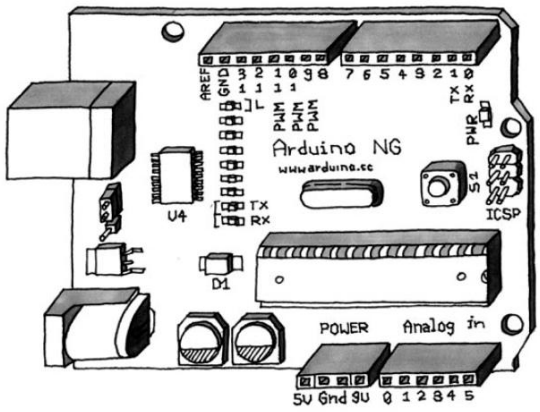
PLATAFORMA DE DESARROLLO ARDUINO:

UNA INTRODUCCIÓN PRÁCTICA



ARQUITECTURA DE COMPUTADORES

10 - febrero - 2007

Enrique José Izuel García
alu.00039@usj.es

José Antonio Esparza Isasa
alu.00033@usj.es

Tabla de contenidos

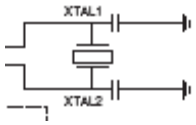
1.- Descripción de arduino, arquitectura de la placa

Arduino es una plataforma de desarrollo libre, creada bajo la licencia creative commons. Como iremos descubriendo a lo largo de este trabajo, arduino es técnicamente equiparable a muchas otras plataformas existentes en el mercado.

Desde su aparición ha ido sufriendo distintas modificaciones hasta llegar a la versión actual, conectable por USB y con acabada con componentes de montaje superficial (SMD). Esta es la versión con la que trabajaremos en este trabajo.

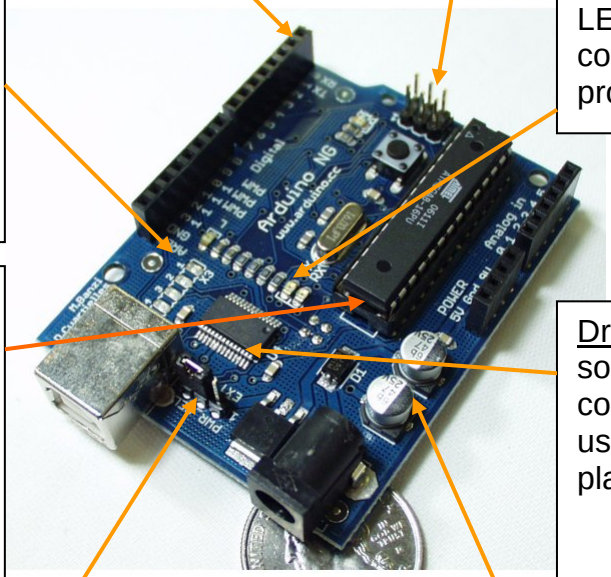
Podemos distinguir las siguientes partes en la arquitectura actual de la placa:

Oscilador: Como cualquier computador (recordemos que un microcontrolador también es un computador, aunque en miniatura) el ATMEL necesita un reloj para poder funcionar. La manera de conectarlo es la habitual en estos casos:



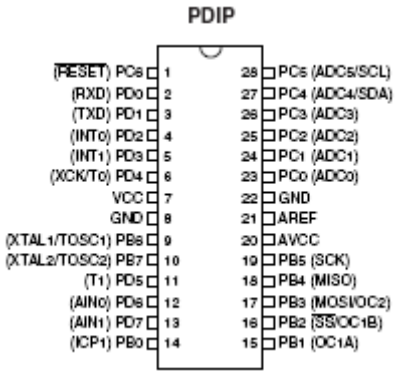
Entradas analógicas, **salidas** digitales y de anchura de pulso modulado (PWM).

ICSP: in circuit serial programming: nos permite reprogramar la rom en caso de borrado accidental.



LED's SMD para comunicarse con el programador.

Microcontrolador ATMEL: La placa integra un microcontrolador atmel atmega de 8 bits, en un encapsulado DIP (dual in-line package). Este microcontrolador está montado sobre un zócalo, para poder reemplazarlo en caso de borrado de la ROM.

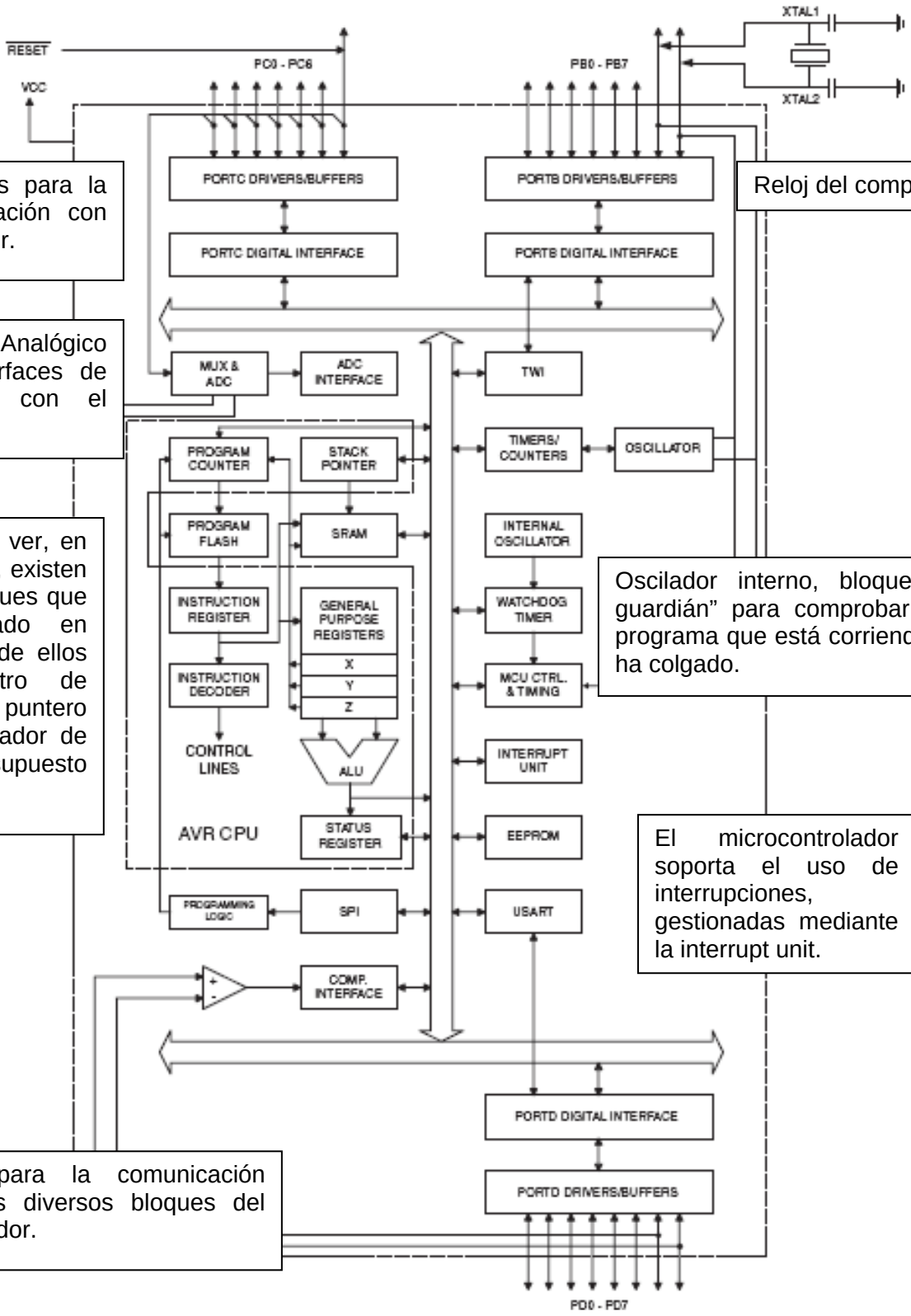


Driver FTDI que da soporte a la comunicación por usb 2.0 entre la placa y el PC.

Jumper para la selección de la fuente de alimentación: externa o por USB.

Reguladores de tensión y diodos para la estabilización de la tensión. Condensadores electrolíticos para el filtrado de la tensión de alimentación externa.

Arquitectura básica del microcontrolador ATMEL 8



Interfaces para la comunicación con el exterior.

Reloj del computador.

Conversores Analógico Digital e interfaces de comunicación con el exterior.

Como podemos ver, en la ruta de datos, existen multitud de bloques que hemos estudiado en clase, algunos de ellos son el registro de instrucción, el puntero de pila, el contador de programa, por supuesto la alu, etc ...

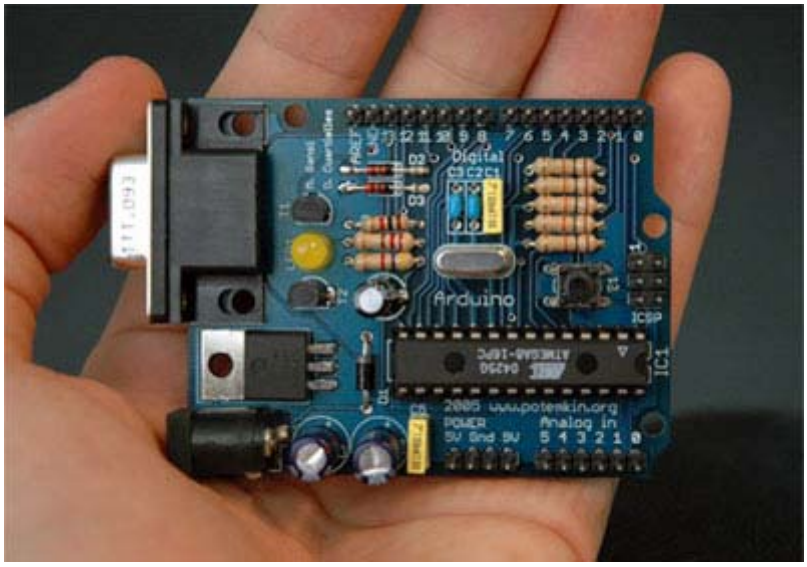
Oscilador interno, bloque “perro guardián” para comprobar que el programa que está corriendo no se ha colgado.

El microcontrolador soporta el uso de interrupciones, gestionadas mediante la interrupt unit.

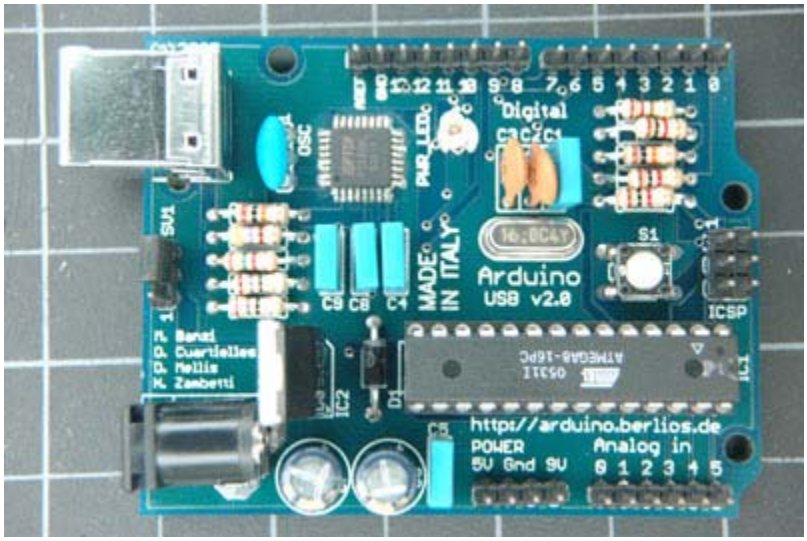
Buses para la comunicación entre los diversos bloques del computador.

2.- Versiones de la placa

Inicialmente la placa se empezó a comercializar con componentes discretos, y montada sobre una placa de circuito impreso monocapa. Al no conectarse por USB, necesitaba alimentación externa constantemente, tanto durante su funcionamiento como en su fase de programación. El hecho de conectarse por el puerto de serie también nos ahorra parte de la circuitería adicional, como los jumpers y el driver FTDI.



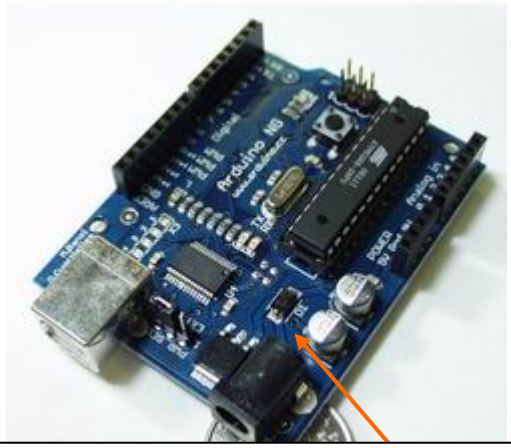
Posteriormente apareció una versión que permitía la conexión por USB de la placa, que seguía haciendo uso de componentes discretos salvo el driver FTDI. Como los esquemas eran de licencia libre se podía adquirir la placa para ser montada por el usuario final:

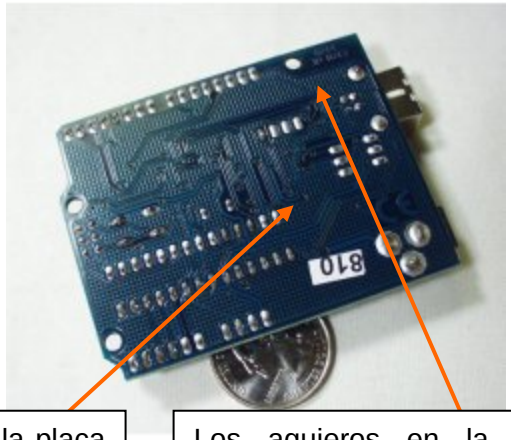


Simultáneamente se empezaron a comercializar adaptadores para convertir la versión serie a USB.

El adaptador USB serie consiste básicamente en el driver FTDI y la circuitería necesaria para su funcionamiento entre ellos los led's tx y rx para monitorizar la transmisión de datos entre el ordenador y la placa. En la parte inferior del adaptador esta el jumper que permite seleccionar la fuente de alimentación de la placa (USB o externa).

La versión actual de la placa es conectable por USB, y de doble capa:

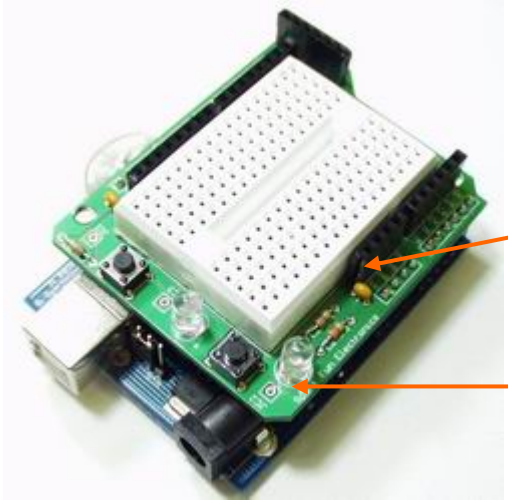




Se pueden apreciar las perforaciones en la placa que permiten interconectar las capas inferior y superior.

Los agujeros en la placa permiten utilizar separadores para fijarla a otras superficies.

Para facilitar el conexionado de otros componentes se han empezado a comercializar otras placas que se pueden acoplar en la parte superior:



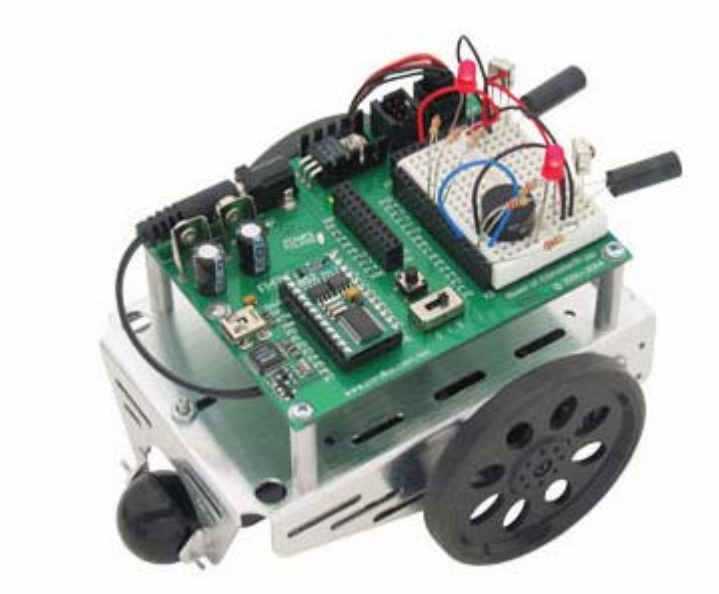
Al circuito le podemos acoplar una placa de prototipos para facilitar el montaje de otros componentes. Por supuesto las conexiones de la parte inferior pasan a la parte superior, además de otros componentes como los led's indicadores y los pulsadores de reset y control.

3.- Comparación con otras soluciones.

Existen muchas otras placas en el mercado con propósitos similares, pero ninguna de ellas cuenta con la comunidad de usuarios que hay alrededor de arduino. Esto hace que halla disponible una gran cantidad de información sobre ella tanto en lo referente a montajes como programas para su control.

Desde el punto de vista económico el tipo de licencia bajo la que está la placa hace que su coste sea muy inferior a la de otras placas que existen en el mercado, como la que vende parallax sobre la que se monta el Basic Stamp, o las placas de microchip que hay en la universidad.

Este tipo de placas se suelen vender con propósitos educativos, para laboratorios de universidades y talleres de secundaria. Algunas de ellas incluso se venden montadas sobre plataformas móviles, con servos de radio-control trucados para rotación continua, para de esta forma poder introducirse fácilmente en la robótica móvil.

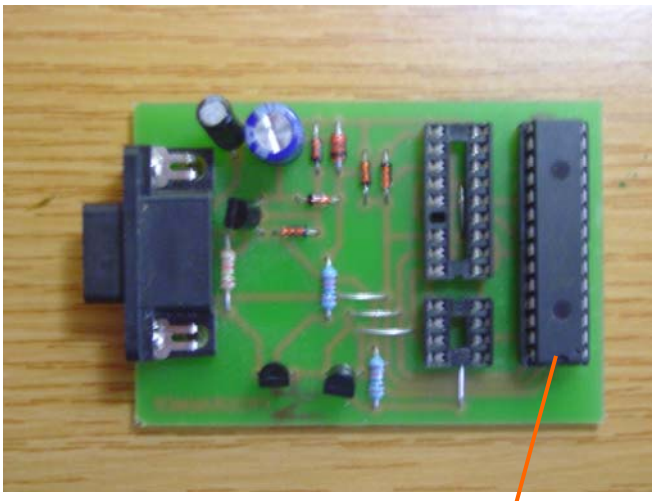


Esta placa Parallax integra un microcontrolador de microchip de montaje smd. Este ic es la parte fundamental del Basic stamp.

Este robot en concreto está programado para detectar obstáculos y evitarlos mediante los emisores y receptores de infrarrojos.

Aparte de estas placas también existen otras soluciones que consisten en la programación directa de los microcontroladores mediante programadores como los que aparecen en la imagen inferior, y que se montan en el circuito con la circuitería básica para su funcionamiento, como el oscilador y el pulsador de reset.

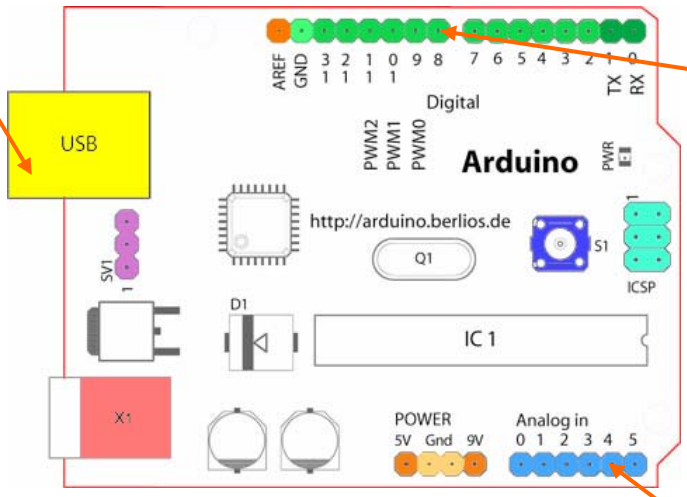
Este programador permite grabar microcontroladores Microchip de varios tipos. Se conecta por el puerto serie, y tras la programación se extrae el microcontrolador para situarlo en el circuito en el que lo vayamos a utilizar.



4.-Uso básico de la placa: programación e interfaces, ejemplo sencillo de control de un LED

La placa se comunica con los periféricos gracias a las entradas y salidas con las que cuenta:

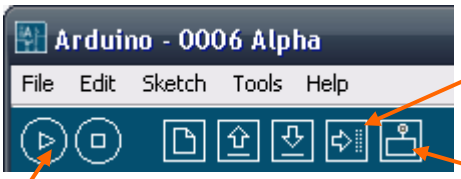
La placa hay que conectarla por USB al ordenador. Esto nos permite programar el atmel y además enviar datos al ordenador, que serán recibidos por el monitor de serie.



Las salidas digitales nos permiten obtener distintas salidas en función de las entradas que recibe, además del programa almacenado en el ID.

Entradas analógicas de la placa.

Una vez hayamos realizado el conexionado de los componentes a los interfaces correspondientes, es el momento de pasar a la programación del circuito integrado. Para esta tarea hay que conectar el dispositivo al ordenador, pulsar el botón de reset y abrir el entorno de programación, vemos un editor de texto en la zona central del programa y la siguiente barra de herramientas:



Botón para subir el programa escrito al microcontrolador.

Tras pulsar el boton “serial monitor”, podemos observar los datos enviados por la placa al ordenador.

El botón play verifica que la sintaxis del programa es la correcta.

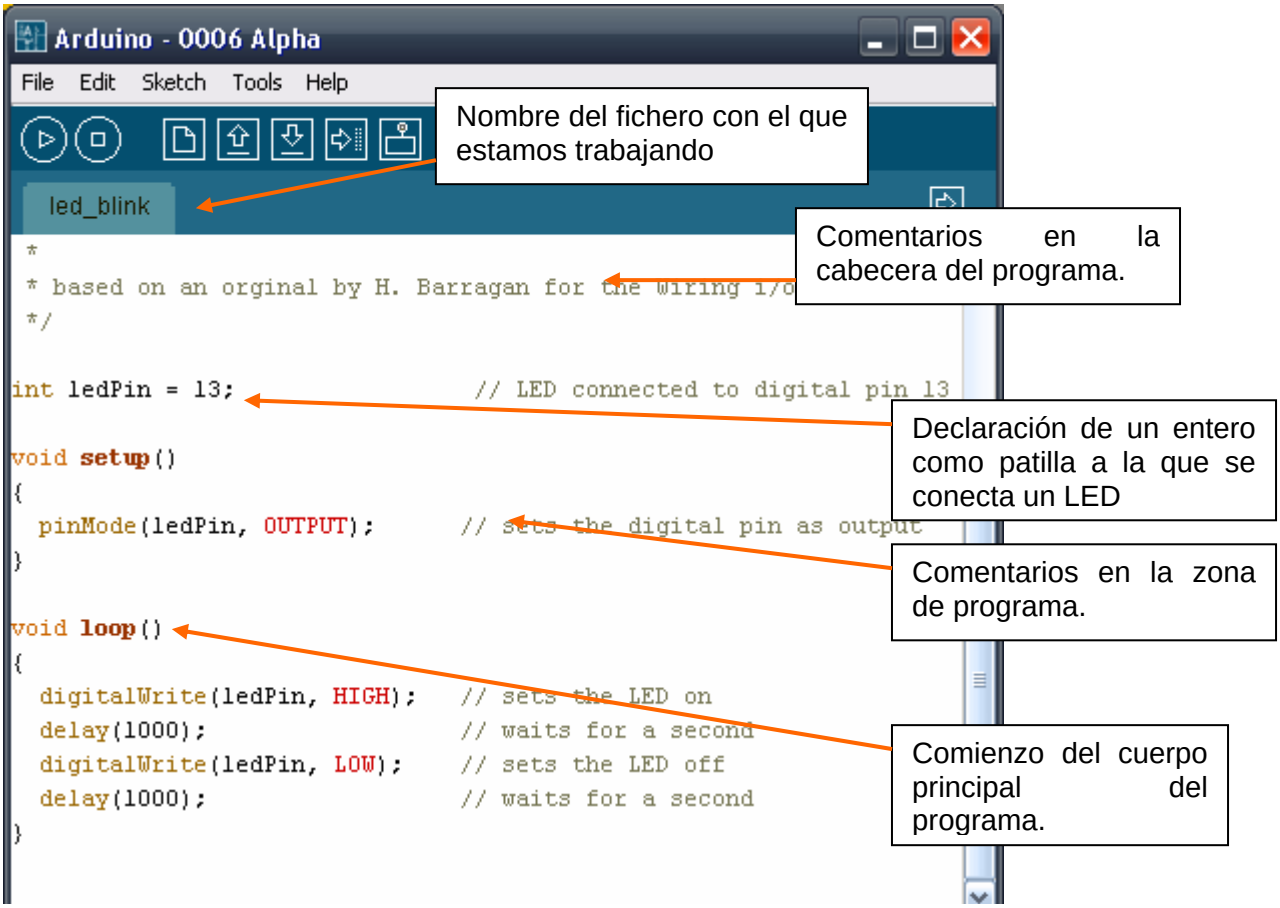
En esta zona teclearemos el código del programa en un lenguaje similar al C.

Monitor del puerto serie
Velocidad de transmisión Área de entrada de datos Botón de envío

Área de datos recibidos por la placa.

El procedimiento es similar en todos los casos, tras editar el código teniendo en cuenta el hardware que hemos conectado a la placa y en que interfaces lo hemos hecho, procederemos a subir el código a la placa, grabando de nuevo la memoria de programa del ATMEL.

La sintaxis del lenguaje es muy similar a C y es muy sencillo de manejar. Para introducir la estructura general de un programa he optado por tomar un código de ejemplo que he encontrado en el entorno de desarrollo: un LED parpadeante.



Como hemos podido apreciar en el ejemplo anterior, la sintaxis es muy sencilla e intuitiva.

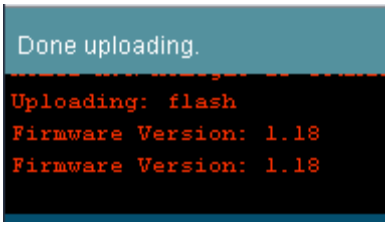
Los programas suelen tener varias zonas relevantes, entre las que destacan la zona de setup de la placa, donde declaramos el modo de trabajo de los pines y el tipo de interfaz que es (entrada, salida digital, salida PWM). La segunda zona, es la del cuerpo principal del programa.

En este código se pueden ver varias instrucciones muy útiles para el programador. En el cuerpo principal del programa se hace uso de una salida digital, poniéndola a nivel alto `digitalWrite(ledPin, HIGH)`. Esta función es muy habitual, ya que el uso de las salidas digitales está presente en muchos programas. La segunda función utilizada en el programa es `delay`, a la cual se le pasa como argumento el número de milisegundos que tiene que esperar el programa.

Este programa es bastante simple y muy ilustrativo y hemos decidido pasarlo a la placa para ver el funcionamiento.

Una vez conectada la placa y pulsado el reset para borrar el código, pinchamos en el botón, que compila y sube el código hexadecimal a la placa. Durante esta fase el microcontrolador se grabará y veremos como los LED's TX y RX parpadean ya que existe comunicación entre el PC y la placa.

Una vez terminada esa fase, aparecerá el siguiente mensaje en la zona de resultados de compilación:

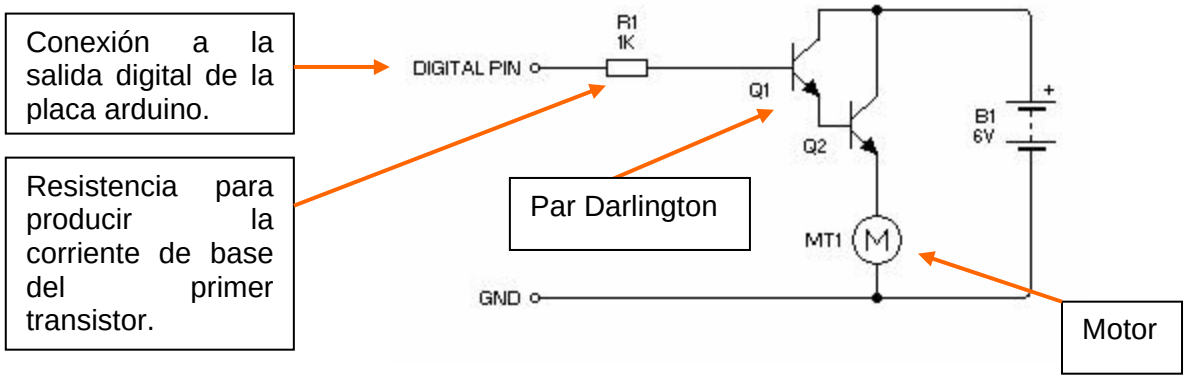


Pasada esta fase podemos ver los resultados en la placa.

5.- Ejemplo de utilización de la placa: control de un motor DC y comunicación serie con el ordenador

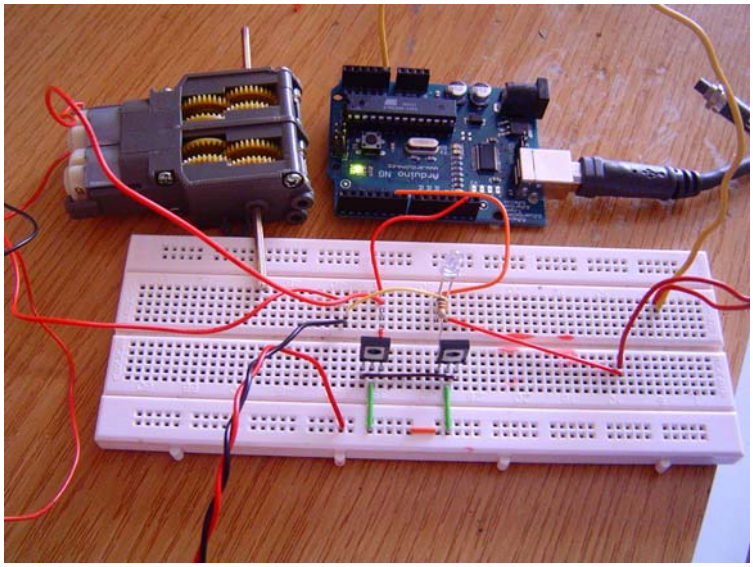
Para probar algo más complejo hemos decidido utilizar un programa que viene de ejemplo en el entorno de desarrollo. Este programa nos permite controlar un motor de corriente continua y controlar su encendido y apagado, retardando su ciclo de puesta en marcha de nuevo tras un periodo de tiempo que proviene de la lectura de un pulsador conectado a la entrada analógica 0.

El problema de controlar un motor tiene un atractivo añadido, y es que presenta una impedancia mayor que la que nos podemos encontrar al intentar encender un LED, o pasar una señal a otros dispositivos lógicos. Esto lo podemos solucionar adaptando la impedancia mediante una etapa de potencia adecuada. Habíamos pensado solucionar esto utilizando un relé que controlase un pequeño circuito de encendido, pero para trabajar con tensiones tan pequeñas hemos recurrido al siguiente circuito que se basa principalmente en dos transistores configurados como par Darlington (de esta forma se multiplican las betas de los transistores, y por lo tanto sus ganancias).



Además de este circuito, hemos conectado un pulsador a la entrada analógica 0 de la placa y un led para resaltar el tiempo de encendido.

La entrada analógica en abierto presenta una resistencia de 1,7KOhm, y cuando está apretado el pulsador que está a su entrada, la resistencia que hay entre ésta y el pin gnd es de 0 ohm lógicamente. Este intervalo de resistencias es en el que nos moveremos para poder variar el tiempo de delay del motor.



Esta es la vista final del circuito, con la etapa de potencia, el motor, la placa, y en la parte inferior izquierda los dos cables que van a la fuente de alimentación.

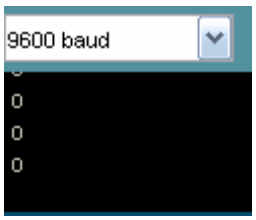
Código que controla el motor:

```
int motorPin = 6;    // motor conectado al pin 6
int value = 0;       // variable para almacenar la lectura del pulsador
int potPin = 0;      // pin en el que se conecta el pulsador

void setup() {
  pinMode(motorPin, OUTPUT);   // declaración del pin 6 como una salida
  Serial.begin(9600);           // conexión por puerto serie a 9600 baudios
}

void loop() {
  // lectura del pulsador y almacenamiento mediante la instrucción analogRead
  value = analogRead(potPin);

  Serial.println(value); // mandamos al ordenador el valor leido

  digitalWrite(motorPin,HIGH); // encendido del motor
  delay(50); // pausa de 50 miliseguindos
  digitalWrite(motorPin,LOW); // apagado del motor

  // retardamos la ejecución del siguiente ciclo del programa durante el tiempo
  // especificado por la lectura del pulsador
  delay(value);
}
```

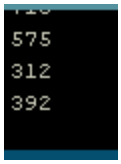
Como ya comentaba antes, este código viene como ejemplo en el entorno de programación, pero no trae consigo los esquemas ni el conexionado de los elementos al circuito, lo cual puede ser algo confuso para usuarios que no hayan trabajado anteriormente con este tipo de placas.

Hay un video del funcionamiento del circuito en la carpeta de nombre *control de motor + led*.

Tras poner en funcionamiento el motor podemos ver como va cambiando el retardo de encendido del motor en función de la pulsación o no del interruptor. Estos datos son transmitidos al ordenador por el bus de serie.



Retardo en milisegundos del encendido del motor. En el primer caso es 0 por que el pulsador esta siendo apretado.

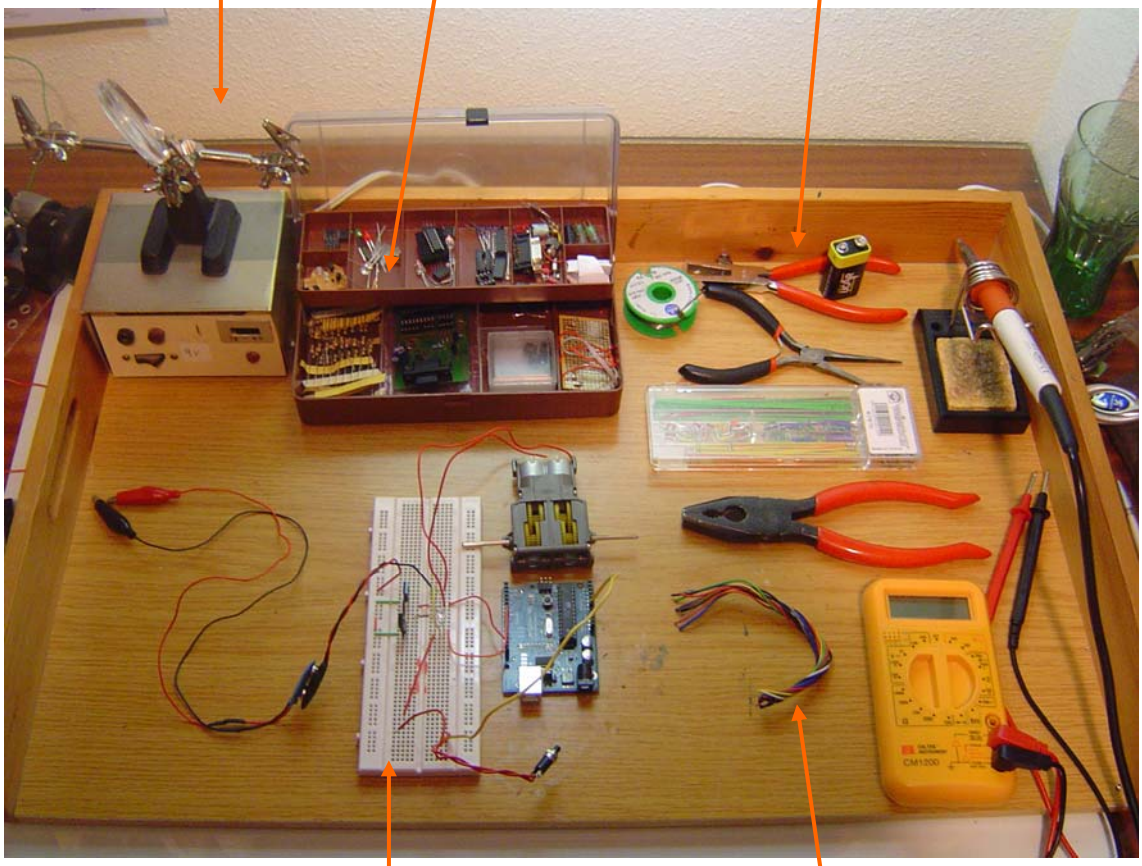


6.-Materiales utilizados

Fuente de alimentación de 9V, de esta forma no gastaremos pilas mientras hacemos pruebas. Lupa con pinzas de soporte.

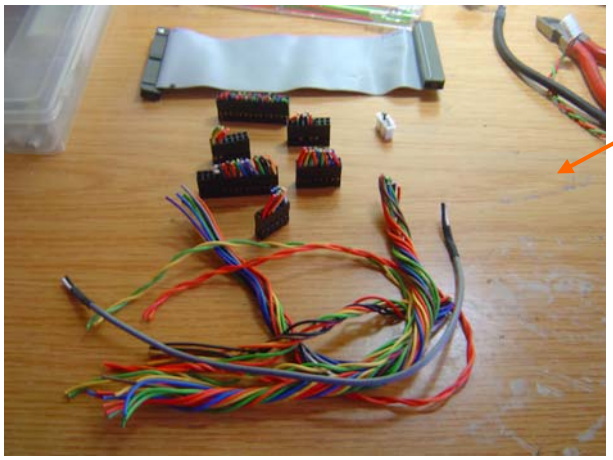
Componentes electrónicos diversos. Entre ellos los transistores de potencia, LED's y resistencias.

Herramientas necesarias: alicates de punta fina, universales y pelacables, soldador de 25W, estaño y téster.
Juego de puentes.



Placa de prototipos, placa arduino, motor de corriente continua, conectores para la alimentación.

Los cables de antiguos conectores son bastante útiles para utilizarlos en nuestros montajes.



7.- Bibliografía y materiales consultados:

USB to UART Bridge - FT232RL

- http://www.ftdichip.com/
- http://www.sparkfun.com/commerce/product_info.php?products_id=650

Datasheets:

- Atmega8 8-bit AVR with 8K Bytes In-System Programmable Flash
- BCD 135 NPN transistor

Libros:

- "123 Robotics experiments for the evil genius", Mike Predko, Tab robotics series, Mc Graw Hill
- "Principios de electrónica" Albert Paul Malvino, Mc Graw Hill
- "Programming robots controllers" Robot DNA Series, Mike Predko, Tab robotics series, Mc Graw Hill

Recursos de Internet:

- "Guia profesores: Apéndice3: Drivers para pequeños motores de contínua"
http://www.arduino.cc/es/Gu%edaProfesores/Apendice3
- "Descripción de arduino"
http://www.arduino.cc/es/Metodolog%eda/Descripcion
- "Getting started with arduino, beta version"