VEET 2 Device Manual

This manual describes the operation of the Visual Environment Evaluation Tool (VEET) device and VEETManager configuration software. This manual is intended to provide information to researchers who will perform studies with the VEET devices and the community that will analyze the data from the device. This manual is not intended for distribution to an end-user or study participant.

VEET 2. *The device is supplied as two temple arms designed to be installed on glasses frames.*

Table of Contents

Introduction

The Visual Environment Evaluation Tool (VEET) device is a pair of temple arm form-factor devices that allow a researcher to gather visual information that the eye receives throughout the day without interruption of normal activities with the following high level goals

- Can be worn all day, by most people
- Collects data continuously for multiple weeks
- Measures information about the level and type of illumination the eye receives
- Measures near objects in the wearer's field of view
- Full transparency on how sensors are calibrated and utilized to produce useable values
- Does not compromise privacy

To achieve these goals each temple arm of the VEET device contains four (4) robust, solid state sensors that provided the following core information:

- Time of Flight Sensor for object detection directly in millimeters
- Ambient Light Sensor for illuminance directly in Lux
- Spectral Sensor for post processing spectral analysis
- Inertial Measurement Unit for determining if the device is being worn

To minimize privacy concerns, the VEET device gathers visual information without the use of any cameras or other image sensors. Each device gathers point-in-time data at rates up to 1/2 Hertz (Hz) (once every two (2) seconds) and is able to measure and retain data for many weeks.

The VEET device is designed to allow the research participant to explore normal environments without undue attention being paid to the VEET device by either the research participant or associated social peers.The device is designed to collect data even when charging so that it will capture information about the environment even at night. It is recommended the research participant charges the device in the same room that they sleep in.

Each temple arm of the VEET device is a standalone data acquisition device with a full sensor suite and an individual set of processing and storage capabilities. Each temple arm of the VEET device only communicates over USB. The device has no other connection capabilities over wifi, bluetooth, or other wireless protocols. Each device charges independently over USB, and at a maximum data collection rate of ½ Hz, provides data acquisition of a full day on a single charge. The device requires the researcher or research participant to charge each temple arm every night to continuously gather data.

Intended Use

The VEET device is intended for researchers studying human ocular development in controlled experiments and studies under supervision of their Institutional Review Board (IRB) or equivalent. The VEET device is not intended for sale or general distribution.

As a research device, only the researcher and the associated support staff should interact with and configure the VEET device. A research participant should only interact with the VEET device to wear and charge the VEET device as defined by the researcher per the protocols of the study. Use of the VEET device outside of study protocols may result in data corruption. Data is accessible using a USB connection to a computer and is not protected against tampering.

This product may not be placed on the market or put into service until it has been made to comply with local law, including laws in the EU, UK and Switzerland if the device is in use in these jurisdictions.

Safety Precautions

Laser Warnings

Class 1 Laser. *Warning on the device label.*

> **WARNING:** This device contains a Class I laser. Use other than as described in the user guide, repair, or disassembly may cause damage, which may result in hazardous exposure to infrared emissions that are not visible. This equipment may only be serviced by Meta.

NOTE: Complies with FDA performance standards for laser products except for conformance with IEC 60825-1 Ed. 3., as described in Laser Notice No. 56, dated May 8, 2019.

FCC Notice

This device has not been authorized as required by the rules of the Federal Communications Commission.

This device is not, and may not be, offered for sale or lease, or sold or leased, until authorization is obtained.

Operating Conditions

The VEET device must be charged at ambient temperatures below 27ºC (80ºF). This includes when the device is connected to a computer for data transfer.

The VEET device is rated for non-charging use from -10ºC to 37ºC (98ºF) up to 95% humidity, non-condensing.

Do not leave the device where it could be exposed to extremely high temperatures, such as an enclosed automobile in direct sunlight.

Do not disassemble

There are no user serviceable internal components. Touching the device's internal components could result in injury. In the event of a malfunction, contact Meta for next steps.

Should the device break open as a result of a fall or other accident, collect and isolate the components and contact Meta for data retrieval and return instructions.

California Proposition 65 Warning

WARNING: This product contains nickel, which is known to the State of California to cause cancer. For more information go to www.P65Warnings.ca.gov.

Skin contact

The device is manufactured with a common spectacle frame and temple arm material, Grilamid TR-90, and is in contact with the skin of a user. Should the user experience any redness, swelling or other skin irritation immediately discontinue use.

Preparing for Use

The VEET device arrives as a pair of temple arms with two mounting screws that is ready to be paired with the following frames:

- Ray-Ban New Wayfarer junior frames

The following frames have been tested and are able to be accommodated with modifications to the device:

- Ray-Ban New Wayfarer adult frames

- Puma frames PJ0009O

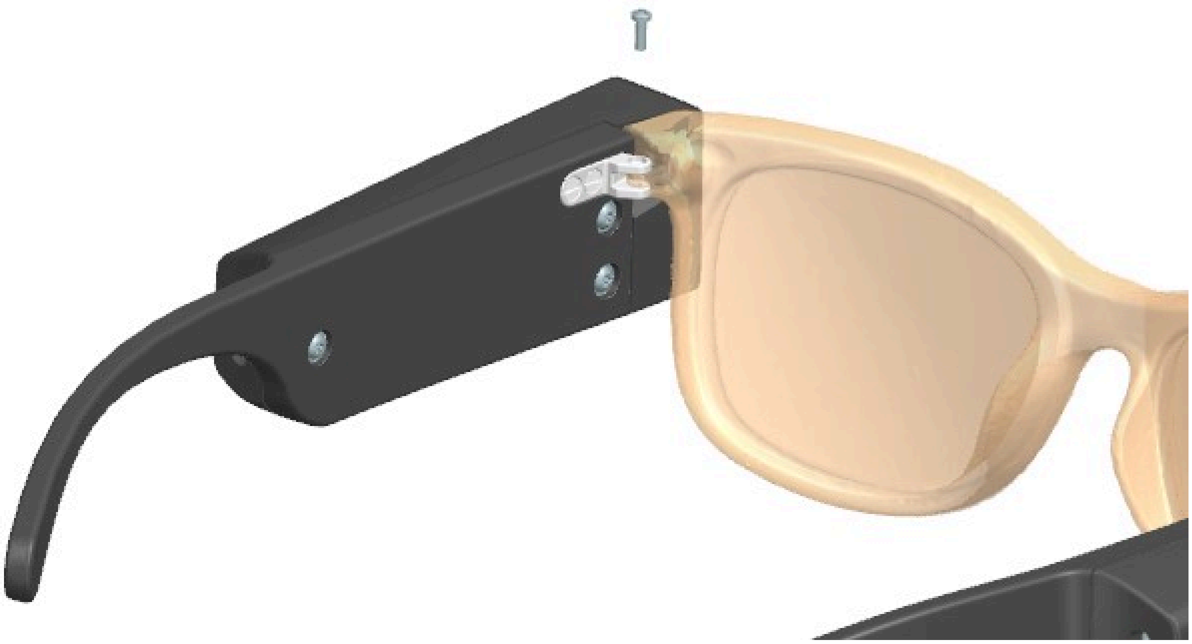
Other frames may also be paired to the VEET. For more information, contact Meta.

Accessories

The researcher is responsible for the procurement of appropriate frames, corrective lenses, and USB charging accessories, as these are not distributed with the VEET. Meta recommends procurement of a dual headed USB-C charge cords to support charging both VEET temple arms simultaneously. A charger current output of 500mA, 5V minimum per port simultaneously is sufficient for max rate charging.

Assembly

The researcher must assemble the temple arms of the VEET device onto a suitable glasses frame and corrective lenses for the participant, if applicable. To attach the VEET temple arm, the researcher removes the screw that retains the existing temple arm and uses the provided screws to replace it with the appropriate VEET temple arm. To complete the assembly, repeat the process on the opposite temple arm.



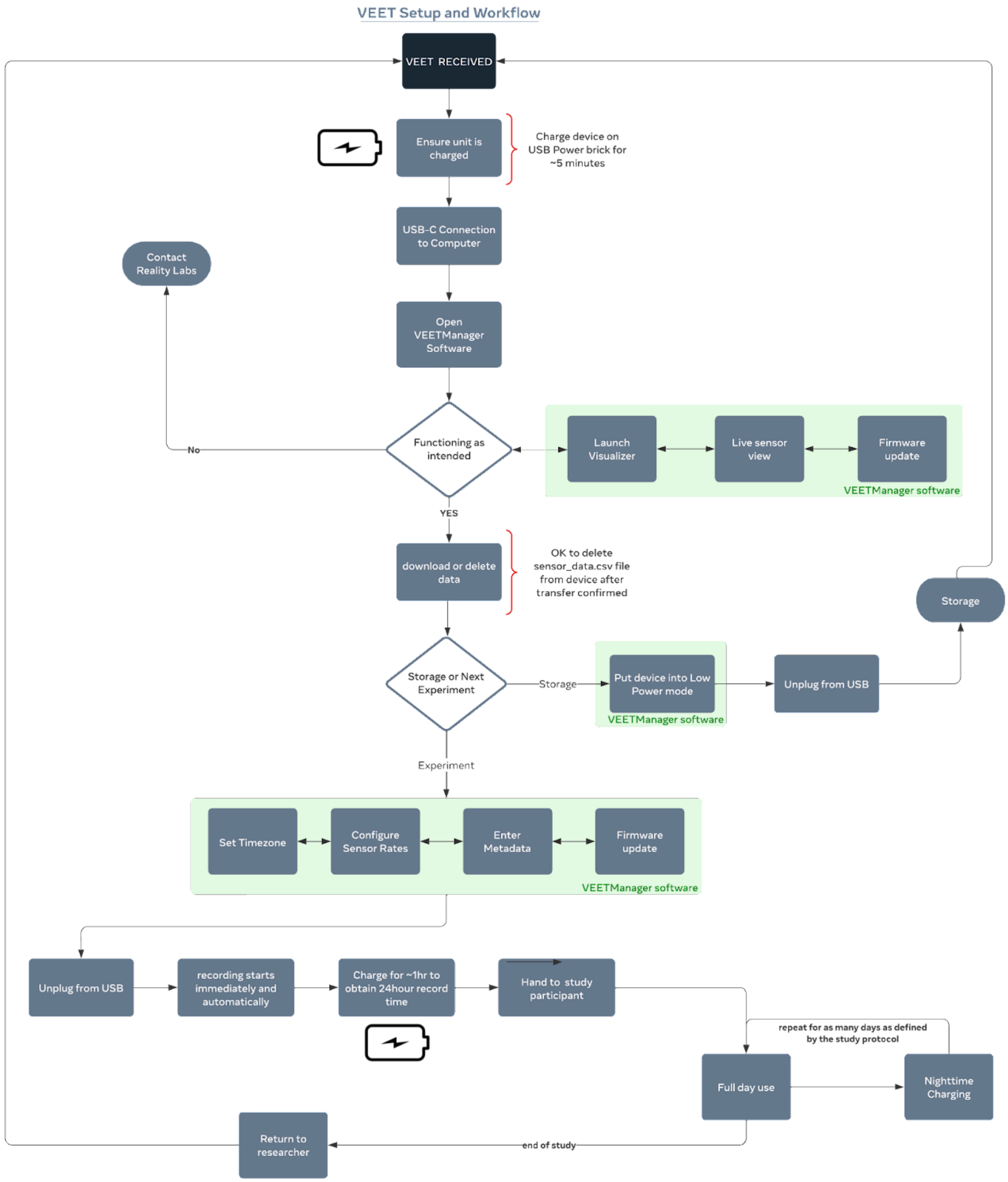
Configuration

The device will arrive with a default configuration file for ½ Hz sensing rates and with a factory calibration. Custom software (VEETManager) is required to interact with the VEET Device, including to change the device configuration. Contact Meta if you do not have a copy.

Install this software prior to connecting the VEET device to a PC. This software will manage firmware updates and packages additional help files and example analysis files. It provides live visualization of each sensor for functionality checks and provides an interface to set the configuration and metadata of the device.

Detailed description of the software interface is contained in the System Configuration section of this manual.

Getting Started with the Device



Typical VEET Workflow. *Anticipated workflow and steps required to use and recycle the VEET device through multiple studies.*

Charging

The temple arms of the VEET device arrive in either a fully discharged or deep sleep state and should be charged using a standard USB wall charger with either a USB type A or C plug. Each temple arm of the VEET device requires the power adapter to be capable of a minimum of 500 milliamp at 5 volt per temple to charge.

Due to power management constraints the device charges at a slower rate when attached to a computer. The devices should be charged on wall power for approximately 5 minutes prior to connection to a computer and using the VEETManager software interface. Each device provides an indication of charging with a red LED followed by a white LED for a few seconds when plugged in.

To obtain a full charge for full day use, the device must be charged for approximately 2 hours.

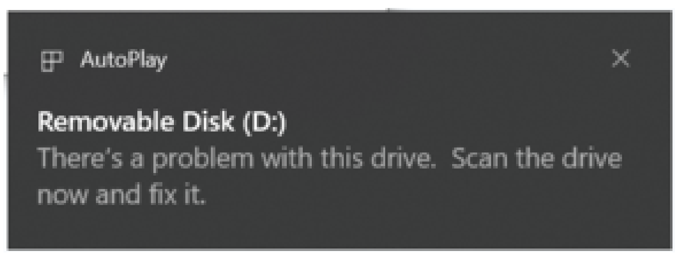
NOTE: It is normal for the temple arm of the VEET device to feel warm while charging.

Connecting to a Computer

Connect the VEET device to a PC via USB-C cable.

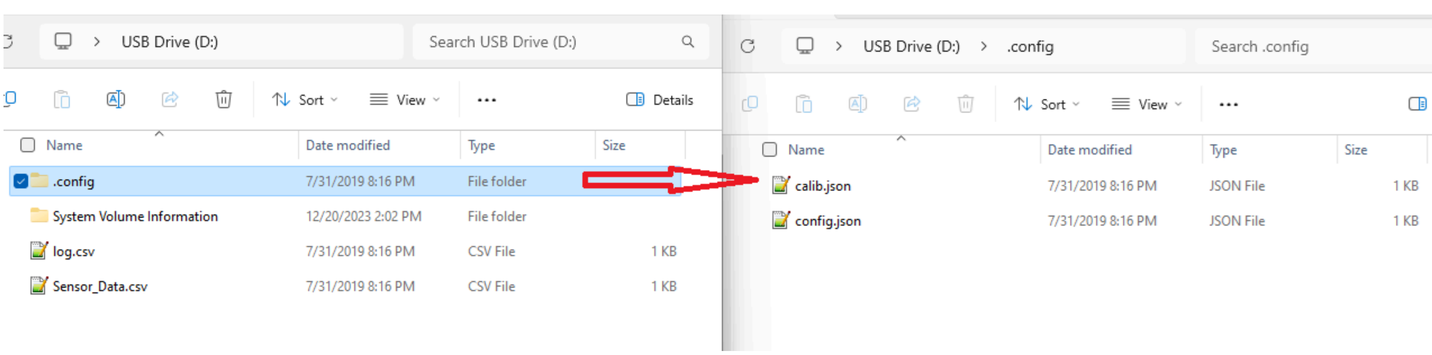
> **NOTE:** Each temple arm of the VEET device is designed to interface to Windows 10 and 11 operating systems. A temple arm of the VEET device may function on other operating systems, but it has not been extensively tested.
>
> **NOTE:** It is normal for the temple arm of the VEET device to feel warm while connected to a PC.

After connecting the cable, a Windows autoplay pop-up window may display the following image:



AutoPlay pop-up window. *This Windows Removable Disk error can be safely ignored.*

After connection, the temple arm of the VEET device mounts as a USB drive. At the initial delivery, the display in Windows File Explorer should be similar to the following image.



VEET as a USB Drive. *On Windows OS, the VEET is automatically mounted as a USB drive. Calibration and* *configuration files are contained in a hidden folder **".config"**.*

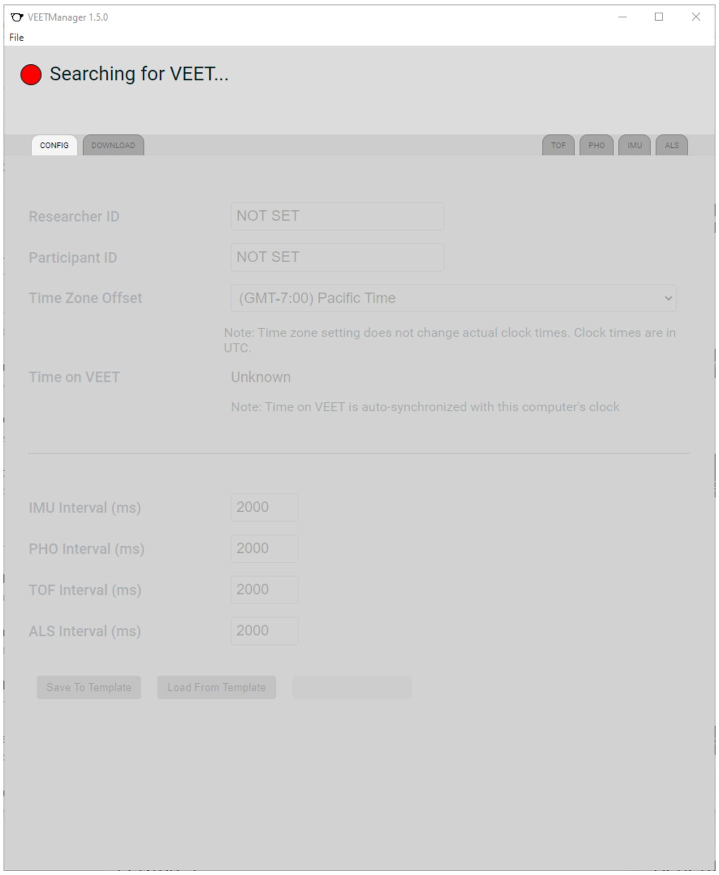
NOTE: <u>Do not delete or manipulate</u> the config.json or the calib.json files.

The log.csv file provides event logging information and will be recreated automatically if deleted.

The Sensor_Data.csv file is where all sensor data is recorded as described later in the document, it will be recreated automatically if deleted.

Starting the VEETManager Software

Extract the VEETManager software to a folder on your PC, and open VEETManager.exe.



Searching for VEET. *The VEETManager software will show the message "Searching for VEET" when it is opened.*

Once the VEETManager software finds the VEET device, it will immediately synchronize the clock on the device with your PC if it is different by more than 5 seconds.

> **NOTE:** The time on device will maintain accuracy within approximately 10 seconds of the time set over the course of a week. Timing between the temple arms is not synchronized to better than 5 seconds.

Configure the device

Configure each temple arm of the VEET device using the VEETManager software for the Windows OS.

NOTE: The VEETManager software can only work with one VEET device at a time. To avoid device conflicts, only connect one VEET device to your computer at a time.

To learn more about the configuration interface for each temple arm of the VEET device, see detailed instructions: *VEET Device Configuration*. Use the VEETManager software to update the following information for each temple arm of the VEET device.

- The sample rate of each sensor.
- The experiment specific metadata.
- The time zone offset from Coordinated Universal Time or UTC.

If the researcher is not able to configure the device, see Appendix: Troubleshoot Device.

Device States

The temple arm of the VEET device has several states that you may encounter during use with different data collection behavior.

State	Detail
Battery depleted or Fully discharged	Data is not collected. State remains until the temple arm is connected to power. Clock timing is lost upon full discharge.
Deep Sleep	Data is not collected. State for VEET device remains for greater than 1 month. Clock time is maintained during deep sleep.
Charging	Data is collected. State remains until the temple arm is disconnected from power. Data collected at configured rates.
On Battery	Data is collected. State remains for configured sensing rates for at least 24 hours. Normal wearing configuration, collects data until battery discharges to Deep Sleep.
Connected to a computer	Data is not collected. State remains until the temple arm is disconnected from the computer. Able to live view sensors, configure the device, download data, and other device maintenance. Reduced charge rate.

After each device obtains enough charge, it continuously logs data at the preloaded configuration rates until the device enters one of the following states.

- Connected to a computer.
- Battery depleted or Fully discharged.

Transitions out of Battery depleted or Fully discharged may take several minutes after being plugged into power.

Field usage

After the researcher configures each temple arm of the VEET device, the VEET device is ready to collect data and be worn by a research participant in a study. A research participant should be instructed to wear the VEET device through normal daily activities.

> **IMPORTANT:** While using the VEET device, the research participant should avoid the following activities.
> - Playing sports that require safety gear or a helmet to be worn.
> - Water-based activities or swimming that may introduce water into the VEET device.

The researcher or research participant should charge each of the temple arms of the VEET device at night, preferably uncovered in the same room in which the research participant sleeps. No other interactions are required between the research participant and the VEET device.

Since the VEET device gathers data continuously when not connected to a PC, the researcher should note the time and date the device was handed to the participant to allow for trimming the data at the end of the study.

Return from Study

After the research participant completes the study, the VEET device should be returned to the researcher and inspected for damage or modification, connecting the device to the VEETManager software will allow inspection of sensor performance post study.

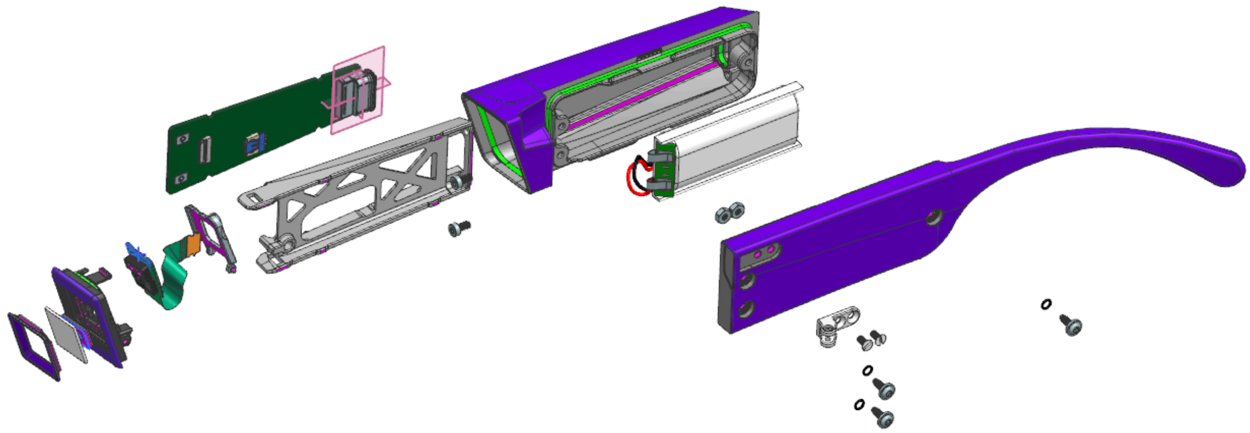
The time and date of receipt should be separately noted to allow for trimming of excess data.

The log and sensor files may become quite large, the researcher or data analyst should delete the onboard `Sensor_data.csv` and `log.csv` files between studies. The VEET device automatically recreates the `Sensor_data.csv` and `log.csv` files when logging resumes.

Note that each device will produce files with the same filename. File management strategies should be considered prior to extensive studies with the device.

Note: log files and sensor_data files should be removed from the device occasionally if the device is collecting data continuously as an in-lab device as files will get large.

System Overview



Each of the temple arms of the VEET device is a custom injection molded housing containing the following main components:

Three (3) front facing main sensors:

- Range detection
- Spectral meter
- Illumination detection

Electronics packaged along the temple arm:

- Inertial measurement sensing
- a microcontroller
- a 350 mAh battery
- 32gb flash memory storage
- a USB Type-C connection

Custom Firmware to control the device

- Clock timing in full second Unix Epoch time (seconds since January 1st, 1970)
- Sensor operation and control
- Reading configuration and calibration information from local disk
- Writing sensor data to local disk
- Event logging

Each temple arm is designed to integrate into a pair of Ray-Ban New Wayfarer junior frames with the following differences:

- The right temple arm has the sensor package aimed down 20 degrees and towards the sagittal plane by 4 degrees.
- The left temple arm has the sensor package placed perpendicular to the glasses frame.

Sensor pointing angles are an attempt to cover a range of potential gaze behaviors. The choice of these two pointing angles are arbitrary, and represent a best guess. The device designers are unaware of gaze tracking or eye tracking studies involving humans developing myopia or premyopic individuals to inform these angles. This is an opportunity for improving the device in the future to refine these static angles. Incorporating eye tracking to better inform data analysis has been considered, but has been determined to be beyond the scope of the VEET device at this time and is in conflict with the all day wearable philosophy of the device.

Privacy philosophy

The VEET device is designed with privacy and safety in mind from the outset. The VEET device supports privacy by not including cameras, microphones, or other sensors that may record personally identifiable information. The VEET device further supports privacy by only including USB communication. The VEET device does not include wireless capabilities or connectivity.

Data philosophy

The VEET device is designed to expose as much of the available information from each sensor to allow a researcher or data analyst to better understand the results and re-analyze the recorded data after data collection, if desired now or in the future. The amount of data coming from each sensor may seem excessive and will require significant post processing to develop usable summaries of the raw data that a researcher may be interested in. The data structures are described in Appendix: Organization of data: Information in detail, and should be thoroughly understood prior to post-processing.

The each temple arm of the VEET device stores the data in a single file with a common file format known as comma-separated values (csv). The researcher or data analyst accesses the data using the VEETManager software or directly as a USB drive.

Sensor descriptions

Time of Flight Sensor (TOF)

The time of flight sensor is also known as a proximity sensor or range meter. The sensor is manufactured by AMS-Osram under their part number TMF8828.

Time of Flight data is identified by the acronym TOF in the VEET device output.

The time of flight sensor illuminates an environment with infrared light and measures the time for the light to return from reflected objects. The time of flight sensor repeats the measurement many thousands of times, based on a specified number of iterations.

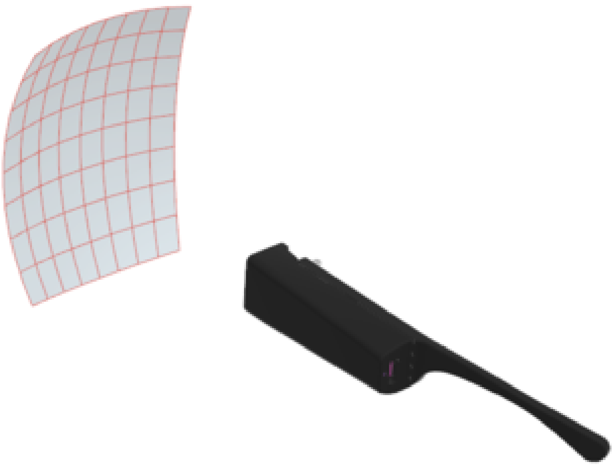
> **NOTE**: The VEET Device iterations are pre-set to 125,000 samples, taking approximately 0.125 seconds. These settings are not configurable by the end user.

The time of flight sensor converts the average return times to a value for `Distance` and the signal-to-noise ratio to a value for `Confidence`.

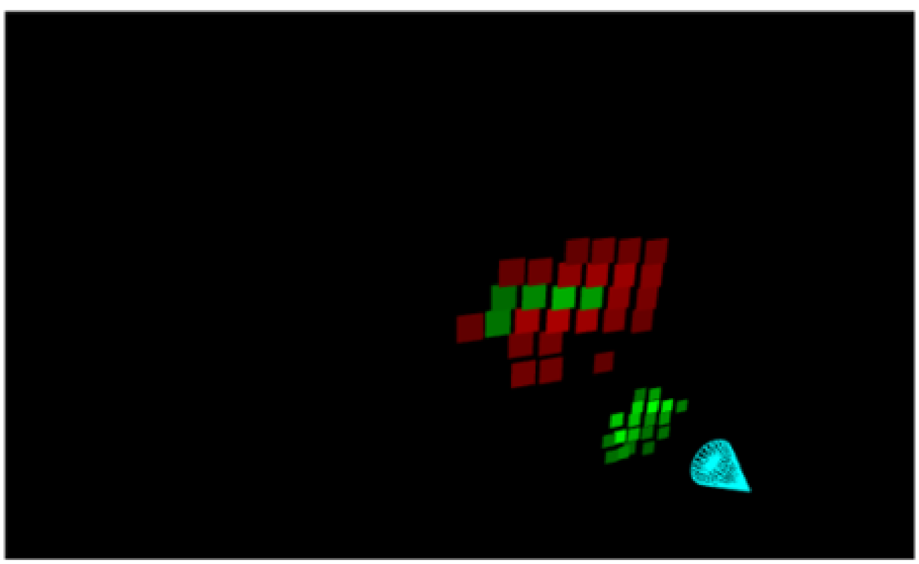
The time of flight sensor reads multiple zones at the same time. The zones are arrayed in an 8 x 8 angular grid with a 41 degree x 52 degree FOV (Approximately `-26` to `+26` degrees vertically and `-20.5` to `+20.5` degrees horizontally). The center point of the array can be assumed to be in line with the temple pointing angle (The right temple arm aimed down 20 degrees and towards the sagittal plane by 4 degrees while the left temple arm is aimed perpendicular to the glasses frame).

> **NOTE**: The value of the `Distance` for a zone is the true distance from the time of flight sensor and is not corrected to a flat plane.

For each zone, the time of flight sensor detects and records the first 2 objects. For an 8x8 angular array, the time of flight sensor reports 2 objects for each of the 64 zones. For an 8x8 angular array, this produces a total of 128 pairs of values for `Distance` and `Confidence`. If the time of flight sensor detects no object in a zone, the value of the `Distance` and `Confidence` for that zone is set to `0`.



Zone pattern. *Zone pattern for the TOF sensor.*



Object 1	
Confidence	43%
Distance	429 mm
Object 2	
Confidence	26%
Distance	753 mm

Typical time of flight report. *The VEETManager software displays TOF with low confidence shown darker.*

Factors that may affect Distance measurements

The time of flight sensor may detect a transparent glass surface as well as an object beyond the transparent glass.

The time of flight sensor may detect the value of the `Distance` to a reflected object as well as the value of the `Distance` to the surface that reflects the object.

Factors that may affect Confidence

Bright infrared light from other sources will reduce the value of the `Confidence`. A common source is strong direct sunlight.

Zones at the outer edges of the sensor field of view have lower `Confidence.`
Smaller objects report lower Confidence as compared to larger objects.
Less reflective materials (in the infrared band) report lower `Confidence`.

The time of flight sensor may report a `Confidence` and `Distance` value in empty zones next to a zone with objects. The researcher should filter the minimum threshold for the value of `Confidence` to remove false detections.

For more details about the time of flight sensor, navigate to the Appendix: Sensor package, Time of flight sensor: TMF8828 section.

For detailed information about how to interpret the sensor output data, navigate to the VEET Sensor Post Processing Appendix Appendix: Sensor post processing.

Spectral Sensor (PHO)

The Spectral sensor is an AMS-Osram AS7341 with an 11-channel multi-spectral sensor for color detection and spectral analysis. The spectral response is defined in the wavelengths from approximately 350nm to 1000nm.

Spectral Sensor data is identified by the acronym PHO in the VEET device output.

The Spectral sensor uses 8 narrow channels plus 4 broad channels to measure the amount of light. The Spectral sensor is paired with a diffusion film to balance the light dispersion on the individual channels and widen the field of view.

Each channel of the Spectral Sensor is a photodiode, the current is related to the amount of photons interacting with the sensor. The components read the amount in an analog to digital conversion (ADC).

The Spectral sensor sets gain and measurement time to record raw counts. The raw counts are correlated to the amount of irradiance on the sensor. The device outputs the `Integration Time`, `Gain`, and for each channel the `Raw Counts.`

The value of `Basic Counts` will be the basis for predicting spectral power distribution (SPD) when post processing the data.

NOTE: Estimating the SPD has not been fully defined and development of these methods is in process

An individual channel is not perfectly filtered and the energy estimate compares the current channel with the other channels to predict the spectral distribution.

For detailed information about how to interpret the sensor output data, navigate to the VEET Sensor Post

Processing Appendix [Appendix: Sensor post processing](#).

For more details about the spectral meter sensor, navigate to the Appendix: Sensor package, Spectral meter sensor: AS7341 section.

Ambient Light Sensor (ALS)

The Ambient Light Sensor is an AMS-OSRAM TSL2585 with UVA and light flicker detection
The main purpose for the ALS sensor is to provide photometric data in the form of lux.
The Ambient Light sensor is also known as an Illumination sensor.

The Ambient Light sensor data is identified by the acronym ALS in the VEET device output.

The Ambient Light sensor uses 3 broad channels to gather information about the environment.

> Ultraviolet (UV)
> Visible (Photopic)
> Infrared (IR)

The Ambient Light Sensor is paired with a diffusing film to balance the light dispersion on the individual channels and widen the field of view.

Each channel of the Ambient Light Sensor is a photodiode, the current related to the amount of photons interacting with the sensor. The components reads the amount in an analog to digital conversion (ADC)

The Ambient Light sensor sets gain and measurement time to record raw counts. The raw counts are correlated to the amount of irradiance on the Ambient Light sensor. The Ambient Light sensor outputs the values of the `Integration`, `Gain`, and `Raw Counts`. The Ambient Light sensor outputs the values of the `Flicker` and the VEET device calculates `Lux`.

Each individual channel is not perfectly filtered. To estimate Lux the internal firmware utilizes a custom algorithm to compare the ratio of visible to infrared light and estimate lux under a variety of lighting conditions.

For detailed information about how to interpret the sensor output data, navigate to the VEET Sensor Post Processing Appendix [Appendix: Sensor post processing](#)..

For more details about the Ambient Light Sensor, navigate to the Appendix: Sensor package, Light meter sensor: TSL2585 section.

Inertial Measurement Unit Sensor (IMU)

The Inertial Measurement Unit is a Bosch BMI270, with a three-axis accelerometer and three-axis gyroscope combined to provide information about movement and orientation.

The Inertial Measurement Unit sensor data is identified by the acronym IMU in the VEET device output.

The IMUs are housed in the side of each temple arm and are pointed in the same orientation and do not need to be adjusted for the pointing angle of the other sensors packaged in the VEET device. The IMU records acceleration and angular velocity on the device at the moment it is sampled

> Acceleration in the X, Y, and Z direction in `aX`, `aY`, and `aZ`.
> Rotation rate about the X, Y, and Z axis in `gX`, `gY`, and `gZ`.

The IMU stores the internal temperature in degrees C.

For more details about the inertial measurement unit, navigate to the Appendix: Sensor package: Inertial measurement unit: BST-BMI270 section.

VEET Device Operation

The VEET device operates (collecting and storing data) at all times when battery reserves are sufficient to operate the system, memory is available, and the device is disconnected from a computer. Each temple arm of the VEET device operates as a separate unit, with the 4 main sensing devices controlled by a microcontroller with onboard battery and USB accessible storage. Interaction with the device data and configuration is solely through the USB Type-C interface on the computer and is only intended for access by a researcher and data analyst.

Research participant interaction

Each temple arm of the VEET device has no buttons or other controls designed to be accessed by a research participant. The research participant is only required to charge each temple arm of the VEET device at night. To charge each temple arm of the VEET device, use any USB Type-C cable to connect to a USB wall charger. Each temple arm of the VEET device must be independently connected to power to recharge.

An LED is available to indicate charge and logging resume and provides a momentary indication of the start of charge and logging. The LED does not remain on continuously during charging to prevent the device from self contaminating the luminance and spectral data in low light conditions.

If a participant forgets to charge the device or otherwise lets the battery rundown data logging will cease until the next time the device is plugged into power to charge. The clock will be maintained for greater than 1 month after the device goes into its low power mode. The end user will not be notified if the battery power is lost.

VEET Device configuration and use

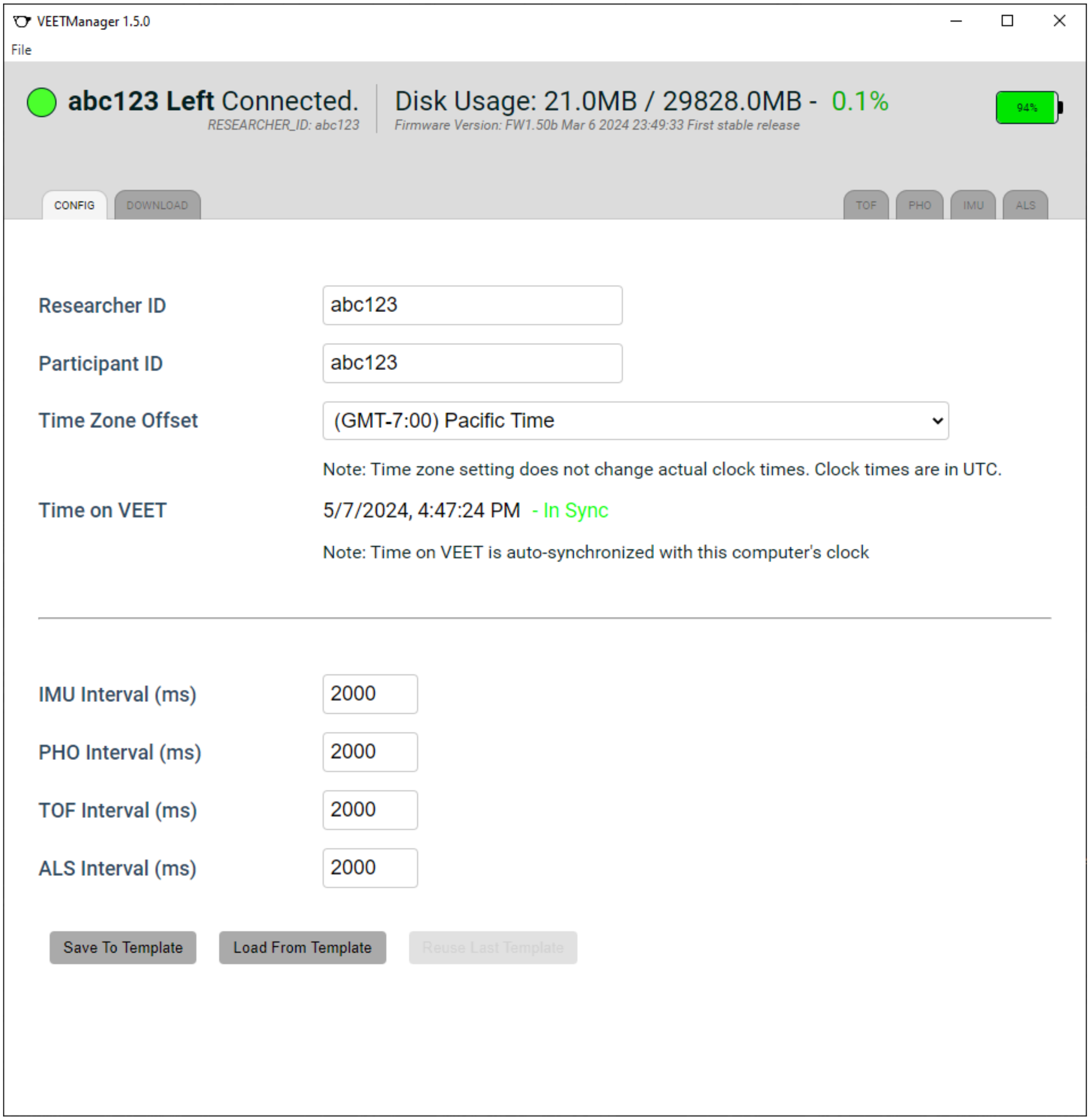
The researcher configures each temple arm of the VEET device using one of the following methods:

VEETManager software
Direct edit the configuration file, config.json

- This method is only recommended for experienced users of the VEET Device and is not described in detail herein. For more information please contact Meta.

Detailed steps to configure the device for use:

- Step 1: Open the VEETManager software.
- Step 2: Plug one temple arm of the VEET into your computer.
    - Confirm that the VEETManager discovers the device and shows an active screen, similar to as shown below
- Step 3: go to the "config" tab of the VEETManager as shown below
- Step 4: Enter the researcher ID.
    - This is a text field
- Step 5: Enter the Participant ID.
    - This is a text field.
    - The intention is to allow connection between the device output and a specific participant without storing any personal identification information on device
- Step 6: Select the timezone offset from the drop down menu.
    - This will be recorded on the INF line as a number (+ or -) in hours from UTC.
    - Time on device will be in Unix Epoch time, this number will allow post processing to adjust the time to match participant's local time.
- Step 7: Confirm time on veet matches the local time.
    - This is automatically set everytime the device is connected to a PC and the VEETManager running and requires the setup PC to have accurate time
- Step 8: Set the recording interval for each sensor.
    - The default interval is 2 seconds (2000ms) for all sensors.
    - For the best battery performance all sensors should be set to the same interval.
    - The minimum time for each interval is also 2 seconds (2000ms).
- Step 9: Unplug this half of the VEET
    - Data recording starts within a few seconds of unplug
- Step 10: Plug in the other half of the VEET and repeat steps 1-9.
- OPTIONAL: for expediency when configuring many devices the "Save to Template" button will save the current interval configuration settings to a local .json file and can be copied to any other device with the "load from Template" or, logically, the "Reuse Last Template" button.
    - Researcher ID, Participant ID and timezone offset will still need to be adjusted

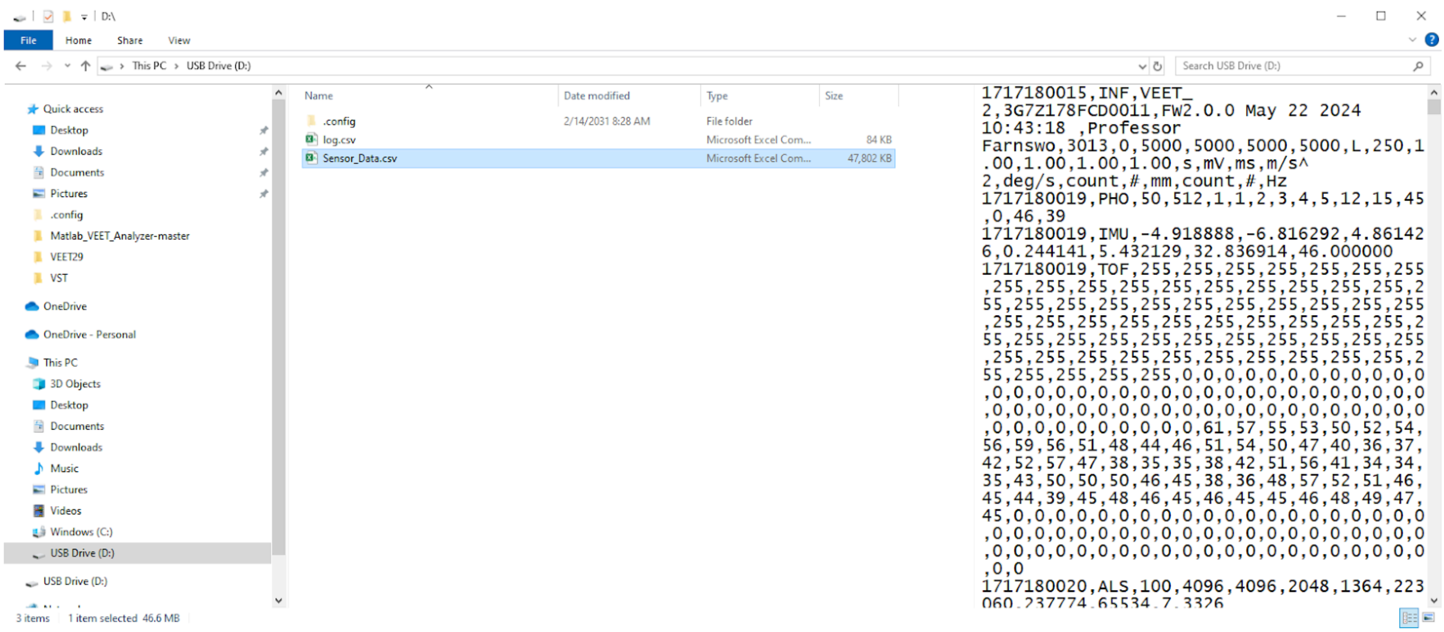


If the power is not restored within approximately 1 month the battery will fully deplete and the clock resets to the default time. The default time begins at 2012-01-01 00:00:00 UTC, which is `1325376000` in Epoch units.

If the memory in a temple arm of the VEET device is full, data is blocked from an overwrite and the log does not write to the file. the memory in the VEET temple arm is approximately 32gb, at maximum data rates this will store more than 1 year's worth of continuous data. We do not recommend attempting to deal with transferring or managing this much data.

Onboard files

Four key files are stored on each temple arm of the VEET device. No other files should be present on the device and should be deleted if present. Calibration and Configuration files are located in a hidden folder, .config, and are not intended to be manipulated directly.



USB Drive on Windows OS

USB Drive on Windows OS

file	**detail**
`calib.json` json	The calibration values of the temple arm. Located in the hidden folder .config
`config.json`	The values are exposed in a plain text json. Normally set using the VEETManager software. Located in the hidden folder .config
`log.csv` csv | table	The log of system events.
`Sensor_data.csv` csv | table	The log of full sensor events.

The log files may be quite large, the researcher or data analyst should delete the onboard `Sensor_data.csv` and `log.csv` files between studies. The temple arm of the VEET device automatically recreates the `Sensor_data.csv` and `log.csv` files when logging resumes.

VEETManager Software

The VEETManager software is a convenient interface to the VEET device, enabling sensor live view, configuration of the device, firmware and calibration updates, convenient links to documentation and ways to

get help from Meta, if needed.
All of the features that the software implements can be independently executed via serial commands or direct edit of device files. Experienced users can find out more about the available commands in Appendix xyx <INSERT LINK>.

The software has three (3) main functions separated into appropriate tabs for easy navigation.

- Configuration
    - configure the device with sensor rates and metadata
- Sensor live view
    - individually view the 4 main sensors sampled several times per second
- Developer tools
    - view log files
    - view and update calibration files

In addition to these functions the software will automatically detect the firmware version on device and if a new version exists request the user update. These firmware updates are data safe and will not erase or modify any files on the VEET device disk. The software will also determine if there is a new version of itself and prompt the user to download

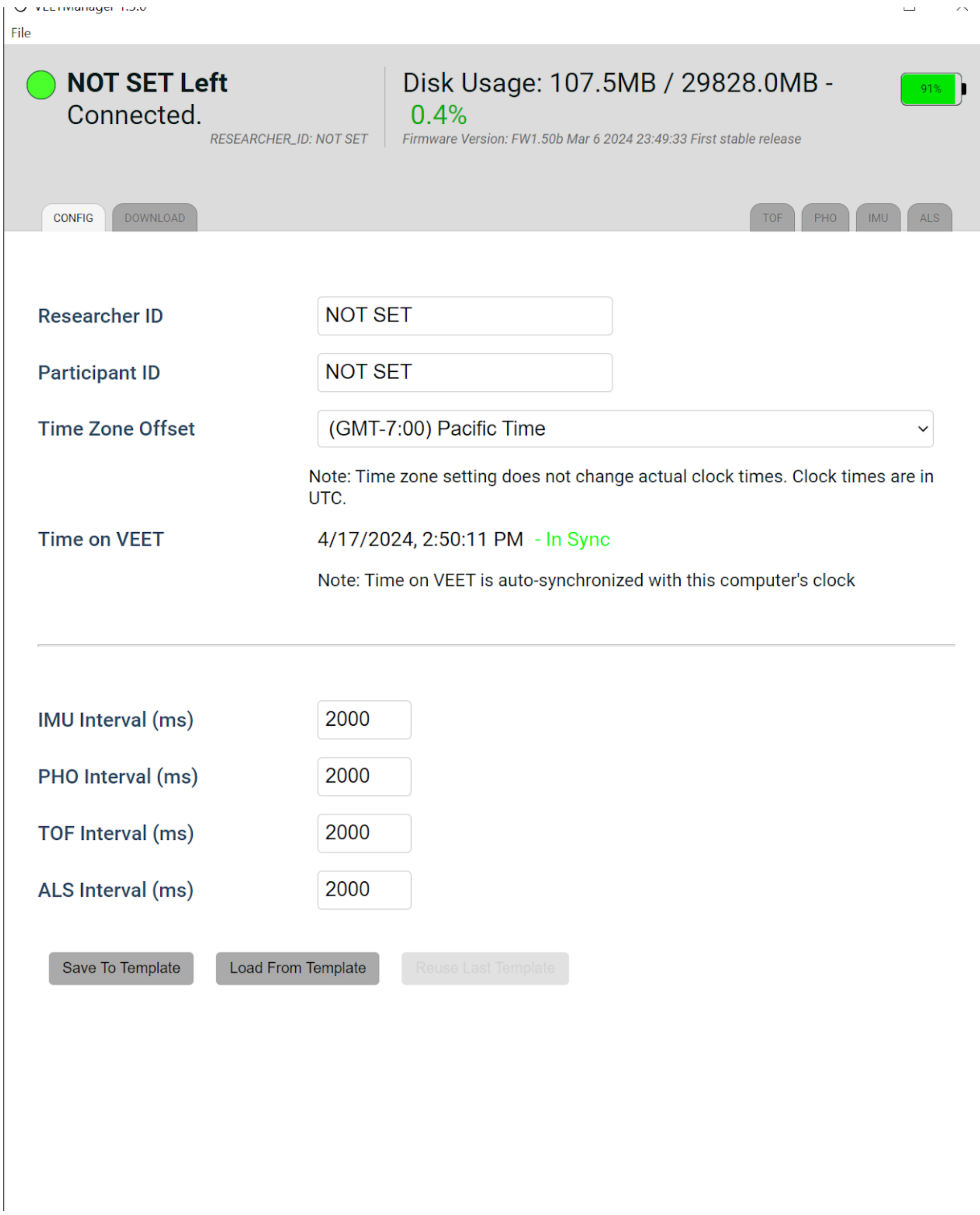
The help menu of the VEETManager contains links to documentation.

Known issues:

During power surges or other power disruptions connection to the VEET device may be disconnected, if this occurs unplug the VEET device for 5 seconds and reconnect the device to the computer.

Occasionally a java script error may occur, close the VEETManager software and restart the program

Connect VEET device to VEETManager software



VEETManager software v1.50 interface connected to the right temple arm of a VEET device

VEETManager software v1.50 interface connected to the right temple arm of a VEET device

1. Ensure the temple arm of the VEET device is connected to power for more than one minute. If the battery of the temple arm of the VEET device is fully depleted, you may need to leave it on the charger for 1 hour.

2. Run the VEETManager software, `VEETManager.exe`
3. Plug the temple arm of the VEET device into the computer
4. Options:
    1. Set Configuration
    2. Download data
    3. Load calibration file

Configurable options using VEETManager software

Time zone offset from UTC

Researcher ID

Participant ID

Sensor read interval in milliseconds (ms)

- IMU
- PHO
- TOF
- ALS

Appendix: Command-line interface

Advanced interface for the VEET Device utilizing a serial terminal. If interested in utilizing this interface please contact Meta for further guidance.

Settings for generic serial terminal software

```
NOTE: Baud Rate, parity and handshaking do not matter since this is a virtual comm port, and those settings correlate to physical characteristics of a UART.
Local Echo: On
New-Line Transmit: CR (Carriage Return)
```

command	**detail**
`??`	Console Help. Display console command list.
`BL`	Enter the bootloader.
`FT`	Format FS.
`GB`	Print Battery in millivolts (mV).
`GM`	Print Mount Side.
`GS`	Print Serial Number.

command	**detail**
GT	Print RTC Time in seconds (s). May print more than one RTC time.
LG	Print Log.
PD	Power Down (Battery Disconnect).
PL	Power Low (Backup Mode) Set Transport Mode (RTC Preserved)
TM	Set Transport Mode (RTC Preserved)
RE	Device reset (SW).
RW	Device reset (WDT).
S0	Sample IMU. Sample from BMI270 component.
S1	Sample PHO. Sample from AS7341 component.
S2	Sample TOF. Sample from TMF8828 component.
S3	Sample ALS. Sample from TSL2585 component.
SM<X>	Set Mount Side. Possible value for <X>: L or R
SN<X>	Set Serial Number. Possible values for <X>: maximum of 16 characters
ST<X>	Set RTC Time. Possible values for <X>: epoch time in seconds
LW<X>	Set White LED intensity. Possible values for <X>: 0-1000
LR<X>	Set Red LED intensity. Possible values for <X>: 0-1000
VR	Print firmware version.

Appendix: Definitions

acronym	**detail**
INF	Information
IMU	IMU sensor data
ALS	Ambient light sensor data
PHO	Multi-Spectral digital sensor data
TOF	Time of Flight sensor data
LOG	Log message
BAT	Battery voltage
FWHM	Full-Width at Half-Maximum
VEET	Visual Environment Evaluation Tool
Gb	Gigabit, 2^20 bits, 8 bits per byte.
GB	GigaByte, 2^30 Bytes. 1 GB = 8 Gb
mAh VCP	milliAmp-hour Virtual Comm Port

acronym	detail
CDC UTC UV RTC	USB Communications Device Class Coordinated Universal Time UltraViolet Real-time Clock
INT	
uint32	
float	
int_16	
FW	
CSV	
MSD	
UART	
USB	
IRB	
VEET	
Counts, Sensor	counts measured prior to adjustment by calibration
Counts, Raw	counts adjusted for unit to unit variability, with no adjustment for gain or integration time
Counts, Normalized	counts normalized to single gain scale, with no adjustment for integration time
Counts, Basic	counts after normalizing gain factor and unifying integration time.

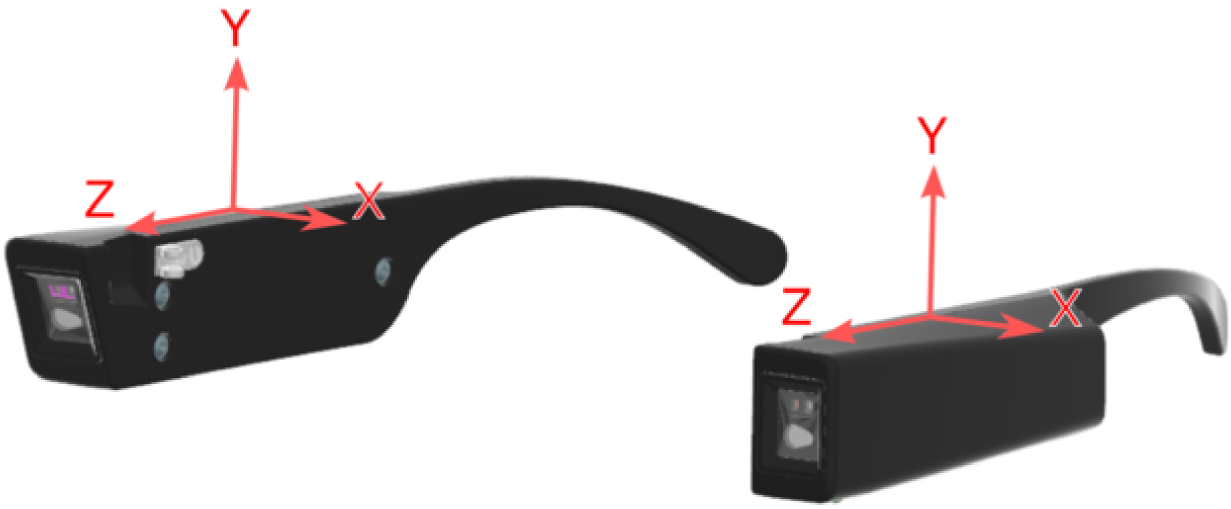
Appendix: Organization of data: Inertial Measurement Unit

An inertial measurement unit or `IMU` line reports the data from the IMU sensor.

`Timestamp,IMU,ax,ay,az,gx,gy,gz,temp`

Item	detail
`time_stamp`	The epoch time since 1970-01-01 00:00:00 UTC in seconds. Possible values: `1325376002` to `1893484800+` Data type: `uint32`
`IMU`	The current line is inertial measurement unit sensor data. Only value: `IMU` Data type: `string`
`ax`	The position of the X-axis accelerometer in meters per second-squared. Possible values: `-4` to `+4` Data type: `float`
`ay`	The position of the Y-axis accelerometer in meters per second-squared. Possible values: `-4` to `+4` Data type: `float`
`az`	The position of the Z-axis accelerometer in meters per second-squared. Possible values: `-4` to `+4` Data type: `float`
`gx`	The angle of the X-axis gyroscope in degrees per second. Possible values: `0` to `2000` Data type: `float`

Item	detail
`gy`	The angle of the Y-axis gyroscope in degrees per second. Possible values: `0` to `2000` Data type: `float`
`gz`	The angle of the Z-axis gyroscope in degrees per second. Possible values: `0` to `2000` Data type: `float`
`temp`	The temperature sent from IMU sensor in degrees Celsius. Possible values: `-41` to `+87` Data type: `int_16`



Orientation of IMU axis

Orientation of IMU axis

Appendix: Organization of data: Information

An information or `INF` line reports the system state at the time of capture. An `INF` line is generated when the system restarts or switches states, including the following conditions.

- Plug and unplug events for charge
- Connected to a computer
- Waking up after deep sleep
- Manual restarts, including a configuration update
- After a firmware update

```
[time_stamp],INF,[product_name],[serial_number],[fw_version],[Researcher_ID],[Participant_
ID],[Time_Zone_offset],[IMU_interval],[PHO_interval],[TOF_interval],[ALS_interval],[Temple
_config],[ToF_Iterations],[IMU_Cal_Table],[PHO_Cal_Table],[ToF_Cal_Table],[ALS_Cal_Table],
[Unit_Timestamp],[Unit_Batt_Voltage],[Unit_Sensor_Interval],[Unit_IMU_Accel],[Unit_IMU_Gyr
o],[Unit_Pho_Cts],[Unit_Pho_Gain],[Unit_ToF],[Unit_ALS_Cts],[Unit_ALS_Gain],[Unit_ALS_Flic
ker]
```

item	detail
`time_stamp`	The epoch time since 1970-01-01 00:00:00 UTC in seconds. Possible values: `1325376002` to `1893484800+` Data type: `uint32`
`INF`	The current line is information data. Only value: `INF` Data type: `string`
`product_name`	The hardware version of the temple arm of the VEET device. Possible values: `VEET_1.0`, `VEET_1.5` Data type: `string`
`serial_number`	The unique identifier of the temple arm of the VEET device. Data type: `string`
`fw_version`	The build version of the firmware of the temple arm of the VEET device. Possible values: `FW1.02`, `FW1.5` Data type: `string`
`Researcher_ID`	The identifier of the researcher.
`Participant_ID`	The identifier of the research participant.
`Time_Zone_offset`	The time zone from UTC of the temple arm of the VEET device.
`IMU_interval`	The time between sensor reads in milliseconds.
`PHO_interval`	The time between sensor reads in milliseconds.
`TOF_interval`	The time between sensor reads in milliseconds.
`ALS_interval`	The time between sensor reads in milliseconds.
`Temple_Config`	Indicates the temple arm is left or right.
`TOF Iterations`	The number of ToF iterations per read in thousands.
`IMU_Cal_Table`	The version of the IMU calibration table. Data type: `integer`
`PHO_Cal_Table`	The version of the PHO calibration table. Data type: `integer`
`ToF_Cal_Table`	The version of the ToF calibration table. Data type: `integer`
`ALS_Cal_Table`	The version of the ALS calibration table. Data type: `integer`
`Unit_Timestamp`	The units of the timestamps. Only value: `s` Data type: `string`
`Unit_Batt_Voltage`	The units of the battery voltage.
`Unit_Sensor_Interval`	The units of the Sensor read interval. Only value: `ms` Data type: `string`
`Unit_IMU_Accel`	The units of the IMU accelerometer. Value: `m/s^2` Data type: `string`
`Unit_IMU_Gyro`	The units of the IMU gyroscope. Value: `deg/s` Data type: `string`
`Unit_Pho_Cts`	The units of the Spectral raw irradiance. Value: `count` Data type: `string`

item	detail
`Unit_Pho_Gain`	The units of the Spectral gain index. Value: `#` Data type: `string`
`Unit_ToF`	The units of the ToF distance. Value: `mm` Data type: `string`
`Unit_ALS_Cts`	The units of the ALS raw irradiance. Value: `count` Data type: `string`
`Unit_ALS_Gain`	The units of the ALS gain index. Value: `#` Data type: `string`
`Unit_ALS_Flicker`	The units of the ALS flicker. Value: `Hz` Data type: `string`

NOTE: More information may be included.

Appendix: Organization of data: Ambient Light Sensor

A light meter or `ALS` line reports the data from the ALS sensor.

`Timestamp,ALS,Integration,uvGain,visGAin,IRGain,uvValue,visValue,irValue,Flicker,Lux`

item	detail
`time_stamp`	The epoch time since 1970-01-01 00:00:00 UTC in seconds. Possible values: `1325376002` to `1893484800+` Data type: `uint32`
`ALS`	The current line is light meter sensor data. Only value: `ALS` Data type: `string`
`Integration`	The length of time for the measurement by the sensor in milliseconds. Analogous to shutter speed. Only value: `100` Data type: `uint16`
`uvGain`	The index of the gain used by the UV channel at time of reading. References to a gain table.
`visGain`	The index of the gain used by the visible channel at time of reading. References to a gain table.
`irGain`	The index of the gain used by the near IR channel at time of reading. References to a gain table.
`uvValue`	The response* in raw counts of the UV channel. Possible values: `0` to `1048575` Data type: `float`
`visValue`	The response* in raw counts of the visible channel. Possible values: `0` to `1048575` Data type: `float`
`irValue`	The response* in raw counts of the near IR channel. Possible values: `0` to `1048575` Data type: `float`

item	detail
`Flicker`	The predominant flicker rate of light during sensor read in hertz.
`Lux`	The estimated function of visible and IR channels in lux.

NOTE: *Response is provided in Raw Counts, which is directly related to the number of photons received by the photodiode sensor. Basic Counts must be post-processed, and are calculated as a function of channel Raw Counts, Integration Time, and Gain, and are the basis of measuring light exposure, see Appendix: Sensor Post Processing . For a given light environment irradiance can be approximated, however as the channels are not ideal they cannot be related to irradiance independent of each other nor independent of light type.

Appendix: Organization of data: Spectral Sensor

The Spectral Sensor or `PHO` line reports the data from the Spectral sensor.

```
Timestamp,PHO,Integration, gain,<sensors[0:11]>
```

where `sensors[0:11]` are

```
'415', '445', '480', '515', '555', '590', '630', '680', 'IR', 'Dark', 'ClearL', 'ClearR'
```

item	detail
`time_stamp`	The epoch time since 1970-01-01 00:00:00 UTC in seconds. Possible values: `1325376002` to `1893484800+` Data type: `uint32`
`PHO`	The current line is spectral sensor data. Only value: `PHO` Data type: `string`
`Integration`	The length of time for the measurement by the sensor in milliseconds. Analogous to shutter speed. Only value: `50` Data type: `uint16`
`Gain`	The index of the gain used by the sensor at time of reading. References to a gain table.
`415`	The response* in raw counts, centered on nominal wavelength of 415nm and bandwidth of 26nm FWHM. Value Range: `0` to `65535`, Data type: `uint16`
`445`	The response* in raw counts, centered on nominal wavelength of 445nm and bandwidth of 30nm FWHM. Value Range: `0` to `65535`, Data type: `uint16`
`480`	The response* in raw counts, centered on nominal wavelength of 480nm and bandwidth of 36nm FWHM. Value Range: `0` to `65535`, Data type: `uint16`
`515`	The response* in raw counts, centered on nominal wavelength of 515nm and bandwidth

item	detail
	of 39nm FWHM. Value Range: `0` to `65535`, Data type: `uint16`
`555`	The response* in raw counts, centered on nominal wavelength of 555nm and bandwidth of 39nm FWHM. Value Range: `0` to `65535`, Data type: `uint16`
`590`	The response* in raw counts, centered on nominal wavelength of 590nm and bandwidth of 40nm FWHM. Value Range: `0` to `65535`, Data type: `uint16`
`630`	The response* in raw counts, centered on nominal wavelength of 630nm and bandwidth of 50nm FWHM. Value Range: `0` to `65535`, Data type: `uint16`
`680`	The response* in raw counts, centered on nominal wavelength of 680nm and bandwidth of 52nm FWHM. Value Range: `0` to `65535`, Data type: `uint16`
`IR`	The response* centered on nominal wavelength of 910nm and bandwidth of 104nm FWHM in raw counts. Value Range: `0` to `65535`, Data type: `uint16`
`Dark`	The response* on the masked channel in raw counts. Used to measure noise level. Value Range: `0` to `65535`, Data type: `uint16`
`ClearL`	The response* at full spectrum in raw counts. Value Range: `0` to `65535`, Data type: `uint16`
`ClearR`	The response* at full spectrum in raw counts. Value Range: `0` to `65535`, Data type: `uint16`

NOTE: *Response is provided in Raw Counts, which is directly related to the number of photons received by the photodiode sensor. Basic Counts must be post-processed, and are calculated as a function of channel Raw Counts, Integration Time, and Gain, and are the basis of measuring light exposure, see Appendix: Sensor Post Processing . For a given light environment irradiance can be approximated, however as the channels are not ideal they cannot be related to irradiance independent of each other nor independent of light type.

Appendix: Organization of data: Time of Flight

A time of flight or `TOF` line reports the data from the ToF sensor.

8x8 configuration

```
Timestamp,TOF, object 1 confidences[0:63], object 2 confidences[0:63], object 1
distances[0:63], object 2 distances[0:63]
```

From the point of view of the research participant looking out, the time of flight (TOF) data is organized as an 8x8 row-major array with top left = `(0,0)`. next data = `(0,1)` (same row, next column). `(1,0)` = second row, first column. lower right corner is `(7,7)`.

item	**detail**
`time_stamp`	The epoch time since 1970-01-01 00:00:00 UTC in seconds. Possible values: `1325376002` to `1893484800+` Data type: `uint32`
`TOF`	The current line is time of flight sensor data. Only value: `TOF` Data type: `string`
`conf1[0:63]`	The signal-to-noise ratio for the 1st object in the zone. Possible values: `0` to `255` Data type: `uint8`
`conf2[0:63]`	The signal-to-noise ratio for the 2nd object in the zone. Possible values: `0` to `255` Data type: `uint8`
`dist1[0:63]`	The distance to 1st object in the zone in millimeters. Possible values: `0` to `5000` Data type: `uint16`
`dist2[0:63]`	The distance to 2nd object in the zone in millimeters. Possible values: `0` to `5000` Data type: `uint16`

The `Data[0:63]` values are organized as a raster scan with `[row, column] [0,0]` in upper left. Orientation is facing away from the sensor (`0,0`: upper left; `7,7`: lower right).

	column 0	**column 1**	**column 2**	**column 3**	**column 4**	**column 5**	**column 6**	**column 7**
row 0	0	1	2	3	4	5	6	7
row 1	8	9	10	11	12	13	14	15
row 2	16	17	18	19	20	21	22	23
row 3	24	25	26	27	28	29	30	31
row 4	32	33	34	35	36	37	38	39
row 5	40	41	42	43	44	45	46	47
row 6	48	49	50	51	52	53	54	55
row 7	56	57	58	59	60	61	62	63

For detailed information about formula to reconstruct a grid, navigate to the VEET Time of Flight Interpretation Example gSheet file.

index of data group	
Line_Start	Cell Location with line start
Offset Distance 1	130
Offset Confidence 1	2
Offset Distance 2	194
Offset Confidence 2	66
Confidence Range	255
item	**formula**

index of data group	
Time	`(((Line_Start/60)/60)/24)+DATE(1970,1,1)`

Each zone

item	formula
Distance 1	`OFFSET(Line_Start,0,([Zone#]+(Offset_Dist1)))`
Confidence 1	`OFFSET(Line_Start,0,(G7+(Offset_Conf1)))/Range_Conf`
Distance 2	`OFFSET(Line_Start,0,([Zone#]+(Offset_Dist2)))`
Confidence 2	`OFFSET(Line_Start,0,(B12+(Offset_Conf2)))/Range_Conf`

Appendix: Onboard file example: calib json


```
{
    "deviceID":"0026L",
    "PHO":{
        "F415":{"gain":1.1458,"off":0.0},
        "F445":{"gain":1.0938,"off":0.0},
        "F480":{"gain":1.155,"off":0.0},
        "F515":{"gain":1.1096,"off":0.0},
        "F555":{"gain":1.1043,"off":0.0},
        "F590":{"gain":1.0996,"off":0.0},
        "F630":{"gain":1.094,"off":0.0},
        "F680":{"gain":1.1211,"off":0.0},
        "F910":{"gain":1.0623,"off":0.0},
        "Fclear1":{"gain":1.2144,"off":0.0},
        "Fclear2":{"gain":1.1251,"off":0.0}
    },
    "ALS":{
        "Fuv":{"gain":0.9629,"off":0.0},
        "Fpho":{"gain":1.0258,"off":0.0},
        "Fir":{"gain":0.9913,"off":0.0}
    },
    "LUX":{
        "IR_PHO_REGION":[0.667,0.958,1.6309],
        "PHO_COEFF":[1.0,1.0,1.0,1.0],
        "IR_COEFF":[1.323,-0.834,1.8424,-0.082],
        "DGF":[8.975,38.131,2.764,12.788]
    },
    "UVI":[4e-05,2.1e-05,0.001048,2.561744]
}
```


Appendix: Onboard file example: log csv


```
1705917869,BAT,3500
1705917929,BAT,3500
1705917989,BAT,3500
1705918050,BAT,3499
```

```
1705918050,BAT,Deep Sleep
1705942896,REBOOT,USER
1705943468,REBOOT,SOFTWARE
1705943472,BAT,3552
1705943474,REBOOT,SOFTWARE
1705943476,REBOOT,SOFTWARE
1705943480,BAT,3628
1705943541,BAT,3622
1705943601,BAT,3620
1705943661,BAT,3618
```

Appendix: Onboard file example as table: log csv

column 00	column 01	column 02
1705917869	BAT	3500
1705917929	BAT	3500
1705917989	BAT	3500
1705918050	BAT	3499
1705918050	BAT	Deep Sleep
1705942896	REBOOT	USER
1705943468	REBOOT	SOFTWARE
1705943472	BAT	3552
1705943474	REBOOT	SOFTWARE
1705943476	REBOOT	SOFTWARE
1705943480	BAT	3628
1705943541	BAT	3622
1705943601	BAT	3620
1705943661	BAT	3618

Appendix: Onboard file example: Sensor_Data csv

```
1706922015,INF,VEET_1.5,3G7Z178FCD0013,FW1.49f Feb 2 2024
15:12:47,abc123,abc123,-8,2000,2000,2000,2000,R,250,1.00,1.00,1.00,1.00,s,mV,ms,m/s^2,deg/
s,count,#,mm,count,#,Hz
1706922019,PHO,50,512,4,7,11,11,9,9,14,13,25,0,52,39
1706922019,IMU,3.044227,-9.203311,-0.355539,0.122070,1.525879,3.540039,39.000000
1706922019,TOF,0,0,0,0,0,0,0,0,0,11,0,17,8,0,0,7,0,22,28,33,19,23,16,20,19,19,24,24,28,2
5,24,255,215,69,14,0,13,0,10,98,59,29,57,0,0,0,0,91,0,27,55,15,0,0,0,49,0,9,10,10,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,36,0,0,0,0,0,0,0,46,0,0,0,12,25,12,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1825,0,1761,1776,0,0,497,0,18
35,1856,1793,1800,1097,1114,507,509,1895,1899,1812,1872,1125,1144,131,128,108,367,0,1996,0
,1254,107,109,361,361,0,0,0,0,113,0,358,359,307,0,0,0,111,0,330,315,297,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,1968,0,0,0,0,0,0,0,1955,0,0,0,515,519,502,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1706922020,ALS,100,4096,2048,4096,415,23862,23762,7,31.6272
1706922025,INF,VEET_1.5,3G7Z178FCD0013,FW1.49f Feb 2 2024
15:12:47,abc123,abc123,-8,2000,2000,2000,2000,R,250,1.00,1.00,1.00,1.00,s,mV,ms,m/s^2,deg/
s,count,#,mm,count,#,Hz
1706922029,PHO,50,512,6,9,14,15,12,15,18,21,0,0,0,0
1706922029,IMU,0.770933,-9.180567,-2.689886,-0.549316,-0.122070,-0.183105,37.000000
```

```
1706922029,TOF,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,34,29,26,23,16,20,25,26,34,25,22,20,11,13,19,25,39,
26,23,22,14,19,25,34,40,26,23,22,13,18,23,33,38,37,26,22,23,24,28,39,37,37,37,28,26,26,33,
41,32,32,32,31,34,33,34,34,31,32,31,31,35,35,34,33,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1706922030,ALS,100,4096,1024,2048,0,0,0,65534,0.0000
```

Appendix: Onboard file example as truncated table: Sensor_Data csv

NOTE: The formatted version of raw data is too large to display.

column 00	column 01	column 02	column 03	column 04	column 05
1706922015	INF	VEET_1.5	3G7Z178FCD 0013	FW1.49f Feb 2 2024 15:12:47	abc123
1706922019	PHO	50	512	4	7
1706922019	IMU	3.044227	-9.203311	-0.355539	0.122070
1706922019	TOF	0	0	0	0
1706922020	ALS	100	4096	2048	4096
1706922025	INF	VEET_1.5	3G7Z178FCD 0013	FW1.49f Feb 2 2024 15:12:47	abc123
1706922029	PHO	50	512	6	9
1706922029	IMU	0.770933	-9.180567	-2.689886	-0.549316
1706922029	TOF	255	255	255	255
1706922030	ALS	100	4096	1024	2048

Appendix: Sensor package: Inertial measurement unit: BST-BMI270

Each temple arm of the VEET device includes a highly-integrated, low-power inertial measurement unit (IMU) that provides precise acceleration and angular rate (gyroscopic) measurement.

For detailed information about the inertial measurement unit sensor package, navigate to the BMI270 pdf file.

Appendix: Sensor package, Ambient Light Sensor: TSL2585

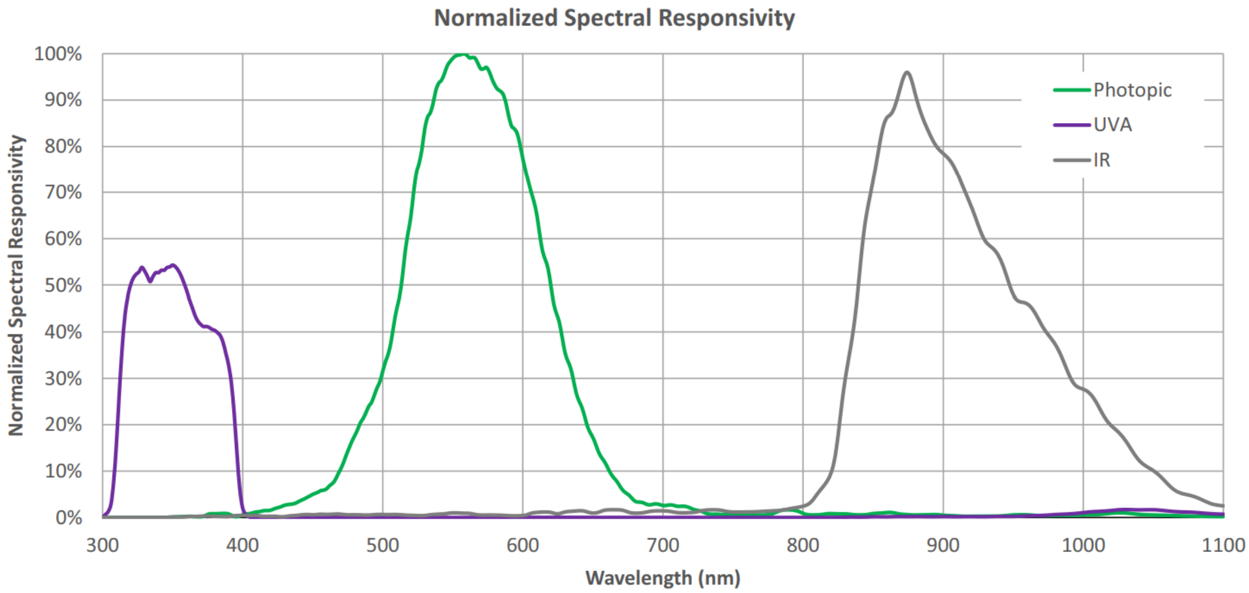
The TSL2585 incorporates photopic, IR, and UV photodiodes that are connected to 3 modulators that provide the following features.

Concurrent ambient light detection.
UV detection.
Light flicker detection.

The photopic photodiode area is covered with an optimized photopic filter. In association with a dedicated IR channel, this architecture accurately measures ambient light and allows the calculation of irradiance of different light sources.

The TSL2585 also integrates the detection of the frequency of ambient light flicker. The detection runs in parallel to the ambient light sensor using the same photodiodes. The TSL2585 is able to measure the flicker of light up to 7000 Hz.

The UV photodiode area is covered with a band-pass UV filter.



Normalized spectral responsivity. Image from [TSL2585-DS001043](), Fig. 12 courtesy of [ams OSRAM]().

The spectral responsivities shown in the figure are measured with 128x AGAIN and 100ms integration time for the Photopic and IR channels and 1024x AGAIN and 100ms integration time for the UV channel. The spectral responsivities are normalized to the Photopic channel.

For detailed information about the Ambient Light Sensor package, navigate to the TSL2585 pdf file.

Appendix: Sensor package, Spectral Sensor: AS7341

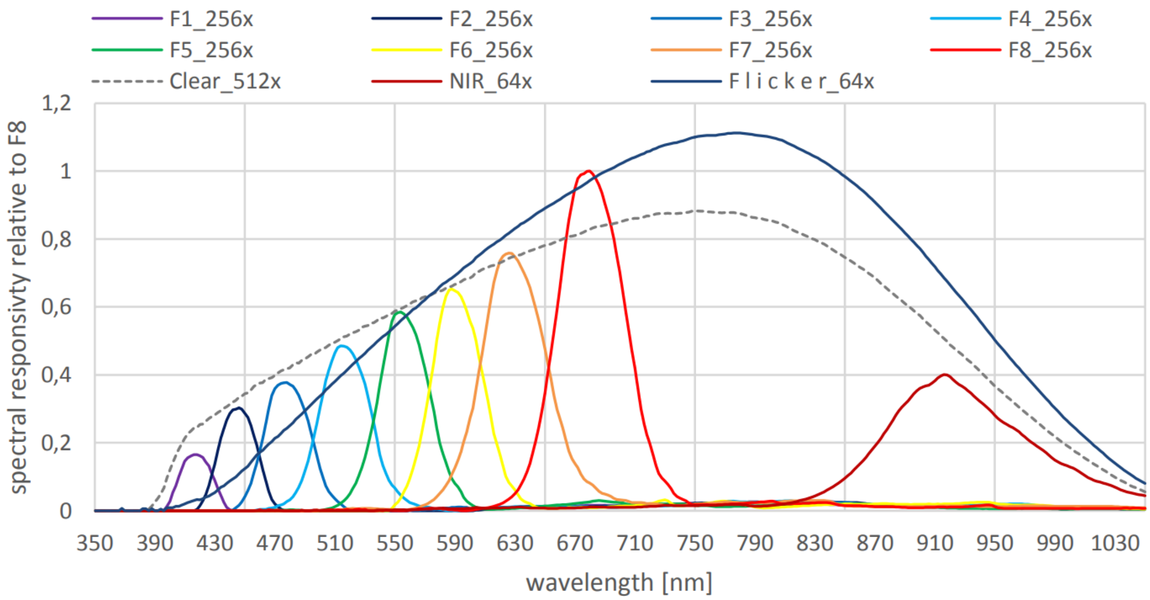
The AS7341 is an 11-channel spectrometer with the following features.

Spectral identification,
Passive ambient light measurement.
Color calibration.

The spectral response is defined by an individual channel that covers approximately 350nm to 1000nm with each of the following channels.

8 channels centered in the visible spectrum (VIS).
one near-infrared (NIR).
one clear channel.

The researcher or data analyst may deduce information about the ambient light conditions and light sources using the NIR channel in combination with the other VIS channels.



Measured spectral responsivity relative to F8. Image from [AS7341-DS000504](), Fig. 19 courtesy of [ams OSRAM]().

For detailed information about the spectral sensor package, navigate to the AS7341 pdf file.

Appendix: Sensor package, Time of flight sensor: TMF8828

The TMF8828 is a direct time-of-flight (dToF) sensor available in a small footprint modular package with integrated Vertical Cavity Surface Emitting Laser (VCSEL). The dToF device is based on the following technologies and has a detection range of 5000 millimeter (mm).

Single photon avalanche diode (SPAD).
Time-to-digital converter (TDC).
Histogram.

The lens on the SPAD of the TMF8828 supports the following multizone output data and a wide field of view that is dynamically adjustable.

3x3
4x4
3x6
8x8

A multi-lens-array (MLA) above the VCSEL widens the field of illumination (FoI). The TMF8828 processes all of the raw data and provides values of `Distance` and associated values of `Confidence` on the integrated I2C interface.

The TMF8828 is capable of 3x3, 4x4, 3x6, and 8x8 zones of operation. Each of the temple arms of the VEET device is set to 8x8 of operation.

For detailed information about the time of flight sensor package, navigate to the TMF8820/21/28 pdf file.

Appendix: VEET Sensor Post Processing

Revision	Description	Date	Author
IR	Document uploaded to share drive	6/6/2023	JDP
1	Updated PHO scale factors, clarified Lux calc variables, removed manufacturer DGF values.	6/9/2023	JDP
2	FW 1.00 - Updated sensor parameters and equations	8/3/2023	JML
3		4/26/2024	JDP

Table of Contents

Introduction

This document describes post processing required, methods and supplementary information regarding the VEET sensors for firmware version 1.02+.

Optical Sensor Primer

Before collecting and interpreting data with the device it is important to have a basic understanding of spectral power distribution (SPD) curves, spectral irradiance, irradiance, lux, and how these values relate to each sensor.

Spectral Irradiance: Irradiance as a function of frequency or wavelength. In this document it will refer to wavelength [W/m^2/nm].

SPD: Spectral power distribution is the measurement of power per unit area per wavelength of an illumination. In this document SPD will refer to a plot of spectral irradiance vs wavelength.

Irradiance: A radiometric unit of measurement of the radiant flux (power) received by a surface per unit area [W/m^2]. This could be calculated from SPD by integrating over the desired range of frequencies.

Lux: A photometric unit of measurement of illuminance. A measurement of luminous flux per unit area as perceived by the human eye.

The optical sensors report sensor counts, which for an idealized sensor is proportional to irradiance of the light in sensor sensitivity range. In a sense it is the integration of the SPD curve. However, the optical sensors are not ideal and have response curves that are not uniform across all wavelengths, therefore sensor counts will not be proportional to irradiance across all light sources due to variation in SPD shape.

In addition to non-uniformity, the sensor channels also do not have perfect blocking of light outside of their desired sensitivity ranges. This is particularly true for IR, not all IR gets blocked by the sensor filters and depending on the SPD shape this can cause a significant contribution to the sensor counts reading.

AMS TSL2585 Ambient Light Sensor (ALS)

Overview

The main purpose identified for the ALS sensor at this time is to provide photometric data in the form of lux.

The ALS sensor outputs 3 gain values, 3 channel [count] values, and 1 flicker [Hz] value. The ALS sensor has automatic gain control (AGC) enabled. There are three gain fields, one for each ultraviolet (UV), visible (VIS), and infrared (IR) channels.

Headers

Timestamp, ALS, Int, uvGain, visGain, irGain, uvValue, visValue, irValue, flickerValue, Lux

Item	unit	Data type	Range/Description
time_stamp	seconds	uint32	Epoch time since 1/1/1970
ALS	N/A	string	Identifies sensor line
Integration Time	milliseconds	int16	Length of measurement time of the sensor reading, analogous to shutter speed. Currently a fixed value.
uvGain	N/A	double	0.5,1,2,4,....2048, 4096
visGain	N/A	double	0.5,1,2,4,....2048, 4096
irGain	N/A	double	0.5,1,2,4,....2048, 4096
uvValue	raw counts~~Counts~~	uint32	0 - 1,048,575
visValue	raw counts~~Counts~~	uint32	0 - 1,048,575
irValue	raw counts~~Counts~~	uint32	0 - 1,048,575
Flicker	Hz	double	0 - 7000 *
Lux	lux	float	0 - 128,000 **

*Flicker values reported are experimental and have not been validated.

**128,000 lux is the theoretical upper limit of the sun at high noon. Upper values around 100,000 are more typical for intense direct sun exposure. However, this is not a device limit, with the current lux coefficients it is possible for the device to output a value of up to 160,361 at full sensor saturation (exact upper limit will change with lux coefficients).

Exception Values

There are three exception values that need to be checked for and handled appropriately when post processing the data. When an exception value is found the corresponding data should be dropped.

Gain output of 'null'

Sensor gain outputs are initialized as 'null' values in FW. The first reading of the sensor will output 'null' gains. This entire line can be disregarded.

Gain output of 65535

This indicates channel saturation. The corresponding channel value should be disregarded. This is not uncommon during highly dynamic shifts in light intensity where the AGC may not keep up and the modulator will saturate before the gain can be adjusted. For highly dynamic changes in lighting conditions it is not uncommon to have one or two data points that will saturate, resulting in dropped data points during post processing.

The only other reason the sensor should saturate is if it is exposed to lighting above its upper range, for the ALS sensor this would need to be a light source brighter than the sun.

Flicker output of 65534

This indicates that the device attempted to obtain a flicker measurement, but it did not pass standard deviation check, therefore the flicker measurement was not considered valid and not reported.

Basic Counts

Gain ratios are shown in Figure 1. Available gains are 0.5x, 1x, 2x, …4096x. The raw counts reported with respect to each gain setting are always between 0 and 1,048,576 (2^20 -1). To report meaningful continuous values, the raw counts must be normalized to a single reference gain value of 128x (gain ratio of 1), and divided by integration time in milliseconds.

To calculation basic~~normalize~~ counts:

1. Read raw counts and gain per sensor channel.
2. Look up gain ratio for corresponding gain read in (1).
3. Normalized counts by ~~D~~dividing~~e~~ raw counts from (1) by gain ratio (2).
4. Divide normalized counts by integration time in milliseconds.

$$normCount = \frac{rawCount}{gainRatios[gainIndex]}$$

$$\text{basicCount} = \frac{normCount}{tint}$$

Normalizing counts using datasheet gain ratios accounts for the non-linearity of the sensor at higher gains. For example a perfectly linear sensor at 4096 gain would have a gain ratio of 32 (i.e. 4096/128), however as shown in Figure 1 shows the datasheet gives a typical gain ratio value of 25.35 for this gain.

~~For quicker gain normalization that does not account for this non-linearity, the gains can be normalized by the gain index directly, divided by 128 to scale to gain ratios, then divide by integration time in milliseconds to obtain basic counts.~~

~~If values need to be compared across multiple integration time settings, then basic counts should be used. To obtain basic counts divide normalize counts by integration time in milliseconds.~~

Figure 6:
ALS Characteristics of TSL2585, ALS Gain = 128x, Integration Time = 10ms (unless otherwise noted)

Parameter	Conditions	Min	Typ	Max	Unit
Dark ADC count value [1]	$E_e = 0 \mu W/cm^2$ ALS gain: 512x Integration time: 98ms	0	1	3	counts
ALS gain ratios [2]	0.5x	1/270.78	1/249.13	1/230.68	
	1x	1/133.17	1/123.85	1/115.74	
	2x	1/66.99	1/62.97	1/59.41	
	4x	1/33.39	1/31.72	1/30.21	
	8x	1/16.17	1/15.53	1/14.93	
	16x	1/8.30	1/7.97	1/7.66	
	32x	1/4.15	1/3.99	1/3.83	
	64x	1/2.09	1/2.01	1/1.93	
	256x	1.78	1.93	2.07	
	512x	3.42	3.80	4.18	
	1024x	6.16	7.42	8.68	
	2048x	10.26	14.06	17.86	
	4096x	11.41	25.35	39.29	
ADC noise [3]	White LED, 2700K [4] Integration time: 100ms		0.05		%
Photopic channel irradiance responsivity	White LED, 2700K [4]	62.7	73.7	84.8	counts/ ($\mu W/cm^2$)
IR channel irradiance responsivity	IR LED =940nm [5]		74.8		counts/ ($\mu W/cm^2$)
UV channel irradiance responsivity	UV LED =365nm [6] ALS gain: 1024x		82.8		
IR / Photopic channel ratio	White LED, 2700K		8.1		%
UV / Photopic channel ratio	White LED, 2700K		0.0		
Photopic / IR channel ratio	IR LED = 940nm		3.6		
UV / IR channel ratio	IR LED = 940nm		0.9		

Note(s):

1. The typical 3-sigma distribution shows less than 1 count. For this measurement, each modulator is always connected to one photodiode whereas the photodiodes are sequentially multiplexed.
2. The gain ratios are calculated relative to the response with ALS gain = 128x.
3. ADC noise is calculated as the standard deviation relative to full scale.
4. The White LED is an InGaN light-emitting diode with integrated phosphor and the following characteristic: correlated color temperature = 2700K.
5. The IR LED is an AlGaAs light-emitting diode with a peak wavelength of λ_P = 940nm.
6. The UV LED is an light-emitting diode with a peak wavelength of λ_P = 365nm.

Figure 1 TSL2585 Characteristics, Datasheet Excerpt

Figure 1 ALS gain ratios are shown as a Python dictionary in Figure 2.

```
gainRatios = {0.5   : 1/249.13,
              1   : 1/123.85,
              2  : 1/62.97,
              4  : 1/31.72,
              8  : 1/15.53,
              16 : 1/7.97,
              32 : 1/3.99,
              64 : 1/2.01,
              128  : 1,
              256  : 1.93,
              512  : 3.80,
              1024 : 7.42,
              2048 : 14.06,
              4096 : 25.35 }
```

Figure 2 Datasheet Gain Ratios as Python Dictionary

$\left(\frac{}{100}\right)$TSL2585 integration time

Note that some of the details here are background information. Reference clock, ATIME, and ASTEP are parameters that are internal to the device FW and not user configurable.

The AS7341 has an internal 720 kHz reference clock that is used for the integration time, (also known as exposure time). 1/ 720 kHz = 1.388889 µs, which is the smallest time increment.

ASTEP: Sets the number of clock cycles to use as a timebase. ASTEP = 0 represents 1.388889 µs.

ASTEP(n) = 1.388889 µs * (n + 1)

ATIME: Sets the number of ASTEPs. ATIME = 0 represents 1 ASTEP period. ATIME(n) = ASTEP * (n + 1)

The total integration time t_{int} = (ATIME +1) x (ASTEP +1) * 1.388889 µs

Fixed (non-controllable) settings

ATIME = 179, ALS STEP = 399, Flicker STEP = 511

These settings create an ALS 100 ms and flicker 128 ms integration times.

Although ALS integration time is currently static, it is included in the device output so that data is still comparable to future FW that may have a dynamic or different integration time.

Converting Basic Counts to Irradiance

See section “Optical Sensor Primer” for a primer on why obtaining irradiance from basic counts is non-trivial. Datasheet counts/(uW/cm^2) factors do give a conversion from basic counts to irradiance, however these are only for a single light source and bare sensor. They do not apply to the VEET device, which is behind its own optical package, nor would they apply uniformly across all SPDs.

If an irradiance conversion is required additional approaches are needed ~~more work would need to be done~~ to create the appropriate correction functions and calibration factors.

Converting Raw Counts to Lux

The ALS sensor visible channel has a photopic filter that closely matches the response of the eye. It does however pass IR which can significantly impact lux measurements. To account for this lux is estimated as a function on IR and Visible channel readings. The lux equation is:

$$lux = \frac{DGF}{TINT}(C_0 VIS + C_1 IR)$$

Where DGF is the device glass factor, TINT is integration time in milliseconds, VIS is the gain normalized visible channel raw counts~~reading~~, IR is the gain normalized IR channel raw counts, C_0 and C_1 are VIS and IR coefficients. In this case VIS and IR coefficients are raw counts normalized using the gain index, which does not account for data sheet non-linearity of the higher gains. Initially gain indices were used in lieu of datasheet gain ratios to match manufacturer sample calculation methods, and later left in place because it has proved to be accurate enough for lux measurements. If non-linearity of higher gains is a concern for lux measurements the calculations can be repeated in post processing using the gain ratios discussed earlier in this paper and then dividing by 128 match firmware calculations (for gain 128 gain ratio is 1).

DGF and coefficients for IR and Vis channels depend on the ratio of VIS/IR in the light being measured, to cover a wide variety of light sources a four piece piecewise function is used to estimate the lux across different light types.

All the necessary coefficients for this calculation are stored on the calib.json file on the device, in the following format:

```
"LUX":{
  "IR_PHO_REGION":[
    0.667,
    0.958,
    1.6309
  ],
  "PHO_COEFF":[
    1.0,
    1.0,
    1.0,
    1.0
  ],
  "IR_COEFF":[
    1.323,
    -0.834,
    1.8424,
    -0.082
  ],
  "DGF":[
    8.97,
    38.13,
    2.76,
    12.79
  ]
}
```

'IR_PHO_REGION' in the calib.json file represents the IR/VIS ratio value that separates the piecewise equations, 3 ratios to separate the 4 sets of equations.

To develop these coefficients the PHO_COEFF, or C_0 value was arbitrarily set to 1 to remove it from the curve fitting, this assumption is for calculation convenience and although it affects the values of DGF and IR_COEFF is has no effect on ultimate value of the predicted lux.

A new set of lux coefficients can be calculated with little effort if a trusted lux detector and a sufficient number of light sources is readily available. The coefficients are obtained by rearranging the lux equation into a form that can be curve fit. The form of the equation used to curve fit is:

$$\frac{lux}{vis} = CPL + C_1 (\frac{IR}{VIS} + CPL)$$

where, $CPL = DGF/TINT$

In this form the equation matches the slope intercept form for a line y=mx+b, where,

y = lux/vis, x = IR/VIS, $m = CPL * C_1$, and B= CPL.

To perform the curve fit and obtain lux coefficients do the following.

1. Collect data from trusted lux detector and VEET. Take care to have data collected from the same scene at a close time interval.
2. Bring all data into excel.
3. Plot Lux/Vis vs IR/VIS
4. Examine the data and divide into 4 linear curve fit regions.
5. For each region perform a linear regression to get the coefficients from the equations above.
6. The x axis value (IR/VIS) that separates each line fit are PHO_COEFF values.

AMS AS7341 Spectral Sensor (PHO)

Overview

The Spectral sensor outputs one gain value, 8 narrow band channel [count] values, and 4 wide band channel [count] values. Available gain values are 0.5x, 1x, 2x, … 512x. An AGC (automatic gain control) function is implemented that adjusts the gain automatically to reduce the chance of saturation due to excessive lighting.

Headers

Timestamp, PHO, Int, ~~g~~Gain, 415, 445, 480, 515, 555, 590, 630, 680, NIR (940), Dark, ClearL, ClearR

Item	**unit**	**Data type**	**Range/Description**
time_stamp	seconds	uint32	Epoch time since 1/1/1970
PHO	N/A	string	Identifies sensor line
Integration time	Milliseconds	int16	Length of measurement time of the sensor reading, analogous to shutter speed. Currently a fixed value.
Gain	n/a	float	Gain value, 0.5, 1, 2, 4…512
415	raw counts	uint16	0-65535 / 415 nm, 26 nm FWHM
445	raw counts	uint16	0-65535 / 445 nm, 30 nm FWHM
480	raw counts	uint16	0-65535 / 480 nm, 36 nm FWHM
515	raw counts	uint16	0-65535 / 515 nm, 39 nm FWHM
555	raw counts	uint16	0-65535 / 555 nm, 39 nm FWHM
590	raw counts	uint16	0-65535 / 590 nm, 40 nm FWHM
630	raw counts	uint16	0-65535 / 630 nm, 50 nm FWHM
680	raw counts	uint16	0-65535 / 680 nm, 52 nm FWHM

Near-IR	raw counts	uint16	0-65535 / 910 nm
Dark	raw counts	uint16	0-65535 / Masked sensor (noise floor)
Clear L	raw counts	uint16	0-65535 / Full spectrum clear
Clear R	raw counts	uint16	0-65535 / Full spectrum clear

Exception Values

Gain value of 65535

Channel saturation. This value is output if any channel is saturated. These channel values should be disregarded.

Basic Counts

https://look.ams-osram.com/m/269928fe0dba7511/original/Spectral-Sensor-Calibration-Methods.pdf

Gain ratios are shown in Figure 3. The raw counts reported with respect to each gain setting are always between 0 and 65,535 (2^16 -1). To report meaningful continuous values, the raw counts must be normalized to a single reference gain value of 64x (gain ratio of 1), and divided by integration time in milliseconds.

$$normCount_{pho} = \frac{rawCount_{pho}}{gainRatios[gainIndex]_{pho}}$$

$$basicCount_{pho} = \frac{normCount_{pho}}{tint}$$

Figure 17:
Optical Characteristics of AS7341, AGAIN: 64x, Integration Time: 27.8ms (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
R_{e_CLEAR}	Irradiance responsivity channel CLEAR	LED: warm white 2700K(5) Ee = 107.67µW/cm²		1750		counts
$R_{e\_FLICKER}$	Irradiance responsivity channel FLICKER	LED: warm white 2700K(5) Ee = 52.32µW/cm²		6810		counts
R_{e_NIR}	Irradiance responsivity channel NIR	LED: warm white 2700K(5) Ee = 107.67µW/cm²		112		counts
		LED: 940nm(5) Ee = 98 µW/cm² AGAIN = 128x tint = 100ms		5135		
$Dark\_1$(1)(6)	Dark ADC 0-4 count value	Ee = 0µW/cm² AGAIN: 512x Integration time: 98ms		0	3	counts
$Dark\_2$(6)	Dark ADC 5 count value	Ee = 0µW/cm² AGAIN: 512x Integration time: 98ms		0	5	counts
Gain(2) ratio	Optical gain ratios, relative to 64x gain setting	AGAIN: 0.5x	0.007	0.008	0.009	
		AGAIN: 1x	0.0145	0.016	0.0175	
		AGAIN: 2x	0.03	0.032	0.034	
		AGAIN: 4x	0.062	0.065	0.068	
		AGAIN: 8x	0.119	0.125	0.131	
		AGAIN: 16x	0.237	0.25	0.263	
		AGAIN: 32x	0.47	0.5	0.53	
		AGAIN: 64x		1		
		AGAIN: 128x	1.8	2	2.1	
		AGAIN: 256x	3.75	3.95	4.25	
		AGAIN: 512x	7.25	7.75	8.25	
ADC noise(3)		AGAIN: 16x Integration time: 10ms		0.005		% full scale
t_{int}	Typical integration time(4)	ASTEP = 599 ATIME = 29		50		ms
t_{ASTEP}	Integration time step size	ASTEP = 999		2.78		ms
h_{ca}	Half cone angle	On the sensor		40		deg

(1) The typical 3-sigma distribution is between 0 and 1 counts for AGAIN setting of 16x.
(2) The gain ratios are calculated relative to the response with integration time: 27.8ms and AGAIN: 64x.
(3) ADC noise is calculated as the standard deviation of 1000 data samples divided by full scale.
(4) Integration time, in milliseconds, is equal to: (ATIME + 1) x (ASTEP + 1) x 2.78µs
(5) Refer to Figure 16:
Typical LED Spectra Used in Final Test of AS7341
(6) Register 0xD6 / AZ_CONFIG is set to "1" – auto zero done before every integration cycle

Figure 3 AS7341 Optical Characteristics Data Sheet Excerpt

AS7341 integration time

The AS7341 has an internal 360 kHz reference clock that is used for the integration time, (also known as exposure time). 1/ 360 kHz = 2.777778 µs, which is the smallest time increment.

ASTEP: Sets the number of clock cycles to use as a timebase. ASTEP = 0 represents 2.777778 µs.

ASTEP(n) = 2.777778 µs * (n + 1)

ATIME: Sets the number of ASTEPs. ATIME = 0 represents 1 ASTEP period. ATIME(n) = ASTEP * (n + 1)

The total integration time t_{int} = (ATIME +1) x (ASTEP +1) * 2.777778 µs

Fixed (non-controllable) settings

ATIME = 29, ASTEP = 599

These settings create a 50 ms integration time

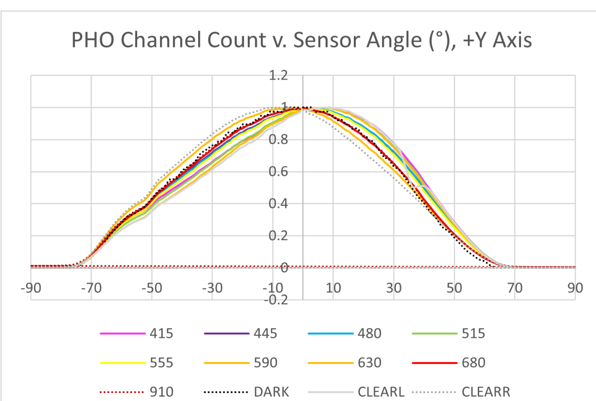
Converting Basic Counts to Irradiance

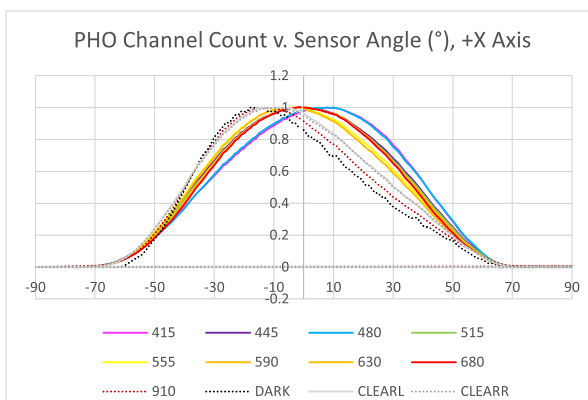
Note this section is currently under additional development.

Appendix: Angular sensitivity of light sensors

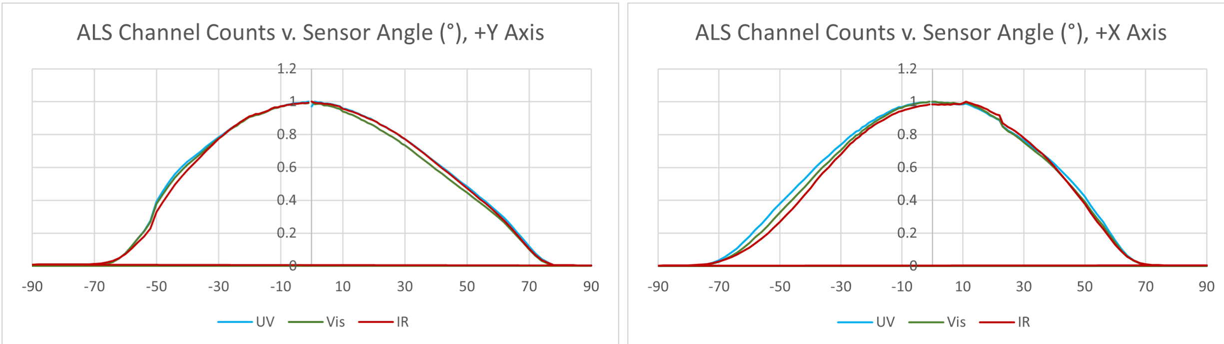
The VEET device receives more light directly in front of the sensors rather than at an angle. This directly correlates to channel raw counts, therefore the ALS and PHO Sensors readings are biased to the light in front of them. When evaluating or characterizing sensor performance or making comparisons to other light logging devices, knowledge of the sensor's angular response is necessary to ensure an accurate evaluation.

The PHO Sensor and ALS Sensor figures below display a plot of values from the sensor data, comparing the angle of the temple arm of the VEET unit from an accurately aligned light source. A measurement of 0 degrees is directly in front of (perpendicular to) the sensor window on all measurements. The vertical axis of rotation is set from `-Y` to `+Y`. The horizontal axis of rotation is set from `-X` to `+X`. When the light moves from directly in front of the sensor in any direction, the incident light on the sensor is effectively reduced. The direction relative to the sensor includes left, right, up, and down. Please contact Meta for data behind these plots if needed.

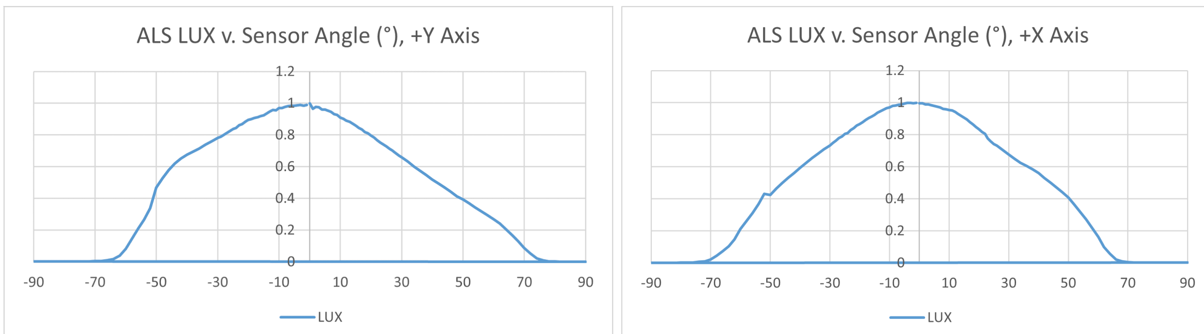




PHO channel response in basic~~normalized~~ counts vs. sensor angle, normalized to unity

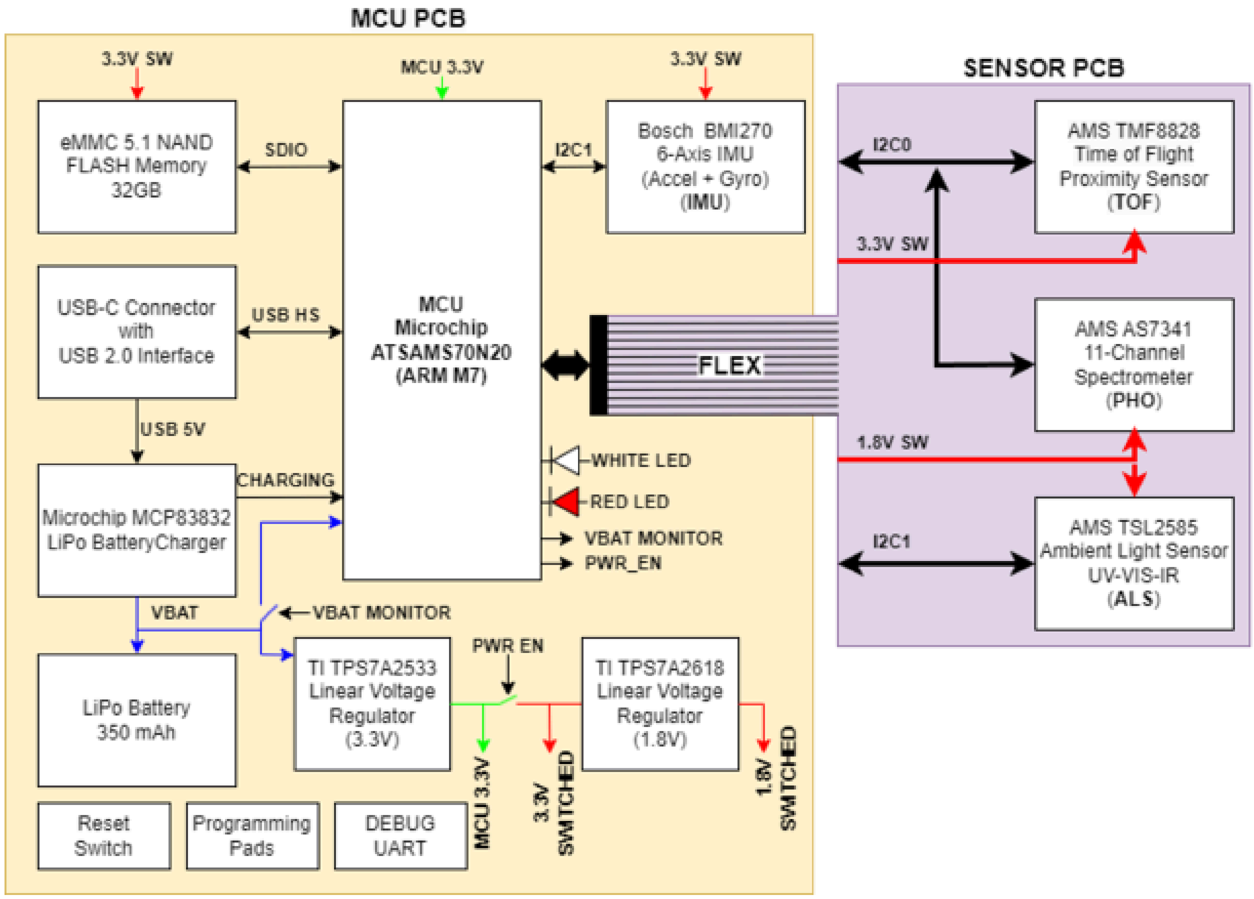


ALS ~~Normalized~~ channel response in basic~~normalized~~ counts vs. sensor angle, normalized to unity



ALS Lux vs. sensor angle, normalized to unity

Appendix: System architecture



Flow diagram for MCU PCB and Sensor PCB

Flow diagram for MCU PCB and Sensor PCB

The electrical architecture of the VEET device consists of two printed circuit boards (PCBs).

A rigid PCB that contains the following main components.

- Microcontroller (MCU)
- Linear battery charger
- Inertial Measurement Unit (IMU)
- 32GB eMMC Flash Memory
- Voltage regulators
- Sensor interface connector
- High speed USB 2.0 interface (480 Mbit)
- Two-color LED

A rigid-flex PCB that contains the following sensor components.

- ToF sensor
- PHO sensor
- ALS sensor

The sensor PCB connects to the main PCB using a flexible printed circuit.

The main PCB contains an Arm Cortex-M7 microcontroller that logs the data from the following four sensors and the battery voltage.

- Bosch BMI270 IMU
- AMS TMF8828 ToF
- AMS AS7341 PHO
- AMS TSL2585 ALS

The microcontroller connects to the sensors using two I2C communication interfaces.

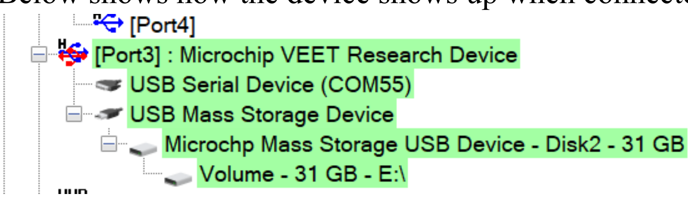
- One I2C communication interface connects to the IMU and ALS sensor.
- One I2C communication interface connects to the ToF and PHO sensors.

The two interfaces allow two sensors to be read simultaneously as long as they don’t interact, such as the LED on time of the ToF sensor interacting with the sensing phase of the ALS and PHO sensors.

A USB Type-C connector allows each temple arm of the VEET device to complete the following actions.

- Connect to a computer to transfer data and charge.
- Connect to a power adapter to charge.

Below shows how the device shows up when connected to a windows PC:



When each temple arm of the VEET device is attached to a computer, the temple arm provides access in the following ways.

- USB Mass Storage Device (MSD).
  - Each temple arm of the VEET v1.5 device contains an internal 32 GB eMMC memory module.
- USB Virtual Communication Port (VCP). Also sometimes called USB Communication Device Class (CDC).
  - This is a virtual serial port. Note that unlike some USB to UART serial ports, there is no physical serial port, so the baud rate, parity, handshaking, etc settings don’t matter.
  - This port hosts a serial console that can be used to command the device to take actions such as read sensors, change settings, and perform maintenance tasks.
  - Most users will not need to interact with this interface, since the VEET manager software typically communicates via this interface.
  - You can find the port by viewing the connected devices in your specific operating system’s settings interface.
  - You can interact with this console via most serial terminal programs such as:
    - TeraTerm - https://teratermproject.github.io/index-en.html
    - Putty - https://www.putty.org/
    - Coolterm - https://freeware.the-meiers.org/
  - Type the “??” double quotation command followed by the “Enter” key in a terminal to get a list of supported commands.

When external power is available at the USB Type-C connector, a Microchip MCP83832 Lithium Polymer (LiPo) linear battery charger controls how the LiPo battery charges. The LiPo battery is rated for 350 mAh at a maximum charge rate of 175 mA.

A 3.3-volt linear voltage regulator directly powers the MCU. The MCU controls a power switch that allows 3.3-volt and 1.8-volt voltages to be removed from the sensors and memory to promote lower power consumption during deep sleep. The following are the internal voltage regulators:

3.3 volts using the TI TPS7A2533
1.8 volts using the TI TPS7A2618

When attached to a computer, each temple arm of the VEET device uses the USB MSD function to provide a researcher or data analyst access to upload or download files on a USB drive. The available files contain data for calibration, sensor information, configuration, and firmware updates. Each temple arm of the VEET device allows transmission of data using the USB VCP function. The USB VCP function allows the VEETManager software to communicate the following information with each temple arm of the VEET device.

Reset hardware
Update firmware
Set parameters
View live sensor data output

See appendix for details.

Appendix: Configuration impacts on data and battery life

The rate at which sensing occurs directly affects the time the device will run between charging. the table below provides general guidance on expected battery life and

Typical battery and storage life

Example Config	Estimated battery life	Estimated data life
1/2hz all sensing	35hrs	XXdays
1/4hz all sensing	YYhrs	XXdays
1/5hz ToF 1hz other sensing	YYhrs	XXdays

Appendix: Troubleshooting Device

Lost communication with PC

In the event that the temple arm of the VEET device loses communication with the VEETManager software, complete the following actions.

1. Unplug the USB cable from the temple arm of the VEET device.
2. Wait five seconds.
3. Plug the USB cable back into the temple arm of the VEET device.
4. Confirm the connection is restored.

Additionally, some USB hubs can disrupt the connection with the VEET device. Plugging directly into the computer USB can resolve connectivity problems with the software.

If the VEET device fails to reconnect after these steps contact Meta.

Data Loss during data logging

Operation of the VEET Device in the presence of strong radio frequency (RF) interference or near high-powered RF sources may result in data loss or degradation. This data degradation may appear as a mismatch between actual time and VEET Device time. This condition has been observed under controlled testing environments but has not been seen in real world testing to date. The device is safe to operate in these environments.

Expected normal behavior

The following tables provide expected behavior of the device relative to battery voltage (charge level). In conjunction with the Log.csv file on device a researcher can discover when battery levels changed to understand when a participant charged their device and correlate any missing data periods to missed charging cycles.

Disconnected from computer

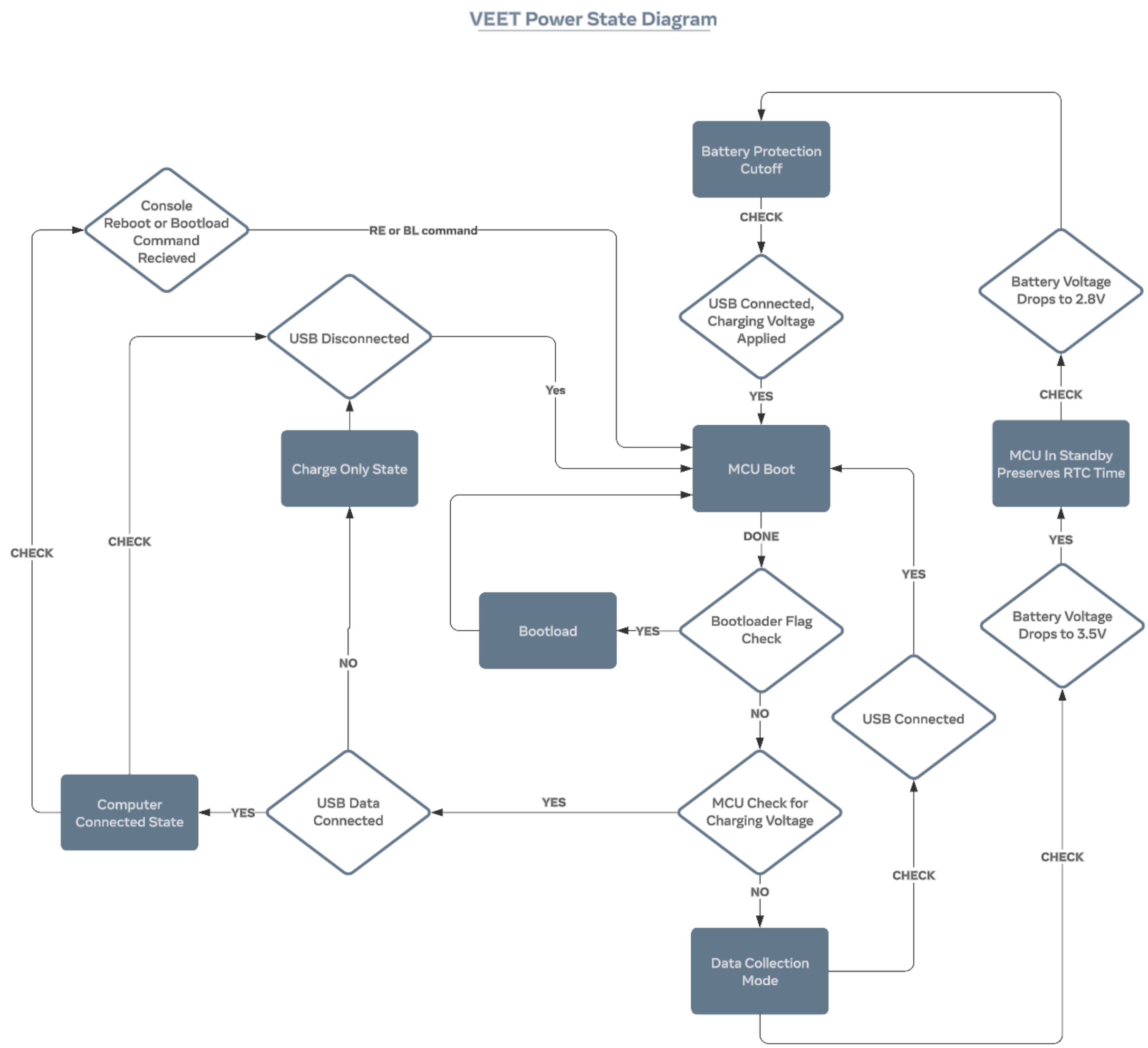
Battery voltage	Log	Battery active
3.5V to 4.2V	X	X
2.8V to < 3.5V		X
< 2.8V		
Battery voltage	**Operation mode**	**Clock retained**
3.5V to 4.2V	Normal	X
2.8V to < 3.5V	Deep Sleep	X
< 2.8V	Block	

Charge while connected to external charger

Battery voltage	Connected	Disconnected
3.5V to 4.2V	Charge	Discharge
2.8V to < 3.5V	Charge	Discharge
< 2.8V	Charge	Block

Charge while attached to computer

Battery voltage	Log	Battery active
3.5V to 4.2V		X
2.8V to < 3.5V		X
< 2.8V		
Battery voltage	**Operation mode**	**Clock retained**
3.5V to 4.2V	Normal	X
2.8V to < 3.5V	Deep Sleep	X
< 2.8V	Block	



Fully discharged device

If the battery decreases below 2.8 volts, the battery is disconnected via a hardware cutoff circuit, entering what is referred to as the shutdown state. The shutdown state of the battery removes power from the entire temple arm circuit boards. The shutdown state causes the clock to stop, but the file data is retained in FLASH memory.

The temple arm of the VEET device does not resume communication until the battery has charged above 3.0 volts. The battery may require up to 10 minutes to charge in order to re-start.

If the temple arm of the VEET device does not respond after connection to power for 10 minutes, disconnect the VEET device from power and reconnect for an additional 10 minutes. If at this point the device does not recover, contact Meta for further information.

Charge duration

For a fully depleted battery, at least two hours are required to fully charge the temple arm of the VEET device.

The temple arm of the VEET device may take longer to charge when attached to a computer since the charging current is shared with the USB communication circuitry. That circuitry is blocked when attached to an external USB power supply.

> **Note**: It is normal for the temple arm of the VEET device to feel warm while charging.

Windows wants to format the drive

For Windows-attached devices, there is an occasional occurrence of Windows stating that the drive in the unit device must be formatted or that an error is detected. If the temple arm of the VEET device is reformatted, all of the associated files are deleted including the calibration file.

> **IMPORTANT**: Only format the device as a last resort to resolve an issue.

The calibration file is required in order to accurately log data. It is expected that formatting is performed by the VEETmanager software.