The Slogger Manual

Zifei Zhong
(draft)

October 23, 2024

Contents

1 Design Purpose

The Slogger App is designed to log raw data from Accelerometer, Gyroscope, and Heart Rate Sensor on the Google Pixel Watches. The data logging frequency is configurable in the settings of the Slogger App. Data is logged into local files on the watches. The files can be uploaded to a cloud server.

2 The Slogger App Version

As of the time of writing this document, the current version of the Slogger App is 1.6.1.

3 The Slogger User Interface

3.1 Main Panel

The Slogger's main panel consists of a status display area and 4 buttons: *Start*, *Stop*, *Upload*, and *Configure*. Here is a summary of the functionality of them:

1. The status display area shows the state of the App. The 4 states (*Idle, Logging, Timing*, and *Uploading*) and their transitions are described in Section 4.

2. The *Start* button: start logging.

3. The *Stop* button: stop logging.

4. The *Upload* button: upload local data files to the server.

5. The *Configure* button: configure logging parameters.

We should always start the Slogger App by configuring its logging parameters. For example, we should configure the start time so that the App is able to schedule logging events to start at the configured time once the *Start* button is clicked.

It is worth noting that a handful of parameters need to be configured before we can start an experiment. The configuration process is detailed in Section 3.2.

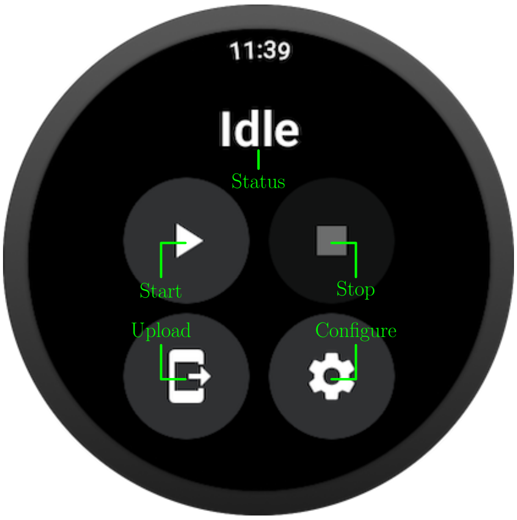
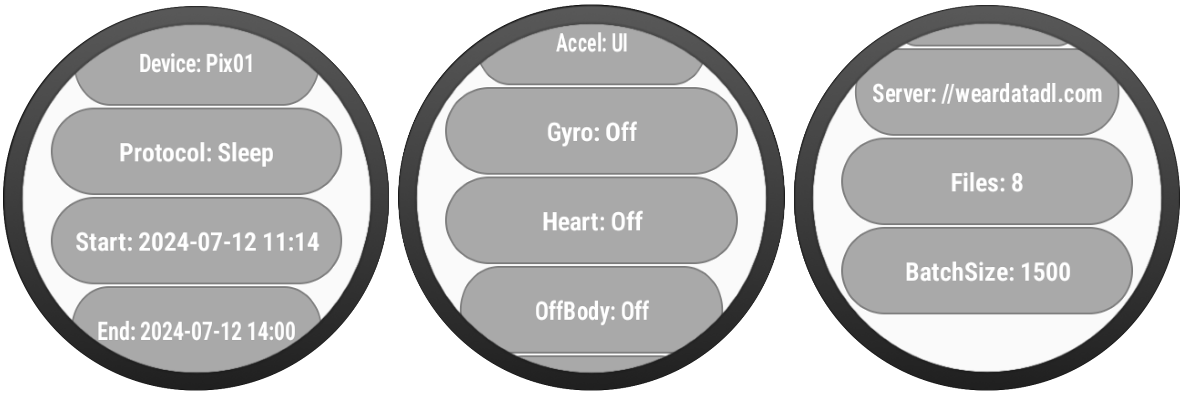


Figure 1: Slogger v1.6.1 main operation panel

3.2 Configuration

If we click the Configure button, it will load the configuration interface, as illustrated in Figure 2. The configuration interface shows a list of parameters. The *operator* can scroll up/down to set each parameter. In the rest of this document, we refer to whoever configures the App as an *operator*, and whoever wears the watch to collect data as a *user*. A typical workflow is that the operator configures the App properly before giving the watch to a user to wear.



Device: Pix01
Protocol: Sleep
Start: 2024-07-12 11:14
End: 2024-07-12 14:00

Accel: UI
Gyro: Off
Heart: Off
OffBody: Off

Server: //weardatadl.com
Files: 8
BatchSize: 1500

Figure 2: Slogger v1.6.1 configuration interface

The parameters include:

1. **Device**: the device name.

   The operator can use a string that uniquely represents the device, for example: “Sam01”, “Pix05”, etc. Alphabets (‘A’ to ‘Z’, ’a’ to ’z’) and numbers ‘0’ to ‘9’ are preferred. No underscore “_” is permitted.

   For example, we carried multiple Google Pixel watches in our experiments. We used “Pix01”, “Pix02”, $\cdots$ as the device names.

2. **Protocol**: the name that characterizes the user activity.

   For example, the operator can set the protocol name as “Sleep” for an experiment that aims to collect data in the night when the users are sleeping.

3. **Start**: the time to start data logging.

   The operator should configure both the **Date** (YY-MM-DD) and **Time** (hh-mm-ss). Figure 3 shows an example time configuration where the start time is configured as “August 20, 2024 09:15:00”.

   It’s the operator’s responsibility to set valid date and time. For example, the operator might configure the date incorrectly to “June 31, 2024”. We suggest the operators double checking the validity of the configured parameters before starting experiments.

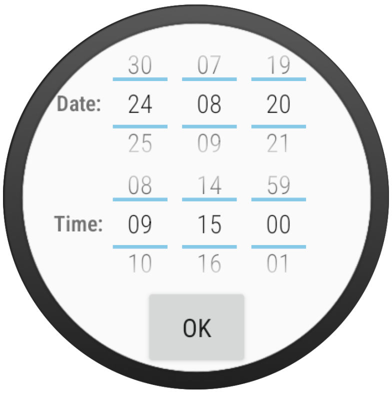
   The Slogger App checks the validity of the time configuration when it starts logging. A valid time configuration would be starting from a time in the past (or future) and ending in the future (after the start time). The logging starts only when the time validity check is passed successfully.

4. **End**: the time to end data logging.

   The operator should configure both the **Date** (YY-MM-DD) and **Time** (hh-mm-ss). It’s the operator’s responsibility to set valid date and time.

5. **Accel**: parameters for the Accelerometer sensor.

   The operator can configure the mode (frequency) of the Accelerometer. There are 5 modes: **Off, Normal, UI, Game**, and **Fastest**. If the mode is **Off**, the sensor doesn’t log data. For other



Date: 24 08 20

Time: 09 15 00

Figure 3: Start time configured as "08/20/2024 09:15:00".

Table 1: Accelerometer frequency vs. mode for the Google Pixel Watches

Mode	Approximate Frequency (Hz)
Normal	5
UI	16
Game	50
Fastest	$\geq$ 100

modes, the sensor will be logging at different frequencies when collecting data. Table 1 shows the Accelerometer frequencies in different modes for the Google Pixel Watches. Watches other than the Google Pixel Watches might have different frequencies in these modes.

6. **Gyro**: parameters for the Gyroscope sensor.

   The operator can configure the mode (frequency) of the Gyroscope. This is similar to the Accelerometer configuration.

7. **Heart**: parameters for the heart rate sensor.

   If the heart rate sensor is configured to **Off**, it doesn't log data. If the heart rate is configured to **Normal**, it will collect data at the frequency of 1 Hz (1 data reading per second).

8. **OffBody**: parameters for the proximity sensor.

   If the proximity sensor is configured to **Off**, it doesn't log data. If the proximity sensor is configured to **Normal**, it will report data/status whenever the device is on/off body.

9. **Server**: the server address for uploading data files.

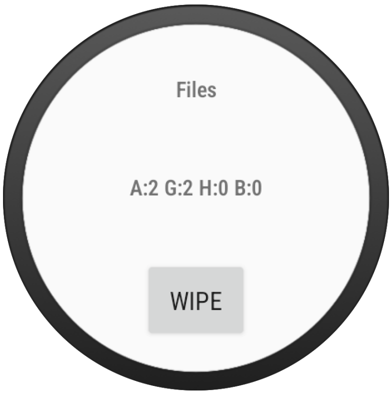
   The operator can set the server address for uploading data files. The server address can be either a **http** or **https** address. The operator has to set both a **Base URL** and a **Suffix**, which together form a valid URL. It's the operator's responsibility to make sure the server has implemented the necessary interfaces to accept data uploading from the Slogger App.

   Please contact the Slogger App developer for details of implementing the necessary interfaces at the server side.

10. **Files**: the number of data files on the device.

    The operator can see the total number of files on the device. The files are categorized as "H: heart rate data files", "A: accelerometer data files", "G: gyroscope data files", and "B: proximity sensor data files".

    Figure 4 shows the **Files** panel which indicates there are 2 Accelerometer data files and 2 Gyroscope data files. There is also a **WIPE** button which can delete all logging data files.



Files

A:2 G:2 H:0 B:0

Figure 4: The Files panel: 2 Accelerometer files and 2 Gyroscope files

Table 2: An example configuration

Parameters	**Values**
Device	Pix01
Procotol	Sleep
Start	Date: `24-08-20` Time: `21:15:00`
End	Date: `24-08-21` Time: `06:30:00`
Accel	Game
Gyro	UI
Heart	Normal
OffBody	Normal
Server	Base: `https://weardatadl.com:8443/` Suffix: `android_xfer/`
BatchSize	1500

11. **BatchSize**: the batch size for collecting sensor data in batch mode. The default value is 1500. Please consult the Slogger App developer before changing the batch size.

3.2.1 An Example Configuration

Table 2 shows an example configuration.

In this configuration, the device name is “Pix01” and the protocol name is “Sleep”. This possibly indicates that it is configured for an experiment to collect data during sleeping with a Pixel watch.

The experiment should start in the night of August 20, 2024 at 21:15 and end in the morning at 6:30 of the next day.

The Accelerometer is set to collect data at 50 Hz (mode: Game), the Gyroscope is set to collect data at 16 Hz (mode: UI), and the heart rate sensor is set to collect data at 1 Hz (mode: Normal). The proximity sensor is also set on to report data whenever the participant wears (or takes off) the device.

The server address is set to URL: `https://weardatadl.com:8443/android_xfer`. The batch size is set to default value 1500.

4 State Transition

It’s better to understand the states and transitions of the Slogger App in order to operate it correctly. It is worth noting that certain operations are permitted under specific states. For Slogger v1.6.1, there are only 4 states: Idle, Timing, Logging, and Uploading.

4.1 Operations permitted at each state

The operator should know the specific operations which are permitted at a given state. Table 3 shows all the states and corresponding operations permitted for each state. When the Slogger App is in either Timing, Logging, or Uploading state, the only operation that is permitted is Stop. When the App is in Idle state, three operations are permitted: Start (logging), Upload, and Configure.

Table 3: Operations permitted at each state

States	**Operations**
Idle	Start, Upload, Configure
Timing	Stop
Logging	Stop
Uploading	Stop

Figure 5 illustrates the state transitions of the Slogger App. When an operator launches the Slogger App, it is in the Idle state. When the Start button is clicked, the Slogger App starts or schedules the logging events based on the configured time: If the configured start time is in the past, the logging starts immediately and the App enters the Logging state; If the configured start time is in the future, the App's scheduler will schedule the logging and the App enters the Timing state. The App ends its Timing state and enters the Logging state when the scheduler timeout.

The App ends its Logging state and enters into the Idle state whenever a timeout event happens (the End time is reached) or the Stop button is clicked. When the App is in its Idle state, the operator can click the Upload button to start uploading local files to the server. The App is in the Uploading state during the uploading process. The uploading state ends when it finishes uploading or the Stop button is clicked.

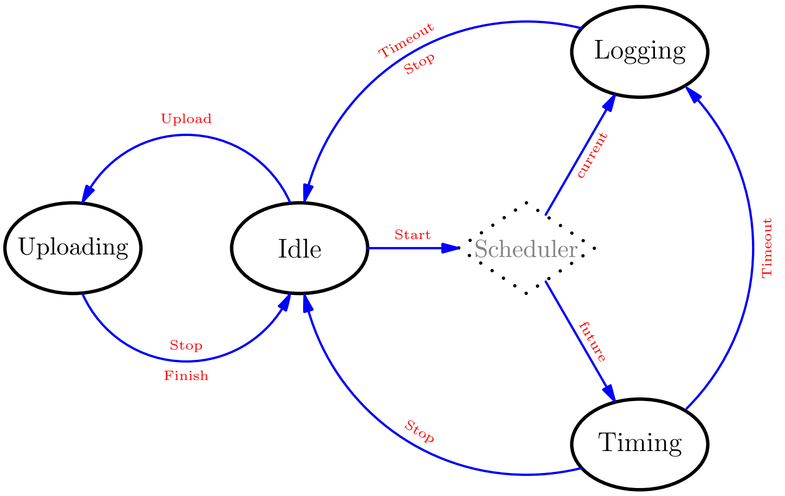
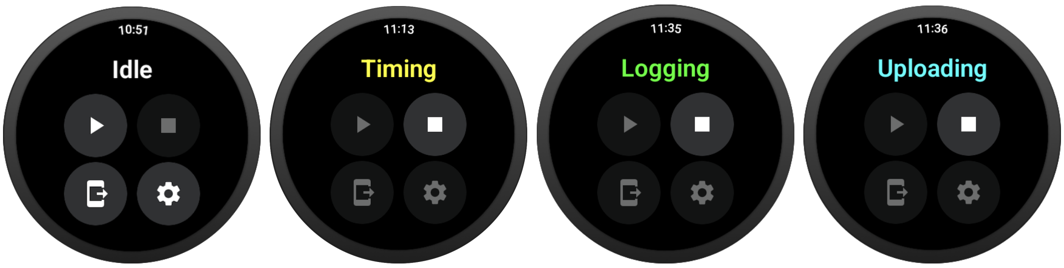


Figure 5: State Transition Diagram

In either of the 4 states, the Slogger App displays the state on the main panel, as illustrated in Figure 6.

5 Logging

No other operations are permitted when the App is in the logging states, except the Stop.



(a) Idle (b) Timing (c) Logging (d) Uploading

Figure 6: The 4 states of the Slogger App

5.1 Naming convention for logging files

The filename of a logging file contains information such as the device name, activity type, sensor type, frequency, start timestamp, device local timestamp, etc.

The naming convention is described in the following:

1. Every logging file should have a suffix “`.csv`”.

2. The main filename of a logging file contains the following 7 fields:
   `deviceName, activityName, sensorType, frequency, seqNum, timeMill, localTimeMill`.

3. The filename is a concatenation of the 7 fields in order by using the underscore symbol: “`_`”. Given an example where the 7 fields have the following values

   ```
   deviceName: Pix01
   activityName: Sleep
   sensorType: Accel
   frequency: 50
   seqNum: 1
   timeMill: 1706040084148
   localTimeMill: 9637320
   ```

   the filename of the logging file is therefore:
   `Pix01_Sleep_Accel_50_1_1706040084148_9637320.csv`

6 File Transfer

The file transfer process uploads all local files sequentially to the server according to their sequence numbers.

The operator should make sure the Watch is connected to the Internet before uploading. It’s a good habit to check the number of files on the device to have a proper estimate of the time needed for uploading.

6.1 File Server Configuration

The operator should configure the file server address properly before starting file transfer. Please refer to Section 3.2 for the configuration parameters. Table 2 provides an example configuration for reference purposes.

6.2 Upload All Logging Files

Since the Slogger App doesn't keep track of which files have been uploaded (or which files don't), once the uploading process starts, it will upload all the logging files on the device to the server (including some the files that may be uploaded before). Therefore, it's a good habit to delete all logging files once the operator confirms that the uploading is completed successfully.

6.3 Uploading Files with Correct Mode

When upload files to the server, the watch might go to idle/sleep mode to save battery. In this case, the file uploading will be halted and some files might not be uploaded to the server. To make sure that the Slogger App can upload all files to the server, it's recommended to turn the watch into 'display always on' mode, so that the watch will always have display on. Once the uploading is completed, the watch's 'display always on' mode can be turned off.

6.4 Format of Raw Data Files

The raw data from the Accelerometer, Gyroscope, Heart Rate Sensor, and OffBody (Proximity) Sensor is stored in local files in different format. Each file has a maximum of 6000 records, e.g., 6000 rows of data.

1. Accelerometer:
   The Accelerometer data record consists of 4 data columns:
   `LocalTimestamp, x, y, z`
   The `LocalTimestamp` is the Pixel watch's local timestamp in nanoseconds.
   The `x, y, z` are the acceleration in three directions in m/s^2.

2. Gyroscope:
   The Gyroscope data record consists of 4 data columns:
   `LocalTimestamp, x, y, z`
   The `x, y, z` are the angle velocity in three directions in $radian/s$.

3. Heart Rate:
   The Heart Rate sensor data record consists of 2 data columns:
   `LocalTimestamp, BPM`
   `BPM`: beats per minute.

4. Presence:
   The Proximity sensor data record (a.k.a, Offbody or Presence) consists of 2 data columns:
   `LocalTimestamp, 0/1`
   0: the device is offbody; 1: the device is onbody.

7 File Server

7.1 Pixel Stream Filter

The Pixel Stream Filter is a unit designed to process a set of logging files and output a single merged data file. The Pixel Watch sensors are quite stable, they don't generate much junk data that needs to be filtered (discarded). The Pixel Stream Filter is mainly used to process the local timestamp of each data record and convert it to a wall clock timestamp.

7.2 Local Time vs. Wall Clock Time

When the data is stored in local files, each data record has a local timestamp. For each local data file, the file name itself has the wall clock time for the first data record of that file. The server aligns the local timestamps to the wall clock time in the following way:

1. Sorts all files in ascending order based on the file sequence number.
2. Takes the first file in that sequence, gets the wall clock from the file name.
3. Takes the first data record from the first file, and calculate the difference between the wall clock time and the local timestamp of the first data record.
4. Add the difference to each local timestamp to convert it to a wall clock timestamp.

The server processes all local files and merges them into a single file in which each data record has its local timestamp and wall clock timestamp. Users can download the processed data from the server at later times.

8 Logging Trace

The Slogger App has a logging trace to record the activities associated with logging. The trace provides useful information for debugging. The trace records the time when the *Start* button is clicked, the battery level at the moment, and the scheduler's scheduled times for the logging to start and stop. This is useful because when an operator deploys the devices into experiments, because timely starting/ending of an experiment is essential to the success of the experiment. Figure 7 shows the content of an example trace log file.

Example: Log Trace

```
..
2024-08-26T15:00:19 -> Start Timing ... APP: 1.6.1; Battery: 100\%.
2024-08-26T15:00:19 -> start: 2024-08-26T18:00, end: 2024-08-28T06:00
2024-08-26T15:00:19 -> [StartLogging] starts in 1724713199240 ms.
2024-08-26T15:00:19 -> [StopLogging] starts in 1724842799245 ms.
2024-08-26T15:00:34.238208 -> mainActivity: onPause called.
...
2024-08-26T15:50:10.026191 -> mainActivity: onResume called.
2024-08-26T17:59:59.383169 -> StateReceiver: LOGGING
2024-08-26T17:59:59.387165 -> battery: 95%
2024-08-26T18:04:07.590720 -> mainActivity: onResume called.
2024-08-26T18:04:10.597083 -> mainActivity: onPause called.
2024-08-28T06:00:30.335805 -> StateReceiver: IDLE
2024-08-28T06:00:30.342616 -> battery: 15%
...
```

Figure 7: Content of an example log trace file