WILDLIFE
ACOUSTICS

SMART System

User Guide

2025-01-29

Terms of Use

Customer agrees to be bound by the terms and conditions of Wildlife Acoustics, Inc., which can be found at https://www.wildlifeacoustics.com/legal-documentation. Customer further agrees that any End User is made aware of and bound by such terms and conditions.

Document History

This documentation receives frequent updates, which are described here.

Document Version	Publish Date	SMART Software Version	Changes
1.0.0	2024-11-12	1.3.4	• Complete rewrite
1.0.1	2025-01-01	1.3.4	• Rearranged sections of SMART Controller specifications • Updated copyright date range
1.0.2	2025-01-29	1.3.4	• Introduced this Document History section • Wrote additional short descriptions for some top-level topics • Added index terms to help find topics when searching for common synonyms. • For all images, added alt text or, in cases where the surrounding text communicates the same meaning as the image, inserted empty alt text values.

Contents

1. Introduction

Song Meter with Analysis and Remote Transfer, or SMART, is a multi-channel, ultrasonic audio recording and analysis platform.

SMART is designed to support bat monitoring deployments with remote configuration, data transfer, and real-time bat call analysis. It is also designed to integrate with third-party hardware, including wind turbine SCADA networks or similar control systems.

Bat Monitoring and Call Analysis

The SMART System comprises the most advanced bat monitoring and recording system Wildlife Acoustics have produced. It incorporates the audio analysis features from Kaleidoscope Pro Sound Analysis Software to detect bat pulses, reject extraneous noise, measure call characteristics, and identify bat species in real time.

Remote Data Transfer

The centerpiece of the SMART System, the SMART Controller, is a ruggedized computer with built-in Wi-Fi and Ethernet connectivity and a browser-based Control Panel. Using any computer or mobile device, you can connect to the SMART on-site, via a private network, or via the Internet to check the status of your SMART, configure its settings, and retrieve your data.

The SMART Control Panel allows you to manually download your recordings and log files to your web browser, and you can configure the SMART to back up any new data to USB storage or networked storage automatically, every day. It can then send a daily status report via email.

Acoustically Triggered Curtailment and Control System Integration

The SMART is designed to integrate with wind turbine control systems and allow you to curtail a turbine based on real-time analysis of nearby bat activity. Design custom criteria to distinguish nearby bats from other sources of noise, and raise alarms to signal bat activity to other devices on a SCADA network or similar control system. The status of these alarms can be communicated to control systems via *OPC Unified Architecture (OPC UA)*, Modbus, or the SMART Controller's programmable digital I/O.

Open-Source, Customizable Platform

The SMART System Software runs on Ubuntu Linux and has all the configurability of any Linux-based system. If you require any functionality not already provided via the SMART System's browser-based Control Panel, you or your IT department can configure the system on a much deeper level via the Ubuntu command line, custom scripts, and third-party tools.

Owning a SMART System means you can access the SMART System's built-in executables and library functions. By writing custom software to interact with these components directly, you can control the SMART with a level of customizability not possible with a browser-based graphic user interface.

The SMART Controller hardware can also be expanded. Connect the SMART to the Internet using a cellular modem accessory, or use the programmable Digital I/O pins to incorporate outside signals, such as those from weather monitoring devices, and output control signals based on your own custom logic.

Curtailment Strategies

The SMART System is designed to facilitate *acoustically triggered curtailment (ATC)* of wind turbines. *ATC* is the practice of slowing or stopping a wind turbine in response to detected bat vocalizations and activity levels.

Wind turbine curtailment strategies can be distinguished by what kinds of information they take into account:

Blanket Curtailment

Blanket curtailment, sometimes called simply *curtailment*, means keeping wind turbines inactive during times of the day and days of the year when bats are expected to be active. This strategy does not incorporate real-time measurements of bat activity or environmental conditions.

Operational Curtailment

Operational curtailment involves taking real-time wind speed measurements into account, in addition to date and time. Bats tend to be most active when wind speeds are low, so the turbine is shut down when wind speeds are slower than a set threshold.

Smart Curtailment

Smart curtailment goes beyond operational curtailment by accounting for environmental factors beyond wind speed, such as precipitation and temperature. The turbine is shut down during environmental conditions associated with bat activity, but real-time bat activity measurements are not directly taken into account.

One use case for the SMART System is to record bat activity prior to a turbine's construction in order to build a model correlating bat activity with environmental conditions. The SMART Controller's connectivity features and the robustness and redundancy of the SMART MIC-1 make the system well suited for installation on a wind turbine or meteorological tower, regardless of the specific goal. However, SMART has additional features that can grant your curtailment program an additional level of sophistication.

Acoustically Triggered Curtailment

SMART is designed to let you use real-time measurements of bat activity to inform your curtailment logic–not just past measurements. Implementations can vary, but this type of *ATC* generally follows this sequence:

1. As the SMART microphones detect and record nearby bat echolocation calls, the SMART System measures the characteristics of each pulse and average measurements for pulses in a given sequence.
2. These call measurements are processed through user-designed logic to distinguish bat calls from mechanical noise.
3. When the number of bat calls in any set of criteria meets a user-specified threshold, a corresponding *alarm* is raised. The SMART can signal eight alarms based on individual pulse measurements and eight alarms based on average measurements of pulse sequences.
4. The status of each of the 16 possible alarms is communicated to the turbine control or SCADA system. The SMART supports multiple communication protocols for this purpose.
5. The turbine control system processes SMART's alarm signals, along with other relevant data, and determines whether to shut down or slow a turbine.

SMART System Physical Components

The SMART System is comprised of the SMART Controller and at least one SMART MIC-1 microphone.

SMART Controller Overview

The SMART Controller is a rugged computer running Ubuntu Linux and the SMART System Software.

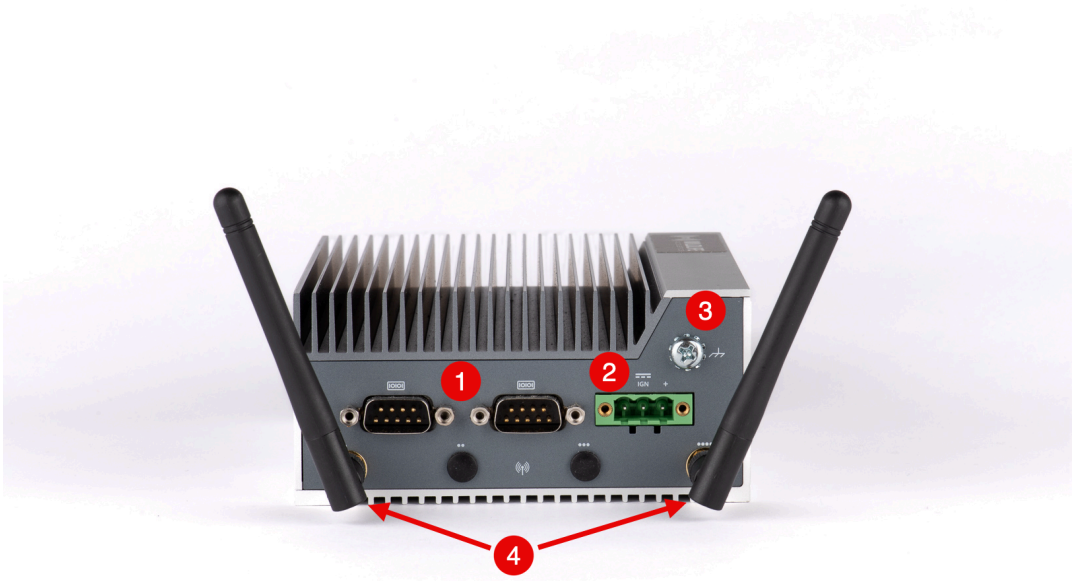
The SMART Controller is the central component of a SMART System. It hosts the browser-based SMART Control Panel used for configuring the SMART System, analyzes and records the audio from any attached microphones, and communicates with a wide variety of external devices.

SMART Controller Connectivity

Figure 1. SMART Controller Connectivity: Ethernet and DisplayPort

1. One Gigabit Ethernet port can be used to connect the SMART Controller to a network.
2. Two Gigabit Ethernet ports with *Power over Ethernet (PoE)* can directly power two SMART MIC-1 microphones. With an external *PoE* switch, up to three SMART MIC-1 microphones can be used simultaneously with the base model SMART Controller.
3. Two DisplayPort sockets allow you to view the SMART Controller's Ubuntu command line on an external display.

Figure 2. SMART Controller Connectivity: Serial, Power Input, and Wi-Fi



1. Two **DB-9 serial ports** can output bat call analysis measurements via the RS-232, RS-422, or RS-485 protocols. These ports can also be used to communicate the status of the SMART Controller's alarms via Modbus.
2. The SMART Controller's **power input** accepts 9-36 V DC.
3. A **grounding lug** can be used to protect the SMART Controller from electrical discharges.
4. Two **Wi-Fi antennas** allow you to connect the SMART Controller to an existing Wi-Fi network or to host a local Wi-Fi network.

Figure 3. SMART Controller Connectivity: Digital I/O, USB, and Power Button

1. Optically isolated **Digital I/O pins** are available for use by custom applications. One possibility is to signal the presence of bats to connected devices using a simple script.
2. A **CAN bus** can be used for custom applications.
3. Four **USB 3.0 Type A ports** allow you to connect storage devices, keyboards, and other accessories.
4. The **power button** is not generally required, as the SMART Controller boots automatically. You can use it to manually reboot the SMART Controller.

Optional Hardware Upgrades

The SMART Controller can be purchased with optional upgrades, including the following:

- Faster CPU with more cores
- Increased RAM capacity
- Built-in 4G LTE cellular mode
- Increased internal storage capacity

Contact Wildlife Acoustics Sales at https://www.wildlifeacoustics.com/contact-us to request additional details.

SMART MIC-1 Overview

The SMART MIC-1 is a weatherproof, ultrasonic microphone with redundant microphone elements, a built-in speaker for sensitivity testing, and a built-in heater for clearing moisture.

The SMART MIC-1 communicates with the SMART Controller via Ethernet and is powered via *PoE*. A single run of Ethernet cable can extend up to 100 meters. Ethernet repeaters or fiber-optic converters can be used to cover even greater distances, as long as *PoE* is supplied to each microphone.

Redundancy and Resiliency

The SMART MIC-1 is designed for deployments during which it may be difficult or impossible to directly access the microphone. It has several features to ensure it can continue recording in the most difficult conditions.

The SMART MIC-1 enclosure is made of anodized aluminum and is rated at IP67, meaning it is dust-tight and can withstand immersion in up to one meter of water for thirty minutes. An included weatherproof cover can be installed on an Ethernet cable and screwed onto the SMART MIC-1 body, protecting the cable from water and preventing it from being pulled loose.

The body of the SMART MIC-1 has a strip of unanodized aluminum, allowing you to ground the body of the microphone with the SMART MIC-1 Mounting Bracket or a similar mounting system to protect the microphone and downstream components from electrical discharges.

Each SMART MIC-1 is built with two microphone elements and a speaker. At regular intervals throughout your deployment, the SMART MIC-1 can test the sensitivity of its two microphone elements, report its results via email or data backup, and switch to the microphone element whose sensitivity more closely matches the target. Only one microphone element is used at a time.

Each SMART MIC-1 also has a built-in heating element. Condensation or frost buildup on a microphone can muffle incoming sound, reducing the microphone's sensitivity. At regular intervals, the built-in heater can turn on to evaporate any moisture before recording continues.

Connecting Multiple Microphones to One SMART Controller

The base model SMART Controller can connect to up to three SMART MIC-1 microphones simultaneously. Each microphone's recording schedule and settings can be set independently. When one or two microphones are connected to a SMART Controller, they can be powered by the SMART Controller via *PoE*. In order to connect more than two microphones, you must use an Ethernet switch capable of providing *PoE*.

The number of microphones you can use to monitor simultaneously depends on the **Sample Rate** you have selected for recording and is limited by the processing capabilities of the SMART Controller's CPU. The SMART Controller can be purchased with an optional CPU upgrade to increase the number of microphones you can use at a given sample rate.

Table 1. Maximum Sample Rate by Number of Microphones

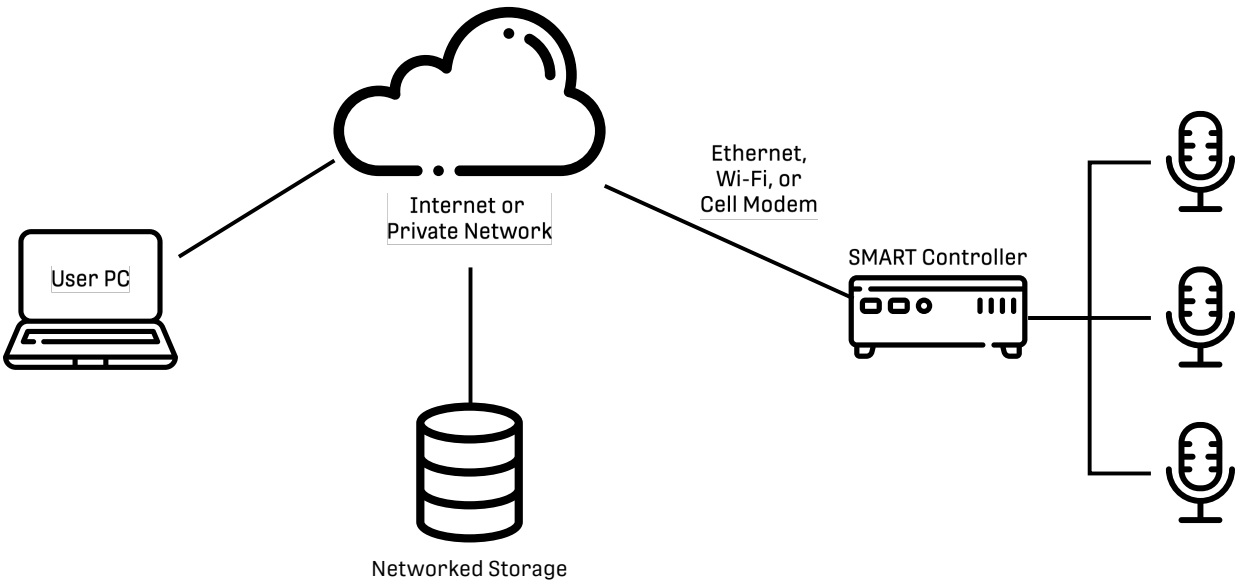
Number of Microphones	Recommended Maximum Sample Rate	
	Base-Model SMART Controller (2-Core CPU)	Upgraded SMART Controller (4-Core CPU)
1	500 kHz	500 kHz
2	384 kHz	500 kHz
3	256 kHz	500 kHz
4	Not supported	500 kHz

Example Deployments

The SMART System is highly configurable and can be used for many types of deployments.

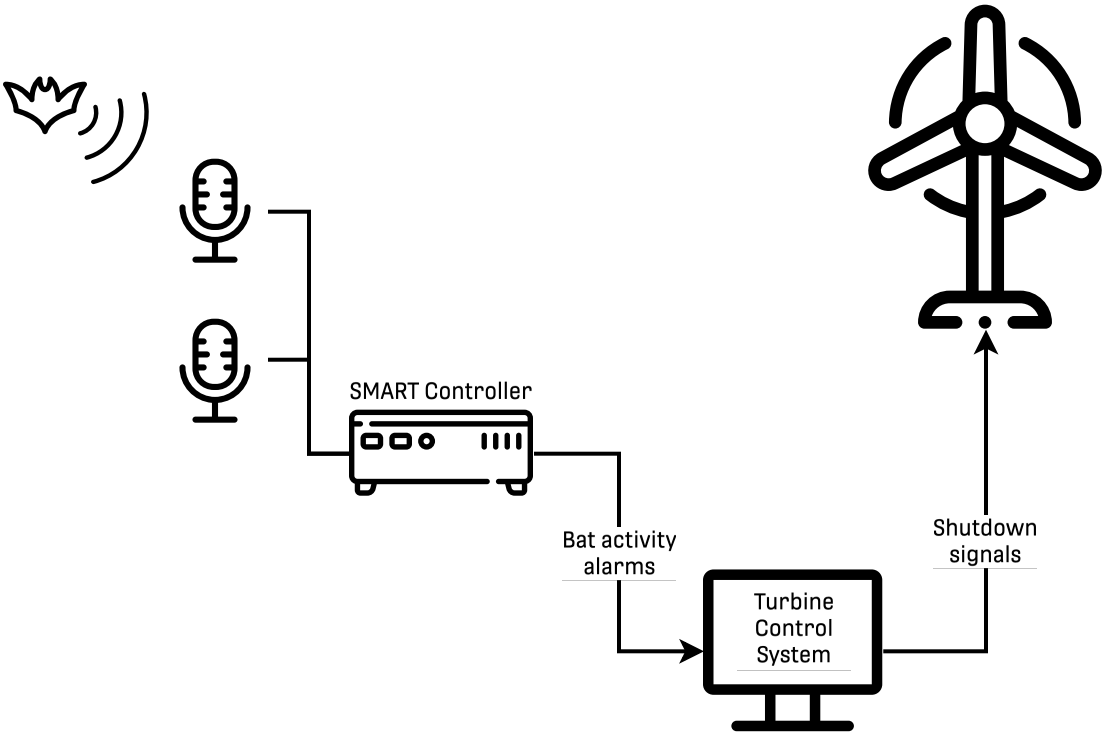
Data Collection with Remote Access

On an installation with connectivity to the global Internet or a private intranet, the SMART System can record bat activity and automatically back up recordings and activity summaries to network storage. Remote users can check the system's status, adjust settings, and retrieve data via the network.



Acoustically Triggered Turbine Curtailment

Installed at a wind farm, the SMART System can detect bat activity near a turbine and signal to the wind farm's control system to stop or slow down the turbine.



Getting Support

If you have questions about your SMART System or encounter any problems, there are many resources available to you.

Frequently Asked Questions (FAQs)

You can find frequently asked questions at https://www.wildlifeacoustics.com/resources/faqs or from our home page under **Resources > Frequently Asked Questions**.

Contacting Wildlife Acoustics Support

For technical questions, contact Wildlife Acoustics Support using one of the methods below. If contacting us via our website, **please provide as much detail as you can** so we can provide effective help as quickly as possible.

- https://www.wildlifeacoustics.com/contact-us
- North America (toll-free): +1 (888) 733-0200
- Outside North America (toll charges may apply): +1 (978) 369-5225

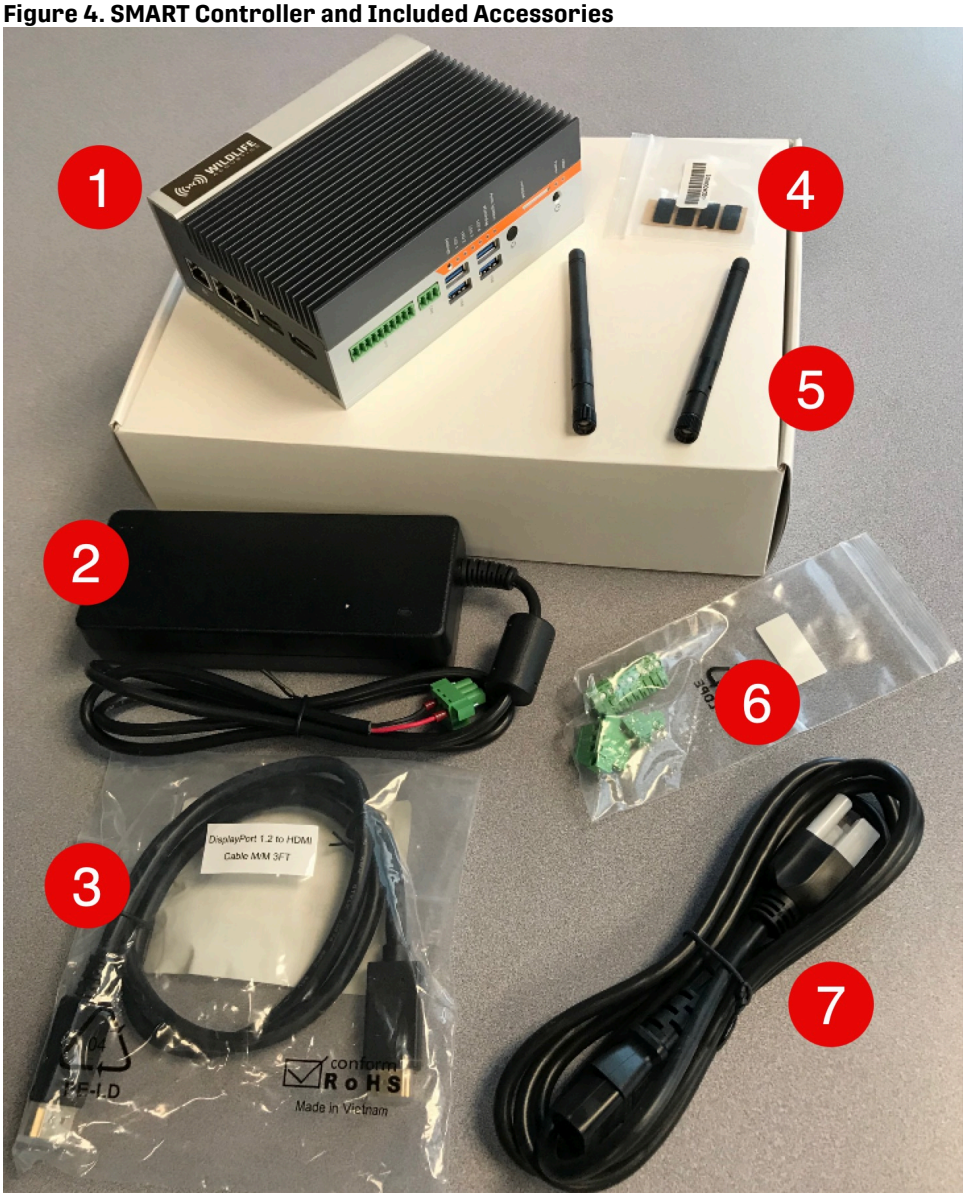
2. Getting Started

The first time you use a SMART System, we recommend setting it up in an accessible location, such as an office. This will make it easier to configure the SMART and perform any necessary troubleshooting.

SMART Controller Package Contents

Each SMART Controller ships with multiple accessories included in the box. These include necessary power and connectivity components.

Figure 4. SMART Controller and Included Accessories



1. SMART Controller
2. 24-volt DC power supply with pre-installed three-pin power plug
3. One DisplayPort-to-HDMI cable

4. Four adhesive rubber feet
5. Two screw-on Wi-Fi antennae
6. Plastic bag containing three additional plugs:
    - One spare three-pin power plug for custom DC power source (Dinkle PN: 2ESDVM-03P)
    - One three-pin CAN bus plug (Dinkle PN: EC350V-03P)
    - One ten-pin DIO bus plug (Dinkle PN: EC350V-10P)
7. IEC 60320 C13 AC power cable for the power supply
   The terminating connector type is based on the purchaser's local standard.

SMART MIC-1 Package Contents

Each SMART MIC-1 microphone ships with several essential accessories for protecting the microphone and SMART Controller from electrical discharge.

Figure 5. SMART MIC-1 and Included Accessories

1. One SMART MIC-1 microphone
2. One RJ45 waterproof cable gland
3. One cable ferrite for EMI rejection
4. One in-line RJ45 surge suppressor
5. One 30-cm RJ45 Ethernet patch cable

Turn the SMART Controller Off and On

If the SMART Controller's power is interrupted, it will boot automatically when power is restored. You can turn the SMART Controller off and on manually using the power button.

About this task
To avoid the possibility of data loss, we recommend shutting down the SMART Controller before disconnecting the power supply.

Procedure
1. To turn the SMART Controller off, press and hold the power button until the **Power** LED turns off.
2. To turn the SMART Controller on again, disconnect and reconnect the power supply or press the power button.
   The SMART Controller usually takes one to two minutes to finish booting.

Connect to the SMART Control Panel for the First Time

When the SMART controller is brand new, you can use any computer to connect to its web interface over a local Wi-Fi connection. This allows you to explore the SMART system's settings, and it does not require specialized technical knowledge.

Before you begin
You will need the following:

- SMART Controller with Wi-Fi antennae and power supply
- Wi-Fi-enabled device with a web browser, such as a laptop computer

Procedure
1. Connect the two included Wi-Fi antennae to SMART Controller via the two screw-on coaxial connectors.
2. Connect the SMART Controller to power.
   a. Plug the green plastic plug from the power supply into the port labeled **IGN**.
   b. Use a flat-blade screwdriver to secure the power plug to the SMART controller using the plug's two built-in bolts.
   c. Connect the power supply to AC power via the included IEC power cable.

   *The SMART Controller will power on automatically, and the **Power** LED will light up blue. Initial startup can take up to two minutes.*
3. From a nearby computer, search for Wi-Fi networks and connect to the SMART Controller's Wi-Fi hotspot.
   The SMART Controller's **default Wi-Fi network name** begins with `smart` and ends with the SMART Controller's twelve-digit MAC address, without periods. For example, `smart-1234567890ab`.
   The SMART Controller's **default Wi-Fi password** is `wildlife`.
4. Open a web browser and enter `https://192.168.19.1/` to navigate to the SMART Controller's default IP address.
   *Your browser will prompt you with a **Sign in** window. The default login credentials are:*
   - *Default Username:* `smart`
   - *Default Password:* `wildlife`

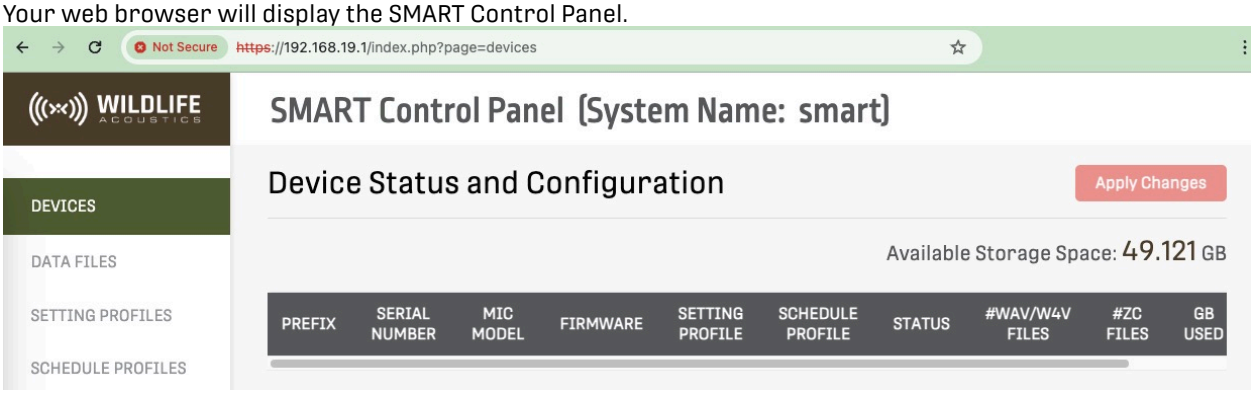
   Troubleshooting

   Your browser may display a warning that your connection is not private. This is normal when connecting to a device like the SMART Controller with its default configuration. Select your browser's option to proceed anyway.

   To avoid this warning in the future, ask your IT department to provide a signed SSL certificate and install it on the SMART Controller.

Results

Your web browser will display the SMART Control Panel.



What to do next

You can use the SMART Control Panel to explore the SMART System's settings. You can also configure the SMART Controller to connect to a private network or the Wildlife Acoustics SMART Portal.

Related information

Privacy or Security Warning When Using SMART Control Panel *(on page 125)*
Local Network Configuration *(on page 48)*
SMART IoT Gateway *(on page 49)*

3. Microphone Settings and Schedules

Each microphone connected to a SMART Controller can have its own *Setting Profile* and *Schedule Profile*. These profiles determine when the microphone is active; the settings it uses to detect, record, and analyze bat echolocations; and set the scheduling of microphone sensitivity testing and moisture clearing.

Setting Profiles

A microphone's Setting Profile determines how a microphone behaves and how its audio is processed while it is active. This includes:

- Basic audio settings, like *Sample Rate*, *Filters*, and *Gain*
- Trigger parameters, which determine what kinds of signals will prompt audio to begin recording
- Bat Classifier settings, such as whether to run bat identification and which species to consider
- File output formats, including *zero-crossing (ZC)* and full-spectrum WAV with optional W4V compression
- Calibration settings, which control how the SMART MIC-1 tests its sensitivity with its built-in speaker
- Heater settings, which control the SMART MIC-1's built-in heating element to prevent moisture from muffling the microphone elements

Schedule Profiles

A microphone's Schedule Profile determines when it is active throughout each day. Schedules can be formed of multiple *schedule blocks*, each of which has a defined Start, End, and optional Duty Cycle. The Start and End of a schedule block can reference a particular time of day or some offset relative to sunrise and sunset, which are calculated based on the SMART Controller's date, time zone, and location.

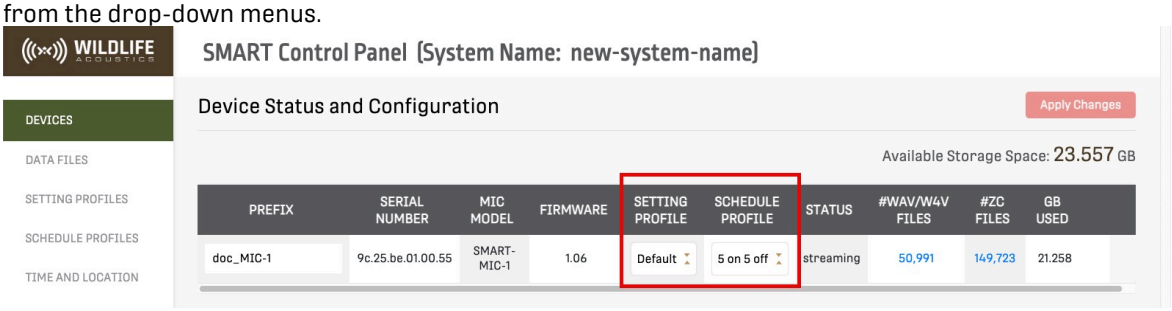
The schedule block system is very similar to the schedule system used by other Song Meter recorders, including the Song Meter SM4, Song Meter Mini, and Song Meter Micro families.

Select a Microphone Setting Profile and Schedule Profile

Each microphone attached to a single SMART Controller can have a unique Setting Profile and Schedule Profile selected from the ones included by default or that you create yourself.

Procedure

1. Open the SMART Control Panel and go to the **Devices** page.
2. For each microphone connected to your SMART Controller, select a Setting Profile and Schedule Profile from the drop-down menus.



Different microphones can use the same Setting Profile and Schedule Profile, or they can use entirely different settings and schedules.

Creating and Editing Settings Profiles

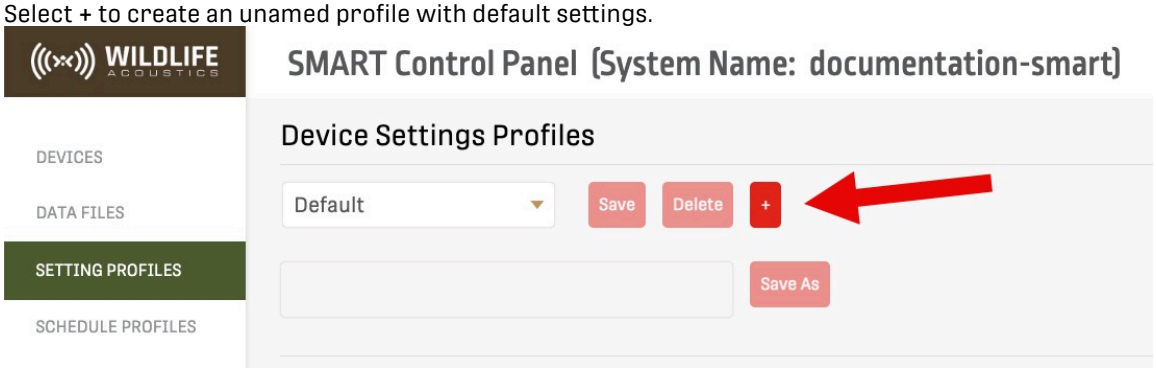
The **Setting Profiles** tab is where you can create and edit profiles to apply to each of the SMART System's microphones.

Create a New Setting Profile

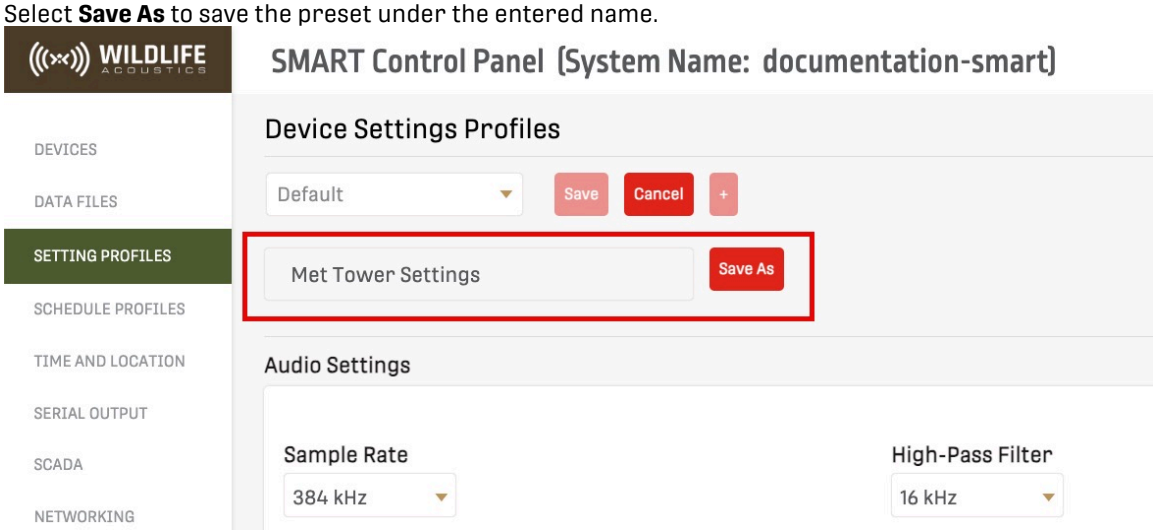
Setting Profiles are collections of microphone settings that can be assigned independently to each microphone in a SMART System. You can create and store as many Setting Profiles as you want for different microphone behaviors.

Procedure

1. Open the SMART Control Panel and go to **Setting Profiles**.
2. Select + to create an unamed profile with default settings.



3. Enter a new preset name and modify any settings.
4. Select **Save As** to save the preset under the entered name.



What to do next

To use your new profile, you must assign it to one or more microphones on the **Devices** tab.

Related information

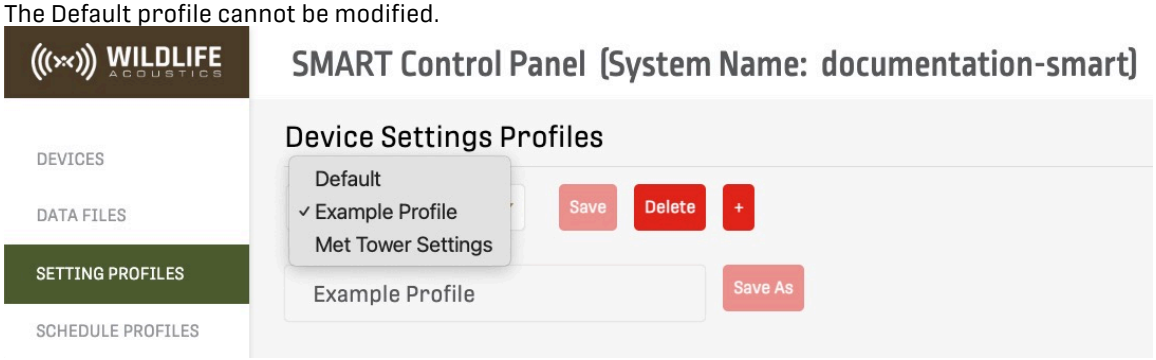
Select a Microphone Setting Profile and Schedule Profile *(on page 20)*

Edit an Existing Setting Profile

You can make changes to an existing Setting Profile or save a modified copy of it as a new profile.

Procedure

1. Open the SMART Control Panel and go to **Setting Profiles**.
2. Select a profile other than Default.
   The Default profile cannot be modified.



3. Make changes to any settings.
4. Save your changes to the existing profile or to a new profile.
   - To overwrite the existing profile, select **Save**.
   - To save your settings to a new profile, edit the profile name and select **Save As**.

What to do next

To use your modified profile, assign it to one or more microphones on the **Devices** tab.

Delete a Setting Profile

To declutter the list of Setting Profiles on your SMART Controller, you can delete unused profiles.

Procedure

1. Open the SMART Control Panel and go to **Devices**.
2. Ensure that the profile you wish to delete is not assigned to any microphones on the **Devices** tab.
   You cannot delete profiles that are in use.
3. Go to **Setting Profiles** and select the profile you wish to delete.
4. Select **Delete**.

SMART MIC-1 Settings

Each SMART MIC-1 microphone connected to a SMART System can have independent settings. These settings govern everything from basic audio parameters to parameters defining how the SMART System detects bat pulses in each microphone's recorded audio.

Audio Settings

The **Audio Settings** set basic parameters for incoming audio, such as recording format, sample rate, and gain.

Sample Rate

The *sample rate* of a digital audio system sets the upper frequency limit on the recorded audio. The sample rate also directly affects the file size for a recording of a given length. The available sample rates depend on how many microphones are connected to a single SMART Controller.

Options

256 kHz

384 kHz

500 kHz

Default

256 kHz

Audio up to half of the sample rate will be recorded in the resulting file. For example, a sample rate of 256 kHz means you can record audio up to 128 kHz. For bat monitoring, your sample rate should be at least two times the highest frequency produced by your species of interest. Choosing the lowest sample rate that meets this criterion will minimize your data storage use.

Native *ZC* recorders do not use a sample rate. However, the SMART produces *ZC* files by converting audio that is first recorded in full-spectrum. Therefore, the sample rate will determine the highest frequency that can be recorded for both full-spectrum and zero-crossing files.

Limitations on Sample Rate

The SMART Controller has been tested to determine recommended limits on sample rate and number of microphones. Exceeding these recommendations can result in dropped samples or audio discontinuities. The SMART Controller can be purchased with an upgraded CPU, which expands the number of microphones that can be used simultaneously with a given sample rate.

Table 2. Maximum Sample Rate by Number of Microphones

Number of Microphones	Recommended Maximum Sample Rate	
	Base-Model SMART Controller (2-Core CPU)	Upgraded SMART Controller (4-Core CPU)
1	500 kHz	500 kHz
2	384 kHz	500 kHz
3	256 kHz	500 kHz
4	Not supported	500 kHz

High-Pass Filter

The *high-pass filter* attenuates incoming audio signals lower in frequency than the specified *cutoff frequency*. This can reduce the effects of wind noise and other low-frequency sounds on the quality of your recordings.

Options

8 kHz

16 kHz

Default

16 kHz

For every factor of two below the cutoff frequency, the audio at that frequency is reduced in amplitude by 12 dB.

Example
If the high-pass filter is set to a cutoff frequency of 16 kHz, sound at 8 kHz will be attenuated by 12 dB, and sound at 4 kHz will be attenuated by 24 dB.

Recommended High-Pass Filter Setting for Bat Monitoring

If none of your bat vocalizations of interest extend below 16 kHz, set the **High-Pass Filter** to 16 kHz for the best noise-reduction effects. If you need to record frequencies between 8 and 16 kHz, set the **High-Pass Filter** to 8 kHz.

Gain

Gain is amplification that can be added to the microphone signal before it is digitized. It should be adjusted to either increase the level of quiet signals or to prevent loud signals from distorting.

Options

0 dB

12 dB

Default

0 dB

Maximum Duration

The **Maximum Duration** setting sets the maximum length for any audio file recorded from a given SMART microphone. This setting has similarities to the **Maximum Sequence Duration**, but **Maximum Duration** should generally be used only during non-triggered recording.

The **Maximum Duration** setting takes effect regardless of whether **Triggered Recording with Bat Call Analysis** is enabled or disabled. If **Maximum Duration** is not specified, the maximum recording length is set by the maximum file size of 2 GB.

During triggered recording, if a recording reaches the specified **Maximum Duration** while the triggering signal is still present, the recording will end, and a new recording will immediately begin.

Setting	Recommended Value
Maximum Duration	Blank

Triggered Recording with Bat Call Analysis

In the default recording mode, with **Triggered Recording with Bat Call Analysis** enabled, the SMART System records only in response to bat calls detected by its microphones, and it analyzes the recording contents to measure common bat call parameters. This setting can be disabled to record continuous audio without bat call analysis.

Triggered recording with bat analysis is the recommended recording mode for most situations because of two key advantages:

- Audio is only recorded in response to ultrasonic activity, which reduces storage usage.
- Audio files are analyzed to measure bat call characteristics, which can be used for bat activity alarms.

Disabling **Triggered Recording with Bat Call Analysis** for a given microphone means the SMART will record audio continuously, without the use of any trigger mechanism and without any form of bat call analysis. This behavior may be desirable for measuring ambient, ultrasonic noise or other less-common use cases.

Setting	Recommended Value
Triggered Recording with Bat Call Analysis	Enabled

Pause Triggering When Heater Is On

Pausing triggered recording while the heater is on can avoid false triggers caused by noisy power.

When the SMART MIC-1 heater is enabled, significantly more power is fed to the microphone than normal. If the power supplied to your SMART System is particularly noisy, this additional power may cause that electrical noise to appear in the SMART MIC-1's audio signal.

Depending on the characteristics of the noise, it may trigger the SMART System to begin recording and to analyze the noise as if it were bat echolocation calls.

If the noise is narrow in bandwidth and does not rapidly change frequency, you can mitigate its effects using the Constant-Frequency Filter. Otherwise, you can pause triggered recording whenever the heater is enabled.

Setting	Recommended Value
Pause Triggering When Heater Is On	Disabled unless the heater creates excess noise in your recordings.

Recording Output Settings

For each Microphone Setting Profile, you can set whether to save processed audio as full-spectrum recordings, zero-crossing recordings, both, or neither.

If neither output format is enabled, the SMART System will still analyze incoming audio to perform the following functions if they are enabled:

- Automatic bat species classification, producing a `results.csv` file containing records of detected bat species by date and time
- Bat activity alarm logic

In most cases, it is strongly recommended to save audio in at least one format. This allows you to manually verify the automatic bat classification results and troubleshoot the behavior of your alarms.

Recommended Output Settings for Bat Monitoring

The SMART can save your recordings as full-spectrum WAV files, zero-crossing ZC files, or both. WAV files provide a much fuller image of the captured audio than ZC files when analyzed in a spectrogram viewer, but they are also much larger files.

If you are recording sample data to design bat activity alarms for wind turbine deployment, we strongly recommend you save both WAV and ZC files. WAV files are much easier to examine for bat call and noise characteristics, but running your test alarms on ZC files direct from the SMART system provides the most accurate simulations.

In recording environments with low levels of ambient noise, you can record only ZC files to dramatically reduce your file storage requirements. We recommend this only if you are familiar with analyzing ZC files and understand their limitations.

Setting	Recommended Value
Output WAV	Enabled, except when storage space is extremely limited
Output ZC	Situational Enabled when recording test data for bat activity alarms

Full-Spectrum Output Settings

For each Microphone Setting Profile, you can configure whether to save full-spectrum recordings in either the uncompressed WAV format or the compressed W4V format.

On the **Setting Profiles** tab, enabling **Output WAV** means that the SMART System will save full-spectrum versions of recorded audio to the internal drive. The SMART System's noise scrubbing and alarm logic run on *ZC* representations of the audio, but full-spectrum recordings provide much more information that can be useful when manually analyzing recordings.

The default **WAV Compression** setting is **None**. This means the format used for full-spectrum recordings is WAV, which is an uncompressed format supported by a wide range of audio analysis applications.

W4V Compressed Output

Setting **WAV Compression** to a **W4V** setting means files will be saved in the W4V format. W4V is a compression format developed by Wildlife Acoustics specifically for bioacoustics applications. Increasing the degree of W4V compression raises the level of background noise in your recordings, but it does not otherwise alter the content of audio in ways that compression formats designed for music or speech might.

Table 3. W4V File Size Reduction from WAV

W4V Setting	Approximate File Size Reduction from WAV
W4V-4	50%
W4V-6	62.5%
W4V-8	75%

To test the different levels of compression, record some example files without compression. Use Kaleidoscope Pro to convert the files to W4V format using different levels of compression, and examine the different versions in the Kaleidoscope Pro viewer. Decide whether the increase in noise level is detrimental before you decide to record directly to a compressed format.

Zero-Crossing Output Settings

For each Microphone Setting Profile, you can configure whether you want to save recorded audio in *ZC* format, with or without noise-removal.

ZC recordings use a file format roughly one thousand times smaller than equivalent WAV files. This format can only record a single frequency component at any point in time, which is sufficient for representing bat calls. However, ZC files are not suited for representing multiple, complex, overlapping sounds in a single recording.

Saving ZC files without saving full-spectrum files may be necessary when you need to severely conserve storage space or have limited bandwidth for data backups.

If you are configuring alarms for an *ATC* system, you should save ZC files to use as sample data for alarm simulation. This is because real-time alarms work by analyzing the SMART System's ZC output, so you should use the same format during testing.

Enhanced ZC Processing

The SMART's default method of recording zero-crossing files, with *enhanced ZC processing*, is much more sophisticated than typical zero-crossing recorders. SMART uses advanced signal processing to isolate likely bat calls from the background noise before it converts the audio to ZC format. This results in much cleaner ZC files with less noise than could be produced by zero-crossing recorders without the same level of processing capabilities.

When **Use Enhanced ZC Processing** is disabled, the SMART produces files similarly to an analog zero-crossing recorder. Only a basic band-pass filter is applied before it calculates zero-crossing intervals. This produces messier files in which bat calls are much more likely to be obscured by noise.

Setting	Recommended Value
Use Enhanced ZC Processing	Enabled

Heater Settings

The SMART MIC-1 features a heating element built into the front face of the enclosure. The purpose of the heating element is to clear condensation from each of the two mic elements, as collected condensation can significantly reduce microphone sensitivity.

When enabled, the heater can raise the temperature of the front face of the SMART MIC-1 to approximately 20°C above ambient temperature.

For a given Setting Profile, the heater can follow one of three basic behaviors:

- Alternating between active and inactive during each schedule period
- Active at the start of each schedule period
- Always inactive

You can configure the heater to run simultaneously with triggered recording or set recording to pause while the heater is on using **Triggered Recording with Bat Call Analysis > Pause Triggering When Heater Is On**.

Choosing Heater Settings

Appropriate heater settings will depend greatly on a variety of factors, including, but not limited to, the following:

Microphone Position

A microphone that is sheltered overhead may experience less moisture collection and need less frequent heating.

Microphone Orientation

A microphone that is angled below horizontal will collect less moisture than one angled above horizontal.

Environmental Factors

Temperature, humidity, and precipitation patterns at the deployment site will affect how often moisture or frost collects on the microphone and how quickly it will evaporate.

In room temperature, low-humidity conditions, with the microphone angled horizontally, it can take up to thirty minutes for the heater to remove enough moisture that a microphone returns to normal sensitivity when tested with an external speaker.

Cycle the SMART MIC-1 Heater On and Off

The heater built into the SMART MIC-1 can be set to alternate between active and inactive at regular intervals during each schedule period, starting with active.

Procedure

When editing a Setting Profile, specify both a **Heater On Period** and **Heater Off Period**, in seconds.

Example: Overnight Recording with a Heater Cycle

A common recording schedule is to use a single recording period from sunset to sunrise. For a schedule with long recording periods like this, a heater cycle can keep the microphone clear of condensation throughout each night.

Parameter	Value
Heater On Period (s)	1800
Heater Off Period (s)	5400

A microphone using a Setting Profile with these heater settings will start each recording period with the heater on for 30 minute, then off for 90 minutes. It will repeat that cycle of 30 minutes on, 90 minutes off until the recording period ends.

The ideal settings will depend on the environmental conditions at your recording site, so you should experiment to find how much heating is required to keep the microphone clear.

Enable the SMART MIC-1 Heater Once Per Schedule Period

The heater can be set to activate once for a specified duration at the start of each schedule period. For example, you can enable the heater once per day, just before sunset.

Procedure

1. When editing a Setting Profile, specify a **Heater On Period**, in seconds, and set **Heater Off Period** to 0.
2. If your Setting Profile includes the setting to disable monitoring when the heater is on, ensure your Schedule Profile starts early enough that the **Heater On Period** is finished by the time you need to begin recording.

Example: Nightly Recording Schedule with a Single Heater On Period

If you do not need to clear the microphone of moisture periodically throughout a night, or if the heater introduces too much noise into your recordings, you may choose to enable the heater only once per day, before your nightly recording period begins.

In this example, we will assume that the heater introduces significant noise into our recordings. Therefore, we will disable monitoring while the heater is on:

Parameter	Value
Triggered Recording with Bat Call Analysis	Enabled
Pause Triggering When Heater Is On	Enabled

Say we want to enable the heater for 30 minutes, then begin monitoring at sunset. We will set the heater to run once for 30 minutes (1800 seconds):

Parameter	Value
Heater On Period (s)	1800
Heater Off Period (s)	0

We will begin our schedule 30 minutes before sunset. That way, our heating period will be finished at sunset, and the SMART will begin monitoring.

```
Start: Sunset - 00:30
Duty: Always
End: Sunrise + 00:00
```

Disable the SMART MIC-1 Heater

The heater can be disabled for a given Setting Profile to reduce power consumption, but this is not recommended in most circumstances.

About this task

Disabling the heater means the SMART MIC-1 cannot clear itself of condensation or rainwater. Collected moisture will significantly reduce the microphone's sensitivity until the water evaporates. We recommend only disabling the heater if the additional power draw required by the heater would create a significant issue for your deployment.

Procedure

When editing a Setting Profile, leave the **Heater On Period** and **Heater Off Period** blank.

Calibration Settings

The SMART MIC-1 has a built-in ultrasonic speaker adjacent to the two microphone elements on the front face of the microphone. At regular intervals during its schedule, the SMART MIC-1 will emit a tone to test the sensitivity of both *microphone elements*.

Calibration can be adjusted to occur at different intervals, at a specified frequency and volume. Each time the SMART MIC-1 calibrates itself, the volume of the test tone recorded by each microphone element is logged to the Microphone Log File. The microphone elements are compared to see which sensitivity is closer to the **Calibration Target**. The result of that comparison is written to the Microphone Log File, and the closer microphone is used for recording until the next calibration occurs.

Scheduling Calibration

Each calibration test takes roughly three to four seconds.

If a **Calibration Period** is not specified in the active Setting Profile, calibration will never occur, and the microphone will always use its primary microphone element.

If a **Calibration Period** is specified, calibration occurs once at the beginning of each recording period. Calibration will then occur once per **Calibration Period** until the end of each recording period.

Example: Calibration Once per Hour

A typical recording schedule is to use one continuous recording period from sunset to sunrise:

```
Start: Sunset + 00:00
Duty: Always
End: Sunrise + 00:00
```

With this schedule and the default **Calibration Period** value of 3600 seconds, calibration will occur at sunset, then once every hour until sunrise.

Calibration Frequency

Calibration Frequency sets the frequency of the test tone generated by the SMART MIC-1 to test each microphone element's sensitivity.

The **Calibration Frequency** can be set up to half the selected **Sample Rate** setting, but the built-in test speaker was designed primarily for use at 40 kHz. You should only use a different value if you have a specific requirement to test each microphone element's sensitivity at a particular frequency.

Setting	Recommended Value
Calibration Frequency	40 kHz

Calibration Amplitude and Target

DANGER: Do not place the SMART MIC-1 close to your ears! The built-in calibration speaker can produce audible sound or inaudible ultrasound that, at short distances, may damage your hearing.

The amplitude of the calibration tone can be specified as a ratio to *full scale*, the maximum output volume of the calibration speaker. We recommend leaving the **Calibration Amplitude** parameter at its default value of 1.

When the SMART MIC-1 performs calibration, it determines which of its two microphone elements produces a measurement closer to the **Calibration Target**, and it uses that element for recording until the next calibration. This is intended to keep the overall system sensitivity relatively constant as one microphone element or the other may be permanently damaged or temporarily muffled by condensation.

Note that the microphone element that measures closer to the **Calibration Target** is chosen. This is not necessarily the more sensitive microphone element.

Setting	Recommended Value
Calibration Amplitude	1
Calibration Target	Situational

Example: The Less Sensitive Microphone May Be Chosen

A SMART MIC-1 is using the default **Calibration Target** of -25 dB relative to full scale (dBFS). When the microphone performs calibration, it reads these measurements:

- Primary element: -15 dBFS
- Backup element: -26 dBFS

The SMART MIC-1 will use the backup microphone element until the next calibration period, even though it is the less sensitive of the two microphones.

If you always want the SMART MIC-1 to use the more sensitive of its two microphone elements, set the **Calibration Target** to 0 dBFS.

Example: The More Sensitive Microphone Is Always Chosen

The maximum amplitude the SMART MIC-1 can register is equivalent to 0 dBFS. If the **Calibration Target** is set to 0 dBFS, then the most sensitive microphone will more closely match the target and will be used until the next calibration.

A SMART MIC-1 has its **Calibration Target** set to 0 dBFS. When the microphone performs calibration, it reads these measurements:

- Primary element: -15 dBFS
- Backup element: -26 dBFS

The SMART MIC-1 will use the primary microphone element until the next calibration because it is closer to the target reading.

Settings for Triggered Recording

Many settings available on the **Setting Profiles** tab only apply when **Triggered Recording with Bat Call Analysis** is enabled. These settings affect what kinds of signals will trigger recording and how the SMART will segment and process the resulting audio.

Pulse-Level Trigger Settings

During triggered recording with bat call analysis, several settings define what signals will trigger recording and be analyzed as bat calls. On the level of individual echolocation pulses, the SMART System can filter signals by frequency range and duration range.

You should tailor these settings to ignore sounds other than bat calls as much as possible, while including the vocalizations of your bat species of interest.

In *ATC* deployments, these criteria may not be enough to completely separate bat calls from the noise produced by a given wind turbine. SMART's bat acitivity alarms provide a much more flexible logic system, which can be used as a second layer of filtering after the trigger parameters have taken effect.

Important:

The following processes will only occur on bat calls that satisfy the pulse-level trigger parameters:

- triggered recording
- measurement of bat call characteristics
- bat species auto-ID
- bat activity alarm logic

Minimum and Maximum Frequency

The **Minimum Frequency** and **Maximum Frequency** parameters set the frequency range in which the SMART will look for possible bat calls. If a signal contains components both inside and outside this frequency range, only the portions inside the range will be analyzed.

You should set the **Minimum Frequency** and **Maximum Frequency** settings to encompass the frequency range of any bat calls you plan on recording. These settings prevent signals that are too low or too high in frequency to be a bat call from triggering recording.

Setting	Recommended Value for Bat Monitoring
Minimum Frequency	Slightly lower than the minimum frequency of nearby bat vocalizations
Maximum Frequency	Slightly higher than the maximum frequency of nearby bat vocalizations

Minimum and Maximum Pulse Duration

The **Minimum Pulse Duration** and **Maximum Pulse Duration** parameters set how long or short a signal must be to trigger recording and be analyzed as a bat call. The typical duration of bat calls depends on the bat species and on the bat's behavior.

Ideally, the **Minimum Pulse Duration** should be set to distinguish between very short signals (such as the sounds of rain drops) from even the shortest calls of nearby bats. Note that approach-phase and feeding buzz calls are typically much shorter than a given species' search-phase calls.

The **Maximum Pulse Duration** should be set to distinguish even the longest bat calls from other, longer ultrasonic sounds, such as the whine of nearby machinery.

Setting	Recommended Value
Minimum Pulse Duration	Slightly shorter than the shortest calls of nearby bat species
Maximum Pulse Duration	Slightly longer than the longest calls of nearby bat species

Pass-Level Trigger Settings

The **Maximum Sequence Duration**, **Pre-Trigger**, and **Post-Trigger** settings directly determine the maximum duration of your triggered recordings. The **Maximum Inter-Pulse Gap** also affects how SMART splits audio into separate files depending on the recorded bat activity.

The **Pre-Trigger** period is a rolling buffer of audio that is saved in RAM until an ultrasonic signal triggers the microphone to start recording. From that point, the SMART will record for the length of the **Maximum Sequence Duration** or until there is a gap between echolocation pulses at least as long as the **Maximum Inter-Pulse Gap**. Then, the SMART will record for the length of the **Post-Trigger** period to complete the file.

Example: Maximum Sequence Duration, Pre-Trigger, and Post-Trigger

In this example, settings for a particular microphone are as follows:

Setting	Value
Maximum Duration	Blank
Triggered Recording with Bat Call Analysis	Enabled
Pre-Trigger	0.25 s
Post-Trigger	1 s
Maximum Inter-Pulse Gap	0.5 s
Maximum Sequence Duration	15 s

During the microphone's recording period, it will keep a rolling Pre-Trigger buffer of 0.25 s stored in RAM. When a bat pulse is detected by the microphone, that 0.25 s of audio will be saved as the beginning of the recording file. While bat pulses continue to be recorded with gaps no longer than 0.5 s in between, the SMART will continue to save audio to the recording.

When 15 s has elapsed since the initial bat pulse, the total length of recorded audio will be 15.25 s, including the Pre-Trigger buffer. At this point, the **Maximum Sequence Duration** has been reached. The SMART will record the Post-Trigger period for 1 s and include that at the end of the recording file.

The total length of the recording file will be 16.25 s.

Example: Maximum Inter-Pulse Gap

In this example, settings for a particular microphone are the same as in Maximum Sequence Duration, Pre-Trigger, and Post-Trigger *(on page 32)*:

Setting	Value
Maximum Duration	Blank
Triggered Recording with Bat Call Analysis	Enabled
Pre-Trigger	0.25 s
Post-Trigger	1 s
Maximum Inter-Pulse Gap	0.5 s
Maximum Sequence Duration	15 s

During the microphone's recording period, it will keep a rolling Pre-Trigger buffer of 0.25 s stored in RAM. When a bat pulse is detected by the microphone, that 0.25 s of audio will be saved as the beginning of the recording file.

Say that 5 s after the initially detected pulse, the bat emits a final pulse and leaves the microphone's detection range. 0.5 s later, the SMART will end the sequence and record an additional 1 s, corresponding to the **Post-Trigger** parameter.

The total length of the recording will be 6.75 s.

Maximum Sequence Duration

During triggered recording, **Maximum Sequence Duration** sets a limit on what the SMART System considers a single *sequence* of bat calls.

A sequence, also called a *pass*, is one series of bat echolocation calls recorded as a single file. Ideally, one sequence should represent a single bat flying through a microphone's detection range, as algorithmic species identification works best on recordings of an individual bat. In reality, it is not always possible to separate echolocation pulses into distinct sequences, particularly when multiple bats are active near the microphone simultaneously. If the SMART detects a series of pulses without a gap long enough to meet the **Maximum Inter-Pulse Gap**, the **Maximum Sequence Duration** parameter will prevent a single recording from being too long to be useful for later analysis.

The maximum length of a triggered recording file is the **Maximum Sequence Duration**, plus the **Pre-Trigger** and **Post-Trigger** durations.

Recommended Maximum Sequence Duration

Some bat survey protocols dictate a maximum recording length. If you are following such a protocol, set the **Maximum Sequence Duration**, **Pre-Trigger**, and **Post-Trigger** parameters such that the sum of all three does not exceed the dictated maximum length.

Otherwise, you can adjust the **Maximum Sequence Duration** based on your particular situation. If you want to decrease the likelihood that multiple bats are recorded in a single recording, decrease the setting. If you want to descrease the likelihood that a single bat's calls are split into multiple recordings, increase the setting.

Pre-Trigger and Post-Trigger Recording

During triggered recording, the SMART has the option to append **Pre-Trigger** and **Post-Trigger** recording periods to each triggered recording.

A typical use case for these settings would be to record any echolocation pulses that are loud enough to be discernible on a spectrogram, but not loud enough to be detected as pulses by the SMART's detection algorithm.

If you decide to use the **Pre-Trigger** and/or **Post-Trigger** settings, keep the following in mind:

- Echolocation pulses that are not loud enough to trigger recording on their own will not be processed by SMART's auto-ID algorithm or bat activity alarms, even if they are recorded to audio during the pre- or post-trigger period.
- It is possible that the pre- or post-trigger period of one recording may overlap with the beginning or ending of another recording. In this case, the same time period may be recorded as part of two separate recordings.

Example: Pre-Trigger and Post-Trigger Recording Periods
Imagine a bat flies in a straight path that passes near a microphone. Without pre-trigger and post-trigger recording periods and with other settings at typical values, the SMART will only record the bat's

activity while it is close enough that its echolocation pulses are loud enough for the SMART to identify them as pulses.

If **Pre-Trigger** is set to 1 second, the SMART will additionally save one second of audio before the bat flew close enough to trigger recording. This may include echolocation pulses that a human vetter can see on a spectrogram but are not loud enough to have triggered recording.

Likewise, if **Post-Trigger** is set to 1 second, the SMART will record one additional second after the bat has flown away from the microphone, possibly including pulses that were too faint for the SMART to identify but loud enough to be discernible to a human vetter.

Maximum Inter-Pulse Gap

The **Maximum Inter-Pulse Gap** parameter sets the maximum duration of silence there can be after a echolocation pulse before the SMART System ends a given sequence.

You should set this parameter long enough that the calls of a single passing bat are not split into multiple files but short enough that multiple bats passing in succession are not treated as single passes too frequently.

Example: Maximum Inter-Pulse Gap Behavior
Many bat species produce search-phase calls with 200-400 ms between each pulse. Say the **Maximum Inter-Pulse Gap** parameter is set to 500 ms. During a triggered recording, if one pulse is not followed by another pulse within 500 ms, the SMART System will end the sequence. The recording in progress will end after the **Post-Trigger** interval, if set. The next pulse to be detected will trigger a new sequence.

Minimum Number of Pulses

The **Minimum Number of Pulses** parameter defines how many pulses must have been detected in a single sequence for the audio to be saved and analyzed.

A triggered sequence will end either when the period of silence after a pulse exceeds the **Maximum Inter-Syllable Gap** or if the total duration, beginning from the first pulse, exceeds the **Maximum Sequence Duration**. When the sequence ends, the SMART System checks whether the sequence contains the **Minimum Number of Pulses** or more. If it does, the SMART System will save the audio to its internal drive, run bat auto-ID on the recording, and process the pass-level bat call analysis measurements through any pass-level alarms.

Every detected pulse is processed through pulse-level alarms, even if the number of pulses in a given sequence does not meet the minimum threshold. If **Minimum Number of Pulses** has a value greater than 1, this can mean that individual pulses are processed through pulse-level alarms without being saved as audio recordings. However, setting **Minimum Number of Pulses** to a value greater than one can help reduce the number of recordings triggered by non-bat sounds.

Constant-Frequency (CF) Filter

The SMART System has a built-in filter to avoid triggering on signals within a set of defined parameters. This filter can help avoid false triggers on signals such as electrical interference or mechanical squeaks.

The constant-frequency filter accepts a maximum frequency, minimum frequency, and maximum bandwidth. The SMART System will prevent a signal from triggering recording if the signal meets all the following conditions:

- The signal's minimum frequency is greater than or equal to the **CF Filter Min Frequency**.
- The signal's maximum frequency is less than or equal to the **CF Filter Max Frequency**.
- The difference between the signal's maximum frequency and minimum frequency is less than or equal to the **CF Filter Max Bandwidth**.

If you use the CF Filter, you should make sure that none of the echolocation calls produced by nearby bat species meet all three of the above parameters.

Example: Using the CF Filter on a Wind Turbine
A common reason to use the CF filter is when a particular wind turbine produces high-frequency squeaks as it rotates. If these squeaks are similar in duration and frequency to typical bat calls, you might get many recordings of these squeaks, and they may trigger your bat activity alarms.

Say that a turbine produces squeaks that fall between 20 kHz and 30 kHz. Each individual squeak is narrow-band and flat in frequency: a single squeak spans less than 1 kHz between its highest and lowest frequency.

To prevent these squeaks from triggering the SMART, we could use the following parameters:

CF Filter Min Frequency	20 kHz
CF Filter Max Frequency	30 kHz
CF Filter Max Bandwidth	1 kHz

Before implementing these settings, we should make sure that the bat echolocation calls we want to record do not meet all of these parameters:

Say there is a nearby species of bat that produces very flat calls around 80 kHz. Even if the SMART detects one of those calls with a bandwidth less than 1 kHz, it will still trigger since the call is higher in frequency than the **CF Filter Max Frequency**.

There may be many other species of bats whose calls fall, in part, between 20 kHz and 30 kHz. As long as each of those calls has a greater bandwidth than 1 kHz, the filter will not interfere with those calls. However, if a call is faint enough that the SMART only registers a narrow portion of it, that call may be removed by the constant-frequency filter.

Bat Auto-ID Classifier Settings

The SMART System can automatically identify each saved recording by bat species using the built-in Kaleidoscope Pro Sound Analysis software.

If **Bat Analysis / Triggering** is enabled, the SMART System will measure key features of detected bat pulses whether bat auto-ID is enabled or not. These measurements are recorded in each microphone's `results.csv` file and are available for bat activity alarm processing. If a bat auto-ID classifier is selected, Kaleidoscope will also assign a species ID to each triggered recording. The results of this identification are saved to the `results.csv` file, and they are also saved to the metadata of each recording. Note that these results do not play a role in the SMART System's bat activity alarms.

All automatic bat species identification systems make mistakes. Always have a human expert verify auto-ID results.

Note that the species assigned in the `results.csv` file on the SMART System may vary from the results of Kaleidoscope Pro running on a desktop system because of differences in how the software processes and labels recordings of non-bat noise.

Enable Bat Auto-ID

You can enable bat species auto-ID by manually selecting species for the algorithm to include as possible results or by selecting species known to be present in a given region.

Procedure

1. Open the SMART Control Panel and go to **Setting Profiles**.
2. Select the Setting Profile you want to use with bat auto-ID.

3. Select a **Classifier** corresponding to your area of the world.

Classifier Name	Area of the World
Bats_of_South_Africa_5.4.0	South Africa
Bats_of_North_America_5.4.0	United States and Canada
Bats_of_the_Neotropics_5.4.0	Mexico, Central America, South America, and the Caribbean
Bats_of_Europe_5.4.0	British Isles, Europe, and parts of West Asia

4. Select a **Classifier Threshold** to influence the likelihood of *NoID* results.

Threshold Setting	Description
(-1) Sensitive	Species results are more likely for ambiguous recordings.
(0) Balanced	Setting in between **Sensitive** and **Accurate**.
(+1) Accurate	*NoID* results are more likely for ambiguous recordings.

5. Select which species should be considered as possible results.
    a. **Optional:** Use the **Populate species from region** drop-down to select a country, province, or state.
    Species known to be present in that region will be selected in the species list.
    b. **Optional:** Manually select or deselect species to include only species you know to be present in your area.
6. Select **Apply Changes**.

SMART MIC-1 Schedules

Each microphone attached to a SMART System can run an independent schedule, which determines when the microphone will monitor for bat activity each day.

You can design custom schedules using a simple set of commands that allow for a wide range of schedule behaviors.

Create a New Schedule

You can create a new schedule for any of your SMART microphones using the SMART Control Panel.

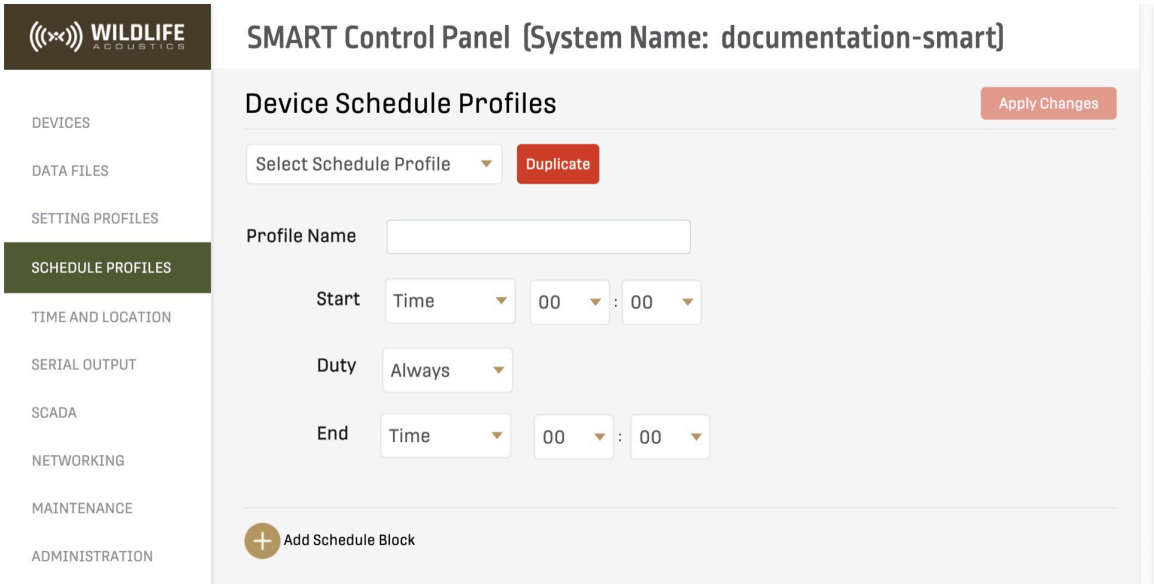
Before you begin

To ensure your recording schedule runs as intended, you must accurately set the following settings on the **Time and Location** tab:

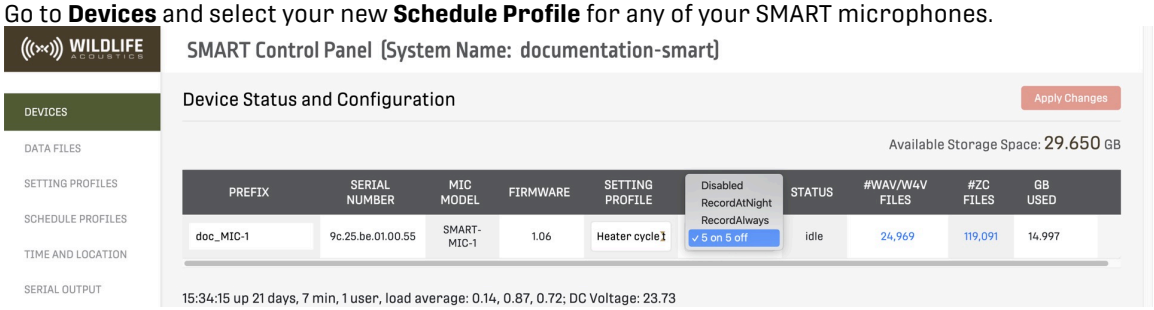
- **Date** and **Time**
- **Time Zone**
- **Location**

Procedure

1. Open the SMART Control Panel and go to **Schedule Profiles**.
By default, no Schedule Profile is selected, and you can create a new one.



2. Enter a **Profile Name** and adjust the **Start**, **Duty**, and **End** parameters for the first schedule block.
3. If desired, select **Add Schedule Block** and adjust the parameters for any additional blocks.
4. Select **Apply Changes** to save the Schedule Profile.
5. Go to **Devices** and select your new **Schedule Profile** for any of your SMART microphones.



Example Schedule: 30 Minutes Before Sunset to 30 Minutes After Sunrise

This schedule is very commonly used for bat monitoring. It records all night with some additional recording at dusk and dawn.

```
Start: Sunset - 00:30
Duty: Always
End: Sunrise + 00:30
```

These commands tell the SMART to begin monitoring 30 minutes before sunset, then monitor continuously until 30 minutes after sunrise.

Schedule Blocks

A microphone schedule is comprised of up to ten blocks, each of which allows you to define a start point, end point, and an optional on-off cycle to occur during the block. Many of the most common recording schedules can be configured using only one or two schedule blocks.

To explain how schedule blocks work in more detail, we'll start with an example schedule comprised of one block:

```
Start: Time 00:00
Duty: Cycle
    On: 00:05
    Off: 00:55
End: Time 00:00
```

This single block tells the SMART MIC-1 to record for five minutes every hour. The following sections expand on how each component of the schedule block works.

Start and End

```
Start: Time 00:00
...
End: Time 00:00
```

The **Start** and **End** commands are best understood in relation to each other. Everything the schedule block does occurs after the **Start** and continues until it reaches the **End**. This is the case even when the **Start** and **End** share the same value, as they do here. The schedule block will begin every time it reaches 00:00 (12:00 AM on the 12-hour clock) and it will do whatever you program it to do until it reaches the next 00:00. Each midnight, the recorder will simultaneously end one cycle of this schedule block and start a new one.

This same logic means that a single schedule block can span two calendar dates:

```
Start: Time 17:00
...
End: Time 05:00
```

In this example, the schedule block will start running every time the clock hits 17:00 (5:00 PM) and will end the next time the clock hits 05:00 (5:00 AM), which will be the next day.

While this may seem obvious, understanding this now is important for when we later introduce schedule blocks that start and end relative to sunrise and sunset.

Duty Cycle

In the engineering world, *duty cycle* describes a pattern of repeating activity and inactivity. This could be a pattern of "on" and "off" signals in a computer that alternate millions of times per second, or it could be a schedule whereby a manufacturing machine is run for three hours every day.

In SMART microphone schedules, a duty cycle tells the microphone to alternate between recording and sleeping, at fixed time intervals, in between a schedule block's **Start** and **End** points.

When **Duty** is set to Cycle, the schedule block will begin at the **Start** point by recording, then it will pause, then it will record, then it will pause, repeating this until the **End** point is reached. When **Duty** is set to Always, the SMART will simply monitor for the entire length of the schedule block.

Returning to the original schedule block example, the duty cycle section of the schedule block is:

```
Duty: Cycle
    On: 00:05
    Off: 00:55
```

At the **Start** point, the recorder begins at the **On** phase by recording for five minutes. Then, it runs the **Off** phase by sleeping for 55 minutes. It will alternate between five minutes of recording and 55 minutes of sleeping for as many time as it takes to reach the schedule block's **End** point.

The **End** point will immediately end the schedule block in the middle of either the **On** or **Off** phase. If the time span between the **Start** and **End** points is short enough, it is possible to end a schedule block before the recorder can complete a single cycle.

Schedules Using Sunrise and Sunset

Schedule blocks can start and end relative to sunrise or sunset, allowing them to automatically shift with seasonal changes.

A schedule block's **Start** and **End** points can be set at fixed times on the clock, or they can be set at some fixed offset from sunrise or sunset. If either point is set relative to sunrise or sunset, the SMART uses its location, date, and time zone information to calculate when sunset and sunrise occur on each day in order to time its schedule.

As a common example, we can program a schedule to record overnight from 30 minutes before sunset to 30 minutes after sunrise.

```
Start: Sunset - 00:30
Duty: Always
End: Sunrise + 00:30
```

Start: Offset from Sunset

Sunset means the **Start** point is defined relative to the time of sunset on a given day. The negative sign, -, means the block starts earlier than sunset, and the offset, 00:30, means it starts 30 minutes prior to sunset.

If you want the schedule block to start exactly at sunset, you should set the offset to **00:00**. The value of the positive or negative sign does not matter in that case.

Duty Cycle: Always

Setting **Duty** to Always means the schedule block runs from the **Start** to the **End** with no pauses in between.

When Always is selected, the **On** and **Off** commands are not used.

End: Offset from Sunrise

The structure of the **End** command is the same as for the **Start** command. Selecting Sunrise means we define the **End** for this schedule block relative to the time of sunrise on each day. The positive sign, +, means we want to specify a time after sunrise occurs. 00:30 specifies 30 minutes after sunrise.

Schedules with Multiple Blocks

Each SMART microphone schedule can be made of up to ten schedule blocks. Blocks can be set independently, and overlapping recording periods from separate blocks are automatically combined.

Ordering of Blocks

Within a schedule, it does not matter which block is positioned at the top of the list and which is at the bottom. The order does not mean one block is prioritized over another.

Block Independence

Within a single schedule, each block can have entirely different commands from each other. For example, the first block could have its **Start** and **End** set as fixed times on the clock, with no duty cycle, and the second block could reference sunrise and sunset for either or both of its **Start** and **End** points and use a **Duty Cycle**.

Overlapping Blocks

When two schedule blocks have recording periods that overlap, the SMART treats the overlapping recording periods as a single, combined recording period.

For example, if block 1 includes a recording period from 13:45 to 14:15, and block 2 includes a recording period from 14:00 to 15:00, the SMART will treat the time from 13:45 to 15:00 as one single recording period.

This is true regardless of whether each block is set to reference fixed times on the clock or sunrise/sunset. This is also true whether each recording period is a single, continuous schedule block or one recording period out of a longer, cyclical schedule block.

Change Microphone Name (Prefix)

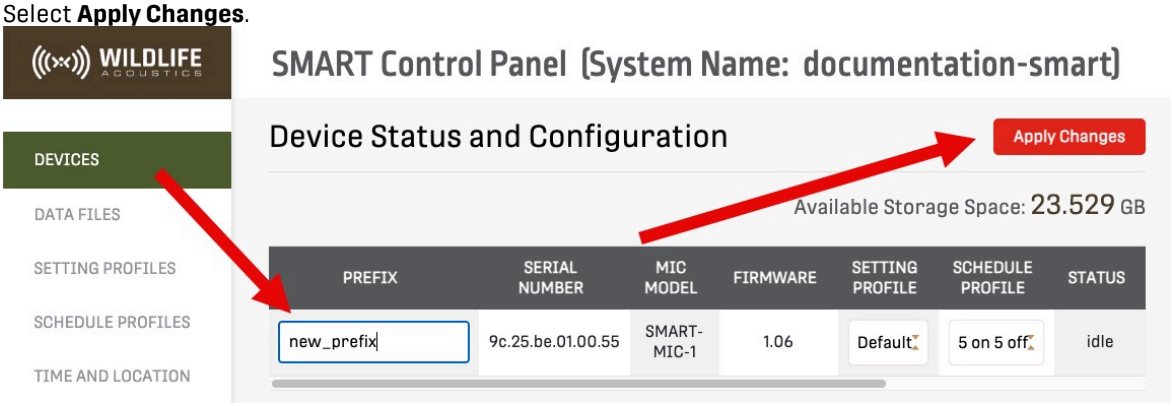
Each microphone attached to a SMART Controller can be given a unique name, or *prefix*. This prefix is used as part of the file name of any recording, results table, or log file associated with the microphone.

About this task

The default prefix for each microphone is its serial number, shown in a dedicated column on the **Devices** tab.

Procedure

1. Open the SMART Control Panel and go to **Devices**.
2. Enter a new value into the **Prefix** column for any listed microphone.
3. Select **Apply Changes**.



Results

The new prefix will be applied to any files produced going forward. It will not be applied to files recorded before the change.

4. SMART Controller Configuration

Before you deploy the SMART System, you should configure basic system and network settings. These include login credentials, date, location, and network configuration settings.

Default System and Network Settings

These default settings are required information when configuring the SMART System for the first time. You should change all passwords before you deploy the SMART System.

Setting	Default Value
System Name	`smart`
Wi-Fi Access Point SSID	`smart-`*`MAC_Address`*, where *MAC_address* is the SMART System's twelve-digit MAC address without periods.
Wi-Fi Access Point Password	`wildlife`
Wi-Fi Access Point IP Address	`192.168.19.1`
`smart` User Password	`wildlife`

Configure Basic System Settings

Before you deploy the SMART System, you should configure basic system settings, including time, location, and credentials.

Configure Time, Date, and Location Settings

In order to run schedules correctly and save accurate recording metadata, you must ensure the time, date, and location settings are accurate.

About this task

The SMART System has three options for setting its internal clock, depending on whether your SMART System is connected to the global Internet, a private network, or neither.

- If the SMART Controller is connected to the global internet, then it can synchronize its internal clock to the default time server, `ntp.ubuntu.com`.
- If the SMART Controller is not connected to the global internet due to lack of connectivity or due to network security restrictions, you or an IT professional can connect the SMART to a *Network Time Protocol* server on your private network. Then, you can set the SMART Controller to synchronize its internal clock with that server.
- If the SMART Controller lacks a connection to the internet or a local *NTP* server, you must set its date and time manually.

Procedure

1. Open the SMART Control Panel and go to **Time and Location**.
2. Set the date and time.

- If the SMART Controller is connected to the global internet or a local *NTP* server, select **Enable automatic time synchronization service**.
- If the SMART Controller is not connected to the global internet or a local *NTP* server, adjust the date and time settings using the menus and select **Change Date** and **Change Time** to confirm the settings changes.

3. Select a time zone.
4. Enter latitude and longitude coordinates and select **Change Location** to confirm the change.
   In order to follow schedules timed to sunset or sunrise, the location must be accurate to within approximately 0.25° latitude and longitude.

Results
The location data entered here will be saved to the SMART System's bat analysis results tables and to the metadata of saved recordings.

Related information
Network Configuration *(on page 45)*

Connect SMART to a Local Network Time Server

In some SMART deployments, access to the default time server (`ntp.ubuntu.com`) is not possible due to firewall and/or cyber-security policies. You can instead synchronize your SMART Systems to a network time server on your local network.

About this task
The SMART Controller's Ubuntu operating system has a time service, `systemd-timesyncd`, that uses a configuration file, `timesyncd.conf`, to control its operation. This file defines the location of *NTP* servers on the reachable network.

Procedure
1. Access the SMART Controller terminal from an SSH session or locally, using a monitor and keyboard.
2. Enable superuser privileges and navigate to `/etc/systemd`.

```
smart@smart:~$ sudo bash
root@smart:/home/smart# cd /etc/systemd
```

3. Open the file `timesyncd.conf` with your text editor of choice.

```
root@smart:/etc/systemd# nano timesyncd.conf
```

The file should look like this:

```
#  This file is part of systemd.
#
#  systemd is free software; you can redistribute it and/or modify it under the
#  terms of the GNU Lesser General Public License as published by the Free
#  Software Foundation; either version 2.1 of the License, or (at your option)
#  any later version.
#
# Entries in this file show the compile time defaults. Local configuration
# should be created by either modifying this file, or by creating "drop-ins" in
# the timesyncd.conf.d/ subdirectory. The latter is generally recommended.
# Defaults can be restored by simply deleting this file and all drop-ins.
#
# See timesyncd.conf(5) for details.

[Time]
```

```
#NTP=
#FallbackNTP=ntp.ubuntu.com
#RootDistanceMaxSec=5
#PollIntervalMinSec=32
#PollIntervalMaxSec=2048
```

4. Edit the line `#NTP=` by removing the '#" and appending your local time server's IP address, then save the file.
   Example
   `NTP=10.0.5.232`
   If needed, you may enter more than one IP address separated by commas.
5. Restart the `timesyncd` service.
   Example
   ```
   root@smart:/etc/systemd# systemctl restart systemd-timesyncd
   ```
6. Verify that your SMART controller is synchronized to your local time server using the command `systemctl status systemd-timesyncd`.
   ```
   root@smart:/etc/systemd# systemctl status systemd-timesyncd
   ```
   The status readout should indicate synchronization with the time server you specified.
   Example
   ```
   systemd-timesyncd.service - Network Time Synchronization
     Loaded: loaded (/lib/systemd/system/systemd-timesyncd.service; enabled; vendor
    preset: enabled)
     Active: active (running) since Tue 2023-08-08 12:17:37 EDT; 1min 31s ago
       Docs: man:systemd-timesyncd.service(8)
   Main PID: 92886 (systemd-timesyn)
     Status: "Initial synchronization to time server 10.0.5.232:123."
      Tasks: 2 (limit: 4344)
     Memory: 1.2M
        CPU: 168ms
     CGroup: /system.slice/systemd-timesyncd.service
             └─92886 /lib/systemd/systemd-timesyncd
   ```
7. Open the SMART Control Panel and go to **Time and Location**.
8. Enable the automatic time synchronization service if it is not enabled already and confirm that the status displays the correct time.

Related information
Local Network Configuration *(on page 48)*

Time Zones and Daylight Saving Time

The SMART System will automatically follow the Daylight Saving Time (DST) convention for the selected time zone in the **Time and Location** tab. For SMART Systems without network connectivity, any future changes to local DST conventions will require system updates.

If a SMART System is connected to the Internet or to a local network with a *NTP* server, and if **Automatic time synchronization service** is enabled in the **Time and Location** tab, the SMART System will follow DST conventions as dictated by the *NTP* server.

If a SMART System is not connected to an *NTP* server, it will follow the time zone conventions saved in the tz database installed in Ubuntu. If laws about the use of DST change, you will need to update this database so that the SMART System will follow the new convention.

Change Password

From within the SMART Control Panel, you can change the password for the user smart.

About this task

The same password is used to log in as the user smart both in the SMART Control Panel and the Ubuntu command line.

CAUTION: If you lose your password, reinstalling Ubuntu is the only way to regain access to the SMART Controller. Reinstallation will erase all data stored on the SMART Controller's internal drive.

Procedure

1. Open the SMART Control Panel and go to **Administration**.
2. Enter a new password into both **Password** fields.
3. Select **Change Password**.

Change System Name

Each SMART Controller can be given a name to distinguish it from other SMART Controllers.

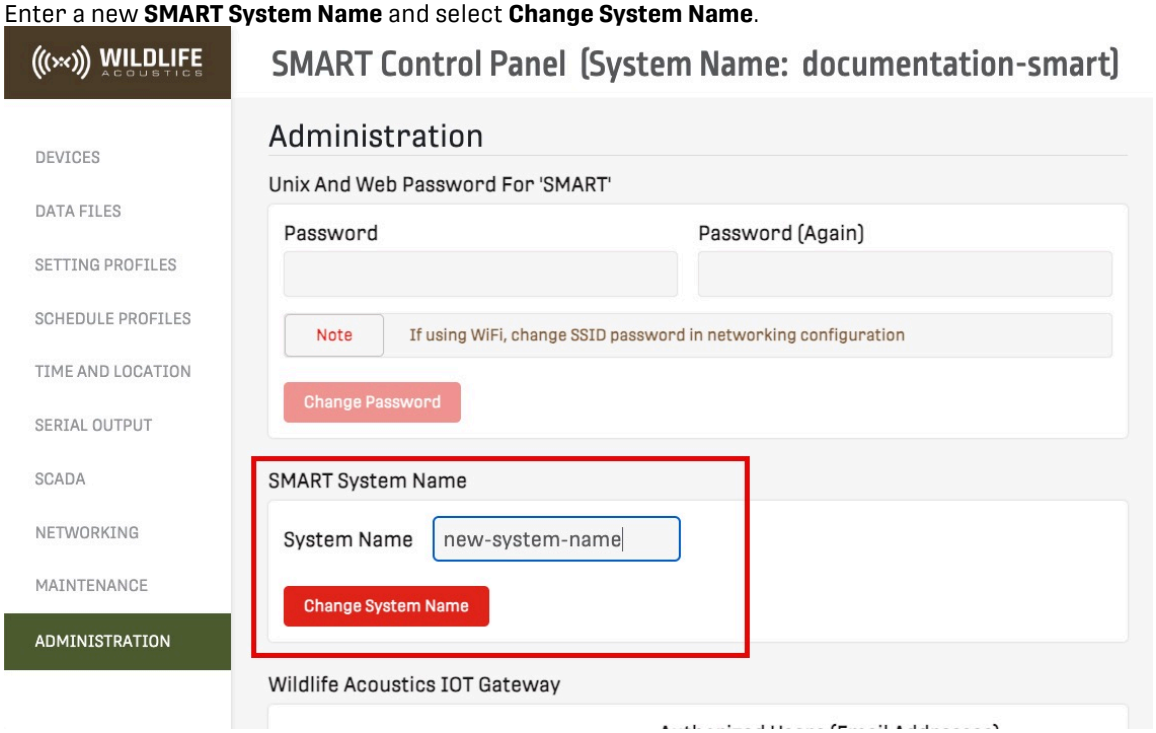
About this task

The SMART System name appears in multiple locations:

- SMART Portal
- SMART Control Panel
- Results and log files
- Recording metadata

Procedure

1. Open the SMART Control Panel and go to **Administration**.
2. Enter a new **SMART System Name** and select **Change System Name**.



Results

The top of the page will display the message `Hostname changed to XYZ`.

Network Configuration

Depending on your goals and network security requirements, you might choose to network your SMART System in several different ways. This can range from a completely disconnected system to Internet-connected and globally configurable.

No Network Connectivity

In deployments without available networking infrastructure or with strict security requirements, the SMART can be disconnected from all existing public or private networks. In this configuration, you might only access the SMART System using a direct keyboard and monitor connection, or you might network a laptop directly to the SMART via Wi-Fi or Ethernet.

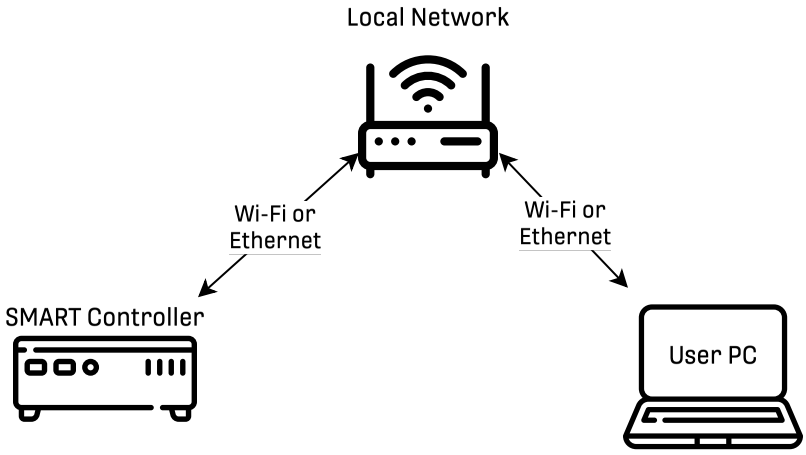
Figure 6. SMART Controller as Wi-Fi Access Point

Private Network Connectivity

A SMART System can be connected to a private network (also known as a *local network*) without a connection to the public Internet. In this configuration, users with access to the private network can be allowed to access

the SMART remotely to retrieve data or configure its settings. A private network connection also allows the SMART to back up its data to a storage location on the same network.

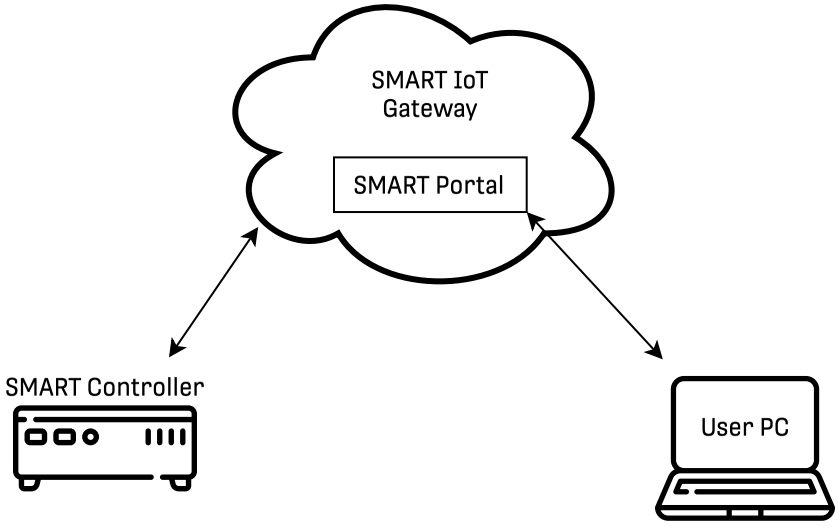
Figure 7. SMART Controller as Client of Local Network



Internet Connectivity

A SMART System can be connected to the *SMART IoT Gateway*, hosted on a Wildlife Acoustics server, via an existing local network or via a direct Internet connection, such as a cellular modem. You can then configure a connected SMART for use with the *SMART Portal*. The SMART Portal is a secure system that allows authorized users to manage multiple SMART Systems from any web browser via their Wildlife Acoustics account.

Figure 8. SMART Controller Connected to the SMART IoT Gateway and SMART Portal



Related information

Configure Time, Date, and Location Settings *(on page 41)*

Standalone Configuration

In scenarios where connecting the SMART to a local network is not possible for reasons of security or infrastructure, the SMART can run independently.

In order to configure the SMART or access its data in this configuration, you must visit the SMART in-person. You can configure the SMART as a Wi-Fi access point to configure it using a nearby computer, directly

connect a computer to the SMART using Ethernet, or you can connect a monitor and keyboard to directly access SMART's Ubuntu shell.

Storage Considerations for Standalone Configuration

In standalone configuration, automatic data backup is limited to a single, USB-connected drive. If you plan on saving recorded audio, you should make sure you have a large enough storage device to save data in between visits to swap storage drives. There are two ways to reduce the storage space used by audio files:

- Enable WAV compression to reduce the size of full-spectrum audio by 50-75%.
- Disable WAV output entirely and only save ZC files, which are roughly one thousand times smaller than equivalent WAV files.

Configure the SMART Controller as a Wi-Fi Access Point

The SMART Controller's default configuration is as a Wi-Fi access point. This allows you to configure the SMART System from a computer without connecting to any other network infrastructure.

About this task

Connecting to the SMART Controller as a Wi-Fi access point provides a simple way to set up the SMART for the first time. It can also be used in deployments where the SMART is disconnected from a broader network, but you need to configure the SMART using a laptop on-site.

Procedure

1. Connect to the SMART Control Panel and go to **Networking**.
2. Set **Wi-Fi Configuration** to **Access Point**.
3. Enter an **SSID** and **Password** for the SMART Controller's Wi-Fi network.

   Default SSID

   `smart-`*`MAC_Address`*, where *MAC_address* is the SMART system's twelve-digit MAC address without periods.

   Default Wi-Fi Password

   `wildlife`
4. Enable **DHCP Server Configuration**.
5. Select **Apply Changes**.

Results

The SMART Controller will host a Wi-Fi network with your selected **SSID** and **Password**. Using a Wi-Fi-enabled device with a web browser, you can now join this network and access the SMART Control Panel via the SMART Controller's default IP address: `192.168.19.1`.

Check the SMART Controller's Wi-Fi Network IP Address

When the SMART Controller is configured to host a Wi-Fi network, you will need its IP address to access the SMART Control Panel.

About this task

When the SMART Controller is configured as a Wi-Fi hotspot, its default IP address is `192.168.19.1`. If you have changed this setting in the past, you can check its current value via the SMART Controller's Ubuntu terminal.

Procedure

1. Connect a monitor and keyboard to the SMART Controller and log into the Ubuntu terminal.
2. Enter the command `ifconfig wlo1 | grep "inet "`.

Results

The IP address associated with the SMART's Wi-Fi interface is printed after `inet`.

Local Network Configuration

Connecting the SMART to a local network can allow you to configure the SMART remotely, with or without connecting the SMART to the global Internet.

Without connecting the SMART to the global Internet, you can use another device with access to the same local network to access the SMART System's data and configure its settings. This may mean that the second device must be directly connected to a private intranet, or it may allow you access from anywhere via a VPN.

Connecting via Ethernet or Wi-Fi

By default, the SMART Controller can be connected to a local network via Ethernet. By default, the SMART's Ethernet interface is configured to accept an IP address assigned by a DHCP server.

The SMART's Wi-Fi interface is configured to host a local network by default. You can configure it to instead connect to a local network as a Wi-Fi client.

Related information

Connect to the SMART Control Panel for the First Time *(on page 18)*
Connect SMART to a Local Network Time Server *(on page 42)*

Connect the SMART Controller to a Local Network via Ethernet

The SMART Controller can be connected to a nearby router or network switch via Ethernet.

Procedure

1. Open the SMART Control Panel and go to **Networking**.
2. If **Advanced Network Configuration** is selected, switch to Basic Network Configuration.
3. Set the **Ethernet Configuration** to use either a Static IP or Dynamic IP (via DHCP).

   If you select **Static IP**, you can also set the following settings:

   - **IP Address**
   - **Mask Bits (0-32)**
   - **Default Gateway IP**
   - **Primary Name Server IP**
   - **Secondary Name Server IP**
4. Disable **DHCP Server Configuration**.
5. **Note:** If you have accessed the SMART Control Panel via an Ethernet connection, applying these changes will cause you to lose connection to the SMART Control Panel.

   Select **Apply Changes**.

Connect the SMART Controller to a Local Network via Wi-Fi

The SMART Controller comes equipped with two Wi-Fi antennas and can be configured to connect to a nearby Wi-Fi network.

Procedure

1. Open the SMART Control Panel and go to **Networking**.
2. If **Advanced Network Configuration** is selected, switch to Basic Network Configuration.
3. Set the **Wi-Fi Configuration** to Client Device.
4. Enter the **SSID** (network name) and **Password** of the Wi-Fi network you want the SMART Controller to connect to.
5. Select whether the SMART will use a Dynamic IP (via DHCP) or Static IP.

   If you select **Static IP**, you can also set the following settings:

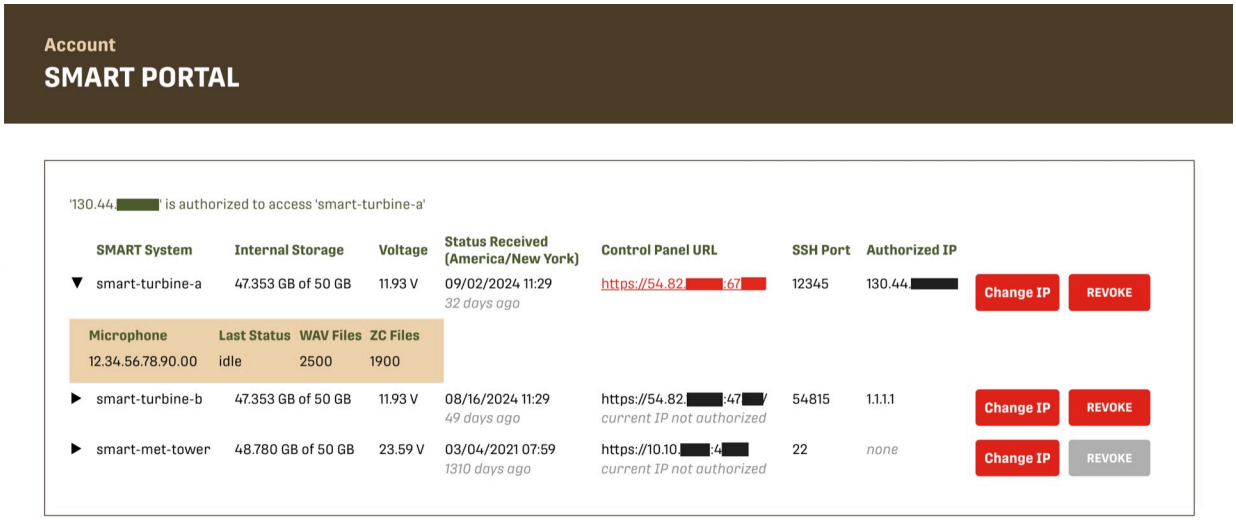
   - **IP Address**
   - **Mask Bits (0-32)**
   - **Default Gateway IP**
   - **Primary Name Server IP**
   - **Secondary Name Server IP**
6. Disable **DHCP Server Configuration**.
7. **Note:** If you accessed the SMART Control Panel via a Wi-Fi connection, applying these changes will cause you to lose connection to the SMART Control Panel.

   Select **Apply Changes**.

SMART IoT Gateway

The SMART IoT Gateway is an optional feature that allows authorized users to securely connect to their SMART Systems via the Internet and configure them using the SMART Control Panel.

When the SMART IoT Gateway is enabled, a SMART System will communicate with Wildlife Acoustics' IoT Gateway server. Authorized users can log into an account at https://www.wildlifeacoustics.com and open the **SMART Portal** to view the status of multiple SMART Systems at once, access each system's Control Panel, or access each system's Ubuntu command line via ssh. This creates a user-friendly and secure system for managing SMART Systems from anywhere in the world.



Related information

Connect to the SMART Control Panel for the First Time *(on page 18)*
Check Available Storage Using the SMART Portal *(on page 74)*

Enable the SMART IoT Gateway Connection

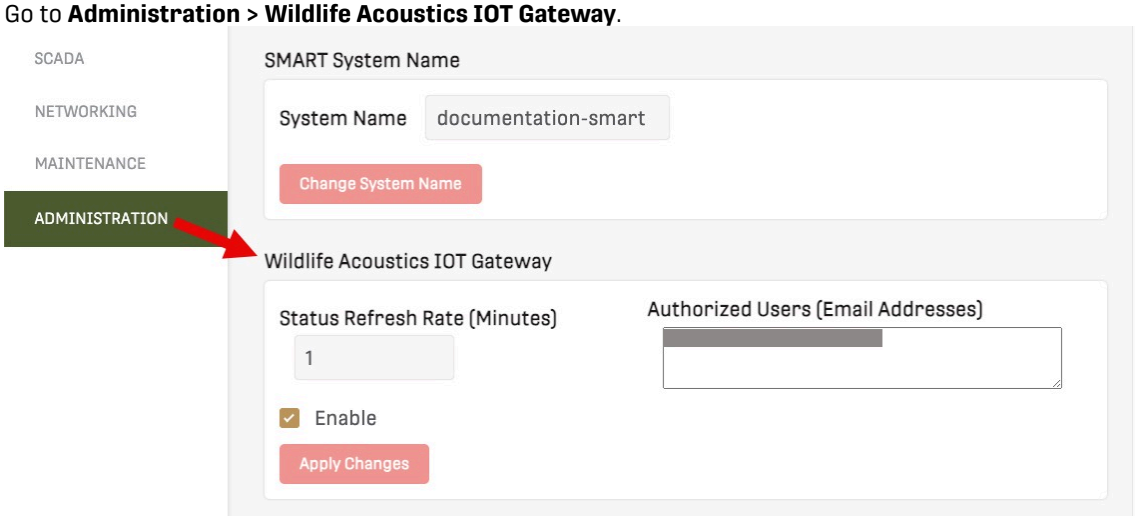
You must access the SMART Control Panel via a local connection and enable the SMART IoT Gateway feature before you can use it.

Before you begin

Your SMART System must be connected to a network with access to the global Internet in order to enable the SMART IoT.

Procedure

1. Connect to the SMART Control Panel via a direct connection or local network.
2. Go to **Administration > Wildlife Acoustics IOT Gateway**.



3. Select the checkbox labeled **Enable** to turn on Gateway functionality.
4. Enter the email addresses of authorized users to grant them access to the SMART System via the SMART Portal on https://www.wildlifeacoustics.com.
   Email addresses must be associated with user accounts on https://www.wildlifeacoustics.com.

Separate multiple email addresses with commas or semicolons.

5. Set the **Status Refresh Rate (Minutes)** to `1`.
   This means you will not have to wait long to see your SMART System in the SMART Portal.
6. Select **Apply Changes**.
7. If your network has a firewall, an IT administrator needs to open port 21577 in the firewall to enable the SMART to communicate with the SMART IoT Gateway.

Access the SMART Control Panel via the SMART Portal

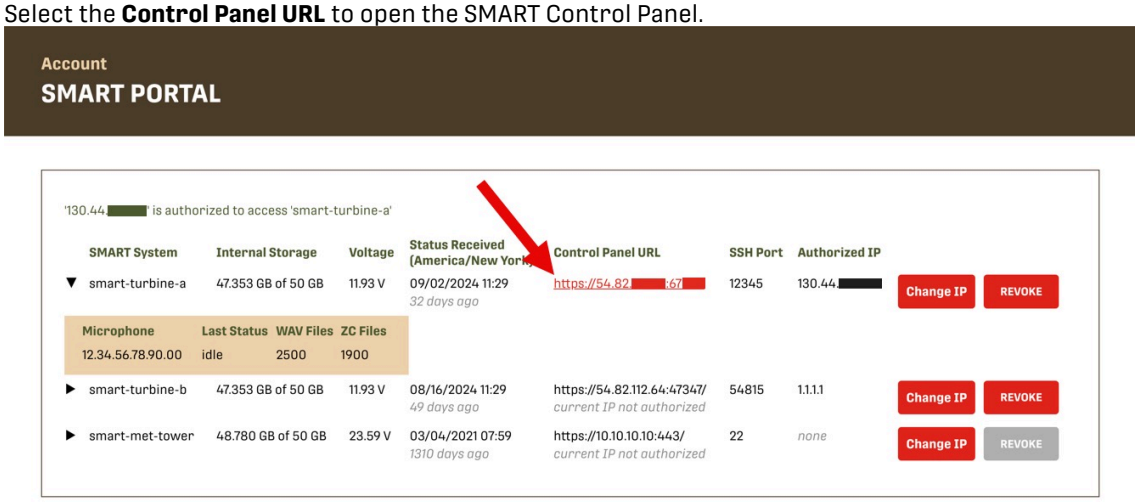
Once you have connected a SMART System to the global Internet and have enabled the SMART IoT Gateway, you can access the SMART Control Panel from any Internet-connected device with a web browser.

Before you begin

- Connect your SMART System to the Internet and enable the SMART IoT Gateway with at least one authorized user.
- Create a user account on https://www.wildlifeacoustics.com using the same email address that is listed as a SMART IoT Gateway authorized user.

Procedure

1. Go to https://www.wildlifeacoustics.com and go to **My Account > SMART Portal**.
2. If your current IP is not authorized, select **Change IP > Authorize My IP Address**.
   Your public IP address will be authorized to access the SMART System via the SMART Portal.
3. Select the **Control Panel URL** to open the SMART Control Panel.



You can navigate directly to this URL from a device with an approved IP address. Logging into an account on https://www.wildlifeacoustics.com is not necessary unless your IP address has changed.

Troubleshooting

Your browser may display a warning that your connection is not private. This is normal when connecting to a device like the SMART Controller with its default configuration. Select your browser's option to proceed anyway.

To avoid this warning in the future, ask your IT department to provide a signed SSL certificate and install it on the SMART Controller.

4. Sign in as the user `smart` using the SMART System password.

Related information

Privacy or Security Warning When Using SMART Control Panel *(on page 125)*

Access the SMART Command Line via the SMART IoT Gateway

Once a SMART System is connected to the SMART IoT Gateway, you can use `ssh` to interact with the SMART command line from any Internet-connected computer.

Before you begin

- Connect your SMART System to a network with Internet access.
- If your network has a firewall, an IT administrator must configure the firewall to open port 21577.
- Enable the **Wildlife Acoustics IOT Gateway** and authorize your Wildlife Acoustics account as a user.

Procedure

1. In a web browser, go to https://www.wildlifeacoustics.com, go to **My Account > SMART Portal**, and find the entry for the SMART System you want to access.
2. If your current IP address is not authorized to access the SMART System, select **Change IP > Authorize My IP Address**
3. Under the **Control Panel URL** column, note the IP address of the SMART, which is the middle portion of the URL.
   Example
   If a SMART System's **Control Panel URL** is `https://12.34.56.78:11223`, the IP address is `12.34.56.78`.
4. Note the value of the SMART System's **SSH Port** number.
5. Open an ssh client and connect to the SMART System's IP address using the SSH port number.

Platform	Instructions
Windows	a. Install and open an SSH client, such as PuTTY. b. Enter the SMART IP address you noted from the SMART Portal as the **Host Name (or IP address)**. c. Enter the SMART's SSH port number as the **Port**. d. Select **Open**.
MacOS or Linux	a. Open your system's Terminal or Command Prompt. b. Enter the command `ssh -p ssh_port smart@smart_ip_address`, where *ssh_port* and *smart_ip_address* are the values you noted from the SMART Portal.

Related information
Unable to Use SSH via SMART IoT Gateway *(on page 128)*

SMART IoT Gateway Security

The SMART IoT Gateway was developed by Wildlife Acoustics as a way to manage multiple Internet-connected SMART Systems. It uses robust security practices and provides simple options for managing user access to the SMART Control Panel.

The SMART IoT Gateway was developed by Wildlife Acoustics using OpenSSL. The server is hosted in Amazon Web Services (AWS), and the client runs on the SMART Controller when enabled in the SMART Control Panel **Administration** tab.

Connection Security

To connect, the SMART Controller establishes an AES128-SHA encrypted connection to the SMART IoT Gateway on a TCP connection. The SMART Controller and IoT Gateway identify and authenticate each other with a SHA256 hash.

The encrypted TCP connection can then multiplex multiple HTTPS and SSH encrypted tunnels between a SMART Controller and authorized clients. This means each session is double-encrypted.

The SMART IoT Gateway maps two unique TCP ports to each connected SMART Controller: one for HTTPS traffic and one for SSH traffic.

User and IP Address Authorization

The SMART IoT Gateway will only allow connections from authorized IP addresses for a given SMART Controller as specified in the **SMART Portal** on wildlifeacoustics.com.

Authorized users can authorize connections from their own current IP address or a manually entered IP address or subnet. The union of all authorized IP addresses from authorized users is allowed to make connections through the SMART IoT Gateway to any given SMART Controller.

Configuring a Firewall for the SMART IoT Gateway

In order to use some of the SMART System's connectivity features, an IT administrator may need to configure your firewall to allow connections between the SMART System and the Wildlife Acoustics SMART IoT Gateway.

Two optional features of the SMART System require the SMART Controller to communicate with Wildlife Acoustics servers:

- The SMART Portal on https://www.wildlifeacoustics.com allows authorized users to access the web-based SMART Control Panel or ssh into the Ubuntu command line via the Internet.
- Email features allow the SMART Controller to send daily status summaries or error alerts to authorized recipients.

To allow these features on a network with a firewall, two ports must be opened:

Port Number	Function
21577	Allows users to access the SMART Control Panel via their account on https://www.wildlifeacoustics.com. Allows users to connect to the SMART's Ubuntu command line via ssh from an Internet-connected device.
21578	Allows the SMART System to send daily status updates and alert messages via email.

Install Custom SSL Certificates

When you access the SMART Control Panel, your web browser may display a warning that the connection is unsafe because the SMART uses self-signed certificates which are not verified by a Certificate Authority. If you install your organization's own SSL certificates on the SMART, you will no longer see these warnings.

Procedure

1. Acquire a valid SSL certificate, private key, and chain file.
   In most cases, your organization's IT department can supply these files.
2. Place the files into a directory on the SMART Controller.
   Example
   This could be the standard `/etc/ssl/certs` directory or `/home/smart`.
3. Edit the file `/etc/apache2/sites-enabled/default-ssl.conf` to replace the default certificate and key files with your own.
   a. Comment out the following lines, which point to the default certificate and key files.

   ```
   #SSLCertificateFile     /etc/ssl/certs/ssl-cert-snakeoil.pem
   #SSLCertificateKeyFile  /etc/ssl/private/ssl-cert-snakeoil.key
   ```

   b. Insert the following line with the path of your `.pem` file.

   ```
   SSLCertificateFile      /filepath/my-pem-file.pem
   ```

   c. Insert the following line with the path of your chain file.

   ```
   SSLCertificateChainFile /filepath/my-chain-file.crt
   ```

   d. Save the file `/etc/apache2/sites-enabled/default-ssl.conf`.
4. Restart the apache web server.

   ```
   systemctl restart apache2.service
   ```

   Depending on the SSL certificate, accessing the SMART using its IP address may still produce security warnings. This is because the certificate may be expecting an explicit domain name (for example, `mysmart.wildlifeacoustics.com`).
5. Assign a fully qualified domain name to your SMART system using one of two methods.
   - Insert your SMART System's IP address and fully qualified domain name into your local hosts file.

     Example
     - Mac and Linux systems: `/etc/hosts`
     - Windows systems: `C:\Windows\system32\drivers\etc\hosts`

     ```
     10.1.1.1 mysmart.mydomain.com
     ```

   - Configure your DNS service to assign a fully qualified domain name to the SMART.

Results

When you access the SMART Console using its IP address or your custom domain name, the security warnings should no longer appear.

Related information

Privacy or Security Warning When Using SMART Control Panel *(on page 125)*

Restore the Default Network Configuration

By default, the SMART Controller is configured to host a local Wi-Fi network. You can restore this configuration via the SMART's terminal.

About this task

If you apply incorrect network settings in the SMART Control Panel, you may lose your connection to the SMART System. If you can connect a keyboard and monitor to the SMART System, you can restore the default network settings, then reconfigure the SMART via a local Wi-Fi connection.

Procedure

1. Connect a monitor and keyboard to the SMART Controller to access the Ubuntu terminal.
2. Go to the directory `/usr/local/share/smart/config/OnLogic-Karbon-300`.

   ```
   cd /usr/local/share/smart/config/OnLogic-Karbon-300
   ```

3. As a superuser, execute the configuration script in this directory.

   ```
   sudo ./config.sh
   ```

4. When the script is complete, restart the Apache server.

   ```
   systemctl restart apache2
   ```

5. Using a Wi-Fi-enabled device with a web browser, join the network created by the SMART Controller.
   The default network name (SSID) is `smart-MAC_Address`, where *MAC_Address* is the MAC address of the SMART Controller without periods.
   The default network password is `wildlife`.
6. Using a web browser, go to `https://192.168.19.1`, and proceed through any warnings about connection privacy.

Results

The default settings will be restored, and you should now be connected to the SMART Control Panel over Wi-Fi.

Edit the Netplan Configuration via the SMART Control Panel

To make changes to the SMART Controller's network configuration beyond the basic options, you can edit the configuration using Netplan YAML syntax directly from the SMART Control Panel.

About this task

CAUTION: Editing advanced network configuration settings should only be done by IT administrators. Making a mistake when editing these settings could completely disable the SMART System's network access.

Procedure

1. Open the SMART Control Panel and go to **Networking**.
2. Switch from Basic Network Configuration to Advanced Network Configuration.
3. Edit the configuration in the **netplan.yaml** window.
4. Select **Apply Changes**.

Results

The SMART System will overwrite the file `/etc/netplan/01-netcfg.yaml` with the contents of the **netplan.yaml** window and restart networking. You will lose connection to the SMART Control Panel and will need to reconnect.

Edit DHCP Server Settings

If you want the SMART Controller to act as a DHCP server, you can configure the relevant settings from the SMART Control Panel, either by adjusting settings in the web-based GUI or by directly editing the `dhcpd.conf` file.

Procedure

1. Open the SMART Control Panel and go to **Networking**.
2. Use either Basic Network Configuration or Advanced Network Configuration to edit the DHCP Server settings.

Configuration Mode	Steps
Basic Network Configuration	a. Enable **DHCP Server Configuration**. b. Configure the relevant settings.
Advanced Network Configuration	a. Enter configuration options into the **dhcpd.conf** window.

3. Select **Apply Changes**.

Results

The new configuration will overwrite the file `/etc/dhcp/dhcpd.conf`, and the SMART Controller will restart networking. You will lose connection to the SMART Control Panel and will need to reconnect.

Update the SMART System Software, Microphone Firmware, and Operating System

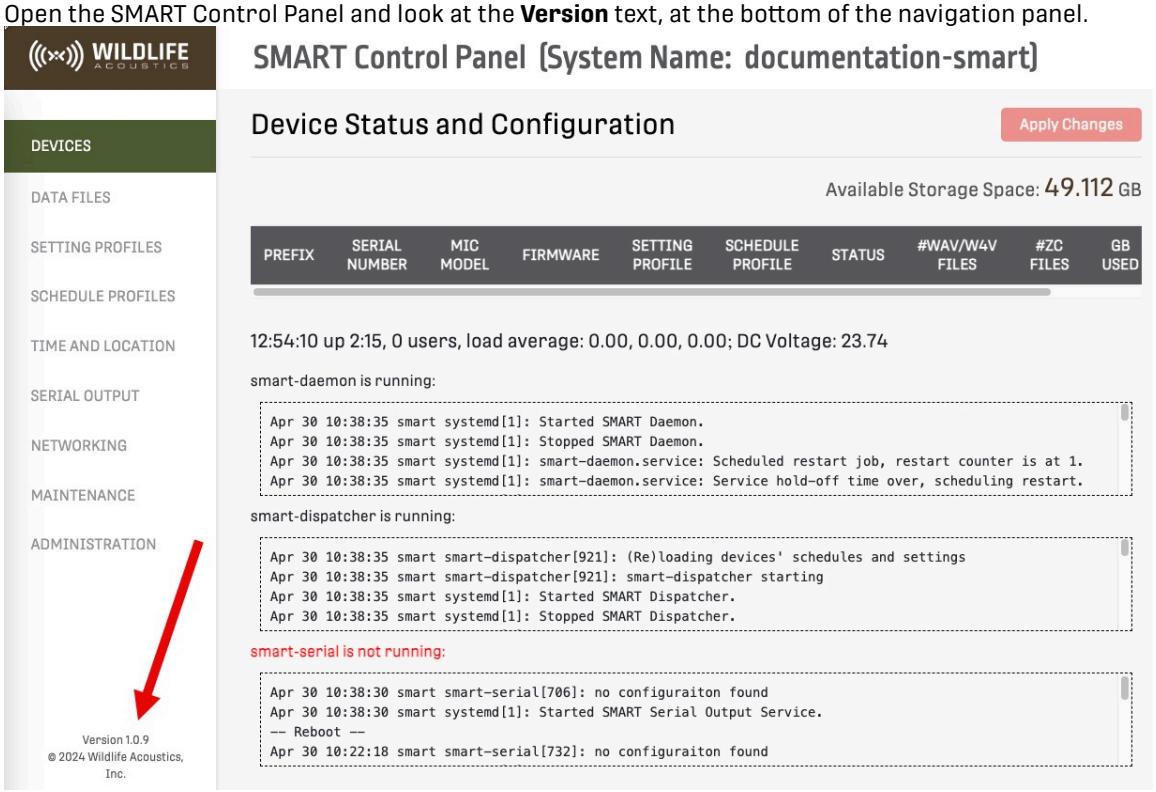
The SMART System software and microphone firmware can be updated remotely using the SMART Control Panel. Operating system updates are required less frequently and are best done with physical access to the SMART Controller.

Check the Installed Ubuntu Version

SMART Systems have shipped with either Ubuntu 18.04LTS or 22.04LTS pre-installed. The version of Ubuntu that is installed determines which version of the SMART System Software you must use to update your SMART Controller.

Procedure

1. Open the SMART Control Panel and look at the **Version** text, at the bottom of the navigation panel.



2. Based on the listed SMART Software Version number, refer to the following table to determine the installed version of Ubuntu.

SMART Software Version Number	Ubuntu Version
1.1.7 or earlier	18.04LTS
18-*x.y.z*	18.04LTS
22-*x.y.z*	22.04LTS

Update the SMART System Software

Updates to the SMART System software regularly add new features and fix bugs. We recommend updating your SMART Controller to the latest available software on a regular basis.

Before you begin

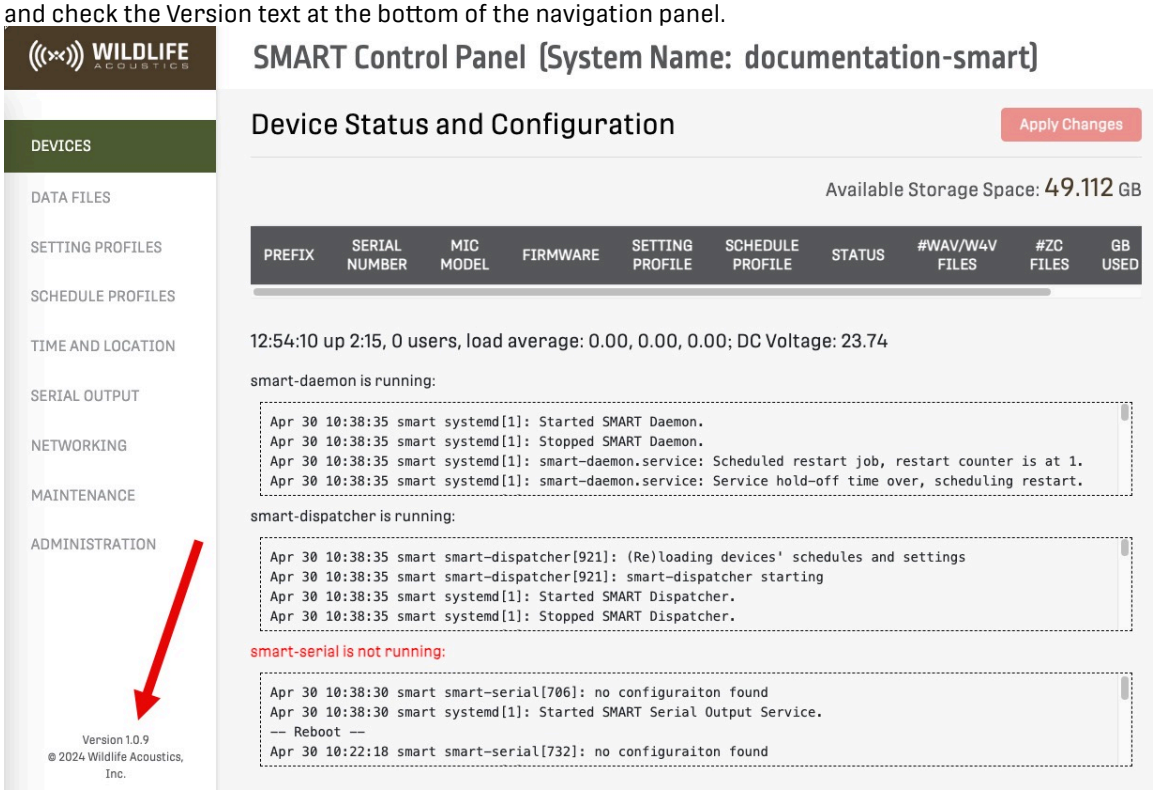
Review the SMART System Software release notes.

About this task

When possible, we recommend installing updates on one SMART System first. Familiarize yourself with any changes before you update a large number of SMART Systems.

Procedure

1. To check the currently installed version of the SMART System Software, open the SMART Control Panel and check the Version text at the bottom of the navigation panel.



2. To download the latest version of the SMART System Software, go to https://www.wildlifeacoustics.com and go to **My Account > Downloads > SMART > SMART System Software**.
3. Next to the entry listing your SMART System's currently installed version of Ubuntu, select **View**.
4. Complete the required Export Certification form and select **Download**.
   The SMART System Software will be downloaded as a ***.****deb file.*
5. Go to the SMART Control Panel and go to **Administration**.
6. Under **Upgrade or Install Package**, drag and drop the SMART System Software file, or click the highlighted area to select the file.
7. Select **Upload and Install**.
   You will lose your connection to the SMART Control Panel as the SMART Controller reboots and completes the installation process. The update process may take up to thirty seconds.
8. Refresh your browser window.

Results

The Version text should now display the new version number of the SMART system software.

Update the SMART MIC-1 Firmware

Each version of the SMART System software includes the most up-to-date version of the firmware for the SMART MIC-1 at the time of the System Software release. After updating the System Software, you must update your SMART MIC-1 microphones from the SMART Control Panel.

Before you begin

Update the SMART System Software to enable the latest microphone firmware update.

Procedure

1. Open the SMART Control Panel and navigate to the **Devices** tab.
   The currently installed firmware version for each microphone is shown in the **Firmware** column.
2. If the **Status** of the microphone is `streaming`, set the **Schedule Profile** to **Disabled**.
   Firmware updates are only possible when the microphone is `idle`.
3. For any microphones that have an update available, select **Upgrade to *X.Y*** under the right-most column of the table.

Update the Smart System to Ubuntu 22

Early SMART Controllers shipped with Ubuntu 18.04 LTS as the installed operating system. These systems can be updated to Ubuntu 22.04 LTS using a bootable USB drive provided by Wildlife Acoustics.

Before you begin

You will need the following equipment:

- SMART Controller
- SMART Ubuntu Upgrade Drive
  Contact Wildlife Acoustics Sales at https://www.wildlifeacoustics.com/contact-us to request a SMART Ubuntu Upgrade Drive.
- USB Keyboard
- Monitor
- Cable to connect the SMART Controller's DisplayPort output to an input on your monitor

This update procedure takes approximately ten minutes.

About this task

Ubuntu 18 reached *End of Standard Support* in April, 2023 and will reach *End of Life* in April 2028. By contrast, Ubuntu 22.04 will reach *End of Standard Support* in April 2027 and will reach *End of Life* in April 2032. See the Ubuntu wiki for a complete list of the ESM and EOL dates for Ubuntu.

Wildlife Acoustics provides a bootable USB Drive that contains both the Operating System (Ubuntu 22) and the latest Ubuntu-22-compatible SMART software release. The upgrade process performs the following tasks:

1. Backups existing SMART configuration and data files
2. Reformats the hard drive
3. Installs Ubuntu 22.04 image
4. Installs `SMART-22-x.y.z.deb`
   x.y.z will differ depending on the latest software version available.
5. Restores the previous SMART configuration and data files

Procedure

1. Remove any USB storage devices from the SMART Controller.
2. Connect your peripherals and check the mount location of the SMART Ubuntu Upgrade Drive.
   a. Use an appropriate cable to connect the SMART Controller's **DP1** DisplayPort output to a monitor.
      Example
      If your monitor accepts HDMI inputs, you can use a DisplayPort-to-HDMI cable.
   b. Plug a USB Keyboard into one of the four USB Ports on the SMART Controller.

c. Run the command `lsblk`.
The output should list several `loop` *entries and* `sda`*, which is the SMART Controller's internal storage drive.*

d. Plug the SMART Ubuntu Upgrade Drive into the SMART Controller and run `lsblk` again.
The output should now list the SMART Ubuntu Upgrade Drive as an additional drive, typically `sdb`.

e. Take note of the drive label assigned to the SMART Ubuntu Upgrade Drive.
You will need to mount this drive at a later step.

3. Reboot the SMART Controller into the **Aptio Setup Utility**.

a. Press the **Power** button to shut down the SMART Controller.

b. Once the SMART Controller is powered off, press and hold the **Escape** key on your keyboard and press the **Power** button again. Continue holding **Escape** until the **Aptio Setup Utility** is displayed.

```
Aptio Setup Utility

Main  Advanced  Chipset  Security  Boot  Save & Exit
-------------------------------------------------------

BIOS Information

BIOS Vendor           American Megatrends
Core Version          5. 12
Compliancy            UEFI 2.5; PI: 1.4
BIOS Version          D8000A10.
Build Date            11/17/2020
Access Level          Administrator

Platform Firmware Information

BXT SOC               F1
MRC Version           0.56
PUNIT FW              1A
PMC FH                03.20
TXE FW                3. 1. 70.2325
ISH FW                4.1.0.3364
GOP                   10.0.1036
CPU Flavor            BXT Notebook/Desktop
Board ID              Oxbow Hill CRB (06)
Fab ID                FAB A

Memory Information

Total Memory          4096 MB
Memory Speed          1600 MHZ
```

4. In the **Aptio Setup Utility** interface, set the first-priority boot drive to the SMART Ubuntu Upgrade Drive and reboot.

a. Use the arrow keys to navigate to the **Boot** panel, select **Boot Option #1**, and press **Return/Enter**.

```
Aptio Setup Utility

Main  Advanced  Chipset  Security  Boot  Save & Exit
-------------------------------------------------------
```

```
Boot Configuration
Setup Prompt Timeout      3
Bootup NumLock State      [On]
Fast Boot                 [Enable]

FIXED BOOT ORDER Priorities
Boot Option #1           [Hard Disk:Windows B...]
Boot Option #2            [SD]
Boot Option #3            [USB Hard Disk]
Boot Option #4            [USB CD/DVD]
Boot Option #5            [USB Key]
Boot Option #6            [USB Floppy]
Boot Option #7            [Network]

UEFI Hard Disk Drive BBS Priorities
```

b. Select the option **USB CD/DVD: UEFI: SanDisk** and press **Return/Enter**.

```
...
FIXED BOOT ORDER Priorities
Boot Option #1           [USB CD/DVD: UEFI: SanDisk]
...
```

c. Use the arrow keys and **Return/Enter** key to select **Save & Exit** > **Save Changes and Exit** > **Yes**.

```
Aptio Setup Utility

Main  Advanced  Chipset  Security  Boot  Save & Exit
-------------------------------------------------------

Save Options

Save Changes and Exit
Discard Changes and Exit

...
```

```
Save & Exit Setup

Save configuration and exit?

Yes  No
```

*The system will reboot and display a new **GNU GRUB** menu.*

5. Use the **Down-Arrow** key to select **Boot SystemRescue and copy system to RAM (copytoram)** and press **Return/Enter** to reboot the SMART Controller.

```
GNU GRUB version 2:2.06.499.ge67a551a4-1

Boot SystemRescue using default options
Boot SystemRescue and copy system to RAM (copytoram)
Boot SystemRescue and verify integrity of the medium (checksum)
Boot SystemRescue using basic display drivers (nomodeset)
Boot SystemRescue with serial console (ttyso, 115200n8)
Boot SystemRescue, do not activate md raid or lvm (nomdivm)
Boot a Linux operating system installed on the disk (findroot)
Stop during the boot process before mounting the root filesystem
Memtest86+ memory tester for UEFI
```

```
Start EFI Shell
EFI Firmware setup
Reboot
Power off
```

Note: The reboot process will take several minutes because the SMART Controller is reading data from the SMART Ubuntu Upgrade Drive.

After several minutes, the system will open a command prompt.

```
========= SystemRescue 18.81 (x86_64) ======== tty1/6 =========
https://www.system-rescue.org/

Console environment:
Run setkmap to choose the keyboard layout (also accessible with the arrow up key)
Run manual to read the documentation of SystemRescue

Graphical environment:
Type startx to run the graphical environment
X.Org comes with the XFCE environment and several graphical tools:
- Partition manager: .. gparted
- Web browser: ........ firefox
- Text editor: ........ featherpad

sysrescue login: root (automatic login)

[root@sysrescue ~]#
```

6. Run the SMART installation script.
   a. Mount the **third partition** of the SMART Ubuntu Upgrade Drive to `/home`.
      Example
      If the SMART Ubuntu Upgrade Drive was mounted to `sdb` in Step 2 *(on page 59)*, you should enter the command:
      ```
      mount /dev/sdb3 /home
      ```
   b. Run the update script by entering the command `/home/update`.
   c. Follow the prompts to install both Ubuntu and the SMART Software.
      ```
      Found SMART installation. Back up data [y|n]?
      y
      Warning: About to erase internal disk. Okay to proceed [y|n]?
      y
      ```
   d. When the installation is complete, enter the command `umount /home`.
7. Remove the SMART Ubuntu Upgrade Drive.
8. Enter the command `reboot`.
   The system will reboot. You will be presented with the system login prompt.

Results

The login banner should indicate that Ubuntu 22.04 is the OS version.

Configure Email Alerts

A SMART System that is connected to the IoT Gateway can be configured to send immediate email alerts in response to several different conditions.

Before you begin

- The SMART IoT Gateway must be enabled in the SMART Control Panel **Administration** tab.
- If the SMART is on a network with a firewall, an IT administrator must open port 21578 to allow the SMART System to communicate with the email server.

About this task

The SMART System can send email alerts to any number of email addresses in response to the following conditions:

- The SMART Controller has rebooted.
- The available space on the SMART Controller's internal drive has fallen below a specified threshold.
- A SMART MIC-1 microphone's status registers as "missing."
- A microphone's measured sensitivity falls below a specified threshold, in dBFS.

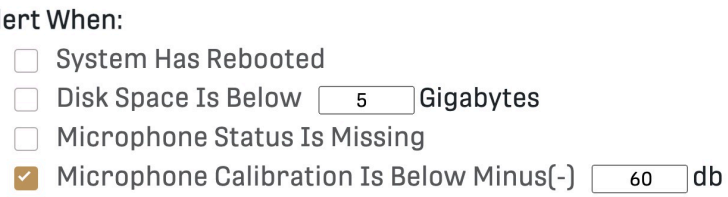
With the exception of a Controller reboot, any of these statuses can trigger repeated alerts if the status does not clear after a specified interval.

Procedure

1. Connect to the SMART Control Panel and go to **Maintenance**.
2. Under **Email Alerts**, specify email addresses to receive the alerts.
   Multiple email addresses should be separated by commas or semicolons.
3. Select which of the available statuses should trigger an alert email, and edit the corresponding thresholds for available disk space and microphone calibration.
   All microphone sensitivity tests yield measurements between 0 dBFS and approximately -96 dBFS. The value you enter is taken as a negative value.

   Example

   Typing `60` into the text box sets the sensitivity threshold to **-60 dBFS**.

4. Select how often alert emails should repeat if the status has not cleared.
5. Select **Send Test Alert Email**.
   The email addresses you listed should receive a test message from `noreply@wildlifeacoustics.com`.
6. Select **Apply Changes**, located at the top of the **Maintenance** tab.

Related information

Enable the SMART IoT Gateway Connection *(on page 50)*

5. Data Management

The SMART System stores recordings and logs on its internal drive initially, and it can periodically transfer new data to a USB or networked storage drive.

Internal Drive Structure

Data recorded by the SMART System, including recordings, results and log files, and device settings, are saved on the SMART Controller's internal drive under `/var/www/html/storage`. Knowing where different types of files are stored can help when you need to manage files via the command line.

Notice: Be very careful when manipulating files via the command line. Moving or editing necessary files could mean you need to reinstall the SMART System software.

`/var/www/html/`

Other than the `storage` directory, this directory stores the assets used by the SMART Control Panel web interface.

`/var/www/html/storage`

Stores settings configured using the SMART Control Panel.

`/var/www/html/storage/devices`

Contains a subfolder for each microphone connected to the SMART System, listed by the microphone's MAC address.

`/var/www/html/storage/devices/`*`MAC address`*

The folder for each attached microphone stores its status information and active settings.

`/var/www/html/storage/devices/`*`MAC address`*`/data`

Stores csv results tables, log files, and recordings saved from a given microphone.

`/var/www/html/storage/dump`

When you generate a diagnostic dump file from the **Maintenance** tab, settings and log files are copied to this folder before being compressed into the file `/var/www/html/storage/logs/dump.zip`.

`/var/www/html/storage/logs`

Stores log text files relating to the SMART Controller, including daily reports, cron logs, and SCADA logs.

When you generate a diagnostic dump file from the SMART Control Panel, it is stored here as `dump.zip`.

`/var/www/html/storage/profiles`

Stores Schedule Profiles and Setting Profiles.

File Name Conventions

Audio recordings, results tables, and log files follow a set of conventions designed to make it easy to identify the source of each file.

Recordings: *prefix_date_time.extension*

prefix

Indicates which microphone recorded each file. By default, a microphone's prefix is the serial number or MAC address, but this can be changed to a custom name in the **Devices** tab.

date* and *time

Record when each audio file began recording.

date is formatted as *YYYYMMDD* and *time* is formatted to the nearest microsecond as *hhmmss_uuuuuu*.

extension

Uncompressed full-spectrum files have the extension `.wav`. WAV files are widely supported by many audio analysis programs.

Compressed full-spectrum files have the extension `.w4v`. W4V files are compressed using an open-source compression algorithm developed by Wildlife Acoustics. They can be converted to WAV files using the Kaleidoscope Pro Sound Analysis Software.

Zero-crossing recordings have the extension `.zc` and follow the Anabat Call Sequence Type 132 format.

Results Tables: *date_time_prefix*_results.csv

prefix

Indicates the microphone that recorded the audio analyzed in the results table. By default, a microphone's prefix is the serial number or MAC address, but this can be changed to a custom name in the **Devices** tab.

date* and *time

Record the date and time of the first entry in the results table.

date is formatted as *YYYYMMDD* and *time* is formatted to the nearest microsecond as *hhmmss_uuuuuu*.

Microphone Log File: *date_time_prefix*_log.txt

prefix

Indicates the microphone whose status is recorded in the log file. By default, a microphone's prefix is the serial number or MAC address, but this can be changed to a custom name in the **Devices** tab.

date* and *time

Record the date and time of the first entry in the log file.

date is formatted as *YYYYMMDD* and *time* is formatted to the nearest microsecond as *hhmmss_uuuuuu*.

Diagnostic Dump File: *systemName_date_time*_dump.zip

systemName

The name assigned to the SMART Controller that generated the dump file.

By default, *systemName* is `smart`. This can be changed in the **Administration** tab.

date* and *time

Indicate when the dump file was generated.

date is formatted as *YYYYMMDD* and *time* is formatted to the nearest microsecond as *hhmmss_uuuuuu*.

SMART Daily Log: *date_time*_smart-daily-log.txt

systemName

The name assigned to the SMART Controller that generated the log.

By default, *systemName* is `smart`. This can be changed in the **Administration** tab.

date* and *time

Indicate when the log was generated.

date is formatted as *YYYYMMDD* and *time* is formatted to the nearest microsecond as *hhmmss_uuuuuu*.

Cron Log: *date_time*-smart-cron-log.txt

systemName

The name assigned to the SMART Controller that generated the cron log.

By default, *systemName* is `smart`. This can be changed in the **Administration** tab.

date* and *time

Indicate the date and time of the first entry in the cron log.

date is formatted as *YYYYMMDD* and *time* is formatted to the nearest microsecond as *hhmmss_uuuuuu*.

Daily Automatic Data Maintenance

The SMART Controller can be configured to perform a set of data management tasks once per day. These tasks include copying data from the internal drive to USB or network storage, deleting old data from the internal drive, and sending an email summary to specified email addresses. The SMART Controller can also be configured to shut down after maintenance until a specified time in order to conserve power.

Daily maintenance can be scheduled to occur at any time, even during recording. However, scheduling it to occur outside of the recording schedule simplifies file management.

Daily maintenance can include the following tasks:

USB Backup

Any new results tables, log files, or recordings on the SMART Controller internal drive can be copied to a single USB-connected storage device.

Network Backup

Any new results tables, log files, or recordings on the SMART Controller internal drive will be copied to a network storage location using **rsync**.

Results Table and Log Rotation

Bat analysis results tables and status logs can be broken up into smaller files based on a time interval specified in days.

Results Table and Log Clean-up

To limit the file size of bat analysis results tables and status logs, the SMART Controller will only store entries from a specified length of time on its internal storage. Files edited prior to the specified time window are deleted automatically.

Audio Recording Deletion

To preserve space on the SMART Controller's internal drive, audio recordings can be automatically deleted a specified number of days after they were recorded.

Daily Email Report

A SMART System connected to the SMART Portal can send a daily report to any number of specified email addresses. This daily report includes key status information about the SMART Controller and any attached microphones.

Custom Shell Script

If you save a shell script to the path `/var/www/html/storage/daily-hook.sh`, the SMART System will execute that script at the end of daily maintenance each day.

Power-Down Period

For deployments where power availability is limited, the SMART System can be configured to power down after daily maintenance is done until a specified time of day.

Schedule Daily Automatic Maintenance and Power-Down

Daily maintenance is scheduled by time of day. For deployments with limited available power, the SMART Controller can be configured to power down after maintenance is done until a specified wakeup time.

Before you begin

Make sure the **Time and Location** settings are configured for your SMART Controller. Daily Maintenance scheduling follows your SMART Controller's internal clock and **Time Zone** setting.

Procedure

1. Open the SMART Control Panel and go to **Maintenance**.
2. Select a time for **Start of Daily Maintenance Window** during a gap in your microphones' schedules.

   Example
   If all microphones are scheduled to record from sunset to sunrise, schedule maintenance to start in the morning, after sunrise. **Account for the latest time sunrise will occur**, depending on seasonal changes and Daylight Saving Time time zone changes.
3. Set a time to end the SMART System's post-maintenance shutdown.
   If your schedules follow sunrise or sunset, be sure to account for changes in sunrise and sunset times and changes in time zone.
4. Set the behavior for the optional power-down period that occurs after maintenance.
   - To disable the power-down period, set **After Maintenance, Power Down Until** to the same value as **Start of Daily Maintenance Window**.
     This is recommended for deployments without power constraints. Disabling power-down keeps the SMART available for remote access.
   - To enable the power-down period, set **After Maintenance, Power Down Until** to the time when you want the SMART to power on again.

   If you enable the power-down period, be sure that it ends before any of your Microphone schedules begin, accounting for seasonal changes in sunrise, sunset, and time zone. The daily power-down period will always run for the fullest possible duration, even if that overlaps with any microphone's recording schedule.
5. Select **Apply Changes**.

Example: Daily Maintenance with No Power-Down

If the maintenance process ends after the **Power Down Until** time has already passed, the SMART will not power down at all.

To avoid powering down each day, set the same time for both **Start of Daily Maintenance Window** and **After Maintenance, Power Down Until** fields. Typically, you should schedule maintenance during a gap in your monitoring schedule.

In this example, maintenance occurs at 09:00 with no power-down.

```
Start of Daily Maintenance Window: 09:00
After Maintenance, Power Down Until: 09:00
```

Example: Daily Maintenance and Power Down During Daylight
If all of the microphones in a SMART System are scheduled to monitor from sunset to sunrise, one possible maintenance schedule would be to perform maintenance in the morning, following sunrise, then sleep until shortly before sunset.

This allows the SMART System to conserve power during the day, when not monitoring. Be sure to leave enough time after the unit powers on to complete any heating or sensitivity testing behaviors included in your settings. If your microphone's schedule follows sunrise or sunset, be sure to time your maintenance and power-on behavior to account for changes in the length of the day throughout your deployment.

```
Start of Daily Maintenance Window: 09:00
After Maintenance, Power Down Until: 17:00
```

USB Backup

During daily maintenance, the SMART Controller can copy new data from its internal drive to a USB storage device. This can include bat analysis results files, log files, and both full-spectrum and zero-crossing recordings.

Connect and Verify a USB Backup Drive

Before you set up automatic USB backup, you should manually mount the drive and ensure that it works correctly.

About this task

We recommend using a Solid State Drive (SSD) as your backup drive instead of a Hard Disk Drive (HDD). SSDs have no moving parts and are therefore more resilient than HDDs to changes in climate.

The SMART System's daily maintenance function only supports a single USB drive at a time.

Procedure

1. Connect the SSD to the SMART Controller via any available USB port.
2. Access the SMART System's command prompt either using a keyboard and display or SSH.
3. Get the device name for the USB drive.
   `ls /dev/disk/by-path/pci*usb*part1`
4. Make sure the output lists a usb device.
   Example
   `/dev/disk/by-path/pci-0000:00:15.0-usb-0:6:1.0-scsi-0:0:0:0-part1`
5. Mount the USB drive to `/mnt/smart-external`.
   `sudo mount -ouid=www-data,gid=www-data `ls /dev/disk/by-path/pci*usb*part1` /mnt/smart-external`
   There should be no error messages.
6. List the contents of the drive.
   `ls /mnt/smart-external`
7. Verify the listing is as expected.

If this is a new drive, there should be no contents, or at most a `lost+found` directory.

8. Unmount the drive.
   `sudo umount /mnt/smart-external`

What to do next
Once you have confirmed that your drive can be mounted and its contents are as expected, you can configure daily backup to the drive using the SMART Control Panel.

Configure Automatic Backup to USB Storage

Once you have connected a USB backup drive and verified that it works, you can enable daily automatic backup using the SMART Control Panel.

Before you begin
Mount your drive manually and verify its contents using the Ubuntu command prompt.

About this task
If USB backup is enabled, the SMART Controller will mount the first USB storage device it detects. For this reason, only one USB storage device, maximum, should be connected to the SMART Controller for daily backup.

Procedure
1. Attach one USB storage device to one of SMART Controller's USB ports.
2. Open the SMART Control Panel and go to **Maintenance**.
3. Under **Copy Data to External USB Drive, If Available**, select the types of files you wish to back up.

   CSV Files

   Bat analysis results files.

   Log Files

   Text files that record status updates from the SMART Controller and each microphone.

   WAV/W4V Files

   Full-spectrum audio recordings. The Microphone Setting Profile must have **Output WAV** enabled to save WAV or W4V files to internal storage.

   ZC Files

   Zero-crossing audio recordings. The Microphone Setting Profile must have **Output ZC** enabled to save ZC files to internal storage.
4. Choose whether to keep the USB drive mounted or not after backup is complete.
   We recommend disabling **Keep Drive Mounted** to avoid corrupting the backup drive in the case of unexpected power loss. In this configuration, the SMART Controller will mount the drive at the start of its daily maintenance, then unmount it when it finishes USB backup.
5. Select **Apply Changes**.

USB Backup Drive Structure

If the SMART System is configured to back up data to a USB drive during daily maintenance, the drive is mounted to `/mnt/smart-external`. If backup settings are configured to keep the drive mounted outside of the maintenance period, you can access the external drive via the command line.

`/mnt/smart-external/`

In the root folder, you will find backups of the SMART system's daily reports, cron logs, and SCADA logs. You will also find a backup of the most recent diagnostic `dump.zip` file, generated via the **Maintenance** tab of the SMART Control Panel.

`/mnt/smart-external/`*`MAC Address`*`/data/`

Stores csv results tables, log files, and recordings saved from a given microphone, identified by its *MAC Address*.

Configure Automatic Data Backup to Network Storage

During daily maintenance, the SMART Controller can copy new data from its internal drive to a network storage location using **rsync**. This can include bat analysis results files, log files, and both full-spectrum and zero-crossing recordings.

Before you begin

Ensure that you can connect to your `rsync` server from the SMART Controller and log in using the appropriate credentials.

Procedure

1. Open the SMART Control Panel and go to **Maintenance**.
2. Under **Copy Data to Network via rsync**, select the types of files you wish to back up.

   CSV Files

   Bat analysis results files.

   Log Files

   Text files that record status updates from the SMART Controller and each microphone.

   WAV/W4V Files

   Full-spectrum audio recordings. The Microphone Setting Profile must have **Output WAV** enabled to save WAV or W4V files to internal storage.

   ZC Files

   Zero-crossing audio recordings. The Microphone Setting Profile must have **Output ZC** enabled to save ZC files to internal storage.
3. Enter the destination path for the files and the required password.
   The destination path is generally formatted as *`username@host:destination_path`*.
4. Select **Apply Changes**.
5. Select **Send Test File** to send an example text file named `testfile_`*`YYYYMMDD_hhmmss`*`.txt` to the destination path.

Automatically Delete Old Recordings from Internal Storage

To preserve space on the SMART Controller's internal drive for new data, you can configure the SMART system to delete old WAV/W4V and ZC files after a specified time period. The time period is specific to file type, so you can delete larger WAV/W4V files more frequently than smaller ZC files.

About this task

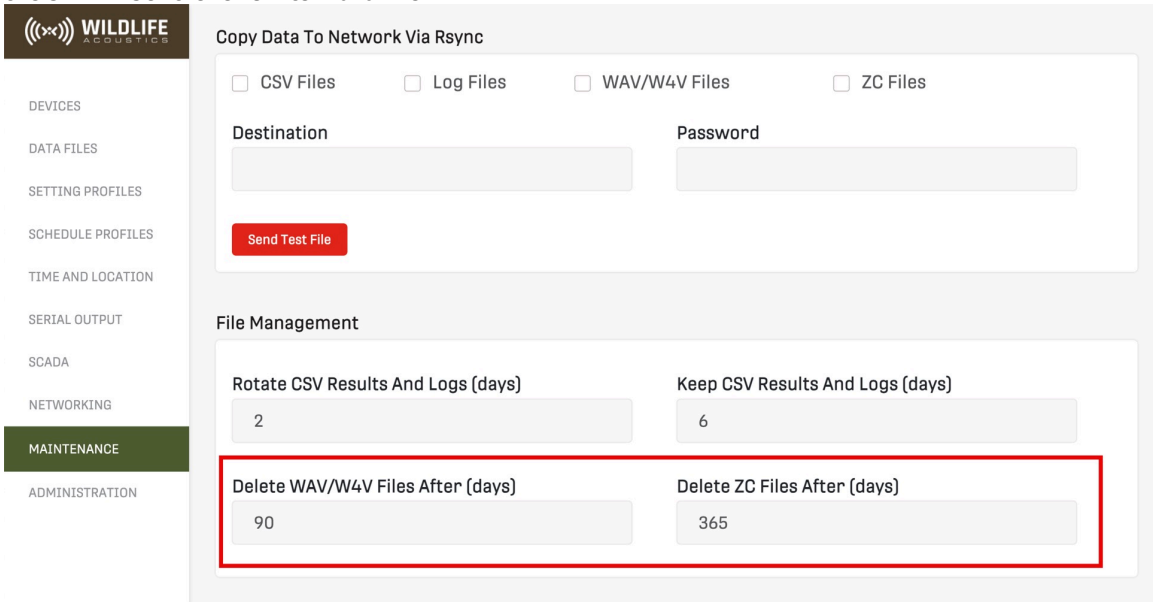
You should treat the SMART System's internal drive as temporary storage. If SMART runs out of space on its internal drive, it will be unable to record new data. You should configure SMART to delete old files often enough that you maintain ample space on the internal drive, and you should configure automatic backup to copy your data to a USB drive or network storage location for long-term storage.

Before you begin

It is highly recommended that you **configure automatic backup** from the SMART Controller's internal drive to USB or network storage and confirm that it is working properly.

Procedure

1. Open the SMART Control Panel and go to **Maintenance**.
2. Under **File Management**, enter the number of days after which each file type should be removed from the SMART Controller's internal drive.



   The size of zero-crossing (ZC) files depends on the recorded content, but they are approximately one thousand times smaller than equivalent, uncompressed WAV files.

3. Select **Apply Changes**.

Configure Automatic Results Table and Log File Rotation and Deletion

Bat analysis results tables and status logs are *rotated*, or split up into separate files, on a regular basis. To limit storage use, these files are deleted from the internal drive after a specified number of days.

Before you begin

If you intend to store bat analysis results or log files from the SMART System long-term, it is highly recommended that you **configure automatic backup** for these files to either USB or network storage.

About this task

Procedure

1. Open the SMART Control Panel and go to **Maintenance**.
2. Decide how bat analysis results tables and status logs should be split up, and enter a duration, in days, into **Rotate CSV Results and Logs**.
   Each results file and log file will span the specified number of days, beginning and ending at the scheduled daily maintenance window.
3. Decide how long results and log files should be kept on the SMART Controller's internal drive, and enter that value, in days, in **Keep CSV Results and Logs**.
   Results and log files will be kept for the specified duration after the last entry in each file.
4. Select **Apply Changes**.

Example: File Rotation and Deletion Schedule

In this example, parameters are set as follows:

- **Rotate CSV Results and Logs (days)**: 7
- **Keep CSV Results and Logs (days)**: 30

The SMART System is scheduled to record on all microphones from sunset to sunrise, then perform daily maintenance at 8:00 AM. The results and log files will follow this behavior.

1. At the start of night 1, the SMART System creates a new results file, microphone log file, and system log file.
2. Results and logs for nights 1 through 7 are saved to the same files.
3. Results and logs for nights 8 through 14 are saved to a different set of files, and so on for each 7-day period.
4. During daily maintenance following night 37, the results and log files for nights 1-7 are removed from the internal drive.

Configure Daily Email Reports

A SMART System connected to the SMART Portal can send daily reports to one or more email addresses during its daily maintenance. Each report summarizes the basic status of the SMART Controller and its microphones, including the sensitivity of each microphone and how much data has been collected.

Before you begin

Your SMART System must be connected to the SMART Portal to send daily email reports.

Procedure

1. Open the SMART Control Panel and go to **Maintenance**.
2. Under **Daily Email Report**, enter the email addresses of the recipients for the daily reports and provide a **Subject** for the email.
   Separate multiple email addresses using commas.
3. Select **Apply Changes**.
4. Select **Send Test Email** to confirm that all the entered email addresses received a test message.

Related information

Enable the SMART IoT Gateway Connection *(on page 50)*

Daily Custom Script Execution

For daily maintenance tasks not possible with the options built into the SMART Control Panel, you can write a custom shell script that will run each day as part of the maintenance period.

Your custom script should be saved to the path `/var/www/html/storage/daily-hook.sh`.

If this file exists, SMART will execute the script as the last task in it daily maintenance routine. If a daily power-down is enabled, the custom script will be the last task prior to powering down.

One possible use for such a script would be to back up files to a cloud storage service that does not support `rsync`.

Manually Download Data from the SMART Control Panel

You can manually retrieve recordings, bat analysis results tables, and log files using the SMART Control Panel. You can filter these files by the microphone associated with them and by a date range to download individual files or batches of files.

About this task

The SMART Control Panel lets you access files stored on the SMART Controller's internal drive at `/var/www/html/storage/devices` without using the command line.

Procedure

1. Open the SMART Control Panel and go to **Data Files**.
2. Select the types of data you wish to retrieve.

   CSV Files

   Bat analysis results files.

   Log Files

   Text files that record status updates from the SMART Controller and each microphone.

   WAV/W4V Files

   Full-spectrum audio recordings. The Microphone Setting Profile must have **Output WAV** enabled to save WAV or W4V files to internal storage.

   ZC Files

   Zero-crossing audio recordings. The Microphone Setting Profile must have **Output ZC** enabled to save ZC files to internal storage.
3. Select specific devices (microphones) from which you want to retrieve data, or leave the selection blank to include all devices.
   If a SMART System includes multiple microphones, the recordings, results tables, and log files are stored separately for each microphone.
4. Select the date range of the data you want to download.
5. Select whether to split up dates at midnight or at noon.
   By default, the date range will select data from 12:00 AM at the start of the first date to 11:59 PM on the last date. **Nighttime** mode selects data from 12:00 PM on the day before the first date to 12:00 PM on the last date.

   Example
   If the selected **Date Range** is **2024-06-02 to 2024-06-04** and the **Time Mode** is **Nighttime**, the selected data will span from **2024-06-01 at 12:00:00 PM** to **2024-06-04 at 12:00:00 PM**.
6. Select **Apply Changes**.
   A list of the selected files will populate at the bottom of the page.
7. Select individual files to download them, or select **Download All to Zip** to download a `.zip` archive containing all the listed files.

Generate Diagnostic Dump Files

If you encounter an issue with your SMART System and contact Wildlife Acoustics Support, they may ask you to generate a file containing copies of your SMART system's configuration and recent log files. This can help Wildlife Acoustics Support recreate and diagnose the issue.

Procedure

1. Connect to the SMART Control Panel and go to **Maintenance**.
2. Select **Create Diagnostic Dump File**.

Results

If you are running version 1.3.4 or later of the SMART System software, a file named *`systemName_date_time`*`_dump.zip` will be downloaded through your browser. Otherwise, you can retrieve the file from the SMART filesystem at `/var/www/html/storage/logs/dump.zip`.

Check Available Internal Storage

You can check the available space on the SMART Controller's internal drive via the SMART Portal or the SMART Control Panel.

Check Available Storage Using the SMART Control Panel

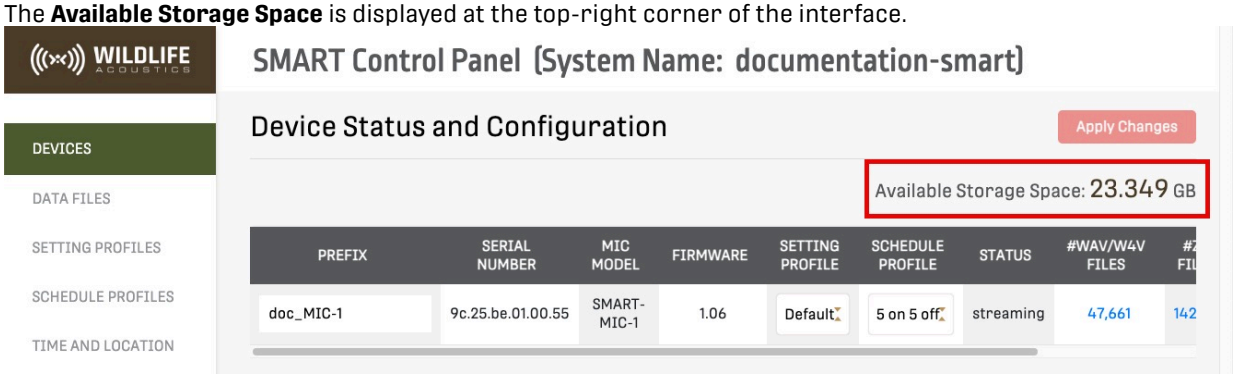
The available storage space on the SMART Controller's internal drive can be checked on the **Devices** tab of the SMART Control Panel.

Procedure

Open the SMART Control Panel and go to the **Devices** tab.

Results

The **Available Storage Space** is displayed at the top-right corner of the interface.



Check Available Storage Using the SMART Portal

If you have enabled the SMART Portal, you can check the available storage on all of your SMART Systems from an authorized account on https://www.wildlifeacoustics.com.

Before you begin

You must have enabled the SMART Portal and authorized your Wildlife Acoustics user account to access at least one SMART System.

Procedure

Go to https://www.wildlifeacoustics.com and go to **My Account > SMART Portal**.

Results

The **Internal Storage** usage will be displayed for each of the SMART Systems you are authorized to use.

Account
SMART PORTAL

'130.44.[redacted]' is authorized to access 'smart-turbine-a'

	SMART System	Internal Storage	Voltage	Status Received (America/New York)	Control Panel URL	SSH Port	Authorized IP		
▼	smart-turbine-a	47.353 GB of 50 GB	11.93 V	09/02/2024 11:29 *32 days ago*	https://54.82.[redacted]:67[redacted]	12345	130.44.[redacted]	Change IP	REVOKE
	Microphone	**Last Status**	**WAV Files**	**ZC Files**					
	12.34.56.78.90.00	idle	2500	1900					
▶	smart-turbine-b	47.353 GB of 50 GB	11.93 V	08/16/2024 11:29 *49 days ago*	https://54.82.[redacted]:47[redacted]/ *current IP not authorized*	54815	1.1.1.1	Change IP	REVOKE
▶	smart-met-tower	48.780 GB of 50 GB	23.59 V	03/04/2021 07:59 *1310 days ago*	https://10.10.[redacted]:4[redacted] *current IP not authorized*	22	*none*	Change IP	REVOKE

Related information

SMART IoT Gateway *(on page 49)*

6. Installing SMART

Installing the SMART System on a wind turbine, meteorological tower, or similar structure requires careful planning. Proper connectivity, grounding, and microphone placement will help ensure a successful deployment.

Preparing to Install

Once a SMART System is installed on a wind turbine or meteorological mast, it may become much more difficult to troubleshoot some aspects of the system. You should make sure your SMART System is fully configured and you have the appropriate materials ready prior to installation.

Procedure

1. Configure your SMART Controller's network settings to match the network it will be connected to on-site.
    - If you will be using a cellular modem, configure and test it prior to installation.
    - If you will be connecting to a local Wi-Fi network, you can use a different network to test the connection process. You may wish to enter the on-site's network credentials in the SMART Controller's network settings prior to installation.
2. Configure your login credentials on the SMART Controller and familiarize yourself with the process of accessing the SMART Control Panel and Ubuntu command line interface.
3. Procure a USB backup drive, format it for Ubuntu compatibility, and test the daily USB backup functionality.
4. Configure and test daily backup to a network storage location via `rsync`.
5. Prepare the cables and additional networking components you will need to power and connect every part of your SMART System. These may include the following:

    Ethernet cables

    Include enough length to account for any possible movement, such as the rotation of a turbine nacelle.

    Use UV-protected cable for any lengths of cable that will be exposed to the sun.

    Ethernet switches

    If you will be connecting more than two microphones to a single SMART Controller, you will need a *PoE* switch to connect multiple microphones to a single Ethernet port on the SMART Controller.

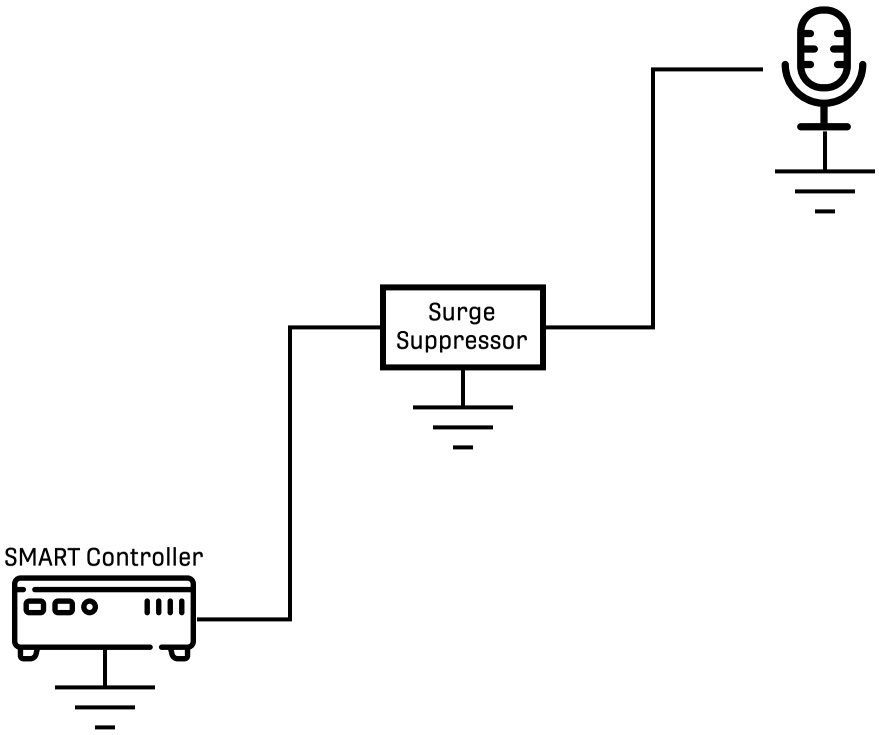
    If the Ethernet switch requires power, ensure you have any necessary power cables or extension cords.

    Ethernet signal repeaters

    A single run of ethernet cable can only extend 100 meters before you risk signal degradation. If you must span a distance greater than 100 meters, you need a means of repeating the signal and extending the total Ethernet range.

Grounding

A SMART System should be electrically grounded at multiple points (the controller, microphones, and in-between) to minimize the potential for damage from electrical discharges.



Warning: Wildlife Acoustics cannot provide instructions on grounding beyond indicating the grounding points available on Wildlife Acoustics equipment. Consult with an electrician or electrical engineer for further details.

The SMART System is designed for use on tall, exposed structures like wind turbines and meteorological towers where lightning and electrical surges are likely. To minimize the possibility of equipment damage and data loss, you should ground the SMART System at multiple points whenever possible.

The SMART System can be grounded at the following points:

Microphones

Typically, the microphone is the component of the SMART System most exposed to electrical discharge. Without proper grounding, electricity that discharges to the microphone may travel throughout the SMART System, damaging the microphone, the SMART Controller, and any networking components in between.

The body of each SMART MIC-1 microphone features a non-anodized strip of aluminum. You should apply conductive grease between the microphone and a conductive mounting clamp, then ground the clamp.

In-Line Surge Suppressors

Each SMART MIC-1 ships with an RJ45 Surge Suppressor. This should be installed in between the microphone and either the SMART Controller or Ethernet switch, if one is used. The Surge Suppressor should be installed as close as possible to the controller or switch, with the bulk of the Ethernet run in between the Surge Suppressor and microphone, to maximize protection. The Surge Suppressor should be grounded using the pre-installed grounding wire.

SMART Controller

Ground the SMART Controller using the exposed grounding nut, adjacent to the power input socket.

Calculating Battery and Solar Panel Needs

If you plan to power a SMART System using a solar panel and external battery, factors that determine how much energy the SMART will use and how much sun exposure is available per day will influence the appropriate equipment choice.

In order to keep a SMART System running indefinitely, a solar panel would need to capture more energy than the SMART System expends on the average day. The battery would also need to be large enough that it can power the SMART System during any periods when the solar panel cannot capture energy.

Listed below are some of the factors that you should take into account when choosing a solar panel and external battery.

Factors Affecting SMART Energy Usage

To estimate how much energy a SMART System expends on an average day, you need the following information:

Number of microphones

Using multiple microphones with a single SMART system will increase how much power the system uses while recording and analyzing audio.

Recording activity per day

How much time per day the SMART spends monitoring and recording will directly affect how much energy it uses, and therefore how much energy must be recuperated. If the SMART's microphone schedules are based on sunset and sunrise, note that seasonal changes may affect how long the SMART records each day.

Heating activity per day

The SMART MIC-1's heating element can almost double power consumption if it is active during recording. How much time the heater spends enabled each day will impact daily energy usage. Note that this can change seasonally if your recording schedule follows sunset and sunrise.

Energy used by accessories

Any accessories attached to the SMART System, such as Ethernet switches or USB drives, may contribute to the total energy usage.

Factors Affecting Solar Energy Capture

To estimate how much energy a solar panel can recapture each day, you need the following information:

Location

The location on the globe determines the average angle of the sun and the duration of daylight during the day. Local weather also plays a major role: a solar panel that sees mostly cloudy weather will not generate energy very effectively.

Time of year

During what times of year do you need to record? If you plan on recording through winter, the solar panel needs to be large enough to capture sufficient energy with limited sunlight.

Shade

If nearby obstacles will block the sun during portions of the day, that must be accounted for in your estimations.

Related information

SMART Controller Specifications *(on page 132)*

Installing Microphones

Microphone Placement on Wind Turbines

Placing your microphones thoughtfully will allow you to detect bat activity near a wind turbine most effectively. Poor microphone placement can hinder your microphones' sensitivity and produce cluttered recordings.

Rotor-Swept Area and Bat Behavior

If you are using SMART to detect bat activity near a wind turbine, you should place microphones to maximize the probability that bats flying in or near the rotor-swept area will be recorded by at least one microphone. Depending on the size of the turbine, this may require one or multiple microphones, and full coverage may not be possible in some situations. If you know the altitudes at which nearby species typically fly, you can adjust your microphone placement to better target those altitudes.

Broadly speaking, common locations for microphones include the following:

- near ground level or low on the turbine mast
- underneath the nacelle
- above the nacelle

Avoiding Obstructions

To the extent possible, microphones should have unobstructed exposure to the open air. Microphones that are more surrounded by multiple surfaces, such as those at the joint between a tower and nacelle, are able to record incoming sound from fewer directions and are likely to capture more prominent echoes.

SMART MIC-1 Mounting Bracket

The SMART MIC-1 Mounting Bracket is an optional accessory that allows multiple microphone orientations.

SMART MIC-1 Mounting Bracket Specifications

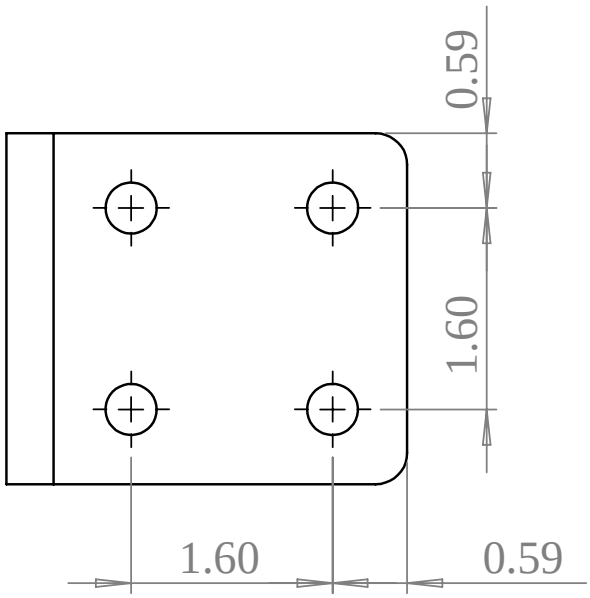
The SMART MIC-1 mounting bracket sold by Wildlife Acoustics is designed to mount the microphone on an external surface or through the wall of a structure.

General Characteristics

- Tested to MIL-STD 810G, Method 514.6, Vibration Category 4, for 30 minutes on each axis.
- Made from fully weatherproof stainless steel.

Mounting Hole Dimensions

Four mounting holes, 0.49 in. internal diameter, 1.60 in. center-to-center spacing.



DIMENSIONS ARE IN INCHES

Orienting the SMART MIC-1 Mounting Bracket

The SMART MIC-1 Mounting Bracket has multiple mounting holes that allow you to angle the microphone in different ways relative to the bracket. The bracket can be used for through-hole mounting or external surface mounting.

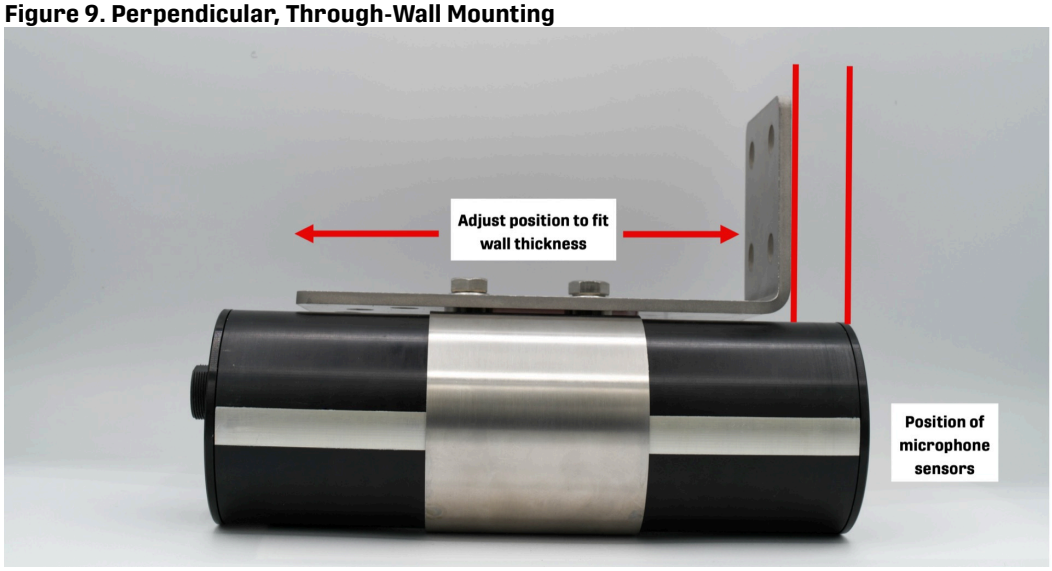
Through-Wall Mounting

Through-wall mounting refers to mounting the microphone to the inside of a structure and pointing it through a hole in the wall so that its face is flush with the struture, facing outwards. When the face of the microphone is flush with the surface of a structure, it eliminates the possibility of recording reflections of bat calls bouncing off of that surface and increases the effective sensitivity of the microphone. Positioning the microphone face recessed or protruding from the flat surface will result in poorer audio quality.

When installing a microphone on a structure with wide, flat surfaces, such as a wind turbine, through-wall mounting is preferred to external mounting because the latter produces recordings with more prominent reflections. These reflections can overlap with the initial calls, reducing clarity both for human analysts and automatic call detection. Through-wall mounting is also preferred when it is easier to access the interior of a structure than the exterior.

Notice: Always consult with structural engineers before cutting new holes in the surface of a structure. Consult with the turbine manufacturer before cutting holes in a turbine body.

Figure 9. Perpendicular, Through-Wall Mounting



When you mount the microphone perpendicular to a flat surface, you can reposition the microphone in the clamp to account for the thickness of the wall.

External Surface Mounting

When through-wall mounting is not possible, the Mounting Bracket can be fastened to an external surface. The microphone can be angled perpendicular to the mounting surface, 30° from perpendicular, or 60° from perpendicular.

Figure 10. Perpendicular Surface Mounting

Figure 11. 30° Surface Mounting

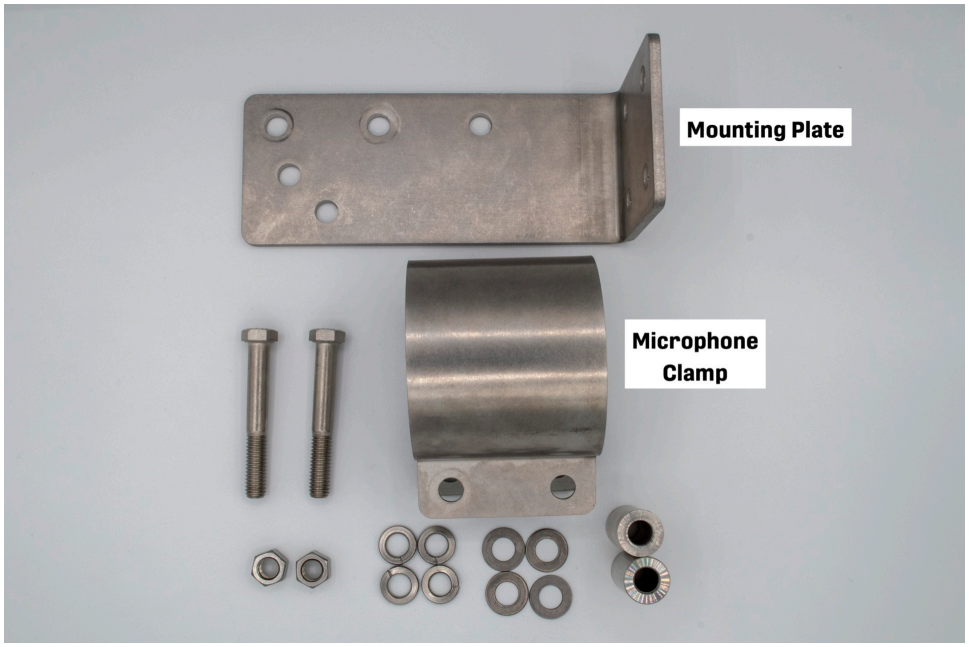
Figure 12. 60° Surface Mounting

Assemble the SMART MIC-1 Mounting Bracket

The SMART MIC-1 bracket can be quickly assembled by hand prior to installing the microphone on-site.

Before you begin

You will need the following components, included with the SMART MIC-1 Mounting Bracket kit.

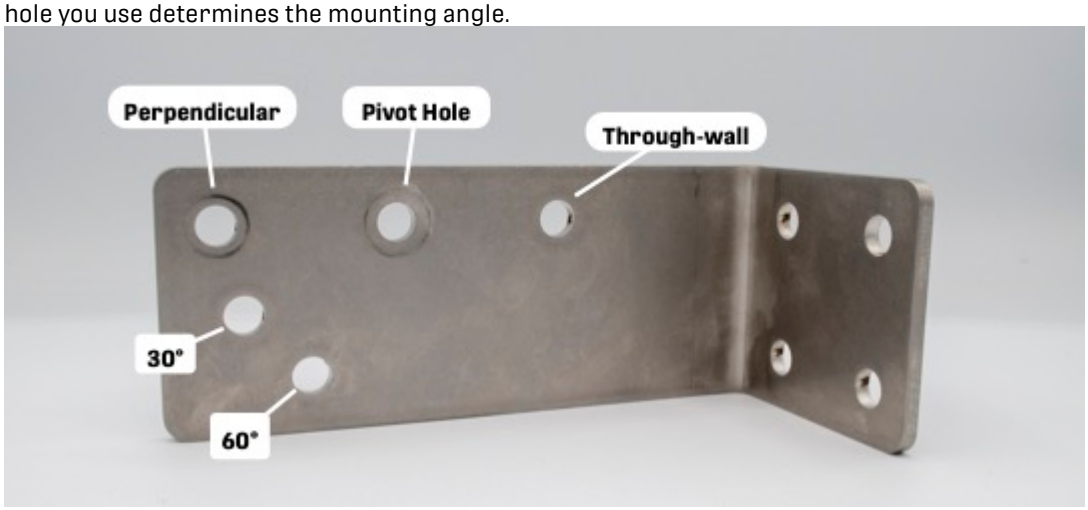


Quantity	Part	Specifications
1	Mounting Plate	
1	Microphone Clamp	

Quantity	Part	Specifications
2	Bolt	• Partially threaded • 2-3/4 in. long • 3/8 in.-16 thread size • 9/16 in. hex head
4	Lock Washer	• 3/8 in. screw size
4	Flat Washer	• 3/8 in. screw size
2	Cylindrical Spacer	• Unthreaded • 0.380 internal diameter
2	Nut	• 3/8 in.-16 thread size • 9/16 in. head head

Procedure

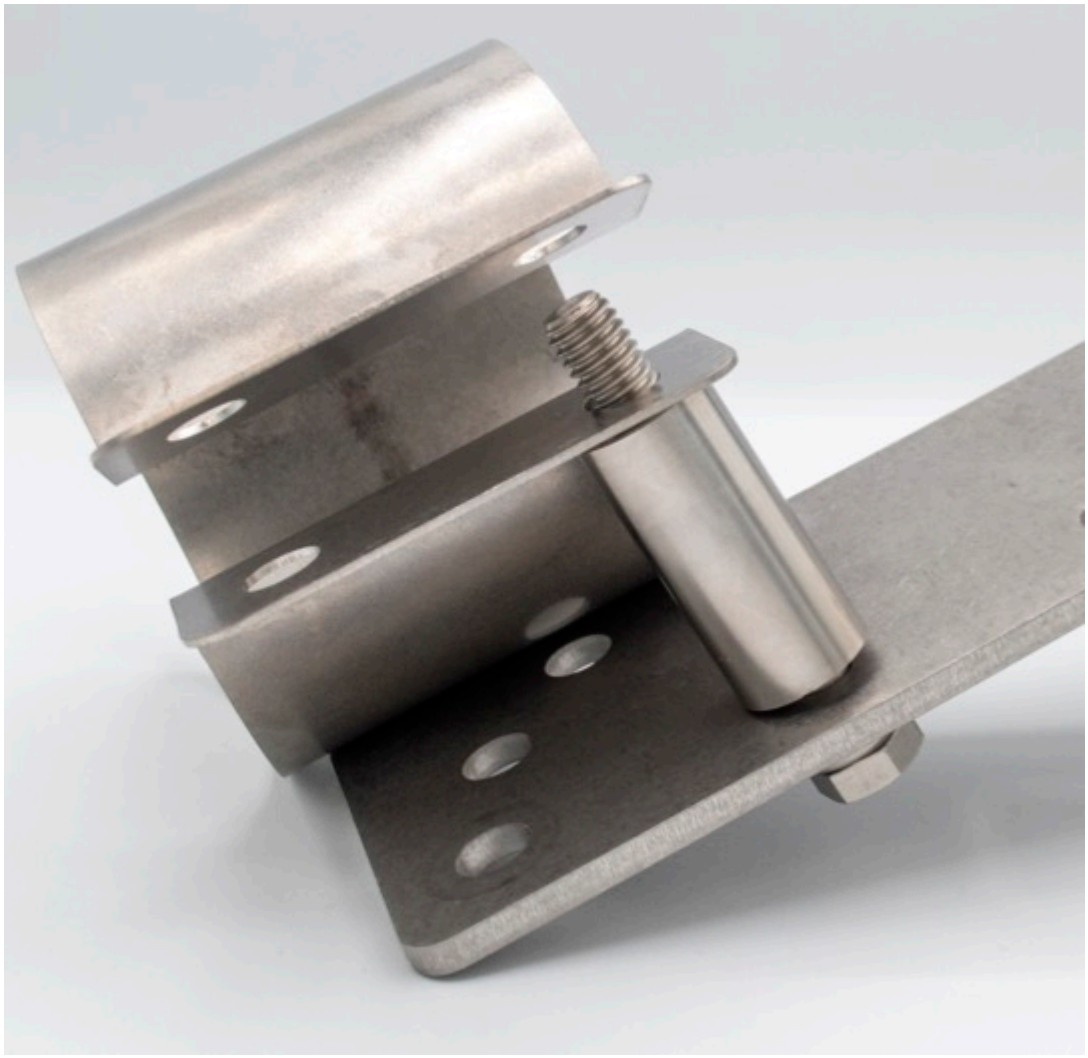
1. Decide which two of the Mounting Plate's holes to use based on the desired mounting angle. You should always use the Pivot Hole, in the middle of the Mounting Plate's longer face. The second hole you use determines the mounting angle.



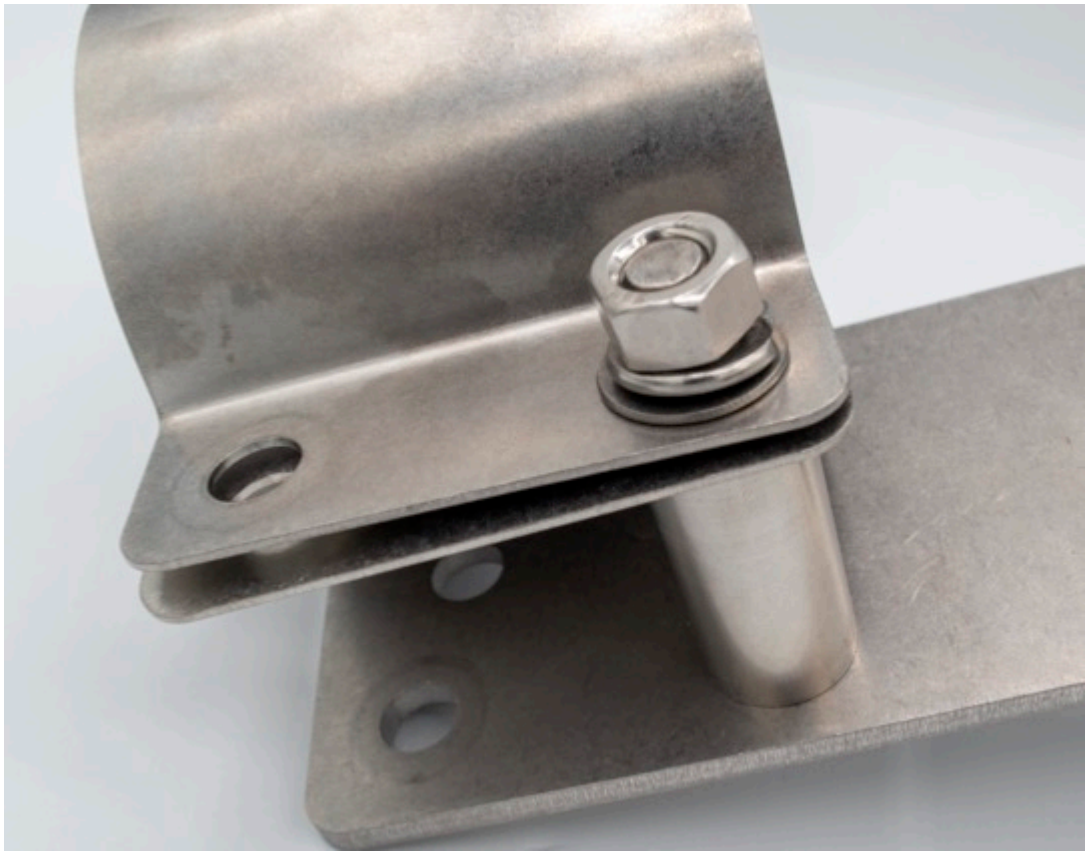
2. Slide one Lock Washer onto a Hex Bolt, followed by one Flat Washer.

3. Insert the Hex Bolt through the Mounting Plate through the first mounting hole.

4. Slide one Cylindrical Spacer onto the Hex Bolt, followed by both flanges of the Microphone Clamp.

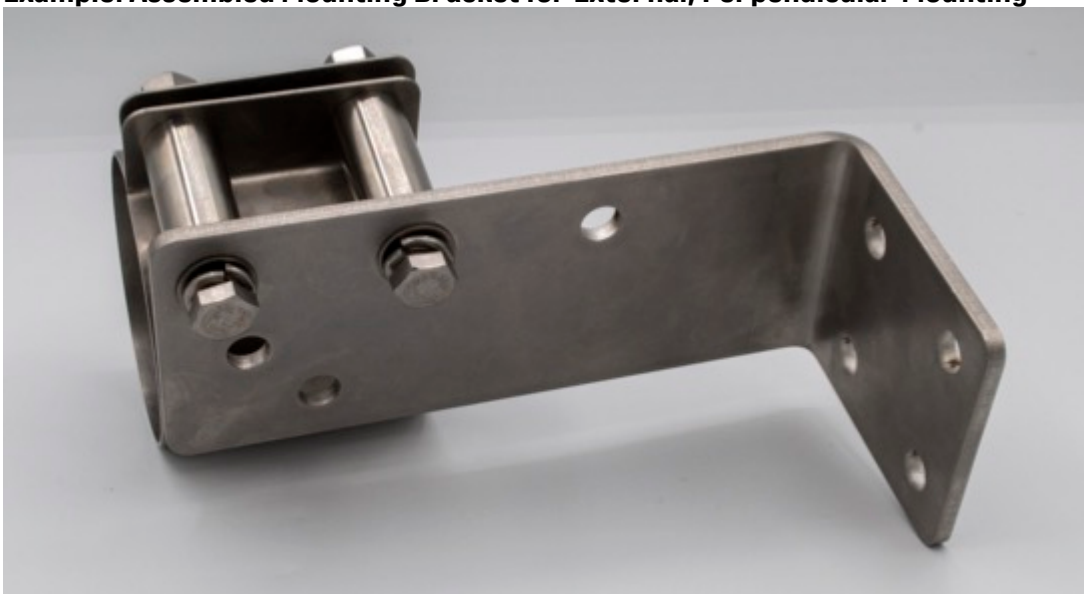


5. Holding the Microphone Clamp closed, slide one Flat Washer and one Lock Washer onto the bolt, then thread one Nut onto the bolt to hold the Microphone Clamp in place. Do not fully tighten the nut yet.



6. Repeat steps 2 *(on page 85)* through 5 *(on page 87)* for the second Hex Bolt and second mounting hole.

Example: Assembled Mounting Bracket for External, Perpendicular Mounting



Install the SMART MIC-1 in the Mounting Bracket

To ensure the SMART MIC-1 mounting bracket can hold the microphone securely and provide a reliable connection from the body of the microphone to ground, you must apply conductive grease and threadlocker and assemble the bracket with the specified torque.

Before you begin

You should assemble the SMART MIC-1 Mounting Bracket before installing the SMART MIC-1.

You will need the following tools and parts (not provided by Wildlife Acoustics):

- 3/8 in. or M10 diameter bolts to attach the bracket to the mounting surface
- Threadlocker: Loctite® Threadlocker Blue 242 or equivalent
- Conductive grease: CAIG DeoxIT® L260DCp or equivalent
- Rag or paper towel
- Torque wrench set to 15 ft.-lb. (20 Nm) and compatible with your 3/8 in. or M10 diameter bolts

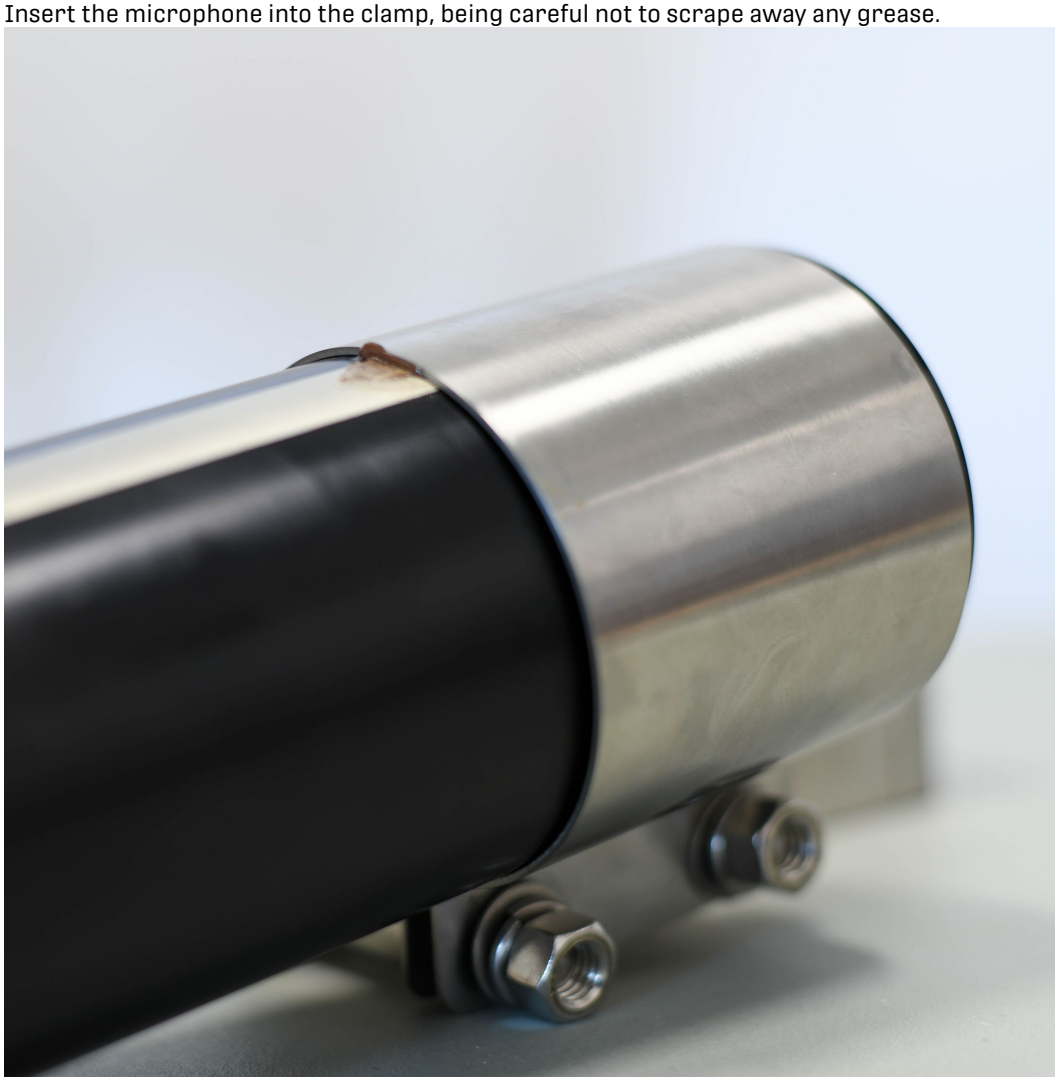
Procedure

1. Reposition the bolts connecting the clamp to the body of the mounting bracket to achieve the desired mounting angle.
2. Loosen the mounting bracket bolts to open the clamp slightly.
3. Position the SMART MIC-1 in the clamp and apply conductive grease.
    a. Ensure the grounding strip on the SMART MIC-1 makes contact with the bracket.
    b. Apply a thin line of conductive grease to the grounding strip of the SMART MIC-1 where it will contact the mounting bracket.

    c. Using a rag or paper towel, smear the grease to ensure the grounding strip is fully coated where it will contact the clamp.

d. Orient the grounding strip opposite from the opening of the clamp to ensure full contact.

e. Insert the microphone into the clamp, being careful not to scrape away any grease.



4. Apply threadlocker and tighten the clamp bolts to secure the microphone.
    a. Use a clean rag to wipe debris and oil from the threads of the clamp bolts.
    b. Apply threadlocker to each clamp bolt where it will contact the nut.
    c. Thread the nut onto each bolt and tighten to 15 ft.-lb. (20 Nm).
    d. Wipe away excess threadlocker with a damp rag.

Example: SMART MIC-1 Installed in Mounting Bracket

What to do next

When you install the SMART MIC-1 and Mounting Bracket on-site, be sure to ground the mounting bracket in order to protect the SMART MIC-1 and SMART Controller from electrical discharges. Consult with an electrician or electrical engineer for detailed grounding guidance.

Connecting the SMART MIC-1 to the SMART Controller

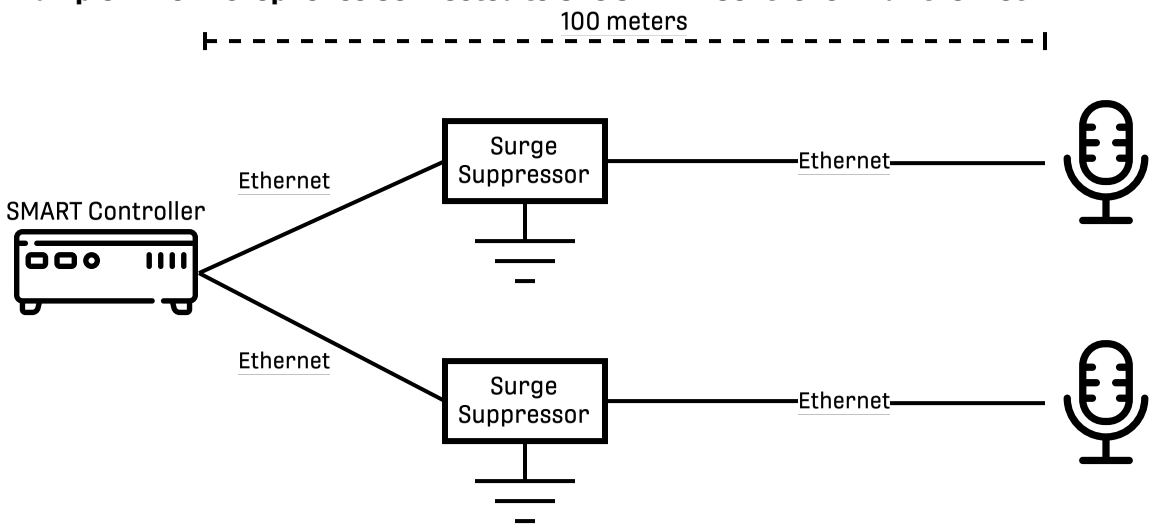
In the simplest configuration, the SMART Controller can connect to two microphones via Ethernet cables with *PoE* support, up to 100 meters in length for a direct connection.

Each SMART microphone ships with two accessories that should be used as part of each deployment:

- A kit for securing an Ethernet cable to the microphone and weatherproofing the connection
- An Ethernet Surge Suppressor for diverting electrical surges away from the SMART Controller

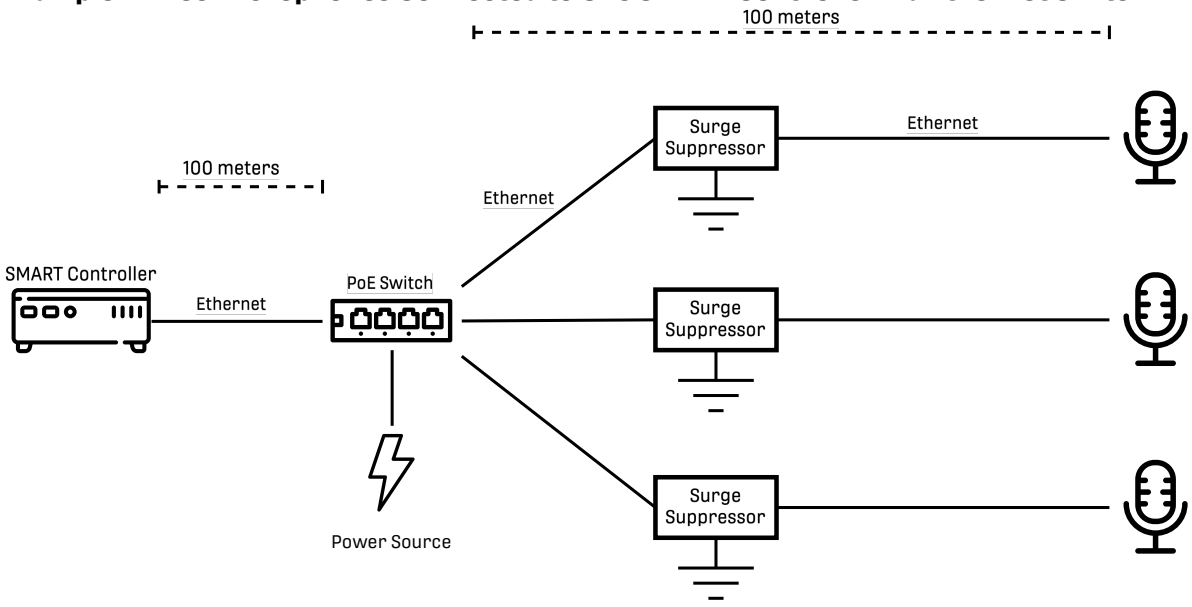
To connect more microphones or to connect over distances greater than 100 meters, *PoE* switches or extenders can be used between the SMART controller and each microphone.

Example: Two Microphones Connected to One SMART Controller via Ethernet



Two microphones can be connected via ethernet directly to the SMART Controller's **PoE+** ports, which provide *PoE* to the microphones. An in-line, grounded surge suppressor is installed between the SMART Controller and each microphone to protect the SMART Controller. The total distance between the SMART Controller and each microphone can extend up to 100 meters without signal degradation.

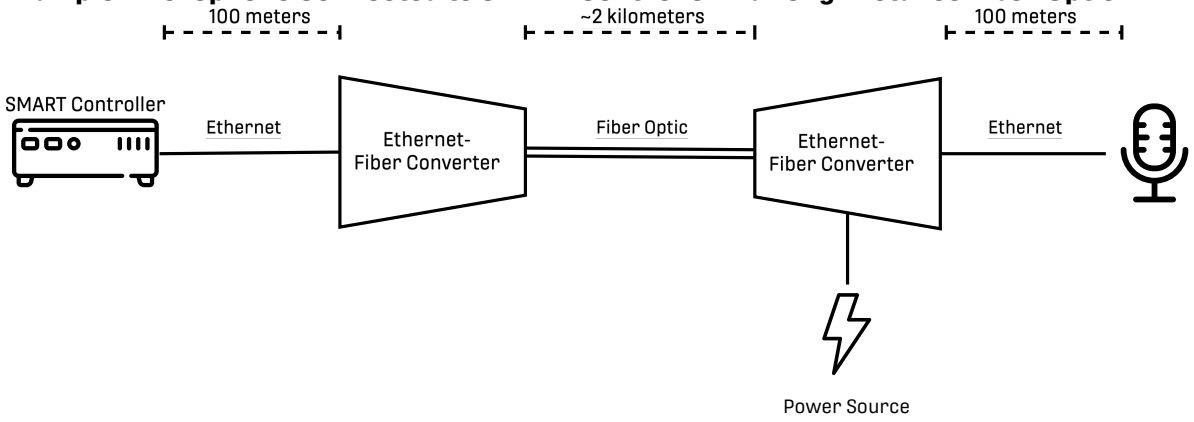
Example: Three Microphones Connected to One SMART Controller via Ethernet Switch



More than two microphones can be connected to a SMART Controller if their signals are combined using a powered, PoE-compatible Ethernet switch. Because the switch is powered and actively repeats outgoing signals, an additional 100-meter span of cable can be added to the total distance.

Typically, microphones are the component most exposed to electrical discharges. Therefore, surge suppressors are installed in between the switch and each microphone to protect all downstream components.

Example: Microphone Connected to SMART Controller via Long-Distance Fiber Optic



Between the SMART Controller and the microphone, the signal is converted from Ethernet to fiber optic and back. To power the microphone, the signal converter on the microphone's end of the fiber optic run must be powered and able to provide *PoE*, or *PoE* must be injected through some other means.

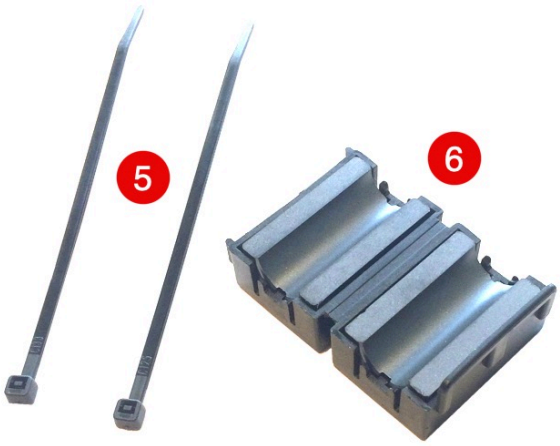
Assemble the RJ45 Weatherproofing and Ferrite Kit

Each SMART microphone ships with a collection of accessories that protect the microphone's Ethernet connection from the weather and help reject electromagnetic interference. The weatherproof cover also provides a screw-on connection that securely holds the Ethernet connection in place.

Before you begin

You will need the following materials:

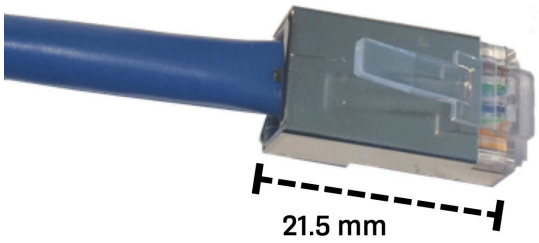
- The Ethernet cable (CAT 5 minimum) that you will connect to your SMART microphone.
  The length of the Ethernet plug on the end of the cable should be approximately 21.5 mm. A plug that is too long may compromise the ability of the Weatherproofing kit to keep water out.
- **Optional:** A craft knife for removing the molded cover from a commercial Ethernet cable.
- The RJ45 Weatherproofing and Ferrite Kit that shipped with your SMART Microphone. This kit includes the following:



1. 1x Connector Body
2. 1x Cable Sleeve
3. 1x Waterproof Grommet
4. 1x End Collar
5. 2x Cable Ties
6. 1x Ferrite

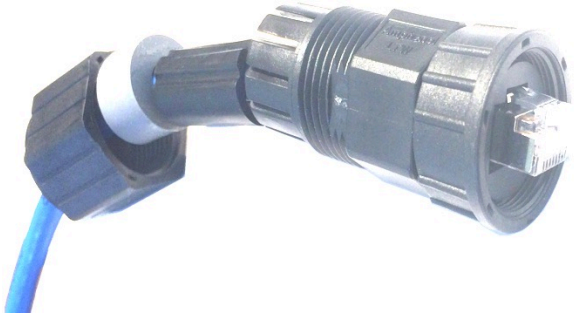
Procedure

1. Make sure the microphone end of your Ethernet cable fits through the end collar.
   If your cable includes a molded plastic sheath, use a craft knife to remove it and expose the bare RJ45 plug underneath. Ensure that the plug is approximately 21.5 mm in length.



2. Thread the microphone end of your Ethernet cable through the End Collar.
3. Install the Waterproof Grommet and Cable Sleeve in between the End Collar and the RJ45 plug.

4. Thread the microphone end of your Ethernet cable through the Connector Body, making sure the Ethernet plug is correctly aligned.
   The Ethernet jack should protrude from the Connector Body slightly.



5. Push the Cable Sleeve into the Connector Body, followed by the Waterproof Grommet.
   The Waterproof Grommet should be flush with the end of the Connector Body.

6. Push the End Collar onto the Connector Body and then turn clockwise to tighten, making sure to line up the threads.
   The End Collar should tighten against the Connector Body, creating a weatherproof seal.

7. To make sure you have assembled the cover correctly, connect the Ethernet cable to the microphone and screw the Connector Body's collar onto the microphone's threaded Ethernet port.

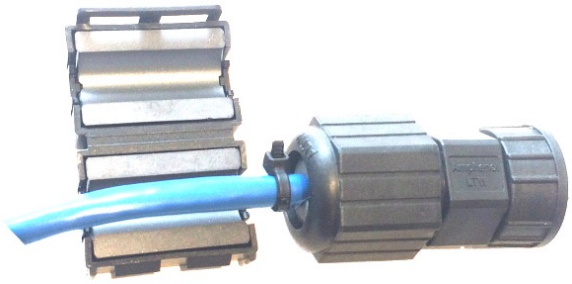
In the following steps, you will install the Ferrite next to the installed Ethernet Cover. Installing the Ferrite as close as possible to the microphone ensures its effectiveness. You will also install two Cable Ties to keep the ferrite in place and prevent it from knocking against the Ethernet Cover.

8. **Notice:** Only tighten Cable Ties enough that they stay in position on the cable. Overtightening can cause the Cable Tie to damage the Ethernet cable.

   Attach the first Cable Tie to the cable, as close as possible to the End Collar.

The first Cable Tie creates a small gap between the Ferrite and End Collar, preventing the two from bumping together and creating noise.

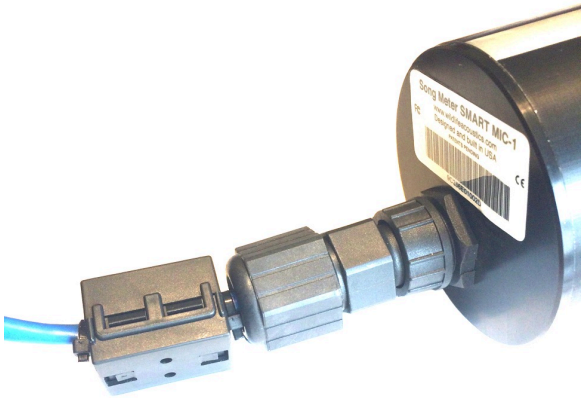
9. Position the Ferrite adjacent to the first Cable Tie and fold it together so it snaps shut around the Ethernet cable.



10. Attach the second Cable Tie adjacent to the Ferrite to prevent it from sliding down the cable.

Results

The Ethernet cable can now be screwed onto the microphone's threaded Ethernet port using the weatherproof cable cover. The ferrite should be held firmly in place next to the cable cover using two Cable Ties.



Install the Ethernet Surge Suppressor

Each SMART microphone ships with an Ethernet Surge Suppressor. When installed correctly, the Surge Suppressor provides some protection against electrical surges or discharges to equipment downstream from the microphone.

About this task

You should install one Surge Suppressor for each microphone you install as part of your SMART System.

The Surge Suppressor is symmetrical. You can connect the microphone to either of the two RJ45 ports.

Procedure

- Connect the Surge Suppressor's ground wire to common ground using as short a connection as possible.
- Install the Surge Suppressor as close as possible to the end of the cable opposite the microphone.

  Example
  If you want to connect the microphone directly to the SMART Controller, connect the Surge Suppressor to the SMART Controller with as short an Ethernet cable as possible. Connect the longer Ethernet cable between the microphone and the Surge Suppressor.

  Example
  If you are using a PoE switch in between your microphone and SMART Controller, connect the Surge Suppressor to the PoE switch with as short an Ethernet cable as possible. Connect the longer Ethernet cable between the microphone and the Surge Suppressor.
- Use Ethernet cables that are at least CAT 5.
- Keep the total run of Ethernet cable, including both segments attached to the Surge Suppressor, to 100 meters or shorter.
  The Surge Suppressor is a passive component and will not extend the possible range of a single Ethernet segment. To extend the possible range, use a *PoE* switch, signal extender, or Ethernet-to-fiber conversion system.

Extending the Connection between the SMART MIC-1 and SMART Controller

A direct Ethernet connection can carry power and signals between a SMART Controller and a SMART MIC-1 for distances up to 100 meters. To cover longer distances or to minimize electrical interference, you can use an Ethernet signal repeater, powered switch, or Ethernet-to-fiber-optic conversion system.

Ethernet Repeaters, Extenders, and Switches

A device such as an Ethernet signal repeater/extender or a powered Ethernet switch can relay an Ethernet signal along an additional 100-meter span of Ethernet cable. Multiple repeaters, extenders, or switches may be used in series. Such devices may be powered by *PoE* or may require dedicated power supplies.

In all cases, ensure that *PoE* is provided to each SMART MIC-1.

Ethernet-to-Fiber Media Conversion

Commercially available systems can transmit an Ethernet signal between two points using a fiber optic connection, converting it back to a standard RJ45 Ethernet connection on each end. Specifications vary, but these products may allow you to connect a microphone to a SMART Controller over distances up to two kilometers.

Fiber optic lines cannot transmit power. Therefore, **you must provide *PoE* to the SMART MIC-1** on its end of the fiber optic line.

7. Bat Activity Alarms and Control System Integration

The SMART System can integrate with a windfarm SCADA network or similar control system to perform *ATC*. You can define custom alarms that detect bat activity based on custom-defined parameters. When these alarms trigger, you can send control signals to other devices on a control system to trigger turbine shutdowns.

Before you start using SMART to trigger turbine shutdowns, you can Kaleidoscope Pro Sound Analysis Software to simulate the outputs of alarms using data recorded from your SMART System. Once you have tested and refined your alarms in Kaleidoscope Pro, you can implement them on the SMART System via the SMART Control Panel.

When alarms are enabled, other devices on a control system can monitor the state of each alarm via Modbus (over a serial connection or TCP), OPC UA, or the programmable digital I/O pins on the SMART Controller.

Alarms

As the SMART System is recording and analyzing, it can trigger alarms when the characteristics of the bat call meet certain parameters. The status of these alarms can be polled or subscribed to by other devices on a control system (such as a SCADA network), which can then send shutdown signals to a turbine.

A single SMART System can have eight pairs of alarms. In each pair, one alarm operates on individual echolocation pulses, and the other operates on *passes*, sequences of pulses recorded as a bat flies past the microphone. For each pair of alarms, you can define matching call characteristics using Boolean logic to determine what kinds of sounds will satisfy the alarms. This flexible logic allows you to avoid false alarms from non-bat sound sources and define unique alarm behavior for different bat species or different microphones depending on their placement.

Each pair of alarms shares a set of call characteristics but has independent settings for the number of matching pulses or passes in a given period required to raise the alarm. You may decide to use one or both alarms in each pair.

Alarm Logic Overview

Each pair of alarms consists of *Pass Settings*, *Pulse Settings*, and *Alarm Criteria*. Together, these settings define the conditions required to raise a given alarm.

The Alarm Criteria define what kinds of pulses and passes count towards the alarm based on recorded call parameters and Boolean logic. The Pass and Pulse Settings set the length of time over which passes or pulses are counted and how many matches are required within a given period to raise the alarm.

Alarm Criteria

Alarm Criteria are defined by Boolean expressions acting on information about a given pulse or pass and the microphone that recorded it. Pulses or passes that meet the criteria are then processed by the Pass Settings and Pulse Settings to determine if either the pass-level alarm or pulse-level alarm should be raised.

Pulse-Level Alarm Behavior

Pulse-level alarms operate on the measurements of each individual pulse recorded by the SMART System, without regard for how pulses are separated into passes based on each microphone's trigger settings.

Each pulse alarm has a **Pulse Period**, a rolling time window over which it counts the number of pulses that meet the **Alarm Criteria**. *Pulse activity* can be defined as the number of matching pulses within the trailing **Pulse Period**.

A pulse-level alarm follows this sequence:

1. If pulse activity exceeds the **High Water Mark**, the alarm is raised.
2. Once the alarm is raised, it will continue to stay raised while pulse activity remains above the **High Water Mark**.
3. How the alarm ends depends on whether a **Pulse Alarm Hold** is set:
   If no **Pulse Alarm Hold** is set, when pulse activity drops below the **Low Water Mark**, the alarm ends.

   If a **Pulse Alarm Hold** is set, the logic follows this sequence:

   a. When pulse activity drops to or below the **Low Water Mark**, the **Pulse Alarm Hold** period starts counting down.

   b. If pulse activity exceeds the **High Water Mark** again before the **Pulse Alarm Hold** period has elapsed, the alarm remains active and the **Pulse Alarm Hold** period resets to its full value.

   c. After pulse activity drops to or below the **Low Water Mark** and does not rise above the **High Water Mark** for the duration of the **Pulse Alarm Hold** period, the alarm ends.

Pass-Level Alarm Behavior

The pass-level alarm acts on the average measurements of multiple pulses, trigger-by-trigger. How many bat pulses constitute a single pass and how passes are separated from each other in time are determined using the trigger settings defined in a Setting Profile.

After each triggered recording, the SMART System calculates average (or cumulative, in the case of pulse count) call parameters for all of the pulses recorded during the trigger. These parameters are processed using the same Alarm Criteria as the pulse-level alarm. Match passes are then processed by the Pass Settings. Though the pulse-level alarm and pass-level alarm use the same **Alarm Criteria**, they can have different settings to determine how many positive matches are required to raise the alarm, and within what time period.

Pass activity can be defined as the number of passes that match the **Alarm Criteria** within the trailing **Pass Period**.

A pass-level alarm follows the same basic logic as pulse-level alarms:

1. If pass activity exceeds the **High Water Mark**, the alarm is raised.
2. Once the alarm is raised, it will continue to stay raised while pass activity remains above the **High Water Mark**.
3. How the alarm ends depends on whether a **Pass Alarm Hold** is set:
   If no **Pass Alarm Hold** is set, when pass activity drops below the **Low Water Mark**, the alarm ends.

   If a **Pass Alarm Hold** is set, the logic follows this sequence:

   a. When pass activity drops to or below the **Low Water Mark**, the **Pass Alarm Hold** period starts counting down.

   b. If pass activity exceeds the **High Water Mark** again before the **Pass Alarm Hold** period has elapsed, the alarm remains active and the **Pass Alarm Hold** period resets to its full value.

   c. After pass activity drops to or below the **Low Water Mark** and does not rise above the **High Water Mark** for the duration of the **Pass Alarm Hold** period, the alarm ends.

Alarm Strategy

The general goal when designing alarms is to count bat pulses or passes near a turbine without counting non-bat noises. Wind turbines can often produce ultrasonic noise, so a set of alarms should be tailored to exclude any noise produced by a particular turbine while including the echolocation calls of bat species in the area.

Each individual turbine can produce noise with different characteristics, and the prominence of the noise can vary depending on where each microphone is positioned. Often this noise can bear superficial resemblance to bat echolocation calls. This means that, typically, you must tailor alarms to each individual turbine to capture nearby bat activity while excluding the noise of the turbine.

Crafting effective alarms for a specific turbine should start with installing the SMART System on the turbine and recording both nearby bat activity and the various kinds of noises the turbine creates in different weather conditions. Once you have collected sample data, you can use the Kaleidoscope Pro Sound Analysis Software to design alarms and simulate how they would have reacted to your data in real time.

Once you have tested your alarms using sample data, you can implement them on the SMART System itself. Program other devices on your control system to read the status of your alarms and use that information to prompt turbine shutdowns.

Design, Test, and Implement Alarms

Kaleidoscope Pro Sound Analysis Software allows you to test SMART Alarms on batches of recordings to see how the alarms would have processed the recordings in real time. This functionality is vital for tailoring your alarms to a particular turbine before implementing *ATC*.

Record Sample Data for Alarm Design

In order to design effective alarms for *ATC* on a particular turbine, you should collect sample recordings from that same turbine. The characteristics of the turbine machinery, local weather, and where microphones are placed on the turbine will affect what kinds of noise your microphones records.

Before you begin

The SMART System's alarms will only process sounds that satisfy the pulse-level trigger parameters defined in a Setting Profile, including minimum and maximum frequency and duration criteria. When recording sample data, you may wish to set these criteria to be broader than normal to ensure that all possible bat activity is recorded and accounted for.

About this task

There is no universally effective set of alarms for *ATC*. Effective alarms detect nearby bat activity while preventing false triggers from the noise of the turbine. On each turbine, you will record noise with different characteristics and will need to employ different strategies to separate the mechanical noise from bat vocalizations. Therefore, collecting sample data from each turbine is key to designing effective alarms for *ATC* on that turbine.

Procedure

- Record sample data using the same microphone placements you will use during *ATC*.
  Changing the placement of a microphone may change the characteristics of the mechanical noise you record.
  Changing the placement of a microphone can also increase or decrease the presence of reflections of bat vocalizations off of hard surfaces. These reflections can change the apparent duration of nearby bat activity.
- Record sample data in a variety of weather conditions to capture different turbine behaviors.
  Example
  The characteristics of a turbine's mechanical noise may vary with wind speed.
- Record across a long-enough timespan to capture seasonal variations in bat behavior.
- Record both WAV and ZC files.
  Actual alarm behavior is based on the SMART System's ZC output, so ZC files should be used when testing alarm behavior in Kaleidoscope Pro. WAV files display fuller information about the audio being captured, so they are useful when manually examining files during testing.

What to do next

As you collect sample data, you can start reviewing it for common types of noise that you will need to separate from the recorded bats. You can start designing and testing alarms in Kaleidoscope Pro, and you can refine the alarms as you collect more data.

Review Sample Data for Bat and Noise Characteristics

To design effective alarms, you need to look for patterns in the types of bat calls and types of non-bat noise that any particular microphone is recording.

Procedure

1. Vet your recordings to identify bat species and also mark any recordings that contained non-bat noise.
2. Using a spreadsheet processor, look for patterns in call parameters that could be used to distinguish bats from noise.

   Example

   Imagine a common type of noise in your recordings produces shorter pulses than low-frequency bats, but narrower bandwidth than high-frequency bats. One strategy for distinguishing between bat calls and noise would be to create separate alarms for low- and high-frequency bats.

   The low-frequency alarm would look for sounds only up to a certain maximum frequency with pulses longer than a certain duration.

   The high-frequency alarm would look for sounds above a minimum frequency, wider than a specific bandwidth.

What to do next

Design alarms to take advantage of the patterns you have found in your data. Then, test how effectively they trigger for bat activity without triggering for noise.

Simulate Alarms in Kaleidoscope Pro

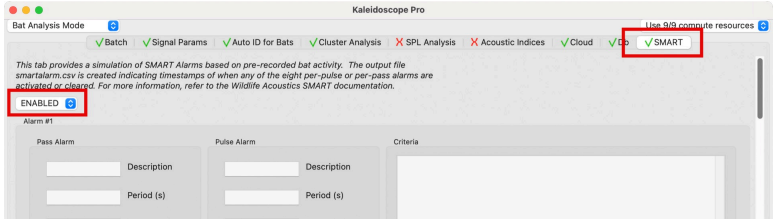
With sample recordings collected from your SMART System, you can run alarms in Kaleidoscope Pro Sound Analysis Software to simulate how those alarms would have responded to the audio in real time.

Before you begin

You should have collected sample data from the turbine where you want to implement *ATC*, and you should have analyzed the data for patterns in call parameters that can distinguish bat calls from other noises.

Procedure

1. Open Kaleidoscope Pro and configure your **Batch** settings.
   a. Select **Bat Analysis Mode**.
   b. Set your **Input directory** to the directory containing your sample data.
   c. Enable **ZC files** for input, assuming your sample data is in ZC format.
   d. Set your **Output Directory** to a blank directory separate from the **Input Directory**.
   e. Disable **WAV (or W4V) files** and **ZC files** for output.
2. Go to **SMART** and enable alarm simulation.



3. Configure one or more alarms.
    - To create a pulse-level alarm, select an unused alarm and configure **Alarm Criteria** and **Pulse Alarm** settings.
    - To create a pass-level alarm, select an unused alarm and configure **Alarm Criteria** and **Pass Alarm** settings.

    Ideally, each bat call should cause at least one alarm to register as true, while non-bat noise should cause all alarms to register as false.
4. Select **Process Files**.

Results

Your output directory will now contain a file named `smartalarm.csv`, which records when each of your alarms would have raised and cleared in response to the incoming audio.

Review Alarm Simulation Results

After you have run a SMART alarm simulation in Kaleidoscope Pro, you should review the file `smartalarm.csv` together with your vetted bat identification results. Known bat activity should raise your alarms, while non-bat noises should be ignored.

Procedure

1. Open both the `smartalarm.csv` and a spreadsheet containing your bat species annotations in a spreadsheet processor.
2. Merge the data from both files into a single spreadsheet with their `Date` and `Time` columns matched, and sort all entries chronologically.
   The combined spreadsheet will show when each alarm raised and cleared in relation to the recordings that were produced.
3. Review the table and look for signs that your alarms processed the recordings incorrectly. These errors can come in two forms:
    - Bat activity failed to raise the alarm.
    - Non-bat noises raised the alarms.
4. When you find errors, examine the relevant recordings to look for characteristics that caused the incorrect behavior.

   Example
   If a recording of noise raised an alarm, look for features of the noise that would have met your alarm criteria.
5. Implement any necessary refinements to your alarm criteria, and run a new simulation to see if results have improved.

Implement Alarms on a SMART System

Once you have designed and tested alarms in Kaleidoscope Pro, you can implement them on a SMART System via the SMART Control Panel.

Procedure

1. Open the SMART Control Panel and go to **SCADA**.
2. **Select an Alarm** to open the **Pass Settings**, **Pulse Settings**, and **Alarm Criteria** panels.
3. Select **Clear Settings** to reset the selected alarm.
4. Edit the settings for one or multiple alarms and select **Apply Changes**.

What to do next

Configure the necessary settings to make the results of your alarms accessible to devices on your turbine control network. You can access the results of your alarms via Modbus, OPC UA, or by implementing a custom bash script in the file `/var/www/html/storage/smart-scada-hook.sh`.

Related information

Control System Integration *(on page 107)*

Alarm Criteria Syntax

Alarm Criteria can include many different arithmetic and logical operations on pulse- and pass-level criteria as well as on the prefix assigned to each microphone.

Numerical Boolean Operators

Operator	Definition
`=`	Equals
`!=` or `<>`	Does not equal
`<`	Is less than
`<=`	Is less than or equal to
`>`	Is greater than
`>=`	Is greater than or equal to

Numerical Arithmetic Operators

You can use addition, subtraction, and parentheses to construct logical tests using pulse and pass characteristics and numerical constants.

Example: Using Parentheses and Subtraction to Test Call Bandwidth

The following expression returns true if the difference between the maximum frequency and minimum frequency is less than 15,000 Hz.

```
( Fmax - Fmin ) < 15000  # Call bandwidth less than 15 kHz
```

Pulse and Pass Characteristics

Pulse and pass characteristics in Alarm Criteria are case-insensitive. For example, *fc*, *Fc*, *fC*, and *FC* all mean *characteristic frequency*.

With the exception of *N*, pass-level alarms operate on the averages of these characteristics across all pulses in a single pass.

Operand	Unit	Definition
N	Pulses	Number of pulses in a pass (equal to 1 for pulse-level alarms)
Fc	Hertz	Characteristic frequency
Sc	Octaves per second	Characteristic slope
Fmax	Hertz	Maximum frequency
Fmin	Hertz	Minimum frequency
Fmean	Hertz	Mean frequency

Operand	Unit	Definition
TBC	Seconds	"Time between calls," or time between individual pulses (equal to 0 for pulse-level alarms)
Fk	Hertz	Knee frequency
Tk	Seconds	Time of the knee after the pulse onset
S1	Octaves per second	Slope before the knee
Tc	Seconds	Time of the characteristic after the pulse onset
Dur	Seconds	Pulse duration

String Operands

Currently, the only string operand is *prefix* (case-insensitive), which is the identifying name given to each microphone.

String Boolean Operators

Operator	Definition
`=`	Equals (case-sensitive)
`~`	Includes substring (case-sensitive)

Example: Boolean Operators on Prefix

Applying Alarm criteria to *prefix* means you can apply separate logic for each microphone connected to a single SMART System.

Say a SMART System includes two microphones with prefixes **TowerHigh** and **TowerLow**, respectively.

The following expressions will return true for any pulses recorded by the **TowerHigh** microphone:

```
prefix = "TowerHigh"
```

```
prefix ~ 'High'
```

The following expression will always return false because string boolean operators are case-sensitive:

```
prefix ~ "high"
```

Combining Boolean Expressions

Boolean expressions can be combined using the following operators:

- `AND`
- `OR`
- `NOT`

These operators are **case-insensitive**.

Comments

The `#` marks everything to its right as a comment until the end of the line.

Example

```
Prefix = "TowerHigh"   # Only test recordings from high microphone
AND Fc > 40000         # Characteristic frequency > 40 kHz
```

Control System Integration

The SMART System provides several means through which other devices can retrieve information about detected bat activity in real time. This information can be used to inform automatic wind turbine curtailment or similar control decisions.

The SMART System provides many information fields about the system status, measurements of incoming bat pulses and passes, and the status of bat activity alarms as Modbus registers. Outside devices can poll these registers directly via Modbus/TCP or Modbus/RTU, or they can subscribe to changes to these fields via *OPC UA*.

When the status of any bat activity alarm changes, the SMART will execute a bash script saved to the path `/var/www/html/storage/smart-scada-hook.sh` if one exists. You can use this script to signal multiple alarm states via the SMART Controller's digital I/O pins or perform any other function via a custom script.

Contents of each row in each microphone's `results.csv` file can be output from the SMART Controller's serial ports. This includes average measurements of call characteristics in each triggered audio file and basic status information about the SMART Controller, but it does not include any information about the status of the SMART System's bat activity alarms.

Related information

Implement Alarms on a SMART System *(on page 104)*

Control System Integration with OPC UA

OPC UA is a machine-to-machine communication protocol used for industrial automation developed by the OPC Foundation. The SMART Controller OPC UA interface provides a secure mechanism for retrieving status information, system updates and alarm notifications from a SMART device.

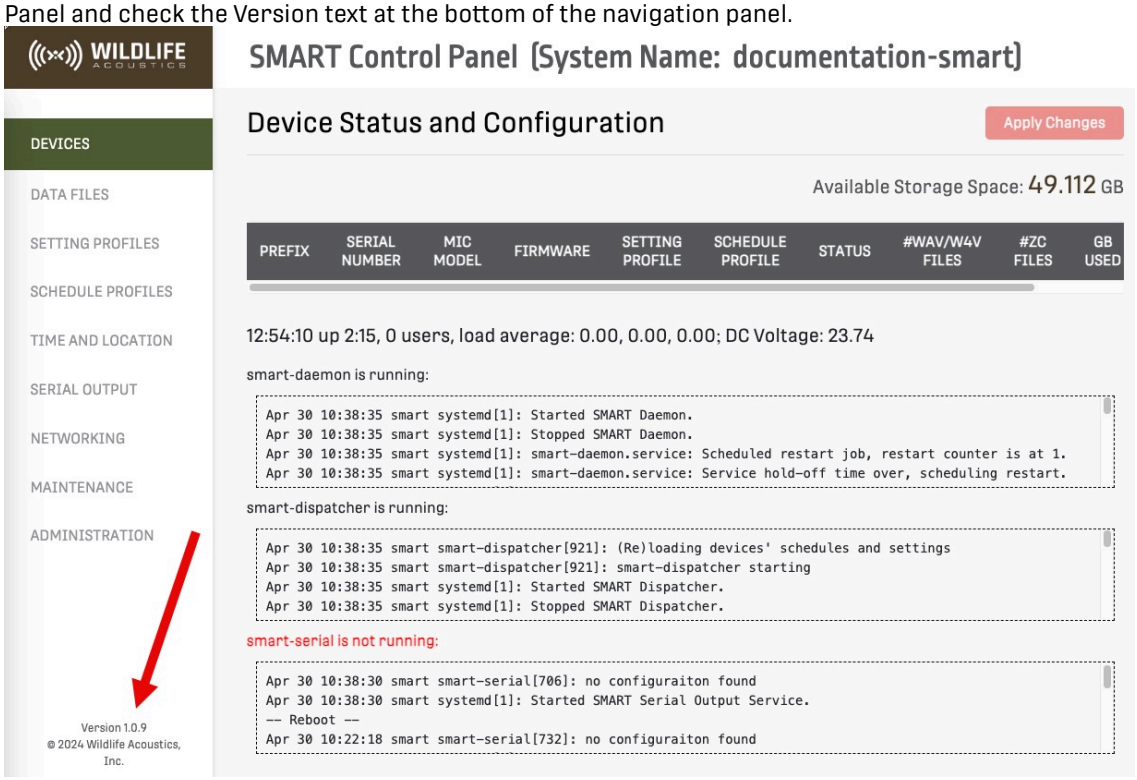
OPC Clients are able to take appropriate actions according to local needs, including bat curtailment processing. The Information Model used by the SMART Controller provides a flexible and comprehensive set of variables, methods and publish/subscribe features to enable full monitoring of a SMART system.

Install and Configure the SMART OPC UA Server

The SMART *OPC UA* Server is installed automatically with the SMART System Software version 1.3.1 or later, for either Ubuntu 18.04LTS or 22.04LTS. In order for the *OPC UA* Server to begin running, you must configure SCADA settings on your SMART Controller.

Procedure

1. To check the currently installed version of the SMART System Software, open the SMART Control Panel and check the Version text at the bottom of the navigation panel.



   If the text displays version 18-1.3.1 or later, or 22-1.3.1 or later, then the *OPC UA* server is installed.

2. In the SMART Control Panel, under the **SCADA** tab, set the **Modbus Port** to TCP and enable at least one alarm.
   Configuring SCADA settings creates the file `/var/www/storage/scada.json`.

3. Configure Secure Certificates for communication between your client and the SMART Controller. Secure Certificates may be installed on the SMART Controller in the directory `/etc/ssl/private`.

4. From a terminal, run the command `systemctl start smart-opcua` to start the OPC UA server.

Shell Commands for the OPC UA Server

The following are useful commands for controlling the *OPC UA* server from a terminal.

Command	Purpose
`systemctl start smart-opcua`	Start the smart-opcua service.
`systemctl stop smart-opcua`	Stop the smart-opcua service.
`systemctl restart smart-opcua`	Restart the smart-opcua service.
`journalctl -u smart-opcua -e -f`	Display OPC UA server log messages as they are generated. Press **Ctrl-C** to stop the output.

Connecting to the OPC UA Interface

The SMART's OPC UA Model is accessible from any client supporting the OPC UA standards (for example, the popular UAExpert client). You can connect to the server using multiple different methods, depending on your network configuration.

The underlying interface is built from the open62541 Server libraries and supports all of the OPC UA foundation's commands and controls. The server supports OPC UA client/server communication, subscriptions, events, method calls and the full range of security paradigms (encryption). The controller's Information Model is built upon the OPC Foundation UA-Model Compiler which utilizes the UA Model Design.xsd core file.

You can connect to a SMART Controller's OPC UA server using either of the following URIs:

- `opc.tcp://hostname.domain:4840`
  hostname is the SMART Controller's hostname provisioned during system configuration. Using this URI assumes *hostname.domain* can be found using your local DNS queries.
- `opc.tcp://IP Address:4840`
  Using this form of the URI assumes the client you are using has access to the SMART Controller's *IP Address*.

Security Certificates

Connecting to the SMART Controller's OPC UA server can be done with openssl certificates. The connection will abide by a Secure Certificate login when the appropriate x.509 certificates are properly exchanged.

On first start, the server will automatically create self-signed files `server_cert.der` and `server_key.pem` files in the `/etc/ssl/private` directory, if they do not already exist. You should copy these files to the appropriate location specified by your OPC UA client.

To have the SMART Controller automatically generate new certificates, remove the existing certificate files and restart the smart-opcua service using the command `systemctl restart smart-opcua`.

Important: The certificates generated by the SMART Controller are **self-signed** and do not include any CA Trust authority. It is highly recommended you place your own certificates in the `etc/ssl/private` directory with the names `server_cert.der` and `server_key.pem` and restart the smart-opcua service.

Using openssl certificate methods mentioned above, these security policies & modes are supported:

- Basic128Rsa15 (Sign, SignAndEncrypt)
- Basic256 (Sign, SignAndEncrypt)
- Aes256_Sha256_RsaPas (Sign, SignAndEncrypt)
- Basic256Sha256 (Sign, SignAndEncrypt)
- Aes128_Sha256_RsaOaep (Sign, SignAndEncrypt)

Customize Startup Arguments for the SMART OPC UA Server

The `smart-opcua` service may be configured with additional startup options to suit the needs of your installation. For example, the default connection port of 4840 can be changed using the `--port` option.

About this task

Startup options can be modified in the `smart-opcua.opts` file which is located in the directory `/etc/defaults`. Simply edit the file and add the options needed to the `SMART_OPCUA_OPTS` setting.

The contents of the default `smart-opcua.opts` file are as follows:

```
# Default settings for smart-opcua
#
# Copyright (c) 2022 Wildlife Acoustics, Inc.
```

```
# All rights reserved.
#
# The opcua service accepts these options:
#
#   --port            TCP Listen Port [4840]
#   --certificate     Full path to the certificate .der file
 [/etc/ssl/private/server_cert.der]
#   --key             Full path to the private key .pem file
 [/etc/ssl/private/server_key.pem]
#   --secure-channel-trust-folder   Full path to folder of secure channel trust certs
 [/etc/ssl/private/trust]
#   --session-trust-folder          Full path to folder of session trust certs
 [/etc/ssl/private/trust]
#
# For example, to override the default listening port:
#   SMART_OPCUA_OPTS="--port=35478"

SMART_OPCUA_OPTS=
```

Procedure

1. As root, edit the file `/etc/default/smart-opcua.opts`.
2. Edit the line containing `SMART_OPCUA_OPTS` and append the desired arguments.
   Example
   `SMART_OPCUA_OPTS="--port=62541"`
3. Save the file, then restart the smart-opcua service using the command `systemctl restart smart-opcua`.
4. Check the logs to ensure the service started successfully using the command `journalctl -u smart-opcua`.
5. Press **Ctrl-C** to close the logs.

Command Line Arguments to the SMART OPC UA Server

Arguments can be used to modify any of the SMART OPC UA Server's default settings, such as available TCP ports and security certificates. These arguments can be passed in the command line when starting the smart-opcua service or added to the file `/etc/default/smart-opcua.opts`.

Argument	Meaning	Default Value
--port	The TCP port the server will accept connections from	4840
--certificate	Full path to `server_cert.der`	`/etc/ssl/private/server_cert.der`
--key	Full path to `server_key.pem`	`etc/ssl/private/server_key.pem`
--secure-channel-trust-folder	Full path to folder of secure channel trust certifications	`/etc/ssl/private/trust`
--session-trust-folder	Full path to folder of session trust certs	`/etc/ssl/private/trust`

SMART OPC UA Information Model

The OPC UA Information Model describes the organization of all information that can be polled via OPC UA. This information is necessary in order to correctly request and interpret outputs from the SMART Controller.

The SMART Information model files can be found in the directory `/usr/local/share/smart/opcua`.

`smartModel.xml`

The UA Model file describing the SMART schema and variables.

`smartModel.csv`

A reference CSV file containing the node IDs for accessing nodes and variables from an OPC UA Client.

`Smart.NodeSet2.xml`

The raw NodeSet2 XML file with node IDs.

Figure 13. OPC UA Information Model Overview

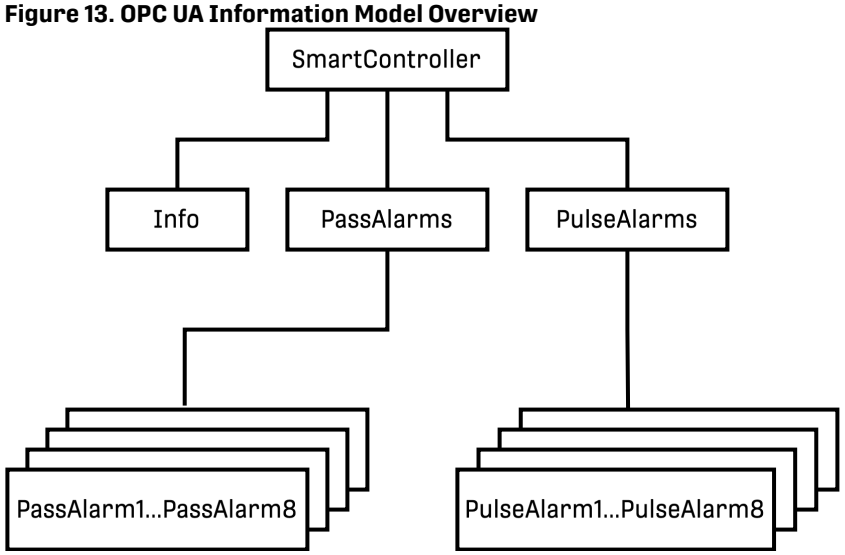


Table 4. OPC UA: SmartController Defined Variables

Variable Name	Description
Hostname	The hostname of the SMART Controller as configured in `/etc/hostname`
CpuUtilization	Current CPU Utilization: 0-100%
DiskUtilization	Current Disk Utilization: 0-100%
MemoryUtilization	Current Memory Utilization
UpSinceDate	The date and time when the SMART Controller last restarted: *YYYY-MM-DD HH:MM:SS*
UpTimeSeconds	The total number of seconds since the SMART Controller last restarted.

Table 5. OPC UA: SmartController defined Methods

Method Name	Description	Input	Output
ReadModbusRegister	Return the value of any Modbus register	Modbus register number	Register value

Table 5. OPC UA: SmartController defined Methods (continued)

Method Name	Description	Input	Output
ReadAllPassAlarms	Return the *AlarmPresent* values for all eight Pass Alarms	None	Array of eight booleans indicating whether each Pass Alarm is raised
ReadAllPulseAlarms	Return the *AlarmPresent* values for all eight Pulse Alarms	None	Array of eight booleans indicating whether each Pulse Alarm is raised

Refer to `/usr/local/share/smart/opcua/smartModel.xml` for lists and descriptions of all Pass and Pulse alarm properties.

OPC UA Events

Your OPC client can subscribe to smart-opcua events published by the SmartController node (*NodeId* = 15740) in order to receive any OPC UA events that may occur. Examples of published events include when the value of a Pass Alarm or Pulse Alarm changes from 1 to 0, or vice-versa.

In addition to the events published from the SmartController node, your client can also subscribe to changes on any individual Pass or Pulse Alarm variable. You can monitor value changes to each Alarm variable in order to take action when the value changes.

Example: OPC UA Event

```
Event(['ConditionSubClassName:None',
    'ConditionSubClassId:None',
    'ConditionClassName:None',
    'ConditionClassId:None',
    'Severity:100',
    "Message:LocalizedText(Locale='en-US', Text='Pass Alarm 1 Raised')",
    'LocalTime:None',
    'ReceiveTime:2023-12-28 01:18:20.609364',
    'Time:2023-12-28 01:18:20.609331',
    'SourceName:fdddde3df7dd',
    'SourceNode:NodeId(Identifier=15087, NamespaceIndex=2,
    NodeIdType=<NodeIdType.FourByte: 1>)',
    'EventType:NodeId(Identifier=15223, NamespaceIndex=2,
 NodeIdType=<NodeIdType.FourByte: 1>)',
    "EventId:b'1\\xf5\\x95\\x19\\xed\\n\\xc9\\xbd\\xcf\\x12;t\\x8f\\xf8\\x84\\x88'"])
```

Control System Integration over Modbus

One method for integrating a SMART System with a SCADA system or similar control system is using the Modbus protocol. Other Modbus-compatible devices on the network can poll the SMART System for information about the system's status and the status of the SMART System's bat activity alarms.

SMART supports Modbus TCP/IP over ethernet and Wi-Fi and Modbus RTU via either of its two DB-9 serial ports, which can be configured for RS-232, RS-422, or RS-485. When Modbus is active, the SMART System acts as a Modbus server. Modbus clients on the same network can poll the SMART to retrieve the status of any of SMART's Modbus registers.

Configure Modbus TCP or Modbus RTU

Before Modbus clients on a SCADA network or other control system can poll the SMART's Modbus registers, and before a *OPC UA* client can interact with the SMART's *OPC UA* server, you must enable Modbus functionality in the SMART Control Panel.

Procedure

1. Connect to the SMART Control Panel and go to **SCADA**.
2. Select a **Modbus Port**.
   - Select TCP if you will connect the SMART to a control system via Ethernet or Wi-Fi. This option is required for *OPC UA* functionality.
   - Select ttyS0 or ttyS1 if you will connect the SMART to a control system via a serial port. ttyS0 corresponds to the serial port nearest to the SMART Controller's power supply input.
3. If you are using a serial port, set the serial communication options to match your control system requirements:

Parameter	Possible Values	Default Value
Baud Rate	50, 75, 110, 134, 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 17900, 38400, 57600, 115200, 230400	9600
Data Bits	5-8	8
Server ID	1-247	1
Parity	None, Even, Odd	None
Stop Bits	1, 2	1

4. Select **Apply Changes**.

What to do next

If you are using a serial port and want to use a standard other than the default RS-232, you must change the serial port mode in the SMART System's BIOS.

Addressable Modbus Registers

You can use Modbus to access information about the SMART System's status and its bat activity alarms. You can also alter some parameters of bat activity alarms using Modbus messages.

Function Code (decimal)	Address (decimal)	Description (where *M* is a digit 1-8 for Alarm *M*) and *X* is any digit 0-9.	Unit
2	1 0000*M*	Alarm *M* bat pass alarm (1 = alarm present, 0 = no alarm)	
2	1 0001*M*	Alarm *M* bat pulse alarm (1 = alarm present, 0 = no alarm)	
2	1 0002*X*	Reserved for customization	
3/6/16	4 000*M*1	Pass Alarm *M* number of rolling seconds for bat pass alarm period	
3/6/16	4 000*M*2	Pass Alarm *M* minimum number of bat passes per period to set alarm (high water)	

Function Code (decimal)	Address (decimal)	Description (where *M* is a digit 1-8 for Alarm *M*) and *X* is any digit 0-9.	Unit
3/6/16	4 000*M*3	Pass Alarm *M* maximum number of bat passes per period to clear alarm (low water)	
3/6/16	4 000*M*4	Pulse Alarm *M* number of rolling seconds for bat pulse alarm period	
3/6/16	4 000*M*5	Pulse Alarm *M* minimum number of bat pulses per period to set alarm (high water)	
3/6/16	4 000*M*6	Pulse Alarm *M* maximum number of bat pulses per period to clear alarm (low water)	
3/6/16	4 0009*X*	Reserved for customization	
4	3 00001	Disk utilization (0-100%)	%
4	3 00002	Memory utilization (0-100%)	%
4	3 00003	CPU utilization (0-100%)	%
4	3 00004	Voltage	0.1 V
4	3 000*M*1	Pass Alarm *M* bat pass event counter (modulo 65536)	
4	3 000*M*2	Pass Alarm *M* bat pass events per period	
4	3 000*M*3	Pass Alarm *M* seconds since most recent pass event (up to 65535)	s
4	3 000*M*4	Pass Alarm *M* bat pass alarm counter (modulo 65536)	
4	3 000*M*5	Pulse Alarm *M* bat pulse event counter (modulo 65536)	
4	3 000*M*6	Pulse Alarm *M* bat pulses per period	
4	3 000*M*7	Pulse Alarm *M* seconds since most recent pulse event (up to 65535)	s
4	3 000*M*8	Pulse Alarm *M* bat pulse alarm counter (modulo 65536)	

Function Code (decimal)	Address (decimal)	Description (where *M* is a digit 1-8 for Alarm *M*) and *X* is any digit 0-9.	Unit
4	3 00*M*01	Pass Alarm *M* most recent value *N* (number of pulses in pass)	
4	3 00*M*02	Pass Alarm *M* most recent value *Fc* (characteristic frequency)	10 Hz
4	3 00*M*03	Pass Alarm *M* most recent value *Sc* (characteristic slope)	*octaves per second (OPS)*
4	3 00*M*04	Pass Alarm *M* most recent value *Fmax* (maximum frequency)	10 Hz
4	3 00*M*05	Pass Alarm *M* most recent value *Fmin* (minimum frequency)	10 Hz
4	3 00*M*06	Pass Alarm *M* most recent value *Fmean* (mean frequency)	10 Hz
4	3 00*M*07	Pass Alarm *M* most recent value *TBC* (time between calls)	0.1 ms
4	3 00*M*08	Pass Alarm *M*most recent value *Fk* (frequency of knee)	0.1 ms
4	3 00*M*09	Pass Alarm *M* most recent value *Tk* (time of knee)	0.1 ms
4	3 00*M*10	Pass Alarm *M* most recent value *S1* (initial slope)	*OPS*
4	3 00*M*11	Pass Alarm *M* most recent value *Tc* (time of characteristic)	0.1 ms
4	3 00*M*12	Pass Alarm *M* most recent value *Dur* (pulse duration)	0.1 ms
4	3 009*XX*	Reserved for customization	

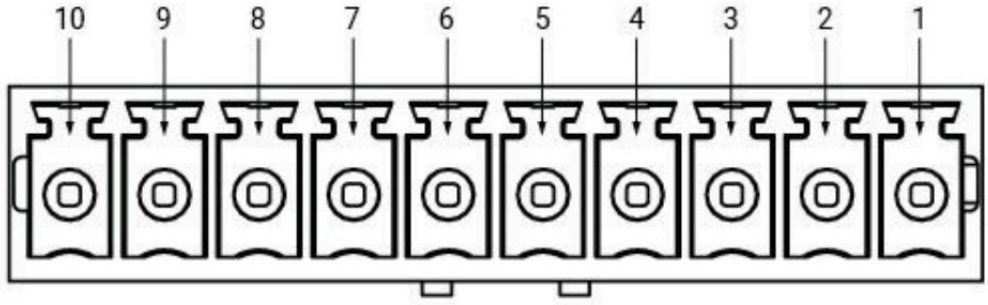
Control System Integration Using the Digital I/O Pins

The SMART Controller features a block of assignable, *Digital I/O (DIO)* pins. You can control these pins with a simple script to signal the states of up to four alarms in response to detected bat activity.

The DIO pins are located on the same face of the SMART Controller as the four USB ports. The SMART Controller ships with a 10-pin connector block with screw terminals you can use to make connections to these pins.

Digital I/O Pin Configuration

The DIO block consists of four input pins, four output pins, a ground pin, and a power input pin. Because these pins are optically isolated from the rest of the SMART Controller's motherboard, **you must supply 9-36 V DC power to Pin 1**, separate from the SMART Controller's main power supply.



Pin Number	Function
1	Power In (9-36 V DC)
2	Out 1
3	Out 2
4	Out 3
5	Out 4
6	In 1
7	In 2
8	In 3

Pin Number	Function
9	In 4
10	Ground

The four output pins, pins 2-5, can be used to signal the states of up to four alarms to an external SCADA interface.

Bat Activity Alarm Hook Script

Any time an alarm raises or lowers, the SMART System will execute a "hook" script if one exists with the file path `/var/www/storage/smart-scada-hook.sh`. You can use this script to perform any task, you would like, such as sending an email or updating a log file. It can also be used to set the state of the four DIO output pins.

The script `smart-scada-hook.sh` is invoked by the smart-scada service with the following arguments, in order:

$1 **- Alarm number**

Integer between 1 and 8.

$2 **- Alarm type**

String with the value "pass" or "pulse".

$3 **- Alarm status**

1 for active, 0 for inactive.

Example: Using a Hook Script to Signal Alarms via the DIO Pins

In this example, the script assigns the states of four alarms to the four DIO output pins:

Alarm	Pin Number	Output Number
Pulse Alarm 1	Pin 2	Out 1
Pulse Alarm 2	Pin 3	Out 2
Pass Alarm 1	Pin 4	Out 3
Pass Alarm 2	Pin 5	Out 4

```
#!/bin/bash
#
# smart-scada-hook.sh
#

#
# On Karbon 300, we have 4 DIO outputs OUT1, OUT2, OUT3, and OUT4.
# We make the following mappings here:
#
#   Pulse Alarm #1: OUT1
#   Pulse Alarm #2: OUT2
#   Pass Alarm #3:  OUT3
#   Pass Alarm #4:  OUT4
#

ALARM="$1"
TYPE="$2"
STATUS="$3"

if [[ $ALARM == "1" && $TYPE == "pulse"  ]]
then
```

```
  /usr/local/bin/karbon-300-controller set-do $STATUS"---"
elif [[ $ALARM == "2" && $TYPE == "pulse" ]]
then
  /usr/local/bin/karbon-300-controller set-do "-"$STATUS"--"
elif [[ $ALARM == "3" && $TYPE == "pass" ]]
then
  /usr/local/bin/karbon-300-controller set-do "--"$STATUS"-"
elif [[ $ALARM == "4" && $TYPE == "pass" ]]
then
  /usr/local/bin/karbon-300-controller set-do "---"$STATUS
fi
```

Serial Output

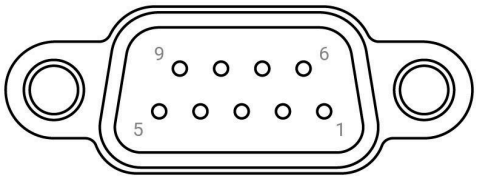
The SMART System can output system status information and pass-level measurements of bat call parameters via its two serial ports.

Serial Port Specifications

The SMART Controller has two male, DB-9 serial ports that support RS-232, RS-422, or RS-485 protocols. By default, both ports are configured for RS-232.

Both ports are *DTE* ports. To connect one of the SMART Controller's serial ports to another device with a DTE serial port, such as another computer, you should use a *null modem* female-to-female cross-over cable or adapter.

Figure 14. Serial Port Pin Arrangement



Pin	RS-232	RS-422	RS-485
1	DCD	TX-	TX-/RX-
2	RX	TX+	TX+/RX+
3	TX	RX+	NC
4	DTR	RX-	NC
5	GND	GND	GND
6	DSR	NC	NC
7	RTS	NC	NC
8	CTS	NC	NC
9	RI	NC	NC

NC = Not Connected

Serial Messages

Two types of serial messages can be generated:

Event Messages

Event messages are output each time a triggered recording is completed and the `results.csv` file is updated. These messages can contain an optional text string and any selected fields from the `results.csv` file, with the exception of the species ID field.

Poll Messages

Poll messages are output at regular intervals. These contain an optional text string and status information about the SMART System.

Enable Serial Output

To enable serial output of SMART System status and measured bat call parameters, you must select the fields to include in the output messages and configure several aspects of the output message format.

Procedure

1. Open the SMART Control Panel and go to **Serial Output**.
2. Configure the general output settings:
    a. Select the **Port** to use: ttyS0 or ttyS1.
    ttyS0 corresponds to the serial port nearest to the SMART Controller's power supply connection.
    b. Set the other output settings to match those used by the device reading the SMART Controller's serial output:
        - **Data Bits**
        - **Stop Bits**
        - **Parity**
        - **Baud Rate**
3. Configure Event messages:
    a. **Optional:** Enter a text string to prepend to each event message under **Prepend to Event Line**.
    b. Select the Event Fields to include in each event message.
    c. Edit the string with which to end each event message.
    The default ending string is `\r\n`, corresponding to a carriage return and line feed.
4. Configure poll messages:
    a. Use the **Polling Seconds** field to set the interval between poll messages.
    A value of 0 disables poll messages.
    b. **Optional:** Enter a text string to prepend to each poll message under **Prepend to Poll Line**.
    c. Select the Poll Fields to include with each poll message.
    d. Edit the string with which to end each poll message.
    The default ending string is `\r\n`, corresponding to a carriage return and line feed.
5. Select **Apply Changes**.

What to do next

If you need to use a serial protocol other than the default, RS-232, you can change the mode for each serial port via the SMART Controller's BIOS menu.

Change Serial Port Mode

The SMART Controller has two serial ports that support RS-232, RS-422, and RS-485. You can configure the active mode for each port using the SMART Controller BIOS.

About this task

The default mode for both serial ports is RS-232.

Procedure

1. Connect a monitor and keyboard to the SMART Controller.
2. Reboot the SMART Controller into the **Aptio Setup Utility**.
    a. Press the **Power** button to shut down the SMART Controller.
    b. Once the SMART Controller is powered off, press and hold the **Escape** key on your keyboard and press the **Power** button again. Continue holding **Escape** until the **Aptio Setup Utility** is displayed.

```
Aptio Setup Utility

Main  Advanced  Chipset  Security  Boot  Save & Exit
-------------------------------------------------------

BIOS Information

BIOS Vendor           American Megatrends
Core Version          5. 12
Compliancy            UEFI 2.5; PI: 1.4
BIOS Version          D8000A10.
Build Date            11/17/2020
Access Level          Administrator

Platform Firmware Information

BXT SOC               F1
MRC Version           0.56
PUNIT FW              1A
PMC FH                03.20
TXE FW                3. 1. 70.2325
ISH FW                4.1.0.3364
GOP                   10.0.1036
CPU Flavor            BXT Notebook/Desktop
Board ID              Oxbow Hill CRB (06)
Fab ID                FAB A

Memory Information

Total Memory          4096 MB
Memory Speed          1600 MHZ
```

3. Go to **Advanced > NCT5524OD Super IO Configuration**.
4. Change the **Serial Port Mode** for **Serial Port 1** (ttyS0) and/or **Serial Port 2** (ttyS1).
    - 1T/1R RS-422
    - 3T/5R RS-232 (Default)
    - 1T/1R RS-485 TX ENABLE Low Active
    - 1T/1R RS-422 with termination resistor

- 1T/1R RS-485 with termination resistor TX ENABLE Low Active
- Disable

5. Press **F4** and select **Yes** to save your changes and reboot the SMART Controller with the new settings.

Serial Port Event Message Format

When enabled, an event message is output from the SMART Controller's serial port whenever a triggered event is analyzed by an attached microphone and its call analysis parameters are saved to the microphone's `results.csv` file. The contents of the message depends on which fields are enabled, but it follows a consistent format.

Optional Leading String

Each event message may begin with a static string, entered into the **Prepend to Event Line** field on the **Serial Output** tab. By default, the optional string is empty.

Event Fields

After the optional leading string, the event message contains any selected fields from the most recent line in the `results.csv` file, separated by commas.

Abbreviation	Full Name	Format or Unit	Definition
Prefix	Microphone prefix		Prefix or name assigned to the microphone that recorded the event.
Date		*YYYY-MM-DD*	Date on which the event began.
Time		*hh:mm:ss.uuuuuu*, where *uuuuuu* are microseconds	Time at which the event began.
Duration		s	Duration of the event.
N	Number of Pulses		Number of detected bat pulses in the event. Other call parameters are averaged across *N* pulses in each event.
Fc	Average Characteristic Frequency	Hz	The frequency of the flattest point of each pulse, where *slope* has its absolute minimum value, averaged across *N* pulses.
Sc	Average Characteristic Slope	*OPS*	The minimum *slope* of each pulse, which defines the *call body*, averaged across *N* pulses.
Fmax	Average Maximum Frequency	Hz	The maximum frequency detected in each pulse, averaged across *N* pulses.

Abbreviation	Full Name	Format or Unit	Definition
Fmin	Average Minimum Frequency	Hz	The minimum frequency detected in each pulse, averaged across *N* pulses.
Fmean	Average Mean Frequency	Hz	The mean frequency of each pulse, averaged across *N* pulses.
TBC	Average Time Between Calls	s	If *N* > 1, the average period from the start of one pulse to the start of the next.
Fk	Average Knee Frequency	Hz	The frequency at the beginning of each pulse's *call body*, averaged across *N* pulses.
Tk	Average Time to Knee	s	The time from the beginning of each pulse to the *call body*, averaged across *N* pulses.
S1	Average Initial Slope	*OPS*	The *slope* from the beginning of each pulse to the *knee*, averaged across *N* pulses.
Tc	Average Time to Characteristic	s	The duration from the beginning of each pulse to the end of the *call body*, averaged across *N* pulses.
Dur	Average Pulse Duration	s	The full duration of each pulse, averaged across *N* pulses.
CRC7		two hexidecimal characters	Optional checksum.

Optional Terminating String

Each event message may end with a static string, entered into the **End Event Line With** field on the **Serial Output** tab. By default, this string is `\r\n`, corresponding to a Carriage Return followed by a Line Feed.

Example: Serial Port Event Message

In this example, the leading string for event messages was set to `EVENT`. The triggering sound source was a series of 40 kHz beeps emitted by a signal generator.

```
EVENT,9c.25.be.01.00.56,2024-11-05,13:24:51.530642,0.401,9,39903,0.05,40619,39802,4001
2,0.328976,40002,0.000350,166.17,0.048,0.048,31
```

Field	Example Value
Leading String	EVENT
Prefix	9c.25.be.01.00.56

Field	Example Value
Date	2024-11-05
Time	13:24:51.530642
Duration	0.401 s
N	9
Fc	39903 Hz
Sc	0.05 *OPS*
Fmax	40619 Hz
Fmin	39802 Hz
Fmean	40012 Hz
TBC	0.328976 s
Fk	40002 Hz
Tk	0.000350 s
S1	166.17 *OPS*
Tc	0.048 s
Dur	0.048 s
CRC7	31

Serial Port Poll Message Format

Poll messages are output at regular intervals. These contain an optional text string and status information about the SMART System.

Optional Leading String

Each poll message may begin with a static string, entered into the **Prepend to Poll Line** field on the **Serial Output** tab. By default, this string is empty.

Poll Fields

After the optional leading string, each poll message consists of any enabled status fields, separated by commas.

Name	Format or Unit	Definition
Date	*YYYY-MM-DD*	Date of the poll message.
Time	*hh:mm:ss*	Time of the poll message.
Disk Usage	%	How much of the SMART Controller's internal storage is in use.
Events		Number of event messages generated since the previous poll message.
Voltage	V	The SMART Controller's measured power supply voltage.
CRC7	two hexadecimal characters	Optional checksum.

Optional Terminating String

Each poll message may end with a static string, entered into the **End Poll Line With** field on the **Serial Output** tab. By default, this string is `\r\n`, corresponding to a Carriage Return followed by a Line Feed.

Example: Serial Port Poll Message

In this example, the prepending string is set to `POLL`.

```
POLL,2024-11-05,13:24:31,35.0,0,12.05,4F
```

Field Name	Example Value
Leading String	POLL
Date	2024-11-05
Time	13:24:31
Disk Usage	35.0 %
Events	0
Voltage	12.05 V
CRC7	4F

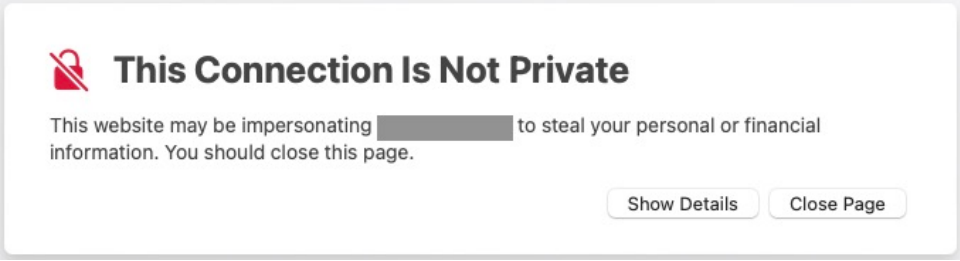
8. Troubleshooting

Privacy or Security Warning When Using SMART Control Panel

When you connect to the SMART Control Panel via a web browser, you may see a warning that the connection is not private.

The phrasing of this warning will depend on the web browser you are using.

Example: Security Warning in Safari



Cause: SSL Certificate Required

The SMART uses the secure HTTPS protocol to prevent traffic between the user and the SMART Control Panel from being monitored by third parties. HTTPS requires that the server (the SMART System in this case) provides an *SSL certificate* that the client (the user's computer) can use to establish a secure connection.

When you use a web browser to connect to a publicly available website, the website's servers provide an SSL certificate that has been generated by a widely known organization called a certificate authority. Your browser is built with the means to trust most major certificate authorities by default.

The default SSL certificate installed on the SMART system has not been verified by a certificate authority. In order for your web browser to automatically trust the connection to your SMART System, the SMART must have an SSL certificate that has been generated and verified by your own organization.

Remedy: Install a Signed SSL Certificate

Procedure

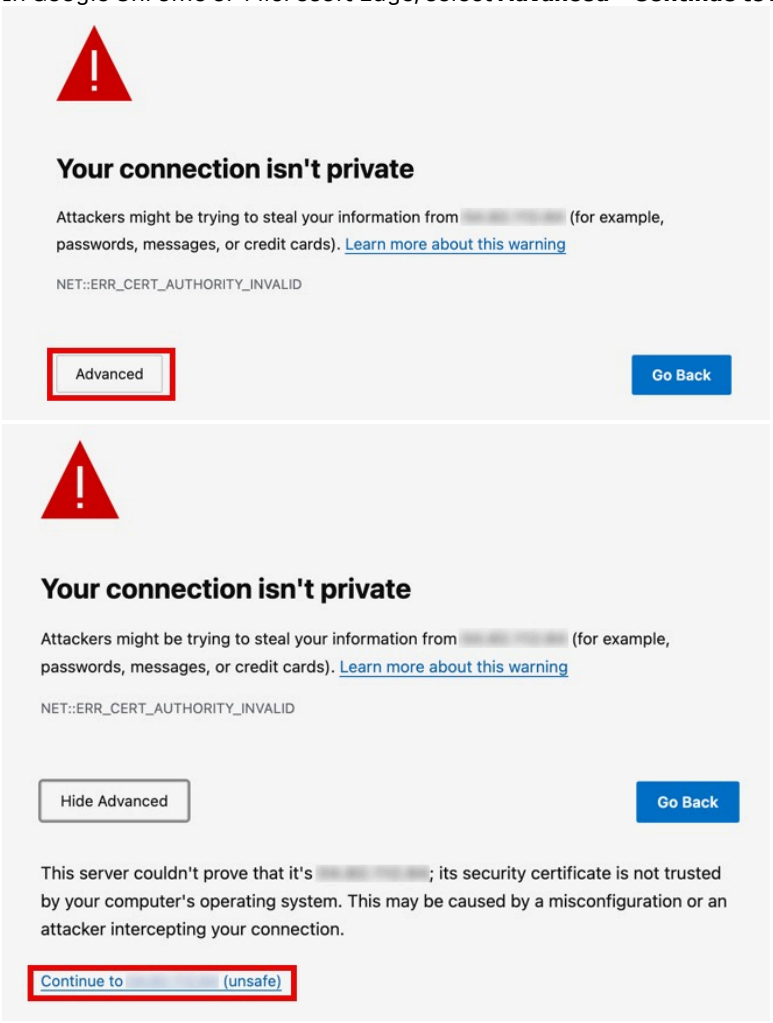
Ask your IT department to install an SSL certificate signed by your organization on your SMART System.

Remedy: Temporarily Suppress the Warning in Your Web Browser

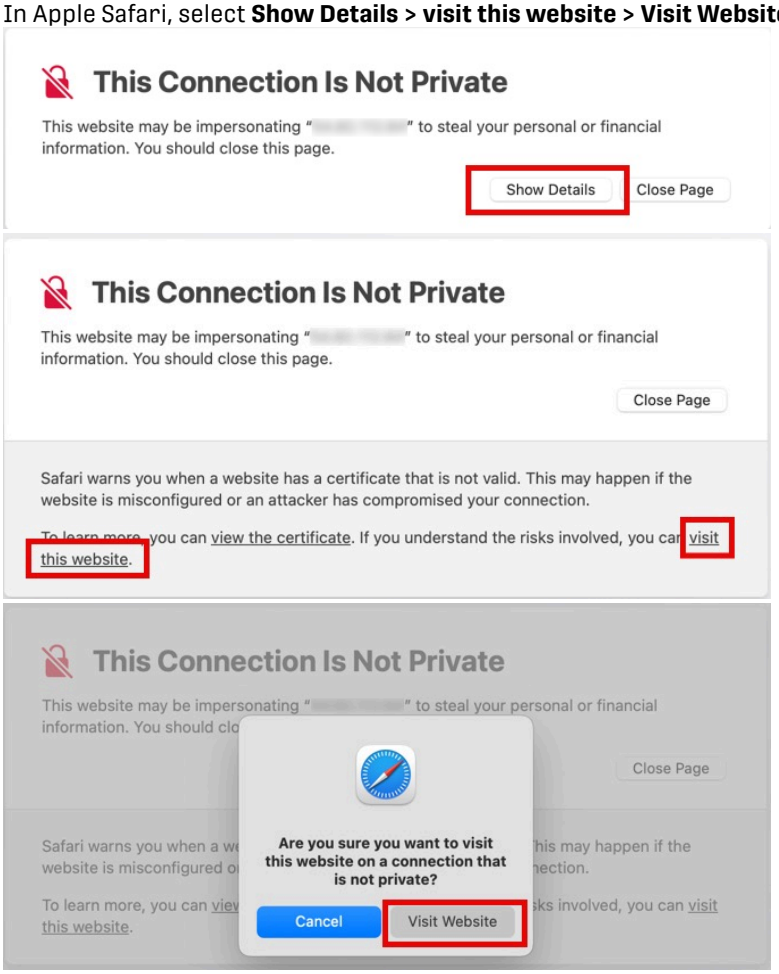
Procedure

You can temporarily bypass this warning to access the SMART Control Panel. You should only consider this a **temporary workaround** suitable only for setting up the SMART System for the first time. The secure and permanent solution to this warning is to install a custom SSL certificate on the SMART System.

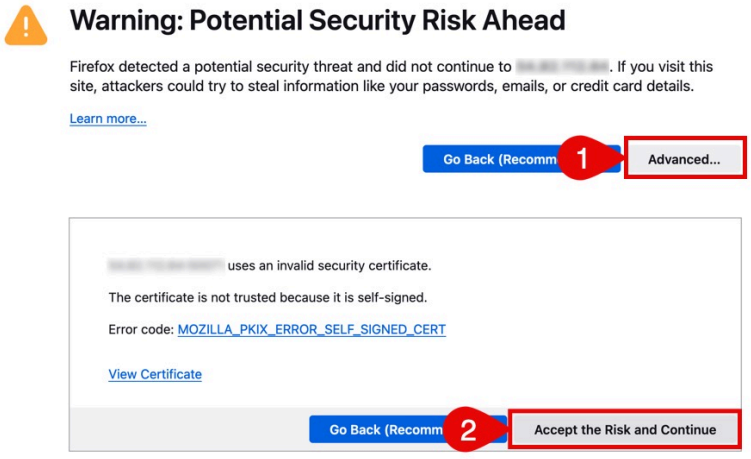
- In Google Chrome or Microsoft Edge, select **Advanced** > **Continue to *IP Address* (unsafe)**.



- In Apple Safari, select **Show Details > visit this website > Visit Website**



- In Mozilla Firefox, select **Advanced... > Accept the Risk and Continue**.



Related information

Install Custom SSL Certificates *(on page 53)*
Connect to the SMART Control Panel for the First Time *(on page 18)*
Access the SMART Control Panel via the SMART Portal *(on page 51)*

Unable to Use SSH via SMART IoT Gateway

You see an error message when you try to SSH to a SMART System that is connected to the SMART IoT Gateway.

Cause: Current IP Not Authorized

Your current IP address may not be authorized to access the SMART System.

Procedure

1. Go to https://www.wildlifeacoustics.com and go to **My Account > SMART Portal**.
2. Under **Control Panel URL**, check for the message *"current IP not authorized"*.
3. To authorize your current IP address, select **Change IP > Authorize My IP Address**.

Cause: Incorrect SSH Port

You may be attempting to use a port that is incorrect, or the port number may have changed since you last connected.

Procedure

1. Go to https://www.wildlifeacoustics.com and go to **My Account > SMART Portal**.
2. Note the **SSH Port** listed for your SMART System.
3. In your SSH client, specify the correct port number when you try to connect.

Related information

Access the SMART Command Line via the SMART IoT Gateway *(on page 52)*

SMART System Does Not Appear in SMART Portal

If your SMART System is not displayed in the SMART Portal or has not updated its status recently, the SMART may not be able to reach the SMART Portal, or you may not be an authorized user.

Condition

Your SMART System is not listed in the SMART Portal, or the last status received is from very long ago.

If the SMART System is listed in the SMART Portal for some users, but not others, proceed to #unique_164_Connect_42_toubleSolution_user_not_authorized *(on page 128)*.

Cause: User Not Authorized

Procedure

1. Open the SMART Control Panel and go to **Administration**.
2. Under **Wildlife Acoustics IOT Gateway**, enter the email addresses of authorized users, separated by commas or semicolons.
   Each email address must also be associated with an active account on https://www.wildlifeacoustics.com.
3. Confirm that **Enable** is checked.
4. Select **Apply Changes**.
5. On https://www.wildlifeacoustics.com, log into an account using one of the authorized email addresses, and go to **My Account > SMART Gateway**, or refresh the page if it is already open.

Cause: IoT Gateway Not Enabled

Procedure

1. Open the SMART Control Panel and go to **Administration**.
2. Under **Wildlife Acoustics IOT Gateway**, select **Enable**.
3. Enter the email addresses of authorized users, separated by commas or semicolons. Each email address must also be associated with an active account on https://www.wildlifeacoustics.com.
4. Select **Apply Changes**.

Cause: Connection Blocked by Firewall

If the SMART System is connected to a network with an active firewall, an IT administrator must configure the firewall to allow connections between the SMART System and the SMART IoT Gateway.

IT administrator

Procedure

In your firewall configuration, open the following ports to enable SMART IoT Gateway functionality:

Port Number	Function
21577	Allows users to access the SMART Control Panel via their account on https://www.wildlifeacoustics.com. Allows users to connect to the SMART's Ubuntu command line via ssh from an Internet-connected device.
21578	Allows the SMART System to send daily status updates and alert messages via email.

Cause: SMART Not Connected to the Internet

Your SMART System must be connected to a network with a means of accessing the global Internet.

Remedy: Reconnect Ethernet Connections

Procedure

1. If using Ethernet, confirm that an Ethernet cable is connected to **Ethernet Port 1**.
2. Confirm that **Ethernet Port 1** displays a solid green or amber LED, indicating an active connection.

Remedy: Correct Network Settings Using the SMART Control Panel

Procedure

If you are able to access the SMART Control Panel via the SMART Controller's Wi-Fi access point or a direct Ethernet connection, you can edit the network settings via the SMART Control Panel.

1. Open the SMART Control Panel and go to **Networking**.
2. Configure the **Ethernet Configuration** or **Wi-Fi Configuration** to connect to a network with a means of accessing the global Internet.

Remedy: Correct Network Settings via the Ubuntu Command Line

Procedure

1. Connect a keyboard and monitor to the SMART Controller.
2. Edit the file `/etc/netplan/01-netcfg.yaml` to configure the SMART Controller's network settings. Refer to the Netplan documentation for detailed instructions on creating and editing a YAML configuration file.
3. Apply the new configuration using the `netplan` command.

   ```
   sudo netplan apply
   ```

Not Receiving SMART Emails

If you are not receiving daily reports or alerts from your SMART System via email, there may be a connection issue with the SMART IoT Gateway, or email settings may not be correctly configured.

Cause: Email Alerts or Reports Not Configured

Procedure

1. Open the SMART Control Panel and go to **Maintenance**.
2. Under **Daily Email Report** or **Email Alerts**, check that at least one email address is entered as a recipient.
   Multiple addresses can be separated with commas or semicolons.
3. Select **Send Test Email** or **Send Test Alert Email**.
   All of the email addresses you entered should receive a test message from `noreply@wildlifeacoustics.com`.
4. Select **Apply Changes**, located at the top of the page.

Cause: IoT Gateway Not Enabled

Procedure

1. Open the SMART Control Panel and go to **Administration**.
2. Under **Wildlife Acoustics IOT Gateway**, select **Enable**.
3. Enter the email addresses of authorized users, separated by commas or semicolons.
   Each email address must also be associated with an active account on https://www.wildlifeacoustics.com.
4. Select **Apply Changes**.

Cause: Connection Blocked by Firewall

If the SMART System is connected to a network with an active firewall, an IT administrator must configure the firewall to allow connections between the SMART System and the SMART IoT Gateway.

IT administrator

Procedure

In your firewall configuration, open the following ports to enable SMART IoT Gateway functionality:

Port Number	Function
21577	Allows users to access the SMART Control Panel via their account on https://www.wildlifeacoustics.com. Allows users to connect to the SMART's Ubuntu command line via ssh from an Internet-connected device.
21578	Allows the SMART System to send daily status updates and alert messages via email.

Cause: SMART Not Connected to the Internet

Your SMART System must be connected to a network with a means of accessing the global Internet.

Remedy: Reconnect Ethernet Connections

Procedure

1. If using Ethernet, confirm that an Ethernet cable is connected to **Ethernet Port 1**.
2. Confirm that **Ethernet Port 1** displays a solid green or amber LED, indicating an active connection.

Remedy: Correct Network Settings Using the SMART Control Panel

Procedure

If you are able to access the SMART Control Panel via the SMART Controller's Wi-Fi access point or a direct Ethernet connection, you can edit the network settings via the SMART Control Panel.

1. Open the SMART Control Panel and go to **Networking**.
2. Configure the **Ethernet Configuration** or **Wi-Fi Configuration** to connect to a network with a means of accessing the global Internet.

Remedy: Correct Network Settings via the Ubuntu Command Line

Procedure

1. Connect a keyboard and monitor to the SMART Controller.
2. Edit the file `/etc/netplan/01-netcfg.yaml` to configure the SMART Controller's network settings. Refer to the Netplan documentation for detailed instructions on creating and editing a YAML configuration file.
3. Apply the new configuration using the `netplan` command.

```
sudo netplan apply
```

9. Specifications

SMART Controller Specifications

General Specifications

Attribute	Value	Notes
Hardware Model	OnLogic K300-E3930-4P-P	
Housing Dimensions	6.3 × 4.7 × 2.2 in. (160 × 119 × 56 mm)	
Weight	3.0 lb. (1.36 kg)	
Mounting Points	6, M3 × 0.5 mounting points, 4.0 mm maximum insertion depth	
Operating Temperature	-4°F to 158°F (-20°C to 70°C)	
USB Connectivity	4x USB 3.0 Type A	For external storage, keyboard
Ethernet Connectivity	1x GbE 2x GbE *PoE*	
Wi-Fi Connectivity	Dual-antenna, 802.11ac	
Cellular Connectivity	Option for 4G LTE	
Display Connectivity	2x DisplayPort	
Serial Connectivity	2x DB-9 Supports RS-232/422/485	
Power Connectivity	9-36 V DC 3-pin connector (Dinkle PN: 2ESDVM-03P)	
ProgrammableDigital I/O	• 4x In • 4x Out • Separate power and ground	8-bit, optically isolated
Grounding Features	1x Grounding Nut	
CAN Bus	3-pin CAN bus 2.0B	
Internal Storage	64 GB M.2 SATA SSD (approximately 48 GB free)	Options up to 1 TB
Processor	2-core Intel Atom x5-E930	Option for 4-core Intel Atom x7-E950
RAM	4 GB	Option for 8 GB

Power Requirements and Consumption

Table 6. SMART Controller Voltage Limits

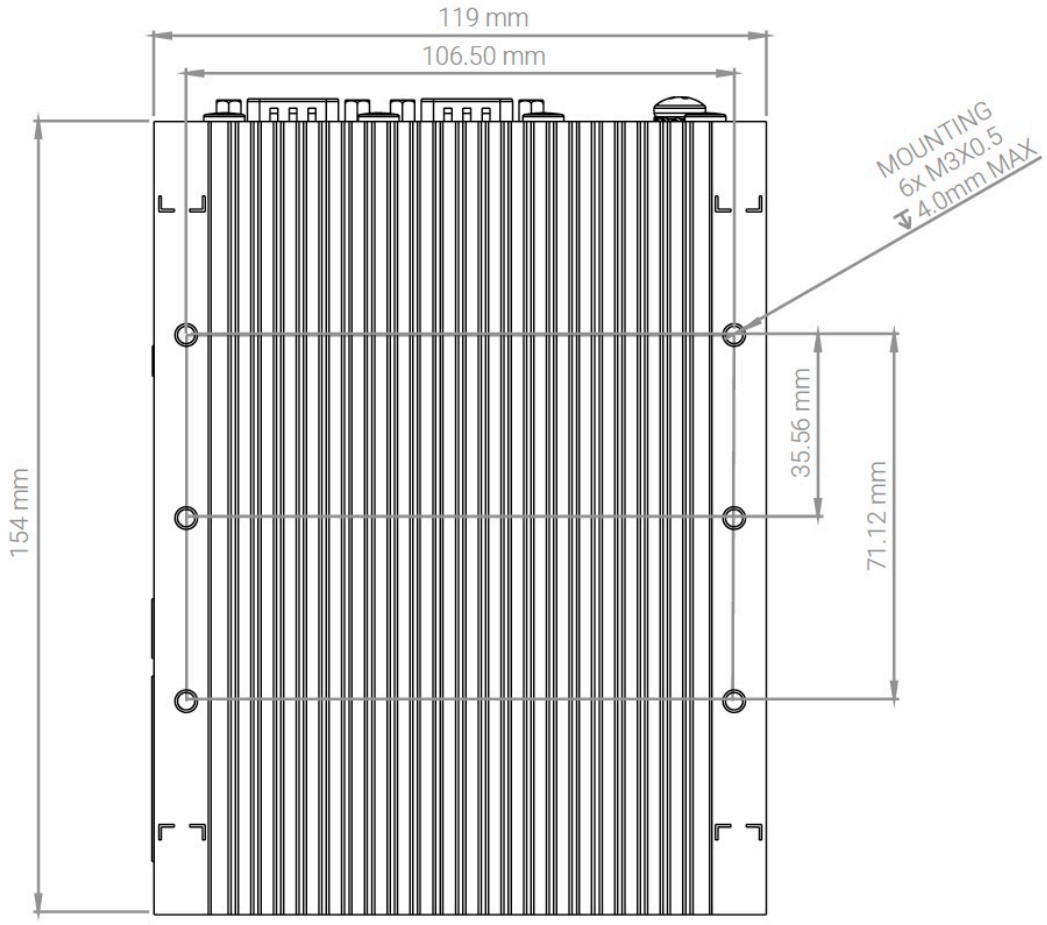
Parameter	Value
Nominal operating voltage (Rated DC value of input)	9~36 V
Undervoltage protection trip DC level (system turns off)	8.1 V
Overvoltage protection trip DC level (system turns off)	42.5 V
Maximum safe DC voltage (system not damaged)	50 V
Minimum safe reverse voltage (system not damaged)	-40 V
Ignition pin safe working voltage range (system not damaged)	-20~50 V

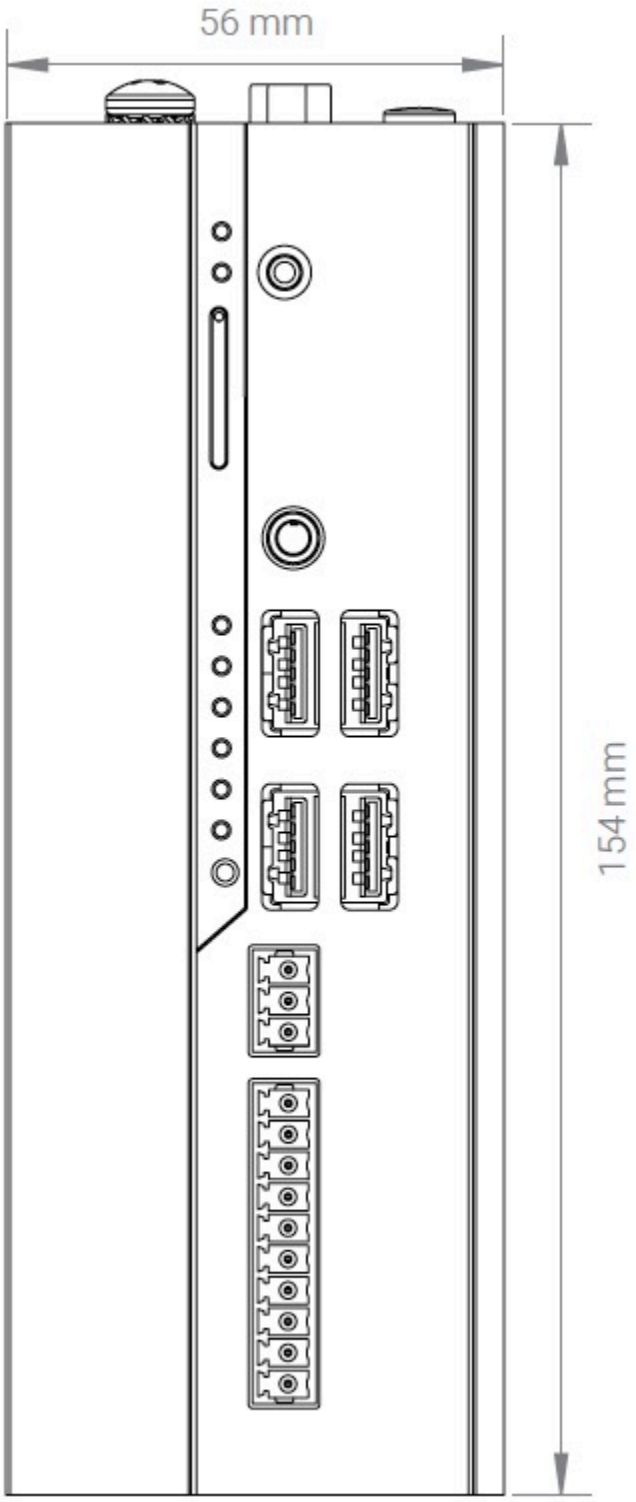
Table 7. Power Consumption by Microphone Count and Activity
The power consumption measurements shown here are approximate and will vary on factors such as attached peripherals, CPU load, and system configuration.

Activity	No Mics	One Mic	Two Mics	Three Mics
Powered down	0.6 W	0.6 W	0.6 W	0.6 W
Idle, powering mics via *PoE*	6.4 W	8.4 W	10.4 W	N/A
Analyzing and/or recording audio	N/A	9.5 W	12.2 W	14.5 W[1]
Streaming and/or recording, SMART MIC-1 heaters enabled	N/A	15.2 W	23.6 W	31.6 W[1]

1. Does not include power for necessary *PoE* switch.

SMART Controller Dimensional Drawings and Mounting Points




56 mm
154 mm

SMART MIC-1 Microphone Specifications

Physical Specifications

Attribute	Value
Length	8.9 in. (226 mm)
Diameter	3.0 in. (76 mm)
Weight	1.4 lb. (0.64 kg)
Operating Temperature	-4 °F to 185 °F (-20°C to 85°C)
Enclosure	IP67-rated weatherproof, anodized aluminum

Audio Specifications

Figure 15. Sensitivity and Noise of the SMART System with SMART MIC-1

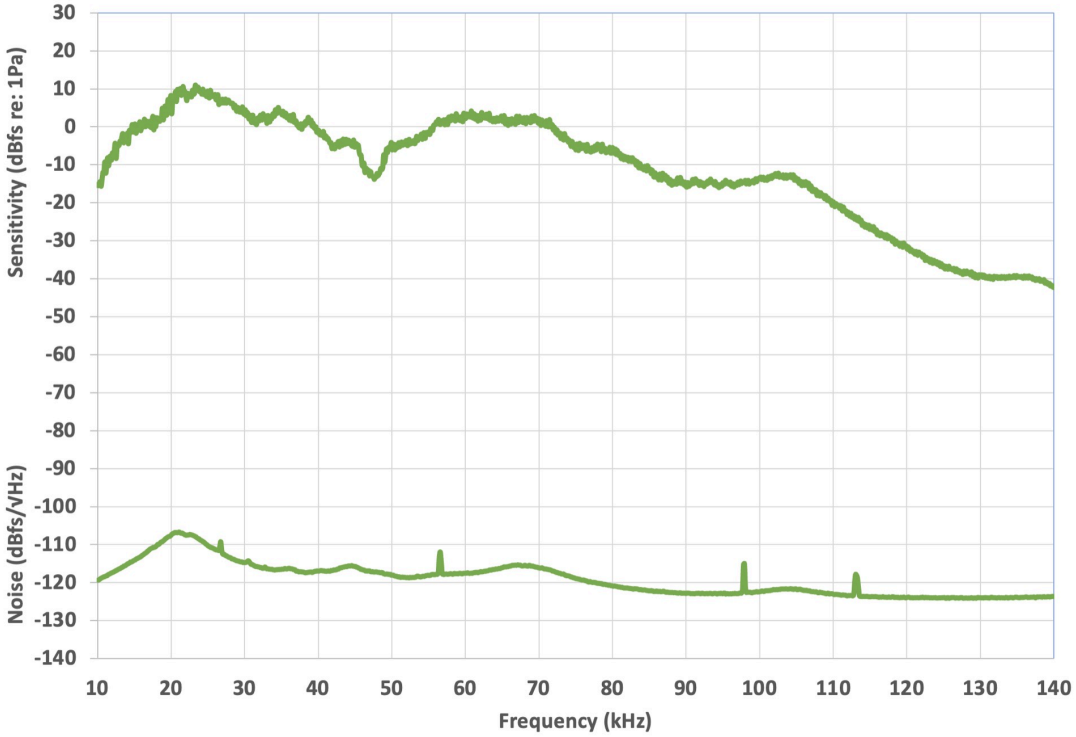
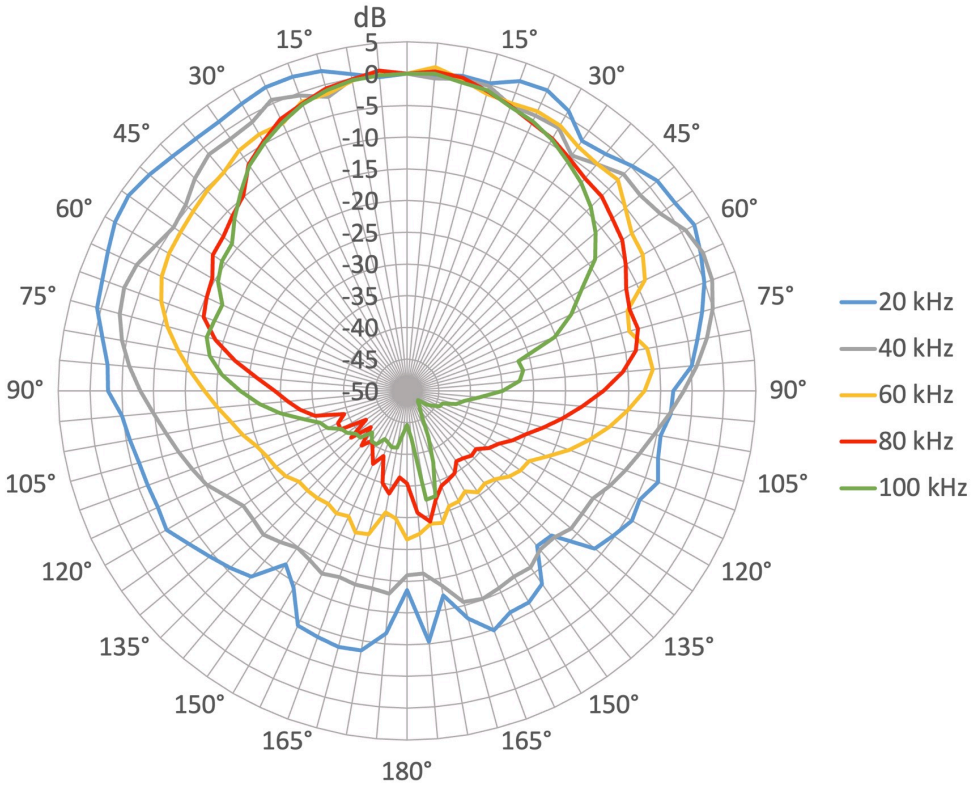


Figure 16. Directionality of the SMART System with SMART MIC-1



10. SMART API and Command Reference

The SMART System software includes utilities, command-line scripts, and APIs that can be used for additional customized programming of the SMART System.

You can read the same content found in this section from the SMART System command line using the `man` command. For example, `man smart-bat-sim` displays the same content as smart-bat-sim *(on page 138)*.

Executable Programs

smart-bat-sim

A Bat Alarm simulator for generating Pass and Pulse events

Synopsis

`smart-bat-sim [OPTIONS]...`

Description

smart-bat-sim is a test tool for generating Bat Pass & Pulse alarms. The simulator will generate either Pass or Pulse events and post them into the Kaleidoscope shared memory event queue. Once queued, the system will process the event using the defined filters from the SCADA configuration. Events are generated at a rate as dictated by the alarm-period, high-water-mark and low-water-mark. The high water rate is calculated as **(alarm-period / high water + 1)** - so with a high water of 20 events within a period of 60 seconds the sim generates an event every 2 seconds. Note that we add an extra event to ensure we meet the time period and account for any variances. The same formula is used to calculate the low water rate.

The sim will start generating events at the calculated rate to reach the high water mark. At that point, you should see the event '**raised**' in the smart-scada-log.txt file. For example:

2024-01-19,10:12:57,1,pulse,alarm1 pulse,raised

Once the number of events generated reaches the high water mark, the sim starts to generate events at the low water rate. Then, there is a final delay simply waiting enough time for the '**clear**' condition to be reached - in the logfile you should see: 2024-01-19,10:13:57,1,pulse,alarm1 pulse,cleared

This cycle is repeated for the desired number of iterations, where one iteration is a raise/clear combination.

Options

--alarm-period*=*alarm-period

The pass/pulse time period. Default 60 seconds

--high-water*=*high-water

The Pass/Pulse high water mark. Default 20 events

--iterations*=*iterations

The Number of iterations (raise/clear cycles) to run. Default 2 iterations

--low-water*=*low-water

The Pass/Pulse low water mark. Default 5 events

--type*=*type

The Type of event to generate {pass | pulse}. Default pulse

--verbose

Set this option for more verbose messages

-?, --help

Give this help list

--usage

Give a short usage message

--version

Print program version

Exit Codes

0

Success

1

Fail

Related information
smart-scada *(on page 144)*

smart-check-filter

check a SMART SCADA filter specification

Synopsis

`smart-check-filter` ***`string`***

Description

Check the syntax of the filter expression provided as a single command line argument. See the SMART documentation for more information.

Exit Codes

0

Expression is valid

1

Expression is not valid

Related information
smart-scada *(on page 144)*

smart-ctl

control SMART devices

Synopsis

`smart-ctl [` ***`sleep`*** `|` ***`wake`*** `|` ***`reset`*** `|` ***`upgrade`*** `]` ***`serial-number`***

Description

Request a given SMART device identified by the 6-octet hexadecimal serial-number change state to either sleep, wake, reset. or upgrade

Files

If upgrade is specified, the file /usr/local/share/smart/firmware_%d where %d is the device model number is sent to the device for firmware upgrade. This file is typically a symbolic link to an actual version-specific file.

Related information
smart-stream *(on page 150)*
smart-daemon *(on page 140)*
smart-list *(on page 142)*

smart-daemon

daemon process to interface with SMART devices

Synopsis

`smart-daemon` [**`--force`**] [**`-i ifname`**]

Description

Binds to the specified Ethernet interface (or the first suitable interface if ifname is not specified) and begins communication with SMART devices over Ethernet. A control socket is created to interface between the daemon and user processes by means of the libSMART shared library.

The SMART daemon must be run as root. A valid license is required to run the SMART daemon and to use the libSMART libraries.

Options

--force

is used to force the SMART daemon to run even if a PID file already exists.

-i interface

specifies the Ethernet device interface to use. Otherwise, the first suitable interface found is used automatically.

Files

/var/run/smart/smart_pid

A file containing the process id of the currently running SMART daemon.

/var/run/smart/smart_ctl

A UNIX socket used by libSMART to communicate with the SMART dameon.

/usr/local/share/smart/smart.lic

A license file required for using the SMART daemon and libSMART.

Related information
smart-stream *(on page 150)*
SMART_Close *(on page 155)*
SMART_CloseWav *(on page 156)*
SMART_CreateWav *(on page 157)*
SMART_GetDeviceInfo, SMART_GetDeviceInfoBySN, SMART_GetNDevices *(on page 158)*

smart-dispatcher

SMART Microphone Scheduler

Synopsis

`smart-dispatcher`

Description

smart-dispatcher runs as a service and schedules attached microphone devices for streaming. For each scheduled microphone, **smart-dispatcher** spawns an instance of **smart-stream-wrapper**.

The **smart-dispatcher** scans the devices directory /var/www/html/storage/devices/ for device-specific subdirectories named with their unique serial number (six dotted hex bytes). Within each subdirectory, the dispatcher looks for a settings and schedule configuration file. If both files are present, the microphone device is scheduled in accordance with the schedule configuration file. When scheduled to record, the **smart-stream-wrapper** is invoked with the --duration parameter set in accordance to the scheduled end time, and other parameters are passed in accordance to the settings configuration file.

The settings configuration file is a list of arguments to be passed to **smart-stream-wrapper**, each argument on a line.

The schedule configuration file is a text file comprising a list of schedule blocks. Each schedule block is three lines as follows:

START { **TIME** | **RISE** {+|-} | **SET** {+|-} } hh:mm

DUTY { **ALWAYS** | **CYCLE ON** hh:mm **OFF** hh:mm }

END { **TIME** | **RISE** {+|-} | **SET** {+|-} } hh:mm

Exit Codes

0

Success

1

Fail

Files

/var/www/html/storage/devices/ <*serial-number*> **/schedule**

Per-microphone schedule configuration

Related information
smart-daemon *(on page 140)*
smart-stream-wrapper *(on page 154)*
smart-stream *(on page 150)*

smart-list

list SMART devices

Synopsis

```
smart-list
```

Description

Lists detected SMART devices and their capabilities.

Related information
smart-stream *(on page 150)*
smart-daemon *(on page 140)*

smart-logger

Periodically log system information for SMART services

Synopsis

```
smart-logger [  -i <seconds>  ]
```

Description

The SMART Logger runs as a service and periodically monitors disk, CPU, and memory utilization as well as battery voltage levels for other SMART services to use (e.g. for serial outputs, SCADA/Modbus, etc). The default interval is 10 seconds unless otherwise specified on the command line.

For battery voltage, a bash shell script /usr/local/bin/SaveVoltage.sh is executed. This script should write the battery voltage, in volts, to the file /tmp/LastVoltage. The value from this file is then read by the smart-logger service or could be used by other applications.

Files

/usr/local/bin/SaveVoltage.sh as described above

/tmp/LastVoltage as described above

Related information
smart-serial *(on page 148)*
smart-scada *(on page 144)*

smart-modbus-probe

Read or Write local Modbus registers via smart-scada.

Synopsis

```
smart-modbus-probe  modbus-address  [  count  [  value, ...  ] ]
```

Description

Read or write local Modbus registers. See the SMART documentation for the Modbus register map which includes registers managed by the smart-scada service as well as registers available for custom applications.

Parameters

modbus-address

6-digit decimal Modbus register starting address

count

If specified, the number of sequential registers to read and write.

values...

If specified, the number of values must match count. Values are 16-bit signed or unsigned values to be written to sequential Modbus registers beginning with the modbus-address. If not specified, the specified registers are read and printed as signed 16-bit values, one per line.

Exit Codes

0

Success

1

Fail

Related information
smart-scada *(on page 144)*

smart-opcua

SMART OPC UA Service

Synopsis

```
smart-opcua
```

Description

smart-opcua runs as a service and provides a secure mechanism for retrieving status information, system updates and alarm notifications from a SMART device using OPC UA communications. OPC UA clients are able to take appropriate actions according to local needs, including bat curtailment processing. The Information Model used by the SMART Controller provides a flexible and comprehensive set of parameters, methods and publish/subscribe features to enable full monitoring of a SMART system. The parameters exposed by smart-opcua are mirrored from the Modbus registers available from the smart-scada service.

smart-opcua will only start if the file /var/www/storage/scada.json exists. This file is created when SCADA has been configured from the SMART Control Panel.

Options

Startup options can be modified in the **/etc/defaults/smart-opcua.opts** file. Simply edit the file and add the option(s) needed to the SMART_OPCUA_OPTS setting.

--port=port number

The TCP port for smnart-opcua to listen. Default 4840

--certificate=certificate file

Full path to the X.509 certificate .der file. Default /etc/ssl/private/server_cert.der

--key=private key file

Full path to the private key .pem file. Default /etc/ssl/private/server_key.pem

--secure-channel-trust-folder=channel trust folder

Full path to the secure channel trust certs. Default /etc/ssl/private/trust

--session-trust-folder=sesion trust folder

Full path to the secure channel trust certs. Default /etc/ssl/private/trust

Exit Codes

0

Success

1

Fail

Files

/var/www/html/storage/scada.json

The SCADA Configuration file

/etc/defaults/smart-opcua.opts

Configuration file used to override smart-opcua startup options

/usr/local/share/smart/opcua/smartModel.xml

The SMART Information Model defiinition

/usr/local/share/smart/opcua/smartModel.csv

The Information Model node ID definitions

/usr/local/share/smart/opcua/Smart.NodeSet2.xml

The compiled smartModel.xml into a NodeSet2 format ingested by the smart-opcua server.

Related information
smart-scada *(on page 144)*

smart-scada

SMART SCADA/Modbus Service

Synopsis

```
smart-scada
```

Description

smart-scada runs as a service and responds to Modbus requests over either Modbus/TCP (over TCP/IP networks) or Modbus/RTU (over serial interfaces). Real-time bat pulse/pass information is collected from the SMART event infrastructure as well as system status information from the SMART logger to update Modbus registers accordingly.

The SMART SCADA system can be configured to use either Modbus/TCP or Modbus/RTU. For Modbus/RTU, additional serial parameters must be specified. Up to eight (8) filters can be defined (numbered one (1) through eight (8)). Each filter can be applied to individual bat echolocation pulses or entire bat passes as they are reported to the SMART event system.

The scada.json configuration file is a JSON formatted file with the following structure:

modbusPort

String indicating either "TCP" for Modbus/TCP or a TTY e.g. "ttyS0" for Modbus/RTU. An empty string indicates that the SMART SCADA service is disabled.

modbusSlaveId

For RTU protocol, this is the server ID (numeric)

modbusBaudRate

For RTU protocol, this is the baud rate (numeric)

modbusParity

For RTU protocol, this string indicates parity as one of "None", "Even", or "Odd".

modbusDataBits

For RTU protocol, this is the number of data bits 5, 6, 7, or 8.

modbusStopBits

For RTU protocol, this is the number of stop bits 1 or 2.

filters

An array of active filters:

filters[N].filterNumber

The filter nubmer one (1) through eight (8) defined by this array element.

filters[N].eventPeriod

Integer number of rolling seconds for bat pass rate monitoring.

filters[N].eventHighWater

Integer bat pass rate threshold triggering the filter bat pass alarm.

filters[N].eventLowWater

Integer bat pass rate threshold clearing the filter bat pass alarm.

filters[N].pulsePeriod

Integer number of rolling seconds for bat pulse rate monitoring.

filters[N].pulseHighWater

Integer bat pulse rate threshold triggering the filter pulse pass alarm.

filters[N].pulseLowWater

Integer bat pulse rate threshold clearing the filter pulse pass alarm.

filters[N].spec

Filter specificiation string (see below)

Filter Specification

The filter specificiation is a boolean expression that can match bat call parameters (and microphone device names as defined by their "prefix"). As individual echolocation calls match a filter expression, the bat pulse period, high-water mark and low-water mark are used to raise or clear a per-filter bat pulse alarm. Similarly, as sequences of echolcoation calls forming a bat pass (as defined by smart-stream triggering parameters match a filter expression, the bat pass period, high-water mark and low-water mark are used to raise or clear a per-filter bat pass alarm.

An expression can be a numeric comparison between floating point values. Values can be specified as floating point literals e.g. -1.23 or any of the bat call parameters **N**, **Fc**, **Sc**, **Fmax**, **Fmin**, **Fmean**, **TBC**, **Fk**, **Tk**, **S1**, **Tc Dur**, or **Qual**. Comparison operators can be = for equality, **!=** or **<>** for not equal, **<** for less than, **<=** for less than or equal, **>** for greater than, or **>=** for greater than or equal.

A floating point value can also be calculated using addition **+** or subtraction **-** between floating point literals and bat call parameters.

In order to differentiate among multiple microphone devices, the variable **prefix** can be compared to a string literal using **=** for equality, **!=** or **<>** for not equal, or **~** if the left hand side contains the right hand side as a substring. String literals are text surrounded by matching single or double quotes.

For multi-channel microphones, the variable **channel** can be compared to a numeric channel number. Note that at this time, the only SMART microphone SMART-MIC-1 has only one channel (0).

Boolean expressions can be combined using the operators **AND**, **OR** or inverted with **NOT**.

Nested parentheses can be used as per normal convention.

A comment can be indicated with any text following the **#** character.

Note that white space is ignored, filter expressions can span multiple lines, and variable names are case insensitive.

Modbus Register Map

The following table describes the Modbus register map. Registers marked as reserved for customization can be read or written using smart-modbus-probe.

Function	Address	Description
2	10000**N**	Filter #**N** bat pass alarm
2	10001**N**	Filter #**N** bat pulse alarm
2	10002**X**	Reserved for customization
3/6/16	4000**N**1	Filter #**N** bat pass alarm period
3/6/16	4000**N**2	Filter #**N** bat pass high-water mark
3/6/16	4000**N**3	Filter #**N** bat pass low-water mark
3/6/16	4000**N**4	Filter #**N** bat pulse alarm period
3/6/16	4000**N**5	Filter #**N** bat pulse high-water mark
3/6/16	4000**N**6	Filter #**N** bat pulse low-water mark
3/6/16	40009**X**	Reserved for customization
4	300001	Disk utilization (%)
4	300002	Memory utilization (%)
4	300003	CPU utilization (%)
4	300004	Battery voltage (0.1V)
4	3000**N**1	Filter #**N** bat pass event counter
4	3000**N**2	Filter #**N** bat pass events per second

Function	Address	Description
4	3000**N**3	Filter #**N** bat pass time since last event (s)
4	3000**N**4	Filter #**N** bat pass alarm counter
4	3000**N**5	Filter #**N** bat pulse event counter
4	3000**N**6	Filter #**N** bat pulse events per second
4	3000**N**7	Filter #**N** bat pulse time since last event (s)
4	3000**N**8	Filter #**N** bat pulse alarm counter
4	300**N**01	Filter #**N** bat pass most recent N
4	300**N**02	Filter #**N** bat pass most recent Fc
4	300**N**03	Filter #**N** bat pass most recent Sc
4	300**N**04	Filter #**N** bat pass most recent Fmax
4	300**N**05	Filter #**N** bat pass most recent Fmin
4	300**N**06	Filter #**N** bat pass most recent Fmean
4	300**N**07	Filter #**N** bat pass most recent TBC
4	300**N**08	Filter #**N** bat pass most recent Fk
4	300**N**09	Filter #**N** bat pass most recent Tk
4	300**N**10	Filter #**N** bat pass most recent S1
4	300**N**11	Filter #**N** bat pass most recent Tc
4	300**N**12	Filter #**N** bat pass most recent Dur
4	300**N**13	Filter #**N** bat pass most recent Qual
4	3009**XX**	Reserved for customization

Event Hook

If a file smart-scada-hook.sh exists, it will be executed as a bash script when an alarm condition changes. The first argument is the alarm number (1-8). The second argument is the alarm type "pass" or "pulse", and the third argument is 1 to indicate the alarm was activated or 0 to indicate the alarm was cleared.

Exit Codes

0

Success

1

Fail

Files

/var/www/html/storage/scada.json

Configuration file

/var/www/html/storage/smart-scada-hook.sh

Hook file

Related information
smart-modbus-probe *(on page 142)*
smart-logger *(on page 142)*
smart-serial *(on page 148)*

smart-serial

SMART Serial Service

Synopsis

```
smart-serial
```

Description

smart-serial runs as a service and writes bat pass events and periodic status data to the specified serial port as events are collected from the SMART event infrastructure.

The serial.conf configuration file contains 8 lines with the following structure:

1. TTY

The serial port to use e.g. "ttyS0".

2. {7|8}{N|E|O}{1|2}<baudrate>

Specifies the data bits, parity, and baudrate e.g. "8N19600" indicates 8 data bits, no parity, 1 stop bit, and 9600 baud.

3. mask

The mask is a 32-bit hexadecimal number indicating which output fields to include lines written to the serial port. See below for details.

4. event line preamble

Characters to send at each event line start. \r and \n can be used to indicate carriage-return and new-line characters respectively.

5. event line epilogue

Characters to send at each event line end including line termination. \r and \n can be used to indicate carriage-return and new-line characters respectively.

6. polling interval

Polling interval for output of periodic status information in seconds, or zero to disable periodic status information output. **7. status line preamble** Characters to send at each status line start. \r and \n can be used to indicate carriage-return and new-line characters respectively.

8. status line epilogue

Characters to send at each status line end including line termination. \r and \n can be used to indicate carriage-return and new-line characters respectively.

The following table describes the mask bits (hex values or'ed together):

Mask Bits	Description
0x00000001	Prefix
0x00000002	Date
0x00000004	Time
0x00000008	Duration
0x00000010	N
0x00000020	Fc
0x00000040	Sc
0x00000080	Fmax
0x00000100	Fmin
0x00000200	Fmean
0x00000400	TBC
0x00000800	Fk
0x00001000	Tk
0x00002000	S1
0x00004000	Tc
0x00008000	Dur
0x00010000	nPulsesClassified
0x00020000	nPulsesMatching
0x00040000	Top1Match
0x00080000	Top1Margin
0x00100000	Top2Match
0x00200000	Top2Margin
0x00400000	Top3Margin
0x00800000	Top3Margin
0x01000000	Event generated CRC7
0x02000000	Status Date
0x04000000	Status Time
0x08000000	Status Disk Utilization (%)
0x10000000	Status Events since last poll
0x20000000	Voltage
0x40000000	Reserved
0x80000000	Status generated CRC7

Exit Codes

0

Success

1

Fail

Files

/var/www/html/storage/serial.conf

Configuration file

Related information
smart-modbus-probe *(on page 142)*
smart-logger *(on page 142)*
smart-serial *(on page 148)*

smart-stream

stream and process audio from SMART device microphone device

Synopsis

`smart-stream [REQUIRED PARAMETER]... [OPTION]...`

Description

Opens an audio stream from a SMART device and optionally creates audio files and/or analyzes the data.

Required Parameters

--microphone serial number

Device serial number, hexadecimal with 6 octets (12 hex digits)

--sample -rate sample rate

Sample rate in Hz matching device capabilites

Run options

--duration duration

Specify the duration to run the stream in seconds, or in mm:ss or hh:mm:ss format. After this duration, the program exits. If not specified, the program will run indefinitely until the connection to the device is lost.

Device options

--channel channel number

Specify the channel number 0 - 7. The default channel is 0 if not specified.

--high-pass high pass filter

High-pass filter in Hz matching device capabilities.

--gain gain

Gain in dB matching device capabilities.

--use-backup-sensor

Specify using backup sensor if available in device capabilities. Otherwise the primary sensor will be used.

--enable-heater

Specify using defogging heater if available in device capabilities. Otherwise the heater will be turned off.

--no-heater-analysis

If used with ***--enable-heater*** above, the data stream will not be analyzed. Specifically, this option overrides and disables ***--bats*** and ***--non-bats*** and there will be no bat triggers or events.

--calibrate frequency amplitude

Generate a calibration signal at the specified frequency (Hz) and amplitude (1.0 full scale). As the microphone signal is returned, the dB level of the calibration signal (narrow-band filtered) is displayed on exit.

--stdout path

Append standard output (csv results) to specified file.

--stderr path

Append standard error (logging) to specified file.

File creation options

Audio files representing each event (if analysis is enabled) or the audio stream for the specified duration can be created. The filename is of the form

prefix_YYYYMMDD_hhmmss_uuuuuu.ext

where .ext is either .wav, .w4v or .zc.

--output-wav directory

If specified, timestamped .wav or .w4v files will be created in the directory specified.

--output-zc directory

If specified, timestamped .zc files will be created in the directory specified. This option requires either ***--bats*** or ***--non-bats***.

--compression compression

If ***--output-wav*** is specified, compression values 4, 6 or 8 can be used to specify Wildlife Acoustics .w4v format with the specified numer of bits per sample. The default is 0 which indicates no compression and creation of standard .wav files.

--prefix prefix

Specify a prefix string that is pre-pended to the filename.

--pre-trigger pre-trigger

Specify the amount of extra time in seconds to include in the output recording prior to the first detected signal.

--post-trigger post-trigger

Specify the amount of extra time in seconds to include in the output recording after the last detected signal (this does not apply to .zc files.

--location latitude longitude

Specify the latitude and longitude to include in metadata.

--guano string

Specify additional lines of GUANO metadata key:value pairs.

Kaleidoscope Analysis Options

--bats or ***--non-bats***

Specify bat analysis mode or non-bat analysis mode. Only one mode can be enabled. One of these options is required to enable analysis.

--min-freq min frequency

Specify the minimum frequency of the expected signal in Hz. The default is 8000 Hz if ***--bats*** is specified and 250 Hz if ***--non-bats*** is specified.

--max-freq max frequency

Specify the maximum frequency of the expected signal in Hz. The default is 120000 Hz if ***--bats*** is specified and 10000 Hz if ***--non-bats*** is specified.

--min-dur min duration

Specify the minimum pulse duration (if ***--bats*** is specified) or detection duration (if ***--non-bats*** is specified) in seconds. The default is 0.002 if ***--bats*** is specified or 0.100 if ***--non-bats*** is specified.

--max-dur max duration

Specify the maximum pulse duration (if ***--bats*** is specified) or detection duration (if ***--non-bats*** is specified) in seconds. The default is 0.500 if ***--bats*** is specified or 7.500 if ***--non-bats*** is specified.

--max-gap max gap

Specify the maximum inter-pulse gap (if ***--bats*** is specified) or inter-detection gap (if ***--non-bats*** is specified) in seconds. The default is 0.500 if ***--bats*** is specified or 0.350 if ***--non-bats*** is specified.

--max-sequence max sequence

Specify the maximum detection event duration (if ***--bats*** is specified) in seconds. The default is 15.0 seconds.

--min-pulses min pulses

Specify the minimum number of pulses required to consider a signal a bat (if ***--bats*** is specified). The default is 2.

--cf-filt-max-freq constant frequency filter max frequency

Specify the maximum frequency in Hz for the constant frequency filter. The default is off (0).

--cf-filt-max-bw constant frequency filter max bandwidth

Specify the maximum bandwidth in Hz for the constant frequency filter. The default is off (0).

--disable-enhanced-zc

Disable the enhanced zero-crossing algorithm used by Kaleidoscope. Instead of more sophisticated narrow-band analysis, the signal will be band-pass filtered by the min frequency and max frequency and then undergo zero-cross analysis.

--classifier classifier

A classifier can be specified. This can be either a cluster .kcs file or a directory containing the unzipped files from a Bat Auto ID classifier.

--threshold threshold

A threshold can be specified to adjust the sensitivity of the classifier. For bat auto id classifiers, this value is either -1 for "more sensitive", 0 for "balanced" or 1 for "more accurate". For cluster analysis, this is the maximum distance, a floating point value between 0 and 2. The default is 1.

--species species list

A comma separated list of species codes to include in the classifier. If not specified, all classifier species codes are included.

Outputs

If ***--calibrate*** is specified, the dB level (re 0 dB = full scale) is output in a one-line message narrow-band filtered on the given frequency.

If ***--output-wav*** is specified, timestamped .wav or .w4v files will be generated. If ***--output-zc*** is specified, timestamped .zc files will be specified.

If analysis mode is enabled (e.g. with either ***--bats*** or ***--non-bats***), a line of text will be output for each detection. Each line has a list of comma separated text fields as follows:

YYYY-MM-DD

Date representing the beginning of the event (not adjusted for pre-trigger)

hh:mm:ss.uuuuuu

Time representing the beginning of the event (not adjusted for pre-trigger)

Duration

Duration of detected event in seconds

N

For ***--bats***, number of pulses detected. The following statistics are measurements averaged over these pulses.

Fc

For ***--bats***, characteristic frequency, Hz

Sc

For ***--bats***, characteristic slope, octaves per second

Fmax

Maximum frequency, Hz

Fmin

Minimum frequency, Hz

Fmean

Mean frequency, Hz

TBC

For ***--bats***, time between calls, seconds

Fk

For ***--bats***, frequency of the knee, Hz

Tk

For ***--bats***, time of the knee, seconds

S1

For ***--bats***, initial slope, octaves per second

Tc

For ***--bats***, time of characteristic, seconds

Dur

For ***--bats***, pulse duration, seconds

nPulsesClassified

For ***--bats***, number of pulses classified

nPulsesMatching

For ***--bats***, number of pulses matching final classification

Top1Match

First matching classification label

Top1Margin

First matching classification margin

Top2Match

Second matching classification label

Top2Margin

Second matching classification margin

Top3Match

Third matching classification label

Top3Margin

Third matching classification margin

Related information
smart-daemon *(on page 140)*
smart-stream-wrapper *(on page 154)*

smart-stream-wrapper

A wrapper for smart-stream for scheduling

Synopsis

```
smart-stream-wrapper [OPTIONS]...
```

Description

smart-stream-wrapper adds a maintenance schedule layer around smart-stream to manage automatic multi-sensor calibration, selection, and schedule a duty-cycle for activating a built-in heater element. smart-stream-wrapper will invoke the smart-stream process sequentially to conform with the maintenance schedule based on the additional maintenance options described below until terminated by SIGINT or error. Other options are passed to the smart-stream verbatim with the exception of ***--duration***. If ***--duration*** is specified, this is the maximum value passed to the underlying smart-stream process. The smart-stream-wrapper may use shorter values to conform to the maintenance schedule.

The smart-stream-wrapper will selectively pass ***--enable-heater***, and ***--use-backup-sensor*** in accordance with the maintenance schedule. For calibration, smart-stream-wrapper will invoke smart-stream using ***--calibrate*** to measure the response of the two redundant sensors and automatically choose the best one.

Options

--schedule-heater on-duration off-duration

Specify the on-duration and off-duration duty-cycle for the heater in seconds. The --enable-heater parameter will be passed to the underlying smart-stream program during the on periods.

--schedule-calibrate freq amp target period

A calibration will run every period seconds to calibrate and measure the primary sensor (for one second) and the backup sensor (for another second) by invoking smart-stream with the --calibrate option. The frequency in Hz and the amplitude as a fraction of full scale is passed to smart-stream. The sensor measuring closest to the target value in dB relative to full scale will be chosen. Subsequent invocations of smart-stream will optionally use --use-backup-sensor.

--stdout path

Append standard output (csv results) to specified file.

--stderr path

Append standard error (logging) to specified file.

Hook

When smart-stream-wrapper performs a calibration sequence, a user-defined bash shell script may be executed to respond to microphone communciation failures and calibration measurements. If the file /var/www/html/storage/smart-calibrate-hook.sh is present, it will be executed with the following arguments:

serial-number ($1)

Device serial number.

primary result ($2)

Primary sensor calibration result, or blank on error.

backup result ($3)

Backup sensor calibration result, or blank on error.

result target ($4)

Target result as passed from smart-stream-wrapper, or blank on error.

Files

/var/www/html/storage/smart-calibrate-hook.sh Optional hook invoked during calibration cycle

Related information
smart-stream *(on page 150)*

Library Functions

SMART_Close

close audio stream with a SMART device such as the calibratoion transducer of a microphone.

Synopsis

```
#include <smart.h>

int SMART_Close(int streamid);
```

Description

Given a streamid returned from a previous call to **SMART_Open(),** close the stream.

Return Value

SMART_Close() returns the number of bytes written from buffer or -1 on error setting errno.

Errors

EINVAL

Invalid streamid

EBUSY

Bad state

Related information
smart-daemon *(on page 140)*
SMART_Open *(on page 170)*
SMART_Read *(on page 173)*
SMART_Write *(on page 178)*
SMART_GetDeviceInfo, SMART_GetDeviceInfoBySN, SMART_GetNDevices *(on page 158)*

SMART_CloseWav

close a .wav or .w4v file previously opened with **SMART_CreateWav().**

Synopsis

```
#include <smart.h>

int SMART_CloseWav(int handle);
```

Description

Given handle returned from a previous call to **SMART_CreateWav()** complete writing out and closing the file.

Return Value

SMART_CloseWav() Returns 0 on success or -1 on error setting errno.

Errors

EINVAL

Invalid handle

EIO

An error occurred trying to write data to the file.

EBUSY

Bad state

Related information
SMART_CreateWav *(on page 157)*
SMART_WriteWav *(on page 178)*

SMART_CreateWav

create a .wav or .w4v file.

Synopsis

```
#include <smart.h>
#include <time.h>

int SMART_CreateWav(struct smart_wav_info_s * infop);
```

Description

Given parameters indicated by infop open a WAV or W4V file for writing.

Parameters for creating the .wav or .w4v file are indicated by a smart_wav_info_s structure as specified in <smart.h>:

```
struct smart_wav_info_s
  {
    const char *path;       // filename
    int         nchannels;  // number of channels
    int         samplerate; // sample rate, Hz
    int         compression;// compression mode
    struct timespec timestamp; // timestamp of metadata
    double      latitude;   // latitude for metadata
    double      longitude;  // longitude for metadata
    const char *guano;      // additional GUANO or NULL
  };
```

The **SMART_CreateWav()** function creates and opens a file indicated by path for writing by subsequent calls to **SMART_WriteWav().** The nchannels indicates the number of channels of interleaved audio are present. The compression indicates either an uncompressed PCM .WAV file if 0, or a compressed Wildlife Acoustics .W4V file specifying 4, 6, or 8 bits per sample. Other parameters are used to create the necessary file headers and include additional GUANO formatted meta information. If guano is non-null, it points to an optional multi-line string of additional GUANO key:value pairs for user-defined meta-data in addition to standard metadata.

Return Value

SMART_CreateWav() returns a stream handle, a small non-negative integer number, used in subsequent calls to **SMART_WriteWav()** and **SMART_CloseWav().** On error, -1 is returned and errno is set.

Errors

EINVAL

Invalid compression value. Must be 0 for uncompressed .wav, or 4, 6, or 8 for .w4v.

EIO

An error occurred trying to write file headers.

EACCESS

Permission denied trying to create or open the file specified by path.

Related information
SMART_WriteWav *(on page 178)*
SMART_CloseWav *(on page 156)*

SMART_GetDeviceInfo, SMART_GetDeviceInfoBySN, SMART_GetNDevices

get information about SMART devices such as microphones.

Synopsis

```
#include <smart.h>

int SMART_GetNDevices();

int SMART_GetDeviceInfo(int device, struct smart_device_info_t *infop);

int SMART_GetDeviceInfoBySN(uint8_t *sn, struct smart_device_info_t *infop);
```

Description

The **SMART_GetNDevices()** function caches information about detected SMART devices such as microphones from the SMART daemon and returns a count N indicating the number of devices discovered.

Subsequent calls to **SMART_getDeviceInfo()** return information about one of the discovered devices from 0 - N -1.

The **SMART_GetDeviceInfoBySN()** returns information about a specific device identified by a 6 octet serial number (mac address) without cacheing.

Device information is returned in a smart_device_info_t structure as sepcified in <smart.h>:

```
struct smart_device_info_t
  {
    uint8_t sn[6];     // serial number (MAC address) of device
    uint8_t model;     // model number
    uint8_t nchannels; // number of channels
    uint32_t capabilities; // bitmask of capabilities (see below)
    uint8_t is_sleeping;  // true if device is asleep
    uint8_t is_streaming; // true if device is streaming
    uint64_t rx_samples; // receive samples
    uint64_t rx_seq_err; // receive sequence errors
    uint64_t rx_drops;   // receive samples dropped due to congestion
    uint64_t tx_samples; // transmit samples (calibration data)
    uint32_t tx_pkts;    // dev eth transmit packet counter
    uint32_t tx_cols;    // dev eth transmit with collission counter
    uint32_t rx_pkts;    // dev eth receive packet counter
    uint32_t rx_crc;     // dev eth receive CRC error counter
    uint32_t rx_align;   // dev eth recieve alignment error counter
};
```

The capabilities is a bit mask of device capabilities defined in <smart_device_modes.h> defining supported sample rates, high-pass filter settings, gains settings, sensor selection, calibration, and heating.

SMART_SR_MASK

Sample rates:

SMART_SR_8KHZ

8,000 Hz

SMART_SR_12KHZ

12,000 Hz

SMART_SR_16KHZ

16,000 Hz

SMART_SR_22KHZ

22,050 Hz

SMART_SR_24KHZ

24,000 Hz

SMART_SR_32KHZ

32,000 Hz

SMART_SR_44KHZ

44,100 Hz

SMART_SR_48KHZ

48,000 Hz

SMART_SR_96KHZ

96,000 Hz

SMART_SR_192KHZ

192,000 Hz

SMART_SR_384KHZ

384,000 Hz

SMART_SR_500KHZ

500,000 Hz

SMART_HP_MASK

High-pass filter settings:

SMART_HP_OFF

No high-pass filter set

SMART_HP_250HZ

250 Hz

SMART_HP_1KHZ

1 kHz

SMART_HP_8KHZ

8 kHz

SMART_HP_16KHZ

16 kHz

SMART_GAIN_MASK

Gain settings

SMART_GAIN_0DB

0 dB

SMART_GAIN_6DB

6 dB

SMART_GAIN_12DB

12 dB

SMART_GAIN_18DB

18 dB

SMART_GAIN_24DB

24 dB

SMART_GAIN_30DB

30 dB

SMART_GAIN_36DB

36 dB

SMART_GAIN_42DB

42 dB

SMART_CH_0

Has a primary microphone sensor

SMART_CH_1

Has a secondary (backup) microphone sensor

SMART_CH_CAL

Has a calibration transducer

SMART_HEATER

Has a heating element (for defogging the sensors)

Return Value

SMART_GetNDevices() returns the number of devices detected or -1 on error setting errno.

SMART_GetDeviceInfo() and **SMART_TetDeviceInfoBySN()** return 0 on success or -1 on error setting errno

Errors

ECONNREFUSED

Unable to connect to the SMART daemon

EACCESS

Permission denied from missing or incorrect license, or unable to connect to SMART daemon socket.

ENODEV

Information about the requested device could not be found

EINTR

Error occurred while communicating with SMART daemon.

Bugs

The **SMART_GetNDevices()** and **SMART_GetDeviceInfo()** functions are not thread safe as they cache a query from the SMART daemon in a static variable.

Related information
smart-daemon *(on page 140)*
SMART_Reset *(on page 174)*
SMART_Wake *(on page 177)*
SMART_Sleep *(on page 175)*

SMART_GetSerial

Get serial number of SMART system

Synopsis

```
#include <smart.h>

int SMART_GetSerial(uint8_t *sn);
```

Description

The **SMART_GetSerial()** function write the 6 octet serial number (mac address) of the SMART system to sn by way of the SMART daemon. This is the MAC address of the Ethernet interface bound by the SMART daemon for communicating with devices such as microphones.

Return Value

SMART_GetSerial() return 0 on success or -1 on error setting errno.

Errors

ECONNREFUSED

Unable to connect to the SMART daemon

EACCESS

Permission denied from missing or incorrect license, or unable to connect to SMART daemon socket.

EINTR

Error occurred while communicating with SMART daemon.

Related information
smart-daemon *(on page 140)*
SMART_GetDeviceInfo, SMART_GetDeviceInfoBySN, SMART_GetNDevices *(on page 158)*

SMART_GetSystemStatus

Read system status from logger

Synopsis

```
#include <smart.h>

intSMART_GetSystemStatus(structsmart_system_status*status);
```

Description

Reads the system status from the logger service.

```
// System status
struct smart_system_status
  {
    time_t timestamp;// last update
    size_t bavail;   // available blocks
    size_t blocks;   // total blocks
    double voltage;  // voltage reading
    double load;     // load average
    size_t memtotal; // total memory (k)
    size_t memavail; // available memory (k)
  };
```

Return Value

Returns 0 on success, -1 otherwise.

Errors

No errors have yet been defined (always returns success).

Related information
SMART_KaleidoscopeCreate *(on page 164)*
SMART_KaleidoscopeAnalyzeFlush *(on page 163)*
SMART_KaleidoscopeAnalyzeGetZCFile *(on page 163)*
SMART_KaleidoscopeDelete *(on page 168)*
SMART_KaleidoscopeEventInit *(on page 168)*
SMART_KaleidoscopeEventPost *(on page 170)*
SMART_KaleidoscopeEventNext *(on page 169)*
SMART_GetSystemStatus *(on page 161)*
SMART_SetSystemStatus *(on page 175)*

SMART_KaleidoscopeAnalyzeBlock256

pass 256 samples to a Kaleidoscope analysis object for processing

Synopsis

```
#include <smart.h>

int SMART_KaleidoscopeAnalyzeBlock256( Kaleidoscope Handle handle, const in16_t *
 samples);
```

Description

Given a Kaleidosocpe Analysis instance identified by handle returned from a previous call to **SMART_KaleidoscopeCreate(),** pass 256 16-bit audio samples (512 bytes) from samples. If a detected event occurs, the callback function provided by **SMART_KaleidoscopeCreate()** may be called zero or more times, once for each detected event.

Return Value

Returns 0 on success, -1 otherwise.

Errors

No errors have yet been defined (always returns success).

Related information
SMART_KaleidoscopeCreate *(on page 164)*
SMART_KaleidoscopeAnalyzeFlush *(on page 163)*
SMART_KaleidoscopeAnalyzeGetZCFile *(on page 163)*
SMART_KaleidoscopeDelete *(on page 168)*
SMART_KaleidoscopeEventInit *(on page 168)*
SMART_KaleidoscopeEventPost *(on page 170)*
SMART_KaleidoscopeEventNext *(on page 169)*

SMART_KaleidoscopeAnalyzeFlush

Complete processing of buffered data

Synopsis

```
#include <smart.h>

int SMART_KaleidoscopeAnalyzeFlush( Kaleidoscope Handle handle);
```

Description

Given a Kaleidosocpe Analysis instance identified by handle returned from a previous call to **SMART_KaleidoscopeCreate(),** complete any processing of buffered data. samples. If a detected event occurs, the callback function provided by **SMART_KaleidoscopeCreate()** may be called zero or more times, once for each detected event.

Return Value

Returns 0 on success, -1 otherwise.

Errors

No errors have yet been defined (always returns success).

Related information
SMART_KaleidoscopeCreate *(on page 164)*
SMART_KaleidoscopeAnalyzeBlock256 *(on page 162)*
SMART_KaleidoscopeAnalyzeGetZCFile *(on page 163)*
SMART_KaleidoscopeDelete *(on page 168)*
SMART_KaleidoscopeEventInit *(on page 168)*
SMART_KaleidoscopeEventPost *(on page 170)*
SMART_KaleidoscopeEventNext *(on page 169)*

SMART_KaleidoscopeAnalyzeGetZCFile

Extract ZC file from detection event.

Synopsis

```
#include <smart.h>
```

```
intSMART_KaleidoscopeAnalyzeGetZCFile
 ( Kaleidoscope Handle  handle
        ,const char *  filename
        ,uint8_t *  buffer
        ,size_t  length
        ,struct timespec *  timestamp
        ,double  latitude
        ,double  longitude
        ,const char *  guano
        ,double  pretrigger
        );
```

Description

The function **SMART_KaleidoscopeGetZCFile()** may be called from the callback function specified in a previous call to **SMART_KaleidoscopeCreate().** The callback is typically invoked when the application calls either SMART_KaleidoscopeAnalyzeBlock256() or SMART_KaleidoscopeAnalyzeFlush(). This function writes the contents of a .zc file to the buffer specified up to length bytes in length. For alignment, the file provides pretrigger seconds ahead of the first detected signal. Other parameters including timestamp, latitude, longitude, and guano are used to provide additional meta data to the file.

A detection can include a zero-crossing representation in "Bat Analysis Mode", or a trace of the peak detected signal throughout a vocalization in "Non-Bat Analysis Mode" which can be rendered in a .zc zero-crossing file. The .zc file format is an extremely compact representation of an acoustic detection event representing a single frequency point for each point in time where signal is detected.

Return Value

Returns the number of bytes written to buffer or -1 on error setting errno.

Errors

ENOBUFS

The buffer provided was not large enough to receive the zero crossing data.

EIO

Unable to determine the SMART system serial number.

Notes

With metadata, a typical .zc file can fit within 65,536 bytes. The size of the file depends on the signal content of the detection.

Related information

SMART_KaleidoscopeCreate *(on page 164)*
SMART_KaleidoscopeAnalyzeBlock256 *(on page 162)*
SMART_KaleidoscopeAnalyzeFlush *(on page 163)*
SMART_KaleidoscopeDelete *(on page 168)*
SMART_KaleidoscopeEventInit *(on page 168)*
SMART_KaleidoscopeEventPost *(on page 170)*
SMART_KaleidoscopeEventNext *(on page 169)*

SMART_KaleidoscopeCreate

Create a Kaleidsocope instance for analysis of a single-channel audio stream.

Synopsis

```
#include <smart.h>

KaleidoscopeHandle SMART_KaleidoscopeCreate(struct smart_kaleidoscope_params * params,
 smart_kaleidoscope_event_callback callback);
```

Description

Given a set of parameters indicated by the structure params, create an instance of a Kaleidoscope analysis channel for analyzing an audio stream. The callback function specified will be called back from subsequent calls of **SMART_KaleidoscopeAnalyzeBlock256()** and **SMART_KaleidoscopenalyzeFlush()** in response to detected acoustic events.

The analysis parameters are provided in a smart_kaleidosocpe_params structure as follows:

```
struct smart_kaleidoscope_params
  {
    // Kaleidoscope analysis mode
    enum
      {
        MODE_BATS     = 0
       ,MODE_NON_BATS = 1
      } mode;          // Kaleidoscope analysis mode

    int samplerate;    // input samplerate, Hz

    // Analysis signal parameters
    //   Note: If using a cluster classifier, these values are
    //   inherited from the .kcs file.
    double minFreq;    // minimum frequency, Hz
    double maxFreq;    // maximum frequency, Hz
    double minDur;     // minimum duration, s
    double maxDur;     // maximum duration, s
    double maxGap;     // maximum inter-syllable gap, s

    // For bat analysis, the maximum sequence duration can be
    // specified to force the end of a trigger if maxGap isn't
    // observed.
    double maxSequence;// maximum duration of bat sequence, s

    // For bat analysis, minimum number of pulses required
    int    minPulses;   // bat mode minimum pulses
    int    zcEnhance;   // bat mode use enhanced processing

    // Classifier:
    //   Null for using signal detector
    //   .kcs file for cluster classifier
    //   .wcl file for old autoid bat classifiers
    //        (3.1.0 and earlier)
    //   directory for new autoid bat classifiers
    //        (4.1.0 and later) unzipped
    const char *path;

    // for autoid <0=sensitive,0=balanced,>0=accurate
    // for cluster classifier, this is max dist
    double      thold;

    // list of species codes with comma delimeters
```

```
    // or NULL for all
    const char *species;

    // constant frequency noise filter parameters
    double cfFilterMaxFreq;       // max frequency Hz to apply filter
    double cfFilterMaxBandwidth;  // max bandwidth Hz to apply filter

    // Prefix (microphone name)
    const char *prefix;
  };
```

Kaleidosocpe analysis has two distinct modes including "Bat Analysis Mode" and "Non-Bat Analysis Mode". In "Bat Analysis Mode", the signal parameters minFreq, maxFreq, minDur, and maxDur describe the range of expected echolocation pulses within a sequence while maxSequence describes the maximum duration of a sequence of pulses within a "detection". The minPulses specifies the minimum number of echolocation pulses in a sequence to be considered as a bat vs. noise. And zcEnhance should be true to indicate use of Kaleidoscope's advanced signal processing to convert full spectrum signals to zero crossing signals. Otherwise a broadband filter (between minFreq and maxFreq is applied and the signal is zero-crossed without further processing.

In "Non-Bat Analysis Mode" minFreq, maxFreq, minDur, and maxDur describe the range of expected "detections".

In both modes, the maxGap parameter indicates the maximum inter-pulse gap (for "Bat Analysis Mode") or detection gap (for "Non-Bat Analysis Mode") to determine when a detection event ends and a new detection event begins.

A bat auto-id classifier can be specified when using "Bat Analysis Mode" by setting path to point to a directory containing the un-zipped files comprising an Auto ID classifier from Kaleidoscope. The thold parameter indicates "more sensitive" (-1), "balanced" (0), or "more accurate" (1). A cluster classifier can be specified (in either "Bat Analysis Mode" or "Non-bat Analysis Mode" by specifying a .kcs file with path and the thold parameter indicates the maximum distance for a successful classification on a scale of 0 - 2. If a bat auto-id classifier or cluster classifier is specified, all species are selected by default unless species is non-null, in which case a comma separated list of species codes can be provided and only those species listed will be considered by the classifiers.

Data is passed to the classifier via **SMART_KaleidoscopeAnalyzeBlock256()** and ultimately flushed at the end of the data stream with **SMART_KaleidoscopeFlush().** These functions can in turn invoke the provided callback function for each detected event. The callback function has the following prototype:

```
typedef void (*smart_kaleidoscope_event_callback)
              (KaleidoscopeHandle handle
              ,struct smart_kaleidoscope_results *results
              );
```

The results describing the detected event are provided in the following smart_kaleidoscope_results structure:

```
enum KaleidoscopeResultType
  {
    RESULT_TYPE_PASS
   ,RESULT_TYPE_CALL
  };

struct smart_kaleidoscope_results
  {
    // Detection:
    //   The Kaleidoscope analysis engine buffers future samples
    //   while analyzing data in the past.  Therefore, when an
    //   event is detected, it may be after several subsequent
```

```
    //   blocks of samples have been passed to Kaleidoscope.
    //
    //   pre_offset indicates the negative offset (in seconds)
    //   prior to the last data fed to
    //   SMART_KaleidoscopeAnalyzeBlock256() where this detected
    //   event begins.
    double pre_offset; // negative offset to beginning, s
    double duration;   // duration of detected event, s

    // Autoid or clustering
    const char *ids[3];     // top 3 classificiation results
    float       margins[3]; // top 3 margin results

    // If autoid for bats
    int         nPulsesClassified; // number of pulses classified
    int         nPulsesMatching;   // number pulses matching id

    // Data specific to bat analysis
    // (not valid for non-bat analysis except Fmax, Fmin, Fmean)
    // These are generally averages across each detected
    // echolocation pulsein the detected trigger event.
    int    N;     // number of detected echolocation pulses
    double Fc;    // average characteristic frequency, Hz
    double Sc;    // average characteristic slope, octaves per second
    double Fmax;  // average maximum frequency, Hz
    double Fmin;  // average minimum frequency, Hz
    double Fmean; // average mean frequency, Hz
    double TBC;   // average time between calls, seconds
    double Fk;    // average frequency at the knee, Hz
    double Tk;    // average time at knee, seconds
    double S1;    // average initial slope, octaves per second
    double Tc;    // average time of characteristic, seconds
    double Dur;   // average pulse duration, seconds
    double Qual;  // average pulse quality, %

    // Prefix populated by libsmart from the smart_kaleidoscope_params
#   define SMART_RESULTS_MAX_PREFIX 32
    char prefix[SMART_RESULTS_MAX_PREFIX];

    // These for event logging, not populated by Kaleidoscope libraries
    uint32_t seq;         // used internally by logger to detect wrap
    struct timespec timestamp; // timestamp, optional, set by client
    uint8_t sn[6];        // serial number, optional, set by client

    // Result type indicated by SMART library
    enum KaleidoscopeResultType resultType;
  };
```

Return Value

SMART_KaleidoscopeCreate() returns a handle of type KaleidoscopeHandle or NULL on error setting errno.

Errors

EACCESS

Permission denied from missing or incorrect license or unable to open classifier file if specified.

EIO

Error occurred trying to read classifier file if specified.

Related information
SMART_KaleidoscopeDelete *(on page 168)*
SMART_KaleidoscopeAnalyzeBlock256 *(on page 162)*
SMART_KaleidoscopeAnalyzeFlush *(on page 163)*
SMART_KaleidoscopeAnalyzeGetZCFile *(on page 163)*
SMART_KaleidoscopeEventInit *(on page 168)*
SMART_KaleidoscopeEventPost *(on page 170)*
SMART_KaleidoscopeEventNext *(on page 169)*

SMART_KaleidoscopeDelete

Delete an instance of Kaleidoscope analysis.

Synopsis

```
#include <smart.h>

int SMART_KaleidoscopeDelete( Kaleidoscope Handle handle);
```

Description

Given a Kaleidosocpe Analysis instance identified by handle returned from a previous call to **SMART_KaleidoscopeCreate(),** delete the instance and free resources.

Return Value

Returns 0 on success, -1 otherwise.

Errors

No errors have yet been defined (always returns success).

Related information
SMART_KaleidoscopeCreate *(on page 164)*
SMART_KaleidoscopeAnalyzeBlock256 *(on page 162)*
SMART_KaleidoscopeAnalyzeFlush *(on page 163)*
SMART_KaleidoscopeAnalyzeGetZCFile *(on page 163)*
SMART_KaleidoscopeEventInit *(on page 168)*
SMART_KaleidoscopeEventPost *(on page 170)*
SMART_KaleidoscopeEventNext *(on page 169)*

SMART_KaleidoscopeEventInit

Initialize event subsystem for posting or retrieving events

Synopsis

```
#include <smart.h>

intSMART_KaleidoscopeEventInit();
```

Description

Connect to the SMART Kaleidoscope Event system for posting or retrieving events. This function must be called before calling **SMART_KaleidoscopeEventPost(), SMART_KaleidoscopeEventNext(), SMART_GetSystemStatus(),** or **SMART_SetSystemStatus().**

Return Value

Returns 0 on success, -1 otherwise.

Errors

No errors have yet been defined (always returns success).

Related information
SMART_KaleidoscopeCreate *(on page 164)*
SMART_KaleidoscopeAnalyzeFlush *(on page 163)*
SMART_KaleidoscopeAnalyzeGetZCFile *(on page 163)*
SMART_KaleidoscopeDelete *(on page 168)*
SMART_KaleidoscopeEventInit *(on page 168)*
SMART_KaleidoscopeEventPost *(on page 170)*
SMART_KaleidoscopeEventNext *(on page 169)*
SMART_GetSystemStatus *(on page 161)*
SMART_SetSystemStatus *(on page 175)*

SMART_KaleidoscopeEventNext

Get next event for specified consumer

Synopsis

```
#include <smart.h>

int SMART_KaleidoscopeEventNext( enum KaleidoscopeEventConsumer consumer
 , struct smart_kaleidoscope_results *event
 , int block
 , struct timespec *timeout);
```

Description

Get the next event for the specified consumer. The following consumers are defined:

```
 KALEIDOSCOPE_EVENT_CONSUMER_SERIAL
 KALEIDOSCOPE_EVENT_CONSUMER_SCADA
 KALEIDOSCOPE_EVENT_CONSUMER_USER1
 KALEIDOSCOPE_EVENT_CONSUMER_USER2

If block is non-zero, the call will block until a new event is posted or
a timeout occurs.  Otherwise, the call returns immediately.
```

Return Value

Returns 0 if non-blocking and no event. Returns 1 if an event is returned. Returns <0 on error.

Errors

No errors have yet been defined (always returns success).

Related information
SMART_KaleidoscopeCreate *(on page 164)*
SMART_KaleidoscopeAnalyzeFlush *(on page 163)*
SMART_KaleidoscopeAnalyzeGetZCFile *(on page 163)*
SMART_KaleidoscopeDelete *(on page 168)*
SMART_KaleidoscopeEventInit *(on page 168)*
SMART_KaleidoscopeEventPost *(on page 170)*
SMART_KaleidoscopeEventNext *(on page 169)*
SMART_GetSystemStatus *(on page 161)*
SMART_SetSystemStatus *(on page 175)*

SMART_KaleidoscopeEventPost

Post an event to the event logger

Synopsis

```
#include <smart.h>

int SMART_KaleidoscopeEventPost( const struct smart_kaleidoscope_results *event);
```

Description

Post the event to the event logger

Return Value

Returns 0 on success, -1 otherwise.

Errors

No errors have yet been defined (always returns success).

Related information
SMART_KaleidoscopeCreate *(on page 164)*
SMART_KaleidoscopeAnalyzeFlush *(on page 163)*
SMART_KaleidoscopeAnalyzeGetZCFile *(on page 163)*
SMART_KaleidoscopeDelete *(on page 168)*
SMART_KaleidoscopeEventInit *(on page 168)*
SMART_KaleidoscopeEventPost *(on page 170)*
SMART_KaleidoscopeEventNext *(on page 169)*
SMART_SetSystemStatus *(on page 175)*
SMART_GetSystemStatus *(on page 161)*

SMART_Open

open a connection to a SMART device such as a microphone.

Synopsis

```
#include <smart.h>

int SMART_Open(uint8_t * sn, uint8_t channelmask, uint8_t channel, uint32_t config, int *
 fdp);
```

Description

Given a sn 6 octet SMART device serial number (mac address), **SMART_Open()** returns a stream identifier, a small, nonnegative integer for use in subsequent calls **SMART_Read(), SMART_Write(),** and **SMART_Close().** Each stream is for a single audio channel specified by channel. If a SMART device is capable of multi-channel streaming (e.g. stereo, etc.), then each call to **SMART_Open()** must specify a channelmask indicating all of the channels to be opened on the SMART device. The first call to **SMART_Open()** will begin streaming all of the channels, but each individual call to **SMART_Open()** will attach to the specific channel identified by channel. The channel number is an integer beginning with zero, and the channelmask is a bitmask with channel 0 corresponding to the least-significant bit.

If fdp is non-NULL, **SMART_Open()** will write the underlying socket file descriptor to *fdp and mark the socket as non-blocking. In this way, callers can use select() to poll for events on the audio stream, but should still use **SMART_Read()** and **SMART_Write()** to read and write data from the stream using streamid.

The config is a bitmask of requested modes from the device and correspond to the capabilities of the device as defined in <smart_device_modes.h> to specify requested sample rate, high-pass filter setting, gains setting, sensor selection, calibration, and heating.

SMART_SR_MASK

Sample rates:

SMART_SR_8KHZ

8,000 Hz

SMART_SR_12KHZ

12,000 Hz

SMART_SR_16KHZ

16,000 Hz

SMART_SR_22KHZ

22,050 Hz

SMART_SR_24KHZ

24,000 Hz

SMART_SR_32KHZ

32,000 Hz

SMART_SR_44KHZ

44,100 Hz

SMART_SR_48KHZ

48,000 Hz

SMART_SR_96KHZ

96,000 Hz

SMART_SR_192KHZ

192,000 Hz

SMART_SR_384KHZ

384,000 Hz

SMART_SR_500KHZ

500,000 Hz

SMART_HP_MASK

High-pass filter settings:

SMART_HP_OFF

No high-pass filter set

SMART_HP_250HZ

250 Hz

SMART_HP_1KHZ

1 kHz

SMART_HP_8KHZ

8 kHz

SMART_HP_16KHZ

16 kHz

SMART_GAIN_MASK

Gain settings

SMART_GAIN_0DB

0 dB

SMART_GAIN_6DB

6 dB

SMART_GAIN_12DB

12 dB

SMART_GAIN_18DB

18 dB

SMART_GAIN_24DB

24 dB

SMART_GAIN_30DB

30 dB

SMART_GAIN_36DB

36 dB

SMART_GAIN_42DB

42 dB

SMART_CH_0

Select the primary microphone sensor

SMART_CH_1

Select the secondary (backup) microphone sensor

SMART_CH_CAL

Enable the calibrator (writes via **SMART_Write()** will send samples to the calibration transducer).

SMART_HEATER

Turn on the heating element (for defogging the sensors)

Return Value

SMART_Open() returns a stream id or -1 on error setting errno.

Errors

ECONNREFUSED

Unable to connect to the SMART daemon

EACCESS

Permission denied from missing or incorrect license, or unable to connect to SMART daemon socket.

ENODEV

Information about the requested device could not be found or there was a mismatch between the device capabilities and the requested configuration.

EINTR

Error occurred while communicating with SMART daemon. **EINVAL** Invalid configuraiton

Related information
smart-daemon *(on page 140)*
SMART_Read *(on page 173)*
SMART_Write *(on page 178)*
SMART_Close *(on page 155)*
SMART_GetDeviceInfo, SMART_GetDeviceInfoBySN, SMART_GetNDevices *(on page 158)*

SMART_Read

read audio samples from a SMART microphone device.

Synopsis

```
#include <smart.h>
#include <time.h>

int SMART_Read(int streamid, uint16_t * buffer, size_t length, struct timespec * tsp);
```

Description

Given a streamid returned from a previous call to **SMART_Open(),** read up to length bytes of 16-bit audio samples into buffer returning the number of bytes read.

If tsp is not NULL, a struct timespec will be written to *tsp indicating the time corresponding to the first sample in the returned buffer. The time has microsecond precision and is typically accurate to less than a millisecond.

The call is blocking unless opened with a non-null fdp specified in the previous call to **SMART_Open().**

Return Value

SMART_Read() returns the number of bytes written to buffer or -1 on error setting errno. A return value of zero indicates the SMART microphone device closed the connection.

Errors

EINVAL

Invalid streamid or an odd value for buflen.

EAGAIN or EWOULDBLOCK

The stream was opened non-blocking and no data is available.

Related information
SMART_Open *(on page 170)*
SMART_Write *(on page 178)*
SMART_Close *(on page 155)*
SMART_GetDeviceInfo, SMART_GetDeviceInfoBySN, SMART_GetNDevices *(on page 158)*

SMART_Reset

reset a SMART device such as a microphone

Synopsis

```
#include <smart.h>

int SMART_Reset(uint8_t *sn);
```

Description

The **SMART_Reset()** function sends a reset signal to the SMART device identified by a 6 octet serial number (mac address) by way of the SMART daemon

Return Value

SMART_Reset() return 0 on success or -1 on error setting errno.

Errors

ECONNREFUSED

Unable to connect to the SMART daemon

EACCESS

Permission denied from missing or incorrect license, or unable to connect to SMART daemon socket.

ENODEV

The requested device could not be found

EINTR

Error occurred while communicating with SMART daemon.

Related information
smart-daemon *(on page 140)*
SMART_Wake *(on page 177)*

SMART_Sleep *(on page 175)*
SMART_GetDeviceInfo, SMART_GetDeviceInfoBySN, SMART_GetNDevices *(on page 158)*

SMART_SetSystemStatus

Used by the logger service to update system status

Synopsis

```
#include <smart.h>

intSMART_SetSystemStatus();
```

Description

Used by the logger service to update the system status.

```
// System status
struct smart_system_status
  {
    time_t timestamp;// last update
    size_t bavail;   // available blocks
    size_t blocks;   // total blocks
    double voltage;  // voltage reading
    double load;     // load average
    size_t memtotal; // total memory (k)
    size_t memavail; // available memory (k)
  };
```

Return Value

Returns 0 on success, -1 otherwise.

Errors

No errors have yet been defined (always returns success).

Related information
SMART_KaleidoscopeCreate *(on page 164)*
SMART_KaleidoscopeAnalyzeFlush *(on page 163)*
SMART_KaleidoscopeAnalyzeGetZCFile *(on page 163)*
SMART_KaleidoscopeDelete *(on page 168)*
SMART_KaleidoscopeEventInit *(on page 168)*
SMART_KaleidoscopeEventPost *(on page 170)*
SMART_KaleidoscopeEventNext *(on page 169)*
SMART_GetSystemStatus *(on page 161)*
SMART_SetSystemStatus *(on page 175)*

SMART_Sleep

put a SMART device such as a microphone to sleep

Synopsis

```
#include <smart.h>

int SMART_Sleep(uint8_t *sn);
```

Description

The **SMART_Sleep()** function sends a sleep signal to the SMART device identified by a 6 octet serial number (mac address) by way of the SMART daemon to request that the device enter a low power sleep state.

Return Value

SMART_Sleep() return 0 on success or -1 on error setting errno.

Errors

ECONNREFUSED

Unable to connect to the SMART daemon

EACCESS

Permission denied from missing or incorrect license, or unable to connect to SMART daemon socket.

ENODEV

The requested device could not be found

EINTR

Error occurred while communicating with SMART daemon.

Related information
smart-daemon *(on page 140)*
SMART_Wake *(on page 177)*
SMART_Reset *(on page 174)*
SMART_GetDeviceInfo, SMART_GetDeviceInfoBySN, SMART_GetNDevices *(on page 158)*

SMART_Upgrade

upgrade a SMART device firmware

Synopsis

```
#include <smart.h>

int SMART_Upgrade(uint8_t *sn);
```

Description

The **SMART_Upgrade()** function sends an upgrade signal to the SMART device identified by a 6 octet serial number (mac address) by way of the SMART daemon to upgrade the device firmware. The SMART daemon will then initiate an upgrade sequence by sending the file at /usr/local/share/SMART/firmware_%d where %d is the device model number. This file would typically be a symbolic link to a model and version specific firmware file. When the firmware upgrade is complete, the device should reboot and then run the new firmware. Applications should verify that the new firmware is running.

Return Value

SMART_Upgrade() return 0 on success or -1 on error setting errno.

Errors

ECONNREFUSED

Unable to connect to the SMART daemon

EACCESS

Permission denied from missing or incorrect license, or unable to connect to SMART daemon socket.

ENODEV

The requested device was not in a sleep state or could not be found.

EINTR

Error occurred while communicating with SMART daemon.

Related information
smart-daemon *(on page 140)*
SMART_Sleep *(on page 175)*
SMART_Reset *(on page 174)*
SMART_GetDeviceInfo, SMART_GetDeviceInfoBySN, SMART_GetNDevices *(on page 158)*

SMART_Wake

wake a SMART device that was previously sleeping

Synopsis

```
#include <smart.h>

int SMART_Wake(uint8_t *sn);
```

Description

The **SMART_Wake()** function sends a wake-up signal to the SMART device identified by a 6 octet serial number (mac address) by way of the SMART daemon to wake the device from sleep mode.

Return Value

SMART_Wake() return 0 on success or -1 on error setting errno.

Errors

ECONNREFUSED

Unable to connect to the SMART daemon

EACCESS

Permission denied from missing or incorrect license, or unable to connect to SMART daemon socket.

ENODEV

The requested device was not in a sleep state or could not be found.

EINTR

Error occurred while communicating with SMART daemon.

Related information
smart-daemon *(on page 140)*
SMART_Sleep *(on page 175)*
SMART_Reset *(on page 174)*
SMART_GetDeviceInfo, SMART_GetDeviceInfoBySN, SMART_GetNDevices *(on page 158)*

SMART_Write

write audio samples to a SMART device such as the calibratoion transducer of a microphone.

Synopsis

```
#include <smart.h>

int SMART_Write(int streamid, uint16_t * buffer, size_t length);
```

Description

Given a streamid returned from a previous call to **SMART_Open(),** write up to length bytes of 16-bit audio samples from buffer returning the number of bytes written.

The call is blocking unless opened with a non-null fdp specified in the previous call to **SMART_Open().**

Return Value

SMART_Write() returns the number of bytes written from buffer or -1 on error setting errno.

Errors

EINVAL

Invalid streamid or an odd value for buflen.

EAGAIN or EWOULDBLOCK

The stream was opened non-blocking and no data is available.

EPIPE

The connection was closed

Related information
SMART_Open *(on page 170)*
SMART_Read *(on page 173)*
SMART_Close *(on page 155)*
SMART_GetDeviceInfo, SMART_GetDeviceInfoBySN, SMART_GetNDevices *(on page 158)*

SMART_WriteWav

write audio samples to a .wav or .w4v file previously opened with **SMART_CreateWav().**

Synopsis

```
#include <smart.h>

int SMART_WriteWav(int handle, const uint16_t * buffer, size_t length);
```

Description

Given handle returned from a previous call to **SMART_CreateWav()** write length bytes of audio samples from buffer.

Return Value

SMART_WriteWav() returns the number of bytes written or -1 on error setting errno.

Errors

EINVAL

Invalid handle or length is ont even. **EIO** An error occurred trying to write data to the file.

Related information

SMART_CreateWav *(on page 157)*
SMART_CloseWav *(on page 156)*

Glossary

acoustically triggered curtailment

The practice of slowing or stopping a wind turbine in response to animal vocalizations. The goal of acoustically triggered curtailment is to prevent animal deaths while minimizing unneccessary turbine downtime.

bat pass

A recording of multiple bat echolocation pulses in a single audio file or triggered event. Ideally, a bat pass represents a single bat flying past a microphone. In reality, a recording of a single "bat pass" may include the vocalizations of multiple bats.

call body

The portion of the call between the *knee* and the *characteristic frequency (Fc)*.

characteristic frequency

The frequency of the flattest point of the call, where *slope* has its absolute minimum value.

knee

The beginning of a *call body*.

microphone element

The component of a microphone responsible for converting sound waves into an electrical signal. A microphone sold as a complete product includes a body an other electronic components, such as an amplification circuit, aside from the microphone element itself. The SMART MIC-1 microphone has two microphone elements and chooses which to use based on the results of its automatic sensitivity tests.

Network Time Protocol

An internet protocol used to synchronize clocks between computer systems.

octaves per second

A logarithmic unit of *slope*. An *octave* is a power of two between frequencies.

> **Example**
> A change from 40 kHz to 10 kHz over the span of 1 second would correspond to an average slope of 2 octaves per second.

OPC Unified Architecture

A machine-to-machine communication protocol used for industrial automation developed by the OPC Foundation. The SMART Controller OPC UA interface provides a secure mechanism for retrieving status information, system updates and alarm notifications from a SMART device.

Power over Ethernet

A system that allows for DC power to be provided over an Ethernet cable in addition to data transmission. The SMART Controller has two Ethernet ports with PoE support that can directly power two SMART MIC-1 microphones.

slope

The measurement of change in frequency over time. In the context of bat call analysis, slope is measured in *OPS*. In the context of Kaleidoscope Pro bat call analysis features, positive slope values correspond to decreasing frequency over time.

zero-crossing

A file format that represents sound using timestamps to indicate when the sound's waveform crosses a reference level a set number of times.

Warranty and Disclosures

Except as specifically provided herein, Wildlife Acoustics makes no warranty of any kind, express or implied, with respect to this product.

Table 8. Hardware Limited Warranty

Product	Component	Warranty Period
SMART System	SMART Controller	2 years
	SMART MIC-1	5 years

Wildlife Acoustics, Inc. Limited Warranty

Hardware

Wildlife Acoustics, Inc. ("WAI") warrants to the original end user ("Customer") that new WAI branded products will be free from defects in workmanship and materials, under normal use. Refer to the Hardware Limited Warranty table at the top of this page for the applicable warranty period from the original date of purchase.

WAI warrants refurbished WAI products, marked and sold as such, for ninety (90) days from the original purchase date.

Software

WAI warrants to Customer that any WAI branded software will perform in substantial conformance to their schedule specifications for a period of ninety (90) days from the date of original purchase. WAI warrants the media containing software against failure during the warranty period. WAI makes no warranty or representation that the operation of the software products will be uninterrupted or error free, or that all defects in the software products will be corrected.

Exclusions

This warranty excludes (1) physical damage to the surface of the product, including cracks or scratches on the outside casing; (2) damage caused by misuse, neglect, improper installation or testing, unauthorized attempts to open, repair, or modify the product, or any other cause beyond the range of the intended use; (3) damage caused by accident, fire, power changes, other hazards, or acts of God; or (4) use of the product with any non-WAI device or service if such device or service causes the problem.

Any third party products, including software, included with WAI products are not covered by this WAI warranty and WAI makes no representations or warranties on behalf of such third parties. Any warranty on such products is from the supplier or licensor of the product.

No warranty is provided by WAI unless the product was purchased from an authorized distributor or authorized reseller.

Exclusive Remedies

Should a covered defect occur during the warranty period and you notify WAI, your sole and exclusive remedy shall be, at WAI's sole option and expense, to repair or replace the product or software. If WAI cannot reasonably repair nor replace then WAI may, in its sole discretion, refund the purchase price paid for the product. Replacement products or parts may be new or reconditioned or comparable versions of the defective item. WAI warrants any replaced or repaired product, part, or software for a period of ninety (90) days from shipment, or through the end of the original warranty, whichever is longer.

Obtaining Warranty Service

Customer should refer to the WAI website at https://www.wildlifeacoustics.com/contact-us for information on obtaining warranty service authorization. Methods for obtaining warranty service may vary depending on whether purchases were made from an authorized provider of WAI products or from WAI directly. All requests for warranty service authorization must be made within the applicable warranty period. Dated proof of original purchase will be required. Products or parts shipped by Customer to WAI must be sent postage-paid and packaged appropriately for safe shipment. WAI is not responsible for Customer products received without a warranty service authorization and may be rejected. Repaired or replacement products will be shipped to Customer at WAI expense. All products or parts that are replaced become the property of WAI. WAI shall not be responsible for Customer software, firmware, information, or memory data contained in, stored on, or integrated with any products returned to WAI for repair, whether under warranty or not. The repair and replacement process for products or parts in locations outside of the United States will vary depending on Customer's location.

Warranties Exclusive

THE FOREGOING WARRANTIES AND REMEDIES ARE EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES, TERMS OR CONDITIONS, EXPRESS, IMPLIED OR STATUTORY, INCLUDING WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY, CORRESPONDENCE WITH DESCRIPTION, SATISFACTORY QUALITY AND NON-INFRINGEMENT, ALL OF WHICH ARE EXPRESSLY DISCLAIMED BY WAI AND ITS SUPPLIERS.

Limitations of Liability

NEITHER WAI NOR ITS SUPPLIERS SHALL BE LIABLE FOR INCIDENTAL, CONSEQUENTIAL, INDIRECT, SPECIAL, OR PUNITIVE DAMAGES OF ANY KIND, LOSS OF INFORMATION OR DATA, LOSS OF REVENUE, LOSS OF BUSINESS OR OTHER FINANCIAL LOSS ARISING OUT OF OR IN CONNECTION WITH THE SALE OR USE OF THIS PRODUCT, WHETHER BASED IN CONTRACT, TORT (INCLUDING NEGLIGENCE), STRICT PRODUCT LIABILITY OR ANY OTHER THEORY, EVEN IF WAI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES AND EVEN IF ANY LIMITED REMEDY SPECIFIED IN THIS LICENCE AGREEMENT IS DEEMED TO HAVE FAILED OF ITS ESSENTIAL PURPOSE. WAI'S ENTIRE LIABILITY SHALL BE LIMITED TO REPLACEMENT, REPAIR, OR REFUND OF THE PURCHASE PRICE PAID, AT WAI'S OPTION. IN NO EVENT SHALL WAI'S LIABILITY FOR ALL DAMAGES RELATED TO THE PURCHASE OF PRODUCT EXCEED THE AMOUNT PAID FOR THE APPLICABLE PRODUCT. THE FOREGOING LIMITATIONS WILL APPLY EVEN IF THE ABOVE STATED REMEDY FAILS OF ITS ESSENTIAL PURPOSE.

Disclaimer

Some countries, states, or provinces do not allow the exclusion or limitation of implied warranties or the limitation of incidental or consequential damages so the above limitations and exclusions may be limited in their application to you. When implied warranties may not be excluded in their entirety, they will be limited to the duration of the applicable written warranty. This warranty gives you specific legal rights; you may have other rights that may vary depending on local law. Your statutory rights are not affected.

Governing Law

This Limited Warranty shall be governed by the laws of the Commonwealth of Massachusetts, and by the laws of the United States, excluding their conflicts of laws principles. The United Nations Convention on Contracts for the International Sale of Goods is hereby excluded in its entirety from application to this Limited Warranty.

Declaration of Conformity

According to EN ISO/IEC 17050-1:2010

No: WAI20210407

Manufacturer:

Wildlife Acoustics, Inc.
3 Mill and Main Place, Suite 110
Maynard, MA 01754
United States of America

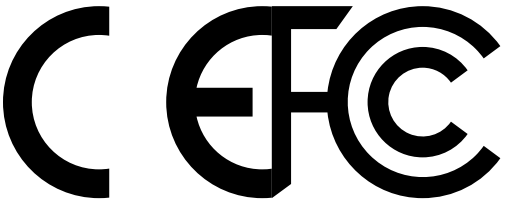
Declares that the following product:

Product Name	Product Model Number	Product Type
SMART	SMART MIC-1	Ethernet Microphone

Conforms to the appropriate country standards and governing regulations listed below. As the manufacturer, we are fully responsible for the design and production of the above-mentioned equipment.

- (FCC) Code of Federal Regulations, Title 47, Part 15, Subpart B: Class A Device (2015): Radio Frequency Devices – Unintentional radiators
- AS CISPR 11, (2017): Industrial, scientific and medical (ISM) radio-frequency equipment – electromagnetic disturbance characteristics – limits and methods of measurement, Class A
- EN 55011, (2016): Industrial, scientific and medical (ISM) radio-frequency equipment – Electromagnetic disturbance characteristics – Limits and methods of measurement, Class A
- ICES-003, (2020): Industry Canada, Interference-Causing Equipment Standard, Digital Apparatus, Class A
- EN 61326-1:2013: Electrical Equipment for Measurement, Control and Laboratory use EMC Requirements
- EN 61000-4-2: Electromagnetic compatibility (EMC) Testing and measurement techniques - Electrostatic discharge immunity test; 4 kV Contact Discharge, 8 kV Air Discharge, 4 kV Horizontal and Vertical Coupling Planes (HCP and VCP, respectively)
- EN 61000-4-3: Electromagnetic compatibility (EMC) Testing and measurement techniques - Radiated, radiofrequency, electromagnetic field immunity test; 3 V/m, 80 to 1000 MHz; 3 V/m, 1.4 GHz to 2 GHz; 1 V/m, 2 GHz to 2.7 GHz
- EN 61000-4-4: Electromagnetic compatibility (EMC) Testing and measurement techniques - Electrical fast transient/burst immunity test; 1 kV on DC Mains; 1 kV on Data Cables
- EN 61000-4-5: Electromagnetic compatibility (EMC) Testing and measurement techniques - Surge immunity test; 1 kV on DC Mains
- EN 61000-4-6: Electromagnetic compatibility (EMC) Testing and measurement techniques - Immunity to conducted disturbances, induced by radio-frequency fields; 3 Vrms on DC Mains, 150 kHz to 80 MHz; 3 Vrms on Data Cables 150 kHz to 80 MHz
- IEC 60529: Ingress Protection IP67

Marking appears as follows:



This product was tested in a typical configuration.

Ian Agranat, President
Wildlife Acoustics, Inc.
April 7, 2021

Electromagnetic Interference

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules.

These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Changes or modifications not expressly approved by Wildlife Acoustics, Inc. could void the user's authority to operate the equipment.

Note: Use of ferrite clamped cables are required to comply with the Class A limits in part 15 of the FCC rules. A Fair-Rite 0443167251, 0431164281, or equivalent clamp must be placed on each cable near the microphone. This clamp is provided with all cables sold by Wildlife Acoustics.

Prohibition Against Eavesdropping

United States law (Federal Communications Commission Part 15 Section 15.9) states:

> "Except for the operations of law enforcement officers conducted under lawful authority, no person shall use, either directly or indirectly, a device operated pursuant to the provisions of this Part for the purpose of overhearing or recording the private conversations of others unless such use is authorized by all of the parties engaging in the conversation."

You are responsible for complying with all applicable laws within your jurisdiction.

Patents

The SMART System is covered under the following patents:

- AU 202210794
- AU 202210795
- AU 202210796
- AU 202210797
- GB 2559839
- GB 6188390
- GB 6188391
- GB 6188392
- GB 6188393
- EP 1661123
- EP 2877820
- US 8,995,230
- US 10,911,854

Copyright Notices

Contact Information

Wildlife Acoustics, Inc.
3 Mill and Main Place, Suite 110
Maynard, MA 01754
United States of America
+1 (978) 369-5225
U.S. toll-free: +1 (888) 733-0200
https://www.wildlifeacoustics.com