

# **INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE**

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial  
15018, publicado en el Diario Oficial de la Federación el 29 de noviembre de 1976.

---

Departamento de Electrónica, Sistemas e Informática  
ESPECIALIDAD EN SISTEMAS EMBEBIDOS



## **Sistema de detección de tránsito vehicular**

Tesina para obtener el grado de:

**ESPECIALISTA EN SISTEMAS EMBEBIDOS**

Presentan: **ALEJANDRO BIZARRO, JUAN HERRERA, JULIO  
VILLANUEVA.**

Director:

**Dr Jorge Pardiñas Mir**

San Pedro Tlaquepaque, Jalisco; a 11 de julio de 2018



# **DEDICATORIA**

Queremos agradecer al Consejo Nacional de Ciencia y Tecnología (CONACYT) por su apoyo mediante las Becas otorgadas: 859542, 857108 y 858766.

A nuestras familias por su apoyo incondicional y su paciencia durante este tiempo.

A nuestros profesores y compañeros del curso, por su ayuda y consejos.

# RESUMEN

La tecnología de detección de tráfico vehicular ha demostrado ser un área de diseño avanzada, capaz de lograr mejoras sustanciales en un rango de aplicaciones debido a su paradigma de la estructura de toma de decisiones basado en el razonamiento cognitivo. En este sentido, la tesis presenta el desarrollo de un avance en la tecnología de detección vehicular que permite incluir un modelo formal basado en un agente autónomo e inteligente capaz de detectar vehículos a través de un solo sensor, de acuerdo a las exigencias y limitaciones de las avenidas. Se utilizaron los periféricos de UART (Universal Asynchronous Receiver-Transmitter), RTC (Real Time Clock), DMA (Direct Memory Access), para el manejo de los datos, así como una aplicación como interfaz de usuario para mostrar la cantidad de tránsito. Este proceso mejora efectiva e inmediatamente la calidad del servicio en una intersección, aumentando el rendimiento de la movilidad de los vehículos. Este prototipo tiene como ventaja no solo la mejora en tiempo de traslados, sino que también el costo de producción. Se realizaron una serie de pruebas en campo con el fin de comparar la funcionalidad, ventajas y desventajas sobre otros sistemas de detección ya creados, con la intención de alcanzar un control de tránsito más barato y eficiente.

## *ABSTRACT.*

Vehicle traffic detection technology has proven to be an area of advanced design, capable of achieving substantial improvements in a range of applications due to its paradigm of decision-making structure based on cognitive reasoning. In this sense, this thesis presents the advance of a vehicular detection technology that allows the inclusion of a formal model based on an autonomous and intelligent agent capable of detecting vehicles through a single sensor, according to the demands and limitations of avenues in big cities. The peripherals of UART (Universal Asynchronous Receiver-Transmitter), RTC (Real Time Clock), DMA (Direct Memory Access), were used for the handling of the data as well as an application implemented as a user interface to show the amount of traffic. This process effectively and immediately improves the quality of service at an intersection, increasing the performance of vehicle mobility. This prototype has the advantage of not only improving the transfer time but also the cost of production. A series of field tests were carried out in order to compare the functionality of the developed traffic detection system prototype, as well as the advantages and disadvantages with other detection systems already available in the market, and the results show that this prototype was more efficient and affordable to detect traffic in avenues.

# CONTENIDO.

DEDICATORIA .....	3
RESUMEN .....	4
<i>ABSTRACT</i> .....	5
1. ANTECEDENTES. ....	14
1.1. Descripción del problema. ....	14
1.2. Productos existentes, otras soluciones propuestas. ....	14
2. MARCO TEÓRICO.....	17
2.1. Dispositivos para monitorear el tráfico vehicular. ....	17
2.2. Microcontrolador. ....	18
2.3. Sensores para el monitoreo de tráfico vehicular. ....	18
2.4. Sensores Ultrasónicos. ....	18
2.5. Sensores Infrarrojos. ....	20
2.6. El radar. ....	20
2.7. Sensores capacitivos. ....	22
2.8. Sensores inductivos.....	23
2.9. Sensores de visión.....	24
2.9.1 Dispositivos para el movimiento del sistema de monitoreo. ....	24
2.9.2. Pan & Tilt.....	24
2.9.3. Servomotores .....	25
3. METODOLOGÍA. ....	27
3.1. Pruebas con el sensor .....	28
3.2. Adquisición de datos.....	30
3.3. Algoritmo de procesamiento de la señal. ....	31
3.4. Implementación para la medición. ....	32
3.5. AFEC. ....	37
3.6. DMA. ....	38
3.7. RTT.....	39
3.8. Descripción de la base utilizada.....	40
3.9. Funcionalidad realizada en la PC.....	46

4. RESULTADOS.....	51
4.1. Descripción del tipo de pruebas que se realizaron.....	51
4.2. Pruebas de comunicación.....	52
4.3. Ventajas encontradas en comparación a proyectos anteriores. ....	56
4.4. Señales de onda.....	58
4.5. Pruebas de la base Pan & tilt: .....	60
4.6. Resultado de cálculo de detección de tránsito .....	61
5.CONCLUSIONES. ....	63
6.Apéndice .....	65

# TABLA DE FIGURAS.

Figura 1. Ilustración de funcionamiento general del sistema. ....	12
Figura 2. Sensor Ultrasónico – Medición. ....	19
Figura 3.Sensor Ultrasónico - Zona de medición. ....	20
Figura 4.Sistema de radar – Monoestático.....	22
Figura 5.Sistema de radar – Biestático. ....	22
Figura 6.Señal PWM generada por el microcontrolador para generar el movimiento de la base. 26	
Figura 7.Movimiento del motor en relación con el ciclo de trabajo. ....	26
Figura 8.Pruebas realizadas en campo. ....	30
Figura 9.Diagrama de conexión para registro de datos.....	30
Figura 10.Respuesta del sensor de radar con tránsito fluido. ....	31
Figura 11 Filtrado de señal obtenida.....	34
Figura 12 Detección de objeto. ....	35
Figura 13 Validación del estado de tránsito vehicular.....	36
Figura 14 Inicialización del módulo AFEC.....	37
Figura 15 Inicialización del módulo DMA.....	38
Figura 16 Configuración del temporizador RTT. ....	39
Figura 17 Diagrama de conexión de base pan and tilt junto con sensor y tarjeta procesadora de datos.....	41
Figura 18 Código de movimiento de motor en una base de tiempo. ....	43
Figura 19 Código de movimiento de motor en base a movimiento de los ejes. ....	44
Figura 20 Código de movimiento de motor en base a movimiento de los ejes. ....	45
Figura 21 Configuración USART1.....	47
Figura 22 Configuración USART1.....	47
Figura 23 Terminal PuTTY – Ejemplo de mensaje.....	48
Figura 24 Terminal PuTTY – configuración de parámetros.....	48
Figura 25 Interface gráfica – GUI.....	49
Figura 26 Recepción mensaje de inicio de comunicación.....	52
Figura 27 Respuesta de la GUI del Mensaje \$AAA* .....	54
Figura 28 Respuesta de la GUI del Mensaje \$BBB* .....	54
Figura 29 Respuesta de la GUI del Mensaje \$CCC* .....	55
Figura 30 Respuesta de la GUI del Mensaje \$DDD* .....	55
Figura 31 Respuesta de la GUI del Mensaje \$EEE* .....	56
Figura 32 Respuesta del PWM para movimiento a la izquierda.....	58
Figura 33 Respuesta del PWM para detener el servo motor.....	59
Figura 34 Respuesta del PWM para movimiento del servomotor a la derecha. ....	59
Figura 35 Respuesta de la FRDM k64 de detección vehicular. ....	60
Figura 36 Señal medida con sensor en un escenario de tránsito ligero. ....	61
Figura 37 Salida del sistema para una señal de tránsito ligero. ....	62



## **LISTA DE TABLAS.**

Tabla 1 Características técnicas del sensor.....	28
Tabla 2 Características técnicas de tarjeta FRDM K64.....	29
Tabla 3 Rangos de detección .....	33
Tabla 4 Significado de los estados del sensor y su tiempo de animación.....	50
Tabla 5 Mensajes y animaciones en la GUI.....	53

# LISTA DE ABREVIATURAS

<b>ADC</b>	Convertido Analógico a Digital (Analog to Digital Converted)
<b>AFEC</b>	Controlador de Entrada Analógico (Analog Front-End Controller)
<b>CMOS</b>	Semiconductor Complementario de Óxido Metálico (Complementary Metal-Oxide-Semiconductor)
<b>DC</b>	Corriente continua (Direct Current)
<b>DMA</b>	Memoria de Acceso Directo (Direct Memory Access)
<b>DSP</b>	Procesador de Señal Digital (Digital Signal Processor)
<b>GUI</b>	Interfaz Gráfica del Usuario (Graphical User Interface)
<b>mA</b>	Miliamperio (Milliampere)
<b>MC</b>	Microcontrolador (Microcontroller)
<b>ms</b>	Milisegundos (Milliseconds)
<b>mV</b>	Milivolt (Millivolt)
<b>RF</b>	Frecuencia de Radio (Radio Frequency)
<b>RTT</b>	Temporizador en Tiempo Real (Real-time Timer)
<b>RX</b>	Recepción de datos (Data Reception)
<b>SRAM</b>	Memoria Estática de Acceso Aleatorio (Static Random-Access Memory)
<b>UART</b>	Transmisor-Receptor Asíncrono Universal (Universal Asynchronous Receiver-Transmitter)
<b>USAR</b>	Receptor / Transmisor Asíncrono Universal (Universal Asynchronous Receiver/Transmitter)
<b>WPF</b>	Fundación de presentación de Windows (Windows Presentation Foundation)

# INTRODUCCIÓN

El presente proyecto está destinado a la detección de tránsito vehicular mediante un sistema basado en un sensor de radar y un mecanismo para direccionar. Discutimos el control de congestión y el problema de detección de tránsito para las principales avenidas que puede llegar a tener una ciudad. La congestión vehicular causa problemas graves de movilidad y contaminación, afectando la calidad de vida de los ciudadanos. La propuesta para nuestro proyecto de titulación es el diseño de un sistema que ayude a mejorar la movilidad y que esté dirigido hacia el reconocimiento del tránsito en tiempo real.

El sistema consiste en la detección de tránsito mediante un sensor de radar que proporciona una medición de distancia mediante una salida normalizada de 4-20 mA. Estará montado sobre una base que le proporciona libertad de movimiento en los ejes cartesianos XY, con dos servomotores que permitirán el direccionamiento del sensor para poder escanear la avenida a lo largo y a lo ancho.

La base móvil facilitará el escaneo rápido en diferentes direcciones sobre un plano XY. Dicho plano está representado por la superficie de la avenida. Debido a que el sensor estará instalado a una distancia conocida, el sistema sabrá las distancias esperadas para una calle sin presencia de vehículos. Una vez que exista presencia vehicular la señal del sensor será reflejada por la superficie del vehículo al cual en ese momento esté apuntado, por lo cual la medición de distancia variará. Las variaciones de distancia serán analizadas por un algoritmo, que permitirá conocer el estado actual del tránsito vehicular. Debido a las características de estos sensores de radar, incluso pueden tener un alcance de 120 metros.

El software diseñado para el sistema embebido se encargará de las siguientes tareas:

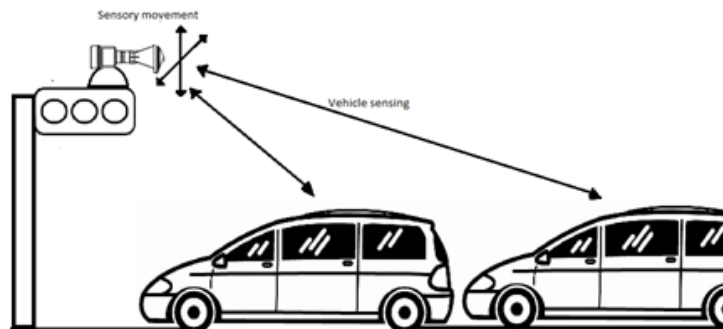
- Procesamiento de la señal normalizada de 4-20 mA que proporciona el sensor de radar, que indica la distancia medida desde el sensor hasta el obstáculo encontrado, siendo la referencia máxima el nivel de calle.
- Control de base móvil para el radar, que consta de dos servomotores y que proveerá dos movimientos angulares.

- Detección de tránsito vehicular en 1 carril sin movimiento del sensor.
- Detección de tránsito vehicular en 1 carril mediante sensor en movimiento (escaneo a lo largo de la calle).
- Transmisión de datos mediante UART.

La información de salida que generará el sistema será útil para los sistemas de control de tránsito, ya sean locales o a nivel metropolitano, lo cual implica que podrían ser aprovechados por semáforos inteligentes.

Como información acerca de los sensores de radar, éstos trabajan en el rango de banda de frecuencia K, la cual oscila entre 18 y 27 GHz. Estos sensores son productos terminados que fabrican numerosas marcas a nivel mundial y entre los cuales se ha considerado un sensor de marca alemana.

A continuación, se muestra un diagrama de cómo estaría montado el sistema.



*Figura 1. Ilustración de funcionamiento general del sistema.*

Una de las razones por la cual nos decantamos por este sistema de detección es que usualmente se hace el uso de cámaras de video para hacer esta medición. El uso de cámaras para la detección de tránsito demanda un sistema embebido potente y, por ende, costoso; además su consumo de energía es mayor. El procesamiento de imágenes de video necesita un sistema embebido poderoso y algoritmos complejos para poder abstraer la información de tránsito necesaria. En cambio, un sistema basado en el sensor propuesto requiere un procesador menos potente, ya que la señal a procesar es una señal normalizada a 4-20 mA, donde 4 mA representa la distancia mínima y 20 mA la distancia máxima de acuerdo a su calibración. El procesamiento de

esta señal se reduce a utilizar el ADC del sistema embebido. Dos servomotores serán controlados por un algoritmo que priorice el ahorro de energía basándose en una rutina diseñada para activar el movimiento durante el tiempo mínimo necesario.

# **1.ANTECEDENTES.**

## **1.1. Descripción del problema.**

Con un aproximado de 33.4 [1] horas al mes atrapados en congestiones viales, los automovilistas de Guadalajara superan la media nacional de 28 horas [2] y es la segunda ciudad del país donde los conductores desperdician más tiempo durante su traslado. El primer lugar nacional es la Ciudad de México, donde los conductores invierten 61.5 [1] horas al mes manejando en atorones viales. Esta problemática nos ha motivado a desarrollar un sistema que ayude a optimizar los tiempos de traslado de los ciudadanos, mediante la detección en tiempo real del tránsito vehicular que permitirá poder controlar los semáforos, sobre todo en avenidas principales, ya que actualmente el control del tiempo de los semáforos es ineficiente e inexacto, debido a que en avenidas principales los tiempos de espera cambian entre 30 segundos y ocho minutos.

## **1.2. Productos existentes, otras soluciones propuestas.**

Una de las primeras ideas para el control de tránsito vehicular era la instalación de sensores sobre cada vehículo; en el artículo *“24-GHz-FM-CW-Radar for Detection of Information for Traffic Purposes”*, creado en 1984 por el doctor Dr. G. Seehausen, se plantea la idea de utilizar un radar de 24 Ghz, que se comunicará con un sistema pasivo reflector de microondas que se encontraría sobre el borde del camino. Dado que el sensor es capaz de medir las distancias entre los vehículos y el sistema reflector de microondas mediante la información proporcionada, se podría calcular el desplazamiento del vehículo y así almacenar y proporcionar datos estadísticos del tránsito. [3]

Los autores Zhijun Li, Chunxiao Li, Yanan Zhang and Xuelong Hu, en su tesis llamada "Sistema inteligente de control de semáforo en función de los flujos de tráfico en tiempo real", resolvieron el problema de tránsito al utilizar un sistema de detección ultrasónico ubicado en las intersecciones principales para obtener la cantidad de flujo de tránsito, en el que el transmisor envía una señal ultrasónica a una frecuencia de 40 Khz con una longitud de onda de 6 mm hacia

el tránsito que, al ser reflejada y luego recibida por un receptor transductor con efecto piezoeléctrico, genera una señal de voltaje débil en el rango de los mV que después es amplificada. El sistema detecta el tiempo en que tarda en responder la señal y lo compara con el tiempo necesario al recorrer la distancia de prueba para obtener la cantidad de tráfico. Este sistema debe desplegar 2 ó 3 grupos de sensores ultrasónicos por carril. [4]

Una de las alternativas al sensor de radar es el uso de cámara de video. Aplicando algoritmos de visión artificial sobre las imágenes obtenidas podemos estimar el número de vehículos y velocidades, entre otros.

En el documento “*Real-time parallel implementation of road traffic radar video processing algorithms on a parallel architecture based on DSP and ARM processors*”, para solucionar el problema de análisis del tránsito vehicular, los ingenieros plantearon la idea de utilizar imágenes de video procesadas mediante el DSP (*Digital Signal Processor*) 66AK2H12 de *Texas Instrument*. Contiene ocho núcleos DSP y cuatro núcleos ARM; cada núcleo DSP puede acumularse a 1.25 GHz y proporciona un rendimiento máximo de 160 GFLOP para un punto flotante y 320 GMAC para un punto fijo sólo 10 vatios. En la tesis plantean la optimización del algoritmo de cadena para la detección y reconocimiento de objetos. [5]

De acuerdo al artículo “*A Low-cost Vehicle Detection and Classification System based on Unmodulated Continuous-wave Radar*” de los autores J.Fang, H. Meng y H. Zhang, la detección de tránsito vehicular basado en radares que trabajan en la banda K de frecuencia proporciona un prometedor rendimiento y precisión de acuerdo a sus resultados experimentales. También se hace la referencia a que dicho tipo de sensores tienen la ventaja de no ser vulnerables al clima, lo cual los hace ideales para este tipo de aplicación. Un tema más que se aborda en este artículo es el bajo costo de la implementación del uso del radar, que fue menor a \$200 USD, así como el pequeño tamaño del sensor y una simple estructura para su montaje hace que esta aplicación sea competitiva en comparación con otras tecnologías como detección por video, ultrasonido, infrarrojos etc. [6]

Consultando la investigación denominada “*A Summary of Vehicle Detection and Surveillance Technologies used in Intelligent Transportation Systems by The Vehicle Detector Clearinghouse*” [7], los sensores de radar de microondas diseñados para la recopilación y el monitoreo de datos de tránsito trabajan bajo el principio del efecto *doppler* y son frecuentemente utilizados en las carreteras de los EE. UU.

Los sensores transmiten una señal hacia un área de la carretera desde una antena aérea. El área en la que se transmite la señal es directamente proporcional al alcance y detección en área que es capaz de percibir. El fabricante generalmente establece las restricciones de diseño del sensor de radar. Cuando un vehículo pasa a través del haz de la antena, una parte de la señal transmitida se refleja hacia la antena. La señal luego ingresa a un receptor donde se realiza la detección y se calculan los datos del vehículo, como el volumen, la velocidad, la ocupación y la longitud.

El sensor de radar puede montarse sobre el medio de un carril, o en el costado de una carretera para medir los parámetros de tráfico a través de varios carriles. Los radares orientados hacia los vehículos reúnen datos representativos del flujo de tránsito. Como desventaja de este tipo de sensores es que no son capaces de detectar vehículos cuando estos están detenidos a menos que cuenten con un sensor auxiliar, así como la eficiencia de estos en ocasiones puede variar al momento de la detección de objetos. [7]

Una investigación similar llamada “*A novel signal processing technique for vehicle detection radar*” de los autores Sang Jin Park, Tae Yong Kim, Sung Min Kang and Kyung Heon Koo, demuestra el efectivo uso de los radares para la medición de velocidad y clasificación por tamaño de vehículos en múltiples carriles. El objetivo de estas mediciones es el de obtener información acerca del tránsito vehicular. La investigación se basó en el uso de un radar de microondas con una frecuencia de 24 GHz que se monta en la parte superior de un poste sobre la avenida. Los resultados obtenidos en la clasificación del tamaño de los vehículos demuestran una exactitud de más del 90 %. Los resultados obtenidos en relación la medición de la velocidad de los vehículos demuestra una exactitud de 95 % para mediciones en un solo carril y del 90 % para la medición en 2 carriles simultáneos. [8]



## **2. MARCO TEÓRICO.**

El presente proyecto está compuesto por módulos, con una funcionalidad que en conjunto realizan la tarea principal del sistema, que es la detección de tráfico vehicular.

Este sistema funciona de manera general de la siguiente manera: un sensor de radar que proporciona una medición de la detección de vehículos generará una salida normalizada de 4-20 mA, que será procesada por un módulo maestro y en base a estos datos definirá la cantidad de tráfico que está ocurriendo. Este sensor estará montado sobre una base móvil que facilitará el escaneo rápido en función de la velocidad de respuesta del sensor y que le proporcionará dos grados de movimiento en el plano X y en el plano Y, con dos servomotores que permitirán el direccionamiento del sensor para poder escanear la avenida a lo largo y a lo ancho.

### **2.1. Dispositivos para monitorear el tráfico vehicular.**

En base a lo descrito de manera general se pueden definir los módulos que integran al sistema completo:

- Base para giro e inclinación (pan & tilt).
- Servomotores.
- Sensor de radar.
- MC (Microcontrolador).
- Tarjeta para procesar la salida del sensor (convertidor de corriente a voltaje

Para poder definir estos módulos se desarrollarán los siguientes temas:

- Microcontrolador.
- Sensores para el monitoreo de tráfico vehicular.
- Dispositivo para el movimiento del sistema de monitoreo.

## **2.2. Microcontrolador.**

El microcontrolador utilizado es de 32 bits del proveedor ATMEL que está integrado en una tarjeta de desarrollo llamada SAMV71. Es una tarjeta de desarrollo utilizada especialmente para aplicaciones automotrices; trabaja a una velocidad de 300 MHz con una memoria flash de 2048 kbytes y una SRAM (*Static Random Access Memory*) de 384 kbytes. Además, cuenta con periférico PWM para generar el control de los servomotores, así como un DMA (*Direct Memory Access*) integrado que se utiliza para acceder a los datos provenientes del sensor a través del módulo ADC (convertidor analógico digital). El DMA transfiere las mediciones del sensor a memoria RAM, que al procesarlas con un algoritmo que será descrito en la metodología generará los resultados para poder determinar la cantidad de tránsito vehicular.

## **2.3. Sensores para el monitoreo de tráfico vehicular.**

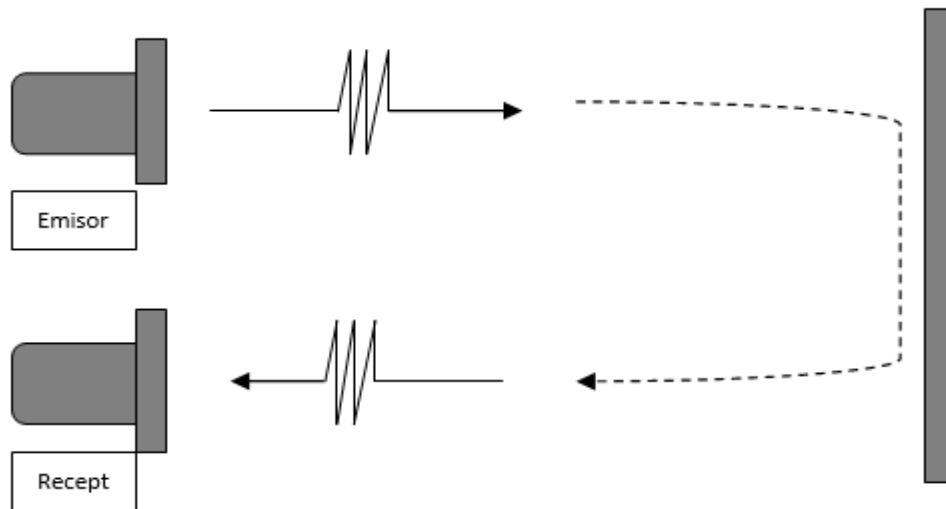
Un sensor es un dispositivo que, a partir de la energía del medio donde se mide, da una señal de salida manipulable en función de la variable física medida; se refiere a la ampliación de los sentidos para adquirir un conocimiento de cantidades físicas que, por su naturaleza o tamaño, no pueden ser percibidos directamente por los sentidos. [10]

## **2.4. Sensores Ultrasónicos.**

Los sensores ultrasónicos son muy utilizados en sistemas de medición no invasivos para determinar la distancia del emisor a un objeto dado; otra aplicación es la medición de distancias, profundidades y alturas que cambian dinámicamente. Las principales ventajas de este tipo de sensores es que al ser un tipo de medición no invasiva (no requiere contacto alguno para realizar la medida), la variedad de objetos que puede medir es muy amplia, es posible realizar las mediciones en superficies variables, líquidos y ambientes hostiles.

Se dice que su onda sonora [MC1] es ultrasónica cuando está por encima de la frecuencia audible para el oído humano (por encima de los 20 kHz). El principio de funcionamiento de estos sensores consiste en la emisión de una honda de manera cíclica, la cual es de alta frecuencia y

corta duración, además de que se propaga en el medio. Al encontrar un objeto en su paso, esta es reflejada y vuelve en forma de eco al receptor (vea la figura 4). El circuito de acondicionamiento tiene la tarea de determinar el periodo transcurrido entre la emisión de la señal acústica y la recepción del eco.



*Figura 2. Sensor Ultrasónico – Medición.*

EL sensor ultrasónico permite medir distancias entre 20 mm hasta 10 m. Cuando los sensores ultrasónicos se colocan, ya sea de manera contigua o frente a frente, se debe guardar una distancia mínima entre estos, dependiendo de la zona de operación del sensor (vea la Figura 3), lo cual depende de la zona de detección (zona en la cual es posible detectar un objeto), la zona ciega (zona en la cual un objeto no puede ser identificado y está muy próxima al sensor), la zona de exploración funcional (zona en la cual el objeto funciona de forma típica) y la zona de exploración límite (zona en la cual es posible detectar objetos que presentan buenas propiedades acústicas).  
[11]

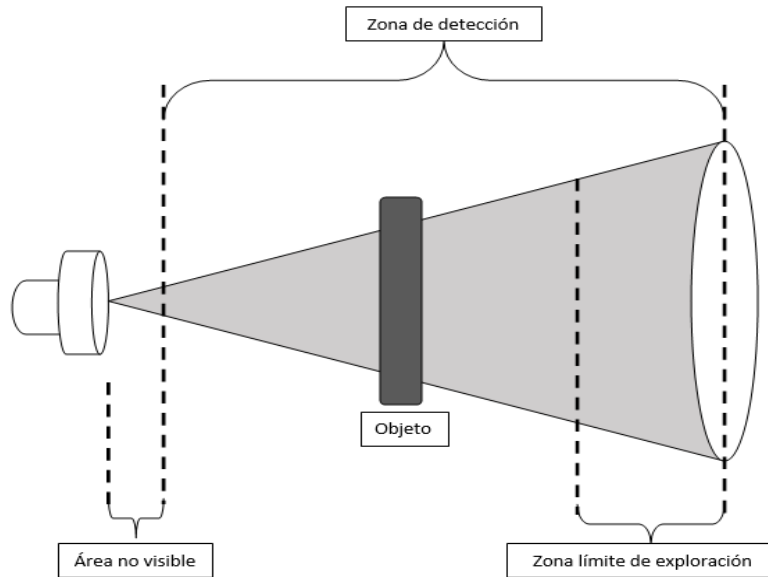


Figura 3. Sensor Ultrasónico - Zona de medición.

## 2.5. Sensores Infrarrojos.

Los sensores infrarrojos son dispositivos fotoeléctricos, emisores y receptores, cuyo espectro de trabajo se encuentra en la longitud de onda infrarroja (entre los 700 y 900 nanómetros). Aunque esta luz no sea visible por el ojo humano, al ser una radiación electromagnética, se puede ver afectada por infinidad de factores ambientales, es por eso que para determinadas aplicaciones se debe utilizar asociada a un módulo de emisión-recepción. [11].

## 2.6. El radar.

El radar es una aplicación importante de la tecnología de microondas. Es la forma abreviada de detección de radio y rango. Se utiliza ampliamente para recopilar información sobre objetos distantes mediante la transmisión de señales electromagnéticas hacia el objeto y el análisis de los ecos recibidos.

Una propiedad importante de la señal electromagnética se usa para detectar la ubicación del objeto. Cuando la onda electromagnética encuentra un cambio repentino en la conductividad en el medio, parte de la energía electromagnética se absorbe y el resto de la energía se refleja o se re-irradia. Esta señal electromagnética reflejada se conoce como eco. Este eco se interrumpe desde el objeto y es recibido por la antena y se analiza para localizar el objetivo.

Cualquier sistema de radar consta de cuatro componentes principales:

- a) Transmisor.
- b) Antena.
- c) Receptor.
- d) Exhibir y controlar el equipo.

El transmisor proporciona la señal de RF (Radiofrequency) con suficiente potencia y la envía a la antena. La antena irradia estas señales de RF en el espacio en la dirección deseada. La señal se propaga en el espacio y parte de la señal es interceptada por cuerpos de reflexión que se mueven en el espacio libre. La antena recoge la señal reflejada y se conoce como señal recibida. Estas señales recibidas son muy débiles y, por lo tanto, se amplifican y luego se detectan. La presencia de cualquier objeto o destino causa un eco de la señal transmitida en la señal recibida. La salida del detector aumenta repentinamente cuando se detecta un eco. El tiempo entre la transmisión y la recepción del eco da el rango del objetivo.

Existen dos sistemas de radar básicos: sistema mono-estático (figura 6) y sistema biestático (figura 7). Los sistemas de radar monoestático usan la misma antena para transmisión y recepción. El sistema de radar biestático usa dos antenas separadas para estas funciones. [11]

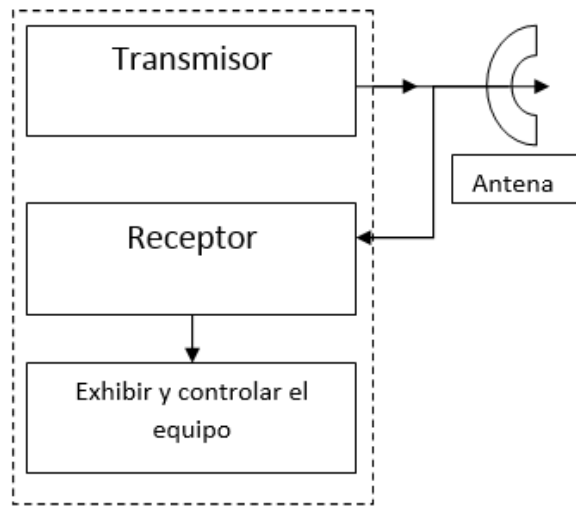


Figura 4. Sistema de radar – Monoestático.

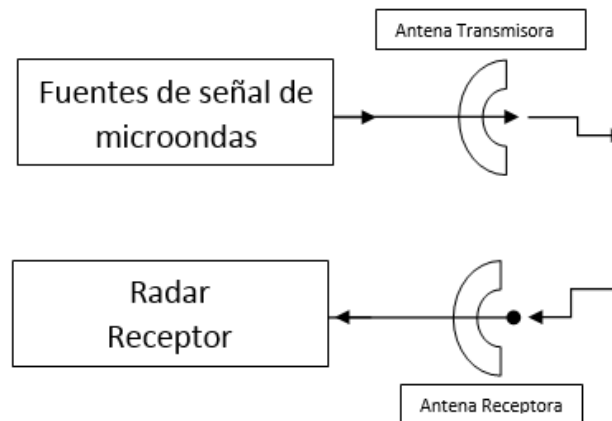


Figura 5. Sistema de radar – Bistático.

## 2.7. Sensores capacitivos.

Estos sensores se caracterizan por generar un campo electrostático, a diferencia de los sensores inductivos de proximidad que generan un campo electromagnético. Un sensor de proximidad capacitivo está formado por dos electrodos en forma de discos concéntricos. Cuando un objeto se aproxima al sensor e interactúa con el campo electrostático formado entre estos, cambia la capacitancia equivalente del dispositivo. Este cambio de capacitancia produce un cambio

de estado en el sensor, activando un circuito oscilador, el cual, a su vez, dispara el circuito activador, lo que produce a la salida un cambio de estado bajo a estado alto.

La distancia de detección de un objeto se puede incrementar al aumentar el tamaño de los electrodos. De manera teórica, un sensor capacitivo puede detectar objetos hasta 13 cm de distancia; sin embargo, en la práctica esto no es posible debido a los voltajes de operación de los sensores que son apenas de unos cuantos volts, además de las variaciones dieléctricas del aire. Esta constante varía con la temperatura, la presión atmosférica y la humedad relativa del medio, llevando las distancias de detección típicas entre los 20 y los 80 mm. [11].

## **2.8. Sensores inductivos.**

Los detectores de proximidad inductivos transversales se basan en el principio de que la inductancia de una bobina cambia en forma considerable en presencia de un conductor. En general, la bobina se coloca como parte de un circuito de puente. La presencia del conductor cerca de la bobina tiene la tarea de sacar de balance al circuito puente del que forma parte la bobina. Este puede utilizarse para operar un circuito de cambio de estado bajo, a estado [MC1] alto. Estos sensores se conocen como sensores autoinducidos, debido a que cuando se aproxima el conductor cambia la inductancia equivalente del circuito.

Una de las ventajas de estos sensores es que no requieren de contacto con el objeto a detectar, lo que minimiza el desgaste. Sin embargo, la principal desventaja es que solo son sensitivos a materiales conductores y dependen en gran medida del tipo de conductor al que se trate; las distancias nominales de detección van desde unas décimas de milímetros hasta centenas de milímetros, dependiendo de la calidad del sensor y el fabricante. [11].

## **2.9. Sensores de visión.**

Los sensores de visión son dispositivos avanzados (generalmente cámaras [MC1]) encargados [MC2] de obtener información del medio a través de la captura de imágenes o video, para su posterior procesamiento.

Un sensor de visión es considerado un sensor inteligente, ya que está compuesto por una pluralidad de sensores, unidades encargadas de procesar la información que entrega esta pluralidad de sensores y convertirla en una serie de imágenes entendibles para el usuario.

Estos sensores forman imágenes, las cuales están compuestas por elementos llamados píxeles y cada pixel es obtenido por una celda unitaria; estas celdas están conformadas por sensores de estado sólido basados en tecnología CMOS (*Complementary metal-oxide-semiconductor*).

Un chip de visión está conformado por un arreglo bidimensional de sensores de estado sólido que se encargan de obtener la luz del exterior, una línea de control de acceso vertical y una línea de acceso horizontal, para determinar qué pixel fue activado. Este arreglo bidimensional de sensores de estado sólido representa los píxeles de cámara, donde cada pixel está formado por un elemento fotosensible y un circuito de lectura de la señal de salida que se encarga de leer el valor de un pixel XY específico. [11].

### **2.9.1 Dispositivos para el movimiento del sistema de monitoreo.**

#### **2.9.2. Pan & Tilt**

Se utilizó el tipo de base PT2645-S Pan & Tilt por su flexibilidad, tanto en diseño como en su manejo. Cuenta con dos ejes de movimiento uno en el plano X y otro en el plano Y, y; además, una base en uno de sus ejes para montar de manera más sencilla y sin tanto material extra el sensor (se puede ver en: [9]).

Como se observa en [9] esta base cuenta con dos engranes que mejoran la eficiencia de movimiento. Cada eje posee un motor de tipo servo que permitirá el movimiento continuo de cada eje, estos se controlan a través de una señal PWM que será generada por el MC.



La razón principal al seleccionar esta base fue porque puede generar un movimiento continuo de 360 grados en cada uno de sus ejes. Con esto pudimos controlar cada eje de manera que fuese capaz de escanear cualquier tipo de avenida sin importar que tan amplia fuera esta. Esto permite que sea posible el uso de un sensor de radar puntual, a diferencia de otro tipo de detectores donde se necesita una antena más grande para abarcar el rango completo de la vialidad que se está analizando, esto impacta en un costo menor.

### 2.9.3. Servomotores

Un servomotor es un motor parecido a un motor de corriente directa convencional, que tiene la capacidad de ser controlado por el usuario a un ángulo definido por este, es decir estos motores pueden trabajar en movimiento continuo, girar solo 180 grados, 90 grados o cualquier ángulo que nosotros deseemos posicionar el sensor dependiendo la aplicación en donde los estemos utilizando

Estos sensores se controlan mediante una señal PWM (modulación por ancho de pulso), dependiendo el ciclo de trabajo en la que esté trabajando se puede tener un ángulo de movimiento como se muestra en la figura 6. [3]

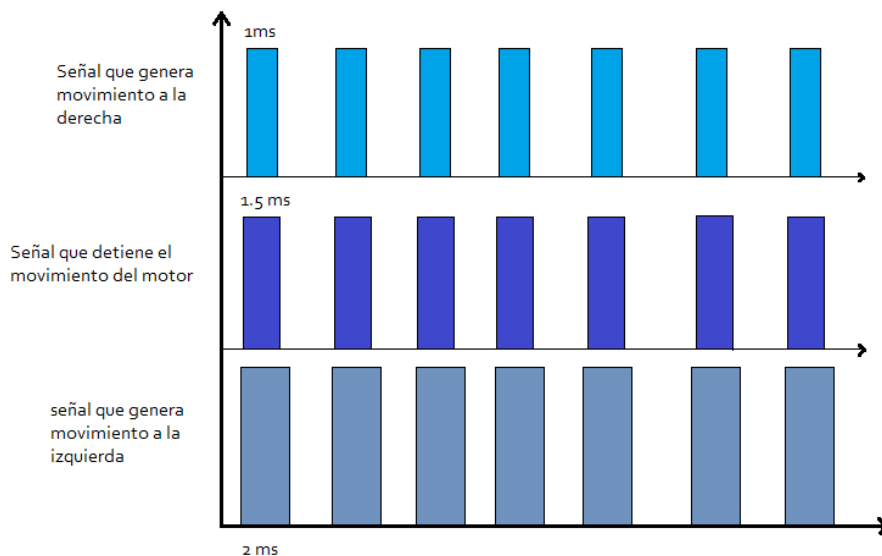


Figura 6. Señal PWM generada por el microcontrolador para generar el movimiento de la base.

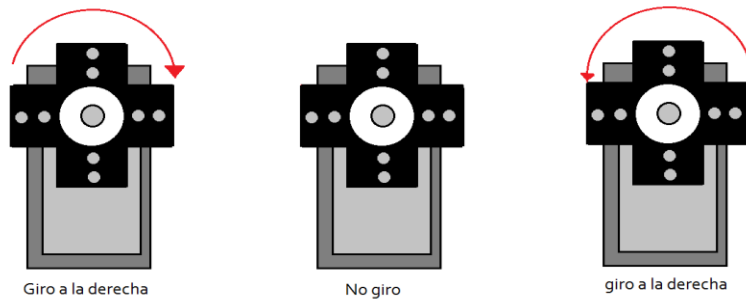


Figura 7. Movimiento del motor en relación con el ciclo de trabajo.

Este sensor se utilizó porque la base Pan & Tilt ya contaba con este en ambos ejes de movimiento. Nosotros solo realizamos el software para poder generar la señal de control (Figura 3) a través del MC.

### 3. METODOLOGÍA.

¿Por qué usar un radar?

La decisión de usar el radar como elemento de detección de tránsito vehicular se basó en los siguientes criterios:

- Señal analógica de fácil procesamiento.
- Sensor de menor costo en comparación con cámaras de video.
- Medición confiable.
- Peso ligero de fácil instalación y manipulación.

El sensor de radar genera una señal de salida normalizada de 4 – 20mA, la cual es fácilmente obtenida por el sistema de detección de tránsito a través de un puerto ADC. La señal es variante en el tiempo con bajo nivel de ruido, lo cual no requiere un nivel de procesamiento elevado. Debido a esto, el requerimiento de desempeño del procesador es bajo y esto permite que su costo no sea elevado. El sensor de radar es considerablemente de menor costo que una cámara de video utilizada para la detección de tránsito, ya que requiere componentes menos costosos, de tal forma que abarata el costo del sistema. También, y debido a que la aplicación solo requiere detección y no análisis de imágenes, el uso de una cámara de video no es necesario.

La tecnología de radar está muy desarrollada, permitiendo que la respuesta del sensor sea predecible y confiable. Las señales de microondas de radar rebotan fácilmente en materiales como el acero y aluminio lo que hace que los vehículos serán detectados de forma confiable. A su vez, las señales de microonda de radar no se ven afectadas por los cambios de temperatura, luminosidad, el viento y el polvo. En cambio, los otros sensores como los ultrasónicos y las cámaras de video si son susceptibles a las condiciones ambientales, como la luminosidad y la temperatura. Las señales de radar incluso pueden superar climas con lluvia y niebla.

Los componentes que necesita un radar para su funcionamiento son mínimos y ligeros, facilitando y abaratando su montaje. Se conoce que el tránsito vehicular es una variable que no

varía instantáneamente, sino que tendrá una respuesta lenta en el orden de los minutos, ya que este no cambia de valor mínimo a máximo en un instante, sino que es un acumulado en el tiempo de una medición constante durante varios minutos, por esta razón sabemos que la capacidad de respuesta del radar no está limitada a ser de alta velocidad.

El protocolo sobre el cual está basado la comunicación entre el sensor y el microcontrolador es una señal analógica de corriente variante de 4 – 20mA, donde 4mA representa la distancia mínima entre el sensor y el objeto, y 20 mA representa la máxima distancia entre el sensor y el objeto. Las señales 4-20 normalmente se acompañan del protocolo llamado Hart, que va montado sobre la señal 4-20. Para esta aplicación no se usará el protocolo Hart. Antes de procesar la señal por el microcontrolador la señal 4-20mA se convertirá a una señal de 0-5v mediante una tarjeta convertidora.

### 3.1. Pruebas con el sensor

Inicialmente se llevaron a cabo pruebas de detección con el sensor a utilizar para obtener una caracterización de la respuesta del sensor. El sensor utilizado es un VEGA pulse 67 que está diseñado para hacer detecciones de objetos sólidos. En la tabla 1 se muestra una descripción de las características técnicas del sensor utilizado.

*Tabla 1 Características técnicas del sensor.*

Marca	VEGA
Modelo	VEGAPULS 67
Rango de medición	15 m (49.21 ft)
Desviación	±2 mm
Salida	4 ... 20mA/HART
Temperatura de proceso	-40 ... +80 °C (-40 ... +176 °F)

Presión de proceso	-1 ... +2 bar/-100 ... +200 kPa (-14.5 ... +29.0 psig)
Voltaje de operación	9.6 ... 48 V DC, 20 ... 42 V AC, 50/60 Hz

Una vez seleccionado el sensor, se eligió un lugar adecuado para hacer las pruebas de detección de vehículos en circulación. El lugar seleccionado es el puente peatonal del ITESO sobre la avenida Periférico Sur, que nos da acceso a la obtención de varios tipos de muestras. En esta prueba se busca obtener un registro de datos para un escenario con tránsito fluido y tránsito denso. A partir de estos datos, se puede generar reproducciones y variaciones de este log para poder simular la señal con un micro controlador y así realizar pruebas en laboratorio sin necesidad de hacerlas en campo. La tarjeta utilizada para reproducir la señal del sensor es la FRDM-k64F del fabricante de Semiconductores Freescale. Se utilizó el periférico del convertidor digital a analógico DAC para hacer una reproducción de los datos registrados en las pruebas en campo. En la tabla 2 está una descripción de las características de esta tarjeta.

*Tabla 2 Características técnicas de tarjeta FRDM K64.*

Marca	Freescale
Modelo	FRDM K64
MCU	MK64FN1M0VLL12 MCU
Frecuencia de operación	120 MHz
Memoria Flash	1 Mb
Memoria RAM	256 Kb
Consumo eléctrico	Bajo consumo

En la figura 8 se puede apreciar cómo se llevaron a cabo las pruebas de detección usando el sensor de radar.

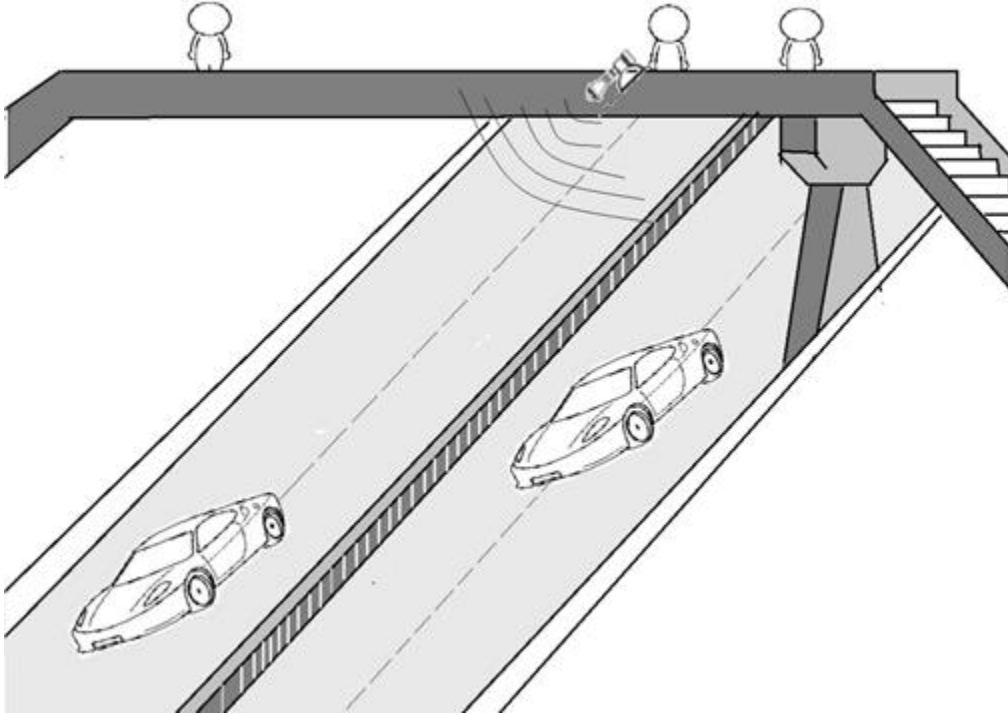


Figura 8. Pruebas realizadas en campo.

### 3.2. Adquisición de datos.

Durante las pruebas se hizo un registro de las mediciones realizadas. La señal del sensor fue enviada a un microcontrolador que, a su vez, envió hacia la PC el registro de las mediciones en tiempo real mediante comunicación serial. En la figura 9 se aprecia lo descrito anteriormente.



Figura 9. Diagrama de conexión para registro de datos.

A partir de las pruebas se pudo observar la respuesta del sensor con diferentes estados de tránsito vehicular. La figura 10 muestra una gráfica de la respuesta del sensor en un escenario de tránsito vehicular fluido. El tiempo de muestreo fue de 100 milisegundos. Como se observa, existen 3 franjas de colores diferentes que tienen una interpretación propia.

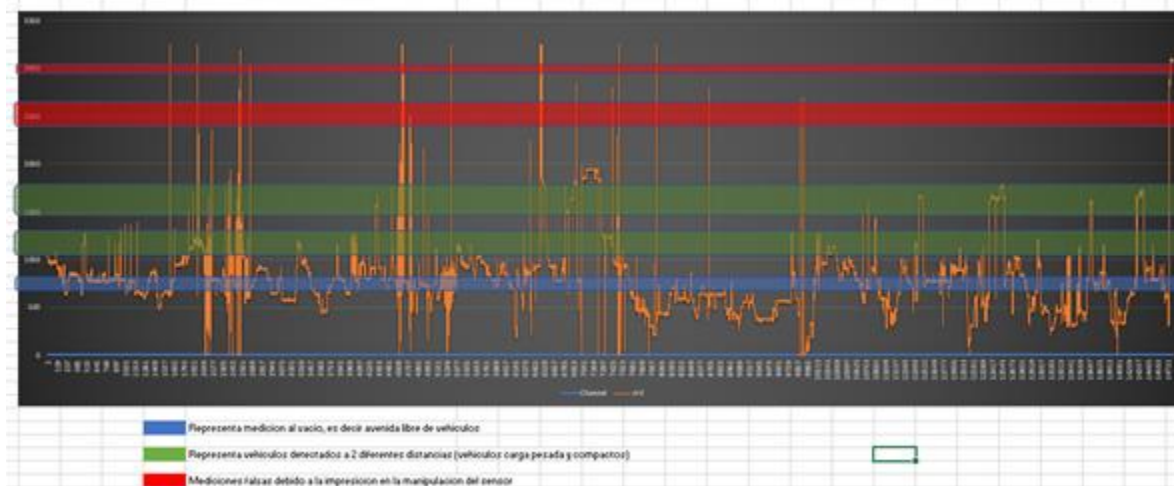


Figura 10. Respuesta del sensor de radar con tránsito fluido.

Mediante esta primera prueba fue posible caracterizar la respuesta del sensor. También dio información acerca de cómo implementar el algoritmo de detección de tránsito; este punto se especificará en este mismo capítulo, en el apartado Implementación para la medición.

### 3.3. Algoritmo de procesamiento de la señal.

En acuerdo con el asesor de tesis, se determinaron los siguientes criterios para la generación del algoritmo de detección:

- Cada minuto se actualizará el estado actual del tránsito vehicular.
- Se analizarán los últimos 5 minutos de registro del estado de tránsito.
- Se definieron 5 estados diferentes de estado de tránsito:
  - Sin tránsito
  - Tránsito ligero

- Tránsito fluido
- Tránsito pesado
- Tránsito detenido

Para la obtención de la señal del sensor se han utilizado los periféricos AFEC (ADC), DMA y el temporizador RTT. De igual manera, aunque el sistema no corre bajo ningún sistema operativo, si se ha implementado un scheduler multitarea “Round Robin” controlado por el temporizador del sistema “SysTick”. A continuación, se detalla el algoritmo usado para el cálculo de tránsito vehicular. Mas adelante, en las secciones 3.5,3.6 y 3.7 se detalla la configuración utilizada de cada uno de los recursos mencionados.

### **3.4. Implementación para la medición.**

La implementación de la medición incluye los siguientes elementos:

- Establecer los rangos de detección.
- Filtrado de señal obtenida.
- Análisis de detección de objetos cada segundo.
- Conteo de detección de objetos por minuto.
- Resultado final basado en el registro de los últimos 5 minutos de tránsito.
- Envío de resultados por periférico UART.

El sensor envía como dato final una señal que representa la distancia de separación entre el sensor y el objeto detectado. Como se pudo observar en la figura 10 (página 27) existe un rango (zonas verdes) en la señal del sensor que representa que un objeto ha sido detectado. Estos rangos de medición están establecidos con base a los siguientes parámetros: altura de instalación del sensor con respecto a nivel de piso y altura promedio de los vehículos con respecto al nivel de piso. Cualquier dato fuera de este rango no será considerado en el conteo de objetos detectados. Para este proyecto, se observa en la figura 10 que los niveles de mili voltaje que representa la detección de un objeto es el siguiente:



Tabla 3 Rangos de detección

Rango 1 mínimo	1100 mV
Rango 1 máximo	1450 mV
Rango 2 mínimo	1550 mV
Rango 2 máximo	1950 mV

Las mediciones contenidas entre el rango de medición 1 representan vehículos de mayor altura como lo pueden ser tráiler, autobuses, camiones de volteo y, en general, vehículos de carga pesada que usualmente son más altos. Debido a la mayor altura de este tipo de vehículos, la distancia entre estos vehículos y el sensor es menor y la respuesta del sensor será de un mili voltaje menor. Las mediciones contenidas entre el rango de medición 2 representan vehículos de menor tamaño como lo son los autos compactos, camionetas tipo sub y la mayoría de los vehículos familiares. Debido a que la altura de estos vehículos es menor, la distancia entre estos y el sensor será mayor, por lo que será correspondida por una señal de mili voltaje mayor.

Mediante una tarea cíclica de 100 mili segundos controlada por el scheduler del sistema, se filtra la señal almacenada en el arreglo principal que almacena los datos. A partir de que la señal enviada por el sensor ya ha sido filtrada por el mismo sensor para eliminar ruido, el filtro implementado es básico y consiste en el promedio de los datos obtenidos (figura 25).

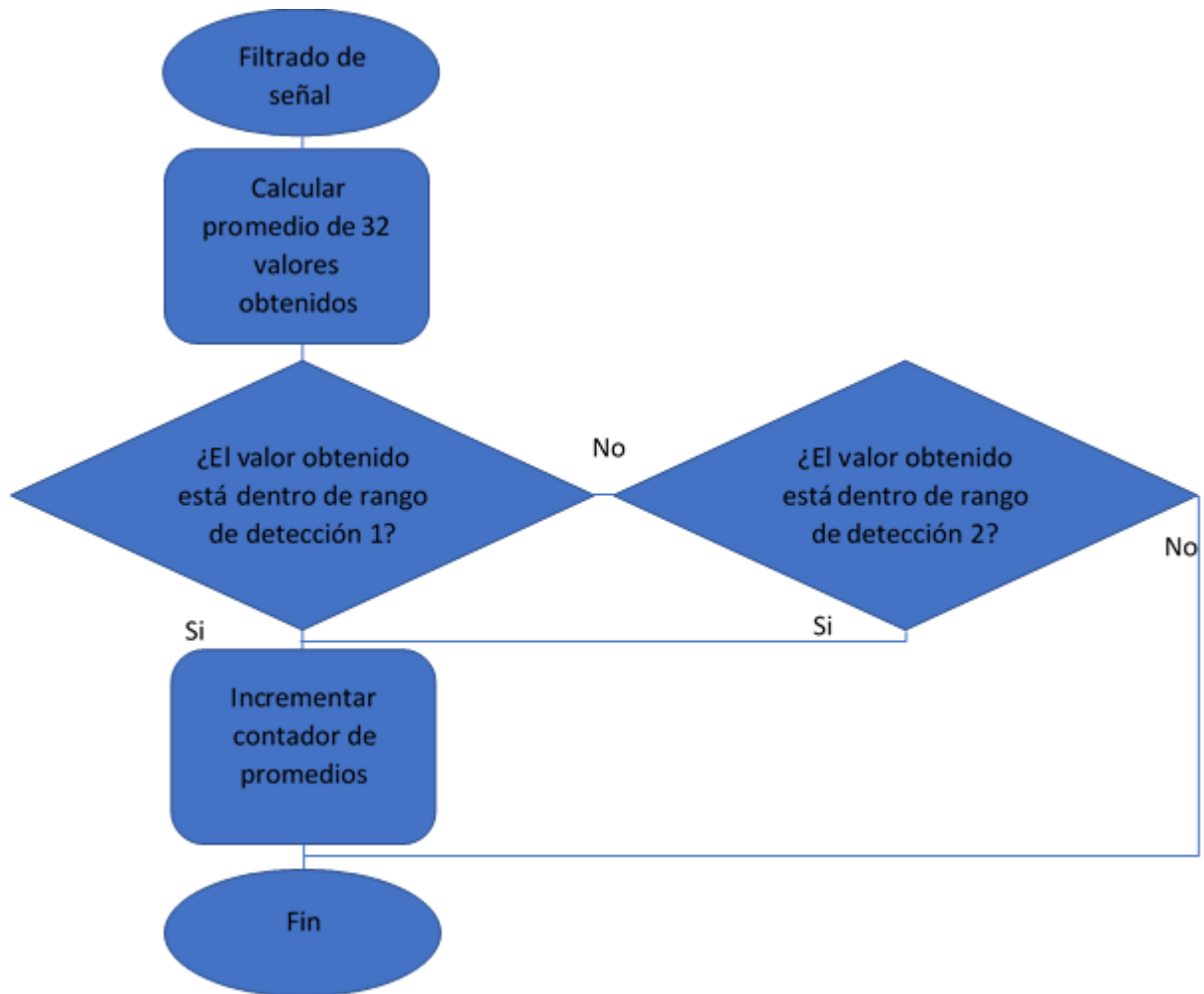


Figura 11 Filtrado de señal obtenida.

Cada segundo se analiza la señal filtrada para así validar si un nuevo objeto ha sido detectado. Se ha definido un umbral de detección, el cual sirve para determinar si la señal obtenida fue suficientemente constante para considerarse un objeto; si no, se considera como dato no válido. La figura 26 demuestra dicha implementación.

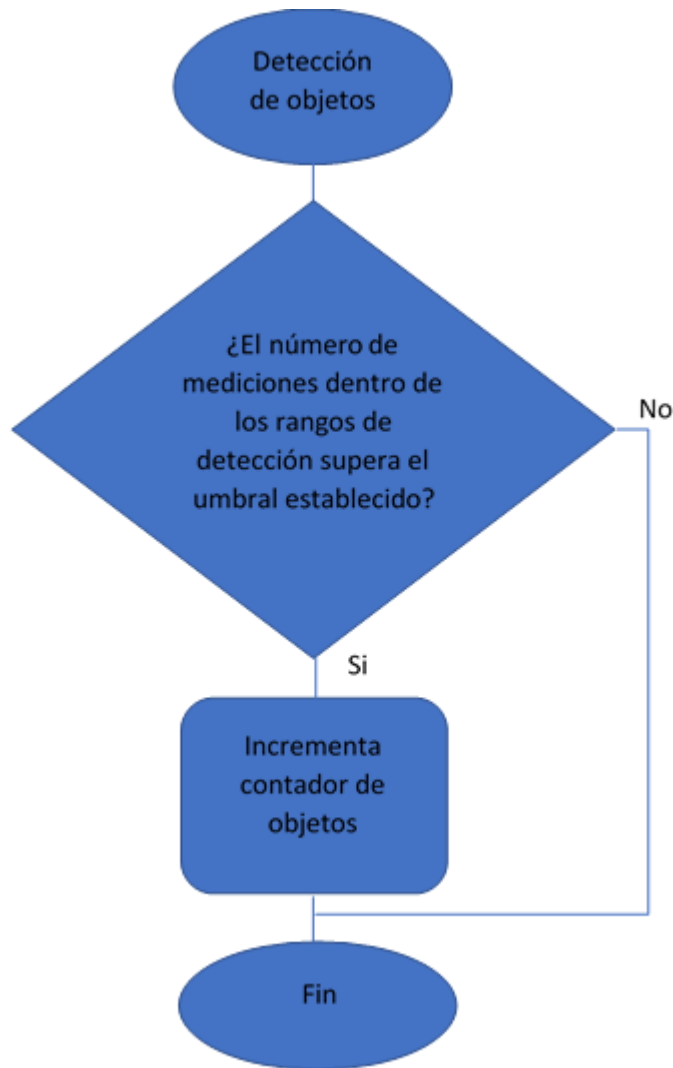
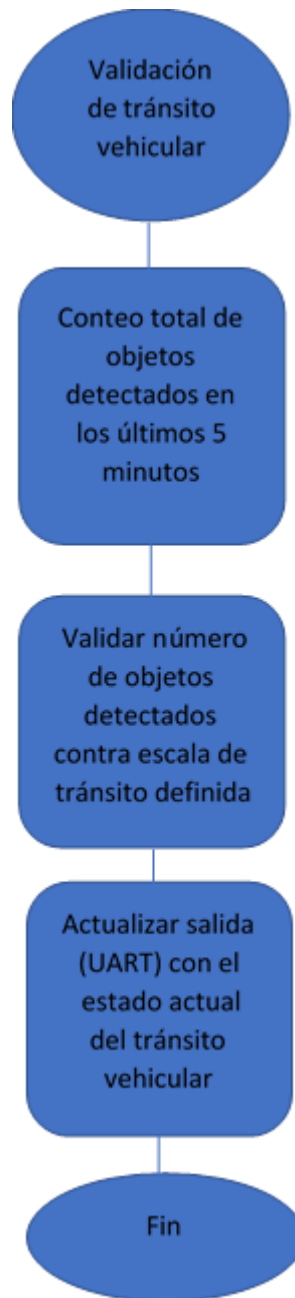


Figura 12 Detección de objeto.

A partir de los datos que contienen la información del estado del tránsito vehicular de los últimos 5 minutos, cada minuto se hace una sumatoria de los últimos 5 minutos. Estos resultados se comparan contra una escala establecida, la cual indica directamente el estado del tránsito actual. Como la validación de objetos detectados se hace cada segundo, como máximo se logrará un conteo de 300 objetos detectados en 5 minutos. Cabe aclarar que un objeto que se detecta cada segundo no necesariamente representa un vehículo completo, más bien indica que ha sido detectado. En la figura 27 se observa dicha implementación.

Una vez determinado el estado actual del tránsito es enviado mediante el periférico UART.



*Figura 13 Validación del estado de tránsito vehicular.*

### 3.5. AFEC.

El módulo AFEC (Analog Front-End Controller) es básicamente el ADC (Analog to Digital Converter) de nuestra tarjeta SAM V71. Es un convertidor analógico digital con una resolución de 12 bits. Como se observa en la figura 22, la configuración de este módulo consiste en los siguientes parámetros:

- 12 bits de resolución.
- Comienzo de conversión iniciada únicamente por software.
- Inhabilitación del Tag del canal utilizado.
- La señal no es promediada por hardware.



Figura 14 Inicialización del módulo AFEC.

El control del inicio de conversión analógica a digital está bajo control del scheduler del sistema mediante la ejecución de una tarea cíclica. Se asignó una tarea cíclica de 2 mili segundos para el inicio de conversión y, por consecuencia, la adquisición de datos. Los datos obtenidos se almacenarán en un arreglo para su posterior procesamiento.

### 3.6. DMA.

El periférico DMA (Direct Memory Access) es un periférico utilizado para transferir datos, ya sea de periférico a memoria o de memoria a periférico. En este caso se configuró el uso del DMA para transferir los datos obtenidos desde el periférico del ADC hacia nuestro arreglo de datos en memoria RAM. Esta configuración nos permite quitarle carga de procesamiento al microprocesador de nuestra ECU, ya que la transferencia de datos se hace mediante hardware dedicado.

Algunas de las configuraciones necesarias para el uso de este periférico son las siguientes:

- Dirección de periférico a relacionar con el DMA, en este caso el periférico asociado es el AFEC.
- Dirección de memoria RAM donde se alojarán los datos transferidos.
- Tamaño en memoria RAM del arreglo donde se alojarán los datos.

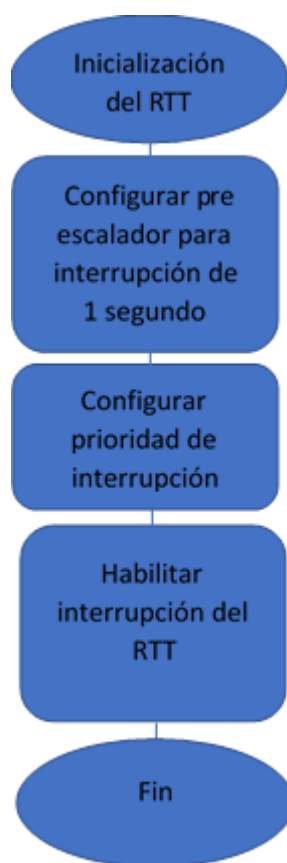


Figura 15 Inicialización del módulo DMA.

El tamaño del arreglo donde se almacenan los datos transferidos por el DMA es de 32 bytes.

### 3.7. RTT.

El temporizador RTT (Real-time Timer) es usado como un generador de base de tiempo de 1 segundo, y lo utiliza directamente el algoritmo de detección de tránsito vehicular para generar los cálculos con una periodicidad confiable, ya que se configura una interrupción para generar eventos cíclicos. Dichos eventos están bajo el control del scheduler, el cual activará tareas específicas para atender estos eventos. En la figura 24 se observan las configuraciones de inicialización del módulo RTT.



*Figura 16 Configuración del temporizador RTT.*

### 3.8. Descripción de la base utilizada

Aquí describiremos el algoritmo del movimiento de los motores que, como se mencionó en el marco teórico, es el encargado del escaneo de los carriles para que el sensor comience a detectar los vehículos.

Primero describiremos el algoritmo y después analizaremos el código encargado de ejecutarlo.

El algoritmo que se seguirá para el manejo de la base esta dado por 5 pasos:

1. La base permanecerá inmóvil durante los primeros 20 segundos una vez que inicie el programa, esto debido a que desde el instante cero empieza el escaneo de los vehículos.
2. Cuando pasan los primeros 20 segundos, la base comienza a moverse en el eje Z para escanear a lo ancho del carril.
3. Una vez terminado de escanear a lo ancho del carril, la base comenzará a moverse en el eje X hacia la izquierda para poder escanear el siguiente carril. Cabe destacar que mientras esté haciendo estos movimientos, el sensor está procesando en todo instante los datos que está tomando del sensor.
4. El sensor se moverá en el eje x durante 3 segundos para cambiar de carril y después escaneará en el eje z a lo largo del nuevo carril.
5. Ese proceso se repetirá para los tres carriles, y una vez terminado el último carril volverá a su posición original remitiéndose de nuevo el algoritmo en el punto uno

Para poder hacer este algoritmo es indispensable saber cómo debe de ir conectado el sensor, junto con la tarjeta procesadora de los datos y la base móvil.

En la Figura 8 podemos ver la versión final de las conexiones necesarias para que el algoritmo de movimiento de la base móvil sea posible. Tiene dos ejes, el eje uno es el representado para el movimiento del eje x y el eje dos es el representante del movimiento de la base para el eje z.



La figura 8 muestra la tarjeta FRDM k64 montada sobre la base. Esta tarjeta está simulando la señal de sensor, que fueron los datos recolectados en durante las pruebas en campo. Podemos percibir que la FRDM k64 se compone de dos salidas (cables color amarillo y negro) que representan a la señal de salida de los datos y la GND de la tarjeta. Esta señal de la que obtenemos los datos representativos del sensor se obtiene a través del pin DAC0\_OUT y el GND del pin GND-12.

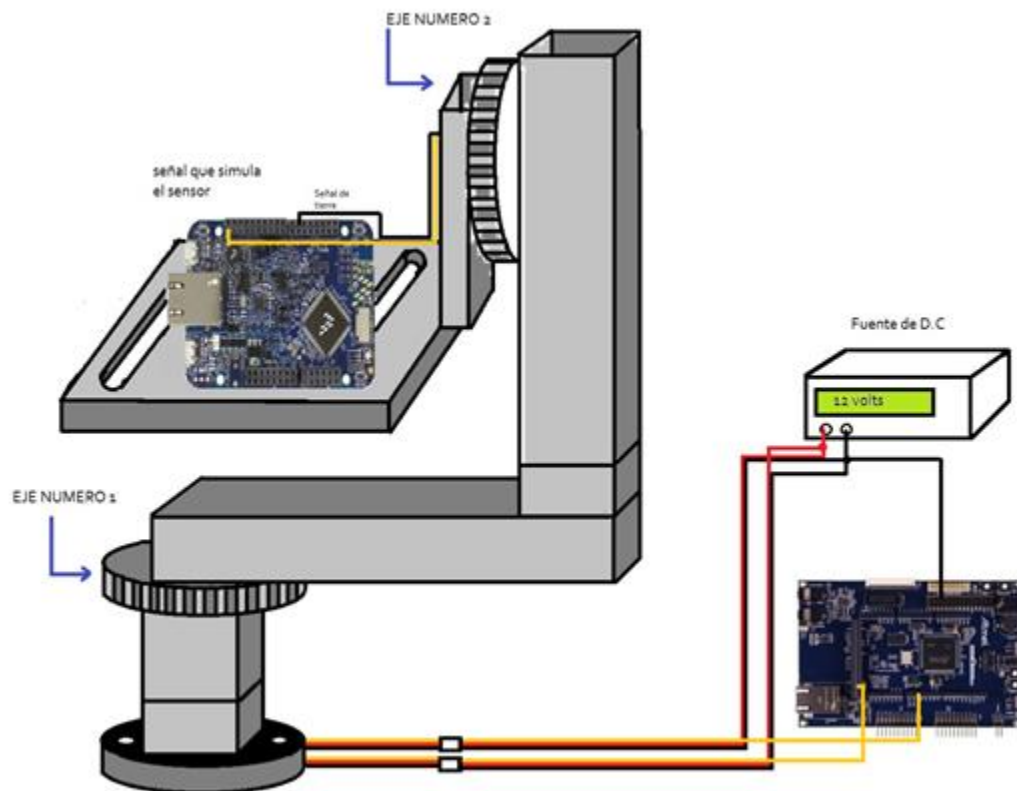


Figura 17 Diagrama de conexión de base pan and tilt junto con sensor y tarjeta procesadora de datos.

Las señales de la tarjeta FRDM k64 (montada sobre la base) irán conectadas a la tarjeta SAMV71 que es la encargada de procesar los datos del sensor.

En la Figura 8 se puede observar también dos conexiones que están sobre la base pan and tilt, y son las correspondientes del servomotor (crf. p. 5 del Marco Teórico) que permiten los

movimientos de los ejes uno y dos. Cada motor contiene tres señales que son señal de sensor (cable amarillo), VCC (cable rojo), GND (cable negro). VCC y GND están conectados a una fuente de D.C (direct current) y la señal de los motores están conectados a la tarjeta SAMV71 a los pines PA00 y PB00.

El algoritmo del movimiento de los motores descrito anteriormente se describe a nivel de código en la figura 9 como una función denominada como `start_moving`, que es un evento generado en un scheduler que se ejecuta inicialmente una vez pasados los primeros 20 segundos como se puede observar a continuación:

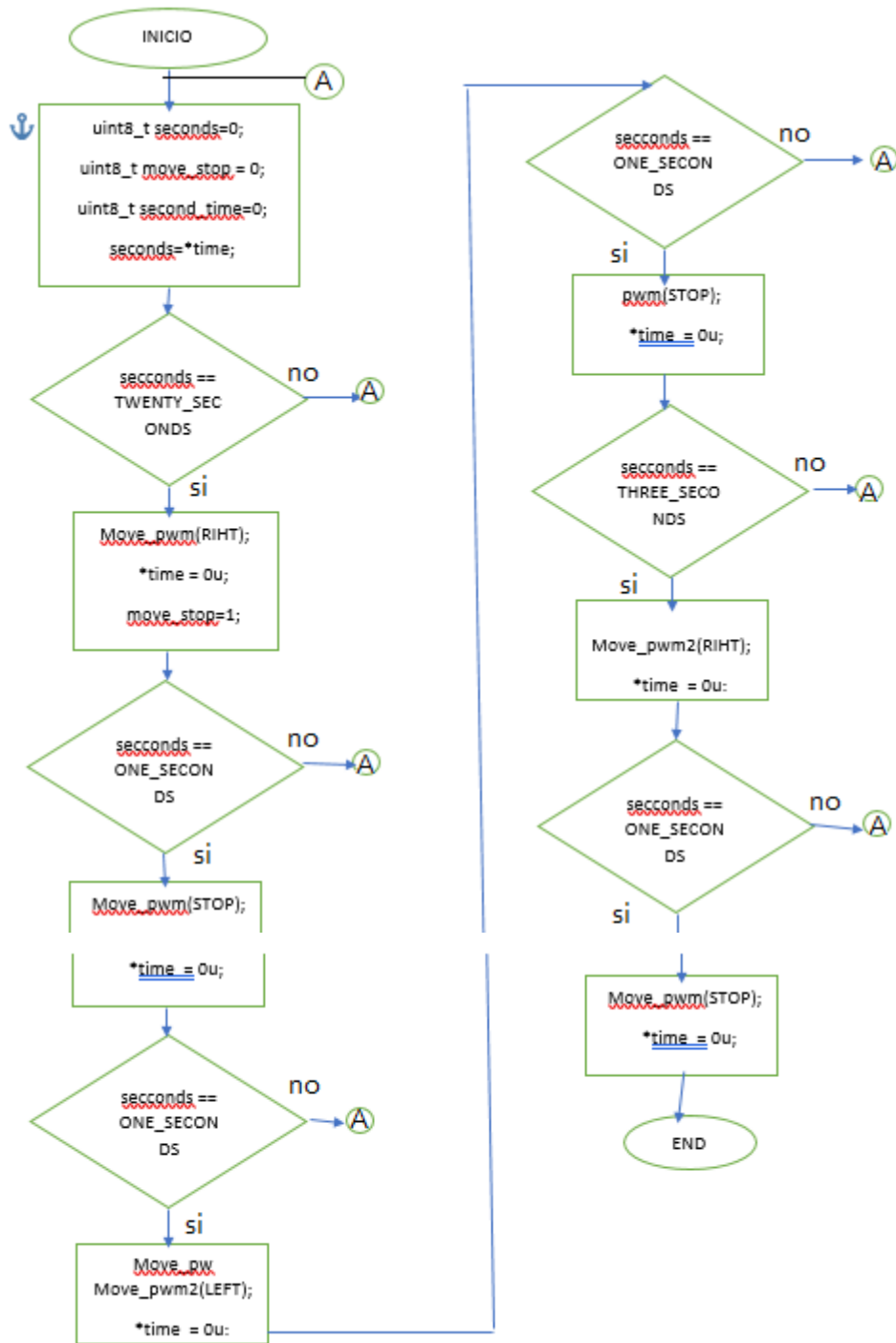


Figura 18 Código de movimiento de motor en una base de tiempo.

La función `move_pwm()` describe el algoritmo del movimiento de los sensores, el cual usa una base de tiempo de 1 segundo y al cumplirse los instantes de tiempo se manda a llamar la función `Move_pwm2 (STOP)`; o `Move_pwm (STOP)`; que son la señales que maneja el movimiento de los motores como se muestra en la figura 10. Estas pasan un parámetro llamado `stop`, `left` o `right` que simboliza la dirección a la cual el motor se moverá.

La función `pwm()`; está descrita en la Figura 10:

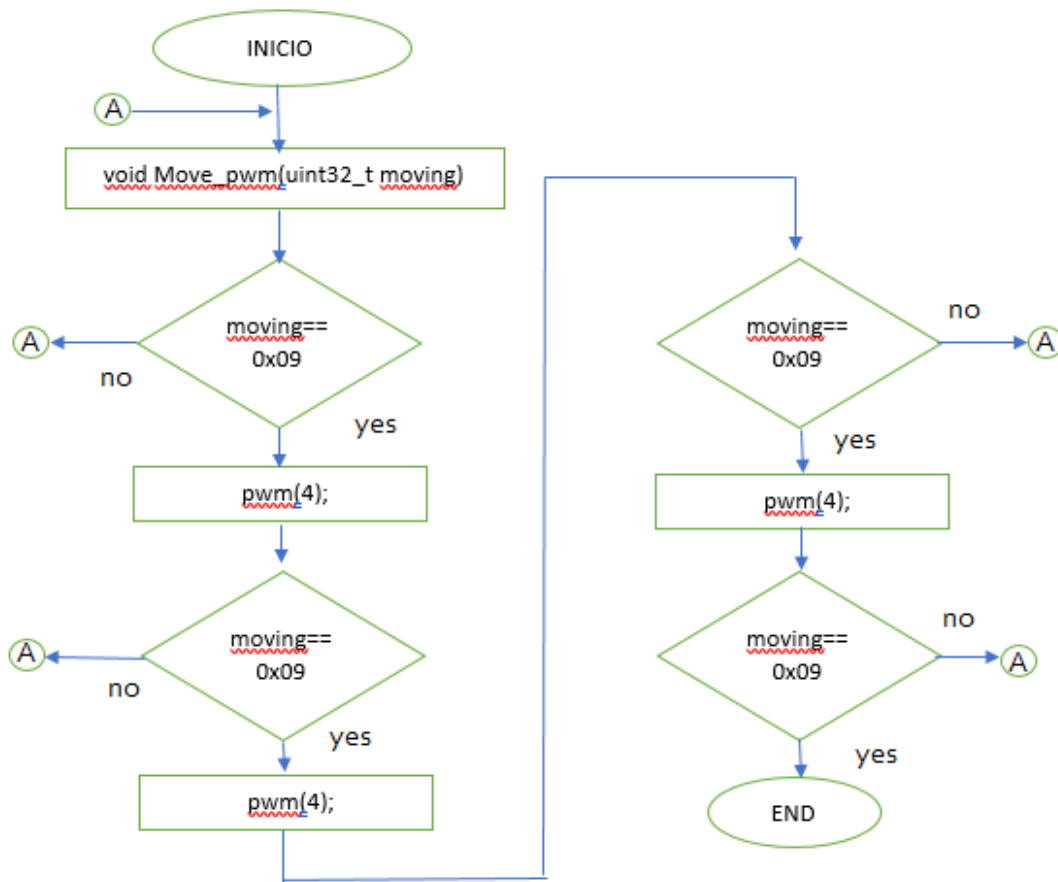


Figura 19 Código de movimiento de motor en base a movimiento de los ejes.

La función `pwm()` es mandada a llamar de acuerdo al parámetro heredado de `move_pm` que indica la dirección de movimiento a la cual debe moverse.

La función `pwm()` está descrita mediante el siguiente código, representado en la figura 11:

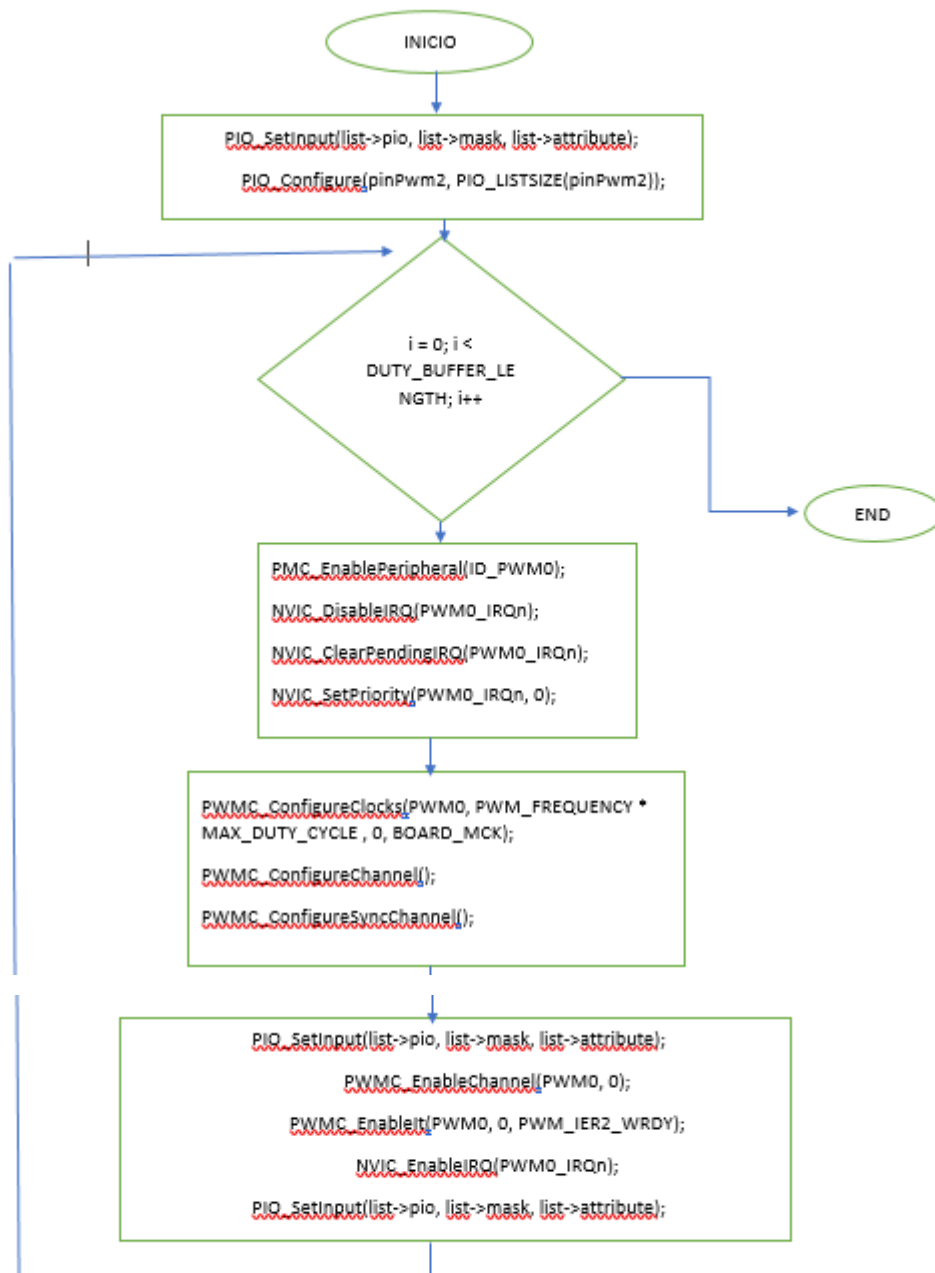


Figura 20 Código de movimiento de motor en base a movimiento de los ejes.

La función PWMC\_configure hace la configuración de los registros correspondientes del PWM para poder mover los motores.

El algoritmo con base a código se describe en tres partes:

- Movimiento con base a una escala de tiempo.
- Movimiento con base a la dirección en la que se desea mover.
- Movimiento con base a la configuración del PWM para generar la señal que permitirá el movimiento de los motores.

El sensor que se encarga del escaneo para fines de demostración será reemplazado por la tarjeta de desarrollo FRDM k64 (ver figura 17 página 37), que generará en base a tiempo transcurrido señales que representan la ausencia y presencia de tráfico hasta el punto de detección de un embotellamiento por la tarjeta SAMV71.

Cada minuto generará un estado de detección de vehículos, de acuerdo con los siguientes colores:

Azul -> NO\_TRAFFIC

Verde-> LIGHT\_TRAFFIC

Amarillo -> SMOOTH\_TRAFFIC

Naranja -> HEAVY\_TRANSIT

Rojo-> STOP\_TRANSIT

Cada que la señal generada simula un nuevo estado de tráfico los colores del led cambiarán.

### **3.9. Funcionalidad realizada en la PC**

Una de las características de nuestro sensor de tránsito es la capacidad para poder compartir los datos estadísticos con una PC mediante el puerto serial: después del análisis de los carriles y del procesamiento de la información (que se lleva a cabo cada minuto), el sensor actualiza el buffer de mensajes. Este buffer contiene el resultado del análisis del tránsito de los carriles.

Para la comunicación serial se utilizó el módulo USAR1 (*Universal asynchronous receiver/transmitter*), apoyándonos en la función *DBG\_Configure* del módulo *fbg\_console.c*

creamos una función para la inicialización del USART1 (ver figura 21) en donde el *baudrate* fue definido a 115200 y el *masterClock* a 150000000.

```
Usart *pUsart = CONSOLE_Usart;
// Disable the MATRIX registers write protection
MATRIX->MATRIX_WPMR = MATRIX_WPMR_WPKEY_PASSWD;
MATRIX->CCFG_SYSIO |= CCFG_SYSIO_SYSIO4;

PIO_Configure( pPins, PIO_LISTSIZE( pPins ) );

// Reset & disable receiver and transmitter, disable interrupts
pUsart->US_CR = US_CR_RSTRX | US_CR_RSTTX | US_CR_RSTSTA;
pUsart->US_IDR = 0xFFFFFFFF;
PMC_EnablePeripheral(CONSOLE_ID);
pUsart->US_BRGR = (masterClock / baudrate) / 16;

// Configure mode register
pUsart->US_MR
    = (US_MR_USART_MODE_NORMAL | US_MR_PAR_NO | US_MR_USCLKS_MCK
        | US_MR_CHRL_8_BIT);

// Enable receiver and transmitter
pUsart->US_CR = US_CR_RXEN | US_CR_TXEN;
```

Figura 21 Configuración USART1.

Dentro de la función de inicialización habilitamos los eventos del USART1, mediante la función *NVIC\_EnableIRQ* (figura 15).

```
extern void DBG_init(void)
{
    Usart *pUsart= CONSOLE_Usart;

    if ( !_ucIsConsoleInitialized )
    {
        DBG_Configure(CONSOLE_BAUDRATE, BOARD_MCK);
    }

    NVIC_EnableIRQ(USART1_IRQn);
}
```

Figura 22 Configuración USART1.

Para hacer las primeras pruebas de comunicación serial estuvimos utilizando la herramienta PuTTY de Simon Tatham (figura 16), que nos ayudó a mostrar los mensajes que se envían por el puerto serial.

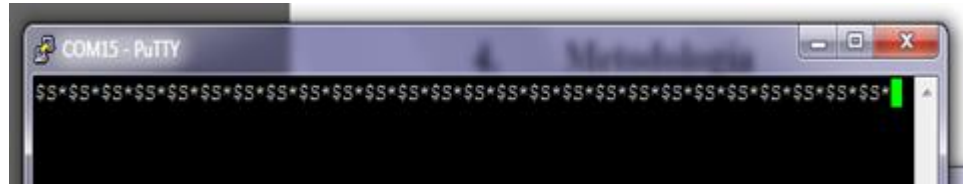


Figura 23 Terminal PuTTY – Ejemplo de mensaje.

Los parámetros de configuración utilizados para la comunicación serial se muestran en la figura 17:

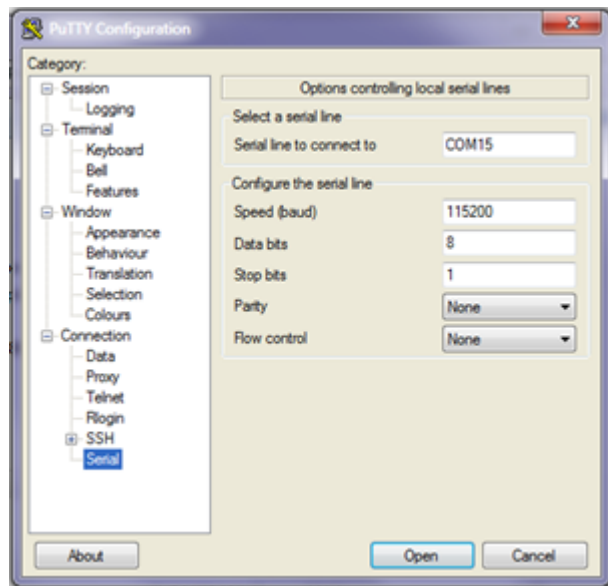


Figura 24 Terminal PuTTY – configuración de parámetros.

La comunicación entre el sensor y la PC se lleva a cabo cada 100 milisegundos, mediante una tarea del Schedule controlamos cuando ejecutar esta acción. Primero se envía un mensaje desde el sensor hacia la PC (\$\$\*), en donde el carácter “\$” representa el inicio de cadena, “S” indicará que iniciará una transferencia de información y “\*” es el fin de la cadena.

Después de enviar el mensaje se activará un contador que durará 80 milisegundos; durante este periodo de tiempo el sensor activará la RX (recepción de datos) del USART1, esperando una



respuesta de la PC. Si la PC no responde, se cierra la recepción, la función termina y se ejecutará nuevamente hasta que pasen los 100 milisegundos de la tarea actual.

Si se estableció una comunicación entre la PC y el sensor, y la GUI (del inglés *graphical user interface*) está activa, esta analiza los mensajes enviados por el sensor (figura 18); en el caso de detectar el mensaje “\$\$\*”, la GUI enviará un mensaje de “acknowledge” con el siguiente formato “\$A\*” y automáticamente cambiará su estado, para esperar los datos estadísticos provenientes del sensor.

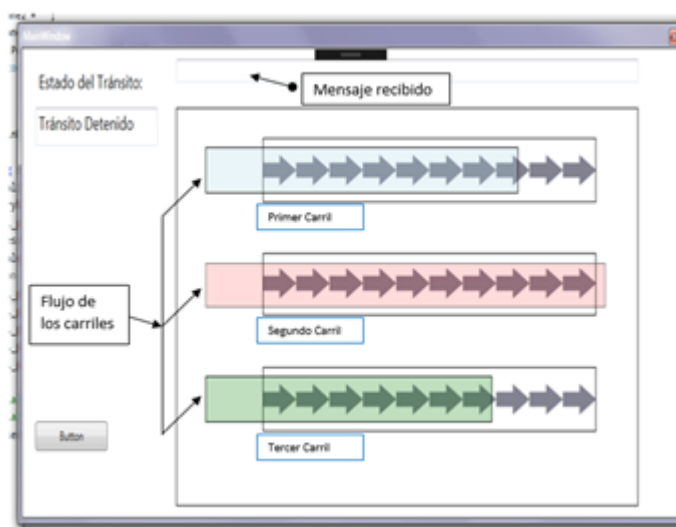


Figura 25 Interface gráfica – GUI.

Cuando el sensor reconoce el mensaje de “acknowledge” se cierra la recepción de datos y cambia su modo de trabajo; el sensor tomará el buffer de mensaje y los enviará por el puerto serial con el siguiente formato: “\$ABC\*”, en donde “A” representa el estado del primer carril, “B” el estado del segundo y “C” es el estado del último.

La GUI, después de recibir el mensaje con los estados de los carriles, valida que el mensaje tenga una estructura correcta, separa los caracteres para obtener los valores de cada carril y los representara en una animación. Esta muestra el flujo del carril con un color y velocidad específico, dependiendo de la cantidad de tránsito del carril.

Como se muestra en la tabla 4 la GUI representa los 3 carriles, el flujo estará representado por un recuadro animado de color:

*Tabla 4 Significado de los estados del sensor y su tiempo de animación.*

Color:	Significado	Tiempo - Animación (ms)
Azul	Sin Tránsito	800
Verde	Transito Ligero	2000
Amarillo	Tránsito Fluido	5000
Naranja	Tránsito Pesado	8000
Rojo	Tránsito Detenido	500 (parpadeo)

Cuando la GUI percibe un cambio de estado de alguno de los carriles, se detiene la animación del carril específico, valida el valor obtenido y genera los parámetros adecuados según el tipo de tránsito. Se actualizan los parámetros de los efectos y se ejecuta la animación.

La GUI fue desarrollada utilizando WPF (*Windows Presentation Foundation*) y el lenguaje de programación C#, que nos permitió crear las animaciones en un solo día y con el soporte de un lenguaje de programación robusto.

La comunicación serial fue realizada usando la librería System.IO.Ports, así como los mismos parámetros utilizados en la herramienta PuTTY y habilitando el parámetro DTR.

Descartamos utilizar el handler de recepción de datos del puerto serial por problemas de comunicación y sincronización. En cambio, utilizamos un timer que se ejecuta cada 30 milisegundos. Al leer el buffer de recepción encontramos todos los mensajes no leídos enviados por el sensor, esto nos permite buscar los mensajes dentro del buffer sin perder datos de transmisión.

El “mensaje recibido” muestra todos los mensajes reconocidos y procesados por la GUI. Este campo se actualiza justo después de recibir el mensaje. Mientras que en el recuadro de “estado del tránsito” se describe el comportamiento del tránsito en los últimos 5 minutos (en este caso del carril uno).

## **4. RESULTADOS.**

### **4.1. Descripción del tipo de pruebas que se realizaron.**

Por limitaciones de tiempo, algunas de las características definidas al principio del desarrollo del proyecto, fueron omitidas. La idea principal del proyecto era poder analizar un máximo de 3 carriles (cada carril se analizaría a lo largo). En este prototipo solo analizamos un carril. Para demostrar la capacidad del proyecto se agregó una secuencia, con la que se muestra cómo es que se controla y se mueve la base para analizar dos carriles a la vez.

Las pruebas del análisis del carril resultaron satisfactorias. Se planteó la idea de comunicar el sensor vía ethernet con un servidor web, y así mostrar los análisis estadísticos en una herramienta web. Esta característica no fue incluida en este prototipo debido al tiempo de entrega, en cambio fue posible terminar la comunicación serial y crear una GUI en la PC para interactuar con el sensor. Se presentaron algunos problemas de sincronización con la GUI y el sensor, debido a los tiempos de procesamiento de la PC, pero fueron resueltos en tiempo.

Para las primeras pruebas con el microcontrolador, pudimos contar físicamente con el sensor; en las dos primeras se identificó el comportamiento de la señal, analizando algunos objetos en movimiento, como personas o vehículos. Las siguientes 2 pruebas las hicimos enfocadas en el proyecto, analizando el flujo vehicular. El sensor estuvo sobre un puente peatonal detectando el flujo vehicular, enviando estos datos hacia el microcontrolador.

Para las pruebas finales no se contó con el sensor, en cambio, utilizando otro microcontrolador pudimos simular la señal del sensor, apoyándonos en las pruebas hechas anteriormente.

## 4.2. Pruebas de comunicación.

Las pruebas de comunicación fueron exitosas, aunque solo se tuvo un problema con la sincronización entre la tarjeta y la PC. El driver para controlar el puerto serial desde la PC se comportaba de manera diferente; mientras que en el microcontrolador el evento de recepción de datos se generaba cada que llegaba un carácter, en la PC el proceso era un poco más lento y se llegaban a tener hasta dos caracteres en el buffer. No perdíamos información, pero para analizar el mensaje completo tuvimos que esperar más tiempo. La comunicación del sensor está controlada con el *scheduler*, mediante una tarea que se ejecuta cada 100 milisegundos, activamos el envío de datos. Fue necesario cambiar los tiempos de ejecución de 20 milisegundos hasta 100. En cambio, en la GUI utilizamos un *timer* de 30 milisegundos para leer los mensajes en el *buffer*, en el cual buscamos los mensajes enviados en la cadena que nos retorna el buffer.

La primera prueba de recepción consiste en recibir el mensaje de inicio de comunicación, como vemos en la figura 28, la GUI muestra en el recuadro superior el mensaje identificado.

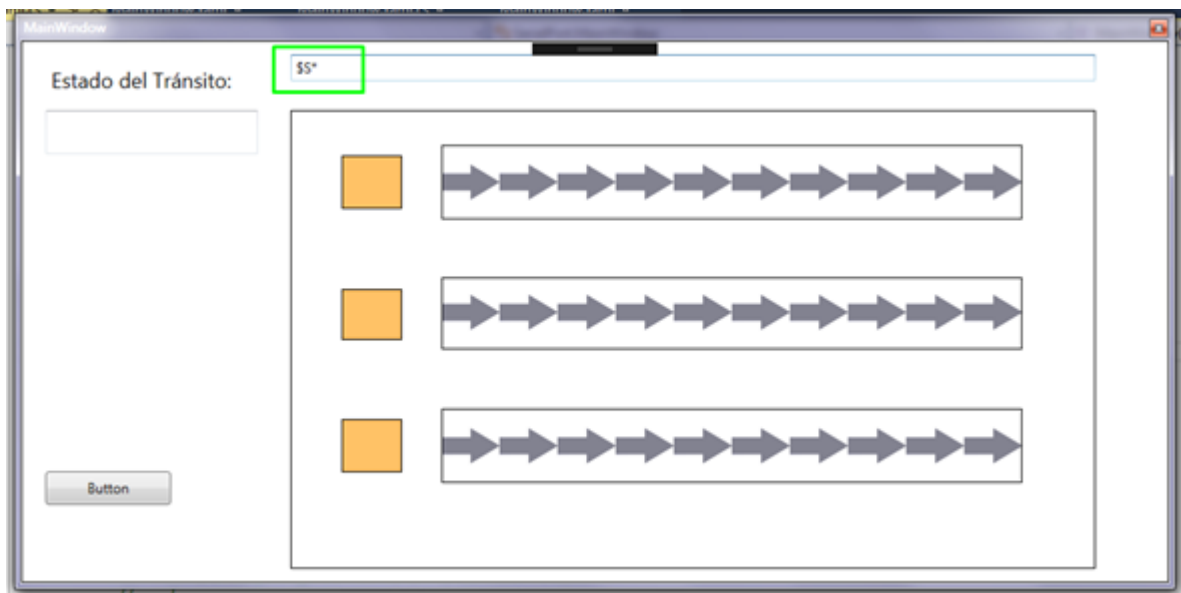


Figura 26 Recepción mensaje de inicio de comunicación.

Justo después de identificar el mensaje, se manda un mensaje de confirmación (\$A\*); el siguiente mensaje a enviar del sensor será el estado de los carriles. Como se comentó anteriormente, los tres caracteres después del carácter de inicio de cadena (\$) representan la

interpretación del tránsito de cada carril: el primer carácter representa el carril número uno; el segundo carácter es para el carril número dos y; el último, representa el carril número tres.

Por cuestiones de tiempo el sensor solo analiza un carril y envía el mismo valor para los tres carriles. Sabiendo esto podemos definir que existen 5 mensajes diferentes para la representación del tránsito, como se aprecia en la tabla 5:

*Tabla 5 Mensajes y animaciones en la GUI.*

Mensaje del sensor	Estado del tránsito	Color Recuadro	Animación
\$AAA*	Sin Tránsito	Azul Claro	Se expande el rectángulo sobre el carril
\$BBB*	Tránsito Ligero	Verde.	Se expande el rectángulo sobre el carril
\$CCC*	Tránsito Fluido	Amarillo.	Se expande el rectángulo sobre el carril
\$DDD*	Tránsito Pesado	Naranja.	Se expande el rectángulo sobre el carril
\$EEE*	Tránsito Detenido	Rojo.	El rectángulo sobre el carril parpadea

Según el mensaje recibido, la GUI mostrará un mensaje diferente para cada estado del tránsito, así como los colores y los tiempos de las animaciones de los recuadros variarán. En las figuras 29, 30, 31, 32 y 33 podemos ver los resultados de las pruebas de estos mensajes.

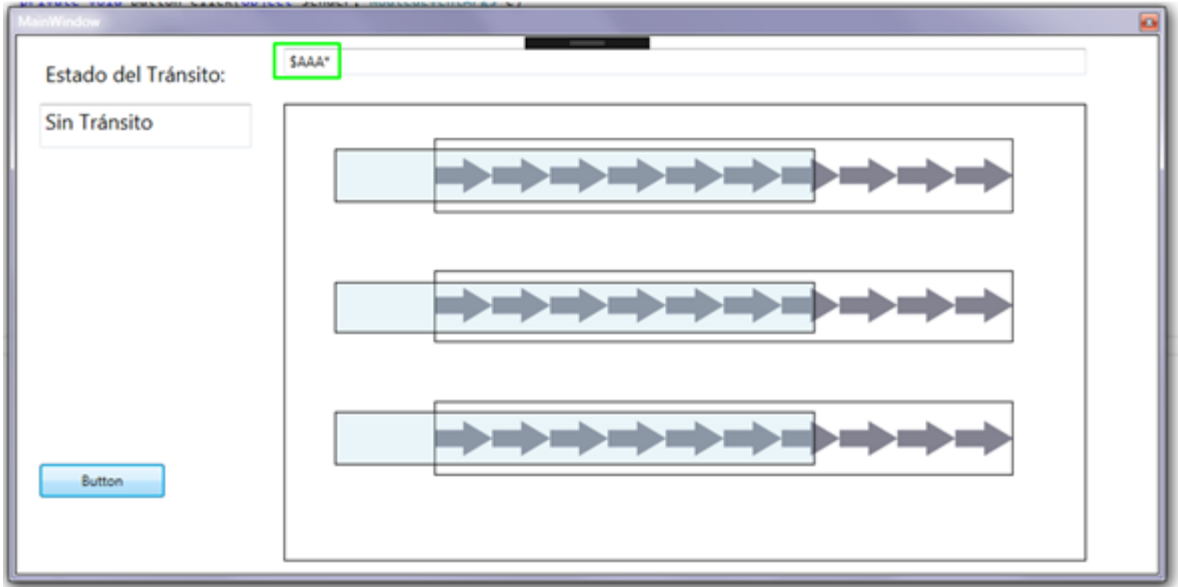


Figura 27 Respuesta de la GUI del Mensaje \$AAA\*

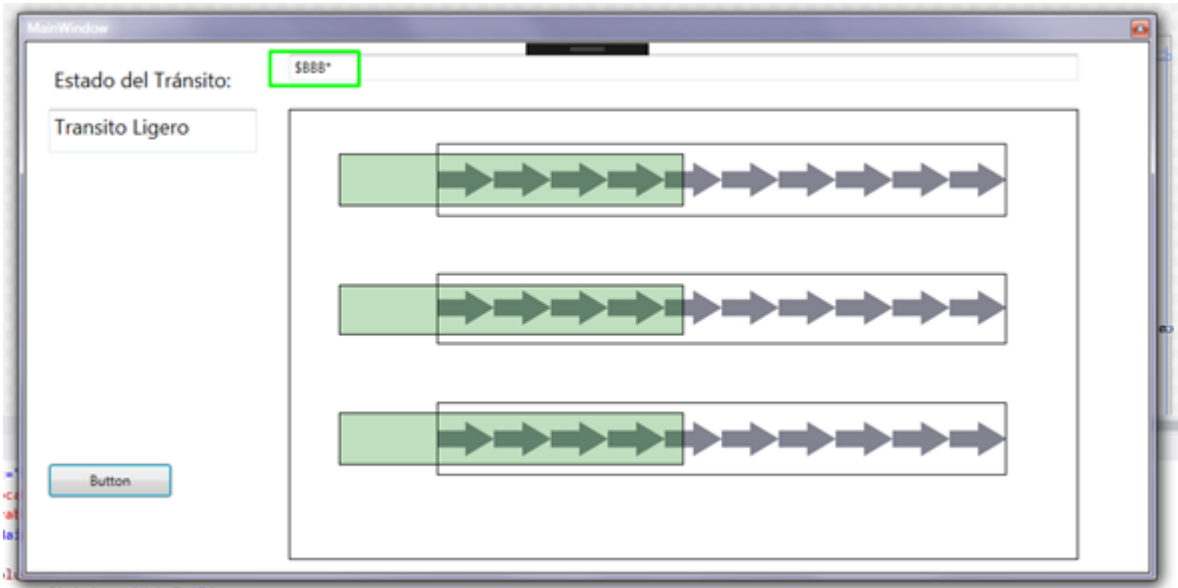


Figura 28 Respuesta de la GUI del Mensaje \$BBB\*

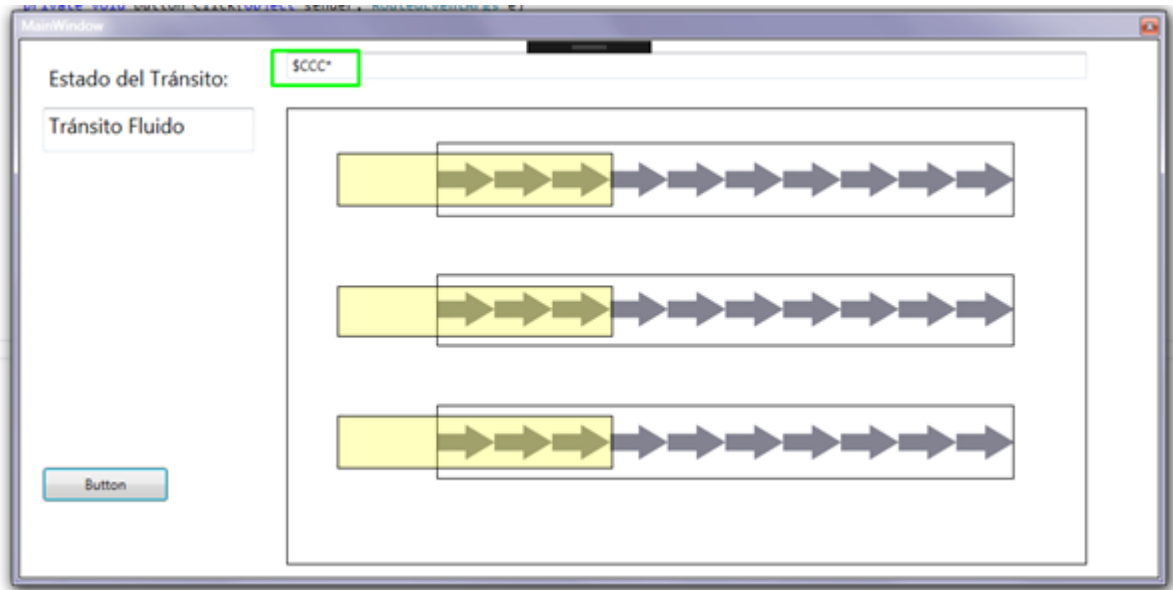


Figura 29 Respuesta de la GUI del Mensaje \$CCC\*

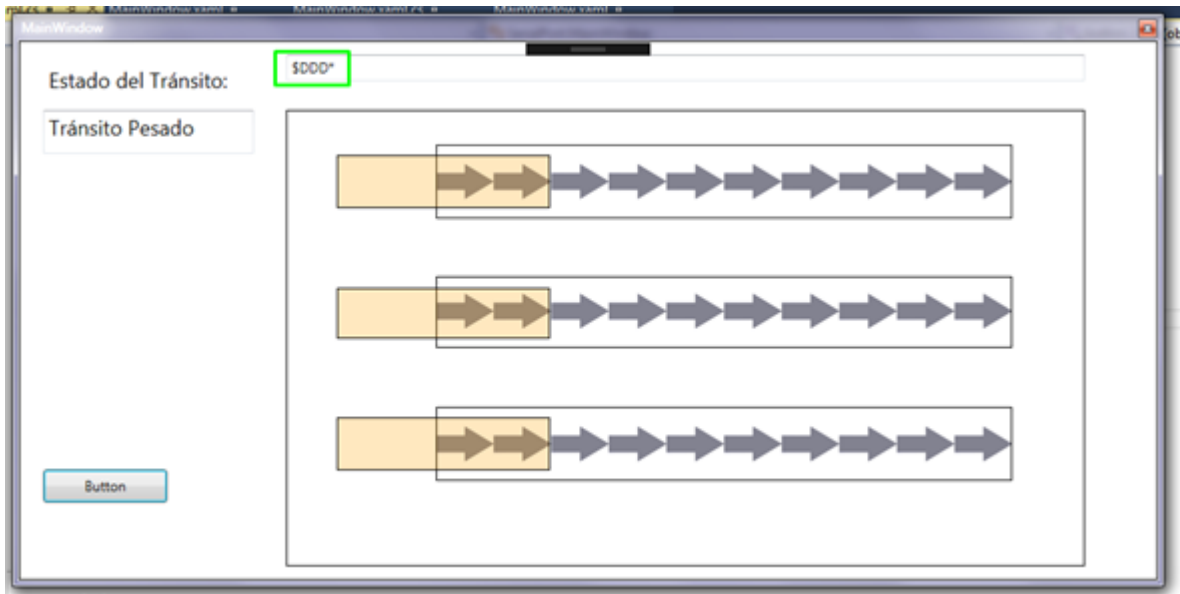


Figura 30 Respuesta de la GUI del Mensaje \$DDD\*

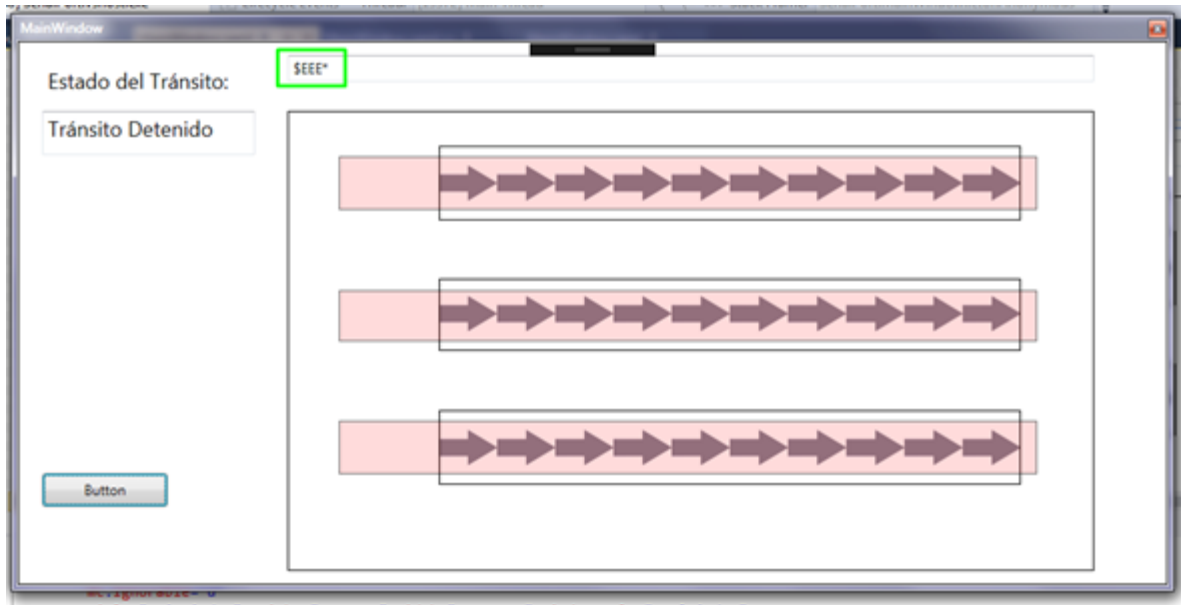


Figura 31 Respuesta de la GUI del Mensaje \$EEE\*

Las animaciones de los recuadros son diferentes entre sí, manteniendo cierta similitud entre los primeros 4 estados, al solo variar el color del recuadro y el tiempo de la animación, pero el último estado “Tránsito detenido” cambia por completo, pues este mantiene extendido un recuadro de color rojo, el cual parpadea cada 500 milisegundos. En las pruebas, las animaciones respondieron de manera correcta, ejecutando los efectos correspondientes según el mensaje enviado por el sensor. Después de la utilización de un *timer* para controlar los eventos de recepción, el procesamiento del mensaje y el control de las animaciones no afectaron la comunicación entre el sensor y la GUI, manteniendo una comunicación fluida, la cual estuvimos validando mediante los mensajes recibidos desde la PC.

### 4.3. Ventajas encontradas en comparación a proyectos anteriores.

Los resultados del proyecto son efectivos y con algunas ventajas sobre los otros sistemas mencionados, como es el “Sistema inteligente de control de semáforo en función de los flujos de tráfico en tiempo real”. Zhijun Li, Chunxiao Li, Yanan Zhang and Xuelong Hu resolvieron el problema de tránsito al utilizar un sistema de detección ultrasónico ubicado en las intersecciones principales para obtener la cantidad de flujo de tránsito, en el que el transmisor envía una señal



ultrasónica a una frecuencia de 40 Khz con una longitud de onda de 6 mm hacia el tránsito que, al ser reflejada y luego recibida por un receptor transductor con efecto piezoeléctrico, genera una señal de voltaje débil en el rango de los mV que después es amplificada. El sistema detecta el tiempo en que tarda en responder la señal y lo compara con el tiempo necesario al recorrer la distancia de prueba para obtener la cantidad de tránsito [4]. En contraste, nuestro sistema de detección vehicular sobresale de este conforme a la cantidad de sensores utilizados, ya que con la base pan and tilt tiene la funcionalidad de escanear todos los carriles en el mismo instante de tiempo, sin mencionar que la función del sensor hace el escaneo de la misma manera que el de este prototipo.

La investigación llamada “A novel signal processing technique for vehicle detection radar” de los autores Sang Jin Park, Tae Yong Kim, Sung Min Kang and Kyung Heon Koo [8] demuestra el efectivo uso de los radares para la medición de velocidad y clasificación por tamaño de vehículos en múltiples carriles. Nuestro proyecto no tiene la capacidad de clasificar los vehículos conforme a su tamaño, pero los datos se pueden llegar a interpretar para poder detectar vehículos grandes como son los tráileres, ya que estos generan una señal constante durante un periodo de tiempo prolongado (esto debido a lo largo del tráiler).

De acuerdo al artículo "A Low-cost Vehicle Detection and Classification System based on Unmodulated Continuous-wave Radar" de los autores J.Fang, H. Meng y H .Zang, la detección de tránsito vehicular basado en radares que trabajan en la banda K de frecuencia, proporcionan un prometedor rendimiento y precisión de acuerdo a sus resultados experimentales. También se hace la referencia a que dicho tipo de sensores tienen la ventaja de no ser vulnerables al clima, lo cual los hace ideales para este tipo de aplicación. Una limitante de nuestro proyecto es que la base pan and tilt no tiene algún grado de protección contra las lluvias, lo que puede llegar a dañar el funcionamiento interno del dispositivo, ya que, aunque el sensor si contenga un grado de protección IP (contra el agua) esta queda a la intemperie.

#### 4.4. Señales de onda.

Los resultados de las pruebas del análisis del carril fueron satisfactorios mostrando cada minuto el análisis del carril, tomando en cuenta el minuto analizado y el histórico de los 4 minutos anteriores. En base a esto podemos definir las señales de control que integran el proyecto:

- señal del sensor simulado por la K64.
- señales de los motores para movimiento continuo.

Como se mencionó en el marco teórico, los motores se controlan mediante una señal de PWM. Esta señal de acuerdo a su ciclo de trabajo puede llegar a controlar el movimiento de los motores hacia la izquierda, derecha o detenerse. Los resultados de esta señal se pueden ver en las figuras 34, 35 y 36.

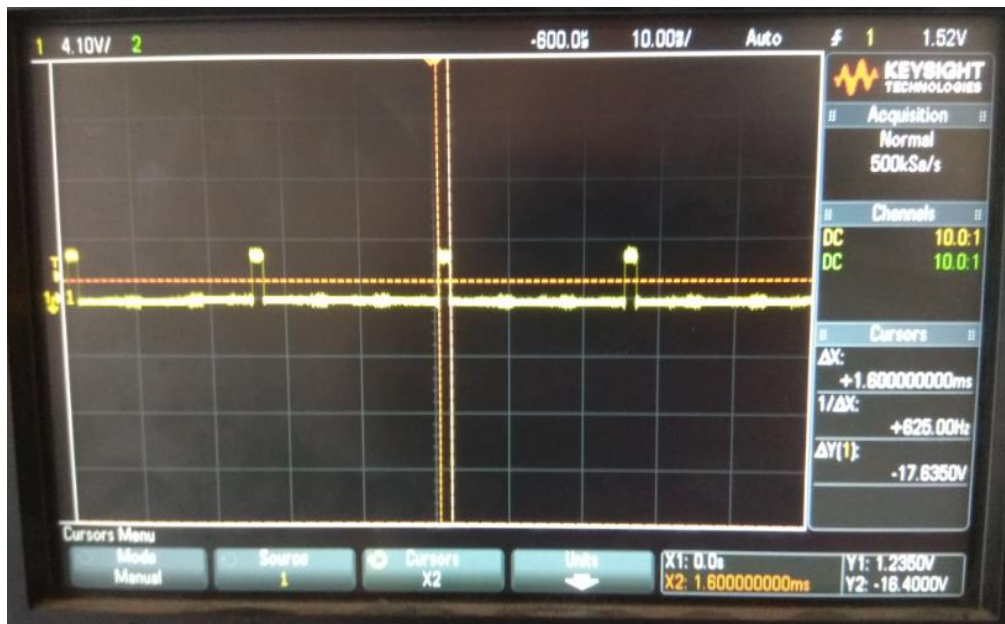


Figura 32 Respuesta del PWM para movimiento a la izquierda.

Como se muestra en la figura 34, el ciclo de trabajo es menor del 10 por ciento, y como se mencionó en el marco teórico la señal tiene un tiempo en alto de 1.6 ms (mili segundos).

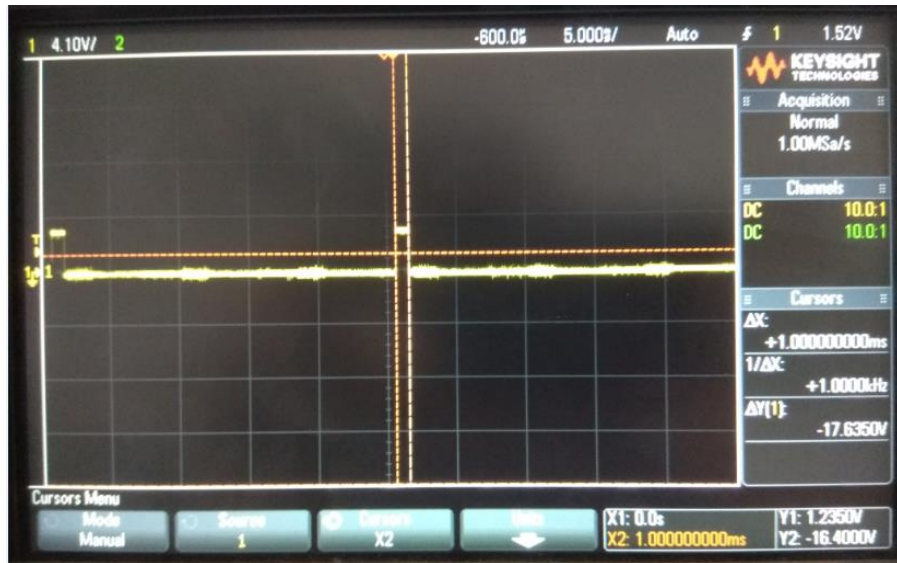


Figura 33 Respuesta del PWM para detener el servo motor.

Como se muestra en la figura 35, el ciclo de trabajo es menos del 10 por ciento, y como se mencionó en el marco teórico la señal tiene un tiempo en alto de 1 ms (mili segundos).

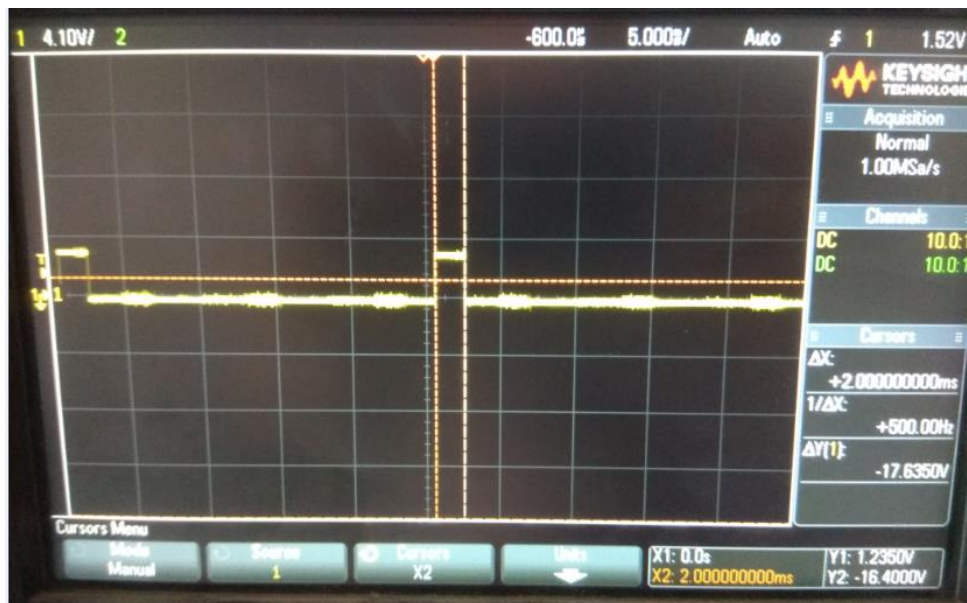


Figura 34 Respuesta del PWM para movimiento del servomotor a la derecha.

Como se muestra en la figura 36, el ciclo de trabajo es menos del 10 por ciento, por lo que la señal tiene un tiempo en alto de 2 ms (mili segundos).

El sensor se simulará para fines de prueba con la FRDM k64, que generará la misma señal de onda de las muestras recabadas en las pruebas con el sensor, como se muestra en la figura 37. Estas ondas son variaciones de voltajes conforme al tiempo, dependiendo el tiempo de detección de algún vehículo; si las señales tenían un pico de voltaje alto significa que se detectó un vehículo o un tráiler (si se mantiene el mismo voltaje en un periodo constante).

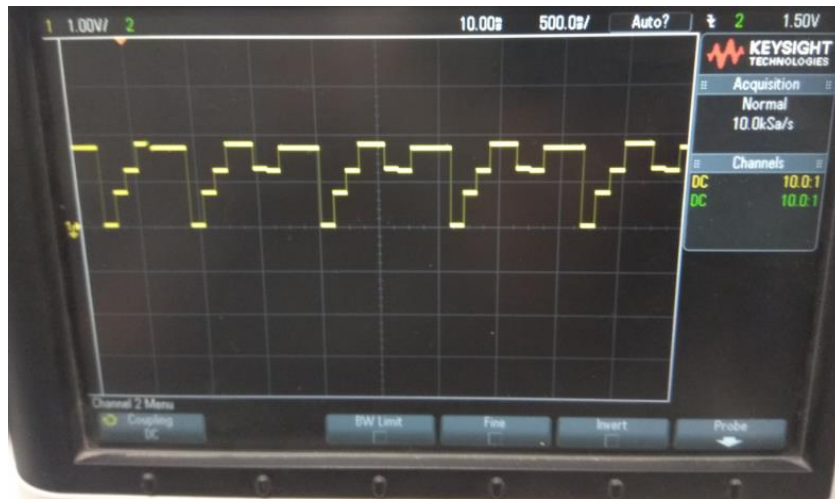


Figura 35 Respuesta de la FRDM k64 de detección vehicular.

#### 4.5. Pruebas de la base Pan & tilt:

Se comprobó la funcionalidad de la base Pan & Tilt y se observó el siguiente comportamiento de acuerdo a lo planteado:

- La base permanecerá inmóvil durante los primeros 20 segundos una vez que inicie el programa, esto debido a que desde el instante cero empieza el escaneo de los vehículos.
- Cuando pasan los primeros 20 segundos, la base comienza a moverse en el eje z para escanear a lo ancho del carril.
- Una vez terminado de escanear a lo ancho del carril, la base comenzará a moverse en el eje x, hacia la izquierda para poder escanear el siguiente carril. Cabe destacar que mientras

esté haciendo estos movimientos, el sensor está procesando en todo instante los datos que está tomando del sensor.

- El sensor se moverá en el eje x durante 3 segundos para cambiar de carril y después escaneará en el eje z a lo largo del nuevo carril.
- Ese proceso se repetirá para los tres carriles, y una vez terminado el último carril volverá a su posición original remitiéndose de nuevo el algoritmo en el punto uno.

#### 4.6. Resultado de cálculo de detección de tránsito

En la figura 38 se observa la respuesta del sensor para un escenario donde el tránsito vehicular es ligero. A partir de esta gráfica mostraremos un primer análisis de los resultados del cálculo. El tiempo representado para esta medición corresponde aproximadamente para 8.8 minutos (533 segundos).

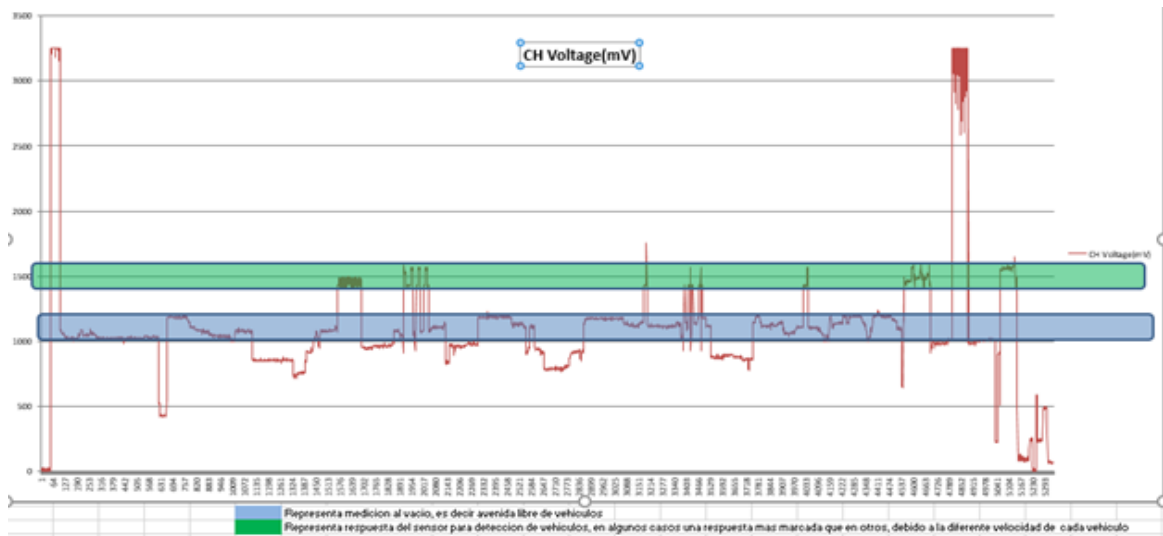


Figura 36 Señal medida con sensor en un escenario de tránsito ligero.

El algoritmo cada minuto calcula el estado del tránsito. En la figura 38 se observa que hubo detección de vehículos hasta el tercer minuto (167 segundos) y, a partir de allí, esporádicamente hubo flujo vehicular. Durante el primer y segundo minuto el resultado del cálculo fue una salida igual a “Sin tránsito vehicular”, ya que el algoritmo no genera ninguna cuenta. La función void

ADC\_Average(void) (ver apéndice) no encuentra ningún valor dentro de los umbrales de detección. Para el tercer minuto se observa que existen algunas mediciones dentro del umbral de detección. Durante este minuto, estas mediciones son contabilizadas por la función void Traffic\_ObjectDetection(void) (ver apéndice). Para que cada una de las mediciones que se encuentran dentro del umbral de detección sea contabilizada deben tener cierta persistencia, lo cual asegura que no fue una medición falsa. Cada minuto, mediante la función void Traffic\_ObjectPerMin(void) (ver apéndice), se hace un sumatorio total de objetos detectados y se compara contra una escala definida que determina el estado del tránsito actual. Dicha escala se puede calibrar.

La salida del sistema a partir de los cálculos realizados, y una vez comparando los resultados con la escala de tránsito definida, se observa en la figura 39.

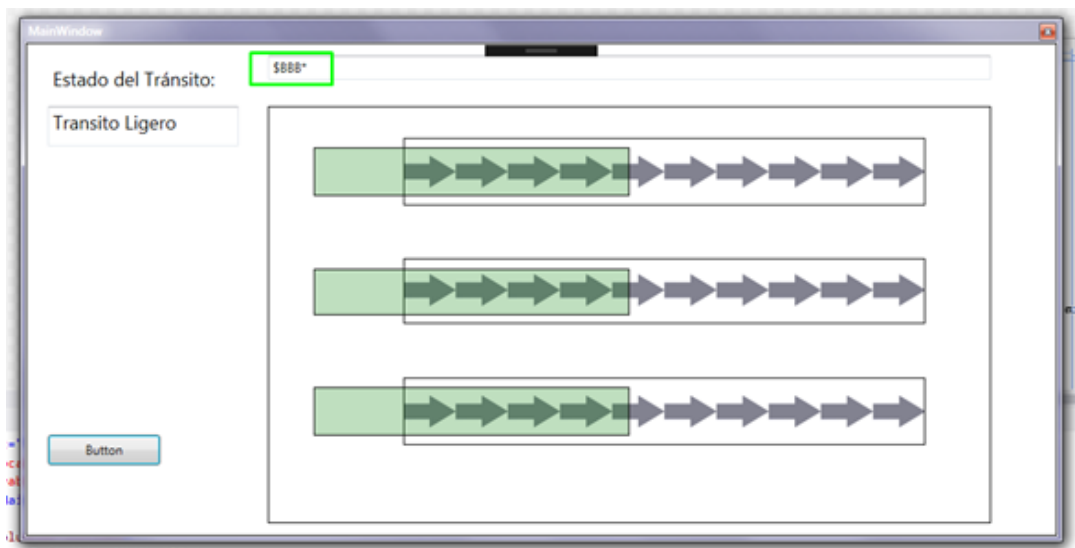


Figura 37 Salida del sistema para una señal de tránsito ligero.

Esto se debe a que el conteo final de objetos detectados al tercer minuto fue menor a las cuentas necesarias para determinar el estado de tránsito fluido. En el siguiente link de internet [https://www.youtube.com/watch?v=mzOE5\\_oX-Jo&t=0s](https://www.youtube.com/watch?v=mzOE5_oX-Jo&t=0s) se puede observar el comportamiento del sistema terminado. Mediante la tarjeta FRDM k64 de freescale se simula la señal del sensor. Se ha generado una forma de onda de tal forma que ocasione en el sistema de detección un cambio en el estado desde “Sin tránsito” hasta “Tránsito detenido” y también agregando algunas otras variaciones.

## 5.CONCLUSIONES.

Mediante el presente proyecto mostramos cómo es que el tránsito lento en las vialidades es uno de los principales problemas en las grandes ciudades y la importancia de encontrar una respuesta. La solución mostrada con el prototipo presentado propone, mediante un sensor de radar, analizar las grandes vialidades, contando con la capacidad de escanear varios carriles y mostrar el estado del flujo vehicular. Utilizando componentes de bajo costo, como lo es el sensor de radar y el mecanismo de pan and tilt, se presentó un proyecto económico que propone una alternativa a los radares de tráfico actuales.

Los dos ejes de libertad con los que cuenta la base otorgan la característica de poder analizar más de un carril sobre una avenida, siendo controlado el movimiento por el mismo microcontrolador con el cual analizamos la señal del sensor. La señal del sensor es analizada durante un minuto, y el estado del sensor se determina tomando en cuenta el minuto actual, así como los 4 minutos anteriores. La característica de comunicación mediante el puerto serial proporciona la capacidad de comunicación con otros sistemas de control de tránsito que complementarían la solución a la reducción del problema de flujo; en este proyecto utilizamos una aplicación en PC, una GUI con la cual se muestran los estados de los carriles, actualizando su información cada minuto. Esta GUI cuenta con animaciones sobre el estado del tránsito de cada carril. Con esto se muestra las capacidades del sensor y cómo es que los sistemas externos pueden aprovechar los resultados de análisis del sensor para tomar decisiones, como, por ejemplo: un semáforo inteligente conectado a este sensor podría decidir a qué avenida darle prioridad según la cantidad de tránsito detectada.

Durante el proceso de investigación y desarrollo del prototipo se obtuvieron resultados que demuestran el correcto funcionamiento del sistema. Aunque se llegó a un sistema determinado, existen varias mejoras que pueden irse integrando. Una mejora es la comunicación mediante TCP IP, la cual permitiría enviar la información a una red central de procesamiento de datos de tránsito vehicular. Otra mejora es el manejo de datos estadísticos como puede ser las horas más conflictivas y las horas con el tránsito más bajo. También es posible agregar algunos comandos que facilitarían

la operación del sistema como puede ser un comando de auto calibración, el cual tomaría en cuenta la altura de instalación del sistema para generar las zonas de detección de objetos, así como para direccionar la base móvil a un carril inicial.



## 6.Apéndice

```
static void Move_pwm(uint32_t moving)
{
    if (moving==0x09)//derecha
    {
        pwm(4);
    }
    if (moving==0x07)//stop
    {
        pwm(6);
    }
    if (moving==0x08) //izquierda
    {
        pwm(8);
    }
    if(moving==0x06)
    {
        pwm(0);
    }
}
```

1. Código de movimiento de motor en base a movimiento de los ejes.

```
void pwm(uint32_t duty_cycle)
{
    uint32_t i;

    const Pin *list = &pinPwm2[0];
    PIO_SetInput(list->pio, list->mask, list->attribute);

    /* PIO configuration */
    PIO_Configure(pinPwm, PIO_LISTSIZE(pinPwm));
    for (i = 0; i < DUTY_BUFFER_LENGTH; i++)
        dwDutys[i] = i / 2;

    /* Enable PWMC peripheral clock */
    PMC_EnablePeripheral(ID_PWM0);

    /* Configure interrupt for PWM transfer */
    NVIC_DisableIRQ(PWMO_IRQn);
    NVIC_ClearPendingIRQ(PWMO_IRQn);
    NVIC_SetPriority(PWMO_IRQn, 0);
}
```

```

/* Set clock A to run at PWM_FREQUENCY * MAX_DUTY_CYCLE (clock B is not
used) */
PWMC_ConfigureClocks(PWM0, PWM_FREQUENCY * MAX_DUTY_CYCLE , 0, BOARD_MCK);

/* Configure PWMC channel for LED0 (left-aligned, enable dead time
generator) */
PWMC_ConfigureChannel( PWM0,
    0, /* channel */
    PWM_CMR_CPRE_CLKA, /* prescaler, CLKA */
    0, /* alignment */
    PWM_CMR_CPOL /* polarity */
);

PWMC_ConfigureSyncChannel(PWM0,
    /* Define the synchronous channels by the bits SYNCx */
    (1 << CHANNEL_PWM_LED0),
    /* Select the manual write of duty-cycle values and the automatic
update by setting the field UPDM */
    PWM_SCM_UPDM_MODE2,
    0,
    0);

/* Configure channel 0 period */
PWMC_SetPeriod(PWM0, 0, DUTY_BUFFER_LENGTH);
/* Configure channel 0 duty cycle */
PWMC_SetDutyCycle(PWM0, 0, duty_cycle);
/* Define the update period by the field UPR in the PWM_SCUP register*/
PWMC_SetSyncChannelUpdatePeriod(PWM0, 8);
/* Enable the synchronous channels by writing CHID0 in the PWM_ENA register */
PWMC_EnableChannel(PWM0, 0);
/* Enable PWM interrupt */
PWMC_EnableIt(PWM0, 0, PWM_IER2_WRDY);
NVIC_EnableIRQ(PWM0_IRQn);
PIO_SetInput(list->pio, list->mask, list->attribute);
}

```

## 2. Código de movimiento del PWM.

```

void start_moving(uint16_t *time)
{
    seconds=*time;
    if(seconds == TWENTY_SECONDS&& move_stop==0 )
    {
        Move_pwm(RIHT);
        *time = 0u;
        move_stop=1;
    }
    if(seconds == ONE_SECONDS && move_stop==1 )
    {
        Move_pwm(STOP);
        *time = 0u;
        if(second_time==0)
        {
            move_stop=2;
            second_time=1;
        }
        else
        {
            move_stop=4;
            second_time=0;
        }
    }
    if(seconds == ONE_SECONDS && move_stop==2 )
    {
        Move_pwm(LEFT);
        *time = 0u;
        move_stop=1;
    }
    if(seconds == ONE_SECONDS && move_stop==4 )
    {
        Move_pwm2(LEFT);
        *time = 0u;
        move_stop=5;
    }
    if(seconds == THREE_SECONDS && move_stop==5 )
    {
        Move_pwm2(STOP);
        *time = 0u;
        if(second_time==0)
        {
            move_stop=6;
            second_time=1;
        }
        else
        {
            move_stop=0;
            second_time=0;
        }
    }
    if(seconds == ONE_SECONDS && move_stop==6 )
    {
        Move_pwm2(RIHT);
        *time = 0u;
        move_stop=5;
    }
}

```

3. Movimiento de los motores.

```

void _ADC_Init(void)
{
    AFEC_Initialize(AFECO, ID_AFECO);
    /* Disable the start ADC conversion by PWM trigger (event line 0), enable the software trigger */
    AFEC_SetModeReg(AFECO, AFEC_MR_TRGEN_DIS | AFEC_MR_FREERUN_OFF | AFEC_MR_TRANSFER(1) | AFEC_MR_TRACKTIM(0) | AFEC_MR_ONE | AFEC_MR_STARTUP_SUT64);
    AFEC_SetClock(AFECO, AFE_CLK_BOARD_MCK);
    AFEC_EnableChannel(AFECO, TEST_CHANNEL);
    AFEC_SetAnalogOffset(AFECO, TEST_CHANNEL, CHANNEL_OFFSET);
    AFEC_SetAnalogControl(AFECO, AFEC_ACR_IBCTL(1) | AFEC_ACR_PGA0_ON | AFEC_ACR_PGA1_ON);
    /* disable the Averaging, TAG channel not used, Single Trigger Mode */
    AFEC_SetExtModeReg(AFECO, 0 | AFEC_EMR_RES(156) | AFEC_EMR_STM);
}

```

#### 4. Inicialización módulo AFEC.

```

void _AFEC_DMA_Init(void)
{
    Afe_ConfigureDma(&MyAfeDma, AFECO, ID_AFECO, &MyxDmad);
    SET_AFEC_SAMPLING(SAMP_PER3, &ADC_BUFF[0], SIZE3);
}

```

#### 5. Inicialización módulo DMA.

```

static void _ConfigureRtt(void)
{
    uint32_t previousTime;

    /* Configure RTT for a 1 second tick interrupt */
    RTT_SetPrescaler(RTT, 32768);
    previousTime = RTT_GetTime(RTT);
    while (previousTime == RTT_GetTime(RTT));

    /* Enable RTT interrupt */
    NVIC_DisableIRQ(RTT_IRQn);
    NVIC_ClearPendingIRQ(RTT_IRQn);
    NVIC_SetPriority(RTT_IRQn, 0);
    NVIC_EnableIRQ(RTT_IRQn);
    RTT_EnableIT(RTT, RTT_MR_RTTINCIEN);
}

```

#### 6. Inicialización módulo RTT.

```

void ADC_Average(void)
{
    uint32_t Average_dist = 0;
    Average_dist = u32Average_cal(ADC_BUFF, ADCSIZE);

    if( (Average_dist > MINVALUE_OB1 && Average_dist < MAXVALUE_OB1) ||
        (Average_dist > MINVALUE_OB2 && Average_dist < MAXVALUE_OB2) )
    {
        ul6AverageCount++;
    }
    else
    {
        // Do nothing
    }
}

```

#### 7. Filtrado de señal del ADC.

```

void Traffic_ObjctDetection(void)
{
    if( ul6AverageCount > THRESHOLD_AVERAGE )
    {
        ul6ObjectCount++;
    }
    else
    {
        // Do nothing
    }
    ul6AverageCount = 0;
}

```

#### 8. Detección de objetos.

```

void Traffic_ObjectPerMin(void)
{
    static uint8_t Index = 0;
    uint8_t i = 0;
    uint16_t ul6AmountPer5Min = 0;
    uint8_t u8Traffic_Status;
    uint8_t buffer[3] = {'0','0','0'};

    ObjectPerMin[Index] = ul6ObjectCount;
    ul6ObjectCount = 0;

    for(i = 0; i <5; i++)
    {
        ul6AmountPer5Min += ObjectPerMin[i];
    }

    u8Traffic_Status = Traffic_Validation(ul6AmountPer5Min);

    switch(u8Traffic_Status)
    {
        case NO_TRAFFIC:
            printf( "NO_TRAFFIC\n\r" );
            usart_buffer_sensor(buffer,3);
            break;
        case LIGHT_TRAFFIC:
            printf( "LIGHT_TRAFFIC\n\r" );
            usart_buffer_sensor(buffer,3);
            break;
        case SMOOTH_TRAFFIC:
            printf( "SMOOTH_TRAFFIC\n\r" );
            usart_buffer_sensor(buffer,3);
            break;
        case HEAVY_TRAFFIC:
            printf( "HEAVY_TRAFFIC\n\r" );
            usart_buffer_sensor(buffer,3);
            break;
        case STOP_TRAFFIC:
            printf( "STOP_TRAFFIC\n\r" );
            usart_buffer_sensor(buffer,3);
            break;
        default:
            break;
    }

    Index++;
    if(Index == 5)
    {
        Index = 0;
    }
}

```

9. Determinación de tránsito vehicular.

# Bibliografía

- [1] “Destaca la ZMG en atorones viales”. [En línea]. Disponible en: <https://www.mural.com/aplicacioneslibre/articulo/default.aspx?id=1113729&v=3&md5=4703bbf5fd5faeca296f7c6516d632fa&ta=0dfdbac11765226904c16cb9ad1b2efe>. [Consultado: 05-jun-2018].
- [2] INRIX, “INRIX Global Traffic Scorecard”, *INRIX - INRIX*. [En línea]. Disponible en: <http://inrix.com/scorecard/>. [Consultado: 05-jun-2018].
- [3] G. Seehausen, “24-GHz-FM-CW-Radar for Detection of Information for Traffic Purposes”, 1984, vol. 84, pp. 251–253.
- [4] Z. Li, C. Li, Y. Zhang, y X. Hu, “Intelligent traffic light control system based on real time traffic flows”, 2017, pp. 624–625.
- [5] A. Klilou, F. Bourzeix, O. Bourja, Y. Zennayi, L. Mabrouk, y S. Belkouch, “Real-time parallel implementation of road traffic radar video processing algorithms on a parallel architecture based on DSP and ARM processors”, 2015, pp. 183–188.
- [6] J. Fang, H. Meng, H. Zhang, y X. Wang, “A Low-cost Vehicle Detection and Classification System based on Unmodulated Continuous-wave Radar”, 2007, pp. 715–720.
- [7] “Office of Highway Policy Information - Policy | Federal Highway Administration”. [En línea]. Disponible en: <https://www.fhwa.dot.gov/policyinformation/pubs/vdstits2007/>. [Consultado: 05-jun-2018].
- [8] Sang Jin Park, Tae Yong Kim, Sung Min Kang, y Kyung Heon Koo, “A novel signal processing technique for vehicle detection radar”, 2003, vol. 1, pp. 607–610.
- [9] “PT2645-S Pan & Tilt”. [En línea]. Disponible en: <https://www.servocity.com/pt2645-s-pan-tilt>. [Consultado: 14-jun-2018].
- [10] L. G. C. Ramírez, G. S. A. Jiménez, y J. M. Carreño, *Sensores y Actuadores*. Grupo Editorial Patria, 2014.

