas veces no suele ser del valor requerido por los componentes electrónicos para su correcto funcionamiento, puede ser un voltaje mayor del que soportan los componentes electrónicos, o menor del requerido por estos para su correcto funcionamiento. Por lo que en estos casos se usa un regulador de tensión que convierte el voltaje de la batería o fuente de tensión al voltaje que necesitan los componentes electrónicos para funcionar correctamente.

En la siguiente imagen podemos ver la representación en el esquema de los reguladores de tensión que encontramos en la placa Arduino Nano Every.



En un esquema electrónico todos los símbolos (un símbolo es un dibujo en el esquema que representa a un componente electrónico de la placa, el símbolo tiene los pines de entrada, salida y alimentación del componente electrónico) llevan un identificador: IC1, U2, R8, C4, etc..

Este identificador es único y no se puede repetir, cada identificador o símbolo representa en el esquema a un componente electrónico que luego encontraremos en la placa.

En la imagen superior podemos ver al identificador IC1, cuyo símbolo representa al componente electrónico [MPM3610](#), el MPM3610 es un **regulador de tensión conmutado (o DC/DC)**. En este caso este regulador recibe a su entrada un voltaje de una batería o fuente de tensión exterior (por ejemplo 10V de tensión continua) y lo convierte a 5V de tensión a su salida. Esos 5V es el voltaje de alimentación que en este caso necesita el microcontrolador de la placa Arduino para su correcto funcionamiento.

En la imagen superior también vemos al símbolo U2, este símbolo se corresponde con el componente [AP2112K-3.3](#), que es un **regulador de tensión disipativo** que convierte 5V de tensión a su entrada a 3.3V de tensión a su salida. Estos 3V de tensión de alimentación los requiere para su funcionamiento el componente de la placa que se usa para comunicar al microcontrolador con el

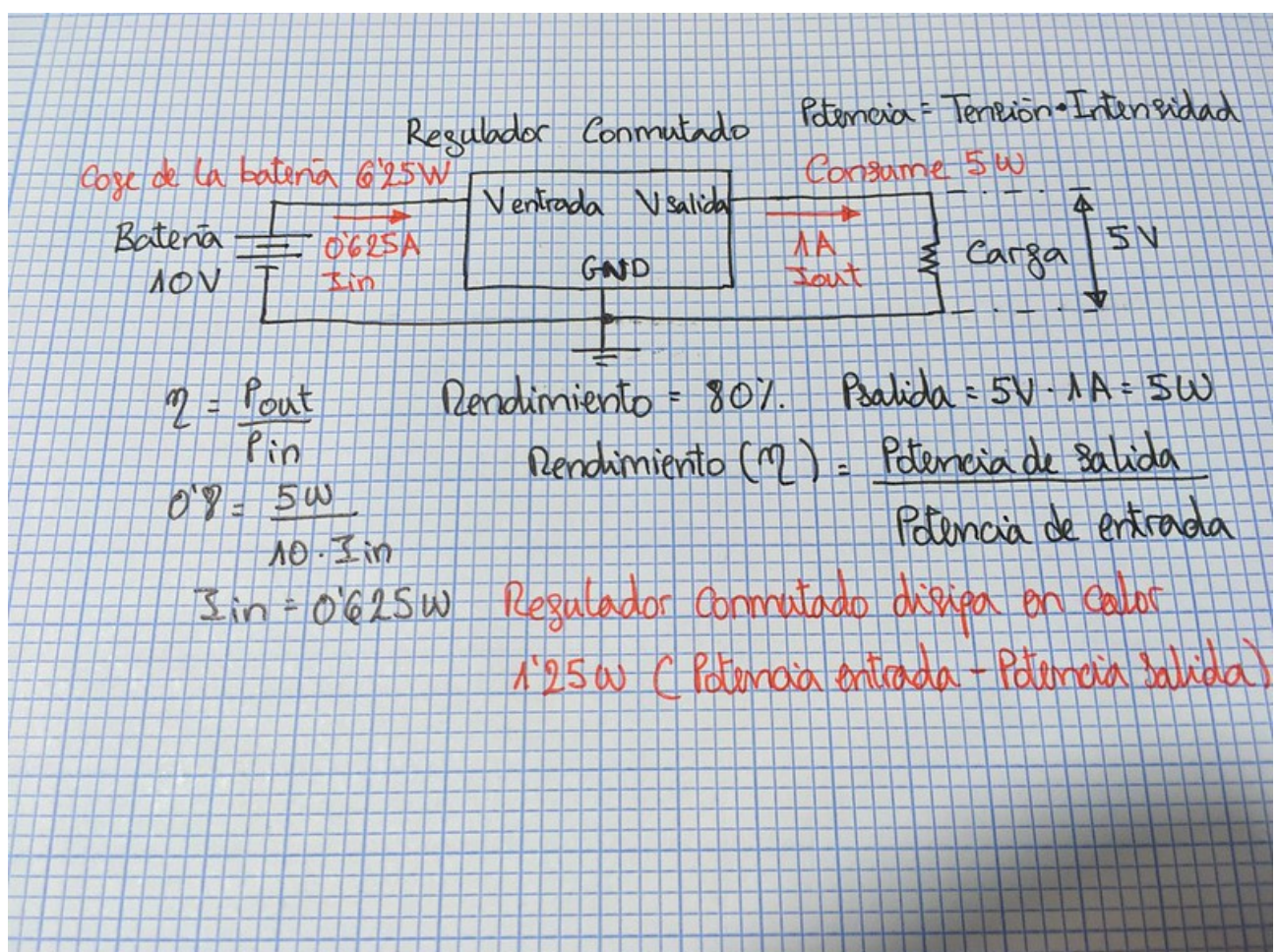
ordenador, componente que no puede funcionar a tensiones de alimentación superiores a 3.3V.

Por lo que en esta placa tenemos dos tipos de reguladores de tensión: un regulador de tensión conmutado y un regulador de tensión disipativo.

La principal diferencia entre los reguladores conmutados y los reguladores disipativos, es que los reguladores conmutados son mucho más eficientes. La corriente a la entrada y a la salida de un regulador disipativo son siempre iguales, mientras que en un regulador conmutado la corriente de entrada y de salida son distintas, y depende del rendimiento del regulador conmutado.

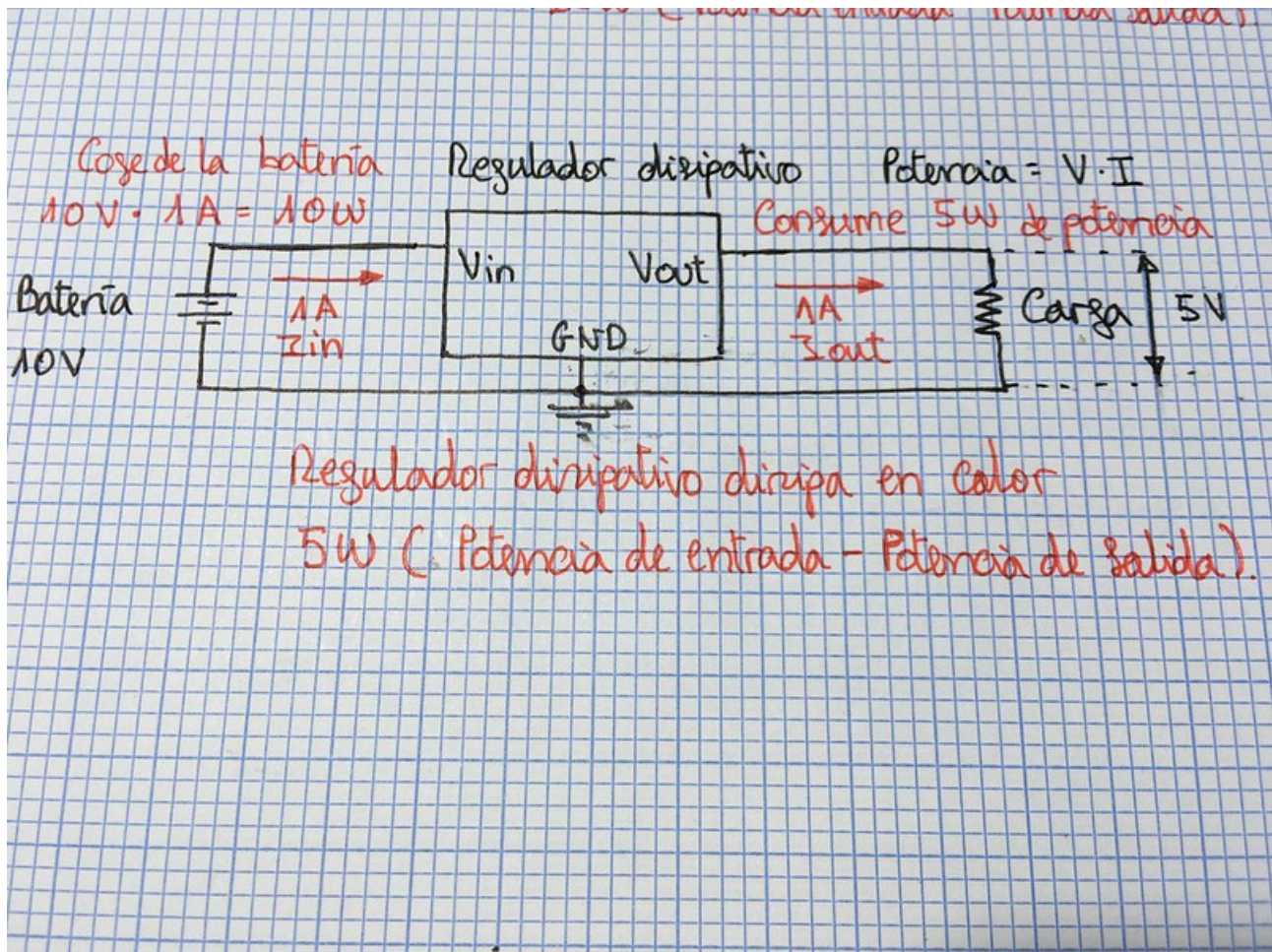
Para explicar lo anterior vamos a poner el siguiente ejemplo, tenemos una carga que necesita una tensión de 5V para funcionar y consume 1A, es decir está consumiendo una potencia de 5W (Potencia (P) = Tensión (V) * Intensidad (I)), el circuito se va a alimentar a través de un regulador desde una batería que nos proporciona una tensión de 10V.

Si usamos un regulador conmutado por ejemplo con un rendimiento del 80%, se puede ver en los cálculos de la siguiente imagen que la carga consume 5W de potencia y coge de la batería 6,25W, es decir estamos perdiendo 1,25W en el regulador conmutado, 1,25W que se disiparán en forma de calor.



En la siguiente página vemos el resultado del mismo ejemplo para un regulador disipativo.

Si hacemos los mismo cálculos del ejemplo anterior para un regulador disipativo, donde la corriente de salida es siempre igual a su corriente de entrada, vemos que en este caso para una carga que consume 5W el regulador coge de la batería 10W (en lugar de los 6.25W que cogía el regulador conmutado), luego este regulador es mucho menos eficiente que el del caso anterior. Esos 5W de diferencia entre potencia que entrega la batería y potencia que consume la carga, se convierten en calor.



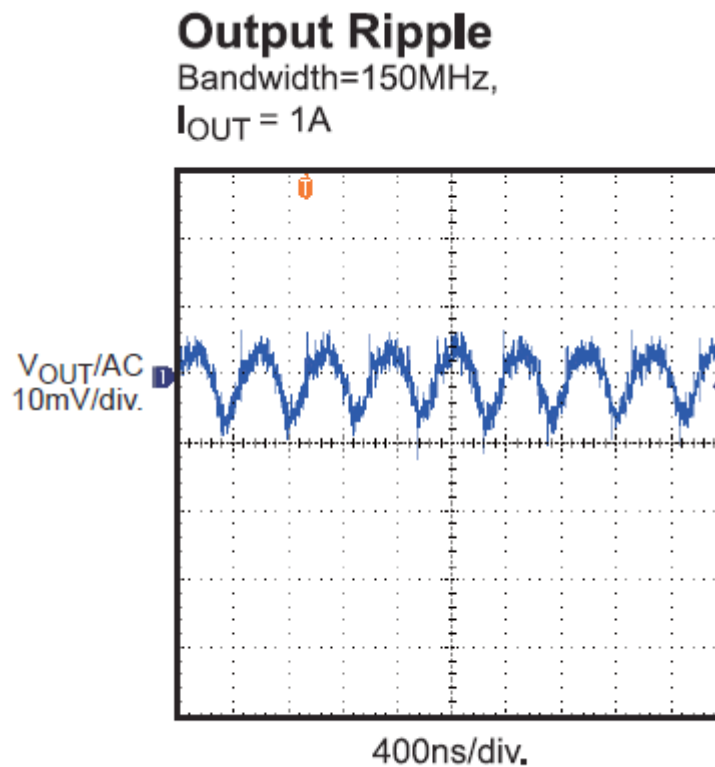
Si la temperatura que alcanza un componente en un circuito está por encima de sus límites de funcionamiento lo acabará rompiendo, o si trabajan a una temperatura alta su vida se acortará.

Cuando tenemos componentes que disipan potencias del orden de unidades o décimas de vatios lo habitual es añadir disipadores (trozos de metal) en los componentes para ayudar a sacar el calor de estos.

Luego si la diferencia de la tensión a la entrada del regulador (la tensión de la batería) es grande frente a la tensión de salida del regulador (la requerida por el circuito), por lo general tendremos que usar un regulador conmutado, ya que el regulador disipativo será muy poco eficiente o generará mucho calor que nos obligará a añadir un disipador muy grande (si se puede).

Por contra, la desventaja de los reguladores conmutados es que son más caros que los disipativos, necesitan más componentes externos para funcionar por lo que incrementa el coste y ocupan más espacio en el PCB, tienen un mayor ruido a su salida, y si no se diseña correctamente el PCB que los lleva son una fuente de emisiones electromagnéticas.

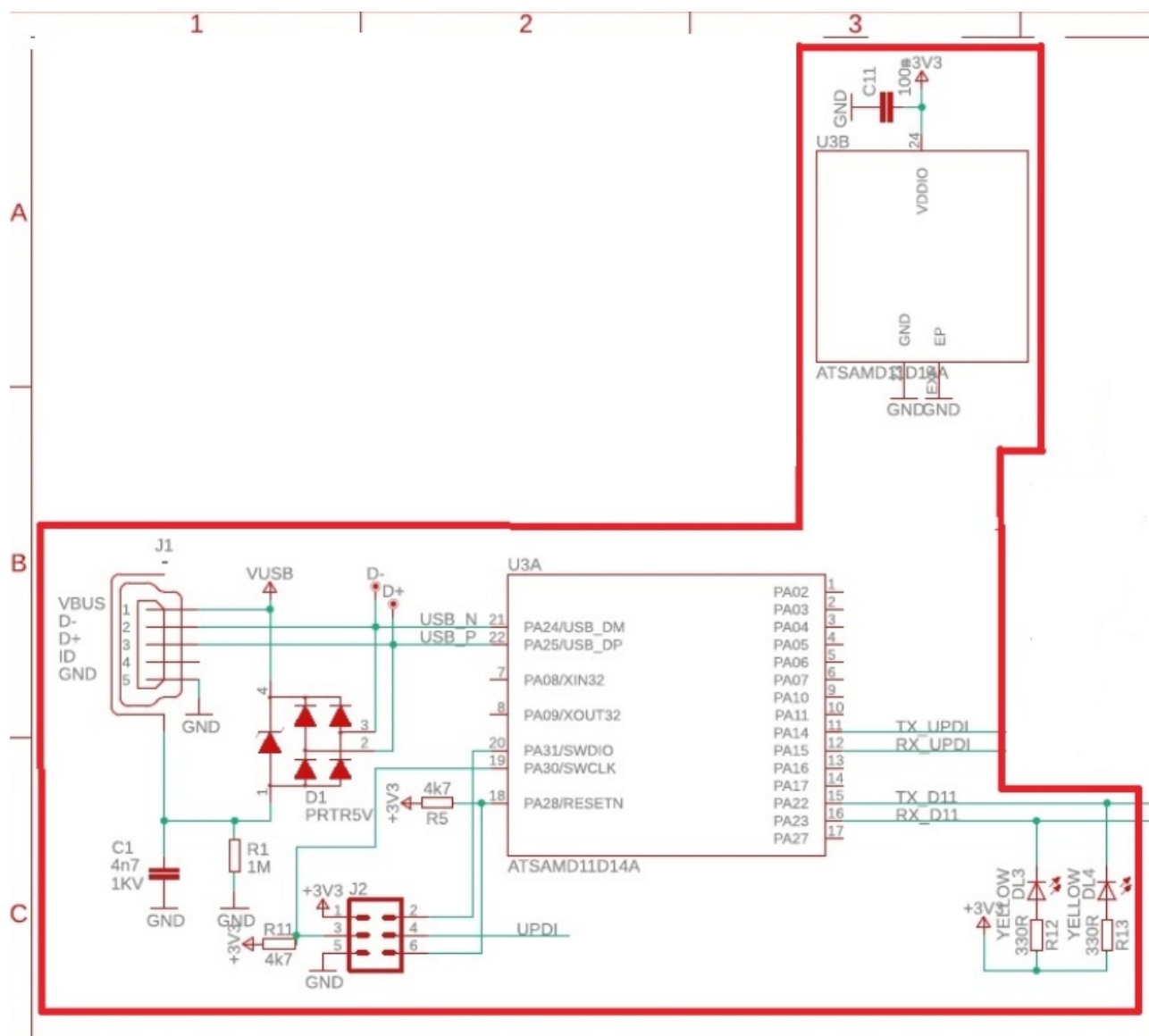
Repecto al ruido del regulador conmutao y respondiendo a una de las preguntas de la sesión, si nos vamos a la hoja de características del [MPM3610](#) vemos la siguiente imagen:



Cuando el regulador conmutado genera una salida de 3.3V y el circuito coge 1A, vemos que a la salida se produce una oscilación de 15 mV (0,015V) sobre el nivel de la salida de 3.3V, es decir la salida oscilaría entre (3,285V y 3.315V), oscilación que se produce por el funcionamiento interno del propio regulador, y que por lo general suelen ser difíciles de filtrar y eliminar de la salida de 3.3V.

Que la salida oscile entre esos valores para un circuito digital como la placa de Arduino es indiferente, pero para otros tipos de circuitos analógicos ese ruido que generan los reguladores conmutados a su salida si puede ser significativo, y puede ser no adecuado su uso.

Hasta aquí sobre los reguladores de tensión que encontramos en la placa Arduino Nano Every.



Conversor de protocolos USB a serie

Para poder programar el microcontrolador desde el ordenador es necesario tener un interfaz (una vía) de comunicación entre el microcontrolador y el pc, en este caso el Arduino Nano Every usa un conector USB para comunicarse con el microcontrolador y poder programarlo.

El circuito de la imagen superior se encarga de convertir el protocolo USB del PC a un protocolo serie que entiende el microcontrolador, ya que este microcontrolador no cuenta con el protocolo USB.

El protocolo de comunicaciones de los circuitos es análogo al lenguaje de las personas, por ejemplo una persona puede hablar inglés, alemán, chino, etc.. y para que dos personas se entiendan es necesario que hablen el mismo lenguaje. Un dispositivo electrónico o circuito integrado "habla" protocolos de comunicaciones I2C/TWI, Serie, USB, SPI, CAN Bus, etc... y para que dos circuitos integrados puedan comunicarse deben ser capaces de usar el mismo protocolo. Si dos dispositivos que quieren comunicarse no tienen un protocolo de comunicaciones común, es necesario usar electrónica adicional que "traduzca" entre protocolos, como en este caso.

De la parte del esquema de la imagen superior anotar que los dos símbolos con los identificadores U3A y U3B, no representan dos componentes distintos en la placa, si no que los dos hacen

referencia al mismo componente. Cuando los componentes tienen muchos pines (este no es el caso) es habitual que un solo componente se parta en varios símbolos (U3A, U3B, ..., U3N) para facilitar la representación y su dibujo en el esquema.

De la parte del esquema anterior también hay que darse cuenta que esta parte del circuito funciona a 3.3V, mientras que el microcontrolador funciona a 5V. Por lo general los circuitos que funcionan a voltajes distintos, no se pueden comunicar directamente entre ellos.

En electrónica digital los pines de un circuito integrado (un microcontrolador por ejemplo) pueden funcionar como entradas o como salidas, y solo entienden dos niveles de tensión, el 1 lógico y el 0 lógico. El valor de voltaje que se corresponde con un 1 lógico o un 0 lógico es función del voltaje al que se alimenta ese circuito integrado y a sus características/tecnología de construcción.

Por ejemplo si consultamos la hoja de características del [Atmega4809](#) (página 463), el microcontrolador que encontramos en la placa del Arduino Nano Every, vemos la siguiente tabla:

32.11 I/O Pin Characteristics

Table 32-17. I/O Pin Characteristics ($T_A = [-40, 85]^{\circ}\text{C}$, $V_{DD} = [1.8, 5.5]\text{V}$ unless otherwise noted)

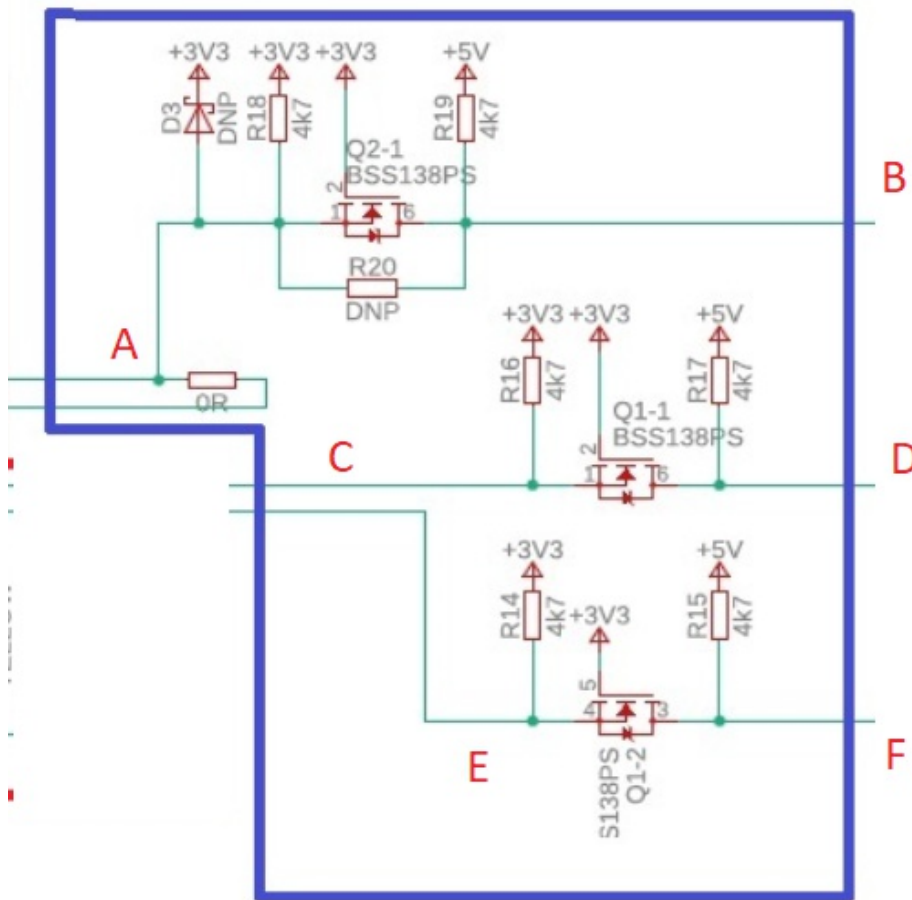
Symbol	Description	Condition	Min.	Typ.	Max.	Unit
V_{IL}	Input Low Voltage		-0.2	-	$0.3 \times V_{DD}$	V
V_{IH}	Input High Voltage		$0.7 \times V_{DD}$	-	$V_{DD} + 0.2\text{V}$	V
I_{IH} / I_{IL}	I/O pin Input Leakage Current	$V_{DD} = 5.5\text{V}$, pin high	-	< 0.05	-	μA
		$V_{DD} = 5.5\text{V}$, pin low	-	< 0.05	-	
V_{OL}	I/O pin drive strength	$V_{DD} = 1.8\text{V}$, $I_{OL} = 1.5\text{ mA}$	-	-	0.36	V
		$V_{DD} = 3.0\text{V}$, $I_{OL} = 7.5\text{ mA}$	-	-	0.6	
		$V_{DD} = 5.0\text{V}$, $I_{OL} = 15\text{ mA}$	-	-	1	
V_{OH}	I/O pin drive strength	$V_{DD} = 1.8\text{V}$, $I_{OH} = 1.5\text{ mA}$	1.44	-	-	V
		$V_{DD} = 3.0\text{V}$, $I_{OH} = 7.5\text{ mA}$	2.4	-	-	
		$V_{DD} = 5.0\text{V}$, $I_{OH} = 15\text{ mA}$	4	-	-	

Esta tabla nos dice que cuando en un pin del microcontrolador (por ejemplo el microcontrolador alimentado a 5V) configurado como entrada, ponemos un nivel de tensión comprendido entre -0.2V y 1.5V ($0.3 \times V_{DD}$, V_{IL} en rojo) se leerá un 0 lógico. Y si en un pin configurado como entrada ponemos un valor de tensión comprendido entre 3.5V y 5.2V (V_{IH} en rojo) leerá un 1 lógico.

En la tabla superior en el recuadro en amarillo, podemos ver los valores de tensión que genera un pin del microcontrolador cuando lo configuramos como salida. Si el pin del microcontrolador (microcontrolador funcionando a 5V) lo configuramos como salida y escribimos en él un 0 lógico, generará en este pin un valor (V_{OL}) comprendido entre 0 y 1V (en función de la corriente que entre por el pin, cuanto menor sea la corriente más cercano está a los 0V), y si un pin lo configuramos como salida y escribimos en él un 1 lógico, generará un valor entre 4V y los 5V de alimentación.

El circuito que convierte el protocolo USB a serie funciona a 3.3V, por lo que el valor de tensión de uno de sus pines de salida cuando quiera mandar un 1 lógico estará cercano a 3.3V, por debajo de los 3.5V que como mínimo necesita una entrada del microcontrolador funcionando a 5V para interpretar que está leyendo un 1 lógico, por lo que no se pueden comunicar directamente ambos circuitos, hace falta un circuito adicional entre ambos que adapte los niveles de tensión.

Conversor de niveles de tensión



Este circuito para convertir los niveles de tensión se puede ver en la imagen superior, es un conversor de niveles bidireccional realizado con un mosfet de enriquecimiento de tipo N, y lo tenemos repetido 3 veces, es decir tenemos 3 líneas o canales de comunicación entre el microcontrolador y el circuito que convierte la comunicación de USB a serie.

Su funcionamiento es muy sencillo, si en A conectamos un pin de salida y en B un pin de entrada, y en A ponemos una tensión de 3.3V en B tenemos una tensión de 5V, y si en A ponemos una tensión de 0V en B tenemos una tensión de 0V. Es decir convierte la tensión de un pin de salida de 3.3V a 5V en el pin de entrada (a un valor en el que el pin de entrada del microcontrolador alimentado a 5V interpreta correctamente el 1 lógico).

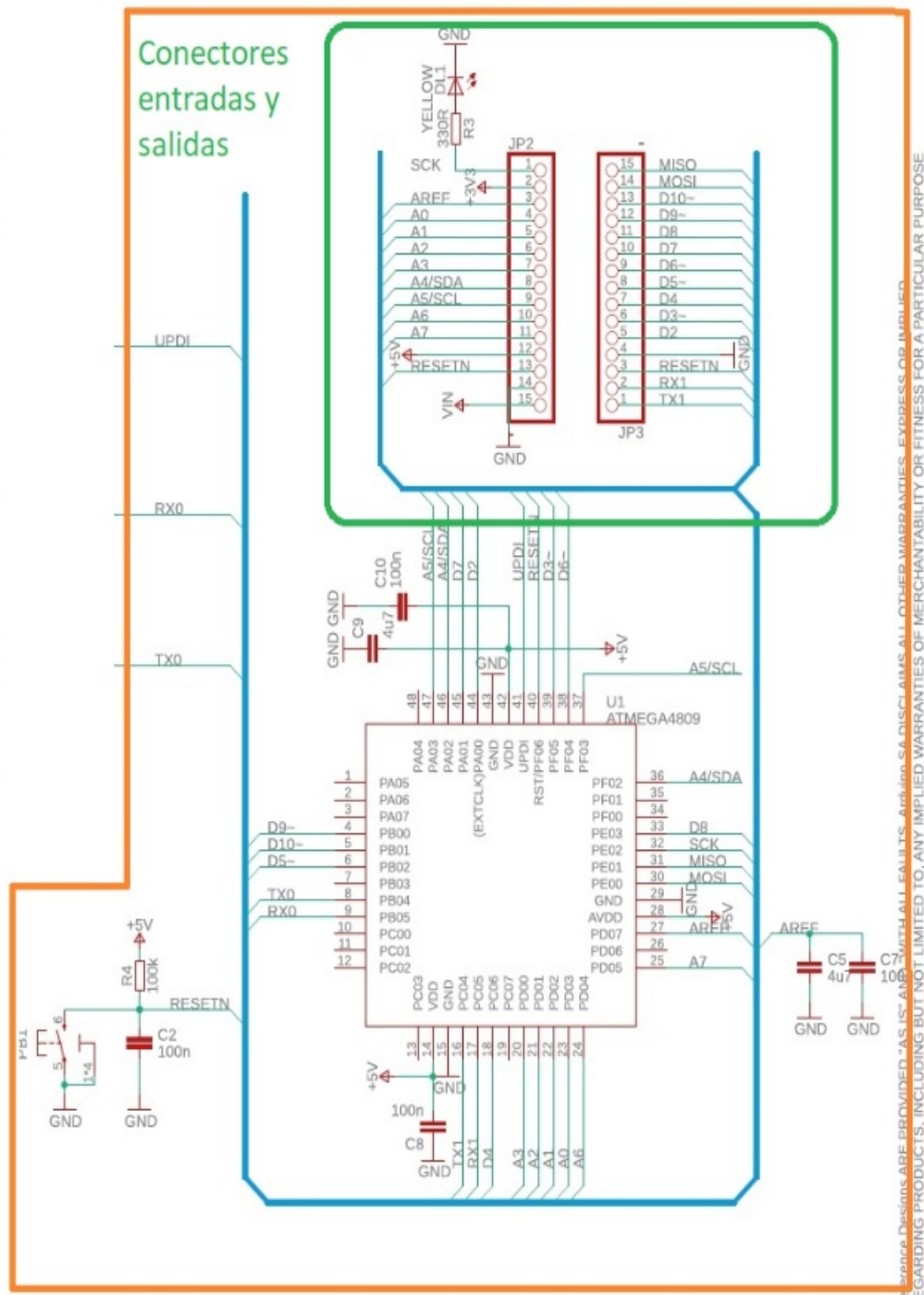
Y también hace la conversión a la inversa, si en D tenemos un pin de salida y en C un pin de entrada, cuando en D tengamos un valor de 5V en C tendremos un valor de 3.3V, y cuando en D tengamos un valor de 0V en C tendremos un valor de 0V.

Si conectamos directamente una salida de un circuito alimentado a 5V a una entrada de un circuito alimentado a 3.3V, probablemente se acabará dañando uno o ambos.

No detallo el funcionamiento de esta parte del circuito ya que no lo hicimos en la sesión, solo tener en cuenta que cuando hay circuitos integrados que funcionan a voltajes distintos, lo más probable es que no se pueda hacer una conexión directa entre sus entradas y salidas, y hay que usar un conversor de niveles entre ellos.

Microcontrolador

Conectores entradas y salidas



Para finalizar hemos hablado sobre el microcontrolador, en este caso encontramos un [Atmega4809](#), que mediante los conectores de la placa (recuadro en verde) se comunica con otros dispositivos para realizar una tarea determinada.

Se ha hablado de la importancia de añadir condensadores cerca de los circuitos integrados, ya que todos los cables y pistas del PCB tienen una pequeña inductancia, la inductancia se opone a los cambios de corriente. Si por ejemplo tenemos un circuito integrado lejos de la batería (hay una inductancia entre la batería y el circuito integrado debido a las pistas del PCB o cables) y ese circuito requiere un pico de corriente, ese pico de corriente tardará "un poco" en llegar desde la batería, provocando que tengamos una pequeña caída de voltaje o ruido en la línea de alimentación

del circuito integrado.

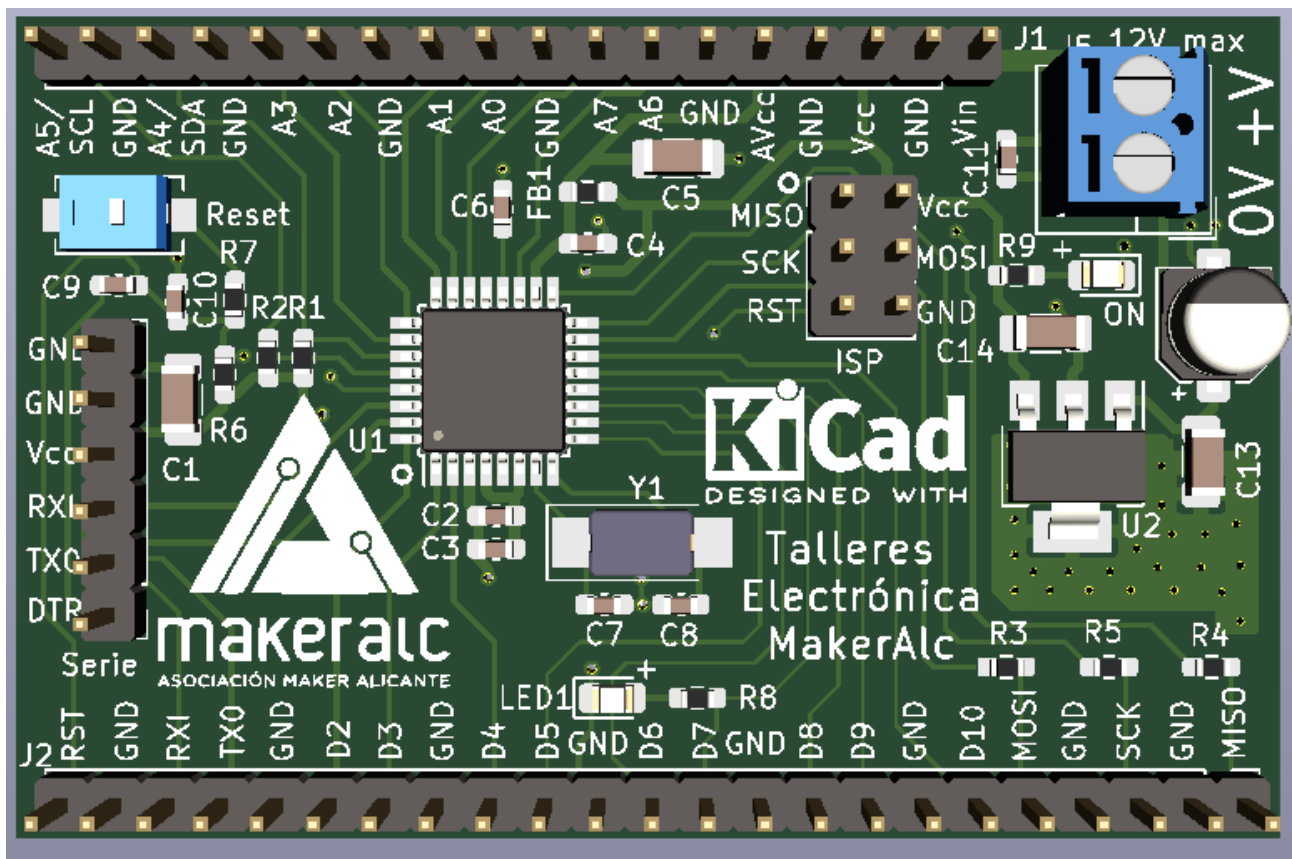
Poniendo un condensador junto al circuito integrado, tenemos una reserva de energía cerca del circuito integrado que puede demandar un pico de corriente en un momento determinado, al estar el condensador cerca la inductancia entre condensador y circuito integrado debido a la inductancia de las pistas del PCB es mínima, por lo que ese condensador proporcionará el pico de corriente al circuito integrado mientras llega la energía desde la batería, minimizando el ruido que se pudiera producir en la línea de alimentación cuando el integrado demanda un pico momentaneo de corriente.

Se ha comentado que los pines de salida de un microcontrolador puede proporcionar o absorber un valor de corriente máximo cuando funcionan como salidas, como se puede ver en la siguiente tabla:

Symbol	Description	Conditions	Min.	Max.	Unit
V_{DD}	Power Supply Voltage		-0.5	6	V
I_{VDD}	Current into a V_{DD} pin	$T_A = [-40, 85]^{\circ}\text{C}$	-	200	mA
		$T_A = [85, 125]^{\circ}\text{C}$	-	100	mA
I_{GND}	Current out of a GND pin	$T_A = [-40, 85]^{\circ}\text{C}$	-	200	mA
		$T_A = [85, 125]^{\circ}\text{C}$	-	100	mA
V_{PIN}	Pin voltage with respect to GND		-0.5	$V_{DD}+0.5$	V
I_{PIN}	I/O pin sink/source current		-40	40	mA

Un pin de salida del microcontrolador como máximo puede absorber o proporcionar una corriente máxima de 40 mA, y todos los pines que forman un puerto del microcontrolador como máximo pueden absorber o proporcionar una corriente máxima de 200 mA en conjunto.

Cuando es necesario encender o apagar un dispositivo que requiere más corriente con uno de los pines del microcontrolador funcionando como salida, es necesario el uso de transistores, siendo los MOSFETs más eficientes que lo BJT, ya que se pierde menos potencia en ellos, ya que el BJT o transistor Bipolar por construcción siempre va a tener como mínimo una caída de tensión de 0.6V entre los terminales (colector-emisor) que funcionarían como interruptor para conmutar la carga.



Para finalizar se propone hacer una placa compatible con Arduino para las siguientes sesiones del taller, es una placa con los componentes mínimos: microcontrolador, conectores de entrada y salida, un conector de programación (el conversor de protocolos USB-Serie no estaría en el PCB y se usaría uno externo) y un regulador disipativo.

Es una placa que permite explicar conceptos sobre el diseño de PCBs, pero que quizás puede tener demasiados componentes para alguien que hace por primera vez un PCB, pudiendo ser complicada para un "primer pcb"

Por lo que se valora una segunda alternativa de PCB más sencillo, con menos contenido de electrónica, para los participantes del taller que quieren aprender a hacer un PCB con un interés más artístico que electrónico, como ha surgido en los últimos años con los "badges":
<https://www.tindie.com/browse/electronic-badges/>

Posibles mejoras para la realización del taller:

- Mejorar el tamaño de la pantalla vista con el proyector.
- Colgar la pizarra.
- Por mi parte preparar un guión similar a esto y subirlo a github antes de cada sesión, para que se sepa con antelación lo que se va a tratar y a mi me facilite realizar la sesión.
- Preparar un PCB sencillo donde la única parte electrónica sean LEDs-Resistencias- Batería, y con el que se pueda aprender el proceso de hacer y mandar a fabricar un PCB al mismo tiempo que se hace una pequeña "obra de arte".