Temperature Measurement System

USER MANUAL

Daniel Estrada, Santiago Osorio.

April, 2024

UNIVERSIDAD DE ANTIOQUIA 1803

GICM Grupo de Instrumentación Científica y microelectrónica

CIDEMAT Centro de Investigación, Innovación y Desarrollo de Materiales

TABLE OF CONTENTS

1. USER MANUAL GENERALITIES

This project has been developed by members of the Scientific Instrumentation and Microelectronics Research Group **(GICM)** at the University of Antioquia for the Research, Innovation, and Materials Development Center **(CIDEMAT)**, aimed at supporting research in the field of solar cells.

1.1. LICENSE

The software is provided under a commercial source code license, allowing non-exclusive and non-transferable rights to use, modify, and distribute it for research, development, and educational purposes. However, commercial redistribution requires a separate license agreement. Redistribution must include a copy of the license and maintain all copyright notices.

COMMERCIAL LICENCE

Software and Firmware Version: 1.0.0

Manual Edition: 1.0

Editor: Valentina Franco Velásquez.

1.2. CONTACT

If you have any questions, suggestions, or technical issues related to this interface, feel free to contact us via GitHub. There, you can open an issue and the solution will be assigned to any of the members of the group as soon as possible. **NOTE:** You can also contact us via email at grupo.instrumentacym@udea.edu.co.

https://github.com/GICM-UdeA/TMS-public

2. WHAT DOES THE SYSTEM CONSIST OF?

The Temperature Measurement System **(TMS)** consists of a Data Acquisition Device **(DAQ)** based on Arduino and a Graphical User Interface (**GUI**) for control based on Python. The embedded system or hardware monitors are a set of six type K thermocouples at a configurable sampling rate. The records are stored in memory and sent to the user via serial communication through the desktop GUI (*see Figure 1*).

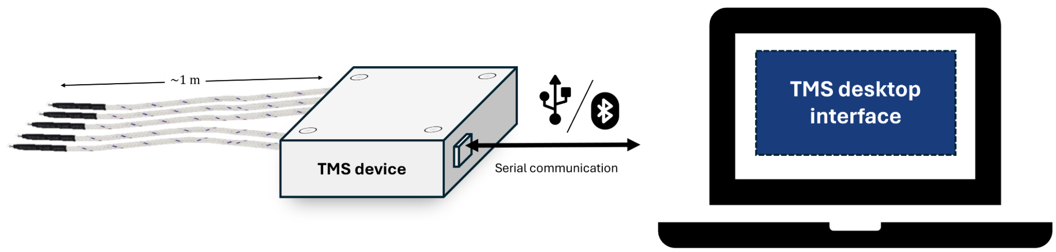

Figure 1. Illustrative representation of the TMS setup.

The firmware operates on a command-based protocol, enabling users to configure measurement parameters such as the "sampling rate" and the "analysis time" (detailed instructions on these functionalities can be found in section 3). This protocol is used by the GUI allowing users to perform these operations and obtain and visualize real-time data.

NOTE: More information regarding how to use the interface will be provided in the next section.

1. Male DB15 connector coupled to six type K thermocouples.
2. Female DB15 connector to connect thermocouples to the TMS device.
3. 5V DC power supply connector.
4. ON/OFF switch.
5. USB-mini connector.
6. LED indicator – 10 Hz flashing indicates that the system is ON and Bluetooth is disconnected, ~1 Hz flashing indicates that Bluetooth is connected.

3. HOW TO USE THE TMS?

- **1ST STEP: PLACEMENT OF THE SENSORS**

  The first step is to locate the six thermocouples in the desired measuring location. These have been designed through a DB15 connector to facilitate their connection to the system. Each one is distinguished by a specific color ranging from T1 to T6. Please refer to *Figure 2* to visualize the connector configuration and the adopted color code.

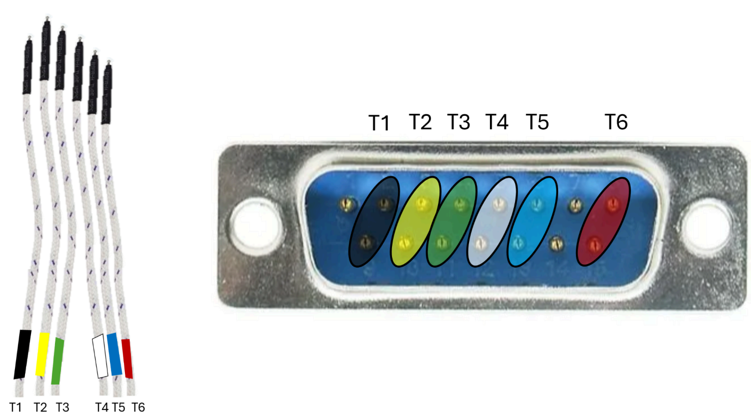

Figure 2. Color code adopted and connector pinout for the TMS.

- **2ND STEP: CONNECTION OF THE SYSTEM**

TMS can be connected directly to the GUI's host computer via USB-mini or Bluetooth. By default, Bluetooth mode is enabled. To change this, you should open the device and turn off the dip switch located on the **PCB** (Printed Circuit Board) as shown in Figure *3*. This is necessary because serial USB and serial Bluetooth could interfere with each other.

If you connect via Bluetooth, you should pair the device "*TMS-DEVICE*" with the password "*1234*". Then, you can use the virtual COM port assigned by your operating system to the device in the connection panel of the interface (*see Figure 3*). If the connection is not via Bluetooth, simply select the correct COM port of the device.

Finally, press the connect toggle button located on the bottom-right side of the GUI and wait for the confirmation message to appear on the serial monitor.

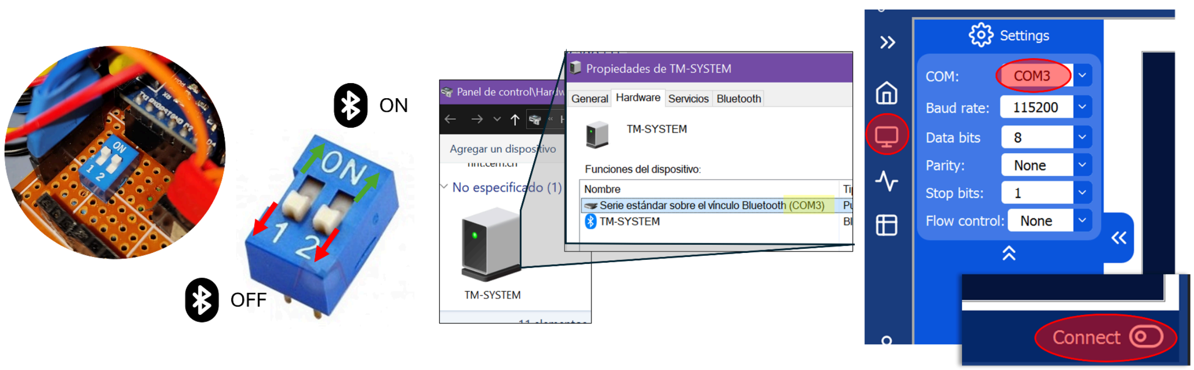

Figure 3. Dip switch to enable Bluetooth (*to the left*) and illustration of connection to the virtual COM assigned by Windows (*to the right*).

- **3RD STEP: START DATA STREAMING**

Once your device is properly connected, you can begin collecting data by pressing the "START" button located at the bottom of the control panel window (see Figure 4). The GUI will first send some configuration commands and then it will start retrieving data from the device at a predefined "sampling rate" during the “analysis time” that was set.

Note: Data is recorded by Arduino at the “sampling rate,” but the visualization on the graphs is done according to the periodic execution (every 10 ms) of some recurring routines.

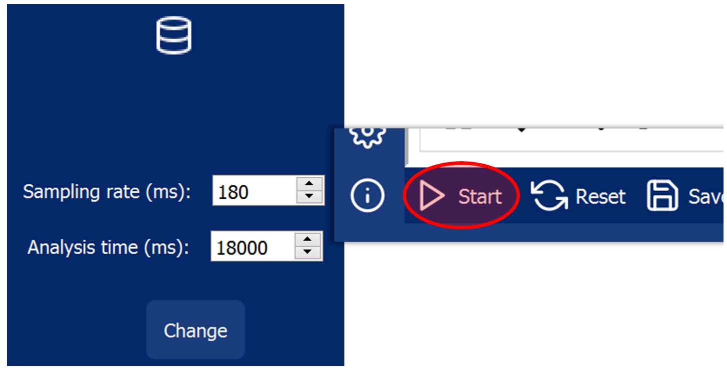

Figure 4. Menu to set the streaming parameters and a mark showing the position of the "*START*" button.

- **4TH STEP: SAVE RESULTS**

  Once your data-taking process is complete, you can save the results by clicking the "*SAVE*" button, also located in the control panel. This action will generate a .xlsx file and two .png images (a linear and an Imshow map time series of the temperatures registered by the thermocouples). The default path to save the results can be modified via the "*settings*" window, and user information included in the .xlsx file can also be configured via the "*user*" window or the "*settings*" window.

  Finally, you can reset the database by clicking the "*RESET*" button, after which the system will be ready to start again.

4. OVERVIEW THROUGH THE TMS'S GUI

To install the program you just have to:

I. Access to the repository on GitHub through the following link: https://github.com/GICM-UdeA/TMS-public.
II. Click on the file called "*TMS-_windows_v1.0_setup.exe*".
III. Click on the top right button called "*Download Raw File*". **NOTE:** Wait a while for the download of the executable file to start.
IV. Once the previous steps are completed, the license agreements will open. It is recommended to read them carefully. Once read, select "*I accept the agreement*" and then click "*Next*". In *Figure 5*, the corresponding options are indicated.

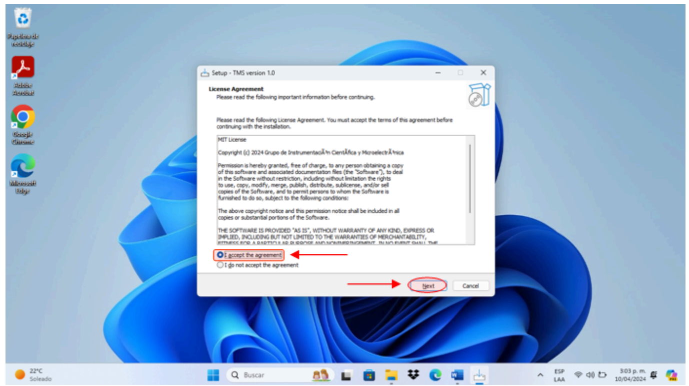

Figure 5. License agreement.

V. Select "*Create a desktop shortcut*" and click "*Next*". The fields to select are shown in *Figure 6*.

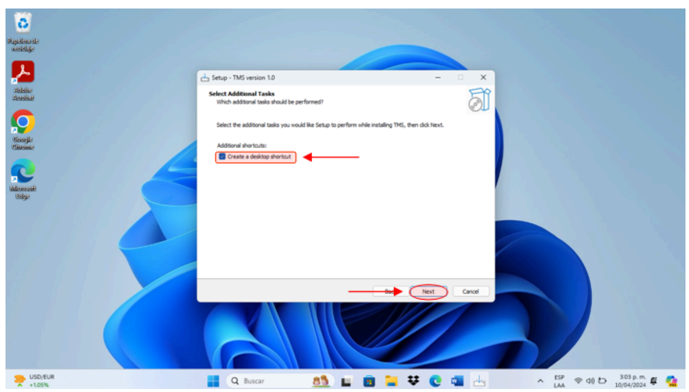

Figure 6. Additional tasks.

VI. Next, click on "*Install*" to begin the installation of the application, as shown in *Figure 7*.

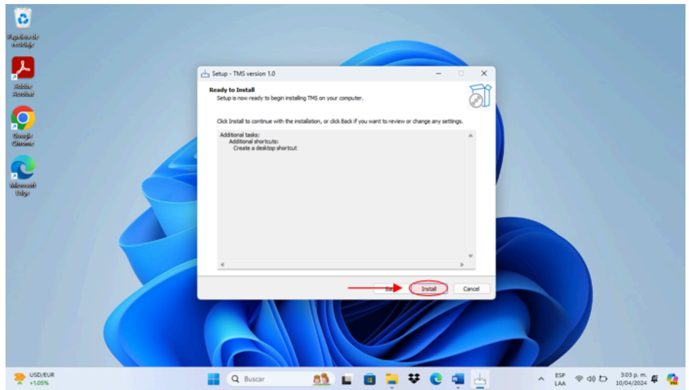

Figure 7. Installation of the executable file.

VII. Finally, as soon as the download is complete, select "*Launch TMS*" and then click the button "*Finish*" to run the application as shown in *Figure 8*.

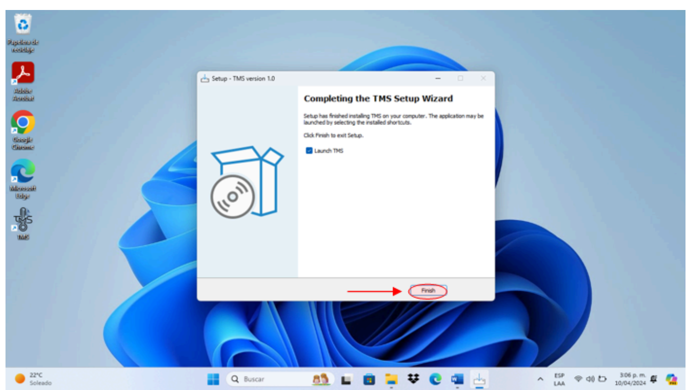

Figure 8. Concluding the installation process.

When opening the application, it will load into the initial section where a brief description of the TMS is located. Additionally, there is a QR code pathway to access the repository on GitHub where information about the application can be found (*see Figure 9*).

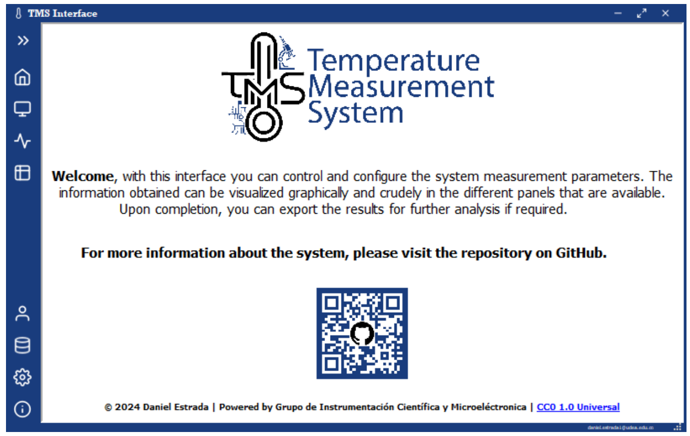

Figure 9. Home of the TMS application.

Through the top right menu, you can observe four display options allowing for application management. As seen in *Figure 10*, the home checkbox takes us to the start of the application.

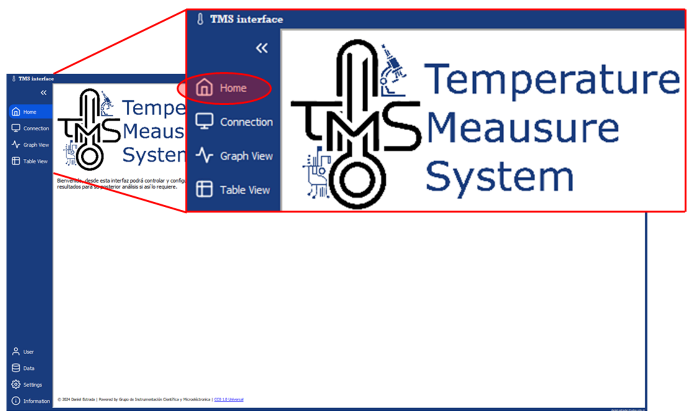

Figure 10. Control Panel.

Once the connection has been established and the program has been run, if we go to the "*graph viewer*" checkbox, we can observe the system monitor temperature readings.

In *Figure 11*, it is shown how, through the map option, one can observe the data acquisition in the form of a heat map, which describes the behavior of the system.

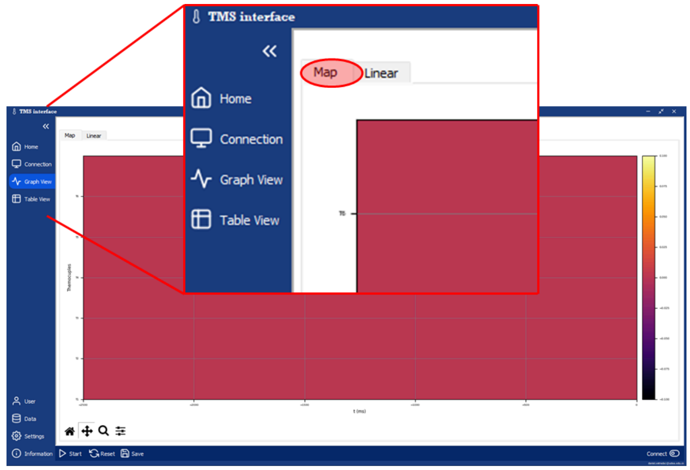

Figure 11. Graph view, heat map.

The application also allows for tracking individual temperature variations over time. As can be seen in *Figure 12*.

Figure 12. Graph view, temperature diagram.

The "*Table Viewer*" option displays tabulated data recorded during the selected measurement interval. This table captures the temperature data taken by the six thermocouples in Celsius degrees and the corresponding times in milliseconds as observed in *Figure 13*.

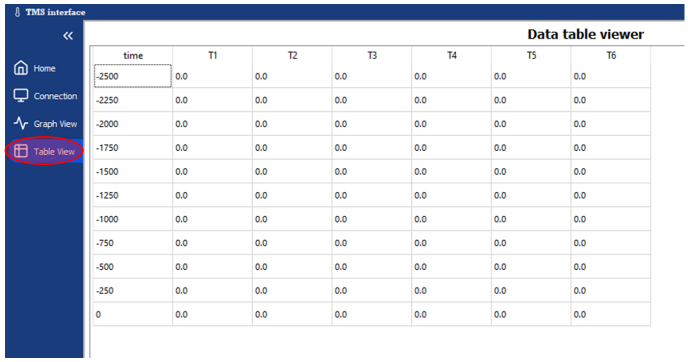

Figure 13. Resulting table.

5. OVERVIEW THROUGH THE TMS'S COMMANDS

Serial communication with the system is achieved by a set of predefined commands:

- **SETS XXX**

  This command sets the sampling time. XXX is an integer not less than the lower bound (~240 ms, see specifications section for more information). If the change is accepted, the system will respond with **SSOK**; otherwise, it returns **SSNOK**.

- **SETA XXX**

  This command sets the analysis time. XXX represents the analysis time in milliseconds and cannot be lower than the sampling time. If the change is accepted, the system responds with **SAOK**; otherwise, it returns **SANOK**.

- **START**

  This command starts the measurement process. The system responds with **STAOK**.

- **STOP**

  This command stops the measurement process. The system responds with **STOOK**.

6. TECHNICAL SPECIFICATIONS

6.1. SENSORS

The length of the thermocouples is approximately 1 meter, and each one has a minimum response time of around 170 ms. According to the manufacturer of the MAX6675 module, which is used to amplify and convert the electrical signal to a digital value of temperature. The accuracy of the measurements corresponds to 0.25 °C (for more information, refer to the datasheet).

6.2. MEASUREMENTS

Given the Arduino-based nature of the device, it is not possible to register measurements from all six thermocouples simultaneously. Therefore, the maximum optimization achievable is to read the sensors consecutively as quickly as possible. *Figure 14* illustrates this situation.

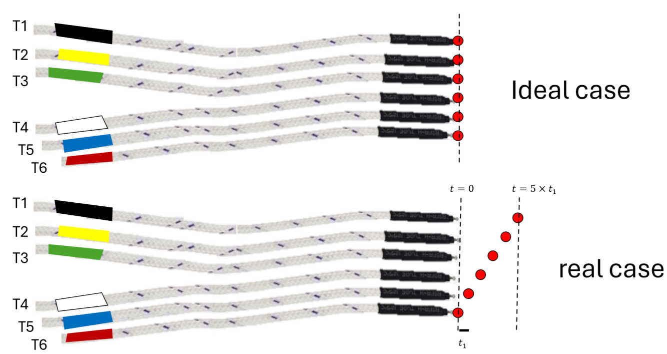

Figure 14. Schematic representation of the shift in measurements due to the impossibility of reading all six sensors simultaneously. t1 represents the time between the measurement of one thermocouple and another, so approximately 5 times t1 is required to read all six sensors.

t1 was estimated to have a time of around 20 μs. Therefore, it is recommended not to use a minimum sampling time (for all six sensor registers) lower than 180 ms. This is to prevent inaccurate readings of the sensor due to the time spent on numerical conversion and 'for' loop operations.