



sonnen ecoLinx

Energy Storage Driver

Compatible Control4 Systems:

Designed to work with OS2.10.x, OS3.x

Compatible Hardware:

[sonnen ecoLinx](#)

Driver Download, Change Log & Documentation:

Please visit <https://drivercentral.io/platforms/control4-drivers/utility/sonnen-ecolinx>

Content

- [Overview](#)
- [Features](#)
- [Dealer Notes](#)
- [License Purchase & Assignment](#)
- [Driver Setup](#)
 - [Driver Properties](#)

- [Composer Variables](#)
 - [Composer Events](#)
 - [Composer Commands](#)
 - [Notification Settings](#)
 - [Weather Service Integration](#)
 - [Email/Text Notification Settings](#)
 - [UI Settings](#)
 - [Support](#)
 - [Warranty & Disclaimer](#)
 - [Developer Information](#)
-

Overview

Integrates sonnen ecoLinx batteries to Control4. sonnen is a manufacture of high end energy storage solutions. This driver only supports the sonnen ecoLinx product line.

Features

- Local IP communication with sonnen ecoLinx
- Create simple **Energy Profiles** to access different energy modes and trigger energy automation actions
- Weather API integration for easy Severe Weather Alert triggering
- Set Reserve Level
- Change Operating Mode
- View sonnen data via Control4 Touchscreens

Dealer Notes

- This driver uses sonnen API V2 REST API with token validation
- Weather API (national_weather_service) does not require an API token
- Cloudy Weather integration is only available on token-based Weather APIs

- Driver controls one ecoLinx. Multiple ecoLinx units will have a separate driver installed
- WebUI is only supported on T3 and T4 touchscreens currently

License Purchase & Assignment

This driver (and accompanying sonnen_energy_profile driver) are provided free of charge for use on Control4 systems by sonnen, Inc.

This driver is also compatible with the DriverCentral driver management system. We recommend setting up a DriverCentral link for future updates and driver management. <https://www.youtube.com/watch?v=xXiX1euk9bUI>

1. Create your project on the drivercentral.io website (this will generate a unique token used in the next step).
2. To download the driver, locate the downloads tab on the driver page on drivercentral.io. Once downloaded, unzip the file and add the driver to your drivers folder.
3. Install and activate the DriverCentral's Cloud Driver in your project by pasting the previously generated token in your project. If there is no license associated with the project a trial will automatically activate so long as the driver has not been installed in the past on the controller.
4. Once the Cloud Driver checks in to the project (it does so once every hour) the driver will activate.
5. If needed, use the Cloud Driver's action "Check Drivers" to force the check in and activate the driver right away.

Further clarification with visual representation can be found in the articles on <https://help.drivercentral.io/> <https://www.drivercentral.io>

Driver Setup

1. Navigate to the Connections tab, under Network and enter the IP address of the sonnen battery
2. Open a web browser and go to: <https://sonnenIP/dash/login>
 1. Login as **Vendor**

- *Note* Other Users may also work, the user account needs to be able to access the *Software-Integration* section.

2. Select **Software-Integration**

3. Under JSON API

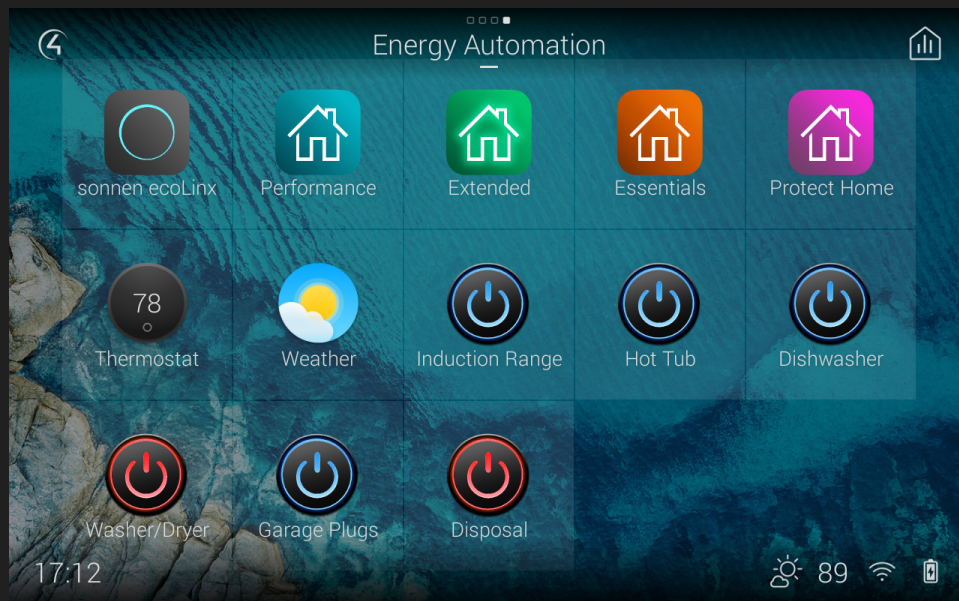
1. Enable **Read API**
2. Enable **Write API**
3. Copy the **API Token** and paste it into the Control4 drivers **sonnen**
API Token property

3. The driver will now attempt to read and configure itself

To use Weather Integration

1. Select a Weather API service
2. Fill in the Weather API token (if needed)
3. Choose which kind of weather alerts the driver will subscribe to

Setup Energy Dashboard



1. Dealers should create a separate room in the project and label it *Energy Automation* or *Power Dashboard* or something similar.
2. Add Energy Profile drivers to the project and complete the installation
 1. Most projects should have 3-5 Energy Modes. Some examples are:
Active, Weather Alert, Full Charge, Max Savings, Low Charge, High Tariff, Home Mode, Vacation Mode
3. Refresh Navigators

Driver Settings

Driver Properties

- *Cloud Status*: Displays driver license state
- *Driver Status*: Displays driver related information
- *Driver Version*: Displays driver version
- *Driver Actions*:
 - *View Status*: Displays general history and event information on the Lua tab
 - *Test Service Email*: Sends a test email to the service email provided
 - *Test Customer Email*: Sends a test email to the customer email provided
 - *Get battery status*: Requests current data from the battery.
 - *Get weather alerts*: Gets weather data from selected weather api
 - *Get latest dashboard*: Downloads the latest WebUI dashboard from the server
 - *Reload energy profile data*: Resets Energy Profile drivers
- *sonnen API Token*: The token needed to access the sonnen API. See [Driver Setup](#) on how to obtain token
- *Device Connection*: The current online\offline communication status
- *Battery Status*: The current battery state, mode and level
- *Energy Profile Running*: The current Energy Profile driver selected
- *Battery Mode*: The current battery operation mode
 - *Self Consumption* - (recommended) Solar power is used first to power the home, extra solar power made is stored in the battery. If the homes consumption is greater than the incoming solar, the extra power in the battery is discharged until it reaches the *Battery Reserve Level*. Remaining power will come from the Grid
 - *Time of Use* - Sets the charge\discharge times for the battery based on times of the day. These times are configured in the sonnenBatterie
 - *Manual* - Used to manually set the battery to a specific level (advanced setting and should not be used in normal system installation)

- *Backup Power* - Sets the stored charge to 100% and reserves all power for backup purposes
- *Battery Reserve Level*: The minimum amount of charge to always remain in the battery for backup purposes
 - **Note** Setting the Reserve Level to 100% is the same as Operating mode *Backup Power*.
 - Setting Reserve Level below 10% may have unintended consequences. The sonnenBatterie requires at least 6% of charge to operate.
- *Energy Events*: See section [Notification Settings](#)
- *Weather*: See section [Weather Service Integration](#)
- *Email\Text Notification Settings*: See section [Email\Text Notification Settings](#)
- *UI Settings*: See section [UI Settings](#)
- *Polling Timer (Seconds)*: The amount of time between requests to get the battery status.

Composer Variables

- *Battery_Charging_PrettyState*: (string) The current charge state including watts.
 - Example: charging 200 w
- *Battery_Charging_State*: (string) charging or discharging or none
 - Example: discharging
- *Battery_Current_level*: (number) 0-100 the battery energy level as percentage
- *Battery_Mode*: (string) The current Operating Mode of the battery
 - Example: Time of Use, Self Consumption, Manual, Backup Power
- *Battery_Reserve_Level*: (number) 0-100 the minimum battery energy to be left for backup purposes as percentage
- *Cloud_Cover*: (number) (if Weather API supports) the amount of cloud cover as a percentage for the next 8 hours
- *DeviceConnectionStatus*: (string) Connection status of the driver to the sonnen battery
 - Example: Online, Offline, Failed to Check In, Polling Stopped
- *Driver_State*: (string) The driver state

- Example: WEATHER_ALERT, NORMAL, GRID_LOSS, OFFLINE
- *Grid_Voltage*: (number) The voltage (2 Phase AC)
- *House_Consumption_Watts*: (number) The amount of power the home is consuming in Watts
- *Solar_Production_Watts*: (number) The amount of power the home is producing in Watts
- *Time_Remaining_On_Battery*: (level) x.x hours left on battery (ex. 1.4 would be 1.4 hours)
- *Weather_Alert_Message*: (string) The newest weather headline received
- *Weather_Alerts*: (string) The current list of active alerts (comma delimited)

Composer Events

- *Grid_Power_Outage*
- *Grid_Power_Restored*
- *is Discharging*
- *Battery Level Changed*
- *Battery Reserve Level Changed*
- *Battery Mode Changed*
- *Grid Voltage Changed*
- *House Consumption Changed*
- *Solar Production Changed*
- *Solar power creation starts*
- *Solar power creation stops*
- *Time Remaining Changed*
- *Driver Communication Failed*

Composer Commands

- *Trigger Driver Event*
 - Event: GRID_LOSS, WEATHER_ALERT, NORMAL
 - Message: The message to be sent to the event trigger (leave blank for default messaging)
- *Set Battery Mode*
 - Mode: Self Consumption, Time of Use, Manual, Backup Power

- *Set Battery Reserve Level*
 - Level: 1-100%
 - *Set Energy Profile*
 - Profile: List of energy profiles in project
-

Notification Settings

Events: Dropdown list of different notification events. This allows dealers to customize how the notification is sent to the customer.

Events are as follows:

- *WEATHER_ALERT:* When a Weather Alert occurs.
- *GRID_LOSS:* When Grid power is lost.
- *NORMAL:* When grid power is restored or weather alerts are over
- *OFFLINE:* When the driver has not been properly connected to Control4 or a licensing issue

Weather Service Integration

After selecting an API, dealers can create *Weather Alerts* that triggers an event when it is received by the driver.

All APIs support severe weather alert responses. Some also support live data which offer cloud cover data. If no Weather API Service is chosen, the driver WILL NOT use Weather Settings.

When an alert is received, it will trigger Energy Event: **WEATHER_ALERT**

Use this event to adjust the backup reserve level of the battery or active other actions in the home.

After all weather alerts have ended, the driver will set the driver state to **NORMAL**. This will happen silently (notifications will NOT be sent). Use Composer Event *All Weather Events Ended* to perform actions needed.

APIs Supported:

- *National Weather Service:* This service supports alerts only. Does not require an API token.
 - <https://www.weather.gov/>

- *Tomorrow.io*: This service supports both alerst and cloudy weather alerts.
Requires API token
 - <https://www.tomorrow.io/>
 - 500 weather requests per day are free per account

Weather Driver Properties

- *Weather: API Service*: This chooses which API to receive alert and live data from
- *Weather: API Status*: The current state of the API communication with driver
- *Weather: Polling Time*: The amount of time between weather alert updates.
- *Weather: Subscribed Alerts*: A comma-delimited list of Weather Alerts the driver will use when alert data is received to trigger the WEATHER_ALERT event.
 - Ex. *Hurricane* will trigger a weather alert in the driver if any kind of alert with the word hurricane is received. Example: Hurricane Watch, Hurricane Warning, Hurricane Advisory
 - Ex. *Winter Storm Warning* will not trigger an alert for Winter Storm Watch.
 - All other weather alerts are rejected
- *Weather: Current Alerts*: The list of active alerts.

Cloud Charging

Some Weather APIs support the ability to read the expected cloud cover forecast. If enabled, the driver trigger an event **CLOUDCOVERCHARGE**

- *Weather: Cloud Cover Charge Time*: This setting determines the time at which the driver will perform its cloudy weather charge test.
 - The driver will average cloud coverage 6hours - 12hours in the future of selected time
 - Ex. Setting property to *3AM* would tell the driver to average cloud cover between 9am-3pm
 - Ex. Setting property to *9AM* would tell teh driver to average cloud cover between 3pm-9pm

- *No disables Cloud Charging*
- *Weather: Cloud Cover:* Sets the cloud cover threshold to trigger the cloudy weather charge time.

Weather Composer Events

- *Weather Alert Occurs* when a new weather alert occurs. Multiple alerts received at the same time only trigger this event one time.
- *Weather Alert Started Occurs* when there is a weather alert the driver is subscribed to
- *Weather Alert Ended Occurs* when a weather alert ends. (Each Weather Event triggers this)
- *All Weather Alerts End Occurs* when all alerts the driver was monitoring end.
- *Cloud Cover Level Received Occurs* when the driver receives cloud cover information from the API

Email\Text Notification Settings

Events have the ability to send customized email, text and push notifications to certain groups of people. The properties listed here set those groups for easier programming.

- **Email: Service:** This property is intended to send an email to the installer or dealership of the project.
- **Email: Customer:** This property is for the customer. When a message is sent the subject line will come from the property *Email: Subject*
- **Email: Subject:** This property is the subject line of the email (or push notification subject line). When the driver will build this subject line based on the following criteria:
 - When first loaded the driver will scan the project and look for another driver with the property *Email: Subject* and if found will use what is already being used
 - If no driver is found with *Email: Subject* the driver will use the zipcode of the project [CITY] Smart Home. Example: Denver Smart Home
 - If the driver cannot find a city the subject line will default to: My Control4 Smart Home

UI Settings

This driver uses the UI button proxy from Control4. This proxy allows custom icons to be displayed on Control4 Navigator. The driver has the ability to trigger quick actions when the icon is pressed. Here are the defaults:

- **UIButton: Single Tap:** Launch Webview (only available on T3/T4 Touchscreens)
 - *Nothing* If clicked, driver will not do anything
- **UIButton: Double Tap:** Nothing
 - *Same options as Single Tap*
- **Control4 Navigators:** This property defines which Touchscreens will receive the Navigator Popup event (if enabled under the Events section).

NOTE As of OS3 double tap must be done slowly (Press... Press). It will NOT work if you quickly press the button twice.

Support

For hardware installation, parts, RMA please contact sonnen directly:

Email: service@sonnen-batterie.com

Phone: [818-824-6363](tel:818-824-6363)

Driver Integration, driver installation, contact DriverCentral for support:

<https://help.drivercentral.io>

Warranty & Disclaimer

[sonnen Factory Limited Warranty](#)

Developer Information

brought to you by:
Cinegration Development, LLC



www.cindev.com
www.drivercentral.io/cindev/

We are always looking to improve our drivers.
Please send your suggestions to: info@cindev.com