

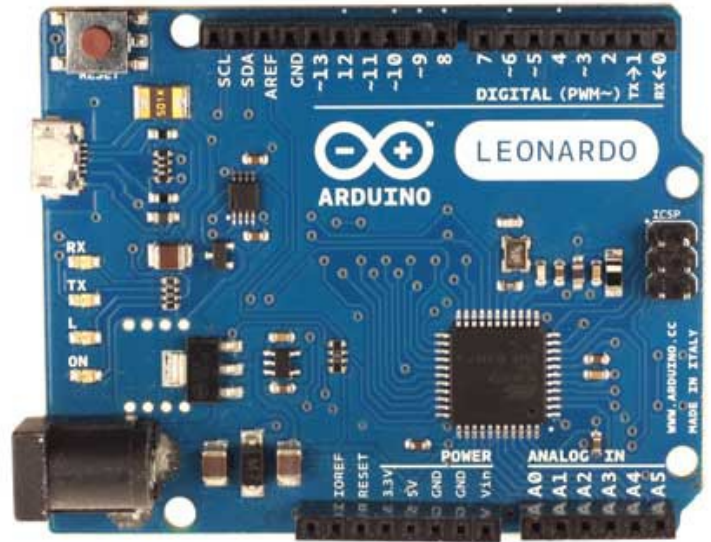
# Proyecto fin de carrera: Lista de posibilidades según dispositivo:

## -Arduino leonardo

El procesador es un Atmega32u4

-Disponemos de 20 I/O digitales. De las cuales se pueden usar 7 como salidas PWM y 12 como entradas analógicas con una resolución de 10 bit que equivale a 1024 valores posibles.

-El reloj trabaja a 16 Mhz. La alimentación es a 5 voltios por el microusb o la entrada de alimentación jack que es de entre 7 y 12 voltios recomendados, aunque tiene otros valores posibles. El jack tiene una dimensión de 2.1mm con centro positivo. Una forma alternativa de alimentación es usar los pines gnd y Vin que estan dentro de los pines de POWER. Zona inferior izquierda cumpliendo las mismas normas de funcionamiento que el jack. Esto sirve por ejemplo para conectar una batería.



Las corrientes maximas por los diferentes pines son las siguientes:

DC current per I/O digital 40 mA. DC current per 3.3 V 50mA. El pin de 5V permite mayor drenaje de corriente debido a que no esta sujeto a tantas limitaciones pero es desconocido cuanto más. Existe un fusible que se corta en el microusb cuando excede la alimentación 500mA. La conexión se restablece de forma automatica una vez desaparece dicha sobrecarga.

Como memoria disponemos de 32KB para el programa de los cuales 4KB son ocupados por el bootloader de arduino. Adicionalmente disponemos de 2.5 KB de SRAM y 1 KB de EEPROM que se puede leer y escribir de forma permanente mediante la librería EEPROM.

Todas las entradas y salidas digitales se pueden configurar como digitales pull-up de entre 20 y 50KOhm. Aparece desconocido por defecto y los pines usan voltaje de 5 voltios. Por lo que hay que tener cuidado para conectarse dispositivos a diferente voltaje.

Si entramos en detalle con estos pines tenemos las siguientes características:

Pin 0-1 RX y TX TTL para el serial data al micro Hay un funcionamiento diferenciado: Serial para la comunicación serial por usb y Serial1 para la comunicación serial por los pines 0 y 1.

Comunicación TWI mediante los pines 2 y 3 correspondientes a SDA y SCL. La comunicación TWI se realiza mediante la librería Wire. Se trata de un protocolo creado por Philips formado por TWI o I2c los cuales se tratan de exactamente el mismo protocolo, es usado principalmente para sensores que disponen de microcontroladores.

#### Características del protocolo:

- Velocidad standard de **100Kbit/s** (100kbaudios). Se puede cambiar al modo de alta velocidad (400Kbit/s)
- Configuración maestro/esclavo. La dirección del esclavo se configura con software
- Solo se necesitan **dos** líneas:
  - SDA (Serial Data Line)**: Línea de datos.
  - SCL/CLK (Serial Clock Line)**: Línea de reloj, será el que marque el tiempo de RW (Lectura/Escritura)
  - Nota: Suponemos que todos los dispositivos tienen masa común, si no fuera así hay que incluir una línea de masa.*
- Los comunicación siempre tiene la estructura siguiente:
  - Transmisor: Byte de datos (8 Bits)
  - Receptor: Bit llamado ACK de confirmación.

SDA y SCL van a su pin correspondiente en cada dispositivo, de manera que todos quedan en paralelo.

Las líneas SDA y SCL están independientemente conectadas a dos resistores Pull-Up que se encargaran de que el valor lógico siempre sea alto a no ser que un dispositivo lo ponga a valor lógico bajo.

Esta es la base de la comunicación pero para leer o escribir, según el dispositivo con el que se comunica el Master la comunicación tendrá una estructura específica.

Más información en <http://www.quadruino.com/guia-2/sensores/protocolo-i2c-twig>

Interruptores externos los pines 3,2,0,1 y el 7 correspondiente a las interrupciones interrupt 0,1,2,3,4. Estos pines pueden ser configurados para ser disparados como una interrupción a valor bajo, o cambio de borde ascendente o descendente. O simplemente cambio en valor. Función `attachInterrupt()`

PWM en los pines 3,5,6,9,10,11 y 13. Proporcionan salidas PWM de 8 bit lo que equivale a 256 valores. Función `analogWrite`.

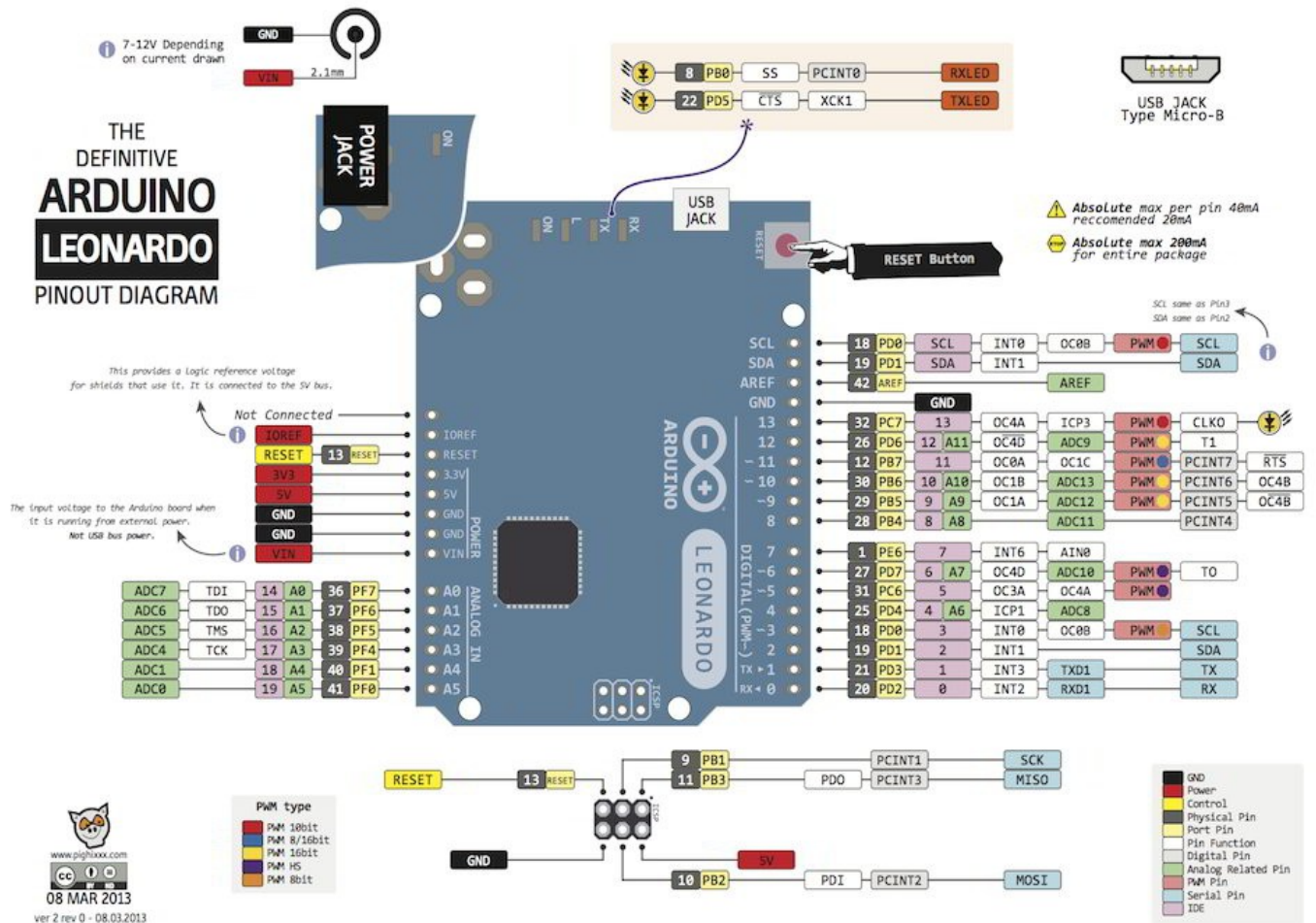
SPI protocolo de comunicación mediante el cabezal ICSP. (6 pines) Permite la comunicación SPI con la librería de mismo nombre. Estos pines se encuentran aislados del resto de pines digitales del arduino. Y únicamente son accesibles mediante el conector ICSP. Por lo que se requiere conexión ICSP para hacer funcionar las shields correspondientes.

Entradas analógicas. A0-A5 señalados correspondientemente. Y en los pines digitales los 4,6,8,9,10,12 correspondiente a los analógicos A6 a A11. Adicionalmente los pines analógicos A0-A5 también pueden ser usados como I/O digitales. Cada entrada analógica aporta 10 bits de resolución corresponde 1024 valores. Generalmente la medición se hace entre gnd y 5 voltios pero se puede cambiar el valor superior mediante el uso del pin AREF y con la función `analogReference()`

El uso de `SoftWareSerial` library permite crear comunicación serial por cualquier pin digital del Leonardo.

El bootloader permite subir nueva programación al arduino mediante el protocolo correspondiente. Se puede hacer un bypass al bootloader para cargar programación mediante el conector ICSP

# THE DEFINITIVE ARDUINO LEONARDO PINOUT DIAGRAM



## -Arduino Ethernet Board

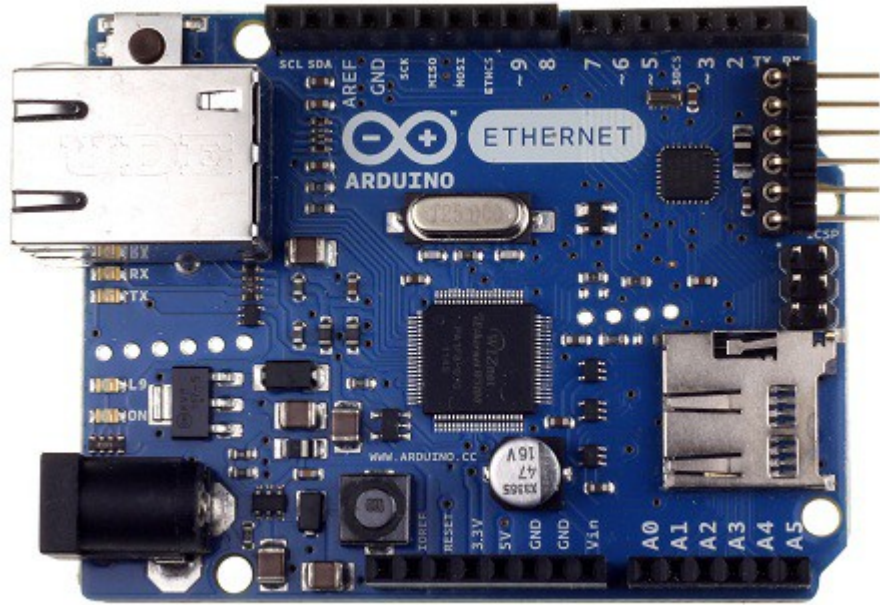
Vamos a comentar unicamente las diferencias que tenemos con el Arduino Leonardo.

En este caso disponemos de un ATmega328 como microprocesador.

Debido a la inclusión del ethernet nos encontramos ligeramente más limitados en cuanto al numero de pines digitales debido a que se encuentran usados por la conexión ethernet.

Disponemos de 14 pines digitales I/O 6 entradas analogicas, una frecuencia de 16 Mhz como el leonardo. La conexión RJ45, powerjack y conexión ICSP.

Los pines 10, 11,12 ,13 se encuentran comunicados al ethernet para la comunicación del ethernet y no deberían ser usados para otros fines. (salvo que se obvie la comunicación ethernet.) Esto hace que solo tengamos disponibles 9 pines digitales, 4 de ellos disponibles como PWM output. Se puede añadir modulos POE para proporcionar la alimentación por medio del ethernet.



Existe un lector de microSD incorporado en la placa disponible mediante la librería SD. SE puede usar para almacenar información para ser distribuida por el ethernet como el pin 10 se encuentra usado por el modulo Wiznet para el ethernet solo es posible acceder a la sd mediante el pin 4.

La programación de la placa se realiza mediante la conexión de 6 pines disponibles en la placa usando para ello un modulo USB-Serial o tambien un adaptador FTDI usb. Al cargar la programación se encuentra alimentado mediante este dispositivo.

Disponemos de los pines SDA y SCL para el protocolo I2C/TWI de la librería wire. Disponible en los pines A4 y A5

Interruptores externos en pines 2 y 3

Comunicación Spi para el ethernet: pin10 SS, pin11 MOSI, pin12 MISO, pin13 SCK. Estos pines permiten el protocolo SPI con su correspondiente librería.

Disponible Software Serial Library para crear nuevas comunicación TX RX en pines digitales



## -Como aumentar el numero de pines de salida digitales de arduino

Realizamos un Serie a Paralelo con un 74HC595. El aumento del numero de salidas digitales de arduino se realiza con un cambiador de registro como puede ser este.

En el caso del 74HC595 nos dice la hoja tecnica que permite el desplazamiento de 8 bit de una entrada serie. De salida serie o con latch. Nos permite el control de 8 salidas extras.

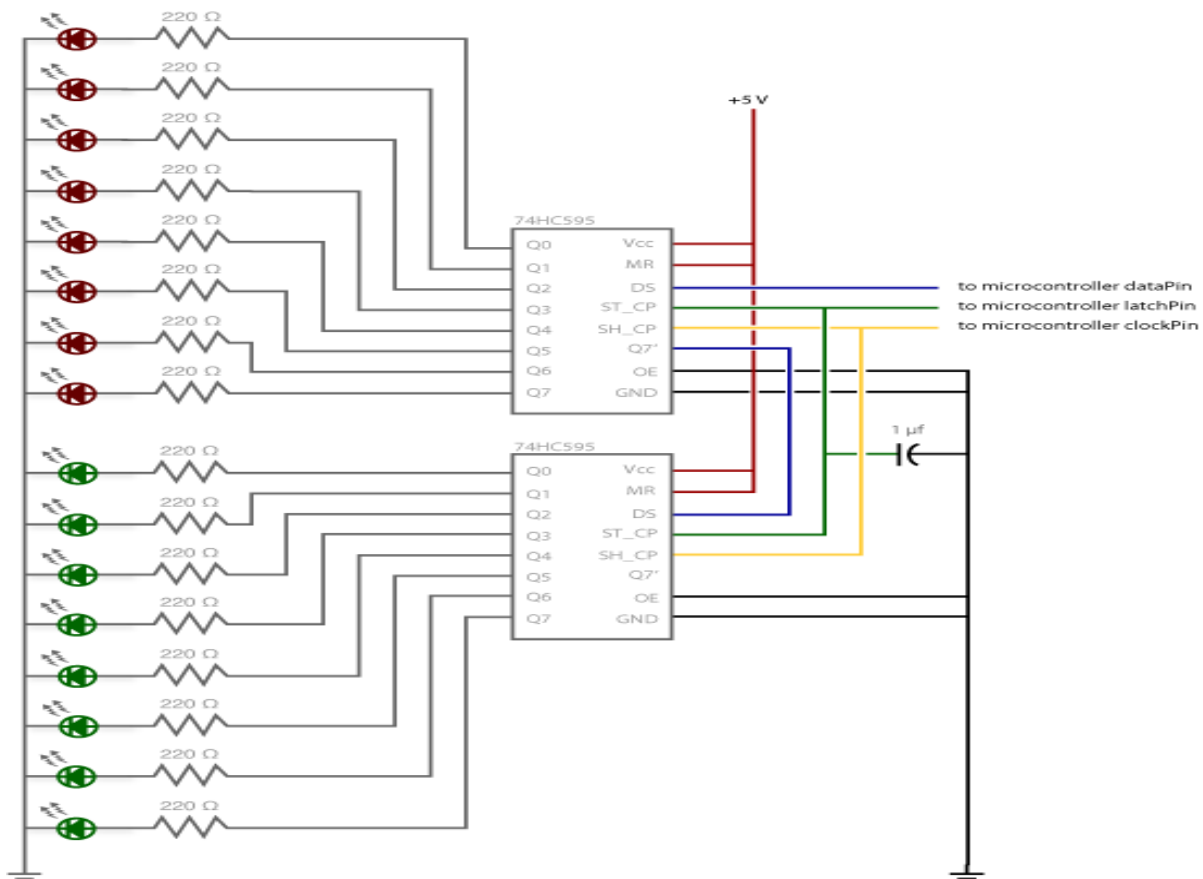
74HC595. En la hoja tecnica de datos se refieren a este integrado como un "registro de desplazamiento de 8 bits con entrada serie, salida serie o paralelo con latch (bloqueo); 3 estados.". En otras palabras, puedes usarlo para controlar 8 salidas simultaneas usando unos pocos pines del microcontrolador. La unicón de varios integrados permite aumentar el numero de pines obtenibles. La nomenclatura terminada en 595 o 596 se refiere a este tipo de dispositivos. Alguno de estos dispone de 16 salidas.

Se basa en un clock para diferenciar los valores de diferentes salidas. Los tres estados nos permite instalar estado high, low y de alta impedancia.

Mas información <http://www.arduino.cc/es/Tutorial/ShiftOut>

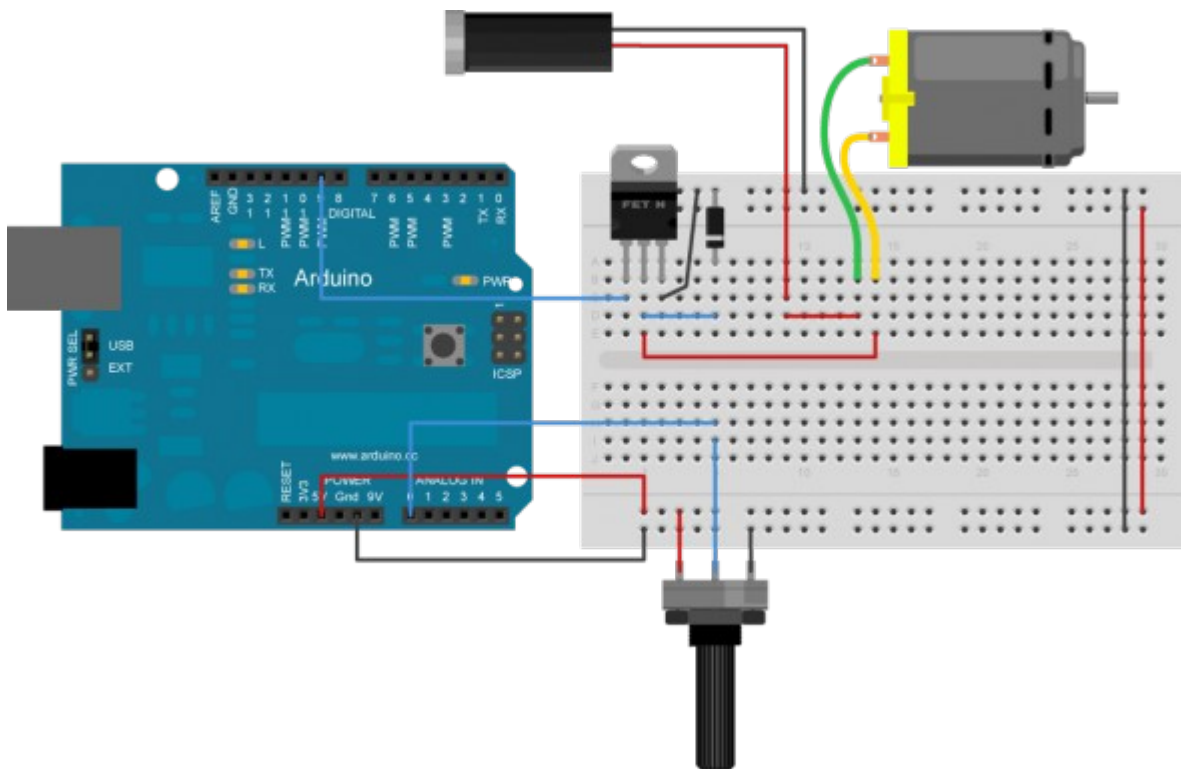
Con 3 digitales podemos ganar 8 digitales más o incluso 16 usando 2 shif register. Funcion shiftOut

**Desconozco si solo funcionan como salidas o también como entradas.**



## -Conectar diversos tipos de motores a arduino

Mas simple conexión desconexión Mediante el uso de un transistor. Se aprecia la alimentación extra.de 12 voltios por ejemplo e independiente de los 5 voltios de arduino. Aunque con gnd en comun.



Otra solución es el uso de **Controlador de motores doble puente H - L298** con alguna placa shield o montandoselo de forma casera. Disponemos de hasta 25 Watios para alimentar a dos salidas, en las que cada salida puede tener por ejemplo dos motores en paralelo.



Para controlar motores dC de mayor voltaje necesitamos un puente en H que soporte ese voltaje y la demanda de corriente/potencia que tendra dicho motor. El caso de motores con drivers especiales es otro tema.

Aqui un ejemplo de driver para grandes corrientes <http://www.pololu.com/product/2502>

Para motores de hasta 24 voltios y 12 amperios (30 de pico) usando dos motores o un solo motor de hasta 24 voltios con 24 amperior y 60 de pico.

Generalmente los drivers para motores de gran consumo y voltaje son más caros por precios de 50 dolares.

**Los drivers de motores steppers son más simples pero sus aplicaciones muy limitadas, principalmente impresoras, robots educativos, y necesidades simples.**

**Control de motores AC**

**Uso de triac + optoacoplador, algun circuito integrado U2007 u217B**

**Proyecto final de carrera realizado e con arduino**

**<http://riunet.upv.es/bitstream/handle/10251/11293/memoria.pdf?sequence=1>**

**PROYECTO FINAL DE CARRERA**

**Diseño e implementación de un circuito electrónico**

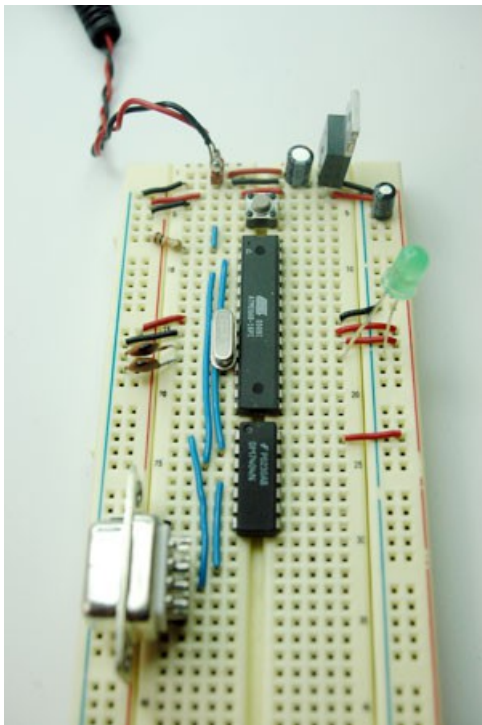
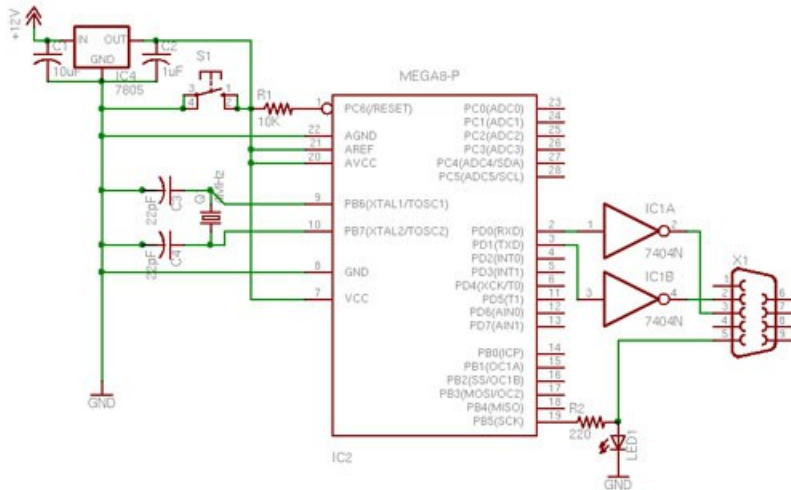
**inteligente para la emisión de sonido para control**

**de plagas**

## -Arduino más básico usar un arduino con el chip+ las resistencias, condensadores y resto de elementos necesarios.

-Usar un arduino mini: nos da salidas analogicas extras y PWM extras. Por internet se puede obtener por 3 euros cada uno.

--Usar atmega8 o atmega168 chip. Se añade los componentes restantes. El atmega se le carga el bootloader de arduino ( programa previo al sketch)



Se puede hacer diferentes variantes para lograr las características necesarias. Máxima compacidad en un PCB.



# RASPBERRY DATOS A TENER EN CUENTA

## Temás de alimentación.

-El consumo de la raspberry sin extras es de 750mA, y el sistema cuenta con un fusible de 1,1A para limitar la corriente que circula por el sistema, debido principalmente a cargas en los puertos usb. Existe un fusible temporal que limita esta corriente, (situado debajo del conector microusb. ). Al mismo tiempo se puede quitar esta limitación mediante un bypass soldando los cables por la zona inferior. La raspberry funciona perfectamente\*\* tanto si le alimentas con 5 voltios por el puerto microusb, puerto 5v y gnd de los pines gpio, incluso desde el puerto usb, a riesgo de posibles inestabilidades debido a que es recomendable usar el fusible y es mejor alimentar lo más cerca posible de la carga.

Existe un problema producido por la conexión repentina a los puertos de usb que producen un repentino incremento del consumo de corriente al conectar dispositivos como usb wifi-bluetooth, discos de alto consumo que se soluciona perfectamente soldando en paralelo al condensador existente en la placa un condensador de mayor capacidad de por ejemplo 6,3 Voltios y x Faradios. Esto evita el malfuncionamiento, reinicios y demás errores del sistema fruto de esta sobrecarga.

Finalmente existe otro detalle importante fruto de principalmente discos duros portátiles sin alimentación propia, generalmente estamos limitados a 1,1Amperios de consumo total, con la raspberry ya consumiendo casi 750 mA nos deja con escaso margen para conectar dispositivos de alta demanda energética.. Principalmente Discos duros. Por ello se usa Hub con alimentación externa para incrementar esta demanda energética, aun teniendo en cuenta de que no es recomendable que el sistema este alimentado por microusb y el hub, porque podrá extraer energía de los dos puntos.

La otra solución es usar un cable Tipo Y, con por ejemplo microusb para el disco duro, un pin usb tipo A típico al raspberry, y el otro extremo a un cargador de alimentación externo con al menos 1 Amperios. De todas formas si que es cierto que algunos discos duros de bajo consumo no tienen esta limitación.

El uso de un cargador de más de 1,1 Amperios, con usb wifi, teclado ratón, y varios usb no entraña problemas energéticos, más allá de una conexión repentina del usb wifi que pueda apagar el sistema por una repentina demanda energética.

PDF con el esquema de la raspberry, diagrama de conexiones y pines internos

[http://www.raspberrypi.org/wp-content/uploads/2012/10/Raspberry-Pi-R2.0-Schematics-Issue2.2\\_027.pdf](http://www.raspberrypi.org/wp-content/uploads/2012/10/Raspberry-Pi-R2.0-Schematics-Issue2.2_027.pdf)

Especificaciones técnicas puertos gpio [http://www.raspberrypi.org/wp-content/uploads/2012/10/Raspberry-Pi-R2.0-Schematics-Issue2.2\\_027.pdf](http://www.raspberrypi.org/wp-content/uploads/2012/10/Raspberry-Pi-R2.0-Schematics-Issue2.2_027.pdf)

## TEMAS DE COMUNICACIÓN

- Ethernet 10/100
- Uso de usb wifi o bluetooth, deben usar unos chip determinado para no tener que instalar drivers.
- ssh para uso de terminal remoto

## Guías de programación y opciones gpio

-Por defecto se suele programar en python por sencillez.

RPI.GPIO (Included by default in Raspbian) – A Python interface for working with the GPIO pins [Download Program](#), [View Documentation](#)

Note that this module is unsuitable for real-time or timing critical applications. This is because you can not predict when Python will be busy garbage collecting. It also runs under the Linux kernel which is not suitable for real time applications - it is multitasking O/S and another process may be given priority over the CPU, causing jitter in your program. If you are after true real-time performance and predictability, buy yourself an Arduino <http://www.arduino.cc> !

Guía de programación

[http://93.174.101.23/bt/2886cba6954bac9b931aa67f43f0def0e53c9e21/data/Raspberry\\_Pi\\_Education\\_Manual.pdf](http://93.174.101.23/bt/2886cba6954bac9b931aa67f43f0def0e53c9e21/data/Raspberry_Pi_Education_Manual.pdf)

También podemos hacer la programación de estos pines con c/c++ WiringPi – A C/C++ Program interfaz for working with the GPIO pins [Main Site](#), [Download Instructions](#), View Basic Setup

Permite trabajar con i2c , spi

Simulación por software de PWM

pi-blasters – A program to provide software emulation for additional PWMs [Git-hub with examples](#)

## CONTENIDO ÚTIL

Revista MagPi donde explican muchos tutoriales ideas, formas de sincronizar con otras plataformas y miles de cosas más. Revista gratuita

<http://www.themagpi.com/>

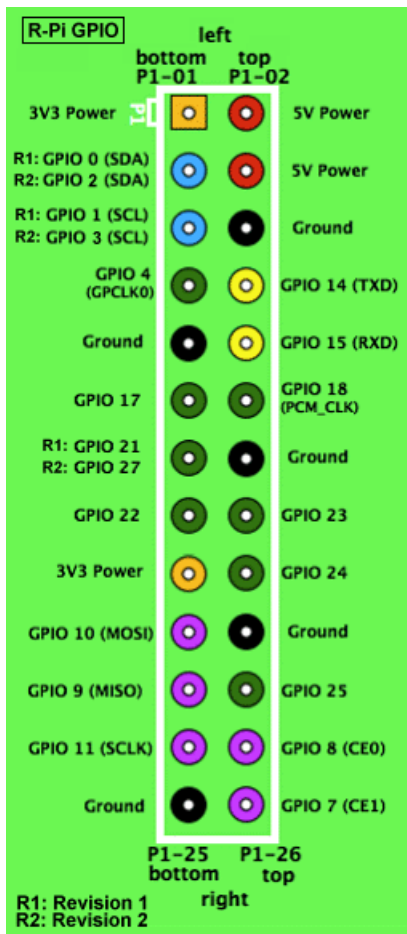
Videotutoriales sobre cientos de cosas sobre la raspberry, tocan todos los temas

<http://www.youtube.com/user/RaspberryPiBeginners/videos>

Manual oficial raspberry [http://downloads.raspberrypi.org/Raspberry\\_Pi\\_Education\\_Manual.pdf](http://downloads.raspberrypi.org/Raspberry_Pi_Education_Manual.pdf)

Más videotutoriales raspberry <http://www.instructables.com/tag/type-id/?sort=none&q=Raspberry+Pi> sobre programación y cientos de ideas.

## GPIO [http://elinux.org/RPi\\_Low-level\\_peripherals](http://elinux.org/RPi_Low-level_peripherals)



<http://en.wikipedia.org/wiki/GPIO>

Vemos los diferentes pines de conexión Podemos identificar los siguientes pines:

SDA/SCL para protocolo TWI / I2C

TXD RXD para comunicación serie.

[http://elinux.org/RPi\\_Serial\\_Connection](http://elinux.org/RPi_Serial_Connection)

MOSI, MISO, SCLK

Existen diversas placas para aumentar el numero de pines, funcionalidades. [http://elinux.org/Rpi\\_Gertboard](http://elinux.org/Rpi_Gertboard)

No tiene protección contra sobrecorriente.

### Power pins

The maximum permitted current draw from the 3.3 V pins is 50 mA.

Maximum permitted current draw from the 5 V pin is the USB input current (usually 1 A) minus any current draw from the rest of the board. [14]

•Model A: 1000 mA - 500 mA -> max current draw: 500 mA

•Model B: 1000 mA - 700 mA -> max current draw: 300 mA

Be very careful with the 5 V pins P1-02 and P1-04, because if you short 5 V to any other P1 pin you may permanently damage your RasPi. Before probing P1, it's a good idea to strip short pieces of insulation off a wire and push them over the 5 V pins so you don't accidentally short them with a probe.

El conector P5, dispone de 5v, gnd pines digitales extra y i2c

El conector P6, permite añadir un boton de reset a la placa.

Listado completo de todas las posibilidades de cada pin

[http://elinux.org/RPi\\_BCM2835\\_GPIOs](http://elinux.org/RPi_BCM2835_GPIOs)