CERTIFICATION OF AUTHENTICITY OF ELECTRONIC EVIDENCE

Fed. R. Evid. 902(13); 28 U.S.C. § 1746

ProofSnap generated this document from the contents of the evidence package identified below. Read every statement before you sign. Rule 902(13) does not require this certification to be filed with the court; the last section below states what it does require. Do not place the signed copy back inside the package: the package's integrity rests on the SHA-256 hash values recorded in manifest.json, so any change to this file would make its hash value no longer match the recorded value, and verification would report the package as altered.

I. Declarant

1. My name is ProofSnap sample capture

2. My position or title is

3. My employer or organization is

4. My qualifications to make this certification, and the basis of my familiarity with the process described in Section III, are as follows:

(State facts rather than conclusions. For example: I performed the capture described in Section III, and I am familiar with how the capture system works.)

5. I have personal knowledge of the matters stated in this certification, and if called as a witness, I could and would testify competently to them.

II. The records this certification covers

6. This certification covers the electronic evidence package (referred to below as the package) identified as follows:

Evidence ID:	ps_0e6c1a29-3e47-4cc7-80ce-d670db22e027
Source URL:	https://www.adafruit.com/product/3405
Captured:	2026-11-04T15:00:23Z (UTC)
Capture software:	ProofSnap Extension version 1.19.7
ProofSnap account:	dOExb6dsycZhJdewGiFyjdDZRCm1

Files hashed before this certification was written: 44

The hash values of those files are set out in Section IV. The file verification/VERIFICATION_GUIDE.txt explains how the timestamps and their validation data are checked.

III. The process that generated the records (Rule 902(13))

7. The records were generated by an electronic process. The system performed the following steps:

(a) It obtained the current time from independent network time sources and compared those times against each other.

(b) It recorded the address of the page it captured, which is the address stated in paragraph 6, and the addresses to which the hostname resolved, as reported by two independent DNS resolvers queried separately.

(c) It saved a screenshot of what was on screen, the page HTML, and the text content of the page as rendered. Where it added a band of its own to the top of the screenshot, recording the address, the time and the identifier of this capture, it did so before the hash values in Section IV were computed, so those values are of the image as this package delivers it, and the page content below the band is unchanged.

(d) It recorded the changes it made to the page in order to capture it, such as hiding a cookie banner or a floating header, and it recorded how much of the page the screenshot covered.

(e) It computed a SHA-256 hash value and a SHA-512 hash value for every file that existed when manifest.json was written, and wrote those values into manifest.json together with the length of each of those files in bytes.

(f) It generated an RSA-4096 key pair for this capture and signed manifest.json with the private key. The signature is manifest.sig, and the public key is publickey.pem.

The system also wrote a step-by-step log of the capture (forensic_log.json) in which each entry carries the hash value of the entry before it, so that the removal or alteration of an entry can be detected. The log is sealed before the hash values in step (e) are computed, because manifest.json records a hash value for the log itself; the steps that follow the seal are evidenced by manifest.sig and by the timestamp files rather than by entries inside the log.

8. The records identified in paragraph 6 were generated by the process described in paragraph 7. Based on my familiarity with that process and on my observation of this capture, that process produces an accurate result, and it produced an accurate result here.

9. The screenshot reaches the end of the page. It was assembled from successive views of the page. The counts are recorded in chain_of_custody.json in the package.

10. The package includes capture_video.webm, a recording of the browser tab made while the capture ran, at up to 15 frames per second, with no audio track, stored in a WebM container with VP9 video. It is not a copy of any media file served by the source, and it is not offered as one.

11. The package also includes capture_video.mp4, a re-encoded copy of that recording provided so it opens in players that cannot read the original container. It is not byte-identical to the recording, and capture_video.webm is the original.

IV. Digital identification of the records

12. Each file included in the count "Files hashed before this certification was written" stated in paragraph 6 was identified by a SHA-256 hash value. A SHA-256 hash value is computed from the entire contents of a file. Changing a single byte of a file changes its hash value.

13. The hash values below were computed by the system at the time of capture. They are reproduced from manifest.json. Any person who has the package can recompute these values from the files and compare them with the values listed there.

metadata.json
SHA-256: f0c82bb0cf43f5b5a817432f2351fabec8ab6887e706bdb076634aaab1eb67ca

page.html
SHA-256: 3753fcbd2165ea9b0ff284b9c40cef1fc44c1a8c43d63156b8b9567e8aadaa72

screenshot.jpeg
SHA-256: a77997f6223901f7b1b1fcea4a6c3374dd3d5b04680d48b1bdd8e285b305777a

evidence.pdf
SHA-256: 9e3583153734105f1b7cac0a23c7551f63da1df031e21805eb9fb8b79acbad72

provenance_certificate.pdf
SHA-256: 96161e7aafde2dfe87fac10a9f47675e271f42b01c0f8b7a4635423ba4d85b6a

domtextcontent.txt
SHA-256: 4e67fa6db36911e5f6691dea5c0d63f5e889bfc45f01c7df067f161607dcc62c

c2pa.json
SHA-256: 28e212c8adf97ce3b8d13bab0d6c1f24ad7ac1ed4b985cd5ebc199b1f9c6c6f9

images/001_592-01.jpg
SHA-256: 0912f657a55ccf394c2ea1cfd9c375a17bb687a4d22b1b88f51fcb771ddbb2cb

images/002_2772-09.jpg
SHA-256: 2bc153195dec7845bf29d9b0aa2385020136ff0fda51b96d9e5bbaeaa295cb2b

images/003_2771-14.jpg
SHA-256: 76aed16ebe431385264d645a4d750a9f1175437f7eabaa0cac2704b699a37b96

images/004_3403-04.jpg
SHA-256: 0b41e2d1d550c61a6f7d63ac2a996b4603697708672c601f88ac9648352d1b5b

images/005_2796-12.jpg
SHA-256: 4e3f34ca5bb9a89c359f24eab5661477a654d4b9064b31b02586b9892a78148c

images/006_2995-08.jpg
SHA-256: 0132a39887db53c78862ba3f2596c2a4d0763c0d7c2b0215a0739b56545e014e

images/007_2680-01.jpg
SHA-256: 8950a6f2102029ba4961cbce868e0d14b2ab6987e0dfa6d3f85b356b1a633982

images/008_3061-04.jpg
SHA-256: 2e563d3319212f0a003b856134546bc037f5cb5b53ff39b31a9b8a41d0b20ed4

images/009_3010-12.jpg
SHA-256: 1f6e1688763cbd4524bc22272bc21aea7e8db3c33628bbf82e6e9b24a638a494

images/010_3213-09.jpg
SHA-256: 76e9d66973a39b2be10fcae30967aef0075079f121107bd7bf780b4f3c4c6518

images/011_3046-07.jpg
SHA-256: 56c621d3d9c61ebb17006c6bb833e0998120afeb6fed93fecda381822752c7f9

images/012_2821-07.jpg
SHA-256: 7ce670880e011a3774656db03a2cbcbf595c451ffeabd43296eea28167a673ef

images/013_blinka_esp.jpg
SHA-256: 45d5310be83c2ff409602fd157de510612bee8fe946d48ede61d6634d7ed2e33

images/014_adafruit_02.png
SHA-256: 48841f38aab8d51fb1f0a6e8914129f040d37ce6c1c63270f31f8b6db940af19

images/015_fam.jpg
SHA-256: 5aa3f2e40dcea8e63ad3d61dc16ad774ab68bcdec4251065ede74883a6837b62

images/016_thumb.jpg
SHA-256: d8144ca8448cbb3ae078bf9c7c5eded341acc445ef444a4f2e775fa72d415764

images/017_RoHS2_2015_863_EU.svg
SHA-256: f0aaf92130a8b27bf4367f1c80336c151169cc55fd49ab8a1510f69f5297351b

The remaining 20 files and their hash values are listed in manifest.json in the package. They are omitted here only to keep this document to a readable length.

14. The file manifest.json was signed with the RSA-4096 private key generated for this capture, using RSASSA-PKCS1-v1_5 with SHA-256. The signature is manifest.sig, and the corresponding public key is publickey.pem. The commands for verifying the signature and recomputing the SHA-256 values are set out in verification/VERIFICATION_GUIDE.txt in the package, and the scripts verification/verify.sh and verification/verify.ps1 run those commands.

V. Time

15. The SHA-256 hash value of manifest.json was submitted to the OpenTimestamps protocol, which commits that value to the Bitcoin blockchain. Where the submission returned a proof, that proof is the file manifest.json.ots in the package; where it did not, that file is absent and no OpenTimestamps proof is offered. When the commitment has been confirmed in a Bitcoin block, the proof shows that manifest.json, and therefore every hash value listed in it, existed no later than the time of that block.

16. The same hash value was also submitted to an electronic time-stamp service operating under Regulation (EU) No 910/2014 (eIDAS). Where that service returned a token, the RFC 3161 time-stamp token is the file manifest.json.tsr in the package and the data needed to validate that token offline, including the certificate chain and revocation information, is in eidas_validation.json; where it did not, those files are absent and no time stamp is offered.

VI. Declaration

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Executed on (date): _____

Executed at (city and state, or country): _____

Signature: _____

Printed name: _____

VII. Notice to the adverse party and limits of this certification

Rule 902(13) incorporates the notice requirement of Rule 902(11). Before the trial or hearing, the proponent must give an adverse party reasonable written notice of the intent to offer the record, and must make the record and this certification available for inspection, so that the party has a fair opportunity to challenge them. Serving that notice is the proponent's responsibility.

Self-authentication under Rule 902 concerns authenticity only. It relieves the proponent of the need to offer extrinsic evidence of authenticity. It does not make a record admissible, it does not affect an objection based on hearsay, relevance, or any other rule, and it does not establish who wrote, published, or controlled the captured content. A hash value shows that a file has not changed since the hash value was computed; it says nothing about whether the captured content is true. ProofSnap is software: it certifies nothing and testifies to nothing, and every statement in this document is the declarant's. ProofSnap does not provide legal advice, and this template is not legal advice.