



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática

Escuela Profesional de Ingeniería de Sistemas

**Implementación de una plataforma hardware/software
de una red de sensores inalámbrica para un entorno
inteligente**

TESIS

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

Gelber Christian USCUCHAGUA FLORES

ASESOR

Lic. Norberto Ulises ROMÁN CONCHA

Lima, Perú

2020



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Uscuchagua, G. (2020). *Implementación de una plataforma hardware/software de una red de sensores inalámbrica para un entorno inteligente*. Tesis para optar el título de Ingeniero de Sistemas. Escuela Profesional de Ingeniería de Sistemas, Facultad de Ingeniería de Sistemas e Informática, Universidad Nacional Mayor de San Marcos, Lima, Perú.

HOJA DE METADATOS COMPLEMENTARIOS

Código ORCID del autor	“—“
DNI o pasaporte del autor	10645375
Código ORCID del asesor	0000-0002-3302-7539
DNI o pasaporte del asesor	08510560
Grupo de investigación	“—“
Agencia financiadora	
Ubicación geográfica donde se desarrolló la investigación	Peru, Lima, Lima, San Juan de Miraflores, jr Lorenzo Lozano N° 928 Latitud: -12.156277 Longitud: -76.970968
Disciplinas OCDE	Ingeniería de sistemas y comunicaciones http://purl.org/pe-repo/ocde/ford#2.02.04



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA
Escuela Profesional de Ingeniería de Sistemas

Acta de Sustentación de Tesis

Siendo las 19:45 horas del día 02 de marzo del año 2020 se reunieron los docentes designados como miembros de Jurado de Tesis, presidido por el Mg. Erwin Mac Dowall Reynoso (Presidente), el Ing. Raúl Marcelo Armas Calderón (Miembro) y el Lic. Norberto Ulises Román Concha (Miembro Asesor) para la sustentación de la Tesis Intitulada: **“IMPLEMENTACIÓN DE UNA PLATAFORMA HADWARE/SOFTWARE DE UNA RED DE SENSORES INALAMBRICA PARA UN ENTORNO INTELIGENTE”**, del Bachiller: **Gelber Christian Uscuchagua Flores**; para obtener el Título Profesional de Ingeniero de Sistemas.

Acto seguido de la exposición de la Tesis, el Presidente invitó al Bachiller a dar las respuestas a las preguntas establecidas por los Miembros del Jurado.

El Bachiller, en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el bachiller obtuvo la nota de.....16..... (En letras)...DIECISEIS

A continuación el Presidente del Jurado Mg. Erwin Mac Dowall Reynoso, declara al Bachiller **Ingeniero de Sistemas**.

Siendo las 20:45 horas, se levantó la sesión.

.....
Presidente

Mg. Erwin Mac Dowall Reynoso

.....
Miembro

Ing. Raúl Marcelo Armas Calderón

.....
Miembro Asesor

Lic. Norberto Ulises Román Concha

Dedicatoria

Quiero dedicar este trabajo de tesis a mi padre que me ha guiado siempre y desde el cielo celebrará conmigo, a mi madre que siempre está presente en mi vida animándome a que no me rinda nunca, a mis hermanos que trato de ser un ejemplo para ellos, a Mabel la mujer que me inspira a superarme, que siempre me apoya en los momentos difíciles, madre de mis hijos y mi futura esposa, mi hija Cristina que sólo el hecho de mirarla hace que me devuelva las ganas de luchar en la vida y mi hija Isabel que pronto nacerá y será quien me inspire a valorar más la familia.

Agradecimientos

En esta oportunidad quiero agradecer especialmente a mi tutor y amigo Ulises Román Concha que, lo mejor de realizar esta tesis es haber estrechado lazos de amistad y que hayamos podido tener muchos momentos académicos y momentos de amistad en el desarrollo de esta tesis.

RESUMEN

Autor: Gelber Christian Uscuchagua Flores

Fecha: Marzo del 2020

En el presente se pueden implementar múltiples dispositivos capaz de monitorizar datos ambientales (temperatura, luminosidad, humedad, etc.) utilizando por ejemplo sistemas SCADA, pero el tener una plataforma para un entorno inteligente todos los componentes que integran esta red de sensores tienen que albergar un sistema operativo para tener la capacidad de procesamiento, realizando tareas fáciles o aprendidas. Se quiere además que los componentes tengan determinadas características como una comunicación inalámbrica, que sean de tamaño pequeño, capacidad de relación con otras redes y de bajo costo.

Vemos en tesis relacionadas a esta, que han hecho esfuerzos en conseguir una red de sensores con las características descritas, pero que no han podido lograrlo en todos los componentes. Se han implementado redes de sensores utilizando la PC como base estación, componente que recibe todos los datos captados del ambiente por parte de los nodos sensores, generando dificultad en el despliegue de los dispositivos en lugares accidentados, encareciendo el costo excesivamente y además que no resuelve la movilidad por no tener la capacidad de relacionarse con redes como GSM/GRPS.

Por consiguiente, la presente tesis tiene como propósito implementar una red de sensores inalámbricos a través de una plataforma de hardware/software, utilizando nodos sensor, nodo sumidero y un SBC o mini computadora de bajo recursos como estación base que estará conectado al nodo sumidero, que ejercerá de puerta de enlace para la recolección de datos de los nodos sensor. En el análisis, se determinó la viabilidad técnica, la definición del modelo lógico y físico, y la implementación de la plataforma.

Palabras clave: Sensores, Hardware, Software, entorno inteligente, SBC, IoT.

ABSTRACT

Author: Gelber Christian Uscuchagua Flores

Date: March 2020

Currently, multiple sensor networks can be implemented capable of monitoring environmental data (temperature, brightness, humidity, etc.) using for example SCADA systems, but having a platform for an intelligent environment all the components that make up this sensor network have that house an operating system to have the processing capacity, performing easy or learned tasks. It is also wanted that the components have certain characteristics such as wireless communication, that they are small in size, capacity to be related to other networks and of low cost.

We see in theses related to this, that they have made efforts to achieve a network of sensors with the described characteristics, but that they have not been able to achieve it in all the components. Sensor networks have been implemented using the PC as a base station, a component that receives all the data collected from the environment by the sensor nodes, generating difficulty in the deployment of the devices in rugged places, making the cost more expensive and also that it does not solve the mobility for not having the ability to interact with networks such as GSM / GRPS.

Therefore, this thesis aims to implement a wireless sensor network through a hardware / software platform, using sensor nodes, sink node and a low-resource SBC or mini computer as a base station that will be connected to the sink node, which will act as a gateway for the collection of data from the sensor nodes. In the analysis, the technical feasibility, the definition of the logical and physical model, and the implementation of the platform were determined.

Keywords: Sensors, Hardware, Software, intelligent environment, SBC. IoT

Índice General

ÍNDICE GENERAL	VIII
FIGURAS	XI
ILUSTRACIONES	XII
TABLAS	XIII
CAPITULO I	1
1.1 INTRODUCCION.....	1
1.2 ANTECEDENTES.....	2
1.2.1 DEL PROBLEMA.....	2
1.3 DEFINICIÓN DEL PROBLEMA.....	4
1.4 JUSTIFICACIÓN.....	4
1.5 OBJETIVOS.....	5
1.5.1 OBJETIVO GENERAL.....	5
1.5.2 OBJETIVOS ESPECÍFICOS.....	5
1.6 ALCANCES Y LIMITACIONES.....	6
1.7 PROPUESTA DE LA TESIS.....	7
CAPITULO II	8
2.1 REDES DE SENSORES - WSN.....	8
2.1.1 ELEMENTOS.....	9
2.2 SOFTWARE DE LAS REDES DE SENSORES.....	11
2.2.1 SISTEMA OPERATIVO DE LOS NODOS.....	11
2.2.2 LENGUAJE PROGRAMACIÓN DE LOS NODOS.....	12
2.3 HARDWARE DE LAS REDES DE SENSORES.....	13
2.3.1 NODO SENSOR.....	13
2.3.2 NODO SUMIDERO.....	18
2.3.3 ESTACION BASE – SBC (Single-Board Computer).....	20
CAPITULO III	25
3.1 IMPLEMENTACION DE ARQUITECTURAS ALTERNATIVAS.....	26
3.1.1 IMPLEMENTACIÓN DE UNA ESTACIÓN BASE PARA UNA RED INALÁMBRICA DE SENSORES DE LARGO ALCANCE CON CONEXIÓN A LA NUBE.....	26

3.1.2	IMPLEMENTACION DE UN PROTOTIPO DE WSN CON NODOS INTELIGENTES PARA EL SISTEMA DE RIEGO APLICADO A LA AGRICULTURA DE PRECISION	27
3.1.3	DISEÑO E IMPLEMENTACION DE UNA RED DE SENSORES INLAMBRICOS PARA EL MONITOREO DE FACTORES AMBIENTALES ABIOTICOS QUE AFECTAN AL CULTIVO DE SOLANUM PHUREJA.....	29
CAPITULO IV		31
4.1	SELECCIÓN DE LOS COMPONENTES PARA LA PLATAFORMA	31
4.1.1	SELECCIÓN DEL NODO PARA LA IMPLEMENTACION DE LA ARQUITECTURA PROPUESTA	31
4.1.2	SELECCIÓN DE LA PLACA SENSORA PARA EL NODO SENSOR	42
4.1.3	SELECCIÓN DE LA PLACA PUERTA DE ENLACE PARA EL NODO SUMIDERO.....	43
4.1.4	SELECCIÓN DE LA SBC PARA LA IMPLEMENTACION DE LA ARQUITECTURA PROPUESTA	45
4.1.5	SELECCIÓN DE LA MEMORIA FLASH PARA LA INSTALACION DE GNU-LINUX EN EL SBC	49
4.1.6	SELECCIÓN DE LA INTERFAZ TTL PARA INTERCONEXION DEL SBC CON UNA PC.....	51
4.2	SELECCION DEL SISTEMA OPERATIVO PARA LA PLATAFORMA.....	54
4.2.1	SISTEMA OPERATIVO EN LA SBC.....	54
4.2.2	SISTEMA OPERATIVO EN LOS NODOS.....	56
4.3	SOLUCIÓN	58
4.4	METODOLOGÍA	60
CAPITULO V		64
5.1	MODELO DE LA PLATAFORMA.....	64
5.1.1	ESTRUCTURA DEL MODELO LOGICO.....	64
5.1.2	ESTRUCTURA DEL MODELO FISICO.....	66
5.1.3	DESCRIPCIÓN FUNCIONAL DE LOS COMPONENTES.....	69
5.2	CONFIGURACION DEL SISTEMA OPERATIVO GNU-LINUX EN EL SBC.....	72
5.2.1	FORMATEAR EL SD CARD	72
5.2.2	GRABAR GNU-LINUX DEBIAN AL SD CARD	74
5.2.3	INSTALAR LA IMAGEN GNU-LINUX DEBIAN EN EL ORANGE PI.....	75
5.3	OPERATIVIDAD DE LA SBC PARA LA COMUNICACIÓN CON EL NODO SUMIDERO.....	81
5.3.1	CONFIGURACION WIFI EN ORANGE PI 2G-IOT.....	81
5.3.2	INSTALACION JAVA EN ORANGE PI 2G-IOT	84
5.3.3	INSTALACIÓN DE TINYOS EN ORANGE PI 2G-IOT	85
5.4	CONFIGURACION Y OPERATIVIDAD EN LOS NODOS DE LA RED.....	86
5.4.1	COMPILACION DEL SOFTWARE EN EL NODO SUMIDERO	88
5.4.2	COMPILACION DEL SOFTWARE EN EL NODO SENSOR	88

5.5	INTEGRACIÓN DE LOS COMPONENTES	89
5.5.1	APLICACIONES EN EL NODO SUMIDERO Y NODO SENSOR	90
5.5.2	PROGRAMAS PARA CAPTAR INFORMACION	91
5.5.3	INICIAR EL MONITOREO	91
CAPITULO VI	93
6.1	RESULTADO DE LOS INDICADORES	93
6.1.1	UBICACIÓN DE LOS NODOS.....	93
6.1.2	MEDICIONES DE TEMPERATURA.....	96
CONCLUSIONES Y RECOMENDACIONES	98
7.1	CONCLUSIONES	98
7.2	RECOMENDACIONES	99
REFERENCIAS BIBLIOGRÁFICAS	100
ANEXOS	102

Figuras

Figura 1 - Nodo MicaZ	15
Figura 2 -Nodo BTNODE3	16
Figura 3 - Nodo TelosB	16
Figura 4- Nodo TINYNODE	17
Figura 5- Nodo Iris	18
Figura 6 Uso del Arduino Due como procesador central de la estación base	27
Figura 7 Arquitectura general del sistema inalámbrico.....	28
Figura 8 Arquitectura general del sistema inalámbrico.....	30
Figura 9 Interface I2C	39
Figura 10 Interfaz SPI.....	39
Figura 11 Interfaz GPIO	40
Figura 12 Placa MTS310CB	43
Figura 13 Placa MIB510 I	44
Figura 14 Placa MIB510 II	44
Figura 15 Arquitectura Orange Pi y Raspberry	48
Figura 16 PI230HX	52
Figura 17 CH340G.....	53
Figura 18 – Los cuatro componentes de la plataforma	71
Figura 19 Formatear microSD.....	73
Figura 20 Grabación de imagen Debian	75
Figura 21 Instalación de Putty	76
Figura 22 Ingreso de tarjeta microSD en el Orange Pi.....	77
Figura 23 Conexión TTL con el Orange Pi	78
Figura 24 TTL conectado a la PC.....	78
Figura 25 Realizamos la conexión de Putty con la PC.....	79
Figura 26 Instalación Debian realizada en el Orange Pi.....	80
Figura 27 Solicitando datos para la instalación del wifi en el Orange Pi.....	81
Figura 28 Instalando wifi en el Orange Pi.....	82
Figura 29 wifi instalado en el Orange Pi	83
Figura 30 Verificando wifi.....	83
Figura 31 Actualizando librerías para instalación de Java	84
Figura 32 Iniciando instalación de TinyOS	85

Ilustraciones

Ilustración 1 – Descripción general de la red de sensores	9
Ilustración 2 -Elemento de la WSN - Nodo final	10
Ilustración 3 -Elemento de la WSN - Nodo final	10
Ilustración 4 -Partes de un nodo sensor	14
Ilustración 5 -Partes de un nodo sumidero	19
Ilustración 6 - Partes de una memoria SRAM.....	36
Ilustración 7 - Puerto paralelo y serial de un UART	38
Ilustración 8 - Solución Propuesta	58
Ilustración 9 – Modelo lógico del nodo sensor	65
Ilustración 10 – Modelo lógico del nodo sumidero	66
Ilustración 11 – Modelo físico del nodo sensor.....	67
Ilustración 12 – Modelo físico del nodo sumidero	67
Ilustración 13 – Modelo físico del SBC	68
Ilustración 14 Diagrama de Secuencia de los componentes	69
Ilustración 15 – Topología de la red de sensores	89
Ilustración 16 área de prueba	94
Ilustración 17 área de prueba	95

Tablas

Tabla 1- Cuadro de tipos de sensores	15
Tabla 2 - Cuadro comparativo de familias ARM	23
Tabla 3 - Investigaciones	25
Tabla 4 - Cuadro comparativo de plataformas investigadas	30
Tabla 5 – Cuadro comparativo del consumo de voltaje de los microcontroladores	33
Tabla 6 - Cuadro comparativo del convertor analógico digital en los nodos	34
Tabla 7 - Cuadro comparativo del consumo de voltaje de los radios	35
Tabla 8 - Cuadro comparativo de tipos de memoria en los nodos	37
Tabla 9 - Cuadro comparativo de las interfaces en los nodos	40
Tabla 10 - Puntajes Metodologías	41
Tabla 11 - Cuadro de evaluación de los nodos	41
Tabla 12 - Cuadro de evaluación de las familias de SBC.....	46
Tabla 13 - Cuadro resumen comparativo SBC - Orange PI 2G IoT vs Raspberry PI Zero W	48
Tabla 14 - (Oficial Debian, 2018)	55
Tabla 15 - Cuadro con los núcleos de la familia Cortex A.....	56
Tabla 16- Cuadro con los núcleos de la familia Cortex A.....	57
Tabla 17 Descripción de las funcionalidades de los componentes.....	70
Tabla 18 - Tabla de conexión TTL	77
Tabla 19 – Identificación de los nodos	93
Tabla 20 - Resultado de los indicadores	96
Tabla 21 – Comportamiento de los parámetros de temperatura	97

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 INTRODUCCION

En la última década para el monitoreo de datos ambientales se han ido desarrollando y mejorando tecnologías inalámbricas que han ayudado a desarrollar diferentes soluciones tecnológicas sin necesidad de utilizar algún tipo de cable, favoreciendo una tecnología ubicua y móvil. Estas implementaciones tienen como objetivo obtener cualquier tipo de información como la temperatura, humedad, luminosidad, etc., del medio que los rodea.

Una WSN dependiendo de su arquitectura está constituida por dispositivos pequeños, y adaptadas para el poco consumo de energía, que para nuestra tesis denominaremos nodos; estos nodos dotan de procesamiento bajo y de comunicación inalámbrica con otros nodos, a su vez puede contener un sistema operativo llamado TinyOS y ejecutar un lenguaje de programación llamada nesC, ; cada nodo tiene características resaltantes que es autonomía, no necesita control para su desempeño y la mimetización en el ambiente dado su tamaño. Además, a los nodos, se le pueden añadir placas de diferentes tipos de sensores, a esta unión entre el nodo y la placa de sensores se le denominará nodo sensor. Dentro de esta red, cada nodo sensor puede captar datos de su entorno para transferir la información hasta llegar a un nodo que centralice la captación de datos todo ello inalámbricamente.

Como hemos descrito una arquitectura WSN está compuesta por los nodos sensor, pero falta otros componentes como el nodo sumidero que recolecta los datos y permite la interconexión entre la red de sensores y la estación base, quien es el encargado de procesar los datos.

Hay estudios realizados sobre las redes de sensores donde se utiliza una PC como estación base, causando muchos inconvenientes en el despliegue de los nodos sensor y nodo sumidero, en nuestro caso la estación base será un SBC o una mini computadora de bajos recursos que pueda funcionar de manera óptima con los demás componentes.

1.2 ANTECEDENTES

En los trabajos previos investigados existen implementaciones de red de sensores utilizando una PC como estación base y la elección de nodos sensor y nodo sumidero sin ningún criterio respecto a su performance, mermando características importantes para tener una red de sensores móvil, e inteligente.

1.2.1 DEL PROBLEMA

Según Oswaldo Fernando Daqui Solano (2011): **“DISEÑO E IMPLEMENTACIÓN DE UNA SOLUCIÓN DE MONITOREO REMOTO VÍA INTERNET, PARA UNA RED INALÁMBRICA DE SENSORES”**, para la Escuela Politécnica Nacional la cual propone la implementación de un sistema de monitoreo remoto para una WSN.

El resultado de su investigación se realizó, implementando una red de sensores con una PC como base estación, utilizando como distribución el sistema operativo Windows en esta base estación; para la comunicación y programación con los nodos sensor un sistema embebido en Windows llamada Cygwin, que facilita la interacción con los nodos y el envío de SMS de un celular con comandos AT, conectado a través del puerto serie a la computadora personal (pc).

El inconveniente en esta implementación es la utilización de una PC como base estación porque cuando se requiera una red de monitorización móvil y ubicuo, se necesitará un dispositivo electrónico con recursos mínimos que posibiliten la interacción con los nodos y el tamaño de un smartphone para su movilidad, también se necesita una comunicación con otras redes (GPRS, GSM) de forma más transparentes, eliminando la conexión de un dispositivo modem (celular) a la estación base de manera externa, por consiguiente se requiere una mini estación con la máxima conectividad posible tanto para wifi como gprs/gsm integrado.

Tesis desarrollada por Joel Orlando Aragon Valladares (2014), en su investigación denominada **“DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA DE GESTIÓN DE UNA RED DE SENSORES APLICADA A LA MONITORIZACIÓN DE LA CALIDAD AMBIENTAL EN LA CUENCA DEL RÍO NAPO”**, para la Pontificia Universidad Católica del Perú, nos presenta como implementar una red sensores como solución para la monitorización de la calidad ambiental, que permitirá medir calidad de aire y agua en parte de la región Loreto.

El resultado de su investigación concluye que la implementación permitió mostrar los niveles de contaminación del aire y del agua, entregando información importante que contribuirá con el control ambiental.

El inconveniente en esta implementación es cuando se requiera desplegar una red de sensores a gran escala utilizando varias redes conectados a varias bases estación, se encarecerá el costo por la utilización de todas las PC como base estación y una falta de adaptabilidad de las PC en el despliegue de una zona geográfica accidentada como el rio Napo.

Tesis desarrollada por Carlos Torrejón Sampedro (2010) en su investigación titulada **“PLATAFORMA MÓVIL PARA EL CONTROL DE UNA RED DE SENSORES INALÁMBRICA EN UN ENTORNO DOMÓTICO”**, para la Universidad Politécnica de Madrid - Facultad de Informática, nos muestra una solución al inconveniente de monitorizar el ambiente, gracias a la implementación de un programa en Java para un PDA como base estación.

Esta tesis se puede acercar en la utilización de una estación base de bajos recursos, pero el inconveniente de esta es que no llega a completar las características suficientes con respecto a la conectividad ni al procesamiento requerido para que esta estación base pueda soportar algún software inteligente, que es característica fundamental en la implementación de esta plataforma.

1.3 DEFINICIÓN DEL PROBLEMA

En los trabajos anteriores analizados, vemos que existen problemas relacionados a la implementación de una red de sensores. Primero que hay un problema de despliegue de los componentes en áreas accidentados o rurales por el tamaño de la estación base, debido a que esta es una PC, aparte no es flexible por su tamaño en un despliegue en áreas grandes, luego encarece el costo de las redes de sensores si se quiere varias estaciones base y su falta de conectividad para la comunicación con redes GSM/GPRS.

Así la alternativa para la solución de estos inconvenientes la utilización de una SBC que hará de mini estación base, dado sus recursos de esta pequeña computadora de placa única son limitados, su pequeño tamaño y la posibilidad de tener GNU-Linux e instalar software en ella. Por consiguiente, formulamos la siguiente pregunta:

¿En qué medida es factible desde el punto de vista técnico, la implementación de una Plataforma Hardware/Software de una red de sensores inalámbrica que permita una mini estación base (SBC) captar datos, siendo este el dispositivo de más baja gama dentro de su categoría y la mejor del mercado con respecto al precio y performance mínimo requerido?

1.4 JUSTIFICACIÓN

Por consiguiente, se tiene que analizar técnicamente una alternativa para la resolución a los problemas suscitados, donde la estación base sea una SBC de tamaño pequeño, con un costo reducido y lo más importante debe tener una performance y procesamiento mínimo para que pueda albergar un sistema operativo libre como GNU-Linux para que pueda soportar y ejecutar múltiples programas.

La implementación de esta plataforma debe conseguir que sea una red de sensores: *móvil e inteligente*; móvil, para que permita sin inconvenientes la movilidad de la estación base sin afectar su capacidad de comunicación con otras redes como la TCP/IP, GPRS, GSM, e inteligente con una capacidad de procesamiento mínimo para poder instalar softwares complejos y realizar análisis en los nodos y la SBC.

1.5 OBJETIVOS

1.5.1 OBJETIVO GENERAL

Determinar la factibilidad técnica de los componentes de las redes de sensores (nodos sensor, nodo sumidero y mini computadora) para crear una plataforma utilizando una estación base de tamaño reducido, de muy bajo recurso que pueda recibir datos de los nodos, que permita la comunicación con otras redes de largo alcance y pueda contener un sistema operativo.

1.5.2 OBJETIVOS ESPECÍFICOS

- Demostrar la factibilidad técnica de todos los componentes y herramientas en la implementación de la plataforma con una SBC como estación base para poder realizar la configuración, interconexión y programación entre las mismas.
- Realizar la elección de un dispositivo móvil (SBC) entre las minicomputadoras con características de bajo recurso que posea un óptimo performance de microprocesador, de memoria, de comunicación, etc. para la monitorización y control de una red de sensores inalámbricos y que se pueda comunicar con otras redes (TCP/IP, GPRS, GSM).
- Implementar un entorno que permita la recepción de datos por parte de la mini estación base con la elección de una distribución GNU Linux y la compatibilidad con el sistema operativo TinyOS sobre la arquitectura ARM el cual está basado los mini ordenadores o SBC.
- Determinar la factibilidad técnica de la conexión de los componentes (TTL, MicroSD) utilizados en la instalación de la imagen del sistema operativo Linux en la mini estación base mediante el puerto USB con el reconocimiento de la MicroSD como parte de unas de sus memorias de la SBC.

1.6 ALCANCES Y LIMITACIONES

La importancia del proyecto es el análisis comparativo de los nodos y SBC que existen, para luego estudiar la arquitectura del nodo sensor y nodo sumidero: el microprocesador, módulo de radio, interfaces, batería, conversor analógico digital, etc., su placa sensor adaptada para datos ambientales físicos como temperatura, humedad, luminosidad, etc. y una placa que hará de puerta de enlace.

Estudio de la arquitectura del procesador ARM de la SBC, y así determinar que software se puede instalar con su respectiva configuración. Por último, el funcionamiento del prototipo con los componentes integrados en la plataforma con los softwares instalados.

1.7 PROPUESTA DE LA TESIS

Se propone implementar una red de sensores donde se utilice una SBC como base estación, para dar solución a los inconvenientes en el despliegue de componentes en lugares accidentados, por el costo, tamaño y conectividad, al implementar una red de sensores con una PC como base estación.

Además, en esta plataforma de red de sensores se realizará un análisis comparativo de todos los componentes, para elegir componentes que puedan albergar un sistema operativo en todos los dispositivos de la plataforma, pero principalmente GNU-Linux en la SBC para que permitan instalar programas de software libre junto con herramientas optimizadas.

CAPITULO II

MARCO TEÓRICO

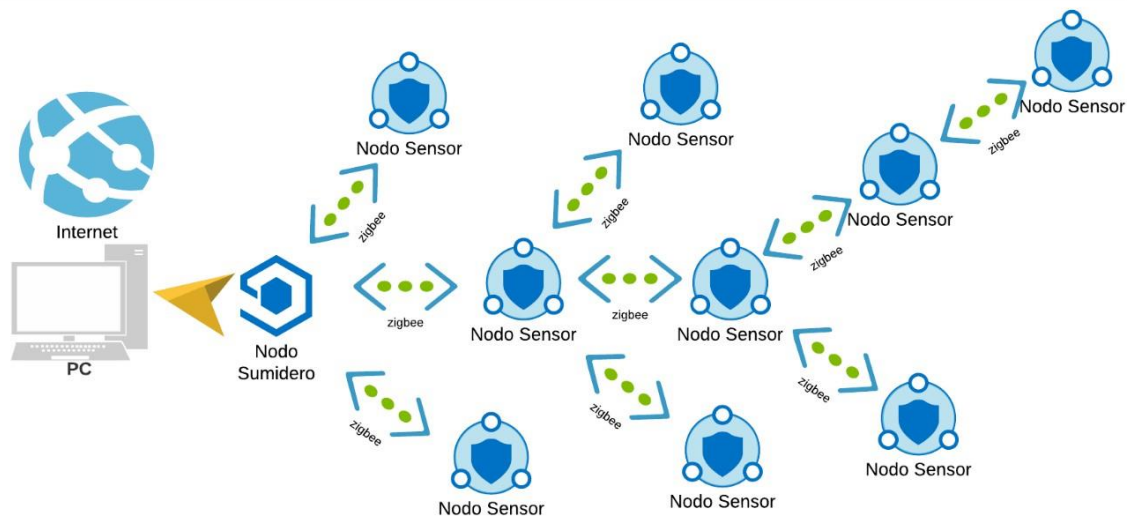
En este capítulo se muestra conceptos como los elementos que componen la plataforma, describiéndolos según su transmisión, además se muestra algunos conceptos sobre los sistemas operativos y lenguajes de programación para una red de sensores, criterio que nos ayudara en discernir a la hora de instalar y configurar los elementos.

2.1 REDES DE SENSORES - WSN

Una red de sensores está conformada por dispositivos con ciertas características que lo hacen especial. Primero todos los componentes deben de tener un tamaño reducido que ayuden al despliegue en un campo de acción, segundo la comunicación debe ser de manera inalámbrica, por ello es importante que tenga poco consumo de energía en la transmisión de datos y por último que sea de bajo costo. Para que podamos diferenciar estos dispositivos de otros es necesario que puedan albergar un sistema operativo para el procesamiento de datos y puedan tener autonomía en su trabajo. (Ver ilustración 1).

Ilustración 1 – Descripción general de la red de sensores

WSN



Fuente: Elaboración propia

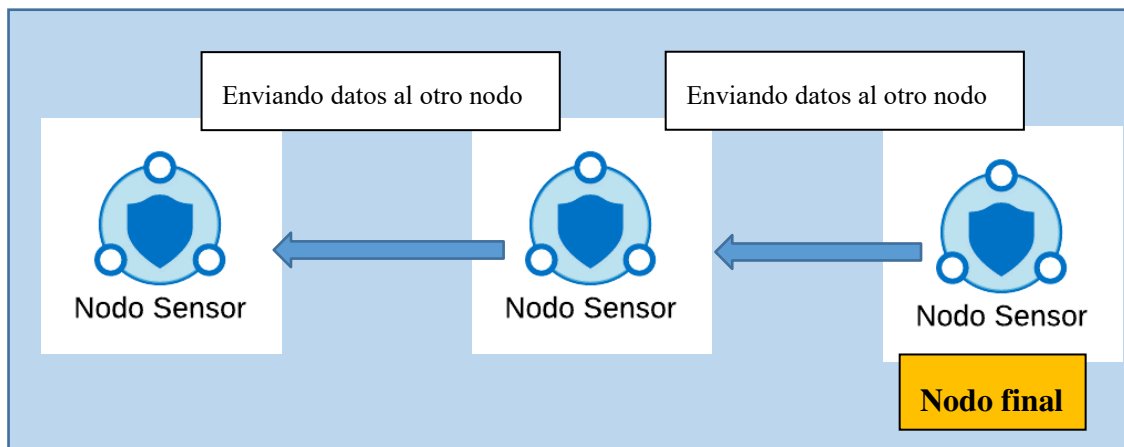
2.1.1 ELEMENTOS

Aquí mencionamos su clasificación de los nodos en función a la transmisión de datos que realizan:

Nodos finales

Son parte de la red de sensores donde su única función es de transmitir datos mas no recibir datos de otros elementos, porque suelen ser los últimos dispositivos colocados en el despliegue de los elementos. Estos elementos captan información del ambiente según los sensores conectados y envían por radio a los nodos intermedios próximos. (Ver ilustración 2).

Ilustración 2 -Elemento de la WSN - Nodo final

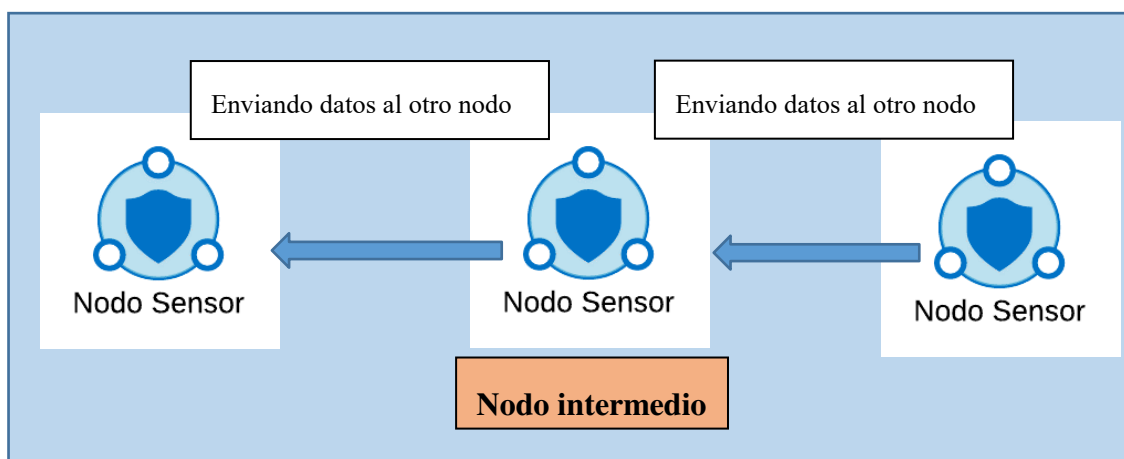


Fuente: Elaboración propia.

Nodos intermedios

Estos elementos actúan como "router", reciben información de otros nodos y luego transmiten sus propios datos más los datos de otros nodos, estos elementos intermedios tienen como tarea buscar caminos alternativos para el mejor tránsito de datos, hasta llegar a la estación de trabajo. (Ver ilustración 3).

Ilustración 3 -Elemento de la WSN - Nodo final



Fuente: Elaboración propia

2.2 SOFTWARE DE LAS REDES DE SENSORES

Para que los elementos de esta plataforma no solo reciban y entreguen datos de manera básica, sino que puedan procesar datos realizando tareas simples o complejas, es necesario que alberguen un sistema operativo tanto en los nodos como en la SBC facilitando así la programación de tareas que se necesiten.

2.2.1 SISTEMA OPERATIVO DE LOS NODOS

Los sistemas operativos para los componentes que vamos a utilizar deberán tener características adaptadas a dispositivos donde su microcontrolador, memoria, etc. son limitados, pero con performance suficiente para procesar datos en el mismo dispositivo.

Para las redes WSN, existen varios sistemas operativos para la programación de motas, entre ellos:

DuinOS	Sistema operativo en tiempo real, multitarea, se basa en el núcleo de FreerTOS, código libre. Deficiencia en el ahorro de energía en transmisión de datos y poca documentación.
TinyOS	Creado exclusivamente para los nodos creados por la universidad de Berkeley, contienen programas y algoritmos de ejemplos avanzados. Derivada del lenguaje C.
FreeRTOS	Es simple y pequeño derivado del lenguaje C. contiene múltiples tareas. Tiene velocidad en su ejecución.
Contiki OS	Portable, multitarea de código abierto, es orientado a eventos, compiten fuertemente con TinyOS.
eCos	Para sistemas en tiempo real, multitarea, puede soportar Linux empotrado, de código abierto.

2.2.2 LENGUAJE PROGRAMACIÓN DE LOS NODOS

Dado que los nodos tienen limitado microprocesador para el cálculo y de bajo recursos, el lenguaje que se ejecute dentro de ellas deberá ser uno que esté adaptado, se descarta un lenguaje orientado a objetos por la cantidad de memoria que se necesitaría. El lenguaje Nesc, es un lenguaje creado para este tipo de dispositivos orientado a componentes.

2.2.2.1 Nesc

NesC (Network Embedded Systems C) es un lenguaje de programación basado en C, enfocado y optimizado para su uso en aplicaciones de redes de sensores. Su origen se debe al deseo de disponer de un lenguaje específico que cumpliera con el modelo de ejecución y los conceptos del sistema operativo TinyOS.

Esta necesidad de desarrollar un nuevo lenguaje de programación específico para redes de sensores inalámbricos viene motivada por el tipo de aplicaciones que se desarrollan, que se caracterizan por:

- ⊗ Son aplicaciones basadas en recolección, difusión y control de la información obtenida del sensor, es decir, no son aplicaciones de propósito general.
- ⊗ Tienen que reaccionar ante cambios en su entorno (eventos).
- ⊗ Es preciso optimizar la limitada cantidad de recursos que ofrecen los nodos (memoria, capacidad de cómputo, consumo de energía).
- ⊗ Deben ser aplicaciones estables, puesto que deben correr durante meses / años sin intervención humana.
- ⊗ Precisan de control de errores en el manejo de datos.
- ⊗ Son aplicaciones en tiempo real (envío de mensajes a la red).

Teniendo en cuenta estos aspectos, nesC aporta como concepto innovador la metodología de lenguaje orientado a componentes. Con respecto al lenguaje C tradicional, en nesC, se precisan restricciones teniendo en cuenta a las limitaciones de los nodos. Estas precondiciones estáticas del lenguaje, son la base de la optimización y permiten al compilador realizar análisis profundos sobre el código. (López Egea, 2012)

2.3 HARDWARE DE LAS REDES DE SENSORES

Aquí describimos los componentes que integran las redes de sensores, así como los dispositivos que existen en el mercado.

2.3.1 NODO SENSOR

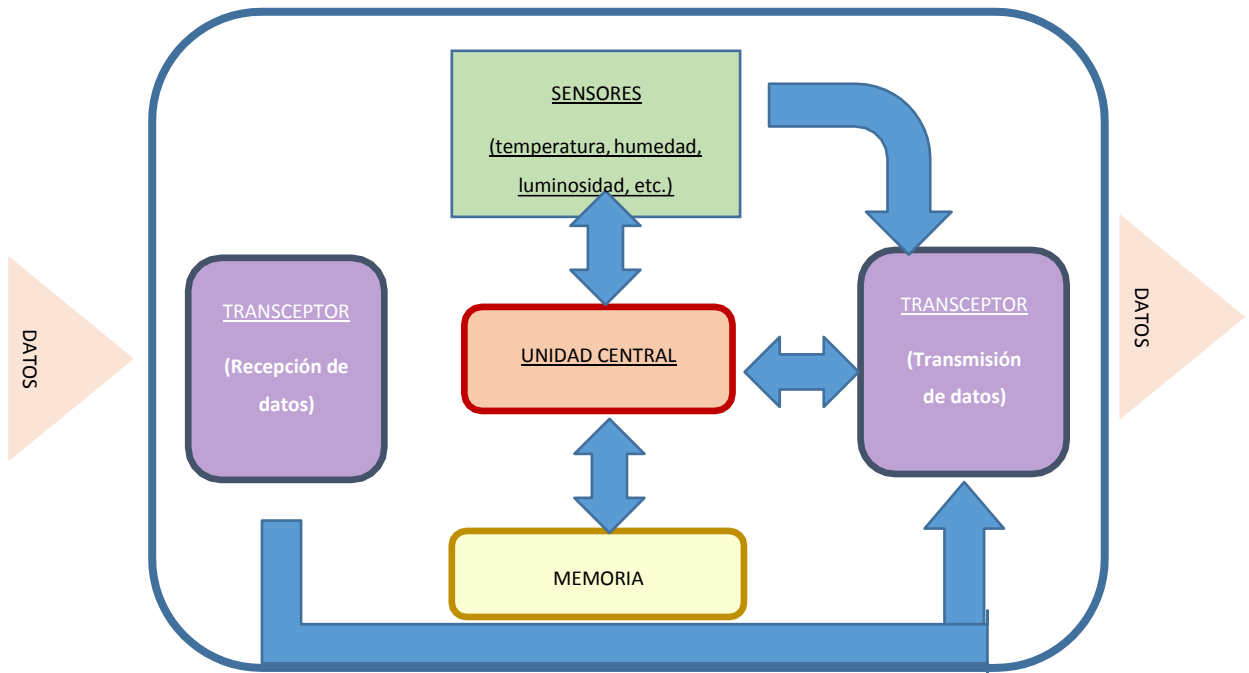
El nodo sensor tiene como objetivo la adquisición en tiempo real de datos del ambiente en cualquier zona, los sensores están conectados a una tarjeta la cual se encarga de recibir datos analógicos y transformar estos datos en una trama que serán enviados hacia el nodo sumidero, actúa como transmisor y receptor utilizando la tecnología de comunicación inalámbrica ZigBee.

2.3.1.1 COMPONENTES DE UN NODO SENSOR

Un nodo sensor normalmente consta de cuatro partes principales, como se muestra en la Ilustración 4:

- Sensores que captan datos del medio ambiente.
- La unidad central, que es el microprocesador que procesa las tareas.
- Un transceptor se capta datos del medio ambiente.
- Una Memoria, que almacena datos temporales o datos generados durante el procesamiento.

Ilustración 4 -Partes de un nodo sensor



Fuente: Elaboración Propia.

2.3.1.1.1 SENSORES

Dispositivos que tienen el objetivo de transformar los datos que quiere medir como temperatura, humedad, etc. en señal eléctrica. Para nuestra plataforma los sensores van a medir son señales físicas.

Tipo de sensores

La elección de un sensor para una aplicación depende de la propiedad física que debe ser controlada, por ejemplo, la temperatura, la presión, la luz o la humedad (Ver tabla 1).

Tipo	Sensor
Humedad	Higrómetros
Presión	Manómetros, barómetros, medidores de ionización
Posición	GPS, ultrasonido, infrarrojos
Temperatura	Termistores, termopares
Movimiento, vibración	Acelerómetro, giroscopio
Acústico	Micrófono, resonadores

Tabla 1- Cuadro de tipos de sensores

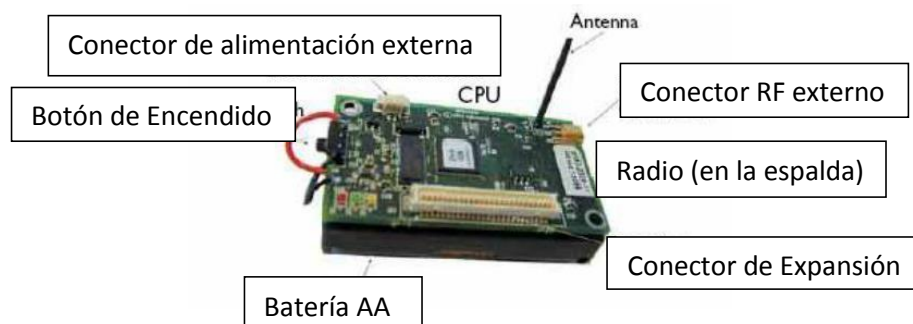
2.3.1.2 MODELOS DE NODO SENSOR

Con el pasar de los años se han ido desarrollando nodos que hoy en día se pueden conseguir a precios accesibles al público en general. Mostramos los nodos más comunes que existe en el mercado y que luego haremos una comparación técnica entre ellos.

2.3.1.3 MICA 2

Los MICA2 fueron desarrollados en la Universidad de Berkeley y eran ampliamente utilizados para la investigación en tanto los MICA2Dot eran una variación más reducida de los MICA2, contaban con 25mm de diámetro (Ver figura 1).

Figura 1 - Nodo Mica2

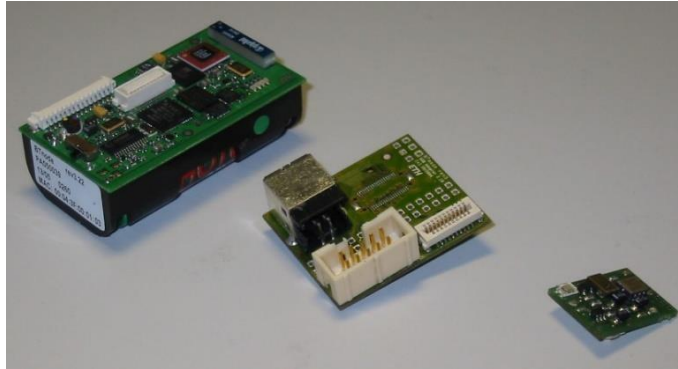


Fuente: (López Egea, 2012)

2.3.1.4 BTNODE3

Es un nodo muy utilizado en los proyectos de monitorización, los nodos tienen baja potencia y un procesador con buen performance. El módulo de transmisión es la misma que en el nodo Mica2 de Berkeley (Ver figura 2).

Figura 2 -Nodo BTNODE3



Fuente: (López Egea, 2012)

2.3.1.5 TELOS B

Según sus características los nodos TelosB realizan una transferencia de información optimizada gracias a su radio. Los nodos TelosB se pueden realizar programas y compilaciones de manera simple gracias a su puerto universal USB. La memoria para los programas compilados es de gran capacidad (Ver figura3).

Figura 3 -Nodo TelosB

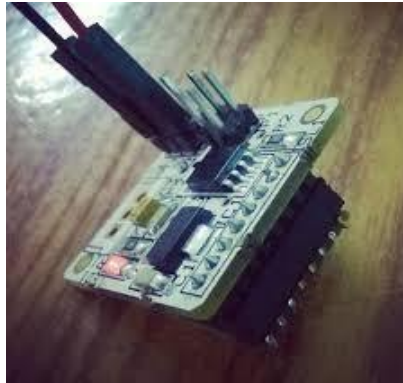


Fuente: (López Egea, 2012)

2.3.1.6 TINYNODE

Es un nodo creado para funcionar exclusivamente con el sistema operativo TinyOS, tiene características buenas con respecto a las potencias que utiliza, lleva consigo un sensor de temperatura pero que se puede integrar más sensores. (Ver figura 4).

Figura 4- Nodo TINYNODE

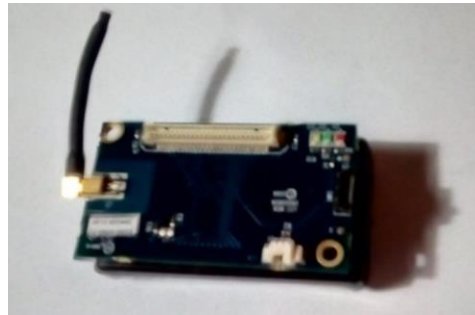
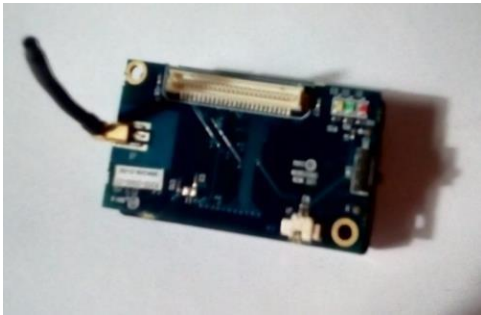


Fuente: (López Egea, 2012)

2.3.1.7 IRIS

El nodo Iris fue creado por la compañía CrossBow Technology Inc., es el nodo por defecto que suelen utilizar en sistemas embebidos gracias a la performance que tienen sus componentes electrónicos, bajo consumo en la transmisión, memoria para poder almacenar variables, y adaptación para colocar todo tipo de sensores.

Figura 5- Nodo Iris



Fuente: (López Egea, 2012)

2.3.2 NODO SUMIDERO

El nodo sumidero es el encargado de la recolección de información, actúa como receptor, recibe información del nodo sensor, consta de una tarjeta encargada de des encapsular los datos recibidos por el nodo.

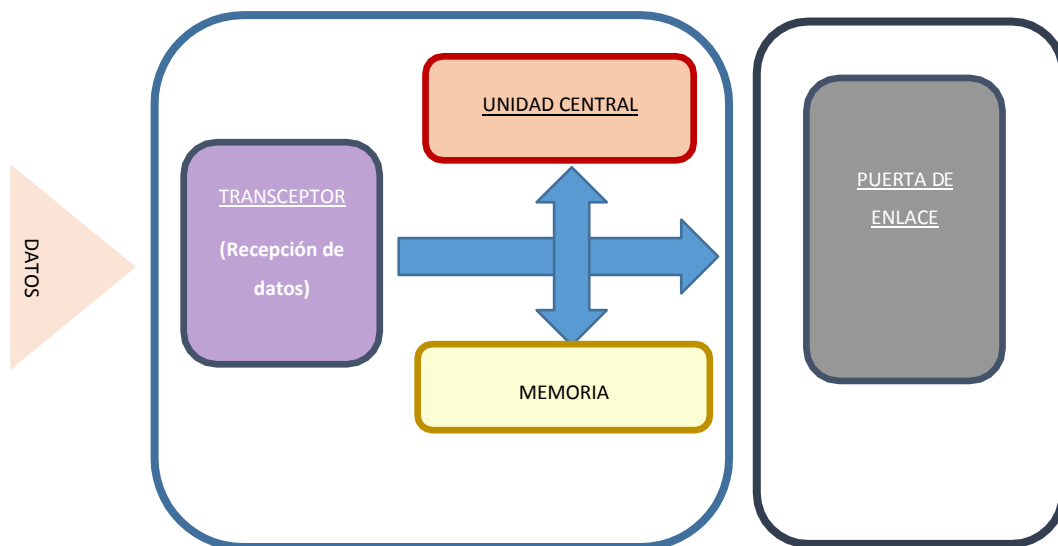
La diferencia entre el nodo sensor y el nodo sumidero, es con respecto a su función, mientras el nodo sensor tiene como objetivo captar los datos ambientales y enviárselo a otro nodo sensor hasta llegar al nodo sumidero, este nodo tiene como función principal la recepción de los datos de la red a través de la puerta de enlace para dirigir estos datos a la SBC.

2.3.2.1 COMPONENTES DE UN NODO SUMIDERO

Un nodo sensor normalmente consta de cuatro partes principales como muestra la ilustración 5:

- Puerta de enlace, recogen los datos de la red hacia la SBC.
- La unidad central, que es el microprocesador que procesa las tareas.
- Un transceptor se capta datos del medio ambiente.
- Una Memoria, que almacena datos temporales o datos generados durante el procesamiento.

Ilustración 5 -Partes de un nodo sumidero



Fuente: Elaboración Propia.

2.3.2.1.1 PUERTA DE ENLACE

La puerta de enlace es la placa que contiene el nodo sumidero entre la estación de trabajo a cargo de la SBC y los nodos sensor que componen la red. Todos los datos recibidos por la puerta de enlace son enviados a la SBC para su debida utilización para diferentes propósitos.

2.3.2.2 MODELOS DE NODO SUMIDERO

Los modelos de nodo sumidero van a depender del tipo de nodo sensor que se elija para la arquitectura, hay multitud de nodos sumideros, tienen una relación directa con el nodo sensor en el trabajo conjunto de transporte de datos. Por consiguiente, una vez decidido el nodo sensor a utilizar se mostrará las opciones de la puerta de enlace y luego haremos una comparación técnica entre ellos.

2.3.3 ESTACION BASE – SBC (Single-Board Computer).

Como ya hemos descrito en nuestra tesis para nuestra plataforma hardware utilizaremos una SBC o mini computadora como base estación, para ello tendremos que describir un poco de ellos.

La SBC es una pequeña computadora de placa única. Además, funcionan en GNU- Linux y sistemas Android, con arquitecturas en ARM que normalmente existe para estos dispositivos.

El conjunto de sistemas operativos basados en Linux para estas SBC puede ser: LibreELEC , Archlinux ARM, , Pidora, Raspbmc, RISC OS, Raspbian OpenWrt , openSuSE, Slackware ARM, , Plan 9, OpenELEC, FreeBSD, Kali Linux, Instant WebKiosk, Ark OS y Minepeon, Debian, Ubuntu.

Las SBC ofrecen cuatro características importantes; versatilidad en su hardware, bajo consumo (1-3 vatios), de reducido su tamaño y a un bajo precio, ideado para que la informática y la electrónica los utilice en diferente propósito (servidor web, robótica, media Center, etc.).

Otra característica importante es que las SBC disponen de un abanico de conexiones de salida que abren un mundo de posibilidades. La interfaz de red Ethernet, puertos USB y el HDMI, entre otros, proporcionan muchas alternativas para diferentes tipos de aplicaciones, en nuestro caso nos interesa las conexiones de red y los puertos USB que harán conexión con el nodo sumidero y esta con los nodos de sensor.

2.3.3.1 PARTES DE LA SBC

Existen varios componentes imprescindibles para que una computadora tenga un buen funcionamiento. En primer lugar, para una PC normal se requiere energía eléctrica, cuyos electrones serán expandidos a todos los circuitos internos. Para esto, se utilizan baterías y cables de alimentación para llevar energía a una computadora. Pero no ocurre lo mismo cuando se trata de un SBC, ya que, por cuestiones de diseño, es prácticamente imposible.

Por el contrario, un SBC recibirá energía mediante un conector USB, ya que con esta interfaz se pueden transferir también datos. Otra opción para que la SBC funcione, sería conectándolo a una pantalla que esté alimentada por un tomacorriente, así recibiría la energía a través de la pantalla y podría funcionar sin problemas.

Asimismo, una computadora va a necesitar de un procesador para recopilar datos y procesarlos. Pero nuevamente, es imposible instalar un procesador de PC de escritorio en un SBC, porque normalmente esta tiene solo una función específica que cumplir.

En su lugar, estos pequeños ordenadores tienden a utilizar procesadores ARM que **es la parte fundamental** y donde inicia la diferencia comparando las dos arquitecturas, estos procesadores son los que suelen utilizarse para dispositivos móviles como los smartphones o teléfonos inteligentes, obteniendo con esto un tamaño reducido y un menor consumo de energía, disminuyendo la emisión de calor.

Además, se debe tener, de una unidad de procesamiento gráfico GPU, pero integrado (iGPU), que es solamente la parte fundamental de una tarjeta gráfica (GPU, los módulos de memoria, su disipador, sus salidas de vídeo, etc.) en una computadora, esta es quien realiza los cálculos matemáticos para representar una imagen en la pantalla.

En cuanto a la memoria, un SBC vendrá generalmente equipado con una memoria flash, la que ya viene integrada en el circuito y que ocupa muy poco espacio que tendrá como función el almacenamiento de variables y objetos de programas realizados y otra memoria flash externa que se utilizará para cargar una imagen del sistema operativo GNU-Linux.

La conectividad es otro punto importante que encontraremos en un SBC, que puede incluir conexiones USB como interfaz física para poder ser conectado a pantallas, ratones u otros periféricos y una conexión de red que podría ser

inalámbrica ya sea de corto alcance (bluetooth) , mediana alcance (wifi) y largo alcance (GSM/GPRS) .

Asimismo, hay otros SBC que pueden ofrecer conexiones HDMI, que también permitirán que este pequeño ordenador establezca comunicación con diferentes dispositivos.

2.3.3.2 PROCESADORES ARM

La arquitectura ARM se diseñó para permitir implementaciones de tamaño muy reducido y de alto rendimiento. Se caracteriza fundamentalmente por ser una computadora de set de instrucciones reducido (Reduced Instruction Set Computer, RISC), como lo indica su propio nombre.

El diseño del ARM se ha convertido en uno de los más usados del mundo, desde discos duros hasta juguetes. Hoy en día, cerca del 75% de los procesadores de 32 bits poseen este chip en su núcleo.

Los procesadores ARM han sido creados para realizar instrucciones más simples, a un nivel muy bajo, es por este motivo que estos procesadores consumen menos energía, que lo convierte en parte primordial para los dispositivos que procesan datos y que requieren de un menor consumo.

Estos procesadores han ido evolucionando a través del tiempo, de su inicio con la arquitectura 6502 de Rockwell hasta los ARMv8 de 64 bit que son requeridos por teléfonos inteligentes como los Apple.

Las familias de ARM trabajan con 32 bits hasta la familia Cortex-A50 la cual pertenece a la arquitectura ARMv8, esta incorpora operabilidad de 64 bits (Ver tabla 2).

El Cortex R tiene esa denominación para aplicaciones en tiempo real, Cortex M que hace de microcontrolador en dispositivos como por ejemplo Arduino y el Cortex A porque uso en ejecuciones de aplicaciones en dispositivos como SBC, SmartPhone o SmartTV.

Algo importante de rescatar ha sido la compatibilidad retroactiva de código para que las aplicaciones de 32 bits puedan seguir funcionando en estos nuevos modelos.

FAMILIA	ARQUITECTURA	NUCLEO
ARM7TDMI	ARMv4T	ARM7TDMI(S)
ARM9 ARM9E	ARMv5TE(J)	ARM926EJ-S,ARM966E-S
ARM11	ARMv6 (T2)	ARM1136(F), 1156T2(F)-S, 1176JZ(F), ARM11 MPCore™
Cortex-A	ARMv7-A	Cortex-A5, A7, A8, A9, A12, A15
Cortex-R	ARMv7-R	Cortex-R4(F), Cortex-R5, R7...
Cortex-M	ARMv7-M	Cortex-M3, M4
	ARMv6-M	Cortex-M1, M0, M0+
Cortex-A50	ARMv8-A	64 Bit: Cortex-A53 and A57

Tabla 2 - Cuadro comparativo de familias ARM

2.3.3.3 MODELOS DE SBC

Cuando surge un modelo de mucho éxito como lo fue las Raspberry en el 2009, es inevitable que surjan empresas tecnológicas que desarrollen SBC parecidas y con mejores opciones, en esta década se han creado una legión de placas alternativas para atraer a diferentes tipos de usuarios según la necesidad requerida, a continuación, hacemos mención de algunas familias de SBC que están en el mercado tecnológico con éxito:

Banana Pi	Placas muy populares en el mundo, los precios son aproximadamente 70\$. Se puede utilizar muchas plataformas GNU-Linux.
HummingBoard	Mayor rival de la Raspberry Pi, junto con Orange Pi. Sus precios entre 50 hasta los 100 dólares. Básicos pero potente su núcleo de procesamiento.
Orange Pi	Puede contener GNU-Linux y Android como sistema operativo, mucha variedad en sus interfaces y sus procesadores basados en ARM. Sus precios pueden variar desde los 8 dólares.
Raspberry Pi	El SBC más utilizado y comprado en el mundo, tiene diferentes gamas, pero la demanda ha hecho que los precios sean excesivos.

CAPITULO III

ESTADO DEL ARTE

En este apartado mostramos un cuadro (Ver tabla 3), donde nombramos las tesis revisadas y que tienen mucha relación con nuestra investigación, Luego una breve descripción de tres de las propuestas más resaltantes investigadas, y finalmente un cuadro comparativo (Ver tabla 4), con los componentes utilizados para las diferentes soluciones descritas.

Título del Paper	Autores
“IMPLEMENTACIÓN DE UN PROTOTIPO DE WSN CON NODOS INTELIGENTES PARA EL SISTEMA DE RIEGO APLICADO A LA AGRICULTURA DE PRECISIÓN”	MARJORIE NATALY PAREDES VINUEZA (2017)
“APLICACIONES PARA REDES DE SENSORES EN IPV6”	MOISÉS E. CORONADO D (2016)
“MODELO DE ESTIMACIÓN DE LA DISTANCIA EN INTERIORES BASADO EN MEDIDA RSSI EN UNA WSN”	MARIA EUGENIA LÓPEZ GARCÍA (2016)
“DISEÑO E IMPLEMENTACIÓN DE UNA SOLUCIÓN DE MONITOREO REMOTO VÍA INTERNET, PARA UNA RED INALÁMBRICA DE SENSORES.”	OSWALDO FERNANDO DAQUI SOLANO (2011)
“EVALUACIÓN DE LA EFICACIA DE UN SISTEMA OPERATIVO EN ARDUINO”	JUAN JOSÉ FERNÁNDEZ RODRÍGUEZ (2015)
“EVALUACIÓN DEL RENDIMIENTO EN EL NIVEL DE RED RPL EN WSN”	ANTONIO MORILLO CANALEJO (2017)
“RED DE SENSORES INALÁMBRICOS PARA MONITORIZACIÓN DE INSTALACIONES ELÉCTRICAS DE BAJA TENSIÓN.”	JOSÉ ANTONIO TARIFA GALISTEO (2011)
“DISEÑO E IMPLEMENTACIÓN DE UNA RED DE SENSORES PARA EL MONITOREO DE VARIABLES CLIMÁTICAS EN UN INVERNADERO DE ORQUÍDEAS”	LUIS FERNANDO FERNÁNDEZ CRUZ (2017)
“DISEÑO E IMPLEMENTACIÓN DE UNA RED DE SENSORES INALÁMBRICOS PARA EL MONITOREO DE FACTORES AMBIENTALES ABIÓTICOS QUE AFECTAN AL CULTIVO DE SOLANUM PHUREJA”	LUIS PATRICIO SUMBA QUITO, DANNY LEONEL TORRES SIMANCAS (2019)

Tabla 3 - Investigaciones

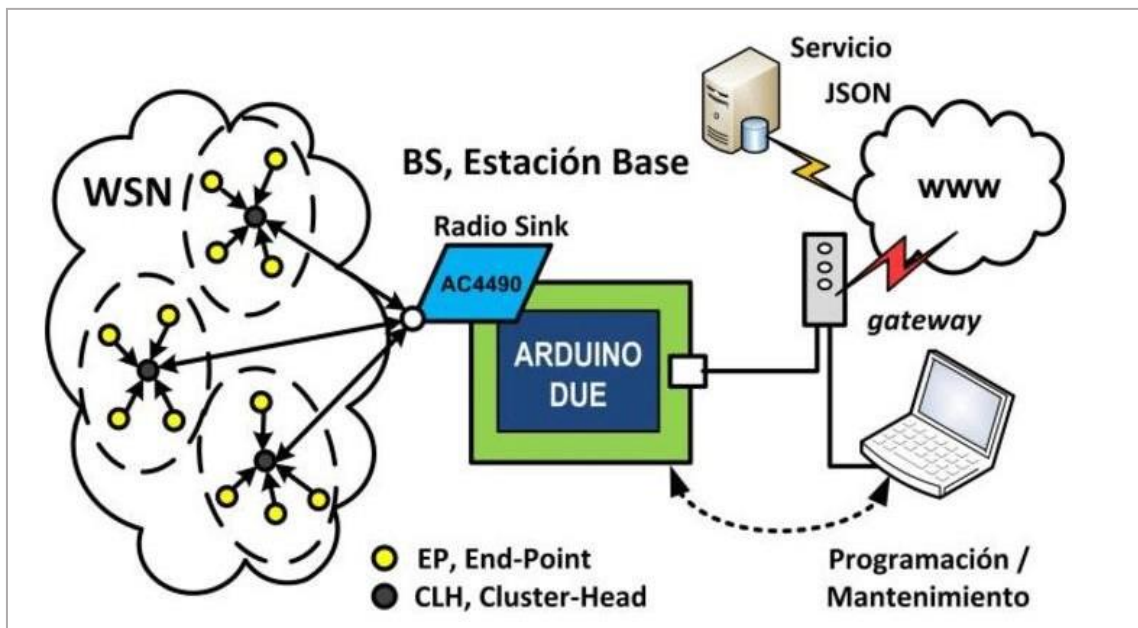
3.1 IMPLEMENTACION DE ARQUITECTURAS ALTERNATIVAS

3.1.1 IMPLEMENTACIÓN DE UNA ESTACIÓN BASE PARA UNA RED INALÁMBRICA DE SENSORES DE LARGO ALCANCE CON CONEXIÓN A LA NUBE.

(Ortega Corral & Acosta del Campo, 2015) Ortega nos presenta la integración de una estación base (BS, del inglés Base Station) de alto rendimiento, encargada en recibir telemetría de una red con arquitectura jerárquica de largo alcance. Su diseño incorpora un sistema procesador de 32 bits ARM (Advanced RISC Machine) en comunicación serie con un transceptor digital de 900MHz (con opción a operar a 2.4GHz), a través del cual la BS recibe los mensajes inalámbricos que envían los nodos de la WSN.

Para el envío de los mensajes de los sensores hacia la nube de datos, la estación base establece comunicación mediante una interfaz Ethernet con conexión a un Gateway vía direccionamiento IP (Internet Protocol). Dichos mensajes están codificados usando un protocolo modificado denominado Light JSON (Java Script Object Notation); por lo que la estación base con programación orientada a objetos se encarga de traducirlos a JSON estándar y de enviarlos a un servidor remoto a través de Sockets TCP/IP. La codificación y decodificación sirve como parte del proceso de la validación de la información en tránsito. Estos mensajes que viajan de extremo a extremo de la WSN, contienen datos de aplicación y métricas de otras capas de la red inalámbrica, que se agregan mientras realizan los saltos establecidos en el enlace, hasta un servidor dedicado donde se guarda la información en un sistema de base de datos (DBS, data-base system). Aquí se presenta la actualización del hardware de la estación base, la migración de su software, sus funciones, la ampliación de sus operaciones y el mejoramiento en el manejo del paradigma de validación con acuses de recibo (Ver figura 10).

Figura 6 Uso del Arduino Due como procesador central de la estación base



Fuente: (Ortega Corral & Acosta del Campo, 2015)

3.1.2 IMPLEMENTACION DE UN PROTOTIPO DE WSN CON NODOS INTELIGENTES PARA EL SISTEMA DE RIEGO APLICADO A LA AGRICULTURA DE PRECISION.

(Paredes Vinueza, 2017) Nataly presenta otra aproximación al modelo de la plataforma que se quiere implementar. Esta red inalámbrica de sensores inteligentes se requiere para la supervisión y control de riego de la parcela de mora en el Centro Experimental del Riego de la Escuela Superior Politécnica de Chimborazo el cual consiste en un sistema electrónico de sensores inalámbricos basados en tarjetas de desarrollo Arduino.

Para el sistema se desarrolló una aplicación móvil que mantiene una comunicación vía Bluetooth para la visualización de las lecturas del nodo sensor; los datos son receptados y visualizados en tiempo real, además la información obtenida es mostrada de manera gráfica y almacenada en la plataforma libre ThinkSpeak, asimismo la información es almacenada en una memoria micro SD en el nodo control.

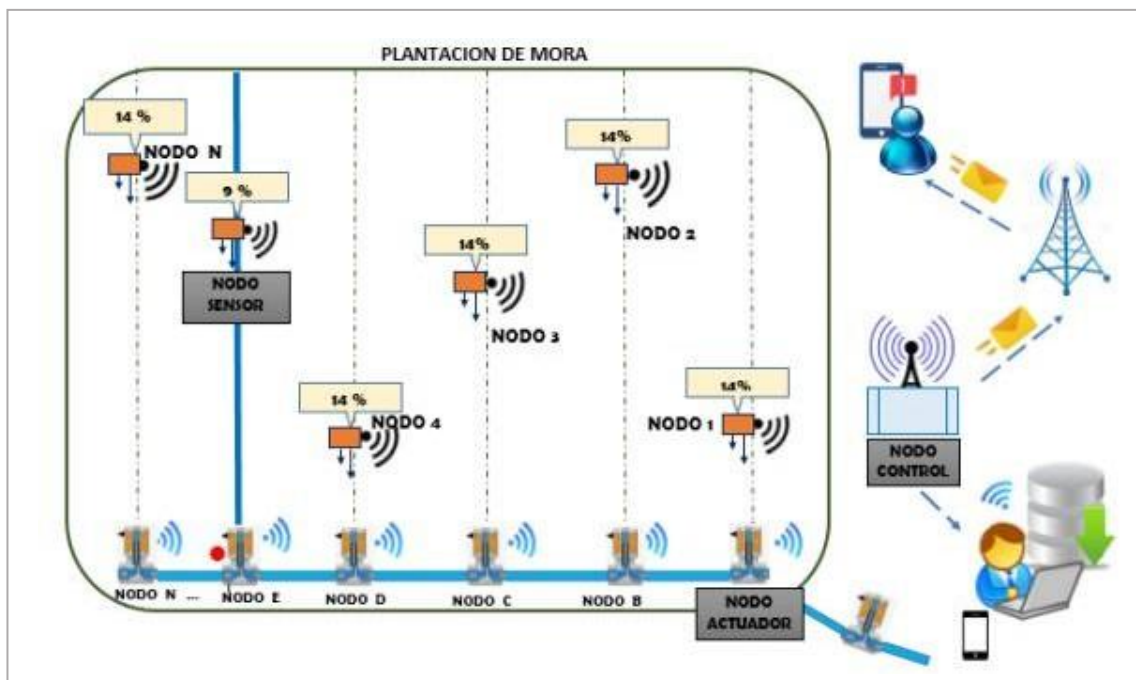
La arquitectura propuesta se muestra en la figura 6, donde se aprecian los tres tipos de nodos principales que conforman el sistema: nodo sensor, nodo actuador y nodo control. La comunicación entre los nodos es a través de ZigBee.

Nodo Sensor: Tiene como objetivo la adquisición en tiempo real de niveles de humedad del suelo y la temperatura ambiente en una zona determinada del cultivo., los sensores están conectados a la tarjeta de desarrollo la cual es encargada de recibir datos analógicos y transformar estos datos en una trama que serán enviados hacia el nodo control utilizando la tecnología de comunicación inalámbrica ZigBee.

Nodo Actuador: Recibe la orden del nodo control a través del módulo XBee para empezar el riego activando una electroválvula, además cuenta con un medidor de flujo para llevar un registro de la cantidad de agua utilizada por el cultivo en un lapso determinado de tiempo.

Nodo Control: Es el encargado de la recolección de información, actúa como transmisor y receptor, recibe información del nodo sensor y genera la orden para el nodo actuador, consta de una tarjeta de desarrollo encargada de des encapsular los datos recibidos por el módulo (Ver figura 7).

Figura 7 Arquitectura general del sistema inalámbrico



Fuente: (Paredes Vinueza, 2017)

3.1.3 DISEÑO E IMPLEMENTACION DE UNA RED DE SENSORES INALÁMBRICOS PARA EL MONITOREO DE FACTORES AMBIENTALES ABIOTICOS QUE AFECTAN AL CULTIVO DE SOLANUM PHUREJA.

(Soto Sarango, 2019) nos dicen que implementaran una red de sensores inalámbricos para el monitoreo de factores ambientales abióticos que afectan al cultivo de solanum phureja (Papa Chaucha Amarilla). Ellos detallan las especificaciones de hardware y software seleccionados para el diseño de la red de sensores inalámbricos. Además, se muestra el diseño de los nodos necesarios para adquirir las señales censadas en la red conjuntamente con el diseño del protector para los nodos (Ver figura 8).

Se puede apreciar los tres tipos de nodos principales para implementar la red WSN en un cultivo de papa. Los nodos principales que conforman el sistema: nodo sensor, nodo actuador y nodo control. La comunicación entre los nodos es a través de ZigBee.

El nodo router R1 y R2: Las señales adquiridas por los sensores de luminosidad y ph son procesadas por un único Arduino UNO para posteriormente ser transmitidas de forma inalámbrica hacia el nodo coordinador.

El nodo coordinador: Es el encargado de gestionar los datos recibidos de forma inalámbrica de los otros nodos que conforman la red WSN y un reloj configurado para que se realicen mediciones cada hora durante 15 días y tres veces al día (4:00, 13:00 y 20:00) durante 45 días. Adicionalmente, el nodo coordinador realiza la comunicación hacia la base de datos MySQL que este servidor de base de dato almacenara los valores obtenidos por los sensores utilizados en nuestra red. Esto se logra mediante la configuración de la placa de una placa de desarrollo que tiene incorporado un módulo WiFi, el cual se conecta a nuestra red de área local (LAN) y posteriormente hacia internet con el fin de mostrar los datos adquiridos en tiempo real a través de una aplicación móvil con interfaz estándar para el usuario final, que, en este caso es el productor de papa.

Figura 8 Arquitectura general del sistema inalámbrico

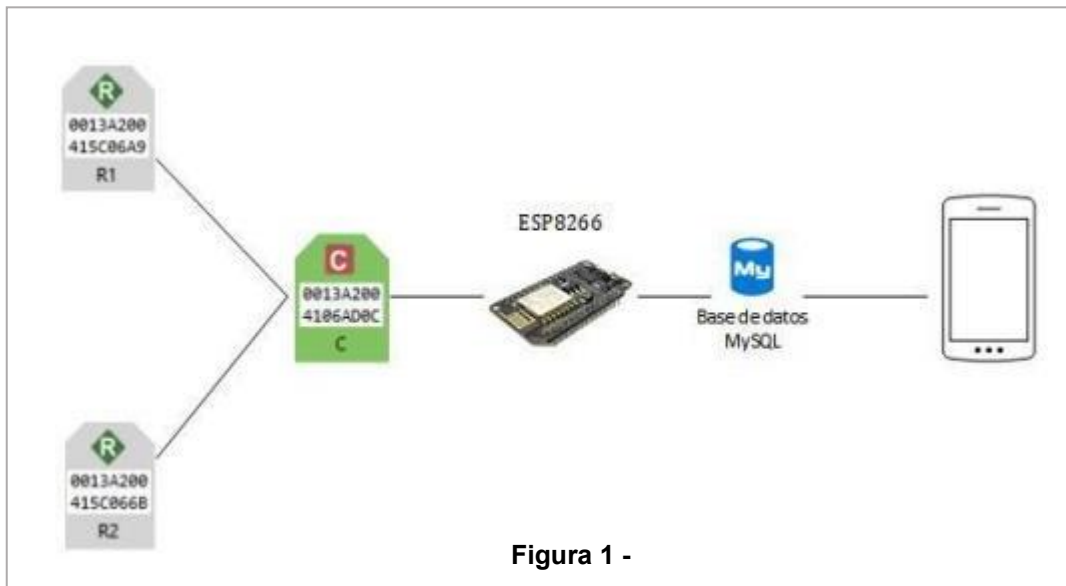


Figura 1 -

Fuente: (Soto Sarango, 2019)

	Nodo sensor	Nodo sumidero	Estación base
WSN con conexión a la nube	Arduino uno / XBee S2C	Arduino uno / XBee S2C	Arduino Due
WSN aplicado al riego de precisión	Arduino uno / XBee S2C	Arduino NANO / ESP8266	Ninguno
WSN aplicado al medio ambiente	Arduino uno / XBee S2C	Arduino uno / XBee S2C/ NodeMCU ESP8266	Ninguno

Tabla 4 - Cuadro comparativo de plataformas investigadas

CAPITULO IV

APORTE TEÓRICO

En este capítulo se elige los componentes a emplear a través de un método comparativo, de las partes que consta la plataforma, finalmente se explica su adaptación y la relación que habrá entre ellos.

4.1 SELECCIÓN DE LOS COMPONENTES PARA LA PLATAFORMA

Considerando el análisis comparativo se selecciona los componentes que van a ser tomados en cuenta para implementar la plataforma.

4.1.1 SELECCIÓN DEL NODO PARA LA IMPLEMENTACION DE LA ARQUITECTURA PROPUESTA

En el capítulo III del estado del arte, se ha visto como en algunas tesis optan por adaptar la placa Arduino con placas de XBee para la comunicación y sensores para captar datos del ambiente y así implementar una plataforma; esta es un implementación practica pero no dota de inteligencia a la plataforma, porque no permite poder instalar un programa en el nodo sensor para múltiples funciones, por ejemplo que nos indique enviando algún dato desde el mismo nodo sensor cuándo la batería está llegando a su fin y así cambiarla o también poder reinstalar un programa desde la estación base, motivo por el cual se deciden seleccionar el nodo sensor que permitan instalar algún sistema operativo en ellas, descritas en el marco teórico.

A continuación, se describen los criterios de evaluación definidos con su respectiva justificación según las prioridades determinadas en la presente tesis de investigación.

Criterio 1: Consumo de voltaje en el microprocesador

Se requerirá voltaje óptimo de alimentación, de tres tipos de gasto de energía en activo, en inactivo y en apagado.

Criterio 2: Cantidad de valores discretos ADC

Se necesitará la mayor cantidad de valores discretos que pueda convertir el chip Analógico - Digital, para representar un rango analógico de interés.

Criterio 3: Consumo de voltaje en la radio

Se requerirá que la radio sea eficiente en consumo de amperaje con respecto a su rendimiento al transmitir y recibir datos

Criterio 4: Tipos de memoria y capacidad

Se requerirá un rango óptimo de capacidad de almacenamiento para el programa que se ejecutaran

Criterio 5: Variedad de interfaces

Es necesario que los nodos presentan mucha variedad de interfaces que sirven para comunicarse con los sensores y también cuentan con entradas y salidas de propósito general.

A continuación, pasamos a describir y comparar los componentes del nodo sensor según sus características.

4.1.1.1 MICROCONTROLADOR

El objetivo de un controlador se ha mantenido hasta el día de hoy desde que inició su utilización, pero desde hace tres décadas han ido evolucionando, primero utilizando componentes de lógica discreta para luego evolucionar en microprocesadores trabajando conjuntamente con chips de memorias y chips de entrada y salida sobre un circuito impreso. A día de hoy todos ellos trabajan en un chip, llamado microcontrolador. Es un pequeño computador que se encuentra en casi todos los circuitos integrados que realicen alguna función determinada.

Por lo tanto, en nuestra propuesta el micro controlador es aquel chip que realizara la interpretación de la información que le llega y los procesara en su envío inalámbricamente a otro nodo sensor o estación base, a continuación, mostramos los micro controladores que contiene cada nodo descrita en el cuadro

comparativo siguiente, podemos ver que lo más resaltante de un micro controlador es la fuente de alimentación (voltaje), el consumo en modo: activo, inactivo y apagado (Ver tabla 5).

	Atmel ATmega 128L (BTnode, MicaZ)	Atmel ATmega 1281 (Iris)	Texas Instruments MSP430 (TelosB, TinyNode)
Fuente de alimentación	2.7.....V	1.8..... V	1.8..... V
Consumo en modo activo	3.6 mA @ 4MHz, 3.0V	0.5 mA @ 1MHz, 1.8V	0.2 mA @ 1MHz, 2.2V
Consumo en modo inactivo	1.0 mA @ 1MHz, 3.0V	---	0.7 μ A @ 1.8V
Consumo en modo apagado	0.5 μ A @ 3.0V	0.1 μ A @ 1.8V	0.1 μ A @ 2.2V

Tabla 5 – Cuadro comparativo del consumo de voltaje de los microcontroladores

Ya que se busca uno que trabaje con menor energía y genera mayor eficiencia. Analizando lo anterior podemos resaltar al Atmega 1281 como un microcontrolador muy eficiente en tanto el Atmega 128L no es muy destacable por presentar un requerimiento muy elevado de voltaje sobre todo en sus consumos.

4.1.1.2 CONVERTOR ANALOGICO DIGITAL - ADC

Los nodos sensores captan datos que se encuentra en el medio ambiente, estos datos son físicos o químicos, y pueden ser temperatura, humedad, luminosidad, etc., como podemos ver estos datos del medio ambiente necesita ser convertidos en datos digitales mediante una tensión eléctrica, de esta función es quien se encarga un conversor analógico digital.

Nuestro nodo usa transductores (sensores) para poder leer un tipo de dato, y como estos se comunican con los microprocesadores, hay una falta de entendimiento al ser los transductores analógicos y los microprocesadores digitales, es por ello que nuestro conversor nos permite convertir estas señales analógicas a señales digitales binarias.

La salida digital de un módulo ADC es un número binario en complemento a 2. La cantidad de valores discretos que puede contener representa el rango analógico de interés.

Por ejemplo, una ADC de 8bits puede representar 256 valores de una señal analógica. Según nuestro cuadro los convertidores de los nodos propuestos varían de 10 bit a 12 bit, ósea de 1024 a 4096 niveles de señal (Ver tabla 6).

ADC	
BTnode	ADC -> 12bit
Iris	ADC -> 10bit
MicaZ	ADC ->10bit
TelosB	ADC ->12bit
TinyNode	ADC ->12bit

Tabla 6 - Cuadro comparativo del conversor analógico digital en los nodos

4.1.1.3 RADIO

Los nodos de una WSN en general utilizan la banda ISM. Esta banda no necesita licencia para ser utilizada.

Las radios se componen de un dispositivo que transmite y recibe señales electromagnéticas llamada transceptor. Este transceptor actúa sobre los cuatro estados, estos pueden ser recepción, emisión, apagado y dormido. Para un consumo eficiente de energía estas se encuentran la mayor cantidad de tiempo apagado, y despertando para escuchar el canal para recibir paquetes o transmitir paquetes.

BANDA ISM (Industrial, Scientific and Medical)

El transceptor de los nodos sensor envían señales por una banda de frecuencia electromagnética, abierta sin licencia. Esta banda ISM son de uso no comercial para sector industrial, científica y médica. En estas décadas ha sido utilizadas popularmente para uso en las comunicaciones de mediano alcance WLAN y corto alcance WPAN.

	Chipcon CC1000	Atmel AT86RF230	Chipcon CC2420	Xemics XE 1205
Voltaje de alimentación	2.1.....V	1.8..... V	2.1..... V	2.4..... V
Consumo en modo dormido	0.2 μ A	0.1 μ A	426 μ A	0.2 μ A
Consumo al recibir	9.6 mA	16 mA	1.8 mA	14 mA
Consumo al transmitir	25.4 mA @ 5dBm	17 mA @ 3dBm	17.4 mA @0dBm	33 mA @ 5dBm
Rendimiento	1.2-500Kbps	250Kbps	250Kbps	---
Frecuencias	315/433/915 MHz	2.4 GHz	2.4 GHz	---

Tabla 7 - Cuadro comparativo del consumo de voltaje de los radios

Al observar (Ver tabla 7) podemos ver que las frecuencias en la que trabaja la radio en su mayoría son en el orden de los GHz. Esto es importante ya que la plataforma hardware tiene como propósito ser utilizado en un ambiente doméstico y rural.

Al igual que el micro controlador se busca un radio que trabaje con menor energía y genera mayor eficiencia. Analizando lo anterior podemos resaltar al Atmel AT86RF230 como un radio eficiente en consumo de amperaje al transmitir y recibir con respecto a su rendimiento.

4.1.1.4 MEMORIA

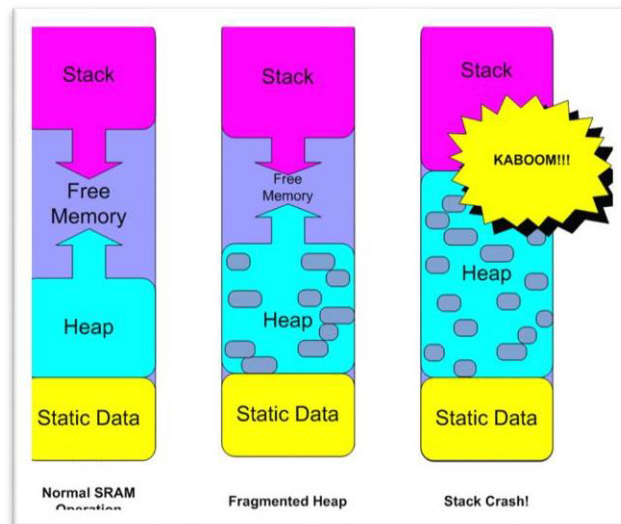
Como se mencionó los microprocesadores trabajan directamente con las memorias e interfaces de entrada y salida, por consiguiente, las memorias deben estar sincronizadas en el “dialogo” con el micro, pero en la realidad no ocurre, trabajan a diferentes velocidades es por esto que la información almacenada se distribuye de diferentes formas para poder solucionar este inconveniente. Las memorias se clasifican en función al tipo de acceso en:

4.1.1.4.1 MEMORIA SRAM (static random access memory)

Tanto los nodos como la estación base utilizarán programas compilados y ejecutados que almacenarán datos, estos datos serán variables locales y datos parciales que han sido creados y se manipularán cuando se ejecuta el programa, por tanto, es necesario disponer de un mejor tiempo de acceso y una buena administración de su uso para evitar agotar este espacio limitado. Esta memoria podría ser leída y escrita desde la ejecución del programa en los dispositivos que se utilizarán.

Para nuestra arquitectura todos los dispositivos son de bajo recurso y habrá programas en ejecución en los nodos sensor, el nodo sumidero y la SBC, es por tanto de suma importancia en esta memoria que no haya problemas de colisión entre el Stack y el Heap ocasionando un “cuelgue”, creados por una mala programación o por una falta de espacio mínimo para un programa requerido.

Ilustración 6 - Partes de una memoria SRAM



Fuente: Elaboración propia

4.1.1.4.2 MEMORIA EEPROM (Electrically Erasable Programmable Read-Only Memory)

Tipo de memoria ROM que podemos programada, borrada y reprogramada eléctricamente desde el programa de un microcontrolador. Memoria no volátil después de una pérdida de alimentación.

4.1.1.4.3 MEMORIA FLASH

En realidad, es una evolución de la EEPROM, que puede programarse, reprogramarse y borrarse electrónicamente, esta puede utilizarse como una unidad de almacenamiento independiente (bootloader), donde se guarda el código ya compilado o puede hacer de disco duro en un dispositivo de poco performace como son los nodos o SBC.

	Memoria SRAM	Memoria FLASH	Plataforma software soportada
BTnode	64k	128k	BTnut y TinyOS
Iris	8k	128k	Moteworks y TinyOS
MicaZ	4k	128k	Moteworks y TinyOS
TelosB	10k	512k	Contiki, TinyOS, SOS y Mantis
TinyNode	8k	512k	TinyOS

Tabla 8 - Cuadro comparativo de tipos de memoria en los nodos

En nuestro caso los tipos de memorias en los nodos es muy limitada, a continuación, mostramos la capacidad de memoria que contiene cada tipo de nodo descrito en el cuadro comparativo siguiente, ahí podemos observar que lo más resaltante es la capacidad máxima de 64k de SRAM y de 512 de Flash (Ver tabla 8).

4.1.1.5 INTERFASES

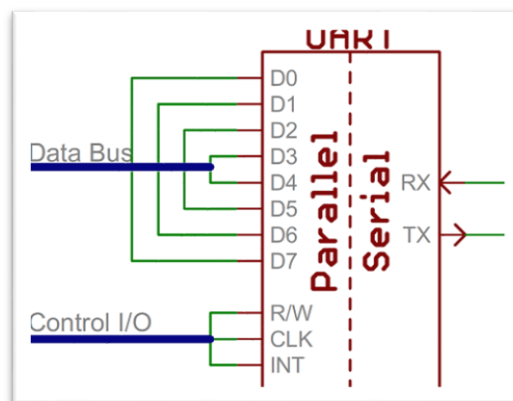
Los nodos presentan las siguientes interfases UART, SPI, I2C (sirven para comunicarse con los sensores) y también cuentan con entradas y salidas de propósito general.

4.1.1.5.1 INTERFASE UART

Es una interfase cuya finalidad es convertir los datos de la PC de formato paralelo en un formato serie hacia el exterior, este importante recordar que este dispositivo controla los puertos serie. También puede realizar el sentido inverso, de transformar los datos recibidos en formato serie en formato paralelo, existe un puerto UART por cada puerto serial.

Un UART dual, o DUART, combina dos UART en un solo chip. El soporte de velocidades de transmisión Kbps (Kilobits por segundo), se dará por los tipos de UART.

Ilustración 7 - Puerto paralelo y serial de un UART



Fuente: autor desconocido 2012

4.1.1.5.2 INTERFAZ I2C Inter-Integrated Circuit

Es un protocolo de comunicación serial que toma e integra lo mejor de las interfaces SPI e UART, básicamente con esta interfaz podemos tener varios chips maestros que controlan varios chips esclavos, es de gran utilidad cuando queremos controlar varios microcontroladores.

Se trata de un bus con dos hilos, una línea de datos (SDA) y una línea de reloj (SCL). La comunicación es de forma sincronizada bit a bit por una señal de reloj .

Figura 9 Interface I2C

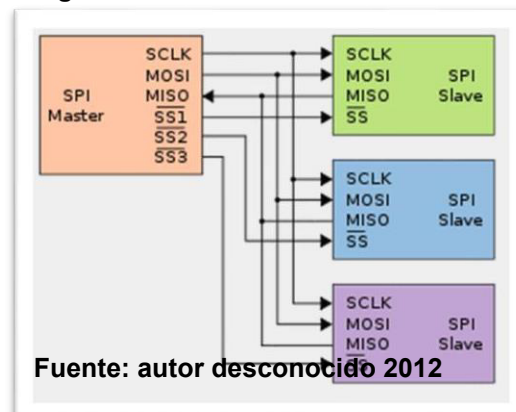


Fuente: autor desconocido 2012

4.1.1.5.3 INTERFAZ SPI Serial Peripheral Interface

Es un protocolo de comunicaciones, usado para la transferencia de información entre dispositivos electrónicos. El bus de interfaz de periféricos serie es un estándar para controlar casi cualquier dispositivo electrónico digital que acepte un flujo de bits serie regulado por un reloj (comunicación sincrónica). Utiliza más pines que la I2C, pero permite una velocidad muy superior, ideal para chips con relativa alta tasa de transferencia

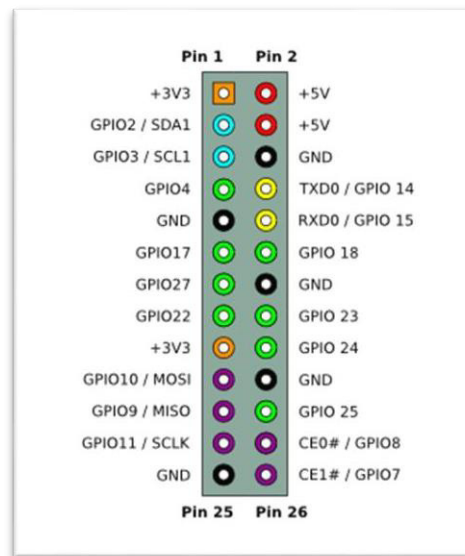
Figura 10 Interfaz SPI



4.1.1.5.4 INTERFAZ GPIO General Purpose Input/Output

Los pines GPIO tendrán realmente un propósito definido después de la programación, y no se utilizan de forma predeterminada. Podremos ver su utilidad en un momento de necesidad de contar con un puñado de líneas digitales de control adicionales, si no lo tuviéramos tendríamos que organizar circuitos adicionales para proporcionarlos generando pérdidas de tiempo en su adaptación.

Figura 11 Interfaz GPIO



Fuente: autor desconocido 2012

	Interfaces
BTnode	UART, SPI, I2C, GPIO
Iris	UART, SPI, I2C, GPIO
MicaZ	UART, SPI, I2C, GPIO
TelosB	UART, SPI, I2C, GPIO
TinyNode	UART, SPI, GPIO

Tabla 9 - Cuadro comparativo de las interfaces en los nodos

En este cuadro comparativo (Ver tabla 9) vemos que casi todos los nodos tienen las cuatro interfaces importantes para poder interactuar con otro dispositivo.

Así finalmente, terminado el análisis de sus partes del nodo, pondremos una puntuación mínima y máxima por su característica y un valor ponderado por la necesidad técnica en la realización de la tesis (Ver tabla 10).

VALOR	CONDICION	NOTA
Bajo	Cumple la condición mínimamente	1
Medio	Cumple la condición de manera mediana	2
Alto	Cumple la condición de manera completa	3

Tabla 10 - Nota Metodológica

Consiguiendo los resultados siguientes:

N°	Módulos analizados	BTnode	Iris	MicaZ	Telos B	TinyNode
1	Microcontrolador	1	3	1	2	2
2	ADC	3	2	2	3	3
3	Radio	2	3	2	2	2
4	Memoria	3	2	1	2	2
5	Interfaces	3	3	3	3	1
	TOTAL	12	13	8	12	11

Tabla 11 - Cuadro de valoración de los nodos

El nodo a utilizar en esta plataforma es Iris, ya que como se ve en la tabla tiene el máximo puntaje en la evaluación (Ver tabla 11).

4.1.2 SELECCIÓN DE LA PLACA SENSORA PARA EL NODO SENSOR

Una vez realizado la selección del nodo según los criterios antes descritos, llegamos a determinar que la mejor opción sería el nodo Iris por las características antes descritas y comparadas, por consiguiente, se tiene que entender ahora que la placa sensor está condicionada a la conexión con el nodo Iris. Tenemos que entender que el nodo sensor es la unión entre el nodo Iris y una placa sensor.

Criterio 1: Cantidad diferente de sensores

Se requerirá una placa sensor que contenga varios tipos de sensores, para las pruebas necesitamos la mayor cantidad de estas como: sensor de luz, temperatura, humedad, ...etc.

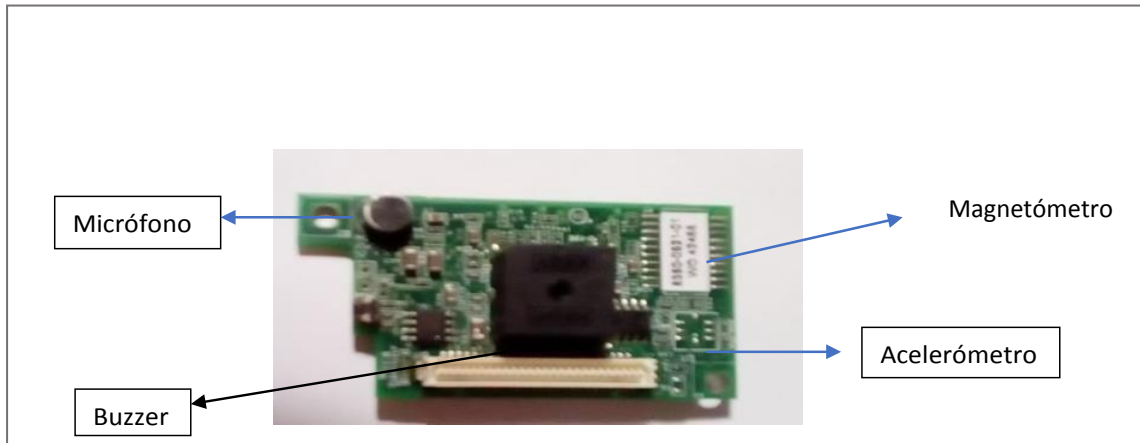
Como ya se mencionó se puede agregar a cada nodo una placa de sensores, y serán conectada a la placa principal a través del conector Molex de 51 pines, que permite medir variables del entorno.

4.1.2.1 Placa MTS310CB

Esta placa sensor tiene una gran variedad de tipos de detección (Ver figura 12). Estos tipos incluyen luz, temperatura, un acelerómetro y un zumbador que emite pitidos. Esta placa sensor puede ser utilizada en los nodos Iris, Mica2, MicaZ , de la compañía Crossbow.

Como el consumo de energía es importante en los nodos como en la SBC, por defecto la tarjeta viene con la opción del sensor apagado, teniendo mecanismo de control de energía.

Figura 12 Placa MTS310CB



Fuente: Elaboración Propia

4.1.3 SELECCIÓN DE LA PLACA PUERTA DE ENLACE PARA EL NODO SUMIDERO

Una vez realizado la selección del nodo según los criterios antes descritos, llegamos a determinar que la mejor opción sería el nodo Iris, por consiguiente, se tiene que entender ahora que el nodo sumidero está condicionada a la conexión con el nodo Iris. El nodo sumidero consta del nodo iris al cual se le agrega una placa (puerta de enlace) que permita la conexión física y lógica con la SBC.

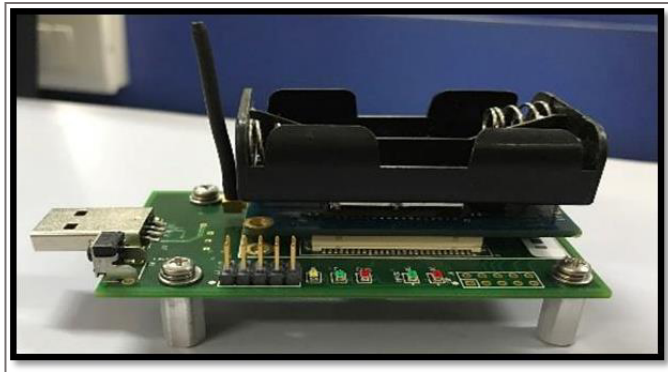
Criterio 1: Bus de comunicación -USB

Se requerirá una placa (puerta de enlace) que contenga un bus de comunicaciones (USB) que sigue una norma y una versión determinada para poder conectar, comunicar y dotar de energía eléctrica entre el componente SBC y el nodo sumidero.

4.1.3.1 Placa MIB520

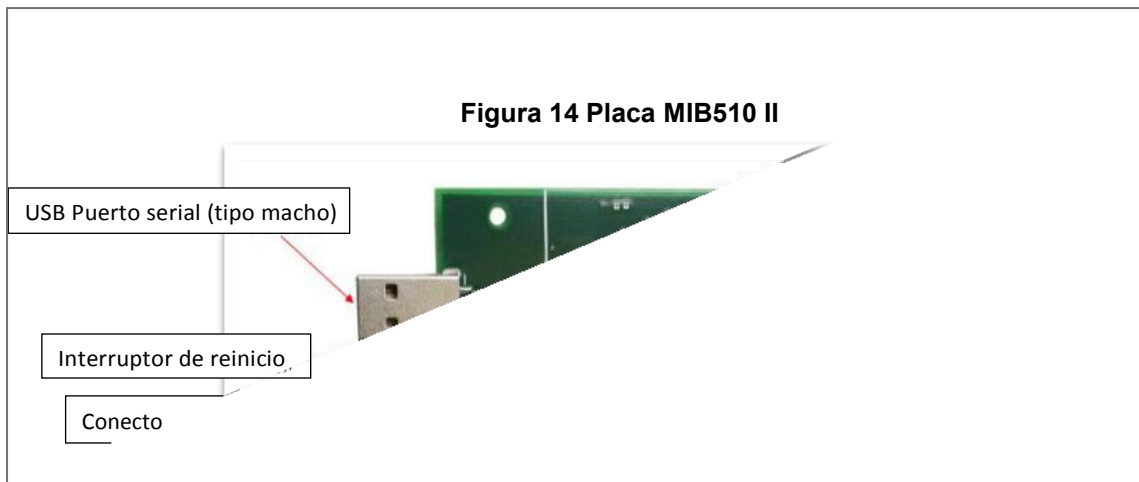
Esta placa tiene dos objetivos básicos gracias a su conectividad USB. Primero, para la comunicación entre el nodo sumidero con la estación base y segundo después de programar permite cargar código ejecutable en los nodos sensor y nodo sumidero, todo esto a través del USB macho. (Ver figura 13).

Figura 13 Placa MIB510 I



Fuente: Elaboración Propia

Figura 14 Placa MIB510 II



Fuente: Elaboración Propia

4.1.4 SELECCIÓN DE LA SBC PARA LA IMPLEMENTACION DE LA ARQUITECTURA PROPUESTA

Para realizar la evaluación de la SBC en la implementación de la arquitectura presentado anteriormente, se muestran los siguientes criterios de evaluación.

Criterio 1: Procesadores ARM

Es una necesidad que este componente tenga arquitectura ARM dado que se encontrara más soporte para la instalación de GNU-Linux como sistema operativo.

Criterio 2: Tipos de memoria

Se requerirá que contenga memoria tipo Flash necesaria para el booteo normal, además que pueda soporte el trabajo de cualquier programa que se ejecute en ella.

Criterio 3: Conexión de red

Se requerirá que el componente pueda realizar una conexión inalámbrica de corto alcance (bluetooth), mediano alcance (wifi) y largo alcance (GSM/GRPS)

Criterio 4: Conectividad

Se requerirá que la SBC pueda incluir necesariamente interfaz USB para la conexión con el nodo sumidero y de complemento interfaz de tipo HDMI.

Criterio 5: Costo bajo

Es necesario que este componente con sus requerimientos mínimos tenga el precio más bajo de todas las SBC.

Criterio 6: Soporte oficial del SO

Es indispensable el soporte oficial de alguno de los dos sistemas operativos líderes en el desarrollo de aplicaciones para dispositivos de baja gama que es GNU-Linux y Android.

En el mercado actual de mini computadoras existen muchos modelos, pero buscamos los que están basados en el objetivo de esta tesina (Ver tabla 12), principalmente a las cuatro características antes descrita en el marco teórico, sobre todo a la versatilidad en su hardware y su precio bajo coste, es por esto que según el cuadro comparativo mostrado de las familias de SBC nos decantamos

por dos familias las Raspberry Pi (el más vendido y popular) y Orange Pi (competencia real del Raspberry Pi) y de estas dos compañías escogeremos dos modelos de las más bajas gamas para realizar la comparación técnica.

Obteniendo los resultados siguientes:

FAMILIA SBC	HummingBoard	Orange Pi	Raspberry Pi	Banana Pi
Procesadores ARM	SI	SI	SI	SI
Memoria Flash	SI	SI	SI	SI
Conexión de red Bluetooth	SI	SI	SI	NO
Conexión de red WIFI	SI	SI	SI	SI
Conexión de red GSM-GRPS	NO	SI	NO	NO
Conectividad	SI	SI	SI	SI
Costo bajo	NO	SI	NO	NO
Soporte oficial	NO	SI	SI	NO

Tabla 12 - Cuadro de análisis de las familias de SBC

De estas dos familias necesitamos un modelo de cada compañía, la de más baja características que sea suficiente para poder actuar como mini estación base y que se pueda acoplar con la puerta de enlace para la captación de datos. Estos modelos vienen hacer la Orange Pi IoT 2g y la Raspberry Pi Zero W.

4.1.4.1 Orange PI 2G IoT vs Raspberry PI Zero W

El Orange Pi 2G IoT se compara con la recientemente lanzada Raspberry Pi Zero W por sus características mínimas dentro de sus respectivas familias, ambas cuestan más o menos unos S/33 soles o \$ 10 dólares.

Orange Pi 2G-IoT dispone de un procesador ARG Cortex-A5 de 32 bits de 1 GHz con un procesador de gráficos Vivante GC860, de memoria con 256 Mb de RAM y un puerto GPIO de 40 pines para múltiples funciones. Tiene una ranura de expansión para tarjetas microsd, y dos puertos USB. Como todos los modelos de Orange Pi, el Orange Pi 2G-IoT cuenta con un botón de encendido y reset.

El Orange Pi 2G-IoT tiene menos cantidad de memoria RAM que Raspberry Pi Zero W que tiene 512 MB, específicamente la mitad de memoria 256 MB, con respecto al tamaño es casi igual y en el precio difieren solo un poco. Por otro lado, el módulo de tarjeta SIM es algo primordial cumplir con el objetivo de nuestra tesis que hará que podamos crear proyectos ubicuos y móvil por un bajo precio. Si la memoria RAM es un problema para un procesamiento duro de datos, siempre se puede disponer del puerto GPIO para conectarlo con otro hardware, implementando una dispositivo más completo y potente.

El dispositivo puede ejecutar Android, Ubuntu, Debian o Raspberry Pi, de acuerdo con el listado de Orange Pi 2G-IoT de la página oficial, sin embargo, en una Raspberry Pi, hay dos sistemas que no podemos instalar al no tener soporte oficial con Ubuntu y Android, aunque desde el año pasado hay un intento de instalar Ubuntu mate en los Raspberry Pi, la gran ventaja del Orange Pi radica en que es compatible con sistemas GNU-Linux conocidos en el mercado.

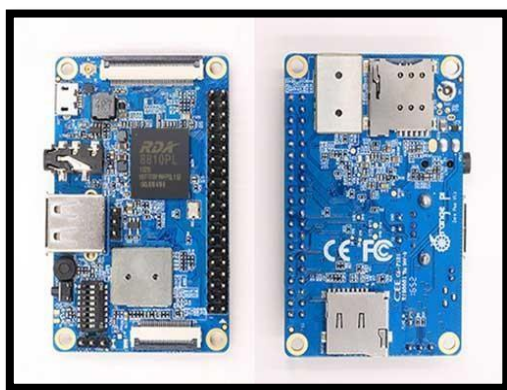
Ahora, Orange Pi 2G-IoT, va en la dirección de adicionar más conectividad, brindando una antena 2G que admite conexiones de datos GSM / GPRS y, por supuesto, hay una ranura para insertar una tarjeta SIM, además de ofrecer Wi-Fi 802.11b / g / n y Bluetooth 2.1 que es importante, porque Raspberry Pi Zero W solo cuenta con Wi-Fi y Bluetooth.

Analizando ambos concluimos que estas SBC son muy similares (Ver tabla 13), aunque el Raspberry Pi Zero W puede llegar incluso a ser superior al Orange Pi 2G-IoT en memoria RAM y en tener un mayor soporte en la comunidad en el desarrollo de aplicaciones en el mundo, Sin embargo, para tener una plataforma completa en conectividad, necesitamos conexión por lo menos en telefonía móvil a nivel 2G algo fundamental que Orange PI ha tenido además es el soporte oficial de dos sistemas operativos líderes en el desarrollo de aplicaciones que es Debian, Ubuntu y Android . Por consiguiente, se decidió la utilización del Orange Pi 2G-IoT para la plataforma propuesta en esta tesis.

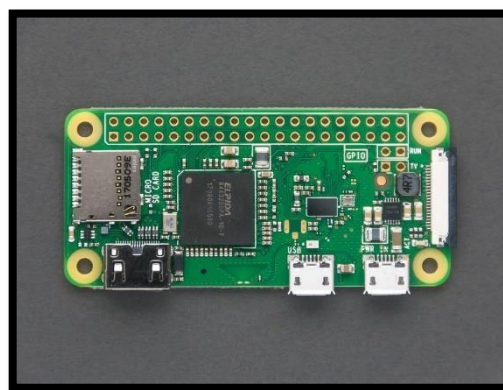
	Orange PI 2G IoT	Raspberry PI Zero W
Procesador	CPU: 32 bits ARM Cortex-A5, 1 GHz	CPU: 32 bits ARMv6 ARM1176JF-S, 1 GHz
Memoria	Integrada 256MB SDRAM LPDDR2	Integrada 512MB SDRAM LPDDR2
Conectividad	WIFI: 802.11 .b Bluetoooh: 2.1 GSM / GPRS: 4 frecuencias, modulo 2G	WIFI: 802.11 n Bluetoooh: 4.0 GSM / GPRS: No soporta
Salidas	HDMI: 1080p HD video USB: 1 micro-USB, OTG	HDMI: 1080p HD video USB: 1 micro-USB
Entradas	GPIO: 40 pines machos en la placa	GPIO: 40 pines por colocar
Sistema Operativo	Linux	Linux

Tabla 13 - Cuadro resumen comparativo SBC - Orange PI 2G IoT vs Raspberry PI Zero W

Figura 15 Arquitectura Orange PI y Raspberry



Orange Pi 2G -



Raspberry Pi Zero

Fuente: Elaboración Propia

4.1.5 SELECCIÓN DE LA MEMORIA FLASH PARA LA INSTALACION DE GNU-LINUX EN EL SBC

Este es un componente que se necesitará para la implementación de la arquitectura, la memoria flash contendrá la imagen del sistema operativo GNU-Linux. Existe mucha incertidumbre en estos componentes por ser de diferentes tipos y características que hay en el mercado, y el proveedor de la SBC no nos dice ni recomienda con cual funcionará en óptimas condiciones, por tanto, el criterio de evaluación para esta tesis se determinó, al probar los diferentes tipos de este componente mediante el ensayo y error.

Criterio 1: Capacidad de almacenamiento

Es necesario que el componente tenga como máximo 8 Gb de almacenamiento para su ejecución, de más capacidad no soportaría la SBC, información muy importante para el descarte.

Criterio 2: Velocidad de transferencia

Se requerirá una velocidad de transferencia optima al ingresar datos, si la velocidad no es adecuada será un impedimento para la transferencia del sistema GNU-Linux.

Criterio 3: Original

Este es un criterio que quizás técnicamente no se tenía en cuenta, pero somos un país donde las imitaciones de marcas abundan, especialmente en componentes electrónicos, es así que las marcas que son copias no realizan una verdadera velocidad transferencia como indica su especificación.

Esto significa que dependiendo de la clase de nuestra tarjeta nos irá mejor o peor para nuestra instalación de nuestro sistema operativo GNU-Linux – Debian-, Tinyos y otros programas, que se instalaran en la tarjeta para nuestra SBC (Orange PI 2g IoT). En nuestro caso intentamos con cuatro microSD, para ver y corroborar entre la capacidad de almacenamiento y velocidad de lectura/escritura.

- (a) Kingston 16Gb clase 4
- (b) SanDisk 8Gb clase 4
- (c) Toshiba 8GB Class10
- (d) Kingston 8Gb clase 10
- (e) Sony 8Gb clase 10

Al intentar con la memoria (a), hubo dos inconvenientes que se pudieron reconocer, una era la capacidad de almacenamiento, según especificaciones del Orange PI 2g IoT soporta solo 8GB de almacenamiento, por más que quieres colocar una tarjeta con más gigas, no lo reconocerá habrá incompatibilidad.

Memoria (b), si bien es cierto tuvo la capacidad de almacenamiento correcto, la velocidad de transferencia (clase 4) fue quien impidió la instalación, quedando en estado latente al querer ingresar datos a la tarjeta.

Memoria (c), en esta memoria se tuvo los dos requerimientos correctos tanto de almacenamiento como de velocidad, pero en la clase 10 existen también subclases y este tenía la velocidad mínima de escritura lo que impidió la transferencia de datos.

Memoria (d) se descubrió que esta memoria era una copia de la marca original (genérico), se entiende que el nivel de transferencia no es el real como indicaba su estuche, por ese motivo en la instalación de la imagen en el Orange PI 2g IoT, se quedaba a mitad del proceso de instalación.

Memoria (e), después de muchos intentos con varias memorias, logramos tener una memoria con la capacidad de almacenamiento requerido, así como la velocidad de transferencia y algo importante de marca original que es la garantía de transferencia real.

4.1.6 SELECCIÓN DE LA INTERFAZ TTL PARA INTERCONEXION DEL SBC CON UNA PC

Otro componente necesario para la implementación de la arquitectura, es la interfaz TTL que nos permitirá conectar la SBC con cualquier computadora que permita la instalación de la imagen del sistema operativo. También existe mucha incertidumbre en este componente por ser de diferentes tipos y características, el proveedor de la SBC tampoco nos dice ni recomienda cual funcionará en óptimas condiciones por tanto el criterio de evaluación para esta tesis ha sido el de ensayo y error de los diferentes tipos de componentes probados.

Criterio 1: Soporte del driver

Es necesario que el componente tenga como máximo 8 Gb de almacenamiento para su ejecución, de más capacidad no soportaría la SBC.

El adaptador interfaz USB-TTL permite comunicar a través del puerto universal USB de tu computadora con la UART de un dispositivo que tenga este tipo de interfaz. En nuestro caso ideal para comunicarse a través de la UART nuestra computadora con el Orange Pi en la instalación de Debian de manera sencilla a través de un puerto USB.

La interfaz UART se utiliza para la descarga de sistemas operativos (mediante bootloader), para la configuración de los parámetros en dispositivos electrónicos, haciendo uso de nuestro teclado y pantalla del pc.

El adaptador interfaz USB TTL puede utilizar 5 y 3.3 volts de los sistemas electrónicos, pudiendo elegir el voltaje correspondiente mediante un jumper.

Se escogieron dos interfaces que se encontró en el mercado electrónico PI2303HX y CH340G.

PI2303HX

Este chip (Ver figura 22) crea un COM virtual. No solo lo podemos usar para instalar el Debian en la SBC, sino también para conectar cualquier dispositivo que maneje el mismo protocolo de comunicación, (Arduino, módulos bluetooth, lectores RFID, etc.) y se pueda alimentar con los voltajes antes mencionados, otorgando hasta 500 mA de corriente. El inconveniente en esta versión del chip (HX) es que requiere la instalación de drivers en algunos sistemas operativos, primero se probó con el sistema operativo Windows 10 y aun instalando el driver hubo inconvenientes porque no reconocía el interfaz especificó entendemos que es por las versiones de actualización de Windows.

Figura 16 PI230HX



Fuente: Elaboración Propia

CH340G

Tiene casi la misma característica que el modelo PI2303HX, es relativamente nuevo en el mercado, pero que han hecho que los precios hayan bajado considerablemente en dispositivos electrónicos. Es importante mencionar que el buen precio de chip se debe a los circuitos integrados seleccionados, sin embargo, esto no afecta su funcionalidad ni pierde características con respecto a otros chips originales, clones o compatibles. Gracias a este componente es que el dispositivo como Arduino ha bajado su precio considerablemente. Los nuevos chips CH340G (Ver figura 23) tienen un inconveniente práctico y es que ni Windows ni OSX lo soportan oficialmente a día de hoy y para activarlo hay que instalar un driver, pero lo más destacable para nosotros es que en Linux soporta

de forma nativa el chip por lo que no es necesario instalar nada. Tanto en Windows como en Linux sí reconoció este TTL.



Por tanto, después de haber adquirido estos dos componentes, para la implementación de la plataforma, el TTL CH340G reconoció el driver instalado para Windows 10 y reconoció automáticamente para cualquier sistema operativo GNU-Linux.

4.2 SELECCION DEL SISTEMA OPERATIVO PARA LA PLATAFORMA

Como en tesis analizadas, hemos visto que se utiliza Windows o Linux en una PC como estación base de la red de sensores, en ese caso no existe inconvenientes en resolver algún tipo de problema de cálculo o de interpretación con los dispositivos, porque estas tenían una arquitectura de procesadores potentes como las amd64(64 bit) o i386 (32 bit) , lo mismo sucede en la utilización del sistema operativo TinyOS cuando se instala en la PC no hay inconveniente en la performance del sistema operativo.

4.2.1 SISTEMA OPERATIVO EN LA SBC

En nuestra plataforma propuesta la estación base estará compuesta por un microprocesador de placa única (SBC), de arquitectura ARM que como mencionamos tiene características de bajo consumo.

Para este tipo de procesadores es indispensable utilizar como sistema operativo GNU-Linux, porque se adapta a este tipo de requerimientos donde el hardware es limitado. Existe innumerables GNU-Linux que se pueden utilizar como Red Hat, Suse, Mandrake, Ubuntu, Debian y otros que viene incorporado y configurado el sistema TinyOS en Ubuntu como Xubuntu .

Aun hayamos decidido utilizar cualquier GNU-Linux como sistema operativo de la SBC es fundamental no solamente conocer si el microprocesador trabaja a 32 o 64 bit sino tenemos que tomar en consideración su arquitectura, porque es en ese punto es donde iniciamos la búsqueda de un tipo de GNU-Linux determinado para nuestra plataforma, dado que el núcleo de Linux se relaciona fuertemente con el hardware utilizado. Dentro de la arquitectura ARM tenemos arm64, armel y armhf , aquí en la siguiente tabla describimos de manera general las características de cada uno.

Armel 32 bits	EABI ARM	La más antigua de las adaptaciones actuales de Debian a ARM, tiene soporte para CPUs ARM little-endian compatibles con el juego de instrucciones v4t.
armhf 32 bits	ABI ARM de punto flotante	Muchas de las modernas placas y dispositivos ARM de 32 bits se lanzan con una unidad de punto flotante (FPU), pero la adaptación armel de Debian no les saca provecho. La adaptación armhf fue (hard float) iniciada para mejorar esta situación y también para aprovechar otras características de los nuevos procesadores ARM. Esta adaptación requiere al menos un procesador ARMv7 con soporte de punto flotante Thumb-2 y VFP3-D16.
arm64 64 bits	ARM de 64 bits (AArch64)	La versión 8 de la arquitectura ARM incluye AArch64, un nuevo juego de instrucciones de 64 bits. Desde Debian 8.0, se ha incluido la adaptación arm64 en Debian, para dar soporte a este nuevo juego de instrucciones en procesadores como el Applied Micro X-Gene, AMD Seattle y Cavium ThunderX

Tabla 14 - (Oficial Debian, 2018)

Podríamos escoger una imagen de la cuenta oficial de Debian, pero como se menciono es determinante el núcleo o kernel de Linux que encaje al tipo de arquitectura de microprocesadores que nos garantizara que se pueda ejecutar de manera segura y confiable, de ahí radica la importancia de que este SBC tenga soporte oficial en alguna de estas versiones de GNU-Linux.

Decidiendo que sea de algún GNU-Linux que tenga soporte oficial, encontramos dos alternativas dentro de la página oficial de Orange PI, Ubuntu y Debian, pero como se sabe Ubuntu fue creado para ser un sistema de usuario y Debian por su parte para ser usado como sistema base de servidores, por su seguridad, por su performance, por su soporte de muchas arquitecturas de microprocesadores y la gran cantidad de comunidad de ayuda.

Es por ello que se decide trabajar por la imagen de Debian que se encuentra en la página oficial de Orange PI, porque además esta imagen estará ya adaptada a cualquier tipo de estas arquitecturas arm64, armel y armhf, de nuestro OrangePI 2G IoT.

4.2.2 SISTEMA OPERATIVO EN LOS NODOS

Tinyos fue creado para ser utilizada en una PC como estación de trabajo en arquitecturas como amd64 e i386, y no para arquitecturas ARM. Ahora es momento de analizar el tipo de arquitectura que soporta el Cortex A5 del Orange PI 2G IoT para determinar qué tipo de arquitectura utilizara TinyOS en Debian.

Una vez que hemos determinado utilizar Debian como sistema operativo en la SBC, ahora toca elegir el sistema operativo de TinyOS , si bien es cierto al descargar de la página oficial de Orange no importo saber la arquitectura, porque Debian tiene soporte oficial en el Orange PI mediante una imagen, con TinyOS tendríamos que saber si el núcleo soporta armhf o armel.

Ya se pudo determinar las características del microprocesador ARM como muestra la tabla 2, ahora para para descargar la imagen de Debian que corresponde tendremos que decidir cual tipo de imagen descargar, si es armel o armhf , la siguiente pregunta que nos hacemos en cuanto al microprocesador, es determinar que hace que un procesador pase de arquitectura amrel (para gama de dispositivos más antiguos de 32 bits) a armhf (dispositivos más potentes de 32 bits) y la respuesta es que esta arquitectura pueda soportar el punto flotante (BrazoHardFloatPort) que soporta operaciones de mucha precisión, este es el motivo de analizar las características del microprocesador ARM, fundamentalmente en su núcleo Cortex 5A . existe poca documentación acerca de las características de cada núcleo de la familia Cortex-A, porque los procesadores ARM utilizados actualmente en el mercado suelen ser modelos ya de 64 bit y sus arquitecturas soportan muchas prestaciones.

Familia	Arquitectura	Núcleo
Cortex-A	ARMv7-A	Cortex-A5, A7, A8, A9, A12, A15

Tabla 15 - Cuadro con los núcleos de la familia Cortex A

Actualmente, el puerto armf de Debian requiere una arquitectura mínima de ARMV7 con **Thumb-2** y **VFP3-D16**, en ARMv5 se introdujo un conjunto

opcional de instrucciones de punto flotante conocido como vector Floating Poing VFP, que se extendió a mejores versiones con el tiempo a, VFP2, VFPv3-D16 y VFPv3 + NEON. En la familia ARMV7 encontramos un sin número de núcleos aparte del Cortex A5, tenemos el Cortex A7, A8, A9, etc., es necesario identificar si estas dos propiedades el Thumb-2 y VFP3-D16 se encuentran en el núcleo del Cortex A5 del OrangePI 2G IoT. Con respecto a Thumb se refiere al conjunto de instrucciones de 16 bits de longitud y el Thumb-2 introdujo instrucciones mixtas de 16 y 32 bits de longitud.

Núcleos ARMv7-A	FPU (unidad de coma flotante)	FPU (registros)	Thumb-2
Cortex-A5	VFPv4 (optional)	16 × 64-bit	Si
Cortex-A7	VFPv4	16 × 64-bit	Si
Cortex-A8	VFPv3	32 × 64-bit	Si
Cortex-A9	VFPv3(optional)	(16 o 32) × 64-bit	Si
Cortex-A12	VFPv4	32 × 64-bit	Si

Tabla 16- Cuadro con los núcleos de la familia Cortex A

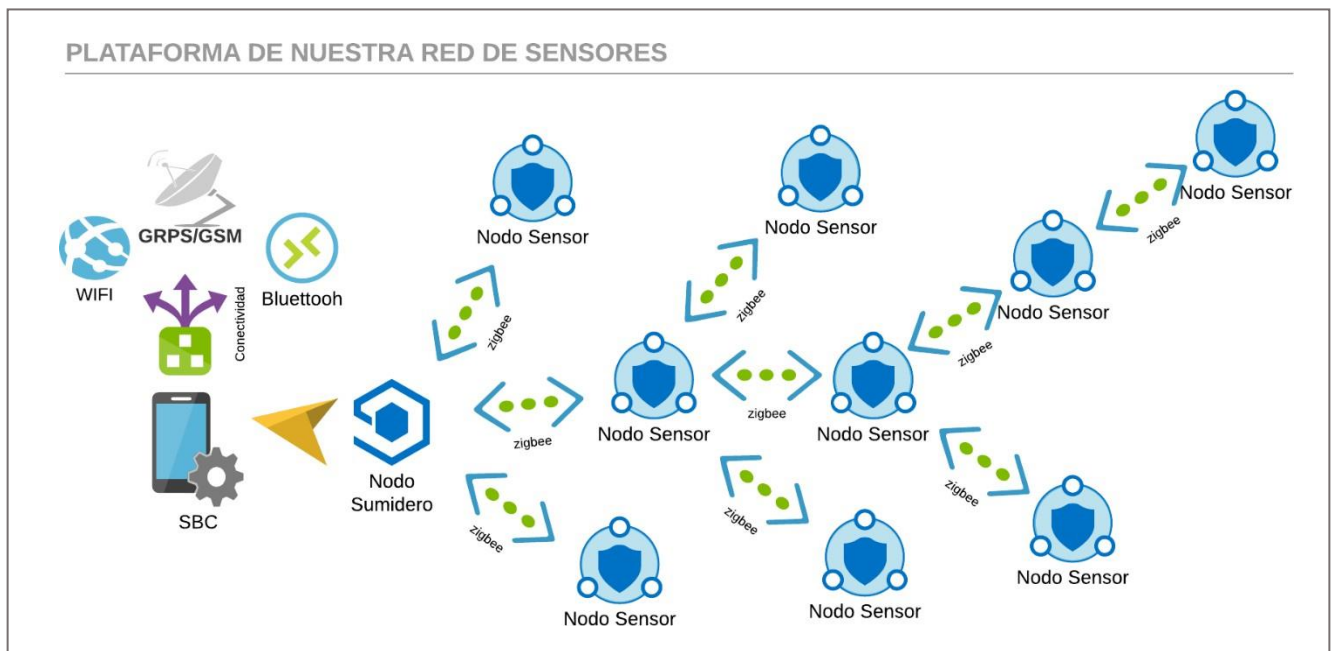
Luego del estudio del núcleo cortex 5A, nos queda afirmar que cumple con los requisitos que soporta la arquitectura armhf en teoría. Será en la implementación donde se determine con certeza del 100% la compatibilidad del núcleo con armhf., dado que este núcleo está al límite en su performance para ejecución de aplicaciones.

Por tanto, su instalación de TinyOS será que contemple la arquitectura armhf que soporta el punto flotante.

4.3 SOLUCIÓN

Ahora elegiremos entre los componentes propuestos, los componentes que se puedan adaptar a la SBC y así implementar una plataforma inteligente y ubicuo dando solución al problema planteado.

Ilustración 8 - Solución Propuesta



Fuente: Elaboración Propia

La Ilustración 8 expone la solución a los inconvenientes planteado, se ha desarrollado la solución en la Etapa I, que se determinó la viabilidad de los componentes que integrarían la plataforma. Se pudo analizar diferentes modelos de SBC, eligiendo Orange PI 2g IoT. En la selección de la SBC, hubo diversos criterios, como, la conectividad con diferentes redes (TCP/IP, GPRS, GSM), el protocolo de sus puertos USB y la compatibilidad con GNU-Linux; además de la compatibilidad de TinyOS con la arquitectura del microprocesador ARM Cortex-A5 32 bits del Orange PI 2g IoT.

También se determinó la elección de ciertos elementos como el convertidor de USB a Serial TTL para comunicar el Orange PI 2g IoT con la PC y la memoria MicroSD en la sincronización de velocidad de datos al instalar la distribución Linux en el Orange PI 2g IoT.

Otro de los componentes que se analizó fueron los diferentes tipos de nodos sensor, analizando su performance, eficiencia en el consumo, radio, etc. eligiendo el nodo de la red inalámbrica, Iris 2.4GHz (Atmel ATmega 1281) y la puerta de enlace MIB520 que son de la compañía Crossbow; con respecto a la puerta de enlace y la SBC se tuvo que verificar la compatibilidad del USB de los dos componentes y si faltaba archivos de configuración en Debian para el reconocimiento del MIB520 con el Orange PI 2g IoT.

Luego, en la Etapa II, se realizará la conexión de los componentes externos (memoria microSD y la interfaz TTL) para la instalación de la imagen del sistema operativo GNU-Linux en la SBC, esta etapa fue una de las más críticas por la arquitectura tipo ARM del microprocesador de la SBC, además se tuvo que adquirir driver de reconocimiento para que puedan “dialogar” los componentes con la SBC, una vez logrado instalar la imagen Linux se procederá a configurar e instalar otros programas como se verá en el próximo capítulo, consecuentemente en la Etapa III, se instalará las herramientas y los programas necesarios en la imagen GNU-Linux Debian como TinyOS (sistema operativo de los nodos sensores) para conectar el nodo sumidero con la SBC y programas para la comunicación con otras redes (TCP/IP, GPRS, GSM).

4.4 METODOLOGÍA

Para la implementación de la arquitectura propuesta, tenemos que mencionar que muchos de estos pasos se han podido lograr gracias al ensayo y error en la implementación, porque no se encuentra mucha documentación de la SBC por ser la gama más baja dentro de las familias Orange PI, esto genero mucha confusión en el intento de configurar la SBC, a continuación, mostramos la secuencia de fases a desarrollar:

1. Modelo de la Plataforma

Lo primero que se hará, es definir el esquema del modelo lógico y modelo físico de la plataforma. Para ello, se describirá conceptualmente los módulos que conformaran en su conjunto.

1.1. *Estructura del modelo lógico*

En este punto se va a definir el objetivo funcional de los módulos que formaran parte de la plataforma, y la relación entre ellos. Los 4 módulos son: el de adquisición, transmisión, recepción y procesamiento de datos.

1.2. *Estructura del modelo físico*

Aquí se describe como las placas electrónicas a utilizar forman parte de varios módulos y como estas placas se vuelven componentes de la red de sensores, como el nodo sumidero que está conformado por el módulo de adquisición y el módulo de recepción, todo ello relacionado físicamente con la placa iris y la placa MIB520.

2. Configuración del Sistema Operativo GNU- Linux en el Orange PI

A continuación, se va a cargar completamente el sistema operativo GNU-Linux en la SBC, utilizando y conectando los dispositivos complementos como paso previo. Finalmente se terminará en esta primera parte con la instalación de la imagen GNU Linux -Debian- en Orange PI 2G-IoT.

2.1. *Formatear el SD Card*

El primer paso comienza con el formateo total de la memoria flash externa que incluirá la imagen GNU-Linux -Debian- en la SBC, por tanto, se necesita un software que proporcione un acceso rápido y sencillo a todos los formatos de tarjetas de memoria SD, SDHC y SCXC. *SD card formatter* está diseñado para que puedas deshacerte cuanta veces necesites todo el contenido almacenado en tu memoria SD.

2.2. *Grabar GNU-Linux al SD Card.*

Una vez que se haya formateado la memoria flash externa, se debe proceder a grabar la imagen GNU-Linux en la memoria para su booteo posterior, esto nos permite realizar gracias a este software *Win32 Disk Imager* que es de gran utilidad.

2.3. *Instalar de la imagen GNU-Linux en el Orange PI 2G-IoT*

En esta etapa, se realizará la instalación de la imagen GNU-Linux en la SBC, aquí primero debemos utilizar una herramienta llamada *PUTTY* que nos permita comunicar entre una computadora y el Orange PI para ello es necesario hallar la velocidad de comunicación entre estos dos dispositivos, es demás recalcar la necesita de saber mínimamente comandos de Linux para una correcta instalación. Esta etapa culminará una vez haya sido instalado la imagen y por ende instalado Debian en el Orange PI.

3. Operatividad de la SBC para la comunicación con el nodo sumidero

Seguidamente, se realizarán la instalación y configuración de programas que nos ayudaran a comunicar la SBC con el nodo sumidero a través de la puerta de enlace que contiene la interfaz dentro de TinyOS, además se instalara java como lenguaje de programación, porque ayudara en la interpretación de datos recibidos de la red de sensores, por las librerías en java que contiene TinyOS, de ahí la importancia de este sistema operativo. Finalmente, se ampliará en la instalación algunos utilitarios que serán de ayuda para esta parte.

3.1. *Configurar Wifi en Orange PI 2G-IoT*

En este paso se deberán explicitar cómo configurar el punto wifi en el Orange PI, para proporcionar acceso a internet, aquí es primordial tener una conexión directa con el router así como sus características, como tipo de cifrado, canal de transmisión entre otros.

3.2. *Instalar Java en Orange PI 2G-IoT*

El objetivo de este paso, es instalar java como lenguaje de programación porque TinyOS está fuertemente relacionado con java en la comunicación con los nodos sensor y nodo sumidero tanto así que contiene librerías para el testeo en la monitorización de la red a un nivel de abstracción simple, clases java como TOSBase, Oscilloscope, SerialForwarder la cual utilizaremos para nuestro testeo.

3.3. *Instalación de TinyOS en Orange PI 2G-IoT*

Una vez que se han establecido la instalación de java es indispensable instalar el sistema operativo TinyOS para realizar una comunicación entre los nodos sensor y la SBC a través de una puerta de enlace que hará de interfaz entre el nodo sumidero y el Orange PI, cabe recordar que los bits en la cual trabaja la arquitectura del microprocesador ARM del Orange PI debe ser igual a los bits que trabajara los sistemas operativos y programas que se instalaran en este.

4. Configuración y Operatividad en los nodos de la red

En este apartado se programa y configura los dos tipos de nodos de la red, utilizando software que nos habilita las librerías que contienen TinyOS.

4.1. *Compilación del software en el nodo sumidero*

Instalar y compilamos el programa llamado TOSBase, que ayudara a la recepción de datos de los sensores por el puerto serie.

4.2. *Compilación del software en el nodo sensor*

Instalar y compilamos el programa llamado Oscilloscope, que ayudara a la captación de datos de los sensores enviando estos al nodo sumidero que está preparado para recibir .

5. Integración de los componentes

Una vez realizado los pasos anteriores se deberá probar la integración de los cuatro componentes (un nodo sumidero, dos nodos sensores y un SBC) en ejecución, para realizar esta prueba resulta adecuado elegir una topología, la cual se escogió la topología estrella para el despliegue

CAPITULO V

APORTE PRÁCTICO

5.1 MODELO DE LA PLATAFORMA

En esta sección definimos el modelo de la plataforma, la estructura del modelo lógico y físico. El modelo lógico se refiere a la definición conceptual de las funciones de los módulos que integraran los componentes de las redes de sensores y el modelo físico son el conjunto de dispositivos relacionados con sus módulos que harán de nodo sensor, nodo sumidero y SBC.

5.1.1 ESTRUCTURA DEL MODELO LOGICO

Ahora se va a definir el modelo que se utilizará, describiendo conceptualmente cada módulo, esto dará inicio al entendimiento conceptual del nodo sensor, nodo sumidero y la SBC, gracias a la relación de conceptos de los módulos. Los 4 módulos son: el de adquisición, transmisión, recepción y procesamiento de datos.

MODULO DE ADQUISICIÓN

El módulo de adquisición recibirá los valores de los sensores, conforman en un solo encapsulado donde se encuentran físicamente los sensores, pudiendo medir las variables simultáneamente, es necesario para el control y sincronización de la adquisición un módulo de procesamiento, el cual recibe del sensor los datos.

MÓDULO DE TRANSMISIÓN Y RECEPCIÓN

El módulo de transmisión funciona como punto de acceso para los sensores, este módulo se comunica inalámbricamente con los dispositivos que tengan el módulo de recepción cuya función es el de recibir tramas con una dirección única para la identificación de estas

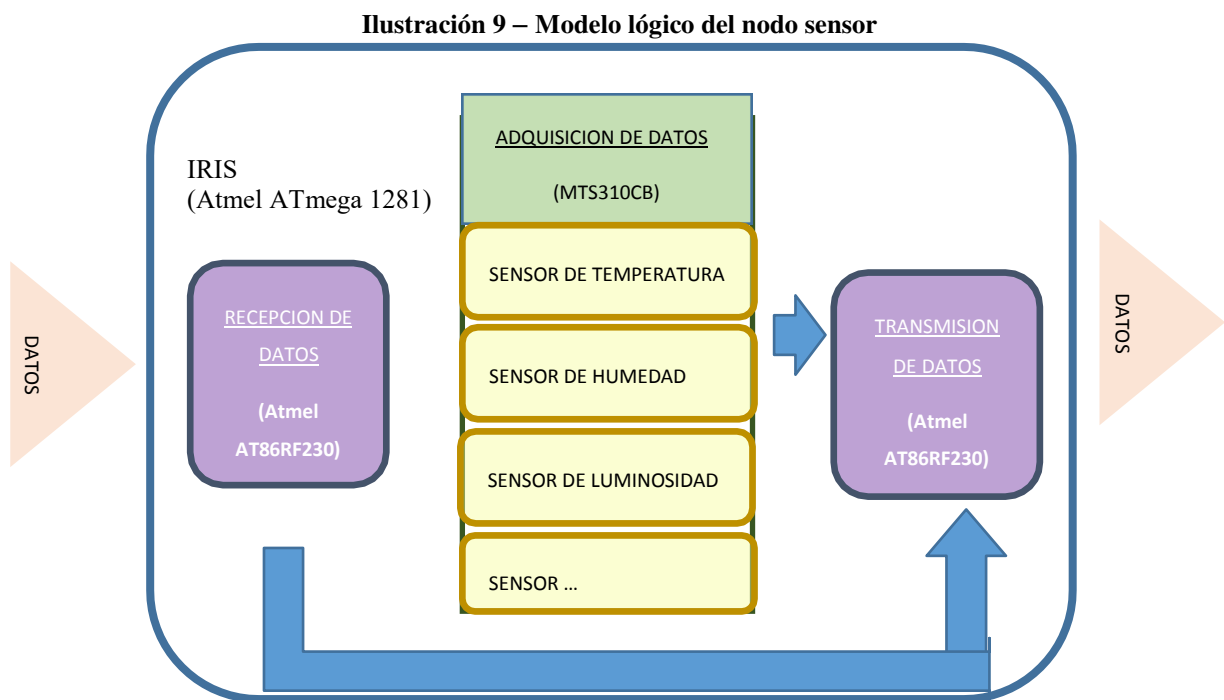
MÓDULO DE RECOPIACION

Este módulo está unido mediante el puerto serial al módulo de procesamiento de datos. Este tiene como función recopilar todas las tramas de datos enviados por el módulo de transmisión. El módulo de recopilación permite controlar y verificar que todos los módulos estén conectados y recibe los datos de los módulos de adquisición, verifica si hay alguna colisión o si el orden de la información está alterada.

MODULO DE PROCESAMIENTO DE DATOS

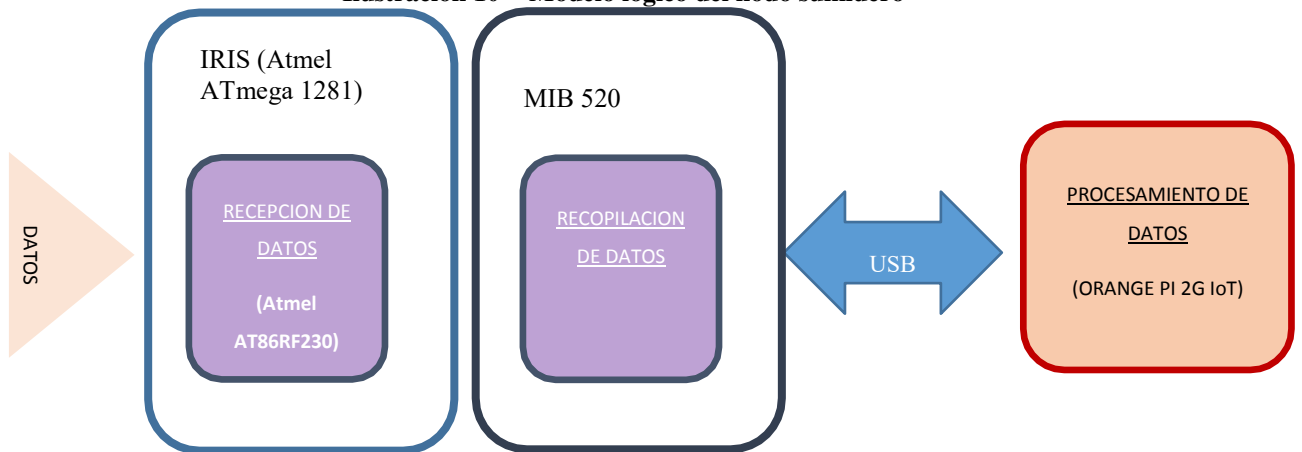
Se encarga de validar, analizar, convertir y clasificar los valores que envían los módulos de transmisión y adquiridos por el módulo de recopilación. Donde, para lograr este proceso ordena los datos obtenidos en secuencia y los convierte en tramas para luego ser enviados donde sea conveniente a través de lo que permita la conectividad, de corto alcance (Bluetooth), de mediano alcance (Wifi) y largo alcance (GSM/GPRS).

El diagrama mostrado en la Ilustración 9 y 10 muestra los módulos que componen el nodo sensor y el nodo sumidero.



Fuente: Elaboración Propia

Ilustración 10 – Modelo lógico del nodo sumidero



Fuente: Elaboración Propia

5.1.2 ESTRUCTURA DEL MODELO FISICO

Una vez que hemos delimitado conceptualmente el funcionamiento de los módulos, pasamos a definir los componentes de la arquitectura de redes de sensores. La cual consta del nodo sensor, el nodo sumidero y la mini estación base a cargo de una SBC, esta definición es esencial porque ayudara a planificar tipos de despliegue para cubrir un área deseada.

Como hemos podido observar en la ilustración 11,12 y 13 se muestra la composición del nodo sensor, nodo sumidero y la SBC; y en la figura 24 los cuatro componentes de la plataforma, dos nodos sensor, un nodo sumidero conectado a la SBC.

NODO SENSOR

Un tipo de nodo que tiene como función captar y enviar datos por el medio inalámbrico, contiene dos componentes, un integrado con sensores y un integrado con un microcontrolador, estos por su funcionalidad se diferencia en nodos finales y nodos intermedios, definidos en el marco teórico.

Los nodos sensor capta variables físicas como, humedad, temperatura, movimiento y luminosidad, utiliza los dispositivos Iris más una placa sensores, la MTS 310CB (Ver ilustración 11).

Ilustración 11 – Modelo físico del nodo sensor



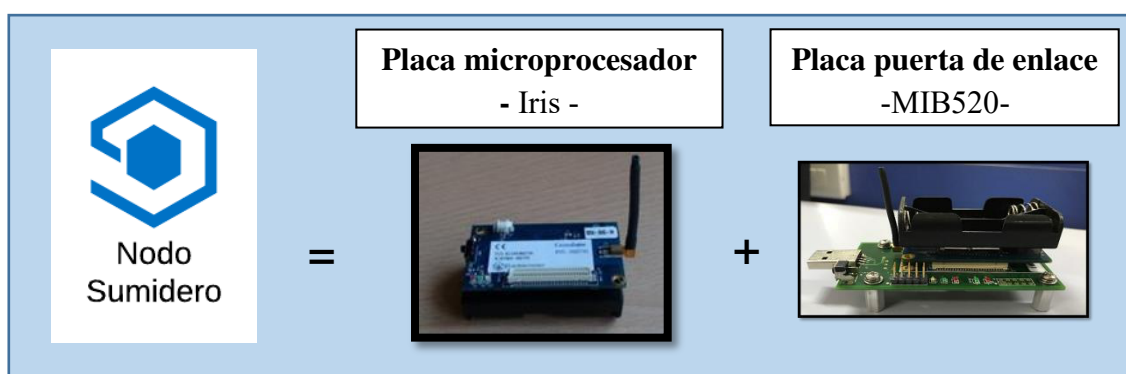
Fuente: Elaboración Propia

NODO SUMIDERO

El otro tipo de nodo denominado sumidero consta de un integrado microcontrolador y un integrado que hace de puerta de enlace con USB adaptado para la conexión a la SBC o mini estación. Este nodo sumidero es el responsable de la captación total de los datos dentro de la red, para luego entregar estos datos a la SBC de manera “legible”, el cual puede contener un sistema de gestión de información.

Para el nodo sumidero se ha utilizado la placa Iris más la placa MIB520, el cual recolecta los datos de la red, porque es un nodo netamente para la recepción de datos (Ver ilustración 12).

Ilustración 12 – Modelo físico del nodo sumidero

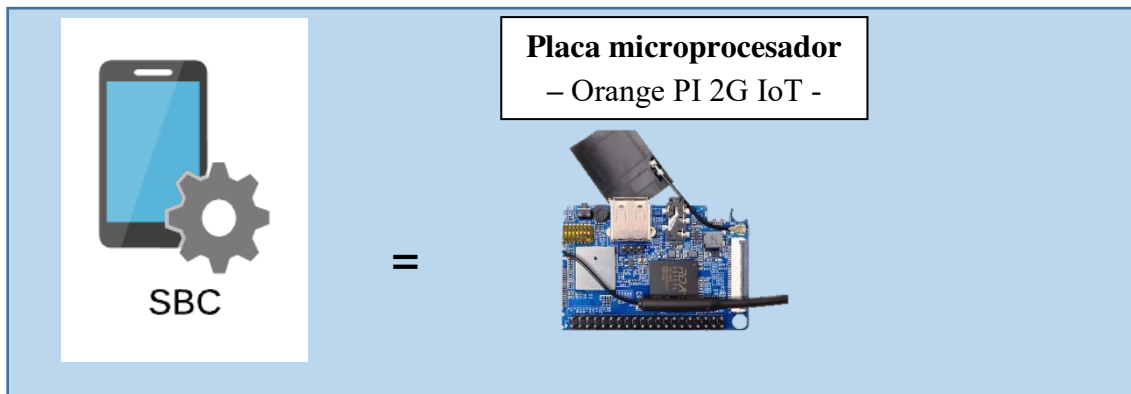


Fuente: Elaboración Propia

MINI ESTACION - SBC

La SBC será quien decida si procesa los datos in situ o deriva los datos a través de una red, porque es un componente netamente de procesamiento (ver ilustración 13). Este a diferencia del resto de dispositivos, es un componente con más performance e inteligente, la figura 24 se muestra dos nodos sensor y el nodo sumidero conectado a la SBC.

Ilustración 13 – Modelo físico del SBC

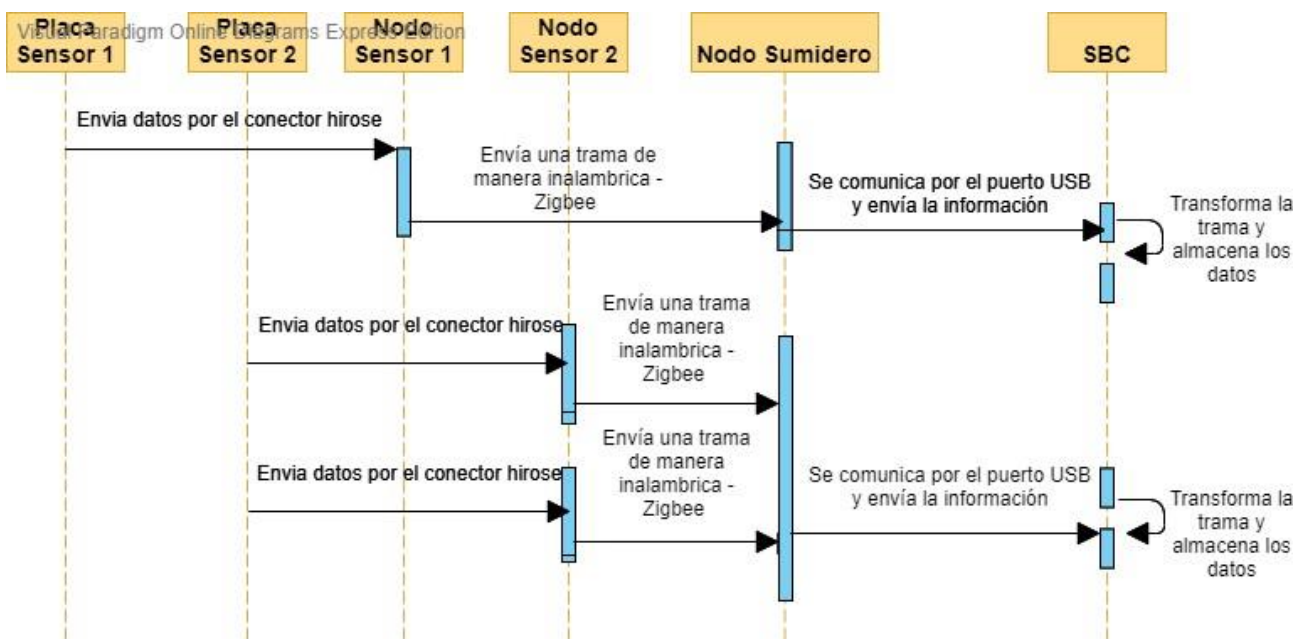


Fuente: Elaboración Propia

5.1.3 DESCRIPCIÓN FUNCIONAL DE LOS COMPONENTES

En la siguiente Ilustración 14, mostramos la secuencia del envío de datos entre los componentes definidos anteriormente, de manera didáctica en esta ilustración se separa la placa sensor del Nodo Sensor para una mejor comprensión de la secuencia en el envío de datos, pero según se definió en el modelo físico el Nodo Sensor está compuesto por la placa sensor y la placa iris.

Ilustración 14 Diagrama de Secuencia de los componentes



Fuente: Elaboración Propia

En la siguiente Tabla 17, se muestra la explicación del flujo de funcionalidad de los componentes de nuestra red de sensores.

Tabla 17 Descripción de las funcionalidades de los componentes

Dispositivo	Descripción
Sensor 1	La placa sensor captan los datos de temperatura o humedad y los envía a través del conector hirose al Nodo Sensor 1, esta captación se realizará cada cierto tiempo según la programación.
Sensor 2	Realiza a la vez el envío de datos al Nodo Sensor 2, haciendo la misma acción que el sensor 1 en un lapso de tiempo desde el inicio de la toma.
Nodo Sensor 1	Transmite por medio del protocolo Zigbee de manera inalámbrica la información de temperatura y humedad del sensor hacia el nodo sumidero.
Nodo Sensor 2	Realiza a la vez la transmisión inalámbrica, igual que el Nodo Sensor 1, enviando la trama al nodo sumidero.
Nodo Sumidero	Recepción de la data del Nodo Sensor 1 y Nodo Sensor 2 y re direcciona la información por el puerto USB hacia la mini estación Base Orange PI 2 IoT.
SBC	Recibe las tramas de información de ambos Nodos Sensor a través del Nodo sumidero para decidir después que hacer con la data, por ejemplo, almacenar los datos de manera local o enviarlo a un servidor remoto.

Fuente: Elaboración Propia

Figura 18 – Los cuatro componentes de la plataforma



Fuente: Elaboración Propia

5.2 CONFIGURACION DEL SISTEMA OPERATIVO GNU-LINUX EN EL SBC

Lo primero que se hará será tener listo los programas para el manejo de SD Card y de la interfaz TTL en la instalación y configuración de los componentes externos.

5.2.1 FORMATEAR EL SD CARD

El primer paso comienza con el formateo de la memoria flash externa que en esta incluirá la imagen GNU-Linux en la SBC, por tanto, luego decidir cual programa a utilizar para el formateo de la memoria flash procedemos descargar e instalarlo.

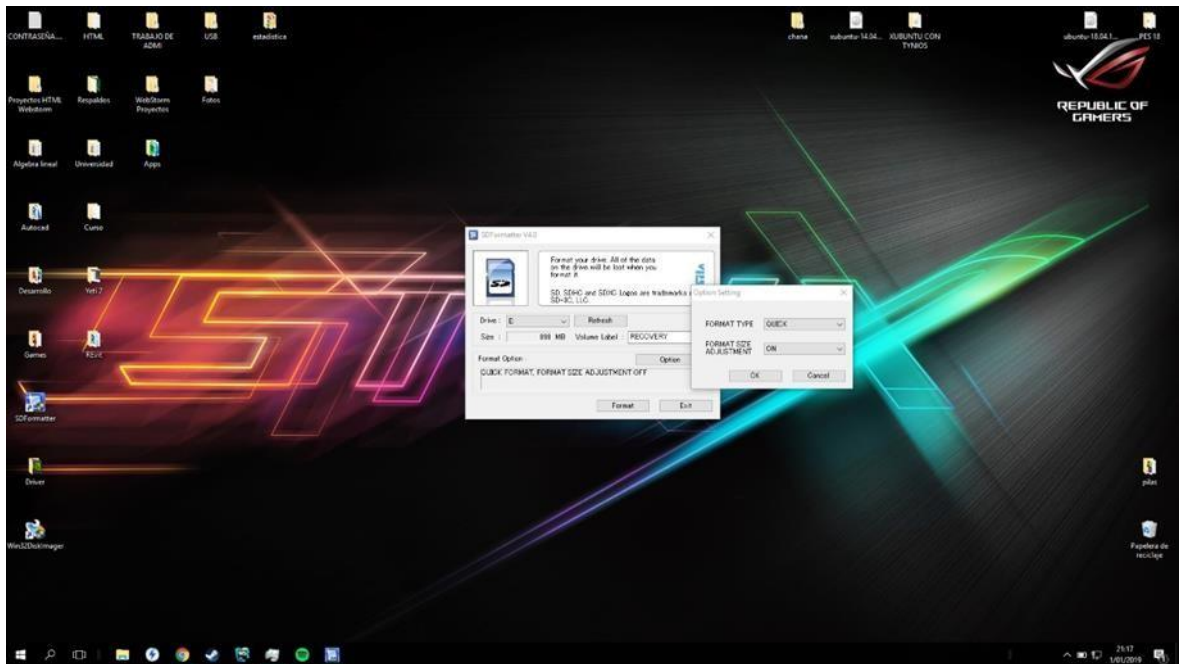
- a) Entramos al enlace de descarga <https://sd-card-formatter.uptodown.com/windows/descargar> y damos clic en la opción para descargar.
- b) Nos quedará un archivo .rar que descomprimiremos y nos quedará un ejecutable.
- c) Abrimos el ejecutable y comenzaremos el proceso de instalación.
- d) Seleccionamos un lugar en el equipo para guardar el programa.
- e) Ahora damos clic en "Install" para que el programa comience a instalarse.
- f) Esperamos unos segundos y listo, damos clic en "Finish" y el programa estará instalado.

Ahora procedemos a ingresar la microSD en la tarjeta SD para posteriormente ingresar a la computadora y que se realice el formateo respectivo.

- a) Ahora conectaremos a nuestra computadora nuestra tarjeta SD y abriremos nuestro SD Formatter,
- b) Una vez hecho el paso anterior, damos clic en "Option", en tipo de formato lo dejamos en "QUICK" y en ajuste de tamaño del formato lo dejamos en "ON".

- c) Damos clic en “OK” y luego en “Format” para empezar la microSD; por último, damos clic en “Exit”, como indica en la figura 25.
- d) Con el paso anterior nuestro microSD ya está formateada para el ingreso de la imagen.

Figura 19 Formatear microSD



Fuente: Elaboración Propia

5.2.2 GRABAR GNU-LINUX DEBIAN AL SD CARD.

Después del formateo de la memoria microSD ahora incluiremos la imagen. Se decidió para la grabación de la imagen el programa Win32 Disk Manager, después de analizar innumerables programas que tienen este fin:

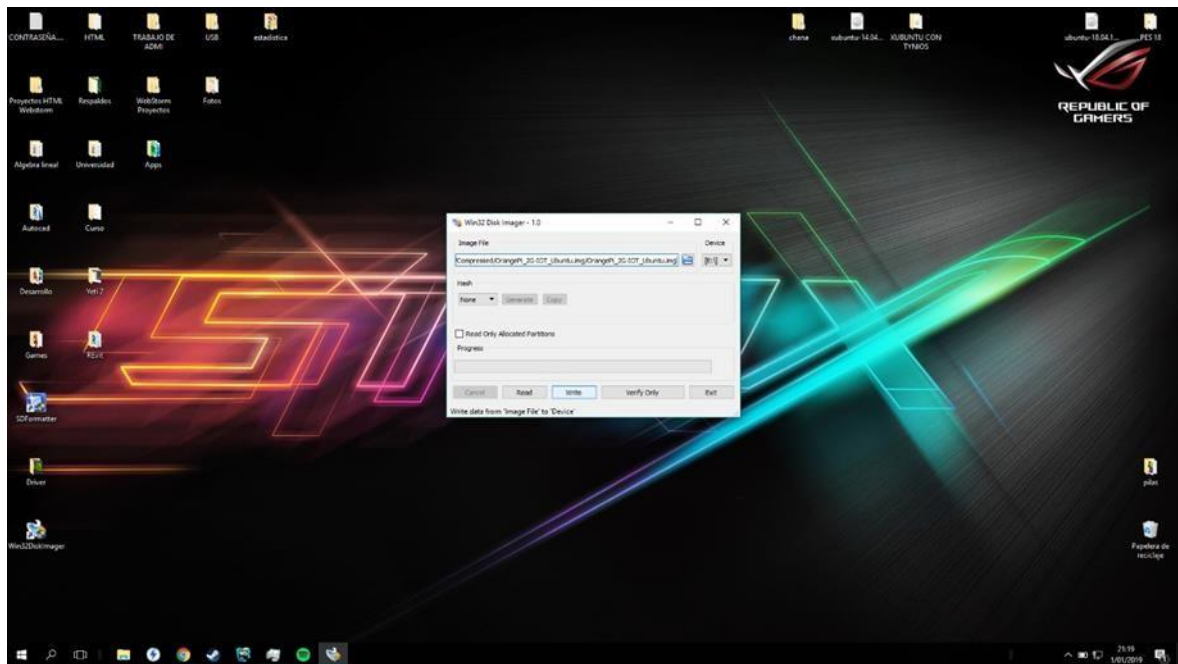
- a) Nos dirigimos al enlace <http://sourceforge.net/projects/win32diskimager/files/Archive/> para descargar el programa que nos permitirá grabar una imagen en una microSD.
- b) Una vez descargado, nos quedará un ejecutable, lo abrimos para comenzar el proceso de instalación.
- c) Ahora damos clic en "Install" para empezar la instalación.
- d) Damos clic en "Finish" y el programa estará exitosamente instalado.

El sistema operativo, GNU-Linux que se eligió fue Debian cuyo soporte es oficial por parte de Orange PI, que se encuentra en su página oficial.

Recordamos que para instalar en el Orange PI, necesitaremos una imagen del tamaño de 4 Gb de mínimo.

- a) Abrimos Win32 Disk Imager y en "Device" buscamos la tarjeta SD ya formateada.
- b) Ahora buscamos la ubicación de nuestra imagen, la seleccionamos y damos en abrir.
- c) Damos clic en "Write" y esperamos a que la imagen copie como en la figura 26.
- d) Una vez hecho esto, la imagen quedará escrita correctamente.

Figura 20 Grabación de imagen Debian



Fuente: Elaboración Propia

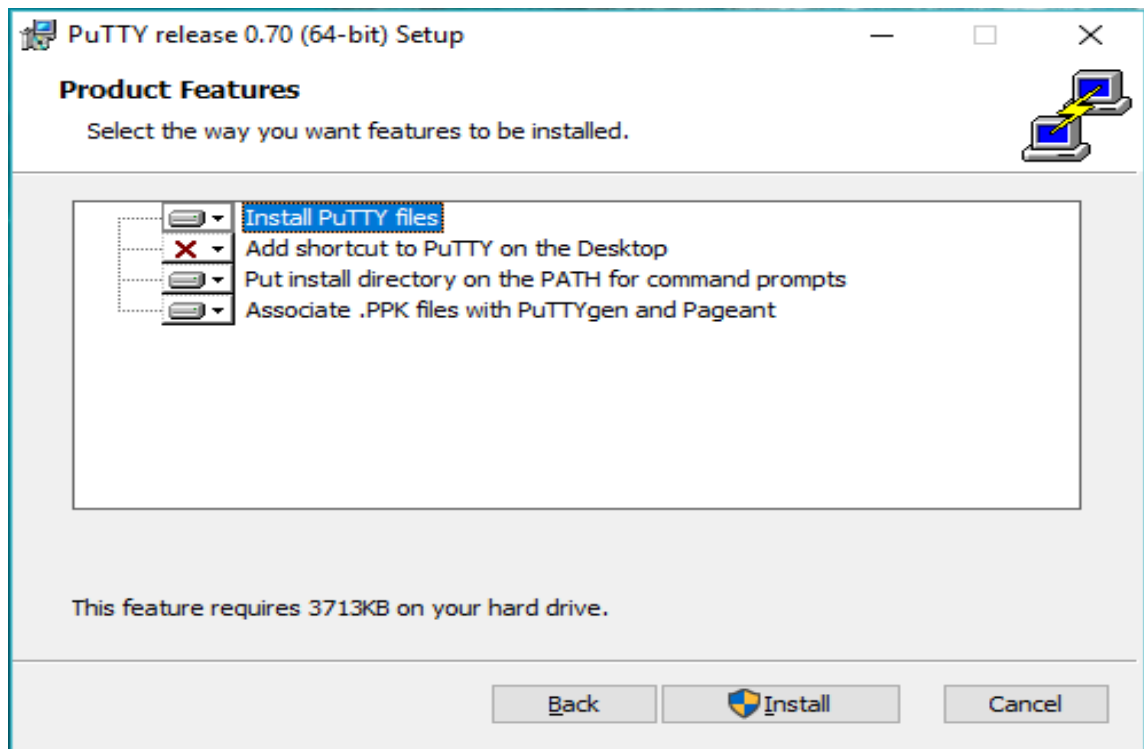
5.2.3 INSTALAR LA IMAGEN GNU-LINUX DEBIAN EN EL ORANGE PI

Aquí debemos utilizar una herramienta llamada PUTTY que nos permita utilizar la pantalla y el teclado de la computadora en el Orange PI, si hubiésemos tenido una pantalla LCD conectada al Orange PI directamente hubiésemos desechado este paso. Por tanto, primero necesitamos la instalación de la herramienta Putty y luego conectar el componente TTL que nos permite conectar la computadora con el Orange PI, por último, se realizara la instalación de Debian en el Orange PI.

Instalación Putty

- a) Para el transporte de la imagen hacia el Orange PI, descargaremos el cliente Putty en el enlace <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>, seleccionamos una de las opciones de descarga dependiendo del sistema operativo que tengamos.
- b) Lo abrimos para empezar el proceso de instalación.
- c) Seleccionamos una ubicación donde será instalado y damos clic en "Install" para comenzar a instalar como indica en la figura 21.
- d) Esperamos unos segundos y finalmente damos click en "Finish"; seguido estos pasos, PuTTY estará completamente instalado.
- e) Una vez terminada la instalación, empezamos a la conexión de la interfaz TTL con la computadora.

Figura 21 Instalación de Putty

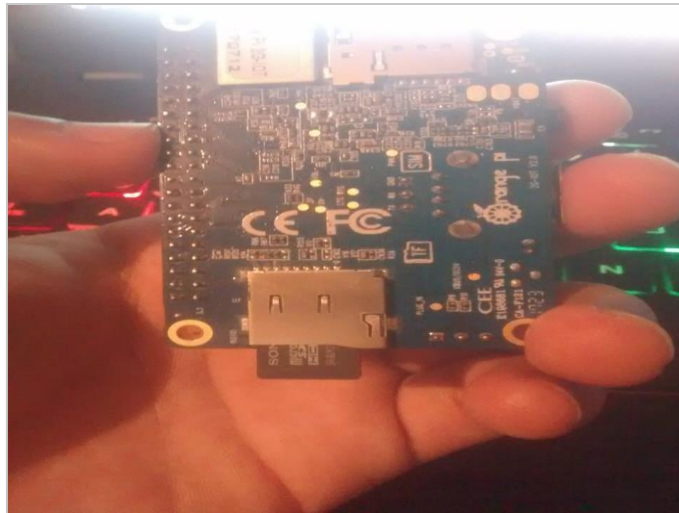


Fuente – Elaboración Propia

Conexión TTL – Orange Pi y PC

- a) Como primer paso insertar la tarjeta microSD con la imagen escrita en el borde lateral izquierdo de la cara inferior de la placa del Orange Pi como se muestra en la figura 22.

Figura 22 Ingreso de tarjeta microSD en el Orange Pi



Fuente – Elaboración Propia

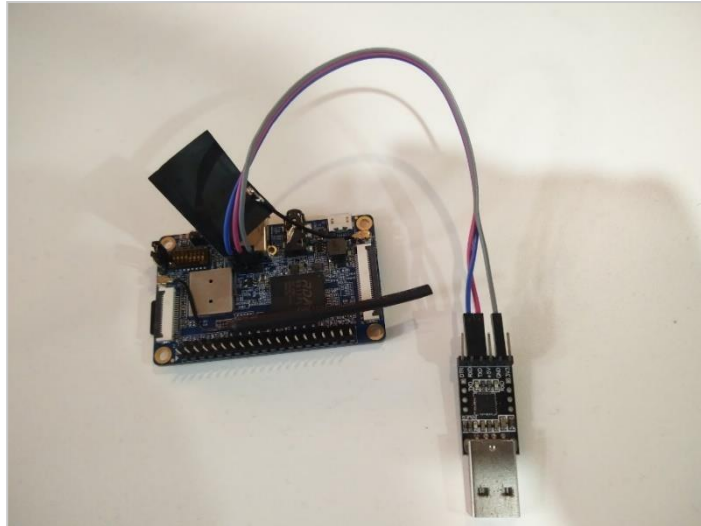
- b) Usaremos estos 3 pines aquí: pines tx, rx y gnd y las conexiones como indica en la tabla 17 y la figura 29. En nuestro puerto serie se tiene dos líneas, Tx y Rx. Estas líneas se deben cruzar para comunicar la SBC con la PC, es decir, el Tx del SBC debe conectar al Rx de la TTL. El Tx de la TTL debe conectar al Rx del SBC. Además, los dos dispositivos deben compartir una tierra o masa común.

Orange Pi 2G IOT	TTL Serial Converter
Gnd	gnd
Tx	rx
Rx	tx

Tabla 18 - Tabla de conexión TTL

- c) Ahora como en la figura 23 mostrada se puede ver el Orange Pi conectado al TTL según la forma como se indica.

Figura 23 Conexión TTL con el Orange Pi



Fuente - Elaboración Propia

- d) Conectamos el TTL por el puerto USB a la computadora, aquí tendrá que reconocer el puerto, si no es así se tendrá que hacer uso del driver o controlador (Ver figura 24).

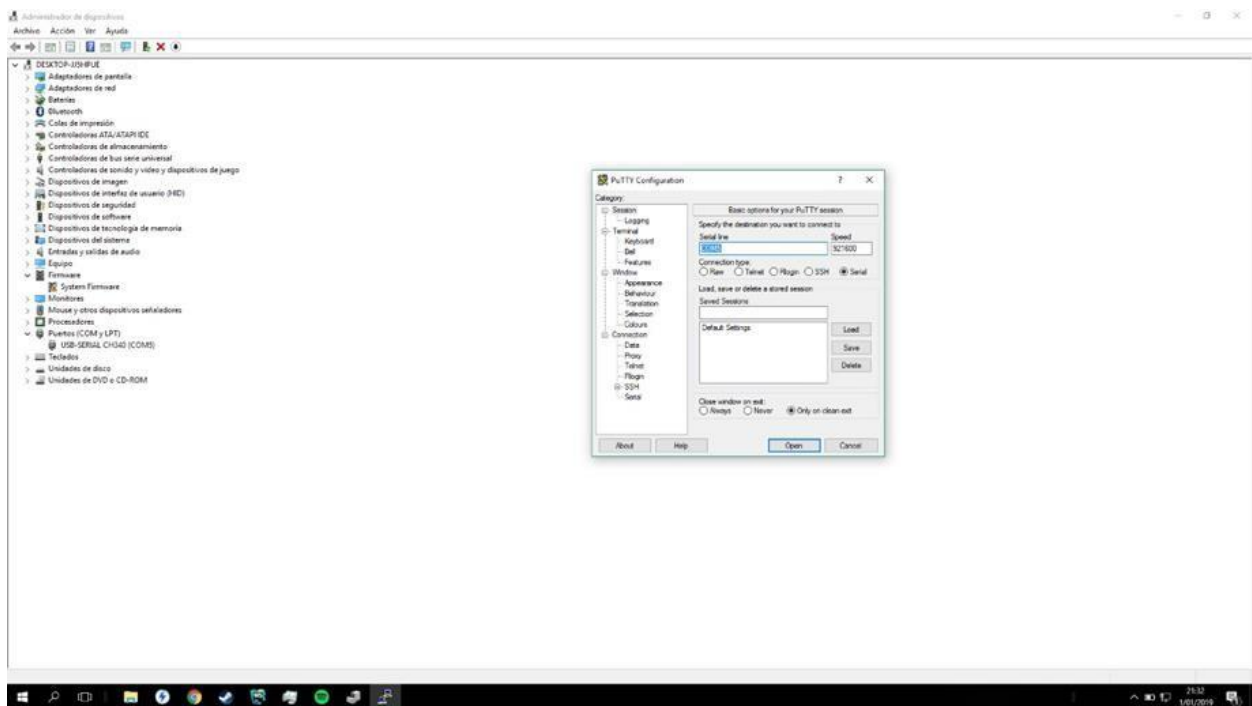
Figura 24 TTL conectado a la PC



Fuente: Elaboración Propia

- e) Buscamos en nuestra computadora el administrador de dispositivos y vamos a la opción de puertos; nuestro TTL fue reconocido como puerto COM5.
- f) Abrimos PuTTY, en línea serial cambiamos el valor a COM5 y la velocidad a 921600. Luego damos clic en “Open”, es importante mencionar que la recomendación de comunicación es de 115200, pero en la práctica hubo problemas, dado que se detiene la instalación después de iniciarla (Ver figura 25).

Figura 25 Realizamos la conexión de Putty con la PC



Fuente: Elaboración Propia

- g) Nos saldrá la terminal, pero no ejecuta nada ya que no hay fuente de alimentación.
- h) Una vez conectada la fuente de alimentación de 5 v, al Orange PI, el led rojo se iluminará y se empezará a ejecutar.

- i) En la instalación nos solicitará el usuario que vendrá a ser "root" y la contraseña "orangepi".
- j) Listo, la imagen Debian ha sido ejecutada cargada en el Orange Pi (Ver figura 26).

Figura 26 Instalación Debian realizada en el Orange Pi

```
Starting LSB: Set the CPU frequency scaling governor to "ondemand"...
[ OK ] Started OrangePi A33 GPIO.
[ OK ] Started Daily apt activities.
[ OK ] Reached target timers.
[ OK ] Started Load/Save RF Kill Switch Status.
[ OK ] Started Remote User Sessions.
[ OK ] Started Restore /etc/resolv.conf if...fore the ppp link was shut down.
[ OK ] Started Save/Restore Sound Card State.
[ OK ] Started LSB: Start Motion detection.
[ OK ] Started LSB: Set the CPU Frequency Scaling governor to "ondemand".
[ OK ] Started Login Service.
Starting Set console scheme...
[ OK ] Started Set console scheme.
[ OK ] Started Network Manager.
Starting Network Manager Wait Online...
Starting Network Manager Script Dispatcher Service...
[ OK ] Started Wake network interfaces.
[ OK ] Reached target Network.
[ OK ] Started Network Manager Script Dispatcher Service.
Starting Hostname Service...
[ OK ] Stopped LSB: Start NTP daemon.
[ OK ] Started Hostname Service.
[ 10.428182] cma-sb: 0: doesn't transmit frame, timeout(10000 ms)
Starting WPA supplicant...
Starting Authenticate and Authorize Users to Run Privileged Tasks...
[ OK ] Started WPA supplicant.
[ OK ] Started Network Manager Wait Online.
[ OK ] Started Authenticate and Authorize Users to Run Privileged Tasks.
Starting LSB: Start NTP daemon...
[ OK ] Reached target Network is Online.
Starting /etc/rc.local Compatibility...
[ 11.642743] rc.local[411]: numid=17,iface=mlxer,name=Loop Mode'
[ 11.661395] rc.local[411]: ; type=INTEGER,access=ro,value=1,min=0,max=65535,step=0
[ 11.683700] rc.local[411]: ; values=
[ 11.706180] rc.local[411]: smixer: Cannot find the given element from control default
[FAILED] Failed to start /etc/rc.local Compatibility.
See 'systemctl status rc.local.service' for details.
[ OK ] Started Serial Getty on ttyS0.
[ OK ] Started LSB: Start NTP daemon.

Ubuntu 16.04.1 LTS orangepi ttyS0
orangepi login: root
Password:
Last login: Thu Feb 11 16:28:19 UTC 2016 on ttyS0
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 3.10.62-rc15.0.2+ armv7l)

 * Documentation:  http://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

OrangePi 2G-10T
*****
Welcome to OrangePi 2G-10T!
No disks! please configure your OrangePi 2G-10T!
sudo orangepi-wifi
Have good trip on OrangePi 2G-10T!
*****
root@orangepi:~#
```

Fuente: Elaboración Propia

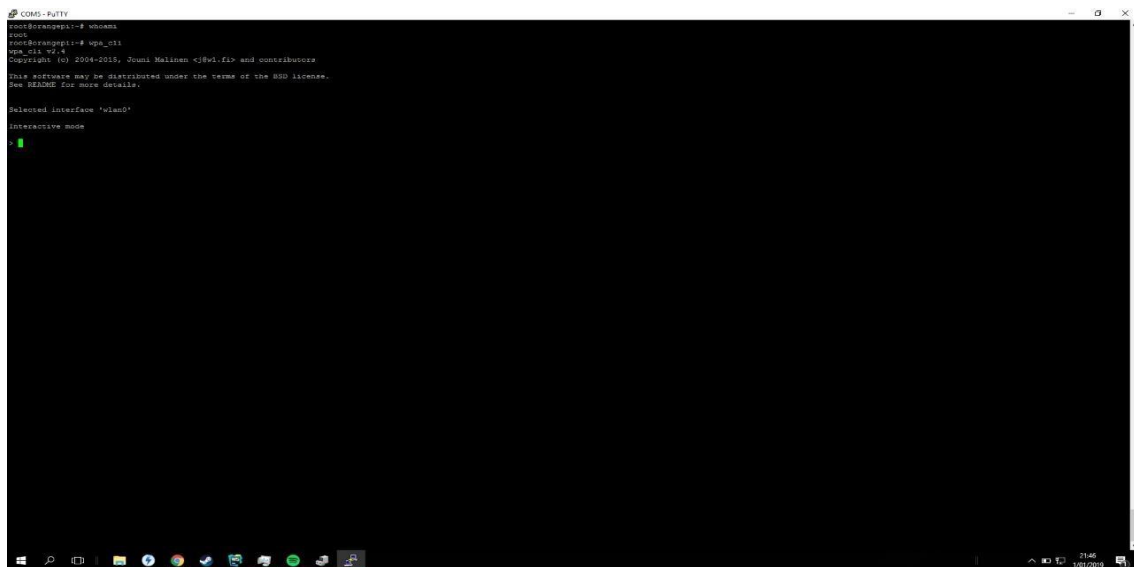
5.3 OPERATIVIDAD DE LA SBC PARA LA COMUNICACIÓN CON EL NODO SUMIDERO

5.3.1 CONFIGURACION WIFI EN ORANGE PI 2G-IOT

Una vez instalado el Debian seguimos conectado a Putty y pasamos a configurar el punto wifi en el Orange PI, para luego instalar los programas necesarios para la comunicación con el nodo sumidero (Ver figura 27).

- a) Introducimos un comando llamado *“whoami”* y luego el comando *wpa_cli* para crear y administrar redes wifi.

Figura 27 Solicitando datos para la instalación del wifi en el Orange Pi



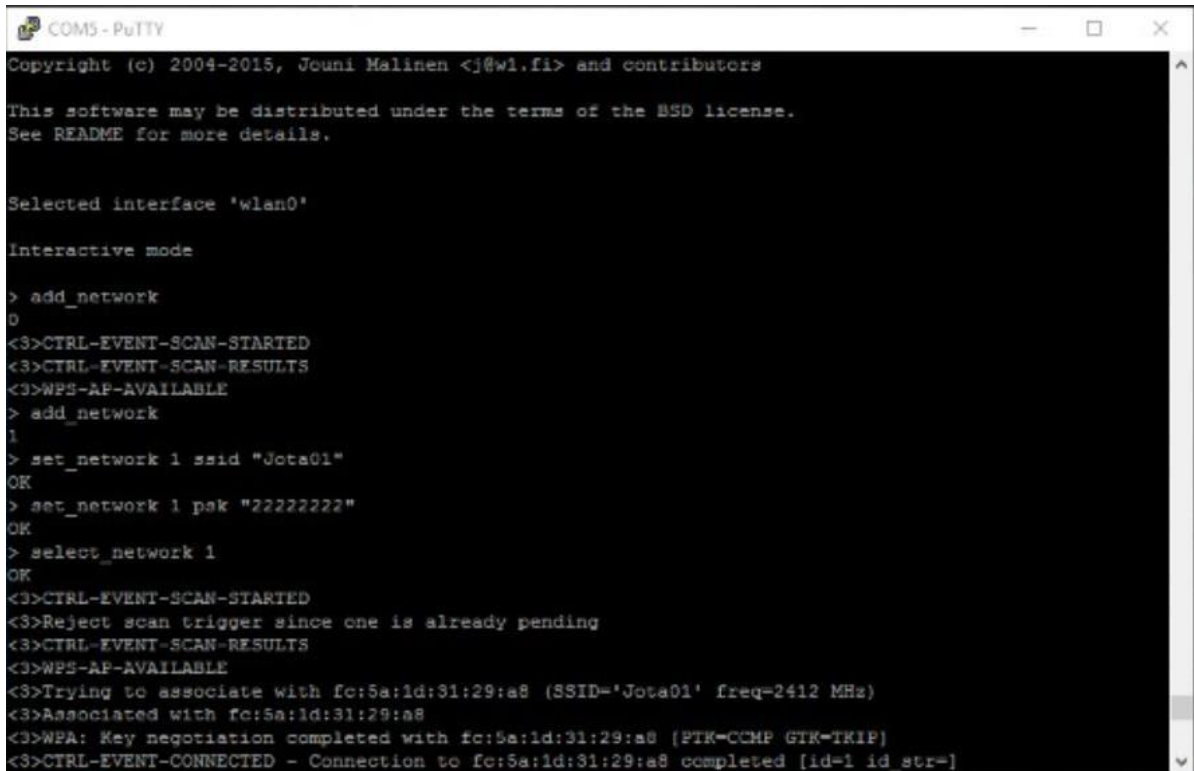
```
COMS-PUTTY
root@orangepi:~# whoami
root
root@orangepi:~# wpa_cli
wpa_cli v2.9
Copyright (c) 2004-2015, Jouni Malinen <j@w1.fi> and contributors
This software may be distributed under the terms of the BSD license.
See README for more details.
Selected interface 'wlan0'
Interactive mode
>
```

Fuente: Elaboración Propia

- b) Ahora agregamos una red con el comando *“add_network”* pero para que nos salga nuestro número de red correctamente, nuestra computadora debe estar conectado por wifi, (Ver figura 34).
- c) Una vez configurado esto, introducimos el comando *“set_network* (número de red) *ssid* (*“nombre de red”*), ahora ponemos

contraseña con el comando “*set_network* (número de red) psk (“contraseña”).

Figura 28 Instalando wifi en el Orange Pi



```
COMS - PuTTY
Copyright (c) 2004-2015, Jouni Malinen <j@w1.fi> and contributors
This software may be distributed under the terms of the BSD license.
See README for more details.

Selected interface 'wlan0'

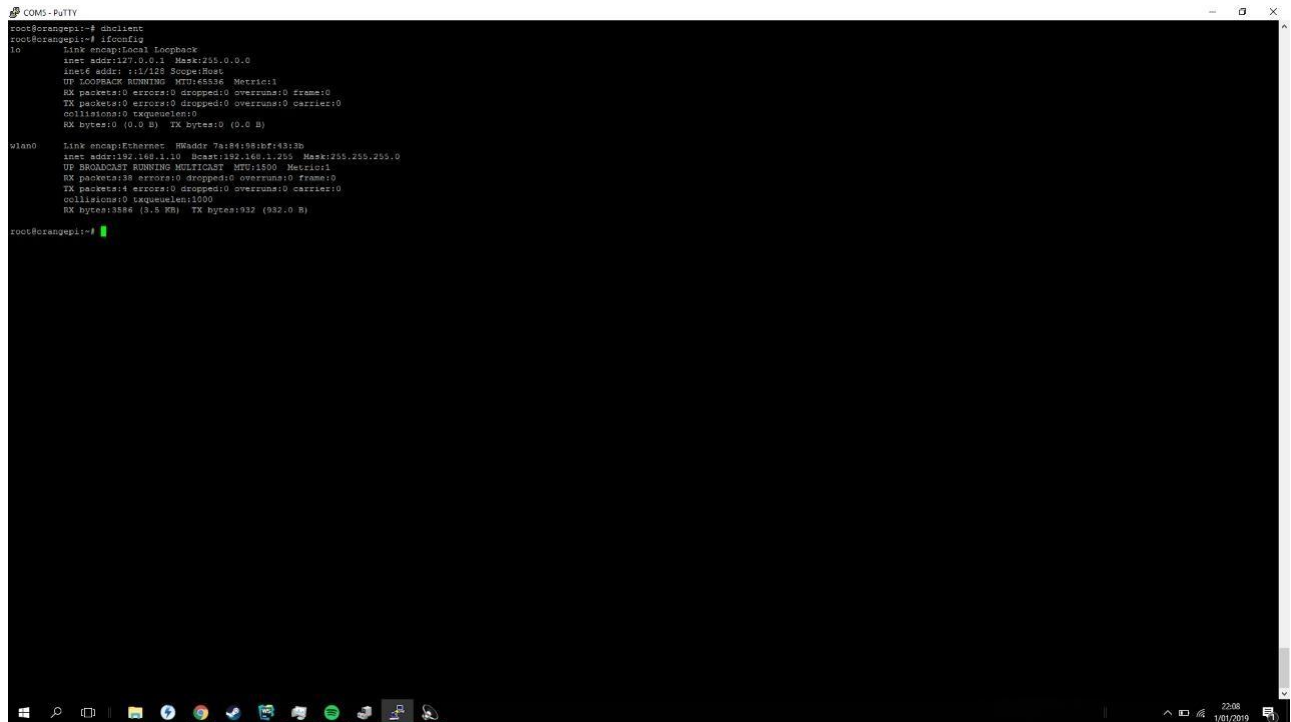
Interactive mode

> add_network
0
<3>CTRL-EVENT-SCAN-STARTED
<3>CTRL-EVENT-SCAN-RESULTS
<3>WPS-AP-AVAILABLE
> add_network
1
> set_network 1 ssid "Jota01"
OK
> set_network 1 psk "22222222"
OK
> select_network 1
OK
<3>CTRL-EVENT-SCAN-STARTED
<3>Reject scan trigger since one is already pending
<3>CTRL-EVENT-SCAN-RESULTS
<3>WPS-AP-AVAILABLE
<3>Trying to associate with fc:5a:1d:31:29:a8 (SSID='Jota01' freq=2412 MHz)
<3>Associated with fc:5a:1d:31:29:a8
<3>WPA: Key negotiation completed with fc:5a:1d:31:29:a8 [PTK-CCMP GTK-TRIP]
<3>CTRL-EVENT-CONNECTED - Connection to fc:5a:1d:31:29:a8 completed [id=1 id_str=]
```

Fuente: Elaboración Propia

- d) Por último, seleccionamos nuestra conexión con el comando “*select_network* (número de red) y con esto estaremos conectados a nuestra red.
- e) Para obtener la dirección IP, introducimos el comando “*dhclient*” y luego el comando “*ifconfig*”, la IP que se muestra en la figura 30 demuestra que estamos conectados a partir de nuestra IP.

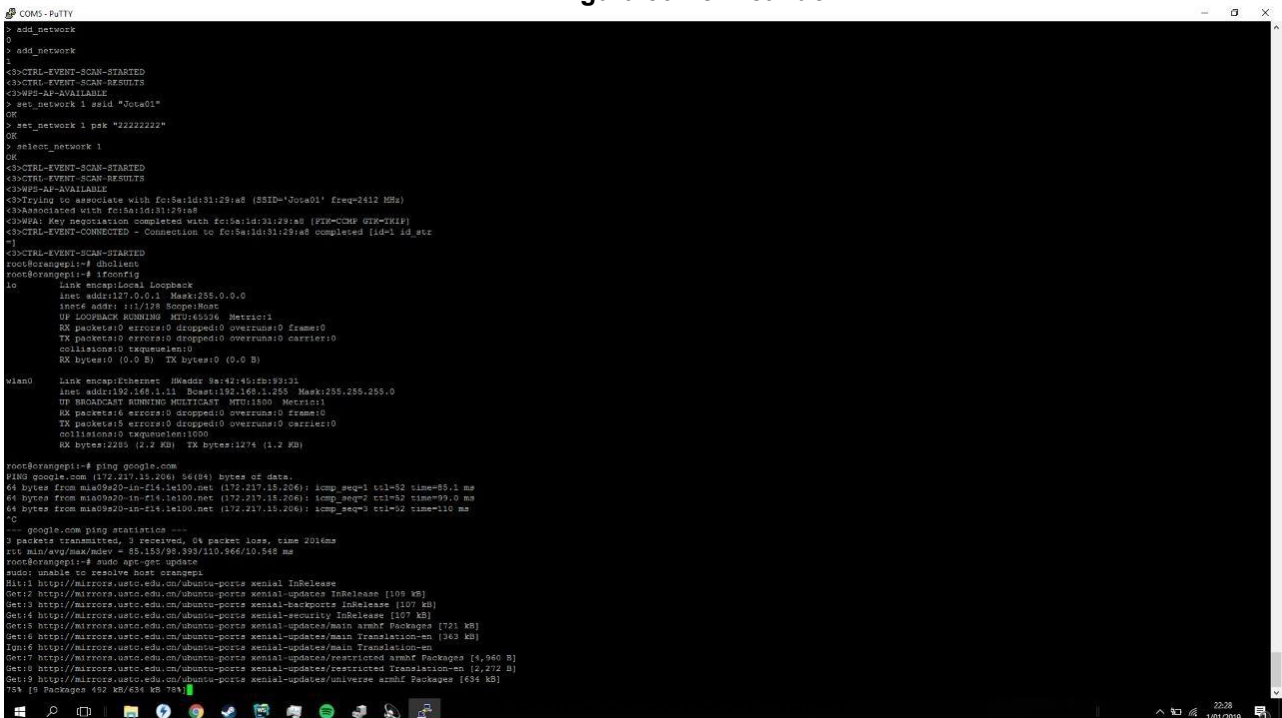
Figura 29 wifi instalado en el Orange Pi



Fuente: Elaboración Propia

- f) Ahora demostraremos que tenemos internet, realizando el comando "ping".
- g) Por último, haremos un update para guardar todo con el comando "sudo apt-get update".

Figura 30 Verificando wifi



5.3.2 INSTALACION JAVA EN ORANGE PI 2G-IOT

La necesidad de instalar Java es por el motivo de que el sistema operativo TinyOS tiene clases java que nos sirven, como TOSBase, Oscilloscope , SerialForwarder la cual utilizaremos para la recepción de datos de los nodos sensores a través del nodo sumidero.

- Actualizamos librerías con el comando ***sudo apt-get install software-properties-common*** y creamos repositorios con el comando ***sudo add-apt-repository ppa:webupd8team/java*** (Ver figura 31).
- Finalmente usamos los comandos ***java -version*** y ***javac -version*** y comprobamos que está instalado y que tenemos la versión 1.8.

Figura 31 Actualizando librerías para instalación de Java

```
COMS-FUTTY
update-alternatives: using /usr/lib/jvm/java-8-oracle/bin/jinfo to provide /usr/bin/jinfo (jinfo) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-oracle/bin/jmap to provide /usr/bin/jmap (jmap) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-oracle/bin/jps to provide /usr/bin/jps (jps) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-oracle/bin/jrunscript to provide /usr/bin/jrunscript (jrunscript) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-oracle/bin/jstack to provide /usr/bin/jstack (jstack) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-oracle/bin/jstat to provide /usr/bin/jstat (jstat) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-oracle/bin/jstatd to provide /usr/bin/jstatd (jstatd) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-oracle/bin/native2ascii to provide /usr/bin/native2ascii (native2ascii) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-oracle/bin/trace to provide /usr/bin/trace (trace) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-oracle/bin/schemagen to provide /usr/bin/schemagen (schemagen) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-oracle/bin/serialver to provide /usr/bin/serialver (serialver) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-oracle/bin/verify to provide /usr/bin/verify (verify) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-oracle/bin/wasm to provide /usr/bin/wasm (wasm) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-oracle/bin/wasmimport to provide /usr/bin/wasmimport (wasmimport) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-oracle/bin/wxc to provide /usr/bin/wxc (wxc) in auto mode

#####Important#####
To set Oracle JRE as default, install the "oracle-java8-set-default" package.
E.G.: sudo apt install oracle-java8-set-default
On Ubuntu systems, oracle-java8-set-default is most probably installed
automatically with this package.
#####

Selecting previously unselected package oracle-java8-set-default.
(Reading database ... 41629 files and directories currently installed.)
Preparing to unpack .../oracle-java8-set-default_8u101-1-webupd8-1_all.deb ...
Unpacking oracle-java8-set-default (8u101-1-webupd8-1) ...
Selecting previously unselected package gfonts.
Preparing to unpack .../gfonts_1884d11ac70c7f1.0.7+prettt-4.1ubuntu1_all.deb ...
Unpacking gfonts (1884d11ac70c7f1.0.7+prettt-4.1ubuntu1) ...
Selecting previously unselected package libfontconfig1:armhf.
Preparing to unpack .../libfontconfig1_1:2.13-1-armhf.deb ...
Unpacking libfontconfig1:armhf (1:2.13-1) ...
Selecting previously unselected package libfontconfig1:armhf.
Preparing to unpack .../libfontconfig1_1:2.13-1-1ubuntu0.16.04.4_armhf.deb ...
Unpacking libfontconfig1:armhf (1:2.13-1-1ubuntu0.16.04.4) ...
Selecting previously unselected package xfntfonts.
Preparing to unpack .../xfntfonts_1:1.0.4-2_all.deb ...
Unpacking xfntfonts (1:1.0.4-2) ...
Selecting previously unselected package xfntfonts-x11.
Preparing to unpack .../xfntfonts-x11_1:1.0.4-2-armhf.deb ...
Unpacking xfntfonts-x11 (1:1.0.4-2) ...
Selecting previously unselected package gfonts-x11.
Preparing to unpack .../gfonts-x11_0.24_all.deb ...
Unpacking gfonts-x11 (0.24) ...
Processing triggers for fontconfig (2.11.3-4ubuntu1) ...
Processing triggers for libx-tls (1.0.2-7ubuntu1) ...
Processing triggers for man-db (2.7.6-1) ...
Setting up oracle-java8-set-default (8u101-1-webupd8-1) ...
Setting up gfonts (1884d11ac70c7f1.0.7+prettt-4.1ubuntu1) ...
Setting up libfontconfig1:armhf (1:2.13-1) ...
Setting up libfontconfig1:armhf (1:2.13-1-1ubuntu0.16.04.4) ...
Setting up xfntfonts (1:1.0.4-2) ...
Setting up xfntfonts-x11 (1:1.0.4-2) ...
Setting up gfonts-x11 (0.24) ...
Processing triggers for libx-tls (1.0.2-7ubuntu1) ...
root@OrangePi:~# java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b12)
Java HotSpot(TM) Client VM (build 25.101-b12, mixed mode)
root@OrangePi:~# javac -version
javac 1.8.0_101
```

Fuente: Elaboración Propia

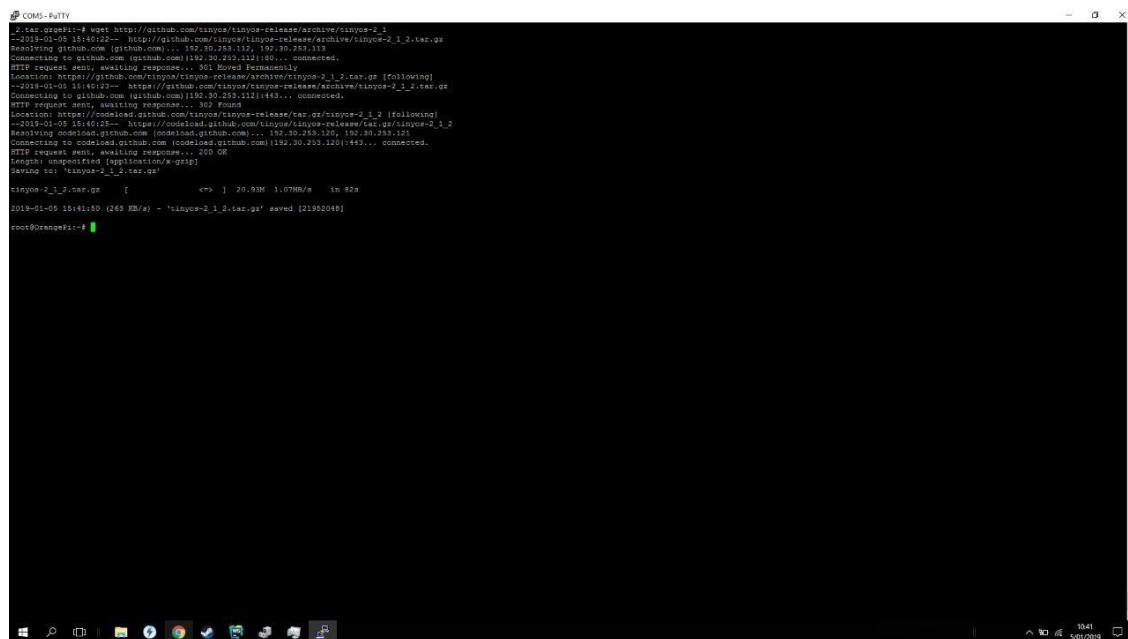
- c) En la instalación nos solicitará el usuario que vendrá a ser “root” y la contraseña “orangepi”.

5.3.3 INSTALACIÓN DE TINYOS EN ORANGE PI 2G-IOT

Una vez que se han establecido la instalación de java es indispensable instalar el sistema operativo TinyOS para realizar una comunicación entre los nodos sensor y la SBC a través de una puerta de enlace que hará de interfaz entre el nodo sumidero y el Orange PI, cabe recordar que los bits en la cual trabaja la arquitectura del microprocesador ARM del Orange PI debe ser igual a los bits que trabajara los sistemas operativos y programas que se instalaran en este.

- a) Usaremos la instalación automática; introducimos el comando “wget http://github.com/tinyos/tinyos-release/archive/tinyos-2_1_2.tar.gz”, (ver figura 32).

Figura 32 Iniciando instalación de TinyOS



```
root@orangepi:~# wget http://github.com/tinyos/tinyos-release/archive/tinyos-2_1_2.tar.gz
--2018-01-05 15:46:22-- http://github.com/tinyos/tinyos-release/archive/tinyos-2_1_2.tar.gz
Resolving github.com [github.com]:: 192.30.253.122, 192.30.253.133
Connecting to github.com [github.com]:192.30.253.122:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://github.com/tinyos/tinyos-release/archive/tinyos-2_1_2.tar.gz [following]
--2018-01-05 15:46:23-- https://github.com/tinyos/tinyos-release/archive/tinyos-2_1_2.tar.gz
Connecting to github.com [github.com]:192.30.253.122:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/tinyos/tinyos-release/tar.gz/tinyos-2_1_2 [following]
--2018-01-05 15:46:25-- https://codeload.github.com/tinyos/tinyos-release/tar.gz/tinyos-2_1_2
Resolving codeload.github.com [codeload.github.com]:: 192.30.253.120, 192.30.253.121
Connecting to codeload.github.com [codeload.github.com]:192.30.253.120:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 109825168 [application/gzip]
Saving to: 'tinyos-2_1_2.tar.gz'

tinyos-2_1_2.tar.gz  [  ]  <=> | 20.90M 1.07MB/s  in 82s
2018-01-05 15:46:10 (263 KB/s) = 'tinyos-2_1_2.tar.gz' saved [21952048]

root@orangepi:~#
```

Fuente: Elaboración Propia

- b) Esta es la ruta, como se puede ver abajo, aparecen las apps de tinyos.

- c) Ahora creamos un archivo “tinyos.env” y agregamos los siguientes comandos:

```
-export TOSROOT=/opt/tinyos-2.1.2
-export TODSDIR=$TOSROOT/tos
-export CLASSPATH=$TOSROOT/support/adk/java/tinyos.jar:.$CLASSPATH
-export MAKERULES=$TOSROOT/support/make/Makerules
-export PATH=/opt/msp430/bin:$PATH
-source /opt/tinyos-2.1.2/tinyos.ah
```

- d) Finalmente, el Orange pi estará configurado con tinyos 2.1.2.
e) Hacemos un update con el comando “sudo add-get update”.

5.4 CONFIGURACION Y OPERATIVILIDAD EN LOS NODOS DE LA RED

Para poner en funcionamiento toda la red de sensores, se debe programar y configurar todos los componentes de la arquitectura, para cada nodo hay un programa según el objetivo a seguir, a continuación, mostramos la forma de instalar estos programas en los dispositivos. Recordar que, para instalar los programas, el nodo sumidero permanecerá conectado al puerto USB de la SBC para poder hacer uso de los programas que se encuentran en el TinyOS.

Si bien es cierto la tesis se enfoca en la incorporación de una SBC como mini estación de trabajo en las redes de sensores, pero para ejecutar el funcionamiento de la plataforma utilizamos dos programas ya incorporados en TinyOS, no se va a describir las características del lenguaje nesC ni su sintaxis porque tendríamos que describir en todo un capítulo, pero sí comentaremos algunos alcances, así como las funcionalidades del código a utilizar para poder entender a grandes rasgos los programas.

TinyOS es un lenguaje desarrollado en nesC y este es un tipo derivado del lenguaje C, lenguaje muy utilizado en el desarrollo de programas de bajo nivel.

El lenguaje nesC toma como paradigma la programación orientada a componentes. Hay dos tipos de componentes, los módulos y las configuraciones. En los módulos se desarrolla las funcionalidades que uno requiere y en las configuraciones se menciona los componentes a utilizar, así como las interfaces que es el único medio que nos permite la comunicación entre los componentes llamados también “wiring”. En los módulos es donde se desarrolla las interfaces, que en si es el código de la aplicación.

Para determinar la relación entre los componentes e interfaces utilizamos el operador (->) , el componente que utiliza la interfaz se encuentra al lado izquierdo y la que proporciona al lado derecho ejem. A.i -> B.i , (A y B son componentes e “i” es la interfaz) .

Para diferenciar cual es modulo y cual la configuración, se suele escribir en el nombre del archivo la letra M para diferenciar.

Modulo:	TOSBaseM.nc , OscilloscopeM.nc
Configuración:	TOSBase.nc , Oscilloscope.nc

5.4.1 COMPILACION DEL SOFTWARE EN EL NODO SUMIDERO

El nodo sumidero es quien recibe los datos inalámbricamente por señales electromagnéticas de los nodos sensor, este nodo envía las tramas por el puerto serie a la SBC, para instalar este programa llamado TOSBase conectamos el USB del nodo sumidero a través de la puerta de enlace a la PC, conectamos un dispositivo iris y ejecutamos el comando siguiente en el prompt de Debian:

```
$ cd /opt/tinyos-1.x/apps/TOSBase  
$ make telosb install
```

5.4.2 COMPILACION DEL SOFTWARE EN EL NODO SENSOR

Los nodos sensores tienen la función de monitorizar los datos ambientales como la temperatura, la humedad, luminosidad, etc. Estos parámetros medidos de los nodos sensores se enviarán al nodo sumidero. Para programar cada nodo sensor se conecta al puerto USB del PC la puerta de enlace y conectamos un dispositivo iris en donde se ejecutará el siguiente comando siguiente en el prompt de Debian:

```
$ cd /opt/tinyos-1.x/contrib/moteiv/apps/Oscilloscope  
$ make telosb  
$ make telosb reinstall, <node_ID>
```

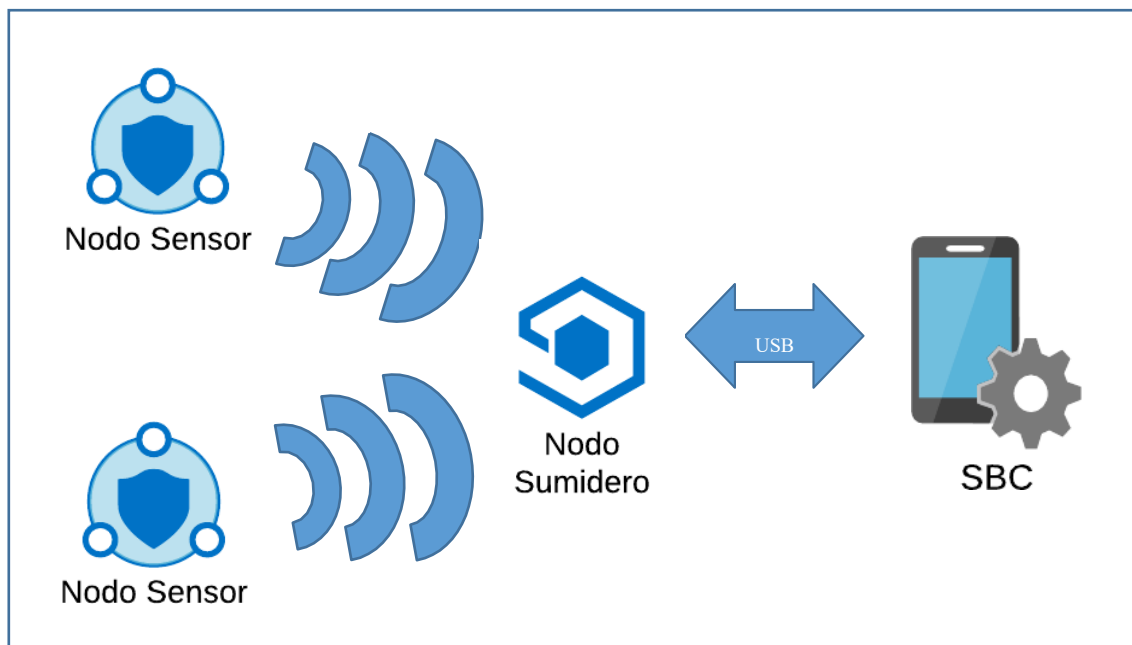
En <node_ID> es un variable que indica el nodo sensor dentro de la red.

5.5 INTEGRACIÓN DE LOS COMPONENTES

Una vez realizado el modelo lógico y modelo físico de la arquitectura, se deberá probar integrando los cuatro componentes (un nodo sumidero, dos nodos sensores y un SBC). Dado que el nodo sumidero necesita ser alimentado con más energía y que será quien traslade la información adquirida, permanecerá conectado al puerto USB de la SBC, para realizar esta prueba resulta adecuado elegir una topología en estrella para el despliegue

En esta topología estrella se despliega un solo elemento coordinador, en nuestro caso un solo nodo sumidero, el cual es el responsable como el nombre lo dice de coordinar e inicializar los nodos sensor, estos dispositivos adquieren un comportamiento esclavo. Cada trama o mensaje debe dirigirse al elemento coordinador el cual transmite los datos adquiridos al correspondiente dispositivo según la programación. En la Ilustración 15 se muestra una red en topología estrella de nuestro prototipo.

Ilustración 15 – Topología de la red de sensores



Fuente: Elaboración Propia

5.5.1 APLICACIONES EN EL NODO SUMIDERO Y NODO SENSOR

Una vez que ha sido instalado y configurado el TinyOS en la SBC, pasaremos a instalar los programas que nos brinda TinyOs en sus librerías para el funcionamiento de nuestra red de sensores. Los programas darán un funcionamiento a los nodos sensor y al nodo sumidero, la instalación de estos se dará en la placa Iris que es la base de los dos nodos descritos anteriormente.

El código a instalar dependerá de las características electrónicas de los nodos, para el nodo sumidero se instalará el programa TOSBase quien recopilará datos de los nodos sensor y para el nodo sensor el programa Oscilloscope quien captará los datos ambientales dependiendo de los tipos de sensor que lleva el nodo.

5.5.1.1 TOSBase

Este programa que se instala en el nodo sumidero tiene el objetivo de transferir los datos que llegan del nodo sensor inalámbricamente hacia el puerto serie conectado a la SBC de manera sincronizada. las tramas que llegan siempre contendrán un ID para diferenciarse entre ellas.

El programa para el nodo sumidero se encuentra en la carpeta:
`/opt/tinyos1.x/apps/TOSBase.`

5.5.1.2 Oscilloscope

Este programa que se instala en el nodo sensor capta información de los sensores de manera periódica, recordemos que el nodo Iris es donde se instalara este programa, que luego de captar la información enviara inalámbricamente al nodo sumidero que tiene instalada el TOSBase.

El programa para el nodo sensor se encuentra en la carpeta:
`/opt/tinyos1.x/contrib/moteiv/apps/Oscilloscope/.`

5.5.2 PROGRAMAS PARA CAPTAR INFORMACION

5.5.2.1 SerialForwarder

Nuestra mini estación base tiene que tener la capacidad de recepcionar los datos, aquí radica la importancia de que soporte java la SBC porque existe una aplicación desarrollada en java llamada SerialForwarder que actúa como un middleware, esta tiene la capacidad de recoger esos datos que ingresan por el puerto serie gracias a la aplicación TOSBase, estos datos que son recogidos se reenvía por un socket TCP, este programa permite conectar múltiples aplicaciones que permitan leer y escribir las tramas recibidas.

5.5.2.2 Aplicación Gráfica oscilloscope

En este punto tenemos ya los programas instalados en los nodos sensor y el nodo sumidero estarían ejecutándose, ahora necesitamos un programa que pueda captar los datos ingresados por el puerto serie a través del puerto TCP, nuevamente comentamos la importancia de java instalado en Debian de la SBC. dado que existe un programa en java llamada oscilloscope que tiene relación directa con el SerialForwarder para mostrar gráficamente en la SBC las mediciones captadas por los nodos sensores.

El oscilloscope mostrara los datos captados y transformado en mediciones en bruto por el conversor ADC de los nodos sensor, para la prueba de nuestra tesis solo necesitaremos verificar que nuestra arquitectura tenga la capacidad de captar datos de los nodos sensores y ya en un futuro podamos transformar esos datos para una aplicación determinada.

5.5.3 INICIAR EL MONITOREO

Para mostrar gráficamente los datos ya se cometo que se debe ejecutar todos los programas instalados en los nodos e iniciar en la SBC el programa SerialForwarder y por último el programa oscilloscope que mostrara una ventana swing con los datos en bruto recibidos.

5.5.3.1 Iniciar SerialForwarder

Se tiene que conectar el nodo sumidero específicamente la puerta de enlace al puerto USB de la SBC, luego abrir Debian y en su prompt escribir 'motelist', a continuación, esta instrucción hallara el puerto COM conectado al nodo sumidero.

Identificado el puerto COM, se inicia el programa SerialForwarder:

```
$ java net.tinyos.sf.SerialForwarder -comm serial@COM5:iris
```

Como se aprecia en la figura de arriba, el programa SerialForwarder recibirá las tramas por el puerto serie USB escuchando por el puerto virtual COM4 que ha sido identificado en nuestro caso, recordar que por defecto esta escucha es por el puerto TCP 9001.

5.5.3.2 Iniciar Oscilloscope GUI

Para ejecutar el programa y aparezca una ventana swing gráfica se escribira el siguiente comando en el prompt de Debian:

```
$ java net.tinyos.oscope.oscilloscope
```

Este programa se comunica con el SerialForwarder y mostrara los datos en código exhadecimal.

CAPITULO VI

ANÁLISIS DE RESULTADOS

En este apartado mostramos el análisis de resultados que se obtuvo al iniciar la captación de datos de los dos nodos sensores junto con la SBC al ejecutar la plataforma propuesta de las redes de sensores.

6.1 RESULTADO DE LOS INDICADORES

En nuestro experimento se evaluará los componentes en la arquitectura estrella propuesta, poniendo en marcha el sistema y obteniendo mediciones que permitan validar nuestra plataforma y el correcto funcionamiento.

Mencionamos que los nodos son identificados como sigue:

Nombre del dispositivo	Descripción
Nodo 1	Nodo sumidero
Nodo 2	Nodo sensor
Nodo 3	Nodo sensor
Nodo 4	Nodo sensor

Tabla 19 – Identificación de los nodos

Si bien es cierto nuestros nodos sensor tienen una placa sensor de varios tipos de transductores (sensores), solo tomaremos como muestra la variable de temperatura de los dos nodos sensor, porque nuestro objetivo es evaluar la ejecución de la arquitectura propuesta.

6.1.1 UBICACIÓN DE LOS NODOS

El lugar donde se desplego los nodos fue en un ambiente de producción de pescadería y mariscos en el terminal pesquero de Villa María (Ver Ilustración 16), el lugar consta de dos pisos, el segundo donde se encuentra la cámara frigorífica y el primer piso donde eviscera el pescado.

Para asegurar la buena conservación en días de este alimento hay varias claves con respecto a la temperatura. En primer lugar, el frigorífico debería ser un sistema donde la temperatura debería estar entre 1 C° y 5 C° en verano o si el almacén está lleno. Y nuestra área de eviscera debería estar entre 10 C° y 15 C°.

Ilustración 16 área de prueba



Fuente: elaboración propia

Con respecto al despliegue, se realiza en dos ambientes, un ambiente abierto a una distancia de 5 metros del nodo 2 al nodo sumidero, de 2 metros del nodo 3 al nodo sumidero y de un área cerrada -el frigorífico- del nodo 4 a 5 metros aproximadamente al nodo sumidero (Ver ilustración 17).

Ilustración 17 área de prueba



Fuente: elaboración propia

6.1.2 MEDICIONES DE TEMPERATURA

En la toma de datos se decidió que se realizara en un día aleatorio y se coja la muestra cada 3 minutos (Ver tabla 20)

Tabla 20 - Resultado de los indicadores

NODO Identificación	Temperatura (°C)	Fecha y Hora
Nodo 2	14,49	2019-12-17 07:02:22
Nodo 3	16,74	2019-12-17 07:02:22
Nodo 4	8,71	2019-12-17 07:02:22
Nodo 2	14,51	2019-12-17 07:05:22
Nodo 3	16,71	2019-12-17 07:05:22
Nodo 4	8,51	2019-12-17 07:05:22
Nodo 2	14,48	2019-12-17 07:08:22
Nodo 3	16,72	2019-12-17 07:08:22
Nodo 4	8,41	2019-12-17 07:08:22
Nodo 2	14,46	2019-12-17 07:11:22
Nodo 3	16,72	2019-12-17 07:11:22
Nodo 4	9,61	2019-12-17 07:11:22
Nodo 2	14,46	2019-12-17 07:14:22
Nodo 3	16,74	2019-12-17 07:14:22
Nodo 4	10,71	2019-12-17 07:14:22
Nodo 2	14,46	2019-12-17 07:17:22
Nodo 3	16,76	2019-12-17 07:17:22
Nodo 4	9,31	2019-12-17 07:17:22
Nodo 2	14,47	2019-12-17 07:20:22
Nodo 3	16,77	2019-12-17 07:20:22
Nodo 4	10,41	2019-12-17 07:20:22

Nodo 2	14,45	2019-12-17 07:23:22
Nodo 3	16,79	2019-12-17 07:23:22
Nodo 4	8,91	2019-12-17 07:23:22
Nodo 2	14,46	2019-12-17 07:26:22
Nodo 3	16,81	2019-12-17 07:26:22
Nodo 4	8,51	2019-12-17 07:26:22
Nodo 2	14,45	2019-12-17 07:29:22
Nodo 3	16,8	2019-12-17 07:29:22
Nodo 4	8,71	2019-12-17 07:29:22

El acumulado de las mediciones en la Tabla 21, nos hacen ver que la temperatura tiene un comportamiento casi lineal en ese día y hora para el nodo 2 y 3, el nodo 3 al estar más cerca a la calle tiene una temperatura más alta, vemos que varía en dos grados centígrados con respecto al nodo 2 en el tiempo de la muestra. Sin embargo, para el nodo 4 que está en la recámara frigorífica la temperatura va a variar cada vez que se ingrese al área, porque ingresa temperatura más alta como se hizo a partir del minuto 08. Otra nota importante es que realizando pruebas en ambiente cerrado el nodo sensor 4 solo llega al rango de emisión de datos aproximadamente a unos 25 metros.

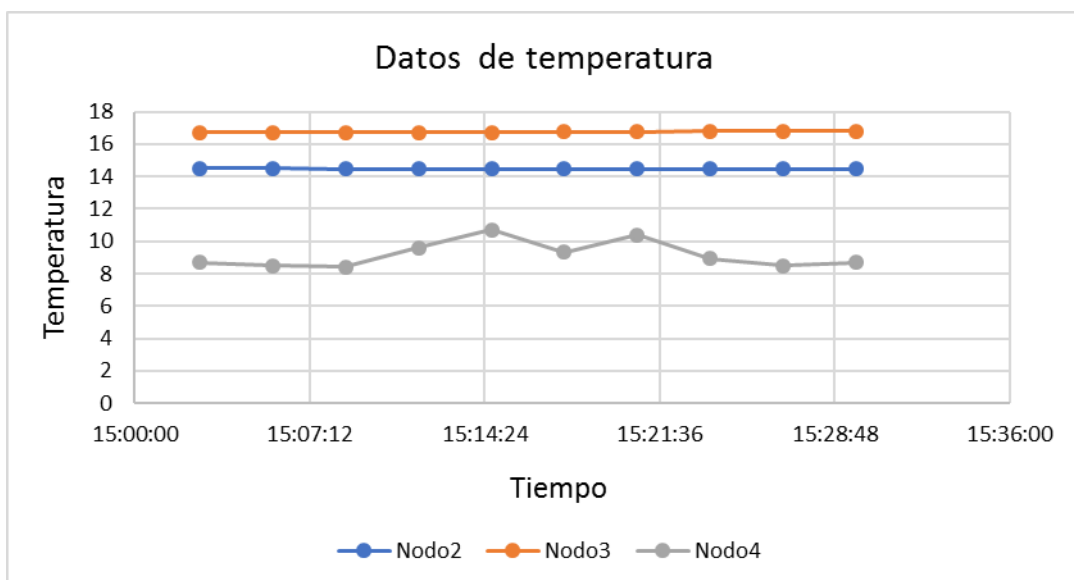


Tabla 21 – Comportamiento de los parámetros de temperatura

CONCLUSIONES Y RECOMENDACIONES

En este último apartado se señala varias conclusiones como consecuencia del proyecto realizado, y además las sugerencias que se pueden realizar en futuros proyectos.

7.1 CONCLUSIONES

En el mundo de las tecnologías el cual se realiza este proyecto, se ve que las tarjetas de placa única SBC, están en proceso de madurez por el avance tecnológico en el cual nos encontramos, el Orange PI 2G IoT y su microprocesador está en el límite de recursos que se requiere para utilizar la plataforma propuesta. Hay que tener en cuenta que cada vez aparecen en el mercado dispositivos tecnológicos con mayor performance de bajo coste facilita la creación de una arquitectura alternativa.

Al principio hubo mucha incertidumbre en el software y hardware a utilizar, porque muchas de las cosas pasan por identificar características del núcleo del microprocesador que nos permitan utilizar un software determinado, por ejemplo, saber si soporta coma flotante para reconocer si es arquitectura armhf que es necesario saber para elegir un tipo de sistema operativo TinyOS y Debian en el microprocesador ARM de la SBC.

Al estar al límite del procesamiento óptimo del microprocesador ARM de la SBC, ha sido muy difícil encontrar complementos tecnológicos como el interfaz TTL que puedan adaptarse con el SBC para la configuración e instalación de Debian o TinyOS, sumando que no hemos encontrado documentación especializada.

La arquitectura planteada de esta tesis se basa también en la ubicuidad al usar tanto en los nodos como en la mini estación dispositivos de tamaño reducido y móvil (por la interconexión GSM/GPRS), así como la utilización de procesamiento mínimo para poder instalar el sistema operativo TinyOS en los nodos sensor, y GNU-Linux en la SBC.

7.2 RECOMENDACIONES

- Se debe crear un middleware que, actuando bajo una perspectiva distribuida para el control y monitorización de datos, sea capaz de reconfigurarse para adaptarse a un entorno cambiante de más tipos de variables, posibilitando cumplir con las restricciones de tiempo real de las redes de sensores. El middleware se debe soportar en una arquitectura propuesta para los nodos en la que se consideran los requisitos reales de las aplicaciones industriales.
- Se recomienda implementar el software multiagente JADE en la SBC, este sistema está basado en el lenguaje java, de ese modo podríamos crear una arquitectura que incluya tecnología inteligente en el procesamiento de datos.
- La solución propuesta se podría integrar otro tipo de nodos sensor que tengan otra placa base como Arduino que se describió en algunas tesis en el estado del arte, y de esta forma tener una arquitectura híbrida, en estos nodos podemos añadir sensores que puedan captar datos químicos como CO₂, monóxido, etc., dado que nuestros nodos sensor solo puede captar datos físicos como temperatura, humedad, luminosidad, etc.
- Esta solución esta implementada para captar datos ambientales, la misma que puede ser incluida una tecnología de identificación de objetos, la cual proporciona RFID, la identificación por radio frecuencia.
- Analizar la posibilidad de incorporar funciones de control a la arquitectura, por ejemplo, en un almacén vitivinícola que haya excedido la humedad, controle el encendido de un artefacto que pueda dar solución a este problema, modificando la humedad en tiempo real.
- Aplicar un estudio de factibilidad para determinar de manera técnica y tecnológica donde se puede implementar una solución real utilizando esta arquitectura en algún sector industrial.

REFERENCIAS BIBLIOGRÁFICAS

Soto Sarango, Á. (2019). *DISEÑO E IMPLEMENTACIÓN DE UNA RED DE SENSORES INALÁMBRICOS PARA EL MONITOREO DE FACTORES AMBIENTALES ABIÓTICOS QUE AFECTAN AL CULTIVO DE SOLANUM PHUREJA*. Cuenca: UNIVERSIDAD POLITÉCNICA SALESIANA SEDE CUENCA.

Aitor Santana. (21 de Marzo de 2018). *EL ANDROIDE LIBRE*. OBTENIDO DE [HTTPS://ELANDROIDELIBRE.ELESPANOL.COM/2013/03/TODO-SOBRE-EL-USB-OTG-QUE-ES-COMO-SE-USA-ES-COMPATIBLE-MI-SMARTPHONE.HTML](https://elandroidelibre.lespanol.com/2013/03/todo-sobre-el-usb-otg-que-es-como-se-usa-es-compatible-mi-smartphone.html)

Almeida, G. (27 de Octubre de 2015). *BITÁCORA SOBRE ANÁLISIS DE NEGOCIOS*. OBTENIDO DE CÓMO APLICAR BUSINESS ANALYSIS A SOLUCIONES DE DATA WAREHOUSE O BUSINESS INTELLIGENCE: <http://gabrielalmeida.com.mx/1043/como-aplicar-business-analysis-a-soluciones-de-data-warehouse-o-business-intelligence/>

Aragon Valladares, J. O. (2014). *DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA DE GESTIÓN DE UNA RED DE SENSORES APLICADA A LA MONITORIZACIÓN DE LA CALIDAD AMBIENTAL EN LA CUENCA DEL RÍO NAPO*. Lima: Pontificia Universidad Católica del Perú.

Business Intelligence Consulting. (04 de Noviembre de 2015). *SAP BUSINESS OBJETCS*. Obtenido de <http://www.biconsulting.com.mx/sap-business-objects/>

Da Paz Teixeira, J. O. (2015). *UNA RED DE SENSORES PARA LAS SMART CITIES*. MADRID: UNIVERSIDAD COMPLUTENSE DE MADRID.

Daqui Solano, O. F. (2011). *DISEÑO E IMPLEMENTACION DE UNA SOLUCION DE MONITOREO REMOTO VIA INTERNET, PARA UNA RED INLAMBRICA DE SENSORES*. GUAYAQUIL: ESCUELA POLITECNICA NACIONAL.

Fernández Cruz, L. F. (2017). *DISEÑO E IMPLEMENTACIÓN DE UNA RED DE SENSORES PARA EL MONITOREO DE VARIABLES CLIMÁTICAS EN UN INVERNADERO DE ORQUÍDEAS*. Cuenca: Universidad Politécnica Sede Cuenca.

- López Egea, J. (2012). *ESTUDIO E IMPLEMENTACIÓN DE UN SISTEMA DE VEHÍCULOS CON UNA RED DE SENSORES INALÁMBRICAS*. CARTAGENA: UNIVERSIDAD POLITÉCNICA DE CARTAGENA.
- López Riquelme, J. A. (2011). *CONTRIBUCIÓN A LAS REDES DE SENSORES INALÁMBRICAS. ESTUDIO E IMPLEMENTACIÓN DE SOLUCIONES HARDWARE PARA AGRICULTURA DE PRECISIÓN*. Cartagena: Universidad Politécnica de Cartagena.
- Morillo Canalejo, A. (2017). *EVALUACIÓN DEL RENDIMIENTO EN EL NIVEL DE RED RPL EN WSN*. SEVILLA: UNIVERSIDAD DE SEVILLA.
- Pallarés Gual, S. (s.f.). *SISTEMA DE LOCALIZACIÓN PARA REDES INALÁMBRICAS DE SENSORES MEDIANTE ZIGBEE*. Castellón: UNIVERSITAT JAUME I DE CASTELLÓN.
- Paredes Vinueza, M. N. (2017). *IMPLEMENTACIÓN DE UN PROTOTIPO DE WSN CON NODOS INTELIGENTES PARA EL SISTEMA DE RIEGO APLICADO A LA AGRICULTURA DE PRECISIÓN PARA EL CER-ESPOCH*. Riobamba: Escuela Superior Politécnica de Chimborazo.
- Torrejon Sampedro, C. (2010). *PLATAFORMA MÓVIL PARA EL CONTROL DE UNA RED DE SENSORES INALÁMBRICA EN UN ENTORNO DOMÓTICO*. MADRID: UNIVERSIDAD POLITÉCNICA DE MADRID.
- Valencia, U. I. (12 de Agosto de 2018). *COMPUTACIÓN UBICUA, LA INFORMACIÓN DE LOS USUARIOS*.
 Obtenido de <https://www.universidadviu.com/computacion-ubicua-la-informacion-entorno-los-usuarios/>
- Villón Valdiviezo, D. (2009). *DISEÑO DE UNA RED DE SENSORES INALÁMBRICA PARA AGRICULTURA DE PRECISIÓN*. Lima: Pontificia Universidad Católica del Perú.

ANEXOS

ACRÓNIMOS

WSN	Wireless sensor networks
PDA	Personal Digital Assistant
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ROLL	Routing over Low power and Lossy Networks
CORE	Constrained Restful Enviroments
COAP	Constrained-node/network Appication
USN	Ubique Sensor Network
ISO	International Organization for Standarization
IEC	The International Electro-technical Comission
BT-LE	Bluetooth Special Interest Group
QoS	Quality of Service
MANTIS	Multimodal Networks In-situ Sensors
ECOS	embedded Configurable operating system
NesC	Network Embedded Systems C
ISM	Industrial, Scientific and Medical
WLAN	Wireless Local Area Network
SRAM	Static Random Access Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
UART	Universal Asynchronous Receiver-Transmitter
I2C	Inter-Integrated Circuit
SPI	Serial Peripheral Interface
SCLK	Serial CLock
MISO	Master In, Slave Out
MOSI	Master Out, Slave In.

CEn	Chip Enable
GPIO	General Purpose Input/Output
APS	Application Support
ZDO	ZigBee Device Object
FCS	Frame Check Sequence
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
MAC	Medium Access Control
DSSS	Direct Sequence Spread Spectrum
PSSS	Parallel Sequence Spread Spectrum
RFD	Reduced Functional Devices
SAP	Service Access Point
SBC	Single Board Computer
OSI	Open Systems Interconnection
BLE	Bluetooth Low Energy
GFSK	Gaussian Frequency Shift Keying
NFC	Near-Field Communications