



UNIVERSIDAD DE VALLADOLID

E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE MÁSTER

GRADO EN INGENIERÍA DE TECNOLOGÍAS  
DE TELECOMUNICACIÓN

# **Integración, testado y propuestas de mejora de una unidad de comunicación V2V**

Autor:

**D. Roberto Herreras Babón**

Tutor:

**D. Juan Carlos Aguado Manzano**

Valladolid, 25 de junio de 2019



---

**TÍTULO:** **Integración, testado y propuestas de mejora de una unidad de comunicación V2V**

**AUTOR:** D. Roberto Herreras Babón

**TUTOR:** D. Juan Carlos Aguado Manzano

**DEPARTAMENTO:** **Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

---

### **TRIBUNAL**

---

**PRESIDENTE:** D. Ignacio de Miguel Jiménez

**VOCAL:** D. Ramón de la Rosa Steinz

**SECRETARIO:** D. Ramón Durán Barroso

**SUPLENTE:** Dña. Patricia Fernández Reguero

**SUPLENTE:** D. Manuel Javier Aguiar Pérez

**SUPLENTE:** D. Jaime Gómez Gil

---

**FECHA:** 25 de junio de 2019

**CALIFICACIÓN:**

---



## Resumen del TFM

El propósito de este proyecto es la integración dos unidades de a bordo de comunicación entre vehículos según el estándar ETSI ITS G5, así como la caracterización de los componentes activos en la comunicación y la realización de cálculos teóricos y procesos experimentales para garantizar el correcto funcionamiento del sistema. El software utilizado para la comunicación V2X ha sido software libre desarrollado por el grupo CSS Labs de la Universidad de Paderborn, Alemania.

Durante el proyecto, se ha caracterizado la unidad de a bordo mediante experimentos controlados en el laboratorio y finalmente, se ha realizado un experimento de campo final, recreando un entorno real de funcionamiento, que nos permitió medir la fiabilidad de las muestras GPS obtenidas, estimando el error introducido por el sistema GPS y el error acumulativo del odómetro del automóvil, además de conocer si hay relación entre la distancia entre vehículos y los mensajes CAM no recibidos entre las unidades de a bordo.

## Abstrat

The purpose of this project is the integration of two on-board communication units in two vehicles according to the ETSI ITS G5 standard, as well as the characterization of the active components in the communication and the realization of theoretical calculations and experimental processes to guarantee the correct performance of the system. The software used for V2X communication has been free software developed by the CSS Labs group of the University of Paderborn, Germany.

During the project, the on-board unit has been characterized doing controlled experiments in the laboratory. Finally, a final experiment was performed, to determinate how well our system works in a real operating environment, allowing us to measure the reliability of the GPS samples, determining the error committed by the GPS system and the accumulative error from the vehicle odometer, as well as proving if there is a relationship between the distance between vehicles and the CAM messages that have not been received by the on-board units.

## Palabras clave

Comunicación vehículo a vehículo, V2X, unidad de a bordo, unidad de comunicación embarcada.



## Agradecimientos

Agradecer al Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática, y especialmente a Juan Carlos Aguado, mi tutor de este TFM, por su ayuda, disponibilidad y dedicación a lo largo de este trabajo.

También quiero agradecer a mi familia y amigos, por estar siempre a mi lado y por toda su ayuda durante estos años. Quiero dar mención especial a Dune Márquez, por ayudarme en todas y cada uno de los experimentos donde he necesitado su ayuda, y a Luis Antonio Domínguez por ofrecerse de ayudante adicional para el experimento final.





# Contenido

<b>1.</b>	<b>INTRODUCCIÓN .....</b>	<b>1</b>
1.1	MOTIVACIÓN .....	1
1.2	OBJETIVOS.....	4
1.3	MEDIOS .....	5
1.4	ESTRUCTURA DE LA MEMORIA.....	5
<b>2.</b>	<b>ESTADO DEL ARTE.....</b>	<b>7</b>
<b>3.</b>	<b>INTEGRACIÓN DEL SOFTWARE.....</b>	<b>13</b>
3.1	INSTALACIÓN DEL SISTEMA OPERATIVO Y DRIVERS .....	13
3.1.1	<i>Sistema operativo.....</i>	14
3.1.2	<i>OpenC2X.....</i>	18
3.1.3	<i>GPS bluetooth.....</i>	22
3.1.4	<i>OBD2 bluetooth.....</i>	23
3.1.5	<i>Gamepad bluetooth.....</i>	25
3.2	AUTOMATIZAR EL ARRANQUE .....	26
3.3	RECOPIACIÓN DE DATOS.....	28
3.4	INCIDENCIAS DE LOS ORDENADORES .....	29
<b>4.</b>	<b>CARACTERIZACIÓN DE LOS ELEMENTOS DEL SISTEMA .....</b>	<b>31</b>
4.1	CÁLCULO TEÓRICO DE LA DISTANCIA MÁXIMA DE COMUNICACIÓN .....	31
4.2	CARACTERIZACIÓN DE LOS ELEMENTOS DEL SISTEMA EN LABORATORIO.....	32
4.2.1	<i>Tipos de antenas disponibles.....</i>	32
4.2.2	<i>Estimación de la ganancia de las antenas.....</i>	33
4.2.3	<i>Coefficiente de reflexión de potencia <math>s_{11}</math>.....</i>	41
4.2.4	<i>Potencia real transmitida por la tarjeta de red.....</i>	44
4.2.5	<i>Modulación utilizada por la tarjeta.....</i>	48
4.3	DISTANCIA MÁXIMA DE COMUNICACIÓN TEÓRICA .....	56
4.3.1	<i>Valores modificados para el cálculo de la máxima distancia de comunicación.....</i>	57
4.3.2	<i>Cálculo de la máxima distancia de comunicación.....</i>	58
4.4	ESTIMACIÓN DE LA DISTANCIA MÁXIMA DE COMUNICACIÓN MEDIANTE PRUEBAS DE CAMPO.....	59
4.4.1	<i>Experimento interior de 120 metros.....</i>	60
4.4.2	<i>Comunicación en campo abierto con distancia fija.....</i>	64
4.4.3	<i>Estudio de la variación de la posición estimada por el GPS en punto fijo.....</i>	67
4.4.4	<i>Comunicación en campo abierto con distancias variables.....</i>	71
4.5	DISCUSIÓN DE RESULTADOS.....	75
<b>5.</b>	<b>DEMOSTRACIÓN DEL SISTEMA: CASO DE USO .....</b>	<b>79</b>
5.1	REALIZACIÓN DEL EXPERIMENTO.....	79
5.2	PROCESADO DE DATOS .....	83
5.2.1	<i>Script Main.m.....</i>	84
5.2.2	<i>Script cargarDatos.m.....</i>	84
5.2.3	<i>Script GPS.m.....</i>	85
5.2.4	<i>Script OBD.m.....</i>	85
5.2.5	<i>Script Velocidad.m.....</i>	86
5.2.6	<i>Script Ruta.m.....</i>	86
5.2.7	<i>Script Gmaps.m.....</i>	89
5.2.8	<i>Script Cams.m.....</i>	89
5.2.9	<i>Script CalculoPosicionGPS.m.....</i>	90
5.2.10	<i>Script OBDLimitado.m.....</i>	94
5.2.11	<i>Script DistanciaEntreCochesGPS.m.....</i>	94

<b>6.</b>	<b>CONCLUSIONES Y LÍNEAS FUTURAS .....</b>	<b>99</b>
6.1	CONCLUSIONES.....	99
6.2	PROPUESTAS DE MEJORA .....	99
6.3	LÍNEAS FUTURAS .....	100
<b>7.</b>	<b>BIBLIOGRAFÍA .....</b>	<b>101</b>
<b>8.</b>	<b>ANEXO DE CÓDIGO .....</b>	<b>107</b>
8.1	INTEGRACIÓN DEL SOFTWARE .....	107
8.1.1	<i>Gamepad</i> .....	107
8.1.2	<i>init.sh</i> .....	109
8.1.3	<i>Copiar.sh</i> .....	110
8.1.4	<i>Borrar.sh</i> .....	110
8.1.5	<i>Ssh.sh</i> .....	110
8.2	CARACTERIZACIÓN DE LOS ELEMENTOS DEL SISTEMA.....	111
8.2.1	<i>Coefficiente de reflexión de potencia <math>s_{11}</math></i> .....	111
8.2.2	<i>Comunicación en campo abierto con distancia fija</i> .....	113
8.3	DEMOSTRACIÓN DEL SISTEMA: CASO DE USO .....	115
8.3.1	<i>Main.m</i> .....	115
8.3.2	<i>cargarDatos.m</i> .....	117
8.3.3	<i>GPS.m</i> .....	118
8.3.4	<i>OBD.m</i> .....	119
8.3.5	<i>Velocidad.m</i> .....	120
8.3.6	<i>Ruta.m</i> .....	121
8.3.7	<i>Gmaps.m</i> .....	122
8.3.8	<i>Cams.m</i> .....	123
8.3.9	<i>CalculoPosicionGPS.m</i> .....	124
8.3.10	<i>OBDLimitado.m</i> .....	127
8.3.11	<i>Script DistanciaEntreCochesGPS.m</i> .....	128
8.3.12	<i>DistanciaEntreCochesGPS.m</i> .....	129
<b>9.</b>	<b>DATASHEET ANTENA PLANA GSA.8859 .....</b>	<b>131</b>
<b>10.</b>	<b>DATASHEET GPLIM236 .....</b>	<b>155</b>

## Índice figuras

FIGURA 1. COMPARATIVA DE ACCIDENTES 2017 Y 2018 [ 1 ] .....	1
FIGURA 2. ESTADÍSTICAS AUTOMOVILÍSTICAS EN ESPAÑA [ 1 ] .....	2
FIGURA 3. VEHÍCULO AUTÓNOMO TESLA AUTOPILOT .....	2
FIGURA 4. DETECCIÓN DE OBSTÁCULOS MEDIANTE RADAR [ 2 ].....	3
FIGURA 5. PARÁMETROS GPS UTILIZADOS POR UNA OBU [ 17 ] .....	8
FIGURA 6. DETECTORES DE SITUACIONES DE PELIGRO [ 17 ] .....	8
FIGURA 7. VN4610 NETWORK INTERFACE [ 20 ] .....	9
FIGURA 8. CANALES COMUNICACIÓN V2X [ 20 ] .....	10
FIGURA 9. ESCENARIOS DE USO PARA CADA TECNOLOGÍA [ 24 ] .....	11
FIGURA 10. ETAPAS DE EVOLUCIÓN V2X [ 25 ] .....	11
FIGURA 11. ORDENADORES UTILIZADOS.....	14
FIGURA 12. CÓDIGO SCRIPT OCB.SH .....	18
FIGURA 13. CÓDIGO TERMINALATTRIBUTES MODIFICADO .....	19
FIGURA 14. NUEVA BIBLIOTECA AÑADIDA EN OBD2SERVICE.CPP .....	20
FIGURA 15. MODIFICACIÓN EN OBD2SERVICE::SENDTOSERVICES.....	20
FIGURA 16. NUEVA BIBLIOTECA AÑADIDA EN GPSSERVICE.CPP .....	20
FIGURA 17. MODIFICACIÓN EN GPSSERVICE::SENDTOSERVICES .....	20
FIGURA 18. CORRECCIÓN EN LAS UNIDADES DEL OBD .....	20
FIGURA 19. MODIFICACIÓN EN EL CÓDIGO READRPM().....	21
FIGURA 20. MODIFICACIÓN EN EL CÓDIGO OBD2.SETRPM().....	21
FIGURA 21. MODIFICACIÓN EN SERIALPORT::READSPEED() .....	21
FIGURA 22.(A) DISPOSITIVO OBD (B) TOMA OBD.....	24
FIGURA 23. DISPOSITIVO GAMEPAD .....	25
FIGURA 24. SCRIPTS DESARROLLADOS .....	27
FIGURA 25. CAMBIAR LA HORA DE LOS ORDENADORES CON GPS .....	28
FIGURA 26. (A) ANTENA PLANA GSA.8859 (B) ANTENA WIFI CUADRADA (C) ANTENA WIFI REDONDA .....	33
FIGURA 27. GANANCIA DE LA ANTENA PLANA SEGÚN EL DATASHEET .....	34
FIGURA 28. EJES DE LA ANTENA .....	34
FIGURA 29. CONFIGURACIÓN DE LA ANTENA EN EL VEHÍCULO .....	35
FIGURA 30. FORMA DE LA ANTENA WIFI.....	35
FIGURA 31. DIPOLO SIMPLE EN 4NEC2 .....	36
FIGURA 32. PATRÓN DE CAMPO LEJANO DE LA FRECUENCIA BAJO ESTUDIO .....	37
FIGURA 33. IMAGEN 3D DEL DIAGRAMA DE CAMPO LEJANO .....	37
FIGURA 34. GANANCIA DEL DIPOLO EN ESPACIO LIBRE.....	38

FIGURA 35. DIAGRAMA DE CAMPO LEJANO CON SUELO .....	38
FIGURA 36. DIAGRAMA 3D DE LA CONFIGURACIÓN CON SUELO .....	39
FIGURA 37. MODELO 3D DEL AUTOMÓVIL.....	39
FIGURA 38. PERFIL DEL MODELO 3D DEL AUTOMÓVIL .....	40
FIGURA 39. GANANCIA CON EL AUTOMÓVIL .....	40
FIGURA 40. DIAGRAMA DE RADIACIÓN EN 3D .....	41
FIGURA 41. ANALIZADOR DE REDES .....	42
FIGURA 42. COEFICIENTE $S_{11}$ EN FUNCIÓN DE LA FRECUENCIA.....	43
FIGURA 43. COEFICIENTE $S_{11}$ EN EL RANGO DE FRECUENCIAS DE INTERÉS Y CANALES UTILIZADOS.....	43
FIGURA 44. TARJETA DE RED ATHEROS ATH9K AR5BHB116 .....	44
FIGURA 45. ANALIZADOR DE ESPECTROS ROHDE&SCHARZ.....	44
FIGURA 46. ALARGADOR CON CONEXIONES MDA .....	45
FIGURA 47. POTENCIAS TRANSMITIDAS POR EL ORDENADOR A CON DETECTOR RMS.....	46
FIGURA 48. POTENCIAS TRANSMITIDAS POR EL ORDENADOR A CON DETECTOR DE POTENCIA MÁXIMA .....	48
FIGURA 49. IMPEDANCIAS ADAPTADAS.....	49
FIGURA 50. IMPEDANCIA ADAPTADA COLOCADA EN EL ORDENADOR.....	49
FIGURA 51. TARJETA USRP-210 .....	50
FIGURA 52. ESQUEMA DEMODULADOR DE GNURADIO .....	51
FIGURA 53. CONFIGURACIÓN DEL EXPERIMENTO PARA LA DETECCIÓN DE LA MODULACIÓN .....	52
FIGURA 54. ESQUEMA DEL EXPERIMENTO REALIZADO .....	52
FIGURA 55. MODULACIÓN OBTENIDA CON POTENCIA MÍNIMA DE TRANSMISIÓN.....	53
FIGURA 56. ATENUADORES ADAPTADOS.....	53
FIGURA 57. CONFIGURACIÓN DE LOS ATENUADORES.....	54
FIGURA 58. ATENUACIÓN REQUERIDA PARA EMPEORAR EL CANAL .....	54
FIGURA 59. MODULACIÓN SIMULANDO UNA DISTANCIA GRANDE ENTRE LOS ORDENADORES .....	55
FIGURA 60. MODULACIÓN QPSK.....	55
FIGURA 61. RELACIÓN MODULACIÓN-SENSIBILIDAD DEL RECEPTOR [ 50 ] .....	56
FIGURA 62. DISTANCIA MÁXIMA EN FUNCIÓN DE LA MODULACIÓN .....	59
FIGURA 63. DISTANCIA PRUEBA INTERIOR DE 120 METROS.....	60
FIGURA 64. ESCENARIO DE LA PRUEBA INTERIOR DE 120 METROS .....	61
FIGURA 65. CONFIGURACIÓN UTILIZADA PARA LA PRUEBA INTERIOR DE 120 METROS.....	61
FIGURA 66. PINGS ENVIADOS .....	62
FIGURA 67. DISTANCIA PRUEBA CON ANTENAS WIFI.....	63
FIGURA 68. RESULTADOS OBTENIDOS TRAS REPETIR EL EXPERIMENTO.....	63
FIGURA 69. CONFIGURACIÓN DEL ORDENADOR INTERIOR .....	64
FIGURA 70. ORDENADOR Y GPS DEL VEHÍCULO .....	64

FIGURA 71. DISPOSICIÓN DE LAS ANTENAS PLANAS DENTRO DEL VEHÍCULO .....	65
FIGURA 72. VEHÍCULO EN VISIÓN DIRECTA CON EL RECEPTOR .....	65
FIGURA 73. PLANO DE LA POSICIÓN DE LOS TRANSMISORES Y DISTANCIA TOTAL DE LA PRUEBA .....	66
FIGURA 74. RESULTADO DE PINGS CON COMUNICACIÓN NO ESTABLECIDA .....	66
FIGURA 75. DISPOSICIÓN DEL ORDENADOR EN EL VEHÍCULO .....	67
FIGURA 76. DIAGRAMA DE HAVERSINE [ 53 ].....	68
FIGURA 77. ERROR GPS POR MUESTRA.....	68
FIGURA 78. ERROR DEL GPS SOBRE MAPA .....	69
FIGURA 79. ERROR MÁXIMO EN EL POSICIONAMIENTO GPS POR PERCENTILES. EN VERDE EL 5%, AMARILLO EL 50%, ROJO EL 95% Y AZUL EL 100% .....	69
FIGURA 80. CONFIGURACIÓN DEL EXPERIMENTO (VISTA EXTERIOR) .....	70
FIGURA 81. CONFIGURACIÓN DEL EXPERIMENTO (VISTA INTERIOR).....	70
FIGURA 82. ERROR EN LA DISTANCIA GPS.....	71
FIGURA 83. VEHÍCULOS UTILIZADOS EN LA PRUEBA COLOCADOS SEGÚN LA DISPOSICIÓN QUE TENDRÁN CAMBIANDO LA DISTANCIA ENTRE ELLOS .....	72
FIGURA 84. LÁSER DE MEDICIÓN UTILIZADO.....	72
FIGURA 85. CONFIGURACIÓN DE LAS ANTENAS PLANAS.....	73
FIGURA 86. POSICIÓN DEL ORDENADOR Y GPS EN LA PARTE TRASERA DEL VEHÍCULO .....	73
FIGURA 87. DISTANCIA MÁXIMA DE COMUNICACIÓN ALCANZADA EXPERIMENTALMENTE EN LA PRUEBA DE CAMPO .....	74
FIGURA 88. PINGS ENVIADOS CON ANTENAS PLANAS .....	74
FIGURA 89. PINGS ENVIADOS CON ANTENAS WIFI.....	75
FIGURA 90. RUTA DISEÑADA PARA EL CASO DE USO. ....	80
FIGURA 91. INSTALACIÓN DEL MINIORDENADOR EN EL SALPICADERO DEL VEHÍCULO A .....	81
FIGURA 92. INSTALACIÓN DEL MINIORDENADOR EN LA BANDEJA TRASERA DEL AUTOMÓVIL B .....	81
FIGURA 93. ZONA CERCANA DESDE EL PUNTO DE VISTA COCHE A .....	82
FIGURA 94. ZONA LEJANA DESDE EL PUNTO DE VISTA COCHE A .....	82
FIGURA 95. ZONA CERCANA DESDE EL PUNTO DE VISTA COCHE B .....	82
FIGURA 96. ZONA DE MÁXIMA VELOCIDAD DESDE EL PUNTO DE VISTA COCHE B.....	83
FIGURA 97. CUENTAKILÓMETROS DEL AUTOMÓVIL .....	83
FIGURA 98. CONSOLA CON RESUMEN DE LA PRUEBA .....	84
FIGURA 99. DISTANCIA TOTAL DE CADA GPS .....	85
FIGURA 100. DISTANCIA ACUMULADA POR EL OBD EN TODA LA PRUEBA .....	85
FIGURA 101. DISTANCIA TOTAL OBTENIDA POR EL OBD .....	86
FIGURA 102. VELOCIDADES ALCANZADAS DURANTE LA PRUEBA.....	86
FIGURA 103. RUTA REGISTRADA POR EL GPS B .....	87
FIGURA 104. DIFERENCIAS ENTRE GPS RECIBIDO Y RUTA GMAPS .....	87

FIGURA 105. RUTAS GPS SIMILARES EN LOS DATOS OBTENIDOS .....	88
FIGURA 106. ERRORES AÑADIDOS POR LOS SEMÁFOROS .....	89
FIGURA 107. DISTANCIA REAL DADA POR GMAPS .....	89
FIGURA 108. RELACIÓN ENTRE CAMS ENVIADOS Y RECIBIDOS POR CADA ORDENADOR .....	90
FIGURA 109. ESQUEMA DEL TEOREMA DE LOS TRIÁNGULOS SEMEJANTES.....	91
FIGURA 110. TRIANGULO RECTÁNGULO PARA COORDINAR TIEMPO GPS-OBD.....	92
FIGURA 111. ERROR ENTRE POSICIÓN ESTIMADA POR EL OBD Y POSICIÓN GPS OBTENIDA .....	93
FIGURA 112. ERROR MEDIO Y MÁXIMO ENTRE GPS Y ESTIMACIÓN DE POSICIÓN CON EL OBD .....	94
FIGURA 113. DISTANCIA ACUMULADA HASTA 3 KILÓMETROS.....	94
FIGURA 114. RELACIÓN TEMPORAL ENTRE MUESTRAS GPS A Y GPS B.....	95
FIGURA 115. AJUSTE DE LA POSICIÓN DEL GPS A A TIEMPO GPS B .....	95
FIGURA 116. DISTANCIA ENTRE COCHES CON GPS .....	96
FIGURA 117. ERROR INTRÍNSECO AL SISTEMA GPS CON COCHES APARCADOS EN PARALELO .....	97
FIGURA 118. DISTANCIA ENTRE COCHES CON FILTRO DE MEDIANA.....	97
FIGURA 119. ERROR INTRODUCIDO POR EL RUIDO IMPULSIVO EN LAS MUESTRAS GPS.....	98
FIGURA 120. ERROR INTRODUCIDO POR EL RUIDO IMPULSIVO EN LAS MUESTRAS GMAPS Y OBD .....	98
FIGURA 121. CÓDIGO GAMEPAD (1/2).....	107
FIGURA 122. CÓDIGO GAMEPAD (2/2).....	108
FIGURA 123. SCRIPT INIT.SH (1/2).....	109
FIGURA 124. SCRIPT INIT.SH (2/2).....	110
FIGURA 125. SCRIPT COPIAR.SH .....	110
FIGURA 126. SCRIPT BORRAR.SH.....	110
FIGURA 127. SCRIPT SSH.SH .....	110
FIGURA 128. CÓDIGO COEFICIENTE DE REFLEXIÓN $S_{11}$ (1/2) .....	111
FIGURA 129. CÓDIGO COEFICIENTE DE REFLEXIÓN $S_{11}$ (2/2) .....	112
FIGURA 130. CÓDIGO PUNTOS DEL GPS (1/2).....	113
FIGURA 131 CÓDIGO COEFICIENTE DE REFLEXIÓN $S_{11}$ (2/2) .....	114
FIGURA 132. CÓDIGO FUNCIÓN MAIN (1/2).....	115
FIGURA 133. CÓDIGO FUNCIÓN MAIN (2/2).....	116
FIGURA 134. CÓDIGO FUNCIÓN CARGARDATOS .....	117
FIGURA 135. CÓDIGO FUNCIÓN GPS .....	118
FIGURA 136. CÓDIGO FUNCIÓN OBD .....	119
FIGURA 137. CÓDIGO FUNCIÓN VELOCIDAD .....	120
FIGURA 138. CÓDIGO FUNCIÓN RUTA.....	121
FIGURA 139. CÓDIGO FUNCIÓN GMAPS.....	122
FIGURA 140. CÓDIGO FUNCIÓN CAMS .....	123

FIGURA 141. CÓDIGO FUNCIÓN CALCULOPOSICIONGPS (1/3) .....	124
FIGURA 142. CÓDIGO FUNCIÓN CALCULOPOSICIONGPS (2/3) .....	125
FIGURA 143. CÓDIGO FUNCIÓN CALCULOPOSICIONGPS (3/3) .....	126
FIGURA 144. CÓDIGO FUNCIÓN OBDLIMITADO .....	127
FIGURA 145. CÓDIGO FUNCIÓN FIGURAS .....	128
FIGURA 146. CÓDIGO FUNCIÓN DISTANCIAENTRECOCHESGPS (1/2) .....	129
FIGURA 147. CÓDIGO FUNCIÓN DISTANCIAENTRECOCHESGPS (2/2) .....	130





## Índice tablas

TABLA 1. POTENCIAS TRANSMITIDAS POR EL ORDENADOR A CON DETECTOR RMS. ....	46
TABLA 2. POTENCIAS TRANSMITIDAS POR EL ORDENADOR A CON DETECTOR DE POTENCIA MÁXIMA	47
TABLA 3. RELACIÓN ENTRE MODULACIÓN DE LA SEÑAL Y SENSIBILIDAD DEL RECEPTOR.....	56
TABLA 4. GANANCIA SEGÚN EL FABRICANTE FRENTE GANANCIA CALCULADA POR SIMULACIÓN.....	57
TABLA 5. COEFICIENTE DE ACOPLLO TEÓRICO FRENTE AL COEFICIENTE CALCULADO EXPERIMENTALMENTE.....	57
TABLA 6. POTENCIA DE LA TARJETA SEGÚN EL FABRICANTE FRENTE LA POTENCIA EXPERIMENTAL ....	57
TABLA 7. RELACIÓN ENTRE MODULACIÓN Y SENSIBILIDAD DE RECEPCIÓN.....	58
TABLA 8. DISTANCIA MÁXIMA TEÓRICA SIN DATOS EXPERIMENTALES.....	58
TABLA 9. SENSIBILIDAD DE RECEPCIÓN FRENTE CONSTELACIÓN.....	58
TABLA 10. DISTANCIA TEÓRICA COND ATOS EXPERIMENTALES .....	59
TABLA 11. RESULTADOS ESTADÍSTICOS GPS.....	71
TABLA 12. RESUMEN DE LAS PRUEBAS DE CAMPO REALIZADAS .....	77
TABLA 13. NÚMERO DE MENSAJES CAM ENVIADOS POR A Y RECIBIDOS POR B .....	90
TABLA 14. NÚMERO DE MENSAJES CAM RECIBIDOS POR A Y ENVIADOS POR B .....	90



# 1. Introducción

## 1.1 Motivación

Los consumidores demandan cada vez mayores medidas de seguridad a la hora de adquirir un vehículo nuevo. Desde la introducción de medidas de seguridad pasivas como son el cinturón de seguridad, el ABS y otros, la tecnología ha avanzado añadiéndose nuevas medidas de seguridad fundamentalmente de tipo activo y formadas principalmente por sensores, radares, limitadores de velocidad, etc. Estas nuevas medidas ayudan a evitar accidentes, pero todavía queda un largo camino en el que se pueden seguir reduciendo el número de ellos.

Durante el año 2018, se han producido 1072 accidentes mortales en las carreteras interurbanas españolas, en los cuales han fallecido un total de 1180 personas y 4515 pasajeros han requerido recibir atención médica (Figura 1). Estos valores suponen una reducción de un 1,5% en el número de fallecidos y un 7,6% en los heridos graves. En contra, ha aumentado un 0,7% el número de accidentes mortales. [ 1 ]

Sin embargo, la cantidad de desplazamientos ha aumentado un total de 7,8 millones más que el año 2017, así como el número de coches registrados (aumento del 3%) y el censo de conductores (crecimiento del 1%).

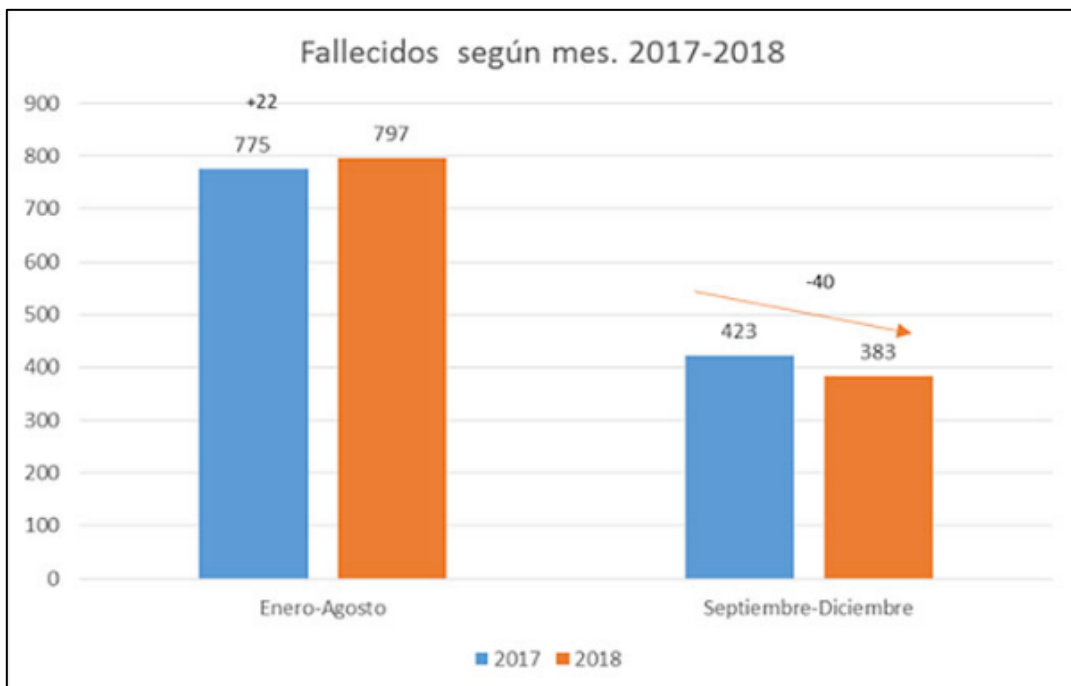


Figura 1. Comparativa de accidentes 2017 y 2018 [ 1 ]

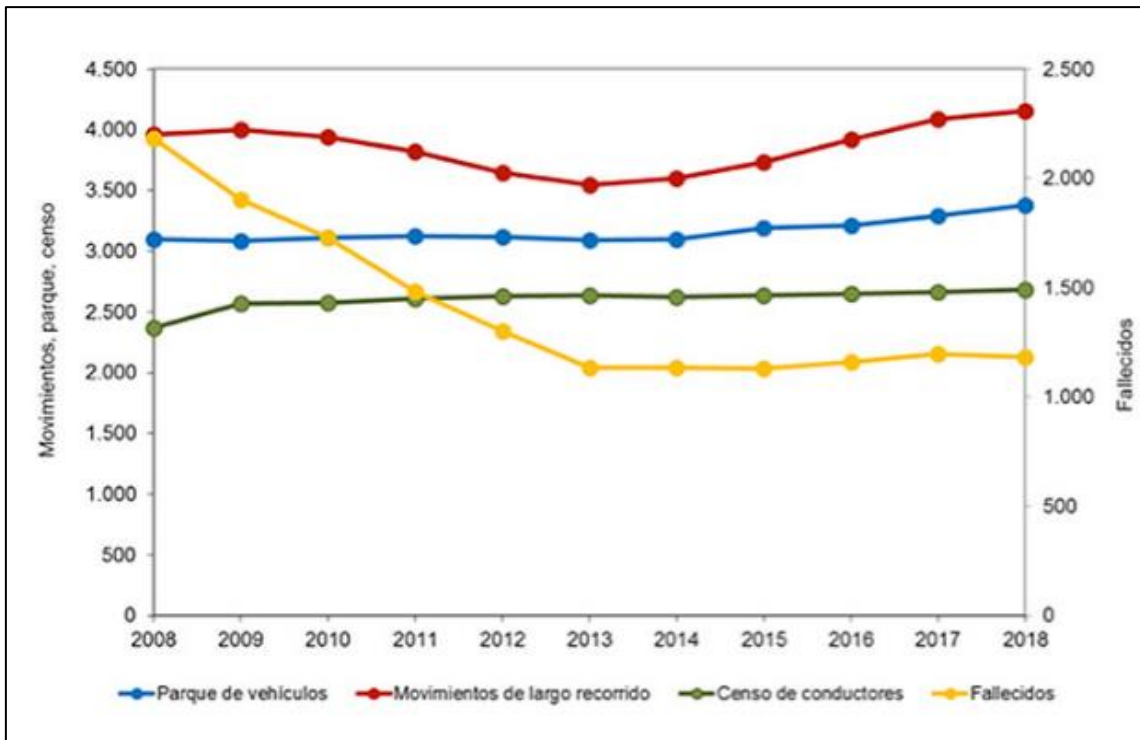


Figura 2. Estadísticas automovilísticas en España [ 1 ]

A pesar de todo esto, como se puede ver en la Figura 2, el número de fallecidos se ha reducido un 46% durante los 10 últimos años, y ello aumentado los desplazamientos (5% más), así como el censo de vehículos (9% más) y el censo de conductores (13% más).

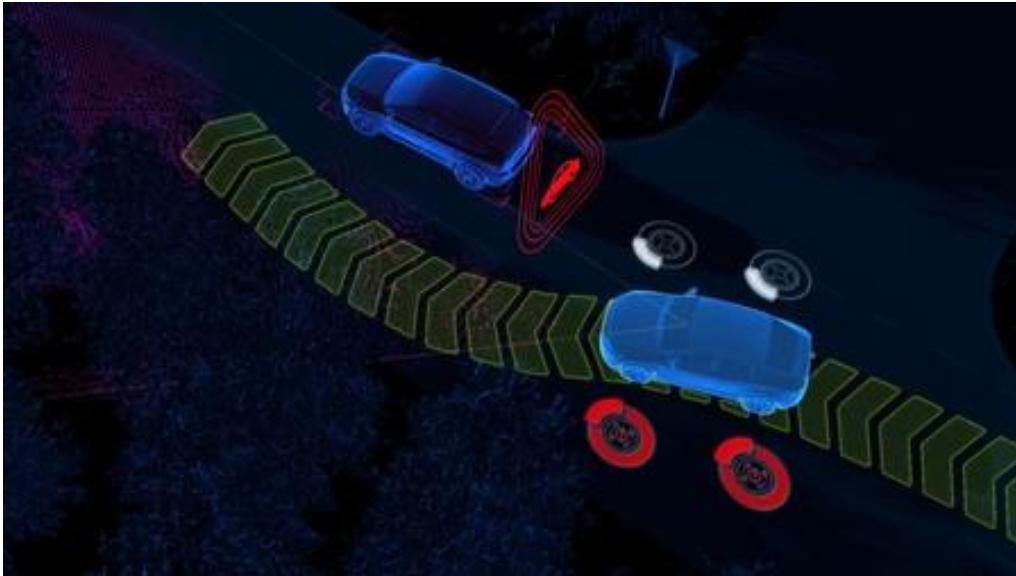
Con estos valores, podemos asegurar que las nuevas medidas de los vehículos (tanto activas como pasivas) son efectivas y tienen buenos resultados, pero no son suficientes, ya que el número de vidas que se han perdido en las carreteras es aún muy elevado.

A lo largo de año 2018, grandes empresas automovilistas, como Tesla, Mercedes-Benz o Audi, han anunciado que están desarrollando e incluso comercializando los primeros vehículos con cierto grado de autonomía (Figura 3). [ 2 ]



Figura 3. Vehículo autónomo Tesla Autopilot

Cuando estén desarrollados completamente, estos vehículos serán capaces de circular por sí solos utilizando una gran gama de medidas de seguridad y sensores para detectar a otros vehículos u obstáculos en su recorrido (Figura 4).



*Figura 4. Detección de obstáculos mediante radar [ 2 ]*

Con esta revolución en el mundo automovilístico, se espera que estos vehículos sean capaces de sustituir al ser humano al volante y reducir el número de fallecidos en accidentes de tráfico, así como proporcionar nuevas funcionalidades al ocupante.

Como es habitual, estos primeros prototipos tienen defectos que han de ser corregidos y perfeccionados. No obstante, no se prevé que el número de accidentes se reduzca a cero. [ 3 ]

Una de las características más importantes de estos vehículos es que van a intercambiar información con el resto de los coches cercanos, mediante un sistema de comunicación inter-vehicular. Esta información transmitida será simplemente de control, como puede ser, velocidad de circulación, estado del vehículo o notificaciones en caso de algún problema o emergencia.

Para lograr esta comunicación, el vehículo debe incorporar un ordenador que se encargue de gestionar dicha comunicación, transmitiendo toda la información que considere necesaria al resto de vehículos cercanos.

Como punto de partida de este trabajo, nos basaremos en el Trabajo de Fin de Máster de Javier Fernández Pastrana y los trabajos de Fin de Grado de Pilar Sánchez Martín y Alejandro Lobo González. [ 4 ][ 5 ][ 6 ]

Originalmente este trabajo final iba a realizarse de forma paralela con otro estudiante, Illán González Horna. Finalmente, Illán no continuó con el proyecto, por lo cual, se recogieron todos los avances realizados por su parte y se prosiguió con dicho trabajo, con el objetivo de terminar la implementación del sistema y realizar los tests y caracterizaciones necesarias, así como dejar registrado en una memoria todos los procesos, métodos y conclusiones que se han realizado. [ 7 ]

Javier F. Pastrana integró el estándar 802.11p en un *kernel* modificado de Linux Ubuntu utilizando unas tarjetas de WiFi de bajo coste. Este estándar está orientado a comunicaciones entre vehículos y recoge la regulación de las capas físicas y capas de acceso al medio de estas redes. [ 8 ]

Por otro lado, Pilar Sánchez y Alejandro Lobo consiguieron instalar un software específico de comunicaciones en los ordenadores de sobremesa del laboratorio de Comunicaciones Ópticas. Las unidades siguen el estándar de comunicaciones ETSI ITS G5 y utilizan el código OpenC2X para enviar información entre ellas. Dicho software es un programa de código abierto desarrollado por la Heinz NixDorf Institute (Universidad de Paderborn) que permite el intercambio de mensajes periódicos con información de control del vehículo (geolocalización GPS, velocidad del vehículo y velocidad del motor RPMs). [ 9 ]

Estas unidades pueden ser de dos tipos:

- *Road Side Unit* (RSU): Unidad alojada en las carreteras que proporciona un soporte de comunicación para los vehículos que circulan por ella. Son unidades fijas.
- *On Board Unit* (OBU): Unidad instalada en el vehículo encarga de intercambiar información con las RSU y otras OBU cercanas. También es la encargada de gestionar y controlar la comunicación (modificar la modulación utilizada, la tasa de transmisión, etc.).

## 1.2 Objetivos

El propósito de este proyecto ha sido la integración, el testado y la caracterización de un sistema de dos unidades de a bordo (OBU). Estas unidades se encargarán de transmitir información relevante de los vehículos de manera periódica y de forma espontánea en situaciones que pueden suponer un riesgo o generar un accidente como, por ejemplo, maniobras de frenado repentinas o cambios de velocidad bruscos, etc.

Para lograr el objetivo primario se definen en este Trabajo Fin de Máster una serie de objetivos secundarios:

- Integración del software en dos ordenadores de reducido tamaño. Esta integración incluye:
  - Instalación del sistema operativo Linux con *kernel* modificado.
  - Configuración de los drivers de la tarjeta de red Atheros ATH9K para dicho *kernel* siguiendo las instrucciones del trabajo de Javier Fernández Pastrana.
  - Instalación del software del proyecto de código abierto OpenC2X siguiendo las instrucciones del trabajo de Pilar Sánchez y Alejandro Lobo.
  - Configurar los nuevos dispositivos a utilizar: toma OBD, mando *gamepad* y geolocalizador GPS.
  - Automatizar todos los elementos de software que interactúan con los dispositivos externos de forma que se ejecuten al iniciar el sistema operativo.

- Realizar una caracterización completa del funcionamiento global del sistema, mediante un análisis individual de los diferentes componentes que afectan a la comunicación entre las dos unidades. Esto puede incluir simulaciones informáticas de elementos concretos, o caracterización experimental en laboratorio.
- Realizar una prueba de campo y analizar resultados de esta para determinar la fiabilidad y eficacia del sistema obtenido.

### **1.3 Medios**

A continuación, se describirán los elementos que han sido necesarios para llevar a cabo este proyecto:

- Dos ordenadores de pequeño tamaño.
- Dos tarjetas de red Atheros ATH9K.
- Dos geo-localizadores GPS Holux GPSlim 236.
- Dos pendrives Bluetooth.
- Ocho antenas de dos tipos diferentes (2 para cada ordenador).
- Analizador de redes HEWLETT PACKARD 8719D.
- Analizador de espectro ROHDE&SCHWARZ FSH6.
- Dos vehículos.
- Tarjeta USRP-210/HackRF One.

### **1.4 Estructura de la memoria**

En el capítulo 2 del documento “Estado del arte” se realizará un estudio en retrospectiva sobre las investigaciones y productos similares al del proyecto que se está siendo desarrollado en este trabajo de Fin de Máster.

En el capítulo 3 “Integración del software” se explicará el proceso de integración e instalación realizada para poner en funcionamiento los ordenadores utilizados y lograr la comunicación inter-vehicular.

En el capítulo 4 “Caracterización de los elementos del sistema” se describe todo el proceso de desarrollo realizado para conocer la información de funcionamiento del sistema en su conjunto, estudiando los elementos que forman parte de la comunicación de manera individual.

En el capítulo 5 “Demostración del sistema: caso de uso” se describe el experimento final realizado para comprobar el correcto funcionamiento del sistema y obtener la fiabilidad de la comunicación. En este apartado se detalla el código utilizado y los resultados obtenidos tras el procesado de los datos del experimento.

En el capítulo 6 “Conclusiones y líneas futuras” se recogen los resultados más importantes y algunas de las posibles mejoras que podrían ser implementadas en próximos Trabajos de Fin de Grado/Máster.

En el capítulo “Bibliografía” se encuentran las fuentes de información utilizadas durante este proyecto y documentación complementaria.

Por último, se encuentra el capítulo Anexo y los *datasheets* donde se recoge toda la información de los códigos que se han desarrollado y documentos consultados en el transcurso de este Trabajo de Fin de Máster.



## 2. Estado del arte

El concepto *Vehicle-to-anything* hace referencia a todo tipo de comunicaciones que utilizan los sistemas para intercambiar información entre un vehículo y otros elementos de un sistema inteligente de transporte. Estos sistemas, conocidos como **ITS**, están formados por otros vehículos (V2V), infraestructuras (V2I), peatones (V2P), etc. [ 10 ] [ 11 ]

Se espera que esta tecnología no solo logre un transporte más seguro y cómodo, sino que también permita desarrollar una gran variedad de nuevas aplicaciones en el entorno automovilístico, tales como, mejoras en la seguridad vial, mayor confort y entretenimiento a los pasajeros, nuevos servicios y mejorar el tráfico tanto en vías urbanas como interurbanas, reducir la contaminación o disminuir en gran medida la tasa de accidentes, entre otras. [ 12 ]

Sin embargo, la comunicación V2X se encuentra aún en una fase experimental donde solo se han realizado pequeños experimentos para comprobar aspectos muy puntuales. Por esta razón, la mayoría de las investigaciones existentes o que se están realizando en este ámbito, están orientadas a la caracterización teórica o a la creación de potentes simuladores en tiempo real de dicha comunicación, como un sistema de gestión de tráfico inteligente [ 13 ] o un simulador de propagación por múltiples caminos [ 14 ], entre otros. En cuanto a los experimentos realizados en campo real, podemos encontrar que están muy orientados a la seguridad de los peatones y de los ocupantes del vehículo [ 15 ], aun así, los investigadores siguen centrándose en investigar y preparar correctamente la implementación de este sistema. [ 16 ]

Antes de comercializar dichos productos o aplicaciones, se debe garantizar una fiabilidad y una madurez de la tecnología, a través de una serie de experimentos y condiciones muy rigurosas. En la actualidad, se están desarrollando aplicaciones V2X a corto plazo, a las que se pueden asignar una de las siguientes categorías:

- Aplicaciones de seguridad: relacionadas con la seguridad del pasajero, tales como advertencias de colisión, peligros en la calzada, avisos de velocidad, etc.
- Aplicaciones de eficiencia: relacionadas con la navegación del usuario, geolocalización, mejor ruta a seguir, información del tráfico existente y advertencias de congestión, etc.
- Aplicaciones de servicios de información: relacionadas con proporcionar información del automóvil al usuario para mejorar la experiencia de manejo, como puede ser las llamadas de emergencia automáticas, estacionamiento automático, recomendación de ruta, etc.

V2X tiene como objetivo que todas estas aplicaciones converjan y se logren los requisitos para una conducción autónoma completa y sistemas de tráfico inteligentes. [ 12 ]

Durante los últimos dos años (2017-2018), han surgido numerosos proyectos de investigación V2X que tienen como objetivo implementar estos sistemas de comunicación en coches reales, de una manera totalmente externa al desarrollo del vehículo por parte de la compañía suministradora, como es el caso de Vector, Mercedes Benz y Qualcomm.

Qualcomm está desarrollando una serie de sistemas orientados al vehículo totalmente autónomo. Estos sistemas se centran en optimizar el procesado de la información recibida de múltiples sensores y fuentes, así como disponer de una red 5G para realizar una comunicación unificada y un mapa de navegación 3D muy preciso (Figura 5), todo ello reunido en una pequeña y potente unidad de a bordo. El objetivo final de Qualcomm es lograr un vehículo 100% autónomo que disponga posición muy precisa, con un error menor a 1 metro de distancia. [ 17 ]

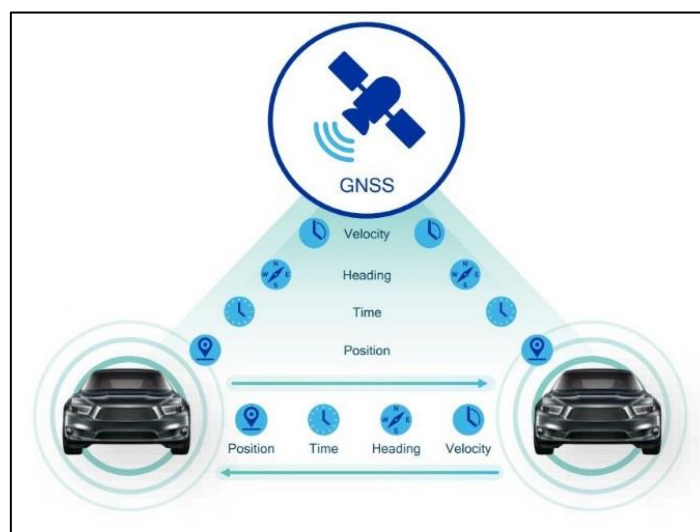


Figura 5. Parámetros GPS utilizados por una OBU [ 17 ]

Con toda esta red final, el sistema sería capaz de avisar al conductor de la existencia de un peligro inminente con un margen mayor de tiempo que si el usuario fuese el responsable de percibir dicha anomalía (Figura 6).

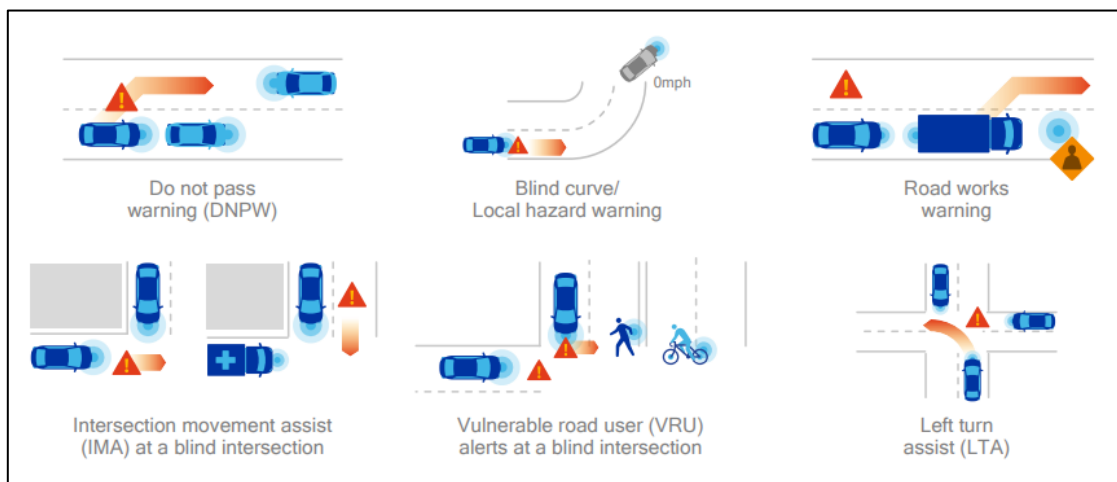


Figura 6. Detectores de situaciones de peligro [ 17 ]

Al igual que ha ido ocurriendo desde 1950, existe una tendencia a añadir un mayor número de sensores a los vehículos que nos proporcionen mayores capacidades de medida, predicción y autonomía. Los sensores que más recientemente se han añadido o que se desean añadir a las unidades de a bordo son:

- Radar: permite detecciones por proximidad en condiciones ambientales desfavorables, distancias largas y situaciones con baja luminosidad.
- Cámaras: que permitan identificar objetos y obstáculos.
- Sistemas de posicionamiento con posición muy precisa.
- Mapeado 3D HD.
- Comunicaciones Wireless V2X que permitan conexiones con visión no directa entre emisor y receptor (obstáculos en el canal) sin importar la dirección de envío (forma omnidireccional de 360°).

Muchos automóviles en el mercado ya cuentan con algunas de estas características, por ejemplo, el Sistema Presafe de Mercedes Benz, el cual combina la información de tres cámaras (largo, medio y corto alcance), junto con la información del sistema de control electrónico de estabilidad (ESP) y sistema de frenado ABS, para detectar situaciones críticas, mostrando al conductor una alerta visual y sonora del peligro, y actuando en caso de que no haya reacción por parte del piloto [ 18 ]. Este tipo de sistemas da al conductor un mayor tiempo de reacción, permitiendo una adaptación más rápida a la situación antes de encontrarse con el evento.

Otra compañía que ya está realizando sistemas V2X es Vector, la cual ha presentado una estación V2X, llamada *VN4610 network interface*, que ha sido diseñada para dar solución a las aplicaciones basadas en el estándar 802.11p y CAN-FD (actualización del protocolo CAN) [ 19 ]. Este dispositivo incorpora un receptor GNSS que proporciona la hora y la posición actual. [ 20 ]



*Figura 7. VN4610 network interface [ 20 ]*

Todos estos elementos presentan un gran número de ventajas en su incorporación a los automóviles actuales. Sin embargo, hay que tener en cuenta dos aspectos de gran importancia: asegurar la ciberseguridad de los usuarios y su privacidad. Actualmente, ya existen numerosas investigaciones y artículos que recogen y explican las ventajas y desventajas de cada una de las formas de implementar estas seguridades, como pueden ser el uso de una infraestructura de claves públicas para entornos V2X. [ 21 ]

En cuanto a la privacidad del usuario en los sistemas inalámbricos, requiere de una serie de estrategias de protección de la privacidad, las cuales aún se encuentran en un proceso

de desarrollo y estandarización. El foco de este tema se centra actualmente en utilizar alias temporales y otros algoritmos dentro los vehículos que sean capaces de confundir a los atacantes que desean robar o rastrear información de los vehículos inteligentes. [ 22 ]

El estándar de comunicación utilizado a lo largo de todo el proyecto es ETSI ITS G5, desarrollado por el Consorcio de Comunicación Vehículo a Vehículo (C2C-CC) en el año 2007. Este consorcio estaba originalmente formado por los grandes fabricantes alemanes del sector automovilístico (Volkswagen, Audi, BMW y Daimler) y tiene como objetivo cohesionar los proyectos referidos a comunicaciones V2X bajo los mismos estándares. [ 23 ]

Este estándar dispone de cuatro bandas de frecuencias en la frecuencia de 5 GHz para realizar las comunicaciones V2X. Estas cuatro bandas de frecuencia son: ITS-G5A, ITS-G5B, ITS-G5C y ITS-G5D.

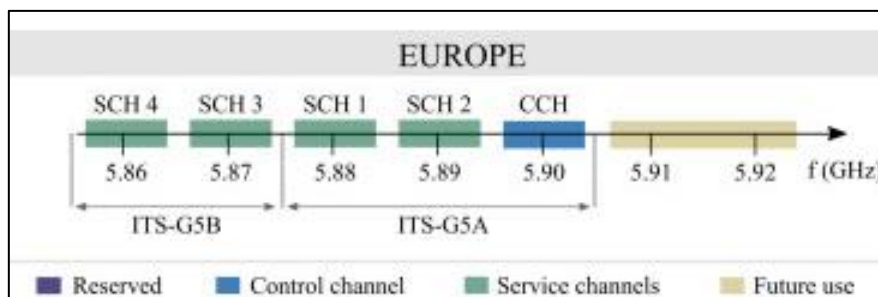


Figura 8. Canales comunicación V2X [ 20 ]

La banda que utilizaremos para realizar las comunicaciones V2X será la banda ITS-G5A, que tiene 30 MHz de anchura espectral y está dividida en tres canales de 10 MHz de espectro cada uno de ellos. El canal en torno a 5,9 GHz es el canal primario de control, utilizado para seguridad vial y los otros dos canales se utilizan para el mismo fin, pero con carácter secundario. En cuanto a las otras bandas representadas en la Figura 8, se dedican al resto de aplicaciones ITS. La banda ITS-G5D, marcada en amarillo, está reservada para las futuras aplicaciones ITS.

Un aspecto muy importante para el futuro de este tipo de comunicaciones es lograr una coordinación con otras tecnologías, como puede ser LTE o 5G. Estas tecnologías presentan retardos muy pequeños, alta fiabilidad y rendimiento alto en escenarios de alta movilidad. Estas características se ajustan en gran medida a los requisitos que necesitamos en las comunicaciones inter-vehiculares, lo que produce una incertidumbre de si es conveniente seguir utilizando el estándar de capa física actual o sustituir la forma de la comunicación por vía 5G. [ 24 ]

Finalmente, se espera que se produzca una hibridación de estas tecnologías, fortaleciendo las debilidades de cada una de ellas por separado. Los nodos de la red móvil LTE serán utilizados para la comunicación V2V mediante reenvío y el estándar 802.11p permitirá la comunicación directa entre vehículos (Figura 9).

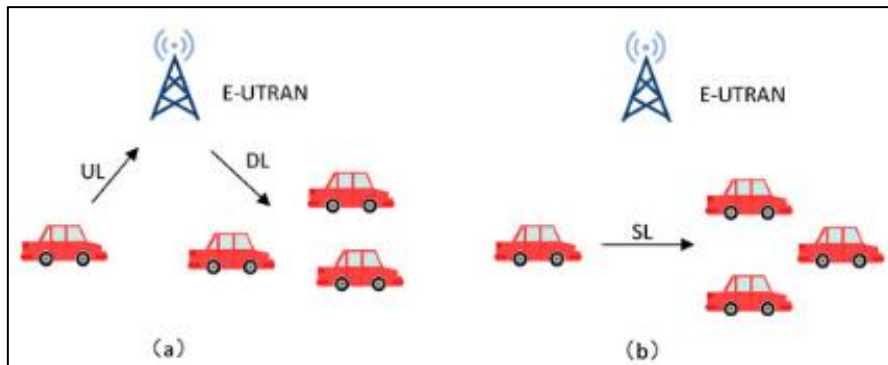


Figura 9. Escenarios de uso para cada tecnología [ 24 ]

Por otro lado, esta hibridación está asociada a la evolución de V2X. Esta transformación se realizará en tres fases y será un proceso progresivo que permita un gran avance en las funcionalidades ITS disponibles, así como una coexistencia entre 802.11p, LTE y 5G (Figura 10). [ 25 ]

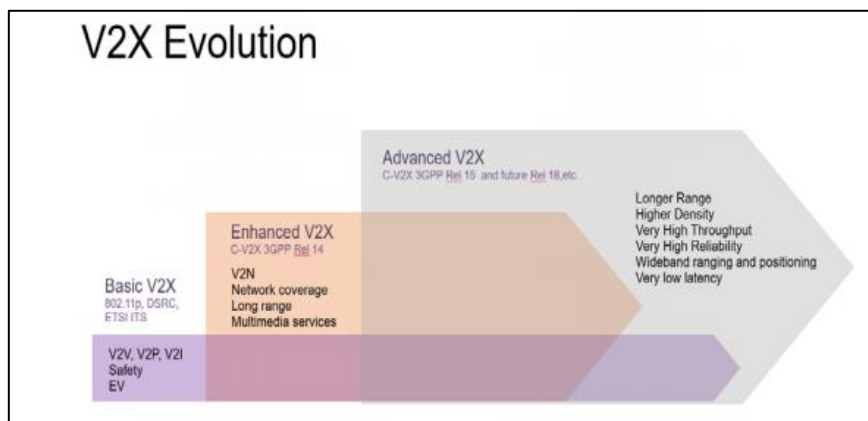


Figura 10. Etapas de evolución V2X [ 25 ]



# 3. Integración del software

En este apartado se explicará el proceso de instalación que ha sido necesario realizar para integrar correctamente el sistema operativo utilizado (Linux Ubuntu), así como el *kernel* modificado (versión 16.04.3) y los drivers necesarios para poder utilizar el protocolo 802.11p de la capa de acceso. Esta primera parte de la implementación fue desarrollada por Illán González Horna. [ 26 ] [ 4 ] [ 7 ]

## 3.1 Instalación del sistema operativo y drivers

Los ordenadores utilizados son unas unidades de abordo de placa única x86\_64 bits distribuida y fabricada por HSIPC. Cada unidad dispone de un procesador Intel Celeron N2920 a una frecuencia de 2,0 GHz, un disco duro Kingston MSATA de 60 Gb, una memoria RAM DDR3 de 2 GB, ampliable hasta 8 GB, y una tarjeta de red Atheros AR5BHB116 con conexión PCI-miniexpress. La placa base incorpora un chip de audio Realtek ALC6662. La parte trasera del ordenador dispone de dos entradas de video HDMI, una conexión ethernet GigaByte, la toma de alimentación de DC100-240V AC/50-60Hz, DC 12V y las conexiones de audio. La parte delantera dispone de 4 conexiones USB 2.0 y una entrada USB 3.0. [ 27 ]

Como se ha comentado en la sección “Introducción”, este proyecto es una continuación del trabajo realizado por otros compañeros en años anteriores. El proyecto heredado se encuentra instalado en ordenadores de sobremesa del laboratorio de investigación y esta configuración nos impide desplazar el software al vehículo para realizar el testado en un entorno realista.

En consecuencia, fue necesario realizar una instalación completa en dos pequeños ordenadores que permitieran una fácil instalación en un vehículo (Figura 11). Esta instalación completa consistió en instalar el sistema operativo, modificar el *kernel*, configurar las interfaces para el modo de funcionamiento OCB (*Outside the Context of a BSS*) e instalar el programa V2X.



Figura 11. Ordenadores utilizados

### 3.1.1 Sistema operativo

En primer lugar, se procedió a la instalación del sistema operativo. Dados los problemas que surgieron durante la instalación de los sistemas operativos en este tipo de ordenadores explicamos los pasos que se siguieron:

1. Descargar el S.O. Ubuntu 16.04.3 LTS Desktop de 64 bits para el ordenador etiquetado como B y versión de 32 bits para el ordenador marcado como A. La versión que se debe utilizar para cada ordenador esta asignada por el procesador instalado en cada uno de ellos.
2. Crear un USB ejecutable (*Bootable*) con *Universal USB Installer*.
3. Insertar USB en el miniPC y presionar F11 para entrar en el menú de *boot*
4. Ir a la BIOS y desactivar SECURE BOOT
5. Eliminar las particiones ya existentes en Windows10 del disco duro
6. Elegir opción *Legacy* en *boot* menú (la que no tiene UEFI<sup>1</sup>)
7. En el menú de particiones seleccionar → Nueva tabla de particiones

```
SWAP --> + : 1024 MB , Primaria , Al principio de este espacio , Área de intercambio
RAÍZ --> + : 25000 MB , Lógica , Al principio de este espacio , sistema de ficheros ext4 transaccional , /
HOME --> + : 33000 MB , Lógica , Al principio de este espacio , sistema de ficheros ext4 transaccional , /home
```

8. Instalar.

---

<sup>1</sup> Aunque se intentó realizar la instalación en el modo más moderno UEFI, no fue posible, debido a que los desarrolladores de Linux decidieron no unirse al consorcio que desarrollo este estándar, dando lugar a veces a problemas de instalación en entornos Linux.



Para continuar la instalación, se cambió el *kernel*<sup>2</sup>, se añadió la configuración de **OCB** y se realizó la instalación del programa **V2X**. Primero, es necesario instalar las dependencias mediante una consola de comandos (cmd).

```
sudo apt-get install build-essential
sudo apt-get install libncurses5
sudo apt-get install libncurses5-dev
sudo apt-get install libnl-utils
sudo apt-get install libnl-3-200
sudo apt-get install libnl-3-dev
sudo apt-get install python-m2crypto
sudo apt-get install libgcrypt20
sudo apt-get install libgcrypt20-dev
sudo apt-get install libgpg-error0
sudo apt-get install libgpg-error-dev
```

Para realizar el cambio de *kernel* hay que descargar dicho *kernel* linux-4.2.8.tar.xz de la página de distribuciones [ 28 ]. También hemos descargado los drivers de GitHub de Javier Fernández Pastrana. [ 29 ]

Una vez tenemos todo, descomprimos ambos (tar -xvf archivo.tar.xz) y copiamos los drivers en /linux-4.2.8, siendo recomendable seguir los pasos del README de linux-4.2.8 (teniendo cuidado con el menuconfig, en el que se han de cambiar algunas cosas: para más información acudir a la referencia [ 4 ], pg. 19, 3.7):

```
sudo cp -R linux-4.2.8 /usr/src/
mkdir /home/gco2/Descargas/build
mkdir /home/gco2/Descargas/build/kernel
cd /usr/src/linux-4.2.8/
make mrproper
```

Hay que expandir la pantalla del terminal al máximo o se producirá un error diciendo que la pantalla es demasiado pequeña.

```
make O=/home/gco2/Descargas/build/kernel menuconfig
```

En el menú, entrar y modificar según el TFM de Javier Fernández Pastrana [ 4 ]:

- Device drivers -> Network device support -> Wireless LAN -> Atheros Wireless Cards (si no hay una M, ponerla)
- Networking support -> Wireless
- Networking support -> Wireless -> Select mac80211 debugging

Guardar como .config

---

<sup>2</sup> Si usamos kernels superiores que 4.2.8 puede que los drivers de Atheros estén ya por defecto (en nuestro caso, 4.10.28, ya están)

A continuación:

```
make O=/home/gco2/Descargas/build/kernel
sudo make O=/home/gco2/Descargas/build/kernel modules_install
```

Cambiamos la configuración GRUB (/etc/default/grub)

```
GRUB_DEFAULT=6 (entry number) OR GRUB_DEFAULT=gnulinux-4.2.8-
advanced-0c..... (entrar en /boot/grub/grub.cfg y buscar el nombre
correspondiente)

GRUB_TIMEOUT=10 (o comentar para que empiece directamente)

Ordenador A → GRUB_CMDLINE_LINUX_DEFAULT="acpi=force"
Ordenador B → GRUB_CMDLINE_LINUX_DEFAULT="acpi=off"

GRUB_CMDLINE_LINUX=""
GRUB_DISABLE_SUBMENU=y % para que no haya varios submenús
```

La siguiente configuración por realizar es la **tarjeta de red**, en nuestro caso, una tarjeta Atheros ATH9K:

```
IW
cd Descargas
git clone https://github.com/CTU-IIG/802.11p-iw.git
cd 802.11p-iw
git checkout its-g5_v3 % comprobación
make clean
make
```

Si existen errores, hay que probar con una de estas posibles soluciones:

- Con Netlink/Netlink.h: `export C_INCLUDE_PATH=./usr/include/libnl3` y hay que comprobar la ruta con `apt-file search /netlink/netlink.h`
- Con nl-genl-3: `sudo apt-get install libnl-genl-3-dev`
- Otro: `export PKG_CONFIG_PATH=/usr/lib/pkgconfig`

```
sudo PREFIX=/ make install OR sudo cp /sbin/iw /sbin/iw_old AND sudo
cp iw /sbin/
/sbin/iw | grep -i ocb
```

Al ejecutar este comando debemos obtener:

```
dev <devname> ocb leave
dev <devname> ocb join <freq in MHz> <5MHZ|10MHZ> [fixed-freq]
```

El siguiente paso es configurar WIRELESS-REGDB y el dominio regulatorio CRDA:

- **WIRELESS-REGDB:**

```
git clone https://github.com/CTU-IIG/802.11p-wireless-regdb.git
cd 802.11p-wireless-regdb
git checkout its-g5_v1
nano db.txt (añadir tal y como se explica en el TFM de J. Fernández [ 4 ], pg.23):
country AA:
(2402 - 2482 @ 40), (30)
(5170 - 5250 @ 80), (30)
(5250 - 5330 @ 80), (30)
(5490 - 5730 @ 80), (30)
(5735 - 5835 @ 80), (30)
# For ITS-G5 evaluation
(5840 - 5935 @ 10), (30)
make
sudo PREFIX=/ make install
```

- **CRDA**

```
git clone https://github.com/CTU-IIG/802.11p-crda.git
cd 802.11p-crda
git checkout its-g5_v1
cp /lib/crda/pubkeys/username.key.pub.pem pubkeys/ DONDE
username=gco OR =gco2 (para el ordenador B)
cp /lib/crda/pubkeys/linville... pubkeys/
cp /lib/crda/regulatory.bin .
make
Si error: modificar make (en primera linea, cambiar /usr/lib/libnl.. por /lib/libnl..)
sudo PREFIX=/ REG_BIN=/lib/crda/regulatory.bin make install
sudo /sbin/regdbdump ../802.11p-wireless-regdb/regulatory.bin | grep
-i ocb % Al ejecutar este comando debemos obtener:
country 00: invalid
(5850.000 - 5925.000 @ 20.000), (20.00), NO-CCK, OCB-ONLY
```

Se ha de tener en cuenta que si decidimos cambiar el fichero `db.txt` para incluir nuevas frecuencias o nuevos países como hicimos en el paso anterior, será necesario hacer una compilación total del dominio regulatorio CRDA, creando de nuevo las claves públicas, tal y como se explica en [ 4 ].

Una vez ya hemos realizado los pasos anteriores, procedemos a configurar la interfaz OCB, para ello, creamos un fichero llamado `ocb.sh` disponible en `/Descargas/scripts/`. (Extraído del TFM de J. F. Pastrana, página 79) [ 4 ].

Creación del fichero: `nano ocb.sh`

Contenido del fichero:

```
#!/bin/bash

WLAN=wlp2s0
OCB_WLAN=ocb0
DIR_IP=192.168.1.2
MASK=255.255.255.0

#Setting down the wireless interface
ip link set $WLAN down
#Adding a new interface in OCB mode
iw dev $WLAN interface add $OCB_WLAN type ocb
#Setting OCB mode
iw dev $OCB_WLAN set type ocb
#Setting down OCB interface
ip link set $OCB_WLAN down
#Setting wireless regulatory domain to AA
iw reg set AA
#Has it changed?
iw reg get
#Rising OCB interface
ip link set $OCB_WLAN up
#Joining interface to Control Channel and 10MHz of bandwidth
#iw dev $OCB_WLAN ocb join 5990 10MHZ
#Leaving the Control Channel
#iw dev $OCB_WLAN ocb leave
#Joining interface to Service Control Channel 1
iw dev $OCB_WLAN ocb join 5880 10MHZ
#Was it successful?
iw dev | iwconfig

#Setting up IP address, netmask and default gateway
ifconfig $OCB_WLAN $DIR_IP netmask $MASK
```

Figura 12. Código script Ocb.sh

### 3.1.2 OpenC2X

Una vez ya tenemos todo el sistema operativo correctamente instalado, procedemos a descargar e instalar el código del programa V2X. Descargamos el programa y seguimos los pasos del README.txt para realizar la instalación: [ 30 ]

```
sudo apt-get install libzmq3-dev libboost-all-dev protobuf-compiler
libprotobuf-dev \
libgps-dev gpsd gpsd-clients libnl-3-200 libnl-3-dev libnl-genl-3-
200 libnl-genl-3-dev sqlite3 \
libsqlite3-dev tmux asnlc build-essential cmake doxygen
cd /home/gco/Descargas/OpenC2X/common/asnl/
./generate.sh
cd /home/gco/Descargas/OpenC2X/

mkdir build
cd build
cmake ..
make all
nano /home/gco/Descargas/OpenC2X/common/config/config.xml
<ethernetDevice>ocb0</ethernetDevice>
```

Si hay error por las rutas:

```
nano /home/gco/Descargas/OpenC2X/scripts/runOpenC2X.sh
OPENC2X=/home/gco2/Descargas/OpenC2X
BUILD_DIR=$OPENC2X/build (antes venía build/)
```

Para lanzar el programa OpenC2X:

```
sudo /home/gco2/Descargas/OpenC2X/scripts/runOpenC2X.sh
```

Para detener el programa:

```
sudo /home/gco2/Descargas/OpenC2X/scripts/stopOpenC2X.sh
```

Para comprobar que los datos han sido recibidos correctamente entramos con DB Browser for SQLite en `OpenC2X/build/lbm/db/lbm-1.db`

Esta aplicación nos permite lanzar mensajes desde un navegador web mediante html. Para ello, hay que ejecutar el fichero: `OpenC2X/webSite/index.html`

Por defecto, OpenC2X coge datos aleatorios en vez de reales. Para coger reales en el GPS y OBD, modificamos los siguientes ficheros:

```
nano /home/gco/Descargas/OpenC2X/gps/config/config.xml →
  <SimulateData>>false</SimulateData>

nano /home/gco/Descargas/OpenC2X/obd2/config/config.xml →
  <SimulateData>>false</SimulateData>
  <Device>/dev/rfcomm1</Device>
  <Frequency>500</Frequency>
```

Para que haya una comunicación entre ambos ordenadores, los identificadores de cada uno de ellos, conocidos como *stationIds*, deben ser diferentes para que no se confundan entre ellos:

```
nano /home/gco/Descargas/OpenC2X/common/config/config.xml
  <ethernetDevice>ocb0</ethernetDevice>
  <stationId>111111111</stationId>
  # <stationId>222222222</stationId> para el ordenador B
```

Para lograr una mejora en el funcionamiento del programa, se han modificado los siguientes ficheros:

- Para que se puedan recuperar las RPM junto con la velocidad (aunque no lo vamos a utilizar) [ 5 ]:

```
nano /home/gco/Descargas/OpenC2X/obd2/src/SerialPort.cpp
```

Se han modificado los parámetros del principio (`terminalAttributes`)

79	-	<code>terminalAttributes.c_cc[VMIN] = 0;</code>
80	-	<code>terminalAttributes.c_cc[VTIME] = 5;</code>
85	+	<code>terminalAttributes.c_cc[VMIN] = 23;</code>
86	+	<code>terminalAttributes.c_cc[VTIME] = 1;</code>

Figura 13. Código `terminalAttributes` modificado

- Para que aparezca la hora en el log del GPS y del OBD2:

```
nano /home/gco/Descargas/OpenC2X/obd2/src/Obd2Service.cpp
```

Se ha añadido una biblioteca de funciones al principio y se ha modificado la función `Obd2Service::sendToServices`

```

34 + // NUEVAS
35 + #include <time.h>
36 +

```

Figura 14. Nueva biblioteca añadida en `Obd2Service.cpp`

```

125 - mLogger->logStats(to_string(Obd2.speed())); // In csv, we log speed in m/sec
129 + //std::time_t temp = Obd2.time()/100000000;
130 + //string ObdAndTime = to_string(Obd2.speed()) + "\t" + ctime(&temp); // new line
131 + string ObdAndTime = to_string(Obd2.speed()) + "\t" + to_string(Obd2.time());
132 + mLogger->logStats(ObdAndTime); // In csv, we log speed in cm/sec and time in ns since epoch
133 + //mLogger->logStats(to_string(Obd2.speed())); // In csv, we log speed in m/sec

```

Figura 15. Modificación en `Obd2Service::SendToServices`

En GPS se ha modificado:

```

nano /home/gco/Descargas/OpenC2X/gps/src/GpsService.cpp

```

Se añadido una biblioteca de funciones al principio y se ha modificado la función `GpsService::sendToServices`.

```

39 + // NUEVAS
40 + #include <time.h>

```

Figura 16. Nueva biblioteca añadida en `GpsService.cpp`

```

262 - string csvPosition = to_string(gps.latitude()) + "\t" + to_string(gps.longitude()) + "\t" + to_string(gps.altitude());
264 + //string csvPosition = to_string(gps.latitude()) + "\t"
265 + + to_string(gps.longitude()) + "\t" + to_string(gps.altitude());
266 + //std::time_t temp = gps.time()/100000000;
267 + //string csvPosition = to_string(gps.latitude()) + "\t"
268 + + to_string(gps.longitude()) + "\t" + to_string(gps.altitude()) + "\t" + ctime(&temp);
269 + string csvPosition = to_string(gps.latitude()) + "\t"
270 + + to_string(gps.longitude()) + "\t" + to_string(gps.altitude()) + "\t" + to_string(gps.time());
263 271 mLogger->logStats(csvPosition);
272 +
273 + //std::ofstream myfile3;
274 + //myfile3.open ("/home/gco/Descargas/gps.txt",std::ios::app);
275 + //myfile3 << to_string(gps.latitude()) << " " << to_string(gps.longitude()) << "\n";
276 + //myfile3.close();

```

Figura 17. Modificación en `GpsService::SendToServices`

- Como parecía que las RPM podían dar lugar a problemas de frecuencia de recuperación de datos, se dejaron de recoger. Para ello:

```

nano /home/gco/Descargas/OpenC2X/obd2/src/Obd2Service.cpp

```

Se comentaron las líneas `readRpm()` y `Obd2.set_rpm(rpm)`

```

48 - mLogger->logStats("speed (m/sec)");
49 -
51 + //mLogger->logStats("speed (m/sec)");
52 + mLogger->logStats("speed (cm/sec)");
53 +

```

Figura 18. Corrección en las unidades del OBD

```

68 - int rpm = serial->readRpm();
72 + // int rpm = serial->readRpm();

```

Figura 19. Modificación en el código readRpm()

```

75 - if (rpm != -1) {
76 -     obd2.set_rpm(rpm);
77 - }
79 + // if (rpm != -1) {
80 + //     obd2.set_rpm(rpm);
81 + // }

```

Figura 20. Modificación en el código OBD2.setrpm()

- Para que reconozca la velocidad (y, en su caso, las RPM) correctamente:

La respuesta del OBD2 normalmente tiene el siguiente formato: ">010D\r41 0D XX", pero en el programa no se espera el ">".

```
nano /home/gco/Descargas/OpenC2X/obd2/src/Serialport.cpp
```

Se han modificado las funciones SerialPort::readSpeed()

```

138 +
139 +
140 +     std::ofstream myfile2;
141 +     myfile2.open ("/home/gco/Descargas/Logs/e2.txt",std::ios::app);
142 +     myfile2 << response << "\n";
143 +     myfile2.close();
144 +
127 145
128 146     //correct response format: "010D\r41 0D XX \r\r" with XX being the speed in hex
129 - if (response.compare(0,4,"010D") == 0 && response.compare(5,5,"41 0D") == 0) {
147 + //if (response.compare(0,4,"010D") == 0 && response.compare(5,5,"41 0D") == 0) {
148 + //if (response.find("010D") != std::string::npos && response.find("41 0D") != std::string::npos) {
149 + if (response.find("41 0D") != std::string::npos) {
150 +     int posicion = response.find("41 0D"); // position of first character
151 +     //convert hex to decimal
152 - stream << hex << response.substr(11,2);
153 + //stream << hex << response.substr(11,2);
154 + stream << hex << response.substr(posicion+6,2);
155 + speedKmh;
156 + speedMs = (double)speedKmh / 3.6; //convert km/h to m/s
157 +
158 + /*
159 +     std::ofstream myfile1;
160 +     myfile1.open ("/home/gco/Descargas/Logs/vel.txt",std::ios::app);
161 +     myfile1 << speedKmh << "\n";
162 +     myfile1.close();
163 + */
164 +
165 + //print
166 + cout << "Speed in hex: " << response.substr(11,2) << endl;
167 + cout << "Speed in km/h: " << speedKmh << endl;
168 + cout << "Speed in m/s: " << speedMs << endl;
169 -
170 + }
171 + else { //incorrect (eg. "010D\rSEARCHING...\r\r")
172 +     cout << "Invalid speed. Plug in OBD2 and start engine." << endl;
173 +     cout << "Response: " << response << endl;

```

Figura 21. Modificación en SerialPort::readSpeed()

- Para no recibir nuestros propios mensajes en los logs:

```
nano /home/gco/Descargas/OpenC2X/dcc/config/config.xml →
<ignoreOwnMessages>true</ignoreOwnMessages>
```

En caso de que realicemos cualquier modificación en el código fuente de OpenC2X, deberemos compilar de nuevo el proyecto con los siguientes comandos:

```
cd Descargas/OpenC2X/build
sudo make all
```

### 3.1.3 GPS bluetooth

A continuación, vamos a configurar el GPS mediante bluetooth. En primer lugar, tendremos que realizar el emparejamiento entre el pc y el GPS correspondiente a cada ordenador.

Para comprobar que el USB Bluetooth ha sido detectado:

```
hcitool dev
```

Este comando muestra la lista de todos los dispositivos que ya están sincronizados con el ordenador.

Cuando queremos enlazar un nuevo dispositivo, hay que realizar una búsqueda de los dispositivos Bluetooth cercanos:

```
hcitool scan
```

El resultado de este comando será una lista de los dispositivos bluetooth detectados en los alrededores, indicando la dirección MAC del dispositivo y su alias.

La primera vez hay que emparejar el ordenador con el GPS y habrá que hacerlo una sola vez. El procedimiento de emparejamiento se explica a continuación.

**sudo apt-get install bluetoothctl OR bluez OR bluez-tools** (en caso de que bluetooth-agent esté obsoleto).

```
bluetoothctl
power on
scan on
agent on
pair 00:12:6F:10:53:34
Pairing code: 1234
trust 00:12:6F:10:53:34
quit
```

La realización de la conexión bluetooth entre el GPS y el ordenador es una acción necesaria pero no suficiente para establecer la conexión. Para ello, hay que escribir la siguiente sentencia en la consola *cmd*. Esta sentencia establece la conexión entre dispositivo GPS y el puerto serie 0 asociado al proyecto C2X:

```
sudo /usr/bin/rfcomm connect 0 00:0B:0D:84:CD:3C
```

Al ejecutar la línea anterior, conectaremos con el GPS, pero dado que el C2X está activo y conectado a ese puerto, nos impedirá lanzar los comandos necesarios para empezar a recibir la información del dispositivo GPS. Por esta razón y para evitar problemas de



conexión, eliminaremos los procesos activos mediante la ejecución de las siguientes líneas de código:

```
sudo killall gpsd
sudo systemctl stop gpsd.service
sudo systemctl stop gpsd.socket
```

Una vez desactivados los procesos, ejecutamos el comando *gpsd* para recibir información del GPS:

```
sudo gpsd -b -N -D 4 /dev/rfcomm0
```

Para comprobar si el GPS está recibiendo datos y estos son correctos, podemos usar dos aplicaciones:

Interfaz gráfica → xGPS (GUI)

Consola → cGPS (CMD)

Si hubiese este error:

```
kernel bluetooth tiocgserial is not supported modemmanager could not
grab port (tty/rfcomm1): cannot add port tty/rfcomm1 unhandled serial
type modemmanager couldn't create modem for device at
/sys/devices/virtual/tty/rfcomm1: failed to find primary at port
GPSd:ERROR: can't bind to IPv4 port GPSd, Address already in use
resulta que systemd lo abre así y hay que detenerlo.
```

Podemos probar las siguientes soluciones:

```
sudo apt-get remove modemmanager -> lo ha solucionado porque el
modemmanager interfería.
sudo systemctl stop gpsd.service ;
sudo systemctl stop gpsd.socket ;
sudo killall gpsd ;
sudo rm /var/run/gpsd.sock
systemctl stop gpsd
systemctl disable gpsd
```

### 3.1.4 OBD2 bluetooth

Los coches disponen de una interfaz OBD que nos permite obtener información del vehículo de una manera sencilla. El primer sistema de diagnóstico de a bordo fue creado por Volkswagen y Bosch para sistema electrónico de inyección de combustible en 1968. [ 31 ], aunque la interfaz que es utilizada actualmente fue introducida por primera vez en la normativa anticontaminación del estado de California en 1996. [ 32 ]

Durante el TFM, utilizaremos dos dispositivos OBD (Figura 22 (a)), conectados a la toma OBD de cada vehículo (Figura 22 (b)), que nos permitirán enviar un identificador (PID de velocidad) y el vehículo nos responderá la petición con la información solicitada.



(a) (b)

Figura 22.(a) Dispositivo OBD (b) Toma OBD

Para conectarse al OBD y empezar a enviar datos, necesitaremos previamente configurar el Bluetooth, como en el caso del GPS. Una vez establecida la comunicación bluetooth, hay que realizar una conexión a `rfcomm`:

```
sudo /usr/bin/rfcomm connect 1 00:12:6F:10:53:34
```

En otra terminal:

```
cd /dev
sudo minicom -D /dev/rfcomm1
```

Una vez dentro del programa, debemos configurar la comunicación: `ctrl+a` para desplegar las opciones → Tecla `O` → Configuración de la puerta serial → `/dev/rfcomm1` y 38400 baudios

Para saber si el OBD está bien conectado y recibimos la información correctamente, introducimos las siguientes sentencias en `minicom`. [ 33 ]

```
Petición(P):   ATI
Respuesta(R):  OK
P:             AT Z
R:             OK
P:             AT SP 0
R:             OK
P:             AT DP
R:             AUTO
```

Tras este último comando, tenemos que enviar el PID que deseemos conocer: [ 34 ]

- PID de la Velocidad → `010D` → La respuesta es `41 0D XX` → `XX` velocidad en hexadecimal
- PID de las RPM → `010C` → Respuesta: `41 0C XX` → `XX` rpm en hexadecimal y luego dividir entre 4 [ 35 ].

Se realizaron pruebas en los coches utilizados para saber cuál era la frecuencia máxima que admitían de petición de datos. Ambos coches no eran capaces de soportar la misma frecuencia, de tal forma que en uno se dejó un periodo de muestreo de 100 ms, y en el

otro de 200 ms. La modificación se puede realizar directamente en el fichero /OpenC2X/obd2/config/config.xml.

```
<obd2>
  <SimulateData>false</SimulateData>
  <Device>/dev/rfcomm1</Device>
  <Frequency>100</Frequency> <!-- in milliseconds -->
</obd2>
```

Por otro lado, se modificó el código en SerialPort.cpp para que los comandos AT de inicialización tuvieran tiempo suficiente para ser procesados y enviados tanto por el coche como por el OBD (1 segundo entre mensajes de control y configuración, y 4 segundos en la petición del primer valor de la velocidad).

```
void SerialPort::init() {
    cout << "Initializing OBD2" << endl;

    write(mFileDescriptor, "AT Z\r", 5);           //reset all (AT
Z + return)
    sleep(1);
    write(mFileDescriptor, "AT SP0\r", 7);        //auto search for
suitable protocol
    sleep(1);
    write(mFileDescriptor, "ATDP\r", 5);         //verify protocol
    sleep(1);
    write(mFileDescriptor, "010D\r", 5);         //Necesitamos 3
segundos en la primera solicitud para que los datos se reciban
correctamente
    sleep(4);
}
```

### 3.1.5 Gamepad bluetooth

En los sistemas de experimentación en campo suele ser muy importante ser capaces de registrar dentro de los dataloggers eventos que los testadores observan en el momento. Para ello, se necesita que el testador sea capaz de interactuar con el datalogger en tiempo real. En nuestro sistema el dispositivo que utilizaremos para tomar muestras temporales es el Gamepad, mostrado en la Figura 23.



Figura 23. Dispositivo Gamepad

Este dispositivo es un mando de videojuego, generalmente denominados gamepads, de reducido tamaño, con batería incorporada y que se conecta mediante bluetooth.

Al igual que en el caso del OBD, lo primero que hay que realizar es la conexión bluetooth entre dispositivos.

```
bluetoothctl
power on
scan on
agent on
pair 2C:FF:8B:01:00:07
trust 2C:FF:8B:01:00:07
connect 2C:FF:8B:01:00:07
quit
```

Solo hay que hacerlo una vez porque después se conecta solo. Hay que esperar al menos 1 minuto después de encender el miniordenador y que se ejecuten los programas automáticos antes de encender el gamepad. Además, es necesario pulsar algún botón del gamepad de vez de en cuando para que no se apague el dispositivo y la conexión siga abierta.

Para realizar la configuración del gamepad introduciremos las sentencias:

```
sudo apt-get install evtest
sudo evtest
```

El comando `evtest` nos permite ver qué puerto (*input device*) es el que tiene asignado el gamepad (el número puede cambiar según los periféricos que haya conectados, p. ej. teclado, ratón, ...). Este programa también deja ver qué códigos devuelven las pulsaciones de la tecla, que luego se recogerán con el código C++ de GitHub.

El código del gamepad se encuentra en GitHub y la primera vez hay que escribir este comando para descargarlo:

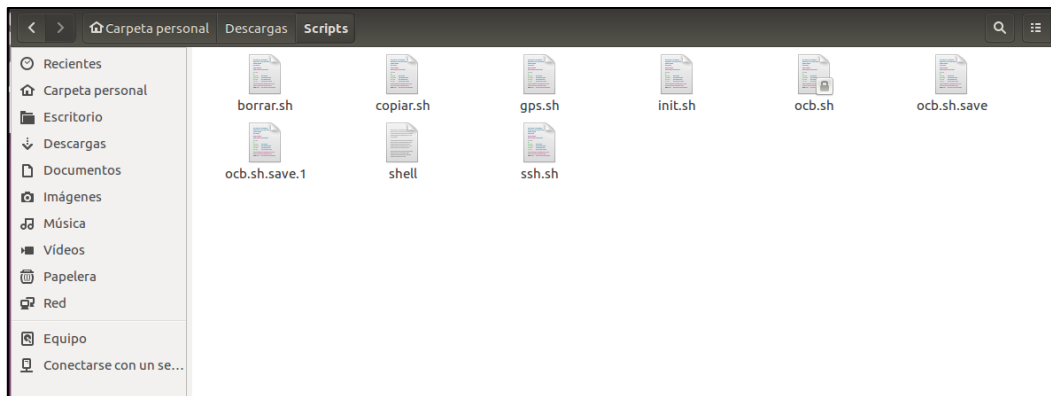
```
git clone https://github.com/illan95/GamePad.git
```

El gamepad está configurado para que recoja marcas temporales durante el experimento final al pulsar el botón @. Estas marcas nos permitirán eliminar los errores introducidos por los sistemas GPS y OBD durante la prueba. Por otro lado, se ha configurado otro de los botones (con forma de X) para enviar mensajes DEMN al otro ordenador y el botón Δ para apagar el ordenador cuando la prueba haya finalizado. El cuarto botón restante (IOS) se ha dejado libre para poder ser pulsado regularmente y mantener activa la conexión Gamepad-Ordenador.

### 3.2 Automatizar el arranque

La ejecución de prueba en nuestro dispositivo requiere el lanzamiento de múltiples programas con diversas opciones, siempre en el mismo orden. Para la ejecución de estas órdenes repetitivas se decidió realizar una serie de scripts que nos permitiera automatizar

y ejecutar automáticamente el sistema al completo. Los scripts creados se pueden encontrar en la carpeta `/Descargas/Scripts` tal y como se puede ver en la Figura 24.



*Figura 24. Scripts desarrollados*

Además, durante las pruebas de campo, no se dispondrá ni de monitor, teclado o ratón que se pueda utilizar para ejecutar la configuración necesaria o los programas a utilizar. Es por esto por lo que se ha desarrollado un script que se autoejecuta en el arranque del sistema. El nombre de este script es `init.sh`.

El script cuenta con numerosos `sleep 10` que proporcionan el tiempo necesario que todos los procesos se terminen de ejecutar correctamente y sin generar ningún error, antes de empezar a trabajar sobre ellos.

En el script `init.sh` inicial, se han añadido las líneas de la Figura 25 para coordinar las horas de los dos ordenadores cuando se van a iniciar las pruebas. Este proceso es especialmente importante en el ordenador B, ya que no dispone de pila en la memoria BIOS, por lo que no es capaz de mantener la hora tras el apagado. Además, también es importante para mayor facilidad a la hora de comparar el flujo de mensajes entre los dos ordenadores.

El método consiste en coger la información enviada mediante el protocolo NMEA del GPS, en concreto, una vez iniciado el flujo continuo de información sobre la posición y hora del automóvil con los comandos anteriormente explicados, se utiliza la primera respuesta recibida, y de ella se obtiene y procesa la hora del satélite. Posteriormente, se utiliza esta información para ajustar la hora a cada uno de los ordenadores.

Para que este método funcione correctamente, es recomendable encender cada GPS durante unos minutos antes que los ordenadores. De esta forma, los GPS son capaces de localizar los satélites y actualizar el almanaque de tal forma que la primera información recogida por los miniordenadores procedentes de los GPS ya contiene tanto la posición como la hora. [ 36 ]. Es importante señalar que el programa da por supuesto que esta información se está recibiendo correctamente, esto es, si cuando el ordenador ha encendido y ha solicitado datos al GPS este todavía no ha actualizado el almanaque, entonces la información que recoja el programa será incorrecta.

```

##### Cambiamos la hora al pc para que ambos tengan la misma
(RHB)
cat /dev/rfcomm0 | grep "GPGGA" > /tmp/hora.txt &
sleep 10
killall cat

## Cogemos la hora de una única línea (1º la hora que está en la
segunda columna separada por , 2º quitamos el .000 del final)
line=$(head -n 1 /tmp/hora.txt)
date=$(awk '{print $2}' FS="," <<< "${line}")
date2=$(awk '{print $1}' FS="." <<< "${date}")
rm /temp/hora.txt

## Separamos la hora de la forma HHMMSS a HH:MM:SS
hora=$(cut -b 1,2 <<< "${date2}")
hora=$((hora+2))
minutos=$(cut -b 3,4 <<< "${date2}")
segundos=$(cut -b 5,6 <<< "${date2}")

#Colocamos la hora
date +%T -s "$hora:$minutos:$segundos"
sleep 10

```

Figura 25. Cambiar la hora de los ordenadores con GPS

Otro script desarrollado, llamado `copiar.sh`, nos permite ejecutar los comandos anteriores al arrancar y en modo superusuario (root). Este script se tiene que ejecutar cada vez que queramos que en las próximas inicializaciones se ejecute el script `init.sh` [ 37 ].

Otro script disponible en `/Descargas/script` es `borrar.sh` que realiza el proceso inverso, elimina el script `init.sh` del inicio automático en el arranque del sistema operativo, es decir, este script será ejecutado cuando queremos que, en los próximos inicios del ordenador, se realice un encendido normal (sin modo OCB y sin realizar las conexiones a los dispositivos bluetooth). El script `copiar.sh` se mantiene activo siempre hasta que lancemos este nuevo script.

El último script que podemos encontrar en `/Descargas/script` es `ssh.sh` que pone operativa la conexión mediante el protocolo ssh y nos permite conectar a cada ordenador a través de una conexión ethernet.

Si se crea un nuevo script, es necesario dar los permisos adecuados de actuación a dicho script con el siguiente comando (permiso de lectura, escritura y ejecución):

```
chmod 755 init.sh
```

### 3.3 Recopilación de datos

El programa OpenC2X nos proporciona un gran número de datos cada vez que se ejecuta. Estos datos se pueden encontrar en la carpeta `OpenC2X/build/logs/`. Cada vez que se ejecuta el programa, se generan 18 ficheros de información nuevos. A continuación, se describen los más utilizados en la realización del trabajo.

- DEMN

Fichero donde se muestra los mensajes tal cual aparecen en la consola. Para abrirlo:

```
nano OpenC2X/build/logs/1_log_DenService(...).log
```

- CAM

Fichero donde recogen los mensajes CAM enviados y recibidos, respectivamente. Los datos más interesantes que almacenan estos ficheros son marcados con la etiqueta HW y forward. En ellos se indica quién ha tramitado el mensaje, el momento en que el mensaje ha sido enviado o recibido en la capa física y qué interfaz ha transmitidos dichos mensajes. En este fichero hay más mensajes registrados, pero solo nos interesan estos. Por esa razón, ejecutamos estas sentencias que filtran los mensajes deseados.

```
nano OpenC2X/build/logs/1_log_Dcc(...).log | grep "Hw:"  
nano OpenC2X/build/logs/1_log_Dcc(...).log | grep "forward"
```

- GPS

El GPS recibe los tiempos en formato *epoch*. Es recomendable utilizar un transformador para visualizar de una manera más sencilla los datos recibidos [ 38 ]. Al tratarse de un fichero con formato csv, se ha utilizado Matlab y Excel para procesar y visualizar los datos.

```
OpenC2X/build/logs/1_stats_GPS(...).csv
```

- OBD

Este fichero almacena la información del OBD, en concreto, la velocidad en cm/seg y el tiempo de cada muestra, también en formato *epoch*.

```
OpenC2X/build/logs/1_stats_Obd2Service(...).csv
```

- Instantes de tiempo para sincronización con el GPS

Se trata de un bloc de notas que contiene los tiempos cuando se ha pulsado el botón en el mando gamepad.

```
nano Logs/GamePad.txt
```

### 3.4 Incidencias de los ordenadores

En este apartado se van a exponer los fallos e incidencias detectadas en los ordenadores durante la realización del proyecto. También se comentará las soluciones temporales que se han encontrado.

#### SSH

Como ya hemos comentado, se ha instalado ssh para poder conectarse a los ordenadores cuando se están realizando las pruebas y no disponemos de teclado, ratón o monitor. Esto es, desde un ordenador portátil nos conectamos mediante ssh a los miniordenadores, de tal manera que no es necesario llevar monitores y otros elementos para controlarlos. Tras

realizar numerosas pruebas, se ha determinado que para realizar una conexión exitosa mediante ssh hay que realizar los siguientes pasos:

1. Encender el ordenador y esperar 3 minutos hasta que todos los scripts se hayan terminado de ejecutar.
2. Tener el cable ethernet **desconectado** del miniordenador y no conectar hasta que hayan pasado los 3 minutos mencionados anteriormente.

### Gamepad

El *gamepad* B a veces no se enciende correctamente, por lo que hay que reiniciarlo, pulsando el botón trasero marcado con *reset*. Tras este paso, hay que realizar de nuevo el proceso de emparejamiento.

### Ordenador A

Este ordenador se apaga solo tras un tiempo de utilización. Este fenómeno comienza cuando la pantalla empieza a temblar y desaparece con un desplazamiento hacia la derecha. Si esto llegase a ocurrir, la luz azul que indica el estado (encendido o apagado) del ordenador se apagará, y al cabo de 5 minutos, el ordenador se apagará por completo. La solución para esto es apagar el ordenador (`poweroff` en consola de comandos o apagar en el menú de inicio) y desconectar la alimentación unos minutos.

### Ordenador B

Este ordenador no dispone de pila en la memoria BIOS, por lo que no es capaz de mantener la hora actualizada de manera constante. Para solucionarlo, se puede conectar a la red (cable ethernet) o mediante las sentencias NMEA recibidas de los GPS.

Ligado con la pila, cada vez que se desee introducir información mediante un dispositivo USB *pendrive*, es necesario montar el sistema de ficheros:

```
mount /dev/sdb1 /media
```

En cuanto a la prueba de campo, el ordenador se inicia automáticamente cuando recibe energía a través del alimentador. En el laboratorio, hay que tener conectada la pantalla por cable HDMI antes de iniciar el pc o no se configurará adecuadamente y no mostrará imagen alguna.

Otro problema asociado a este ordenador es que no dispone de interfaz gráfica porque al inicializar el entorno gráfico se producían muchos problemas y originaba que el sistema se congelase.

### GPS

Prestar mucha atención al realizar pruebas dentro de edificios con los dispositivos GPS, ya que, si no detectan los satélites GPS necesarios, devolverán al ordenador información inválida y no se actualizará la fecha ni leerá los datos con las aplicaciones *cgps* o *xgps*. La solución es colocarlos en una zona exterior o en la ventana del edificio en el que se encuentre.



# 4. Caracterización de los elementos del sistema

En este apartado se explicarán todas las pruebas y cálculos realizados en el desarrollo de este trabajo de fin de Máster. Estas pruebas tienen como objetivo garantizar el correcto funcionamiento del sistema a diferentes distancias entre los vehículos y caracterizar todos y cada uno de los elementos activos o pasivos en la comunicación.

Al inicio del desarrollo de este trabajo, disponíamos de una serie de características que se debían cumplir para que la comunicación entre los ordenadores fuese correcta. Inicialmente, supusimos que esos valores se cumplían y la comunicación iba a ser exitosa. Durante las primeras pruebas de campo, descubrimos que los ordenadores no eran capaces de establecer la conexión estando a unas pocas decenas de metros (~ 30 metros).

Por esta razón, se decidió detectar cuál era el problema y buscar una solución. Tras descubrir los errores a pequeña distancia, nos propusimos caracterizar cuál era la distancia máxima de funcionamiento entre los dos dispositivos, tanto teórica como experimental.

## 4.1 Cálculo teórico de la distancia máxima de comunicación

Para obtener una distancia teórica, se han utilizado las fórmulas habituales de propagación de ondas electromagnéticas en campo abierto. Estos cálculos se han realizado con una gran cantidad de aproximaciones, ya que no era el objetivo de este proyecto hacer una descripción teórica rigurosa de la comunicación a través del estándar de capa física 802.11p, sino que deseábamos obtener una estimación suficientemente acertada para saber si el comportamiento de nuestro sistema se aproximaba a lo que se podría esperar. Una fuente donde se explica con profundidad cómo realizar este estudio es [ 39 ].

Para calcular la estimación deseada se utilizó la fórmula de propagación de una onda en espacio libre representada en la Fórmula 1 y Fórmula 2.

$$P_r(\text{dBm}) = P_t(\text{dBm}) + G_{ta}(\text{dB}) - L_{tf}(\text{dB}) - L_{path} + G_{ra}(\text{dB}) - L_{rf}(\text{dB})$$

*Fórmula 1. Propagación de una onda en espacio libre*

donde  $P_r$  es la potencia recibida,  $P_t$  es la potencia transmitida,  $G_{ta}$  es la ganancia de la antena transmisora,  $L_{tf}$  son las pérdidas de alimentación del transmisor,  $L_{path}$  son las pérdidas de la comunicación,  $G_{ra}$  es la ganancia de la antena receptora y  $L_{rf}$  son las pérdidas de la alimentación del receptor.

$$Loss(\text{dB}) = 32,45 + 20 \log_{10}(\text{frecuencia en MHz}) + 20 \log_{10}(\text{distancia en Km})$$

*Fórmula 2. Fórmula de pérdidas debidas a la propagación en espacio libre*

Estas ecuaciones se utilizarán como si dispusiéramos de un sistema SISO, es decir, una antena para transmisión y para recepción, aunque nuestro sistema sea MIMO 2X2 (dos antenas de recepción y transmisión). Se espera que con el sistema MIMO se logren mejorar los resultados obtenidos para el sistema SISO.

A pesar de ser las ecuaciones más sencillas que podemos utilizar para el cálculo de pérdidas en espacio libre y alcance máximo, no está exento de dificultades aplicarlas, fundamentalmente porque algunos de los parámetros que hay que introducir para el cálculo nos resultan desconocidos o no tenemos razones para fiarnos de los datos que nos da el fabricante. Por ejemplo, las antenas WiFi que se utilizaban, solo venían con información muy poco fiable del fabricante chino, y no estaban realmente caracterizadas para la banda que nos interesaba. Por lo tanto, en las siguientes secciones se desgranarán los cálculos y pruebas que se hicieron para caracterizar cada uno de los componentes, fundamentalmente la ganancia de las antenas, la potencia emitida por las tarjetas y conocer qué modulación estaban usando las tarjetas para conocer la sensibilidad de los receptores.

## **4.2 Caracterización de los elementos del sistema en laboratorio**

A continuación, se explican todos los experimentos que se realizaron en el laboratorio de investigación del Grupo de Comunicaciones Ópticas de la Universidad de Valladolid. Estos experimentos se realizaron en un entorno controlando con el objetivo de obtener un valor razonable sobre las características que no podemos conocer a través de los *datasheets* disponibles o de los que tenemos dudas.

Los parámetros bajo estudio fueron:

- Ganancia de la antena de WiFi, que denominaremos cuadrada
- Coeficiente de reflexión  $s_{11}$  de la misma antena
- Potencia real transmitida por la tarjeta de red inalámbrica
- Modulación utilizada por la tarjeta de red inalámbrica (relacionada con la sensibilidad del receptor)

Todos estos valores serán utilizados en la fórmula del alcance máximo, vista en el apartado 4.1, Cálculo teórico, para lograr una distancia de comunicación máxima aproximada.

### **4.2.1 Tipos de antenas disponibles**

Al inicio del proyecto, disponíamos de tres tipos diferentes de antenas que podían ser usadas para la comunicación entre los miniordenadores. Para que fuese más sencillo, dimos nombre a estos tres tipos de antenas en función de su forma:

- Antenas de comunicación inter-vehiculares modelo GSA.8859 Mini DSRC: Estas antenas están principalmente diseñadas para realizar comunicaciones entre vehículos. Diseñadas y distribuidas por Taoglas. Las llamaremos antenas planas y se muestran en la Figura 26 (a).

- Antenas WiFi tipo 1: De estas antenas no tenemos ninguna información complementaria, más que la geometría de la antena de su interior y la forma del recubrimiento que la protege. Las llamaremos Antenas WiFi cuadradas y se pueden apreciar en la Figura 26 (b).
- Antenas WiFi tipo 2: Para estas antenas estamos en la misma situación que para las antenas WiFi de tipo 1. Las conoceremos como antenas WiFi redondas y su forma se ve en la Figura 26 (c).

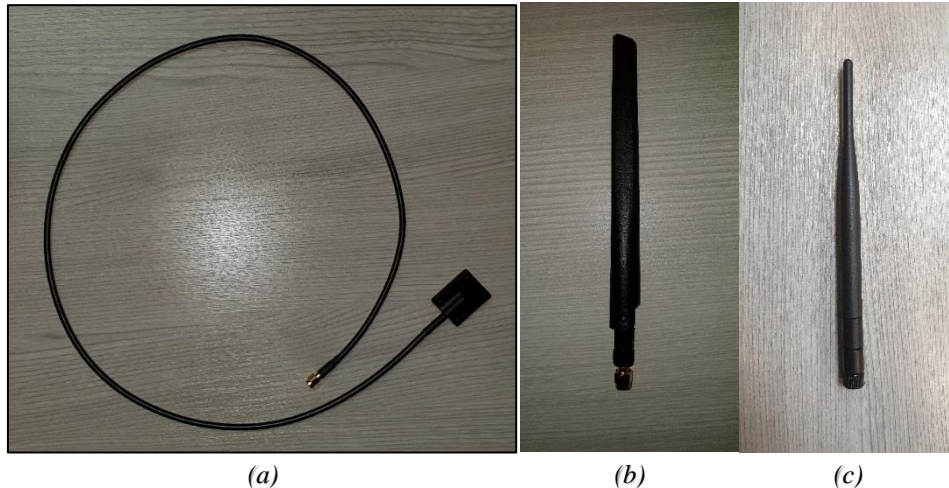


Figura 26. (a) Antena plana GSA.8859 (b) Antena WiFi cuadrada (c) Antena WiFi redonda

#### 4.2.2 Estimación de la ganancia de las antenas

Uno de los parámetros que necesitamos para aplicar la función de alcance máximo es la ganancia de las antenas receptora y emisora. En nuestra configuración, ambas ganancias serán iguales ya que estamos utilizando antenas similares en ambos lados de la comunicación. Para calcular la ganancia de la antena se han seguido dos métodos diferentes, en función de la antena. Las antenas WiFi redondas no fueron modeladas ya que se descartaron previamente y la realización de su modelado era bastante complicado.

##### Antenas de comunicación planas modelo GSA.8859 Mini DSRC

Para obtener la ganancia de las antenas diseñadas específicamente para comunicaciones inter-vehiculares se ha utilizado el *datasheet* de dicha antena. (Figura 27). [ 40 ]

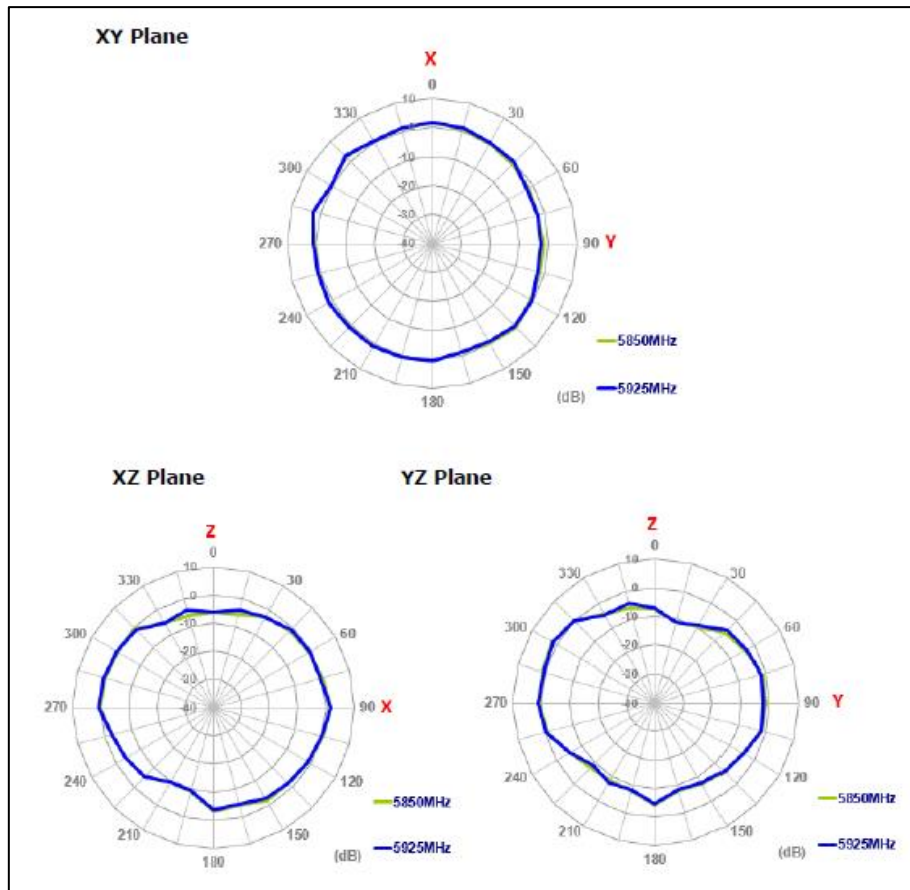


Figura 27. Ganancia de la antena plana según el datasheet

Como podemos apreciar en la Figura anterior el plano XY es donde se produce la mejor ganancia, variando entre 3 dB en el eje X hasta -2dB en el eje Y. Con estos datos y conociendo cuales son los ejes en la antena (Figura 28) colocaremos la antena de forma que el eje con mayor ganancia apunte en la dirección donde circulan los vehículos. Por esta razón, se colocaron las antenas en las ventanillas traseras de los vehículos (es la configuración óptima para la ganancia y no molestaría al conductor durante el experimento final) (Figura 29).

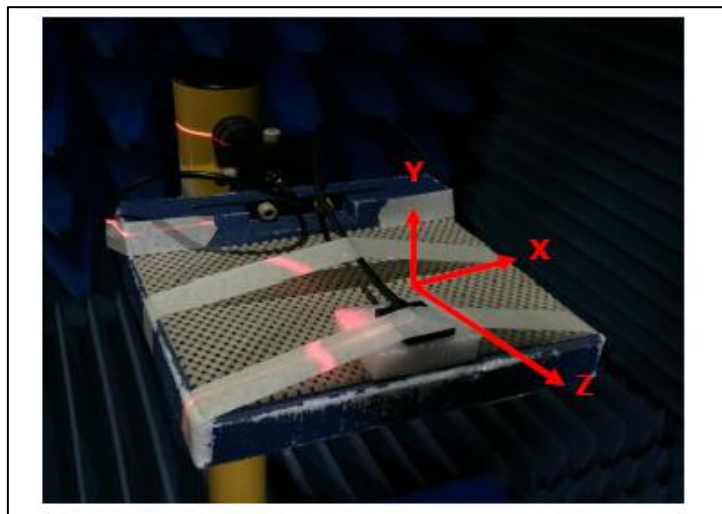


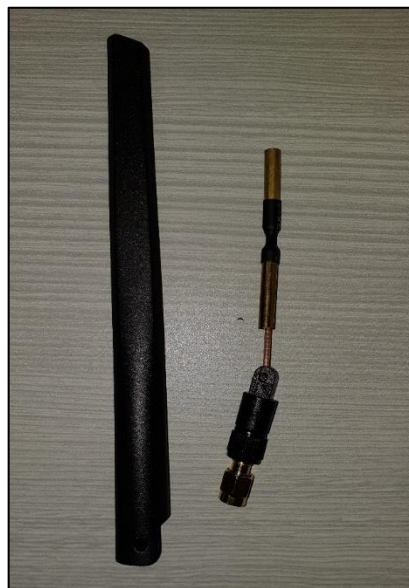
Figura 28. Ejes de la antena



*Figura 29. Configuración de la antena en el vehículo*

### **Antenas cuadradas de comunicación WiFi**

Para la antena WiFi, al no disponer del *datasheet* correspondiente, se decidió calcular un valor aproximado de la ganancia mediante el programa 4NEC2. Este software de código abierto permite simular una gran variedad de antenas, como modelos ya predefinidos (dipolo, bicónica, ...) o formas menos convencionales mediante un editor. En la Figura 30 se muestra la distribución que esconde nuestra antena bajo el plástico protector. [ 41 ]



*Figura 30. Forma de la antena WiFi*

En 4NEC2, introduciremos esta geométrica aproximándola por un dipolo simple de las siguientes dimensiones: 5 cm de largo y 0,05 cm de ancho, las cuales han sido tomadas

con un micrómetro (Figura 31). Para simplificar el modelado, se ha obviado la pequeña hendidura de la zona central del dipolo, que es donde se realiza la alimentación de este.

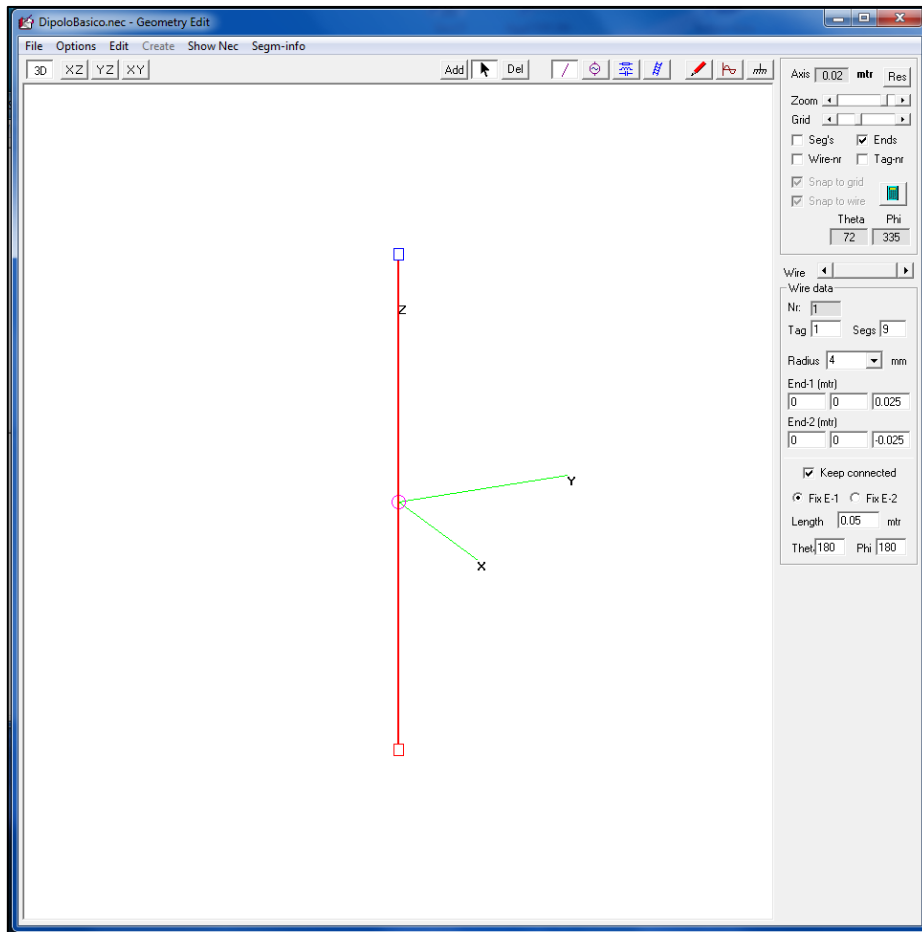


Figura 31. Dipolo simple en 4NEC2

Una vez que tenemos introducida la antena en el programa, realizamos un cálculo del patrón de campo lejano en la frecuencia de interés (Figura 32). Con este cálculo, obtendremos los diagramas de campos correspondientes a la antena. Como se ve en la Figura 33, la radiación de esta antena es omnidireccional (forma de rosco). La ganancia de esta antena es 4,07 dB (Figura 34).

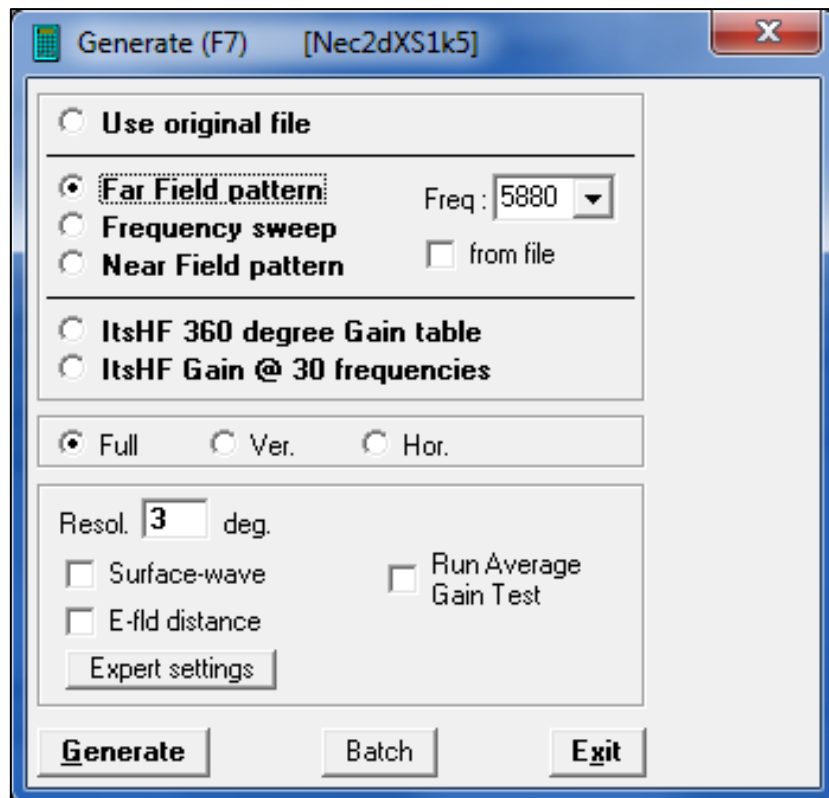


Figura 32. Patrón de campo lejano de la frecuencia bajo estudio

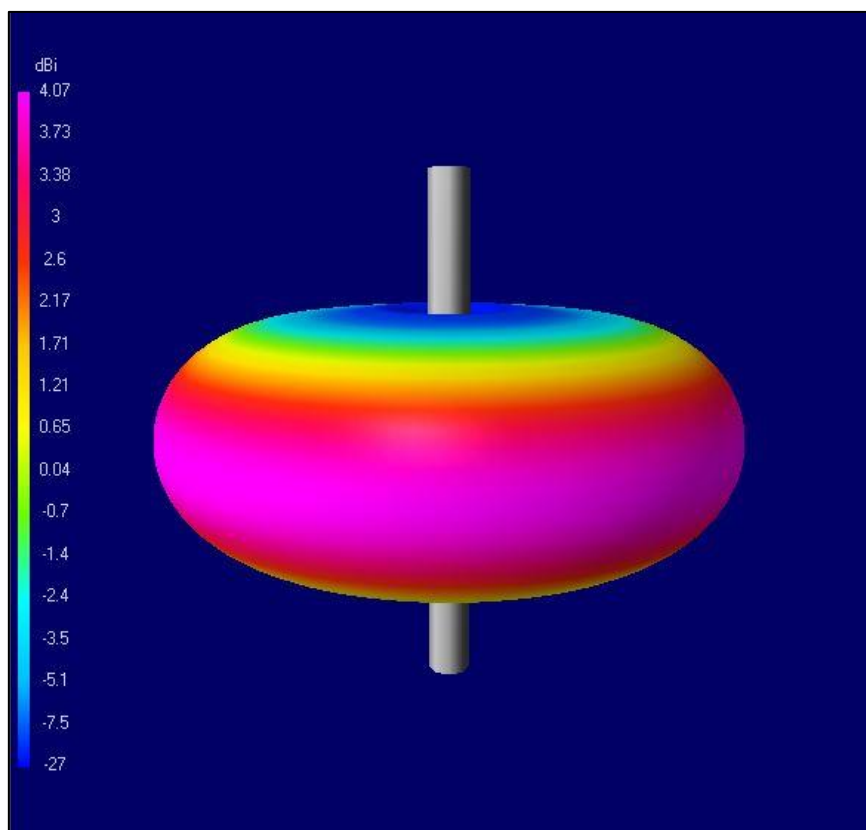


Figura 33. Imagen 3D del diagrama de campo lejano

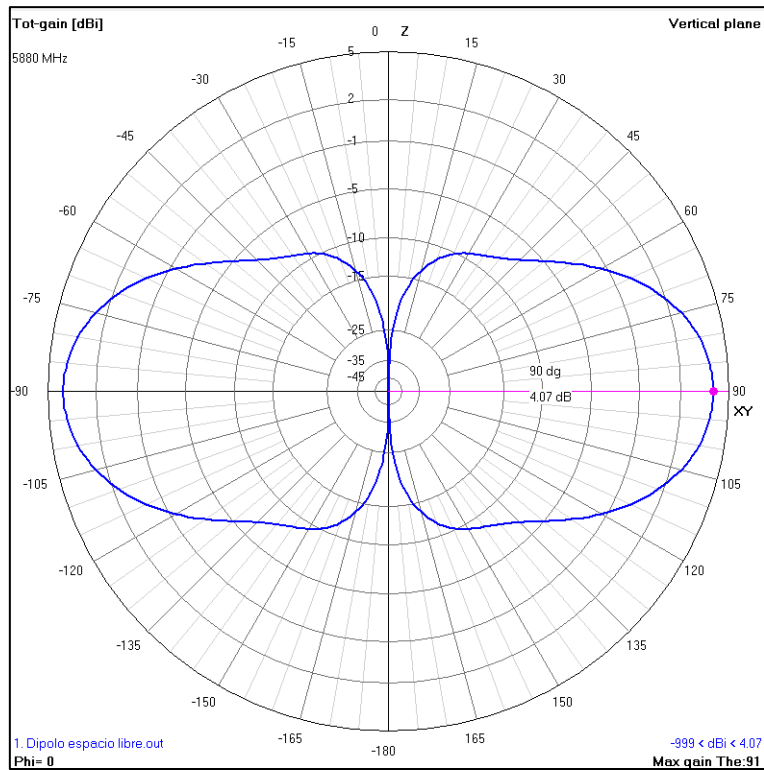


Figura 34. Ganancia del dipolo en espacio libre

Para profundizar más en la ganancia de esta antena, se realizó una segunda configuración añadiendo suelo simulado de propiedades moderadas, que en el programa se le da unas características de conductividad de 0,003 y una constante dieléctrica de 4. Con esta nueva simulación obtenemos el siguiente diagrama (Figura 35 y Figura 36) y una ganancia de -5,62 dB.

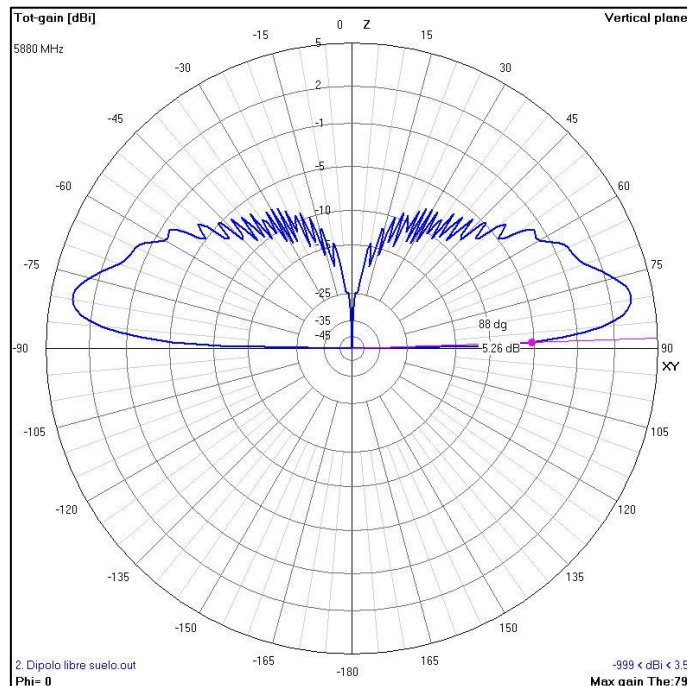
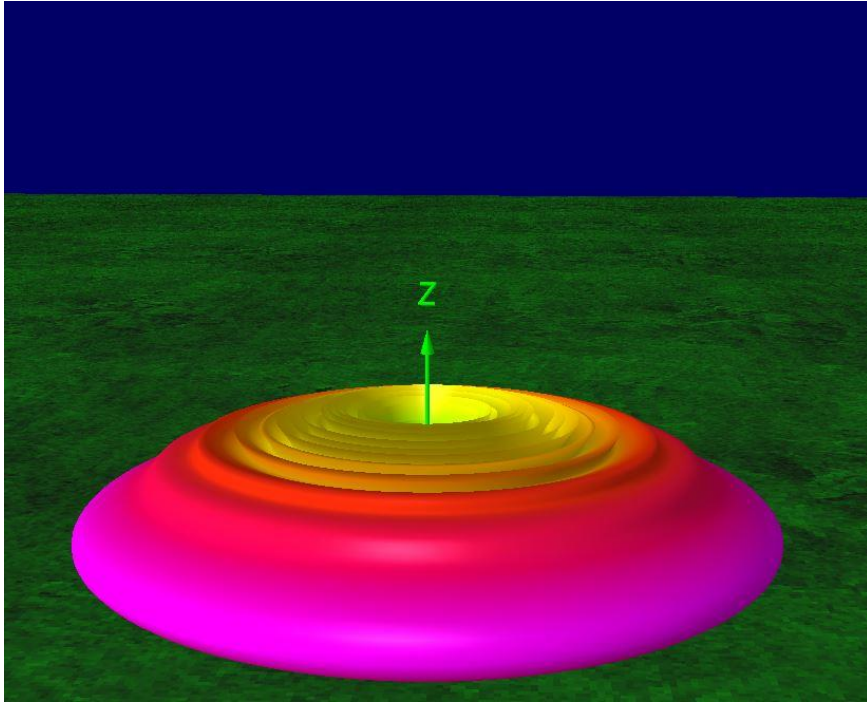


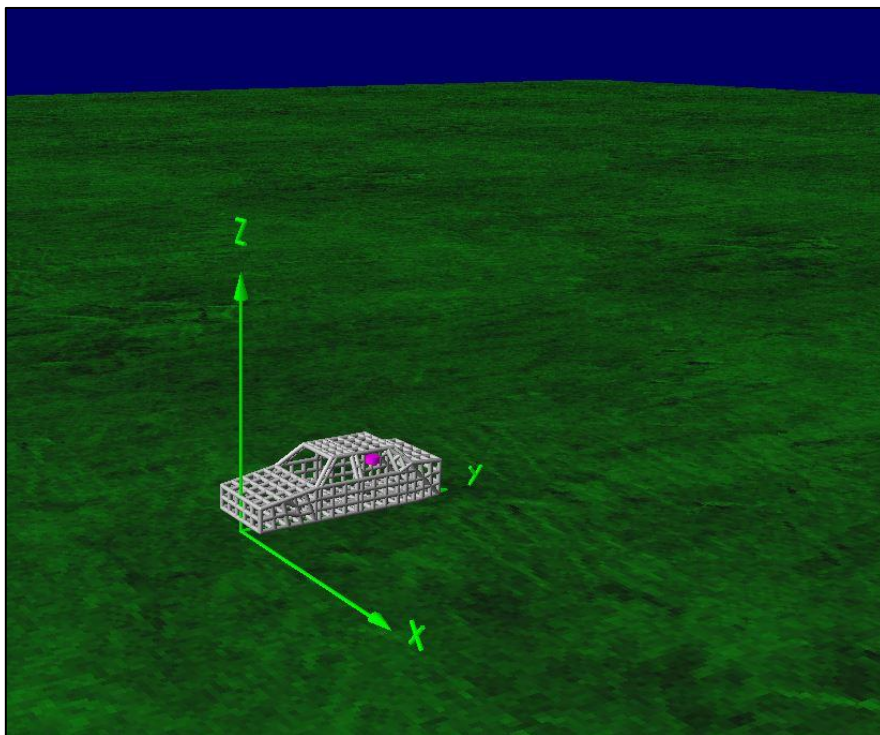
Figura 35. Diagrama de campo lejano con suelo





*Figura 36. Diagrama 3D de la configuración con suelo*

Por último, se intentó modelar un vehículo dentro de 4NEC2 y colocar la antena en una posición similar a donde iría en el caso de uso final. El objetivo de esta configuración es observar cómo interactúa nuestra antena con la carrocería de las proximidades (Figura 37 y Figura 38).



*Figura 37. Modelo 3D del automóvil*

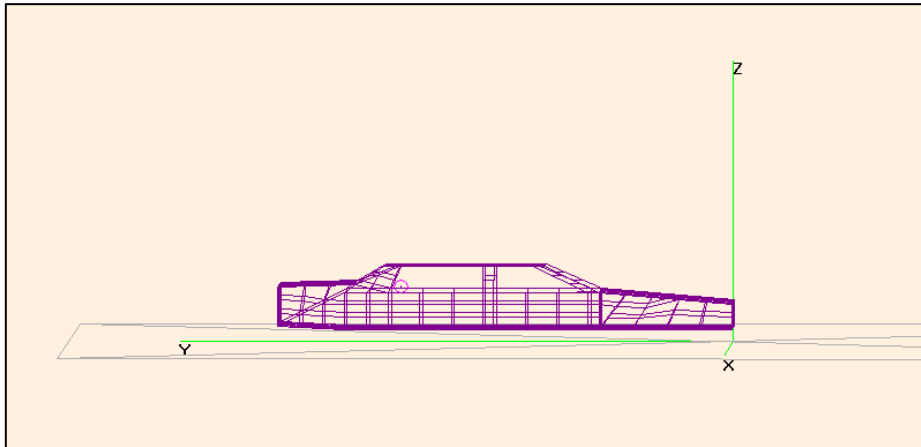


Figura 38. Perfil del modelo 3D del automóvil

Con esta nueva simulación obtenemos que la ganancia de esta antena sería  $-23,2$  dB. Finalmente, se ha decidido no utilizar este resultado, ya que necesitaríamos un modelado más detallado, donde la carrocería del vehículo fuese láminas continuas de material que no permita las transmisiones a través de la carrocería del vehículo, y esto es imposible de realizar en 4NEC2 porque está limitado en cuanto a la memoria de procesamiento que es capaz de gestionar.

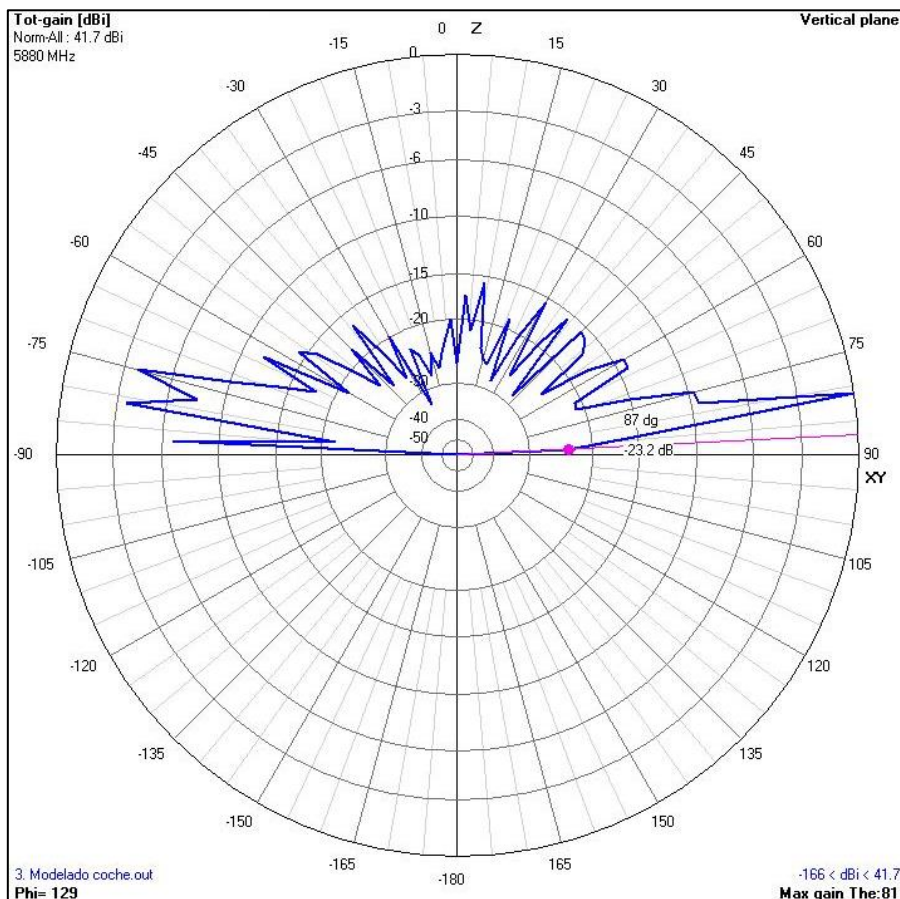


Figura 39. Ganancia con el automóvil

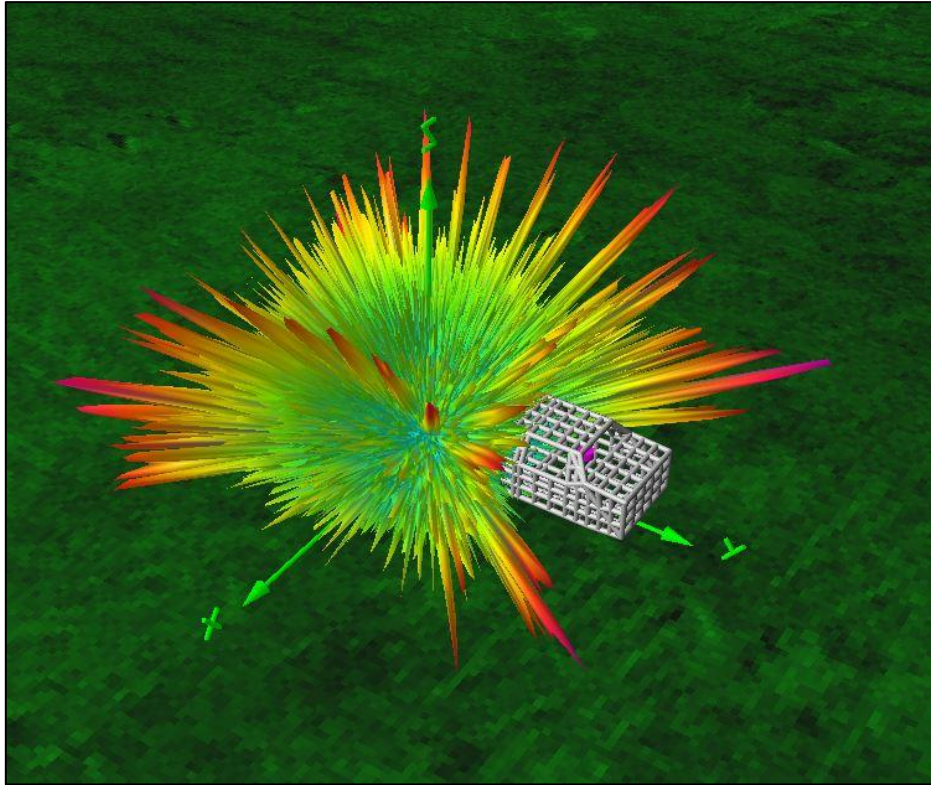


Figura 40. Diagrama de radiación en 3D

#### 4.2.3 Coeficiente de reflexión de potencia $s_{11}$

El siguiente parámetro bajo estudio indica cuánta de la potencia que suministramos a la antena se acopla y transmite de manera correcta. Este parámetro depende de la antena, más en concreto, su forma, diseño, materiales, etc., y de la frecuencia de trabajo en la que nos encontremos, en nuestro caso, 5880 MHz.

Para obtener este parámetro se pueden utilizar dos métodos:

- Modelar las antenas en 4NEC2 y realizar un “*Frequency Sweep*”. Este método no se ha utilizado por dos razones. La primera es que no sabemos cuál es la geometría de este tipo de antenas planas, y la segunda razón es que, aun conociendo la forma de la antena, tendríamos que utilizar un gran número de simplificaciones durante el modelado (ya explicadas con anterioridad en la ganancia de la antena).
- Utilizar un analizador de redes. Se ha escogido este método ya que nos permite una medida fiable de este parámetro y por lo tanto una mejor aproximación de la distancia máxima de comunicación, aparte de que disponíamos del equipamiento necesario en los laboratorios de la escuela. El analizador de redes utilizado es un HP Hewlett Packard 8719D que permite barrer frecuencias entre 50 MHz hasta 13,5 GHz (Figura 41).

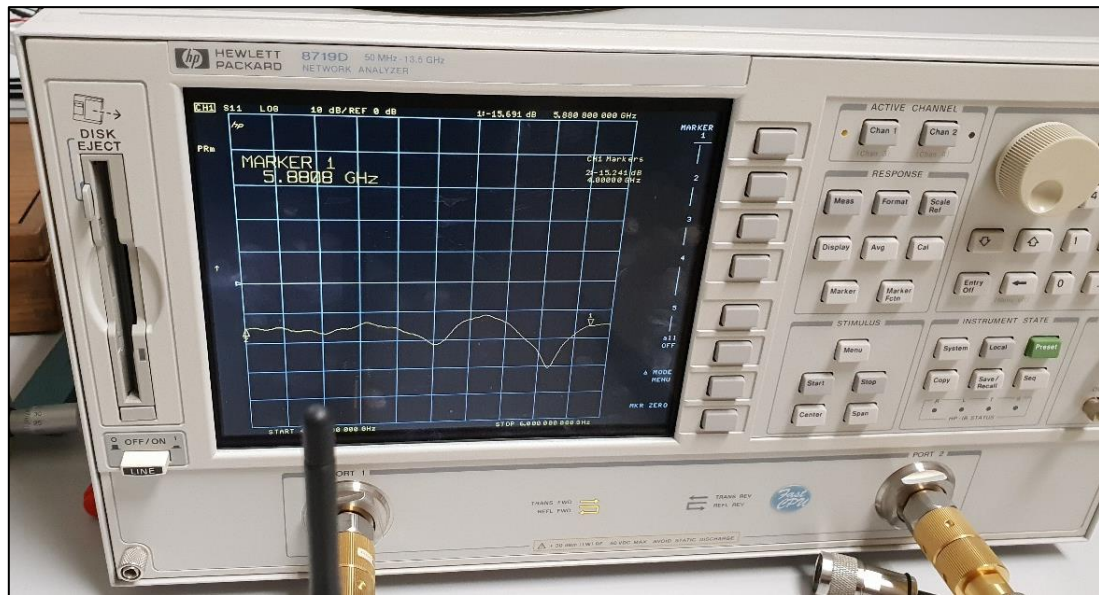


Figura 41. Analizador de redes

Para obtener este parámetro se han tenido que seguir los siguientes pasos [ 42 ]:

1. Calibración del analizador de redes con una serie de cargas adaptadas SMA (cortocircuito, circuito abierto y filtro paso-banda). Para ello, hay que entrar en el menú de calibración e ir conectando todas las cargas en uno de los puertos del analizador.
2. Ajustar el rango de frecuencias deseadas, conectar las antenas bajo estudio en el puerto calibrado y obtener la gráfica del parámetro  $S_{11}$  correspondiente.
3. Guardar los valores obtenidos en un disquete en forma ASCII y representarlos en una gráfica, en nuestro caso, utilizando el software MATLAB.

La Figura 42 muestra cómo varía este coeficiente  $s_{11}$  en función de las frecuencias. Hay representadas tres gráficas de forma conjunta, para comparar de una manera más sencilla este parámetro para cada una de las antenas que están bajo estudio. La Figura 43 muestra esa misma gráfica en nuestra zona de interés, es decir, en las bandas de frecuencia del estándar ETSI ITS G5.

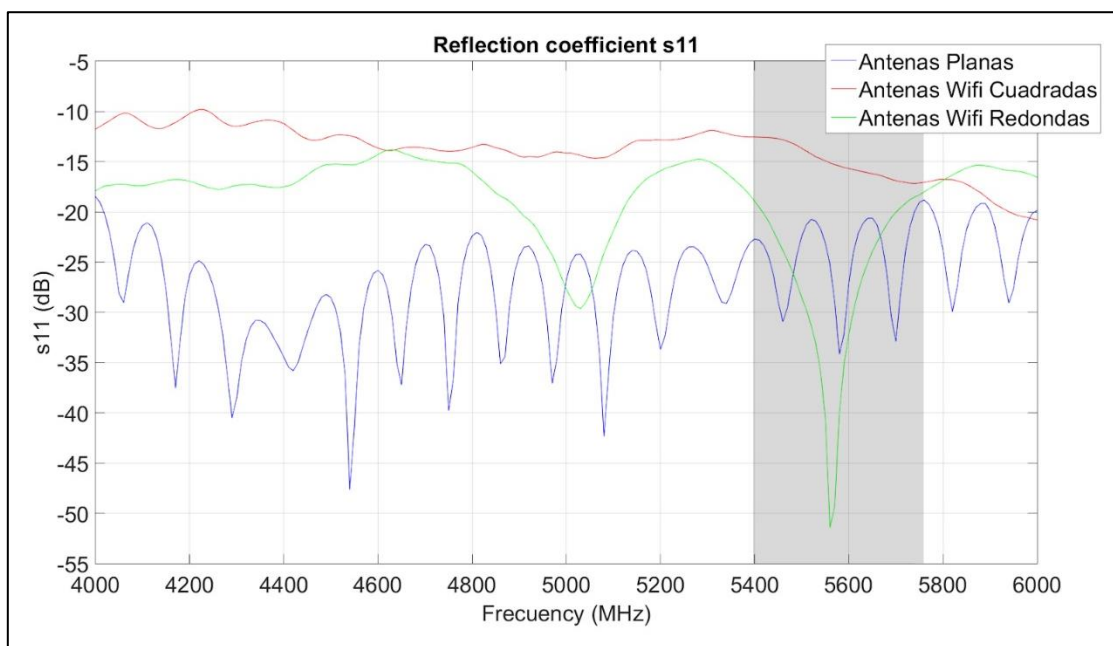


Figura 42. Coeficiente  $s_{11}$  en función de la frecuencia

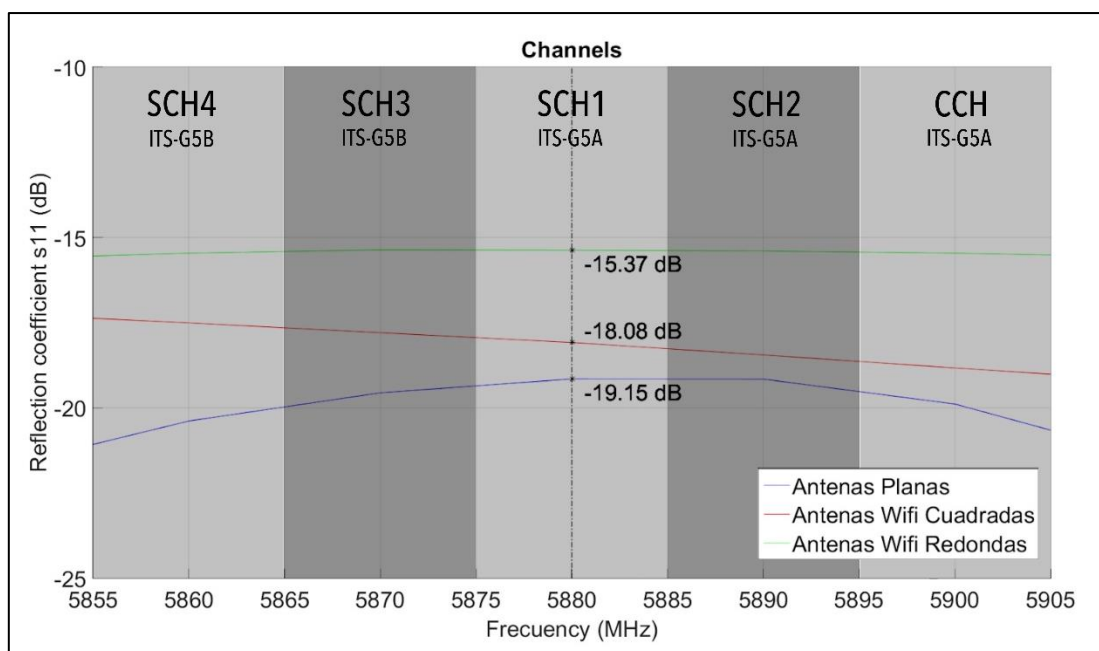


Figura 43. Coeficiente  $s_{11}$  en el rango de frecuencias de interés y canales utilizados

#### 4.2.4 Potencia real transmitida por la tarjeta de red

Una de las grandes incertidumbres que hemos tenido a lo largo de este trabajo fue conocer cuánta es la potencia real que nos suministra nuestras tarjetas de red Atheros ATH9K AR5BHB116 (Figura 44), concretamente con respecto a lo especificado en el software de control, y si esa potencia se divide entre los dos puertos de entrada/salida o es la emitida en cada uno de ellos (3 dBs de diferencia).



Figura 44. Tarjeta de red Atheros ATH9K AR5BHB116

Por esta razón, fue necesario diseñar un experimento que nos permitiera medir la potencia en la salida de las conexiones. Para conseguirlo, se utilizó un analizador de espectros Rohde&Scharz (Modelo FSH6 de 100 kHz hasta 6 GHz) (Figura 45). [ 43 ]



Figura 45. Analizador de espectros Rohde&Scharz

La configuración consistió en conectar una de las salidas del ordenador con este analizador de espectros mediante un cable alargador MDA (Figura 46). Previamente hemos calculado las pérdidas que añade este cable a la potencia transmitida con el

analizador de redes, como hemos visto en el apartado anterior "Coeficiente de reflexión de potencia  $S_{11}$ ". Este cable añade 2,7 dB de pérdidas a la frecuencia de interés, que posteriormente serán añadidos al resultado que obtengamos con el analizador de espectros.



Figura 46. Alargador con conexiones MDA

Una vez conectado el ordenador con el analizador de espectros, vamos a configurar éste para que nos proporcione la potencia total transmitida en el ancho de banda de nuestro interés, es decir, 10 MHz en torno a la frecuencia de 5,880 GHz. Para ello, seleccionamos la opción "Measure  $\rightarrow$  Channel Power". Dentro de este modo hay varias posibles formas de medida, que se pueden encontrar en "Trace  $\rightarrow$  Detector". La recomendada por el manual y la que se puede encontrar por defecto es la denominada RMS (Root Median Square)

El siguiente paso consiste en modificar la potencia transmitida mediante un comando de Linux: `iwconfig ocb0 txpower x` (en dBm). El comando que se ha utilizado para transmitir de forma contante con el ordenador es `iperf`. Para ello, lanzaremos el comando: `iperf -s` en el ordenador servidor, el cual no tiene conexión con el analizador de espectros y el comando: `iperf -c [ip del servidor]` en el ordenador conectado al analizador. [ 44]

Con este proceso, se han barrido todas las potencias que permite la tarjeta de red, que van desde 3 dBm hasta 18 dBm, y se han probado todas las tarjetas disponibles en el laboratorio (un total de 4 tarjetas iguales, numeradas, para saber cuál es la que mayor potencia proporciona). Los resultados se muestran en la Tabla 1.

Potencia transmisión solicitada	Tarjeta 1 (dBm)	Tarjeta 2 (dBm)	Tarjeta 3 (dBm)	Tarjeta 4 (dBm)
3	-2,91	-4,19	-3,44	-3,79
4	-2,6	-3,58	-3,22	-3,55
5	-2,15	-2,56	-2,92	-3,3
6	-1,8	-1,82	-2,58	-3
7	2,57	1,75	1,26	0,54
8	3,12	3,13	1,5	0,85
9	4,52	3,59	2,7	2,2

10	4,7	4,63	3,06	2,7
11	6,18	5,15	4,74	4,25
12	6,84	6,72	5,26	5,3
13	7,01	8,03	6,73	6,72
14	8,41	8,31	7,75	7,48
15	9,4	8,74	9,32	7,74
16	9,88	10,1	9,64	9,08
17	10,25	10,45	9,9	9,64
18	11,7	11,51	11,23	10,76

Tabla 1. Potencias transmitidas por el ordenador A con detector RMS.

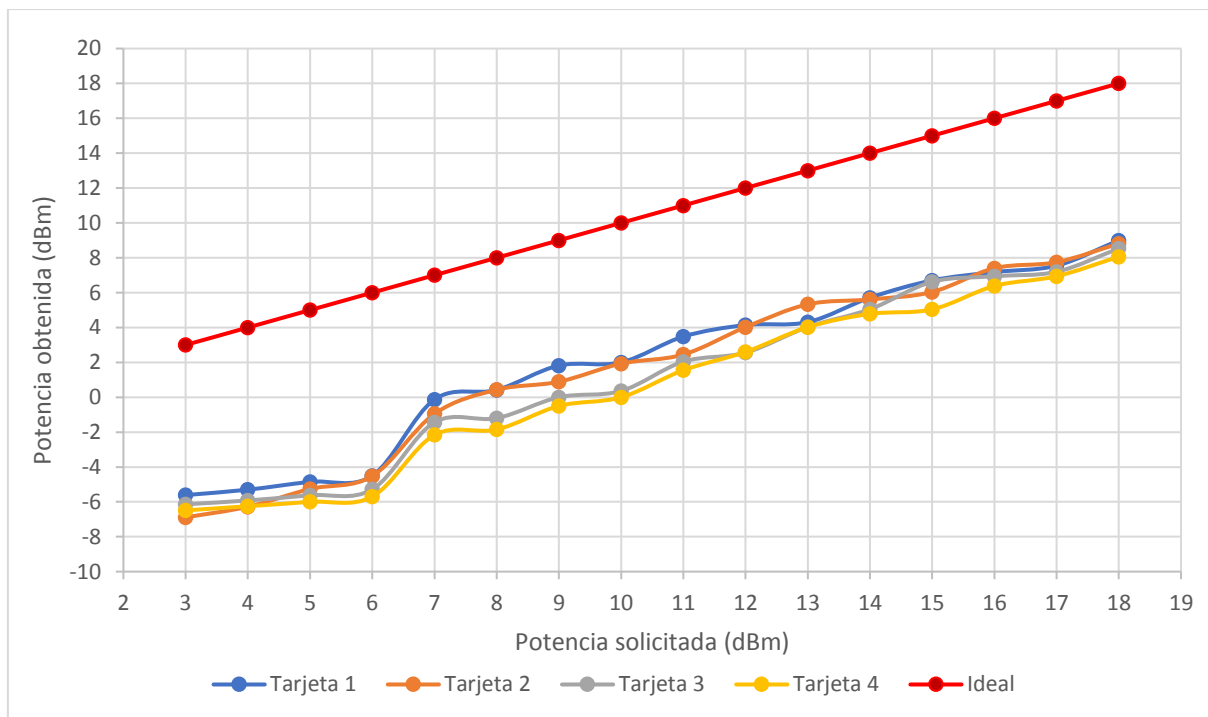


Figura 47. Potencias transmitidas por el ordenador A con detector RMS

Como se puede comprobar, las medidas así tomadas están muy lejos de lo que en teoría dan las tarjetas. Esto se debe básicamente a que las tarjetas no están continuamente emitiendo, y por lo tanto el valor que nos está dando el equipo será más bajo de lo que debería por ser una media a lo largo del tiempo. Tiene por lo tanto en este caso más sentido utilizar la opción de utilizar la potencia de pico que detecte el equipo. Aunque de esta forma la medida podría estar algo falseada dado que el equipo de medida registrará las variaciones naturales al alza durante la transmisión, será mucho más exacta al eliminar los tiempos muertos de transmisión que estábamos contando anteriormente. Para detectar el máximo seleccionaremos “Trace → Detector → Max peak”. Podemos comprobar que hemos cambiado el modo de detección activo mirándolo en la esquina superior izquierda de la pantalla del equipo. Los resultados se pueden ver en la Tabla 2, que como podemos ver están mucho más ajustados a lo que se espera.

Otra forma para calcular este valor es mediante el modo *analyzer* y calcular esta potencia manualmente. Para ello seleccionamos “Meas → Measure → Analyzer” y mantenemos



el máximo de potencia mediante “Trace → Trace mode → Max hold”. Las fórmulas para calcular en este caso la potencia del canal son:

- Buscar el ancho de banda entre los bordes del trapecio → 7,5 MHz.
- Dividir el ancho de banda entre el valor Resolution BandWidth (por ejemplo, RBW = 300 KHz)
- Calcular el logaritmo del cociente, por ejemplo,  $\log_{10} \left( \frac{7,5}{0,3} \right) = 1,4 \text{ dB}$
- Sumar este valor a la máxima potencia detectada en la banda de transmisión y a las pérdidas introducidas por el cable de conexión, en el ejemplo que estamos llevando:  $18 + 1,4 + 2,7 = 22,1 \text{ dBm}$

Utilizaremos estos datos como aproximaciones, ya que lo ideal sería coger la media de toda la potencia cuando se está transmitiendo un símbolo, pero con nuestro equipo es imposible realizar este proceso.

Potencia transmisión solicitada	Tarjeta 1 (dBm)	Tarjeta 2 (dBm)	Tarjeta 3 (dBm)	Tarjeta 4 (dBm)
3	4,84	5,1	2,15	4,85
4	5,72	5,85	3,2	5,9
5	6,5	6,52	4	6,7
6	7,41	7,57	4,63	7,33
7	12,01	12,06	8	10,7
8	13,03	12,7	9,2	11,9
9	13,7	13,51	10,33	13,03
10	14,6	14,4	11,1	13,8
11	15,54	15,5	12,1	14,8
12	16,3	17,2	13	15,7
13	17,1	17,9	14,5	17,2
14	17,9	18,56	15,4	18,1
15	18,77	19,43	16,24	18,94
16	19,64	20	16,83	19,53
17	20,2	20,9	17,71	20,41
18	21,02	21,4	18,4	21,1

Tabla 2. Potencias transmitidas por el ordenador A con detector de potencia máxima

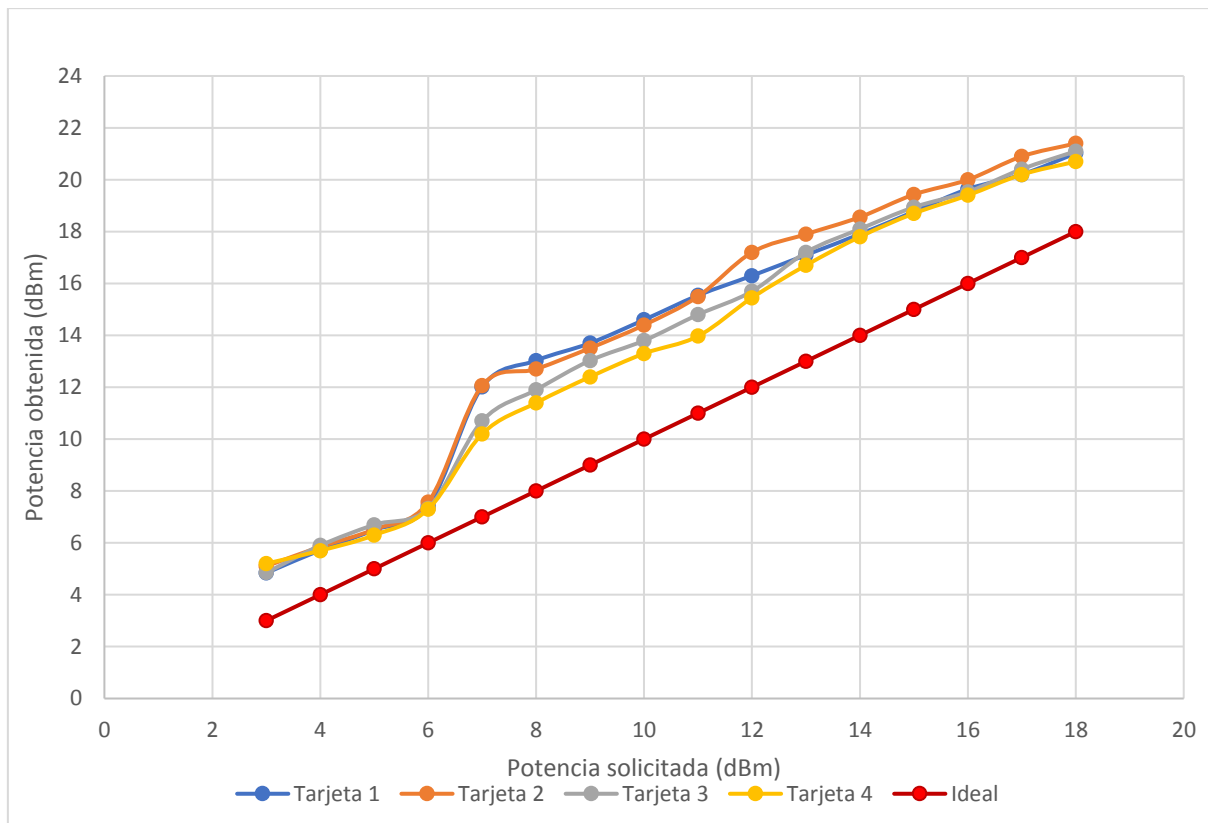


Figura 48. Potencias transmitidas por el ordenador A con detector de potencia máxima

Finalmente, se ha decidido instalar la tarjeta número 1 en el ordenador A ya que, en cuanto a potencia media, nos proporciona 11,7 dBm de salida y la tarjeta número 4 en el ordenador B, con 10,76 dBm de potencia de salida (el funcionamiento de las tarjetas en el ordenador B era igual que en el ordenador A). Esta potencia calculada es la potencia suministrada por cada antena de forma individual. Finalmente, se decidió analizar sistema como si se tratase de un sistema de una sola antena para recepción y otra para transmisión, ya que el cálculo del sistema MIMO no entraba en los objetivos de este proyecto, dado requeriría una simulación con algún software específico. Se espera que los datos obtenidos a partir de estos valores se vean mejoradas por el uso del sistema MIMO.

Es interesante notar que los 18 dBm que marcan como salida de potencia máxima las tarjetas son unos 9 dB inferiores a lo que marca el estándar, por lo que, el alcance máximo de la comunicación resultante podría verse aumentando si decidiésemos instalar otras tarjetas que nos proporcionen mayor potencia de media. Utilizaremos en la fórmula de propagación de una onda las potencias obtenidas mediante el método de la potencia máxima.

#### 4.2.5 Modulación utilizada por la tarjeta

La estándar 802.11p señala la utilización de diversas modulaciones en función de las pérdidas detectadas en el medio. En el estándar no se aclara si el control de la modulación debe ser hecho automáticamente por la propia tarjeta o si por el contrario debe ser algún tipo de servicio de orden superior. En esta sección en base a las pruebas que se realizaron aclaramos ambos puntos, el tipo de modulación utilizada y quién está al cargo de modificar dicha modulación.

Determinar el tipo de modulación utilizada es relevante, dado que la sensibilidad del receptor varía en función de ésta. La tarjeta de red de la que disponemos no es capaz de indicarnos cual es la tasa de transmisión que está siendo utilizada en cada momento cuando estamos en modo OCB (utilizando el comando `iwconfig`), al contrario que en los otros modos. Por ello, se buscó una solución que nos permitiera observar la modulación, constelación y la tasa de transmisión asociada (*bit rate*) de manera experimental.

La configuración utilizada es la siguiente: colocamos dos ordenadores, transmitiendo a muy baja potencia y separadas en torno a 3 metros. A estos ordenadores les hemos colocado una carga adaptada de  $50 \Omega$  en una de sus salidas, para simplificar el experimento a un único canal de comunicación (Figura 49 y Figura 50).



Figura 49. Impedancias adaptadas



Figura 50. Impedancia adaptada colocada en el ordenador

Adicionalmente, tenemos un tercer ordenador que dispone de una tarjeta auxiliar USRP-210 (Figura 51) y el software GNU radio [ 45 ][ 46 ] [ 47]. Esta tarjeta la utilizamos para muestrear la señal transmitida y analizarla para dibujar la constelación que se recibía. Dado que estaba fuera de los objetivos de este trabajo fin de máster crear un decodificador capaz de hacer este tipo de análisis utilizando el software de GNU radio, se utilizó un proyecto de software libre [ 49 ].

Dado que la instalación fue bastante compleja, ya que requiere de un gran número de dependencias para que el software funcione correctamente, y se tuvo que realizar bastantes pruebas antes de conseguir que el programa funcionara correctamente, describimos a continuación el proceso de instalación. En primer lugar, hay que instalar una serie de paquetes antes de instalar UHP. La dirección donde podemos encontrar estos paquetes:

[https://kb.ettus.com/Building and Installing the USRP Open-Source Toolchain \(UHD and GNU Radio\) on Linux#Update and Install dependencies](https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux#Update_and_Install_dependencies)

A continuación, instalamos las bibliotecas de USRP Hardware Driver (UHD) del enlace [http://files.ettus.com/manual/page\\_install.html](http://files.ettus.com/manual/page_install.html). Linux permite descargar estos paquetes directamente desde el repositorio mediante el comando: `sudo apt-get install libuhd-dev libuhd003 uhd-host`. Estas dependencias también se pueden descargar de GitHub: <https://github.com/EttusResearch/uhd>

En la página de USRP Hardware Driver and USRP Manual se pueden encontrar las instrucciones para instalar todo lo referido a GitHub [http://files.ettus.com/manual/page\\_build\\_guide.html](http://files.ettus.com/manual/page_build_guide.html)

Para continuar con el proceso de instalación, es necesario conocer cuál es el driver compatible que hay que cargar en el dispositivo USRP. En nuestro caso, hemos utilizado el NI-USRP *Configuration Utility* y vamos a cargar `usrp_b210_fpga.bin`, que imitaría una tarjeta modelo B210. Esta utilidad se descarga de: <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019MyFSAU&l=es-ES>

Únicamente es necesario instalar la versión más reciente de UHD. La primera vez que conectemos la tarjeta USRP, el software UHD cargará automáticamente la imagen FPGA y el firmware a utilizar.

Finalmente, será necesario instalar la configuración desarrollada por Bastibl [ 48 ][ 49 ]. La instalación necesaria se encuentra descrita detalladamente en el fichero *readme.md* de <https://github.com/bastibl/gr-ieee802-11>. Esta configuración nos proporciona un transmisor-receptor WiFi, que utilizaremos para detectar la modulación. Para lanzar esta funcionalidad, debemos buscar el ejecutable *wifi\_rx.grc* dentro del directorio `gr-ieee802-11/examples/`. [ 48 ][ 49 ]

El ordenador, donde hemos realizado esta instalación, estará ejecutando la aplicación de GNU radio que recibe la modulación que se está utilizando en el medio, realiza una demodulación completa y muestra por pantalla la constelación resultante (Figura 52).



Figura 51. Tarjeta USRP-210

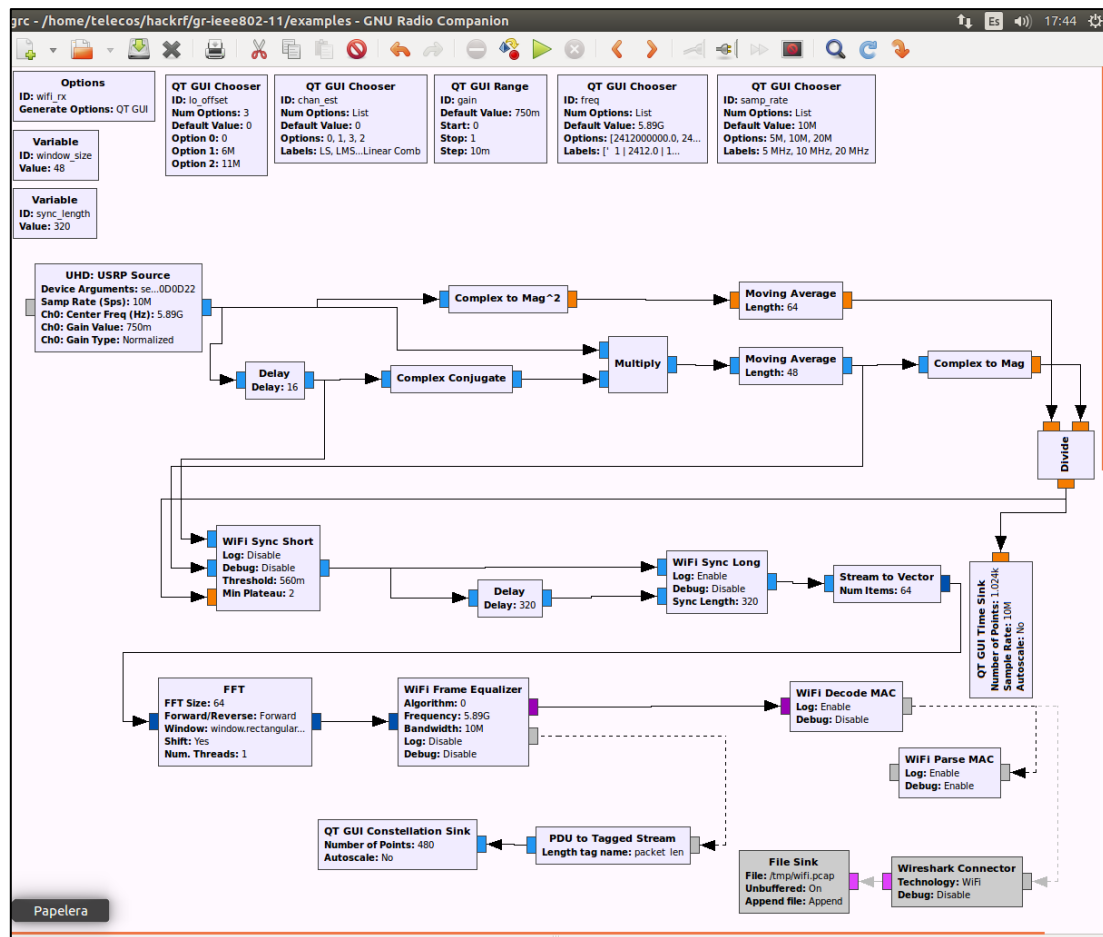


Figura 52. Esquema demodulador de GNUradio

Una vez abierto el programa, en la pantalla principal (ver por ejemplo la Figura 55 ), disponemos de una serie de variables que podemos modificar en función de las necesidades de nuestro sistema:

- **Samp\_rate**: El valor asignado a esta variable es 10 MHz, ya que coincide con el ancho espectral del canal que está siendo utilizado en la comunicación.
- **Lo\_offset**: valor de inicio introducido en el oscilador local. De forma prefijada, este valor se encuentra a 0 y no encontramos necesidad de modificarlo.
- **Gain**: ganancia del sistema. El valor asignado se determinó experimentalmente, hasta lograr una buena recepción y visualización de la modulación final.
- **Chan\_est**: Permite seleccionar uno de los 4 estimadores implementados. En este experimento, se escogió el estimador linear simple, ya que fue el único que nos proporcionó una salida coherente.

El escenario final obtenido se muestra en la Figura 53, donde podemos observar los ordenadores A y B, elementos encargados de enviar y contestar los mensajes pings, respectivamente. Por otro lado, el elemento C es el encargado de captar la comunicación y demodularla. La Figura 54 muestra un esquema simplificado del escenario.



Figura 53. Configuración del experimento para la detección de la modulación

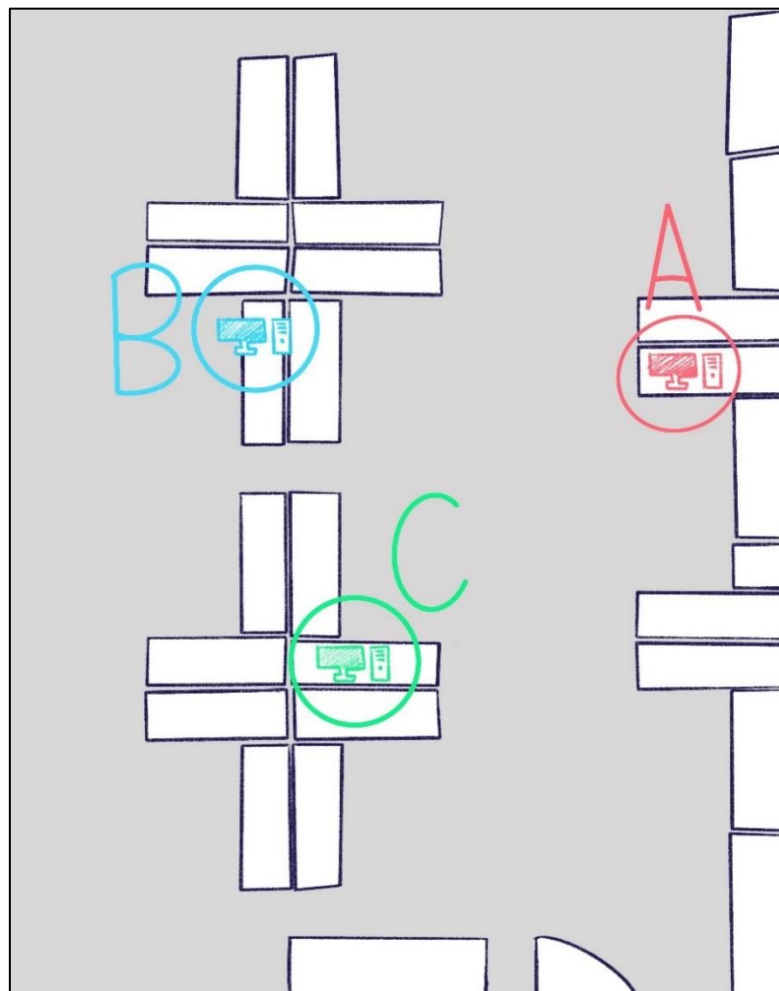


Figura 54. Esquema del experimento realizado

Una vez conectados todos los elementos, empezamos a enviar mensajes *pings* entre los ordenadores A y B, configurados ambos en modo OCB. Inicialmente, la constelación, transmitiendo a la mínima potencia posible, es 16QAM. Esto significa que el canal es lo

suficientemente bueno para poder transmitir con la tasa de transmisión máxima (12 Mbits/segundo). En la Figura 55 podemos ver la modulación resultante.

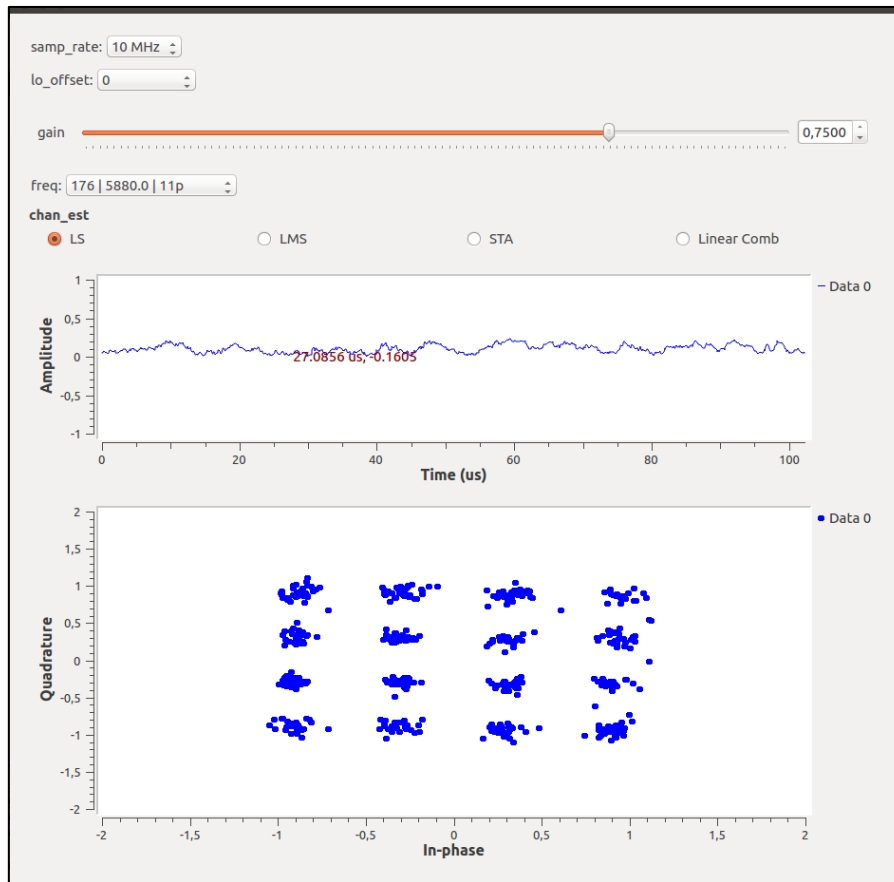


Figura 55. Modulación obtenida con potencia mínima de transmisión

Con el objetivo de comprobar si la constelación se ve afectada por el estado del canal, se utilizaron una serie de atenuadores adaptados a 50  $\Omega$  (Figura 56) para simular una comunicación en un canal con una distancia mayor entre los ordenadores. Estos atenuadores serán colocados a la entrada de la antena en el ordenador receptor B (Figura 57 y Figura 58).



Figura 56. Atenuadores adaptados



*Figura 57. Configuración de los atenuadores*



*Figura 58. Atenuación requerida para empeorar el canal*

Según aumentamos la atenuación se pudo comprobar que la constelación cambiaba de forma automática, por lo que el control de la modulación está a cargo de la propia tarjeta, y ni el driver ni el sistema operativo tienen ningún control sobre el mismo. Para conseguir una modulación BPSK, fue necesario incluir atenuadores hasta un total de 23 dB. En ese caso y en el escenario planteado la modulación se adapta a una BPSK, tal y como se puede ver en la Figura 59. Con esta modulación, la tasa de transmisión es 3 Mbits/segundo, permitiendo una comunicación a mayores distancias, aunque más lenta.



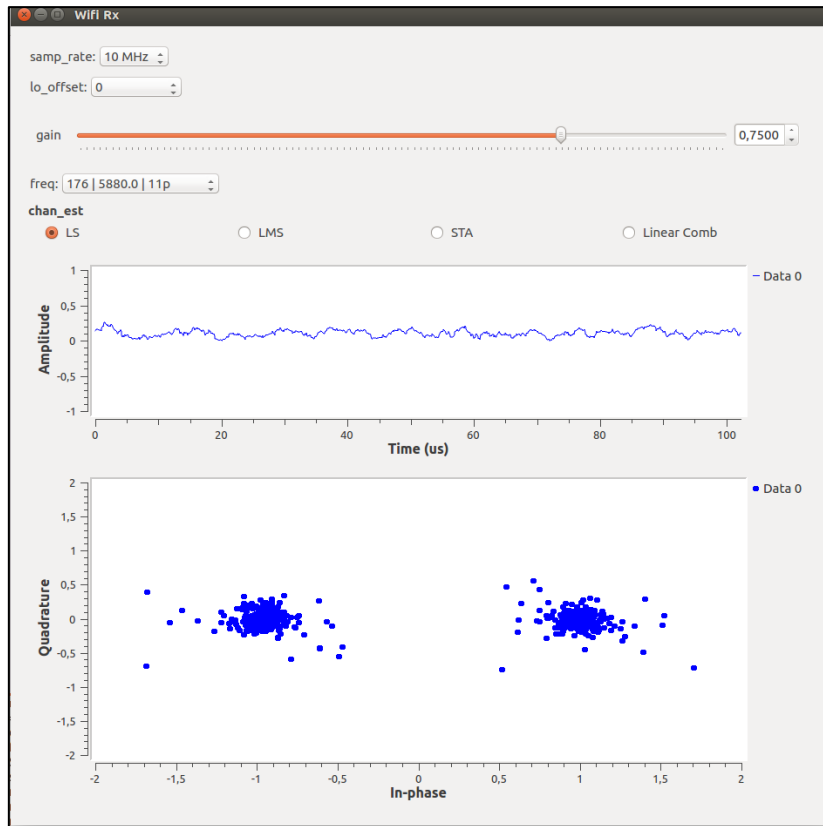


Figura 59. Modulación simulando una distancia grande entre los ordenadores

En un punto intermedio de la atenuación, se logró obtener la modulación QPSK, transmitiendo a 6 Mbits/segundo, tal y como se puede ver en la Figura 60.

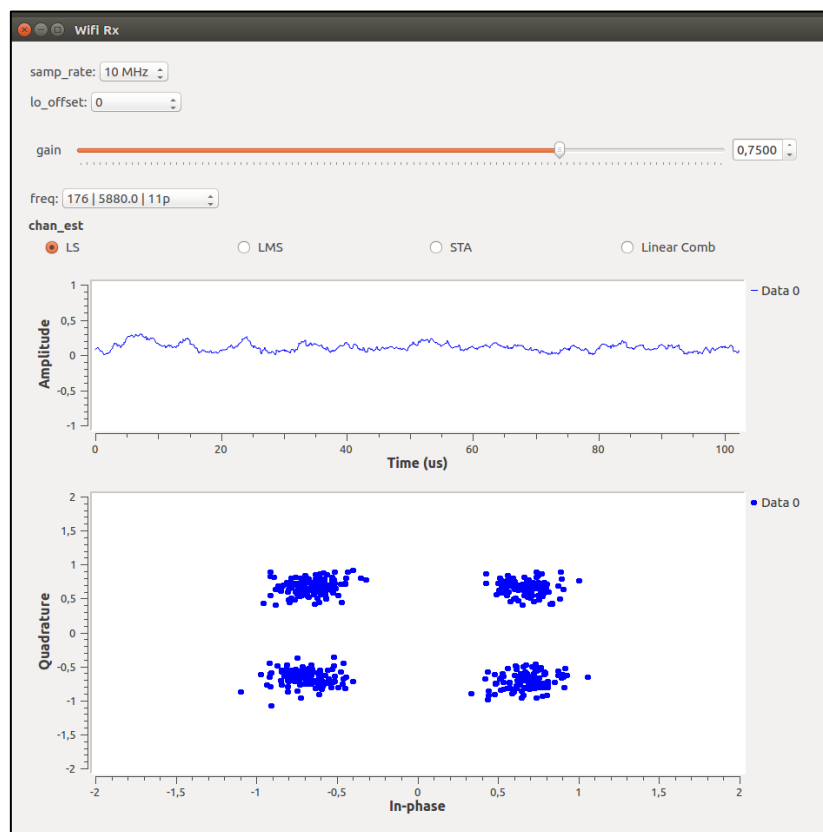


Figura 60. Modulación QPSK

Una vez que sabemos que la modulación es variable en función del estado del canal, es posible relacionar la sensibilidad mínima requerida en el receptor en función de dicha modulación, tal y como se puede ver en la Figura 61. Para una selección correcta de los valores hay que tener en cuenta que el estándar 802.11p utiliza un *coding rate* R de 1/2 y un ancho de banda de 10Mhz.

Modulation	Coding rate (R)	Adjacent channel rejection (dB)	Alternate adjacent channel rejection (dB)	Minimum sensitivity (dBm) (20 MHz channel spacing)	Minimum sensitivity (dBm) (10 MHz channel spacing)	Minimum sensitivity (dBm) (5 MHz channel spacing)
BPSK	1/2	16	32	-82	-85	-88
BPSK	3/4	15	31	-81	-84	-87
QPSK	1/2	13	29	-79	-82	-85
QPSK	3/4	11	27	-77	-80	-83
16-QAM	1/2	8	24	-74	-77	-80
16-QAM	3/4	4	20	-70	-73	-76
64-QAM	2/3	0	16	-66	-69	-72
64-QAM	3/4	-1	15	-65	-68	-71

Figura 61. Relación modulación-sensibilidad del receptor [ 50 ]

En nuestro caso, utilizaremos los siguientes valores para actualizar la fórmula de distancia máxima, como se muestra en la Tabla 3.

Constelación	Bit Rate (Mbits/seg)	Sensibilidad receptor (dBm)
BPSK	3	-85
QPSK	6	-82
16 QAM	12	-77

Tabla 3. Relación entre modulación de la señal y sensibilidad del receptor

### 4.3 Distancia máxima de comunicación teórica

Utilizando las fórmulas vistas en el 4.1 Cálculo teórico de la distancia máxima de comunicación, se ha realizado una hoja de Excel que nos permite automatizar el resultado, realizando modificaciones en cada uno de los elementos de las ecuaciones. Las modificaciones tienen como objetivo distanciarse del entorno perfecto y acercarse a un escenario más parecido a la realidad (reflexiones del suelo, perturbaciones de la carrocería del vehículo, etc.). Estos cambios en los datos se explican a continuación, ya que muchos de ellos se han ido precisando mediante pruebas experimentales en el laboratorio.

Inicialmente, obtuvimos una primera aproximación de la distancia, utilizando unos datos que creíamos que eran los reales. Como las medidas experimentales que se han recogido en la sección Caracterización de los elementos del sistema en laboratorio, muchos de estos datos fueron modificados según los resultados obtenidos en cada experimento.

### 4.3.1 Valores modificados para el cálculo de la máxima distancia de comunicación

#### Estimación de la ganancia de las antenas

Tras el modelado y el estudio del *datasheet* de las diferentes antenas, fuimos capaces de obtener una aproximación de la ganancia real de transmisión y recepción. Esta ganancia ha sufrido una gran modificación en comparación con el valor utilizado al inicio del proyecto. Las modificaciones se muestran en la Tabla 4

Antena	Ganancia según fabricante (dBi)	Ganancia calculada mediante simulación (dBi)
Planas	6	2,3
Cuadradas	6	4,07

Tabla 4. Ganancia según el fabricante frente ganancia calculada por simulación

#### Coefficiente de reflexión de potencia s11

En una primera aproximación, no tuvimos en cuenta este parámetro, por lo que la investigación sobre él ha tenido un impacto bastante grande. Tras el análisis realizado, este parámetro afecta directamente a la potencia transmitida según la siguiente formula:

$$P_{tx} = P_o * (1 - |\rho|^2)$$

Fórmula 3. Potencia acoplada

Donde  $P_{tx}$  es la potencia transmitida acoplada,  $P_o$  es la potencia suministrada por la tarjeta y  $\rho$  se refiere al coeficiente  $s_{11}$  en unidades naturales.

Tras el experimento, los parámetros serán modificados se recogen en la Tabla 5:

Antena	Coefficiente teórico	Coefficiente calculado experimentalmente (dB)	$\rho$
Planas	0	-19,15	0,110280823
Cuadradas	0	-18,08	0,124738351

Tabla 5. Coeficiente de acoplo teórico frente al coeficiente calculado experimentalmente

#### Potencia real transmitida por la tarjeta de red

Originalmente, creíamos que la potencia media suministrada eran 18 dB, dato que coincidía con la información proporcionada por la tarjeta de red. Sin embargo, Tras realizar el experimento Potencia real transmitida por la tarjeta de red, el valor de la potencia final se muestra en la Tabla 6:

Antena	Potencia según fabricante (dB)	Potencia calculada experimentalmente (dB)	Potencia después del acoplo (dBm)
Planas	18	21,02	20,88378807
Cuadradas	18	21,02	20,88378807

Tabla 6. Potencia de la tarjeta según el fabricante frente la potencia experimental

### **Modulación utilizada por la tarjeta**

Por último, este experimento ratificó que la modulación estaba cambiando y no era fija, y por lo tanto la sensibilidad que debería utilizarse debía depender de la modulación que el transmisor utilizará (Tabla 7).

<b>Modulación</b>	<b>Sensibilidad según modulación (dBm)</b>
BPSK	-85
QPSK	-82
16QAM	-77

*Tabla 7. Relación entre modulación y sensibilidad de recepción*

### **4.3.2 Cálculo de la máxima distancia de comunicación**

En primer lugar, al no conocer como operarían cada una de las antenas disponibles, decidimos realizar los cálculos solamente cambiando los valores correspondientes a las antenas, es decir, la ganancia de la antena, las pérdidas introducidas por las conexiones (cableado) y los coeficientes de pérdidas ( $s_{11}$ ). El resultado es mostrado en la Tabla 8.

<b>Tipo antena</b>	<b>Distancia (km)</b>
Plana	1,36084438
WiFi cuadrada	2,038506391
WiFi redonda	1,571534471

*Tabla 8. Distancia máxima teórica sin datos experimentales*

Ante este resultado y tras la prueba explicada en la sección 4.4.4 Comunicación en campo abierto con distancias variables, se decidió descartar todas las antenas y centrar el estudio en las antenas denominadas Antenas WiFi Cuadradas. Esto nos permitió la focalización en otros aspectos relacionados con la comunicación. El estudio que más afecto a la distancia máxima es 4.2.5 Modulación utilizada por la tarjeta, explicado más adelante, donde la sensibilidad del lado receptor varía en función de la constelación utilizada, según la Tabla 9.

<b>Constelación</b>	<b>Bit Rate (Mbits/seg)</b>	<b>Sensibilidad receptor (dBm)</b>
BPSK	3	-85
QPSK	6	-82
16 QAM	12	-77

*Tabla 9. Sensibilidad de recepción frente constelación*

Con esta nueva información y junto con la potencia real de transmisión utilizada, las pérdidas de potencia en acoplamiento, la frecuencia, etc, se modificaron todos los parámetros de la Fórmula 1. Propagación de una onda en espacio libre y la **distancia teórica obtenida** se muestran en la Tabla 10.

Antenas WiFi Cuadradas	
Modulación	Distancia teórica máxima (kms)
BPSK	2,038506391
QPSK	1,443152006
16QAM	0,811544011

Tabla 10. Distancia teórica cond atos experimentales

Según estos cálculos, nuestros coches serían capaces de comunicarse hasta una distancia máxima de 2,0385 kilómetros, distancia que cumpliría ampliamente los requisitos previos acordados. Sin embargo, al tratarse de unos cálculos teóricos que han sufrido muchas simplificaciones y, además, no conocemos a ciencia cierta todos los aspectos y valores que están en juego durante la comunicación, se hace muy difícil cuadrar estos valores con los datos obtenidos experimentalmente, los cuales distan mucho de estos valores. No obstante, estos valores nos permiten conocer ligeramente cual es la distancia en la que estará nuestro sistema real y asegurar que nuestro sistema va a poder realizar una comunicación correcta con el resto de los vehículos cercanos.

En la Figura 62 muestra la distancia máxima en función de los cambios en la modulación y sin ningún otro elemento que pueda afectar directamente en la comunicación (obstáculos, forma de la vía, condiciones ambientales) .

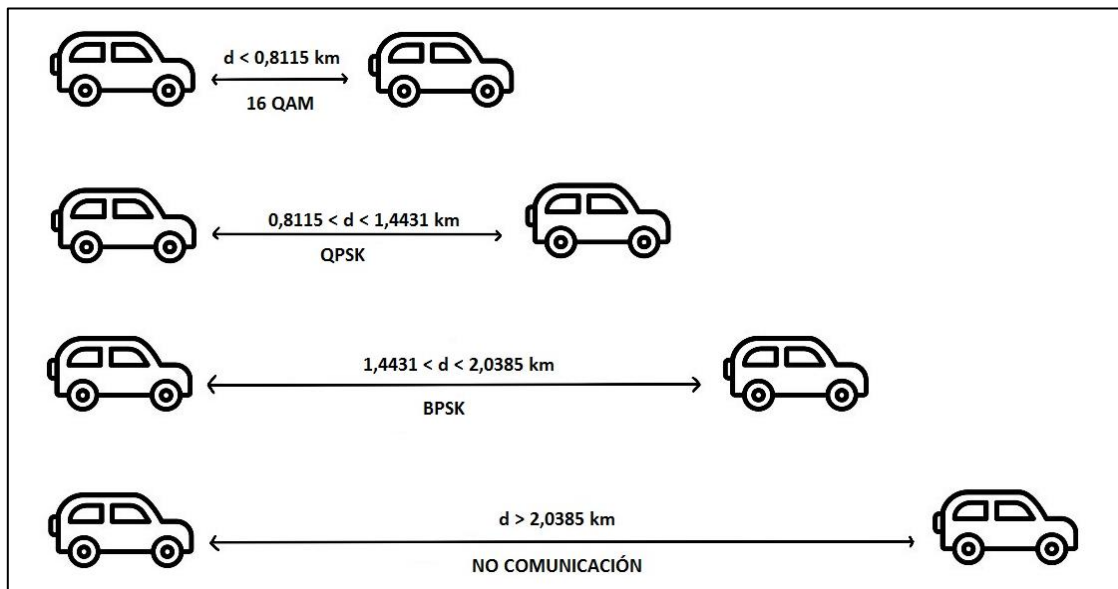


Figura 62. Distancia máxima en función de la modulación

#### 4.4 Estimación de la distancia máxima de comunicación mediante pruebas de campo

En este apartado, se explicarán los experimentos llevados a cabo para determinar en qué condiciones reales es posible la comunicación entre los vehículos y la distancia máxima que podemos esperar de comunicación sin errores.

#### 4.4.1 Experimento interior de 120 metros

El objetivo de esta prueba es conocer si la configuración de la que disponemos es capaz de transmitir una serie de paquetes *CAM* y mensajes *PING* a una distancia de 120 metros con el menor número de fallos posibles. El escenario planteado se caracteriza por ser el más controlado con menor interferencia de posibles factores externos como condiciones medioambientales o posibles obstáculos móviles en el camino.

La configuración utilizada para ello fue la colocación de los dos miniordenadores en los extremos del edificio de Tecnologías de la Información y la Comunicación donde se encuentra la E.T.S.I.T. Los miniordenadores estarán separados una distancia aproximada de 120 metros, como se puede observar en la Figura 63.

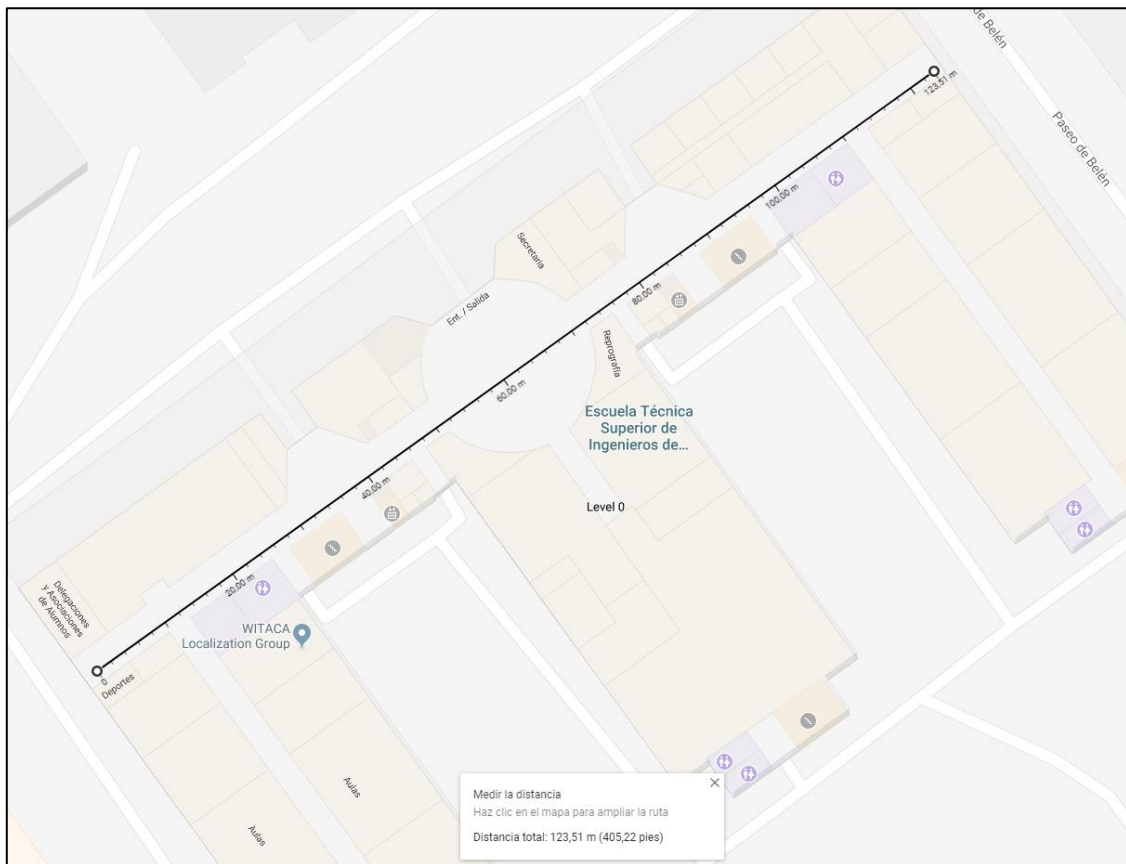
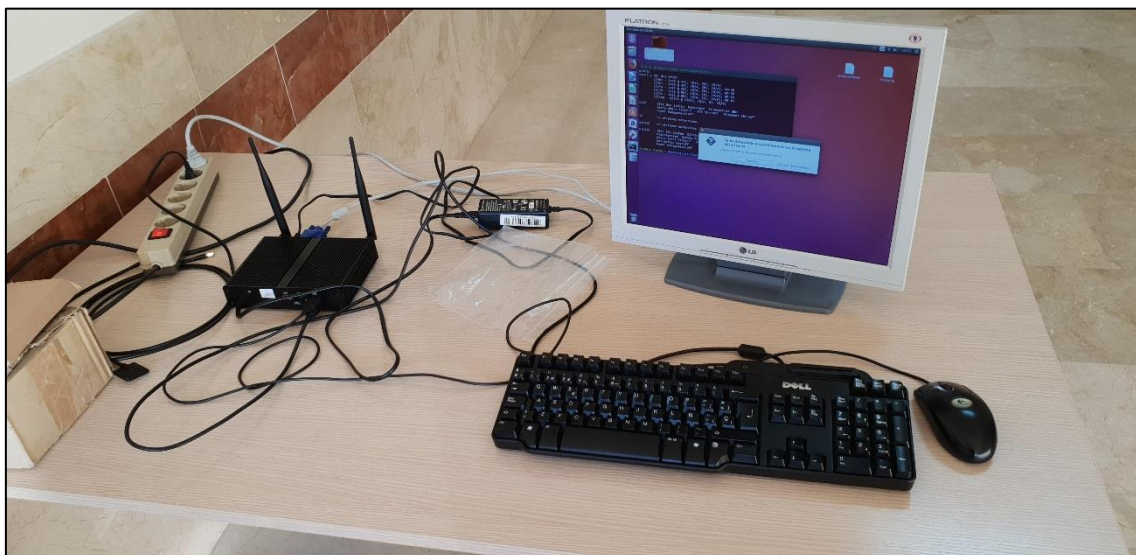


Figura 63. Distancia prueba interior de 120 metros

Como se puede observar (ver Figura 64 y Figura 65), el escenario se caracteriza por existir visión directa entre los dos ordenadores. Utilizaremos las dos antenas, utilizando comunicación MIMO, dado que es el escenario más realista. Además, en este escenario no se esperan obstáculos que impidan la visión directa entre los ordenadores y por lo tanto esperamos una interferencia mínima en la transmisión.



*Figura 64. Escenario de la prueba interior de 120 metros*



*Figura 65. Configuración utilizada para la prueba interior de 120 metros*

### **Resultados utilizando las antenas planas**

El primer experimento consistió en enviar *pings* entre las dos máquinas utilizando las antenas diseñadas para comunicación entre vehículos. Esto nos permitirá conocer el error de la comunicación entre ellas. Se enviaron un total de 1050 pings, de los cuales, todos fueron recibidos correctamente.

```

gco@gco-minipc: ~/Descargas/Scripts
64 bytes from 192.168.1.4: icmp_seq=1032 ttl=64 time=2.01 ms
64 bytes from 192.168.1.4: icmp_seq=1033 ttl=64 time=1.89 ms
64 bytes from 192.168.1.4: icmp_seq=1034 ttl=64 time=3.20 ms
64 bytes from 192.168.1.4: icmp_seq=1035 ttl=64 time=4.15 ms
64 bytes from 192.168.1.4: icmp_seq=1036 ttl=64 time=2.24 ms
64 bytes from 192.168.1.4: icmp_seq=1037 ttl=64 time=3.74 ms
64 bytes from 192.168.1.4: icmp_seq=1038 ttl=64 time=4.80 ms
64 bytes from 192.168.1.4: icmp_seq=1039 ttl=64 time=2.30 ms
64 bytes from 192.168.1.4: icmp_seq=1040 ttl=64 time=2.14 ms
64 bytes from 192.168.1.4: icmp_seq=1041 ttl=64 time=4.27 ms
64 bytes from 192.168.1.4: icmp_seq=1042 ttl=64 time=2.49 ms
64 bytes from 192.168.1.4: icmp_seq=1043 ttl=64 time=2.20 ms
64 bytes from 192.168.1.4: icmp_seq=1044 ttl=64 time=2.58 ms
64 bytes from 192.168.1.4: icmp_seq=1045 ttl=64 time=2.55 ms
64 bytes from 192.168.1.4: icmp_seq=1046 ttl=64 time=3.18 ms
64 bytes from 192.168.1.4: icmp_seq=1047 ttl=64 time=4.29 ms
64 bytes from 192.168.1.4: icmp_seq=1048 ttl=64 time=2.20 ms
64 bytes from 192.168.1.4: icmp_seq=1049 ttl=64 time=3.80 ms
64 bytes from 192.168.1.4: icmp_seq=1050 ttl=64 time=1.99 ms
^C
--- 192.168.1.4 ping statistics ---
1050 packets transmitted, 1050 received, 0% packet loss, time 1050524ms
rtt min/avg/max/mdev = 1.507/2.658/5.309/0.810 ms
gco@gco-minipc:~/Descargas/Scripts$

```

Figura 66. Pings enviados

Para garantizar los resultados obtenidos, se procedió a realizar el mismo escenario, pero, en esta ocasión, enviando mensajes de información y control de OpenC2X (mensajes CAMs) y así obtener la relación de mensajes recibidos y mensajes enviados.

Con estos resultados, podemos calcular una probabilidad de error máxima en dicha comunicación utilizando la Fórmula 4 y la Fórmula 5. [ 51]

$$N_{packet} = \frac{-\ln(1 - CL)}{PER}$$

Fórmula 4. Ecuación de tasa de error sin errores

$$PER = \frac{N_{errores}}{N_{packet}}$$

Fórmula 5. Ecuación de tasa de error con errores

Donde  $N_{packet}$  es el número de paquetes enviados, CL es el nivel de confianza deseado, PER es el error por paquete y  $N_{errores}$  es el número de errores que se han producido en la comunicación.

Utilizando la primera ecuación, porque no se ha producido ningún fallo en la comunicación, y sustituyendo las variables por los siguientes valores:  $N_{packet} = 1050$  y un nivel de confianza  $CL = 0,95$  (95%), obtenemos unas PER de **0,002853078**.<sup>3</sup>

### Resultados utilizando antenas WiFi cuadradas

Al igual que en el caso de las antenas planas, queremos saber si es factible utilizar en el proyecto unas antenas WiFi cuadradas. Para ello, se procedió de la misma manera que en el caso anterior.

Los resultados obtenidos son que los mensajes ping enviados no eran capaces de alcanzar el ordenador de destino. En consecuencia, se decidió acercar los ordenadores para lograr

<sup>3</sup> Las fórmulas originalmente estaban dirigidas a obtener la tasa de error por **BIT**, pero en nuestros experimentos nos importan los **paquetes**, por lo que las siglas utilizadas serán  $N_{packet}$  (Número de paquetes) y PER (*Packet Error Rate*).



una correcta comunicación entre ambos. La distancia máxima a la cual se podía establecer la comunicación fue de 36 metros, como se muestra en la Figura 67.

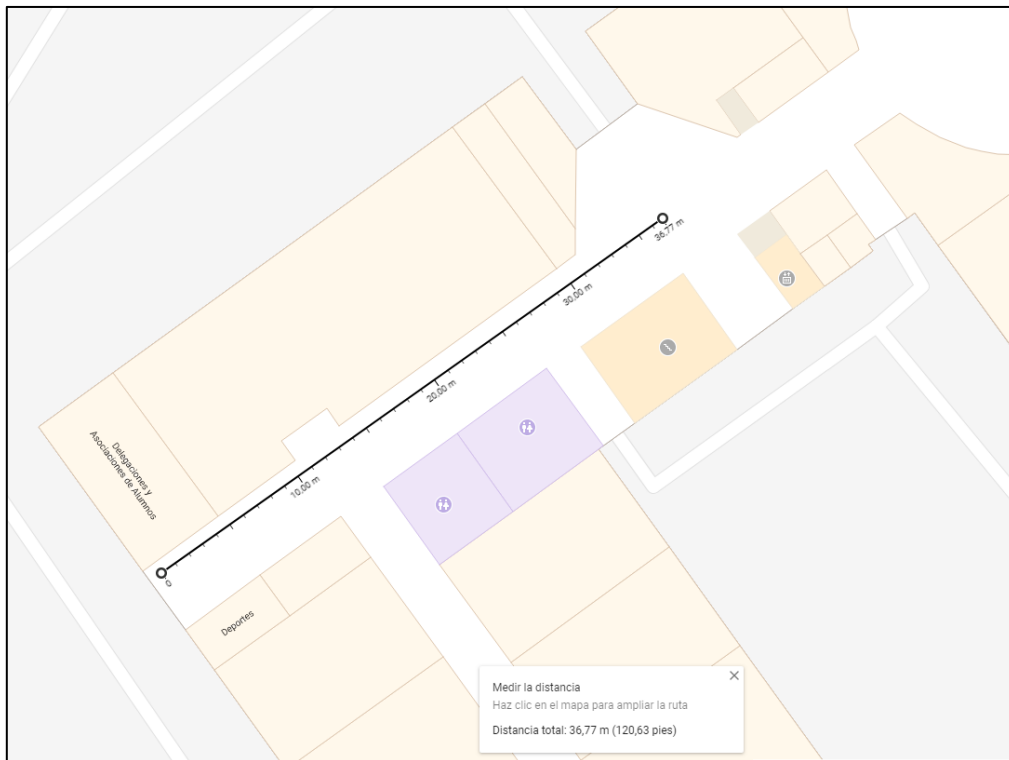


Figura 67. Distancia prueba con antenas WiFi

Tras este experimento, se decidió investigar por qué se alcanzaba tan poca distancia, cuando en los cálculos teóricos se lograban valores similares a las antenas planas. Se descubrió que faltaba un adaptador entre las antenas y la tarjeta de red. Una vez adquiridos estos adaptadores SMA macho-macho se decidió repetir de nuevo este experimento, manteniendo la distancia de la Figura 63, esto es, 120 m. Los mensajes *ping* enviados y recibidos se muestran en la Figura 68, comprobando como se había predicho en el cálculo teórico que para esta distancia estas antenas deberían funcionar al menos igual de bien que las planas.

```
64 bytes from 192.168.1.4: icmp_seq=990 ttl=64 time=1.98 ms
64 bytes from 192.168.1.4: icmp_seq=991 ttl=64 time=3.18 ms
64 bytes from 192.168.1.4: icmp_seq=992 ttl=64 time=0.723 ms
64 bytes from 192.168.1.4: icmp_seq=993 ttl=64 time=4.06 ms
64 bytes from 192.168.1.4: icmp_seq=994 ttl=64 time=2.27 ms
64 bytes from 192.168.1.4: icmp_seq=995 ttl=64 time=1.91 ms
64 bytes from 192.168.1.4: icmp_seq=996 ttl=64 time=2.23 ms
64 bytes from 192.168.1.4: icmp_seq=997 ttl=64 time=2.52 ms
64 bytes from 192.168.1.4: icmp_seq=998 ttl=64 time=1.90 ms
64 bytes from 192.168.1.4: icmp_seq=999 ttl=64 time=1.98 ms
64 bytes from 192.168.1.4: icmp_seq=1000 ttl=64 time=2.72 ms
^C
--- 192.168.1.4 ping statistics ---
1000 packets transmitted, 1000 received, 0% packet loss, time 1000521ms
rtt min/avg/max/mdev = 0.723/2.790/6.978/0.849 ms
gco@gco-mln1pc:~$
```

Figura 68. Resultados obtenidos tras repetir el experimento

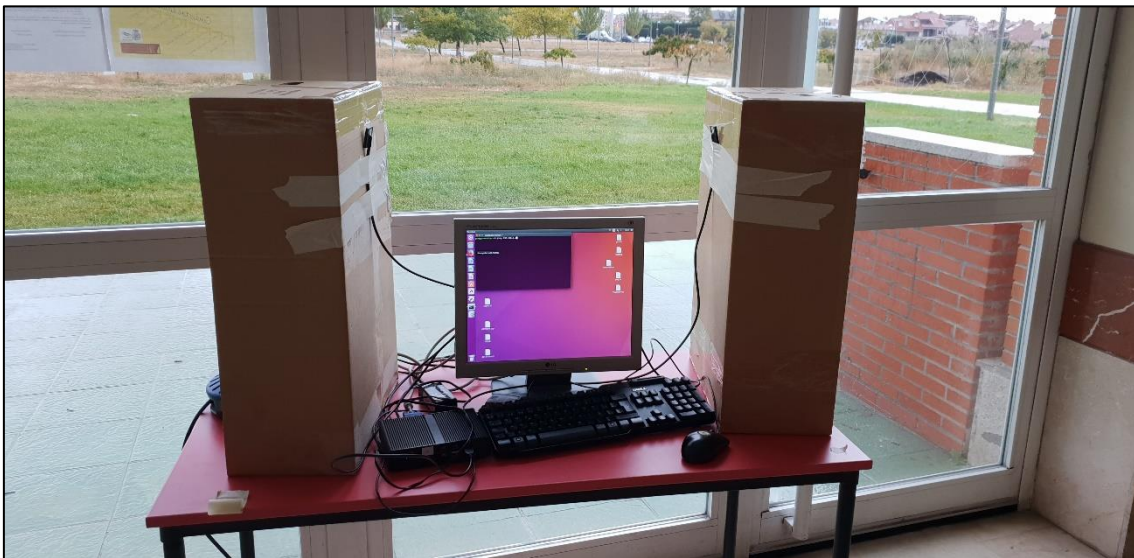
Con esta nueva distancia, se procedió a enviar numerosos mensajes ping para obtener la probabilidad de error de dicha comunicación. Los valores obtenidos se calcularon utilizando las fórmulas anteriormente descritas. Para este caso, la PER es **0,002995732**.

## **Conclusiones del experimento**

En vista de los resultados anteriores, ambas antenas tienen una probabilidad de error baja, pero la distancia máxima de funcionamiento es un aspecto muy importante, ya que, según la Dirección General de Tráfico, la distancia de seguridad circulando a una velocidad máxima de 120 km/h tiene que ser de 100 metros como mínimo [ 52 ]. Por esta razón, a priori, parecen mucho más prometedoras las antenas planas que las antenas de WiFi comunes. Sin embargo, tras repetir la prueba en el caso de las antenas WiFi, no podemos descartar ninguna de las dos antenas porque ambas han funcionado correctamente a esa distancia.

### **4.4.2 Comunicación en campo abierto con distancia fija**

La finalidad de esta prueba es conocer si las antenas son capaces de transmitir a una distancia mayor que la anterior y en un entorno menos controlado, o al menos con un mayor número de obstáculos. Para realizar la prueba, se colocó uno de los ordenadores dentro de un vehículo en el exterior del edificio de la escuela (latitud 41,6601273 y longitud -4,7049513), y el otro ordenador se instaló en una mesa dentro del edificio (coordenadas 41,6620771, -4,7055217). La disposición concreta de los equipos se puede ver en la Figura 69, Figura 70 y Figura 71.



*Figura 69. Configuración del ordenador interior*



*Figura 70. Ordenador y GPS del vehículo*



*Figura 71. Disposición de las antenas planas dentro del vehículo*

Las antenas planas están instaladas de tal manera que el plano XY de la antena sea horizontal al suelo para aprovechar al máximo la mayor ganancia de transmisión de la antena en ese plano, tal y como se advirtió al principio del capítulo. [ 40 ]

El vehículo se estacionó intentando asegurar la visión directa entre emisor y receptor, a una distancia de 222,9 metros, tal y como se ve en la Figura 72 y Figura 73.



*Figura 72. Vehículo en visión directa con el receptor*

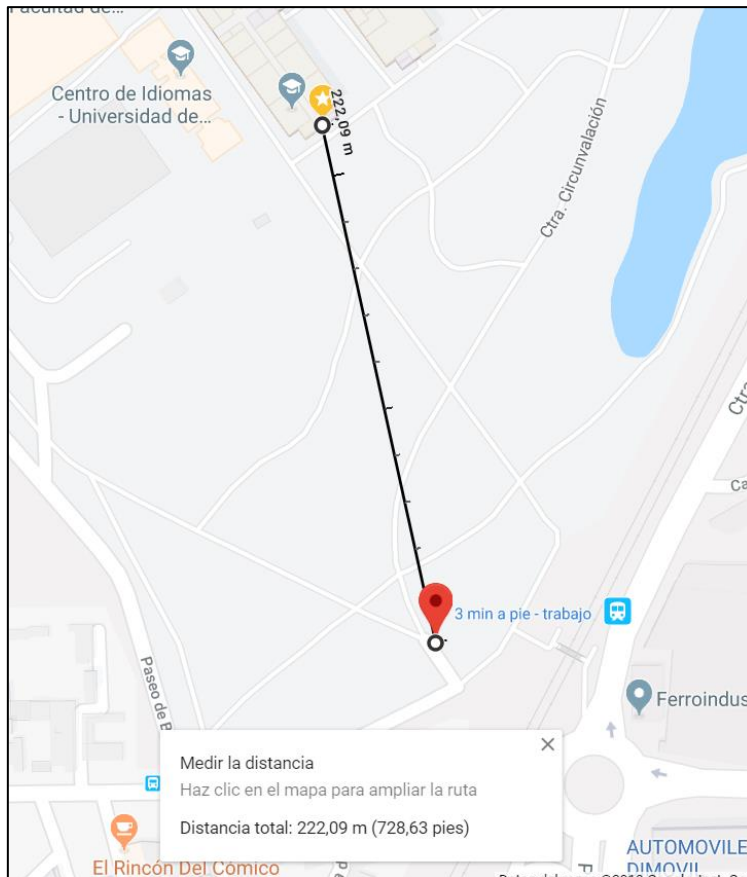


Figura 73. Plano de la posición de los transmisores y distancia total de la prueba

Una vez configurados los dos equipos y al igual que en la prueba anterior, se procedió a enviar números mensajes *pings* para comprobar si la comunicación era posible y obtener una tasa de error de paquete. El resultado obtenido se puede ver en la Figura 74.

```

From 192.168.1.2 icmp_seq=114 Destination Host Unreachable
From 192.168.1.2 icmp_seq=115 Destination Host Unreachable
From 192.168.1.2 icmp_seq=116 Destination Host Unreachable
From 192.168.1.2 icmp_seq=117 Destination Host Unreachable
From 192.168.1.2 icmp_seq=118 Destination Host Unreachable
From 192.168.1.2 icmp_seq=119 Destination Host Unreachable
From 192.168.1.2 icmp_seq=120 Destination Host Unreachable
From 192.168.1.2 icmp_seq=121 Destination Host Unreachable
From 192.168.1.2 icmp_seq=122 Destination Host Unreachable
From 192.168.1.2 icmp_seq=123 Destination Host Unreachable
From 192.168.1.2 icmp_seq=124 Destination Host Unreachable
From 192.168.1.2 icmp_seq=125 Destination Host Unreachable
From 192.168.1.2 icmp_seq=126 Destination Host Unreachable
From 192.168.1.2 icmp_seq=127 Destination Host Unreachable
From 192.168.1.2 icmp_seq=128 Destination Host Unreachable
From 192.168.1.2 icmp_seq=129 Destination Host Unreachable
From 192.168.1.2 icmp_seq=130 Destination Host Unreachable
From 192.168.1.2 icmp_seq=131 Destination Host Unreachable
From 192.168.1.2 icmp_seq=132 Destination Host Unreachable
From 192.168.1.2 icmp_seq=133 Destination Host Unreachable
From 192.168.1.2 icmp_seq=134 Destination Host Unreachable
From 192.168.1.2 icmp_seq=135 Destination Host Unreachable
From 192.168.1.2 icmp_seq=136 Destination Host Unreachable
  
```

Figura 74. Resultado de pings con comunicación no establecida

En este experimento los ordenadores fueron incapaces de establecer comunicación. En el momento de ver los resultados del experimento se conjeturaron tres posibles razones para que no funcionara:

- El alcance de estas antenas se encuentra en un punto intermedio entre la distancia del primer experimento y esta nueva distancia.
- La prueba se realizó un día lluvioso, que afecta en gran medida a las comunicaciones a esta frecuencia. Es conocido la gran atenuación que provoca la lluvia en las comunicaciones por radiofrecuencia. Dado que ese día hubo lluvia intensa durante el experimento, ésta pudo influir en el resultado de la prueba.
- La carrocería del vehículo afecta gravemente las señales que estamos transmitiendo.

También se probó a realizar dicha prueba con las antenas WiFi, las cuales conseguían lograr comunicación, pero la tasa de pérdidas excedía con mucho el 50% de los paquetes.



*Figura 75. Disposición del ordenador en el vehículo*

#### **4.4.3 Estudio de la variación de la posición estimada por el GPS en punto fijo**

En el apartado 4.4.2 Comunicación en campo abierto con distancia fija se explicó la disposición del experimento para comprobar la comunicación en campo abierto y con distancia fija. Como ya se comentó el resultado en cuanto a la comunicación no fue el deseado, razón por la que se tuvo que hacer otro experimento, pero se aprovechó la disposición de este experimento para estudiar como de exacta resultaba la posición GPS cuando el receptor no se mueve. Utilizamos los datos recibidos por el GPS B, colocado en el interior del vehículo y posteriormente, procesamos las muestras recibidas para ver cómo cambia la posición, permitiéndonos estimar como varía el error del GPS a lo largo del tiempo y dar un valor medio.

En primer lugar, calculamos la distancia recibida entre cada muestra y la posición real conocida. La posición real se estimó mediante la orografía donde estacionamos el vehículo y el sistema de mapas de Google. Este proceso fue sencillo porque el vehículo fue estacionado al final de un paseo delimitado por unos bolardos, fácilmente detectables



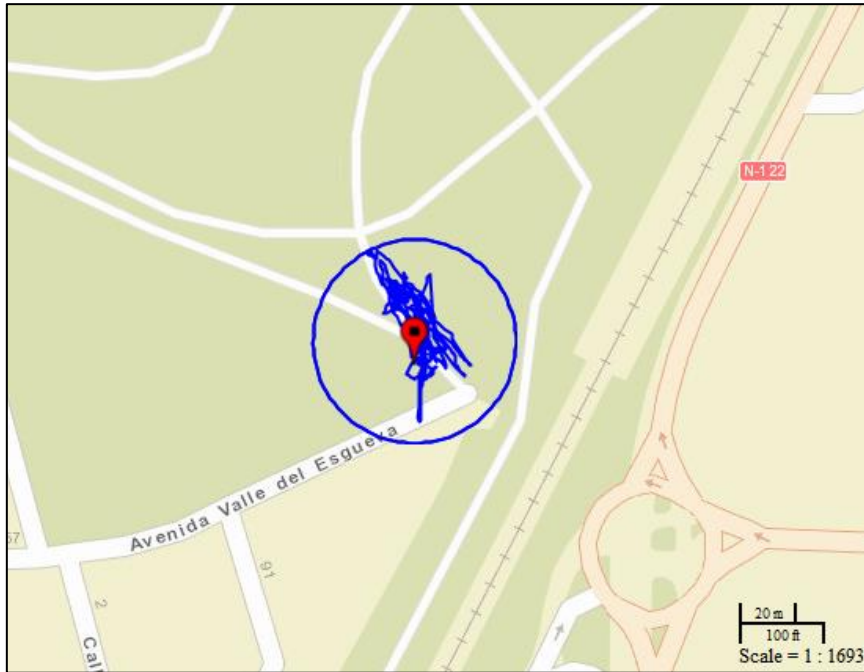


Figura 78. Error del GPS sobre mapa

Como el día en que se realizó el experimento fue un día muy nublado y hubo una gran cantidad de lluvia, los datos obtenidos se vieron gravemente perjudicados por culpa de las condiciones atmosféricas. Se han calculado los datos estadísticos de las muestras recibidas y obtenemos que el error medio es de 7,90 metros, el error máximo fue de 28,85 metros con una varianza de 22,34. Por otro lado, los percentiles obtenidos se pueden apreciar en la Figura 79. Consideramos que el máximo es aceptable, ya que se sobrepasa levemente la precisión indicada por el fabricante y esto se puede deber a las condiciones ambientales desfavorables. [ 54 ]

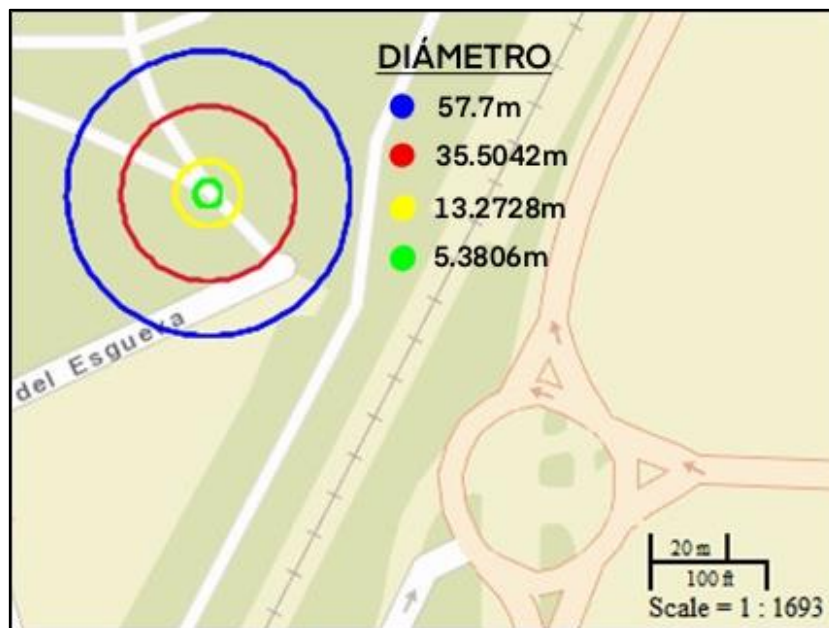


Figura 79. Error máximo en el posicionamiento GPS por percentiles. En verde el 5%, amarillo el 50%, rojo el 95% y azul el 100%

El círculo verde contiene el 5% de las muestras obtenidas, el círculo rojo representa la mediana de las distancias alcanzadas (percentil 50%) y el círculo rojo contiene el 95% de las muestras.

Para concretar más en detalle la precisión de nuestros sistemas GPS, se decidió repetir parte de este experimento. Se colocaron los dos ordenadores juntos con sus respectivos dispositivos GPS dentro del mismo vehículo. En la Figura 80 y en la Figura 81 podemos observar la configuración utilizada.



*Figura 80. Configuración del experimento (vista exterior)*



*Figura 81. Configuración del experimento (vista interior)*

La posición escogida será la misma que en el caso anterior, ya que se trata de un espacio abierto y libre de obstáculos que puedan interferir con las señales recibidas (sin edificios, ni árboles, ni torres de comunicación o eléctricas) y el día fue soleado con unas condiciones atmosféricas totalmente favorables. El error en la distancia obtenido se puede comprobar en la Figura 82 y los resultados estadísticos en la Tabla 11.



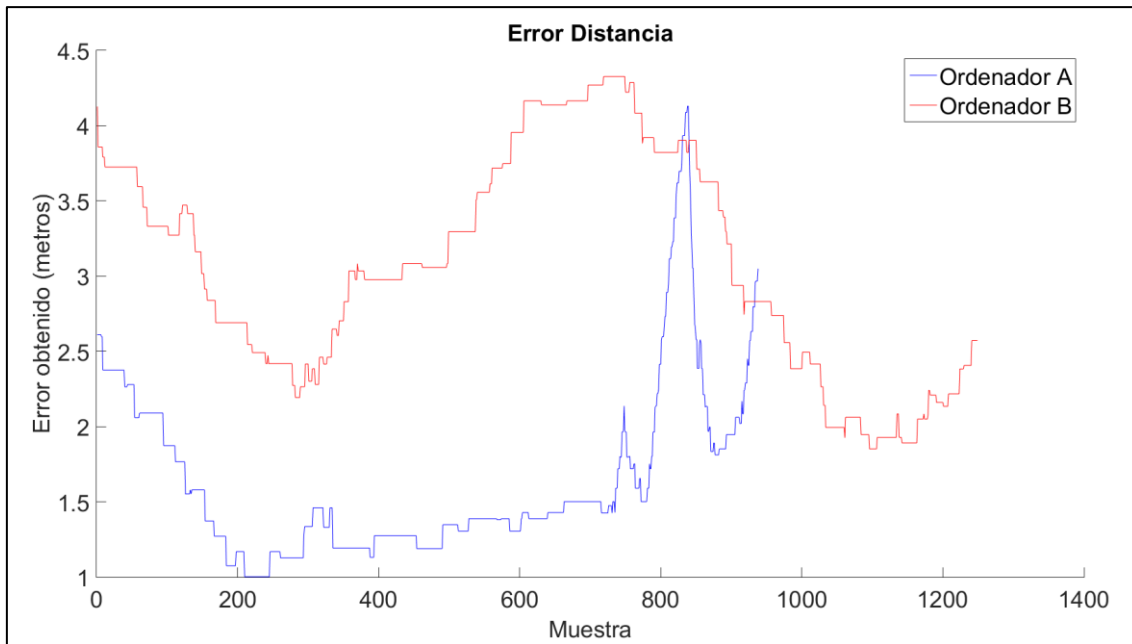


Figura 82. Error en la distancia GPS

Ordenador	Percentil 5 %	Percentil 50 %	Percentil 95 %	Máximo
A	1,0747 metros	1,4269 metros	2,7946 metros	4,1304 metros
B	1,9284 metros	3,0350 metros	4,2231 metros	4,3269 metros

Tabla 11. Resultados estadísticos GPS

Como se puede apreciar, los resultados obtenidos por cada GPS son diferentes. Esto se debe a que, aunque el sistema comparte gran parte del sistema global, cada receptor GPS es diferente entre sí y no reciben la misma señal cada uno de ellos. Ahí es donde se producen estas diferencias.

A la vista de estos resultados, podemos asegurar que nuestro sistema se ve muy afectado por los efectos ambientales y elementos externos a la comunicación. Sin embargo, los valores obtenidos en condiciones atmosféricas favorables son muy aceptables para nuestro proyecto, pero no son suficientes si deseamos determinar cuál es el carril que está utilizando un vehículo en concreto.

#### 4.4.4 Comunicación en campo abierto con distancias variables

Con el objetivo de determinar con mayor precisión la distancia máxima de operación de las diferentes antenas, se planteó una tercera prueba en el Paseo Juan Carlos I. Las condiciones atmosféricas del día fueron muy favorables, ya que se realizó un día muy soleado con alguna nube ocasional. Esta prueba consistió en dos coches separados una determinada distancia y colocados en el mismo sentido, como se muestra en la Figura 83



Figura 83. Vehículos utilizados en la prueba colocados según la disposición que tendrán cambiando la distancia entre ellos

El método seguido es el siguiente: se separan los coches una distancia determinada, la cual es medida con un telémetro láser (Figura 84). Se enviaban pings de uno de los vehículos a otro e, *in situ*, y se comprobaba si al menos había recepción de mensajes. Esta comprobación serviría para calcular la distancia máxima de comunicación y la tasa de pérdidas de mensajes que se producía.



Figura 84. Láser de medición utilizado

Ambos ordenadores están configurados de manera similar en los dos vehículos. En caso de las antenas planas, el ordenador se situó en los asientos traseros y las antenas enganchadas en las ventanillas posteriores (Figura 85 y Figura 86), mientras que en el caso de las antenas WiFi convencionales, el ordenador se situó en la zona central del salpicadero en el caso del coche trasero y en la bandeja del maletero en el caso del coche situado delante .



*Figura 85. Configuración de las antenas planas*



*Figura 86. Posición del ordenador y GPS en la parte trasera del vehículo*

Con las antenas planas, se alcanzaron 31 metros antes de que los mensajes se empezaran a perder durante la comunicación. Por otro lado, utilizando la configuración de antenas WiFi, se lograron alcanzar en torno a 314 metros (329 metros según GoogleMaps) como se muestra en la Figura 87.

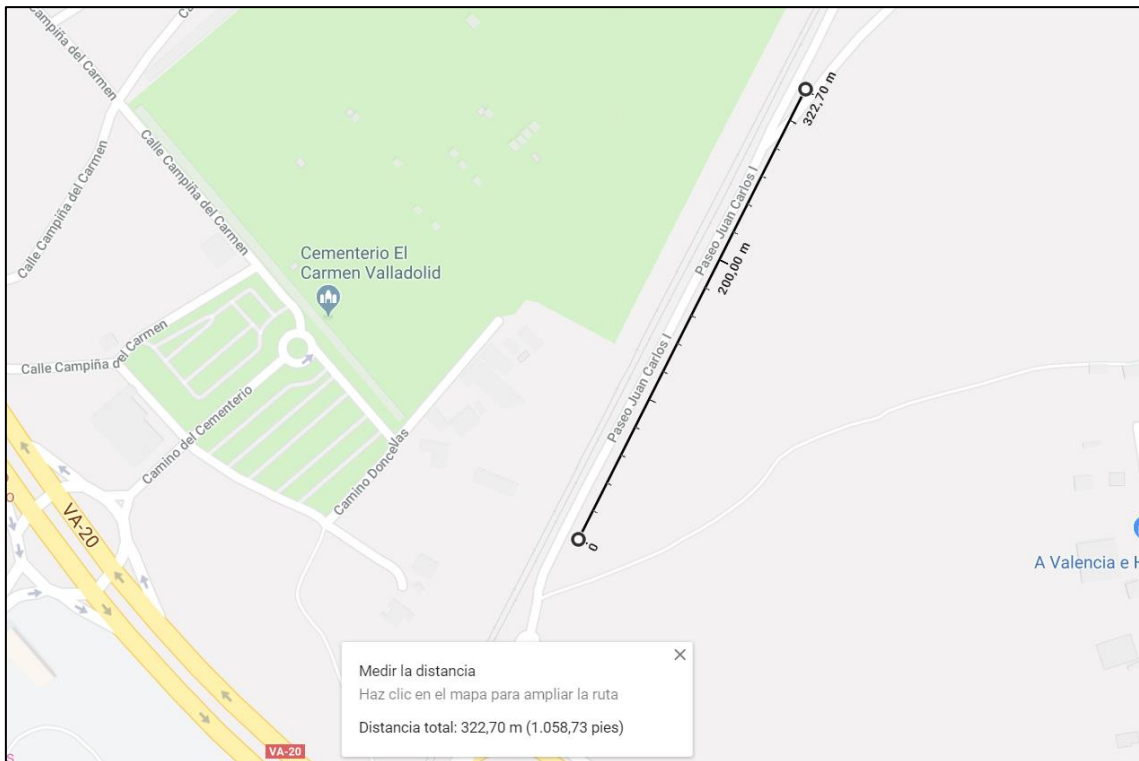


Figura 87. Distancia máxima de comunicación alcanzada experimentalmente en la prueba de campo

Una vez obtenida la distancia máxima de ambas configuraciones, se procedió a enviar un elevado número de paquetes pings para determinar el *Package Error Rate* (PER) de cada comunicación.

```
64 bytes from 192.168.1.4: icmp_seq=134 ttl=64 time=4.10 ms
64 bytes from 192.168.1.4: icmp_seq=135 ttl=64 time=10.2 ms
64 bytes from 192.168.1.4: icmp_seq=136 ttl=64 time=23.1 ms
64 bytes from 192.168.1.4: icmp_seq=139 ttl=64 time=12.5 ms
64 bytes from 192.168.1.4: icmp_seq=140 ttl=64 time=3.57 ms
64 bytes from 192.168.1.4: icmp_seq=142 ttl=64 time=3.04 ms
64 bytes from 192.168.1.4: icmp_seq=145 ttl=64 time=9.94 ms
64 bytes from 192.168.1.4: icmp_seq=146 ttl=64 time=2.92 ms
64 bytes from 192.168.1.4: icmp_seq=147 ttl=64 time=4.26 ms
64 bytes from 192.168.1.4: icmp_seq=148 ttl=64 time=6.30 ms
64 bytes from 192.168.1.4: icmp_seq=149 ttl=64 time=12.3 ms
64 bytes from 192.168.1.4: icmp_seq=150 ttl=64 time=2.43 ms
64 bytes from 192.168.1.4: icmp_seq=151 ttl=64 time=9.83 ms
64 bytes from 192.168.1.4: icmp_seq=152 ttl=64 time=21.4 ms
64 bytes from 192.168.1.4: icmp_seq=157 ttl=64 time=8.62 ms
64 bytes from 192.168.1.4: icmp_seq=158 ttl=64 time=2.37 ms
64 bytes from 192.168.1.4: icmp_seq=159 ttl=64 time=4.41 ms
64 bytes from 192.168.1.4: icmp_seq=161 ttl=64 time=5.04 ms
64 bytes from 192.168.1.4: icmp_seq=162 ttl=64 time=2.43 ms
^C
--- 192.168.1.4 ping statistics ---
162 packets transmitted, 94 received, 41% packet loss, time 161575ms
rtt min/avg/max/mdev = 2.091/9.184/51.580/11.928 ms
gco@gco-minipc:~$
```

Figura 88. Pings enviados con antenas planas

```
64 bytes from 192.168.1.4: icmp_seq=57 ttl=64 time=24.2 ms
64 bytes from 192.168.1.4: icmp_seq=58 ttl=64 time=2.21 ms
64 bytes from 192.168.1.4: icmp_seq=59 ttl=64 time=2.74 ms
64 bytes from 192.168.1.4: icmp_seq=60 ttl=64 time=8.37 ms
64 bytes from 192.168.1.4: icmp_seq=61 ttl=64 time=2.55 ms
64 bytes from 192.168.1.4: icmp_seq=62 ttl=64 time=3.16 ms
64 bytes from 192.168.1.4: icmp_seq=63 ttl=64 time=4.81 ms
64 bytes from 192.168.1.4: icmp_seq=64 ttl=64 time=5.18 ms
64 bytes from 192.168.1.4: icmp_seq=65 ttl=64 time=6.93 ms
64 bytes from 192.168.1.4: icmp_seq=66 ttl=64 time=3.14 ms
64 bytes from 192.168.1.4: icmp_seq=67 ttl=64 time=2.58 ms
64 bytes from 192.168.1.4: icmp_seq=68 ttl=64 time=3.31 ms
64 bytes from 192.168.1.4: icmp_seq=69 ttl=64 time=3.14 ms
64 bytes from 192.168.1.4: icmp_seq=70 ttl=64 time=2.88 ms
64 bytes from 192.168.1.4: icmp_seq=72 ttl=64 time=5.68 ms
64 bytes from 192.168.1.4: icmp_seq=74 ttl=64 time=54.5 ms
64 bytes from 192.168.1.4: icmp_seq=76 ttl=64 time=13.3 ms
64 bytes from 192.168.1.4: icmp_seq=77 ttl=64 time=2.53 ms
64 bytes from 192.168.1.4: icmp_seq=79 ttl=64 time=2.73 ms
64 bytes from 192.168.1.4: icmp_seq=81 ttl=64 time=7.08 ms
64 bytes from 192.168.1.4: icmp_seq=82 ttl=64 time=4.30 ms
64 bytes from 192.168.1.4: icmp_seq=83 ttl=64 time=19.6 ms
64 bytes from 192.168.1.4: icmp_seq=85 ttl=64 time=13.2 ms
64 bytes from 192.168.1.4: icmp_seq=87 ttl=64 time=30.9 ms
64 bytes from 192.168.1.4: icmp_seq=88 ttl=64 time=6.04 ms
64 bytes from 192.168.1.4: icmp_seq=89 ttl=64 time=27.3 ms
64 bytes from 192.168.1.4: icmp_seq=90 ttl=64 time=2.87 ms
64 bytes from 192.168.1.4: icmp_seq=94 ttl=64 time=3.01 ms
64 bytes from 192.168.1.4: icmp_seq=95 ttl=64 time=2.38 ms
64 bytes from 192.168.1.4: icmp_seq=96 ttl=64 time=3.34 ms
64 bytes from 192.168.1.4: icmp_seq=97 ttl=64 time=8.27 ms
64 bytes from 192.168.1.4: icmp_seq=98 ttl=64 time=2.74 ms
64 bytes from 192.168.1.4: icmp_seq=99 ttl=64 time=3.42 ms
64 bytes from 192.168.1.4: icmp_seq=101 ttl=64 time=9.23 ms
^C
--- 192.168.1.4 ping statistics ---
101 packets transmitted, 87 received, 13% packet loss, time 100197ms
rtt min/avg/max/mdev = 2.162/6.995/54.585/9.465 ms
qco@qco-minipc:~$ ping -I ocb0 192.168.1.4
```

Figura 89. Pings enviados con antenas WiFi

El PER obtenido por cada antena a la distancia máxima de comunicación es:

- Antenas Planas: PER = 0,4125 enviando 160 mensajes a una distancia de 31 metros. De los mensajes, solo 94 llegaron correctamente.
- Antenas Wifi cuadradas: PER = 0,138613861 enviando 101 mensajes a una distancia de 314 metros. De los mensajes, 87 llegaron correctamente.

#### 4.5 Discusión de resultados

Los resultados obtenidos parecen sugerir que la carrocería de los vehículos tiene un impacto importante que afecta en gran medida a la comunicación que queremos realizar. La carrocería no ha sido considerada en los cálculos que hemos realizado para obtener la distancia máxima, por lo que aquellos cálculos no son sino una burda aproximación. Es evidente que las antenas inter-vehiculares planas estén probablemente diseñadas para ser colocadas en el exterior del vehículo, fuera ya de todos los paramentos, lo cual mejoraría en mucho su rendimiento, pero que en nuestro caso no nos lo podemos permitir, por seguridad y por no disponer de cables de comunicación suficientemente largos (que en cualquier caso añadirían más pérdidas).

Tras esta serie de experimentos se decidió utilizar únicamente las **antenas WiFi cuadradas** para el resto de los experimentos que se llevaron a cabo, ya que nos permiten alcanzar una distancia mayor de comunicación. La razón para que esto ocurra así puede deberse a que, primero, no soportan pérdidas debido a los cables de comunicación que son necesarios en las otras antenas, segundo, se pueden situar directamente en el parabrisas delantero o en la luna trasera, con lo que la interferencia con la carrocería del automóvil parece ser menor.

Por otro lado, la distancia máxima obtenida teóricamente sigue siendo muy superior a la obtenida empíricamente, incluso tras caracterizar y concretar de una manera más precisa

los elementos de los ordenadores (modulación utilizada, sensibilidad de la tarjeta de red, potencia de transmisión, etc.). Esto puede deberse a los problemas con la carrocería de los que hemos hablado antes, aparte de que el estudio teórico se ha realizado con los supuestos más sencillos.

Finalmente, es interesante apuntar que, durante el experimento, un tercer vehículo se colocó entre nuestros dos vehículos, limitando la visión directa entre ellos. El resultado fue la pérdida total de la comunicación. Este suceso nos permitió concretar que el canal dominante es el directo y los otros canales obtenidos por utilizar tecnología MIMO son muy débiles para seguir manteniendo una comunicación correcta. Esto podría indicar que en una situación real en la que haya muchos automóviles en una carretera, solo se recibiría información de los automóviles adyacentes. Más pruebas se deberían hacer a este respecto, aunque habría que hacerlas con la disposición real que las antenas tendrían en el automóvil para eliminar cualquier otro impedimento en la comunicación.

La Tabla 12 recoge un resumen con los datos obtenidos en los experimentos realizados.

Nombre	Descripción	Tipo de antenas	Distancia máxima (metros)	Numero de paquetes enviados	Numero de paquetes correctos	Numero de paquetes incorrectos	Nivel de confianza	PER
Experimento interior de 120 metros	Realizada en el pasillo de la escuela utilizando los dos tipos de antenas disponibles (distancia 130 metros)	Planas	<b>130</b>	1050	1050	0	0,95	<b>0,002853078</b>
		Wifi Cuadradas	<b>130</b>	1000	1000	0	0,95	<b>0,002995732</b>
Comunicación en campo abierto con distancia fija	Realizada entre uno de los extremos de la universidad y un coche aparcado a 200 metros (aproximadamente)	Planas	<b>220</b>	150	0	150	-	<b>1</b>
		Wifi Cuadradas	<b>220</b>	150	0	150	-	<b>1</b>
Comunicación en campo abierto con distancias variables	Realizada entre dos coches, con el objetivo de determinar la distancia máxima de funcionamiento	Planas	<b>31</b>	160	94	66	-	<b>0,4125</b>
		Wifi Cuadradas	<b>314</b>	101	87	14	-	<b>0,138613861</b>

*Tabla 12. Resumen de las pruebas de campo realizadas*





# 5. Demostración del sistema:

## caso de uso

Una vez que ya se encuentra el sistema caracterizado y totalmente funcional, el último paso a realizar es un experimento simulando un entorno realista y así comprobar como estaría integrado nuestro sistema de comunicación en un vehículo final. Finalmente, se procesarán los datos obtenidos durante el experimento y daremos una conclusión del sistema en su cómputo global.

### 5.1 Realización del experimento

El primer proceso realizado fue la planificación y organización de la prueba, es decir, qué se quería conseguir (objetivos), qué elementos eran necesarios, qué ruta debíamos seguir con los vehículos, escoger un día con condiciones atmosféricas favorables (evitar días lluviosos) y finalmente coordinación con varias personas para poder realizar la prueba de una manera segura.

El objetivo de este experimento es recoger datos de los 3 dispositivos externos (GPS, Gamepad y OBD) que posteriormente fueron utilizados para obtener información del funcionamiento del sistema al completo. Entre los datos obtenidos se encuentran: qué datos han sido recibidos correctamente, qué elementos utilizados son más propensos a fallos y conocer la tasa de pérdida de paquetes entre los vehículos en función de la distancia.

Por otro lado, se estableció una ruta con las siguientes propiedades:

- Debía ser una ruta relativamente sencilla que permitiera a dos automóviles seguirse y estar la mayor parte del tiempo en visión directa.
- Debía ser una ruta en la que no hubiera obstáculos para la comunicación, como edificios, arboleda, etc.
- Debía ser relativamente larga para que los resultados obtenidos fueran fiables.
- La ruta debía permitir encontrar puntos de referencia que nos pudiera permitir dividir el conjunto de datos recuperados en varios bloques si esto fuera necesario.

La ruta se realizó por la carretera de circunvalación VA-20 de Valladolid, como se muestra en la Figura 90. Esta ruta cumplía todas las condiciones impuestas. Además, la ruta se dividió en numerosos tramos, tal y como se ve en la misma figura, para tener tramos relativamente cortos y poder “reiniciar” así los resultados y no acumular errores del GPS o del odómetro. Aunque el experimento se diseñó para este fin, finalmente, estos tramos no fueron utilizados ya que los mandos gamepad se desconectaron al inicio de la prueba y no captaron información deseada durante la misma.

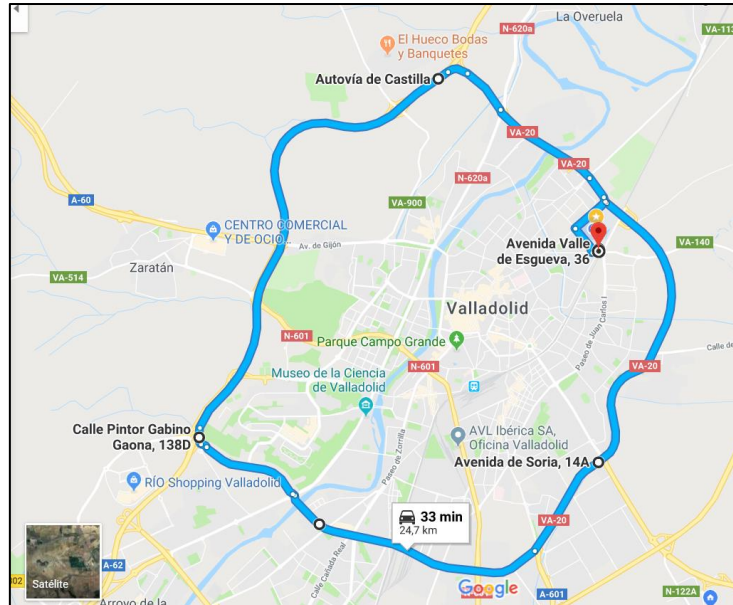


Figura 90. Ruta diseñada para el caso de uso.

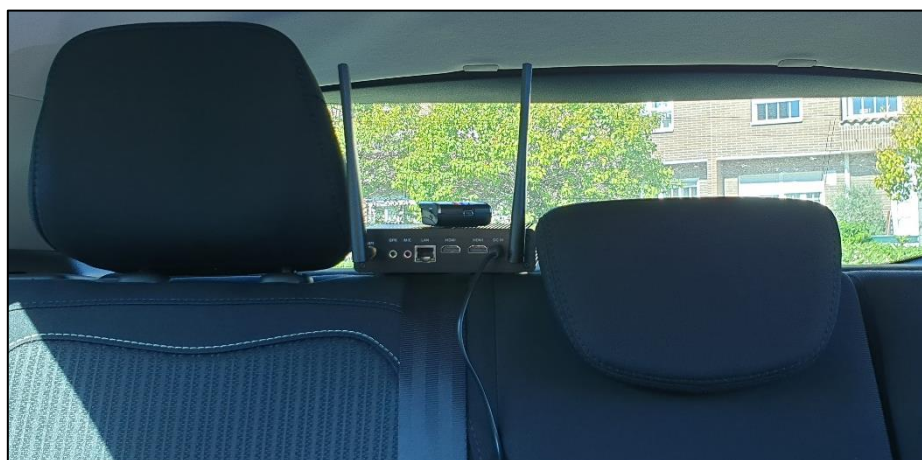
Una vez diseñado el recorrido, utilizando la aplicación *Google Maps* se estimó la distancia total del recorrido a realizar, así como el tiempo total de duración de la prueba [ 55 ]. A mayores, estos datos GPS fueron exportados con la aplicación *My Gmaps* a un fichero KML/KMZ [ 56 ]. Esta información fue utilizada a posteriori en el procesamiento de los datos como los puntos reales que se deberían haber seguido durante la ruta. La estimación resultante es 24,7 kilómetros en 33 minutos.

La prueba se realizó utilizando dos vehículos, donde se instalaron los equipos como se verá a continuación y en cada vehículo iban dos personas. Una de ellas se encargó de conducir, y el acompañante estaba al cargo del experimento, fundamentalmente que los equipos no se movieran dentro del coche durante el trayecto y pulsando los botones del gamepad para registrar el paso por los puntos de referencia.

En cuanto a la preparación del experimento, los ordenadores se instalaron uno en el salpicadero y otro en la bandeja trasera de cada automóvil, acompañados con sus respectivos gamepads, GPS y dispositivos OBD conectados a cada una de las tomas OBD, tal y como se puede ver en la Figura 91 y Figura 92.



*Figura 91. Instalación del miniordenador en el salpicadero del vehículo A*



*Figura 92. Instalación del miniordenador en la bandeja trasera del automóvil B*

La secuencia de pasos que se siguieron para realizar la prueba fue, en primer lugar, con el fin de empezar a recibir valores válidos desde el primer dato recibido se conectaron y encendieron todos los elementos menos los ordenadores. Este paso era especialmente importante para los datos GPS, que contienen la hora que se utilizará como sincronismo entre las máquinas (Ver el capítulo 3. Integración del software). Los ordenadores se encendieron a la vez para intentar tener la misma cantidad de datos en las mismas marcas de tiempo. Evidentemente la sincronización no es perfecta, pero sí lo suficientemente buena como para facilitar el análisis de los datos.

Para garantizar el correcto funcionamiento de todos los componentes antes de iniciar el recorrido, se realizó una breve conexión SSH a cada uno de los ordenadores y se

comprobó que los ficheros de datos estaban siendo actualizados con la información recibida de cada vehículo en tiempo real.

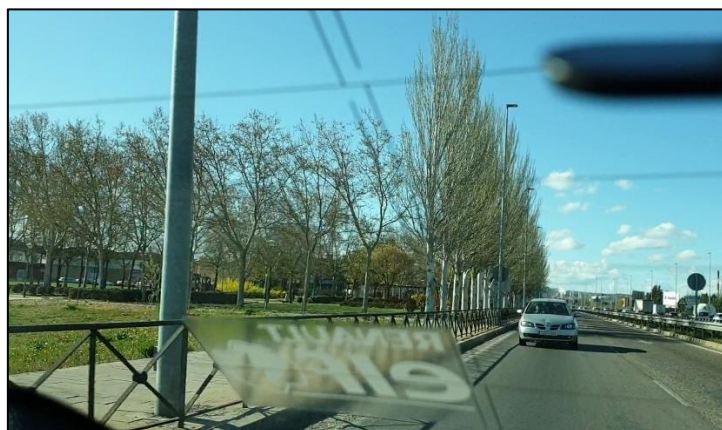
El experimento consistió en dos vehículos, circulando en fila por la ruta determinada, y variando las distancias entre ellos, para poder determinar si la comunicación entre los ordenadores es correcta en todo momento o si se pierden mensajes. Además, el recorrido se grabó con sendas cámaras para poder analizar el entorno en caso de que en algún lugar se produjeran errores no esperados. De la Figura 93 a la Figura 96 se pueden ver capturas del video grabado en diversos puntos del recorrido.



*Figura 93. Zona cercana desde el punto de vista coche A*



*Figura 94. Zona lejana desde el punto de vista coche A*



*Figura 95. Zona cercana desde el punto de vista coche B*



*Figura 96. Zona de máxima velocidad desde el punto de vista coche B*

Durante la prueba, no se realizó ningún adelantamiento ni cambio de carril, lo que garantizó que podamos utilizar los datos extraídos de Google Maps para calcular la ruta como puntos de referencia de posición del automóvil, con un error que sería la variación relativa de posición de los automóviles dentro del ancho de los carriles.

Al finalizar la prueba, se apagaron los dos ordenadores a la vez y se comprobaron que ambos habían estado registrando datos hasta el final del experimento. Por otro lado, se contabilizó la distancia recorrida mediante el panel del coche, habiendo recorrido 24,5 kilómetros tras la prueba.



*Figura 97. Cuentakilómetros del automóvil*

Dado que Gmaps nos dio una longitud del recorrido de 24,7 km, mientras que en la Figura 97 el odómetro nos da una longitud recorrida de 24,5 km, existe un pequeño error del uno por ciento, que podemos considerar asumible.

## **5.2 Procesado de datos**

El procesamiento de los datos se ha realizado con un programa realizado en Matlab. El código desarrollado nos permitió una automatización del procesado para evitar tener que realizar de nuevo todos los cálculos en caso de ser necesaria una segunda prueba. Además, se obtuvieron una gran cantidad de datos que hacen totalmente imposible el tratamiento de forma manual. A continuación, se expondrá cuál es el objetivo del tratamiento automático de los datos en las funciones más importantes y los resultados obtenidos tras esta prueba.

Los códigos desarrollados se encuentran detallados en el apartado Anexo sección 8.3 Demostración del sistema: caso de uso.

### 5.2.1 Script Main.m

Este script es el fichero principal del programa. Sirve como punto de unión entre el resto de los scripts (funciones). Desde este script se lanza el comando `run` para ejecutar el procesado de los datos y donde se muestran las gráficas y datos más relevantes. A medida que se va ejecutando el código, varios mensajes con información resumida son mostrados en la ventana llamada “*Command Window*” (Figura 98).

```
Command Window

GPS
La distancia calculada por el GPS del ordenador A es: 24.915992 kilometros
La distancia calculada por el GPS del ordenador B es: 24.900042 kilometros

OBD
La distancia calculada por el OBD del ordenador A es: 24.112890 kilometros
La distancia calculada por el OBD del ordenador B es: 24.621417 kilometros

La distancia calculada por GMAPS es: 24.837878 kilometros

Cams perdidos
Los cams perdidos que han sido enviados por A y recibidos por B son: 4
Los cams perdidos que han sido enviados por B y recibidos por A son: 22

fx >>
```

Figura 98. Consola con resumen de la prueba

### 5.2.2 Script cargarDatos.m

Esta función es la encargada de leer los diferentes ficheros `.xlsx` obtenidos de cada ordenador. Este script nos permite cambiar rápidamente los ficheros de entrada en caso de que realicemos algún experimento nuevo. La función devuelve al programa principal una serie de vectores con toda la información que va a ser procesada en el resto del código (posición GPS, velocidad, tiempos, CAMS enviados y recibidos por cada ordenador, etc).

La base de tiempo utilizada en los ficheros es *epoch*, es decir, los milisegundos transcurridos desde el 1 de enero de 1970. En nuestro caso, casi siempre vamos a utilizar diferencias de tiempo, por lo que no necesitamos traducir esos nanosegundos a fecha actual, pero sí que se han transformado a segundos para agilizar y simplificar las cuentas, evitando errores en las unidades.

Hay que destacar que ha sido necesario un ajuste de tiempo los datos del ordenador B, ya que la fecha de dicho ordenador se mantuvo en el 27-03-2019 en vez de cuando se realizó la prueba (04-04-2019).

```
DatosGPS_B(:,4) = DatosGPS_B(:,4) + 6.048e+14; % !!!!!!! OJO Esta sentencia se ha
utilizado para poner en el mismo día la
% grabación del ordenador B con el ordenador A, ya que B ha creado el fichero el día
27/03/2019
```

### 5.2.3 Script GPS.m

Este código procesa los datos GPS del experimento y calcula la distancia recorrida por el GPS mediante la fórmula de *Haversine* para determinar dicha distancia entre dos coordenadas GPS. [ 53 ]

Las salidas de esta función son un entero con la distancia recorrida según el GPS y un vector de distancias donde se va acumulando la distancia que se ha recorrido hasta el momento. En la Figura 99 podemos ver que ambos GPS han recorrido distancias similares.

```
GPS
La distancia calculada por el GPS del ordenador A es: 24.915992 kilometros
La distancia calculada por el GPS del ordenador B es: 24.900042 kilometros
```

Figura 99. Distancia total de cada GPS

### 5.2.4 Script OBD.m

Esta función es la encargada de procesar los datos obtenidos del dispositivo OBD, devolviendo la matriz de distancia acumulada en cada momento y mostrando en una gráfica dicha distancia acumulada durante la totalidad del experimento como se ve en la Figura 100.

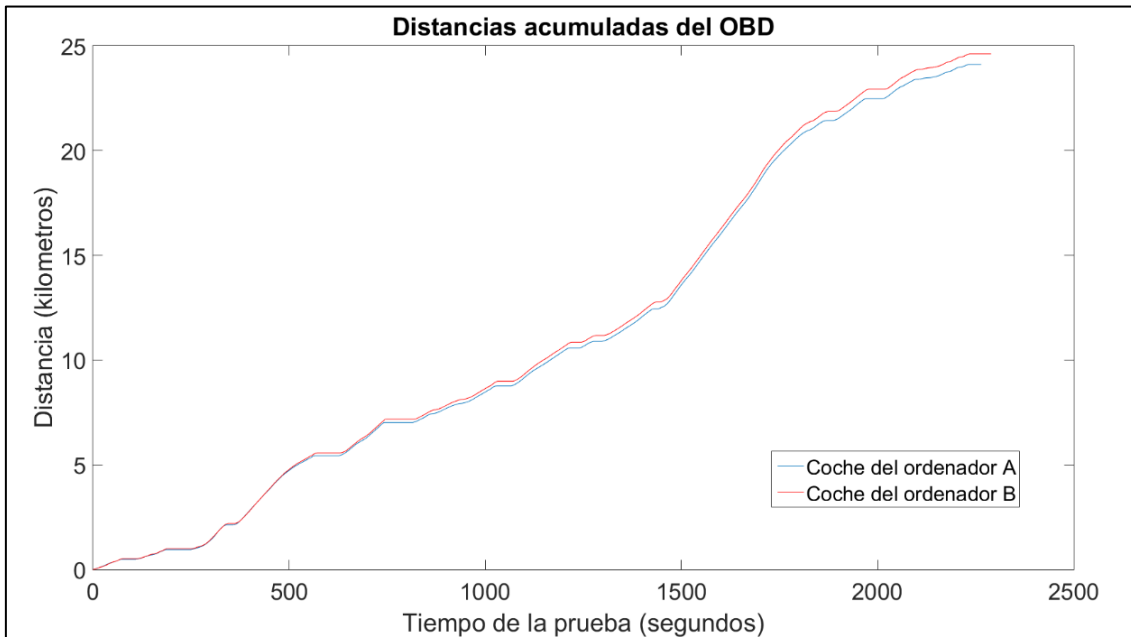


Figura 100. Distancia acumulada por el OBD en toda la prueba

Analizando la Figura 100, hemos decidido tomar solo los primeros valores donde los dos OBD tienen medidas más parecidas, porque al final del experimento existen grandes discrepancias entre los dos OBD utilizados. Esta diferencia no se debe a los dispositivos utilizados, sino que es culpa de los vehículos. De esta manera, no añadiremos los errores introducidos por el propio OBD al cálculo del error en el sistema GPS.

```
OBD
La distancia calculada por el OBD del ordenador A es: 24.112890 kilometros
La distancia calculada por el OBD del ordenador B es: 24.621417 kilometros
```

Figura 101. Distancia total obtenida por el OBD

En la Figura 101 detectamos que la distancia obtenida por el ordenador B es más parecida a la distancia real recorrida (24,7 Km en Gmaps frente a 24,62 kilómetros en OBD), por esta razón se decidió que no era necesario repetir experimento con otros modelos de gamepad y captar las balizas temporales previstas en la planificación del caso de uso.

### 5.2.5 Script Velocidad.m

Este script es el encargado de transformar las velocidades obtenida del OBD, de cm/s a Km/h y mostrar al usuario en una única gráfica la evolución de la velocidad de ambos vehículos (Figura 102).

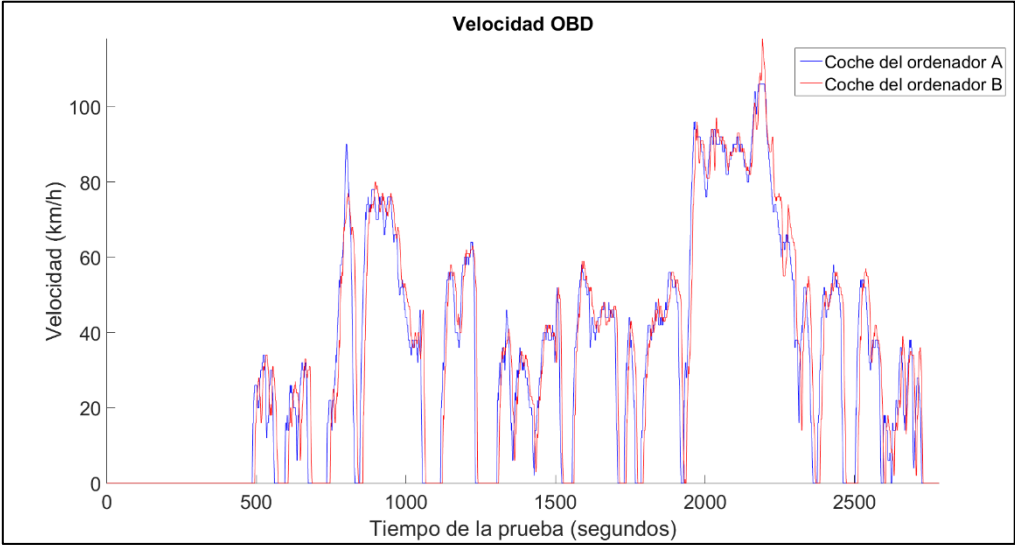
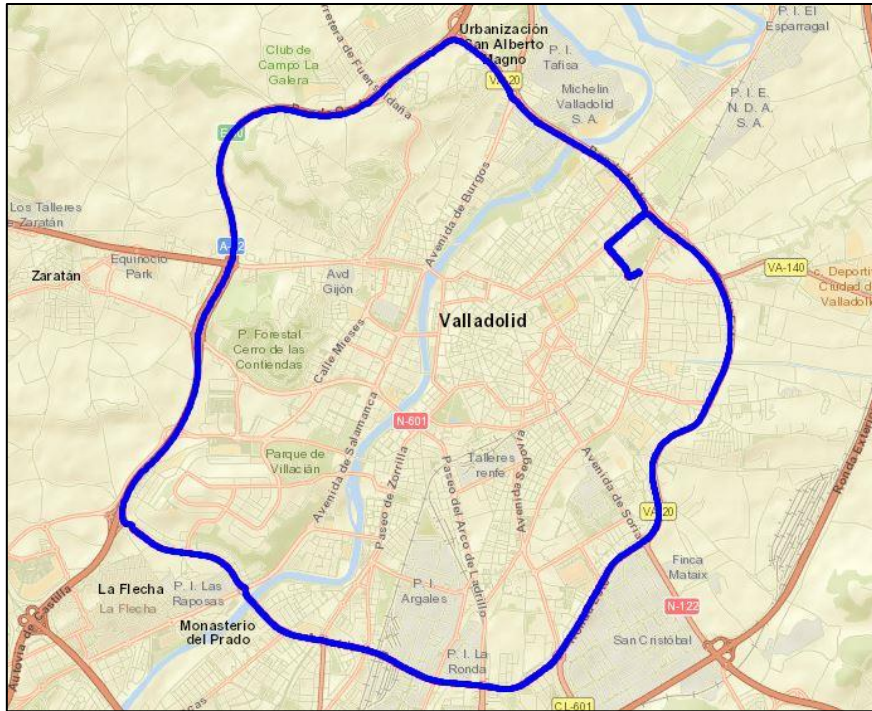


Figura 102. Velocidades alcanzadas durante la prueba

### 5.2.6 Script Ruta.m

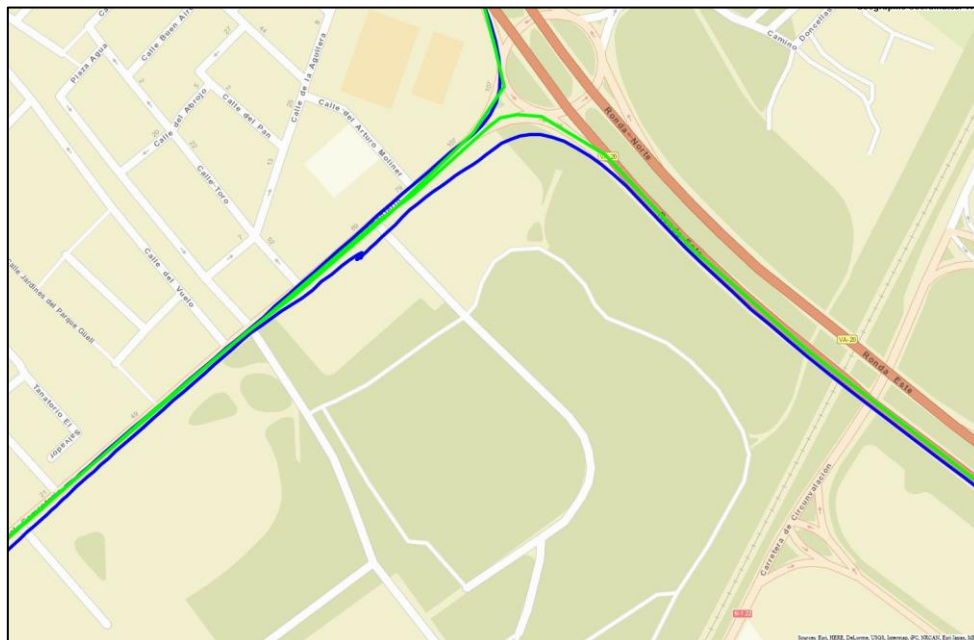
El script ruta es el encargado de dibujar sobre un mapa las coordenadas de la ruta que hemos realizado durante el caso de uso. El principal uso de este script es que nos permite comparar a simple vista las rutas seguidas por el GPS durante el experimento y la ruta que debería haber seguido realmente. En la Figura 103 se puede ver que la ruta seguida por el coche B, la cual coincide con la ruta planificada.





*Figura 103. Ruta registrada por el GPS B*

Utilizando este mapa, podemos observar que la relación entre los datos GPS y los datos Gmaps ya tienen discrepancias entre sí, que son detectables fácilmente. Esto se debe a los errores discutidos en secciones anteriores sobre los datos GPS, mientras que las rutas generadas con Gmaps dibujan la ruta por la zona central de la carretera (invadiendo ambos carriles). Este efecto se puede apreciar en la Figura 104.



*Figura 104. Diferencias entre GPS recibido y ruta Gmaps*

En cuanto a la comparativa entre las dos rutas GPS obtenidas de los diferentes automóviles, podemos apreciar que ambos GPS circulan por la misma ruta dentro de la vía (Figura 105). Aunque como vimos en secciones anteriores esto no es totalmente cierto, podemos realizar la asunción por simplificación que el error relativo entre los receptores

GPS es nulo, tal y como indicamos en la Fórmula 8, hecho que utilizaremos posteriormente para el desarrollo de nuestros cálculos.

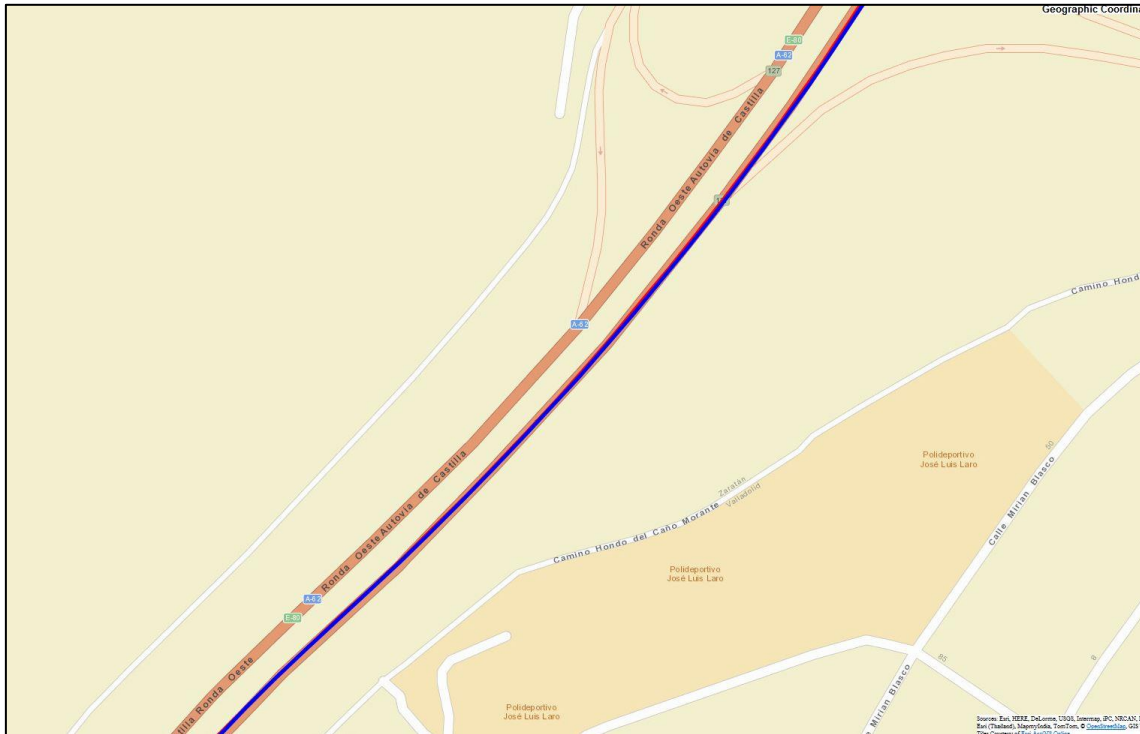


Figura 105. Rutas GPS similares en los datos obtenidos

$$\begin{aligned} (Lat_{GPS_1} + \varepsilon_{Lat}) - (Lat_{GPS_2} + \varepsilon_{Lat}) &= Lat_{GPS_1} - Lat_{GPS_2} \rightarrow \varepsilon_{Lat} \approx 0 \\ (Lon_{GPS_1} + \varepsilon_{Lon}) - (Lon_{GPS_2} + \varepsilon_{Lon}) &= Lon_{GPS_1} - Lon_{GPS_2} \rightarrow \varepsilon_{Lon} \approx 0 \end{aligned}$$

*Fórmula 8. Ecuación del error GPS*

Donde  $Lat_{GPS}$  es la latitud obtenida,  $Lon_{GPS}$  es la longitud del punto,  $\varepsilon_{Lat}$  es el error asignado a la componente en latitud y  $\varepsilon_{Lon}$  es el error de la componente en longitud.

Por otro lado, al ser inevitable las paradas en la ruta por los semáforos, esto provoca pequeños cambios en la posición recibida por el GPS. En el caso en el que el dispositivo fuese ideal, la posición sería siempre la misma y no se introducirían estos errores. En la Figura 106 podemos apreciar estos fallos.

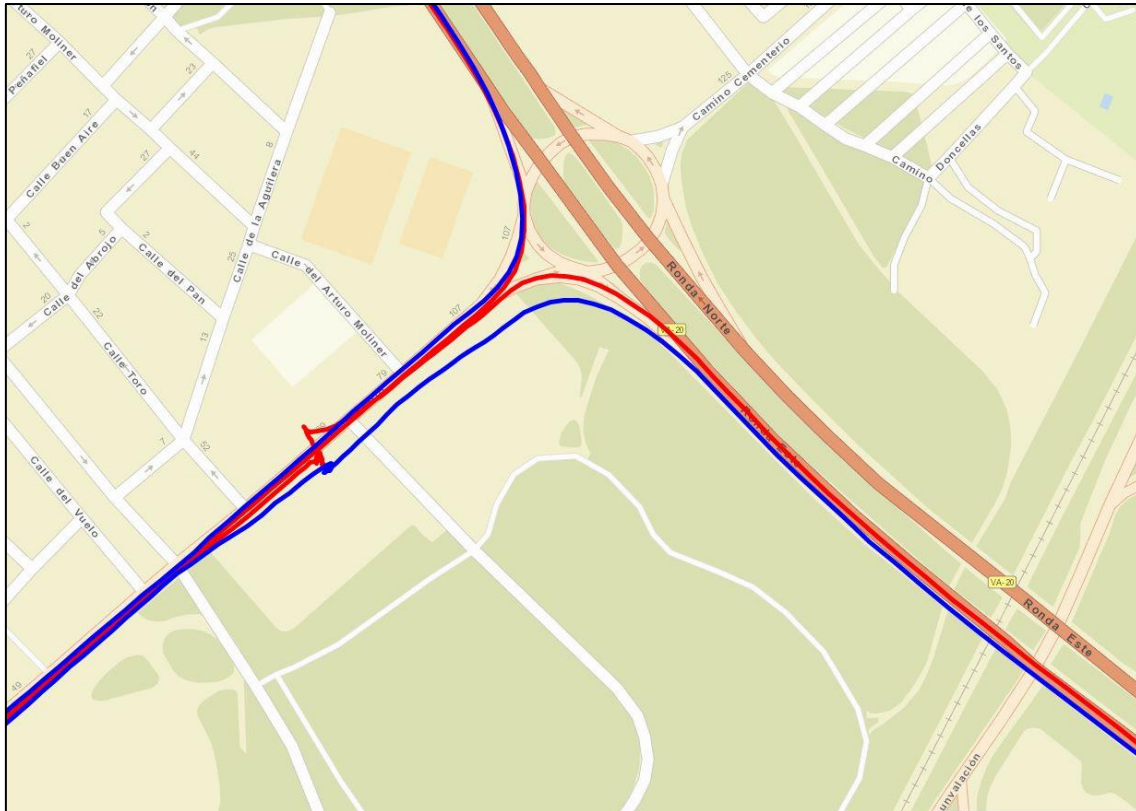


Figura 106. Errores añadidos por los semáforos

### 5.2.7 Script Gmaps.m

Esta parte del código desarrollado se encarga de procesar los datos de la ruta real (Gmaps), calcular la matriz de las distancias acumuladas en cada coordenada y mostrar la ruta de dichas coordenadas (en color verde) junto con las rutas GPS obtenidas. Dibujarlas de manera conjunta nos permite ver los errores de los que ya hemos hablado con anterioridad.

Por otro lado, también hemos calculado la distancia total de esta ruta, teniendo una ligera diferencia con la dada por la página web de Google (Figura 107).

```
La distancia calculada por GMAPS es: 24.837878 kilometros
```

Figura 107. Distancia real dada por Gmaps

### 5.2.8 Script Cams.m

La función *cams.m* se encarga de desconcatenar la información recogida en el logger, es decir, coge la información tanto de los mensajes CAMS enviados como de los CAMS recibidos por cada ordenador y realiza una comparación. De esta función se esperaba que hubiese una gran diferencia entre los CAMS enviados y recibidos, ya que, en tres ocasiones, vehículos ajenos se interpusieron entre los coches del experimento. Los resultados obtenidos indican que el número de CAMS perdidos es relativamente bajo. Los cams enviados y recibidos por cada ordenador se muestran en la Tabla 13 y Tabla 14

Enviados por A	Recibidos por B
1392	1388

Tabla 13. Número de mensajes CAM enviados por A y recibidos por B

Recibidos por A	Enviados por B
1379	1401

Tabla 14. Número de mensajes CAM recibidos por A y enviados por B

En el segundo caso, vemos que hay una mayor diferencia entre los enviados y los recibidos. Teniendo en cuenta que los 5 primeros enviados por B se perdieron porque A aún no estaba encendido (8 segundos de diferencia) y A dejó de transmitir 18 segundos antes que B, eso hace 9 mensajes. De los 22 que se han perdido de B hacia A, realmente son 8.

```
Cams perdidos
Los cams perdidos que han sido enviados por A y recibidos por B son: 4
Los cams perdidos que han sido enviados por B y recibidos por A son: 22
```

Figura 108. Relación entre cams enviados y recibidos por cada ordenador

En el fichero `config.xml` localizado en `/OpenC2X/dcc/config/`, hemos detectado que los valores de `NDL_minPacketInterval` y `NDL_maxPacketInterval` fueron modificados respecto a la que dicta el estándar, siendo su valor de 2 y 4 segundos, respectivamente. El estándar indica que estos valores han de ser **0,04 y 1 segundos**, es decir, que como mínimo transcurran 0,04 segundos entre un CAM y otro consecutivo, y no pueden pasar más de 1 segundo entre dos CAMS sucesivos. Este cambio provocó que los CAMS esperados durante la prueba no fuesen los realmente obtenidos.

Por otro lado, un CAM debe ser enviado cuando se produce un cambio de distancia de más de 4 metros. Sin embargo, nuestro GPS tiene un periodo mínimo (frecuencia máxima) de 1 segundo, lo que implica que a velocidades mayores de 15 km/h (que corresponde con 4 metros/segundo) es imposible que siga el periodo de las muestras que se solicitan en el DCC de OpenC2X y no se manden tantos CAMS por el cambio de posición como ordena el estándar. [ 54 ]

En primera instancia, el objetivo inicial de este código era ver si había alguna relación entre la distancia entre los vehículos y la pérdida de los mensajes CAMS. Finalmente, al ver que la relación de pérdidas es muy pequeña, se ha descartado este proceso.

### 5.2.9 Script `CalculoPosicionGPS.m`

Este código tiene la misión de coordinar los datos Gmaps, GPS y OBD para determinar el valor de la posición GPS en los momentos en los que se toman los datos OBD. Esto nos permitirá comprobar cuál es la diferencia entre los dos tipos de medidas. Si suponemos que la del OBD es más exacta que la del GPS, entonces podríamos hablar de error del GPS y, al contrario, aunque debido a la poca finura de ambos datos solo hablaremos de errores relativos, sin referirnos a quien se imputa este error. Para hacer todo este cálculo, usaremos los datos de velocidad del OBD para trasladar los datos de Gmaps a las posiciones temporales de GPS.

El resultado final será un gráfico que nos indica la distancia entre la posición que nos indica el OBD a través de la integral de velocidad y la que obtenemos a través del GPS.

El primer paso realizado consiste en cuadrar los datos del Gmaps con el OBD. Primero se han comparado las distancias acumuladas en cada punto disponible, datos obtenidos de funciones anteriores. En el caso de Gmaps, este vector se obtiene mediante la distancia entre dos puntos consecutivos en el mapa, y en OBD, utilizando la fórmula de la distancia en función de la velocidad y la diferencia de tiempo:

$$D = v * \Delta t$$

Donde D es la distancia recorrida, v es la velocidad y  $\Delta t$  es el intervalo de tiempo donde se registró la velocidad.

Una vez tenemos los vectores de distancias, se ha ido comparando todas y cada una de las distancias acumuladas del OBD con el vector de distancias de Gmaps, y así obtenemos los dos puntos más próximos entre estos sistemas para cada distancia recorrida. De esta manera, sabemos entre que dos puntos del Gmaps se encuentra la distancia del OBD.

Mediante este dato, hemos interpolado las coordenadas correspondientes de Gmaps para cada instante de tiempo del OBD. Este proceso se ha llevado a cabo mediante el teorema de los triángulos semejantes (Figura 109) y las fórmulas obtenidas de dicho teorema (desde la Fórmula 9 hasta la Fórmula 14).

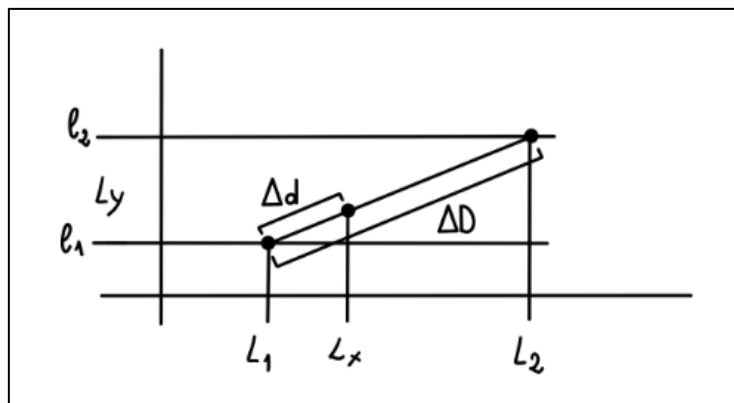


Figura 109. Esquema del teorema de los triángulos semejantes

$$\Delta D = D_2 - D_1$$

Fórmula 9. Distancia entre puntos Gmaps

$$\Delta d = d_1 - D_1$$

Fórmula 10. Distancia entre OBD y Gmaps

$$\frac{L_2 - L_1}{\Delta D} = \frac{L_x - L_1}{\Delta d}$$

Fórmula 11. Relación de triángulos semejante para la Latitud

$$\frac{l_2 - l_1}{\Delta D} = \frac{l_y - l_1}{\Delta d}$$

Fórmula 12. Relación de triángulos semejante para la Longitud

Despejando de la Fórmula 11 y la Fórmula 12, obtenemos la siguientes ecuaciones:

$$L_x = \frac{(L_2 - L_1)\Delta d + L_1\Delta D}{\Delta D}$$

Fórmula 13. Nueva Latitud calculada

$$l_y = \frac{(l_2 - l_1)\Delta d + l_1\Delta D}{\Delta D}$$

Fórmula 14. Nueva Longitud calculada

Donde  $L_x$  es la Latitud real deseada,  $l_y$  es la longitud real,  $L_1$  y  $L_2$  son las latitudes de Gmaps,  $l_1$  y  $l_2$  son las longitudes de Gmaps,  $\Delta D$  es la distancia entre los dos puntos de Gmaps y  $\Delta d$  es la distancia entre el punto Gmaps y el punto OBD. Es importante señalar que estamos haciendo cálculos sobre un área muy pequeña, se ha hecho la asunción de que las coordenadas se comportan para esta área como un sistema cartesiano de coordenadas.

Tras este proceso, obtenemos la latitud y longitud “real” para cada medida obtenida con el OBD y guardamos la marca temporal del OBD, que será utilizada a continuación.

El siguiente paso será relacionar este parámetro con las medidas GPS obtenidas durante la prueba. Al igual que en el caso anterior, tenemos que cuadrar los datos para que puedan ser comparados directamente, mediante el tiempo del GPS. Moveremos las coordenadas del OBD-Gmaps al tiempo obtenido con el GPS. El proceso será el mismo, utilizando el teorema de los triángulos semejantes, lograremos pasar de un tiempo inicial OBD al tiempo final GPS (Figura 110). Para ello, usaremos la velocidad y la aceleración del OBD y así obtener un cálculo mucho más preciso (Fórmula 15 y Fórmula 16).

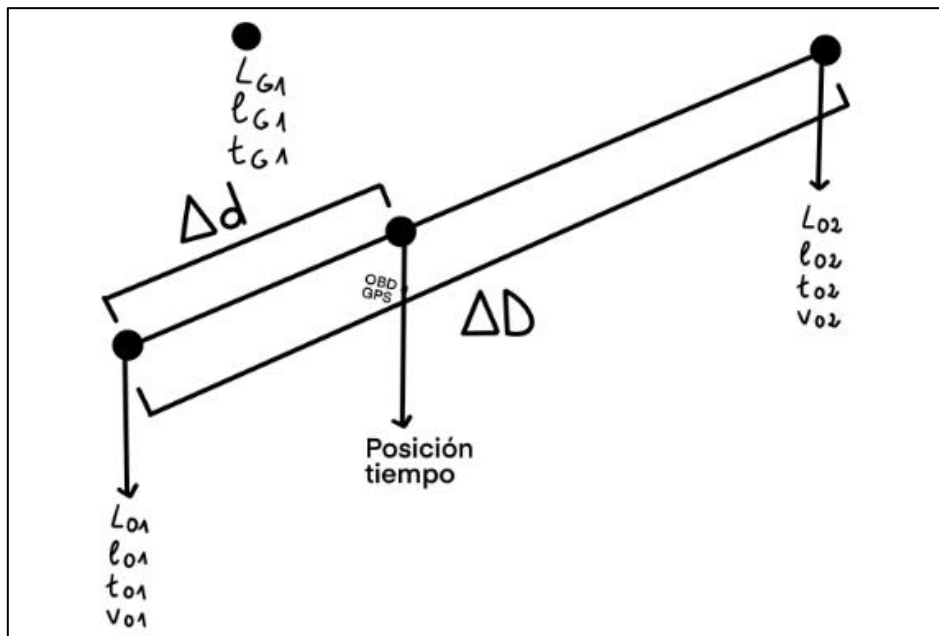


Figura 110. Triángulo rectángulo para coordinar tiempo GPS-OBD

$$a_{01-2} = \frac{v_{02} - v_{01}}{t_{02} - t_{01}}$$

Fórmula 15. Ecuación de la aceleración

$$\Delta d = \frac{1}{2} a_{01-2} (t_{G1} - t_{01})^2 + v_{01} ((t_{G1} - t_{01}))$$

Fórmula 16. Diferencia de distancia en función de la aceleración, velocidad y tiempo

Donde  $a_{0_{1-2}}$  es la aceleración entre los puntos,  $v_0$  es la velocidad en cada instante,  $t_0$  es el instante de tiempo de la muestra,  $t_G$  es el instante de tiempo del GPS (punto a calcular) y  $\Delta d$  es la distancia entre el punto 1 de Gmaps-OBD y el punto GPS.

El punto  $0_{01}$  y el punto  $0_{02}$  son los puntos OBD-Gmaps entre los que se encuentra temporalmente la muestra GPS, representada con el símbolo  $G_1$  (Figura 110). El punto intermedio en la recta, separado  $\Delta d$  de  $P_{01}$  es donde coinciden en tiempo la muestra OBD-Gmaps y la muestra GPS.

Una vez tenemos todos los resultados, procedemos a calcular el error obtenido mediante el cálculo de las distancias entre dos coordenadas geográficas (*Haversine*). El error obtenido se muestra en la Figura 111.

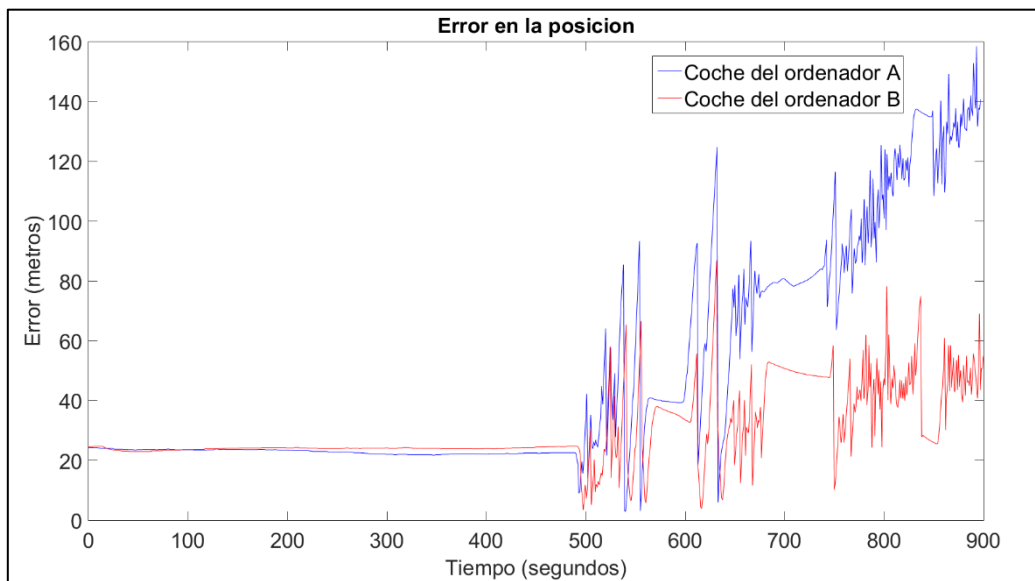


Figura 111. Error entre posición estimada por el OBD y posición GPS obtenida

Como se puede apreciar, al final del recorrido tenemos un error muy elevado, esto se debe a los errores introducidos tanto por Gmaps (la ruta no es perfectamente real) y por el error propio del OBD. Ambos dispositivos OBD utilizados son iguales, pero la tasa de obtención de datos de uno de ellos es mayor que la del otro vehículo ( $f_b > f_a$ ) obteniendo así datos mucho más reales. Esta decisión es debida a los vehículos utilizados para realizar la prueba (El coche A no permitía una frecuencia más alta).

Por otro lado, vemos que, en las primeras medidas de error obtenida, existen errores entre 20 y 40 metros. Este valor es el error típico introducido por los sistemas GPS, por lo que podríamos decir que en la primera parte de la prueba las medidas obtenidas con el OBD no acumularon mucho error.

Con el objetivo de obtener un error más cuantitativo, se han decidido coger los primeros 3 kilómetros del recorrido, y así poder estimar un error más preciso. Este proceso se explicará en el siguiente script. Los errores medios y máximo de las distancias se exponen en la Figura 112.

```

Error maximo
El error maximo del ordenador A es: 158.541790 metros en los 3 primeros kilometros
El error maximo del ordenador B es: 86.933019 metros en los 3 primeros kilometros

Error medio
El error medio del ordenador A es: 48.348451 metros en los 3 primeros kilometros
El error medio del ordenador B es: 30.088145 metros en los 3 primeros kilometros

```

Figura 112. Error medio y máximo entre GPS y estimación de posición con el OBD

### 5.2.10 Script OBDLimitado.m

Esta parte del código se encarga de recortar y mostrar los datos recibidos del OBD en el rango donde podemos suponer fiables las medidas recibidas, en los primeros 3 kilómetros (Figura 113).

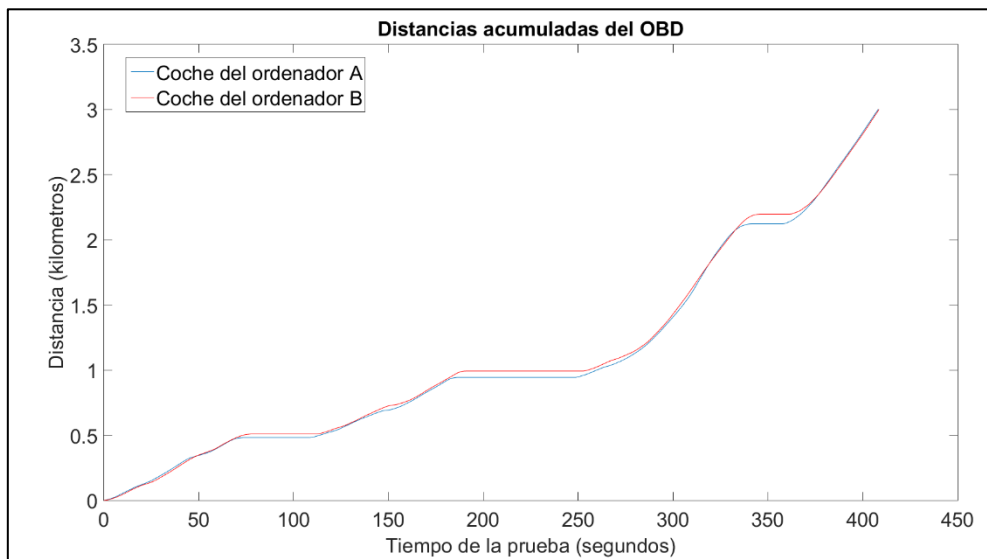


Figura 113. Distancia acumulada hasta 3 kilómetros

### 5.2.11 Script DistanciaEntreCochesGPS.m

El siguiente script nos permitirá calcular distancia entre los vehículos en los dos casos bajo estudio:

- Utilizando los datos de los dos GPS directamente.
- Utilizando los datos obtenidos con Gmaps y OBD de cada ordenador.

En ambos casos, el código utilizado es el mismo. Finalmente, obtendremos un gráfico donde se representa la distancia entre cada coche en cada instante de tiempo. El resultado esperado es una distancia no muy lejana, ya que el recorrido tenía zonas lentas y un máximo de 150 metros en las zonas rápidas, en torno al final del experimento.

Al igual que en el script Script CalculoPosicionGPS.m, utilizaremos el teorema de semejanza de triángulos para emparejar los tiempos de los dos GPS y así poder comparar la latitud y longitud en el mismo instante temporal, es decir, obtener la distancia entre los vehículos. Para ello, al igual que en el caso anterior, se tiene que dar una coordinación entre los tiempos, esto es, si queremos comparar la distancia entre dos coches, la posición GPS debería ser cogida en el mismo instante. Como esto no es así, deberemos trasladar



la posición del segundo coche a la que tendría si la hubiéramos tomado en el mismo instante que el primer coche. El esquema utilizado se muestra en la Figura 114.

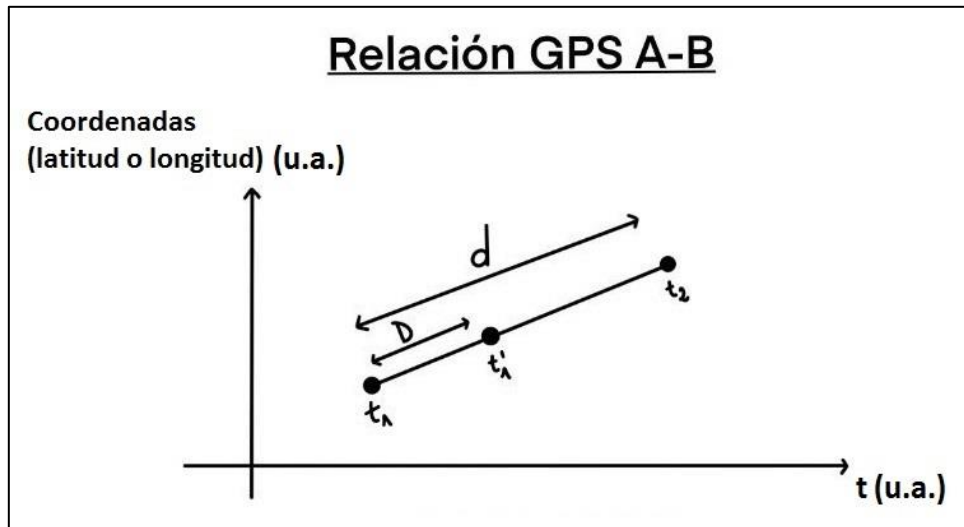


Figura 114. Relación temporal entre muestras GPS A y GPS B

Conociendo la distancia entre los puntos en  $t_2$  y  $t_1$  y el tiempo  $t_1'$ , podemos calcular la distancia entre los puntos  $t_1$  y  $t_1'$  con la Fórmula 17

$$d = D * \frac{t_2 - t_1'}{t_2 - t_1}$$

Fórmula 17. Distancia entre dos puntos en función del tiempo

Donde  $d$  es la distancia entre los puntos del GPS A,  $D$  es la distancia entre el punto GPS A número 1 y el punto GPS B,  $t_2$  y  $t_1$  son los tiempos del GPS A y  $t_1'$  es el tiempo del GPS B.

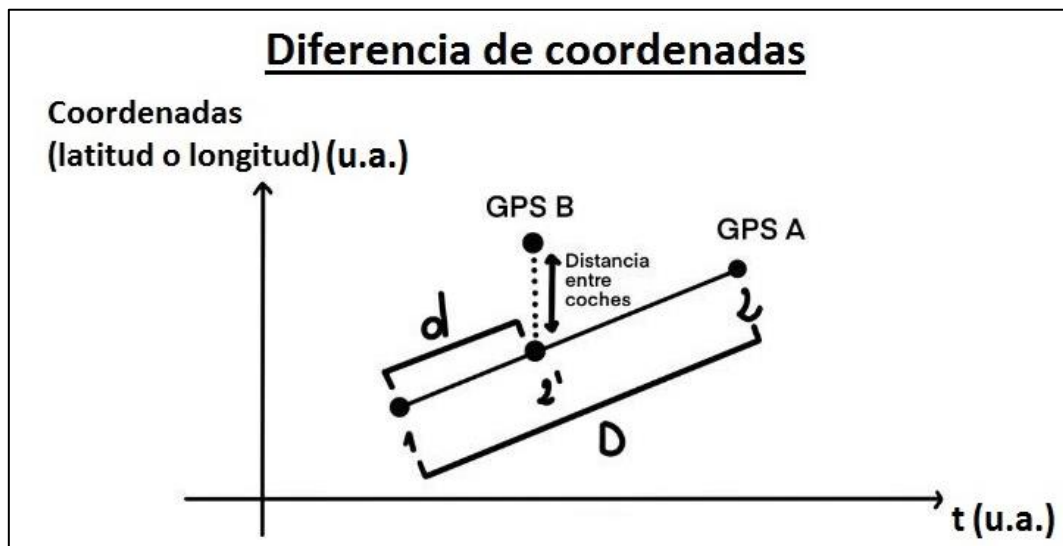


Figura 115. Ajuste de la posición del GPS A a tiempo GPS B

Aplicando el teorema de los triángulos de nuevo en la Figura 115, podemos sacar las longitudes y latitudes correspondientes al nuevo tiempo, que coincide con el GPS B, mediante la Fórmula 18.

$$\frac{L_2 - L_1}{d} = \frac{L'_2 - L_1}{D} \rightarrow L'_2 = (L_2 - L_1) \frac{D}{d} + L_1$$

Fórmula 18. Fórmula para calcular las coordenadas de GPS A en tiempo de GPS B

Ante la Figura 116, se puede ver una gran diferencia entre los casos explicados. Esta diferencia se atribuye al error introducido por el OBD a lo largo de todo el experimento. Sin embargo, sí que se puede apreciar una correlación entre los datos, es decir, ambos se juntan y se separan en las mismas zonas y tienen relación con la ruta realizada, ya que, en las zonas de mayor velocidad, el vehículo de delante aumenta su velocidad antes que el segundo, provocando un distanciamiento, mientras que, en las zonas lentas con semáforos, los vehículos están próximos.

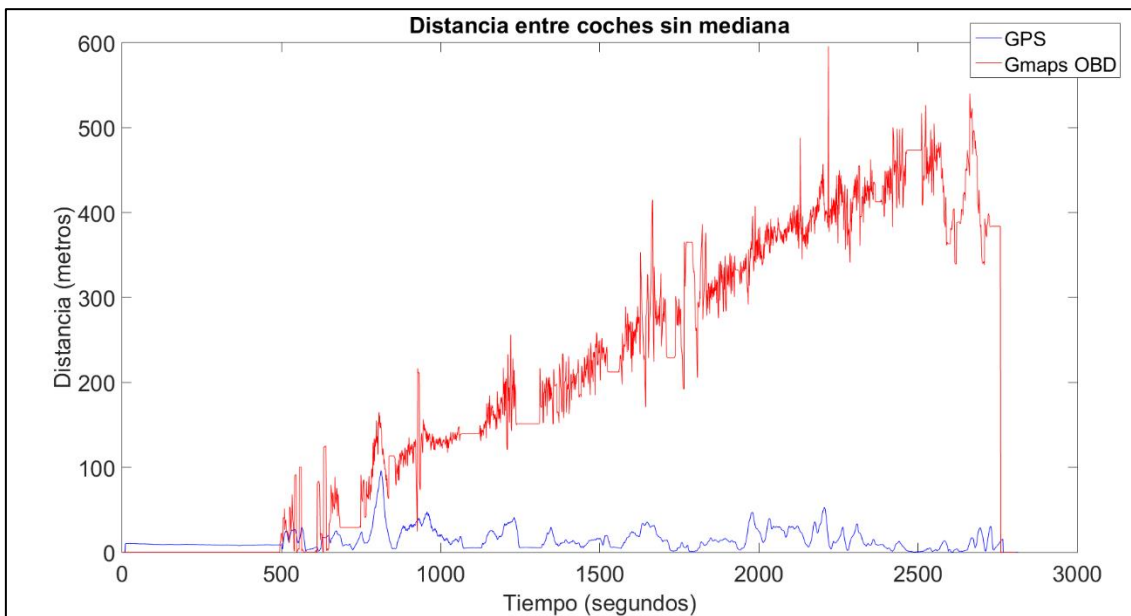


Figura 116. Distancia entre coches con GPS

Observando la figura anterior, comprobamos que existe un error estático en los resultados obtenidos, ya que hasta el segundo 450, los vehículos estaban parados mientras realizábamos las correspondientes comprobaciones en todos los elementos. Durante este proceso, ambos vehículos estaban parados en paralelo, con una distancia máxima entre ellos de 2 metros. La Figura 117 muestra que la distancia entre GPS obtenida es de unos 10 metros, es decir, existe un error de 8 metros respecto a la distancia real, valor comprendido en el error indicado por el fabricante [ 54 ]. En cuanto a OBD, también existe un pequeño error inicial, ya que aún no se tiene muestras de la velocidad.

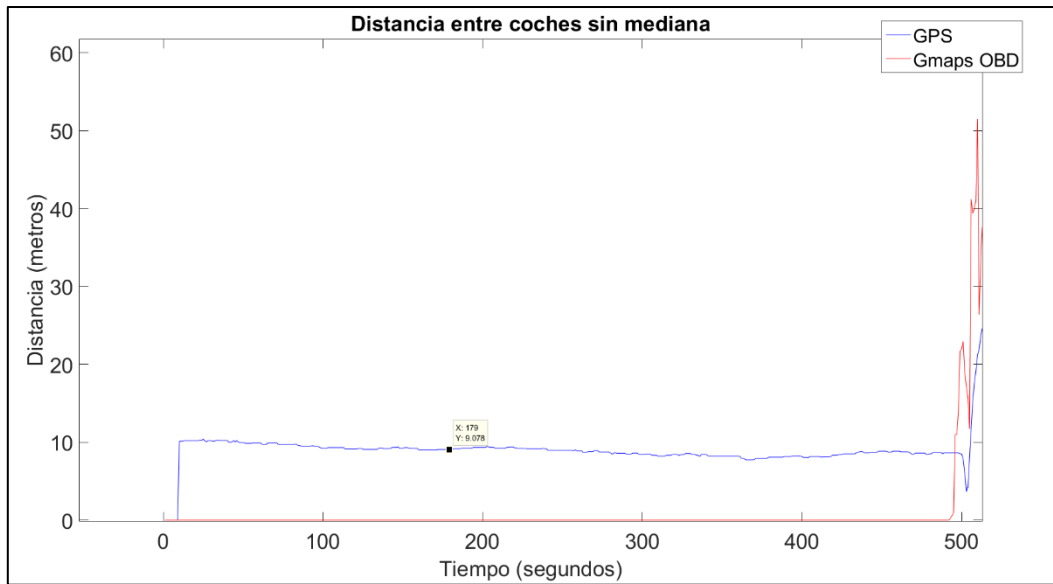


Figura 117. Error intrínseco al sistema GPS con coches aparcados en paralelo

Para poder determinar el error introducido por el GPS y también, tener una visualización más nítida de la distancia entre vehículos, se ha aplicado un filtro de mediana para eliminar el ruido impulsivo de las muestras recogidas, es decir, cada muestra se promedia junto al valor de la muestra anterior y la muestra siguiente (Figura 118).

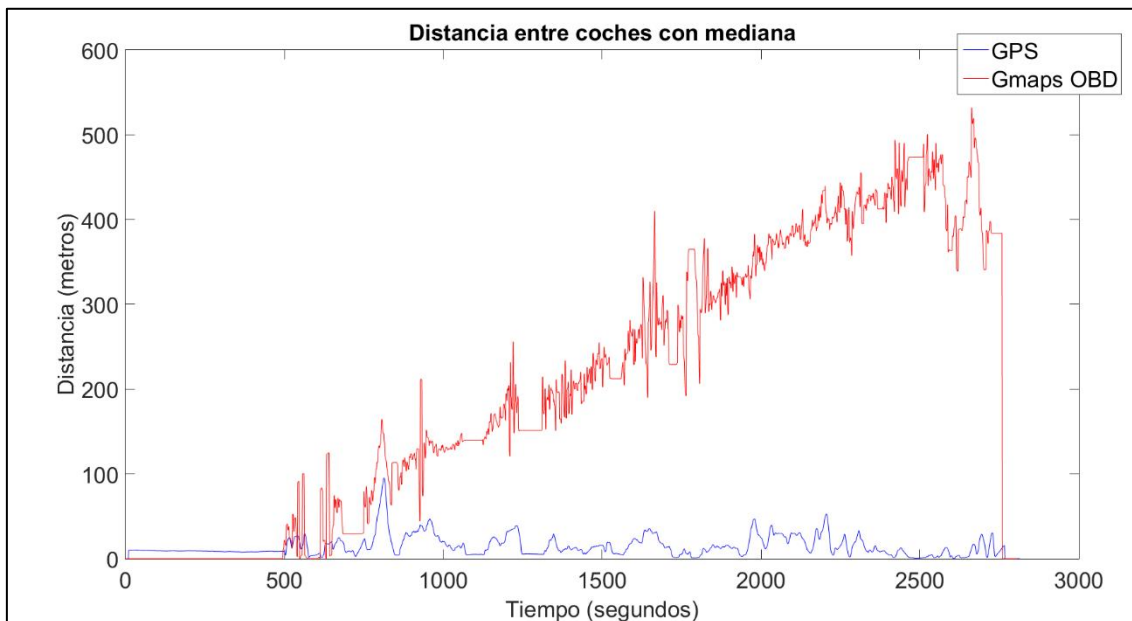


Figura 118. Distancia entre coches con filtro de mediana

Finalmente, realizando una resta entre la gráfica GPS obtenida y la gráfica filtrada, obtenemos el error introducido por el sistema GPS en cada muestra tomada, tanto en el caso de solo muestras GPS (Figura 119), como en la distancia OBD (Figura 120).

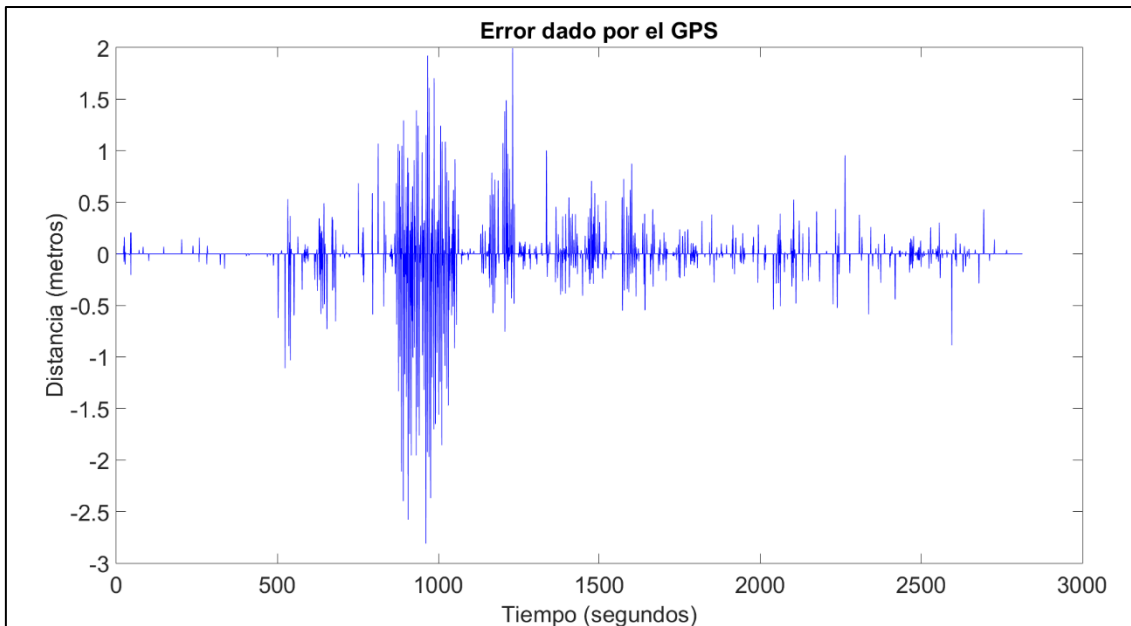


Figura 119. Error introducido por el ruido impulsivo en las muestras GPS

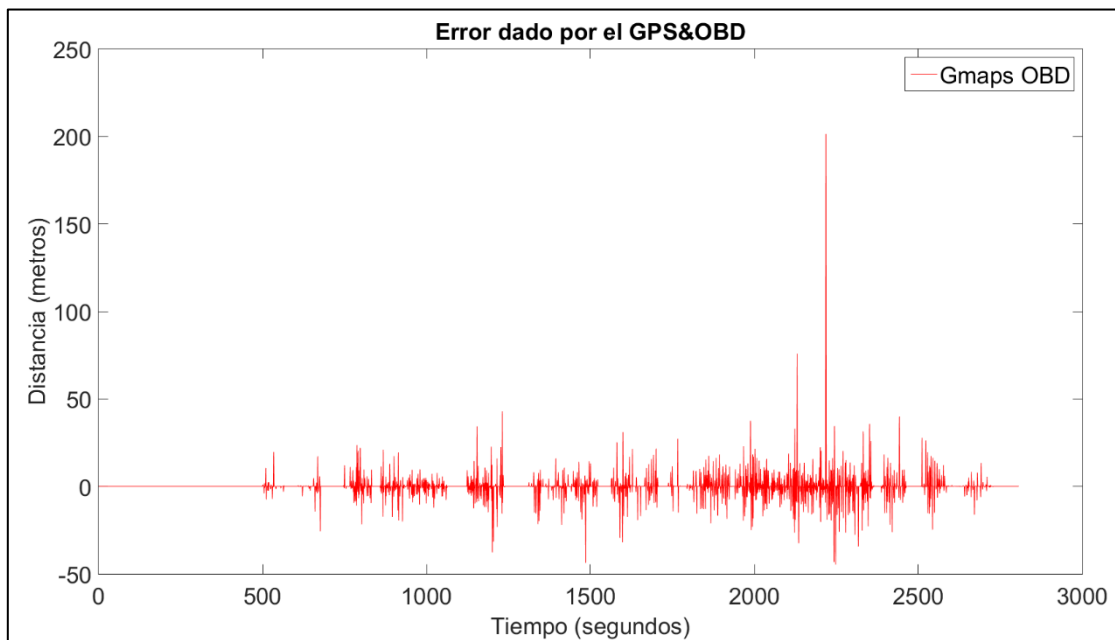


Figura 120. Error introducido por el ruido impulsivo en las muestras Gmaps y OBD

El error introducido por solo el GPS es como maximo de hasta 2,5 metros, mientras que en el caso del OBD, a este error se le suma el error introducido por Gmaps y el propio sistema OBD.

# 6. Conclusiones y líneas futuras

## 6.1 Conclusiones

El principal objetivo de este trabajo de Fin de Máster era lograr una correcta comunicación entre dos unidades de a bordo en un entorno realista, utilizando el software OpenC2X para transmitir información de control entre estos sistemas.

En primer lugar, se realizó una correcta integración del software en los dos ordenadores de pequeño tamaño. Esta integración consistió en la instalación del sistema operativo Linux con *kernel* modificado, la configuración de los drivers de la tarjeta de red Atheros ATH9K para dicho *kernel*, instalación del software del proyecto de código abierto OpenC2X, configuración de los nuevos dispositivos a utilizar: dispositivo OBD, mando *gamepad* y geolocalizador GPS y, por último, automatizar todos los elementos para que se ejecuten al iniciar el sistema operativo.

Una vez lograda la integración del software, se realizó una caracterización completa del funcionamiento global del sistema, mediante análisis individuales de los diferentes componentes que afectan a la comunicación entre las dos unidades (ganancia de la antena, potencia final transmitida, modulación utilizada y coeficiente de pérdidas).

Por último, se realizó un experimento en un entorno real para determinar la fiabilidad y eficacia del sistema obtenido, en concreto, los errores introducidos por el sistema GPS, el error acumulado por el dispositivo OBD en cada vehículo, una comparación entre la posición que se debería obtener con el GPS y las coordenadas recibidas, y las pérdidas de paquetes CAMS que se produjeron durante la prueba.

Tras estos resultados se ha de reseñar que el sistema es totalmente operativo y funcional pero que requiere de varios aspectos que pueden ser mejorados en gran medida, como el *gamepad* utilizado o los ordenadores, debido a su mal funcionamiento en general.

En cuanto a los resultados de la comunicación, se ha demostrado que el sistema es capaz de transmitir información correctamente hasta la distancia mínima de seguridad, cubriendo el objetivo marcado. En el aspecto referido a GPS, tenemos dos receptores que tienen un error estático asociado considerable, pero que podría resolverse con un nuevo dispositivo, utilizando el sistema de localización de Galileo o con otro algoritmo de procesamiento de los datos crudos del GPS.

## 6.2 Propuestas de mejora

Durante este proyecto, se han detectado pequeñas deficiencias en el funcionamiento de los dispositivos externos con el software OpenC2X. Por esta razón se proponen una serie de cambios que permitan solucionar estos fallos:

- En cuanto al GPS, el software OpenC2X debe enviar un nuevo mensaje CAM cuando se ha producido un cambio en la posición de 4 metros. Esto significa que, circulando a la velocidad máxima permitida en carreteras españolas, 120 km/hora o 33,33 metros por segundo, necesitaríamos que nuestro GPS recopilará de nuevo la posición del vehículo cada 0,12 segundos. Actualmente, el GPS utilizado tiene un periodo mínimo de una muestra por segundo, es decir, enviamos un único nuevo CAM cuando deberíamos haber enviado 9 mensajes nuevos.  
Por otro lado, hay que tener en cuenta la precisión del dispositivo GPS. Lo ideal sería que fuésemos capaces de determinar por qué carril circula cada vehículo,
- En cuanto al OBD, se ha determinado que una tasa de muestreo de 100 ms es suficiente para poder utilizar esta información
- El mando gamepad utilizado en el experimento: caso de uso tiene un funcionamiento muy aleatorio debido a la calidad del producto. Por esta razón, se han adquirido nuevos mandos con una mejor conexión bluetooth.
- Sustituir las actuales tarjetas de red Atheros por otras que permitan transmitir 23 dBm/MHz de potencia de pico, como indica el estándar.

En cualquiera de los casos, si fuese imposible adquirir hardware que cumpliera con estos requisitos descritos, sería necesario hacer un post-procesado de la información mediante la fusión de datos para calcular la posición y la velocidad aparente del automóvil.

### 6.3 Líneas futuras

Este proyecto se encuentra en un estado muy evolucionado respecto al trabajo inicial desarrollado en años anteriores. Sin embargo, este campo de investigación aún tiene mucho margen de mejora, sobre todo en algunas funcionalidades que no están implementadas todavía y corrección de fallos asociados a los ordenadores utilizados:

- Implementación de un sistema de *GeoNetworking* que permita un realizar un mapa de posicionamiento de los vehículos en la zona.
- Integración de una capa de seguridad en el software de comunicación.
- Implementar una *Road Side Unit* (RSU) que permita una comunicación V2I y el reenvío de los mensajes recibidos por las unidades de a bordo.
- Mejorar la precisión del sistema GPS mediante algoritmos de procesamiento de datos y lograr diferenciar el carril en que se encuentran los vehículos.
- Sustituir los ordenadores por *HummingBoards* para facilitar los experimentos de campo.
- Realizar pruebas de campo con un mayor número de vehículos.
- Explorar la tecnología LTE v.11 y 5G, y una posible hibridación con la capa de acceso del estándar 802.11p.

# 7. Bibliografía

- [ 1 ] Revista DGT. “En 2018, 1.180 fallecidos, 18 menos que el año anterior” <http://revista.dgt.es/es/noticias/nacional/2019/01ENERO/0103-Presentacion-balance-accidentes-2018.shtml>. Último acceso: 19/05/2019
- [ 2 ] Carlos López J. “Dónde está realmente el coche autónomo a día de hoy y qué han prometido las marcas en el Salón de Ginebra para el futuro” Accesible desde <https://www.xataka.com/automovil/donde-esta-realmente-el-coche-autonomo-a-dia-de-hoy-y-que-han-prometido-las-marcas-en-el-salon-de-ginebra-para-el-futuro> Último acceso: 19/05/2019
- [ 3 ] Iglesias D. “Primer atropello mortal de un vehículo sin conductor “ <https://www.elmundo.es/motor/2018/03/19/5aaff150e2704ec82e8b463f.html> Último acceso: 19/05/2019
- [ 4 ] Javier Fernández Pastrana, 2017. “802.11p standard and V2X applications on comercial Wi-Fi cards”, Universidad de Valladolid.
- [ 5 ] Pilar Sánchez Martín, “Implementación de una unidad de a bordo de comunicación entre vehículos según el estándar ETSI ITS G5”, Universidad de Valladolid, 2017.
- [ 6 ] Lobo González, A. “Implementación de Road Side Unit de comunicación infraestructura vehículos según el estándar ETSI ITS G5”, Universidad de Valladolid, 2017.
- [ 7 ] Perfil de Illan Gonzales Horna: Disponible en: <https://www.linkedin.com/in/illan-glez-horna/?originalSubdomain=es> Último acceso: 19/05/2019
- [ 8 ] Daniel Jiang ; Luca Delgrossi “IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments” Disponible en: <https://ieeexplore.ieee.org/document/4526014> Último acceso: 19/05/2019
- [ 9 ] S. Laux, G. Singh Pannu, S. Schneider, J. Tiemann, F. Klingler, C. Sommer and F. Dressler. 2016. “Demo: OpenC2X – An Open Source Experimental and Prototyping Platform Supporting ETSI ITS-G5”, CSS-Labs. Accesible desde <http://www.ccs-labs.org/bib/laux2016openc2x/laux2016openc2x.pdf>. Último acceso: 19/05/2019
- [ 10 ] Khadige Abboud, Hassan Aboubakr Omar; Weihua Zhuang. “Interworking of DSRC and Cellular Network Technologies for V2X Communications: A Survey” Disponible en: <https://ieeexplore.ieee.org/abstract/document/7513432> Último acceso: 19/05/2019
- [ 11 ] 5G Automotive Association. “The Case for Cellular V2X for Safety and Cooperative Driving” Disponible en: <http://5gaa.org/wp->

[content/uploads/2017/10/5GAA-whitepaper-23-Nov-2016.pdf](#) Último acceso: 19/05/2019

[ 12 ] Jian Wang, Yameng Shao, Yuming Ge and Rundong Yu. “A Survey of Vehicle to Everything (V2X) Testing” Disponible en: <https://www.mdpi.com/1424-8220/19/2/334> Último acceso: 19/05/2019

[ 13 ] Björn Schünemann. “V2X simulation runtime infrastructure VSimRTI: An assessment tool to design smart traffic management systems “ Disponible en: <https://www.sciencedirect.com/science/article/abs/pii/S1389128611001605> Último acceso: 26/05/2019

[ 14 ] Mikael Nilsson; Paul Hallbjörner; Niklas Arabäck; Björn Bergqvist et al “Multipath propagation simulator for V2X Communication Tests on Cars” Disponible en: <https://ieeexplore.ieee.org/abstract/document/6546497> Último acceso: 26/05/2019

[ 15 ] Zishan Liu, Zhenyu Liu, Zhen Meng, et al. “Implementation and performance measurement of a V2X communication system for vehicle and pedestrian safety”. Disponible en: [https://journals.sagepub.com/doi/full/10.1177/1550147716671267#\\_i15](https://journals.sagepub.com/doi/full/10.1177/1550147716671267#_i15) Último acceso: 26/05/2019

[ 16 ] Christian Weiß. “V2X communication in Europe – From research projects towards standardization and field testing of vehicle communication technology” Disponible en: <https://www.sciencedirect.com/science/article/abs/pii/S1389128611001198> Último acceso: 26/05/2019

[ 17 ] Qualcomm. “Accelerating C-V2X Commercialization”. Disponible en: <https://www.qualcomm.com/media/documents/files/accelerating-c-v2x-commercialization.pdf> Último acceso: 19/05/2019

[ 18 ] Mercedes-Benz. “Sistema PRE-SAFE®” Disponible en: [https://www.mercedes-benz.es/passengercars/mercedes-benz-cars/models/a-class/hatchback/explore/intelligent\\_drive\\_detail/presafe.html](https://www.mercedes-benz.es/passengercars/mercedes-benz-cars/models/a-class/hatchback/explore/intelligent_drive_detail/presafe.html) Último acceso: 19/05/2019

[ 19 ] “Introduction to CAN FD” Disponible en: <https://www.can-fd.com/introduction-can-fd/> Último acceso: 19/05/2019

[ 20 ] Mexpert. “First Network Interface for Car2x/V2X Communication” Disponible en: [https://www.presseagentur.com/vector/detail.php?pr\\_id=5164&lang=en](https://www.presseagentur.com/vector/detail.php?pr_id=5164&lang=en) Último acceso: 19/05/2019

[ 21 ] Benedikt Brecht, Dean Therriault, André Weimerskirch, William Whyte Virendra Kumar, Thorsten Hehn y Roy Goudy. “A Security Credential Management System for V2X Communications“ Disponible en: <https://ieeexplore.ieee.org/abstract/document/8309336> Último acceso: 19/05/2019

[ 22 ] Stéphanie Lefèvre, Jonathan Petit, Ruzena Bajcsy, Christian Laugier y Frank Kargl. “Impact of V2X privacy strategies on Intersection Collision Avoidance systems” Disponible en: <https://ieeexplore.ieee.org/abstract/document/6737592> Último acceso: 19/05/2019



- [ 23 ] Página web del Car 2 Car Communication Consortium. Disponible en <https://www.car-2-car.org/> Último acceso: 19/05/2019
- [ 24 ] Mahbubul Alam. “V2X for Autonomous Vehicle: 5G or IEEE 802.11p?” Disponible en: <http://mahbubulalam.com/v2x-for-autonomous-vehicle-5g-or-ieee-802-11p/> Último acceso: 19/05/2019.
- [ 25 ] Revista 5G Americas White Paper “Cellular V2X Communications Towards 5G” Último acceso: 19/05/2019. Disponible en: [http://www.5gamericas.org/files/9615/2096/4441/2018\\_5G\\_Americas\\_White\\_Paper\\_Cellular\\_V2X\\_Communications\\_Towards\\_5G\\_Final\\_for\\_Distribution.pdf](http://www.5gamericas.org/files/9615/2096/4441/2018_5G_Americas_White_Paper_Cellular_V2X_Communications_Towards_5G_Final_for_Distribution.pdf)
- [ 26 ] Pagina web oficial Linux Ubuntu: <https://www.ubuntu.com/> Último acceso: 19/05/2019.
- [ 27 ] Especificaciones de las unidades de abordó: [https://www.amazon.es/gp/product/B01N00MZ50/ref=ppx\\_yo\\_dt\\_b\\_asin\\_title\\_o00\\_s0?ie=UTF8&th=1](https://www.amazon.es/gp/product/B01N00MZ50/ref=ppx_yo_dt_b_asin_title_o00_s0?ie=UTF8&th=1) Último acceso: 18/06/2019.
- [ 28 ] Página web de distribuciones de Ubuntu: <https://www.kernel.org/pub/linux/kernel/v4.x/> . Último acceso: 19/05/2019.
- [ 29 ] Javier Fernández Pastrana “ath9k\_ocr\_linux\_4.2.8” Disponible en: [https://github.com/0x90/atheros-arsenal/tree/master/ath9k\\_ocr\\_linux\\_4.2.8](https://github.com/0x90/atheros-arsenal/tree/master/ath9k_ocr_linux_4.2.8) Último acceso: 19/05/2019.
- [ 30 ] Universidad de Paderborn. “OPENC2X” Disponible en <http://www.ccs-labs.org/software/openc2x/> Último acceso: 19/05/2019
- [ 31 ] “¿Cómo funciona un lector de códigos OBD?” Disponible en: <https://iberisasl.com/blog/como-funciona-un-lector-de-codigos-OBd/> Último acceso: 19/05/2019.
- [ 32 ] Allen Lyons, Michael McCarthy “Transitioning Away from Smog Check Tailpipe Emission. Testing in California for OBD II Equipped Vehicles”. Disponible en: [http://cetia.us/yahoo\\_site\\_admin/assets/docs/transitioning\\_to\\_obd\\_only\\_im\\_ARB\\_White\\_Paper\\_March\\_2009.173131454.pdf](http://cetia.us/yahoo_site_admin/assets/docs/transitioning_to_obd_only_im_ARB_White_Paper_March_2009.173131454.pdf) Último acceso: 06/06/2019
- [ 33 ] Ubuntu manuals “Minicom” Disponible en: <http://manpages.ubuntu.com/manpages/trusty/man1/minicom.1.html> Último acceso: 19/05/2019.
- [ 34 ] Obd solutions “reading real-time data” Disponible en: <https://www.OBdSol.com/knowledgebase/OBD-software-development/reading-real-time-data/> Último acceso: 19/05/2019.
- [ 35 ] Hexadecimal to Decimal Converter: <https://www.binaryhexconverter.com/hex-to-decimal-converter>
- [ 36 ] “NMEA data” Disponible en: <https://www.GPSinformation.org/dale/nmea.htm> Último acceso: 19/05/2019.

- [ 37 ] Ask Ubuntu “How to run a script during boot as root” Disponible en: <https://askubuntu.com/questions/290099/how-to-run-a-script-during-boot-as-root> Último acceso: 19/05/2019.
- [ 38 ] Epoch & Unix Timestamp Conversion Tools. Disponible en: <https://www.epochconverter.com/> Último acceso: 19/05/2019.
- [ 39 ] Poole, Ian. “Newnes Guide to Radio and Communications Technology” Disponible en: <http://web.b.ebscohost.com.ponton.uva.es/ehost/detail/detail?vid=0&sid=433b56cc-85eb-434a-a1f2-578299f288c7%40pdc-v-sessmgr01&bdata=Jmxhbmc9ZXMmc2l0ZT1laG9zdC1saXZlJnNjb3BIPXNpdGU%3d#AN=104803&db=nlebk> Último acceso: 19/05/2019.
- [ 40 ] Hoja de especificaciones antenas Taoglas GSA.8859.A.105111. Disponible en: <https://cdn.taoglas.com/datasheets/GSA.8859.A.105111.pdf> Último acceso: 19/05/2019.
- [ 41 ] Arie Voors. “NEC based antenna modeler and optimizer” Software disponible en: <https://www.qsl.net/4nec2/> Último acceso: 19/05/2019.
- [ 42 ] “Agilent Technologies 8719D/20D/22D Network Analyzers”. Analizador de redes HP Hewlett Packard 8719D. Disponible en: <https://literature.cdn.keysight.com/litweb/pdf/08720-90288.pdf> Último acceso: 19/05/2019.
- [ 43 ] Analizador de espectros Rohde&Scharz Disponible en: [https://cdn.rohde-schwarz.com/pws/dl\\_downloads/dl\\_common\\_library/dl\\_manuals/gb\\_1/f/fsh\\_1/archive\\_27/FSH\\_Quick\\_Start\\_Manual\\_Espanol\\_12.pdf](https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_common_library/dl_manuals/gb_1/f/fsh_1/archive_27/FSH_Quick_Start_Manual_Espanol_12.pdf) Último acceso: 19/05/2019.
- [ 44 ] What is iPerf / iPerf3? Disponible en: <https://iperf.fr/>. Último acceso: 19/05/2019.
- [ 45 ] USRP-210. Disponible en: <https://kb.ettus.com/N200/N210> Último acceso: 19/05/2019.
- [ 46 ] HackRF. Disponible en: <https://greatscottgadgets.com/hackrf/> Último acceso: 19/05/2019.
- [ 47 ] Champlain College, Leahy center for digital investigation “HackRF One” Disponible en: [https://www.champlain.edu/Documents/LCDI/HackRF%20One%20Tutorial\\_F2017%20-%20Report.docx.pdf](https://www.champlain.edu/Documents/LCDI/HackRF%20One%20Tutorial_F2017%20-%20Report.docx.pdf) Último acceso: 19/05/2019.
- [ 48 ] GNU Radio. Disponible en: <https://www.gnuradio.org/> Último acceso: 19/05/2019.
- [ 49 ] GNU Radio WiFi Reception <https://github.com/bastibl/gr-ieee802-11> Último acceso: 19/05/2019
- [ 50 ] IEEE Standards Association, ‘IEEE Standard for information technology Telecommunications and information exchange between systems Local and metropolitan area networks’. IEEE Computer Society, 03-2012.

- [ 51 ] Ecuaciones para calcular el Bit Error Rate (BER). Disponible en: <https://www.keysight.com/main/editorial.jspx?ckey=1481106&id=1481106&nid=-11143.0.00&lc=eng&cc=ES> . Último acceso: 19/05/2019.
- [ 52 ] Carlos Nicolás Fraile. “Distancia de seguridad: escudo protector” 10 octubre 2017. Disponible desde: <http://revista.dgt.es/es/educacion-formacion/conducir-mejor/2017/1026Distancia-de-seguridad.shtml>. Último acceso: 19/05/2019.
- [ 53 ] Distance on a sphere: The Haversine Formula. Disponible en: <https://community.esri.com/groups/coordinate-reference-systems/blog/2017/10/05/haversine-formula> Último acceso: 19/05/2019.
- [ 54 ] Holux GPSlim 236. Bluetooth GPS Receiver. User’s Guide. Disponible en: [http://www.iamnota.net/media/hw:gpslim236\\_manual\\_e-031705.pdf?id=hw%3Agpslim\\_236&cache=cache](http://www.iamnota.net/media/hw:gpslim236_manual_e-031705.pdf?id=hw%3Agpslim_236&cache=cache) Último acceso: 05/06/2019
- [ 55 ] Google Maps <https://www.google.es/maps/> Último acceso: 19/05/2019.
- [ 56 ] Creación de mapas con Google Maps. Disponible en: [https://www.google.com/intl/es\\_ES/maps/about/mymaps/](https://www.google.com/intl/es_ES/maps/about/mymaps/) Último acceso: 19/05/2019.



# 8. Anexo de código

## 8.1 Integración del software

### 8.1.1 Gamepad

```
#include "gamepad.h"

GamePad::GamePad(boost::asio::io_service& io):
    mIo(io), mSd(io), mEvents()
{
}

GamePad::~GamePad() {
}

void GamePad::start()
{
    boost::system::error_code error;
    int dev;
    for(int i=0;i<5;i++) {
        //event6 sin conectar y event8 con mouse, pantalla y teclado conectado
        dev = open("/dev/input/event6", O_RDONLY); // Configuración A
        //dev = open("/dev/input/event2", O_RDONLY); // Configuración B
        if(dev == -1){
            std::cout << "[GamePad::start] Error when opening the GamePad"
<< std::endl;
            boost::this_thread::sleep(boost::posix_time::seconds(10));
        } else {
            break;
        }
    }
    mSd.assign(dev,error);
    if(!error) {
        std::cout << "[GamePad::start] Assigned!" << std::endl;
        mSd.non_blocking(true);
    }
    async_read();
}

void GamePad::stop()
{
    std::cout << "Turning off the computer" << std::endl;
    system("/home/gco/Descargas/OpenC2X/scripts/stopOpenC2X.sh"); // Configuración
A
    // system("/home/gco2/Descargas/OpenC2X/scripts/stopOpenC2X.sh"); //
Configuración B
    //system("rm /etc/rc5.d/S06init.sh"); // Configuración A
    //system("rm /etc/rc3.d/S06init.sh"); // Configuración B
    system("pkill gpsd");
    system("pkill rfcmm");
    system("poweroff");
}
```

Figura 121. Código Gamepad (1/2)

```

void GamePad::handler_input(boost::system::error_code ec, size_t bytes_transferred)
{
    if (!ec) {
        auto const n = bytes_transferred / sizeof(input_event);
        for(size_t i=0; i<n; ++i) {
            auto& ev = mEvents[i];
            //using namespace boost::posix_time;
            //ptime ts({1970,1,1}, seconds(ev.time.tv_sec) +
microsec(ev.time.tv_usec));
            //std::cout << std::dec << ts.time_of_day() << "\t" <<
std::hex << std::hex << ev.type << " " << ev.code << " " << ev.value << "\n";
            manage_event(ev.code, ev.value, ev.time.tv_sec, ev.time.tv_usec);
        }
        //std::cout << "\n";
        async_read();
    } else {
        std::cerr << ec.message() << "\n";
    }
}

void GamePad::async_read()
{
    mEvents.resize(32);
    mSd.async_read_some(boost::asio::buffer(mEvents),
boost::bind(&GamePad::handler_input, this, boost::asio::placeholders::error,
boost::asio::placeholders::bytes_transferred));
}

void GamePad::manage_event(int code, int value, int sec, int usec)
{
    //std::cout << "[GamePad::manage_value] The value is: " << value << std::endl;
    if((code == 304 || code == 168) && value == 1){
        boost::posix_time::ptime ts({1970,1,1},
boost::posix_time::seconds(sec) + boost::posix_time::microsec(usec));
        //std::cout << ts.time_of_day() << std::endl;
        std::ofstream myfile1;
        // Opens the file and writes at the end
        myfile1.open ("/home/gco/Descargas/Logs/GamePad.txt",std::ios::app);
// Configuración A
        //myfile1.open
("/home/gco2/Descargas/Logs/GamePad.txt",std::ios::app); // Configuración B
        myfile1 << ts.time_of_day() << "\n";
        myfile1.close();
        std::cout << "X" << std::endl;
    }
    else if((code == 305 || code == 28) && value == 1){
        system("curl -vi -X POST -d '{"content": "triggered by GUI"}'
http://localhost:1188/trigger_denm");
        std::cout << "@" << std::endl;
        // Back slashes (\) are used to escape quotation marks (") at the
system call
    }
    else if(code == 1 && value == 2){
        std::cout << "IOS" << std::endl;
        stop();
    }
}

```

Figura 122. Código Gamepad (2/2)

## 8.1.2 init.sh

```
#!/bin/bash

#Configuración A
WLAN=wlp2s0
OCB_WLAN=ocb0
DIR_IP=192.168.1.2
MASK=255.255.255.0
MAC_GPS="00:0B:0D:84:CD:3C"
MAC_OBD="00:12:6F:10:53:34"
#MAC_GAMEPAD="2C:FF:8B:01:00:07"

#Setting down the wireless interface
ip link set $WLAN down
#Adding a new interface in OCB mode
iw dev $WLAN interface add $OCB_WLAN type ocb
#Setting OCB mode
iw dev $OCB_WLAN set type ocb
#Setting down OCB interface
ip link set $OCB_WLAN down
#Setting wireless regulatory domain to AA
iw reg set AA
#Has it changed?
iw reg get
#Rising OCB interface
ip link set $OCB_WLAN up
#Joining interface to Control Channel and 10MHz of bandwidth
#iw dev $OCB_WLAN ocb join 5990 10MHZ
#Leaving the Control Channel
#iw dev $OCB_WLAN ocb leave
#Joining interface to Service Control Channel 1
iw dev $OCB_WLAN ocb join 5880 10MHZ
#Was it successful?
iw dev | iwconfig

#Setting up IP address, netmask and default gateway
ifconfig $OCB_WLAN $DIR_IP netmask $MASK

#rm /etc/rc5.d/S06init.sh

/usr/bin/rfcomm connect 0 $SMAC_GPS &
sleep 10
killall gpsd
systemctl stop gpsd.socket
systemctl stop gpsd.service

sleep 10
/home/gco/Descargas/Scripts/ssh.sh &
sleep 10

##### Cambiamos la hora al pc para que ambos tengan la misma (RHB)
cat /dev/rfcomm0 | grep "GPGGA" > /tmp/hora.txt &
sleep 10
killall cat
```

Figura 123. Script init.sh (1/2)

```

## Separamos la hora de la forma HHMMSS a HH:MM:SS
hora=$(cut -b 1,2 <<< "${date2}")
hora=$((hora+2))
minutos=$(cut -b 3,4 <<< "${date2}")
segundos=$(cut -b 5,6 <<< "${date2}")

#Colocamos la hora
date +%T -s "$hora:$minutos:$segundos"
sleep 10
#####

gpsd -b -N -D 4 /dev/rfcomm0 &
sleep 10
#####
#ping 192.168.1.4 -n 1000 > /home/gco/Escritorio/comandoping"${hora}":"${minutos}".txt &
#####

/usr/bin/rfcomm connect 1 SMAC_OBD &
sleep 10
/home/gco/Descargas/OpenC2X/scripts/runOpenC2X.sh &
sleep 30
while true
do
    /home/gco/Descargas/GamePad/gamepad
done
#/home/gco/Descargas/GamePad/gamepad &
sleep 20 #300
#/home/gco/Descargas/OpenC2X/scripts/stopOpenC2X.sh
#rm /etc/init.d/init.sh
#rm /etc/rc5.d/S06init.sh
#poweroff

```

Figura 124. Script init.sh (2/2)

### 8.1.3 Copiar.sh

```

#!/bin/bash

rm /etc/init.d/init.sh
cp /home/gco/Descargas/Scripts/init.sh /etc/init.d/init.sh
ln -s /etc/init.d/init.sh /etc/rc5.d/S06init.sh

```

Figura 125. Script copiar.sh

### 8.1.4 Borrar.sh

```

#!/bin/bash

rm /etc/init.d/init.sh
rm /etc/rc5.d/S06init.sh

```

Figura 126. Script borrar.sh

### 8.1.5 Ssh.sh

```

#Red privada para poder hacer ssh con Windows
ip link set enpls0 down
sleep 10
ip link set enpls0 up
sleep 15
ifconfig enpls0 169.254.201.169 netmask 255.255.0.0

```

Figura 127. Script ssh.sh



## 8.2 Caracterización de los elementos del sistema

### 8.2.1 Coeficiente de reflexión de potencia $s_{11}$

```
planas = load('DATA00.F1');
wificuadrada = load('DATA01.F1');
wifiredonda = load('DATA02.F1');

puntos=2000/200; %MHz
ejex=4000:puntos:6000;

figure (1)
set(0,'defaultfigurecolor',[1 1 1])
plot(ejex,planas(:,1),'b',ejex,wificuadrada(:,1),'r',ejex,wifiredonda(:,1),'g');
title('Reflection coefficient s11')
xlabel('Frecuency (MHz)')
ylabel('s11 (dB)')
legend('Antenas Planas','Antenas Wifi Cuadradas','Antenas Wifi Redondas');
grid on
set(findall(gcf,'-property','FontSize'),'FontSize',26)
set(gcf, 'Color', 'white')

figure (2)
hold on
set(0,'defaultfigurecolor',[1 1 1])

% rectangle('Position',[5855 -55 10 50], 'FaceColor',[0.6350, 0.0780, 0.1840], 'EdgeColor','r')
% rectangle('Position',[5865 -55 10 50], 'FaceColor',[0.4660, 0.6740, 0.1880], 'EdgeColor','g')
% rectangle('Position',[5875 -55 10 50], 'FaceColor',[0, 0.4470, 0.7410], 'EdgeColor','b')
% rectangle('Position',[5885 -55 10 50], 'FaceColor',[0.25, 0.25, 0.25], 'EdgeColor','black')
% rectangle('Position',[5895 -55 10 50], 'FaceColor',[0.9290, 0.6940, 0.1250], 'EdgeColor',
[0.9290, 0.6940, 0.1250])
axis ([5855 5905 -25 -10])

plot(ejex,planas(:,1),'b',ejex,wificuadrada(:,1),'r',ejex,wifiredonda(:,1),'g');
title('Channels')
xlabel('Frecuency (MHz)')
ylabel('Reflection coefficient s11 (dB)')
legend('Antenas Planas','Antenas Wifi Cuadradas','Antenas Wifi Redondas');
grid on

%Dibujamos la linea vertical en la frecuencia de interes
x1 = 5880; %Mhz de interes
y1=get(gca,'ylim')
hold on
plot([x1 x1],y1, 'black-.')
%set(gca,'ycolor','w');
% set(gcf,'color','w');

%Texto plana
txt = ' -19.15 dB';
text(5880,-19.15-0.37,txt);
plot(ejex(189),planas(189,1),'black*')

%Texto cuadrada
txt = ' -18.08 dB';
text(5880,-18.08+0.37,txt);
plot(ejex(189),wificuadrada(189,1),'black*')

%Texto redonda
txt = ' -15.37 dB';
text(5880,-15.37-0.37,txt);
plot(ejex(189),wifiredonda(189,1),'black*')

set(findall(gcf,'-property','FontSize'),'FontSize',26)
set(gcf, 'Color', 'white')
```

Figura 128. Código coeficiente de reflexión  $s_{11}$  (1/2)

```
figure (3)
hold on
set(0,'defaultfigurecolor',[1 1 1])
axis ([5400 5800 -60 -5])

plot(ejex,planas(:,1),'b',ejex,wificuadrada(:,1),'r',ejex,wifiredonda(:,1),'g');
title('WiFi 5G range')
xlabel('Frecuency (MHz)')
ylabel('Reflection coefficient s11 (dB)')
legend('Antenas Planas','Antenas Wifi Cuadradas','Antenas Wifi Redondas');
grid on
set(findall(gcf,'-property','FontSize'),'FontSize',26)
set(gcf, 'Color', 'white')
```

*Figura 129. Código coeficiente de reflexión  $s_{11}$  (2/2)*



```

%
% % Dibujamos los puntos dentro de la circunferencia
% [m,n] = size(DatosGPS)
% radius = 0.01 * ones(m,1);
% [lat1, lon2] = scircle1(DatosGPS(:,1), DatosGPS(:,2), radius, az, e);
% wmline(lat1, lon2, 'Color', 'blue'); %Con esto tarda 270
segundos
%
% wmmarker(DatoRealLatitud,DatoRealLongitud)

% Analisis
Error_maximo = max(Distancias);
Error_medio = mean (Distancias);
Varianza = var(Distancias);
Percentil = prctile(Distancias, [5 50 95])

radius = max(Percentil(1));
az = [];
e = wgs84Ellipsoid;
[lat, lon] = scircle1(DatoRealLatitud, DatoRealLongitud, radius, az, e);
wmline(lat, lon, 'Color', 'green');

radius = max(Percentil(2));
az = [];
e = wgs84Ellipsoid;
[lat, lon] = scircle1(DatoRealLatitud, DatoRealLongitud, radius, az, e);
wmline(lat, lon, 'Color', 'yellow');

radius = max(Percentil(3));
az = [];
e = wgs84Ellipsoid;
[lat, lon] = scircle1(DatoRealLatitud, DatoRealLongitud, radius, az, e);
wmline(lat, lon, 'Color', 'red');

wmzoom(13);
wmcenter(DatoRealLatitud, DatoRealLongitud) %Centro de valladolid

toc

```

Figura 131 Código coeficiente de reflexión  $s_{11}$  (2/2)

## 8.3 Demostración del sistema: caso de uso

### 8.3.1 Main.m

```
%% MAIN CASO de USO
clc
clear
% Function 1: cargarDatos;
%Primeramente, cargamos y adecuamos los datos obtenidos en la prueba realizada.
[DatosGPS_A, DatosOBD_A, DatosGPS_B, DatosOBD_B, DatosGmaps, CAMS_Enviados_por_A,
CAMS_Enviados_por_B, CAMS_Recibidos_por_A, CAMS_Recibidos_por_B] = cargarDatos();

% Function 2: Gps;
%Calculamos los datos obtenidos del GPS.
[DistanciaGPSTotal_A, DistanciaAcumuladaGPS_A] = Gps(DatosGPS_A);
[DistanciaGPSTotal_B, DistanciaAcumuladaGPS_B] = Gps(DatosGPS_B);
fprintf('GPS\n');
fprintf('La distancia calculada por el GPS del ordenador A es: %f
kilometros\n', DistanciaGPSTotal_A);
fprintf('La distancia calculada por el GPS del ordenador B es: %f
kilometros\n\n\n', DistanciaGPSTotal_B);

%Function 3: OBD;
[DistanciaTotalODB_A, DistanciaAcumuladaODB_A] = Obd(DatosOBD_A, 1);
[DistanciaTotalODB_B, DistanciaAcumuladaODB_B] = Obd(DatosOBD_B, 2);

[DistanciaconLimiteODB_A] = OBDLimitado(DistanciaAcumuladaODB_A, DatosOBD_A, 1);
[DistanciaconLimiteODB_B] = OBDLimitado(DistanciaAcumuladaODB_B, DatosOBD_B, 2);

fprintf('OBD\n');
fprintf('La distancia calculada por el OBD del ordenador A es: %f
kilometros\n', DistanciaTotalODB_A);
fprintf('La distancia calculada por el OBD del ordenador B es: %f
kilometros\n\n\n', DistanciaTotalODB_B);

% %Funcion 4: velocidad del recorrido
velocidad(DatosOBD_A, 1);
velocidad(DatosOBD_B, 2);

% %Funcion 5: Ruta del gps realizada
ruta(DatosGPS_A, 1)
ruta(DatosGPS_B, 2)

%Function 6: GMaps
[DistanciaAcumuladaGmaps, DistanciaGPSTotalGmaps] = Gmaps(DatosGmaps);
fprintf('La distancia calculada por GMAPS es: %f
kilometros\n\n\n', DistanciaGPSTotalGmaps);
%
% %Function 7: CAMS
[Perdidos_A_to_B, Perdidos_B_to_A] = Cams(CAMS_Enviados_por_A, CAMS_Enviados_por_B,
CAMS_Recibidos_por_A, CAMS_Recibidos_por_B);
fprintf('Cams perdidos\n');
fprintf('Los cams perdidos que han sido enviados por A y recibidos por B son: %i
\n', Perdidos_A_to_B);
fprintf('Los cams perdidos que han sido enviados por B y recibidos por A son: %i
\n\n\n', Perdidos_B_to_A);

% %Function 8: calcular la posicion real del gps
[MatrizA] = CalculoPosicionGPS(DistanciaAcumuladaGmaps,
DistanciaAcumuladaODB_A, DatosGPS_A, DatosGmaps, DatosOBD_A, 0);
[MatrizB] = CalculoPosicionGPS(DistanciaAcumuladaGmaps,
DistanciaAcumuladaODB_B, DatosGPS_B, DatosGmaps, DatosOBD_B, 0);
[MatrizAReducida] = CalculoPosicionGPS(DistanciaAcumuladaGmaps,
DistanciaAcumuladaODB_A, DatosGPS_A, DatosGmaps, DatosOBD_A, 1);
[MatrizBReducida] = CalculoPosicionGPS(DistanciaAcumuladaGmaps,
DistanciaAcumuladaODB_B, DatosGPS_B, DatosGmaps, DatosOBD_B, 2);
```

Figura 132. Código función main (1/2)

```

ErrorA = max(MatrizA(:,3)); %Error maximo entre coordenadas reales y obtenidas
ErrorB = max(MatrizB(:,3));
ErrorAReducida = max(MatrizAReducida(:,3));
ErrorBReducida = max(MatrizBReducida(:,3));

fprintf ('Error maximo\n');
fprintf('El error maximo del ordenador A es: %f metros en los 3 primeros
kilometros\n',ErrorAReducida);
fprintf('El error maximo del ordenador B es: %f metros en los 3 primeros
kilometros\n\n',ErrorBReducida);

figure()
plot(MatrizAReducida(:,4)-MatrizAReducida(1,4),MatrizAReducida(:,3),'b');
hold on
plot(MatrizBReducida(:,4)-MatrizBReducida(1,4),MatrizBReducida(:,3), 'r');
hold off
title('Error en la posicion');
xlabel('Tiempo (segundos)');
ylabel ('Error (metros)');
legend('Coche del ordenador A', 'Coche del ordenador B')
set(gcf, 'Color', 'white')

MediaA = mean(MatrizA(:,3)); %Error medio entre coordenadas reales y obtenidas
MediaB = mean(MatrizB(:,3));
MediaBReducida = mean(MatrizBReducida(:,3));
MediaAReducida = mean(MatrizAReducida(:,3));
fprintf ('Error medio\n');
fprintf('El error medio del ordenador A es: %f metros en los 3 primeros
kilometros\n',MediaAReducida);
fprintf('El error medio del ordenador B es: %f metros en los 3 primeros
kilometros\n',MediaBReducida);

% %Function 9: calcular la distancia entre coches con los valores obtenidos
% %del GPS directamente
[ MatrizDistanciasGPS, MatrizDistanciasMedianaGPS ] = DistanciaEntreCochesGPS( DatosGPS_A,
DatosGPS_B, 0 );

%Function 10: calcular la distancia entre coches con los valores obtenidos
%de GMAPS, OBD y GPS
[ MatrizDistanciasGmapsObdGps, MatrizDistanciasMediana ] = DistanciaEntreCochesGPS(
MatrizA, MatrizB ,1);

Figuras( MatrizDistanciasGPS, MatrizDistanciasMedianaGPS,
MatrizDistanciasGmapsObdGps,MatrizDistanciasMediana )

```

*Figura 133. Código función main (2/2)*

### 8.3.2 cargarDatos.m

```
Function [DatosGPS_A, DatosOBD_A , DatosGPS_B, DatosOBD_B, DatosGmaps, CAMS_Enviados_por_A
,CAMS_Enviados_por_B, CAMS_Recibidos_por_A,CAMS_Recibidos_por_B ] = cargarDatos( )

%GPS
%Para que los datos coincidan en tiempo hay que eliminar los 8 primeros del B
%y los ultimos 48 del B
DatosGPS_A = xlsread('A1_stats_GPS_2019-04-03_17_24.xlsx'); %Hay que cambiar los puntos por
comas en el libro original
DatosGPS_B = xlsread('B1_stats_GPS_2019-03-27_17_23.xlsx');
DatosGPS_B(:,4) = DatosGPS_B(:,4) + 6.048e+14; % !!!!!!! OJO Esta sentencia se ha
utilizado para poner en el mismo dia la
                                % grabacion del ordenador B, ya que ese
ordenador ha creido que grababa el
                                % dia 27/03/2017
DatosGPS_A(:,4)=DatosGPS_A(:,4)/1e9; %de ns a seg
DatosGPS_B(:,4)=DatosGPS_B(:,4)/1e9;

%GPS con Gmaps
DatosGmaps = xlsread('Ruta Caso de Uso.xlsx'); %Cargamos el excel

%OBD
%Primer paso: abrir el excel, seleecioanr la columna y dar Datos--> Texto
%en columnas --> Delimitados --> Siguiete --> TAbulacion --> Siguiete --> Cambiar
separador decimal . y miles ,
%--> Finalizar--> Columna de tiempo --> Numero
DatosOBD_A = xlsread('A1_stats_Obd2Service_2019-04-03_17_24.xlsx'); %Cargamos el excel
DatosOBD_B = xlsread('B1_stats_Obd2Service_2019-03-27_17_23.xlsx');
DatosOBD_B(:,2) = DatosOBD_B(:,2) + 6.048e+14; % !!!!!!! OJO Esta sentencia se ha utilizado
para poner en el mismo dia la
                                % grabacion del ordenador B, ya que ese
ordenador ha creido que grababa el
                                % dia 27/03/2017

%CAMS
CAMS_Enviados_por_A = fileread('Cams_enviados_por_A');
CAMS_Enviados_por_B = fileread('Cams_enviados_por_B');
CAMS_Recibidos_por_A = fileread('Cams_recibidos_por_A');
CAMS_Recibidos_por_B = fileread('Cams_recibidos_por_B');

end
```

Figura 134. Código función cargarDatos















### 8.3.9 CalculoPosicionGPS.m

```
function [ MatrizOBDGPS ] = CalculoPosicionGPS( DistanciaAcumuladaGmaps,
DistanciaAcumuladaOBD, DatosGPS, DatosGmaps, DatosOBD, color)

if(color == 1)
    LimiteSuperiorA = find (DistanciaAcumuladaOBD(1:end,1) == 3.003545806267645);
    DistanciaAcumuladaOBD = DistanciaAcumuladaOBD(1:LimiteSuperiorA,:);
end
if (color == 2)
    LimiteSuperiorB = find (DistanciaAcumuladaOBD(1:end,1) == 3.000995134773562);
    DistanciaAcumuladaOBD = DistanciaAcumuladaOBD(1:LimiteSuperiorB,:);
end

% Relacion GMAPS - OBD
AD = diff(DistanciaAcumuladaGmaps); %Incremento de distancias de GMAPS y OBD
Ad = diff(DistanciaAcumuladaOBD(:,1));

[m,n]=size(DistanciaAcumuladaOBD);
punto = zeros(m,2);
MatrizOBDGPMAS = zeros(m-1,4);
R = 6378.137; % Radius of earth in KM
DatosOBD(:,1) = DatosOBD(:,1) ./ 1e2; %Velocidad m/s

for i = 1:m-1

    dist=(DistanciaAcumuladaGmaps - DistanciaAcumuladaOBD(i));
    punto(i,1) = find(abs(dist)== min(abs(dist)));
    punto(i,2) = punto(i,1) - sign(dist(punto(i,1)));

    LatOBD = ((DatosGmaps(punto(i,2),1) -
DatosGmaps(punto(i,1),1))*Ad(i,1)+DatosGmaps(punto(i,1),1)*AD(punto(i,1),1))/AD(punto(i,
1));
    LonOBD = ((DatosGmaps(punto(i,2),2) -
DatosGmaps(punto(i,1),2))*Ad(i,1)+DatosGmaps(punto(i,1),2)*AD(punto(i,1),1))/AD(punto(i,
1));

    MatrizOBDGPMAS(i,1) = LatOBD;
    MatrizOBDGPMAS(i,2) = LonOBD;
    MatrizOBDGPMAS(i,3) = DistanciaAcumuladaOBD(i,2);

    % Columna1 --> latitud punto 1
    % Columna2 --> longitud punto 1
    % Columna3 --> timestamp
end

%Relacion OBD&GMAPS con GPS

[k,l]=size(DatosGPS);
[x,z]=size(MatrizOBDGPMAS);

DatosOBD(:,2) = DatosOBD(:,2) ./10^9; % Tiempo de ns a seg
puntoGPS = zeros(k,2);
% aceleracion = zeros(m,1);
% aceleracion = ((DatosOBD_A(2:m,1) - DatosOBD_A(1:m-1,1))./(DatosOBD_A(2:m,2) -
DatosOBD_A(1:m-1,2)));

Error = zeros(k-1,1);
MatrizOBDGPS = zeros(k-1,3);
```

Figura 141. Código función CalculoPosicionGPS (1/3)

```

for i = 1:k-1

    dist=(MatrizOBDGPMAS(:,3) - DatosGPS(i,4));
    puntoGPS(i,1) = find(abs(dist)== min(abs(dist)));
    puntoGPS(i,2) = puntoGPS(i,1) - sign(dist(puntoGPS(i,1)));

    if ( puntoGPS(i,1) > 0 && puntoGPS(i,2) > 0 && puntoGPS(i,1) < x && puntoGPS(i,2) < x ||
color == 0 && puntoGPS(i,1) > 0 && puntoGPS(i,2) > 0)

        if (puntoGPS(i,2) < puntoGPS(i,1) ) % Ordenado de menor a mayor
            aux = puntoGPS(i,1);
            puntoGPS(i,1) = puntoGPS(i,2);
            puntoGPS(i,2) = aux;
        end

        aceleracion = ((DatosOBD(puntoGPS(i,2),1) -
DatosOBD(puntoGPS(i,1),1))./(DatosOBD(puntoGPS(i,2),2) - DatosOBD(puntoGPS(i,1),2))); %
aceleracion = (v2 - v1) / (t2 - t1)
        Tiempo = DatosGPS(i,4) - DatosOBD(puntoGPS(i,1),2);
        Ad = 0.5 * aceleracion.* (Tiempo.^2) + DatosOBD(puntoGPS(i,1),1) .* Tiempo;

        if(Ad == 0 )
            LatOBDGPS = MatrizOBDGPMAS(puntoGPS(i,1),1);
            LonOBDGPS = MatrizOBDGPMAS(puntoGPS(i,1),2);
        else
            %Distancia entre dos puntos de coordenadas

            dLat = (MatrizOBDGPMAS(puntoGPS(i,2),1) .* pi /180) -
(MatrizOBDGPMAS(puntoGPS(i,1),1) .* pi / 180);
            dLon = (MatrizOBDGPMAS(puntoGPS(i,2),2) .* pi / 180) -
(MatrizOBDGPMAS(puntoGPS(i,1),2) .* pi / 180);

            a = (sin(dLat/2) .* sin(dLat/2)) + (cos(MatrizOBDGPMAS(puntoGPS(i,1),1) .* pi /
180) .* cos(MatrizOBDGPMAS(puntoGPS(i,2),1) .* pi / 180) .* sin(dLon/2) .* sin(dLon/2));
            c = 2 * atan2(sqrt(a), sqrt(1-a));
            AD = R * c * 1e5; %Distancia entre los puntos extremos del OBD y GMAPS

            if(Ad > AD)
                LatOBDGPS = MatrizOBDGPMAS(puntoGPS(i,2),1);
                LonOBDGPS = MatrizOBDGPMAS(puntoGPS(i,2),2);
            end

            if(AD ~= 0 && Ad < AD)
                LatOBDGPS = ((MatrizOBDGPMAS(puntoGPS(i,2),1) -
MatrizOBDGPMAS(puntoGPS(i,1),1))*Ad + MatrizOBDGPMAS(puntoGPS(i,1),1)*AD)/AD;
                LonOBDGPS = ((MatrizOBDGPMAS(puntoGPS(i,2),2) -
MatrizOBDGPMAS(puntoGPS(i,1),2))*Ad + MatrizOBDGPMAS(puntoGPS(i,1),2)*AD)/AD;
            else
                LatOBDGPS = MatrizOBDGPMAS(puntoGPS(i,1),1);
                LonOBDGPS = MatrizOBDGPMAS(puntoGPS(i,1),2);
            end
        end

        end

        % Calcular el error entre los datos del GPS obtenidos y los datos GPS reales (Gmaps
- Obd)

        dLat = (LatOBDGPS .* pi /180) - (DatosGPS(i,1) .* pi / 180);
        dLon = (LonOBDGPS .* pi / 180) - (DatosGPS(i,2) .* pi / 180);

        a = (sin(dLat/2) .* sin(dLat/2)) + (cos(DatosGPS(i,1) .* pi / 180) .* cos(LatOBDGPS
.* pi / 180) .* sin(dLon/2) .* sin(dLon/2));
        c = 2 * atan2(sqrt(a), sqrt(1-a));
        Error(i,1) = R * c * 1e3; %Error final
        Errorr = R * c * 1e3;
    end
end

```

Figura 142. Código función CalculoPosicionGPS (2/3)

```

    MatrizOBDGPS(i,1) = LatOBDGPS; % Latitud real
    MatrizOBDGPS(i,2) = LonOBDGPS; % Longitud real
    MatrizOBDGPS(i,3) = Error(i,1); % Error coordenadas reales y coordeandas
GPS
    MatrizOBDGPS(i,4) = DatosGPS(i,4); %Tiempo GPS

    end

end

if(color == 1 || color == 2) %Quitamos todos los 0 finales de la matriz reducida
    Ceros = find(diff(find(MatrizOBDGPS(:,1)==0))>1);
    Posicion = find(MatrizOBDGPS(:,1)==0);
    MatrizOBDGPS = MatrizOBDGPS(Posicion(Ceros)+1:Posicion(Ceros+1)-1,:);
End
end

```

*Figura 143. Código función CalculoPosicionGPS (3/3)*



### 8.3.10 OBDLimitado.m

```
function [ DistanciaconLimite ] = OBDLimitado( DistanciaAcumulada,DatosOBD, color )

if (color == 1)

    figure(2)
    tiempoA = cumsum(diff(DatosOBD(:,2)/1e9));
    LimiteSuperiorA = find (DistanciaAcumulada(1:end,1) == 3.003545806267645); % 7
    km == 6.916854826014735    3km == 3.003545806267645
    DistanciaconLimite = DistanciaAcumulada(1500:LimiteSuperiorA,:);
    plot(DistanciaconLimite(:,2)-DistanciaconLimite(1,2),DistanciaconLimite(:,1));
    hold on;

else
    %%Fiabilidad del OBD B
    figure(2);
    tiempoB = cumsum(diff(DatosOBD(:,2)/1e9));
    LimiteSuperiorB = find (DistanciaAcumulada(1:end,1) == 3.000995134773562); %7 km
    == 7.005544424131096    3km == 3.000995134773562
    DistanciaconLimite = DistanciaAcumulada(1714:LimiteSuperiorB,:);
    plot(DistanciaconLimite(:,2)-DistanciaconLimite(1,2),DistanciaconLimite(:,1),
'r');
    set(gcf, 'Color', 'white')
    title('Distancias acumuladas del OBD');
    xlabel('Tiempo de la prueba (segundos)');
    ylabel ('Distancia (kilometros)');
    legend('Coche del ordenador A', 'Coche del ordenador B')
    hold off;

end
end
```

Figura 144. Código función OBDLimitado

### 8.3.11 Script DistanciaEntreCochesGPS.m.

```
function [ ] = Figuras( MatrizDistanciasGPS, MatrizDistanciasMedianaGPS,
MatrizDistanciasGmapsObdGps,MatrizDistanciasMediana )

% Figura con la distancia GPS directamente
figure()           %Enseñar a JC
plot(MatrizDistanciasGPS(:,1),'b');
title('Distancia entre coches sin mediana');
xlabel('Tiempo (segundos)');
ylabel ('Distancia (metros)');
set(gcf, 'Color', 'white')
hold on

plot(MatrizDistanciasGmapsObdGps(:,1),'r');
legend('GPS', 'Gmaps OBD')

% Figura con la distancia GPS aplicada la mediana
figure()
plot(MatrizDistanciasMedianaGPS(:,1),'b');
title('Distancia entre coches con mediana');
xlabel('Tiempo (segundos)');
ylabel ('Distancia (metros)');
set(gcf, 'Color', 'white')
hold on

plot(MatrizDistanciasMediana(:,1),'r');
legend('GPS', 'Gmaps OBD')

% Figura con el error entre el GPS y la mediana
figure()
plot(MatrizDistanciasGPS(:,1)-MatrizDistanciasMedianaGPS(:,1),'b');
title('Error dado por el GPS');
xlabel('Tiempo (segundos)');
ylabel ('Distancia (metros)');
set(gcf, 'Color', 'white')
hold on

% Figura con el error entre el GPS y OBD y la mediana
figure()
plot(MatrizDistanciasGmapsObdGps(:,1)-MatrizDistanciasMediana(:,1),'r');
title('Error dado por el GPS&OBD');
xlabel('Tiempo (segundos)');
ylabel ('Distancia (metros)');
set(gcf, 'Color', 'white')
hold on

legend('Gmaps OBD')

end
```

Figura 145. Código función Figuras

### 8.3.12 DistanciaEntreCocheGPS.m

```

function [ MatrizDistancias, MatrizDistanciasMediana ] = DistanciaEntreCocheGPS (
DatosGPS_A, DatosGPS_B, color )

if (color == 1) %Elmiminamos los 1° zeros del array

    DatosGPS_A = DatosGPS_A(8:end,:);
    DatosGPS_B = DatosGPS_B(8:end,:);

end

[m,n] = size(DatosGPS_B);
[x,z] = size(DatosGPS_A);

MatrizDistancias = zeros(m-1,3);
R = 6378.137; % Radius of earth in KM

for i = 1:m-1

    dist=(DatosGPS_A(:,4) - DatosGPS_B(i,4)); %Segundos
    punto(i,1) = find(abs(dist)== min(abs(dist)));
    punto(i,2) = punto(i,1) - sign(dist(punto(i,1))); %Par de puntos del GPS A entre
los que esta el dato del GPS B
    minimo(i)=min(abs(dist));
    if ( punto(i,2) > 0 && punto(i,2) < x+1 )

        if (punto(i,2) < punto(i,1) ) % Ordenado de menor a mayor
            aux = punto(i,1);
            punto(i,1) = punto(i,2);
            punto(i,2) = aux;
        end

        %
        dLat = (DatosGPS_A(punto(i,2),1) .* pi /180) - (DatosGPS_B(i,1) .* pi /
180);
        %
        dLon = (DatosGPS_A(punto(i,2),2) .* pi / 180) - (DatosGPS_B(i,2) .* pi /
180);
        %
        a = (sin(dLat/2) .* sin(dLat/2)) + ((cos(DatosGPS_B(i,1)) .* pi / 180) .*
(cos(DatosGPS_A(punto(i,2),1)) .* pi / 180) .* sin(dLon/2) .* sin(dLon/2));
        %
        c = 2 * atan2(sqrt(a), sqrt(1-a));
        %
        d = R * c * 1e5; %Distancia entre los puntos extremos del GPS en el punto
1 y el GPS en el punto 2

        dLat = (DatosGPS_A(punto(i,2),1) .* pi /180) - (DatosGPS_A(punto(i,1),1) .*
pi / 180);
        dLon = (DatosGPS_A(punto(i,2),2) .* pi / 180) - (DatosGPS_A(punto(i,1),2) .*
pi / 180);

        a = (sin(dLat/2) .* sin(dLat/2)) + (cos(DatosGPS_A(punto(i,1),1) .* pi / 180)
.* cos(DatosGPS_A(punto(i,2),1) .* pi / 180) .* sin(dLon/2) .* sin(dLon/2));
        c = 2 * atan2(sqrt(a), sqrt(1-a));
        d = R * c * 1e5; %Distancia entre los puntos extremos del GPS en el punto 1
y el GPS en el punto 2
    end
end

```

Figura 146. Código función DistanciaEntreCocheGPS (1/2)

```

D = d * ((DatosGPS_B(i,4)-
DatosGPS_A(punto(i,1),4)))/(DatosGPS_A(punto(i,2),4)-DatosGPS_A(punto(i,1),4));

if (d == 0)
    LatA = DatosGPS_A(punto(i,1),1);
    LonA = DatosGPS_A(punto(i,1),2);
else
    LatA = (DatosGPS_A(punto(i,2),1)-DatosGPS_A(punto(i,1),1))* D/d +
DatosGPS_A(punto(i,1),1); % Lon2' = ((Lon2 - Lon1) * d + lon1 * D)/D
    LonA = (DatosGPS_A(punto(i,2),2)-DatosGPS_A(punto(i,1),2))*D/d +
DatosGPS_A(punto(i,1),2);
end

% Calculo de la distancia
Coordenada(i,1) = LatA;
Coordenada(i,2) = LonA;

dLat = (LatA .* pi / 180) - (DatosGPS_B(i,1) .* pi / 180);
dLon = (LonA .* pi / 180) - (DatosGPS_B(i,2) .* pi / 180);

a = (sin(dLat/2) .* sin(dLat/2)) + (cos(DatosGPS_B(i,1) .* pi / 180) .*
cos(LatA .* pi / 180) .* sin(dLon/2) .* sin(dLon/2));
c = 2 * atan2(sqrt(a), sqrt(1-a));
d = R * c * 1e3; %Error final en metros

MatrizDistancias(i,1) = d; % Columnal --> distancia entre vehiculos
MatrizDistancias(i,2) = DatosGPS_B(i,1); % Columnal --> distancia entre
vehiculos
MatrizDistancias(i,3) = DatosGPS_B(i,2); % Columnal --> distancia entre
vehiculos
end
end

MatrizDistanciasMediana = MatrizDistancias;
for i=2:m-2
    MatrizDistanciasMediana(i,1)=median(MatrizDistancias(i-1:i+1,1));
end
end
end

```

Figura 147. Código función *DistanciaEntreCochesGPS (2/2)*

# 9. Datasheet antenna plana

## GSA.8859



### SPECIFICATION

Part No.	:	<b>GSA.8859.A.105111</b>
Product Name	:	4dBi Adhesive Mini DSRC 5.9GHz Antenna 1M CFD200 with SMA(M) Connector
Feature	:	Omnidirectional For V2V and V2X Applications High Efficiency and High Peak Gain IP67 Rating Adhesive Mount on Plastic or Glass Low Loss 1M CFD200 Cable SMA(M) ST Connector Customizable Cable Type, Length, Connector Dims: 36*30*10mm RoHS compliant





## 1. Introduction

The GSA.8859 Mini DSRC antenna is an external adhesive mount antenna that operates from 5850-5925MHz for DSRC systems.

DSRC (Dedicated Short Range Communications) is the communications medium of choice for active safety V2V (Vehicle-to-Vehicle) and V2X (Vehicle-to-Other) systems, primarily allocated for vehicle safety applications. DSRC supports high speed, low latency, short-range V2V/V2X wireless communications. The GSA.8859 at only 10mm comes in a very compact size enabling flexibility of integration. It can be mounted on glass or plastic surfaces easily with the double-sided adhesive. The antenna features high peak gain at 4.14 dBi on glass and 3.24 dBi on 2mm thick plastic.

Contact your regional Taoglas office for support to integrate and test this antenna's performance in your device.

## 2. Specification

WI-FI		
Frequency	5850~5925MHz	
Efficiency (%)		
In free space	0.3m	80.23
	1m	68.30
	2m	54.24
	3m	44.09
	5m	28.26
On glass	0.3m	72.05
	1m	61.33
	2m	48.71
	3m	39.59
	5m	25.38
On the 2mm ABS	0.3m	78.34
	1m	66.67
	2m	52.96
	3m	43.05
	5m	27.60
Average Gain (dBi)		
In free space	0.3m	-0.96
	1m	-1.66
	2m	-2.66
	3m	-3.56
	5m	-5.49
On glass	0.3m	-1.42
	1m	-2.12
	2m	-3.12
	3m	-4.02
	5m	-5.96
On the 2mm ABS	0.3m	-1.06
	1m	-1.76
	2m	-2.76
	3m	-3.66
	5m	-5.59



Peak Gain (dBi)		
In free space	0.3m	3.27
	1m	2.57
	2m	1.57
	3m	0.67
	5m	-1.30
On glass	0.3m	4.84
	1m	4.14
	2m	3.14
	3m	2.24
	5m	0.34
On the 2mm ABS	0.3m	3.94
	1m	3.24
	2m	2.24
	3m	1.34
	5m	-0.66
Return loss	<-10	
VSWR	<2	
Impedance	50	
Polarization	Linear	
Radiation Pattern	Omnidirectional	
Input Power	5W	
MECHANICAL		
Dimensions	36*30*10mm	
Casing	PP	
Connector	SMA(M) ST, fully customizable	
Cable	1M CFD200, fully customizable	
Waterproof	IP67	
Weight	42g	
ENVIRONMENTAL		
Temperature Range	-40°C to 85°C	
Humidity	Non-condensino 65°C 95% RH	

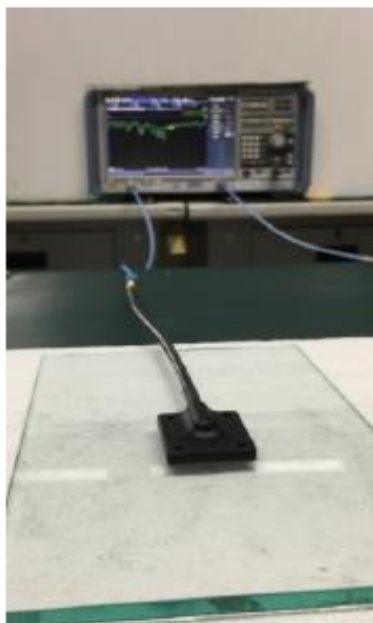


### 3. Antenna Characteristics

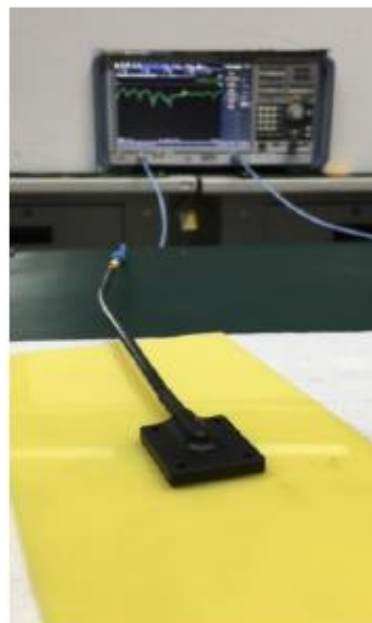
#### 3.1 Antenna Test Setup



Free Space

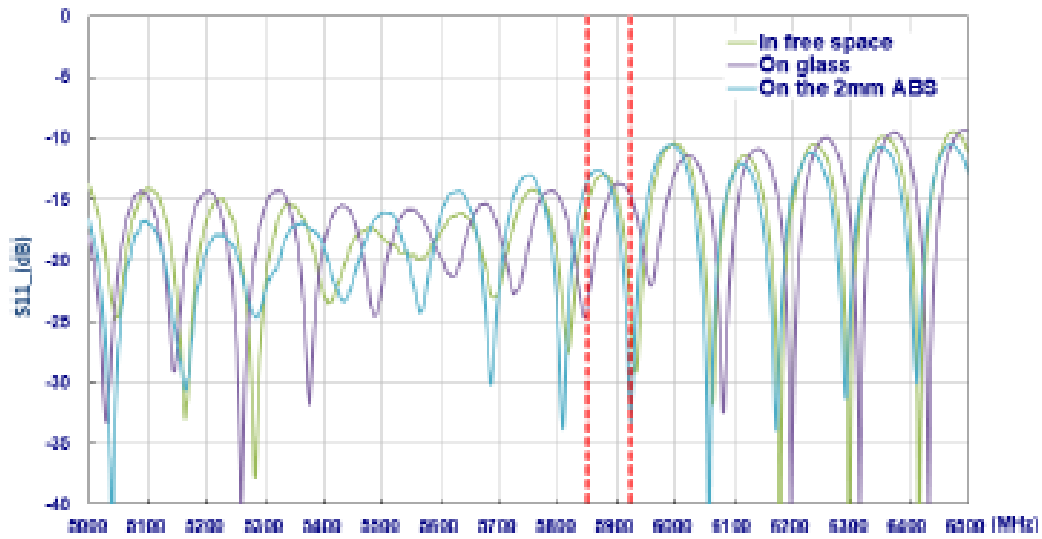


On Glass

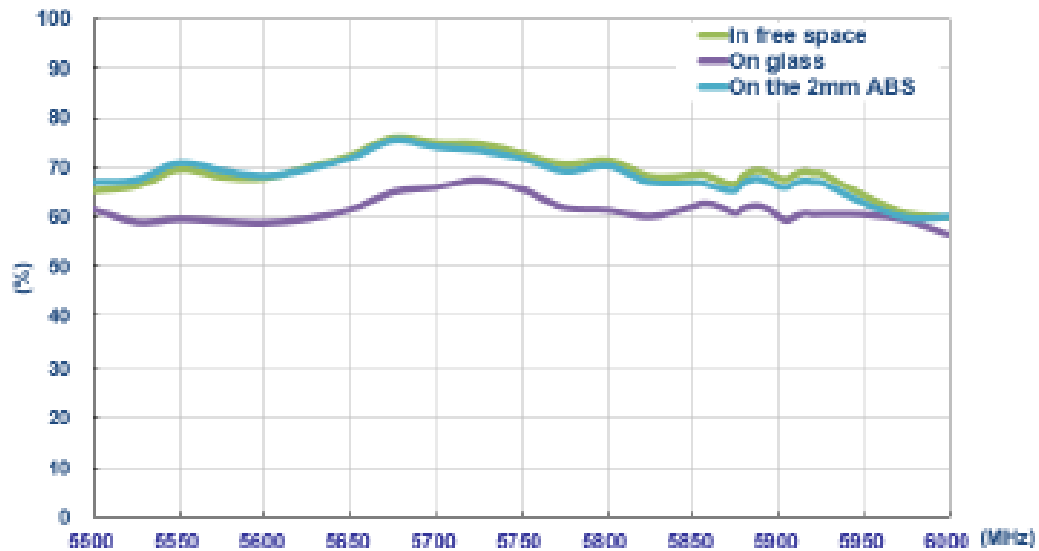


On 2mm ABS

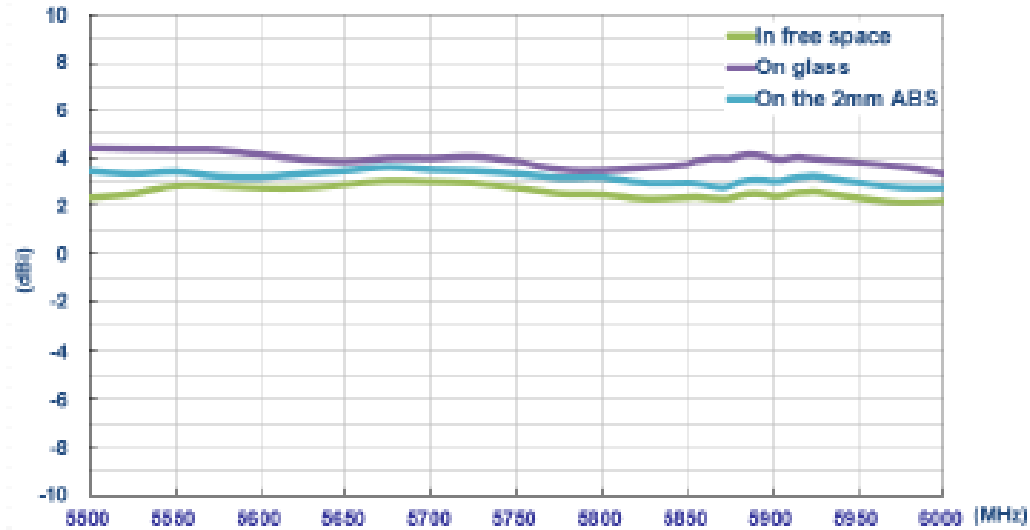
### 3.2 Return Loss



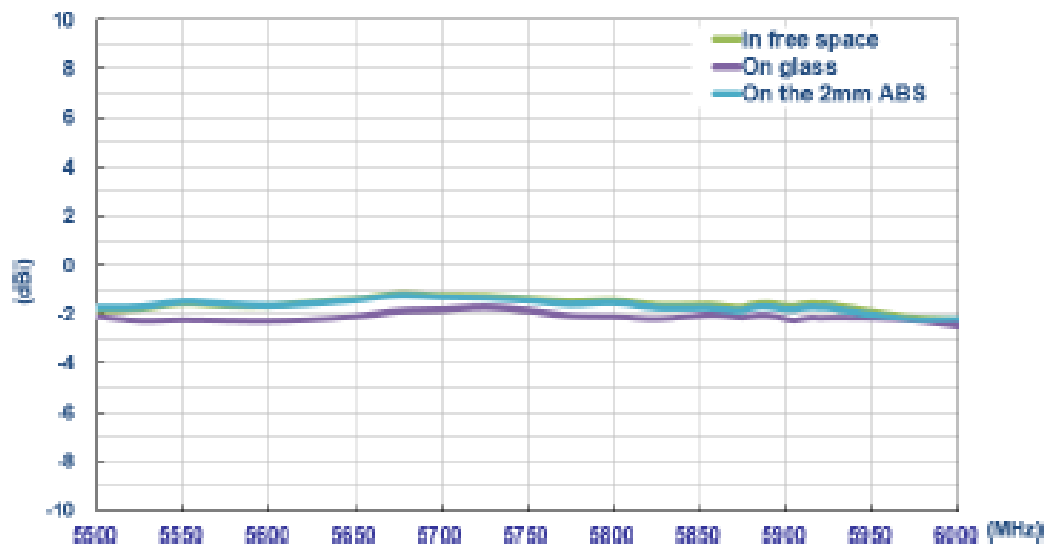
### 3.3 Efficiency



### 3.4 Peak Gain

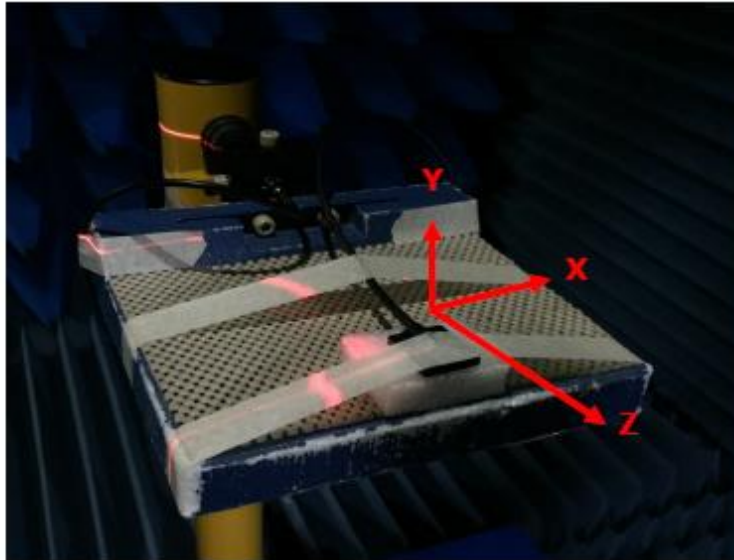


### 3.5 Average Gain

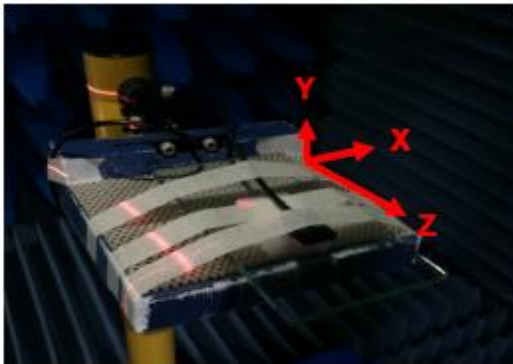


## 4. Antenna Radiation Patterns

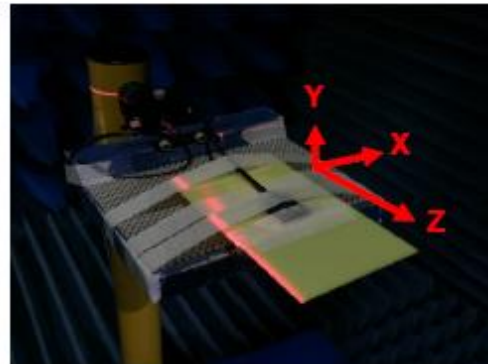
### 4.1 Antenna setup (Free space with 1m cable)



Free Space



On Glass



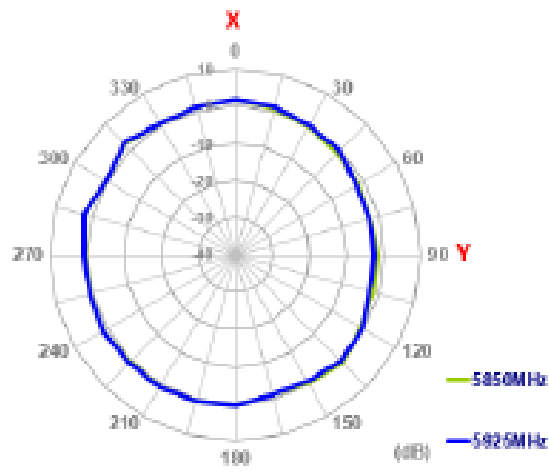
On 2mm ABS

### Antenna testing Setup in ETS Anechoic Chamber

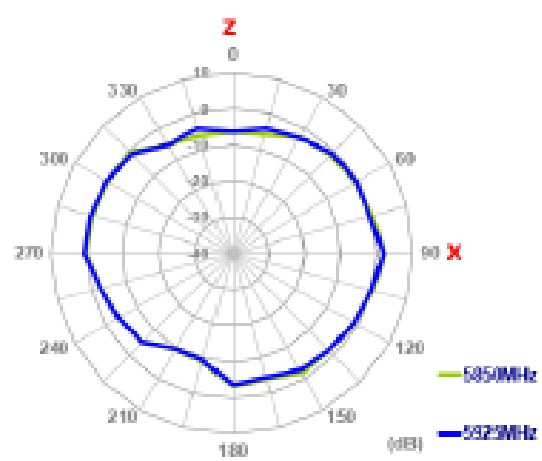
## 4.2 2D Radiation Patterns

### 4.2.1 In Free Space

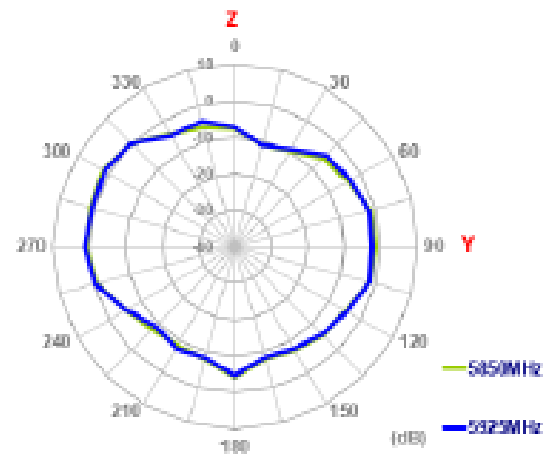
#### XY Plane



#### XZ Plane

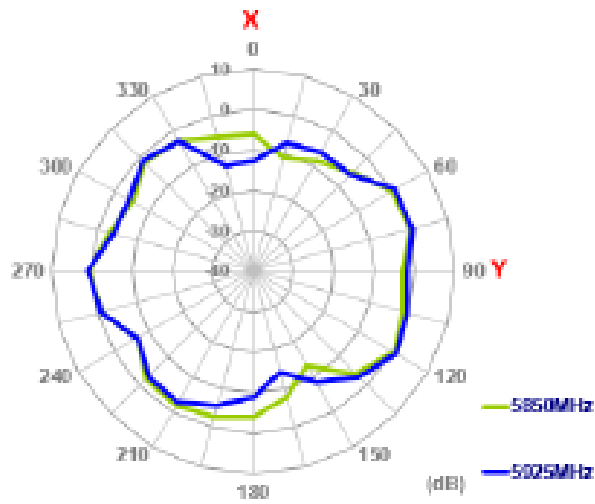


#### YZ Plane

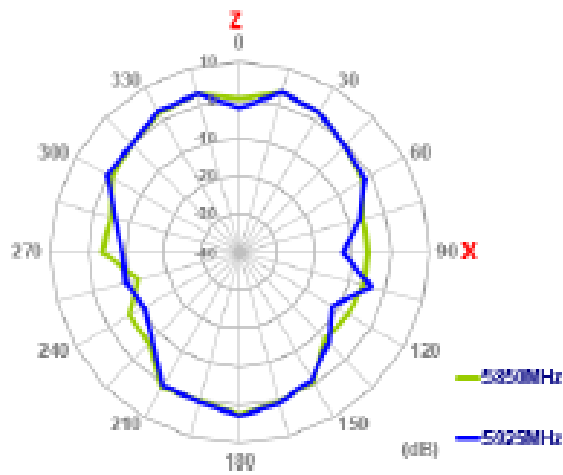


### 4.2.2 On Glass

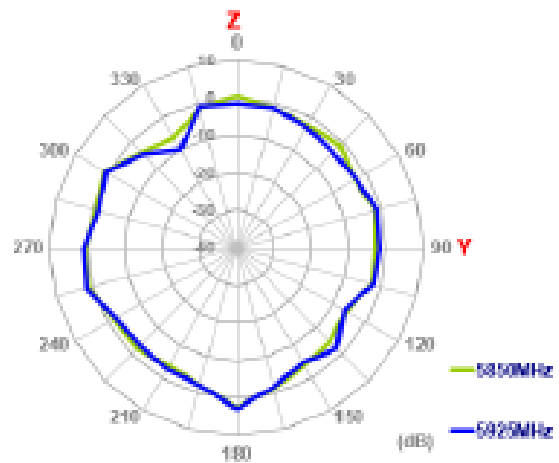
#### XY Plane



#### XZ Plane

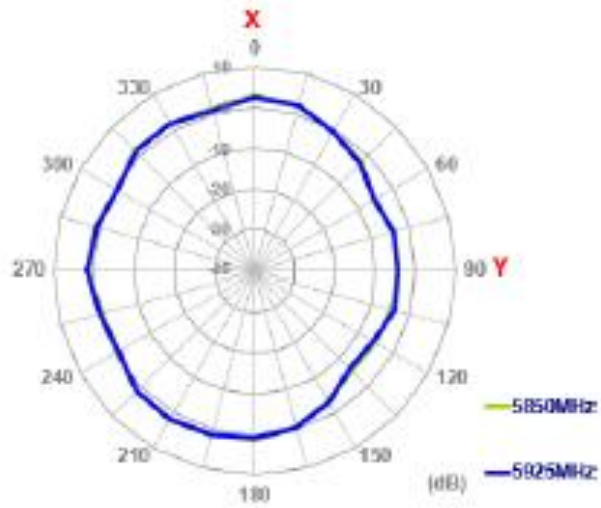


#### YZ Plane

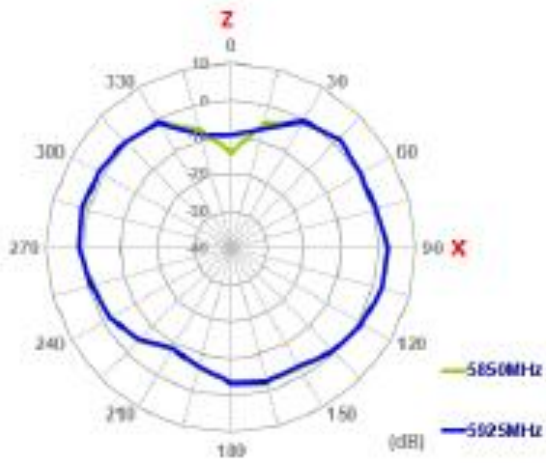


### 4.2.3 On 2mm ABS

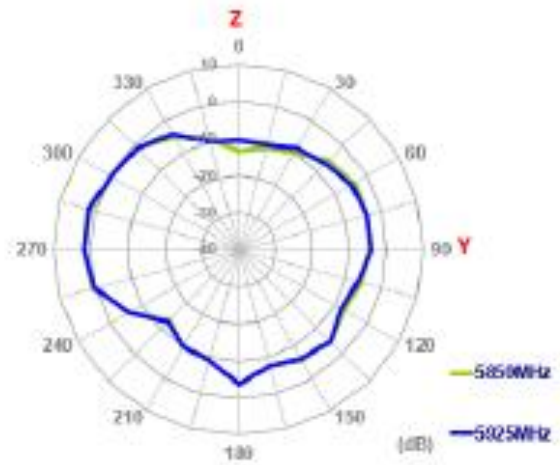
#### XY Plane



#### XZ Plane

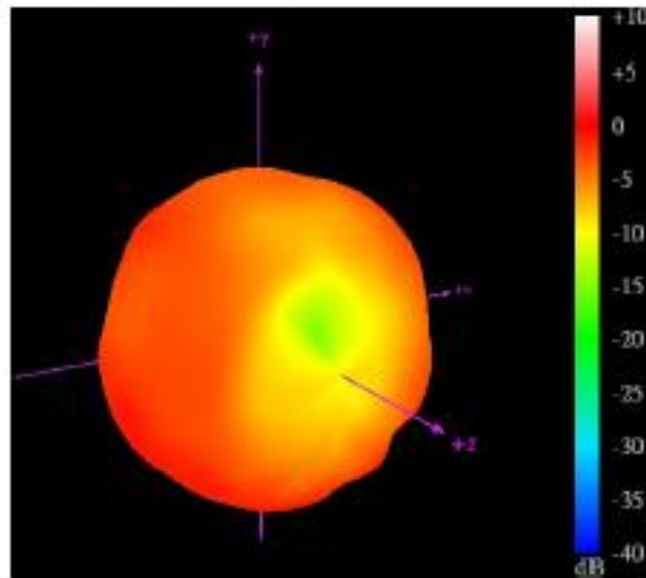


#### YZ Plane

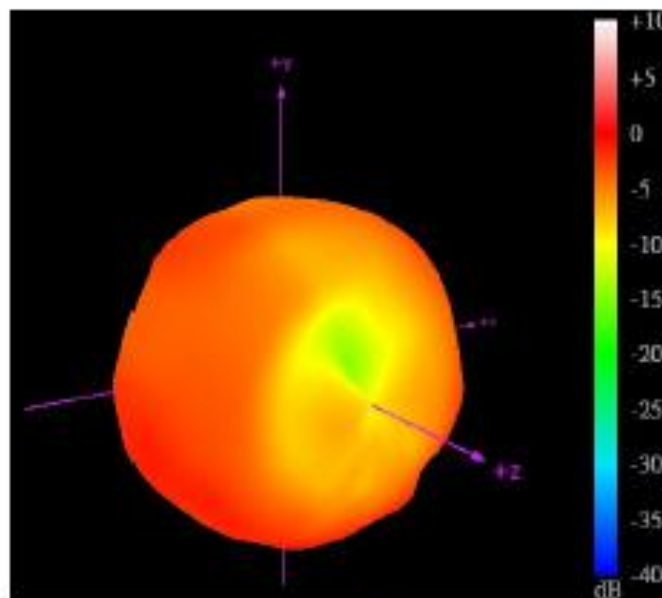


### 4.3 Antenna 3D Radiation Pattern

#### 4.3.1 In Free Space



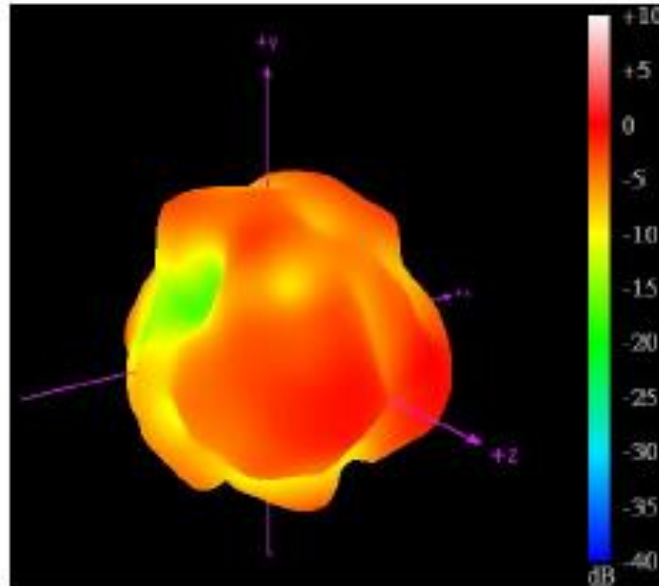
5850MHz



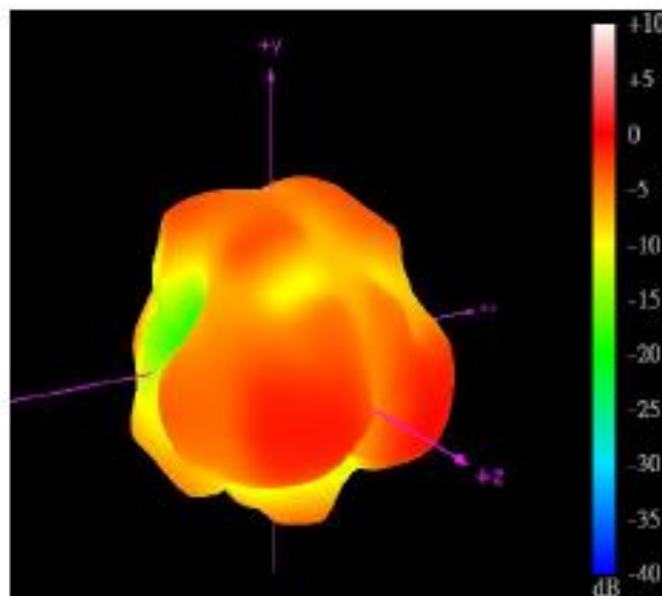
5925MHz



### 4.3.2 On Glass

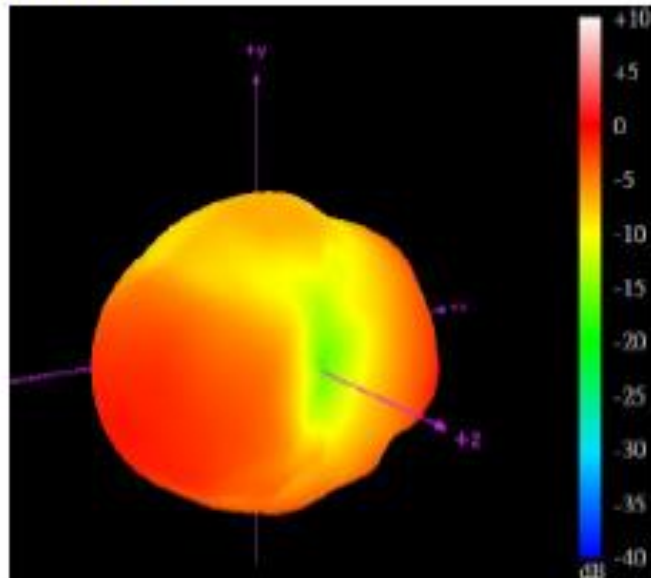


5850MHz

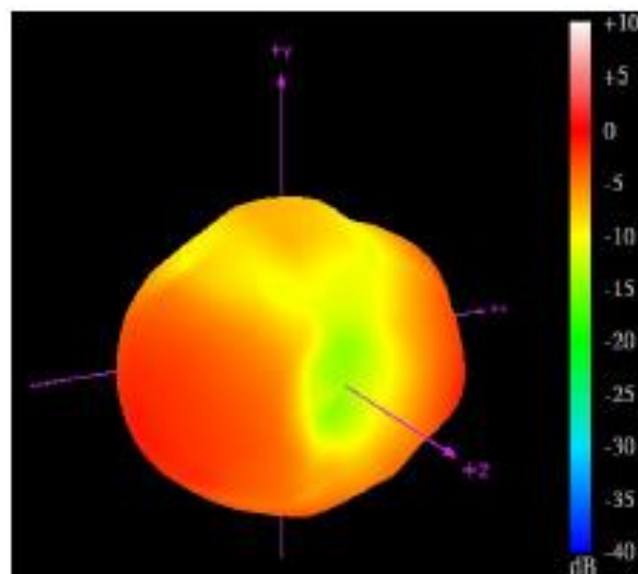


5925MHz

### 4.3.3 On 2mm ABS

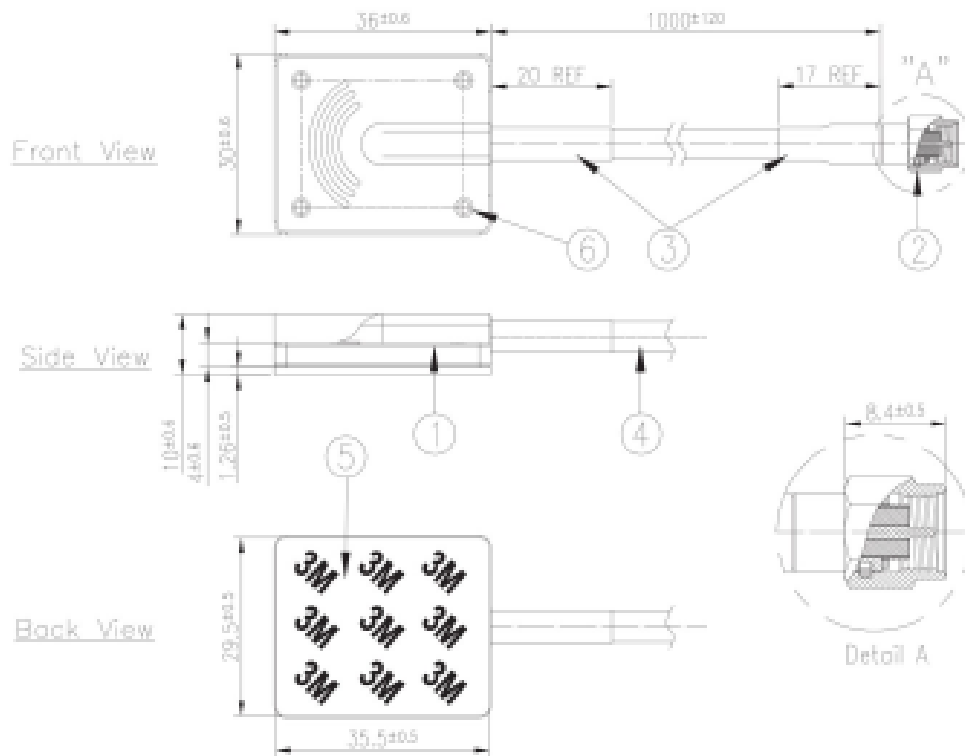


5850MHz



5925MHz

## 5. Drawing (Unit: mm)



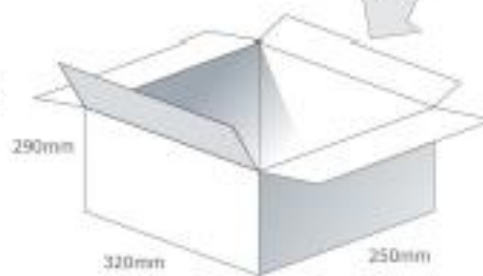
	Name	Material	Finish	Qty
1	CSA-8054 Housing	PP	Black	1
2	SMA(M)ST	Brass	As Plated	1
3	Heat Shrink Tube	PE	Black	2
4	CR0208 Coaxial Cable	PE	Black	1
5	Double-Side Adhesive With Dry foam	VHB 4041 1.261	White Linear	1
6	CSA-8054 PCB	FR4 1.61	Black	1

## 6. Packaging

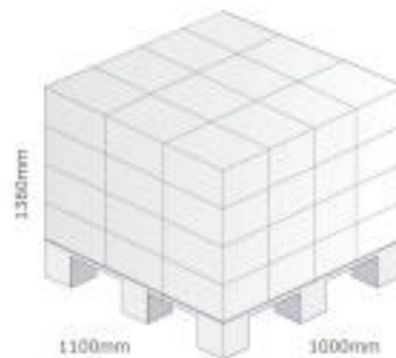
1pc GSA.8859.A.105111 per Small PE Bag  
 Bag Dimensions - 160\*300 mm  
 Weight - 50g



50pcs GSA.8859.A.105111 per Carton  
 Carton Dimensions - 320\*250\*290mm  
 Weight - 2.6Kg



Pallet Dimensions:  
 1100mm\*1000mm\*1360mm  
 48 Cartons per Pallet  
 12 Cartons per Layer, 4 Layers

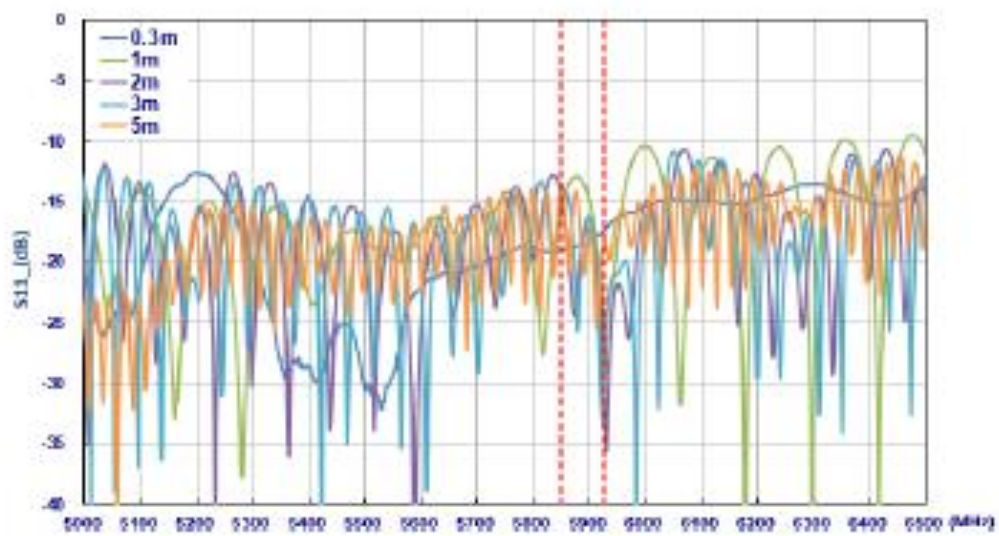


## 7. Application Note

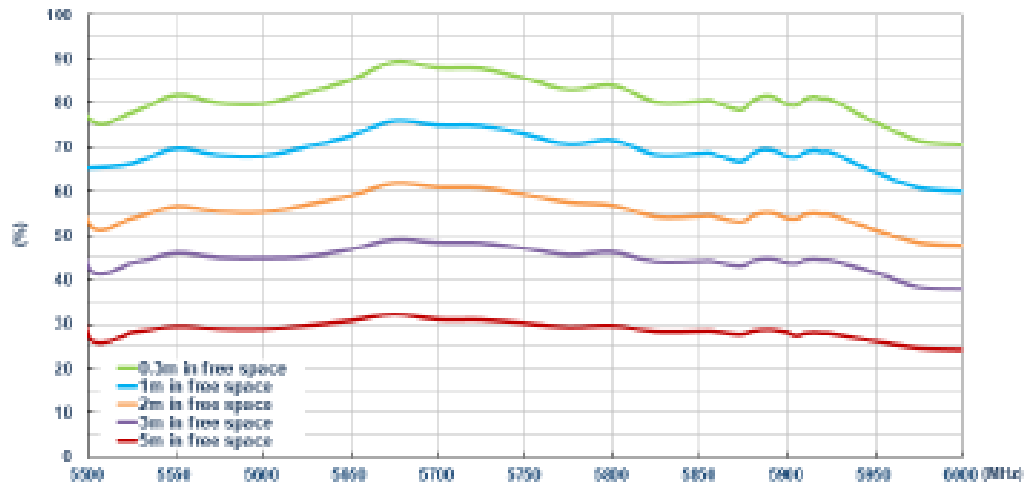
The WSA.2458 antenna performance with different cable lengths is shown below.

### 7.1 In free Space

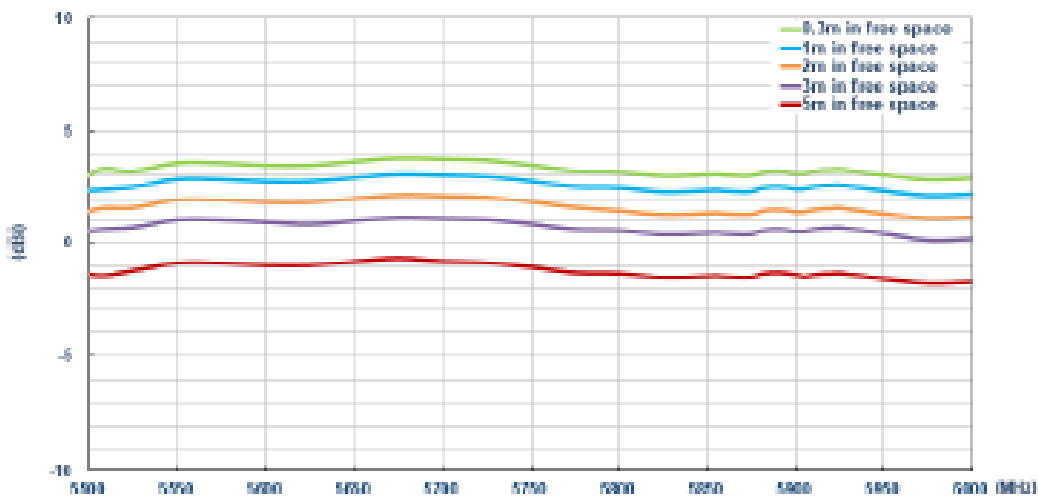
#### 7.1.1 Return Loss



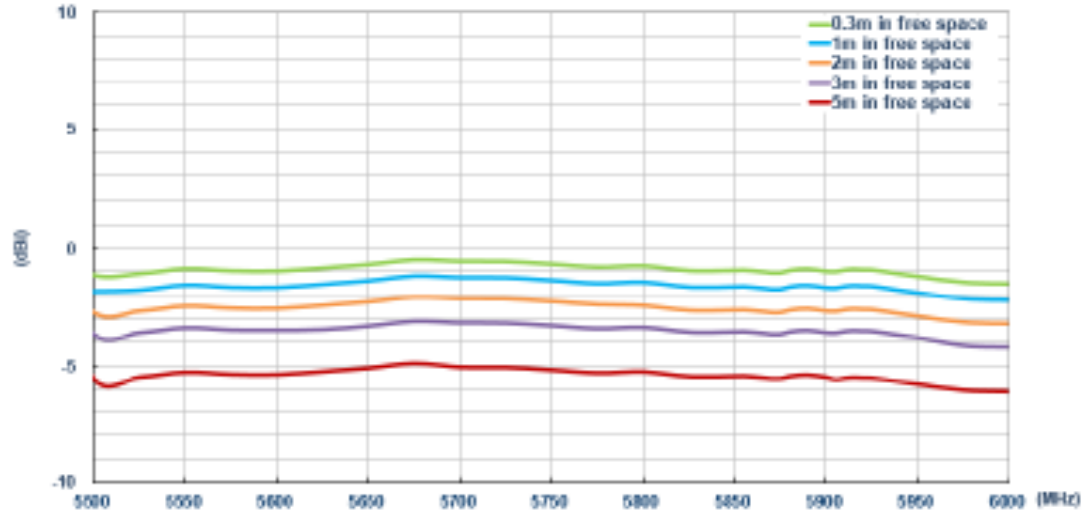
### 7.1.2 Efficiency



### 7.1.3 Peak Gain

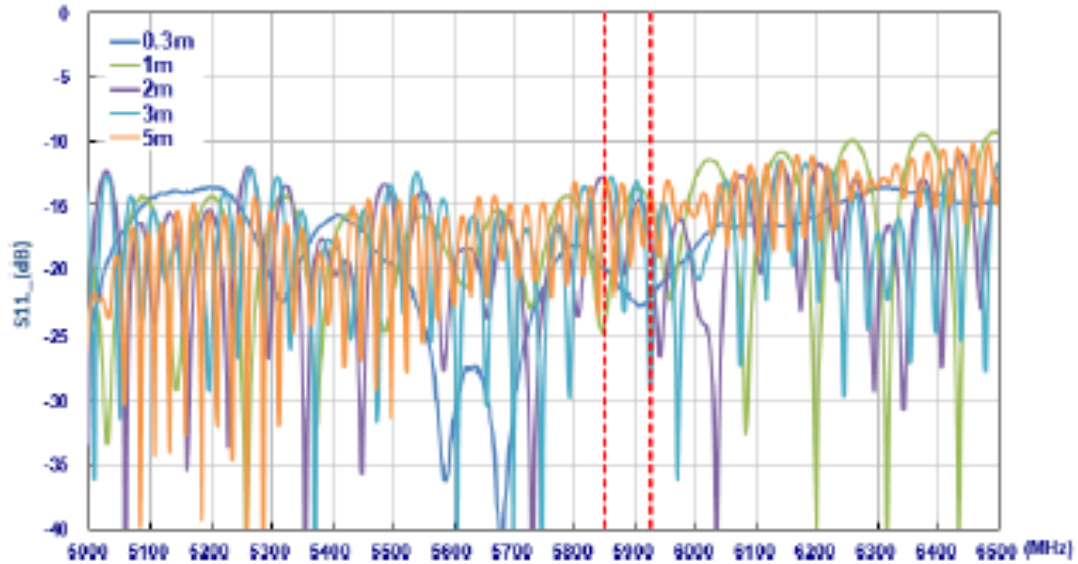


### 7.1.4 Average Gain

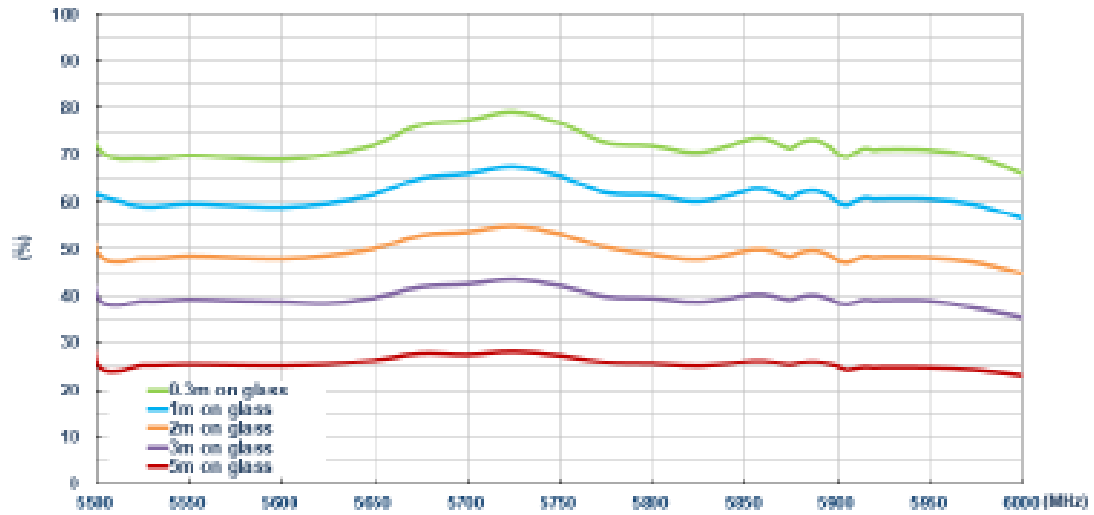


## 7.2 On Glass

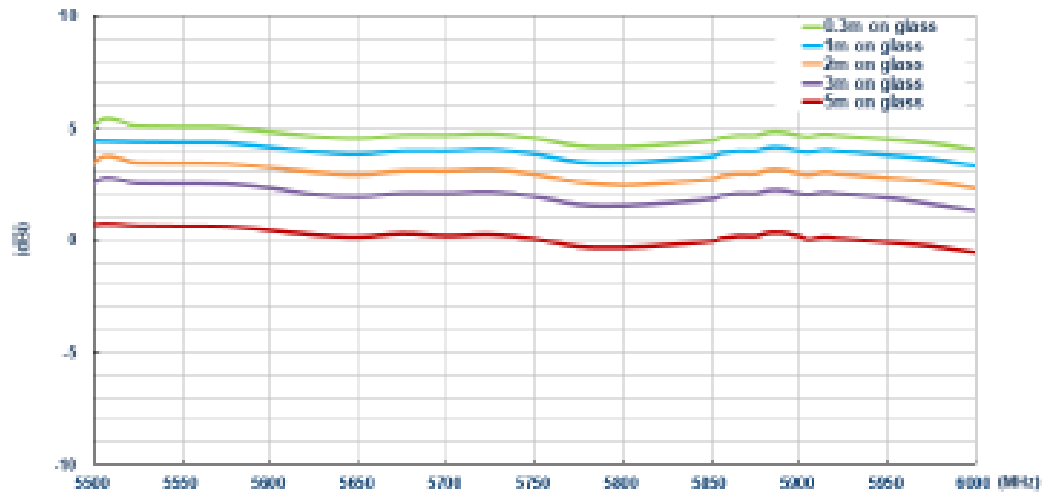
### 7.2.1 Return Loss



### 7.2.2 Efficiency

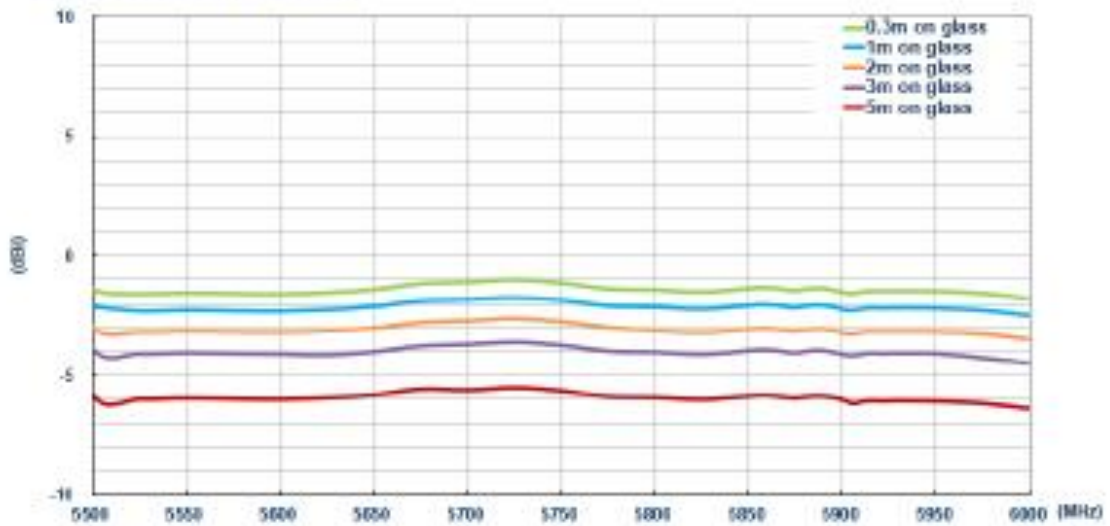


### 7.2.3 Peak Gain



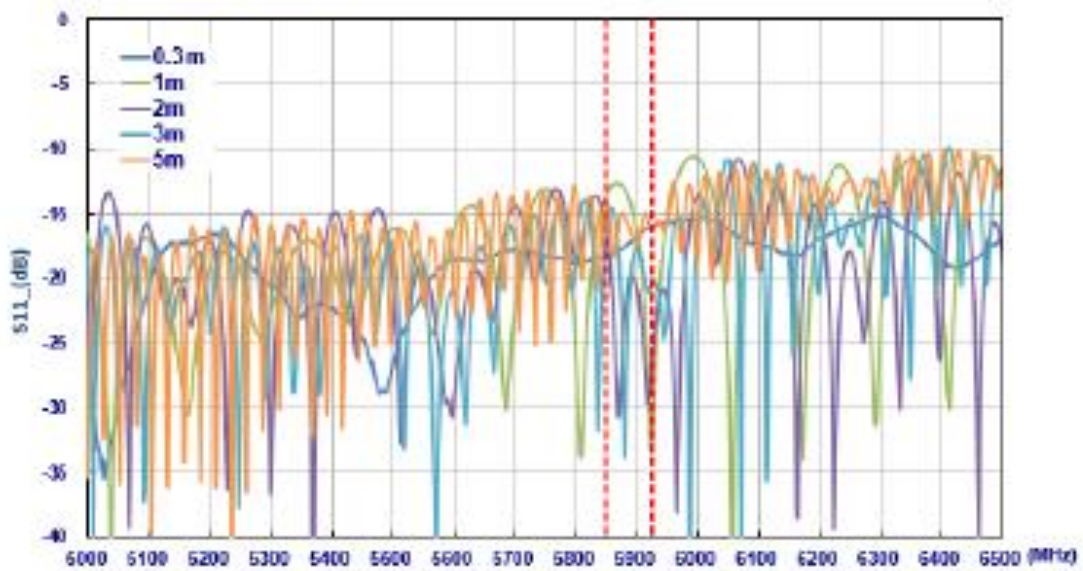


### 7.2.4 Average Gain

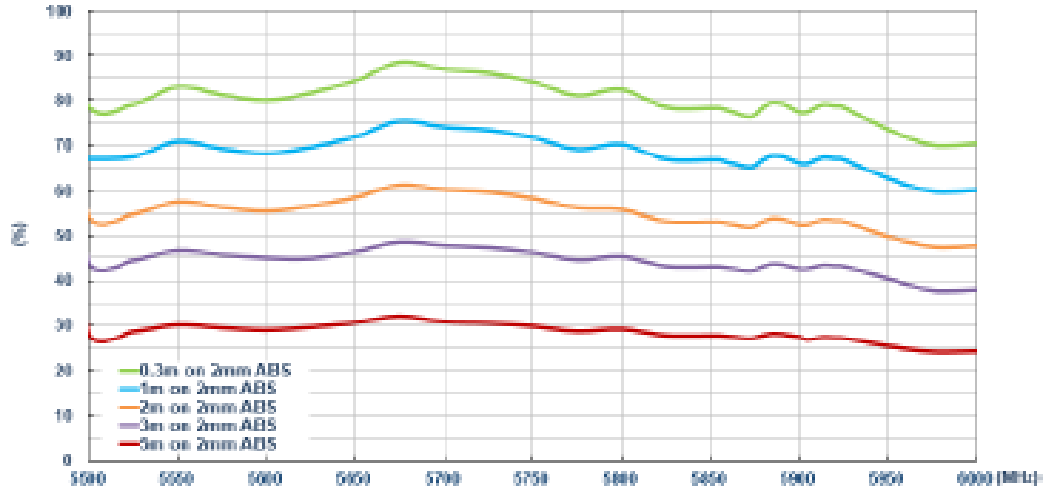


## 7.3 On 2mm ABS

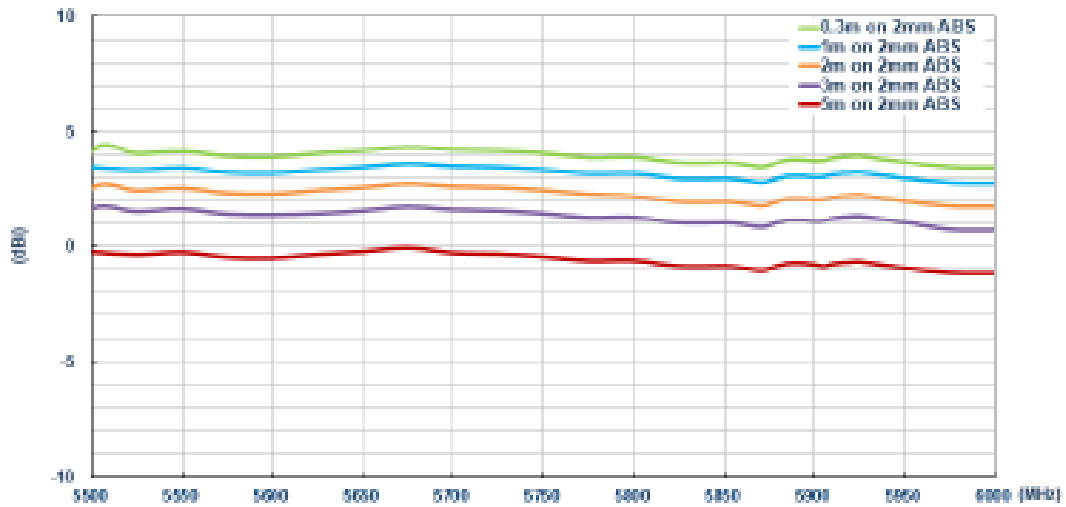
### 7.3.1 Return Loss



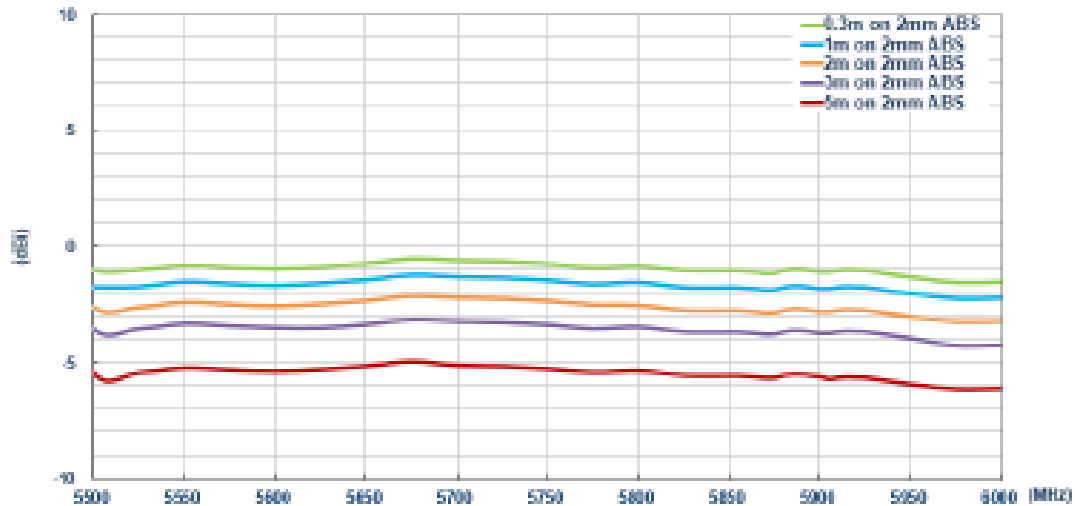
### 7.3.2 Efficiency



### 7.3.3 Peak Gain



### 7.3.4 Average Gain



Taoglas makes no warranties based on the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice.

Taoglas reserves the rights to this document and the information contained herein. Reproduction, use or disclosure to third parties without express permission is strictly prohibited.

Copyright © Taoglas Ltd.



# 10. Datasheet GPSlim236

**HOLUX** GPSlim236

Bluetooth **GPS Receiver**

**User's Guide**

**HOLUX** Technology Inc.

1F, No. 30, R&D Rd. II, HsinChu City 300, Science-based Industrial Park, Taiwan

TEL: 03-6687000 FAX: 03-6687111

E-Mail: [info@holux.com.tw](mailto:info@holux.com.tw) Website: [www.holux.com.tw](http://www.holux.com.tw)

Rev.: A

All Right Reserved

## Table of Contents

TABLE OF CONTENTS .....	2
1. OVERVIEW .....	3
2. PACKING LIST .....	4
3. MAIN FUNCTIONS .....	4
4. TECHNICAL SPECIFICATION .....	5
4.1. BASIC SPECIFICATION .....	5
4.2. ACQUISITION TIME (AVERAGED).....	5
4.3. RECEIVER ACCURACY.....	5
4.4. USE LIMITATION .....	5
4.5. POWER SUPPLY .....	5
4.6. OUTPUT AND INTERFACE .....	6
4.7. PHYSICAL .....	6
4.8. OTHER FUNCTIONS.....	6
5. GETTING STARTED.....	8
5.1. HARDWARE DESCRIPTION.....	9
5.2. SOFTWARE INSTALLATION.....	12
5.3. INSTALLATION OF TESTING PROGRAM .....	13
6. OPTIONAL ACCESSORIES.....	15
7. DRIVER INSTALLATION.....	16
7.1 SYSTEM REQUIREMENT.....	16
7.2 INSTALLATION.....	16
7.3 IMPORTANT .....	16
8. WARRANTY .....	17
9. TROUBLE SHOOTING.....	17

## 1. Overview



(Fig.1)

The **HOLUX GPSlim236 Wireless Bluetooth GPS Receiver** (Fig. 1) is a total solution GPS receiver with Bluetooth, UART interface and built-in rechargeable battery for high sensitivity to tracking signal. **GPSlim236** design is based on SiRF Star III low power Architecture.

**GPSlim236** is a dual-function GPS receiver. Not only transmit satellite information through the PDA or Notebook with Bluetooth interfaces but also is a G-Mouse GPS receiver through a data cable to deliver satellite signal to the device without Bluetooth interface.

This positioning application meets strict needs such as car navigation, mapping, surveying, security, agriculture and so on. Only clear view of sky and certain power supply are necessary to the unit. **GPSlim236** contacts to other device through Bluetooth device, compatible interface of RS-232 or USB, and built-in recharge battery to save satellite information such as the status of satellite signal, the last location, date and time of last use.

With low power consumption, the **GPSlim236** tracks up to 20 satellites at a time, re-acquires satellite signals in 100 ms and updates position data every second. Trickle-Power allows the unit operates a fraction of the time and Push-to-Fix permits user to have a quick position fix even though the receiver usually stays off.

## 2. Packing List

Congratulations on your purchase of the GPSlim236 GPS Receiver. We hope it will be useful to you for a long time. Before you begin, make sure that your package includes the following items. If any of these items are missing, please contact your local HOLUX dealer or distributor.

- HOLUX **GPSlim236** Wireless Bluetooth GPS receiver 1 Set
- Travel power supply / Cigarette adapter 1 Set
- Manual and Driver CD 1 Piece
- **GPSlim236** Quick guide 1 Piece
- Warranty card 1 Piece

## 3. Main functions

**GPSlim236** provides a series of functions. It is well suited to system integration and users who use PDA, Notebook PC with Bluetooth device.

- Built in SiRF Star III Low power consumption chipset.
- 20 parallel satellite-tracking channels for fast acquisition and reacquisition.
- High speed signal acquisition using 200,000 time/frequency search channels.
- Built-in WAAS/EGNOS Demodulator without additional any hardware. Or use the high-sensitive software to get the fast acquisition and reacquisition in the urban, canyon and foliage environments.
- Compatible with Bluetooth Serial Port Profile (SPP) completely.
- Low power consumption. Built-in rechargeable and changeable Lithium-ion battery without external power supply, and the working time lasts at least 10 hours.
- Provide Continue mode and Power saving mode for user's requirement.
- Provide expand terminal contact to other system without Bluetooth device.
- Built-in rechargeable battery for memory and RTC backup and for fast Time To First Fix (TTFF).
- Support NMEA0183 v2.2 data protocol and SiRF binary code.
- 4 colors LED to show the status of device.
- Active antenna connector for getting better satellites signal.
- FLASH based program memory. New software revisions upgradeable through serial interface.



- Small, sleek, and lightweight design easily fits in your hand.
- Over-Temperature protection
- Enhanced algorithms -SnapLock and SnapStart provide superior navigation, performance in urban, canyon and foliage environments.
- For Car navigation, Marine navigation, Fleet management, AVL, Personal navigation, Tracking System, and Mapping device application.

## 4. Technical Specification

### 4.1. Basic Specification

- Chipset : SiRF Star III chipset.
- Channels : 20 parallel satellite-tracking channels.
- Frequency : 1575.42 MHZ.
- Receiver : L1, C/A code.

### 4.2. Acquisition Time (averaged)

- Reacquisition : 0.1sec.
- Cold start : < 42 seconds.
- Warm start : < 38 seconds .
- Hot start : < 1 seconds

### 4.3. Receiver Accuracy

- Normal : 5-25 meters CEP without SA.
- Enable EGNOS or WAAS :
  - Position: < 2.2 meters, horizontal 95% of time
  - < 5 meters, Vertical 95% of time
- Velocity : within 0.1 meters / second
- Time : 1 microsecond synchronized GPS time

### 4.4. Use Limitation

- Altitude : < 18,000 meters (60,000 feet)
- Velocity: : < 736 meters/ second (1000Knots)
- Acceleration : 4 G.
- Jerk : 20 meters / second, max

### 4.5. Power Supply

- External Voltage: 5VDC +/- 10%
- Batteries :
  - Main Power: Built-in rechargeable Lithium-ion for system power.
  - Backup Power: Rechargeable Lithium-ion battery for memory & RTC backup.
- Working voltage: 75-85mA (Normal mode).  
30mA (Power Saving).

- Working period (In Battery full power status):
  - > 10 hours on Continue mode.
  - > 16 hours on Power Saving mode.
- Protection circuit on **GPSiM236** should stop charging the cell when over-temperature condition  $-50^{\circ}\text{C}$  occurs.

#### 4.6. Output and Interface

- **Output**

- I. Output protocol

- Baud Rate : 38400 bps

- Data bit : 8

- Parity : No

- Stop bit : 1

- II. Format. NMEA0183 V2.2 : GPGGA (1time/1 sec), GPGSA (1 time/5 sec.), GPGSV (1time /5 sec.), GPRMC (1time /1 sec.), GPVTG (1 time/1 sec), (GLL, or SiRF binary format for optional).

- III. Datum : WGS84.

- **Input/ Output Interface:**

- I. Compatible Bluetooth Serial Port Profile (SPP), Version1.1 and class 2(up to 10 meter range).

- II. In/Out Port. GPS signal (Out)/Command(In) with CMOS/TTL Level • Mini

- USB Type B Connector and Cable option :

- (a) GR230-A1(RS232 data cable)

- (b) GR230-A2 (USB data cable)

- (c) GR230-A3 (Mini USB port to PS2 port ).

- **External Antenna interface:**

- 3.0V input MMCX type active antenna connector

#### 4.7. Physical

- Size : 46.3 × 67 × 19 mm
- Weight : < 56 g
- Operating Temperature :  $-10^{\circ}\text{C}$  to  $+60^{\circ}\text{C}$ (under the un-charging condition); Charging Temperature  $0^{\circ}\text{C}$  to  $+45^{\circ}\text{C}$
- Storage Temperature :  $-20^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$
- Operating humidity : 5% to 95% No condensing

#### 4.8. Other Functions

- Bluetooth frequency: 2.4 ~2.48GHZ
- Bluetooth Input Sensitivity: -80dbm

- 
- Low sensitivity of receiving satellite signal : -189 dBW
  - External antenna interface: MMCX
  - LED Functions : Indicate Bluetooth status, GPS status, Battery Status and Battery charging status

## 5. Getting Started

### STEP 1. Charge Battery

Please charge battery till LED off for the first time.

**Power cable plug in** Power cable connect to power socket



**Charge Battery**

Battery indicator light:

**Power too low** ----- red LED

**Charging** ----- green LED

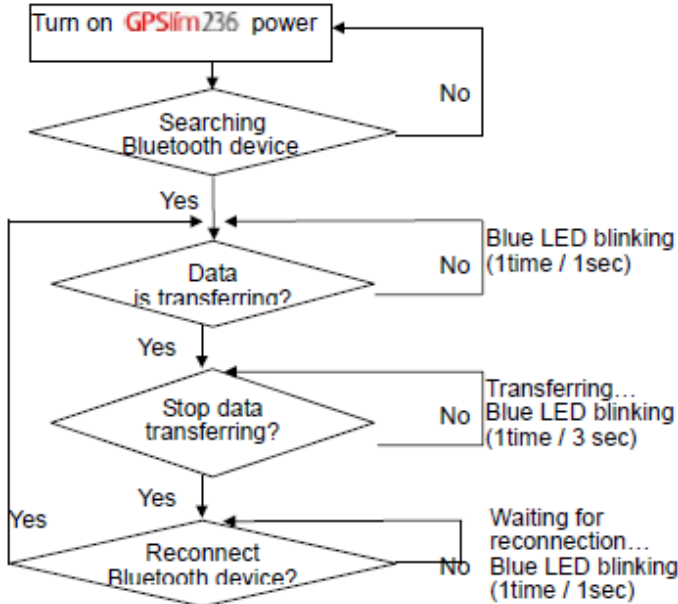
**Full or Not in charging** -- LED off



Mini USB power socket

### STEP 2. Turn on Power

Bluetooth Status –



Note:  
Some PDAs have to re-open Bluetooth manager for Bluetooth re-connection.

GPS Status ---

Put **GPSlim236** in clear view of the sky without any obstruction for better satellite acquiring.

**Turn on GPSlim236 power**

**Turn off GPSlim236 power**



**Search GPS** -- Steady orange LED on

**Power Off** -- Orange LED off

**Position Fixed** -- Orange LED blinking






## 5.1. Hardware Description

1). **GPSim236** Body description see Fig. 2:



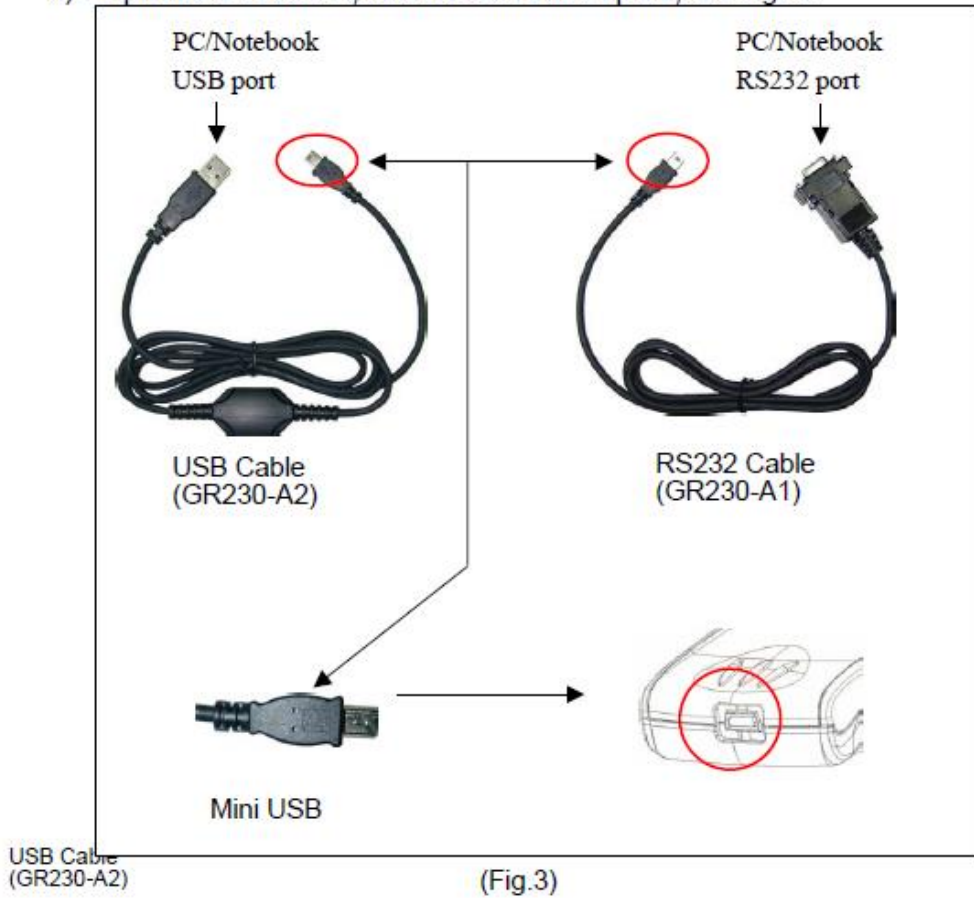
(Fig.2)

2). LED status :

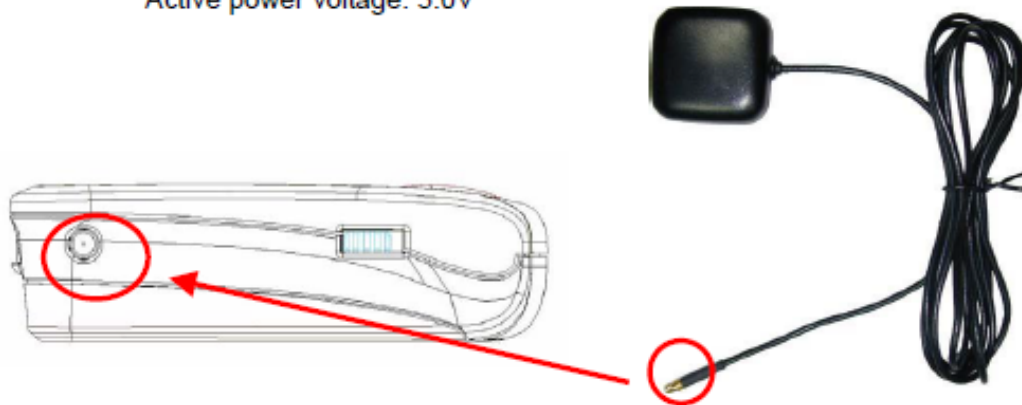
SYMBOL	COLOR	STATUS		DESCRIPTION
 Bluetooth	Blue	Blinking	1 times / 1 sec	Search Bluetooth Device
			1 time / 1 sec	Standby Mode
			1 time / 3 sec	Transferring Data
 Battery	Red	Light on		Power too low
	Green	Light on		In charging
	N/A	Light off		Battery full or Not in charging
 GPS	Orange	Light on		Acquiring Satellites
		Blinking		Position Fixed

3). Power Switch :  
Slide Switch.

4). Optional accessories, and connector description, see Fig. 3.

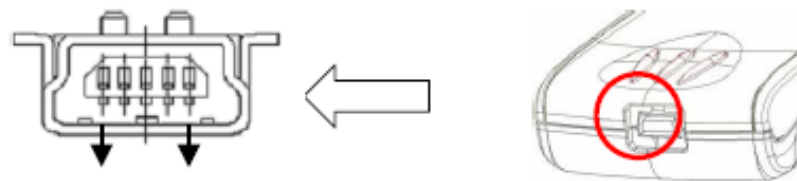


- 3) External active antenna connector, see Fig. 5 ◦  
 Connector type: MMCX.  
 Active power voltage: 3.0V



(Fig.5)

- 4) Power Jack & Data Port, see Fig. 6 ◦  
 Jack type: Mating face of 5 pin Mini USB Type B female ◦  
 Pin definition see table 1 ◦



(Fig.6)

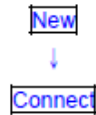
Table 1

Pin	Pin Name	Signal and description
1	GND	Signal ground, Battery charging ground.
2	VOUT	Unregulated voltage out: 3.6 V max 100mA.
3	TXD	Transmit Data. From organizer to peripheral.(Voltage Level is 3.3V ~ 5.0V).
4	RXD	Receive Data. Form peripheral to organizer.(Voltage level is 3.3V ~ 5.0V).
5	VCHARG	Positive terminal of DC adaptor that powers the internal charging circuit of Li-Ion battery. The approved power supply is 5.0V +/- 5%@1A.

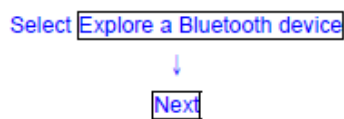
## 5.2. Software Installation

The following is the steps of software installation to setup on PDA, DELL AXIM with Bluetooth Manager. For other PDA, the steps may be a little different.  
( Bluetooth Manger is one of popular program used for Bluetooth device)

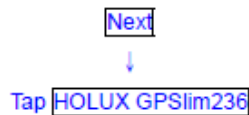
### 1. Open “Bluetooth Manager” on pocket pc.



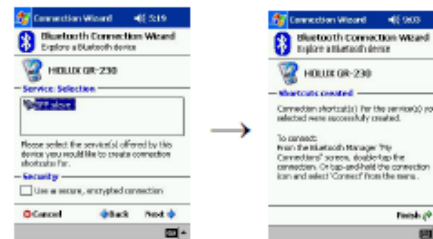
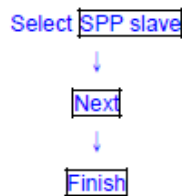
### 2. Search Bluetooth device “HOLUX GPSim236”



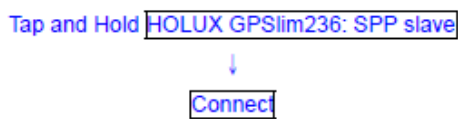
### 3. Find the Bluetooth device



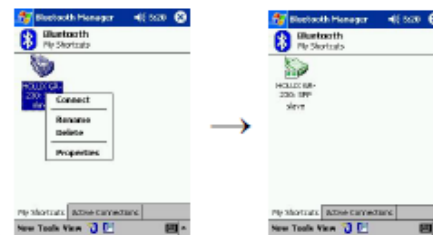
### 4. Connect to SPP Slave



### 6. 5. Finish Bluetooth Manager Setup



Finish Bluetooth setup (opposite arrow is displayed)



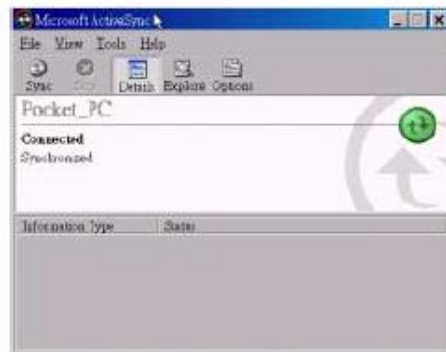


### 5.3. Installation of testing program

(GPSViewer.exe is compatible with Microsoft Pocket PC or other operation system alike.)

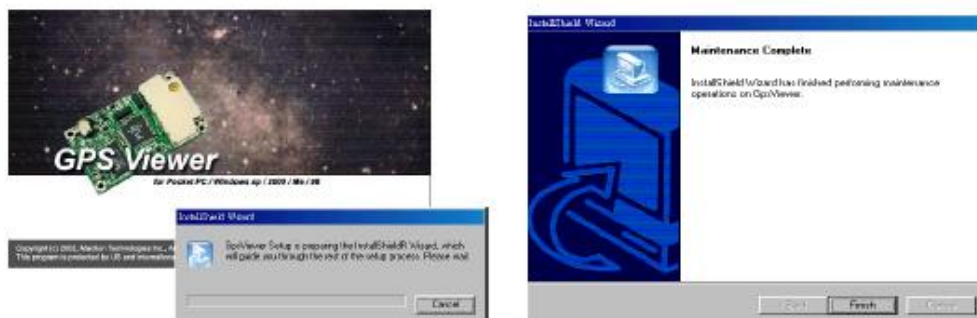
- 1). Install Microsoft ActiveSync to your PC, refer to your Pocket PC manual for installation procedure, as Fig. 5.
- 2). Setup your Pocket PC cradle to Desktop PC UART port. The Microsoft ActiveSync will detect your Pocket PC automatically.

Setup your Pocket PC cradle to Desktop PC UART port. The Microsoft ActiveSync will detect your Pocket PC automatically, as Fig. 7.



(Fig. 7)

- 3). Double click the GPSViewer.exe on your PC, then Holux GPSViewer.exe program will install automatically, as Fig. 8.



(Fig. 8)

- 4) Push "Start" → "Programs" → "GPSViewer" on PDA, as Fig. 9.



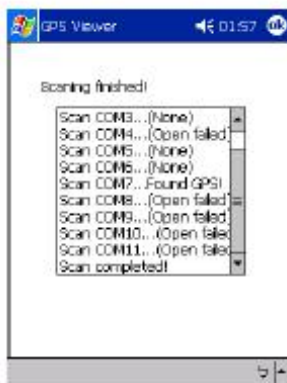
(Fig. 9)

5) The following window is show after executing GPSViewer, as Fig. 10.



(Fig. 10)

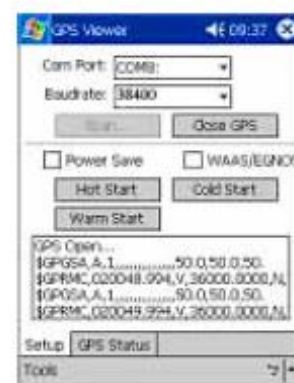
6) Setup Baud rate: 38400, then push "Scan" bottom to scan your COM Port (Example thePAQ 3970 is the output port COM8). Select your COM Port (COM1 ~ COM10), then push "Open GPN" bottom, as Fig. 11, Fig. 12, and Fig. 13.



(Fig. 11)

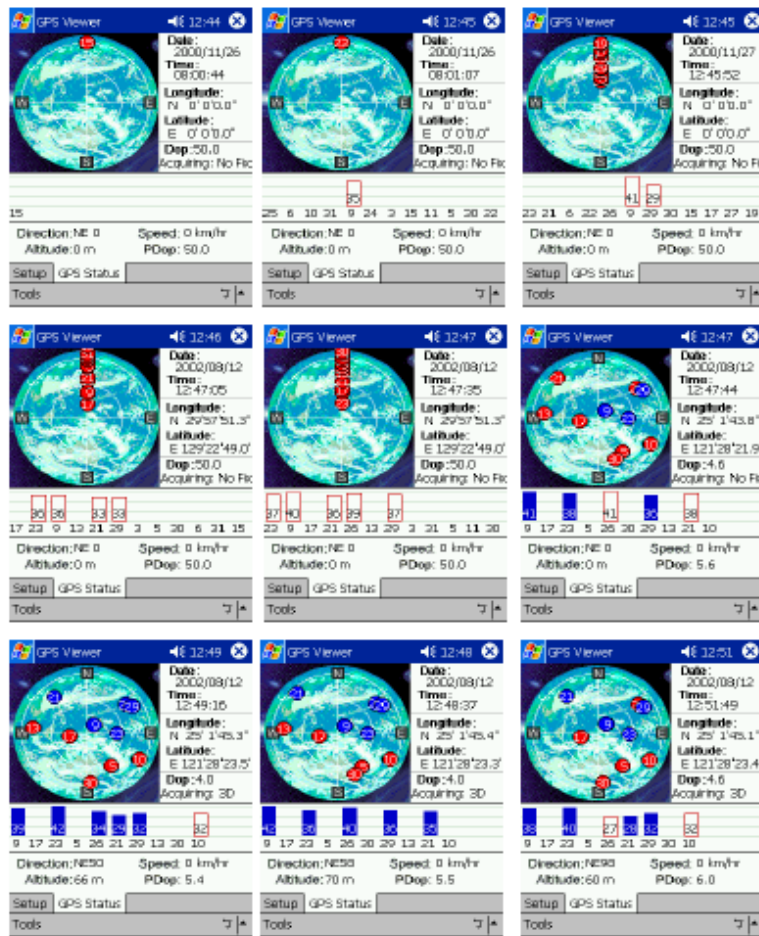


(Fig. 12)



(Fig. 13)

7) Select "GPS Status" to show the satellite diagram like below, as Fig. 14.



(Fig. 14)

## 6. Optional accessories

GPSi<sup>™</sup>m236 has many accessories to satisfy customers' requirement see table 2. After using GPSi<sup>™</sup>m236 with the following accessories, it can transmit message with PDA, Note Book easily.

Table 2

Item	Description	Note
GR230-A1	1.5M RS232 data cable	
GR230-A2	1.5M USB data cable	
GR230-A3	Output convert to GM-210's PDA Car charger adaptor	
GR230-B1	2M 28db MMCX connector active antenna	

## 7. Driver Installation

You can use any **GPSírm236** accessories data cable without installing driver except GR230-A2 USB cable. The following is the steps of installation GR230-A2 USB cable.

### 7.1 System Requirement

CPU: IBM, Pentium, or other compatible PC.

Memory: above 16 MB

System: Windows 98/Me/2000/XP

### 7.2 Installation

- I. Copy entire **GPSírm236** USB folder from CD to hard disk.
- II. Connect GR-230-A2 USB connector to computer without **GPSírm236** GPS receiver. While the computer automatically starts the installation program, please direct the driver to the **GPSírm236** USB folder.

### 7.3 Important

Verify the COM port to start using your own navigation software.

- I. Click **<Start>** menu, select → **<Setting>**, then enter→ **<Controller>**
- II. After entering **<Controller>**, and select **<System>**.
- III. Select **<Device Manager>**.
- IV. Find the **< Connector(COM & LPT)>** and check the Virtual COM Port, which was created by the USB driver.

Please note that the virtual COM port number might be different from every computer. Before using navigation software, please confirm the COM Port numbers created by your computer and provided by your navigation software. Otherwise, the navigating software won't receive the satellite signal, because of the un-match COM Port setting.

## 8. Warranty

The **GPSlim236** is warranted to be free from defects in material and functions for a period of one year from the date of purchase. Any failure of this product within this period under normal conditions will be replaced at no charge to the customers.

- **GPSlim236** has built Li-battery inside, please avoid closing high temperature environment or sun shine directly for a long time.
- \* User has to return **GPSlim236** to HOLUX if the inner Li battery has to be replaced.

## 9. Trouble Shooting

Problems	Reasons	Methods
No position output but timer is counting	Weak or no GPS signal can be received at the place of GPSlim236	Connect an external antenna, which locate as a open space to your <b>GPSlim236</b> and then run GPSViewer Cold start function.
	At outdoor space but GPS signal is blocked by building or car roof.	Go outdoor and run GPSViewer Cold start function to try again, or connect an external antenna to improve the poor GPS signal.
Execute fail	Bluetooth function unstable	Power On/Off <b>GPSlim236</b> . Re-Start PDA or PC and reference sec 5.2 re-install software
Can not turn on the COM port	Install <b>GPSlim236</b> incompletely or operate the device is being used with same COM port	Install <b>GPSlim236</b> completely or stop other device that is being used.
Can not find out <b>GPSlim236</b>	Poor connection	Re-Start PDA or PC and reference sec. 5.2 re-install software.
No Signal	No action for few minutes may cause Pocket PC entry power save mode. It will close the COM port at the same time.	Close the application and execute it again to reopen the COM port.
	Weak or no GPS signal when using <b>GPSlim236</b> indoor	Connect an external antenna to your <b>GPSlim236</b> .

---

## **Federal Communications Commission (FCC) Statement**

### **15.21**

You are cautioned that changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

### **15.105(b)**

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
  - Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

You are cautioned that changes or modifications not expressly approved by the party responsible for compliance could void your authority to operate the equipment.

### **FCC RF Radiation Exposure Statement:**

This Transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.