

Control por Computador

Manual de la Práctica 4: Control de iluminación mediante Arduino



Jorge Pomares Baeza

Francisco Andrés Candelas Herías

Grupo de **Innovación Educativa en Automática**



Universitat d'Alacant
Universidad de Alicante

© 2009 GITE – IEA

Introducción

En esta práctica se empleará la placa arduino para realizar el control de la iluminación del entorno. Para ello en primer lugar se describirá el funcionamiento básico de la placa.

Objetivos

- Aprender el funcionamiento básico de la placa arduino.
- Implementar un controlador PID real en una placa con entradas/salidas.

1. La placa Arduino USB Duemilanove

En esta práctica se empleará la placa Arduino USB Duemilanove. Se trata de una placa open hardware por lo que su diseño es de libre distribución y utilización, que incluso podemos construirnos nosotros mismos (En la Figura 1 se observa el aspecto de la placa). En la web de la asignatura existe un apartado dedicado a arduino donde puede encontrarse mucha información relativa a la placa:

<http://blogs.ua.es/control>

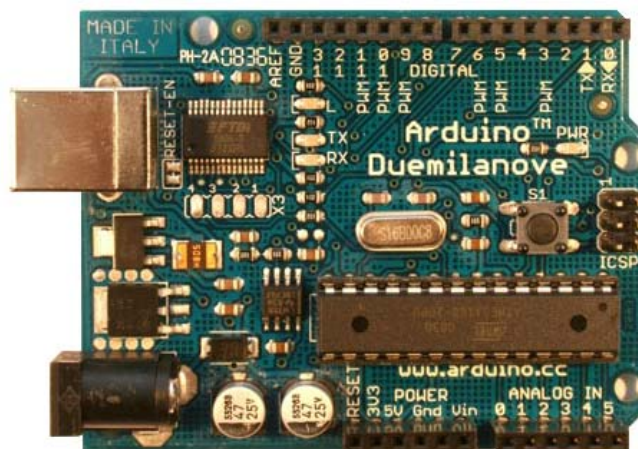


Figura 1. Aspecto de la placa Arduino Duemilanove.

El programa se implementará haciendo uso del entorno de programación propio de arduino y se transferirá empleando un cable USB. Si bien en el caso de la placa USB no es preciso utilizar una fuente de alimentación externa, ya que el propio cable USB la proporciona, para la realización de algunos de los experimentos prácticos sí que será necesario disponer de una fuente de alimentación externa ya que la alimentación proporcionada por el USB puede no ser suficiente. El voltaje de la fuente puede estar entre 6 y 25 Voltios.

2. Entorno de desarrollo

Para programar la placa es necesario descargarse de la página web de Arduino el entorno de desarrollo (IDE). Se dispone de versiones para Windows y para MAC, así como las fuentes para compilarlas en LINUX. En la Figura 2 se muestra el aspecto del entorno de programación. En el caso de disponer de una placa USB es necesario instalar los drivers FTDI. Estos drivers vienen incluidos en el paquete de Arduino mencionado anteriormente. Existen en la web versiones para distintos sistemas operativos.

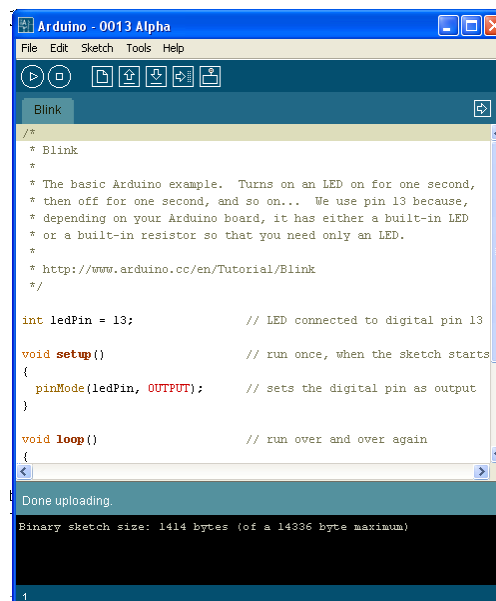


Figura 2. Entorno de desarrollo.

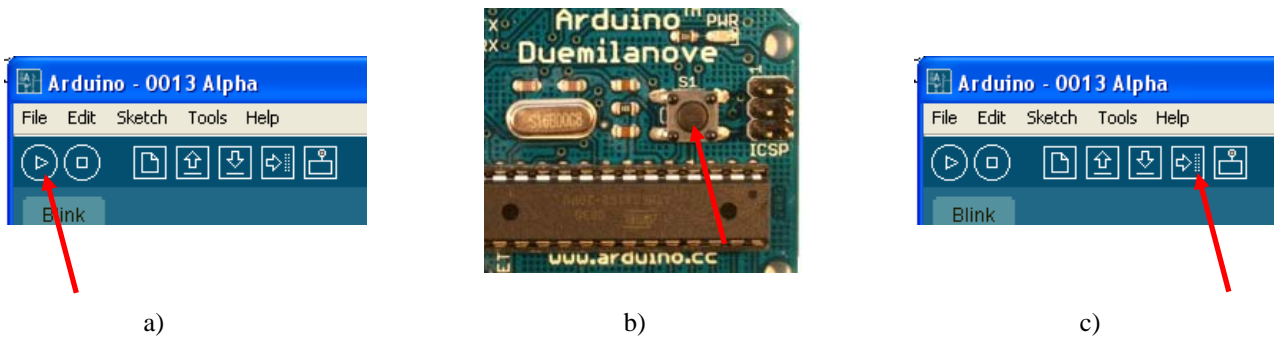
Lo primero que tenemos que hacer para comenzar a trabajar con el entorno de desarrollo de arduino es configurar las comunicaciones entre la placa Arduino y el PC. Para ello deberemos abrir en el menú “Tools” la opción “Serial Port”. En esta opción deberemos seleccionar el puerto serie al que está conectada nuestra placa. En Windows, si desconocemos el puerto al que está conectado nuestra placa podemos descubrirlo a través del Administrador de dispositivos (Puertos COM & LPT/ USB Serial Port).

El primer paso para comprobar que todo lo que hemos hecho hasta ahora está bien y familiarizarnos con el interfaz de desarrollo, es abrir uno de los ejemplos. Se recomienda abrir el ejemplo “Blink”. Para ello debemos acceder a través del menú File → Sketchbook → Examples → Digital → Blink.

El ejemplo “Blink” lo único que hace es parpadear un LED que está colocado en el pin número 13 de la placa. Vamos a ver qué hay que hacer para subir el programa a la placa Arduino. Primero comprobamos que el código fuente es el correcto. Para ello pulsamos el botón de verificación de código que tiene forma de triángulo inclinado 90 grados (Figura 3.a). Si todo va bien deberá aparecer un mensaje en la parte inferior de la interfaz indicando “Done compiling”. Una vez que el código ha sido verificado procederemos a cargarlo en la placa. Para ello tenemos que pulsar el botón de reset de la placa (Figura 3.b) e inmediatamente después pulsar el botón que comienza la carga (Figura 3.c).

Durante la carga del programa, en la placa USB, se encenderán los LED que indican que se están enviando y recibiendo información por el puerto serie: TX/RX. Si todo se ha realizado correctamente debe aparecer el

mensaje “Done uploading”. Ahora tan sólo queda esperar unos 8 segundos aproximadamente para comprobar que todo ha salido bien. Si el led colocado en el pin 13 de la placa se enciende y se apaga cada segundo entonces todo ha ido bien. Por fin tenemos todo listo para empezar a trabajar con la placa Arduino.



a) Compilar programa. b) Botón de reset. c) Transferir programa a la placa.

2. Programa a desarrollar

En esta práctica se va a emplear Arduino para controlar la iluminación en un entorno. Para ello se empleará un sensor de luz LDR. La información de luz será leída a través de Arduino y, dependiendo del valor de la referencia o cantidad de luz deseada se determinará la intensidad de luz a aplicar una bombilla. El esquema general del sistema de control se representa en el diagrama de bloques de la Figura 4.

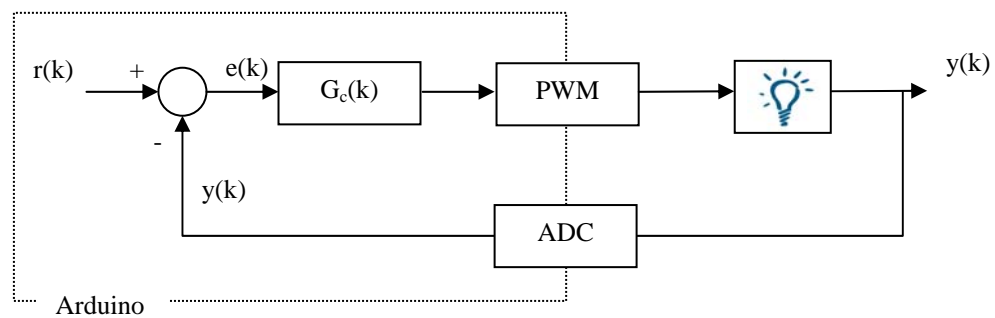


Figura 4. Esquema de control a implementar.

El controlador $G_c(k)$ implementado será un regulador PID que generará una señal PWM para controlar la intensidad de la bombilla. Por lo tanto, se habrá de emplear los pines 3, 5, 6, 9 ó 10 para determinar en cada momento la intensidad de la bombilla en función del ancho de pulso generado por dicho pin. Este ancho de pulso adquiere valores de entre 0 a 255. Un valor 0 genera 0 V. en el pin especificado (pulso nulo) y 255 genera 5 V. (anchura total). Para valores de 0 a 255, el pin alterna rápidamente entre 0 y 5 V., cuanto mayor sea el valor, más a menudo el pin se encuentra en HIGH (5 V.). Por ejemplo, un valor de 64 será 0 tres cuartas partes del tiempo y 5 Voltios una cuarta parte. Un valor de 128 será 0 la mitad del tiempo y 5 V. la otra mitad. Un valor de 192 será 0 V. una cuarta parte del tiempo y 5 V. tres cuartas partes. De esta manera, a mayor ancho de pulso mayor será la intensidad de la bombilla.

El primer paso para desarrollar la práctica consiste en realizar con la placa Arduino el montaje que se muestra en la Figura 5.

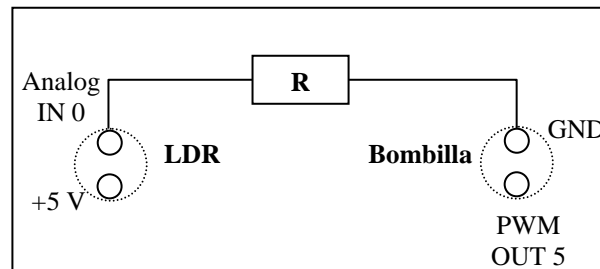


Figura 5. Montaje en la placa Arduino.

Una vez realizado el montaje se inicializarán las constantes del controlador y la referencia. Para determinar la luz de referencia se puede hacer pruebas hasta determinar con el sensor la luz deseada (alrededor de 500). A partir de la referencia se calculará la señal de error y a partir de esta el controlador. Como se ha indicado anteriormente la salida del controlador debe ser una señal PWM con un valor de 0 a 255.

Se habrá de presentar un informe con el código desarrollado y el proceso de ajuste de las constantes del controlador hasta obtener el comportamiento deseado.