Control de tráfico

Objetivo:

Diseñar y desarrollar en lenguaje C un programa que permita controlar el tráfico, y calcular la multa si el vehículo comete una incidencia (exceder en velocidad o desobediencia de semáforos).

Especificaciones del proyecto:

Fuente de información control de tráfico:
https://es.wikipedia.org/wiki/Control_de_tr%C3%A1nsito
Fuente de información del radar:
https://www.serviciositv.es/sgs-te-informa/funcionamiento-de-los-radares-como-funciona-un-radar-de-trafico

El control de tránsito o control de tráfico vial implica la organización de la circulación vehicular y peatonal, asegurando la seguridad de los equipos operativos involucrados y del público. Muchas veces, se utilizan diversos medios tecnológicos para manejar el flujo de tránsito proveyendo en caso necesario advertencias sobre congestiones u otros inconvenientes.

En este caso, se ha pensado en incorporar medios como radares de velocidad, para que los usuarios no superen el límite de velocidad, además del radar de movimiento, con el fin de evitar que los peatones y/o conductores obedezcan al semáforo.

El funcionamiento de los radares se basa en lo que se conoce como efecto Doppler, este efecto fue inventado por el físico Christian Andreas Doppler en el año 1842. Es un sistema de frecuencias que mide la velocidad de los objetos que ha detectado previamente. Para saber la velocidad del vehículo, mide las ondas electromagnéticas que se crean entre el objeto y el radar.

En los radares de tráfico, se estudia la frecuencia de las ondas. Las ondas traseras tienden a tener más frecuencia conforme la velocidad del vehículo es mayor, aumentando así la longitud de onda. Cuanto más elevada sea la velocidad del coche, más longitud de onda se genera, y por lo tanto la frecuencia que mide el radar será mayor.

El radar salta y activa su cámara cuando la frecuencia entre las ondas supera un límite que está ligado a los límites de velocidad permitidos en carretera. Por ejemplo, si un vehículo va

a 120 kilómetros por hora la frecuencia de las ondas será más elevada que la de un vehículo que circule a 100 kilómetros por hora.

Dinámica:

Nuestro proyecto dispone de dos sensores (uno de velocidad y otro de movimiento) que activarán una cámara haciendo ésta una foto al instante si el coche incumple las normas de circulación.

Una vez se cometa la incidencia, la foto la almacenaremos manualmente.
Para reconocer la matrícula nosotras accederemos a las fotos y rellenaremos la ficha de la persona que haya cometido la multa.

Para acceder a la información al abrir la aplicación aparecerán una serie de opciones con la información ordenada. Así será más fácil encontrar la información requerida.

1. **Exceso de velocidad** (si se activa el sensor 1 la información entra aquí).
    1.1 Fotos.
2. **Incumplimiento de semáforos** (si se activa el sensor dos la información entra aquí).
    2.1 Fotos.
3. **Salir**.

Primera opción

Si abrimos: ”exceso de velocidad” aparecerán todas las multas cometidas por exceso de velocidad (que habremos rellenado a mano). Además, podemos acceder a las fotos que ha hecho la cámara colocada en este sensor que se encuentran almacenadas en “fotos”.

Segunda opción

Es análogo a la primera opción, pero en este caso las fotos las habrá tomado la cámara colocada en el sensor de movimiento.

Tercera opción

Si se escoge la tercera opción: “salir” el programa finaliza.

Para introducir la información el ordenador pedirá al usuario por medio de un menú de opciones si desea realizar la operación de manera automática o manual:
(esta opción la usaremos para ordenar las fotos y rellenar las fichas de los conductores que hayan cometido la incidencia, ya que el programa almacena por defecto todo de forma automática).

Especificación de requisitos obligatorios:

1) Habrá un fichero en el ordenador que recoja la información de las multas: información del tipo de radar (semáforo o velocidad), además en estos ficheros habrá información adicional como la matrícula del coche y el día o la hora a la que se puso la multa. Esto lo conseguiremos creando estructuras. Además asignaremos memoria dinámica para que el programa pueda obtener la información mientras se ejecuta (calloc/malloc).

2) Además, al final con el uso de una función podremos mostrar por pantalla cuánto dinero deberá abonar cada usuario dependiendo de las infracciones que cometió y a que hora y día fueron.
   Crearemos un nuevo fichero que escriba estos datos y los guarde correctamente en el ordenador.

Especificaciones de requisitos opcionales:

1) También se asignará memoria dinámica para los datos de la misma manera que de forma automática. La única diferencia es que en este caso es el propio usuario el que introduce los datos de forma manual: el ordenador no lee el fichero.

2) Además, el usuario podrá acceder a cualquier información sobre las multas cuando lo necesite gracias a las funciones del programa que muestran por pantalla la información de manera organizada.

Sensor de movimiento:

Fuente: https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/

Los sensores infrarrojos pasivos (PIR) son dispositivos para la detección de movimiento. Son baratos, pequeños, de baja potencia, y fáciles de usar. Por esta razón son frecuentemente usados en juguetes, aplicaciones domóticas o sistemas de seguridad.

Los sensores PIR se basan en la medición de la radiación infrarroja. Todos los cuerpos (vivos o no) emiten una cierta cantidad de energía infrarroja, mayor cuanto mayor es su temperatura. Los dispositivos PIR disponen de un sensor piro eléctrico capaz de captar esta radiación y convertirla en una señal eléctrica.

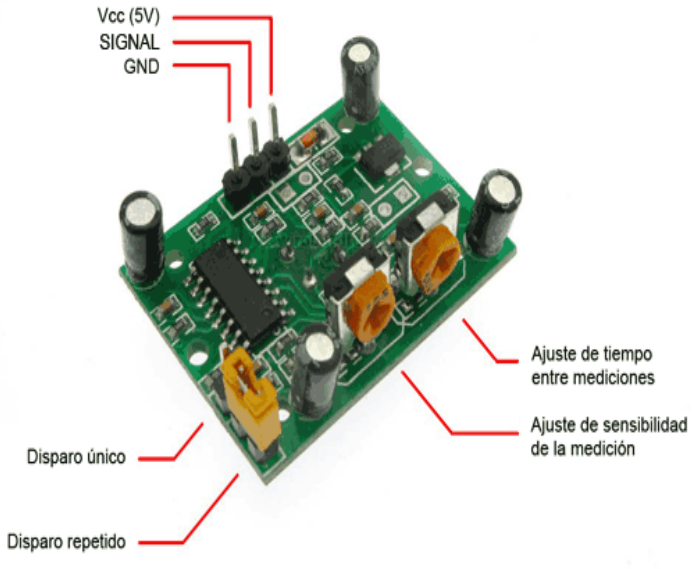
En realidad cada sensor está dividido en dos campos y se dispone de un circuito eléctrico que compensa ambas mediciones. Si ambos campos reciben la misma cantidad de infrarrojos la señal eléctrica resultante es nula. Por el contrario, si los dos campos realizan una medición diferente, se genera una señal eléctrica.

De esta forma, si un objeto atraviesa uno de los campos se genera una señal eléctrica diferencial, que es captada por el sensor, y se emite una señal digital.

El otro elemento restante para que todo funcione es la óptica del sensor. Básicamente es una cúpula de plástico formada por lentes de fresnel, que divide el espacio en zonas, y enfoca la radiación infrarroja a cada uno de los campos del PIR.

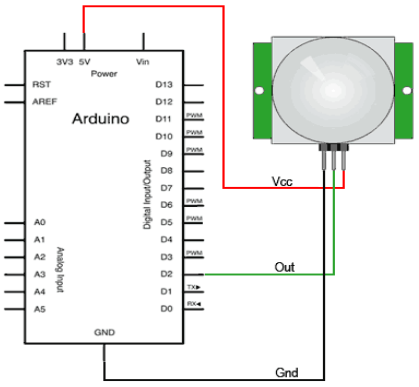
De esta manera, cada uno de los sensores capta un promedio de la radiación infrarroja del entorno. Cuando un objeto entra en el rango del sensor, alguna de las zonas marcadas por la óptica recibirá una cantidad distinta de radiación, que será captado por uno de los campos del sensor PIR, disparando la alarma.

Identificación sensores y actuadores



El esquema de patillaje de un sensor PIR

El esquema eléctrico que necesitamos es el siguiente:



El código necesario para realizar la lectura es simple. Simplemente leemos la salida del PIR, y hacemos parpadear el LED mientras la señal esté activa.

```
const int LEDPin= 13;
const int PIRPin= 2;

void setup()
{
  pinMode(LEDPin, OUTPUT);
  pinMode(PIRPin, INPUT);
}

void loop()
{
  int value= digitalRead(PIRPin);

  if (value == HIGH)
  {
    digitalWrite(LEDPin, HIGH);
    delay(50);
    digitalWrite(LEDPin, LOW);
    delay(50);
  }
  else
  {
    digitalWrite(LEDPin, LOW);
  }
}
```

Si quisiéramos ejecutar una acción una única vez al detectar movimiento, para evitar que la señal esté activa todo el tiempo, empleamos el siguiente código:

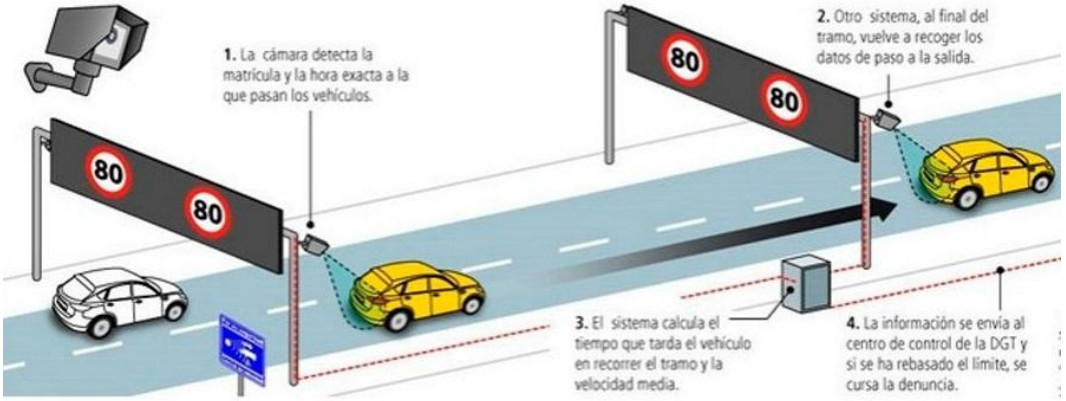
```
const int LEDPin = 13;        // pin para el LED
const int PIRPin = 2;         // pin de entrada (for PIR sensor)

int pirState = LOW;           // de inicio no hay movimiento
int val = 0;                  // estado del pin

void setup()
{
   pinMode(LEDPin, OUTPUT);
   pinMode(PIRPin, INPUT);
   Serial.begin(9600);
}

void loop()
{
   val = digitalRead(PIRPin);
   if (val == HIGH)   //si está activado
   {
      digitalWrite(LEDPin, HIGH);  //LED ON
      if (pirState == LOW)  //si previamente estaba apagado
      {
        Serial.println("Sensor activado");
        pirState = HIGH;
      }
   }
   else   //si esta desactivado
   {
      digitalWrite(LEDPin, LOW); // LED OFF
      if (pirState == HIGH)  //si previamente estaba encendido
      {
        Serial.println("Sensor parado");
        pirState = LOW;
      }
   }
}
```

Sensor de velocidad:

Fuente de información: https://juegosrobotica.es/retos/reto-11/

Cuando hablamos de sensor de velocidad, nos referimos a él para describir cómo queremos que funcione nuestro “radar”; sin embargo, no vamos a emplear ningún sensor de velocidad, sino que hemos decidido utilizar en su lugar dos detecciones (ultrasónicos, infrarrojos, capacitivos, fotocélulas) separadas a una distancia concreta por las que pasará el coche. Cronometrando el tiempo que pasa entre una detección y otra, y siendo conocida la distancia, podemos calcular la velocidad. Así funcionan algunos radares de tramo y de bandas.

De esta forma, obtendremos el mismo resultado que con un sensor de velocidad.



La otra manera de resolver el reto del radar con Arduino sería medir cuánta distancia recorre el coche en un tiempo determinado. Mediante un sensor de ultrasonidos tomaremos dos medidas de distancia en un intervalo de tiempo concreto. Restando las distancias podemos saber la distancia recorrida en esa décima de segundo y por lo tanto calcular la velocidad.

Código en C++ :

```
/*Aqui se configuran los pines donde debemos conectar el sensor*/
#define TRIGGER_PIN  12
#define ECHO_PIN     11
#define MAX_DISTANCE 200 /Calibracion de distancia

/*Crear el objeto de la clase NewPing*/
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

void setup() {
  Serial.begin(9600);
}

void loop() {
  // Esperar 1 segundo entre mediciones
  delay(1000);
  // Obtener medicion de tiempo de viaje del sonido y guardar en variable uS
  int uS = sonar.ping_median();
  // Imprimir la distancia medida a la consola serial
  Serial.print("Distancia: ");
  // Calcular la distancia con base en una constante
  Serial.print(uS / US_ROUNDTRIP_CM);
  Serial.println("cm");
}
```

Con ese codigo obtienes distancia y tiempo. Ya solo es un depeje y tienes tu velocidad.