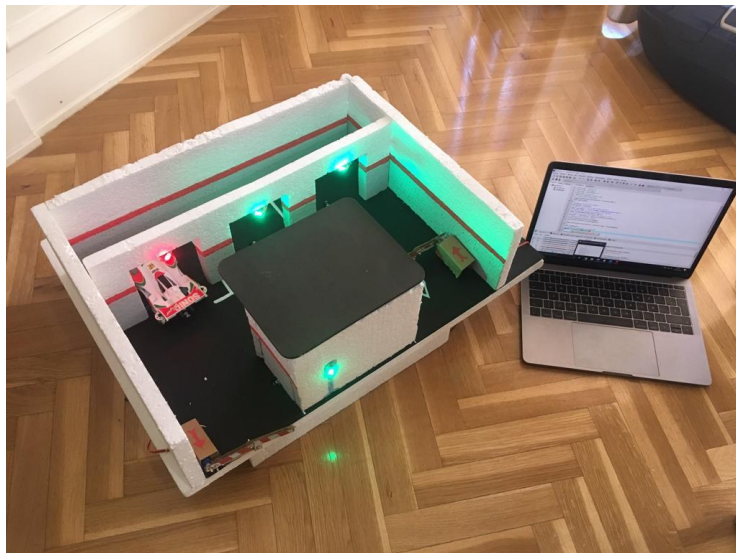


Trabajo informática
Curso 18/19 – Grupo A109

Parking Automático



Nombre	Apellidos	Nº Matricula	Email
Maurizio	Lo Truglio	54698	m.lotruglio@alumnos.upm.es
Oscar	Rudek	54844	oscar.rudek@alumnos.upm.es
Víctor	Rodríguez	54838	victor.rodriguez.sanchaz@alumnos.ump.es

Objetivos

- Crear una maqueta representativa de un parking con sensores infrarrojos que detecten la presencia de los coches.
- Aprender a usar Arduino y entender la comunicación con el ordenador.
- Trabajar en equipo.

Resumen

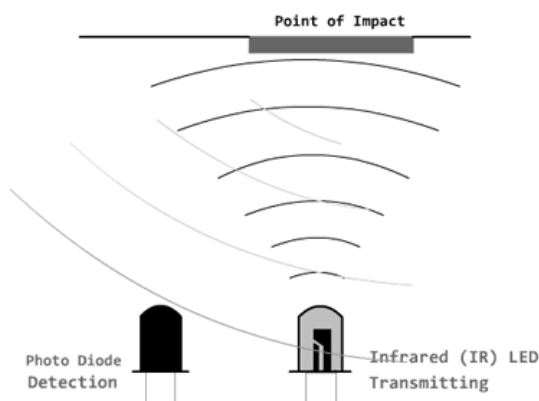
Sistema electrónico automático basado en sensores infrarrojos capaces de detectar la presencia de un vehículo dentro del parking. El trabajo consta de dos servomotores (que actúan de barreras del parking, uno en la entrada y otro en la salida, que se ponen en funcionamiento cuando los sensores situados en la entrada y en la salida localizan los coches. Cuando el parking tiene todas sus plazas ocupadas, el servomotor de la entrada se bloquea y no deja el paso a más coches hasta que la barrera de salida se vuelva a activar. Al estar el parking lleno se enciende el LED rojo del semáforo, mientras haya plazas libres dentro permanecerá encendido el LED verde.

En las plazas del parking están situados tres sensores más, de forma que cuando el coche se sitúe encima del sensor, este se activará y mandará la orden de que el LED verde (que simboliza plaza libre) se apague y se encienda el Led rojo.

Hardware

Un detector de obstáculos infrarrojo es un dispositivo que detecta la presencia de un objeto mediante la reflexión que produce en la luz. El uso de luz infrarroja (IR) es simplemente para que esta no sea visible para los humanos.

Constitutivamente son sensores sencillos. Se dispone de un LED emisor de luz infrarroja y de un fotodiodo (tipo BPV10NF o similar) que recibe la luz reflejada por un posible obstáculo.

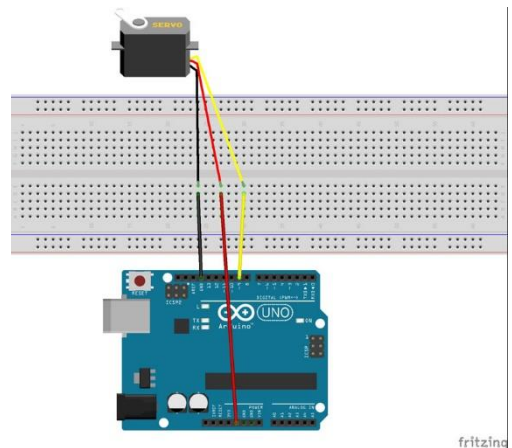


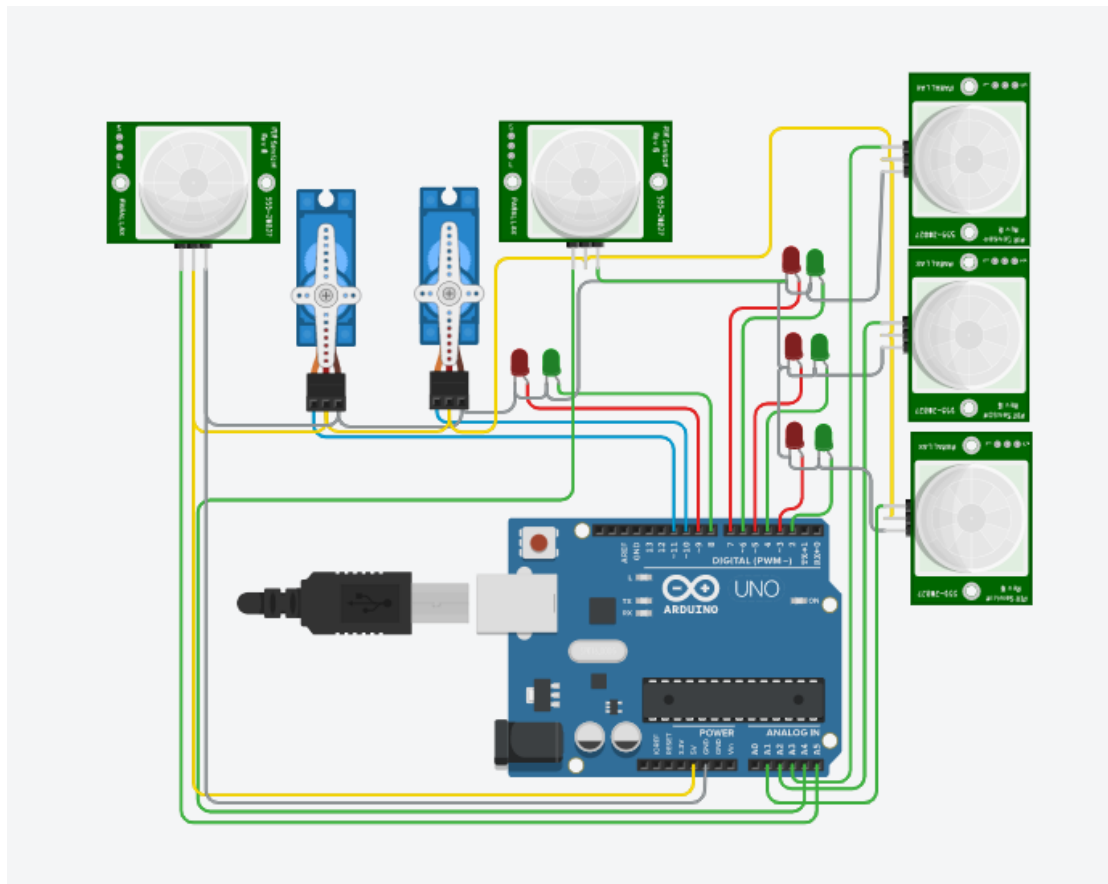
Este tipo de sensores actúan a distancias cortas, típicamente de 5 a 20mm. Además, la cantidad de luz infrarroja recibida depende del color, material, forma y posición del obstáculo, por lo que no disponen de una precisión suficiente para proporcionar una estimación de la distancia al obstáculo.

Pese a esta limitación son ampliamente utilizados para la detección de obstáculos en pequeños vehículos o robots. Su bajo coste hace que sea frecuente ubicarlos en el perímetro, de forma que detectemos obstáculos en varias direcciones.

También son útiles en otro tipo de aplicaciones como, por ejemplo, detectar la presencia de un objeto en una determinada zona, determinar una puerta está abierta o cerrada, o si una máquina ha alcanzado un cierto punto en su desplazamiento.

Un servomotor o servo es un motor eléctrico pero con dos características especiales. Por un lado, nos permite mantener una posición que indiquemos, siempre que esté dentro del rango de operación del propio dispositivo. Por otro lado nos permite controlar la velocidad de giro, podemos hacer que antes de que se mueva a la siguiente posición espere un tiempo.





Boceto del circuito eléctrico utilizado

Código utilizado

```
#include <Servo.h>

#define max 3

#define verde1 2
#define rojo1 3

#define verde2 4
#define rojo2 5

#define verde3 6
#define rojo3 7

#define semaforoverde 8
#define semafororojo 9

#define servoentrada 10
#define servosalida 11
```

```
#define ir1 1
```

```
#define ir2 2
```

```
#define ir3 3
```

```
#define irentrada 4
```

```
#define irsalida 5
```

```
int barreras(int); // Una función encargada de subir y bajar las barreras y contar  
el numero de plazas que hay
```

```
void plazas(void); // Función encargada de indicar si la plaza está libre u  
ocupada mediante LEDS
```

```
void semaforo(int); // Función encargada de mostrar si hay plazas libres o no
```

```
Servo myservoent; // Servo de la barrera de la entrada
```

```
Servo myservosal; // Servo barrera de salida
```

```
int plazasocupadas=0; //variable que cuenta las plazas ocupadas
```

```
void setup(){
```

```
//Definición de los pines y posición inicial de los servos
```

```
    Serial.begin(9600);
```

```
    pinMode(verde1, OUTPUT);
```

```
    pinMode(verde2, OUTPUT);
```

```
    pinMode(verde3, OUTPUT);
```

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

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

```
    pinMode(rojo3, OUTPUT);
```

```
    pinMode(semaforoverde, OUTPUT);
```

```
    pinMode(semafororojo, OUTPUT);
```

```
    myservoent.attach(servoentrada);
```

```
    myservosal.attach(servosalida);
```

```
    myservoent.write(0);
```

```

    myservosal.write(0);
}

void loop(){

    int aux=plazasocupadas;
    plazas();
    plazasocupadas=plazasocupadas+barreras(plazasocupadas);
    semaforo(plazasocupadas);
    if(plazasocupadas>max)plazasocupadas=max;
    if(plazasocupadas<0)plazasocupadas=0;
    if(aux!=plazasocupadas){
        Serial.print("El numero de plazas ocupadas es ");
        Serial.println(plazasocupadas);
    }
}

int barreras(int pocupadas){
    int cuenta=0;
    if(analogRead(irentrada)<100&&pocupadas<max){
        myservoent.write(90);
        delay(2000);
        myservoent.write(0);
        cuenta++;
    }
    if(analogRead(irsalida)<100){
        myservosal.write(90);
        delay(2000);
        myservosal.write(0);
        cuenta--;
    }
}

```

```
    }  
    return cuenta;  
}  
  
void plazas(){  
    if(analogRead(ir1)<=100){  
        analogWrite(rojo1,200);  
        digitalWrite(verde1,LOW);  
    }  
    if(analogRead(ir2)<=100){  
        analogWrite(rojo2,200);  
        digitalWrite(verde2,LOW);  
    }  
    if(analogRead(ir3)<=100){  
        analogWrite(rojo3,200);  
        digitalWrite(verde3,LOW);  
    }  
    if(analogRead(ir1)>100){  
        digitalWrite(rojo1,LOW);  
        analogWrite(verde1,200);  
    }  
    if(analogRead(ir2)>100){  
        digitalWrite(rojo2,LOW);  
        analogWrite(verde2,200);  
    }  
    if(analogRead(ir3)>100){  
        digitalWrite(rojo3,LOW);  
        analogWrite(verde3,200);  
    }  
}
```

```
void semaforo(int pocupadas){  
    if(pocupadas==max){  
        digitalWrite(semafororojo,HIGH);  
        digitalWrite(semaforoverde,LOW);  
    }  
    if(pocupadas<max){  
        digitalWrite(semafororojo,LOW);  
        digitalWrite(semaforoverde,HIGH);  
    }  
}
```