



NOMBRE DEL ALUMNO:

Everardo Estrella Rojo

CARRERA:

Ing. Mecatrónica

MATERIA:

Programación de Robots Industriales

GRADO Y GRUPO:

6°-B

CUATRIMESTRE:

Mayo - Agosto

NOMBRE DEL DOCENTE:

Carlos Enrique Morán Garabito

INDICE

INTRODUCCION.....	2
CAPITULO I.....	3
JUSTIFICACION.....	4
CAPITULO II.....	5
MARCO TEORICO.....	6
CAPITULO III.....	11
DESARROLLO.....	12
TABLA DE COSTOS.....	20
BIBLIOGRAFIAS.....	21

INTRODUCCION

Desde que apareció el automóvil, primero como medio de transporte de uso particular, fue necesario de un sitio fuera de la vía publica para guardar dicho vehículo, durante la mayor parte del día y durante la noche.

El proceso de cambio, en cuanto a dimensionamiento, diseño, funcionamiento y confort, represento la creación de un sin numero de vehículos de uso particular, de transporte de pasajeros y de carga, los cuales requieren el espacio necesario para maniobrar y estacionarse dentro y fuera del predio.

Esto ha creado especialidades, como la ingeniería automotriz que se encarga de su funcionamiento y confort; la ingeniería de transito que cuantifica la cantidad de vehículos que circulan por las calles.

Dicha problemática ha motivado a los compañeros del equipo para realizar a escala un estacionamiento automatizado, de manera funcional para que los autoabientes de la región en la cual se plante poner el estacionamiento sea mucho mas práctica de ser parte de la nueva tecnología que es la automatización de diferentes herramientas parte de la vida cotidiana.

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 JUSTIFICACION

El proyecto que se realizó sirve para hacer de manera eficaz el paso del carro habiente de la zona en la que se desee implementar, en este caso se optó por tomar como ejemplo la universidad politécnica de la zona metropolitana de Guadalajara. Dicho prospecto también es funcional en contabilizar los lugares ocupados y lugares desocupados.

1.2 META

Diseñar y automatizar un estacionamiento a escala para la Universidad politécnica de la zona metropolitana de Guadalajara.

1.3 OBJETIVOS

- Diseñar un estacionamiento a escala.
- Automatizar estacionamiento a escala.
- Controlar la entrada y salida de vehículos, así como también la cantidad de espacios vacíos y ocupados.

CAPITULO II

MARCO TEÓRICO

2 MARCO TEÓRICO

2.1 CONTROL DE ACCESO VEHICULAR

Los sistemas de control de accesos vehicular se implementan para tener el control de los vehículos permitidos y restringiendo a aquellos que no estén autorizados. Al integrar un sistema de control de accesos vehicular, podemos tener el control total, tanto de los estudiantes, maestros como de los visitantes.

2.2 VENTAJAS

Ahorro en personal extra dedicado a la vigilancia y control de acceso vehicular.

Mayor seguridad con los registros de entradas y salidas.

ingreso de automóviles de forma controlada y organizada.

Sistema automatizado mejorando el acceso vehicular.

Integración con todos los sistemas de seguridad para una gestión centralizada.

2.2 Barreras vehiculares automáticas

Las barreras de los estacionamientos se utilizan en integración con los controles de accesos vehicular para un correcto manejo del flujo vehicular en un determinado parqueadero. Su principal función se basa en permitir e impedir el paso a los vehículos, realizando la tarea de forma automática, eficiente, rápida y segura.

2.3 Desarrollo e investigación

Las Aplicaciones actuales de estos sistemas son el control de accesos y la inmovilización de vehículos. En el control de accesos se gana en comodidad, no es necesario el control físico de la tarjeta con el lector, lo que lo hace más cómodo y más rápido de usar.

2.4 Actuador

Un actuador es un dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado. Este recibe la orden de un regulador o controlador y en función a ella genera la orden para activar un elemento final de control como, por ejemplo, una válvula, existen varios tipos de actuadores como son:

- Electrónico
- Hidráulico
- Neumáticos
- Electrónico

Los actuadores hidráulicos, neumáticos y eléctricos son usados para manejar aparatos mecánicos. Por lo general, los actuadores hidráulicos se emplean cuando lo que se necesita es potencia, y los neumáticos son simples posicionamientos. Sin embargo, los hidráulicos requieren mucho equipo para suministro de energía, así como de mantenimiento periódico. Por otro lado, las aplicaciones de los modelos neumáticos también con limitadas desde el punto de vista de precisión y mantenimiento

El Arduino Uno es una placa de microcontrolador de código abierto basada en el microcontrolador Microchip ATmega328P y desarrollada por Arduino.cc . La placa está equipada con conjuntos de pines de entrada / salida (E / S) digitales y analógicas que pueden conectarse a varias placas de expansión (pantallas) y otros circuitos. El tablero tiene 14 pines digitales, analógicas, 6 pines y programable con el Arduino IDE (Integrated Development Environment) a través de un tipo B cable USB . Puede ser alimentado por el cable USB o por un externo Batería de 9 voltios , aunque acepta voltajes entre 7 y 20 voltios. También es similar al Arduino Nano y Leonardo.El diseño de referencia de hardware se distribuye bajo una licencia Creative Commons Attribution Share-Alike 2.5 y está disponible en el sitio web de Arduino. Los archivos de diseño y producción para algunas versiones del hardware también están disponibles.

La palabra "uno" tiene el mismo significado y fue elegida para marcar el lanzamiento inicial del software Arduino . La placa Uno es la primera de una serie de placas Arduino basadas en USB, y la versión del Arduino IDE fueron las versiones de referencia de Arduino, ahora evolucionadas a nuevas versiones.El ATmega328 en la placa viene preprogramado con un gestor de arranque que permite cargar un nuevo código sin el uso de un programador de hardware externo.



2.5 Barreras vehiculares

Las Barreras vehiculares son la solución ideal para controlar el tránsito de vehículos en áreas de acceso restringido, su campo de aplicación pasa desde accesos a fraccionamientos, parques industriales, edificios residenciales, edificios corporativos, universidades, estacionamientos, incluso a lugares de alta densidad de tráfico como eventos masivos entre otros.

Al ser integradas con sistemas de control de acceso podemos obtener importantes ventajas al facilitar el tráfico de automóviles previamente registrados facilitando así las actividades de los guardias de seguridad o personal de control.

En AR comercializamos e instalamos solo las barreras vehiculares de las mejores marcas del mercado, garantizando así que su inversión este protegida, todos los productos están respaldados por garantía de fabrica, la garantía por los trabajos de instalación va por nuestra cuenta.

2.6 Espacio de estacionamiento

La creciente cantidad de vehículos en las ciudades provoca una mayor necesidad de espacios de estacionamiento. Muchas edificaciones nuevas contemplan este problema y destinan un espacio amplio para la construcción de estacionamientos. Sin embargo, en el caso de edificaciones anteriores, cuyo diseño incluía pocos o nulos espacios de estacionamiento, representan un problema para los inquilinos que cuentan con vehículo.

No sólo se trata de conseguir espacios de estacionamiento, sino de garantizar el ahorro de espacio, los diseños innovadores y, sobre todo, la seguridad de los automóviles.


Una solución común que suelen llevar a cabo los conductores es estacionar sus autos en las calles, arriba o debajo de las aceras. Esto es un grave problema en varios aspectos. Por principio de cuentas se tiene el riesgo de que el vehículo sufra un accidente o sea robado. También se dificulta el tránsito tanto de otros vehículos como de los peatones.

2.7 Normatividad de los estacionamientos

Reservar cinco espacios de estacionamiento para el uso de bicicletas por cada 50 cajones o menos existentes en el estacionamiento público correspondiente, excediendo esta cantidad, habrá dos espacios por cada 50 cajones adicionales de estacionamiento hasta 500 cajones; excediendo esta cantidad, habrá un espacio por cada 100 cajones adicionales de estacionamiento. Para dar cumplimiento a lo anterior, se puede utilizar cualquier espacio disponible en el estacionamiento, preferentemente en la entrada y salida o en su caso, en el primer piso. Así como se

debe instalar señalización para advertir a los automovilistas que dentro del estacionamiento circulan ciclistas. (Reforma aprobada en sesión ordinaria celebrada el 16 de agosto de 2016 y publicada el 22 de agosto de 2016 en el Suplemento de la Gaceta Municipal) VII. Vigilar y controlar que los cajones de estacionamientos para personas con discapacidad, sean exclusivamente utilizados para ello, en caso de que un usuario sin necesidad ni derecho de usarlo lo utilice, deberán dar aviso a la Dirección de Movilidad y Transporte; VIII. Expedir un comprobante de ingreso por cada vehículo, con excepción de los estacionamientos privados sin cobro.

2.8 Tabla de algunos de los estacionamientos de Guadalajara

	Gobierno	Ciudad	Trámites y Servicios	Transparencia	
Alcalde 130				301 lugares	ver detalle
Alcalde 336				30 lugares	ver detalle
Aldama 1299				13 lugares	ver detalle
Aldama 53				55 lugares	ver detalle
Alvaro Obregon 1156				35 lugares	ver detalle
Alvaro Obregon 1275				14 lugares	ver detalle
Alvaro Obregon 280				249 lugares	ver detalle
Alvaro Obregon 30				57 lugares	ver detalle
Angulo 127				36 lugares	ver detalle
Angulo 1418				29 lugares	ver detalle
Angulo 1490				31 lugares	ver detalle
Antonio Bravo 31				29 lugares	ver detalle
Antonio Tello 134				35 lugares	ver detalle

2.9 Barreras de estacionamiento con Tag

Los módulos disponibles para este sistema se integran fácilmente según la arquitectura de cada estacionamiento, desde un simple ingreso con salida manual hasta un complejo sistema automatizado con múltiples vías de ingreso y egreso.

El sistema permite la integración con múltiples medios de pago a fin de facilitar el servicio a los clientes y usuarios, y mejorando así la escalabilidad del mismo: Caja Manual, Cajero Automático, Kiosco de Autopago, Tarjeta de Crédito, Cobros por billetera virtual, Celulares y Plataformas de pago vía Web y/o mediante el uso del Tag que se utiliza en la Red de Autopistas TELEPEAJE.

CAPITULO III

Desarrollo

DESARROLLO

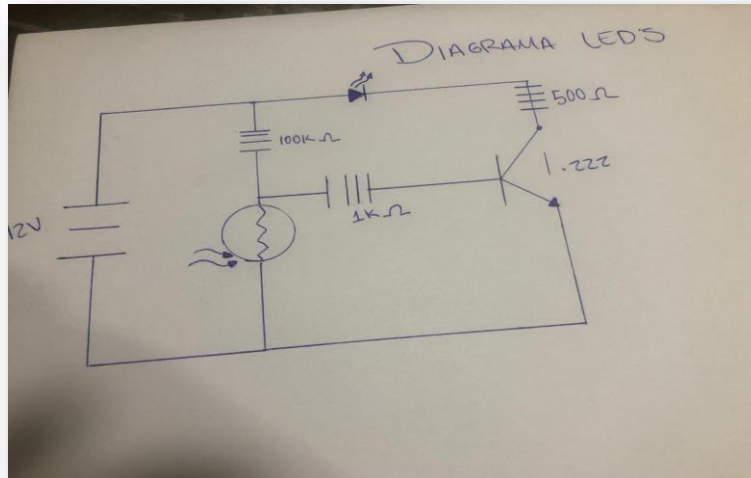


Figura 3.1.1 Diagrama LED

Se comenzó por realizar el diagrama de cada led mediante una LDR (fotorresistencia) conectado a un led cuando este interrumpa la entrada de luz prenda stand así de manera invertida. Ahí entra en función el componente n2222 un transistor.

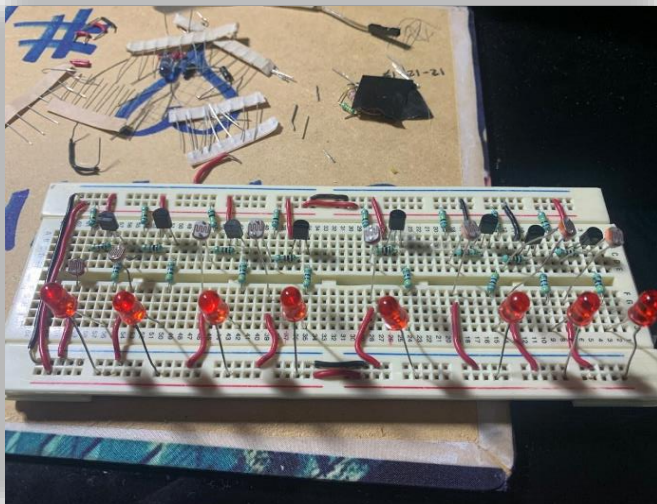


Figura 3.1.2 Armado de circuito de LEDS

Repitiendo esta condición para cada uno de los espacios del estacionamiento. Quedando de la como la figura 3.1.2 que se puede apreciar en la imagen.

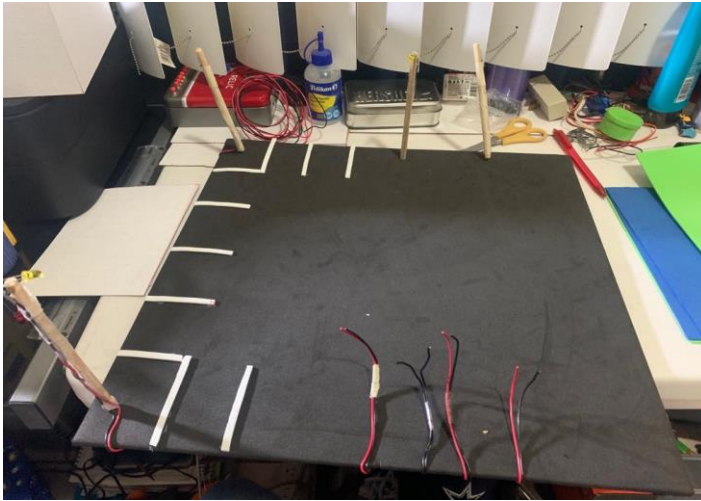


Figura 3.1.3 Cableado e inicio de modelo a escala

Por consecuente después de haber armado el circuito en físico, se procede a realizar el modelo a escala del estacionamiento, como se puede apreciar en la figura 3.1.3 cabe mencionar que el cableado para esta maqueta se encuentra en la parte de abajo.

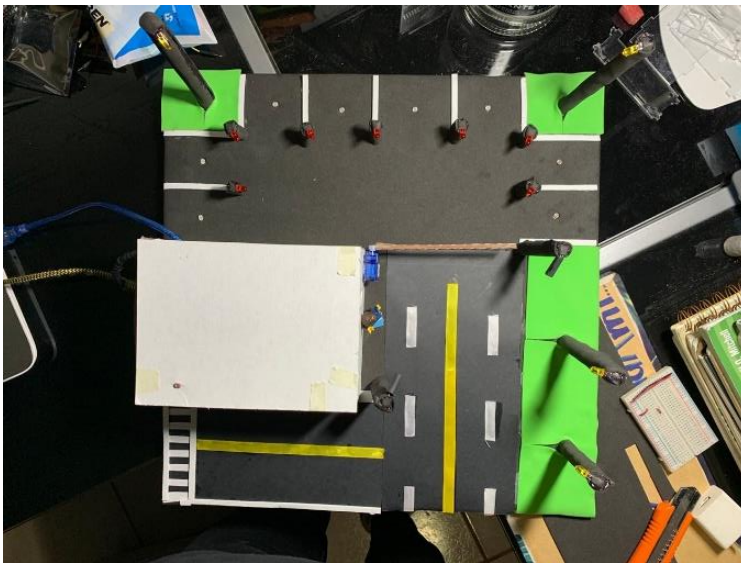


Figura 3.1.4 Maqueta terminada

Para el siguiente paso en el desarrollo del proyecto, y en específico, de la maqueta se colocaron adornos acordes al tema elegido y una caseta simulando la original de un guardia, se puede observar en la figura 3.1.4



3.1.5 Lugares inteligentes

En la figura 3.1.5 se aprecia mas a detalle que se colocó los LDR de forma que cuando el vehiculo cuando arribe al espacio libre o de preferencia, el led que esta conectado en serie al LDR apague, anunciando que este lugar se ocupo.

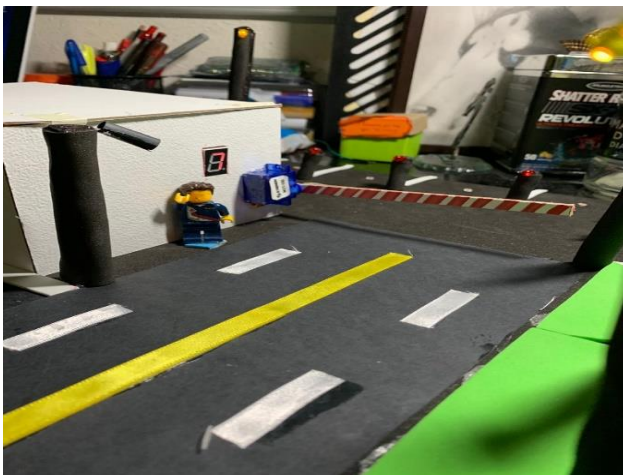


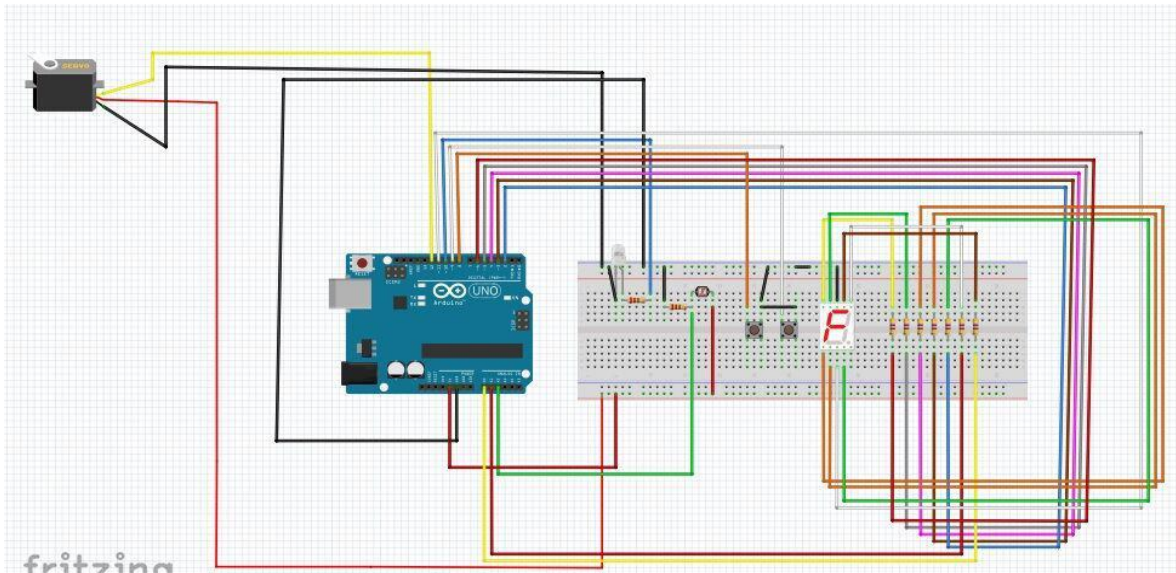
Figura 3.1.6 Barrera de estacionamiento automatizada

Por ultimo el funcionamiento de la barrera de estacionamiento automatizada, esta cuenta con dos LDR programado a controlador Arduino, dada la condición que cuando detecte luz (En este caso utilizado como tag) Esta abrá y deje pasar al conductor residente en este caso de la Universidad Politecnica de la zona Metropolitana de Guadalajara. Es misma condición para el otro caso, con una

detalle, cuando la pluma detecte Tag, y el conductor ocupe un lugar el display con su programación correspondiente esta suma o resta dependiendo cual sea la situación el display hará la conversión.

Como se puede observar en la imagen 3.1.7 se muestra la simulación en el programa Arduino, en conjunto con una Protoboard y un display de cátodo común conectado a resistencias en paralelo a cada una de las entradas de señal del display. Incluyendo también dos botones de paro.

Figura 3.1.7 Circuito simulado en Arduino



A continuación, se muestra el circuito armado en forma física quedando como se puede observar en la imagen 3.1.8 el circuito montado en una carcasa de cartón. Cabe mencionar que en lugar de colocar un par de botones normalmente cerrado se utilizó un switch normalmente abierto.

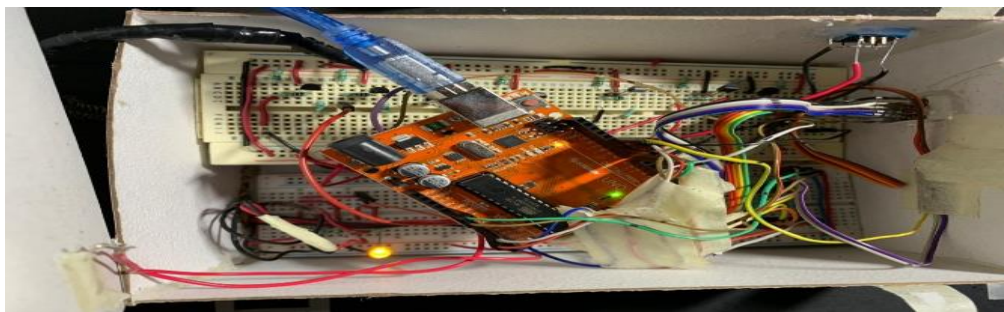


Figura 3.1.8 Circuito armado físicamente

3.1.9 Programación del Arduino

<code>#include <Servo.h></code>	<i>{Incluir la librería del servo motor}</i>
<code>Servo myservo; // creamos un objeto servo para controlar el servo</code>	<i>{Hacer una variable}</i>
<code>#define ServoM 12 // Pin conectado con el servomotor.</code>	<i>{El pin que se conecta al Arduino}</i>
<code>#define Brillo 11 // biblioteca servo, desactiva PWM de los pines 9 y 10.</code>	<i>{Desactiva el PWM}</i>
<code>#define luces 10 // Pin de salida a LEDs</code>	<i>{Pin de salida}</i>
<code>#define Salida 9 // Pin conectado al botón de salir.</code>	<i>{Botón de salida}</i>
<code>#define Entrada 8 // Pin conectado al botón de entrar.</code>	<i>{Pin de entrada}</i>
<code>#define rojo 7 // Pin semáforo rojo</code>	
<code>#define verde 13 // Pin semáforo verde</code>	
<code>#define analogica A2 // Pin del LDR</code>	<i>{Pin de la foto resistencia}</i>
<code>#define BarCer 177 // posicion barrera cerrada.</code>	
<code>#define BarAb 95 // posicion barera abierta.</code>	
<code>#define CAPACIDAD 8 // Capacidad del parking (coches).</code>	
<code>#define INTEN 80 // Intensidad del display en %.</code>	
<code>//Pines conexiones a los segmentos (catodos).</code>	

El comando `#include servo moto` es utilizado para agregar la librería del servo motor a la programación del Arduino, como tal es correcto decir que es la cabeza del código. Continúa definiendo las variables, que en este caso son las salidas del controlador, una para la LDR, y cada uno de los LED's. Por ultimo en los penúltimos párrafos se define la posición de la barrera de estacionamiento.

```
int valor = 0; // Variable recibida por el LDR.

#define segA A0
#define segB A1
#define segC 2
#define segD 3
#define segE 4
#define segF 5
#define segG 6

//Matriz con los segmentos para representar los numeros decimales del 0 al 8.
byte segments[9] = {
```

```
// pgfedcba <--- segmentos
```

```
B00111111, // number 0
```

```
B00000110, // number 1
```

```
B01011011, // number 2
```

```
B01001111, // number 3
```

```
B01100110, // number 4
```

```
B01101101, // number 5
```

```
B01111101, // number 6
```

```
B00000111, // number 7
```

```
B11111111, // number 8
```

```
};
```

Esta programación tiene que ver netamente con la energización del display, y como es un cátodo común la programación es negada en todos los casos definiendo el cátodo que se ilumina.

```
void setup(){
```

```
myservo.attach(ServoM);
```

```
pinMode(Salida, INPUT_PULLUP);
```

```
pinMode(Entrada, INPUT_PULLUP);
```

```
digitalWrite(Salida, HIGH);
```

```
digitalWrite(Entrada, HIGH);
```

```
pinMode(segA, OUTPUT);
```

```
pinMode(segB, OUTPUT);
```

```
pinMode(segC, OUTPUT);
```

```
pinMode(segD, OUTPUT);
```

```
pinMode(segE, OUTPUT);
```

```
pinMode(segF, OUTPUT);
```

```
pinMode(segG, OUTPUT);
```

```
pinMode(Brillo, OUTPUT);
```

```
myservo.write(BarCer); //Barrera en la posicion bajada
```

```
// delay(1000);
```

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

```
pinMode(luces, OUTPUT);
```

```
}
```

En esta parte de la programación se define la salida de los pines del Arduino, esta condición se da por un pequeño truco en la programación, el uso de “void” es una estructura que se manda a llamar dando de alta la variable.

```
int Available= 8;          // Numero de plazas disponibles.
```

```
-
```

```
void loop(){
```

```
    Display(Available);
```

```
    if (!digitalRead(Entrada)) {
```

```
        Serial.println("Entrada");
```

```
        if(Available != 0){
```

```
            Available--;
```

```
            Serial.println(Available);
```

```
            myservo.write(BarAb);
```

```
            delay(3000);
```

```
            myservo.write(BarCer);
```

```
        }
```

```
    }
```

```
    if (!digitalRead(Salida)) {
```

```
        Serial.println("Salida");
```

```
        if (Available != CAPACIDAD) {
```

```
            Available++;
```

```
            Serial.println(Available);
```

```
            myservo.write(BarAb);
```

```
            delay(3000);
```

```
            myservo.write(BarCer);
```

```
        }
```

```
    }
```

```
    valor = analogRead(analogica); // El valor recibido por el LDR lo guardamos en la variable.
```

```
    //println(valor); // Valor será mostrado en Serial Monitor, para calibrar el encendido del LED de iluminación
```

```
    delay(250); // Pausa de 250 ms.
```

En este apartado de la programación se da principalmente el tiempo de espera que tiene el servo motor para ascender y descender la pluma, la herramienta “delay” es un comando que marca el retraso en el proceso que hace el motor (en este caso el tiempo de espera) se le dio 250 milisegundos.







```
if (valor < 150) {  
    digitalWrite(luces, HIGH);  
} else{  
    digitalWrite(luces, LOW);  
}  
}  
  
// Pone los segmentos de acuerdo al número de plazas libres.  
  
void Display(int number){  
    byte segs = segments[number];    //"" se usa para el anodo comun.  
  
    digitalWrite(segA, bitRead(segs, 0) );  
    digitalWrite(segB, bitRead(segs, 1) );  
    digitalWrite(segC, bitRead(segs, 2) );  
    digitalWrite(segD, bitRead(segs, 3) );  
    digitalWrite(segE, bitRead(segs, 4) );  
    digitalWrite(segF, bitRead(segs, 5) );  
    digitalWrite(segG, bitRead(segs, 6) );  
}
```

La herramienta if, elif, else es un bucle para repetir la condición una y otra vez.

3.2 Tabla de costos

Materiales	Piezas	Costo
Arduino Uno	1	350.00 \$
Transistores	8	45.00 \$
Leds	16	80\$
Maqueta y artículos	Indefinido	125\$
Resistencias	36	54\$
Servomotor	1	90\$
Ldr	11	132\$
Display	1	54.50\$
Alambre	9 metros	30\$
Protoboard	2	190\$
Switch	1	25\$
Jumper	200	156\$
	Subtotal	1,331.5 \$
	Iva (1.6%)	213.04 \$
	Total	1,543.54 \$

BIBLIOGRAFÍAS

-  <https://acsys.mx/control-de-acceso-vehicular/> Control de acceso vehicular
-  <https://www.samson.d> Accionamientos
-  <https://forum.arduino.cc/index.php?board=118.0> Información de Arduino y programación
-  [*"Arduino - ArduinoBoardDiecimila"*](#) . *www.arduino.cc . Consultado el 20 de febrero de 2018 .*
-  [*"Arduino - ArduinoBoardDuemilanove"*](#) . *www.arduino.cc . Consultado el 20 de febrero de 2018 .*
-  http://gdl.dimep.mx/controlstacionamientos.html?gclid=Cj0KCQjw4s7qBRCzARIsAlmcAxaGRMsOP_c84aCJdbQcHHd2mRNlozxhFI7OoVL-g9zp_9IDf5zfzpwAAlSyEALw_wcB