



AHCOMPUTING // COVERT HARDWARE

## PROJECT 1984

### Covert Recorder Build Guide

---

*A concealed audio recorder built inside a hardcover copy of George Orwell's 1984. Adafruit Feather ESP32, INMP441 microphone, microSD storage, WiFi retrieval. Open the book, flip the switch, set it down.*

FOR PERSONAL PROTECTION USE ONLY · UTAH ONE-PARTY CONSENT APPLIES

---

BEFORE YOU BUILD

### Know the Law First

#### UTAH LAW – ONE-PARTY CONSENT

Utah is a one-party consent state (Utah Code § 77-23a-4). This means a person can legally record any conversation they are a participant in, without notifying the other parties. If you are present in the room during the conversation being recorded, the recording is legal. You do not need to inform the other parties.

#### IMPORTANT LIMITATIONS

One-party consent only applies when the person doing the recording is an active participant in the conversation. Recording a conversation you are not part of – leaving the device running in an empty room, for example – is a

different matter legally. Use it only in conversations you are personally present for.

This is a legitimate and legal use in Utah for anyone recording conversations they are a participant in. Laws vary by state and country – understand the law in your jurisdiction before building or using this device.

## CHAPTER I

### Materials

*Everything you need. New spend is roughly \$35–45 depending on what you have on hand.*

Adafruit Feather ESP32 V2	Assume owned
8MB flash, built-in LiPo charging via USB-C, WiFi. <a href="https://adafruit.com/product/5400">adafruit.com/product/5400</a> .	
INMP441 I2S Microphone	~\$10
24-bit omnidirectional MEMS mic, 3.3V, designed for ESP32. <a href="#">5-pack on Amazon</a> ~\$10. Get the pack – they're cheap.	
Adafruit MicroSD Breakout+	~\$8
SPI interface, onboard level shifting and regulator. <a href="https://adafruit.com/product/254">adafruit.com/product/254</a> .	
MicroSD Card, 8–32GB	~\$5
Format FAT32. At 16kHz mono WAV, 8GB holds ~270 hours. Any brand works.	
LiPo Battery, 1000mAh 3.7V	~\$10
Standard JST-PH 2-pin. <a href="https://adafruit.com/product/258">adafruit.com/product/258</a> . ~5.5 hours recording on a full charge.	
SPDT Toggle Switch (qty: 3)	~\$3
Small SPDT toggles for internal use – record mode, WiFi mode, and battery power cutoff. Search "mini SPDT toggle switch" on Amazon, ~\$1 each.	
Hardcover 1984 – Orwell	~\$2–5

Needs to be thick enough for the components. A chunky hardcover edition – at least 250 pages. Thrift store copies work perfectly. Avoids buying a nice copy you'll hollow out.

### Hot Glue Gun + Glue

Assume owned

For securing components inside the book. Nothing moves, nothing rattles.

### 3mm Drill Bit + Drill

Assume owned

For the microphone hole in the cover. One small hole is all you need.

### Sharp Knife + Ruler

Assume owned

Box cutter or X-Acto knife for hollowing the book pages.

## CHAPTER II

### Wiring

*All connections. Keep wires short – you're working inside a book. Solder everything, no breadboard connections in the final build.*

COMPONENT	PIN	FEATHER ESP32 PIN	NOTES
INMP441 Mic	VDD	3.3V	Power
INMP441 Mic	GND	GND	
INMP441 Mic	SCK	GPIO 26 – labeled A0 on board	I2S clock
INMP441 Mic	WS	GPIO 25 – labeled A1 on board	I2S word select
INMP441 Mic	SD	GPIO 33 – labeled 33 on board	I2S data
INMP441 Mic	L/R	GND	Tie to GND for left channel
MicroSD	CS	GPIO 4 – labeled A5 on board	SPI chip select – 3rd pin from bottom-right, marked A5
MicroSD	GND	GND	Ground

MicroSD	3V	3V	Power
MicroSD	SI (MOSI)	GPIO 19 – labeled MO on board	SPI data out – pin labeled SI on SD breakout
MicroSD	SO (MISO)	GPIO 21 – labeled MI on board	SPI data in – pin labeled SO on SD breakout
MicroSD	CLK (SCK)	GPIO 5 – labeled SCK on board	SPI clock – pin labeled CLK on SD breakout
Toggle 1 – Record	Signal	GPIO 14 – labeled 14 on board	Internal pullup, switch to GND
Toggle 2 – WiFi	Signal	GPIO 27 – labeled 27 on board	For retrieval mode
Toggle 3 – Power	Inline GND	Battery GND → Feather GND	Cut the battery ground cable, splice toggle in-line – disconnects battery entirely when OFF
LiPo Battery	JST	BAT connector	Built-in Feather charging – charge with Toggle 3 ON

#### NO LEDES

The original design includes status LEDs. Omit them entirely for this build. Any light bleed from inside a closed book is a problem. Switch position tells you the recording state – you don't need a light.

#### POWER TOGGLE MOD

Toggle 3 disconnects the battery at the hardware level, eliminating standby drain entirely. Cut the black (ground) wire on the JST cable coming from the battery and splice the toggle in-line – one end to the battery side, one end to the Feather side. When Toggle 3 is OFF, the battery is fully disconnected. The device draws nothing. Flip it ON when you need to charge or use the device.

## Preparing the Book

*The hollowing process. Takes about an hour, most of which is waiting for glue to dry. Do it right once.*

### 1 Choose the right copy

Find a hardcover edition at least 1 inch thick. Thrift stores usually have multiple copies. You want a copy you don't mind ruining. The older and more worn the better – a pristine copy looks out of place, a weathered one looks read.

### 2 Protect the first and last pages

Leave the first 10–15 pages and the last 10–15 pages intact. This makes the book look full when opened from either end. Only the middle is hollowed. Mark your cut lines lightly in pencil.

### 3 Glue the pages together

Brush a thin layer of PVA glue (white glue) around the outside edges of the pages you're going to cut – top, bottom, and outer edge, not the spine. Close the book and let dry completely under weight. This makes the pages solid so they cut cleanly and stay together. Takes an hour minimum.

### 4 Cut the cavity

With a sharp box cutter and ruler, cut a rectangular cavity into the glued pages. Leave at least half an inch on all sides. Cut in layers, removing a few pages at a time. Take your time – a clean cut looks much better than a ragged one. The cavity needs to fit the Feather, SD breakout, and battery stacked flat.

### 5 Drill the microphone hole

Decide on mic placement. The front cover bottom corner is ideal – when the book lies face-up on a desk, the mic faces up into the room. Drill a clean 3mm hole through the front cover only. Sand the edges smooth so there are no paper fibers sticking out. Keep it near an edge where it looks like it could be a manufacturing artifact.

### 6 Seal the cavity walls

Brush another thin layer of PVA glue on the inside walls of the cavity and let dry. This strengthens the edges, prevents fraying, and stops paper dust from getting on the components.

7

## Install and secure components

Hot glue the components into the cavity in this order: battery flat on the bottom, SD breakout beside it, Feather on top. Run the mic on a short wire to the hole in the cover – hot glue it flush against the inside of the hole, port facing out. Hot glue all three toggle switches somewhere accessible when the book is open – Toggle 1 (record), Toggle 2 (WiFi), and Toggle 3 (power). Route wires along the spine edge so they don't bunch up.

8

## Test before closing

Flash the firmware, insert the SD card, flip the switch. Let it record for a few minutes while you talk at it from a meter away. Then retrieve the file over WiFi and check audio quality. Fix anything before sealing. Much harder to fix after.

CHAPTER IV

## Firmware

*The complete sketch. Two modes – record and WiFi retrieval. Mutual exclusion ensures you can't corrupt a recording by switching modes mid-session.*

## Arduino IDE Setup

Use **Arduino IDE 2.x**. In the Boards Manager, install the **Adafruit ESP32** board package. Select **Adafruit Feather ESP32 V2** as your board.

DO NOT USE LIBRARY MANAGER FOR THESE

The Library Manager versions of ESPAsyncWebServer have compatibility issues with the current ESP32 Arduino core. Install from GitHub directly via the terminal instead.

TERMINAL – INSTALL LIBRARIES VIA GIT

```
cd ~/Arduino/libraries
rm -rf ESPAsyncWebServer ESP_Async_WebServer AsyncTCP ESPAsyncTCP
```

```
git clone https://github.com/ESP32Async/ESPAsyncWebServer
git clone https://github.com/ESP32Async/AsyncTCP
```

Restart Arduino IDE after cloning. SD, SPI, WiFi, and DNSServer come bundled with the ESP32 board package – no separate install needed.

Open a new sketch, paste the firmware below, select the correct port, and click the upload arrow (→). Do not use the Debug button.

```
PROJECT1984.INO – COMPLETE FIRMWARE (TESTED AND WORKING)
```

– complete firmware

```
// Project 1984 – Covert Recorder
// Toggle 1 (GPIO 14): Record to SD as WAV
// Toggle 2 (GPIO 27): WiFi AP + captive portal file server
// No LEDs – all status is silent

#include <driver/i2s.h>
#include <SD.h>
#include <SPI.h>
#include <WiFi.h>
#include <ESPAsyncWebServer.h>
#include <DNSServer.h>

// — Pins —
#define I2S_SCK      26
#define I2S_WS      25
#define I2S_SD_PIN  33
#define SD_CS       4   // Avoiding GPIO 13 (Built-in LED)
#define REC_SWITCH  14
#define WIFI_SWITCH 27

// — Audio config —
#define SAMPLE_RATE 16000
#define BUFFER_SIZE  512

// — WiFi config —
const char* AP_SSID = "FieldRecorder";
const IPAddress apIP(192, 168, 4, 1);

AsyncWebServer server(80);
DNSServer dnsServer;

// — State —
File audioFile;
bool recording = false;
bool wifiActive = false;
```

```

uint32_t dataSize = 0;
int fileIndex = 0;

// — WAV helpers —————
void writeWavHeader(File &f) {
    byte h[44] = {0}; f.write(h, 44);
}

void finalizeWav(File &f, uint32_t samples) {
    uint32_t db = samples * 2;
    uint32_t sr = SAMPLE_RATE;
    uint32_t br = SAMPLE_RATE * 2;
    f.seek(0);
    f.write((const uint8_t*)"RIFF", 4);
    uint32_t cs = 36 + db; f.write((uint8_t*)&cs, 4);
    f.write((const uint8_t*)"WAVE", 4); f.write((const uint8_t*)"fmt ", 4);
    uint32_t s1 = 16; f.write((uint8_t*)&s1, 4);
    uint16_t af=1, ch=1, ba=2, bp=16;
    f.write((uint8_t*)&af,2); f.write((uint8_t*)&ch,2);
    f.write((uint8_t*)&sr,4); f.write((uint8_t*)&br,4);
    f.write((uint8_t*)&ba,2); f.write((uint8_t*)&bp,2);
    f.write((const uint8_t*)"data",4); f.write((uint8_t*)&db,4);
}

// — Recording —————
void startRecording() {
    String fn;
    // Find an unused filename to prevent overwrite on reboot
    do {
        fn = "/rec_" + String(fileIndex++) + ".wav";
    } while (SD.exists(fn));

    audioFile = SD.open(fn, FILE_WRITE);
    if (!audioFile) return;
    writeWavHeader(audioFile);
    dataSize = 0; recording = true;
}

void stopRecording() {
    recording = false;
    finalizeWav(audioFile, dataSize);
    audioFile.close();
}

// — WiFi file server —————
String buildPage() {
    String h = "<!DOCTYPE html><html><head>"
        "<meta name='viewport' content='width=device-width,initial-scale=1'>"
        "<title>1984</title>"
        "<style>body{background:#0e0d0b;color:#c8bfa8;font-family:monospace;"
        "padding:2rem;}h1{color:#c43030;font-size:1.4rem;margin-bottom:1rem;}"
        "a{color:#c8850a;display:block;padding:.5rem 0;}"

```

```

"border-bottom:1px solid #2e2b24;text-decoration:none;}"
"a:hover{color:#e8dfc8;}.m{color:#5a5448;font-size:.8em}"
"</style></head><body>"
"<h1>RECORDINGS</h1>";
File root = SD.open("/");
File f = root.openNextFile();
bool found = false;
while (f) {
  String n = String(f.name());
  if (n.endsWith(".wav")) {
    h += "<a href='/dl/' + n + "'>" + n +
        " <span class='m'>(" + String(f.size())/1024.0,1) + "KB)</span></a>";
    found = true;
  }
  f = root.openNextFile();
}
if (!found) h += "<p class='m'>No recordings.</p>";
h += "</body></html>";
return h;
}

void startWifi() {
  WiFi.softAPConfig(apIP, apIP, IPAddress(255,255,255,0));
  WiFi.softAP(AP_SSID);
  dnsServer.start(53, "*", apIP);
  server.on("/", HTTP_GET, [](AsyncWebServerRequest* r){
    r->send(200, "text/html", buildPage());
  });
  server.onNotFound([](AsyncWebServerRequest* r){
    r->redirect("http://192.168.4.1/");
  });

  // Automatically map /dl/ endpoints to the root directory of the SD
  server.serveStatic("/dl/", SD, "/");

  server.begin();
  wifiActive = true;
}

void stopWifi() {
  server.end();
  dnsServer.stop();
  WiFi.softAPdisconnect(true);
  wifiActive = false;
}

// — Setup —————
void setup() {
  pinMode(REC_SWITCH, INPUT_PULLUP);
  pinMode(WIFI_SWITCH, INPUT_PULLUP);

  if (!SD.begin(SD_CS)) {

```

```

// Fallback: Silet signal that the SD card failed to mount
WiFi.softAPConfig(apIP, apIP, IPAddress(255,255,255,0));
WiFi.softAP("SD_ERROR_1984");
while(1);
}

i2s_config_t cfg = {
    .mode = (i2s_mode_t)(I2S_MODE_MASTER|I2S_MODE_RX),
    .sample_rate = SAMPLE_RATE,
    .bits_per_sample = I2S_BITS_PER_SAMPLE_16BIT,
    .channel_format = I2S_CHANNEL_FMT_ONLY_LEFT,
    .communication_format = I2S_COMM_FORMAT_STAND_I2S, // Updated ESP-IDF constant
    .intr_alloc_flags = ESP_INTR_FLAG_LEVEL1,
    .dma_buf_count = 8, .dma_buf_len = BUFFER_SIZE
};

i2s_pin_config_t pins = {
    .bck_io_num=I2S_SCK, .ws_io_num=I2S_WS,
    .data_out_num=I2S_PIN_NO_CHANGE, .data_in_num=I2S_SD_PIN
};

i2s_driver_install(I2S_NUM_0, &cfg, 0, NULL);
i2s_set_pin(I2S_NUM_0, &pins);
}

// — Loop —————
void loop() {
    bool recOn = (digitalRead(REC_SWITCH) == LOW);
    bool wifiOn = (digitalRead(WIFI_SWITCH) == LOW);

    // Record mode – wifi switch ignored while recording
    if (recOn && !recording && !wifiActive) startRecording();
    if (!recOn && recording) stopRecording();

    // WiFi mode – only when not recording
    if (wifiOn && !wifiActive && !recording) startWifi();
    if (!wifiOn && wifiActive) stopWifi();
    if (wifiActive) dnsServer.processNextRequest();

    // Write audio if recording
    if (recording) {
        int16_t buf[BUFFER_SIZE];
        size_t n = 0;
        i2s_read(I2S_NUM_0, buf, sizeof(buf), &n, portMAX_DELAY);
        audioFile.write((uint8_t*)buf, n);
        dataSize += n/2;
    }
}
}

```

## Operation

### RECORDING SESSION

- 1 The night before – flip **Toggle 3 (Power)** to ON, then charge via USB-C until the Feather's charge LED goes out. Flip Toggle 3 OFF when done charging.
- 2 Before the meeting – open the book, flip **Toggle 3 (Power)** to ON, then flip **Toggle 1 (Record)** to ON. Close the book.
- 3 Set the book on the desk or hold it naturally. Front cover up, mic hole facing into the room. Nobody is looking at a book.
- 4 After the meeting – open the book, flip **Toggle 1 (Record)** to OFF. Recording stops and WAV file is finalized.
- 5 Flip **Toggle 2 (WiFi)** to ON. Connect phone to **FieldRecorder WiFi**. Browser opens automatically. Download the recording. Flip Toggle 2 OFF.
- 6 Flip **Toggle 3 (Power)** to OFF. Battery is fully disconnected – no standby drain.

### BATTERY NOTE

At 16kHz mono, a 1000mAh battery gives approximately 5.5 hours of continuous recording. A typical meeting is 30-60 minutes. Toggle 3 disconnects the battery entirely when not in use – no standby drain means a full charge stays full until you need it.

### REMINDER

In Utah, a participant in a conversation may legally record it without notifying other parties. Use this device only in conversations you are personally present in. This is a protective tool, not a surveillance device.