**CleanVoltage**

**Proyecto Zebras**

**Area Electrónica**

**Resumen**

**ESTADO ACTUAL:**

**Hardware:**

1.- Hoy en dia se cuenta con una única placa que conglomera tanto la etapa de potencia como la de control, por lo que, existe ruido en la alimetación del hardware encargado del control.

2.- Además por la disposición de los componentes se dificulta la colocación de disipadores de calor en la etapa de potencia, generando dos fenomenos no menores: primero, la operación optima del hardware hoy utilizado se alcanza a temperaturas no mayores a los 75°C, por lo que, si se supera este umbral el circuito comienza a fallar produciendo cambios en la tonalidad de los leds; segundo, genera ruido térmico que altera los valores resistivos y carateristicas electricas de los componentes de hardware de control lo que llevaria a un comportamiento errático del sistema.

3.- Debido a los constantes cambios realizados sobre la marcha es que la placa presenta ingertos y conexiones no estandar, las que aumenta la probabilidad de error en el montaje de conectores y la probabilidad de fallo en pleno funcionamiento.

4.- Los sensores no cuentan con la adecuada aislación a fuentes de ruido externas al sistema, como por ejemplo, ruido electromagnético. Este efecto se hace más notorio en la medida de que los cables sean más largos, ya que hacen una suerte de antena.

5.- El sistema cuenta con un solo sensor laser reflex, que nos permite detectar la presenscia de un peatón de forma unívoca, sin embargo, con este hardware, no es posible saber si el peatón esta ingresando a cruce peatonal o saliendo de éste.

6.- El sistema de control está compuesto por 3 arduinos pro mini, cada uno encargado de una tarea distinta. Esto debido a las bajas prestaciones que ofrece esta plataforma. Esto tiene una ventaja y una desventaja evidente: ventaja, cada arduino, al estar encargados de tareas distintas, ejecuta su rutina de forma paralela; desventaja, al ser 3 arduinos ocupan mucho espacio y además es necesario sincronizarlos mediante un protocolo de comunicación.

**Firmware:**

1.- Se encuentra distribuido en 3 microcontroladores: uno encargado de un panel; otro encargado del otro panel y el reprodutor mp3; y el último encargado de las cintas, las coronas y de coordinar el funcionamiento de los 3 en conjunto. Además cabe mensionar que existe un Firmware distinto para este último microcontrolador, que depende de si se encarga de sincronizar ambos postes o si sólo espera las órdenes de sincronización. A éstos se les distingue como Master and Slave (Maestro y Esclavo).

2.- La lógica implementada en este control se describe a continuación:

**a) MASTER:**

1.- Al alimentar el sistema existe un transiente inestable que dura alrededor de 2 segundos, para luego estabilizarse y sincronizarse. (Se recomienda encender primero al SLAVE, el cual quedará a la espera de las instrucciones del MASTER).

2.- Mientras no exista una detección de peatón, el sistema ejecuta una rutina de luces que tienen un período no menor a los 2 segundos y al mismo tiempo enviará mensajes de sincronización al SLAVE, por lo que, ambos postes ejecutarán la misma rutina de forma sincronizada.

3.- Al ocurrir una detección mientras se encuentra en el estado descrito en el punto anterior, el sistema emitirá una recomendación al peatón y cambiará a una rutina de luces de frecuencia más alta durante los 15 segundos siguientes. Adicionalmente se enviará un mensaje al SLAVE para advertir la presencia de un peatón.

Cabe destacar que dependiendo de si es de día o noche, se despliega una animación de peatón esperando o caminando en los paneles LED, en función de si es seguro cruzar o no.

4.- Si dentro de los 15 segundos mencionados en el punto anterior ocurre otra detección, el ciclo se prolongará por 15 segundos más, sin embargo, no se emitirá el sonido con la recomendación. En caso de que ocurran detecciones sucesivas durante estos segundos cada una de ellas reseteará el tiempo de la rutina, es decir, la rutina terminará 15 segundos despues de la última detección, al mismo tiempo que se enviará al SLAVE un mensaje para que resetee su tiempo de ejecución de la rutina.

5.- Una vez terminada la rutina del punto anterior, se vuelve a ejecutar lo explicitado en el punto 2.

**b) SLAVE:**

1.- Idem al MASTER.

2.- Mientras no exista una detección de peatón, el sistema ejecuta una rutina de luces que tienen un período no menor a los 2 segundos que estará determinada por mensajes de sincronización desde el MASTER.

3.- Idem al MASTER.

4.- Idem al MASTER.

5.- Idem al MASTER.

**CONSIDERACIONES**:

- Como no podemos saber si el peatón esta ingresando o saliendo del cruce peatonal, éste será detectado tanto al ingresar como al salir del mismo. Por ello, en el peor caso para un solo peatón la rutina completa durará 30 segundos. Para el caso de un cruce masivo de peatones, el sistema mantendrá la rutina durante 15 segundos despues de que el último peatón termine de cruzar. Ya que el sensor genera una deteccion por cada corte del haz de luz, se nos hace casi imposible saber con certeza si sólo cruzó una persona o varias (el sensor está dispuesto a la altura de las pantorrillas por lo que una persona corta mas de una vez el laser).

**REDISEÑO:**

**HARDWARE:**

1.- Se propone separar eléctricamente las etapas de control y potencia. Para ello se diseñará una única placa con 4 capas donde las señales de control hacia la etapa de potencia estén optoacopladas.

2.- Para acotar el espacio del circuito de control se propone unificar todo en un único controlador con mayor potencia de procesamiento, memoria y GPIO.

3.- Se distribuirán de mejor manera los CI de potencia y se unirán todos a un gran disipador de calor.

4.- Se agregarán al diseño de la placa los últimos cambios efectuados a las conexiones de los periféricos.

5.- Se cambiará el tipo de conexionado de los periféricos a unos conectores de acople rápido.

6.- Se cambiarán todos los cables de periféricos por unos cables apantallados para disminuir las funtes de ruido y se tomarán las medidas necesarias para filtrar el ruido electrico en la alimentación de cada uno de los circuitos.

7.- Se agregará una forma de programación inalámrica.

**FIRMWARE:**

1.- Mejorar la fluidez de las rutinas.

2.- Mejorar el protocolo de comunicación.

3.- Implementar máquinas de estado.

4.- Optimizar el uso de recursos.

5.- Programación inalámbrica.

6.- Implementar métodos de detección de errores.

PLAZO DE FIRMWARE: 28 DÍAS HABILES.-

1.- Equematización de Sistema completo (MAESTRO-ESCLAVO) y verificar nuevas prestaciones del sistema. (1 día)

2.- Creación de un nuevo Protocolo de comunicación. (1 día)

3.- Creación de las máquinas de estado que sean necesarias (diagramación) y los diagramas de flujo de cada rutina que sea necesaria. (2 días)

4.- Elección del nuevo microcontrolador en función de lo resuelto en los puntos anteriores. (1 día)

5.- Busqueda de bibliotecas que resuelvan el problema de forma eficiente y sensilla. (1 día)

6.- Implementación en código de las rutinas necesarias. (5 días)

7.- Pruebas y correcciones. (2 días)

8.- Implementación en código de las Maquinas de estado. (3 dias)

9.- Pruebas y correcciones. (2 dias)

10.- Integración en código de todas las partes del sistema. ( 5 dias)

11.- Pruebas y Correcciones. (3 dias)

12.- Depuración de Código Final. (2 dias)

PLAZO DE HARDWARE: 48 DÍAS CORRIDOS.

1.- Esquematización del sistema completo. (1 dia)

2.- Definición del hardware a utilizar. (1 dia)

3.- Prototipado. (5 dias)

4.- Correcciones y Mejoras. (3 dias)

5.- Diseño de PCB (10 dias)

6.- Fabricación PCB (8 dias)

7.- Montaje. (8 dias)

8.- Internación. (7 dias)

9.- Integración. (2 dias)

10.- Validación. (3 dias)

**PLAZO TOTAL : 48 DÍAS CORRIDOS.**

**PLAZO ESTIMADO DE ENTREGA: 28 de febrero del 2018.**