

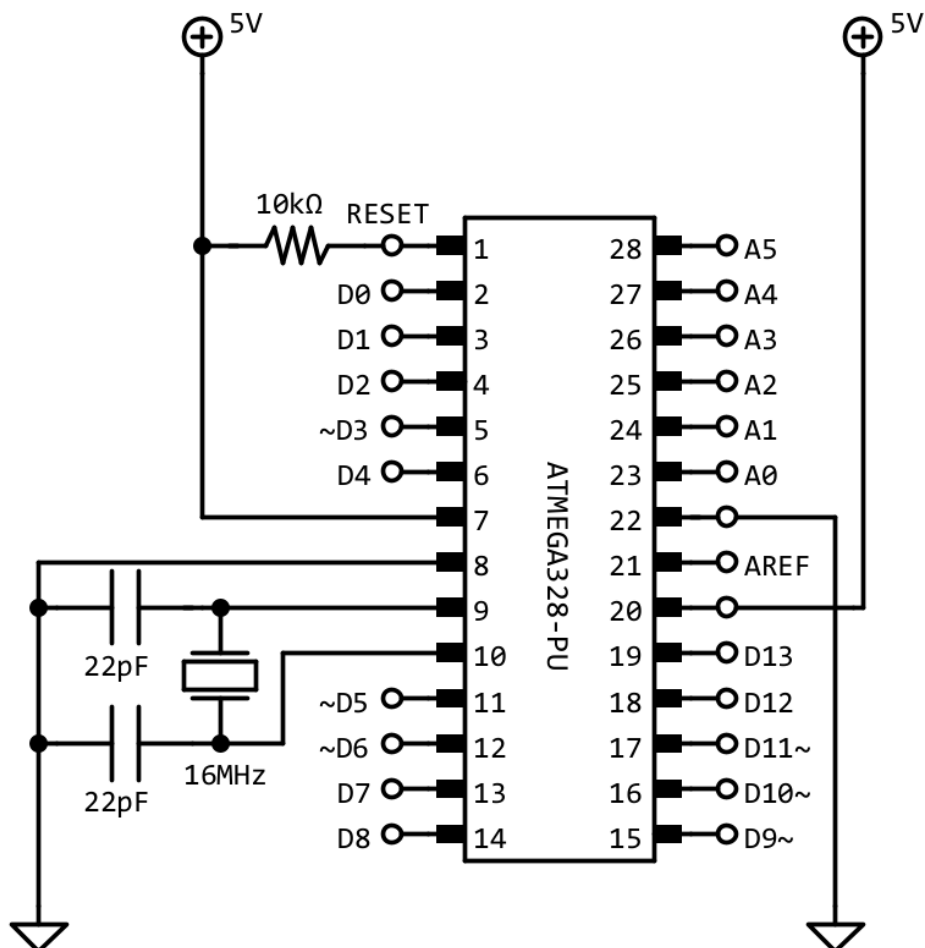
## Sistemas embebidos con Arduino

Muchas veces iniciamos un proyecto experimentando y probando cosas con nuestra placa Arduino y una protoboard. Luego las cosas se ponen un poco más serias y es necesario dar un paso más, por ejemplo fabricando un circuito impreso o [shield](#) para contener todos los componentes adicionales al Arduino. En este artículo veremos como construir un Arduino Uno a medida, es decir, bajo nuestros requerimientos.

El prototipo estará formado por el microcontrolador, la fuente de alimentación, un teclado de entrada de datos y una interfaz para el control de dos motores paso a paso.

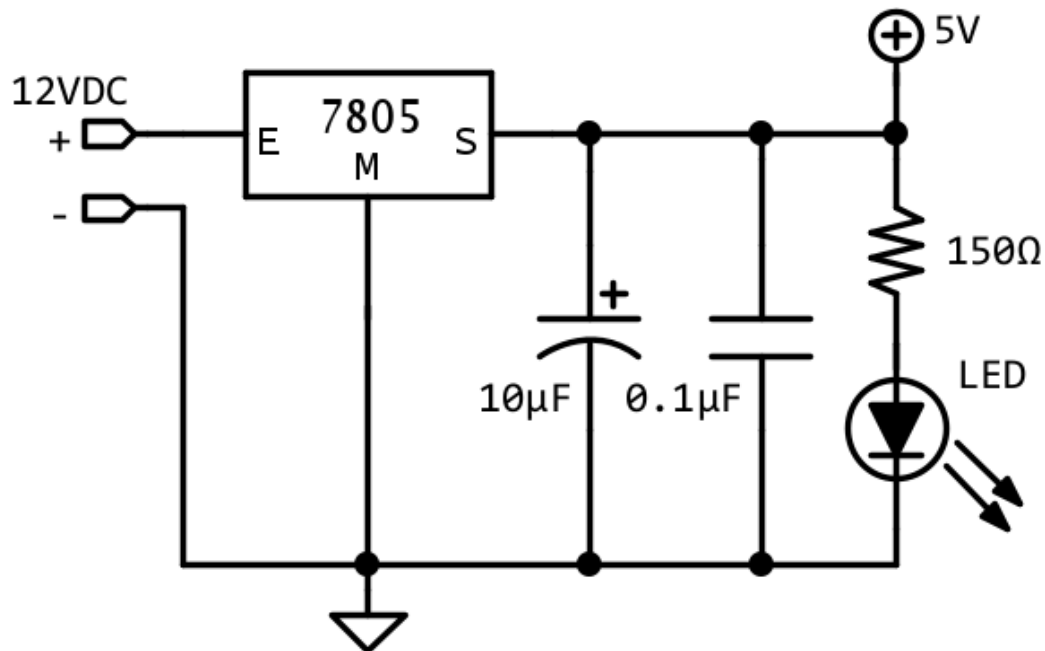
### Arduino Breadboard

Investigando en Internet bajo los términos [Arduino Breadboard](#) o [Arduino Standalone](#) encontraremos muchos proyectos para montar nuestra propia Arduino. El circuito mínimo se basa en un microcontrolador **ATMEGA328-PU**, un cristal de 16MHz, dos capacitores cerámicos y una resistencia.



## Fuente de alimentación

Nuestra placa Arduino a medida funciona con 5V de corriente continua, por lo que será necesario contar con un regulador de tensión para poder alimentarla desde una batería de 9V o una fuente. Estoy utilizando el [regulador lineal LM7805](#) junto a algunos componentes más:



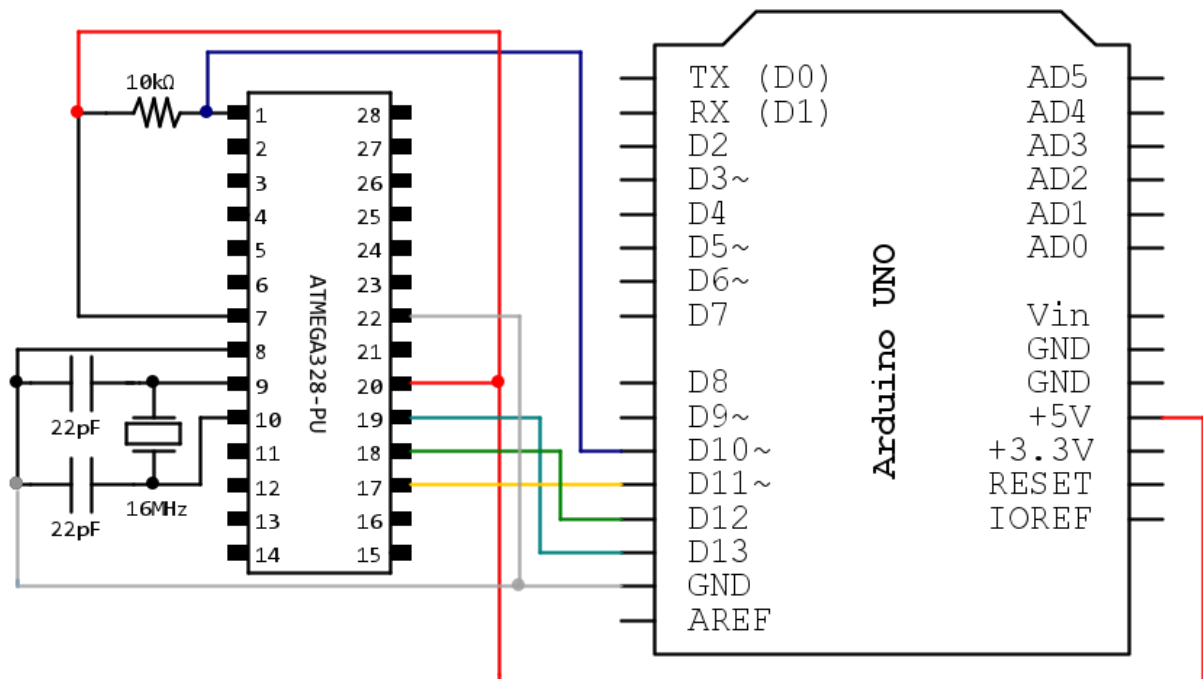
Una vez realizadas todas las conexiones de nuestra fuente de alimentación es importante comprobar que no existan cortocircuitos o problemas de soldadura. Hay que probar con un multímetro que la polaridad y la tensión sean correctas antes de conectar el microcontrolador.

## Cargando el bootloader

Una vez que nuestro hardware está terminado, es hora de cargar el **bootloader**. Un microcontrolador generalmente se programa a través de un dispositivo programador, a menos que tenga en su memoria un pequeño firmware que permita instalar un nuevo programa sin la necesidad de un programador externo. Esta es la función del bootloader o gestor de arranque.

Para cargar el bootloader vamos a necesitar una placa Arduino Uno a la que le debemos subir el sketch **ArduinoISP**. Para hacer esto abrimos el sketch desde el menú **Archivo > Ejemplos > 11. ArduinoISP > ArduinoISP**.

El siguiente paso será conectar la placa Arduino Uno a nuestra placa Arduino «a medida».



Ahora es necesario cambiar el programador. Vamos al menú **Herramientas > Programador** y elegimos la opción **Arduino as ISP**. Volvemos al menú **Herramientas** y elegimos la opción **Quemar Bootloader**. ¡Listo! Tenemos nuestra placa Arduino lista para poder trabajar.

### Problemas al cargar el bootloader

Puede ocurrir que al intentar cargar el bootloader, el IDE de Arduino informe sobre un error de firma y no permita continuar con el proceso.



En este caso ocurrió debido a que el microcontrolador que se está utilizando es el ATMEGA328-PU y el que utiliza la placa Arduino Uno es el **ATMEGA328P-PU**. La gran diferencia entre uno y otro es el consumo de energía. Para poder continuar vamos a engañar al IDE modificando el archivo **avrdude.conf**.

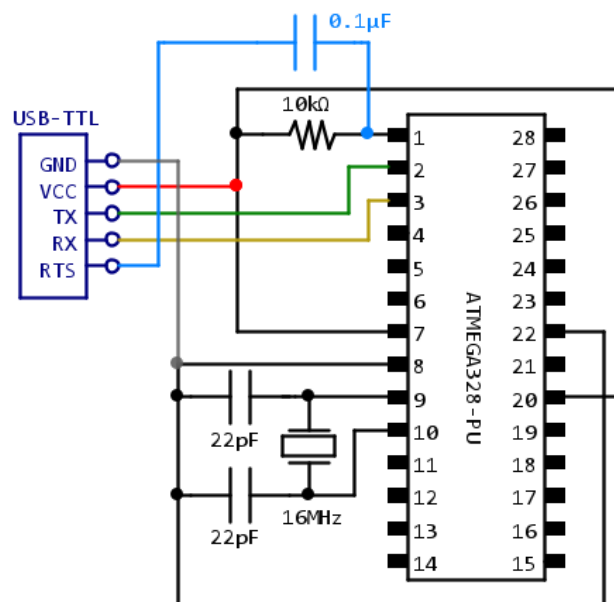
- Buscar y abrir la carpeta en la que está instalado Arduino IDE, luego navegar hasta la subcarpeta **\hardware\tools\avr\etc.**
- Hacer una copia de respaldo del archivo **avrdude.conf**.
- Abrir el archivo **avrdude.conf** en un editor de texto.
- Buscar **0x1e 0x95 0x0F**, que es la firma ATMEGA328P.
- Reemplazarla por **0x1e 0x95 0x14** (esta es la firma ATMEGA328).
- Guardar el archivo.
- Reiniciar el IDE de Arduino.
- Continuar grabando el bootloader, y una vez que se complete la carga, restaurar la copia de seguridad.

```
avrduide.conf
~/arduino15/packages/arduino/tools/avrduide/6.3.0-arduino17/etc
Guardar

8788     size = 3;
8789     read = "0 0 1 1 0 0 0 0 0 0 x x x x x",
8790           "x x x x x x a1 a0 o o o o o o o";
8791     ;
8792 ;
8793
8794 part parent "m328"
8795     id           = "m328p";
8796     desc         = "ATmega328P";
8797     signature    = 0x1e 0x95 0x0f;
8798
8799     ocdrev       = 1;
8800 ;
8801
8802 part parent "m328"
8803     id           = "m328pb";
8804     desc         = "ATmega328PB";
8805     signature    = 0x1e 0x95 0x16;
8806
8807     ocdrev       = 1;
8808 ;
8809
8810 #-----
8811 # ATmega32m1
8812 #-----
8813
8814 part parent "m328"
```

## Conexión USB-TTL

Para cargar nuestros sketches deberemos utilizar un adaptador USB-TTL y conectarlo de la siguiente manera:



## Recursos

- [AVRDUDE - AVR Downloader/UploaDEr](#)
- [AVRDUDE](#)
- [AVRDUDESS – A GUI for AVRDUDE](#)
- [Arduino «a medida»](#)
- [Exportar binarios compilados en Arduino](#)
- [AVRdudess + USBasp](#)