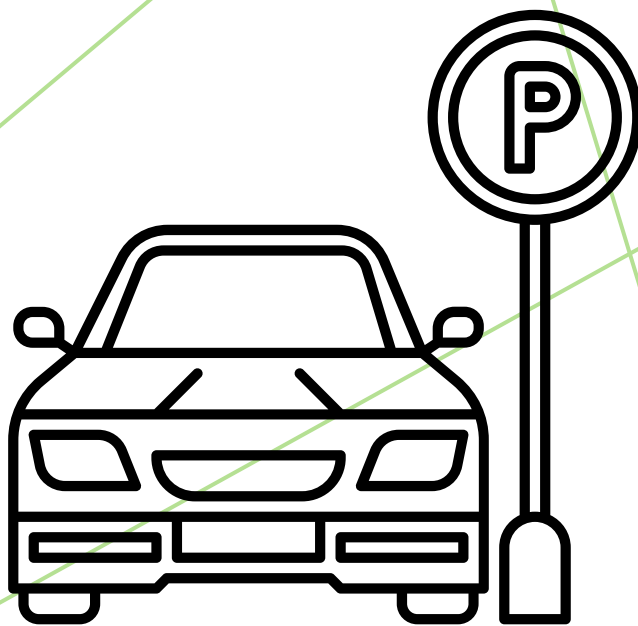


UNIVERSIDAD DE EXTREMADURA

# ***CAR PARKING SYSTEM***

## **ARDUINO UNO**



SISTEMAS DIGITALES

INGENIERÍA DE SONIDO E IMAGEN EN TELECOMUNICACIONES

LAURA ROSA GIL

# ÍNDICE

• 1. Motivación.....	2
• 2. Estado del arte.....	2
• 3. Técnicas y materiales utilizados.....	2
• 4. Esquemas, gráficos, ficheros o códigos.....	4
• 5. Resultado obtenido.....	6
• 6. Futuros desarrollos posibles.....	7
• 7. Conclusión.....	7

## 1. Motivación.

Para la realización de este miniproyecto con Arduino el profesor dio libertad de elección para el tema. Con motivo de algunos fallos recientes en la barrera de entrada al parking de la Escuela Politécnica de Cáceres, se planteó el hecho de recrear el funcionamiento de ésta y añadir alguna innovación, como es el acceso por sensor de proximidad.

Hablamos entonces de un proyecto basado en la implementación de una barrera de entrada/salida en un parking acompañado de una pantalla que muestra el espacio disponible en éste y, en el caso de que estuviese completo, un mensaje que avisase de que no hay espacio para aparcar.


## 2. Estado del arte.




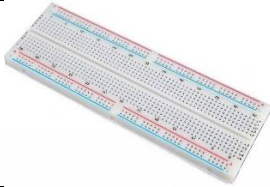


Antes de comenzar, se utilizaron como base diversos tutoriales y artículos de internet que permiten conocer el funcionamiento de los materiales necesarios y también algunos ejemplos de otros tipos de parkings ya existentes.

- <https://programarfacil.com/blog/arduino-blog/servomotor-con-arduino/>
- <https://www.luisllamas.es/detectar-obstaculos-con-sensor-infrarrojo-y-arduino/>
- <https://programarfacil.com/blog/arduino-blog/texto-en-movimiento-en-un-lcd-con-arduino/>
- [http://www.practicasconarduino.com/manualrapido/proyecto\\_parking\\_con\\_barrera.html](http://www.practicasconarduino.com/manualrapido/proyecto_parking_con_barrera.html)
- <https://create.arduino.cc/projecthub/Techatronic/automatic-car-parking-system-project-using-arduino-ba2cb8>
- <https://create.arduino.cc/projecthub/embeddedlab786/automatic-car-parking-system-95a9dc>
- <https://create.arduino.cc/projecthub/projects/tags/parking>

## 3. Técnicas y materiales utilizados.

Los materiales utilizados en el proyecto fueron:

Arduino UNO	
-------------	--

Dos sensores de obstáculos IR	
SG90 Micro-servomotor	
Cables de puente	
Protoboard	
Módulo de pantalla LCD Arduino DFRobot I2C 16x2	
Cable USB para cargar el código	

Al comenzar a desarrollarlo, en primer lugar, se planteó un código básico que permitiese la organización del flujo de trabajo de Arduino. Declarando así las variables más relevantes, el tratamiento de los sensores, las condiciones del servomotor y de la pantalla LCD.

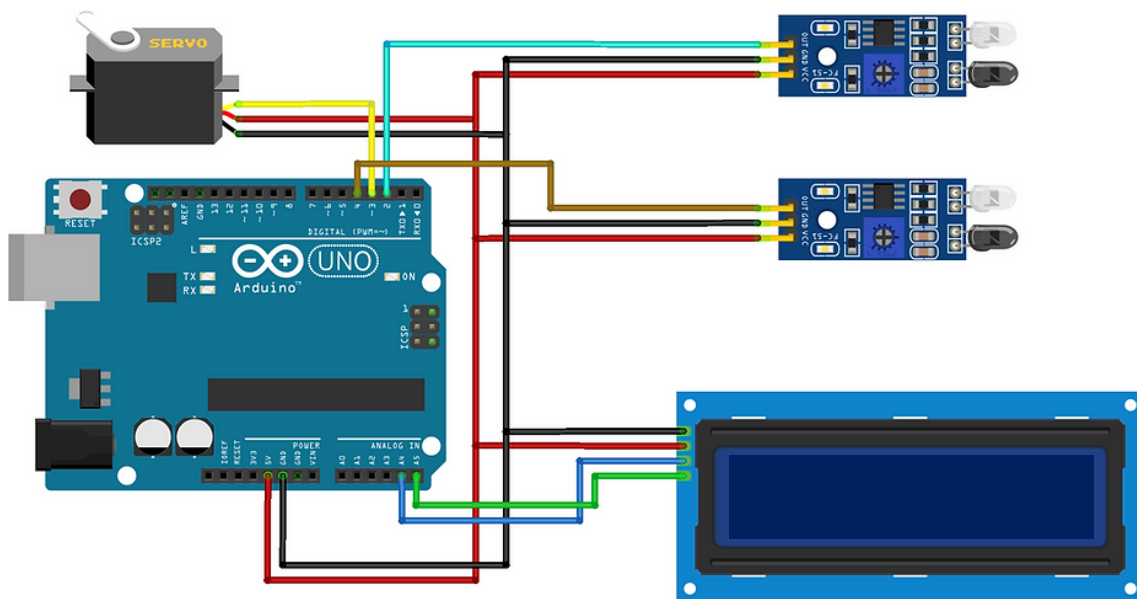
Además del código principal del proyecto, se crearon tres códigos distintos que permitían comprobar el estado y funcionamiento de los principales componentes otorgados por el profesor y pertenecientes al laboratorio. Uno de esos códigos de prueba era para los sensores de proximidad IR, otro para el Servomotor y finalmente uno para la pantalla LCD.

Se llevó a cabo el montaje del proyecto sobre la protoboard y se comprobaron uno a uno los componentes hardware instalados a través de los códigos mencionados anteriormente. Una vez verificado el correcto funcionamiento de los distintos materiales, se probó con el código general para ver si todo el conjunto cumplía con su función. Luego, se añadieron retoques al código para cuadrarlo definitivamente con las funciones.

Finalmente, una vez ultimado tanto el código como el conexionado de los componentes externos, se ha realizado un montaje sobre una tabla decorada que asemeja al parking para poder exponerlo y mostrar su funcionamiento.

#### 4. Esquemas, gráficos, ficheros o códigos.

El esquema que se ha seguido para las conexiones en Arduino UNO ha sido el siguiente.



En cuanto al código:

```
// Arduino Car Parking System Definitivo

/*Sistema de aparcamiento para coches basado en la utilización de sensores de proximidad
infrarrojos para la detección de los vehiculos y la utilización de un servomotor que
levante la valla cuando el vehículo vaya a entrar al parking y la cierre tras él cuando
este haya entrado.

*Autora: Laura Rosa Gil

*Grado en Ingenieria de Sonido e Imagen en Telecomunicaciones (Escuela Politécnica de
Cáceres)

*/

//Incluimos las librerías que vamos a usar

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);
```

```

#include <Servo.h>

Servo myservol;

int IR1 = 2;    //Asignamos las entradas y salidas de la placa

int IR2 = 4;

int Slot = 5;   //Número total de aparcamientos

int flag1 = 0; //Banderas que controlan los sensores

int flag2 = 0;

void setup() {

    lcd.init();

    lcd.backlight();

    pinMode(IR1, INPUT);

    pinMode(IR2, INPUT);    //Asignamos IR1 e IR2 como entrada en la placa

    myservol.attach(9);

    myservol.write(90);    //Primera posición (ángulo) que adopta el servomotor

    //Inicialización de la pantalla LCD

    lcd.setCursor (0,0);

    lcd.print("    ARDUINO    ");

    lcd.setCursor (0,1);

    lcd.print(" PARKING SYSTEM ");

    delay (2000);

    lcd.clear();

}

void loop(){

if(digitalRead (IR1) == LOW && flag1==0){    //Controla el movimiento por el sensor 1

    if(Slot>0){flag1=1;

        if(flag2==0){myservol.write(180); Slot = Slot-1;}

    } //Levanta la valla para dejar pasar al vehículo y avisa que hay un aparcamiento
    menos disponible

    else{ //Cuando el parking está completo, avisa en la pantalla LCD

```

```

    lcd.setCursor (0,0);

    lcd.print("    SORRY :(    ");

    lcd.setCursor (0,1);

    lcd.print("  Parking Full  ");

    delay (3000);

    lcd.clear();

}

}

if(digitalRead (IR2) == LOW && flag2==0){flag2=1;    //Controla el movimiento por el
sensor 2

    if(flag1==0){myservo1.write(180); Slot = Slot+1;}

}    //Levanta la valla para dejar pasar al vehículo y avisa que hay un aparcamiento
más disponible

if(flag1==1 && flag2==1){ //Comprueba si ambos sensores se han activado

    delay (1000);

    myservo1.write(100); //Baja la valla porque el vehículo ya habrá entrado o salido

    flag1=0, flag2=0;

}

lcd.setCursor (0,0);    //Configuración del mensaje de la pantalla

lcd.print("    WELCOME!    ");

lcd.setCursor (0,1);

lcd.print("Slot Left: ");

lcd.print(Slot);

}

```

## 5. Resultado obtenido.

El resultado de aplicar este código a la placa Arduino ha sido el correcto funcionamiento de la entrada al parking que se buscaba. Se consigue la detección por proximidad de los vehículos gracias a los sensores IR junto con el movimiento de la valla hacia arriba para permitir el paso de este y, después, su movimiento hacia abajo tras él, gracias al servomotor.

Todo ello está acompañado de un contador que controla el espacio libre que existe en el interior del parking y que avisa a través de la pantalla LCD de cómo se encuentra; si ocupado o, en el caso de que esté libre, muestra el número de

aparcamientos disponibles. En el caso de que esté ocupado, el motor no levantará la valla por no haber espacio dentro del parking, impidiendo la entrada de más vehículos.

## **6. Futuros desarrollos posibles.**

Este sistema de aparcamiento es un proyecto que, aunque ya existe, siempre se puede mejorar y adecuar para que los usuarios tengan un acceso más sencillo y rápido a los diferentes parkings.

Se trata de un sistema que podría ser de gran interés en casos como la barrera de parking de la Escuela Politécnica de Cáceres, donde se avisaría a través de un mensaje por pantalla si no hubiese espacio disponible. También es de interés la activación de la barrera por sensor ya que el uso del pulsador existente actualmente ha ocasionado problemas en varias ocasiones.

Una de las muchas mejoras que podrían incluirse sería, por ejemplo el añadido de sensores LED en cada aparcamiento para registrar cuál está ocupado y cuál está libre.

## **7. Conclusión.**

Con la realización de este proyecto, se ha podido poner en práctica todo lo aprendido en esta parte de la asignatura, así como hacer uso de conocimientos obtenidos en otras asignaturas. También, se ha comprendido la capacidad de los microcontroladores para llevar a cabo tareas para lo que son programados expresamente.

Por otro lado, me he defendido muy bien con el lenguaje Arduino, que era la primera vez que veía, pero que ha resultado ser muy intuitivo y con similitudes a otros lenguajes ya vistos en otras asignaturas como C++. Además, Arduino cuenta con una comunidad donde se pueden encontrar gran cantidad de códigos y librerías que sirven de ayuda y reducen el tiempo de trabajo enormemente.

Para finalizar, he de decir que, aunque los conocimientos teóricos son imprescindibles para comprender la asignatura, resulta de gran ayuda poder ponerlos en práctica mientras se estudian. De esta manera, hemos podido enfrentarnos a los problemas que se pueden generar a lo largo de la realización del proyecto y tras solucionarlos, se entiende mejor la base de lo que se está aprendiendo.