



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH  
Escola d'Enginyeria de Barcelona Est

TRABAJO FIN DE GRADO

Grado en Ingeniería Industrial y Automática.

**CONTROL Y GUIADO DE UN ROBOT MÓVIL.**



**Volumen I**

**Memoria - Presupuesto**

**Autor:** Raúl Cruzado Muñoz  
**Director:** Juan Gámiz  
**Convocatoria:** Mayo 2017



## **Índice Volumen I.**

Memoria.

Presupuesto.









UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH  
Escola d'Enginyeria de Barcelona Est

TRABAJO FIN DE GRADO

**Grado en Ingeniería Industrial y Automática.**

**CONTROL Y GUIADO DE UN ROBOT MÓVIL.**



**Memoria**

**Autor:** Raúl Cruzado Muñoz  
**Director:** Juan Gámiz  
**Convocatoria:** Mayo 201









## Resum

En aquest Projecte de fi de grau s'ha realitzat un prototip pel control i el guiat d'un robot mòbil autònom, que és un robot SR1 Explorer que incorpora una sèrie de sensors i un mòdul de comunicació sense fil per poder ser governat per un ordinador.

El prototip és, a més del propi robot de quatre mòduls balises fixes i mòdul mòbil que va col·locat sobre el robot, les balises envien informació a l'ordinador que mitjançant un algoritme és capaç de calcular la posició del robot.

Les balises estan formades principalment per un microcontrolador i de un transceptor de radiofreqüència XBee PRO S1.

El software s'ha realitzat en llenguatge de programació de LabVIEW i consta de tres programes diferents. El primer és el programa guiat del robot on se li envien al robot diferents ordres que ell interpreta. El segon programa és la memorització i repetició de trajectòries i el tercer programa, una vegada s'han registrat la potència rebuda de les balises en diferents punts de la zona de treball, ens permet dirigir al robot a les coordenades que l'indiquem i mitjançant les balises i l'algoritme de posicionament ens indica en quines coordenades es troba.

El resultat és una aplicació molt útil i econòmica per a la localització de robots en llocs a l'interior.

## Resumen

En este Proyecto de Fin de Grado se ha realizado un prototipo para el control y el guiado de un robot móvil autónomo, es un SR1 Explorer que incorpora una serie de sensores y un modulo de comunicación inalámbrica para poder ser gobernado desde un ordenador.

El prototipo consta además del propio robot de cuatro módulos balizas fijas y módulo móvil que va colocado sobre el robot, las balizas envían información al ordenador que mediante un algoritmo es capaz de calcular la posición del robot.

Las balizas están formadas principalmente por un microcontrolador y de un transceptor de radiofrecuencia XBee PRO S1.

El software se ha realizado en lenguaje de programación LabVIEW y consta de tres programas diferentes. El primero programa es el guiado del robot donde se le envían al robot diferentes ordenes que él interpreta. El segundo programa es la memorización y repetición de trayectorias y el tercer programa, una vez se han registrado la potencia recibida de las balizas en diferentes puntos de la zona de trabajo, nos permite dirigir al robot a las coordenadas que le indiquemos y mediante las balizas y el algoritmo de posicionamiento nos indica en qué coordenadas se encuentra.

El resultado es una aplicación muy útil y económica para la localización de robots en lugares en interior.

## **Abstract**

In this end of degree Project, I have made a prototype for the control (and guidance) of an autonomous mobile robot. It is an SR1 explorer that incorporates a series of sensors and a Wireless communication module to be controlled from a computer.

In addition to the robot itself, the prototype consists of four fixed beacons modules and a mobile module that is placed on the robot. The beacons send information to the computer that by means of an algorithm is able to calculate the position of the robot.

The beacons consist mainly of a microcontroller and an XBee PRO S1 radio frequency transceiver.

The software has been made in LabVIEW programming language and consists of three different programs. The first program is the guidance of the robot where different orders are sent to the robot that he interprets. The second program is the memorization and repetition of trajectories and the third program, once the power received from the beacons at different points in the work area has been recorded, allows us to direct the robot to the coordinates we indicate and using the beacons And the positioning algorithm tells us what coordinates it is in.

The result is a very useful and economical application for locating robots in indoor locations.



# Índice Memoria

|   |            |
|---|------------|
| <b>RESUM</b>  | <b>I</b>   |
| <b>RESUMEN</b>  | <b>II</b>  |
| <b>ABSTRACT</b>   | <b>III</b> |
| <b>1. INTRODUCCIÓN</b>  | <b>1</b>   |
| 1.1. Objetivos del trabajo. ....                                    | 2          |
| 1.2. Alcance. ....  | 2          |
| <b>2. ANTECEDENTES.</b>   | <b>3</b>   |
| 2.1. Introducción. ....   | 3          |
| 2.2. Robots móviles. ....   | 3          |
| 2.2.1. SR1 Explorer. ....   | 3          |
| 2.2.2. Robot Stellaris de Texas Instruments. ....                   | 5          |
| 2.2.3. LabVIEW Robotics Starter Kit ....                            | 6          |
| 2.2.4. Robotino® de Festo ....                                      | 9          |
| 2.2.5. Robot LEGO Mindstorms NXT ....                               | 11         |
| 2.3. Sistemas de localización. ....                                 | 12         |
| 2.3.1. Cricket Indoor Location System. ....                         | 12         |
| 2.3.2. Active Bat Location System ....                              | 13         |
| 2.3.3. Ubisense Real Time Location System. ....                     | 14         |
| 2.3.4. WiLocT Location System. ....                                 | 15         |
| <b>3. INGENIERÍA DE CONCEPCIÓN.</b>                                 | <b>16</b>  |
| 3.1. Sistemas de localización relativos. ....                       | 16         |
| 3.1.1. Sistemas de localización odométrica. ....                    | 16         |
| 3.1.2. Sistemas de localización inerciales. ....                    | 18         |
| 3.2. Sistemas de localización basados en puntos de referencia. .... | 18         |
| 3.2.1. Principio de localización. ....                              | 19         |
| 3.2.2. Tecnología. ....   | 22         |
| 3.3. Elección de la técnica a implementar. ....                     | 26         |
| <b>4. INGENIERÍA DE DESARROLLO.</b>                                 | <b>27</b>  |
| 4.1. Localización odométrica. ....                                  | 27         |
| 4.1.1. Modelo cinemático del robot. ....                            | 27         |
| 4.2. Localización por potencia de señal recibida RSSI. ....         | 29         |

|           |  |            |
|-----------|--|------------|
| 4.2.1.    | Ecuación de FRISS. ....  | 29         |
| 4.2.2.    | Índice de la potencia de la señal recibida (RSSI).....                 | 30         |
| 4.2.3.    | Técnica para la localización. ....                                     | 31         |
| 4.3.      | Hardware del sistema. ....   | 35         |
| 4.3.1.    | Transceptor inalámbrico. ....  | 35         |
| 4.3.2.    | Microcontrolador. ....   | 38         |
| 4.3.3.    | Interfaz ZigBee-PC. ....   | 40         |
| 4.3.4.    | Baterías. ....   | 41         |
| 4.3.5.    | Robot SR1 Explorer.....  | 42         |
| 4.3.6.    | Transceptor inalámbrico externo .....                                  | 49         |
| 4.3.7.    | Herramientas hardware del sistema. ....                                | 50         |
| 4.4.      | Software del sistema. ....   | 51         |
| 4.4.1.    | Programación XBee. ....  | 51         |
| 4.4.2.    | Software de configuración XBee.....                                    | 55         |
| 4.4.3.    | Programación de los microcontroladores. ....                           | 59         |
| 4.4.4.    | Programación SR1. ....   | 65         |
| 4.4.5.    | Programa en LabVIEW para el PC de control y guiado del robot móvil. 68 |            |
| 4.4.6.    | Herramientas de software. ....   | 87         |
| <b>5.</b> | <b>ANÁLISIS DEL IMPACTO AMBIENTAL. _____</b>                           | <b>89</b>  |
|           | <b>PRUEBAS Y RESULTADOS. _____</b>                                     | <b>91</b>  |
|           | <b>CONCLUSIONES. _____</b>   | <b>99</b>  |
|           | <b>BIBLIOGRAFÍA _____</b>  | <b>100</b> |



# Índice de ilustraciones y tablas.

## Lista de ilustraciones.

|  |    |
|--|----|
| ILUSTRACIÓN 2.2-1 ROBOT SR1 EXPLORER.....  | 4  |
| ILUSTRACIÓN 2.2-2 ROBOT PLATAFORMA ARDUINO.....  | 5  |
| ILUSTRACIÓN 2.2-3 ROBOT STELLARIS DE TEXAS INSTRUMENTS.....  | 6  |
| ILUSTRACIÓN 2.2-4 ROBOT SBRIO-9632.....  | 9  |
| ILUSTRACIÓN 2.2-5 ROBOTINO DE FESTO.....   | 11 |
| ILUSTRACIÓN 2.2-6 ROBOT LEGO MINDSTORMS.....   | 12 |
| ILUSTRACIÓN 2.3-1 NODO DEL SISTEMA CRICKET.....  | 13 |
| ILUSTRACIÓN 2.3-2 TRIANGULACIÓN CON EL SISTEMA ACTIVE BAT.....   | 14 |
| ILUSTRACIÓN 2.3-3 TAG COMPACTO DEL SISTEMA UBISENSE.....   | 15 |
| ILUSTRACIÓN 2.3-4 TAG COMPACTO DEL SISTEMA WILoCT.....   | 15 |
| ILUSTRACIÓN 3.2-1 SISTEMA DE LOCALIZACIÓN DOA POR DIRECCIÓN DE LLEGADA.....                                  | 19 |
| ILUSTRACIÓN 3.2-2 SISTEMA DE LOCALIZACIÓN POR TOA TIEMPO DE LLEGADA.....                                     | 20 |
| ILUSTRACIÓN 3.2-3 SISTEMA DE LOCALIZACIÓN POR TDOA DIFERENCIA DE TIEMPO DE LLEGADA.....                      | 21 |
| ILUSTRACIÓN 3.2-4 LOCALIZACIÓN POR IMAGEN.....   | 24 |
| ILUSTRACIÓN 3.2-5 SISTEMA CENTRALIZADO BASADO EN PROCESAMIENTO DE IMÁGENES.....                              | 24 |
| ILUSTRACIÓN 4.1-1 MODELO CINEMÁTICO DEL ROBOT SR1.....   | 27 |
| ILUSTRACIÓN 4.1-2 AUMENTO DE LA INCERTIDUMBRE EN LA LOCALIZACIÓN ODOMÉTRICO.....                             | 29 |
| ILUSTRACIÓN 4.2-1 VALORES TEÓRICOS RSSI EN FUNCIÓN DE LA DISTANCIA.....                                      | 30 |
| ILUSTRACIÓN 4.2-2 LOCALIZACIÓN BASADA EN RECONOCIMIENTO DE PATRONES.....                                     | 31 |
| ILUSTRACIÓN 4.2-3 VALOR RSSI EN FUNCIÓN DE LA DISTANCIA DE LA MATRIZ $MR_N - X_N$ A UNA DISTANCIA DE 2M..... | 32 |
| ILUSTRACIÓN 4.2-4 VALOR RSSI EN FUNCIÓN DE LA DISTANCIA DE LA MATRIZ $MA_N$ A UNA DISTANCIA DE 2M.....       | 33 |
| ILUSTRACIÓN 4.2-5 VALOR RSSI EN FUNCIÓN DE LA DISTANCIA DE LA MATRIZ $MT_N$ A UNA DISTANCIA DE 2M.....       | 34 |
| ILUSTRACIÓN 4.3-1 MÓDULOS XBEE Y XBEE PRO SERIES 1.....  | 35 |
| ILUSTRACIÓN 4.3-2 DIMENSIONADO DE MÓDULOS XBEE Y XBEE PRO SERIES 1.....                                      | 36 |
| ILUSTRACIÓN 4.3-3 TIPOS DE ANTENAS DE LOS MÓDULOS XBEE.....  | 37 |
| ILUSTRACIÓN 4.3-4 COMUNICACIÓN UART DE LOS MÓDULOS XBEE.....   | 38 |
| ILUSTRACIÓN 4.3-5 PAQUETE DE DATOS TRANSMITIDOS POR EL MÓDULO UART DE XBEE.....                              | 38 |
| ILUSTRACIÓN 4.3-6 ENCAPSULADO SOIC DEL PIC18F2550.....   | 40 |
| ILUSTRACIÓN 4.3-7 PLACA DE DESROLLO BBEE EXPLORER USB.....   | 41 |
| ILUSTRACIÓN 4.3-8 BATERÍAS, PORTA BATERÍAS Y CARGADOR DE VERÍAS DEL SISTEMA.....                             | 42 |
| ILUSTRACIÓN 4.3-9 ROBOT SR1 CON EL KIT COMPLETO.....   | 42 |
| ILUSTRACIÓN 4.3-10 CHASIS DEL ROBOT SR1.....   | 43 |
| ILUSTRACIÓN 4.3-11 CIRCUITO DE CONTROL DEL ROBOT SR1.....  | 43 |
| ILUSTRACIÓN 4.3-12 SENSOR BRÚJULA DIGITAL.....   | 46 |
| ILUSTRACIÓN 4.3-13 DECLINACIÓN MAGNÉTICA 2015.....   | 47 |
| ILUSTRACIÓN 4.3-14 SENSOR ULTRASONIDOS ROBOT SR1.....  | 48 |
| ILUSTRACIÓN 4.3-15 TRANSCÉPTOR INALÁMBRICO EXTERNO DEL ROBOT SR1.....  | 49 |
| ILUSTRACIÓN 4.4-1 ESTRUCTURA DEL MODO API DE LOS MÓDULOS XBEE.....   | 52 |

|   |    |
|---|----|
| <b>ILUSTRACIÓN 4.4-2</b> ESTRUCTURA DE UNA TRAMA API DE PETICIÓN DE TRANSMISIÓN DE LOS MÓDULOS XBEE. ....         | 53 |
| <b>ILUSTRACIÓN 4.4-3</b> ESTRUCTURA DE UNA TRAMA API DE PAQUETE RECIBIDO DE LOS MÓDULOS XBEE. ....                | 54 |
| <b>ILUSTRACIÓN 4.4-4</b> ESTRUCTURA DEL MODO API DE COMANDO AT DE LOS MÓDULOS XBEE. ....                          | 55 |
| <b>ILUSTRACIÓN 4.4-5</b> GENERADOR DE TRAMAS API DEL SOFTWARE X-CTU. ....   | 56 |
| <b>ILUSTRACIÓN 4.4-6</b> CONSOLA DE COMUNICACIÓN SERIE DEL SOFTWARE X-CTU. ....                                   | 56 |
| <b>ILUSTRACIÓN 4.4-7</b> VENTANA DE CONFIGURACIÓN DE LOS MÓDULOS XBEE DEL SOFTWARE X-CTU. ....                    | 57 |
| <b>ILUSTRACIÓN 4.4-8</b> DIAGRAMA DE FLUJO DEL PROGRAMA DEL NODO MÓVIL. ....                                      | 63 |
| <b>ILUSTRACIÓN 4.4-9</b> DIAGRAMA DE FLUJO DEL PROGRAMA DE LOS NODOS BALIZA. ....                                 | 64 |
| <b>ILUSTRACIÓN 4.4-10</b> APLICACIÓN PROGRAMA DE GUIADO DEL ROBOT SR1. ....                                       | 68 |
| <b>ILUSTRACIÓN 4.4-11</b> DIAGRAMA DE FLUJO DEL PROGRAMA DE GUIADO DEL ROBOT SR1. ....                            | 70 |
| <b>ILUSTRACIÓN 4.4-12</b> APLICACIÓN GRÁFICA GRABACIÓN DE TRAYECTORIA. ....                                       | 71 |
| <b>ILUSTRACIÓN 4.4-13</b> DIAGRAMA DE FLUJO PROGRAMA GRABACIÓN DE TRAYECTORIA. ....                               | 73 |
| <b>ILUSTRACIÓN 4.4-14</b> APLICACIÓN GRÁFICA REPRODUCCIÓN DE TRAYECTORIAS. ....                                   | 74 |
| <b>ILUSTRACIÓN 4.4-15</b> DIAGRAMA DE FLUJO REPRODUCCIÓN DE TRAYECTORIA. ....                                     | 76 |
| <b>ILUSTRACIÓN 4.4-16</b> LOCALIZACIÓN BASADA EN RECONOCIMIENTO DE PATRONES. ....                                 | 77 |
| <b>ILUSTRACIÓN 4.4-17</b> APLICACIÓN GRÁFICA DE REGISTRO DE POTENCIA RSSI. ....                                   | 78 |
| <b>ILUSTRACIÓN 4.4-18</b> DIAGRAMA DE FLUJO REGISTRO DE POTENCIA RSSI (1/2). ....                                 | 79 |
| <b>ILUSTRACIÓN 4.4-19</b> DIAGRAMA DE FLUJO REGISTRO DE POTENCIA RSSI (2/2). ....                                 | 80 |
| <b>ILUSTRACIÓN 4.4-20</b> ROTACIÓN DE EJES X Y DE REFERENCIA RESPECTO AL NORTE MAGNÉTICO. ....                    | 81 |
| <b>ILUSTRACIÓN 4.4-21</b> APLICACIÓN GRÁFICA LOCALIZACIÓN ODOMÉTRICA Y DE MEDICIÓN DE POTENCIA DE RECEPCIÓN. .... | 83 |
| <b>ILUSTRACIÓN 4.4-22</b> DIAGRAMA DE FLUJO APLICACIÓN LOCALIZACIÓN ODOMÉTRICA Y MEDICIÓN DE RSSI (1/3). ....     | 84 |
| <b>ILUSTRACIÓN 4.4-23</b> DIAGRAMA DE FLUJO APLICACIÓN LOCALIZACIÓN ODOMÉTRICA Y MEDICIÓN DE RSSI (2/3). ....     | 85 |
| <b>ILUSTRACIÓN 4.4-24</b> DIAGRAMA DE FLUJO APLICACIÓN LOCALIZACIÓN ODOMÉTRICA Y MEDICIÓN DE RSSI (3/3). ....     | 86 |
| <br>  |    |
| <b>ILUSTRACIÓN C 1</b> POTENCIA RSSI TEÓRICA DE LOS MÓDULOS XBEE. ....  | 92 |
| <b>ILUSTRACIÓN C 2</b> POTENCIA RSSI TEÓRICA DE LOS MÓDULOS XBEE. ....  | 93 |
| <b>ILUSTRACIÓN C 3</b> POTENCIA RSSI REAL DE LOS MÓDULOS XBEE. ....   | 94 |
| <b>ILUSTRACIÓN C 4</b> POTENCIA RSSI REAL DE LOS MÓDULOS XBEE. ....   | 94 |
| <b>ILUSTRACIÓN C 5</b> PRUEBA DE RANGO X-CTU. ....  | 95 |
| <b>ILUSTRACIÓN C 6</b> RSSI DISTANCIA 0,80 M TIEMPO 30 MINUTOS. ....  | 96 |
| <b>ILUSTRACIÓN C 7</b> RSSI DISTANCIA 2,5 M TIEMPO 30 MINUTOS. ....   | 96 |
| <b>ILUSTRACIÓN C 8</b> RSSI DISTANCIA 3,6 M TIEMPO 10 MINUTOS. ....   | 96 |
| <b>ILUSTRACIÓN C 9</b> GRÁFICA VALORES RSSI BALIZA 1 (0,0). ....  | 97 |
| <b>ILUSTRACIÓN C 10</b> GRÁFICA VALORES RSSI BALIZA 2 (270,0). ....   | 97 |
| <b>ILUSTRACIÓN C 11</b> GRÁFICA VALORES RSSI BALIZA 3 (470,0). ....   | 97 |
| <b>ILUSTRACIÓN C 12</b> GRÁFICA VALORES RSSI BALIZA 4 (270,470). ....   | 98 |
| <b>ILUSTRACIÓN C 13</b> GRÁFICA DE RESULTADOS DE LOCALIZACIÓN POR INTENSIDAD DE SEÑAL RECIBIDA. ....              | 98 |

## Lista de tablas

|   |    |
|---|----|
| <b>TABLA 1</b> FUNCIONALIDAD DE LOS PINES DE LOS MODULOS XBEE. ....                       | 36 |
| <b>TABLA 2</b> HERRAMIENTAS HARDWARE DEL SISTEMA.....                                     | 50 |
| <b>TABLA 3</b> EJEMPLO DE CONFIGURACIÓN DE LOS MÓDULOS XBEE MEDIANTE COMANDOS AT. ....    | 51 |
| <b>TABLA 4</b> IDENTIFICADORES API Y SU TIPO DE MENSAJE DE LOS MÓDULOS XBEE. ....         | 53 |
| <b>TABLA 5</b> DISPOSITIVOS DE LA RED DEL SISTEMA DE LOCALIZACIÓN Y SU FUNCIÓN.....       | 58 |
| <b>TABLA 6</b> PARÁMETROS DE CONFIGURACIÓN DE CADA NODO DEL SISTEMA DE LOCALIZACIÓN. .... | 59 |
| <b>TABLA 7</b> HERRAMIENTAS DE SOFTWARE. ....   | 87 |



# 1. Introducción

Hoy día, los robots forman parte elemental en las labores cotidianas del ser humano; son usados en tareas repetitivas, difíciles y peligrosas. La robótica ha tenido un desarrollo tan rápido e intenso, que en pocos años ha alcanzado metas que en el pasado se creían temas de ciencia ficción. Los robots son muy utilizados en plantas de manufactura, montaje, transporte, exploraciones en la Tierra y en el espacio, cirugía, armamento, investigación de laboratorios, entre otros, siendo la industria, el principal mercado donde los robots son utilizados.

La necesidad de movimiento no teledirigido en robótica se aprecia claramente en sistemas que desarrollan su operación en entornos remotos.

Se entiende el control de un robot como la monitorización continua del mismo, es decir, conocer en cada momento la posición y orientación respecto un cero u origen. A este concepto se le denomina localización.

El conocimiento en tiempo real de la posición precisa del robot móvil dentro de un entorno, nos permite optimizar la planificación de las trayectorias ante situaciones imprevistas, llevando a cabo las tareas impuestas con precisión y eficiencia.

El giro de los elementos motrices de un vehículo móvil, independientemente de que sean ruedas o cadenas, puede proporcionar información suficiente como para determinar su posición. En estas situaciones se dice que la localización está basada en la odometría.

Sin embargo, existen errores asociados a deslizamientos, fricciones o la propia orografía del terreno que son acumulables. De este modo, llegará un momento en el que el sistema no pueda determinar con el grado de exactitud necesario la posición del robot.

Por tanto, se precisa de un sistema alternativo de localización, que sin omitir los resultados provenientes de las mediciones odométricas, corrijan la posición del vehículo y localicen el mismo con la mayor exactitud posible. Un proceso de localización, el cuál, a través de la información proveniente de los sensores externos, sea capaz de determinar la posición y orientación ocupada por el robot móvil.

Existen una gran variedad de técnicas de localización de robots móviles. El enfoque de cada una de ellas va determinado en función de las características previas, como el conjunto de sensores disponible, el medio en el cuál se mueve el robot o el grado de conocimiento del entorno y de las tareas a realizar.

## 1.1. Objetivos del trabajo.

El principal objetivo de este proyecto fin de grado es proponer una solución óptima y de bajo coste, usando tecnología de redes inalámbricas que permita el control y guiado de un robot móvil en espacios cerrados.

Los principales puntos propuestos para poder llevar a cabo el proyecto son los siguientes:

- Definir y diseñar una arquitectura de sistema que permita la implementación de un sistema de control y guiado de robots en interiores.
- Desarrollar subrutinas que nos permitan el control y guiado de un robot autónomo, el robot utilizado es un SR1 Explorer, cedido para la realización del proyecto por la EEBE.
- Investigación y estudio de los dispositivos de radiofrecuencia XBee PRO S1, también cedidos por la EEBE para la realización del proyecto, que realizarán la función de balizas para el posicionamiento del robot móvil.
- Implementar todo el software necesario para el sistema de control y guiado en lenguaje de programación LabVIEW de National Instruments.
- Realizar las pruebas necesarias para comprobar la funcionalidad del sistema diseñado.

## 1.2. Alcance.

El alcance de este proyecto se basa en una serie de aplicaciones para el control y el guiado de un robot móvil.

Las aplicación indica una posible posición del robot dentro un área previamente cartografiada, la corrección de la posición del robot y salvar posibles obstáculos queda fuera del alcance del proyecto.

La parte de hardware queda fuera del alcance del proyecto, solo se describen las partes esenciales para el funcionamiento del sistema.

## 2. Antecedentes.

### 2.1. Introducción.

El problema de la localización de robots móviles consiste en contestar, desde el punto de vista del robot, a la pregunta: ¿Dónde estoy? Esto significa que el robot, valiéndose de sus sensores, debe encontrar su posición relativa al entorno donde se encuentra. El problema general de la localización consta de un cierto número de instancias que se pueden clasificar según su grado ascendente de dificultad: seguimiento de posición, posicionamiento global, recuperación ante transferencias intempestivas y localización en ambientes dinámicos.

La localización del robot resulta clave en el diseño de robots verdaderamente autónomos. Si un robot no sabe dónde está, probablemente resulte difícil determinar qué hacer a continuación. Con el fin de localizarse a sí mismo, un robot tiene acceso a la información relativa y absoluta proporcionada por sus sensores internos y externos que nos permiten conocer, respectivamente, la distancia recorrida y las características del entorno del robot. Teniendo en cuenta esta información, el robot tiene que determinar su posición con la mayor precisión posible.

Esta adquisición de información se ve afectada por la imprecisión propia de los dispositivos sensoriales por lo que es preciso combinarlas de forma óptima con diferentes sistemas de localización con el fin de estimar, con la mayor precisión posible, la posición real del robot.

### 2.2. Robots móviles.

En la actualidad la mayoría de los robots móviles que se distribuyen tienen la finalidad didáctica o de investigación, existen plataformas abiertas de código libre donde los participantes comparten sus experiencias y avances sin ánimo de lucro. A continuación se presentan algunos ejemplos de robots móviles comerciales.

#### 2.2.1. SR1 Explorer.

SR1 Explorer es un robot multifuncional enfocado al aprendizaje del desarrollo y construcción de robots móviles de experimentación. La idea de diseño original buscó una relación de equilibrio entre un número muy elevado de funciones y un entorno de programación amigable y relativamente sencillo que permitiera el aprendizaje para usuarios no demasiado avanzados, y que facilita el propio desarrollador.

Además, está diseñado para interactuar de manera autónoma con el entorno debido a su robustez y a la calidad de sus materiales. Entre los sensores que incorpora destacan sensores de contacto, de inclinación, de luz, de infrarrojos, de distancia por ultrasonido, de temperatura y una brújula.

Por otro lado, cuenta con dispositivos de salida como un zumbador piezoeléctrico, diodos LED o radio módem.

Con el kit completo se proporciona al cliente una versión actualizada del software con el juego de programas y ejemplos para aprender a controlar el robot sin dificultades. Incluye la opción de programación libre además de funciones predefinidas como 'Navegación inteligente' o 'Perro guardián'.

También se incluyen programas para el ordenador que permiten el control remoto del robot, de los datos sonda o de su cámara de video. Se han lanzado al mercado tres versiones del robot, actualizables entre sí, que permiten escoger la dificultad y la autonomía y que van incorporando funcionalidades.



Ilustración 2.2-1 Robot SR1 Explorer.

#### 2.2.1.1. Plataforma Arduino.

Arduino es una compañía de hardware libre, que desarrolla plataformas que integran un microcontrolador, conectores periféricos para manejar sensores y actuadores y que incluyen un entorno de desarrollo software. El proyecto se inicia en el año 2006 en el instituto IVREA, en Italia.



Desde entonces esta empresa fabrica placas con un coste bajo y que presentan características ideales para familiarizarse con proyectos que interactúen con el entorno

El robot de Arduino utiliza un microcontrolador Atmega32U4. Tiene numerosos puertos de entrada y salida con los que interactuar con elementos periféricos como un altavoz, brújula, diodos LED, pulsadores, una pantalla LCD o sensores infrarrojos seguidores de línea.

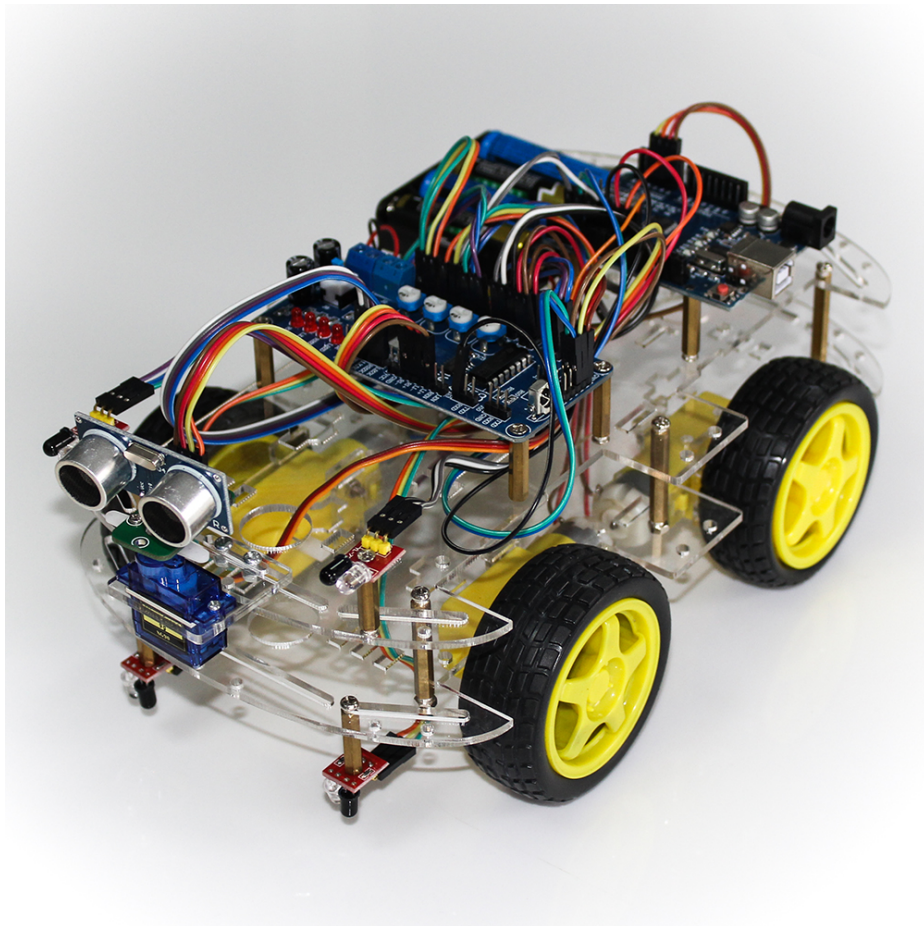


Ilustración 2.2-2 Robot plataforma Arduino.

### 2.2.2. Robot Stellaris de Texas Instruments.

Esta versión de robot educativo desarrollada por Texas Instruments utiliza el microcontrolador LM3S9B92 y se comercializa en un completo kit que incluye todo el hardware y el software necesario para que el usuario pueda sacarle el máximo rendimiento.

Se puede alimentar vía 4 pilas AA o bien a través del puerto mini USB. Destaca por incorporar dos conectores macho de montaje para PCB de 20 pines que permiten futuras conexiones inalámbricas con módulos de radio de baja potencia estandarizados por Texas Instruments y una pantalla OLED.

Además, es capaz de reproducir archivos con la extensión WAV almacenados en la memoria FLASH interna del microcontrolador.

También incorpora conector de tarjeta micro-SD, códec de audio I2S con altavoz y conector RJ45 Ethernet.



**Ilustración 2.2-3** Robot Stellaris de Texas Instruments.

### 2.2.3. LabVIEW Robotics Starter Kit

El NI LabVIEW Robotics Starter Kit es una plataforma robótica móvil que incluye sensores, motores y hardware de NI Single-Board RIO para el control incrustado. El software LabVIEW Robotics incluido con la plataforma incluye características para principiantes y para aquellos que tienen más experiencia. Si es nuevo en LabVIEW, puede utilizar la API de alto nivel de LabVIEW Robotics Starter Kit para iniciar y controlar rápidamente el robot en tiempo real. Si es un usuario más avanzado, puede acceder al FPGA y realizar una personalización de bajo nivel.

La simplicidad de este kit de arranque lo hace ideal para enseñar conceptos de robótica y mecatrónica o para desarrollar un prototipo de robot con LabVIEW Robotics. El kit de inicio de LabVIEW Robotics incluye un programa pre-construido que ejecuta un algoritmo de evitación de obstáculos de

histograma de campo vectorial (VFH) basado en la retroalimentación de los sensores ultrasónicos incluidos. Con el Módulo de Robótica de LabVIEW, puede cambiar fácilmente el comportamiento del robot desarrollando sus propios algoritmos desde la base o utilizando algoritmos incorporados en el software de LabVIEW Robotics, como la planificación de rutas.

Características del kit de inicio de LabVIEW Robotics:

- Motores de 12 VDC de Pitsco Education con 152 rpm y 300 oz-in. De par
- Codificadores ópticos de cuadratura con 400 pulsos por revolución
- Sensor de distancia ultrasónico para mediciones de distancia entre 2 cm y 3 m
- Soporte de montaje para un barrido de 180 grados del medio ambiente

El dispositivo de control y adquisición incorporado sbRIO-9632 de NI integra un procesador en tiempo real, una matriz de puerta programable programable por el usuario (FPGA) y E / S en una sola placa de circuito impreso (PCB). Cuenta con un procesador industrial de 400 MHz, una puerta de enlace 2M Xilinx Spartan FPGA, 110 líneas de E / S digitales de 3.3 V (compatibles con 5 V / TTL), 32 canales de entrada analógica de 16 bits diferenciales y 16 canales de 250 kS / s , Y cuatro canales de salida analógica de 16 bits a 100 kS / s. También tiene tres conectores para E / S de expansión que utilizan módulos de E / S de la serie NI C de nivel de placa. El sbRIO-9632 ofrece un rango de temperatura de funcionamiento de -20 a 55 ° C e incluye un rango de entrada de alimentación de 19 a 30 VCC, 128 MB de DRAM para operación integrada y 256 MB de memoria no volátil para almacenar programas y registro de datos

Este dispositivo cuenta con un puerto Ethernet integrado de 10/100 Mbit / s que puede utilizar para realizar comunicaciones programáticas a través de la red y alojar servidores Web (HTTP) y de archivos (FTP) incorporados. También puede utilizar el puerto serie RS232 para controlar dispositivos periféricos.

Características del sbRIO-9632:

- Controlador en tiempo real integrado, FPGA reconfigurable y E / S en una sola placa Puerta 2M Xilinx Spartan FPGA
- Procesador en tiempo real Freescale de 400 MHz
- 128 MB de memoria DRAM, 256 MB de almacenamiento no volátil
- Puerto serie RS232 para dispositivos periféricos
- 110 líneas de E / S digitales de 3.3 V (compatibles con 5 V / TTL)
- 32 canales de entrada analógica de 16 bits de un solo extremo / 16 canales a 250 kS / s Cuatro canales de salida analógicos de 16 bits a 100 kS / s
- Puerto Ethernet 10 / 100BASE-T

- Bajo consumo de energía con una sola entrada de alimentación de 19 a 30 V CC

Programa el dispositivo sbRIO-9632 utilizando el entorno de desarrollo gráfico de LabVIEW. El procesador en tiempo real ejecuta el módulo en tiempo real de LabVIEW en el sistema operativo en tiempo real (RTOS) Wind River VxWorks para una fiabilidad y determinismo extremos. Con la adición del módulo LabVIEW MathScript RT, puede implementar fácilmente los archivos .m personalizados en el hardware de NI en tiempo real, combinando sintaxis gráfica y textual.

Puede programar rápidamente el FPGA reconfigurable a bordo utilizando el módulo LabVIEW FPGA para el control de alta velocidad, la sincronización de E / S personalizada y el procesamiento de señales en línea. LabVIEW contiene controladores integrados y APIs para manejar la transferencia de datos entre el FPGA y el procesador en tiempo real.

El módulo LabVIEW Robotics se conecta al entorno de desarrollo de LabVIEW y ofrece una extensa biblioteca de robótica que incluye lo siguiente:

- Conectividad a sensores robóticos
- Algoritmos fundamentales para la operación inteligente y la percepción robusta Simulador de ambiente basado en física incorporado
- Funciones de movimiento para hacer que su robot o vehículo se mueva
- Ejemplos de aplicación del mundo real
- Cinemática inversa y directa
- Bibliotecas para protocolos incluyendo I2C, SPI, PWM y JAUS

El Módulo LabVIEW Robotics proporciona herramientas para desarrollar una variedad de robots, desde robots simples y educativos hasta sofisticados sistemas autónomos. El software ofrece arquitecturas o plantillas de software que puede utilizar para programar fácilmente un robot basado en comportamiento simple y para programar robots complejos con múltiples subsistemas ejecutándose en paralelo.

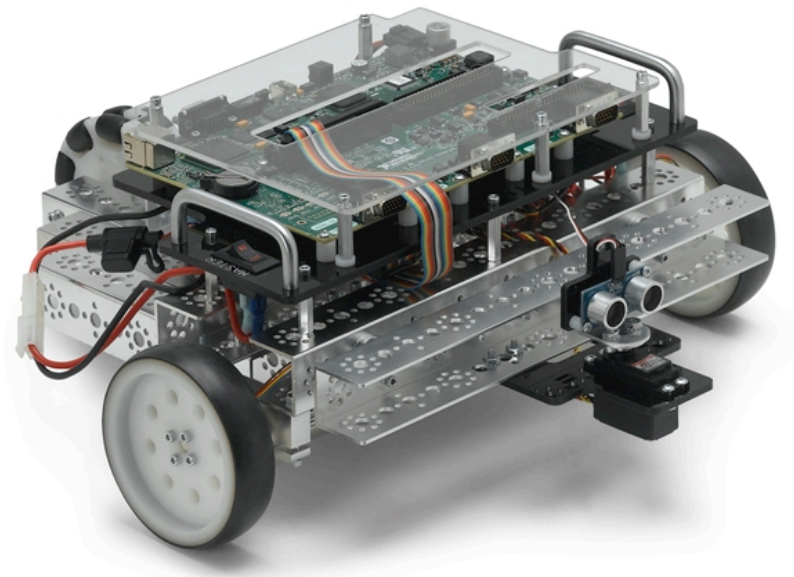


Ilustración 2.2-4 Robot sbRIO-9632.

#### 2.2.4. Robotino® de Festo

Robotino® es la plataforma de robots móviles para investigación y educación. Con su accionamiento omnidireccional, sensores, interfaces y extensiones específicas de aplicación, Robotino® se puede utilizar de forma muy flexible. Los lenguajes y sistemas de programación más importantes están disponibles para la programación de aplicaciones individuales.

Robotino® es un sistema de robot móvil de alta calidad, plenamente funcional con accionamiento omnidireccional. Las tres unidades de accionamiento permiten realizar movimientos en todas direcciones adelante, atrás y lateralmente. Además, el robot también puede girar sobre un punto. También está equipado con una webcam y varios tipos de sensores analógicos para medición de distancias, por ejemplo, sensores binarios para protección de colisiones y sensores digitales para detectar la velocidad real. Esto asegura que se cumplan todas las prestaciones que se exigen a este tipo de sistemas.

El sistema puede ponerse en marcha inmediatamente sin necesidad de conectarlo a un PC.

El controlador de Robotino® consiste en un PC embebido con una tarjeta compact flash, en la cual se han instalado varias aplicaciones de demostración y el sistema operativo (Linux). Las aplicaciones de demostración pueden ejecutarse directamente desde el panel de control del Robotino®.

Robotino® puede programarse con el software Robotino®View en un PC a través del LAN inalámbrico. Robotino®View es capaz de transmitir señales al controlador del motor, así como visualizar, cambiar y

evaluar valores de los sensores. Robotino® puede programarse con Robotino®View incluso durante el funcionamiento real.

También están disponibles APIs Linux y C++ para la programación del Robotino®.

La webcam permite visualizar y evaluar una imagen de cámara en vivo con ayuda del Robotino®View. Con ello, pueden implementarse aplicaciones tales como el trazado de rutas y seguimiento de objetos.

Robotino® es autónomo. Numerosos sensores, una cámara y un controlador de altas prestaciones aportan al sistema la necesaria “inteligencia”.

Puede accederse al controlador directamente a través de la LAN inalámbrica (WLAN). Cuando está correctamente programado, Robotino® realiza de forma autónoma las tareas asignadas.

Pueden conectarse actuadores y sensores adicionales a través de un interface de E/S.

La unidad de control puede configurarse de forma flexible utilizando diversos módulos plug-in.

El controlador del Robotino® consta de 3 componentes:

- Procesador PC 104, compatible con MOPSIcdVE, 300 MHz, y sistema operativo
- Linux con kernel en tiempo real, SDRAM 128 MB
- Tarjeta Compact flash con API C++ para controlar el Robotino®
- Punto de acceso LAN inalámbrico

Robotino® está equipado con nueve sensores de medición de distancia por infrarrojos, que se hallan montados en el chasis formando un ángulo de 40° entre sí. Con estos sensores, Robotino® puede detectar objetos en las zonas circundantes. Cada uno de estos sensores puede ser interrogado individualmente por medio de la placa de circuito de E/S. Con ello pueden evitarse obstáculos, pueden mantenerse distancias y adoptar protecciones frente a un determinado objetivo. Los sensores son capaces de medir distancias con precisión o relativas a objetos, con valores entre 4 y 30 cm. La conexión del sensor es especialmente sencilla e incluye tan sólo una señal de salida analógica y la alimentación. La electrónica de evaluación del sensor determina la distancia, que puede leerse como una señal analógica.



Ilustración 2.2-5 Robotino de Festo.

### 2.2.5. Robot LEGO Mindstorms NXT

Los sistemas LEGO Mindstorms NXT son robots programables que aparecieron en el año 2006. El objetivo de estos robots es permitir el desarrollo de sistemas robóticos con facilidad mediante el uso de los bloques LEGO y con componente conocido como block inteligente NXT, este componente se puede programar con distintos lenguajes de programación el software incluido se conoce como NXT-G el cual es un ambiente de programación gráfica dicho software fue desarrollado por National instruments.

El componente más importante es el bloque inteligente NXT este se encarga de realizar el procesamiento de datos de leer los sensores y controlar los actuales.

El bloque inteligente NXT tiene como base un Microprocesador principal Atmel ARM7 de 32 bits con una memoria tipo flash de 256 kB una memoria RAM de 64 kB quiere brindar soporte de procesamiento y mayores prestaciones de ejecución programas.



Ilustración 2.2-6 Robot LEGO Mindstorms.

## 2.3. Sistemas de localización.

En este apartado se muestran algunos ejemplos de sistemas de localización en espacios cerrados que han sido desarrollados utilizando alguna de las diversas tecnologías (Ultrasonidos, Radiofrecuencia, Infrarrojos, WiFi, etc.) para la comunicación inalámbrica. La mayoría de estos sistemas son proyectos en proceso de mejora y no se venden oficialmente en el mercado.

### 2.3.1. Cricket Indoor Location System.

El sistema Cricket es uno de los proyectos más relevantes en el campo de la localización en interiores. Este proyecto del MIT (Massachusetts Institute of Technology) utiliza unos pequeños nodos basados, principalmente, en un transceptor de radiofrecuencia, un microcontrolador y un sensor de ultrasonidos.

Existen dos tipos de nodos: los nodos “baliza” y los nodos “escuchadores”. Los nodos baliza son puntos de referencia fijos y están normalmente colocados en el techo o las paredes de un edificio. Los nodos escuchadores pueden ser fijos o estar posicionados sobre objetos móviles y son los que necesitan ser posicionados.

El principio de funcionamiento del sistema se basa en la diferencia de tiempo de llegada entre las señales de radiofrecuencia y las de ultrasonidos para determinar cuál es la orientación de los nodos



móviles. Para determinar la posición, cada baliza transmite periódicamente cierta información por radiofrecuencia, como por ejemplo, la identificación de la baliza, y los nodos escuchadores reciben estas señales y miden la distancia a las balizas próximas.

Las balizas del sistema Cricket pueden transmitir de forma simultánea pero el nodo móvil sólo puede escuchar la señal de una baliza a la vez. Además, para conocer la posición, se necesitan tres lecturas diferentes por lo que, si alguna de ellas no es válida, el tiempo para resolver la posición se ve aumentado todavía más.

El problema anterior se resuelve utilizando un filtrado de Kalman extendido que mantiene la posición y velocidad actuales y las ajusta según los datos reales que se reciben. Esto produce un error de estimación y cuando éste es muy elevado se reinicia y se empieza la estimación desde cero de nuevo.

La precisión que Cricket proporciona está comprendida entre 1 y 3 cm.

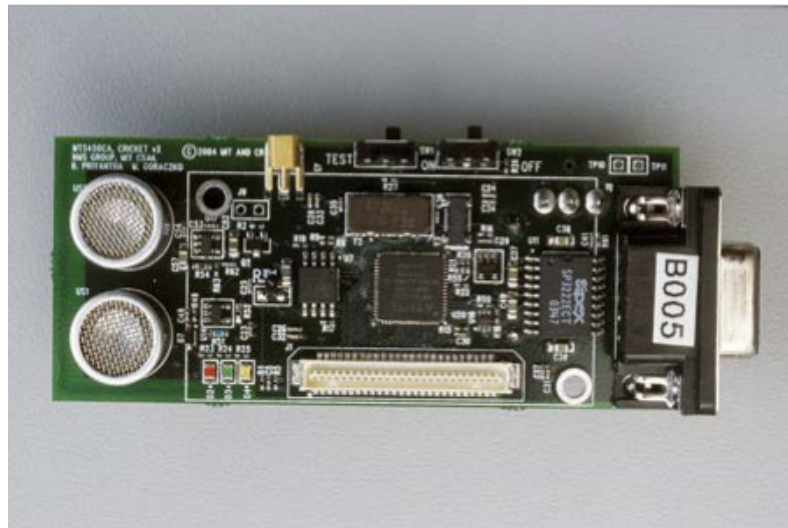


Ilustración 2.3-1 Nodo del sistema Cricket.

### 2.3.2. Active Bat Location System

El sistema de localización Active Bat, sucesor del sistema Active Badge, es un proyecto de AT&T que utiliza una matriz de sensores receptores ubicados en el techo de una habitación separados 1,2 metros entre ellos, uno o varios transmisores inalámbricos (fijos o móviles) y una estación central de radiofrecuencia.

Los transmisores están compuestos por un transceptor de RF, diversos transmisores de ultrasonidos, una FPGA y un microprocesador y disponen de una ID única asociada a ellos. Los receptores se componen de un receptor de RF y una interfaz serie de datos. La estación base organiza la actividad de los transmisores enviéndoles señales de radiofrecuencia a cada uno de ellos. Cuando los transmisores reciben esta señal, envían otra con los sensores de ultrasonidos con destinación a los receptores del

techo. Por último, los receptores reciben esta señal de ultrasonidos y calculan el tiempo de diferencia entre ésta y la señal inicial de RF, estimando así la distancia aproximada de los transmisores. Todo este sistema está conectado a un ordenador cuya función es la de analizar todos los datos para ubicar los transmisores.

A diferencia del sistema Cricket, los nodos receptores pueden escuchar la señal del transmisor a la vez, lo que agiliza el proceso de detección. En cambio, el número de nodos receptores es mayor y la configuración de la matriz es más costosa.

Este sistema dispone de una precisión de hasta 3 cm.

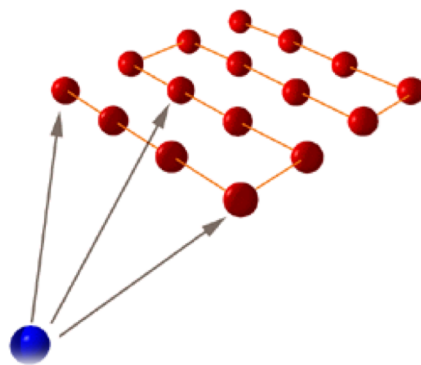


Ilustración 2.3-2 Triangulación con el sistema Active Bat.

### 2.3.3. Ubisense Real Time Location System

La empresa Ubisense dispone de un sistema de localización en tiempo real (RTLS) diseñado especialmente para su uso en interiores y la tecnología empleada se basa en la Banda Ultra Ancha (UWB). A diferencia de los ultrasonidos, esta tecnología de RF permite realizar mediciones a distancias mayores.

Como en los casos anteriores, se dispone de sensores transmisores y receptores. Los dispositivos móviles que se desean detectar se dotan de unos tags activos que transmiten unos pulsos cuando alguno de los sensores de la red, ubicados en posiciones conocidas, lo pide. Los sensores se agrupan en células, habiendo un sensor maestro en cada célula que coordina el resto de sensores y manda por Ethernet o WiFi la posición de los tags. Además, los tags van equipados con un detector de movimiento que envía un pulso cuando detecta cierta aceleración, avisando así de que existe movimiento. Esto ayuda al sistema para evitar tener un gran número de sensores.

La precisión de este tipo de sistema es de 15 cm y se puede actualizar la posición alrededor de 20 veces por segundo.



Ilustración 2.3-3 Tag compacto del sistema Ubisense.

#### 2.3.4. WiLocT Location System

WiLocT es otro sistema de posicionamiento creado por Cedetel, basado en RFID (Identificación por Radiofrecuencia) y cuyo uso es destinado a la localización de equipos médicos en hospitales.

Igual que en el caso de Ubisense, a los equipos que se desea conocer su ubicación se les acopla un tag que emite señales de RF que son recibidas por elementos fijos lectores instalados en el centro hospitalario. Todo el sistema es controlado mediante una aplicación web, donde se dan de alta los tags y los equipos médicos a ubicar y donde se dispone de un mapa de la zona en el cual se posicionan los equipos con un punto. Adicionalmente, se pueden configurar alarmas como, por ejemplo, si un equipo se sale de la zona de donde debe estar.

En cuanto a la precisión del sistema la media es de entre 3 y 5 metros que puede ser reducida aumentando el número de sensores lectores y su colocación.



Ilustración 2.3-4 Tag compacto del sistema WiLocT.

## 3. Ingeniería de concepción.

En este capítulo se exponen diferentes alternativas de sistemas de localización de robots y se destacan sus principales ventajas e inconvenientes.

### 3.1. Sistemas de localización relativos.

La principal ventaja de estos sistemas es que todo el proceso de localización esta centrado en el mismo nodo, lo cual aumenta su autonomía y elimina la necesidad de desplegar un sistema de balizas como en los basado en puntos de referencia.

#### 3.1.1. Sistemas de localización odométrica.

Es el método más común para la estimación de la localización para un vehículo móvil autónomo constituye uno de los requerimientos esenciales para su navegación. Este sistema integra la trayectoria, estimando a posición y orientación del robot por el movimiento del las ruedas motrices.

El principal inconveniente radica en la acumulación de errores a lo largo de la trayectoria, siendo preciso desarrollar sistemas de corrección, que actúen de forma periódica o en determinados puntos críticos del camino.

Estos errores son asociados a los sistemas mecánicos empleados y que son acumulativas en el tiempo, y otras debidas a deslizamientos, falta de simetría de las ruedas, etc. Estos errores provocan un aumento de la incertidumbre en la posición y orientación, pudiendo llegar un punto en el que la información proporcionado por el sistema odométrico no sea acorde con la realidad.

Podemos considerar que los sistemas basados en la odometría proporcionan ayudas válidas, que deberán ser complementadas con otros procesos que actualicen la posición y que permitan mantener el error dentro de unos márgenes aceptables para la aplicación. Al proceso que se encarga de determinar la posición y orientación del robot móvil, utilizando la información procedente de los sensores externos, se le suele denominar “relocalización”.

Para el proceso de localización de un robot móvil a través de sus sistema sensorial parte de un conocimiento previo del entorno de trabajo y de la representación geométrica de la información asociada a dicho entorno. Para completar la información de localización en todos los grados de libertad del sistema, será preciso emplear un modelo de representación de la incertidumbre y un método de integración que permita manejar dicha representación.

En los últimos años ha habido un gran desarrollo, tanto de sensores, como de algoritmos para la interpretación de la información proporcionada por cada uno de ellos. Sin embargo, ningún sensor es capaz de resolver toda las necesidades de percepción considerando que su información es parcial.

Los sensores de ultrasonidos usados en la medición de distancias, su precisión depende de la forma geométrica de la superficie de reflexión, grado de inclinación de ésta, etc. Es preciso considerar que la información sensorial se encuentra sometida a una incertidumbre debida a la imprecisión del sensor, errores de calibración, ruido de la medida, etc.

La exactitud de la odometría en las plataformas móviles depende básicamente del diseño cinemático y de las siguientes consideraciones:

- los vehículos con ruedas pequeñas son más propensos a la acumulación de error en la orientación, que los que presentan ruedas grandes.
- la rueda ideal debería estar fabricada de aluminio y cubierta de una delgada capa de goma para una mejor tracción, al mejorar la superficie de contacto con el suelo.
- la configuración diferencial considera el desplazamiento y la velocidad del vehículo como los valores medios asociados a la rueda derecha e izquierda.
- la configuración en triciclo utiliza una única rueda directriz, y dos ruedas pasivas (se trata de una configuración básicamente utilizada en AGVs debido a su simplicidad), ello provoca un desplazamiento del centro de gravedad del vehículo en movimientos en planos inclinados, así como pérdidas de tracción.
- la configuración synchro-drive proporciona mejores resultados que la diferencial y triciclo, sobre todo en suelos irregulares, debido a que todas las ruedas se encuentran acopladas, girando en la misma dirección y a la misma velocidad.

Los errores asociados a la odometría los podemos clasificar en dos grupos: errores sistemáticos (distintos diámetros en las ruedas del vehículo, diferencias entre el diámetro nominal de la rueda y su valor real, desalineamientos de las ruedas, resolución del encoder, etc.) y errores no sistemáticos (deslizamientos con la superficie de contacto, pérdida del punto de contacto de la rueda con el suelo, posibles fuerzas internas y externas aplicadas sobre las ruedas, etc.).

Esta distinción entre los errores es importante de cara a poder evaluar el error cometido por la odometría. Los errores sistemáticos provocan una acumulación constante de error, pudiendo considerarse como la principal fuente de error en entornos interiores; por el contrario, si el vehículo se mueve sobre superficies rugosas con muchas irregularidades, los errores no sistemáticos son los que predominan, apareciendo de forma inesperada y provocando alteraciones importantes en la posición del robot. Normalmente, cuando un robot móvil presenta un sistema híbrido de localización formado

por odometría y marcas, la frecuencia de aparición de éstas últimas depende de los errores que presente el sistema.

### **3.1.2. Sistemas de localización inerciales.**

Otra alternativa consiste en substituir la odometría por sistemas de navegación inercial, cuyo funcionamiento esta asociado al uso de acelerómetros para el calculo de las variaciones de velocidad y de giroscopios para los cambios de orientación. Su precisión depende de la calidad de los elementos que los conformen, estando también sujetos a errores del mismo tipo.

Los sistemas de navegación inercial (INS) inicialmente fueron desarrollados para su empleo en aeronáutica. Básicamente su modo de funcionamiento está asociado a una medición continua de la aceleración en cada uno de los tres ejes direccionales. Una plataforma giroscópica estabilizada es utilizada para mantener la orientación de los tres acelerómetros empleados en el proceso. Los giroscopios son los encargados de proporcionar información respecto a la variación angular, mientras que los acelerómetros facilitan las variaciones de velocidad.

Para obtener la orientación y posición del vehículo será preciso integrar una y dos veces respectivamente la información relacionada con las variaciones angulares y de velocidad lineal, eso conduce a que cualquier pequeño error cometido en la medida puede causar un crecimiento importante en el error de la posición y orientación del vehículo.

El principal atractivo del uso de los sistemas de navegación inercial está asociado a que no precisan de información externa del movimiento del vehículo para su posicionamiento, así como de su rapidez en el flujo de medidas enviadas. Por contrario, presenta como principal inconveniente el elevado coste de fabricación y posterior mantenimiento.

## **3.2. Sistemas de localización basados en puntos de referencia.**

Estos sistema tienen la principal ventaja que no acumulan errores, ya que la posición se calcula sin tener en cuenta la posición anterior del robot, Sin embargo, este tipo de sistema tienden a ser mas costosos que los sistemas de localización relativos al ser necesario desplegar balizas en el entorno.

Este sistema se basa en la medición de la potencia de emisión de tres o mas marcas emisoras, siendo conocida su ubicación en el entorno de trabajo, presenta un sistema de localización externo al robot, que es capaz de cubrir un área de trabajo restringido.

### 3.2.1. Principio de localización.

Define que tipo de medida respecto a las balizas es necesaria para localizar al robot, así como la formulación necesaria para convertir dichas medidas en una posición del entorno.

#### 3.2.1.1. Dirección de Llegada.

Del inglés “Direction Of Arrival”, son mundialmente conocidos como sistemas DOA (también se les conoce por “Angle Of Arrival” o AoA).

Este tipo de sistemas se basan en calcular el ángulo con el que se recibe la señal de referencia para determinar la dirección en la que se encuentra la baliza.

Inicialmente, fueron utilizados como el principal sistema de localización en los barcos (antes de la llegada del GPS). Utilizando una brújula para conocer la dirección del norte magnético y antenas muy direccionales que se van rotando hasta que la potencia recibida es máxima respecto a diferentes balizas localizadas en puntos conocidos, el sistema es capaz de determinar el ángulo existente entre el norte magnético y la baliza.

Dado que el norte magnético es muy estable (para una determinada región), basta con conocer el ángulo existente entre el norte magnético y una baliza para poder trazar una línea desde dicha baliza que cumpla con dicho ángulo. Repitiendo el proceso respecto a una segunda baliza, se obtiene un punto (en donde se intersecan ambas rectas) que es la posición del robot.

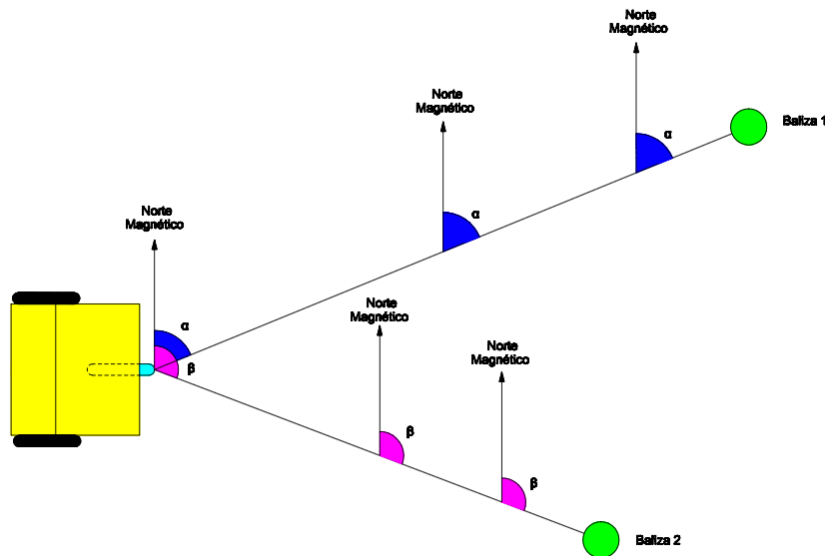


Ilustración 3.2-1 Sistema de localización DOA por dirección de llegada.

### 3.2.1.2. Tiempo de llegada.

Del inglés “Time Of Arrival”, son mundialmente conocidos como sistemas TOA (también aparecen referenciados como “Time of Flight” o TOF).

Estos sistemas utilizan el tiempo que tarda en viajar la señal desde el emisor al receptor para conocer la distancia entre ambos puntos y obtener la posición del objeto. Por cada tiempo obtenido, se traza una circunferencia o esfera centrada en el transmisor para posteriormente intersecar dichas formas para determinar la posición del objeto.

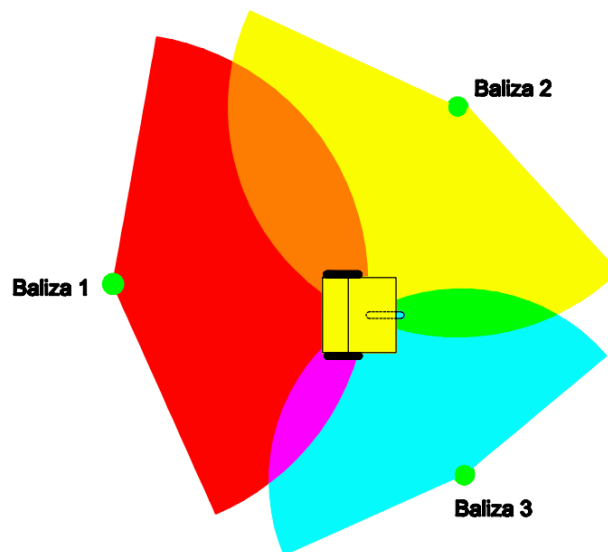


Ilustración 3.2-2 Sistema de localización por TOA tiempo de llegada.

El dispositivo más popular basado en este principio de funcionamiento es el GPS. En este sistema, todos los satélites están sincronizados entre ellos y cada satélite emite un patrón que le identifica. El receptor utiliza un reloj que se sincroniza con el satélite. Cuando el receptor captura los patrones provenientes de varios satélites, este es capaz de determinar la deriva de su reloj respecto al de los satélites y, al detectar los patrones enviados desfasados, el receptor puede estimar la diferencia del tiempo de propagación de la señal entre los diferentes satélites y el receptor, pudiendo obtener su posición en base a esta información.

La principal ventaja de estos sistemas es que el proceso para obtener la posición del objeto es bastante simple al tratarse de intersecar circunferencias o esferas. Sin embargo, se requiere que exista una sincronización entre emisores y receptores, lo cual incrementa considerablemente la complejidad del sistema.

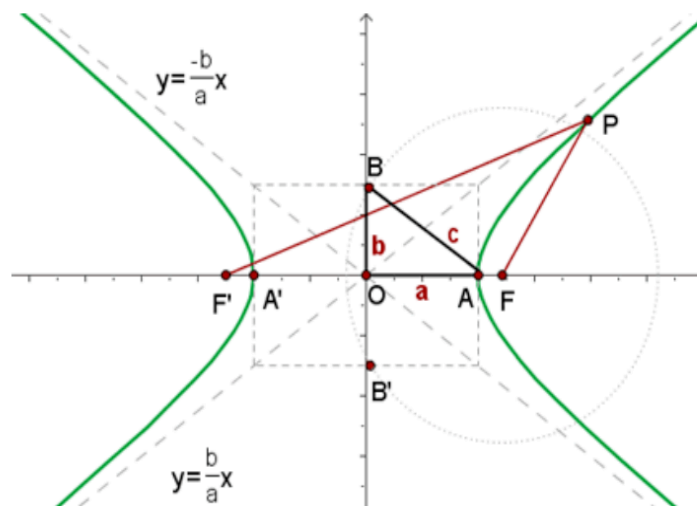


### 3.2.1.3. Diferencia de tiempo de llegada.

Del inglés “Differential Times Of Arrival”, son mundialmente conocidos como sistemas DTOA (también aparecen referenciados como “Time Difference of Arrival” o TDOA).

Este tipo de sistemas basan su funcionamiento en calcular la diferencia del tiempo de propagación de una señal generada por una baliza y varios puntos de referencia. Por cada diferencia de tiempos capturada en cada par de puntos de referencia, el sistema conoce que la posición del transmisor se encuentra en algún punto de una hipérbola (o hiperboloide para el caso de tres dimensiones) cuyos focos ( $F$  y  $F'$  en la figura) son los puntos de referencia y cuyos vértices ( $A$  y  $A'$ ) se encuentran a la mitad de la diferencia del tiempo medido multiplicado por la velocidad de propagación de la señal transmitida.

Nota: La hipérbola es la curva (superficie en el caso del hiperboloide) cuyo valor absoluto de la diferencia de sus distancias a dos puntos fijos, llamados focos, es constante e igual a la distancia entre vértices.



**Ilustración 3.2-3** Sistema de localización por TDOA diferencia de tiempo de llegada.

Si el sistema dispone de varios puntos de referencia, este puede intersectar las hipérbolas o hiperboloides generados por cada par de receptores y de este modo conocer la posición del emisor.

Uno de los sistemas que se basan en este principio es el que utiliza el ser humano para localizar sonidos. Entre los múltiples factores que se tienen en cuenta para determinar la procedencia de un sonido, uno de los más importantes es el desfase que percibimos en la señal acústica con nuestros dos oídos, el cual nos permite determinar la dirección de la que procede.

El principal inconveniente de este tipo de sistemas es que son relativamente complejos, ya que se necesita que los nodos receptores o transmisores estén sincronizados entre ellos, pero son más sencillos que los sistemas TOA, los cuales necesitan que exista una fuerte sincronización entre el sistema emisor y el sistema receptor. Además, la necesidad de operar con hipérbolas o hiperboloides hace que el proceso para calcular la posición del objeto conlleve un alto coste computacional.

#### **3.2.1.4. Intensidad de la señal recibida.**

Del inglés "Receiver Signal Strength Indicator RSSI" Estas medidas están basadas en la intensidad de señal en el receptor. Dos formas para tratar los datos basados en RSS.

- Basados en las condiciones de propagación incluyendo ecuación de Friss.
- Los valores medidos de RSSI son preprocesados y guardados en una base de datos, durante la localización se comparan los valores de RSS actuales con los de la base de datos.

Esta técnica es la más popular en muchas redes inalámbricas, normalmente la información sobre la intensidad ya viene dada en una parte del mensaje del reader.

Esta técnica es muy dependiente del escenario en el cual se realizan la localización, para escenarios diferentes los resultados varían mucho. También susceptibles a los "rebotes" es decir objetos que estén cerca pueden alterar los resultados. Los métodos de localización basados en RSS dependen de la calidad del canal de propagación y este es muy sensible a los efectos del "Faldin"

### **3.2.2. Tecnología.**

Tipo de tecnología utilizada para conseguir la medida utilizada por el algoritmo de localización.

#### **3.2.2.1. Radiofrecuencia.**

Este tipo de sistemas utilizan ondas de radio como señales de referencia para extraer la información relativa a las balizas. Dos aproximaciones son las más comúnmente utilizadas:

##### Medir la potencia de la señal recibida:

Este es el método más sencillo para extraer la información relativa a una baliza en base a una señal de radiofrecuencia. La mayoría de los sistemas de radiofrecuencia proporcionan junto con la señal recibida una medida de la potencia recibida (con el fin de poder determinar la calidad de la señal recibida).

La potencia recibida depende, entre otros factores, de la distancia entre el transmisor y el receptor, por lo que se puede hacer una estimación de esta distancia asumiendo como constante el resto de factores. Esto permite implementar algoritmos TOA para localizar al objeto.

#### Medir el retardo de la señal recibida:

Si se disponen de varios transmisores y/o receptores sincronizados entre sí, es posible extraer el tiempo que tarda en propagarse la señal de referencia entre transmisor y receptor (o la diferencia de tiempo entre varios receptores o respecto a varios transmisores).

La señal de radiofrecuencia viaja a la velocidad de la luz ( $3 \cdot 10^8$  m/s), lo que hace que para medir de forma precisa el retardo de propagación sea necesario procesar la fase de la señal recibida. Para hacernos una idea del motivo que nos llega a medir este tiempo de esta manera, si un sistema fuese capaz de medir con una precisión de 1 ns el tiempo de propagación de una señal, estaría cometiendo un error de  $\pm 30$  cm en la estimación de dicha medida. Sin embargo, si el sistema estimase la diferencia de fase y cometiese un error del 10% (sobre una señal de 2GHz), dicho error sería de 0,05 ns (20 veces menos que en el caso anterior).

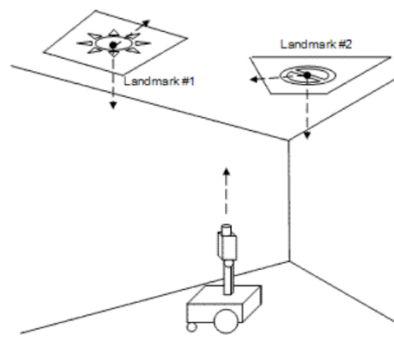
Hay que resaltar que en este tipo de sistemas puede ser necesario contemplar el caso de que el tiempo a medir sea mayor que el propio periodo de la señal de radiofrecuencia, y que la señal recibida en el receptor sea la suma de la señal que viaja de forma directa desde el transmisor al receptor (la que aporta información útil) junto con los rebotes de la misma señal en el entorno (que perturban la fase de la señal recibida). Esto hace que sea necesario un patrón que permita diferenciar estos casos lo cual incrementa la complejidad del sistema.

El principal inconveniente de este tipo de sistemas es que son bastantes más complejos que el resto de sistemas (a excepción del basado en procesamiento de imagen) y que demandan una gran capacidad computacional, lo que hace que su uso esté poco extendido en aplicaciones en interiores.

#### **3.2.2.2. Imagen**

Esta tecnología se basa en capturar las medidas utilizadas durante el proceso de localización de marcas o puntos de referencia a través de una cámara .

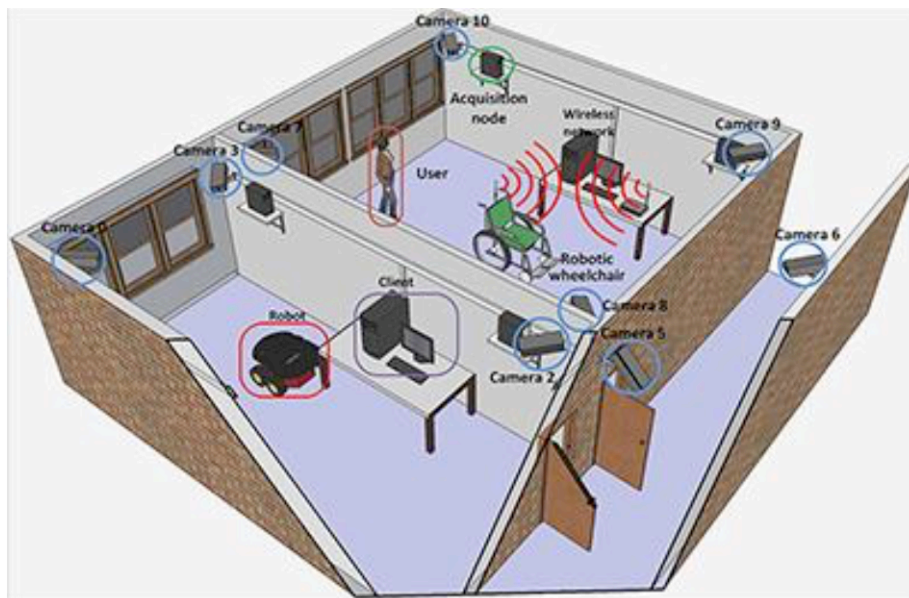
La idea principal de este tipo de sistemas consiste en desplegar una serie de marcas (en el techo, paredes o suelo del entorno) y que se busquen dichos patrones para determinar su orientación y posición respecto a dichas referencias, implementando, generalmente, sistemas DOA.



**Il·lustració 3.2-4** Localització per imatge.

Los puntos más negativos de este tipo de sistema es que son los sistemas de localización que requieren mayor capacidad de cálculo (ya que necesitan procesar imágenes en tiempo real), son bastante costosos y demandan una gran cantidad de energía para poder funcionar.

Existe otra aproximación basada en el mismo principio que intenta mitigar estas limitaciones distribuyendo una red de cámaras fijas conectadas a un servidor centralizado. Al estar las cámaras fijas, estas pueden alimentarse desde la red eléctrica, y al disponer de un nodo que hace de centralita, se puede implementar en dicho nodo un dispositivo especializado en procesamiento de imagen, el cual se encarga de determinar la posición de los objetos de interés en el entorno y reportárselo a aquellos que lo necesiten (reduciendo la carga computacional y el coste energético de los nodos en el sistema):



**Il·lustració 3.2-5** sistema centralizado basado en procesamiento de imágenes.

El problema de esta aproximación es que al estar toda la capacidad de cálculo concentrada en un solo nodo y necesitar reportar a cada nodo su posición, se reduce la escalabilidad del sistema, se reduce la autonomía de cada nodo y se reduce la confidencialidad del sistema (al comunicarse por radio la posición de cada nodo).

### 3.2.2.3. Ultrasonido

La principal ventaja de este tipo de sistemas es que al enviarse una señal acústica, dicha señal viaja a unos 340 m/s ( $\sim 10^5$  veces más lenta que una señal de radiofrecuencia). Esto simplifica enormemente el sistema, permitiendo capturar el tiempo de propagación con simples contadores. Por ejemplo, si la precisión del sistema cuando captura la señal de referencia es de 1  $\mu$ s, el error cometido en la medida capturada es de  $\pm 0,34$  mm.

La gran diferencia existente entre la velocidad de propagación de la señal de ultrasonidos y la señal de radiofrecuencia permite, además, implementar sistemas TOA de forma sencilla. Para ello, el sistema emite junto con la señal de ultrasonido/sonido la señal de radiofrecuencia, la cual llega al receptor tras unos pocos nanosegundos. Como la señal de ultrasonidos se recibe tras unos cuantos microsegundos, se puede asumir que cuando se empezó a recibir la señal de radio es el instante en que se empezó a transmitir la señal de referencia, ya que el error del orden de nanosegundos representa una ínfima parte de la medida capturada.

Este tipo de sistemas se suelen implementar con ultrasonidos, con el principal objetivo de no generar una fuente ruidosa molesta para el ser humano en el entorno de la aplicación. Sin embargo, para algunas aplicaciones (como los sistemas para localizar fuentes de ruido) se sustituyen los transmisores y receptores de ultrasonidos por micrófonos y altavoces.

Actualmente este tipo de sistemas se restringen casi exclusivamente al ámbito de investigación. Pese al bajo coste de los transductores y la alta precisión permitida por el sistema, el escaso alcance de las balizas hace que desplegar este tipo de sistemas no sea sencillo ni eficiente. Además, aunque la precisión proporcionada es superior al resto de sistemas, actualmente apenas existen aplicaciones que demanden dicha precisión, siendo más eficiente recurrir a combinar otros sistemas de localización cuando dicha precisión es necesaria.

En el caso de los sistemas de visión 2D no se puede obtenerse información de profundidad, además presentan limitaciones de uso según la iluminación, óptica, etc.

### 3.3. Elección de la técnica a implementar.

Para la realización del proyecto de control y guiado del robot se ha usado un método de localización basado en odometría, con los sensores que incorpora el robot SR1, ya que es el sistema de control más sencillo que existe aunque tiene las desventajas citadas en este mismo capítulo.

También se ha utilizado un sistema de localización basado en puntos referenciales, estos puntos referenciales son del tipo de radiofrecuencia, concretamente cuatro balizas ZigBee, se ha utilizado el principio de localización de intensidad de la señal recibida, Los valores medidos de RSSI son preprocesados y guardados en una base de datos, durante la localización se comparan los valores de RSS actuales con los de la base de datos.

La ventajas de estos módulos ZigBee son:

- Bajo coste.
- Número de materiales a utilizar reducido.
- Prioridad en la duración de las baterías que se implementarán.
- Alcance de la señal aceptable.
- Disminución de los problemas que causan los obstáculos en las señales.
- No son necesarias altas tasas de envío de datos.
- Acceso al indicador de intensidad de señal recibida (RSSI).

## 4. Ingeniería de desarrollo.

En este capítulo se desarrollan todos los elementos que intervienen en el TFG.

### 4.1. Localización odométrica.

La odometría se utiliza para estimar la posición relativa de un robot móvil, mediante el modelo matemático y la información que obtenemos de los sensores, se puede llegar a calcular la posición del robot móvil con una cierta incertidumbre.

#### 4.1.1. Modelo cinemático del robot.

El robot cuenta con dos ruedas de tracción donde están acoplados dos motores y una rueda de estabilización. La traslación y rotación se determinan por el movimiento independiente de cada una de las ruedas de tracción.

Las ecuaciones cinemáticas del punto medio del eje entre las ruedas activas, referidas a un sistema de referencia global son:

$$\dot{x} = v(t)\cos(\theta(t)) \quad [1]$$

$$\dot{y} = v(t)\sin(\theta(t)) \quad [2]$$

$$\dot{\theta} = \omega(t) \quad [3]$$

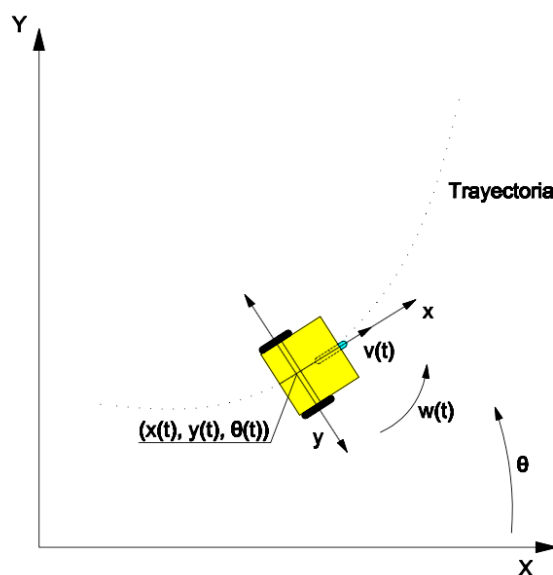


Ilustración 4.1-1 Modelo cinemático del Robot SR1.

La posición y orientación del robot móvil se obtiene integrando las velocidades de éste en un periodo de tiempo  $\Delta t$ .

$$x(t) = x(t_0) + \int_{\Delta t} v(t) \cos(\theta(t)) dt \quad [4]$$

$$y(t) = y(t_0) + \int_{\Delta t} v(t) \text{sen}(\theta(t)) dt \quad [5]$$

$$\theta(t) = \theta(t_0) + \int_{\Delta t} \omega(t) dt \quad [6]$$

Si el periodo de observación  $\Delta t$ , tiende a 0, entonces las integrales anteriores pueden ser remplazadas por los desplazamientos diferenciales  $\Delta x$ ,  $\Delta y$ ,  $\Delta \theta$  (El límite de las integrales de las variables  $x(t)$ ,  $y(t)$  y  $\theta(t)$ , cuando  $\Delta t$ , tiende a 0).

Si se mantiene una frecuencia de muestreo constante y elevada sobre la odometría de un móvil, se puede estimar la posición y orientación del mismo mediante las siguientes ecuaciones en diferencia:

$$x_k = x_{k-1} + \Delta x_k \quad [7]$$

$$y_k = y_{k-1} + \Delta y_k \quad [8]$$

$$\theta_k = \theta_{k-1} + \Delta \theta_k \quad [9]$$

Además se puede considerar que la velocidad angular en cada una de las ruedas se mantiene constante durante el periodo de muestreo. En la figura 3,  $\Delta S_k$  es la distancia recorrida por el centro del robot en el intervalo  $(k-1, k)$ , y  $\Delta \theta_k$  es el cambio de orientación durante el mismo intervalo.

Cálculo de la distancia recorrida:

$$d = \frac{R \cdot 2\pi}{ppv} \cdot p \quad [10]$$

donde:

$d$  es la distancia recorrida.

$R$  es el radio de la rueda.

$ppv$  son los pulsos por vuelta.

$p$  es el número de pulsos.



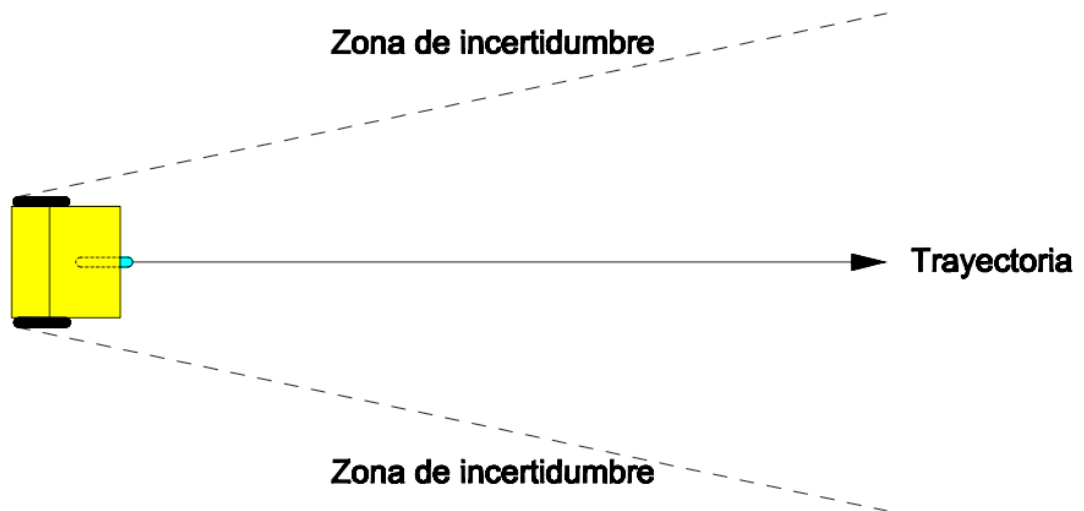


Ilustración 4.1-2 Aumento de la incertidumbre en la localización odométrico.

## 4.2. Localización por potencia de señal recibida RSSI.

La localización por medición de la potencia de recepción es un sistema que no arrastra errores, ya que la posición se calcula sin tener en cuenta la posición anterior del robot, evalúa la potencia de la señal recibida por una antena colocada en el espacio libre, cuando se transmite desde otra antena colocada a una distancia de una potencia conocida.

### 4.2.1. Ecuación de FRISS.

La ecuación de FRISS es el modelo matemático de la propagación para el espacio libre, donde se predice la caída de potencia en relación con la distancia de separación entre el receptor y el transmisor y la frecuencia de operación.

$$Prx = Ptx \cdot Gtx \cdot Grx \cdot (\lambda/4\pi d)^2 \quad [11]$$

donde:

$P_{rx}$  es la potencia en watts de la señal recibida.

$P_{tx}$  es la potencia de la antena conocida.

$G_{tx}$  es la ganancia de la antena de transmisión.

$G_{rx}$  es la ganancia de la antena de recepción.

$\lambda$  es longitud de onda de la señal.

$d$  es la distancia.

$$\lambda = \frac{c}{f} \quad [12]$$

f es la frecuencia.

c es la velocidad de la luz en el vacío (300Km/s.).

#### 4.2.2. Índice de la potencia de la señal recibida (RSSI)

El indicador de intensidad de señal recibida (Received Signal Strength Indicator, RSSI) es la técnica de estimación en interiores más común que se utiliza en redes inalámbricas de área local y personal (WLAP y WPAN). Esta técnica usa la activación de la señal de radio para determinar la distancia entre nodos, siendo necesario como mínimo tres nodos para la localización fiable.

Definido como la relación de potencia de la señal recibida y la potencia de referencia de 1 mW ( $P_{ref}=1$  mW) en dBm.

$$RSSI = 10 \cdot \log(P_{rx} / P_{ref}) \quad [13]$$

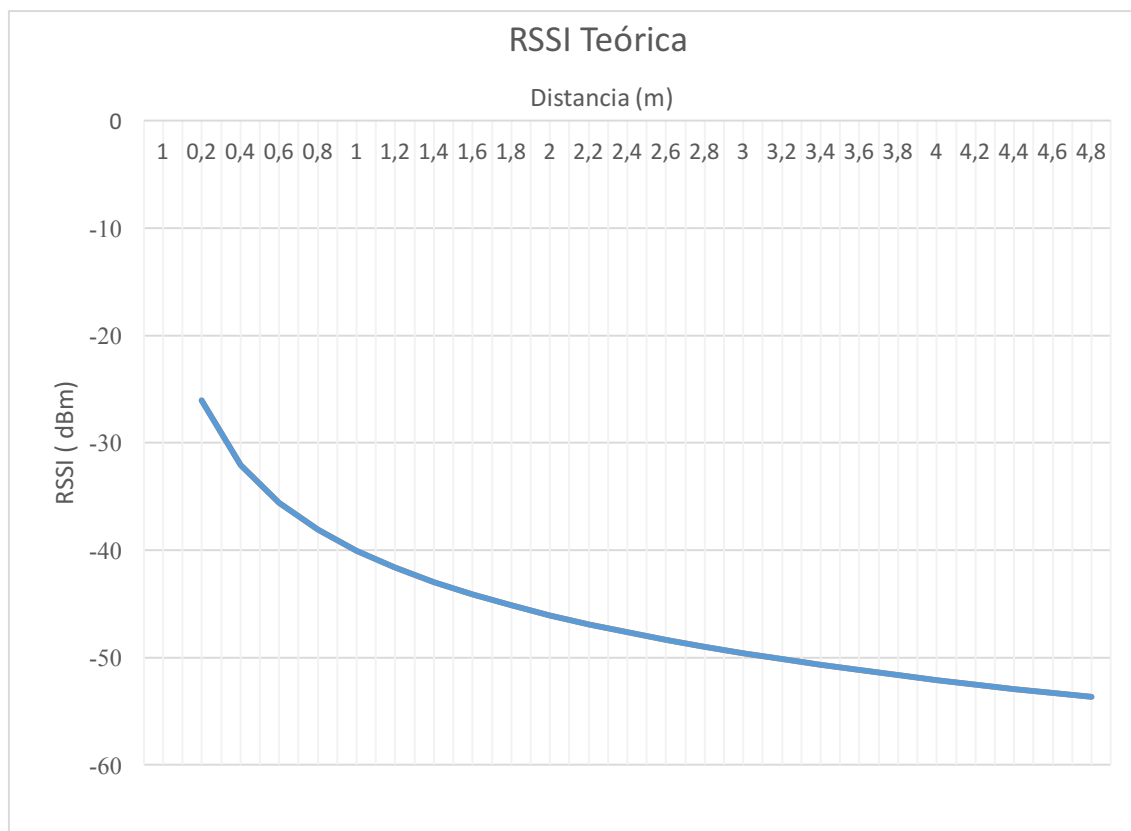


Ilustración 4.2-1 Valores teóricos RSSI en función de la distancia.

Todos los dispositivos WLAN y WPAN disponen de esta característica para leer el nivel de atenuación de la señal por lo que nos necesario ningún tipo de hardware adicional. Esto convierte esta técnica en la más que económica.

Sin embargo, se siguen obteniendo errores a causa de los problemas comunes en todas las técnicas de estimación. La relación entre distancia y potencia de la señal recibida se ve afectada por las reflexiones y el multirrayecto, entre otras causas.

Existen varias técnicas para la localización pasadas en la intensidad de la señal recibida, Como la triangulación o el reconocimiento de patrones.

### 4.2.3. Técnica para la localización.

La técnica de localización implementada en este proyecto es una combinación de varias técnicas, la primera parte se basa en el reconocimiento de patrones de huellas de radiación, esta técnica es implementada en dos fases, muestreo y operación.

El muestreo consiste en una fase en la que se genera una base de datos de valores RSSI obtenidos de las señales que emiten los nodos localizadores en determinadas posiciones. Este enfoque se conoce como centralizado, pero también es posible un enfoque conocido como "orientado a la red", en el que lo que se almacena es la potencia con la que se reciben las señales emitidas por el nodo a localizar, en los nodos localizadores. El entorno en el que se quiere desplegar el sistema se divide en una cuadrícula como la de la ilustración 4.1-4, en cada celda de la cuadrícula se obtiene el valor RSSI que recibe el nodo móvil de cada uno de los nodo baliza.

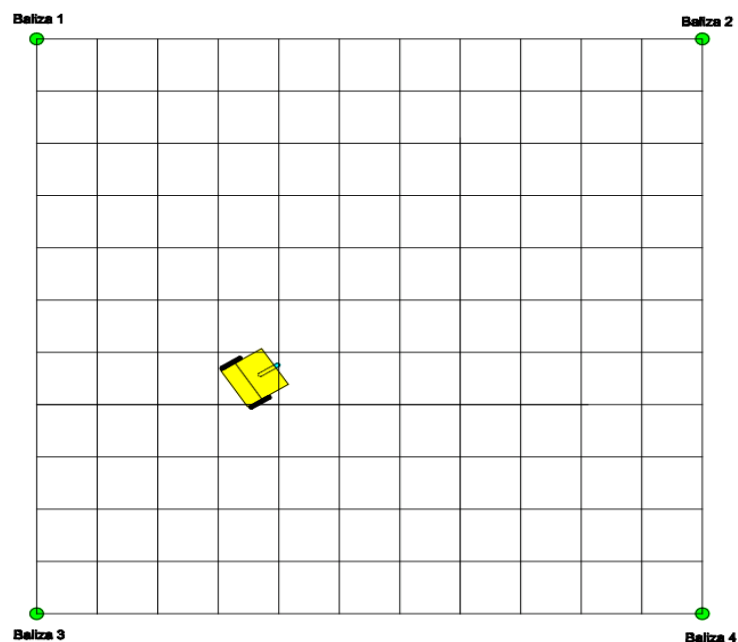


Ilustración 4.2-2 Localización basada en reconocimiento de patrones.

Al iniciar la fase de operación el software interpola los valores obtenidos en la fase de muestreo con el fin de obtener mas resolución en la búsqueda de posición del robot móvil y el resultado es una matriz de potencias RSSI respecto a las coordenadas X,Y de la cuadrícula por cada nodo baliza ( $M_1$ ,  $M_2$ ,  $M_3$  y  $M_4$ ).

El siguiente paso es solicitar la medida RSSI al nodo móvil respecto las balizas y se obtiene un valor  $X_1$ ,  $X_2$ ,  $X_3$  y  $X_4$  cada valor representa la potencia recibida por cada una de las cuatro nodos balizas que consta el sistema.

Seguidamente se le resta a cada una de las cuatro matrices donde están almacenadas de potencias RSSI su respectivo valor  $X_n$  obtenidos en el punto anterior.

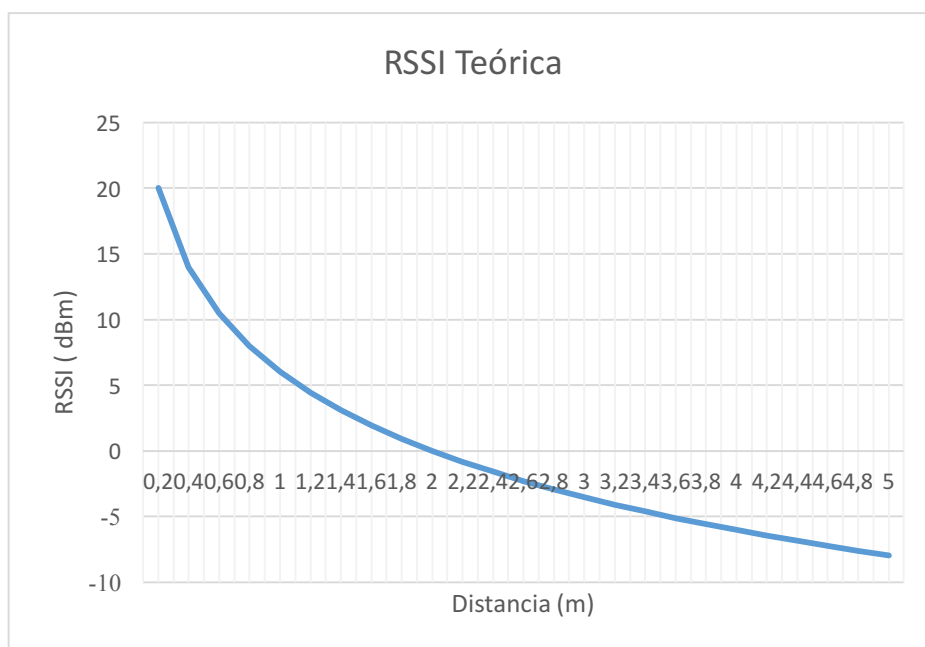
$$MR_1 = M_1 - X_1 \quad [14]$$

$$MR_2 = M_2 - X_2 \quad [15]$$

$$MR_3 = M_3 - X_3 \quad [16]$$

$$MR_4 = M_4 - X_4 \quad [17]$$

En las coordenadas donde cada matriz resultante  $MR_n$  el valor RSSI es 0 significa que coinciden el valor RSSI de las matrices  $M_n$  y el valor  $X_n$ .



**Ilustración 4.2-3** Valor RSSI en función de la distancia de la matriz  $MR_n - X_n$  a una distancia de 2m.

Posteriormente se convierten los valores de cada matriz  $MR_n$  a valores absolutos convirtiendo los valores negativos en positivos ( $MA_n$ ) y se suman las cuatro matrices.

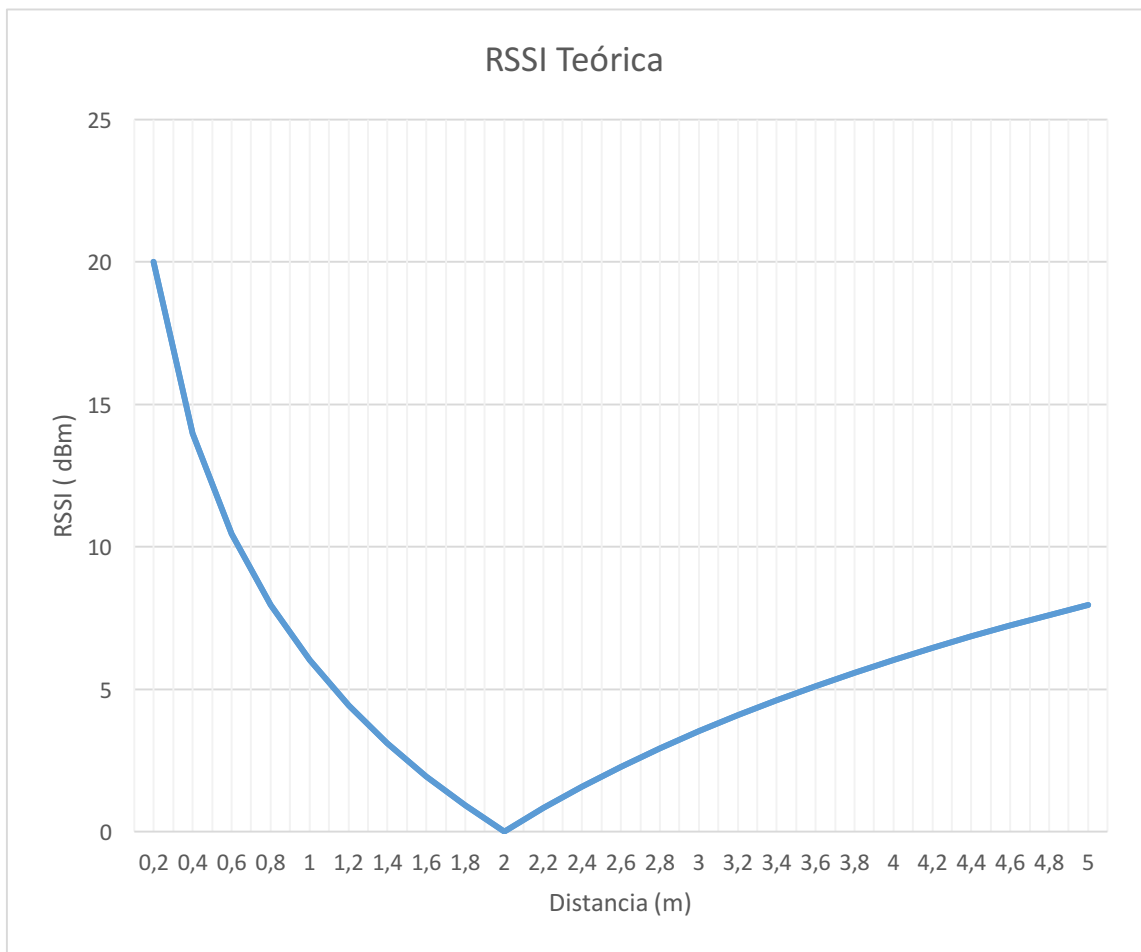
$$MA_1 = |MR_1| \quad [18]$$

$$MA_2 = |MR_2| \quad [19]$$

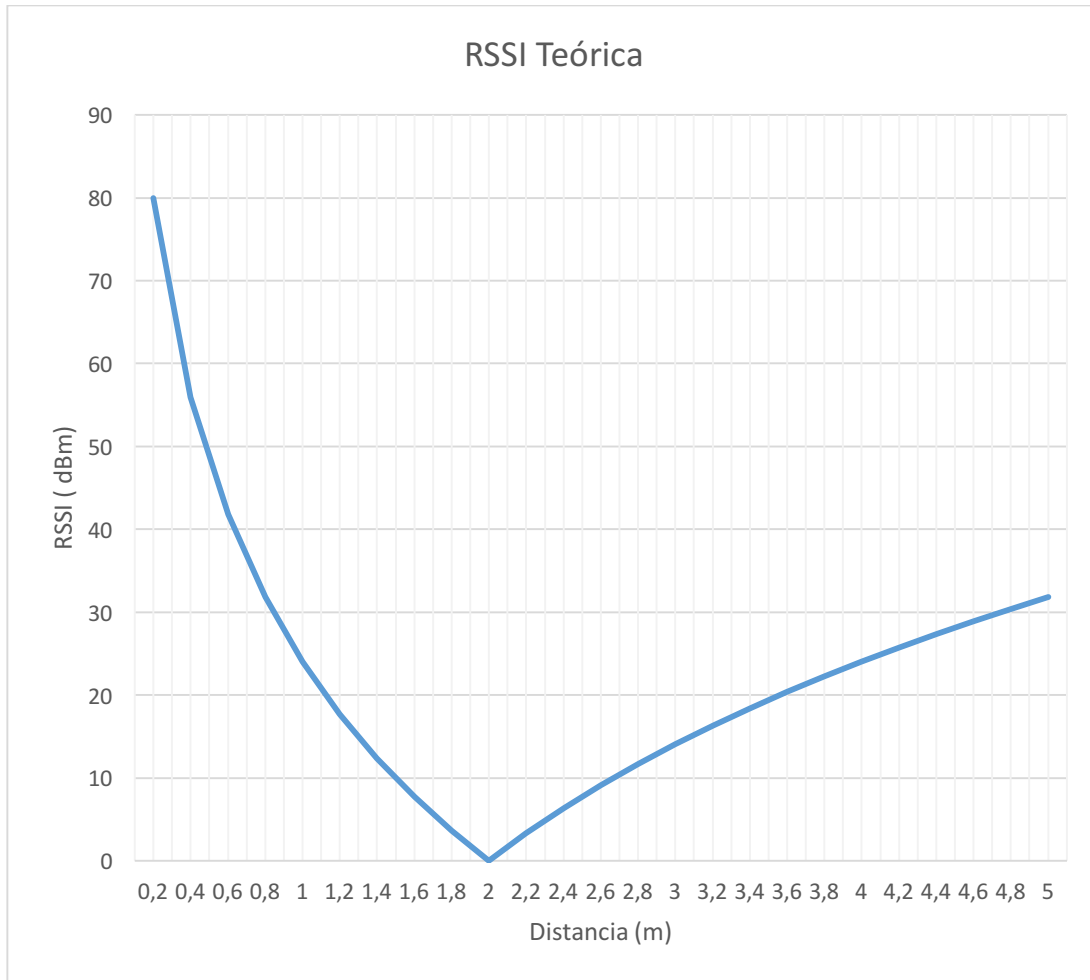
$$MA_3 = |MR_3| \quad [20]$$

$$MA_4 = |MR_4| \quad [21]$$

$$MT = MA_1 + MA_2 + MA_3 + MA_4 \quad [22]$$



**Ilustración 4.2-4** Valor RSSI en función de la distancia de la matriz  $MA_n$  a una distancia de 2m.



**Ilustración 4.2-5** Valor RSSI en función de la distancia de la matriz  $MT_n$  a una distancia de 2m.

Idealmente en la matriz MT en la posición donde el valor RSSI fuera 0 coincidiría con la posición X, Y de la baliza móvil.

Cuanto más elevado sea el valor de RSSI más alejados se está de la posición del robot.

### 4.3. Hardware del sistema.

En este apartado se describen los elementos hardware que intervienen en el control y guiado del robot móvil.

#### 4.3.1. Transceptor inalámbrico.

El transceptor inalámbrico escogido presente en todos los tipos de nodos del sistema es el mismo componente para todos ellos. De esta forma se garantiza una completa compatibilidad y fiabilidad entre elementos del sistema.

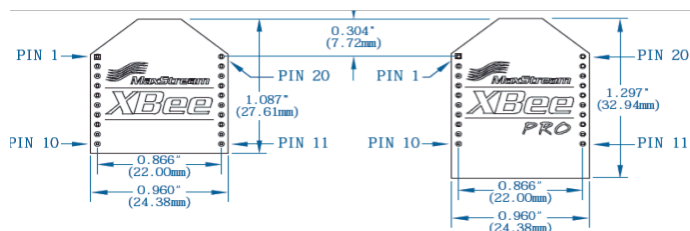
De entre la diversidad de fabricantes que disponen de módulos ZigBee en el mercado, los módulos XBee de DIGI International son los más populares. Se distribuyen diversos tipos según los estándares a los que van dirigidos pero los que interesan son los módulos XBee y XBee PRO 802.15.4 OEM RF, también conocidos como XBee Series 1, que son los que operan con el estándar IEEE 802.15.4.

Los módulos XBee son unos módulos de tamaño pequeño ideales para aplicaciones inalámbricas de bajo consumo y baja tasa de envío de datos. Están diseñados a partir de un chip Ember que incorpora el módulo de radio de 2,4 GHz y un microcontrolador interno. La velocidad de las tasas de información se comprende entre 20 Kbps y 250 Kbps.



**Ilustración 4.3-1** Módulos XBee y XBee Pro Series 1.

Existen dos versiones de módulos XBee de la Serie 1: los módulos XBee y los módulos XBee PRO. La diferencia entre ellos es, principalmente, la potencia de transmisión. En los XBee se obtiene una potencia de transmisión de 1 mW (0 dBm) lo que proporciona un alcance de 30 metros en interiores y 100 metros en exteriores. Por otra parte, los módulos XBee PRO disponen una potencia superior, siendo ésta de 100 mW (20 dBm), derivando en un alcance máximo de 100 metros en interiores y 1500 metros en exteriores, pero el consumo es 5 veces mayor a la hora de transmitir en comparación con los XBee. Según las normativas la potencia máxima a la que se puede transmitir es de 100 mW (20 dBm) en Europa, que coincide con la potencia máxima del módulo XBee PRO.



**Ilustración 4.3-2** Dimensionado de módulos Xbee y Xbee Pro Series 1.

Ambos módulos operan a una tensión de 3,3 V y tienen un consumo de 50 mA a la hora de recibir y 45 mA al transmitir (Xbee) y 215 mA (Xbee PRO). El consumo en estado de reposo, la gran ventaja de estos módulos que alarga la vida útil de las baterías, es inferior a 10 µA.

En cuanto a su integración, los módulos Xbee vienen acompañados de un soporte de pines para ser directamente montados sobre él y ahorrar tiempo en soldaduras.

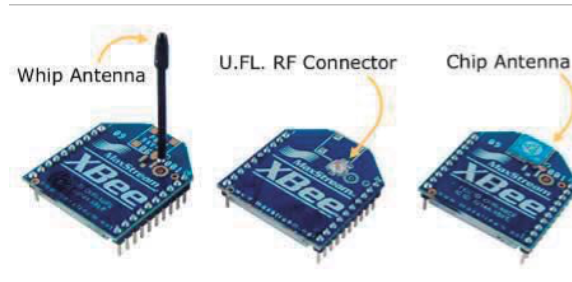
La funcionalidad de cada uno de los 20 pines que tiene el módulo se muestra en la siguiente tabla:

**Tabla 1** Funcionalidad de los pines de los módulos Xbee.

| Pin # | Nombre                 | Dirección | Descripción   |
|-------|------------------------|-----------|---|
| 1     | VCC                    | -         | Tensión de Alimentación   |
| 2     | DOUT                   | Salida    | Datos de Salida UART  |
| 3     | DIN/CONFIG             | Entrada   | Datos de Entrada UART   |
| 4     | D08                    | Salida    | Salida Digital 8  |
| 5     | RESET                  | Entrada   | Módulo de RESET (el pulso debe ser de al menos 200 ns)                          |
| 6     | PWM0/RSSI              | Salida    | Salida PWM0 / Indicador RX de Intensidad de la Señal                            |
| 7     | PWM1                   | Salida    | Salida PWM1   |
| 8     | [Reservado]            | -         | No conectar   |
| 9     | DTR / SLEEP_RQ / DI8   | Entrada   | Pin de Control Sleep o Entrada Digital 8  |
| 10    | GND                    | -         | Tierra  |
| 11    | AD4/DIO4               | Ambos     | Entrada Analógica 4 o Entrada / Salida Digital 4                                |
| 12    | CTS/DIO7               | Ambos     | Clear-To-Send Flow Control o Entrada / Salida Digital 7                         |
| 13    | ON/SLEEP               | Salida    | Indicador del Estado del Módulo   |
| 14    | VREF                   | Entrada   | Tensión de Referencia   |
| 15    | Associate / AD5 / DIO5 | Ambos     | Indicador de Asociación / Entrada Analógica 5 o Entrada / Salida Digital 5      |
| 16    | RTS / AD6 / DIO6       | Ambos     | Request-to-Send Flow Control / Entrada Analógica 6 o Entrada / Salida Digital 6 |
| 17    | AD3 / DIO3             | Ambos     | Entrada Analógica 3 o Entrada / Salida Digital 3                                |
| 18    | AD2 / DIO2             | Ambos     | Entrada Analógica 2 o Entrada / Salida Digital 2                                |
| 19    | AD1 / DIO1             | Ambos     | Entrada Analógica 1 o Entrada / Salida Digital 1                                |
| 20    | AD0 / DIO0             | Ambos     | Entrada Analógica 0 o Entrada / Salida Digital 0                                |



En referencia a la antena que incorporan los módulos, hay 3 tipos disponibles:



**Ilustración 4.3-3** Tipos de antenas de los módulos XBee.

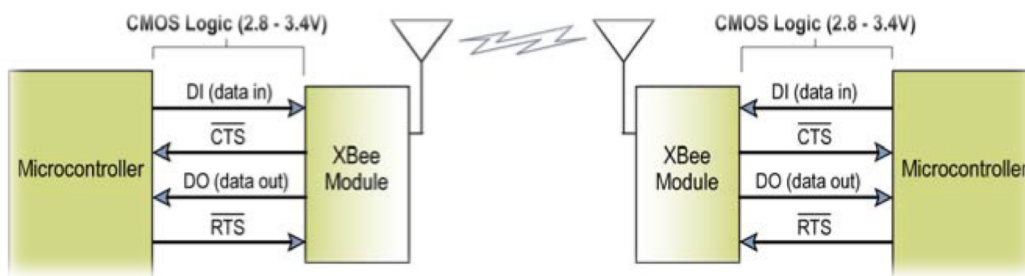
La antena tipo Whip es una pequeña antena externa que viene incorporada en el módulo y es la opción más común. El conector RF U.F.L. es un simple conector para añadir la antena que se desee. La antena tipo Chip es un integrado especialmente diseñado para actuar como antena de forma sencilla.

La gran mayoría de las patas del módulo se pueden configurar como entradas/salidas digitales de propósito general o como entradas analógicas, ideales para leer sensores que se quieran conectar al módulo. En una de estas entradas analógicas es donde se puede leer el indicador de potencia de señal recibida.

Los modos de operación con los que cuenta un módulo XBee pueden ser:

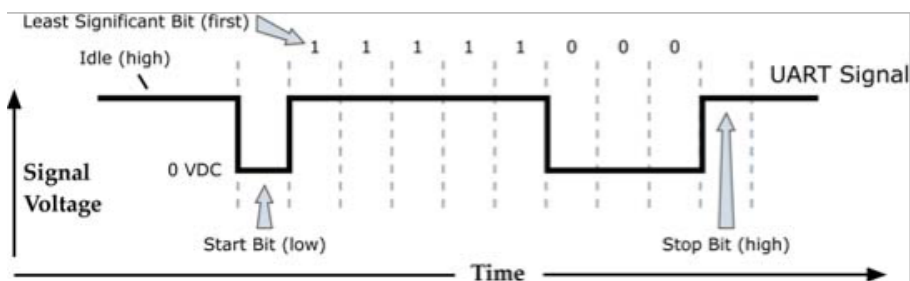
- *Idle Mode*: es un estado en el que se encuentra el módulo cuando no se recibe ni se transmite información.
- *Transmit Mode*: estado cuando se recibe información serie en el *buffer* DI.
- *Receive Mode*: cuando se recibe datos RF válidos a través de la antena.
- *Sleep Mode*: estado de bajo consumo de energía.
- *Command Mode*: se ejecuta una serie de caracteres interpretados como un comando.

Para la comunicación con el exterior, el microcontrolador interno del módulo utiliza una comunicación a través del puerto serie mediante un módulo UART (Universal Asynchronous Receiver-Transmitter). Cualquier dispositivo que disponga también de otro módulo UART se puede comunicar con el módulo, como puede ser un microcontrolador externo.



**Ilustración 4.3-4** Comunicación UART de los módulos XBee.

Los datos entran al módulo UART a través del pin 3 (DIN) como una señal serie asíncrona. La señal debe estar en estado alto cuando no se están transmitiendo datos. Cada byte de datos se compone de un bit de inicio (bajo), 8 bits de datos (bit menos significativo primero) y un bit de parada (alto).



**Ilustración 4.3-5** Paquete de datos transmitidos por el módulo UART de XBee.

Por defecto, la velocidad de comunicación de los módulos está configurada en 9600 baudios.

Conclusión: se ha escogido un módulo XBee PRO OEM RF modelo XBP24-AWI-001. La elección de este módulo en lugar del módulo XBee se basa, viendo las características de ambos, en la potencia de transmisión. Aún sabiendo que el modelo elegido es económicamente más costoso y consume más cuando está transmitiendo, es prioritario disponer de mayor rango de alcance para abarcar el área máximo posible.

#### 4.3.2. Microcontrolador.

Cada uno de los circuitos que forman los nodos necesitan de un microcontrolador para comunicarse con el transceptor inalámbrico y poder procesar la información que se envía y se recibe.

Por definición, un microcontrolador es un circuito integrado que contiene toda la estructura de un ordenador, es decir, CPU, RAM, ROM, además de diversos periféricos integrados como convertidores analógico/digital, temporizadores, contadores, sistemas que permiten conexiones serie y paralelo y otros más según la especialización del microcontrolador, permitiendo aplicaciones muy diversas.

En el mercado hay una gran multitud de fabricantes de microcontroladores, como MicroChip, Atmel, Analog Devices, Toshiba, Texas Instruments y Fairchild entre otros, cada uno con su propia arquitectura y lenguaje de programación.

Para el proyecto se necesitan unas características del microcontrolador específicas que cumplan con los siguientes requisitos: tensión de alimentación de entre 3,3 y 5 V, puerto serie UART, memoria RAM suficiente, buena velocidad de operación y compatibilidad para ser programado en lenguaje C.

Conclusión: De entre todas las posibilidades barajadas, la mejor opción ha sido un microcontrolador PIC de Microchip Technology Inc., familia de microcontroladores con los cuales se ha tenido la posibilidad de trabajar con anterioridad. Los microcontroladores PIC son de los más usados desde hace unos años por su versatilidad, sus características y su fácil uso. El modelo seleccionado es el de 8 bits PIC18LF2550, el cual cumple con los requisitos anteriores y dispone de las siguientes características:

Tensión de alimentación: entre 2 y 5,5 V.

Memoria RAM: 2 Kbits.

Módulo USART mejorado.

Optimización para compiladores de lenguaje C.

Soporte USB.

Admisión de frecuencias de oscilador externo de hasta 48 MHz.

4 temporizadores.

Corriente máxima de salida de los puertos: 200 mA.

Potencia total de disipación: 1 W.

El principal motivo por el que se ha seleccionado este microcontrolador, además del previo conocimiento que se tiene de los microcontroladores PIC, es la característica que posee de poder trabajar a una tensión de 3,3 V, la misma tensión a la que trabajan los módulos XBee. Esto permite la opción de suministrar la misma tensión para ambos dispositivos sin necesidad de hacer una adaptación de niveles de tensión para cada uno. Sin embargo, para evitar conflictos en el completo funcionamiento del PIC, en este proyecto sí que se alimenta el PIC a 5 V pero se podría alimentar a 3,3 V realizando un simple puente.

Finalmente, el tipo de encapsulado por el que se ha optado es SOIC de 28 pines, un encapsulado SMD con el objetivo de diseñar una placa más pequeña y situar el microcontrolador justo debajo del módulo transceptor.

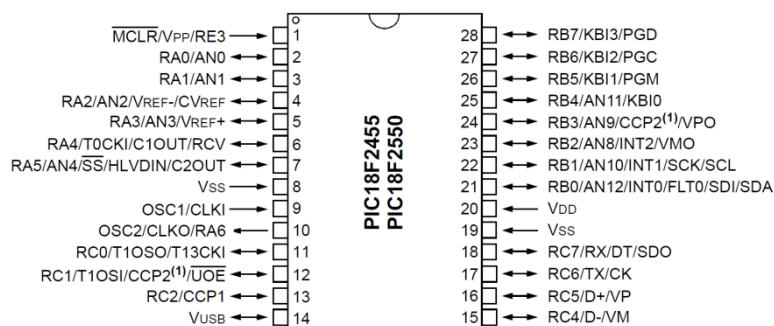


Ilustración 4.3-6 Encapsulado SOIC del PIC18F2550.

### 4.3.3. Interfaz ZigBee-PC.

La interfaz ZigBee-PC es concretamente el nodo coordinador. Como se ha explicado con anterioridad, el nodo coordinador debe existir para poder configurar la red ZigBee y para desempeñar el papel de interfaz entre la red y el ordenador, ya que este nodo debe estar constantemente recibiendo información de los nodos baliza y transmitiéndola a un ordenador.

En primera instancia, se valoró la opción de diseñar esta placa pero, finalmente, se optó por la solución de adquirir un módulo ya implementado. Existen diversas tarjetas de desarrollo enfocadas a la misma aplicación que se necesita para este proyecto. Una de estas tarjetas está diseñada especialmente para la comunicación entre ordenadores y módulos XBee y es la opción elegida en este proyecto: el XBee Explorer USB.

El XBee Explorer USB es una placa de desarrollo que consta, básicamente, de un chip FTDI que trabaja como convertidor miniUSB a puerto serie para poder comunicarse con el ordenador, convirtiendo las señales RS232 a niveles de tensión de 3,3 V y viceversa. Consta también de dos tiras de 20 pines donde se puede conectar cualquier modelo XBee de la serie 1 y 2, un regulador de tensión para alimentar el módulo acoplado y LEDs indicando el estado de transmisión, recepción, alimentación y RSSI.

Los únicos requisitos para usar esta placa de desarrollo son: disponer de un cable miniUSB a USB para la conexión entre la placa y el ordenador e instalar los drivers FTDI necesarios. Como ocurre con el transceptor inalámbrico del robot SR1, esta placa se comporta como un puerto serie virtual. No se necesita alimentación externa ya que se alimenta de la señal proveniente del mismo puerto USB.

Esta placa viene acompañada de un software de configuración llamado X-CTU el cual se explica en el apartado de diseño de software.



**Ilustración 4.3-7** Placa de desarrollo Bbee Explorer USB.

#### **4.3.4. Baterías.**

La batería es uno de los elementos clave y comúnmente uno de los más olvidados en cualquier aplicación electrónica.

El objetivo de las baterías en este proyecto es proporcionar una larga duración del funcionamiento del sistema de localización. Esto es debido a que, si se quiere implementar un sistema como este en un recinto público, además de poder ofrecer el servicio de forma continua, los nodos baliza deben estar posicionados en lugares de poca visibilidad y difícil acceso, comúnmente en el techo de la instalación.

Al disponer de módulos transceptores que consumen más que los modelos básicos de la gama XBee, la mejor opción es utilizar baterías que tengan la máxima capacidad posible y un tamaño considerablemente pequeño teniendo en cuenta las dimensiones de las placas a las que se añaden. Por todas estas razones, la solución es utilizar baterías de iones de litio, también llamadas baterías Li-ion.

Conclusión: Se ha elegido un conjunto de 10 baterías Li-ion recargables de la marca UltraFire del modelo 18650, con una capacidad de 5000 mAh y un voltaje de 3,7 V cada una. La gran capacidad que ofrecen garantizan el requisito de obtener una larga duración de funcionamiento del sistema.

Las tensiones necesarias para alimentar cada placa son 5 V y 3,3 V , correspondientes al microcontrolador y al módulo XBee, por lo que se escogen grupos de 2 baterías para alimentar cada placa. Colocándolas en serie se obtiene una tensión de 7,4 V, suficiente para los dos voltajes indicados.

Para poder aplicar estas baterías sin el uso de cableado se ha decidido utilizar un porta baterías para PCB de dos pilas en cada una de las placas, de modo que solamente es necesario realizar la soldadura y da la opción de posicionarlas en paralelo o en serie. Por otra parte, ha sido necesario adquirir un cargador de baterías de este modelo con el que se pueden cargar dos baterías al mismo tiempo.



**Ilustración 4.3-8** Baterías, porta baterías y cargador de verías del sistema.

#### 4.3.5. Robot SR1 Explorer.

El robot SR1 es una estructura robótica muy similar a las descritas en la sección 2.1. diseñada y distribuida por la empresa INPLUS S.L. y cedida por la escuela E.E.B.E. para la realización de este proyecto.

La idea por la cual fue diseñado se centra en el aprendizaje, dirigido a usuarios novatos y expertos como herramienta de enseñanza para aprender a construir robots móviles de experimentación, su funcionamiento y su programación.

El SR1 se distribuye en diferentes kits, siendo cada kit un conjunto de ampliaciones y accesorios aplicables de forma sencilla al robot. En el caso de este proyecto, el robot dispone del kit completo donde, además de los componentes presentes en el resto de kits, se incluye un cabezal móvil con una cámara inalámbrica e iluminación. El montaje del robot y sus componentes no ha sido realizado en este proyecto debido a que ya se proporcionó completamente montado.

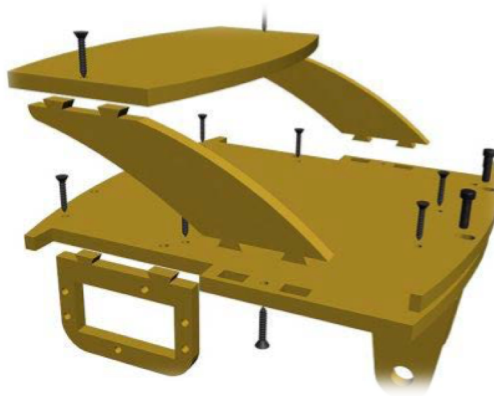


**Ilustración 4.3-9** Robot SR1 con el kit completo.

#### 4.3.5.1. Estructura principal

La estructura principal del robot SR1 se divide en cuatro partes fundamentales: el chasis, el circuito de control, el sistema motriz y la alimentación.

El chasis está formado por planchas de PVC y su ensamblaje se lleva a cabo mediante lengüetas y se asegura con tornillos, proporcionando gran solidez y el fácil desmontaje de las piezas. En la parte frontal se halla un paragolpes robusto diseñado de forma que el robot no quede atrapado en obstáculos al hacer giros o dar marcha atrás. El techo, donde se ubica la expansión móvil de la cámara inalámbrica, deja al descubierto la mayor parte del circuito electrónico para poder realizar los cambios necesarios y se dispone de una superficie ideal para transportar objetos o colocar otras placas electrónicas.



**Ilustración 4.3-10** Chasis del robot SR1.

El circuito de control es un solo circuito impreso donde se ubican el microcontrolador, los diferentes sensores, dispositivos de salida y conectores para modificar el modo de funcionamiento del robot. El circuito está realizado con tecnología de montaje convencional para facilitar el montaje por el usuario. El microcontrolador, el BasicX24P de la compañía Netmedia Inc., es el encargado de ejecutar el programa y realizar todas las tareas que controlan el robot y es muy fácil de programar en lenguaje de programación BasicX mediante un ordenador y un cable serie RS232 estándar.



**Ilustración 4.3-11** Circuito de control del Robot SR1.

El sistema motriz está formado por dos ruedas motrices independientes y una tercera rueda omnidireccional sin tracción. Las ruedas independientes actúan gracias a un servomotor añadido a cada una de ellas que trabaja con una alimentación de entre 4,8 y 7,2 V. Para el chasis de la cámara se utilizan dos servomotores más, uno para realizar el movimiento horizontal y otro para el movimiento vertical.

La alimentación se compone de 6 baterías AA que proporcionan alimentación a los servomotores y a todos los componentes electrónicos. En el SR1 se dispone de diversos tipos de alimentación, pudiendo escoger entre una alimentación independiente para los servomotores del resto de componentes o la misma alimentación para todos los componentes utilizando reguladores de tensión, entre otros.

#### 4.3.5.2. Sensores

El robot SR1 contiene un buen número de sensores que hace aumentar las funcionalidades que puede poseer.

La lista de sensores es la siguiente:

2 sensores de contacto del tipo final de carrera, ubicados en la parte frontal del chasis, uno a cada lado, que se activan cuando el parachoques toca un obstáculo.

1 sensor de inclinación que se activa cuando el robot sobrepasa un nivel de inclinación de unos 15 grados en referencia a la posición horizontal.

2 sensores de luz, del formato fotorresistencias LDR, los cuales toman un valor de resistencia en función del nivel de luz que reciben.

1 sensor de infrarrojos modulado, colocado en la zona frontal del robot, que sólo actúa cuando la señal que recibe es una señal de infrarrojos, ignorando el resto de señales luminosas.

1 sensor de ultrasonidos, formado por un emisor y un receptor de ultrasonidos y que dispone de su propio procesador. Se puede la misma aplicación con este sensor que con el sensor de infrarrojos para detectar la distancia a un obstáculo, pero también se puede usar a modo de sonar. Su posición puede variar entre la zona frontal del robot o junto a la cámara inalámbrica.

1 sensor de temperatura digital, muy útil para conocer la temperatura ambiente en cada momento.

1 brújula digital, mide la orientación a la que se encuentra el robot en todo momento. Se puede utilizar para saber el rumbo o para indicar al robot que se posicione en una orientación determinada con respecto al norte.



1 sensor seguidor de líneas, situado en la zona inferior trasera del chasis junto con 3 diodos LED a modo de indicador. Este sensor solamente hace que el robot siga el trazado de una línea negra cuando el sensor la detecta.

#### **4.3.5.3. Dispositivos de salida**

El robot SR1 también dispone de un gran número de dispositivos de salida que actúan según la información recibida por los sensores.

Los dispositivos de salida son:

1 zumbador piezoeléctrico, utilizado para generar tonos de señal de diferentes frecuencias a modo de avisos, alarmas, indicación de estados, etc.

2 emisores de infrarrojos, uno en cada extremo frontal del robot, utilizados para detectar la proximidad de obstáculos. La señal de infrarrojos que emiten es la que recibe rebotada el sensor de infrarrojos cuando un objeto está cerca.

2 diodos LED de alta potencia, útiles para realizar señales ópticas cuando un sensor actúa o para iluminar el robot en zonas oscuras. Estos diodos pueden ir en la zona frontal o montados al lado de la cámara inalámbrica.

LEDs rojo y verde a modo de señalización de estados o modos de funcionamiento del robot.

1 controlador de motores DC para conectar motores de corriente continua en sustitución de los servomotores.

1 puerto serie RS232, cuya función principal es la de poder descargar los programas desde un ordenador al microcontrolador y, también, poder enviar y recibir datos a través para controlar el robot o recibir información de los sensores.

1 transceptor inalámbrico, con 50 metros de rango en interiores. Su función es la misma que la del puerto serie pero de forma inalámbrica comunicándose con otro transceptor conectado a un ordenador.

1 conector de expansión, que permite conectar otros dispositivos y periféricos al robot, tales como la cámara inalámbrica.

1 cámara de audio y vídeo inalámbrica, con movimiento gracias al chasis del techo móvil. Su función es la de recoger imágenes y enviarlas por radiofrecuencia a un receptor que esté conectado a un ordenador o monitor.

#### 4.3.5.4. Servomotores de rotación continua.

Los motores eléctricos del SR1 son servomotores que tiene un rango de trabajo de  $180^\circ$  que han sido modificados para trabajar en rotación continua, giran libremente  $360^\circ$ . Desde el punto de vista eléctrico un servo lo único que necesita es recibir un tren de impulsos que mantenga el sistema en equilibrio. La longitud del pulso varía entre 1 y 2 ms siendo 1,5 ms la posición de equilibrio, si se mandan pulsos de 1,5 ms las ruedas permanecerán estáticas. Mientras que cuando se aumenta o se disminuye la longitud del pulso, la rueda gira en un sentido o en otro.

#### 4.3.5.5. Brújula magnética.

El sensor de brújula digital es un sensor de campos magnéticos formado por el módulo CMP03 de Devantech. La función principal de la brújula es la de determinar el rumbo, en grados, en el que el robot se encuentra, para así poder trazar rumbos o incluso para hacer un sistema de posicionamiento y determinar en donde se encuentra el robot.



Ilustración 4.3-12 Sensor brújula digital.

Una de las formas más tradicionales para conocer la orientación de un objeto es conocer el ángulo respecto al norte magnético. Su uso como sistema de orientación es anterior al siglo IX después de Cristo, y desde entonces hasta la llegada del GPS, ha sido uno de los principales pilares de los sistemas de navegación.

Hay que tener en cuenta que debido a que el norte magnético no coincide con el norte geográfico, existe una desviación entre ambos que depende de la posición del objeto en el globo terráqueo. Esta desviación se conoce como declinación magnética.

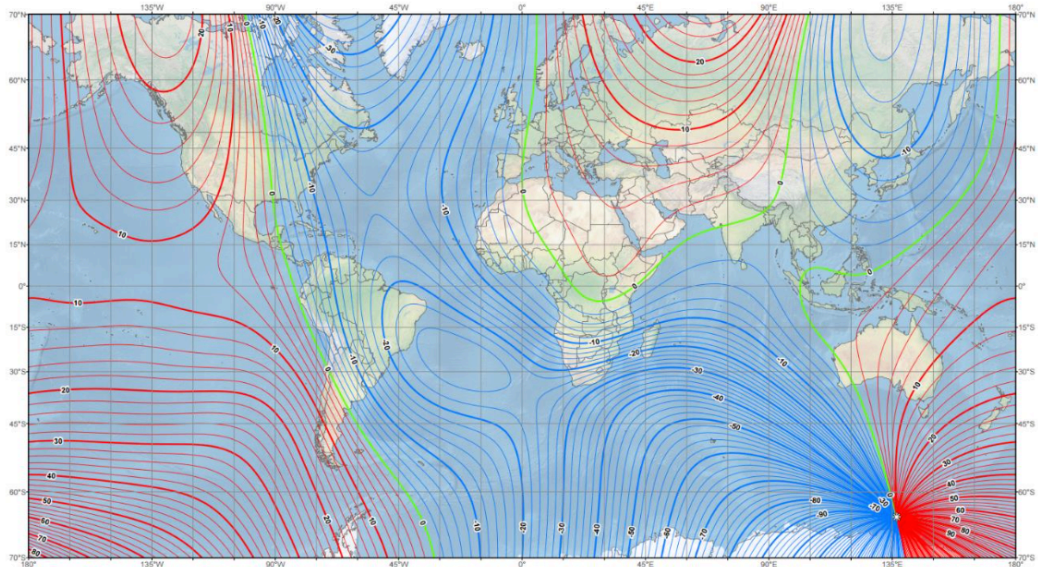


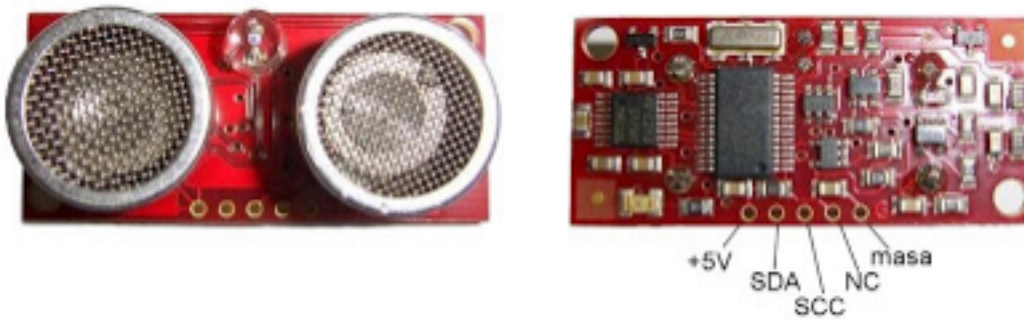
Ilustración 4.3-13 Declinación magnética 2015.

La gran ventaja de este tipo de sistemas es que el norte magnético apenas depende de la posición del objeto por lo que si se conoce la zona sobre la que está el objeto y la deriva respecto al norte magnético en dicha zona, el sistema es capaz de deducir de forma sencilla la orientación del Robot.

#### 4.3.5.6. Medidor de distancia por ultrasonidos.

El sensor de ultrasonidos empleado en el robot SR1 es el modelo SRF08 de Devantch, el sensor es un completo módulo electrónico que incluye su propio procesador para hacer el calculo de la distancia, junto con el emisor y receptor de ultrasonido. Su funcionamiento esta basado en el sónar y consiste en mandar una serie de impulsos ultrasónicos por el aire y luego medir el tiempo que tarda en recibir el eco, cuando el sonido rebota en algún objeto.

Otra característica de este sensor es que utiliza un sistema de comunicación con el mundo exterior denominado Bus I2C. El bus I2C es un sistema de comunicación creado por Phillips que utiliza sólo dos hilos para hacer una comunicación bidireccional de alta velocidad entre los diferentes elementos de un circuito. Uno de los hilos es el cable de datos, también denominado SDA, y el otro hilo es el cable de reloj SCL que contiene la señal de sincronización de los datos. Esto permite utilizar velocidades variables, que pueden ser bastante altas ,con total seguridad que los datos llegan correctamente. Dado que sólo existe un solo hilo para enviar información, el flujo de la misma sólo se puede hacer en un sentido cada vez. Para solucionar este problema, se ha implantado un protocolo de equipo maestro-esclavo, en que siempre ese maestro (normalmente el procesador) el que inicia la comunicación y los esclavos sólo responden, cuando son interrogados por el maestro.



**Ilustración 4.3-14** Sensor ultrasonidos Robot SR1.

- Rango máximo de distancia hasta 6 m.
- El SRF04 necesita dos conexiones por cada módulo a controlar. El SRF08 se controla desde un bus I2C estándar, por lo que se pueden gobernar varios módulos empleando 2 únicas conexiones.
- El consumo se reduce a 3 mA en standby y 15mA en funcionamiento.
- Es capaz de medir diferentes ecos recibidos por la señal ultrasónica que puede rebotar contra uno o varios objetos a diferentes distancias.
- Tanto la ganancia de los amplificadores internos como el rango de mediciones es ajustable por el usuario.
- Dispone de una célula LDR que permite realizar medidas de luz ambiente.
- Ofrece una lectura directa que se puede representar en centímetros, pulgadas o micro segundos.
- Un diodo led en la parte posterior del módulo genera un código de intermitencias que expresa la dirección I2C actual del módulo así como el inicio de una nueva medida.

La comunicación con el módulo SRF08 se realiza según el protocolo I2C, disponible en la mayor parte de los microcontroladores actuales aunque también puede ser implementado por software. La comunicación se realiza de la misma manera que con cualquier otro dispositivos I2C. La dirección del módulo es, por defecto, la 0xE0, aunque existe la posibilidad de que el usuario cambie esta dirección por cualquiera de las 16 siguientes: 0xE0, 0xE2, 0xE4, 0xE6, 0xE8, 0xEA, 0xEC, 0xEE, 0xF0, 0xF2, 0xF4, 0xF6, 0xF8, 0xFA, 0xFC o 0xFE. Esto permite controlar hasta 16 módulos SRF08 con un mismo bus (2 líneas).

Además de estas 16 direcciones, todos los módulos responden a la dirección 0x00 de llamada general. Esto permite escribir un determinado comando en esa dirección que inicie una nueva medida en todos los módulos SRF08 disponibles en el bus, al mismo tiempo. Posteriormente, la medida de cada módulo se lee de forma individual, indicando su dirección particular.

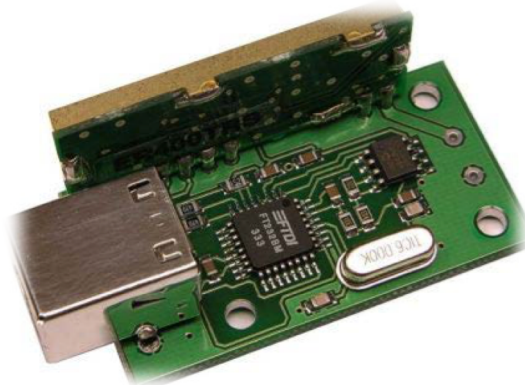
#### 4.3.6. Transceptor inalámbrico externo

Uno de los elementos que no se incluyen en el kit del robot SR1 es el transceptor de radiofrecuencia que se conecta a un ordenador para enviar y recibir datos a través de radiofrecuencia, con un alcance de 150 metros en exteriores y unos 50 metros en interiores.

Esta placa dispone de un convertidor USB a puerto serie para poder conectarlo a cualquier ordenador. El transceptor de radiofrecuencia es el mismo que lleva incorporado el SR1 en el circuito de control, el modelo ER400TRS de Easy Radio, lo que garantiza una perfecta compatibilidad entre ambos elementos. este transceptor opera a una frecuencia de 433 MHz y 9600 baudios por defecto.

Con esta placa es posible enviar y recibir comandos de tal forma que se puedan adquirir datos desde el robot hacia el ordenador y viceversa de manera inalámbrica, sin necesidad de tener conectado el cable RS232 y consiguiendo mayor comodidad. Para su uso sólo se necesita instalarle una antena que puede estar formada por un trozo de cable de 17 cm e instalar los controladores adecuados para su funcionamiento. Con estos requisitos, la placa se comporta como un puerto serie virtual. La alimentación proviene del puerto USB del ordenador, por lo que no requiere de ningún tipo de batería.

Como esta placa no estaba incluida en el kit ha sido necesario adquirirla de la empresa distribuidora.



**Ilustración 4.3-15** Transceptor inalámbrico externo del robot SR1.

#### 4.3.7. Herramientas hardware del sistema.

Tabla 2 Herramientas hardware del sistema..

| Nombre                                    | Tipo     | Descripción   |
|---|----------|---|
| Baterías 18650                            | Hardware | 10 baterías tipo 18650 5000 mAh 3,7 V                                   |
| Cable USB Tipo AB                         | Hardware | Cable de comunicación del transceptor del robot SR1 - PC / Eclipse - PC |
| Cable USB A - DB9                         | Hardware | Cable de comunicación PC - robot SR1                                    |
| Cable USB - miniUSB                       | Hardware | Cable de comunicación nodo coordinador - PC                             |
| Cargador de baterías                      | Hardware | Cargador de baterías 18650  |
| Programador USB de PICs Eclipse           | Hardware | Placa programadora de PICs  |
| Robot SR1                                 | Hardware | Robot SR1 acoplado al nodo remoto                                       |
| Transceptor inalámbrico externo robot SR1 | Hardware | Placa para la comunicación RF del robot SR1                             |
| Balizas Xbee                              | Hardware | 5 Módulos Xbee  |
| XBee Explorer USB                         | Hardware | Placa del nodo coordinador y configurador de los módulos XBee           |

## 4.4. Software del sistema.

En este apartado se describen los elementos de software que intervienen en el control y guiado del robot móvil.

### 4.4.1. Programación XBee.

Los módulos XBee se pueden programar para trabajar en dos tipos de operación diferentes: operación transparente y operación API.

#### 4.4.1.1. Operación transparente

Por defecto, los módulos vienen configurados con este modo de operación. Operando de esta forma, todos los datos recibidos en el módulo UART por el pin DIN son puestos en cola en el buffer DI para ser enviados por RF y quedan almacenados en el buffer en caso de que no puedan ser enviados porque se están recibiendo datos por RF. Cuando se reciben datos por RF estos se envían por el pin DOUT, sin añadir ningún tipo de información adicional.

En este modo transparente la conexión es punto a punto, de forma que sirve para reemplazar cualquier conexión serie por cable entre dos puntos.

Los módulos se configuran mediante comandos AT con los que se pueden definir todos los parámetros de los módulos como su dirección. Al ser una configuración entre dos puntos y las direcciones ya están definidas previamente a la comunicación, no es necesario ningún tipo de control sobre los datos.

Un ejemplo de configuración mediante comandos AT se muestra en la siguiente tabla, donde los comandos AT se pueden enviar en formato ASCII o en hexadecimal, <Enter> significa salto de línea y <CR> retorno de carro.

**Tabla 3** Ejemplo de configuración de los módulos XBee mediante comandos AT.

| Comando AT enviado | Respuesta          | Descripción del comando                   |
|--------------------|--------------------|---|
| +++                | OK<CR>             | Entra en modo comando                     |
| ATDL<Enter>        | {Valor actual}<CR> | Lee la dirección de destino actual        |
| ATDL1A0D<Enter>    | OK<CR>             | Modifica la dirección de destino a 0x1A0D |
| ATWR<Enter>        | OK<CR>             | Guarda la configuración                   |
| ATCN<Enter>        | OK<CR>             | Sal del modo comando                      |

#### 4.4.1.2. Operación API

La operación API (Application Programming Interface) es una alternativa al modo de operación transparente. En este modo todos los datos que entran y salen del módulo son empaquetados en frames o tramas que definen operaciones o eventos dentro del módulo. Estas tramas disponen de una cabecera que aseguran la transmisión de los datos, tal y como hace el protocolo TCP.

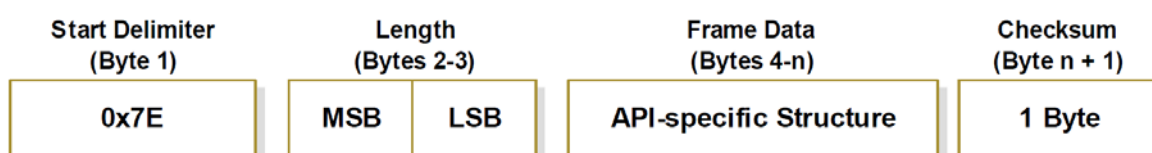
La operación API presenta las siguiente ventajas frente al modo transparente:

Conexión punto a multipunto sin entrar en modo Comando.

Estado de recepción exitosa/fallida de cada paquete RF transmitido.

Identificación de la dirección fuente de cada paquete.

Información sobre el nivel de señal recibido (RSSI) y otros parámetros. La estructura del modo API se define de la siguiente manera:



MSB = Most Significant Byte, LSB = Least Significant Byte

**Ilustración 4.4-1** Estructura del modo API de los módulos XBee.

Los campos que forman la estructura API son:

**Delimitador de inicio:** es una secuencia de 8 bits que indica el inicio de la trama y se define con el valor 0x7E. Cualquier información recibida anteriormente a este delimitador es descartada.

**Longitud:** formado por dos octetos que indican el número de bytes comprendidos en la trama de datos, entre la longitud y el checksum.

**Datos de la trama:** tipo de mensaje especificado por cada identificador API.

**Suma de verificación (Checksum):** byte que sirve para verificar la integridad de los datos. Para calcularlo se suman todos los bytes comprendidos en el campo de datos de la trama, obteniendo un valor en el cual sólo se consideran los 8 bits menos significativos. Este resultado se resta al valor 0xFF y se obtiene el valor de checksum. Para verificar que el checksum es correcto, se suma su valor y el de todos los bytes que forman el campo de datos de la trama. Si el resultado es 0xFF el checksum es correcto. En la siguiente tabla se muestran los distintos identificadores API en su valor hexadecimal y el tipo de mensaje al que corresponde:

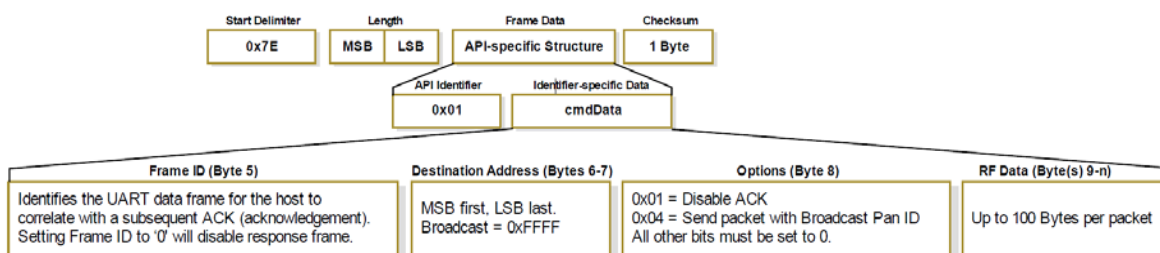


**Tabla 4** Identificadores API y su tipo de mensaje de los módulos XBee.

| Identificador API | Tipo de mensaje                                    |
|-------------------|--|
| 0x00              | Petición de transmisión (dirección de 64 bits)     |
| 0x01              | Petición de transmisión (dirección de 16 bits)     |
| 0x08              | Comando AT   |
| 0x09              | Registro de valor en cola de un comando AT         |
| 0x17              | Comando AT remoto                                  |
| 0x80              | Paquete recibido (dirección de 64 bits)            |
| 0x81              | Paquete recibido (dirección de 16 bits)            |
| 0x82              | Paquete recibido IO (dirección de 64 bits)         |
| 0x83              | Paquete recibido IO (dirección de 16 bits)         |
| 0x88              | Respuesta de un comando AT                         |
| 0x89              | Estado de la transmisión                           |
| 0x8A 0x97         | Estado del modem Respuesta de un comando AT remoto |

De los dos modos de operación, por sus características, las direcciones de los emisores y destinatarios y el indicador de potencia de señal, la operación API es la utilizada en la programación de los módulos XBee de este proyecto. Dentro de los diversos tipos de mensajes que se han citado se utilizan tres: la petición de transmisión (dirección 16 bits), el paquete recibido (dirección 16 bits) y el comando AT.

Petición de transmisión (TX Request) para direcciones de 16 bits: La petición de transmisión hace que el módulo envíe los datos recibidos por el módulo UART. Su estructura es la siguiente:

**Ilustración 4.4-2** Estructura de una trama API de petición de transmisión de los módulos XBee.

Los campos que forman los datos de la trama son:

Identificador API: formado por un único byte que indica el tipo de mensaje. Para una petición de transmisión en direcciones de 16 bits, el valor hexadecimal correspondiente es 0x01.

Identificador de trama: este campo permite relacionar la trama con su correspondiente respuesta. Si el valor es 0x00 se deshabilita el estado de la trama enviada.

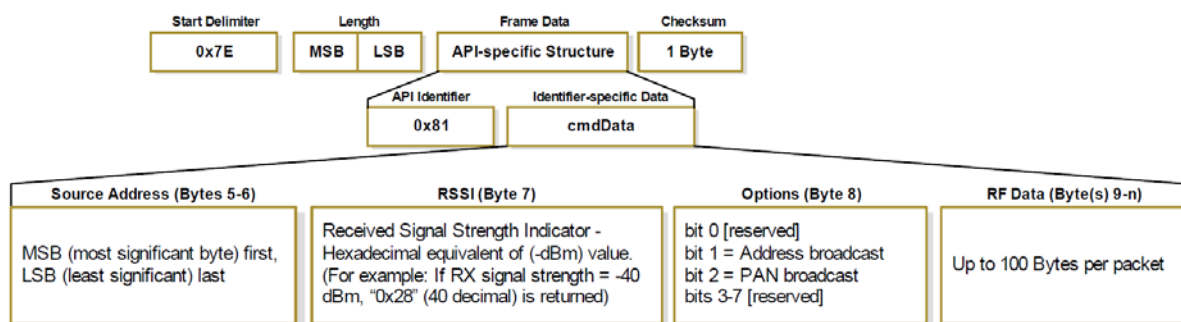
Dirección de destino: campo formado por dos bytes que indica la dirección del módulo al cual está dirigida la trama. Si el valor es 0x00 se configura para una transmisión en modo broadcast, enviando la información a todos los dispositivos disponibles en la red.

Opciones: se puede seleccionar entre deshabilitar la respuesta generada a la trama o enviar el paquete a todas las PAN.

Datos: contiene la información deseada a enviar con un máximo de 100 bytes por paquete.

Paquete recibido (RX Receive) para direcciones de 16 bits: Cuando se recibe un paquete por el módulo RF, éste lo envía por el módulo UART.

con la siguiente estructura:



**Ilustración 4.4-3** Estructura de una trama API de paquete recibido de los módulos XBee.

Los campos que forman los datos de la trama son:

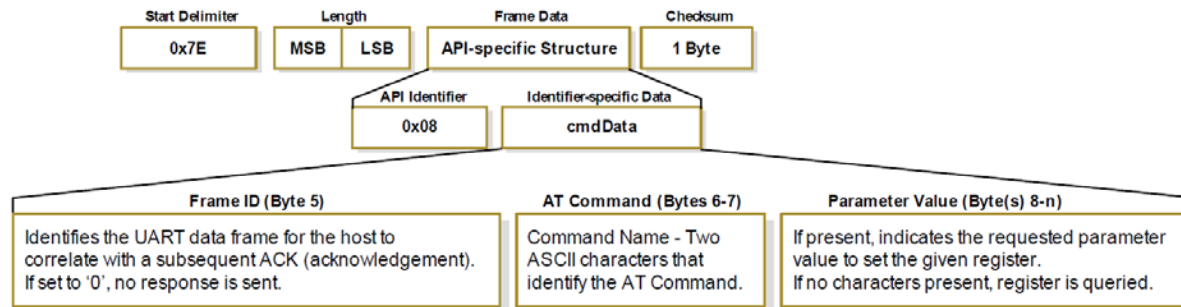
Identificador API: formado por un único byte que indica el tipo de mensaje. Para un paquete recibido en direcciones de 16 bits, el valor hexadecimal correspondiente es 0x81.

Dirección de origen: estos dos bytes indican la dirección del módulo del que proviene el paquete recibido.

RSSI: este campo muestra la potencia de la señal recibida con la que se recibió el paquete. Su valor corresponde al valor de la potencia (-dBm) en hexadecimal. Por ejemplo, si el valor de la potencia es -40 dBm en decimal, el valor hexadecimal correspondiente es 0x28.

Opciones: permite saber al módulo receptor cómo fue enviada la trama. El bit 1 indica si el paquete fue enviado mediante dirección de difusión y el bit 2 indica si se envió mediante difusión PAN.

Datos: contiene la información deseada enviada en la transmisión con un máximo de 100 bytes por paquete. Comando AT: La trama API Comando AT permite establecer los parámetros de configuración del módulo XBee al que se envía del mismo modo que se realiza en la operación transparente. Con esta trama los parámetros se aplican de forma inmediata. Su estructura es la siguiente:



**Ilustración 4.4-4** Estructura del modo API de comando AT de los módulos XBee.

Los campos que forman los datos de la trama son:

Identificador API: formado por un único byte que indica el tipo de mensaje. Para un comando AT el valor hexadecimal correspondiente es 0x08.

Identificador de trama: este campo permite relacionar la trama con su correspondiente respuesta. Si el valor es 0x00 se deshabilita el estado de la trama enviada.

Comando AT: dos bytes en formato ASCII o hexadecimal que identifican el comando AT a aplicar en el registro.

Valor del parámetro: bytes que indican el parámetro que se ha de establecer en el nada hola programado Juan una pequeña hola ir olí, registro dado previamente.

#### 4.4.2. Software de configuración XBee.

##### 4.4.2.1. X-CTU.

Los módulos XBee vienen provistos de una herramienta software multiplataforma llamada X-CTU. Esta aplicación ha sido diseñada por DIGI International para interactuar con los módulos XBee mediante una simple interfaz gráfica, pudiendo configurarlos y realizar pruebas con ellos. X-CTU 6.2.0 incluye las siguientes características:

Gestión y configuración de múltiples dispositivos RF, incluso remotamente por RF.

Actualización del firmware de cada módulo de radio.

Dos consolas para comandos AT y API para comunicarse con los dispositivos.

Opción de guardar sesiones de consola para poder cargarlas en ordenadores diferentes.

Generador de tramas: genera cualquier tipo de trama API para obtener su valor .

Intérprete de tramas: decodifica cualquier trama API para obtener sus valores.

Modo de recuperación: restaura los valores de fábrica de un módulo dañado o en modo de programación.

Prueba de rango: realiza una prueba de alcance entre dos módulos conectados a la misma red.

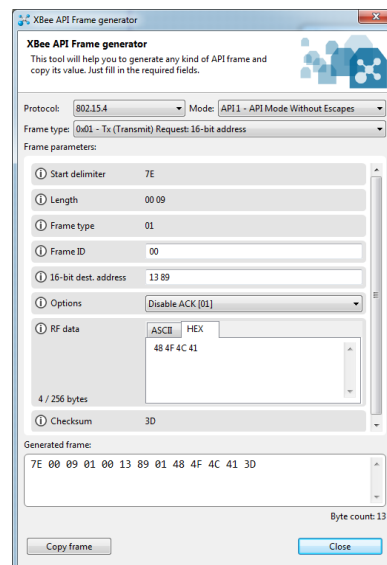


Ilustración 4.4-5 Generador de tramas API del software X-CTU.

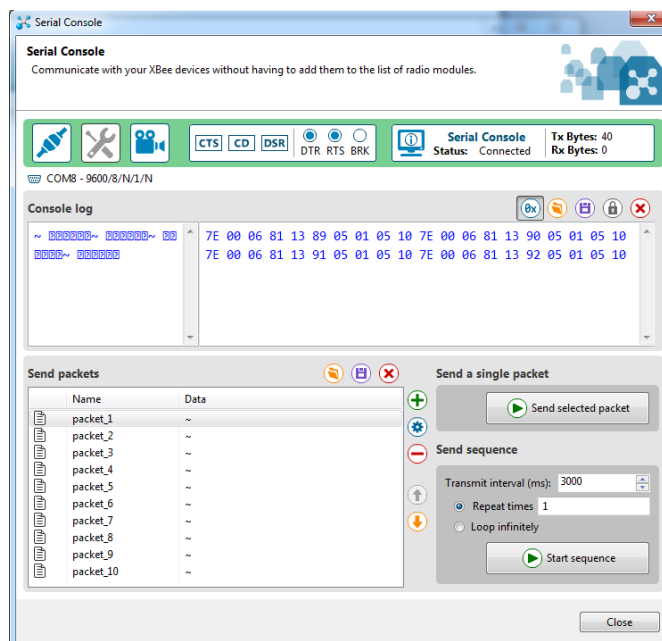


Ilustración 4.4-6 Consola de comunicación serie del software X-CTU.

El apartado básico de X-CTU es el de configuración de los dispositivos. En la ventana principal del programa se muestra la opción de añadir el dispositivo conectado al ordenador, en este caso el nodo coordinador. Si está conectado y se realiza una búsqueda en el puerto serie que utiliza el nodo coordinador, automáticamente lo detectará y se podrá añadir a la lista de dispositivos encontrados.

Con el módulo del nodo coordinador añadido, se pueden detectar el resto de módulos que operan en la misma red de forma remota pulsando sobre el botón "Descubrir nodos de radio en la misma red", sin necesidad de conectar cada módulo al XBee Explorer USB. Por defecto, todos los módulos están configurados para operar en la misma identificación de red y el mismo canal, por lo que no debe haber problemas a la hora de detectarlos. Una vez detectados, se añaden a una lista de dispositivos remotos asociados al nodo coordinador.

Cuando la lista de dispositivos detectados está completa, se puede seleccionar cada uno de los módulos para ver todos los parámetros de configuración de los que disponen y su actual valor. Estos valores se pueden modificar y aplicar al instante.

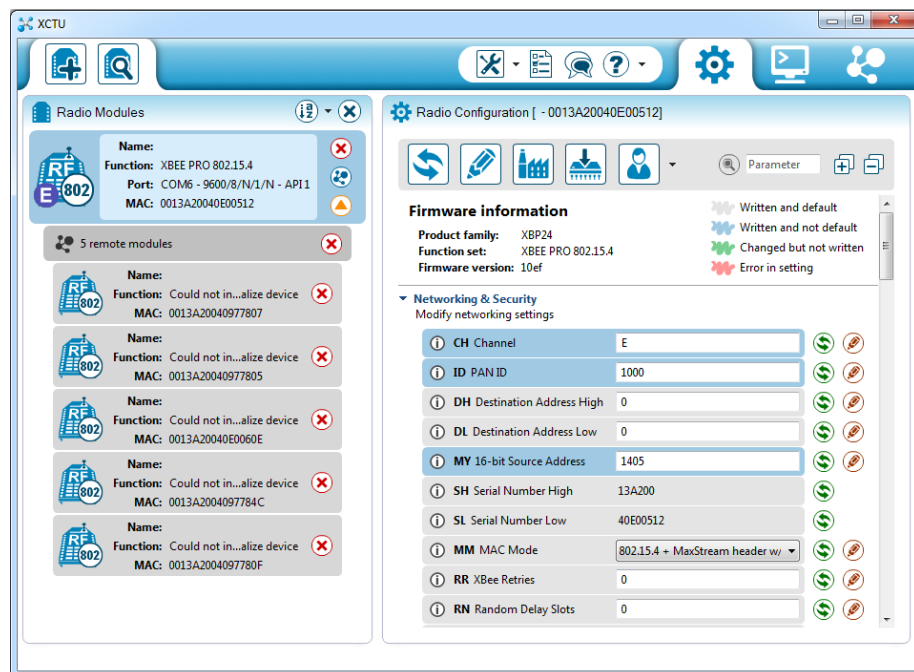


Ilustración 4.4-7 Ventana de configuración de los módulos XBee del software X-CTU.

#### 4.4.2.2. Configuración de los módulos XBee.

Como bien se ha explicado en apartados anteriores, los dispositivos que forman el sistema de localización son los que se muestran en la siguiente tabla:

Tabla 5 Dispositivos de la red del sistema de localización y su función.

| Dispositivo | Alimentación  | Posición | Función  |
|-------------|---------------|----------|--|
| Remoto      | Baterías      | Móvil    | Notificar posición   |
| Baliza 1    | Baterías      | Fija     | Recoger datos del nodo móvil y enviarlos al coordinador                |
| Baliza 2    | Baterías      | Fija     | Recoger datos del nodo móvil y enviarlos al coordinador                |
| Baliza 3    | Baterías      | Fija     | Recoger datos del nodo móvil y enviarlos al coordinador                |
| Baliza 4    | Baterías      | Fija     | Recoger datos del nodo móvil y enviarlos al coordinador                |
| Coordinador | USB Ordenador | Fija     | Almacenar los datos de los nodos baliza y transmitirlos a un ordenador |

Cada uno de los módulos XBee que llevan incorporados los dispositivos del sistema deben ser configurados de una forma especial para que se puedan comunicar entre ellos.

Los parámetros básicos y necesarios para configurar una red XBee son los que se enumeran a continuación:

ID: la identificación de la red PAN que se utiliza. Cada red PAN dispone de su propia ID y su valor puede ser elegido aleatoriamente entre 0 y 0xFFFF. Todos los módulos deben trabajar en la misma red y, por tanto, se les ha de asignar el mismo valor de ID.

CH: número de canal usado para transmitir y recibir datos entre módulos. Cada módulo ha de tener configurado el mismo canal. El valor puede estar comprendido entre 0x0C y 0x17 para los módulos XBEE-PRO.

MY: es la dirección de origen de 16 bits del módulo. Su función es la de identificar el dispositivo desde el cual se recibe información. Los valores aceptados van desde 0 hasta 0xFFFF y se pueden escoger de forma aleatoria. Cada módulo debe disponer de su propia dirección de origen.

DH: dirección de destino (alta) que, combinado con el parámetro DL, se utiliza para determinar la dirección a quien va destinado el mensaje a transmitir. Para direccionamientos de 16 bits, este parámetro debe configurarse con el valor 0.

DL: dirección de destino (baja). En direccionamientos de 16 bits, este parámetro define la dirección del módulo destinatario que debe coincidir con el parámetro MY de ese módulo. Se debe configurar en cada mensaje para definir a quién va dirigido.

CE: habilita el módulo como coordinador o dispositivo final. Como la red se configura punto a punto, todos los módulos deben configurarse con el valor 0 para que actúen como dispositivos finales.

A1: asociación de los dispositivos finales con un coordinador. Todos los módulos deben tener este parámetro deshabilitado con un valor de 0.

PL: potencia de emisión de los módulos. Según la normativa europea, la máxima potencia permitida es de 20 dBm (100 mW), que corresponde a un valor PL de 4.

AP: habilita el modo de operación API. Todos los módulos deben tener este parámetro habilitado.

En la tabla siguiente se muestra como quedan definidos los parámetros de todos los módulos:

**Tabla 6** Parámetros de configuración de cada nodo del sistema de localización.

| Parámetro | Módulo remoto | Baliza 1      | Baliza 2      | Baliza 3      | Baliza 4      | Módulo coordinador |
|-----------|---------------|---------------|---------------|---------------|---------------|--------------------|
| ID        | 0x1000        | 0x1000        | 0x1000        | 0x1000        | 0x1000        | 0x1000             |
| CH        | 0x0E          | 0x0E          | 0x0E          | 0x0E          | 0x0E          | 0x0E               |
| MY        | 0x1400        | 0x1401        | 0x1402        | 0x1403        | 0x1404        | 0x1405             |
| DH        | 0             | 0             | 0             | 0             | 0             | 0                  |
| DL        | Según destino | Según destino | Según destino | Según destino | Según destino | Según destino      |
| CE        | 0             | 0             | 0             | 0             | 0             | 0                  |
| A1        | 0             | 0             | 0             | 0             | 0             | 0                  |
| PL        | 4             | 4             | 4             | 4             | 4             | 4                  |
| AP        | Habilitado    | Habilitado    | Habilitado    | Habilitado    | Habilitado    | Habilitado         |

Con estos parámetros definidos en cada módulo se obtiene la red configurada para una correcta comunicación.

Cabe destacar que el nodo coordinador se puede configurar como coordinador de la red pero eso implica que el resto de nodos no puedan comunicarse entre ellos y sólo puedan hacerlo con el coordinador, estableciendo una configuración punto a multipunto. Con este método no es posible determinar la potencia de la señal que hay entre el nodo remoto y las balizas, por lo que la mejor opción es configurar todos los módulos como dispositivos finales.

#### 4.4.3. Programación de los microcontroladores.

La programación de los microcontroladores de los que dispone cada nodo se ha realizado con lenguaje C para PICs. El entorno de programación utilizado es el MPLAB IDE versión 8.89 y el compilador es el HI-TECH ANSI C Compiler 18F versión 9.80, un compilador de lenguaje C para PICs de la familia 18F. Ambos son de la empresa Microchip Technology Inc.

Las razones por las que se escogió este entorno de programación en concreto son: es uno de los entornos de programación más utilizados en la industria, su interfaz gráfica es intuitiva, la gran variedad de herramientas de las que dispone, compatibilidad con un gran número de compiladores y la facilidad con la que se pueden configurar los proyectos y los microcontroladores asociados a él.

El compilador HI-TECH es uno de los más conocidos de entre la gran variedad que existe en el mercado, junto con el compilador CCS. Es un compilador rápido y eficaz. La principal razón para su elección es el previo conocimiento de haber trabajado con él y su lenguaje.

Por otra parte, al ser de la misma empresa que los mismos microcontroladores, este software garantiza una plena compatibilidad entre ambos.

El programa de los microcontroladores se ha realizado en un mismo archivo para los diferentes nodos, ya que el único cambio existente entre la función de los nodos repercute en el tratamiento de los datos recibidos por el módulo XBee.

#### **4.4.3.1. Nodo móvil**

El programa del microcontrolador del nodo móvil se encarga de recoger los datos recibidos a través del módulo XBee provenientes de los nodos baliza. Estos datos se tratan conforme a la dirección de origen de la trama API recibida, correspondiente a cada una de las balizas.

Primero se configuran los parámetros iniciales del microcontrolador (configuración de puertos) y los parámetros del módulo UART (modo asíncrono, velocidad = 9600 baudios, sin paridad, transmisión de 8 bits).

La función del programa es esperar a que se reciban datos por el módulo UART conectado al módulo transceptor. En este estado de espera, el LED verde conectado al pin DIO5 se mantiene encendido, indicando que el nodo se mantiene en funcionamiento a la espera de una comunicación entrante.

Una vez se detecta la entrada de datos en el módulo UART, estos se almacenan en una variable con la que poder trabajar. El primer paso es comprobar la dirección de origen de la que provienen los datos, la cual debe coincidir con una de las direcciones definidas en cada uno de los nodos baliza. Para ello se comprueban los bytes 5 y 6 del paquete recibido API que contienen esta información. Si la dirección de origen coincide con una de las balizas, se recoge el valor de RSSI con la que se ha recibido el paquete que también está incluido en la estructura de la trama API recibida, concretamente se incluye en el byte número 7.

El segundo paso es construir la trama API para enviar el valor de RSSI recogido en el paso anterior. Este valor se asigna al byte 9 de la trama a enviar que corresponde al mensaje que se quiere transmitir. A



continuación, es necesario realizar la verificación checksum y se asigna el resultado al byte número 10 de la trama.

Por último, cuando la trama ya está correctamente construida, se envía a la misma dirección de la baliza con la cual se recibió el primer mensaje, asignando este valor a los bytes 6 y 7.

Un ejemplo de cálculo de checksum se muestra a continuación:

Se quiere enviar desde el nodo remoto al nodo baliza con dirección 0x1401 el valor de RSSI sin ningún tipo de mensaje. Por tanto, la trama API es [0x7E, 0x00, 0x06, 0x01, 0x00, 0x14, 0x01, 0x01, 0x00, checksum].

La suma de bytes:  $= 0\ 01 + 0\ 00 + 0\ 14 + 0\ 01 + 0\ 01 + 0\ 00 = 0\ 17$  (6)

Como este resultado no supera el valor 0xFF, los 8 bits menos significativos son el mismo resultado. Al restar este resultado a 0xFF se obtiene el valor de checksum:

$h = 0 - 0\ 17 = 0\ 8$  (7) • Aplicando este proceso en lenguaje C:

```
uint EnviaBall[] = {0x7E, 0x00, 0x06, 0x01, 0x00, 0x14, 0x01, 0x01, 0x00, 0x00}; EnviaBall[9] = 0xFF - (EnviaBall[3] + EnviaBall[4] + EnviaBall[5] + EnviaBall[6] + EnviaBall[7] + EnviaBall[8]);
```

donde:

EnviaBall = nombre de la trama API.

EnviaBall[9] = byte número 9 (posición 10) de la trama EnviaBall, correspondiente al checksum.

Cuando se reciben datos, el LED verde se apaga y se enciende el LED amarillo conectado al pin DIO1, indicando el tratamiento de los datos recibidos y su posterior envío, apagándose cuando ya se han enviado. Para encender y apagar los LEDs asociados se recurre a enviar al módulo el comando AT en formato API para su activación o desactivación.

#### 4.4.3.2. Nodos baliza

Aunque la función del programa de los nodos baliza es muy similar a la del nodo móvil, la diferencia radica en las direcciones de origen y destino de los mensajes y el tratamiento de los datos.

De igual forma que en el nodo móvil, el programa configura los parámetros iniciales del microcontrolador y el módulo UART y se mantiene a la espera de la recepción de datos válidos por radiofrecuencia, siendo el sistema de funcionamiento de los LEDs idéntico.

Una vez se detecta la entrada de datos en el módulo UART, estos se almacenan en una variable con la que poder trabajar. Se comprueba la dirección de origen de la trama API, la cual ha de coincidir con la dirección del nodo coordinador o la del nodo móvil.

Si la dirección de origen coincide con la del nodo coordinador significa que es una orden de petición de la ubicación del nodo móvil por parte del coordinador. En este caso, no es necesario realizar ningún tipo de lectura de los otros parámetros que forman la trama API recibida. Lo siguiente es crear la trama API para enviar al nodo móvil la petición de ubicación, asignando a los bytes 6 y 7 el valor de su dirección, calcular y asignar el checksum de la trama y enviarla mediante el módulo UART.

Si la dirección de origen proviene del nodo móvil, se realiza un paso más. El mensaje de la trama del nodo móvil lleva incluido el valor de RSSI de la petición enviada por la baliza en el byte 9 y, además, se debe leer el valor de RSSI del mensaje actual que se encuentra en el byte número 7. Se realiza la media de estos dos valores en hexadecimal para obtener más precisión sobre las lecturas hechas y se crea el nuevo mensaje para enviarlo al nodo coordinador. El procedimiento de la nueva trama sigue siendo idéntico: asignar como dirección de destino la del nodo coordinador en los bytes 6 y 7, asignar el resultado de la media de valores RSSI al byte 9 de la trama y calcular el checksum en el byte 10 para verificar que los datos son correctos. De esta manera, el nodo coordinador obtiene el valor de potencia entre la baliza y el nodo remoto y lo almacena.

En el código de los nodos baliza, el procedimiento de calcular la media de valores RSSI se realiza de la siguiente manera:

```

for (i=0; i<=9; i++) {
    DataRec[i] = RecibeByteUSART();
}
if(DataRec[4] == 0x14 && DataRec[5] == 0x00) {
    EnviaCor[8] = (DataRec[6] + DataRec[8]) / 2;
    EnviaCor[9] = 0xFF - (EnviaCor[3] + EnviaCor[4] +
EnviaCor[5] + EnviaCor[6] +
    EnviaCor[7] + EnviaCor[8]);
for (i=0; i<=9; i++) {
        EnviaByteUSART (EnviaCor[i]);
    }
}

```

"DataRec" es el nombre de la trama API recibida por el módulo UART, en este caso del nodo remoto, y contiene los dos valores de potencia de la señal correspondientes a los bytes de las posiciones 6 y 8 de la trama. "EnviaCor" es la trama API que se envía al nodo coordinador con el resultado final del valor

RSSI. El byte de la posición 8 de "EnviaCor" corresponde al mensaje del paquete y es donde va destinado el valor de la media calculado en la línea anterior. Por último, se calcula el checksum y se envía el paquete.

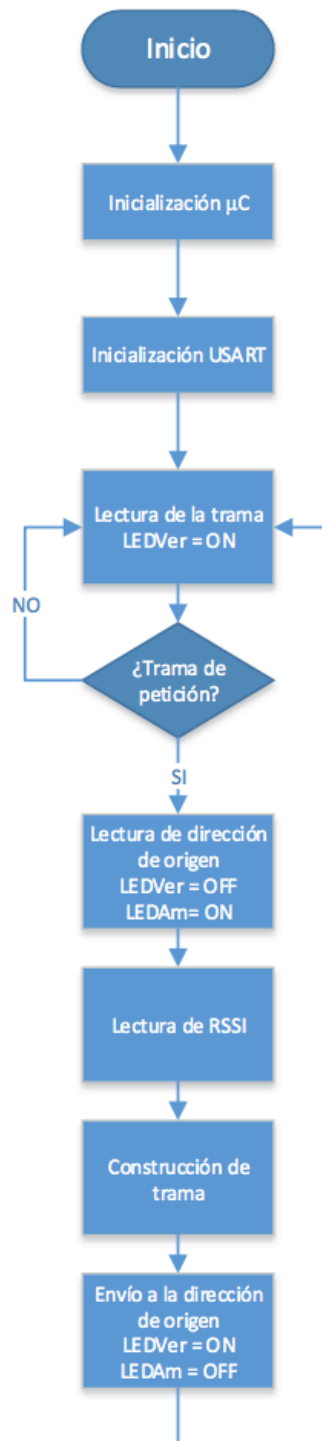


Ilustración 4.4-8 Diagrama de flujo del programa del nodo móvil.

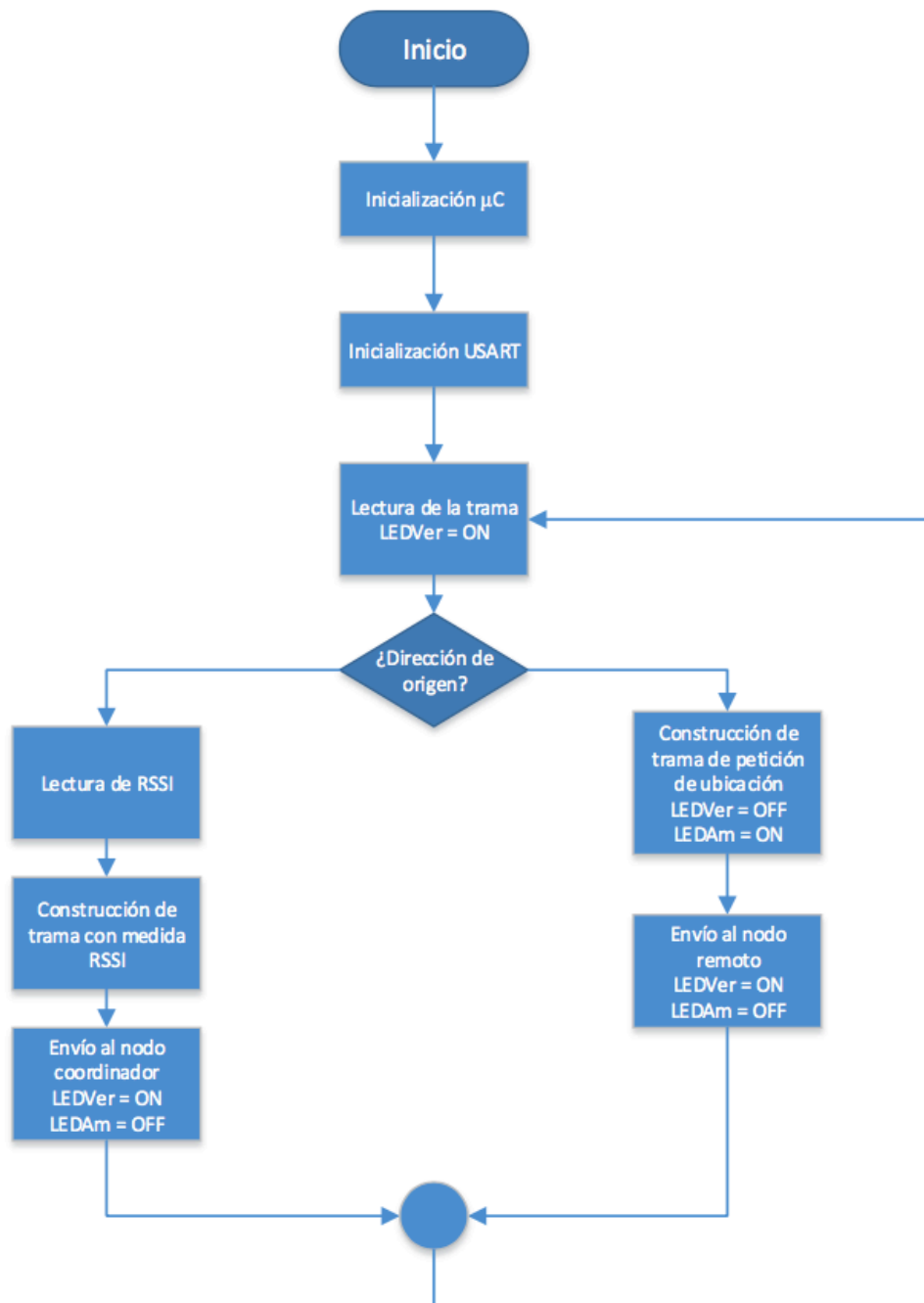


Ilustración 4.4-9 Diagrama de flujo del programa de los nodos baliza.

#### 4.4.4. Programación SR1.

El microcontrolador del SR1 incorpora un software del fabricante es cual se ha modificado para el mejor funcionamiento del sistema de control y guiado del robot móvil. El software completo que incorpora el robot SR1 puede encontrarse en el Anexo I de esta memoria.

El programa que se ejecuta en el robot, dispone del programa principal. Lo primero que hace es preparar los búfers de recepción y transmisión del puerto serie y el puerto Com1, para que el robot pueda recibir los comandos que el usuario le está mandando a través del ordenador.

Se asegura de que tanto los servos de la cámara como los del robot estén en su posición central y parada, que la cámara esté desactivada y los leds estén apagados. Cuando todo ya está prácticamente listo hace un beep para decir que ya está preparado.

Una vez ya está todo preparado, se espera a recibir un número de caracteres por el búfer de recepción del puerto serie.

Cuando los recibe, comprueba el número de caracteres recibidos por el búfer de recepción del puerto serie. Acto seguido hace una comparación y comprueba que si el número de bytes recibidos en el puerto serie es mayor a cuatro, entonces hay un comando. Si se da el caso, lee ocho caracteres del puerto serie y asigna a la variable comando los cinco caracteres leídos. Seguidamente asigna a la subrutina SubComando los tres primeros caracteres recibidos en el puerto serie para identificar la familia del comando.

Para el funcionamiento óptimo del sistema se ha modificado que lea 8 bytes por el puerto serie, que los tres último los asigne a una variable y convierta los string en numero integer. Para la modificación del programa se emplea la herramienta software BasicX V.2.10

```
'Asigna a Tx los 3 último caracteres leidos
Tx = Chr(Orden(6)) & Chr(Orden(7)) & Chr(Orden(8))

'Convierte los strings en numero Integer
Call ValueS(Tx, Valrs, Success)
Valr=CInt(Valrs)
```

Se han implementado las siguientes subrutinas que se instancian enviado por el puerto serie los siguientes comandos.

- MORRF Mueve el robot hacia delante la distancia indicada.
- MORRR Mueve el robot hacia atrás la distancia indicada.
- MORRD Gira hacia la izquierda los grados indicados (servomotores).
- MORRI Gira hacia la derecha los grados indicados (servomotores).
- MORRU Gira al valor de grados indicado (con brújula).

```

IF Comando ="MORRF" Then          'Se mueve hacia delante la distancia indicada
Call Adelantepar (Valr,0)
End If
IF Comando ="MORRR" Then          'Se mueve hacia atrás la distancia indicada
Call Atraspar (Valr,0)
End If
If Comando= "MORRD" Then          'Gira hacia la derecha los grados indicados (servomotores)
Call GDerecha (Valr,0)
End If
If Comando= "MORRI" Then          'Gira hacia la izquierda los grados indicados
(servomotores)
Call Gizquierda (Valr,0)
End If
if Comando= "MORRU" Then          'Gira al valor de grados indicado (con brújula)
Call GRum (Valr,0)
End If

'Subrutina que hace que el robot avance la distancia indicada
Sub Adelantepar(ByVal Pasos As Integer,ByVal Salto As Integer)
Dim N As Integer
Se1Value = Se1Stop                'Parte de la posición de parados
Se2Value = Se2Stop
For N = 1 To Pasos                'Este número controla la cantidad
Se1Value = Se1Value + ServoAcel    'Si no hay obstáculos rueda recto
Se2Value = Se2Value - ServoAce2    'Rueda derecha e izquierda adelante
Call Pulsos                        'Envía los impulsos a los servos
Next
End Sub

'Subrutina que hace que el robot retroceda la distancia indicada
Sub Atraspar(ByVal Pasos As Integer,ByVal Salto As Integer)
Dim N As Integer
Se1Value = Se1Stop                'Parte de la posición de parados
Se2Value = Se2Stop
For N = 1 To Pasos                'Este número controla la cantidad
Se1Value = Se1Value - ServoAce2    'Si no hay obstáculos rueda recto
Se2Value = Se2Value + ServoAcel    'Rueda derecha e izquierda adelante
Call Pulsos                        'Envía los impulsos a los servos
Next
End Sub

Sub GDerecha(ByVal GiroI As Integer, ByVal Salto As Integer)
Dim Grados As Integer
Dim Veces As Integer
Acel1 = 85 'Estos valoren varían dependiendo
Acel2 = 85 'del valor de parada de cada servo
Grados = GiroI
For Veces = 1 To Grados \ 2
Se1Value = Se1Stop - Acel1 'Frena la rueda derecha
Se2Value = Se2Stop - Acel2 'Acelera la rueda izquierda
Call Pulsos 'Manda los pulsos a los servos
Next
End Sub

```

```
'Subrutina que hace que el robot realice un giro hacia la derecha con el valor pasado en grados
aproximadamente (no brújula)
Sub GIzquierda(ByVal GiroD As Integer, ByVal Salto As Integer)
Dim Grados As Integer
Dim Veces As Integer
Acel1 = 78 'Estos valoren varían dependiendo
Acel2 = 78 'del valor de parada de cada servo
Grados = GiroD
For Veces = 1 To Grados \ 2
Se1Value = Se1Stop + Acel1 'Frena la rueda derecha
Se2Value = Se2Stop + Acel2 'Acelera la rueda izquierda
Call Pulsos 'Manda los pulsos a los servos
Next
End Sub
```

```
'Subrutina que hace que el robot gire hacia la izquierda con el valor pasado en grados.
Sub Izquierda(ByVal GiroI As Integer, ByVal Salto As Integer)
Dim Destino As Integer
Dim Ofset As Integer 'Sirve como valor diferencial
Dim NuevaPos As Integer 'Posición que se alcanza cada vez
Acel1 = 82
Acel2 = 82
Ofset = 0
Call Dir 'Comprueba el rumbo actual
If DirACT <= GiroI Then 'Si el rumbo actual es menor que el
DirACT = DirACT + 360 'giro que hay que dar, se suma una
Ofset = 360 'vuelta completa para hacer los cálculos
End If
Destino = DirACT - GiroI - 1 'Calcula el ángulo de destino
Do
Se1Value = Se1Stop - Acel1 'Frena la rueda derecha
Se2Value = Se2Stop - Acel2 'Acelera la rueda izquierda
Call Pulsos 'Manda los pulsos a los servos
Call Dir 'Lee el rumbo actual
If Salto > 0 Then 'Gira también la cámara
If NuevaPos <> DirACT Then
NuevaPos = DirACT 'Calcula desde la posición actual
Se4Value = Se4Value - Salto 'y le quita el salto
If Se4Value < Se4Med Then 'Hasta que esté en la posición
Se4Value = Se4Med 'central
End If
End If
Call PulseOut(Se4, Se4Value, 1) 'Pulso servo horizontal
End If
If DirACT <= GiroI Then 'Si todavía no ha llegado
DirACT = DirACT + Ofset 'actualiza la dirección actual
End If
If DirACT - Destino < 20 Then 'Cuando falta poco para llegar
Acel1 = 40 'gira más lentamente para no pasarse
Acel2 = 40
End If
If DirACT <= Destino Then 'Si se alcanza el destino se sale
Exit Do
End If
```

#### 4.4.5. Programa en LabVIEW para el PC de control y guiado del robot móvil.

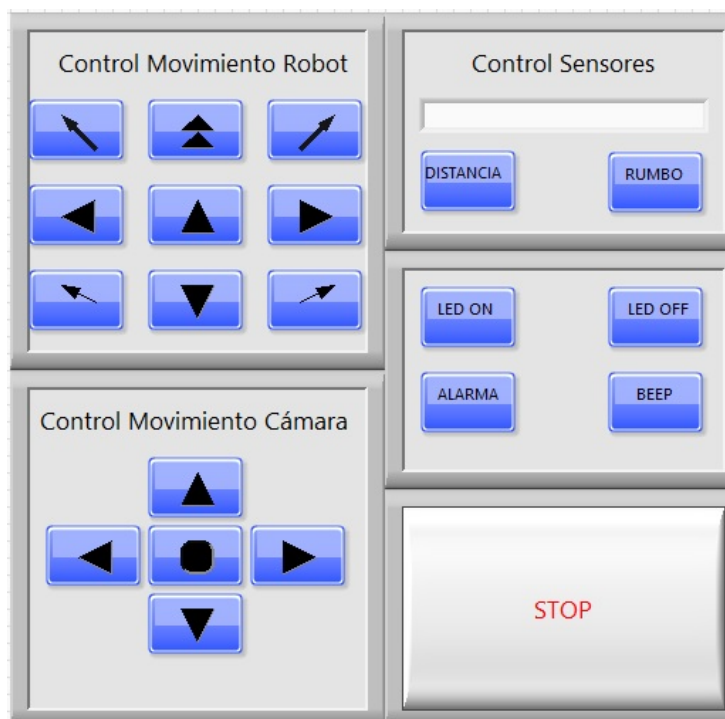
Se han realizado tres programas diferentes para el PC para el control y guiado del robot móvil, todos ellos en lenguaje de programación LabVIEW.

##### 4.4.5.1. Programa de guiado.

Esta aplicación esta diseñada para el funcionamiento libre de robot, teniendo un control remoto desde el PC, pudiéndolo mover por todo el área y utilizar sus sensores y actuadores.

Cuando se ejecuta este programa, se dispone de una aplicación gráfica con cinco zonas de controles diferenciadas.

- Control movimiento robot.
- Control movimiento cámara.
- Control sensores.
- Control actuadores.
- Salida del programa.



**Ilustración 4.4-10** Aplicación programa de guiado del Robot SR1.

En la zona superior izquierda se encuentran los controles que hacen referencia al movimiento del robot (de izquierda a derecha, de arriba a abajo).



- Gira el robot 15 grados a la izquierda.
- Mueve el robot adelante 30 cm.
- Gira el robot 15 grados a la derecha.
- Gira el robot 45 grados a la izquierda.
- Mueve el robot adelante 15 cm.
- Gira el robot 45 grados a la derecha.
- Gira el robot 5 grados a la izquierda.
- Mueve el robot adelante 15 cm.
- Gira el robot 5 grados a la derecha.

En la zona inferior izquierda se encuentran los controles que hacen referencia al movimiento de la cámara (de izquierda a derecha, de arriba a abajo).

- Inclina el módulo superior del robot hacia arriba.
- Gira el módulo superior del robot hacia la izquierda.
- Retorno del módulo superior del robot hacia a la posición central.
- Gira el módulo superior del robot hacia la derecha.
- Inclina el módulo superior del robot hacia abajo.

En la zona superior derecha se encuentran los controles de los sensores del robot que nos aportan información reflejada en el indicador.

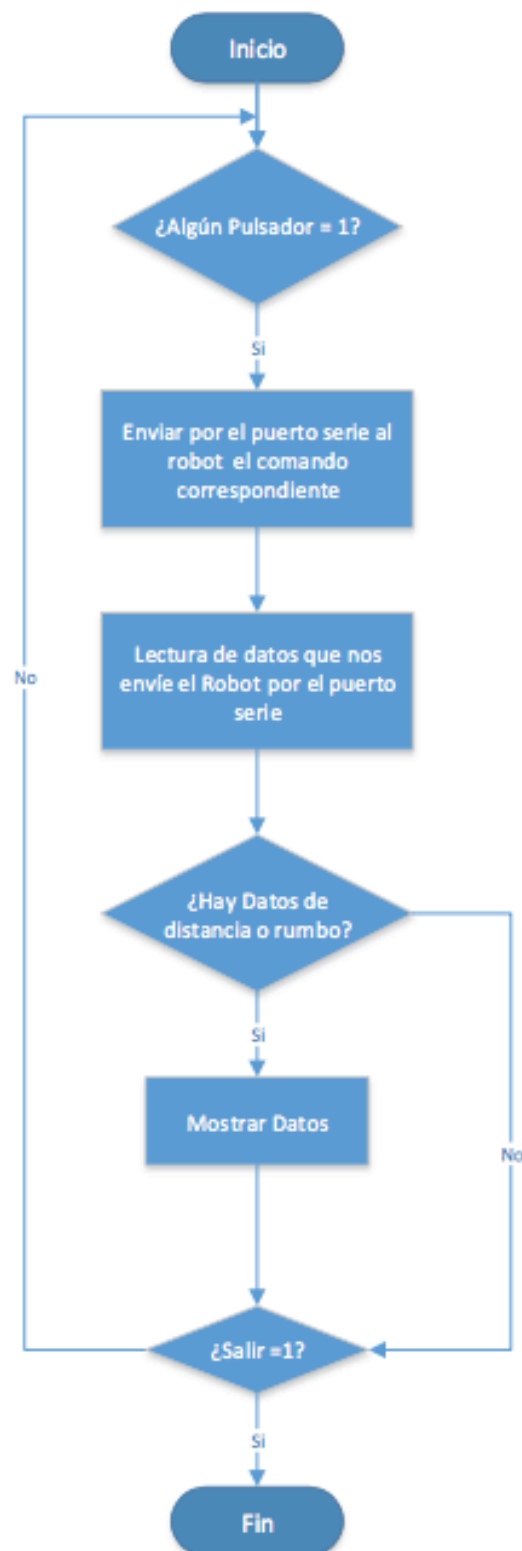
- Indica la distancia/rumbo.
- Mide la distancia frontal a objetos con el sensor de ultrasonidos SRF08.
- Mide el rumbo con la brújula digital CMP03.

En la zona central derecha se encuentran los actuadores que dispone el robot.

- Enciende los leds del robot.
- Apaga los leds del robot.
- Acciona la alarma del robot.
- Acciona el zumbador piezoeléctrico del robot.

En la zona inferior derecha se encuentra el botón de salida de la aplicación.

Con el accionamiento de cada pulsador lo que hacemos es enviar por el puerto serie un string que el robot reconoce y ejecuta la subrutina correspondiente al comando enviado. También leemos los datos que el robot nos devuelve por el puerto serie, los interpretamos y los mostramos en el indicador.



**Ilustración 4.4-11** Diagrama de flujo del programa de guiado del Robot SR1.

#### 4.4.5.2. Programa de repetición de trayectorias.

Esta aplicación está diseñada para la repetición de trayectorias del robot, es muy habitual que los robots repitan movimientos o acciones para este tipo de caso es de gran utilidad, el inconveniente es que es un sistema nada flexible y que existen infinitas trayectorias posibles, resulta poco práctico cuando el robot tenga gran cantidad de trayectorias diferentes que tenga que reproducir.

Este programa tiene dos partes:

- La memorización de trayectorias.
- La reproducción de trayectorias.

En la memorización de trayectorias se dispone de una aplicación gráfica donde encontramos cuatro partes diferenciadas.

- Inicialización del programa.
- Movimientos del robot.
- Localización del robot.
- Exportación de datos y salida del programa.



Ilustración 4.4-12 Aplicación gráfica grabación de trayectoria.

En la parte superior izquierda se encuentra la parte inicialización donde previamente a la memorización de trayectoria, se le indica al sistema las coordenada de partida del Robot. Por medio de un diálogo emergente se pide que se coloque el robot paralelo a una de las paredes de la sala. Una vez realizados estos pasos cuando se pulse iniciar se da paso a la fase de enviar comandos al Robot.

En la parte inferior izquierda se encuentran la parte de control del robot, existen cuatro cuatro pulsadores con las instrucciones que se le pueden enviar al robot y un dial con su indicador digital (de izquierda a derecha, de arriba a abajo).

- El robot se mueve hacia delante.
- El robot se mueve hacia atrás.
- El robot gira hacia la izquierda.
- El robot gira hacia la derecha.

La función del dial es seleccionar la distancia o los grados que queremos que se mueva o gire.

En la parte superior derecha se encuentra la representación gráfica de la posición del robot y de la trayectoria que esta siguiendo.

En la parte inferior derecha se encuentra un indicador de distancia frontal, el robot después de realizar cada instrucción mide la distancia que hay frontal hacia objetos, si encuentra un objeto a una distancia inferior a un metro la muestra en el indicador y la almacena para poderla comparar con la que se mida en la fase de reproducción de trayectorias.

Finalmente existe un control donde se debe escribir el nombre con el que se quiere exportar el archivo y el pulsador de grabar y salir.

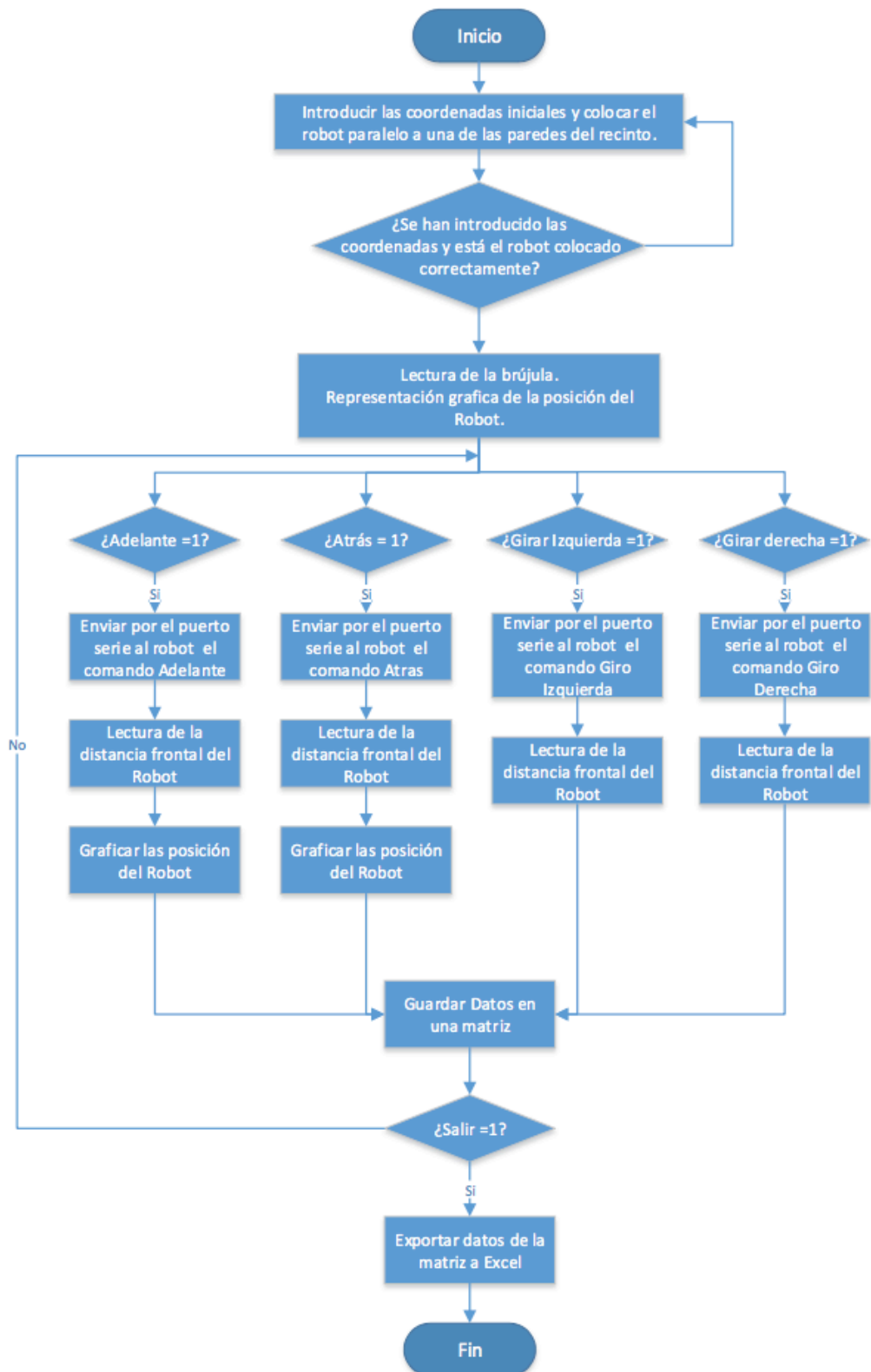
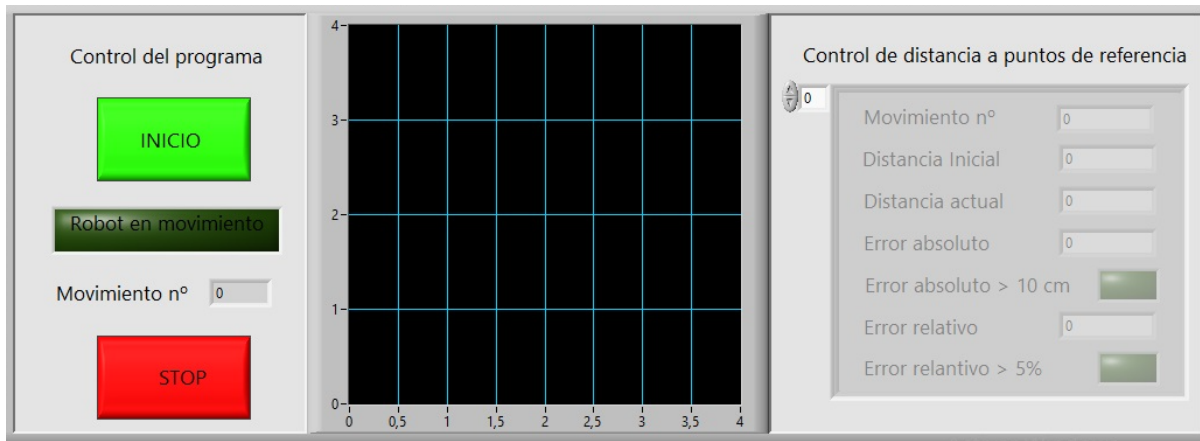


Ilustración 4.4-13 Diagrama de flujo programa grabación de trayectoria.



**Ilustración 4.4-14** Aplicación gráfica reproducción de trayectorias.

En la reproducción de trayectorias se dispone de una aplicación gráfica donde aparecen tres partes diferenciadas.

- Inicialización del programa.
- Localización del robot.
- Control de distancia a puntos de referencia.

En la parte izquierda se encuentra la parte de control del programa, al iniciarse el programa se debe seleccionar el archivo que tiene almacenadas las instrucciones a reproducir, seguidamente aparece una primera ventana emergente donde se indica que se debe colocar el robot a 0 grados en referencia a los cerramientos del recinto y las coordenadas donde se debe colocar el robot, cuando estos pasos estén completados se solicita que se pulse el botón iniciar .

Se ejecutan las instrucciones seguidas hasta que se finalice la última almacenada, existe un botón de stop con el que se puede interrumpir la secuencia, el led verde indica que el robot está en movimiento y el indicador de movimiento indica que número de instrucción se está ejecutando.

En la parte central se encuentra la representación gráfica de la posición del robot y de la trayectoria que esta siguiendo.

En la parte derecha se encuentra el control a puntos de referencia. De la misma manera que en la fase de memorización de trayectoria después de ejecutar cada instrucción se mide la distancia que hay frontal hacia objetos, si encuentra un objeto a una distancia inferior a un metro la almacena en una matriz y muestra los siguientes datos:

- En que número de instrucción se han encontrado referencia.
- La distancia medida en la fase de memorización de trayectorias.
- La distancia medida en la fase de reproducción de trayectorias.
- El error absoluto entre las dos medidas.
- El error relativo entre las dos medidas.
- Un indicador que se ilumina si el error absoluto es mayor a 10 cm.
- Un indicador que se ilumina si el error relativo es superior al 5%.

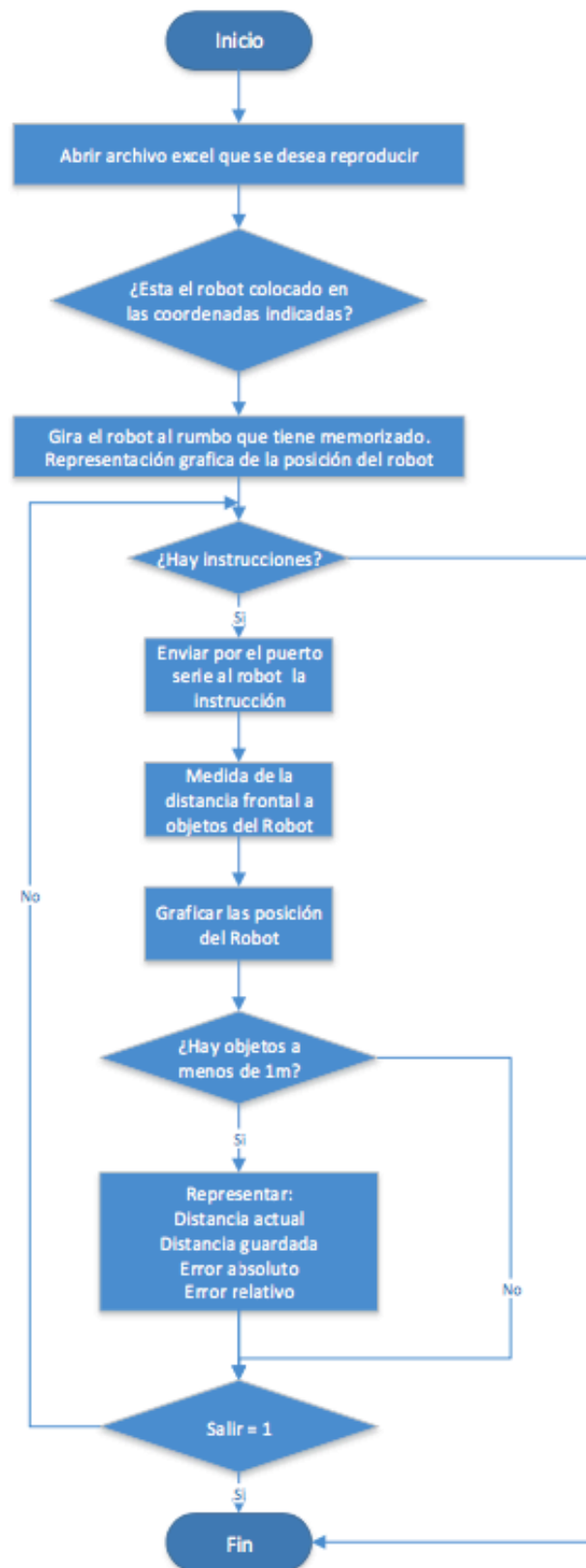


Ilustración 4.4-15 Diagrama de flujo reproducción de trayectoria.



#### 4.4.5.3. Programa para la localización odométrica y de medición de potencia de recepción.

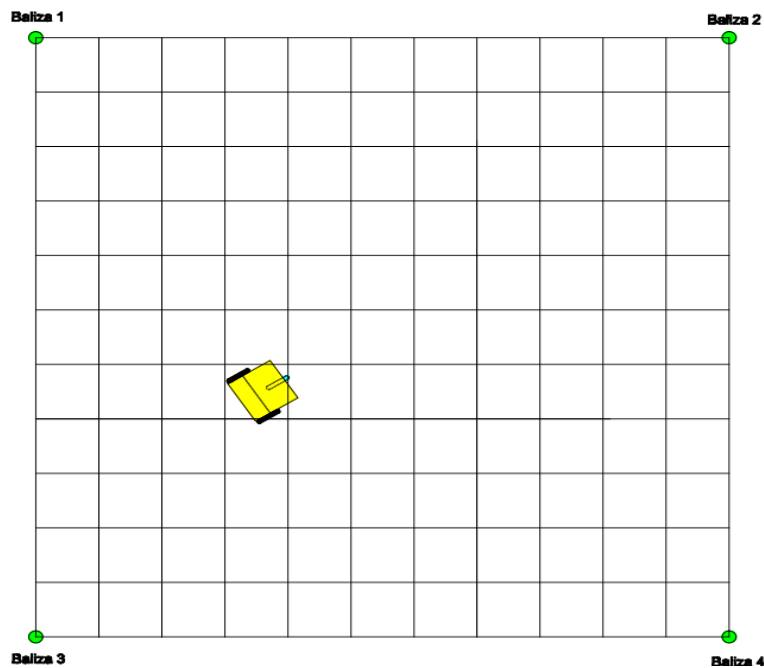
Esta aplicación esta diseñada para el guiado del robot y la localización de dos formas, una primera odométrica y una segunda por potencia de recepción de señal RSSI.

Esté programa se divide en dos partes:

- Memorización de valores RSSI.
- Aplicación de guiado y localización.

La aplicación de memorización de valores RSSI tiene como objetivo hacer un mapa de valores RSSI de cada una de los nodos balizas en la zona de trabajo del robot, para ello se deben colocarse deben los módulos baliza como se indica en la figura 4.4-17.

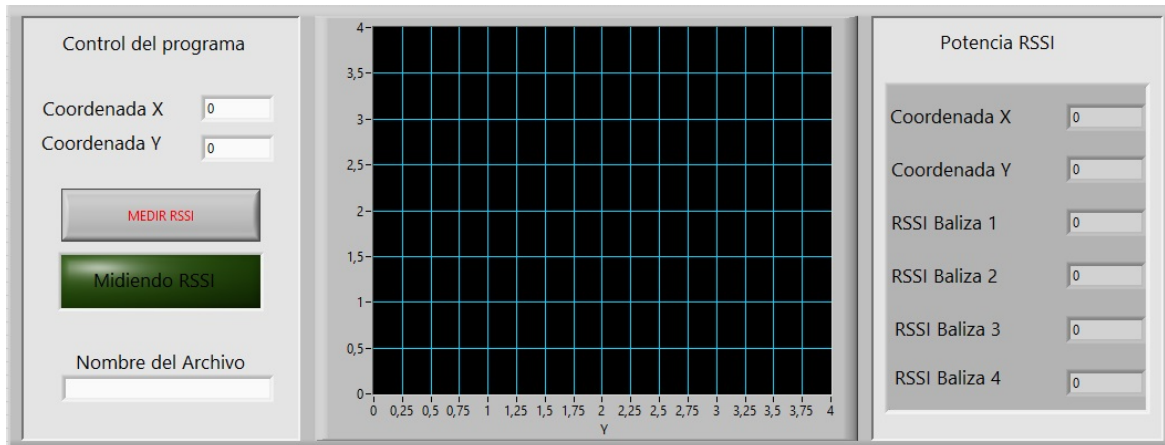
Seguidamente hay que colocar el nodo remoto en diferentes puntos del recinto y capturar la potencia RSSI que recibe de cada baliza. Cuantos más puntos se guarden mayor precisión y más real será el sistema.



**Ilustración 4.4-16** Localización basada en reconocimiento de patrones.

En la memorización de valores RSSI se dispone de una aplicación gráfica donde encontramos tres partes diferenciadas.

- Control del programa.
- Localización del robot.
- Indicación de resultados.



**Ilustración 4.4-17** Aplicación gráfica de registro de potencia RSSI.

En la parte izquierda se encuentra el control del programa. Se han de introducir las coordenadas X, Y donde está situado el módulo remoto y pulsar “Medir RSSI” el indicador luminoso indica que se están tomando los valores.

La aplicación manda una petición de potencia RSSI por el puerto serie del PC a cada una de las balizas. Éstas renvían esta petición al nodo móvil que les contesta con que potencia ha recibido el mensaje. Las balizas guardan ese valor y los reenvían al PC. Éste hace un checksum del mensaje si es correcto guarda el valor de potencia RSSI conociendo el remitente ya que esta escrito en el mensaje.

La aplicación realiza esta acción 10 veces, calcula la media de los valores obtenidos y los guarda junto a las coordenadas X, Y en una matriz.

Cuando acabe de medir un punto el diálogo nos preguntará si se desea medir otro punto o salir, si se desea salir los datos se exportaran en un archivo Excel con el nombre indicado.

En la parte central de la aplicación se encuentra la representación gráfica de la posición del nodo remoto.

En la parte derecha se indican los valores que se van a guardar.

- Coordenada X.
- Coordenada Y.
- RSSI del módulo Baliza 1.
- RSSI del módulo Baliza 2.
- RSSI del módulo Baliza 3.
- RSSI del módulo Baliza 4.

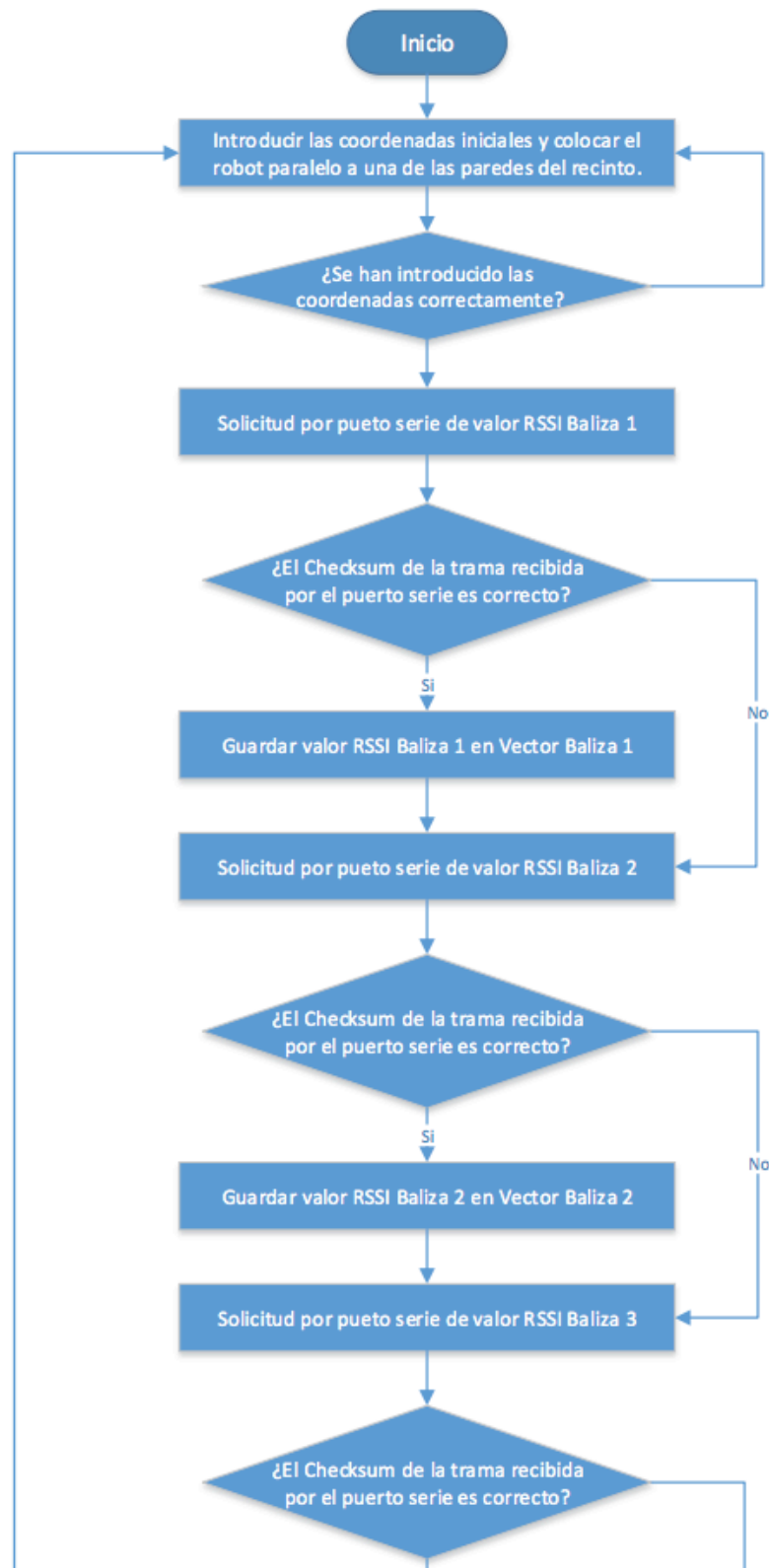


Ilustración 4.4-18 Diagrama de flujo registro de potencia RSSI (1/2).

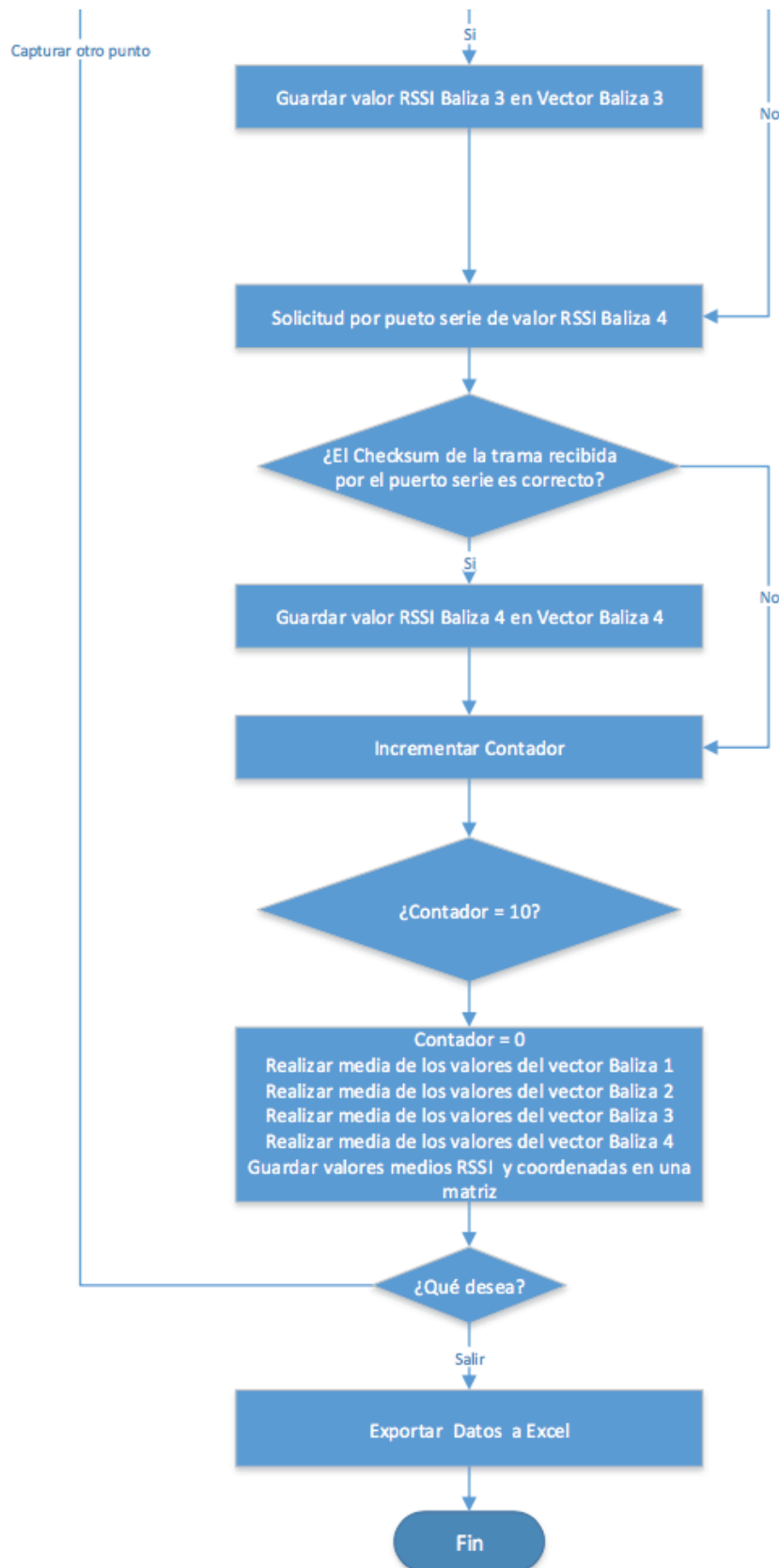


Ilustración 4.4-19 Diagrama de flujo registro de potencia RSSI (2/2).

La aplicación de guiado y localización tiene como objetivo indicarle al robot unos coordenadas X,Y a donde se desea que el robot se dirija y que se posicione con el menor error posible.

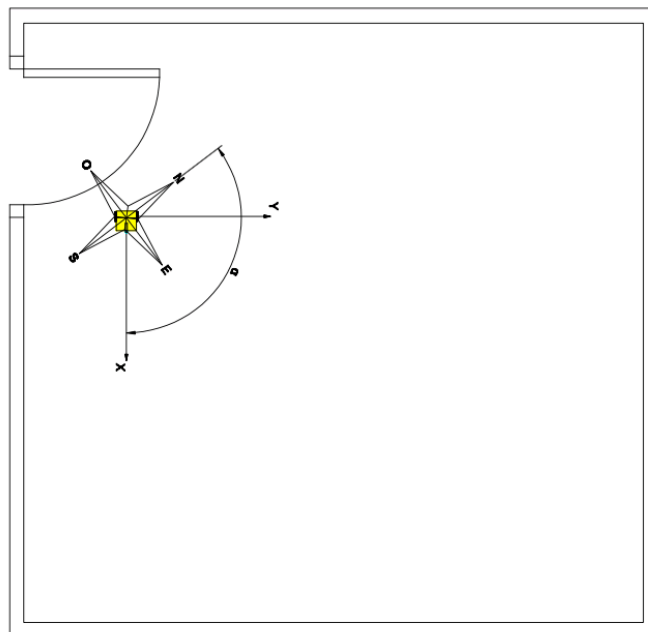
La aplicación grafica se divide en cuatro zonas.

- Gráfica 3D de los valores RSSI de cada módulo baliza de la zona de trabajo del robot .
- Control de la aplicación.
- Localización del robot por método de odometría.
- Resultados de localización del robot por intensidad de señal recibida.

Al iniciar la aplicación se ha de importar el archivo que contiene el mapa con valores RSSI de las balizas de la zona de trabajo del robot. Par obtener mayor resolución en la posición, se interpolan los valores de RSSI de los puntos del recinto done no se ha medido la potencia. El método de interpolación se selecciona en la parte de control de la aplicación en la parte superior derecha.

En la parte superior izquierda se pueden ver las Gráfica 3D de los valores RSSI de cada módulo baliza de la zona de trabajo del robot con los valores interpolados.

Por medio de un diálogo emergente se pide que se coloque el robot paralelo a una de las paredes de la sala. Se realiza esta paso con el fin de trasladar los ejes de la brújula (Norte, Sur, Este, Oeste) a los ejes de la sala, X, Y. Para ello se mide el rumbo en esta posición y durante todo el programa se tendrá en cuenta ese valor de rumbo como offset.



**Ilustración 4.4-20** Rotación de ejes X Y de referencia respecto al norte magnético.

Una vez introducidas las coordenadas iniciales y finales el sistema calcula el rumbo que debe llevar el robot y a la distancia que se encuentra, primero le envía por el puerto serie el comando que para que gire hasta tener el rumbo correcto y seguidamente que avance la distancia necesaria. Mientras el robot está en movimiento el indicador está encendido.

El parte inferior izquierda se encuentra la representación gráfica de la posición del robot y de la trayectoria que está siguiendo por el método de odometría.

Cuando el robot llega a su destino empieza la localización por medición de potencia de recepción y mientras ésta dura se enciende el indicador correspondiente.

La aplicación manda una petición de potencia RSSI por el puerto serie del PC a cada una de las balizas. Éstas reenvían esta petición al nodo móvil que les contesta con que potencia ha recibido el mensaje. Las balizas guardan ese valor y los reenvían al PC. Éste hace un checksum del mensaje si es correcto guarda el valor de potencia RSSI conociendo el remitente ya que está escrito en el mensaje.

La aplicación realiza esta acción 10 veces, calcula la media de los valores obtenidos. Una vez obtenidos estos valores se muestran en la parte inferior derecha de la aplicación gráfica (RSSI medido) y se emplea la técnica de localización descrita en el apartado 4.2.3 de esta memoria.

Se indica el valor mínimo de RSSI y la coordenada de esa posición que es donde se encuentra el robot. El valor de RSSI si el funcionamiento fuera ideal sería de 0, cuanto más alejado sea de 0 aumenta la probabilidad de que la indicación de posición del robot no sea correcta.

En la parte inferior derecha de la aplicación gráfica, se dispone de un indicador de intensidad donde se representa la zona de trabajo del robot, como se aprecia por la escala, en negro serían los puntos donde es más probable que se encuentre el robot y en blanco donde es menos probable.

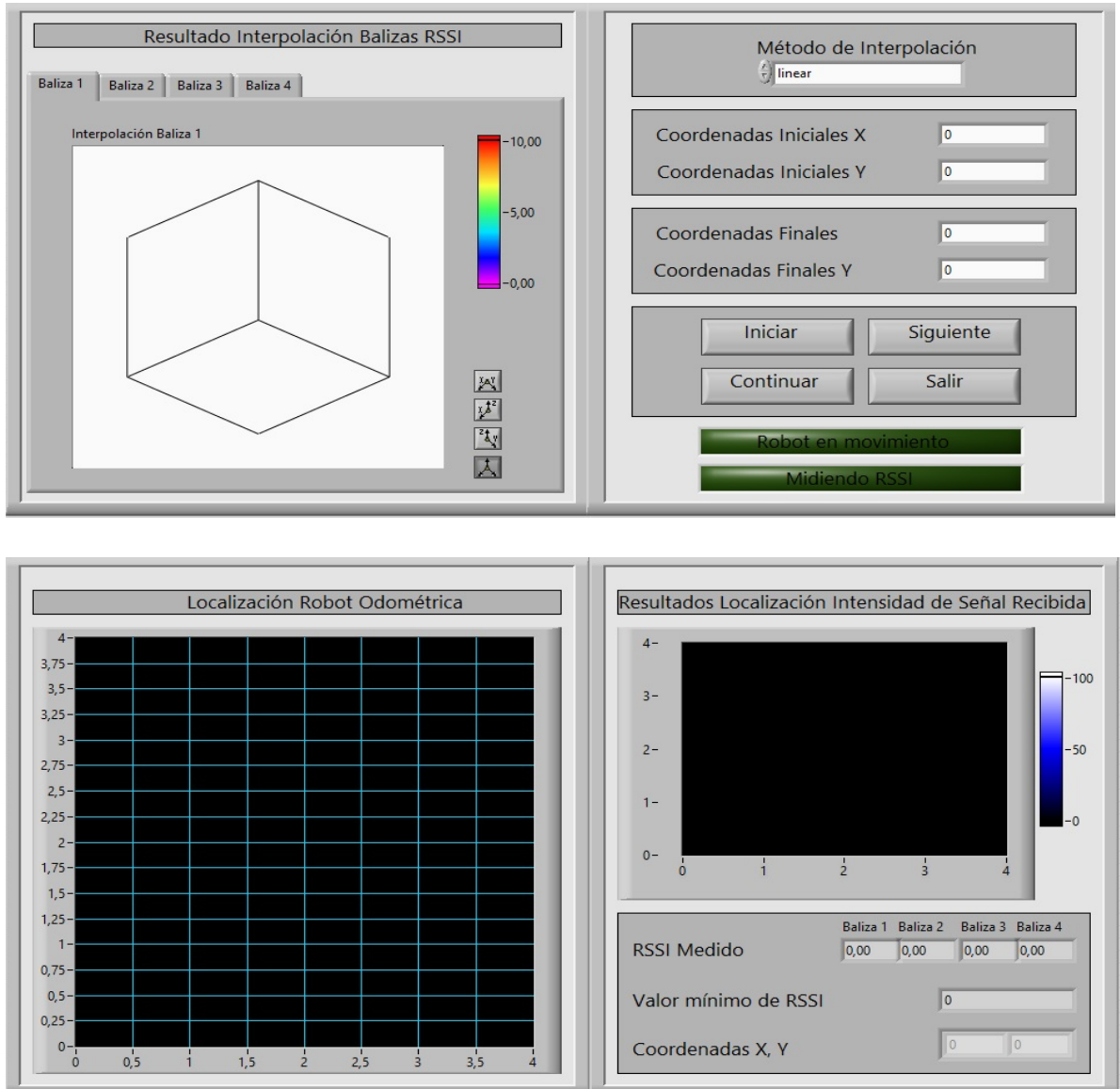
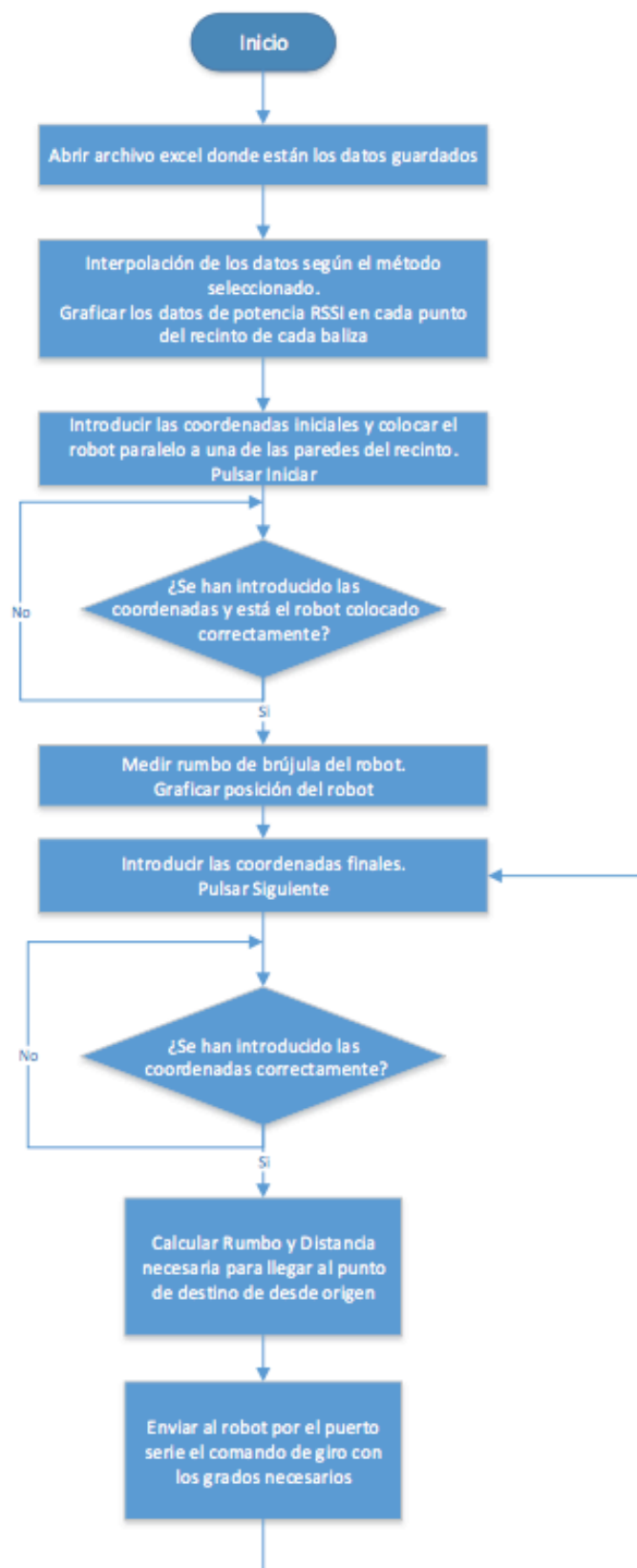


Ilustración 4.4-21 Aplicación gráfica localización odométrica y de medición de potencia de recepción.



**Ilustración 4.4-22** Diagrama de flujo aplicación localización odométrica y medición de RSSI (1/3).



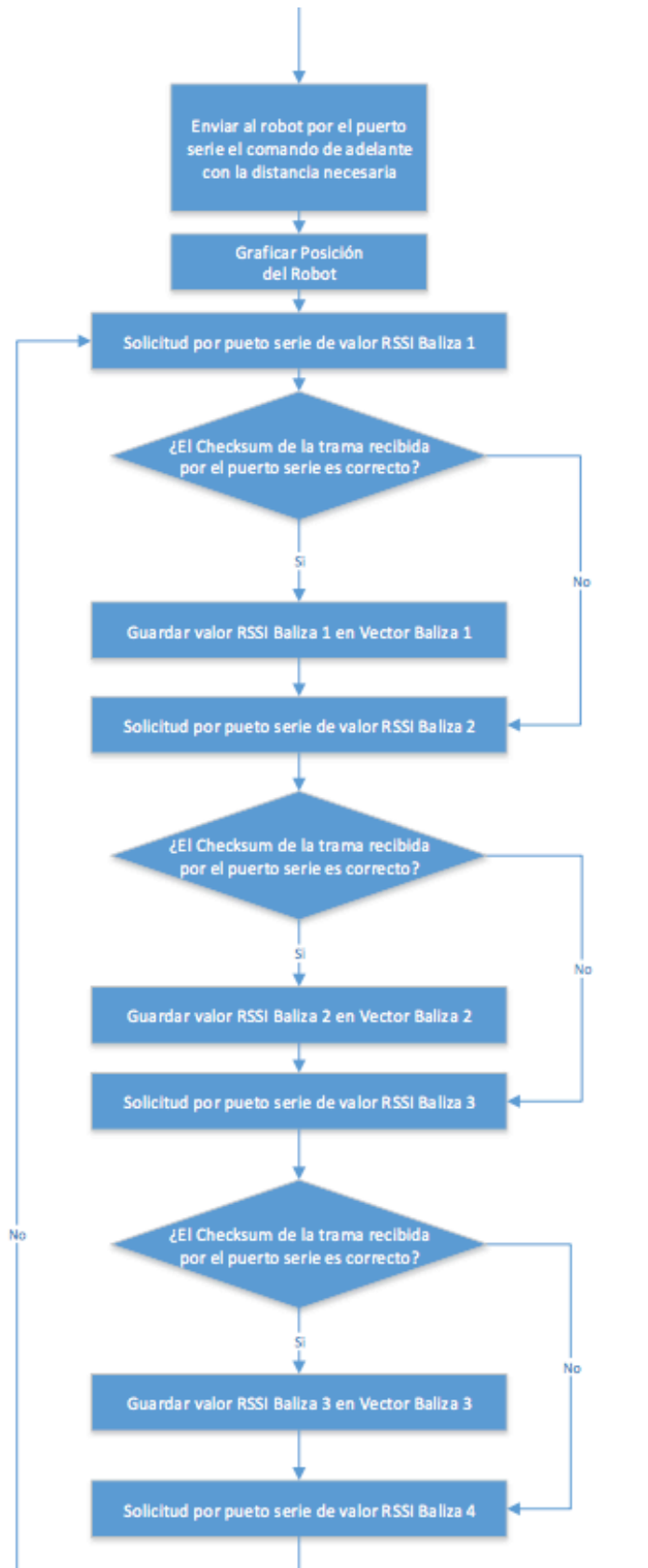


Ilustración 4.4-23 Diagrama de flujo aplicación localización odométrica y medición de RSSI (2/3).

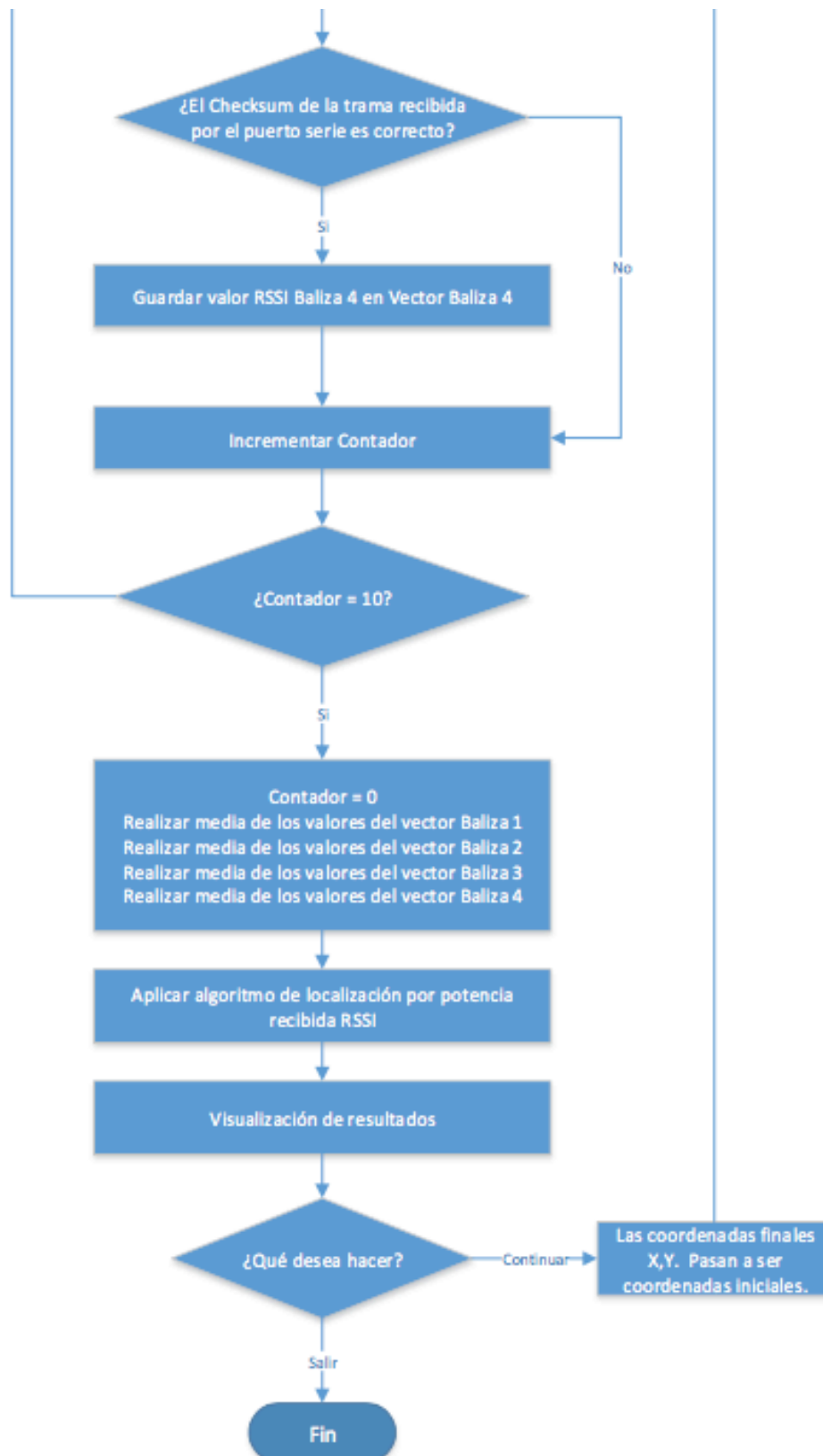


Ilustración 4.4-24 Diagrama de flujo aplicación localización odométrica y medición de RSSI (3/3).

#### 4.4.6. Herramientas de software.

A modo de resumen, en la siguiente tabla se muestran las herramientas software del proyecto, es decir, los programas que se utilizan en el proyecto y que no forman parte del software diseñado.

**Tabla 7** Herramientas de software.

| Nombre                    | Tipo     | Descripción  |
|---------------------------|----------|--|
| BasicX V.2.10             | Software | Programa para programar, compilar y cargar los programas del robot SR1 |
| DIGI XCTU 6.3.2           | Software | Programa de configuración y comunicación de módulos XBEE               |
| LabVIEW 2015 Versión 15.0 | Software | Software para el desarrollo integrado de sistemas.                     |



## 5. Análisis del impacto ambiental.

El impacto medioambiental del proyecto debe evaluarse desde dos puntos de vista que se complementan, aportando cada uno datos que ayudan a obtener una visión global de este. Por un lado está el ciclo de vida de los materiales y componentes empleados en el proyecto, y por otro la alteración del medio que este puede provocar, incluyendo en este segundo ámbito a las personas y al medio físico alrededor del proyecto durante su utilización.

Respecto al ciclo de vida de los componentes y materiales utilizados, una vez alcanzado el final de su vida útil, para la correcta gestión del reciclado y de los desechos se deberá llevar los componentes a algún centro de gestión de residuos de aparatos eléctricos y electrónicos (RAEE). Allí se separan, principalmente con procesos mecánicos, los elementos contaminantes de los no contaminantes. Los primeros se gestionan de forma apropiada mientras que los segundos se acumulan y transforman en materias primas aprovechables. En caso contrario el contenido en materiales contaminantes de estos componentes (PVC, PCB o TBBA entre otros) será altamente perjudicial para el medio ambiente. Aproximadamente el 70% de cada dispositivo se puede volver a aprovechar.

Se concluye por lo tanto, que el impacto medioambiental de cada dispositivo a lo largo de su ciclo de vida depende en gran medida de la manera en que se gestiona tanto su uso como su trato como residuo. En la medida en que se alargue al máximo su vida útil y reciba el trato adecuado como residuo electrónico en una planta capacitada para gestionar de forma integral los RAEE se puede reducir en gran medida su impacto medioambiental.

En cuanto al impacto medioambiental de los dispositivos utilizados en este proyecto, debe considerarse como causa principal las ondas de RF emitidas en la comunicación entre las distintas balizas. El efecto que esta pueda tener sobre su entorno está ligado principalmente a cuatro parámetros:

- La potencia generada por los módulos XBee.
- El alcance y rango de las ondas de Radio Frecuencia.
- Frecuencia electromagnética de la onda con que se trabaja.
- El tiempo de exposición a la radiación.

La potencia generada en el campo electromagnético de los módulos XBee Pro S1 es de 100 mW a una frecuencia de 2.4 GHz.

Según la Organización Mundial de la Salud los campos de RF por debajo de 10 GHz (a 1 MHz) penetran los tejidos expuestos y producen calor debido a la absorción de energía. La profundidad de penetración depende de la frecuencia del campo y es mayor para frecuencias menores. La absorción de los campos de RF en los tejidos es medida como una Tasa de Absorción Específica (SAR) dentro una masa de tejido dada. La unidad de la Tasa de Absorción Específica es vatios por kilogramo (W/Kg.). El SAR es la cantidad utilizada para medir la “dosis” de los campos de RF aproximadamente entre 1 MHz y 10 GHz.

Se necesita un SAR por lo menos de 4 W/kg para producir efectos adversos a la salud conocidos, en personas expuestas a los campos de RF en este rango de frecuencia.

En vista de los datos que tiene la OMS se puede afirmar que ante la baja potencia de los módulos de radio frecuencia, no se conoce que exista ningún impacto hacia las salud de las personas derivado de las radiaciones de los módulos, independientemente del tiempo de exposición.

## Pruebas y Resultados.

En este apartado se exponen las pruebas de las áreas que se han desarrollado a lo largo de su realización y los resultados finalmente obtenidos del mismo.

### Robot SR1.

El robot SR1, con el que se ha realizado el trabajo como se ha comentado anteriormente es fabricado y comercializado por la empresa INTPLUS, la cual es una pequeña empresa que ha distribuido su producto a pequeña escala por lo que no hay una comunidad de desarrolladores detrás. Se le hicieron unas consultas técnicas a la que la empresa no dio respuesta.

El sistema de hardware es muy rígido, no es sencilla la manipulación para la ampliación de módulos, el módulo de comunicación inalámbrica no funciona correctamente, el robot recibe datos que el pc le envía, pero no a la inversa.

El hecho de que el Robot esté conectado con cable limita su autonomía y afecta a su simetría siendo un lastre al lado derecho haciendo que tienda a ir en esa dirección.

Para el funcionamiento de la cámara es necesario un receptor y una capturadora o digitalizadora que permita conectar la señal de video del receptor de la cámara al PC. Y solo funciona el software del fabricante que acompaña el Robot, si se desea utilizar en un software propio es necesario comprar las librerías VideoCapX.

El robot corre sobre un microcontrolador concretamente el BasicX24P, como se relata en esta memoria la tendencia actual de los robots de investigación es que el microcontrolador vaya acompañado de una FPGA, funciona en tiempo real y son capaces de realizar varias tareas simultáneas.

Los robots actuales de investigación también cuentan con sensores redundantes que aseguran el correcto funcionamiento, comunicaciones WIFI y lo acompañan de plataformas software integradas de programación de alto nivel.

### Localización Odométrica.

Como se ha comentado en la memoria el inconveniente de la localización odométrica es la acumulación de errores, y eso es lo que se ha podido observar, cuanto más largo son los recorridos más crece la incertidumbre de la zona de localización del robot.

Según el fabricante la brújula tiene un error de  $\pm 3^\circ$  este error en distancias de un metro sería una desviación de 5 cm, error aceptable para el sistema pero la brújula es muy influenciada por los campos magnéticos por lo que es muy habitual que el rumbo que toma el robot no es que se desea por lo que la posición en la que el robot se sitúa es completamente errónea, esto ocurre de manera demasiado habitual por lo que el sistema localización por brújula tiene una fiabilidad muy baja.

También es frecuente que las ruedas del robot sufran deslizamientos o falta de tracción con el pavimento, situación que acarrea errores en la localización aunque menores que los derivados de los de error de rumbo.

## Localización por potencia recibida RSSI.

Si graficamos el modelo ideal según la ecuación de FRISS de los módulos XBEE se obtienen la siguiente gráfica.

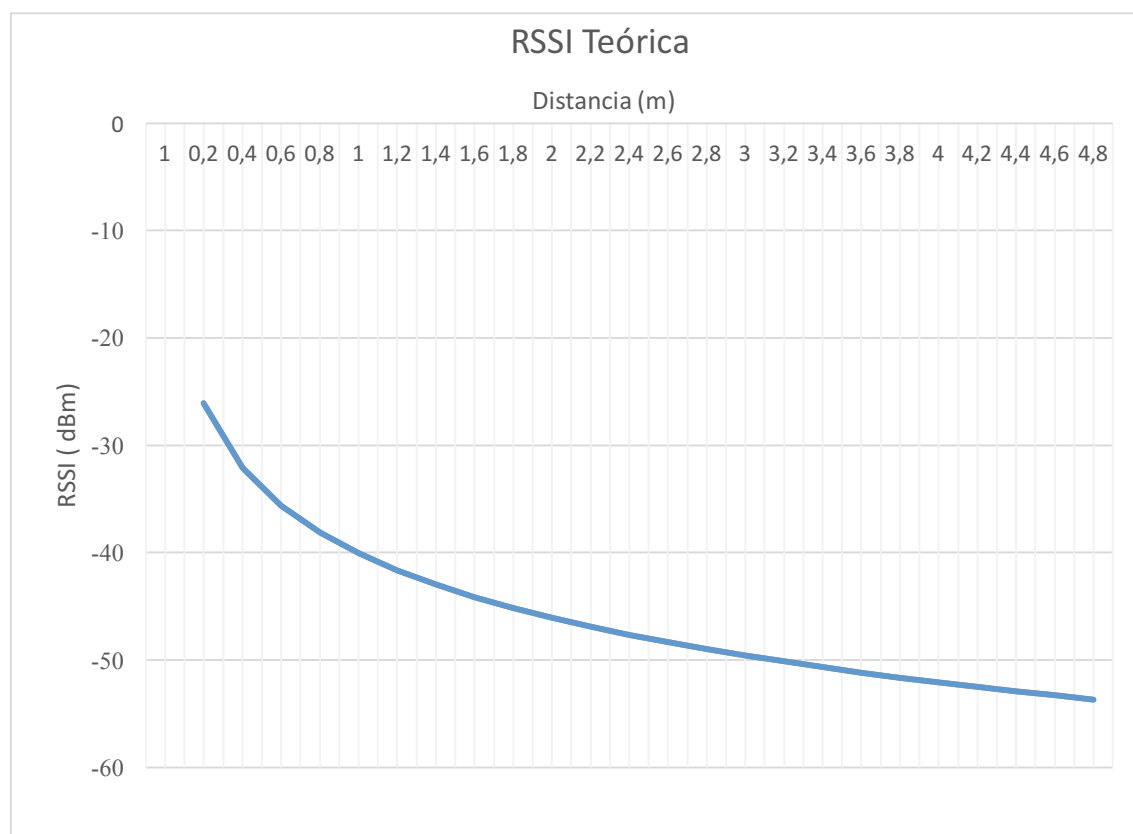
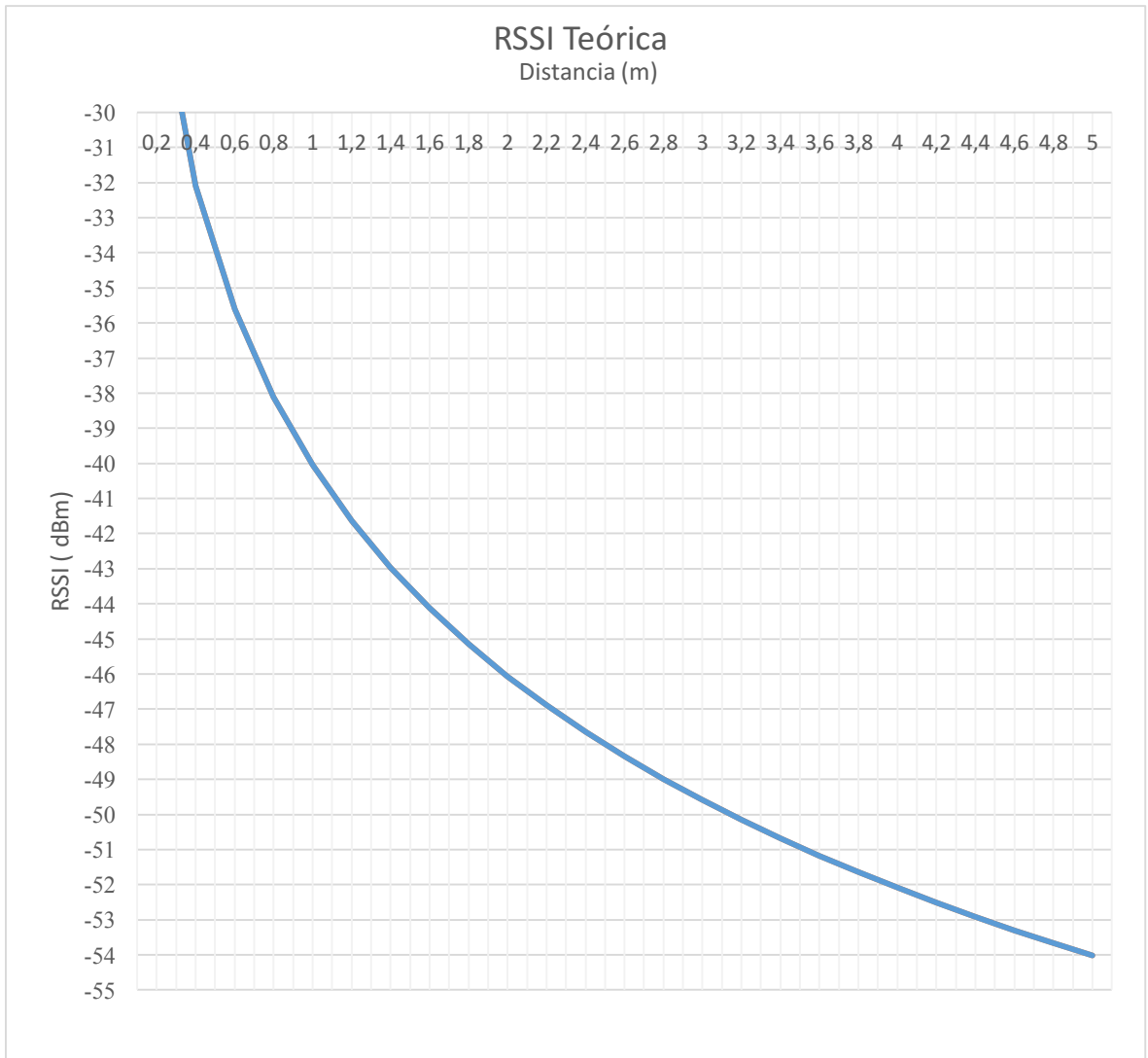


Ilustración C 1 Potencia RSSI teórica de los módulos Xbee.



Si se amplia se obtiene la siguiente gráfica:



**Ilustración C 2** Potencia RSSI teórica de los módulos Xbee.

Si se observa la gráfica detenidamente y teniendo cuenta que los módulos XBee tiene una resolución de 1dBm, a distancias superiores a 2,4m de la baliza, la incertidumbre en la medida debido únicamente a la resolución es de 0,4 m.

Se han realizado las mismas gráficas de las balizas en modo práctico en una zona diáfana sin obstáculos y los resultados obtenidos son los siguientes:

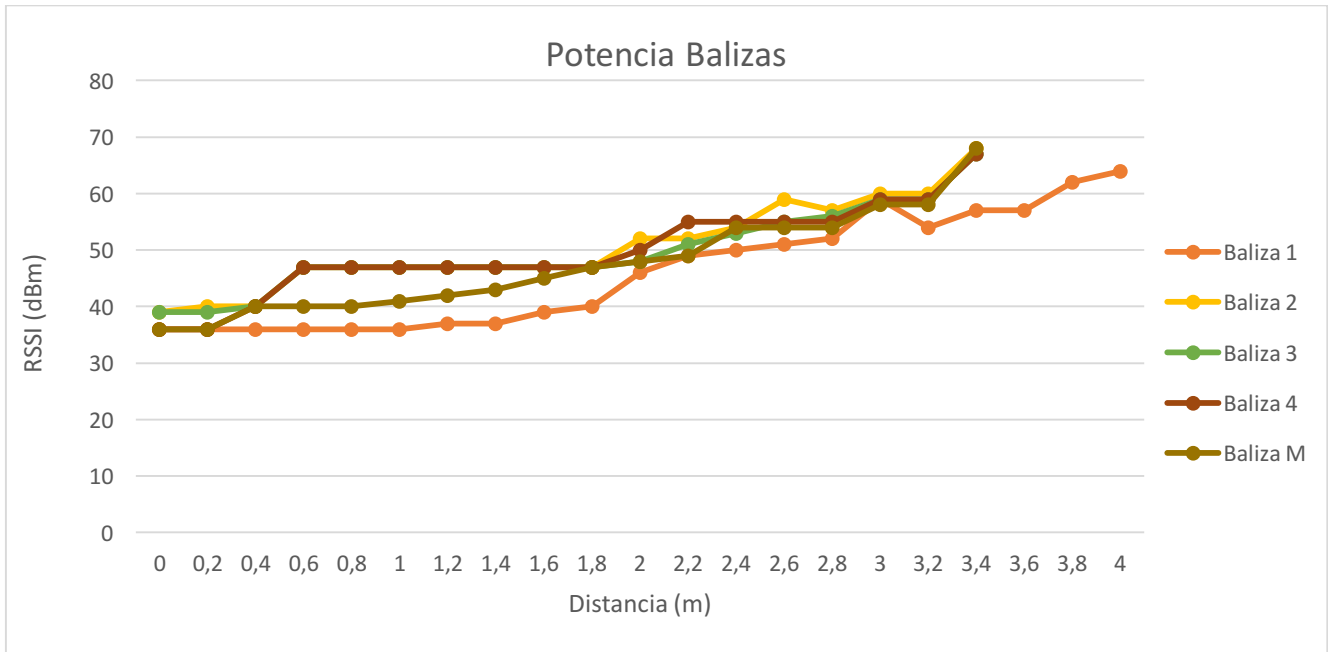


Ilustración C 3 Potencia RSSI real de los módulos XBee.

Los valores de RSSI son representados positivos dado que los módulos XBee los envían como positivos. Si ampliamos la zona como se ha hecho anteriormente se obtiene la siguiente gráfica.

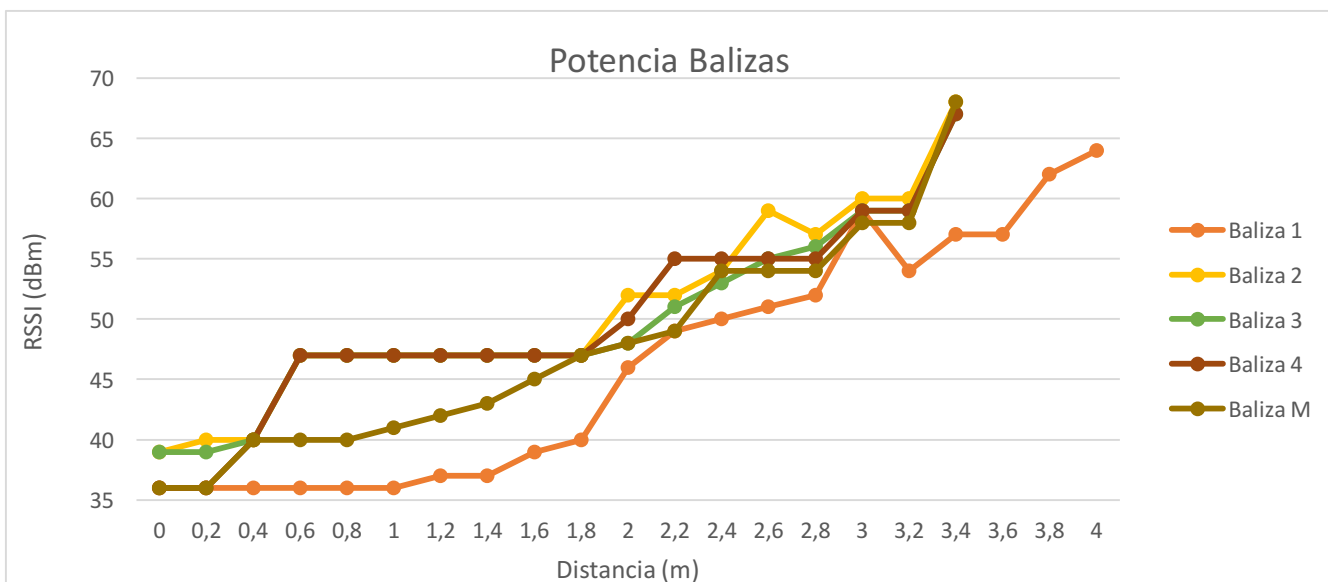


Ilustración C 4 Potencia RSSI real de los módulos XBee.

Si se observa la gráfica con detenimiento podemos ver que para distancias inferiores a 1,8 m la potencia de RSSI se mantiene casi constante por lo que la incertidumbre en la posición del robot en esa franja será de 1,8m.

La siguiente prueba pretende comprobar la repetitividad de los módulos XBee, el software X-CTU posee una herramienta de prueba de rango entre dos módulos conectados a la misma red.

Al escoger la opción de prueba de rango, se debe seleccionar el módulo local conectado al ordenador, es decir, el módulo acoplado al nodo coordinador, y uno de los módulos asociados a este de forma remota que puede ser cualquiera del resto de nodos que esté en funcionamiento. Para esta prueba es necesario que el nodo conectado de forma remota tenga el modo API deshabilitado. A continuación, se escoge el tipo de prueba "Loopback" y se asigna la opción "Loop infinitely" para que se realicen las pruebas del nivel de señal repetidamente y sin fin.

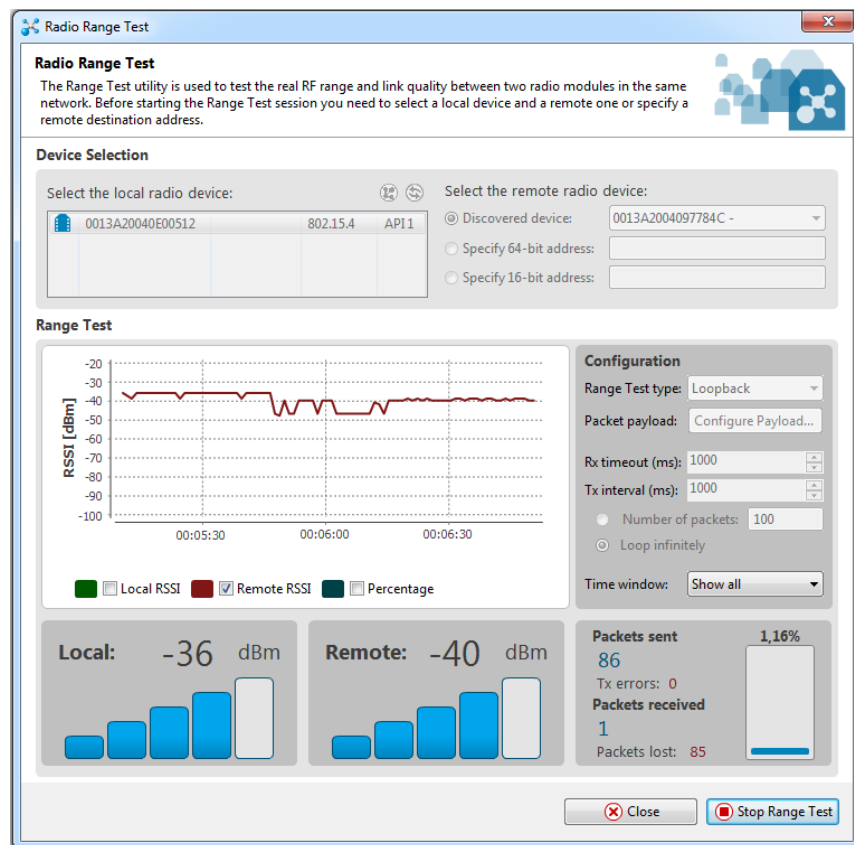


Ilustración C 5 Prueba de rango X-CTU.

Se ha realizado esta prueba a diferentes distancias y periodos de tiempo.

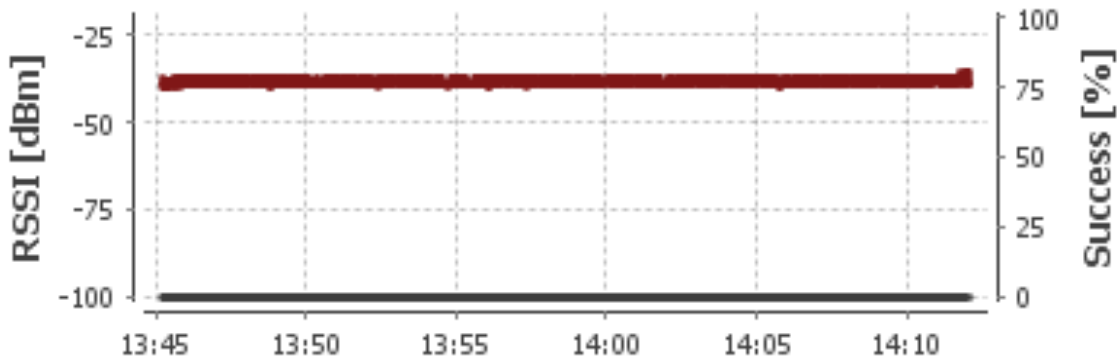


Ilustración C 6 RSSI distancia 0,80 m tiempo 30 minutos.

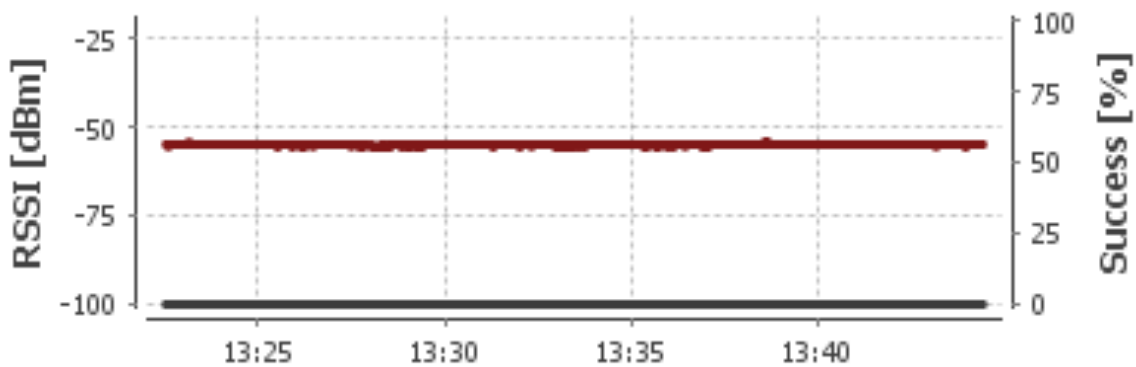


Ilustración C 7 RSSI distancia 2,5 m tiempo 30 minutos.



Ilustración C 8 RSSI distancia 3,6 m tiempo 10 minutos.

Como se aprecia en las gráficas, la repetitividad es variable a lo largo del tiempo puedes oscilar  $\pm 3$  dBm incluso más a medida que la distancia a la baliza es mayor, con la incertidumbre en la medida de distancia que acarrea este hecho.

## Aplicación localización por potencia recibida RSSI.

Si se observan los valores obtenidos después de cartografiar una área de 270 cm por 470 cm, se aprecia que las medidas de RSSI no siguen ningún modelo numérico y que es bastante arbitraria, la variación de RSSI que tienen en los diferentes puntos es debida más a la fluctuación de la potencia emitida que a la variación de la distancia con el receptor.

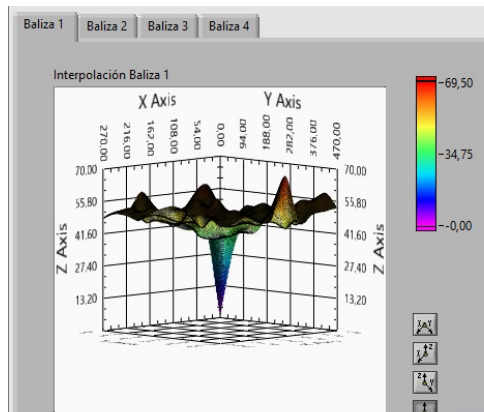


Ilustración C 9 Gráfica valores RSSI Baliza 1 (0,0).

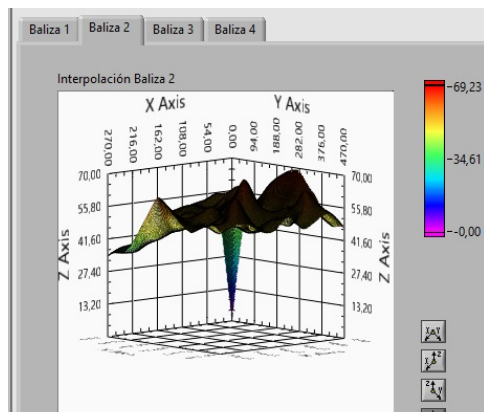


Ilustración C 10 Gráfica valores RSSI Baliza 2 (270,0).

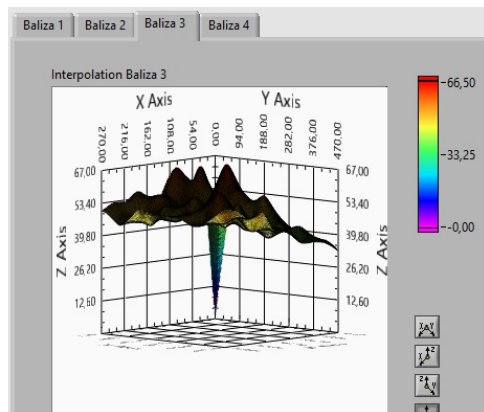


Ilustración C 11 Gráfica valores RSSI Baliza 3 (0,470).

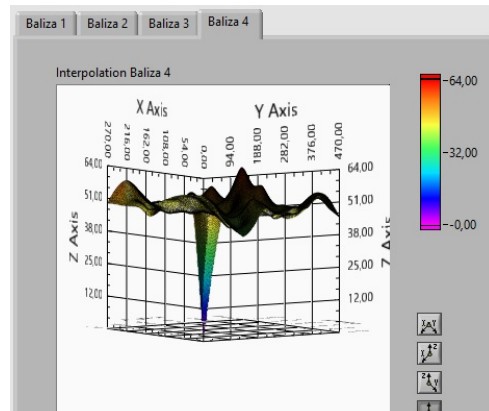


Ilustración C 12 Gráfica valores RSSI Baliza 4 (270,470).

En la Gráfica de resultados de localización por intensidad recibida se aprecia que hay varias áreas donde es posible que se encuentre el robot, dado la poca fiabilidad del sistema se ha limitado el área de búsqueda a  $\pm 1m$  de la posición de destino por localización odométrica.

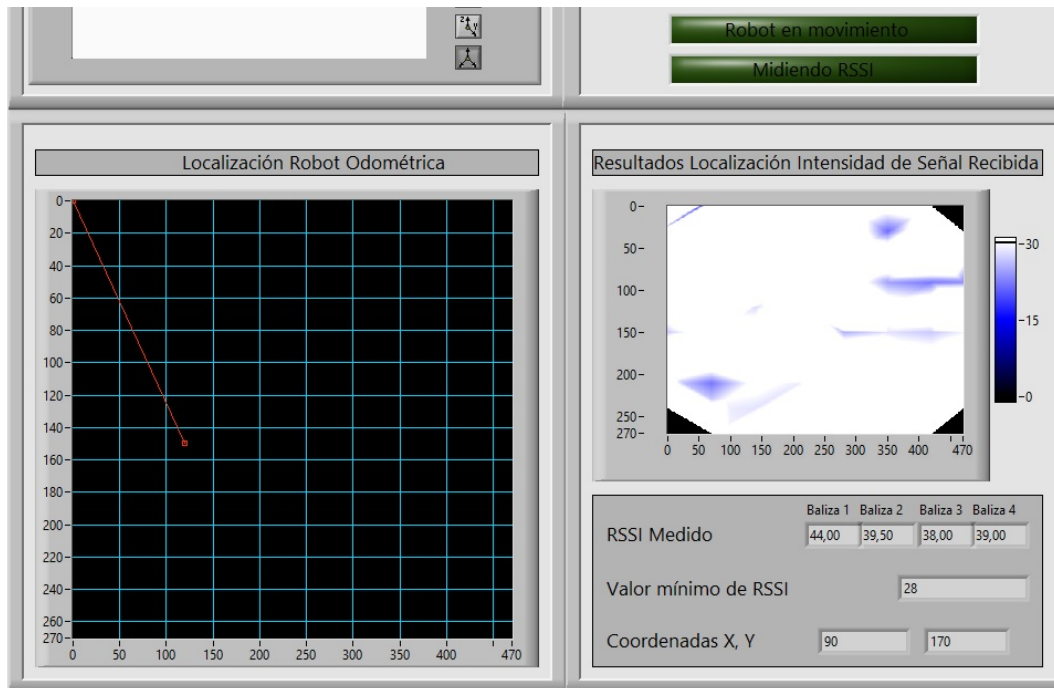


Ilustración C 13 Gráfica de resultados de localización por Intensidad de Señal Recibida.

## Conclusiones.

En este apartado se exponen las conclusiones obtenidas mediante la realización del trabajo. Gracias a este proyecto se han adquirido los conocimientos sobre el funcionamiento de las balizas XBee, los Robots que se comercializan y los diferentes sistemas de localización que existen.

Se ha comprobado que a nivel teórico el sistema funciona correctamente pero al llevarlo a la práctica influyen otros factores que no se pueden predecir y que hacen que los resultados no sean satisfactorios. Se ha reflejado el funcionamiento real de las balizas, como evoluciona la potencia RSSI en función con la distancia en la práctica y como es afectada por diferentes factores.

Se ha podido constatar como la localización odométrica depende en gran medida de la calidad de los sensores y actuadores y que el error aumenta a medida que se aumentan el número de movimientos.

El mayor aporte de haber realizado este trabajo han sido los conocimientos adquiridos sobre el lenguaje de programación LabVIEW. Se ha comprobado su gran potencial y funcionalidad, capaz de realizar todo tipo de acciones, comunicaciones con diferentes equipos, cálculos, resolución de algoritmos y representación gráfica de resultados de una forma muy intuitiva.

Durante el transcurso del proyecto no se ha dispuesto de todos los programas del robot, como por ejemplo los de calibración. También ha sido un inconveniente que no se reciban datos del robot mediante el módulo inalámbrico y que tenga que estar comunicado vía cable.

En el desarrollo del proyecto se ha intentado construir un sistema funcional disponiendo de una fácil implementación y puesta en marcha. El resultado no ha sido satisfactorio por lo que no se ha conseguido ese objetivo.

## Bibliografía

- [1] LabVIEW Users Manual NATIONALS INSTRUMENTS Versión Abril 2003 EDITION.
- [2] LabVIEW Tutorials Manual NATIONALS INSTRUMENTS Versión Enero 1996.
- [3] Getting Started with the LabVIEW™ Robotics Module Version Enero 2009.
- [4] LabVIEW™ Robotics Programming Guide for the FIRST Robotics Competition Versión Enero 2009.
- [5] Intelligent Control Systems with LabVIEW Fernando D. Ramírez-Figueroa, Research Assistant to Doctor Ponce Springer London Dordrecht Heidelberg New York 2010
- [6] Título: LabVIEW. Entorno gráfico de programación. Autores: José Rafael Lajara Vizcaíno José Pelegrí Sebastia MARCOMBO 2007
- [7] OMS. “Campos electromagnéticos”. Último acceso: 15 de abril, 2017  
<http://www.who.int/peh-emf/publications/facts/fs226/es/>
- [8] Albert Bachs Alonso “Localización en interiores” Trabajo fin de carrera, EUETIB, 2015.
- [9] Javier Salvador Marco, Joan Martínez Recasens, “Montaje, Calibración y programación del Robot SR1” Trabajo fin de carrera, UPC 2007,
- [10] Eduardo Rivas González. “Localización mediante differential evolution de un robot móvil” Trabajo fin de carrera, Universidad Carlos III de Madrid Escuela Politécnica Superior.
- [11] Miguel Ángel Alonso Rios. “Sistema de localización de objetos en espacios cerrados por medio de RF” Trabajo fin de carrera, UPC 2007,
- [12] Javier Salvador Marco, Joan Martínez Recasens. “Montaje, Calibración y programación del Robot SR1” Trabajo fin de carrera, Ciudad Universitaria México D.F. 2012.









UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH  
Escola d'Enginyeria de Barcelona Est

TRABAJO FIN DE GRADO

Grado en Ingeniería Industrial y Automática.

**CONTROL Y GUIADO DE UN ROBOT MÓVIL.**



**Presupuesto.**

**Autor:** Raúl Cruzado Muñoz  
**Director:** Juan Gámiz  
**Convocatoria:** Mayo 2017



## Índice Presupuesto

|                                   |   |
|-----------------------------------|---|
| Importe de mano de obra.          | 2 |
| Importe de material.              | 3 |
| Importe de licencias de software. | 3 |
| Otros Importes.                   | 4 |
| Costes totales.                   | 4 |

## Presupuesto.

En este capítulo se establecen los costes asociados a la elaboración del prototipo del sistema para el control y guiado del robot móvil.

Los costes se desglosan en los siguientes grupos:

- Mano de obra
- Material
- Licencias software
- Otros costes

### Importe de mano de obra.

En este apartado, la mano de obra se divide en ingenieril, de supervisión y técnica atendiendo a la naturaleza de las tareas realizadas y/o propuestas, siendo éstas el desarrollo conceptual, la dirección del desarrollo y el montaje del sistema de control, respectivamente. Con ello, la siguiente tabla especifica los costes asociados a la mano de obra para unas determinadas horas de trabajo.

| Importe mano de obra                 |                  |                       |                 |
|--------------------------------------|------------------|-----------------------|-----------------|
| Tipo de mano de obra                 | Cantidad [horas] | Precio unitario [€/h] | Coste total [€] |
| Ingenieril                           | 300              | 25,00                 | 7500,00         |
| Supervisión                          | 20               | 90,00                 | 1800,00         |
| Técnica                              | 80               | 15,00                 | 1200,00         |
| <b>Subtotal importe mano de obra</b> |                  |                       | <b>10500,00</b> |

## Importe de material.

A continuación se muestran en forma de tabla los costes asociados a la adquisición del material necesario para la elaboración del estudio así como para la implementación de los dispositivos propuestos a lo largo de dicho estudio.

| Importe material                  |              |                       |                 |
|-----------------------------------|--------------|-----------------------|-----------------|
| Material                          | Cantidad [u] | Precio unitario [€/u] | Coste total [€] |
| Kit completo Robot SR1            | 1            | 619,95                | 619,95          |
| Módulos Xbee Pro S1               | 6            | 42,00                 | 252,00          |
| Baterias BRC 18650 5000 mAh 3,7 V | 10           | 4,00                  | 40,00           |
| Placa Xbee USB Adapter            | 1            | 5,64                  | 5,64            |
| Cable USB - DB9                   | 1            | 5,77                  | 5,77            |
| Cable USB - miniUSB               | 1            | 1,80                  | 1,80            |
| Cargador de Baterias 18650        | 1            | 4,49                  | 4,49            |
| Microcontrolador PIC18LF2550      | 5            | 4,47                  | 22,35           |
| Soporte batería 18650             | 5            | 4,48                  | 22,40           |
| Pequeño material electrónico      | 1            | 50,00                 | 50,00           |
| <b>Subtotal importe material</b>  |              |                       | <b>1024,40</b>  |

## Importe de licencias de software

A continuación se muestran los costes de las diversas licencias de software empleadas durante el desarrollo del estudio.

| Importe licencia software                 |              |                       |                 |
|---|--------------|-----------------------|-----------------|
| Licencia software                         | Cantidad [u] | Precio unitario [€/u] | Coste total [€] |
| LabVIEW 2015 Version 15.0                 | 1            | 2431,00               | 2431,00         |
| Microsoft office 2016                     | 1            | 99,99                 | 99,99           |
| DIGI XCTU 6.3.2                           | 1            | 0,00                  | 0,00            |
| BasicX 2.10                               | 1            | 0,00                  | 0,00            |
| <b>Subtotal importe licencia software</b> |              |                       | <b>2530,99</b>  |

## Otros Importes.

| Otros costes                 |              |                       |                 |
|------------------------------|--------------|-----------------------|-----------------|
| Concepto                     | Cantidad [u] | Precio unitario [€/u] | Coste total [€] |
| Ordenador Portatil           | 1            | 699,00                | 699,00          |
| Documentación                | 1            | 90,00                 | 90,00           |
| Matricula EEBE               | 1            | 1037,56               | 1037,56         |
| <b>Subtotal otros costes</b> |              |                       | <b>1826,56</b>  |

## Costes totales.

En la siguiente tabla se resumen el total de los costes asociados.

| Tipo de coste                      | Coste [€]       |
|------------------------------------|-----------------|
| Subtotal importe mano de obra      | 10500,00        |
| Subtotal importe material          | 1024,40         |
| Subtotal importe licencia software | 2530,99         |
| Subtotal otros costes              | 1826,56         |
| <b>Total</b>                       | <b>15881,95</b> |