



Universidad de Murcia

Facultad de Informática

Programación de Sistemas Embebidos en Red

Proyecto Final - Arduparking

Alumnos

Daniel Ruiz Villa

daniel.ruiz7@um.es

Pedro José Fernández Campillo

pedrojose.fernandez@um.es

Edison Omar Loza Guzmán

edisonomar.loza@um.es

Profesor

Benito Úbeda Miñarro

bubeda@um.es

Índice

1. ¿Qué es Arduparking?	3
2. Elementos y material	3
2.1. Display LCD	3
2.2. Sensor de luminosidad LDR	4
2.3. Sensor de humedad DHT11	4
2.4. Servomotores	4
2.5. Sensor de ultrasonidos HC-SR04	5
2.6. LED RGB	6
2.7. Pulsador	6
2.8. Sensor MiFare RC522	6
3. Funcionamiento y montaje	7
4. Código	14
5. Problemas encontrados	18
6. Conclusiones	18

1. ¿Qué es Arduparking?

El planteamiento de nuestro proyecto final, Arduparking, consiste en la sala de control de un parking, donde varios coches pueden acceder a él, podemos comprobar los valores de temperatura, luminosidad y humedad del mismo. Para el desarrollo de este proyecto nos hemos basado en conocimientos aprendidos durante los 3 meses y medio de prácticas de la asignatura. A lo largo de este documento explicaremos los elementos de los que está compuesto nuestro proyecto y su funcionamiento en profundidad, tanto en forma de código como adjuntando imágenes para que quede más visual.

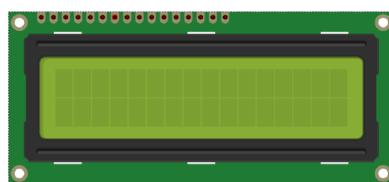


2. Elementos y material

Para este proyecto hemos utilizado muchos componentes típicos entre los accesorios de Arduino UNO y de otros microcontroladores. Vamos a nombrarlos, incluir una imagen aclaratoria de cada uno de ellos para una mejor identificación, y explicar de forma breve su funcionalidad en nuestro proyecto. Vamos a obviar dos de ellos, como son el propio microcontrolador Arduino y la placa de conexión.

2.1. Display LCD

Este display LCD, de tamaño 16x2, nos proporciona una pantalla de dimensiones reducidas donde poder mostrar salidas de nuestro programa. En nuestro caso vamos a utilizar dos de ellas. Al tener tres parámetros de medición en nuestro parking, como son la temperatura, humedad y luminosidad, en una pantalla un valor estará fijo y en la otra rotarán temporalmente los restantes.



2.2. Sensor de luminosidad LDR

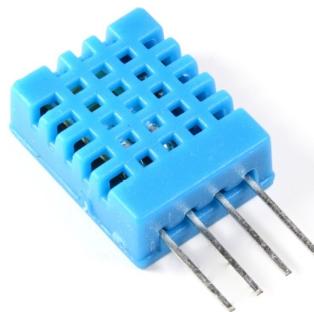
Este fotoresistor ya lo hemos utilizado, al igual que el display, en las prácticas de la asignatura. Su resistencia varía en función de la luz recibida, y podemos usar esa variación para medir a través de las entradas analógicas el nivel de luminosidad de nuestro parking.



2.3. Sensor de humedad DHT11

Este sensor con procesador interno proporciona la medición de temperatura y humedad mediante una señal digital. Vamos a aprovechar este único dispositivo para obtener los dos parámetros. Es capaz de medir la humedad de un 20 % a un 80 %, con una precisión del 5 %.

Por otro lado, su frecuencia de muestreo es de 1Hz, es decir, podemos ver como máximo 1 muestra de humedad por segundo en nuestro display, lo cual es suficiente, dado que la humedad únicamente variará cuando manualmente hidratemos el entorno.



2.4. Servomotores

Las puertas de nuestro parking estarán simuladas por dos servomotores, que harán las veces de entrada y salida del mismo. Un servomotor, comúnmente llamado como servo, es un accionador al cual se le asignan unos grados y el mecanismo se coloca en estos. No son capaces de hacer un movimiento completo, por lo general, siendo posible unos 180º como máximo.

Internamente consta de un mecanismo reductor, y cuenta con un alto grado de precisión. Por contra, es cierto que las velocidades de movimiento no son muy elevadas, o al menos no tan altas como las de los motores de corriente continua. Aunque admiten valores de entre 4,8 y 7,2 V, el más adecuado suele ser en torno a los 6V. La tensión elegida es muy importante para la fuerza, velocidad y precisión del servo.



2.5. Sensor de ultrasonidos HC-SR04

Un sensor de ultrasonidos nos sirve para medir distancias. Su funcionamiento se basa en el envío de un pulso de alta frecuencia, el cual rebota en el objeto deseado y es reflejado hacia el sensor, que dispone de un micrófono para esa frecuencia. Al medir el tiempo entre pulsos y conociendo la velocidad del sonido, se estima la distancia.

Aunque son baratos y fáciles de usar, tienen baja precisión, dado que la orientación de la superficie a medir puede falsear la medición. Tampoco son recomendados para usarse al aire libre.

En cualquier caso, es capaz de medir desde los 20cm hasta los 2 metros. Y estos valores entran dentro de lo que buscamos para nuestro parking, donde los usaremos para ver si una plaza de aparcamiento está ocupada o no por un vehículo, tal como indicará el led correspondiente. En nuestro caso incorporaremos dos LED de este tipo, dado que tendremos dos plazas de aparcamiento.



2.6. LED RGB

Un LED RGB se trata de un LED que contiene tres emisores de luz en su interior. El objetivo es poder crear toda la gama de colores posibles mezclando intensidades y tonalidades distintas. Cuanto los tres son combinados a la misma intensidad se obtiene el blanco.



2.7. Pulsador

Los pulsadores en los microcontroladores son esenciales, ya que nos permiten interactuar en el momento deseado con nuestro sistema, pudiendo dotar a la pulsación del efecto que queramos. El pulsador tiene cuatro patillas que están conectadas a pares, como vemos en la fotografía inferior. Su funcionamiento es muy sencillo, ya que cuando pulsamos se cierra el circuito y dejamos pasar la corriente.



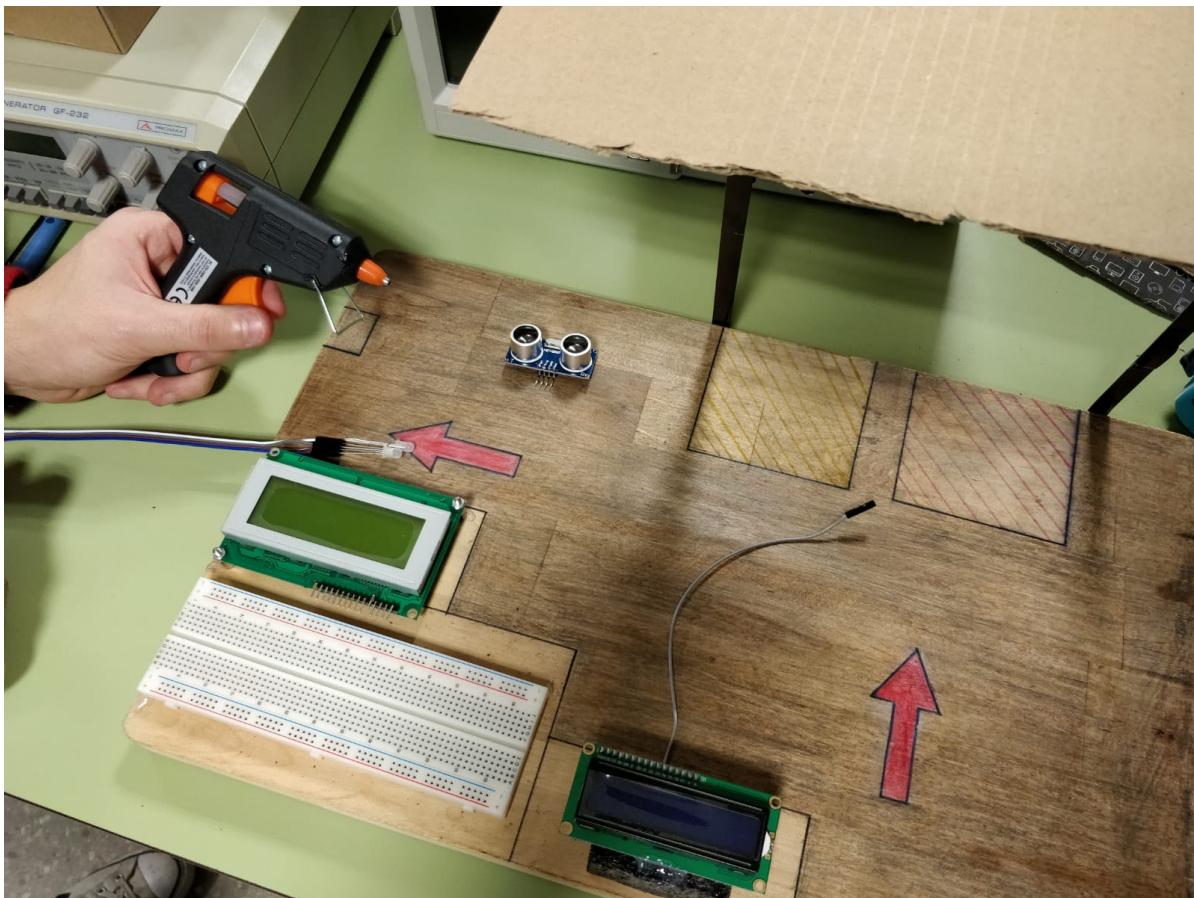
2.8. Sensor MiFare RC522

Este sensor de identificador por radiofrecuencia, RFID, nos va a permitir leer la tarjeta con la que vamos a acceder al parking en la entrada.

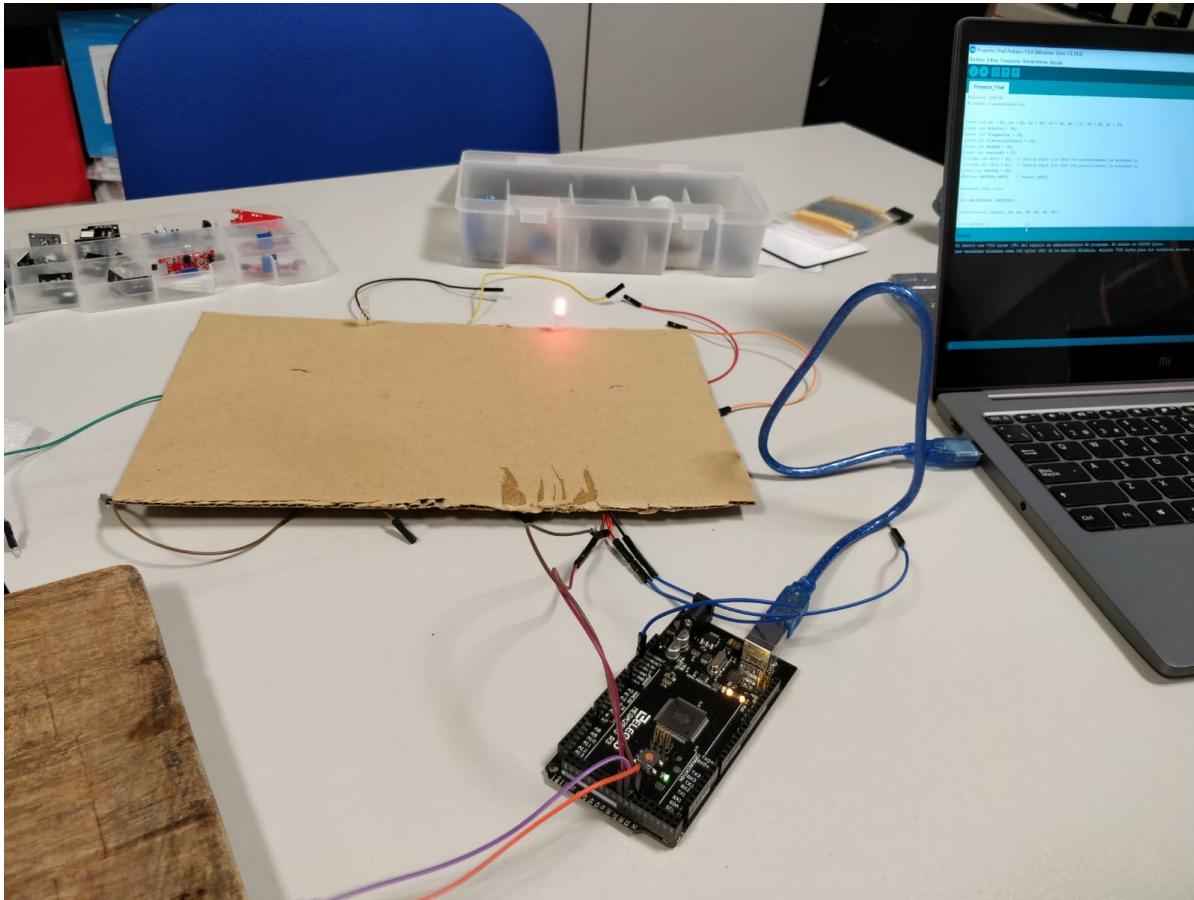


3. Funcionamiento y montaje

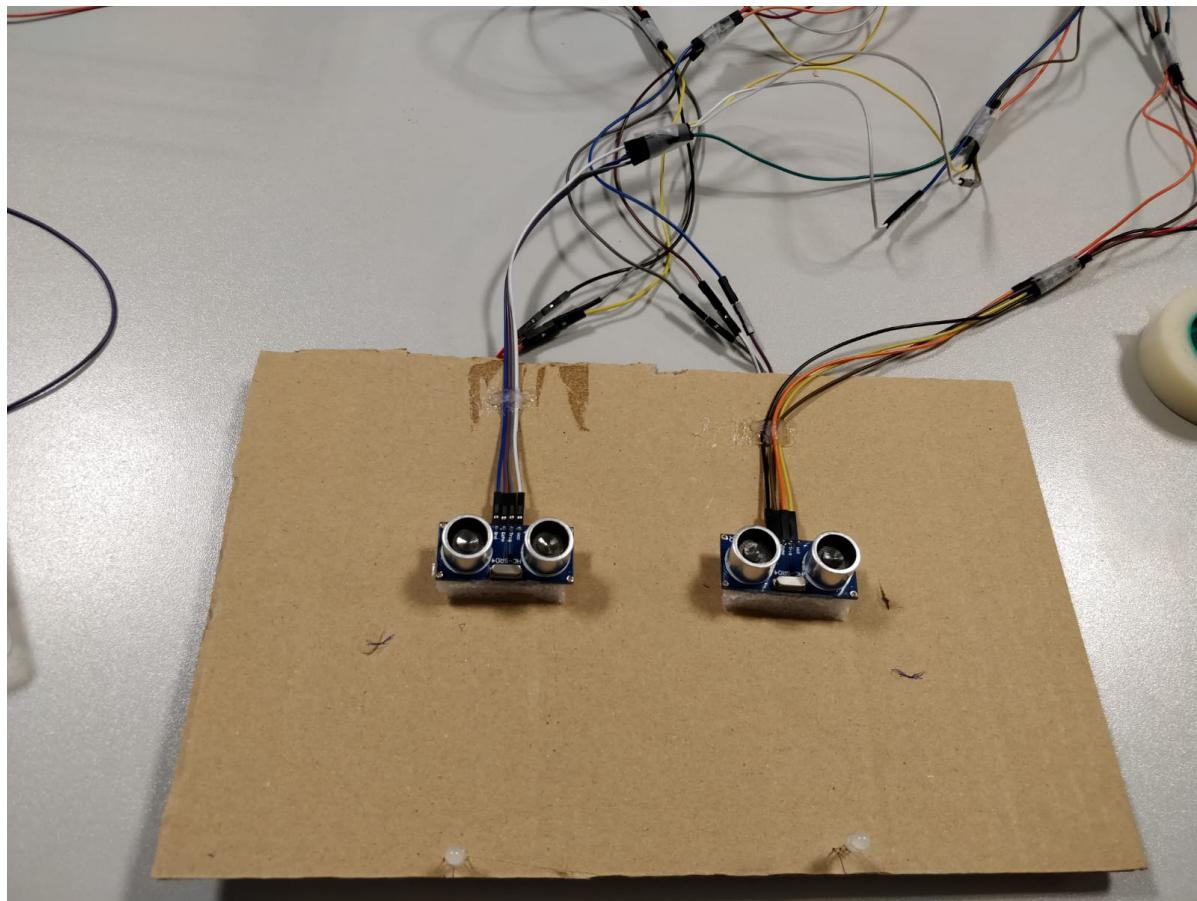
Como hemos explicado, Arduparking consta de dos plazas de aparcamiento las cuales pueden estar en dos estados, ocupada o libre. Nuestro sensor de ultrasonidos se encarga de comprobar si existe un obstáculo a menos de la distancia para la cual está calibrado, es decir, el "suelo" del parking. En caso afirmativo, el LED RGB pasa de estar verde a rojo, indicando que la plaza se encuentra ocupada. Mientras tanto, llevamos un control de la temperatura, humedad y luminosidad del parking en tiempo real. En sí mismo el funcionamiento tiene poco que explicar y es sencillo, por lo que vamos a ilustrar esta sección con fotos del montaje.



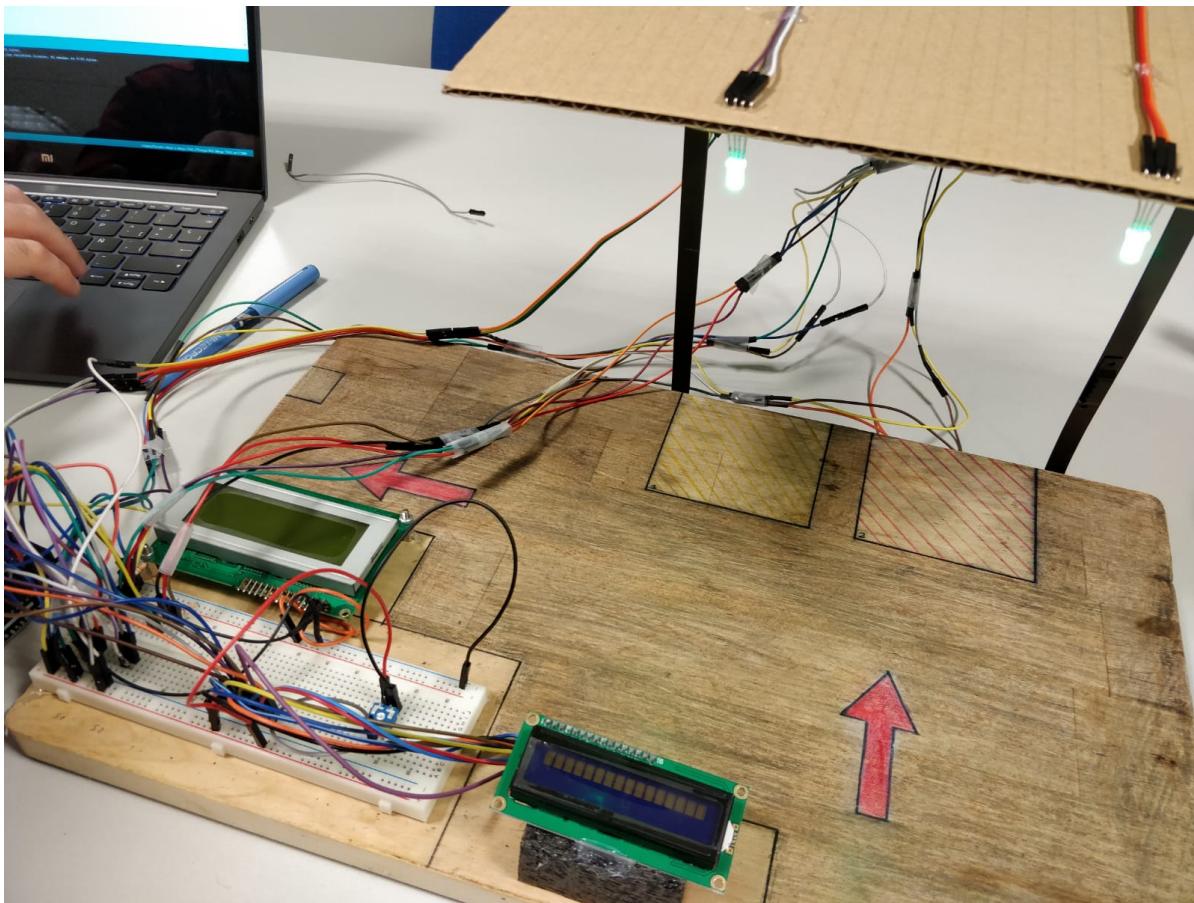
Aquí vemos como estamos pegando con una pistola termofusible los primeros elementos a la tabla del parking. En este caso hemos colocado la protoboard y los dos displays.



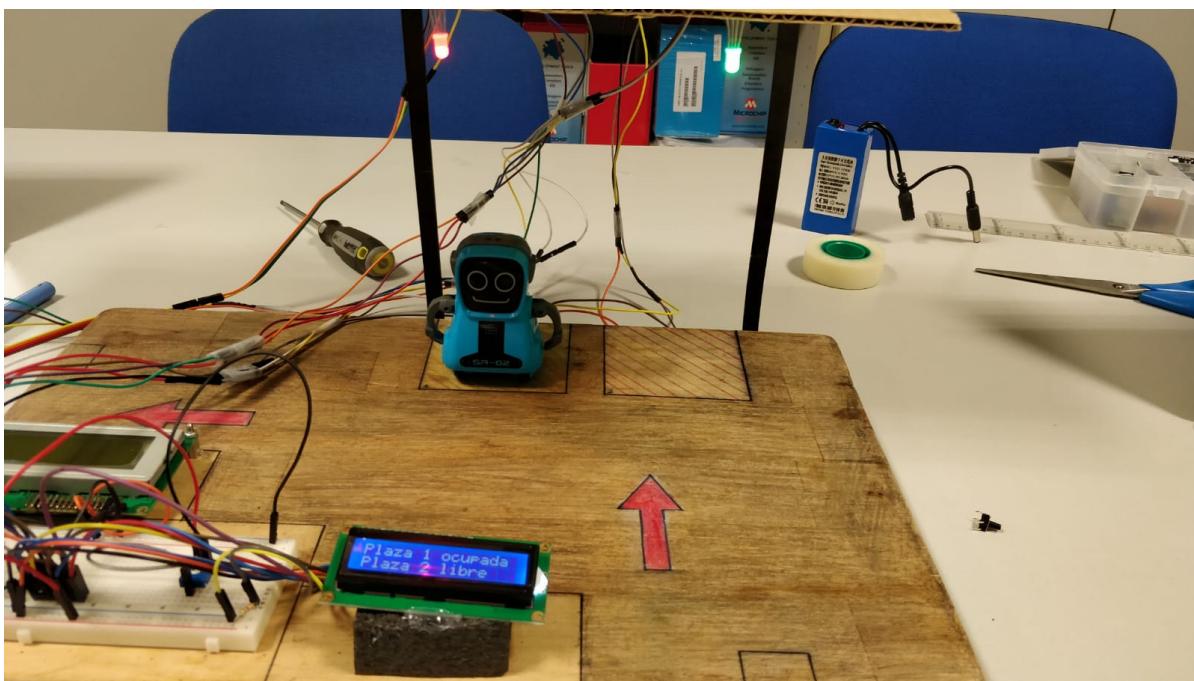
En esta imagen anterior vemos como hemos conectado el primer LED al techo del aparcamiento y estamos realizando el código. Ya se empiezan a ver los primeros cables, los cuales uniremos posteriormente para que el montaje quede más claro.



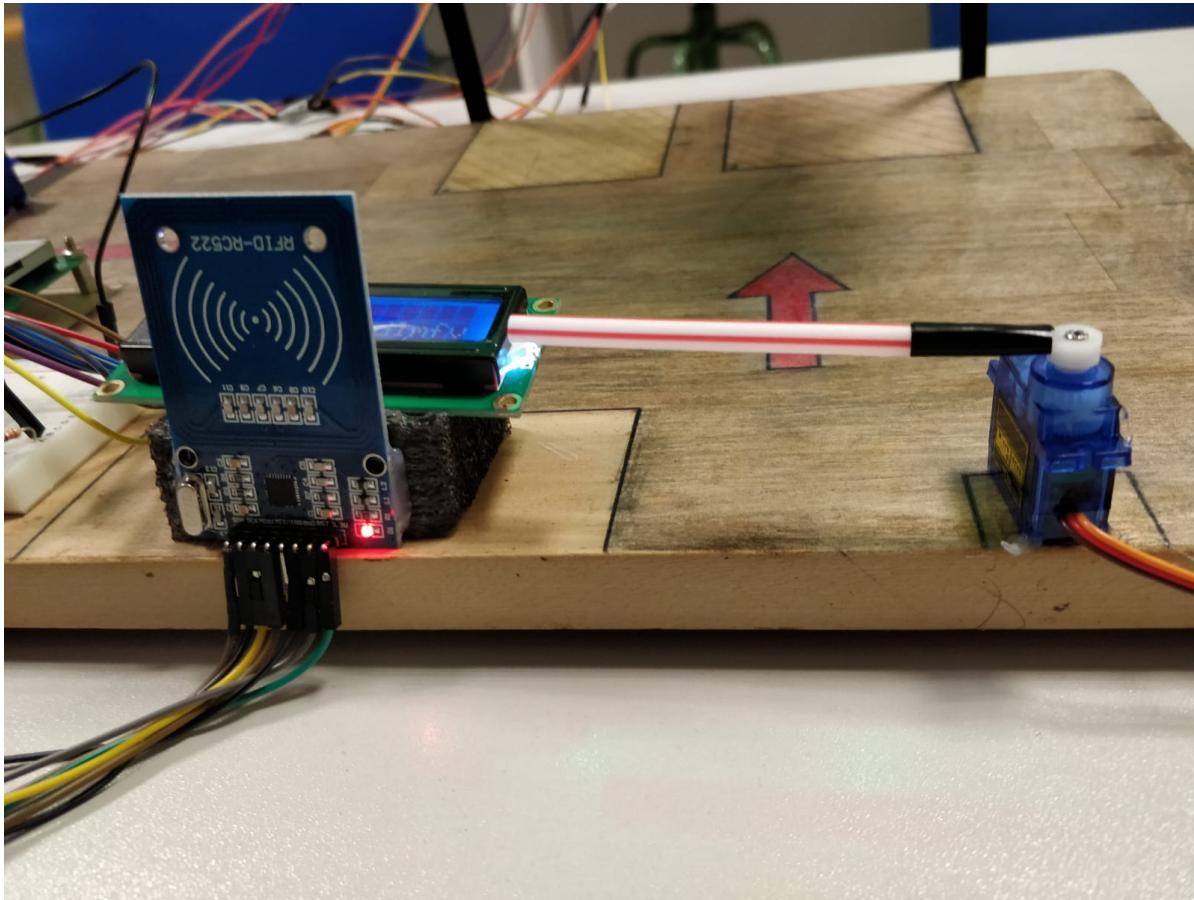
En este caso, vemos como ya hemos pegado los sensores de ultrasonidos al techo del parking y debajo vemos los LED RGB previamente colocados. También se ve que hemos agrupado los cables mediante cinta aislante para evitar problemas y confusiones.



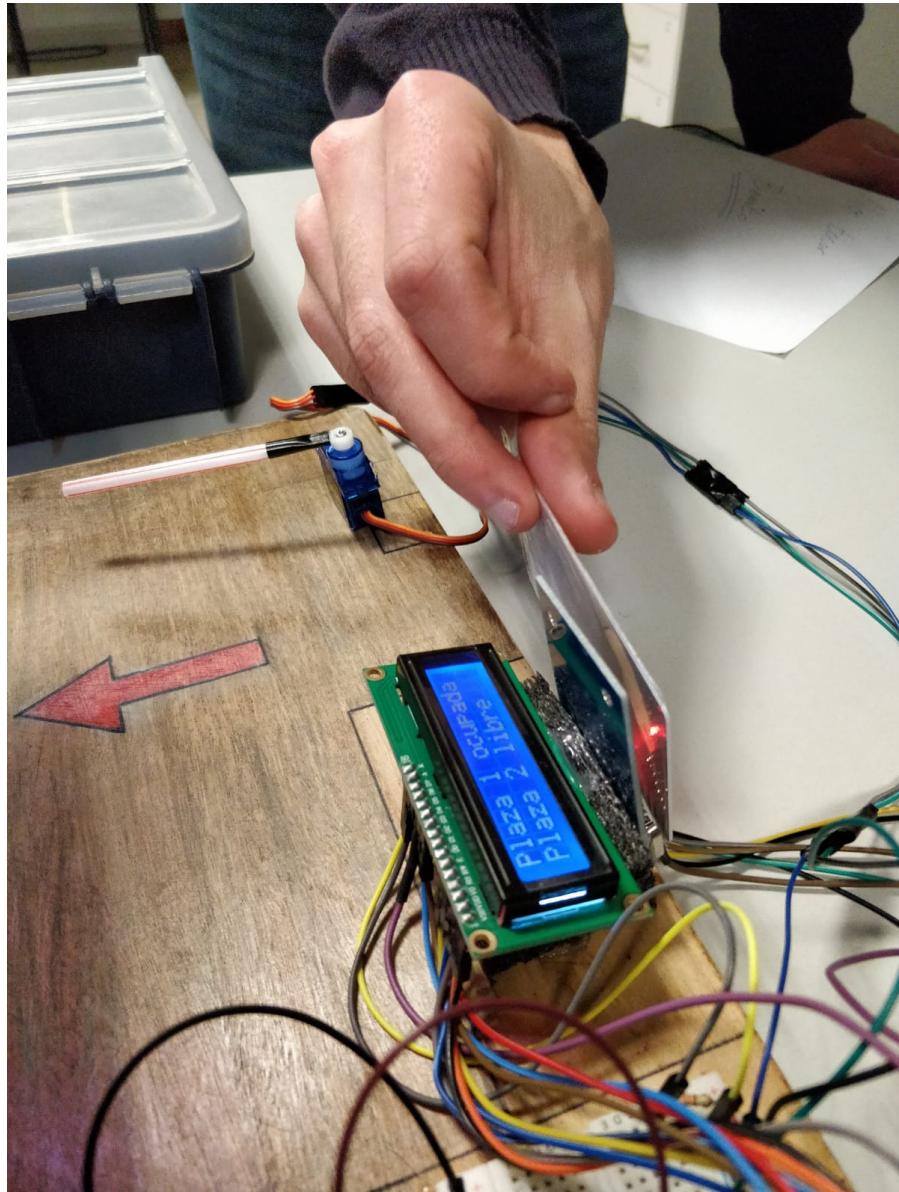
Como podemos observar, ya tenemos los displays conectados y las plazas de aparcamiento muestran un estado de "libre", es decir, verde. Hay muchos cables sueltos todavía pero aún estamos lejos de tener el montaje finalizado, aunque vamos avanzando.



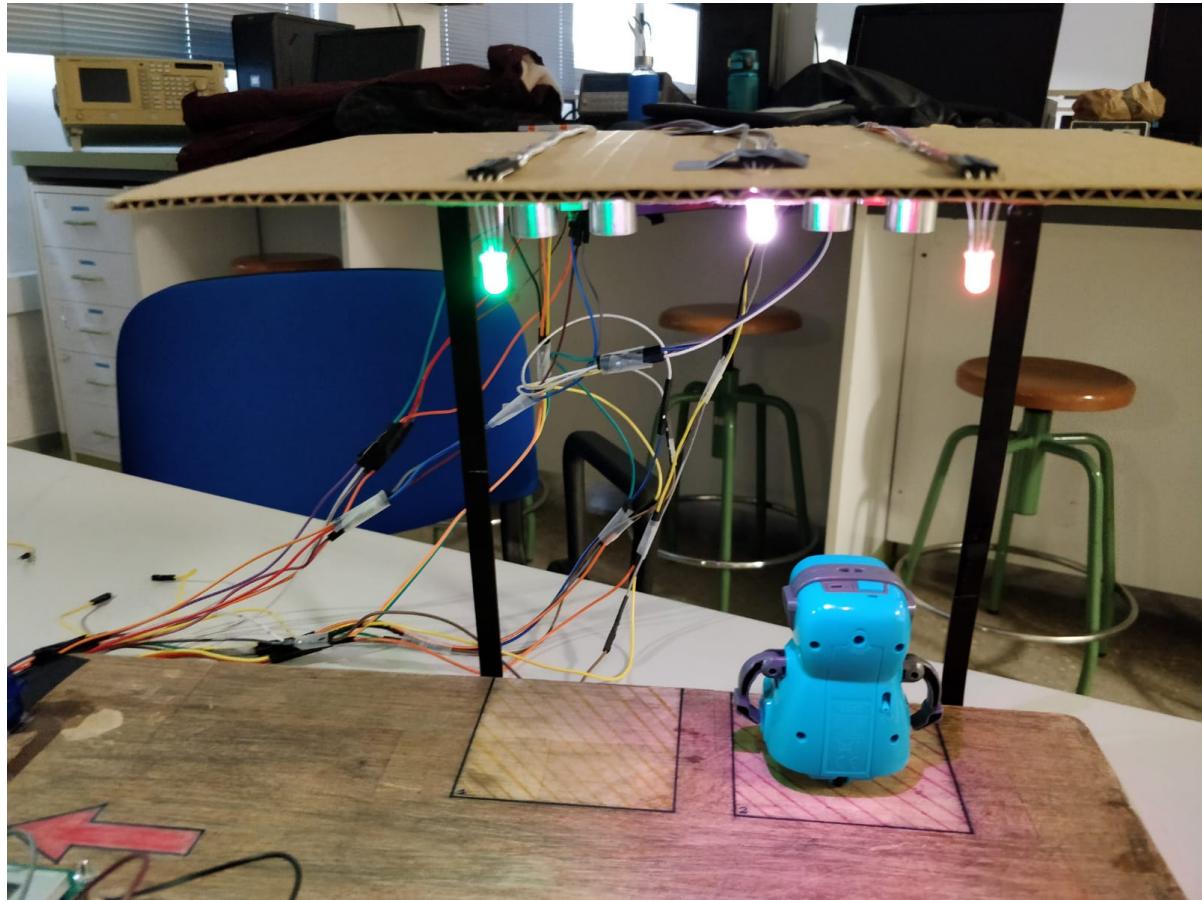
Aquí ya empezamos a ver una de las funcionalidades de la práctica funcionando correctamente. El robot está en la plaza 1, y el display nos indica que está ocupada, al igual que el color de LED rojo. Además, hemos introducido la luz trasera del display LCD para obtener un mejor contraste de las letras.



Aquí podemos ver la puerta de entrada al parking ya colocada en su servo correspondiente. La puerta la vamos a abrir mediante una tarjeta NFC al aproximarla a su sensor.



Aquí podemos apreciar como al acerar la tarjeta RFID el sensor la detecta y la puerta se abre. La imagen está tomada justo en el instante donde el servo ha terminado su giro, antes de volver a cerrarse.



En esta última imagen vemos como la luz de color blanco del tercer LED RGB se encuentra encendida. Esto se debe a que la luz que detecta el sensor LDR ha disminuido por debajo del umbral que le hemos marcado, y por tanto el LED pasa a encenderse para iluminar el parking dado que entiende que es de noche.

4. Código

Aunque se adjunta el código junto a este documento, vamos a incluir una imagen de las partes más importantes del mismo. Se basa en la estructura de código clásica de Arduino y en general de muchos microcontroladores, con los métodos *setup()* y *loop()*.

Antes de estos métodos, podemos ver la inicialización de los pines a usar, que al ser tantos sensores llega a un tamaño considerable.

```

const int PINSERVO1 = 10;
const int PINSERVO2 = 11;
const int LUZ = 12;
const int RST_PIN = 9;           // Pin 9 para el reset del RC522
const int SS_PIN = 53;          // Pin 10 para el SS (SDA) del RC522
const int BOTONSALIDA = 20;
const int DHTPIN = 27;
const int ECHOPIN1 = 28;
const int TRIGGERPIN1 = 29;
const int ECHOPIN2 = 30;
const int TRIGGERPIN2 = 31;
const int RedLED1 = 32;
const int GreenLED1 = 33;
const int RedLED2 = 34;
const int GreenLED2 = 35;
const int rs2 = 7, rw2 = 6, en2 = 5, d42 = 4, d52 = 3, d62 = 2, d72 = 8;
const int rs1 = 36, rw1 = 38, en1 = 40, d41 = 42, d51 = 44, d61 = 46, d71 = 48;

const long DistanciaAlSuelo1 = 14;
const long DistanciaAlSuelo2 = 14.5;

int sensorLDRPin = A0;
unsigned long time;
Servo servo1; // crea el objeto servo
Servo servo2; // crea el objeto servo

DHT dht(DHTPIN, DHTTYPE);

LiquidCrystal lcd1(rs1, rw1, en1, d41, d51, d61, d71);
LiquidCrystal lcd2(rs2, rw2, en2, d42, d52, d62, d72);

byte validKey1[4] = { 0x91, 0x9C, 0xD3, 0x2F }; // clave NFC valida
MFRC522 mfrc522(SS_PIN, RST_PIN); // Crear instancia del MFRC522

```

En el setup, inicializamos todas las variables, los dos LCDs, el sensor de temperatura y humedad, el lector RFID, los servos, un timer para dotar a los servos de una leve pausa antes de volver a cerrar las puertas (que debería estar regido por un sensor infrarrojo o algo por el estilo pero por falta de tiempo es lo que tenemos), una marca de tiempo que usaremos para dejar 2 segundos entre cada actualización de temperatura y humedad, configuramos la entrada o salida de los pines restantes que controlaremos a mano y montamos la interrupción que será lanzada por el botón de la puerta de salida.

```
Serial.begin(9600);
lcd1.begin(16, 4);
lcd2.begin(16, 2);

dht.begin();
SPI.begin();           // Iniciar SPI
mfrc522.PCD_Init(); // Iniciar MFRC522
servo1.attach(PINSERVO1); // vincula el servo al pin digital 9
servo2.attach(PINSERVO2); // vincula el servo al pin digital 9
servo1.write(179);

Timer3.initialize(2000);

time = millis();

pinMode(RedLED1, OUTPUT);
pinMode(GreenLED1, OUTPUT);
pinMode(RedLED2, OUTPUT);
pinMode(GreenLED2, OUTPUT);
pinMode(LUZ, OUTPUT);
digitalWrite(GreenLED1, HIGH);
digitalWrite(RedLED1, LOW);
digitalWrite(GreenLED2, HIGH);
digitalWrite(RedLED2, LOW);

pinMode(TRIGGERPIN1, OUTPUT);
pinMode(ECHOPIN1, INPUT);
pinMode(TRIGGERPIN2, OUTPUT);
pinMode(ECHOPIN2, INPUT);
pinMode(BOTONSALIDA, INPUT_PULLUP);
attachInterrupt(digitalPinToInterruption(20), servoFin, RISING);
```

En el loop, tenemos tres bloques básicos:

En una parte, comprobamos la luz existente y, si baja de un cierto umbral, enciende el LED de luz, y si han pasado más de 2 segundos, realizamos la actualización de los valores de temperatura y humedad.

```
lcd1.setCursor(0, 0);
lcd2.setCursor(0, 0);

int luz = analogRead(sensorLDRPin);

if (luz < 100)
    digitalWrite(LUZ, LOW);
else
    digitalWrite(LUZ, HIGH);

if(millis() >= time+1000){
    lcd1.clear();
    float h = dht.readHumidity(); //Leemos la Humedad
    float t = dht.readTemperature(); //Leemos la temperatura en grados Celsius
    float hic = dht.computeHeatIndex(t, h, false );
    lcd1.print("Humedad: ");
    lcd1.print(h);
    lcd1.print(" %\t");
    lcd1.setCursor(0, 1);
    lcd1.print("Temp.: ");
    lcd1.print(t);
    lcd1.print(" *C ");
    lcd1.setCursor(0, 2);
    lcd1.print("Heat index: ");
    lcd1.print(hic);
    lcd1.println(" *C");
    time = millis();
}
```

En la segunda parte, realizamos la comprobación de las plazas de aparcamiento, actuando en caso de que esté libre u ocupada.

```

int cm1 = ping(TRIGGERPIN1, ECHOPIN1);
delayMicroseconds(100);
int cm2 = ping(TRIGGERPIN2, ECHOPIN2);

//////////////////////////////Ultrasonidos 1
//Serial.print("Distancia1: ");
//Serial.println(cm1);
if(cm1 < DistanciaAlSuelo1){
    digitalWrite(RedLED1, HIGH);
    digitalWrite(GreenLED1, LOW);
    Serial.println("Coche aparcado en plaza 1");
    lcd2.print("Plaza 1 ocupada");
} else{
    digitalWrite(GreenLED1, HIGH);
    digitalWrite(RedLED1, LOW);
    lcd2.print("Plaza 1 libre ");
}
//////////////////////////////Ultrasonidos 1

lcd2.setCursor(0, 1);
//////////////////////////////Ultrasonidos 2
//Serial.print("Distancia2: ");
//Serial.println(cm2);
if(cm2 < DistanciaAlSuelo2){
    digitalWrite(RedLED2, HIGH);
    digitalWrite(GreenLED2, LOW);
    Serial.println("Coche aparcado en plaza 2");
    lcd2.print("Plaza 2 ocupada");
} else{
    digitalWrite(GreenLED2, HIGH);
    digitalWrite(RedLED2, LOW);
    lcd2.print("Plaza 2 libre ");
}
//////////////////////////////Ultrasonidos 2

```

En la última parte, comprobamos si se ha pasado una tarjeta NFC en por el lector, y en caso afirmativo comprobamos si es un usuario registrado para abrir la barrera o no.

```

if (mfrc522.PICC_IsNewCardPresent()) {
    //Seleccionamos una tarjeta
    if (mfrc522.PICC_ReadCardSerial())
    {
        // Comparar ID con las claves válidas
        if (isEqualArray(mfrc522.uid.uidByte, validKey1, 4)){
            Serial.println("Tarjeta valida");
            servo1.write(95);
            delay(2000);
            servo1.write(175);
            //Timer3.attachInterrupt(servoInicio);
        } else
            Serial.println("Tarjeta invalida");
        // Finalizar lectura actual
        mfrc522.PICC_HaltA();
    }
}

```

5. Problemas encontrados

En un primer momento quisimos diseñar un juego basando en una máquina que lanzaba pompas y que al impactar en distintos lugares de una diana, mediante sensores de humedad, se obtenía una puntuación para el jugador que movía el ventilador del pompero con un joystick. Desechamos la idea por ciertas dudas con respecto al montaje, en concreto con los sensores de humedad. No sabíamos si iban a ser capaces de detectar la variación de humedad necesaria, además de que el montaje en sí mismo del 'escenario' del juego iba a ser bastante complejo.

Más tarde se nos ocurrió la idea de Arduparking, y nos pareció una forma de unir prácticamente todos los conocimientos aprendidos en clase mientras que imaginábamos una aplicación real. En muchos centros comerciales se tiene control sobre los parámetros ambientales del mismo, por si existe algún problema. Además, todas las filas de coches disponen de sensores de proximidad que indican mediante un LED adyacente si la plaza está ocupada o no.

En general y hablando de Arduparking, durante el desarrollo de la práctica no hemos encontrado grandes problemas a la hora de realizar el código o montar los componentes. La duda más grande era cómo representar el escenario del propio parking, y tener cuidado a la hora de la colocación de los elementos en él. Los sensores de ultrasonidos debían estar apuntando hacia abajo, simulando estar en el techo del parking, y esto nos supuso algunas dudas. En resumen, al ya haber trabajado con estas herramientas, la mayor complejidad ha residido en cómo utilizarlas en un proyecto completamente distinto, y no simplemente conectadas a la placa.

6. Conclusiones

La asignatura nos ha fascinado, literalmente. Es cierto que las clases teóricas pueden no parecer tan 'fascinantes' como las prácticas, dado que no estás usando las herramientas y los componentes directamente, sino que te limitas a aprender sobre ellos. Las bases teóricas en clase han estado bien explicadas, y las charlas que se han impartido han sido muy interesantes. Conocer a personas que se dedican de primera mano al mundo de los microcontroladores es un privilegio que nos ayuda a darnos cuenta del alcance de los mismos.

Respecto a las prácticas, las sesiones de laboratorio han sido muy divertidas e instructivas. Ir trabajando sobre la marcha era muy curioso, dado que debíamos tomar una fotografía de cada montaje para usarla en nuestras prácticas finales. No solo estás aprendiendo, sino que estás avanzando las prácticas de la asignatura. Por otro lado, las dudas están muy bien resueltas por el profesor, que se muestra totalmente receptivo para ayudar a los grupos a seguir avanzando.

Por último, dar las gracias al profesor por su atención y dedicación durante estos meses, y esperamos que se haya entretenido tanto como nosotros, algo de lo que estamos prácticamente seguros.