

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica



**Diseño de una red inalámbrica de sensores para la monitorización del
desempeño estructural de puentes**

**Informe de Proyecto de Graduación para optar por el título de Ingeniera en
Electrónica con el grado académico de Licenciatura**

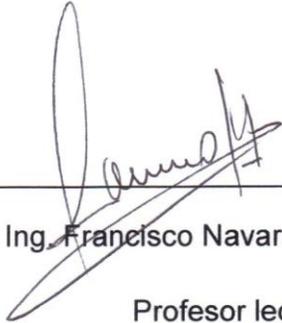
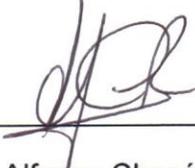
Eugenia María Guzmán Rivera

Cartago, Enero del 2012

INSTITUTO TECNOLOGICO DE COSTA RICA
ESCUELA DE INGENIERIA ELECTRONICA
PROYECTO DE GRADUACIÓN
TRIBUNAL EVALUADOR

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal

 _____ Ing. Francisco Navarro Henríquez Profesor lector	 _____ Dr. Alfonso Chacón Rodríguez Profesor lector
 _____ Ing. Johan Carvajal Godínez Profesor asesor	

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Cartago, 23 de enero del 2012

Declaratoria de autenticidad

Declaro que el presente proyecto ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo realizado y por el contenido del correspondiente proyecto

Cartago, 23 de enero del 2012

_____

Firma del autor

Eugenia María Guzmán Rivera

Cédula: 3-420-916

Resumen

El siguiente proyecto trata sobre el diseño de una red inalámbrica de sensores y la caracterización de sensores para la medición del desempeño estructural de puentes, con el fin de resolver el problema tan común en los últimos años en los puentes debido a su deterioro y falta de mantenimiento. Este problema le ha traído al país no solo gastos económicos muy elevados sino también pérdida de vidas y congestionamientos en las carreteras.

La falta de mantenimiento y de conocimiento real del estado de la infraestructura de puentes en el país, hace que sea necesaria la implementación de un sistema capaz de conocer el estado real de la estructura de los puentes para así lograr darle el mantenimiento adecuado a éstos y evitar los accidentes.

Se describirá la solución a éste problema caracterizando sensores que existen en el mercado para este fin; los cuales son: el transductor de tensión ST350 para la medición de la tensión superficial en la superficie del material y el SR50A, que es un sensor ultrasónico para medir distancias, fabricados por la empresa estadounidense Campbell's Scientific.

Se explicará la implementación de sistemas de medición para cada uno de los sensores, los cuales deben ser viables económicamente y de fácil uso para los inspectores de puentes. Además, se incluirá un manual de instalación y mantenimiento para cada sensor, aparte de un manual de utilización del sistema.

Palabras claves: puentes, caracterización, tensión superficial, deflexión, ST350, SR50A.

Abstract

This project is about the characterization of sensors for the measure of the bridge structure performance to resolve the common problem in bridges, caused by their deterioration and lack of maintenance. This problem is very expensive for Costa Rica and causes life losses and traffic congestion.

The lack of maintenance and knowledge about the real condition of the bridges in the country, make necessary the development of a system able to know the real state of the bridges structure and avoid accidents.

One solution of this problem will be describe with the characterization of the ST350 transducer for the measure of Strain in structures and the SR50A, which is an ultrasonic transducer for the measure of distance. Both sensors are manufactured by Campbell's Scientific.

The systems of measure for each sensor will be explained and they will be economically viable and easy construction. Besides, a manual of the calibration, installation and maintenance of the sensors and the system will be included.

Key word: bridges, characterization, ST350, SR50A, strain, deflection.

ÍNDICE GENERAL

Capítulo 1. Introducción	14
1.1 Antecedentes	14
1.2 Solución seleccionada.....	15
Capítulo 2. Meta y objetivos	17
2.1 Meta.....	17
2.2 Objetivo general	17
2.3 Objetivos específicos	17
Capítulo 3. Marco teórico	18
3.1 Descripción del sistema	18
3.2 Variables de desempeño estructural utilizadas.....	18
3.2.1 Deflexión.....	18
3.2.2 Mediciones de tensión superficial en el material de la viga	21
3.2.3 Circuito Acondicionador de Señales (CAS) [17]	23
3.2.4 Waspote [13].....	26
3.2.5 Protocolo FTP	29
Capítulo 4. Proceso metodológico.....	31
4.1 Síntesis del problema y requerimientos	31
4.2 Obtención y análisis de información.....	31
4.3 Evaluación de las alternativas y síntesis de la solución	32
4.4 Implementación de la solución	32
4.5 Reevaluación y diseño	33
Capítulo 5. Descripción detallada de la solución	35
5.1 Análisis de soluciones y selección final.....	35
5.1.1 Soluciones sensor de tensión superficial ST350	35
5.1.2 Soluciones sensor de deflexión SR50A	37
5.1.3 Red inalámbrica de sensores	38
5.1.4 Recepción, procesamiento y envío de datos en la PC	39
5.2 Descripción del hardware.....	40
5.2.1 Módulos de hardware para el sensor ST350.....	40

5.2.2	Módulo para el procesamiento de la señal del sensor SR50A	45
5.2.3	Red inalámbrica de sensores	48
5.3	Descripción del software	50
5.3.1	Programación del PIC para el sensor SR50A	50
5.3.2	Red inalámbrica de sensores	54
5.3.3	Programa para la recepción, procesamiento y envío de la información en la PC	62
5.4	Calibración de sensores	65
5.4.1	Sensor ST350 para la medición de la tensión superficial	65
5.4.2	Sensor SR50A para la medición de la deflexión [6]	66
5.5	Instalación de los sensores	70
5.3.1	Sensor ST350 [7]	70
5.3.2	Sensor SR50A [6]	72
5.6	Mantenimiento de los sensores	75
5.6.1	Sensor ST350	75
5.6.2	Sensor SR50A	75
Capítulo 6.	Análisis de resultados	76
6.1	Resultados	76
6.1.1	Resultados para el sensor ST350	76
6.1.2	Resultados para el sensor SR50A	81
6.1.3	Caracterización de los sensores	83
6.1.4	Red inalámbrica de sensores y recepción en la PC	86
6.1.5	Resultados del sistema	87
6.2	Análisis de resultados	93
6.2.1	Análisis de resultados para el sensor ST350	93
6.2.2	Análisis de resultados para el sensor SR50A	95
6.2.3	Análisis de resultados para la caracterización de los sensores	95
6.2.4	Análisis de resultados para la red inalámbrica de sensores y el programa en la PC	96
6.2.5	Análisis de resultados del sistema completo	97
Capítulo 7.	Conclusiones y recomendaciones	98

7.1 Conclusiones.....	98
7.2 Recomendaciones	99
Bibliografía	101
Apéndices.....	103
A.1 Glosario, abreviaturas y simbología.....	103
A.2 Manuales de usuario	104
A.2.1 Manual de instalación del sensor ST350 [7]	104
A.2.2 Manual de instalación del sensor SR50A [6]	106
A.2.3 Manual de mantenimiento para el sensor ST350.....	112
A.2.4 Manual de mantenimiento del sensor SR50A.....	112
A.2.5 Manual de usuario del sistema	113
Anexos	130
Anexo B.1 Hoja de datos de los módulos Waspote	130
Anexo B.2 Sensor ST350.....	132
Anexo B.3 Sensor SR50A.....	135

ÍNDICE DE FIGURAS

Figura 1. Etapas del sistema de caracterización estructural de puentes	18
Figura 2. Viga Deflectada [1]	19
Figura 3. Sensor SR50A [6].....	19
Figura 4. Puentes internos para cambiar de formato de salida [6]	20
Figura 5. Ángulo de trabajo del SR50A [6]	21
Figura 6. Diagrama eléctrico del sensor ST350 de Campbell Scientific. [7]	22
Figura 7. Esquema general de un CAS [17]	23
Figura 8. Esquema básico de un CAS [17].....	24
Figura 9. Circuito básico para un CAS [17]	24
Figura 10. Esquema alternativo de un CAS [17].....	25
Figura 11. Circuito alternativo para un CAS [17]	25
Figura 12. Módulo Waspnote [13].....	26
Figura 13. Conectores I/O en Waspnote [13]	27
Figura 14. Indicadores LED's en Waspnote [13]	28
Figura 15. Protocolo FTP [3].....	29
Figura 16. Topología de la red inalámbrica de sensores utilizada	35
Figura 17. Estructura de trama API zigbee [12].....	39
Figura 18. Módulos de hardware para el sensor ST350	40
Figura 19. Esquema básico de un circuito restador con OPAM's [2].....	40
Figura 20. Circuito restador implementado para el sensor ST350.....	42
Figura 21. CAS implementado para el sensor ST350	44
Figura 22. Esquemático del hardware diseñado para el sensor ST350	45
Figura 23. Esquema utilizado para el MAX232 [14].....	46
Figura 24. Esquemático para el módulo de hardware diseñado para el sensor SR50A.....	47
Figura 25. Diseño final del PCB para el módulo del sensor SR50A	47
Figura 26. Circuito típico para la programación con ICSP.[15].....	48
Figura 27. Módulo XBee 802.15.4 [13]	49
Figura 28. Waspnote Gateway [13]	49

Figura 29. Topología en estrella [13].....	50
Figura 30. Diagrama de flujo del programa principal para el PIC18F4550	51
Figura 31. Continuación rutina del programa principal para el PIC18F4550	53
Figura 32. Rutina de interrupción por puerto serie	54
Figura 33. Diagrama de flujo para el programa principal del Waspnote para la medición de la tensión superficial.....	55
Figura 34. Diagrama de flujo del Waspnote para la lectura de la medición de la deflexión	57
Figura 35. Pantalla principal X-CTU	59
Figura 36. Configuración para la comunicación de los módulos XBee con la PC	60
Figura 37. Prueba de comunicación entre el XBee y la PC.....	61
Figura 38. Configuración de los módulos XBee.....	62
Figura 39. Diagrama de flujo del procesamiento de datos en la PC.....	63
Figura 40. Diagrama de flujo del envío de datos por FTP	64
Figura 41. Diagrama de flujo del guardado en el archivo	65
Figura 42. Remover los tornillos de la cubierta del transductor SR50A y desconectar los cables [6].....	67
Figura 43. Absorbentes de humedad del sensor SR50A [6].....	68
Figura 44. Remover tornillos para exponer la PCB del SR50A [6]	68
Figura 45. PCB del SR50A [6].....	69
Figura 46. Menú en el hyperterminal para la configuración del sensor SR50A [6]	70
Figura 47. Marcas para la alineación [7].....	71
Figura 48. Instalación del sensor SR50A [7].....	71
Figura 49. Ángulo de medición del sensor [6].....	73
Figura 50. Distancia de la parrilla frontal a la cubierta de plástico [6].....	73
Figura 51. Montaje utilizando el Kit de montaje 19517 [6]	74
Figura 52. Montaje utilizando un cilindro y el UN-RAIL [6]	74
Figura 53. Simulación para el CAS del sensor ST350 utilizando Multisim	76
Figura 54. Simulación para el CAS del sensor ST350 cambiando el valor de R6	77
Figura 55. Resultados de la prueba experimental para el CAS del sensor ST35078	
Figura 56. Sensor instalado en la tabla de madera para las pruebas experimentales.....	79

Figura 57. Gráfico de las mediciones realizadas al sensor ST350 con el datalogger y con el sistema, datos tomados de la tabla 1.....	81
Figura 58. Gráfico de la distancia medida para el sistema y experimentalmente para el sensor SR50A	83
Figura 59. Datos recibidos en la PC provenientes de los nodos en la red inalámbrica de sensores	86
Figura 60. Archivo creado por la PC para guardar los promedios de los datos recibidos.....	87
Figura 61. Carpeta que contiene los programas y archivos creados para la PC..	88
Figura 62. Recepción de datos en la PC	89
Figura 63. Al iniciar el programa en la PC se crea el archivo .csv.....	89
Figura 64. Servidor donde se enviará el archivo por FTP.	90
Figura 65. Envío del archivo por FTP	91
Figura 66. Archivo .csv creado por la PC	92
Figura 67. Recepción en el servidor FTP del archivo	93
Figura A.2.1.1. Marcas para la alineación [7]	104
Figura A.2.1.2. Instalación del sensor SR50A [7]	105
Figura A.2.2.1. Remover los tornillos de la cubierta del transductor SR50A y desconectar los cables [6].....	106
Figura A.2.2.2. Absorbentes de humedad del sensor SR50A [6]	107
Figura A.2.2.3. Remover tornillos para exponer la PCB del SR50A [6].....	107
Figura A.2.2.4. PCB del SR50A [6]	108
Figura A.2.2.5. Menú en el hyperterminal para la configuración del sensor SR50A [6]	109
Figura A.2.2.6. Ángulo de medición del sensor [6]	110
Figura A.2.2.7. Distancia de la parrilla frontal a la cubierta de plástico [6]	110
Figura A.2.2.8. Montaje utilizando el Kit de montaje 19517 [6].....	111
Figura A.2.2.8. Montaje utilizando un cilindro y el UN-RAIL [6].....	111
Figura A.2.5.1 Waspnote IDE.....	113
Figura A.2.5.2 Programa para la lectura del sensor ST350 en el Waspnote.....	114
Figura A.2.5.3 Cambio del nombre que identifica el sensor en la red	115
Figura A.2.5.4 Compilación del programa	116

Figura A.2.5.5	Verificación de que la compilación fue exitosa.....	117
Figura A.2.5.6	Conector para el cable USB y el Switch de ON/OFF en el Waspnote [13]	118
Figura A.2.5.7	Programación del módulo Waspnote	119
Figura A.2.5.8	Programa para la lectura del sensor SR50A en el Waspnote	121
Figura A.2.5.9	Cambio del nombre que identifica el sensor en la red	122
Figura A.2.5.10	Serial Monitor del Waspnote IDE	124
Figura A.2.5.11	Abrir archivo Lectura_USB.py.....	125
Figura A.2.5.12	Configuración del puerto serie en el programa Lectura_USB.py	126
Figura A.2.5.13	Declaración de la variable de distancia inicial para el sensor SR50A.....	126
Figura A.2.5.14	Datos del servidor al que se enviará el archivo .csv	127
Figura A.2.5.15	Identificadores para cada sensor.....	128
Figura A.2.5.16	Definición de la ecuación para el cálculo de la tensión en el material para el sensor ST350	128
Figura A.2.5.17	Comando para correr el programa en la consola de Linux	129
Figura B.1.1	Hoja de datos del Waspnote.....	130
Figura B.1.2	Características generales de los módulos XBee.....	131
Figura B.2.1	Especificaciones generales para el sensor ST350	132
Figura B.2.2	Calibración del sensor ST350 con número de serie B2465	133
Figura B.2.3	Especificaciones sobre la calibración del sensor ST350	134
Figura B.3.1	Especificaciones generales del sensor SR50A.....	135

ÍNDICE DE TABLAS

Tabla 1. Valores de tensión de salida medidos con el datalogger y con el sistema y su porcentaje de error.	80
Tabla 2. Distancias medidas con el sistema y el metro para el sensor SR50A.....	82

Capítulo 1. Introducción

1.1 Antecedentes

En Costa Rica, se ha hecho cada vez más común escuchar en los medios de comunicación, noticias sobre los problemas existentes en el país debido al deterioro de los puentes y la infraestructura de transporte en general.

Al revisar las noticias del año 2010, se puede observar que más de 900 tienen relación con el tema de mantenimiento y fallas en los puentes de Costa Rica. Como ejemplo, se tienen el derrumbe del puente de hamaca sobre el Río Tárcoles, con la trágica consecuencia de personas fallecidas, el accidente del puente sobre el Río Lagarto, también con consecuencias mortales y daños estructurales al puente, entre otros más que se pueden citar.²

Con cada uno de estos colapsos de puentes, se provocan pérdidas económicas y humanas para el país y se arriesga la vida de miles de personas, además del impacto económico que se genera al detener o retrasar actividades productivas que utilizan dicha infraestructura. Este problema se da debido a la falta de inversión en el mantenimiento de las estructuras y el desarrollo y puesta en marcha de sistemas de monitorización que permitan conocer el estado del puente periódicamente.

Para la monitorización del estado estructural de un puente, se pueden monitorizar muchas variables siendo las más significativas 3 variables de desempeño, las cuales son: deflexión, tensión superficial y desplazamiento de las juntas. En la actualidad existen diferentes sensores que miden estas variables, pero en el país se desconoce su funcionamiento y por ende, su forma de instalación, de calibración y el tipo de mantenimiento que se les debe dar para que se puedan utilizar en conjunto con una red inalámbrica de sensores.

Sin la previa investigación sobre estos sensores, no se pueden instalar sistemas de monitorización y continuaría el problema existente con los puentes del país, donde el mantenimiento es casi nulo. En la mayoría de los casos, hasta que el puente no colapse o las condiciones de éste estén notablemente deterioradas, no se le realiza ningún trabajo.

Así, la única forma de conocer el estado real de un puente, es realizándole una monitorización constante pero el país carece de un protocolo de instalación, calibración y mantenimiento de los sensores para la monitorización estructural de éstos mediante el uso de redes inalámbricas de sensores.

Los beneficios para el país derivados del desarrollo de este proyecto son tanto de carácter económico como de seguridad, con los que se puede:

- Conocer el estado real de los puentes en el país.
- Evitar accidentes grandes que puedan significarle al país muchas vidas humanas.
- Conocer cuándo darle un mantenimiento preventivo a los puentes para evitar daños mayores que signifiquen un gasto económico mucho mayor para el país.
- Seguridad a la población de que la infraestructura vial de los puentes se encuentra en buen estado.

1.2 Solución seleccionada

Se pretende con este proyecto el desarrollo y puesta en marcha de una red inalámbrica de sensores que permita recolectar, procesar y almacenar resultados del desempeño del puente. Para ello se definieron tres variables a medir para las cuales se describen los sensores utilizados.

Para la medición de la tensión superficial en el concreto del puente, se utilizó el sensor ST350 de la empresa Campbell Scientific, el cual es un transductor de tensión utilizado en la medición del estrés provocado por algún evento en las estructuras de concreto.

En la medición de la deflexión presente en el puente debido al peso de los autos que transitan en él, se seleccionó el sensor SR50A-L de Campbell Scientific, el cual es un sensor ultrasónico para medir la distancia desde un punto y cuenta con una precisión de $\pm 0.25\text{mm}$.

Para cada uno de estos sensores se diseñó un circuito acondicionador de señales (CAS), el cual se conecta con el nodo de medición, para el cual se utiliza la tecnología de los nodos Waspote del fabricante español libellium, el cual se explicará más adelante. Estos circuitos acondicionan la señal para que pueda ser procesada por el nodo de medición dentro de los rangos eléctricos permitidos.

Para la transmisión inalámbrica de la información de cada sensor, se conecta a cada uno de los módulos Waspote, un módulo XBee que se utilizan para la comunicación en bandas de frecuencia libre de 2.4 GHz. Con esto, se crea una red de sensores que se comunican a un nodo central, el cual recibe la información y la envía a una PC. En el computador, se procesa la información y se envía por FTP a un servidor central en el cual se puede visualizar esta información y procesarla según las necesidades de post procesamiento del usuario final.

Para la comprobación del correcto funcionamiento del sistema, se comparan los datos obtenidos por medio de la red inalámbrica de sensores con los datos los datos medidos con el Datalogger CR1000 en condiciones exactas. Para la comparación, se utiliza la prueba T pareada de promedios, la cual es una prueba estadística para determinar las diferencias entre las medias de dos muestras diferentes, a fin de determinar una significancia estadística.

En los capítulos siguientes, se explicarán con más detalle todos los aspectos referentes al proyecto, tanto los aspectos teóricos como los referentes a la solución del problema. Así, en el capítulo 5, se explica más detalladamente la implementación, tanto a nivel de hardware como de software. Luego, en el capítulo 6, se pueden observar todos los resultados obtenidos en las pruebas realizadas para comprobar el funcionamiento del sistema y en el capítulo 7, se enlistan todas las conclusiones obtenidas con el desarrollo del proyecto y las recomendaciones para la mejora de éste.

Capítulo 2. Meta y objetivos

2.1 Meta

Contar con una red inalámbrica de sensores, capaz de medir de manera efectiva variables de desempeño estructural de un puente de concreto.

2.2 Objetivo general

Desarrollar un prototipo funcional de una red inalámbrica de sensores para la medición del desempeño estructural de un puente.

2.3 Objetivos específicos

1. Establecer la arquitectura de la red inalámbrica de sensores que permita la recolección adecuada de los datos de desempeño estructural de un puente, mediante el análisis de aplicaciones similares en el área.
2. Diseñar los circuitos acondicionadores de señal que garanticen el acople adecuado de los sensores de medición de desempeño estructural a los nodos de medición de la red inalámbrica de sensores.
3. Desarrollar una propuesta del protocolo para la instalación, calibración y mantenimiento de los sensores para la monitorización estructural de puentes que debe utilizar el proyecto eBridge.

Capítulo 3. Marco teórico

3.1 Descripción del sistema

El sistema desarrollado para este proyecto, cuenta con varias etapas, las cuales se pueden observar en la figura 1.

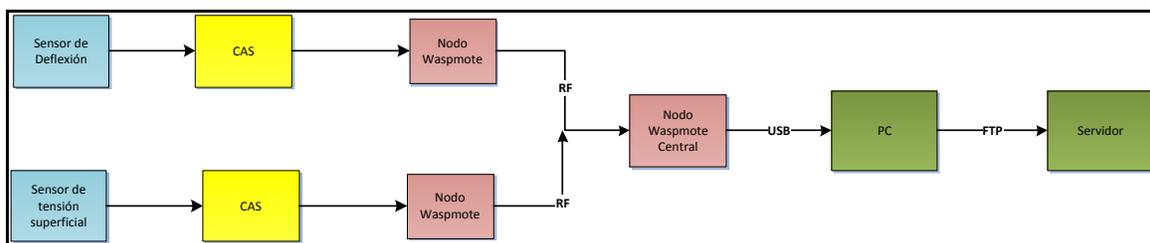


Figura 1. Etapas del sistema de caracterización estructural de puentes

La primera etapa del proyecto consiste en los sensores de medición de deflexión, tensión superficial y distancia entre juntas. Para estos sensores, se debe obtener la curva característica, además de su forma de instalación y calibración. Luego, estos sensores deben acoplarse a un circuito acondicionador de señales (CAS), el cual se encarga de transformar la tensión entregada por cada sensor, a un rango de tensiones que pueda ser legible por la placa Wasp mote, que representa la etapa 3 del proyecto.

La salida de cada CAS, se conecta a un nodo Wasp mote, el cual lee la información y la envía de forma inalámbrica a un nodo central, que toma la información y la envía por USB a una PC. En el computador, se lee la información del USB y se analiza, transformando el dato a la unidad de medida correspondiente a cada sensor; por ejemplo, el dato tomado del sensor de deflexión, se analiza y se pasa de tensión a centímetros. Una vez analizada la información, se almacena en un archivo y se envía por el protocolo FTP a un servidor.

3.2 Variables de desempeño estructural utilizadas

3.2.1 Deflexión

La deflexión o llamada también flexión mecánica, es la deformación que sufren los elementos estructurales alargados como las vigas, en dirección perpendicular a su eje longitudinal al aplicarle una carga. En la siguiente imagen se puede observar este fenómeno en una viga.[11]

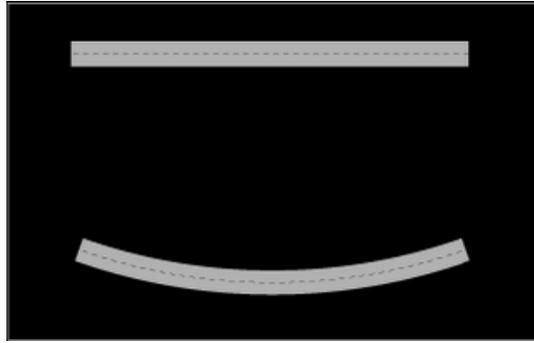


Figura 2. Viga Deflectada [1]

Para medir este fenómeno que ocurre en los puentes debido a la carga aplicada por los autos que transitan sobre él, se utilizan diferentes tipos de sensores, en el caso del proyecto eBridge se seleccionó el sensor ultrasónico SR50A-L de Campbell Scientific. Este sensor mide la distancia desde el sensor hasta un blanco, o punto de referencia, enviando pulsos ultrasónicos a 50kHz y escuchando los ecos que son reflejados; por lo que el blanco debe ser un buen reflector de sonido. La distancia se calcula dependiendo del tiempo que le tome regresar al eco al sensor. Basa su medición realizando varias mediciones a las que les aplica un algoritmo para mejorar la confiabilidad de los datos. En la figura 2, se puede observar el sensor.



Figura 3. Sensor SR50A [6]

Para su funcionamiento, requiere una entrada de 9 a 18 Vdc y cuenta con un rango de medición de 0.5 a 10 m con una resolución de 0.25mm. Además, tiene un tamaño de 10.1 cm de largo y 7.6 cm de diámetro, con un peso de 0.4 kg. [6]

Debido a que la velocidad del sonido varía dependiendo de la temperatura, se requiere una medida de temperatura adicional para poder realizar la compensación en la medición. Esta fórmula es la siguiente:

$$Dis\ tan\ cia = Lectura_{SR50A} \sqrt{\frac{Temperatura(Kelvin)}{273.15}} \quad (3.2.1.1)$$

Este sensor cuenta con tres tipos de salidas: SDI-12, RS-232 y RS-485. De fábrica, viene configurado para que tenga una salida en formato SDI-12 pero moviendo un grupo de 3 puentes dentro de él, se puede variar el formato de salida a cualquiera de los otros 2 mencionados anteriormente.

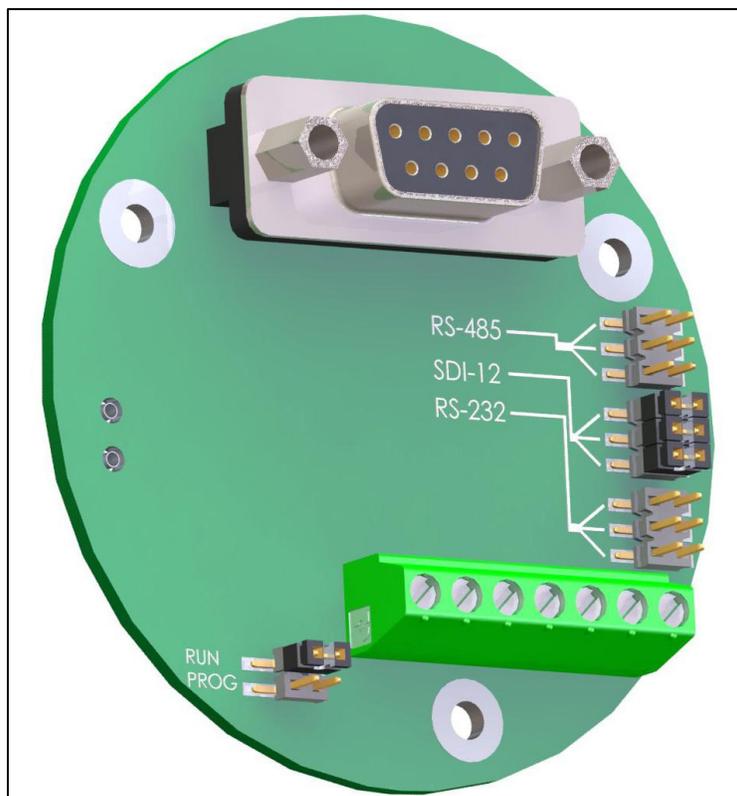


Figura 4. Puentes internos para cambiar de formato de salida [6]

El SR50A, puede recoger respuesta del blanco en un campo de 30° o menos. Esto es que si el blanco se encuentra fuera de un ángulo de 30° del centro del sensor, éste no logrará recibir la respuesta y no habrá medición; por lo que cualquier objeto que pueda reflejar el sonido, debe encontrarse fuera de este campo de vista. En el caso de que el blanco sea movable, el sensor va a rechazar cualquier medición si la distancia entre cada una varía a una velocidad de 4cm/s o más. [6]

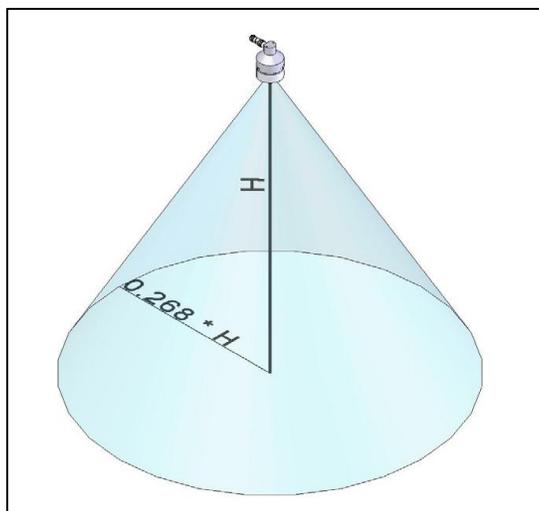


Figura 5. Ángulo de trabajo del SR50A [6]

El tiempo típico en que se logra completar una medición es de 1 segundo. En el caso de los modos seriales (RS-232 y RS-485), se logrará este tiempo para de 9600 bps en adelante. En el caso de SDI-12, puede que se exceda el minuto debido al largo tiempo de comunicación asociado a los 1200 bps.

El sensor SR50A, cuenta con un dato de salida que indica la calidad de la medición, así el usuario puede configurar el sensor para que obtenga este dato o no. No es necesario hacer uso de este dato de salida pero puede proveer información adicional sobre la confiabilidad de las mediciones realizadas.

3.2.2 Mediciones de tensión superficial en el material de la viga

Al aplicarle una carga a un material como por ejemplo el concreto, se produce una deformación en el material debido a la tensión generada por la carga. Dependiendo del estado del material, éste se puede deformar más o menos, de acuerdo a la carga aplicada. Así, si el material está en buen estado, va a tener menos deformación que un material que se encuentre en mal estado o deteriorado. Para poder medir esta tensión ocasionada en las vigas de un puente, se utilizó el sensor ST350 de Campbell Scientific.

El sensor ST350 es un transductor de tensión utilizado típicamente en eventos que provoquen algún tipo de estrés en la estructura de un puente o edificio. Se asume que durante la secuencia en que el sensor está tomando las muestras, no existen cambios de temperatura. Esto debido a que el sensor se coloca de tal forma que tome la misma deformación de la estructura. Por lo tanto, si se da un incremento o decremento de temperatura significativos, provocará que la estructura se expanda

o contraiga, lo cual será registrado por el sensor. Así, si el sensor se colocará por un período largo de tiempo, se deberá reiniciar a cero periódicamente. [7]

Especificaciones generales [7]

Calibre de longitud efectiva: 76.2 mm

Tamaño: 111mm x 32mm x 13mm

Longitud de cable: estándar son 3m, pero se puede utilizar cualquier longitud.

Material: aluminio.

Rango de temperatura: -50°C a 85°C.

Entorno: cobertura protectora y resistente al agua.

Se recomienda una tensión de excitación entre 2.5 y 5V DC.

Está formado por un puente Wheatstone completo con 4 resistencias de 350Ω. El circuito se puede observar en la figura 6.

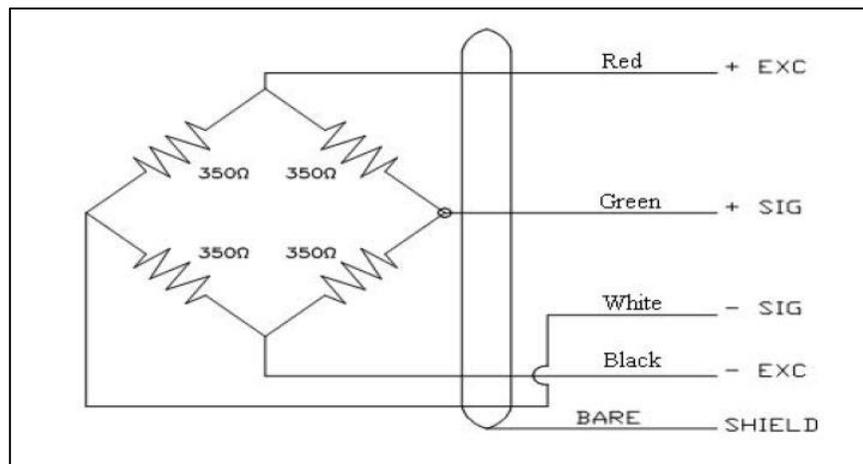


Figura 6. Diagrama eléctrico del sensor ST350 de Campbell Scientific. [7]

La salida del circuito se define de la siguiente manera:

$$V_{out} = (+Sig) - (-Sig) \quad (3.2.2.1)$$

Si V_{out} es positivo, hay tensión en la superficie.

Si V_{out} es negativo, hay compresión en la superficie.

3.2.3 Circuito Acondicionador de Señales (CAS) [17]

Los circuitos acondicionadores de señales (CAS), son los que reciben como entrada una señal proveniente de un transductor y la convierte en una señal que se encuentra en el ámbito de trabajo requerido por el circuito que procesará esta señal.

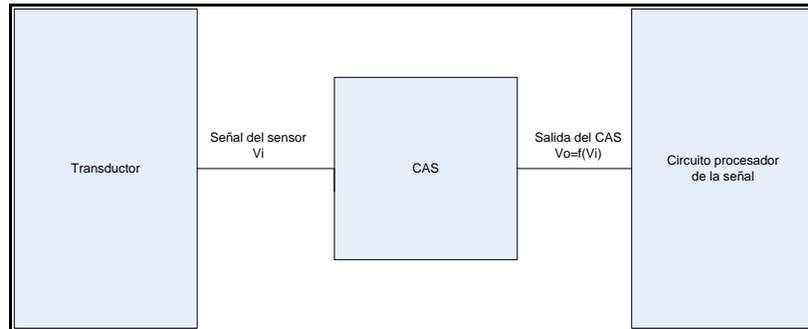


Figura 7. Esquema general de un CAS [17]

Procedimiento para el diseño de un CAS [17]

Para el diseño de un CAS, cualquiera que sea el circuito elegido, se deben seguir varios pasos, los cuales se explicarán a continuación:

1. Definición del ámbito de medición de la variable: Se debe definir el máximo y mínimo de la variable a medir.
2. Medición de la respuesta del transductor: Realizar mediciones para conocer el ámbito de salida que proporciona el transductor.
3. Definición del ámbito de salida requerido: Se debe definir el valor máximo y mínimo requerido por el circuito procesador de la señal.
4. Determinación de la relación entre la salida y la entrada del CAS: Se debe obtener la relación entre la salida y la entrada del CAS; para lograr esto, es de mucha ayuda utilizar un gráfico para observar mejor la relación.
5. Determinación de los parámetros del CAS: Se determinan los parámetros como ganancia, relaciones de resistencias, etc., para el tipo de circuito elegido.

Tipos de CAS [17]

Esquema Básico de un CAS

En la figura 8, se puede observar el esquema básico de un CAS.

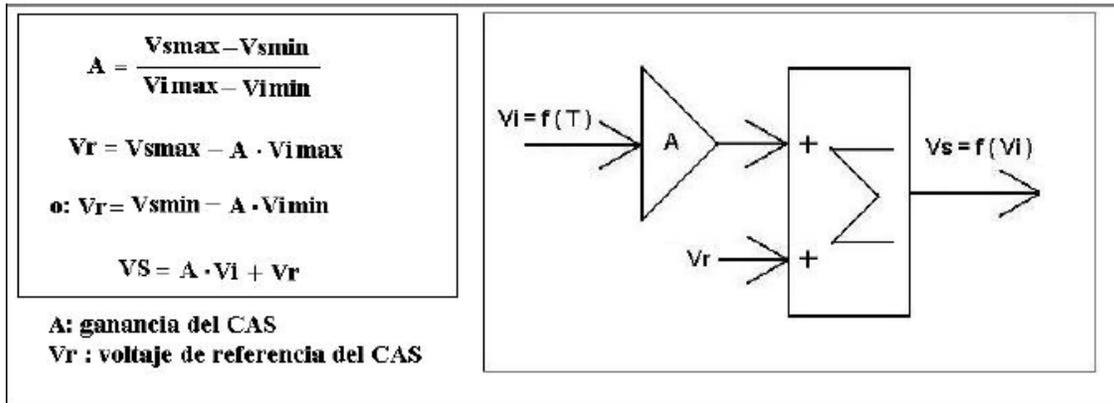


Figura 8. Esquema básico de un CAS [17]

En el esquema mostrado en la figura anterior, la señal proveniente del transductor, pasa primero por un circuito amplificador para luego, entrar a un circuito sumador en donde se le sumará una tensión de referencia para así lograr los valores requeridos a la salida.

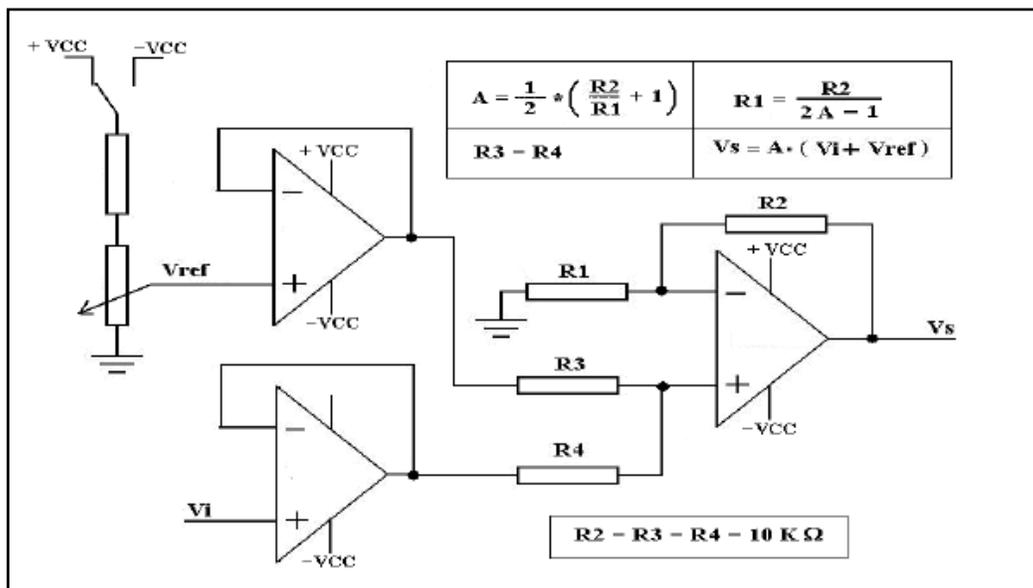


Figura 9. Circuito básico para un CAS [17]

Esquema alternativo de un CAS

En la figura 10, se puede observar el esquema alternativo de un CAS.

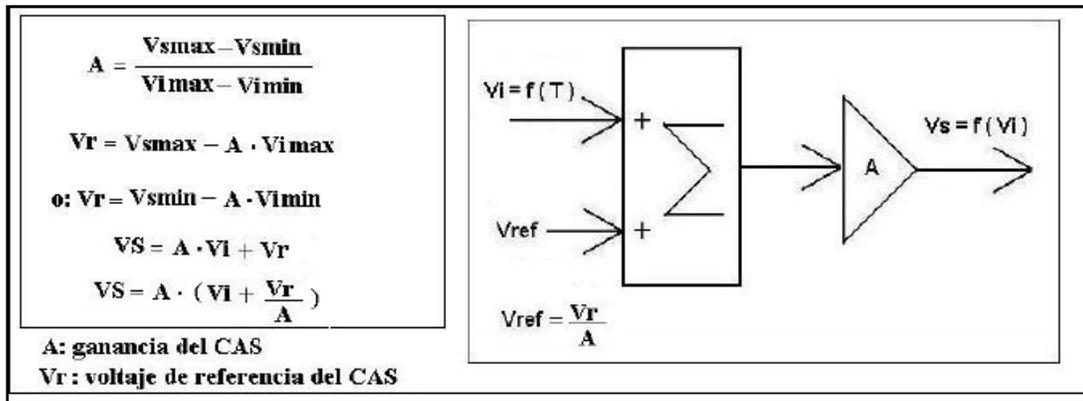


Figura 10. Esquema alternativo de un CAS [17]

En el esquema mostrado en la figura anterior, la señal proveniente del transductor, pasa primero por un circuito sumador, en donde se le sumará una tensión de referencia para lograr los valores requeridos para la salida y luego, entrar a un circuito amplificador.

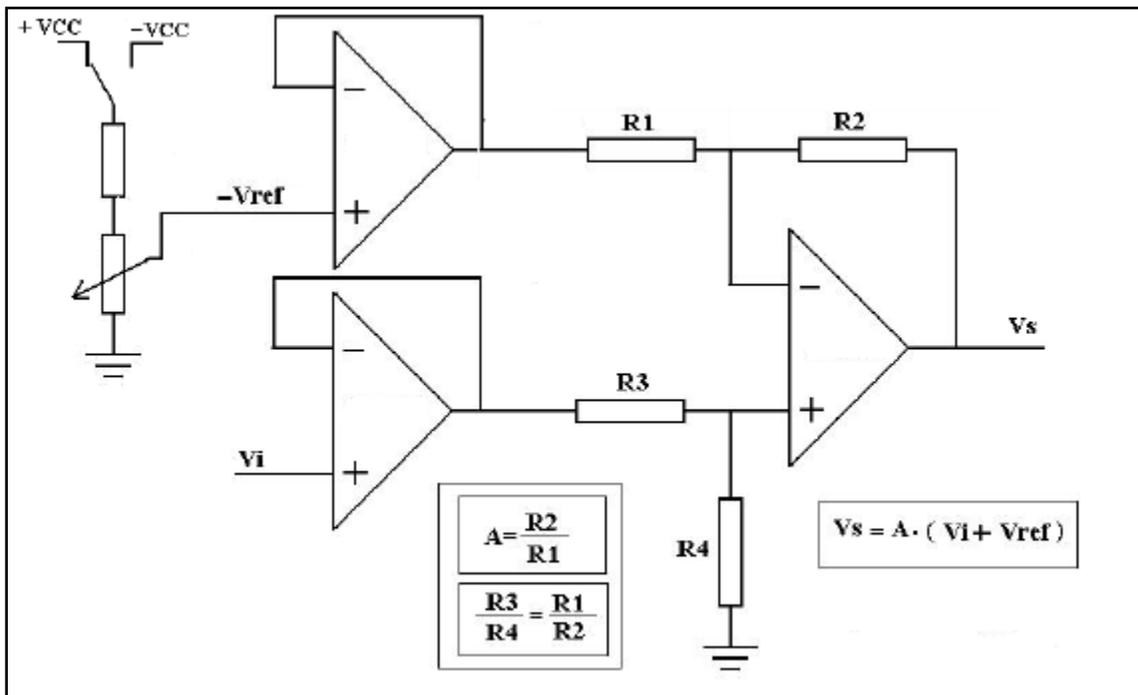


Figura 11. Circuito alternativo para un CAS [17]

3.2.4 Wasmote [13]

Los módulos Wasmote fueron diseñados y son distribuidos por la empresa Libellium. Esta empresa española diseña y fabrica tecnología de hardware para la implementación de redes inalámbricas de sensores para que los desarrolladores puedan llevar a cabo soluciones fiables y rápidas.

Este sistema embebido en específico es un dispositivo sensorial de bajo consumo para la creación de redes inalámbricas de sensores que pueden integrar más de 50 sensores diferentes. Se basa en una arquitectura modular y en él se adaptan solo los módulos necesarios, los cuales se pueden cambiar o ampliar.



Figura 12. Módulo Wasmote [13]

Los módulos que se pueden integrar a Wasmote son:

- Almacenamiento: SD Cards.
- GPS.
- Sensores.
- GSM o GPRS.
- XBEE.

Sus especificaciones generales son:

- Procesador ATmega1281.
- Frecuencia de 8Mhz.
- SRAM de 8KB.

- EEPROM de 4KB.
- FLASH de 128KB.
- SD Card de 2GB.
- Peso de 20gr.
- Dimensiones de 73.5 x 51 x 13 mm.
- Rango de temperatura de -20°C a 65°C.

Cuentan con una batería la cual puede ser recargada mediante conexión USB o un módulo de tensión por placa solar que puede ser adaptado fácilmente. Además, puede comunicarse con otros dispositivos externos mediante sus puertos de entrada/salida. Para esto, posee 7 entradas analógicas, las cuales están conectadas directamente al microcontrolador que utiliza un ADC de 10 bits y su valor máximo de tensión de entrada es de 3.3V que corresponde con el valor de alimentación general del microcontrolador.

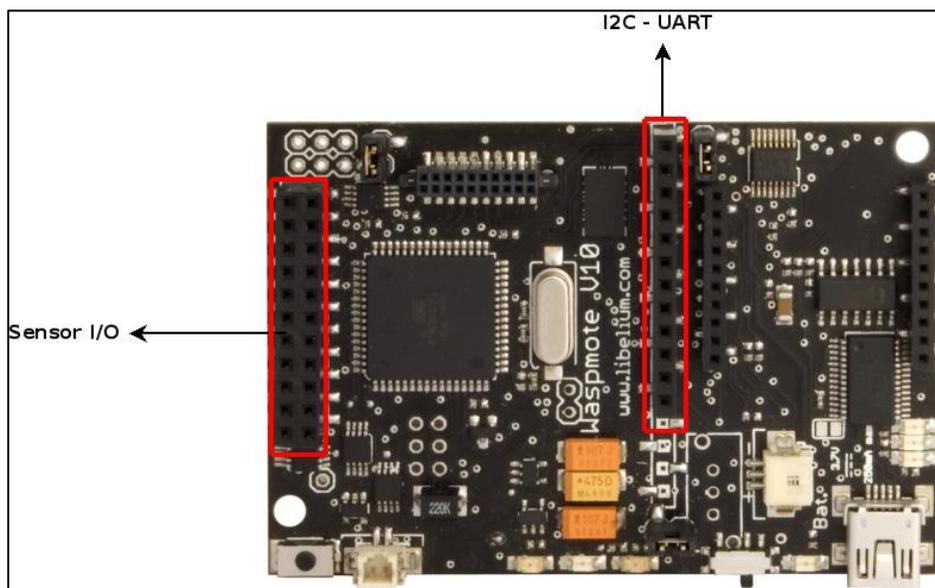


Figura 13. Conectores I/O en Waspote [13]

También, posee 8 pines digitales que pueden ser configurables como de entrada o de salida. Los valores de tensión digitales son:

- 0V para el 0 lógico.
- 3.3V para el 1 lógico.

El pin digital 1 también se puede utilizar como salida PWM que tiene una resolución de 8 bits. Para comunicarse vía serial, el waspote cuenta con 6 puertos serie, uno que está conectado al módulo XBee y el otro al puerto USB.

Luego existe un multiplexor de 4 canales; 1 canal para el módulo GPRS, otro al GPS y los últimos 2 quedan accesibles al usuario.

Además, en Waspote también cuenta con el bus de comunicación I2C, donde se conectan en paralelo el acelerómetro, el reloj de tiempo real (RTC) y el potenciómetro que configura el nivel de threshold de alarma por batería baja. En todos los casos, el microcontrolador es el maestro y los dispositivos son los esclavos.

En cuanto al puerto SPI, se encuentra conectado con la tarjeta micro SD y por último, está el puerto USB, el cual se puede comunicar con cualquier PC o dispositivo compatible. Con este puerto, se carga el programa al microcontrolador y se comunican los datos durante la ejecución del programa.

Para las indicaciones visuales, Waspote cuenta con 5 LED's, los cuales se pueden observar en la figura 14.

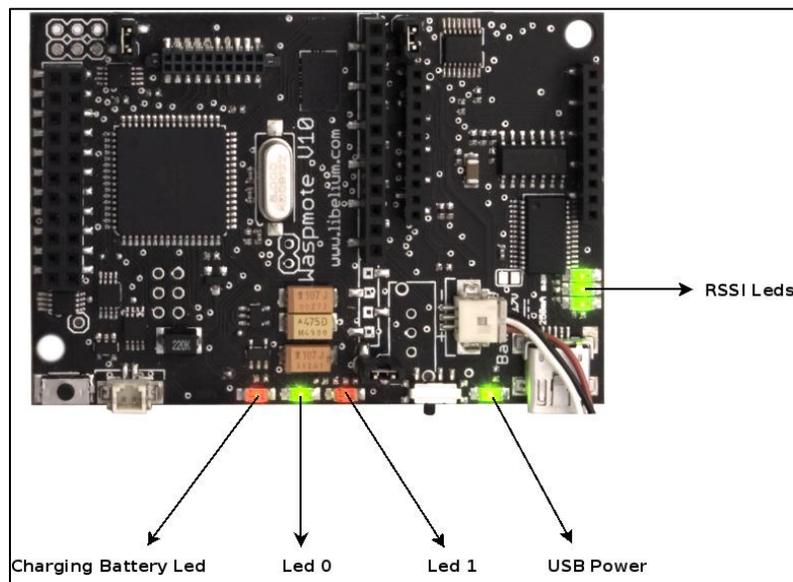


Figura 14. Indicadores LED's en Waspote [13]

Los LED's 0 y 1, pueden ser programados por el usuario en el código del programa.

En cuanto a las interrupciones, trabaja con 2 tipos: síncronas y asíncronas. Las primeras, son las que se programan mediante los timers, las cuales pueden ser periódicas (se especifica concretamente el momento) y las relativas (se programan según el momento actual). El segundo tipo de interrupciones, se pueden dar por sensores, batería baja, acelerómetro, módulo GPRS.

El Waspote cuenta con 4 modos de funcionamiento:

- On
- Sleep
- Deep Sleep
- Hibernate

A este sistema embebido se le pueden adaptar muchas otras funciones al adaptarle diferentes tipos de placas de sensores. Cuenta con muchas más especificaciones que pueden ser vistas en la página de Libelium, es un sistema muy útil y sencillo de usar, que utiliza el software Arduino para su programación.

3.2.5 Protocolo FTP

El FTP (File Transfer Protocol) es un protocolo de transferencia de archivos entre sistemas conectados a una red TCP basado en la arquitectura de cliente-servidor; esto quiere decir, que desde una computadora cliente, nos podemos comunicar con el servidor para subir o bajar archivos de éste con solo autenticarnos con un login y password.

El protocolo FTP nace en 1969 cuando unos investigadores del Instituto Tecnológico de Massachusetts presentaron una propuesta para el primer “Protocolo para la transmisión de archivos en internet, el cual era basado en el sistema de correo electrónico pero fue la base para el desarrollo posterior del FTP. [9]

En la siguiente figura se puede observar el modelo de este protocolo.

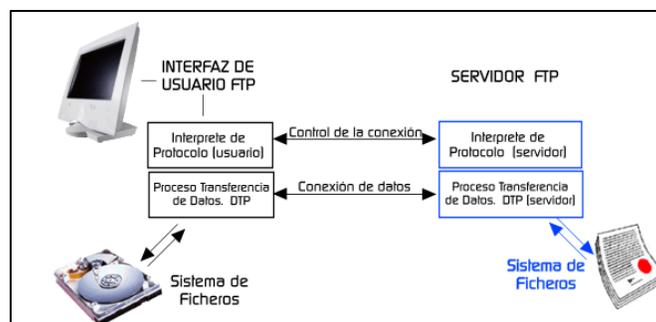


Figura 15. Protocolo FTP [3]

El intérprete de protocolo en el usuario, inicia la conexión con el servidor y envía comandos FTP y espera la respuesta del Intérprete de Protocolo en el servidor. Una vez que se tenga la respuesta y se especifiquen los parámetros para la conexión, se puede iniciar con la transferencia de archivos en la capa de Proceso de Transferencia de Datos.

Tanto el Intérprete de protocolo con el proceso de transferencia de datos, son bidireccionales, por lo tanto, se pueden usar tanto para recibir como para enviar información.

Capítulo 4. Proceso metodológico

4.1 Síntesis del problema y requerimientos

El problema encontrado y por el cual se desarrolla este proyecto, salió de la necesidad latente en el país de mejorar la infraestructura vial. En este caso, se tomó el problema existente en la mayoría de los puentes del país, los cuales carecen de mantenimiento y sobre todo de control sobre el verdadero estado de salud de los mismos.

Los múltiples accidentes que han sucedido en los últimos años en diferentes puentes del país, abonado a los muchos de éstos que se encuentran deshabilitados porque ya representan un peligro visible para la población, hace que surja la necesidad de monitorizarlos para conocer el estado real y así anticipar cualquier accidente o incidente de seguridad.

Teniendo lo anterior en cuenta, se puso como meta encontrar la solución más viable en cuanto a la arquitectura de la red y la caracterización de los sensores existentes en el mercado y definidos por la coordinación del proyecto eBridge, para lograr conocer su funcionamiento y así, diseñar unos circuitos que logren obtener mediciones confiables de cada sensor.

4.2 Obtención y análisis de información

Para la obtención de la información relevante para el desarrollo del proyecto, se utilizó la investigación en internet acerca de la importancia de la monitorización de puentes y de los métodos utilizados para éste.

Se analizó la información de varias formas, una de ellas fue la viabilidad y relación de la información a las condiciones existentes en el país, y el otro método de análisis utilizado, fue el de costos, ya que cualquier solución que se pudiera plantear, debía estar acorde a la situación del país por lo que el tema económico es de suma importancia.

Otro factor importante, es que se debían encontrar equipos y sensores que no necesitaran mantenimiento ni calibraciones constantes, debido a que si se van a instalar debajo de un puente, estarlos poniendo y quitando, significaría un costo elevado para el país. Además, ser capaces de soportar estar al aire libre sin preocuparnos de su oxidación o cualquier otro daño ocasionados por humedad, cambios de temperatura, entre otros.

4.3 Evaluación de las alternativas y síntesis de la solución

Para el planteamiento de las alternativas para la solución se utilizó la investigación en internet y las consultas con el profesor asesor del proyecto, ya que la coordinación del proyecto había elegido algunos equipos que debían ser utilizados como los Wasmote de Libellium y el sensor de tensión superficial.

Las alternativas de solución debían leer los datos del sensor y convertirlos a un valor que pudiera ser leído por el Wasmote para luego que éste lo enviara a la PC y de ahí se almacenaran los datos en un servidor. Para verificar que las lecturas realizadas de cada sensor, fueran veraces, se utilizaron las simulaciones y las pruebas controladas de laboratorio. Además, de la comparación de los resultados obtenidos con los desplegados por el datalogger, el cual ya está programado para leer ese tipo de sensores.

Con las simulaciones, se logró verificar que las alternativas eran viables con las salidas de los sensores y los voltajes de entrada necesarios para el Wasmote. Luego, con las pruebas en el laboratorio, se fueron modificando los circuitos debido a que en las simulaciones se suponen que los componentes utilizados son ideales, pero en la realidad no lo son y en algunos casos pueden provocar variaciones importantes en los resultados obtenidos.

Además, gracias a estas pruebas, se pudieron descartar algunas de las alternativas planteadas y así poder llegar a la solución final. En algunos casos se descartó alguna solución por su pobre desempeño o porque la facilidad de implementación superaba a alguna otra de las alternativas.

4.4 Implementación de la solución

Para la implementación de la solución, lo primero era entender el funcionamiento del sensor y de su salida. En el caso del sensor SR50A para la medición de la deflexión, éste cuenta con 3 formatos diferentes para la salida; por lo tanto, se debía analizar cuál era el formato más sencillo y viable a utilizar, tomando en cuenta los equipos y componentes con los que se contaba. Se eligió el formato serie RS-232, debida a que la mayoría de equipos y componentes están acondicionados para la transferencia de información con este tipo de formato.

La propuesta inicial consistía en conectar directamente el sensor al Wasmote pero debió ser modificada, ya que el lenguaje con el cual se deben programar los módulos Wasmote es poco robusto y no realizaba correctamente la lectura del sensor; por lo que se debió colocar una etapa intermedia entre el sensor y el Wasmote.

En cuanto al sensor de medición de tensión superficial ST350, existían 3 alternativas de medición, las cuales comprendían la utilización del Wasmote para

leer directamente la salida del sensor; la segunda utilizar un circuito acondicionador de señales y por último, la utilización de un PIC para su lectura, estas últimas dos como etapas intermedias entre el sensor y el Wasmote. Al evaluar cada una de estas alternativas, se eligió la alternativa del CAS por su facilidad de implementación y bajo costo; se descartó la primera opción porque el módulo no realizaba correctamente la medida de las dos entradas analógicas del sensor y su diferencia.

Los módulos Wasmote funcionan como una red inalámbrica, en donde cada módulo es un nodo de la red y estos nodos se comunican a un nodo central, el cual es Gateway que se conecta a la PC y le transmite la información recibida a ésta.

La información es recibida en formato RS-232 por la PC mediante un programa realizado en Python y en Linux. Este programa crea un archivo de Excel donde guardará la información procesada y la enviará por FTP a un servidor.

Para verificar la validez de los datos obtenidos con el sensor para la medición de la deflexión, se realizó un banco de pruebas en donde se realizaba la medición del sensor y se leía el dato en el Wasmote para luego compararlo con la medición realizada con un metro.

Para el sensor de tensión superficial, se comparó el valor medido por el CAS y leído en el Wasmote, con el valor entregado por el datalogger CR1000 de Campbell Scientific el cual está configurado para leer estos sensores. Se utilizó la prueba T pareada para comprobar la validez de los datos.

Con la red inalámbrica y el software realizado en python, se realizaron pruebas con un banco de datos conocidos para verificar que los datos se comunicaran correctamente dentro de la red, además de que fueran leídos correctamente en la PC y guardados en el archivo con extensión .csv.

4.5 Reevaluación y diseño

Para mejorar el sistema desarrollado, se deben buscar alternativas para diseñar las tensiones necesarias para alimentar los diferentes circuitos. Esto debido a que el sensor SR50A necesita una tensión de alimentación de 12V y el CAS para el sensor ST350, necesita una alimentación de 5V y -5V, debido a que este sensor genera tensiones de salidas tanto positivas como negativas.

Además, se debe reemplazar la PC por algún sistema embebido, el cual soporte la plataforma Linux y así, este sistema pueda ser también instalado en el puente y sea capaz de enviar la información vía FTP al servidor.

Los módulos Waspote son muy útiles para la creación de redes de sensores, debido a su fácil programación y uso, además de que cuentan con el hardware integrado para la conexión de un panel solar para la recarga de su batería. Por lo tanto, una mejora al sistema, sería incluir en la programación de estos módulos, el software para la lectura del estado de la batería y en caso de que se encuentre baja, poner en funcionamiento el panel solar y así tener un sistema que no necesite mantenimiento constante, solo en casos de que el sistema falle o se dañe, lo cual es poco probable.

Otro aspecto que se debe incluir en este proyecto, es la visualización de los datos en el servidor. Se debe crear algún programa que permita ver gráficamente los datos obtenidos para así comprobar el estado del puente y poder conocer fácilmente si el puente requiere algún tipo de mantenimiento.

El diseño de unas cajas especiales que eviten el contacto, tanto de los sensores como de los módulos usados, con el ambiente, es necesario debido a que así se evita la corrosión y el daño por la humedad, agua, entre otros. Con esto se lograría aumentar significativamente la vida útil del sistema y un menor mantenimiento.

Capítulo 5. Descripción detallada de la solución

En este capítulo se explicará detalladamente la solución propuesta. En primera instancia, se hará un análisis de las soluciones propuestas para cada etapa del sistema y se explicará el por qué se eligió una de ellas y el por qué se descartaron las otras.

Luego, se explicará el funcionamiento y estructura de cada uno de los módulos diseñados, así como el criterio utilizado para la escogencia de éstos. Por último, se hará una descripción detallada de los diferentes programas de software desarrollados para lograr el debido funcionamiento de los diferentes módulos. Se presentarán los diagramas de flujo para cada rutina y los algoritmos utilizados.

En la siguiente figura, se muestra el diagrama de bloques que ilustra la topología de red escogida. En este caso se utilizó una arquitectura de red con topología estrella debido a que esta es la única que se puede implementar con los nodos Wasp mote con los que se contaba para el desarrollo del proyecto.

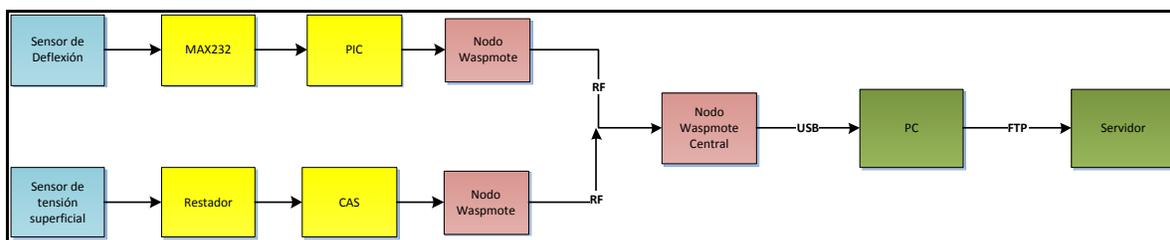


Figura 16. Topología de la red inalámbrica de sensores utilizada

5.1 Análisis de soluciones y selección final

A continuación, se explicarán las diferentes soluciones encontradas para cada módulo del sistema. Se analizará cada una de ellas y se verán sus ventajas y desventajas para así poder entender los criterios que llevaron a la selección de la solución final.

5.1.1 Soluciones sensor de tensión superficial ST350

Para la lectura de la medición de la tensión superficial con el sensor ST350, se plantearon 2 soluciones. Este sensor, como se vio en la sección 3.2.2 y la figura 6, está formado por un puente Wheatstone, en el cual la tensión de salida va a ser la diferencia entre la tensión medida en la señal “+Sig” y la señal “-Sig”.

Tomando en cuenta lo anterior, la primera solución propuesta fue la lectura directa del sensor por parte del Wasp mote. Para esto, se tomaría la alimentación para el sensor de los 5V con los que cuenta el módulo Wasp mote para la alimentación de

sensores. Además, se conectarían las dos salidas del ST350 con dos de las entradas analógicas de este equipo.

Así, se tomaría cada segundo, una medición de cada puerto analógico y se procesaría la medición hasta obtener la tensión de cada entrada y su diferencia; para luego, enviar el dato inalámbricamente de este nodo al nodo central.

Esta solución presentaba la ventaja de ser más económica que las demás, debido a que no se necesitaba una etapa intermedia entre el sensor y el Waspote y además, resultaba una solución sencilla; simplemente había que realizar un PCB que conectara las entradas del sensor con las entradas analógicas y la alimentación del módulo.

El problema con esta solución se vio en el momento de realizar las pruebas. Al conectar el sensor con el Waspote y realizar mediciones, éstas no variaban a la hora de agregar o quitarle peso a la estructura a la que estaba conectado el sensor. Lo anterior, puede ser causado por la velocidad de conversión y de muestreo del ADC del procesador ATmega1281. Este problema no se pudo solucionar debido a que el lenguaje de programación de los waspote, es poco flexible y no permite la modificación de estos parámetros; por lo tanto, esta solución fue descartada.

La segunda solución propuesta, fue incluir una etapa adicional entre el sensor y el Waspote. Esta etapa correspondería a un circuito restador junto con un circuito acondicionador de señales (CAS). En un inicio se diseñó el restador de tal forma que no tuviera ninguna ganancia y así poder ver a la salida el valor real de las variaciones en la tensión. Las salidas que tendría el circuito andarían entre los -20mV y los 20mV, dependiendo del sensor, ya que aunque cada uno de estos sensores funciona igual, sus curvas de calibración varían.

Al conectar el sensor al circuito restador, realizarle pruebas y compararlas con los valores que se medían con el mismo sensor en el datalogger CR1000, se pudo observar que estos valores no coincidían y andaban entre -7mV y 7mV, mientras que en el datalogger andaba entre los -3mV y los 3mV. Por lo tanto, se analizó el circuito y se pudo determinar que las tensiones de salida eran muy bajas y en ese caso, el ruido podía llegar a ser significativo y afectar la medición.

Para poder corregir el problema que el ruido podría estar afectando las mediciones, se le incluyó una ganancia de 100 al circuito restador. Con esta ganancia y analizando las tensiones de salida de las curvas de calibración de los sensores, se obtendrían valores de salida del restador en el rango de -2V y 2V. Al realizar pruebas en el laboratorio y compararlas con los valores medidos en el datalogger, se comprobó que coincidían los valores.

La siguiente etapa sería el diseño del CAS, para este se utilizó el esquema básico que se puede observar en las figuras 8 y 9. Este circuito tendría valores de entrada entre los -2V y los 2V y tensiones de salida entre 0V y 3.3V que son los valores de entrada en el Waspote.

Una vez listo el circuito, se conectó el sensor al restador y éste al CAS, para la salida conectarla al Waspote. A la vez, se conectó el ST350 al datalogger y de éste, salía una señal de control para que los dos equipos realizaran la medición al mismo tiempo. Se comprobó que los valores eran muy cercanos, por lo que los valores medidos eran correctos.

La ventaja de los circuitos restador y CAS, es que son sencillos de diseñar y además, muy económicos. La desventaja se da porque para el circuito restador, se necesitan tensiones de alimentación de -5V y 5V. La tensión de 5V se toma del Waspote pero la de -5V no, por lo que se necesita una fuente de alimentación adicional.

5.1.2 Soluciones sensor de deflexión SR50A

Para el sensor SR50A, se plantearon también dos soluciones. Este sensor ultrasónico, cuenta con 3 formatos de salida. Se eligió el formato RS-232, debido a que es compatible con la mayoría de equipos. Por lo tanto, se plantearon dos soluciones, la primera era conectar directamente el sensor al Waspote y la segunda, incluir una etapa adicional entre éstos dos.

Para la primera solución planteada, solo se debía cablear el sensor a las entradas de recepción y transmisión serie en el Waspote y utilizar una fuente de alimentación externa de 12V para el SR50A.

El problema en esta solución vino con la programación. Como se había comentado antes, el lenguaje de programación de los waspote es poco flexible por lo que las interrupciones se manejan realizando una sentencia condicional al inicio del programa principal. Esto provocó que si se entraba en la interrupción porque había un dato disponible en el puerto serie, leyera el dato o carácter y luego, siguiera con las demás instrucciones del programa y hasta que no terminara con éstas, no volvía a revisar si existía algún dato para realizar la interrupción.

Esto ocasionó que al leer la trama de datos del sensor, se perdieran algunos de éstos y por lo tanto no llegara completa. La ventaja de esta solución era que no se necesitaba una etapa extra entre el SR50A y el Waspote, pero la desventaja fue la pérdida de información, lo que al final que provocó que esta solución se descartara y se siguiera la propuesta con etapa intermedia.

Para la segunda solución, se incluyó un MAX232 y un PIC 18F4550, los cuales toman la alimentación del Waspote. El MAX232 se encarga de recibir la trama de datos del sensor en formato RS232 y transformarla a TTL que puede ser leída por el PIC. En éste último, se utiliza una interrupción por recepción en el puerto serie, para leer la trama de datos. Luego identifica los 4 caracteres que corresponden a la distancia medida y envía progresivamente los 4 caracteres por un puerto digital hacia el Waspote; el cual, lee éstos y forma el dato medido para enviarlo vía inalámbricamente al nodo central.

La ventaja de ésta solución, es su bajo costo económico y además, con el PIC se puede realizar el análisis de la trama e identificar los datos de interés y así, el trabajo del Waspote va a ser muy pequeño, lo que es importante porque la memoria de éste es muy pequeña y además, su lenguaje de programación muy básico. La desventaja tanto en ésta como en la primera solución, es que se necesita una fuente de alimentación adicional de 12V para el sensor.

5.1.3 Red inalámbrica de sensores

La coordinación del proyecto eligió la red de sensores Waspote para la implementación del sistema de monitorización, dado que se contaba con el hardware y las herramientas. Como se había comentado anteriormente, éstos cuentan con un módulo XBEE 802.15.4, el cual se encarga de la comunicación inalámbrica y a la vez, se comunica con el microcontrolador ATmega1281 mediante la UART0 a una velocidad de 38400bps. Para la transmisión, se utiliza la frecuencia de banda libre de 2.4GHz y un ancho de banda de 5 MHz, además de que cumple con el estándar IEEE 802.15.4 que define el nivel físico y de la capa MAC que es el de enlace.

Para la programación de éstos módulos, se tenían dos opciones, las cuales eran enviar los datos con el encabezado API o sin éste. API son una serie de librerías diseñadas para facilitar la programación de los módulos Waspote.

La estructura de trama API se muestra en la siguiente figura:

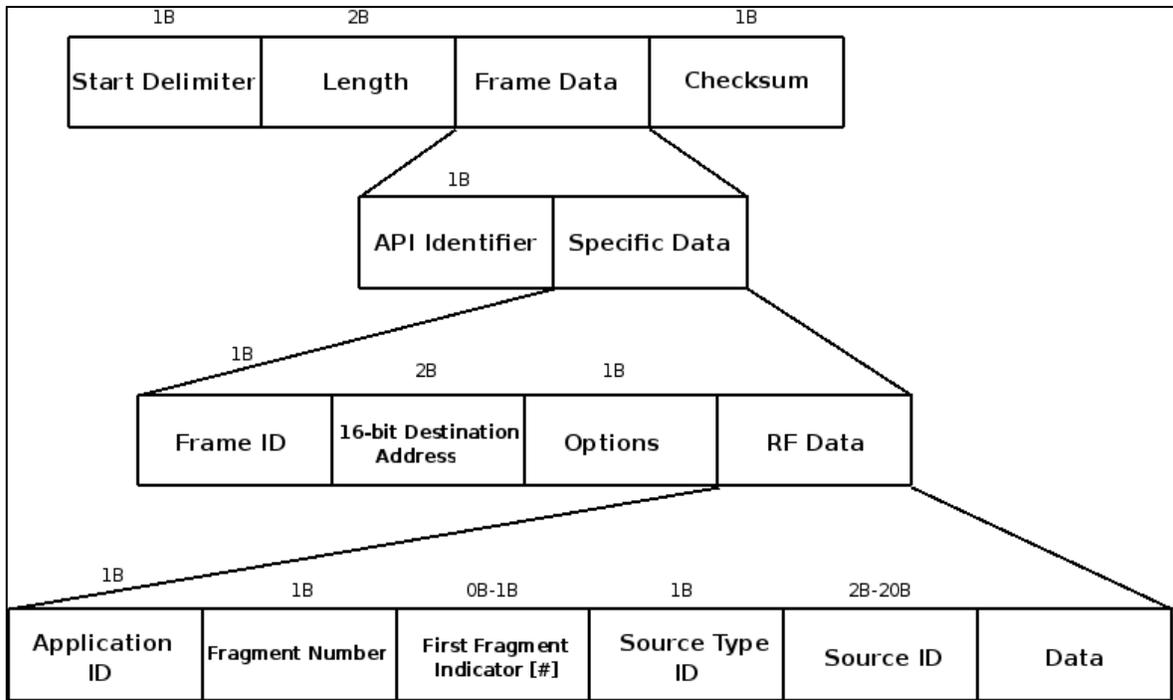


Figura 17. Estructura de trama API zigbee [12]

El encabezado API correspondería a la última fila de datos en la estructura mostrada en la figura anterior y sería lo que el usuario debe ingresar para poder enviar el dato. Como se puede observar también, esta trama es muy larga y la mayoría de información no es necesaria. La ventaja de esta opción, es que se puede determinar de qué nodo viene la información, pero su longitud representa una desventaja muy grande.

Cuando la trama se envía sin el encabezado API, lo único que se envía es el dato necesario. La desventaja de este método de envío es que no se puede determinar de cuál nodo viene la información, pero tiene la ventaja de que la trama es pequeña y con solo la información necesaria. Para contrarrestar el problema de no conocer el nodo origen, se creó un arreglo en donde se envía el identificador del nodo origen seguido del dato medido.

5.1.4 Recepción, procesamiento y envío de datos en la PC

Para la realización del software para la recepción, procesamiento y envío de datos en la PC, se eligió el lenguaje de programación Python por encima de muchos otros, debido a su robustez y fácil programación. Además, porque contiene librerías ya diseñadas para la programación de la lectura por puerto serie, la creación de archivos con formato .csv y en envío y recepción de archivos por FTP.

Además, se eligió el sistema operativo Linux por ser un software libre, sencillo de encontrar y porque el software con el cual se reconocen los módulos Waspote en la PC, no está disponible para todas las versiones de Windows.

5.2 Descripción del hardware

En esta sección, se explicará detalladamente el hardware de cada uno de los módulos que componen el sistema. Se incluirán los diagramas detallados de los circuitos y el por qué la elección de sus componentes.

Se iniciará con la explicación de los módulos para el sensor ST350; siguiendo con los módulos para el procesamiento del sensor SR50A y finalizando con la red inalámbrica de sensores.

5.2.1 Módulos de hardware para el sensor ST350

Para el sensor ST350, se crearon dos módulos de hardware. El primero consta de un circuito restador y el segundo, un circuito acondicionador de señales (CAS).

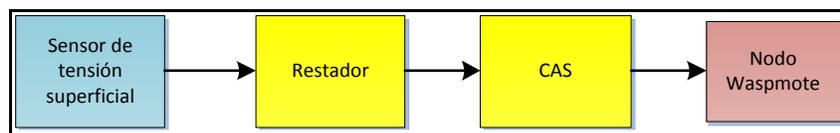


Figura 18. Módulos de hardware para el sensor ST350

5.2.1.1 Circuito Restador

Como se explicó en la sección 5.1.1, fue necesario el diseño de un circuito restador para realizar la diferencia entre las dos señales del sensor y así obtener la tensión de salida para la medición de la tensión superficial.

Para la construcción de este circuito, se utilizó un amplificador operacional LM324. El diseño básico de un restador con OPAM's, se muestra en la figura 19.

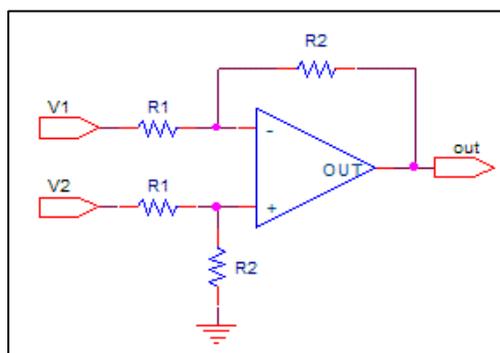


Figura 19. Esquema básico de un circuito restador con OPAM's [2]

El circuito mostrado en la figura anterior, resta a la entrada positiva del operacional, la entrada negativa y el resultado es amplificado con la ganancia. La ganancia se calcula de la siguiente forma:

$$A_v = \frac{R_2}{R_1} \quad (5.2.1.1)$$

Por lo tanto, la tensión de salida del circuito es:

$$V_{out} = \frac{R_2}{R_1} (V_+ - V_-) \quad (5.2.1.2)$$

Como se necesita una ganancia de 100 para obtener una tensión de salida entre -2V y 2V para una entrada entre los -20mV y los 20mV, valores tomados de las hojas de calibración de los sensores. Los valores de las resistencias serían los siguientes:

$$R_1 = 1.5k\Omega$$

$$R_2 = 150k\Omega$$

$$A_v = \frac{150k\Omega}{1.5k\Omega} = 100$$

En la figura 20, se puede observar el esquema del circuito diseñado.

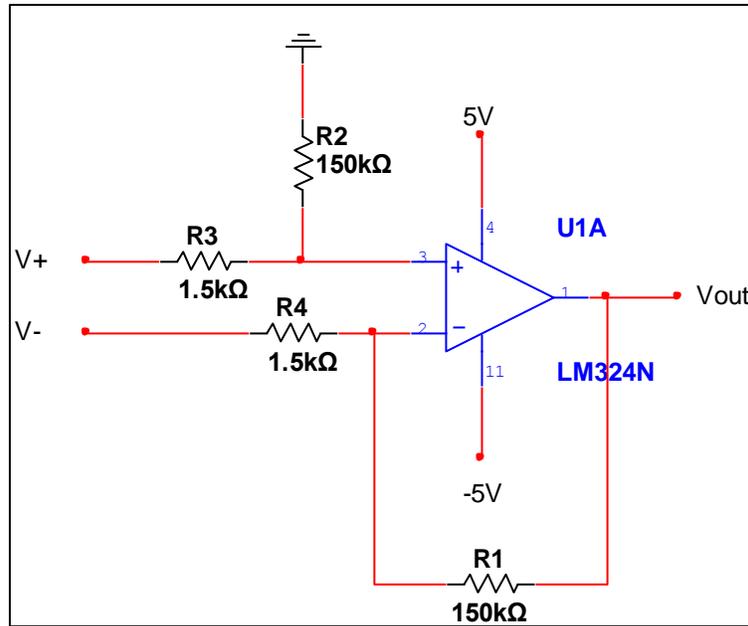


Figura 20. Circuito restador implementado para el sensor ST350

5.2.1.2 Circuito Acondicionador de Señales para sensor ST350

Para el diseño del circuito acondicionador de señales, se eligió el esquema básico explicado en el apartado 3.2.3 y que se observa en la figura 9. La entrada a este CAS, es la salida del circuito restador del apartado anterior, el cual tiene una salida entre -2V y 2V.

La finalidad del diseño de este CAS, es poder adaptar la señal medida en el sensor para que pueda ser leída por el Waspote, que maneja como entrada valores de tensión entre 0V y 3.3V.

En la figura 8, se pueden observar los cálculos necesarios para el diseño de este circuito acondicionador de señales. Los cuales se mostrarán a continuación:

Ganancia del sistema

$$A_v = \frac{V_{out_{max}} - V_{out_{min}}}{V_{in_{máx}} - V_{in_{min}}} \quad (5.2.1.2.1)$$

$$A_v = \frac{3.3V - 0V}{2V - -2V}$$

$$A_v = 0.825$$

Tensión de referencia

$$V_r = V_{out_{max}} - A_v V_{in_{max}} \quad (5.2.1.2.2)$$

$$V_r = 3.3V - 0.825 * 2V$$

$$V_r = 1.65V$$

Teniendo estos valores, la ecuación de la recta para este CAS sería:

$$V_{out} = A_v V_{in} + V_r \quad (5.2.1.2.3)$$

$$V_{out} = 0.825V_{in} + 1.65V$$

Conociendo la ganancia y la tensión de referencia del circuito, se procede a calcular los valores de sus componentes. Al igual que en el circuito restador, se utilizan los amplificadores operacionales LM324, con tensiones de alimentación de 5V y -5V. El valor de sus resistencias se calculó de la siguiente forma según la figura 9.

$$R_2 = R_3 = R_4 = 10k\Omega$$

$$R_1 = \frac{R_2}{2A_v - 1} \quad (5.2.1.2.4)$$

$$R_1 = \frac{10k\Omega}{2 * 0.825 - 1}$$

$$R_1 = 15.38k\Omega \approx 15k\Omega$$

Por lo tanto, el esquema del circuito queda de la siguiente forma:

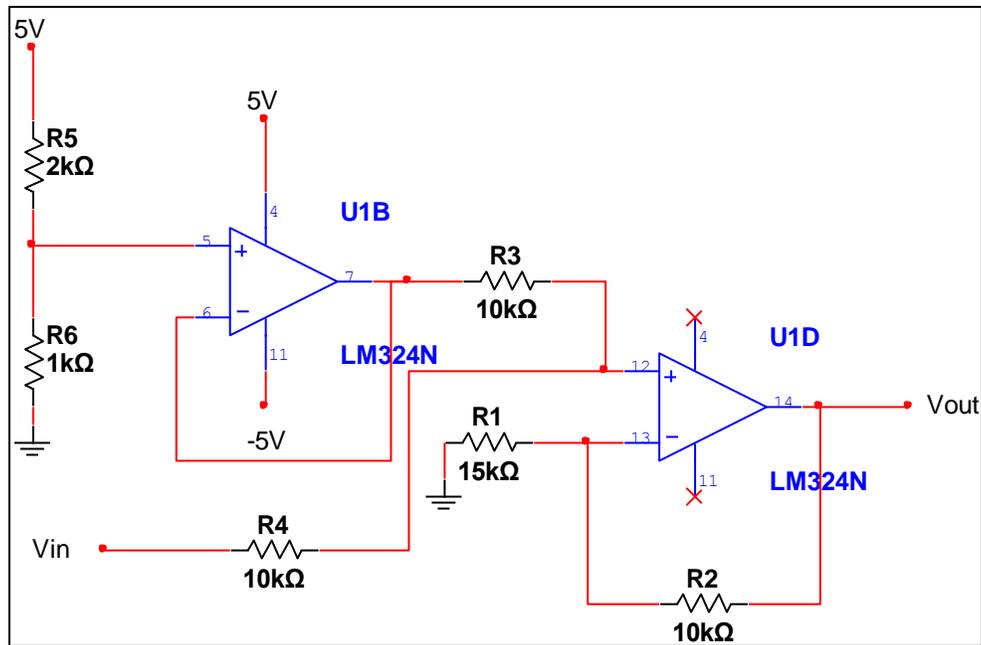


Figura 21. CAS implementado para el sensor ST350

Para lograr la tensión de referencia, se tomó el valor de R5 de 2kΩ y se calculó la resistencia R6 de la siguiente forma:

$$V_r = \frac{5V * R_6}{R_6 + R_5} \quad (5.2.1.2.5)$$

$$R_6 = 1k\Omega$$

El esquemático final para el diseño del PCB, queda de la siguiente forma.

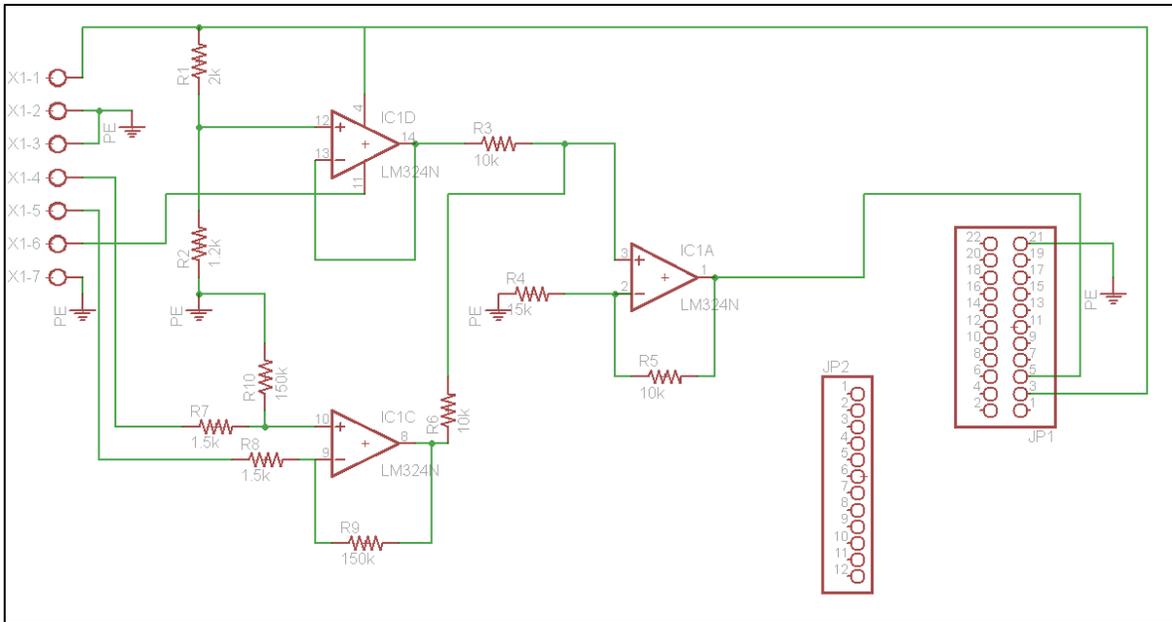


Figura 22. Esquemático del hardware diseñado para el sensor ST350

5.2.2 Módulo para el procesamiento de la señal del sensor SR50A

Para el procesamiento de la señal del sensor SR50A, se utilizó un PIC18F4550 y un MAX232, el cual se encarga de transformar la señal en formato RS232 entregada por el sensor, a TTL para poder ser leída por el PIC. Éste último, se encarga de tomar el dato proveniente del MAX232, la procesa y la envía por puerto digital al Waspnote. Para poder realizar lo anterior, se utiliza un cristal de 4MHz conectado a dos condensadores de 20pF y un sistema para el reset manual del PIC.

Para la elección de los valores de los capacitores para el MAX232, se tomó el esquema básico que se puede observar en la hoja de datos del componente.

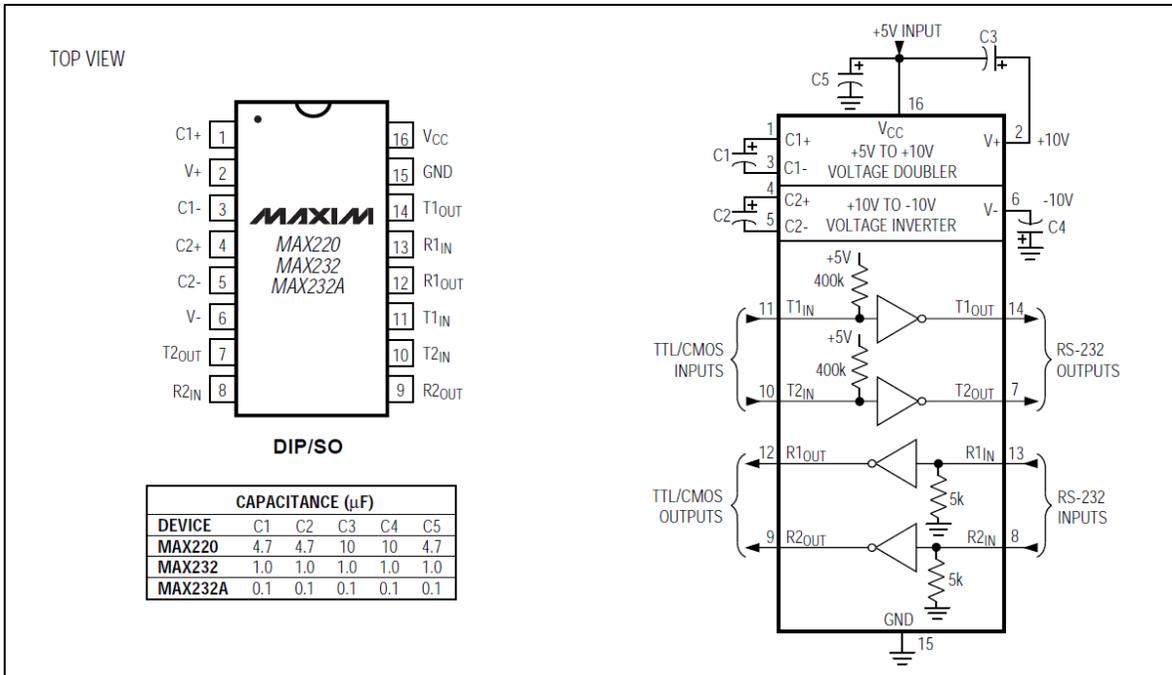


Figura 23. Esquema utilizado para el MAX232 [14]

Con base en la figura anterior, se eligieron todos los condensadores de $1\mu\text{F}$ y la alimentación se toma de los 5V entregados por el Waspote. Los pines elegidos para la conexión con el sensor y con el PIC son:

- Pin 11: patilla del transmisor del PIC.
- Pin 12: patilla del receptor del PIC.
- Pin 13: Transmisor del sensor.
- Pin 14: Receptor del sensor.

Para la alimentación del PIC, también se utilizan los 5V entregados por el Waspote. El esquemático final para el diseño de la PCB, se puede observar en la siguiente figura.

En las figuras 24 y 25, el componente X1 se utiliza para la conexión de las entradas del sensor; el componente JP1 y JP2 para poder montar el PCB en el Wasp mote, ya que coinciden con las entradas de este módulo. Además tiene el componente JP3, que se utiliza para la programación del PIC con el ICSP que es una forma de programación de PIC's diseñado por Microchip.

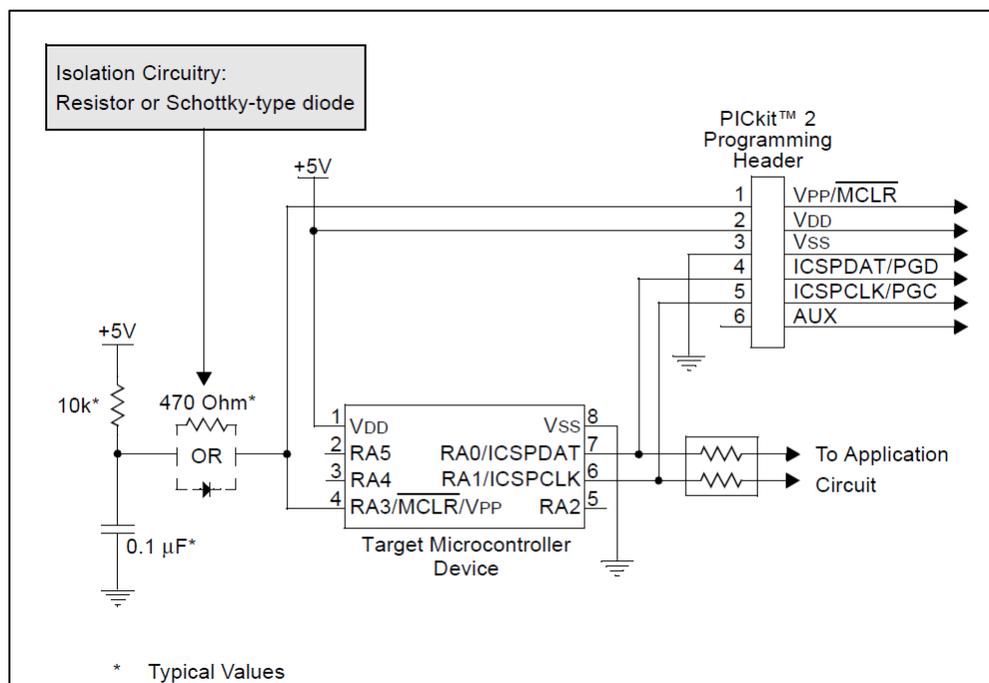


Figura 26. Circuito típico para la programación con ICSP.[15]

5.2.3 Red inalámbrica de sensores

Para la red inalámbrica de sensores, por decisión de la coordinación del proyecto, se utilizan los sistemas embebidos Wasp mote de Libelium, los cuales cuentan con módulo XBee 802.15.4 que se encargan de esta transmisión.



Figura 27. Módulo Xbee 802.15.4 [13]

Así, los Wasmote se encargan de recibir en sus puertos tanto analógicos como digitales, las señales recibidas por los diferentes módulos de hardware diseñados para cada sensor, procesan los datos recibidos y los envían inalámbricamente a un nodo central que corresponde al Gateway, el cual solo recibe y envía a la PC.



Figura 28. Wasmote Gateway [13]

Cada nodo puede estar a una distancia menor a 500 metros del Gateway para que se logre la comunicación entre ambos. En este caso, se utiliza una topología en estrella, en donde cada nodo se comunica solamente con el nodo central.

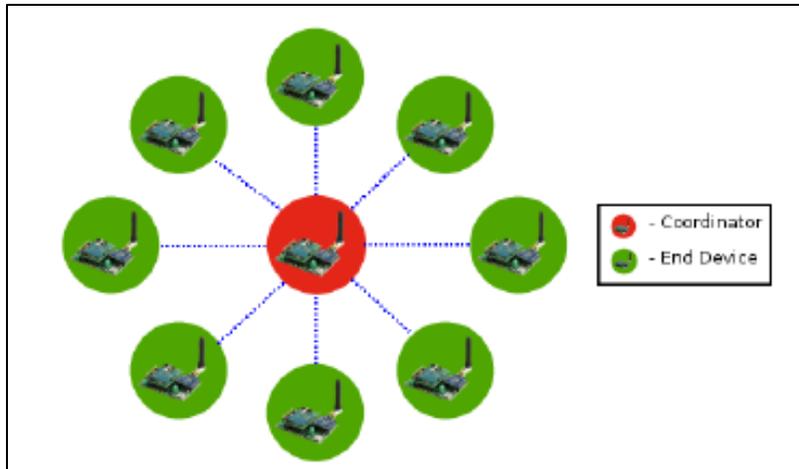


Figura 29. Topología en estrella [13]

5.3 Descripción del software

En este apartado, se explicará detalladamente cada una de las rutinas de software implementadas para la programación de las diferentes etapas del sistema. Se utilizarán diagramas de flujo para la explicación de cada algoritmo.

Se iniciará con las rutinas para la programación del PIC para el sensor SR50A, seguido de los algoritmos para los Wasmote y los módulos XBee. Luego, se explicará detalladamente el programa realizado para la recepción, procesamiento y envío de la información en la PC. Por último, se hablará de los programas utilizados para la recepción de la información en el servidor que es el que recibirá los datos vía FTP y los almacenará para su posterior análisis.

5.3.1 Programación del PIC para el sensor SR50A

Para la programación del PIC18F4550, se utilizó el programa MPLAB de Microchip, el cual es un software libre creado por esta empresa para la programación de sus componentes. Además, se utilizó el lenguaje de programación C18 el cual es un compilador de 32-bits para Windows.

En el programa implementado para el PIC, se tienen 2 rutinas:

- Programa principal.
- Interrupción por puerto serie.

El programa principal es el que lleva la mayor parte de código y es el que se encarga de procesar el dato recibido y enviarlo por puerto digital al Wasmote. En la siguiente figura se observa el diagrama de flujo para esta rutina.

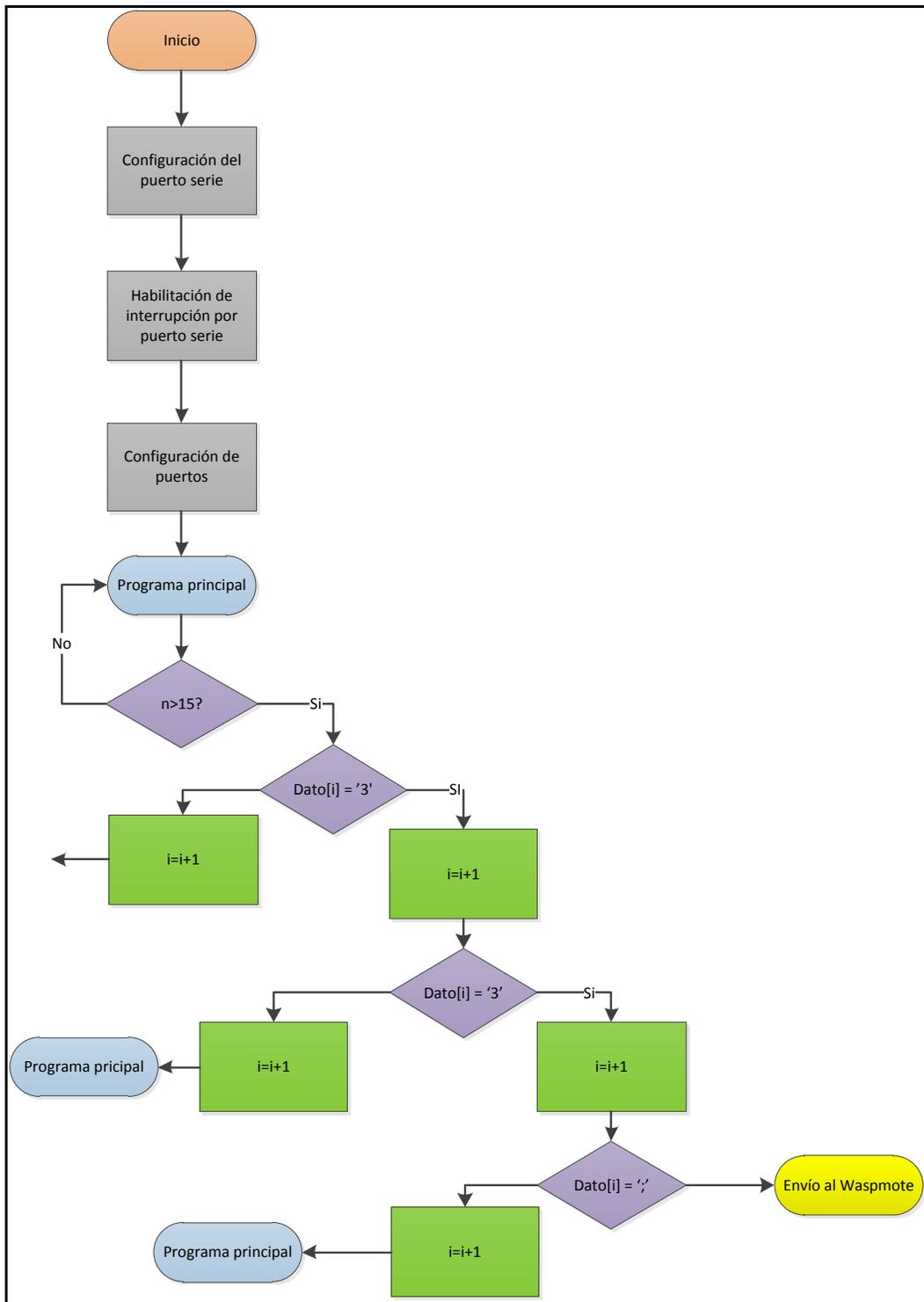


Figura 30. Diagrama de flujo del programa principal para el PIC18F4550

Tomando como base la figura 30, se explicará detalladamente la rutina para el programa principal del PIC18F4550. Primero, se configura el puerto serie de la siguiente forma:

- Interrupción por recepción.
- Modo asíncrono.
- Transmisión y recepción de 8 bits.
- Modo continuo de recepción.
- Baud rate alto.
- SPBRG=25.

El puerto serie se configura para un baud rate de 9600 bps, por lo que el valor SPBRG se elige según la tabla 20-3 de la hoja de datos para este PIC.

Luego de que se configura el puerto serie, se procede a configurar los registros de interrupción. Así, se configura el registro RCON para deshabilitar las prioridades, el INTCON para habilitar la interrupción de los periféricos y las interrupciones globales. Por último antes de empezar con la rutina principal del programa, se configura el puerto D como salida, ya que éste se utilizará en el envío del dato al Waspnote.

La trama de datos enviada por el sensor es de 22 caracteres. Pero el dato de la medición realizada por el sensor, corresponde a los caracteres del 6 al 10, por lo que se utiliza una variable llamada "n" que cuenta los caracteres recibidos de la trama.

En la rutina principal, primero se revisa si "n" es mayor a 15, si es así, el dato de la medición en milímetros de la distancia entre el sensor y el blanco, ya fue recibido. Por lo tanto, se empieza a analizar la trama hasta encontrar dos caracteres 3 seguidos de un punto y coma. Esto quiere decir que los 4 caracteres siguientes corresponden a la medición de la distancia en milímetros; por lo que se puede iniciar con el envío del dato al Waspnote.

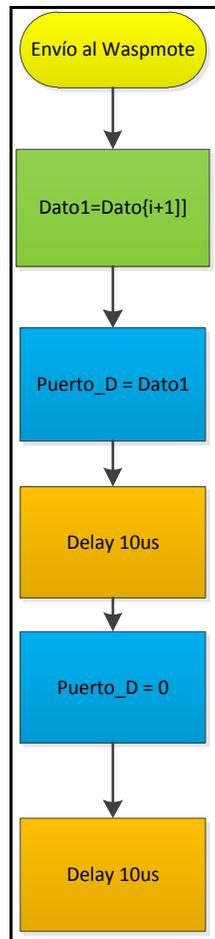


Figura 31. Continuación rutina del programa principal para el PIC18F4550

El envío de los datos al Wasmote, se hace mediante el puerto digital D del PIC. Primero, se envía por este puerto el primer carácter, se hace un retraso (delay) de 10ms, se vuelve a poner este puerto en 0 y se repite el retraso de 10ms. Lo anterior se repite 3 veces más, para enviar los 4 datos de la distancia.

Una vez que se ha enviado el dato medido al Wasmote, se reinician las variables de control a 0 y se vuelve al inicio del programa a consultar si ya se han leído los primeros 15 caracteres de la trama.

El PIC hace una interrupción en lo que esté realizando, cuando haya un dato disponible en el puerto serie. Si es así, revisa la bandera de interrupción, lee el dato, aumenta el contador de caracteres recibidos y pone en bajo la bandera.

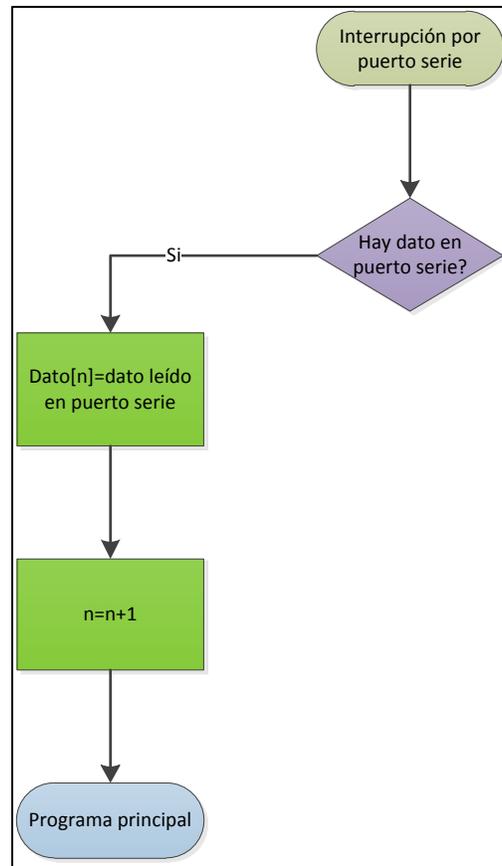


Figura 32. Rutina de interrupción por puerto serie

5.3.2 Red inalámbrica de sensores

Para la programación de la red inalámbrica de sensores, se tuvieron que realizar varias rutinas. Una para la lectura de cada uno de los dos sensores utilizados, el de medición de tensión superficial y el de deflexión, y por último la programación de los módulos XBee.

Los módulos Waspote fueron programados con el entorno de programación IDE-Waspote el cual se puede descargar de la página de internet de Libelium. Como se mencionó anteriormente, este entorno usa Arduino como lenguaje de programación, el cual se basa en C/C++.

Todo programa realizado con este entorno, se divide en 3 partes:

- Declaración de variables.

- Setup: esta rutina se ejecuta solo una vez en el programa y corresponde a las configuraciones de puertos.
- Loop: programa principal que se ejecuta infinitamente.

En el setup para la programación de los Waspnote, se inicializan las librerías XBee para la transmisión inalámbrica y se habilitan los 5V para los sensores.

5.3.2.1 Programación del Waspnote para la medición de la tensión superficial

Para la programación del Waspnote para la medición del dato de tensión superficial, se utiliza el puerto analógico uno para la lectura del dato de salida del CAS.

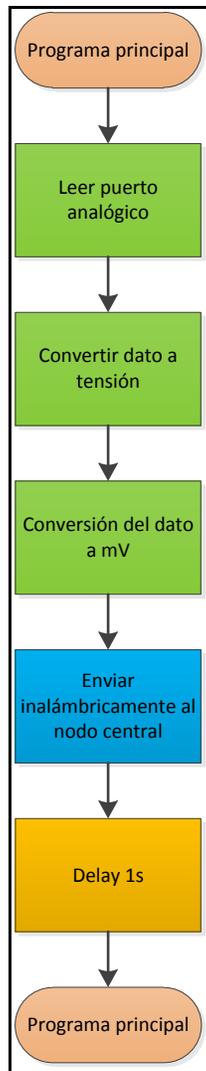


Figura 33. Diagrama de flujo para el programa principal del Waspnote para la medición de la tensión superficial

El programa inicia leyendo el puerto analógico, luego transforma este dato en un valor de tensión entre 0V y 3.3V para después, convertir el valor al valor medido por el sensor en milivoltios. Por lo tanto, se aplica la ecuación del CAS, teniendo el valor de salida, calcular la tensión de entrada y ésta dividiéndola entre 100 que sería la ganancia del circuito restador. Con esto, se conoce el valor de la tensión diferencial en el sensor ST350.

Una vez que se calcula la tensión diferencial, se procede a enviar el dato inalámbricamente hacia el Gateway o nodo central. Para esto, se crea un arreglo de caracteres, en donde se guarda el identificador del sensor seguido del dato y este arreglo es el dato que se envía.

Por último, se realiza un retraso de un segundo para empezar de nuevo con la lectura del puerto analógico.

5.3.2.2 Programación del Waspote para la medición de la deflexión

Para la programación del Waspote que se encarga de leer el dato proveniente del PIC que lee el sensor SR50A, se utilizan las primeras 5 entradas digitales, las cuales se configuran como entradas. En la siguiente figura, se observa el diagrama de flujo del programa principal para este módulo.

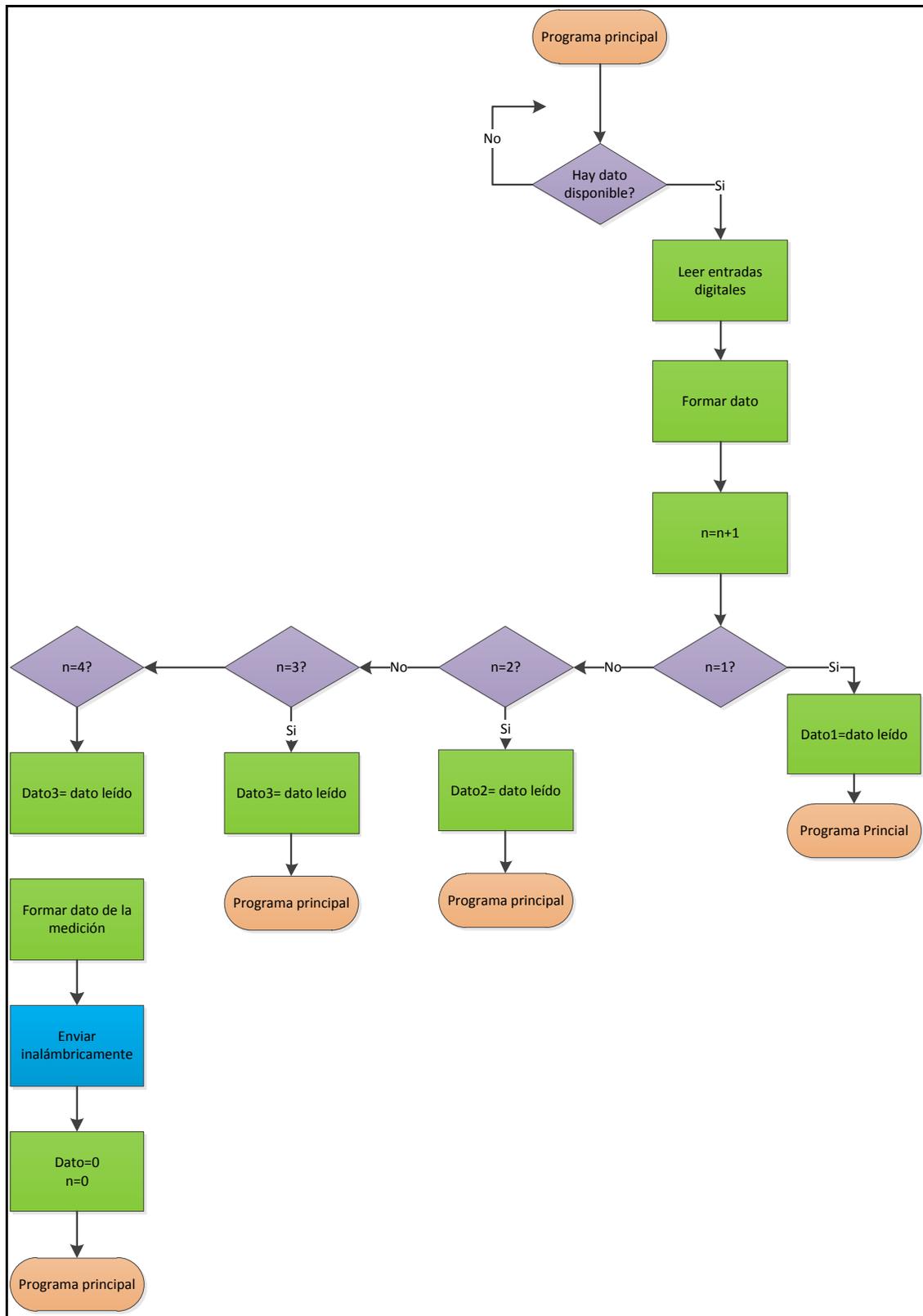


Figura 34. Diagrama de flujo del Waspote para la lectura de la medición de la deflexión

La entrada digital 5, se toma como control para que el Wasp mote conozca cuando hay un dato disponible. En caso de que hay un dato disponible, se lee las entradas digitales y se forma el dato de la siguiente forma:

- Entrada digital 4=1: al dato se le suma 8.
- Entrada digital 3=1: al dato se le suma 4.
- Entrada digital 2=1: al dato se le suma 2.
- Entrada digital 1=1: al dato se le suma 1.

Existe otra variable que controla cuántos datos han sido leídos, si es el primer dato, se guarda en una variable llamada Dato1 y se vuelve al inicio del programa. Si es el segundo o tercer dato, se realiza lo mismo pero en las variables Dato2 y Dato 3.

Cuando es el 4to dato leído, se guarda en la variable Dato4 y se forma el dato medido de la siguiente forma:

$$dato = Dato1*1000 + Dato2*100 + Dato3*10 + Dato4 \quad (5.3.2.2.1)$$

Luego, se crea el arreglo de caracteres que se enviará, el cual contiene el identificador del sensor y el dato. Se envía este arreglo inalámbricamente al nodo central y se reinician las variables de control a cero para volver al inicio del programa y esperar un dato nuevo.

5.3.2.3 Programación de los módulos XBee 802.15.4.

En una red inalámbrica formada con los módulos XBee, cada uno de los módulos debe estar programado con una serie de parámetros iguales, que son los que lo identificaran de otras redes. Estos parámetros son los siguientes: [12]

- PAN-ID: es un número de 16bits que identifica la red, debe ser diferente al número de otras redes y todos los nodos de la red deben tener el mismo PAN-ID.
- Canal: define el canal de frecuencia usado por la red para transmitir y recibir. Cuenta con 16 canales entre 2.4GHz y 2.48GHz.

Como se eligió realizar la transmisión sin API, no se necesitan otros parámetros como el identificador y la dirección de los nodos.

Para la programación de los módulos, se utilizó el software libre X-CTU de Digi, el cual sirve para la configuración y prueba de radio módems. En él se pueden realizar pruebas de comunicación con el módem y realizarle las configuraciones necesarias. En la siguiente figura se observa la pantalla principal del programa.

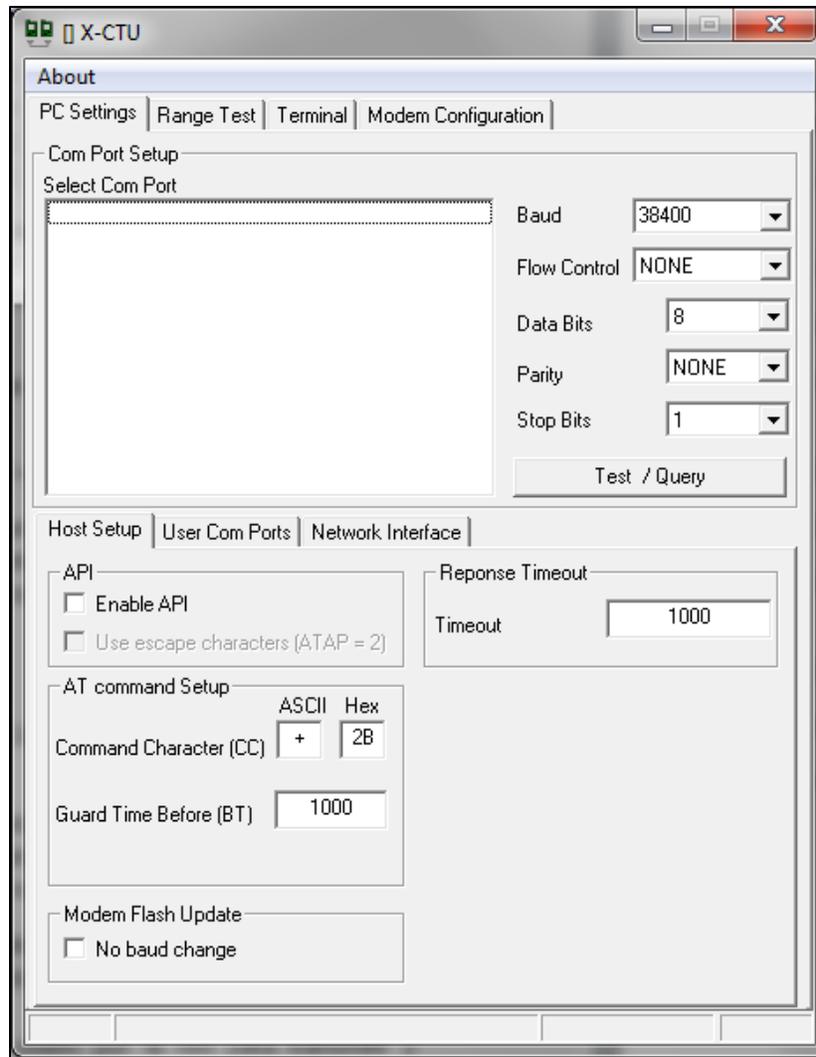


Figura 35. Pantalla principal X-CTU

Cuando existen puertos serie conectados a la PC, éste los reconoce y aparecen en la ventana Select Com Port, ahí se debe seleccionar el puerto que se desea configurar y al lado, elegir las opciones de Baud Rate, Control de flujo, cantidad de bits del dato, si existen bit de paridad y de parada. En nuestro caso, se configuró como se observa en la figura 36.

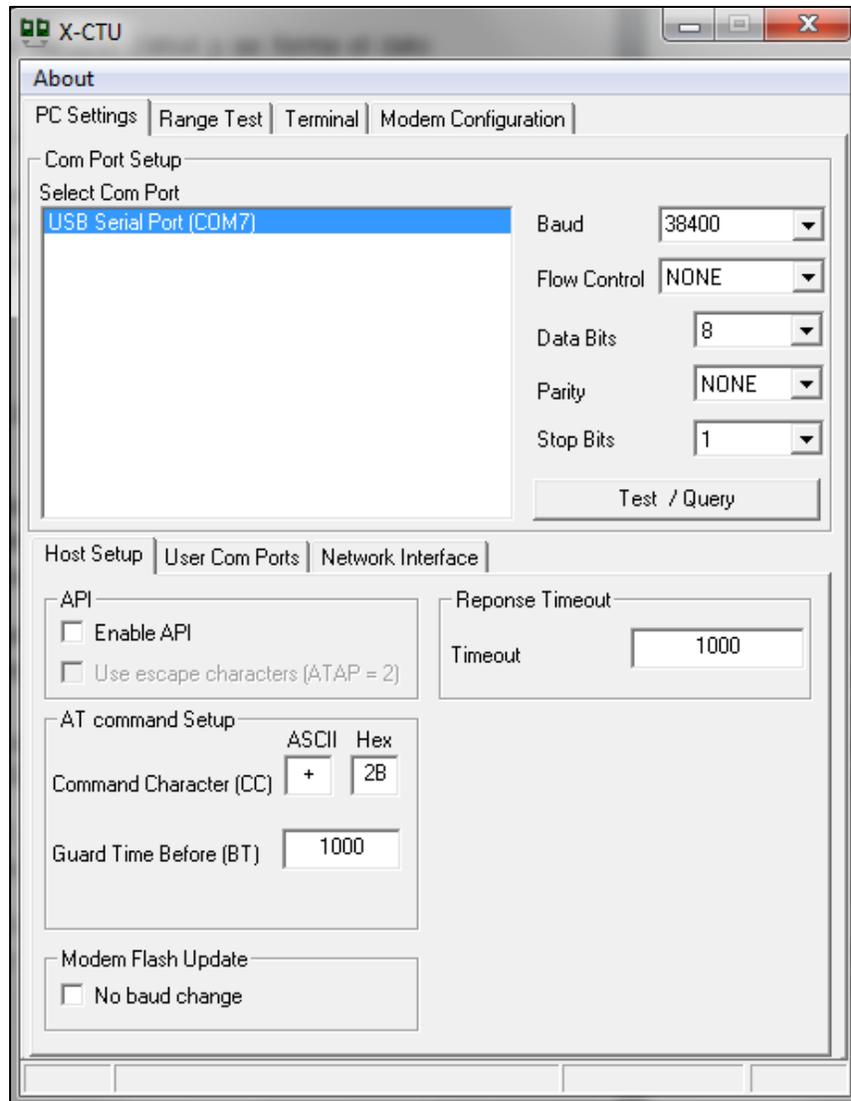


Figura 36. Configuración para la comunicación de los módulos XBee con la PC

Una vez elegidos estos datos, se selecciona el botón Test/Query para realizar una prueba de comunicación entre la PC y el módem.

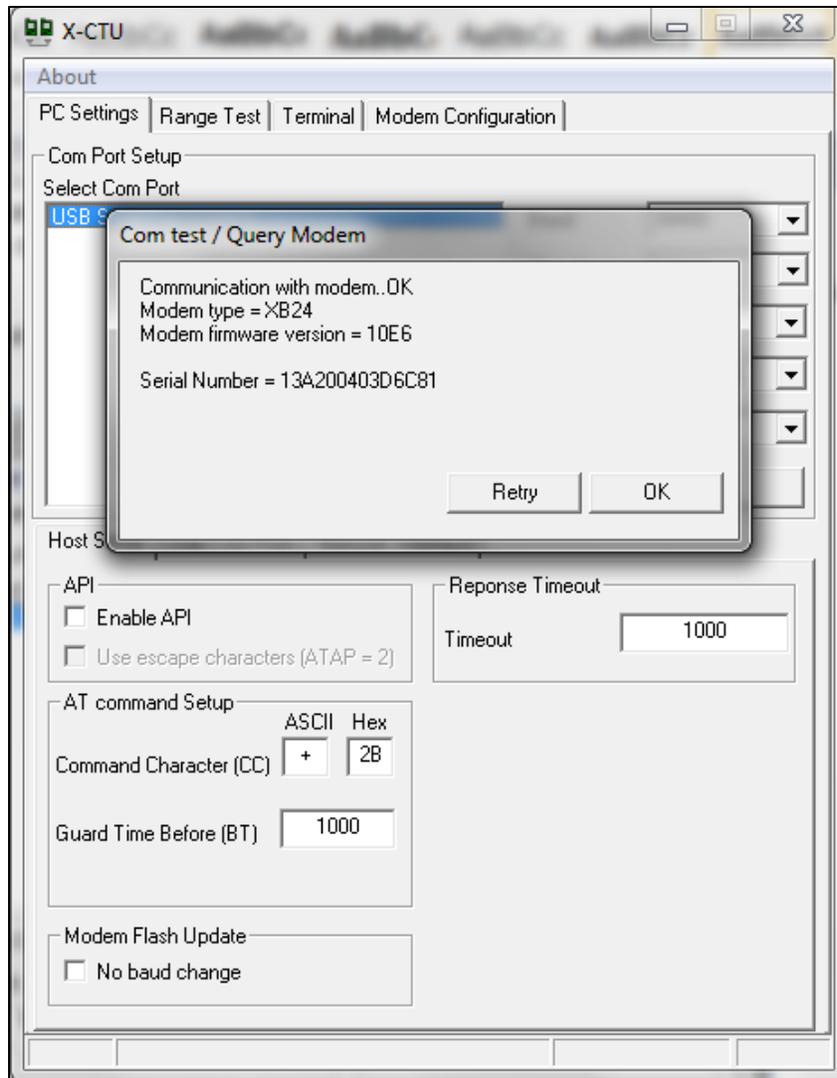


Figura 37. Prueba de comunicación entre el XBee y la PC

Si la comunicación fue exitosa, ya estamos listos para realizar la configuración del terminal. Para esto, elegimos la pestaña Modem Configuration, elegimos el botón Read y nos aparece el módem y la configuración que posee en ese momento. Cambiamos los parámetros PAN-ID y Channel a los elegidos y por último, elegimos API 0. En nuestro caso, se eligió un PAN-ID de 1134, el canal B. En el caso del XBee conectado al Gateway, se debe configurar como coordinador. Luego, presionamos el botón Write y nuestro XBee se encuentra configurado.

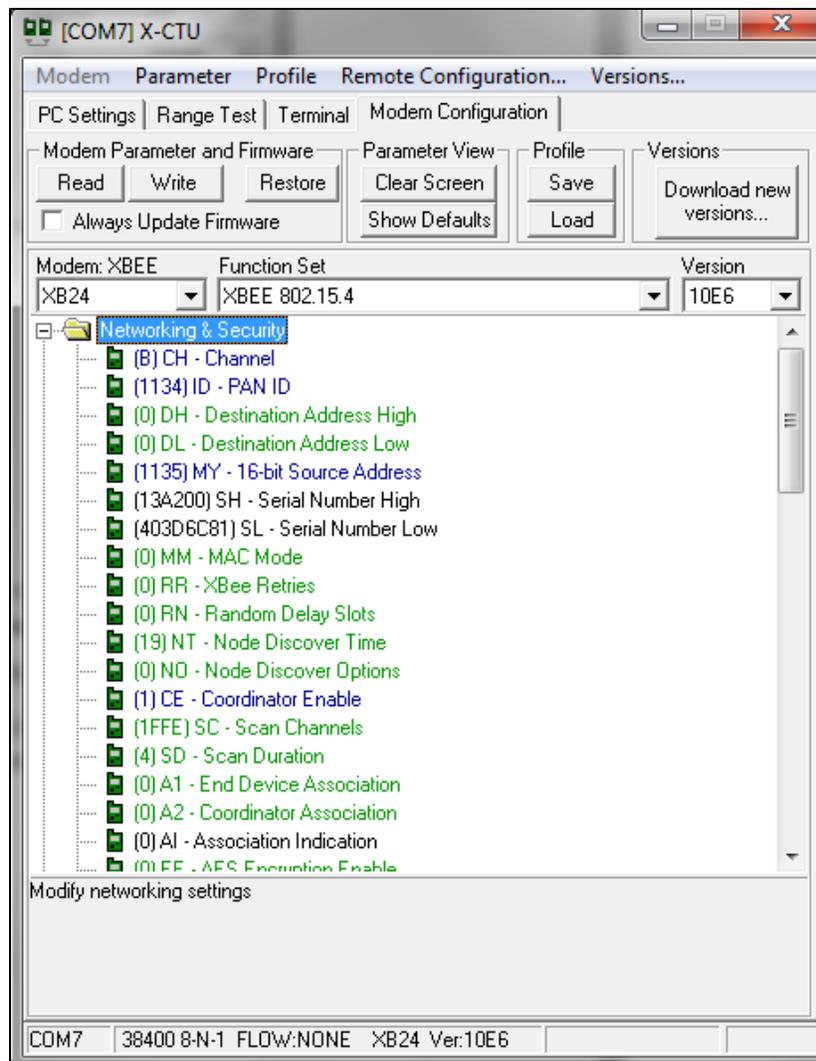


Figura 38. Configuración de los módulos XBee

5.3.3 Programa para la recepción, procesamiento y envío de la información en la PC

Para la recepción de los datos en la PC, se eligió el lenguaje Python y el sistema operativo Linux. Así, se abre un editor de texto y se guarda el archivo con un .py y listo, tenemos nuestro archivo para el programa en Python. El propósito de este programa, es recibir los datos leídos de los sensores y enviados del Gateway hacia la PC, procesarlos, guardarlos en un archivo .csv que puede ser leído por Excel y luego, enviar el archivo vía FTP al servidor.

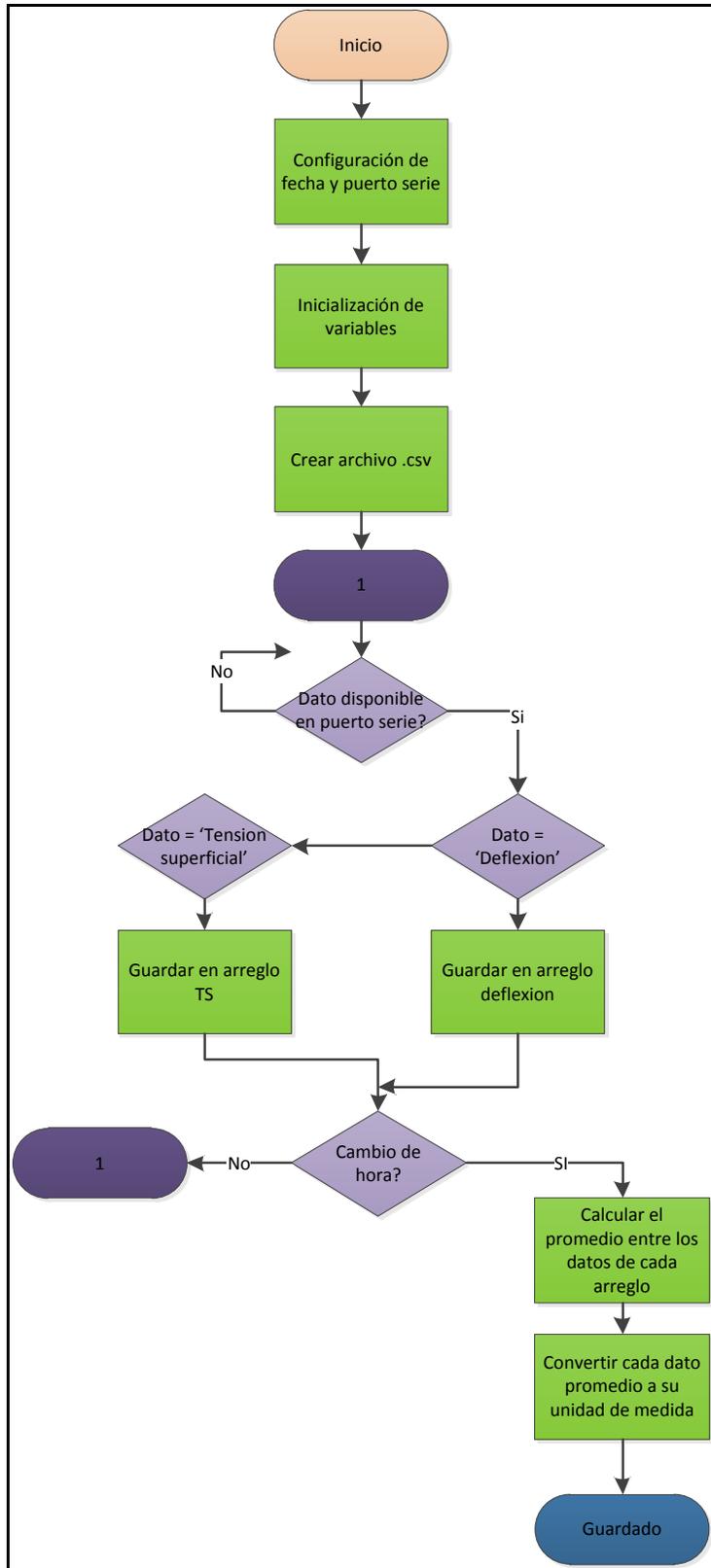


Figura 39. Diagrama de flujo del procesamiento de datos en la PC

El programa inicia con la configuración del puerto serie, ya que debe estar configurado a 38400bps para que pueda comunicarse con el Waspnote Gateway. La fecha también se configura para que tenga el formato yyyy-mm-dd. Luego, se crea el archivo .csv que lleva como nombre la fecha. A este archivo se le escriben los datos en la primera fila que corresponderán a las columnas:

- Hora
- Valor de deflexión.
- Valor de tensión superficial.

Ahora sí iniciamos con el loop infinito que irá recibiendo la información, almacenándola y enviándola. Se inicia esperando un dato en el puerto serie, si el dato está disponible, se compara para ver si es un dato de deflexión o de tensión superficial. Si fuera del primero, se guarda en un arreglo para esta variable y si fuera del segundo, se guarda en un arreglo para éste. Lo anterior se realiza cada vez que haya un dato disponible hasta que cambia la hora, la cual tiene un formato de HH:MM. Así, luego de cada minuto, se toma cada arreglo y se calcula el valor promedio. Este valor se convierte a su unidad de medida si es necesario convertirlo y con esto, ya se puede guardar los datos en el archivo.

Antes de explicar la rutina de guardado, se va a explicar la de envío FTP, ya que ésta es llamada dentro de la guardar datos en el archivo.

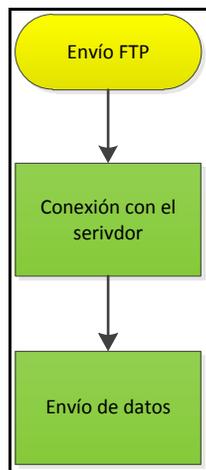


Figura 40.Diagrama de flujo del envío de datos por FTP

En el envío de datos por FTP, lo primero que se realiza es ingresar la dirección del servidor, el usuario y la clave. Se realiza la conexión con éste y si la conexión fue exitosa, se envía el archivo a la raíz especificada por el usuario. Si la conexión no se realizara, se le avisa al usuario que hubo un error.

El diagrama de flujo de la rutina de guardado, se muestra a continuación.

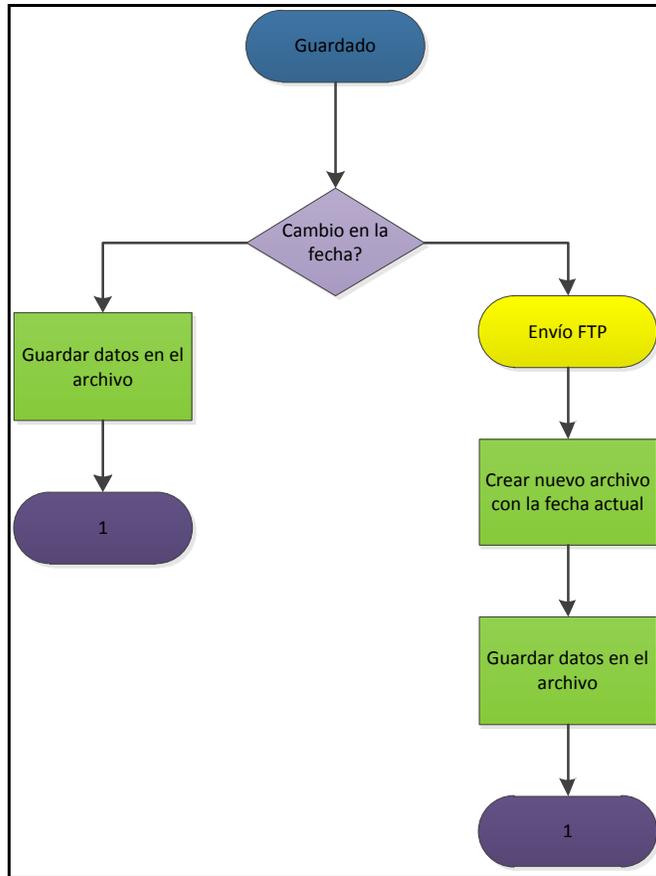


Figura 41. Diagrama de flujo del guardado en el archivo

En esta rutina, lo primero que se verifica es que la fecha actual, sea la misma del nombre del archivo, ya que se tendrán archivos diferentes para cada día. Si es la misma fecha, se guarda los datos en el archivo y se regresa al inicio del loop para la recepción de datos.

Si la fecha cambia, se envía el archivo por FTP y se crea uno nuevo con la fecha actual, se guarda los datos en este archivo y se regresa al inicio del programa.

5.4 Calibración de sensores

En este apartado, se hablará sobre la calibración de cada uno de los sensores utilizados en este proyecto. Primero se hablará sobre la calibración del sensor ST350 utilizado para la medición de la tensión superficial y luego, se hablará sobre el SR50A que es un sensor ultrasónico para la medición de la deflexión.

5.4.1 Sensor ST350 para la medición de la tensión superficial

Los sensores ST350 vienen calibrados de fábrica y no existe un proceso de calibración manual. Antes de utilizar un sensor nuevo, se debe comprobar su correcto funcionamiento. Para esto, se debe medir con un ohmímetro la resistencia

entre los cables rojo y negro y entre el verde y blanco; ambas mediciones deben ser cercanas a 350Ω . Si no concuerda esta medición, se debe devolver el sensor.

Para la calibración de estos sensores en la fábrica, utilizan un motor de paso de precisión acoplado a una fase lineal y una tensión de excitación utilizada es de 2.5Vdc. El método de calibración se llama Sistema de Calibración Automatizada para el Transductor de Tensión BDI (BDI-ASTCS).

Este método consiste en aplicarle una tensión de excitación conocida y una tensión en la superficie también conocida, se guardan los datos de salida en un rango superior a aproximadamente $1000\mu\epsilon$.

La calibración se realiza utilizando la NIST-traceablesystem que consiste en una pequeña tabla de precisión deslizable sobre un sensor de desplazamiento óptico y se verifica con un micrómetro de precisión calibrado.

Realizando pruebas de campo por parte de la empresa, se ha encontrado el error en las mediciones anda en aproximadamente $\pm 2\mu\epsilon$.

Debido a que cada sensor se calibra por aparte, tienen diferentes curvas de calibración las cuales relacionan la tensión aplicada en $\mu\epsilon$ con la tensión de salida del transductor.

Esto hace que a la hora de convertir el dato medido en mV para cada sensor, se deba buscar su curva de calibración para encontrar la ecuación a su recta y así poder calcular la tensión aplicada a éste.

5.4.2 Sensor SR50A para la medición de la deflexión [6]

El sensor SR50A es un sensor ultrasónico que mide la distancia de éste a un blanco que sea capaz de reflejar el sonido. Por ser un sensor digital, no necesita calibración y viene listo de fábrica para su utilización.

Lo que se puede variar al inicio, son sus configuraciones, ya que éste viene de fábrica configurado para que tenga una salida en el formato SDI-12. Pero moviendo 3 puentes dentro de él, se puede cambiar a otro tipo de formato.

Para desarmar el sensor, se debe primero desconectar el cable del sensor. Luego, se deben remover los 6 tornillos de la cubierta del transductor y desconectar los cables.



Figura 42. Remover los tornillos de la cubierta del transductor SR50A y desconectar los cables [6]

Luego, cuenta con unos absorbentes de humedad los cuales se remueven, se quitan 2 tornillos Philips para poder exponer la PCB.



Figura 43. Absorbentes de humedad del sensor SR50A [6]

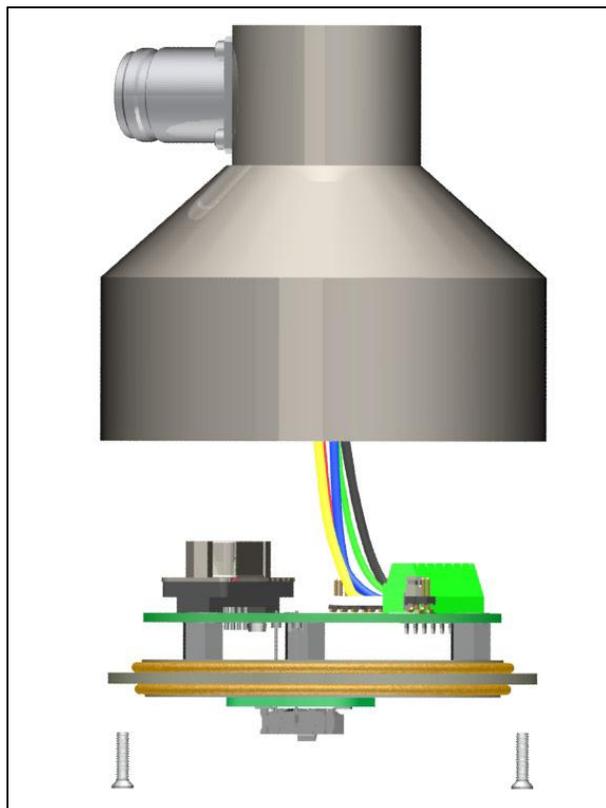


Figura 44. Remover tornillos para exponer la PCB del SR50A [6]

Con esto, se puede mover los puentes ya sea para poder utilizar en formato RS-232, RS-485 o SDI-12.

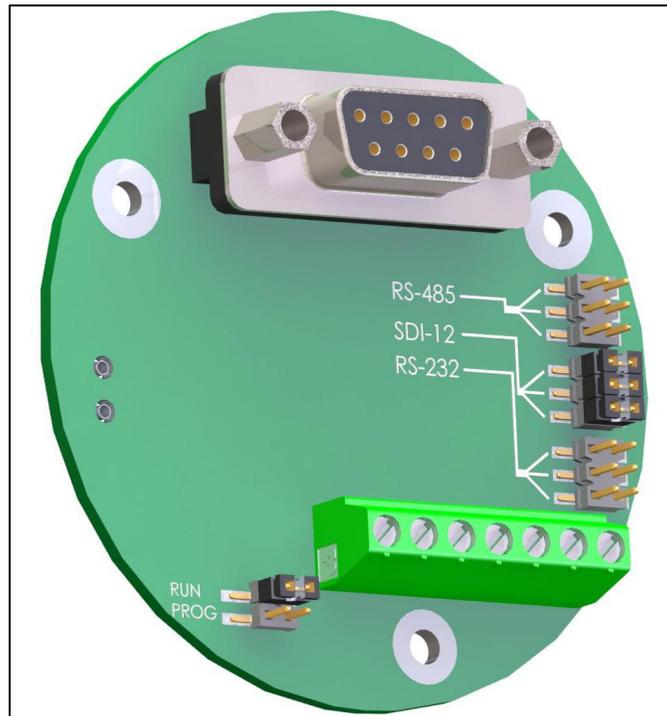


Figura 45. PCB del SR50A [6]

Además, para cambiarle las configuraciones de software, se puede utilizar el programa Hyperterminal para Windows o cualquier otro programa usado para comunicaciones. Si se utiliza el programa Hyperterminal, se debe realizar lo siguiente:

- Iniciar la sesión en hyperterminal.
- Ingresar el comando "SETUP" y presionar la tecla Enter.
- Cambiar el dato a la configuración necesaria, ya sea el baud rate, el intervalo de medición, unidad de salida, entre otros.

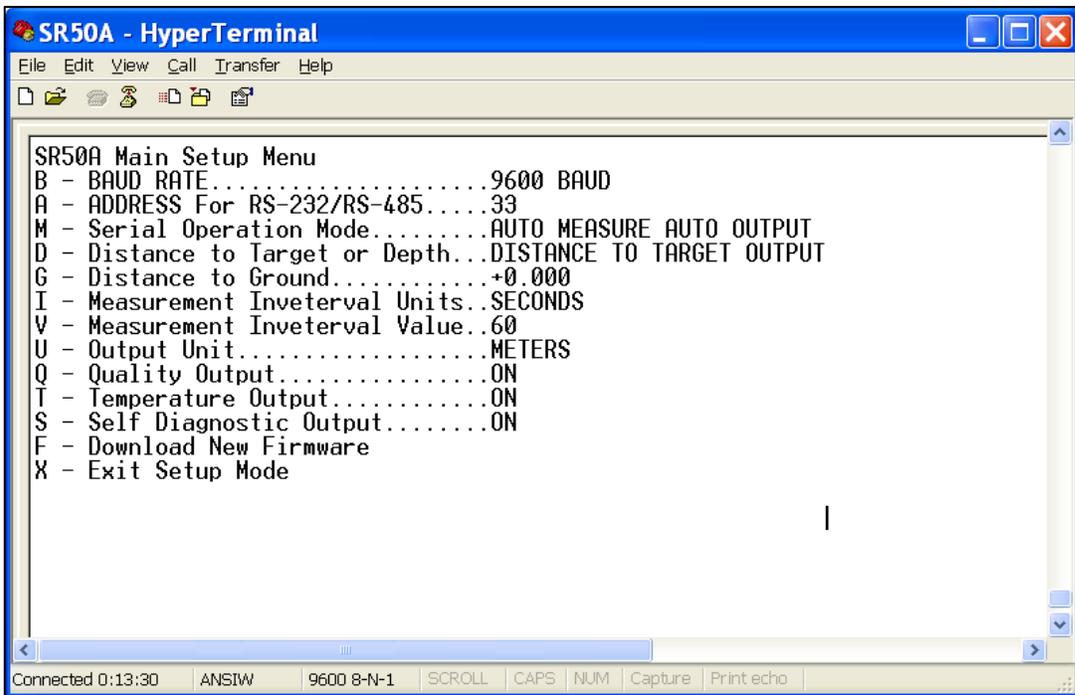


Figura 46. Menú en el hyperterminal para la configuración del sensor SR50A [6]

Una vez configurados los diferentes datos a los valores requeridos, se debe ingresar la letra “X” para salir del modo de configuración y el sensor está listo para usarse.

5.5 Instalación de los sensores

En este apartado, se explicará la forma de instalar cada sensor para su utilización. Primero se iniciará con la instalación del sensor ST350 para la medición de la tensión en el material, para luego pasar a la instalación del sensor de deflexión SR50A.

5.3.1 Sensor ST350 [7]

Alineación

El sensor solo podrá medir tensión en el eje en que se encuentra alineado; por lo tanto, entre más exacta sea la alineación, más exacta será la medición. Para realizar la alineación, se debe marcar primero la línea central en el material tanto en la dirección transversal como en la longitudinal.

La marca en la dirección longitudinal debe ser de unos 20cm y la transversal de unos 10cm aproximadamente. En la marca transversal, se deben realizar 2 marcas adicionales a 3.81cm del centro. En la siguiente figura se pueden observar las marcas que se deben realizar. Las partes encerradas en círculos, son las

superficies que son necesarias de preparar, ya que la preparación del material dependerá del material en que se colocará el sensor.

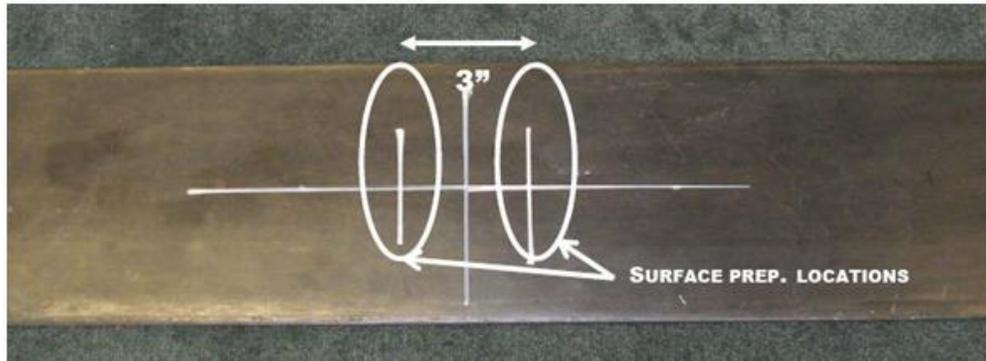


Figura 47. Marcas para la alineación [7]

Instalación

Las dos marcas realizadas a 3.81cm del centro, se utilizan para montar el sensor longitudinalmente, alineando las marcas con el centro y los extremos del sensor. El transductor se debe alinear con el eje de medición para asegurarse que la tensión será medida paralelamente a este eje. Es importante que las líneas de marcas sean dibujadas cuidadosamente debido a que los errores en las tensiones son más susceptibles a aparecer cuando hay desalineamiento. Una vez alineado el sensor, se colocan dos tornillos en cada uno de los huecos que coinciden con las líneas marcadas a 3.81cm del centro.

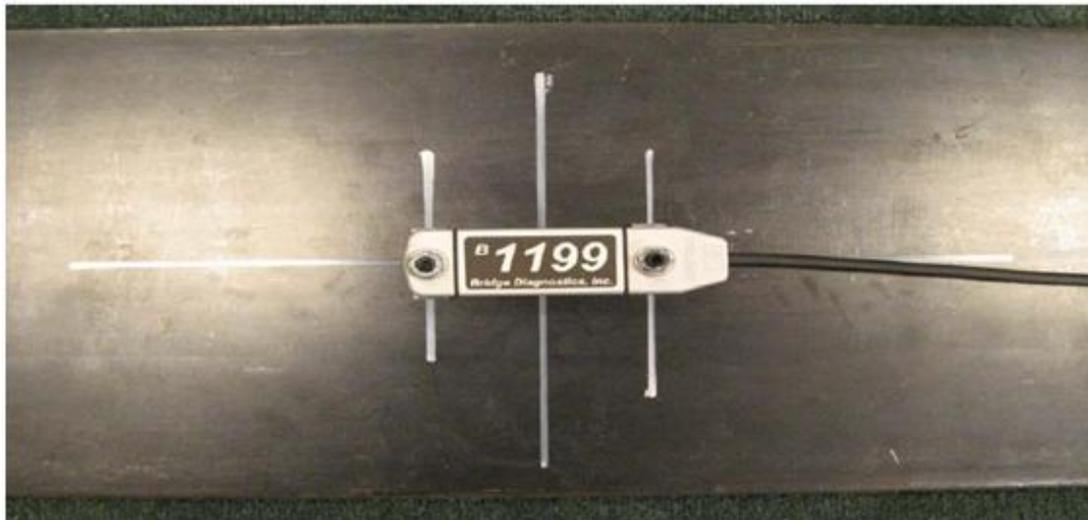


Figura 48. Instalación del sensor SR50A [7]

Ajuste de un excesivo offset en el transductor

En algunos casos, el estado de cero estrés no se podrá lograr por sí solo y existirá un offset en la medición. Esto puede ser causado por un cambio en la temperatura de la superficie. Para corregir este offset, se debe girar la tuerca con que se sostiene el sensor a la superficie, para alinearlo al estado de “cero estrés”. Si aún así, no se logra este estado, compruebe que la superficie, en que está montada el sensor, sea plana y sino, prepare el material para lograr esta condición. Para lograr el estado de “cero estrés” se deben seguir los siguientes pasos:

- Determine la dirección del offset.
- Si se encuentre muy comprimado, libere el extremo opuesto donde se coloca el cable.
- Tire de este extremo del sensor y vuelva a colocar la tuerca suavemente hasta observar en el monitor el momento en que se acerque a cero.
- Si necesita más presión en la tuerca de la que se puede, retire también la tuerca del otro extremo y vuelva a colocar las 2 tuercas hasta obtener el estado cero.
- Si el offset se encuentra en la otra dirección, muy expandido, realice los pasos anteriores pero en lugar de tirar del extremo, empuje de este.

Si aún así no se logra el estado cero, devuelva el sensor, ya que podría estar dañado.

5.3.2 Sensor SR50A [6]

Para la instalación del sensor SR50A, se debe tomar en cuenta el ángulo de medición del sensor, dentro del cual, cualquier blanco que refleje sonido puede afectar la medición. Lo recomendable es colocar el blanco perpendicular al sensor, ya que cuenta con un ángulo de aproximadamente 30°. Lo que quiere decir que en un ángulo de 30°, no debe haber ningún objeto que pueda afectar la medición.

Para determinar el radio de medición, se utiliza una fórmula en donde se debe ingresar la distancia a la que se encuentra el blanco que colocamos para las mediciones.

$$CONO_{radio} = 0.268(CONO_{altura}) \quad (5.3.2.1)$$

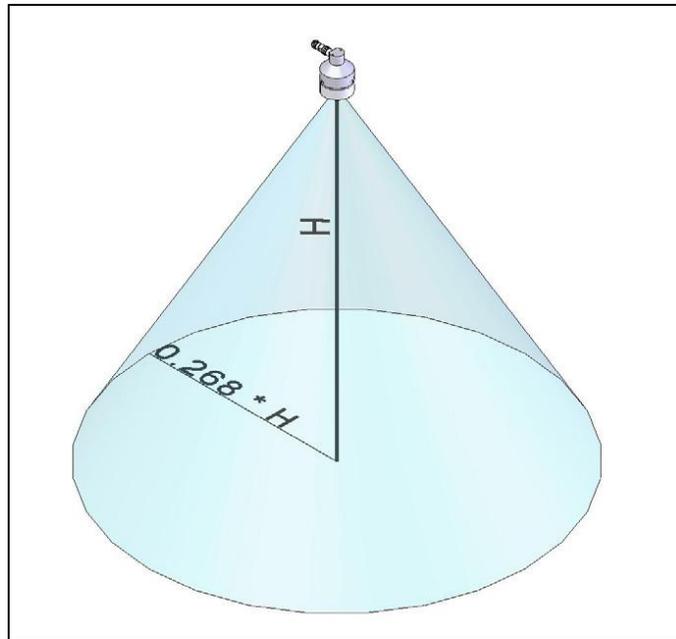


Figura 49. Ángulo de medición del sensor [6]

El blanco colocado para la medición de distancia con el SR50A, debe estar ubicado al menos a 50cm de la cara del transductor y se debe cerciorar que no se encuentre demasiado largo ya que la máxima distancia que mide es de 9999mm.

Si se desea comparar la medición utilizando un metro, se deben agregar 8mm que es la distancia de la parrilla frontal del sensor, hasta la cubierta de plástico que se puede usar como referencia.



Figura 50. Distancia de la parrilla frontal a la cubierta de plástico [6]

Para montar el sensor al lugar de medición, se debe colocar una barra de metal en donde se colocará el sensor. Existen 2 métodos:

- Usar el Kit de Montaje para este sensor: el número de la pieza es 19517, el cual se monta a la pieza CM206, la cual se puede observar en la figura 51.



Figura 51. Montaje utilizando el Kit de montaje 19517 [6]

- Montarlo sobre un cilindro de montaje, que es la pieza número 19484 y ésta pieza, conectarla a un UN-RAIL (pieza número 1049).

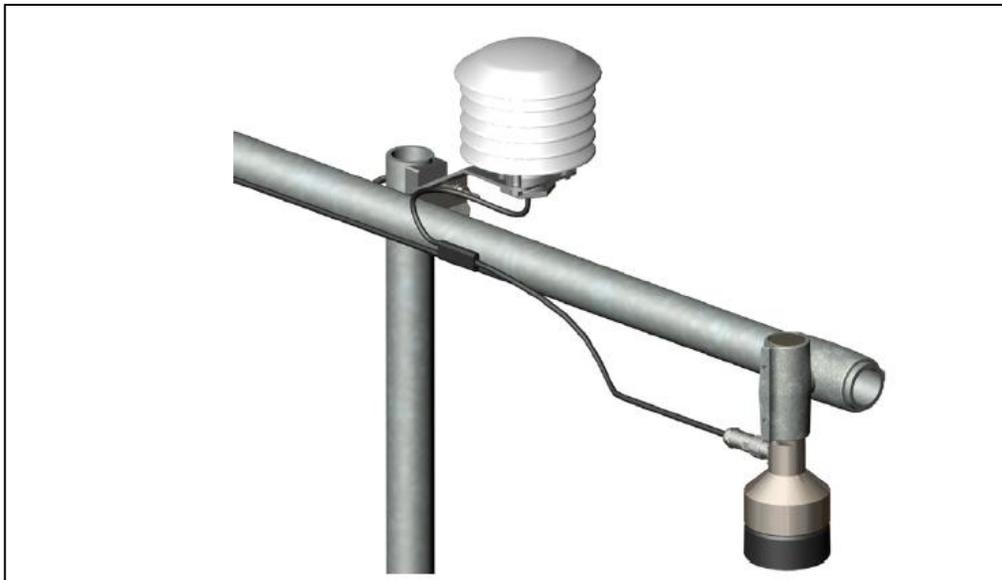


Figura 52. Montaje utilizando un cilindro y el UN-RAIL [6]

5.6 Mantenimiento de los sensores

A continuación, se explicará el mantenimiento periódico que se le debe dar a cada uno de los sensores. Se iniciará explicando el mantenimiento para el sensor ST350 y luego para el SR50A.

5.6.1 Sensor ST350

Debido a que éste sensor viene calibrado de fábrica y totalmente sellado, no existe un método manual para su calibración, su mantenimiento es casi nulo. Campbell's Scientific recomienda realizarle una recalibración cada 1 ó 2 años, o cada 15-25 instalaciones, en el caso de que se cambie de lugar constantemente. Esto con el fin, de minimizar la cantidad de mantenimiento requerido por el sensor.

Para esta recalibración, se debe mandar el sensor a la fábrica en Estados Unidos. Se recomienda que cada cierto tiempo, se revise que el valor de las resistencias entre los cables continúen siendo 150Ω.

5.6.2 Sensor SR50A

Al igual que el transductor ST350, el SR50A está diseñado para que su mantenimiento sea mínimo. Por esto, cuenta con los absorbentes de humedad que hacen que ésta no entre al sensor por los huecos de ventilación. Estos absorbentes cuentan con un gel que se mantiene de color azul mientras se encuentren en buen estado, pero si cambian a color rosado, deben ser reemplazadas debido a que éstos se encuentran saturados.

Para revisarlos y reemplazarlos, se debe seguir el mismo procedimiento visto en el apartado 5.4.2 para la configuración del sensor, pero en este caso, se llega hasta donde se encuentran los absorbentes, se retiran y se cambian.

La cubierta del transductor, se recomienda reemplazarla cada 3 años, pero si se encuentra en un lugar con una humedad muy alta, se debe reemplazar cada año.

Capítulo 6. Análisis de resultados

En este capítulo se presentan los resultados obtenidos en las diferentes pruebas realizadas a los módulos desarrollados. Se explicarán los procedimientos desarrollados para llegar a obtener estos resultados y los equipos utilizados.

Además, se analizarán estos resultados para verificar su funcionamiento y los alcances o limitaciones de las soluciones para cada etapa del sistema.

6.1 Resultados

6.1.1 Resultados para el sensor ST350

Para comprobar el funcionamiento del CAS diseñado para el sensor ST350, se simuló éste utilizando el programa Multisim. La entrada al CAS es una onda senoidal que varía entre -2V y 2V y debe tener una salida de 0V a 3,3V, que corresponden a los parámetros para los que fue diseñado el circuito.

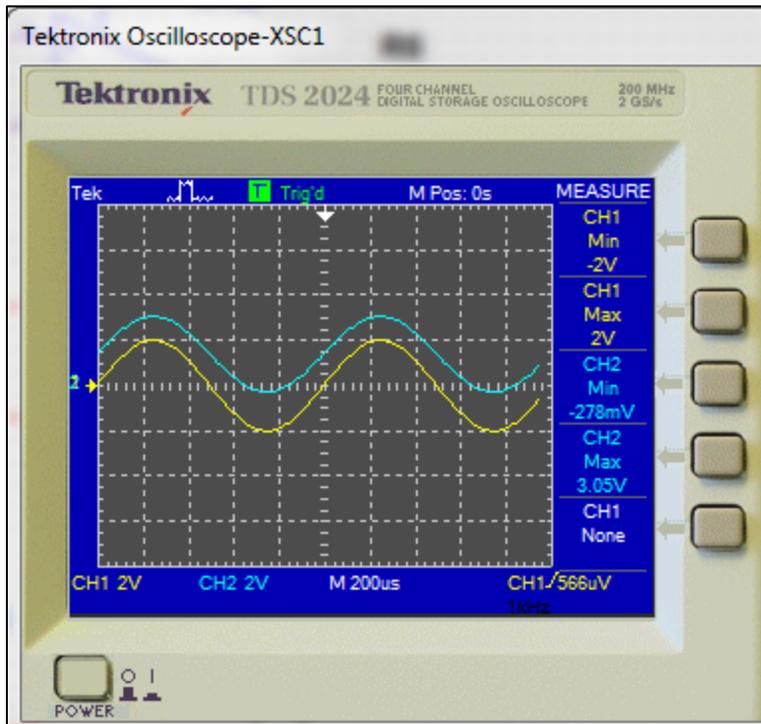


Figura 53. Simulación para el CAS del sensor ST350 utilizando Multisim

En la figura anterior, la onda de color amarillo representa la entrada al CAS y la de color verde, la salida. Se puede observar al costado que para una entrada de -2V a 2V , se tiene una salida de -278mV a 3.05V . Lo cual no concuerda con lo esperado, por lo que se varió el valor de la tensión de referencia, variando la resistencia $R6$ en la figura 21 a $1,2\text{k}\Omega$ y se obtuvo un valor mínimo de salida de -26.2mV y un valor máximo de 3.31V , que se acerca más a los valores esperados de 0V y 3.3V .

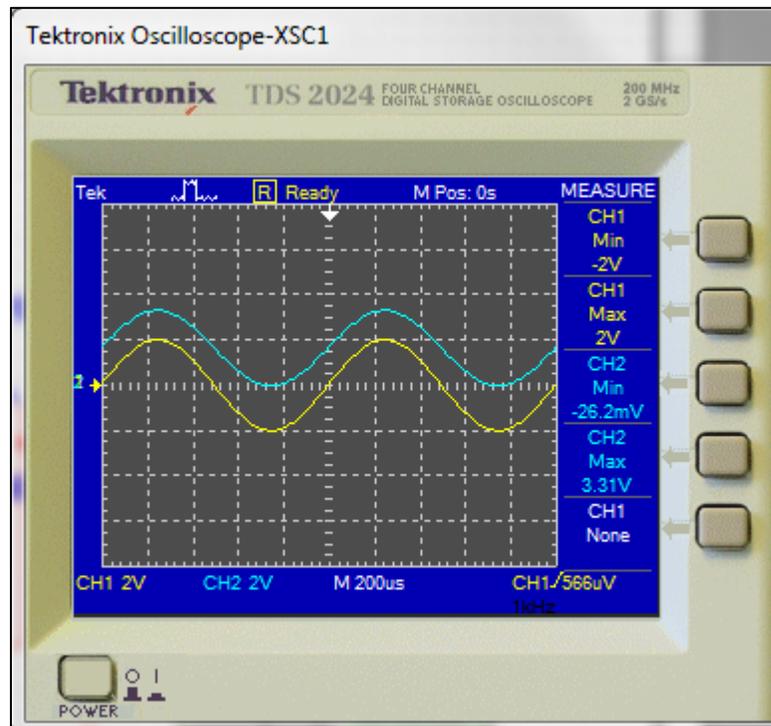


Figura 54. Simulación para el CAS del sensor ST350 cambiando el valor de $R6$

Para comprobar su funcionamiento en el laboratorio, se montó el CAS y se le realizó la misma prueba que en la simulación. A la entrada del circuito se puso una ondasenoidal con una amplitud de $-1,96\text{V}$ a $1,96\text{V}$ y se midió la salida.

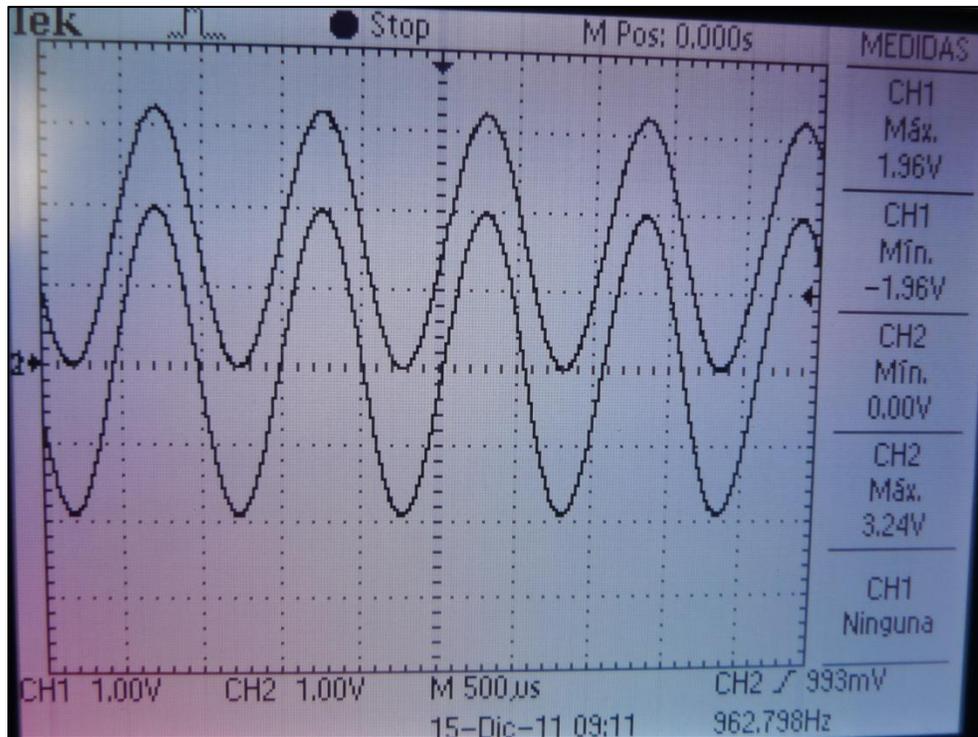


Figura 55. Resultados de la prueba experimental para el CAS del sensor ST350

En la figura anterior se observa la entrada y la salida y al lado derecho, los valores máximos y mínimos para cada uno; como se puede ver la salida tiene un valor mínimo de 0V y un valor máximo de 3.24V.

Una vez comprobado el funcionamiento del CAS, se le realizaron las pruebas a todo el sistema. Para esto, se conectó el sensor ST350 con el número de serie B2465, a la PCB y ésta al Waspote. Para poder comparar los valores medidos del sistema diseñado con los valores medidos en el datalogger, se programó éste último para enviar una señal al Waspote en el momento en que realiza la medición para que éste también la realice. El sensor se instaló siguiendo los pasos del apartado 5.3.1 en una tabla de madera flexible. A esta tabla, se le fueron poniendo diferentes pesos encima para lograr deflexiones diferentes; así, entre mayor es el peso, mayor la deflexión en el material. Los resultados obtenidos se muestran en la siguiente tabla.

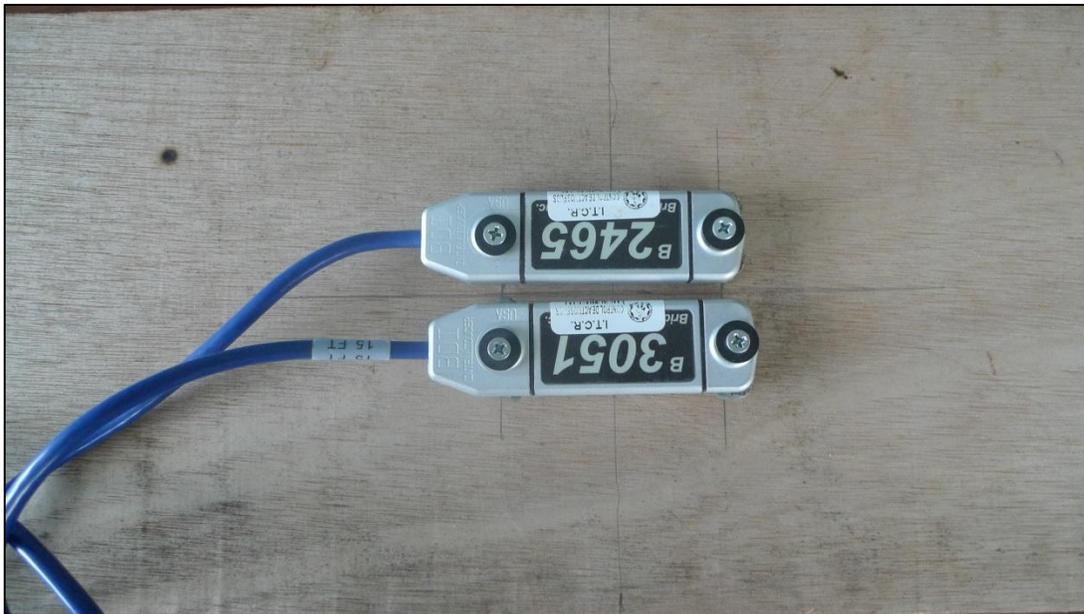


Figura 56. Sensor instalado en la tabla de madera para las pruebas experimentales

Tabla 1. Valores de tensión de salida medidos con el datalogger y con el sistema y su porcentaje de error.

# Medición	Tensión _{Dat} (mV)	Tensión _{Sistema} (mV)	% Error
1	-1,49	-1,48	0,67
2	-1,45	-1,42	2,07
3	-1,41	-1,37	2,84
4	-1,37	-1,33	2,92
5	-1,35	-1,29	4,44
6	-1,24	-1,21	2,42
7	-1,1	-1,05	4,55
8	-1,06	-1	5,66
9	-1,04	-0,98	5,77
10	-0,98	-0,93	5,10
11	-1,37	-1,38	0,73
12	-1,45	-1,5	3,45
13	-1,54	-1,62	5,19
14	-1,7	-1,78	4,71
15	-1,56	-1,64	5,13
16	-1,68	-1,78	5,95
17	-1,72	-1,77	2,91
18	-1,77	-1,83	3,39
19	-1,78	-1,84	3,37
20	-1,8	-1,86	3,33
21	-1,9	-1,97	3,68
22	-2	-2,1	5,00
23	-1,28	-1,23	3,91
24	-1,2	-1,14	5,00

En la figura siguiente, se observa el gráfico realizado con la tabla anterior para observar el comportamiento del sistema.

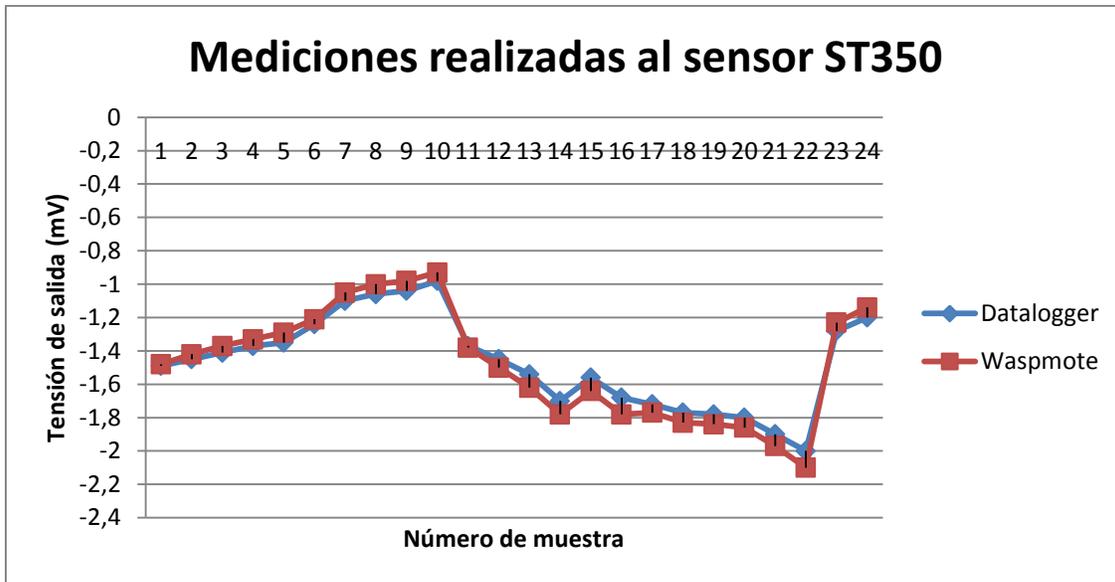


Figura 57. Gráfico de las mediciones realizadas al sensor ST350 con el datalogger y con el sistema, datos tomados de la tabla 1.

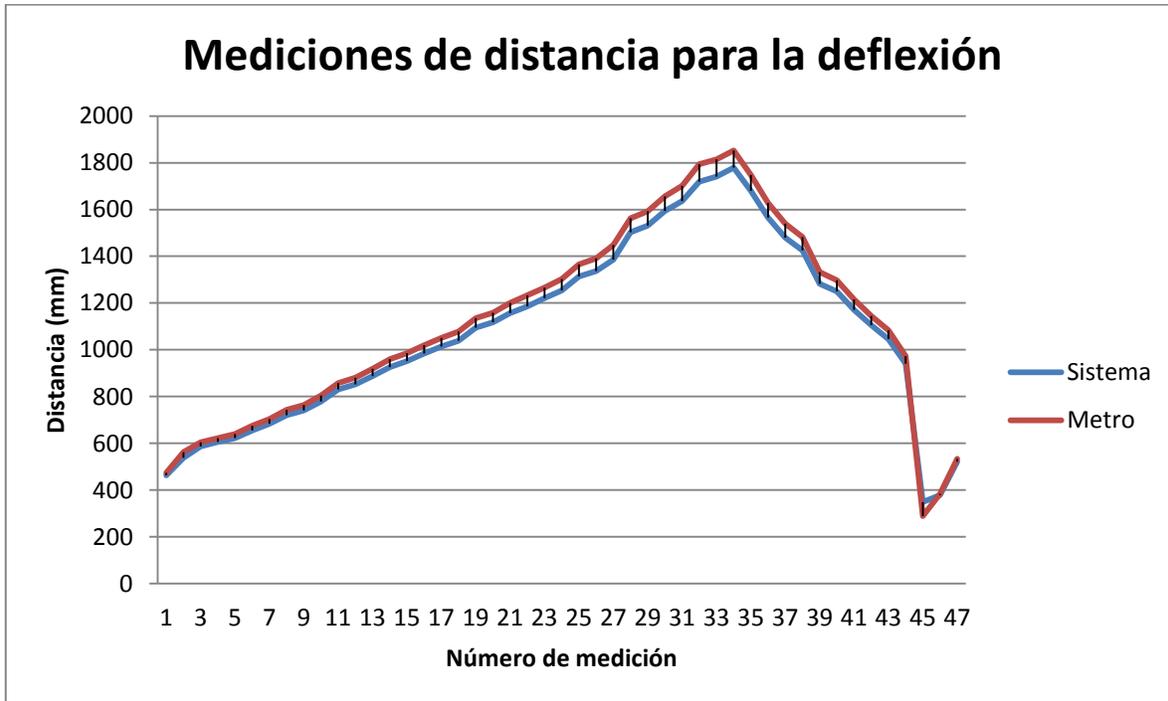
6.1.2 Resultados para el sensor SR50A

Para probar el funcionamiento del módulo diseñado para la lectura del sensor SR50A, se colocó el sensor en una posición fija y se tomó una placa de metal como blanco. Esta placa se fue moviendo aleatoriamente a diferentes distancias y midiendo con un metro para comparar este valor con el medido en el sistema. El sensor se conectó a la PCB desarrollada para éste con el PIC18F4550 y el MAX232, y esta PCB al Waspote. En la siguiente tabla se puede observar los valores medidos y el porcentaje de error entre la distancia medida con el metro y la medida con el sistema.

Tabla 2. Distancias medidas con el sistema y el metro para el sensor SR50A

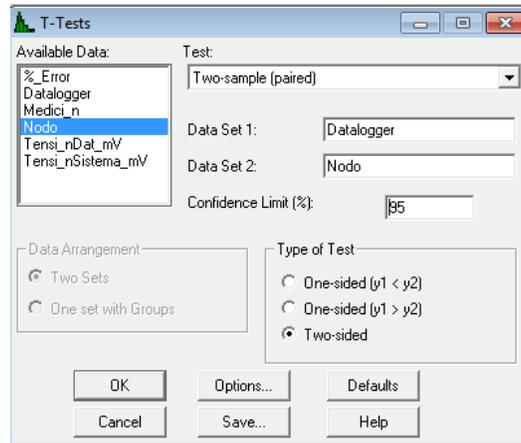
# Medición	Distancia _{sensor} ($\pm 25\text{mm}$)	Distancia _{experimental} ($\pm 0,5\text{mm}$)	%Error
1	463	474	2,32
2	537	562	4,45
3	587	604	2,81
4	605	622	2,73
5	622	640	2,81
6	655	675	2,96
7	683	704	2,98
8	720	743	3,10
9	740	764	3,14
10	779	805	3,23
11	830	858	3,26
12	852	881	3,29
13	888	919	3,37
14	926	959	3,44
15	951	985	3,45
16	984	1019	3,43
17	1014	1051	3,52
18	1039	1078	3,62
19	1095	1135	3,52
20	1117	1158	3,54
21	1157	1201	3,66
22	1186	1233	3,81
23	1221	1266	3,55
24	1254	1303	3,76
25	1313	1365	3,81
26	1337	1391	3,88
27	1386	1448	4,28
28	1503	1563	3,84
29	1531	1593	3,89
30	1594	1656	3,74
31	1636	1701	3,82
32	1719	1794	4,18
33	1741	1814	4,02
34	1779	1853	3,99
35	1680	1748	3,89
36	1565	1628	3,87
37	1480	1539	3,83
38	1425	1482	3,85
39	1282	1332	3,75
40	1250	1297	3,62
41	1171	1215	3,62
42	1105	1145	3,49
43	1046	1084	3,51
44	940	973	3,39
45	349	288,5	20,97
46	378	384	1,56
47	521	534	2,43

Con la tabla anterior, se construyó el siguiente gráfico, en donde se pueden ver los valores medidos tanto con el sistema como experimentalmente, para poder observar el comportamiento de ambos.



H1: El promedio de las mediciones de tensión superficial tomadas por medio de grabador de datos (Datalogger) es diferente al promedio medido por el nodo de la red inalámbrica de sensores para el sensor ST350.

Seguidamente se provee el conjunto de datos recolectados en el laboratorio correspondientes a la tabla 1 para realizar la prueba de hipótesis planteada arriba y se procede a realizar la prueba de hipótesis utilizando el software GENSTAT con el que se genera la siguiente entrada:



Y se obtiene el siguiente resultado:

*** Summary ***

Sample	Size	Mean	Variance	Standard Deviation	Standard error of mean
Datalogger- Nodo	24	0.01083	0.003660	0.06050	<u>0.01235</u>

95% confidence interval for mean: (-0.01471, 0.03638)

*** Test of null hypothesis that mean of Datalogger- Nodo is equal to 0 ***

Test statistic $t = 0.88$ on 23 d.f.

Probability = 0.389

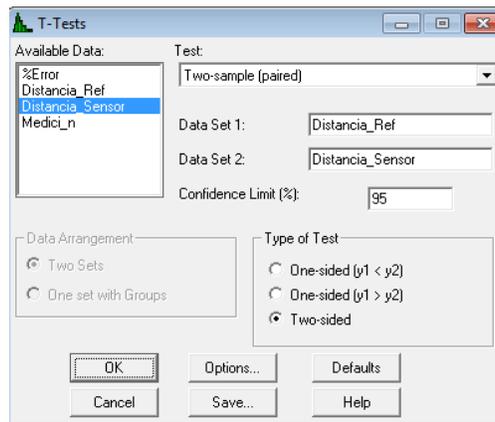
Lo anterior indica que se puede afirmar con un 95% de nivel de confianza que el valor medido por el datalogger es igual al valor medido por el nodo para el mismo sensor y la misma prueba (paired test).

Se procede seguidamente a plantear la prueba de hipótesis para el sensor ultrasónico de medición de distancia (SR50A) para el cual se plantea la siguiente hipótesis:

H0: El valor de distancia promedio medido por medio del nodo de la red inalámbrica de sensores para el sensor SR50A es igual al valor promedio medido manualmente utilizando cinta métrica.

H1: El valor de distancia promedio medido por medio del nodo de la red inalámbrica de sensores para el sensor SR50A diferente al valor promedio medido manualmente utilizando cinta métrica.

La tabla de datos obtenida en el laboratorio para la realización de la prueba de hipótesis corresponde a la tabla 2. Se muestra la entrada al software de análisis estadístico Genstat



Se obtiene la siguiente salida:

***** Two-sample T-test (paired) *****

Calculated using one-sample t-test with the null hypothesis that the mean of Distancia_Ref - Distancia_Sensor is equal to 0

***** One-sample T-test *****

Variate: Y[1].

*** Summary ***

Sample	Size	Mean	Variance	Standard deviation	Std error of mean
Dist_Ref- Dist_Sen	40	42.92	328.3	18.12	2.865

95% confidence interval for mean: (37.13, 48.72)

*** Test of null hypothesis that mean of Distancia_Ref- Distancia_Sensor is equal to 0 ***

Test statistic t = 14.98 on 39 d.f.

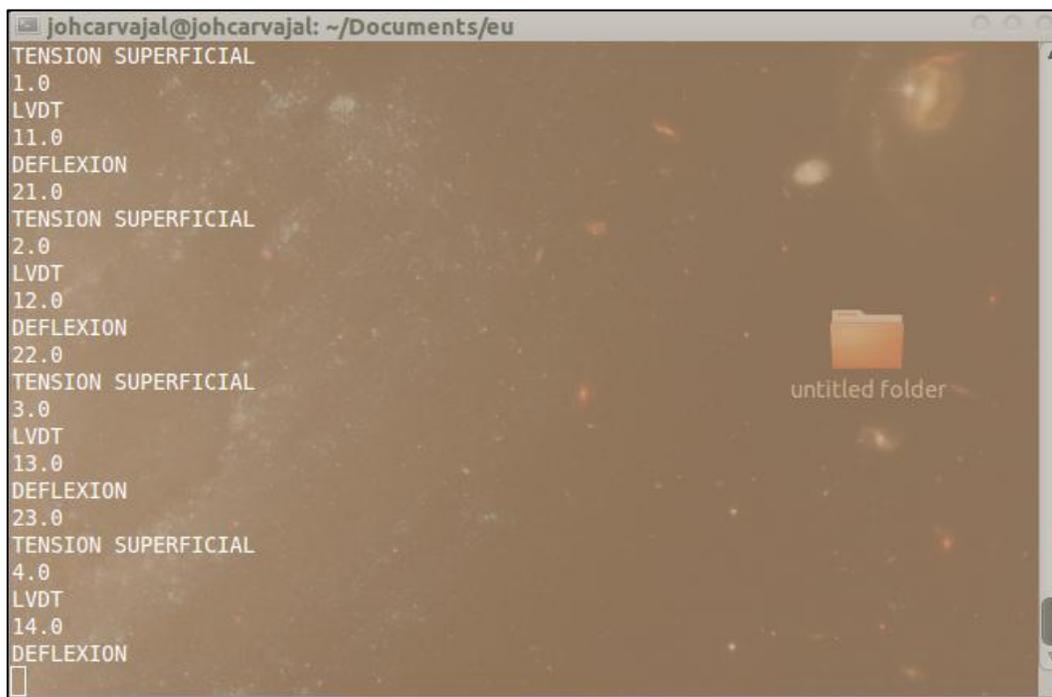
Probability < 0.001

La probabilidad de que la hipótesis H0 se cumpla es menor al 0.001.

6.1.4 Red inalámbrica de sensores y recepción en la PC

Para comprobar el funcionamiento de la red inalámbrica de sensores, se programaron los módulos Wasp mote para que enviaran datos conocidos hacia el Gateway y éste a la PC. Así, uno de los módulos se programó para que se identificara como la medición de tensión superficial y que enviara los números del 1 al 10 sucesivamente y cuando llegara la 10, volviera al 1. El otro, se programó para simular la medición de un posible sensor LVDT y que enviara del 11 al 20; el último con sensor de deflexión y que enviara del 21 al 30.

El programa diseñado en la PC en Python, se encargaría de recibir estos valores, mostrar los valores recibidos en la consola de Linux y guardar los datos en el arreglo, calcular su promedio cada 15 segundos para simular lo que realizaría realmente y guardar el valor promedio en el archivo .csv. En las siguientes figuras se muestran pantallazos de la consola de Linux con los datos recibidos y del archivo .csv.



```
johcarvajal@johcarvajal: ~/Documents/eu
TENSION SUPERFICIAL
1.0
LVDT
11.0
DEFLEXION
21.0
TENSION SUPERFICIAL
2.0
LVDT
12.0
DEFLEXION
22.0
TENSION SUPERFICIAL
3.0
LVDT
13.0
DEFLEXION
23.0
TENSION SUPERFICIAL
4.0
LVDT
14.0
DEFLEXION

```

Figura 59. Datos recibidos en la PC provenientes de los nodos en la red inalámbrica de sensores

	A	B	C	D	E
1	Hora	Deflexion	LVDT	Tension superficial	
2	12:13	24.6923076923	14.6923076923	4.64285714286	
3	12:13	24.9230769231	14.9285714286	5.0	
4	12:14	25.1428571429	15.2307692308	5.23076923077	
5	12:14	24.6923076923	14.6923076923	4.64285714286	
6	12:14	24.9230769231	14.9285714286	5.0	
7	12:14	25.1428571429	15.2307692308	5.23076923077	
8	12:14	24.6923076923	14.6923076923	4.64285714286	
9	12:15	24.9230769231	14.9285714286	5.0	
10	12:15	25.1428571429	15.2307692308	5.23076923077	
11	12:15	24.6923076923	14.6923076923	4.64285714286	
12	12:15	24.9230769231	14.9285714286	5.0	
13	12:16	25.1428571429	15.2307692308	5.23076923077	
14	12:16	24.6923076923	14.6923076923	4.64285714286	
15	12:16	24.9230769231	14.9285714286	5.0	
16	12:16	25.1428571429	15.2307692308	5.23076923077	
17					
18					

Figura 60. Archivo creado por la PC para guardar los promedios de los datos recibidos

6.1.5 Resultados del sistema

Para la comprobación del sistema completo funcionando, se conectó cada uno de los sensores a los circuitos diseñados para ellos y éstos, a un nodo Waspnote diferente.

Se conectó el Waspnote Gateway a la PC y se programó ésta para que tomara datos durante 30 minutos. Se inició la transferencia y recepción de datos. En la siguiente figura se observa la carpeta donde se encuentra el archivo de la programación en la PC, en ésta misma carpeta se guardará el archivo creado .csv, como se observará en la figura 61, el archivo no existía hasta ese momento.

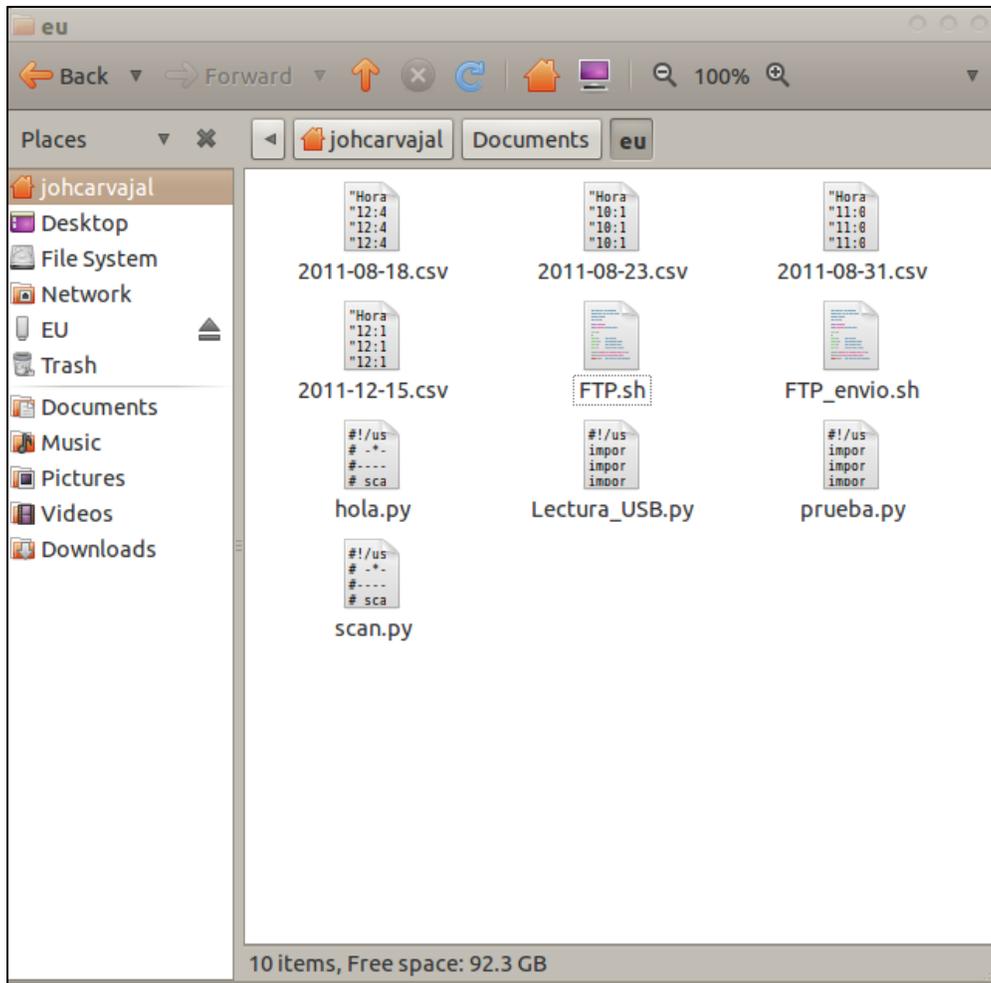


Figura 61. Carpeta que contiene los programas y archivos creados para la PC

Al iniciarse la transferencia de datos en la red y la recepción en la PC, se imprimen los datos recibidos en la consola de Linux.

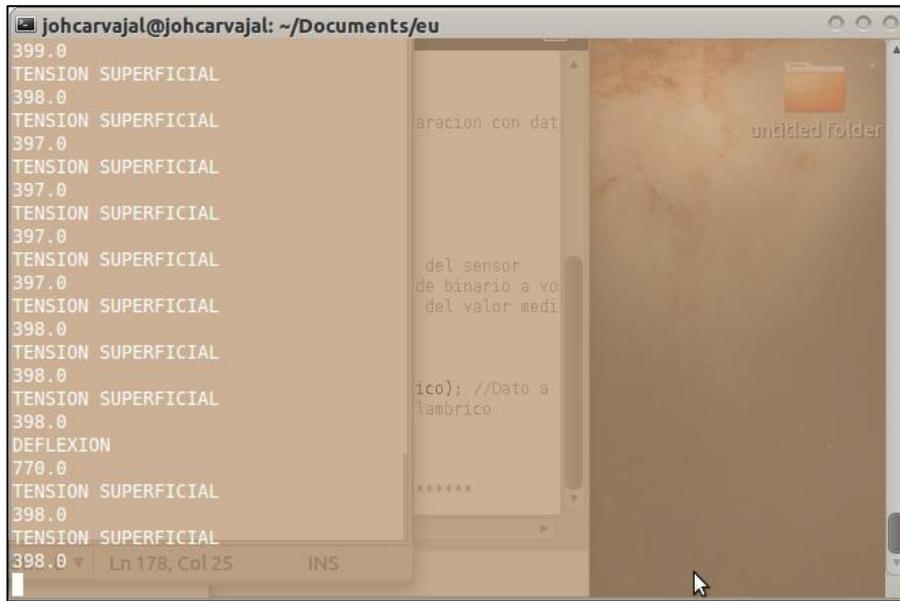


Figura 62. Recepción de datos en la PC

Se debe recordar que el tiempo de muestreo mínimo para el sensor de Deflexión SR50A, es de 1 minuto; mientras que para el de tensión superficial se eligió un tiempo de 1 segundo. Al finalizar los 30 minutos, el programa envía por FTP el archivo creado .csv que lleva como nombre la fecha. En las figuras siguientes se observa el archivo creado y el envío por FTP.

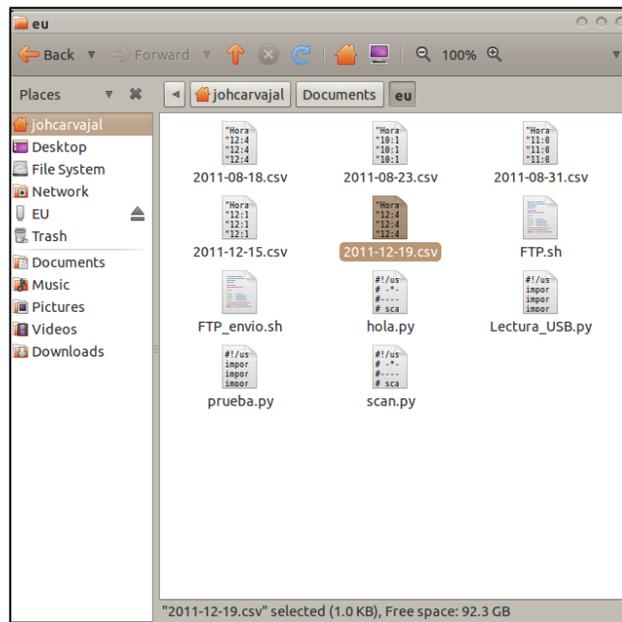


Figura 63. Al iniciar el programa en la PC se crea el archivo .csv

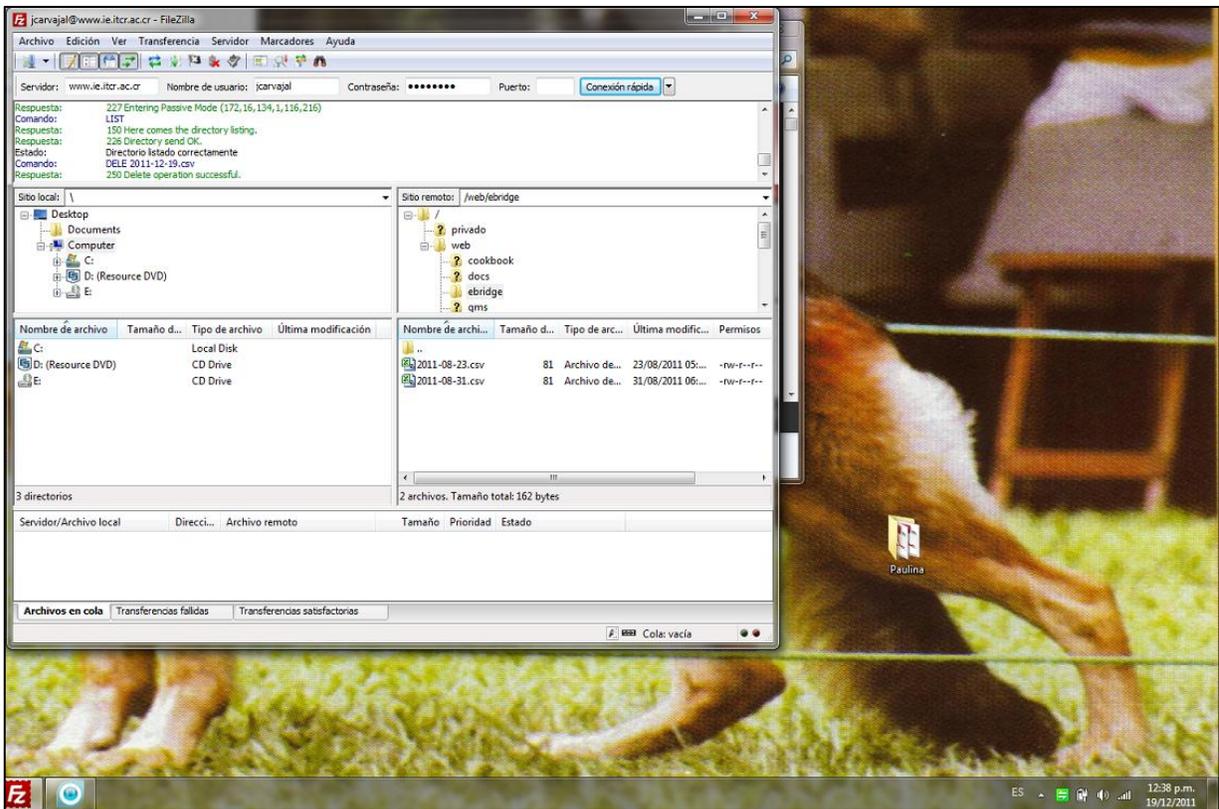


Figura 64. Servidor donde se enviará el archivo por FTP.

En la figura anterior se puede observar que eran las 12:38pm del 19 de diciembre del 2011, esta la ventana al servidor donde se guardará el archivo. Como se puede observar, existen solo 2 archivos Excel en la carpeta y ninguno con un archivo de la fecha de ese día.

Con el inicio del programa en la PC, se crea el archivo donde se guardarán los datos, el cual se puede observar en la figura 63. Al para los 30 minutos, se envía el archivo por FTP, lo cual se puede observar en la consola de Linux.

```
johcarvajal@johcarvajal: ~/Documents/eu
361.0
TENSION SUPERFICIAL
352.0
TENSION SUPERFICIAL
355.0
TENSION SUPERFICIAL
354.0
TENSION SUPERFICIAL
340.0
TENSION SUPERFICIAL
338.0
TENSION SUPERFICIAL
342.0
TENSION SUPERFICIAL
341.0
ya
promedio deflexion:
667.0
promedio TS:
-4.43560813162
enviar ftp
2011-12-19.csv
fin
```

Figura 65. Envío del archivo por FTP

2011-12-19.csv - LibreOffice Calc

File Edit View Insert Format Tools Data Window Help

Find Text

Arial 10

A1 f(x) Σ = Hora

	A	B	C	D	E	F
1	Hora	Deflexion	LVDT			
2	12:46	921.0	-4.3295121301			
3	12:47	770.0	-4.45482711202			
4	12:48	680.0	-4.46940702818			
5	12:49	655.0	-4.49071358749			
6	12:50	655.0	-4.47537153934			
7	12:51	1257.0	-4.44024719585			
8	12:52	1068.0	-4.45812968394			
9	12:53	1068.0	-4.41307553391			
10	12:54	1068.0	-4.41572642775			
11	12:55	923.0	-4.45096122515			
12	12:56	923.0	-4.40048378813			
13	12:57	923.0	-4.42633000315			
14	12:58	923.0	-4.46073639622			
15	12:59	896.0	-4.43560813162			
16	13:00	896.0	-4.43494540815			
17	13:01	883.0	-4.46920821114			
18	13:02	883.0	-4.43096906738			
19	13:03	883.0	-4.46464646465			
20	13:04	883.0	-4.43428268469			
21	13:05	770.0	-4.43428268469			
22	13:06	770.0	-4.45878136201			
23	13:07	770.0	-4.44965786901			
24	13:08	770.0	-4.44965786901			
25	13:09	770.0	-4.41506370429			
26	13:10	770.0	-4.41572642775			
27	13:11	770.0	-4.45030954708			
28	13:12	770.0	-4.41572642775			
29	13:13	667.0	-4.43560813162			
30	13:14	667.0	-4.46985988921			
31	13:15	667.0	-4.43560813162			

Figura 66. Archivo .csv creado por la PC

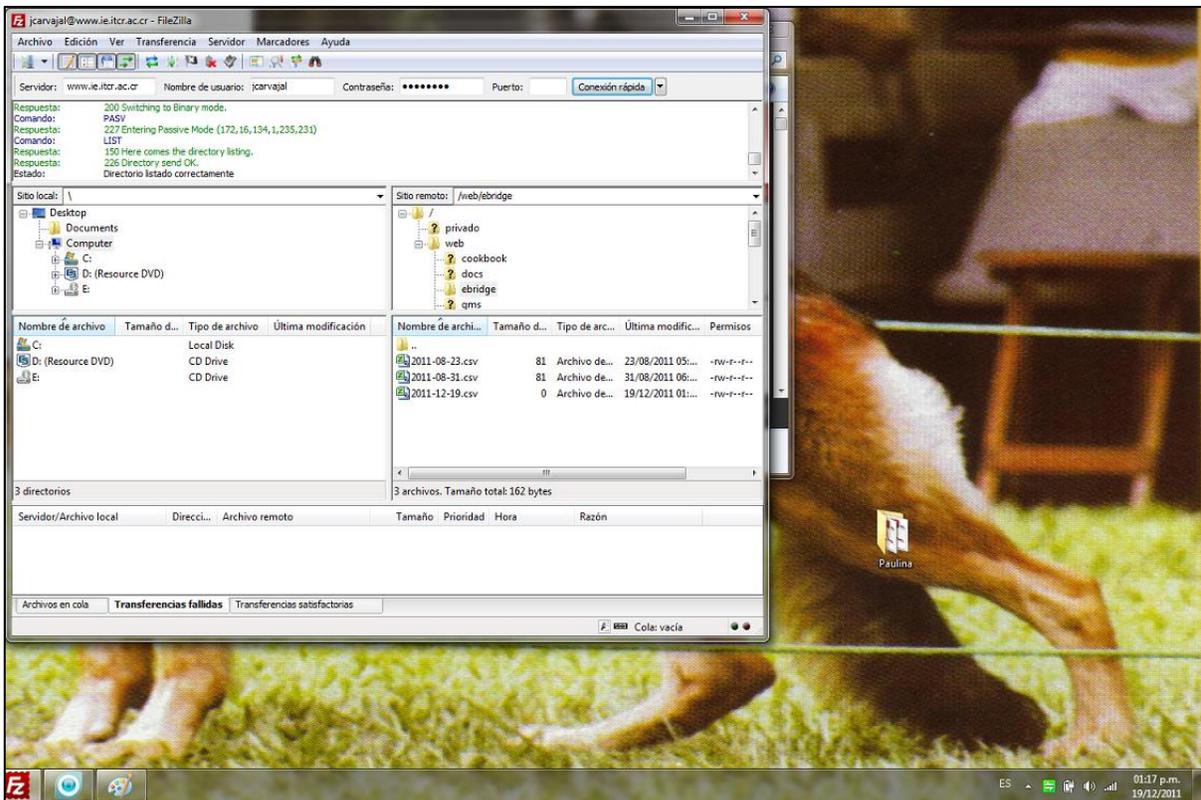


Figura 67. Recepción en el servidor FTP del archivo

En la figura 66, se observa que el archivo fue recibido en la carpeta ebridge del servidor.

6.2 Análisis de resultados

Se iniciarán con el análisis de los resultados obtenidos para el sensor ST350, seguido de los del sensor SR50A, la red inalámbrica de sensores y por últimos del sistema completo. Se analizará si los resultados son significativos o no, el por qué de ello y las posibles causas de error, en el caso de que existiera.

6.2.1 Análisis de resultados para el sensor ST350

Para el sensor encargado de la medición de la tensión superficial (ST350 de Campbell's Scientific), se diseñó un sistema capaz de leer su salida e interpretarla. Este transductor está formado por un puente Wheatstone mostrado en la figura 6, en donde su salida corresponde a la diferencia de tensión entre 2 nodos del circuito. Para poder obtener esta diferencia, se diseñó un circuito restador con una amplificación de 100 y además, se diseñó un circuito acondicionador de señales, encargado de acondicionar la salida del restador para poder ser leída por el

módulo Waspnote, el cual debe leer el dato, analizarlo y enviarlo inalámbricamente al nodo central.

En las figuras 53 y 54, se observan los resultados de las simulaciones realizadas a este circuito utilizando Multisim. Como se puede observar en la figura 53, los valores de salida no son los esperados de 0V a 3.3V, sino que se obtiene una tensión mínima de -278mV y una máxima de 3.05V para una entrada senoidal que va de -2V a 2V. Debido a esto, se varió la tensión de referencia a la entrada del CAS hasta obtener tensiones de salida de -26.2mV a 3.31V, los cuales se acercan mucho más a los valores esperados.

En el laboratorio, se montó el CAS y se le realizó la misma prueba que a la simulación. Se le conectó una entrada senoidal de -1.96V a 1.96V y con un osciloscopio Tektronix, se midió su salida. En la figura 55 se pueden observar los resultados, en los cuales, para una tensión de entrada de -1.96V, se obtuvo una salida de 0.00V y para una entrada de 1.96V, una salida de 3.24V. Esto se acerca mucho a los valores esperados para este circuito, en donde se debe recordar que los componentes utilizados no son ideales, aparte de que las fuentes utilizadas y la protoboard pueden adicionarle un ruido pequeño a la medición y por lo tanto, es muy difícil lograr valores exactos, pero lo importante es que se obtuvieron valores muy cercanos a éstos.

Luego, se procedió a conectar todo el sistema para la lectura y análisis del sensor ST350. Para esto, se programó el datalogger CR1000 para enviar una señal al Waspnote, con el fin de que el momento de la medición fuera el mismo. Así, se conectó el sensor al circuito restador, éste al CAS y la salida al Waspnote. Se utilizó el sensor con el número de serie B2465 y se conectó paralelamente a ambos, al sistema y al datalogger utilizando la alimentación del datalogger. Los resultados obtenidos se observan en la tabla 1.

En esta tabla se calcularon los porcentajes de error entre la medición del datalogger y la del sistema diseñado. Se puede ver que se obtienen errores menores a 5.95%, lo cual concuerda con lo esperado debido a que ningún equipo o componente utilizado es exacto ni ideal, tanto el datalogger como el Waspnote y los circuitos diseñados cuentan con un error asociado, pero se pueden tomar las mediciones como significativas por tener un porcentaje de error bajo.

En la figura 57, se observa el gráfico de los resultados con el datalogger y con el sistema, en éste se puede observar que el comportamiento de ambos fue el mismo, cuando la curva de la medición del datalogger aumentaba, la del sistema también y viceversa por lo que se corrobora que el sistema trabaja correctamente.

En cuanto a la calibración del sensor ST350, no es necesario calibrarlo antes de utilizarlo ya que viene calibrado de fábrica y como es totalmente sellado, la calibración manual es muy complicada de realizar y solo con equipo especializado. Por lo tanto, la empresa Campbell's Scientific, recomienda realizarle calibraciones periódicas cada 1 ó 2 años, esto con el fin de minimizar el mantenimiento que haya que darle al sensor, el cual es nulo.

En cuanto a la instalación, los pasos para ésta se encuentran en el apartado 5.3.1, en donde se debe ubicar el punto de mayor tensión en la viga y realizarle la instalación, la cual es muy sencilla.

6.2.2 Análisis de resultados para el sensor SR50A

En el caso del sensor ultrasónico para medir distancia SR50A, se implementó un circuito con un MAX232 y un PIC18F4550. Como este sensor es digital, se eligió su formato de salida RS-232 debido a que la mayoría de equipos soportan este formato y se conectó al MAX232 para convertirlo a salida TTL para poder ser interpretado por el PIC. El PIC se encarga de leer la trama, tomar el dato de la medición de distancia y enviarlo por puerto digital al Waspote.

Para comprobar el funcionamiento del sistema, se comparó el valor de la distancia mostrado en el Waspote, con el valor medido utilizando una cinta métrica. Los resultados se encuentran en la tabla 2. En esta se puede ver que se obtuvieron porcentajes de error menores a 4.45%, lo cual es muy bueno tomando en cuenta que el sensor tiene un error asociado de 0.25mm y que el error de medir con la cinta métrica es de 0.05mm, además del error de paralelaje típico en este tipo de mediciones. El único dato medido que obtuvo un porcentaje de error muy elevado, fue la medición en la que el blanco se colocó a una distancia menor a 50cm del sensor, este error fue de un 20%, lo que era de esperarse ya que según las especificaciones para este transductor, el blanco se debe ubicar a una distancia mayor a ésta.

En la figura 58, se puede ver el gráfico construido con la tabla 2, en donde las curvas siguen el mismo patrón y sus valores son muy cercanos, en algunos casos prácticamente iguales, lo que nos hace concluir que el sistema trabaja correctamente en la medición de la distancia del sensor a un blanco.

6.2.3 Análisis de resultados para la caracterización de los sensores

En cuanto a la caracterización de los sensores, para el sensor ST350 (StrainGage), se obtuvo un porcentaje muy alto de probabilidad de que los datos medidos con el sistema desarrollado, sean iguales a los datos medidos por el datalogger. Esto nos da la confianza de que el sistema funciona con una exactitud mayor al 95%, por lo que los datos son confiables.

Para el sensor ultrasónico SR50A, se obtuvo una probabilidad menor al 0,001 de que el valor promedio medido por el sistema sea igual al valor promedio medido manualmente con la cinta métrica.

Al analizar la razón de esta diferencia se pueden encontrar dos elementos que afectan la medición de la distancia. El primero es la precisión del sistema manual que es de 0.05 mm contra 0.25 mm del sensor. La segunda razón es la linealidad del sensor que es de 500 mm en adelante, por lo que se debe instalar para que la distancia de medición del cero corresponda a los 500 mm.

Otra razón que afecta el desempeño del sensor SR50A es su gran tiempo de adquisición (cerca de un minuto), esto puede afectar la toma de muestras en ambientes altamente dinámicos. Además, a las mediciones de este sensor, las pueden afectar cualquier objeto que refleje el sonido, por lo que en un laboratorio donde trabajan muchas otras personas, siempre existen otros objetos que lo reflejan y se dificulta obtener un medio controlado en este sentido.

6.2.4 Análisis de resultados para la red inalámbrica de sensores y el programa en la PC

Los resultados para la red inalámbrica de sensores, la recepción de datos en la PC y la creación del archivo donde se guardan los datos leídos, se muestran en el apartado 6.1.4.

En este se explica el procedimiento seguido para la comprobación de su funcionamiento y las imágenes de los resultados obtenidos. Así, en la figura 59, se observa un pantallazo de la consola de Linux, en donde se pueden ir observando los datos en la PC provenientes de los diferentes nodos. Se observa que se recibieron datos de los 3 nodos conectados a la red. Además, como el dato enviado por cada nodo es un número sucesivo, cuando el sensor de tensión superficial envía el número 1, los demás deberían enviar 11 y 21 y en esta figura se comprueba lo anterior.

Cada 15 segundos se realiza el cálculo del promedio de los datos recibidos, simulando lo que realizaría el sistema, en el cual cada minuto se calcula el promedio y se guarda el dato en el archivo. Si calculamos el promedio de 14 datos en donde hay una sucesión de 1 a 10, se obtiene:

$$prom = \frac{1+2+3+4+5+6+7+8+9+10+1+2+3+4}{14}$$

$$prom = 4.6428$$

En la figura 60, se puede ver el archivo .csv, en el cual se guardan los datos de los promedios calculados. En la primera columna se ve la hora a la cual se guardó el dato y en las otras columnas los datos. Se puede ver que en el caso de la tensión superficial, se tiene un promedio de 4.6428, que coincide con el promedio para una sucesión de sumas de 1 a 10. En algunos casos, el valor del promedio es mayor, esto se debe a que el primer dato para el cálculo del promedio no siempre es 1, puede que la sucesión comience en 5 y esto provoca que el promedio sea un poco mayor. Por lo tanto, se comprueba que el archivo se crea correctamente y que el cálculo de los promedios es correcto.

6.2.5 Análisis de resultados del sistema completo

En cuanto a los resultados obtenidos en el sistema completo, se observa en la sección 6.1.5, que los datos se recibieron satisfactoriamente en la PC, además, el archivo fue creado con la fecha del día en que se realizó la prueba y que al finalizar el tiempo de prueba elegido de 30 minutos, el archivo fue enviado y recibido con éxito por el servidor en otra PC.

En la consola de Linux, se puede observar cómo se van recibiendo los datos y al finalizar cada minuto, el cálculo del promedio de los datos para éste. Además, al finalizar el tiempo, muestra que envía el dato por FTP y el nombre del archivo que envía; si éste se envía satisfactoriamente, ahí termina el programa, sino muestra dónde se dio el error; ya que pudo haber sido porque no logró conectarse al servidor, o que no encontró la carpeta donde se pretendía guardar.

Capítulo 7. Conclusiones y recomendaciones

7.1 Conclusiones

1. Los sensores no necesitan una calibración previa.
2. Una vez instalados los sensores no requieren mantenimiento periódico.
3. El sistema desarrollado para la medición de la tensión superficial, presentó porcentajes de error menores al 5.77% con respecto a las mediciones realizadas con el datalogger CR1000.
4. Los posibles errores en las mediciones de la tensión superficial, se debe a que los equipos y componentes utilizados no son ideales.
5. El sistema desarrollado para el sensor ST350, cuenta con porcentajes de error bajos que muestran que el sistema trabaja correctamente.
6. El sistema desarrollado para la medición de la deflexión, obtuvo porcentajes de error menores al 4.45%, los cuales se deben a que los equipos utilizados no son ideales y a posibles errores de paralelaje en la medición.
7. Los errores en la medición de la deflexión pudieron deberse a que no se contaba con un medio controlado.
8. No existieron pérdidas de datos en la transmisión de la red inalámbrica de sensores.
9. El software realizado en la PC, recibe la información, la procesa y la guarda en un archivo .csv.
10. La recepción de datos y envío del archivo por FTP se realiza correctamente por el sistema.

7.2 Recomendaciones

1. Mejorar el sistema para la medición de la tensión superficial para lograr porcentajes de error menores al 5%.
2. Sustituir los circuitos desarrollados para la caracterización del sensor ST350 por un amplificador de instrumentación.
3. Sustituir la PC por un sistema embebido debido a que se debe colocar y dejar permanentemente debajo de un puente.
4. Diseñar cajas selladas para aislar los circuitos de las condiciones ambientales.
5. Diseñar un circuito capaz de tener como salida -5V a partir de una tensión de 5V.
6. Diseñar un circuito capaz de tener como salida 12V a partir de una tensión de 5V, para la alimentación del sensor SR50A.
7. Buscar una mejor alternativa para la medición de la deflexión, ya que un tiempo de muestreo de 1 minuto es muy alto para los requerimientos del sistema.

Bibliografía

1. Arqhys Architects Site. **Deflexión de sistemas estructurales**. En línea: <http://www.arqhys.com/construccion/deflexion-sistemas-estructurales.html>. Visitado el 14 de diciembre del 2010.
2. Autor desconocido. **Amplificadores Operacionales**. En línea: <http://www.lcardaba.com/articulos/opamps.html>. Visitado el 05 de octubre del 2011.
3. Autor desconocido. **El Protocolo FTP**. En línea: <http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/ftp.html>. Visitado el 10 de mayo del 2011.
4. Autor desconocido. **Python. Subir archivos a un FTP**. En línea: <http://nideaderedes.urlansoft.com/2008/09/03/python-subir-archivos-a-un-ftp/>. Visitado el 20 de junio del 2011.
5. Campbell Scientific, Inc. **CR1000 Measurement and Control System**. En línea: <http://www.campbellsci.com/documents/manuals/cr1000.pdf>. Visitado el 10 de setiembre 2011.
6. Campbell Scientific, Inc. **SR50A Sonic Ranging Sensor Instruction Manual**. En línea: <http://www.campbellsci.com/documents/manuals/sr50a.pdf>. Visitado el 20 de junio del 2011.
7. Campbell Scientific, Inc. **ST350 Strain Transducer Instruction Manual**. En línea: <http://www.campbellsci.com/documents/manuals/st350.pdf>. Visitado el 20 de junio del 2011.
8. eScience Group. **Modelo de predicción de fallas en puentes**. Instituto Tecnológico de Costa Rica.
9. Fernández, Marcos. **File Transfer Protocol**. En línea: <http://www.mitecnologico.com/Main/FtpProtocolo>. Visitador el: 10 de mayo del 2011.
10. González, Raúl. **Python para todos**. En línea: <http://mundogeek.net/tutorial-python/>. Visitado el 28 de mayo del 2011.
11. Huambachano, Max. **Resistencia de Materiales**. En línea: <http://resistenciadematerialesmjhm.blogspot.com/>. Visitado el 29 de agosto de 2011.
12. Libelium. **Waspote 802.15.4 Networking Guide**. En línea: http://www.libelium.com/documentation/waspote/waspote-802.15.4-networking_guide.pdf. Visitado el 25 de mayo del 2011.
13. Libelium. **Waspote Guía Técnica**. En línea: http://www.libelium.com/documentation/waspote/waspote-technical_guide_esp.pdf. Visitado el 10 de mayo del 2011.

14. MAXIM. **+5V Powered, Multichannel RS-232 Drivers/Receivers**. En línea: <http://www.datasheetcatalog.org/datasheet/maxim/MAX220-MAX249.pdf>. Visitado el 09 de agosto del 2011.
15. Microchip. **PICKIT 2 Microcontroller Programmer User's Guide**. En línea: <http://www.esat.kuleuven.be/cde/pickit2.pdf>. Visitado el 10 de agosto del 2011.
16. Microchip. **PIC18F2455/2550/4455/4550 Data Sheet**. En línea: <http://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf>. Visitado el 07 de agosto del 2011.
17. Navarro, Francisco. **Circuitos con retroalimentación resistiva**. Instituto Tecnológico de Costa Rica.
18. Texas Instruments. **LM124, LM224, LM324, LM2902 Low Power Quad Operational Amplifiers**. En línea: <http://www.ti.com/lit/ds/symlink/lm324-n.pdf>. Visitado el 06 de junio del 2011.

Apéndices

A.1 Glosario, abreviaturas y simbología

BaudRate: número de unidades de señal por segundo.

Datalogger: dispositivo electrónico para grabar datos de mediciones de diferentes parámetros.

Deflexión: deformación que sufren los elementos estructurales alargados como las vigas, en dirección perpendicular a su eje longitudinal al aplicarle una carga.

Microcontrolador: circuito integrado programable capaz de realizar las órdenes programadas en su memoria.

PCB: circuito impreso.

Protocolo FTP: protocolo de transferencia de archivos entre los clientes conectados a una red TCP.

Python: lenguaje de programación de alto nivel.

Linux: sistema operativo de software libre similar a Unix.

RS232: interfaz diseñada para el intercambio de una serie de datos binarios entre un transmisor y un receptor.

Tensión superficial: deformación en el material debido a la tensión generada por una carga.

Transductor: dispositivo capaz de transformar un tipo de entrada de energía en otro tipo diferente de salida.

Ultrasónico: onda sonora de frecuencia mayor a la audible por el oído humano.

A.2 Manuales de usuario

A.2.1 Manual de instalación del sensor ST350 [7]

Alineación

El sensor solo podrá medir tensión en el eje en que se encuentra alineado; por lo tanto, entre más exacta sea la alineación, más exacta será la medición. Para realizar la alineación, se debe marcar primero la línea central en el material tanto en la dirección transversal como en la longitudinal.

La marca en la dirección longitudinal debe ser de unos 20cm y la transversal de unos 10cm aproximadamente. En la marca transversal, se deben realizar 2 marcas adicionales a 3.81cm del centro. En la siguiente figura se pueden observar las marcas que se deben realizar. Las partes encerradas en círculos, son las superficies que son necesarias de preparar, ya que la preparación del material dependerá del material en que se colocará el sensor.

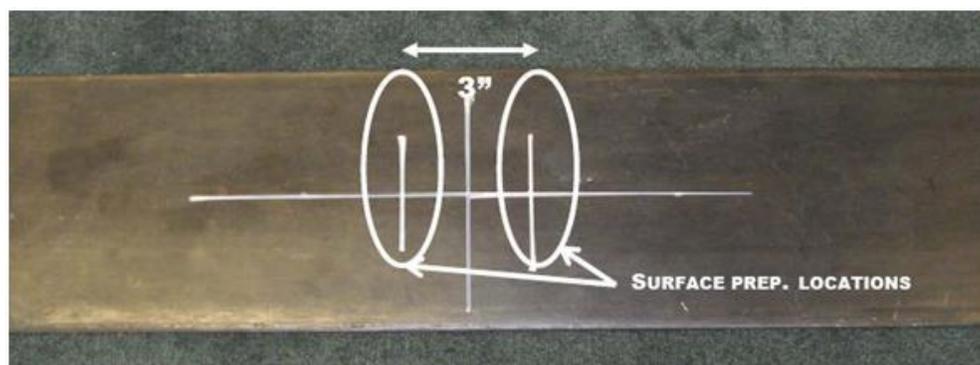


Figura A.2.1.1. Marcas para la alineación [7]

Instalación

Las dos marcas realizadas a 3.81cm del centro, se utilizan para montar el sensor longitudinalmente, alineando las marcas con el centro y los extremos del sensor. El transductor se debe alinear con el eje de medición para asegurarse que la tensión será medida paralelamente a este eje. Es importante que las líneas de marcas sean dibujadas cuidadosamente debido a que los errores en las tensiones son más susceptibles a aparecer cuando hay desalineamiento. Una vez alineado

el sensor, se colocan dos tornillos en cada uno de los huecos que coinciden con las líneas marcadas a 3.81cm del centro.

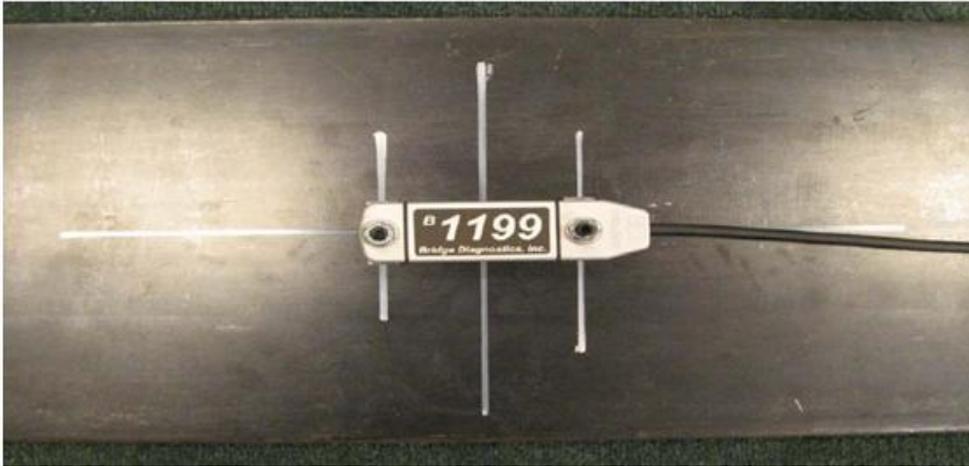


Figura A.2.1.2. Instalación del sensor SR50A [7]

Ajuste de un excesivo offset en el transductor

En algunos casos, el estado de cero estrés no se podrá lograr por sí solo y existirá un offset en la medición. Esto puede ser causado por un cambio en la temperatura de la superficie. Para corregir este offset, se debe girar la tuerca con que se sostiene el sensor a la superficie, para alinearlo al estado de “cero estrés”. Si aun así, no se logra este estado, compruebe que la superficie, en que está montada el sensor, sea plana y sino, prepare el material para lograr esta condición. Para lograr el estado de “cero estrés” se deben seguir los siguientes pasos:

- Determine la dirección del offset.
- Si se encuentre muy comprimado, libere el extremo opuesto donde se coloca el cable.
- Tire de este extremo del sensor y vuelva a colocar la tuerca suavemente hasta observar en el monitor el momento en que se acerque a cero.
- Si necesita más presión en la tuerca de la que se puede, retire también la tuerca del otro extremo y vuelva a colocar las 2 tuercas hasta obtener el estado cero.
- Si el offset se encuentra en la otra dirección, muy expandido, realice los pasos anteriores pero en lugar de tirar del extremo, empuje de este.

Si aun así no se logra el estado cero, devuelva el sensor, ya que podría estar dañado.

A.2.2 Manual de instalación del sensor SR50A [6]

Configuración inicial

Al inicio, se debe variar su configuración, ya que éste viene de fábrica configurado para que tenga una salida en el formato SDI-12. Pero moviendo 3 puentes dentro de él, se puede cambiar a RS232 que es con lo que trabaja el sistema desarrollado.

Para desarmar el sensor, se debe primero desconectar el cable del sensor. Luego, se deben remover los 6 tornillos de la cubierta del transductor y desconectar los cables.

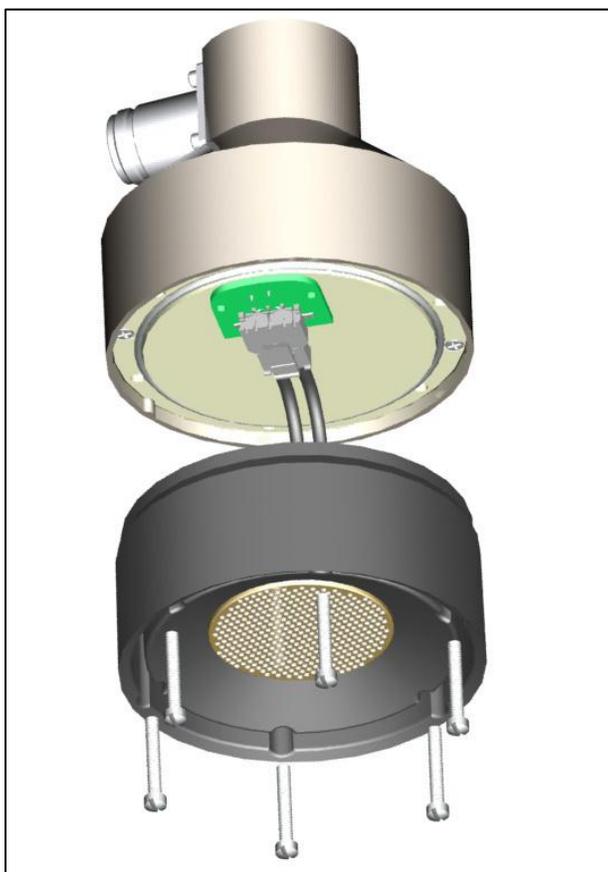


Figura A.2.2.1. Remover los tornillos de la cubierta del transductor SR50A y desconectar los cables [6]

Luego, cuenta con unos absorbentes de humedad los cuales se remueven, se quitan 2 tornillos Philips para poder exponer la PCB.



Figura A.2.2.2. Absorbentes de humedad del sensor SR50A [6]

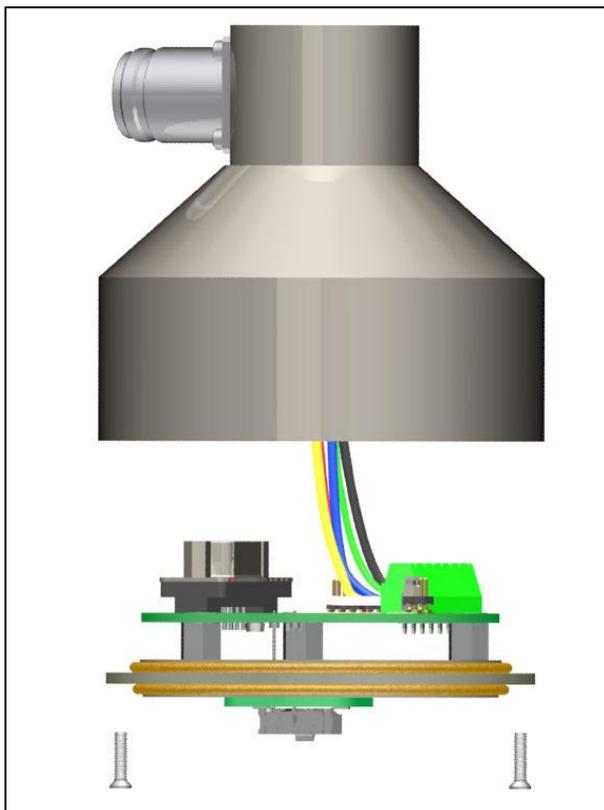


Figura A.2.2.3. Remover tornillos para exponer la PCB del SR50A [6]

Con esto, se puede mover los puentes ya sea para poder utilizar en formato RS-232, RS-485 o SDI-12. En este caso, los colocamos en la posición para el RS-232.

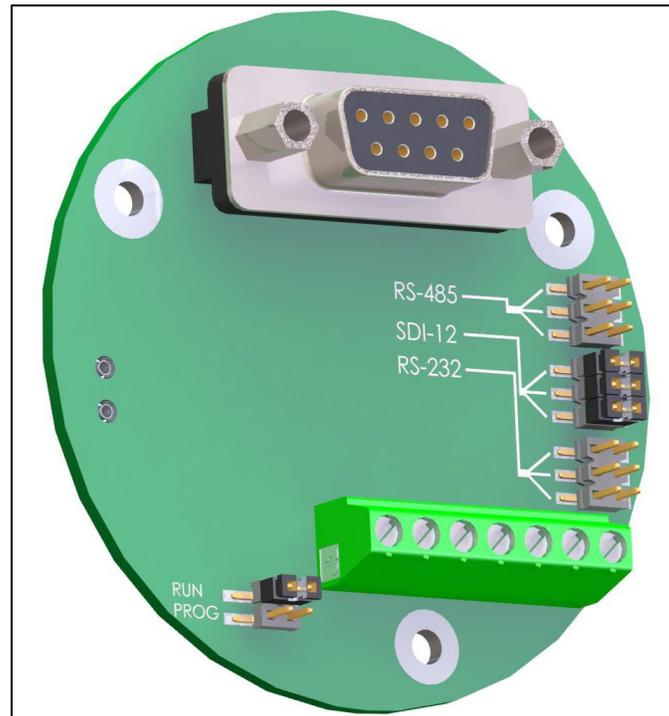


Figura A.2.2.4. PCB del SR50A [6]

Además, para cambiarle las configuraciones de software, se puede utilizar el programa Hyperterminal para Windows o cualquier otro programa usado para comunicaciones. Si se utiliza el programa Hyperterminal, se debe realizar lo siguiente:

- Iniciar la sesión en hyperterminal.
- Ingresar el comando "SETUP" y presionar la tecla Enter.
- Cambiar el dato a la configuración necesaria, ya sea el baud rate, el intervalo de medición, unidad de salida, entre otros.

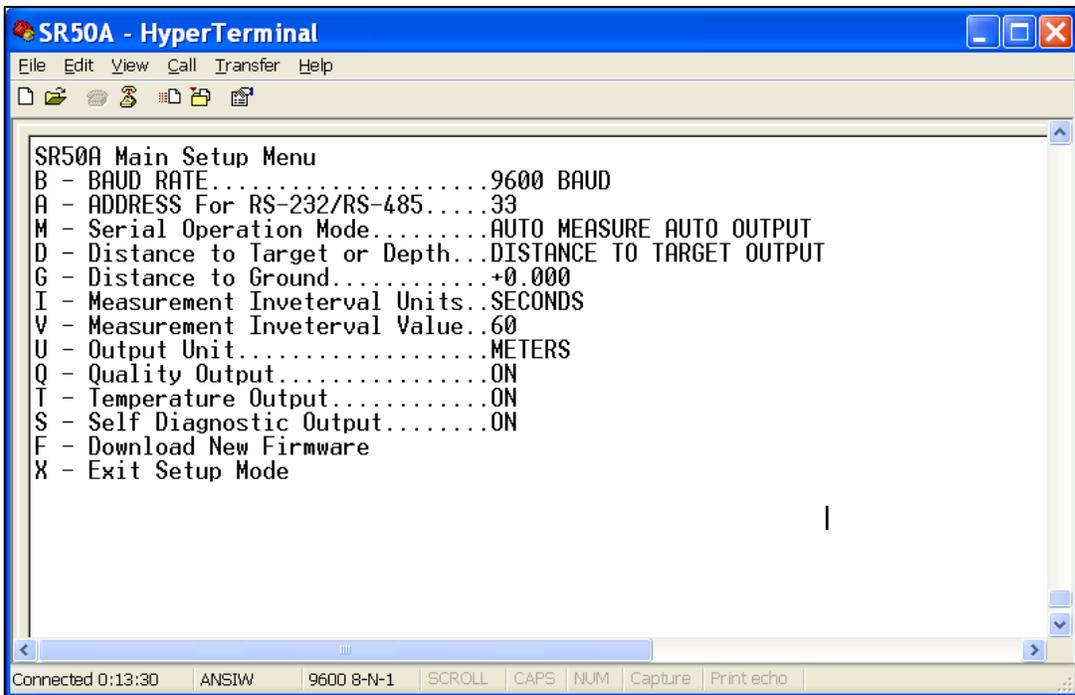


Figura A.2.2.5. Menú en el hyperterminal para la configuración del sensor SR50A [6]

Una vez configurados los diferentes datos a los valores requeridos, se debe ingresar la letra “X” para salir del modo de configuración y el sensor está listo para usarse.

Instalación

Para la instalación del sensor SR50A, se debe tomar en cuenta el ángulo de medición del sensor, dentro del cual, cualquier blanco que refleje sonido puede afectar la medición. Lo recomendable es colocar el blanco perpendicular al sensor, ya que cuenta con un ángulo de aproximadamente 30°. Lo que quiere decir que en un ángulo de 30°, no debe haber ningún objeto que pueda afectar la medición.

Para determinar el radio de medición, se utiliza una fórmula en donde se debe ingresar la distancia a la que se encuentra el blanco que colocamos para las mediciones.

$$CONO_{radio} = 0.268(CONO_{altura}) \quad (A.2.2.1)$$

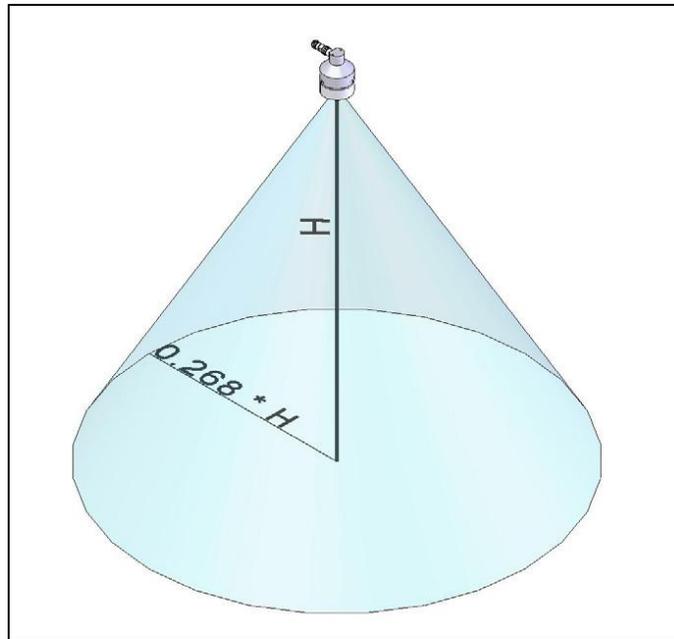


Figura A.2.2.6. Ángulo de medición del sensor [6]

El blanco colocado para la medición de distancia con el SR50A, debe estar ubicado al menos a 50cm de la cara del transductor y se debe cerciorar que no se encuentre demasiado largo ya que la máxima distancia que mide es de 9999mm.

Si se desea comparar la medición utilizando un metro, se deben agregar 8mm que es la distancia de la parrilla frontal del sensor, hasta la cubierta de plástico que se puede usar como referencia.



Figura A.2.2.7. Distancia de la parrilla frontal a la cubierta de plástico [6]

Para montar el sensor al lugar de medición, se debe colocar una barra de metal en donde se colocará el sensor. Existen 2 métodos:

- Usar el Kit de Montaje para este sensor: el número de la pieza es 19517, el cual se monta a la pieza CM206, la cual se puede observar en la figura 51.



Figura A.2.2.8. Montaje utilizando el Kit de montaje 19517 [6]

- Montarlo sobre un cilindro de montaje, que es la pieza número 19484 y ésta pieza, conectarla a un UN-RAIL (pieza número 1049).

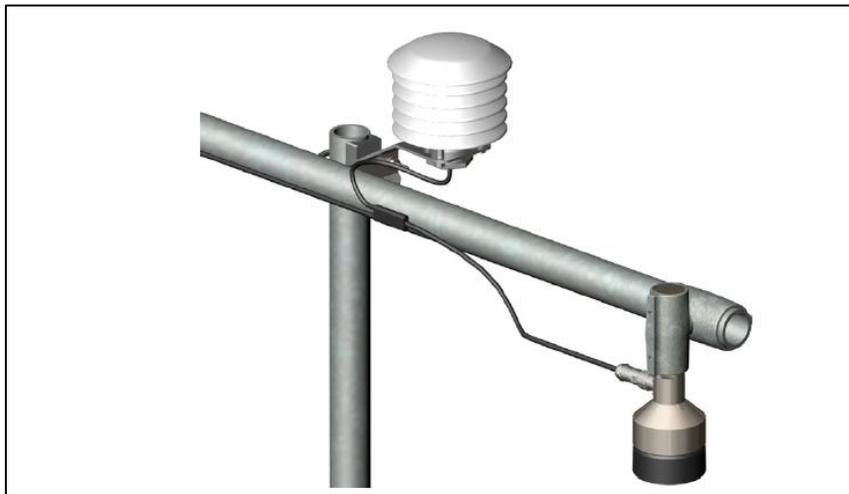


Figura A.2.2.8. Montaje utilizando un cilindro y el UN-RAIL [6]

A.2.3 Manual de mantenimiento para el sensor ST350

Debido a que éste sensor viene calibrado de fábrica y totalmente sellado, no existe un método manual para su calibración, su mantenimiento es casi nulo. Campbell's Scientific recomienda realizarle una recalibración cada 1 ó 2 años, o cada 15-25 instalaciones, en el caso de que se cambie de lugar constantemente. Esto con el fin, de minimizar la cantidad de mantenimiento requerido por el sensor.

Para esta recalibración, se debe mandar el sensor a la fábrica en Estados Unidos. Se recomienda que cada cierto tiempo, se revise que el valor de las resistencias entre los cables continúen siendo 150Ω.

A.2.4 Manual de mantenimiento del sensor SR50A

Al igual que el transductor ST350, el SR50A está diseñado para que su mantenimiento sea mínimo. Por esto, cuenta con los absorbentes de humedad que hacen que ésta no entre al sensor por los huecos de ventilación. Estos absorbentes cuentan con un gel que se mantiene de color azul mientras se encuentren en buen estado, pero si cambian a color rosado, deben ser remplazadas debido a que éstos se encuentran saturados.

Para revisarlos y remplazarlos, se debe seguir el mismo procedimiento visto en el manual de instalación para la configuración del sensor, pero en este caso, se llega hasta donde se encuentran los absorbentes, se retiran y se cambian.

La cubierta del transductor, se recomienda remplazarla cada 3 años, pero si se encuentra en un lugar con una humedad muy alta, se debe remplazar cada año.

A.2.5 Manual de usuario del sistema

Conexión del sistema para el sensor ST350

La PCB diseñada para el sensor ST350, consta de un circuito restador, junto con un circuito acondicionador de señales. Esta PCB se conecta a un nodo Waspote el cual se encargará de recibir el dato medido y enviarlo inalámbricamente a la PC.

Para programación del módulo Waspote en el cual se realizará la recepción de datos del sensor ST350 para la medición de la tensión superficial, se deben seguir los siguientes pasos.

1. Abrir el software Waspote IDE para la programación de los módulos.

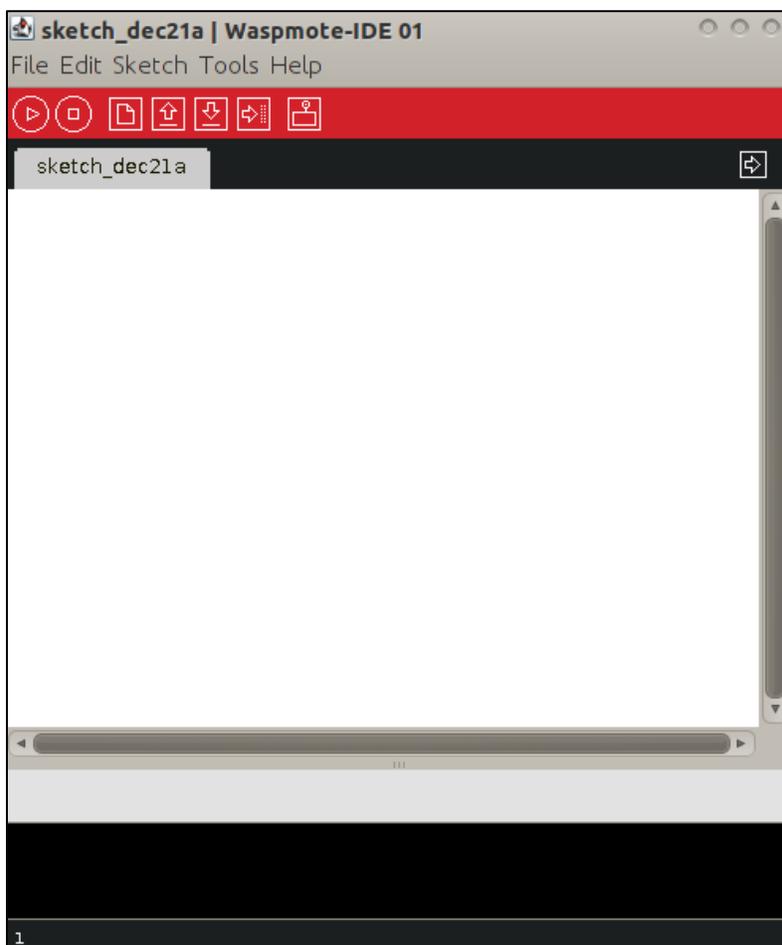


Figura A.2.5.1 Waspote IDE



```
XBee_TS | Wasp mote-IDE 01
File Edit Sketch Tools Help

XBee_TS $
//Inicializaciones
void setup()
{
  //Inicializando las librerias XBee
  xbee802.init(XBEE_802_15_4, FREQ2_4G, NORMAL);
  xbee802.ON();
  //Habilitacion de los 5V de alimentacion para los sensores
  PWR.setSensorPower(SENS_5V,SENS_ON);
  //USB.begin(); //Para realizar pruebas con el USB
  //USB.println("INICIO");
  //pinMode(DIGITAL1,INPUT); //Se utiliza para la comparacion con datalogger
}

//Programa Principal
void loop()
{
  //if(digitalRead(DIGITAL1)==1){ //Para las pruebas con el datalogger
  long(analogico)=analogRead(ANALOG1); //Lectura del sensor
  Utils.setLED(LED0, LED_ON);
  //USB.println(voltaje);
  //USB.println(sensor);
  sprintf(envio,"Tension superficial:\n%d\n",analogico); //Dato a enviar inalambri
  xbee802.send("0013A200403D6C81",envio); //Envio inalambrico
  Utils.setLED(LED0, LED_OFF);
  delay(1000);
  //}
}
//*****FIN DEL PROGRAMA*****
```

Figura A.2.5.3 Cambio del nombre que identifica el sensor en la red

4. Dar click en el botón “Verify” el cual compila el programa, este botón se señala con la flecha en la siguiente figura.

```
XBee_TS | Wasp mote-IDE 01
File Edit Sketch Tools Help
Verify
XBee_TS 5
//Inicializaciones
void setup()
{
  //Inicializando las librerias XBee
  xbee802.init(XBEE_802_15_4, FREQ2_4G, NORMAL);
  xbee802.ON();
  //Habilitacion de los 5V de alimentacion para los sensores
  PWR.setSensorPower(SENS_5V,SENS_ON);
  //USB.begin(); //Para realizar pruebas con el USB
  //USB.println("INICIO");
  //pinMode(DIGITAL1,INPUT); //Se utiliza para la comparacion con datalogger
}

//Programa Principal
void loop()
{
  //if(digitalRead(DIGITAL1)==1){ //Para las pruebas con el datalogger
  long(analogico)=analogRead(ANALOG1); //Lectura del sensor
  Utils.setLED(LED0, LED_ON);
  //USB.println(voltaje);
  //USB.println(sensor);
  sprintf(envio,"Tension superficial:\n%d\n",analogico); //Dato a enviar inalambri
  xbee802.send("0013A200403D6C81",envio); //Envio inalambri
  Utils.setLED(LED0, LED_OFF);
  delay(1000);
  //}
}
//*****FIN DEL PROGRAMA*****
32
```

Figura A.2.5.4 Compilación del programa

5. Si la compilación fue exitosa, aparecerá el mensaje de “Done compiling” en la parte de debajo de la ventana.



Figura A.2.5.5 Verificación de que la compilación fue exitosa.

6. Se debe conectar el WaspMote a la PC con el cable USB y encender el Switch ON/OFF. Se debe tener desconectado el módulo XBEE de transmisión inalámbrica.

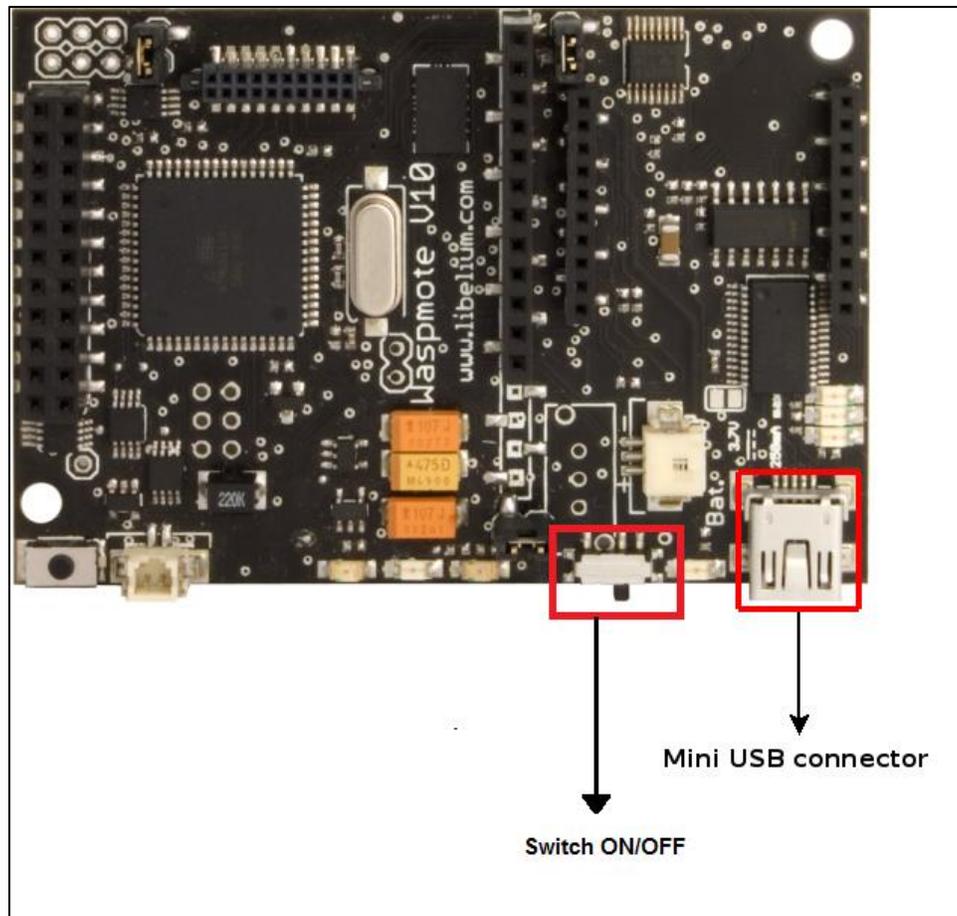


Figura A.2.5.6 Conector para el cable USB y el Switch de ON/OFF en el Waspote [13]

7. En el Waspote IDE, ir a la pestaña Tools, Serial Port y elegir el puerto en el que se encuentra conectado el módulo.
8. Para programar el módulo, dar click en el botón de Upload que se muestra en la siguiente figura señalado con la flecha.

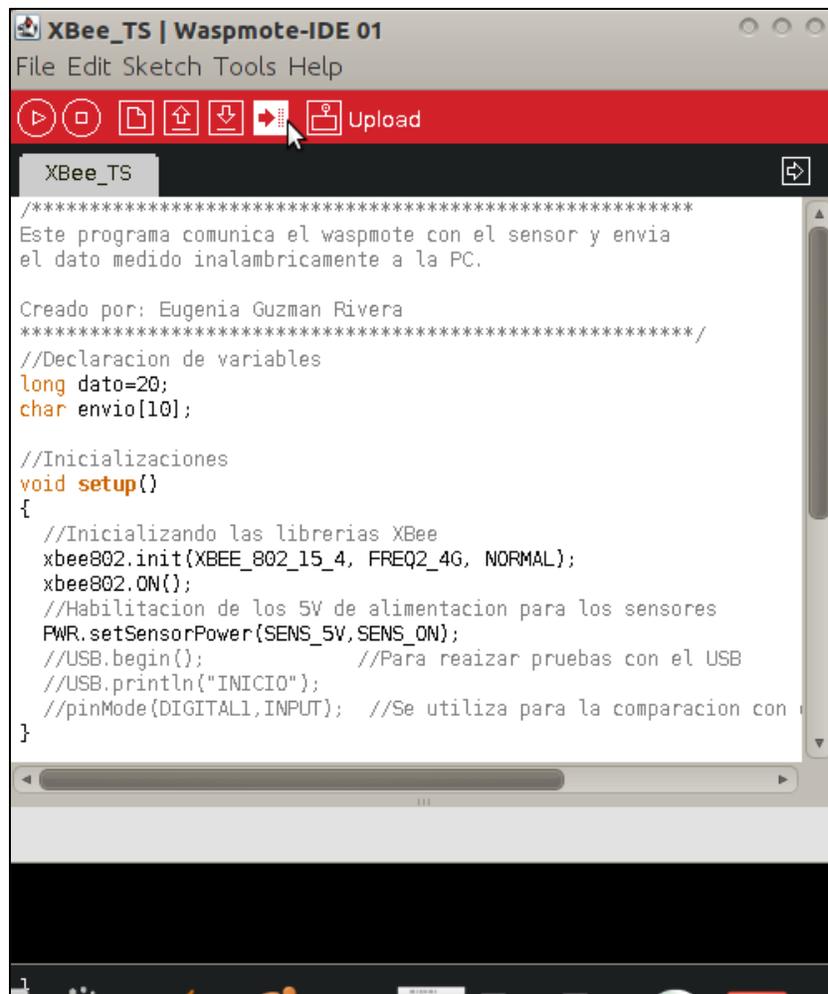


Figura A.2.5.7 Programación del módulo Waspote

9. Al igual que en el caso de la compilación, si la programación fue exitosa mostrará el mensaje “Done uploading”.
10. Apagar el Switch ON/OFF, desconectar el cable USB, conectar el módulo XBEE y la PCB al Waspote.

Conexión del sistema para el sensor SR50A

La PCB diseñada para el sensor SR50A, consta de un MAX232 que se encarga de transformar la trama enviada por el sensor en formato RS-232 a TTL y un PIC18F4550, el cual recibe la trama e identifica la el dato de la medición para enviarlo al Wasmote.

Para la programación del módulo Wasmote en el cual se realizará la recepción de datos del PIC para la medición de la deflexión, se deben seguir los siguientes pasos.

1. Abrir el software Wasmote IDE para la programación de los módulos, el cual se puede observar en la figura A.2.5.1.
2. Abrir el archivo XBEE_Deflexión.



Figura A.2.5.8 Programa para la lectura del sensor SR50A en el Waspote

3. Cambiar el nombre con que se quiere identificar el sensor dentro de la red. Se debe poner el nombre elegido en lugar de las palabras "Deflexion" mostradas en la siguiente imagen.



```
XBee_Deflexion | Wasp mote-IDE 01
File Edit Sketch Tools Help
XBee_Deflexion 5
  dato=dato+1;
}
if(n==1){
  dato1=dato;
  dato=0;
}
else if(n==2){
  dato2=dato;
  dato=0;
}
else if(n==3){
  dato3=dato;
  dato=0;
}
else if(n==4){
  dato4=dato;
  dato=dato1*1000+dato2*100+dato3*10+dato4; //Calculo de la medicion
  //USB.println(dato);
  sprintf(envio, "Deflexion: \n%d\n", dato); //Dato a enviar inalambicamente
  xbee002.send("0013A200403D6C81",envio); //Envio inalambicamente
  dato=0;
  n=0;
}
}
while(digitalRead(DIGITAL5)==1){
}
}
//*****FIN DEL PROGRAMA*****
```

Figura A.2.5.9 Cambio del nombre que identifica el sensor en la red

4. Dar click en el botón “Verify” el cual compila el programa, el cual se señala en la figura A.2.5.4.
5. Si la compilación fue exitosa, aparecerá el mensaje de “Done compiling” en la parte de debajo de la ventana.

6. Se debe conectar el Wasmote a la PC con el cable USB y encender el Switch ON/OFF. Se debe tener desconectado el módulo XBEE de transmisión inalámbrica.
7. En el Wasmote IDE, ir a la pestaña Tools, Serial Port y elegir el puerto en el que se encuentra conectado el módulo.
8. Para programar el módulo, dar click en el botón de Upload que se muestra en la siguiente figura A.2.5.7 señalado con la flecha.
9. Al igual que en el caso de la compilación, si la programación fue exitosa mostrará el mensaje "Done uploading".
10. Apagar el Switch ON/OFF, conectar el módulo XBEE y la PCB al Wasmote.
11. Con el cable USB conectado, encender el Wasmote y realizar una medición inicial para conocer la distancia a la que se encuentra el blanco y así, poder calcular la deflexión de la estructura. Para esto, abrir el Serial Monitor, el cual se señala en la siguiente figura.



Figura A.2.5.10 Serial Monitor del WaspMote IDE

12. En la pantalla del Serial Monitor, se observarán las mediciones realizadas por el sensor, se debe apuntar este dato porque se necesitará más adelante.
13. Desconectar el cable USB de la PC.

Configuraciones en la programación de la PC

Se deben realizar algunos cambios en la programación de la PC para lograr la recepción y el análisis correcto de los datos medidos. Para esto, se deben seguir los siguientes pasos:

1. Dar doble click al archivo Lectura_USB.py y en la ventana que se abre a continuación, elegir Display.

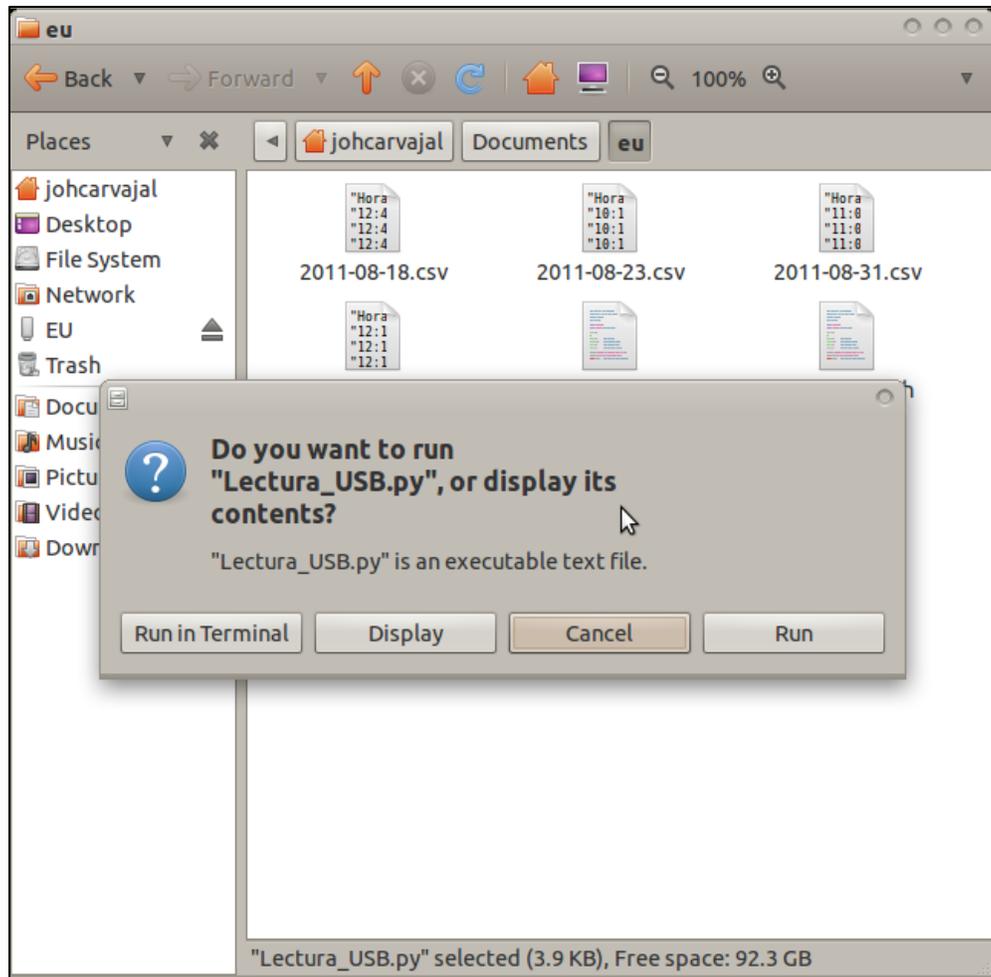


Figura A.2.5.11 Abrir archivo Lectura_USB.py

2. Conectar el Waspnote Gateway al puerto USB de la PC e identificar el puerto utilizado por éste.
3. En la configuración del puerto serie del programa en Python, cambiar el puerto que utiliza el Gateway.

```

#####
#El siguiente programa se encarga de la recepcion de datos de las mediciones realizadas a los sensores en la red
#inalambrica de sensores. Recibe los datos, los analiza y los guarda en un archivo para luego enviar este via FTP
#a un servidor.

#Creado por: Eugenia Guzman Rivera
#####

#####Librerias importadas#####
#!/usr/bin/env python
import sys
import os
import serial
import time
import csv
import ftplib
#####

#####Declaracion de variables#####
fecha=time.strftime('%Y-%m-%d')           #Configuracion del formato de la fecha
fecha_arc=fecha
ser=serial.Serial('/dev/ttyUSB1',38400)   #Configuracion del puerto serie
fecha_arc=fecha
y=0
deflexion=[0]
Lvdt=[0]
TS=[0]
prom_deflexion=0
prom_LVDT=0
prom_TS=0
Dist_inicial = 0
d=0
l=0
ts=0
i = 0
#####

```

Figura A.2.5.12 Configuración del puerto serie en el programa Lectura_USB.py

4. Se debe declarar la variable “Dist_inicial” con el valor medido para el sensor SR50A, el cual es la distancia inicial del sensor al blanco.

```

#####Declaracion de variables#####
fecha=time.strftime('%Y-%m-%d')           #Configuracion del formato de la fecha
fecha_arc=fecha
ser=serial.Serial('/dev/ttyUSB1',38400)   #Configuracion del puerto serie
fecha_arc=fecha
y=0
deflexion=[0]
Lvdt=[0]
TS=[0]
prom_deflexion=0
prom_LVDT=0
prom_TS=0
Dist_inicial = 0
d=0
l=0
ts=0
i = 0
#####

```

Figura A.2.5.13 Declaración de la variable de distancia inicial para el sensor SR50A

5. En la función FTP() se deben ingresar los datos del servidor al cual se enviará el archivo.

```
#####Envio del archivo por FTP#####
def FTP():
    print "enviar ftp"
    ftp_servidor = 'www.ie.itcr.ac.cr'           #Direccion del servidor
    ftp_usuario = 'jcarvajal'                   #Usuario
    ftp_clave = '12345678'                       #Contraseña
    ftp_raiz = '/web/ebridge'                   #Carpeta donde se guardara el archivo
    fichero_origen= fecha_arc+'.csv'             #Nombre del archivo a enviar
    print fichero_origen
    try:
        s=ftplib.FTP(ftp_servidor, ftp_usuario, ftp_clave)
        try:
            f=open(fichero_origen,'rb')
            s.cwd(ftp_raiz)
            s.storbinary('STOR ' + fichero_origen, f)
            f.close()
            s.quit()
        except:
            print "No se ha podido encontrar el fichero"
    except:
        print "No se ha podido conectar al servidor"
#####
```

Figura A.2.5.14 Datos del servidor al que se enviará el archivo .csv

6. En la función principal del programa, se deben cambiar las palabras “Deflexion” y “Tension superficial” por los identificadores definidos en la programación de los Waspote para cada sensor y en el caso de que existan más sensores, incluir más comparaciones para poder guardar los datos de cada sensor.

```

if t == "Deflexion:":
    print "DEFLEXION"
    if d==0:
        serie = ser.readline().strip()
        datol=serie
        serie = ser.readline().strip()
        dato2=serie
        dato=datol+dato2
        deflexion[d]=long(dato)
        print deflexion[d]
    else:
        serie = ser.readline().strip()
        datol=serie
        serie = ser.readline().strip()
        dato2=serie
        dato=datol+dato2
        deflexion.append(long(dato))
        print deflexion[d]
    d = d+1
    i=i+1

elif t == "Tension superficial:":
    print "TENSION SUPERFICIAL"
    if ts==0:
        serie = ser.readline().strip()
        TS[ts] = float(serie)
        print TS[ts]
    else:
        serie = ser.readline().strip()
        TS.append(float(serie))
        print TS[ts]
    ts = ts+1
    i=i+1

```

Figura A.2.5.15 Identificadores para cada sensor.

7. En la línea señalada en la siguiente figura, se debe ingresar la ecuación para el cálculo de la tensión en el material a partir de la curva de calibración para el sensor ST350.

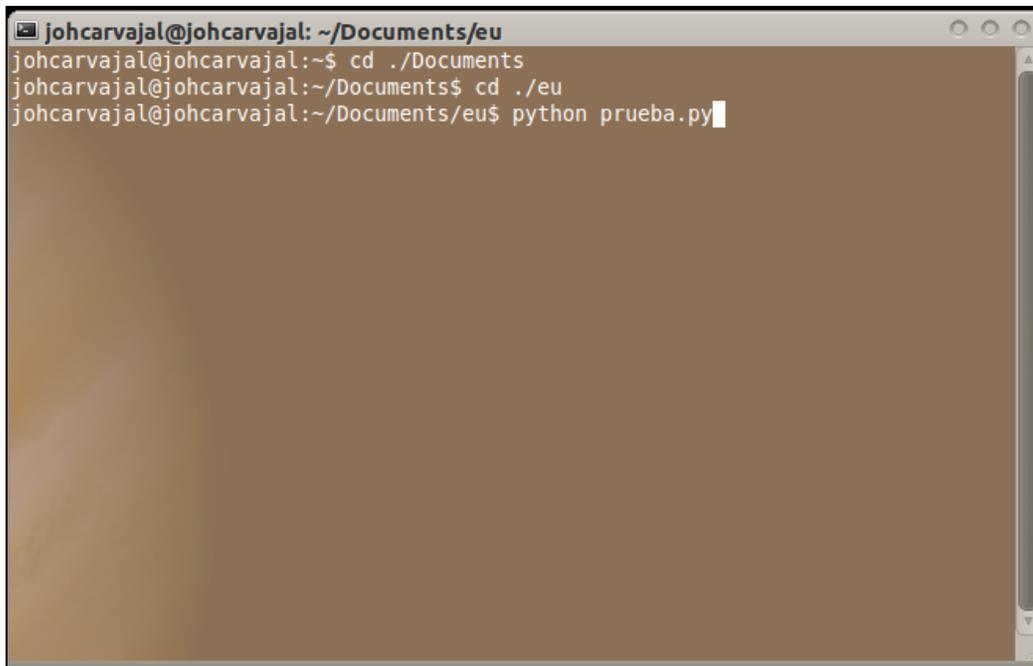
```

if(ts!=0):
    prom_TS = prom_TS/(ts)
    prom_TS = prom_TS * 3.3 / 1023
    prom_TS = 1000*((prom_TS - 1.6)/82.5)
    #prom TS = x #Ecuacion para el calculo de la tension en la superficie
print "promedio TS:"
print prom_TS

```

Figura A.2.5.16 Definición de la ecuación para el cálculo de la tensión en el material para el sensor ST350

8. Una vez realizados estos cambios, se deben guardar.
9. Abrir la consola de Linux.
10. Abrir la carpeta donde se encuentra guardado el programa.
11. Correr el programa ingresando el comando “python Lectura_USB.py”



```
johcarvajal@johcarvajal: ~/Documents/eu
johcarvajal@johcarvajal:~$ cd ./Documents
johcarvajal@johcarvajal:~/Documents$ cd ./eu
johcarvajal@johcarvajal:~/Documents/eu$ python prueba.py
```

Figura A.2.5.17 Comando para correr el programa en la consola de Linux

12. El programa empezará a correr y mostrará los datos recibidos en la consola de Linux, además de que se creará el archivo con la fecha en la misma carpeta donde se encuentra el programa.
13. Cuando finalice el día, se enviará el archivo por FTP y se creará un nuevo archivo donde se guardarán los datos del siguiente día.

Anexos

Anexo B.1 Hoja de datos de los módulos Waspote


Waspote - Datasheet

Waspote

Características Generales:

Microcontrolador: ATmega1281
 Frecuencia: 8MHz
 SRAM: 8KB
 EEPROM: 4KB
 FLASH: 128KB
 SD Card: 2GB
 Peso: 20gr
 Dimensiones: 73.5 x 51 x 13 mm
 Rango de Temperatura: [-20°C, +65°C]
 Reloj: RTC (32KHz)

Consumo:

ON: 9mA
 Sleep: 62µA
 Deep Sleep: 62µA
 Hiberate: 0,7µA

Funcionamiento sin recarga: 1año *

*Tiempo obtenido usando el modo Hiberate como modo de ahorro energético

Entradas/Salidas:

7 Analógicas (I), 8 Digitales (I/O), 1 PWM,
 2 UART, 1 I2C, 1 USB

Características Eléctricas:

Tensión de batería: 3.3 V - 4.2V
 Carga USB: 5 V - 100mA
 Carga placa solar: 6 - 12 V - 240mA
 Tensión batería auxilia: 3V

Sensores Integrados en la placa:

Temperatura (+/-): -40°C, +85°C. Precisión: 0.25°C
 Acelerómetro: ±2g (1024 LSB/g) / ±6g (340LSB/g)
 40Hz/160Hz/640Hz/2560Hz

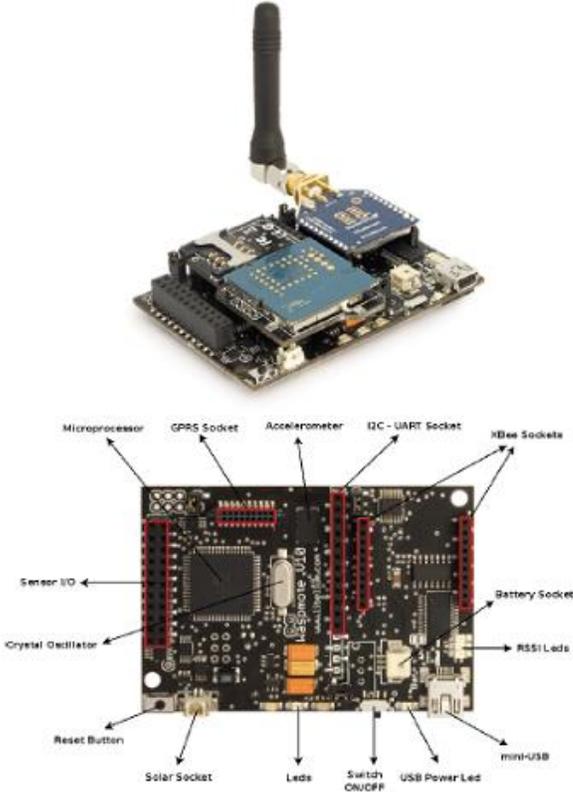


Figura 1: Cara Superior Waspote

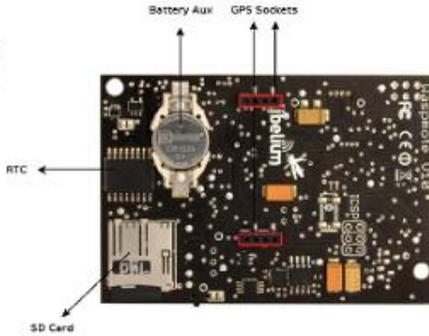


Figura 2: Cara Inferior Waspote

-2- <http://www.libelium.com/waspote> v0.9

Figura B.1.1 Hoja de datos del Waspote

802.15.4/ZigBee

Modelo	Protocolo	Frecuencia	txPower	Sensibilidad	Rango*
XBee-802.15.4	802.15.4	2.4GHz	1mW	-92dB	500m
XBee-802.15.4-Pro	802.15.4	2.4GHz	100mW	-100dBm	7000m
XBee-ZB	ZigBee-Pro	2.4GHz	2mW	-96dBm	500m
XBee-ZB-Pro	ZigBee-Pro	2.4GHz	50mW	-102dBm	7000m
XBee-868	RF	868MHz	315mW	-112dBm	12km
XBee-900	RF	900MHz	50mW	100dBm	10km
XBee-XSC	RF	900MHz	100mW	-106dBm	12km



Figura 3. XBee

* Línea vista y antena dipolo 9dB

Antenas: 2.4GHz: 2dBi / 5dBi
 868/900MHz: 0dBi / 4.5dBi
Conector: RPSMA
Cifrado: AES 128b
Control Señal: RSSI
Estándares: XBee-802.15.4 - 802.15.4 Compliant / XBee-ZB - ZigBee-Pro v2007 Compliant
Topologías: p2p, árbol, mesh

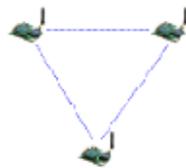


Figura 4: p2p

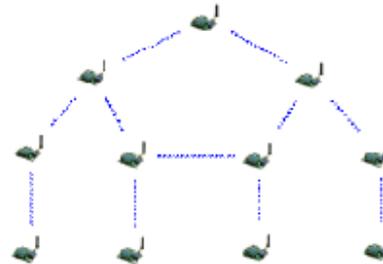


Figura 5: árbol

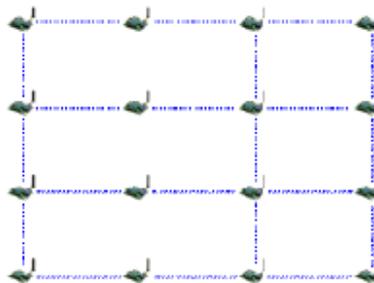


Figura 6: mesh

Figura B.1.2 Características generales de los módulos XBee

Anexo B.2 Sensor ST350

Section 2. Specifications

Effective gage length:	3.0 in (76.2 mm). Extensions available for use on R/C structures.
Overall Size:	4.375 in x 1.25 in x 0.5 in (111 mm x 32 mm x 13 mm).
Cable Length:	10 ft (3 m) standard, any length available.
Material:	Aluminum
Circuit:	Full wheatstone bridge with four active 350 Ω foil gages, 4-wire hookup.
Accuracy:	$\pm 2\%$, reading individually calibrated to NIST standards.
Strain Range:	Approximately $\pm 2000 \epsilon\mu$.
Force req'd for 1000 $\mu\epsilon$:	Approximately 17 lbs. (76 N).
Sensitivity:	Approximately <u>500 $\epsilon\mu$/mV/V</u> .
Weight:	Approximately 3 oz. (85 g).
Environmental:	Built-in protective cover, also water resistant.
Temperature Range:	-58°F to 185°F (-50°C to 85°C) operation range.
Cable:	BDI RC-187: 22 gage, two individually-shielded pairs w/drain.
Options:	Fully waterproofed, Heavy-duty cable, Special quick-lock connector (available upon request).
Attachment Methods:	C-clamps, threaded mounting tabs & quick-setting adhesive, wood screws, or concrete anchors.

Figura B.2.1 Especificaciones generales para el sensor ST350



Certificate of Calibration

Serial Number: B2465

Model: BDI ST350

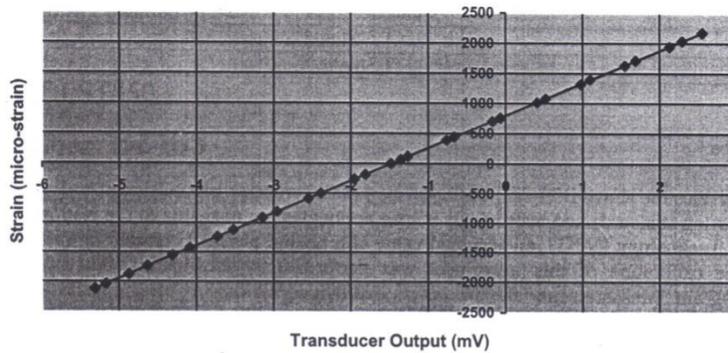
Initial Offset Voltage: -1.471 mV

General Gage Factor: 546.154 $\mu\epsilon/mV/V$

Table 1 – Representative Calibration Data

Applied Strain ($\mu\epsilon$)	Transducer Output (mV)	Applied Strain ($\mu\epsilon$)	Transducer Output (mV)
-10.636118	-1.477629	56.210930	-1.351819
-190.322804	-1.810163	380.924827	-0.756979
-507.950197	-2.385795	699.751146	-0.164539
-819.275764	-2.951250	1013.329939	0.413060
-1129.247243	-3.518373	1319.841321	0.979900
-1439.610250	-4.079178	1629.074334	1.549542
-1742.901000	-4.626937	1942.646603	2.129659
-2041.321033	-5.166300	2175.435011	2.555221
-2121.388861	-5.307320	2174.049374	2.553448
-2121.676421	-5.307645	2036.949613	2.294168
-1979.823368	-4.870326	1711.979200	1.692303
-1566.911629	-4.304715	1393.203136	1.100376
-1246.970471	-3.724762	1078.063613	0.523821
-922.477624	-3.137395	757.728963	-0.063010
-597.572664	-2.547011	437.232653	-0.658810
-269.647269	-1.949221	109.998838	-1.262248

Calibration Curve



Mean Linear Correlation Coefficient (LCC): 9.99986377E-1

LCC Standard Deviation: 2.530820E-7

Calibrated By: ACP

Signature: *[Handwritten Signature]*

Date/Time: 7/28/2010 10:14 AM

Temperature: 26.642

Figura B.2.2 Calibración del sensor ST350 con número de serie B2465

Specifications

BDI Automated Strain Transducer Calibration System (BDI-ASTCS)

BDI-ASTCS Calibration Information	
BDI-ASTCS Serial Number:	BDI CAL 0001
BDI-ASTCS Software Version:	1.0
BDI-ASTCS Independent Verification Date:	1-1-2010
Strain Transducer Gage Length:	3.0 inches (762 mm)
Applied Full Scale Displacement Range:	±0.006 inches (±0.015 mm)
Method for Applying Displacement:	Precision Step Motor Coupled to Linear Stage
Excitation Voltage for Calibration:	2.5 VDC
Displacement Measurements:	Dual Precision AC LVDT's, Output Averaged
Displacement Certification:	NIST Certificate #729970 & #821/274437-07
Linearity Verification Technique:	Linear Correlation Coefficient > 0.9999
Repeatability Verification Technique:	Standard Deviation < 0.5 % (of mean)
BDI-ASTCS System Check	
BDI Reference Strain Transducer:	B2296
BDI Reference General Gage Factor:	488.030 $\mu\epsilon/mV/V$
LVDT #1 Sensitivity:	5.556362E-3
LVDT #2 Sensitivity:	5.606809E-3
Date/Time of Last System Check:	7/28/2010 9:55 AM
BDI Strain Transducer Connections	
Red:	+ Excitation
Black:	- Excitation
Green:	+ Signal
White:	- Signal
Grey:	Shield

NIST Reference:

BDI certifies the above BDI-ASTCS instrument meets or exceeds published specifications and has been verified using standards and instruments whose accuracies are traceable to the National Institute of Standards and Technology (NIST), an accepted value of a natural physical constant or a ratio calibration technique. The calibration of this instrument was performed in accordance with the BDI Quality Assurance program. Measurements and information provided on this report are valid at the time of calibration only.

Note: For proper use of the General Gage Factor if using a data acquisition system other than a BDI Structural Testing System (BDI-STS or STS-WiFi), please refer to the BDI document entitled "Applying Calibration Factors for BDI Strain Transducers" available at www.bridgetest.com.

Figura B.2.3 Especificaciones sobre la calibración del sensor ST350

Anexo B.3 Sensor SR50A

SR50A Sonic Ranging Sensor	
1. Specifications	
Power Requirements:	9 to 18 Vdc
Power Consumption:	Quiescent SDI-12 Mode < 1.0 mA RS-232/RS485 Mode < 2.25 mA ^{Note1}
	Peak Measurement Current 250 mA typical
Measurement Time:	Less than 1.0 second
Selectable Outputs:	SDI-12 (version 1.3) RS-232 (1200 to 38400 bps) RS-485 (1200 to 38400 bps)
Measurement Range:	1.6 to 32.8 ft (0.5 to 10 m)
Accuracy:	±0.4 in (±1 cm) or 0.4% of distance to target, whichever is greater. Accuracy specification excludes errors in the temperature compensation.
Resolution:	0.01 in (0.25 mm)
Required Beam Angle Clearance:	30°
Operating Temperature:	-45° to +50°C
Maximum Cable Length:	SDI-12 196.9 ft (60 m) RS-232 (9600 bps or less) 98.4 ft (30 m) RS-485 984.3 ft (300 m) ^{Note2}
Cable Type:	4 conductor, 2-twisted pair, 22 AWG, Santoprene jacket
Dimensions:	Length 4 in (10.1 cm) Diameter 3 in (7.6 cm)
Weight:	Sensor only 0.88 lbs (0.4 kg) Cable (SR50A) 15 feet 0.55 lbs (0.25 kg)
NOTES	1: The quiescent current draw is less than 1.25 mA if the baud rate is 9600 bps or less. 2: Power supply must not drop below 11.0 Volts or heavier gauge wire is required.

Figura B.3.1 Especificaciones generales del sensor SR50A