

Sistema embebido para la automatización de la transmisión de datos recolectados por estaciones de observación en campo

Gabriel Agustín García y Emiliano Pedro López

Centro de Estudios Hidro-Ambientales, Facultad de Ingeniera y Ciencias Hídricas (UNL). Ciudad Universitaria, CC217, Paraje el Pozo (3000) Santa F. Argentina. Tel: 54342-4575233 int 190; {gabiagus@gmail.com elopez@fich.unl.edu.ar}

Resumen El Centro de Estudios Hidroambientales (CENEHA) posee actualmente una red de observación compuesta por estaciones semiautomáticas ubicadas en diferentes puntos de observación pertenecientes a la Cuenca del Arroyo Cululú. A través de las estaciones se monitorea niveles freáticos, temperatura, potencial matricial del suelo, entre otros. En el presente trabajo se incorpora a la red mencionada una estación básica Campbell Scientific y otra conformada por un datalogger Stevens Dotlogger junto a un sensor de humedad Hydra-Probe II. Los datos son utilizados por el Centro para la modelación hidrológica, optimización del uso de suelos, validación con otras técnicas de medición (teledetección), entre otros.

El componente principal del sistema, encargado de comunicarse con las estaciones de monitoreo, es un dispositivo embebido hogareño el cual ha sido modificado mediante un sistema operativo embebido basado en Linux con el fin de otorgarle la flexibilidad equivalente a la de una computadora de propósito general.

El sistema implementado permite la automatización de la captura, la transmisión y el almacenamiento de los datos obtenidos por las estaciones de medición en campo.

Palabras Claves: Sistema embebido, OpenWRT, estaciones de medición, librería orientada a objetos.

1. Introducción

El agua es el recurso que va a determinar la riqueza, el bienestar y la estabilidad de muchos países en este siglo. Hoy en día, más de mil millones de personas carecen de acceso al agua potable. La gestión de los recursos hídricos es una temática compleja que cruza las fronteras e involucra a múltiples actores, con necesidades y exigencias diversas. La competencia entre las necesidades y demandas es relevante para el acceso y la asignación del recurso [1].

En este contexto es importante enfatizar que la medición, el modelado y la evaluación de los recursos hídricos es crítico, ya que estas cuantificaciones determinarán si el recurso es escaso o no. Por lo tanto, el monitoreo y la modelación de

los procesos involucrados en el Balance Hídrico (BH) adquieren especial interés [1].

Para el estudio de estos fenómenos comúnmente se utilizan estaciones de monitoreo puntuales que incluyen un conjunto de sensores de alta frecuencia y precisión. Estos equipos comerciales son ampliamente utilizados a nivel mundial por centros de investigación, de información climática y organismos gubernamentales, entre otros. Para realizar un análisis global de los procesos usualmente se instalan una serie de estaciones de medición distribuidas geográficamente.

La ubicación de estos equipos dificulta el acceso a los datos registrados, lo que implica realizar visitas periódicas a los puntos de observación donde se encuentran instalados, llevando a cabo la descarga en forma manual siendo este modo de operación costoso (viajes, viáticos, etc.) y poco práctico.

Como solución a este inconveniente una alternativa usualmente brindada por los fabricantes de sistemas de monitoreo, consiste en incorporar a las estaciones básicas módulos de comunicación inalámbricos (satelital, GSM/3G) que permiten extraer los datos de campo y enviarlos en forma automática a una estación de recepción.

Es importante destacar que tanto las estaciones básicas como los módulos de comunicación presentan un costo elevado incrementándose aún más por su origen extranjero.

El presente sistema intenta aportar una solución novedosa para extraer los datos de las estaciones en campo y posteriormente transmitirlos hacia Internet, utilizando para tal fin herramientas y tecnologías de amplio uso y probada confiabilidad reduciendo considerablemente el costo de las alternativas existentes en el mercado.

1.1. Planteo del problema

Los equipos de monitoreo que se instalan en campo generalmente vienen preparados para soportar las inclemencias climáticas y cuentan con su propio suministro de energía a través de paneles solares. Estos sistemas son autónomos ya que una vez puestos en funcionamiento no requieren ningún tipo de intervención manual, hasta el momento en el que se extraen los datos almacenados.

Es común, que debido al financiamiento acotado en centros de investigación y/o entidades públicas se adquieran estaciones de monitoreo básicas, es decir, capacidades de almacenamiento limitadas sin módulos de transmisión automática, de esta manera es necesario realizar viajes periódicos para descargar la información almacenada evitando la pérdida de datos.

Este modo de operación es poco práctico, costoso y, además, no permite tener la certeza que la estación se encuentra operativa hasta la próxima descarga in situ.

Los equipos utilizados en este proyecto son por un lado el sistema Eddy Covariance fabricado por Campbell Scientific que permite calcular el flujo de calor latente, flujo de cantidad de movimiento y el flujo de otros escalares entre la atmósfera y la superficie de la tierra (capa lmite), para esto el equipo mide la velocidad del viento, temperatura y humedad [2]. Y por otro, una estación

conformada por un datalogger Stevens Dotlogger junto a un sensor de humedad Hydra-Probe II.

El proceso de monitoreo llevado a cabo por estas estaciones implica el uso de sensores electrónicos, dataloggers, enlaces de comunicación (cableados o inalámbricos) y software de soporte. La función del datalogger consiste en convertir y almacenar las señales eléctricas medidas por los sensores, permaneciendo en la memoria hasta ser descargados mediante una computadora [3].

Es importante destacar que estos equipos generan un gran caudal de información colmando la capacidad de la memoria en un lapso de tiempo. El fabricante brinda la posibilidad de aumentar la capacidad de almacenamiento del datalogger a costo considerable.

Tal como se mencionó con anterioridad los módulos existentes que posibilitan la transmisión de los datos capturados son de elevado costo, requiriendo además la adquisición de un software para tal fin. Esta solución además de ser costosa es poco flexible, ya que impide adaptar el software a las necesidades particulares, estando sujeto a la funcionalidad provista de origen.

Como solución a los problemas mencionados se desarrolló e implementó en este trabajo una alternativa que permitie suplantar la tecnología brindada por el fabricante a través del uso de dispositivos de red económicos adaptados para utilizar sistemas operativos embebidos basados en GNU/Linux, herramientas de administración de sistemas libres (Licencias GPL) y software desarrollado bajo el lenguaje de programación Python, dando como resultado una solución escalable, flexible, robusta y de bajo costo.

1.2. Estado del Arte

En los últimos años han tomado un gran impulso las tecnologías de comunicación inalámbricas como GSM o CDMA (tecnologías móviles de segunda generación) en redes de monitoreo ambientales con transmisión automática de la información hacia centros de almacenamiento y análisis distantes de los puntos de observación. Como es el caso del desarrollo de un sistema que a partir de la transmisión inalámbrica facilita la colección de la información capturada y el monitoreo remoto de los equipos en campo [4]. Por otro lado se puede mencionar la implementación de un sistema de monitoreo de calidad de agua en tiempo real utilizado en acuicultura[5].

Linux posee un amplio uso como sistema operativo embebido sobre diversas plataformas, tal es el caso de un sistema de navegación autónomo para un vehículo submarino no tripulado, realizando un análisis del entorno a través de diversos sensores [6]. En cuanto a sistemas de medición ambiental es común el uso de procesadores con arquitectura ARM (Advanced RISC Machine) para implementar redes inalámbricas en malla (WSN, del inglés Wireless sensor network) para monitorear variables atmosféricas [7].

Además, se han implementado sistemas de monitoreo y control basados en el mismo hardware que utilizan Linux embebido y software de código abierto [8]. Como así también sistemas de adquisición de datos remotos utilizando sistemas embebidos y tecnología 3G [9].

Existen proyectos que se enfocan en la comunicación con estaciones de monitoreo de uso doméstico (PyWWS5 ¹, del inglés Python software for USB Wireless Weather Stations), con el fin de reemplazar el software original basado en sistemas Windows por uno de código abierto, la limitante de este proyecto es que se encuentra acotado a estaciones de un fabricante específico ².

En contraste con los trabajos mencionados en donde se aborda la problemática del monitoreo ambiental en forma completa, es decir desde los sensores a la transmisión del dato, el desarrollo aquí expuesto pretende incorporar al equipo de monitoreo de Campbell Scientific un módulo de comunicación empleando tecnologías inalámbricas de amplio uso y Linux como sistema operativo embebido. El desarrollo aquí realizado amplía el sistema automático de captura implementado en el centro de investigación [10], incorporando la comunicación con nuevas estaciones.

Para este fin se programó una librería que permite unificar en forma transparente las funciones de captura, almacenamiento y transmisión independientemente de la marca del dispositivo.

2. Materiales y métodos

Las estaciones de medición presentan ciertas restricciones para el funcionamiento referidas a la escasa memoria de los dataloggers para almacenar mediciones, la dependencia del software comercializado con la estación para extraer los datos y, el aislamiento geográfico de los equipos.

Como solución a estos problemas se implementó un sistema que permite mayor flexibilidad para comunicarse con el datalogger suplantando de esta manera la alternativa brindada por el fabricante a un costo inferior.

El desarrollo del software de comunicación fue posible a partir del análisis realizado sobre el funcionamiento de los distintos dataloggers que poseen las estaciones.

La virtud principal de contar con software propio para la comunicación con el datalogger es la posibilidad de ejecutarlo sobre diversas plataformas de hardware (dispositivos embebidos), evitando de este modo el uso de una computadora de propósito general (PC, arquitectura x86) y sistemas operativos Windows para la ejecución del software de escritorio.

Una ventaja relacionada a la ejecución de estos programas sobre hardware que cuente con tecnología de comunicación inalámbrica, es la posibilidad de transmitir los datos, registrados en sitios remotos, hacia un servidor con el fin de facilitar su acceso.

El dispositivo seleccionado para realizar las tareas de comunicación con el datalogger es el Router/Acces Point (AP) Tp-Link MR3020. La mayoría de los AP hogareños, como Tp-Link, Edimax, Lynksys, D-Link, etc., permiten el

¹ <http://code.google.com/p/pywws/>

² Fine Offset Electronics Co.,LTD.

reemplazo del firmware³ por sistemas embebidos basados en Linux, con una amplia comunidad de usuarios activa que dan soporte para diversas aplicaciones.

Es importante destacar que el equipo seleccionado es utilizado como un prototipo, lo que no impide migrar el software a dispositivos con mayores prestaciones en caso que los requerimientos a futuro se incrementen o que su desempeño no sea el esperado. A continuación se presenta el esquema propuesto.

2.1. Esquema propuesto

Como se mencionó anteriormente, el software desarrollado encargado de la comunicación con el datalogger se ejecuta sobre el Sistema Operativo Embebido (SOE) instalado en el Router/AP. Este programa permite capturar la información almacenada para su posterior transmisión hacia un servidor a partir del uso de tecnología inalámbrica. En (Fig. 1) se puede observar el esquema de la solución implementado.

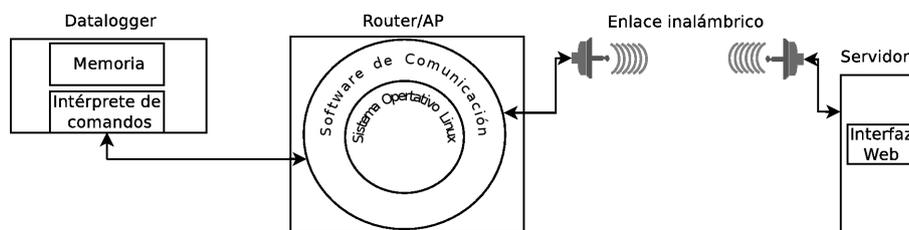


Figura 1. Esquema implementado

Para implementar el esquema propuesto se requiere de componentes para la comunicación con el datalogger (conectores y adaptadores), la alimentación (adaptadores de tensión) y la transmisión automática de los datos (módem) hacia un servidor. A continuación se describen los componentes del sistema.

2.2. Componentes del sistema

El sistema implementado en este proyecto está compuesto por tres elementos, un Nodo de monitoreo en Campo (NMC), un Servidor de Control Remoto (SCR) y un Generador Eléctrico Solar (GES).

El NMC se encuentra conectado a la estación de monitoreo y tiene como función principal interrogar al datalogger para obtener la información almacenada, realizar un procesamiento mínimo y transmitirlos hacia el servidor utilizando una conexión inalámbrica. Las tareas que puede realizar el nodo están organizadas en diferentes modos de operación.

³ Se denomina Firmware a una compilación estática del software

El servidor permite gestionar el funcionamiento del nodo y acceder a los datos colectados y, a la información sobre el estado de los equipos a través de una interfaz web.

Tanto el NMC como la estación de monitoreo son alimentados a partir de un Generador Eléctrico Solar como se observa en (Fig. 2).

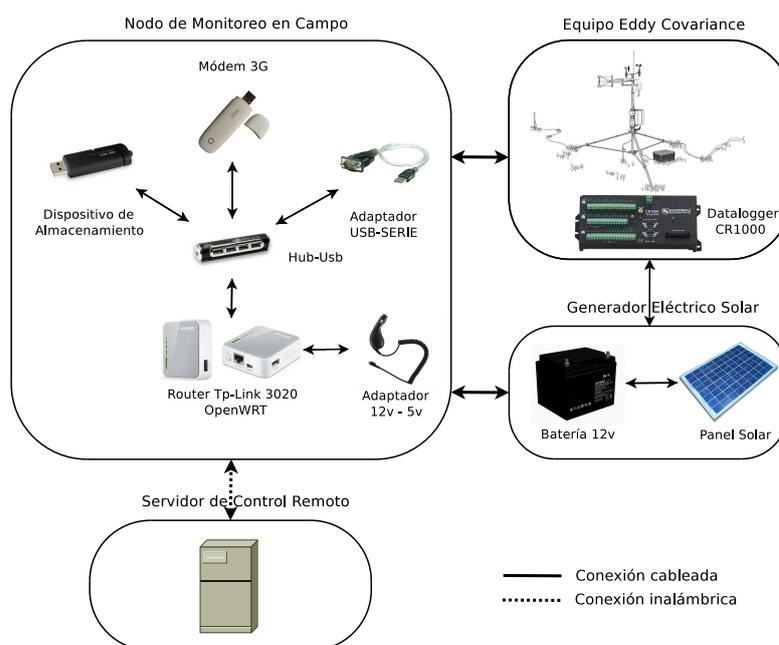


Figura 2. Interconexión entre componentes del sistema

Nodo de monitoreo en campo El componente principal del NMC es un AP y, debido a que éste cuenta con un único puerto USB se utiliza un multiplicador de puertos (hub USB) para conectarse a un dispositivo de almacenamiento USB, a un módem 3G-USB y al datalogger.

Las función principal del AP, tal como es comercializado de fábrica, es brindar acceso a Internet a través de una red inalámbrica o cableada, además de algunas prestaciones como firewall, redirección de puertos, servidor DHCP (del inglés, Dynamic Host Configuration Protocol), entre otros. Estos sistemas de propósito específico, denominados Sistemas Embebidos (SE), están diseñados para realizar un conjunto limitado de actividades [11].

Comúnmente los SE poseen recursos escasos (memoria, capacidad de almacenamiento y procesamiento acotada) y sin posibilidad de ampliación en comparación con computadoras de escritorio [12], sin embargo, para incrementar las funcionalidades del AP se implementó un Sistema Operativo Embebido (SOE)

basado en Linux obteniendo de esta manera la flexibilidad equivalente a la de una computadora de propósito general, ampliando además la capacidad de almacenamiento.

Si bien actualmente existe una gran variedad de SOE basados en Linux, para este proyecto se seleccionó OpenWRT⁴ por su larga trayectoria como SOE Linux para dispositivos de red, la gran cantidad de dispositivos soportados⁵ y, su amplia comunidad activa de desarrolladores y usuarios. Además, es distribuido bajo licencia GPL (del inglés, General Public License) y se encuentra en constante evolución.

Es importante destacar que el dispositivo AP, de funcionalidad originalmente acotada a tareas de conectividad, fue modificado obteniendo como resultado un sistema Linux de una gran flexibilidad en el que se instalaron y configuraron diferentes servicios y paquetes de software.

El NMC posee cuatro modos de operación definidos, dos corresponden a estados de captura y transmisión de datos (Tiempo real y Periódico), otro donde el equipo se encuentra en reposo (Espera) y el restante es un estado de ahorro de energía (Ahorro). En (Fig. 3) se presenta el diagrama de estados y a continuación se los describe.

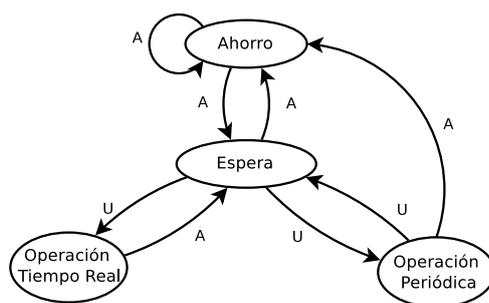


Figura 3. Modos de operación, transición automática (A) y manual (U)

Modo Espera Se encuentra en este estado cuando no fue establecido algún tipo de diálogo con el dispositivo datalogger. El ingreso a este modo de operación tiene tres alternativas, una establecida por el usuario mediante la interfaz web en forma remota, otra automática, luego de encontrarse en el estado de ahorro energético y por último, luego de haber finalizado el estado tiempo real.

Bajo este estado se realiza la re-inicialización y configuración de los parámetros necesarios para el funcionamiento global del NMC.

Se abandona este estado en forma manual, cuando el usuario activa alguno de los modos de captura (Tiempo real y Periódico) o bien, en forma automática cuando ingresa al modo Ahorro.

⁴ <https://openwrt.org>

⁵ <http://wiki.openwrt.org/toh/start>

Modo Ahorro Para determinar si es conveniente que el NMC realice las operaciones de captura de datos se efectúa una consulta al datalogger sobre el nivel de voltaje de la batería. Esta comprobación es útil para evitar entrar en un modo de operación que incremente el consumo y por ende deje la estación inoperativa. El objetivo de máxima es que la autonomía del GES, sea usada por la estación para registrar las variables hidro-ambientales.

La consulta para obtener el nivel de batería implica a su vez un consumo energético, por lo que estando en modo ahorro esta comprobación se realiza a intervalos de tiempo incrementales (potencia de 2) hasta obtener un valor que supere el umbral preestablecido, cambiando al modo de operación Espera y dando aviso al servidor.

Del mismo modo, el aviso del NMC al servidor sobre este estado se realiza por única vez al ingresar en el modo Ahorro, con el fin de evitar un consumo extra provocado por la comunicación inalámbrica. Tanto el ingreso como el egreso de este modo es comunicado mediante el envío de un correo electrónico al administrador del sistema.

Modo Operación Tiempo Real Este modo de operación se vincula con la ejecución del método de captura de datos Tiempo Real Durante un Intervalo (TRDI) que obtiene los valores crudos medidos por los sensores que componen la estación, en tiempo real. Una vez realizada la captura son enviados.

El ingreso a este estado se realiza a partir de la configuración manual, donde se establece el período durante el cual esta captura se ejecutará hasta finalizar, retornando al estado de Espera automáticamente. Previo al ingreso a este estado se corrobora que el nivel de la batería sea superior a un valor umbral establecido previamente, a fin de evitar la ejecución y agotar la carga.

Modo Operación Periódico Consiste en la captura y transmisión periódica de los datos procesados y almacenados en el datalogger. Para la adquisición de los datos se utiliza el método de captura Obtención de Datos Procesados (ODP).

Tanto el ingreso como el egreso desde y hacia el modo Espera es realizado por el usuario en forma manual, a través de la interfaz web. El ingreso a este estado se realiza siempre que la comprobación del nivel de carga sea el óptimo. La salida hacia el modo ahorro se da cuando el nivel de carga está por debajo del nivel umbral.

Servidor de control remoto Otro componente físico utilizado en este sistema es el Servidor de Control Remoto (SCR), que se comunica con el NMC a través de Internet utilizando una Red Privada Virtual (VPN del inglés, Virtual Private Network) con el fin de brindar una conexión segura utilizando encriptación de 128 bits (Fig. 4).

Las funciones principales de este equipo son recibir los datos enviados desde el NMC, brindar acceso a los datos colectados, obtener información sobre el estado de la estación, el GES y el NMC y, gestionar el funcionamiento del NMC.

Para brindar las funcionalidades mencionadas el servidor corre un sistema operativo Linux Debian stable.

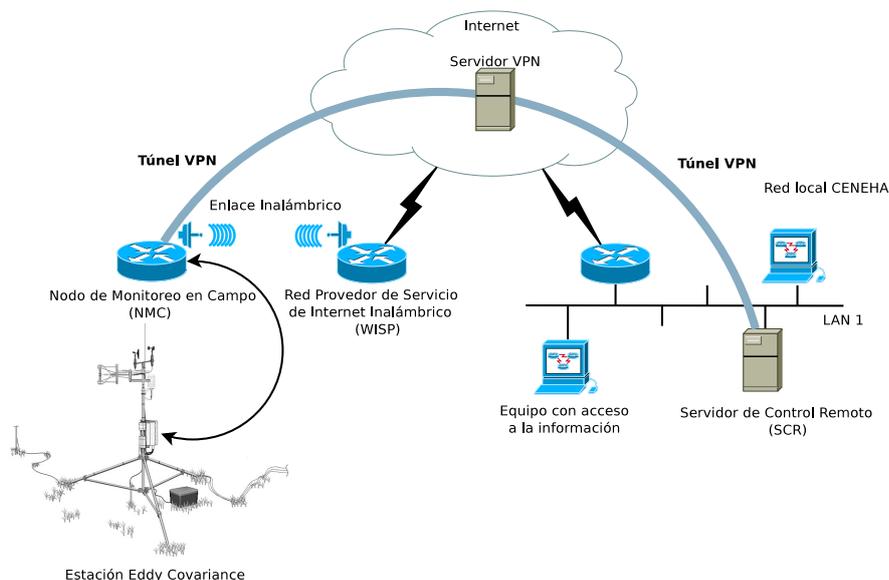


Figura 4. Diagrama de red físico, interconexión de los componentes

2.3. Funcionamiento global del sistema

El intercambio de información entre el NMC y el SCR es bidireccional, los datos referidos a la configuración del nodo en campo son descargados desde el servidor, y las mediciones y la información sobre el estado del nodo son enviadas hacia el servidor.

El ciclo de funcionamiento, detallado en (Fig. 5), se inicia cuando el NMC descarga el archivo de configuración desde el servidor (1), modificado por el usuario a partir de la interfaz web, luego en el nodo se analiza si existe una nueva configuración y en caso afirmativo se la aplica (2). La misma puede estar vinculada al modo Espera, es decir sin que implique un diálogo con la estación, o bien a la extracción de datos (3) ya sea en el modo Periódico o Tiempo Real.

Una vez concluida la descarga de datos del datalogger se realiza un procesamiento mínimo para descartar información innecesaria (4), dejando los archivos listos para su transmisión al servidor (5), donde finalmente se publican los datos colectados permitiendo al usuario accederlos (6).

Dentro de las acciones realizadas en este ciclo se pueden identificar las etapas de captura, transmisión y almacenamiento que son explicadas en detalle a continuación.

Captura El diálogo entre el nodo y la estación se realiza cuando se requiere extraer los datos almacenados en el datalogger o las mediciones crudas directamente de los sensores.

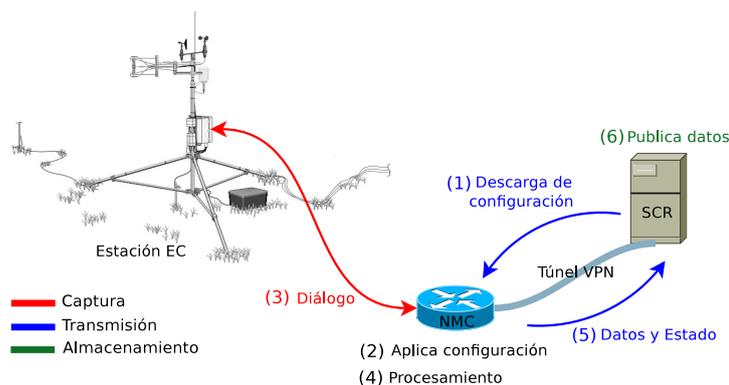


Figura 5. Diálogo entre los componentes del sistema

Los programas de captura fueron desarrollados en el lenguaje Python⁶, esta elección se basa en que es fundamental que el software utilizado para la comunicación sea completamente portable, es decir que sea independiente de la arquitectura de hardware sobre la que corre.

Otra ventaja de Python es la gran cantidad de librerías existentes, entre ellas cabe destacar las que brindan el acceso a dispositivos de hardware en forma transparente. Esta característica es de gran utilidad para este proyecto ya que se requiere la comunicación a través del puerto serie para comunicarse con el datalogger (la librería utilizada se denomina PySerial⁷).

El software programado fue modularizado para funcionar como una librería orientada a objetos, permitiendo incorporar nuevos dispositivos/sensores y, los parámetros y comandos de configuración necesarios para establecer una comunicación exitosa.

La librería contiene una clase desarrollada denominada Datalogger, que utiliza para su funcionamiento un archivo de configuración en donde se encuentra la lista de dispositivos con los que puede operar. En base a esta lista, la clase identifica el equipo y obtiene información relativa a la configuración de la comunicación y comandos necesarios para realizar el diálogo. Un esquema de esta descripción se puede observar en (Fig. 6).

La función principal de la clase Datalogger es establecer la comunicación con el datalogger, extraer los datos almacenados y obtener los valores medidos por los sensores.

Al crear una instancia de la clase Datalogger se establece el parámetro relacionado al tipo datalogger a partir del cual se obtiene la configuración de la comunicación y los comandos de extracción de datos soportados por el tipo de dispositivo.

⁶ <http://www.python.org/>

⁷ <http://pyserial.sourceforge.net/>

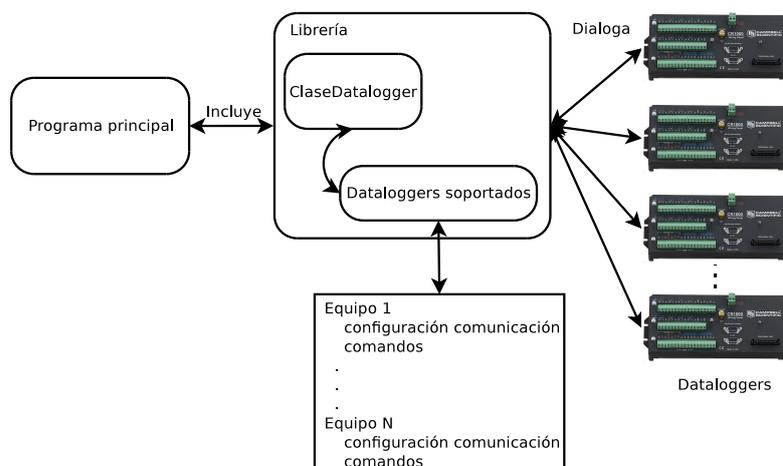


Figura 6. Esquema funcionamiento de la librería

Una vez creado el objeto Datalogger la clase permite iniciar la comunicación, ejecutar algún método de captura de datos y luego de obtenerlos, cerrar la comunicación.

Utilizando los distintos métodos de captura programados es posible obtener tanto valores calculados (datos procesados) por el datalogger como mediciones crudas (datos tiempo real). Estos métodos se describen a continuación.

Datos procesados El método Obtención de Datos Procesados (ODP) interroga al datalogger a intervalos de tiempo preestablecidos, obteniendo la información calculada a partir de los datos crudos por el software del fabricante implementado en el dispositivo.

Datos en Tiempo Real Los siguientes métodos obtienen los valores medidos directamente desde los sensores a través del datalogger:

Por única vez: este método retorna la consulta de los valores medidos por los sensores en el instante en que es ejecutado.

Durante un intervalo (TRDI): a diferencia del método anterior se realiza la consulta durante un intervalo de tiempo establecido, almacenando los resultados (datos crudos) en un archivo.

Transmisión La transmisión se realiza en dos sentidos desde el servidor al nodo y viceversa, en ambos casos quien comanda el proceso es el NMC.

En la transmisión con sentido servidor-nodo se descarga información referida a los modos de operación del NMC que pueden ser configurados por los usuarios.

Por otro lado, en la transmisión en sentido nodo-servidor se envían los archivos de datos creados en la etapa de captura y la información relativa al estado del NMC en caso que se haya pasado automáticamente al modo Ahorro.

Almacenamiento La información transmitida por el equipo en campo es recibida por el servidor y almacenada en un repositorio de archivos. Los archivos son alojados bajo una estructura de directorios, en donde cada punto de observación posee varios equipos de medición y a su vez, cada equipo posee dos directorios, uno para los datos y otro destinado para la configuración.

En el directorio 'Datos' se almacena la información obtenida en campo, subida al servidor en forma automática. Bajo el directorio 'Configuración' se encuentra el archivo conf.ini que contiene la información necesaria para el funcionamiento del NMC.

2.4. Acceso a los datos y configuración

El acceso a los datos en el servidor y a la configuración remota del NMC se realiza a partir de una interfaz web, que permite navegar por los puntos de observación que conforman la red de observación, la misma fue programada en los lenguajes PHP, AJAX y MySQL.

La misma permite al grupo de investigación acceder y descargar los datos, configurar el funcionamiento del NMC y analizar el funcionamiento de la estación de medición.

Se definieron dos tipos de usuarios denominados 'Espectador' y 'Administrador'. Las acciones que puede realizar el usuario tipo 'Espectador' consisten en el acceso al navegador de archivos de los diferentes puntos de observación de la red. Dentro de éste, es posible navegar por el repositorio en el que se encuentran almacenados los archivos, permitiendo visualizar el contenido o descargarlos.

El usuario de tipo 'Administrador' además de visualizar el repositorio, puede modificar la configuración del equipo y realizar un análisis de su funcionamiento. Las acciones referidas a la configuración son la modificación del modo de operación deseado (Tiempo Real, Periódica o Espera) y establecer los parámetros para su funcionamiento. En cuanto al análisis del funcionamiento, el administrador puede observar los valores medidos por las estaciones en tiempo real y corroborar la evolución del nivel de la batería durante su funcionamiento, permitiéndole analizar el desempeño.

3. Resultados

Para el análisis del funcionamiento del sistema se definieron dos tipos de pruebas una 'bajo condiciones controladas' y otra en 'situaciones reales de operación'. En el primer caso, las pruebas fueron realizadas en las instalaciones del CENEHA, donde, el NMC y las estaciones se encontraban ensambladas en laboratorio protegidas de las inclemencias climáticas como tormentas, temperaturas extremas, humedad, entre otras. En el segundo caso, se ensambló el prototipo en campo, en el campus de la Universidad Nacional del Litoral.

En primer instancia se analizó el estabilidad del router marca Tp-link, modelo MR3020 funcionando con OpenWrt como sistema operativo embebido. Sobre este sistema se evaluó el comportamiento de los programas desarrollados y las

diferentes tecnologías de acceso cableado e inalámbrico a Internet (802.11, GPRS, 3G).

Tanto en laboratorio como en campo se probaron de forma exhaustiva los modos de funcionamiento Espera, Operación Periódica y Operación Tiempo Real y el rendimiento para cada caso, evaluando el consumo de memoria y el uso del CPU.

Los métodos captura probados fueron Obtención Datos Procesados (ODP) y Tiempo Real Durante un Intervalo (TRDI).

Se evaluó, además el intercambio de información entre el NMC y el SCR, por un lado la transmisión de la información capturada hacia Internet y, por otro lado, se corroboró la descarga del archivo de configuración cada 2 minutos junto a la aplicación de la configuración descargada.

Estas pruebas permitieron detectar fallas e inconvenientes dando como resultado un sistema robusto y confiable.

4. Conclusiones

El sistema propuesto demostró ser una alternativa confiable y de bajo costo en contraste con las existentes en el mercado, dando como resultado un sistema flexible, robusto y escalable.

La flexibilidad esta dada en más de un aspecto, por un lado la portabilidad del código, lo que significa la posibilidad de utilizar diferentes dispositivos embebidos sin estar atado a un hardware específico. Por otro lado, haciendo referencia a la transmisión, el sistema es independiente de la tecnología de acceso a Internet y, en caso de no contar con esta posibilidad, se reduce considerablemente la frecuencia de descarga manual debido a la ampliación lograda en la capacidad de almacenamiento.

La robustez del nodo fue corroborada mediante pruebas intensivas en laboratorio y en campo durante un período prolongado de tiempo, en las cuales se corrigieron fallas y se implementaron mejoras dando como resultado un comportamiento estable.

La escalabilidad del sistema esta vinculada con la capacidad del nodo de incrementar la cantidad de equipos de medición con los que se comunica brindando las mismas funcionalidades. Esto se logra ya que el software fue diseñado como una librería orientada a objetos que permite acoplar fácilmente dispositivos, incorporando solamente los comandos necesarios para su comunicación.

Con la solución aquí planteada se replicó el comportamiento del software de fábrica añadiendo la automatización de la captura y transmisión de datos procesados y, además se incorporó una nueva funcionalidad, no presente originalmente, que consiste en la obtención de los valores crudos en tiempo real.

El acceso a estos valores crudos posibilita realizar cálculos en base a diferentes modelos teóricos independizándose de este modo del implementado por el fabricante.

Otra funcionalidad destacable respecto a la autonomía del sistema fue la optimización del consumo de energía. Esta característica fue posible a partir

del diálogo con el datalogger que permitió contar con el nivel de tensión de la batería, ingresando automáticamente en un modo de ahorro cuando se detecta un nivel bajo.

Para concluir es importante mencionar que es posible conocer el estado del equipo en campo en forma remota, permitiendo detectar una anomalía en el funcionamiento y evitando de este modo la pérdida masiva de datos.

Referencias

1. Brutsaert, G.: Hydrology, An Introduction. Cambridge Univ.Press, New York (2005). pp. 1 12
2. Campbell Scientific, Inc.: Open Path Eddy Covariance System Operator Manual CSAT3, LI-7500, and KH20. Tech. rep., (2006).
3. Campbell Scientific, Inc.: CR1000 Measurement and Control System. Tech. rep. (2011).
4. Kunrong, Z., Dayong, Y., Kui, L., Jian, Y., dec. 2010. An embedded environmental monitoring data collecting system. In: Information Science and Engineering (ICISE), 2010 2nd International Conference on. pp. 1 4.
5. Zhu, X., Li, D., He, D., Wang, J., Ma, D., Li, F., 2010. A remote wireless system for water quality online monitoring in intensive fish culture. computers and electronics in agriculture 71s (1), s3s9.
6. Thakur, S., Conrad, J., march 2007. An embedded linux based navigation system for an autonomous underwater vehicle. In: SoutheastCon, 2007. Proceedings. IEEE. pp. 237 242.
7. Gao, M., Zhang, F., Tian, J., oct. 2008. Environmental monitoring system with wireless mesh network based on embedded system. In: Embedded Computing, 2008. SEC 08. Fifth IEEE International Symposium on. pp. 174 179.
8. Li, X., Liu, J., oct. 2010. Design of embedded data acquisition and remote control system based on linux. In: Computer Application and System Modeling (IC-CASM), 2010 International Conference on. Vol. 3. pp. V3299 V3302.
9. Bin, Q., Xuanang, M., Junda, Z., Fang, L., jan. 2012. Design of remote data acquisition system based on 3g. In: Intelligent System Design and Engineering Application (ISDEA), 2012 Second International Conference on. pp. 1315 1318.
10. Lpez, E., 2012. Captura y transmisión automática de datos hidro-meteorológicos. Maestría en computación aplicada a la ciencia y la ingeniería, Facultad de Ingeniería y Ciencias Hídricas, UNL.
11. Raghavan, P., Lad, A., Neelakandan, S.: Embedded Linux System Design and Development. Auerbach Publications (2006). pp. 1 28
12. Hallinan, C.: Embedded Linux Primer: A Practical, Real-World Approach. Prentice Hall (2006). pp. 1 62