

Diseño y construcción de un limnógrafo electrónico

Felipe Santiago Espinosa

Universidad Tecnológica de la Mixteca, Instituto de Electrónica y Mecatrónica,
Carretera a Acatlima km 2.5, Huajuapán de León Oaxaca

Teléfono: (01-953) 53 203 99 ext. 555

fsantiag@mixteco.utm.mx

Gabriela Álvarez Olguín

Universidad Tecnológica de la Mixteca, Instituto de Electrónica y Mecatrónica,
Carretera a Acatlima km 2.5, Huajuapán de León Oaxaca

Teléfono: (01-953) 53 203 99 ext. 550

galvarez@mixteco.utm.mx

Fermín Hugo Ramírez Leyva

Universidad Tecnológica de la Mixteca, Instituto de Electrónica y Mecatrónica,
Carretera a Acatlima km 2.5, Huajuapán de León Oaxaca

Teléfono: (01-953) 53 203 99 ext. 555

hugo@mixteco.utm.mx

Resumen

Un limnógrafo es un mecanismo que mide el nivel del agua, está formado por un flotador unido a una plumilla que marca el nivel en un papel fijado a un tambor, que gira mediante un mecanismo de relojería, generando una gráfica de niveles contra el tiempo conocida como limnograma. El limnograma es interpretado y capturado para usarse en diversos estudios hidrológicos, este procedimiento es manual y queda expuesto a errores. Buscando automatizar el registro y captura se desarrolló un limnógrafo electrónico con el microcontrolador ATmega8, el nivel se obtiene por medio de un

sensor ultrasónico y los datos se almacenan en una memoria EEPROM para preservarlos en ausencia de energía, se utiliza la versión 24PL512 de Microchip, la cual tiene una interfaz I2C. El periodo de muestreo es configurable por medio de dos botones. Mediante una pantalla LCD se muestra al usuario el estado del sistema. También se agrega un adaptador USB-TTL para la comunicación con una computadora, desde donde se puede solicitar el periodo de muestreo, el total de muestras y el conjunto de datos; así como comandar para reiniciar al contador de muestras. Todo desde una aplicación en LabVIEW, la cual también permite graficar las muestras.

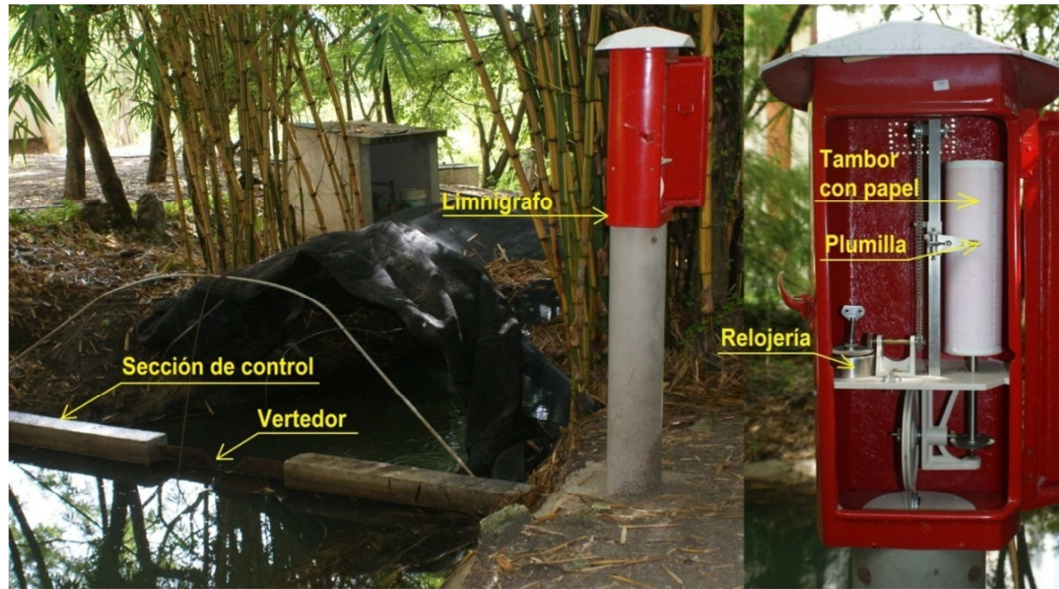
Palabra(s) Clave(s): ATmega8, LabVIEW, limnógrafo, nivel de agua.

1. Introducción

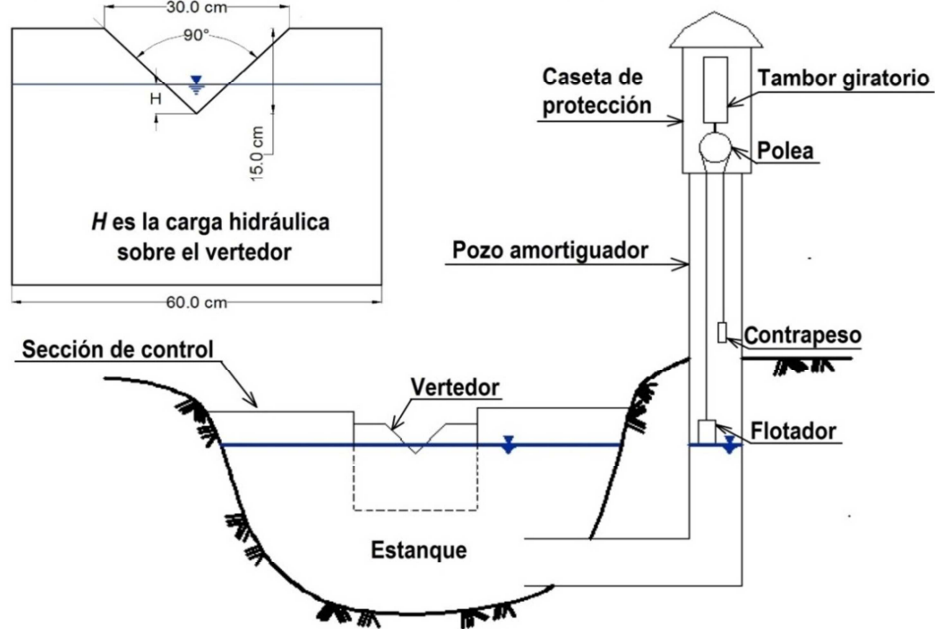
La medición de los niveles de agua en estanques, canales, ríos y presas, se utiliza para estimar el componente del ciclo hidrológico denominado escurrimiento; también conocido como gasto de un cauce, rendimiento de una cuenca o aportación líquida [1]. Los datos de escurrimiento son la base de una serie de estudios que contribuyen al bienestar social, entre los que están: la evaluación de la disponibilidad de agua superficial, la estimación de eventos de diseño en obras hidráulicas y la caracterización de sequías e inundaciones.

Las mediciones de los niveles de un cuerpo de agua se realizan en una estación de aforos formada por diferentes dispositivos e instrumentos de medición, de los cuales el limnógrafo se destaca por ser el único capaz de proporcionar registros continuos. Tradicionalmente en México el registro de datos con limnógrafos se realiza a través de un mecanismo formado por un flotador unido a una plumilla que marca los niveles del agua en un papel fijado a un tambor, que gira mediante un mecanismo de relojería, en la figura 1(a) se muestra una estación de aforo ubicada en la Universidad Tecnológica de la Mixteca, la cual es controlada y mantenida por el Instituto de Hidrología de la

misma institución, mientras que en la figura 1(b) se esquematiza a la imagen 1(a) para distinguir sus diferentes partes.



(a)



(b)

Fig. 1. (a) Estación de aforo instalada en la UTM, conformada por un limnígrafo convencional y una sección de control, (b) representación esquemática de la misma.

El resultado del limnógrafo es una gráfica de los niveles contra el tiempo conocida como limnograma [2]. Con esta información es posible determinar los gastos en función de la elevación, para ello se debe contar con una sección de control (figura 1), en la cual existe una única relación entre el gasto de descarga y la carga hidráulica, cuya gráfica es conocida como curva elevaciones-gastos, en la figura 2 se muestra un ejemplo de esta curva.

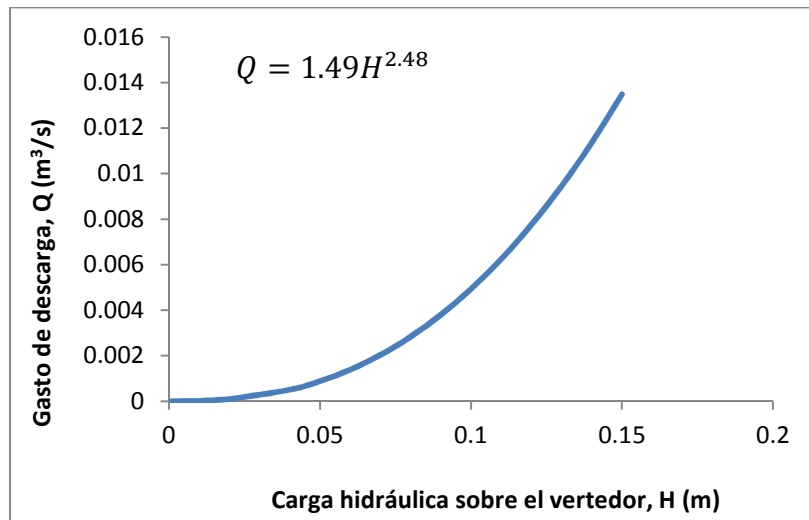


Fig. 2. Curva elevaciones-gastos, correspondiente a la sección de control de la estación de aforo instalada en la Universidad Tecnológica de la Mixteca.

El mecanismo del limnógrafo convencional representa una opción sencilla y económica, sin embargo, la interpretación visual del limnograma suele ser una tarea laboriosa debido a la gran cantidad de información contenida en una sola hoja de papel.

En las estaciones de aforo localizadas en zonas remotas, donde no siempre es posible contar con un operador permanente, se dificulta la medición constante de gastos. Para resolver este problema, en diferentes investigaciones se presentan propuestas para automatizar el monitoreo, registro y procesamiento de datos, adaptando dispositivos electrónicos a los mecanismos convencionales de medición existentes, como lo reflejan los trabajos de Vidal *et al.* [3], Morales [4] y Tamari *et al.* [5].

En la UTM se implementó un limnógrafo electrónico para las estaciones de aforo mantenidas por el instituto de Hidrología, el limnógrafo está basado en un ATmega8, un microcontrolador AVR de 8 bits de ATMEL [6], este dispositivo es suficiente para el manejo de periféricos que permiten obtener el nivel, capturar y almacenar la información, así como el envío de los datos a una computadora para su posterior procesamiento.

2. Desarrollo

Para la implementación del sistema se utilizó la Metodología Simplificada para Desarrollar Sistemas basados en Microcontroladores [7], esta metodología comprende los pasos que se irán cubriendo en los apartados de esta sección.

2.1. Planteamiento del problema

En la Figura 3 se muestran los elementos propuestos para el limnógrafo electrónico. La pantalla LCD proporcionará una interfaz visual para conocer el estado actual del sistema. El sensor ultrasónico trabaja por rebote, emite una ráfaga de pulsos y espera su retorno, conociendo el tiempo que tardó desde su emisión hasta su regreso, se estima la distancia a donde encontró un obstáculo, con este sensor se medirá la distancia al nivel del agua.



Fig. 3. Requerimientos del Limnógrafo Electrónico.

Las mediciones se irán almacenando en la memoria del sistema, el periodo de almacenamiento debe ser configurable para seleccionar la mejor opción de acuerdo con

las condiciones ambientales, por ello, se requieren dos botones para configurar el periodo de muestreo. Con un botón se dará paso a la configuración y con el otro se podrá seleccionar uno de los intervalos de tiempo preestablecidos, se consideran las siguientes opciones: 1, 5, 10, 20 y 30 minutos.

La memoria en donde se almacenen los datos debe ser no-volátil, lo que significa que debe mantener la información aún en ausencia de energía, es importante que la información no se pierda si después de algunos días de operación llega a ocurrir una falla temporal en la alimentación del sistema.

El sistema debe contar con un puerto USB para establecer la comunicación con una computadora, por este medio se enviarán los datos almacenados para su interpretación y graficado por medio de una aplicación. Desde la computadora se enviarán comandos que el sistema interpretará para que les pueda dar respuesta.

2.2. Requerimientos de hardware y software

En cuanto al Hardware, el primer elemento a seleccionar es el microcontrolador, se empleará un ATmega8 porque tiene 3 puertos que resultan suficientes para los periféricos requeridos.

Adicionalmente se requiere de una pantalla LCD, empleándose una versión alfanumérica de 2 renglones y 16 caracteres por renglón. Así como los botones de configuración.

El sensor ultrasónico a utilizar es el HC-SR04, tiene un alcance hasta de 4 metros que resulta suficiente para la aplicación y, aunque su resolución es de 3 mm, por su disponibilidad y costo se considera adecuado para el desarrollo del primer prototipo [8].

El ATmega8 no cuenta con un puerto USB pero si tiene una USART, por lo que se empleará un adaptador de USB a TTL, de manera que para la aplicación de la computadora se presentará como un puerto serial (COM) convencional.

La memoria interna del microcontrolador no le proporciona suficiente autonomía porque los datos deben mantenerse aún ante ausencias breves de energía y el ATmega8 sólo tiene 512 bytes de EEPROM. Para ello se elige al dispositivo 24LC512, se trata de una memoria EEPROM de 64 Kbytes (512 Kbits) con interfaz I2C [6]. Con esta capacidad y seleccionando el periodo de muestreo más corto (1 minuto), al sistema se le dará una autonomía de 22 días puesto que cada dato ocupa 2 bytes. En el periodo más largo (30 minutos) la autonomía es de más de 600 días. Con la interfaz I2C se tiene la ventaja de que sólo se utilizan 2 terminales del microcontrolador y dos resistores de 2.2 K Ω para *pull-up*.

El último elemento de hardware a considerar es un cristal de 32.768 KHz, con el cual se operará al temporizador 2 del ATmega8 para que genere los intervalos de tiempo de muestreo, con el pre-escalador interno se pueden manejar periodos correspondientes a fracciones o múltiplos exactos de segundos reales.

Referente al software, se requiere de una biblioteca básica para el manejo del LCD con funciones básicas para su inicialización y manejo. Para simplificar el programa principal se planea que los recursos internos del MCU sean atendidos por interrupciones.

Además del programa del ATmega8, el limnógrafo debe complementarse con una aplicación para una computadora, por medio de la cual se haga la transferencia de los datos almacenados en la memoria del sistema, se considera el uso de LabVIEW para simplificar su desarrollo.

2.3. Diseño del hardware

El microcontrolador ATmega8 tiene tres puertos, puertos B y D de 8 bits, y puerto C de 7 bits, pero como el bit PC6 también corresponde con el *reset* del sistema, no se puede emplear para un propósito general. Dado que algunos elementos de hardware serán manejados por los recursos del MCU, éstos son los primeros que deben ubicarse porque utilizan las terminales dedicadas a cada recurso.

Para el oscilador externo se tienen destinadas las terminales PB6 y PB7 (OSC1 y OSC2, respectivamente), ahí se conectará el cristal de 32.768 KHz, que será la base de tiempo para el temporizador 2 del ATmega8.

En el puerto C está la terminal de *reset* (PC6) y en ella se colocará un botón. También está disponible la interfaz I2C, ocupando los pines PC4 (SDA, para datos) y PC5 (SCL, para reloj), en esta interfaz se conectará la memoria 24LC512.

Con respecto al puerto D, el adaptador USB-TTL se conectará en las terminales PD0 (RXD) y PD1 (TXD), para ser manejado por la USART del ATmega8. Los botones de configuración se manejarán con las interrupciones externas del MCU, ocupando los pines PD.2 (INT0) y PD.3 (INT1).

Después de colocar los dispositivos que requieren terminales dedicadas del MCU sólo falta por ubicar al LCD y al sensor ultrasónico. El LCD se manejará con una interfaz de 4 bits y se puede conectar en el puerto B, en donde aún quedan disponibles las terminales de PB0 a PB5, que resultan suficientes. El sensor ultrasónico se conectará en los dos bits menos significativos del puerto C, empleando PC0 como la salida para el disparo del sensor y PC1 para la entrada del eco que se recibe como respuesta.

En la figura 4 se muestra el diseño completo del hardware, aún queda disponible la parte alta del puerto D y dos terminales del puerto C para posibles expansiones del sistema.

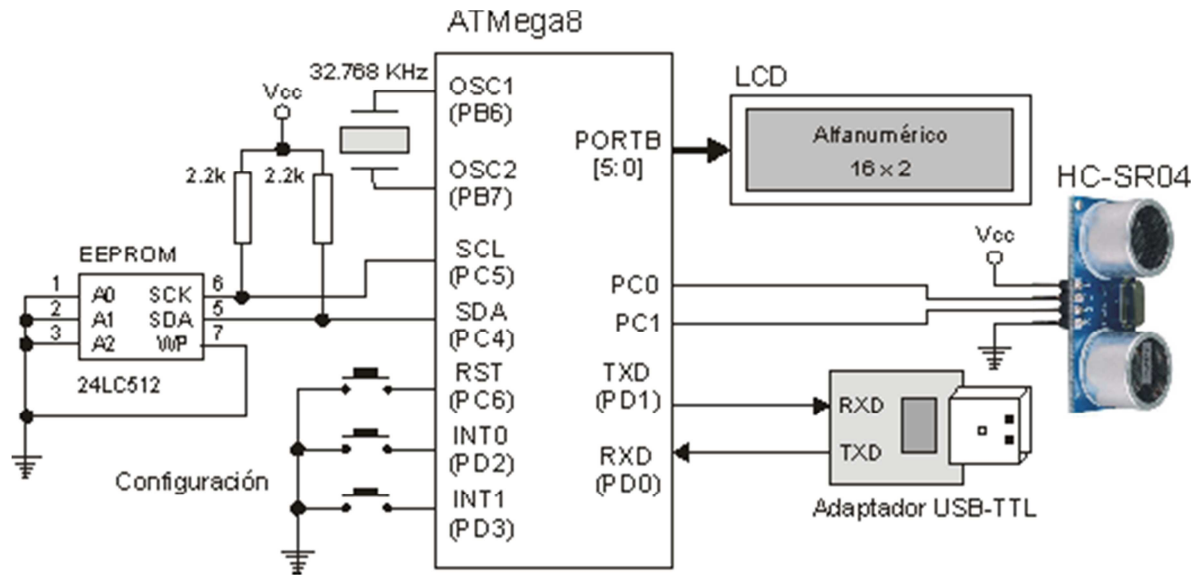


Fig. 4. Diseño del hardware para el limnógrafo electrónico.

2.4. Diseño del software

El sistema está soportado principalmente por los recursos del microcontrolador y por lo tanto, su funcionalidad está determinada por los eventos que ocurren en ellos. El programa principal básicamente seguirá el algoritmo de la figura 5, en donde se realiza la inicialización de los recursos. En las rutinas de atención a las interrupciones (ISR's) se estarán realizando las diferentes tareas del sistema.

1. Configuración de Entradas y Salidas
2. Configuración de las Interrupciones Externas
3. Inicialización del LCD
4. Lectura de la EEPROM interna del periodo de muestreo
5. Lectura de la EEPROM interna del total de datos
6. Configuración del Temporizador 2
7. Configuración de la USART
8. Habilitación de interrupciones
9. Lazo infinito ocioso

Fig. 5. Algoritmo del programa principal.

En los pasos 3 y 4 se realizan lecturas de la EEPROM interna, en ese espacio se almacenará la configuración elegida por el usuario para la toma de muestras y el total de datos almacenados, al usar una memoria no volátil para estos datos, esta información no se pierde aún en ausencia de energía. Cuando el microcontrolador se programe por primera vez se considerará un periodo de muestreo de 1 minuto. Los datos también serán almacenados en memoria EEPROM, pero en el dispositivo 24LC512, que es una memoria EEPROM de 64 Kbytes (512 Kbits) con interfaz I2C.

Los recursos que podrán generar eventos son:

- Interrupciones externas (INT0 e INT1), con la INT0 se dará paso a la configuración, mostrando el valor actual y permitiendo su modificación, los ajustes se harán con la INT1 y con la misma INT0 se aceptará y registrará el cambio, en ese momento también se respaldará en EEPROM interna. La INT1 será ignorada si con anterioridad no se presionó la INT0, con la INT1 habilitada sólo se mostrarán las diferentes opciones de configuración.
- Temporizador 2, dado que este recurso será manejado con el oscilador externo de 32.768 KHz, generará la base de tiempo para la toma de muestras. El temporizador 2 se configurará para que desborde cada 2 segundos, con un desbordamiento se incrementará un contador para luego comparar y evaluar si ya se debe almacenar de un dato. Con la finalidad de almacenar información más precisa, la distancia se medirá 10 veces y se obtendrá el promedio, este dato es el que será almacenado.
- Puerto Serie (USART), la comunicación con una computadora es esencial para el envío de la información almacenada. El microcontrolador podrá recibir comandos desde una computadora en cualquier momento, por medio de los cuales se le solicitará que realice una de las siguientes acciones:
 - Envío del total de datos (un entero formado con 2 bytes, iniciando con la parte alta).
 - Envío del periodo de muestro (un entero corto de 1 byte).
 - Envío de los datos (2 bytes por dato, enteros representando milímetros).

- Inicialización del contador de datos, con ello se pondrán ceros en las localidades destinadas para este parámetro en la memoria local. No es necesario limpiar todas las localidades de la memoria, con la puesta en ceros de este parámetro las nuevas escrituras remplazarán a las anteriores.

Para el cálculo de la distancia se considera que la velocidad de una señal de ultrasonido en el aire es: $v = 340 \text{ m/s}$, puesto que el MCU trabaja en el orden de μS y la distancia se medirá en mm, $v = 0.34 \text{ mm}/\mu\text{S}$.

Para calcular la distancia, se parte del hecho que $v=d/t$ (velocidad = distancia/ tiempo). Despejando d se obteniendo $d = v \times t$, dado que v tiene el valor constante previamente citado, únicamente se debe medir el tiempo que tardan en viajar los pulsos ultrasónicos.

El sensor HC-SR04, después de ser disparado, genera un pulso en alto cuya duración corresponde al tiempo que tardó en recibir la ráfaga ultrasónica de respuesta. Ese tiempo debe dividirse entre 2 debido a que involucra la ida y retorno, por lo tanto, conociendo a t en μs se tiene que

$$d = 0.17 \times t \text{ mm}, \quad (1)$$

con t en $\mu\text{segundos}$ la distancia se obtiene en milímetros.

En el microcontrolador se empleará al temporizador 1 para medir el ancho del pulso, este temporizador es de 16 bits, por lo que puede medir pulsos con ancho de hasta 65 535 μs , muy superior a lo requerido por la aplicación.

2.5. Implementación de hardware

En la figura 6 se muestra una fotografía con el hardware implementado en una tablilla de pruebas. Se resalta al microcontrolador ATmega8, al sensor ultrasónico, la memoria EEPROM y al adaptador USB-TTL. Al cristal se le agregan dos capacitores cerámicos para estabilización.

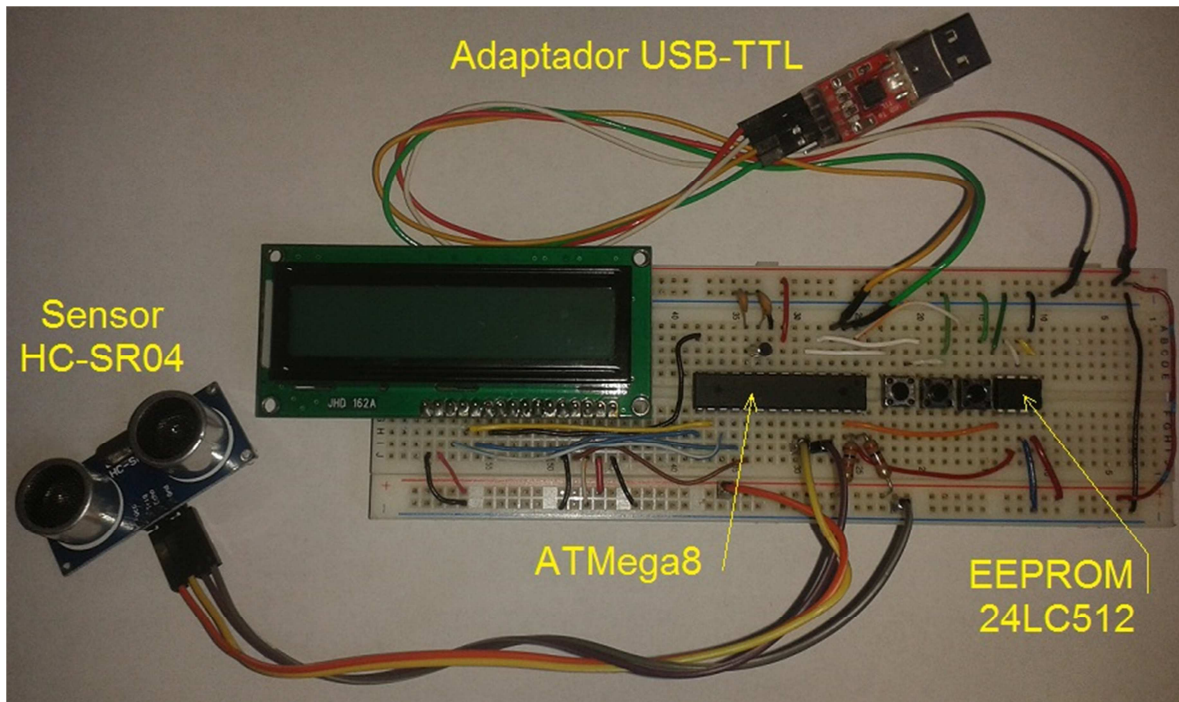


Fig. 6. Implementación del hardware.

2.6. Implementación del software

El programa fue desarrollado en lenguaje C, el programa principal sigue fielmente al algoritmo mostrado en la figura 5, como puede verse en la secuencia de código ilustrada en la figura 7.

```
int main(void){

    // Configuración de entradas y salidas
    DDRB = 0x3F;    // PORTB [5:0] - LCD, 6 y 7 para el oscilador
    DDRC = 0x01;    // PC0 - Salida de disparo y PC1 - Entrada de Eco
    DDRD = 0x02;    // Ent: RXD(0),INT0(2),INT1(3), Sal: TXD(1)
    PORTD = 0x0C;   // Pull-Up para las interrupciones

    // Configuración de las interrupciones externas
    MCUCR = 0x0A;   // Flanco de bajada
    GICR = 0xC0;    // Sólo habilita la INT0

    // Inicializa LCD
    LCD_reset();
    LCD_write_cad("    - U T M -        LIMNIGRAFO 1.0", 31);

    // Obtiene información de EEPROM interna
    Muestreo = eeprom_read_byte(&T_Muestreo);
    TDatos    = eeprom_read_word(&T_Datos);

    delay_Seg(2);
    sprintf(Cadena, "Muestreo= %2d min  Muestras = %3d",
                Muestreo, TDatos );
    LCD_write_cad(Cadena, 32);

    // Configuración del Temporizador 2
    ASSR = 0x08;    // Reloj externo
    TCCR2 = 0x06;   // División entre 256
    TIMSK = 0x40;   // Interrupción por desbordamiento del TIMER2

    // Configuración de la USART
    UBRRL = 12;     // Transmisión a 9600 baudios
    UBRRH = 0;
    UCSRA = 0x02;
    UCSRB = 0x98;   // Habilita TX y RX con interrupción por RX
    UCSRC = 0x86;   // Datos de 8 bits, 1 bit de paro y sin paridad

    delay_Seg(2);  // Sistema Operando
    LCD_write_cad("    SISTEMA EN        OPERACION", 29);

    sei();         // Habilitador global de las interrupciones

    while(1)
        asm("NOP");
}
```

Fig. 7. Programa principal.

Puede notarse que, con excepción de los mensajes iniciales, el procesamiento no se realiza en el programa principal, éste se distribuye en las rutinas que dan servicio a las interrupciones cuya funcionalidad fue descrita en la sección 2.4, el programa principal queda en un lazo ocioso, es decir, sin realizar operaciones.

2.7. Integración

El programa se editó y compiló desde el entorno AVRStudio, generando el archivo binario (.HEX) para ser descargado en el dispositivo. La evaluación del programa fue exitosa excepto porque fue necesario dar formato a la información antes de enviarla a la computadora.

3. Aplicación de LabVIEW

En LabVIEW se desarrolló una aplicación para enviar comandos al sistema y obtener información, básicamente son cuatro comandos. En la figura 8 se muestra el diagrama de secuencias ilustrando la relación entre el microcontrolador y la computadora, por medio de la aplicación en LabVIEW.

Para enlazar al microcontrolador con la computadora se utiliza la Arquitectura de Software para Instrumentos Virtuales (VISA, *Virtual Instrument Software Architecture*), que es un estándar para configuración y programación de sistemas de instrumentación por medio de las interfaces GPIB, VXI, PXI, Serial, Ethernet y/o USB [10].

Para la comunicación serial, VISA provee los módulos de LabVIEW que permiten la configuración del puerto, escrituras y lecturas, así como el cierre del puerto para liberarlo de la aplicación y posibilitarlo para ser usado en otras aplicaciones.

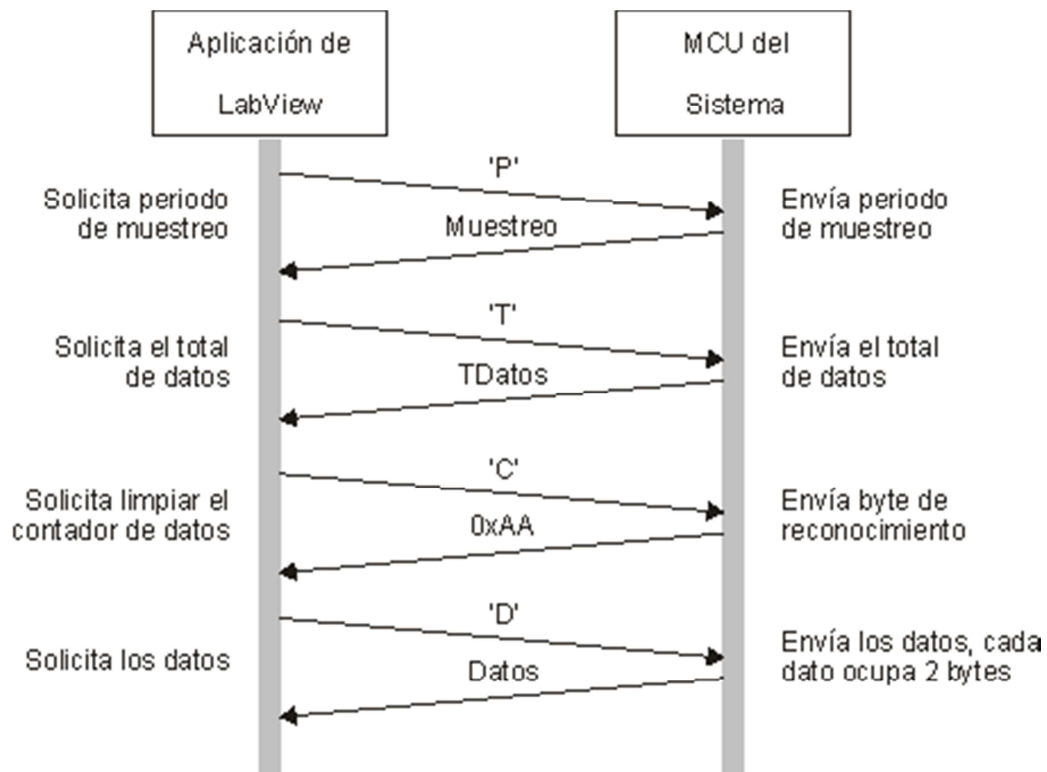


Fig. 8. Diagrama de secuencias que muestra la relación entre el sistema y una aplicación de computadora.

Los requerimientos para la aplicación serial son:

- Selección del puerto.
- Selección del comando a enviar al sistema.
- Botón para iniciar con el proceso de petición-recepción de información.
- Buffer para mostrar los datos recibidos.
- Gráfica de resultados.
- Botón para almacenar la información en un archivo.

Con estos requerimientos, se crea la interfaz mostrada en la figura 9, se agrega un botón para escalar la gráfica resultante conforme al periodo de muestreo seleccionado en el sistema.

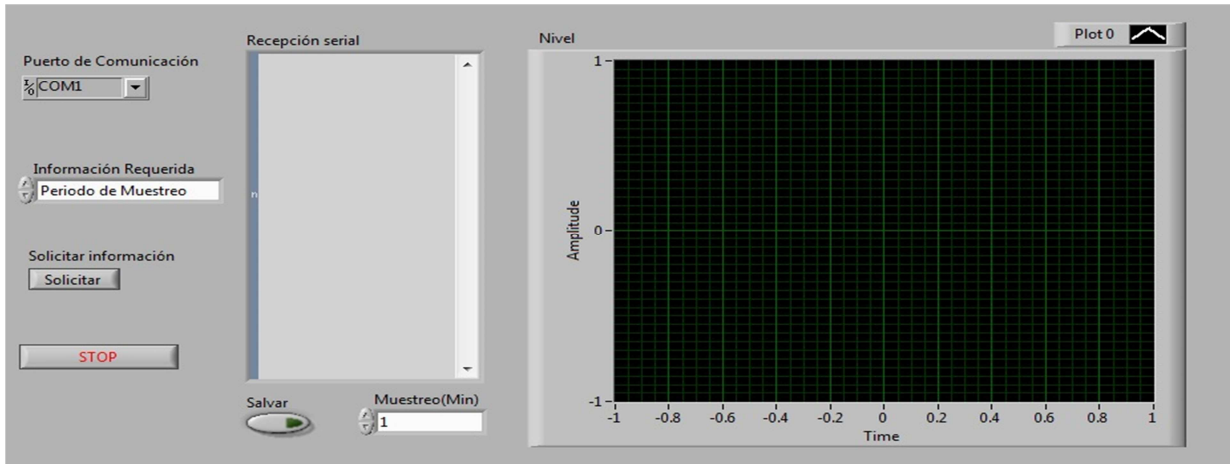


Fig. 9. Interfaz de la aplicación en LabVIEW.

El diagrama a bloques de la aplicación se muestra en la figura 10, en donde puede verse el flujo del programa. La inicialización del puerto se realiza sólo una vez y dentro de un ciclo infinito se pedirán los datos cuando se presione un botón, previamente se debió seleccionar el tipo de información. Se introduce un lapso de espera antes de realizar las lecturas. Las lecturas se realizan cuando hay datos disponibles y el resultado se muestra como una cadena para después generar su gráfica.

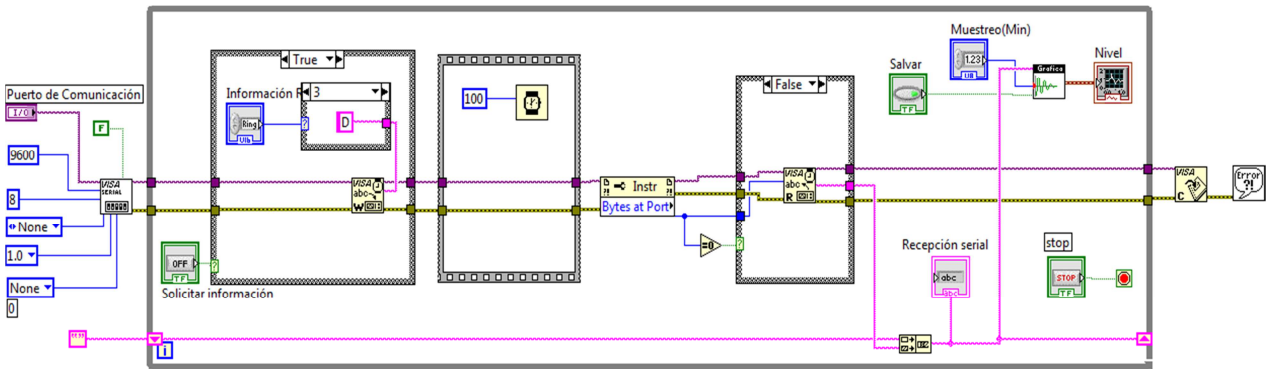


Fig. 10. Diagrama a bloques de la aplicación en LabVIEW.

El proceso se desarrollará de manera continua hasta que el botón de paro (STOP) sea presionado. El botón Salvar es para solicitar el respaldo de los datos en un archivo de texto.

4. Resultados

El programa en el microcontrolador ocupó los recursos mostrados en la tabla 1, puede notarse que la memoria de programa está muy próxima a agotarse, para ampliar la funcionalidad del sistema sería necesario migrar a un dispositivo con mayores prestaciones.

Memoria	Tamaño	Uso	Porcentaje
Programa (Flash)	8192 bytes	7586 bytes	92.6 %
Datos (SRAM)	1024 bytes	650 bytes	63.5 %
Datos (EEPROM)	512 bytes	3 bytes	0.6

Tabla 1. Porcentaje de uso de la memoria en el microcontrolador.

En la figura 11 se muestran algunas pantallas con el circuito en operación, en (a) se presenta la pantalla de bienvenida, en (b) el periodo de muestreo y el total de muestras, en (c) el modo de configuración, en (d) la toma de una muestra y en (e) el tiempo transcurrido en segundos entre la toma de una muestra y otra.



Fig. 11. Diversas pantallas del sistema en operación, (a) bienvenida, (b) periodo de muestreo y total de muestras, (c) configuración, (d) toma de una muestra y (e) tiempo transcurrido en segundos.

La figura 12 muestra al limnógrafo electrónico instalado en la estación de aforo de la UTM, se alimentó con una batería de 7.8 Volts para una operación independiente, por lo que fue necesario agregar un regulador de 5 volts.

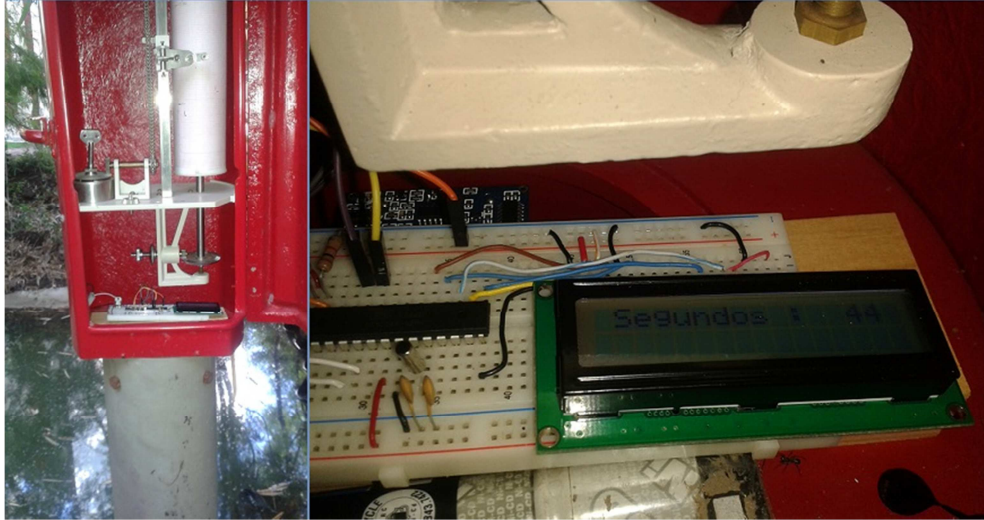


Fig. 12. Puesta en operación del Limnógrafo Electrónico.

El sistema se dejó operando por más de 15 horas con un periodo de muestreo de 1 minuto, en total se capturaron 913 datos, en la figura 13 se muestra la pantalla con la aplicación de LabVIEW en ejecución, se observa que la distancia desde la ubicación del sensor hasta el nivel del agua estuvo alrededor de 1700 mm.

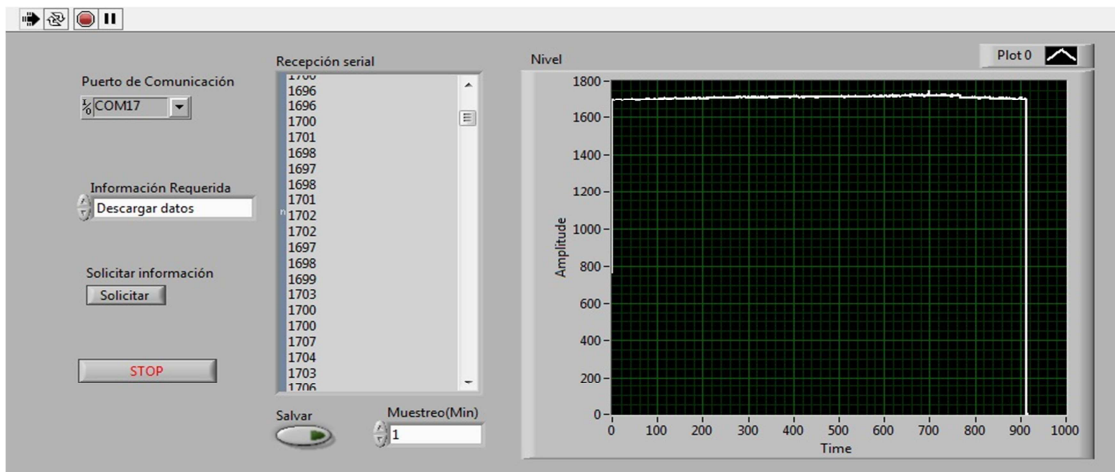
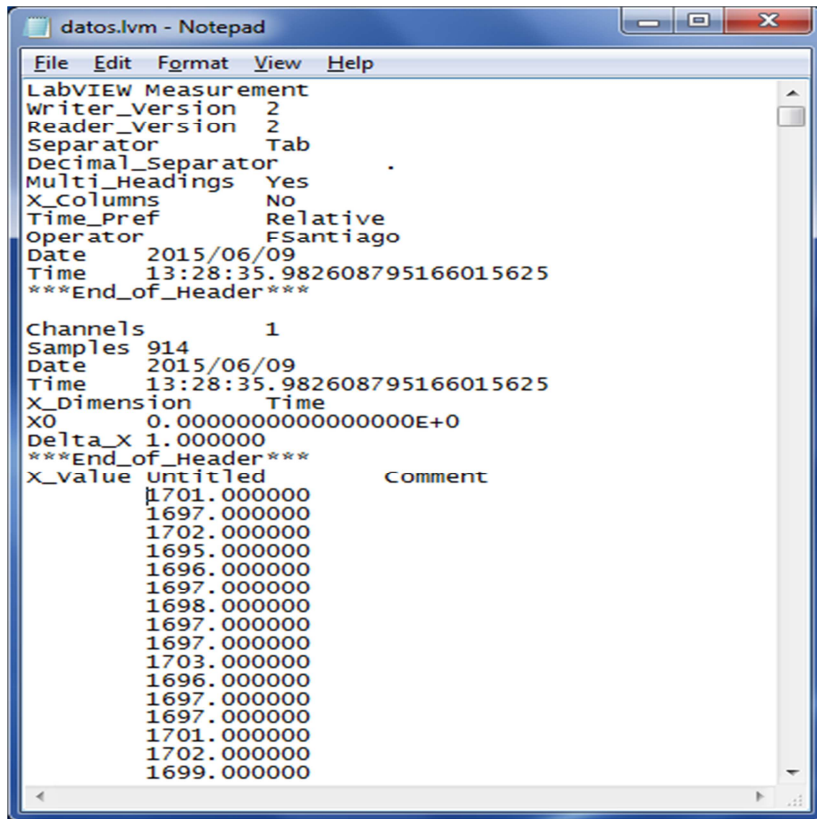


Fig. 13. Programa en LabVIEW mostrando los datos del sistema.

La figura 14 muestra parte del archivo generado por LabVIEW, los datos están dados en milímetros y la información inicial es agregada automáticamente por la aplicación.



```
datos.lvm - Notepad
File Edit Format View Help
LabVIEW Measurement
Writer_Version 2
Reader_Version 2
Separator Tab
Decimal_Separator .
Multi_Headings Yes
X_Columns NO
Time_Pref Relative
Operator FSantiago
Date 2015/06/09
Time 13:28:35.982608795166015625
***End_of_Header***

Channels 1
Samples 914
Date 2015/06/09
Time 13:28:35.982608795166015625
X_dimension Time
x0 0.0000000000000000E+0
Delta_X 1.000000
***End_of_Header***
X_value untitled Comment
1701.000000
1697.000000
1702.000000
1695.000000
1696.000000
1697.000000
1698.000000
1697.000000
1697.000000
1703.000000
1696.000000
1697.000000
1697.000000
1701.000000
1702.000000
1699.000000
```

Fig. 14. Muestras consecutivas.

5. Conclusiones

Con el limnógrafo electrónico terminado y puesto en operación se tienen las siguientes conclusiones:

El sistema funcionó favorablemente durante un tiempo razonable (15 horas en forma continua), aún con eso es necesario operarlo por más tiempo para validar su operación y detectar posibles errores, después de ello ya se podrá realizar el circuito impreso y el acondicionamiento para su colocación en la estación de aforo.

El sistema proporciona directamente el resultado del caudal, con lo cual se evita una tarea laboriosa de interpretación y análisis de datos, de esta forma se está automatizando un proceso que en la UTM aún se hace en forma manual.

Por la capacidad de la memoria EEPROM es posible almacenar la información de un número considerable de días (22 con un periodo de 1 minuto), pero como esta memoria se maneja con la interfaz I2C, en el mismo bus se pueden agregar más memorias con ajustes mínimos en el programa. Con ello el sistema podrá operar independientemente durante más tiempo, lo que resulta muy conveniente para sitios de difícil acceso, un buen complemento sería un panel solar junto con una batería recargable.

Como trabajo inmediato se planea el desarrollo de un circuito de acondicionamiento para la alimentación del sistema desde un panel solar y posteriormente realizar el circuito impreso. También se considera el desarrollo de un programa propietario en Java, para no depender del uso de LabView.

6. Referencias

- [1] D. F. Campos-Aranda, Procesos del ciclo hidrológico. Universidad de San Luis Potosí. México. 1998.
- [2] F. J. Aparicio-Mijares, Fundamentos de hidrología de superficie. 1997. Editorial Limusa, México.
- [3] S. Vidal, A. Solé, R. Lázaro, A. Duran, J. Puigdefábregas, “Estaciones automáticas de aforo y muestreo de caudales, de bajo costo y mantenimiento, para zonas remotas”. In Proceedings of the IV Simposio sobre el Agua en Andalucía. Vol. 1. 1996. 193-203 pp.
- [4] S. Tamari, A. Aguilar, V. Mejía, A. López, R. Álvarez, S. Ortega, Instrumentación de la estación hidrométrica Los Aldamas. Nuevo León. Anuario IMTA 2005.

- [5] F. H. F. Morales, “Desarrollo de un caudalímetro digital para la medición de caudal en ríos”. *Revista de Investigación, Desarrollo e Innovación*. Vol. 3. No. 1. 2013.
- [6] ATmega8: Avr RISC Microcontroller, Datasheet, 2006 Atmel Corporation.
- [7] Los Microcontroladores AVR de Atmel. Editado e Impreso por la Universidad Tecnológica de la Mixteca. México. 2012.
- [8] Ultrasonic Ranging Module HC - SR04, ELECFreaks, www.Electfreaks.com.
- [9] 24AA512/24LC512/24FC512, 512K I2C™ Serial EEPROM. Microchip Technology Inc. 2010.
- [10] LabView User Manual. National Instruments Corporation. January 1998.

7. Autores

M. C. Felipe Santiago Espinosa es Maestro en Ciencias con especialidad en Electrónica por parte del INAOE, incorporado al IEM de la Universidad Tecnológica de la Mixteca, en donde es Profesor-Investigador desde 1998. Actualmente está cursando el Doctorado en Robótica en la misma institución. En el año de 2012 publicó su libro titulado “Los Microcontroladores AVR de ATMEL”.

Dra. Gabriela Álvarez Olguín obtuvo el grado de Doctora en Ingeniería por la Universidad Nacional Autónoma de México. Actualmente es profesor-investigador del Instituto de Ingeniería de la Universidad Tecnológica de la Mixteca.

Dr. Fermín Hugo Ramírez Leyva obtuvo el doctorado en Ingeniería Mecatrónica en la Universidad Popular Autónoma del Estado de Puebla y la maestría en electrónica por el Instituto Nacional de Astrofísica Óptica y Electrónica. Actualmente es profesor investigador de la UTM, adscrito al Instituto de Electrónica y Mecatrónica hay desde 1999.