A. Artifact Appendix

A.1 Abstract

LiT is a toothbrushing monitoring system based on a commercial LED toothbrush. LiT relies on commercial blue light sterilization toothbrushes, uses blue LEDs as transmitters, and only requires 2 low-cost photosensors to be mounted on the toothbrush head as receivers. LiT can monitor 16 Bass technique surfaces by analyzing the dynamic light intensity change.

This artifact includes the workflow instruction for making LiT run successfully, the source files and deployment guidance for the hardware, and the source code for signal processing.

A.2 Artifact check-list (meta-information)

- **Program:**

  (1) Arduino burning code (used to control the photosensors to collect the brushing signal),

  (2) Matlab code (used to process the collected signal to distinguish the toothbrushing surfaces).

- **Run-time environment:**

  (1) Arduino UNO,

  (2) Windows Personal computer with Matlab.

- **Hardware:**

  (1) Modified toothbrush hardware (containing Abitelax F7 blue light sterilization electric toothbrush, Printed flexible PCB, Arduino UNO, DuPont wire, and Breadboard),

  (2) Windows Personal computer (used for Arduino code burning and signal processing),

  (3) USB cable (connecting Modified toothbrush hardware and personal computer).

- **Metrics:** The recognition accuracy of toothbrushing surfaces.
- **Output:** The brushing surfaces.
- **Experiments:** Our working example is LiT's brushing surfaces recognition accuracy of a user's brushing signal.
- **How much disk space required (approximately)?:** 100 MBytes for personal computer to run Matlab code.
- **How much time is needed to prepare workflow (approximately)?:** Within 30 minutes.
- **How much time is needed to complete experiments (approximately)?:** Within 20 minutes.
- **Publicly available?:** Yes.
- **Code licenses (if publicly available)?:** GNU General Public License.

A.3 Description

A.3.1 How to access

LiT artifact is open-sourced on GitHub, please download it through `https://github.com/LusGroup/LiT_Mobicom2023`.

A.3.2 Hardware dependencies

(1) Modified toothbrush hardware (can be produced by following the steps in Section A.4), (2) Windows Personal computer (used for Arduino code burning), and (3) USB cable (connecting Modified toothbrush hardware and personal computer).

A.3.3 Software dependencies

(1) Arduino IDE 1.8.19 for burning code to Arduino UNO, and (2) Matlab R2021a for signal processing.

A.4 Installation

(1) Please download the Gerber file (Gerber_PCB_single_pd.zip from `https://github.com/LusGroup/LiT_Mobicom2023/tree/main/hardware`) and send it to a professional flexible PCB manufacturer to print the flexible PCB (one toothbrush needs two printed flexible PCBs). Figure 1 is the flexible PCB preview opened with Gerbv.

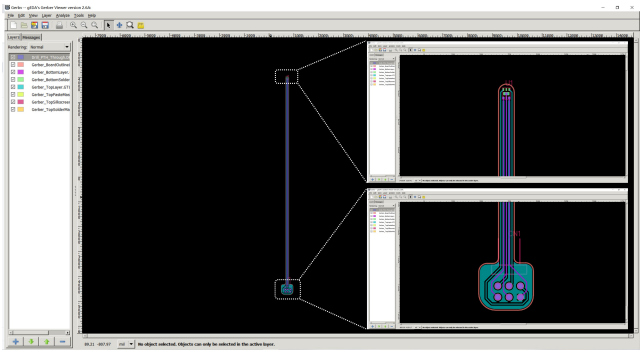

Figure 1. Flexible PCB preview.

(2) As shown in Figure 2(middle), connect the photosensors (flexible PCBs) and Arduino UNO with the Dupont wire and breadboard according to the circuit diagram (Figure 2(left)). Note that, the ADDR pins of the two photosensors were connected to the GND and VCC pins of the Arduino, respectively, to assign different addresses.

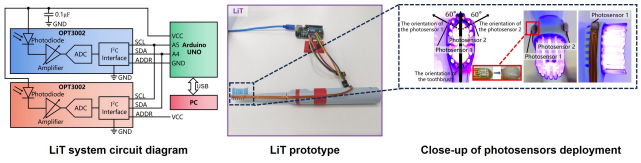

Figure 2. (left) Hardware circuit diagram. (middle) LiT prototype. (right) Close-up of photosensors deployment.

(3) As shown in Figure 2(right), glue two photosensors to the side of the toothbrush hand using silicone rubber near the top, and at approximately 60° from the toothbrush orientation, and then cover the sensors' surface with a layer of about 0.6mm transparent food-grade silicone rubber, which prevented saliva from short-circuiting the sensor and ensured excellent light transmittance. Note that, make sure the silicone rubber is fully cured.

(4) As shown in Figure 3, connect the Arduino UNO to a personal computer using a USB cable, and use Arduino IDE to burn the Arduino program (arduino_code.ino from `https://github.com/LusGroup/LiT_Mobicom2023/tree/main/hardware`) for Arduino UNO.

(5) As shown in Figure 4, connect the Arduino UNO to a personal computer using a USB cable, and use Matlab code (Step1_collect_data.m from `https://github.com/LusGroup/LiT_Mobicom2023/blob/main`) to collect a toothbrushing signal with Matlab.

A.5 Experiment workflow

(1) As shown in Figure 5, connect the Arduino UNO to a personal computer using a USB cable, and run the Matlab code (Step1_collect_data.m) to collect toothbrushing signal.

(2) Brush teeth using the correct Bass technique (`https://www.youtube.com/watch?v=olsUdRrYY70`) with deployed LiT toothbrush hardware. Meanwhile, use a smartphone to record video as Ground truth.

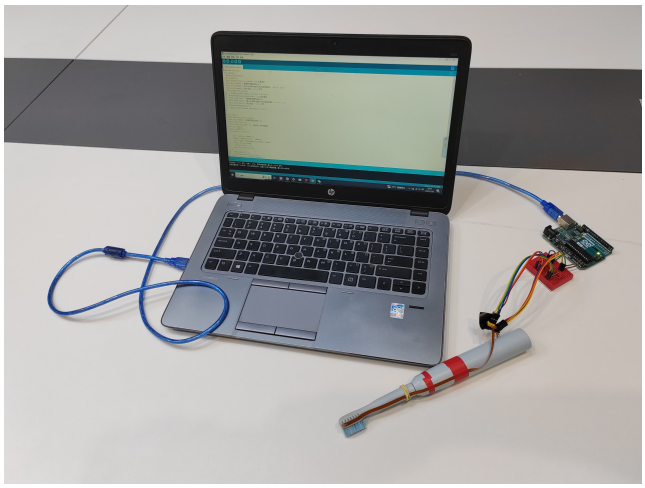

Figure 3. Burning arduino_code.ino for Arduino UNO with Arduino IDE.

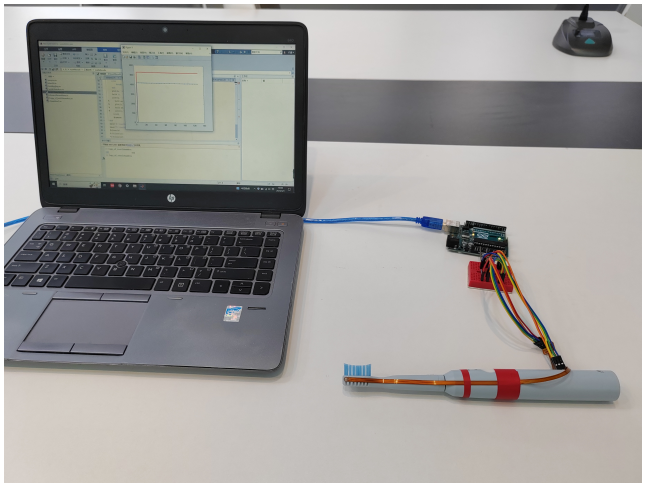

Figure 4. Collecting brushing signal via USB with Matlab.

(3) Run the Step2_signal_processing_and_analysis.m from `https://github.com/LusGroup/LiT_Mobicom2023/blob/main` to execute the LiT algorithm's MATLAB code, which processes the collected toothbrushing signal with labels.

(4) Obtain the 10-fold cross-validation accuracy of toothbrushing surface recognition from the labeled data.

A.6 Evaluation and expected results

The metric is the recognition accuracy of toothbrushing surfaces. By simply running our program, our code can automatically calculate the ten-fold cross-validation results of the collected data. Generally speaking, LiT can achieve a recognition accuracy of around 95% under different lighting conditions and user movements. Since reproduced/replicated evaluation requires the use of the LiT hardware to collect a substantial amount of toothbrushing signals, we provide a demo video of operating the artifact and an example dataset of a user to demonstrate the successful operation of the artifact and the usability of LiT.

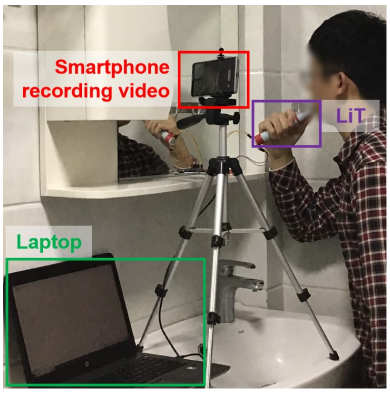

Figure 5. Experimental setup.

A.7 Notes

LiT demonstrates cross-user generality in subjects with good oral structures, meaning that users with good oral structures can directly use the models trained with brushing signals from other users. However, it is worth mentioning that, if a minority of users have conditions that affect the light reflection model in the oral cavity, such as wearing braces, missing teeth, or having impacted teeth, then these users can train the model once through a two-minute brushing session to achieve optimal toothbrushing monitoring results.

A.8 Methodology

Submission, reviewing and badging methodology:

- `https://www.acm.org/publications/policies/artifact-review-ba`
- `http://cTuning.org/ae/submission-20201122.html`
- `http://cTuning.org/ae/reviewing-20201122.html`