

UNIVERSIDAD RICARDO PALMA

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

**“CONTROL DE TRÁFICO VEHICULAR
AUTOMATIZADO UTILIZANDO LÓGICA
DIFUSA”**



TESIS

PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO
ELECTRÓNICO

PRESENTADO POR:

RUIZ DE SOMOCURCIO SALAS, ALVARO ENRIQUE

LIMA – PERÚ

2008

Para todas aquellas personas que confiaron en mí y que siempre supieron apoyarme y darme ánimos para continuar: a mis papás por su apoyo incondicional; a mis amigos de la universidad, en especial a Víctor, Manolo, Emilio, Jaime y Sandra, a los cuales siempre tengo presente; a Tatiana, por estar ahí para aconsejarme cuando más lo necesitaba; al Ingeniero Pedro Huamaní, por confiar en mi trabajo y por ser un gran profesor

TABLA DE CONTENIDO

INTRODUCCIÓN	4
DESCRIPCIÓN DEL SISTEMA	20
PRUEBAS Y VERIFICACIÓN DE DATOS OBTENIDOS	50
CONCLUSIONES Y PERSPECTIVAS	64
REFERENCIAS BIBLIOGRÁFICAS	67
APÉNDICES	69

INTRODUCCION

El presente trabajo de investigación busca mostrar una alternativa a los sistemas de control de tráfico vehicular existentes, integrando teorías y tecnologías existentes con el fin de poder desarrollar un sistema que sea más eficiente e implementable de una manera segura y confiable. En este caso se ha elegido utilizar la teoría de la Lógica Difusa para poder llevar a cabo esta investigación, ya que es un contexto que se amolda de muy buena manera con la problemática a tratar.

El tráfico vehicular es un problema que se presenta ante todos en la vida cotidiana. Por ejemplo: niños que asisten a sus escuelas utilizando el medio de transporte público, movilizadas por sus padres o por movilizaciones escolares; adolescentes que asisten a sus respectivas universidades, personas que se dirigen diariamente a sus centros de trabajo, así como personas que se movilizan a la ciudad por diversas circunstancias.

El tráfico vehicular y el congestionamiento que este conlleva afecta a todos los ciudadanos, no solo a aquellos que se movilizan a pie, sino también a aquellos que utilizan un medio no motorizado. Por lo tanto, siempre se hará imprescindible cruzar una avenida para poder llegar al destino final.

Muchas de las alternativas actuales al control del tráfico vehicular son bastante costosas, ello se debe a que requieren de equipos especializados o centrales de control muy bien equipadas para poder manejar volúmenes grandes de tráfico. Por ello, da origen a que los costos administrativos y operativos sean bastante altos y, en muchos casos, la necesidad de contar con un sistema tan complejo no es una solución apropiada para el problema, pudiendo llegarse a sobredimensionar.

De esta manera, el contenido que se presenta a continuación muestra como se ha estado llevando a cabo esta problemática a lo largo de los años y se muestra una alternativa que puede aprovecharse bajo el contexto en el que se vive en muchas ciudades en donde el parque automotor rebasó los cálculos de tráfico vehicular que se realizaron a la hora de

diseñar una avenida y sus respectivas intersecciones, provocando el colapso de dichas intersecciones, volviéndolas peligrosas para los transeúntes y conductores que circulan por dichas vías.

El desarrollo de este proyecto permitirá reducir los tiempos de espera de los conductores, optimizando las temporizaciones de los semáforos de tal manera que los periodos asignados a cada vía sean los más apropiados para el cruce de la cantidad de vehículos en espera detectados por el sistema, además de poseer la visualización del tiempo para poder reducir el estrés en los conductores.

Asimismo, se puede obtener una estadística del congestionamiento vehicular de las diversas horas del día, mediante el uso de una aplicación en una computadora para el almacenamiento de la información y luego poder ser trabajada y publicada, para que los conductores puedan utilizar rutas alternativas en el caso de presentarse demasiado congestionamiento en la avenida controlada.

De esta manera, las municipalidades pueden utilizar este sistema en calles y avenidas en donde el problema del congestionamiento es debido a semáforos mal programados, o cuya programación no hace óptimo el tránsito fluido de vehículos por ser una programación de tipo estática.

1.1 Historia de los sistemas de semaforización

El **tránsito vehicular** (también llamado **tráfico vehicular**) es el fenómeno causado por el flujo de vehículos en una vía, calle, avenida o carretera. Presenta también muchas similitudes con otros fenómenos como el flujo de partículas (líquidos, gases o sólidos), flujo de peatones, entre otros.

En español no existe la diferenciación que se hace en inglés entre las palabras "tránsito" y "tráfico". En inglés, la primera ("transit") se refiere exclusivamente a lo que en español puede llamarse "transporte público", mientras que la segunda ("tráfico") es aproximadamente igual a "tráfico vehicular" o "tránsito vehicular"[1].

En las grandes urbes, como en el caso de Lima y algunas provincias del país, el tráfico vehicular se encuentra presente en casi todas las esferas de la actividad diaria de la gente, y ocasiona numerosos fenómenos entre los que destacan especialmente los congestionamientos, además de ciertos problemas con respecto al comportamiento de las personas bajo entornos de alto estrés presentes en zonas de alto congestionamiento.

El dispositivo de control de tránsito vehicular conocido como semáforo, tiene sus orígenes en el año 1868 en el Reino Unido y fue creado por el Ingeniero J.P. Knight. Este dispositivo inicialmente utilizaba en la noche luces de color rojo y verde, las cuales utilizaban gas como uno de sus componentes, y emitía dos zumbidos que indicaban la disponibilidad del tráfico de la calle, sin embargo este semáforo no estuvo mucho tiempo en funcionamiento debido a que era peligroso. Un 02 de Enero de 1869, hirió a un policía, que lo operaba, debido a una explosión del mecanismo. El semáforo retorna con la aparición del automóvil, y un 05 de Agosto de 1914 se instala el primer semáforo en Cleveland, Ohio en Estados Unidos. El inventor de este dispositivo fue el policía Lester Wire, fue quién inventó el primer semáforo eléctrico. Este dispositivo tenía luces rojas y verdes colocadas en un soporte en forma de brazo, y al igual que su antecesor emitía zumbidos [2].

El primer sistema interconectado de semáforos fue instalado en Salt Lake City en 1917, en donde seis intersecciones interconectadas eran controladas por un interruptor manual. Los sistemas de semáforos automáticos interconectados fueron introducidos en 1922 en Houston, Texas [2].

En Estados Unidos los controladores son estandarizados por NEMA, quienes crean los estándares para los conectores, los límites de operación y los intervalos. El estándar TS-1 fue introducido en 1976 para la primera generación de controladores de estado sólido [2]. Actualmente, no hay ninguna organización a nivel mundial que estandarice el uso de estos dispositivos, por esto que cada país se amolda de acuerdo a sus necesidades.

1.2 Sistemas Actuales de control de Tráfico Vehicular

En la actualidad, los sistemas de control de tráfico vehicular existentes son muy variados; tal es el caso de los sistemas convencionales que trabajan en base a un controlador simple, en el cual se programan los tiempos de operación en la intersección, según el nivel de tráfico estimado previamente.

Otro sistema, que también es muy utilizado es aquel con múltiples programaciones, llegando a manejar mejor algunas situaciones de alto y bajo tráfico, y a la vez alcanzando hasta un total de 8 programaciones, las cuales pueden ser incluidas en una memoria que puede ser accedida mediante interruptores, o como también siendo programables mediante el uso de algún software especial. Seguidamente, la figura 1 muestra una fotografía de un Controlador de Tráfico de la compañía Trelec modelo 5GD.

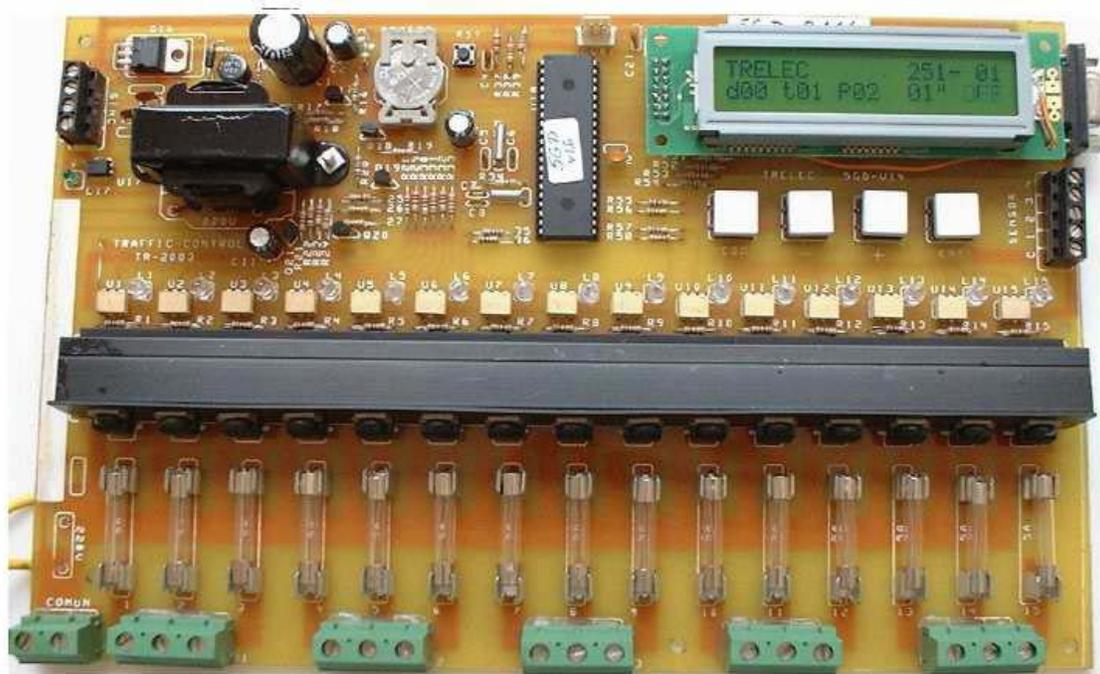


Fig.1 Controlador Trelec 5GD

Por otro lado, un sistema más complejo es aquel que se interconecta con un servidor para poder administrar el tráfico vehicular mediante una central de control, en la que los operarios verifican el estado del tránsito y toman las acciones correspondientes dependiendo de la situación en la que se encuentra la avenida en cuestión. Estos sistemas

son más complejos porque la infraestructura a utilizarse es mucho mayor; asimismo, la cantidad de equipos a controlar conforman una red más compleja, llegándose así a controlar avenidas completas, en donde es posible emplear el concepto de ola verdes; es decir, acción en donde se asigna prioridad a una avenida principal a través de una sincronización de todas las luces, de tal forma que el tráfico sea lo más fluido posible en dicha avenida.

1.3 Situación del Tráfico Vehicular en el Perú

En nuestros días, muchos de los sistemas de control de tránsito instalados en las principales avenidas de la capital y de muchas provincias presentan demasiadas deficiencias. Ello se debe a que son sistemas netamente de conmutación, los cuales son programados desde un inicio sin contemplar los cambios de flujo de vehículos en las diversas horas del día. Es por tal razón que se producen embotellamientos cuando no deberían ocurrir; o, hacen esperar al conductor más de la cuenta cuando en la intersección no se encuentra mayor cantidad de vehículos.

Otro problema que surge a raíz de la deficiencia de estos sistemas, es la necesidad de contar con personal policial para poder cubrir estas deficiencias en horas punta, cuando sería mejor poder asignar al personal policial a labores más de seguridad y de control de infractores, en lugar del control de tránsito de vehículos netamente.

Un siguiente factor es referido a los conductores. Pues, muchas veces no se cuenta con la información respecto a los tiempos asignados a los cambios de luz, lo cual origina una gran desesperación de querer cruzar antes del cambio de este, ocasionando accidentes de tránsito que se pueden evitar si se tuviera muy claro el tiempo restante para poder pasar por un cruce.

Otro factor determinante es el congestionamiento de vehículos, el cual se debe a la irresponsabilidad de los conductores de vehículos de transporte público, los cuales utilizan la parada de autobús en cualquier sitio que les crea conveniente, con tal de poder hacer subir a un pasajero más, cosa que se podría evitar con una mejor distribución del personal policial en las avenidas donde este problema sea crítico.

Actualmente los sistemas de control de tráfico vehicular instalados en el país son los mismos que los de hace más de 40 años, por lo que se han vuelto obsoletos y requieren ser renovados con urgencia debido a que el parque automotor sigue aumentando descontroladamente, y cada día se hace menos soportable el congestionamiento que se produce en las principales vías de la capital y de algunas ciudades del interior del país.

En el Perú son pocas las empresas que se dedican a este rubro. Dentro de estas empresas existe TEC Corporation, que se dedica a la instalación de sistemas de control de tránsito vehicular. Estos sistemas no son automatizados, sino mas bien cuentan con un centro de monitoreo donde existen personas encargadas de supervisar el correcto control del tránsito, lo que genera un gasto operativo muy alto.

Asimismo, esta empresa trabaja con los controles electrónicos de tránsito CT-3A, CT- 3B, CT-4B, CT-4PLUS, CT-800D, CT-800I, los cuales se programan con supervisión constante mediante un cuarto de control, el cual genera un gasto extra por requerir personal especializado durante el día para su correcto funcionamiento.

1.4 Teoría de la Lógica Difusa

La lógica difusa se basa en lo relativo de lo observado. Este tipo de lógica toma dos valores aleatorios, pero contextualizados y referidos entre sí. Así, por ejemplo, una persona que mida 2 metros es claramente una persona alta, si previamente se ha tomado el valor de persona baja y se ha establecido en 1 metro. Ambos valores están contextualizados a personas y referidos a una medida métrica lineal.

Esta simple idea nació en un artículo de Lofti A. Zadeh publicado en 1965 y titulado "Fuzzy Sets" (*Conjuntos Difusos*) [22]. La lógica difusa permite representar de forma matemática conceptos o conjuntos imprecisos, tales como días fríos, meses calurosos, personas altas, salarios bajos, guisos con mucho condimento, profesores poco valorados, etc.

Pero hay que tener en cuenta que la idea en sí de que las cosas no son blancas o negras, sino que existen infinitos matices de grises viene ya desde la época de los primeros grandes filósofos como Platón.

Posteriormente a muchos filósofos, otros grandes pensadores como David Hume o Kant apoyaban esta idea manteniendo que el razonamiento venía dado por las observaciones de las que somos testigos a lo largo de nuestra vida y la detección de algunos principios contradictorios en la lógica clásica.

Tras la publicación de Lotfi A. Zadeh, se comenzó rápidamente a usar la lógica difusa en distintas aplicaciones prácticas, llegando a su máximo auge a principios de los años 90, y continuando éste hasta la época actual, tomándose en cuenta mucho la aplicación en sistemas industriales donde las variables no son fácilmente representables matemáticamente, así como el tráfico vehicular no puede ser definido matemáticamente de manera exacta por depender mucho del comportamiento de las personas, el cual difiere mucho una respecto a la otra.

Un conjunto difuso es una función que asocia a cada objeto del universo un valor en el intervalo $[0,1]$. Si “ x ” es un objeto en el universo “ y ”, entonces “ $y = C(x)$ ” es el valor asociado a x , y se dice que “ y ” es el **grado de pertenencia** del objeto “ x ” al conjunto difuso “ C ”.

En el sistema desarrollado de este trabajo de tesis, se ha trabajado con conjuntos difusos de forma triangular, debido a las limitaciones de memoria en cuanto al procesamiento del microcontrolador. La figura 2 muestra dos ejemplos de conjuntos difusos triangulares.

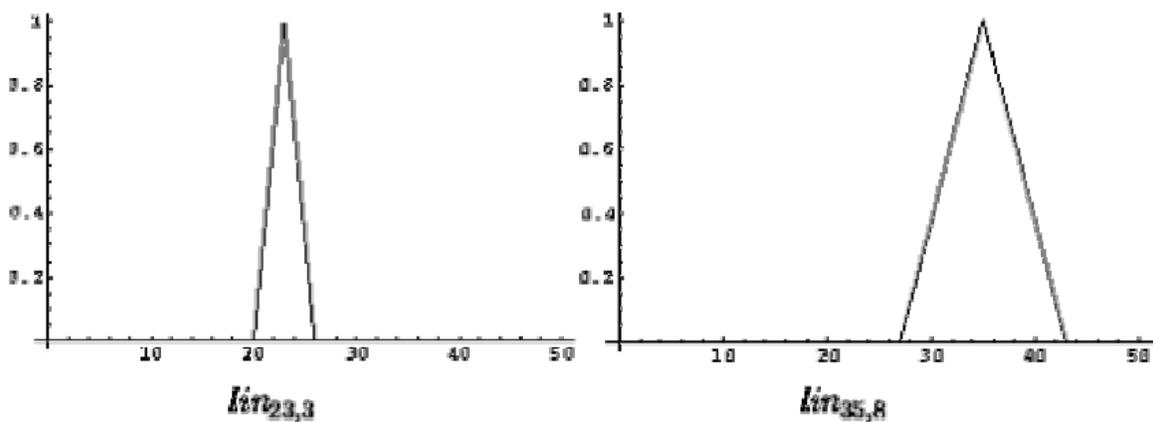


Fig.2 Conjuntos difusos triangulares

Para las operaciones entre los conjuntos difusos y todo el procedimiento, se ha requerido de los siguientes cálculos:

1. En primer lugar, se tiene el cálculo de los grados de pertenencia de un valor asociado a un conjunto difuso determinado. A continuación se ve un ejemplo en la ecuación 01.

$$\text{"contribuyentes mayores"}(x) = \begin{cases} 1 & \text{si } i(x) \geq 10^6, \\ \frac{i(x)-10^4}{99 \cdot 10^4} & \text{si } 10^4 \leq i(x) \leq 10^6, \\ 0 & \text{en cualquier otro caso,} \end{cases} \quad (01)$$

2. Un segundo paso es comparar los grados de pertenencia de dos conjuntos difusos y seleccionar el menor valor dentro de los tabulados.
3. En tercer lugar se procede a la operación de defuzzificación, la cual se consigue tomando los máximos de los valores obtenidos superpuestos en los conjuntos de salida, de los conjuntos difusos asociados. De esta manera, se forma un conjunto final de salida. A continuación la ecuación 02 muestra lo comentado.

$$A(x_A) = \text{Max}\{A(x)|x \in X\} \quad (02)$$

4. Finalmente, se procede al cálculo del centroide basado en el criterio mostrado en la ecuación 03. Una explicación más detallada del funcionamiento del sistema se muestra a continuación.

$$X_C = (x_{C,1}, \dots, x_{C,n}) \quad x_{C,i} = \frac{\int_R x_i d^n \mathbf{X}}{\int_R d^n \mathbf{X}} \quad (03)$$

El modelo trabajado e implementado en este informe de tesis, está basado en el Modelo del tipo Mamdani, y cuya configuración básica es la que se muestra en la figura 3.

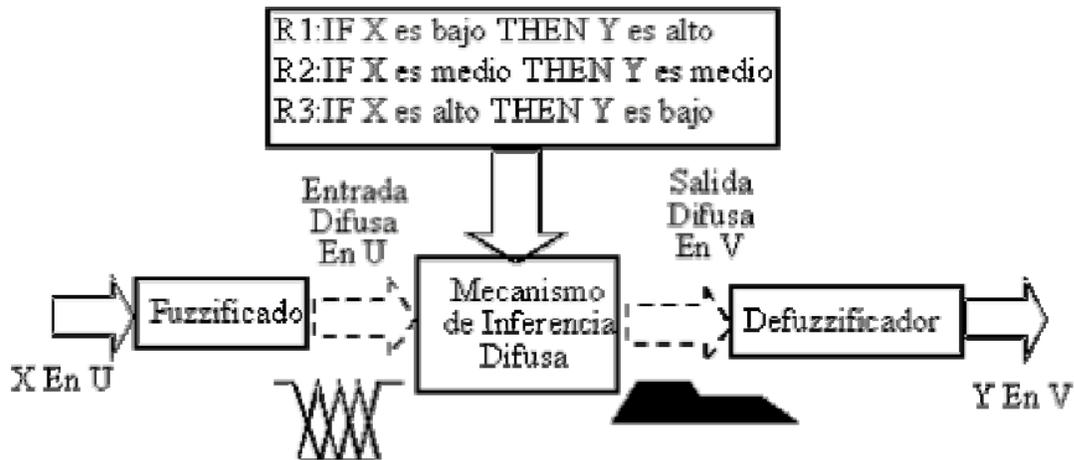


Fig.3 Modelo tipo Mamdani [3]

Fuzzificador

La entrada de un sistema de lógica difusa tipo Mamdani normalmente es un valor numérico proveniente, por ejemplo, de un sensor; para que este valor pueda ser procesado por el sistema difuso se hace necesario convertirlo a un "lenguaje" que el mecanismo de inferencia pueda procesar. Esta es la función del fuzzificador, que toma los valores numéricos provenientes del exterior para convertirlos en valores "difusos" que pueden ser procesados por el mecanismo de inferencia. Estos valores difusos son los niveles de pertenencia de los valores de entrada, a los diferentes conjuntos difusos en los cuales se ha dividido el universo de discurso de las diferentes variables de entrada al sistema.

Mecanismo de inferencia difusa

Teniendo los diferentes niveles de pertenencia arrojados por el fuzzificador, los mismos deben ser procesados para generar una salida difusa. La tarea del sistema de inferencia es tomar los niveles de pertenencia y apoyado en la base de reglas generar la salida del sistema difuso.

Conjunto de Reglas Difusas

El conjunto de reglas difusas es la manera que tiene el sistema difuso de guardar el conocimiento lingüístico que le permite resolver el problema para el cual ha sido diseñado. Estas reglas son del tipo IF-THEN.

Una regla de la base de reglas o base de conocimiento tiene dos partes, el antecedente y la conclusión, tal como se observa en la figura 4:

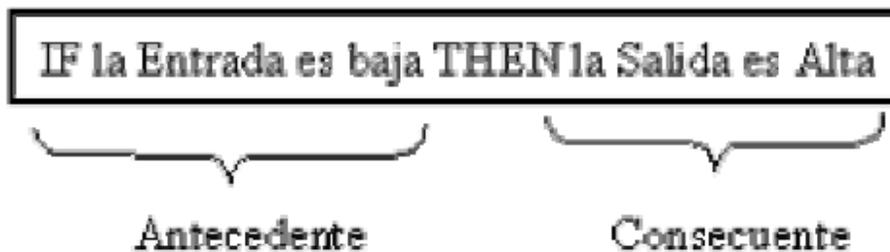


Fig.4 Regla Difusa

En un sistema difuso tipo Mamdani, tanto el antecedente como el consecuente de las reglas están dados por expresiones lingüísticas.

Defuzzificador

La salida que genera el mecanismo de inferencia es una salida difusa, lo cual significa que no puede ser interpretada por un elemento externo (por ejemplo un controlador), que solo manipule información numérica. Para lograr que la salida del sistema difuso pueda ser interpretada por elementos que solo procesen información numérica, hay que convertir la salida difusa del mecanismo de inferencia; este proceso lo realiza el defuzzificador.

La conversión de la salida difusa del mecanismo de inferencia, puede ser realizada a partir de varias opciones, tales como: el Centro de Gravedad (ecuación 04), los Centros Promediados (ecuación 05), entre otros.

$$y = \frac{\sum_i b_i \int \mu(i)}{\sum_i \int \mu(i)} \quad (04)$$

Donde:

b_i : es el peso de cada conjunto i

$U(i)$: es el universo del conjunto difuso i

$$y = \frac{\sum_i b_i \mu_{premise}(i)}{\sum_i \mu_{premise}(i)} \quad (05)$$

Donde:

b_i : es el peso de cada conjunto i

$U_{premise}(i)$: es el centro promediado de cada conjunto difuso i

En el siguiente gráfico (figura 5) se muestra de manera más detallada como es el funcionamiento de un sistema difuso Mamdani

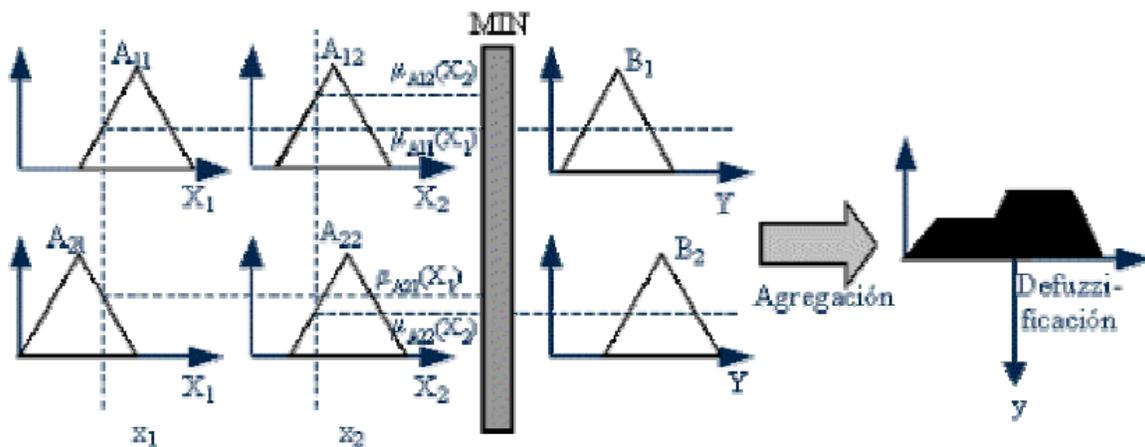


Fig. 5 Sistema Difuso Mamdani, Procesamiento Detallado

1.5 Aplicaciones de la Lógica Difusa

La lógica difusa se utiliza cuando la complejidad del proceso en cuestión es muy elevada, y asimismo no existen modelos matemáticos precisos: También es utilizada cuando los procesos son altamente no lineales y no se envuelven definiciones ni conocimientos estrictamente definidos (impreciso o subjetivo).

En cambio, no es una buena idea aplicarla cuando algún modelo matemático consigue solucionar eficientemente el problema, o cuando los problemas son lineales o no exista solución.

Esta técnica se ha empleado con bastante éxito en la industria, principalmente en Japón, y cada vez se está usando en una gran diversidad de campos. La primera vez que se usó de forma importante fue en el metro japonés, con excelentes resultados. A continuación se citan algunos ejemplos de su aplicación:

- En productos al consumidor: lavadoras, hornos de microondas, ollas arroceras, limpiadores al vacío, refrigeradores, cámaras de video, televisores, sistemas térmicos, traductores.
- Sistemas de control de acondicionadores de aire
- Optimización de sistemas de control industriales
- Sistemas de reconocimiento de escritura
- Mejora en la eficiencia del uso de combustible en motores
- Sistemas expertos del conocimiento (simular el comportamiento de un experto humano)
- Tecnología informática (compresión de datos)
- Bases de datos difusas: Almacenar y consultar información imprecisa. Para este punto, por ejemplo, existe el lenguaje FSQL.
- En sistemas: Elevadores, trenes, automóviles (máquinas, transmisiones, frenos), controles de tráfico.
- En general, en la gran mayoría de los sistemas de control que no dependen de un Sí/No.

En la tabla 1 se muestra un cuadro comparativo entre los sistemas convencionales y las aplicaciones de la Lógica Difusa.

	Problemas de interfaces Hombre/máquina	Problemas no lineales variantes en el tiempo	Clasificación de problemas
Problemas de los métodos convencionales	Dificultad para expresar numéricamente los objetivos del control. Evaluación del control por interpretación humana	La dinámica de la planta varía con el tiempo. Plantas no lineales Sobreflujo oscilación	La acción a tomar no es clara No es posible describir todas las trayectorias de solución Limitaciones hardware/velocidad
Aplicaciones	Control de suspensión Transmisiones automáticas Metro de Sendai	Control de temperatura de Control de posición de las cabezas de un disco duro Pilotos automáticos	Auto ZOOM Reconocimiento de patrones escritos a mano Transmisiones automáticas

Tabla 1. Aplicaciones de la Lógica Difusa [4]

1.6 Objetivos

Objetivos Generales

- Desarrollar un sistema de control de tráfico de bajo costo, energético y monetario, basado en conteo de vehículos utilizando las tecnologías existentes de conteo; además, de una interfaz de control inteligente que se adapte a los diversos momentos del día, y una adecuada sincronización para un tránsito de vehículos más fluido.
- Generar un reporte semanal con las condiciones de tráfico según la hora y día de la semana pudiendo calcularse las horas pico y la duración de éstas, para finalmente poder obtener una estadística para posteriores estudios.

Objetivos Específicos

- Desarrollar un módulo de control basado en tecnología PIC para el manejo de las variables utilizando algoritmos de Lógica Difusa, además de ser adecuado para un ambiente de alto ruido.

- Desarrollar una fuente de alimentación especial para el módulo de control diseñado.
- Desarrollar una interfaz gráfica de cuenta regresiva para los vehículos presentes en la avenida.
- Desarrollar una interfaz a una computadora utilizando el protocolo de comunicaciones RS-232.
- Diseñar un software aplicativo para el almacenamiento de la información y poder realizar las estadísticas necesarias.

1.7 Trabajos relacionados

Actualmente están apareciendo diversos trabajos relacionados con el control de tráfico vehicular, pero enfocado de diversas maneras, principalmente trabajando con todo lo que es el procesamiento digital de imágenes para el reconocimiento de vehículos.

Hay un trabajo muy similar basado en el mismo principio el cual fue desarrollado por Shahariz Abdul Aziz, Jeyakody Parthiban, en donde se observan resultados bastante interesantes en comparación con diferentes sistemas que funcionan actualmente, entre ellos la interfaz humana basada en los policías de tránsito.

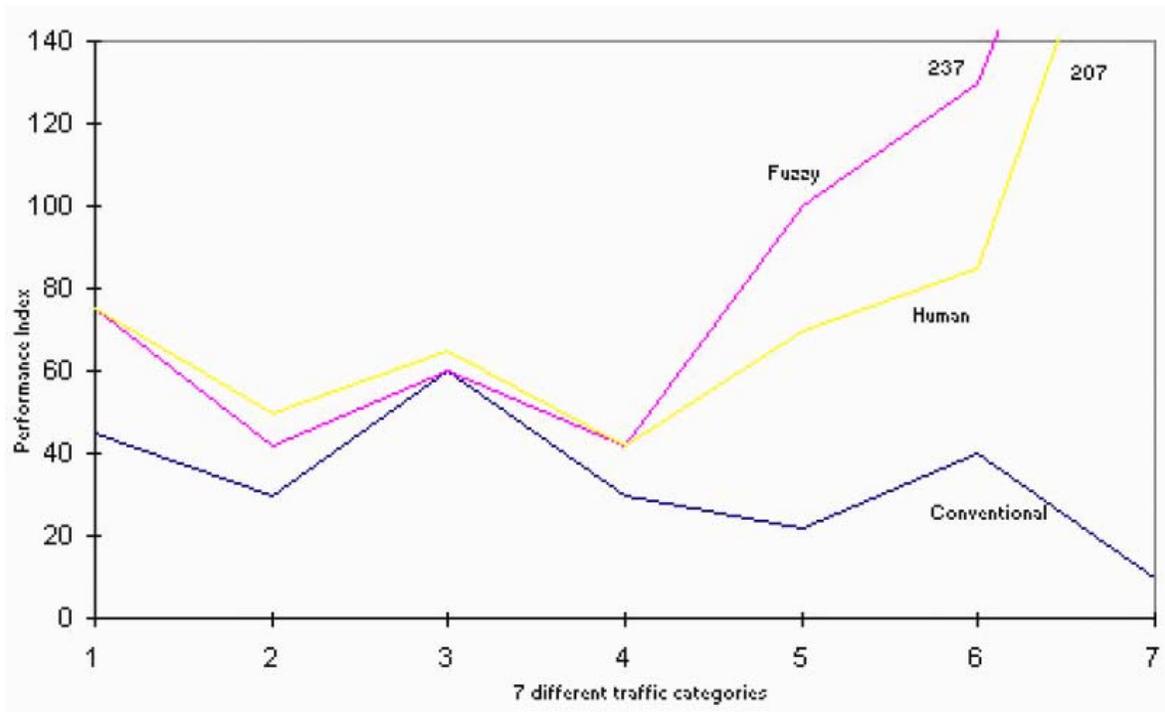


Fig.6 Resultados del trabajo relacionado [5]

La diferencia entre el trabajo de Shahariz y el desarrollado en este informe de tesis, radica en el controlador principal, pues el trabajo de Shahariz está basado en la teoría de redes neuronales en combinación con la Lógica Difusa para que su algoritmo tenga un mayor grado de inteligencia (ver figuras 7 y 8).



Fig.7 Modelo de controlador seleccionado [5]

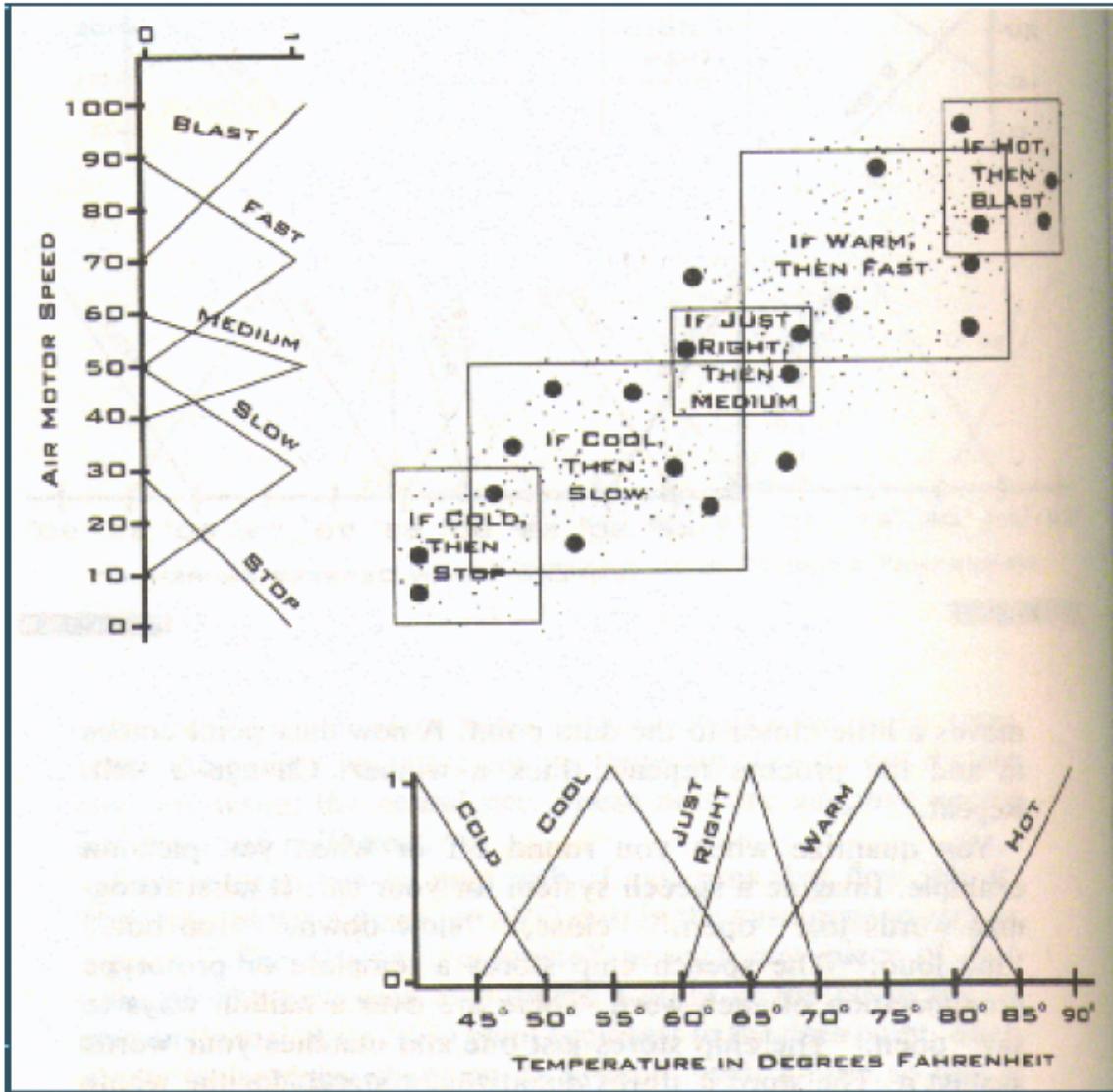


Fig.8 Combinación de la lógica Difusa con la redes neuronales [5]

DESCRIPCION DEL SISTEMA

2.1 Generalidades

En esta investigación se ha analizado específicamente una intersección de dos avenidas muy conocidas y a la vez congestionadas: cruce de las avenidas Benavides y Caminos del Inca, en donde se tiene la particularidad de que no es simplemente una intersección con dos periodos de semáforo, sino que cuenta con tres periodos de semáforo: primero da el paso a la avenida Benavides; luego da el paso a la avenida Caminos del Inca con dirección hacia el distrito de San Borja, pudiendo además de dar la posibilidad de giro hacia la izquierda para los vehículos que se dirigen hacia Miraflores; y, en tercer lugar, da el paso a la avenida Caminos del Inca con dirección hacia la Bolichera, dando además la posibilidad a los vehículos de poder dar la curva a la izquierda para poder ingresar a la avenida Benavides.

El proyecto se ha dividido en cuatro partes: en primer lugar se tiene el diseño de la fuente de alimentación requerida para el funcionamiento de la placa de control con todos los periféricos requeridos; en segundo lugar se tiene el diseño de la placa principal de control con todos los aspectos especificados en los objetivos; en tercer lugar se tiene las pruebas correspondientes a un posible tipo de sensor adecuado en la pista para el conteo de vehículos, el cual quedó finalmente en pruebas preliminares mas no fue integrado al sistema para pruebas más profundas; finalmente, se tiene la aplicación correspondiente para el almacenamiento de la información en una computadora y poder ser visualizada luego a través de un gráfico que muestre la cantidad de vehículos presentes en la intersección.

Se han realizado pruebas comparativas entre los resultados obtenidos en la intersección, con el sistema actual instalado y un posible resultado basado en el sistema que emplea la lógica difusa para la toma de decisiones.

En conclusión, todos los diseños del programa de control fueron implementados inicialmente en Matlab utilizando la librería Fuzzy, como referencia. En base a ello se ha realizado la implementación del algoritmo final en un dispositivo de hardware: PIC, para

realizar el procesamiento de la información utilizando el lenguaje de programación PIC basic.

2.2 Análisis de la intersección a implementar

Se realizó un trabajo estadístico en campo para así obtener las curvas que pudieran representar el comportamiento de los conductores a determinadas horas del día, pudiéndose así determinar claramente los tiempos de alto tráfico, tráfico medio y bajo tráfico durante toda la semana. En el eje horizontal se tiene el tiempo en segundos y en el eje vertical se encuentra la cantidad de vehículos en unidades. El modelado de dichos gráficos ha sido de acuerdo al tipo de curva característica que los contiene, siendo en la mayoría de los casos curvas polinomiales de segundo y tercer grado, observándose en el valor de R el grado de pertenencia de los puntos tabulados en la curva.

Los resultados se describen a continuación, en las figuras 9, 10, y 11.

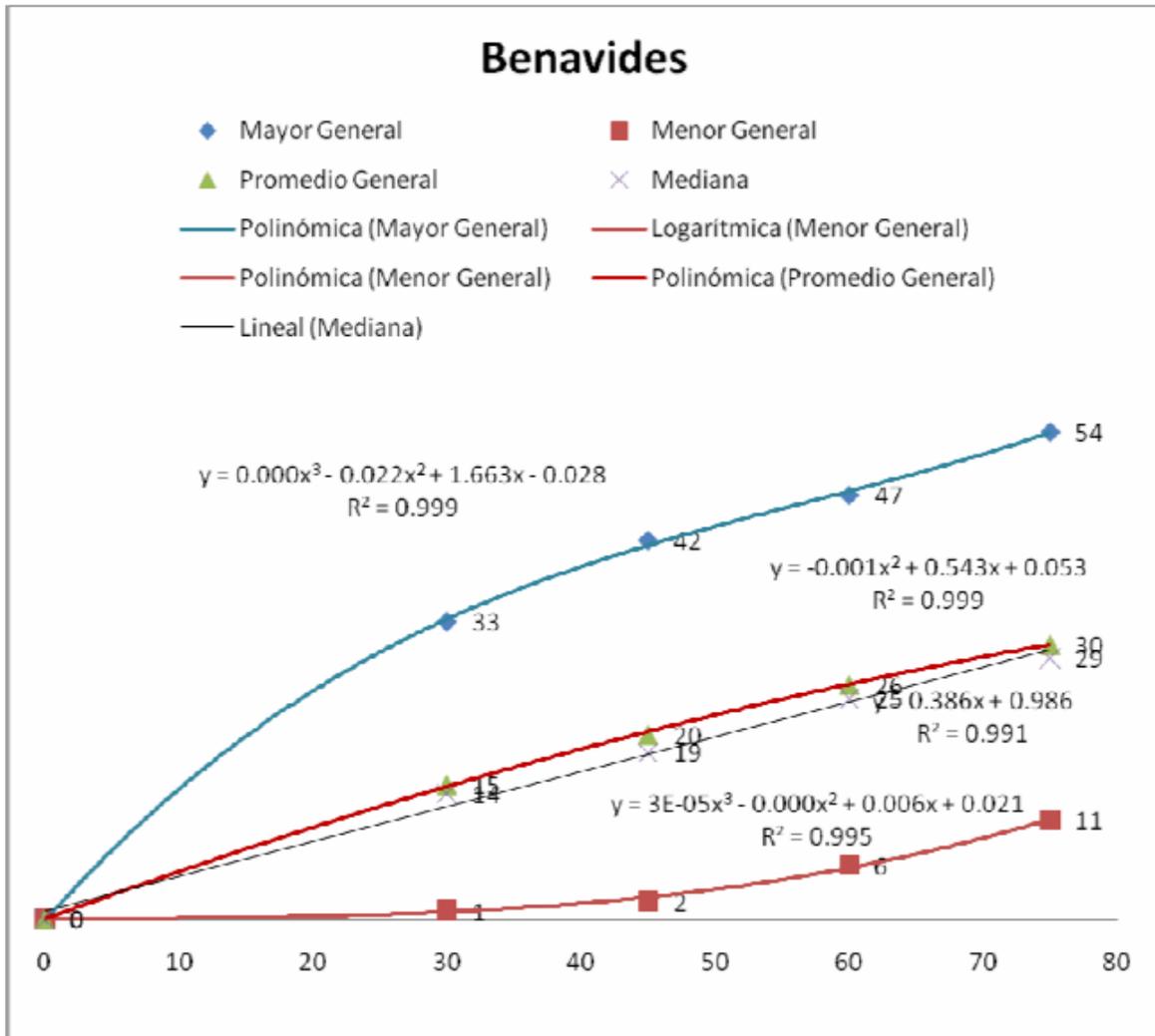


Fig.9 Curva de tráfico en la avenida Benavides

En la figura 9 se observa la curva que describe el flujo de vehículos en la avenida Benavides en sus dos sentidos, observándose que a medida que aumenta la cantidad de vehículos, la curva va perdiendo linealidad apreciándose la tendencia en cada caso.

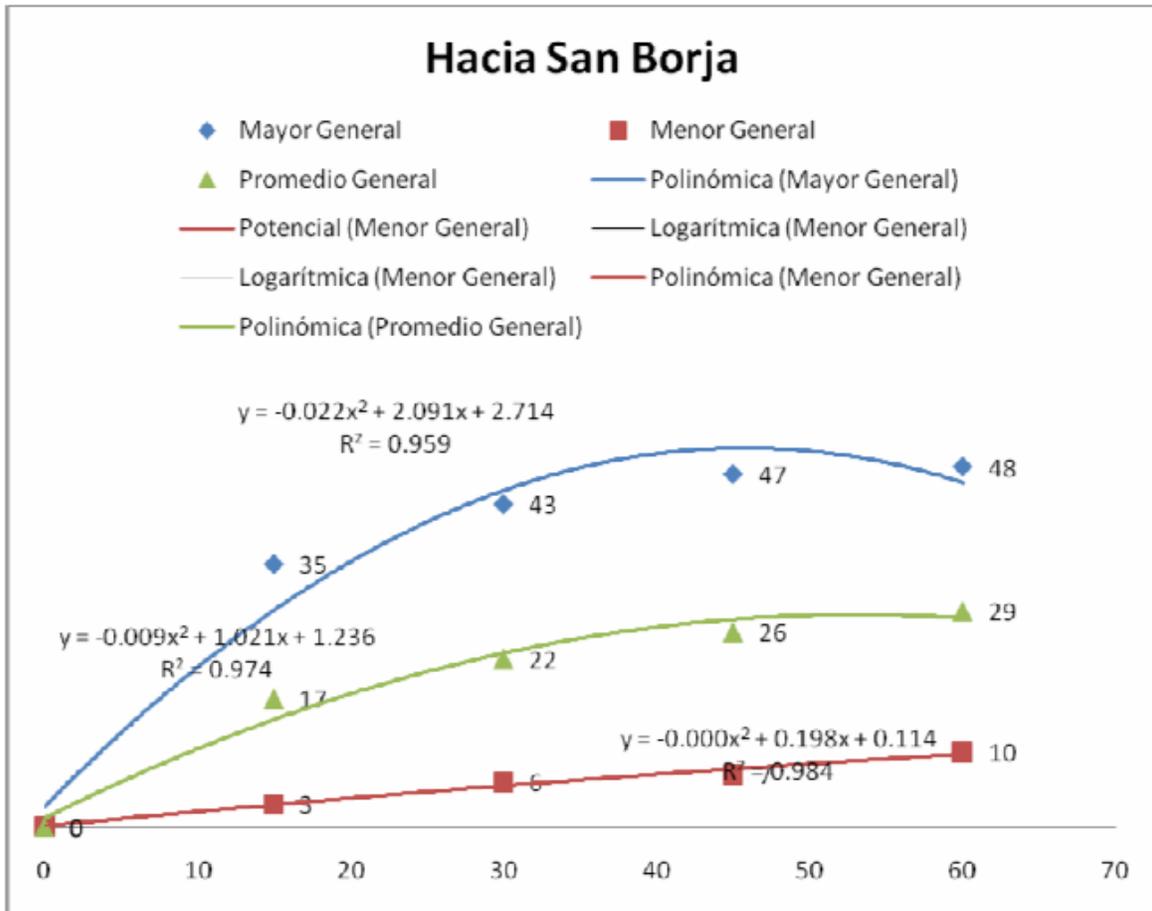


Fig.10 Curva de tráfico de la avenida Caminos del Inca hacia San Borja

En la figura 10 se observa la curva que describe el flujo de vehículos en la avenida Caminos del Inca en dirección al distrito de San Borja, observándose que a medida que aumenta la cantidad de vehículos el embotellamiento se va haciendo más crítico; por lo tanto, se observa que la tendencia es a que circule menor cantidad de vehículos en esta parte de la intersección. Esto se debe principalmente a que en dicho segmento de la intersección se encuentra ubicado un paradero de autobuses, el cual obstruye casi la totalidad de las vías debido al uso inapropiado de dicho lugar.

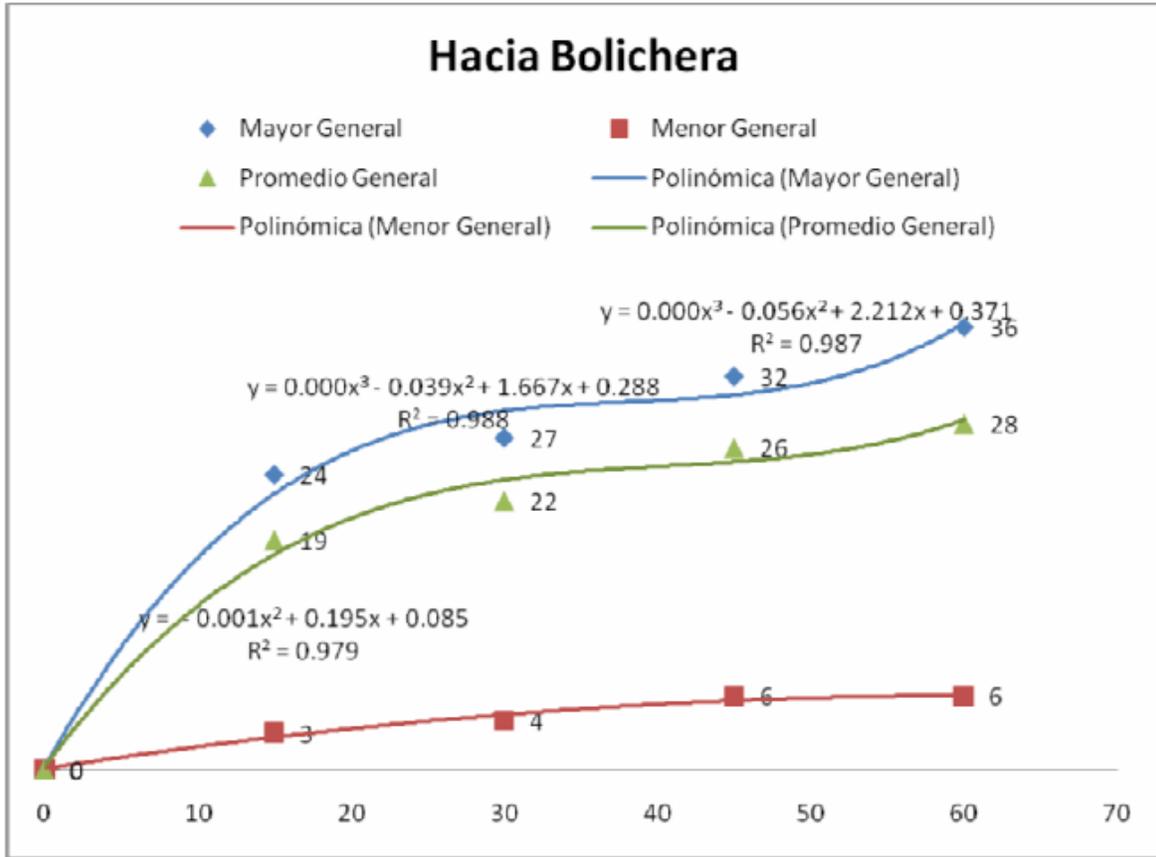


Fig.11 Curva de tráfico de la avenida Caminos del Inca hacia la Bolichera

Asimismo, en la figura 11 se observa la curva que describe el flujo de vehículos en la avenida Caminos del Inca en dirección a la Bolichera. En este caso la situación es contraria a lo que sucede en la avenida Caminos del Inca con dirección al distrito de San Borja. Esto se debe a que la afluencia de vehículos en este segmento de la intersección es mucho menor que en los casos anteriores y actualmente no presenta mayores problemas.

Como resultado del comportamiento de estos tres segmentos de la intersección, se presentan los siguientes resultados mostrándose finalmente la eficiencia del sistema de control de tránsito instalado actualmente en esta intersección (ver figuras 12, 13, 14, 15, 16 y 17). Tales resultados se obtienen a partir de los datos conseguidos tras realizar un conteo de vehículos en la intersección y comparar la relación de vehículos que llegan a la intersección y los que salen de ella, teniéndose en las figuras 12, 14 y 16 en el eje horizontal el número de muestra y en el eje vertical la cantidad de vehículos en unidades y

en las figuras 13, 15 y 17 en el eje horizontal el número de muestra y en el eje vertical el porcentaje de eficiencia en cada muestra.

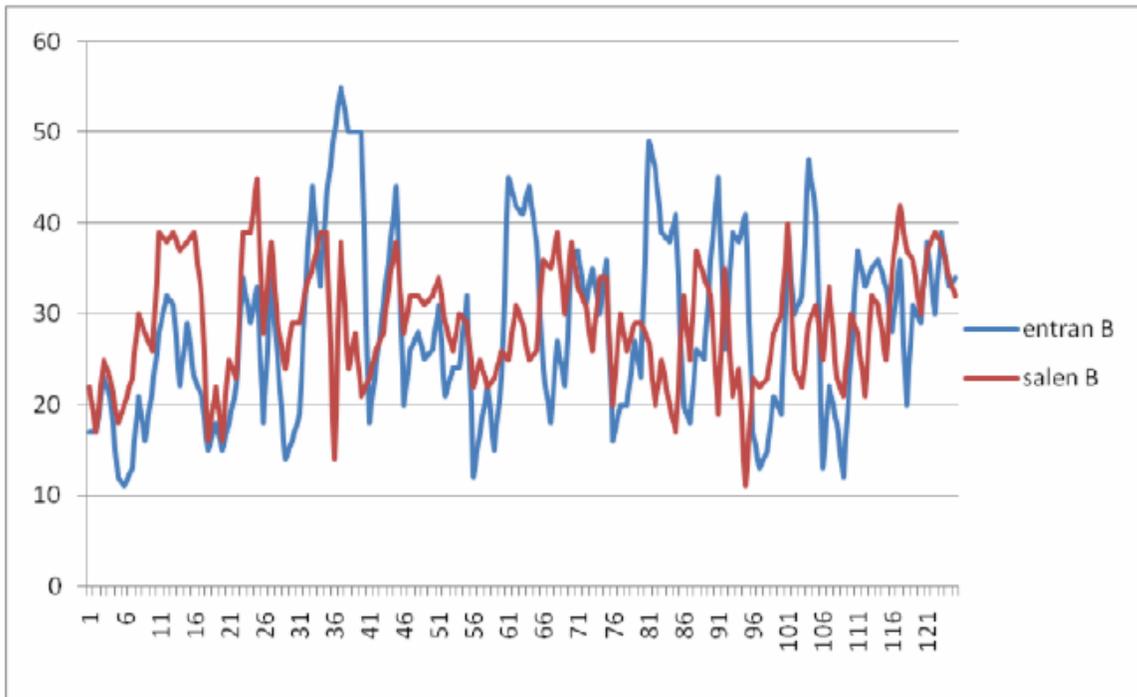


Fig.12 Relación de vehículos que entran y salen de la Intersección por parte de la avenida Benavides

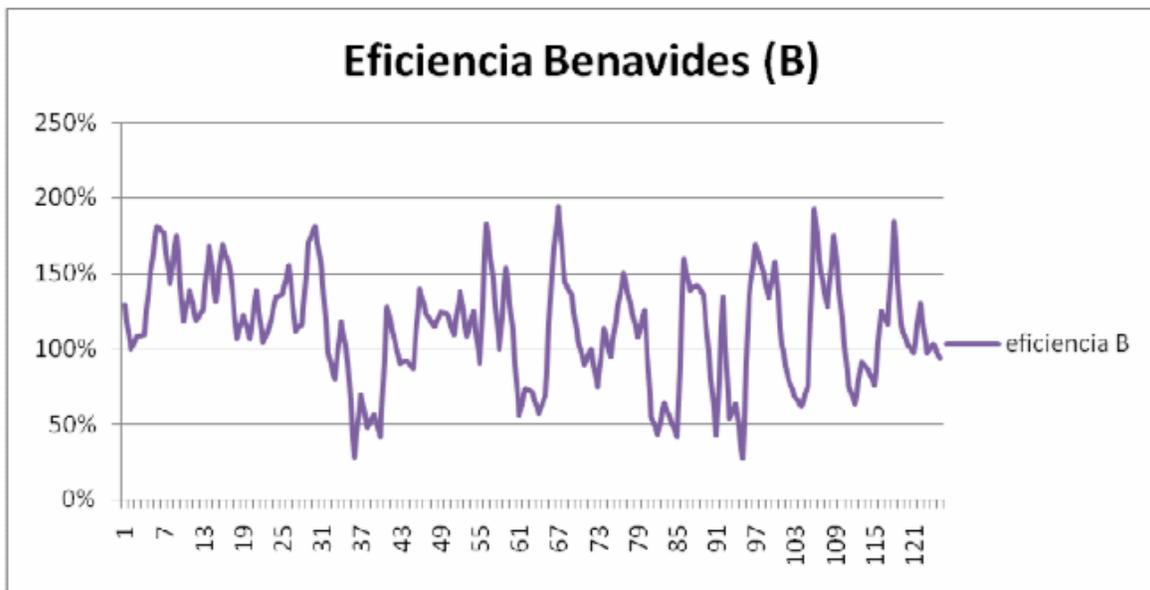


Fig.13 Eficiencia de la Intersección por parte de la avenida Benavides

Se observa en la figura 11 que la cantidad de vehículos que salen de la intersección en varias de las muestras es mucho mayor que la cantidad de vehículos que salen de la intersección, apreciándose varios picos. En la figura 12 se observa que la eficiencia se ve seriamente disminuida a causa de este factor. Otro factor que afecta el resultado es el tiempo excesivo que en muchos casos se aprecia en la intersección, por lo que en las figuras también se observan picos de vehículos que salen de la intersección debido a que la cantidad de vehículos presentes al momento del cambio es mucho menor que cuando el semáforo ya se encuentra en verde, en donde siguen llegando vehículos. Del mismo modo se pueden apreciar estas inferencias en las figuras 13, 14, 15 y 16.

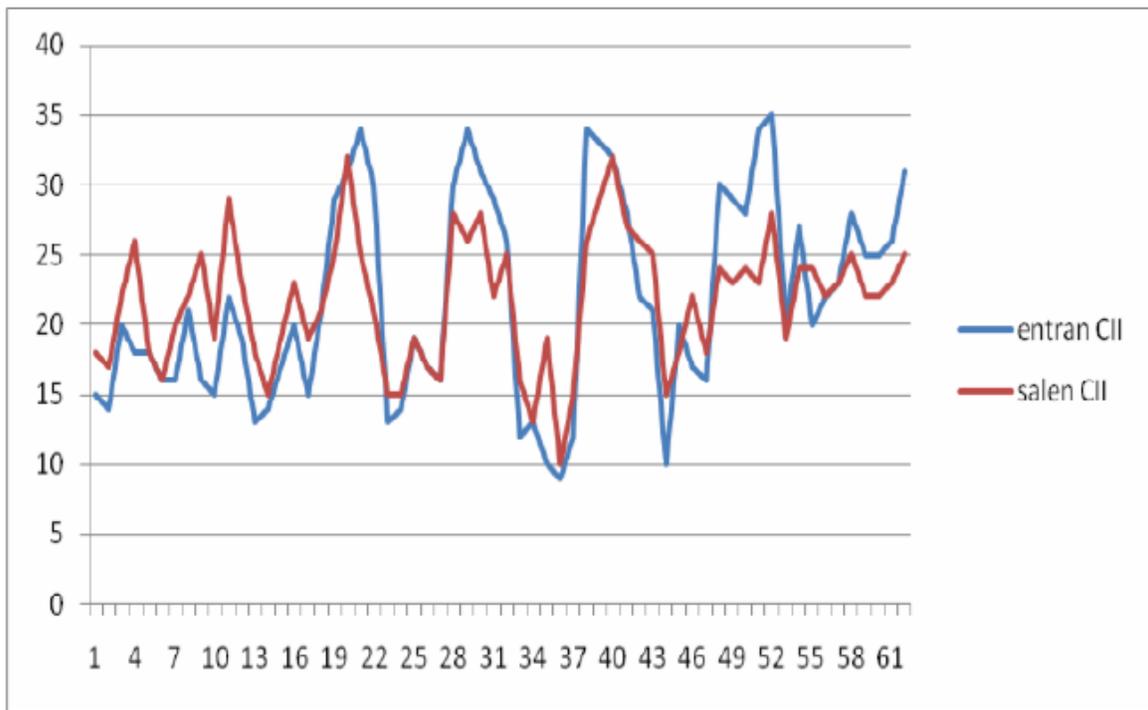


Fig.14 Relación de vehículos que entran y salen de la Intersección por parte de la avenida Caminos del Inca con dirección a la Bolichera

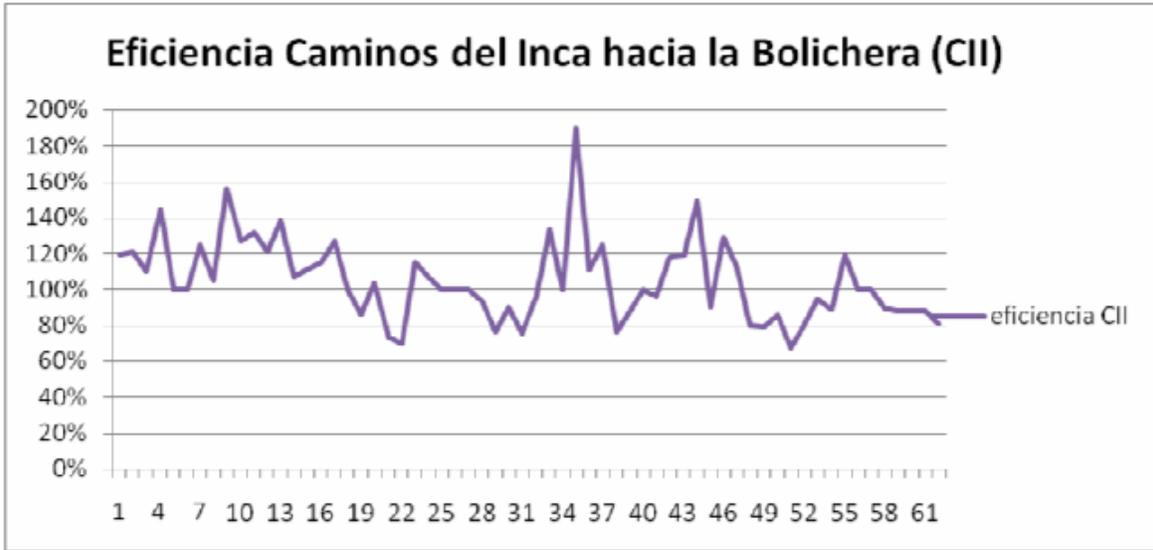


Fig.15 Eficiencia de la Intersección por parte de la avenida Caminos del Inca hacia la Bolichera

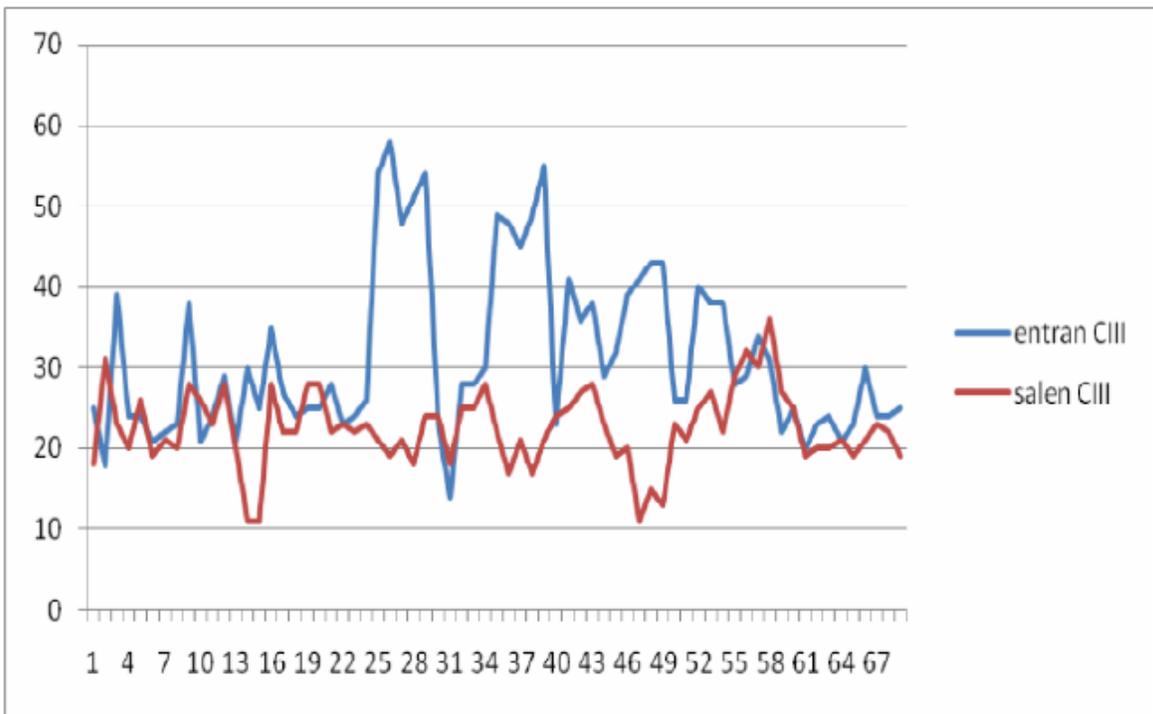


Fig.16 Relación de vehículos que entran y salen de la Intersección por parte de la avenida Caminos del Inca con dirección a San Borja

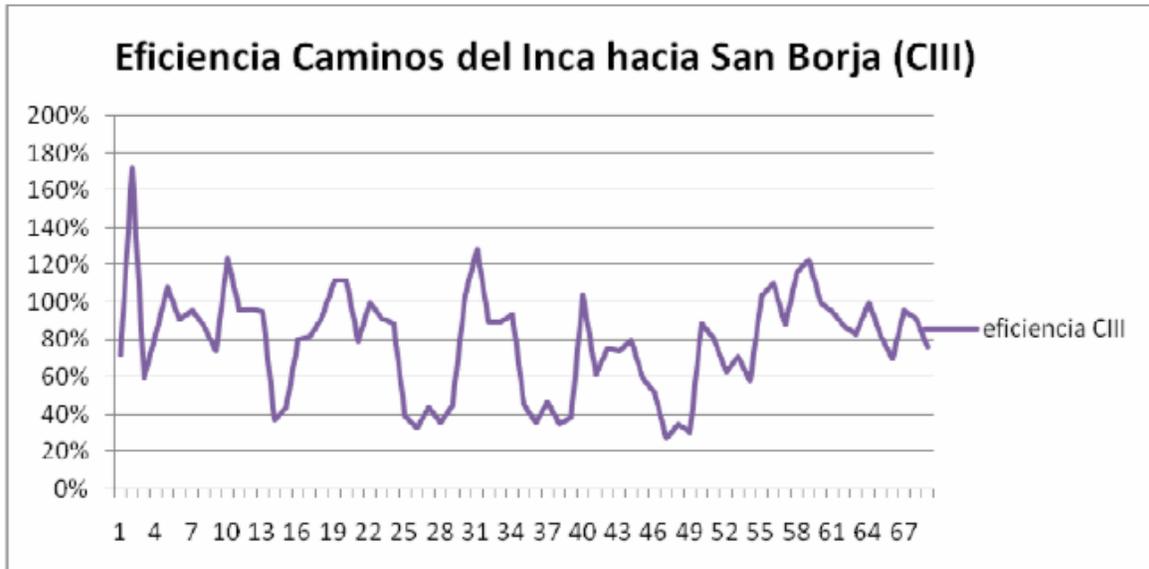


Fig.17 Eficiencia de la Intersección por parte de la avenida Caminos del Inca hacia San Borja

De los resultados obtenidos y mostrados anteriormente, se puede apreciar finalmente que los tiempos en la intersección no se encuentran bien distribuidos, ya que presentan muchos altibajos en la eficiencia llegando en el peor de los casos a una eficiencia de menos del 30%, siendo crítico en especial para el caso de la avenida Caminos del Inca con dirección a San Borja, donde se presentan los picos más bajos en la eficiencia y cuyo promedio es el menor, lo cual es inaceptable.

2.3 Diseño de la Fuente de Alimentación

Para el diseño de la fuente de alimentación se tomaron las siguientes consideraciones:

- Operar con el voltaje alterno de 220 voltios y con un circuito de protección contra sobrecorrientes, sobrevoltajes y eliminación de picos de arranque para evitar el deterioro de los equipos que trabajan bajo este voltaje.
- Operar, además, con un voltaje de 24V DC con un pico de corriente máximo de 1.6 Amperios a carga completa, teniendo un margen de corriente del 85% adicional para proteger la fuente en sí o para futuras expansiones.
- Asimismo, debe operar con un voltaje de 5V DC para la alimentación de los microcontroladores y algunos periféricos presentes en la placa principal.

Para la protección contra sobrevoltaje se está utilizando un varistor, el cual opera de la siguiente manera: cuando ocurre un pico de voltaje en la alimentación, el varistor absorbe este pico manteniendo el voltaje pico en el indicado para el sistema, estos picos no pueden presentarse por periodos demasiado largos, porque el dispositivo se degrada y puede llegar a quemarse.

A fin de calcular la energía disipada durante un impulso, se hace con la referencia generalmente a una onda normalizada de la corriente. Esta onda esta prescrita por la norma “IEC 60-2 sección 6”, y tiene una forma que aumenta desde cero al valor de pico en un tiempo corto, disminuyendo hasta cero o de una manera exponencial, o bien sinusoidal.

Esta curva es definida por el tiempo principal virtual (t_1) y el tiempo virtual al valor medio (t_2) (ver la figura 18).

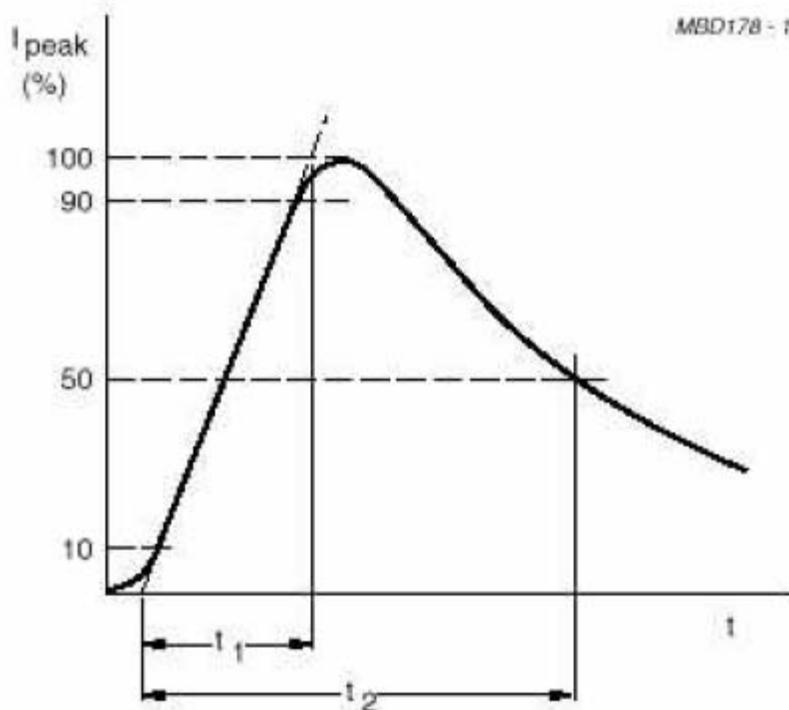


Fig.18 Curva de carga del varistor [6]

El cálculo de energía durante la aplicación de tal impulso viene dado por la ecuación 06.

$$E = V_{\text{pico}} \times I_{\text{pico}} \times t_2 \times K$$

(06)

donde:

I_{pico} = corriente de pico

V_{pico} = voltaje a la corriente de pico

K = constante que depende de t_2 , cuando t_1 va de 8 a 10 microsegundos.

En la figura 19 se muestra el comportamiento de la corriente del voltaje con respecto a la corriente; por lo tanto, a medida que aumenta la corriente en el varistor debido al aumento de voltaje, este absorbe parte de la energía entregada hasta llegar al punto de ruptura, en donde la corriente se dispara y el dispositivo se quema si es que no se elimina dicho pico de voltaje.

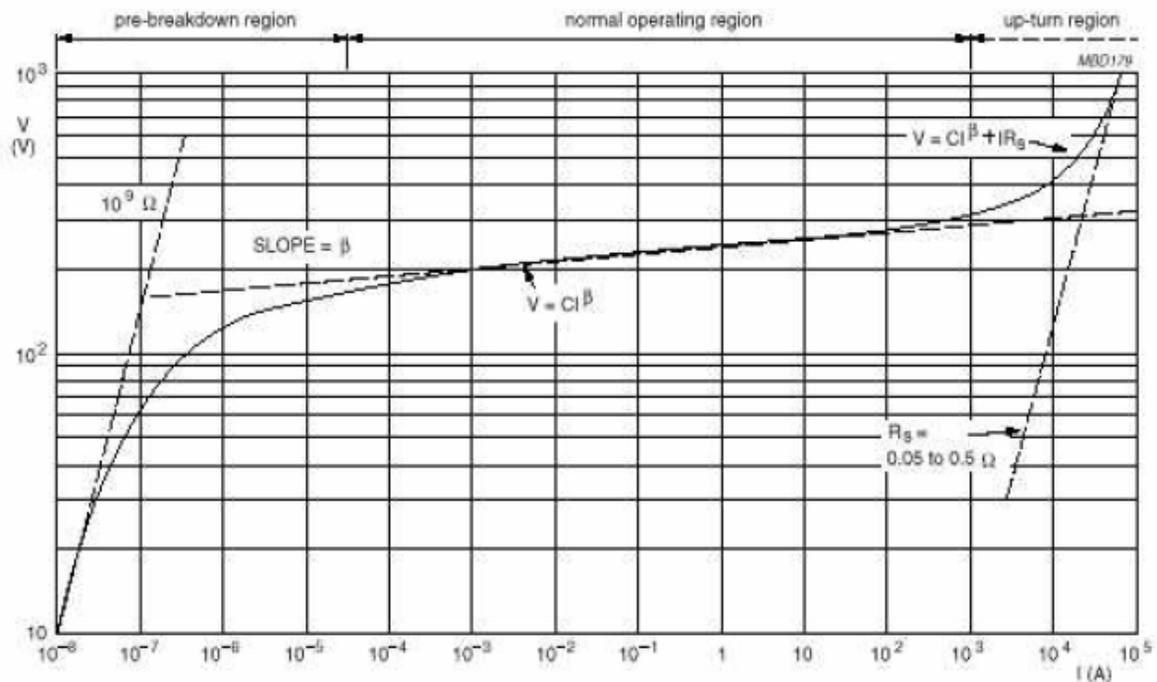


Fig.19 Curva Voltaje VS Corriente en el Varistor [6]

En el caso de la protección contra sobrecorriente, se tiene colocado un fusible para el caso de que se supere la cantidad de corriente necesaria para el funcionamiento del sistema, debido a algún problema presentado en alguno de los dispositivos periféricos. Además se

ha colocado un termistor NTC para una protección contra sobrecorrientes de arranque, ya que gracias a la curva característica de este elemento se consigue una carga del equipo de manera gradual.

En las figuras 20 y 21 se puede apreciar el comportamiento del termistor frente a los cambios de temperatura con respecto a la potencia, y de la corriente con respecto al tiempo. En este último se aprecia el efecto que tiene el termistor sobre un pico de corriente en el arranque, atenuándolo considerablemente.

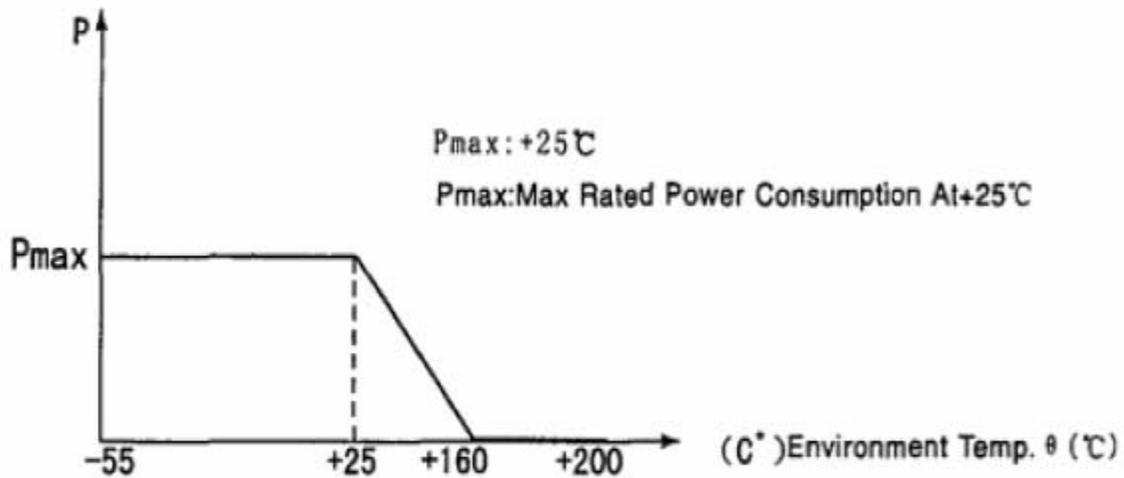


Fig.20 Curva de Potencia VS Temperatura del Termistor [7]

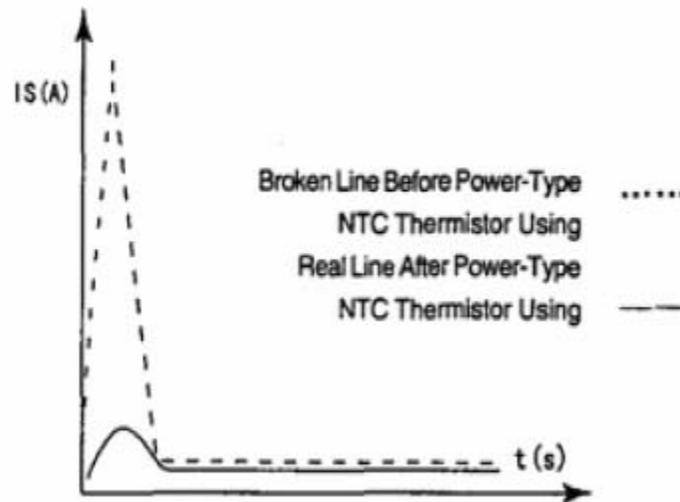


Fig.21 Curva de carga corriente VS tiempo [7]

En las figuras 22, 23 y 24 se muestra el diseño realizado para la fuente de alimentación, en donde se puede apreciar que se utilizó un puente de diodos a la salida del transformador para así poder hacer la rectificación del voltaje alterno que sale del mismo, se utilizó el 1N5404 por ser capaz de soportar corrientes mayores a 3 amperios. Luego se tiene el condensador C1 el cual realiza el rizado del voltaje antes de la entrada al regulador. Se

utilizó un regulador tipo chapa para así poder trabajar con corrientes que llegan hasta los 3 amperios, sin tener que llegar a utilizar multiplicadores de corriente. Al regulador de 24 voltios se le agregó un diodo de protección en el caso de que se produzca algún corto y no dañe al dispositivo. Luego del regulador de 24 voltios se tiene el regulador de 5 voltios el cual sirve para la alimentación del sistema principal, es decir el microcontrolador principal y secundario.

La etapa de 220 voltios se encuentra protegida con un varistor que opera a 220 voltios y un NTC el cual elimina los picos en el arranque del sistema, además se colocó un fusible en caso se produzca algún cortocircuito en el sistema. El interruptor utilizado es de un solo terminal para que de producirse un pico de corriente en el arranque, este sea filtrado por el NTC.

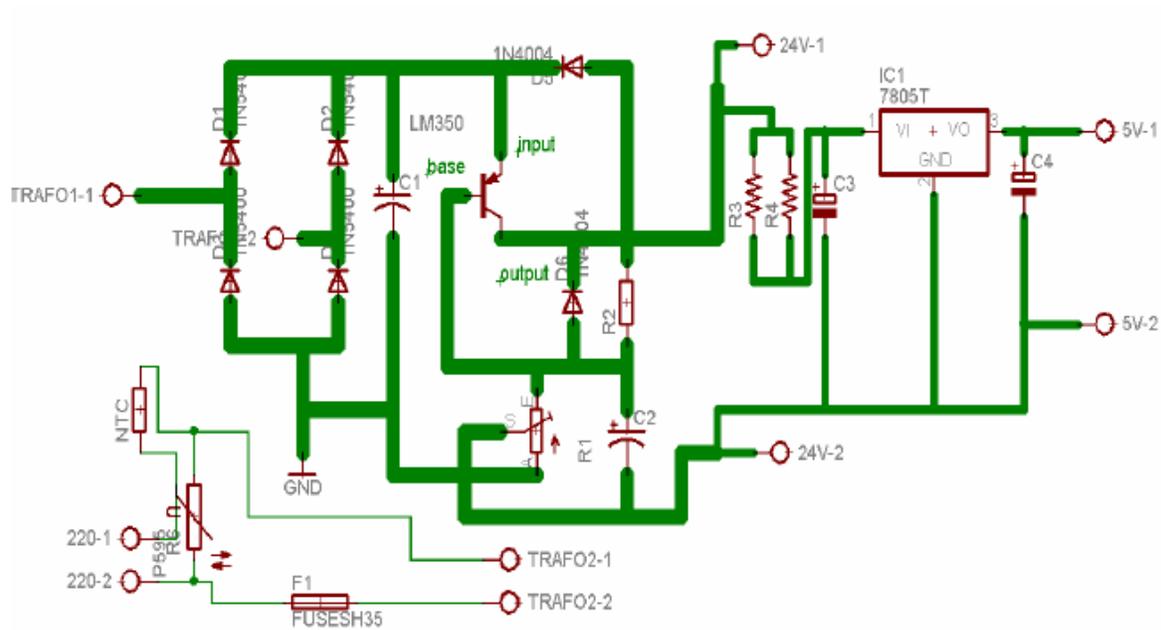


Fig.22 Esquemático de la fuente de alimentación

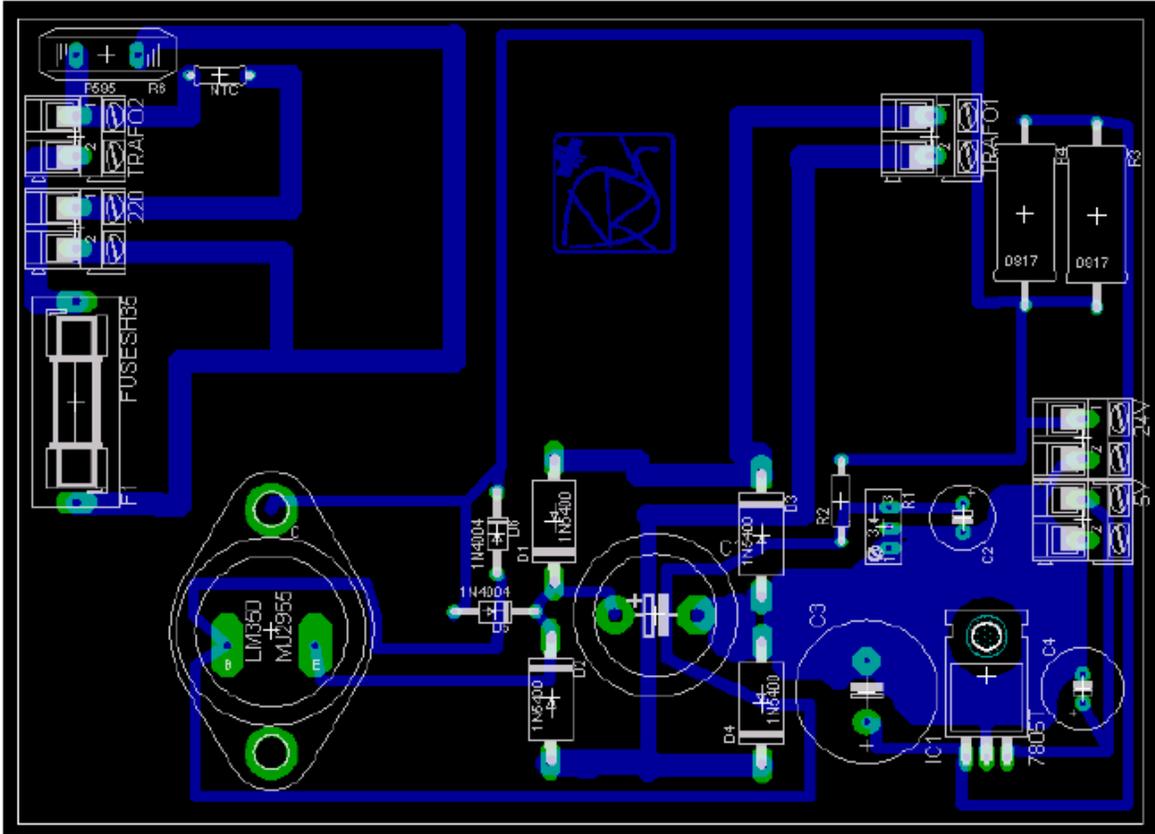


Fig.23 PCB de la fuente de alimentación



Fig.24 Fuente de alimentación terminada

Finalmente, la fuente de alimentación, se ubicaría en el interior de una caja para poder protegerla del medio ambiente, con la posibilidad de poder colocarle un ventilador con el objetivo de disipar mejor el calor que emite cuando opera a full carga.

2.4 Descripción del sistema de Control

El sistema se ha pensado para trabajar utilizando dos sensores por vía en cada uno de los accesos a la intersección, de esta manera es posible realizar la cuenta de los vehículos que llegan a dicha intersección. La separación que existe entre sensores en cada una de las vías, es aproximadamente de 80 metros lo cual facilita la obtención de un rango más amplio de vehículos que pueden ser contabilizados para el procesamiento de la información por parte del microcontrolador.

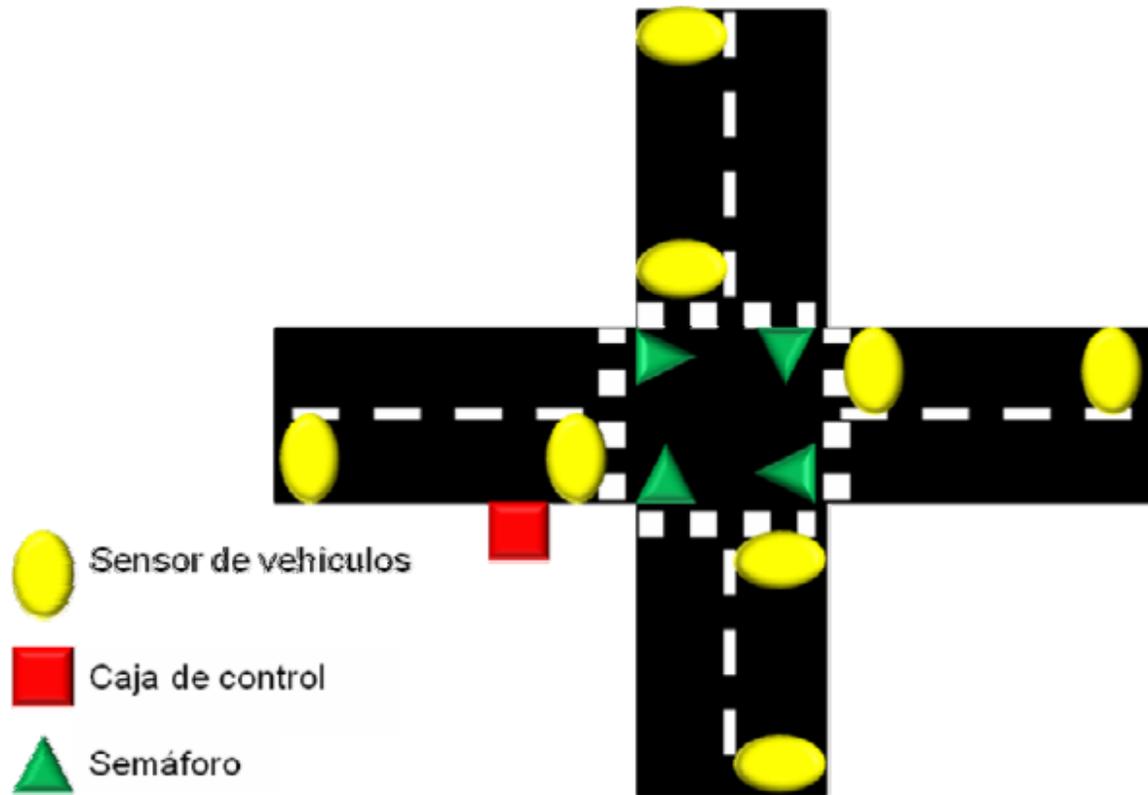


Fig.25 Ubicación y distribución del sistema

La figura 25 muestra la distribución de los dispositivos dentro de la intersección, además de la ubicación de los semáforos y de la caja de control donde se encontraría la placa de control (tarjeta electrónica que cuenta con todos los dispositivos interconectados).

Para el diseño de la placa de Control se han tenido las siguientes consideraciones:

- En la placa principal se tiene el sistema de control principal basado en el microcontrolador PIC18F458, el cual realiza el procesamiento del algoritmo de la lógica difusa, el procesamiento de los sensores en pista, y la comunicación entre la placa de control y la computadora a través de la interfaz serial RS-232, y el microcontrolador PIC16F877A, el cual se encarga de administrar los dispositivos periféricos, entre ellos: el control de los displays y el control de los focos de Leds para los vehículos y los peatones.

- Para la comunicación entre los microcontroladores se ha utilizado la interfaz serial RS-232, logrando así una adecuada comunicación entre dispositivos.
- La alimentación de los focos de leds para los semáforos vehiculares y peatonales trabajan en 220 AC, los cuales están aislados eléctricamente de los microcontroladores mediante el uso de optocopladores y controlando el encendido y apagado de éstos mediante triacs.
- La alimentación de los displays de siete segmentos es mediante 24V DC, los cuales son activados mediante el uso de un arreglo de transistores (KSP2222A), y que a la vez cumplen la función de interruptor para el encendido y apagado de éstos.
- Todos los focos y displays están protegidos contra corto circuito mediante el uso de fusibles de baja corriente, evitando así el deterioro de la placa principal en caso que alguno de estos dispositivos falle o se quemé.

Seguidamente, la figura 26 muestra un diagrama de bloques de la subdivisión del sistema de control.

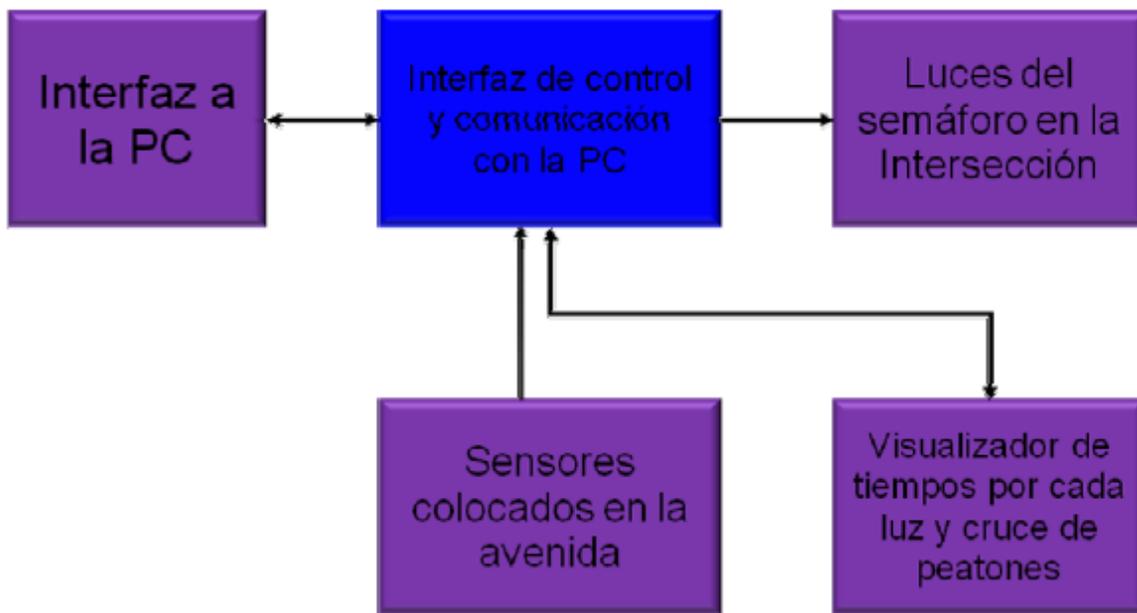


Fig.26 Diagrama de bloques del sistema

En este diagrama de bloques se puede observar que la interfaz de control es la que maneja finalmente todo el sistema y realiza la comunicación con la computadora, con la finalidad de realizar los reportes con la información obtenida de la intersección, pudiéndose además realizar reportes de tráfico y estadísticas con respecto a la situación de la intersección de acuerdo al momento del día.

A continuación, en el diagrama de flujo mostrado en la figura 27, se puede apreciar cómo es que el sistema distribuye las funciones y cómo éste realiza el proceso completo del algoritmo de la lógica difusa, en donde en primer lugar el PIC principal otorga la señal de inicio para que comience el desarrollo de la lógica difusa. Una vez realizada la operación, el PIC principal envía una señal de confirmación de realización e inmediatamente manda la información procesada al PIC secundario, para que éste pueda realizar la configuración de luces de semáforo apropiada con los tiempos calculados.

Una vez que el ciclo completo de la intersección se cumple, el PIC secundario manda una señal de término de la operación al PIC principal, para que éste pueda realizar un nuevo procesamiento de la información colectada mientras se realizaba el ciclo de semaforización en la intersección. Este proceso se realiza de manera continua e ininterrumpida durante las 24 horas del día, y para todos los días de la semana.

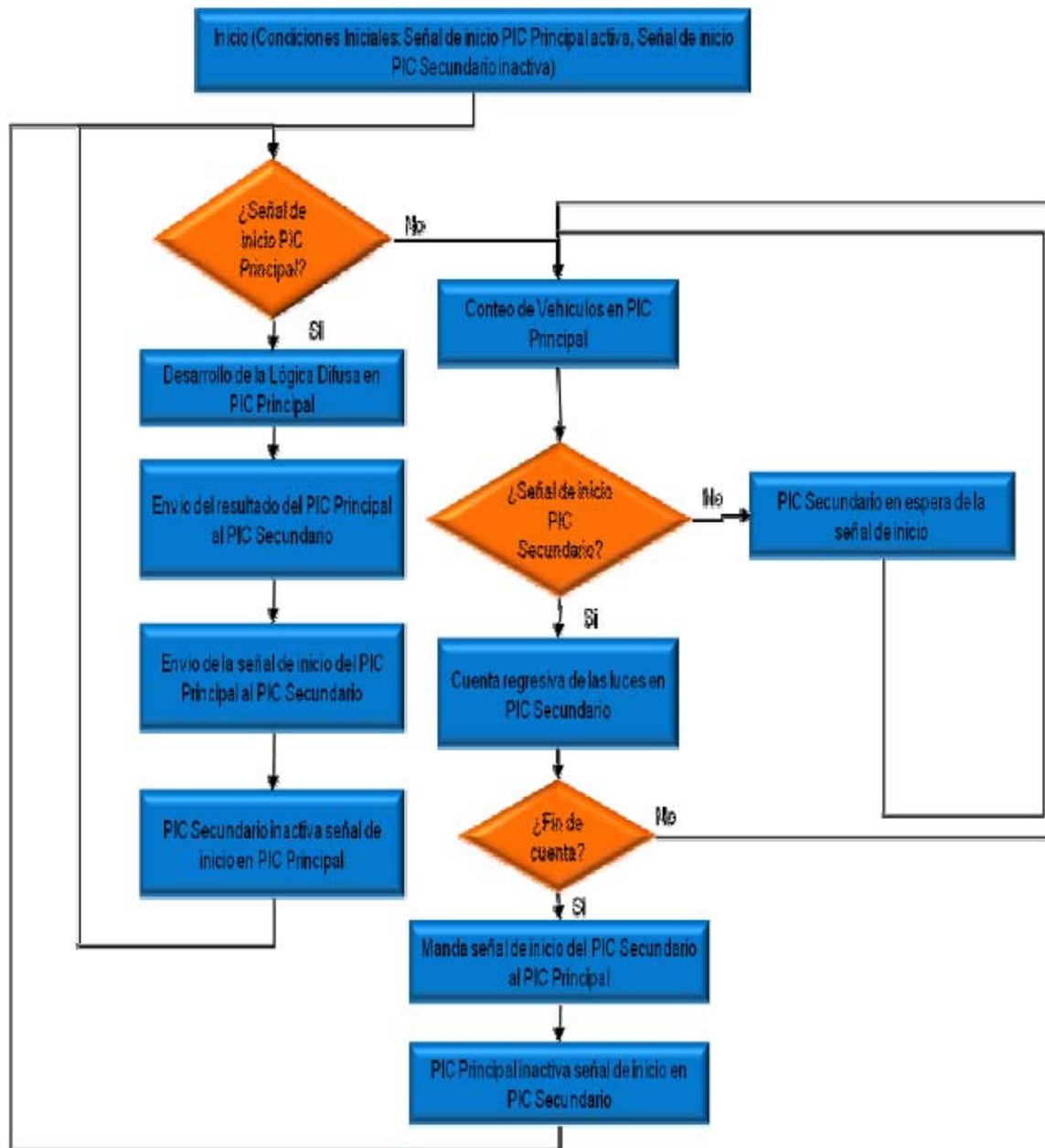


Fig.27 Diagrama de flujo de la placa de control

En el desarrollo del algoritmo se consideró la información obtenida de la intersección, para el cálculo de los conjuntos difusos y la aproximación de los conjuntos de salida; la obtención de los tiempos de semaforización se basó en la observación visual y la comparación con la curva característica de comportamiento obtenida de la estadística anteriormente realizada.

Los diseños desarrollados y correspondientes a las placas de control e interfaces, se muestran a continuación en las figuras 28, 29, 30 y 31.

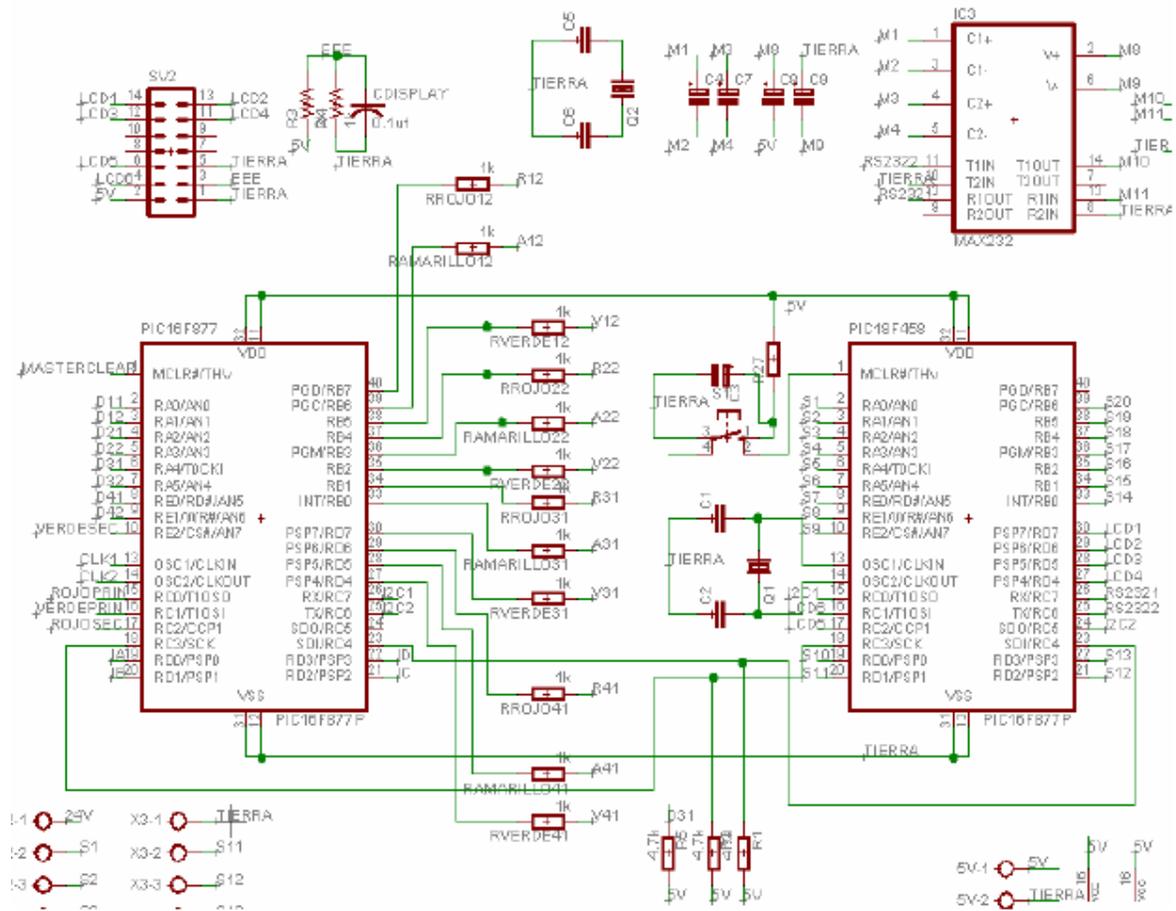


Fig.28 Microcontroladores de la placa de control

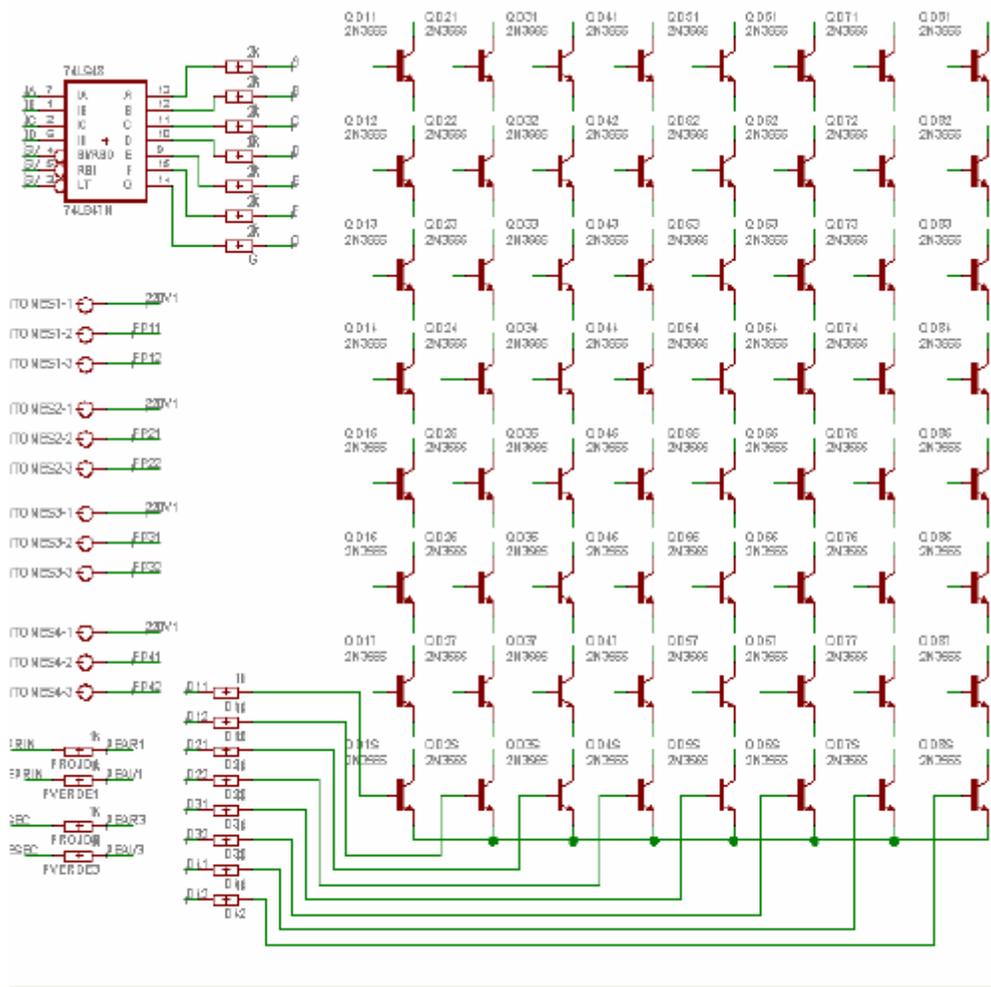


Fig.29 Interfaz de control de displays de cuenta regresiva

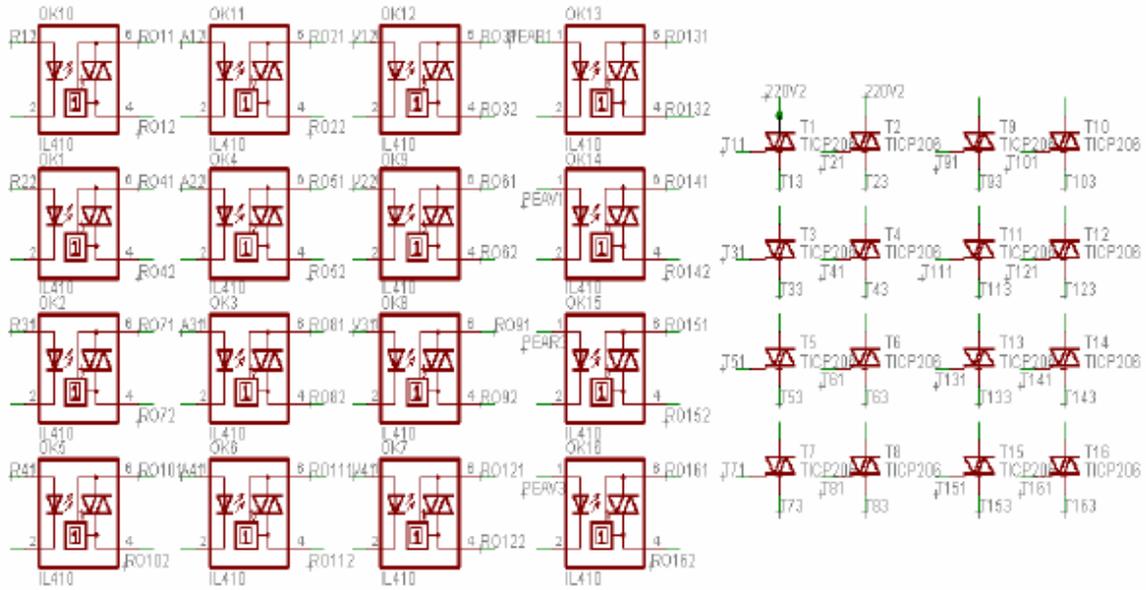


Fig.30 Interfaz de control de focos de Leds

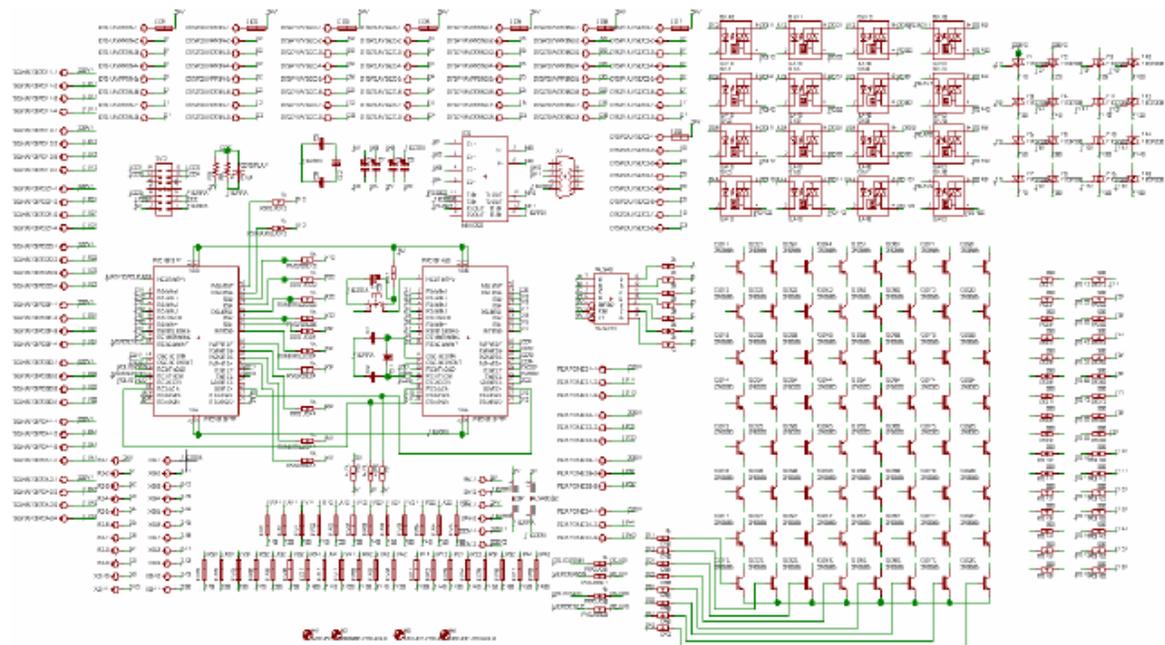


Fig.31 Placa de control Principal

De la figura 31, se observa que la placa de control principal integra todas las interfaces mostradas en las figuras anteriores, logrando así un diseño compacto y estructurado de tal manera que los diferentes voltajes que ingresan a la placa de control están debidamente direccionados de tal manera que no interfieran unos con otros.

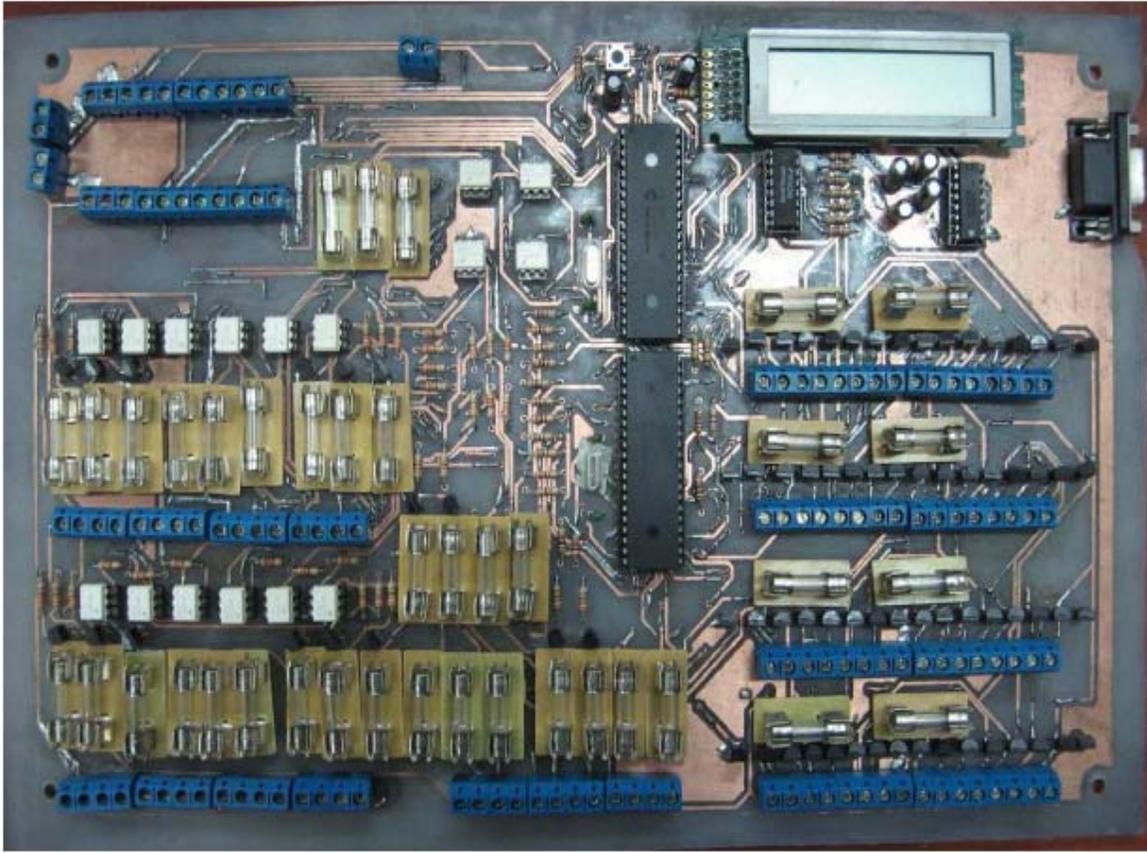


Fig.32 Placa de control terminada

En la placa de control terminada (figura 32), se puede apreciar además que se ha colocado un LCD para poder visualizar algunos de los procesos que el sistema realiza, de esta forma se puede llevar un monitoreo del sistema y en caso de producirse algún error durante su operación poder detectarlo y realizar las correcciones respectivas.

2.5 Comunicación Serial RS-232 hacia la PC

Para la comunicación serial desde la placa de control hacia la PC, se está utilizando la interfaz de comunicación RS-232 en complemento con el dispositivo MAX232, para poder interconectar el PIC principal a la PC.

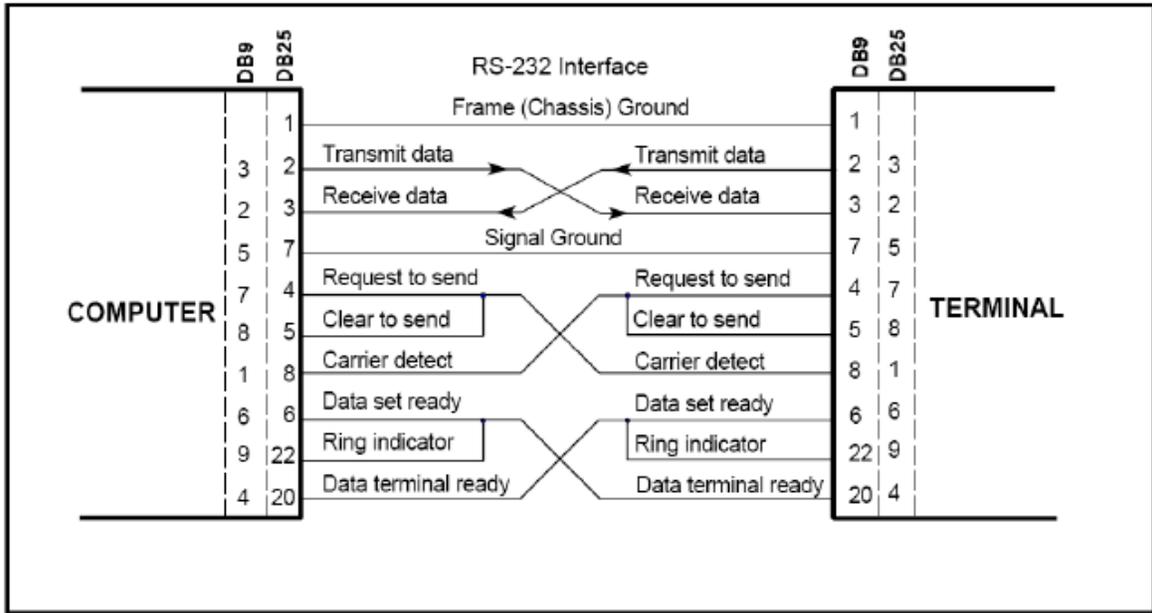


Fig.33 Interconexión entre la PC y el Terminal [8]

En la figura 33 se aprecia la conexión completa entre la computadora y la placa de control. En el proyecto realizado sólo se están utilizando los pines de transmisión de datos, recepción de datos y la señal de tierra, utilizando un conector DB9.

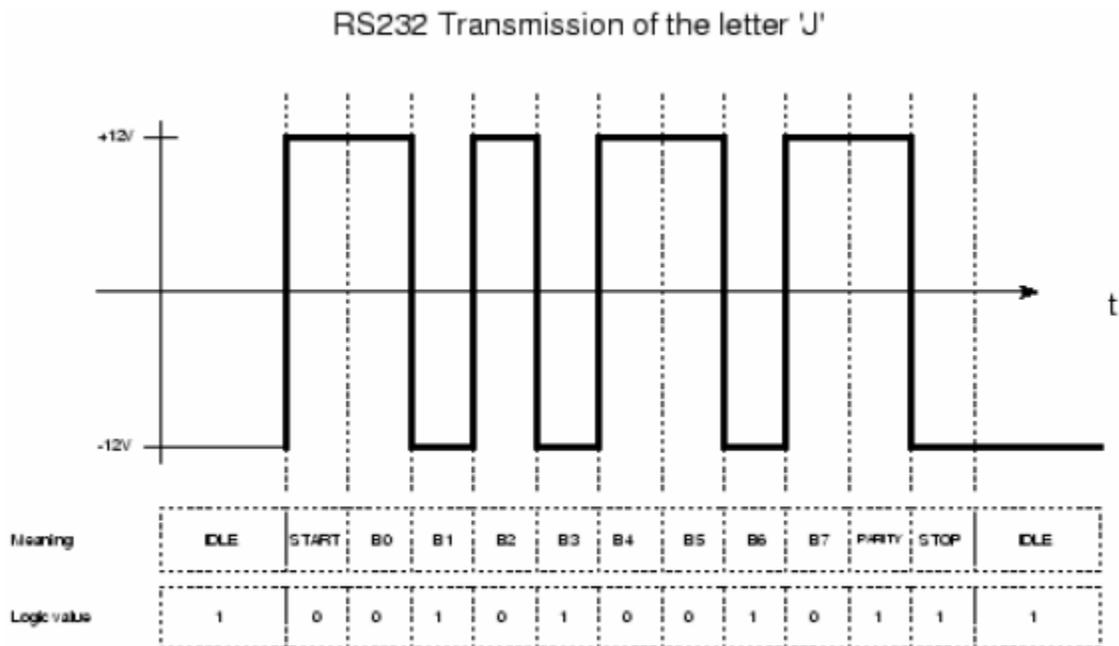


Fig.34 Formación de la data en interfaz RS-232[9]

Tal como lo muestra la figura 34, los datos son enviados en cadenas de ocho bits para el caso de transmisión de información, siendo esta una comunicación asíncrona, utilizando un bit de inicio, un bit de parada y un bit de paridad.

La configuración y/o características utilizadas para la comunicación del sistema a través del puerto serial de la PC, es la siguiente:

- Bits por segundo: 2400
- Bits de datos: 8
- Bit de paridad: Ninguno
- Bit de parada: 1
- Control de flujo: Ninguno

La figura 35 muestra el diseño correspondiente a la interfaz de comunicación RS-232, mientras que la figura 36 exhibe una fotografía de la interfaz físicamente finalizada.

2.6 Realización de la aplicación para la PC

La aplicación para poder realizar el almacenamiento de datos provenientes de la placa de control ha sido realizada en Visual Basic para así poder tener una aplicación ligera y de fácil instalación y manejo. En la figura 37 se puede apreciar la aplicación para la computadora.

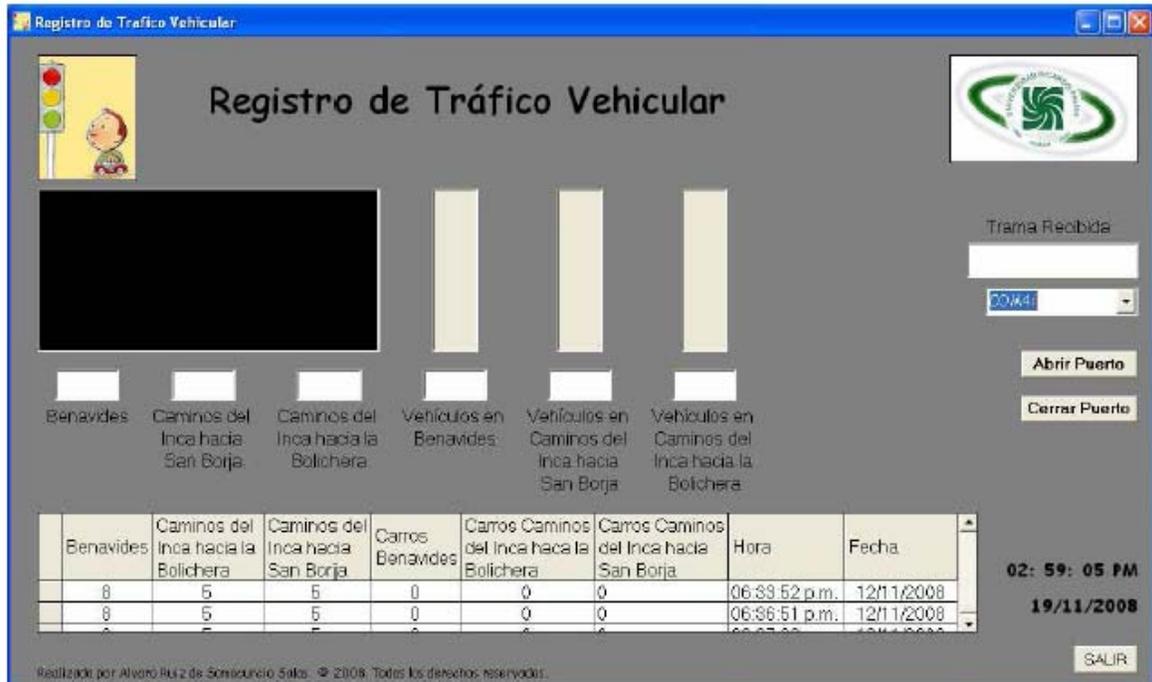
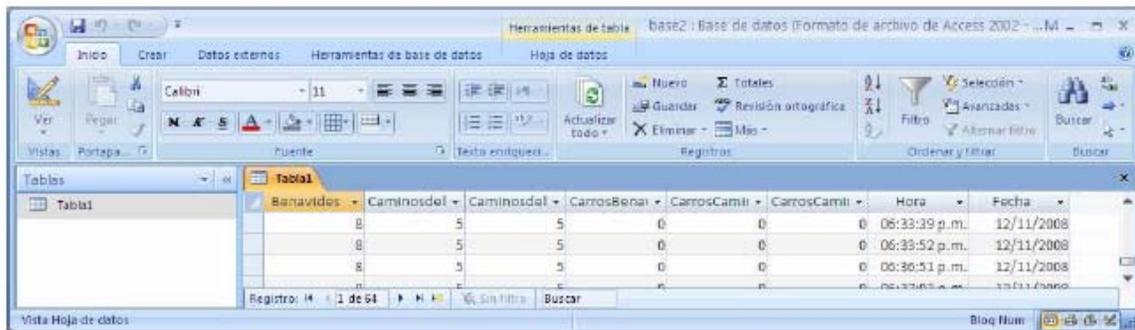


Fig.37 Aplicación para PC para almacenamiento de información

Con la aplicación se obtienen los datos correspondientes a la cantidad de vehículos presentes en la avenida Benavides, la cantidad de vehículos presentes en la avenida Caminos del Inca con dirección a la Bolichera, la cantidad de vehículos presentes en la avenida Caminos del Inca con dirección a San Borja, los tiempos asignados a los tres segmentos de la intersección, además de almacenarse la información con la fecha y la hora en la que se realizó la toma de resultados.

Para el almacenamiento de los datos obtenidos, el programa exporta la información a Access para que en éste se tengan los datos indexados con la información recogida de campo, además de proporcionar la fecha y la hora en el momento que se recogió la

información para poder segmentarla luego por hora o fecha. En la figura 38 se aprecia la base de datos realizada.



Benavides	Caminosdel	Caminosdal	CarrosBenav	CarrosCamli	CarrosCamli	Hora	Fecha
0	5	5	0	0	0	06:33:39 p.m.	12/11/2008
0	5	5	0	0	0	06:35:52 p.m.	12/11/2008
0	5	5	0	0	0	06:36:51 p.m.	12/11/2008

Fig.38 Base de datos generada a partir de la aplicación realizada

2.7 Diseño del sensor en pista

Para el diseño del sensor en pista se trabajó con una manguera para alta presión, en la cual se colocó un aceite especial con las siguientes características:

- Punto de Ebullición: Superior a 260oC
- Punto de Ebullición húmedo: 145oC
- Viscosidad cinemática: 5,74 Centistokes a 50oC
- Valor del PH: 8,5

La finalidad de utilizar el aceite fue de obtener una velocidad de respuesta mayor al momento de detectarse un vehículo, ya que con otros aceites se obtuvo una velocidad de respuesta mucho menor.

Además, se utilizó un detector de presión de aceite, el cual es normalmente abierto. Por ello, cada vez que un vehículo pasa por encima de la manguera hidráulica, el contacto se cierra, permitiendo así poder realizar el conteo de los vehículos.

Para poder realizar la comunicación entre el sensor y la placa de control habrá que acondicionar la señal. Por ello, se envía un pulso cuadrado independientemente del tiempo que pueda estar la rueda sobre el sensor, para así evitar falsas cuentas; además de

acondicionarle un circuito anti rebote para que no haya errores en el conteo. Finalmente, como la detección se realiza por eje, el resultado se debe dividir entre dos, para que la placa de control lo pueda contabilizar como un vehículo.

Las pruebas realizadas para el caso de este sensor fueron preliminares, no siendo éste un objetivo dentro del trabajo de investigación de esta tesis; más sí, una referencia a tomar en cuenta para el desarrollo de uno de los dispositivos periféricos realizados.

En las pruebas realizadas con el sensor, se logró captar sin error vehículos viajando hasta la velocidad de 40 Km/h, pero cuando los vehículos pasan a través del sensor a una velocidad mayor, éste presenta inestabilidad producto de las características físicas del líquido utilizado, es por ello que se requiere de un proceso de llenado especial de la manguera evitando así que se generen burbujas en el interior, ya que éstas distorsionan la sensibilidad de sensado del dispositivo. La figura 39 muestra una fotografía de la manguera utilizada, así como su ubicación sobre la pista para sensar el paso de vehículos.



Fig.39 Sensor en pista

PRUEBAS Y VERIFICACIÓN DE DATOS OBTENIDOS

3.1 Equipo utilizado para la verificación

Para la verificación de los datos obtenidos, se ha utilizado la librería de Matlab Fuzzy, la cual sirve como referencia para corroborar que el algoritmo realiza las operaciones correspondientes a la lógica difusa de manera adecuada, obteniéndose un error mínimo en comparación a los cálculos teóricos.

También se ha optado por utilizar el software Proteus 7 Professional, con la finalidad de corroborar que el algoritmo desarrollado en Matlab y pasado al lenguaje Pic basic sea correcto y no haya desviaciones en los resultados.

Para la verificación de las placas realizadas se utilizó el multímetro digital Fluke, con el cual se hicieron las pruebas de continuidad, medición de voltajes, detección de falsos contactos entre terminales y algunos análisis en las salidas de los pines y las interfaces.

Asimismo, para la verificación del correcto funcionamiento de la interfaz de comunicación RS-232 se utilizó la aplicación Hyper Terminal del Sistema Operativo Windows, aplicando las propiedades de transmisión establecidas por la placa de control (ver la figura 40).

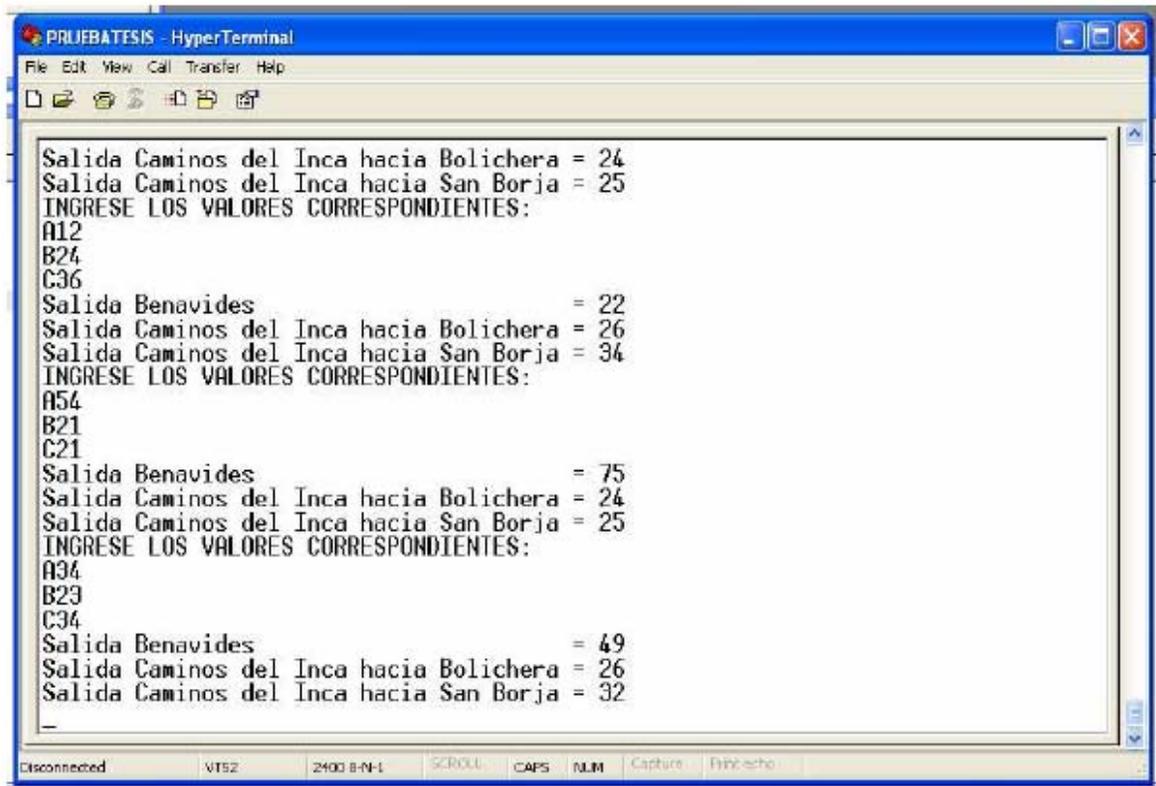


Fig.40 Pruebas en Hyper Terminal

También se realizaron pruebas de los puertos de los microcontroladores con programas de prueba, los cuales se realizaron para verificar inicialmente el funcionamiento del LCD, de la comunicación entre microcontroladores y la salida de los puertos.

3.2 Datos y Error Máximo Obtenidos de los Datos del Sistema de Control

La comparación de la información fue realizada con parte de la muestra de campo obtenida durante la semana en donde se hicieron las mediciones en la intersección objeto de la investigación. Los gráficos obtenidos de dichas muestras se encuentran en el Capítulo II sección 2.2.

Las variables lingüísticas elaboradas a partir de las muestras, se verificaron en el software Matlab para ir afinando los valores de los conjuntos. Finalmente, se optó por implementar tres conjuntos difusos para las variables de entrada y tres para las de salida, para el caso de

la avenida Caminos del Inca en sus dos direcciones; y, cuatro conjuntos difusos para las entradas y cuatro de salida para el caso de la avenida Benavides, con un total de 36 reglas. La figura 41 muestra una ventana capturada del GUI Fuzzy del Matlab, donde se exhiben bloques simbolizando las tres variables de entrada y las tres variables de salida, del controlador difuso.

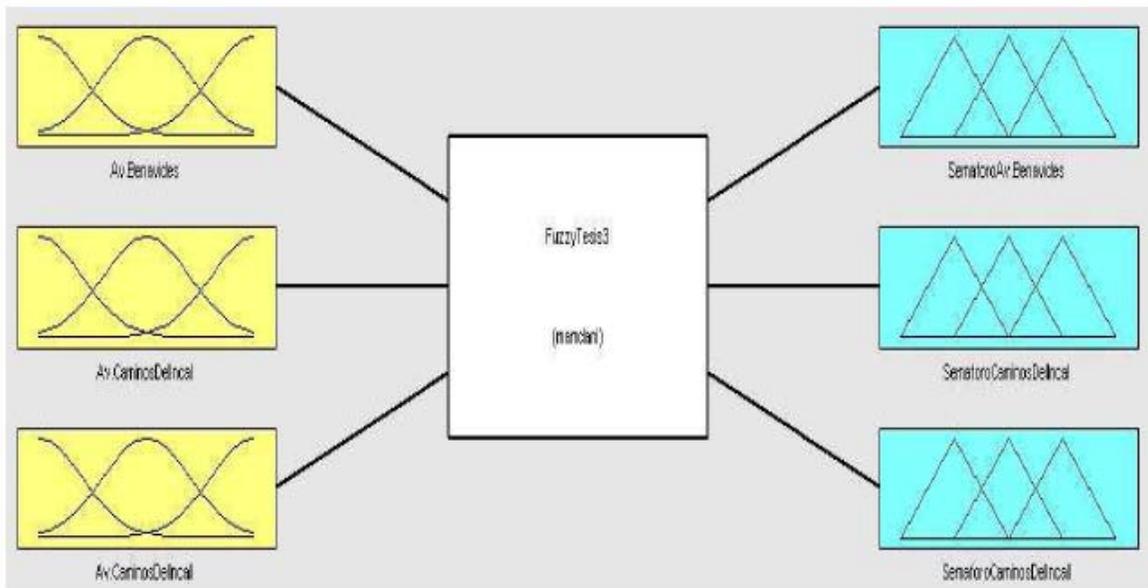


Fig.42 Bloques de entrada y salida del sistema de lógica difusa

Asimismo, las figuras 43, 44, 45, 46, 47 y 48 representan a cada variable de entrada al controlador difuso, indicando los tipos, rangos, nombres y cantidad de conjuntos difusos utilizados, por cada una de ellas.

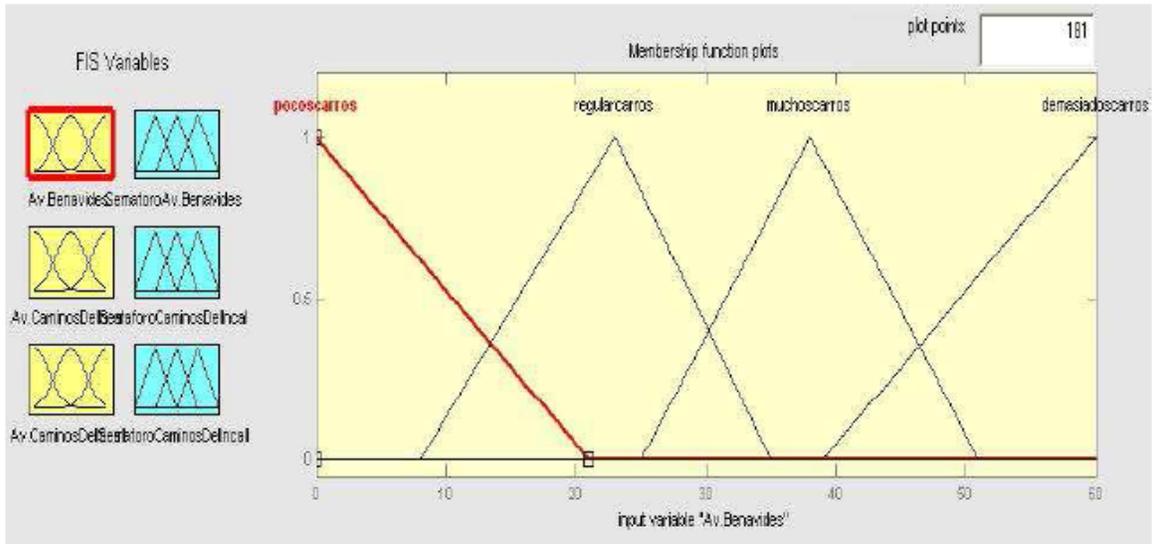


Fig.43 Conjuntos difusos de la variable de entrada – Av. Benavides

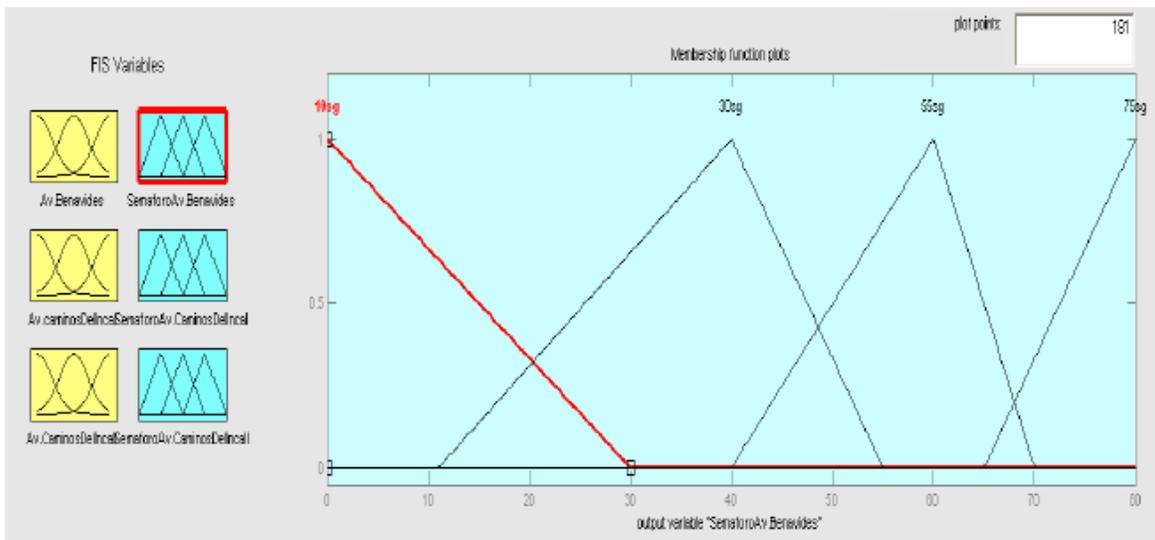


Fig.44 Conjuntos difusos de la variable de salida – Av. Benavides

En las variables lingüísticas de la avenida Benavides se consideró un mayor número de conjuntos difusos, debido a que la cantidad de vehículos presentes en esta vía es mayor a los que se encuentran en las otras vías que llegan a la intersección, por lo que necesita de un tratamiento más profundo.

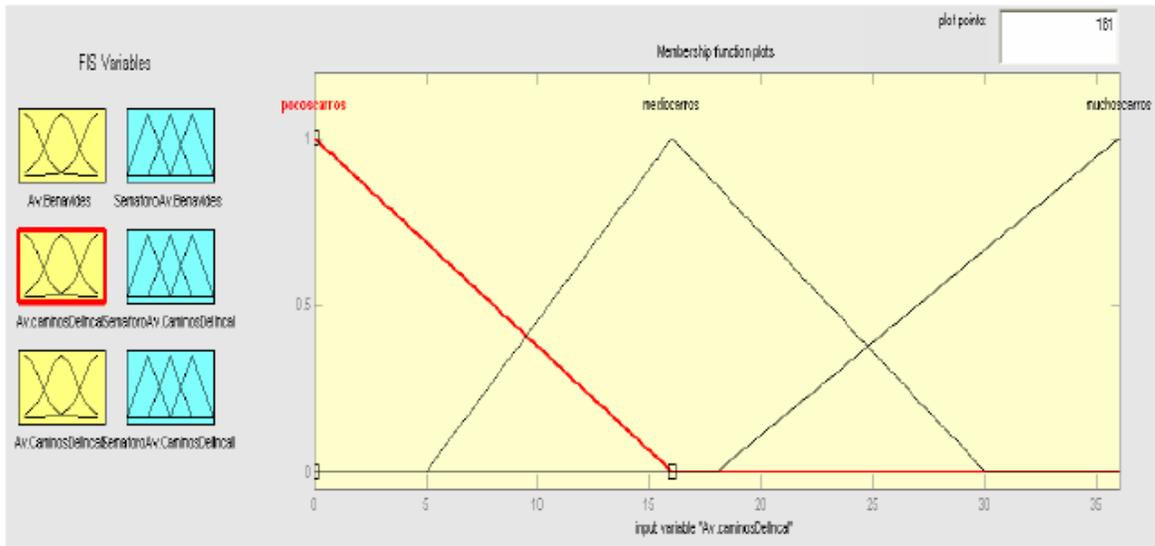


Fig.45 Conjuntos difusos de la variable de entrada – Av. Caminos del Inca hacia la Bolichera

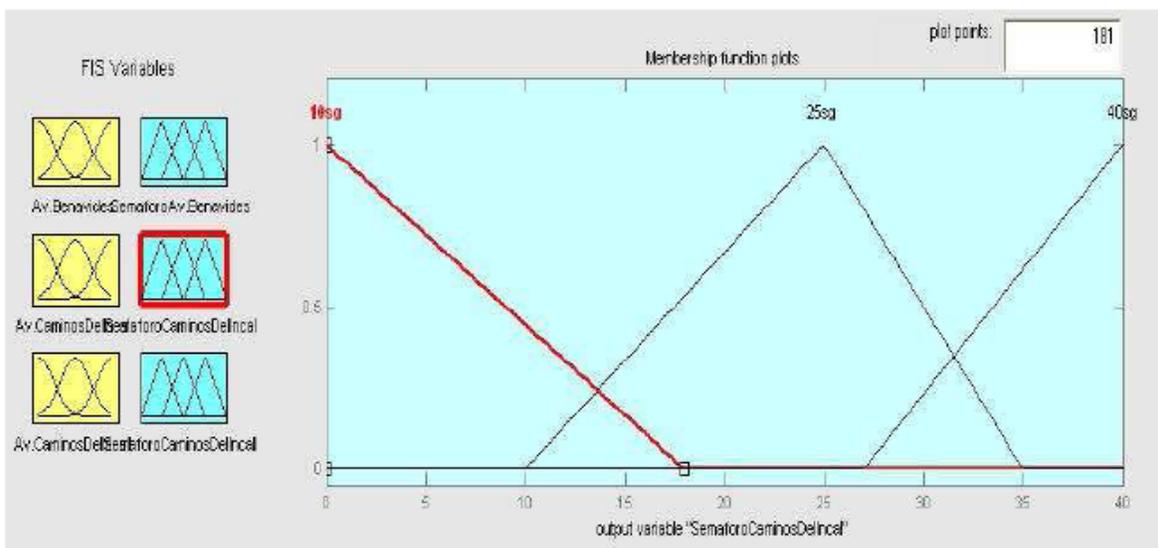


Fig.46 Conjuntos difusos de la variable de salida – Av. Caminos del Inca hacia la Bolichera

En el caso de las variables lingüísticas de entrada y de salida de la avenida Caminos del Inca con dirección a la Bolichera, se aprecia que la cantidad de vehículos considerada para el cálculo de las salidas es menor al resto, debido a que la afluencia de vehículos en esta vía es menor en relación a las otras vías que llegan a la intersección.

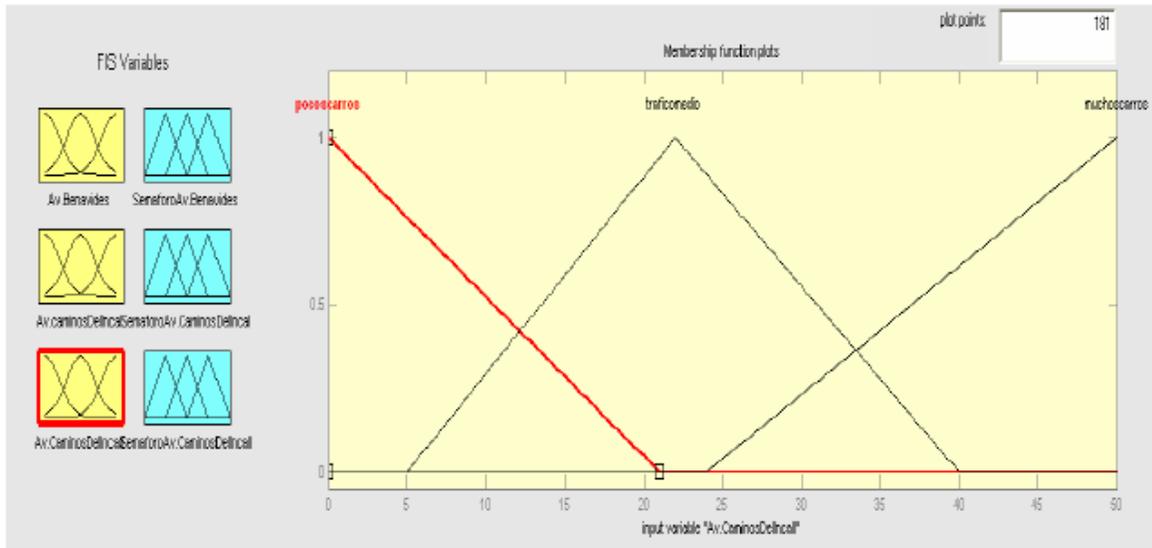


Fig.47 Conjuntos difusos de la variable de entrada - Av. Caminos del Inca hacia San Borja

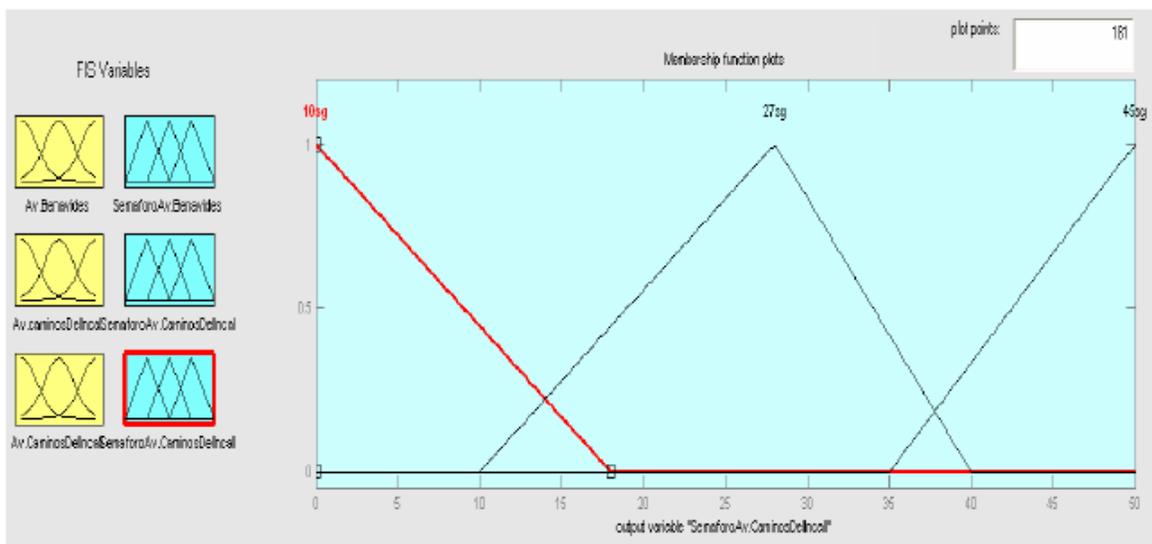


Fig.48 Conjuntos difusos de la variable de salida – Av. Caminos del Inca hacia San Borja

En la avenida Caminos del Inca con dirección hacia San Borja, se tiene una afluencia de vehículos regular, teniéndose picos en la noche y en menor intensidad en la mañana, pero los picos que se producen en las horas punta de la tarde son críticos para el sistema, ya que

son en estos momentos en donde realmente el sistema es más eficiente en relación al sistema convencional.

Por otro lado, la figura 49, que corresponde a una captura de una de las ventanas del GUI Fuzzy del Matlab, muestra todo el conjunto de reglas difusas, relacionando variables de entradas y variables de salidas.

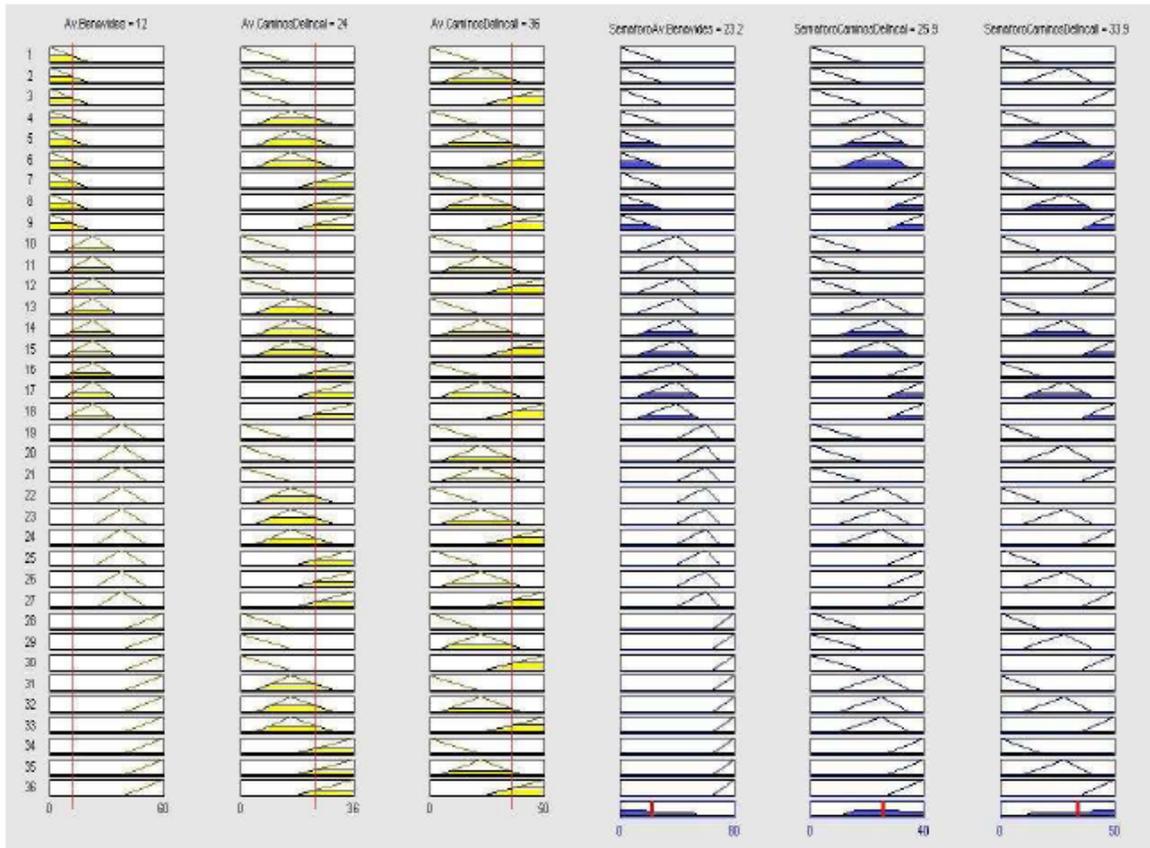


Fig.49 Conjunto de reglas de la lógica difusa del sistema

El algoritmo aplicado al microcontrolador en relación a la interfaz gráfica para el usuario GUI Fuzzy del Matlab utilizado, presenta un error del 1.33%, lo cual equivale a un segundo en promedio para todos los casos aplicados al sistema, lo que muestra que el algoritmo es bastante confiable.

Este tiempo promedio de un segundo, se obtuvo al realizar un cálculo entre un promedio de los tiempos obtenidos mediante el cálculo desarrollado en el toolbox Fuzzy de Matlab,

tomándose este valor como teórico, y el promedio de los tiempos obtenidos mediante el algoritmo desarrollado para el microcontrolador, siendo este el valor experimental.

Interpolando las curvas características de las avenidas con los resultados obtenidos mediante el uso del algoritmo, se obtuvieron las siguientes tablas (tabla 2 y tabla 3) tomándose en cuenta la cantidad de vehículos que entran a la intersección, la cantidad de vehículos que salen de la intersección y la cantidad de tiempo que el semáforo debería estar en verde, esto aplicado a cada una de las vías trabajadas.

Benavides			Bolichera			San Borja		
Entran	Salen	Tiempo en Verde	Entran	Salen	Tiempo en verde	Entran	Salen	Tiempo en Verde
17	19	32	14	17	21	18	17	24
11	12	23	16	18	23	21	18	26
12	14	25	18	19	23	24	20	26
13	18	26	16	23	23	22	38	26
15	20	30	21	27	25	24	40	26
17	22	32	15	23	22	25	41	26
22	27	35	15	23	23	21	37	26
15	20	29	25	28	27	22	38	26
12	17	25	22	27	25	29	44	29
21	26	35	18	25	23	24	40	26
19	24	33	29	29	33	14	28	21
21	26	35	15	23	22	27	42	28
21	26	35	21	27	25	23	39	26
22	27	35	30	29	36	23	39	26
22	27	35	14	22	21	30	45	29
18	23	33	29	29	33	25	41	26
15	20	30	31	30	35	25	41	26
23	28	34	25	28	27	25	41	26
17	22	31	23	27	26	34	47	32

Benavides			Bolichera			San Borja		
Entran	Salen	Tiempo en Verde	Entran	Salen	Tiempo en verde	Entran	Salen	Tiempo en Verde
31	33	43	13	21	20	21	37	26
29	32	40	14	22	21	26	42	27
16	21	31	16	23	23	38	49	45
34	35	49	13	21	21	24	40	26
37	36	56	10	17	18	29	44	29
20	25	34	17	24	23	39	49	41
24	28	35	20	26	24	38	49	38
16	21	31	31	30	36	23	39	26
22	27	35	28	29	31	31	46	30
33	34	47	13	21	20	30	45	29
29	32	40	17	24	23	25	41	26
23	28	35	20	26	24	39	49	41
18	23	33	34	30	36	28	43	28
50	40	70	9	16	17	48	50	44
32	34	46	20	26	24	28	43	28
26	30	37	16	23	23	41	50	45
24	28	35	22	27	25	36	48	35
44	39	60	12	20	20	28	43	28
26	30	36	28	29	31	26	42	27
18	23	32	28	29	31	41	50	45
45	39	61	26	28	28	20	35	25
18	23	33	17	24	23	58	47	45
23	28	34	20	26	24	35	48	33
32	34	46	19	25	24	29	44	29
28	31	38	22	27	25	24	40	26
31	33	44	21	27	25	38	49	38
24	28	35	27	29	29	38	49	38
34	35	50	26	28	28	28	43	29
31	33	43	34	30	35	26	42	27
25	29	35	29	29	33	43	50	45
42	38	58	31	30	36	23	39	26
44	39	60	20	26	24	32	46	30

34	35	49	16	23	23	48	50	45
14	19	28	34	30	35	54	49	44
21	26	35	35	30	36	40	50	45
28	31	38	30	29	36	43	50	45
50	40	71	32	30	35	23	39	26
43	38	60	10	17	18	49	50	45
25	29	35	30	29	36	51	50	45
55	41	76	12	20	20	45	50	45
33	34	46	19	25	24	54	49	45
50	40	71	34	30	35	49	50	45

Benavides			Bolichera			San Borja		
Entran	Salen	Tiempo en Verde	Entran	Salen	Tiempo en verde	Entran	Salen	Tiempo en Verde
50	40	71	33	30	35	55	48	45

Tabla 2. Resultados tiempos y cantidad de vehículos salientes

Tiempo tomado para 62 periodos de semáforo utilizando la lógica difusa	6217
Tiempo tomado para 62 periodos de semáforo utilizando el sistema convencional	6448
Diferencia (segundos)	231

Tabla 3. Resultado acumulado tiempos

En esta última tabla se puede apreciar que por cada 62 periodos de semáforo se produce un ahorro de tiempo acumulado de 231 segundos en comparación al sistema de tráfico vehicular instalado actualmente, lo que significa un ahorro significativo si se toma en cuenta que el sistema funciona las 24 horas del día y esta relación es por aproximadamente 1.75 horas.

A continuación, las figuras 50, 51, 52, 53 y 54, representan las curvas de relación y eficiencia encontradas en las avenidas Benavides, Caminos del Inca hacia la Bolichera y Caminos del Inca hacia San Borja. Las figuras 50, 52 y 54 presentan en el eje horizontal el número de muestra y en el eje vertical la cantidad de vehículos en unidades, describiéndose la cantidad de vehículos que entran y salen de cada una de las intersecciones. Las figuras

51, 53 y 55 presentan en el eje horizontal el número de muestra y en el eje vertical el porcentaje de eficiencia del sistema aplicado a cada muestra.

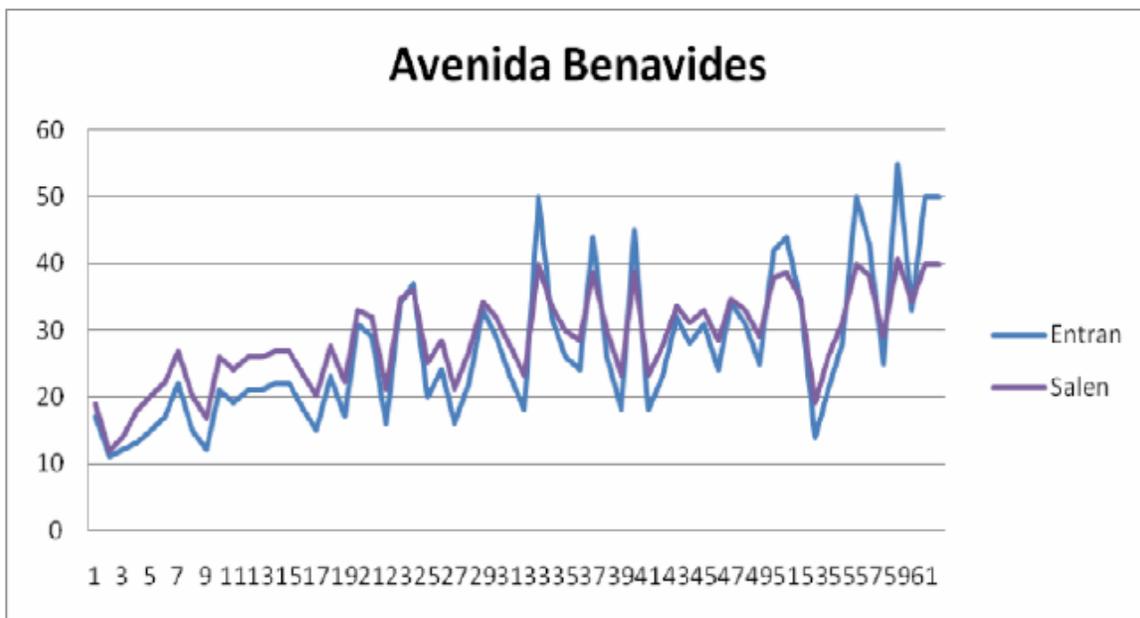


Fig.50 Relación de vehículos que entran y salen de la Intersección por parte de la avenida Benavides aplicando la lógica difusa

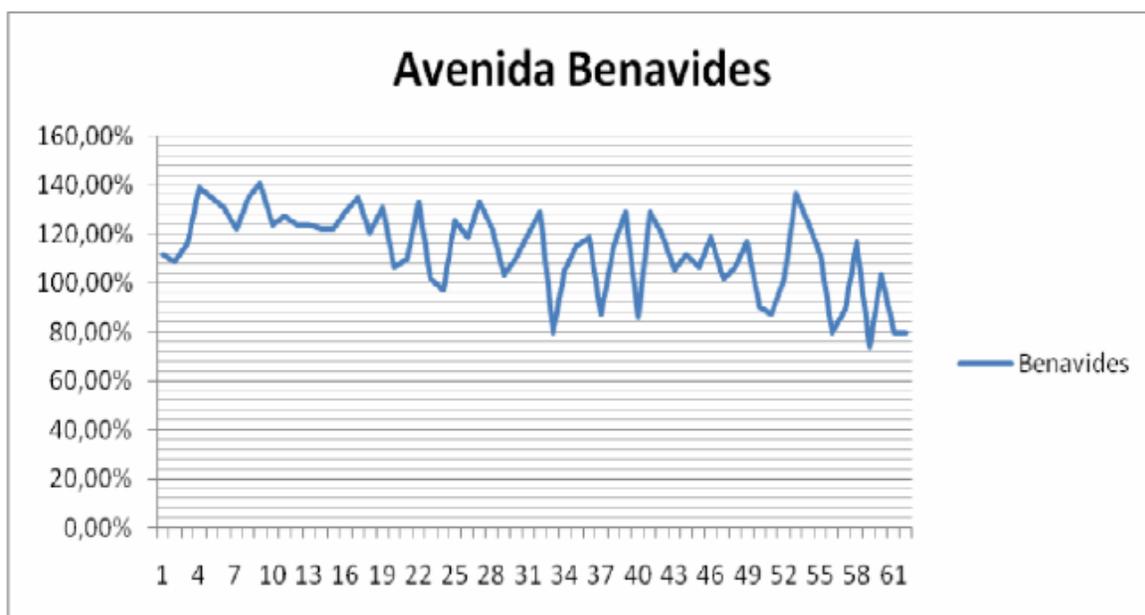


Fig.51 Eficiencia de la Intersección por parte de la avenida Benavides aplicando la lógica difusa

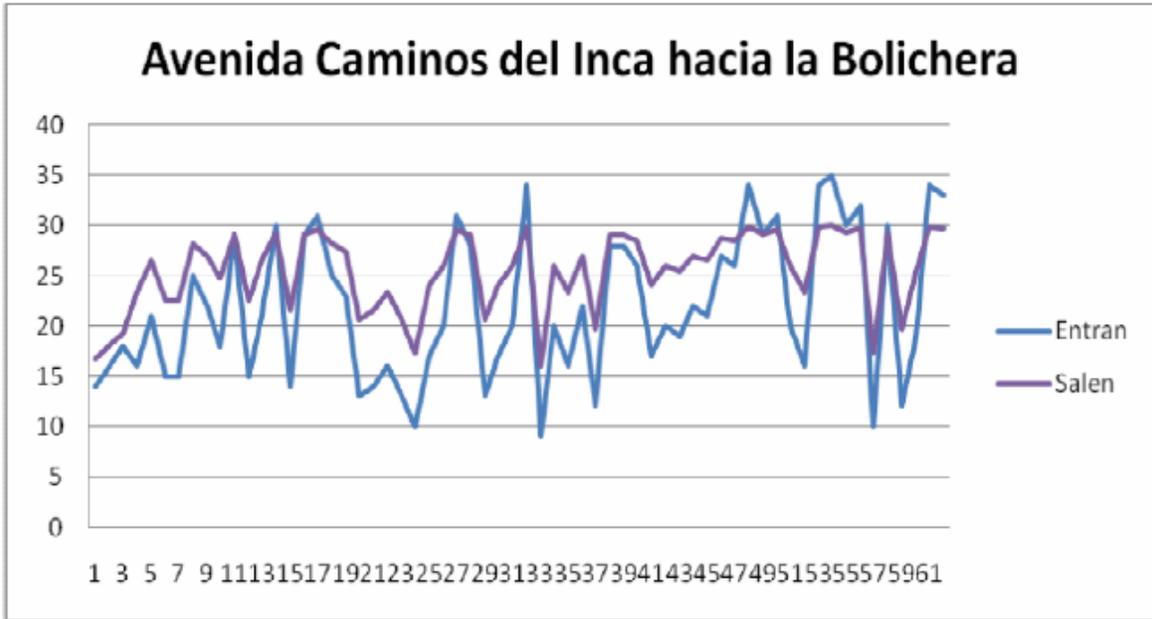


Fig.52 Relación de vehículos que entran y salen de la Intersección por parte de la avenida Caminos del Inca hacia la Bolichera aplicando la lógica difusa

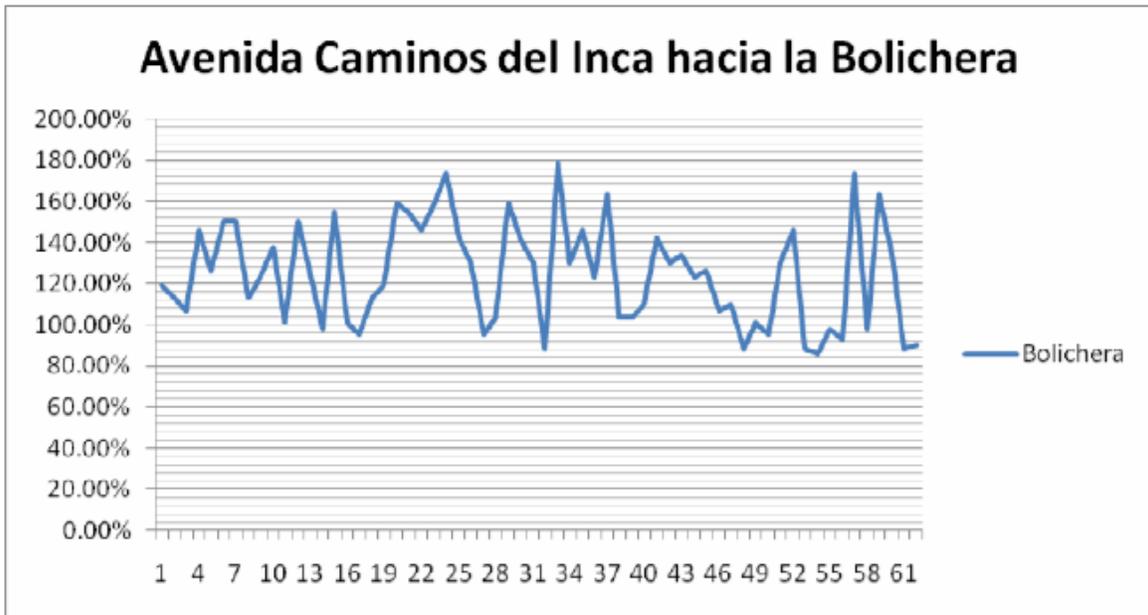


Fig.53 Eficiencia de la Intersección por parte de la avenida Caminos del Inca hacia la Bolichera aplicando la lógica difusa

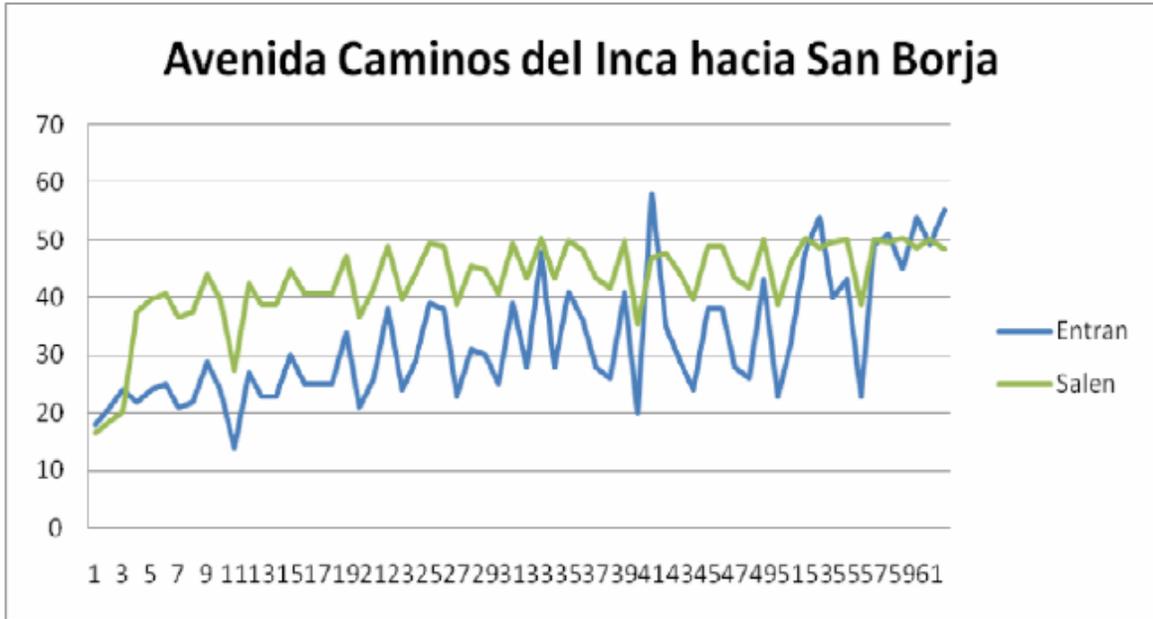


Fig.54 Relación de vehículos que entran y salen de la Intersección por parte de la avenida Caminos del Inca hacia la San Borja aplicando la lógica difusa

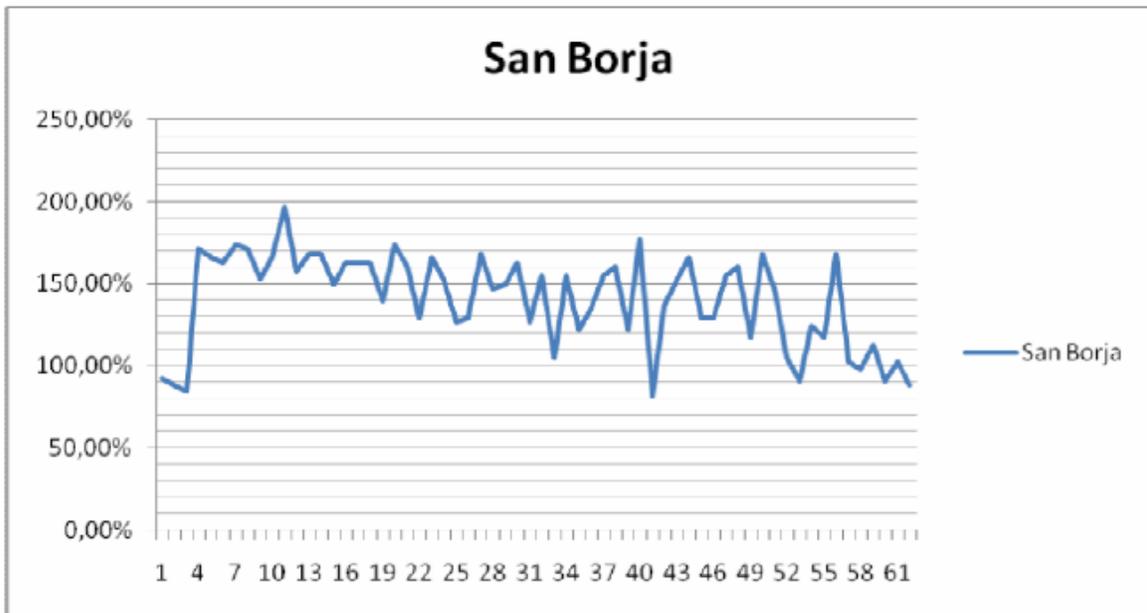


Fig.55 Eficiencia de la Intersección por parte de la avenida Caminos del Inca hacia la San Borja aplicando la lógica difusa

En las figuras 50, 52 y 54 se observa que la cantidad de vehículos que entran y salen de la intersección de las avenidas se encuentra optimizada, permitiendo la salida de vehículos de

la intersección en tiempos más apropiados, disminuyendo en gran medida los picos que se producen con el sistema actual instalado.

Se observa que en promedio las eficiencias para cada uno de los casos es del 133.55% para el caso de la avenida Benavides, 124.66% para el caso de la avenida Caminos del Inca hacia la Bolichera y 140.40% para el caso de la avenida Caminos del Inca hacia San Borja, con lo que la intersección se ve mejorada en relación a las eficiencias obtenidas por el sistema anterior, las cuales son del 113% para el caso de la avenida Benavides, 105% para el caso de la avenida Caminos del Inca hacia la Bolichera y 79% para el caso de la avenida Caminos del Inca hacia San Borja.

Cabe resaltar que en estos resultados se está combinando las situaciones de alto tráfico y tráfico medio, por lo que los resultados en eficiencia aparentan ser relativamente altos en ambos sistemas, pero que en realidad los puntos críticos se encuentran en las horas punta del día, en especial entre las 6:30 pm y 8:30 pm.

CONCLUSIONES Y PERSPECTIVAS

4.1 Conclusiones

En base a los resultados obtenidos se pueden citar las siguientes conclusiones:

- El sistema opera de manera adecuada para los requerimientos obtenidos, mediante el análisis estadístico obtenido de las salidas de campo para el conteo de vehículos. Se podrían haber obtenido mejores resultados de haber podido realizarse un estudio más profundo de la intersección, considerando más horas del día para las muestras y por lo menos haber hecho el estudio de la intersección por un mes entero, pudiéndose obtener resultados más definidos con respecto a los niveles de congestión vehicular en la intersección.
- En relación a los dispositivos para el conteo de vehículos, se debe realizar investigaciones más profundas sobre el sensor de pista y considerar además un mecanismo de protección de la manguera hidráulica para así poder evitar un desgaste prematuro del sensor, además de poder mejorarse la sensibilidad a mayores velocidades y así poder contabilizar los vehículos que circulen a más de 40 Km/h, poniéndose como objetivo poder llegar a los 65 Km/h como mínimo.
- La fuente de alimentación opera de manera adecuada, brindando los tres niveles de voltaje requeridos por el sistema, teniendo un nivel de estabilidad de la calidad del voltaje entregado adecuado y con los márgenes de protección debidamente establecidos.
- La placa principal de control opera de manera adecuada para las condiciones del sistema, comprobándose que no hay interferencias ocasionadas por los diferentes niveles de voltajes presentes en la placa.
- La diferencia que se produce entre el sistema desarrollado en base al GUI fuzzy y el algoritmo final desarrollado, se debe a que el microcontrolador utilizado no realiza operaciones en punto flotante, sino en enteros, por lo que se realizó un artificio para así poder cubrir dos cifras decimales, pero aún así se pierde efectividad en el mismo.

- Si se deseara mejorar el nivel de error del algoritmo para poder reducirlo, hay dos opciones: poder optar por utilizar operaciones en punto flotante, lo cual haría necesaria la presencia de un microprocesador adicional que realice estas operaciones, probablemente debiendo emigrar al sistema basado en FPGAs; o pudiendo realizar un artificio para ya no considerar solo dos dígitos de manera inicial, sino utilizando tres dígitos (por ejemplo si se tiene el número 0.125 multiplicarlo por 1000 para así poder trabajar en base al número 125, y luego finalmente dividirlo entre 1000 para así tener el resultado real).
- En el caso de la aplicación para la computadora para el almacenamiento de la información, éste es un programa básico de procesamiento del mismo para así poder visualizar de manera simple los resultados obtenidos y la cantidad de vehículos presentes en la intersección.
- El sistema trabaja con focos de Leds, por lo que se puede apreciar un ahorro de energía de entre el 92.5% y el 96.7%, lo que representa una cantidad importantísima de energía ahorrada simplemente por un cambio en el sistema de luces, ya que el sistema tradicional trabaja en base a focos de 135 Watts, mientras este sistema opera en base a focos de entre 4.4 a 10 Watts.
- Se puede optar por utilizar en lugar de sensores en pista un sistema de conteo de vehículos basado en procesamiento digital de imágenes, pudiéndose integrar al sistema mediante el uso de DSPIC's, procesando éstos la información gráfica y mandando el resultado a la placa de control principal.

4.2 Trabajo a futuro

Para poder mejorar este sistema y poderlo hacer más confiable sería necesario realizar los siguientes trabajos a futuro:

- Poder emigrar al sistema de un diseño compacto a un diseño modular, separando la placa de control principal de la parte de control de luces y de la parte de cuenta regresiva de la intersección.

- Por ser un sistema que va a operar las 24 horas del día sería necesario poder realizar un sistema que posea redundancia para así poder tener la certeza de que si uno de los componentes falla, habrá otro que pueda suplir sus funciones para que el sistema pueda seguir operando de manera ininterrumpida.
- En el caso de la fuente de alimentación, también se debe realizar un sistema redundante para así poder tener una fuente ininterrumpida de energía en caso de una caída de la red, pudiendo integrar un banco de baterías con un cargador a la fuente de alimentación, agregando así a la fuente de alimentación un cargador de baterías y un convertidor DC-AC para poder alimentar con las baterías la parte del sistema que opera a 220V AC.
- Poder seguir trabajando con el diseño del sensor en pista, con la posibilidad de migrar a un sensor basado en procesamiento digital de señales para así poder abaratar los costos de instalación del sistema y que pueda ser más confiable en el tiempo, ya que los sistemas basados en procesamiento de imágenes no entran en contacto directo con el vehículo, desgastándolo como en el caso de los sensores en pista.
- Poder mejorar la aplicación para que pueda procesar mayor cantidad de información y que pueda realizar estadísticas de manera automática, colgando luego estos resultados en una dirección web para que esta información pueda ser accesada y utilizada por el público en general.
- Finalmente, investigar más a profundidad sobre los estándares internacionales existentes y tratar de adoptar alguno de ellos para así poder tener finalmente un producto de exportación.

REFERENCIAS BIBLIOGRÁFICAS

Fuentes Bibliográficas

- [1] http://es.wikipedia.org/wiki/Tr%C3%A1nsito_vehicular, revisado en setiembre del 2008
- [2] http://en.wikipedia.org/wiki/Traffic_light#History, revisado en octubre del 2008
- [3] http://members.tripod.com/jesus_alfonso_lopez/FuzzyIntro2.html, revisado en octubre del 2008
- [4] <http://www.itq.edu.mx/vidatec/espacio/aiee/fuzzy.ppt>. Revisado en octubre del 2008
- [5] http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/sbaa/report.traff.html, Shahariz Abdul Aziz, Jeyakody Parthiban, 1996. Revisado en octubre del 2008
- [6] <http://www.ifent.org/lecciones/varistores/>. Revisado en octubre del 2008
- [7] http://www.cantherm.com/products/thermistors/choosing_ntc.html. Revisado en octubre del 2008
- [8] <http://www.tscm.com/rs-232.pdf>. Revisado en octubre del 2008
- [9] <http://www.best-microcontroller-projects.com/image-files/how-rs232-works-txdiag.png>. Revisado en setiembre del 2008
- [10] <http://www.tecnovial-sac.com/w4.swf>. Revisado en agosto del 2008
- [11] <http://www.tec-corp.com.pe/semaforizacion.htm>. Revisado en agosto del 2008
- [12] <http://www.roadtraffic-technology.com/contractors/detection/>. Revisado en agosto del 2008
- [13] <http://www.semex.com.mx/inicio.html>. Revisado en julio del 2008
- [14] <http://www.dot.state.sc.us/getting/signals/loopdet.shtml>. Revisado en agosto del 2008
- [15] <http://www.sensourceinc.com/TC-MS30.htm>. Revisado en julio del 2008
- [16] <http://www.vehicle-counters.com/>. Revisado den junio del 2008
- [17] <http://www.exemys.com>. Revisado en agosto del 2008
- [18] <http://www.tfhr.gov/pubrds/septoct98/loop.htm>. Revisado en julio del 2008
- [19] http://www.digital-measure.com/html/lc_meter.htm. Revisado en julio del 2008
- [20] <http://www.tectraffic.nl/en/homepage>. Revisado en setiembre del 2008
- [21] <http://www.datasheetcatalog.com>. Revisado en octubre del 2008
- [22] http://es.wikipedia.org/wiki/L%C3%B3gica_difusa. Revisado en octubre del 2008
- [23] <http://www.seattlerobotics.org/encoder/mar98/fuz/flindex.html>. Revisado en setiembre del 2008

[24] <http://blog.pucp.edu.pe/item/24976>. Revisado en octubre del 2008

[25] http://www.ercim.org/publication/Ercim_News/enw65/varga.html. Revisado en
setiembre del 2008

APÉNDICES

A. Costo del equipo de control de tráfico vehicular

Placa Principal	Cantidad	Costo unitario (S/.)	Total (S/.)
PIC 18F458	1	80.00	80.00
PIC 16F877	1	24.00	24.00
Max 232	1	10.00	10.00
KSP2222A	64	0.40	25.60
MOC3041	16	2.00	32.00
BT131	16	0.50	8.00
LCD Display	1	20.00	20.00
74LS48	1	3.50	3.50
Placa Control Principal	1	110.00	110.00
(Resistencias, Condensadores, etc.)	1	80.00	80.00
Cable Serial	1	5.00	5.00
Diseño e implementación	1	1600.00	1600.00
Total			1998.10

Tabla A1. Costos de implementación de la placa principal

Fuente Alimentación	Cantidad	Costo unitario (S/.)	Total (S/.)
Transformador	1	25.00	25.00
LM350	1	24.00	24.00
LM7805	1	5.00	5.00
1N4007	4	0.50	2.00
Varistor	1	2.50	2.50
NTC	1	2.00	2.00
LCD Display	1	20.00	20.00
74LS48	1	3.50	3.50
Placa Fuente Alimentación	1	50.00	50.00
(Resistencias, Condensadores, etc.)	1	20.00	20.00
Diseño e implementación	1	300.00	300.00
Total			454.00

Tabla A2. Costos de implementación de la fuente alimentación

Programa de Aplicación	Cantidad	Costo unitario (S/.)	Total
Diseño e implementación	1	200.00	200.00
Total			200.00

Tabla A3. Costos de diseño del programa de aplicación

Implementación en General	Cantidad	Costo unitario (S/.)	Total (S/.)
Placa Principal	1	1998.10	1998.10
Fuente Alimentación	1	454.00	454.00
Programa de Aplicación	1	200.00	200.00
Total			2652.10

Tabla A4. Costos totales por desarrollo del sistema

B. Hojas Técnicas

PIC18FXX8

1.0 DEVICE OVERVIEW

This document contains device specific information for the following devices:

- PIC18F248
- PIC18F258
- PIC18F448
- PIC18F468

These devices are available in 28-pin, 40-pin and 44-pin packages. They are differentiated from each other in four ways:

1. PIC18FXX8 devices have twice the Flash program memory and data RAM of PIC18FX48 devices (32 Kbytes and 1536 bytes vs. 16 Kbytes and 768 bytes, respectively).

2. PIC18F2X8 devices implement 5 A/D channels, as opposed to 8 for PIC18F4X8 devices.
3. PIC18F2X8 devices implement 3 I/O ports, while PIC18F4X8 devices implement 5.
4. Only PIC18F4X8 devices implement the Enhanced CCP module, analog comparators and the Parallel Slave Port.

All other features for devices in the PIC18FXX8 family, including the serial communications modules, are identical. These are summarized in Table 1-1.

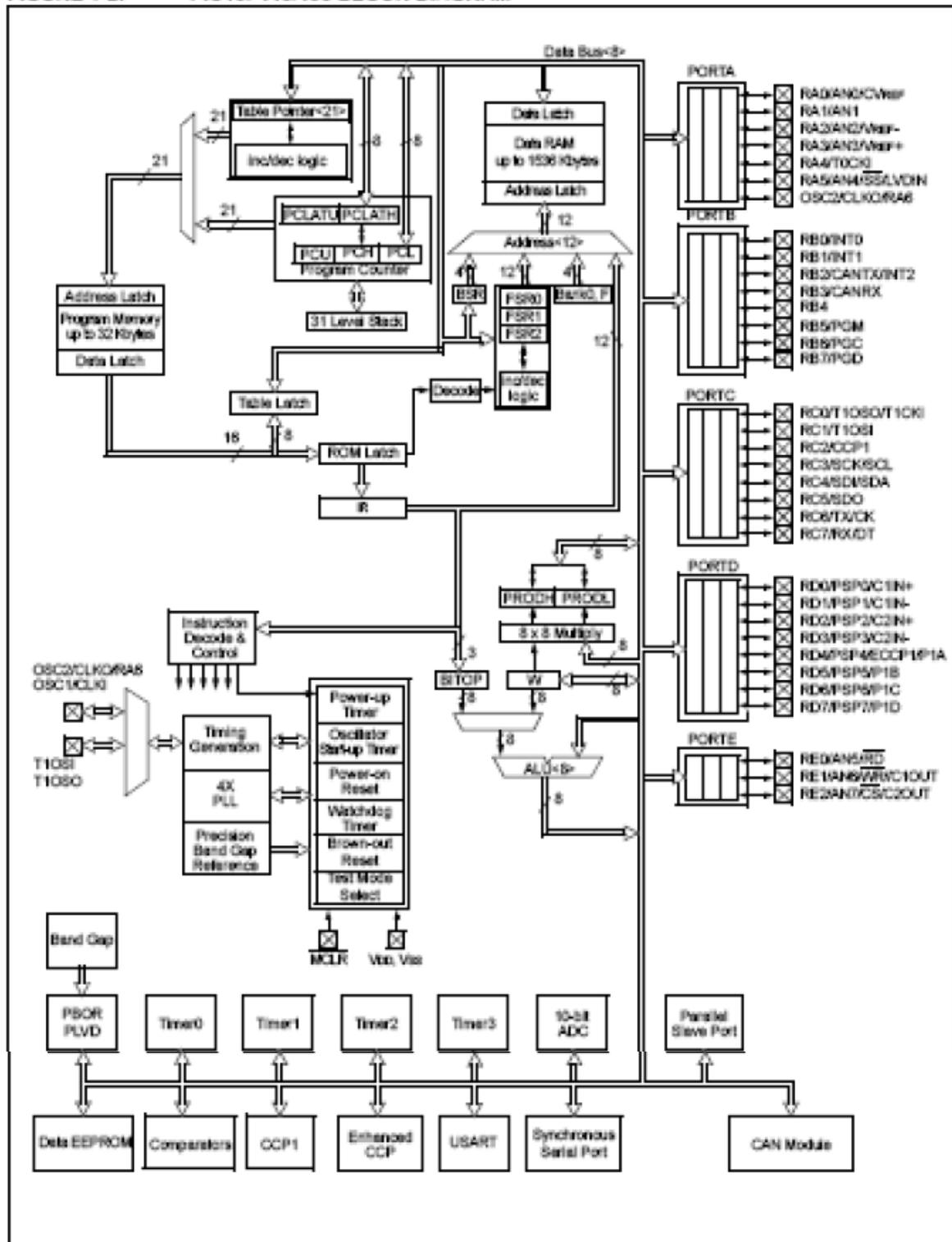
Block diagrams of the PIC18F2X8 and PIC18F4X8 devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2.

TABLE 1-1: PIC18FXX8 DEVICE FEATURES

Features		PIC18F248	PIC18F258	PIC18F448	PIC18F468
Operating Frequency		DC – 40 MHz	DC – 40 MHz	DC – 40 MHz	DC – 40 MHz
Internal Program Memory	Bytes	16K	32K	16K	32K
	# of Single-Word Instructions	8192	16384	8192	16384
Data Memory (Bytes)		768	1536	768	1536
Data EEPROM Memory (Bytes)		256	256	256	256
Interrupt Sources		17	17	21	21
I/O Ports		Ports A, B, C	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C, D, E
Timers		4	4	4	4
Capture/Compare/PWM Modules		1	1	1	1
Enhanced Capture/Compare/PWM Modules		—	—	1	1
Serial Communications		MSSP, CAN, Addressable USART	MSSP, CAN, Addressable USART	MSSP, CAN, Addressable USART	MSSP, CAN, Addressable USART
Parallel Communications (PSP)		No	No	Yes	Yes
10-bit Analog-to-Digital Converter		5 input channels	5 input channels	8 input channels	8 input channels
Analog Comparators		No	No	2	2
Analog Comparators VREF Output		N/A	N/A	Yes	Yes
Resets (and Delays)		POR, BOR	POR, BOR	POR, BOR	POR, BOR
Instruction Set		75 instructions	75 instructions	75 instructions	75 instructions
Packages		28-pin SPDIP 28-pin SOIC	28-pin SPDIP 28-pin SOIC	40-pin PDIP 44-pin PLCC 44-pin TQFP	40-pin PDIP 44-pin PLCC 44-pin TQFP
Programmable Brown-out Reset		Yes	Yes	Yes	Yes
CAN Module		Yes	Yes	Yes	Yes
In-Circuit Serial Programming™ (ICSP™)		Yes	Yes	Yes	Yes

PIC18FXX8

FIGURE 1-2: PIC18F448/458 BLOCK DIAGRAM



PIC18FXX8

TABLE 1-2: PIC18FXX8 PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number				Pin Type	Buffer Type	Description
	PIC18F248/268	PIC18F448/468					
	8 PDIP, 8 OIC	PDIP	TQFP	PLCC			
MCLR/VPP MCLR VPP	1	1	18	2	I P	ST —	Master Clear (Input) or programming voltage (output). Master Clear (Reset) Input. This pin is an active low Reset to the device. Programming voltage Input.
NC	—	—	12, 13, 33, 34	1, 17, 28, 40	—	—	These pins should be left unconnected.
OSC1/CLKI OSC1 CLKI	9	13	30	14	I I	CMOS/ST CMOS	Oscillator crystal or external clock Input. Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; otherwise, CMOS. External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins).
OSC2/CLKO/RA6 OSC2 CLKO RA6	10	14	31	15	O O I/O	— — TTL	Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate. General purpose I/O pin.

Legend: TTL = TTL compatible Input

ST = Schmitt Trigger Input with CMOS levels

I = Input

P = Power

CMOS = CMOS compatible Input or output

Analog = Analog Input

O = Output

OD = Open-Drain (no P diode to VDD)

PIC18FXX8

TABLE 1-2: PIC18FXX8 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number				Pin Type	Buffer Type	Description
	PIC18F248/258		PIC18F448/458				
	8PDIP, 8OIC	PDIP	TQFP	PLCC			
RA0/AN0/CVREF RA0 AN0 CVREF	2	2	19	3	I/O I O	TTL Analog Analog	PORTA is a bidirectional I/O port. Digital I/O. Analog Input 0. Comparator voltage reference output.
RA1/AN1 RA1 AN1	3	3	20	4	I/O I	TTL Analog	Digital I/O. Analog Input 1.
RA2/AN2/VREF- RA2 AN2 VREF-	4	4	21	5	I/O I I	TTL Analog Analog	Digital I/O. Analog Input 2. A/D reference voltage (Low) input.
RA3/AN3/VREF+ RA3 AN3 VREF+	5	5	22	6	I/O I I	TTL Analog Analog	Digital I/O. Analog Input 3. A/D reference voltage (High) input.
RA4/T0CKI RA4 T0CKI	6	6	23	7	I/O I	TTL/OD ST	Digital I/O – open-drain when configured as output. Timer0 external clock input.
RA5/AN4/ \overline{SS} /LVDIN RA5 AN4 \overline{SS} LVDIN	7	7	24	8	I/O I I I	TTL Analog ST Analog	Digital I/O. Analog Input 4. SPI™ slave select input. Low-Voltage Detect input.
RA6							See the OSC2/CLKO/RA6 pin.

Legend: TTL = TTL compatible input
 ST = Schmitt Trigger input with CMOS levels
 I = Input
 P = Power
 CMOS = CMOS compatible input or output
 Analog = Analog input
 O = Output
 OD = Open-Drain (no P diode to VDD)

PIC18FXX8

TABLE 1-2: PIC18FXX8 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number				Pin Type	Buffer Type	Description
	PIC18F248/268	PIC18F448/458					
	8PDIP, 8QIC	PDIP	TQFP	PLCC			
RB0/INT0 RB0 INT0	21	33	8	36	I/O I	TTL ST	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs. Digital I/O. External Interrupt 0.
RB1/INT1 RB1 INT1	22	34	9	37	I/O I	TTL ST	Digital I/O. External Interrupt 1.
RB2/CANTX/INT2 RB2 CANTX INT2	23	35	10	38	I/O O I	TTL TTL ST	Digital I/O. Transmit signal for CAN bus. External Interrupt 2.
RB3/CANRX RB3 CANRX	24	36	11	39	I/O I	TTL TTL	Digital I/O. Receive signal for CAN bus.
RB4	25	37	14	41	I/O	TTL	Digital I/O. Interrupt-on-change pin.
RB5/PGM RB5 PGM	26	38	15	42	I/O I	TTL ST	Digital I/O. Interrupt-on-change pin. Low-voltage ICSP™ programming enable.
RB6/PGC RB6 PGC	27	39	16	43	I/O I	TTL ST	Digital I/O. In-Circuit Debugger pin. Interrupt-on-change pin. ICSP programming clock.
RB7/PGD RB7 PGD	28	40	17	44	I/O I/O	TTL ST	Digital I/O. In-Circuit Debugger pin. Interrupt-on-change pin. ICSP programming data.

Legend: TTL = TTL compatible input

ST = Schmitt Trigger Input with CMOS levels

I = Input

P = Power

CMOS = CMOS compatible input or output

Analog = Analog Input

O = Output

OD = Open-Drain (no P diode to VDD)

PIC18FXX8

TABLE 1-2: PIC18FXX8 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number				Pin Type	Buffer Type	Description
	PIC18F248/258		PIC18F448/458				
	8PDIP, 8OIC	PDIP	TQFP	PLCC			
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	11	15	32	16	I/O O I	ST — ST	PORTC is a bidirectional I/O port. Digital I/O. Timer1 oscillator output. Timer1/Timer3 external clock input.
RC1/T1OSI RC1 T1OSI	12	16	35	18	I/O I	ST CMOS	Digital I/O. Timer1 oscillator input.
RC2/CCP1 RC2 CCP1	13	17	36	19	I/O I/O	ST ST	Digital I/O. Capture 1 Input/Compare 1 output/PWM1 output.
RC3/SCK/SCL RC3 SCK SCL	14	18	37	20	I/O I/O I/O	ST ST ST	Digital I/O. Synchronous serial clock input/output for SPI™ mode. Synchronous serial clock input/output for I ² C™ mode.
RC4/SDI/SDA RC4 SDI SDA	15	23	42	25	I/O I I/O	ST ST ST	Digital I/O. SPI data in. I ² C data I/O.
RC5/SDO RC5 SDO	16	24	43	26	I/O O	ST —	Digital I/O. SPI data out.
RC6/TX/CK RC6 TX CK	17	25	44	27	I/O O I/O	ST — ST	Digital I/O. USART asynchronous transmit. USART synchronous clock (see RX/DT).
RC7/RX/DT RC7 RX DT	18	26	1	29	I/O I I/O	ST ST ST	Digital I/O. USART asynchronous receive. USART synchronous data (see TX/CK).

Legend: TTL = TTL compatible input

ST = Schmitt Trigger input with CMOS levels

CMOS = CMOS compatible input or output

Analog = Analog input

PIC18FXX8

TABLE 1-2: PIC18FXX8 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number				Pin Type	Buffer Type	Description
	PIC18F248/268		PIC18F448/468				
	8PDI, 8OIC	PDIP	TQFP	PLCC			
RD0/P3P0/C1IN+ RD0 P3P0 C1IN+	—	19	38	21	I/O I/O I	ST TTL Analog	PORTD is a bidirectional I/O port. These pins have TTL input buffers when external memory is enabled. Digital I/O. Parallel Slave Port data. Comparator 1 Input.
RD1/P3P1/C1IN- RD1 P3P1 C1IN-	—	20	39	22	I/O I/O I	ST TTL Analog	
RD2/P3P2/C2IN+ RD2 P3P2 C2IN+	—	21	40	23	I/O I/O I	ST TTL Analog	
RD3/P3P3/C2IN- RD3 P3P3 C2IN-	—	22	41	24	I/O I/O I	ST TTL Analog	
RD4/P3P4/ECCP1/ P1A RD4 P3P4 ECCP1 P1A	—	27	2	30	I/O I/O I/O O	ST TTL ST —	
RD5/P3P5/P1B RD5 P3P5 P1B	—	28	3	31	I/O I/O O	ST TTL —	
RD6/P3P6/P1C RD6 P3P6 P1C	—	29	4	32	I/O I/O O	ST TTL —	
RD7/P3P7/P1D RD7 P3P7 P1D	—	30	5	33	I/O I/O O	ST TTL —	

Legend: TTL = TTL compatible input

ST = Schmitt Trigger input with CMOS levels

I = Input

P = Power

CMOS = CMOS compatible input or output

Analog = Analog input

O = Output

OD = Open-Drain (no P diode to VDD)

PIC18FXX8

TABLE 1-2: PIC18FXX8 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number				Pin Type	Buffer Type	Description
	PIC18F248/258	PIC18F448/458					
	8PDIP, 8OIC	PDIP	TQFP	PLCC			
$\overline{RE0}/AN5/\overline{RD}$ RE0 AN5 \overline{RD}	—	8	25	9	I/O I I	ST Analog TTL	PORTE is a bidirectional I/O port. Digital I/O. Analog input 5. Read control for Parallel Slave Port (see \overline{WR} and \overline{CS} pins).
$\overline{RE1}/AN6/\overline{WR}/C1OUT$ RE1 AN6 \overline{WR} C1OUT	—	9	26	10	I/O I I O	ST Analog TTL Analog	Digital I/O. Analog input 6. Write control for Parallel Slave Port (see \overline{CS} and \overline{RD} pins). Comparator 1 output.
$\overline{RE2}/AN7/\overline{CS}/C2OUT$ RE2 AN7 \overline{CS} C2OUT	—	10	27	11	I/O I I O	ST Analog TTL Analog	Digital I/O. Analog input 7. Chip select control for Parallel Slave Port (see \overline{RD} and \overline{WR} pins). Comparator 2 output.
VSS	19, 8	12, 31	6, 29	13, 34	—	—	Ground reference for logic and I/O pins.
VDD	20	11, 32	7, 28	12, 35	—	—	Positive supply for logic and I/O pins.

Legend: TTL = TTL compatible input

ST = Schmitt Trigger input with CMOS levels

I = Input

P = Power

CMOS = CMOS compatible input or output

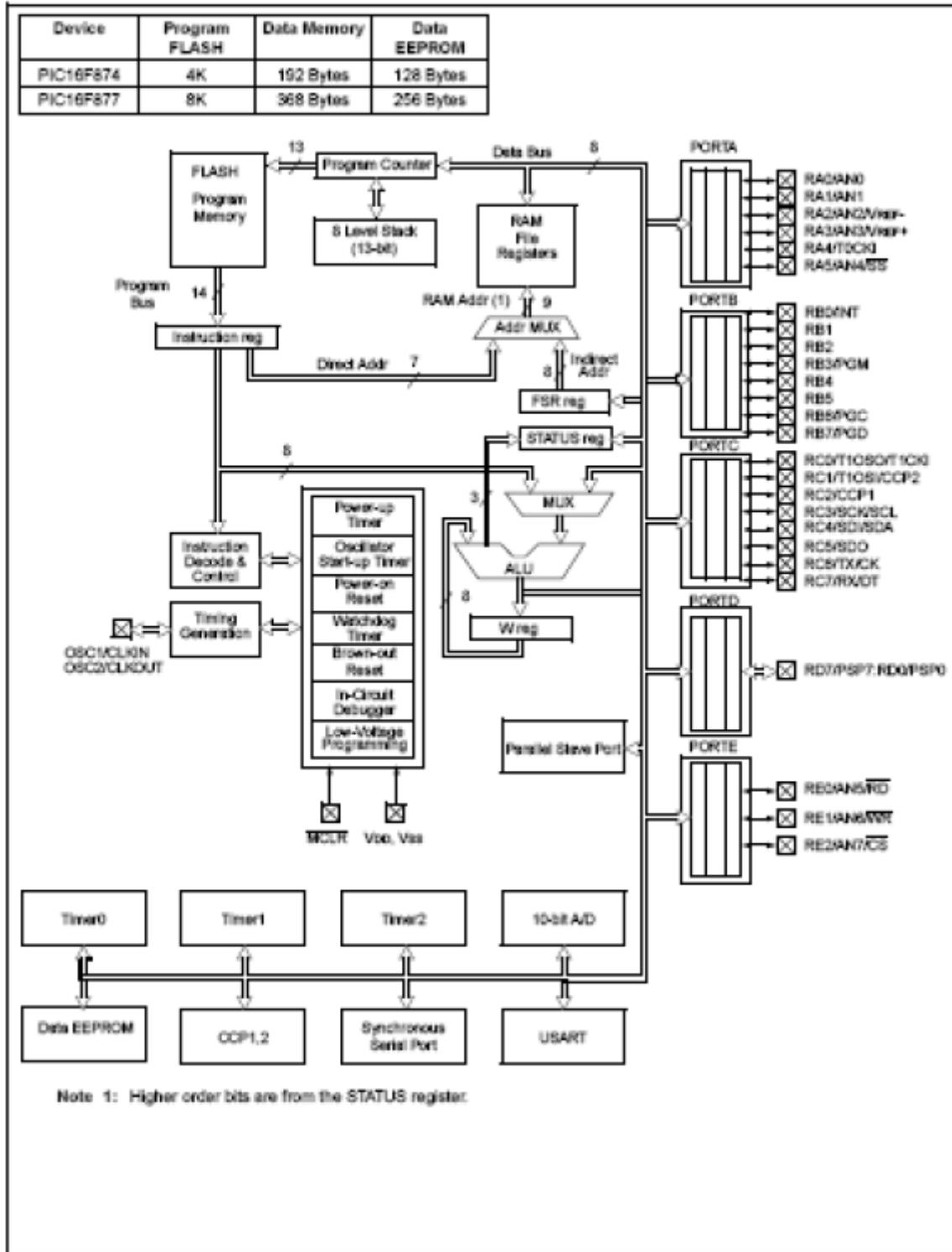
Analog = Analog input

O = Output

OD = Open-Drain (no P diode to VDD)

PIC16F87X

FIGURE 1-2: PIC16F874 AND PIC16F877 BLOCK DIAGRAM



PIC16F87X

TABLE 1-1: PIC16F873 AND PIC16F876 PINOUT DESCRIPTION

Pin Name	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	I	ST/CMOS ¹⁾	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP/THV	1	1	I/P	ST	Master clear (reset) input or programming voltage input or high voltage test mode control. This pin is an active low reset to the device.
RA0/AN0	2	2	I/O	TTL	PORTA is a bi-directional I/O port. RA0 can also be analog input0 RA1 can also be analog input1 RA2 can also be analog input2 or negative analog reference voltage RA3 can also be analog input3 or positive analog reference voltage RA4 can also be the clock input to the Timer0 module. Output is open drain type. RA5 can also be analog input4 or the slave select for the synchronous serial port.
RA1/AN1	3	3	I/O	TTL	
RA2/AN2/VREF-	4	4	I/O	TTL	
RA3/AN3/VREF+	5	5	I/O	TTL	
RA4/T0CKI	6	6	I/O	ST	
RA5/SS/AN4	7	7	I/O	TTL	
RB0/INT	21	21	I/O	TTL/ST ²⁾	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. RB0 can also be the external interrupt pin. RB3 can also be the low voltage programming input Interrupt on change pin. Interrupt on change pin. Interrupt on change pin or In-Circuit Debugger pin. Serial programming clock. Interrupt on change pin or In-Circuit Debugger pin. Serial programming data.
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3/PGM	24	24	I/O	TTL	
RB4	25	25	I/O	TTL	
RB5	26	26	I/O	TTL	
RB6/PGC	27	27	I/O	TTL/ST ²⁾	
RB7/PGD	28	28	I/O	TTL/ST ²⁾	
RC0/T1OSO/T1CKI	11	11	I/O	ST	PORTC is a bi-directional I/O port. RC0 can also be the Timer1 oscillator output or Timer1 clock input. RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output. RC2 can also be the Capture1 input/Compare1 output/PWM1 output. RC3 can also be the synchronous serial clock input/output for both SPI and I ² C modes. RC4 can also be the SPI Data In (SPI mode) or data I/O (I ² C mode). RC5 can also be the SPI Data Out (SPI mode). RC6 can also be the USART Asynchronous Transmit or Synchronous Clock. RC7 can also be the USART Asynchronous Receive or Synchronous Data.
RC1/T1OSI/CCP2	12	12	I/O	ST	
RC2/CCP1	13	13	I/O	ST	
RC3/SCK/SCL	14	14	I/O	ST	
RC4/SDI/SDA	15	15	I/O	ST	
RC5/SDO	16	16	I/O	ST	
RC6/TX/CK	17	17	I/O	ST	
RC7/RX/DT	18	18	I/O	ST	
VSS	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
VDD	20	20	P	—	Positive supply for logic and I/O pins.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

- Note: 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

PIC16F87X

TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	IO/P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	I	ST/CMOS ⁽¹⁾	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	14	15	31	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP/THV	1	2	18	IP	ST	Master clear (reset) input or programming voltage input or high voltage test mode control. This pin is an active low reset to the device.
RA0/AN0	2	3	19	IO	TTL	PORTA is a bi-directional IO port. RA0 can also be analog input ⁽²⁾ RA1 can also be analog input ⁽²⁾ RA2 can also be analog input ⁽²⁾ or negative analog reference voltage RA3 can also be analog input ⁽²⁾ or positive analog reference voltage RA4 can also be the clock input to the Timer0 binary counter. Output is open drain type. RA5 can also be analog input ⁽²⁾ or the slave select for the synchronous serial port.
RA1/AN1	3	4	20	IO	TTL	
RA2/AN2/VREF-	4	5	21	IO	TTL	
RA3/AN3/VREF+	5	6	22	IO	TTL	
RA4/T0CKI	6	7	23	IO	ST	
RA5/SS/AN4	7	8	24	IO	TTL	
RB0/INT	33	38	8	IO	TTL/ST ⁽¹⁾	PORTB is a bi-directional IO port. PORTB can be software programmed for internal weak pull-up on all inputs. RB0 can also be the external interrupt pin. RB3 can also be the low voltage programming input Interrupt on change pin. Interrupt on change pin. Interrupt on change pin or In-Circuit Debugger pin. Serial programming clock. Interrupt on change pin or In-Circuit Debugger pin. Serial programming data.
RB1	34	37	9	IO	TTL	
RB2	35	36	10	IO	TTL	
RB3/PGM	36	39	11	IO	TTL	
RB4	37	41	14	IO	TTL	
RB5	38	42	15	IO	TTL	
RB6/PGC	39	43	16	IO	TTL/ST ⁽²⁾	
RB7/PGD	40	44	17	IO	TTL/ST ⁽²⁾	
RC0/T1OSO/T1CKI	15	16	32	IO	ST	PORTC is a bi-directional IO port. RC0 can also be the Timer1 oscillator output or a Timer1 clock input. RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output. RC2 can also be the Capture1 input/Compare1 output/PWM1 output. RC3 can also be the synchronous serial clock input/output for both SPI and I ² C modes. RC4 can also be the SPI Data In (SPI mode) or data I/O (I ² C mode). RC5 can also be the SPI Data Out (SPI mode). RC6 can also be the USART Asynchronous Transmit or Synchronous Clock. RC7 can also be the USART Asynchronous Receive or Synchronous Data.
RC1/T1OSI/CCP2	16	18	35	IO	ST	
RC2/CCP1	17	19	36	IO	ST	
RC3/CK/SCL	18	20	37	IO	ST	
RC4/SDA	23	25	42	IO	ST	
RC5/SDO	24	26	43	IO	ST	
RC6/TX/CK	25	27	44	IO	ST	
RC7/RX/DT	26	29	1	IO	ST	

Legend: I = input O = output IO = input/output P = power
— = not used TTL = TTL input ST = Schmitt Trigger input

- Notes: 1. This buffer is a Schmitt Trigger input when configured as an external interrupt.
2. This buffer is a Schmitt Trigger input when used in serial programming mode.
3. This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4. This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

PIC16F87X

TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION (CONTINUED)

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	IO/P Type	Buffer Type	Description
RD0PSP0	19	21	38	IO	ST/TTL ⁽²⁾	PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus.
RD1PSP1	20	22	39	IO	ST/TTL ⁽²⁾	
RD2PSP2	21	23	40	IO	ST/TTL ⁽²⁾	
RD3PSP3	22	24	41	IO	ST/TTL ⁽²⁾	
RD4PSP4	27	30	2	IO	ST/TTL ⁽²⁾	
RD5PSP5	28	31	3	IO	ST/TTL ⁽²⁾	
RD6PSP6	29	32	4	IO	ST/TTL ⁽²⁾	
RD7PSP7	30	33	5	IO	ST/TTL ⁽²⁾	
RE0RDIAN5	8	9	25	IO	ST/TTL ⁽²⁾	PORTE is a bi-directional I/O port. RE0 can also be read control for the parallel slave port, or analog input5. RE1 can also be write control for the parallel slave port, or analog input6. RE2 can also be select control for the parallel slave port, or analog input7.
RE1WRIAN6	9	10	26	IO	ST/TTL ⁽²⁾	
RE2CSIAN7	10	11	27	IO	ST/TTL ⁽²⁾	
VSS	12,31	13,34	6,29	P	—	Ground reference for logic and I/O pins.
VDD	11,32	12,35	7,28	P	—	Positive supply for logic and I/O pins.
NC	—	1,17,28, 40	12,13, 33,34		—	These pins are not internally connected. These pins should be left unconnected.

Legend: I = input O = output IO = input/output P = power
 — = Not used TTL = TTL input ST = Schmitt Trigger input

- Note: 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
 2: This buffer is a Schmitt Trigger input when used in serial programming mode.
 3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
 4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

Triacs logic level

BT131 series

GENERAL DESCRIPTION

Glass passivated, sensitive gate triacs in a plastic envelope, intended for use in general purpose bidirectional switching and phase control applications. These devices are intended to be interfaced directly to microcontrollers, logic integrated circuits and other low power gate trigger circuits.

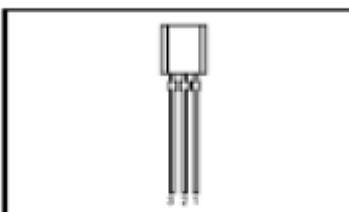
QUICK REFERENCE DATA

SYMBOL	PARAMETER	MAX.	MAX.	UNIT
V_{DRM}	Repetitive peak off-state voltages	500	600	V
$I_{T(RMS)}$	RMS on-state current	500	600	A
I_{TSM}	Non-repetitive peak on-state current	1	1	A
		16	16	A

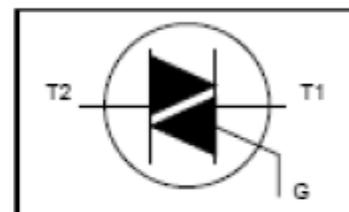
PINNING - TO92

PIN	DESCRIPTION
1	main terminal 2
2	gate
3	main terminal 1

PIN CONFIGURATION



SYMBOL



LIMITING VALUES

Limiting values in accordance with the Absolute Maximum System (IEC 134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.		UNIT
				-500 500 ¹	-600 600 ¹	
V_{DRM}	Repetitive peak off-state voltages		-	-500 500 ¹	-600 600 ¹	V
$I_{T(RMS)}$	RMS on-state current	full sine wave; $T_{jmax} \leq 51^\circ\text{C}$	-	1		A
I_{TSM}	Non-repetitive peak on-state current	full sine wave; $T_j = 25^\circ\text{C}$ prior to surge	-	16		A
		$t = 20\text{ ms}$	-	17.6		A
i^2t	i^2t for fusing	$t = 16.7\text{ ms}$	-	1.28		A ² s
di_T/dt	Repetitive rate of rise of on-state current after triggering	$t = 10\text{ ms}$	-			A ² s
		$I_{TM} = 1.5\text{ A}; I_G = 0.2\text{ A}; di_G/dt = 0.2\text{ A}/\mu\text{s}$				
		T2+ G+	-	50		A/ μs
		T2+ G-	-	50		A/ μs
		T2- G-	-	50		A/ μs
		T2- G+	-	10		A/ μs
I_{GM}	Peak gate current		-	2		A
V_{GM}	Peak gate voltage		-	5		V
P_{GM}	Peak gate power		-	5		W
$P_{GM(AV)}$	Average gate power	over any 20 ms period	-	0.5		W
T_{stg}	Storage temperature		-40	150		$^\circ\text{C}$
T_j	Operating junction temperature		-	125		$^\circ\text{C}$

¹ Although not recommended, off-state voltages up to 800V may be applied without damage, but the triac may switch to the on-state. The rate of rise of current should not exceed 3 A/ μs .



6-Pin DIP Zero-Cross Optoisolators Triac Driver Output (400 Volts Peak)

The MOC3041, MOC3042 and MOC3043 devices consist of gallium arsenide infrared emitting diodes optically coupled to a monolithic silicon detector performing the function of a Zero Voltage Crossing bilateral triac driver.

They are designed for use with a triac in the interface of logic systems to equipment powered from 115 Vac lines, such as solid-state relays, industrial controls, motors, solenoids and consumer appliances, etc.

- Simplifies Logic Control of 115 Vac Power
- Zero Voltage Crossing
- dV/dt of 2000 V/μs Typical, 1000 V/μs Guaranteed
- *To order devices that are tested and marked per VDE 0884 requirements, the suffix "V" must be included at end of part number. VDE 0884 is a test option.*

Recommended for 115/240 Vac(rms) Applications:

- Solenoid/Valve Controls
- Lighting Controls
- Static Power Switches
- AC Motor Drives
- Temperature Controls
- E.M. Contactors
- AC Motor Starters
- Solid State Relays

MAXIMUM RATINGS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Rating	Symbol	Value	Unit
--------	--------	-------	------

INFRARED EMITTING DIODE

Reverse Voltage	V_R	6	Volts
Forward Current — Continuous	I_F	60	mA
Total Power Dissipation @ $T_A = 25^\circ\text{C}$ Negligible Power in Output Driver Derate above 25°C	P_D	120	mW
		1.41	mW/°C

OUTPUT DRIVER

Off-State Output Terminal Voltage	V_{DRM}	400	Volts
Peak Repetitive Surge Current (PW = 100 μs, 120 pps)	I_{TSM}	1	A
Total Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	150	mW
		1.76	mW/°C

TOTAL DEVICE

Isolation Surge Voltage ⁽¹⁾ (Peak ac Voltage, 60 Hz, 1 Second Duration)	V_{ISO}	7500	Vac(pk)
Total Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	250	mW
		2.94	mW/°C
Junction Temperature Range	T_J	-40 to +100	°C
Ambient Operating Temperature Range ⁽²⁾	T_A	-40 to +85	°C
Storage Temperature Range ⁽²⁾	T_{stg}	-40 to +150	°C
Soldering Temperature (10 s)	T_L	260	°C

1. Isolation surge voltage, V_{ISO} , is an internal device dielectric breakdown rating.

For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.

2. Refer to Quality and Reliability Section in Opto Data Book for information on test conditions.

Preferred devices are Motorola recommended choices for future use and best overall value.

GlobalOptoisolator is a trademark of Motorola, Inc.

(Replaces MOC3040/D)

MOC3041
[IFT = 15 mA Max]
MOC3042
[IFT = 10 mA Max]
MOC3043*
[IFT = 5 mA Max]

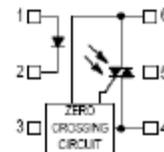
*Motorola Preferred Device

STYLE 6 PLASTIC



STANDARD THRU HOLE
CASE 730A-04

COUPLER SCHEMATIC



1. ANODE
2. CATHODE
3. NC
4. MAIN TERMINAL
5. SUBSTRATE
DO NOT CONNECT
6. MAIN TERMINAL



DM74LS48 BCD to 7-Segment Decoder

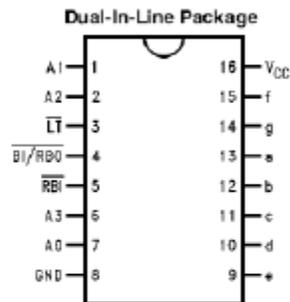
General Description

The 'LS48 translates four lines of BCD (8421) input data into the 7-segment numeral code and provides seven corresponding outputs having pull-up resistors, as opposed to totem pole pull-ups. These outputs can serve as logic signals, with a HIGH output corresponding to a lighted lamp segment, or can provide a 1.3 mA base current to npn lamp

driver transistors. Auxiliary inputs provide lamp test, blanking and cascadable zero-suppression functions.

The 'LS48 decodes the input data in the pattern indicated in the Truth Table and the segment identification illustration.

Connection Diagram



TL/F/10172-1

Order Number **DM74LS48M** or **DM74LS48N**
See NS Package Number **M16A** or **N16E**

Numerical Designations—Resultant Displays



TL/F/10172-4

Truth Table

Decimal Or Function	Inputs						Outputs							
	$\overline{\text{LT}}$	$\overline{\text{RBI}}$	A ₃	A ₂	A ₁	A ₀	$\overline{\text{BI/RBO}}$	a	b	c	d	e	f	g
0 (Note 1)	H	H	L	L	L	L	H	H	H	H	H	H	H	L
1 (Note 1)	H	X	L	L	L	H	H	L	H	H	L	L	L	L
2	H	X	L	L	H	L	H	H	H	L	H	L	L	H
3	H	X	L	L	H	H	H	H	H	H	H	L	L	H
4	H	X	L	H	L	L	H	L	H	H	L	L	H	H
5	H	X	L	H	L	H	H	H	L	H	H	L	H	H
6	H	X	L	H	H	L	H	L	L	H	H	H	H	H
7	H	X	L	H	H	H	H	H	H	H	L	L	L	L
8	H	X	H	L	L	L	H	H	H	H	H	H	H	H
9	H	X	H	L	L	H	H	H	H	H	L	L	H	H
10	H	X	H	L	H	L	H	L	L	L	H	H	L	H
11	H	X	H	L	H	H	H	L	L	H	H	L	L	H
12	H	X	H	H	L	L	H	L	H	L	L	L	H	H
13	H	X	H	H	L	H	H	H	L	L	H	L	H	H
14	H	X	H	H	H	L	H	L	L	L	H	H	H	H
15	H	X	H	H	H	H	H	L	L	L	L	L	L	L
$\overline{\text{BI}}$ (Note 2)	X	X	X	X	X	X	L	L	L	L	L	L	L	L
$\overline{\text{RBI}}$ (Note 3)	H	L	L	L	L	L	L	L	L	L	L	L	L	L
$\overline{\text{LT}}$ (Note 4)	L	X	X	X	X	X	H	H	H	H	H	H	H	H

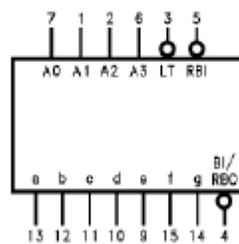
Note 1: $\overline{\text{BI/RBO}}$ is wired-AND logic serving as blanking input ($\overline{\text{BI}}$) and/or ripple-blanking output ($\overline{\text{RBO}}$). The blanking out ($\overline{\text{BI}}$) must be open or held at a HIGH level when output functions 0 through 15 are desired, and ripple-blanking input ($\overline{\text{RBI}}$) must be open or at a HIGH level if blanking of a decimal 0 is not desired. X = input may be HIGH or LOW.

Note 2: When a LOW level is applied to the blanking input (forced condition) all segment outputs go to a LOW level, regardless of the state of any other input condition.

Note 3: When ripple-blanking input ($\overline{\text{RBI}}$) and inputs A₀, A₁, A₂, and A₃ are at LOW level, with the lamp test input at HIGH level, all segment outputs go to a LOW level and the ripple-blanking output ($\overline{\text{RBO}}$) goes to a LOW level (response condition).

Note 4: When the blanking input/ripple-blanking output ($\overline{\text{BI/RBO}}$) is open or held at a HIGH level, and a LOW level is applied to lamp test input, all segment outputs go to a HIGH level.

Logic Symbol



TL/F/10172-2

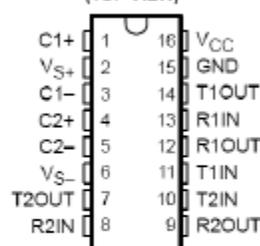
V_{CC} = Pin 15
GND = Pin 8

MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS0471 – FEBRUARY 1989 – REVISED OCTOBER 2002

- Meet or Exceed TIA/EIA-232-F and ITU Recommendation V.28
- Operate With Single 5-V Power Supply
- Operate Up to 120 kbit/s
- Two Drivers and Two Receivers
- ± 30 -V Input Levels
- Low Supply Current . . . 8 mA Typical
- Designed to be Interchangeable With Maxim MAX232
- ESD Protection Exceeds JESD 22 – 2000-V Human-Body Model (A114-A)
- Applications
 - TIA/EIA-232-F
 - Battery-Powered Systems
 - Terminals
 - Modems
 - Computers

MAX232 . . . D, DW, N, OR NS PACKAGE
MAX232I . . . D, DW, OR N PACKAGE
(TOP VIEW)



description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept ± 30 -V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

ORDERING INFORMATION

TA	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube	MAX232N	MAX232N
		Tube	MAX232D	MAX232
		Tape and reel	MAX232DR	
	SOIC (DW)	Tube	MAX232DW	MAX232
			Tape and reel	
	SOP (NS)	Tape and reel	MAX232NSR	MAX232
-40°C to 85°C	PDIP (N)	Tube	MAX232IN	MAX232IN
		Tube	MAX232ID	MAX232I
		Tape and reel	MAX232IDR	
	SOIC (DW)	Tube	MAX232IDW	MAX232I
			Tape and reel	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

LinASIC is a trademark of Texas Instruments.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

**TEXAS
INSTRUMENTS**

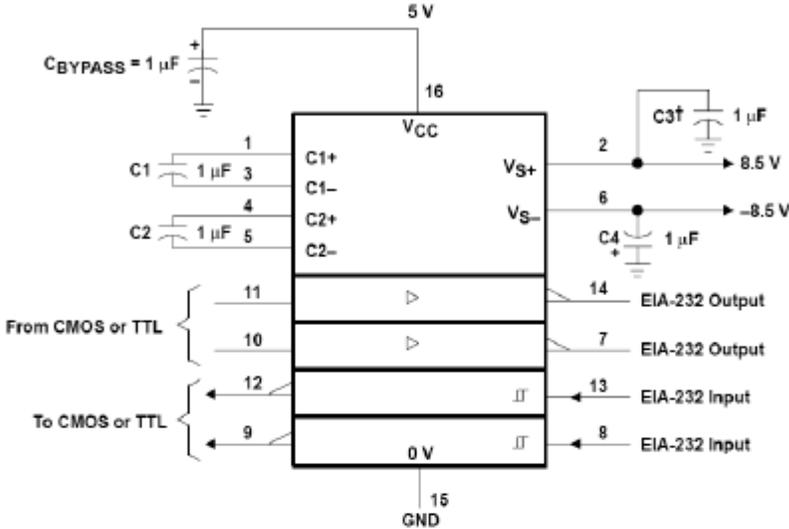
POST OFFICE BOX 655303 • DALLAS, TEXAS 75285

Copyright © 2002, Texas Instruments Incorporated

MAX232, MAX232I
 DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047I - FEBRUARY 1989 - REVISED OCTOBER 2002

APPLICATION INFORMATION



† C3 can be connected to V_{CC} or GND.

Figure 4. Typical Operating Circuit

KSP2222A

General Purpose Transistor

- Collector-Emitter Voltage: $V_{CE0} = 40V$
- Collector Power Dissipation: $P_C(\text{max}) = 625mW$
- Refer KSP2222 for graphs



NPN Epitaxial Silicon Transistor

Absolute Maximum Ratings $T_a = 25^\circ C$ unless otherwise noted

Symbol	Parameter	Value	Units
V_{CBO}	Collector-Base Voltage	75	V
V_{CEO}	Collector-Emitter Voltage	40	V
V_{EBO}	Emitter-Base Voltage	6	V
I_C	Collector Current	600	mA
P_C	Collector Power Dissipation	625	mW
T_J	Junction Temperature	150	$^\circ C$
T_{STG}	Storage Temperature	-55 ~ 150	$^\circ C$

Electrical Characteristics $T_a = 25^\circ C$ unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
BV_{CBO}	Collector-Base Breakdown Voltage	$I_C = 10\mu A, I_E = 0$	75			V
BV_{CEO}	Collector Emitter Breakdown Voltage	$I_C = 10mA, I_B = 0$	40			V
BV_{EBO}	Emitter-Base Breakdown Voltage	$I_E = 10\mu A, I_C = 0$	6			V
I_{CBO}	Collector Cut-off Current	$V_{CB} = 80V, I_E = 0$			0.01	μA
I_{EBO}	Emitter Cut-off Current	$V_{EB} = 3V, I_C = 0$			10	nA
h_{FE}	DC Current Gain	$I_C = 0.1mA, V_{CE} = 10V$ $V_{CE} = 10V, I_C = 1mA$ $V_{CE} = 10V, I_C = 10mA$ $V_{CE} = 10V, I_C = 150mA$ $V_{CE} = 10V, I_C = 500mA$	35 50 75 100 40		300	
$V_{CE(sat)}$	* Collector-Emitter Saturation Voltage	$I_C = 150mA, I_B = 15mA$ $I_C = 500mA, I_B = 50mA$			0.3 1	V
$V_{BE(sat)}$	* Base-Emitter Saturation Voltage	$I_C = 150mA, I_B = 15mA$ $I_C = 500mA, I_B = 50mA$		0.6	1.2 2	V
f_T	Current Gain Bandwidth Product	$V_{CE} = 20V, I_C = 20mA$ $f = 100MHz$	300			MHz
C_{ob}	Output Capacitance	$V_{CB} = 10V, I_E = 0, f = 1MHz$			8	pF
t_{ON}	Turn On Time	$V_{CC} = 30V, I_C = 150mA$ $I_{B1} = 15mA, V_{BE(off)} = 0.5V$			35	ns
t_{OFF}	Turn Off Time	$V_{CC} = 30V, I_C = 150mA$ $I_{B1} = I_{B2} = 15mA$			285	ns
NF	Noise Figure	$I_C = 100\mu A, V_{CE} = 10V$ $R_S = 1K\Omega, f = 1KHz$			4	dB

* Pulse Test: Pulse Widths: 300 μs , Duty Cycles: 2%
* Also available as and PN2222A

C. El primer automóvil “Made in Perú”

Existe una especie de resignación, por parte de la mayoría de peruanos, frente al marasmo en el que nos encontramos como país y, por eso, considero que es necesario tratar propuestas distintas a las que hasta ahora se han aplicado.

En el Perú no existe conciencia de la importancia del desarrollo endógeno en ciencia y tecnología. Como un ejemplo que caracteriza perfectamente a nuestro país, se presenta el siguiente extracto de la biografía de Juan Alberto Grieve, insigne inventor peruano de comienzos del siglo XX:

(..)Entusiasmado el ingeniero por esta nueva actividad, y siendo él la más grande autoridad en motores en el Perú, decidió, en 1907, construirse un automóvil. Grieve pasaba todos sus momentos libres en su taller, ubicado en el jirón Washington 117. Allí, con la ayuda de algunos operarios inició el desafío. A sus 30 años, joven para iniciar tamaña empresa, se convirtió en el pionero de la industria automovilística. Los coches que recorrían Lima a inicios del siglo, europeos en su mayoría, tenían entre **seis y ocho caballos de fuerza (HP)**, pues se pensaba que un carro de pequeña potencia era lo que la ciudad necesitaba.

Pero, la falta de caminos y carreteras en las afueras de la ciudad y las dificultades que eso producía a la circulación de autos, es lo que llevó a Grieve a diseñar un motor de **20HP**, para vencer la falta de buenos caminos. Así, su automóvil podría también ser un vehículo de turismo, que pudiera recorrer las periferias de Lima.



Fig. C1. Juan Alberto Grieve

Todos los componentes mecánicos del automóvil fueron diseñados en el taller, utilizando planos elaborados por el propio Grieve: motor, chasis, transmisión y diferencial. Todo esto en la **Lima de 1908**. Los únicos elementos que se importaron fueron las llantas Michellín (Francia), el encendido Bosch (Alemania), y el carburador.

El automóvil tenía cinco asientos para paseos fuera de la ciudad, dos adelante y tres atrás. Estos últimos podían retirarse, quedando un coche de dos asientos con un amplio espacio atrás.

A fines de 1908, se encontró listo el **primer automóvil diseñado y construido en América del Sur**, que en su época fue descrito como una "**joya de precisión mecánica**". Su performance fue comparada con la de un Renault o un Brassiere, considerado los mejores coches a principios del siglo XX.

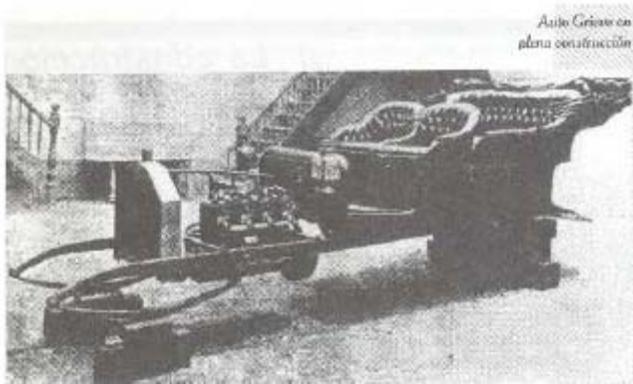


Fig. C2. Armazón del vehículo diseñado por Grieve

El costo total del automóvil fue de 300 libras, la mitad de lo que costaba un carro europeo y con igual potencia. "Grieve" fue el nombre de la patente que le dio el constructor y su idea era comercializar una flota de alrededor de veinte autos. Esta idea fue apoyada e incentivada desde el principio por su cercano amigo, Octavio Espinoza, inquieto periodista de la época y subdirector de la revista *CINEMA*, en cuyas páginas describió el automóvil de la siguiente manera:

"Los cuatro cilindros, conforme a las exigencias de la construcción moderna, están desaxiados (desplazamiento del eje) a 18 m/m. El motor da, con estas características, una potencia, al freno, de 20 h.p., con 1800 revoluciones por

minuto, pudiendo estas reducirse, merced a la maravillosa elasticidad del motor, a 200 revoluciones.

El señor Grieve no ha seguido la moda de las válvulas mandadas, y emplea en sus cilindros las automáticas. El encendido es doble, por magneto Sims Bosch, de alta tensión con una sola bobina; embrague cono de cuero; cambio de velocidades progresivo, tres velocidades adelante y marcha atrás, con un solo balador; ruedas iguales 815x105 m/m; neumáticos Michelin; radiador nido de abejas".

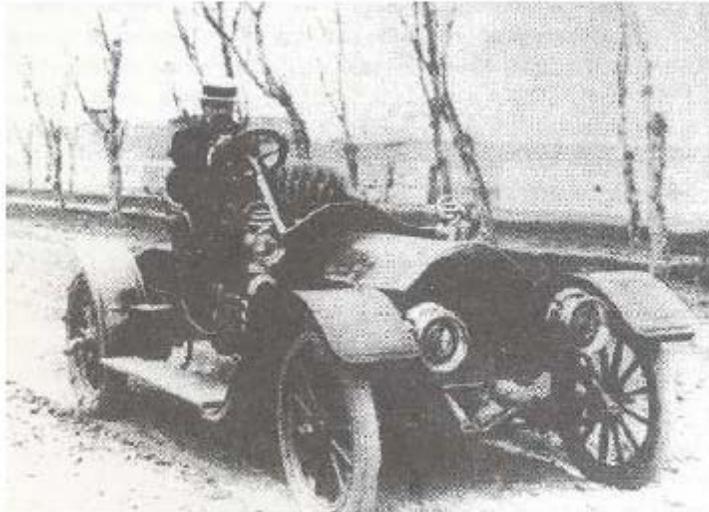


Fig. C3. Automóvil Grieve conducido por su creador

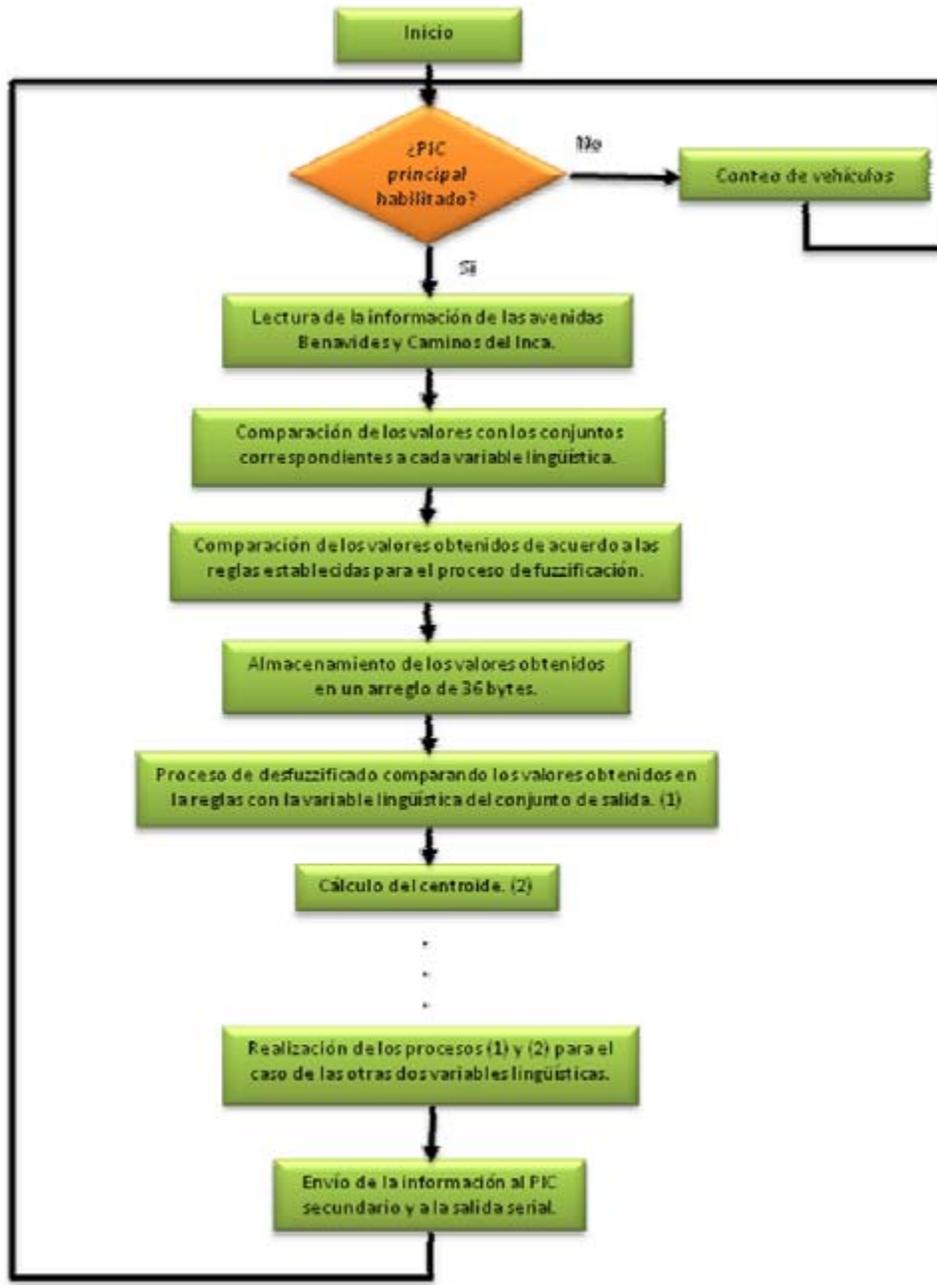
(..)Jorge Grieve Madge contó que su padre (Juan Alberto Grieve) se decidió hablar con el Presidente Leguía para, con auspicio del gobierno, construir tres vehículos para la Dirección de Correos y otros tres para la Prefectura. La respuesta que dio el Presidente fue: **"Nosotros necesitamos de los productos de países avanzados y no experiencias con productos peruanos"**.

Al no encontrar incentivos del gobierno, lo que pudo haberse convertido en una **florecente industria** no prosperó. A pesar que el **GRIEVE**, nombre con el cual patentó su vehículo, había demostrado ser un potente auto de ciudad que podía circular sin problemas en caminos difíciles, y a bajo costo, muchas personas prefirieron los precios de los **Ford** que empezaban a llegar. Los autos americanos no eran muy apreciados en ese momento, ya que su calidad era incomparable con la de un auto europeo.

Es difícil leer esta historia sin sentir una fuerte desazón. No me queda la menor duda que, con el emprendimiento "peruano" que nos caracteriza, si este proyecto del Sr. Grieve se hubiera materializado, el Perú tendría ahora una empresa automovilística multinacional, posiblemente no como los pesos pesados (General Motors, Grupo Volkswagen, Grupo FIAT, etc.) pero si una como Daewoo, KIA, Hyundai (que se desarrollaron en países emergentes, pero que ahora son "new industrial countries"). Así como le ocurrió al "adelantado a su tiempo" Sr. Grieve, otros peruanos, ilustres y desconocidos, se han topado con dos paredes infranqueables. Por un lado, un gobierno que no está interesado en "hacer experiencias con productos peruanos porque hacen las cosas supuestamente mejor"; y, por otro, unos consumidores que prefieren comprar productos extranjeros, incluso a pesar de que sean más costosos o de inferior calidad.

Katya Rodríguez Valencia. *Juan Alberto Grieve Becerra*. Universidad Nacional de Ingeniería. Proyecto Historia UNI. 89 pp.

D. Diagrama de flujo y códigos fuente del algoritmo de Lógica Difusa



En la figura D1 se puede apreciar el diagrama de flujo del microcontrolador principal en donde se realizan todas las etapas del proceso de fuzzificado de la información obtenida para así conseguir los valores de salida necesarios para que el sistema trabaje en óptimas condiciones. A continuación se presentan los códigos fuente de los microcontroladores utilizados.

Código fuente para el microcontrolador principal

```
include "modedefs.bas"
```

```
DEFINE      LCD_DREG PORTD  
DEFINE      LCD_DBIT 4  
DEFINE      LCD_RSREG PORTC  
DEFINE      LCD_RSBIT 1  
DEFINE      LCD_EREG PORTC  
DEFINE      LCD_EBIT 2  
DEFINE      LCD_BITS 4  
DEFINE      LCD_LINES 2
```

```
SENSORES   VAR BYTE
```

```
Ya         var word [4]  
Yb         var word [3]  
Yc         var word [3]
```

```
i          var byte  
j          var byte  
k          var byte
```

```
Z          var word [36]  
Z2         var word [36]  
Z3         var word [36]
```

```
Xa         var word [36]  
Xb         var word [36]  
Xc         var word [36]
```

```
A          var word  
B          var word  
C          var word
```

```
w          var byte  
l          var byte
```

```
V1         var word [21]  
V2         var word [21]  
V3         var word [21]  
V4         var word [21]  
V5         var word [21]  
V6         var word [21]  
V7         var word [21]  
V8         var word [26]  
V9         var word [26]  
V10        var word [26]
```

```
T0         var word  
T1         var word  
T2         var word  
T3         var word  
T4         var word  
T5         var word
```

Za var word [4]

G var byte [4]

P var byte

R var byte

salidaprincipal var word

salidasecundaria1 var word

salidasecundaria2 var word

DEFINE OSC 20

ADCON0=%00000000

ADCON1=%00000110

TRISB=%01111111

TRISD=%00001111

TRISC=%10010101

TRISA=%111111

TRISE=%011

PORTB =0

PORTD =0

PORTC =0

Salidaprincipal =0

salidasecundaria1 =0

salidasecundaria2 =0

A =0

B =0

C =0

P =10

PORTC.5 =0

goto principal

principal:

salidaprincipal =0

salidasecundaria1 =0

salidasecundaria2 =0

siguiente:

if PortC.0=1 then

if P=10 then

'Las funciones random son solo para la fase de demostración, no forman parte del programa final

random A

pause 1

random B

pause 1

random C

for i=0 to 35

```

next i
For i=0 to 3
  Ya(i) =0
Next i

For j=0 to 2
  Yb(j) =0
  Yc(j) =0
  Za(i) =0
Next j

```

'Procesamiento de la información recolectada para el proceso de la fuzzificación

'Avenida Principal

```

If A<=21 then
  Ya(0)=(100-((100*A)/21))
endif

If A>=8 and A<=35 then
  If A<=23 then
    Ya(1)=(100*(A-8))/15
  endif
  If A>23 then
    Ya(1)=(100-((100*(A-23))/12))
  endif
endif

If A>=25 and A<=51 then
  If A<=38 then
    Ya(2)=(100*(A-25))/13
  endif
  If A>38 then
    Ya(2)=(100-((100*(A-38))/13))
  endif
endif

If A>=39 then
  If A<=60 then
    Ya(3)=(100*(A-39))/21
  else
    Ya(3)=100
  endif
endif

```

'Transversal 1

```

If B<=16 then
  Yb(0)=(100-((25*B)/4))
endif

If B>=5 and B<=30 then
  If B<=16 then
    Yb(1)=((100*(B-5))/11)
  endif
  If B>16 then
    Yb(1)=(100-((50*(B-16))/7))
  endif
endif

```

```

if B>=18 then
    if B<=36 then
        Yb(2)=(50*(B-18))/9
    else
        Yb(2)=100
    endif
endif
endif

```

'Transversal 2

```

If C<=21 then
    Yc(0)=(100-((100*C)/21))
endif

If C>=5 and C<=40 then
    If C<=22 then
        Yc(1)=((100*(C-5))/17)
    endif
    If C>22 then
        Yc(1)=100-((50*(C-22))/9)
    endif
endif

if C>=24 then
    if C<=50 then
        Yc(2)=(50*(C-24))/13
    else
        Yc(2)=100
    endif
endif
endif

```

'Fuzificación

fuzzy:

```

For i=0 to 3
    For j=0 to 2
        For k=0 to 2
            if Ya(i)>Yc(k) and Yb(j)>Yc(k) then
                Za(k)=Yc(k)
            endif
            if Ya(i)<Yb(j) and Ya(i)<Yc(k) then
                Za(k)=Ya(i)
            endif
            if Yb(j)<Ya(i) and Yb(j)<Yc(k) then
                Za(k)=Yb(j)
            endif
            if Ya(i)==Yb(j) and Ya(i)==Yc(k) then
                Za(k)=Ya(i)
            endif
            if Ya(i)==Yb(j) and Ya(i)<Yc(k) then
                Za(k)=Ya(i)
            endif
            if Ya(i)==Yc(k) and Ya(i)<Yb(j) then
                Za(k)=Ya(i)
            endif
            if Yc(k)==Yb(j) and Yc(k)<Ya(i) then
                Za(k)=Yc(k)
            endif
            if j==0 & i==0 then
                Z(k)=Za(k)
            endif
        Next k
    Next j
Next i

```

```

endif
if j==1 & i==0 then
    Z(3+k)=Za(k)
endif
if j==2 & i==0 then
    Z(6+k)=Za(k)
endif
if j==0 & i==1 then
    Z(9+k)=Za(k)
endif
if j==1 & i==1 then
    Z(12+k)=Za(k)
endif
if j==2 & i==1 then
    Z(15+k)=Za(k)
endif
if j==0 & i==2 then
    Z(18+k)=Za(k)
endif
if j==1 & i==2 then
    Z(21+k)=Za(k)
endif
if j==2 & i==2 then
    Z(24+k)=Za(k)
endif
if j==0 & i==3 then
    Z(27+k)=Za(k)
endif
if j==1 & i==3 then
    Z(30+k)=Za(k)
endif
if j==2 & i==3 then
    Z(33+k)=Za(k)
endif

```

Next k

Next j

Next i

```

Z2(0)=Z(0)
Z2(1)=Z(1)
Z2(2)=Z(2)
Z2(3)=Z(9)
Z2(4)=Z(10)
Z2(5)=Z(11)
Z2(6)=Z(18)
Z2(7)=Z(19)
Z2(8)=Z(20)
Z2(9)=Z(27)
Z2(10)=Z(28)
Z2(11)=Z(29)
Z2(12)=Z(3)
Z2(13)=Z(4)
Z2(14)=Z(5)
Z2(15)=Z(12)
Z2(16)=Z(13)
Z2(17)=Z(14)
Z2(18)=Z(21)
Z2(19)=Z(22)
Z2(20)=Z(23)
Z2(21)=Z(30)

```

Z2(22)=Z(31)
Z2(23)=Z(32)
Z2(24)=Z(6)
Z2(25)=Z(7)
Z2(26)=Z(8)
Z2(27)=Z(15)
Z2(28)=Z(16)
Z2(29)=Z(17)
Z2(30)=Z(24)
Z2(31)=Z(25)
Z2(32)=Z(26)
Z2(33)=Z(33)
Z2(34)=Z(34)
Z2(35)=Z(35)

Z3(0)=Z(0)
Z3(1)=Z(3)
Z3(2)=Z(6)
Z3(3)=Z(9)
Z3(4)=Z(12)
Z3(5)=Z(15)
Z3(6)=Z(18)
Z3(7)=Z(21)
Z3(8)=Z(24)
Z3(9)=Z(27)
Z3(10)=Z(30)
Z3(11)=Z(33)
Z3(12)=Z(1)
Z3(13)=Z(4)
Z3(14)=Z(7)
Z3(15)=Z(10)
Z3(16)=Z(13)
Z3(17)=Z(16)
Z3(18)=Z(19)
Z3(19)=Z(22)
Z3(20)=Z(25)
Z3(21)=Z(28)
Z3(22)=Z(31)
Z3(23)=Z(34)
Z3(24)=Z(2)
Z3(25)=Z(5)
Z3(26)=Z(8)
Z3(27)=Z(11)
Z3(28)=Z(14)
Z3(29)=Z(17)
Z3(30)=Z(20)
Z3(31)=Z(23)
Z3(32)=Z(26)
Z3(33)=Z(29)
Z3(34)=Z(32)
Z3(35)=Z(35)

'Salida primaria (Avenida Benavides)

for i=0 to 20

 V1(i) = 0

 V2(i) = 0

 V3(i) = 0

 V4(i) = 0

next i

```

for i=0 to 35
  Xa(i) = 0
next i

for j=0 to 8
  if Z(j)>0 then
    k=0
    Xa(j) = (100-Z(j))*3/10

    for i = 0 to Xa(j)
      if i/4 == 0 then
        if Z(j)>V1(k) then
          V1(k)=Z(j)
        endif
        k=k+1
      endif
    next i

    for i = Xa(j)+1 to 30
      if i/4 == 0 then
        if Z(j)>V1(k) then
          V1(k)=(100-((10*i)/3))
        endif
        k=k+1
      endif
    next i

    for i = 31 to 80
      if i/4 == 0 then
        V1(k)=0
        k=k+1
      endif
    next i
  else
    k=0
    for i = 0 to 80
      if i/4==0 then
        if Z(j)>V1(k) then
          V1(k)=0
        endif
        k=k+1
      endif
    next i
  endif
next j

for j=9 to 17
  if Z(j)>0 then
    k=0
    Xa(j) = (((Z(j)*29)/100) + 11)

    for i = 0 to 10
      if i/4 == 0 then
        V2(k)=0
        k=k+1
      endif
    next i

    for i = 11 to Xa(j)
      if i/4 == 0 then

```

```

        if Z(j)>V2(k) then
            V2(k)=((i-11)*100)/29
        endif
        k=k+1
    endif
next i

for i = Xa(j)+1 to 40
    if i/4 == 0 then
        if Z(j)>V2(k) then
            V2(k)=Z(j)
        endif
        k=k+1
    endif
endif
next i

Xa(j)=(((100-Z(j))*3)/20) + 40

for i = 41 to Xa(j)
    if i/4 == 0 then
        if Z(j)>V2(k) then
            V2(k)=Z(j)
        endif
        k=k+1
    endif
endif
next i

for i = Xa(j)+1 to 55
    if i/4 == 0 then
        if Z(j)>V2(k) then
            V2(k)=100-(((i-40)*20)/3)
        endif
        k=k+1
    endif
endif
next i

for i = 56 to 80
    if i/4 == 0 then
        V2(k)=0
        k=k+1
    endif
endif
next i

else
    k=0
    for i = 0 to 80
        if i/4==0 then
            if Z(j)>V2(k) then
                V2(k)=0
            endif
            k=k+1
        endif
    endif
next i
endif
next j

for j = 18 to 26
    if Z(j)>0 then
        k=0
        Xa(j) = (Z(j)/5)+40
        for i = 0 to 40

```

```

        if i/4 == 0 then
            V3(k)=0
            k=k+1
        endif
    next i

    for i = 41 to Xa(j)
        if i/4 == 0 then
            if Z(j)>V3(k) then
                V3(k)=((i-40)*5)
            endif
            k=k+1
        endif
    next i

    for i = Xa(j)+1 to 60
        if i/4 == 0 then
            if Z(j)>V3(k) then
                V3(k)=Z(j)
            endif
            k=k+1
        endif
    next i

    Xa(j)=((100-Z(j))/10) + 60
    for i = 61 to Xa(j)
        if i/4 == 0 then
            if Z(j)>V3(k) then
                V3(k)=Z(j)
            endif
            k=k+1
        endif
    next i

    for i = Xa(j)+1 to 70
        if i/4 == 0 then
            if Z(j)>V3(k) then
                V3(k)=(100-(i-60)*10)
            endif
            k=k+1
        endif
    next i

    for i = 71 to 80
        if i/4 == 0 then
            V3(k)=0
            k=k+1
        endif
    next i

else
    k=0
    for i = 0 to 80
        if i/4==0 then
            if Z(j)>V3(k) then
                V3(k)=0
            endif
            k=k+1
        endif
    next i
endif

```

```

next j
for j=27 to 35
  if Z(j)>0 then
    k=0
    Xa(j) = ((Z(j)*3)/20)+65
    for i = 0 to 65
      if i/4 == 0 then
        V4(k)=0
        k=k+1
      endif
    next i

    for i = 66 to Xa(j)
      if i/4 == 0 then
        if Z(j)>V4(k) then
          V4(k)=((i-65)*20)/3
        endif
        k=k+1
      endif
    next i

    for i = Xa(j)+1 to 80
      if i/4 == 0 then
        if Z(j)>V4(k) then
          V4(k)=Z(j)
        endif
        k=k+1
      endif
    next i
  else
    k=0
    for i = 0 to 80
      if i/4==0 then
        if Z(j)>V4(k) then
          V4(k)=0
        endif
        k=k+1
      endif
    next i
  endif
next j

'totales
T0 =0
T1 =0
K =0

for i = 0 to 80
  if i/4==0 then
    if V1(k) >= V2(k) and V1(k) >= V3(k) and V1(k) >= V4(k) then
      T0=(V1(k)+T0)
      T1=((V1(k)*i)+T1)
    endif

    if V2(k) >= V1(k) and V2(k) >= V3(k) and V2(k) >= V4(k) then
      T0=(V2(k)+T0)
      T1=((V2(k)*i)+T1)
    endif
  endif
next i

```

```

endif

if V3(k) >= V1(k) and V3(k) >= V2(k) and V3(k) >= V4(k) then
    T0=(V3(k)+T0)
    T1=((V3(k)*i)+T1)
endif

if V4(k) >= V1(k) and V4(k) >= V2(k) and V4(k) >= V3(k) then
    T0=(V4(k)+T0)
    T1=((V4(k)*i)+T1)
endif

if V1(k) == V2(k) and V1(k) == V3(k) and V1(k) == V4(k) then
    T0=V1(k)+T0;
    T1=V1(k)*i+T1;
endif
k=k+1
endif
next i

'total final
salidaprincipal = T1/T0

```

'Salida secundaria 1 (Caminos del Inca hacia Bolichera)

```

for i=0 to 20
    V5(i) = 0
    V6(i) = 0
    V7(i) = 0
next i

for i=0 to 19
    Xb(i) = 0
next i

for j=0 to 11
    if Z2(j)>0 then
        k=0
        Xb(j) = (((100-Z2(j))*9)/50)
        for i = 0 to Xb(j)
            if i/2 == 0 then
                if Z2(j)>V5(k) then
                    V5(k)=Z2(j)
                endif
                k=k+1
            endif
        next i

        for i = Xb(j)+1 to 18
            if i/2 == 0 then
                if Z2(j)>V5(k) then
                    V5(k)= (100-(50*i)/9)
                endif
                k=k+1
            endif
        next i

        for i = 19 to 40
            if i/2 == 0 then
                V5(k)=0
            endif
        next i
    endif
next j

```

```

        k=k+1
    endif
next i
else
    k=0
    for i = 0 to 40
        if i/2==0 then
            if Z2(j)>V5(k) then
                V5(k)=0
            endif
            k=k+1
        endif
    next i
endif
next j
for j=12 to 23
    if Z2(j)>0 then
        k=0
        Xb(j) = ((Z2(j)*3)/20)+ 10
        for i = 0 to 10
            if i/2 == 0 then
                V6(k)=0
                k=k+1
            endif
        next i

        for i = 11 to Xb(j)
            if i/2 == 0 then
                if Z2(j)>V6(k) then
                    V6(k)=((i-10)*20)/3
                endif
                k=k+1
            endif
        next i

        for i = Xb(j)+1 to 25
            if i/2 == 0 then
                if Z2(j)>V6(k) then
                    V6(k)=Z2(j)
                endif
                k=k+1
            endif
        next i

        Xb(j)=((100-Z2(j))/10)+25
        for i = 26 to Xb(j)
            if i/2 == 0 then
                if Z2(j)>V6(k) then
                    V6(k)=Z2(j)
                endif
                k=k+1
            endif
        next i

        for i = Xb(j)+1 to 35
            if i/2 == 0 then
                if Z2(j)>V6(k) then
                    V6(k)=100-((i-25)*10)
                endif
            endif
        next i
    endif
next j

```

```

                                endif
                                k=k+1
                            endif
                        next i

                    for i = 36 to 40
                        if i/2 == 0 then
                            V6(k)=0
                            k=k+1
                        endif
                    next i
                else
                    k=0
                    for i = 0 to 40
                        if i/2==0 then
                            if Z2(j)>V6(k) then
                                V6(k)=0
                            endif
                            k=k+1
                        endif
                    next i
                endif
            next j

            for j=24 to 35
                if Z2(j)>0 then
                    k=0
                    Xb(j) = ((Z2(j)*13)/100)+27
                    for i = 0 to 27
                        if i/2 == 0 then
                            V7(k)=0
                            k=k+1
                        endif
                    next i

                    for i = 28 to Xb(j)
                        if i/2 == 0 then
                            if Z2(j)>V7(k) then
                                V7(k)=100-(((i-27)*100)/13)
                            endif
                            k=k+1
                        endif
                    next i

                    for i = Xb(j)+1 to 40
                        if i/2 == 0 then
                            if Z2(j)>V7(k) then
                                V7(k)=Z2(j)
                            endif
                            k=k+1
                        endif
                    next i
                else
                    k=0
                    for i = 0 to 40
                        if i/2==0 then
                            if Z2(j)>V7(k) then
                                V7(k)=0
                            endif
                        endif
                    next i
                endif
            next j
        endif
    next j
endfor

```

```

                                endif
                                k=k+1
                            endif
                        next i
                    endif
                next j

'totales
T2=0
T3=0
k=0
for i = 0 to 40
    if i/2==0 then
        if V5(k) >= V6(k) and V5(k) >= V7(k) then
            T2=(V5(k)+T2)
            T3=((V5(k)*i)+T3)
        endif

        if V6(k) >= V5(k) and V6(k) >= V7(k) then
            T2=(V6(k)+T2)
            T3=((V6(k)*i)+T3)
        endif

        if V7(k) >= V5(k) and V7(k) >= V6(k) then
            T2=(V7(k)+T2)
            T3=((V7(k)*i)+T3)
        endif

        if V5(k) == V6(k) and V5(k) == V7(k) then
            T2=(V5(k)+T2)
            T3=((V5(k)*i)+T3)
        endif
        k=k+1
    endif
next i

'total final
salidasecundaria1 = T3/T2

'Salida secundaria 2 (Caminos del Inca hacia San Borja)
for i=0 to 25
    V10(i) = 0
    V8(i) = 0
    V9(i) = 0
next i

for i=0 to 35
    Xc(i) = 0
next i

for j=0 to 11
    if Z3(j)>0 then
        k=0
        Xc(j) = (((100-Z3(j))*9)/50)
        for i = 0 to Xc(j)
            if i/2 == 0 then
                if Z3(j)>V8(k) then
                    V8(k)=Z3(j)
                endif
                k=k+1
            endif
        next i
    endif
next j

```

```

        endif
    next i

    for i = Xc(j)+1 to 18
        if i/2 == 0 then
            if Z3(j)>V8(k) then
                V8(k)= 100-((50*i)/9)
            endif
            k=k+1
        endif
    next i

    for i = 19 to 50
        if i/2 == 0 then
            V8(k)=0
            k=k+1
        endif
    next i

else
    k=0
    for i = 0 to 50
        if i/2==0 then
            if Z3(j)>V8(k) then
                V8(k)=0
            endif
            k=k+1
        endif
    next i
endif
next j

for j=12 to 23
    if Z3(j)>0 then
        k=0
        Xc(j) = ((Z3(j)*9)/50)+10
        for i = 0 to 10
            if i/2 == 0 then
                V9(k)=0
                k=k+1
            endif
        next i

        for i = 11 to Xc(j)
            if i/2 == 0 then
                if Z3(j)>V9(k) then
                    V9(k)=((i-10)*50)/9
                endif
                k=k+1
            endif
        next i

        for i = Xc(j)+1 to 28
            if i/2 == 0 then
                if Z3(j)>V9(k) then
                    V9(k)=Z3(j)
                endif
                k=k+1
            endif
        next i
    endif
endfor

```

```

Xc(j)=((100-Z3(j))*3/25)+28
for i = 29 to Xc(j)
    if i/2 == 0 then
        if Z3(j)>V9(k) then
            V9(k)=Z3(j)
        endif
        k=k+1
    endif
next i

for i = Xc(j)+1 to 40
    if i/2 == 0 then
        if Z3(j)>V9(k) then
            V9(k)=100-(((i-28)*25)/3)
        endif
        k=k+1
    endif
next i

for i = 41 to 50
    if i/2 == 0 then
        V9(k)=0
        k=k+1
    endif
next i

else
    k=0
    for i = 0 to 50
        if i/2==0 then
            if Z3(j)>V9(k) then
                V9(k)=0
            endif
            k=k+1
        endif
    next i
endif

next j

for j=24 to 35
    if Z3(j)>0 then
        k=0
        Xc(j) = (((Z3(j))*3)/20 )+35
        for i = 0 to 35
            if i/2 == 0 then
                V10(k)=0
                k=k+1
            endif
        next i

        for i = 36 to Xc(j)
            if i/2 == 0 then
                if Z3(j)>V10(k) then
                    V10(k)=((20*(i-35))/3)
                endif
                k=k+1
            endif
        next i
    endif
next j

```

```

        for i = Xc(j)+1 to 50
            if i/2 == 0 then
                if Z3(j)>V10(k) then
                    V10(k)=Z3(j)
                endif
                k=k+1
            endif
        next i
    else
        k=0
        for i = 0 to 50
            if i/2==0 then
                if Z3(j)>V10(k) then
                    V10(k)=0
                endif
                k=k+1
            endif
        next i
    endif
next j

'totales
T4=0
T5=0
k=0
for i = 0 to 50
    if i/2==0 then
        if V8(k) >= V9(k) and V8(k) >= V10(k) then
            T4=(V8(k)+T4)
            T5=((V8(k)*i)+T5)
        endif

        if V9(k) >= V8(k) and V9(k) >= V10(k) then
            T4=(V9(k)+T4)
            T5=((V9(k)*i)+T5)
        endif

        if V10(k) >= V8(k) and V10(k) >= V9(k) then
            T4=(V10(k)+T4)
            T5=((V10(k)*i)+T5)
        endif

        if V8(k) == V9(k) and V8(k) == V10(k) then
            T4=(V8(k)+T4)
            T5=((V8(k)*i)+T5)
        endif
        k=k+1
    endif
next i

'total final
salidasecundaria2 = T5/T4

LCDout $FE,1
pause 10
LCDout "Benavides ", #salidaprincipal
LCDout $FE,$C0
pause 10
LCDout "C11 ", #salidasecundaria1, " C12 ", #salidasecundaria2

```

```

        pause 10
        serout2 PORTC.6, 396,
        ["A",#salidaprincipal,"B",#salidasecundaria1,"C",#salidasecundaria2,"D",#A,"E",
        #B,"F",#C,"G", 13,10]
        pause 80

        G[0] = salidaprincipal
        G[1] = salidasecundaria1
        G[2] = salidasecundaria2
        G[3] = 10
        P=0
    endif
    PortC.5=1
endif

```

```

SENSORES.0 =PORTA.0
SENSORES.1 =PORTA.1
SENSORES.2 =PORTA.2
SENSORES.3 =PORTA.3
SENSORES.4 =PORTA.4
SENSORES.5 =PORTA.5
SENSORES.6 =PORTE.0
SENSORES.7 =PORTE.1

```

```

SELECT CASE SENSORES

```

```

CASE 89
    PAUSE 4
    A=A-2
    B=B-1
    C=C-1

```

```

CASE 82,94
    PAUSE 4
    A=A-2
    B=B
    C=C-1

```

```

CASE 86
    PAUSE 4
    A=A-2
    B=B+1
    C=C-1

```

```

CASE 88, 91
    PAUSE 4
    A=A-2
    B=B-1
    C=C

```

```

CASE 80, 83, 92
    PAUSE 4
    A=A-2
    B=B
    C=C

```

```

CASE 84, 87
    PAUSE 4
    A=A-2
    B=B+1

```

C=C

CASE 89

PAUSE 4
A=A-2
B=B-1
C=C+1

CASE 81, 83

PAUSE 4
A=A-2
B=B
C=C+1

CASE 85

PAUSE 4
A=A-2
B=B+1
C=C+1

CASE 26, 74, 122, 218

PAUSE 4
A=A-1
B=B-1
C=C-1

CASE 18,30,66,78,114,126,210,222

PAUSE 4
A=A-1
B=B
C=C-1

CASE 22,70,118,214

PAUSE 4
A=A-1
B=B+1
C=C-1

CASE 24,27,72,75,120,123,216,219

PAUSE 4
A=A-1
B=B-1
C=C

CASE 16,19,28,64,67,76,79,95,112,115,124,127,208,211,220

PAUSE 4
A=A-1
B=B
C=C

CASE 20,23,68,71,116,119,212,215

PAUSE 4
A=A-1
B=B+1
C=C

CASE 25,73,121,217

PAUSE 4
A=A-1
B=B-1

C=C+1

CASE 17,29,65,77,113,125,209,221

PAUSE 4

A=A-1

B=B

C=C+1

CASE 21,69,117,213

PAUSE 4

A=A-1

B=B+1

C=C+1

CASE 10,58,106,154,202,250

PAUSE 4

A=A

B=B-1

C=C-1

CASE 2,14,50,62,98,110,146,158,194,206,242,254

PAUSE 4

A=A

B=B

C=C-1

CASE 6,54,102,150,198,246

PAUSE 4

A=A

B=B+1

C=C-1

CASE 8,11,56,59,104,107,152,155,200,203,248,251

PAUSE 4

A=A

B=B-1

C=C

CASE 0,3,12,15,31,48,51,60,63,96,99,108,111,144,147,156,192,195,204,207,223,240,
243,252,255

PAUSE 4

A=A

B=B

C=C

CASE 4,7,52,55,100,103,148,151,196,199,244,247

PAUSE 4

A=A

B=B+1

C=C

CASE 9,57,105,153,201,249

PAUSE 4

A=A

B=B-1

C=C+1

CASE 1,13,49,61,97,109,145,157,193,205,241,253

PAUSE 4

A=A

B=B
C=C+1

CASE 5,53,101,149,197,245
PAUSE 4
A=A
B=B+1
C=C+1

CASE 42,138,186,234
PAUSE 4
A=A+1
B=B-1
C=C-1

CASE 34,46,130,142,178,190,226,238
PAUSE 4
A=A+1
B=B
C=C-1

CASE 38,134,182,230
PAUSE 4
A=A+1
B=B+1
C=C-1

CASE 40,43,136,139,184,187,232,235
PAUSE 4
A=A+1
B=B-1
C=C

CASE 32,35,44,47,128,131,140,143,159,176,179,188,191,224,227,236,239
PAUSE 4
A=A+1
B=B
C=C

CASE 36,39,132,135,180,183,228,231
PAUSE 4
A=A+1
B=B+1
C=C

CASE 41,137,185,233
PAUSE 4
A=A+1
B=B-1
C=C+1

CASE 33,45,129,141,177,189,225,237
PAUSE 4
A=A+1
B=B
C=C+1

CASE 37,133,181,229
PAUSE 4
A=A+1

B=B+1
C=C+1

CASE 170
PAUSE 4
A=A+2
B=B-1
C=C-1

CASE 162,174
PAUSE 4
A=A+2
B=B
C=C-1

CASE 166
PAUSE 4
A=A+2
B=B+1
C=C-1

CASE 168,171
PAUSE 4
A=A+2
B=B-1
C=C

CASE 160,163,172,175
PAUSE 4
A=A+2
B=B
C=C

CASE 164,167
PAUSE 4
A=A+2
B=B+1
C=C

CASE 169
PAUSE 4
A=A+2
B=B-1
C=C+1

CASE 161,173
PAUSE 4
A=A+2
B=B
C=C+1

CASE 165
PAUSE 4
A=A+2
B=B+1
C=C+1

CASE ELSE
PAUSE 4

```

        A=A
        B=B
        C=C

END SELECT

IF PORTC.5= 1 THEN
    serout PORTC.3,N2400, ["A", G[0],G[1],G[2],G[3]]

    if portC.0=0 then
        PORTC.5=0
        P=10
    endif
ENDIF

```

branchl 0, [principal]

end

Código fuente para el microcontrolador secundario

```
include "modedefs.bas"
```

```

DEFINE OSC 20
ADCON1=%00000111
TRISA=%000000
TRISE=%000
TRISC=%01011000
TRISB=%00000000
TRISD=%00000000

```

```

a    VAR  BYTE[4]
Z    VAR  BYTE
B    VAR  BYTE
F    VAR  WORD
AM1  VAR  WORD
AM2  VAR  WORD
VE1  VAR  WORD
VE2  VAR  WORD
RO1  VAR  WORD
RO2  VAR  WORD

```

```

regbenav  VAR WORD
regbolich VAR WORD
regsbj    VAR WORD
unidades  VAR BYTE
decenas   VAR BYTE
R         VAR BYTE

```

```

PORTC=0
Z=2
AM1=0
AM2=0
VE1=0
VE2=0
RO1=0
RO2=0

```

```

PORTD=0
PORTB=0
PORTA=1
PortC.7=1
regbenav=0
regbolich=0
regsbj=0
unidades=0
decenas=0
for F=0 to 2
a[F]=0
next F
a[3]=10

```

```
branchl 0, [mainloop]
```

```
mainloop:      ' Lazo principal
```

```
if PORTC.6=1 then 'El PIC Principal Terminó de procesar la información y habilita la lectura del
PIC Secundario
```

```

serin PORTC.3,N2400,["A"], a[0],a[1],a[2],a[3]      ' Lee los datos del PIC Principal
portc.7=0
GOSUB ROJOBENBOL
GOSUB AMARILLOSBJ
GOSUB VERDEBOL
GOSUB AMARILLOBOL
GOSUB VERDEBEN
GOSUB AMARILLOBEN
B=10
RO1=0 : AM1=0 : VE1=0 : RO2=0 : AM2=0 : VE2=0

```

```
endif
```

```

if B=10 then
porta=%111111
porte.0=1
porte.1=1
PortC.7=1
B=0

```

```
endif
```

```
branchl 0, [mainloop]
```

```
ROJOBENBOL:
```

```
PORTB=%10010010 : PORTD.7=0 : PORTD.6=0 : PORTD.5=0 : PORTD.4=1
```

```
F=a[2]*50
```

```
for RO1=0 to F
```

```
if RO1=0 then
```

```
regbenav      = a[2]+a[1]+4-RO1
```

```
regbolich     = a[2]+2-RO1
```

```
regsbj        = a[2]-RO1
```

```
endif
```

```
if RO1//50=0 then
```

```
regbenav      = (a[2]*50+a[1]*50+4*50-RO1)/50
```

```
regbolich     = (a[2]*50+2*50-RO1)/50
```

```
regsbj        = (a[2]*50-RO1)/50
```

```
endif
```

```
if regsbj>=10 then
    unidades=regsbj//10 : decenas=regsbj/10
else
    unidades=regsbj//10 : decenas=0
endif
```

'San Borja

```
portd.0=unidades.0
portd.1=unidades.1
portd.2=unidades.2
portd.3=unidades.3
pause 1
porta=%000000 : porte.0=0 : porte.1=1
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
portd.0=decenas.0
portd.1=decenas.1
portd.2=decenas.2
portd.3=decenas.3
pause 1
porta=%000000 : porte.0=1 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
if regbenav>=10 then
    unidades=regbenav//10 : decenas=regbenav/10
else
    unidades=regbenav//10 : decenas=0
endif
```

'Benavides

```
portd.0=unidades.0
portd.1=unidades.1
portd.2=unidades.2
portd.3=unidades.3
pause 1
porta=%001000 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
porta=%000010 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
portd.0=decenas.0
portd.1=decenas.1
portd.2=decenas.2
portd.3=decenas.3
pause 1
porta=%000100 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
```

```

pause 1

porta=%000001 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1

if regbolich>=10 then
    unidades=regbolich//10 : decenas=regbolich/10
else
    unidades=regbolich//10 : decenas=0
endif

```

'Bolochera

```

portd.0=unidades.0
portd.1=unidades.1
portd.2=unidades.2
portd.3=unidades.3
pause 1
porta=%100000 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1

portd.0=decenas.0
portd.1=decenas.1
portd.2=decenas.2
portd.3=decenas.3
pause 1
porta=%010000 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1

```

next RO1

RETURN

AMARILLOSBJ:

```

PORTB=%10010010
PORTD.7=0 : PORTD.6=0 : PORTD.5=1 : PORTD.4=0

```

```

for AM1=0 to 100
    if AM1=0 then
        regbenav = a[1]+4-AM1
        regbolich = 2-AM1
        regsbj = Z-AM1
    endif

    if AM1//50=0 then
        regbenav = (a[1]*50+4*50-AM1)/50
        regbolich = (100-AM1)/50
        regsbj = (Z*50-AM1*50)/50
    endif

    if regsbj>=10 then

```

```
        unidades=regsbj//10 : decenas=regsbj/10
    else
        unidades=regsbj//10 : decenas=0
    endif
```

'San Borja

```
portd.0=unidades.0
portd.1=unidades.1
portd.2=unidades.2
portd.3=unidades.3
pause 1
porta=%000000 : porte.0=0 : porte.1=1
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
portd.0=decenas.0
portd.1=decenas.1
portd.2=decenas.2
portd.3=decenas.3
pause 1
porta=%000000 : porte.0=1 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
if regbenav>=10 then
    unidades=regbenav//10 : decenas=regbenav/10
else
    unidades=regbenav//10 : decenas=0
endif
```

'Benavides

```
portd.0=unidades.0
portd.1=unidades.1
portd.2=unidades.2
portd.3=unidades.3
pause 1
porta=%001000 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
porta=%000010 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
portd.0=decenas.0
portd.1=decenas.1
portd.2=decenas.2
portd.3=decenas.3
pause 1
porta=%000100 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```

porta=%000001 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1

if regbolich>=10 then
    unidades=regbolich//10 : decenas=regbolich/10
else
    unidades=regbolich//10 : decenas=0
endif

```

'Bolochera

```

portd.0=unidades.0
portd.1=unidades.1
portd.2=unidades.2
portd.3=unidades.3
pause 1
porta=%100000 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1

portd.0=decenas.0
portd.1=decenas.1
portd.2=decenas.2
portd.3=decenas.3
pause 1
porta=%010000 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1

```

next AM1

RETURN

VERDEBOL:

PORTB=%10010000 : PORTD.7=1 : PORTD.6=1 : PORTD.5=0 : PORTD.4=0

```

F=50*a[1]
for VE1=0 to F
    if VE1 = 0 then
        regbenav = a[1]+2-VE1
        regbolich = a[1]-VE1
        regsbj = a[0]+a[1]+4-VE1
    endif

    if VE1//50 = 0 then
        regbenav = (a[1]*50+2*50-VE1)/50
        regbolich = (a[1]*50-VE1)/50
        regsbj = (a[0]*50+a[1]*50+4*50-VE1)/50
    endif

    if regsbj>=10 then
        unidades=regsbj//10 : decenas=regsbj/10
    else
        unidades=regsbj//10 : decenas=0
    endif
endif

```

'San Borja

```
portd.0=unidades.0
portd.1=unidades.1
portd.2=unidades.2
portd.3=unidades.3
pause 1
porta=%000000 : porte.0=0 : porte.1=1
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1

portd.0=decenas.0
portd.1=decenas.1
portd.2=decenas.2
portd.3=decenas.3
pause 1
porta=%000000 : porte.0=1 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1

if regbenav>=10 then
    unidades=regbenav//10 : decenas=regbenav/10
else
    unidades=regbenav//10 : decenas=0
endif
```

'Benavides

```
portd.0=unidades.0
portd.1=unidades.1
portd.2=unidades.2
portd.3=unidades.3
pause 1
porta=%001000 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1

porta=%000010 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1

portd.0=decenas.0
portd.1=decenas.1
portd.2=decenas.2
portd.3=decenas.3
pause 1
porta=%000100 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1

porta=%000001 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```

if regbolich>=10 then
    unidades=regbolich//10 : decenas=regbolich/10
else
    unidades=regbolich//10 : decenas=0
endif

'Bolochera

portd.0=unidades.0
portd.1=unidades.1
portd.2=unidades.2
portd.3=unidades.3
pause 1
porta=%100000 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1

portd.0=decenas.0
portd.1=decenas.1
portd.2=decenas.2
portd.3=decenas.3
pause 1
porta=%010000 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1

next VE1

RETURN

AMARILLOBOL:

PORTB=%10010001 : PORTD.7=0 : PORTD.6=1 : PORTD.5=0 : PORTD.4=0

for RO2=0 to 100
    if RO2=0 then
        regbenav = 2-RO2
        regbolich = Z-RO2
        regsbj = a[0]+4-RO2
    endif

    if RO2//50 = 0 then
        regbenav = (2*50-RO2)/50
        regbolich = (Z*50-RO2)/50
        regsbj = (a[0]*50+4*50-RO2)/50
    endif

    if regsbj>=10 then
        unidades=regsbj//10 : decenas=regsbj/10
    else
        unidades=regsbj//10 : decenas=0
    endif

'San Borja

portd.0=unidades.0

```

```
portd.1=unidades.1
portd.2=unidades.2
portd.3=unidades.3
pause 1
porta=%000000 : porte.0=0 : porte.1=1
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
portd.0=decenas.0
portd.1=decenas.1
portd.2=decenas.2
portd.3=decenas.3
pause 1
porta=%000000 : porte.0=1 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
if regbenav>=10 then
    unidades=regbenav//10 : decenas=regbenav/10
else
    unidades=regbenav//10 : decenas=0
endif
```

'Benavides

```
portd.0=unidades.0
portd.1=unidades.1
portd.2=unidades.2
portd.3=unidades.3
pause 1
porta=%001000 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
porta=%000010 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
portd.0=decenas.0
portd.1=decenas.1
portd.2=decenas.2
portd.3=decenas.3
pause 1
porta=%000100 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
porta=%000001 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

'Bolochera

```
if regbolich>=10 then
```

```

        unidades=regbolich//10 : decenas=regbolich/10
    else
        unidades=regbolich//10 : decenas=0
    endif

    portd.0=unidades.0
    portd.1=unidades.1
    portd.2=unidades.2
    portd.3=unidades.3
    pause 1
    porta=%100000 : porte.0=0 : porte.1=0
    pause 1
    porta=%000000 : porte.0=0 : porte.1=0
    pause 1

    portd.0=decenas.0
    portd.1=decenas.1
    portd.2=decenas.2
    portd.3=decenas.3
    pause 1
    porta=%010000 : porte.0=0 : porte.1=0
    pause 1
    porta=%000000 : porte.0=0 : porte.1=0
    pause 1

next RO2

RETURN

VERDEBEN:

PORTB=%00100110 : PORTD.7=0 : PORTD.6=1 : PORTD.5=0 : PORTD.4=0
F=50*a[0]

for AM2=0 to F
    if AM2 = 0 then
        regbenav = a[0]-AM2
        regbolich = a[0]+a[2]+4-AM2
        regsbj = a[0]+2-AM2
    endif

    if AM2//50 = 0 then
        regbenav = (a[0]*50-AM2)/50
        regbolich = (a[0]*50+a[2]*50+4*50-AM2)/50
        regsbj = (a[0]*50+2*50-AM2)/50
    endif

    if regsbj>=10 then
        unidades=regsbj//10 : decenas=regsbj/10
    else
        unidades=regsbj//10 : decenas=0
    endif

'San Borja

    portd.0=unidades.0
    portd.1=unidades.1
    portd.2=unidades.2
    portd.3=unidades.3
    pause 1

```

```
porta=%000000 : porte.0=0 : porte.1=1
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
portd.0=decenas.0
portd.1=decenas.1
portd.2=decenas.2
portd.3=decenas.3
pause 1
porta=%000000 : porte.0=1 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
if regbenav>=10 then
    unidades=regbenav//10 : decenas=regbenav/10
else
    unidades=regbenav//10 : decenas=0
endif
```

'Benavides

```
portd.0=unidades.0
portd.1=unidades.1
portd.2=unidades.2
portd.3=unidades.3
pause 1
porta=%001000 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
porta=%000010 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
portd.0=decenas.0
portd.1=decenas.1
portd.2=decenas.2
portd.3=decenas.3
pause 1
porta=%000100 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
porta=%000001 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

'Bolochera

```
if regbolich>=10 then
    unidades=regbolich//10 : decenas=regbolich/10
else
    unidades=regbolich//10 : decenas=0
endif
```

```
portd.0=0
portd.1=1
portd.2=0
portd.3=1
pause 1
porta=%100000 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
portd.0=0
portd.1=1
portd.2=0
portd.3=1
pause 1
porta=%010000 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

next AM2

RETURN

AMARILLOBEN:

```
PORTB=%01001010 : PORTD.7=0 : PORTD.6=1 : PORTD.5=0 : PORTD.4=0
```

for VE2=0 to 100

```
if VE2 = 0 then
    regbenav = 2-VE2
    regbolich = a[2]+4-VE2
    regsbj = 2-VE2
endif
```

```
if VE2//50 = 0 then
    regbenav = (2*50-VE2)/50
    regbolich = (a[2]*50+4*50-VE2)/50
    regsbj = (2*50-VE2)/50
endif
```

```
if regsbj>=10 then
    unidades=regsbj//10 : decenas=regsbj/10
else
    unidades=regsbj//10 : decenas=0
endif
```

'San Borja

```
portd.0=unidades.0
portd.1=unidades.1
portd.2=unidades.2
portd.3=unidades.3
pause 1
porta=%000000 : porte.0=0 : porte.1=1
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```

portd.0=decenas.0
portd.1=decenas.1
portd.2=decenas.2
portd.3=decenas.3
pause 1
porta=%000000 : porte.0=1 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1

if regbenav>=10 then
    unidades=regbenav//10 : decenas=regbenav/10
else
    unidades=regbenav//10 : decenas=0
endif

'Benavides

portd.0=unidades.0
portd.1=unidades.1
portd.2=unidades.2
portd.3=unidades.3
pause 1
porta=%001000 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1

porta=%000010 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1

portd.0=decenas.0
portd.1=decenas.1
portd.2=decenas.2
portd.3=decenas.3
pause 1
porta=%000100 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1

porta=%000001 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1

'Bolochera

if regbolich>=10 then
    unidades=regbolich//10 : decenas=regbolich/10
else
    unidades=regbolich//10 : decenas=0
endif

portd.0=0
portd.1=1
portd.2=0
portd.3=1

```

```
pause 1
porta=%100000 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
portd.0=0
portd.1=1
portd.2=0
portd.3=1
pause 1
porta=%010000 : porte.0=0 : porte.1=0
pause 1
porta=%000000 : porte.0=0 : porte.1=0
pause 1
```

```
next VE2
```

```
RETURN
```

```
end
```