Document Title:	**Verisense Data Dictionary**
Document Number:	ASM-DOC06
Revision Number:	03
Number of Pages:	24

Approvals

	Name	Signature	Date
Originator :	Ruaidhri Molloy	RMolloy	08-Sep-2021
Reviewed by:	Mark Nolan	Mark Nolan	08-Sep-2021
Approved by:	Thaynoara Gonclaves	Thaynoara Gonçalves	08-Sep-2021

Revision History

Revision	Author	CCF Number	Description of Changes
01	N Davidson	NA	Initial Input
02	Ruaidhri Molloy	21-022	Updates to support latest file contents (as created by firmware v1.02.87 and file parser v1.02.033)
03	Ruaidhri Molloy	21-024	Added details on payload metadata file columns

Distribution

Person	Department
All Engineers	Engineering

Table of Contents

1 Introduction

1.1 Purpose

To document the output of the Verisense system with regards to captured data.

1.2 Scope

The definition and use of the data within exported from the Verisense System.

1.1 Associated Documents

Document Number	Document Title
ASM- DOC07	Verisense Data GGIR Dictionary

1.2 Departments Affected

- Quality Assurance (QA)
- Regulatory Affairs (RA)
- Research & Development (R&D)

1.3 Responsibility

The Engineering Department is responsible for the maintenance of this document.

1.4 Definitions

Definition	Meaning
AWS S3	Amazon Simple Storage Service (Amazon S3) is storage for the Internet
UNIX time	It is the number of seconds that have elapsed since the *Unix epoch*, minus leap seconds; the Unix epoch is 00:00:00 UTC on 1 January 1970 (an arbitrary date)

1.5 Abbreviations

Abbreviation	Meaning
CSV	Comma Separated Variables
PPG	Photoplethysmography
GSR	Galvanic Skin Response
AWS	Amazon Web Services
CAL	Calibrated
CLI	Command Line Interface

2 Verisense Data

2.1 Overview

The Verisense system captures the raw accelerometer, gyroscope, PPG andGSR data from the Verisense hardware and stores it in a repository (AWS S3) within the Verisense Cloud. What raw data is available depends on what hardware is being used in the Verisense system.

1. SR61 -> Verisense IMU -> Accelerometer and Gyroscope
2. SR63 -> Verisense PPG -> Accelerometer and PPG
3. SR68 -> Verisense Pulse+ -> Accelerometer, PPG and GSR

Further to this derived metrics are produced on the raw data via a number of algorithms

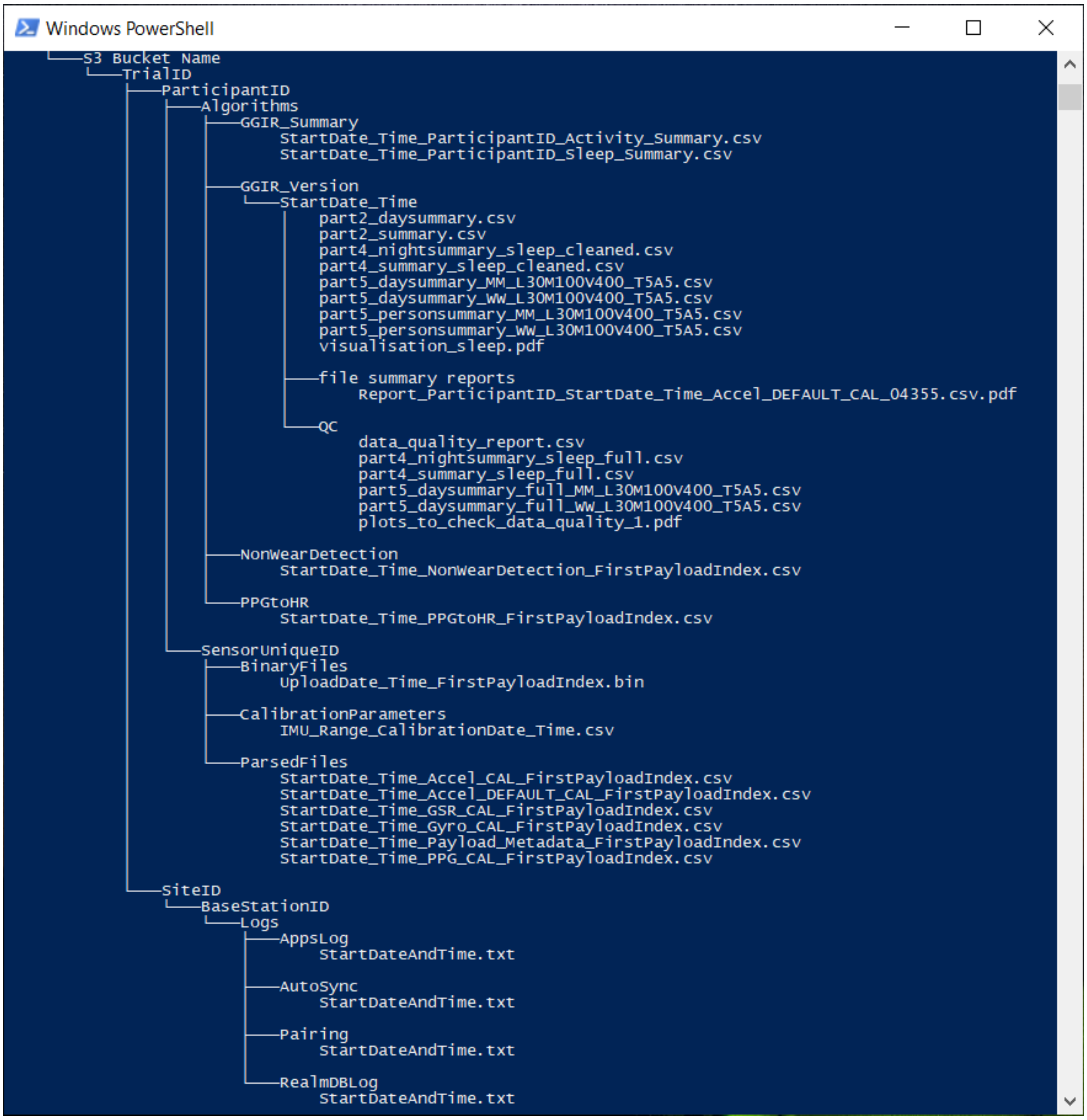
1. Accelerometer -> GGIR algorithm -> Activity and sleep
2. PPG and Accelerometer -> PPGtoHR algorithm -> Heart rate and heart rate variability

The raw and derived metric data can be accessed in two ways, via

1. Verisense web portal – export data functionality.
2. AWS command line interface, contact Shimmer for further information

2.2 AWS S3 File System

All raw sensor, derived metric (algorithms) and base station (log files) data is stored in a bucket dedicated to the one unique trial. The file structure that this data is stored in is shown below in Figure 2-1: AWS S3 file system structure



```
Windows PowerShell
└───S3 Bucket Name
    └───TrialID
        ├───ParticipantID
        │   ├───Algorithms
        │   │   ├───GGIR_Summary
        │   │   │       StartDate_Time_ParticipantID_Activity_Summary.csv
        │   │   │       StartDate_Time_ParticipantID_Sleep_Summary.csv
        │   │   │
        │   │   ├───GGIR_Version
        │   │   │   └───StartDate_Time
        │   │   │       │   part2_daysummary.csv
        │   │   │       │   part2_summary.csv
        │   │   │       │   part4_nightsummary_sleep_cleaned.csv
        │   │   │       │   part4_summary_sleep_cleaned.csv
        │   │   │       │   part5_daysummary_MM_L30M100V400_T5A5.csv
        │   │   │       │   part5_daysummary_WW_L30M100V400_T5A5.csv
        │   │   │       │   part5_personsummary_MM_L30M100V400_T5A5.csv
        │   │   │       │   part5_personsummary_WW_L30M100V400_T5A5.csv
        │   │   │       │   visualisation_sleep.pdf
        │   │   │       │
        │   │   │       ├───file summary reports
        │   │   │       │       Report_ParticipantID_StartDate_Time_Accel_DEFAULT_CAL_04355.csv.pdf
        │   │   │       │
        │   │   │       └───QC
        │   │   │               data_quality_report.csv
        │   │   │               part4_nightsummary_sleep_full.csv
        │   │   │               part4_summary_sleep_full.csv
        │   │   │               part5_daysummary_full_MM_L30M100V400_T5A5.csv
        │   │   │               part5_daysummary_full_WW_L30M100V400_T5A5.csv
        │   │   │               plots_to_check_data_quality_1.pdf
        │   │   │
        │   │   ├───NonWearDetection
        │   │   │       StartDate_Time_NonWearDetection_FirstPayloadIndex.csv
        │   │   │
        │   │   └───PPGtoHR
        │   │           StartDate_Time_PPGtoHR_FirstPayloadIndex.csv
        │   │
        │   └───SensorUniqueID
        │       ├───BinaryFiles
        │       │       UploadDate_Time_FirstPayloadIndex.bin
        │       │
        │       ├───CalibrationParameters
        │       │       IMU_Range_CalibrationDate_Time.csv
        │       │
        │       └───ParsedFiles
        │               StartDate_Time_Accel_CAL_FirstPayloadIndex.csv
        │               StartDate_Time_Accel_DEFAULT_CAL_FirstPayloadIndex.csv
        │               StartDate_Time_GSR_CAL_FirstPayloadIndex.csv
        │               StartDate_Time_Gyro_CAL_FirstPayloadIndex.csv
        │               StartDate_Time_Payload_Metadata_FirstPayloadIndex.csv
        │               StartDate_Time_PPG_CAL_FirstPayloadIndex.csv
        │
        └───SiteID
            └───BaseStationID
                └───Logs
                    ├───AppsLog
                    │       StartDateAndTime.txt
                    │
                    ├───AutoSync
                    │       StartDateAndTime.txt
                    │
                    ├───Pairing
                    │       StartDateAndTime.txt
                    │
                    └───RealmDBLog
                            StartDateAndTime.txt
```

Figure 2-1: AWS S3 file system structure

2.3 Accessing the Data

2.3.1 Via the Web Portal

A limited amount of data (1.5GB per participant at a time) can be exported through the Verisense web portal using the interface shown below in Figure 2-2: Export data via the Verisense web portal.

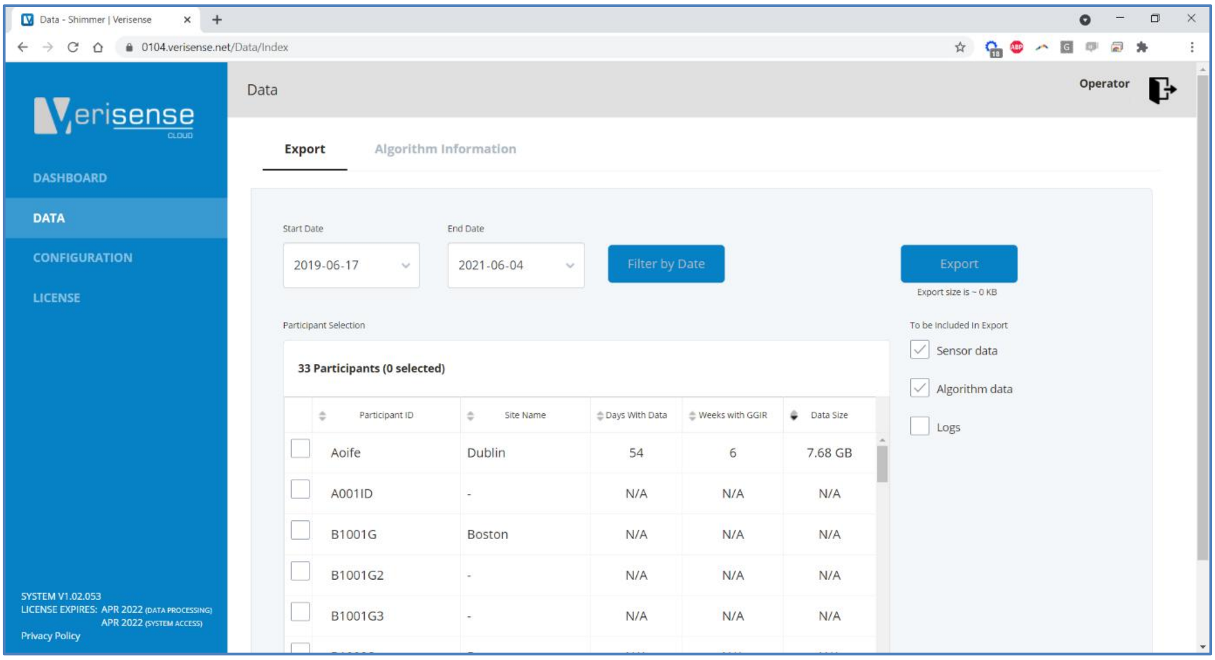


Figure 2-2: Export data via the Verisense web portal

The data exported from the web portal is compressed as a zip file with the filename in the format below.

<TRIAL PARTICIPANT>_DD_MM_YYYY_DD_MM_YYYY.zip

where:

<TRIAL PARTICIPANT>is the participant identifier assigned to the wearer of the Verisense sensor.
DD_MM_YYYY is the requested start date of the data as specified on the web portal.
DD_MM_YYYY is the requested end date of the data as specified on the web portal.

2.3.2 Via the AWS CLI

To export larger datasets or to transfer the data to another cloud system, it is recommend to use the AWS CLI, contact the Shimmer team for more details.

2.4 File Naming Convention

The raw data filenames used the following formats

YYMMDD_HHMMSS_Accel_DEFAULT_CAL_XXXXX.csv

YYMMDD_HHMMSS_Accel _CAL_XXXXX.csv

YYMMDD_HHMMSS_Gyro _CAL_XXXXX.csv

YYMMDD_HHMMSS_PPG_CAL_XXXXX.csv

YYMMDD_HHMMSS_GSR_CAL_XXXXX.csv

YYMMDD_HHMMSS_Payload_Metadata_XXXXX.csv

YYMMDD_HHMMSS_NonWearDetection_XXXXX.csv

YYMMDD_HHMMSS_PPGtoHR_CAL_XXXXX.csv

where:

YY is the year.

MM is the month of the year as a numeral.

DD is the day of the month as a numeral.

HH is the hour of the day

MM is the minutes within the hour

SS is the seconds within the minute

Accel_DEFAULT_CAL_XXXXX is used for raw accelerometer data calibrated with default calibration parameters (obtained from the sensor's spec sheet)

Accel _CAL_XXXXX is used for raw accelerometer data calibrated with custom calibration parameters (obtained from the GGIR algorithm, see section 2.6.2GGIR (Activity and Sleep))

Gyro_CAL_XXXXX is used for raw gyroscope data calibrated with default calibration parameters (obtained from the sensor's spec sheet)

PPG_CAL_XXXXX is used for the raw PPG data

GSR_CAL_XXXXX is used for the raw GSR data

NonWearDetection_XXXXX is used for information relating to whether the sensor is being worn or not

Payload_Metadata_XXXXX is used for storing payload information relating to how the sensor data is bundled for transmission to the host

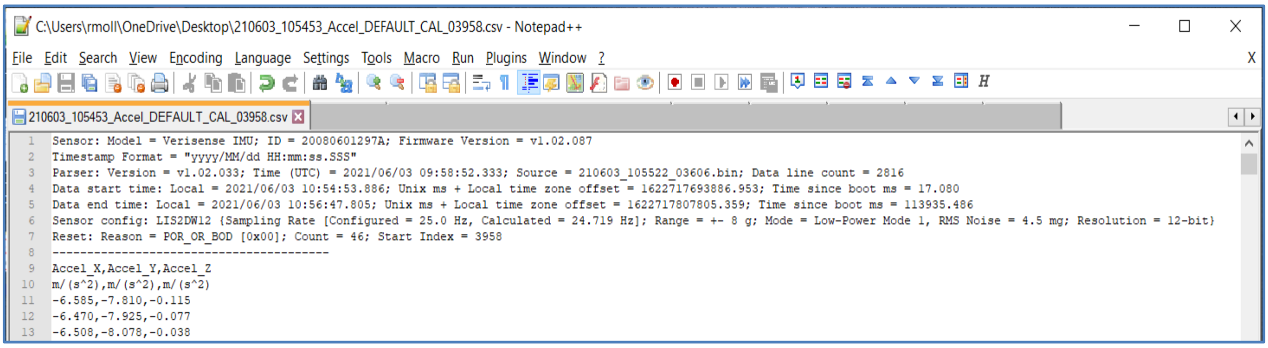
2.5 Raw Data

2.5.1 Accelerometer

2.5.1.1 Example File

The first lines of an example file containing raw data are displayed below. The raw data consists of a header which contains sensor metadata and is followed the data from the sensoritself.

The raw data is stored as a CSV file.




```
Sensor: Model = Verisense IMU; ID = 20080601297A; Firmware Version = v1.02.087
Timestamp Format = "yyyy/MM/dd HH:mm:ss.SSS"
Parser: Version = v1.02.033; Time (UTC) = 2021/06/03 09:58:52.333; Source = 210603_105522_03606.bin; Data line count = 2816
Data start time: Local = 2021/06/03 10:54:53.886; Unix ms + Local time zone offset = 1622717693886.953; Time since boot ms = 17.080
Data end time: Local = 2021/06/03 10:56:47.805; Unix ms + Local time zone offset = 1622717807805.359; Time since boot ms = 113935.486
Sensor config: LIS2DW12 {Sampling Rate [Configured = 25.0 Hz, Calculated = 24.719 Hz]; Range = +- 8 g; Mode = Low-Power Mode 1, RMS Noise = 4.5 mg; Resolution = 12-bit}
Reset: Reason = POR_OR_BOD [0x00]; Count = 46; Start Index = 3958
----------------------------------------
Accel_X,Accel_Y,Accel_Z
m/(s^2),m/(s^2),m/(s^2)
-6.585,-7.810,-0.115
-6.470,-7.925,-0.077
-6.508,-8.078,-0.038
```

Figure 2-3: "Example Accelerometer File

2.5.1.2 Header

The header contains information relating to the raw data for the specific raw data file.

2.5.1.2.1 Sensor

The type of sensor used to collect the data. Valid entries in this field are *Verisense IMU, Verisense PPG* and *Verisense Pulse+*.

2.5.1.2.2 ID

The unique identifier of the sensor, this will be a 12 character alphanumeric string e.g. *20080601297A*.

2.5.1.2.3 Firmware Version

The firmware version that is programmed on the sensor.

2.5.1.2.4 Timestamp Format

The format of the data and time used (*yyyy/MM/dd HH:mm:ss.SSS*), where

yyyy is the year.

MM is the month of the year as a numeral.

dd is the day of the month as a numeral.HH is the hour of the day

mm is the minutes within the hour

ss.SSS is the seconds and milliseconds respectively within the minute.

2.5.1.2.5 Parser

The version number of the parser used to convert the raw binary data intoCSV format.

2.5.1.2.6 Time

The date and time in the format specified by section 2.5.1.2.4Timestamp Formatof when the raw binary data was parsed into CSV format

2.5.1.2.7 Source

The binary file source that the CSV file is parsed from.

2.5.1.2.8 Data line count

The number lines in the file (number of data packets) excluding the header, that the raw data file contains.

2.5.1.2.9 Data start time

2.5.1.2.10 *Local*

The start date and time of the data in the format specified by section 2.5.1.2.4Timestamp Format

2.5.1.2.11 Unix ms + Local time zone offset

The start date and time of the data in UNIX time in milliseconds, the same value as section 2.5.1.2.10*Local* just presented in a different format

2.5.1.2.12 Time since boot ms

Time since the microcontroller started in milliseconds.

2.5.1.2.13 Data end time

2.5.1.2.13.1 *Local*

The end date and time of the data in the format specified by section 2.5.1.2.4Timestamp Format

2.5.1.2.13.2 *Unix ms + Local time zone offset*

The end date and time of the data in UNIX time in milliseconds, the same value as section2.5.1.2.13.1*Local*just presented in a different format

2.5.1.2.13.3 *Time since boot ms*

Time since the microcontroller started in milliseconds.

2.5.1.2.14 Sensor config

The configuration of the accelerometer used to capture the raw data. Note that the Verisense IMU has two accelerometers, LIS2DW12 and LSM6DS3. The LSM6DS3 chip also contains a gyroscope so if the gyroscope is enabled in the Verisense IMU's configuration then the sensor will use LSM6DS3 otherwise LIS2DW12 will be used.

2.5.1.3 Sensor Data

Each data packet is represented on separate line within the CSV file.Channelnames and the units of measurement are included.

2.5.1.3.1 Channel names

Text that describes what the sensor or sensor component that the data is associated with e.g. for accelerometer as it's tri-axial, there's a 3 channel output, *Accel_X, Accel_Y* and *Accel_Z* which represent the X, Y and Z axes respectively of the accelerometer.

2.5.1.3.2 Channel units

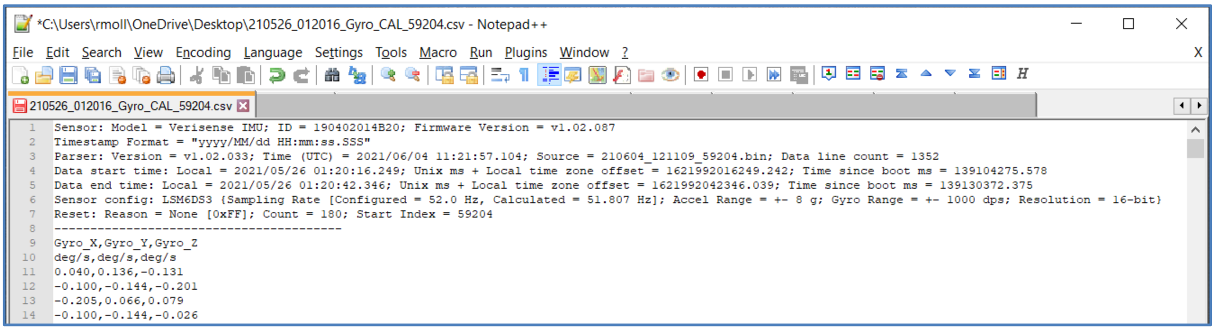
The SI units of the data for each channel isspecified e.g. *m/(s^2)* for accelerometer

2.5.1.3.3 Channel data

Data point for each of the channel names e.g. *-6.585, -7.81, -0.115*which would represent the accelerometer output on the X, Y and Z axes at one point in time

2.5.2 Gyroscope

2.5.2.1 Example File



```
Sensor: Model = Verisense IMU; ID = 190402014B20; Firmware Version = v1.02.087
Timestamp Format = "yyyy/MM/dd HH:mm:ss.SSS"
Parser: Version = v1.02.033; Time (UTC) = 2021/06/04 11:21:57.104; Source = 210604_121109_59204.bin; Data line count = 1352
Data start time: Local = 2021/05/26 01:20:16.249; Unix ms + Local time zone offset = 1621992016249.242; Time since boot ms = 139104275.578
Data end time: Local = 2021/05/26 01:20:42.346; Unix ms + Local time zone offset = 1621992042346.039; Time since boot ms = 139130372.375
Sensor config: LSM6DS3 {Sampling Rate [Configured = 52.0 Hz, Calculated = 51.807 Hz]; Accel Range = +- 8 g; Gyro Range = +- 1000 dps; Resolution = 16-bit}
Reset: Reason = None [0xFF]; Count = 180; Start Index = 59204
----------------------------------------
Gyro_X,Gyro_Y,Gyro_Z
deg/s,deg/s,deg/s
0.040,0.136,-0.131
-0.100,-0.144,-0.201
-0.205,0.066,0.079
-0.100,-0.144,-0.026
```

Figure 2-4: "Example Gyroscope File"

2.5.2.2 File Breakdown

The makeup of the raw gyroscope CSV file is the same as detailed for the accelerometer from sections 2.5.1.2Header to 2.5.1.3.3Channel data other than the sensor configuration and channel names, units and data being specific to the gyroscope as opposed to the accelerometer

2.5.3 PPG

2.5.3.1 Example File

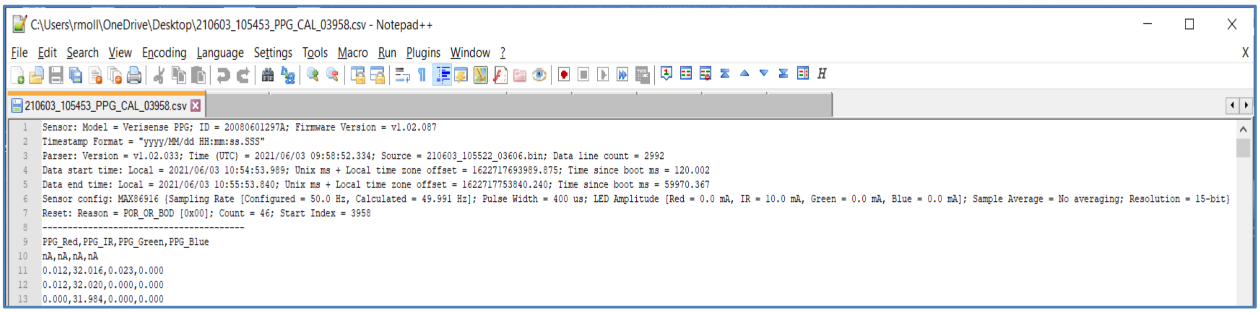


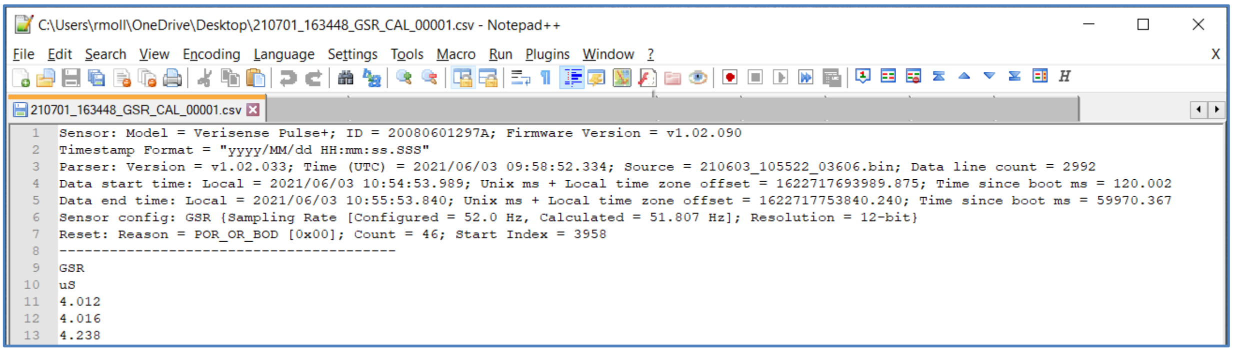
Figure 2-5: "Example PPG File"

2.5.3.2 File Breakdown

The makeup of the raw PPG CSV file is the same as detailed for the accelerometer from sections 2.5.1.2Header to 2.5.1.3.3Channel data other than the sensor configuration and channel names, units and data being specific to the PPG as opposed to the accelerometer. The PPG is capable of outputting four channels with the source of each being an LED, Red/Infrared/Green/Blue.

2.5.4 GSR

2.5.4.1 Example File



```
Sensor: Model = Verisense Pulse+; ID = 20080601297A; Firmware Version = v1.02.090
Timestamp Format = "yyyy/MM/dd HH:mm:ss.SSS"
Parser: Version = v1.02.033; Time (UTC) = 2021/06/03 09:58:52.334; Source = 210603_105522_03606.bin; Data line count = 2992
Data start time: Local = 2021/06/03 10:54:53.989; Unix ms + Local time zone offset = 1622717693989.875; Time since boot ms = 120.002
Data end time: Local = 2021/06/03 10:55:53.840; Unix ms + Local time zone offset = 1622717753840.240; Time since boot ms = 59970.367
Sensor config: GSR {Sampling Rate [Configured = 52.0 Hz, Calculated = 51.807 Hz]; Resolution = 12-bit}
Reset: Reason = POR_OR_BOD [0x00]; Count = 46; Start Index = 3958
----------------------------------------
GSR
uS
4.012
4.016
4.238
```

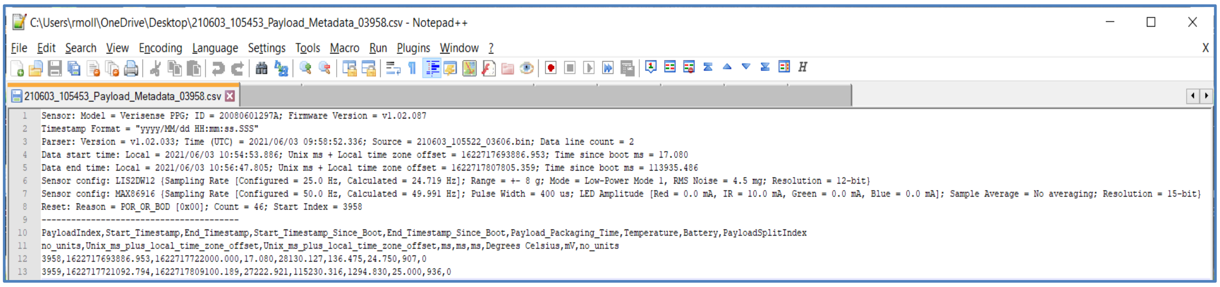
Figure 2-6: "Example GSR File"

2.5.4.2 File Breakdown

The makeup of the raw GSR CSV file is the same as detailed for the accelerometer from sections 2.5.1.2Header to 2.5.1.3.3Channel data other than the sensor configuration and channel names, units and data being specific to the GSR as opposed to the accelerometer

2.5.5 Payload Metadata

2.5.5.1 Example File




```
Sensor: Model = Verisense PPG; ID = 20080601297A; Firmware Version = v1.02.087
Timestamp Format = "yyyy/MM/dd HH:mm:ss.SSS"
Parser: Version = v1.02.033; Time (UTC) = 2021/06/03 09:58:52.336; Source = 210603_105522_03606.bin; Data line count = 2
Data start time: Local = 2021/06/03 10:54:53.886; Unix ms + Local time zone offset = 1622717693886.953; Time since boot ms = 17.080
Data end time: Local = 2021/06/03 10:56:47.805; Unix ms + Local time zone offset = 1622717807805.359; Time since boot ms = 113935.486
Sensor config: LIS2DW12 {Sampling Rate [Configured = 25.0 Hz, Calculated = 24.719 Hz]; Range = +- 8 g; Mode = Low-Power Mode 1, RMS Noise = 4.5 mg; Resolution = 12-bit}
Sensor config: MAX86916 {Sampling Rate [Configured = 50.0 Hz, Calculated = 49.991 Hz]; Pulse Width = 400 us; LED Amplitude [Red = 0.0 mA, IR = 10.0 mA, Green = 0.0 mA, Blue = 0.0 mA]; Sample Average = No averaging; Resolution = 15-bit}
Reset: Reason = POR_OR_BOD [0x00]; Count = 46; Start Index = 3958
----------------------------------------
PayloadIndex,Start_Timestamp,End_Timestamp,Start_Timestamp_Since_Boot,End_Timestamp_Since_Boot,Payload_Packaging_Time,Temperature,Battery,PayloadSplitIndex
no_units,Unix_ms_plus_local_time_zone_offset,Unix_ms_plus_local_time_zone_offset,ms,ms,ms,Degrees Celsius,mV,no_units
3958,1622717693886.953,1622717722000.000,17.080,28130.127,136.475,24.750,907,0
3959,1622717721092.794,1622717809100.189,27222.921,115230.316,1294.830,25.000,936,0
```

Figure 2-7: "Example Payload Metadata File"

2.5.5.2 File Breakdown

The makeup of the Payload Metadata CSV file is the same as detailed for the accelerometer from sections 2.5.1.2Header to 2.5.1.3.3Channel data other than the sensor configuration and channel names, units and data being specific to the Payloads as opposed to the accelerometer. A payload is made up of sensor data and structured optimally for storage and for transmission from sensor to host.

2.5.5.2.1 PayloadIndex

The index of the payload, starting at 0 and ascending to 65535 with rollover back to 0.

2.5.5.2.2 Start_Timestamp

The start time of the payload in UNIX time in milliseconds

2.5.5.2.3 End_Timestamp

The end time of the payload in UNIX time in milliseconds

2.5.5.2.4 Start_Timestamp_Since_Boot

The start time of the payloadrelative to when the microcontroller powered-up in milliseconds

2.5.5.2.5 End_Timestamp_Since_Boot

The end time of the payloadrelative to when the microcontroller powered-up in milliseconds

2.5.5.2.6 Payload_Packaging_Time

The time in milliseconds that the microcontroller takes to package the payload

2.5.5.2.7 Temperature

The temperature of the microcontroller in degrees Celsius

2.5.5.2.8 Battery

The measured battery voltage in millivolts. The battery voltage is 12-bit value with a full-scale value representing 1.8 volts

2.5.5.2.9 PayloadSplitIndex

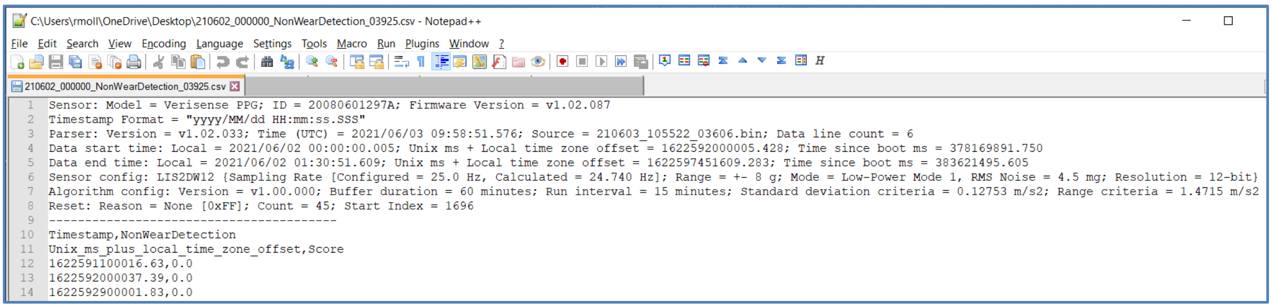
An integer value representing the index of the payload if the payload was split. As the file parser splits sensor data at midday/midnight transition points, it is possible that a single payload may be split across files so an index is used to track this for background data processing services. A value of 0 represents that the payload entry is not split, 1 represents the first half of a split and 2 represents the second half of a split.

2.6 Derived Data

2.6.1 Non-Wear Detection

The non-wear detection algorithm uses the raw accelerometer data to decide whether the sensor is being worn or not.

2.6.1.1 Example File




```
Sensor: Model = Verisense PPG; ID = 20080601297A; Firmware Version = v1.02.087
Timestamp Format = "yyyy/MM/dd HH:mm:ss.SSS"
Parser: Version = v1.02.033; Time (UTC) = 2021/06/03 09:58:51.576; Source = 210603_105522_03606.bin; Data line count = 6
Data start time: Local = 2021/06/02 00:00:00.005; Unix ms + Local time zone offset = 1622592000005.428; Time since boot ms = 378169891.750
Data end time: Local = 2021/06/02 01:30:51.609; Unix ms + Local time zone offset = 1622597451609.283; Time since boot ms = 383621495.605
Sensor config: LIS2DW12 {Sampling Rate [Configured = 25.0 Hz, Calculated = 24.740 Hz]; Range = +- 8 g; Mode = Low-Power Mode 1, RMS Noise = 4.5 mg; Resolution = 12-bit}
Algorithm config: Version = v1.00.000; Buffer duration = 60 minutes; Run interval = 15 minutes; Standard deviation criteria = 0.12753 m/s2; Range criteria = 1.4715 m/s2
Reset: Reason = None [0xFF]; Count = 45; Start Index = 1696
----------------------------------------
Timestamp,NonWearDetection
Unix_ms_plus_local_time_zone_offset,Score
1622591100016.63,0.0
1622592000037.39,0.0
1622592900001.83,0.0
```

Figure 2-8: "Example Non-Wear Detection File"

2.6.1.1.1 Timestamp (Unix_ms_plus_local_time_zone_offset)

The time of the non-wear detection computation in UNIX time in milliseconds local to where the sensor was recording data

2.6.1.1.2 NonWearDetection (Score)

An integer value between 0-3 which indicates how many of the three accelerometer axes are detected as having no motion. A value of 0 indicates motion has been detected on all three axes, a value of 1 indicates one axis has been detected as having motion.

2.6.2 GGIR (Activity and Sleep)

GGIR is an R-package to process multi-day raw accelerometer data for physical activity and sleep research[1]. Details of the GGIR algorithm and data dictionary are available in another document, ASM- DOC07Verisense Data GGIR Dictionary.

2.6.3 PPGtoHR

The PPGtoHR algorithm uses the raw PPG data to produce heart rate, inter-beat-interval and HRV metrics.

---

[1] https://cran.r-project.org/web/packages/GGIR/vignettes/GGIR.html

2.6.3.1 Example File

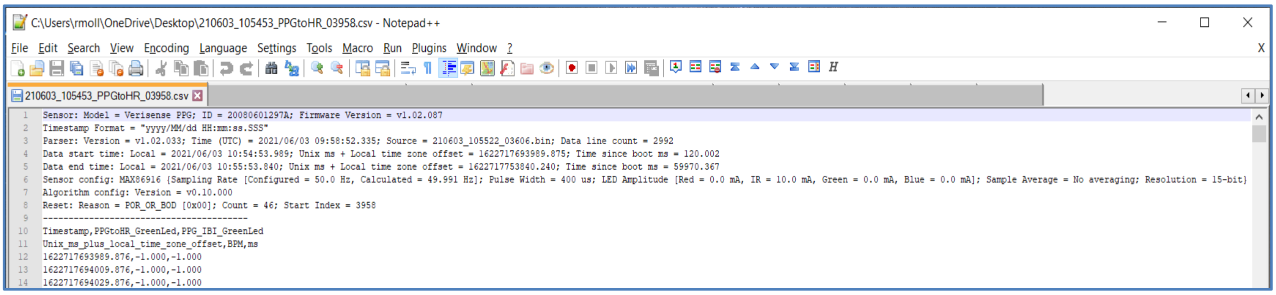


Figure 2-9: "Example PPGtoHR File"

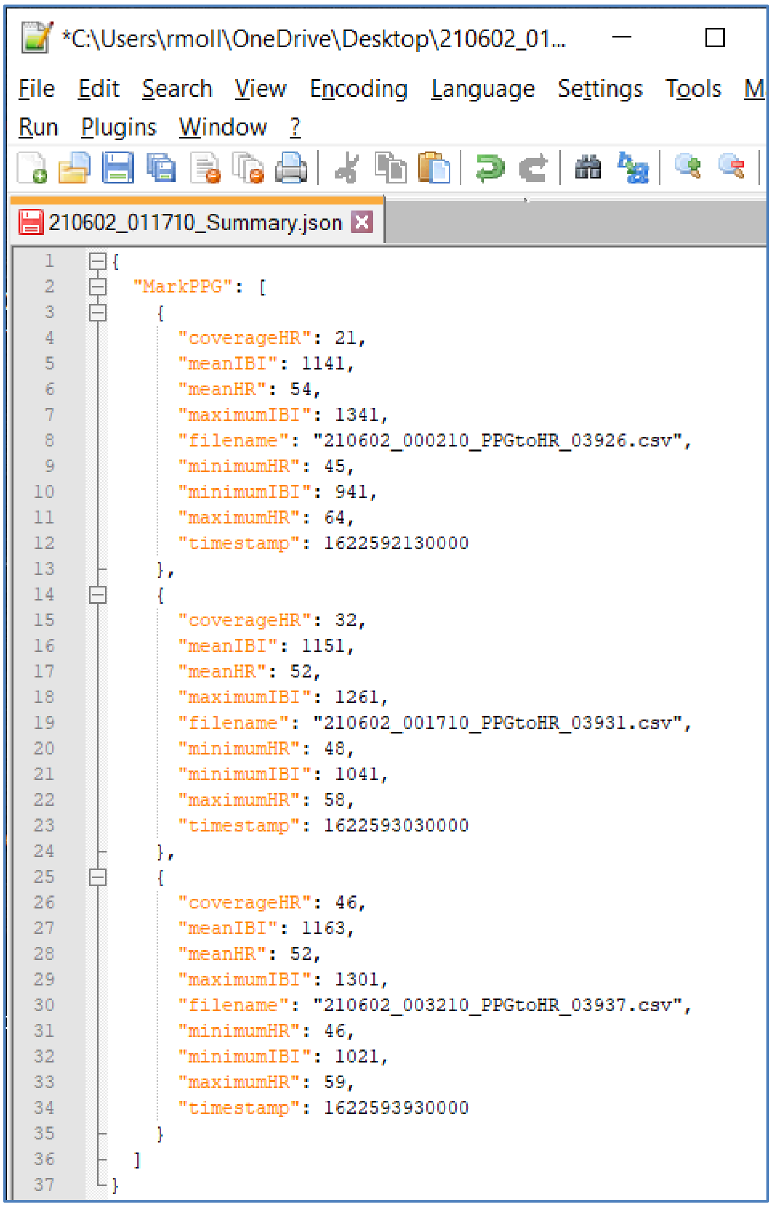
Note: A value of -1 means that the algorithm was unable to produce a valid result which may happen if there is a data quality issue e.g. if the sensor isn't being worn or there is too much motion artifact in the signal

2.6.3.2 File Breakdown

The makeup of the PPGtoHR CSV file is the same as detailed for the accelerometer from sections 2.5.1.2Header to 2.5.1.3.3Channel data other than the sensor configuration and channel names, units and data being specific to the PPGtoHR as opposed to the accelerometer.

2.6.3.3 PPGtoHR Summary

A summary of PPGtoHR metrics are provided for each day of PPG data and are provided in a JSON file




```
{
  "MarkPPG": [
    {
      "coverageHR": 21,
      "meanIBI": 1141,
      "meanHR": 54,
      "maximumIBI": 1341,
      "filename": "210602_000210_PPGtoHR_03926.csv",
      "minimumHR": 45,
      "minimumIBI": 941,
      "maximumHR": 64,
      "timestamp": 1622592130000
    },
    {
      "coverageHR": 32,
      "meanIBI": 1151,
      "meanHR": 52,
      "maximumIBI": 1261,
      "filename": "210602_001710_PPGtoHR_03931.csv",
      "minimumHR": 48,
      "minimumIBI": 1041,
      "maximumHR": 58,
      "timestamp": 1622593030000
    },
    {
      "coverageHR": 46,
      "meanIBI": 1163,
      "meanHR": 52,
      "maximumIBI": 1301,
      "filename": "210602_003210_PPGtoHR_03937.csv",
      "minimumHR": 46,
      "minimumIBI": 1021,
      "maximumHR": 59,
      "timestamp": 1622593930000
    }
  ]
}
```


Figure 2-10: "Example PPGtoHR Summary File"

3 Verisense Logs

The Verisense system logs the status message from both the base station and the sensor for each participant.

3.1 Base Station Logs

3.1.1 File Name Convention

The filename of each base station log file is Unix time when the log file was created e.g. *1622216804448.txt*.

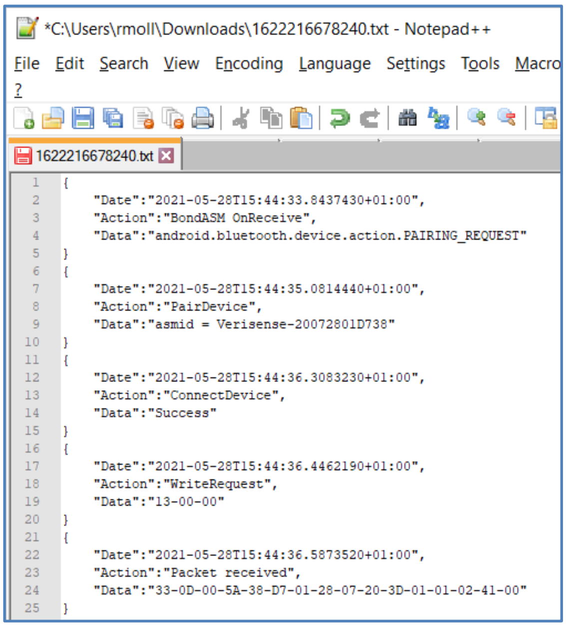
3.1.2 File Breakdown

3.1.2.1 AutoSync

The AutoSync operation is in charge of various operations such as

- Configuration of the Verisense Sensor via BLE
- DFU of the Verisense Sensor via BLE
- Retrieval of sensor specific info such as the Status and Pending Events via BLE
- Data retrieval of logged sensor data via BLE, this data is saved to a bin file
- Uploading of bin and log files to the S3 server

3.1.2.1.1 Example File



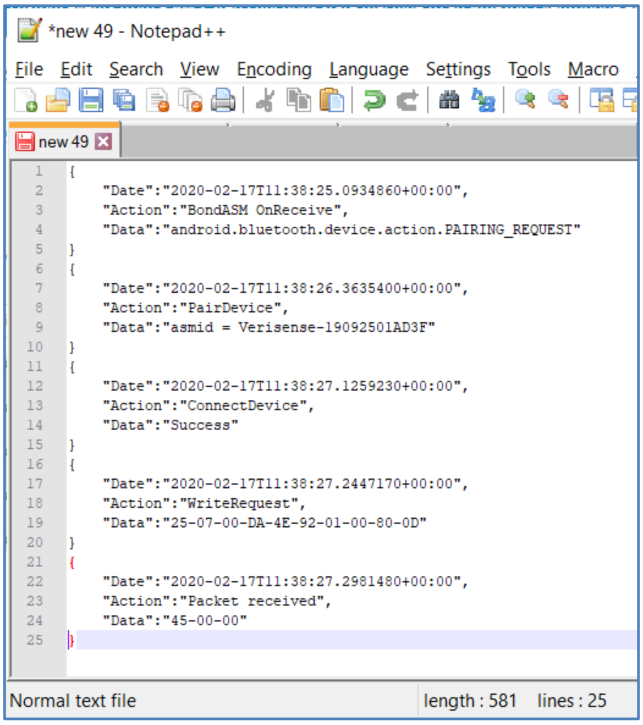

```
{
    "Date":"2021-05-28T15:44:33.8437430+01:00",
    "Action":"BondASM OnReceive",
    "Data":"android.bluetooth.device.action.PAIRING_REQUEST"
}
{
    "Date":"2021-05-28T15:44:35.0814440+01:00",
    "Action":"PairDevice",
    "Data":"asmid = Verisense-20072801D738"
}
{
    "Date":"2021-05-28T15:44:36.3083230+01:00",
    "Action":"ConnectDevice",
    "Data":"Success"
}
{
    "Date":"2021-05-28T15:44:36.4462190+01:00",
    "Action":"WriteRequest",
    "Data":"13-00-00"
}
{
    "Date":"2021-05-28T15:44:36.5873520+01:00",
    "Action":"Packet received",
    "Data":"33-0D-00-5A-38-D7-01-28-07-20-3D-01-01-02-41-00"
}
```

Figure 3-1: "Example AutoSync Log File"

3.1.2.2 Pairing

The pairing log files contain information on the sensor pairing events that occurred on the base station.

3.1.2.2.1 Example File




```
{
    "Date":"2020-02-17T11:38:25.0934860+00:00",
    "Action":"BondASM OnReceive",
    "Data":"android.bluetooth.device.action.PAIRING_REQUEST"
}
{
    "Date":"2020-02-17T11:38:26.3635400+00:00",
    "Action":"PairDevice",
    "Data":"asmid = Verisense-19092501AD3F"
}
{
    "Date":"2020-02-17T11:38:27.1259230+00:00",
    "Action":"ConnectDevice",
    "Data":"Success"
}
{
    "Date":"2020-02-17T11:38:27.2447170+00:00",
    "Action":"WriteRequest",
    "Data":"25-07-00-DA-4E-92-01-00-80-0D"
}
{
    "Date":"2020-02-17T11:38:27.2981480+00:00",
    "Action":"Packet received",
    "Data":"45-00-00"
}
```

Figure 3-2: "Example Pairing Log File"

3.1.2.3 RealmDBLog

The realm DB log is used to log any App related informationwhich can be looked back on in order to gain insight as to the status/condition of the app at a given time. This can be useful in debugging various issues such as

- User error
- BLE
- S3
- REST API

This logged entries in the Database is converted into a log file during the Verisense sensor sync process, where it is eventually uploaded to the S3 Server.

3.1.2.3.1 Example File

Contents too large to represent in a document, contact Shimmer for example file if required

3.2 Sensor Logs

3.2.1 Filename Convention

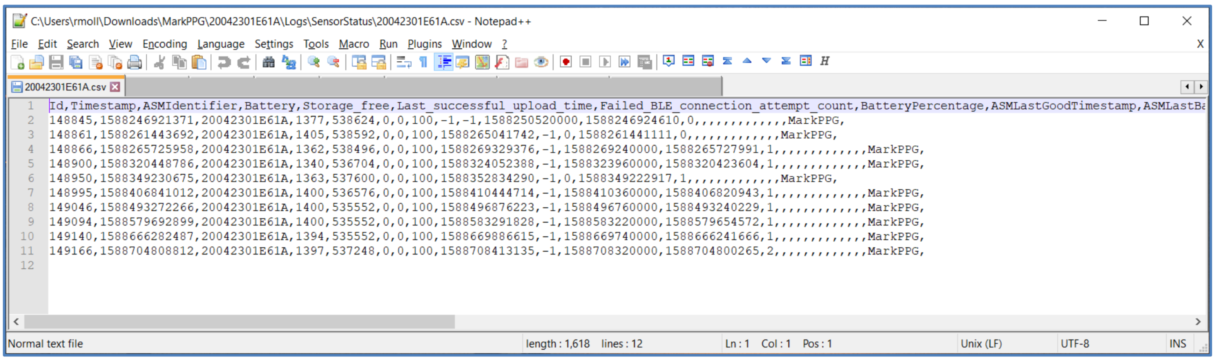
The filename of each sensor log file is the Bluetooth MAC addressof the sensor e.g. *20072801D738.txt*

3.2.2 File Breakdown

3.2.2.1 Sensor Status

The sensor status file contains a row for each status message that the sensor sends

3.2.2.1.1 Example File



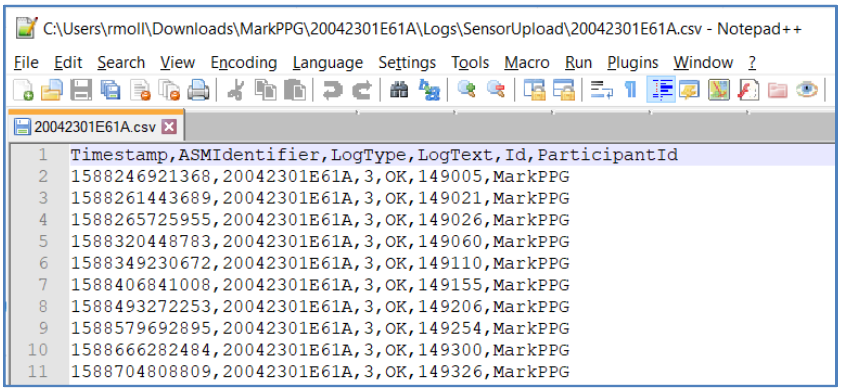

```
Id,Timestamp,ASMIdentifier,Battery,Storage_free,Last_successful_upload_time,Failed_BLE_connection_attempt_count,BatteryPercentage,ASMLastGoodTimestamp,ASMLastBa
148845,1588246921371,20042301E61A,1377,538624,0,0,100,-1,-1,1588250520000,1588246924610,0,,,,,,,,,,,,,MarkPPG,
148861,1588261443692,20042301E61A,1405,538592,0,0,100,1588265041742,-1,0,1588261441111,0,,,,,,,,,,,,,MarkPPG,
148866,1588265725958,20042301E61A,1362,538496,0,0,100,1588269329376,-1,1588269240000,1588265727991,1,,,,,,,,,,,,,MarkPPG,
148900,1588320448786,20042301E61A,1340,536704,0,0,100,1588324052388,-1,1588323960000,1588320423604,1,,,,,,,,,,,,,MarkPPG,
148950,1588349230675,20042301E61A,1363,537600,0,0,100,1588352834290,-1,0,1588349222917,1,,,,,,,,,,,,,MarkPPG,
148995,1588406841012,20042301E61A,1400,536576,0,0,100,1588410444714,-1,1588410360000,1588406820943,1,,,,,,,,,,,,,MarkPPG,
149046,1588493272266,20042301E61A,1400,535552,0,0,100,1588496876223,-1,1588496760000,1588493240229,1,,,,,,,,,,,,,MarkPPG,
149094,1588579692899,20042301E61A,1400,535552,0,0,100,1588583291828,-1,1588583220000,1588579654572,1,,,,,,,,,,,,,MarkPPG,
149140,1588666282487,20042301E61A,1394,535552,0,0,100,1588669886615,-1,1588669740000,1588666241666,1,,,,,,,,,,,,,MarkPPG,
149166,1588704808812,20042301E61A,1397,537248,0,0,100,1588708413135,-1,1588708320000,1588704800265,2,,,,,,,,,,,,,MarkPPG,
```

Figure 3-3: "Example Sensor Status Log File"

3.2.2.2 Sensor Upload

The sensor upload file contains a row for each data upload event that the sensor performs

3.2.2.2.1 Example File



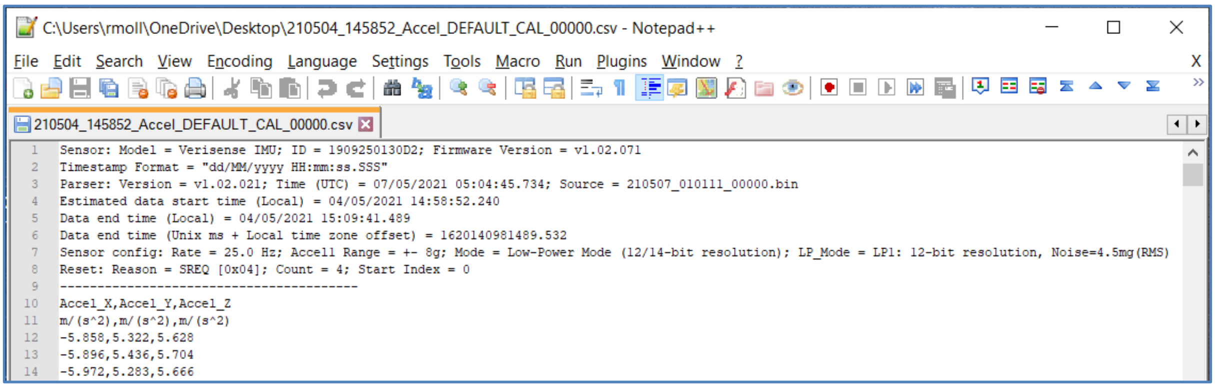

```
Timestamp,ASMIdentifier,LogType,LogText,Id,ParticipantId
1588246921368,20042301E61A,3,OK,149005,MarkPPG
1588261443689,20042301E61A,3,OK,149021,MarkPPG
1588265725955,20042301E61A,3,OK,149026,MarkPPG
1588320448783,20042301E61A,3,OK,149060,MarkPPG
1588349230672,20042301E61A,3,OK,149110,MarkPPG
1588406841008,20042301E61A,3,OK,149155,MarkPPG
1588493272253,20042301E61A,3,OK,149206,MarkPPG
1588579692895,20042301E61A,3,OK,149254,MarkPPG
1588666282484,20042301E61A,3,OK,149300,MarkPPG
1588704808809,20042301E61A,3,OK,149326,MarkPPG
```

Figure 3-4: "Example Sensor Upload Log File"

A. LEGACY FILE HEADERS

A-1. FILE PARSER V1.02.021

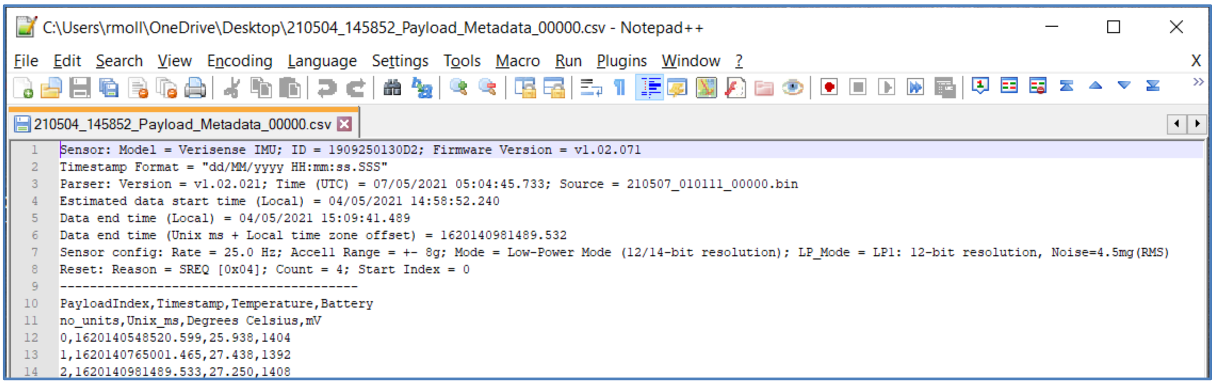
A-1.1. ACCELEROMETER




```
Sensor: Model = Verisense IMU; ID = 1909250130D2; Firmware Version = v1.02.071
Timestamp Format = "dd/MM/yyyy HH:mm:ss.SSS"
Parser: Version = v1.02.021; Time (UTC) = 07/05/2021 05:04:45.734; Source = 210507_010111_00000.bin
Estimated data start time (Local) = 04/05/2021 14:58:52.240
Data end time (Local) = 04/05/2021 15:09:41.489
Data end time (Unix ms + Local time zone offset) = 1620140981489.532
Sensor config: Rate = 25.0 Hz; Accell Range = +- 8g; Mode = Low-Power Mode (12/14-bit resolution); LP_Mode = LP1: 12-bit resolution, Noise=4.5mg(RMS)
Reset: Reason = SREQ [0x04]; Count = 4; Start Index = 0
----------------------------------------
Accel_X,Accel_Y,Accel_Z
m/(s^2),m/(s^2),m/(s^2)
-5.858,5.322,5.628
-5.896,5.436,5.704
-5.972,5.283,5.666
```

Figure 3-5: "Example Accelerometer File (Legacy)"

A-1.2. PAYLOAD METADATA




```
Sensor: Model = Verisense IMU; ID = 1909250130D2; Firmware Version = v1.02.071
Timestamp Format = "dd/MM/yyyy HH:mm:ss.SSS"
Parser: Version = v1.02.021; Time (UTC) = 07/05/2021 05:04:45.733; Source = 210507_010111_00000.bin
Estimated data start time (Local) = 04/05/2021 14:58:52.240
Data end time (Local) = 04/05/2021 15:09:41.489
Data end time (Unix ms + Local time zone offset) = 1620140981489.532
Sensor config: Rate = 25.0 Hz; Accell Range = +- 8g; Mode = Low-Power Mode (12/14-bit resolution); LP_Mode = LP1: 12-bit resolution, Noise=4.5mg(RMS)
Reset: Reason = SREQ [0x04]; Count = 4; Start Index = 0
----------------------------------------
PayloadIndex,Timestamp,Temperature,Battery
no_units,Unix_ms,Degrees Celsius,mV
0,1620140548520.599,25.938,1404
1,1620140765001.465,27.438,1392
2,1620140981489.533,27.250,1408
```

Figure 3-6: "Example Payload Metadata File (Legacy)"