

**UNIVERSIDAD NACIONAL DE MAR DEL PLATA**

**FACULTAD DE INGENIERÍA**

**DEPARTAMENTO DE ELECTRÓNICA**

**LABORATORIO DE COMUNICACIONES**

**SISTEMA DE REGISTRO AUTOMÁTICO  
DE INFORMACIÓN PARA EL ANÁLISIS  
Y GESTIÓN DEL TRÁNSITO VIAL**

**TRABAJO PRESENTADO POR:**

**MANUEL STILLITANO**

**MATRICULA: 11756**

**DIRECTOR DEL PROYECTO:**

**ING. JUAN CARLOS BONADERO**

**CODIRECTOR DEL PROYECTO:**

**DR. ING. ALEJANDRO URIZ**



RINFI se desarrolla en forma conjunta entre el INTEMA y la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución-  
NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).



## RESUMEN

El presente proyecto final tuvo como objetivo el desarrollo de un sistema automático de registro y transmisión de información útil para el análisis y planificación del tráfico urbano y mantenimiento de los caminos. Este sistema permitirá planificar de mejor manera el tránsito, realizar más eficientemente el mantenimiento de los canales por los cuales este circula y hacer estudios tanto sobre la ubicación geográfica de los problemas como su distribución temporal. El uso de dispositivos de sensado y recolección de datos en automóviles permite recolectar información desde el punto de vista del conductor, lo que significa agregar una nueva perspectiva dinámica al análisis de las problemáticas, que tradicionalmente se analizan desde lo estático, es decir, desde un punto geográfico específico.

Para este proyecto final se hará uso de dos tipos de dispositivos, con intercomunicación inalámbrica entre los mismos: unos con autonomía y capacidad de movilidad y otros ubicados en lugares fijos y estratégico. El primer conjunto (los dispositivos M1) está comprendido por un módulo inteligente, autónomo y geolocalizado (uso de tecnología GPS) que estará ubicado dentro de un vehículo de tránsito continuo, sin recorrido estable. Dicho dispositivo se encargará de sensar diversos parámetros del automóvil e identificar, mediante algoritmos de reconocimiento, categorizados y específicos, la presencia de una o varias situaciones predefinidas para luego registrar una serie de datos relevantes a la misma. El segundo conjunto (los dispositivos M2) consiste en un receptor maestro (coordinador) que se encargará de reconocer a cualquiera de los dispositivos móviles cuando se encuentren en un área de alcance determinada del coordinador. El dispositivo coordinador M2 establecerá una comunicación con el dispositivo móvil M1 y recolectará todos los datos almacenados en este. Luego los datos recolectados serán subidos a una base de datos on-line mediante conexión Ethernet para su disponibilidad y posterior análisis. La información generada por los dispositivos M1 y recolectada por los M2 será procesada en servidores para su fácil visualización, análisis y uso en la toma de decisiones posteriores.

El rango de uso potencial de esta solución es muy variado y altamente escalable, permitiendo el agregado en forma paulatina de nuevas funcionalidades con completa compatibilidad hacia atrás. Además, mediante la comunicación bidireccional de los dispositivos móviles, es posible su reprogramación y actualización a distancia.

# ÍNDICE GENERAL

RESUMEN .....	I
ÍNDICE GENERAL .....	II
<b>1. INTRODUCCIÓN .....</b>	<b>1</b>
1.1. EL PROBLEMA .....	4
1.2. RELEVAMIENTO AUTOMÁTICO VIAL: DEFINICIÓN Y ESTADO DEL ARTE .....	7
1.3. PROPUESTA DE NUEVAS TÉCNICAS DE RELEVAMIENTO AUTOMÁTICO VIAL.....	17
1.4. DESCRIPCIÓN DEL INFORME .....	18
<b>2. SISTEMA PROPUESTO .....</b>	<b>19</b>
2.1. PRINCIPIO DE FUNCIONAMIENTO Y COMPONENTES DEL SISTEMA.....	19
2.1.1. <i>Diagrama en bloques del sistema completo</i> .....	21
2.1.2. <i>Bloque de posicionamiento global</i> .....	21
2.1.3. <i>Bloque de almacenamiento de datos permanente</i> .....	23
2.1.4. <i>Bloque de comunicación inalámbrica</i> .....	24
2.1.5. <i>Bloque Ethernet</i> .....	27
2.1.6. <i>Servidor y base de datos</i> .....	27
2.1.7. <i>Dispositivos M1 y M2</i> .....	28
2.2. DETALLE DEL MICROCONTROLADOR .....	29
2.2.1. <i>Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART)</i> .....	32
2.2.2. <i>Master Synchronous Serial Port (MSSP)</i> .....	37
2.2.3. <i>Puertos I/O y Peripheral Pin Select (PPS)</i> .....	41
2.2.4. <i>Oscilador del Sistema</i> .....	45
2.3. DESCRIPCIÓN DE LOS CÓDIGOS DE PROGRAMA .....	48
2.3.1. <i>Código del dispositivo M1</i> .....	48
2.3.2. <i>Código del dispositivo M2</i> .....	52
<b>3. DISEÑO, DESARROLLO Y EXPERIMENTACIÓN .....</b>	<b>54</b>
3.1. DISEÑO DEL CIRCUITO ESQUEMÁTICOS DEL DISPOSITIVO M1 .....	54
3.2. DISEÑO DE LOS CIRCUITOS IMPRESOS .....	55
3.3. DISEÑO DE LOS DISPOSITIVOS FINALES .....	58
3.4. DESARROLLO Y EXPERIMENTACIÓN .....	60
3.4.1. <i>Experimentación</i> .....	62
3.4.2. <i>Cálculos y estimaciones</i> .....	64
<b>4. CONCLUSIONES Y LÍNEAS FUTURAS DE TRABAJO.....</b>	<b>66</b>
<b>5. BIBLIOGRAFÍA .....</b>	<b>68</b>

# 1. INTRODUCCIÓN

La República Argentina posee una tasa de urbanización históricamente alta. La población urbana representaba para el 2001 el 89,4% de la población total del país pero para el año 2010 dicho porcentaje ascendía al 91%. Mientras la mitad de su población reside en las 6 grandes ciudades o aglomerados, un tercio (13 millones) lo hace en las 273 ciudades de tamaño medio, de entre 10 mil y 500 mil habitantes. Es para destacar el caso de la Región Metropolitana de Buenos Aires, donde habita el 37% de la población del país. En la Ilustración 1 se puede visualizar claramente esta tendencia:

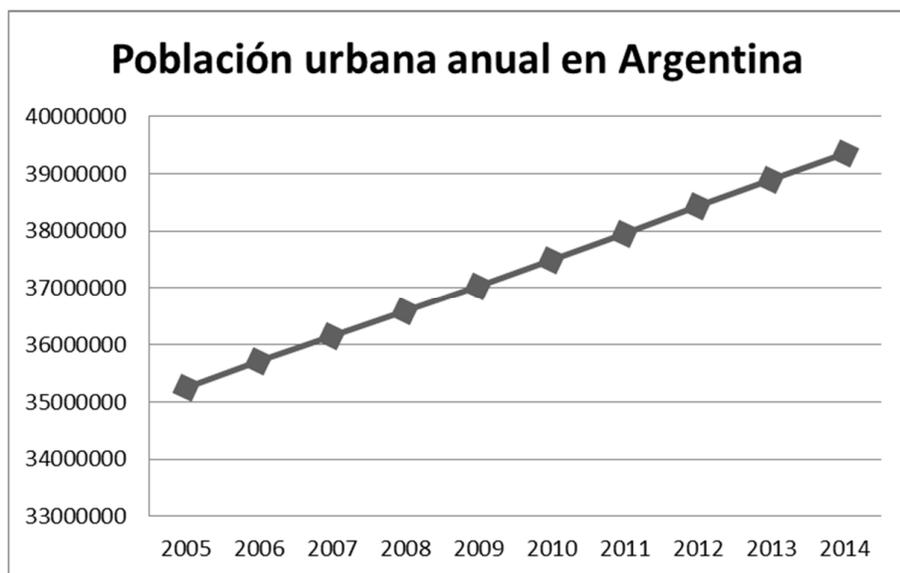


Ilustración 1: crecimiento de la población urbana. Argentina.

Además del constante incremento poblacional de las urbes Argentinas se puede sumar al análisis otro fenómeno de relevancia, el aumento anual del parque automotor. El informe anual correspondiente al 2015 de la Asociación de Fábricas de Automotores (ADEFA) dio cuenta de la evolución del parque de automóviles, vehículos utilitarios livianos, vehículos de carga y ómnibus por provincia. En el país la flota automotriz circulante a fines de 2014 fue de unos 919.123 vehículos más que en 2013 cuando se registraron 12.456.864 unidades. Dicho incremento fue del 7,4 %, dando un parque automotor total de 13.375.987 vehículos. El informe aseguró que con más de 13 millones de vehículos en las calles se desprende otro dato interesante: Argentina es el país de la región con más vehículos por habitante, ya que tiene 3,2 habitantes por unidad (42.980.026 de personas), al cierre de 2014. En dicho informe el parque automotor fue clasificado por tipo de combustible, ubicación geográfica, marcas, antigüedad de la unidad y por segmento, lo que permitió determinar que un 75,8% son automóviles, un 18,7% vehículos comerciales livianos y un 5,5% de pesados, incluyendo en este último

segmento camiones y ómnibus. Esto quiere decir que la gran mayoría del parque automotor corresponde a automóviles de uso no comercial. ADEFA precisa que más del 50% de la flota se encuentra concentrada en la Provincia de Buenos Aires y Ciudad Autónoma de Buenos Aires, secundado por Córdoba y Santa Fe que en conjunto no llegan a sumar el 20%. Todo lo mencionado puede verse en detalle en la Tabla 1, Tabla 2 y Tabla 3.

<b>Año</b>	<b>Automóviles</b>	<b>Vehículos utilitarios livianos</b>	<b>Vehículos de carga</b>	<b>Ómnibus</b>	<b>Total</b>
<b>2006</b>	5.325.231	1.370.312	417.042	62.785	<b>7.175.370</b>
<b>2007</b>	5.745.200	1.488.040	440.708	64.954	<b>7.738.902</b>
<b>2008</b>	6.270.915	1.623.612	495.997	69.248	<b>8.459.772</b>
<b>2009</b>	6.706.100	1.681.549	498.957	68.267	<b>8.954.873</b>
<b>2010</b>	7.604.921	1.891.846	546.164	73.087	<b>10.116.018</b>
<b>2011</b>	8.269.443	2.040.269	573.481	75.891	<b>10.959.084</b>
<b>2012</b>	8.682.726	2.122.901	593.476	77.445	<b>11.476.548</b>
<b>2013</b>	9.451.329	2.293.172	625.276	87.087	<b>12.456.864</b>
<b>2014</b>	10.143.583	2.495.304	654.215	82.885	<b>13.375.987</b>

Tabla 1: evolución del parque automotor por categoría para todo el país.

<b>Año</b>	<b>Automóviles</b>	<b>Vehículos utilitarios livianos</b>	<b>Vehículos de carga</b>	<b>Ómnibus</b>	<b>Total</b>
<b>2006 - 2007</b>	7,9%	8,6%	5,7%	3,5%	<b>7,9%</b>
<b>2007 - 2008</b>	9,2%	9,1%	12,5%	6,6%	<b>9,3%</b>
<b>2008 - 2009</b>	6,9%	3,6%	0,6%	-1,4%	<b>5,9%</b>
<b>2009 - 2010</b>	13,4%	12,5%	9,5%	7,1%	<b>13,0%</b>
<b>2010 - 2011</b>	8,7%	7,8%	5,0%	3,8%	<b>8,3%</b>
<b>2011 - 2012</b>	5,0%	4,1%	3,5%	2,0%	<b>4,7%</b>
<b>2012 - 2013</b>	8,9%	8,0%	5,4%	12,5%	<b>8,5%</b>
<b>2013 - 2014</b>	7,3%	8,8%	4,6%	-4,8%	<b>7,4%</b>

Tabla 2: evolución anual del parque automotor por categoría para el total del país.

<b>Año</b>	<b>Buenos Aires</b>	<b>CABA</b>	<b>Córdoba</b>	<b>Santa Fe</b>	<b>Mendoza</b>	<b>Entre Ríos</b>	<b>Tucumán</b>	<b>Chubut</b>
<b>2011</b>	3.251.818	1.216.732	809.330	712.494	397.185	249.017	154.346	139.119
<b>2012</b>	3.444.620	1.219.995	845.814	742.385	419.692	265.313	166.997	146.169
<b>2013</b>	3.758.929	1.288.040	914.964	802.584	456.418	289.752	185.831	158.393
<b>2014</b>	4.042.076	1.333.392	977.787	856.082	489.606	312.650	204.377	171.136

Tabla 3: parque automotor de Argentina por jurisdicción (automóviles).

Los datos mencionados anteriormente denotan el enorme crecimiento tanto de la urbanización como del sector productivo automotriz durante la última década en Argentina. El impacto que este crecimiento genera en las ciudades es notable. Estos incrementos, respecto

de la cantidad de individuos por metro cuadrado y de vehículos por habitante, implican inherentemente la intensificación de los niveles y tipos de desplazamiento de personas, bienes y productos. Un mayor número de desplazamientos viene acompañado de un aumento sobre la probabilidad de conflictos entre sus protagonistas y, por ende, cambios en el comportamiento del día a día de los ciudadanos. Debido a estas cambiantes circunstancias nace la necesidad de actualizar y mejorar los sistemas y vías de transporte urbanos para así adaptarse a las mismas. Dicha necesidad interpela a los gobiernos locales hacia el desarrollo de sistemas de gerenciamiento más eficaces y la búsqueda de soluciones más eficientes para los continuos, crecientes y cambiantes problemas de movilidad y transporte interurbano. En este marco, es cuando se presentan nuevas estrategias para una gestión municipal más eficiente del tránsito y para una mayor concientización urbana sobre la seguridad vial.

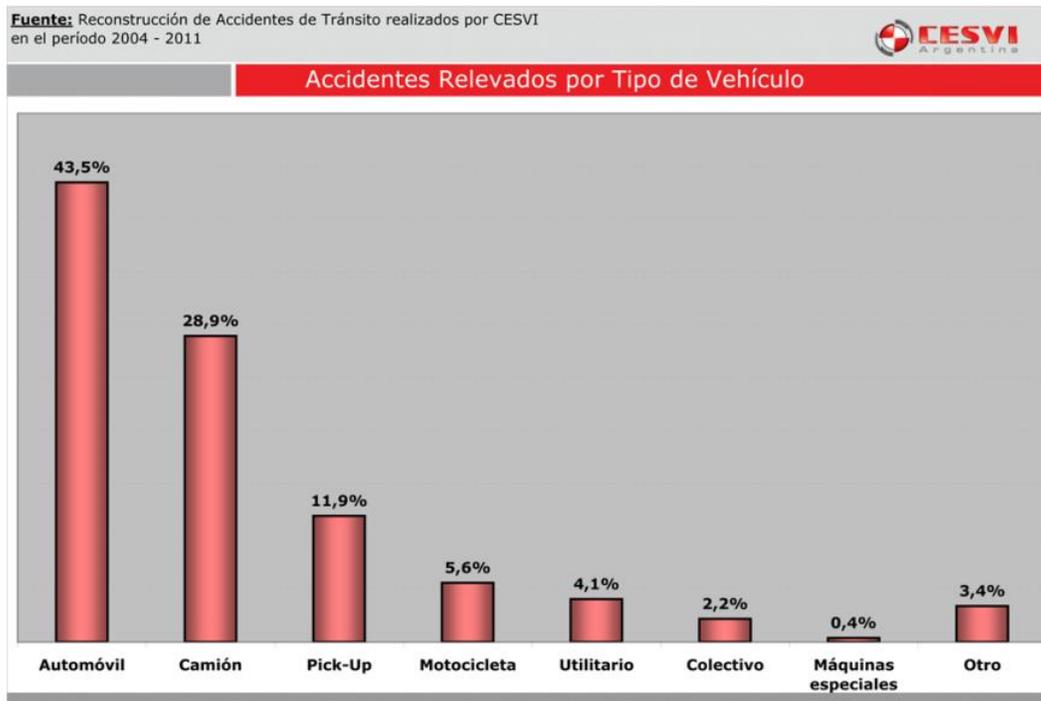


Ilustración 2: accidentes relevados por tipo de vehículo. Argentina 2004-2011.

En la Ilustración 2 se puede ver como la mayoría de los siniestros de tránsito son protagonizados por los automóviles, los cuales en general comprenden el uso privado de transporte de personas y bienes. Por otro lado, en la Ilustración 3 es notable la concentración de siniestros en áreas donde la urbanización es mayor, como es el caso de Buenos Aires, Santa Fe y Córdoba.

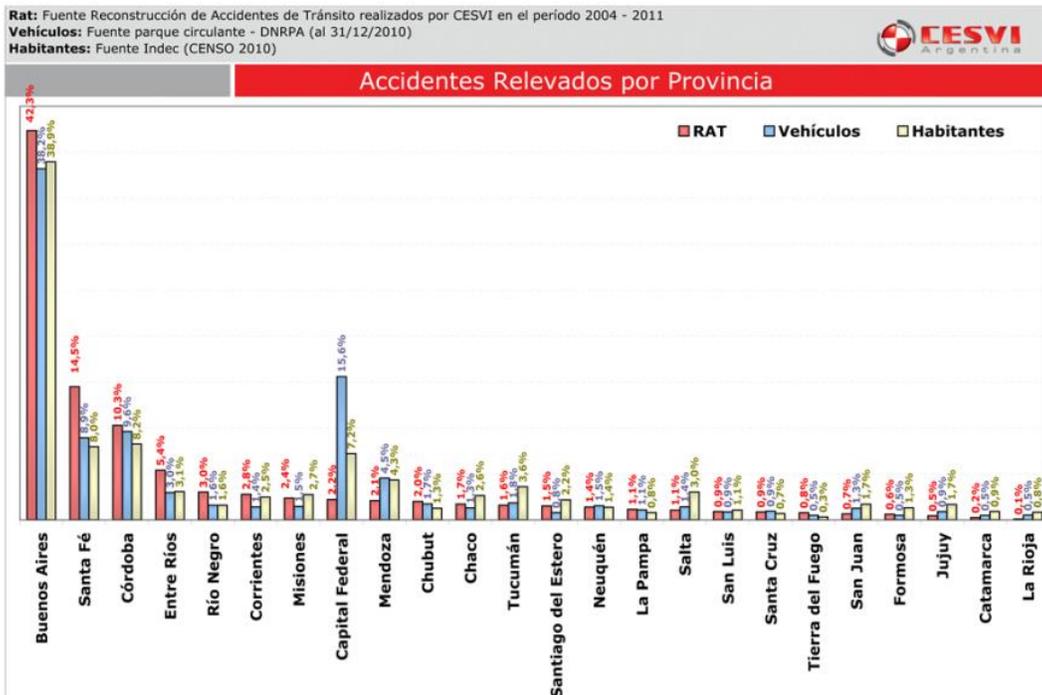


Ilustración 3: accidentes relevados por provincia. Argentina 2004-2011.

Uno de los aspectos fundamentales de la seguridad vial es el control y su correcta gestión por parte de las instituciones encargadas de la misma. A partir de un cuidadoso y adecuado seguimiento del tránsito, su comportamiento y del espacio donde este se desarrolla, se puede minimizar problemáticas como los accidentes, el comportamiento incorrecto e imprudente de algunos automovilistas, la manutención de las vías y el funcionamiento ineficiente de sistemas de regulación del flujo de tránsito. Todo esto garantiza la seguridad, calidad y el confort del peatón y conductor que transitan la vía pública.

## 1.1. El problema

El transporte motorizado puede llegar a producir un gran número de impactos ambientales y sociales que bien podrían tener una fuerte y negativa repercusión en la calidad de vida de las personas. Las consecuencias del actual modelo de transporte nos afectan por vías distintas. Por un lado, están los problemas relacionados con la congestión, que se materializan en una pérdida de tiempo a la hora de desplazarse y una incomodidad social que muchas veces genera estrés y hasta conflictos entre las personas. Por otro, existen afecciones directas a la salud por la contaminación acústica y la polución, así como por la siniestralidad. Por último, no hay que olvidar los problemas que generan la ocupación y fragmentación del territorio, debido a la alta densidad de vehículos en movimiento o detenidos, que limitan o imposibilitan la utilización de las calles como algo más que canales de transporte, pareciendo más una pista de

obstáculos que un medio para circular con comodidad por la ciudad. Esto se puede apreciar en la Ilustración 4, Ilustración 5 y Ilustración 6. En las mismas se evidencia la densidad que el tráfico puede llegar a alcanzar además de la compleja situación del espacio utilizado vs libre.



Ilustración 4: tráfico en Mar del Plata, Argentina.



Ilustración 5: Mar del Plata, Argentina. Zona centro, estacionar puede ser una odisea.

Problemáticas de congestión y tráfico típicas van desde embotellamientos en zonas críticas, como intersecciones entre avenidas y calles con elevado caudal de tránsito, baches que pueden provocar averías en el vehículo o ser fuentes de maniobras peligrosas e imprudentes, falta de sincronismo entre semáforos que atentan al flujo controlado y ordenado del tráfico, por citar algunas. En la Ilustración 5 se destaca la continua e inalterable situación que se presenta en la zona céntrica de Mar del Plata donde las autos estacionados ocupan casi todo el tiempo los sectores disponibles para tal fin, siendo una constante el hecho de no poder estacionar, generando conflictos entre los automóviles circulantes y los que aminoran la marcha para encontrar algún espacio donde poder dejar su vehículo. A su vez, la situación

recién mencionada se puede agravar aún más en lugares con mucha concurrencia de personas. Un ejemplo puede ser un colegio o escuela, donde los automovilistas suelen estacionar en doble fila en horas picos, generando así una saturación de automóviles detenidos y muy posiblemente un atoramiento en el flujo del tránsito. Esto puede verse claramente en la Ilustración 6. También podríamos mencionar situaciones inseguras, como por ejemplo alguna zona donde el contexto y la geografía se presentan de manera tal que algunos automovilistas se ven motivados a aumentar la velocidad y, a su vez, incentivando un comportamiento colectivo en el resto de los conductores que podría poner en riesgo la seguridad propia y de los transeúntes. Otras situaciones inseguras pueden presentarse por presencia de obstáculos o un desmejoramiento de las vías de tráfico. Un ejemplo simple serían los baches en las calles. Es muy común ver maniobras comúnmente denominadas como “volantazos” al querer esquivar los mismos, pudiendo generarse un siniestro debido a esta y otras maniobras imprudentes. Todas las mencionadas son situaciones que atentan contra el orden y la seguridad del día a día de conductores y peatones.



Ilustración 6: autos en doble fila. Espacio saturado por los vehículos. Mar del Plata, Argentina.

Por otro lado, existen los problemas del espacio público consumido durante el tiempo de circulación, como también se pueden ver en la Ilustración 4. Estos dependerán de la relación entre el tamaño del vehículo, la cantidad de viajeros que pueda llevar y las tasas de ocupación del mismo, y principalmente la fluidez del tránsito y estado las vías por las cuales circulan los

vehículos. El automóvil es el medio de transporte que más espacio público requiere, tanto parado como en circulación: el espacio que ocupa un viaje diario medio del hogar al trabajo en auto es 20 veces más que si se realiza en ómnibus. El automóvil es, con su elevado requerimiento de espacio público, el principal responsable de las congestiones urbanas. Todos estos problemas se ven considerablemente agravados por las tasas de ocupación tan bajas que presenta el automóvil: una media de 1,2 personas por vehículo. Tasas tan bajas que multiplican el consumo de energía, la contaminación emitida, el ruido generado, y el consumo de espacio por persona transportada.

En definitiva, el excesivo uso del automóvil es la principal causa de los problemas de congestión, ruido, contaminación del aire, siniestralidad y elevados costes del transporte en las ciudades. Una forma, entre muchas otras, de minimizar estos problemas es una correcta gestión en el flujo del tránsito, facilitando un desplazamiento eficiente, minimizando los consumos de energía, los tiempos de utilización de la calzada urbana y reduciendo la duración de viajes. Todo esto favorece positivamente al bienestar social y ambiental, en general.

## **1.2. Relevamiento automático vial: definición y estado del arte**

La planificación del tránsito consiste esencialmente en la aplicación de procedimientos y técnicas de Ingeniería de Tránsito para la optimización de los sistemas tanto existentes como por desarrollar. En otras palabras, en la planificación del tránsito y del transporte se contemplan los factores que afectan al equilibrio de los sistemas mencionados que se han desarrollado, tanto en condiciones actuales como futuras, considerando básicamente la interrelación de los elementos que forman dichos sistemas: USUARIO - VEHÍCULO - MEDIOS - CONTROL. Su valor reside en la interrelación entre el análisis de factores tales como los desplazamientos de vehículos, su respectivo uso del suelo, las redes viales por donde transitan y la interacción con la población involucrada, entre otros. Sumado a lo anterior, es necesario hacer hincapié en:

- La integración de un abordaje multidisciplinario al análisis mencionado.
- El empleo de modelos matemáticos de las condiciones donde se desenvuelven los sucesos de interés.
- El uso de procesos de simulación por computadora.

Se ha desarrollado una metodología completa para el estudio de la planificación del tránsito y transporte urbano. Dicho proceso comprende la integración de muchas de las características de acción recíproca del medio ambiente urbano, distinguiéndose etapas de:

- Obtención de datos (inventarios de tránsito).

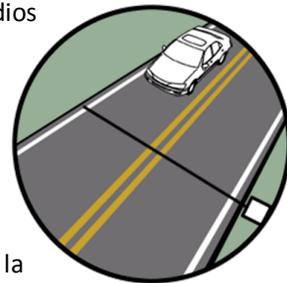
- Análisis del inventario generado.
- Cálculo de proyecciones para determinar situaciones futuras.
- Formulación y comprobación de planes de alternativa.
- Elección de planes definitivos y programas de realización e implementación.

Las condiciones existentes de un sistema de tránsito quedan reflejadas a partir de la obtención de datos empíricos, es decir el Inventario de Tránsito, contexto que representa la situación real del tránsito en espacio y tiempo. El Inventario de Tránsito es una etapa importante en el desarrollo del área en estudio ya que indica el potencial y las características de la generación actual de tránsito, el cual conduce posteriormente al cálculo de sus proyecciones correspondientes. Es de suma importancia, por lo tanto, recabar la mayor cantidad (y la mejor calidad) de datos útiles de manera eficiente para identificar eficazmente las situaciones que se deben analizar. Las herramientas utilizadas en esta temprana etapa son cruciales para poder realizar estudios posteriores de los cuales sea posible extraer conclusiones que reflejen una realidad lo más aproximada posible a la que acontece y que se está queriendo abordar.

En general, luego de analizar la problemática e identificar qué factores se deben analizar, el siguiente paso es la toma de datos empíricos, donde aparecen las herramientas antes mencionadas. Es crucial para las siguientes etapas a desarrollar la utilización de métodos e instrumental apropiado para cada tipo de situación. Existen variadas herramientas para recabar datos de diversa índole, que van desde contadores de tránsito, odómetros, equipamientos de pesaje dinámico, medidores de velocidad, hasta sistemas de vigilancia video-monitorizados. En la práctica, el equipamiento más utilizado para realizar un relevamiento de naturaleza automática podría ser:

❖ **Sistemas contadores/clasificadores automáticos:**

Los conteos de tránsito con clasificación vehicular son los estudios que permiten obtener la información relacionada con el movimiento de vehículos sobre puntos o secciones específicas dentro de un sistema vial. Miden el volumen de tránsito y, algunos, lo clasifican por rangos de velocidad y longitud del vehículo. Las técnicas más básicas de conteo emplean un método de conteo que requiere la



instalación, sobre el paso del tránsito, de detectores de superficie, como tubos neumáticos o de contacto eléctrico, los cuales detectan el vehículo que pasa y transmiten la información a registros conectados al detector, a un lado del camino. Los contadores/clasificadores de la firma Peek Traffic son de uso fijo y portátil y los distintos módulos controladores de sensores

permiten utilizar entradas de espiras, piezoeléctricos y mangueras. Este equipo admite alimentación solar o de red, comunicación y monitoreo remoto. Existen diferentes modelos de este equipamiento de conteo, como ser el ADR 1000 Plus [15], ADR 2000 Plus [15], ADR 3000 [15] y ADR SABRE [15], que son los más utilizados en el mercado. Todos poseen gran capacidad de memoria interna y tienen disponible una memoria extraíble para extender la capacidad de almacenamiento. Los tipos de datos que se pueden obtener con los ADR de Peek Traffic son: clasificación de vehículos por ejes, velocidad, longitud, avance, brecha, peso bruto, peso por eje y peso por grupo de ejes. Según el modelo de que se trate, los ADR pueden operar simultáneamente en 16 carriles o más. Los principales beneficios de esta tecnología están relacionados con la facilidad que conlleva su instalación y operación. Asimismo cuentan con comunicaciones de alta velocidad y telemetría y un panel de control integrado con lectura LCD. A continuación se describen algunas especificaciones de estos modelos:

- ADR-2000 Plus Portable Counter/Classifier: proporciona una funcionalidad óptima como una unidad portátil o semipermanente. En su configuración básica como un contador/clasificador es posible usar 2 ó 4 tubos sensores para carretera. Los tipos de datos disponibles incluyen registro por vehículo, datos por carril, clasificación de vehículos agrupados por eje, velocidad, longitud, distancia, intervalo de tiempo entre vehículos o combinaciones de los anteriores. Los vehículos pueden ser clasificados de acuerdo con el esquema FHWA (*Federal Highway Administration, E.U.A*) "F" o un esquema de clasificación personalizada definida por el usuario.



Ilustración 7: Peek ADR-2000 Plus Portable Counter/Classifier.

- ADR Sabre: es el sistema automático contador/clasificador (AVCC) de vehículos portátil más avanzado, flexible e integral en el mundo, proporcionando una gama incomparable de características y beneficios. Es una solución económica y versátil para casi todo tipo de

estudio vehicular o tecnología de sensado. El ADR Sabre es compatible con el Traffic Operations And Planning Software (TOPS) [15], así como el VIPER [15], la próxima generación en software de obtención y procesamiento de datos de Peek. Características y beneficios:

- Posibilidad de almacenamiento de los datos en tres formatos seleccionables por el usuario: agrupados, PVR o evento.
- Una variedad "ilimitada" de estudios pueden ser producidos en el propio Sabre utilizando los registros de eventos.
- Memoria de gran capacidad: 16 MB de memoria principal y hasta 4 GB de memoria SD extraíble.
- Electrónica de alta eficiencia combinada con una batería de larga vida y recarga rápida.
- Múltiples puertos de comunicación de alta velocidad.
- Líder en la Industria de interfaz (IU): amplia pantalla LCD con teclado de fácil uso y un sistema de menú simplificado para mejorar la configuración y el control.
- Múltiples características de firmware mejoradas.
- Actualizable en unidades existentes ADR 1000.



Ilustración 8: Peek ADR Sabre.

❖ **Sistemas contadores y clasificadores no invasivos:**

Los sensores más complejos utilizan una técnica denominada *Vehicle Magnetic Imaging* (VMI) [21]. Esta técnica consiste en detectar las perturbaciones que genera la masa de un vehículo al pasar a través de un campo magnético generado por el sensor.

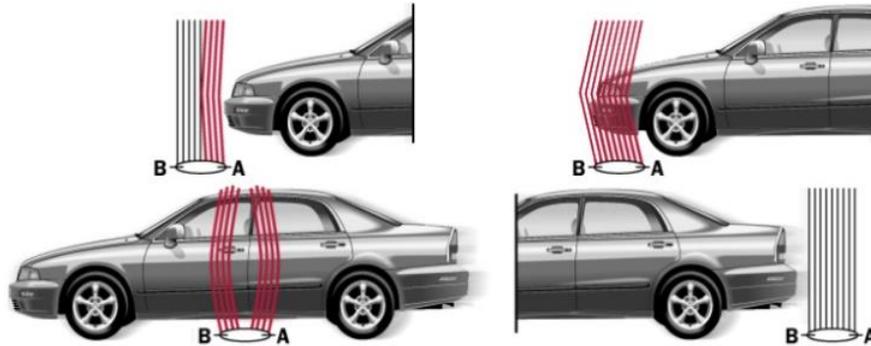


Ilustración 9: tecnología de formación de imagen magnética.

Detalle de algunos modelos:

- Groundhog G-10 [21]: es un analizador de tráfico inalámbrico y permanente que recoge datos sin ningún tipo de sensores externos, lazos, o tubos. El Groundhog es un sensor autónomo colocado dentro del pavimento que utiliza la tecnología de generación imagen magnética para detectar cantidad de vehículos, velocidad y clasificación. También transfiere los datos de forma inalámbrica a un controlador estático junto a la carretera, o puede informar a una estación meteorológica de carretera.

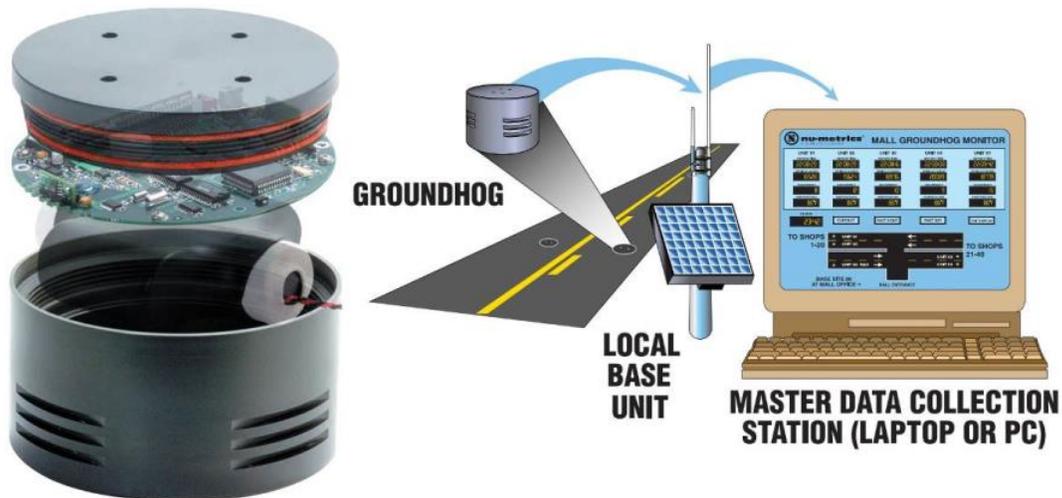


Ilustración 10: esquema de trabajo del Groundhog G-10.

El dispositivo cuenta con una tapa fácilmente desmontable, que permite la extracción rápida de los componentes del sensor durante el mantenimiento de carreteras o el mantenimiento del mismo. Además proporciona datos precisos y esenciales necesarios para el análisis, control y gestión de tráfico eficaz. Cuando se utiliza con un software de gestión de datos inalámbrica, se pueden presentar fácilmente datos históricos en forma de informes, tablas y gráficos.

- NC-100/200™ [21]: está diseñado para proporcionar datos precisos de tamaño, velocidad y clasificación. El sensor se coloca directamente en la vía de circulación para medir los datos y se puede ser instalado y retirado fácil y rápidamente. El modelo NC-100 proporciona la posibilidad de contar solamente, mientras que el modelo NC-200 posee la capacidad de conteo, de medición de velocidad y de clasificación de los vehículos. El analizador de tráfico combina la precisión y portabilidad, monitoreando las condiciones de flujo de tráfico justo donde se necesita. Ya sea un estudio del tráfico en una carretera local, un puente, estacionamiento, área de construcción, o dentro y fuera de los puntos locales de interés, el sensor proporciona datos clave necesarios para un análisis de tráfico eficaz.



Ilustración 11: NC-100/200™ (izquierda). Trabajo de campo (derecha).

#### ❖ **Sistemas no invasivos con tecnología de Radar:**

Son dispositivos basados en tecnología radar que permiten la realización conteos de tráfico, la determinación de las velocidades de los vehículos y su clasificación en función de la longitud de los mismos. Este tipo de sensores se instalan sobre algún poste o valla existentes en el lateral de la carretera. Detalle de algunos modelos:

- Trans-Q [23]: es un radar clasificador de tráfico el cual cumple con la creciente demanda de un dispositivo de medición de tráfico fiable y no intrusivo. El trans-Q proporciona a los ingenieros de tráfico una solución al problema de instalar sensores de tráfico en la superficie de la calzada. El trans-Q es fácil de instalar en el borde de la carretera evitando

perturbar el flujo del tráfico en todo momento. El sensor detecta todos los vehículos que circulan en ambas direcciones, obteniendo de su longitud, velocidad y clasificación. Los datos se pueden recuperar fácilmente, ya sea con una pantalla o una conexión inalámbrica (GPRS o Bluetooth). Características y beneficios:

- Capacidad para dos carriles.
- La tecnología no intrusiva permite analizar el tráfico sin sensores en la calzada.
- Instalación fácil, a un lado de la carretera, sobre un poste o soporte.
- No perturba el flujo de tráfico.
- Reduce el riesgo de seguridad para los trabajadores porque no hay mantenimiento de sensores embebidos sobre la calzada.
- No utiliza tubos, lazos o cadenas.



Ilustración 12: radar Trans-Q en apostado para el análisis.

- Wavetronix Smart-Sensor Series [23]: la serie Smart-Sensor cuenta con tecnología de doble radar que proporciona datos precisos sin necesidad de instalar ningún tipo de sensores en la calzada (ni espiras, ni piezoeléctricos) para recoger y entregar las estadísticas de tráfico. Esto evita cortes de tránsito en rutas o avenidas, permitiendo una puesta en marcha muy rápida y segura para personal. El Smart-Sensor HD recoge la información a través del uso de una señal de radiofrecuencia de 24,125 GHz (banda K) y es capaz de medir el volumen de tráfico, clasificación, velocidad media, velocidad del vehículo individual, de ocupación de carril y la presencia. Clasificado como radar de frecuencia modulada de onda continua (FMCW), detecta y reporta las condiciones del tráfico de forma simultánea sobre un

máximo de diez carriles de tráfico, para el modelo HD125, y veintidós carriles para el modelo HD126. Esto permite la detección de ramas troncales y colectoras de manera simultánea. Otra ventaja, resultante de su doble radar interno, es la detección precisa de los vehículos que circulan cambiando de carril, evitando el doble conteo. Una vez instalado, el proceso de configuración del dispositivo es rápida y fácil. A partir de ese momento, la unidad requiere poco o ningún mantenimiento in situ y puede ser reconfigurado de forma remota para un rendimiento óptimo.

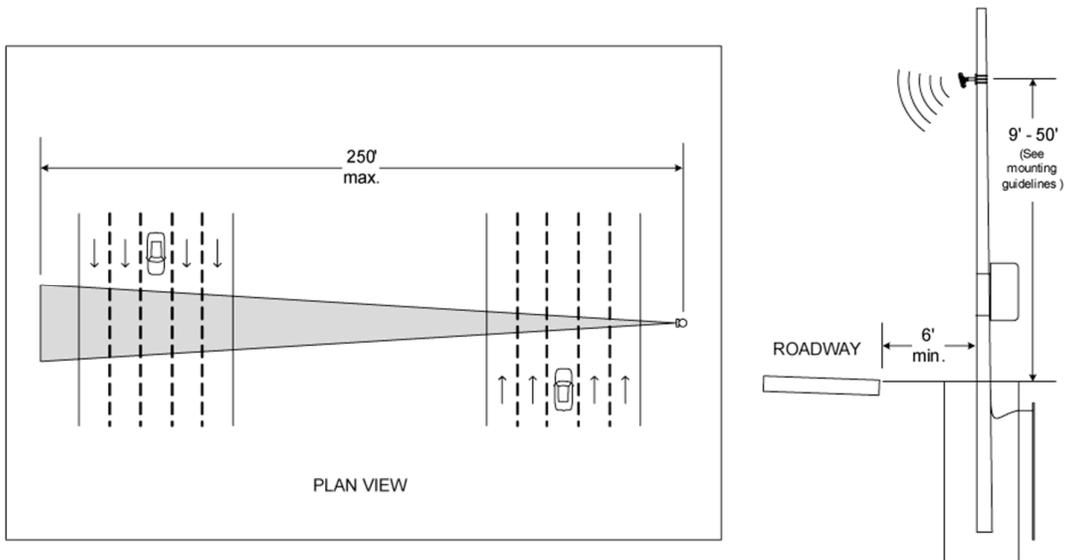


Ilustración 13: esquema de trabajo y posicionamiento del SmartSensor HD.



Ilustración 14: imagen demostrativa del método de trabajo del SmartSensor HD.

❖ **Sistemas inteligentes de video-detección vehicular:**

La video-detección vehicular utiliza algoritmos de tratamiento de imagen a fin de extraer la información pertinentes provenientes de las señales de video de cámaras de vigilancia estándar o compatibles con el protocolo TCP-IP, blanco y negro o color, según la aplicación. De

esta forma permiten detectar vehículos, además de contar el flujo de tránsito y otras variables como la velocidad y la clasificación por longitud.



Ilustración 15: ejemplos de sistemas de video-detección vehicular.

Los algoritmos de procesamiento de imagen desarrollados para este tipo de tratamientos son muy potentes y diseñados para aplicaciones especializadas. Estos, trabajan sobre las más adversas condiciones ambientales, día y noche, pudiendo ser dimensionados a la magnitud que sea requerida. Estos sistemas permiten ser colocados en distintos tramos de la red, autopistas o rutas, en intersecciones semaforizadas, túneles o puentes y cruces peatonales. Detalle de algunos modelos:

- Iteris RZ-4™ Advanced WDR (RZ-4 AWDR) [11]: optimizada para la detección de tráfico por vídeo, la cámara RZ4 AWDR combina el mejor rendimiento en detección de vídeo para todo clima con tecnología de rango dinámico ampliada (WDR) - con ayuda de la tecnología avanzada de imágenes para manejar condiciones de visualización extremas de luz y oscuridad severa. En condiciones de contraluz fuertes, los vehículos pueden ser detectados con un rango dinámico menor a 100 dB; la cámara puede manejar la escena más complicada. La simpleza de la instalación de la cámara RZ-4 AWDR, la compatibilidad hacia atrás, y la capacidad para ajustar la cámara de la cabina ofrecen una solución avanzada y fácil de usar para la detección de vehículos mediante vídeo. La cámara RZ-4 AWDR detecta los vehículos en cualquier condición de iluminación y meteorológica. En contraste con otro tipo de cámara CCTV o de imagen térmica, la cámara RZ-4 AWDR entrega una señal de vídeo que se ha optimizado para su procesamiento por los sistemas de detección de vídeo Vantage [11]. Beneficios y características:
  - Mejora de color y la claridad de la imagen de vídeo, ideal para la conexión a Vantage EdgeConnect™.
  - Conectores de rápido anclaje, una simplificada instalación del montaje ajustable de la cámara y un tiempo de instalación minimizado. No es necesario el uso de crimpeadora.

- Instalación y configuración desde la cámara o desde el suelo.
- La terminación del conector situado en la parte trasera de la cámara simplifica la conexión del cableado.
- Actúa en las condiciones de iluminación más difíciles.
- Un calentador avanzado permite un rendimiento de detección de vídeo óptimo en condiciones climáticas adversas.



Ilustración 16: cámara RZ-4 AWDR.



Ilustración 17: el LAM permite realizar ajustes de campo sobre la configuración de la cámara.

Los sistemas de monitoreo del tráfico anteriormente vistos son variados pero son solo algunos de la gran cantidad de opciones disponibles. Igualmente, dicha selección engloba a la gran mayoría y sus variantes. Cabe destacar que todos estos sistemas recopilan datos de forma dinámica, siguiendo los sucesos a medida que transcurren con mayor o menor detalle, pero desde un punto de vista único. Cuando se analizan las situaciones de este modo se tiene una perspectiva acotada a un espacio geográfico determinado. Este tipo de sistemas otorga una información parcial o específica sobre las problemáticas planteadas en la sección anterior. La realidad es que existen muchas más contingencias para analizar las cuales dichos sistemas no permiten abordar. Los mecanismos que permiten la recopilación de información para el análisis y planificación del tránsito urbano en muchas ocasiones no se encuentran relacionados. Es decir, la información no es cruzada para obtener conclusiones acerca de la interacción de los factores. En consecuencia, las soluciones que se puedan aportar resultarán subóptimas debido a un análisis parcial de la situación. Es necesario ampliar el panorama de análisis.

### **1.3. Propuesta de nuevas técnicas de relevamiento automático vial**

Uno de los objetivos básicos de la *Teoría General de Sistemas* [Karl Ludwig von Bertalanffy, 1950] es el descubrimiento de aquellos componentes, o subsistemas pertenecientes a un sistema mayor, cuyo funcionamiento proporciona una medida real del comportamiento del sistema global. En este caso, si el rendimiento de dichos componentes aumenta, permaneciendo inalterados todos los restantes elementos del sistema global, el rendimiento del funcionamiento del sistema total debe así mismo aumentar. En caso contrario, el componente no está contribuyendo realmente al funcionamiento del sistema. Los sistemas se pueden clasificar basándose en varios factores, por ejemplo, según el tipo de elemento (sistemas naturales o artificiales), según su complejidad (sistemas determinísticos o probabilísticos), según su relación con el entorno (sistemas abiertos o cerrados), etc.

Un análisis cuidadoso de la estructura de una ciudad demuestra que es imposible estudiar cualquiera de sus partes en forma aislada, ya que cada una de ellas se encuentra fuertemente relacionada con los demás elementos de la estructura general. Existen muchos ejemplos que demuestran lo inadecuada que resulta la consideración de las partes de un sistema urbano en forma independiente por lo que sus problemas y soluciones deben abordarse desde la teoría de sistemas, aplicada al ámbito urbano.

Es interesante agregar otro punto de vista al análisis de las problemáticas planteadas anteriormente, que sea dinámico y que no esté sujeto a una ubicación en particular. Una

opción podría ser obtener la información desde el punto de vista del automovilista, es decir, abordar el análisis desde la perspectiva de los protagonistas de los sucesos.

En este trabajo se propone implementar un sistema distribuido de recolección de datos que permita disponer de información actualizada en cuestión de minutos en forma automática y con generación de reportes que permitan la toma de decisiones.

#### **1.4. Descripción del informe**

En el presente informe se detallará el principio de funcionamiento del sistema propuesto, su estructura y una descripción detallada de cada uno de sus componentes. Luego se describirá el diseño y construcción de los dispositivos M1 y M2. También se incluirán una serie de experimentos y cálculos que permitan elaborar un panorama de las aptitudes y los límites que posee el sistema, de modo de comprender de qué es capaz. Por último, se presentará una conclusión y una serie de proposiciones de desarrollo a futuro del sistema propuesto.

## **2. SISTEMA PROPUESTO**

En este capítulo se analizarán el concepto, la estructura y funcionamiento del sistema propuesto, y los distintos bloques funcionales que conforman el dispositivo desarrollado. Por último, se describirán los programas implementados en los microcontroladores que fueron utilizados en los dispositivos M1 y M2. El capítulo se dividirá en tres secciones, la primera correspondiente a la estructura global y específica de cada bloque componente, así como los periféricos utilizados. La segunda parte aborda en detalle los distintos módulos internos del microcontrolador utilizados para el proyecto. La última sección se enfoca en el software implementado en ambos dispositivos, desarrollado para llevar a cabo las aplicaciones principales.

### **2.1. Principio de funcionamiento y componentes del sistema**

El proyecto desarrollado consiste en un sistema automático de registro y transmisión de información útil para el análisis y planificación del tráfico urbano y mantenimiento de las vías. Dicho sistema está constituido por dos tipos diferentes de dispositivos con intercomunicación inalámbrica. Los primeros poseen autonomía y capacidad de movilidad, y los segundos poseen la capacidad de contactar y requerir la información almacenada en los primeros, estando en posiciones fijas y distribuidos de manera estratégica sobre la ciudad. El primer conjunto, denominado como M1, está comprendido por un módulo inteligente, autónomo, automático y geolocalizado (uso de tecnología GPS) que estará ubicado dentro de un vehículo de tránsito continuo, sin recorrido estable, como por ejemplo un remise o taxi. Su función principal es el sensado, análisis y registro de las contingencias del vehículo anfitrión. Dicho dispositivo se encargará de monitorear diversos parámetros y condiciones del automóvil e identificar, mediante algoritmos categorizados de reconocimiento específicos, la presencia o eventual acontecimiento de diversas situaciones predefinidas para luego registrar una serie de datos relevantes a las mismas. Los datos se utilizarán para generar un paquete que se almacenará en una memoria SD. Cada paquete será único y contendrá la siguiente información:

- Identificador del paquete.
- Número del móvil.
- Identificador del tipo de acontecimiento (que fue lo que originó el paquete).
- Hora (hhmmss.sss) y fecha (DDMMAA).
- Las coordenadas geográficas, velocidad y dirección del automóvil.
- Validez de los datos del GPS (si está o no sincronizado con los satélites).

El segundo conjunto de dispositivos, denominados como M2, consiste en un receptor/coordinador que se encarga de buscar y reconocer a cualquiera de los M1 que ingrese dentro de su área de cobertura. El dispositivo coordinador M2 establecerá una comunicación inalámbrica con el dispositivo móvil M1 y le solicitará uno a uno todos los paquetes almacenados en este. Luego los datos recolectados serán subidos a una base de datos on-line, mediante una conexión TCP/IP (Ethernet) con un servidor distante, para su disponibilidad y posterior análisis. La información generada por los dispositivos M1 y recolectada por los M2 será accesible en todo momento desde el servidor. Dicha información será mostrada mediante una página dinámica para su fácil visualización, análisis y uso en la toma de decisiones posteriores.

Cabe destacar, respecto al monitoreo del automóvil que realiza el dispositivo M1, que su fin principal es reconocer diversas situaciones o condiciones específicas que puedan llegar a presentarse. Dichas condiciones pueden ser, por ejemplo, una frenada repentina, un periodo o trayecto prolongado a una velocidad excesivamente reducida o una condición de inmovilidad, patrones característicos de una congestión en el tránsito. También se podría reconocer un movimiento brusco en alguno de los ejes del vehículo, característico de alguna irregularidad en el pavimento como podría ser un bache, excesos de velocidad sobre algún trayecto muy extenso o alguna maniobra brusca e imprudente, entre otras. Es decir, cualquier comportamiento o circunstancia de interés que atente contra la seguridad general o un flujo controlado y uniforme del tránsito. Las condiciones antes mencionadas serán monitorizadas por distintos sensores distribuidos sobre el vehículo, además de los sensores que ya incluyen los automóviles desde su fabricación. El dispositivo M1 poseerá algoritmos lógicos de reconocimiento de patrón que, al presentarse un patrón reconocible en los parámetros sensados, se disparará la orden de generar el paquete de datos correspondiente a dicho patrón. El diseño y naturaleza de los algoritmos dependerá exclusivamente de las contingencias que se quieran sensar e identificar. Por esto, los algoritmos pueden ser variados y actualizables, además de poder agregarse nuevos algoritmos según sean las necesidades del momento. En última instancia, la función de la página dinámica es permitir la visualización, sobre un mapa de la ciudad al estilo del Google Earth™, cada uno de los datos almacenados en la base de datos, distribuidos geográficamente y pudiendo ser seleccionados por tipo de contingencia, fecha y cualquier otro parámetro de interés. El usuario podrá acceder a través de internet y revisar cualquier dato que necesite, siendo posible agrupar o discriminar los datos según corresponda y se requiera. De esta manera se puede lograr un análisis visual y de naturaleza estadística de las condiciones del tránsito sobre toda el área de cobertura de los

dispositivos. A partir del análisis sobre el mapa se podrían reconocer problemáticas para elaborar estrategias de planificación.

Este sistema permitirá planificar de mejor manera el tránsito, realizar más eficientemente el mantenimiento de los canales por los cuales este circula y hacer estudios tanto sobre la ubicación geográfica de los problemas como su distribución temporal. El uso de dispositivos de sensado y recolección de datos en automóviles permite recolectar información desde el punto de vista del conductor, lo que significa agregar una nueva perspectiva dinámica al análisis de las problemáticas, que tradicionalmente se analizan desde lo estático, es decir, desde un punto geográfico específico.

### 2.1.1. Diagrama en bloques del sistema completo

A continuación se presenta un diagrama en bloques del sistema completo:

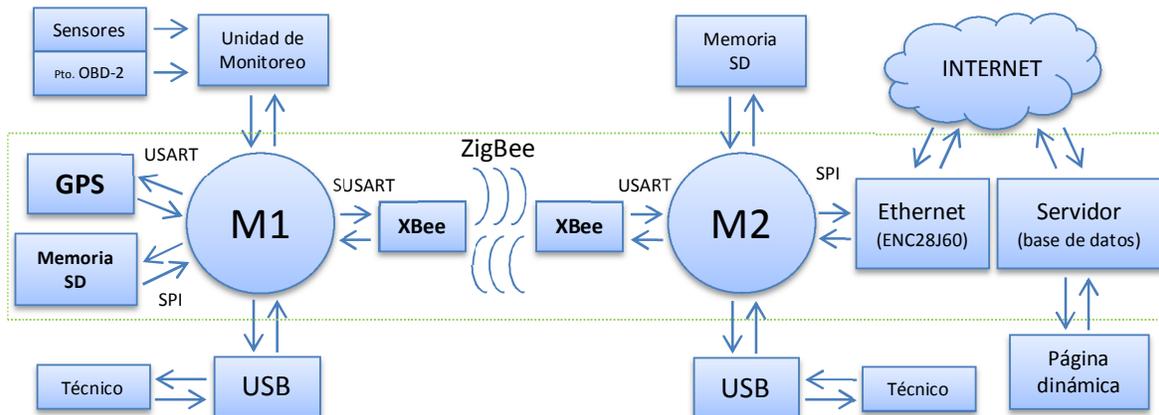


Ilustración 18: diagrama en bloques del sistema completo.

Dada la complejidad del sistema y los tiempos que el mismo requeriría, este proyecto final de grado se limitó a abarcar los bloques encerrados por el recuadro punteado verde de la Ilustración 18. De esta manera queda para un trabajo futuro el desarrollo de la Unidad de Monitoreo, las interfaces USB, la implementación de una memoria SD en el dispositivo M2 y la creación de la una página dinámica para visualizar los datos almacenados en la base de datos.

### 2.1.2. Bloque de posicionamiento global

La tecnología de posicionamiento global está provista por el dispositivo *EM-411* de la empresa *GlobalSat*, [www.globalsat.com.tw](http://www.globalsat.com.tw). El GPS EM-411 es un componente de bajo costo, pero de óptima fidelidad y precisión, el cual cuenta con una antena de conexión integrada que permite una implementación completa. Algunas de sus características son:

- Circuito integrado GPS SiRF Star III [16] de alto rendimiento.

- Supercondensador [25] integrado de respaldo de última adquisición.
- TTF (Time To First Fix) extremadamente corto para bajos niveles de señal.
- Procesamiento en paralelo de hasta 20 satélites a la vista.
- Sensibilidad de seguimiento: -159dBm.
- Soporta protocolo NMEA 0183 [16].
- Antena incorporada.
- 100ms de readquisición.
- Rendimiento mejorado en zonas urbanas.
- Soporte de sistema WAAS y EGNOS [24].



El EM-411 posee un rango de tensión de alimentación de 4,5V a 6,5V de continua. El nivel de consumo del GPS es de entre 35mA y 60mA, con una corriente de back-up de 10uA. Para los niveles de protocolo de comunicación utiliza niveles TTL de entre 0V y 2,85V. La interfaz de comunicación es a través del protocolo serie asincrónico RS-232, utilizando mensajes de salida NMEA 0183 GGA, GSA, GSV, RMC, VTG y GLL. Permite manejar velocidades (configurables) de 1200 bps hasta 38400 bps, siendo la configuración por defecto 4800 bps.

Los **datos más importantes** que se pretenden obtener del GPS son:

- Latitud y Longitud (con una precisión de hasta 5 metros).
- Fecha y hora (UTC).
- Velocidad y dirección del desplazamiento.

Para este proyecto se utilizó el mensaje RMC (*RMC-Recommended Minimum Specific GNSS Data*), el cual posee todos los datos necesarios para completar el paquete generado por el dispositivo M1. El mismo consta de la siguiente información:

```
$GPRMC,161229.487,A,3723.2475,N,12158.3416,W,10.13,309.62,120598,*10<CR><LF>
```

DATO	NOMBRE	DESCRIPCIÓN
\$GPRMC	Message ID (\$GPxxx)	Cabecera del protocolo RMC
161229.487	Hora UTC	hhmmss.sss
A	Estado	A: válido, V: inválido
3723.2475	Latitud	ddmm.mmmm
N	Emisferio	N: norte, S: sur
12158.3416	Longitud	dddmm.mmmm
W	Emisferio	E: este, W: oeste
10.13	Velocidad sobre el suelo	nudos
309.62	Dirección (Norte = 0º)	grados
120598	Fecha	DDMMAA

Tabla 4: descripción del mensaje RMC.

Es importante destacar que los datos están separados por comas y que si algún dato no se encuentra disponible, su correspondiente espacio quedará vacío. En tal caso, el carácter de

estado tendrá el valor “V”. Para poder enviarle comandos al GPS se debe hacer uso del siguiente formato de trama:

$\$PSRF103,<msg>,<mode>,<rate>,<cksumEnable>*CKSUM<CR><LF>$

DATO	DESCRIPCIÓN
\$PSRF103	Cabecera de control
<msg>	0=GGA, 1=GLL, 2=GSA, 3=GSV, 4=RMC, 5=VTG
<mode>	0 = frecuente, 1 = a demanda
<rate>	Dato cada <rate> segundos. Off = 0, max = 255
<cksumEnable>	0= deshabilitado, 1= habilitado

Tabla 5: descripción del mensaje de control del GPS.

Para solicitar los datos del GPS el dispositivo M1 utiliza la siguiente trama:

$\$PSRF103,04,01,00,01*21\r\n$

El GPS, al recibir dicha trama, interpreta que inmediatamente debe enviar un único paquete RMC.

### 2.1.3. Bloque de almacenamiento de datos permanente

El sistema de almacenamiento de datos está constituido por un módulo lector de tarjetas SD sobre la cual se guardaban los datos en un archivo .csv (tipo Excel) sobre un sistema de archivos FAT32. En la Ilustración 19 se muestra una imagen del módulo mencionado.



Ilustración 19: lector de tarjetas SD/microSD.

Para implementar el sistema de archivos FAT32 y el control de la tarjeta SD se utilizó la librería “Memory Disk Drive File System for PIC18 / PIC24 / dsPIC / PIC32” desarrollada por la empresa *Microchip* [12]. La misma fue modificada para hacer uso solo de las funcionalidades necesarias para el proyecto y adaptada al microcontrolador utilizado en el dispositivo M1. El módulo posee una interface de comunicación SPI y un conmutador de niveles para poder trabajar con 3,3V o 5V.

#### 2.1.4. Bloque de comunicación inalámbrica

El dispositivo de comunicación inalámbrico elegido fue el *XBeePRO Series 2* de la empresa *Digi International*, [www.digi.com](http://www.digi.com). Este dispositivo opera con el estándar de comunicaciones *ZigBee* [19]. Los módulos de comunicación *XBee* son utilizados por los dispositivos M1 y M2 como la interfaz de transmisión/recepción inalámbrica del sistema. Este dispositivo es de suma importancia para el proyecto, debido a que sus características de funcionamiento permitieron la idealización y realización de todo el sistema. Por eso, se hará un hincapié especial sobre las características de este componente del sistema. En la Ilustración 20 se muestra el hardware del *XBee PRO Series 2*.

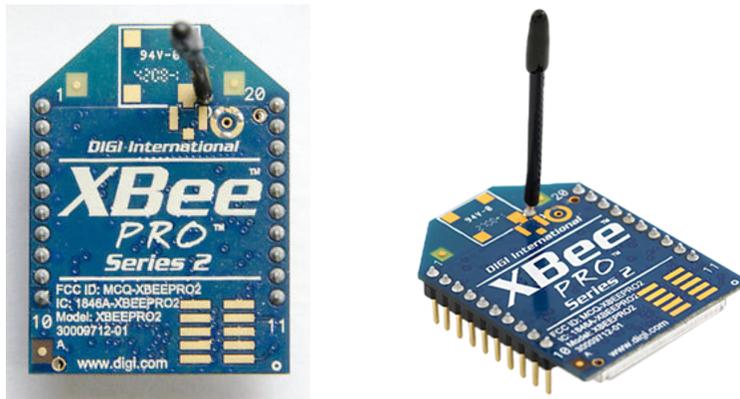


Ilustración 20: XBee PRO Series 2 de DIGI International.

ZigBee es un estándar de comunicaciones inalámbricas diseñado por la *ZigBee Alliance* [19]. ZigBee está basado en el estándar *IEEE 802.15.4* [8] de redes inalámbricas de área personal (WPAN, *Wireless Personal Area Network*) y tiene como objetivo las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías. ZigBee opera en las bandas libres ISM (*Industrial, Scientific & Medical*) de 2.4 GHz, 868 MHz (Europa) y 915 MHz (Estados Unidos). Para el rango de 2.4GHz, se definen hasta 16 canales, cada uno de ellos con un ancho de banda de 5 MHz. Los modelos *XBeePRO S2* de DIGI tienen una velocidad de transmisión de datos de 250 Kbps y hasta 35 Kbps de información. Posee un rango de cobertura de 90 metros para ambientes internos o que presenten obstáculos y de hasta 1500 metros en ambientes externos o libres de obstáculos. Los datos transmitidos son codificados usando la técnica de *espectro esparcido de secuencia directa* [9] utilizando un código pseudo-aleatorio de 32 bits (8Mchip/s). La modulación consiste en MSK con 16 símbolos ortogonales (4 bits/símbolo) a 2MBaud/s. A pesar de coexistir en la misma frecuencia con otro tipo de redes como WiFi o Bluetooth su desempeño no se ve afectado, debido a su baja tasa de transmisión y a características propias del estándar *IEEE 802.15.4*.

Los módulos XBee poseen la capacidad de operar en redes de gran densidad. Esta característica ayuda a aumentar la confiabilidad de la comunicación, ya que entre más nodos existan dentro de una red, entonces, existirán mayor número de rutas alternativas para garantizar que un paquete llegue a su destino. Cada red ZigBee tiene un **identificador de red único**, lo que permite que **coexistan varias redes en un mismo canal** de comunicación sin ningún problema. ZigBee es un protocolo de comunicación **multi-salto**, es decir, que **se puede establecer comunicación entre dos nodos aun cuando estos se encuentren fuera del rango de transmisión**, siempre y cuando existan otros nodos intermedios que los interconecten. De esta manera, se incrementa el área de cobertura de la red.

Existen dos modos de operación, el modo AT y el modo API. En modo AT o modo transparente la comunicación es directa y en modo caracter: se envían los caracteres al módulo XBee y este los transmite a la dirección que tiene programada como destino. En cambio, en modo API o modo de Interfaz de Programación de Aplicaciones, la transmisión se realiza por medio de tramas de datos en los cuales la dirección destino, si la hubiera, es parte de la trama.

Se definen tres tipos diferentes de dispositivos ZigBee según su papel en la red:

- **Coordinador ZigBee** (*ZigBee Coordinator, ZC*). El tipo de funcionamiento más completo. Puede actuar como director de una red en árbol así como servir de enlace a otras redes. Existe exactamente un coordinador por cada red, que es el nodo que la comienza en principio. Puede almacenar información sobre la red y actuar como su *centro de confianza* en la distribución de claves de cifrado.
- **Router ZigBee** (*ZR*). Además de ofrecer un nivel de aplicación para la ejecución de código de usuario, puede actuar como router interconectando dispositivos separados en la topología de la red.
- **Dispositivo final** (*ZigBee End Device, ZED*). Posee la funcionalidad necesaria para comunicarse con su nodo maestro (el coordinador o un router), pero no puede retransmitir información destinada a otros dispositivos. De esta forma, este tipo de nodo puede estar dormido la mayor parte del tiempo, aumentando la vida media de sus baterías. Un ZED tiene requerimientos mínimos de memoria y es por tanto significativamente más barato.

Para la programación de los módulos XBee se utilizó el software de programación *X-CTU*, de la empresa *Digi International*. Mediante una placa de programación (adaptador USB-Serie) de la empresa *ITyT* ([www.ityt.com.ar](http://www.ityt.com.ar)) utilizada como interface entre los módulos XBee y la PC fue posible configurar y realizar pruebas de campo con los módulos. En la Ilustración 21 puede verse la placa utilizada para programar los módulos XBee. Más abajo, en la Ilustración 22,

pueden verse unas capturas del software X-CTU. A la izquierda se puede ver la solapa de programación, donde se configuran los bits de configuración. A la derecha, una secuencia de recepción en tiempo real de una trama API enviada por otro módulo XBee.



Ilustración 21: interface USB de programación para XBee's.

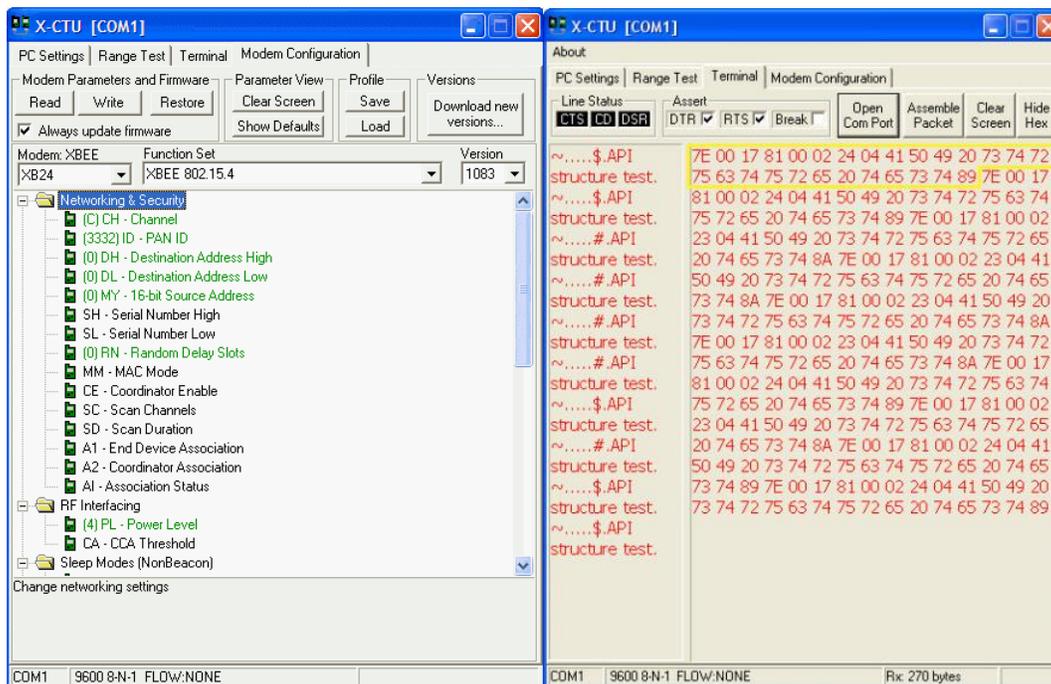


Ilustración 22: capturas de pantalla del software de programación X-CTU de Digi International.

### 2.1.5. Bloque Ethernet

El sistema de conexión para redes de área local está constituido por el módulo Ethernet *ENC28J60* de la empresa *Microchip* [12]. Trabajando con una interface SPI, es utilizado para establecer una conexión TCP (socket) con un servidor y poder enviar la información que luego será subida a una base de datos por dicho servidor. El sistema de conexión para redes de área local se implementó a partir de la librería “*TCP/IP Stack Software*” desarrollada por la empresa *Microchip* [12]. La misma fue modificada para hacer uso solo de las funcionalidades necesarias para el proyecto y adaptada al microcontrolador utilizado en el dispositivo M2. En la Ilustración 23 se muestra una imagen del módulo *ENC28J60*.

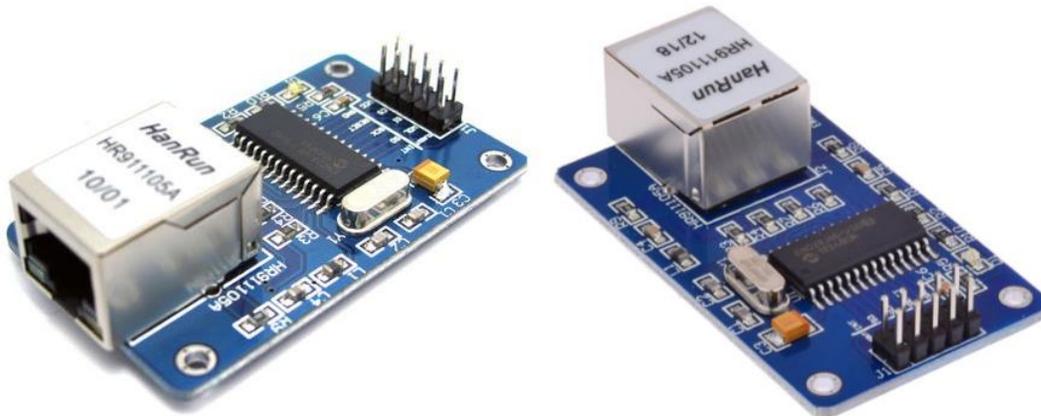


Ilustración 23: módulo Ethernet *ENC28J60* de la empresa *Microchip*.

### 2.1.6. Servidor y base de datos

El bloque encargado de responder a la solicitud del *ENC28J60* para establecer una conexión TPC a través de Internet y recibir los datos para luego almacenarlos en una base de datos está formado por un servidor WAMP 2.2 [22]. El mismo incluye el servidor HTTP *Apache* 2.2.22 [18] de código abierto y el gestor de bases de datos *MySQL* 5.5.24 [13], entre otras herramientas. En el servidor se aloja una página con código PHP la cual se encarga de recibir la trama enviada por el *ENC28J60*, discriminar los datos y luego enviarlos al gestor *MySQL* para que los almacene en una base de datos. La trama enviada por el *ENC* es la siguiente:

```
http://.../m2/sendMySQL.php?nPaq=±32767&n_movil=999&ID_paq=SOS&Hora=hmmss.sss&Fecha=12  
0598&Lat=3723.2475&NS=N&Lon=12158.3416&EO=W&Vel=00.13&Direccion=309.62&Valid=W
```

En la trama figura el enlace al servidor (representado por los tres puntos), el nombre del archivo PHP al cual se le envía la trama (“*sendMySQL.php*”) y cada una de las variables con sus valores correspondientes (*nPaq*, *n\_movil*, *ID\_paq*, *Hora*, *Fecha*, *Lat*, *NS*, *Lon*, *EO*, *Vel*, *Direccion* y *Valid*). El código presente en “*sendMySQL.php*” es el siguiente:

```
<?php  
$conexion=mysqli_connect("localhost","root","","manu")
```

```

or die("C");
mysqli_query($conexion,"insert into
paquetes(nPaq,n_movil,ID_paq,Hora,Fecha,Lat,NS,Lon,EO,Vel,Direccion,Valid)
values(
'$_REQUEST[nPaq]',
'$_REQUEST[n_movil]',
'$_REQUEST[ID_paq]',
'$_REQUEST[Hora]',
'$_REQUEST[Fecha]',
'$_REQUEST[Lat]',
'$_REQUEST[NS]',
'$_REQUEST[Lon]',
'$_REQUEST[EO]',
'$_REQUEST[Vel]',
'$_REQUEST[Direccion]',
'$_REQUEST[Valid]'
)")
or die("S");
mysqli_close($conexion);
echo "Y";
// C = problemas de conexión, S = problemas en el Select, Y = todo OK.
?>

```

En la base de datos, los datos almacenados se pueden visualizar de la siguiente manera:

The screenshot shows the phpMyAdmin interface for a MySQL database named 'paquetes'. The SQL query executed is: `SELECT * FROM paquetes LIMIT 0, 30`. The results are displayed in a table with the following columns: `cod`, `nPaq`, `n_movil`, `ID_paq`, `Hora`, `Fecha`, `Lat`, `NS`, `Lon`, `EO`, `Vel`, `Direccion`, and `Valid`. The table contains 8 rows of data, each with a set of icons for editing and deleting records.

cod	nPaq	n_movil	ID_paq	Hora	Fecha	Lat	NS	Lon	EO	Vel	Direccion	Valid
136	27	666	SOS	171142.057	170816	ddmm.mmmm	V	dddmm.mmmm	H	nn.nn	ddd.dd	X
140	0	666	SOS	193710.059	170816	ddmm.mmmm	V	dddmm.mmmm	H	nn.nn	ddd.dd	X
141	1	666	SOS	193710.059	170816	ddmm.mmmm	V	dddmm.mmmm	H	nn.nn	ddd.dd	X
142	2	666	SOS	193712.058	170816	ddmm.mmmm	V	dddmm.mmmm	H	nn.nn	ddd.dd	X
143	3	666	SOS	193710.059	170816	ddmm.mmmm	V	dddmm.mmmm	H	nn.nn	ddd.dd	X
144	4	666	SOS	193900.059	170816	ddmm.mmmm	V	dddmm.mmmm	H	nn.nn	ddd.dd	X
145	5	666	SOS	193920.059	170816	ddmm.mmmm	V	dddmm.mmmm	H	nn.nn	ddd.dd	X
146	6	666	SOS	193932.058	170816	ddmm.mmmm	V	dddmm.mmmm	H	nn.nn	ddd.dd	X

Ilustración 24: captura de pantalla de una base de datos de MySQL.

### 2.1.7. Dispositivos M1 y M2

El componente principal de los dispositivos M1 y M2 está constituido por el microcontrolador *PIC18F27J53*, fabricado por la empresa *Microchip* [12]. En los PICs se encuentra cargado el programa principal de cada dispositivo.

El dispositivo M1 está constituido por los siguientes componentes y sus respectivos niveles de alimentación:

- Dispositivo microcontrolador PIC18F27J53 (3,3V).
- Dispositivo GPS (5V). Interfaz serie RS232 con niveles TTL de 2.85V.
- Módulo lector de memorias SDs (conmutable entre 3,3V y 5V). Interface SPI.
- Módulo XBee (3,3V). Interface serie RS232 con niveles TTL de 3,3V.
- Fuente de alimentación fija de 5V y variable ajustada a 3,3V.

El dispositivo M2 está constituido por los siguientes componentes y sus respectivos niveles de alimentación:

- Dispositivo microcontrolador PIC18F27J53 (3,3V).
- Dispositivo ENC28J60 (3,3V). Interface SPI.
- Dispositivo XBee (3,3V). Interface serie RS232 con niveles TTL de 3,3V.
- Fuente de alimentación fija de 3,3V.

Como se mencionó anteriormente, este proyecto final de grado se limitó a abarcar los bloques encerrados por el recuadro punteado verde de la Ilustración 18. En el dispositivo M1, el bloque de Unidad de Monitoreo se sustituyó por un pulsador, el cual es accionado a voluntad, simulando la señal de disparo que enviaría dicho bloque al bloque M1 para que se almacene el respectivo paquete de datos. Cada pulsación genera un paquete y el mismo se almacena automáticamente en la memoria SD. Las interfaces USB simplemente se ignoraron y el bloque de almacenamiento del dispositivo M2 no fue necesario desarrollarlo ya que alcanzaba con desarrollar el bloque del dispositivo M1.

## **2.2. Detalle del microcontrolador**

El microcontrolador elegido para el proyecto fue el *PIC18F27J53* de la empresa *Microchip* [12]. El PIC18F27J53 es un microcontrolador de 8 bits, bajo consumo, alto rendimiento y con interface USB 2.0 full-speed integrada. Es ideal para aplicaciones que requieren soluciones USB de bajo costo con alta capacidad de memoria de programa y un robusto set de periféricos en un encapsulado mínimo. El mismo puede funcionar con dos modos distintos de oscilador externo y trabajar a una frecuencia de hasta 48MHz. Permite la utilización de cristales externos de hasta 16MHz y posee un PLL interno para incrementar o dividir esa frecuencia. Posee un encapsulado de 28-Pines tipo SPDIP/SOIC/SSOP y la capacidad de remapeo de varios periféricos sobre 16 de los 28 pines. Esto lo dota de una gran versatilidad para operar la variada cantidad de puertos de comunicación y lógicos sobre los periféricos externos

requeridos. En las Ilustración 25 y Ilustración 26 se muestra un diagrama en bloques interno y la distribución de pines del PIC18F27J53.

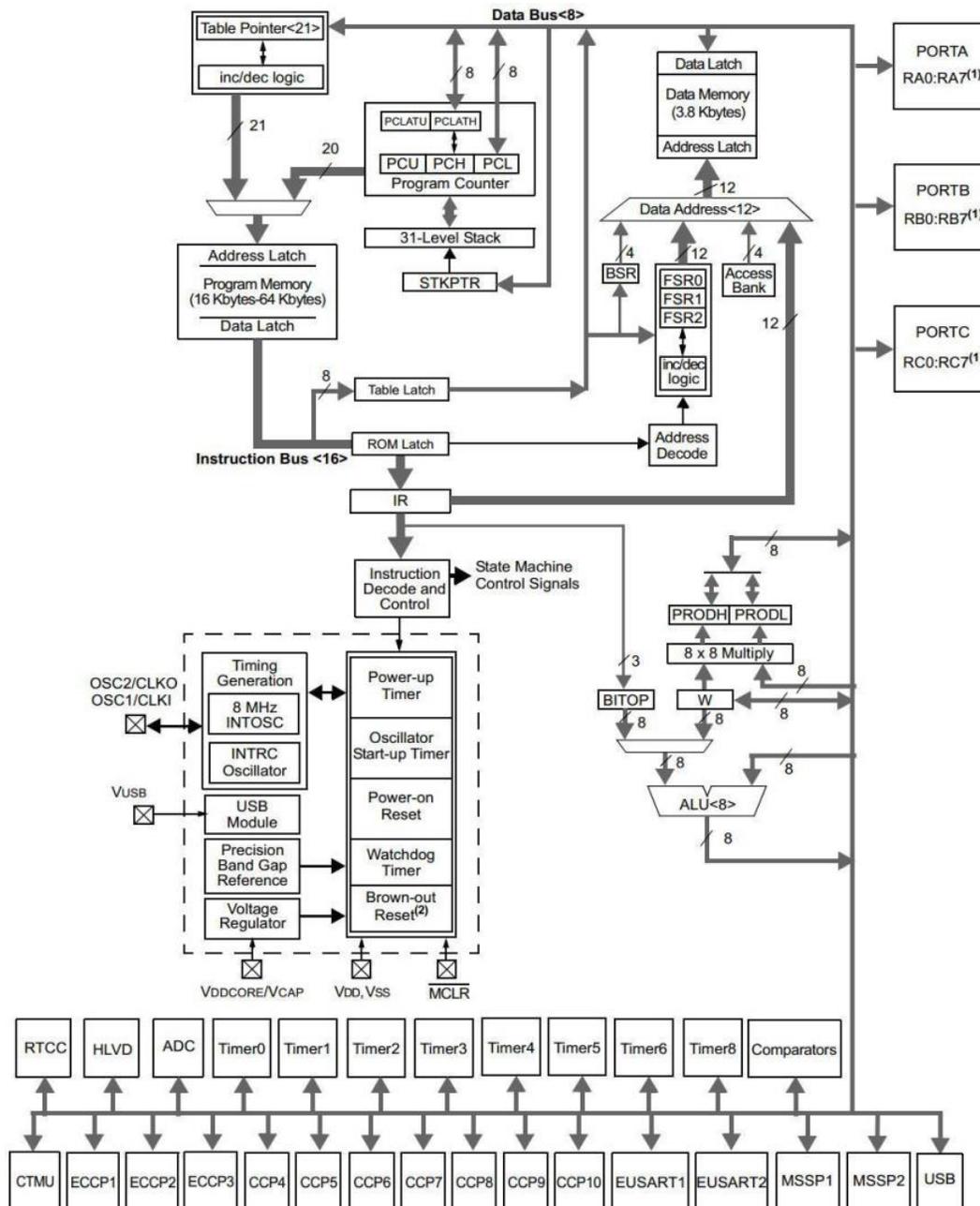


Ilustración 25: diagrama en bloques del PIC18F27J53 de la empresa Microchip.

Sus características principales son las siguientes:

- Consumo de 9 nA en modo *Deep Sleep* (*Watch Webinar*).
- Consumo de 200nA en modo *Sleep* (*RAM retention*).
- Consumo de 700nA en operación *Real Time Clock* y Calendario en modo *Sleep*.
- Consumo de 330nA con *Watch Dog Timer* activo en modo *Sleep*.

- USB 2.0 *full-speed* integrado.
- *Peripheral Pin Selection* para mapear periféricos digitales sobre varias I/O para una mayor flexibilidad de diseño.
- RTCC (*Real Time Clock Calendar*) provee función de reloj, calendario y alarma.
- CTMU (*Charge Time Measurement Unit*) permite pantallas *touch* capacitivas.
- 3 módulos Captura / Comparación / PWM mejorados.
- 2 MSSP (*Master Synchronous Serial Port*) para comunicación SPI o I2C™.
- 2 módulos USART mejorados.
- 3 comparadores analógicos.
- 10 canales de 12-bit ADC (*Analog-To-Digital Converter*).
- Memoria *Flash* de programa soporta 10.000 ciclos de escritura y 20 años de retención.
- Voltaje de operación entre 2.0 y 3.6V, entradas digitales con tolerancia de hasta 5.5V.

La razón de haberlo elegido fue el hecho de que permite utilizar dos interfaces USART independientes, *Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART)*, una de ellas con la posibilidad de remapear sus terminales entre los pines que dispone el microcontrolador para esta funcionalidad especial. A demás, posee dos interfaces serie sincrónicas, *Master Synchronous Serial Port (MSSP) Module*, para comunicación entre periféricos, pudiendo una de las mencionadas ser remapeada. También posee la posibilidad de trabajar con una interface estándar USB 2.0. Estas características fueron ideales para este proyecto.

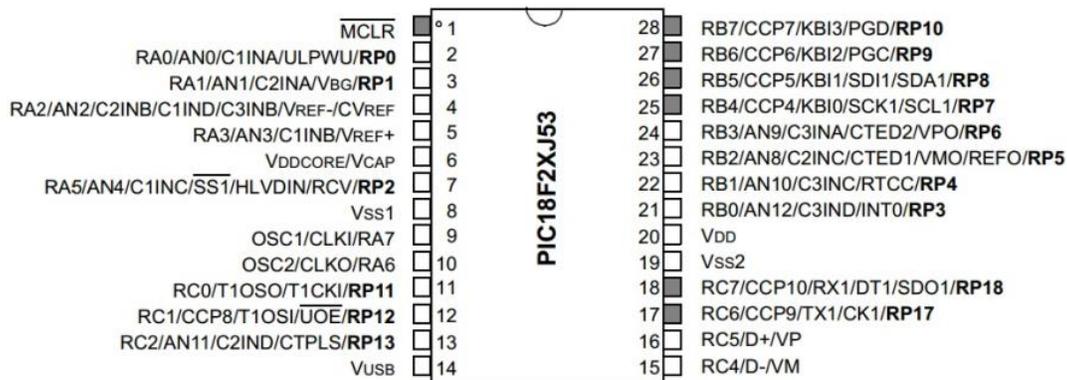


Ilustración 26: diagrama de pines del microcontrolador PIC18F27J53.

En un principio se pensaba utilizar, en el dispositivo M1, el módulo interno EUSART 1 para comunicarse con el módulo XBee y el EUSART2 (remapeable) para comunicarse con el GPS. Luego, se usaría la interface SPI2 (remapeable) para comunicarse con el módulo de almacenamiento (lector de tarjetas SD). Pero esto no fue posible porque hubo inconvenientes con la interface SPI2 la cual, por alguna razón desconocida, no podía procesar la recepción de

información en su terminal MISO (*Master Input Slave Output*). Este problema se presentó tanto en el microcontrolador utilizado en el dispositivo M1 como en M2 (idénticos). Debido a esto fue necesario hacer uso del módulo SPI1 para la comunicación con el lector de tarjetas SD. Aquí surgió un nuevo problema. La interface SPI1 y EUSART1 tienen sus terminales asignados por hardware, siendo los mismos NO remapeables. Tanto el terminal RX1 (receptor del EUSART1) como el SDO1 (*digital out del SPI1*) están asignados al puerto RC7, haciendo de esta manera inviable la posibilidad de usar ambos módulos al mismo tiempo. Debido a ese inconveniente se optó por usar el módulo interno SPI1 e implementar una interface USART por software, a la cual se mencionará de ahora en más como SUSART (*SoftwareUSART*). Finalmente, en el dispositivo M1, se utilizó el módulo SPI1 para comunicarse con el lector de tarjetas SD, el módulo EUSART2 para comunicarse con el GPS y una interface SUSART para comunicarse con el módulo XBee. Respecto al dispositivo M2, se utilizó el módulo SPI1 para comunicarse con el módulo ENC28J60 y el módulo EUSART2 para comunicarse con el módulo XBee.

En las siguientes subsecciones se describirá con mayor detalle tanto los periféricos internos del microcontrolador como los registros utilizados en este proyecto.

### **2.2.1. Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART)**

El microcontrolador cuenta con dos módulos SCI (*Serial Communications Interface*), independientes, comúnmente conocidos como USART. Los módulos SCI del PIC18F27J53 pueden ser configurados como un sistema asincrónico *full-duplex* pudiendo comunicarse con diversos periféricos y computadoras personales. También pueden ser configurados en modo *half-duplex* síncrono para comunicarse con dispositivos como conversores A/D o D/A, EEPROMs seriales, entre otros. El módulo EUSART implementa características adicionales, incluyendo detección automática de *baud-rate* y calibración, *wake-up* automático al recibir un *Sync Break* (equivalente a una señal de *Wake-up* según el protocolo *LIN/J2602* [17]) y transmisión de un carácter *Break* de 12-bits. Esto lo hace ideal para su uso en un sistema con bus *LIN/J2602* (*Local Interconnect Network*). El segundo módulo SCI, el EUSART2, tiene la característica de poseer terminales remapeables, es decir, asignar sus terminales a los puertos de entrada/salida disponibles para esta funcionalidad. A continuación se muestran las características y modos que ambos módulos poseen:

- Asincrónico (*full-duplex*) con:
  - *Auto-wake-up* al recibir un caracter.
  - Auto calibración de *baudrate*.
  - Transmisión de caracter *Break* de 12-bit.

- Sincrónico – Master (*half-duplex*) con selección de polaridad de clock.
- Sincrónico – Slave (*half-duplex*) con selección de polaridad de clock.
- *Full-duplex*, Formato estándar *non-return-to-zero* (NRZ).
- Transmisor y receptor con *buffer* y *enable* independientes.
- Transmisor con *open-colector* opcional (*push-pull* CMOS por defecto).
- Generador de *Baud Rate* de 8 y 16bits.
- Posibilidad de funcionamiento mediante interrupciones:
  - Registro de transmisión completa.
  - Registro de recepción de datos lleno.
  - *Overrun* de recepción, error de paridad, error de *framing*, error por ruido.
- Hardware de generación y comprobación de paridad.
- Longitud de caracteres de 8 ó 9 bits.
- Polaridad de salida del transmisor seleccionable.

Ambos módulos cuenta con tres registros que permiten configurar la totalidad de los parámetros de los módulos. A continuación se los describirá en detalle:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Tabla 6: descripción de la leyenda de los registros.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
<b>CSRC</b>	<b>TX9</b>	<b>TXEN</b>	<b>SYNC</b>	<b>SENDB</b>	<b>BRGH</b>	<b>TRMT</b>	<b>TX9D</b>
bit 7						bit 0	

Tabla 7: registro TXSTAx (Transmit Status And Control Register).

- bit 7 **CSRC**: Clock Source Select bit.  
Asynchronous mode: don't care.  
Synchronous mode:  
 1 = Master mode (clock generated internally from BRG).  
 0 = Slave mode (clock from external source).
- bit 6 **TX9**: 9-Bit Transmit Enable bit.  
 1 = Selects 9-bit transmission.  
 0 = Selects 8-bit transmission
- bit 5 **TXEN**: Transmit Enable bit(1).  
 1 = Transmit is enabled.  
 0 = Transmit is disabled.

- bit 4 **SYNC**: EUSART Mode Select bit.  
1 = Synchronous mode.  
0 = Asynchronous mode.
- bit 3 **SENDB**: Send Break Character bit.  
Asynchronous mode:  
1 = Send Sync Break on next transmission (cleared by hardware upon completion).  
0 = Sync Break transmission completed.  
Synchronous mode: don't care.
- bit 2 **BRGH**: High Baud Rate Select bit.  
Asynchronous mode:  
1 = High speed.  
0 = Low speed.  
Synchronous mode: unused in this mode.
- bit 1 **TRMT**: Transmit Shift Register Status bit.  
1 = TSR is empty.  
0 = TSR is full.
- bit 0 **TX9D**: 9th bit of Transmit Data.  
Can be address/data bit or a parity bit.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
<b>SPEN</b>	<b>RX9</b>	<b>SREN</b>	<b>CREN</b>	<b>ADDEN</b>	<b>FERR</b>	<b>OERR</b>	<b>RX9D</b>
bit 7							bit 0

Tabla 8: registro RCSTAx (Receive Status And Control Register)

- bit 7 **SPEN**: Serial Port Enable bit.  
1 = Serial port enabled (configures RXx/DTx and TXx/CKx pins as serial port pins).  
0 = Serial port disabled (held in Reset).
- bit 6 **RX9**: 9-Bit Receive Enable bit.  
1 = Selects 9-bit reception.  
0 = Selects 8-bit reception.
- bit 5 **SREN**: Single Receive Enable bit.  
Asynchronous mode: don't care.  
Synchronous mode – Master:  
1 = Enables single receive.

0 = Disables single receive.

This bit is cleared after reception is complete.

Synchronous mode – Slave: don't care.

- bit 4 **CREN**: Continuous Receive Enable bit.

Asynchronous mode:

1 = Enables receiver.

0 = Disables receiver.

Synchronous mode:

1 = Enables continuous receive until enable bit, CREN, is cleared (CREN overrides SREN).

0 = Disables continuous receive.

- bit 3 **ADDEN**: Address Detect Enable bit.

Asynchronous mode 9-Bit (RX9 = 1):

1 = Enables address detection, enables interrupt and loads the receive buffer when RSR<8> is set.

0 = Disables address detection, all bytes are received and the ninth bit can be used as a parity bit.

Asynchronous mode 9-Bit (RX9 = 0): don't care.

- bit 2 **FERR**: Framing Error bit.

1 = Framing error (can be updated by clearing the RCREGx register and receiving the next valid byte).

0 = No framing error.

- bit 1 **OERR**: Overrun Error bit.

1 = Overrun error (can be cleared by clearing bit, CREN).

0 = No overrun error.

- bit 0 **RX9D**: 9th bit of Received Data.

This can be address/data bit or a parity bit and must be calculated by user firmware.

R/W-0	R-1	R/W-0	R/W-0	R/W-0	U-0	R-0	R-x
<b>ABDOVF</b>	<b>RCIDL</b>	<b>RXDTP</b>	<b>TXCKP</b>	<b>BRG16</b>	-	<b>WUE</b>	<b>ABDEN</b>
bit 7							bit 0

Tabla 9: registro BAUDCONx (Baud Rate Control Register).

- bit 7 **ABDOVF**: Auto-Baud Acquisition Rollover Status bit.

1 = A BRG rollover has occurred during Auto-Baud Rate Detect mode (must be cleared in software).

0 = No BRG rollover has occurred.

- bit 6 **RCIDL**: Receive Operation Idle Status bit.
  - 1 = Receive operation is Idle.
  - 0 = Receive operation is active.
- bit 5 **RXDTP**: Data/Receive Polarity Select bit.
  - Asynchronous mode:
    - 1 = Receive data (RXx) is inverted (active-low).
    - 0 = Receive data (RXx) is not inverted (active-high).
  - Synchronous mode:
    - 1 = Data (DTx) is inverted (active-low).
    - 0 = Data (DTx) is not inverted (active-high).
- bit 4 **TXCKP**: Synchronous Clock Polarity Select bit.
  - Asynchronous mode:
    - 1 = Idle state for transmit (TXx) is a low level.
    - 0 = Idle state for transmit (TXx) is a high level.
  - Synchronous mode:
    - 1 = Idle state for clock (CKx) is a high level.
    - 0 = Idle state for clock (CKx) is a low level.
- bit 3 **BRG16**: 16-Bit Baud Rate Register Enable bit.
  - 1 = 16-bit Baud Rate Generator – SPBRGHx and SPBRGx.
  - 0 = 8-bit Baud Rate Generator – SPBRGx only (Compatible mode), SPBRGHx value is ignored.
- bit 2 **Unimplemented**: Read as '0'.
- bit 1 **WUE**: Wake-up Enable bit.
  - Asynchronous mode:
    - 1 = EUSART will continue to sample the RXx pin – interrupt is generated on the falling edge; bit is cleared in hardware on the following rising edge.
    - 0 = RXx pin is not monitored or the rising edge detected.
  - Synchronous mode: unused in this mode.
- bit 0 **ABDEN**: Auto-Baud Detect Enable bit.
  - Asynchronous mode:
    - 1 = Enable baud rate measurement on the next character; requires reception

of a Sync field (55h); cleared in hardware upon completion.

0 = Baud rate measurement disabled or completed.

Synchronous mode: unused in this mode.

Para poder establecer una comunicación a través de un enlace serie es necesario configurar correctamente la velocidad de transmisión dada en bits, o *bauds*, por segundo (bps). Para esto, el módulo SCI cuenta con un *Baud Rate Generator* (BRG) de 8 y 16 bits, lo que permite configurar distintas velocidades de acuerdo a lo requerido. Este generador es independiente para cada una de las interfaces disponibles en el microcontrolador, es decir que pueden trabajar a velocidades distintas tanto en modo sincrónico como asincrónico. Setenado el bit BRG16 (BAUDCONx<3>) se selecciona el modo 16 bit. Los registros SPBRGHx y SPBRGx controlan el periodo de un *free-running timer*. En modo asincrónico, los bits BRGH (TXSTAx<2>) y BRG16 (BAUDCONx<3>) también controlan el *baud rate*. En modo sincrónico, BRGH es ignorado. Una vez determinado el *Baud Rate* deseado y sabiendo el valor *FOSC* (Frecuencia del OSCilador del sistema), el valor entero más cercado que se debe introducir en los registros SPBRGHx:SPBRGx se puede calcular con las fórmulas de la Tabla 10.

Configuration Bits			BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-bit/Asynchronous	FOSC/[64 (n + 1)]
0	0	1	8-bit/Asynchronous	
0	1	0	16-bit/Asynchronous	FOSC/[16 (n + 1)]
0	1	1	16-bit/Asynchronous	
1	0	x	8-bit/Synchronous	FOSC/[4 (n + 1)]
1	1	x	16-bit/Synchronous	

x = don't care, n = value of SPBRGHx:SPBRGx register pair.

Tabla 10: fórmulas para calcular el Baud Rate.

### 2.2.2. Master Synchronous Serial Port (MSSP)

El módulo MSSP es una interface serial apropiada para comunicarse con otros dispositivos microcontroladores o periféricos, tales como EEPROMs seriales, registros de desplazamiento, displays y conversores A/D.

El módulo MSSP puede operar en uno de dos modos:

- *Serial Peripheral Interface* (SPI).
- *Inter-Integrated Circuit* (I2C™).
  - Modo *Full Master*.
  - Modo *Slave* (con direccionamiento generalizado)

La interface I2C soporta en hardware los siguientes modos:

- Modo *Master*.

- Modo *Multi-Master*.
- Modo *Slave* con 5 y 7 bits de máscara de dirección (con enmascaramiento de dirección para direccionamiento de 10 y 7 bits).

El protocolo SPI es un protocolo de comunicación desarrollado por Motorola que permite el intercambio de datos de manera sencilla entre distintos procesadores y periféricos externos, como pueden ser memorias, Conversores Analógico Digital (ADC), entre otros. Se transmite sobre 4 líneas de manera sincrónica y *full-duplex*, con una topología maestro-esclavo. Posee una línea de reloj (SPSCK), una salida de datos serie (MOSI: *Master Output-Slave Input*), una entrada de datos serie (MISO: *Master Input Slave Output*) y una de selección (SS: *Slave Select*) para cada esclavo que esté conectado al bus. El maestro inicia la comunicación seleccionando al esclavo mediante la línea SS, y por cada flanco de reloj, envía un bit de manera serie a través de la línea MOSI. El bus I2C fue diseñado por Philips a principios de la década del 80 para permitir una fácil comunicación entre componentes que residen en la misma placa. El bus solo requiere dos líneas de conexión, una de datos y otra de *clock*. No requiere un *Baud Rate* stricto ya que el master genera el sincronismo en la línea de *clock*. Los dispositivos conectados al bus son direccionados por software habiendo una única dirección para cada uno. Provee la posibilidad de la existencia de un bus *multi-master* con arbitrariedad y detección de colisión. Las velocidades de comunicación van desde 100 kbit por segundo hasta 3,4 Mbits en modo *High Speed*. I2C no solo se puede usar para interconectar componentes en la misma placa, sino también para hacerlo con componentes de otras placas, conectados a través de cables. Seguido, se muestran unos diagramas de bloques de los módulos SPI e I2C del PIC18F27J53.

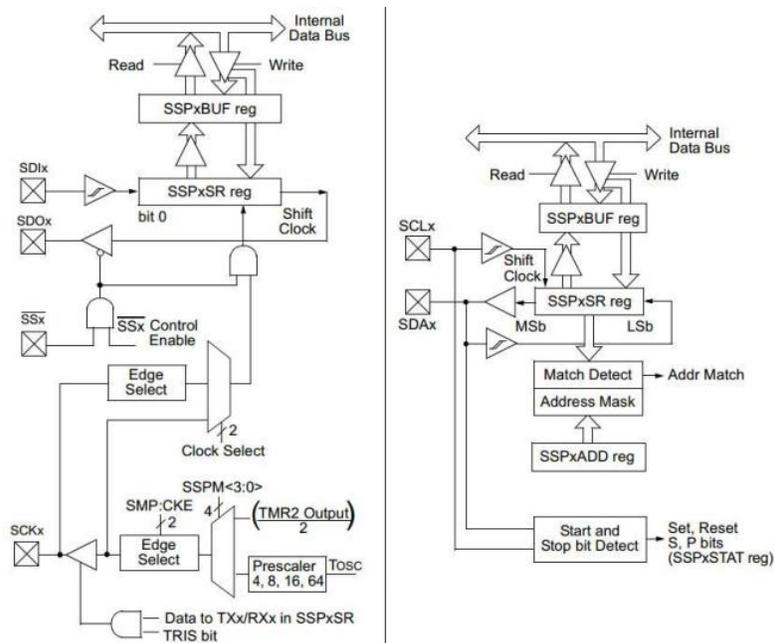


Ilustración 27: diagrama de bloques del módulo SPI (izq.) e I2C (der.) del PIC18F27J53.

Cada módulo MSSP tiene tres registros de control asociados: un registro de estado (*SSPxSTAT*), y dos registros de control (*SSPxCON1* y *SSPxCON2*). Su uso y bits de configuración individuales difieren significativamente dependiendo de si se opera en modo SPI o I2C. Nos enfocaremos en el modo SPI ya que fue el utilizado en este proyecto.

El modo SPI permite transmitir y recibir sincrónicamente datos de 8 bits de manera simultánea. Existen cuatro modos SPI. Cada módulo MSSP, operando en modo SPI, posee cuatro registros:

- MSSPx Control Register 1 (*SSPxCON1*).
- MSSPx Status Register (*SSPxSTAT*).
- Serial Receive/Transmit Buffer Register (*SSPxBUF*).
- MSSPx Shift Register (*SSPxSR*) – No accesible directamente.

*SSPxCON1* y *SSPxSTAT* son los registros de control y estado. El registro *SSPxCON1* se puede tanto leer como escribir. Los seis bits menos significativos de *SSPxSTAT* son de solo lectura. Los dos bits más significativos del *SSPxSTAT* se pueden leer y escribir. *SSPxSR* es el registro de desplazamiento usado para el ingreso o egreso de los bits. *SSPxBUF* es el buffer en el cual se escriben o se leen los bytes de salida o entrada, respectivamente. En operación de recepción, juntos *SSPxSR* y *SSPxBUF*, crean un buffer doble de recepción. Cuando *SSPxSR* recibe un byte completo es transferido a *SSPxBUF* y el bit de interrupción *SSPxIF* (bandera de interrupción del MSSP para la recepción de dato) es setead. Durante la transmisión el *SSPxBUF* no funciona como doble buffer. Una acción de escritura en *SSPxBUF* también se realizará sobre *SSPxSR*. A continuación se detallan los registros de control:

R/W-1	R/W-1	R-1	R-1	R-1	R-1	R-1	R-1
<b>SMP</b>	<b>CKE<sup>(1)</sup></b>	<b>D/<math>\bar{A}</math></b>	<b>P</b>	<b>S</b>	<b>R/<math>\bar{W}</math></b>	<b>UA</b>	<b>BF</b>
bit 7						bit 0	

Tabla 11: registro *SSPxSTAT* (MSSPx STATUS REGISTER) – MODO SPI.

- bit 7 **SMP**: Sample bit.  
SPI Master mode:  
 1 = Input data sampled at end of data output time.  
 0 = Input data sampled at middle of data output time.  
SPI Slave mode: SMP must be cleared when SPI is used in Slave mode.
- bit 6 **CKE**: SPI Clock Select bit<sup>(1)</sup>.  
 1 = Transmit occurs on transition from active to Idle clock state.  
 0 = Transmit occurs on transition from Idle to active clock state.
- bit 5 **D/ $\bar{A}$** : Data/ $\overline{\text{Address}}$  bit

- Used in I2C™ mode only.
- bit 4 **P**: Stop bit  
Used in I2C mode only; this bit is cleared when the MSSP module is disabled, SSPEN is cleared.
- bit 3 **S**: Start bit  
Used in I2C mode only.
- bit 2 **R/ $\overline{W}$** : Read/ $\overline{\text{Write}}$  Information bit  
Used in I2C mode only.
- bit 1 **UA**: Update Address bit  
Used in I2C mode only.
- bit 0 **BF**: Buffer Full Status bit  
1 = Receive complete, SSPxBUF is full.  
0 = Receive not complete, SSPxBUF is empty.

**Note 1:** La polaridad del estado del clock es seteada por el bit CKP (SSPxCON1<4>).

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>WCOL</b>	<b>SSPOV</b> <sup>(1)</sup>	<b>SSPEN</b> <sup>(2)</sup>	<b>CKP</b>	<b>SSPM3</b> <sup>(3)</sup>	<b>SSPM2</b> <sup>(3)</sup>	<b>SSPM1</b> <sup>(3)</sup>	<b>SSPM0</b> <sup>(3)</sup>
bit 7							bit 0

Tabla 12: registro SSPxCON1 (MSSPx CONTROL REGISTER 1).

- bit 7 **WCOL**: Write Collision Detect bit.  
1 = The SSPxBUF register is written while it is still transmitting the previous word (must be cleared in software).  
0 = No collision.
- bit 6 **SSPOV**: Receive Overflow Indicator bit(1).  
SPI Slave mode:  
1 = A new byte is received while the SSPxBUF register is still holding the previous data. In case of overflow, the data in SSPxSR is lost. Overflow can only occur in Slave mode. The user must read the SSPxBUF, even if only transmitting data, to avoid setting overflow (must be cleared in software).  
0 = No overflow.
- bit 5 **SSPEN**: Master Synchronous Serial Port Enable bit<sup>(2)</sup>.  
1 = Enables serial port and configures SCKx, SDOx, SDIx and  $\overline{\text{SSx}}$  as serial port pins.  
0 = Disables serial port and configures these pins as I/O port pins.
- bit 4 **CKP**: Clock Polarity Select bit.

1 = Idle state for clock is a high level.

0 = Idle state for clock is a low level.

- bit 3-0 **SSPM<3:0>**: Master Synchronous Serial Port Mode Select bits<sup>(3)</sup>.
  - 0101 = SPI Slave mode, clock = SCKx pin;  $\overline{SSx}$  pin control disabled,  $\overline{SSx}$  can be used as I/O pin.
  - 0100 = SPI Slave mode, clock = SCKx pin;  $\overline{SSx}$  pin control enabled.
  - 0011 = SPI Master mode, clock = TMR2 output/2.
  - 0010 = SPI Master mode, clock = FOSC/64.
  - 0001 = SPI Master mode, clock = FOSC/16.
  - 1010 = SPI Master mode, clock = FOSC/8.
  - 0000 = SPI Master mode, clock = FOSC/4.

- Note**
- 1: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPxBUF register.
  - 2: When enabled, this pin must be properly configured as input or output.
  - 3: Bit combinations not specifically listed here are either reserved or implemented in I2C™ mode only.

### 2.2.3. Puertos I/O y *Peripheral Pin Select* (PPS)

El PIC18F27J53 puede usar hasta tres puertos de ocho bits, dos semicompletos y uno completo (PORT A de 7 bits, PORT B completo y PORTC de 7 bits). La gran mayoría de los pines de los puertos I/O son multiplexados con una función alternativa de uno o varios de los periféricos incluidos en el dispositivo. En general, cuando un periférico asociado a un pin está habilitado, dicho pin no debe ser usado para propósitos generales sino específicamente para la función asociada al periférico. Cada puerto posee tres registros:

- TRIS: registro de dirección de datos (especifica que pin es entrada o salida).
- PORT: lee el nivel presente en el correspondiente pin del dispositivo.
- LAT: latch de datos (*flip-flop* de salida).

El registro de datos LAT es útil para operaciones de tipo leer-modificar-escribir sobre el nivel lógico que el pin I/o está manejando.

Los niveles de potencia de salida que los pines pueden manejar cambian según los grupos de pines de manera de cumplir con los requisitos de variadas aplicaciones. PORT B y C están diseñados para manejar altas cargas, como podrían ser LEDs. El resto de los puertos están diseñados para soportar bajas cargas, como lo serían señales lógicas. La tolerancia de voltaje de los pines utilizados como entrada depende de la función asociada al pin. Los pins que son

usados como entradas digitales exclusivamente están capacitadas para manejar niveles de voltaje continuo de hasta 5,5V; un nivel típico de circuitos lógicos. En contraste, los pines que también tienen función de entrada analógica de cualquier tipo solo pueden tolerar voltajes de hasta VDD. En la Ilustración 26 pueden verse los pines que toleran hasta 5.5V sombreados. Para el caso de los pines operando como salidas digitales, es posible seleccionar la funcionalidad de *Open Drain Colector* (salida con colector abierto) y haciendo uso de una resistencia de *pull-up*, poder trabajar con niveles superiores a VDD. Para ello existe el registro ODCON1 (*peripheral Open-Drain CONTROL register 1*).

A continuación se muestra un diagrama de bloques general de los pines de los puertos I/O y una serie de tablas con los registros asociados a cada puerto:

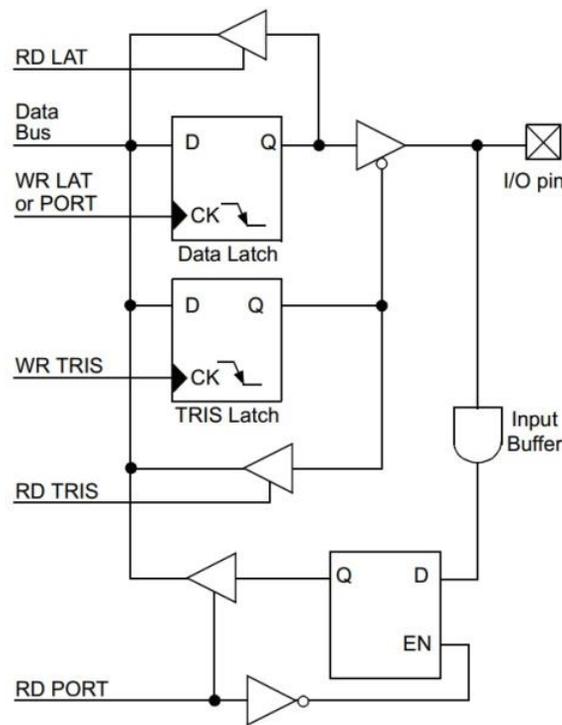


Ilustración 28: diagrama de bloques genérico de la operación de un puerto I/O del PIC18F27J53.

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTA	RA7	RA6	RA5	-	RA3	RA2	RA1	RA0
LATA	LATA7	LATA6	LATA5	-	LATA3	LATA2	LATA1	LATA0
TRISA	TRISA7	TRISA6	TRISA5	-	TRISA3	TRISA2	TRISA1	TRISA0
ANCON0	-	-	-	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
CMxCON	CON	COE	CPOL	EVPOL1	EVPOLO	CREF	CCH1	CCH0
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0
WDTCON	REGSLP	LVDSTAT	ULPLVL	VBGOE	DS	ULPEN	ULPSINK	SWDTEN
HLVDCON	VDIRMAG	BGVST	IRVST	HLVDEN	HLVDL3	HLVDL2	HLVDL1	HLVDL0

Tabla 13: resumen de los registros asociados al PORTA.

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
INTCON2	RBPŪ	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF
CM3CON	CON	COE	CPOL	EVPOL1	EVPOLO	CREF	CCH1	CCH0
PADCFG1	-	-	-	-	-	RTSECSEL1	RTSECSEL0	PMPTTL
RTCCFG	RTCEN	-	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR1	RTCPTRO

Tabla 14: resumen de los registros asociados al PORTB.

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTC	RC7	RC6	RC5	RC4	-	RC2	RC1	RC0
LATC	LATC7	LATC6	-	-	-	LATC2	LATC1	LATC0
TRISC	TRISC7	TRISC6	-	-	-	TRISC2	TRISC1	TRISC0
ANCON1	VBGEN	-	-	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
CM2CON	CON	COE	CPOL	EVPOL1	EVPOLO	CREF	CCH1	CCH0
UCON	-	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	-
UCFG	UTEYE	UOEMON	-	UPUEN	UTRDIS	FSEN	PPB1	PPB0
RTCCFG	RTCEN	-	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR1	RTCPTRO

Tabla 15: resumen de los registros asociados al PORTC.

La funcionalidad de **Peripheral Pin Select** (PPS) permite asociar físicamente tanto entradas como salidas de distintos periféricos internos del PIC18F27J53 a cualquiera de los pines identificados como *RPx* (*Remappable Peripheral x*). Los pines disponibles pueden verse en la Ilustración 26. Esto brinda al programador la posibilidad de personalizar la distribución de los puertos del microcontrolador según las necesidades presentes. El usuario debe mapear cada entrada y/o salida digital de cualquiera de los muchos periféricos que se deseen utilizar de forma independiente unas de otras. La selección de periféricos se realiza por software y, generalmente, no es necesario reprogramar el microcontrolador. El hardware incluye protecciones para prevenir cambios accidentales o espurios al mapeo una vez que ha sido establecido. A demás de esto, existe una secuencia de desbloqueo y bloqueo por software para poder efectivamente realizar un nuevo mapeado o remapeado, sumando una instancia más de protección. Cuando el terminal de un periférico remapeable está activo en un determinado I/O pin, el terminal mapeado tiene prioridad por sobre el resto de las funciones digitales o funciones asociadas a otros periféricos fijos. La prioridad es dada sin importar que tipo de periférico sea mapeado. Por el contrario, PPS nunca tendrá prioridad por sobre una función analógica asociada al pin.

Las funcionalidades PPS son controladas a través de dos sets de Registros de Funciones Especiales (SFRs): uno para mapear entradas de periféricos y el otro mapear salidas. Como las entradas y salidas de los periféricos son tratadas por separado, las mismas pueden ser

mapeadas en cualquiera de los pines RPx sin restricción, sin importar si pertenecen al mismo periférico. El modo de asociación del terminal de un periférico a un pin RPx depende de si el mismo es una entrada o una salida. Para mapear la entrada de un determinado periférico existe un registro dedicado a dicha entrada sobre el cual se selecciona el terminal RPx que se quiere mapear. Los registros RPINRx contienen un campo de 5 bits donde se ingresa el valor asociado al número de pin RPx al cual se quiere asociar la respectiva entrada. A continuación se muestra una lista de los registros RPINRx y las entradas de los periféricos asociados:

Input Name	Function Name	Register	Configuration Bits
External Interrupt 1	INT1	RPINR1	INTR1R<4:0>
External Interrupt 2	INT2	RPINR2	INTR2R<4:0>
External Interrupt 3	INT3	RPINR3	INTR3R<4:0>
Timer0 External Clock Input	T0CKI	RPINR4	T0CKR<4:0>
Timer3 External Clock Input	T3CKI	RPINR6	T3CKR<4:0>
Timer5 External Clock Input	T5CKI	RPINR15	T5CKR<4:0>
Input Capture 1	CCP1	RPINR7	IC1R<4:0>
Input Capture 2	CCP2	RPINR8	IC2R<4:0>
Input Capture 3	CCP3	RPINR9	IC3R<4:0>
Timer1 Gate Input	T1G	RPINR12	T1GR<4:0>
Timer3 Gate Input	T3G	RPINR13	T3GR<4:0>
Timer5 Gate Input	T5G	RPINR14	T5GR<4:0>
EUSART2 Asynchronous Receive/Synchronous Receive	RX2/DT2	RPINR16	RX2DT2R<4:0>
EUSART2 Asynchronous Clock Input	CK2	RPINR17	CK2R<4:0>
SPI2 Data Input	SDI2	RPINR21	SDI2R<4:0>
SPI2 Clock Input	SCK2IN	RPINR22	SCK2R<4:0>
SPI2 Slave Select Input	SS2IN	RPINR23	SS2R<4:0>
PWM Fault Input	FLT0	RPINR24	OCFAR<4:0>

Tabla 16: registros RPINRx y las entradas de los periféricos asociados.

En contraste con las entradas, las salidas son mapeadas según el pin RPx que se quiera utilizar. En este caso existen los registros RPORx, cada uno dedicado a cada pin RPx, sobre los cuales se escribe el respectivo valor o código de la salida del periférico al cual se quiere asociar el pin. Debido a esta metodología de mapeo se incluye el valor “00000” a la lista de códigos de mapeo de salidas de periféricos, permitiendo esto dejar el pin RPx desconectado de cualquier salida. A continuación se muestra una lista de los valores de las salidas de los periféricos remapeables:

Function	Output Function Number <sup>(1)</sup>	Output Name
NULL	0	NULL <sup>(2)</sup>
C1OUT	1	Comparator 1 Output
C2OUT	2	Comparator 2 Output
C3OUT	3	Comparator 3 Output
TX2/CK2	6	EUSART2 Asynchronous Transmit/Asynchronous Clock Output

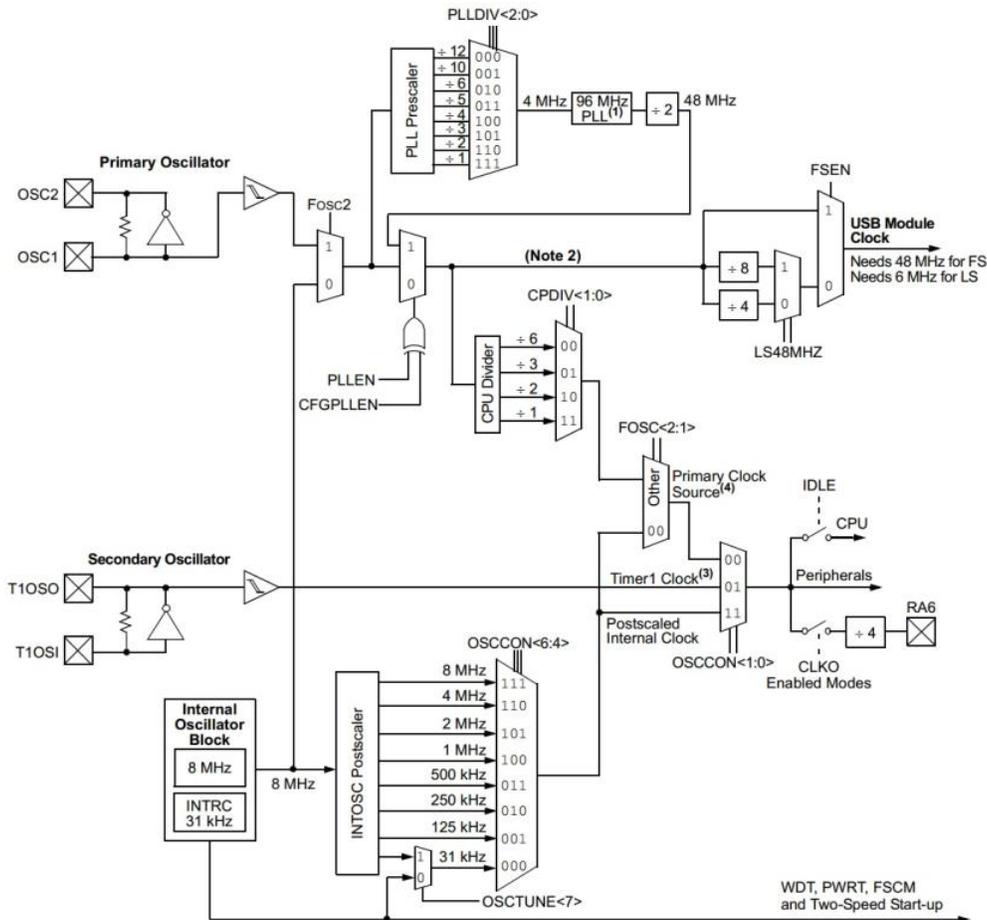
DT2	7	EUSART2 Synchronous Transmit
SDO2	10	SPI2 Data Output
SCK2	11	SPI2 Clock Output
SSDMA	12	SPI DMA Slave Select
ULPOUT	13	Ultra Low-Power Wake-up Event
CCP1/P1A	14	ECCP1 Compare or PWM Output Channel A
P1B	15	ECCP1 Enhanced PWM Output, Channel B
P1C	16	ECCP1 Enhanced PWM Output, Channel C
P1D	17	ECCP1 Enhanced PWM Output, Channel D
CCP2/P2A	18	ECCP2 Compare or PWM Output
P2B	19	ECCP2 Enhanced PWM Output, Channel B
P2C	20	ECCP2 Enhanced PWM Output, Channel C
P2D	21	ECCP2 Enhanced PWM Output, Channel D
CCP3/P3A	22	ECCP3 Compare or PWM Output
P3B	23	ECCP3 Enhanced PWM Output, Channel B
P3C	24	ECCP3 Enhanced PWM Output, Channel C
P3D	25	ECCP3 Enhanced PWM Output, Channel D
<p><b>Note 1:</b> el valor asignado al campo RPORx&lt;4:0&gt; corresponde al número de la salida del periférico.  <b>2:</b> la función NULL es asignada por defecto a todos los registros RPORx.</p>		

Tabla 17: lista de los valores de las salidas de los periféricos remapeables.

#### 2.2.4. Oscilador del Sistema

Los dispositivos microcontroladores de la familia PIC18F47J53 incorporan un oscilador y reloj de sistema distintos a los dispositivos de propósito general PIC18F. Además, el módulo USB, cuyo único requerimiento es un reloj estable, hace necesario el uso de fuentes de reloj separadas que sean compatibles con ambas especificaciones de *low* y *high speed* USB.

En la Ilustración 29 se puede observar la estructura del sistema de oscilador del PIC18F27J53. La familia de microcontroladores PIC18F47J53 posee *prescalers* y *postcalers* para lograr un gran rango de frecuencias de oscilación. El funcionamiento de oscilador se controla a través de tres registros de configuración y dos de control. Los registros de configuración CONFIG1L, CONFIG1H y CONFIG2L seleccionan el modo del oscilador, el *prescaler* del PLL (*Phase Locked Loop*) y las opciones del divisor de frecuencia de la CPU. Los registros de configuración son seteados en la programación y permanecen en el mismo estado hasta una nueva reprogramación. El registro OSCCON selecciona el modo de *Active Clock* (fuera del modo *Sleep* o bajo consumo); es usado principalmente para controlar los cambios del reloj en los modos *power-managed*. El registro OSCTUNE es usado para ajustar fuente de frecuencia de INTOSC y seleccionar la fuente de reloj de *low-frequency* que alimenta varias características especiales.



- Note 1:** The PLL requires a 4 MHz input and it produces a 96 MHz output. The PLL will not be available until the PLEN bit in the OSC TUNE register is set. Once the PLEN bit is set, the PLL requires up to  $t_{rc}$  to lock. During this time, the device continues to be clocked at the PLL bypassed frequency.
- Note 2:** In order to use the USB module in Full-Speed mode, this node must be run at 48 MHz. For Low-Speed mode, this node may be run at either 48 MHz or 24 MHz, but the CPDIV bits must be set such that the USB module is clocked at 6 MHz.
- Note 3:** Selecting the Timer1 clock or postscaled internal clock will turn off the primary oscillator (unless required by the reference clock of **Section 3.6 "Reference Clock Output"**) and PLL.
- Note 4:** The USB module cannot be used to communicate unless the primary clock source is selected.

Ilustración 29: visión general de la estructura del oscilador del PIC18F27J53.

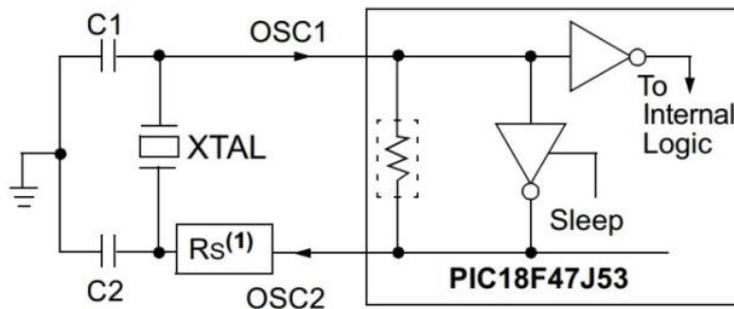
El registro OSCTUNE también se utiliza para habilitar o deshabilitar el PLL. Existen ocho modos distintos de oscilación:

- **ECPLL (External Clock Input).** El PLL puede ser habilitado o deshabilitado por software. Presencia de CLKO (*Clock Output*) en el puerto RA6.
- **EC (External Clock Input).** El PLL está siempre deshabilitado. Presencia de CLKO (*Clock Output*) en el puerto RA6.
- **HSPLL (High-Speed Crystal/Resonator).** El PLL puede ser habilitado o deshabilitado por software. Cristal/resonador conectado entre RA6 y RA7.
- **HS (High-Speed Crystal/Resonator).** Cristal/resonador conectado entre RA6 y RA7. El PLL está siempre deshabilitado.

- **INTOSCPLLO** (*Internal Oscillator*). El PLL puede ser habilitado o deshabilitado por software. Presencia de CLKO (*Clock Output*) en el puerto RA6. RA7 funciona como puerto I/O. El bloque del oscilador interno es usado para alimentar tanto la fuente de reloj primaria como el reloj interno *postscaled*.
- **INTOSCPLL** (*Internal Oscillator*). El PLL puede ser habilitado o deshabilitado por software. RA6 y RA7 funcionan como puertos I/O. El bloque del oscilador interno es usado para alimentar tanto la fuente de reloj primaria como el reloj interno *postscaled*.
- **INTOSCO** (*Internal Oscillator*). El PLL está siempre deshabilitado. Presencia de CLKO (*Clock Output*) en el puerto RA6. RA7 funciona como puerto I/O. La salida del *postscaler* INTOSC sirve de fuente tanto para el reloj interno *postscaled* como el reloj primario.
- **INTOSC** (*Internal Oscillator*). El PLL está siempre deshabilitado. RA6 y RA7 funcionan como puertos I/O. La salida del *postscaler* INTOSC sirve de fuente tanto para el reloj interno *postscaled* como el reloj primario.

Los modos de oscilación se seleccionan al programar el dispositivo a través de los bits de configuración FOSC<2:0>. Para los modos que producen una salida de reloj (CLKO) en el puerto RA6, la frecuencia de salida será de un cuarto de la frecuencia de reloj de los periféricos. En el modo *Sleep*, CLKO cesará pero continuará en el modo Idle (bajo consumo).

Para este proyecto se utilizó el oscilador principal en el modo HSPLL, un cristal de 4MHz y dos capacitores de 27pF, según el esquema de la Ilustración 30. Configurando apropiadamente los bits del campo PLLDIV de manera de que el *PLL-prescaler* no altere la frecuencia de entrada del oscilador principal se logró obtener una frecuencia de trabajo de la CPU y periféricos de 48MHz. A continuación se muestra un esquema del circuito del oscilador externo:



**Note 1:** Rs may be required to avoid overdriving crystals with low drive level specifications.

Ilustración 30: esquema general del circuito del oscilador principal con cristal.

## 2.3. Descripción de los códigos de programa

El software fue desarrollado mediante el editor de código *MPLAB IDE v8.92* y el compilador *MPLAB C Compiler for PIC18 MCUs v3.45* de la empresa *Microchip* [12]. Para este proyecto fue necesario realizar dos códigos, uno para el dispositivo M1 y otro para el dispositivo M2. El primer código se encarga principalmente de monitorear dos eventos importantes. Por un lado, la señal de disparo que proviene de la Unidad de Monitoreo (simulada por un pulsador) para generar el paquete de datos, utilizando la información que proporciona el GPS, relacionado al evento que generó el disparo. Por otro, la recepción via XBee de la orden, enviada por el dispositivo M2, para transmitir cada paquete almacenado en la memoria SD. El segundo código está basado en el “*MainDemo.c*” de la librería “*TCP/IP Stack Software*”, el cual se encarga de gestionar una conexión Ethernet a través del ENC28J60 y realizar las funciones necesarias para el dispositivo M2. A continuación se describirán ambos código en detalle.

### 2.3.1. Código del dispositivo M1

El programa consiste de tres procesos principales y de varias funciones secundarias asociadas a los procesos principales. Primero se ejecuta un proceso de inicialización y configuración del hardware y los periféricos. En dicho proceso se inicializan y configuran los puertos y periféricos del microcontrolador, se resetea y configura al GPS, y finalmente se inicializa la memoria SD y se crea el archivo donde se van a escribir los paquetes. Luego, se activan los procesos de atención a interrupciones en segundo plano y por último el programa entra en un bucle “while” infinito. El contenido del “while” consiste en una rutina de encuestas para decidir si ejecutar diferentes subrutinas, las cuales son:

- Verificar si se presionó el pulsador, que simula la señal de disparo del Dispositivo de Monitoreo, y así solicitar los datos al GPS.
- Verificar si se recibieron los datos del GPS para poder generar el paquete de datos correspondiente.
- Verificar si hay un paquete listo para almacenarlo en la memoria SD.
- Verificar si se recibió un comando “*SND*” proveniente del M2 para proceder a enviar un paquete almacenado en la memoria SD.

En la Ilustración 31 se muestra un diagrama de flujo del programa principal del dispositivo M1 y en la Ilustración 32 se puede observar otro diagrama de flujo pero en este caso de la rutina de atención a interrupción de alta prioridad, cuya fuente es INT0.

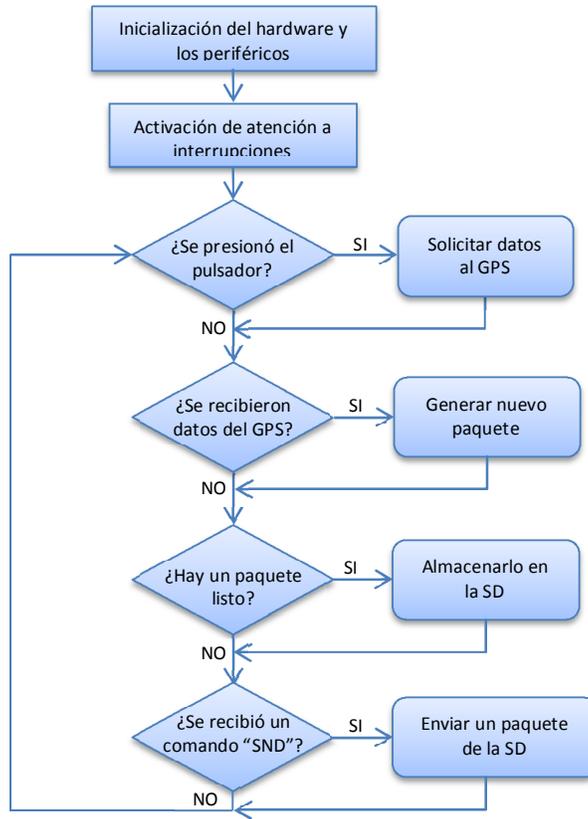


Ilustración 31: diagrama de flujo del programa principal del dispositivo M1.

Las rutinas de atención a interrupción se dividen en rutinas de alta y baja prioridad. En la rutina de alta prioridad se monitorea la aparición de un flanco descendente en el terminal designado como receptor de la interfaz serie dedicada a comunicarse con el módulo XBee. Debido al conflicto antes mencionado en la sección 292.2 con el módulo interno SPI2 el cual no permitía utilizar el puerto EUSART1, fue necesario implementar una USART por software, denominada como *SUSART (SoftwareUSART)*. El receptor de la *SUSART (RxS)* se asoció al puerto RB0 y el transmisor al puerto RC6. Para detectar la recepción de los bits provenientes del terminal transmisor del puerto USART del módulo XBee se recurrió al uso de la fuente de interrupción *INT0 (External Interrupts 0)* la cual genera una interrupción al detectar un cambio de nivel en el terminal asignado como fuente de interrupción. En este caso, la *INT0* estaba asignada por hardware al puerto RB0 (pin 21), y se configuró para detectar un flanco descendente ya que el estado de cese de transmisión del protocolo USART es un nivel alto y el *START-bit* es un cambio a un nivel bajo. De esta manera, al llegar un *START-bit* al puerto RB0 proveniente del módulo XBee se genera un evento de interrupción *INT0*. Dentro de la respectiva rutina de interrupción se llama a los subprocesos encargados de procesar la recepción del byte entrante. Para poder procesar inmediatamente el dato, una vez recibido correctamente, se recurrió al uso de una bandera de interrupción de un periférico no utilizado.

La misma es activada (puesta en 1 lógico) por software al realizarse una correcta recepción de un dato a través del RxS, generándose un evento de interrupción de baja prioridad y así poder procesar el dato recibido, en otra rutina de atención, inmediatamente después de salir de la rutina de alta prioridad.

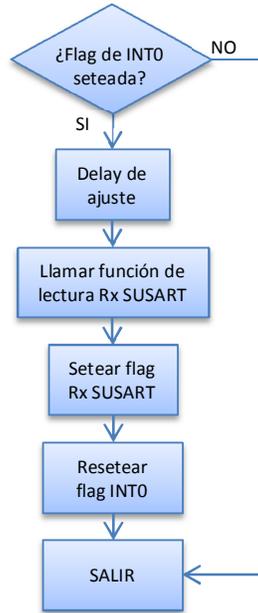


Ilustración 32: diagrama de flujo de la rutina de atención a interrupciones de alta prioridad.

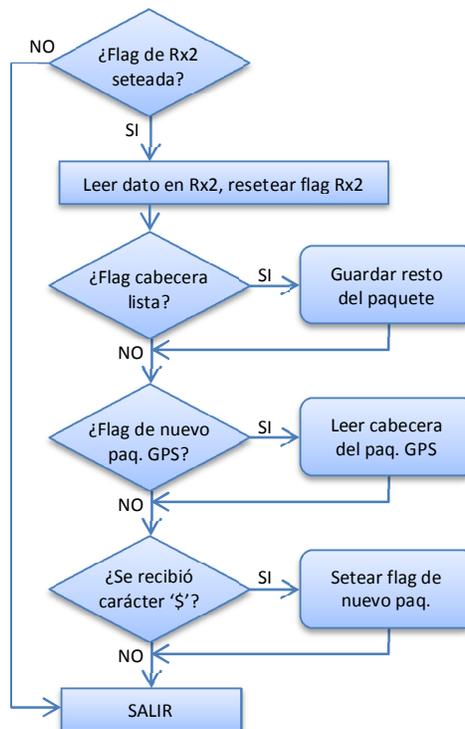


Ilustración 33: diagrama de flujo de la rutina de atención a interrupciones de Rx2.

En las rutinas de atención de interrupciones de baja prioridad se tratan las recepciones de datos de los puertos EUSART2 (Rx2) y SUSART (RxS), cuyos diagramas de flujo pueden verse en la Ilustración 33 y Ilustración 34. Mediante una estructura condicional adecuada se decide si atender la recepción del dato recibido, procesarlo o simplemente ignorarlo. Para el caso del RxS, se espera la recepción de los caracteres "S", "N" y "D", en orden y uno después del otro. Al recibirse dicha secuencia de caracteres se setea una bandera para activar el envío de un paquete desde la memoria SD hacia el módulo XBee. Este proceso se logra poniendo en 1 lógico una bandera por cada carácter recibido correctamente y en el orden debido. Si se interrumpiera el proceso debido a una recepción no esperada se resetean las banderas y el proceso retorna al principio.

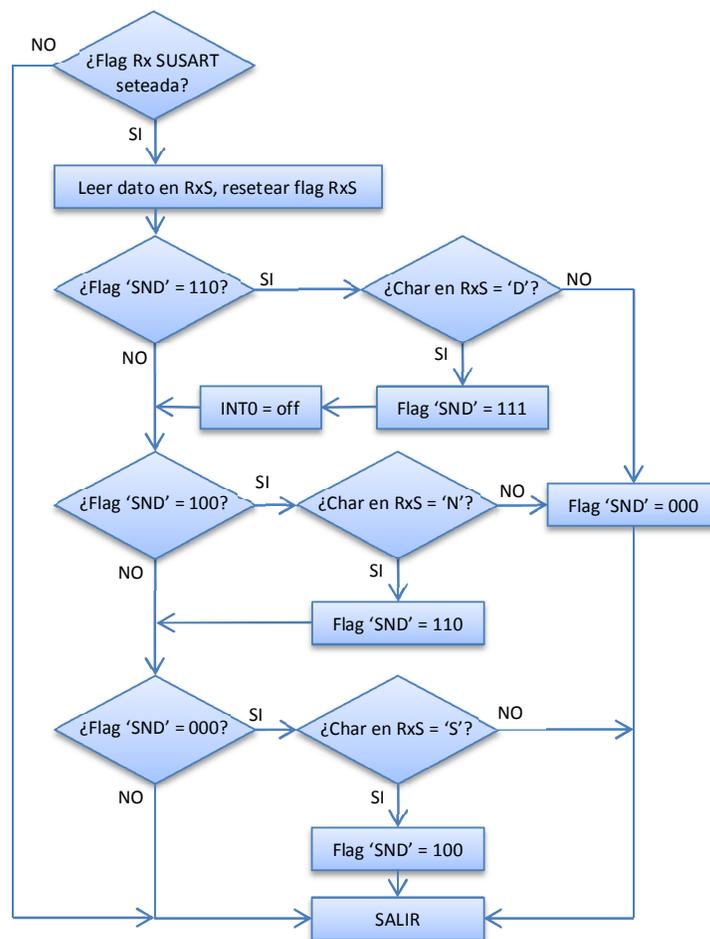


Ilustración 34: diagrama de flujo de la rutina de atención a interrupciones RxS.

Por otro lado, la rutina de atención del puerto Rx2 se encarga en primera instancia de reconocer la recepción del carácter "\$". Dicho carácter es el primer carácter de la cabecera que incluyen los paquetes enviados por el GPS. Si se verifica la recepción de dicho carácter, se activa una bandera para recibir y analizar la cabecera del paquete, a medida que se van

recibiendo los sucesivos caracteres. Si, una vez recibida, la cabecera es la correcta, se activa una bandera para recibir el resto del paquete y guardarlo en un buffer para luego poder extraer los datos del mismo. Cabe destacar que en este proceso, cada vez que se activa una bandera, se resetea la bandera antes activada.

### 2.3.2. Código del dispositivo M2

El programa del dispositivo M2 está basado en la librería “*TCP/IP Stack Software*”. Específicamente, se utilizó uno de los códigos del ejemplo que incluye la librería. A partir del código original del ejemplo se desarrolló el código final, modificándolo para cumplir con las necesidades del hardware utilizado y agregándole todos los algoritmos necesarios para poder operar correctamente el dispositivo M2.

La estructura del código principal del dispositivo M2 es muy parecida a la del M1. Primero se realiza un proceso de inicialización del hardware y el software que utiliza la librería. Luego se activan las rutinas de atención a interrupciones y luego el programa entra en un bucle “while” infinito. El proceso de inicialización consiste en el llamado a una serie de funciones diseñadas para tal propósito, que se describen a continuación:

- *InitializeBoard()*: se encarga de inicializar el hardware (periféricos internos al micro y externos) y software asociado al mismo.
- *TickInit()*: inicializa componentes relacionados al *TPC/IP stack* que podrían llegar a ser requeridos por las rutinas de configuración por UART.
- *InitAppConfig()*: inicializa el *stack* y variables relacionadas a aplicaciones dentro de “*AppConfig*”.
- *StackInit()*: inicializa las capas del núcleo del *stack* (MAC, ARP, TCP, UDP) y los módulos de aplicación (HTTP, SNMP, etc.).

Dentro del bucle “while” infinito se realiza el llamado a una serie de funciones operativas de control con fines variados. Las mismas se describen a continuación:

- *StackTask()*: realiza tareas regulares del *stack* incluyendo chequeo de paquete entrante, reconocimiento del tipo de paquete y llamado a la entidad apropiada del *stack* para procesarlo.
- *StackApplications()*: invoca a cada tarea de la aplicación del núcleo del *stack*.
- *GenericTCPClient()*: ejemplo de aplicación de un cliente TCP genérico. Esta función en realidad consiste de un módulo agregado al “*main.c*” el cual implementa un ejemplo de cliente HTTP. El mismo fue modificado de acuerdo a las necesidades del proyecto.
- *ProcessIO()*: aquí se realizan las operaciones de control específicas del dispositivos M2. Primero se verifica, mediante un condicional, si está habilitado el envío de un nuevo

comando "SND" a través del módulo XBee para así enviarlo o no. El envío de un comando 'SND' lo deshabilita hasta que un nuevo evento lo vuelva a habilitar. Luego se verifican algunas banderas operativas para realizar acciones relacionadas a la recepción de paquetes enviados por el dispositivo M1 y el envío de los mismos a través de la conexión TCP al servidor.

Con respecto a las rutinas de atención a interrupción, también se dividen en rutinas de alta y baja prioridad, como en el código del M1. En la rutina de atención de alta prioridad se procesan los datos recibidos a través de la EUSART2 provenientes del módulo XBee. Dichos datos deberían ser los paquetes enviados por el dispositivo M1, los cuales poseen una cabecera "\$ZB". Al recibirse los datos se analiza la correcta recepción de la cabecera mediante un proceso similar al usado en la rutina de atención de recepción del comando "SND" de la SUSART en el dispositivo M1. Al corroborarse que los datos entrantes corresponden a un paquete enviado por el M1, se procede a guardar en un buffer el resto de los datos entrantes llamando a una función encargada de realizar dicha tarea.

En la rutina de baja prioridad se atiende una interrupción generada por un *timer* una vez por segundo. Dicha interrupción se utiliza para generar una señalización intermitente con un LED y, en el caso de que no se esté procesando ningún paquete entrante, habilitar el envío de un nuevo comando 'SND'. De esta forma, si algún M1 entra en el área de cobertura del M2, en un tiempo menor a un segundo recibirá un comando 'SND', el cual responderá con un paquete. Cabe destacar que, cuando M2 recibe un nuevo paquete del M1, automáticamente es procesado y enviado al servidor. Al terminar esta acción, inmediatamente es enviado otro comando 'SND' y este proceso se repite hasta que el M1 envíe todos sus paquetes y cese de responder, o salga del área de cobertura.

### **3. DISEÑO, DESARROLLO Y EXPERIMENTACIÓN**

En el presente capítulo se describe el desarrollo físico del proyecto y las experimentaciones realizadas. En un principio el proyecto se originó a partir de la iniciativa del Laboratorio de Comunicaciones de la Facultad de Ingeniería de la UNMDP de idear aplicaciones para los por entonces novedosos módulos XBee y aprovechar sus capacidades. Una de las ideas era generar una red de monitoreo orientada al control del tráfico que incluyera tecnología GPS. A partir de dicha idea, luego de refinar el concepto y agregarle las funcionalidades necesarias, surgió lo que terminaría siendo el dispositivo M1. La conceptualización del dispositivo M2 es consecuencia del primer dispositivo, ya que entre ambos se podía llevar al campo de aplicación los módulos XBee utilizando su potencial inalámbrico al máximo.

Respecto a la placa utilizada para el dispositivo M2, se utilizó una placa de desarrollo basada en un microcontrolador PIC18F2550 de Microchip. Dado que el hardware del dispositivo M2 no requería de una gran complejidad, dicha placa de desarrollo pudo satisfacer las necesidades del proyecto y realizar las pruebas necesarias para comprobar las funcionalidades del sistema completo. Por lo dicho, se procederá a describirla pero no se hará hincapié en el diseño de la misma, ya que no se construyó como parte del proyecto.

#### **3.1. Diseño del circuito esquemáticos del dispositivo M1**

El diseño del circuito esquemático del dispositivo M1 se realizó con el software *EAGLE 7.6* desarrollado por *CadSoft* [3]. En un principio solo se buscó implementar en una misma placa el GPS EM-411, el módulo XBee PRO S2, el microcontrolador PIC18F27J53 y las respectivas fuentes de alimentación. El circuito esquemático diseñado en la primera etapa de la placa se puede apreciar en la Ilustración 35. En el mismo puede verse el PIC18F27J53 (arriba), el módulo XBee (derecha) y dos fuentes de alimentación con tres reguladores. El nivel de alimentación del PIC y del XBee es de 3,3V pero el GPS posee un nivel de 4,5V a 6,5V por lo que se optó por usar un regulador LM7805 de 5V fijos para el GPS y un regulador LM317 (variable) ajustado a 3,3V para el resto del circuito. Las fuentes se diseñaron de manera de poder utilizar los 12V de la batería del auto como fuente de alimentación de todo el sistema. Se realizó el diseño con la posibilidad de reemplazar el regulador LM317 por un regulador fijo de 3,3V, debido a que no estaba disponible en el mercado al momento de realizar la plaqueta. También se agregaron conectores para todos los pines del PIC y del XBee, se implementaron conectores para los distintos niveles de alimentación, un conector para el GPS y conectores de programación para el PIC.

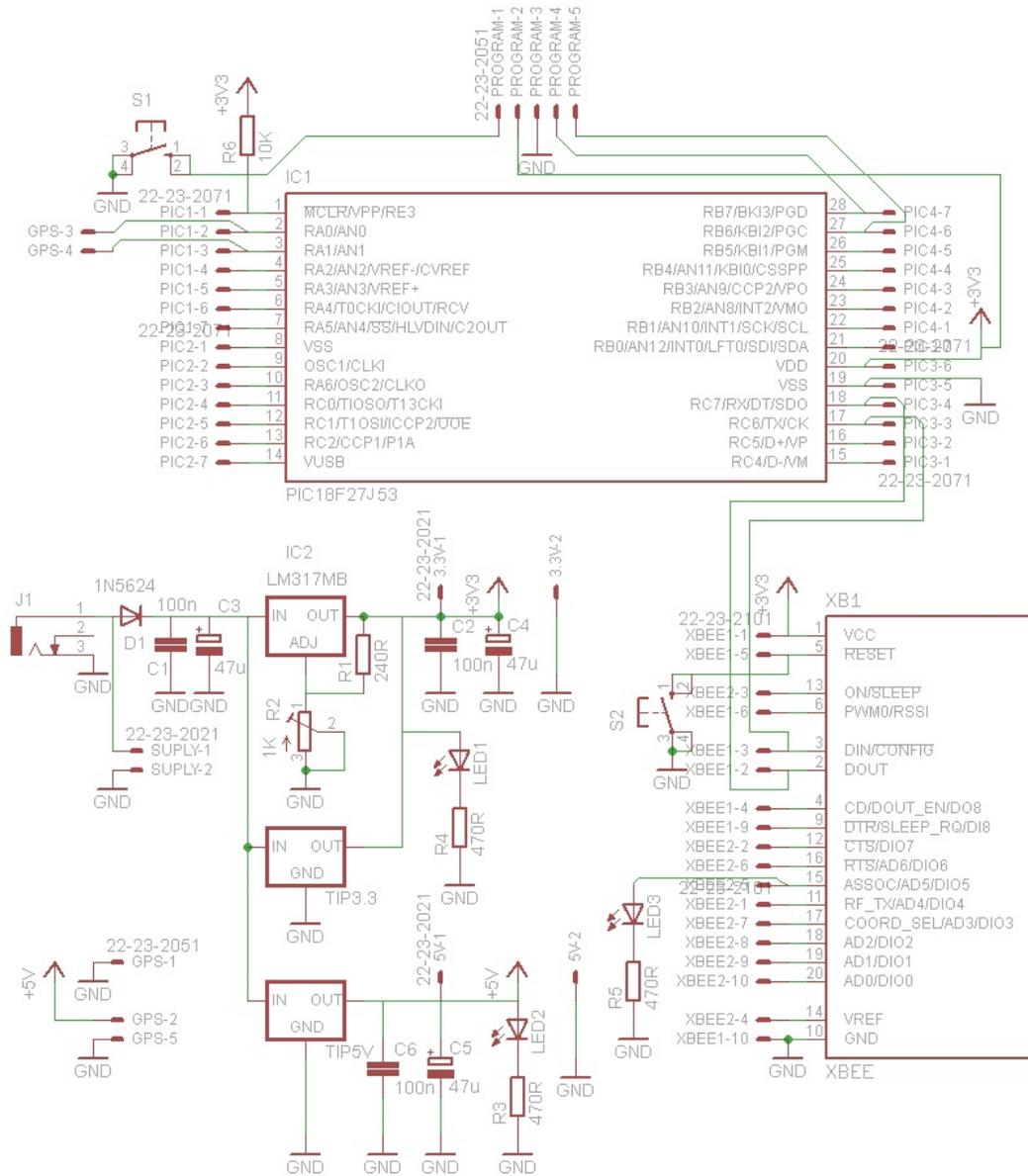


Ilustración 35: circuito esquemático del dispositivo M1 (primera etapa).

Cabe destacar que en este diseño no se incluyó el módulo lector de tarjetas SD ni el circuito del oscilador externo con cristal resonador. Ambos se agregaron a la placa, a base de experimentaciones, a través de los conectores disponibles conectados por cables y tiras de postes, respectivamente.

### 3.2. Diseño de los circuitos impresos

El diseño del circuito impreso del dispositivo M1 fue realizado también con el EAGLE 7.6. El mismo puede verse en la Ilustración 36 junto con una imagen de la placa terminada. Se buscó

realizar una placa compacta pero a la vez cómoda para realizar experimentaciones y pruebas con las distintas funcionalidades del dispositivo M1.

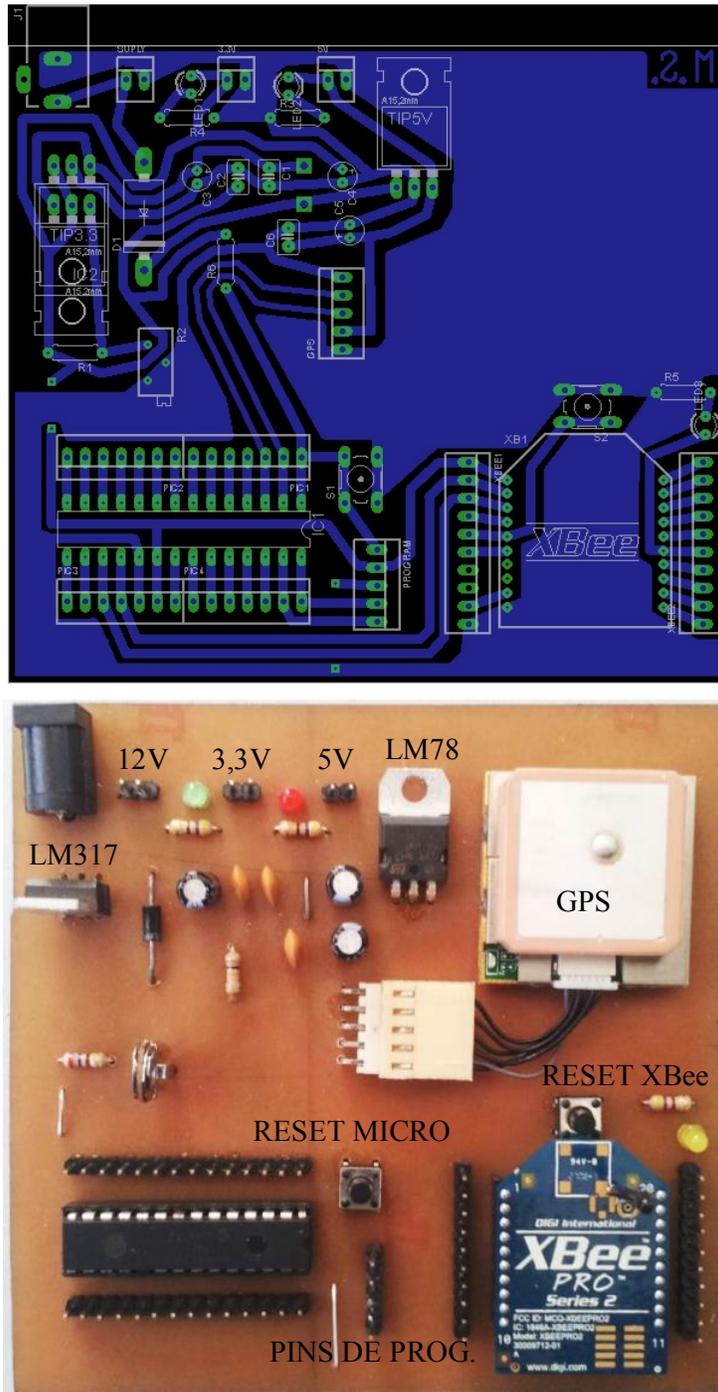


Ilustración 36: circuito impreso y placa terminada del dispositivo M1.

El diseño del circuito impreso del dispositivo M2 fue realizado con el software *DesignSpark PCB 6.0* de la marca comercial *RS Components* [3]. El mismo puede verse en la Ilustración 37 junto con una imagen de la placa.

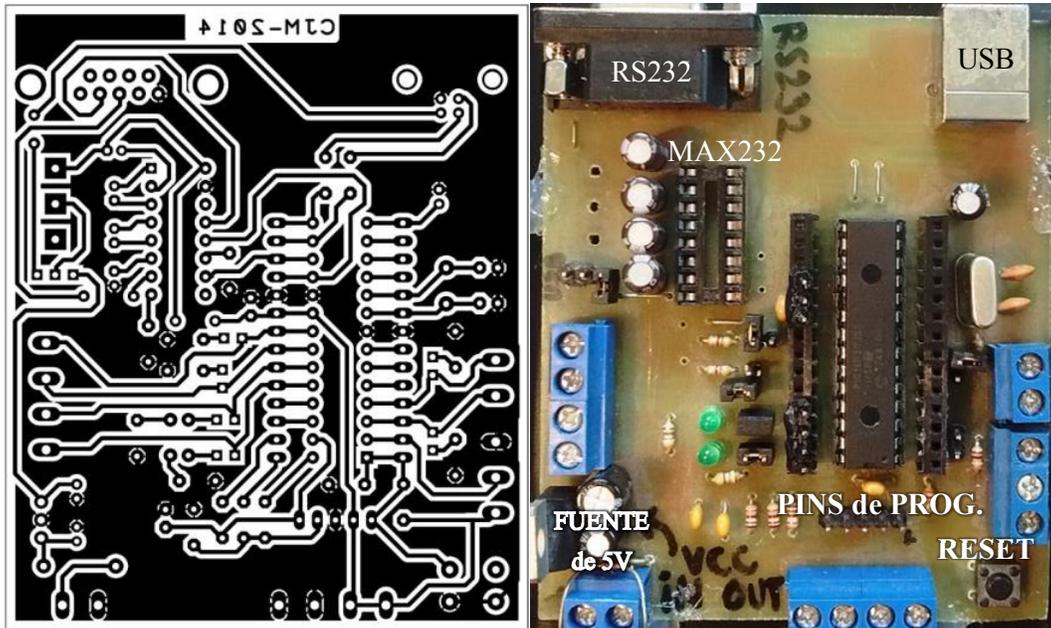


Ilustración 37: circuito impreso de la placa utilizada para el dispositivo M2.

A demás de las placas se construyeron dos circuitos impresos extra para el dispositivo M2. Uno para el módulo XBee y otro para la fuente de alimentación. El primero se realizó para darle un soporte físico al XBee y a su vez simplificar el conexionado. El segundo fue necesario debido a que originalmente la placa de desarrollo utilizada para el dispositivo M2 se alimentaba con un nivel de tensión de 5V y el sistema del dispositivo M2 trabaja con un nivel de 3,3V. A continuación se muestran los dos circuitos impresos:



Ilustración 38: placa adaptadora para el XBee del dispositivo M2.

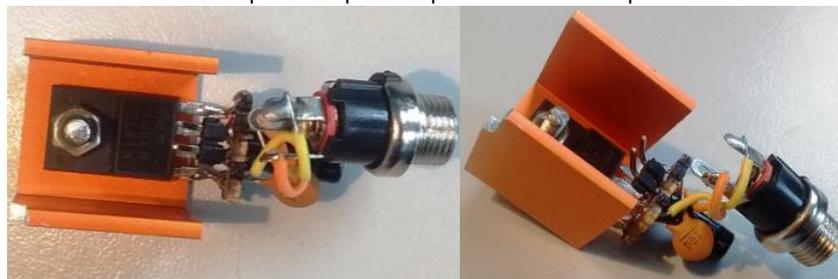


Ilustración 39: fuente de alimentación del dispositivo M2.

### 3.3. Diseño de los dispositivos finales

A continuación se mostrarán imágenes de los dispositivos ya terminados con todos sus componentes conectados.



Ilustración 40: frente y trasera del dispositivo M1.

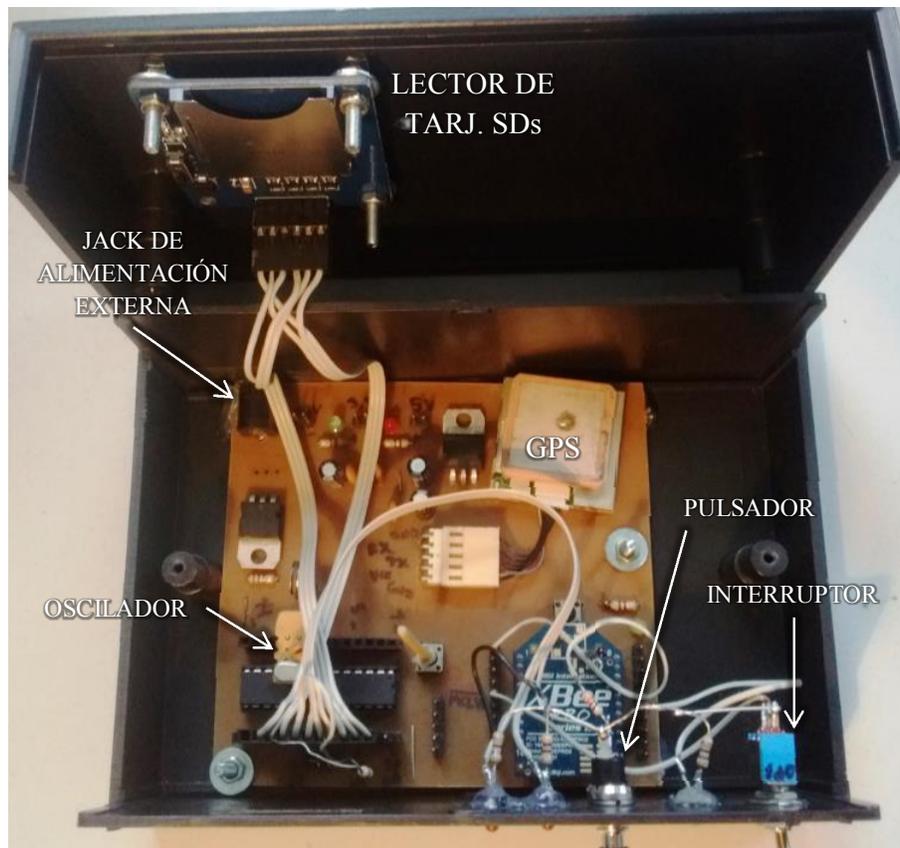


Ilustración 41: interior del dispositivo M1.

En la Ilustración 40 puede verse el frente y la parte trasera del dispositivo M1. En el frente se puede apreciar, de izquierda a derecha, una ranura para una memoria SD, dos LEDs rojos, un pulsador, un LED verde y un interruptor. Los LEDs rojos emiten señales de control que permiten corroborar el correcto funcionamiento del dispositivo. El pulsador se utiliza para simular la señal de disparo que emitiría el bloque de la Unidad de Monitorización y generar los paquetes. El LED verde emite la señal de sincronización del módulo XBee. Cuando el mismo está sincronizado con una red ZigBee el LED emite de forma intermitente. Por último, el interruptor mantiene reseteado al módulo XBee. En la parte trasera del dispositivo se encuentra el orificio para conectar la fuente de alimentación externa (batería de 12V).

En la Ilustración 41 puede observarse el interior del dispositivo M1. En la misma es posible apreciar la placa principal, con la cual se comenzó a trabajar desde el principio del proyecto. También es posible distinguir todo el cableado con interconecta los terminales del PIC con el módulo lector de tarjetas SD y con los LEDs, el pulsador y el interruptor del panel frontal. El circuito del oscilador principal, que también puede verse en detalle en la Ilustración 44, está conectado, mediante pines soldados al mismo, directamente a la tira de postes del PIC.



Ilustración 42: dispositivo M2.

El dispositivo M2 no posee interfaz humana ya que es totalmente automático. Solo posee los conectores Ethernet y de alimentación externa. El mismo puede verse en la Ilustración 42 y, desde su interior, en la Ilustración 43. Dentro del dispositivo M2 puede verse el ENC28J60, el regulador de 3,3V, la placa adaptadora para el módulo XBee y la placa de desarrollo que permitió utilizar y conectar el PIC.



Ilustración 43: interior del dispositivo M2.

### 3.4. Desarrollo y experimentación

El proyecto dio sus primeros pasos con la construcción de la placa para el dispositivo M1 sobre la cual se trabajó con los primeros algoritmos de programa. A base de tutoriales y experimentaciones se logró aprender a utilizar el editor de código MPLAB, comenzar a programar las primeras líneas de código y descubrir las características de un microcontrolador. Lo primero que se realizó fue la edición de las primeras rutinas del software del M1, encargados de controlar la comunicación con el GPS y el XBee a través de los puertos USART. Para facilitar el uso de los puertos EUSARTs del PIC se recurrió a las librerías incluidas en el compilador “MPLAB C Compiler for PIC18” utilizado para el proyecto. Una vez logrado un aceptable funcionamiento y control sobre los puertos USART se prosiguió a la configuración de los módulos XBee. Para ello, se recurrió a la placa de programación de los XBee, la cual permitía ser utilizada como interface con la PC para enviar y recibir datos entre el módulo de la placa de programación y el módulo de la placa del M1. De esta manera fue posible probar simultáneamente las funcionalidades de los puertos EUSARTs, la comunicación con el GPS, la comunicación con el XBee y la correcta recepción y transmisión inalámbrica entre los módulos XBee.

Luego de obtener buenos resultados con la placa del M1 se comenzó con la laboriosa tarea de trabajar sobre el ENC28J60 y las librerías TCP/IP de Microchip. Con ayuda del Dr. Ing. Pablo Agüero y muchas horas de trabajo depurando el código de las librerías TCP/IP se logró una exitosa compilación. Con el código obtenido se pudo establecer un socket entre el PIC y un software de testeo de conexiones TCP a través del ENC28J60. Para las pruebas se utilizó la placa del M1 pero se trabajó desde un código nuevo, separado del que se venía trabajando anteriormente, el cual con el tiempo terminó siendo parte del software del dispositivo M2.

Hasta el momento se venía trabajando con el oscilador interno del PIC. Pero, debido a la necesidad de obtener respuestas más rápidas en las interrupciones y mayor estabilidad, se decidió agregar a la placa un circuito de oscilador externo. De esta manera se comenzó a trabajar con una frecuencia de 48MHz a base de un cristal resonador de 4MHz, el cual puede verse en la Ilustración 44.



Ilustración 44: circuito del oscilador externo del dispositivo M1.

La siguiente etapa fue agregar al primer código, que ya poseía la capacidad de trabajar con los dos puertos EUSARTs, la funcionalidad SPI para comunicarse con el módulo lector de tarjetas SD. Para ello se utilizó como referencia un tutorial basado en las librerías “*Memory Disk Drive File System*” de Microchip. En dicha instancia fue que surgió el ya mencionado problema del puerto SPI2, el cual no podía procesar la recepción de datos. El inconveniente fue ignorado y se siguió trabajando sobre el código utilizando el puerto SPI1, pero más adelante debía ser solucionado de alguna forma. A pesar del inconveniente, se logró poner en marcha el módulo lector de tarjetas SD y realizar pruebas de forma satisfactoria.

El dispositivo M1 estaba casi completo. Solo faltaba resolver el inconveniente asociado al el puerto SPI2. Dado que utilizar el puerto SPI1 imposibilitaba el uso del puerto EUSART1, utilizado para comunicarse con el XBee se creó, de forma íntegra, una librería para utilizar una USART por software, la cual fue denominada como SUSART. De esta manera, se utilizó el puerto SPI1 para comunicarse con el módulo lector de tarjetas SD y los terminales RB0 como receptor y RC6 como transmisor de la SUSART. Para utilizar la SUSART fue necesario realizar una leve modificación al circuito impreso. La modificación consistió en cortar el camino que unía el terminal RC7 del PIC con el Tx del puerto USART del XBee, agregar unos tirapostes y

conectar el Tx del puerto USART del XBee, mediante un cable, al terminal RBO del PIC. Gracias a las medidas mencionadas se logró completar y unificar todos los componentes del dispositivo M1.

Ya con el M1 completo, se prosiguió a desarrollar los bloques formados por el servidor y la base de datos. Para ello se comenzó haciendo pruebas con código PHP, tomando como referencia algunos ejemplos prácticos de la web "[www.tutorialesprogramacionya.com](http://www.tutorialesprogramacionya.com)". Simultáneamente, se comenzó a trabajar con el servidor Apache y el gestor de bases de datos MySQL, contenidos en el sistema WAMP [22]. Luego de conseguir configurar correctamente los módulos del WAMP y realizar algunos ensayos experimentales sobre código PHP se logró generar el archivo "*sendMySQL.php*", encargado de recibir los datos enviados por el PIC y cargarlos en la base de datos gestionada por MySQL. Para ese entonces, era imperativo crear una placa para el dispositivo M2. Fue así que se utilizó la ya mencionada placa de desarrollo, originalmente construida para la cátedra de Instrumentación Virtual de la Facultad de Ingeniería de la UNMDP. Para poder ser utilizada en el proyecto fue necesario armar una fuente externa de 3,3V y un circuito para adaptar el módulo XBee y poder conectarlo al PIC. Por último, ya con la placa correctamente adaptada y funcional, se desarrolló el código adecuado para que el PIC del M2 reciba los datos enviados por el dispositivo M1, los procese y los retransmita al servidor dentro de una trama codificada en PHP, a través del ENC28J60. Con esto se completó el desarrollo de los dos dispositivos. El paso siguiente consistió en verificar el correcto funcionamiento del sistema.

#### **3.4.1. Experimentación**

A partir de un sencillo experimento se intentó obtener algunos parámetros de interés para el desarrollo del proyecto, tales como **alcance del enlace RF, tiempo de establecimiento y volumen de datos de transmisión vs velocidad del vehículo.**

El ensayo consistió en montar el dispositivo móvil dentro de un automóvil y realizar una serie de pasadas a distintas velocidades desde una distancia que superara al área de la cobertura del enlace inalámbrico hasta el punto donde estaba ubicado el módulo receptor y observar el desempeño del sistema. No se tuvo en cuenta el desempeño del sistema al alejarse el vehículo.

Se configuraron ambos módulos en modo AT para que establecieran un vínculo directo, a partir de sus direcciones de identidad. Desde una ventana del Laboratorio de Comunicaciones de la Facultad de Ingeniería de la UNMDP, que daba a la calle por donde pasaría el vehículo con el dispositivo móvil, se dispuso el módulo receptor conectado a la PC y al X-CTU para visualizar los paquetes recibidos. Luego, se realizaron tres pasadas a veinte, treinta y cuarenta

kilómetros por hora, tomando las debidas precauciones de seguridad y siendo la última velocidad la máxima admisible para la experimentación (el límite de velocidad en calles es de 40 km/h). El módulo móvil se configuró para que entregara continuamente los datos que suministraba el GPS (una trama de datos por segundo). En la recepción, se volcaban los datos recolectados a un archivo de texto para luego analizarlos detenidamente. Del análisis hecho sobre los datos obtenidos se pudo extraer la posición geográfica del vehículo (en latitud y longitud a cada segundo), la hora UTC y la velocidad. Los resultados obtenidos fueron los siguientes:

- En la primera pasada, el **primer punto** registrado satisfactoriamente fue **38° 00.6140' S** (sur), **057° 35.0815' W** (oeste) a las **14:12:12.000 horas UTC**, con una velocidad de **20,80 Km/h**. Cuando se dice “satisfactoriamente” es debido a que en la recepción de los primeros datos se registran pérdidas de paquetes, hasta que se ingresa plenamente en lo que parece ser la zona de alcance. El tiempo de finalización, que corresponde a la ubicación **38° 00.7289' S, 057° 34.9296' W**, fue de **14:12:56.000 horas UTC**.
- En la segunda pasada, el **primer punto** registrado satisfactoriamente fue **38° 00.6249' S, 057° 35.0638' W** a las **14:17:15.000 horas UTC**, con una velocidad de **29,85 Km/h**. El tiempo de finalización fue de **14:17:45.000 horas UTC** para el punto **38° 00.7293' S, 057° 34.9270' W**.
- En la tercera pasada, el **primer punto** registrado satisfactoriamente fue **38° 00.6308' S, 057° 35.0553' W** a las **14:22:51.000 horas UTC**, con una velocidad de **36,19 Km/h**. El tiempo de finalización fue de **14:23:13.000 horas UTC** para el punto **38° 00.7266' S, 057° 34.9291' W**.

A continuación se presenta un gráfico realizado con Google Earth indicando los tres puntos registrados en cada pasada, siendo en verde la primera pasada, en amarillo la segunda y en rojo la tercera.



Ilustración 45: gráfico de Google Earth con los puntos iniciales y finales de cada pasada de prueba.

### 3.4.2. Cálculos y estimaciones

Se puede ver claramente que la **zona de alcance es mayor a 300 metros**. Hay que tener en cuenta que el enlace producido en el ensayo tiene un espacio libre, por lo que un enlace con obstáculos debiera ser sustancialmente más reducido. Considerando que, según las especificaciones, la velocidad máxima de transferencia de datos del XBee es de 250Kbps y el tiempo de duración de cada pasada, se puede estimar un caudal de datos para cada velocidad:

- En el primer caso se registró un tiempo de enlace de 44 segundos y se recorrieron (según la “regla” que permite medir en el Google Earth) 300 metros, aproximadamente. Para una velocidad promedio de **24,5 km/h**, se estima un caudal de datos de **11 Mb**.
- En el segundo caso se registró un tiempo de enlace de 30 segundos y se recorrieron 276 metros, aproximadamente. Para una velocidad promedio de **33,12 km/h**, se estima un caudal de datos de aproximadamente **7,5 Mb**.
- En el tercer caso se registró un tiempo de enlace de 22 segundos y se recorrieron 250 metros, aproximadamente. Para una velocidad promedio de **40,91 km/h**, se estima un caudal de datos de aproximadamente **5,5 Mb**.

Para estimar un tiempo de establecimiento se necesita plantear algunas ecuaciones dado que solo se conoce los puntos donde se consolidan los enlaces y no el punto de inicio de asociación. Asignamos como “*a*” la distancia que hay entre el primer punto registrado para la primera pasada, y el punto de inicio de asociación. Análogamente se hace lo mismo con “*b*” y el primer punto de la segunda pasada, “*c*” y el primer punto de la tercera pasada y designamos a “*ta*” como el tiempo de asignación. Los datos son, las distancias entre cada punto y las velocidades en cada punto (en m/s). Se supone velocidad constante en cada intervalo y que todos los puntos tienen un mismo punto de inicio de asociación. Luego:

- |                      |                            |             |
|----------------------|----------------------------|-------------|
| • $b-a = 32\text{m}$ | • $va = 6,81 \text{ m/s}$  | • $va=a/ta$ |
| • $c-a = 48\text{m}$ | • $vb = 9,2 \text{ m/s}$   | • $vb=b/ta$ |
| • $c-b = 16\text{m}$ | • $vc = 11,36 \text{ m/s}$ | • $vc=c/ta$ |

Se puede plantear la siguiente ecuación:

$$\begin{aligned}a + (b - a) &= b \\ta \cdot va + (b - a) &= ta \cdot vb \\ta(vb - va) &= b - a \\ta &= \frac{b - a}{vb - va}\end{aligned}$$

Si se usa las combinaciones de los tres puntos se puede obtener tres valores de *ta*, con los cuales calcularemos un promedio para obtener un valor final de *ta*:

- $ta1 = 13,39$  segundos
- $ta2 = 10,55$  segundos
- $ta3 = 7,71$  segundos

**$ta \approx 10,45$  segundos**

Si se considera el tiempo de establecimiento, se puede estimar la zona de alcance del enlace para las condiciones del ensayo. Sumando la distancia al primer paquete recibido y la distancia al punto estimado de inicialización de asociación se calcula una estimación del área de cobertura (" $d$ ") para cada pasada y promediaremos los valores obtenidos, luego:

- $da = 332m + ta.va = 403$  metros
- $db = 300m + ta.vb = 396$  metros
- $da = 283m + ta.va = 401$  metros

**$d \approx 400$  metros**

Con los datos obtenidos se puede hacer una extrapolación para una situación con una velocidad de 60 km/h (velocidad máxima en avenidas), de la siguiente manera:

- Distancia efectiva de enlace =  $400 \text{ metros} - ta.V = 400m - 10,45s.16,67m/s = \mathbf{225,8 \text{ m}}$ .
- Volumen de datos a 60 km/h =  $(225,8m/16,67s).250Kbps = \mathbf{3,4 \text{ Mb}}$ .

## 4. CONCLUSIONES Y LÍNEAS FUTURAS DE TRABAJO

El uso de dispositivos de sensado móviles como método complementario para el relevamiento de información útil para la planificación del tránsito y el mantenimiento de las vías permitirá la obtención de un mayor y mejor caudal de información para la planificación del tráfico y el mantenimiento de las vías.

La información aportada por los dispositivos, que es actualizada continuamente, permitirá hacer estudios estadísticos tanto sobre la **ubicación geográfica de los problemas** como su **distribución temporal y grado de ocurrencia**.

Es recomendable, para un mejor rendimiento del sistema, que los dispositivos concentradores estén ubicados en zonas donde exista un gran volumen de tránsito, como por ejemplo una intersección entre avenidas. Esto tiene como ventaja el incremento de la probabilidad de que un móvil se tope con un concentrador, sin la necesidad de establecer un recorrido determinado o realizar viajes para reportar la información a los concentradores. Esto trae aparejada la desventaja de que los vehículos circulan a mayor velocidad en una avenida, disminuyendo el caudal de datos neto del enlace. En contraste con lo anteriormente mencionado, se pueden aprovechar las intersecciones con semáforos que, al presentar luz roja, generan que el vehículo se detenga, lo que permite que exista un tiempo de enlace mayor a lo normal. Además, las avenidas tienen mucho espacio abierto, lo que favorece al enlace entre los dispositivos e incrementa el alcance máximo.

El rango de uso potencial del proyecto es de hecho muy variado y altamente escalable, permitiendo el agregado en forma paulatina de nuevas funcionalidades con completa compatibilidad hacia atrás. Se puede considerar el sistema totalmente programable y expandible, siendo posible un desarrollo continuo en el código de operación de los dispositivos móviles y permitiendo la actualización de los mismos cuando sea necesario.

El producto desarrollado cumplió con las especificaciones de diseño, siendo asimismo probado experimentalmente.

Para el trabajo a futuro se propone los siguientes tópicos:

- Desarrollo y pruebas de los distintos dispositivos de sensado para el vehículo que ofrece el mercado local, como por ejemplo acelerómetros.
- Desarrollo de un subsistema y los respectivos algoritmos para la Unidad de Monitoreo.
- Desarrollo de una interfaz a base de protocolo USB que permita interconectar ambos dispositivos directamente a una PC, con el fin de facilitar el acceso al código de programación de los mismos.

- Desarrollo de una página dinámica que pueda acceder a la base de datos e interprete los datos recolectados. Dicho software debe mostrar, mediante una interfaz gráfica, un mapa de la ciudad y sus calles donde se puedan visualizar los puntos registrados por los dispositivos móviles. También debe permitir filtrar los datos según el o los parámetros de interés.

## 5. BIBLIOGRAFÍA

- [1] A.E. Traffic Technologies Ltd, <http://www.ae-traffic.com>
- [2] American Traffic S.A., «ATSA» 2016. <http://www.americantrafficsa.com.ar>
- [3] DesignSpark PCB software, 2016. <https://www.rs-online.com/designspark/pcb-software>
- [4] EAGLE PCB Design, 2016. <https://cadsoft.io>
- [5] Enterprise Flasher Company, 2010. <http://www.enterpriseflasher.com>
- [6] Fundación Metropolitana, «Planificación y participación para la Gran Buenos Aires.» 2016. <http://metropolitana.org.ar>
- [7] Grupo del Banco Mundial, «Banco de datos del Banco Mundial.» 2016. <http://databank.bancomundial.org>
- [8] IEEE | IEEE 802.15 WPAN™ Task Group 4 (TG4), 2016. <http://www.ieee802.org/15/pub/TG4.html>
- [9] IEEE | IEEE 802.11™ WIRELESS LOCAL AREA NETWORKS, 2016. <http://www.ieee802.org/11/>
- [10] Ingeniería en Relevamientos Viales S.A., «IRV S.A.» <http://www.irvsa.com.ar>
- [11] Iteris, Inc., «Innovation for better mobility.» 2016. <http://www.iteris.com>
- [12] Microchip Technology Inc, 2016. <http://www.microchip.com>
- [13] MySQL, 2016. <http://www.mysql.com>
- [14] Observatorio Nacional de Datos de Transporte, «ONDaT» 2015. <http://ondat.fra.utn.edu.ar>
- [15] Peek Traffic Corporation, 2015. <https://www.peaktraffic.com>
- [16] Qualcomm Technologies, Inc, 2016. <https://www.qualcomm.com>
- [17] SAE International, 2012. [http://standards.sae.org/j2602/2\\_200509](http://standards.sae.org/j2602/2_200509)
- [18] The Apache Software Foundation, 2016. <https://www.apache.org>
- [19] The ZigBee Alliance | Control your World, 2016. <http://www.zigbee.org>
- [20] USGlobalSat Inc, 2014. [http://usglobalsat.com/downloads/NMEA\\_commands.pdf](http://usglobalsat.com/downloads/NMEA_commands.pdf)
- [21] Vaisala, «A global leader in environmental and industrial measurement.» 2016. <http://www.vaisala.com>
- [22] WAMPSEVER, 2016. <http://www.wampserver.es>
- [23] Wavetronix LLC., «Making the world's traffic safer and more efficient.» 2016. <http://www.wavetronix.com>
- [24] <http://www.elgps.com/mensajes/sobrewaas.html>
- [25] <https://es.wikipedia.org/wiki/Supercondensador>