

Control de un circuito de Slot con Arduino

ALEJANDRO DOMÍNGUEZ AL-LAL
JOSE LUIS ARROYO GARCÍA

CICLO SUPERIOR DE ADMINISTRACION DE SISTEMAS
INFORMATICOS EN RED

Control de un circuito de Slot con Arduino

Índice	página 1
Introducción y justificación	página 2
Objetivos	página 3
Análisis de contexto	página 4
Desarrollo de contenido	página 7
Módulo semáforo	página 7
Módulo cuentavueltas-cronometro	página 8
Módulo menú	página 10
Módulo de control	página 15
Conclusión y valoración final	página 17
José Luis Arroyo García	página 17
Alejandro Domínguez Al-lal	página 18
Bibliografía	página 21
Anexos	página 22

INTRODUCCION Y JUSTIFICACION

El presente proyecto consiste en la utilización de una placa microcontroladora Arduino para el control de un coche de slot, a través de los distintos módulos que esta placa tiene disponibles, así como el uso de la misma como dispositivo cuentavueeltas, cronometro y semáforo.

La idea surge del hecho de que habitualmente estos módulos pueden ser comprados a los fabricantes de slot en forma de piezas separadas, lo que hace que el precio sea bastante alto, (a veces, dependiendo de las marcas, inaccesibles desde las tiendas físicas del país), además del hecho de que cada dispositivo suele llevar su propia alimentación, con el consecuente aumento del material a preparar.

El concepto del proyecto es la unificación de todos estos dispositivos en uno solo, siendo en este caso el dispositivo Arduino, congregando las funciones de semáforo, cuentavueeltas y trainer, reduciendo el tamaño y facilitando el acceso a este dispositivo, tanto económica, como físicamente, por poder encontrarse vendedores de estos dispositivos por todo el continente.

Además, el hecho de usar componentes electrónicos que pueden ser comprados en cualquier tienda de electrónica, así como por disponibilidad por ser bastante asequibles, facilita su difusión, así como su expansión, al ser altamente configurable y ampliable, tanto físicamente como a través de software.

Otro punto a favor del uso de estos dispositivos, es la gran cantidad de recursos que podemos encontrar en internet, tanto libros como materiales, así como videos y tutoriales gratuitos para adquirir los conocimientos de programación necesarios, y los conocimientos sobre electricidad y electrónica necesarios para su puesta en funcionamiento.

OBJETIVOS

Los objetivos a cumplir son la unificación de las funciones semáforo, cuentavueltas, cronometro y trainer en un solo dispositivo, el cual, con las mínimas configuraciones, tanto físicas como de código, permita la realización de estas funciones de la manera más económica posible.

En el caso del trainer, consistirá en un sistema por el cual, la Arduino, sea capaz, de manera autónoma, de controlar un coche en una pista.

En el caso de los demás accesorios la Arduino servirá como dispositivo para el control de tiempos (última vuelta, vuelta más rápida y vuelta actual.), mientras que el semáforo servirá para controlar la alimentación de pista, para que la salida sea de la manera más justa posible (solo comenzara a correr cuando el sistema lo permita.) además, contara con un sistema por el cual una vez cumplido el numero de vueltas designado, procederá al corte de la alimentación de corriente de pista, terminando de esa manera la carrera.



Ejemplo de placa Arduino

ANALISIS DE CONTEXTO

En este punto, explicaremos los materiales que utilizaremos de forma simple, siendo esta información ampliada más adelante cuando se explique su implementación

La idea a desarrollar es, mediante el uso de una placa Arduino en alguno de sus distintos tipos, la implementación de un sistema de control y gestión unificado para un circuito de Slot, siendo este sistema escalable y lo más económico posible.

En este caso, debido a la cantidad de dispositivos que se pueden conectar, usaremos una placa Arduino modelo MEGA 2560 REV.3, por ser este modelo el que más capacidad tiene, tanto en escalabilidad como en capacidad de procesamiento, al ser una de las placas más potentes de las que podemos encontrar en el mercado.

Para el cuentavueeltas usaremos una fotorresistencia, configurada para que cree una señal en el momento en que deja de darle luz (cuando el coche pase por encima). Este elemento, además, servirá a su vez para el módulo cronómetro, ya que marcará el momento en el cual debe de comenzar a cronometrar cada vuelta.

Para el semáforo usaremos 4 diodos leds (tres rojos y uno verde), que servirán, además de la señal de zumbador (o altavoz), como señales luminosas y sonoras de la secuencia de inicio de carreras. Además, se utilizará un rele, para gestionar el suministro de corriente a la pista.

En el caso del trainer, para el control del coche en pista, se utilizarán varias fotorresistencias (al tratarse de un circuito sencillo, con forma de ovalo, optaremos por el uso de 4 de estos elementos, antes de entrar en las curvas y al salir) como punto de referencia para la Arduino en la que aplicar las diferentes velocidades. Como elemento de control del coche, usaremos un regulador de voltaje L298N, que

puede adquirirse ya montado, pudiendo ser configurado de varias maneras dependiendo de lo que la situación requiera.

Hay que destacar, que en un principio, la idea para el proyecto era más compleja, siendo esta la de la utilización de un circuito divisor de tensión, para sacar desde una partida grabada previamente a través del mando, un archivo capaz de ser leído y reproducido por la Arduino, siendo el modulo divisor un DRock-L2596S, estando este archivo guardado en una tarjea microSD, pudiendo ser así, exportado a otros sistemas, agregando la posibilidad de poder jugar con varios coches en pista, aunque solo haya una persona jugando.

El problema que ha surgido para que esta idea haya sido desechada, el un retraso en la recepción de piezas por motivos ajenos a nuestro control (huelga de correos), así como problemas con archivos de librerías necesarios para la utilización de ciertos componentes, tales como una pantalla de 3'5 pulgadas con lector de microSD integrada, la cual, después de venir con retraso, no se encontraron las librerías adecuadas para su utilización hasta cuatro días después de su recepción, acumulando así, un retraso significativo en el proceso de implementación.

Debido a estos problemas se ha decidido la utilización de fotorresistencias para marcar el punto en que el módulo de control deberá variar su velocidad, para adecuarse a la parte del circuito en la que se encuentra, por lo que la utilización de un sistema para guardar una partida, queda descartado para este desarrollo, aunque es posible implementarlo en un futuro.

Como consecuencia de la falta de tiempo para la correcta configuración de la pantalla y el lector, se utilizara una pantalla diferente, más simple, y con mucha menos resolución, siendo sus especificaciones comentadas más adelante. Además, de la simplificación de las variables a almacenar, al disponer de menos capacidad de memoria para su uso.

A partir de este punto, se abren dos posibles vías de desarrollo para el proyecto, siendo estas la implementación de la idea anteriormente descrita, o un sistema para control remoto del módulo de control del coche mediante el uso de una conexión a internet, siendo para ello necesario la adquisición y configuración de un shield con conector RJ-45, con sus correspondientes librerías, así como la creación de una interfaz de control accesible vía web, lo cual permitiría, mediante la inclusión de una cámara web al sistema, de la posibilidad de hacer partidas con usuarios jugando desde otra localización, lo que permitiría romper una de las barreas con las que cuenta este tipo de juegos en la actualidad.

Otra posibilidad sería la de agregar ambas funciones al sistema, con lo que conseguiríamos generar una nueva manera de usar estos juegos, al poder jugar una partida con un amigo a través de internet, y, una vez terminada la partida, poder guardar dicha partida para su posterior utilización.

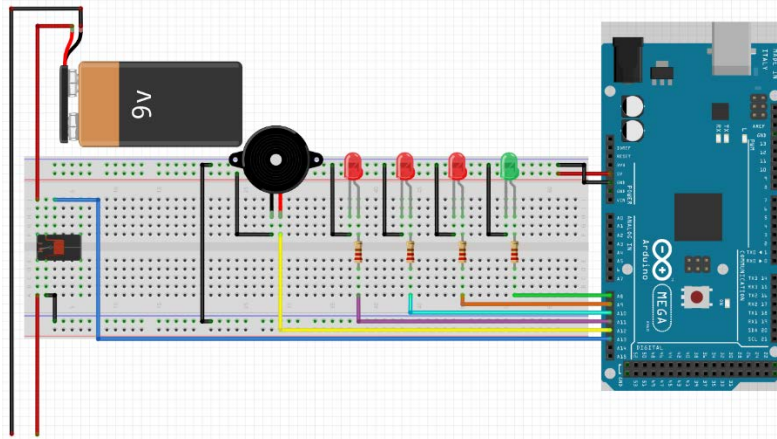
Aun así, para su implementación, haría falta bastante tiempo, debido a la falta de conocimientos necesarios sobre los elementos Arduino por parte de los autores de este documento.

DESARROLLO DE CONTENIDO

En esta sección detallaremos los diferentes módulos que conforman este proyecto, mostrando algunas partes de código, así como sus esquemas de circuitería.

Los módulos son los siguientes:

Modulo semáforo



Este módulo se encargara de gestionar el inicio y el fin de la carrera.

Como podemos comprobar por la imagen, contamos con un rele para gestionar el paso o no de corriente a la pista, cuatro leds y un altavoz para las señales de inicio de carrera.

Aunque en la imagen se muestra como alimentación para el rele de una pila de nueve voltios, en realidad será un trasformador de dieciséis voltios de slot lo que tendrá conectado.

La idea de este módulo es que una vez vayamos a comenzar la partida comience una secuencia con los leds, en la cual, se van encendiendo de manera progresiva, acompañada de un pitido del altavoz con cada led, y terminando, cuando enciende el led verde con un pitido más largo y al activación a la vez del rele, permitiendo así el paso de corriente a pista.

Extracto de la secuencia en la que el rele se enciende tras haber apagado los demás leds

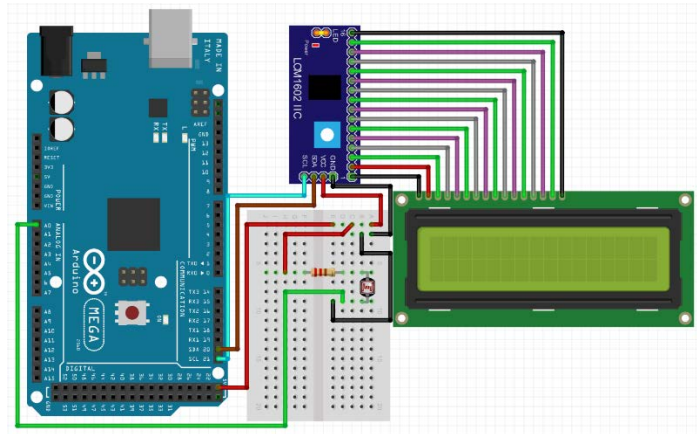
```
digitalWrite(rojo1, LOW); //Apaga los leds rojos
digitalWrite(rojo2, LOW);
digitalWrite(rojo3, LOW);
digitalWrite(verde, HIGH); // Enciende el verde
digitalWrite(rele, HIGH); // Enciende el rele
tone(buzzer, sol, intervalo*2);
delay(intervalo*2);
digitalWrite(verde, LOW); //Apaga verde
```


Control de un circuito de Slot con Arduino

Una vez terminada la partida, el rele se desactivara, cortando la corriente del circuito.

Modulo cuentavueltas-cronometro

Este módulo se encargara, mediante el uso de una fotorresistencia, de monitorizar los tiempos y las vueltas que lleva el coche en pista, siendo esto datos mostrados en pantalla. El circuito es el siguiente:



A diferencia de las pantallas que podemos encontrar de manera genérica en los distintos vendedores de componentes que podemos encontrar por internet, las pantallas que se usan en este módulo, llevan conectados un módulo I2C, el cual se encarga de multiplexar los dieciséis pines de la pantalla a solo cuatro, siendo, de izquierda a derecha, los dos primeros del control de los datos que se suministran a la pantalla, y los dos últimos para alimentación de la misma.

Para el uso de esta pantalla, se comprobó que las librerías genéricas incluidas en el entorno de programación de Arduino no son válidas para dicho complemento, por lo que se vio la necesidad de buscar y descargar una librería externa que soportase dicha pantalla.

Aquí hay que añadir, como se ha dicho en la sección anterior, que esta pantalla no es la que se iba a utilizar en un principio, siendo la otra pantalla, de 3'5 pulgadas, a color, con una resolución mucho mayor (480x320), y con lector de tarjeta incorporado.

El principal problema para su utilización es la falta de tiempo para configurar correctamente dicha pantalla, debido también a la falta de controladores genéricos en el IDE de Arduino, teniendo que contactar con el vendedor de la pantalla para que este suministrase las librerías correctas, puesto que las que se pueden encontrar por internet, no soportan la pantalla, debido a errores de programación por parte del desarrollador.

Por esos motivos, la pantalla escogida es un LCD como el de la imagen, con una resolución de 16x2, lo que limita en gran medida la cantidad de datos a mostrar.

Control de un circuito de Slot con Arduino

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

//variables
const byte lightPin = 0;
int lightSensor = 0;
unsigned long lapMillis = 0;
unsigned long startMillis = millis();
unsigned long bestMillis = 9999999;
float lapTime = 0.00;
float bestLap = 0.00;
boolean firstTrigger = true;
boolean newBest = false;
int vuelta = 1;
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  lcd.begin();
  lcd.setCursor(0, 0);
  lcd.print("Start!          ");
  Serial.begin(9600);
}

void loop() {

  //Si detecta el coche
  if(analogRead(lightSensor) < 15){

    //Determina el tiempo de la vuelta en milisegundos
    lapMillis = millis() - startMillis;
    startMillis = millis();
    lapTime = lapMillis / 1000.00;

    //Si es el primer paso, imprime "race started"
    if (firstTrigger == true){
      lcd.setCursor(0, 0);
      lcd.print("Race Started!    ");
    }

    //print if not first trigger
    if (firstTrigger != true){

      Serial.print("lapTime: ");
      Serial.println(lapTime);
      Serial.print("lapMillis: ");
      Serial.println(lapMillis);
      //print lap time
      lcd.setCursor(0, 0);
      lcd.print("Last: ");
      lcd.print(lapTime);
      lcd.print("    ");
      lcd.setCursor(13,0);
      lcd.print("Lap");
      lcd.setCursor(14,1);
```

```
        lcd.print(vuelta);

        vuelta= vuelta+1;
    }

    //if last lap is better than best lap
    if(lapMillis < bestMillis && firstTrigger != true)
    {
        bestMillis = lapMillis;
        bestLap = lapTime;
        Serial.print("bestLap: ");
        Serial.println(bestLap);
        //print best lap time
        lcd.setCursor(0, 1);
        lcd.print("Best: ");
        lcd.print(bestLap);
        lcd.print("  ");
        newBest = true;
    }
    firstTrigger = false;
    Serial.println("-----");
}
}
```

Durante el proceso de unificación de los módulos se descubrió por un fallo de implementación en el módulo cronometro, que una sección provocaba que este módulo retrasase a todas las demás secciones, además de provocar lecturas erróneas en los sensores, al haber usado la función delay en vez de la función millis, puesto que la función delay provoca la interrupción de la ejecución de todo el código cargado en la Arduino.

Módulo Menu

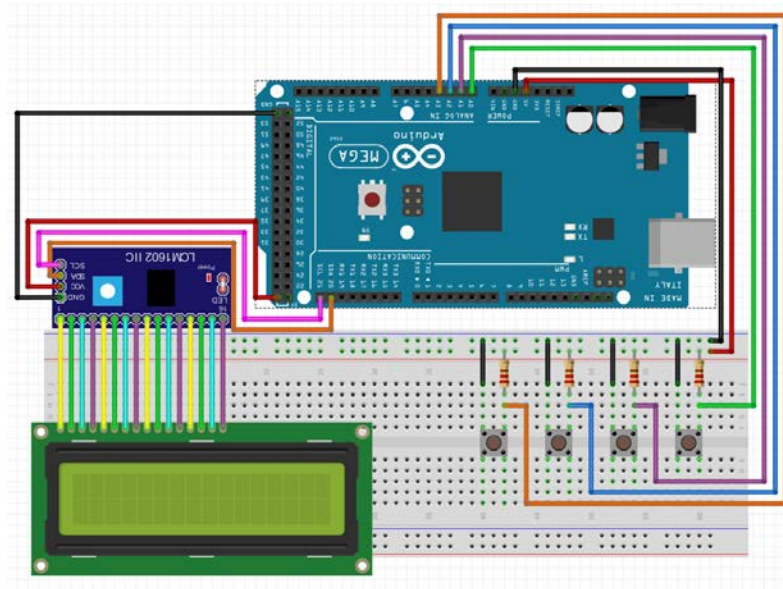
Este módulo será el encargado de preguntar el número de vueltas, así como el encargado de mostrar el mejor tiempo de vuelta.

Como podemos apreciar en la imagen, este módulo no difiere mucho en el implementado para el cuentavueltas, siendo la única diferencia que en este módulo, únicamente nos encargaremos de la elección de vueltas, la de mostrar el mejor tiempo, y activar el modulo semáforo para comenzar la carrera. Los botones que se muestran sirven uno para incrementar el número de vueltas, otro para bajarlo, otro para aceptar

Control de un circuito de Slot con Arduino

y otro como reset, el cual para la carrera y vuelve a la pantalla de selección del número de vueltas.

En este caso el código lleva además integrado ya el semáforo, siendo esta la razón por la que no se ha puesto el código completo en su apartado.



```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

int boton1 = 39;
int boton2 = 37;
int boton3 = 41;
int boton4 = 35;
int vuelta = 0;
int ok = 0;
int val = 0;    // variable que da el estado

int rojo1 = 52; //Pin led rojo 1
int rojo2 = 53; //Pin led rojo 2
int rojo3 = 50; //Pin led rojo 3
int verde = 51; //Pin led verde
int buzzer = 47; //Pin del buzzer
int doh=261.63;
int sol=392;
int intervalo = 1000; //Intervalo de la secuencia
int intervalo2 = 500; //Intervalo de la secuencia
int rele = 42;
```

```
void setup() {
  lcd.backlight();
  lcd.begin();
  lcd.print("Gestor de");
  lcd.setCursor(1,1);
  lcd.print("Carrera");
  delay(2000);
  pinMode(rojo1, OUTPUT);
  pinMode(rojo2, OUTPUT);
  pinMode(rojo3, OUTPUT);
  pinMode(verde, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(rele, OUTPUT);
}

void loop () {

  //*****Marquesina del
  Menu*****
  lcd.clear();
  lcd.setCursor(1,0);
  lcd.print("Seleccione numero de vueltas");
  delay(500);
  for (int positionCounter = 0; positionCounter < 13;
positionCounter++) {
    lcd.scrollDisplayLeft();
    delay(500);
  }
  for (int positionCounter = 0; positionCounter < 13;
positionCounter++) {
    lcd.scrollDisplayRight();
    delay(500);
  }
  delay(1000);

  //*****Boton para sumar*****
  while(ok < 1){
    val = digitalRead(boton1); // lee ell valor de la variable
    val
    if (val == HIGH) {          // condcion para los led
      lcd.clear();
      lcd.setCursor(1,1);
      lcd.print(vuelta);

    } else {
      vuelta = vuelta + 1;
      delay(100);
      lcd.clear();
      lcd.setCursor(1,1);
      lcd.print(vuelta);
    }
  }

  //*****Boton para restar*****
```

Control de un circuito de Slot con Arduino

```
val = digitalRead(boton2); // lee ell valor de la variable val
if (val == HIGH) {        // condicion para los led
    delay(50);
    lcd.clear();
    lcd.setCursor(1,1);
    lcd.print(vuelta);

} else {
    vuelta = vuelta - 1;
    delay(100);
    lcd.clear();
    lcd.setCursor(1,1);
    lcd.print(vuelta);
}

//*****Boton Ok*****

val = digitalRead(boton3); // lee ell valor de la variable val
if (val == HIGH) {        // condcion para los led
    delay(50);
    lcd.clear();

} else {
    ok = ok + 1;
    delay(50);
    lcd.clear();
}

}

//*****Activar
circuito*****
while(ok >= 1){
val = digitalRead(boton3); // lee ell valor de la variable val
    if (val == HIGH) {        // condcion para los led

    }else{
        lcd.clear();
        lcd.setCursor(1,0);
        lcd.print("Comenzando");
        lcd.setCursor(1,1);
        lcd.print("Carrera");
        delay(100);

        digitalWrite(rojo1, HIGH); //Encendido de los rojos
        tone(buzzer, doh, intervalo2); //Enciende el buzzer
        delay(intervalo);
        digitalWrite(rojo2, HIGH);
        tone(buzzer, doh, intervalo2);
        delay(intervalo);
        digitalWrite(rojo3, HIGH);
        tone(buzzer, doh, intervalo2);
        delay(intervalo);

        digitalWrite(rojo1, LOW); //Apaga los leds rojos
```

Control de un circuito de Slot con Arduino

```
digitalWrite(rojo2, LOW);
digitalWrite(rojo3, LOW);
digitalWrite(verde, HIGH); // Enciende el verde
digitalWrite(rele, HIGH); // Enciende el rele

lcd.clear();
lcd.setCursor(1,1);
lcd.print("Start!");

tone(buzzer, sol, intervalo*2);
delay(intervalo*2);
digitalWrite(verde, LOW); //Apaga verde
}

//*****Boton de
retroceso*****
val = digitalRead(boton4); // lee el valor de la variable
val
    if (val == HIGH) {          // condcion para los led
        delay(50);
        lcd.clear();

    } else {
        ok = ok - 1;
        delay(50);
        lcd.clear();
        vuelta = 0;

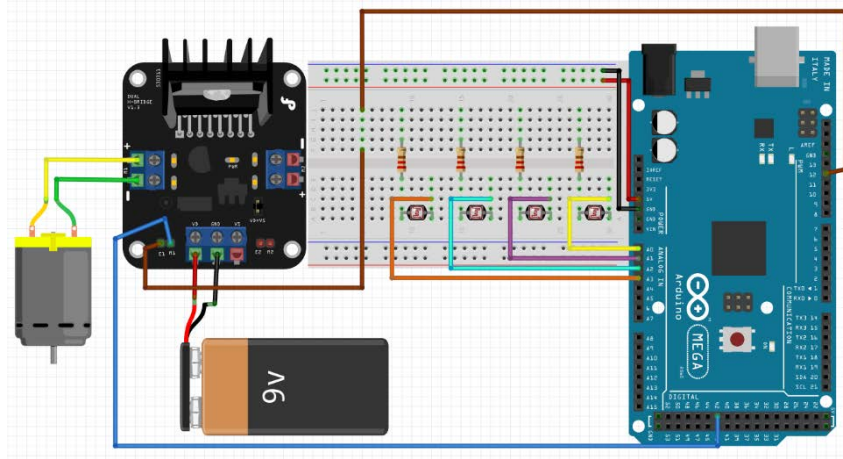
    }
//*****
*
}

}
```

Control de un circuito de Slot con Arduino

Módulo de control

Este módulo será el encargado de controlar la velocidad del coche en la pista. Para ello se usará un circuito como el siguiente



Al igual que en el circuito del semáforo, la pila deberá sustituirse por una fuente de alimentación de 16 voltios.

Como podemos apreciar en el diagrama, el modulo estará formado por cuatro fotorresistencias, que van situadas tanto a la entrada como a la salida de las curvas para tener un punto de referencia la placa de donde debe acelerar o aminorar, y un módulo de control de motores L298N, estando este configurado para funcionar con corrientes de más de 12 voltios mediante la eliminación del puente que lo controla, así como utilizando uno de los terminales para controlar el motor.

En este caso, el circuito varía un poco de la versión implementada, puesto que el diseño físico está hecho para ser conectado al puerto Jack de 3.5 milímetros que se usa para los mandos actuales.

```
int EN1 = 12;
int IN1 = 33;
int IN2 = 32;

int sensor1 = 1;
int sensor2 = 2;
int sensor3 = 3;
int sensor4 = 4;

boolean activo1 = true;
boolean activo2 = false;
```



```
void setup (){
  pinMode(EN1, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  Serial.begin(9600);
}

void loop() {

  if (analogRead(sensor1) < 15){
    activo1 == true;
    activo2 == false;

    if (activo1 = true){
      digitalWrite(IN1,LOW);
      digitalWrite(IN2,HIGH);
      analogWrite(EN1,255);
    }
  }

  if (analogRead(sensor2) < 15){
    activo2 == true;
    activo1 == false;
    if (activo2 = true){
      digitalWrite(IN1,LOW);
      digitalWrite(IN2,HIGH);
      analogWrite(EN1,55);
    }
  }

  if (analogRead(sensor3) < 15){
    activo1 == true;
    activo2 == false;

    if (activo1 = true){
      digitalWrite(IN1,LOW);
      digitalWrite(IN2,HIGH);
      analogWrite(EN1,255);
    }
  }

  if (analogRead(sensor4) < 15){
    activo1 == true;
    activo2 == false;

    if (activo2 = true){
      digitalWrite(IN1,LOW);
      digitalWrite(IN2,HIGH);
      analogWrite(EN1,55);
    }
  }
}
```

El código final, con todos los módulos interconectados será incluido en los anexos junto con la representación completa de los módulos interconectados.

CONCLUSION Y VALORACION FINAL

Este apartado será dividido en dos partes, puesto que al haber realizado el proyecto de manera conjunta entre dos alumnos, cada uno ha obtenido sus propias conclusiones y valoraciones, las cuales son:

José Luis Arroyo García

Para mí, este proyecto ha sido la ocasión perfecta para iniciarme en el mundo de la electrónica y la programación, en el cual tenía desde hace tiempo ganas de probar.

Siempre me ha llamado la atención la gran cantidad de cosas que se pueden hacer con las distintas placas que venden (la Raspberry y la Arduino...). En mi caso automatizar unos coches de slot ha sido un inicio perfecto en este mundillo.

Es un proyecto llamativo y con el que puedes lucirte con tus amigos, no mentiré no ha sido nada fácil crear este proyecto ya sea por el tiempo que han tardado las piezas en llegar tanto por las limitaciones a nivel de programación y electrónica que tengo pero he de decir que ha sido muy divertido, ir desarrollando este proyecto, a cada paso que iban saliendo las cosas.

Lo primero que hice fue el semáforo que no son más que 4 leds programados en un orden determinado y acompañadas de un zumbador sin embargo me invadía una sensación de euforia y felicidad cuando el código que escribía en el ordenador se volvía real .

Recuerdo en especial el día en que conseguimos que el coche arrancara solo ese día estábamos los dos saltando por las paredes y en ese momento no podíamos estar más contentos y eso aunque parezca una tontería es increíble aunque siempre había personas que te arrojaban un jarro de agua fría y te baja de las nubes decían que era una tontería que no nos daría tiempo que para que nos complicamos , aunque no nos desanimamos en ningún momento y no nos rendimos y hemos dado nuestro máximo esfuerzo para presentar este proyecto hasta el último día.

Igual para los que estáis leyendo este el proyecto no os parece gran cosa y pensáis que esto lo podéis hacer en unas horas pero para nosotros los cuales empezamos con muy pocos conocimientos de cómo hacerlo, llegar hasta este punto ha sido un largo camino muchas noches sin dormir y un sinfín de libros, manuales y videos pero ha merecido la pena, por lo menos para mí.

Una vez terminado si me preguntaran si pudiera elegir otro proyecto más sencillo de los que había, lo haría les respondería que no. Este es, en mi opinión, el mejor proyecto, es cierto que también es el más complicado y el que más recursos necesita, como la gran cantidad de piezas que hemos tenido que comprar.

Este proyecto me ha permitido conocer mejor a mi compañero y sinceramente no se me ocurre compañero mejor, ya que me ha aguantado todo este camino, y nuestro tutor Enrique Neble el cual nos ha ayudado muchísimo.

En conclusión, para mi es mucho lo que hemos logrado, que un coche inicie la carrera cuando el semáforo marca la salida, que pueda contar las vueltas y los tiempos y conducirse solo era algo que hubiera considerado una locura el día que empecé este curso sin embargo ahora mi opinión ha cambiado y solo puedo pensar que cual será el próximo proyecto en cual me embarque.

Alejandro Domínguez Al-lal

En mi caso, este proyecto me ha servido para retomar programación, ya que desde que deje la carrera no había vuelto a programar. Una cosa que ha tenido buena en este sentido, es que el programar para Arduino me ha ayudado a asentar conocimientos que, en su día, o se me olvidaron, o no pude asimilar, fuese por el motivo que fuere.

Otra de las cosas que me ayudo a retomar fue mi afición por el Slot. Es verdad que en un principio, no sabía que proyecto escoger, o pensar, ya que tampoco se me ocurrían ideas, teniendo en cuenta la prohibición de nuestro profesor Francisco Sáez de ponernos a programar, a sabiendas de lo que pasó en años anteriores. Fue gracias a una charla con el profesor Enrique Neble, también aficionado a este hobby, el que eligiese como proyecto la “automatización” de un circuito usando una placa Arduino.

Aunque al empezar el proyecto, era un mundo nuevo, pronto me di cuenta que no difería tanto de los conocimientos que tenía sobre el lenguaje C, sobre el que se programa la Arduino. Así, tras preguntarle a mi compañero si quería unirse al proyecto, pedimos unos kits básicos, de esos que traen varios módulos para hacer vario proyectos, y empezamos a trastear. Aunque parezca una tontería, es bastante gratificante cuando veíamos por primera vez que, tras incorporar la pantalla y una fotorresistencia, al pasar el coche por encima, la Arduino contaba las vueltas.

A partir de ahí, empezamos a programar los siguientes módulos del proyecto, tales como el cronometro, el cual tocó reprogramar entero desde cero, ya que no se dejaba integrar y “colgaba la Arduino” (hasta que nos dimos cuenta que el error estaba en el uso de la función “delay” en vez de “millis”, que sirve mejor para estos menesteres).

Aunque el proyecto no está exento de complicaciones, como el hecho de que, al pedir la segunda pantalla (una más grande y completa que la que veníamos usando del kit), no contáramos con el hecho de que las librerías no estuviesen correctas, o que la idea

de grabar una partida y reproducirla más tarde a través de los datos guardados en un fichero no pudiéramos llevarla a cabo, precisamente por problemas a la hora de guardar los datos.

Es verdad que eso puede provocar grandes decepciones, y ganas de jugar al baloncesto con las piezas cuando no funciona algo que crees que está bien hecho.

La verdad es que según se va comenzando, e introduciendo en el mundillo de la Arduino, se empiezan a descubrir cosas que se pueden hacer con las piezas, incluso cuando vas por la calle, como la idea de hacerme un termostato casero para el ordenador en verano, tras ver como saltaba el radiador del coche para bajar la temperatura al motor por exceso de temperatura.

El trabajo con mi compañero ha sido divertido, sobre todo en algunos momentos, como cuando descubrió cómo funcionan las fotorresistencias, aunque a él se le resista su uso.

También ha sido bastante entretenido ver a nuestro tutor de proyecto, Enrique, haciendo algo distinto a scripts de Linux, o administración de sistemas, y viendo cómo, incluso jugando a cambiar la velocidad de un ventilador a través de la Arduino, demostraba una capacidad para la programación que “ya la quisiera yo para mí”.

En definitiva, creo que el proyecto me ha cambiado un poco, sobre todo a la hora de programar, que ya me estructuro algo mejor, y me ha ayudado a tener más paciencia, puesto que el hecho de estar varias horas con el mismo modulo del proyecto, a ver por qué no funcionaba, hasta averiguar que el problema era que el led estaba puesto al revés, o que un bucle dentro del programa provocaba que otro dejase de funcionar, requiere de grandes dosis de esta, para no dejar el proyecto y dedicarte a otra cosa.

Ahora, toca esperar a ver cuál va a ser la siguiente idea para desarrollar usando esta placa.

BIBLIOGRAFIA Y FUENTES DE CONSULTA

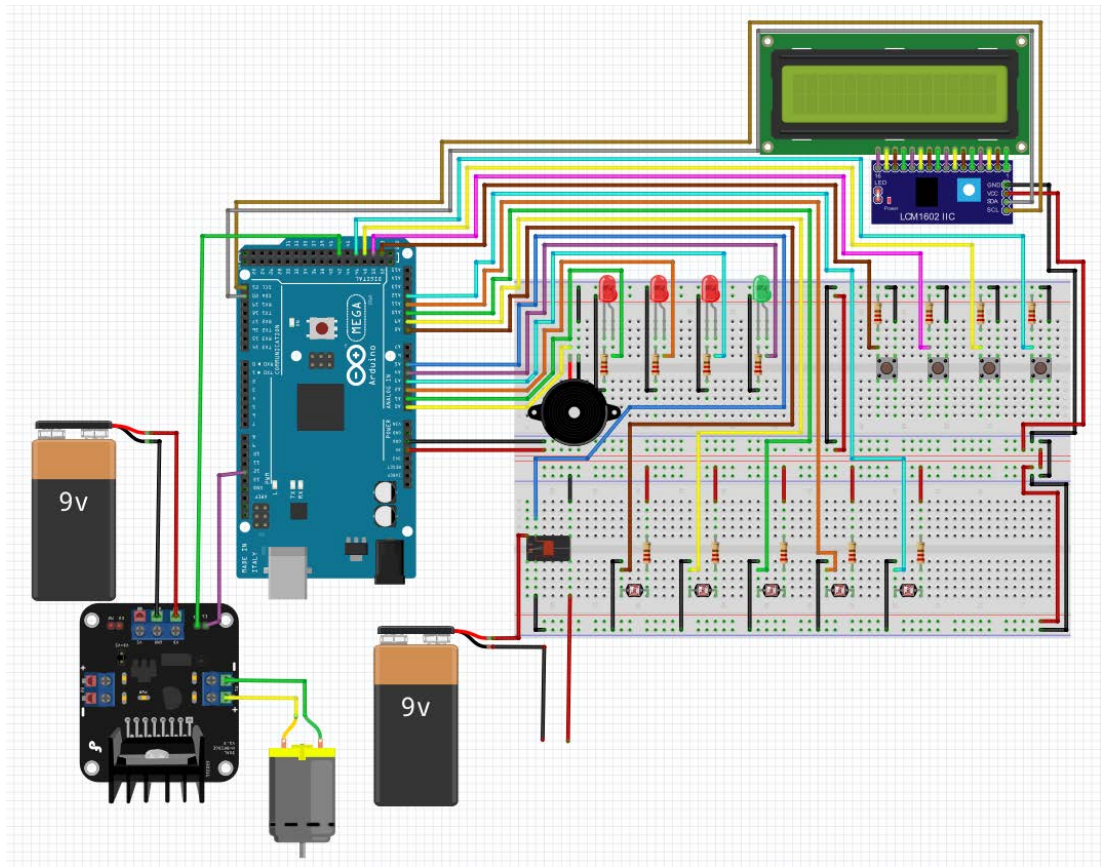
- Foro oficial de Arduino para el uso de diversos componentes, tales como relés, o módulos de control de voltaje.
- Libro Arduino: Curso práctico de formación, de la editorial Alfaomega-RClibros.
- Página web <http://www.rcnewz.com/slottimer.ino> , para el desarrollo del cronometro.
- Página web <http://www.ardumania.es/ejercicio-4-ldr/> , para el uso de fotorresistencias.
- Página web <https://codigofacilito.com/cursos/Arduino> para la obtención de conocimientos generales sobre la programación y el uso de algunos componentes electrónicos.
- Paginas varias de internet y videos para el uso de algunos componentes no explicados o mal explicados en los foros de Arduino.

Fecha de finalización:

31/05/2015

Anexos

Esquema del cableado y componentes del circuito completo



Código fuente del proyecto, ya con todos los módulos integrados (351 líneas)

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

// Variables del menu

int boton1 = 39;
int boton2 = 37;
int boton3 = 41;
int boton4 = 35;
int nvuelta = 0;
int ok = 0;
int val = 0; // variable que da el estado

// Variables del semaforo

int rojo1 = 52; //Pin led rojo 1
int rojo2 = 53; //Pin led rojo 2
int rojo3 = 50; //Pin led rojo 3
```

```
int verde = 51; //Pin led verde
int buzzer = 47; //Pin del buzzer
int doh=261.63;
int sol=392;
int intervalo = 1000; //Intervalo de la secuencia
int intervalo2 = 500; //Intervalo de la secuencia
int rele = 42;
boolean inicio = false;

// Variables del cronometro-cuentavueltas

const byte lightPin = 0;
int lightSensor = 0;
unsigned long lapMillis = 0;
unsigned long startMillis = millis();
unsigned long bestMillis = 9999999;
float lapTime = 0.00;
float bestLap = 0.00;
boolean firstTrigger = true;
boolean newBest = false;
int vuelta = 0;

// Variables del reproductor

int EN1 = 12;
int IN1 = 33;
int IN2 = 32;

int sensor1 = 1;
int sensor2 = 2;
int sensor3 = 3;
int sensor4 = 4;

boolean activo1 = true;
boolean activo2 = false;

void setup() {
  lcd.backlight();
  lcd.begin();

  lcd.print("Gestor de");
  lcd.setCursor(1,1);
  lcd.print("Carrera");
  delay(2000);

  pinMode(rojo1, OUTPUT);
  pinMode(rojo2, OUTPUT);
```



```
pinMode(rojo3, OUTPUT);
pinMode(verde, OUTPUT);
pinMode(buzzer, OUTPUT);
pinMode(rele, OUTPUT);

pinMode(EN1, OUTPUT);
pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);

Serial.begin(9600);
}

void loop () {
// ***** MENU *****
//Marquesina del Menu

  lcd.clear();
  lcd.setCursor(1,0);
  lcd.print("Seleccione numero de vueltas");
  delay(500);

  for (int positionCounter = 0; positionCounter < 13; positionCounter++) {
    lcd.scrollDisplayLeft();
    delay(500);
  }

  for (int positionCounter = 0; positionCounter < 13; positionCounter++) {
    lcd.scrollDisplayRight();
    delay(500);
  }

  delay(1000);

  //Boton para sumar

  while(ok < 1){
    val = digitalRead(boton1); // lee el valor de la variable val
    if (val == HIGH) {      // condicion para los led
      lcd.clear();
      lcd.setCursor(1,1);
      lcd.print(nvuelta);

    } else {
      nvuelta = nvuelta + 1;
      delay(100);
      lcd.clear();
      lcd.setCursor(1,1);
      lcd.print(nvuelta);
    }
  }
}
```

```
}

//Boton para restar

val = digitalRead(boton2); // lee ell valor de la variable val
if (val == HIGH) {        // condicion para los led
  delay(50);
  lcd.clear();
  lcd.setCursor(1,1);
  lcd.print(nvuelta);
} else {
  nvuelta = nvuelta - 1;
  delay(100);
  lcd.clear();
  lcd.setCursor(1,1);
  lcd.print(nvuelta);
}

//Boton Ok

val = digitalRead(boton3); // lee ell valor de la variable val
if (val == HIGH) {        // condcion para los led
  delay(50);
  lcd.clear();
} else {
  ok = ok + 1;
  delay(50);
  lcd.clear();
}
}

//Activar circuito

while(ok >= 1){

  val = digitalRead(boton3);    // lee ell valor de la variable val

  if (val == HIGH) {           // condcion para los led

  }else{
    lcd.clear();
    lcd.setCursor(1,0);
    lcd.print("Comenzando");
    lcd.setCursor(1,1);
    lcd.print("Carrera");
    delay(100);
```

```
digitalWrite(rojo1, HIGH);    //Encendido de los rojos
tone(buzzer, doh, intervalo2); //Enciende el buzzer
delay(intervalo);

digitalWrite(rojo2, HIGH);
tone(buzzer, doh, intervalo2);
delay(intervalo);

digitalWrite(rojo3, HIGH);
tone(buzzer, doh, intervalo2);
delay(intervalo);

digitalWrite(rojo1, LOW);    //Apaga los leds rojos
digitalWrite(rojo2, LOW);
digitalWrite(rojo3, LOW);
digitalWrite(verde, HIGH);    // Enciende el verde
digitalWrite(rele, HIGH);    // Enciende el rele

lcd.clear();
lcd.setCursor(1,1);
lcd.print("Start!");

tone(buzzer, sol, intervalo*2);
delay(intervalo*2);
digitalWrite(verde, LOW); //Apaga verde

inicio = true;

}

//Boton de atras

val = digitalRead(boton4); // lee ell valor de la variable val
if (val == HIGH) {        // condcion para los led
  delay(50);
  lcd.clear();

} else {
  ok = ok - 1;
  delay(50);
  lcd.clear();
  nvuelta = 0;

}

//*****

// ***** Crono - Cuentavueltas *****
```

```
if (inicio = true) {

    // Ultima vuelta

    lcd.setCursor(0, 0);
    lcd.print("Last: ");
    lcd.print(lapTime);
    lcd.print(" ");

    // titulo del cuentavueltas en la linea superior y numero de vuelta en la linea
inferior

    lcd.setCursor(13,0);
    lcd.print("Lap");

    lcd.setCursor(14,1);
    lcd.print(vuelta);

    // Mejor Vuelta

    lcd.setCursor(0, 1);
    lcd.print("Best: ");
    lcd.print(bestLap);
    lcd.print(" ");
    newBest = true;

//Si detecta el coche
if(analogRead(lightSensor) < 15){

    //Determina el tiempo de la vuelta en milisegundos
    lapMillis = millis() - startMillis;
    startMillis = millis();
    lapTime = lapMillis / 1000.00;

    //Si es el primer paso, imprime "race started"
    if (firstTrigger == true){

        lcd.setCursor(0, 0);
        lcd.print("Race Started! ");

    }

//imprime si no es el primer paso

    if (firstTrigger != true){

        Serial.print("lapTime: ");
        Serial.println(lapTime);
```

```

        Serial.print("lapMillis: ");
        Serial.println(lapMillis);

// Imprime el tiempo de la vuelta
    lcd.setCursor(0, 0);
    lcd.print("Last: ");
    lcd.print(lapTime);
    lcd.print(" ");
    lcd.setCursor(13,0);
    lcd.print("Lap");
    lcd.setCursor(14,1);
    lcd.print(vuelta);
    vuelta= vuelta+1;
}

// Si la ultima vuelta es mejor que la mejor vuelta guardada

    if(lapMillis < bestMillis && firstTrigger != true){

        bestMillis = lapMillis;
        bestLap = lapTime;
        Serial.print("bestLap: ");
        Serial.println(bestLap);

// Imprime el mejor tiempo
        lcd.setCursor(0, 1);
        lcd.print("Best: ");
        lcd.print(bestLap);
        lcd.print(" ");
        newBest = true;
    }

    firstTrigger = false;
    Serial.println("-----");
}

}

//*****

// ***** Reproductor *****

if (inicio = true){
if (analogRead(sensor1) < 15){
    activo1 == true;
    activo2 == false;

    if (activo1 = true){
        digitalWrite(IN1,LOW);
        digitalWrite(IN2,HIGH);
        analogWrite(EN1,255);
    }
}
}

```

```
}

if (analogRead(sensor2) < 15){
  activo2 == true;
  activo1 == false;
  if (activo2 = true){
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,HIGH);
    analogWrite(EN1,55);
  }
}

if (analogRead(sensor3) < 15){
  activo1 == true;
  activo2 == false;

  if (activo1 = true){
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,HIGH);
    analogWrite(EN1,255);
  }
}

if (analogRead(sensor4) < 15){
  activo1 == true;
  activo2 == false;

  if (activo2 = true){
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,HIGH);
    analogWrite(EN1,55);
  }
}
}
if (vuelta == nvuelta){
  digitalWrite(rele, LOW);
}
}
}
```