Human Factors in Design, Engineering, and Computing, Vol. 238, 2026, 298–308
https://doi.org/10.54941/ahfe1008263

A Lifelog System for Automatically Linking Physical Operations and Measurement Data in Hardware Debugging

Kairi Miyazaki[1,2], Masato Yazawa[2], Yutaro Takeuchi[2], and Norihisa Segawa[1]

[1]Graduate School of Frontier Informatics, Kyoto Sangyo University, Kyoto 603-8555, Japan
[2]Mathematical Assist Design Laboratory, Ota, Gunma 373-0019, Japan

ABSTRACT

In hardware debugging, waveforms observed on measuring instruments must be interpreted together with the physical context of the measurement, such as which pin was probed or which circuit change had just been made. Manual note-taking preserves this context but interrupts the debugging process. We developed a lifelog system that detects the worker's contact with an oscilloscope using an externally attached piezoelectric sensor and, upon each detected event, automatically captures the waveform via SCPI and extracts a 10-second first-person-view (FPV) video clip centered on the event (5 s before and 5 s after). In a within-subjects pilot study with five undergraduate students, four of five participants completed the debugging workflow, including documentation, more quickly with the proposed system (mean 389.0 s vs. 564.8 s). Mean ratings on three subjective workload items adapted from NASA-TLX were lower with the proposed system, although no statistically reliable differences were observed. In contrast, all five participants took longer to identify the measurement context of a waveform during review with the video logs than with handwritten notes. These observations suggest that automated capture may reduce documentation burden during debugging while shifting part of that burden to later retrieval. We discuss the implication that capture automation should be accompanied by retrieval-oriented interface design, such as an event-centered interface that integrates waveforms and FPV clips.

Keywords: Lifelogging, Hardware debugging, Documentation burden, Information retrieval, First-person video

INTRODUCTION

In electronics and embedded system development, observing waveforms with instruments such as oscilloscopes is an essential part of hardware debugging, and a debugging session may involve numerous waveform captures. To identify the cause of a fault, each waveform must be interpreted together with the physical context of the measurement: which component or pin was being probed, and which operation had just been performed.

Received September 26, 2026; Revised November 6, 2026; Accepted November 17, 2026; Available online December 1, 2026

© 2026 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

In small-scale prototyping and debugging workflows, this context is frequently recorded manually, for example by handwriting notes or by typing descriptive file names when saving screenshots. Manual recording imposes a documentation burden during debugging: the worker must take their hands off the probe and switch to a pen or keyboard, interrupting the trial-and-error process. Studies of interrupted work have shown that people compensate for interruptions by working faster, at the cost of more stress, frustration, and time pressure (Mark et al., 2008). In addition, when waveforms are reviewed days or weeks later, the memory of where each waveform was measured may have faded. Continuously recording the work on video avoids manual effort, but finding the few seconds corresponding to a particular waveform in long footage is inefficient.

We therefore developed a lifelog system that treats the worker's physical contact with the oscilloscope as a recording event. A piezoelectric sensor attached near the control panel detects the vibration caused when the worker presses a button or turns a knob; upon each event, the system captures the waveform via SCPI commands (Balaji et al., 2021) and extracts a short clip from continuously recorded first-person-view (FPV) video. This approach assumes that the moment of operating the instrument is a moment at which the displayed waveform is of interest; this assumption is not directly validated in this paper. Using this system, the pilot study reported here examines whether event-based logging can reduce the documentation burden during hardware debugging, and what retrieval costs arise when the automatically captured records are later reviewed.

The contributions of this paper are as follows. (1) A method for non-invasive sensing of oscilloscope interaction using a piezoelectric sensor attached externally near the control panel. (2) An event-based logging system that automatically associates oscilloscope waveform captures with FPV video clips. (3) **A pilot study with five participants suggesting a trade-off between a reduced documentation burden during debugging and an increased retrieval burden during later review.** The third contribution is the main human factors finding of this work.

RELATED WORK

Hardware Debugging Support

Environments that automate data acquisition from measuring instruments have been developed, such as an integrated test environment that controls multiple instruments from a single platform using SCPI (Balaji et al., 2021) and comparisons of instrument automation pipelines implemented in different programming languages (Kumar and Goswami, 2023). In HCI research, Booth et al. (2016) observed end-user developers building a physical computing circuit and found that most fatal faults stemmed from incorrect circuit construction and were often misdiagnosed as program bugs. To support such debugging, Toastboard instruments an entire breadboard and visualizes its voltage state (Drew et al., 2016), Bifröst

instruments both embedded code and microcontroller pins to help find errors at the hardware–software boundary (McGrath et al., 2017), and Pinpoint enables programmatic probing and isolation of components on PCBs (Strasnick et al., 2019). These tools focus on integrating data inside circuits and programs, and do not record the physical context of the worker's actions.

Lifelogging and Egocentric Vision

SenseCam, a representative wearable-camera lifelogging device, records the wearer's day as still images and sensor logs and was designed as a retrospective memory aid (Hodges et al., 2006); in addition to manual and timed capture, it captures images automatically when sensor readings change significantly. Using SenseCam, Sellen et al. (2007) showed that lifelog images help people connect to past events, but that their effect differs between remembering and knowing about the past and changes over time. Egocentric vision research has produced datasets for 3D hand pose estimation in assembly tasks (AssemblyHands; Ohkawa et al., 2023) and interactive task assistance (HoloAssist; Wang et al., 2023), and ENIGMA integrates localization, object recognition, and human–object interaction detection with wearable devices in industrial settings (Ragusa et al., 2023). Touch & Activate adds touch sensing to existing objects by attaching a vibration speaker and a piezoelectric microphone (Ono et al., 2013), showing that retrofitted piezoelectric elements can detect contact with existing objects.

Comparison and Research Gap

Sensor-triggered automatic recording already exists in lifelogging (e.g., SenseCam), but it targets daily life in general and does not link records with instrument waveforms, whereas debugging tools link signals with circuits but not with the worker's physical context. We are not aware of prior work that connects these two aspects by automatically linking measured waveforms with FPV video on a per-event basis using sensed instrument interaction. Moreover, prior lifelogging studies have focused on the benefit of automatic capture for recall, and less on the effort required to retrieve specific information from the captured records, which is the focus of our evaluation.

PROPOSED SYSTEM

The proposed system combines continuous recording by a wearable camera worn by the worker with event detection by a piezoelectric sensor attached to the oscilloscope housing. The moment at which the worker operates the oscilloscope is defined as an "important event to be recorded," and the video before and after that moment is extracted from the continuously recorded data, thereby automatically generating a debugging log based on the worker's viewpoint.

Hardware Configuration

Figure 1 shows the overall configuration of the system.

- **Piezoelectric sensor:** Attached to the oscilloscope housing or near the control panel, it detects the physical vibration caused by operations.
- **Wearable camera:** Mounted on the worker's head, it continuously records the worker's field of view (such as the probe position and the circuit board at hand) from a first-person perspective.
- **Raspberry Pi:** It processes the sensor signal and manages the recorded video data. Python is used to coordinate the sensor input with video processing.

Event-Triggered Video Clipping

The system extracts video clips without stopping the recording, by specifying time intervals relative to the trigger time. Let $T_{trigger}$ be the time at which the piezoelectric sensor detects an operation. Using a preset pre-trigger duration t_{pre} and post-trigger duration t_{post}, the system extracts the video in the interval $[T_{trigger} - t_{pre}, T_{trigger} + t_{post}]$.

System Overview and Workflow

From the worker's perspective, the system requires no additional action during debugging. The worker wears eyeglasses fitted with a small USB camera and operates the oscilloscope as usual, and each panel operation automatically produces a record that pairs the displayed waveform with a short FPV clip of what the worker was seeing and doing (Figure 1). Figure 3 shows the devices used: the piezoelectric sensor attached to the upper right of the oscilloscope housing near the control panel, the Raspberry Pi placed beside the oscilloscope, and the USB camera mounted on the eyeglass frame.

Figure 2 shows the components and the data flow between them. The worker operates an oscilloscope (SIGLENT SDS1104X-HD) and probes the target circuit while wearing a low-cost general-purpose USB camera mounted on eyeglasses, which records FPV video at 1080p and 30 fps and is connected to the control computer (Raspberry Pi 5) via USB. A vibration sensor (Grove Piezo Sensor) is attached externally to the oscilloscope housing near the control panel, and the control computer communicates with the oscilloscope via SCPI over a LAN (TCP/IP socket).

The workflow proceeds as follows: (1) the worker operates a button or knob; (2) the resulting vibration is detected by the sensor and raises a GPIO interrupt, defining the trigger time $T_{trigger}$; (3) the control computer immediately sends an SCPI command to capture the waveform image; (4) a 10-second FPV clip from $T_{trigger} - 5$ s to $T_{trigger} + 5$ s is extracted from the video buffer; and (5) the waveform image, timestamp, and clip are stored together as one event in an integrated log. The software is written in Python and handles video buffering and GPIO interrupts in parallel threads. The sensor sensitivity was empirically adjusted with the onboard trimmer potentiometer during preliminary setup so that the sensor responded when the oscilloscope was touched, resulting in a resistance of 590 Ω. A dead time of 1.0 s is applied after each detection to suppress repeated triggers.

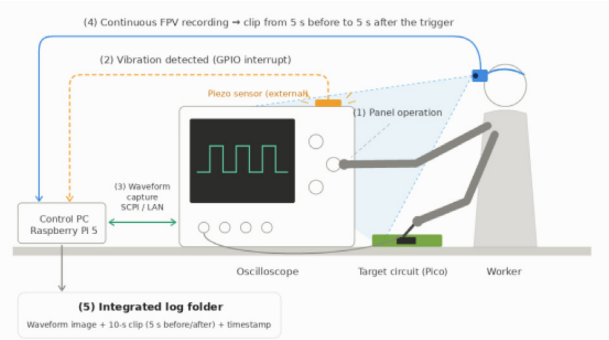


Figure 1 (Left): Overview of the system in use. The piezoelectric sensor (orange) is attached externally to the oscilloscope housing; the numbers indicate the processing steps (1)–(5) described in the text.

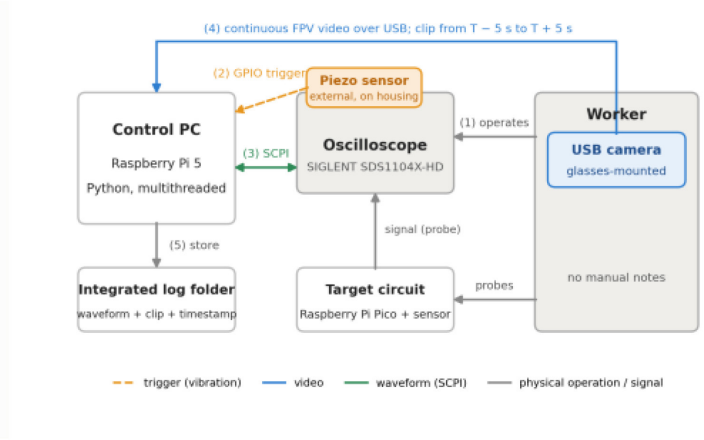


Figure 2 (Right): Components and data flow of the system. Dashed orange: trigger; blue: video; green: SCPI communication; gray: physical operation or signal.

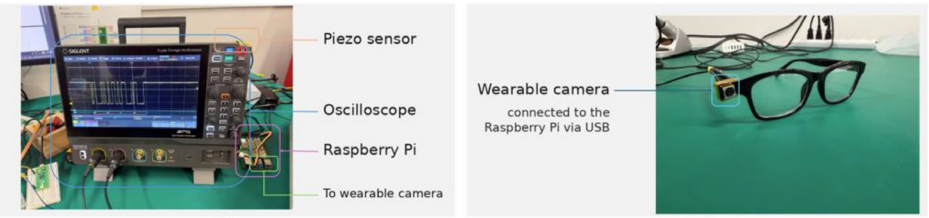


Figure 3: Devices used in the system: (a) oscilloscope with the externally attached piezoelectric sensor and the Raspberry Pi; (b) glasses-mounted USB camera.

Status of Trigger Detection Performance

The true positive rate, false negative rate, false positive rate, and latency of the piezoelectric trigger have not yet been measured systematically. During operation, including the experiment, the experimenters observed no obvious false triggers caused by environmental vibration, but this qualitative observation does not quantitatively validate the sensor settings.

PILOT STUDY

Design and Participants

This pilot study aimed to observe how automatic logging affects the worker's documentation burden during debugging and retrieval burden during review, and to identify usability issues; it was not intended to confirm statistical effects. A within-subjects design was used. Five undergraduate students in an information science department (ID002–ID006) participated. None had practical experience in hardware debugging, and their experience with oscilloscopes was limited to learning basic operations in class. A preliminary trial (ID001) was conducted to check the procedure and was excluded from the analysis. The order of conditions was partially counterbalanced: three participants (ID002, ID004, ID006) performed the conventional condition first and two (ID003, ID005) performed the proposed condition first.

Tasks and Conditions

Intentional physical faults were embedded in measurement circuits built with a Raspberry Pi Pico and sensors. Task X (SPI) contained a high-frequency short on the SCLK line, an open circuit on the MISO line, and an insufficient voltage level on the MOSI line. Task Y (I2C/UART) contained a missing pull-up resistor on the SDA line, an incorrect pull-up value on the SCL line, and a small leakage short on the TX line. To avoid learning of fault locations, each participant performed a different task in each condition, and the assignment of tasks to conditions was varied across participants; with five participants, the combinations of order and task assignment could not be fully balanced.

In condition A (conventional), participants took handwritten notes with a pen and notebook, and the experimenter saved the waveform images manually. In condition B (proposed), participants wore the camera, handwritten notes were prohibited, and waveform images and videos were recorded automatically. Condition A therefore included manual documentation, whereas condition B did not.

MEASURES

(1) Workflow completion time: the total time to identify all three faults in each condition, including any documentation performed during the task. (2) Retrieval time and correctness: after each task, participants were shown one waveform image captured in that condition and asked to identify which component or pin had been measured. Participants could refer to their notes in condition A and to both the 10-second clips and the full recording in condition B. This review took place immediately after the task in the same session. (3) Subjective workload: three items adapted from NASA-TLX (Hart and Staveland, 1988), namely mental demand, temporal demand, and frustration, were rated on a five-point scale (1: low, 5: high). These ratings are not comparable with standard NASA-TLX scores.

Ethical Considerations

This experiment was conducted by Mathematical Assist Design Laboratory in accordance with the company's research ethics guidelines. Before participation, participants were informed of the purpose of the study, the experimental procedure, and the handling of the collected data, including video data, and each participant gave signed consent. Participants were informed that participation was voluntary and that they could withdraw at any time. The collected video data were used only for research purposes, and no personally identifiable information is included in this paper.

RESULTS

Given the sample size, we focus on participant-level results and descriptive statistics. Inferential statistics are reported for completeness only; with $N = 5$, the exact two-sided Wilcoxon signed-rank test cannot yield $p < .0625$ even when all differences have the same sign.

Workflow Completion Time and Retrieval Time

Figure 4 shows the individual results, and Table 1 summarizes them. Four of the five participants (ID002–ID005) completed the debugging workflow more quickly in condition B, whereas ID006 took slightly longer (558 s to 572 s); the paired effect size was large (Cohen's $d_z = 1.03$), although the difference was not statistically reliable (paired $t(4) = 2.31, p = .082$; Wilcoxon $p = .125$). Because condition B involved no manual documentation, this result indicates a shorter workflow including documentation, not improved debugging skill itself.

In contrast, all five participants took longer to identify the measurement context in condition B (median 60 s vs. 3 s; Wilcoxon $p = .0625$; paired $t(4) = -4.14$, $p = .014$). Four of the five participants answered correctly in each condition.

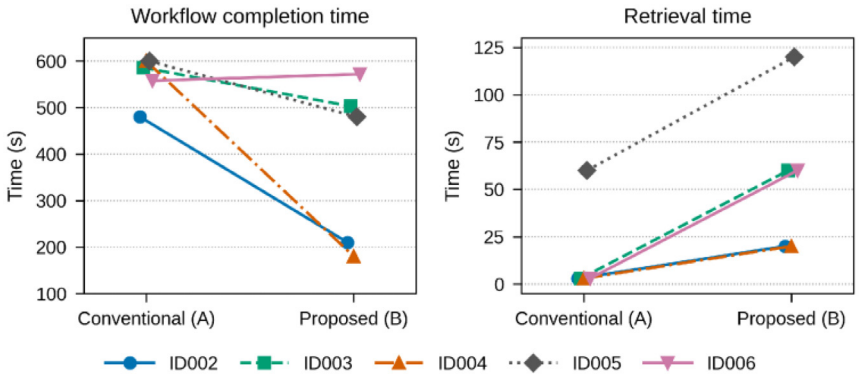


Figure 4: Participant-level workflow completion time (left) and retrieval time (right). In the right panel, lines of participants with identical values are slightly offset horizontally for visibility.

Table 1: Workflow completion time and retrieval results (N = 5).

Measure	Condition A (Conventional)	Condition B (Proposed)	Participants With Shorter Time in B
Workflow completion time (s), median; mean ± SD	586; 564.8 ± 50.4	480; 389.0 ± 180.6	4/5
Retrieval time (s), median; mean ± SD	3; 14.4 ± 25.5	60; 56.0 ± 41.0	0/5
Correct identifications	4/5	4/5	–

Subjective Workload and Comments

Ratings were mapped to conditions according to each participant's order of experience. As shown in Table 2, the median ratings of temporal demand and frustration were lower in condition B, and frustration was lower for four of five participants, whereas the pattern for mental demand was mixed. None of the differences was statistically reliable (all $p > .10$).

Table 2: Subjective workload items adapted from NASA-TLX (5-point scale, N = 5).

Item	A: Median (Mean ± SD)	B: Median (Mean ± SD)	Lower in B	Wilcoxon
Mental demand	3 (3.00 ± 1.00)	3 (2.60 ± 1.52)	3/5	p = .813
Temporal demand	4 (3.40 ± 1.82)	2 (2.60 ± 1.82)	3/5	p = .750
Frustration	3 (2.80 ± 1.10)	2 (1.80 ± 0.45)	4/5	p = .250

When asked which method made it easier to recall the measurement context, three participants chose the proposed system and two chose the conventional condition. Positive comments noted that recording took no time and that continuous recording felt easy. Negative comments noted that it took time to determine which waveform belonged to which task because all tasks were in the same video, that one participant did not understand how to use the system and searched the full video, and that the correspondence between waveforms and system records should be made clearer.

DISCUSSION

Documentation Burden During Debugging

In condition B, participants did not need to shift attention to notes or saving operations during debugging. The shorter workflow completion time for four of five participants, the lower frustration ratings for four of five participants, and comments such as recording taking no time are consistent with a reduced documentation burden, in line with findings that interruptions increase time pressure and frustration (Mark et al., 2008). Given the sample size, this interpretation should be regarded as a hypothesis for further testing.

Retrieval Burden

During review, however, all participants needed more time to locate the relevant information in condition B. Handwritten notes are already selected and summarized, whereas video logs are raw temporal records

that must be searched. This result reflects immediate retrieval cost rather than long-term recall because review occurred directly after each task. Interestingly, three participants nevertheless reported that the proposed system made context recall easier, suggesting a difference between perceived usefulness and objective retrieval efficiency. The longer retrieval time may also reflect the current interface: although event clips were available, their correspondence with waveforms was unclear, and at least one participant searched the full recording instead.

Burden Shift

Taken together, these observations suggest that automation does not eliminate the documentation burden but may shift part of it from the debugging phase to the retrieval phase. In the manual condition, the worker pays the cost of summarizing information during debugging, which makes later retrieval fast. In the automated condition, the worker is relieved of this cost during debugging, but the unsummarized records must be searched later. We regard this trade-off as the main human factors implication of this pilot study.

Design Implication

The observed retrieval cost suggests that capture automation must be accompanied by retrieval-oriented interface design. The value of automatic logging depends not only on what is captured but also on how easily a specific event can be found afterwards. Accordingly, Figure 5 shows a future interface concept, not yet implemented or evaluated, that presents waveform, timestamp, FPV clip, and task or event information in an event-centered interface; selecting a waveform marker on a shared timeline would seek directly to its clip. Improving how seek positions and bookmarks are presented on a video timeline has been shown to make video browsing more efficient (Carlier et al., 2015), and such techniques could be applied here.

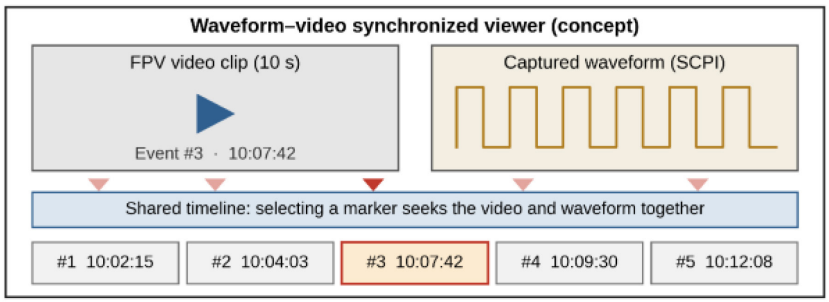


Figure 5: Future event-centered interface concept (not implemented or evaluated) for browsing waveforms and FPV clips.

Limitations

This study has the following limitations. (1) The sample size was small ($N = 5$), and (2) the study was a pilot, so the results indicate preliminary trends only. (3) Retrieval was tested immediately after the task rather than

after days or weeks; since the effect of lifelogs on recall can change over time (Sellen et al., 2007), delayed tests are needed. (4) All participants were novice students, limiting generalization to experienced engineers and real-world debugging. (5) The trigger detection performance of the piezoelectric sensor has not been validated systematically. (6) The assumption that moments of instrument operation are worth recording has not been validated. (7) The difficulty of Task X and Task Y was not verified to be equivalent. (8) Wearing the glasses-mounted camera may itself have influenced behavior or workload. (9) The retrieval results may depend on the current browsing interface, not only on video as a medium.

CONCLUSION

This paper presented a lifelog system that automatically links oscilloscope waveforms with FPV video clips, using externally sensed contact with the oscilloscope as a trigger. Automating documentation may reduce interruptions and the documentation burden during hardware debugging. However, the pilot study also suggests that this benefit can be accompanied by an increased retrieval burden during later review. These observations motivate an event-centered interface that integrates waveform captures and FPV video to support both efficient recording and subsequent information retrieval. Future work includes implementing and evaluating such an interface, systematically validating trigger detection, and conducting a larger study with delayed retrieval tests and experienced engineers.

ACKNOWLEDGMENT

The authors thank the Gunma University students who participated in the experiment. This work was supported by JSPS KAKENHI Grant Number 25K15200.

REFERENCES

Balaji, B.A., Sasikumar, S. and Ramesh, K. (2021) 'SCPI based integrated test and measurement environment using LabVIEW', *IOP Conference Series: Materials Science and Engineering*, 1045(1), 012036.

Booth, T., Stumpf, S., Bird, J. and Jones, S. (2016) 'Crossed wires: Investigating the problems of end-user developers in a physical computing task', in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. New York: ACM, pp. 3485–3497.

Carlier, A., Charvillat, V. and Ooi, W.T. (2015) 'A video timeline with bookmarks and prefetch state for faster video browsing', in *Proceedings of the 23rd ACM International Conference on Multimedia*. New York: ACM, pp. 967–970.

Drew, D., Newcomb, J.L., McGrath, W., Maksimovic, F., Mellis, D. and Hartmann, B. (2016) 'The Toastboard: Ubiquitous instrumentation and automated checking of breadboarded circuits', in *Proceedings of the 29th Annual ACM Symposium on User Interface Software and Technology*. New York: ACM, pp. 677–686.

Hart, S.G. and Staveland, L.E. (1988) 'Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research', in Hancock, P.A. and Meshkati, N. (eds.) *Human Mental Workload*. Amsterdam: North-Holland, pp. 139–183.

Hodges, S., Williams, L., Berry, E., Izadi, S., Srinivasan, J., Butler, A., Smyth, G., Kapur, N. and Wood, K. (2006) 'SenseCam: A retrospective memory aid', in Dourish, P. and Friday, A. (eds.) *UbiComp 2006: Ubiquitous Computing*. Lecture Notes in Computer Science, vol. 4206. Berlin: Springer, pp. 177–193.

Kumar, A. and Goswami, M. (2023) 'Performance comparison of instrument automation pipelines using different programming languages', *Scientific Reports*, 13, 18579.

Mark, G., Gudith, D. and Klocke, U. (2008) 'The cost of interrupted work: More speed and stress', in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York: ACM, pp. 107–110.

McGrath, W., Drew, D., Warner, J., Kazemitabaar, M., Karchemsky, M., Mellis, D. and Hartmann, B. (2017) 'Bifröst: Visualizing and checking behavior of embedded systems across hardware and software', in *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. New York: ACM, pp. 299–310.

Ohkawa, T., He, K., Sener, F., Hodan, T., Tran, L. and Keskin, C. (2023) 'AssemblyHands: Towards egocentric activity understanding via 3D hand pose estimation', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12999–13008.

Ono, M., Shizuki, B. and Tanaka, J. (2013) 'Touch & Activate: Adding interactivity to existing objects using active acoustic sensing', in *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. New York: ACM, pp. 31–40.

Ragusa, F., Furnari, A., Lopes, A., Moltisanti, M., Ragusa, E., Samarotto, M., Santo, L., Picone, N., Scarso, L. and Farinella, G.M. (2023) 'ENIGMA: Egocentric navigator for industrial guidance, monitoring and anticipation', in *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, pp. 695–702.

Sellen, A.J., Fogg, A., Aitken, M., Hodges, S., Rother, C. and Wood, K. (2007) 'Do life-logging technologies support memory for the past? An experimental study using SenseCam', in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York: ACM, pp. 81–90.

Strasnick, E., Follmer, S. and Agrawala, M. (2019) 'Pinpoint: A PCB debugging pipeline using interruptible routing and instrumentation', in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. New York: ACM, pp. 1–11.

Wang, X. et al. (2023) 'HoloAssist: An egocentric human interaction dataset for interactive AI assistants in the real world', in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20270–20281.