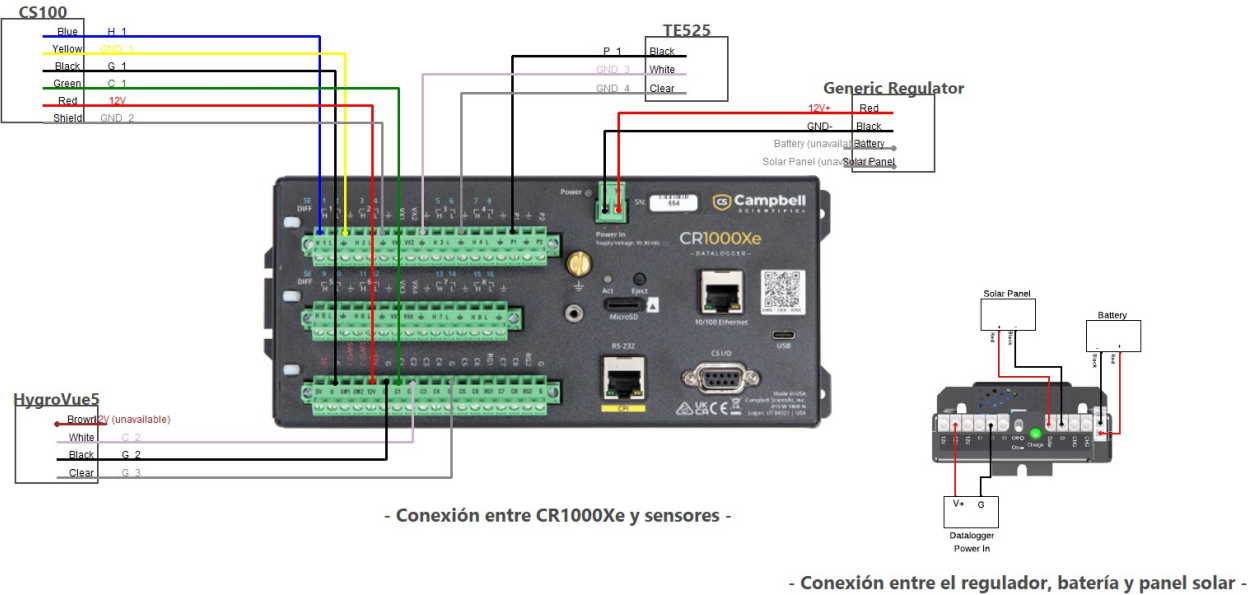


Welcome to the CS_AutoWiring Wiki!

CS Wiring Tool is an application used to generate connection diagrams between Campbell Scientific dataloggers and sensors.



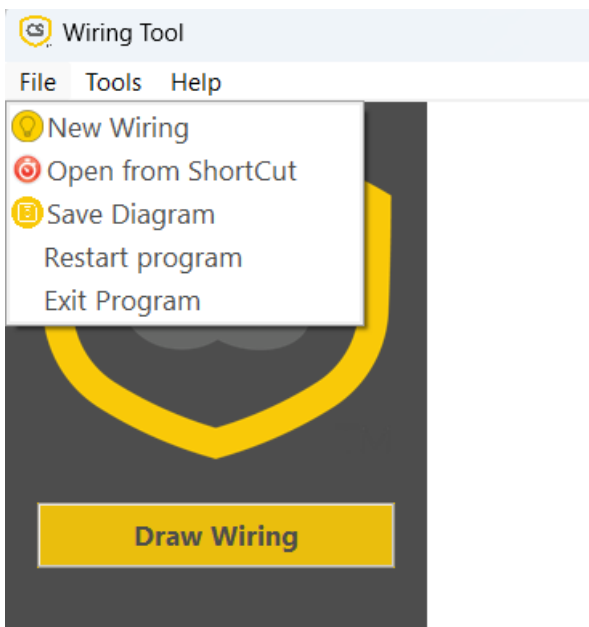
Company:	<input type="text" value="Company Name"/>	Content:	<input type="text" value="Sensor Wiring Diagram"/>
		Project:	<input type="text" value="Weather Station CR1000X"/>
		Date:	<input type="text" value="2025-09-18"/>
		Made by:	<input type="text" value="Campbell Scientific"/>

Initial development by Natalia González Bermúdez

USER MANUAL

New Diagram

1. To create a new diagram, go to **File** in the menu bar.




2. There are two possible options to select the wiring source:

- **New Wiring:**

Choose this option to create a new diagram and manually select the sensors and protocols.

When you select "New Wiring," a window will appear where you can choose the datalogger and sensors.

- First, select the datalogger.
- Then, add sensors using the **Add Selection +** button.
- To remove a sensor, select it from the white box on the right and click **Delete Selected -**.
- Once all sensors are added, click **Save & Close** to store the configuration.

 Select Datalogger and Sensor

Select Datalogger:

img/cr1000x.png ▼

Filter by Measurement:

All ▼

Select Sensor:

▼


Select Protocol:

▼

Add Selection +

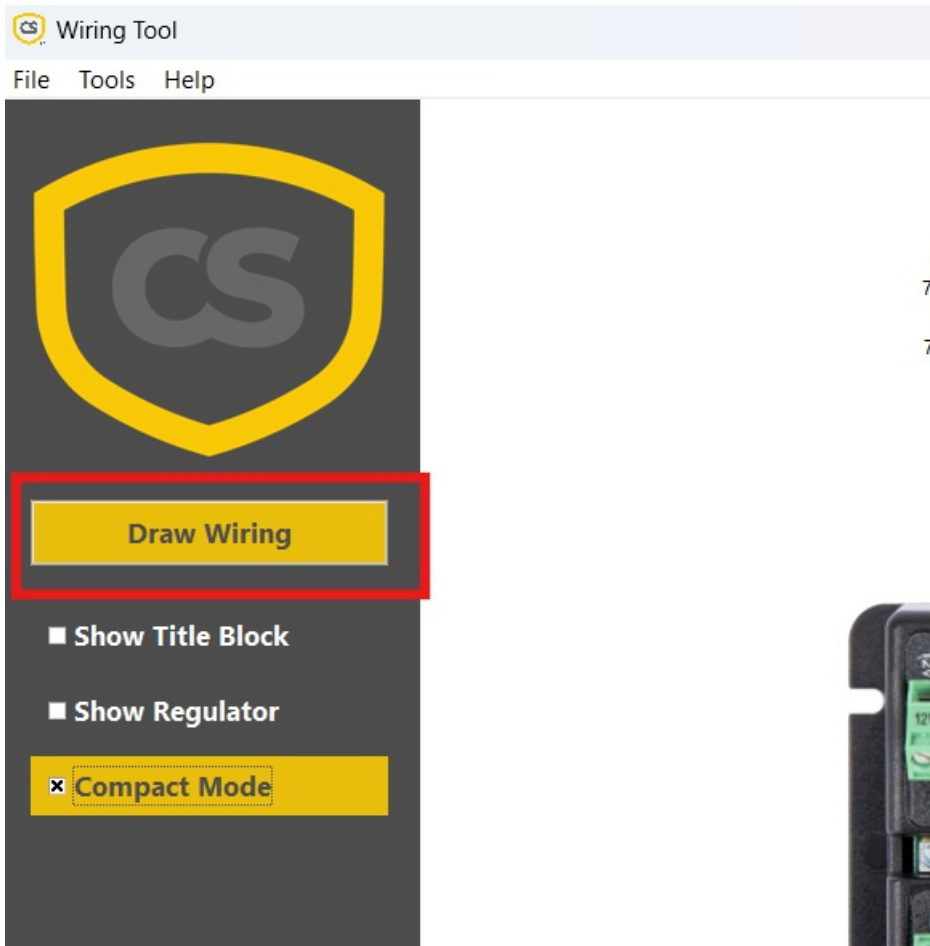
Delete Selected -

Save & Close

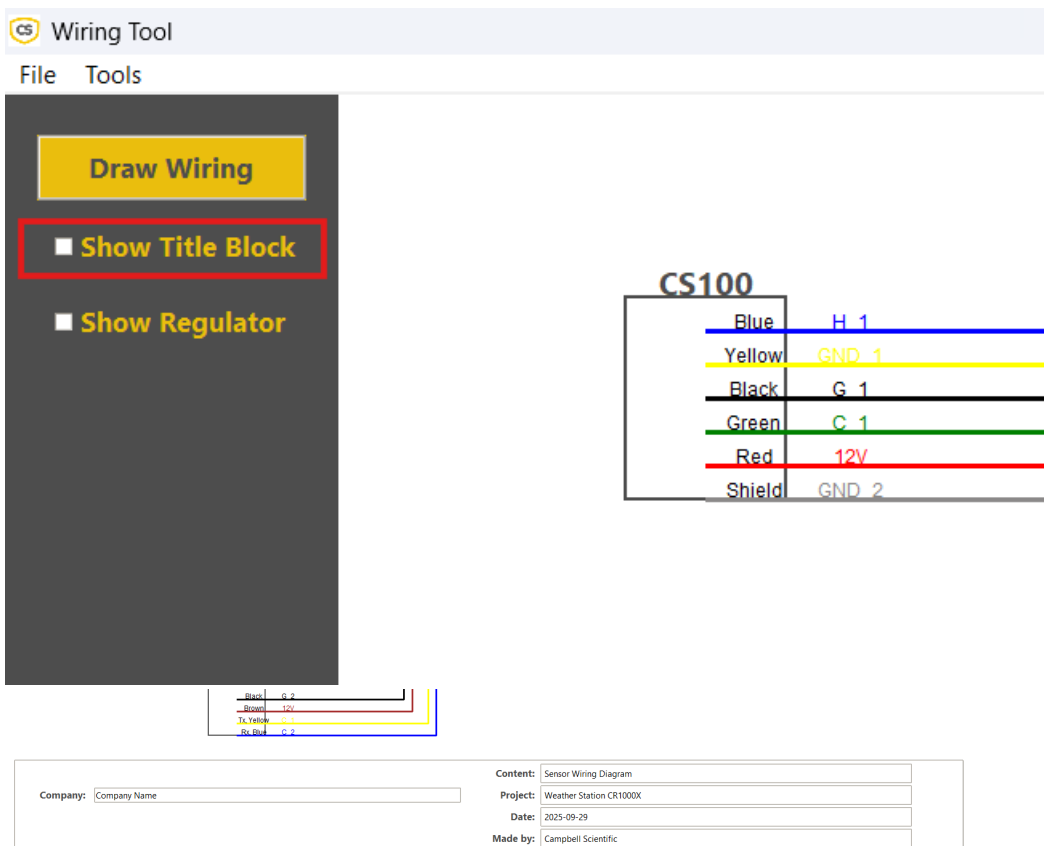


Selections:

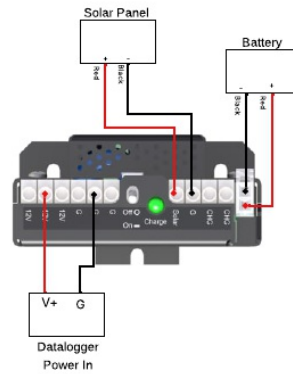
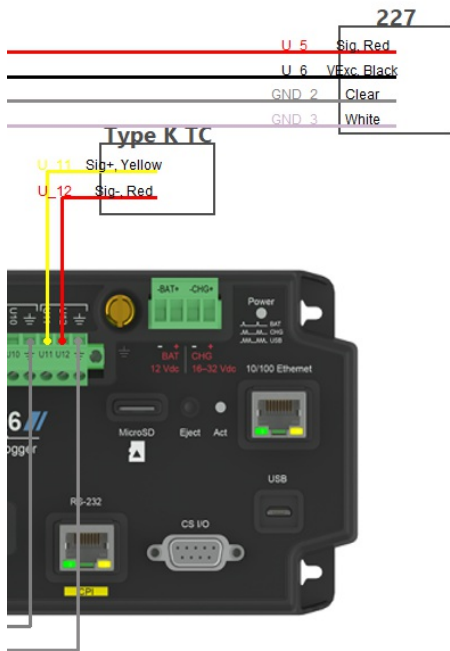
- Open from ShortCut:**
 Choose this option if you want to create a diagram from files generated by Campbell Scientific's ShortCut software.
 Select a `.DEF` file.
3. After selecting the wiring source, click **Draw Wiring** to generate the diagram showing the datalogger and selected sensors.



4. To include a title block in the diagram, check the **Show Title Block** checkbox.



5. To include a wiring diagram for a regulator, solar panel, and battery, check the **Show Regulator** checkbox.



Content: Sensor Wiring Diagram

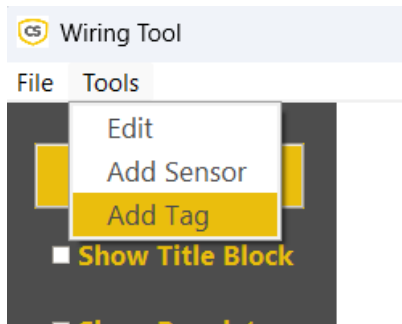
Project: Weather Station CR1000X

Date: 2025-09-29

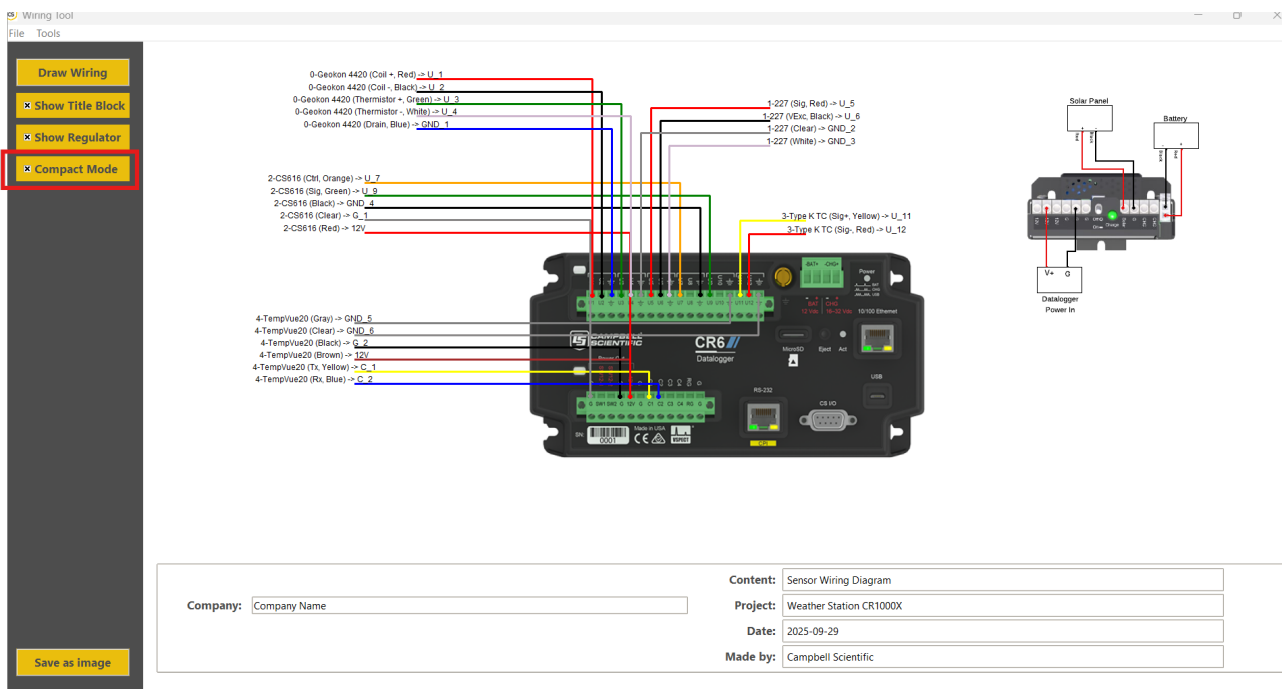
Made by: Campbell Scientific

6. To save the diagram as an image file (.png), click the **Save as Image** button.

7. To add extra labels, go to **Tools > Add Tag**.
These labels can be dragged to reposition them and removed with a right-click.

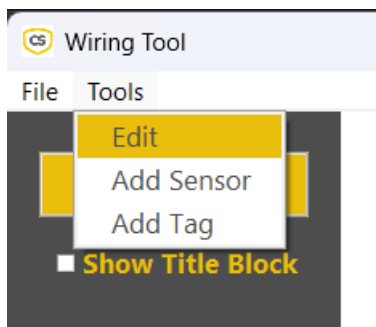


8. If there are many sensors and you need a more compact layout, enable **Compact Mode**.
This mode displays only the wire descriptions for a simplified view.



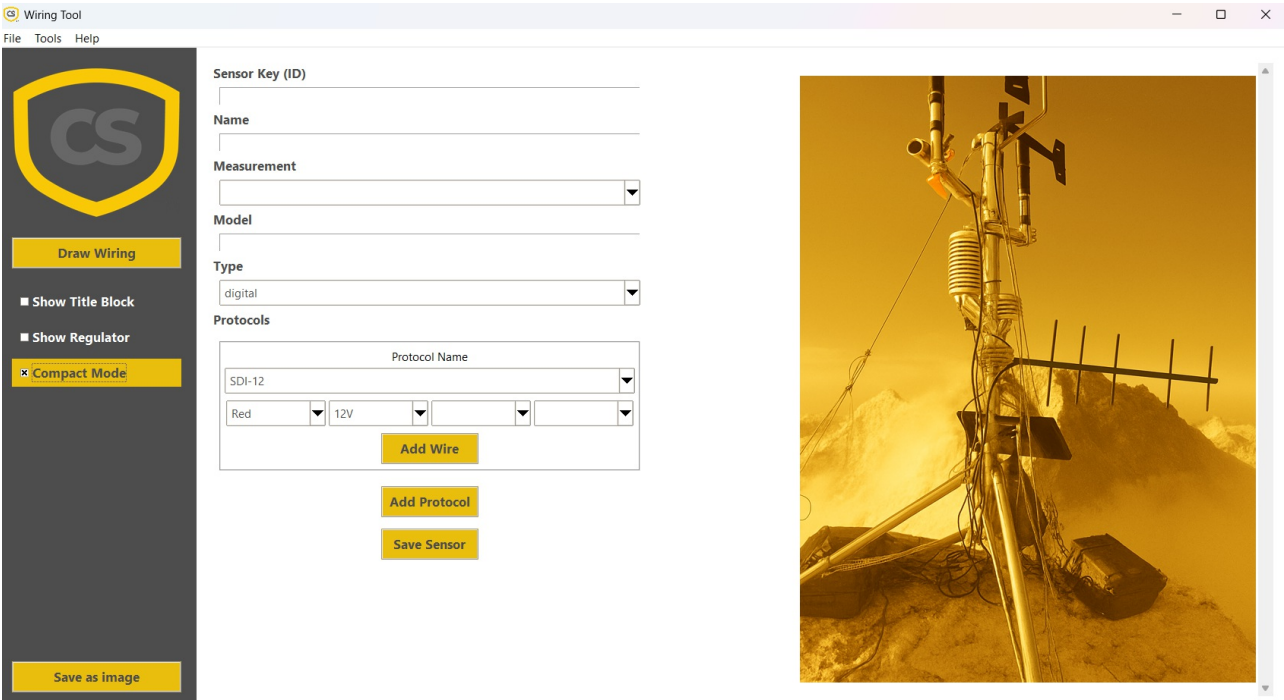
Edit Diagram

If you need to change a wire connection to a different port, go to **Tools > Edit** in the menu bar. Port names follow the convention **PortType_PortNumber**. Click [here](#) to view the datalogger pin naming conventions.

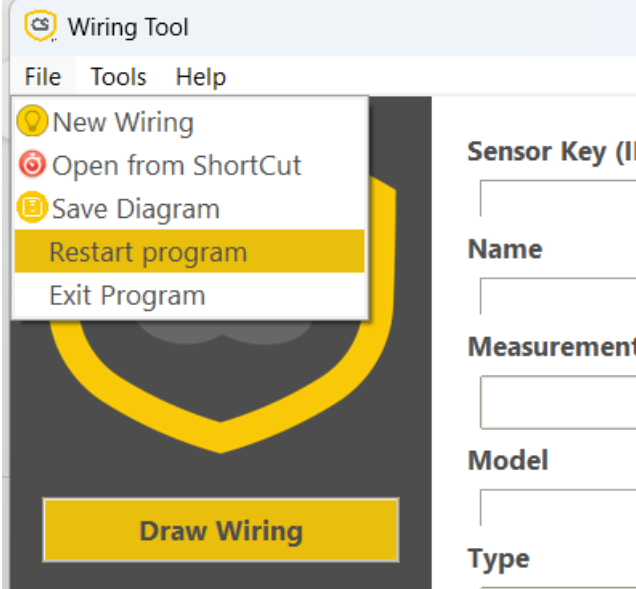


Add New Sensor

To add a new sensor, go to **Tools > Add Sensor**. For each sensor, you can define one or more connection protocols with the datalogger, and for each protocol, specify the wire colors and corresponding ports.



To update the list of sensors, restart the program via **File > Restart Program**.



Manual Wire Drawing Mode (Wire Mode)

This mode allows you to manually draw wires on the diagram with 90° angles, choose wire colors, move sensors, and delete wires easily.

☒ Quick Summary

- W → Toggle wire drawing mode
- Left-click → Add a point (wire mode only)
- Enter → Finalize current wire
- Esc → Cancel current wire & exit wire mode
- Right-click → Delete a wire (when wire mode is OFF)

☒ Wire Mode Controls

☒ Activate/Deactivate Wire Mode

Action	Key	Description

Action	Key	Description
Toggle wire drawing mode	W	Activates or deactivates wire mode. When active, sensor dragging is disabled.

🗨 Color Picker

- The color picker appears in a **floating TopLevel window**.
- It always stays on top of the main interface.
- You can change the wire color **before or during** drawing.
- The selected color applies to the next wire you draw.

🖋 Draw a Wire

1. Press W → **Wire Mode: ON**
2. Left-click on the canvas to place points.
3. The cable will automatically draw with **90° (horizontal → vertical)** routing.
4. Add as many points as you need.

✓ Finalize a Wire

Action	Key	Result
Finalize the current wire	Enter	Converts the preview into a permanent wire. Wire mode remains active so you can draw more.

🗨 Cancel the Wire in Progress

Action	Key	Result
Cancel and exit wire mode	Esc	Clears the preview and exits wire drawing mode.

🗨 Delete a Wire

When NOT in wire mode:

1. Right-click on any manually drawn wire.
2. The entire wire will be deleted.

Note: When in `Wire Mode: ON` , a right-click cancels the in-progress wire and exits wire mode.

Move Sensors

- Sensors can only be moved when **Wire Mode is OFF**.
- Left-click and drag a sensor to reposition it.

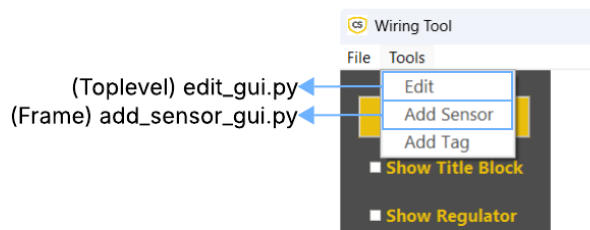
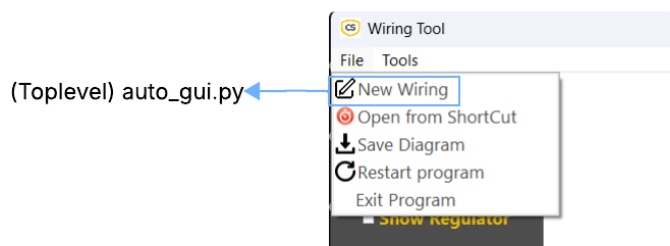
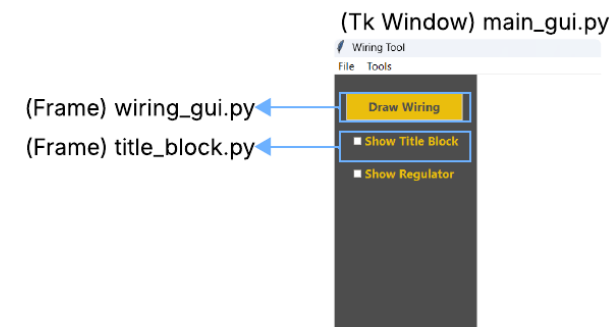
PROGRAMMER MANUAL

Requerimientos de instalación

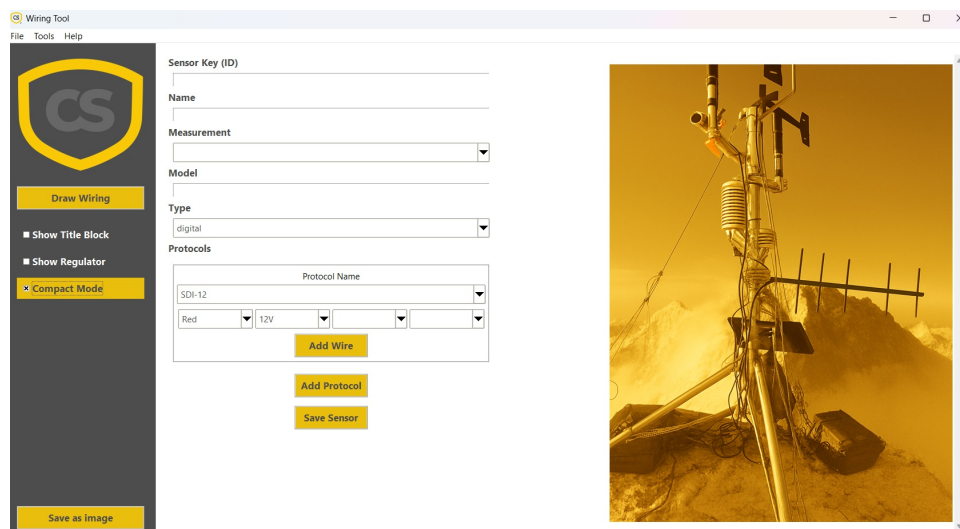
- Python 3
- Bibliotecas externas: PIL, pyautogui

Interfaz gráfica

El repositorio tiene los siguientes archivos de interfaz gráfica. El principal es main_gui.py, a partir de esta se llaman a los diferentes frames o ventanas toplevels de la siguiente forma:



- add_sensor_gui.py:



- auto_gui.py:

Select Datalogger and Sensor

Select Datalogger:

img/cr1000x.png

Filter by Measurement:

All

Select Sensor:


Select Protocol:

Selections:

Add Selection +

Delete Selected -

Save & Close



• edit_gui.py:

Wiring Tool

FileToolsHelp

CS

Draw Wiring

Show Title Block

Show Regulator

Compact Mode

Save as Image

Select Sensor:

B-Climavue50 G2


U,1White

G,1Clear

12VBrown

G,2Black

Save Changes



• title_block.py:

Company:

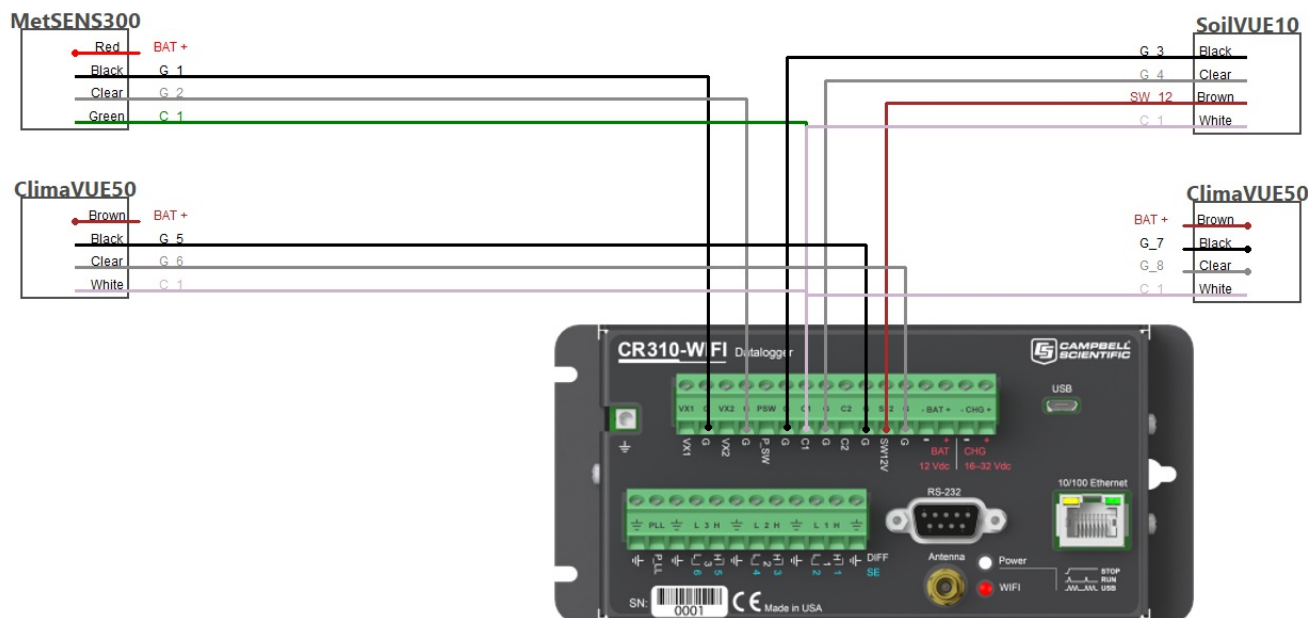
Content:

Project:

Date:

Made by:

• wiring_gui.py:



Agregar un nuevo datalogger o dispositivo principal

Si se quiere agregar un nuevo dispositivo, ya sea un datalogger, o un multiplexor, se deben realizar por medio de código, siguiendo los siguientes pasos:

1. Agregar la imagen en la carpeta img
2. Mapear los puertos de la imagen. Las coordenadas de cada puerto se deben agregar en el archivo `ports_coordinates.py` y son relativas a la imagen. Se puede usar el archivo `coordinates.py` para obtener las coordenadas del puerto y se copiar automáticamente haciendo click derecho.
3. Agregar el dispositivo al diccionario. Esto se agrega a dataloggers en `dictionary.py`

Lógica

En cuanto a lógica, se encuentra en el archivo `wiring_functions.py` que contiene las funciones que retornan el cableado que se va a dibujar utilizando el Frame creado con `wiring_gui.py`

wiring_functions.py

En este archivo hay 2 funciones principales, `get_wiring_from_SC` y `get_auto_wiring`

get_wiring_from_SC(filename)

Esta función convierte los datos tomados de un archivo autogenerado por ShortCut a datos que pueden ser graficados utilizando el `WiringFrame`.

Input:

- filename: un string con la ruta del archivo .DEF auto generado por ShortCut.

Output: [wiring, image_name]

- wiring: Son los datos que pueden ser interpretados por el `WiringFrame`. Tiene la siguiente forma: `{ sensor_name(1):{ datalogger_pin_name(1) : color, datalogger_pin_name(2) : color,...}, sensor_name(2):{ datalogger_pin_name(1) : color, datalogger_pin_name(2) : color,...} }`
- image_name: El nombre de la imagen del datalogger que se interpretó del archivo de ShortCut

get_auto_wiring(datalogger_ports, sensors)

Input:

- datalogger_ports: Son los puertos y coordenadas del datalogger. Se utilizan para asignar los puertos a los sensores conectados. Se toman del archivo `ports_coordinates.py`. Tienen la siguiente forma: `{"image": (width, lenght), "datalogger_pin_name(1)": (x, y), "datalogger_pin_name(2)": (x, y),...}` Los posibles `datalogger_pin_name` se pueden ver [aquí](#).
- sensors: Son los sensores de los que se quiere realizar el diagrama. En el programa este objeto viene como resultado del Frame de `auto_gui.py`. Tienen la siguiente forma: `{ sensor_name(1):{ color(1) : (generic_pin(1), generic_pin(2), generic_pin(3), color(2) : (generic_pin(1), generic_pin(2), generic_pin(3)...} sensor_name(2):{ color(1) : (generic_pin(1), generic_pin(2), generic_pin(3), color(2) : (generic_pin(1), generic_pin(2), generic_pin(3)...} }` Los `generic_pin` se refiere a todos los posibles tipos de pin a los que se podría conectar cada cable, pueden ser: C, VX, H, L, P, etc.

Output: Esta función retorna una lista en la cual el primer elemento es el objeto del cableado y el segundo elemento es el nombre de la imagen. Tiene la siguiente forma: `{ sensor_name(1):{ datalogger_pin_name(1) : color(1), datalogger_pin_name(2) : color(2),...}, sensor_name(1):{ datalogger_pin_name(1) : color(1), datalogger_pin_name(2) : color(2),...} }` El parámetro `port_name` se refiere al nombre de los puertos indicados

en el archivo ports_coordinates.py. Se pueden ver [aquí](#).

Mejoras por hacer

- Al realizar un diagrama, se puede entrar a la pantalla en Tools->Edit. Cuando se edita el cableado de un sensor y se pasa a otro, no se guardan los datos a menos que se edite por medio del dropdown. Si se escribe el dato con el teclado se debe guardar cada vez que se cambia entre sensores.
- Cuando se agrega un sensor o dispositivo nuevo, se debe reiniciar la aplicación, esto se debe a que los imports se hace cuando inicia el main y si se cambia el archivo de dictionary.py no se va a ver reflejado en los sensores hasta que se reinicie.
- Cada vez que se edita el diagrama, se agrega un sensor, o se cambia a la configuración "Compact Mode" el diagrama vuelve a autogenerar las posiciones y no guarda las anteriores.
- Para guardar el diagrama se toma un screenshot del frame de wiring por lo que se debe ver el diagrama completo al guardar la imagen.

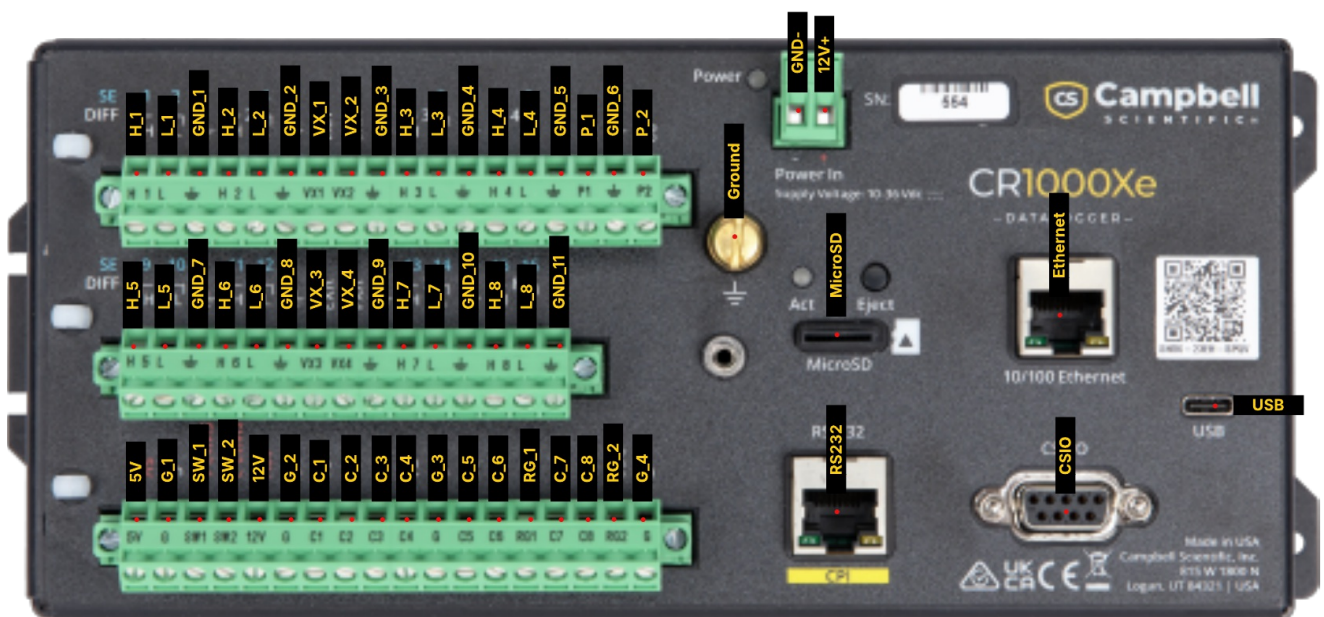
PIN NAMES

NOTE: You can use coordinates instead of pin name in format (x, y)

CR1000Xe

Pin Names in order:

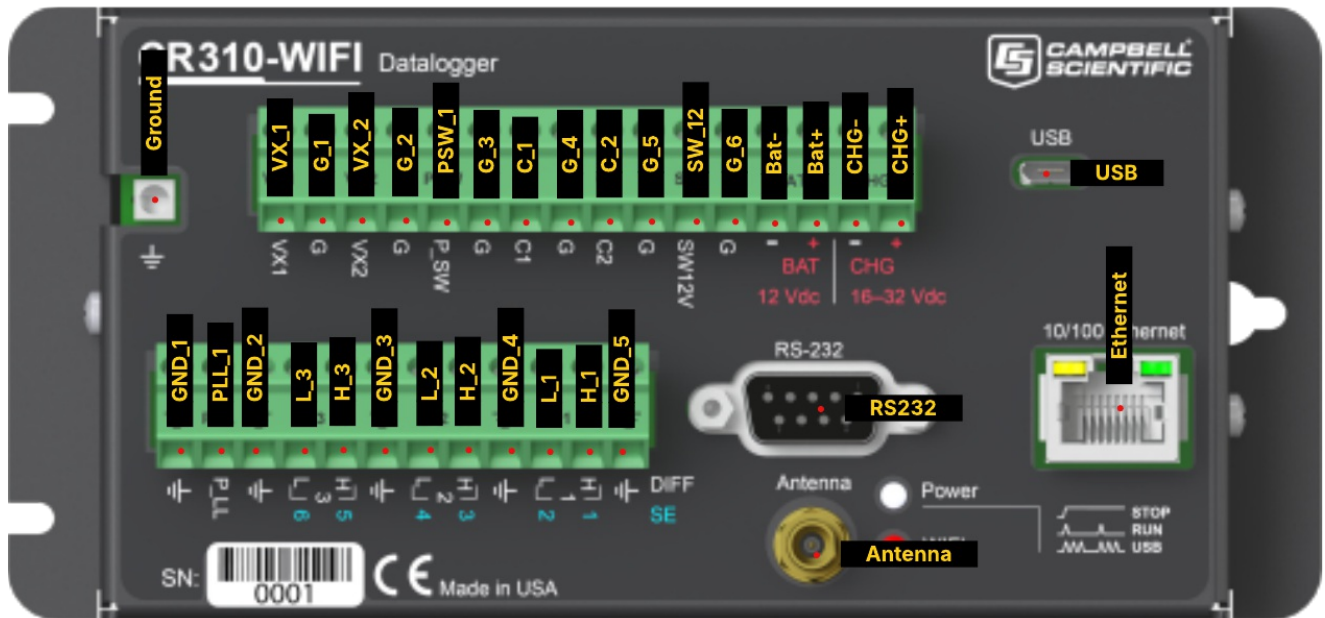
- Top connector: H_1, L_1, GND_1, H_2, L_2, GND_2, VX_1, VX_2, GND_3, H_3, L_3, GND_4, H_4, L_4, GND_5, P_1, GND_6, P_2
- Central connector: H_5, L_5, GND_7, H_6, L_6, GND_8, VX_3, VX_4, GND_9, H_7, L_7, GND_10, H_8, L_8, GND_11
- Bottom connector: 5V, G_1, SW_1, SW_2, 12V, G_2, C_1, C_2, C_3, C_4, C_5, C_6, RG_1, C_7, C_8, RG_2, G_4
- Power In Connector: GND-, 12V+
- Interfaces: Ground, MicroSD, Ethernet, USB, RS232, CSIO



CR310

Pin Names in order:

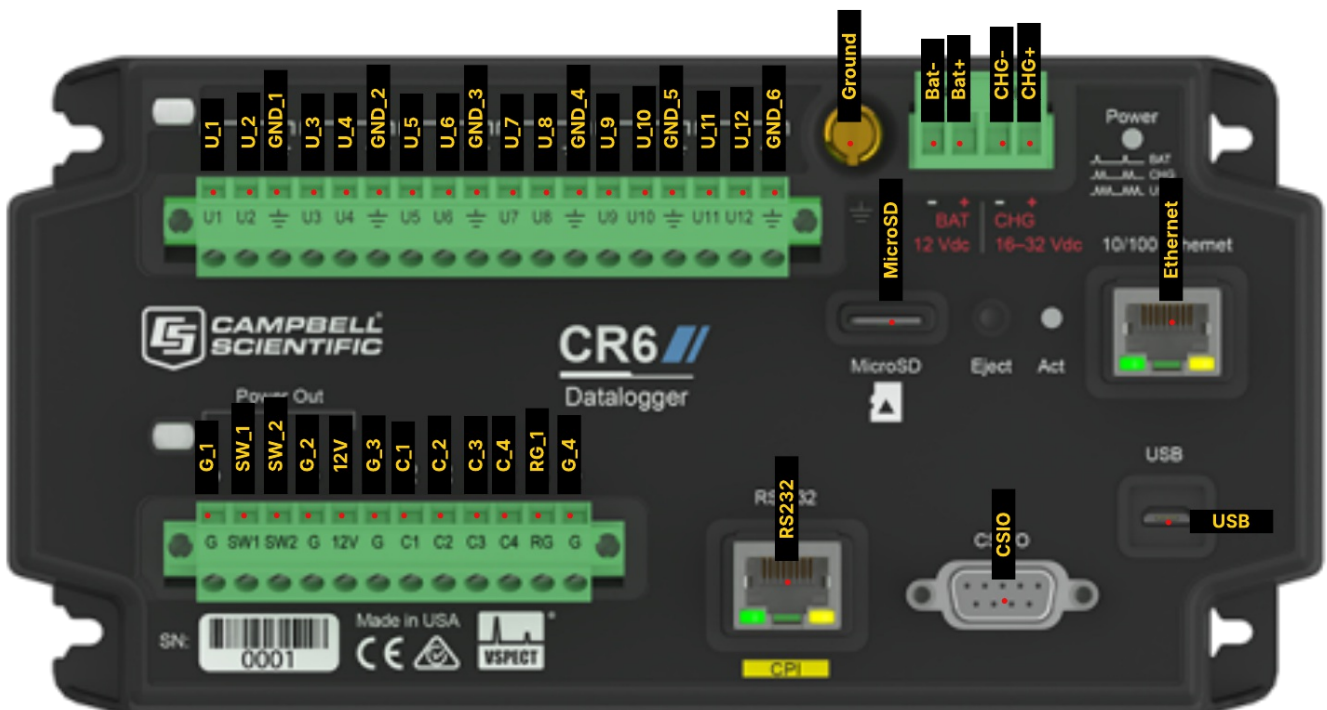
- Top Connector: VX_1, G_1, VX_2, G_2, PSW_1, G_3, C_1, G_4, C_2, G_5, SW_12, G_6, Bat-, Bat+, CHG-, CHG+
- Bottom connector: GND_1, PLL_1, GND_2, L_3, H_3, GND_3, L_2, H_2, GND_4, L_1, H_1, GND_5
- Interfaces: Ground, USB, RS232, Ethernet, Antenna



CR6

Pin Names in order:

- Top connector: U_1, U_2, GND_1, U_3, U_4, GND_2, U_5, U_6, GND_3, U_7, U_8, GND_4, U_9, U_10, GND_5, U_11, U_12, GND_6
- Bottom connector: G_1, SW_1, SW_2, G_2, 12V, G_3, C_1, C_2, C_3, C_4, RG_1, G_4
- Power In Connector: Bat-, Bat+, CHG-, CHG+
- Interfaces: Ground, MicroSD, Ethernet, RS232, CSIO, USB



CR350

Pin Names in order:

- Left connector: 12V, G_1, C_1, 12V_2, G_2, C_2, SW_1, G_3, TX_2, RX_2, RG_1, SW_2, G_4
- Right connector: TX_3, RX_3, PLL_1, PSW, GND_1, VX_1, H_1, L_1, GND_2, VX_2, H_2, L_2, GND_3
- Power In Connector: Bat-, Bat+, CHG-, CHG+
- Interfaces: Ground, RS232, USB, Antenna, WIFI



AM16/32B

Pin names in order:

- Top connector: H_1, L_1, GND_1, H_2, L_2, GND_2, H_3, L_3, GND_3, H_4, L_4, GND_4, H_5, L_5, GND_5, H_6, L_6, GND_6, H_7, L_7, GND_7, H_8, L_8, GND_8, H_9, L_9, GND_9, H_10, L_10, GND_10, H_11, L_11, GND_11, H_12, L_12, GND_12
- Middle left connector: RES, CLK, G_1, 12V_1
- Middle right connector: H_0, L_0, GND_0, H_E, L_E, GND_E, H_13, L_13, GND_13, H_14, L_14, GND_14, H_15, L_15, GND_15, H_16, L_16, GND_16, H_17, L_17, GND_17, H_18, L_18, GND_18, H_19, L_19, GND_19, H_20, L_20, GND_20
- Botton connector: H_21, L_21, GND_21, H_22, L_22, GND_22, H_23, L_23, GND_23, H_24, L_24, GND_24, H_25, L_25, GND_25, H_26, L_26, GND_26, H_27, L_27, GND_27, H_28, L_28, GND_28, H_29, L_29, GND_29, H_30, L_30, GND_30, H_31, L_31, GND_31, H_32, L_32, GND_32

