ANEXO A

MARCO PROBLEMÁTICO EN LA AGRICULTURA DEL PERÚ

1. Situación Actual

La agricultura peruana es una de las actividades económicas más importantes para el país, ya que presenta un alto impacto a la hora de combatir la pobreza rural en el Perú. Su destacada importancia radica en los siguientes factores:

- Excluyendo a Lima, la agricultura ocupa aproximadamente el 40% de la PEA (Población Económicamente Activa). En la sierra alcanza el 55%. [5]
- Aporta con un 6% al PBI (Producto Bruto Interno) Nacional. [6]
- Contribuye con un 3% a las exportaciones peruanas, mediante productos tradicionales como al café y azúcar, los cuales lideran la lista. [7]
- La producción nacional se desarrolla en 2.5 millones de hectáreas, de las cuales el 84% se dedica a la producción de cultivos transitorios y el restante a frutales. Con transitorios nos referimos a que tienen un ciclo de crecimiento menor a un año y pueden ser destinados tanto para el consumo humano o animal, como para fines industriales u otros usos.
- Los principales productos agrícolas peruanos que tienen mayor demanda nacional son el arroz (19%), maíz amarillo duro (14%), papa (13%), maíz amiláceo (10%), trigo (7.5%), cebada grano (7.4%). [8]

2. Principales Recursos

Entre los principales factores que intervienen en la agricultura y que se ven afectados, tenemos los siguientes:

2.1 Tierra

El país cuenta con una superficie agrícola de 7.12 millones de hectáreas (Ha), esto se traduce como el 18.5%, la cual se clasifica en tierras con cultivos y tierras que se encuentran en descanso o no trabajadas. Por otro lado la diferencia, 81.5%, es superficie no agrícola y contiene áreas de pastos naturales, montes y bosques. Esto indica que si hacemos un análisis de las áreas que están inoperativas, podremos aprovechar terrenos aptos y empezar con una producción a mayor escala, realizando un cultivo más intensivo en capital, de mayor tecnología y principalmente orientados a mercados de exportación.

En la Tabla 2.1 se puede observar que el motivo principal de atraso, en el desarrollo de cultivos, es la falta de agua y que esto se ve principalmente en la Costa con un 55%. Por otro lado, la falta de recursos económicos se hace presente con más notoriedad en la selva y la falta de mano de obra, también. Esto es contradictorio,

ya que en esta región se encuentra la mayor cantidad de tierras agrícolas y es cuna de muchos alimentos que se consumen en el hogar y que son utilizados para exportación.

Razón principal	Total	Estructura %	Costa	Sierra	Selva
Total	**774 882.24**	**100**	**301 463.84**	**237 712.78**	**235 705.62**
Falta de agua	378 912.23	48.9	207 826,53	122 923.67	48 162,04
Falta de semilla	32 491.35	4.2	2 929.24	17 612,65	11 949,46
Falta de crédito	186 386.05	24.1	41 740.40	40 403,12	104 242,54
Falta de mano de obra	87 866.64	11.3	11 274.91	31 419,84	45 171,89
Por sanidad, erosión	38 653.52	5.0	19 308.17	13 070,39	6 274,97
Consiguió otro trabajo	5 752.25	0.7	927.09	1 850,27	2 974,89
Robo	2 095.75	0.3	1 662.21	220,14	213,40
Por terrorismo	355.37	0.0	1.28	99,29	254,80
Por desastre natural	14 480.85	1.9	3 467.16	4 088,99	6 924,70
Otra	27 888.23	3.60	12 326.86	6 024,43	9 536,93

Tabla 2.1 Superficie no trabajada en Perú [9]

2.2 Riego

El país cuenta con una superficie agrícola que debe estar siempre bajo riego, sin embargo las estadísticas muestran una realidad distinta. En la Tabla 2.2, podemos observar que la superficie agrícola bajo riego se encuentra principalmente en la Región Costa con un 57% y la superficie bajo secano se encuentra mayoritariamente en la Región de la Sierra con un 50.7%.

Región Natural	Total		Riego		Secano	
	Ha.	%	Ha	%	Ha	%
Total	**7 125 007.77**	**100**	**2 579 899.88**	**100**	**4 545 107.88**	**100**
Costa	1 686 777.58	23.7	1 469 422.55	57	217 355.03	4.8
Sierra	3 296 008.11	46.3	989 481.65	38.4	2 306 526.45	50.7
Selva	2 142 222.09	30.1	120 995.68	4.7	2 021 226.40	44.5

Tabla 2.2 Superficie agrícola bajo riego y secano [9]

2.3 Minifundios

La mayoría de los agricultores (casi el 85%) son dueños de parcelas con menos de 10 hectáreas, es decir, sus tierras ocupan un área entre 3 y 10 ha. Además de ello, según estadísticas, existen 5.7 millones de predios o terrenos rurales, los cuales deberían estar inscritos en registros públicos; sin embargo, solamente un tercio (1.9 millones) lo está. Esto representa un impedimento importante en el desarrollo agrícola de nuestro país, ya que eleva la miseria y pobreza de los mismos campesinos, quienes al morir dejan sus tierras como herencia dividiéndolas entre los hijos y tocándole a cada uno una porción cada vez más pequeña. En la Fig. 2.1 se puede observar las desventajas del minifundio y como desencadena otros problemas que afectan al campesino.

Fig. 2.1 Desventajas de los minifundios [10]

2.4 Otras razones

Algunos otros motivos que escapan a la clasificación que se realizó en este apartado, no pueden pasarse por alto y por ello también se indican en las siguientes líneas:

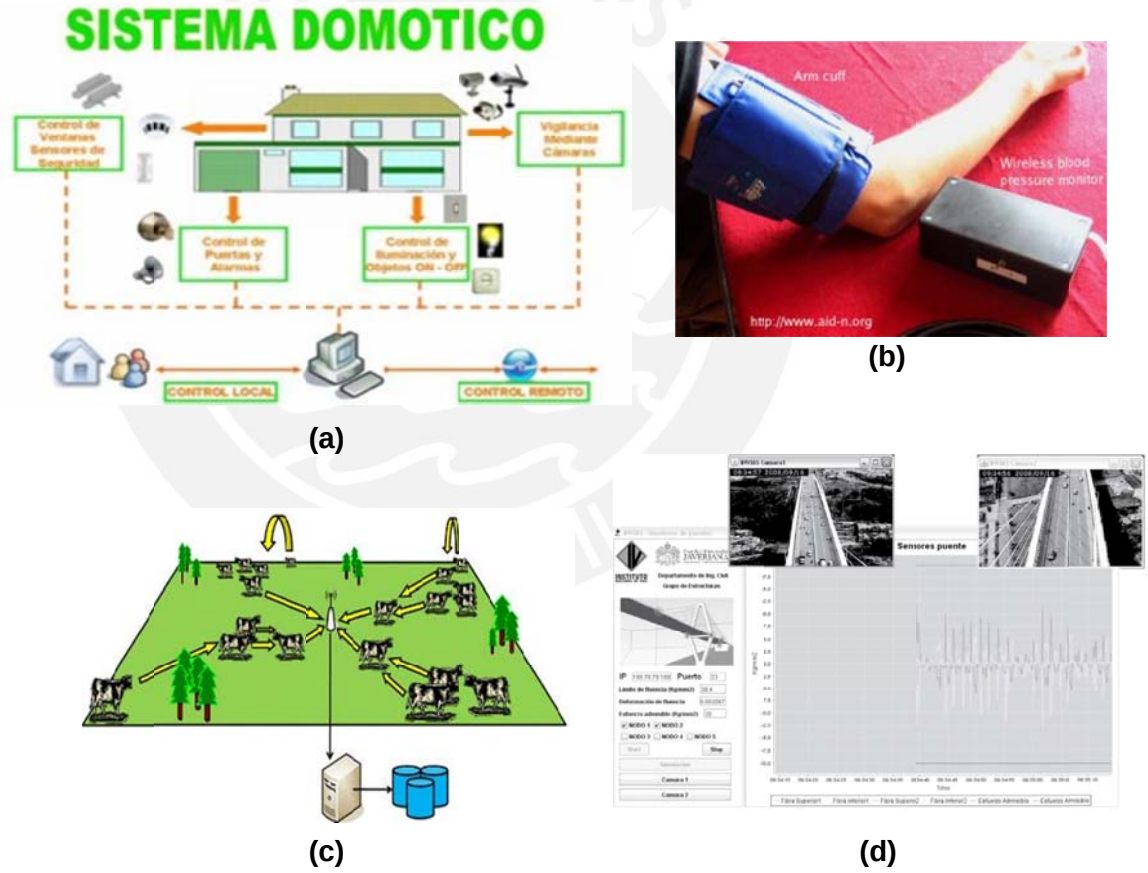
- Falta de apoyo e inversión del sector público y privado en la innovación de tecnologías y desarrollo de proyectos para monitorear los cultivos a fin de evitar las plagas y los posibles daños que paralizan su producción.
- Capital humano escaso y bajo nivel de la calidad de vida del agricultor.
- Mal funcionamiento de los servicios agrarios, y con esto nos referimos a falta de capacitación, apoyo de personal profesional, información e investigación. La mayoría de conocimientos adquiridos es empírico.
- Precios bajos y distorsionados.
- La agricultura vista como forma de vida para muchos agricultores y no como actividad económica, lo cual no aporta superación al agricultor.
- Comunicación inadecuada entre parte política y parte técnica. Las personas con cargos en el estado o ministerios no ejecutan un plan de mejoría y por el contrario solo buscan soluciones rápidas pasando por encima del agricultor.
- Poco acceso a tecnologías (radio, televisor, internet). Esta es la razón más notoria a nivel de provincia. Se puede apreciar fácilmente, las condiciones precarias en la que viven y se desarrollan y como es que su manutención y sustento se aboca a la agricultura, entre otras actividades.

ANEXO B

APLICACIONES DE LAS REDES WSN

1. Aplicaciones

Este tipo de redes entrelaza tecnologías de información con comunicaciones inalámbricas en dispositivos compactos y resistentes. Gracias a ello es que podemos beneficiarnos a través de su diseño e implementación en varios sectores de nuestra vida, entre ellos la medicina, la agricultura, la domótica, los ambientes militares, monitoreo de estructuras, etc. También podemos apreciar su gran virtud funcional en procesos industriales, medios de transporte y un sinfín de áreas y actividades donde se puede sacar provecho y optimizar recursos. A continuación, se dará una breve explicación de dichas aplicaciones, en los aspectos más importantes de nuestra vida diaria. En la Fig. 1.1, tenemos ejemplos de sistemas implementados a la fecha.



(a) **(b)** **(c)** **(d)**

Fig. 1.1 (a) Sistema domótico. [14] **(b)** Dispositivo que mide frecuencia cardiaca, saturación de oxígeno, datos de ECG y lo transmite en tiempo real a PDA's, ordenadores portátiles, ambulancia. [15] **(c)** Comunicación multi-salto para reunir datos por telemetría del ganado agrícola. [16] **(d)** Diagrama esquemático del viaducto César Gaviria Trujillo y posición de sensores. [17][18]

1.1 Medicina

La salud es un deber y derecho del ser humano y los medios para preservarla son importantes. Por lo tanto, se vienen desarrollando diferentes sistemas para monitorear los signos vitales del paciente obteniendo resultados en tiempo real, por ejemplo: frecuencia cardiaca, presión arterial, frecuencia respiratoria, pulsaciones, nivel de azúcar, etc. Los resultados pueden ser monitoreados de forma remota. De manera opcional y como propuesta de mejora, estos nodos sensores pueden ser instalados en el hogar del paciente a fin de que si presenta alguna recaída, la alerta llegue a tiempo para que pueda ser atendido por profesionales de manera inmediata.

1.2 Militar

El origen de esta tecnología se debe a la necesidad de comunicarse y detectar otros buques durante las guerras y, debido a ello se diseñó y empleó un equipo llamado “SONAR”, el mismo que utilizaba impulsos acústicos.

Sus características le permiten ser usados en campos de batalla, ya que algunos nodos pueden dejar de funcionar, sin embargo la red no pierde operatividad. Los vehículos de guerra, como por ejemplo: tanques, aviones, misiles, torpedos, etc., son una buena representación de esta aplicación, pues en ellos se despliegan los nodos sensores a fin de que puedan ser dirigidos hacia una posición exacta y a su vez, permite la comunicación y coordinación entre ellos. Además, pueden servir para detectar ataques del enemigo, posibles amenazas, invasión y armas de todo tipo.

1.3 Medio Ambiente

Aquí los nodos se despliegan a fin de que sean capaces de monitorear el comportamiento del entorno que rodea a estos dispositivos y a su vez se mantengan alertas a determinadas catástrofes naturales como por ejemplo: incendios en el bosque, inundaciones, aluviones; o a sucesos causados por el hombre y sus acciones como la explotación de una especie en su propio hábitat o el tráfico en la ciudad.

1.4 Sistema Vehicular

Los vehículos en los que nos transportamos a diario no son la excepción a esta tecnología, ya que cada vez son más las aplicaciones que permiten lograr velocidad, comodidad y rendimiento durante un viaje o camino a nuestro destino. Por mencionar

algunos ejemplos tenemos la detección de curvas y la adaptación de la iluminación del carro hacia ellas, sistemas automáticos para encender o apagar las luces y el limpiaparabrisas, ajuste del cinturón de seguridad, etc.

Hoy en día es más fácil encontrar proyectos donde los consorcios e instituciones de investigación pretenden motivar diseños futuristas con aplicaciones más completas y siempre conservando el confort sin dejar de lado el rendimiento y la capacidad motora del vehículo. Algunos de estos proyectos son “Trackss” [19], “Caring Cars” [20].

1.5 Domótica

La domótica es uno de los campos que constituye una fusión de varias aplicaciones para la vivienda o edificios, cuyo fin es de reducir costos a largo plazo y buscar el aumento de confort sin dejar de lado la seguridad personal y patrimonial.

Nos referimos a la automatización y control (encendido/apagado, apertura/cierre y regulación) de aparatos y sistemas de instalaciones eléctricas y electrotécnicos (iluminación, climatización, persianas y toldos, puertas y ventanas motorizados, el riego, etc.) de forma centralizada y/o remota.

Como ejemplo tenemos un sistema de cine en casa, donde todo se activa con un solo botón, el cual enciende el proyector, luego baja una pantalla a la vez que las cortinas se cierran a fin de dar un aspecto más oscuro. Mientras los equipos de video se van encendiendo, las lámparas se apagan y los focos halógenos disminuyen su intensidad progresivamente. Todo esto guarda sincronismo con el proyector, por tanto, hasta que no aparezca la imagen proyectada, la luz no se apagará completamente. [21]

1.6 Agricultura

El uso de redes WSN en este ambiente se desarrolla cada vez con mayor fuerza, ya que las áreas agrícolas son manejadas por los mismos pobladores de la zona, quienes no poseen los medios económicos ni tecnológicos para detectar a tiempo las deficiencias y erradicarlas. Por ello, el presente proyecto de tesis está enfocado en esta área y en la optimización de costos, así como un mejor resultado en la cosecha de los agricultores y acceso a la información. Algunas acciones llevadas a cabo en las parcelas son las siguientes:

- Seguimiento de tiempo atmosférico, humedad, temperatura, control de cauce fluvial, nivel del río, etc.
- Monitorización de condiciones climáticas, estado del suelo para cosecha, cálculo de insumos y agua para agricultura de precisión.
- Sistema de alarma para proteger la cosecha de animales salvajes o intrusos.

ANEXO C

ESTÁNDAR IEEE 802.15.4 Y ZIGBEE

1. Estándar IEEE 802.15.4

Esta norma define el protocolo de comunicación entre los distintos dispositivos que conforman la red inalámbrica y esto lo consigue a través de la configuración en las capas inferiores del modelo OSI, las cuales son física y de acceso al medio (MAC), tal como se observa en la Fig. 1.1.

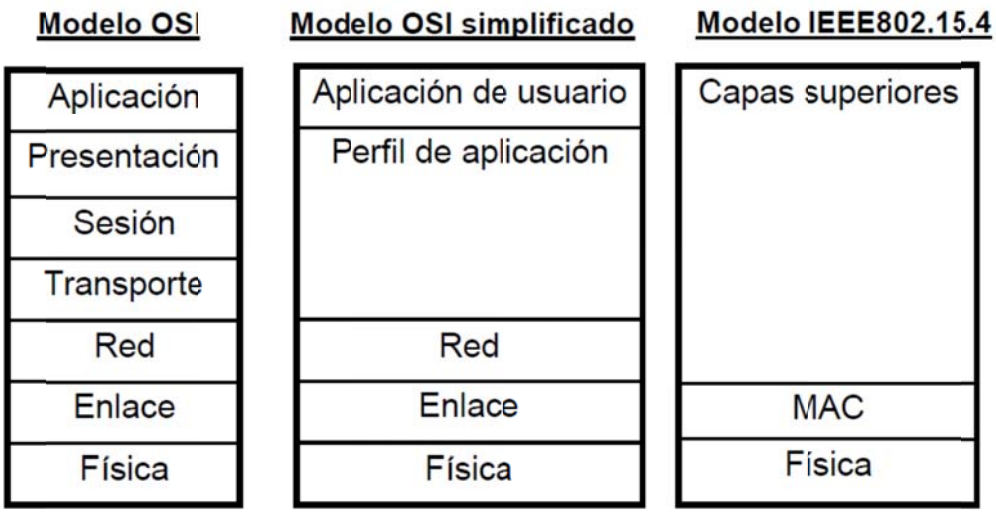


Fig. 1.1 Arquitectura de capas orientadas a WSN [34]

Así mismo cabe señalar que existen otros estándares (Tabla 1.1), los mismos que son usados en aplicaciones que requieren otro tipo de requisitos como mayor ancho de banda y mayor tasa de datos, es decir no en redes WSN. Para este este tipo de redes (WSN) la atención está en el bajo consumo de energía principalmente, lo cual implica una menor tasa de datos y una menor cobertura. Para ello se usan tramas “Beacon”, las cuales permiten sincronizar la red para que los nodos permanezcan la mayor parte del tiempo dormidos, y solo unos cuantos estén monitoreando, dejando la posta al siguiente cuando intercambien sus roles. De manera adicional, estas redes son elegidas para entornos hostiles, por ello se han asegurado otorgándole ciertos factores que la hacen más resistente como:

- CSMA-CA: se basa en la detección de la portadora para evitar colisiones.
- Tramas con confirmación (ACK): nos indica que la trama ha sido recibida correctamente.

- Verificación de redundancia cíclica (CRC): protege las tramas a través de datos redundantes para que si hay error se detecte y se solucione.

Esta norma define las bandas de 868MHz, 915MHz y 2.4GHz, no obstante, tiene la ventaja de trabajar en bandas sin licencia. De ello se desprende que trabaja mejor en la de 2.4GHz que se utiliza en todo el mundo. Su rango de frecuencias va de 2405 MHz a 2480MHz y utiliza 16 canales (11-26). Tiene una tasa de datos de 250Kbps.

Estándar	Ancho de Banda	Consumo de Potencia	Ventajas	Aplicaciones
WI-fi	Hasta 54 Mbps	400 mA transmitiendo 20 mA en reposo	Gran Ancho de banda	Navegar por internet, redes de ordenadores, transferencia de ficheros
Bluetooth	1 Mbps	40 mA transmitiendo 0.2 mA en reposo	Interoperatividad, sustituto del cable	Wireless USB, móviles, informática doméstica
802.15.4	250 Kbps	1.8 mA transmitiendo 5.1 uA en reposo	Batería de larga duración, bajo coste	Control remoto, productos dependientes de la batería, sensores, etc.

Tabla 1.1 Comparación entre estándares inalámbricos [35]

1.1 Capa Física (PHY)

Su objetivo es modular y demodular la comunicación que se realiza por el transceptor, controlar el consumo de energía de la señal y seleccionar los canales de acceso, además de indicar la calidad de enlace. Debido a que la comunicación es inalámbrica, por radiofrecuencia, en la Tabla 1.2 se indican las bandas de frecuencia donde se permite que la red opere.

PHY (MHz)	Frecuencia de Banda (MHz)	Difusión		Parámetros de datos		
		Tasa de chip	Modulación	Tasa de bits(Kb/s)	Velocidad de símbolo (ksymbol/s)	Símbolos
868/915	868-868.6	300	BPSK	20	20	Binario
	902-928	600	BPSK	40	40	Binario
868/915 (opcional)	868-868.6	400	ASK	250	12.5	20-bit PSSS
	902-928	1600	ASK	250	50	5-bit PSSS
868/915 (opcional)	868-868.6	400	Q-PSK	100	25	16-ary Ortogonal
	902-928	1000	Q-PSK	250	62.5	16-ary Ortogonal
2450	2400 – 2483.5	2000	Q-PSK	250	62.5	16-ary Ortogonal

Tabla 1.2 Bandas de frecuencia para la capa física [35]

1.2 Capa de Acceso al Medio (MAC)

Esta capa se encarga de dos servicios. Uno de ellos es servicio de datos MAC, el cual está capacitado para permitir la comunicación de las MPDU (MAC Protocol Data Units). Mientras que el otro es el servicio de gestión MAC, el cual vendría a ser la interfaz entre la entidad de gestión de la capa MAC con el servicio de acceso (SAP).

Entre sus principales funciones está la de otorgar los servicios para que los dispositivos puedan integrarse o desintegrarse de la red; y controlar el acceso a los canales compartidos. Se encarga, además, de la generación de beacons y permite la gestión de Guarenteed Timeslot (GTS). Los beacons son tramas que el coordinador manda para sincronizar, identificar la PAN y configurar la supertrama [36][37]

Utiliza la tecnica CSMA CD (Carrier Sensie Multiple Acces Collision Avoidance) y de esta manera evita colisiones de tramas. En el caso de que un nodo desee transmitir una trama, en primer lugar el tansceptor escuchara el medio. En caso que el medio

este ocupado, la transmisión no podrá realizarse, y tendrá que esperarse un cierto tiempo para volver a intentarlo, así sucesivamente hasta que encuentre el canal libre.

2. Zigbee

Para elegir entre el estándar IEEE 802.15.4 o el estándar ZigBee, debemos hacer previamente un análisis de lo que nuestro proyecto requiere y según ello, elegiremos el estándar que tenga características más específicas con la aplicación. Debemos dejar claro que no es que una sea mejor que la otra. La diferencia entre ambas radica en lo siguiente:

- Throughput y latencia

  ZigBee es una extensión de la arquitectura 802.15.4, motivo por el cual requiere mayor latencia (retardos de la red) y agrega más overhead (desperdicio de ancho de banda por información adicional), reduciendo el throughput (cantidad de datos por unidad de tiempo que se entregan en un nodo).

- Self-healing y routing

  En 802.15.4 solo se permite la comunicación directa entre un nodo y el otro, sin intermediarios. Es decir, una comunicación punto a punto. Mientras que ZigBee, provee saltos, mediante los cuales se transmite la información de un transmisor hasta el receptor final. Esto permite mayor cobertura a la red.

- Complejidad de la red

  Una red 802.15.4 tan sólo permite la topología estrella, es decir, punto a multipunto; o peer-to-peer con visión directa. Por otro lado, Zigbee debido al uso de sus nodos intermedios para transmitir la data y a la no necesidad de visión directa, permite ser más flexible en ese sentido, llegando incluso a sitios remotos, donde es difícil que una persona pueda acceder.

- Complejidad de la pila

  Dado que ZigBee se basa en 802.15.4, la pila de protocolo del primero es mucho más compleja que la de este último. Partiendo de ello, una pila 802.15.4 es lo suficientemente pequeña como para caber en micros de pocos recursos.

Por otra parte, una pila ZigBee requiere mucha mayor cantidad de memoria y micros más caros. [37]

El desarrollo de la tecnología se centra en la sencillez y el bajo coste más que otras redes inalámbricas semejantes de la familia WPAN, como por ejemplo Bluetooth. El nodo ZigBee más completo, requiere en teoría cerca del 10% del hardware de un nodo Bluetooth o Wi-Fi típico; esta cifra baja al 2% para los nodos más sencillos. No obstante, el tamaño del código en sí es bastante mayor y se acerca al 50% del tamaño que el que maneja Bluetooth. [38][39]

ANEXO D

ESTADOS DEL NODO EN UNA RED WSN

1. Estados del nodo

Una vez que desplegamos nuestras motas por todo el terreno. Estas podrán interoperar entre estos cuatro estados, sin embargo, el modo reposo/espera es el que se le asigna por default, si es que no está en los otros. Es importante entender cómo se comportan en cada estado. Esto vamos a tomar en cuenta para nuestro diseño, puesto que a mayor ahorro de energía mejor, aunque contamos con una placa solar que permite ser más flexibles con ello.

- Modo Tx y/o Rx

Este es el modo activo en el que se encuentran los nodos finales cuando reciben paquetes RF o cuando contienen, estos paquetes, en el buffer de salida, los cuales serán transmitidos a otro nodo. Cuando el nodo final está despierto, primero busca algún dispositivo que le permita unirse a la red Zigbee. Entonces, se forma un vínculo padre-hijo. Siendo el padre, un nodo router o un nodo coordinador y, el hijo por descarte, el nodo final. El "hijo" envía una petición de sondeo al padre, la cual una vez recibida, permite que el "padre" busque en la cola de paquetes si tiene algo para enviarle. Luego de ello, envía un acuse de envío (ACK), indicando SI o NO.

Si la rpta. es sí, entonces se queda despierto hasta recibir todos los paquetes RF. De lo contrario, vuelvo al modo sleep o al modo reposo/espera (según configuración). Ver Fig. 1.1. Por otro lado, los nodos router y nodos coordinadores, mientras se encuentran despiertos, realizan tablas con las direcciones de los módulos Xbee conectados a ellos, así como las rutas y los saltos que requieren para enviar un paquete a otro destino.

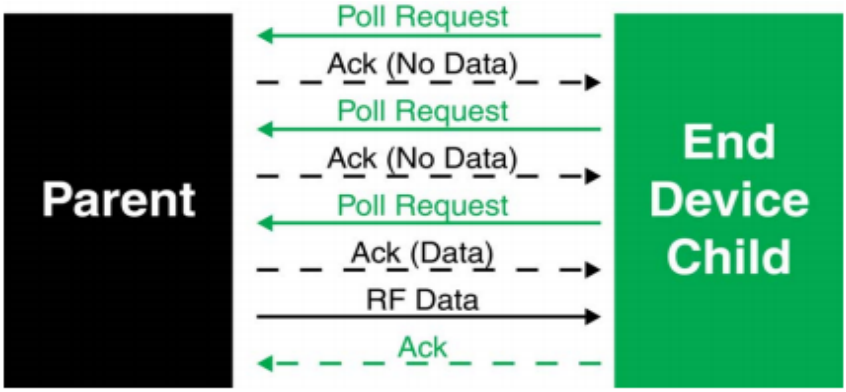


Fig. 1.1 Operación de nodo final [39]

➢ Modo Sleep o de sueño

También conocido como modo de Bajo Consumo. Se puede configurar de dos formas:

- Pin sleep, permite determinar cuando el nodo debe dormir y cuando levantarse según el estado del pin Sleep_RQ (pin 9). Se habilita cuando SM=1
- Cyclic sleep, permite que el nodo duerma por un tiempo definido y despierte un ratito para sondear a sus vecinos por algún paquete que esté dirigido a él. Luego vuelve al modo sleep. Se habilita cuando SM=4 o SM=5.

Existe un modo más (SM=0) que significa que este modo está desactivado por default. El período para cualquiera de las formas explicadas, es configurado con los comandos SP, SN y SO.

➢ Modo Comando

Nos da acceso como usuario para configurar el módulo Xbee a través de la pestaña "Terminal" del programa X-CTU con los comandos AT (Fig. 1.2). Una vez dentro de esta pestaña, podemos configurar, ajustar o modificar parámetros. En nuestro caso, lo utilizaremos para comprobar con que dispositivos nos estamos conectado, mediante el comando "ATND". Este nos devuelve algunos valores que representan lo siguiente:

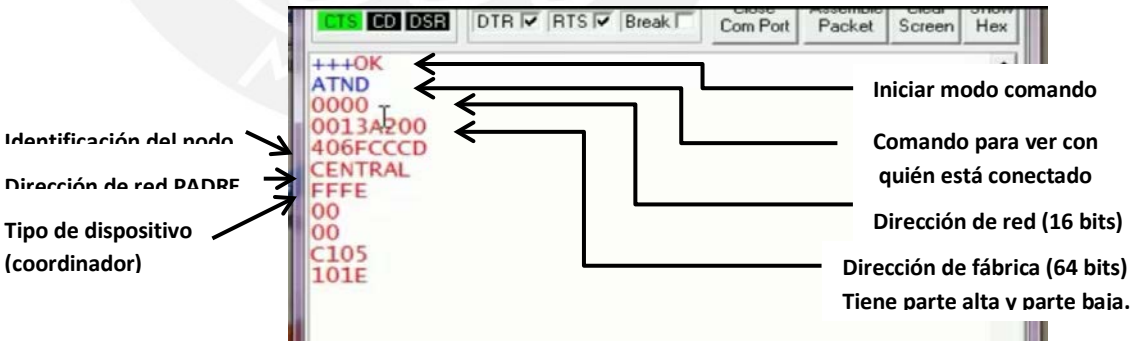


Fig. 1.2 Comprobación de dispositivos conectado al módulo Xbee S2

➢ Modo reposo/espera

Es cuando no está transmitiendo ni recibiendo, ni ahorrando energía, ni en el modo de comandos.

ANEXO E

CARACTERÍSTICAS GENERALES DE LOS ELEMENTOS QUE CONFORMAN EL DISEÑO DEL SISTEMA DE MONITOREO Y CONTROL

1. Seeeduino Stalker v2.3

Esta placa mejorada es un dispositivo sensor que trabaja de forma inalámbrica y es compatible con Arduino (Fig. 1.1). Se basa en una tarjeta con microcontrolador Atmega328P [41] de 8 bits que consume menos potencia que su análoga Atmega328 (sin la “p”). Además de ello dispone de las siguientes memorias: 32kB en memoria Flash, 1kB en memoria EEPROM y 2kB en memoria RAM. Cuenta con un chip DS3231 [42], el cual es un reloj de tiempo real que lleva consigo un oscilador de cristal con compensación de temperatura integrada que usa para recalibrarse y ampliar la precisión de este dispositivo. Así pues, nos entrega información del momento temporal en segundos, minutos, horas, días, meses, años. Podemos programarlo para 24 o 12 horas (AM/PM). Esto es muy importante, por ejemplo, cuando queremos saber los momentos exactos de la toma de cada dato.

Utiliza el bus I^2C para programar acciones, como por ejemplo, hacer que la mota hiberne y se despierte en un determinado momento gracias al uso de interrupciones o para acceder al sensor de temperatura integrado. Además trae consigo una fuente de energía que se incorpora en la placa Seeeduino, la cual tiene forma de moneda y cuyas características se pueden observar en la Tabla 1.1.

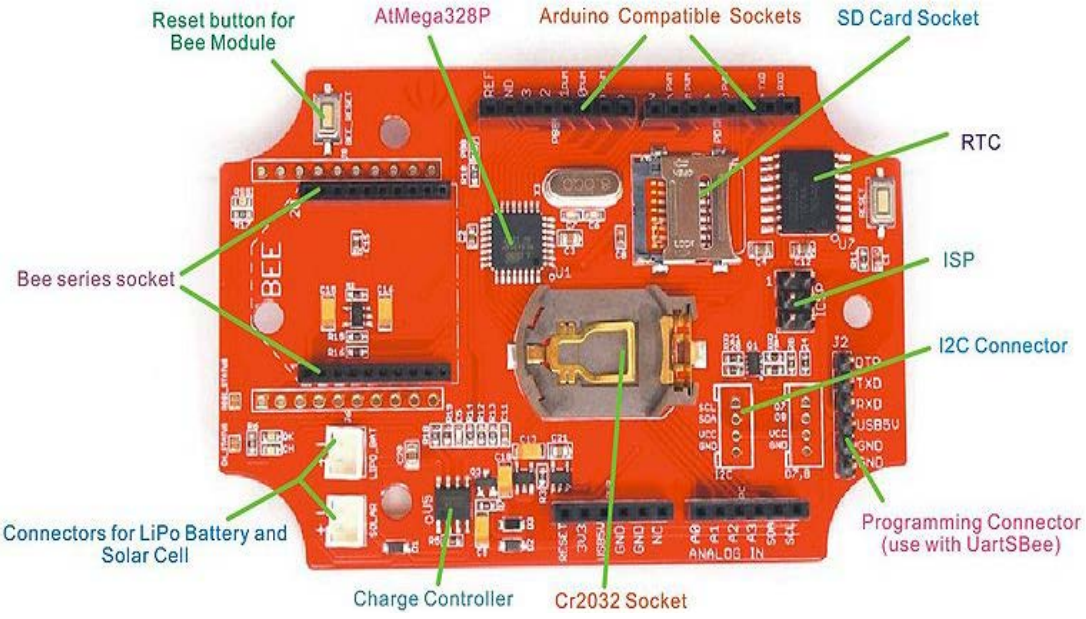


Fig. 1.1 Partes de la mota Seeeduino Stalker V2.3 [40]

Nombre del IEC	CR2032
Primera letra "C"	Indica sistema electroquímico usado: Litio – dióxido de manganeso
Segunda letra "R"	Indica forma circular
2032	Diámetro: 20,0 ± 0,2 mm Altura: 3,2 ± 0,5 mm
Nombre ANSI / NEDA	5004LC
Capacidad típica	220 mAh
Tensión nominal	3,0 V
Rango de temperatura utilizable	**-20 a 85 ° C**

Tabla 1.1 Características de pila CR2032

Entre un sinfín de recursos que nos ofrece esta placa, encontramos una ranura para tarjeta micro SD de 2GB de capacidad, la cual sirve para almacenar los datos en la mota y más tarde enviarlos por la red a un servidor. Asimismo, nos ofrece 4 pines analógicos con conversor interno ADC y 14 pines digitales. Algunos de los cuales usaremos para conectar los sensores y actuadores.

Los protocolos de comunicación con los que trabaja son los siguientes: I^2C, UART, SPI.

- Protocolo I^2C

  Seeeduino ha separado una fila extra para conexiones I^2C, las mismas que están reservadas para reducir la complejidad de la placa. Para ello, solo se debe soldar unos jumpers y listo. Trabaja tanto con dispositivos de 3.3V o de 5V. Las señales con la que cuenta este conector son SCL, señal de reloj que sincroniza el sistema; SDA, señal por la que se transmiten los datos; VCC y GND.

  Este bus permite conectar, por lo general, un microcontrolador a uno o varios periféricos de baja velocidad, pues comparten todos ellos las mismas líneas del bus (SDA, SCL y GND), sin embargo, cada equipo tiene su propia dirección de esclavo

y es única. Estos equipos pueden ser ADCs, DACs, memorias EEPROMs, RTC, potenciómetros, entre otros. En este diseño no haremos uso de este protocolo.

- Protocolo UART

Otro conector que debemos destacar es el UartSbee, el cual usaremos para programar el microcontrolador. Esto se logra a través del chip FT232RL. Este último hace de conversor USB a serial y por ello es que se puede conectar a la placa seeeduino con comunicación serial y a la computadora con un cable mini USB. También cuenta con un regulador de voltaje y un interruptor que le permite elegir entre dos valores de tensión que quiera proveer a la placa seeeduino (3.3V o 5V). Nos da la opción de conectar otros módulos Bee a la placa Seeeduino, como los que aparecen en la Fig. 1.2. Cabe aclarar que para este caso solo trabajaremos con Xbee S2.

Hay que tener en cuenta que la mota seeeduino no tiene lugar para la placa UartSbee, motivo por el cual se destinó 5 pines denominados: VCC, TXD, RXD, DTR (para controlar la señal de reset) y GND. La conexión va de la siguiente manera entre Seeeduino y UartSbee, respectivamente (Fig. 1.3): USB5V con VCC, RXD con TXD, TXD con RXD, GND con GND y DTR con DTR. Es relevante realizar la conexión tal y como se indica ya que más adelante la utilizaremos para cargar nuestro programa a través de la IDE de Arduino.

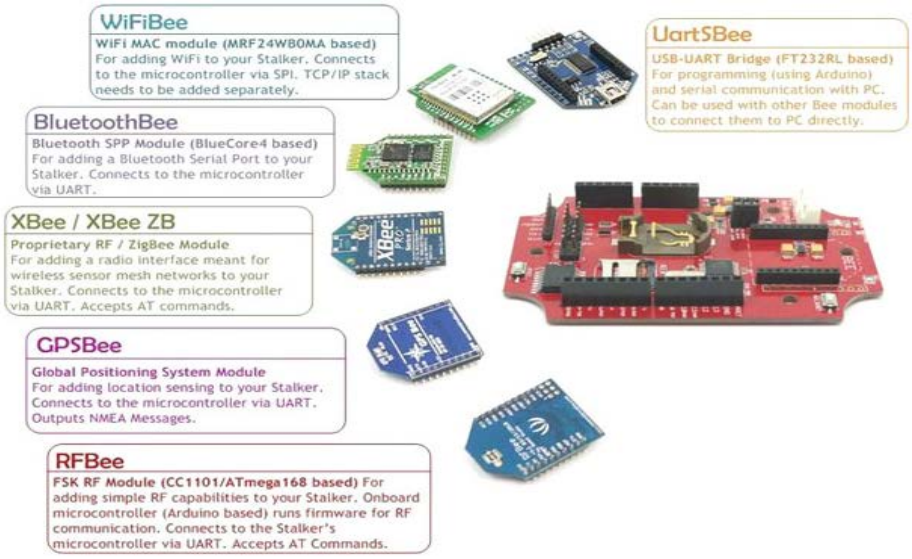


Fig. 1.2 Módulos de radio que pueden ir conectado al UartSbee [40]

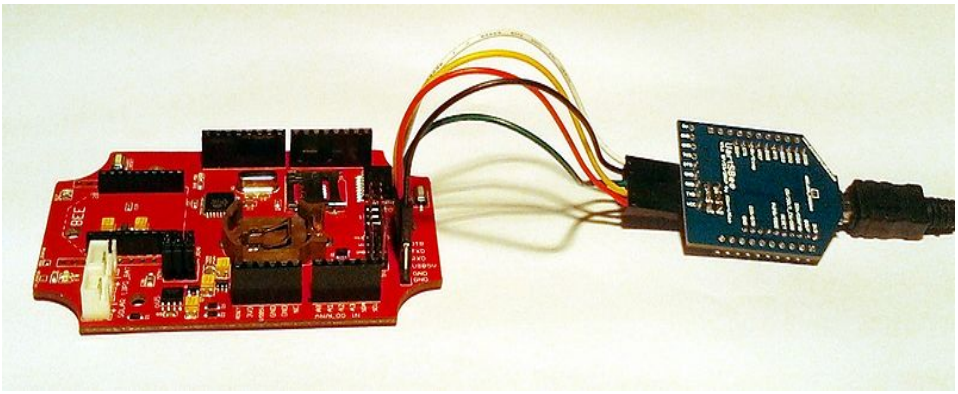


Fig. 1.3 Conexión entre placa Seeeduino y módulo UartSbee [40]

- Protocolo SPI

Comunicación serial full dúplex, es decir que los dispositivos conectados a este bus pueden actuar como transmisor o receptor al mismo tiempo. La placa Seeeduino tiene 6 pines destinados a este conector (Fig. 1.4).

MISO: línea de esclavo para enviar datos al maestro
MOSI: línea principal para enviar datos a los periféricos
SCLK: señal que sincroniza el envío de datos
SS/SELECT: señal destinada a seleccionar un esclavo o decirle que se active.

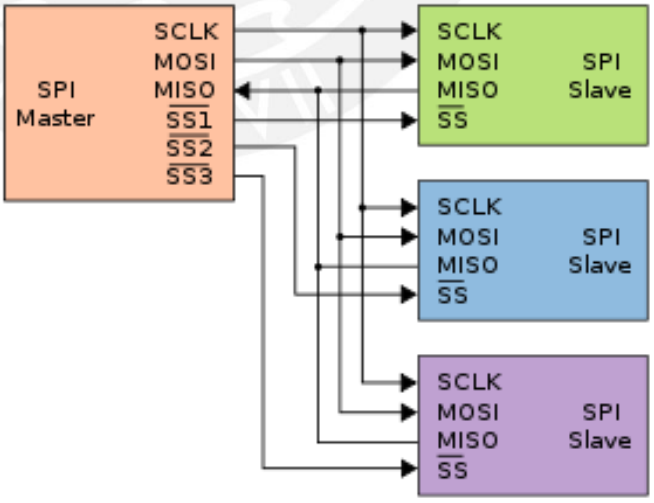


Fig. 1.4 Distribución de pines para ISP [44]

Este protocolo compite mucho con el I^2C, ya que es más rápido y sencillo a la hora de transmitir datos. Además de esto, consume menos energía porque tiene un hardware simple. Su contraparte es que utiliza más pines de cada chip y no tiene forma de comprobar errores. Lo cual resulta en la posibilidad de que el maestro, sin saberlo, esté enviando información sin que llegue a ningún esclavo. Además presenta mucho ruido en la comunicación.

Este protocolo no lo vamos a usar por ahora, pero es importante saber las herramientas con las que contamos al adquirir esta placa.

Seeeduino, entre sus ventajas, nos ofrece un software libre; es decir, que se puede descargar de manera gratuita a través de su página web. Asimismo es open-hardware, es decir, que sus especificaciones, datasheet y diagramas esquemáticos son de acceso público, lo cual nos permite entender sus conexiones y funcionamiento.

La fuente de alimentación lo constituye una batería de litio-ion de 3.7V, 980 mAh, sin embargo, puede ser alimentado por un panel solar de 0.5W, o incluso cuando se conecta al UartSbee y este, a su vez, a la computadora. Todo esto va dentro de un kit (Fig. 1.5), el cual viene en una caja de plástico resistente al agua para proteger el circuito de lluvias fuertes, tan características en la selva. Esta ventaja es importante, pues la prioridad se encuentra en el máximo tiempo de vida del sistema. La placa Seeeduino consta de un integrado, CN3063 [46], el cual viene a ser un gestor de carga de la batería de litio a través del panel solar. Este dispositivo lo que hace es ajustar automáticamente la corriente de carga en función de la capacidad de salida de la fuente de alimentación de entrada, de forma tal que maximiza el uso de la corriente.

Fig. 1.5 Kit Seeeduino Stalker V2.3 [40]

2. Sensores Grove

Entre la diversidad de sensores tentativos para nuestro diseño de red WSN, seleccionamos dos sensores de la marca Grove que estarán conectados a cada mota cuyas características serán descritas a continuación:

2.1 Sensor de Temperatura LM35

Como se puede apreciar en la Fig. 2.1, este sensor es el que usaremos para medir el entorno ambiental en el que se desarrolla nuestra plantación. Está calibrado para un amplio rango de temperaturas, entre -55° y 150° (mide temperaturas menores a 0°, sin embargo no las necesitamos para este caso). Tiene un factor de conversión de 10 mV/°C. Además, no requiere ninguna calibración externa o ajuste para proporcionar una precisión típica de ± 1.4 ºC a temperatura ambiente, y ± 3.4 ºC a lo largo de su rango de temperatura. Para mayor información ver Tabla 2.1.

Para convertir a temperatura lo que se lee por el pin donde está conectado el sensor, utilizamos la fórmula básica:

Tempera en °C

ºC = [1.2 * (Valor en pin de Seeeduino) * 100.0] / 1024

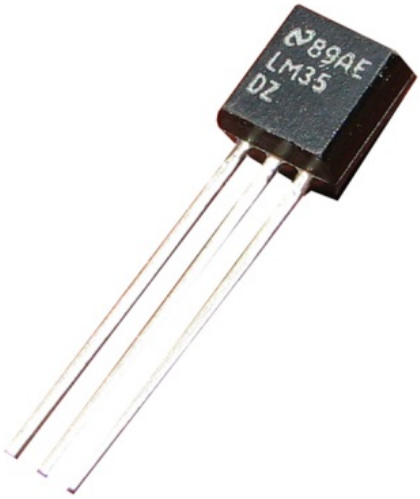


Fig. 2.1 Sensor de temperatura ambiental LM35 [47]

ITEMS	TÍPICO	LIMITE TESTEADO	LIMITE DISEÑADO	UNIDADES
Precisión	±0.4	±1	-	°C
No Linealidad	±0.3	-	±0.5	°C
Ganancia de sensor	+ 10	+ 9.8, +10.2	-	mV/°C
Regulación de la carga	±0.4	±1	-	mV/mA
Corriente Inactiva	56	80	-	uA

Tabla 2.1 Características eléctricas de sensor [48]

2.2 Sensor de Humedad de Suelo SEN92355P

El sensor en mención es el más importante para nosotros, ya que a partir de los valores tomados por este, se determinará cuándo y cómo funcionará nuestro sistema de riego automático. En la Fig. 2.2 se observa el sensor, el cual es de fácil uso y barato. Hay que resaltar que la señal de salida es analógica. Ver Tabla 2.2 para especificaciones técnicas del elemento. Más adelante se explicará cómo programarlo.

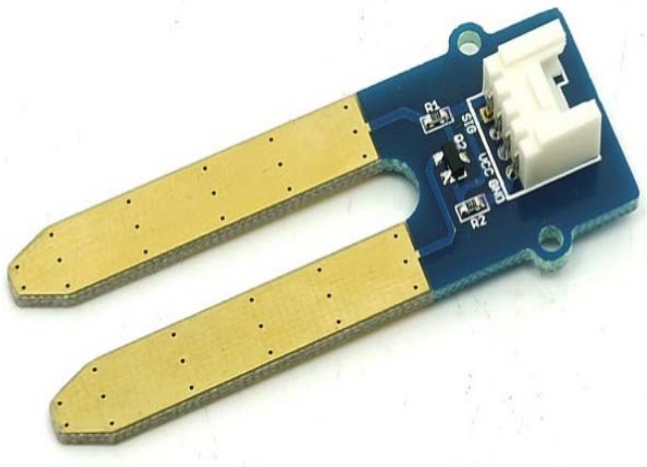


Fig. 2.2 Sensor de humedad de suelo [49]

ITEM	TIPO DE SUELO	MÍNIMO	MÁXIMO	CONVERSIÓN	UNIDADES
V_{CC}	-	3.3	5	-	Volts
I_{FUENTE}	-	0	35	-	mA
V_{OUT}	Seco	0	300	0-29.32	%
	Húmedo	301	700	29.33.1-68.42	%
	Con agua	701	950	68.43-100	%

Tabla 2.2 Características eléctricas de sensor [50]

3. Radio Xbee S2

La placa viene con un socket para colocar la radio denominado “Bee series socket” (ver Fig. 1.1). Es un elemento independiente y de gran importancia para el proyecto que vamos a diseñar. Razón por la cual, más adelante, explicaremos las características más importantes sobre su configuración y los pasos a seguir para cada una, según la función que desempeñe cada nodo.

Dentro de los productos de esta familia ofrecida por Digi, tenemos varios modelos que se guían bajo el protocolo Zigbee. Estos modelos son denominados Xbee S2, Xbee-PRO (S2) y Xbee-PRO (S2B). Sin embargo, difieren entre ellos por el alcance, por la potencia de consumo y el costo [51]. Para este diseño, nosotros elegiremos el módulo Xbee Zb o S2, ya que cubre toda el área con pocos dispositivos y son fáciles de usar, además de ser menos costosos que los otros. Trabajan en la banda de 2.4 GHz y está permitido en todas partes del mundo. Aquí encontramos nodos que son configurados como coordinador, enrutadores y/o dispositivos finales (end device).

3.1 Conexión entre Xbee S2 y Atmega 328P

En la Fig. 3.1 se muestran las conexiones que necesita como mínimo para ser utilizado. El módulo Xbee S2 requiere una tensión (Vcc) entre 2.8 y 3.4 V, conexión a tierra y a través de DIN Y DOUT se transmiten los datos por la interfaz UART con el Atmega328P o a través de un puerto serial con conversor de voltaje. El funcionamiento

es así, por DIN (pin 3) entran los paquetes de datos del microcontrolador en serie al módulo Xbee. Estos paquetes son almacenados en un buffer de entrada hasta que sean procesados. Cuando los paquetes son demasiados, la señal de control de flujo CTS (pin 12) evita el desbordamiento de buffer diciéndole al microcontrolador que deje de enviar. Por otro lado, cuando el módulo va a enviar información al microcontrolador, lo hace a través de DOUT (pin 2), sin embargo, previo a esto, los paquetes son movidos al buffer de salida, el cual si está muy lleno corre el riesgo de descartar ciertos paquetes. Por ello, hace uso de una señal de control de flujo, RTS (pin 16), el cual controla que siempre haya espacio para la transmisión.

A su vez, también se muestran las características eléctricas de este dispositivo en la Tabla 3.1, las mismas que se deben tener en cuenta para conocer el consumo de corriente cuando el nodo se encuentra activo o en modo Sleep.

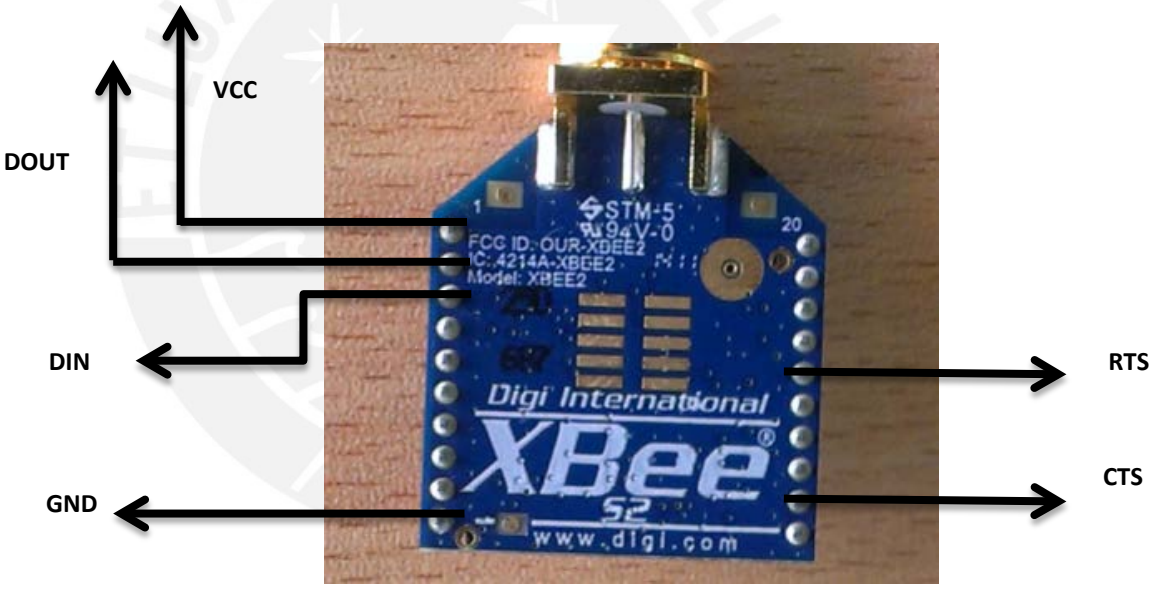


Fig. 3.1 Xbee S2 con conector RP-SMA y antena dipolo de 5 dBi

ESPECIFICACIONES	XBEE ZB
Performance	
Alcance Indoor	d 40m•
Alcance Outdoor	d120m•
Potencia de Tx en la salida	2• m•W • (• +•3 d• B•m) • , • • b• o• h• 1•. • 2 • 5m• W •(•+• 1• dB • m•)• ,• • b• h•
Velocidad de datos RF	2• 5•0• , •0• b•
Transferencia de datos	d3• 5• 0•0• b•
Velocidad Interfaz Serial	1• 2• 0• 0• b•1• • M• (• no•s•t • a• n• d• a• r •d• • b•a•u• d• s• u• p•p• o• b•
Sensibilidad de receptor	9• 6• • dB• m• ,• • b• o• o• s•t• h• •9• 5• • dB• m• ,• • b• o• o• s• t• h•
Requisito de Potencia	
Vcc	2•. 4• •3•. 6•
Corriente de operación (Tx, Pout max)	4• 0 • mA • •(•@• 3• . •3• • V•, • • b• o• h• 3• 5 • mA • •(•@• 3• . •3• • V•, • • b• o• h•
Corriente de operación (Rx)	4• 0 • mA • •(•@• 3• . •3• • V•, • • b• o• h• 3• 8 • mA • •(•@• 3• . •3• • V•, • • b• o• h•
Corriente de reposo	1• 5 •A•
Corriente Power-down	<• •1• • u•A•
General	
Banda de frecuencia de operación	I• S •M• •2 •
Dimensiones	0•. • 9•6 0 • • x• 1• . • 0• 8 7 • • (• 2
Temperatura de operación	•4• 0 •t • o• •85• º • •C
Opciones de antenas	I• n t •e•g• r• a•t •e d• •Wh • i • p, • •C• • C• o• n• n• e
Red y Seguridad	
Topologías	P• u•n•t •o• •a• • p• u• n• t• o•,• • p•u•n t • o • a•
Número de canales	1• 6 • s• e• c u•e• n c• i •a• s• •d• e•
Canales	1• 1• •
Opciones de direccionamiento	P• A•N • I • D • a• n• d• •A•d• d• r•e s• a• n•d• • E•n•d•p o• i• n• t•

Tabla 3.1 C• a• r •a• c• t•e• r•í • s• t • i •c • as•

3.2 Descripción de Pines

En este apartado se mencionan los pines que tiene este dispositivo y una pequeña descripción de cada uno. Ver Tabla 3.2. Esto es importante, ya que al momento de configurarlo mediante el software X-CTU debemos conocer con que puerto estamos trabajando, o cual es el pin de sueño o por cual pin se puede resetear la configuración, entre otras aplicaciones.

N° PIN	NOMBRE	DESCRIPCIÓN
1	VCC	Fuente de Energía
2	DOUT	Salida UART
3	DIN / CONFIG	Entrada UART
4	DIO12	Digital I/O 12
5	RESET	• •
6	RSSI PWM / DIO10	Indicador de fuerza de Señal RX - Digital I/O 10
7	DIO11	Digital I/O 11
8	RESERVADO	No conectado
9	DTR / SLEEP_RQ / DIO8	Control de pin sleep / Digital I/O 8
10	GND	Tierra
11	DIO4	Digital I/O 4
12	CTS / DIO7	Control “Listo para enviar” - Digital I/O 7. CTS se habilita como salida.
13	ON / SLEEP	Indicador de estado de módulo - Digital I/O 9
14	VREF	No usado en este módulo. Para compatibilidad con otros Xbee, se recomienda conectar este pin a una tensión de referencia, si se desea muestreo analógico. Caso contrario, conectar a GND.
15	ASOCIADO / DIO5	Indicador asociado - Digital I/O 5
16	RTS / DIO6	Control “solicitud para envío” - Digital I/O 6. RTS se habilita como entrada.
17	AD3 / DIO3	Entrada analógica 3 - Digital I/O 3
18	AD2 / DIO2	Entrada analógica 2 - Digital I/O 2
19	AD1 / DIO1	Entrada analógica 1 - Digital I/O 1
20	AD0 / DIO 0	Entrada analógica 0 - Digital I/O 0 – Botón de puesta en servicio

Tabla 3.2 Descripción de pines del módulo Xbee S2 [52]

4. Xbee Explorador Regulado

Este elemento se utilizará en el diseño para regular la tensión a 3.3V. Acondiciona la señal y tiene indicadores básicos de alimentación, RSSI, DIN y DOUT. Hace posible insertarlo en un protoboard a través de accesorios llamados “pin header” que se sueldan a la placa y tienen la separación exacta de los pines de este. Esta placa puede ser conectada a una tensión máxima de 16V y una corriente de 150mA. (Fig. 4.1)

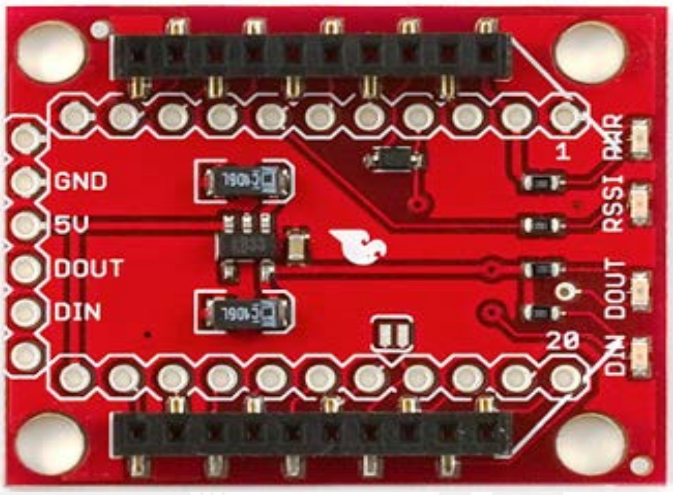


Fig. 4.1 Xbee Explorador Regulado [53]

5. Electroválvula

La válvula de control que usaremos para el presente diseño es de la empresa Orbit. Se eligió una de tapa roscada y fácil de desmontar sin necesidad de usar alguna herramienta, lo cual nos facilita su instalación. Como se puede observar en la Fig. 5.1, está construida en PVC, lo cual nos da una mayor resistencia a la corrosión y a los rayos UV. Incluye ciertos privilegios como una malla de filtración que se limpia con el paso del flujo de agua, y además, cuenta con un regulador de caudal, el cual se ajusta de manera manual. A la hora de darle mantenimiento se puede extraer el selenoide sin dañar otras piezas, lo cual nos ofrece otra ventaja y un mayor tiempo de vida para este actuador. La válvula se acciona por un solenoide encapsulado de 120 VAC, con una corriente de arranque de 0,43 A y una corriente de mantenimiento de 0,25 A. Ver especificaciones técnicas en la Tabla 5.1

Fig. 5.1 Electroválvula Serie PRO 100 [54]

ESPECIFICACIONES TÉCNICAS	
Dimensiones	
Altura	10,16 cm ó 4”
Ancho	7,62 cm ó 3”
Longitud	13,34 cm ó 5-1/4”
Características eléctricas	
Caudal	2,84 – 132,5 L/M ó 0,17 – 7,95 m^3/h ó 0.75 – 35 GPM
Índice de presión	1,38 – 10,34 bares ó 137,9 – 1034 kPa ó 20 – 150 PSI
Caudal	2,84 – 132,5 L/M
Solenoide	120 VAC @50/60 Hz
Corriente de arranque	0,43 A
Corriente de mantenimiento	0,25 A

Tabla 5.1 Características de Electroválvula Orbit [54]

6. Software X-CTU

Para configurar las motas que serán ubicadas estratégicamente en la plantación de piñón blanco hacemos uso de un software libre, llamado X-CTU [55], el cual es bastante sencillo de manipular por el usuario. Está diseñado para ser usado en equipos que ejecutan Windows (a partir del 98), el cual tenemos instalado (Fig. 6.1).

Una vez descargado este software, procedemos a descargar los drivers FTDI [56] para que la computadora reconozca el puerto donde vaya conectado el módulo Xbee S2. Este programa cuenta con cuatro pestañas, en la que cada una será descrita brevemente para una mejor orientación respecto al manejo del programa.

- PC Settings

  Aquí se elige el puerto COM al que está conectado nuestro módulo a configurar. Una vez seleccionado, se hace un click en "Test/Query", para verificar la correcta conexión y reconocimiento del módulo. Cuando esto sucede, aparece una ventana que indica el estado de la comunicación, el número serie (el cual viene de fábrica), la versión y el tipo de módulo. Por otra parte, esta ventana también nos permite configurar opciones más generales para el modo AT o habilitar el modo API. Lo demás por lo general se queda igual por defecto.

- Range Test

  Esta sección nos muestra el rango de cobertura. La opción de "Loop Back" nos permite enviar datos siempre y a través de la opción "RSSI" podemos observar la calidad de la señal. En "Range Test" observamos de toda la información enviada por el módulo Xbee, que porcentaje llega correctamente y cual no.

- Terminal

  Por esta pestaña podemos enviar y recibir datos en hexadecimal y en ASCII. Es la simulación que realizamos de la comunicación entre radios. Se puede enviar bit por bit o un paquete completo. Para enviar un paquete de data debemos hacer click en la opción "Assemble Packet" y escribir en formato ASCII o en Hexadecimal.

➢ Modem Configuration

En esta pestaña realizamos la configuración más específica. No debemos olvidar, antes que nada, actualizar la versión del software. Luego de ello, con los botones “read” y “write”, leemos lo que hay en cada variable y escribimos en ella. Lo más importante, para configurar, es la dirección de la PAN (ID PAN), la dirección destino (DL y DH), la notificación de unión al a red (JN), el modo sleep para ahorrar energía. Previo a esto, elegimos el tipo y la función del módulo que estamos configurando y descargamos las versiones más recientes de este software.

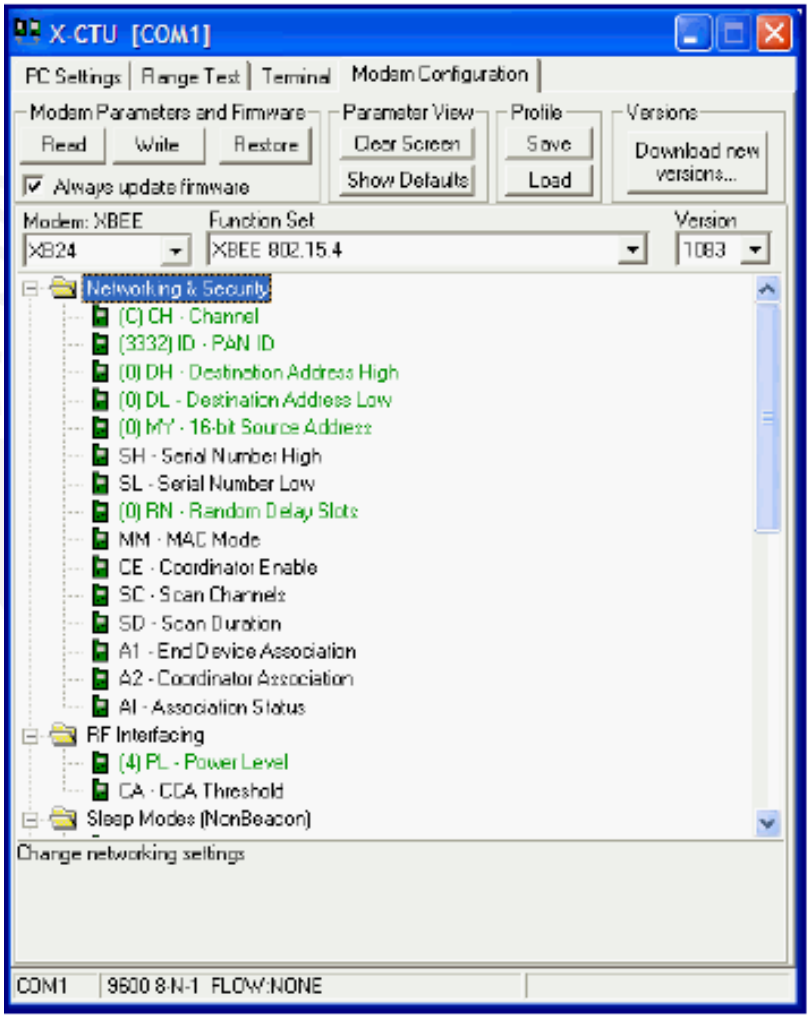


Fig. 6.1 Interfaz del programa X-CTU

7. IDE Arduino

El entorno de desarrollo Arduino (Fig. 7.1) nos permite escribir el código fuente que será grabado en la memoria del microcontrolador, ya que es compatible con la placa Seeeduino, que es la estamos utilizando. Este archivo se guarda con la extensión ".Ino". Nos provee de una serie de herramientas que nos permite, entre otras cosas, elegir la placa en la que vamos a quemar el programa fuente. Esta placa incluye la velocidad del CPU y la velocidad de trasmisión. Por ejemplo, en nuestro caso, seleccionamos "Arduino Fio", ya que trabaja con el mismo microcontrolador que nuestra placa Seeeduino, es decir, Atmega 328 con 8MHz y cuenta con auto-reset.

Cuenta con una interfaz bastante sencilla y también contiene librerías que nos dan una ayuda a la hora de programar. Si no encontramos alguna en particular, éstas se pueden descargar de otras fuentes y luego instalarlas y descomprimirlas en el folder denominado "libraries". Por ejemplo, para instalar la librería "Xbee", los archivos deben estar en el siguiente enlace Arduino/libraries/Xbee.

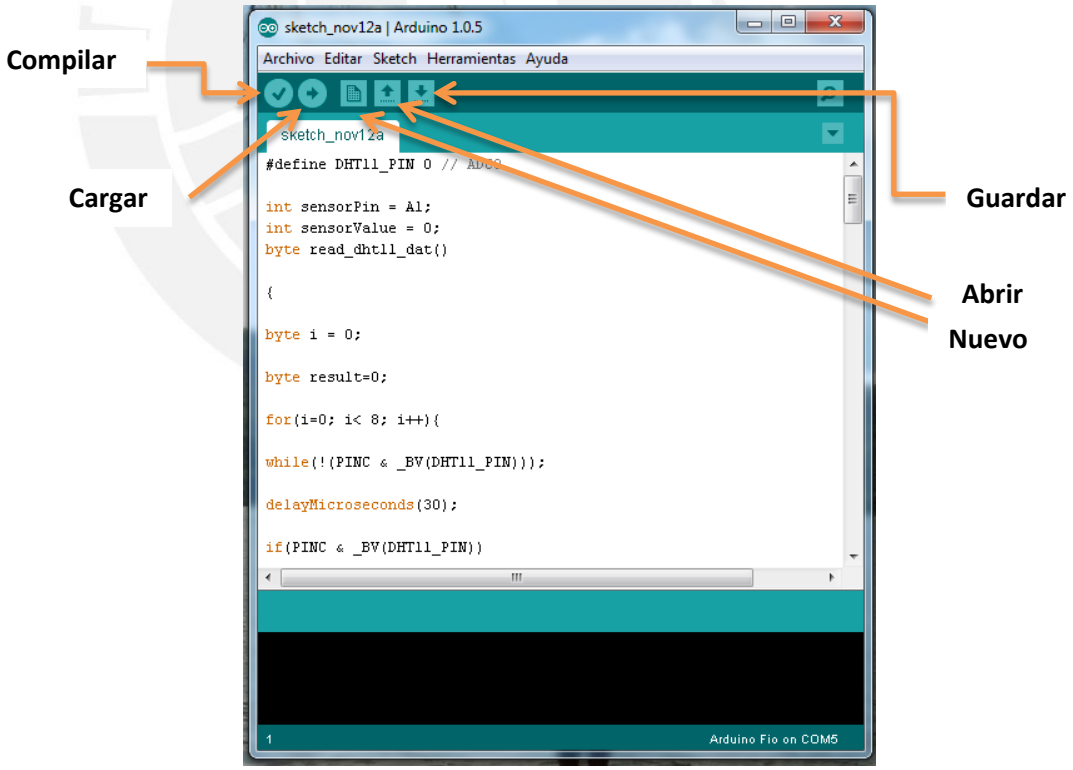


Fig. 7.1 Interfaz de IDE Arduino

ANEXO F

CÓDIGO FUENTE PARA SENSOR DE TEMPERATURA AMBIENTAL LM35

1. **Lectura de entrada analógica proveniente de sensor LM35 desde Xbee remoto (nodo Router/sensor) hasta nodo Coordinador que muestra data en computador.**

```
float temp;

void setup() {
      Serial.begin(9600);             // Inicia comunicación serial
}

void loop() {
      if (Serial.available() >= 21) {  // Devuelve el número de bytes a ser leídos
            if (Serial.read() == 0x7E) {    // Verifica si la trama es API
                  for (int i = 1; i < 19; i++) {  // Elimina los 18 primeros bytes
                        byte discardByte = Serial.read();
                   }
                   int analogMSB = Serial.read();      // lee parte alta de dato
                   int analogLSB = Serial.read();      // lee parte baja de dato
                   int analogReading = analogLSB + (analogMSB * 256)
                  // Juntamos ambos bytes en un solo número decimal

                   temp = [ 1.2 * analogReading * 100.0 ] / 1023;
                  // convierte lo que lee en temperatura

                  if (temp < 20.00) {
                        Serial.print (temp);
                        Serial.println (" grados C");
                        Serial.println ("Temperatura baja");
                  }
                  else if (temp <= 35.00) {
                        Serial.print (temp);
                        Serial.println (" grados C");
                        Serial.println ("Temperatura correcta");
                  }
                   else{
                        Serial.print (temp);
                        Serial.println (" grados C");
                        Serial.println("Temperatura alta");
                  }
            }
      }
}
```

2. Lectura de la trama API con data de temperatura ambiental

El nodo Router/sensor captura un dato del sensor de temperatura LM35, y lo transmite automáticamente al nodo Coordinador por medio de una trama API. Dicha trama es recibida por el XBee configurado como nodo Coordinador y este la transmite al computador por el puerto serie. La trama recibida es la siguiente:

7E 00 12 92 00 13 A2 00 40 76 91 4E 00 10 01 01 00 00 01 02 2F DF

Valor (HEX) **Descripción**

7E ⇨ Inicio de las tramas API
00 12 ⇨ Longitud de la trama
92 ⇨ Tipo de trama, en este caso, indica recepción de data.
00 13 A2 00 ⇨ Parte alta de la dirección origen (SH), nodo Router que envía.
40 76 91 4E ⇨ Parte baja de la dirección origen (SL), nodo Router que envía.
00 10 ⇨ Dirección de red dinámica del nodo Router que envía.
01 ⇨ Acuse de recepción que debe enviar el Coordinador al Router.
01 ⇨ Número de muestras solicitadas (siempre setea 1).
00 00 ⇨ Entrada digital configurada. En este caso no usamos línea digital.
01 ⇨ Entrada analógica configurada. En este caso usamos D0.
02 2F ⇨ Valor en decimal 559, es el valor de la entrada analógica D0. Como el voltaje de referencia en los módulos Xbee S2 es interno y corresponde a 1.2V y el módulo ADC es de 10 bits, lo cual nos da 1024 valores; entonces podemos decir que el voltaje en esa entrada es de:

(1.2V / 1023) x 559 = 0.65 V, los cuales se pueden medir en el pin 20 (D0) del Xbee. Este valor equivale a 15 °C según fórmula de conversión explicada en capítulo 3.

DF ⇨ Ckecksum o también conocido como algoritmo de verificación de la información recibida completa. Consiste en la suma de todos los bytes, desde el tipo de trama. Luego nos quedamos con el byte menos significativo del resultado, el cual se le resta a FF. Sería como lo que sigue:

92 + 00 + 13 + A2 + 00 + 40 + 76 + 91 + 4E + 00 + 10 + 01 + 01 + 00 + 00 + 01 + 02 + 2F = 320. Solo me quedo con 20, entonces:

Checksum = FF – 20 = DF

ANEXO G

CÓDIGO FUENTE PARA SENSOR DE HUMEDAD DE SUELO SEN92355P

1. **Lectura de entrada analógica proveniente de sensor SEN92355P desde Xbee remoto (nodo Router/sensor) hasta nodo Coordinador que muestra data en PC.**

```
float tens;
float porcent;

void setup() {
      Serial.begin(9600);             // Inicia comunicación serial
};

void loop() {
      if (Serial.available() >=21) {  // Devuelve el número de bytes a ser leídos
            if (Serial.read() == 0x7E) {    // Verifica si la trama es API
                  for (int i = 1; i < 19; i++) {  // Elimina los 18 primeros bytes
                        byte discardByte = Serial.read();
                  }
                  int analogMSB = Serial.read();       // lee parte alta de dato
                  int analogLSB = Serial.read();       // lee parte baja de dato
                  int analogReading = analogLSB + (analogMSB * 256)
                 // Juntamos ambos bytes en un solo número decimal

                   tens = analogReading / 1023.0 * 1.23; // convierte lo que lee en voltaje
                   porcent= analogReading/1023.0 * 100;

                  if (porcent <= 24) {
                        Serial.print(porcent);
                        Serial.println(" %");
                        Serial.println ("Suelo seco");
                        Serial.println("Activar riego");
                  }
                  else if (porcent <= 49) {
                        Serial.print(porcent);
                        Serial.println(" %");
                        Serial.println ("Suelo Humedo");
                        Serial.println("Desactivar Riego");
                  }
                   else{
                        Serial.print(porcent);
                        Serial.println(" %");
                        Serial.println ("Suelo con Agua");
                        Serial.println("No requiere riego");
                  }
```

```
        }
    }
}
```

2. Lectura de la trama API con data de humedad de suelo

Una vez que el nodo Router/sensor captura un dato que proviene del sensor de Humedad de suelo SEN92355P lo transmite automáticamente al nodo Coordinador por medio de una trama API. Dicha trama es recibida por el XBee configurado como nodo Coordinador y este la transmite al computador por el puerto serie. La trama recibida es la siguiente:

7E 00 12 92 00 13 A2 00 40 76 91 4E 00 10 01 01 00 00 02 01 2C E2

Valor (HEX)	**Descripción**
7E	⇨ Inicio de las tramas API
00 12	⇨ Longitud de la trama
92	⇨ Tipo de trama, en este caso, indica recepción de data.
00 13 A2 00	⇨ Parte alta de la dirección origen (SH), nodo Router que envía.
40 76 91 4E	⇨ Parte baja de la dirección origen (SL), nodo Router que envía.
00 10	⇨ Dirección de red dinámica del nodo Router que envía.
01	⇨ Acuse de recepción que debe enviar el Coordinador al Router.
01	⇨ Número de muestras solicitadas (siempre setea 1).
00 00	⇨ Entrada digital configurada. En este caso no usamos línea digital.
02	⇨ Entrada analógica configurada. En este caso usamos D1.
01 2C	⇨ Valor en decimal 300, es el valor de la entrada analógica D1. Como el voltaje de referencia en los módulos Xbee S2 es interno y corresponde a 1.2V y el módulo ADC es de 10 bits, lo cual nos da 1024 valores; entonces podemos decir que el voltaje en esa entrada es de:

(1,2V / 1023) x 300 = 0.351 V, los cuales se pueden medir en el pin 19 (D1) del Xbee.

E2 ⇨ Ckecksum o también conocido como algoritmo de verificación de la información recibida completa. Consiste en la suma de todos los bytes, desde el tipo de trama. Luego nos quedamos con el byte menos significativo del resultado, el cual se le resta a FF. Sería como lo que sigue:

92 + 00 + 13 + A2 + 00 + 40 + 76 + 91 + 4E + 00 + 10 + 01 + 01 + 00 + 00 + 02 + 01 + 2C = 31D. Solo me quedo con 1D, entonces:

Checksum = FF – 1D = E2

ANEXO H

CÓDIGO FUENTE PARA ELECTROVÁLVULA

1. **Escritura de salida digital proveniente para activar/desactivar electroválvula desde nodo Coordinador hasta Xbee remoto (nodo Router/actuador)**

```
int led = 9;

void setup() {
      pinMode (led, OUTPUT);   // Configura el pin como salida
      Serial.begin(9600);            // Inicia comunicación serial
}


void loop() {

     digitalWrite(led, HIGH);   // Prendemos led o ponemos en alta la salida
     setRemoteState(0x5);
     Serial.println("Se mandó señal en alta '1' ");
     Serial.println("Se activa electroválvula");
     delay(5000);                    // Esperamos por 5 segundos

     digitalWrite(led, LOW);    // Apagamos led o ponemos en baja la salida
     setRemoteState(0x4);
     Serial.println("Se mandó señal en baja '0' ");
      Serial.println("Se desactiva electroválvula");
     delay(5000);                    // Esperamos por 5 segundos
}

void setRemoteState(char value) {

   Serial.write(0x7E);              // Inicio de una trama API
   Serial.write((byte)0x0);      // Parte alta de la longitud de trama
   Serial.write(0x10);              // Parte baja de la longitud de trama
   Serial.write(0x17);              // Tipo de trama
   Serial.write((byte)0x0);      // Sin aviso de recepción
   Serial.write((byte)0x0);      // Dirección de Xbee remoto, hacemos broadcast
   Serial.write((byte)0x0);
   Serial.write((byte)0x0);
   Serial.write((byte)0x0);
   Serial.write((byte)0x0);
   Serial.write((byte)0x0);
   Serial.write(0xFF);
   Serial.write(0xFF);
```

```
Serial.write(0xFF);          // Dirección dinámica de Xbee remoto cuando no se
                                conoce
Serial.write(0xFE);
Serial.write(0x02);          // Aplica cambios configurados


Serial.write('P');           // Caracter en Ascii, se refiere al pin de salida
Serial.write('0');
Serial.write(value);         // Si la salida se pone en alta será 5, sino 4

long sum = 0x17 +0xFF + 0xFF + 0XFF + 0XFE + 0x02 + 'P' + '0' + value;
Serial.write(0xFF – (sum &        0xFF));        // Se calcula checksum
```

}

2. Escritura de la trama API para Activar/Desactivar Electroválvula

Cuando el nodo Coordinador quiere cambiar el estado de una salida del nodo Router/Actuador, éste envía una trama API para, por ejemplo, poner en alto esta salida y de esta forma activar la electroválvula según los parámetros de entrada y el control que hagamos en Labview. Dado esto, la trama que se envía en estos casos tiene la siguiente estructura:

7E 00 10 17 00 00 00 00 00 00 00 FF FF FF FE 02 50 30 05 66

Valor (HEX)	**Descripción**
7E	➪ Inicio de las tramas API
00 10	➪ Longitud de la trama
17	➪ Tipo de trama, en este caso, indica recepción de data
00	➪ Sin aviso de recepción (ACK)
00 00 00 00	➪ Parte alta de la dirección destino (DH)
00 00 FF FF	➪ Parte baja de la dirección destino (DL)
FF FE	➪ Dirección de red dinámica broadcast
02	➪ Para que se apliquen los cambios
50 30	➪ Código ASCII del parámetro que vamos a cambiar 'P' y '0'
05	➪ Comando a enviar, salida en alto.

➪ Ckecksum o también conocido como algoritmo de verificación de la información recibida completa. Consiste en la suma de todos los bytes, desde el tipo de trama. Luego nos quedamos con el byte menos significativo del resultado, el cual se le resta a FF. Sería como lo que sigue:

17 + 00 + 00 + 00 + 00 + 00 + 00 + 00 + FF + FF + FF + FE + 02 + 50 + 30 + 05 = 499.

Solo me quedo con 99, entonces:
Checksum = FF – 99 = 66