Sigmark Energy Overhead Raspberry Pi 5 – Technical Report

Nilma Abbas
Zealand – Sjællands Erhvervsakademi
Roskilde, Denmark
Email: niab@zealand.dk

Jesper Nordentoft
Zealand – Sjællands Erhvervsakademi
Roskilde, Denmark
Email: jeno@zealand.dk

Maja H. Kirkeby
Roskilde University
Roskilde, Denmark
Email: majaht@ruc.dk

***Abstract*—Software-based signaling markers are commonly used to align externally measured power traces with software execution on embedded systems. While typically assumed to incur negligible overhead, the runtime and energy impact of such markers is rarely quantified. This technical report presents an empirical study of the execution-time and energy overhead introduced by SigMark signaling on a Raspberry Pi 5.**

Using an externally powered measurement setup and a controlled signaling workload, we measured execution duration and total energy consumption while varying the number of transmitted SigMark messages. The results show that message transmission introduces a deterministic per-message execution-time overhead of 15.3 ms (95 % CI: [15.0, 15.6] ms), leading to a linear increase in execution duration with message count. Total energy consumption increases correspondingly, with an estimated system-level energy overhead of 0.091 J per message (95 % CI: [0.090, 0.093] J).

The observed overhead is stable across repeated measurements and reflects both the direct energy cost of message processing and the additional baseline energy consumed during extended execution time. These findings demonstrate that SigMark signaling overhead is non-negligible on resource-constrained platforms and should be explicitly accounted for when designing external energy measurement experiments.

***Index Terms*—External Power Measurement, Alignment Signals, Raspberry Pi 5**

I. INTRODUCTION

Energy consumption of Software depends on the device it is executed on, i.e., the device under test. When the energy consumption of the device is measured externally and the measurement data is collected on a separate data collection device. In such setups it is important to align start and stop of the software under test (running on the device under test) with the external measurement data. This is often indicated via signals from the device under test to the data collection device. One such collection and alignment system is Sigless [1] which uses SigMark to send signals from the device under test to the data collecting server.

A common assumption in setup designs is that these lightweight signals incur negligible energy overhead. However, this assumption is rarely validated empirically, and particularly on platforms such as the Raspberry Pi 5, the energy impact of sending a signal may be high compared to idle power dissipation and cause an unintended effect on the measured energy consumption.

The objective of this report is to quantify the energy consumption caused by sending a SigMark signal (using version 0.4.3) [2] from a Raspberry Pi 5 under controlled conditions.

II. EXPERIMENTAL DESIGN

The experimental setup was designed to ensure repeatability, minimize external interference, and isolate the effects of SigMark signaling overhead. All setup, data, and analyses scripts is available online at https://github.com/Nilma/sigmark-overhead-study.

A. Setup

The experimental setup comprises a Raspberry Pi 5 (device under test), a Siglent SPD3303X-E programmable power supply, a data collection computer (Thinkpad X260, IBM), and a dedicated Gigabit Ethernet switch. The Raspberry Pi 5 is powered exclusively by the Siglent supply, which delivers a regulated 5 V output, and all components are connected to the same local network to ensure stable and low-latency coordination. All experiments were conducted without active cooling (no fan, open case) under stable indoor ambient conditions, with no direct sunlight or external heat sources, to reflect typical real-world operation and avoid introducing cooling-induced variability.

B. Software Environment

The platform ran the Raspberry Pi OS (bookworm, Debian 12), a Linux-based operating system with a minimal user-space configuration. Background services were minimized to reduce interference during measurements.

C. Experimental procedure

The experiment was implemented as an automated signaling workload executed on the device under test. The workload repeatedly transmitted SigMark signals at controlled and predefined rates, while total energy consumption and execution duration were measured externally.

The experiment uses a shell script that communicates with a remote Sigless monitoring endpoint via the SigMark tool. Each SigMark transmission sends a lightweight marker message to the remote system without producing local log output on the device under test, ensuring that the signaling workload itself constitutes the primary source of computational activity.

Each experimental configuration consisted of a baseline duration of 80 s and a set of SigMark "tick" messages transmitted in regular timee intervals. Using a scheduling horizon of approximately 80 s ensures that energy measurements integrate system behavior over a sufficiently long period, thereby reducing the influence of short-term stochastic fluctuations in power consumption. By transmitting SigMark messages at fixed inter-message intervals, signaling activity is evenly distributed over the execution period, reducing sensitivity to short-term temporal effects such as scheduling jitter or transient system activity.

Specifically, the following message configurations were evaluated within each run:

- 0 messages (baseline, no signaling),
- 4 messages (one message every 20 s),
- 8 messages (one message every 10 s),
- 16 messages (one message every 5 s),
- 20 messages (one message every 4 s),
- 40 messages (one message every 2 s),
- 80 messages (one message every 1 s).

Each configuration was delimited by explicit `start` and `stop` SigMark markers to enable precise alignment between the signaling workload and the energy measurements. All configurations were executed sequentially within a single experimental cycle, and the full cycle was repeated 35 times to obtain statistically stable measurements. Algorithm 1 outlines the experimental procedure in pseudocode. The three different SigMark messages: start, stop, and tick are of similar structures and lengths: (1) the message's keywords `start`, `stop`, and `tick` are interchanged, (2) while the start and stop messages include one or two characters indicating the message intervals, the tick message does not.

By structuring the workload in this way, the experiment ensures that the only systematic difference between configurations is the number of transmitted SigMark messages, enabling direct comparison of energy consumption, execution duration, and per-message energy overhead across signaling rates.

III. THEORETICAL MODEL

Given that SigMark messages are transmitted sequentially and followed by fixed sleep intervals, an increase in execution time proportional to the number of transmitted messages is expected as part of the experimental design. This will also affect the energy consumption.

A. Execution time model

The experimental workload consists of a fixed-duration measurement window in which SigMark messages are transmitted sequentially at predefined intervals. Each message transmission incurs a small processing overhead due to signaling, system calls, and communication with the monitoring endpoint. Because message transmission and waiting periods are executed sequentially, this overhead accumulates with the number of transmitted messages.

Let T_{total} denote the total execution time of a measurement window and N_{messages} the number of transmitted messages.

Algorithm 1 SigMark signaling experiment

```
message_intervals ← $\{20, 10, 5, 4, 2, 1\}$    ▷ Seconds between messages
repetitions ← $35$
baseline_duration ← $80$    ▷ Nominal duration in seconds
for $r = 1$ to repetitions do
    SendStartMarker(1)
    Sleep(baseline_duration)
    SendStopMarker(1)
    for all $\Delta t \in$ message_intervals do
        SendStartMarker($\Delta t$)
        $N \leftarrow \lfloor$baseline_duration$/\Delta t\rfloor$
        for $i = 1$ to $N$ do
            SendSigMarkTick
            Sleep($\Delta t$)
        end for
        SendStopMarker($\Delta t$)
    end for
end for
```

The execution time can be modeled as the sum of a baseline duration and a per-message time overhead:

$$T_{\text{total}} = T_{\text{baseline}} + t_{\text{message}} \cdot N_{\text{messages}},$$

where T_{baseline} is the nominal execution duration in the absence of message transmission and t_{message} is the average time overhead incurred by sending a single SigMark message.

This formulation reflects the expected behavior of the experimental workload, in which message transmission is synchronous and no compensation mechanism is applied to maintain a fixed execution duration. As a result, execution time is expected to increase linearly with the number of transmitted messages.

B. Energy consumption model

The total energy consumption during a measurement window can be modeled as the sum of a baseline energy component and a message-dependent component:

$$E_{\text{total}} \approx E_{\text{baseline}} + E_{\text{message}} \cdot N_{\text{messages}}.$$

Here, $E_{\text{baseline}} = P_{\text{baseline}} \cdot T_{\text{baseline}}$ denotes the energy consumed during the baseline execution window in the absence of message transmission. The term E_{message} represents the average marginal energy cost of transmitting a single SigMark message.

Importantly, E_{message} captures both the direct energy cost of message processing and the additional baseline energy consumed due to the increase in execution time associated with message transmission. This formulation reflects the experimental design, in which message transmission and waiting periods are executed sequentially and execution duration is not held constant.

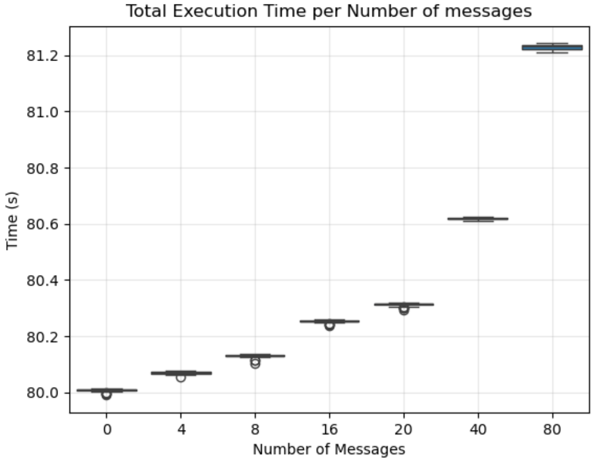

Fig. 1. Boxplots of total duration measured for different numbers of transmitted messages. Each box summarizes 35 repeated measurements and is used to illustrate variability and identify potential outliers within each configuration.

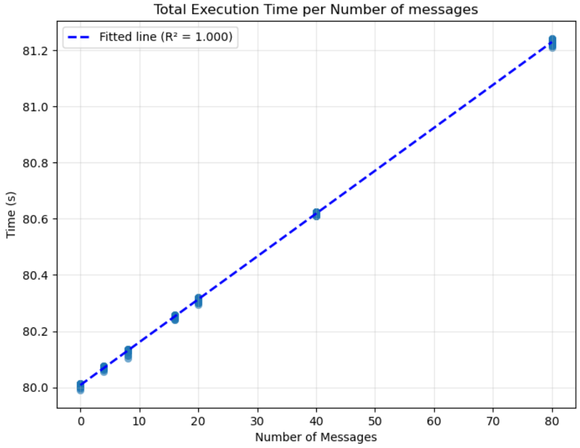

Fig. 2. Execution duration as a function of the number of transmitted messages with an ordinary least squares (OLS) regression fit. Each message adds approximately 15 ms of execution time.

IV. TIME RESULTS

A. *Time variability and outliers*

Figure 1 shows the distribution of total execution time T_{total} for each message configuration. Across all configurations, the distributions show limited spread, with narrow interquartile ranges and no extreme outliers, indicating low run-to-run variability in execution time.

B. *Total Execution Time Model*

As discussed in Section III-A, the experimental workload schedules SigMark message transmissions sequentially and does not enforce a fixed execution duration. Consequently, execution time is expected to increase proportionally with the number of transmitted messages due to the per-message signaling and waiting overhead.

To validate this behavior, a linear regression model, see Figure 2, was fitted with the measured execution duration as the dependent variable and the number of transmitted messages as the independent variable. The model, see Table I, explains virtually all observed variation in execution duration ($R^2 = 1.000$), indicating a deterministic and highly stable relationship.

The resulting execution time model is:

$$T_{\text{total}} = 80.0078 + 0.0153 \cdot N_{\text{messages}},$$

where T_{total} is the measured execution duration in seconds and N_{messages} is the number of transmitted SigMark messages.

The estimated intercept corresponds to a baseline execution duration of approximately 80.01 s in the absence of message transmission. The slope indicates that each additional message increases execution time by 15.3 ms on average (95 % CI: [0.015, 0.015]). This confirms that message transmission introduces a small but consistent per-message time overhead.

This systematic increase in execution duration implies that part of the observed increase in total energy consumption with higher message counts is attributable to baseline power consumption during the extended execution time, an effect that is explicitly accounted for in the subsequent energy consumption models.

TABLE I
EXECUTION TIME MODEL PARAMETERS ESTIMATED USING ORDINARY LEAST SQUARES (OLS).

Parameter	Estimate	95% CI
Baseline duration (T_{baseline})	80.0078 s	[80.007, 80.009]
Time per message (t_{message})	0.0153 s	[0.015, 0.015]
R^2	1.000	–

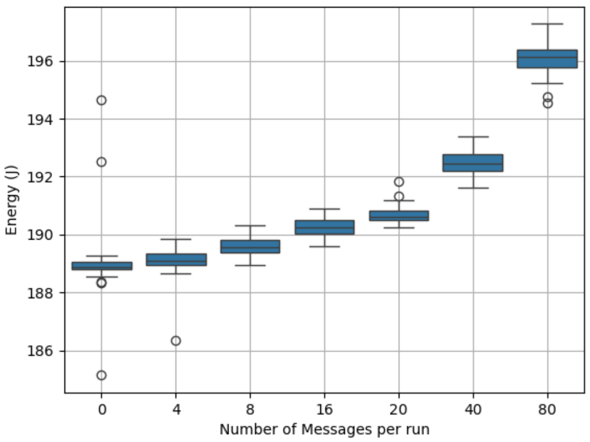

Fig. 3. Boxplots of total energy consumption measured over an 80 s interval for different numbers of transmitted messages. Each box summarizes 35 repeated measurements and is used to illustrate variability and identify potential outliers within each configuration.

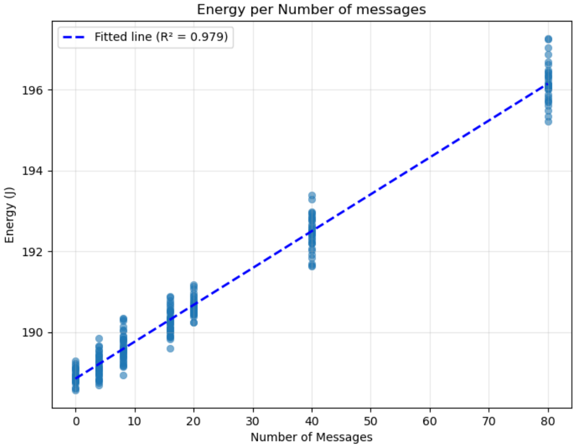

Fig. 4. Measured total energy consumption as a function of the number of transmitted messages with an ordinary least squares (OLS) regression fit. Each point represents a single experimental run measured over an 80 s interval. The slope of the regression corresponds to the average energy overhead per transmitted message.

V. ENERGY RESULTS

A. Energy consumption variability and outliers

Figure 3 shows the distribution of total energy consumption for different numbers of transmitted messages, summarized using boxplots based on 35 repeated measurements per configuration. Treating the message count as a categorical variable enables direct comparison of variability and dispersion across configurations. The distributions exhibit limited within-configuration spread, indicating low run-to-run variability, while a clear increase in energy consumption is observed with increasing message count relative to the idle baseline.

B. Total Energy Model

The relationship between total energy consumption and the number of transmitted messages was quantified using linear regression to estimate the incremental energy overhead per message. Figure 4 visualise the least-squares linear regression model we have fitted with total energy consumption as the dependent variable and the number of transmitted messages as the independent variable. The model, see Table II, explains 97.9 % of the variance in the observed energy consumption ($R^2 = 0.979$). The resulting linear model for total energy consumption as a function of the number of messages (N_{messages}) is:

$$E_{\text{Total}} = 188.84 + 0.091 \cdot N_{\text{messages}}.$$

TABLE II
TOTAL ENERGY CONSUMPTION MODEL PARAMETERS ESTIMATED USING ORDINARY LEAST SQUARES (OLS).

Parameter	Estimate	95% CI
Baseline energy (E_{baseline})	188.8422 J	[188.78, 188.90]
Energy per message (E_{message})	0.0913 J	[0.090, 0.093]
R^2	0.979	–

The estimated intercept corresponds to a baseline energy consumption of 188.84 J (95 % CI: [188.78, 188.90]), while each additional message increases energy consumption by 0.091 J on average (95 % CI: [0.090, 0.093]). The effect of the number of messages is statistically significant ($p < 0.001$).

VI. VALIDITY

a) Interval validity.: Each experimental configuration is delimited by explicit SigMark `start` and `stop` messages to align the signaling workload with the externally recorded energy measurements. These markers incur the same type of signaling overhead as regular `tick` messages; however, they are transmitted exactly once per configuration and are constant across all repetitions. As a result, their contribution constitutes a fixed overhead that is absorbed into the intercept of the regression models and does not affect the estimated per-message slopes. Consequently, the inclusion of start and stop markers does not bias the estimated execution-time or energy overhead per transmitted message.

b) External validity.: The reported results are specific to the tested Raspberry Pi 5 device, operating system configuration, and SigMark implementation. While the observed linear relationships between message count, execution duration, and energy consumption are expected to hold qualitatively on similar platforms, the quantitative overhead values may vary across individual devices, system configurations, and environmental conditions. Accordingly, the results should be interpreted as representative measurements rather than guaranteed bounds for all Raspberry Pi 5 systems.

VII. CONCLUSION

This report quantified the execution-time and energy overhead introduced by SigMark signaling on a Raspberry Pi 5 using an external power measurement setup. By varying the number of transmitted messages under controlled conditions, the study isolated the impact of signaling frequency on both execution duration and total energy consumption.

The results show that SigMark message transmission introduces a deterministic per-message execution-time overhead of approximately 15 ms, causing execution duration to increase linearly with the number of transmitted messages. Total energy consumption exhibits a corresponding linear increase, with an estimated system-level energy overhead of approximately 0.091 J per message. These effects were observed consistently across repeated measurements, indicating stable and reproducible behavior under the tested conditions.

The combined execution-time and energy models demonstrate that the measured energy overhead reflects both the direct cost of message processing and the additional baseline energy consumed during the extended execution time. Consequently, SigMark signaling overhead cannot be assumed negligible on resource-constrained platforms such as the Raspberry Pi 5.

Overall, these findings highlight the importance of explicitly accounting for signaling overhead when designing and interpreting external energy measurement experiments and provide a quantitative basis for selecting signaling rates that balance temporal alignment accuracy against measurement intrusiveness.

REFERENCES

[1] NoveltyGraph, “Sigless is a laboratory-grade monitoring tool,” https://siglessmonitor.com, 2025.

[2] ——, “Sigmark events — part of sigless monitor documentation (v0.4.3),” Online, 2025, accessed 29 Jan 2026; SigMark is a companion tool to Sigless that posts messages or checkpoints to the Sigless web server. [Online]. Available: https://siglessmonitor.com/