Proyecto:

Controlador de semáforos con prevención de congestión vial.

Jafet Soto arrieta - B77543.

Resumen.

Se quiere elaborar un controlador de 4 cruces generados por 4 calles, de 2 carriles cada calle, que sea capaz de prevenir la congestión vehicular en la zona de los cruces.

Objetivo general.

Implementar un controlador de tráfico vehicular para un sistema de 4 cruces que permita evitar la congestiòn vial .

Objetivos específicos.

- Crear un sistema que simule los 16 semáforos de los 4 cruces vehiculares.
- Detectar y prevenir la congestión vehicular según los cambios en el sistema.
- Implementar un algoritmo de detección de patrones para las calles del sistema (HoltWinters).

Justificación:

- **Problema:** La congestión vehicular afecta la movilidad y el tiempo de las personas.
- **Solución:** Un controlador inteligente de tráfico mejora el flujo vehicular.
- **Impacto:** Reduce tiempos de espera, mejora la eficiencia y la seguridad en los cruces.

Arduino UNO R3 :

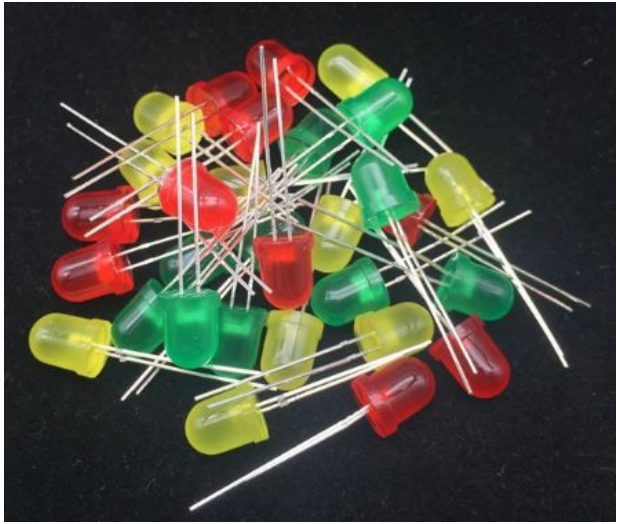
Placa de desarrollo de código abierto basada en el microcontrolador ATmega328P

Modulo RGB y 30 LEDs:

Un led RGB y 30 LEDs conforman los 16 semáforo

Modulo RGB y 30 LEDs:

Un led RGB y 30 LEDs conforman los 16 semáforo

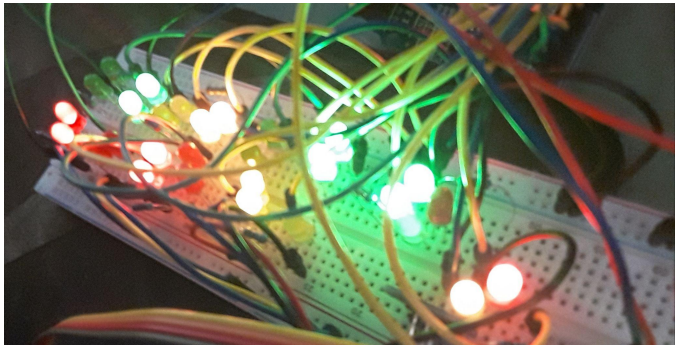


Pantalla LCD:

El estado de las pruebas se puede ver en un pantalla externa.

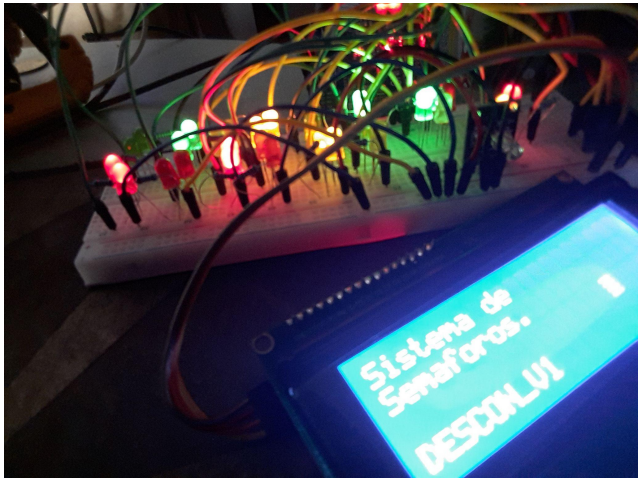
Desarrollo y análisis:

Distribución de LEDs

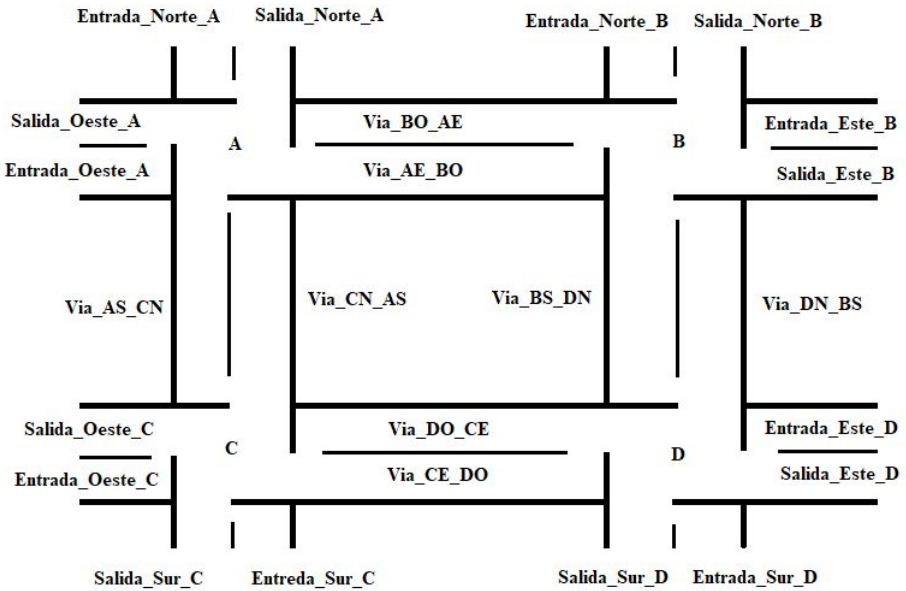


Desarrollo y análisis:

Diseño final del hardware

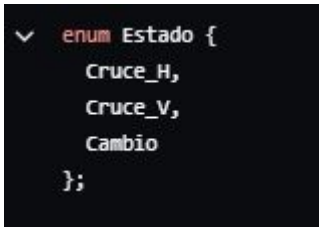


Desarrollo y análisis:



Desarrollo y análisis:

Class Semaforo y FSM_Semaforo



Desarrollo y análisis:

Cruces_FSM()

```
153  enum S_CONGEST{
154      IDLE,
155      WARNING,
156      ALERT,
157      DESCON_H1,
158      DESCON_H2,
159      DESCON_V1,
160      DESCON_V2
161  };
162
```

Desarrollo y análisis:

Modo de operación normal para semaforos.

```
260  void controladorIdle() {
261      // Primero fase: Cruces Horizontales (Cruce_H)
262      estadoA = Cruce_H;
263      estadoB = Cruce_H;
264      estadoC = Cruce_H;
265      estadoD = Cruce_H;
266
267      semaforoA.FSM_Semaforo(estadoA);
268      semaforoB.FSM_Semaforo(estadoB);
269      semaforoC.FSM_Semaforo(estadoC);
270      semaforoD.FSM_Semaforo(estadoD);
271
272      delay(5000);
```

Conclusiones:

- Se implementa un circuito capaz de simular los 4 cruces.
- Se agregan funciones y métodos para detectar y atender la congestión vial.
- No se logró implementar el método de HoltWinters en el proyecto.

Observaciones:

-Se puede modificar el último objetivo, y sustituirlo por una implementación en un dashboard de ThinksBoard que mejore la visualización de la simulación.

-A pesar de que el sistema tiene algunas condiciones de ejecución su implementación, en un sistema real, no debería ser muy compleja

Recomendaciones:

-Si se desea mejorar el proyecto se recomienda aumentar los casos de prueba.

-Además se puede verificar el código de la máquina de estados para corregir el diley óptimo.

¡Gracias por su atención!

Bibliografía y anexos:

Arduino UNO R3:https://www.microjpm.com/products/arduino-uno-r3-version-generica/

Bibliotec para LCD:
https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library/blob/master/examples/HelloWorld/HelloWorld.ino