2011

Unidad didáctica con placa arduino

José Antonio Caballero Barba
[Escribir el nombre de la compañía]
24/10/2011

UNIDAD DIDÁCTICA: CONTROL DE CRUCE SEMAFÓRICO

OBJETIVOS

Arduino es una plataforma de hardware de código abierto, basada en una sencilla placa con entradas y salidas, analógicas y digitales, en un entorno de desarrollo que está basado en el lenguaje de programación Processing. Es un dispositivo que conecta el mundo físico con el mundo virtual, o el mundo analógico con el digital.

Vamos hacer un proyecto muy fácil que nos servirá de introducción A la placa Arduino y a su programación, en concreto será el control de un cruce de semáforos, con diodos LEDs.

Para hacerlo dentro del tema de la programación de 4º de la E.S.O. dedicado a SISTEMAS DE CONTROL.

- Conocer el funcionamiento y utilizar una tarjeta controladora.
- Aprender a utilizar los diagramas de flujo al realizar tareas de programación.
- Introducir el concepto de controladora.
- Mostrar cuáles son las principales controladoras disponibles en el aula de Tecnología y en el ámbito educativo.
- Mostrar las conexiones básicas.
- Conocer las interfaces de alguna de las controladoras empleadas en el taller de tecnología.
- Conocer los fundamentos básicos del lenguaje para tarjetas controladoras.
- Presentar el diagrama de bloques de un sistema de control por ordenador.
- Revisar el concepto de señal digital.
- Mostrar las acciones básicas que pueden realizarse con un control de ordenador: accionamiento de diodos luminiscentes LEDs.
- Presentar un sistema sencillo de control por ordenador.

CONCEPTOS

- Control por ordenador.
- Controladoras e interfaces de control.
- Dispositivos de entrada-salida de control.
- Tipos de controladoras centrándonos en concreto en una Arduino.
- Codificación de programas para tarjetas controladoras.
- Interfaces de control y programación.
- Diagramas de flujo.

PROCEDIMIENTOS

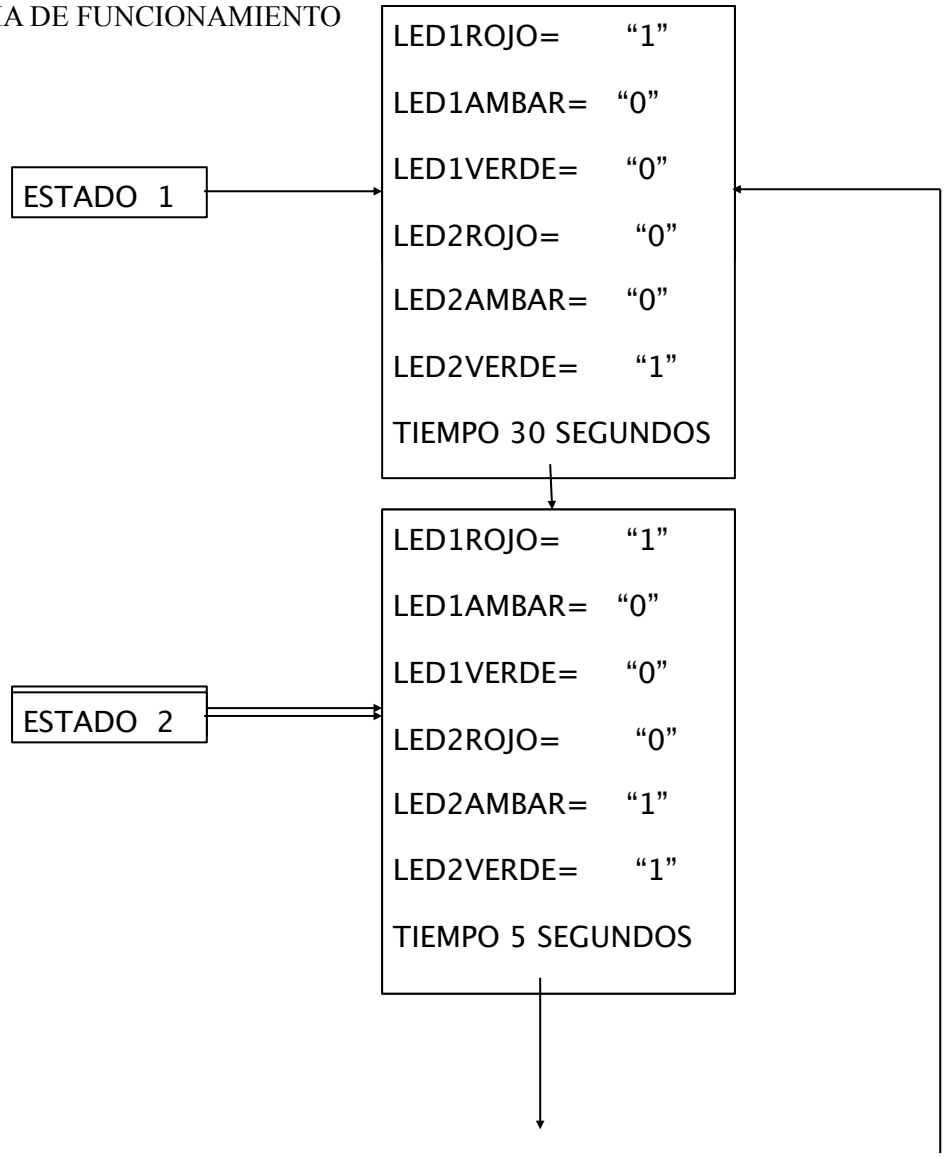
- Utilizar la tarjeta controladora.

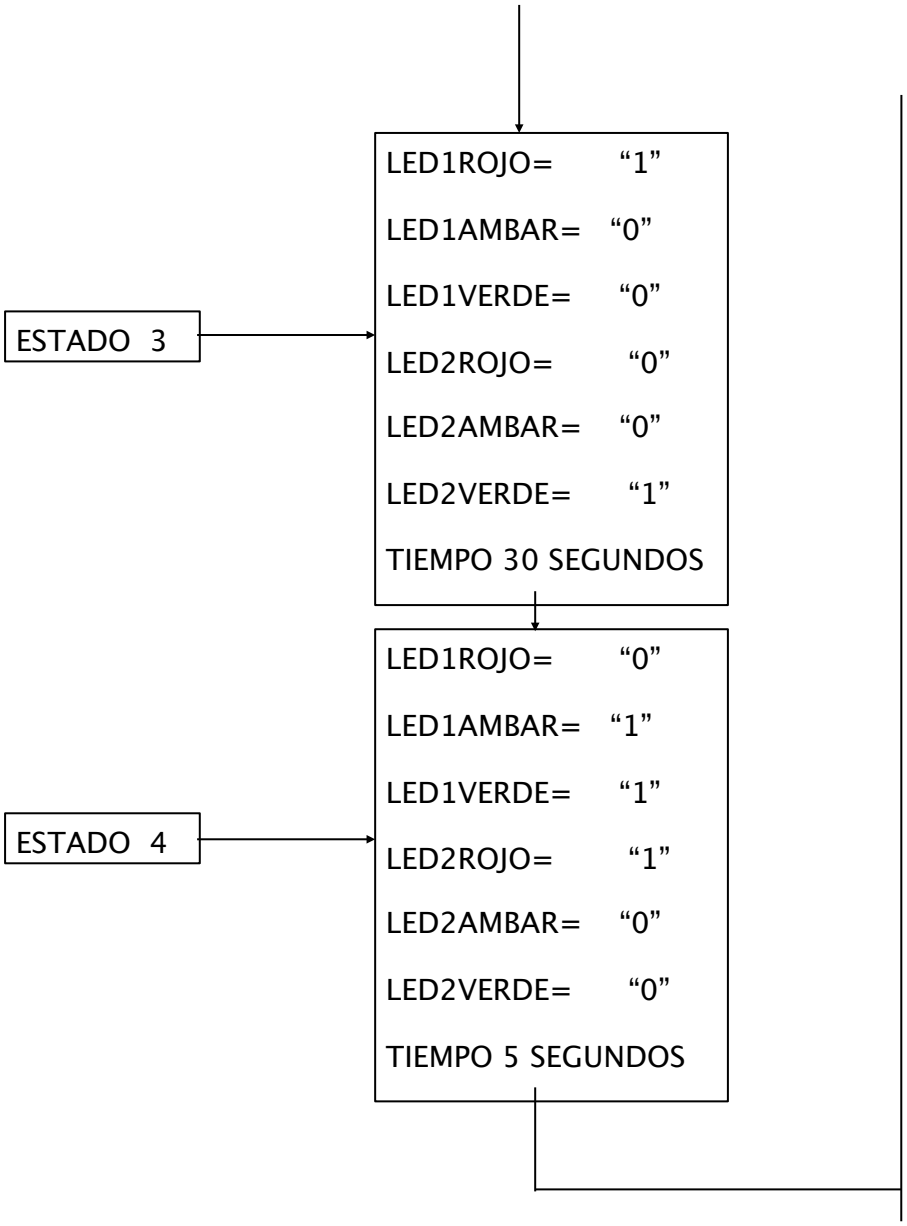
- Interpretar y elaborar de diagramas de flujo.
- Diseñar programa para controlar las entradas y salidas digitales de una controladora.
- Utilizar una controladora para regular el funcionamiento de circuitos eléctricos con la ayuda de un ordenador.
- Interpretar programas sencillos.
- Elaborar programas sencillos en lenguaje para Arduino y utilizarlos a continuación para el control de sistemas.

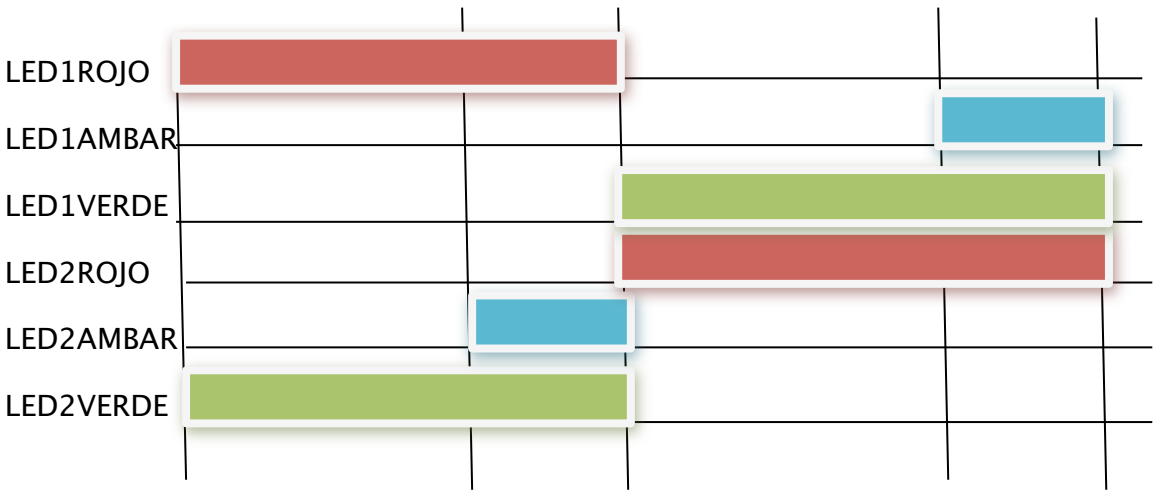
Actitudes

- Gusto por el orden y la limpieza en la elaboración de dibujos y esquemas.
- Apreciar el trabajo complejo y planificado que exige el montaje de sistemas de control.
- Interés por abordar problemas que, a priori, pueden parecer difíciles de solucionar.
- Interés por abordar trabajos en grupo.

DIAGRAMA DE FUNCIONAMIENTO




ESTADO 3
LED1ROJO= "1"
LED1AMBAR= "0"
LED1VERDE= "0"
LED2ROJO= "0"
LED2AMBAR= "0"
LED2VERDE= "1"
TIEMPO 30 SEGUNDOS
ESTADO 4
LED1ROJO= "0"
LED1AMBAR= "1"
LED1VERDE= "1"
LED2ROJO= "1"
LED2AMBAR= "0"
LED2VERDE= "0"
TIEMPO 5 SEGUNDOS


LED1ROJO
LED1AMBAR
LED1VERDE
LED2ROJO
LED2AMBAR
LED2VERDE

PROGRAMA

```
//Crear variables  a las salidas
Int led1Rojo=3;                //semáforo1
Int led1Ambar=4;
Int led1Verde=5;
Int led2Rojo=6;                //Semáforo2
Int led2Ambar=7;
Int led2Verde=8;
Void setup ()  {
pinMode (led1Rojo , OUTPUT);              //Habilitar las salidas digitales
pinMode (led1Ambar , OUTPUT);
pinMode (led1Verde , OUTPUT);
pinMode (led2Rojo , OUTPUT);
pinMode (led2Ambar , OUTPUT);
pinMode (led2Verde , OUTPUT);
}
Void loop ()   {
digitalWrite (ledRojo1 , HIGH);              //Estado1
digitalWrite (ledAmbar1 , LOW);
digitalWrite (ledVerde1 , LOW)     ;
digitalWrite (ledRojo2 , LOW);
digitalWrite (ledAmbar2 , LOW);
digitalWrite (ledVerde2 , HIGH)    ;
delay (30000);                 // 30 segundos



digitalWrite (ledRojo1 , HIGH);              //Estado2
digitalWrite (ledAmbar1 , LOW);
```

```
digitalWrite (ledVerde1 , LOW)     ;
digitalWrite (ledRojo2 , LOW);
digitalWrite (ledAmbar2 , HIGH);
digitalWrite (ledVerde2 , HIGH)    ;
delay (5000);          // 5 segundos


digitalWrite (ledRojo1 , LOW);               //Estado3
digitalWrite (ledAmbar1 , LOW);
digitalWrite (ledVerde1 , HIGH)    ;
digitalWrite (ledRojo2 , HIGH);
digitalWrite (ledAmbar2 , LOW);
digitalWrite (ledVerde2 , LOW)     ;
delay (30000);                 // 30 segundos


digitalWrite (ledRojo1 , HIGH);              //Estado4
digitalWrite (ledAmbar1 , LOW);
digitalWrite (ledVerde1 , LOW)     ;
digitalWrite (ledRojo2 , LOW);
digitalWrite (ledAmbar2 , LOW);
digitalWrite (ledVerde2 , HIGH)    ;
delay (5000);          // 5 segundos
}
```

MONTAJE CONEXIONES

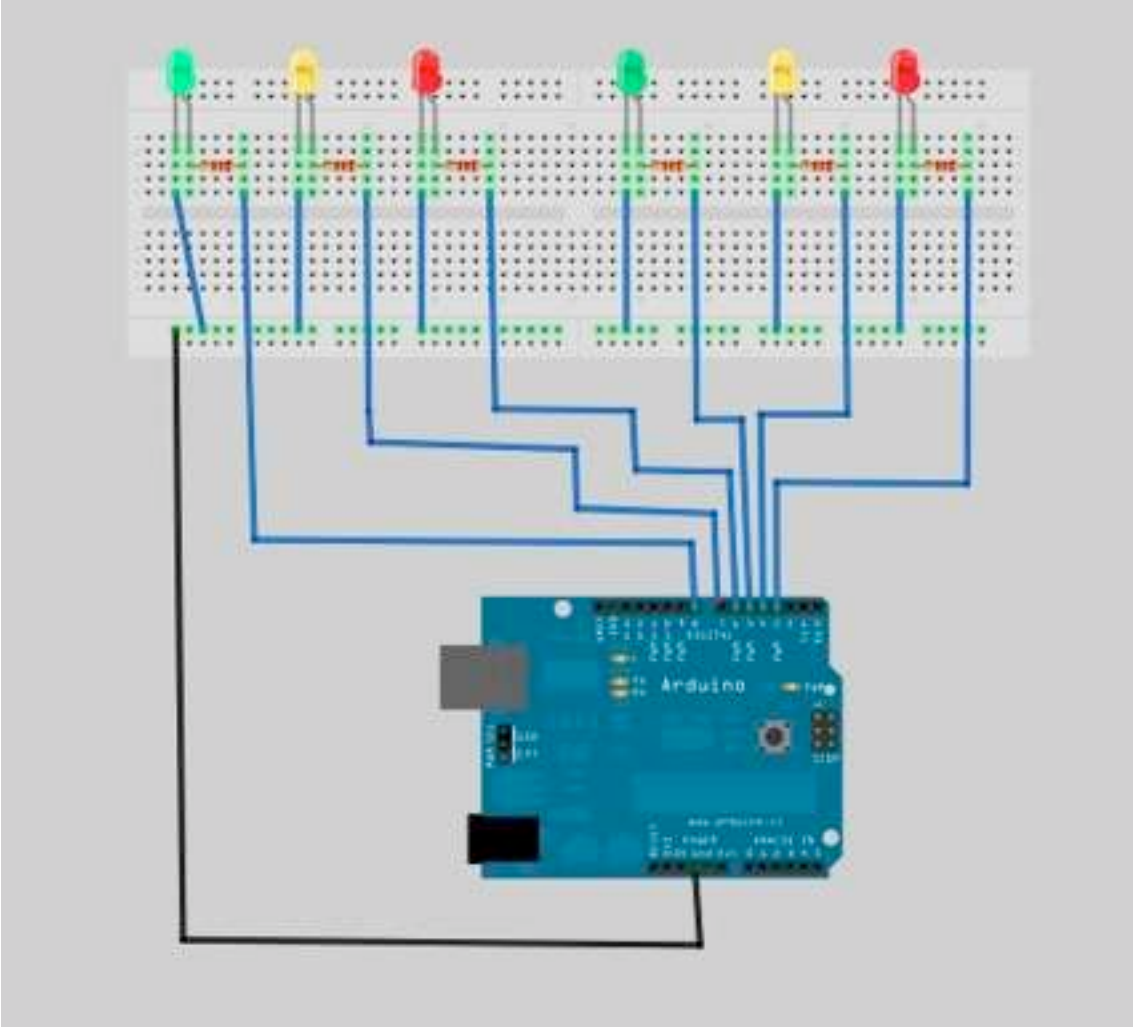
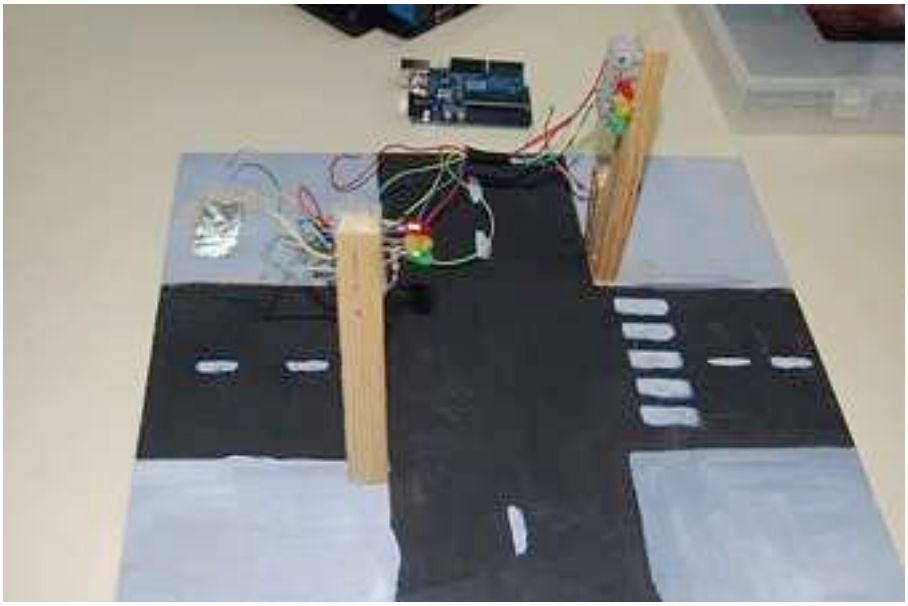


FOTO TRABAJO



EDUCACIÓN EN VALORES

Educación medioambiental

El control automático de un nudo semafórico, puede tener consecuencias interesantes desde el punto de vista ambiental. En este sentido pueden aprovecharse mecanismos como los propuestos en el proyecto de esta unidad para no malgastar energía.

COMPETENCIAS QUE SE TRABAJAN

Autonomía e iniciativa personal

Muchos alumnos se enfrentan a una tarea nueva: utilizar una controladora y programarla para controlar las acciones que lleva a cabo un circuito eléctrico. Los diferentes procedimientos propuestos a lo largo de la unidad pretenden que el alumno aborde estas nuevas tareas sin miedo a equivocarse (siempre, lógicamente, con el apoyo del profesor).

Competencia social y ciudadana

El trabajo en grupo es esencial en la sociedad moderna, sobre todo a la hora de diseñar y montar nuevos proyectos, muchos de ellos relacionados con las tareas que aparecen en esta unidad. Con el trabajo en equipo se fomenta el compromiso por realizar una tarea (no puedo fallar a mis colegas) o el respeto hacia las opiniones y gustos de los otros.

Además, dado que siempre habrá alumnos más aventajados, este trabajo en equipo debe tener también una función de apoyo hacia aquellos alumnos que presentan más dificultades a la hora de llevar a cabo las tareas propuestas.

Tratamiento de la información y competencia digital

Los alumnos constatarán la importancia de la programación en el control automático. Verán que con no demasiado esfuerzo y pocos medios es posible controlar de manera automática el encendido y apagado de diversos sistemas electrónicos.

CRITERIOS DE EVALUACIÓN

- Distinguir los principales elementos de entrada y salida de un sistema de control.
- Describir las características de una controladora, prestando especial atención a sus salidas y entradas, tanto analógicas como digitales.
- Utilizar la controladora para examinar el funcionamiento de un sistema a través del ordenador.
- Elaborar procedimientos sencillos de control en lenguaje para Arduino.
- Elaborar diagramas de flujo.
- Elaborar programas que controlen las entradas y salidas de una controladora.
- Manejar sencillos circuitos electrónicos a partir de un ordenador y una controladora.

Reloj digital sencillo.

Introducción

Hay sencillos aparatos que a veces te sorprenden su funcionamiento. Un reloj es como algo mágico, que parece dominar el tiempo. Sin duda, un invento imprescindible en al Historia de la Humanidad.

Este ejemplo de Arduino, fantástico como proyecto de fin de aprendizaje con la placa, es a la vez sencillo y efectivo. Sus resultados deben ser inmediatos y ofrecer una alta satisfacción al alumno.

Nota: este proyecto es simplemente el esbozo de una idea. No está testado ni montado físicamente en placa.

Competencias Básicas

1. Competencia matemática: medición del tiempo; reflexión sobre distintos ritmos y frecuencias.
2. Competencia lingüística: sintaxis de programación; expresión de ideas.
3. Competencia TIC: instalación de programas; procedimientos de programación.
4. Aprender a Aprender: búsqueda de referencias del lenguaje; imaginación para desarrollar circuitos.

Descripción del circuito electrónico

Como sabemos Arduino Uno tiene una capacidad de 14 entradas o salidas digitales, algunas de las cuales pueden ser moduladas analógicamente, más 6 entradas analógicas. Nuestros displays de 7 segmentos necesitan conectarse a 7 salidas digitales, numeradas de la “a” a la “g”, que activen cada led del display más un ánodo común que, conectado a través de una resistencia, lleve la intensidad a tierra.

Un reloj que muestre horas y minutos necesitaría cuatro displays de este tipo, con lo cual hablamos de 28 conexiones, el doble de las que puede ofrecer Arduino. Además, conectaríamos a través de una resistencia pequeña (unos 100Ω – 500Ω) cada ánodo a tierra.

Sin embargo si los ánodos no son conectados a tierra, sino a Vcc (tensión de alimentación), no circulará intensidad por el diodo y no se iluminarán. Este hecho puedo usarlo para reducir el conexionado de los displays a Arduino activando los cuatro displays en secuencia.

Efectivamente, conectando siete salidas digitales de Arduino a los displays, en paralelo, sólo se activará el display que en ese momento tenga el ánodo conectado a tierra. Así que, además de esas siete salidas digitales, necesito conectar cuatro salidas a cada ánodo de cada display y enviar en secuencia un cero lógico a la salida que quiero activar (podríamos decir que esas cuatro salidas son activas por lógica inversa). Dando un barrido del primer al cuarto dígito, del primer al cuarto display, si la velocidad de cambio es lo suficientemente lenta para activar los leds del display y lo suficientemente rápida para que el ojo humano no distinga el parpadeo podremos ofrecer los cuatros dígitos a la vez.

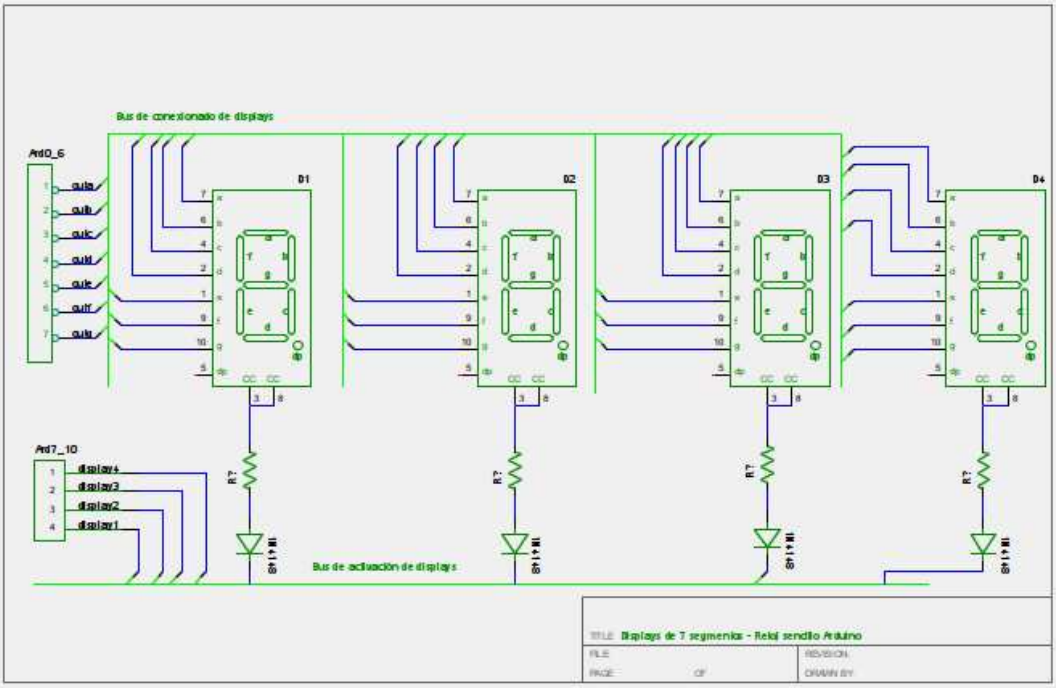
Y habremos usado, en total, 11 salidas digitales. Con dos pines más, esta vez configurados como entradas digitales, y conectados a pulsadores, podremos además modificar los valores ofrecidos por los dígitos de la hora y de los minutos. También puedo contemplar la posibilidad de usar dos potenciómetros conectados a entradas analógicas para el ajuste de la hora en el reloj.

Por último, y como medida de protección, es conveniente acompañar la resistencia del ánodo con un diodo tipo 1N4148 de baja tensión de activación. Esto evitaría posibles corrientes inversas del ánodo (cuando se conecta a 5V) con los cátodos que puedan estar a cero. Teóricamente no deberían

activarse los tramos del display por esta circunstancia, ya que son leds y por tanto diodos, pero no está de más asegurar que esta posibilidad no ocurra.

Montaje

Esquema del montaje de los displays para arduino es el siguiente:



Programación

Una vez realizado el montaje tenemos que realizar el programa que cuente el tiempo, mande las señales correctas a los displays y realice la secuencia de activación. Un ejemplo de programa, que aún tiene que ser comprobado, sería el siguiente (con comentarios en el mismo):

```
// siete salidas digitales a cada tramo del display

int outa = 0;
int outb = 1;
int outc = 2;
int outd = 3;
int oute = 4;
int outf = 5;
int outg = 6;

// a los anodos de los displays. Actuan por logica inversa
int anodo[4]={7,8,9,10};
```

```
 // entradas para modificar la hora y minuto
 int inhoraria = 11;
 int inminutero = 12;

 // cuenta el tiempo
 long tiempo = 0; //maximo 86400000. Cuenta en milisegundos el tiempo transcurrido en un dia
 int hora = 0; // de 0 a 23.
 int minuto = 0; // de 0 a 60
 int horaria = 0; // variable para detectar si cambia la hora
 int minutero = 0; // variable para detectar si cambia el minutero

 // variables de los displays
 int displays[4]; // variable para cada display
 int cualactivo = 0 ; //variable para guardar cual se activa

 /* ================= */
 /* Funciones y subrutinas   */
 /* ================= */

 void mandasenal(int numdisplay, int valor) {
   // recibe el numero de display y el valor que tiene que entregar
   // a) activa cada tramo del display segun valor
   switch (valor) {
     case 1:
       digitalWrite(outa,LOW);
       digitalWrite(outb,HIGH);
       digitalWrite(outc,HIGH);
       digitalWrite(outd,LOW);
       digitalWrite(oute,LOW);
       digitalWrite(outf,LOW);
       digitalWrite(outg,LOW);
     break;
     case 2:
       digitalWrite(outa,HIGH);
       digitalWrite(outb,HIGH);
       digitalWrite(outc,LOW);
       digitalWrite(outd,HIGH);
       digitalWrite(oute,HIGH);
       digitalWrite(outf,LOW);
       digitalWrite(outg,HIGH);
     break;
     case 3:
       digitalWrite(outa,HIGH);
       digitalWrite(outb,HIGH);
       digitalWrite(outc,HIGH);
       digitalWrite(outd,HIGH);
       digitalWrite(oute,LOW);
       digitalWrite(outf,LOW);
       digitalWrite(outg,HIGH);
     break;
```

```
case 4:
  digitalWrite(outa,LOW);
  digitalWrite(outb,HIGH);
  digitalWrite(outc,HIGH);
  digitalWrite(outd,LOW);
  digitalWrite(oute,LOW);
  digitalWrite(outf,HIGH);
  digitalWrite(outg,HIGH);
break;
case 5:
  digitalWrite(outa,HIGH);
  digitalWrite(outb,LOW);
  digitalWrite(outc,HIGH);
  digitalWrite(outd,HIGH);
  digitalWrite(oute,LOW);
  digitalWrite(outf,HIGH);
  digitalWrite(outg,HIGH);
break;
case 6:
  digitalWrite(outa,LOW);
  digitalWrite(outb,LOW);
  digitalWrite(outc,HIGH);
  digitalWrite(outd,HIGH);
  digitalWrite(oute,HIGH);
  digitalWrite(outf,HIGH);
  digitalWrite(outg,HIGH);
break;
case 7:
  digitalWrite(outa,HIGH);
  digitalWrite(outb,HIGH);
  digitalWrite(outc,HIGH);
  digitalWrite(outd,LOW);
  digitalWrite(oute,LOW);
  digitalWrite(outf,LOW);
  digitalWrite(outg,LOW);
break;
case 8:
  digitalWrite(outa,HIGH);
  digitalWrite(outb,HIGH);
  digitalWrite(outc,HIGH);
  digitalWrite(outd,HIGH);
  digitalWrite(oute,HIGH);
  digitalWrite(outf,HIGH);
  digitalWrite(outg,HIGH);
break;
case 9:
  digitalWrite(outa,HIGH);
  digitalWrite(outb,HIGH);
  digitalWrite(outc,HIGH);
  digitalWrite(outd,HIGH);
  digitalWrite(oute,LOW);
  digitalWrite(outf,LOW);
```

```
      digitalWrite(outg,HIGH);
    break;
  }
  // b) activo un solo display cada vez
  for (int i=0;i<=3;i++) { // recorrido de 0 hasta 3
    if (i==numdisplay) {
       digitalWrite(anodo[i],LOW); // activo por logica inversa. En el pin almacenado por el array 'anodo' posicion 0 es donde se pone el cero
    } else {
      digitalWrite(anodo[i],HIGH); // desactivo por logica inversa
    } // fin del if
  } // fin del for
}

/* ============================================== */
/* Bucles principales del programa: loop y setup */
/* ============================================== */

void setup () { // inicializo
  pinMode(outa, OUTPUT);
  pinMode(outb, OUTPUT);
  pinMode(outc, OUTPUT);
  pinMode(outd, OUTPUT);
  pinMode(oute, OUTPUT);
  pinMode(outf, OUTPUT);
  pinMode(outg, OUTPUT);
  pinMode(anodo[0], OUTPUT);
  pinMode(anodo[1], OUTPUT);
  pinMode(anodo[2], OUTPUT);
  pinMode(anodo[3], OUTPUT);
  pinMode(inhoraria, INPUT);
  pinMode(inminutero, INPUT);
}

void loop () { // bucle

 // 1) comprueba horaria
  horaria = digitalRead(inhoraria); //Lee la entrada digital
  if (horaria==true) { //compara
      delay(50); // tiempo de espera 50 milisegundos de pulsacion de boton. El boton debe mantenerse pulsado 50 milisegundos
      // hacerlo de esta manera evita que una pulsacion fisica sea reconocida como muchas pulsaciones
    // ajustar el retraso por cada boton. Se ha puesto un valor de 50 pero podria tener que ser mas
    if (horaria==digitalRead(inhoraria)) {
    tiempo = tiempo + 3600000; // suma  3600 segundos, o sea, una hora
      if (tiempo>=86400000) {tiempo=tiempo-86400000;} // si me paso del tiempo de un dia, resta 86400 seg
    }
  }
```

```
   // 2) comprueba minutero
   minutero = digitalRead(inminutero); //Lee la entrada digital
   if (minutero==true) { //compara
       delay(50); // tiempo de espera 50 milisegundos de pulsacion de boton. El boton debe mantenerse pulsado 50 milisegundos
     if (minutero==digitalRead(inminutero)) {
     tiempo = tiempo + 60000; // suma  60 segundos, o sea, un minuto
       if (tiempo>=86400000) {tiempo=tiempo-86400000;} // si me paso del tiempo de un dia, resta 86400 seg
     }
   }

   // 3) hora y minuto. Tiempo en los displays
  hora = tiempo / 3600000; // indica la hora
  minuto = (tiempo % 3600000) / 60000; // el resto del de la hora, son los segundos que se pasa de esta. Dividido entre 60 nos da los minutos
   // de izquierda a derecha, el primer display corresponde a las decenas de las horas, el segundo a las unidades de las horas,
  // el tercero a las decenas del minutero y el cuarto a las unidades del minutero
  displays[0]= hora/10;
  displays[1]= hora % 10; // resto de la division entre 10
  displays[2]= minuto / 10:
  displays[3]= minuto % 10: // resto de la division entre 10

  // 4) activa uno de los displays
  cualactivo=tiempo % 100; // el resto de la division entre 100 es un numero de 0 a 99. Cada 100 milisegundos se repite...
  // ...la secuencia de 0 a 99 y empieza de nuevo.
   cualactivo = cualactivo / 25; // en esos 100 milisegundos, la variable 'cualactivo' vale 0 los primeros 25 milisegundos, 1, 2 y 3 los
   // ultimos 25 milisegundos... Asi activare cada display en secuencia 25 milisegundos cada uno.
  mandasenal(cualactivo,displays[cualactivo]); // activa los pines del display elegido

  // comando de espera
  delay(1); // espera cada vez un milisegundo.

  // el bucle finaliza mandando un milisegundo mas al contador
  tiempo = (tiempo+1)*(tiempo<86400000) //si es mayor o igual que 86400000 se pone a cero.
 }
 // Fin del programa
```

Comentarios

El programa principal básicamente comprueba si se están pulsando los pulsadores de las horarias y el minutero. Si así es, suma una hora o un minuto al tiempo, restándose 86400000 milisegundos si me paso de esa cantidad. Esto ajustaría la hora del reloj.

Posteriormente calcula los valores que hay que poner en cada display, calculando previamente la variable hora y la variable minuto según los milisegundos recorridos.

Calcula después, cada milisegundo, a que display corresponde encenderse y pasa a la subrutina

“**mandasenal”** el display que toca encenderse y con qué valor. Cada display se enciende a intervalos de 25 milisegundos. Una de las cosas que habría que comprobar es si este régimen de cambio de display es suficiente para engañar al ojo humano o se necesita un régimen más rápido o más lento, para lo cual habría que modificar los valores en la programación.

Por último espera un milisegundo y añade 1 al contador general del tiempo, reinicializándolo a cero si se pasa de 8640000000 milisegundos.

La subrutina “**mandasenal”** se encarga de activar los tramos del display según el valor (código siete-segmentos) y elegir qué display es el que se activa mandando un “cero” al mismo.

Unidad Didáctica para el curso "La microcontroladora Arduino en secundaria"

Juan Antonio Villalpando Nuño. Octubre 2011.

Nombre de la Unidad Didáctica: **Juego de luces con diodos LED.**

Objetivo del proyecto:

Se trata de establecer 4 juego de luces con varios pulsadores.

Se parte de 5 pulsadores, con 4 de ellos podemos establecer distintos juegos de luces en 5 diodos.
Cada secuencia comenzará en el momento en que se accione un pulsador , y continuará después de dejar de pulsarlo.
El quinto pulsador se utilizará para parar cualquier secuencia de luces que esté en funcionamiento.

Las entradas del circuito estarán formadas por 5 pulsadores llevados a masa mediante sendas resistencias de 10K.
Las salidas constarán de 5 diodos LED en configuración de cátodo común con una resistencia limitadora de 1K.

Según se accione un pulsador del 1 al 4, los diodos LED se encenderán y apagarán según una secuencia establecida.
Un quinto pulsador actuará como señal de parada de cualquier secuencia.

Se utilizará un solo bucle para realizar todo el proceso, ya que la rapidez del Arduino es suficiente para obtener una respuesta operativa.

El profesor sugerirá la secuencia que se deberá obtener según cada botón pulsado.

Objetivo didáctico:

Es un proyecto que se puede incorporar a las primeras fases de aprendizaje de Arduino donde el alumnado tiene que realizar un cableado sencillo pero abundante, por lo cual desarrollará destrezas en el conexionado.
La profusión de pines de conexiones puede provocar un mal funcionamiento del circuito debido a errores de conexión o malos contactos. Por lo cual el alumnado debe repasar y corregir la circuitería en caso de fallos.

En cuanto al programa, también presenta las mismas características que el conexionado, es sencillo de deducir pero su escritura se puede prestar a errores de sintaxis debido a la cantidad de líneas utilizadas.

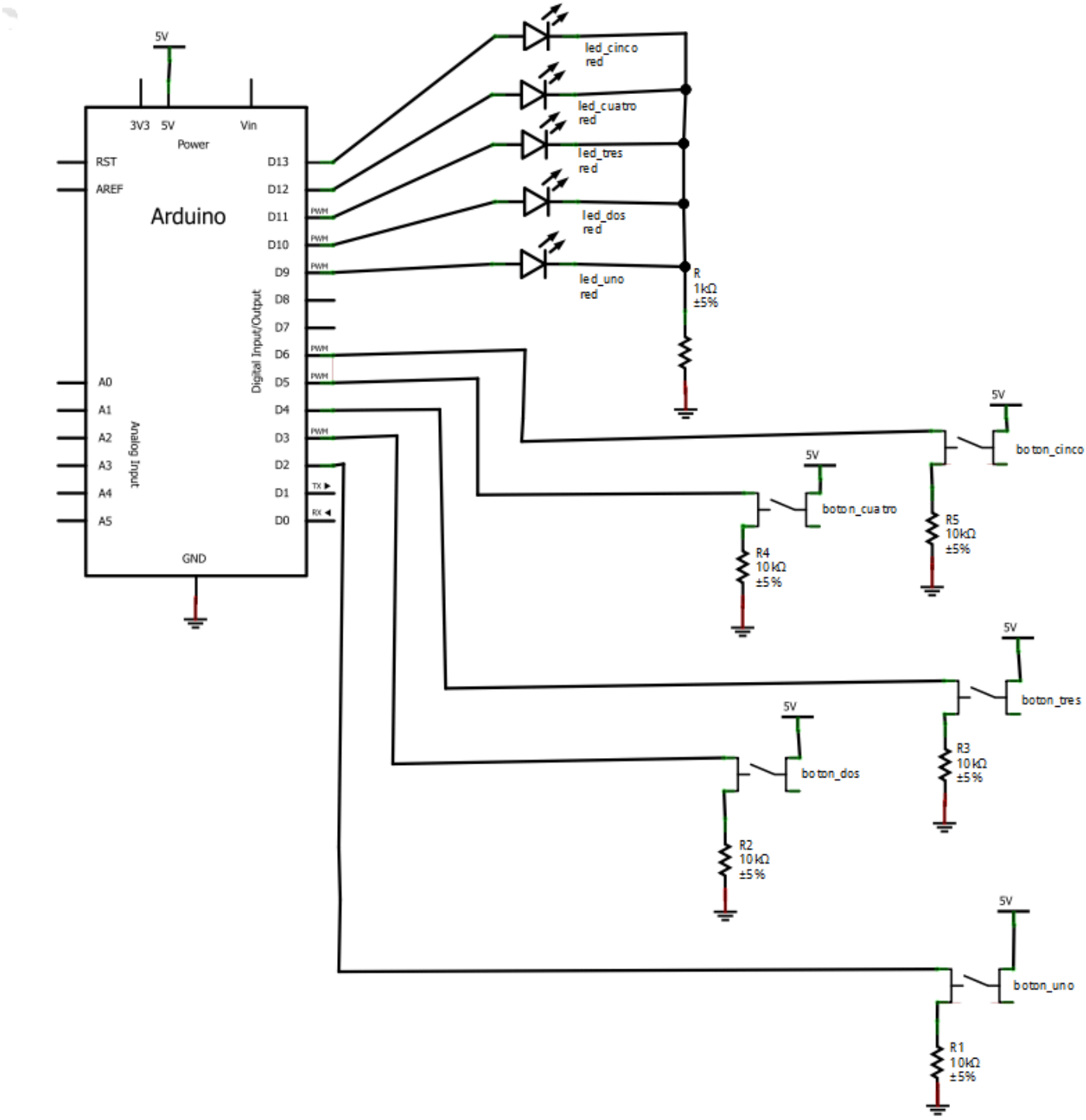
Ampliación de contenido:

En el circuito:
Utilizar una resistencia para cada LED en vez de una común.

En el programa:
Al alumnado que realice con éxito la propuesta de programa anterior, se le propondrá que modifique el código con el uso intensivo de void, de tal manera que deberá declarar tanto las secuencias de luces como la respuesta del botón de parada en funciones void (funciones sin respuestas).

Creará las funciones void uno(), void dos(), void tres(), void cuatro() y void cinco(), que serán llamadas desde dentro del bucle (loop) del programa.
(Ver "Programa-2" al final del trabajo)

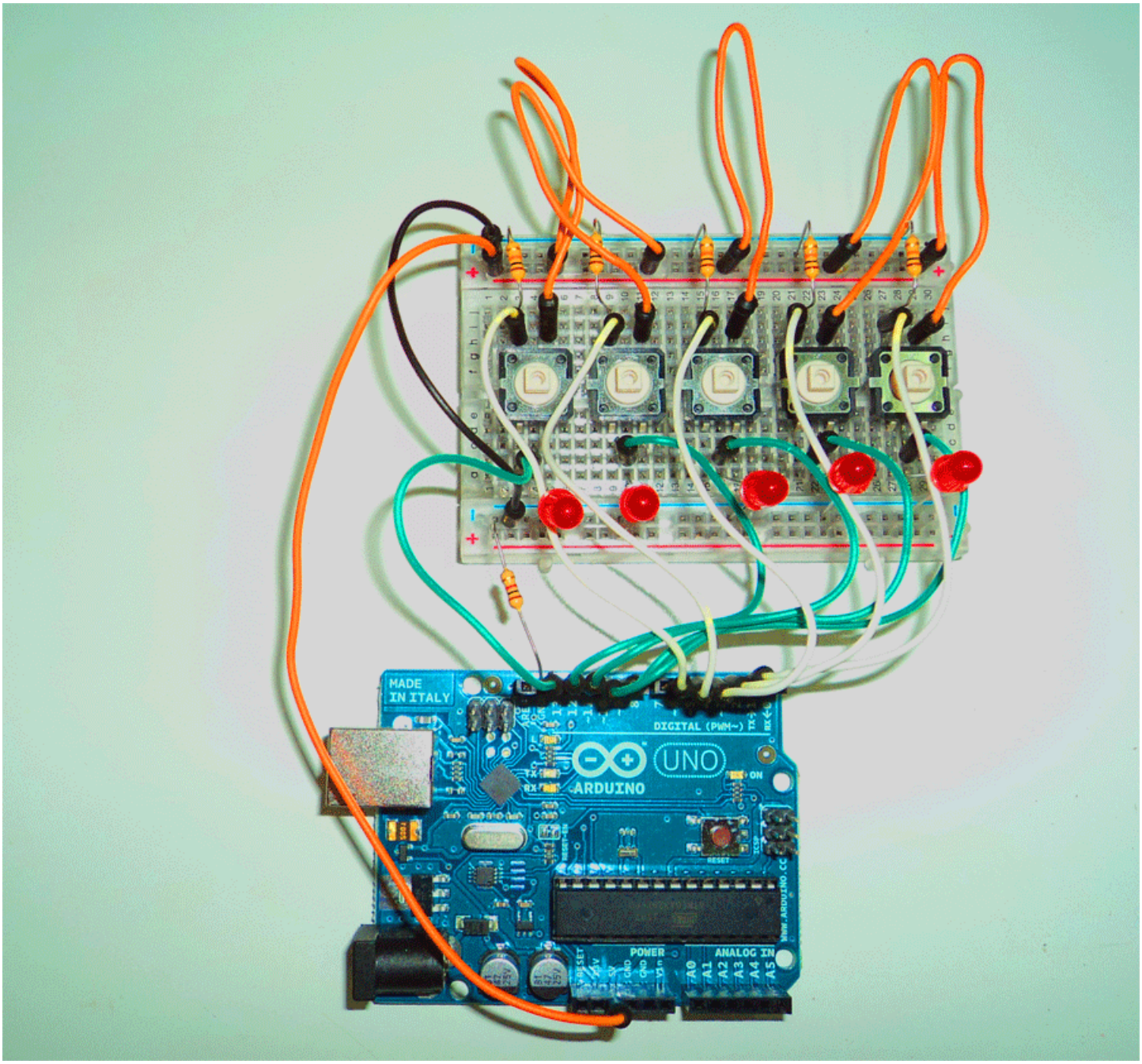
Esquema de conexiones.



Componentes de circuitería necesarios:

- Placa Arduino.
- Protoboard.
- Cinco diodos LED.
- Cinco pulsadores.
- Una resistencia de 1K.
- Cinco resistencias de 10K.
- Cables de conexión.

Cableado.



- PROGRAMA (I)

```
// Declaración de constantes
const int boton_uno = 2;
const int boton_dos = 3;
const int boton_tres = 4;
const int boton_cuatro = 5;
const int boton_cinco = 6;
const int led_uno = 9;
const int led_dos = 10;
const int led_tres = 11;
const int led_cuatro = 12;
const int led_cinco = 13;


// Declaración de variables
int b_uno = 0;
int b_dos = 0;
int b_tres = 0;
int b_cuatro = 0;
int b_cinco = 0;
int marca = 0;
int tiempo = 100;

// Iniciación
void setup() {
  pinMode(boton_uno, INPUT);
  pinMode(boton_dos, INPUT);
  pinMode(boton_tres, INPUT);
  pinMode(boton_cuatro, INPUT);
  pinMode(boton_cinco, INPUT);
  pinMode(led_uno, OUTPUT);
  pinMode(led_dos, OUTPUT);
  pinMode(led_tres, OUTPUT);
  pinMode(led_cuatro, OUTPUT);
  pinMode(led_cinco, OUTPUT);

}

void loop(){

  b_uno = digitalRead(boton_uno);
  b_dos = digitalRead(boton_dos);
  b_tres = digitalRead(boton_tres);
  b_cuatro = digitalRead(boton_cuatro);
  b_cinco = digitalRead(boton_cinco);

  if (b_uno == HIGH) { marca = 1; }
  if (b_dos == HIGH) { marca = 2; }
  if (b_tres == HIGH) { marca = 3; }
  if (b_cuatro == HIGH) { marca = 4; }
  if (b_cinco == HIGH) { marca = 5; }

  if (marca == 1)
  {
 digitalWrite(led_uno, HIGH);
  delay(tiempo);;
 digitalWrite(led_uno, LOW);
 digitalWrite(led_dos, HIGH);
 delay(tiempo);
 digitalWrite(led_dos, LOW);
 digitalWrite(led_tres, HIGH);
 delay(tiempo);
 digitalWrite(led_tres, LOW);
 digitalWrite(led_cuatro, HIGH);
 delay(tiempo);
 digitalWrite(led_cuatro, LOW);
 digitalWrite(led_cinco, HIGH);
```

```
delay(tiempo);
digitalWrite(led_cinco, LOW);
delay(tiempo);
}
 if (marca == 2)
 {
digitalWrite(led_uno, HIGH);
digitalWrite(led_cinco, HIGH);
delay(tiempo);
digitalWrite(led_uno, LOW);
digitalWrite(led_cinco, LOW);
digitalWrite(led_dos, HIGH);
digitalWrite(led_cuatro, HIGH);
delay(tiempo);
digitalWrite(led_dos, LOW);
digitalWrite(led_cuatro, LOW);
digitalWrite(led_tres, HIGH);
delay(tiempo);
digitalWrite(led_tres, LOW);
digitalWrite(led_dos, HIGH);
digitalWrite(led_cuatro, HIGH);
delay(tiempo);
digitalWrite(led_dos, LOW);
digitalWrite(led_cuatro, LOW);
delay(tiempo);
}
if (marca == 3)
 {
digitalWrite(led_uno, HIGH);
digitalWrite(led_dos, HIGH);
digitalWrite(led_cuatro, LOW);
digitalWrite(led_cinco, LOW);
delay(tiempo);
digitalWrite(led_uno, LOW);
digitalWrite(led_dos, LOW);
digitalWrite(led_cuatro, HIGH);
digitalWrite(led_cinco, HIGH);
delay(tiempo);
digitalWrite(led_uno, LOW);
digitalWrite(led_dos, LOW);
digitalWrite(led_cuatro, LOW);
digitalWrite(led_cinco, LOW);
delay(tiempo);
digitalWrite(led_uno, HIGH);
digitalWrite(led_dos, HIGH);
digitalWrite(led_tres, HIGH);
digitalWrite(led_cuatro, HIGH);
digitalWrite(led_cinco, HIGH);
delay(tiempo);
digitalWrite(led_uno, LOW);
digitalWrite(led_dos, LOW);
digitalWrite(led_tres, LOW);
digitalWrite(led_cuatro, LOW);
digitalWrite(led_cinco, LOW);
delay(tiempo);
}
if (marca == 4)
 {
digitalWrite(led_uno, HIGH);
digitalWrite(led_dos, HIGH);
digitalWrite(led_tres, HIGH);
digitalWrite(led_cuatro, HIGH);
digitalWrite(led_cinco, HIGH);
delay(tiempo);
digitalWrite(led_uno, LOW);
digitalWrite(led_dos, LOW);
digitalWrite(led_tres, LOW);
digitalWrite(led_cuatro, LOW);
digitalWrite(led_cinco, LOW);
delay(tiempo);
```

```
 }
  if (marca == 5)
  {
 digitalWrite(led_uno, LOW);
 digitalWrite(led_dos, LOW);
 digitalWrite(led_tres, LOW);
 digitalWrite(led_cuatro, LOW);
 digitalWrite(led_cinco, LOW);
 }

}
```

****
****
****
****
****

- PROGRAMA (II). Mejora del programa anterior. Uso de void.

```
const int boton_uno = 2;
const int boton_dos = 3;
const int boton_tres = 4;
const int boton_cuatro = 5;
const int boton_cinco = 6;
const int led_uno = 9;
const int led_dos = 10;
const int led_tres = 11;
const int led_cuatro = 12;
const int led_cinco = 13;


// variables will change:
int b_uno = 0;
int b_dos = 0;
int b_tres = 0;
int b_cuatro = 0;
int b_cinco = 0;
int marca = 0;
int tiempo = 100;

void setup() {
  pinMode(boton_uno, INPUT);
  pinMode(boton_dos, INPUT);
  pinMode(boton_tres, INPUT);
  pinMode(boton_cuatro, INPUT);
  pinMode(boton_cinco, INPUT);
  pinMode(led_uno, OUTPUT);
  pinMode(led_dos, OUTPUT);
  pinMode(led_tres, OUTPUT);
  pinMode(led_cuatro, OUTPUT);
  pinMode(led_cinco, OUTPUT);

}
void uno()
```

```
{
 digitalWrite(led_uno, HIGH);
 delay(tiempo);;
 digitalWrite(led_uno, LOW);
 digitalWrite(led_dos, HIGH);
 delay(tiempo);
 digitalWrite(led_dos, LOW);
 digitalWrite(led_tres, HIGH);
 delay(tiempo);
 digitalWrite(led_tres, LOW);
 digitalWrite(led_cuatro, HIGH);
 delay(tiempo);
 digitalWrite(led_cuatro, LOW);
 digitalWrite(led_cinco, HIGH);
 delay(tiempo);
 digitalWrite(led_cinco, LOW);
 delay(tiempo);
}
void dos()
{
 digitalWrite(led_uno, HIGH);
 digitalWrite(led_cinco, HIGH);
 delay(tiempo);
 digitalWrite(led_uno, LOW);
 digitalWrite(led_cinco, LOW);
 digitalWrite(led_dos, HIGH);
 digitalWrite(led_cuatro, HIGH);
 delay(tiempo);
 digitalWrite(led_dos, LOW);
 digitalWrite(led_cuatro, LOW);
 digitalWrite(led_tres, HIGH);
 delay(tiempo);
 digitalWrite(led_tres, LOW);
 digitalWrite(led_dos, HIGH);
 digitalWrite(led_cuatro, HIGH);
 delay(tiempo);
 digitalWrite(led_dos, LOW);
 digitalWrite(led_cuatro, LOW);
 delay(tiempo);
}
void tres()
{
 digitalWrite(led_uno, HIGH);
 digitalWrite(led_dos, HIGH);
 digitalWrite(led_cuatro, LOW);
 digitalWrite(led_cinco, LOW);
 delay(tiempo);
 digitalWrite(led_uno, LOW);
 digitalWrite(led_dos, LOW);
 digitalWrite(led_cuatro, HIGH);
 digitalWrite(led_cinco, HIGH);
 delay(tiempo);
 digitalWrite(led_uno, LOW);
 digitalWrite(led_dos, LOW);
 digitalWrite(led_cuatro, LOW);
 digitalWrite(led_cinco, LOW);
 delay(tiempo);
 digitalWrite(led_uno, HIGH);
 digitalWrite(led_dos, HIGH);
 digitalWrite(led_tres, HIGH);
 digitalWrite(led_cuatro, HIGH);
 digitalWrite(led_cinco, HIGH);
 delay(tiempo);
 digitalWrite(led_uno, LOW);
 digitalWrite(led_dos, LOW);
 digitalWrite(led_tres, LOW);
 digitalWrite(led_cuatro, LOW);
 digitalWrite(led_cinco, LOW);
 delay(tiempo);
}
```

```
void cuatro()
{
 digitalWrite(led_uno, HIGH);
 digitalWrite(led_dos, HIGH);
 digitalWrite(led_tres, HIGH);
 digitalWrite(led_cuatro, HIGH);
 digitalWrite(led_cinco, HIGH);
 delay(tiempo);
 digitalWrite(led_uno, LOW);
 digitalWrite(led_dos, LOW);
 digitalWrite(led_tres, LOW);
 digitalWrite(led_cuatro, LOW);
 digitalWrite(led_cinco, LOW);
 delay(tiempo);
}
void parar()
{
 digitalWrite(led_uno, LOW);
 digitalWrite(led_dos, LOW);
 digitalWrite(led_tres, LOW);
 digitalWrite(led_cuatro, LOW);
 digitalWrite(led_cinco, LOW);
}

void loop(){

  b_uno = digitalRead(boton_uno);
  b_dos = digitalRead(boton_dos);
  b_tres = digitalRead(boton_tres);
  b_cuatro = digitalRead(boton_cuatro);
  b_cinco = digitalRead(boton_cinco);

  if (b_uno == HIGH) { marca = 1; }
  if (b_dos == HIGH) { marca = 2; }
  if (b_tres == HIGH) { marca = 3; }
  if (b_cuatro == HIGH) { marca = 4; }
  if (b_cinco == HIGH) { marca = 5; }

  if (marca == 1)
  {
  uno();
 }
  if (marca == 2)
  {
  dos();
 }
 if (marca == 3)
  {
  tres();
 }
 if (marca == 4)
  {
  cuatro();
 }
  if (marca == 5)
  {
  parar();
 }

}
```

LA MICROCONTROLADORA ARDUINO EN SECUNDARIA

Una propuesta de Daniel Gallardo García

PROPUESTA: TECNO-PORTAL DE BELÉN

Descripción de la propuesta didáctica:

Como la Navidad está a la vuelta de la esquina, nuestro Departamento de Tecnología ha decidido deleitar a toda la comunidad educativa del Centro Educativo con un Tecno-Portal de Belén. Vamos a justificar en qué nos hemos ido gastando el dinero del Departamento y alguna que otra partida extra durante estos tres (ya cuento este curso) últimos años, y vamos a realizar un portal de Belén empleando piezas y muñecos de Lego, y tendrá un simple dispositivo electrónico controlado por una controladora Arduino UNO.

El Tecno-portal de Belén deberá tener:

- Una fogata para dar calorcito al niño Jesús.
- Un río por el que corra agua. Como el consumo eléctrico de la bomba es elevado, éste solo se accionará (a través de un pulsador) por petición exclusiva de la persona que contemple el Belén.
- Una alarma sonora, que avise de manera gradual el peligro que acarrea para la integridad del Belén la proximidad de la mano y/o cara del espectador.
- Un lego-caganet.

Metodología:

Evidentemente, me saltaré la descripción de las sesiones previas con los alumnos para la explicación del funcionamiento de la controladora Arduino, así como de la parte de la programación didáctica de la unidad (objetivos, competencias, contenidos, temas transversales, cultura andaluza, actividades complementarias, criterios de calificación y de evaluación, etc.).
Un buen ejemplo de cómo abordar las sesiones necesarias es justamente tal y como las hemos desarrollado en este magnífico curso (tengo todo bien apuntado).

¿Cómo solucionaremos todas estas necesidades?:

Lógicamente, emplearemos soluciones sencillas y eficaces, empleando casi la totalidad de lo aprendido en el curso.

Fogata:

Lo resolveremos con un led rojo. Dicho led deberá iluminar de tal forma que imite el cimbreado de una candela. Para ello utilizaremos números aleatorios que determinen la potencia de brillo y también la duración de dicho brillo.

Río:

Lo resolveremos con un motor que actuará de bomba hidráulica, impulsando hacia arriba un circuito cerrado de agua. Dicho motor sólo se accionará cuando se haga presión sobre el pulsador, por lo cual necesitaremos una entrada digital que determine el estado de dicho pulsador. Para alimentar correctamente al motor deberemos utilizar un montaje con un transitor npn BD135 y una resistencia de base.

Alarma:

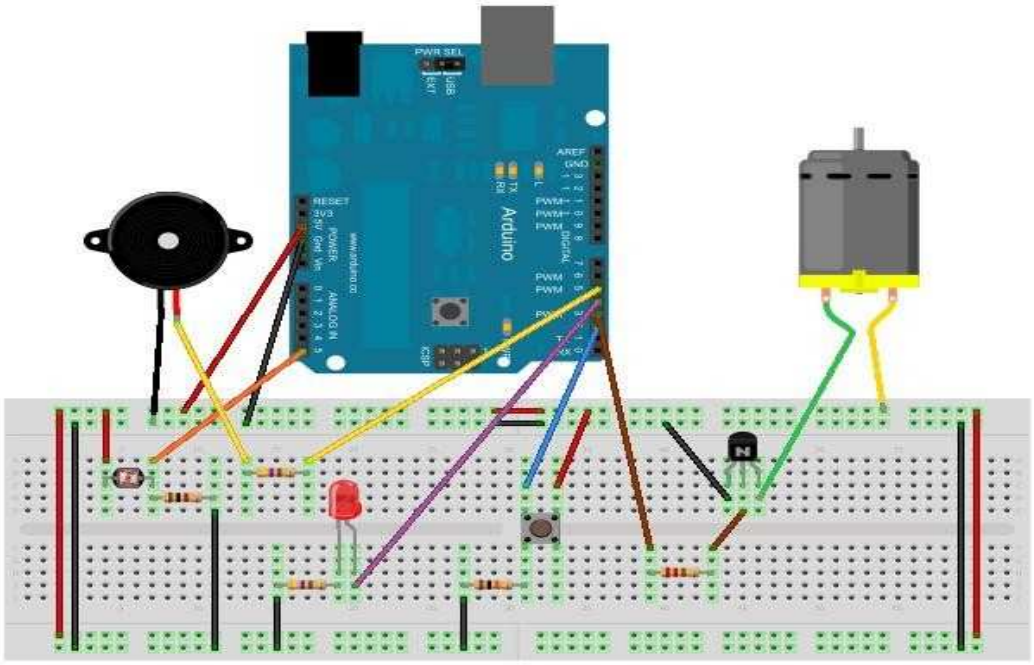
Como sensor de proximidad de la mano o cara emplearemos una resistencia LDR, que la asociaremos a una entrada analógica, y nos ayudará a determinar el grado de proximidad del cuerpo. Como queremos que la señal sonora sea gradual en función con la cercanía de la mano, necesitaremos una salida analógica, que será el resultado de mapear la señal de entrada de la LDR a un rango de frecuencias adecuado.

Lego-caganet:

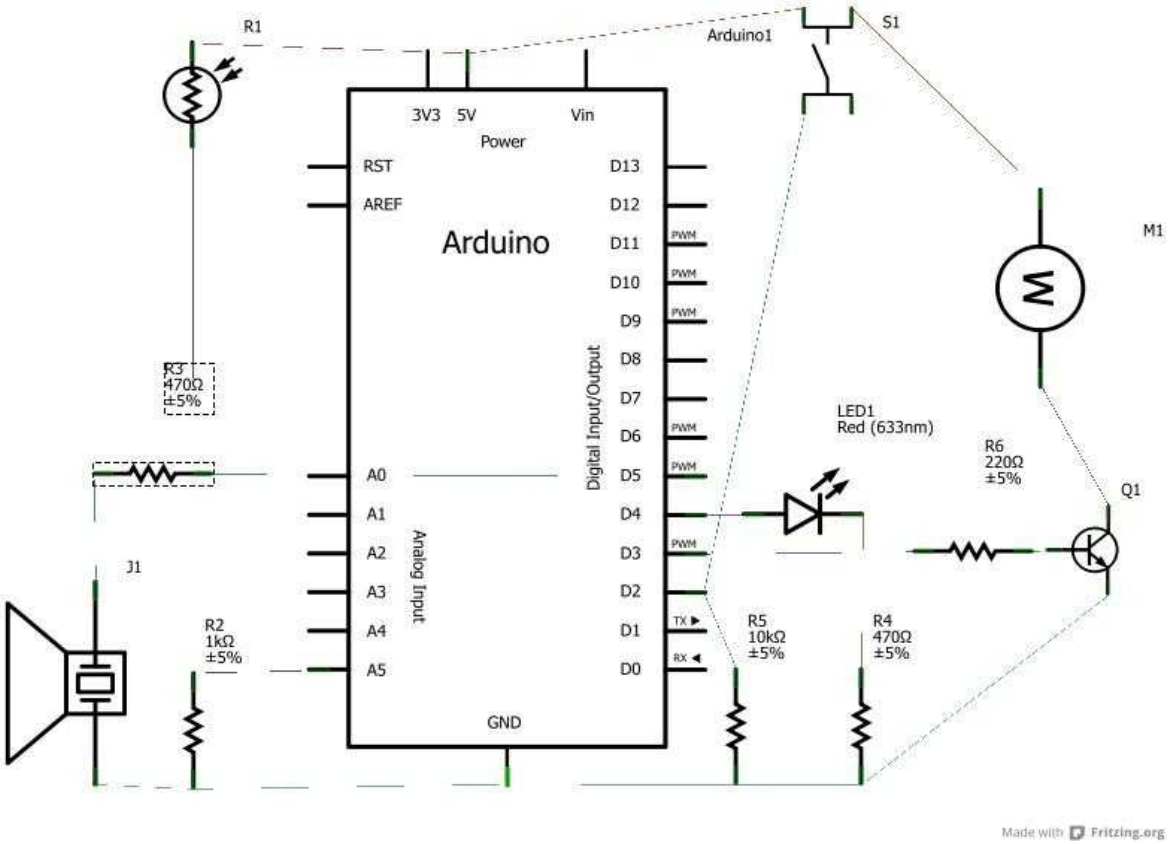
Por supuesto, este es el aspecto más difícil de ejecutar. Para que un muñeco de Lego tenga pose de estar cagando, deberemos cortar a inglete las piernas a la altura de la rodilla, y posteriormente realizar la unión con pegamento de contacto. Por último, se le dibujará cara de estar apretando con un rotulador indeleble.

Montaje sobre la placa protoboard:

A través de la herramienta Firtzing podemos ver de manera clara y limpia el conexionado de los distintos elementos que componen el circuito:



El esquema quedaría así (no he podido evitar cruzar líneas para hacer el circuito impreso):



Programación con Arduino 0022:

El programa que hará funcionar correctamente este circuito electrónico es el siguiente:

```
int luz;                // será el valor de entrada de la LDR
int boton;              // será el estado del pulsador
int sonido;             // será el valor de salida para el altavoz
int candela;            // será el valor del brillo para el LED
int tiempo;             // será el valor del tiempo de iluminación para el LED
int pulsador=2;         // conectaré la entrada para la LDR en el pin 2
int led=3;              // conectaré la salida para el LED en el pin 3
int altavoz=5;          // conectaré el altavoz en pin 5
int motor=6;            // conectaré el motor en el pin 6

void setup(){
  Serial.begin(9600);         // me interesará ver en pantalla algunos valores de entrada y
                              // salida para optimizar la función map
  randomSeed(0);              // necesitaré un ramdom para crear mi fogata
  pinMode(pulsador,INPUT);
  pinMode(led,OUTPUT);
  pinMode(altavoz,OUTPUT);
  pinMode(motor,OUTPUT);      // también utilizaremos la entrada analógica A5 para la LDR
}

void loop(){
  luz=analogRead(5);
  boton=digitalRead(pulsador);
  candela=random(50,255);
  tiempo=random(20,100);
  Serial.print("Luz recibida: "); Serial.print(luz); Serial.print('\t');
  sonido=map(luz,20,500,3000,20);
  Serial.print("frecuencia sonido: "); Serial.println(sonido);
  if (sonido>350){
    tone(altavoz,sonido,100);         // el altavoz se utilizará como alarma en caso de que
                                      // alguien acerque la mano sobre el montaje sonará tanto
                                      //más agudo cuanto más cerca se ponga la mano
  } else {analogWrite(altavoz,0);}  // elimino el ruido de fondo para valores bajos de sonido
  analogWrite(led,candela);
  delay(tiempo);                      //el LED se utilizará como fogata en mi portal de Belén
  if(boton==HIGH){
    analogWrite(motor,150);
  } else {analogWrite(motor,0);}    //el motor de la bomba de agua solamente se encenderá
                                    //cuando accionemos el pulsador

}
```

Problemas detectados:

Entiendo que esta propuesta didáctica no sea muy creativa, es decir: la programación no es para nada difícil. No obstante, para nuestros alumnos tampoco es nada fácil. Incluso para mí: y es que me ha surgido una duda importante:

Al igual que hacíamos con los Mindstorms programas paralelos a través de las bifurcaciones, ¿cómo se puede hacer esto con Arduino? Por ejemplo: en mi circuito, cuando acerco la mano a la LDR y suena el altavoz, mi led-fogata deja de brillar (porque el programa queda “atascado” haciendo sonar el altavoz, y hasta que no quito la mano y subo el umbral de iluminación sobre la LDR, no vuelve a encender el led.

Otro ejemplo: ¿qué ocurriría si durante la ejecución de este programa yo quiero además que otro led esté parpadeando cada dos segundo? Si utilizo un delay, ¡¡me paralizará todo el programa!!.

Conclusión:

Hace falta hacer el curso *Arduino Avanzado*.

PROPUESTA DE EJERCICIO

Se desea automatizar un repartidor de bolas, tal como aparece en la figura (ver esquema de situación), de tal forma que el repartidor será un cilindro neumático de doble efecto, controlado por una electroválvula 5/2, que funciona con una alimentación de 24V.

Para provocar el cambio de sentido del cilindro neumático, utilizaremos unos detectores finales de carrera, que fabricaremos mediante dos células LDR. Usaremos Arduino para provocar que esta secuencia sea ininterrumpida conforme el cilindro esté próximo a las células LDR, que detectarán la posición y actuarán como finales de carrera. Una vez detectada la posición del vástago del cilindro, el final de carrera actuará sobre la electroválvula, cambiando el sentido de avance ó retroceso del mismo, lo cual conseguiremos mediante el uso de relés.

Es importante en nuestro programa establecer las indicaciones oportunas, que eviten que una vez detectado la posición del vástago y provocado el cambio en la electroválvula, éste permanezca en tanto no se detecte el final de carrera siguiente, que volverá a invertir el ciclo.

Igualmente, y puesto que utilizaremos la microcontroladora Arduino, podemos incluir otros elementos tales como señales luminosas (diodos led), acústicas ó cuantas variantes sean necesarias.

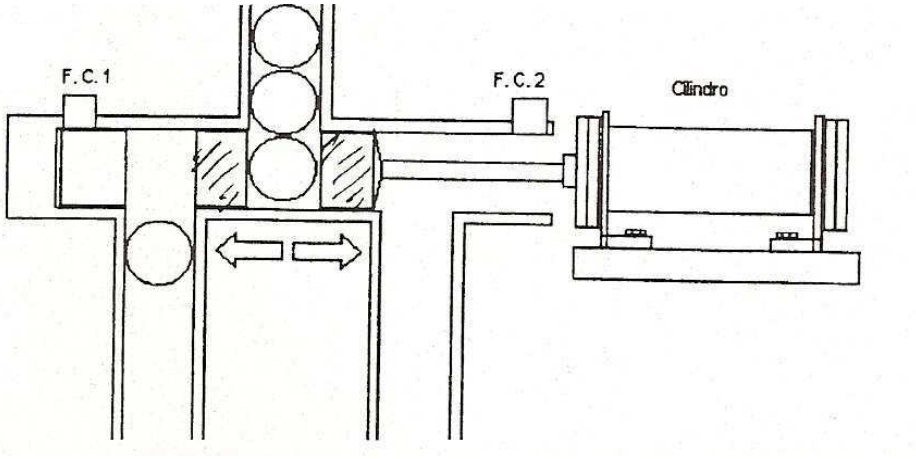
JUSTIFICACIÓN

Con este ejercicio, se pretende mostrar las diferentes posibilidades de arduino, implementado su uso con el de otras tecnologías (como puede ser la Neumática en este caso), de posible aplicación en el ámbito Industrial; con lo que conseguimos conectar al alumno con el mundo laboral, a través de nuestra asignatura, a la vez que relaccionamos el uso de las tecnologías principales de las que actualmente se hace uso no sólo en la Industria, sino en gran parte del ámbito laboral.

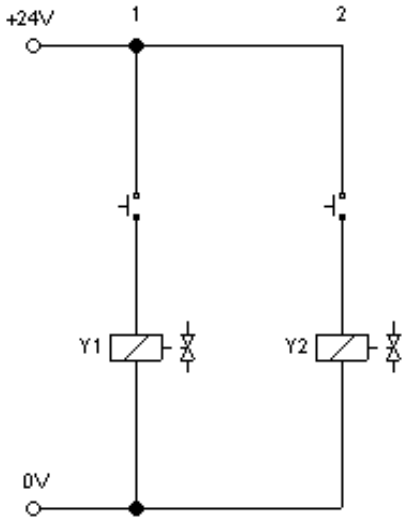
En lo que a nuestro problema se refiere utilizaremos Arduino, como ya se ha mencionado como medio de control de un proceso electroneumático, por lo que es necesario tener conocimientos básicos tando de neumática, electricidad, electrónica y programación; con lo que el alumno pondrá en práctica los conocimientos que ha ido adquiriendo en la asignatura.

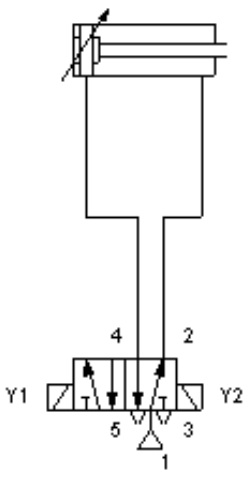
Con este ejercicio en concreto, podrá apreciar aspectos de control sobre un proceso tan rápido como este que describimos, por lo que podrá evaluar otros aspectos como “optimización”, “rendimiento”, “productividad”...ect.

ESQUEMA DE SITUACIÓN

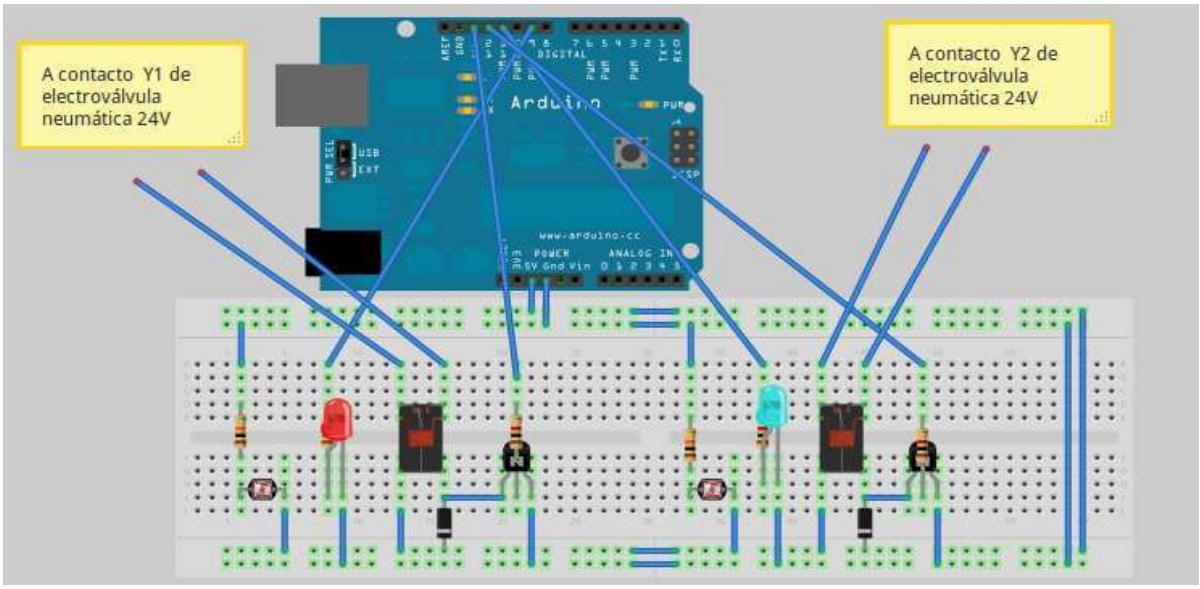


ESQUEMA ELECTRONEUMÁTICO

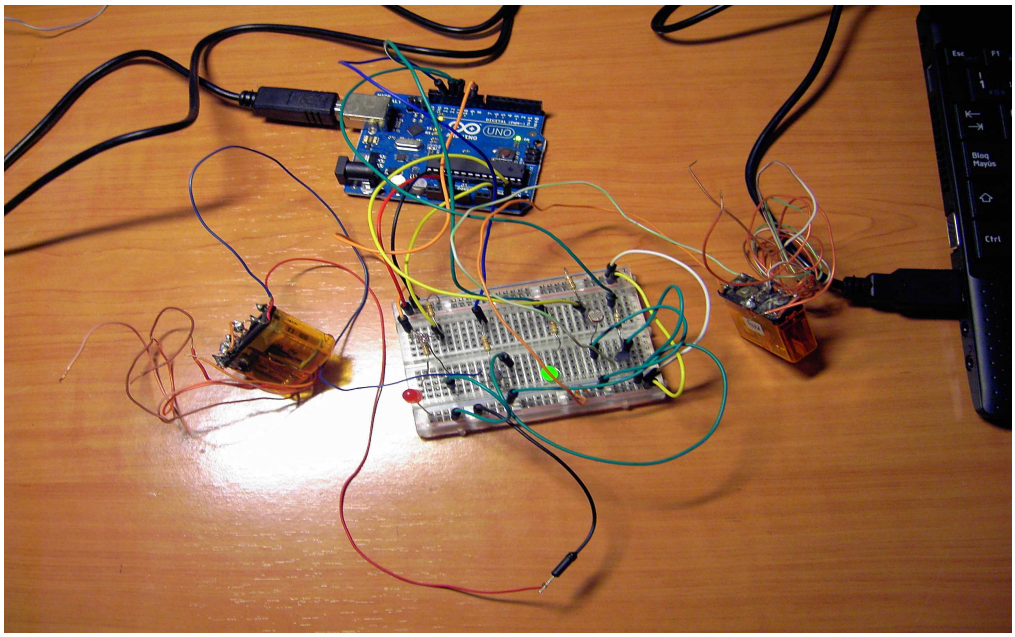




CONEXIONADO ELÉCTRICO



FOTOGRAFÍA DE MONTAJE EN PLACA DE PRUEBAS.



PROGRAMA "C" PARA ARDUINO

```
int ldr1=2;
int ldr2=4;
int led1=9;
int led2=11;
int rele1=12;
int rele2=13;

int avance;
int retroceso;

int cerca1;
int cerca2;

void setup(){
pinMode(ldr1,INPUT);
pinMode (ldr2, INPUT);
pinMode(led1, OUTPUT);
pinMode (led2, OUTPUT);
pinMode (rele1, OUTPUT);
pinMode (rele2, OUTPUT);

Serial.begin(9600);
}
void loop(){
avance = analogRead(ldr1);
retroceso = analogRead (ldr2);
cerca1=map(avance,130,500,1,4);
cerca2=map (retroceso,5, 11,1,4);

Serial.println(avance);
Serial.println(retroceso);

if (cerca1==1){

digitalWrite (rele2, HIGH);
digitalWrite (led2, HIGH);
digitalWrite (rele1, LOW);
digitalWrite (led1, LOW);
}
else{ }

if (cerca2==1){
digitalWrite (rele2, LOW);
digitalWrite (led2, LOW);
digitalWrite (rele1, HIGH);
digitalWrite (led1, HIGH);
}
else{}
}
```

PROYECTO TÉCNICO: NORIA CON LUZ Y MOTOR

JUSTIFICACIÓN

La siguiente unidad didáctica se enmarca en el contexto de la Educación Secundaria Obligatoria como actividad final al curso "La microcontroladora Arduino en Secundaria". Nos proponemos realizar un proyecto técnico "Noria con luz y motor". Aplicaremos los conceptos de electricidad, dibujo, estructuras, mecanismos y materiales, e introduciremos al alumnado en el mundo de la programación y la electrónica a través de la microcontroladora Arduino.
Podríamos proponer el proyecto para un alumnado de 4º de la E.S.O. como colofón final a la Tecnología en Secundaria.

COMPETENCIAS QUE SE TRABAJAN

- Comunicación lingüística (1)
- Matemática (2)
- Conocimiento y la interacción con el mundo físico (3)
- Digital (4)
- Social y ciudadana (5)
- Cultural y artística (6)
- Aprender a aprender (7)
- Autonomía e iniciativa personal (8)

OBJETIVOS

- Realizar bocetos y croquis de piezas y circuitos que compondrán el proyecto.
- Identificar las herramientas y los útiles que se emplean en las operaciones de medida, trazado, aserrado, limado y taladrado.
- Realizar trabajos de construcción con la madera.
- Construir una transmisión para la noria.
- Realizar el cableado para la iluminación, motor y fuente de alimentación.
- Conocer y respetar las normas de seguridad en el empleo de herramientas.
- Reconocer los distintos tipos de unión y acabado de piezas de madera y las herramientas y los útiles que se emplean en cada uno de ellos.
- Conocer el funcionamiento de la microcontroladora Arduino.
- Programar y quemar instrucciones en la microocontroladora Arduino a través de un ordenador para gobernar distintos elementos de un circuito.
- Realizar el cableado para la iluminación, motor y fuente de alimentación de la microcontroladora.
- Realizar un informe del trabajo realizado en el taller.

CONTENIDOS

Conceptos

- Herramientas y útiles de trabajo con madera y material metálico.
- Informes de trabajo: documentos que lo componen.
- Normas de seguridad en el aula taller.
- Transmisión de movimiento.
- Conceptos fundamentales de electrónica.
- Lenguajes de programación.
- La microcontroladora Arduino.

Procedimientos, destrezas y habilidades

- Construcción de un proyecto técnico con madera que sea capaz de moverse con energía eléctrica y tenga iluminación controlada por Arduino.
- Programación de la unidad Arduino para gobernar un motor y un sistema de iluminación con interruptor.
- Elaboración de planos del proyecto haciendo uso del QCad (planos y vistas de las piezas del proyecto) y de Fritzing (planos y esquemas de circuitos).
- Distribución de las tareas dentro de un grupo de trabajo.

- Elaboración de un informe de trabajo.
- Aplicación de las normas básicas de seguridad en el taller.

Actitudes
- Respeto hacia los demás en el trabajo en grupo.
- Fomento de la igualdad de sexos en el reparto de tareas.
- Fomento de la responsabilidad individual dentro del trabajo en equipo.

CRITERIOS DE EVALUACIÓN:
Se valorará la capacidad del alumnado para:
1. Realizar bocetos y croquis de piezas y circuitos que compondrán el proyecto.
2. Identificar las herramientas y los útiles que se emplean en las operaciones de medida, trazado, aserrado, limado y taladrado.
3. Realizar trabajos de construcción con la madera.
4. Construir una transmisión.
5. Realizar el cableado para la iluminación, motor y fuente de alimentación.
6. Conocer y respetar las normas de seguridad en el empleo de herramientas.
7. Reconocer los distintos tipos de unión y acabado de piezas de madera y las herramientas y los útiles que se emplean en cada uno de ellos.
8. Conocer el funcionamiento de la microcontroladora Arduino.
9. Programar y quemar instrucciones en la microocontroladora Arduino a través de un ordenador para gobernar distintos elementos de un circuito.
10. Realizar el cableado para la iluminación, motor y fuente de alimentación de la microcontroladora.
11. Realizar un informe del trabajo realizado en el taller.

INTERDISCIPLINARIEDAD:
- Informática, utilización del ordenador en la redacción del informe y elaboración de planos. Utilización de procesador de textos y programas de diseño asistido.
- Matemáticas en la realización de cálculos para el proyecto: medidas, presupuestos, tiempos, etc.
- Educación plástica y visual, a través del diseño del proyecto.

EDUCACIÓN EN VALORES Y CULTURA ANDALUZA
- Fomentaremos la igualdad y el rechazo a comportamientos sexistas a través del reparto de tareas en el trabajo en pequeño grupo.
- Educación para el consumidor, a través del conocimiento de las distintas formas de presentación de los materiales en función de su utilidad.
- Educación ambiental, a través del conocimiento del gasto en materias primas y de energía que supone el empleo de los distintos materiales.
- Salud laboral, a través del conocimiento de normas de seguridad e higiene en el trabajo con los materiales.

METODOLOGÍA Y SECUENCIACIÓN DE CONTENIDOS – PROYECTO TÉCNICO: NORIA CON LUZ Y MOTOR

SESIÓN	ACTIVIDAD	TIPO	TIEMPO	AGRUP.	RECURSOS	AT. DIVERSIDAD	COMP.	RECURSOS AT. DIVERISDAD	INST. EVAL.
1 y 2	Presentación del proyecto y de sus requerimientos técnicos. Conceptos básicos sobre electrónica Presentación de la microcontroladora Arduino	Transmisiva	2h.	GG	Material electrónico (resistencias, LEDS, etc.). Placa Arduino. Instrucciones del proyecto. Cuaderno de proyecto. Ordenadores portátiles.	-	1, 2, 3, 4 y 7	Si se detectase la necesidad de atención a la diversidad, además de lo planteado en el apartado anterior, se podrá: **1.** Optar por la distribución del alumnado de forma tal que los alumnos con menos dificultades ayuden a los que tienen más (refuerzo). **2.** La dificultad del diseño de la noria y de los circuitos también se puede variar (refuerzo y ampliación).	Cuaderno de proyecto a través de las actividades planteadas. Archivos informáticos de las actividades planteadas. Actitud y trabajo en el taller. Proyecto acabado.
3 y 4	Conceptos básicos sobre el lenguaje de programación que se va a utilizar y el software de programación. Realización de prácticas de programación con Arduino. Montajes sencillos de control de dispositivos electrónicos (Fade, Parpadeo, Coche Fantástico, etc).	Transmisiva Recapitulación	2h.	PG	Instrucciones de programación (guías y trípticos), cuaderno de proyecto, ordenadores portátiles y material electrónico.	Se puede profundizar en la dificultad del ejercicio a voluntad (refuerzo o ampliación)	1, 2, 3, 4, 7 y 8		
5 a 7	Realización de bocetos y planos de las piezas que compondrán el proyecto y de los circuitos en el cuaderno del alumnado y posteriormente por medio de aplicaciones de diseño asistido por ordenador (QCad y Fritzing).	Recapitulación	1h.	I	Tutoriales de las aplicaciones de diseño asistido, cuaderno de proyecto y ordenadores portátiles.	Se puede profundizar en la dificultad del ejercicio (exigencia del boceto más o menos sencillas) a voluntad (refuerzo o ampliación)	1, 2, 4, 6, 7 y 8		

AGRUP: Gran Grupo (GG), Pequeño Grupo (PG), Individual (I)

METODOLOGÍA Y SECUENCIACIÓN DE CONTENIDOS – PROYECTO TÉCNICO: NORIA CON LUZ Y MOTOR

SESIÓN	ACTIVIDAD	TIPO	TIEMPO	AGRUP.	RECURSOS	AT. DIVERSIDAD	COMP.	RECURSOS AT. DIVERISDAD	INST. EVAL.
8	Herramientas de trabajo en el taller y normas de seguridad. Instrucciones para realizar un informe de un proyecto técnico. Instrucciones para la construcción del la noria y sus requerimientos.	Transmisiva	1h.	GG	Material de construcción y herramientas. Instrucciones del proyecto. Cuaderno de proyecto.	-	1, 3, 7	Si se detectase la necesidad de atención a la diversidad, además de lo planteado en el apartado anterior, se podrá: **1.** Optar por la distribución del alumnado de forma tal que los alumnos con menos dificultades ayuden a los que tienen más (refuerzo). **2.** La dificultad del diseño de la noria y de los circuitos también se puede variar (refuerzo y ampliación).	Cuaderno de proyecto a través de las actividades planteadas. Archivos informáticos de las actividades planteadas. Actitud y trabajo en el taller. Proyecto acabado.
9 y 10	Planificación: reparto de tareas y tiempos en el grupo de trabajo. Trabajo con madera: medir, marcar, cortar, limar, lijar y unir las piezas para construir el proyecto. Realización de hojas de procesos con OpenOffice Calc.	Recapitulación	2h.	PG	Cuaderno de proyecto y documento modelo de informes de proyectos.	Se entregará un guión de informe para que el alumnado pueda seguirlo en la redacción del informe.	1, 2, 5, 7 y 8		
11 a 15	Trabajo con madera: medir, marcar, cortar, limar, lijar y unir las piezas para construir el proyecto.	Recapitulación	4h.	PG	Material de construcción y herramientas. Instrucciones del proyecto. Cuaderno de proyecto.	Se prestará especial atención al alumnado en el que detectemos problemas relacionadas con la destreza en el uso de herramientas.	1, 3, 7 y 8		
16 a 18	Trabajo con dispositivos eléctricos y electrónicos: montaje de circuitos necesarios para el funcionamiento de la noria. Programación y quemado de las instrucciones en Arduino	Recapitulación	2h.	PG	Material de construcción y herramientas. Instrucciones del proyecto. Cuaderno de proyecto. Ordenadores portátiles	Se prestará especial atención al alumnado en el que detectemos problemas relacionadas con la destreza en el uso de herramientas.	1, 3, 7 y 8		

AGRUP: Gran Grupo (GG), Pequeño Grupo (PG), Individual (I)

METODOLOGÍA Y SECUENCIACIÓN DE CONTENIDOS – PROYECTO TÉCNICO: NORIA CON LUZ Y MOTOR

SESIÓN	ACTIVIDAD	TIPO	TIEMPO	AGRUP.	RECURSOS	AT. DIVERSIDAD	COMP.	RECURSOS AT. DIVERISDAD	INST. EVAL.
19 a 21	Redacción del informe	Recapitulación	1h.	PG	Ordenadores y cuaderno del alumnado	Los alumnos más aventajados en el uso de las NTIC ayudarán a compañeros que presenten dificultades.	1, 3, 7 y 8		Cuaderno de proyecto a través de las actividades planteadas. Archivos informáticos de las actividades planteadas. Actitud y trabajo en el taller. Proyecto acabado.
22	Evaluación de proyectos y recogida de informes.	Evaluación	1h.	GG	-	-	-		

AGRUP: Gran Grupo (GG), Pequeño Grupo (PG), Individual (I)

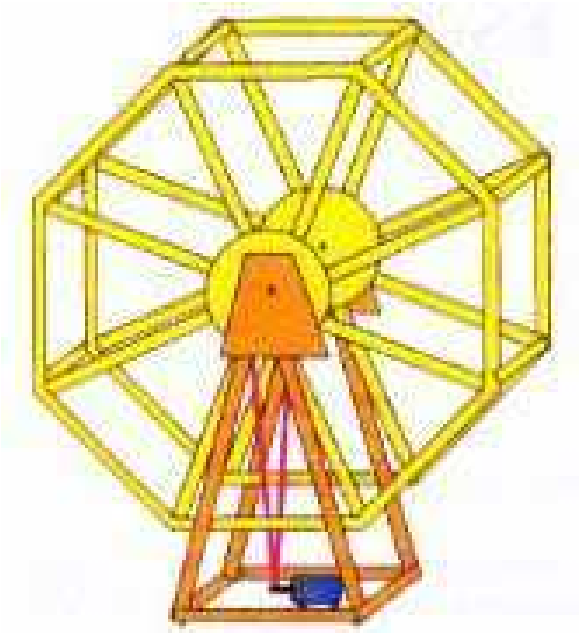
POSIBLE SOLUCIÓN:

REQUISITOS GENERALES:

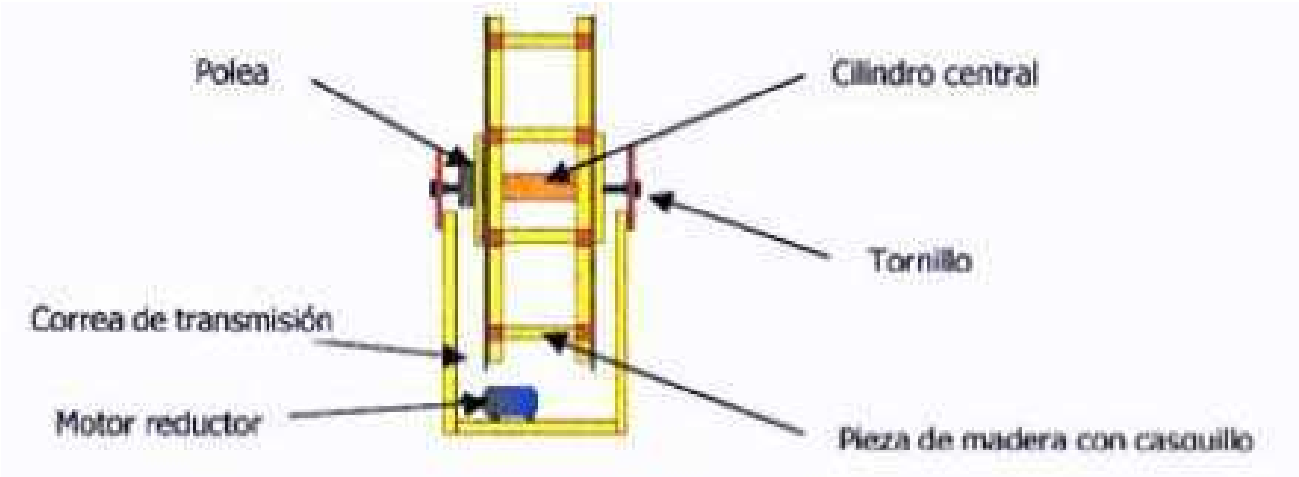
La construcción de la noria será realizada en madera (listones de 10x10 o similar) y será impulsada por un motor reductor gobernado por un interruptor.

Será requisito indispensable la fabricación de la transmisión (en madera).

Deberá incluir iluminación. Cada uno de los sistemas de iluminación será independiente y estará gobernado por un interruptor y programado con Arduino para la realización de un determinado bucle. Como ejemplo mostraremos el bucle "fade" y "coche fantástico" que pueden instalarse en los soportes o en la base de la noria (eliminamos el problema de la iluminación en los radios).



Vista general



Soporte (interior) Soporte (exterior)

CIRCUITOS

Simulamos el funcionamiento de un motor y dos o tres leds (en el montaje final irían tantos como fuesen necesarios). Los efectos perseguidos serán "fade" y "coche fantástico". Como se dijo anteriormente, los circuitos estarán gobernados por interruptores.

MOTOR Y EFECTO COCHE FANTÁSTICO

a) Esquema de programación:

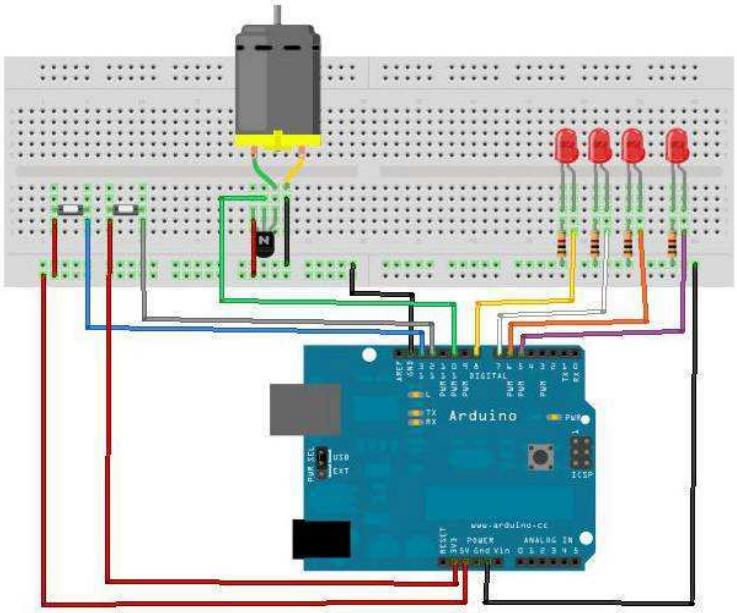
```
int val2;
int val1;
int i;
int tiempo=75;

void setup() {
  pinMode(13, INPUT);
  pinMode(12, INPUT);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(5, OUTPUT);
}

void loop()  {
  val1=digitalRead(13);
   if(val1==HIGH){
    digitalWrite(10, HIGH);
    }
    else {
    digitalWrite(10, LOW);
    }

  val2=digitalRead(11);
    if (val3==LOW){
       for (i = 5; i <= 9; i ++){
       digitalWrite(i, HIGH);
       delay(tiempo);
       digitalWrite(i, LOW);
       }
       for (i = 9; i > 5; i --){
       digitalWrite(i, HIGH);
       delay(tiempo);
       digitalWrite(i, LOW);
       }
    }
    else {
       digitalWrite(9,LOW);
       digitalWrite(7,LOW);
       digitalWrite(6,LOW);
       digitalWrite(5,LOW);
    }
}
```

USAMOS LAS ENTRADAS 13 (INTERRUPTOR MOTOR) Y 12 (INTERRUPTOR LUCES). COMO SALIDAS USAMOS 10 (PARA EL MOTOR) Y 8, 7, 6 Y 5 (PARA LAS LUCES)

CONTROL DEL MOTOR

CONTROL LUCES

b) Esquema de placa de prototipos:



MOTOR Y EFECTO FADE

a) Esquema de programación:

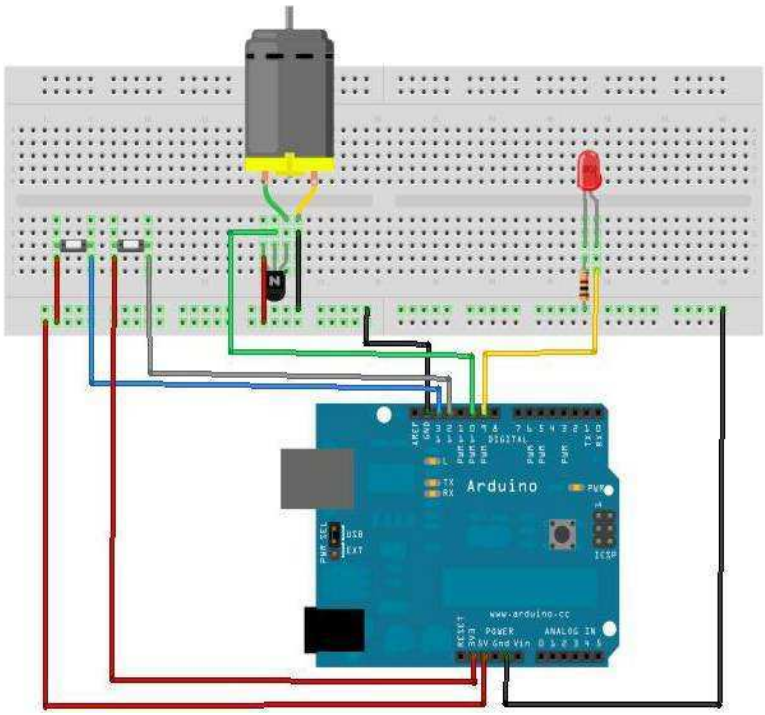
```
int brillo = 0;
int incremento = 5;
int val2;
int val1;

void setup() {
  pinMode(13, INPUT);
  pinMode(12, INPUT);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
}

void loop()  {
  val1=digitalRead(13);

   if(val1==LOW){
    digitalWrite(10, HIGH);
    }
    else {
    digitalWrite(10, LOW);
    }

  val2=digitalRead(12);
   if (val2==LOW){
      analogWrite(9,     brillo);
     brillo = brillo + incremento;
      if (brillo == 0 || brillo == 255) {
      incremento = -incremento;
          }
      delay     (30);
  }
  else {
      digitalWrite(9,LOW);
  }
}
```

USAMOS LAS ENTRADAS 13 (INTERRUPTOR MOTOR) Y 12 (INTERRUPTOR LUCES). COMO SALIDAS USAMOS 10 (PARA EL MOTOR) Y 9 (PARA LAS LUCES)

CONTROL DEL MOTOR

CONTROL LUCES

b) Esquema de placa de prototipos:



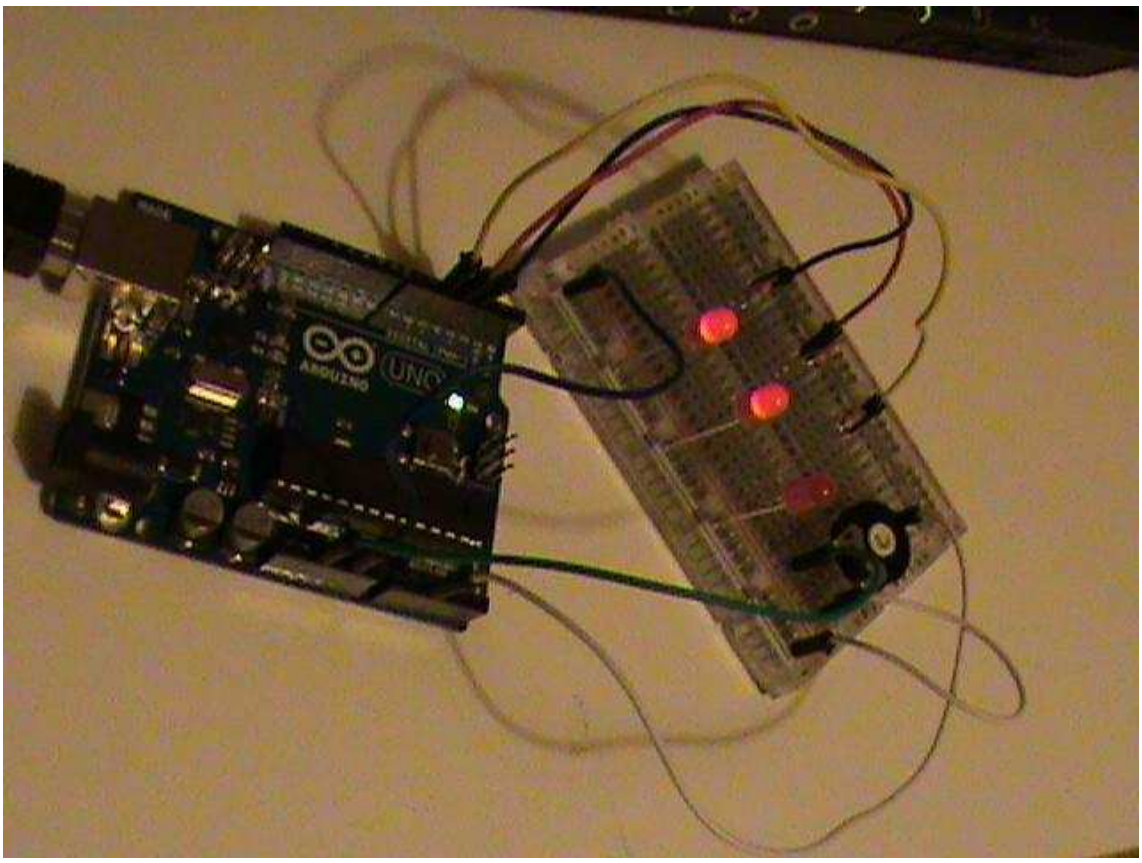
GUÍA DE RECURSOS PARA EL ALUMNADO:

Páginas Web:

- Arduino:
  http://www.arduino.cc/
  http://www.arduino.cc/es/
- Fritzing:
  http://fritzing.org/
- QCad:
  http://www.qcad.org/qcad/manual_reference/es/ (manual de uso)
- OpenOffice:
  http://es.openoffice.org/

Documentos pdf:

- Acordeón Arduino. Referencia de Lenguaje de Programación de Arduino.
- Arduino Programming Notebook español.
- Guía del usuario de Arduino en español.



MICROCONTROLADORA ARDUINO.
SECUENCIA DE LEDS.
4º ESO

BRAVO AVÍS, JUAN ANDRÉS

UNIDAD DIDÁCTICA: SECUENCIA DE LEDS.

INDICE:

1. **INTRODUCCIÓN.**
2. **OBJETIVOS.**
3. **CONTENIDOS.**
4. **COMPETENCIAS BÁSICAS.**
5. **METODOLOGÍA.**
6. **CRITERIOS, PROCEDIMIENTOS Y ESTRATEGIAS DE EVALUACIÓN.**
7. **TEMAS TRANSVERSALES.**
8. **MEDIDAS DE ATENCIÓN A LA DIVERSIDAD.**
9. **MATERIALES Y RECURSOS.**
10. **ANEXO.**

1. INTRODUCCIÓN. La presente unidad se encuentra ubicada en cuarto curso de la secundaria y pretende abordar integrando los contenidos relacionados con electrónica, tecnologías de la información y comunicación y robótica.

En dicha unidad se pretenden interrelacionar la electrónica y sus componentes principales, y el uso del ordenador para controlar una placa protoboard, mediante un lenguaje de programación, lo que constituirá el eslabón para aplicar los contenidos, referente a robótica.

Es necesario por tanto que dicha unidad se desarrolle en segunda evaluación, con la idea de tener todos los espacios y recursos necesarios preparados para abordarlos.

2. OBJETIVOS. Los objetivos definidos y concretados a partir de los generales, marcados en RD 1631/2004 para este cuarto curso son los siguientes:

1. Planificar proyectos tecnológicos sencillos, en grupo e individualmente, trabajando de forma ordenada y metódica, anticipando los recursos necesarios y elaborando la documentación pertinente
2. Utilizar las técnicas básicas de dibujo y comunicación gráfica asistida por ordenador necesarias para la actividad tecnológica valorando la importancia de las mismas en diferentes ámbitos de la sociedad.
3. Utilizar las tecnologías de la información y la comunicación para localizar y seleccionar información contenida en diversas fuentes y soportes, organizarla y presentarla y también como elemento de acercamiento a la Comunidad Autónoma de Andalucía y al resto del mundo.
4. Relacionar la evolución de los objetos técnicos y los avances tecnológicos con el desarrollo de los conocimientos científicos y tecnológicos, las estructuras socio-económicas y la disponibilidad de distintas energías en Andalucía.
5. Reconocer en un plano y en el contexto real los distintos elementos que forman las instalaciones de una vivienda entendiendo su funcionamiento y potenciando el buen uso para conseguir ahorro energético.
6. Analizar sistemas electrónicos, neumáticos, hidráulicos y relacionados con la robótica para comprender su funcionamiento y finalidad, relacionando esto con la mejor forma de usarlos y controlarlos.
7. Valorar críticamente el uso de las tecnologías y su influencia sobre el medio ambiente y la sociedad, especialmente en Andalucía.
8. Utilizar la terminología, la simbología y los recursos gráficos adecuados para comunicar ideas y soluciones técnicas organizando la información recogida en las diversas búsquedas y elaborando documentos para presentarla correctamente.
9. Analizar las necesidades individuales y sociales más próximas y las soluciones tecnológicas que se dan desde la Comunidad Autónoma de Andalucía.
10. Participar en debates y coloquios relacionados con las repercusiones de determinados problemas técnicos en el entorno de Andalucía manifestando preparación respecto a los contenidos tratados y respeto por las opiniones fundamentadas

Principalmente se desarrollarán los objetivos 1, 2, 3, 7 y 8.

3. CONTENIDOS.

Bloque 2. Electrónica.

Estructura y funcionamiento de los equipos electrónicos.

Análisis de los componentes de un equipo electrónico: componentes discretos, integrados y elementos auxiliares.

Análisis de los mecanismos de control y regulación: la conmutación electrónica.

Simulación de circuitos por ordenador.

Uso y mantenimiento de equipos electrónicos.

Reconocimiento de la importancia de los sistemas electrónicos en nuestra sociedad y en Andalucía.

La electrónica digital.

Las señales eléctricas como portadoras de información.

Diferenciación entre señal analógica y digital.

Reconocimiento de los tipos de señales que se utilizan para transmitir información**.**

Descripción de situaciones de la vida cotidiana en las que estén presentes señales analógicas y señales digitales.

Representación simbólica de cantidades mediante los sistemas de numeración.

Análisis de problemas tecnológicos mediante el uso de la lógica binaria.

Respeto por la normas de seguridad en el manejo de herramientas y útiles para la electrónica.

Uso de los elementos de protección personal al trabajar con útiles y herramientas para la electrónica.

Interés por descubrir algunas aplicaciones de la electrónica.

La electrónica y sus repercusiones en la economía de Andalucía.

Bloque 3. Tecnologías de la comunicación.

Los sistemas de telecomunicación.

La información y las señales.

La transmisión de la información

Las comunicaciones por cable.

Las comunicaciones inalámbricas.

Los protocolos de transmisión.

Tipos de conexión a Internet.

Redes informáticas de gran alcance.

Control protección de datos.

Comunicaciones móviles.

Utilización de las tecnologías de la comunicación.

Búsqueda y selección de información en la Red.

Descarga de archivos desde páginas web.

Utilización del correo electrónico.

Reconocimiento del papel de Internet en la sociedad actual.

Uso de Internet para la búsqueda de información.

Uso de las tecnologías de la comunicación en Andalucía.

Valoración de las ventajas que la comunicación globalizada aporta a Andalucía y a el Estado

Respeto por las diferentes creencias y opiniones que puedan encontrarse en las páginas web de Internet.

Respeto de las normas de uso y manejo de los equipos informáticos.

Bloque 4. Control y robótica.

Los sistemas automáticos.

La percepción del entorno y los sensores.

Sensor de contacto. Interruptor de final de carrera.

Sensor magnético. Interruptor reed.

Sensor de humedad.

Sensor de temperatura. NTC y PTC.

Sensor de luz. LDR.

Sensores de infrarrojos. Optoaclopadores.

Aplicaciones de los sensores en la industria y la medicina.

Diseño y construcción de un circuito impreso.

Las máquinas automáticas y los robots en la historia.

Los robots industriales.

Los robots móviles.

Identificación de los componentes necesarios para la construcción de un robot.

Análisis del presente y futuro de la robótica.

Interés por conocer las aplicaciones de los robots en la industria.

Ejemplificación de los beneficios que para nuestra sociedad aporta el desarrollo de la robótica

Uso del ordenador como elemento de programación y control.

El lenguaje de programación C. Las primitivas.

Interpretación de programas sencillos escritos en C.

Ventanas y botones. Procedimiento, variable y recursividad.

La simulación de luces. La simulación de movimiento.

Valoración y reconocimiento de la importancia de la robótica en la sociedad actual y en Andalucía.

Funcionamiento y elementos del ordenador.

Señales analógicas y digitales.

Transmisión de la información por medio de señal eléctrica.

Tratamiento de la información numérica adquirida. Programas de control.

Valoración de la trascendencia de las nuevas tecnologías en la vida diaria.

Actitud ordenada a la hora de manipular los componentes de un ordenador.

Respeto de las normas de uso y manejo de los equipos informáticos.

Destreza en la manipulación de los sistemas operativos.

4. COMPETENCIAS BÁSICAS. Los nuevos currículos de la ESO han identificado ocho competencias básicas para el conjunto de la escolaridad obligatoria. Son las siguientes:

Comunicación lingüística.

Matemática.

Conocimiento y en la interacción con el mundo físico.

Tratamiento de la información y competencia digital.

Social y ciudadana.

Cultural y artística.

Aprender a aprender.

Autonomía e iniciativa personal.

La contribución a la competencia en comunicación lingüística se realiza a través de la adquisición de vocabulario específico, necesario para abordar la unidad, y especialmente para justificar mediante un informe los resultados de la práctica.

La competencia matemática, se trabaja mediante el manejo de unidades eléctricas y sus derivados y analizando los resultados de las salidas en la placa protoboard. También el uso de un lenguaje específico de programación requiere un esfuerzo para convertir los objetivos e intereses en un código comprensible para el ordenador.

La Competencia en el conocimiento e interacción con el medio físico y natural está relacionada con el conocimiento y comprensión de objetos, procesos, sistemas y entornos tecnológicos y a través del desarrollo de destrezas técnicas y habilidades para manipular objetos con precisión y seguridad. El estudio de las señales analógicas y digitales, y su aplicación al entorno del alumno en diferentes sistemas, constituye un eje importante de la unidad.

La competencia en el tratamiento de la información y la competencia digital, se centra en el conocimiento de los ordenadores y adquisición de destrezas básicas asociadas a un uso suficientemente autónomo de estas tecnologías. En esta unidad adquiere vital importancia al utilizar un ordenador como mecanismo de control de una tarjeta Arduino.

A la adquisición de la competencia de aprender a aprender se contribuye por el desarrollo de estrategias de resolución de problemas tecnológicos, en particular mediante la obtención, análisis y selección de información útil para abordar un proyecto. Por otra parte, el estudio metódico de objetos, sistemas o entornos proporciona habilidades y estrategias cognitivas y promueve actitudes y valores necesarios para el aprendizaje.

La competencia en Autonomía e iniciativa personal pierde valor, dado que el proyecto será guidado y apenas se plantearán alternativas. No obstante se permitirá en el informe final, realizar una aplicación tecnológica de dicho montaje.

5.- METODOLOGÍA. La metodología de trabajo se puede desglosar en los siguientes apartados:

Primero se realizará una exposición de los componentes electrónicos a utilizar, características y compresión de su funcionamiento aislado e integrado en distintos circuitos. Esta etapa durará 6 horas lectivas, y requerirá tanto el aula teórica como el taller para realizar montajes sencillos.

Segundo. Descripción de los elementos de un ordenador (hardware y software), centrándose posteriormente en el programa Arduino, que deberán instalar en el

ordenador. Para abordar esta parte se accederá al aula de informática donde se describirá y preparará el programa de control a utilizar. Tiempo estimado 3 horas.

Tercero. El siguiente paso se desarrollará en el taller donde se montará el circuito, que posteriormente se conectará al ordenador para controlar el funcionamiento del mismo, y se comprobará la interacción entre máquina y ordenador. En caso de alumnado muy motivado, o avanzado se plantearán posibles modificaciones sobre el programa base. Tiempo estimado 3 horas.

Cuarto. Por último un informe que se irá esbozando conforme se avanza en la práctica, constituirá el culmen del proceso, aparte de una prueba escrita de los contenidos abordados. Tiempo estimado 2 horas en clase y 2 horas en casa.

Quinto. Como medida de refuerzo a la lectura y desarrollo de la competencia en comunicación lingüística y del tratamiento de la información y competencia digital se hará una lectura comprensiva de un texto relacionado con la robótica en Andalucía y el Mundo, y sus repercusiones sociales y económicas. A partir de él se obtendrán las ideas fundamentales, se redactarán resúmenes en el cuaderno y se abrirá un debate en clase. Tiempo estimado, una hora de clase.

6.- CRITERIOS, ESTRATEGIAS Y PROCEDIMIENTOS DE EVALUACIÓN.

Los criterios de evaluación serán los siguientes.

1. Explicar el funcionamiento de un circuito electrónico, conociendo la simbología, diferenciando la función de cada componente.
2. Planificar un proyecto tecnológico sencillo, en grupo e individualmente, mediante la elaboración de un plan, reparto de tareas y distribución temporal de las mismas.
3. Respetar las normas de seguridad en el manejo de herramientas y útiles para el montaje de un circuito electrónico.
4. Buscar información utilizando Internet de forma crítica.
5. Configurar un ordenador para su acceso a Internet.
6. Explicar el funcionamiento de sensores, actuadores y la aplicación de la realimentación en dispositivos de control.
7. Identificar los componentes necesarios para la construcción de un robot analizando la función que desempeñan.
8. Interpretar programas sencillos escritos en C.
9. Analizar los beneficios y problemas derivados de la actividad tecnológica en Andalucía.
10. Reconocer la importancia de la tecnología en la sociedad actual y sus repercusiones positivas en la calidad de vida.

Las estrategias y procedimientos serán los siguientes:

- Prueba escrita sobre el funcionamiento de los componentes electrónicos y su funcionamiento. Individual.

- Informe final de la práctica. Al final del trabajo se entregará un documento en folio, justificando los resultados, documentando los pasos seguidos en clase como un diario. Individual.

- Cuaderno de clase. Se revisará el cuaderno para comprobar el seguimiento diario de las actividades de taller e informática. Individual.

- Montaje de la placa protoboard e interconexión con el ordenador. Como culmen del proceso, la efectiva comunicación entre la placa protoboard y el ordenador constituirá otra pieza clave de evaluación. Trabajo en grupo.

7.- TEMAS TRANSVERSALES. Los temas transversales que se tratarán en esta unidad son:

Educación para la paz. Las actividades grupales son el perfecto sistema para fomentar la tolerancia, el respeto y fomentar la convivencia. El trabajo en equipo permite obtener óptimos resultados siempre que establezcamos un orden y respetemos con equidad las opciones planteadas por los miembros del grupo.

Educación del consumidor. En esta unidad, se permite al alumno actuar como diseñador de sistemas de control, comprendiendo el valor de los materiales y

Educación para la igualdad de oportunidades entre sexos. Las enseñanzas técnicas y los ciclos formativos han sido convencionalmente asignados al hombre, y es un trabajo continuo y diario, el fomentar el estudio de dichas enseñanzas al mundo femenino. De hecho se potenciará la creación de grupos de trabajo mixtos.

Educación cívica y moral. El mundo de la robótica y la informática constituirán pilares fundamentales en la sociedad del futuro y saber integrarlas en el desarrollo de la humanidad debe ser un objetivo importante de la unidad. Se debe razonar y analizar las ventajas e inconvenientes estas dos ramas de la tecnología.

Educación para el conocimiento de la Comunidad Autónoma. El estudio y análisis del documento final sobre la situación de Andalucía y el Mundo en referencia a la robótica constituye el eje de desarrollo de este tema transversal. Permitirá abrir un debate sobre las ventajas e inconvenientes del desarrollo tecnológico en este campo.

8.- MEDIAS DE ATENCIÓN A LA DIVERSIDAD Y PROGRAMAS DE REFUERZO.

En caso de un grupo con ciertas dificultades de aprendizaje como los Programas de diversificación se limitará el número de componentes electrónicos, elaborando una lista de actividades sencillas y detalladas para llevar a cabo el montaje de la placa, y conexión al ordenador. Incluso se puede dar la secuencia de actividades en el programa ya redactadas, y en la elaboración del informe ser menos exigente.

Alumnos con materias del departamento suspensas. En caso de alumnos con materias suspensas, no debe existir complicación, pues son contenidos no abordados en tercer curso.

9.- MATERIALES Y RECURSOS DIDÁCTICOS.

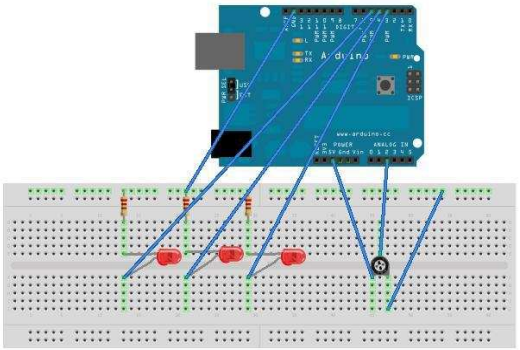
Para el desarrollo de los contenidos teóricos de la materia se requiere el libro de texto (facilitado por la consejería con el programa de gratuidad) y cuaderno. 4º ESO. Tecnología. Editorial SM. González, Ángel y otros. ISBN 978-84-675-3035-3

Para la aplicación práctica de esos contenidos son necesarios los espacios de taller e informática del centro y sus recursos. Por tanto habrá que prever la asignación de los grupos los días especificados para abordarlos.

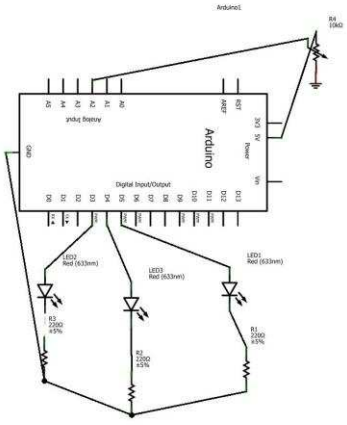
10. ANEXO.

A continuación se detalla el programa a desarrollar en clase y el montaje en la protoboard y el esquema eléctrico. La secuencia de comandos en arduino es la siguiente:

```
int tiempo;

int i;

int potPin=2;

int val;

        Void setup()[

pinMode (3,OUTPUT);

pinMode(4, OUTPUT);

pinMode(5, OUTPUT);]

        Void loop()[

Val= analogRead(potPin);

tiempo=map(val, 0, 1024, 50, 500);

for (i=3; i<=5; i++)[

        digitalWrite (i, HIGH);

        delay (tiempo);]

for (i=5; i>=3;i--)[

        digitalWrite (i, LOW);

        delay (tiempo);]

]
```



Diseño de placa protoboard.



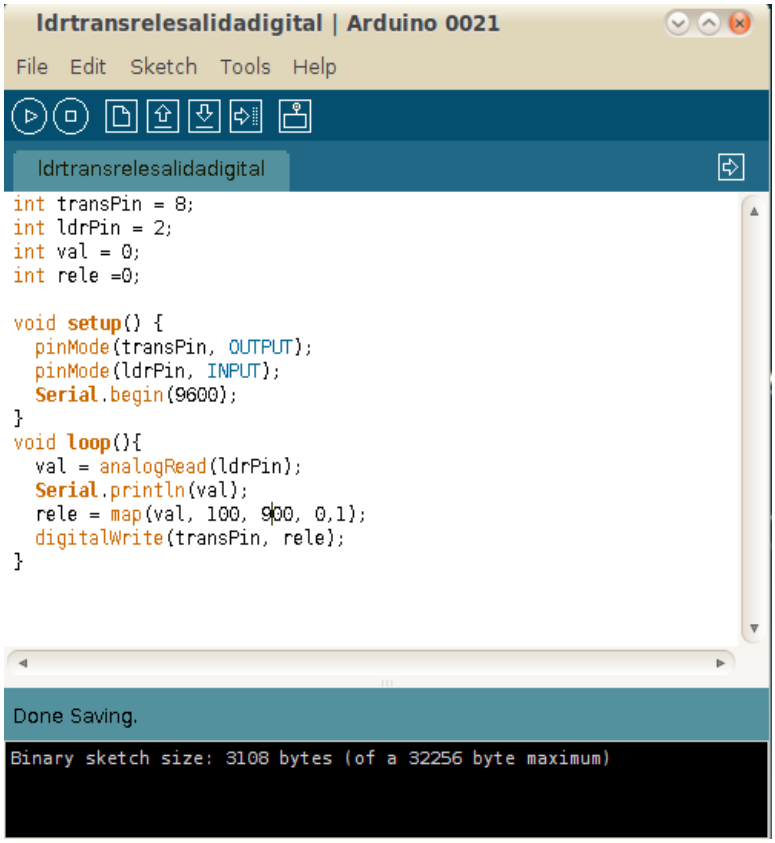
Esquema eléctrico

<u>UNIDAD DIDÁCTICA:</u> ROBOT SEGUIDOR DE LUZ CONTROLADO CON PLACA ARDUINO UNO.

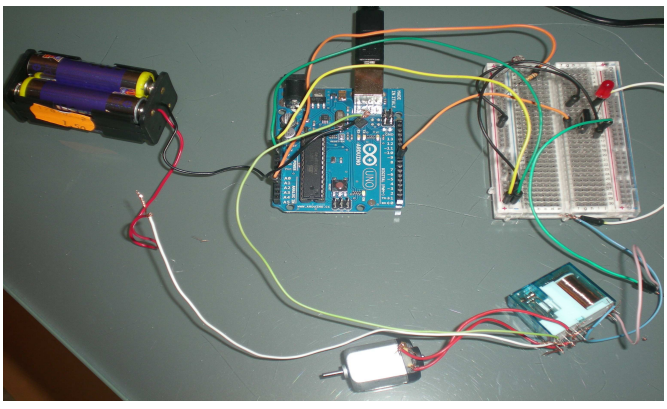
Propuesta de trabajo:

Partiendo de un proyecto del curso pasado para 4º E.S.O. en la materia de Tecnología, robot seguidor de luz con placa de circuito electrónico, voy a proponer sustituir el control con la placa de circuito electrónico por el control con la placa arduino.

Para ello a continuación mostraré el programa con el que cargaremos la placa arduino:



El montaje de la placa arduino con el programa anterior cargado que sustituiría a la placa de circuito electrónico original del robot sería el siguiente:



PROGRAMACIÓN DE AULA

UNIDAD DIDÁCTICA: ROBÓTICA

PROPUESTA DE PROYECTO CON ARDUINO.

ASCENSOR DE CUATRO PLANTAS.

ALUMNADO:

4º ESO / 1º Bachillerato.

TEMPORIZACIÓN:

Diez sesiones para el montaje físico de la propuesta.

Seis sesiones para realización del programa y depuración.

(Suponemos que el alumno ya realiza programas sencillos con la placa microcontroladora Arduino).

OBJETIVOS DEL PROYECTO:

- Que el alumno se afiance en la programación en general y en el manejo de esta tarjeta microcontroladora en particular.
- Experimentar con sistemas automáticos, sensores, actuadores, toma de decisiones y se familiarice con el concepto de entradas/salidas de un sistema automatizado.
- Se potencie la autoestima del alumno al llegar a controlar el ascensor con su propio programa.
- Que use el ordenador como elemento de programación y control.
- Que participe de manera autónoma y creativa en la realización del trabajo, potenciando una actitud crítica y abierta a propuestas de otros miembros de su equipo.

COMPETENCIAS QUE SE TRABAJAN:

Tratamiento de la Información y competencia digital

Se trabaja específicamente con ordenadores realizando el propio programa y montador el circuito.

Aprender a aprender e Iniciativa personal

El propio desafío implica un incentivo suficiente para que trate de solucionar el problema buscando sus propios recursos.

Competencia Matemática

Planteamiento del problema y resolución del mismo. Uso de algoritmos y razonamiento necesario para la consecución del objetivo.

Competencia lingüística

Exposición al resto de los compañeros del trabajo realizado y redacción del proyecto correspondiente.

DEFINICIÓN DE LA PROPUESTA:

Se propone, a un alumnado de cuarto de ESO o de Primero de Bachillerato, el control de un ascensor de cuatro plantas. (Desde la planta baja a la tercera). El funcionamiento será el propio de un ascensor: Cuando se llame desde una planta acudirá a ella.

El ascensor, que se montará físicamente, dispondrá de cuatro pulsadores para realizar la llamada desde cada una de las plantas. Para la detección de planta utilizaremos un relé Reed en cada una de ellas, convenientemente situado sobre la estructura de madera del ascensor. También se colocará estratégicamente un pequeño imán, en el propio ascensor, que irá activando los relé Reed a su paso por cada una de las plantas.

Como salida utilizaremos un pequeño motor de corriente continua, con una caja reductora, al que haremos girar en los dos sentidos de rotación. Mediante una cuerda y una pequeña polea haremos subir o bajar la cabina.

El control de todo el sistema se llevaría a cabo mediante la placa microcontroladora "Arduino".

UNA SOLUCIÓN A LA PROPUESTA DEL PROYECTO

Se propone una posible solución al proyecto para ser analizada una vez que los alumnos hayan trabajado sobre el suyo propio o bien encuentren dificultades en la solución del mismo.

COMENTARIOS

Como se comentará en el propio programa se utilizarán como entradas los pines 2 a 9. De los cuales del 2 a 5 serán los correspondientes a las llamadas desde planta. Los pines 6 a 9 serán los detectores de planta. (Esto es el ascensor se encontrará situado en esa planta). El Pin 2 corresponderá a la planta baja y el Pin 5 a la tercera planta. Igualmente en los detectores de planta el Pin 6 detectarán que el ascensor se encuentra en la planta baja y el Pin 9 en la planta tercera.

Como salida se utilizarán los Pines 12 y 13.

Haciendo un pequeño resumen:

Las entradas serían:

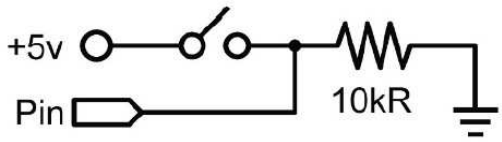
PIN 2	Llamada desde planta baja.
PIN 3	Llamada desde planta primera.
PIN 4	Llamada desde planta segunda.
PIN 5	Llamada desde planta tercera.
PIN 6	Detector de posición planta baja.
PIN 7	Detector de posición planta primera.
PIN 8	Detector de posición planta segunda.
PIN 9	Detector de posición planta tercera.

Y las salidas:

PIN 12	PIN 13	ESTADO DEL MOTOR
0	0	Motor parado. (Bloqueado).
0	1	Motor sube. (Gira a la derecha).
1	0	Motor baja. (Gira a la izquierda).

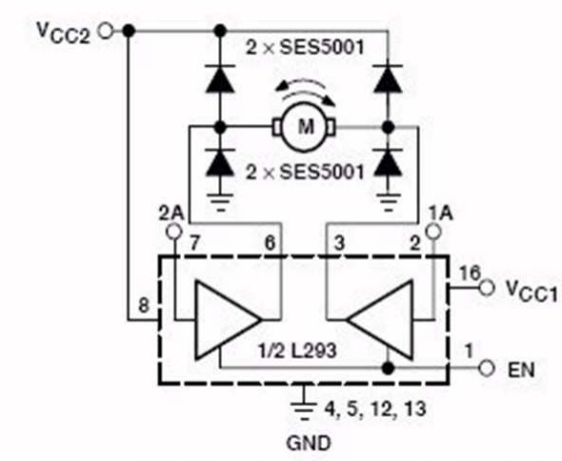
CIRCUITO ELÉCTRICO

Para cada una de las entradas se ha realizado el siguiente montaje:



El interruptor que aparece en el esquema representa en unos casos al pulsador y en otros a los contactos del relé reed que corresponda.

Para las salidas (Pines 12 y 13) emplearemos este otro montaje:



EN	1A	2A	FUNCCION
H	L	H	Gira a la derecha
H	H	L	Gira a la izquierda
H	L	L	Parada rápida del motor
H	H	H	Parada rápida del motor
L	X	X	Parada rápida del motor

L = low, H = high, X = don't care

Las entradas 1A y 2A se conectan respectivamente a las salidas de los Pines 12 y 13.
La entrada EN (Enable=Habilitación) irá permanentemente a 5V. De esta manera el control no quedará nunca con los terminales libres (alta impedancia).
Mediante este IC nos aseguramos una alta impedancia a la salida de la tarjeta microcontroladora y una fuerte corriente de salida hacia el motor.
La tensión de Vcc1 es de 5V.
Los diodos que aparecen en el esquema protegen al integrado y a la propia tarjeta microcontroladora de los posibles picos de tensión, de carácter inductivo, que pudieran generarse en el motor.

La hoja de características de este IC es la siguiente:

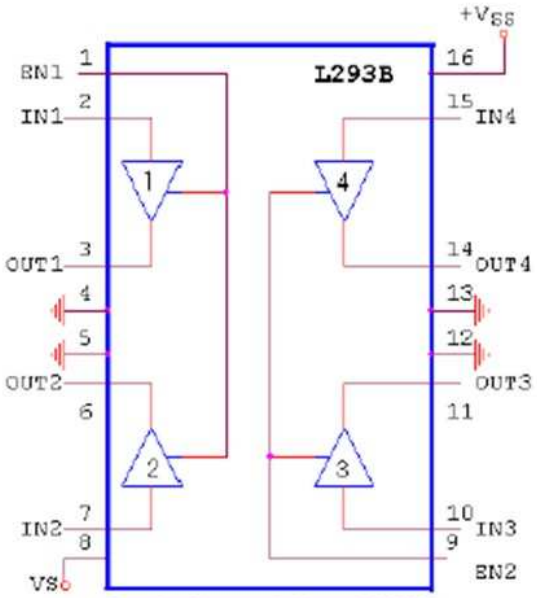
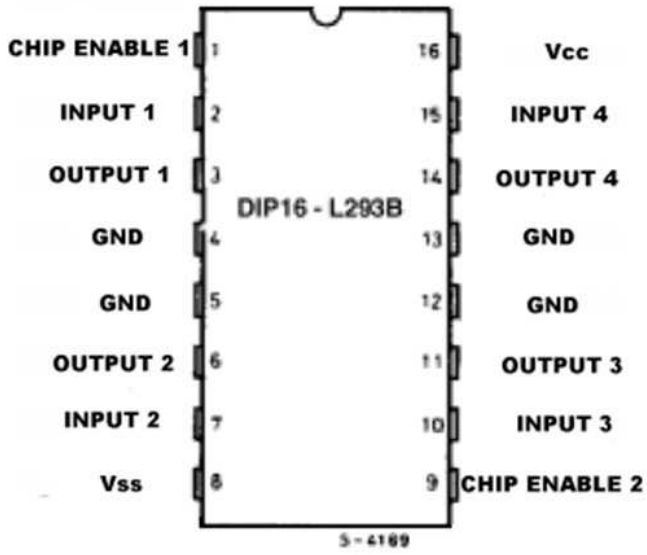


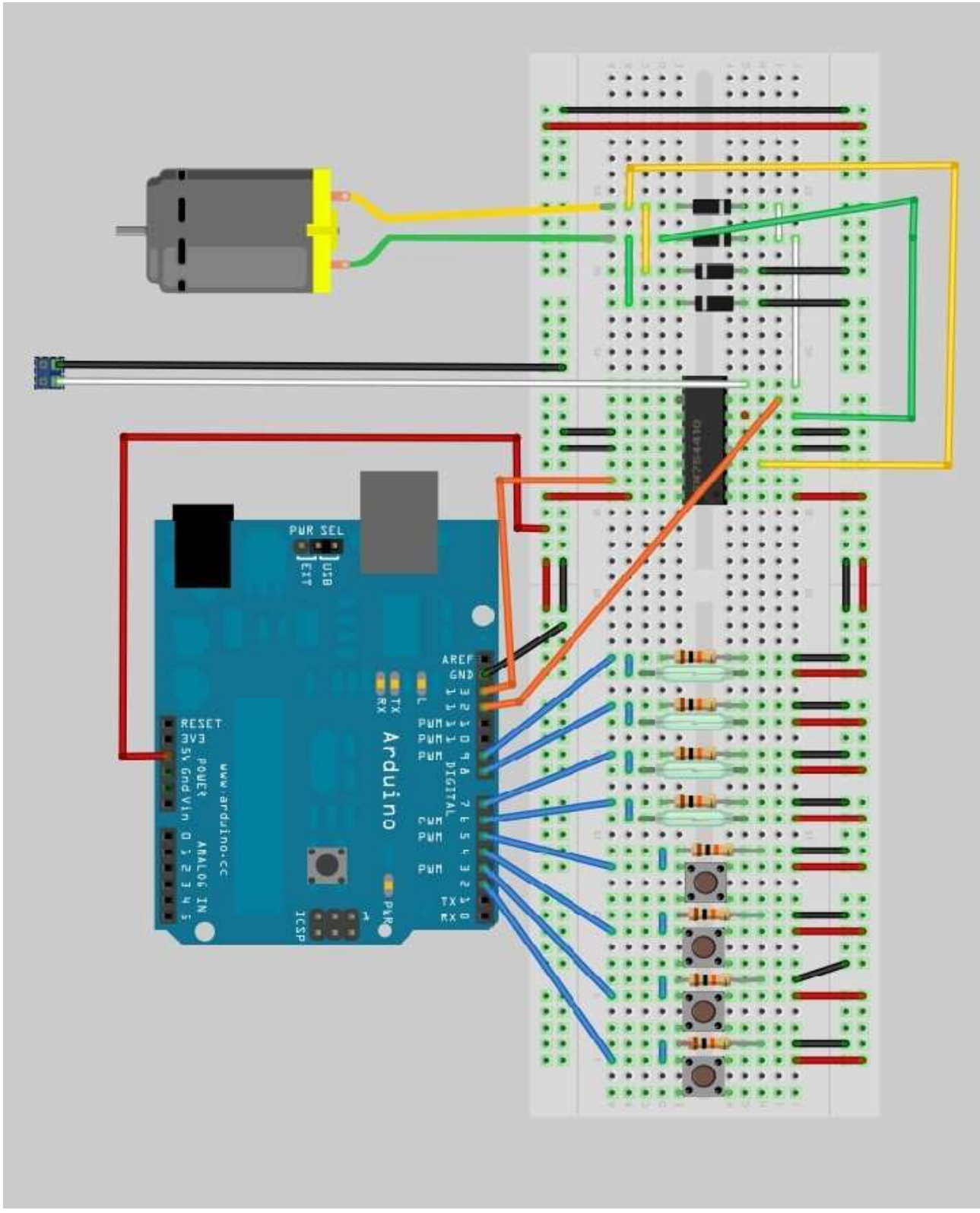
Figura 3.- Diagrama de bloques del L293B



La tabla de funcionamiento para cada uno de los driver es la siguiente:

V_{INn}	V_{OUTn}	V_{ENn}	Donde:
H	H	H	
L	L	H	H = Nivel alto "1"
H	Z	L	L = Nivel bajo "0"
L	Z	L	Z = Alta Impedancia

Nota: En el esquema realizado con Fritzing aparece un IC distinto (es el 754410).
Como el patillaje coincide y las características también no lo he cambiado en este esquema.


PWR SEL
USB
EXT
AREF
GND
RESET
3V3
5V
GND
Vin
POWER
ANALOG IN
DIGITAL
PWM
TX
RX
L
ON
ICSP
www.arduino.cc
Arduino

PROGRAMA

```
/* ascensor de 4 plantas;  */


int n = 0;            //contador para inicialización
int llamada = 0;      // ¿hay llamada?
int cual = 0;         //desde donde se llama
int posicion = 2;     //donde está el ascensor



void motorSube (int  donde) {
  // sube el motor hasta el piso correspondiente "el PIN"
  // donde es el número de PIN al que debe subir
    while (digitalRead (donde)==0){
    digitalWrite (12, LOW);
    digitalWrite (13, HIGH);
    }
    digitalWrite (13, LOW);
    digitalWrite (12, LOW);
    posicion = donde;

}
void motorBaja (int donde) {
   // baja el motor hasta el piso correspondiente
   while (digitalRead (donde)==0){
    digitalWrite (12, HIGH);
    digitalWrite (13, LOW);
    }
    digitalWrite (13, LOW);
    digitalWrite (12, LOW);
    posicion = donde;
}

int hayLlamada (){
  /* compruebo si hay llamada. devuelve número de PIN al que hay que ir, o "0" si no hay
llamada */
  int puerto = 0;

  if (digitalRead (2)==1) {puerto=2;}  //piso 0
  if (digitalRead (3)==1) {puerto=3;}  //piso 1
  if (digitalRead (4)==1) {puerto=4;}  //piso 2
  if (digitalRead (5)==1) {puerto=5;}  //piso 3
  return puerto;
}
```

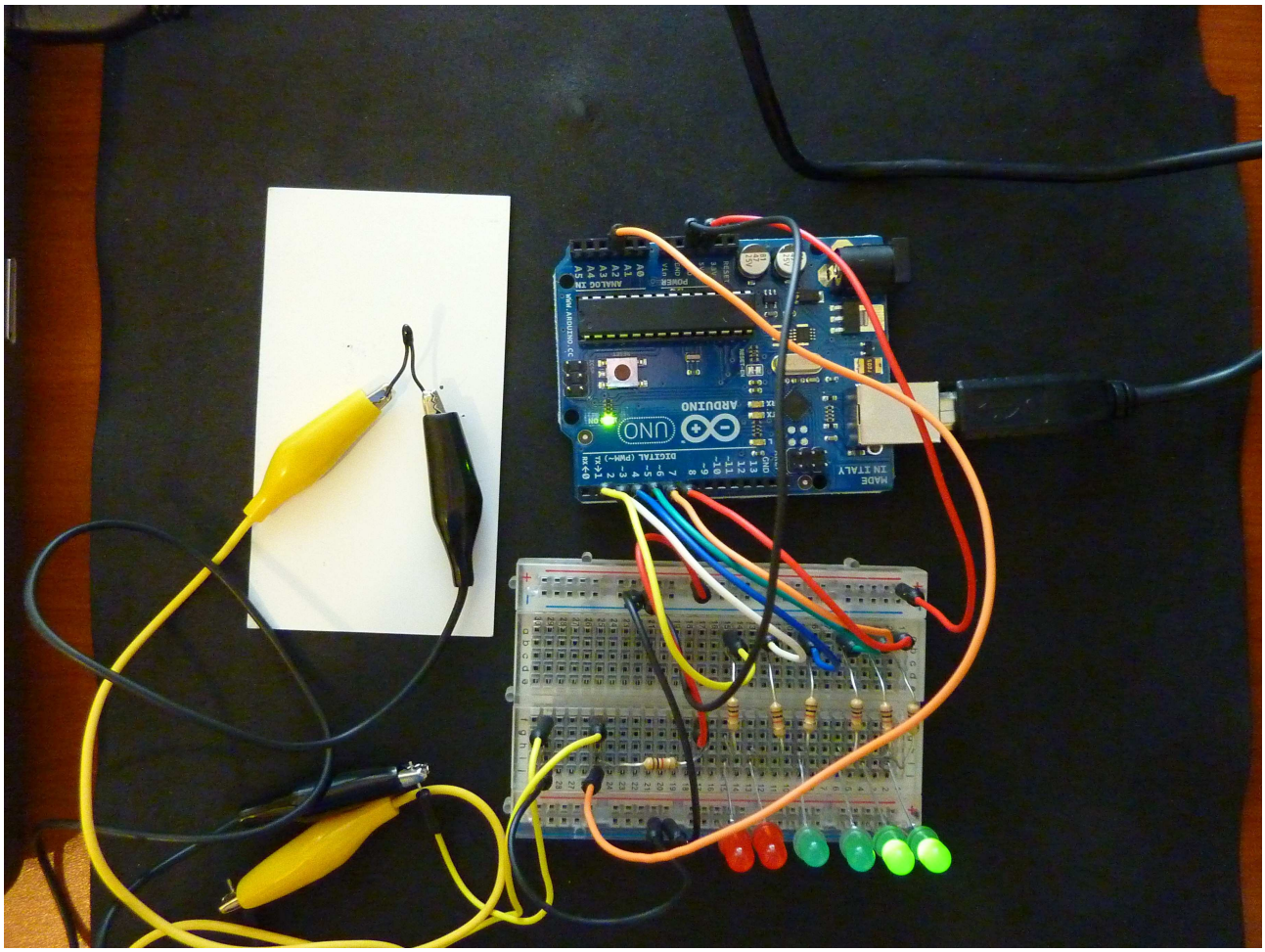
```
void setup() {
  // los PINES 2 a 5 son las llamadas de plantas 0 a 3;
  // los PINES 6 a 9 son los detectores de planta 0 a 3;
  // PINES 12 y 13 (0, 0) Motor parado, (1, 0) Motor sube, (0, 1) Motor baja
  for (n = 2; n < 11; n++ ) {
    pinMode (n, 1);
  }
  pinMode (12, OUTPUT);
  pinMode (13, OUTPUT);

}


void loop() {
  int x;
  while (hayLlamada () == 0);      //mientras no haya llamada espera a que llamen
  x=hayLlamada();
    switch (x){
    case 2: motorBaja (6); break; //llaman de planta baja
    case 3:                       //llaman 1º planta
    if (posicion < 3) {motorSube (7);}
      else motorBaja (7); break;
    case 4:                       //llaman 2º planta
    if (posicion < 4) {motorSube (8);}
      else motorBaja (8); break;
    case 5: motorSube (9); break;  //llaman de última planta
  }

}
```

COMENTARIOS AL PROGRAMA

En el **setup** de programan los pines 2 a 9 como entradas. Los pines 12 y 13 serán salidas.
Habrá que inicializar primeramente el ascensor a la planta baja.
En el bucle se esperará a que haya una llamada devolviendo el número de Pin al que ha de subir o bajar.
Tras comprobar si ha de subir o bajar se harán las llamadas a las funciones motorSube o motorBaja. Las funciones **motorSube** y **motorBaja** recibirán el número de Pin hasta el que han de moverse y harán lo propio con el ascensor, hasta que la lectura de ese Pin se ponga a nivel alto. Lo que significará que el ascensor ha llegado a la planta. Entonces detendrán el motor.

INDICADOR DE TEMPERATURA

CON LEDS



VÍCTOR ANDRÉS ÁLVAREZ SALGADO
PROYECTO CURSO CEP JEREZ
"MICROCONTROLADORA ARDUINO"
OCTUBRE 2011

INTRODUCCIÓN

El proyecto consiste en el montaje de un circuito con un sensor de temperatura (termistor NTC) y la programación en lenguaje *C* de la microcontroladora *Arduino UNO.*

Su funcionamiento es el siguiente: a través de un divisor de tensión, se miden los cambios en el voltaje debido a la variación de resistencia del NTC con la temperatura. Según esos valores, a través del mapeado correspondiente, se irán encendiendo una serie de LEDs, que nos mostrarán visualmente si la temperatura está subiendo o bajando. Hemos calibrado nuestro montaje para que sea sensible a cambios próximos a la temperatura ambiente.

OBJETIVOS

- Conocer y comprender el concepto de microcontrolador.
- Manejar una unidad Arduino.
- Programar los dispositivos.

CONTENIDOS

- Estructura y funcionamiento de los equipos electrónicos.
- Análisis de los componentes de un equipo electrónico: componentes discretos, integrados y elementos auxiliares.
- Diferenciación entre señal analógica y digital.
- Simulación de circuitos por ordenador.
- Sensor de temperatura. NTC y PTC.
- Sensor de luz. LDR.

COMPETENCIAS BÁSICAS

- *Matemática:* Realización de cálculos para la conversión y adaptación de variables.
- *Conocimiento e interacción con el mundo físico*: Manejo de variables físicas del entorno (temperatura).
- *Digital:* Búsqueda de información y lenguaje de programación.

- *Aprender a aprender*: Desarrollo de estrategias de resolución de problemas tecnológicos.
- *Autonomía:* Búsqueda de soluciones propias.

METODOLOGÍA

- Exposición de los componentes electrónicos, características y funcionamiento.
- Introducción al hardware y software de la controladora. Instalación y principales funciones.
- Montaje práctico del circuito en el taller. Conexión y programación de la controladora.
- Evaluación y propuestas de mejora. Elaboración de una memoria.

EVALUACIÓN

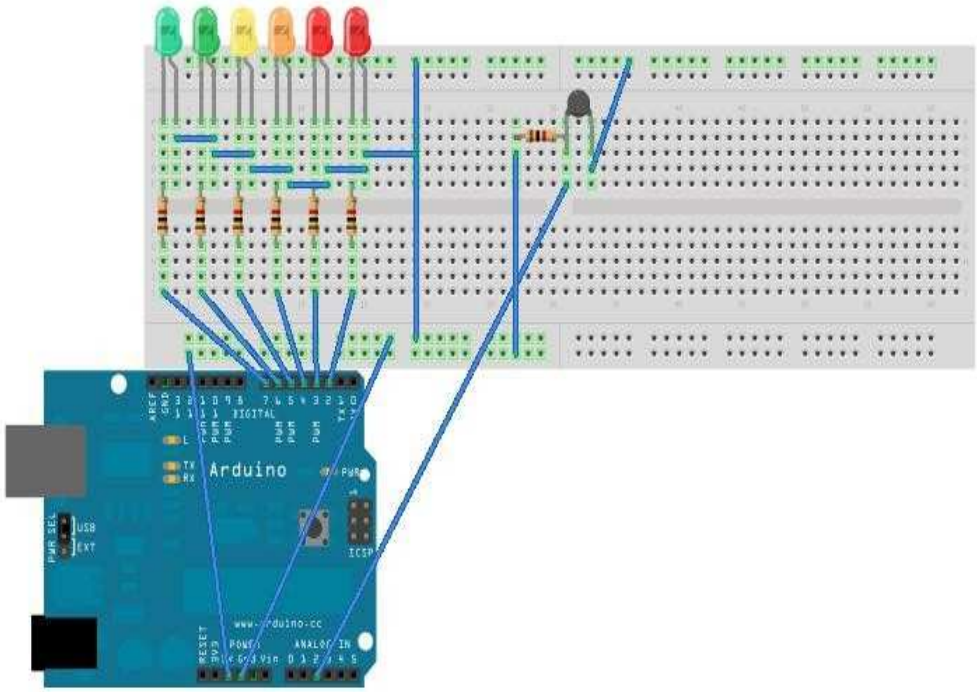
- *Criterios*

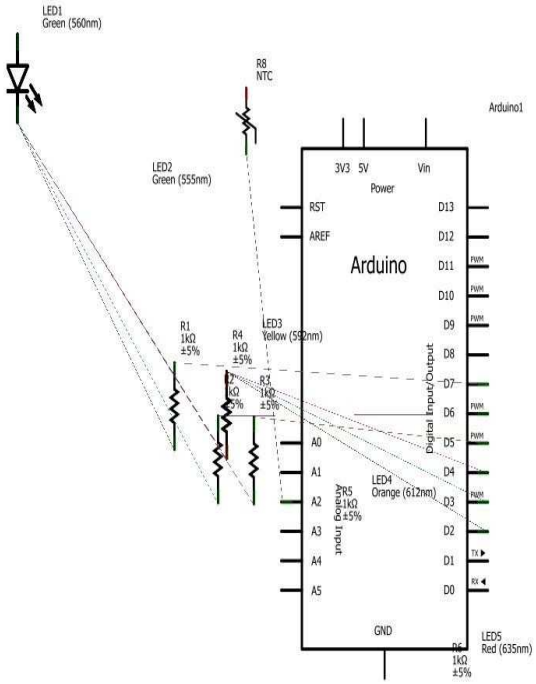
  Explicar el funcionamiento de un circuito, simbología y función.
  Planificar un proyecto tecnológico.
  Buscar información en Internet.
  Interpretar programas sencillos escritos en lenguaje C.

- *Procedimientos*

  Prueba escrita.
  Informe de la práctica.
  Cuaderno de clase.
  Montaje de la placa, la controladora y conexión con el ordenador.

DIAGRAMA DEL CIRCUITO

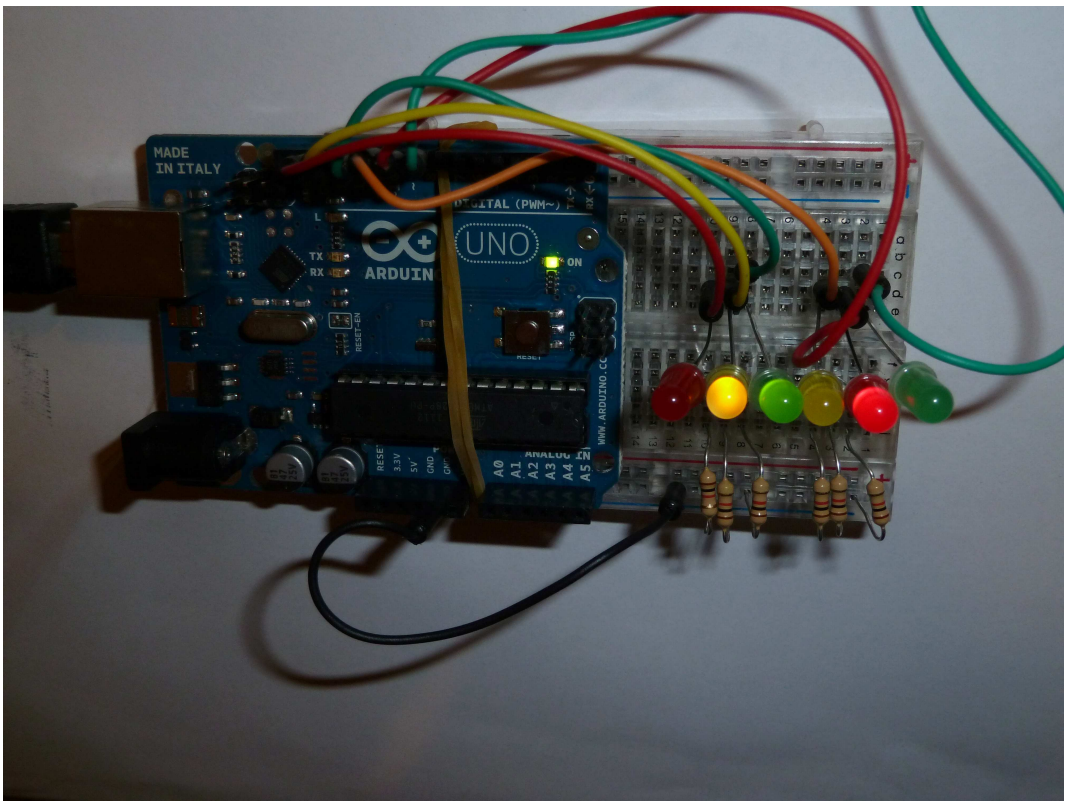




SECUENCIA DE INSTRUCCIONES

```
int ntcPin=2;
int val;
int pot;

void setup()
{
  pinMode (ntcPin,INPUT);
  pinMode (2, OUTPUT);
  pinMode (3,OUTPUT);
  pinMode (4,OUTPUT);
  pinMode (5,OUTPUT);
  pinMode (6,OUTPUT);
  pinMode (7,OUTPUT);

   Serial.begin (9600);
 }

void loop()
{
val=analogRead(ntcPin);
pot=map(val,680,745,2,8);

 Serial.println(val);

switch(pot)
{
 case 2:
  digitalWrite(2,HIGH);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(5,HIGH);
  digitalWrite(6,HIGH);
  digitalWrite(7,HIGH);
  break;

 case 3:
  digitalWrite(2,LOW);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(5,HIGH);
  digitalWrite(6,HIGH);
  digitalWrite(7,HIGH);
  break;

 case 4:
  digitalWrite(2,LOW);
  digitalWrite(3,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(5,HIGH);
  digitalWrite(6,HIGH);
```

```
 digitalWrite(7,HIGH);
 break;

case 5:
 digitalWrite(2,LOW);
 digitalWrite(3,LOW);
 digitalWrite(4,LOW);
 digitalWrite(5,HIGH);
 digitalWrite(6,HIGH);
 digitalWrite(7,HIGH);
 break;

  case 6:
 digitalWrite(2,LOW);
 digitalWrite(3,LOW);
 digitalWrite(4,LOW);
 digitalWrite(5,LOW);
 digitalWrite(6,HIGH);
 digitalWrite(7,HIGH);
 break;

  case 7:
 digitalWrite(2,LOW);
 digitalWrite(3,LOW);
 digitalWrite(4,LOW);
 digitalWrite(5,LOW);
 digitalWrite(6,LOW);
 digitalWrite(7,HIGH);
 break;


  case 8:
 digitalWrite(2,LOW);
 digitalWrite(3,LOW);
 digitalWrite(4,LOW);
 digitalWrite(5,LOW);
 digitalWrite(6,LOW);
 digitalWrite(7,LOW);
 break;
}
}
```



PROTOTIPO
SEMAFORO CONTROLADO POR TECLADO

AUTOR: JOSE IGNACIO CASCAJO JIMÉNEZ
24-octubre-2011

(Dentro del curso MICROCONTROLADORA ARDUINO EN ESO)

1. JUSTIFICACIÓN

Durante la realización del curso 'La microcontroladora Auduino en Secundaria' desarrollado en el IES Andres Benitez de Jerez, organizado por el CEP de Jerez e impartido por D. Francisco J. Perea Reyes (profesor de Informática del IES Ciudad Jardín de Málaga), durante los días 29-sept al 05-oct del año 2011, se tenía como parte final del curso la preparación y presentación de un programa desarrollado por cada alumno asistente al curso, de tema libre y que funcionase. De este modo se trata de cubrir un aspecto muy importante dentro de la formación que es el llevar a la práctica lo aprendido y ponerlo a funcionar.

El curso ha estado dirigido a profesores de secundaria de Tecnología e Informática y han asistido unos 17 alumnos. Hemos utilizado la placa Arduino Uno, además de un set interesante de dispositivos electrónicos (básico, pero muy interesante) con resistencias (fijas y variables), pulsadores, leds, cables conectorizados cortos y largos, armazón para pilas, … y la caja de plástico para guardar todo.

El objeto del curso era dotar al profesorado de conocimientos y habilidades para el trabajo con microcontroladores de cara a su aplicación en el segundo ciclo de educación secundaria obligatoria a través de la asignatura de Tecnología. Para ello se ha trabajado sobre una plataforma de código libre Arduino, ampliamente utilizada en la comunidad educativa y premiada en numerosos certámenes sobre desarrollos electrónicos.

2. INTRODUCCIÓN

Es la primera vez que me he enfrentado en tener delante un microcontrolador y tener que "cacharrear" para hacerlo funcionar.

Inicialmente no parecía difícil ni complicado, todo eran dos o tres comandos o instrucciones para luego, casi por arte de magia, después de disponer la "electrónica", se hacía que un simple led se encendiera o apagara, o que una simple "chicharra" hiciera más o menos ruido, o que un motorcillo de juguete fuera más o menos veloz. La cosa vista así no parecía difícil.

Sin embargo, todo se hacía cada vez algo farragoso: nuevos comandos y nuevos elementos electrónicos. También había que darle al "cerebro" para dar alguna solución mediante instrucciones y Entradas y Salidas digitales o analógicas. En fin, al final parece que lo que has aprendido o entendido no deja de ser una mera gota de agua dentro de un océano de electrónica e instrucciones.

No obstante el párrafo anterior, he llegado a casa y después de días dándole vueltas y tener despistados a toda mi familia, pues "cacharrear" con el Arduino, unos cuantos leds, resistencias y cables sobre la mesa de la cocina no es de muy inteligentes, he llegado a "construir" un semáforo. ¡Vaya! Un semáforo pero algo especial, o al menos para mí.

Se trata de un semáforo hecho de verdad para la ciudad, en el que el peatón tenga preferencia y mejore la señalización ya que actualmente no está muy clara.

Espero poder dar toda la información en el apartado DESCRIPCIÓN.

3. EQUIPAMIENTO UTILIZADO

a) Para el desarrollo del programa se han utilizado

- microcomputador ARDUINO UNO
- entorno de desarrollo de Arduino 0021 que hay en los ordenadores de la Junta, que para instalarlo hay que realizar las siguientes acciones

Aplicaciones
Educación
Tecnologia
Entorno de desarrollo de Arduino
Se abre la aplicación de desarrollo Arduino 0021

- conexión PC-Arduino con cable USB
- para poder funcionar correctamente, tanto con la placa como con la conexión serie vía el cable USB es necesario realizar las siguientes acciones desde la aplicación de desarrollo Arduino 0021

En Tools
- Board
  - Seleccionamos la opción Arduino Uno
- Serial Port
  - Seleccionamos el puerto serie (USB del portátil) donde hemos conectado la placa o microprocesador.

- Para cargar el programa en el que se tenga grabadas las instrucciones se debe hacer del siguiente modo

En File
- Sketchbook
  - Seleccionar el nombre del archivo que tiene cargado el programa y debemos descargar en el micro Arduino

b) Para los componentes electrónicos se han utilizado

- cableados varios (6 cables para los leds y 1 cable para conectar GND-tierra)
- 6 leds (2 rojos, 2 amarillos y 2 verdes) y sus respectivas resistencias de 10k para limitar la corriente y tensión)
- 1 placa breadboard

c) Para el desarrollo de la memoria: cámara de fotos digital, Aplicación Fritzing, para los esquemas y ordenador con procesador de textos y .pdf

Para más información puede contactarse con plataforma Arduino en www.arduino.cc o www.arduino.cc/es y sobre Fritzing en www.fritzing.org

4. DESCRIPCIÓN

Bueno, el nombre que he dado al proyecto es Semáforo controlado por Teclado ya que se trata de un Semáforo, similar al que vemos en las calles de nuestras ciudades con las Luces para los vehículos y las luces para los peatones pero que tiene preparada una conexión por teclado por si se quieren desarrollar mejoras para el futuro.

a) El funcionamiento del programa es básicamente el siguiente.

Se dispone de un semáforo con 3 luces para vehículos (roja, amarilla y verde, con el significado que tienen esas tres luces habitualmente): roja, parar; amarilla, precaución y verde, marcha.

Los cambios de una a otra se han variado algo respecto a lo convencional. Si el semáforo esté en verde (que es la situación de inicio del sistema, verde para vehículos y rojo para peatones), cuando se ha agotado el tiempo o el peatón pulsa la tecla 'H', el verde sigue encendido y simultáneamente se enciende la amarilla durante unos segundos. Pasados esos segundos se apagan el verde y el amarillo y se enciende el rojo.

Transcurrido un tiempo de encendido del rojo para vehículos (pocos segundos), se apaga el rojo de peatones y se enciende el verde de peatones.

El verde de peatones realiza un parpadeo que va aumentando de velocidad a medida que transcurre el tiempo (quiere indicar al peatón que se está agotando el tiempo). Cuando el tiempo finaliza se apaga el verde del peatón y se enciende el rojo del peatón. De este modo están rojo la luz de los vehículos y la de los peatones durante unos 2-3 seg. Pasados este tiempo el rojo de vehículos se apaga y se enciende el verde de vehículos, volviendo a la situación inicial.

Por tanto, en resumen el funcionamiento es similar al de uno de nuestras calles excepto que el amarillo de los vehículos se enciende simultáneamente con el verde de vehículos durante un tiempo prudencial. El tiempo aproximado de funcionamiento en verde para los vehículos es de unos 20 seg y para el verde de los peatones es de unos 16 seg.

b) Funcionamiento del semáforo cuando pulsa la tecla 'H' el peatón

El semáforo funciona siempre como el caso anterior si es que el peatón no ha pulsado la tecla 'H', ya que si es esta el funcionamiento varía.

En caso de pulsar el peatón la tecla 'H', el semáforo minimiza el tiempo de espera del peatón (ya no tiene que esperar todo el ciclo de los 20 segundos de espera). Mediante un programa en el que se realiza una especie de bucle de supervisión de la tecla que llega al Arduino (durante el tiempo que el semáforo tiene el verde para vehículos y rojo para peatones). Si en este tiempo llega la tecla 'H' el semáforo tarda muy poco en iluminar la luz amarilla del peatón (significa que el peatón ha pulsado la tecla necesaria para pasar) y sigue la luz roja del peatón y la verde de los vehículos.

En pocos segundos desde que el peatón ha pulsado comienza el proceso entero de cambio de luces del semáforo (tal y como se explicó al principio en el funcionamiento básico, sólo que ahora parte de que la situación es que para el peatón están encendidas la roja y la amarilla y para el vehículo está encendida la roja). De este modo el peatón tarda poco tiempo en esperar para cruzar y la señalización, respecto a lo actual, mejora sustancialmente.

c) Servicio del teclado

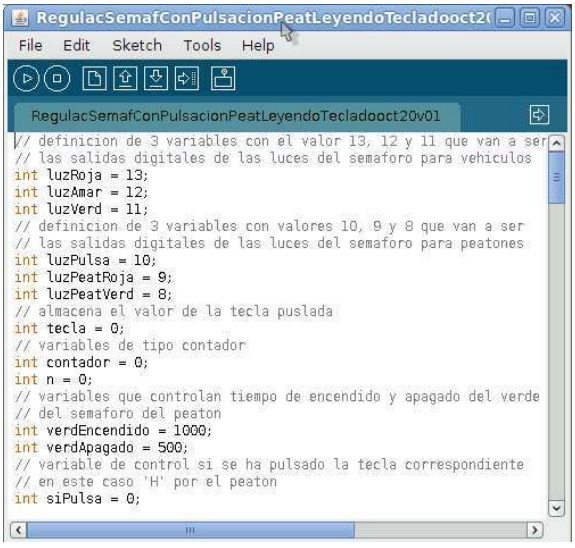
La introducción del teclado en el proyecto del semáforo no es otra que disponer del mismo para realizar otras aplicaciones y/o añadir funcionalidades. En este caso sólo recoge las teclas que se han pulsado y sólo hay actuación cuando se pulsa la 'H'.

Como mejoras podrían proponerse otras actuaciones que realizara este semáforo o grupos de semáforos según cual fuese la tecla que se pulsara o, también, realizar un control remoto del mismo.

Se dejan aquí estos comentarios para si alguien, próximamente, quisiera seguir trabajando este proyecto e incorporarle mejoras o novedades.

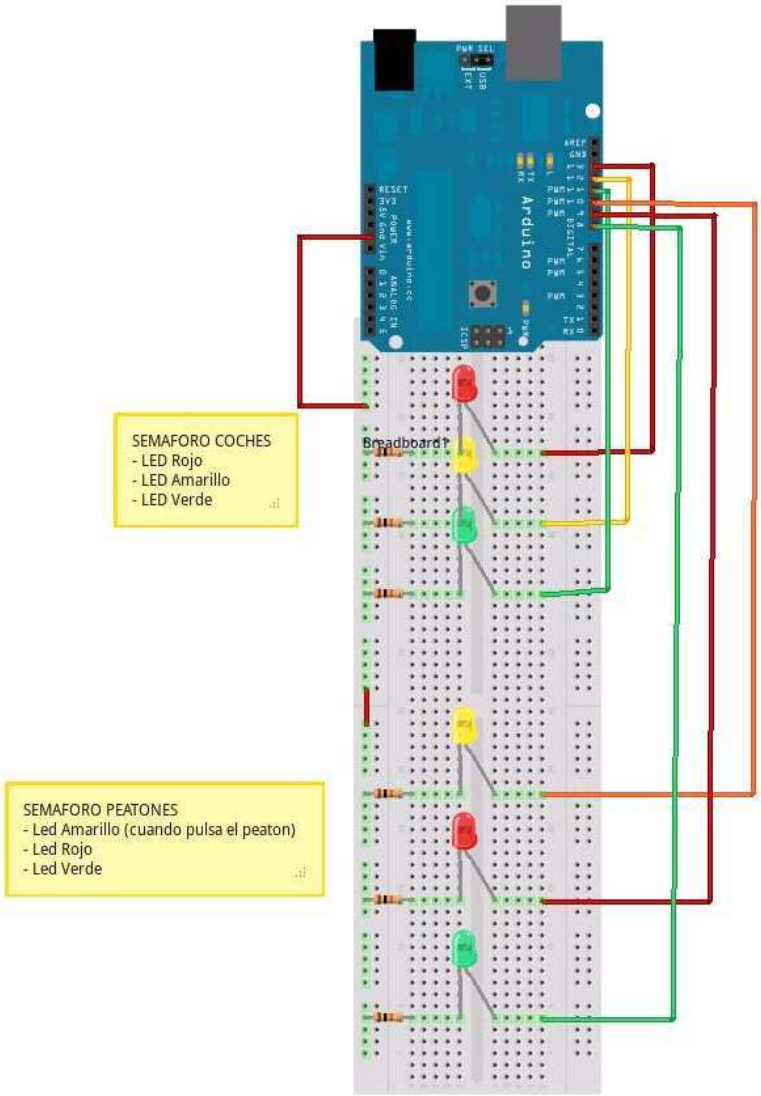
5. PROGRAMA REAL

Imagen del programa y comandos sobre el ordenador de la Junta y con el programa Arduino 0021.



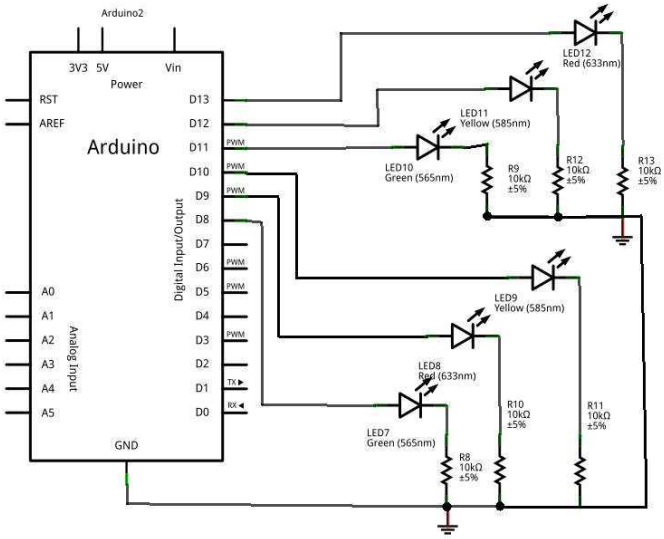
6. PLACA DE PROTOTIPOS

Placa de prototipos del semáforo con control por teclado. Se han separado los leds en vertical para que se vean mejor las pistas y los conexionados, además se han pintado los cables para diferenciar unos de otros y saber cuales son las entradas/salidas utilizadas.



7. ESQUEMA DEL PROTOTIPO ELÉCTRICO

Esquema del Prototipo eléctrico del Semáforo (realizado con la aplicación Fritzing)



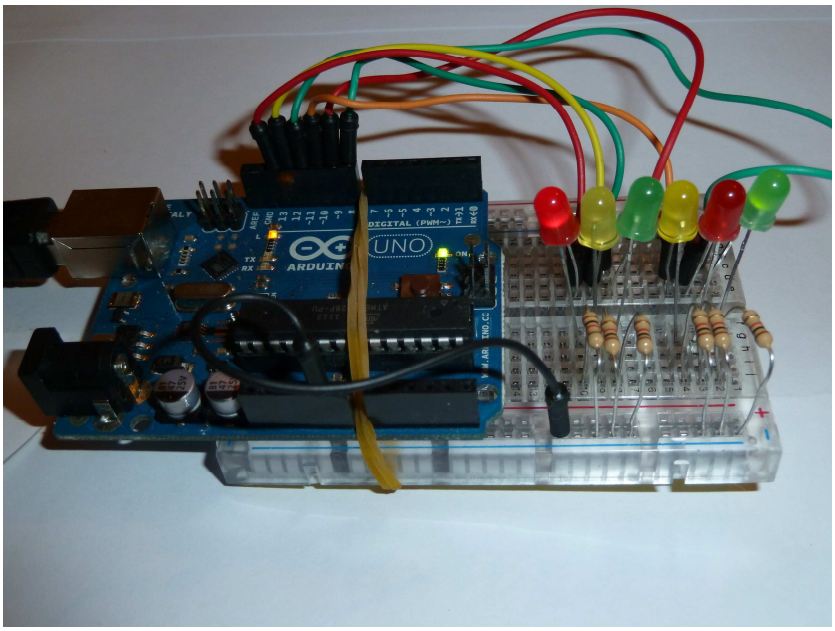
8. INTRUCCIONES Y LÍNEAS DE COMANDO

Se incluye un listado completo de las instrucciones y las explicaciones de las mismas (están realizadas sobre las mismas líneas de comando).

```
// definicion de 3 variables con el valor 13, 12 y 11 que van a ser
// las salidas digitales de las luces del semaforo para vehiculos
int luzRoja = 13;
int luzAmar = 12;
int luzVerd = 11;
// definicion de 3 variables con valores 10, 9 y 8 que van a ser
// las salidas digitales de las luces del semaforo para peatones
int luzPulsa = 10;
int luzPeatRoja = 9;
int luzPeatVerd = 8;
// almacena el valor de la tecla puslada
int tecla = 0;
// variables de tipo contador
int contador = 0;
int n = 0;
// variables que controlan tiempo de encendido y apagado del verde
// del semaforo del peaton
int verdEncendido = 1000;
int verdApagado = 500;
// variable de control si se ha pulsado la tecla correspondiente
// en este caso 'H' por el peaton
int siPulsa = 0;


void setup() {
  // declara como salidas las luces roja, amarilla y verde del semaforo de coches
  // y las amarilla (luzPulsa), verde y roja del semaforo de peatones
  pinMode(luzRoja, OUTPUT);
  pinMode(luzAmar, OUTPUT);
  pinMode(luzVerd, OUTPUT);
  pinMode(luzPulsa, OUTPUT);
  pinMode(luzPeatRoja, OUTPUT);
  pinMode(luzPeatVerd, OUTPUT);
  // declaracion del interface de comunicacion con el teclado
  Serial.begin(9600);
}

void loop() {
  // inicializa siPulsa a 0, pues no se ha pulsado la tecla 'H' por el peaton
  // tambien inicializa tecla a 0
  siPulsa=0;
  tecla=0;
  // inicia todas las luces del semaforo Verde para coches y Rojo para peatones
  digitalWrite(luzRoja,LOW);
  digitalWrite(luzAmar,LOW);
  digitalWrite(luzVerd,HIGH);
  digitalWrite(luzPulsa,LOW);
  digitalWrite(luzPeatRoja,HIGH);
  digitalWrite(luzPeatVerd,LOW);
  delay(1000);

  // durante unos 20 segundo realiza un seguimiento de ver si se pulta el teclado
  // por parte del peaton, para eso esta este bucle for
  for(n=0;n<=200;n++){
    if(siPulsa==0){
      delay(100);
```

```
      pulsaTecla();
    }
  }
  // si en el bucle fior anterior No se ha pulsado la tecla 'H' entones ha pasado el tiempo
  // de verde para los coches y ya, por tiempo, hay que cambiar las luces del semaforo de verde a rojo
  if (siPulsa==0){
    cambiaLuces();
  }
}
  //funcion que detecta si se ha pulsado una tecla y que tecla es
  //si la tecla es 'H' entones cambia las luces para que pase el peaton
  //en caso contrario sigue ejecutandose dentro de un bucle hasta
  //acabar los aproximadamente 20 segundos

  void pulsaTecla(){
    if (Serial.available()){
      tecla=Serial.read();
      if (tecla == 'H'){
        digitalWrite(luzPulsa,HIGH);
        delay (3000);
        digitalWrite(luzAmar, HIGH);
        delay (3000);
        digitalWrite(luzRoja, HIGH);
        digitalWrite(luzAmar, LOW);
        digitalWrite(luzVerd, LOW);
        digitalWrite(luzPulsa, LOW);
        delay (2000);
        digitalWrite(luzPeatRoja, LOW),
        digitalWrite(luzPeatVerd, HIGH);
        //llama a la funcion que hace el cambio de luces para que
        //pase el peaton
        pasaPeaton();
        //pone a 1 siPulsa para saber que ha pulsado la tecla 'H'
        //antes de salir del la instruccion if
        //de este modo no se ejecutara otra vez el bucle
        siPulsa=1;
      }
    }
  }

  //cambia las luces del semaforo en caso de que no se pulse
  //la tecla 'H' por parte del peaton

  void cambiaLuces(){
    //cambia el semaforo de coches
    digitalWrite(luzAmar, HIGH);
    delay(3000);
    digitalWrite(luzRoja, HIGH);
    digitalWrite(luzAmar, LOW);
    digitalWrite(luzVerd, LOW);
    delay(2000);
    // cambia el semaforo del peaton
    digitalWrite(luzPulsa, LOW);
    digitalWrite(luzPeatRoja, LOW),
    digitalWrite(luzPeatVerd, HIGH);
    //realiza el encendido parpadeante del paso del peaton
    pasaPeaton();

  }

  //realiza el encendido parpadeante del paso del peaton
```

```
void pasaPeaton(){
// inicializacion tiempos de encendido y apagado luz cruce peaton
  verdEncendido=1000;
  verdApagado=500;
  // bucle para realizar encendido y apagado de luz verde peaton
  for (contador =0; contador<=25; contador++){
    delay(verdEncendido);
    //disminucion del tiempo de encendido
    verdEncendido=verdEncendido*0.9;
    digitalWrite(luzPeatVerd, LOW);
    delay(verdApagado);
    //disminucion del tiempo de apagado
    verdApagado=verdApagado*0.9;
    digitalWrite(luzPeatVerd, HIGH);
  }
  // termina de cambiar las luces del semaforo
  digitalWrite(luzPeatVerd, LOW);
  digitalWrite(luzPeatRoja, HIGH);
  delay(3000);
  digitalWrite(luzVerd, HIGH);
  digitalWrite(luzRoja, LOW);
}
```



Teclado Electrónico

1. **Introducción**

La unidad didáctica pretende dar a conocer el funcionamiento de la tarjeta controladora Arduino, empleando como hilo conductor el montaje de un teclado electrónico de una escala.

Hemos elegido este montaje tan repetitivo para permitir a nuestros alumnos/as observar cómo se realizan los primeros pasos y que, a continuación, sigan solos hasta completar el circuito y el código de manera intuitiva.

2. **Objetivos**

Conocer los principios fundamentales de la programación y la automatización.

Realizar un montaje real de un objeto de uso cotidiano.

Comprender los principios que subyacen en el funcionamiento de los componentes electrónicos empleados: resistencia, pulsador y zumbador.

Valorar la importancia del control y la robótica para resolver problemas de nuestro entorno.

3. **Temporalización**

1ª Sesión: El zumbador

Se explica el funcionamiento del zumbador.

Se calculan las frecuencias de cada nota musical de la escala 3. Básicamente, generamos un tren de pulsos, pero con la frecuencia de cada una de las notas. El tono está calculado en función de la inversa de la frecuencia de la nota.

2ª Sesión: El piano de una tecla

Se entrega el montaje en protoboard y el código que permite hacer funcionar un teclado de una sola tecla.

Análisis del montaje y del código.

3ª y 4ª Sesiones: El teclado

Nuestros alumnos/as completan el código y los componentes sobre la protoboard para construir un teclado completo de una escala 3.

4. **Código**

```
int speakerOut =13; int notado = 1915;  int notare = 1700;  int notami = 1519;  int notafa = 1432; int notasol = 1275; int notala = 1136; int notasi = 1014;     int notados = 1804; int notares = 1607; int notafas = 1351; int notasols = 1204; int notalas = 1073; int d;  int re;     int mi; int fa; int sol; int la; int si; int dos;  int res;  int fas;  int ols;  int las;

oid setup(){

pinMode (speakerOut, OUTPUT);   pinMode (1, INPUT);  pinMode (2, INPUT);  pinMode (3, INPUT);  pinMode (4, INPUT);  pinMode (5, INPUT);           pinMode (6, INPUT);  pinMode (7, INPUT);  pinMode (8, INPUT);  pinMode (9, INPUT);  pinMode (10, INPUT);  pinMode (11, INPUT);  pinMode (12, INPUT);

}

void loop() {

d=digitalRead(12); re=digitalRead(11); mi=digitalRead(10); fa=digitalRead(9); sol=digitalRead(8); la=digitalRead(7); si=digitalRead(6); dos=digitalRead(5);  res=digitalRead(4);   fas=digitalRead(3);   sols=digitalRead(2);   las=digitalRead(1);

  if (d==HIGH){

    digitalWrite(speakerOut,HIGH);

    delayMicroseconds(notado);

    digitalWrite(speakerOut, LOW);
```

```
  delayMicroseconds(notado);
}
if (re==HIGH){
  digitalWrite(speakerOut,HIGH);
  delayMicroseconds(notare);
  digitalWrite(speakerOut, LOW);
  delayMicroseconds(notare);
}
if (mi==HIGH){
  digitalWrite(speakerOut,HIGH);
  delayMicroseconds(notami);
  digitalWrite(speakerOut, LOW);
  delayMicroseconds(notami);
}
if (fa==HIGH){
  digitalWrite(speakerOut,HIGH);
  delayMicroseconds(notafa);
  digitalWrite(speakerOut, LOW);
  delayMicroseconds(notafa);
```

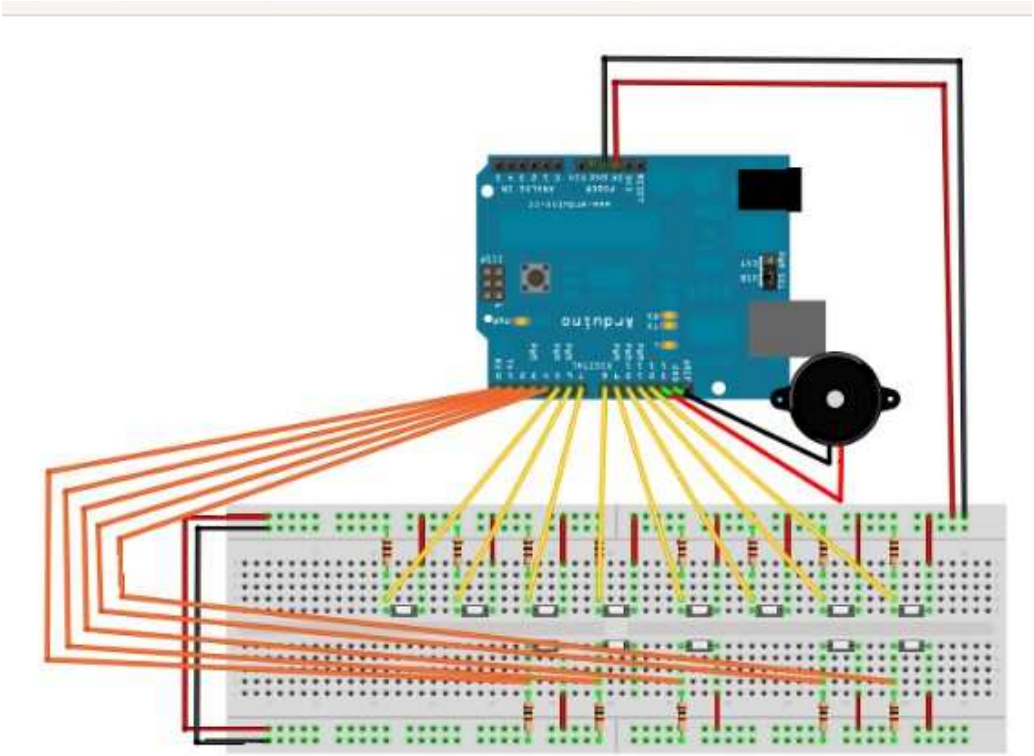
```
}
if (sol==HIGH){
  digitalWrite(speakerOut,HIGH);
  delayMicroseconds(notasol);
  digitalWrite(speakerOut, LOW);
  delayMicroseconds(notasol);
}
if (la==HIGH){
  digitalWrite(speakerOut,HIGH);
  delayMicroseconds(notala);
  digitalWrite(speakerOut, LOW);
  delayMicroseconds(notala);
}
if (si==HIGH){
  digitalWrite(speakerOut,HIGH);
  delayMicroseconds(notasi);
  digitalWrite(speakerOut, LOW);
  delayMicroseconds(notasi);
}
```

```
if (dos==HIGH){
  digitalWrite(speakerOut,HIGH);
  delayMicroseconds(notados);
  digitalWrite(speakerOut, LOW);
  delayMicroseconds(notados);
}
if (res==HIGH){
  digitalWrite(speakerOut,HIGH);
  delayMicroseconds(notares);
  digitalWrite(speakerOut, LOW);
  delayMicroseconds(notares);
}
if (fas==HIGH){
  digitalWrite(speakerOut,HIGH);
  delayMicroseconds(notafas);
  digitalWrite(speakerOut, LOW);
  delayMicroseconds(notafas);
}
if (sols==HIGH){
```

```
  digitalWrite(speakerOut,HIGH);
  delayMicroseconds(notasols);
  digitalWrite(speakerOut, LOW);
  delayMicroseconds(notasols);
}
if (las==HIGH){
  digitalWrite(speakerOut,HIGH);
  delayMicroseconds(notalas);
  digitalWrite(speakerOut, LOW);
  delayMicroseconds(notalas);
} }
```

5. **Montaje**

Se necesitan los siguientes componentes: 1 zumbador, 12 resistencias de 1kΩ y 12 `pulsadores.



UNIDAD DIDÁCTICA

GENERACIÓN ACÚSTICO LUMINOSA DEL CÓDIGO MORSE USANDO LA MICROCONTROLADORA ARDUINO

Moisés García Martínez

ÍNDICE:

- INTRODUCCIÓN
- CARACTERÍSTICAS DE LA MICROCONTROLADORA ARDUINO UNO
- PROGRAMA
- FOTOS DEL MONTAJE
- COMPETENCIAS BÁSICAS TRABAJADAS EN ESTA UNIDAD

INTRODUCCIÓN:

La presente unidad didáctica está realizada como actividad de fase no presencial del curso "La microcontroladora Arduino en Secundaria", y consiste en la generación de una señal tanto acústica como luminosa del código Morse, para lo cual aparte de la controladora, usaremos un pequeño zumbador y un LED con su resistencia limitadora.

El código Morse es un código o sistema de comunicación que permite la comunicación telegráfica a través de la transmisión de impulsos eléctricos de longitudes diversas o por medios visuales, como luz, sonoros o mecánicos. Este código consta de una serie de puntos, rayas y espacios, que al ser combinados entre si pueden formar palabras, números y otros símbolos.

Este sistema de comunicación fue creado en el año 1830 por Samuel F.B. Morse, un inventor, pintor y físico proveniente de los Estados Unidos, quien pretendía encontrar un medio de comunicación telegráfica. La creación de éste código tiene su origen en la creación del señor Morse de un telégrafo, invento que le trajo bastante dificultades, ya que, en un principio, el registro de este fabuloso invento le fue negado tanto en Europa como en los Estados Unidos. Finalmente, logró conseguir el financiamiento del gobierno americano, el que le permitió construir una línea telegráfica entre Baltimore y Washington. Un año después se realizaron las primeras transmisiones, resultando éstas bastante exitosas, lo que dio pie a la formación de una enorme compañía que cubriría a todos los Estados Unidos de líneas telegráficas.

Como se dijo anteriormente, las letras, números y demás signos, se representan en el código Morse a través de puntos y líneas que se transmiten como impulsos eléctricos que producen una señal de luz o de sonido de una duración determinada:

CÓDIGO MORSE (Alfanumérico)

Letra	Código	Letra/Número	Código
A	. −	S	. . .
B	− . . .	T	−
C	− . − .	U	. . −
D	− . .	V	. . . −
E	.	W	. − −
F	. . − .	X	− . . −
G	− − .	Y	− . − −
H	. . . .	Z	− − . .
I	. .		
J	. − − −	1	. − − − −
K	− . −	2	. . − − −
L	. − . .	3	. . . − −
M	− −	4	. . . . −
N	− .	5	. . . . .
Ñ	− − . − −	6	− . . . .
O	− − −	7	− − . . .
P	. − − .	8	− − − . .
Q	− − . −	9	− − − − .
R	. − .	0	− − − − −

El punto es considerado, en el código Morse, como una unidad, que en tiempo equivale, aproximadamente a 1/25 de segundo. La línea, en tiempo, equivale a tres puntos. Cada letra se separa por un silencio de tres puntos de duración, mientras que cada palabra se separa por cinco puntos.

CARACTERÍSTICAS DE LA MICROCONTROLADORA ARDUINO UNO

Arduino puede ser utilizado para desarrollar objetos autónomos e interactivos, como prototipos o interactuar con software instalado en el ordenador. Dada su rápida curva de aprendizaje y su precio económico es ideal para educadores, diseñadores y cualquiera interesado en la electrónica y robótica.

El compilador necesario para programarlo está disponible de forma gratuita en www.arduino.cc y está disponible para Mac OS X, Windows y Linux.

Arduino UNO es la versión mejorada de su predecesor Duemilanove. Incluye función de autoreset, protección de sobrecargas, conector USB para programarlo, totalmente montado con componentes miniatura SMD (salvo el microcontrolador, para poder cambiarlo facilmente) y nuevo bootloader OptiBoot a 155kbps.

La placa se entrega completamente ensamblada y probada con un microcontrolador AVR ATmega328 con un cristal de cuazo de 32 Mhz.

Se entrega con el nueo chip Atmega328 de AVR con 32 KB de memoria de programa en lugar de 16 KB de la anterior versión, RAM de 2KB (antes 1KB) y EEPROM de 1 KB (antes 512 bytes).

La carga de los programas también es más rápida ya que el bootloader fué actualizado a una velocidad de 115000 baudios.

La Arduino Uno posee:

• 14 entrada/salida digitales, de los cuales 6 pueden ser usados como salidas PWM

• Posee 6 entradas analógicas

• Los pin 0 y 1 pueden funcionar como RX y TX serial.

• Un oscilador de cristal de 32 MHz

• Conector USB

• Una entrada de alimentación

• Un conector ICSP

• Botón de Reset

La Arduino UNO posee todo lo que se necesita para manejar el controlador, simplemente se conecta a un ordenador por medio del cable USB o se puede alimentar utilizando una batería o un adaptador AC-DC. Si se conecta por USB, la alimentación externa no es necesaria.

Para programar sólo necesita el IDE de Arduino, que se encuentra en la sección de descargas en el sitio web de www.arduino.cc

Algunas características son:

- Microcontroladora: ATmega328P-PU
- Voltaje operativo: 5 V
- Voltaje de entrada (recomendado): 7 – 12 V
- Límite voltaje de entrada: 20 V
- Digital I/O Pins: 14 (6 de los cuales proporcionan salidas analógicas)
- Pins de entradas analógicas: 6
- Intensidad de corriente continua por I/O Pin: 40 mA
- Intensidad de corriente continua por 3.3V Pin: 50 mA
- Memoria: 32 KB
- SRAM: 2 KB
- EEPROM: 1 KB
- Frecuencia del oscilador: 32 MHz

Las Entradas analógicas son de 10 bits, por lo que entregan valores entre 0 y 1023. El rango de voltaje está dado entre 0 y 5 volts, pero utilizando el pin AREF disponible, este rango se puede variar a algún otro deseado.

PROGRAMA

El programa que va a generar las señales acústico-luminosas es muy sencillo. Para reproducir cada código se ha usado la instrucción “swicht” que nos permite distinguir entre cada letra o número. Además se ha tenido en cuenta que nos pueda permitir usar tanto las mayúsculas como las minúsculas, para ello se ha añadido la opción “case” anidada que hace que se realice la misma función en los dos casos.

El programa, cada vez que se introduce una letra o número en el monitor del programa de Arduino, ejecuta las subrutinas “punto” y “raya” según el código y las muestra en la pantalla del monitor. El programa sólamente ejecuta una letra o número cada vez.

Para ello se ha usado la instrucción `if(Serial.available())` que nos introduce en el bucle siempre que recibamos un dato por la entrada de datos serie. Una vez recibido este dato se asigna a la variable `"val"` con la instrucción `"Serial.read()"` y se va realizando la comparación con todos los códigos mediante la instrucción `"switch"`.

Para simplificar el programa se ha usado el concepto de subrutina. Esta opción nos permite realizar un trozo de programa repetidas veces, sin tener que escribirlo cada vez. La forma de uso de una subrutina es la siguiente: antes de comenzar el programa principal (loop), se realiza una declaración de cada subrutina, en este caso "punto" y "raya". Para estas subrutinas no hay que pasarle ninguna información externa, por eso va seguida de unos paréntesis vacíos ( ). Cada vez que se llama a una subrutina desde el programa principal, el proceso sale de éste y continúa su ejecución en la subrutina llamada. Una vez concluida } vuelve al programa principal en el punto donde lo abandonó.

```
/* unidad didáctica morse */

int pito=9;
int val=0;
int draya=300;
int dpunto=150;
int espera=200;


/* SUBRUTINA PUNTO */
void punto(){
  digitalWrite(pito,HIGH);
  delay(250);
  digitalWrite(pito,LOW);
  Serial.println("Punto");
  delay (espera);
}

/* SUBRUTINA RAYA */
void raya(){
  digitalWrite(pito,HIGH);
  delay(500);
  digitalWrite(pito,LOW);
 Serial.println("Raya");
 delay (espera);
}


void setup (){
  pinMode (pito, OUTPUT);
  Serial.begin (19200);
 }

void loop(){
  if(Serial.available()){
    val=Serial.read();
    switch(val){
```

```
case 'A':
case 'a':
  punto();
  raya();
  break;

case 'B':
case 'b':
  raya();
  punto();
  punto();
  punto();
  break;

case 'C':
case 'c':
  raya();
  punto();
  raya();
  punto();
  break;

case 'D':
case 'd':
  raya();
  punto();
  punto();
  break;

case 'E':
case 'e':
  punto();
  break;


case 'F':
case 'f':
  punto();
  punto();
  raya();
  punto();
  break;

case 'G':
case 'g':
  raya();
  raya();
  punto();
  break;

case 'H':
case 'h':
  punto();
  punto();
  punto();
  punto();
  break;
```

```
case 'I':
case 'i':
  punto();
  punto();
  break;

case 'J':
case 'j':
  punto();
  raya();
  raya();
  raya();
  break;

case 'K':
case 'k':
  raya();
  punto();
  raya();
  break;

case 'L':
case 'l':
  punto();
  raya();
  punto();
  punto();
  break;

case 'M':
case 'm':
  raya();
  raya();
  break;

case 'N':
case 'n':
  raya();
  punto();
  break;

case 'Ñ:
case 'ñ:
  raya();
  raya();
  punto();
  raya();
  raya();
  break;

case 'O':
case 'o':
  raya();
  raya();
  raya();
  break;
```

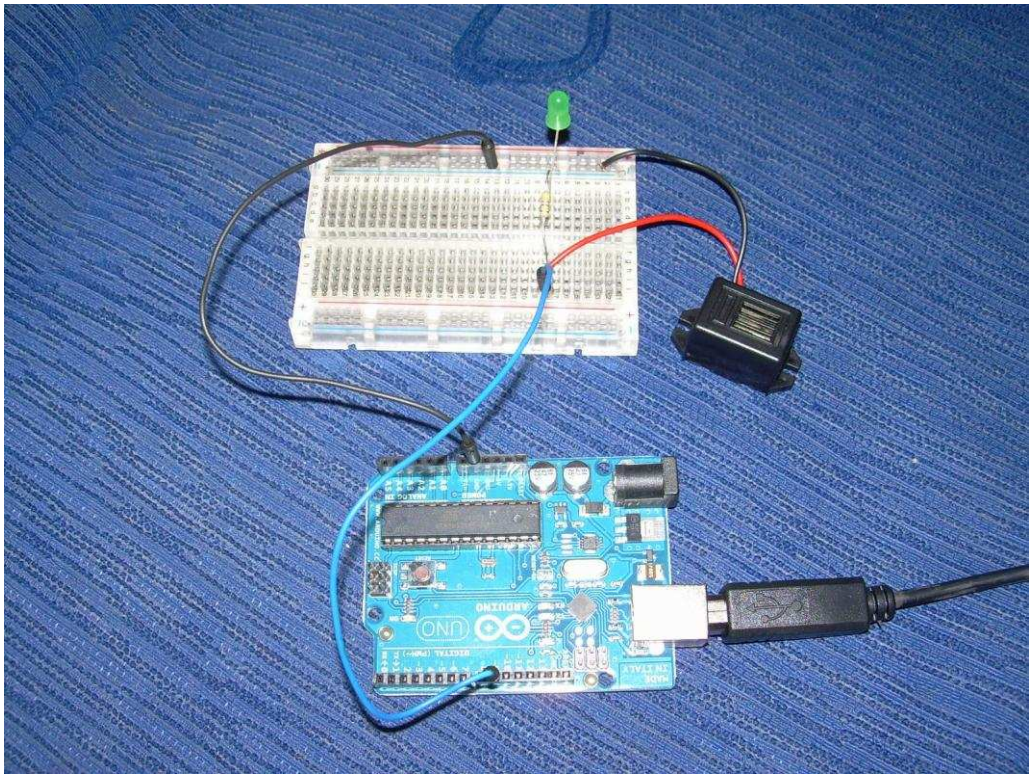
```
case 'P':
case 'p':
  punto();
  raya();
  raya();
  punto();
  break;

case 'Q':
case 'q':
  raya();
  raya();
  punto();
  raya();
  break;

case 'R':
case 'r':
  punto();
  raya();
  punto();
  break;

case 'S':
case 's':
  punto();
  punto();
  punto();
  break;

case 'T':
case 't':
  raya();
  break;

case 'U':
case 'u':
  punto();
  punto();
  raya();
  break;

case 'V':
case 'v':
  punto();
  punto();
  punto();
  raya();
  break;

case 'W':
case 'w':
  punto();
  raya();
  raya();
  break;
```

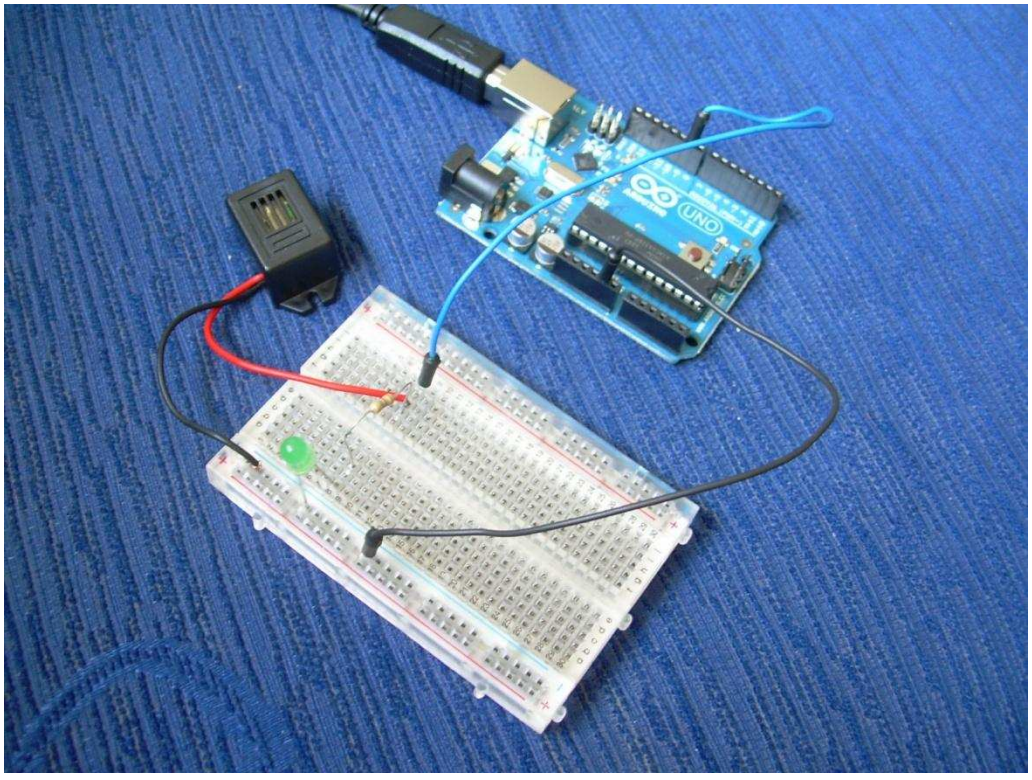
```
case 'X':
case 'x':
  raya();
  punto();
  punto();
  raya();
  break;

case 'Y':
case 'y':
  raya();
  punto();
  raya();
  raya();
  break;

case 'Z':
case 'z':
  raya();
  raya();
  punto();
  punto();
  break;

case '1':
  punto();
  raya();
  raya(k);
  raya();
  raya();
  break;

case '2':
  punto();
  punto();
  raya();
  raya();
  raya();
  break;

case '3':
  punto();
  punto();
  punto();
  raya();
  raya();
  break;

case '4':
  punto();
  punto();
  punto();
  punto();
  raya();
  break;
```

```
      case '5':
        punto();
        punto();
        punto();
        punto();
        punto();
        break;

      case '6':
        raya();
        punto();
        punto();
        punto();
        punto();
        break;

      case '7':
        raya();
        raya();
        punto();
        punto();
        punto();
        break;

      case '8':
        raya();
        raya();
        raya();
        punto();
        punto();
        break;

      case '9':
        raya();
        raya();
        raya();
        raya();
        punto();
        break;

      case '0':
        raya();
        raya();
        raya();
        raya();
        raya();
        break;


    }
  }
}
```

FOTOS DEL MONTAJE

De la salida digital número 9 se lleva un cable para conectar el diodo LED con su resistencia limitadora y en paralelo va el zumbador. Asimismo, se lleva de la placa Arduino la toma del negativo de la patilla GND correspondiente





COMPETENCIAS BÁSICAS TRABAJADAS EN ESTA UNIDAD

1.- Competencia en comunicación lingüística.

El código Morse es en si un lenguaje y requiere un tratamiento similar a la hora de aprenderlo como otro lenguaje, de hecho solo consiste en relacionar cada letra o número con su correspondiente código expresado en forma de rayas y o puntos. Se hace obvio que para su aprendizaje es imprescindible la práctica del mismo para mandar y recibir los mensajes con rapidez en este código así como la correcta interpretación de los mismos.

4.- Competencia en el tratamiento de la información y competencia digital.

El uso de las tecnologías de la información y comunicación como un elemento esencial para informarse y comunicarse.

Por otra parte, hemos de trabajar con componentes electrónicos, bien sea analógicos y o digitales. En esta unidad en concreto usamos componentes analógicos (resistencia, LED, zumbador) y en la parte digital, la propia microcontroladora Arduino.

6.- Competencia cultural y artística.

Aunque este tipo de comunicación se hace aún hoy día pero de forma puntual y en ámbitos muy concretos, desde comienzos del siglo XX, tuvo un uso muy generalizado en comunicaciones terrestres, marítimas y militares, por lo que hay que valorarlo como una manifestación del mundo social de una época ya pasada.

7.- Competencia para aprender a aprender.-

Este código necesita de la memorización de su tipología (raya y punto), por lo que se hace necesario usar unas técnicas de organización, memorización e incluso inventar y desarrollar reglas nemotécnicas para que cada alumno de forma individual pueda recordar el código asociado a cada letra o número. Se hace evidente que para su uso de forma fluida es muy necesaria la práctica de forma continuada.

8.- Competencia de autonomía e iniciativa personal.

Hay que desarrollar la habilidad para relacionarse y trabajar en equipo, ya que si se pretende utilizar este código como medio de comunicación, ha de haber al menos una persona que actúe como emisor del mensaje y otra como receptora e interpretadora del mismo y viceversa.

UNIDAD DIDÁCTICA

GENERACIÓN ACÚSTICO LUMINOSA DEL CÓDIGO MORSE USANDO LA MICROCONTROLADORA ARDUINO

Manuel García Carrera

ÍNDICE:

- INTRODUCCIÓN
- CARACTERÍSTICAS DE LA MICROCONTROLADORA ARDUINO UNO
- PROGRAMA
- FOTOS DEL MONTAJE
- COMPETENCIAS BÁSICAS TRABAJADAS EN ESTA UNIDAD

INTRODUCCIÓN:

La presente unidad didáctica está realizada como actividad de fase no presencial del curso "La microcontroladora Arduino en Secundaria", y consiste en la generación de una señal tanto acústica como luminosa del código Morse, para lo cual aparte de la controladora, usaremos un pequeño zumbador y un LED con su resistencia limitadora.

El código Morse es un código o sistema de comunicación que permite la comunicación telegráfica a través de la transmisión de impulsos eléctricos de longitudes diversas o por medios visuales, como luz, sonoros o mecánicos. Este código consta de una serie de puntos, rayas y espacios, que al ser combinados entre si pueden formar palabras, números y otros símbolos.

Este sistema de comunicación fue creado en el año 1830 por Samuel F.B. Morse, un inventor, pintor y físico proveniente de los Estados Unidos, quien pretendía encontrar un medio de comunicación telegráfica. La creación de éste código tiene su origen en la creación del señor Morse de un telégrafo, invento que le trajo bastante dificultades, ya que, en un principio, el registro de este fabuloso invento le fue negado tanto en Europa como en los Estados Unidos. Finalmente, logró conseguir el financiamiento del gobierno americano, el que le permitió construir una línea telegráfica entre Baltimore y Washington. Un año después se realizaron las primeras transmisiones, resultando éstas bastante exitosas, lo que dio pie a la formación de una enorme compañía que cubriría a todos los Estados Unidos de líneas telegráficas.

Como se dijo anteriormente, las letras, números y demás signos, se representan en el código Morse a través de puntos y líneas que se transmiten como impulsos eléctricos que producen una señal de luz o de sonido de una duración determinada:

CÓDIGO MORSE (Alfanumérico)

Letra	Código	Letra/Número	Código
A	. −	S	. . .
B	− . . .	T	−
C	− . − .	U	. . −
D	− . .	V	. . . −
E	.	W	. − −
F	. . − .	X	− . . −
G	− − .	Y	− . − −
H	. . . .	Z	− − . .
I	. .		
J	. − − −	1	. − − − −
K	− . −	2	. . − − −
L	. − . .	3	. . . − −
M	− −	4	. . . . −
N	− .	5	. . . . .
Ñ	− − . − −	6	− . . . .
O	− − −	7	− − . . .
P	. − − .	8	− − − . .
Q	− − . −	9	− − − − .
R	. − .	0	− − − − −

El punto es considerado, en el código Morse, como una unidad, que en tiempo equivale, aproximadamente a 1/25 de segundo. La línea, en tiempo, equivale a tres puntos. Cada letra se separa por un silencio de tres puntos de duración, mientras que cada palabra se separa por cinco puntos.

CARACTERÍSTICAS DE LA MICROCONTROLADORA ARDUINO UNO

Arduino puede ser utilizado para desarrollar objetos autónomos e interactivos, como prototipos o interactuar con software instalado en el ordenador. Dada su rápida curva de aprendizaje y su precio económico es ideal para educadores, diseñadores y cualquiera interesado en la electrónica y robótica.

El compilador necesario para programarlo está disponible de forma gratuita en www.arduino.cc y está disponible para Mac OS X, Windows y Linux.

Arduino UNO es la versión mejorada de su predecesor Duemilanove. Incluye función de autoreset, protección de sobrecargas, conector USB para programarlo, totalmente montado con componentes miniatura SMD (salvo el microcontrolador, para poder cambiarlo facilmente) y nuevo bootloader OptiBoot a 155kbps.

La placa se entrega completamente ensamblada y probada con un microcontrolador AVR ATmega328 con un cristal de cuazo de 32 Mhz.

Se entrega con el nueo chip Atmega328 de AVR con 32 KB de memoria de programa en lugar de 16 KB de la anterior versión, RAM de 2KB (antes 1KB) y EEPROM de 1 KB (antes 512 bytes).

La carga de los programas también es más rápida ya que el bootloader fué actualizado a una velocidad de 115000 baudios.

La Arduino Uno posee:

• 14 entrada/salida digitales, de los cuales 6 pueden ser usados como salidas PWM

• Posee 6 entradas analógicas

• Los pin 0 y 1 pueden funcionar como RX y TX serial.

• Un oscilador de cristal de 32 MHz

• Conector USB

• Una entrada de alimentación

• Un conector ICSP

• Botón de Reset

La Arduino UNO posee todo lo que se necesita para manejar el controlador, simplemente se conecta a un ordenador por medio del cable USB o se puede alimentar utilizando una batería o un adaptador AC-DC. Si se conecta por USB, la alimentación externa no es necesaria.

Para programar sólo necesita el IDE de Arduino, que se encuentra en la sección de descargas en el sitio web de www.arduino.cc

Algunas características son:

- Microcontroladora: ATmega328P-PU
- Voltaje operativo: 5 V
- Voltaje de entrada (recomendado): 7 – 12 V
- Límite voltaje de entrada: 20 V
- Digital I/O Pins: 14 (6 de los cuales proporcionan salidas analógicas)
- Pins de entradas analógicas: 6
- Intensidad de corriente continua por I/O Pin: 40 mA
- Intensidad de corriente continua por 3.3V Pin: 50 mA
- Memoria: 32 KB
- SRAM: 2 KB
- EEPROM: 1 KB
- Frecuencia del oscilador: 32 MHz

Las Entradas analógicas son de 10 bits, por lo que entregan valores entre 0 y 1023. El rango de voltaje está dado entre 0 y 5 volts, pero utilizando el pin AREF disponible, este rango se puede variar a algún otro deseado.

PROGRAMA

El programa que va a generar las señales acústico-luminosas es muy sencillo. Para reproducir cada código se ha usado la instrucción “swicht” que nos permite distinguir entre cada letra o número. Además se ha tenido en cuenta que nos pueda permitir usar tanto las mayúsculas como las minúsculas, para ello se ha añadido la opción “case” anidada que hace que se realice la misma función en los dos casos.

El programa, cada vez que se introduce una letra o número en el monitor del programa de Arduino, ejecuta las subrutinas “punto” y “raya” según el código y las muestra en la pantalla del monitor. El programa sólamente ejecuta una letra o número cada vez.

Para ello se ha usado la instrucción `if(Serial.available())` que nos introduce en el bucle siempre que recibamos un dato por la entrada de datos serie. Una vez recibido este dato se asigna a la variable `"val"` con la instrucción `"Serial.read()"` y se va realizando la comparación con todos los códigos mediante la instrucción `"switch"`.

Para simplificar el programa se ha usado el concepto de subrutina. Esta opción nos permite realizar un trozo de programa repetidas veces, sin tener que escribirlo cada vez. La forma de uso de una subrutina es la siguiente: antes de comenzar el programa principal (loop), se realiza una declaración de cada subrutina, en este caso "punto" y "raya". Para estas subrutinas no hay que pasarle ninguna información externa, por eso va seguida de unos paréntesis vacíos ( ). Cada vez que se llama a una subrutina desde el programa principal, el proceso sale de éste y continúa su ejecución en la subrutina llamada. Una vez concluida } vuelve al programa principal en el punto donde lo abandonó.

```
/* unidad didáctica morse */

int pito=9;
int val=0;
int draya=300;
int dpunto=150;
int espera=200;


/* SUBRUTINA PUNTO */
void punto(){
  digitalWrite(pito,HIGH);
  delay(250);
  digitalWrite(pito,LOW);
  Serial.println("Punto");
  delay (espera);
}

/* SUBRUTINA RAYA */
void raya(){
  digitalWrite(pito,HIGH);
  delay(500);
  digitalWrite(pito,LOW);
 Serial.println("Raya");
 delay (espera);
}


void setup (){
  pinMode (pito, OUTPUT);
  Serial.begin (19200);
 }

void loop(){
  if(Serial.available()){
    val=Serial.read();
    switch(val){
      case 'A':
      case 'a':
```

```
  punto();
  raya();
  break;

case 'B':
case 'b':
  raya();
  punto();
  punto();
  punto();
  break;

case 'C':
case 'c':
  raya();
  punto();
  raya();
  punto();
  break;

case 'D':
case 'd':
  raya();
  punto();
  punto();
  break;

case 'E':
case 'e':
  punto();
  break;


case 'F':
case 'f':
  punto();
  punto();
  raya();
  punto();
  break;

case 'G':
case 'g':
  raya();
  raya();
  punto();
  break;

case 'H':
case 'h':
  punto();
  punto();
  punto();
  punto();
  break;

case 'I':
case 'i':
  punto();
  punto();
  break;
```

```
case 'J':
case 'j':
  punto();
  raya();
  raya();
  raya();
  break;

case 'K':
case 'k':
  raya();
  punto();
  raya();
  break;

case 'L':
case 'l':
  punto();
  raya();
  punto();
  punto();
  break;

case 'M':
case 'm':
  raya();
  raya();
  break;



case 'N':
case 'n':
  raya();
  punto();
  break;

case 'Ñ:
case 'ñ:
  raya();
  raya();
  punto();
  raya();
  raya();
  break;

case 'O':
case 'o':
  raya();
  raya();
  raya();
  break;

case 'P':
case 'p':
  punto();
  raya();
  raya();
  punto();
```

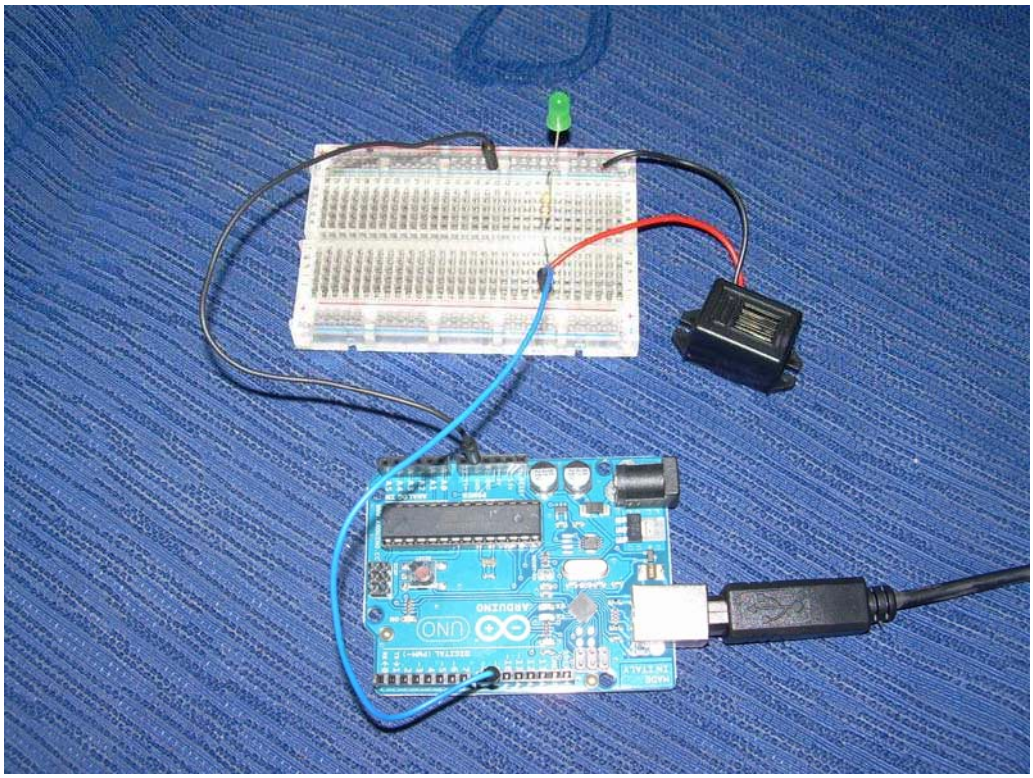
```
  break;

case 'Q':
case 'q':
  raya();
  raya();
  punto();
  raya();
  break;

case 'R':
case 'r':
  punto();
  raya();
  punto();
  break;

case 'S':
case 's':
  punto();
  punto();
  punto();
  break;

case 'T':
case 't':
  raya();
  break;




case 'U':
case 'u':
  punto();
  punto();
  raya();
  break;

case 'V':
case 'v':
  punto();
  punto();
  punto();
  raya();
  break;

case 'W':
case 'w':
  punto();
  raya();
  raya();
  break;

case 'X':
case 'x':
  raya();
  punto();
  punto();
  raya();
```

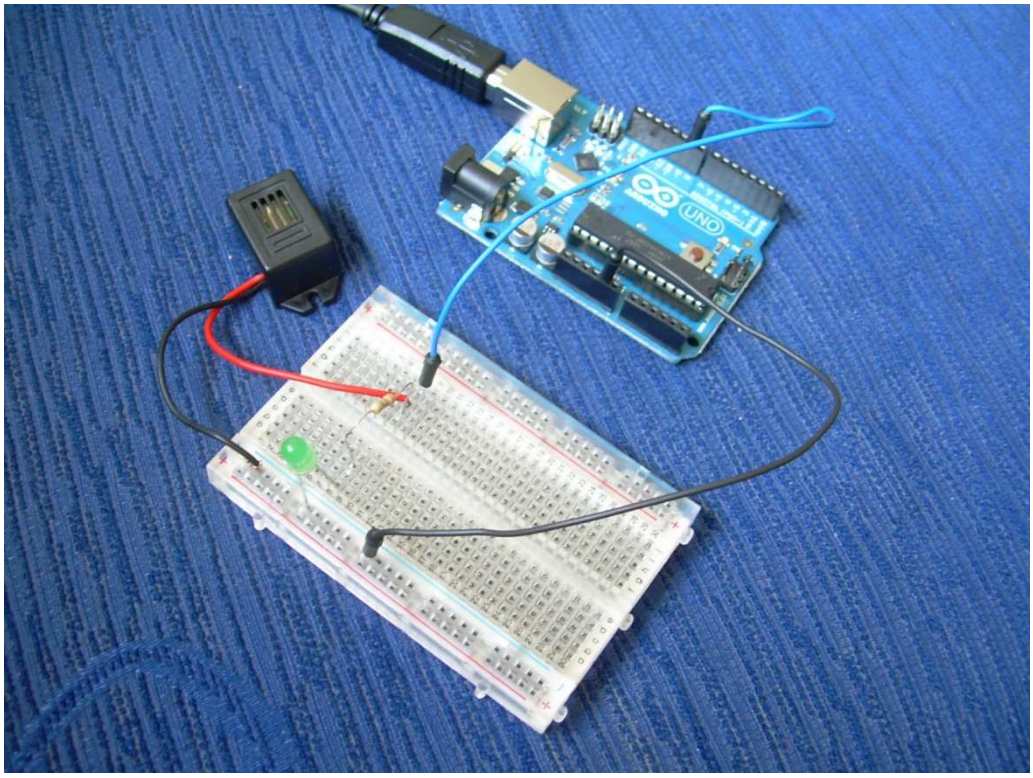
```
  break;

case 'Y':
case 'y':
  raya();
  punto();
  raya();
  raya();
  break;

case 'Z':
case 'z':
  raya();
  raya();
  punto();
  punto();
  break;

case '1':
  punto();
  raya();
  raya(k);
  raya();
  raya();
  break;

case '2':
  punto();
  punto();
  raya();
  raya();
  raya();
  break;

case '3':
  punto();
  punto();
  punto();
  raya();
  raya();
  break;

case '4':
  punto();
  punto();
  punto();
  punto();
  raya();
  break;

case '5':
  punto();
  punto();
  punto();
  punto();
  punto();
  break;

case '6':
  raya();
  punto();
```

```
        punto();
        punto();
        punto();
        break;

      case '7':
        raya();
        raya();
        punto();
        punto();
        punto();
        break;

      case '8':
        raya();
        raya();
        raya();
        punto();
        punto();
        break;

      case '9':
        raya();
        raya();
        raya();
        raya();
        punto();
        break;




      case '0':
        raya();
        raya();
        raya();
        raya();
        raya();
        break;


    }
  }
}
```

FOTOS DEL MONTAJE

De la salida digital número 9 se lleva un cable para conectar el diodo LED con su resistencia limitadora y en paralelo va el zumbador. Asimismo, se lleva de la placa Arduino la toma del negativo de la patilla GND correspondiente





COMPETENCIAS BÁSICAS TRABAJADAS EN ESTA UNIDAD

1.- Competencia en comunicación lingüística.

El código Morse es en si un lenguaje y requiere un tratamiento similar a la hora de aprenderlo como otro lenguaje, de hecho solo consiste en relacionar cada letra o número con su correspondiente código expresado en forma de rayas y o puntos. Se hace obvio que para su aprendizaje es imprescindible la práctica del mismo para mandar y recibir los mensajes con rapidez en este código así como la correcta interpretación de los mismos.

4.- Competencia en el tratamiento de la información y competencia digital.

El uso de las tecnologías de la información y comunicación como un elemento esencial para informarse y comunicarse.

Por otra parte, hemos de trabajar con componentes electrónicos, bien sea analógicos y o digitales. En esta unidad en concreto usamos componentes analógicos (resistencia, LED, zumbador) y en la parte digital, la propia microcontroladora Arduino.

6.- Competencia cultural y artística.

Aunque este tipo de comunicación se hace aún hoy día pero de forma puntual y en ámbitos muy concretos, desde comienzos del siglo XX, tuvo un uso muy generalizado en comunicaciones terrestres, marítimas y militares, por lo que hay que valorarlo como una manifestación del mundo social de una época ya pasada.

7.- Competencia para aprender a aprender.-

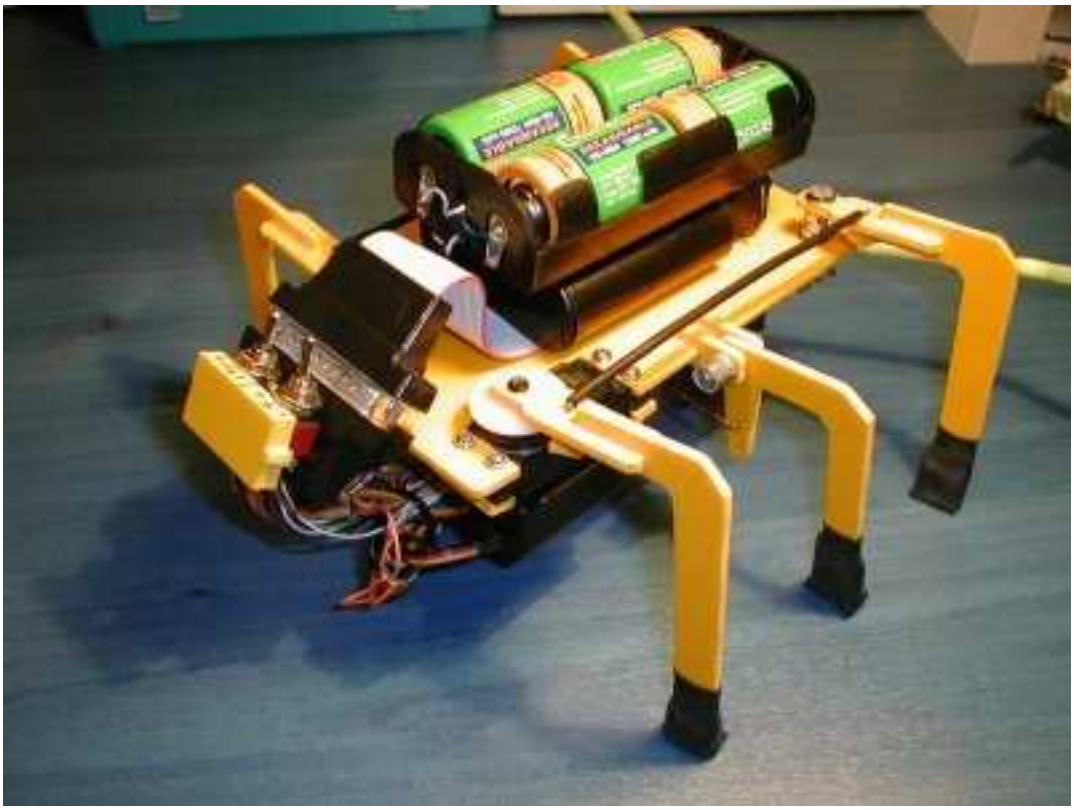
Este código necesita de la memorización de su tipología (raya y punto), por lo que se hace necesario usar unas técnicas de organización, memorización e incluso inventar y desarrollar reglas nemotécnicas para que cada alumno de forma individual pueda recordar el código asociado a cada letra o número. Se hace evidente que para su uso de forma fluida es muy necesaria la práctica de forma continuada.

8.- Competencia de autonomía e iniciativa personal.

Hay que desarrollar la habilidad para relacionarse y trabajar en equipo, ya que si se pretende utilizar este código como medio de comunicación, ha de haber al menos una persona que actúe como emisor del mensaje y otra como receptora e interpretadora del mismo y viceversa.

ROBOT HEXÁPODO

CONTROLADO CON ARDUINO UNO



RAMÓN SÁNCHEZ-FERRAGUT SERRANO

I.E.S. ASTA REGIA. Jerez

DEFINICIÓN

El hexápodo consta de seis patas dispuestas paralelamente en una estructura o chasis, las cuales se mueven dos a dos, gobernadas por un microcontrolador. Estos robots pueden ser más o menos complejos dependiendo de los grados de movilidad de sus patas y de los obstáculos que se quieran sortear, teniendo así que dotar al robot de sensores y crear una aplicación software complejo. Hay muchos modelos diferentes, diferenciándose sobre todo en el tipo de patas empleadas.

OBJETIVOS

- Conocer las partes de un Robot.
- Diseño y construcción de Robot hexápodo.
- Manejar el programa de arduino en sus funciones básicas.
- Programar el robot para que realice las tareas que se especifiquen.

CURSO AL QUE VA DIRIGIDO

El robot hexápodo se proyecta para realizar en cursos de 4º de la ESO.

TEMPORALIZACIÓN

Se diseña para su realización en 4 semanas, 12 sesiones.

METODOLOGÍA

Se utilizará distintas metodologías. Por un lado, la transmisiva que se usará para explicar los distintos conceptos que aparecen; por otro lado la de construcción, ya que deberán llevar a cabo el montaje del robot, y por último la metodología de experimentación, dado que tendrán que realizar ejercicios de programación, descargándolos sobre el robot construido y explicar que realiza este en cada caso, además, de llevar a cabo distintos desafios.

RECURSOS

1 Panel de plástico o madera

3 unidades de servomotores
1 Placa de Arduino UNO
1 Portapilas

HERRAMIENTAS

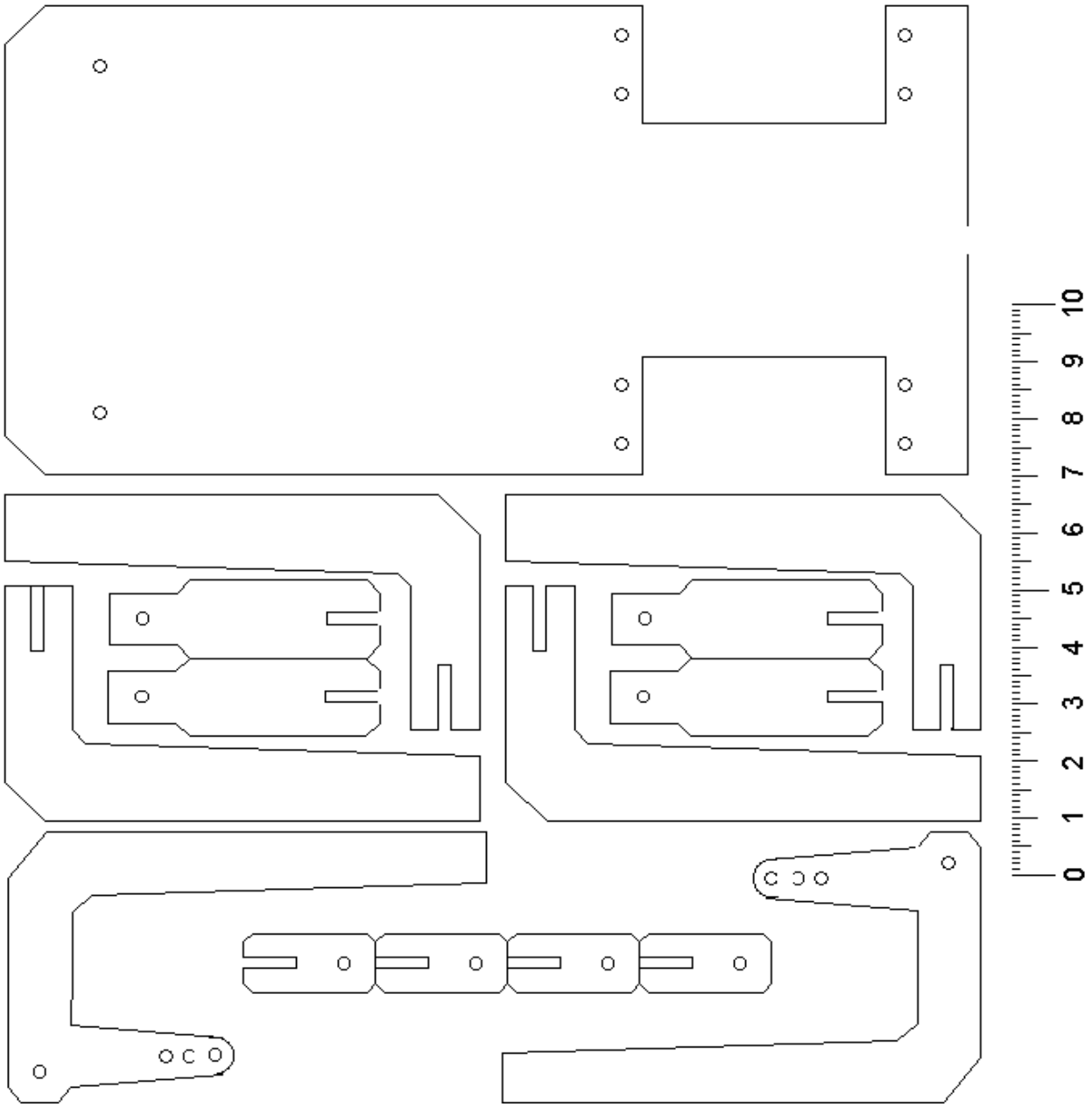
Segueta o sierra de calar
Limas, lijas.
Taladro.
Brocas.

CRITERIOS DE EVALUACIÓN

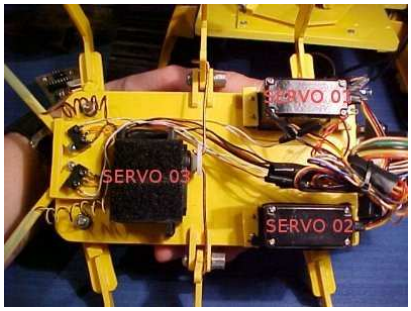
Se valorará la capacidad de conocer las partes de un Robot.
Se valorará la capacidad de construir el Robot utilizando los recursos y herramientas descritas.
Se valorará la capacidad de utilizar el programa para el manejo y control del Robot.

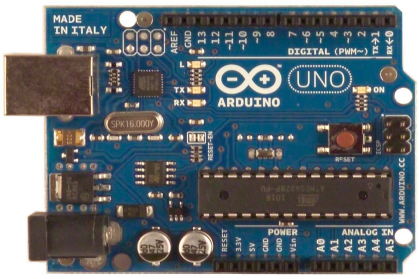
CONSTRUCIÓN

Pasaremos las piezas del plano a la placa de plástico o panel de madera. Recortaremos las piezas y las montaremos según el dibujo (PLANO adjunto). Montaremos los servos en los huecos indicados y le colocaremos las patas pegadas con pegamento rápido.
Una vez construido el esqueleto del Hexápodo, prepararemos la placa controladora realizando el montaje de los servos y programandola.
Colocaremos el portapilas para dar corriente a los distintos elementos y que el hexápodo tenga autonomía de movimiento.
Se realizarán las pruebas y ajustes necesarios para que el Robot Hexápodo realice las funciones que hemos programado.



PROGRAMACIÓN

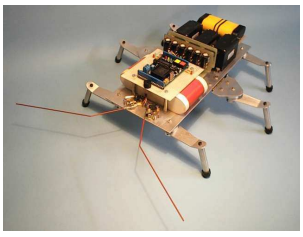


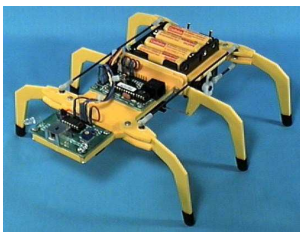



```
#include <Servo.h>

Servo myservo01;  // crea un objeto tipo servo para controlar el servo
Servo myservo02;  // crea un objeto tipo servo para controlar el servo
Servo myservo03;  // crea un objeto tipo servo para controlar el servo


void setup()
{
  myservo01.attach(3);  // liga el servo conectado en el pin 3 al objeto servo
  myservo02.attach(5);  // liga el servo conectado en el pin 5 al objeto servo
  myservo03.attach(6);  // liga el servo conectado en el pin 6 al objeto servo
}

void loop()
{

    myservo03.write(0);       // dice al servo que se posicione en un lado
    delay(15);                 // espera 15 ms para que el servo se posicione
    myservo01.write(0);        // dice al servo que se coloque a un giro avante
    delay(15);                 //espera 15 ms para que el servo se posicione
    myservo02.write(180);      // dice al servo que se coloque a un giro atras
    delay(15);                 // espera 15 ms para que el servo se posicione
    myservo03.write(180);     // dice al servo que se posicione al otro lado
    delay(15);                 //espera 15 ms para que el servo se posicione
    myservo01.write(180);       // dice al servo que se posicione en un lado
    delay(15);                 // espera 15 ms para que el servo se posicione
    myservo02.write(0);        // dice al servo que se coloque a un giro avante
    delay(15);                 //espera 15 ms para que el servo se posicione

  }
}
```

Enlace a vídeo para ver su funcionamiento: http://youtu.be/r2GZPKhXlUY

IMÁGENES DE DIFERENTES MODELOS





Dpto. TECNOLOGÍA
IES FERNANDO SAVATER
tecnologiasavater.blogspot.com

Unidad Didáctica

Semáforo controlado por PC

AUTOR:
FERNANDO MURILLO HALCÓN

INDICE:

1. Contextualización de la unidad. 2

2. Elementos de Aprendizaje: Objetivos, Contenidos y Criterios de evaluación. 2

3. Metodología 5

4. Desarrollo de la Unidad 5

4.1 Fase Inicial 5

4.2 Fase de Desarrollo 6

5. Evaluación de la Unidad 8

5.1Instrumento de evaluación 8

TITULO: Semáforo controlado por PC

Autor: Fernando Murillo Halcón

1. Contextualización de la unidad.

El instituto se encuentra situado en una manzana junto a un colegio y un parque; existe bastante tráfico a su alrededor, y sobre todo en determinados momentos. No hay elementos para regulación del tráfico, y es por ello que queremos diseñar y construir un semáforo para este fin para dar respuesta a este problema y situarlo en el punto más adecuado que estimen oportuno teniendo en cuento todo tipo de factores.

"*Construir y programar la secuencia de funcionamiento de un semáforo para vehículos y junto con uno para peatones*".

Condiciones de diseño:

- La maqueta del semáforo tendrá una base o soporte con las medidas máximas de 20*20 cm y su altura no será superior a 25 cm.

- La secuencia de funcionamiento y control con el ordenador se hará con la tarjeta Arduino y con su entorno de programación.

- Deberá simular las pautas de funcionamiento de un semáforo real.

- A excepción de cables, LEDs, resistencias, etc, los demás elementos necesarios para construir la maqueta se construirán en el aula, valorándose el uso de materiales reciclados.

- El proyecto-construcción se hará por parejas.

Nivel: 4º de ESO
Bloque de Contenidos: Control por ordenador
Temporalización: 12

Sesiones Fase Inicial	Sesiones Fase Desarrollo	Sesiones Fase Síntesis y Eval.	Total Sesiones
1	10	1	12

2. Elementos de Aprendizaje: Objetivos, Contenidos y Criterios de evaluación.

Se pretende que esta unidad aporte a los objetivos de área y etapa aquellos aspectos que se vayan a trabajar con el fin de cumplir esos objetivos, como refleja la tabla adjunta

Relación de Objetivos de Etapa y área					Tecnologías							
					1	2	3	4	5	6	7	8
					Metodo Proyectos	Manipulac. Técnica	Método de Análisis	Expresión de ideas	Tecnolog. VS M. Amb.	Informática-Redes	Nuevas tecnologías	Trabajo en equipo
Educación Secundaria Obligatoria	Real Decreto		a	Respeto a los demás					X			X
			b	Trabajo en equipo		X						X
			c	Igualdad de sexos								X
			d	Relaciones con otros								X
			e	Tecnologías Información	X					X	X	
			f	Método científico	X		X		X			
			g	Iniciativa autodidacta	X		X		X		X	X
			h	Lengua castellana	X			X				
			i	Lengua extranjera	X		X	X				
			j	Historia, arte y cultura					X	X		
			k	Salud, cons y M.Amb		X			X		X	
			l	Creación artística	X			X				
			m	Conocer y apreciar			X					
	Andalucía	And	a	Habilidades domést		X	X					X
		And	b	Códigos cient y tecn	X	X	X	X				
		And	c	Democracia, ciudadanía					X			
		And	d	Medio Ambiente					X		X	X
		And	e	Lengua Andaluza						X		
		And	f	Cultura Andaluza			X		X			

Objetivos

- Conocer el funcionamiento y utilizar una tarjeta controladora.
- Aprender a utilizar los diagramas de flujo al realizar tareas de programación.
- Introducir el concepto de controladora.
- Mostrar cuáles son las principales controladoras disponibles en el aula de Tecnología y en el ámbito educativo.
- Mostrar las conexiones básicas.
- Conocer las interfaces de alguna de las controladoras empleadas en el taller de tecnología.
- Conocer los fundamentos básicos del lenguaje para la controladora.
- Presentar el diagrama de bloques de un sistema de control por ordenador.
- Revisar el concepto de señal analógica y de señal digital.
- Mostrar las acciones básicas que pueden realizarse con un control de ordenador: accionamiento de interruptores y motores, captación de señales de sensores.
- Presentar un sistema sencillo de control por ordenador.

Contenidos

Conceptos

- Control por ordenador.
- Controladoras e interfaces de control.
- Dispositivos de entrada-salida de control.
- Tipos de controladoras.
- Codificación de programas.
- Interfaces de control y programación.
- Diagramas de flujo.

Procedimientos, destrezas y habilidades

- Utilizar la tarjeta controladora.
- Interpretar y elaborar de diagramas de flujo.
- Diseñar programas para controlar las entradas y salidas digitales de una controladora.
- Utilizar una controladora para regular el funcionamiento de circuitos eléctricos con la ayuda de un ordenador.
- Elaborar programas sencillos en el lenguaje de programación de la controladora y utilizarlos a continuación para el control de sistemas.

Actitudes

- Gusto por el orden y la limpieza en la elaboración de dibujos y esquemas.
- Valorar positivamente el impacto que puede suponer en la vida cotidiana, en particular en el hogar, la adopción de automatismos y el control remoto por ordenador.

- Apreciar el trabajo complejo y planificado que exige el montaje de sistemas de control.
- Interés por abordar problemas que, a priori, pueden parecer difíciles de solucionar.
- Interés por abordar trabajos en grupo.

Con esta unidad queremos que nuestros alumnos/as trabajen los siguientes campos que les permitan llegar a ser capaces de:

1.- Resolución de problemas, buscando soluciones y sin miedo a equivocarse, con iniciativa, valorando positivamente tanto los aciertos y los errores, y aprendiendo de ellos – Competencia en autonomía e iniciativa personal.

2.- Aprendan a trabajar en grupo y valoren la importancia que cada uno de los componentes del mismo tiene a nivel individual como grupal. – Competencia social y ciudadana

3.- Aprendan a controlar un proyecto con el ordenador, comprendiendo las variables que le afectan y puedan darle diferentes formas de funcionamiento. – Competencia digital y tratamiento de la información.

4.- Aprendan el lenguaje de programación para la tarjeta controladora, y que sepan interpretar lo que con él se escribe. – Competencia en comunicación lingüística.

Los criterios de evaluación que se emplearán para esta unidad serán los siguientes:

1. Distinguir los principales elementos de entrada y salida de un sistema de control.
2. Describir las características de una controladora, prestando especial atención a sus salidas y entradas, tanto analógicas como digitales.
3. Utilizar la controladora para examinar el funcionamiento de un sistema a través del ordenador.
4. Elaborar procedimientos sencillos de control mediante lenguaje de programación.
5. Elaborar diagramas de flujo.
6. Elaborar programas que controlen las entradas y salidas de una controladora.
7. Manejar sencillos circuitos electrónicos a partir de un ordenador y una controladora.

3. Metodología

La metodología va a ser completamente práctica, facilitando y fomentando la autonomía personal, individual y grupal de las parejas. Tendrán que manipular, manejar, probar y experimentar por ensayo/error las soluciones a las que se vaya llegando, y así, valora las mismas y determinar cuál es la más apropiada para la resolución del problema.

4. Desarrollo de la Unidad

4.1 Fase Inicial

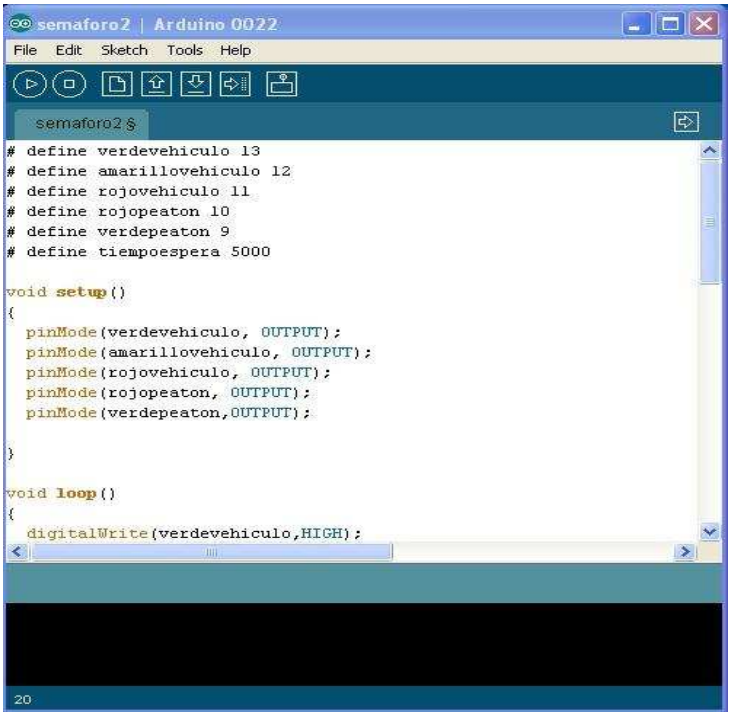
En esta fase se planteará la problemática en los alrededores del Centro y las posibles soluciones que les podríamos dar. Todo ello trabajando con el grupo, mediante charla-debate-lluvia de ideas, en la que cada uno aportaría su opinión. Planteada y elegida la solución del semáforo, ahora exponemos otro problema que puede surgir: los tiempos para el paso de vehículos y peatones.

4.2 Fase de Desarrollo

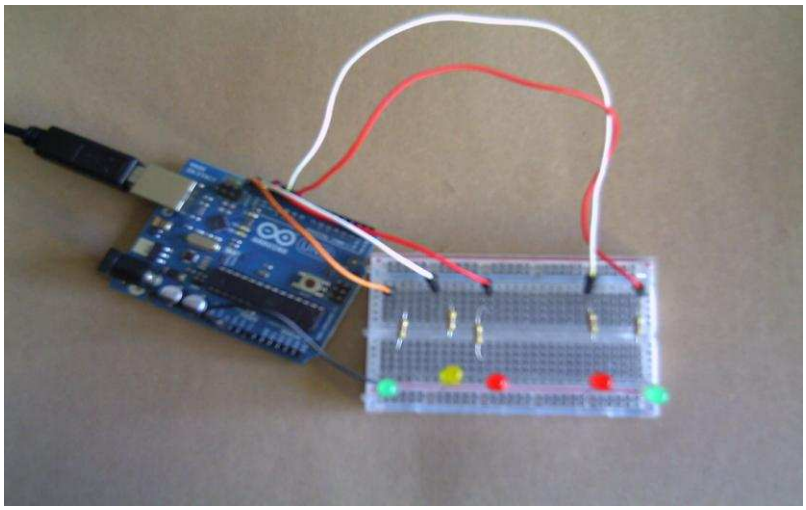
Esta es la programación que se intentará que lleguen y mejore el alumnado.

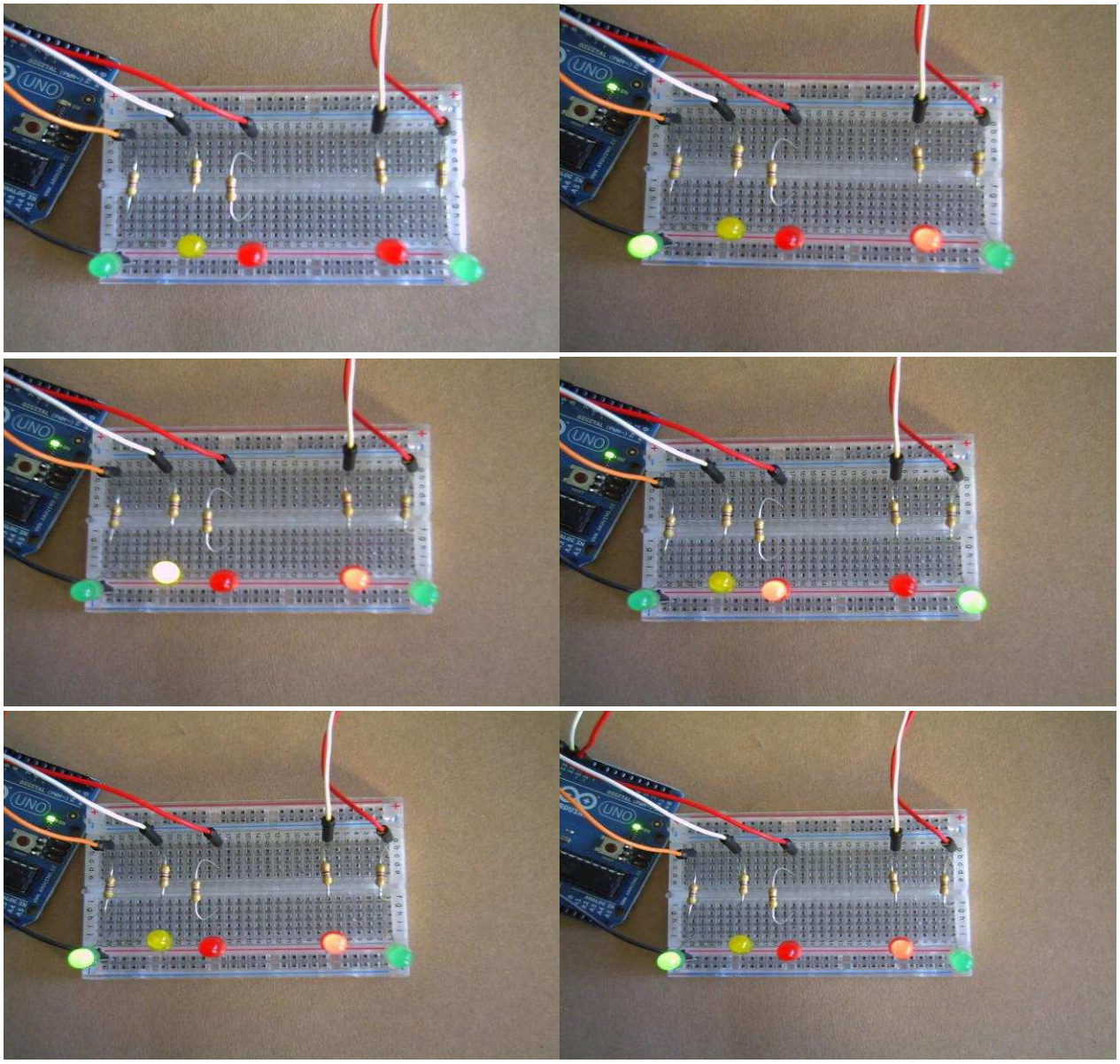
```
# define verdevehiculo 13
# define amarillovehiculo 12
# define rojovehiculo 11
# define rojopeaton 10
# define verdepeaton 9
# define tiempoespera 5000

void setup()
{
 pinMode(verdevehiculo, OUTPUT);
 pinMode(amarillovehiculo, OUTPUT);
 pinMode(rojovehiculo, OUTPUT);
 pinMode(rojopeaton, OUTPUT);
 pinMode(verdepeaton,OUTPUT);
}

void loop()
{
 digitalWrite(verdevehiculo,HIGH);   //verde coches encendido
 digitalWrite(rojopeaton,HIGH); //rojo peaton encendido
 delay(tiempoespera);        // verde coches tiempo espera
 digitalWrite(verdevehiculo,LOW);  // verde coches apagado
 digitalWrite(amarillovehiculo,HIGH); //amarillo coches encendido
 delay(tiempoespera);        //amarillo coches tiempo espera
 digitalWrite(amarillovehiculo,LOW); // amarillo coches apagado
 digitalWrite(rojopeaton,LOW); //rojo peaton apagado
 digitalWrite(rojovehiculo,HIGH); // rojo coches encendido
 digitalWrite(verdepeaton,HIGH);//verde peatos encendido
 delay(tiempoespera);                // rojo coches tiempo espera..
 digitalWrite(verdepeaton,LOW);// verde peaton apagado
 delay(1000);                // tiempo espera intermitencia verde peaton
 digitalWrite(verdepeaton,HIGH);// verde peaton encendido intermitente
 delay(1000);                // tiempo espera intermitencia verde peaton
```

```
  digitalWrite(verdepeaton,LOW); // verde peaton apagagado intermitente
  delay(1000);                   // tiempo espera intermitencia verde peaton
  digitalWrite(verdepeaton,HIGH);
  delay(1000);                   // tiempo espera intermitencia verde peaton
  digitalWrite(verdepeaton,LOW); //apagar verde peatones
  delay(500);                    // tiempo espera fin intermitencia verde peaton
  digitalWrite(rojovehiculo,LOW); // rojo coches apagado
  digitalWrite(rojopeaton,HIGH); // rojo peaton encendido
  digitalWrite(rojovehiculo,HIGH); // rojo coches encendido-junto al rojo peaton-seguridad
  delay(800);                    // tiempo espera encendido rojos coches y rojo peaton
  digitalWrite(rojopeaton,LOW); // rojo peaton apagado
  digitalWrite(rojovehiculo,LOW); // rojo coches apagado
  //final_loop();
 }

 void final_loop()
 {
  while (true == true) {
  }
}
```



Nuestra tarjeta debe de responder de la siguiente manera:





5. Evaluación de la Unidad

Se evaluará usando los siguientes instrumentos de evaluación:

- ACTIVIDADES. INDIVIDUALES Y DE GRUPO (en clase y en casa).

- Cuaderno de clase.
- Resolución de ejercicios y problemas.
- Trabajo de construcción y diseño.

- PROYECTOS (conceptos, procedimientos y actitudes)

- Acabado.
- Relación entre lo diseñado y lo construido.

- Cuaderno de proyectos.
- Exposición del proyecto acabado. Uso de vocabulario técnico.
    - ACTITUD.
- Participación e interés.
- Disciplina.
- Cuidado y respeto del material, herramientas, mobiliario, etc., así como a sus compañeros.
- Seguridad y orden en el puesto de trabajo.
- Aprovechamiento del material.
    - ASISTENCIA Y PUNTUALIDAD.

En todos los casos se valorará el orden, la limpieza, los contenidos, la presentación, la expresión escrita y oral.

REALIZAR UN MEDIDOR DE LUZ CON DOS DISPLAYS 7 SEGMENTOS

OBJETIVOS

Realizar un medidor de luz con dos displays 7 segmentos (medida en porcentaje de 0 a 99%)

COMPETENCIAS

Especificas.

- Conocer el uso y funcionamiento del display de 7 segmentos
- Conocer el uso y funcionamiento de Arduino como multiplexor.
- Utilizar Arduino para desplegar números decimales en el display de 7 segmentos.

Generales.

- Manejar las herramientas apropiadas.
- Utilizar de forma coherente y correcta las unidades adecuadas para cada magnitud.
- Presentar los resultados de los cálculos con la precisión requerida.
- Utilizar herramientas informáticas de simulación.
- Montar circuitos y realizar medidas en ellos para comprobar cálculos previos.
- Realizar informes sobre las prácticas realizadas que incluyan una adecuada explicación teórica, los cálculos y simulaciones realizadas, los resultados medidos y los errores encontrados.

METODOLOGÍA

Se combinarán diferentes métodos didácticos con el fin de presentar los contenidos bajo distintas perspectivas que favorezcan la motivación del alumnado:

MÉTODO EXPOSITIVO

- Informativa
- Esquema general de los contenidos a tratar.
- Exposición explicativa de conceptos utilizando diferentes recursos de exposición: pizarra, proyector (presentaciones), directamente sobre el/los Toma de apuntes e información complementaria (esquemas, diagramas,.)

MÉTODO PRÁCTICO.

- Adquisición de destrezas mediante la realización de trabajos prácticos.

TEMPORALIZACIÓN.

Dos sesiones de 2 horas (1 hora de exposición y 3 de desarrollo de la práctica).

MATERIAL:

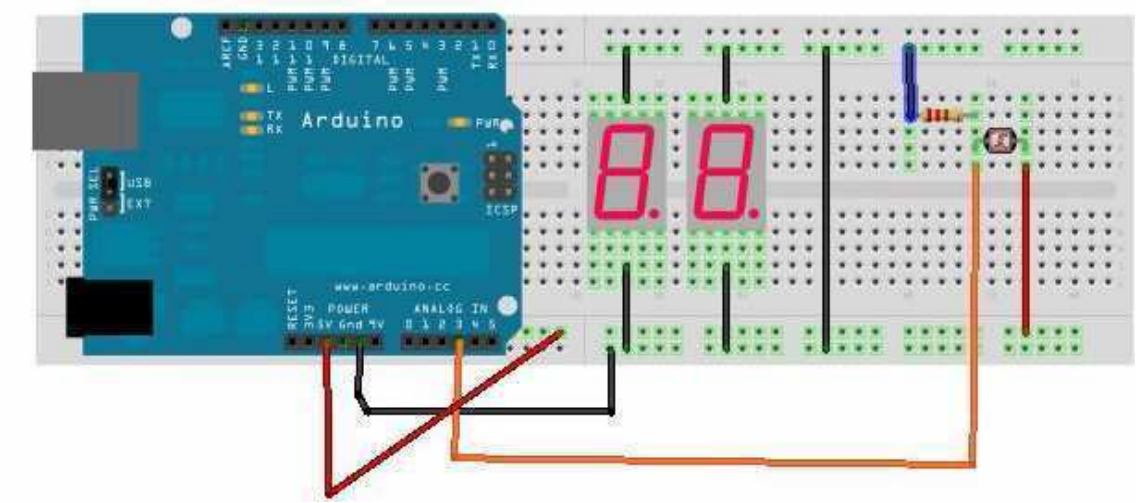
- 2 Displays 7 segmentos de cátodo común
- 1 LDR

• Resistencia de 1 k
• Placa arduino
• Cable.

FUNDAMENTO TEÓRICO

Una LDR (Light Dependent Resistors) es una resistencia que disminuye el valor óhmico al aumentar la luz que incide sobre ella. Se emplean como sensores de luz, barreras fotoeléctricas, etc.

MONTAJE



PROGRAMA

```
int pin_ldr = 3; // Patilla de arduino para la Medida del sensor
int lectura = 0; // Variable donde almacenamos la medida
int val = 0; // Variable donde almacenamos el mapeo
int unidad = 0; //variable donde se almacenará el dígito unidad de la medida mapeada
int decena = 0; // variable donde se almacenará el dígito decena de la magnitud medida

void loop()
{
lectura = analogRead(pin_ldr);// Tomamos la medida
//Serial.println(lectura);
//delay(1000);
val = map(lectura, 0, 1023, 0, 99); //Mapeamos entre esos valores
//Serial.println(val);
descompon(val); //Ejecutamos esta función para descomponer el valor en dos dígitos
//Serial.println(unidad);
//Serial.println(decena);
//delay(1000);
switch (unidad) {
case 0:
D1_num_0();
break;
case 1:
D1_num_1();
break;
case 2:
D1_num_2();
break;
case 3:
D1_num_3();
break;
case 4:
D1_num_4();
```

```
break;
case 5:
D1_num_5();
break;
case 6:
D1_num_6();
break;
case 7:
D1_num_7();
break;
case 8:
D1_num_8();
break;
case 9:
D1_num_9();
break;
//default:
}
switch (decena) {
case 0:
D2_num_0();
break;
case 1:
D2_num_1();
break;
case 2:
D2_num_2();
break;
case 3:
D2_num_3();
break;
case 4:
D2_num_4();
break;
case 5:
D2_num_5();
break;
case 6:

D2_num_6();
break;
case 7:
D2_num_7();
break;
case 8:
D2_num_8();
break;
case 9:
D2_num_9();
break;
//default:
}
}
void descompon (int valor)
{
if(10 - val > 0)
{
unidad = val;
decena = 0;
}
else if(20 - val > 0)
{
unidad = val - 10;
decena = 1;
}
else if(30 - val > 0)
{
unidad = val - 20;
decena = 2;
}
else if(40 - val > 0)
{
unidad = val - 30;
decena = 3;
}
```

```
else if(50 - val > 0)
{
unidad = val - 40;
decena = 4;
}
else if(60 - val > 0)
{
unidad = val - 50;
decena = 5;
}
else if(70 - val > 0)
{
unidad = val - 60;
decena = 6;
}
else if(80 - val > 0)
{
unidad = val - 70;
decena = 7;
}
else if(90 - val > 0)
{
unidad = val - 80;
decena = 8;
}
else
{
unidad = val - 90;
decena = 9;
}
```