

CONTROL DE ENTRADA Y SALIDA DE VEHICULOS EN UN PARKING PROGRAMADO EN ARDUINO



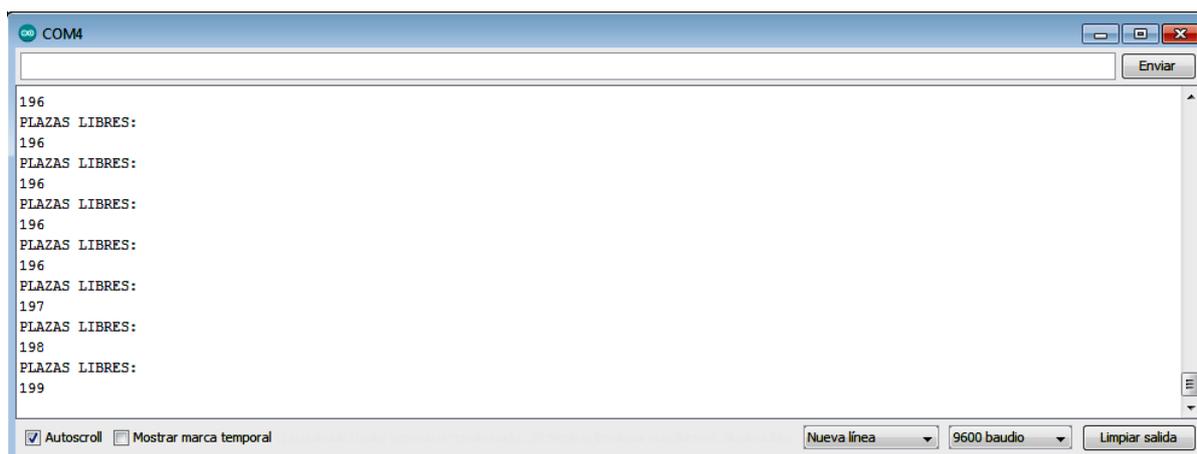
Introducción

Este proyecto que se describe a continuación consiste en controlar la entrada y salida de los vehículos en un Parking y tiene como principal tarea de detectar el número de vehículos que entran y salen del parking informando continuamente del número de plazas libres y avisando cuando el parking está completo. Para ello, también se va a disponer exteriormente de un semáforo que nos indica que el Parking está libre o completo, en nuestro caso, son dos diodos LED de color verde para libre y rojo para completo.

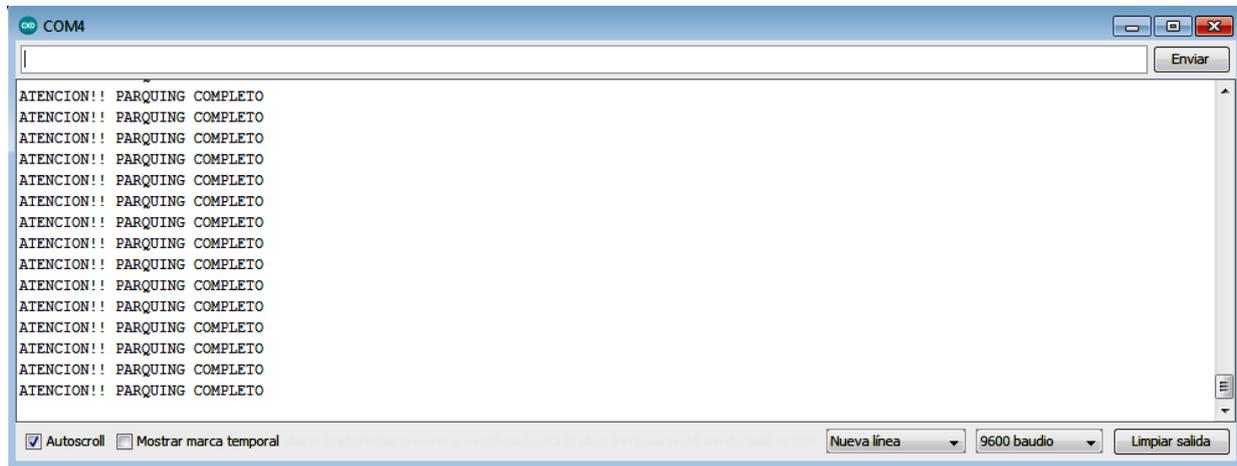
Los componentes utilizados en este proyecto son básicamente diodos **LED** y resistencias dependientes de la luz **LDR**, con ello se pretende obtener una base referente a como trabajan en las instalaciones profesionales.

Todo el proceso de control se realiza mediante la programación en **Arduino UNO** del **microcontrolador ATmega328P**, que es quien procesa las señales de entrada de los detectores y hace señalizar el estado de las plazas libres o completo y el estado correcto de los detectores fotoeléctricos.

A través del **monitor serie** de Arduino se podrá visualizar el número de plazas libres en cada momento y cuando está completo. A la misma vez que entra un vehículo o sale se va contabilizando las **PLAZAS LIBRES: XXX** y **LED verde**, visualizando cuando está completo: **ATENCIÓN!!! PARKING COMPLETO** y **LED rojo**.



A través del monitor serie de Arduino se puede visualizar el número de plazas libres, obtenido de la diferencia de plazas disponibles (salidas) y de las que se van ocupando (entradas).



Del mismo modo cuando las plazas esten todas ocupadas, se visualiza en el monitor serie de Arduino “Parking Completo” y se enciende el LED rojo de “Completo”, momento que desaparece cuando un vehículo salga del parking y vuelve aparecer “Plazas libres” y se encienda el LED verde de “Libre”.

Componentes electronicos utilizados

Los componentes empleados en nuestro proyecto son básicamente dos resistencias LDR y cinco diodos LED.

LDR o FOTORRESISTENCIA COMO DETECTOR

Las siglas **LDR** provienen del ingles *LIGHT DEPENDENT RESISTOR* (resistencia que depende de la luz).

El valor ohmico que nos proporciona una **LDR** variará dependiendo de la cantidad de luz que incida sobre ella. El valor de la resistencia será bajo cuando la luz incida sobre ella, y será alto cuando no incida ninguna luz. Por ejemplo, en nuestro caso se ha medido un valor de resistencia de unos $10K\Omega$ cuando no recibia absolutamente ninguna luz, y de unos $2K\Omega$ cuando recibia luz directamente, variando su valor conforme la intensidad de luz que incide varia de mayor a menor.

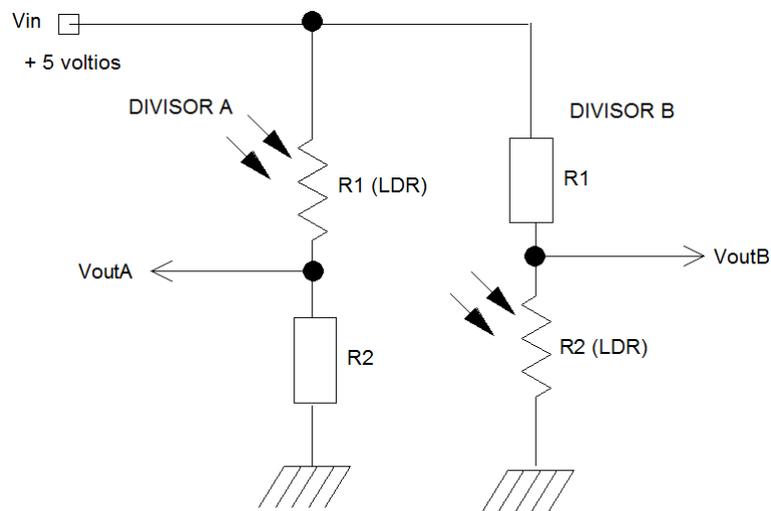


Aspecto físico de una resistencia LDR

En nuestro proyecto el funcionamiento de las LDRs se ha configurado utilizando un divisor de tensión resistivo. Para ello se ha tenido en cuenta para la programación la necesidad de un nivel lógico bajo (0) para establecer la entrada detectora digital a nivel bajo, puesto que las dos entradas de las dos LDR irán a puertos de entradas digitales del microcontrolador D6 y D7.

Para nuestra programación necesitamos seleccionar el valor que nos permita tener dos valores perfectamente discretos para la tensión de salida VoutA, estos son, tensión alta que nos identifica que la LDR está recibiendo luz y tensión baja cuando la LDR no recibe ninguna luz.

Para tener una pequeña referencia de como se ha configurado una LDR para nuestro proyecto partimos de la configuración y calculo de dos tipos de divisores de tensión que se representa en el siguiente esquema denominados DIVISOR A y DIVISOR B.



El calculo de la tensión de salida de las diferentes configuraciones de divisores se calcula mediante la siguiente formula:

$$V_{out} = \frac{R_2}{R_1 + R_2} \times V_{in}$$

Se ha creado dos tablas para cada configuración con diferentes valores de resistencias de la LDR para el divisor A y el divisor B, calculados mediante la formula anterior:

Configuración divisor A			
LDR sin luz		LDR con luz	
R1 (LDR)	10000Ω	R1 (LDR)	2000Ω
R2	680Ω	R2	680Ω
Vin	5 voltios	Vin	5 voltios
VoutA	0,454 voltios	VoutA	1,6 voltios

Si la R1 es la LDR, obtendremos una tensión V_{outA} alta en la salida si incide luz en la LDR, mientras que obtendremos una tensión baja (0) o aproximadamente cuando no incide luz en la LDR.

Configuración divisor B			
LDR sin luz		LDR con luz	
R1	680 Ω	R1	680 Ω
R2 (LDR)	10000 Ω	R2 (LDR)	2000 Ω
Vin	5 voltios	Vin	5 voltios
VoutB	4,54 voltios	VoutB	3,3 voltios

Actuando en la configuración divisor B sólo debemos intercambiar la LDR (R1) con la R2, y aplicar la fórmula, con lo que cuando la luz incide, la tensión es baja, y cuando la luz no incide en la LDR, la tensión de salida V_{outB} es alta.

Por lo tanto, elegiremos el divisor A que nos permite obtener los estados (0) y (1): el estado bajo (0) cuando no recibimos ninguna luz y el estado alto (1) cuando se recibe luz.

DIODOS LED

El diodo LED es un componente semiconductor empleado para señalar emitiendo luz a bajas tensiones. En nuestro proyecto lo utilizamos como señalización: Parking Libre (LED verde), Parking Completo (LED rojo) y como comprobante y señalización del estado de los detectores LDR (LED verde OK).

Existen muchas variedades de diodos LED cuya tensión de trabajo varían de unos a otros, por lo general los más utilizados pueden variar su tensión de trabajo debido al material que lo componen, así tenemos:

Diodo LED	Tensión (voltios)
Color rojo	1,7 V
Color verde	2 V
Color amarillo	2 V
Color azul	2,4 V
Color blanco	2,6 V

La tensión utilizada en nuestro proyecto es de 5 voltios en continua, por lo tanto, hay que tener en cuenta que es necesario colocar una resistencia limitadora en serie para evitar que el diodo LED se funda a causa de una tensión mayor de la que está preparado, por ejemplo, aplicar

directamente 5 voltios en un LED cuya tensión de funcionamiento es de 1,7 voltios lo llevaría al extremo de fundirlo a corto plazo.

Para calcular la resistencia limitadora y de protección del diodo LED se puede calcular mediante la siguiente formula:

$$R_{Lim} = \frac{V_{in} - V_{Led}}{I_{nt Led}}$$

Para un valor de tensión de entrada de 5 voltios y la tensión del LED de 1,7 voltios con una corriente de funcionamiento de 20 mA obtenemos un valor de 165 Ω , seleccionamos por lo alto un valor estandar y sería aproximadamente de unos 220 Ω .

Con una tensión de 5 voltios se puede elegir un valor de resistencia en serie para el diodo LED entre 220-330 Ω .

Un aspecto que se debe tener en cuenta es que el diodo LED, a diferencia de la resistencia, tiene polaridad y se diferencia que el terminal cátodo, internamente su estructura es mucho más ancha que la del ánodo, aparte tiene en la superficie exterior como un pequeño borde achatado.

Como se ha comentado anteriormente se ha utilizado en nuestro proyecto cinco diodos LED, dos de ellos de color blanco de alta luminosidad conectados directamente a la tensión de +5Vcc para iluminar la LDR sin problemas, en nuestro caso la distancia entre la LDR y el LED sería bastante pequeña, solamente nos serviría para comprobar como funciona el proyecto. Para obtener una mayor distancia se puede utilizar el **diodo LASER**, que como sabrás tiene un alcance mucho mayor y su luz es más intensa, pero a la vez su alineación con la LDR, según la distancia, es un poco más laborioso.

Los otros tres diodos LED se conectan a la salida de los puertos D9, D10 y D11 del microcontrolador para indicar que hay plazas libres (verde) o está completo (rojo) y otro LED verde para informarnos de que las dos LDR están correctas y detectando.

Esquema eléctrico. Microcontrolador ATmega328P

La realización de este proyecto está basada en el control y programación a través de la tarjeta UNO R3 de Arduino con el microcontrolador ATmega 328P de 8bits. El formato utilizado de este microcontrolador es un DIP, que viene introducido en un zócalo y se puede extraer con relativa facilidad si utilizamos un extractor adecuado y lo introducimos, ya programado, en una PCI con su correspondiente zócalo de 28 pines, y sus elementos de oscilación: cristal de cuarzo de 16MHz y condensadores de 22pF para la velocidad de proceso. (Ver esquema eléctrico)

La alimentación del microcontrolador es de 5 voltios de corriente continua.

La comunicación del microcontrolador con el PC se lleva a cabo mediante comunicación serie; esto quiere decir que los bits llegan a Arduino de uno en uno, y el microcontrolador trabaja con grupos de 8 bits, por lo que se dispone de otro circuito integrado soldado en la tarjeta de Arduino UNO llamado UART para adecuar la llegada de los bits al microcontrolador. Por lo tanto, la comunicación entre el PC y la tarjeta de Arduino se realiza por un cable o interfaz USB, que es una forma de transmitir los bits uno en uno, proporcionando la visualización de datos mediante el monitor serie cuando lo requiera la programación.

NOTA: En el caso de un montaje independiente y utilizando una placa de circuito impreso PCI se deberá disponer del montaje del circuito integrado UART (Transmisor – Receptor Asíncrono Universal) que se encarga de gestionar la transmisión y recepción de los bits y poder visualizar a través del PC el monitor serie de Arduino, si así lo requiere la programación.

Datos para la programación

Como se ha comentado anteriormente utilizamos cinco puertos del microcontrolador ATmega328P, dos de entradas y tres de salida.

El puerto digital D7 (pin13) lo utilizaremos de **entrada INPUT** para controlar la entrada de vehículos mediante la **LDR2** y el puerto D6 (pin12) lo utilizamos de **entrada INPUT** para controlar la salida de vehículos mediante la **LDR1**.

Las salidas digitales **OUTPUT** de los puertos D10 (pin16) y D11 (pin17) se encargarán de señalar el LED4 verde de “Plazas Libres” y el LED5 rojo de “Parking completo”.

La salida digital **OUTPUT** conectado al puerto D9 (pin15) del LED3 verde, comprobará en todo momento cuando las dos fotoresistencias LDR1 y LDR2 se encuentran en buen estado recibiendo luz. Si hubiera algún obstáculo o fallos permanente estaría apagado este LED3.

La tensión de alimentación utilizada es de 5 voltios en continua conectados +5Vcc a los pines 7 y 20 del microcontrolador y la negativa 0V conectadas a los pines 8 y 22 del microcontrolador.

Su sincronización se realiza mediante un oscilador formado por un cristal de cuarzo de 16MHz XTL1, conectado en los pines PB6 (9) y PB7 (10) y su correspondiente polarización mediante dos condensadores cerámicos de discos C1 y C2 de 22pF.

Estableceremos la monitorización serie de Arduino para visualizar las plazas libres que van quedando, comenzando con el número de 200 plazas.

Programación en Arduino

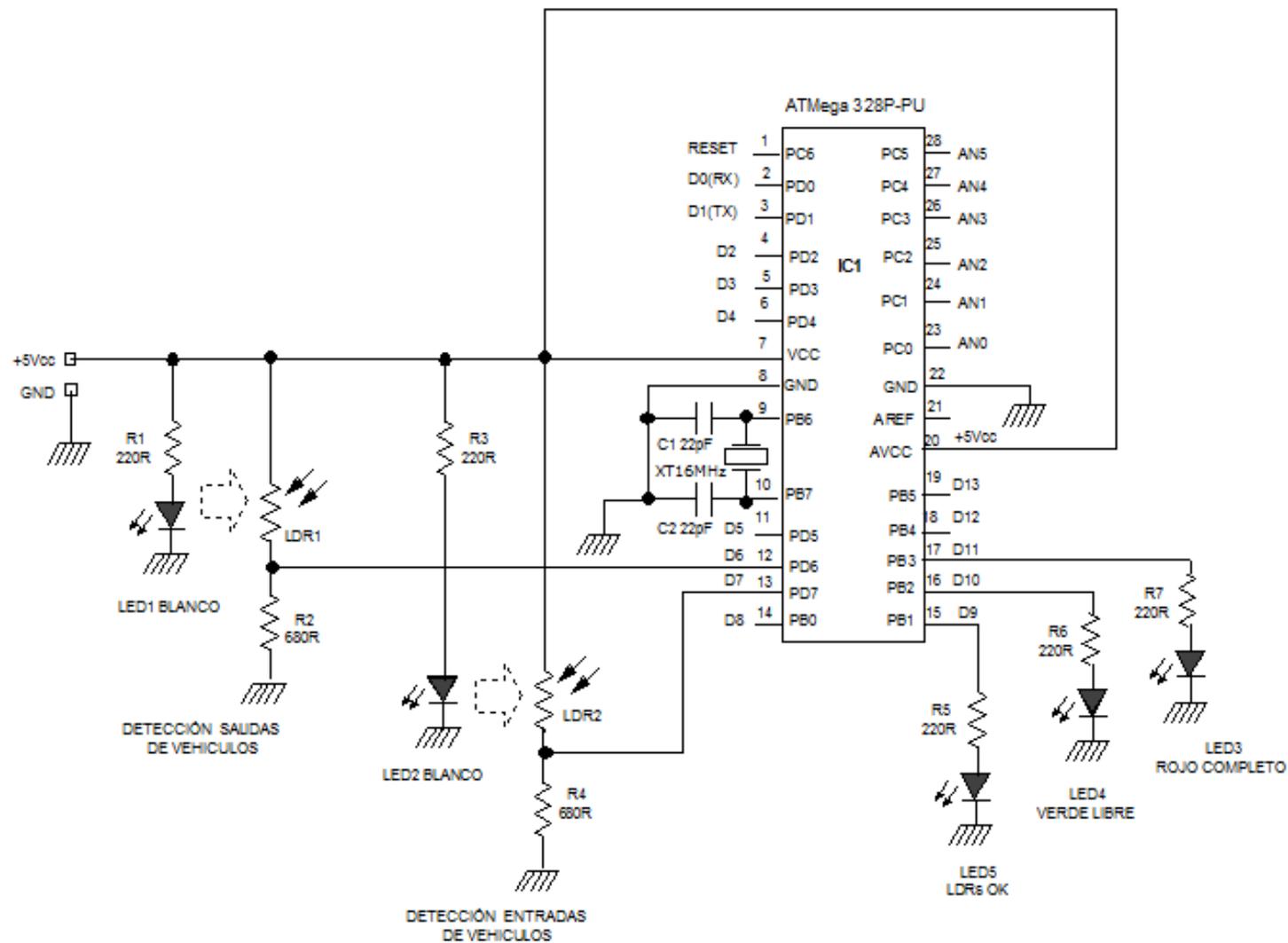
```
/* Control de entrada y salida de vehiculos a un parking*/
int ledrojo=11; // pin D11 para el led rojo parking completo
int ledverde=10; // pin D10 para el led verde plazas libres
int ledverde2=9; // pin D9 para el led verde de OK fotoresistencias
int ldrE=7; //pin D7 de conexion LDR2 de entrada
int ldrS=6; //pin D6 de conexion LDR1 de salida
int contadorE=0; // contador de vehiculos de entrada
int contadorS=0; // contador de vehiculos de salida
int contador=0; //contador total
int valorE; //almacena el estado de la LDR2 de entrada
int valorS; //almacena el estado de la LDR1 de salida

void setup() {
  Serial.begin (9600); //activamos el monitor serie
  pinMode (ledrojo, OUTPUT); //declaramos el pin led rojo de salida
  pinMode (ledverde, OUTPUT); //declaramos el pin led verde de salida
  pinMode (ldrE, INPUT); //declaramos el pin ldrE de entrada
  pinMode (ldrS, INPUT); // declaramos el pin ldrS de salida
  pinMode (ledverde2, OUTPUT); //declaramos este pin para señalizar barrera
  LDR OK
}
void loop() {

  valorE=digitalRead(ldrE); //se lee el valor de la LDR de entrada
  if (valorE==LOW) { // si la LDR de entrada detecta corte de luz
    contadorE++; //almacena cada vez que se corte la luz
    delay(1000); /* los retardos utilizados evita que la visualización de los
datos en el monitor serie se vean muy rapidos */
  }
  valorS=digitalRead(ldrS); //se lee el valor de la LDR de salida
  if (valorS==LOW) { //si la LDR de salida detecta corte de luz
    contadorS++; //almacena cada vez que se corte la luz
    delay(1000); //esperamos un segundo
  }
  if (valorE==HIGH && valorS==HIGH){ //si las dos LDRs estan activas
    digitalWrite(ledverde2, HIGH); // enciende el LED verde de OK
  }
  else { //de lo contrario
    digitalWrite(ledverde2, LOW); //apaga el LED verde de OK
  }
  contador=contadorE-contadorS; /* establecemos el valor de contador
con la diferencia entre el valor del contadorE de entrada
y el valor del contadorS de salida */

  if (contador==200) { /*establecemos el número maximo de plazas de 200 si
llega a este valor*/
    Serial.println ("ATENCION!! PARKING COMPLETO");// se visualiza en el PC
    delay(1000); //esperamos un segundo
  }
}
```

```
digitalWrite (ledrojo, HIGH);// se enciende el led rojo de completo
delay(1000); //esperamos un segundo
digitalWrite (ledverde, LOW);// se apaga el led verde de libre
delay(1000); //esperamos un segundo
}
else { // de lo contrario
  digitalWrite (ledverde, HIGH);// enciende el led verde plazas libres
  digitalWrite (ledrojo, LOW); // apaga el led rojo de completo
  delay(1000); //esperamos un segundo
  Serial.println ("PLAZAS LIBRES:"); //visualiza en el PC
  Serial.println (200-contador); // visualiza en el PC las plazas libres
  delay(1000); //esperamos un segundo
}
}
```



CONTROL DE ENTRADA Y SALIDA DE VEHICULOS EN UN PARKING PROGRAMADO EN ARDUINO

Plano:	Fecha: 04/12/2022	Nº de Hojas: 1/1
P-0126	Dibujado: Jose M. Castillo Castillo	

