How to Add ChatGPT to a TI-84 Calculator

The TI-84 is great for math, but what if it could also explain problems in plain English? By adding a small Wi-Fi chip (like an ESP32) and connecting it to a ChatGPT server, you can turn your calculator into an **AI-powered helper**.

This guide will show you, step by step, how to:

- Connect extra hardware to your TI-84.
- Link it to a server that runs ChatGPT.
- Type a question into your calculator and see an AI answer right on the screen.

No deep coding or electronics knowledge is needed - just a bit of patience and curiosity.

MADE FROM WHAT?

1. **TI-84 Calculator**
   – Basic TI-84 model Plus (Found on Ebay/Amazon)
2. **ESP32C3 Microcontroller**
   – Acts as the “bridge” between your calculator and the internet.
   From SEEED Studio: https://www.seeedstudio.com/
3. **USBC Cable (for ESP32 programming)**
   – To connect the ESP32 to your computer for programming.
4. **Computer (Windows/Mac/Linux)**
   – For flashing the ESP32 with the code that connects to ChatGPT & Creating the server
5. **API Key from ChatGPT**
   – Direct access to the ChatGPT API with your API key.

MATERIALS (RIGHT CLICK ON MATERIALS FOR LINK)

- ✓ Wires (to connect ESP32C3 to Calculator)

- ✓ A screwdriver to open the Ti84 (More Specifically T6 Torx screwdriver)

- ✓ Solder and Soldering wire (To put it all together)

- ✓ **X4 Resistors (the 1k Ohms)**

- ✓ **2x Mosfets**

- ✓ **X1 PCB (WHERE EVERYTHING SITS ON) – EXPLAINED LATER**

A PCB (Printed Circuit Board) is the foundation where all the electronic components are mounted and connected. Instead of using messy wires, a PCB neatly routes signals through copper tracks, making your project more reliable and compact. For our ChatGPT-enabled TI-84, the PCB is where the ESP32 chip and supporting components will sit, tucked neatly inside the calculator. This board acts as the “brain” that links your calculator to the internet and powers the AI functions.

To make things easy, the PCB is already designed for you, all you need to do is download the file from this link and upload it to JLCPCB . Once you paste the file in and confirm the order, JLCPCB will manufacture your custom PCB and ship it to you. This way, you don’t need to design anything yourself, the board is already built and ready.

START:

Now, while your PCB is being made, let's program your ESP32 and set up your server. This is the part that lets your calculator actually talk to ChatGPT. To do this, head over to the GitHub repository (link here) and download the files needed for the "**ESP32"**. You'll also need to install the Arduino IDE (that's the app used to program and edit files for the ESP32 module). Once you have both, open the downloaded files in Arduino IDE, connect your ESP32 to your computer with a USBC cable, and get ready to flash the code onto it. Double check the ESP32 is connected to the right port.

Part 1 — Server Setup (Private, local, HTTPS, port 9090)

0) Prerequisites (one-time)

1. **Node.js (LTS)**: install from nodejs.org (includes npm).
2. **INSTALL SERVER FILES HERE**
3. **OpenAI API key**: have it ready (you'll paste it in a file).

1) Get the server code

DOWNLOAD FILES and EXTRACT

FIRST AND MOST IMPORTANTLY GO TO ROUTE FOLDER THEN CHATGPT FILE AND INSERT YOUR API KEY. SAVE!

Open Powershell and write:

cd "C:\path\to\your\server" ←Paste Address to file

---

2) Install

npm install

If your template uses a hardcoded string (what you asked for):
Open routes/chatgpt.js (or routes/chatgpt.java if that's literally the filename—likely you meant JavaScript). Find the placeholder and paste your key **between the quotes**:

// Example pattern if your file expects a direct string:

const OPENAI_API_KEY = "sk-...paste-your-key-here...";

---

4) Generate a local HTTPS

Try one of these (depends on the repo):

npm start

or

npm run dev

or

node server.js (USUALLY THIS)

5. **Confirm it's listening**

- You should see something like:
  Listening on https://localhost:1010 (or http://localhost:1010)

6. **Find your LAN IP** (for the ESP32 to reach your PC) ipconfig

- Copy your active adapter's **IPv4 Address**, e.g. xxx.xxx.x.xx.

7. **Quick test (optional) TROUBELSHOOT**

- In a browser on the same Wi-Fi:
  https://xxx.xxx.x.xx:xxx/health (or /api/health if that's your route)
- If there's a POST chat route, you can also test with:
- "https://xxx.xxx.xxx:1010/api/chat" `
- "Content-Type: application/json" `
- "Hello from PowerShell" }'

If your server is **HTTP** not HTTPS, the URL will be http://... and you'll use NOT SECURE in the ESP32 (see below). If it's already set up with HTTPS, stick with **SECURE**.

---

2) Point your ESP32 firmware at the server

Open your Arduino project and edit "secrets.h" (names may vary, but follow this pattern):

Wi-Fi your ESP32 will join:

#define WIFI_SSID "YourWifiName"

#define WIFI_PASSWORD "YourWifiPassword"

#define SERVER "https://xxx.xxx.x.xx:1010" //

#define CHAT_NAME "ANY NAME you WANT"

// Choose ONE mode based on how your server runs**:**

#define SECURE // if your server is HTTPS

// #define NOT_SECURE // if your server is plain HTTP

---

3) Upload to the ESP32 (first-time tips)

1. **Select board/port** in Arduino IDE:

- Tools → Board (e.g., **ESP32 Dev Module** or **ESP32C3 Dev Module**)
- Tools → Port → pick the COM port that appears when you plug the ESP32 in.

2. **First upload = BOOT button trick**

- Hold the **BOOT** button on the ESP32.
- Click **Upload** in Arduino.
- When the IDE says **"Connecting…"**, **release BOOT**.
- If it stalls, tap **RESET (R)**, and try again with the timing.

3. **Serial Monitor** at …

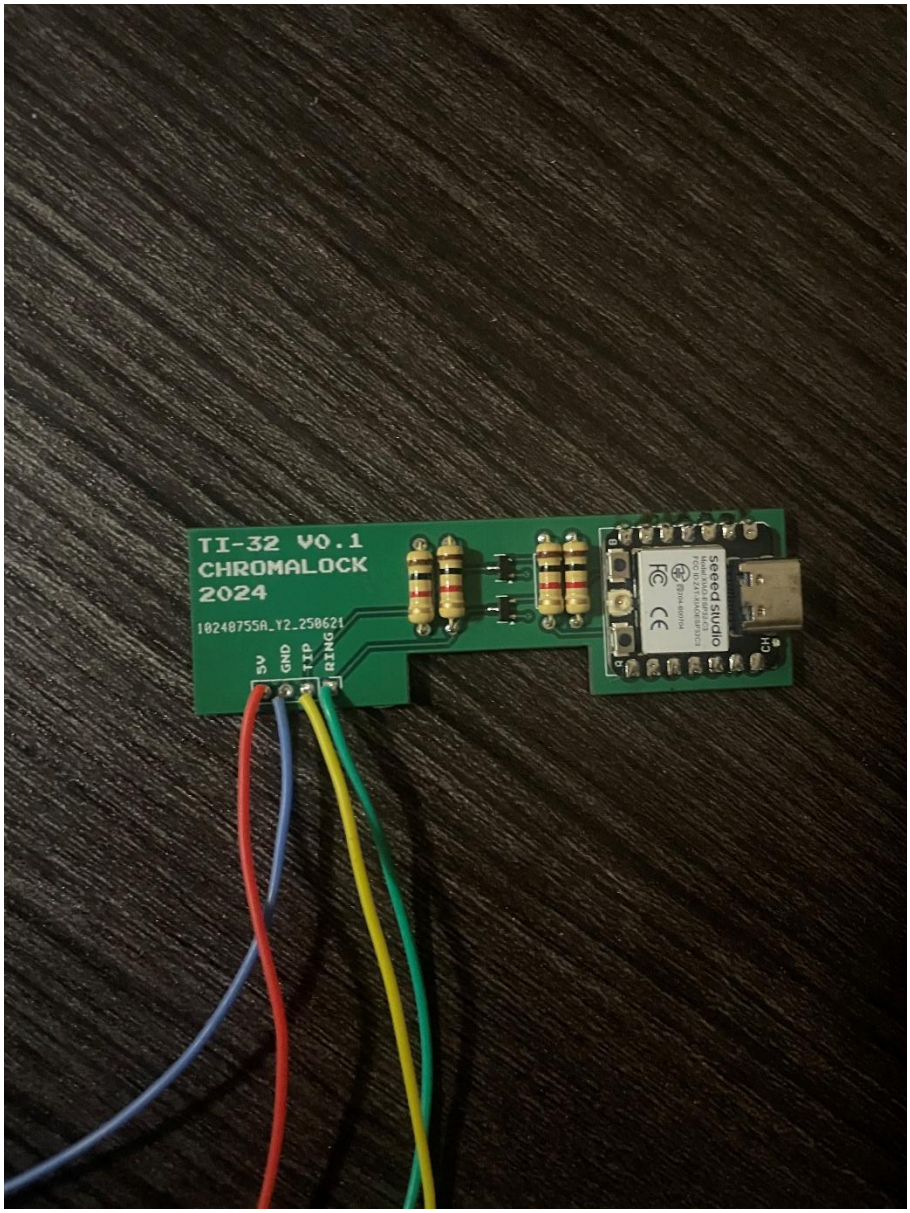
- You should see it connect to Wi-Fi and then call your server.
- If it can't reach the server, re-check:
    - Server is running and still says **listening on 1010**
    - ESP32 and PC are on the **same Wi-Fi**
    - Windows Firewall allowed Node.js on **Private** network
    - The **IPv4** in secrets.h matches ipconfig

Quick fixes if something's off

- **401"missing key"** → Your API (CHATGPT FILE) is empty or not loaded. Fill OPENAI_API_KEY and restart the server.
- **NO CONNECTION/FAILED** → Wrong IP or port, server not running, or firewall blocking.

FINAL STEPS:

Perfect. Your board’s here, you’ve got the tools, ESP32, and the server’s running. now just solder all the parts onto the PCB (small parts first, ESP32 last), then wire it to the calculator exactly like in the picture you’re following.




seeedstudio
TI-32 V0.1
CHROMALOCK
2024
AAA
1.5V


AAA
1.5V
USE NEW BATTERIES
OF THE SAME TYPE
TG2837MB-4424