

UNIVERSIDAD NACIONAL DE CÓRDOBA
FACULTAD DE CIENCIAS EXACTAS, FÍSICAS y NATURALES



ELECTRONICA DIGITAL 3

Trabajo Práctico Final
"SeMaKu"
2023

Autores:

Matrícula	Apellidos y Nombres	Mail
42935935	Segura, Gaston	gastonsegura2908@mi.unc.edu.ar
39438057	Mayorga, Federica	federica.mayorga@mi.unc.edu.ar
19DH50302	Kulyk, Antonin	antonin.kulyk@insa-lyon.fr

ÍNDICE

ELECTRONICA DIGITAL 3	1
Introducción	3
Presentación del tema	3
Objetivos del proyecto	3
Desarrollo del trabajo	4
Pines	4
Timers 0, 1 y 2	4
ADC - Analog to Digital Converter	5
EXTINT - External Interrupts	5
UART - Universal Asynchronous Receiver Transmitter	5
DMA - Direct Memory Access	5
Diseño del proyecto	6
Componentes y elementos	6
Esquemáticos	6
Implementación	7
Descripción de los pasos seguidos durante la implementación	7
Resultados	8
Interfaz gráfica con MATLAB	8
Pruebas realizadas	9
Imágenes	9
Conclusiones	14
Lecciones aprendidas	15
Posibles mejoras	15
Anexos	16

Introducción

Presentación del tema

El objetivo del proyecto es realizar una interfaz con sensores de temperatura y humedad en suelo, con un microcontrolador LPC1769 y enviar los datos a PC mediante comunicación UART bidireccional. Los datos de temperatura y humedad se encuentran al obtener, encender y mostrar en una interfaz gráfica.

Buscando aplicar los conceptos aprendidos durante el cursado de la materia

Le but du projet est de réaliser un interfaçage de capteurs de température et d'humidité à un LPC1769 dans le but d'envoyer les datas sur un PC par l'intermédiaire d'une communication UART bidirectionnelle. Les datas de températures et d'humidité seront alors récupérées, lues et affichées sur un interface graphique..

Objetivos del proyecto

- Realizar una interfaz de captura de datos de humedad y de temperatura, desde el microcontrolador LPC1769.
- Programar el microcontrolador con las configuraciones de GPIO, UART, Interrupciones, Timers y la lectura de los canales de ADC.
- Permitir el reconocimiento de la comunicación UART por el puerto COM y recuperar los datos, para luego poder graficarlos.
- Réaliser un interfaçage des capteurs de d'humidité et de température à la plaque LPC1769.
- Firmware embarqué LPC1769 permettant le setup du GPIO de l'UART des interruptions Timers, la lecture dans l'ADC.
- Interface sur le pc permettant de détecter la connexion au port COM et récupérer les Frames envoyées dans les canaux UART et de les traiter en fonction de la donnée récupérée dans le corps.

Desarrollo del trabajo

Este proyecto se ha desarrollado utilizando el microcontrolador LPC1769 y se ha programado en lenguaje C, aprovechando la eficiencia y potencia proporcionada por la interfaz CMSIS (Cortex Microcontroller Software Interface Standard), en su contraparte de programación sobre registros.

Los sensores análogos se conectan a la placa a través del módulo de ADC (Analog to Digital Converter). El funcionamiento de los mismos, está indicado con luces LED (Azul para el sensor de humedad, Rojo para el sensor de temperatura, ambos pueden estar encendidos y a la vez no), contando también con un LED extra que indicará cuando se presiona de forma exitosa el pulsador que cambiara entre estos cuatro modos de uso.

Se hará una breve explicación de las principales funciones y configuraciones del software realizado para este proyecto, el cual se encuentra debidamente comentado y anexado al final de este informe.

Pines

La primera implementación del código de este proyecto, es la definición y configuración de los pines, tanto de entrada como de salida de la placa. De forma similar, se han configurado los pines de los diferentes puertos utilizando estructuras, que definen también su funcionamiento y si tienen programadas o no resistencias. Se han configurado los pines ADC0.4 y ADC0.5 del módulo de ADC. Luego, los pines que recibirán las interrupciones externas con los pulsadores y finalmente los de transmisión y recepción que se utilizaran en comunicación serial UART.

Timers 0, 1 y 2

La configuración de estructuras se repetirá a lo largo del proyecto, como se podrá observar. El timer 0 se configura para realizar la gestión del anti-rebote y su interrupción será cada 1,5 segundos que se traduce por un valor de Reload:

$$MatchValue = (FreqRelej * T) - 1 = (100/4(prescale)MHz * 1.5) - 1 = 37500000 - 1.$$

El timer 1 se configura para realizar una interrupción cada 1 segundo y así comenzar con la conversión del módulo de ADC en los canales ADC0.4 y ADC0.5. Para el timer 2, la valor de Reload será:

$$MatchValue = (FreqRelej * T) - 1 = (100/4MHz * 1) - 1 = 25000000 - 1.$$

Así también, el timer 2 se configura para realizar una interrupción cada 2 segundos, y así enviar los datos que se obtienen de los sensores. Su valor de Reload será:

$$MatchValue = (FreqRelej * T) - 1 = (100/4MHz * 2) - 1 = 50000000 - 1.$$

Todos los timers habilitan las interrupciones por uso del NVIC (Nested Vector Interrupt Controller).

ADC - Analog to Digital Converter

Una vez inicializado el ADC, los canales 4 y 5 se habilitan o deshabilitan según los diferentes modos de utilización de los sensores. Esto también es la parte encargada de el encendido de los LEDs que indican que sensor se encuentra activo en el momento.

Tiene habilitada la interrupción por uso del NVIC.

En éste módulo conectamos el sensor de temperatura LM35DZ. Éste tiene un voltaje entre sus terminales de salida y masa de $10\text{mV}/^{\circ}\text{C}$, llegando a un máximo de 1.5V para una medición de 150°C , en su conexión en modo básico. Se encuentra conectado directamente a la alimentación del puente CP2102 de 5V , ya que su funcionamiento está determinado entre los 4 y 30V .

A este mismo, le conectamos el sensor de humedad en suelo FC28. Este fue calibrado mediante mediciones directamente sobre agua, para poder ver sus valores límites por ADC, cuando está conectado directamente a la alimentación de la placa LPC1769, que brinda 3.3V . No se conecto de forma directa a la alimentación del puente CP2102 de 5V ya que en su salida encontramos valores superiores a 4.6V , que es el valor máximo que un pin analógico puede soportar.

EXTINT - External Interrupts

La configuración de las interrupciones externas, es más acotada y se centra en definir cuando se realiza la misma. En este caso, ambas estructuras son sensibles por nivel bajo. Tienen habilitadas las interrupciones por uso del NVIC.

En estos pines, fueron conectadas las entradas de los pulsadores de dos pines, que serán quienes producen las interrupciones externas por bajo nivel y enciendan las luces LED que indican el funcionamiento de los sensores.

UART - Universal Asynchronous Receiver Transmitter

La configuración de la comunicación, es sobre habilitación FIFOs, encargadas para la transmisión y recepción de datos, así también como la inicialización del periférico UART en sí. Se habilitan las interrupciones por su receptor y también, por uno del NVIC.

DMA - Direct Memory Access

La configuración de la estructura, pensada para acceder a la memoria de forma automática, sin pasar por el CORE de la placa. Inicializando su controlador, configurando su canal y las direcciones de memoria que serán necesarias para la transferencia de datos. Finalmente se habilita su canal para movimientos de datos.

Implementación

Antes de la implementación del proyecto, se debió planificar la temática y el objetivo, para así poder concordar y reunir los recursos necesarios, los plazos de tiempo disponibles. Una vez consolidado este paso, se procedió a la proyección del proyecto, el uso de sus pines de los diferentes puertos y los módulos necesarios a utilizar.

En los primeros avances, confirmar el estado de la placa con el microcontrolador a utilizar, comprobar que sus pines y salidas de ADC estaban funcionando.

Repasar las especificaciones electrónicas de todos los componentes y ver que se cumplan para un correcto funcionamiento y evitar así desperfectos sobre el avance del proyecto.

Sobre estos pasos, le seguía la realización del código de software que nos permitirá implementar el funcionamiento del proyecto. Y en una negociación sobre los aspectos a cumplir para la entrega de este proyecto, se agregó la creación de una interfaz para visualizar los datos muestreados por los sensores.

Descripción de los pasos seguidos durante la implementación

- 1) Planificación de Pines y Puertos, en base a los módulos de hardware.
- 2) Recopilación de componentes y requerimientos para los módulos y sensores.
- 3) Verificación sobre la placa y los componentes, sus especificaciones electrónicas.
- 4) Realización y pruebas de código.
- 5) Creación de la interfaz gráfica utilizando MATLAB.

Resultados

Interfaz gráfica con MATLAB

Para obtener un mejor entendimiento de los datos obtenidos por los sensores, que han pasado por la placa, se realizó una interfaz gráfica utilizando la herramienta de software “MATLAB”, la cual a su vez es clave en la visualización de la bidireccionalidad de la comunicación serie con UART.

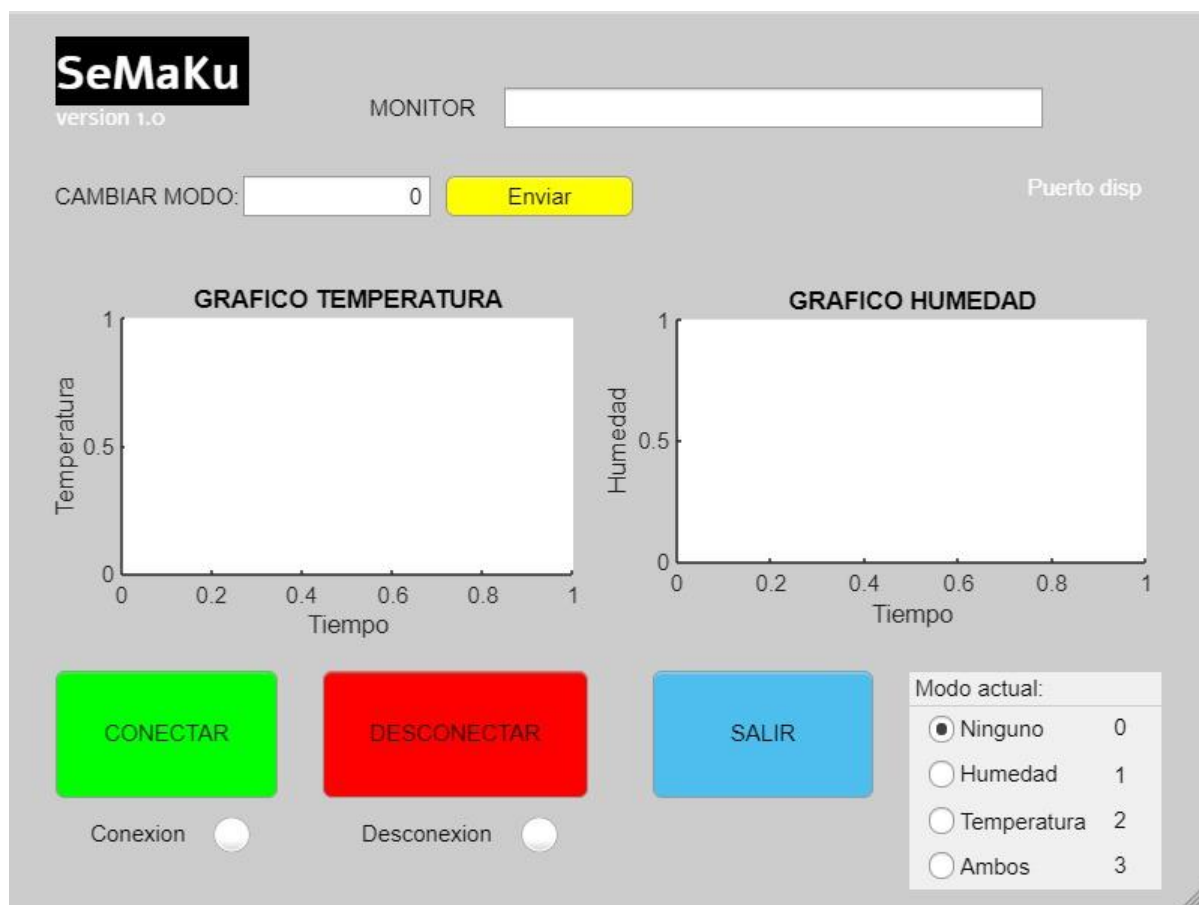


Figura 2: Interfaz gráfica de “SeMaKu” en su estado inicial.

Se observa como la interfaz también comunica el modo actual, es decir, que sensores se encuentran activos en el momento de conversión de datos. Es también posible cambiar el modo por PC o dispositivo apto para el uso de la aplicación portátil.

El software necesario para poder desarrollar esta interfaz, se va a encontrar anexo al archivo, así mismo como el acceso a su aplicación portátil para una mejor visualización en cualquier otro dispositivo o PC que cuente con MATLAB en su sistema.

Pruebas realizadas

Imágenes

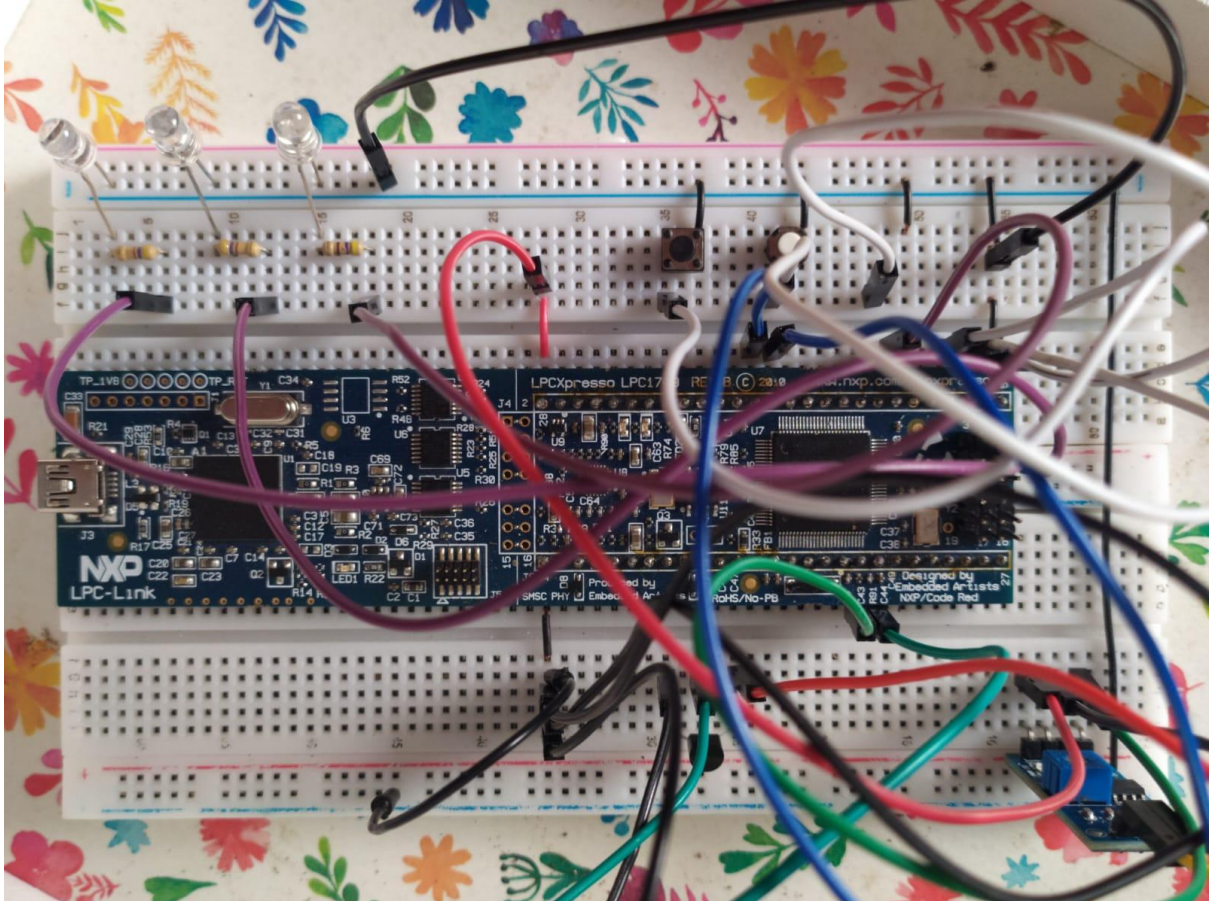


Figura 3: Vista superior del trabajo realizado.



Figura 4: Puente USB a UART CP2102.



Figura 5: Sensor de humedad en suelo FC28.

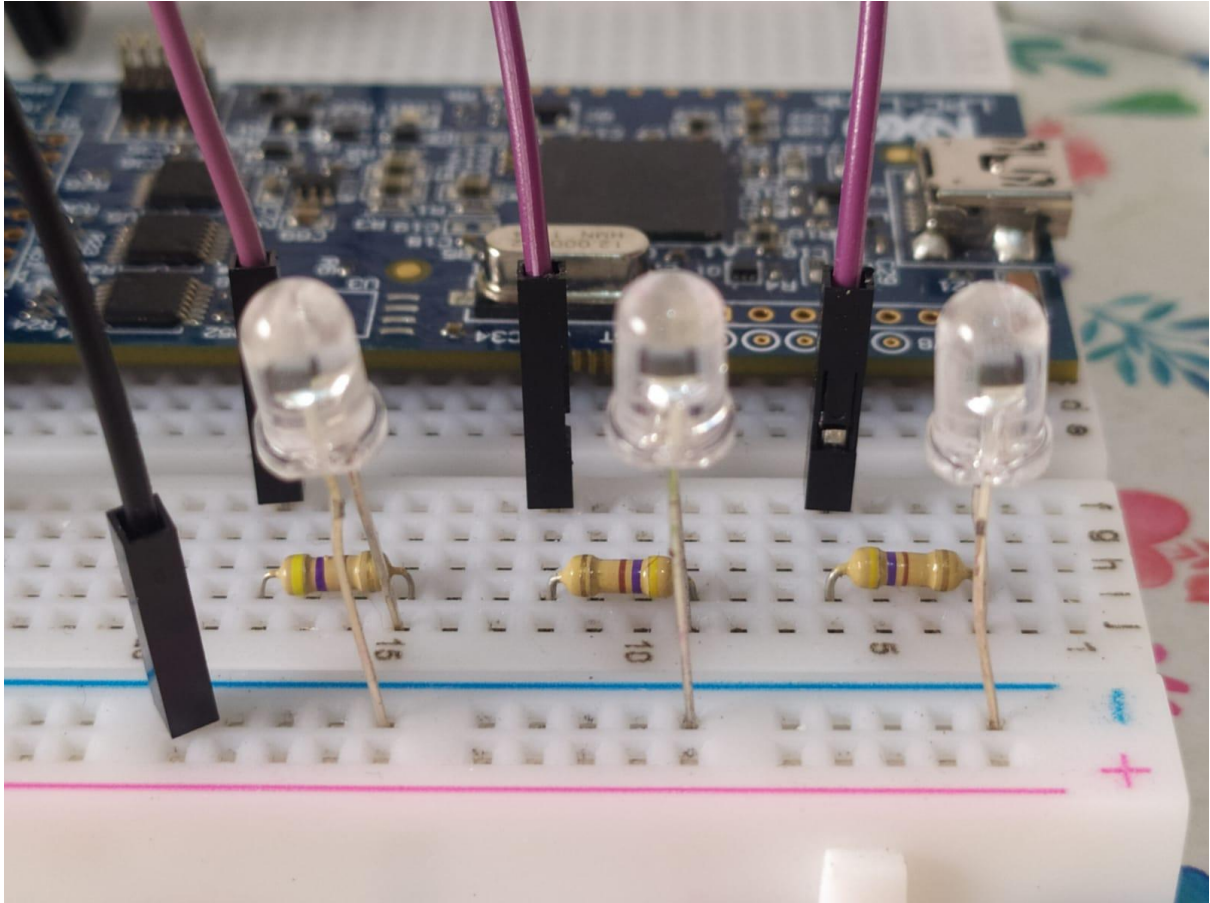


Figura 7: Implementación de los pulsadores e indicadores LED.

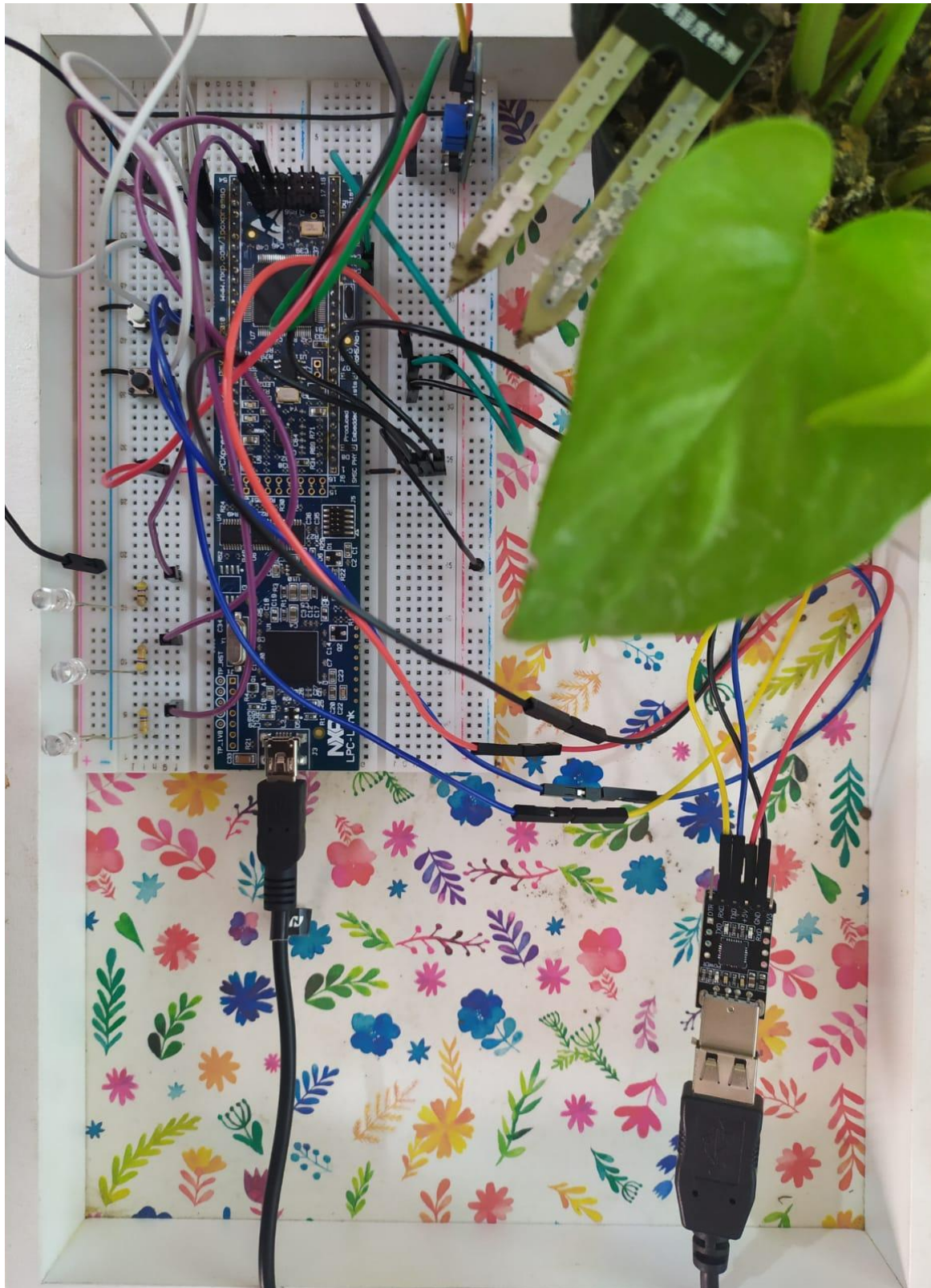


Figura 8: Trabajo Práctico Final de Electrónica Digital 3.

Conclusiones

A lo largo del proyecto, aprendimos a utilizar de manera práctica las placas microcontroladoras, centrándonos en la LPC1769. Estas al ser programables, ofrecen flexibilidad para diversas aplicaciones, convirtiéndolas en herramientas altamente útiles para una variedad de proyectos o trabajos. Utilizamos sus módulos para desarrollar un dispositivo de control, como se describió anteriormente.

Durante el proceso, nos enfrentamos a varias situaciones problemáticas que se resolvieron de manera efectiva mediante implementaciones tanto de hardware como de software. Un ejemplo destacado fue la experimentación sobre el funcionamiento de los módulos trabajando de forma simultánea. Asimismo, se emplearon estrategias de software para abordar las variaciones de lectura del sensor LM35, así como la calibración del sensor FC28.

Exploramos a fondo el uso de sensores de temperatura y humedad, ampliando así la capacidad de nuestro proyecto para abordar aspectos ambientales. Esta interacción proporcionó una comprensión más profunda de la relación entre hardware y software, resaltando la importancia de la precisión en la medición de estas variables.

Adicionalmente, destacamos la integración de representaciones gráficas utilizando MATLAB para visualizar los datos medidos. Esta dimensión gráfica añadió un componente visual valioso al proyecto, facilitando la interpretación y análisis de las lecturas de temperatura y humedad en tiempo real. La capacidad de visualizar los datos de manera gráfica no solo mejoró la presentación de resultados, sino que también proporcionó una herramienta efectiva para la detección de patrones y tendencias en el comportamiento de las variables a lo largo del tiempo.

En conclusión, este proyecto no solo contribuye conocimientos sustanciales sobre el manejo de placas de desarrollo y la implementación de hardware y software, sino que también nos permitió explorar y aplicar conceptos específicos de sensores de temperatura y humedad, así como técnicas gráficas para la representación visual de datos. Estas habilidades y conocimientos combinados ofrecen una base sólida para futuros proyectos en el ámbito de la electrónica, computación y automatización.

Lecciones aprendidas

En primer lugar, comprendimos la importancia de dimensionar adecuadamente los resultados obtenidos por el Convertidor Analógico-Digital (ADC) para garantizar una recolección de datos precisa y eficiente.

Además, se adquirieron conocimientos profundos sobre los principios de comunicación UART, incluyendo la configuración de canales y la generación efectiva de mensajes. La implementación de una interfaz gráfica fue esencial, permitiendo una comunicación fluida entre el microcontrolador y la PC, destacando la utilización de la misma para recibir y actualizar las tramas UART de manera eficaz.

Otra lección clave fue la mejora del sistema mediante la implementación de funciones anti rebotes, lo que resultó fundamental para la estabilidad y precisión de las lecturas de los sensores. En resumen, estas lecciones aprendidas han fortalecido nuestra comprensión de la integración de hardware y software, mejorando la eficiencia y confiabilidad del proyecto en su conjunto.

Posibles mejoras

Se pueden implementar estrategias de ahorro de energía en el proyecto mediante la activación de un modo de suspensión profunda ("deep sleep mode") durante períodos de inactividad, logrando así apagar las partes del microcontrolador que no son necesarias. Esto se puede lograr mediante el uso de un Watchdog Timer :

```
void WatchDog_Init(uint32_t TimeOut)
{
    // Set the watchdog timer constant reload value
    LPC_WDT->WDTC = TimeOut * 256 * 4; // timeout value

    // Setup the Watchdog time operating mode in WDMOD register
    LPC_WDT->WDMOD = 0x5; // Watch dog enabled, reset disable , watchdog cleared

    // Enable watchdog by writing 0xAA followed by 0x55 to WDFEED register
    // Reload the watchdog timer with the WDTC value.
    LPC_WDT->WDFEED = 0xAA;
    LPC_WDT->WDFEED = 0x55;

    if ( ! LPC_WDT->WDCLKSEL | ( 1 << 31 ) )
    {
        // Select internal IRC oscillator to be able to wake up from deep-sleep mod
        LPC_WDT->WDCLKSEL &= ~(0x11);
    }

    NVIC_EnableIRQ(WDT_IRQn);
}
```

```
void WDT_IRQHandler(void)
{
    // //LPC_WDT->WDMOD &= ~WDTOF; /* clear the time-out interrupt flag */
    // LPC_WDT->WDMOD |= ( 0 << 2 ); /* clear the time-out interrupt flag */

    if ( LPC_WDT->WDMOD & 1 << 2 )
    {
        m_count++;

        // TODO: Wake up CPU!

        // Disable WatchDog Interrupt
        // or the watchdog interrupt request will be generated indefinitely...
        // NOT WORKING :S
        LPC_WDT->WDMOD &= ~(0x4);
        NVIC_DisableIRQ(WDT_IRQn);
    }
}
```

Figuras 9 y 10: Implementaciones para futuras mejoras.

Además, es recomendable implementar medidas de seguridad adecuadas para proteger los datos sensibles intercambiados entre el microcontrolador y la PC a través de la comunicación UART.

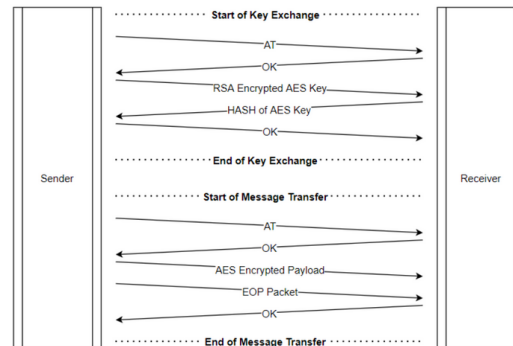


Figure 2: A hash function ensures the integrity of communications during secure key exchanges and message transfers. (Source: Infineon)

Figura 11: Trazo de los principios y fines de la comunicación protocolada.

Otra mejora importante sería la implementación de algoritmos de compensación para corregir posibles errores sistemáticos de los sensores.

Para asegurar la integridad de los datos con un backup, se podría realizar una copia de seguridad y almacenar los datos en la memoria DMA.

Finalmente, una forma de diversificar los medios de comunicación sería la incorporación de una transmisión LoRa mediante el uso de un módulo LoRa.

Anexos

- 1 - [Repositorio de GitHub - código SeMaKu](#)
- 2 - [LPC1769 - Data sheet](#)
- 3 - [LPC1769 - Manual de usuario](#)
- 4 - [Sensor de humedad](#)
- 5 - [Sensor de temperatura](#)
- 6 - [Puente USB a UART CP2102](#)
- 7 - [Carpeta con app, codigos y recursos](#)