

# Taller de Desarrollo de Sistemas Domóticos Basados en Arduino

José Luis Poza Luján. Sergio Sáez Barona

## Actividad 1. Introducción a Arduino

### Contenido

1	Introducción.....	1
1.1	Justificación .....	1
1.2	Objetivos.....	1
1.3	Descripción de la actividad .....	1
2	Instalando los drivers.....	2
3	Conociendo el IDE.....	3
3.1	Iniciando el programa y ventana principal .....	3
3.2	Menú de la aplicación.....	4
3.3	Barra de herramientas.....	7
4	Software relacionado con Arduino.....	7
4.1	Simuladores .....	7
5	Conclusiones.....	8

### 1 Introducción

#### 1.1 Justificación

Para poder trabajar con Arduino es necesario instalar los drivers y conocer el entorno integrado de desarrollo (IDE), por ello se plantea la actividad en la que el alumno conozca los detalles de la instalación del software. Además se revisará con todo detalle las opciones que proporciona el IDE Arduino.

#### 1.2 Objetivos

El objetivo principal de la actividad es conocer el entorno de trabajo de Arduino, lo que implica aprender a instalar los drivers y aprender el entorno. Además se plantea, como objetivo secundario, concienciarse acerca del software existente relacionado con Arduino.

#### 1.3 Descripción de la actividad

La actividad comienza con unos ejercicios donde se explican los detalles de la instalación de los drivers y de conexionado de la placa al PC. A continuación se revisarán todos los aspectos del entorno de programación Arduino. Finalmente se revisará el software relacionado con Arduino.

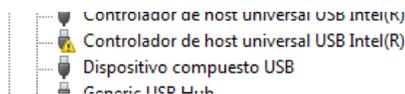
## 2 Instalando los drivers

La instalación de los drivers de Arduino es muy sencilla, dado que se trata de un interfaz de comunicaciones USB. Si los drivers no están instalados, cuando se conecte el Arduino al PC con Windows7 aparecerá el mensaje de nuevo hardware localizado y la ventana de selección del software.



En este caso se deberá seleccionar “Instalar desde una lista o ubicación específica (avanzado)” y seleccionar la carpeta “Drivers” de la ubicación donde se haya descargado el IDE de Arduino. Los drivers se instalan sin problemas ya que el Arduino se ve como un puerto COM más del sistema.

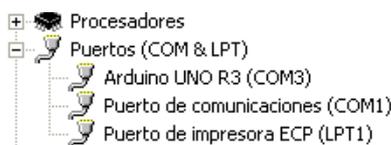
Sin embargo, el driver USB puede colisionar con algunos escasísimos drivers de USB (como antiguos escáneres), en tal caso habrá que actualizar el driver USB que haya instalado. Para ello habrá que ir al “Panel de Control”, y seleccionar el “Administrador de Dispositivos” en Windows 7 y en Windows XP por medio del icono de “Sistema” (pestaña “hardware” → botón “Administrador de Dispositivos”). Una vez abierto el administrador de dispositivos, aparecerá alguno de los controladores USB con un icono de advertencia.



Se deberá hacer clic el dispositivo con el botón derecho y seleccionar la opción de menú “Actualizar software de controlador”



Para comprobar que los drivers de Arduino están bien instalados se debe buscar, dentro del administrador de dispositivos los puertos (COM & LPT), ahí deberá aparecer como puerto COM el Arduino Uno



**Ejercicio 1.** Comprobar que los drivers del Arduino Uno están instalados en el ordenador. Si no están, entonces instalarlos. Comprobar la configuración de puerto que se tiene para poder emplearla posteriormente en el código de la actividad.

Puerto COM	
Bits por segundo	
Bits de datos	
Paridad	
Bits de parada	
Control de Flujo	

**Ejercicio 2.** Desconectar el Arduino del PC y comprobar si en el “Administrador de dispositivos” continúa el Arduino Uno. En caso de no aparecer, ¿qué implicaciones tiene que se le asigne un puerto cada vez que se conecte?



**IMPORTANTE:** los conectores de la placa de Arduino están soldados, pero no atornillados a una caja que los proteja. Junto a las malas conexiones, el manejo inadecuado de los conectores es la mayor causa de averías en los Arduino. Por ello para conectar y desconectarlo del PC se debe hacer por medio del conector USB del PC, no el del Arduino.

## 3 Conociendo el IDE

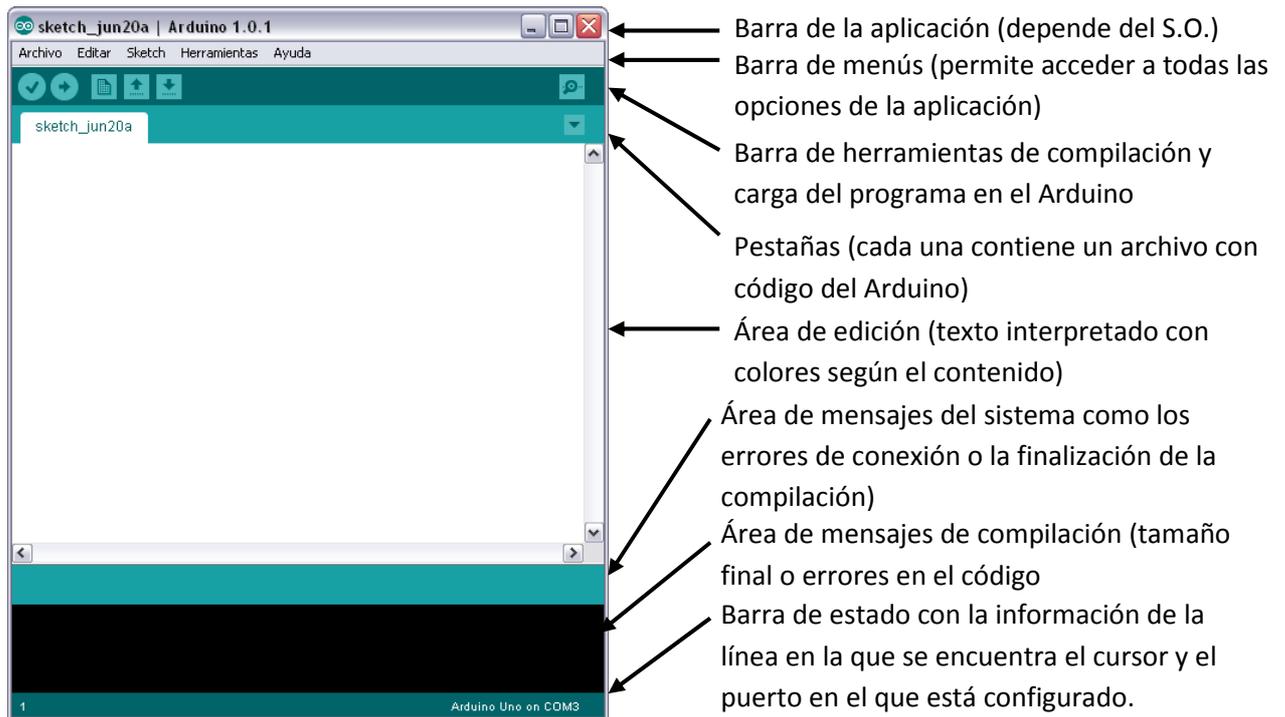
### 3.1 Iniciando el programa y ventana principal

Aunque es posible programar el Arduino desde una gran cantidad de entornos (Eclipse, Visual Studio) los desarrolladores proporcionan un sencillo, pero eficiente, entorno de programación basado en Processing. El entorno se puede conseguir directamente de la página de Arduino (<http://arduino.cc/en/Main/Software>). Se recomienda emplear la página en inglés para acceder a las últimas versiones.

El entorno Arduino no precisa instalación, se lanza desde la misma carpeta en la que se haya descargado, haciendo doble clic en “arduino.exe”.



Una vez se ha iniciado el entorno, aparecerá la ventana principal, que contiene las siguientes partes principales (luego se verán con detalle cada una de ellas)



### 3.2 Menú de la aplicación

Las opciones que tiene el menú de Arduino 1.0.1 (junio 2012) son las siguientes. Empezando por el menú “Archivo”.

- Archivo
  - Nuevo. Genera un nuevo sketch, por defecto en una ventana nueva
  - Abrir. Abre un sketch a partir de un archivo del disco
  - Sketchbook. Abre la carpeta “Arduino” donde se almacenan (por defecto) los sketches que se vayan editando
  - Ejemplos. Muestra los ejemplos básicos de funcionamiento de Arduino
  - Cerrar. Cierra la actual ventana con el sketch
  - Guardar. Guarda el sketch actual con el nombre por defecto
  - Guardar Como. Guarda el sketch con un nombre diferente.
  - Cargar. Compila el sketch y lo carga en la placa Arduino que se haya seleccionado por el puerto correspondiente.
  - Cargar usando programador. Carga el sketch usando un programador diferente al de Arduino (el programador debe estar conectado al puerto USB)
  - Configuración de página. Configura los márgenes y la visualización de la página para imprimir.
  - Imprimir. Imprime la página del sketch
  - Preferencias. Permite configurar el comportamiento de la aplicación. También da acceso directo al archivo de configuración (con más características configurables)
  - Salir. Sale de la aplicación

**Ejercicio 3.** Abrir el menú Archivo→Ejemplos→Basics→Blink y cargarlo en la placa Arduino. Comprobar el funcionamiento observando el resultado en la ventana de la aplicación y en la placa Arduino. ¿Funciona? ¿Qué LEDs se encienden o apagan durante el proceso?

**Ejercicio 4.** Abrir el menú Archivo→Preferencias y revisar las opciones. Seleccionar las opciones de “Mostrar resultado detallado durante: compilación y carga. ¿Qué sucede?

Las opciones del menú de edición son las siguientes.

- Editar
  - Deshacer. Deshace la última acción de edición.
  - Rehacer. Rehace la última acción de edición.
  - Cortar. Corta (elimina) el texto seleccionado para ser pegado en otra posición.
  - Copiar. Copia (sin eliminar) el texto seleccionado para ser pegado en otra posición del sketch.
  - Copiar para el foro. Copia el texto en formato maquetado y coloreado adecuado para ser publicado en el foro de Arduino.
  - Copiar como HTML. Copia el código, incluyendo el formato (colores, tabulados, etc.) con etiquetas HTML adecuado para ser publicado en Web.
  - Pegar. Pega el contenido del portapapeles en la posición actual del cursor.
  - Seleccionar todo. Selecciona todo el código del sketch.
  - Comentar/Descomentar. Añade/Quita los símbolos de los comentarios (‘//’) en cada una de las líneas del sketch.
  - Incrementar margen. Añade una tabulación a cada una de las líneas del texto seleccionado.
  - Reducir margen. Quita una tabulación a cada una de las líneas del texto seleccionado.
  - Buscar. Abre la ventana de búsqueda (contiene más opciones que el menú)

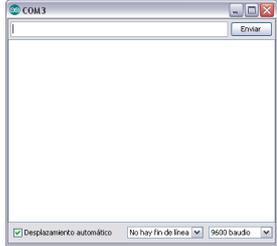


- Buscar siguiente. Busca la siguiente aparición de la cadena buscada a partir de la posición actual del cursor (búsqueda “hacia delante”)
- Buscar anterior. Busca la anterior aparición de la cadena buscada a partir de la posición actual del cursor (búsqueda “hacia atrás”)
- Utilizar actual selección para buscar. Emplea el texto seleccionado para buscar en el sketch

Sobre las búsquedas, cabe destacar que las búsquedas se realizan en la actual ventana y pestaña, no buscando en otras pestañas ni en otras ventanas.

**Ejercicio 5.** Algunas de las teclas rápidas (combinaciones de teclado que realizan una acción del menú) que están definidas en Arduino, no son las habituales. Localizarlas.

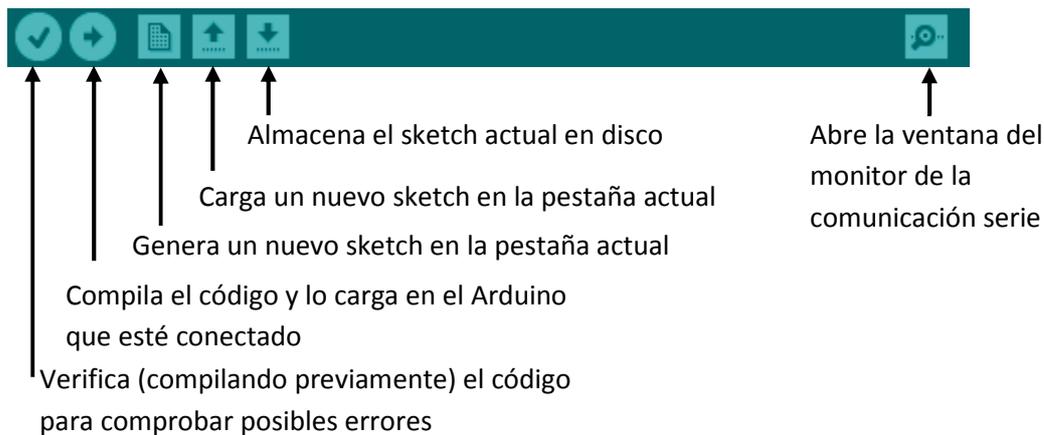
Las opciones del resto de menús son las siguientes.

- Sketch
    - Verificar/compilar. Comprueba la sintaxis del código (llaves bien cerradas, etc.) y seguidamente (en caso de ser correcta la comprobación anterior) compila el código. No carga el código en el Arduino (para ello hay que emplear la opción del menú Archivo→Cargar)
    - Mostrar la carpeta del sketch. Muestra la carpeta donde está almacenado el sketch que se está editando actualmente (de gran utilidad para comprobar duplicados)
    - Agregar Archivo. Abre un archivo del disco y lo añade una nueva pestaña.
    - Importar Librería. Incluye la cabecera que permite importar las librerías que estén en el directorio “Libraries” de la carpeta donde se encuentra el IDE
  - Herramientas
    - Formato automático. Realiza labores de formato automático del código. Añade las tabulaciones donde son necesarias, iguala llaves y tareas similares. De gran utilidad cuando se pega código copiado de Internet o de archivos de texto.
    - Archivar el sketch. Genera un archivo comprimido con todos los archivos de código pertenecientes al sketch.
    - Reparar codificación y recargar. Limpia el código de caracteres no ASCII
    - Monitor Serial. Abre la ventana del monitor serie que permite ver las transferencias de caracteres entre el Arduino y el ordenador por el puerto serie
- 
- Tarjeta: permite seleccionar la tarjeta Arduino con la que trabajar
  - Serie: permite seleccionar el puerto serie para las comunicaciones
  - Programador. Selecciona el cargador de código que se empleará para enviarlo al Arduino (los cargadores se encuentran en la carpeta “hardware\arduino\bootloaders” del directorio donde se encuentre el IDE)
  - Grabar secuencia de inicio. Carga la secuencia de “boot” en el Arduino (precisa de un cargador en el USB)
- Ayuda. Acceso a los archivos de ayuda en red y la información de la versión del IDE que se está empleando

**Ejercicio 6.** Para probar el formato automático, abrir el bloc de notas y copiar el siguiente texto “void setup(){{{{{}}}} void loop(){{}}”. Ir al entorno Arduino y crear un nuevo sketch. Pegar el nuevo código y aplicar la opción de “Formato Automático”. ¿Qué sucede?

### 3.3 Barra de herramientas

La barra de herramientas del IDE de Arduino permite el acceso directo a las funciones más habituales, en concreto a las siguientes.



## 4 Software relacionado con Arduino

### 4.1 Simuladores

Aunque no es complicada la programación de sistemas basados en Arduino, el uso de simuladores ayuda a disponer de herramientas que permiten obtener un resultado previo de un diseño sin necesidad de disponer de la tarjeta o sin poner en riesgo ésta. Existen una gran cantidad de simuladores y programas de diseño relacionados con Arduino. Los ejemplos más destacados son:

	<p>Virtual BreadBoard (<a href="http://www.virtualbreadboard.com">http://www.virtualbreadboard.com</a>). Simulador de placas con una sección dedicada a Arduino (en sus últimas versiones es de pago)</p>
	<p>Fritzing (<a href="http://fritzing.org/">http://fritzing.org/</a>). Interesante proyecto de automatización en el diseño de sistema. En las últimas versiones dispone de una gran cantidad de componentes, pero no simula.</p>
	<p>Simulator For Arduino (<a href="http://www.arduino.com.au/Simulator-for-Arduino.html">http://www.arduino.com.au/Simulator-for-Arduino.html</a>) Proyecto de los desarrolladores de Arduino. Muy potente, aunque es de pago, el precio (actualmente 4€) es muy adecuado al producto</p>

**Ejercicio 7.** Lanzar las aplicaciones antes mencionadas e implementar el ejemplo “blink”. Exponer las posibles utilidades que tienen los programas anteriores para el diseño de sistemas basados en Arduino.

Otro ejercicio interesante consiste en revisar el software desarrollado (IDEs, etc.) de utilidad para los diseñadores de sistemas basados en Arduino.

**Ejercicio 8.** Revisar las Herramientas de desarrollo que se recopilan en el “playground” de Arduino <http://arduino.cc/playground/Main/DevelopmentTools> y exponer las posibles utilidades que tienen los programas anteriores para el diseño de sistemas basados en Arduino.

## 5 Conclusiones

En esta actividad se ha visto el entorno que rodea a Arduino desde la localización de las partes de la placa “Arduino Uno” hasta el software y documentación que es posible localizar.

**Ejercicio 9.** Autoevaluación. Determinar el grado de utilidad (1: nada, 2: poco, 3: alguno, 4: mucho, 5: imprescindible) obtenido para cada apartado de los desarrollados y el nivel adquirido en el aprendizaje de los mismos (1: ninguno, 2: poco, 3: suficiente, 4: mucho, 5: experto)

Apartado	Utilidad					Nivel adquirido				
	1	2	3	4	5	1	2	3	4	5
Instalación de los drivers										
Conociendo el IDE										
Software relacionado										
Documentación y referencias										

**Ejercicio 10.** Ampliaciones y trabajo personal. Determinar qué posibles inquietudes personales se pueden realizar relacionadas con la actividad.