

Interfaz para la adquisición de datos en tiempo real basado en un datalogger de nivel de agua de efecto capacitivo

Antony García González, Licenciatura en Ingeniería, Yessica Sáez, PhD, Ignacio Chang, PhD
Universidad Tecnológica de Panamá, Panamá
{antony.garcía, yessica.saez, ignacio.chang}@utp.ac.pa

Abstract— Los registradores de datos o datalogger son dispositivos utilizados a nivel científico e industrial para la captura y almacenamiento de datos a partir de los muestreos de un sensor. Se utilizan para mediciones de variables ambientales como humedad, temperatura, pureza del aire, etc. En su mayoría se caracterizan por almacenar muestreos y generalmente no cuentan con la capacidad de entregar información en tiempo real, a menos que permanezcan conectados a una computadora. En este trabajo se ha diseñado una interfaz de comunicación entre un datalogger comercial para la medición de niveles de agua por efecto capacitivo y una placa electrónica de bajo costo basada en una combinación entre un microcontrolador y un microprocesador. Este diseño ha permitido extraer información en tiempo real del datalogger, utilizando las capacidades del microcontrolador para establecer comunicación a través del protocolo RS-232. El microprocesador se encarga de almacenar la información en una base de datos y le brinda al usuario la capacidad de transmitirla a través de Internet. Con esta interfaz, un dispositivo diseñado para guardar información durante cierto tiempo adquiere una mayor capacidad de almacenamiento a través de la expansión de memoria y hace posible la transmisión de datos muestreados de forma instantánea. Los resultados obtenidos en las pruebas de campo para este diseño fueron satisfactorios. El diseño propuesto es útil para aquellas aplicaciones en las cuales se cuenta con suministro de energía y una conexión estable a Internet para la transmisión de la información.

Keywords— Sensor de nivel de agua de efecto capacitivo, datalogger en tiempo real, sistemas embebidos, RS-232, comunicación serial

I. INTRODUCCIÓN

Los sensores capacitivos de nivel de agua son muy utilizados en experimentos a nivel científico. Este tipo de sensor normalmente utiliza un cable aislado con un material impermeable que posee constante dieléctrica conocida [1]. Este cable se introduce en el líquido cuyo nivel se desea medir. En el interior del cable se encuentran dos conductores separados, los cuales funcionan como las placas de un capacitor. Al lograr que uno de los conductores toque el agua, esta pasa a formar parte del capacitor, de tal forma que entre más alto sea el nivel del agua, más grande será la placa y mayor será la capacitancia [2]. Un circuito electrónico multivibrador (comúnmente se utiliza un LM555 CMOS) se encarga de transformar la capacitancia en frecuencia, de forma tal que la frecuencia de la salida del circuito será proporcional a la longitud del cable inmersa en el agua [3].

Por medio de un microcontrolador se lee la frecuencia de la salida del multivibrador y se transforma en capacitancia. El

microcontrolador almacena la información en una memoria no volátil (EEPROM) y permite extraer la información desde una computadora a través de un software diseñado para este propósito. Este tipo de *datalogger* se caracteriza por un bajo consumo de energía, permitiendo muestrear durante meses sin desgastar la batería interna del dispositivo.

Es muy común la utilización de estos sensores para la medición de aguas subterráneas, por el hecho de que el cable se puede introducir en tuberías o agujeros sin comprometer el circuito de recolección de datos. Éste se ubica en uno de los dos extremos del cable que funciona como un capacitor, por lo que es posible asegurarlo a la superficie sin tener que exponerlo al agua o insertarlo dentro de alguna cavidad.

Un modelo muy utilizado de este tipo de registradores es el medidor de nivel de agua de efecto capacitivo de la marca Odyssey (Capacitance Water Level Logger, Odyssey, Nueva Zelanda) [4], principalmente por su capacidad de almacenamiento de aproximadamente 32000 lecturas (64 KB), su precisión de 0.8mm y la larga duración de su batería. Con este *datalogger* es posible leer información de forma ininterrumpida hasta por 6 meses. Ha sido ampliamente utilizado a nivel científico para muestreos de niveles de agua en pozos y depósitos de agua subterráneos en general [5]. Sin embargo, al igual que otros sensores capacitivos similares, no es posible extraer la información en tiempo real, a menos que se mantenga conectado a una computadora de forma permanente, por medio de un cable con un convertidor de puerto DB9 (RS-232) a USB. Esto representa un inconveniente en las aplicaciones donde los *datalogger* se encuentren instalados a la intemperie.

La incapacidad de entregar información instantánea hace que este tipo de dispositivos solamente pueda ser utilizado en aplicaciones donde no se necesita la información al instante. Generalmente, es necesario esperar días, semanas o meses para extraer la información y ver el resultado del muestreo.

Una opción viable para extraer la información del sensor es hacerlo a través de placas electrónicas basadas en sistemas embebidos con la capacidad de comunicarse por medio de RS-232. Los dispositivos de la marca Arduino son una opción interesante para lograr este propósito. Es necesario decodificar la forma cómo el *datalogger* establece comunicación con la computadora a través del puerto serie. Luego se utiliza el MAX3232 [16], un circuito integrado para la adecuación de los niveles lógicos de voltaje (TTL) entre el sensor y el Arduino.

En el lado del microprocesador es posible almacenar mayores volúmenes de información en comparación con la memoria interna del *datalogger*. También es posible transmitir los datos muestreados en tiempo real, incluso a través de Internet.

El objetivo de este trabajo es diseñar una interfaz de comunicación entre un *datalogger* comercial utilizado en la medición de niveles de agua por efecto capacitivo y una placa electrónica de bajo costo basada en una combinación entre un microcontrolador y un microprocesador. El resto de este artículo está organizado de la siguiente manera. La sección II describe la composición y el funcionamiento del dispositivo para el cual se hará el diseño de la interfaz. La sección III muestra el diseño de dicha interfaz. La sección IV presenta las pruebas de funcionamiento. La sección V discute los resultados mientras que la sección VI presenta las conclusiones junto con importantes iniciativas de aplicaciones que abren oportunidades de investigación en el futuro.

II. DECODIFICACIÓN DEL DATALOGGER

A. Composición y funcionamiento del datalogger

Antes de proceder con el diseño de la interfaz de comunicación es necesario conocer el funcionamiento del dispositivo para el cual se hará el diseño.

Se considerará el caso del *datalogger* de la marca Odyssey para el registro de nivel de agua por efecto capacitivo (Odyssey Capacitance Water Level Logger) por ser el más ampliamente utilizado. Éste posee un cable con una cubierta de teflón con dos conductores en su interior (ver **¡Error! No se encuentra el origen de la referencia.**). Los dos conductores se encuentran aislados entre sí, de tal forma que al aplicarse una diferencia de potencial sobre ellos se forma un capacitor. En un extremo del cable se encuentra un contrapeso de bronce (ver Fig. 2) directamente conectado a uno de los dos conductores que forma el cable.

Fig. 1 Composición interna del cable del Odyssey Capacitance Water Level Sensor

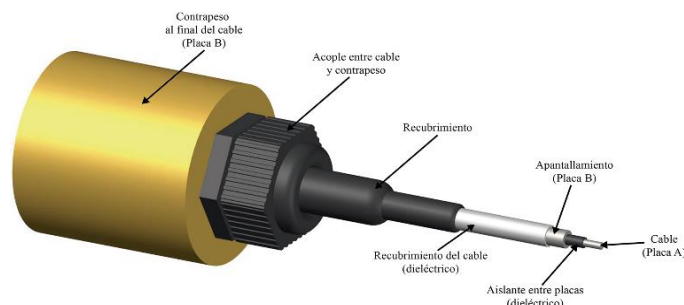


Fig. 2 Contrapeso en el extremo del cable del Odyssey Capacitance Water Level Sensor

El contrapeso debe hacer contacto con el agua para que el sensor pueda muestrear los datos. El agua entonces pasa a ser parte del capacitor, una de las placas. Este es el modo de

funcionamiento típico de los sensores de nivel de agua por capacitancia.

El circuito del *datalogger* contiene los componentes electrónicos necesarios para transformar la capacitancia a frecuencia y la frecuencia a un valor numérico que se almacena en una memoria EEPROM de 64 kb. A través de un puerto RS-232 es posible extraer la información desde una computadora, utilizando la aplicación distribuida por el fabricante.

El *datalogger* se conecta a la computadora a través de un convertidor de USB a RS-232 y con el adaptador distribuido por el fabricante. Con el software es posible extraer la información contenida en la memoria y programar los muestreos. El software también cuenta con un modo de calibración con el cual se puede llevar a cabo una prueba experimental para determinar la relación entre capacitancia y nivel de agua en la medida de longitud de preferencia.

Para calibrar el sensor, el *datalogger* entra en modo de Lectura en Tiempo Real (LTR), una opción que permite que los sensores entren en un modo de prueba. Este modo de funcionamiento provoca que los sensores envíen 64 lecturas desde el *datalogger* a la computadora, a intervalos de aproximadamente 5 segundos entre cada lectura. Esto es lo más cercano a un muestreo en tiempo real que se puede lograr con este dispositivo.

B. Decodificación de la comunicación RS-232

El modo LTR obliga al *datalogger* a enviar datos separados por intervalos cortos de tiempo. Es posible lograr que la misma información que se envía a la aplicación del sensor en una computadora pueda ser leída e interpretada por un microcontrolador.

Utilizando el software *Device Monitoring Studio*, en su versión de prueba, se logró observar el flujo de bytes entre el *datalogger* y la computadora a través del Puerto Serie mientras se inicia modo LTR. La TABLA I muestra los resultados obtenidos.

TABLA I
BYTES ENVIADOS Y RECIBIDOS POR LA COMPUTADORA AL ODYSSEY WATER LEVEL SENSOR VISTOS DESDE EL DEVICE MONITORING STUDIO Y COMPARADOS CON LOS VALORES DEL MODO LTR VISUALIZADOS EN LA APLICACIÓN DEL DATALOGGER

Device Monitoring Studio		Aplicación del fabricante
Bytes Enviados	Bytes Recibidos	Valor mostrado
0x01	-	-
-	0x48 0x69 0x21	-
0x54 0x0A	-	-
-	0x3F 0x07	1855
-	0x3F 0x07	1855
-	0x3F 0x07	1855
-	0xFE 0x09	2558
-	0xB7 0x09	2487
-	0x3D 0x07	1853
-	0x3D 0x07	1853
-	0xA3 0x0A	2713
-	0xB5 0x0A	2741
-	0xC3 0x0C	3267
-	0xC5 0x0C	3269

Digital Object Identifier: (to be inserted by LACCEI).
ISSN, ISBN: (to be inserted by LACCEI).

-	0x3A 0x07	1850
-	0x39 0x07	1849
-	0x6A 0x09	2410

Durante esta prueba se sumergió el sensor en agua durante breves instantes, lo cual causó que la capacitancia registrada se elevara por encima del nivel registrado en seco (aproximadamente 1855).

Los resultados mostrados en la TABLA I indican que sólo es necesario enviar los bytes 0x01 y 0x54 0x0A para que el *datalogger* entre en modo LTR y empiece a enviar datos a través del puerto RS-232. Para encontrar la correlación entre los bytes y los valores impresos en la aplicación del *datalogger*, se eliminaron los datos repetidos y se ordenaron de forma ascendente. También se convirtió los bytes hexadecimales a formato decimal, tal como se muestra en la TABLA II.

TABLA II
BYTES ENVIADOS Y RECIBIDOS POR LA COMPUTADORA AL ODYSSEY WATER LEVEL SENSOR VISTOS DESDE EL DEVICE MONITORING STUDIO Y COMPARADOS CON LOS VALORES DEL MODO LTR VISUALIZADOS EN LA APLICACIÓN DEL DATALOGGER

Bytes Recibidos	Valores Decimales	Resultados en la aplicación del fabricante
0x3A 0x07	58 7	1850
0x3D 0x07	61 7	1853
0x3F 0x07	63 7	1855
0x6A 0x09	106 9	2410
0xB7 0x09	183 9	2487
0xFE 0x09	254 9	2558
0xA3 0x0A	163 10	2713
0xB5 0x0A	181 10	2741
0xC3 0x0C	195 12	3267
0xC5 0x0C	197 12	3269

A simple vista se puede observar un patrón entre los valores decimales y los mostrados la aplicación en la computadora. Si se considera que los bytes recibidos obedecen la forma 0xAB 0xCD, se deduce que los bytes AB son mucho más susceptible a cambios que los bytes CD, tomando como referencia los valores vistos desde la aplicación del fabricante. Por otra parte, las variaciones en los bytes AB vistos en formato decimal coinciden con las variaciones de los datos visualizados en la aplicación del fabricante. Por ejemplo, entre 1850 y 1853 la diferencia es 3. Los bytes AB vistos como decimales también muestran una diferencia de 3 unidades (61-58 = 3).

Para la decodificación de estos datos se eliminó la componente decimal AB de los datos visualizados en la computadora. Por ejemplo, al valor 1850 se le restó 58. A 1853 se le restó 61. Se encontró una correlación directa entre los bytes CD y los datos mostrados en el software del fabricante. El resultado de esta operación se muestra la A partir de estos resultados es posible construir un gráfico (Ver Fig. 3) donde se comparan los valores de los datos de la Tabla III. El gráfico resultante muestra una perfecta línea con una pendiente positiva. Al calcular el valor de esta pendiente se obtiene 256. De esta forma queda establecido que para convertir los bytes 0xAB 0xCD a los datos mostrados por el software en la computadora, se utiliza la siguiente expresión:

$$\text{Datos ODLS} = (CD_{10})(256) + (AB_{10}) \quad (1)$$

TABLA III.

A partir de estos resultados es posible construir un gráfico (Ver Fig. 3) donde se comparan los valores de los datos de la Tabla III. El gráfico resultante muestra una perfecta línea con una pendiente positiva. Al calcular el valor de esta pendiente se obtiene 256. De esta forma queda establecido que para convertir los bytes 0xAB 0xCD a los datos mostrados por el software en la computadora, se utiliza la siguiente expresión:

$$\text{Datos ODLS} = (CD_{10})(256) + (AB_{10}) \quad (1)$$

TABLA III
BYTES ENVIADOS Y RECIBIDOS POR LA COMPUTADORA AL ODYSSEY WATER LEVEL SENSOR VISTOS DESDE EL DEVICE MONITORING STUDIO. SE HA ELIMINADO LA COMPONENTE DECIMAL AB DE LOS DATOS DEL MODO LTR VISUALIZADOS EN LA APLICACIÓN DEL DATALOGGER

Bytes Recibidos	Valores Decimales	Datos sin la componente AB
0x3A 0x07	58 7	1792
0x3D 0x07	61 7	1792
0x3F 0x07	63 7	1792
0x6A 0x09	106 9	2304
0xB7 0x09	183 9	2304
0xFE 0x09	254 9	2304
0xA3 0x0A	163 10	2560
0xB5 0x0A	181 10	2560
0xC3 0x0C	195 12	3072
0xC5 0x0C	197 12	3072

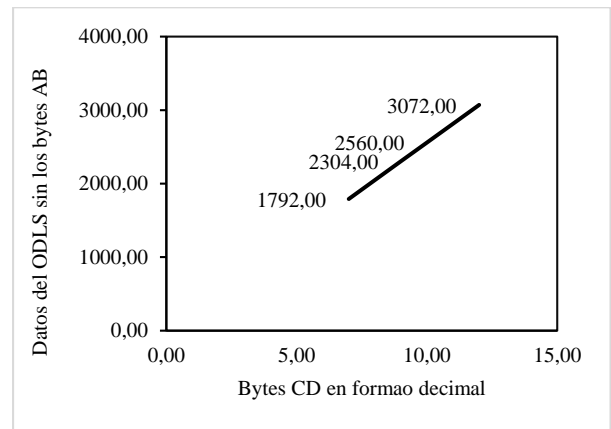


Fig. 3 Relación entre los datos mostrados en el software del fabricante, sin la componente decimal AB y los bytes CD en formato decimal

III. DISEÑO DE INTERFAZ DE COMUNICACIÓN

A. Selección de Sistemas Embebidos

Los dispositivos de la marca Arduino son una opción accesible y económica para el desarrollo de soluciones tecnológicas [17]-[18]. Para el diseño de la Interfaz de Comunicación se puede utilizar cualquier modelo de Arduino, ya que todos tienen la capacidad de establecer comunicación por medio de RS-232 con una computadora u otro dispositivo que utilice este protocolo. Sin embargo, para sacar el mejor provecho de los sensores de nivel de agua es importante que se exploren

opciones como la transmisión de datos a través de Internet, ya sea de forma directa o indirecta.

Uno de los modelos de Arduino que ofrece más posibilidades es el Arduino Yún, nacido de la combinación de un microcontrolador ATmega 32u4 y un microprocesador Atheros AR9331 [19]. Ambos elementos se combinan en una sola placa muy similar al Arduino Leonardo en cuanto a apariencia física. Gracias a la integración del microprocesador es posible conectar el Arduino a Internet a través de WiFi o cable Ethernet. También se cuenta con una distribución de Linux basada en OpenWRT. Esto abre la puerta a nuevas posibilidades como establecer comunicación por SSH, correr scripts de lenguajes de programación de alto nivel (Python, Java, PHP, NodeJS, entre otros), montar bases de datos (MySQL, MongoDB, SQLite, etc) y un servidor web desde una simple memoria MicroSD.

Sin embargo, el principal inconveniente con el Arduino Yún es su elevado costo en el mercado. El precio promedio de un Arduino simple como el modelo UNO oscila entre 15 y 20 USD, mientras que el precio del Arduino Yún está entre 70 y 80 USD [20].

Afortunadamente en el mercado existen opciones similares como el Linkit Smart Duo 7688 (LSD7), diseñado por la empresa Seeed Studio. Este modelo está basado en un Arduino Yún, pero a una fracción del costo [21]. El LSD7 es básicamente un Arduino Yún de tamaño reducido, pero con mejores prestaciones de hardware de procesamiento y con un precio que oscila entre 15 y 20 USD. El lenguaje de programación es el mismo que el del Arduino, siendo muy sencillo reemplazar un Arduino Yún por un LSD7 en cualquier aplicación.

Otra opción para el desarrollo de la interfaz de comunicación es el Raspberry Pi, aunque éste presenta un mayor consumo energético [22] y menor desempeño en condiciones de trabajo a la intemperie.

B. Comunicación por RS-232

Para la comunicación entre el *datalogger* y el Linkit Smart Duo 7688 se utiliza una interfaz basada en el circuito integrado MAX3232 (ver **¡Error! No se encuentra el origen de la referencia.**).



Fig. 4 Módulo de conversión de RS-232 a TTL para la comunicación entre el microcontrolador y el *datalogger*

Este se encarga de adecuar los niveles lógicos de voltaje (TTL) entre ambas partes. Existen modelos comerciales de módulos que incluyen este circuito integrado con todos los elementos electrónicos que se requieran para su funcionamiento. El dispositivo utilizado para el diseño de la interfaz de comunicación incluye un puerto DB-9 para facilitar la conexión con el *datalogger*.

En el Linkit Smart Duo se utiliza la librería SoftwareSerial de Arduino para emular el Puerto Serie al que se conectará el módulo de conversión a TTL. Es importante tomar en cuenta que con Software Serial no es posible obtener velocidades de transmisión muy altas. Se recomienda trabajar entre 4800 y 9600 baudios para un mejor desempeño.

Una vez establecida la conexión entre el *datalogger* y el LSD7 es necesario iniciar el modo LTR del *datalogger* enviando los bytes 0x01 y 0x54 0x0A, con un retraso de al menos 100 milisegundos entre los dos grupos de bytes. Al utilizar un Puerto Serie virtual, el LSD7 mantiene su capacidad de comunicarse con la computadora a través de USB. Mapeando las entradas de bytes del puerto virtual al puerto serie nativo y aplicando la expresión de conversión de bytes a valores decimales se obtienen las lecturas del sensor en la computadora a través del LSD7 (ver Fig. 5).

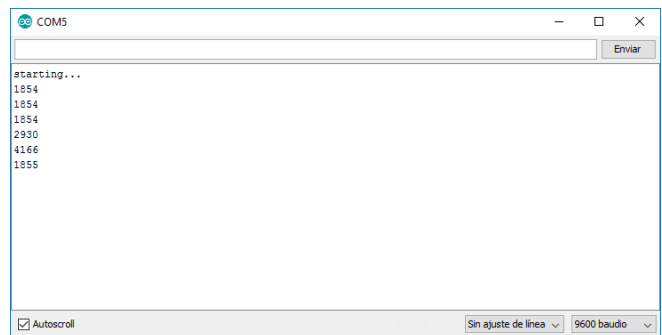


Fig. 5 Valores de capacitancia registrados en el *datalogger* y visualizados en una computadora a través del Monitor Serie del Arduino IDE, utilizando el Linkit Smart Duo 7688 como medio de comunicación

Es importante recordar que después de 64 muestreos, el modo LTR se desactivará automáticamente, por lo que es necesario establecer en el algoritmo que se reenvíen los bytes que dan inicio a este modo de pruebas. De esta forma se asegura la continuidad de los muestreos y se mantiene la capacidad de obtener información desde el sensor en tiempo real.

C. Transmisión de datos a través de Internet

El LSD7, al igual que el Arduino Yún, posee la capacidad de establecer comunicación entre el microcontrolador y el microprocesador. La Fig. 5 muestra las lecturas de capacitancia del *datalogger* impresas en el Puerto Serie desde el microcontrolador. Estableciendo comunicación interna con el microprocesador, el LSD7 puede almacenar las lecturas del sensor conectado al microcontrolador y guardarlas en una base de datos montada en una memoria MicroSD. Esto aumenta

drásticamente la capacidad de almacenamiento del *datalogger* de 32000 a decenas de millones de lecturas. También permite gozar de las ventajas del uso de bases de datos relacionales basadas en lenguaje SQL.

La posibilidad de transmitir datos a través de Internet depende de la disponibilidad de una conexión fija a una red WAN. Ya sea a través de WiFi o Ethernet, con el LSD7 resulta muy sencillo enviar la información a través del Protocolo de Transferencia de Hipertexto (HTTP) a un servidor remoto. La información puede ser transmitida a través de los métodos GET (inseguro) o POST (seguro) del HTTP, dependiendo de cuan sensible sea la información que se desea transmitir al servidor web. En el lado del servidor se establecen los sripts encargados de leer las peticiones por HTTP, recibir la información e introducirla a la base de datos.

Si se requiere un alto nivel de seguridad para el envío de información es posible utilizar lenguajes de programación del lado del servidor, tales como PHP, Python o NodeJS, entre otros. Con estos lenguajes se puede gestionar la encriptación de los datos desde el LSD7 para luego ser des-encriptados en el lado del servidor remoto en el que reposará la información.

Si no se cuenta con una red de Internet se pueden explorar otras opciones. El propio LSD7 posee un modo de funcionamiento como Access Point y la capacidad de ser utilizado como servidor web. De esta forma los datos se pueden extraer directamente desde el LSD7 a través de WiFi o Ethernet, ya sea por medio de una red LAN o por conexión directa (Ver Fig. 6).

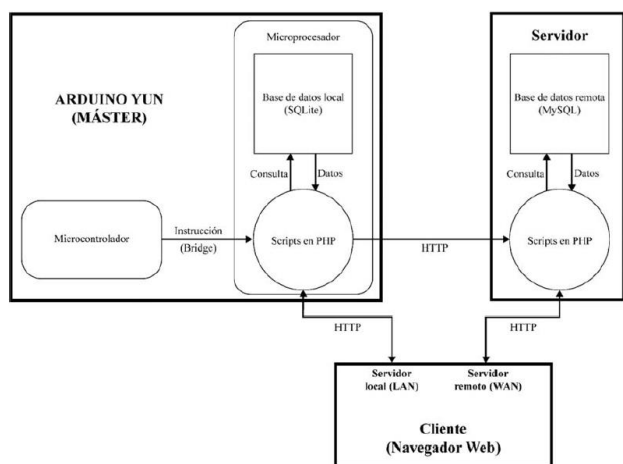


Fig. 6 Diagrama de bloques de la conexión un cliente y el Arduino Yún (o Linkit Smart Duo 7688), ya sea a través de un servidor web o de forma directa

Otras opciones han sido exploradas para los casos en los cuales no se cuenta con la disponibilidad de una conexión a Internet cerca del lugar en donde se instala el *datalogger*. Dependiendo de la zona geográfica en la que vaya a instalar el dispositivo se puede aplicar la tecnología GSM para transmisión de datos a través de la red celular. Otra opción es el uso de radiofrecuencia (ZigBee o Xbee) para llevar la información desde el *datalogger* hasta un punto en el que se cuenta con Internet o una red LAN o WLAN, si la distancia no

es muy larga. En este caso se utilizaría un modelo cualquiera de Arduino para extraer los datos del *datalogger* y transmitirlos inalámbricamente. El receptor de radiofrecuencia sería el LSD7 y éste se encargaría de almacenar la información y transmitirla para darle el tratamiento necesario.

En el mercado existen los módulos para Arduino que brindan la posibilidad de utilizar GSM y múltiples tecnologías de transmisión de datos por RF con diferentes protocolos y frecuencias. Estos módulos son muy económicos y fáciles de utilizar.

IV. PRUEBAS DE FUNCIONAMIENTO

El *datalogger* fue instalado junto a la interfaz de comunicación en un ambiente controlado durante un periodo de 7 días. El cable del sensor se colocó en un recipiente con un nivel fijo de agua. Durante este periodo se mantuvo un suministro fijo de energía para el LSD7, con el respaldo de una unidad UPS. Los resultados obtenidos fueron satisfactorios.

La interfaz de comunicación reportó registros de datos cada 30 segundos durante los 7 días de pruebas, con pequeñas variaciones de ± 2 segundos. Los datos de capacitancia mostraron pequeñas variaciones en el nivel de agua registrado a través del tiempo, equivalentes a ± 2.5 milímetros alrededor de un valor fijo. Estas variaciones están asociadas al propio *datalogger* y no la interfaz de comunicación.

Se utilizó una base de datos MySQL para almacenar los datos de capacitancia en el interior del LSD7. Para transmitir los datos al servidor remoto se utilizó un script en PHP. Este recibe la información directamente desde el microcontrolador y la envía por HTTP a través del método POST. No se utilizó ningún método de encriptación de datos.

Se diseñaron los scripts en lenguaje PHP para que la data pudiese ser descargada directamente en formato de hoja de cálculo de MS Excel, desde el servidor remoto y directamente desde el LSD7. De esta forma basta con entrar a un navegador web y acceder por HTTP a un dominio pre configurado en el servidor remoto. Al acceder a la dirección web se inicia automáticamente la descarga de la información almacenada en la base de datos empacada en un solo archivo con formato *.xls. El mismo resultado se obtiene si se establece conexión con con el LSD7 a través de WiFi y se accede al script a través de la puerta de enlace predeterminada.

Por último, se logró diseñar una interfaz web para la visualización de los datos muestreados en tiempo real a través de una gráfica en una página web montada en el servidor remoto. Utilizando HTML y Javascript se logró que el gráfico de los niveles de agua almacenados en la base de datos reaccione en tiempo real a cada muestreo ejecutado por el *datalogger*, tal como se muestra en la Fig. 7.

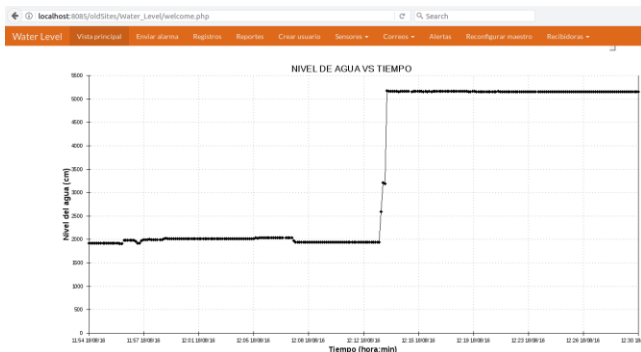


Fig. 7 Interfaz de visualización de datos en tiempo real, vista desde el navegador Google Chrome directamente desde el servidor web montado en la memoria MicroSD en el LSD7

V. DISCUSIÓN DE LOS RESULTADOS OBTENIDOS

A. Ventajas de la interfaz diseñada

Con el diseño de la interfaz de comunicación se brinda una solución económica y práctica para las pruebas experimentales en las cuales se emplea el medidor de nivel de agua capacitivo de Odyssey y se necesita obtener información muestreada en tiempo real. Es posible adaptar el diseño de esta interfaz a otros modelos de *datalogger* de la misma marca y a cualquier dispositivo con la capacidad de comunicarse por medio de RS-232 con otros equipos. Con este diseño se aprovecha las capacidades de recolectores de datos comerciales para la medición de parámetros de forma instantánea. Comúnmente los equipos con este tipo de características poseen precios muy altos en comparación con los recolectores de datos sin la capacidad de comunicarse en tiempo real.

La posibilidad de enviar información a un servidor remoto a través de Internet de forma segura abre las puertas para los *datalogger* sean utilizados en aplicaciones de automatización y control y en sistemas seguridad y monitoreo orientados a prevención de desastres, entre otras aplicaciones. Una vez del lado del servidor remoto se puede utilizar cualquier recurso web para brindarle el tratamiento adecuado a la información registrada en la base de datos.

B. Desventajas de la interfaz diseñada

El uso de esta interfaz no representa una ventaja en zonas sin una conexión a Internet, sin señal GSM o sin un lugar cercano al cual se pueda transmitir la información. El único beneficio sería la memoria expandida y la posibilidad de extraer los datos inalámbricamente por medio de WiFi. El uso del LSD7 supone un aumento en los requerimientos de energía consumida por parte del sistema de medición. En el caso del *datalogger* de Odyssey, éste cuenta con una batería que le brinda autonomía durante varios meses. Para que la interfaz de comunicación pueda ser utilizada se necesita un suministro fijo de energía o la aplicación de un banco de baterías junto con el uso de formas alternativas de energía, como paneles solares.

La interfaz de comunicación se alimenta en 5 voltios DC y puede llegar a consumir hasta 500 mA cuando se utiliza la comunicación inalámbrica por medio de WiFi. El promedio de consumo del LSD7 se ubica entre 200 mA y 300 mA. Los requerimientos de energía crecen si se requiere de la utilización de GSM o comunicación inalámbrica por RF. Para este tipo de situaciones el consumo puede crecer 1 Amperio, con picos de consumo de hasta 2 Amperios.

La utilización de un banco de baterías y de paneles solares complica el diseño y afecta económicamente cualquier proyecto en el que se decida utilizar este sistema.

VI. CONCLUSIONES

La interfaz de comunicación presentada en este documento representa una solución práctica para aumentar la capacidad de un *datalogger* comercial y convertirlo en un dispositivo para la transmisión de datos muestreados en tiempo real a un servidor a través de Internet.

El método utilizado para decodificar la transmisión y recepción de datos entre el sensor y la computadora es válido para cualquier tipo de dispositivo que se comunique datos por medio del puerto serie utilizando el protocolo RS-232. No importa si es un *datalogger*, un sensor u otro tipo de dispositivo. Esto representa una importante ventaja en los proyectos en los que se necesite usar equipos que deban estar conectados a una computadora para funcionar.

La utilización de sistemas embebidos para el desarrollo de soluciones tecnológicas es una alternativa eficiente a un costo económico muy bajo. La gran variedad de módulos de comunicación que existe y la amplia documentación disponible en Internet facilita la construcción de prototipos donde se aprovechan las características de dispositivos comerciales para construir nuevas aplicaciones que se adapten a las necesidades de los usuarios.

Gracias a la posibilidad de contar con una distribución de Linux montada en una placa electrónica, los datos pueden ser extraídos del dispositivo de medición, procesados y transmitidos al servidor remoto de forma segura, aprovechando las capacidades de los lenguajes de programación web orientados a servidores. Esto simplifica las tareas de almacenamiento de datos, encriptación, transmisión y visualización de resultados.

Todas las ventajas que ofrecen los sistemas embebidos requieren de un consumo energético superior al promedio de consumo de un *datalogger*. Por esta razón no es posible utilizar este tipo de soluciones tecnológicas a menos que se cumplan ciertas condiciones. Siempre será necesario contar con un suministro fijo de energía, ya sea a través de redes de suministro, bancos de baterías o energías renovables. También se necesita una conexión a Internet para aprovechar al máximo las características de estos sistemas. Sin estos mínimos requeridos, no vale la pena implementar esta tecnología.

REFERENCIAS

- [1] A. Crabit, F. Colin, J. Bailly, H. Ayroles and F. Garnier, "Soft Water Level Sensors for Characterizing the Hydrological Behaviour of Agricultural Catchments", *Sensors*, vol. 11, no. 12, pp. 4656-4673, 2011.
- [2] F. Reverter, X. Li and G. Meijer, "Liquid-level measurement system based on a remote grounded capacitive sensor", *Sensors and Actuators A: Physical*, vol. 138, no. 1, pp. 1-8, 2007.
- [3] P. Ross, "A water-level sensor using a capacitance to frequency converter", *Journal of Physics E: Scientific Instruments*, vol. 16, no. 9, pp. 827-828, 1983.
- [4] "Odyssey Capacitance Water Level Logger", *Odysseydatarecording.com*, 2017. [Online]. Available: http://odysseydatarecording.com/index.php?route=product/product&product_id=50. [Accessed: 7- Jan- 2017].
- [5] M. Blumstock, D. Tetzlaff, J. Dick, G. Nuetzmann and C. Soulsby, "Spatial organization of groundwater dynamics and streamflow response from different hydrogeological units in a montane catchment", *Hydrological Processes*, vol. 30, no. 21, pp. 3735-3753, 2016.
- [6] M. Marden, D. Rown and C. Phillips, "Sediment sources and delivery following plantation harvesting in a weathered volcanic terrain, Coromandel Peninsula, North Island, New Zealand", *Australian Journal of Soil Research*, vol. 44, no. 3, p. 219, 2006.
- [7] S. Noorduijn, P. Cook, C. Wood and N. White, "Using Sealed Wells to Measure Water Levels Beneath Streams and Floodplains", *Groundwater*, vol. 53, no. 6, pp. 872-876, 2015.
- [8] M. Rinderer, H. van Meerveld and J. Seibert, "Topographic controls on shallow groundwater levels in a steep, prealpine catchment: When are the TWI assumptions valid?", *Water Resources Research*, vol. 50, no. 7, pp. 6067-6080, 2014.
- [9] C. Stoof, R. Vervoort, J. Iwema, E. van den Elsen, A. Ferreira and C. Ritsema, "Hydrological response of a small catchment burned by experimental fire", *Hydrology and Earth System Sciences*, vol. 16, no. 2, pp. 267-285, 2012.
- [10] C. A. Starzyk, "Simulating surface water - groundwater interaction in the Bertrand Creek Watershed, B.C.," T, University of British Columbia, 2012.
- [11] B. Cui, Q. Yang, K. Zhang, X. Zhao and Z. You, "Responses of saltcedar (*Tamarix chinensis*) to water table depth and soil salinity in the Yellow River Delta, China", *Plant Ecology*, vol. 209, no. 2, pp. 279-290, 2010.
- [12] D. Haught and H. Meerveld, "Spatial variation in transient water table responses: differences between an upper and lower hillslope zone", *Hydrological Processes*, vol. 25, no. 25, pp. 3866-3877, 2011.
- [13] E. Heinz, P. Kraft, C. Buchen, H. Frede, E. Aquino and L. Breuer, "Set Up of an Automatic Water Quality Sampling System in Irrigation Agriculture", *Sensors*, vol. 14, no. 1, pp. 212-228, 2013.
- [14] T. Xie, X. Liu and T. Sun, "The effects of groundwater table and flood irrigation strategies on soil water and salt dynamics and reed water use in the Yellow River Delta, China", *Ecological Modelling*, vol. 222, no. 2, pp. 241-252, 2011.
- [15] P. Larson and C. Runyan, Evaluation of a Capacitance Water Level Recorder and Calibration Methods in an Urban Environment. Baltimore: University of Maryland Baltimore County, Center for Urban Environmental Research and Education, Baltimore, MD. September 2009.
- [16] Zhao Fumeng and Chen Shouyuan, "Design of temperature monitoring system for red wine fermentation based on ZigBee", 2012 International Symposium on Information Technologies in Medicine and Education, 2012.
- [17] A. Al-Busaidi, "Development of an educational environment for online control of a biped robot using MATLAB and Arduino", 2012 9th France-Japan & 7th Europe-Asia Congress on Mechatronics (MECATRONICS) / 13th Int'l Workshop on Research and Education in Mechatronics (REM), 2012.
- [18] A. D'Ausilio, "Arduino: A low-cost multipurpose lab equipment", *Behavior Research Methods*, vol. 44, no. 2, pp. 305-313, 2011.
- [19] "Arduino - ArduinoBoardYun", *Arduino.cc*, 2017. [Online]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardYun>. [Accessed: 11- Jan- 2017].
- [20] "Arduino Store USA", *Arduino Store USA*, 2017. [Online]. Available: <https://store.arduino.cc/>. [Accessed: 11- Jan- 2017].
- [21] J. Rodrigo and A. García, "LinkIt Smart Duo 7688: Primeros pasos" Panama Hitek, 2017. [Online]. Available: <http://panamahitek.com/linkit-smart-duo-7688-primeros-pasos/>. [Accessed: 11- Jan- 2017].
- [22] V. Vujovic and M. Maksimovic, "Raspberry Pi as a Wireless Sensor node: Performances and constraints", 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014.