



Diseño detallado de las sesiones de laboratorio con Arduino

*Miguel Ángel Rubio Escudero- Dpto. Ciencias de la Computación
Universidad de Granada*

Índice

1. Introducción	2
2. Ejercicio 1: Variables	5
3. Ejercicio 2: Percibiendo el mundo exterior	8
4. Ejercicio 3: Actividades adicionales.....	10
Bibliografía	11



1. Introducción

En este apartado vamos a proceder a describir el conjunto de actividades que se han desarrollado para las clases de prácticas. Comenzaremos con una descripción de la plataforma que se ha diseñado para realizar las prácticas y seguiremos con una breve descripción de cada demostración y el software necesario.

La plataforma que se ha desarrollado para las clases prácticas está basada en la tarjeta Arduino. En el caso de las prácticas es necesario montar la tarjeta y la placa de prototipado en una plataforma común. Esto evita que haya tirones en los cables y disminuye la aparición de errores y averías durante las prácticas.

La plataforma diseñada para la realización de prácticas se basa en la primera plataforma utilizada en clases de teoría. La razón es sencilla: nuestra experiencia indica que los estudiantes suelen quedarse con bastantes preguntas sobre las demostraciones realizadas en clase. Podemos aprovecharnos de esta curiosidad no satisfecha si les presentamos en prácticas experiencias parecidas a las vistas en teoría.

La plataforma que presentamos consta de un fotoresistor LDR-VT90N2 un sensor de temperatura Sparkfun TMP36 y de cinco leds de color rojo cuya descripción se adjunta.



Figura 1-1. Fotoresistor LDR-VT90N2 y sensor de temperatura Sparkfun TMP36.

619-4937	L-7113SRD-H	5 mm difuso rojo. IF=20mA. VF=1.85V 1000mcd. 30º
----------	-------------	---

Figura 1-2. Características de los LEDs utilizados

Resistencias a utilizar con los distintos LEDs

- Led rojo difuso -> 470 ohmios

El diseño incluye algunos aspectos que facilitan la labor docente. Se intenta ser sistemático con los colores: rojo para la potencia, marrón para la tierra y amarillo para la recogida de información. Uno de los leds tiene un cable de mayor longitud y de color azul. Este cable permite realizar algunas experiencias que demuestra a los alumnos que la mecánica subyacente es sencilla.

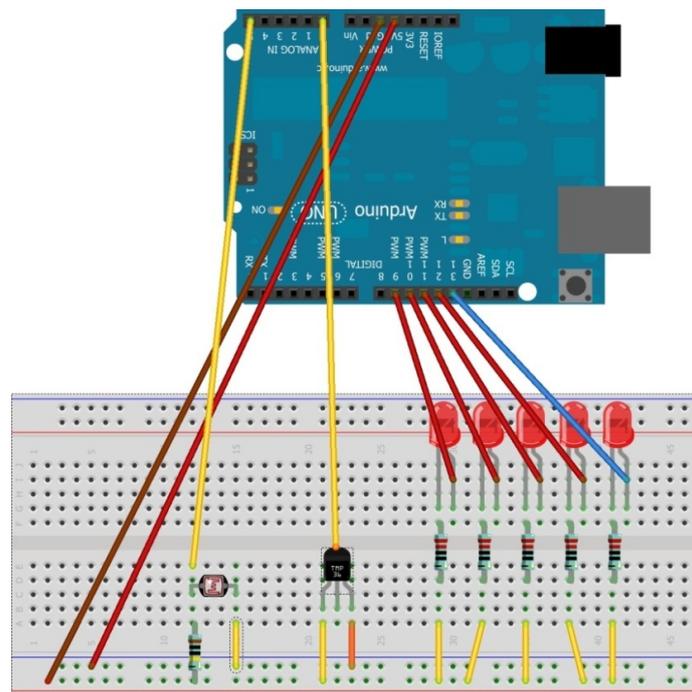


Figura 1-3. *Diagrama de la plataforma*

Los ejercicios que se presentan se han diseñado para presentar distintos conceptos de la programación y el uso de ordenadores a través de la tarjeta Arduino. Nuestro objetivo es que los estudiantes vean cómo lo aprendido en teoría sobre programación les permiten implementar comportamientos llamativos en dispositivos reales.

El diseño busca apoyar al alumno en sus distintas fases de aprendizaje –técnica conocida como scaffolding (Sorva 2012). Aunque las actividades está relacionadas se ha intentado que sean lo más independiente posibles. El profesor puede elegir -en función del tiempo disponible y los conocimientos de sus alumnos- qué actividades incluir en la sesión de laboratorio y cuáles no.



Se presupone que los estudiantes ya están familiarizados con la tarjeta Arduino y con la programación a través de la IDE de Arduino. El responsable docente puede decidir presentar estos principios en clase de teoría, al principio de la clase práctica o en una clase práctica anterior.



2. Ejercicio 1: Variables

Vamos a comenzar la práctica encendiendo y apagando leds. Comenzaremos encendiendo el led 10. Copia el siguiente código, compílalo y cárgalo en Arduino.

```
/*  
Ejemplo del uso de leds  
This example code is in the public domain.  
*/  
  
#include <PhysicalComputingDemonstrations.h>  
  
void setup() {  

```

- Prueba ahora a modificar el código para encender el led 11 en vez del diez.

A continuación vamos a proceder a encender simultáneamente los leds 9,11 y 13.

```
/*  
Ejemplo del uso de leds  
This example code is in the public domain.  
*/  
  
#include <PhysicalComputingDemonstrations.h>  
  
void setup() {  

```

- Prueba ahora a modificar el código para encender los leds 10,11 y 12.

Para apagar el led conectado al pin 9 después de encender los tres el código debería indicar:



```
/*  
Ejemplo del uso de leds  
This example code is in the public domain.  
*/  
  
#include <PhysicalComputingDemonstrations.h>  
  
void setup() {  

```

- Prueba ahora a modificar el código para apagar el led 13.

Para apagar los leds conectados a los pins 9, 11 y 13 el comando sería:

```
/*  
Ejemplo del uso de leds  
This example code is in the public domain.  
*/  
  
#include <PhysicalComputingDemonstrations.h>  
  
void setup() {  

```

- Prueba ahora a modificar el código para encender los leds 9, 11 y 13 y apagar los leds 10,11 y 12. ¿Qué ocurre?

Al menos uno de los leds está conectado a Arduino mediante un cable largo. Desconecta ese cable del Arduino e intenta encender el led. ¿Qué pasa?

Ahora tienes que copiar el código del siguiente programa y guardarlo. ¿Qué crees que hace?



```
/*  
Ejemplo del uso de leds.  
Programa que ejecuta un número de destellos.  
*/  
  
#include <PhysicalComputingDemonstrations.h>  
  
void setup() {  

```

Compíllalo y cárgalo en Arduino. ¿Hace lo que esperabas?

1. Modifica el código de destellos para que en vez de utilizar sólo dos leds utilice todos los leds disponibles (como si fuese una luz de navidad, un cylon o el coche fantástico).
2. Modifica el valor dado en `para(1)` para que las luces se muevan más deprisa.
3. ¿Qué pasa si utilizamos `para()` en vez de `para(1)`?



3. Ejercicio 2: Percibiendo el mundo exterior

Una vez que hemos aprendido como utilizar un programa para encender y apagar leds vamos a ver como percibir el mundo exterior utilizando sensores electrónicos. Para ello vamos a comenzar midiendo la temperatura de la sala.

Para leer la temperatura debemos fijarnos cuál es el pin analógico al cual está conectado nuestro sensor. Podéis verlo en el siguiente diagrama, aunque es mejor que lo miréis directamente en la placa que se os ha dado. Seguramente vuestro sensor de temperatura esté conectados *al pin analógico 0*.

Una vez que ya hemos identificado el pin al que está conectado podemos comenzar a ver cuál es la temperatura. Si el sensor está conectado al pin 0, sería:

```
/*  
  Ejemplo del uso de sensores de tempatura y luz.  
  */  
  
#include <PhysicalComputingDemonstrations.h>  
  
void setup() {  
  Serial.begin(9600); //permite ver la salida por pantalla  
}  
  
void loop() {  
  int pin_temperatura = 0;  
  int temper = lee_temperatura(pin_temperatura);  
  Serial.println(temper); //muestra la temperatura en el monitor serial  
  para(1);  
}
```

Para ver la salida necesitas abrir la ventana del monitor serial. Está en *Herramientas->Monitor Serial*.

Fíjate que la salida de la función `lee_temperatura` va a depender de la temperatura de la sala. Pon ahora tus dedos en la parte negra del sensor de temperatura. Espera unos segundos y vuelve a medir la temperatura. ¿Qué ha pasado? Describe qué resultados has obtenido.

Ahora vamos a proceder a medir la luminosidad en el laboratorio. Para ello vamos a utilizar un sensor que nos da una medida entre cero y uno. Cero será una oscuridad total y uno el máximo de luminosidad.

Para leer la luminosidad debemos fijarnos cuál es el pin analógico al cual está conectado nuestro sensor. Podéis verlo en el siguiente diagrama, aunque es mejor que lo miréis



directamente en la placa que se os ha dado. Seguramente vuestro sensor de temperatura esté conectado al *pin analógico 5*.

Una vez que ya hemos identificado el pin al que está conectado podemos comenzar a ver cual es la luminosidad. Si el sensor está conectado al pin 5, sería:

```
/*  
Ejemplo del uso de sensores de temperatura y luz.  
*/  
  
#include <PhysicalComputingDemonstrations.h>  
  
void setup() {  
  Serial.begin(9600); //permite ver la salida por pantalla  
}  
  
void loop() {  
  int pin_luminosidad = 5;  
  float luz = lee_luminosidad(pin_luminosidad);  
  Serial.println(luz); //muestra la luminosidad en el monitor serial  
  para(1);  
}
```

La salida de la función `lee_luminosidad` va a depender de la luz en la sala. Cubre ahora con tus dedos el sensor de luminosidad. Vuelve a medirla. ¿El nuevo valor es mayor o menor que el anterior? Comenta los resultados.



4. Ejercicio 3: Actividades adicionales

Para terminar, se proponen al alumno una serie de ejercicios para afianzar los conocimientos adquiridos. Así mismo, se pretende que el alumno sea capaz de imaginar posibles aplicaciones reales utilizando lo aprendido hasta ahora.

1. Modifica el código de la función `encendido_automático` para que funcione correctamente. Luego describe cual es su funcionalidad y cómo se consigue implementar dicha funcionalidad.

```
/*  
 Ejemplo del uso de sensores de temperatura y luz.  
*/  
  
#include <PhysicalComputingDemonstrations.h>  
  
void setup() {  
}  
  
void loop() {  
  
 //Programa encendido_automático  
 //Enciende los leds si es de noche  
 //Si hay luz los apaga  
 //-----  
  
 //Veo qué luminosidad hay  
 float luz = lee_luminosidad(5);  
 int leds[5] = {9,10,11,12,13};  
  
 //Si es menor que un valor dado enciendo la farola  
 if (luz < 0.9)  
   enciende_led(leds,5);  
 else //Apago la farola  
   enciende_led(leds,5);  
  
 para (.3); //para unas decimas antes de volver a repetir  
}
```

2. Modifica el programa que usamos para emitir destellos y haz que la frecuencia de los destellos dependa de la luminosidad del ambiente. A menor luminosidad mayor frecuencia de destellos.
3. Analiza el código de la función `encendido_automático` y escribe e indica qué hace. Para ello debes



indicar cuál es su funcionalidad y cómo se consigue implementar dicha funcionalidad.

```
//Saca por led el número que introduzcamos en binario
//Máximo número 31
//Se utilizan los pines del 13 al 9
//13 menos significativo
//9 más significativo
//Para poder hacer el juego de adivinar el método es necesario
//que los leds tengan distintos colores.
//Argumento de entrada
//num: el número que vamos a pasar a binario

void escribe (int num) {

//Los leds con los que voy a trabajar
int rango_led [5]= {9,10,11,12,13};
int val;

for (int cont = 0; cont < 5; cont++){
    val = num%2; //Calculo el resto de dividir por dos
    num = num/2; //Calculo el cociente y lo asigno a num

    //Si la cifra en binario vale 1
    if (val == 1)
        enciende_led(rango_led[cont]); //enciendo el led
    //Si no
    else
        apaga_led(rango_led[cont]); //apago el led
}
}
```

La función se llama con el siguiente comando: `escribe (16)`

4. Escribir una función de nombre `escribe_temperatura` que use las funciones `lee_temperatura` y `escribe`. Esta función lee la temperatura del sensor de temperatura y la escribe en binario utilizando los leds disponibles. Crear una versión alternativa de nombre `lecturaContinua` en la que el Arduino muestre de manera continua la temperatura en los leds.

Bibliografía

Sorva, Juha. 2012. "Visual Program Simulation in Introductory Programming Education."



ugr | Universidad
de Granada



DECSAI



Este obra está bajo una [licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Para ver una copia de esta licencia visita:
<http://creativecommons.org/licenses/by-sa/4.0/>

O manda una carta a:
Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.
