

# USO NO ESTÁNDAR E IMPLEMENTACIÓN EXITOSA DEL PROTOCOLO I2C PARA UN SISTEMA DE MEDICIÓN DE TEMPERATURA EN ALDEAS ANDINAS A GRAN ALTITUD

JOEL FERNANDO PALOMINO MASCO  
j.palomino@pucp.edu.pe

JUAN ANTONIO PACO FERNÁNDEZ  
jpaco@pucp.edu.pe

MICHEL ANYELO ZARZOSA ROJAS  
michel.zaro@gmail.com

Grupo de Telecomunicaciones Rurales de la Pontificia Universidad Católica del Perú

## Resumen

Este artículo describe el diseño y desarrollo de un sistema de adquisición de datos de temperatura compacto y de bajo consumo energético, el cual emplea el protocolo I2C con cables de hasta ocho metros de largo. Este sistema se ha utilizado en la implementación de un proyecto del Gobierno del Perú con el objetivo de validar el uso de muros Trombe en localidades ubicadas a gran altitud (4000-5000 m s. n. m.). En este artículo se explica la construcción del módulo, el subsistema de adquisición de datos y se ofrece una visión general de la implementación del sistema desarrollado.

PALABRAS CLAVE: protocolo I2C / sensores / temperatura / adquisición de datos

## Abstract

### NON-STANDARD USE AND SUCCESSFUL IMPLEMENTATION OF I2C PROTOCOL FOR A TEMPERATURE MEASUREMENT SYSTEM IN HIGH ALTITUDE ANDEAN SMALL TOWNS

This article describes the design and development of a compact low energy consumption temperature data acquisition system using the I2C protocol with cables up to eight meters long. This system has been used in the implementation of a Peruvian government project aimed at validating the use of Trombe walls in small towns located at high altitude (4000 - 5000 m.a.s.l.). This article explains the construction of the module, the data acquisition subsystem, and presents an overview of the implementation of the developed system.

KEYWORDS: I2C protocol / sensors / temperature / data acquisition

## 1. INTRODUCCIÓN

En zonas rurales de los Andes peruanos (por encima de los 4000 m s. n. m.) donde hay falta de energía eléctrica y las bajas temperaturas congelan el agua, el Gobierno, a través del Fondo de Cooperación para el Desarrollo Social (Foncodes), en su tarea de ayudar a las poblaciones vulnerables que viven en condiciones de pobreza o pobreza extrema, ha desarrollado el proyecto Mi Abrigo (Foncodes, 2017). Este proyecto tiene como objetivo “calentar” las casas mediante el uso de muros Trombe (Ana, Anabela e Ivo, 2016; National Renewable Energy Laboratory, 2017). Esta tecnología consiste en una pared orientada al sol, preferiblemente hacia el norte en el hemisferio sur y hacia el sur en el hemisferio norte, que va absorbiendo el calor del día y lo entrega lentamente a la vivienda durante la noche. El muro está construido con materiales comunes como piedras, hormigón, arcilla y agua, los cuales se encapsulan con vidrio o incluso plástico para formar un colector de energía solar.

Durante la primera etapa de ese proyecto se seleccionaron algunas casas para medir el impacto social de la instalación de esa solución. Una vez instalados los muros Trombe, se requería un sistema de medición de temperatura para verificar si estos muros, efectivamente, aumentaron la temperatura de las casas respecto a su estado anterior (figura 1). Por esta razón, el Gobierno solicitó el desarrollo de un módulo de adquisición de datos de temperatura.

Según lo anterior, el sistema de medición desarrollado tiene como objetivo registrar el incremento de temperatura al interior de las viviendas y compararlo con la temperatura externa durante el periodo crítico del año (temperaturas más bajas); la expectativa del Gobierno era alcanzar 10 °C de diferencia en el momento más frío, teniendo en consideración que la diferencia habitual es de menos de 3 °C en horas de la noche. Técnicamente, el sistema se basa en el uso del protocolo I2C en cables eléctricos de hasta ocho metros de longitud para lo cual fue necesario diseñar la electrónica adecuada para asegurar, bajo consumo energético, un correcto funcionamiento en condiciones extremas de altitud y la adecuada gestión de tres sensores de temperatura en un solo bus.

En ese sentido, el presente artículo describe, en la segunda sección, tanto los requisitos del sistema y el protocolo de comunicación usado como las características del *datalogger* y la placa de propósito específico (*shield*) diseñadas como parte del sistema. En la tercera sección se mencionan, brevemente, algunos aspectos relacionados con la instalación de los sistemas mientras que en la cuarta y última sección se detallan las conclusiones obtenidas.



*Figura 1.* Muro Trombe instalado

Fuente: foto tomada durante actividades de instalación  
(archivo fotográfico del autor)

## 2. SISTEMA DE MEDICIÓN DE TEMPERATURA

En atención a los plazos establecidos por Foncodes, el prototipo del sistema de medición se desarrolló utilizando las placas y sensores disponibles en el mercado local al momento del diseño. Anteriormente, nuestro grupo de investigación había elaborado un sistema de medición con el sensor LM35; al tener una salida analógica no es fácil transportarla por cables sin realizar una digitalización que evite perder información. Una termocupla es un sensor con mucho potencial, pero su costo elevado y dimensiones no eran los adecuados para nuestro proyecto. Por esta razón, el sensor de temperatura debe ser de tamaño reducido, con salida digital y de bajo costo.

Para este prototipo se utilizó la placa Arduino Due con el sensor de temperatura TMP112 de Texas Instruments. Sobre este primer desarrollo se realizaron algunas optimizaciones para obtener la versión a instalarse en el marco del proyecto.

Según lo dispuesto por el Gobierno, el esfuerzo principal se orientó en obtener un dispositivo funcional y eficiente usando los componentes disponibles en el mercado local; no fue posible, por ejemplo, optimizar el sistema mediante el diseño de una placa hecha a medida de muy bajo consumo energético.

## 2.1 Requisitos

El sistema de adquisición debía utilizar tres sensores de temperatura con una precisión de  $\pm 0,5$  °C, para registrar las variaciones de temperatura cada cinco minutos; además, tenía que trabajar de manera autónoma por un período de tres meses como mínimo. Los sensores debían instalarse en tres ubicaciones predeterminadas: el primero a 1,8 metros de altura, que representa la temperatura que siente la persona mientras realiza sus actividades diarias; el segundo debería ubicarse a 0,5 metros, correspondiente a la altura en la cual la persona está durmiendo y el último sensor estaría ubicado en el exterior de la casa para comparar las lecturas de temperatura.

Según la información recibida, la distancia entre los sensores y la caja de control podría ser de hasta ocho metros. En este proyecto, uno de los mayores desafíos fue desarrollar un sistema de medición confiable con sensores conectados por un cable de ocho metros de largo, al contrario de la mayoría de las aplicaciones que usan sensores con cables cortos o soldados en la misma placa con el microprocesador. El principal inconveniente de esta distribución física es que el voltaje no es estable cuando se transmite información con sensores analógicos. Una solución para este problema es realizar calibraciones individuales por cada juego de sensores y cable.

## 2.2 Protocolo de comunicaciones

Muchos sensores digitales poseen un puerto de comunicación serial para transmitir los datos de manera confiable; con ello se evita la pérdida de información por caída de voltaje. Para los microprocesadores ATMEL, los protocolos seriales disponibles son los siguientes: UART, SPI, I2C y One Wire. Los dos primeros protocolos se utilizan para la comunicación punto a punto. En el caso de SPI, si bien se podría usar en comunicaciones punto-multipunto bajo configuración maestro-esclavo. Para lograr esto se debía agregar un dispositivo que conmute la línea entre cada sensor con el maestro, como un *buffer* tri-estado. Esta alternativa es un poco más complicada pues requiere usar una línea de selección para cada esclavo (sensor) y el requerimiento pasaba por usar un único cable. Lo mismo ocurre en el caso del protocolo UART, se debe tener un cable por cada sensor. Por otro lado, los protocolos seriales I2C y One Wire están diseñados para comunicar varios dispositivos a través de un único cable.

Para el diseño se requería sensores de temperatura que poseyeran un protocolo serial y con la capacidad para medir temperaturas negativas, por ejemplo, el *datalogger* de temperatura de LabJack (LabJack Measurement & Automation, 2017; Sousa, 2017). Se encontraron dos sensores: DS18B20 de MAXIM y TMP112 de Texas Instruments, cada uno con una precisión de 0,5 °C, y capaces de medir temperaturas de -10 °C. Luego de un análisis, se eligió el TMP112, no solo por sus características sino también por su disponibilidad en el mercado local. El sensor DS18B20 tuvo que ser descartado porque no había

suficientes unidades disponibles para venta (se requerían al menos 160 sensores) y el tiempo de espera para la importación era extenso.

El sensor TMP112 utiliza el protocolo I2C para transmitir información al microprocesador; este funciona con un suministro de 3,3 V, posee una resolución de 12 bits y una precisión de 0,5 °C. Su rango de medición va desde -40 °C a +125 °C. El protocolo I2C típicamente se emplea para transmitir información en cables cortos en la misma PCB; sin embargo, para el propósito previsto los sensores tenían que conectarse con un cable de hasta ocho metros de largo, pues esta era la distancia máxima de separación prevista para la instalación. El uso de integrados para extender el bus I2C como el P82B715 es una buena opción para este problema; sin embargo, como este sistema debía trabajar el mayor tiempo posible consumiendo la menor cantidad de energía posible, se optó por otra solución. De acuerdo con lo indicado por Truchsess (2010) sobre cómo alcanzar largas distancias con este protocolo en serie, se decidió, como solución, modificar el valor de las resistencias *pull-up* I2C, teniendo en cuenta que esta modificación implica un impacto despreciable en el consumo de corriente. De acuerdo con las ecuaciones (1) y (2), las cuales están descritas en el capítulo 7 de las especificaciones del protocolo I2C (NXP Semiconductors, 2014), las resistencias (máxima y mínima) son:

$$R_{p(MAX)} = \frac{t_r}{0.8473 * C_b} \quad (1)$$

$$R_{p(MIN)} = \frac{V_{DD} - V_{ol}}{I_{OL}} \quad (2)$$

Para el denominado Fast Mode, la frecuencia de la línea SCL está entre 10 KHz y 400 KHz, y la capacitancia del bus es la suma de la capacitancia del sensor TMP112 más la capacitancia del cable. Para el sensor de temperatura, de acuerdo con la hoja de datos (Texas Instruments, 2018), la capacitancia del pin digital es de 3 pF; el tiempo de subida es de 1 ms para una velocidad inferior a 100 KHz; el voltaje de suministro es de 3,3 voltios; el nivel de salida de voltaje es 0,4 voltios; la corriente de entrada es de 3 mA. Para el cable, la capacitancia es 33 pF / pie (General Cable/ Carol Brand, 2018); con una longitud de 8 metros la capacitancia del cable es 866 pF. Sumando ambos valores, la capacitancia total del bus, con 3 sensores, es 875 pF. Con estos valores, se define el nuevo rango como sigue:

$$R_{p(MAX)} = \frac{1 * 10^{-6}}{0.8473 * 875 * 10^{-12}} = 1,348.8\Omega \quad (3)$$

$$R_{p(MIN)} = \frac{3.3 - 0.4}{3 * 10^{-3}} = 966\Omega \quad (4)$$

Según las ecuaciones, la resistencia *pull-up* se estableció en un valor conveniente de 1 K $\Omega$  (ver resistencias R1 y R2 en figura 2). El cable usado es del tipo multifilar de cuatro hilos apantallado calibre 24. Adicionalmente, se tuvo en cuenta que en la mayoría de las placas Arduino el voltaje de funcionamiento del bus I2C está fijado en cinco voltios, mientras que el sensor de temperatura funciona a 3,3 voltios. Visto que las únicas placas que permiten este último nivel de voltaje son Arduino Due y Pro Mini, se eligió la placa Arduino Due, utilizando el segundo puerto I2C (SDA1, SCL1) que no tiene una resistencia *pull-up* fija. Debe indicarse que estas fórmulas son pertinentes debido a que la frecuencia usada es menor a 100 KHz.

### 2.3 Datalogger

Si bien los sensores son una parte fundamental del mismo sistema, por sí solos no pueden funcionar como un sistema de medición y adquisición de datos, por lo que es importante definir los requisitos necesarios para los demás componentes del *datalogger*:

- Reloj en tiempo real basado en el DS3231 de Maxim Integrated. Además de brindar la información de la fecha y hora de la medición, este dispositivo posee un pin de interrupción para informar al Arduino el momento en que debe medir.
- Tarjeta micro SD de 2 GB para almacenar un promedio de dos millones de lecturas de temperatura.
- Dos LED para mostrar el estado del sistema: uno que indica si el dispositivo está encendido ("DEVICE ON") y otro para indicar que está en operación ("DEVICE WORKING").
- Conector para unir el cable con los sensores.

Asimismo, se creó una librería especial que gestiona los componentes de la placa. Con ella el usuario podrá, además, establecer el nombre del *datalogger* de acuerdo con el lugar de instalación y elegir el tiempo de muestreo en minutos. Esta librería se implementó para trabajar con el entorno de Arduino IDE, y están implementadas en lenguaje C++. Se definieron las siguientes funciones para esta librería:

- Configurar las entradas y salidas digitales necesarias para cada periférico
- Leer la temperatura de cada sensor
- Almacenar las lecturas de temperatura en la tarjeta SD
- Obtener la marca de tiempo del reloj de tiempo real

El reloj de tiempo real también usa un puerto I2C para comunicarse. Si este dispositivo se hubiera conectado junto con el sensor de temperatura, la capacitancia del bus aumentaría y afectaría el valor de la resistencia *pull-up*. Además, el reloj DS3231 no

funciona bien en frecuencias inferiores a 100 KHz. Por esta razón, para evitar la pérdida de datos, el dispositivo se conectó al otro puerto I2C.

Para leer y escribir datos desde/hacia la tarjeta SD, el Arduino usa el puerto de SPI para realizar la acción. El *software* Arduino tiene una librería que permite administrar memorias de hasta 16 GB de memoria. Finalmente, los LED de estado se conectaron a dos pines digitales.

## 2.4 Suministro de energía

Para que el sistema funcione en forma autónoma, al menos durante tres meses, se debía implementar y optimizar el consumo de energía. Dado que la variación de temperatura es un proceso relativamente lento, no es necesario medir cada segundo o, en cambio, la medición se puede hacer después de algunos minutos. El Arduino Due ofrece dos modos de ahorro de energía: WAIT y BACKUP. Durante las pruebas, el modo WAIT no logró resultados positivos, mientras que el modo BACKUP sí logró los resultados esperados. En el modo BACKUP, el programa Arduino se reinicia al despertar, por lo que el programa tuvo que ajustarse para funcionar con esta condición. El pin de interrupción del DS3231 se usó para activar el sistema cada 1, 5, 10, 15 o 20 minutos, estas opciones se pueden elegir con la librería creada de acuerdo con los criterios del usuario. Esta interrupción se conectó al pin analógico A6 que corresponde a WKUP1 de acuerdo con la hoja de datos del SAM3X8E.

Para calcular la capacidad de batería requerida, el estado de funcionamiento del sistema se dividió en dos modos: activo e inactivo. Mientras está activo, el *datalogger* lee los datos del sensor de temperatura y los almacena en la tarjeta SD. Estas tareas tardan unos tres segundos en completarse y el consumo de energía es de 58 mA. En modo inactivo el *datalogger* entra en suspensión, deshabilitando todos los componentes y usando el modo BACKUP de ahorro de energía. En este modo, el consumo de energía cae a 13 mA. Para el cálculo de la capacidad de la batería requerida se calcula la cantidad de horas que está activo el dispositivo en un día y se multiplica por su consumo de corriente en amperios. Esto nos da la cantidad de AH por día. Por ejemplo, para el caso de tres muestras por horas, el dispositivo está activo por 0,06 horas y en modo inactivo 23,94 horas. El consumo es de 0,31 AH por día, esto se multiplica por la cantidad de días que va a estar prendido. Finalmente, el consumo obtenido se multiplica por 1,4 pues la batería se debe sobredimensionar con una capacidad adicional equivalente al 40 por ciento del total.

La tabla 1 resume la capacidad de batería requerida para una autonomía de 30, 60 y 90 días de acuerdo con el tiempo de muestreo del sistema.

Tabla 1  
 Capacidad de batería en función a los días de autonomía

Muestras por hora	Capacidad requerida de batería (A-H)		
	30 días de autonomía	60 días de autonomía	90 días de autonomía
60	15,69	31,37	47,06
12	13,83	27,67	41,50
6	13,60	27,21	40,81
4	13,53	27,05	40,58
3	13,49	26,97	40,46

Elaboración propia

El sistema propuesto debía trabajar por un periodo mínimo de tres meses. Debido a que el cambio de temperatura es un proceso lento, no es necesario tomar muestras muy seguido. Por ello se escogió un periodo de muestreo de doce muestras por hora con una autonomía de noventa días. La batería debía tener una capacidad mínima de 45 A-H. En el mercado se encontró la batería de plomo marca Ritar que trabaja con un voltaje de 12 voltios y una capacidad de 45 AH. La reducción de voltaje se realizó mediante un convertor DC que viene integrado en el mismo módulo Arduino.

## 2.5 Placa de propósito específico (*shield*)

Después de seleccionar y probar cuidadosamente cada componente del sistema de registro de datos, se diseñó y desarrolló una placa de propósito específico para integrar cada uno de estos componentes. Por otro lado, construir una PCB personalizada usando componentes integrados disponibles en el mercado local, permitió un ahorro significativo de espacio y brindó confiabilidad al sistema. Adicionalmente, esto permitió la simplificación del proceso de instalación, ya que el *datalogger* se reduce a una sola PCB, como se ve en las figuras 2 y 3. Las resistencias R1 y R2 son las resistencias *pull-up* calculadas en el apartado B.



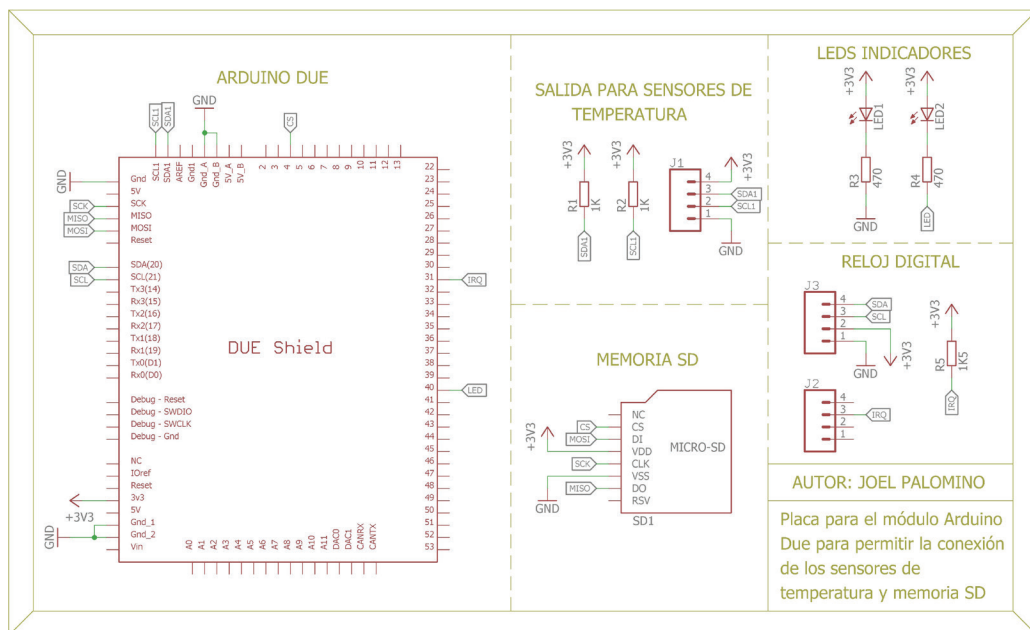


Figura 2. Diagrama esquemático de placa de propósito específico

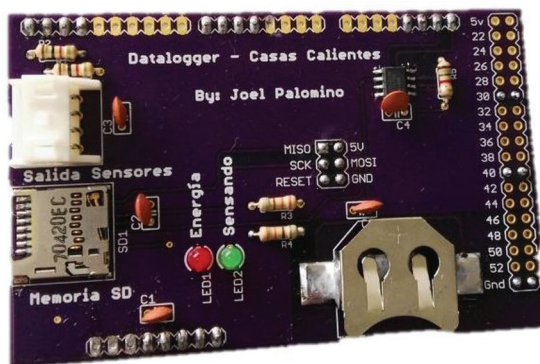


Figura 3. Placa de propósito específico

El *shield* fue diseñado para ubicarse encima del módulo Arduino Due, el cual tiene tres lados con borneras y solo tiene un borde libre para conexión USB y entrada de CC. En el diseño se requerían tres conectores para permitir la inserción de la tarjeta de memoria, el cable del sensor y el reloj de pila para el RTC. El espacio para montar estos conectores en el borde de la placa no era suficiente. Para resolver este problema, se suprimieron los conectores de los pines analógicos en la placa Arduino Due con el fin de permitir el montaje de la pila RTC.

Según se ha indicado, además del *shield*, también se desarrollaron tres placas para los sensores de temperatura (figuras 4 y 5). Estas placas fueron diseñadas para ser pequeñas y facilitar su instalación, pero también con el espacio mínimo indispensable para permitir la colocación de los conectores y dejar al sensor en el borde de la placa. Cada una de las placas desarrolladas fue rotulada para evitar confusiones en el proceso de instalación: fuera de la casa (EXTERIOR), dentro de la casa a 0,5 metros (INTERIOR 0,5) y dentro de la casa a 1,8 metros (INTERIOR 1,8)

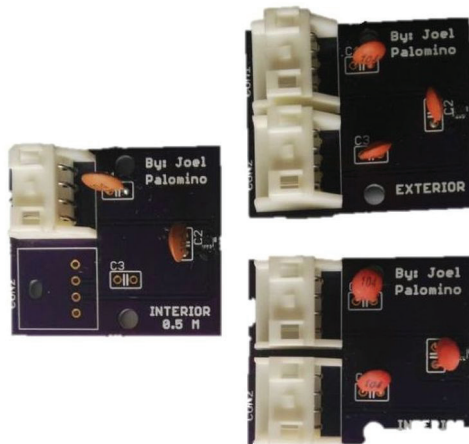


Figura 4. Placa de sensores de temperatura

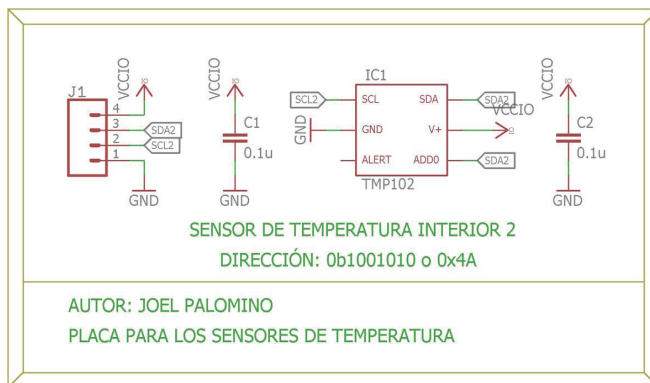


Figura 5. Diagrama esquemático de placa de sensores

De acuerdo con el diseño del *shield*, el RTC está conectado a los pines SDA y SCL del puerto I2C, en tanto que los pines SDA1 y SCL1 se conectan al sensor de temperatura. La razón, como se mencionó anteriormente, es que si el RTC está conectado junto con el sensor de temperatura, la capacitancia del bus aumentará y afectará el desempeño del bus I2C. Es necesario indicar que la precisión de los sensores fue validada mediante comparaciones con un equipo certificado (Fluke 971), según se muestra en la figura 6.

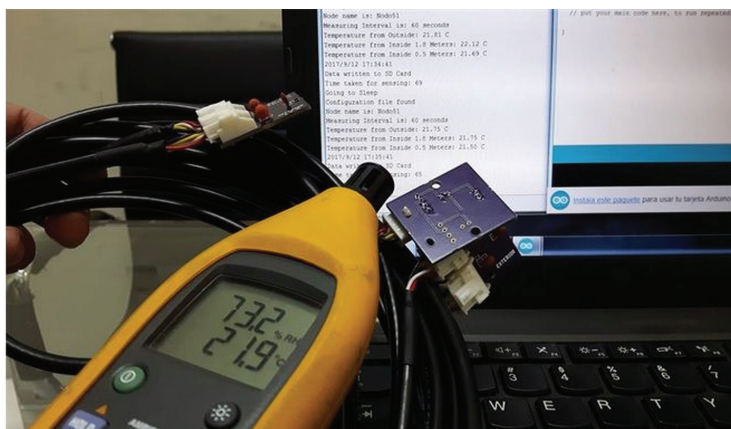


Figura 6. Validación de resultados de medición de sensores

Para almacenar los datos, se utilizó una tarjeta micro SD para ahorrar espacio y permitir su extracción y lectura como memoria externa en una computadora. Para este proyecto, la capacidad mínima de las tarjetas de memoria era de 4 GB. Sin embargo, solo se requiere una capacidad de memoria de 1 MB para los datos recopilados en un mes con una velocidad de muestreo de un minuto.

Finalmente, en la figura 7 se muestra el diagrama general del sistema desarrollado con todos los componentes y las conexiones entre ellos.

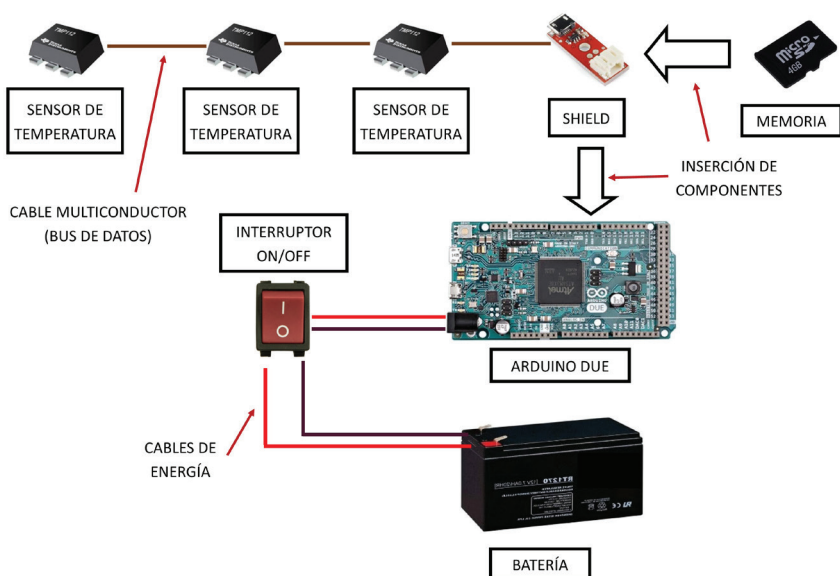


Figura 7. Esquema general del módulo desarrollado

### 3. INSTALACIÓN DEL SISTEMA DE MEDICIÓN

El *datalogger* desarrollado fue instalado en 54 casas que habían sido equipadas previamente con los muros Trombe. Estas casas estaban ubicadas en 18 comunidades en la región sur del Perú. Se eligieron las ubicaciones de los sensores según lo indicado en el apartado 2.1 (Requisitos) de este artículo.

Las zonas de instalación son áreas rurales sin electricidad, saneamiento, ni servicios de telecomunicaciones. Y en la mayoría de los casos, son áreas de difícil acceso (ver figura 8). Además, los pobladores en general solo hablan el idioma quechua.



Figura 8. Instalación del sistema de medición de temperatura

### 3.1 Tratamiento de los datos

Los datos se guardan en formato CSV para facilitar su visualización y gestión mediante un editor de hojas de datos. La figura 9 muestra los registros de medición durante un período de 24 horas, y se puede verificar que los valores de temperatura son coherentes. La totalidad de los datos recogidos son almacenados por Foncodes, cuyos funcionarios viajan periódicamente para cumplir la tarea de recopilar esta información. Finalmente, debe indicarse que estos registros están tabulados y resumidos.

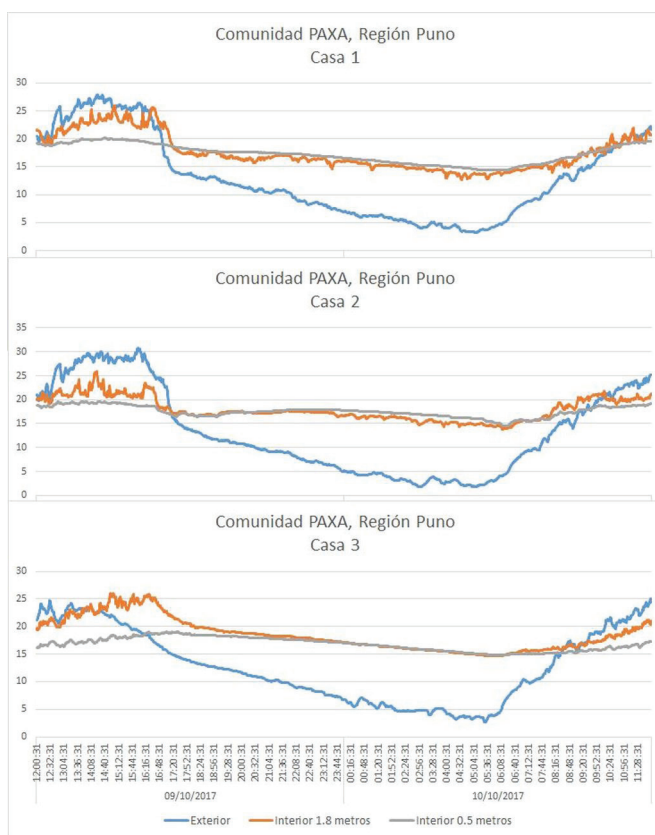


Figura 9. Registro de medición (24 horas) en tres casas

## 4. CONCLUSIONES

Este documento presenta la descripción y el proceso de implementación de un *shield* para el Arduino Due, para que trabaje como un *datalogger* de bajo consumo de energía para medición de temperatura. Este sistema se instaló en condiciones geográficas extremas

y se pudo comprobar su correcto funcionamiento (lecturas estables y consistentes) a pesar de las condiciones ambientales difíciles. El *datalogger* es empleado actualmente por el Gobierno para recopilar datos de temperatura. Además, se verificó también que los sensores de temperatura han medido valores negativos según lo requerido por Foncodes (el rango completo es de -45 a 12 °C).

Por otro lado, la implementación exitosa del sistema de medición de temperatura ha demostrado que el protocolo I2C puede usarse sobre cables relativamente largos, validando también el cálculo realizado para encontrar un valor adecuado para las resistencias *pull-up*. Esto abre la puerta a aplicaciones que pueden emplear el protocolo I2C en condiciones no estándar.

Este sistema de medición se utiliza actualmente en el proyecto Mi Abrigo para verificar que las casas acondicionadas con muros Trombe mejoran las condiciones de vida de las personas en zonas rurales aisladas de los Andes peruanos.

Finalmente, como un resultado colateral, se considera necesario indicar que, gracias al apoyo del Gobierno, se ha demostrado que es posible desarrollar soluciones confiables a medida y de bajo costo para cubrir necesidades específicas del sector público. Además, se ha probado que no en todos los casos la solución técnica pasa, necesariamente, por importar tecnología.

## REFERENCIAS

- Ana, B., Anabela, P., e Ivo, P. (2016). Dynamic Simulation of the Trombe Wall Thermal Performance. *IAHS World Conference*. Algarbe.
- Foncodes. (2017). *Ministerio de Desarrollo e Inclusión Social. Proyecto Mi Abrigo*. Recuperado de <http://www.foncodes.gob.pe/portal/index.php/proyectos/miabrigo>
- General Cable/ Carol Brand. (2018). *4 Wire Multi-Conductor Foild Shield Computer Cable*. Recuperado de <https://www.generalcable.com/assets/documents/LATAM%20Documents/Mexico%20Site/Cat%C3%A1logos/Electronics.pdf?ext=.pd>
- LabJack Measurement & Automation. (2017). *Digit Datasheet*. Recuperado de <https://labjack.com/support/datasheets/digit>
- National Renewable Energy Laboratory. (2017). *NREL*. Recuperado de U.S. Department of Energy: <https://www.nrel.gov/docs/legosti/fy98/22834.pdf>
- NXP Semiconductors. (2014). *I2C Bus Specification and User Manual*. Recuperado de <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>
- Sousa, J. (2017). *I2C Network for Home Automation*. Portugal: University of Porto.

Texas Instruments. (2018). *Digital Temperature Sensors With SMBus and TwoWire Serial Interface in SOT563*. Dallas.

Truchsess, W. (2010). *Effects on Varying I2C Pull Up Resistors*. Recuperado de [https://www.thebackshed.com/forum/uploads/BobD/2013-01-26\\_175209\\_I2C\\_-\\_Effects\\_of\\_Varying\\_Pullup\\_Resistors.pdf](https://www.thebackshed.com/forum/uploads/BobD/2013-01-26_175209_I2C_-_Effects_of_Varying_Pullup_Resistors.pdf)