

USER CREATED MEATLOAF DEVICE MANUAL

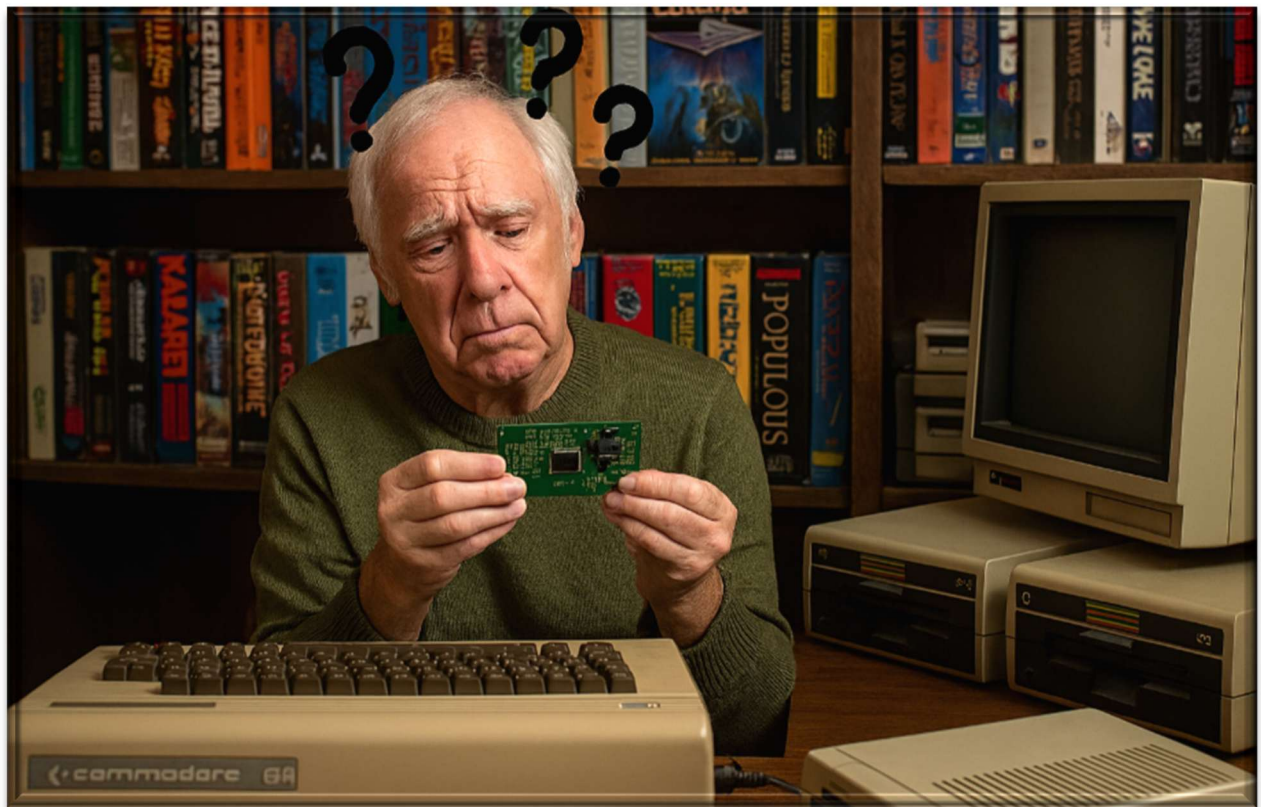
MY real MEATLOAF EXPERIENCE on the Commodore 64/128/VIC-20/Plus 4

The Meatloaf as shipped has features and functions that are amazing and intuitive. I was loading games and utilities from the internet in minutes.

SO THIS IS “MY” MEATLOAF JOURNEY

WWW.MYC64.COM WWW.MYC128.COM WWW.MYVIC20.COM

If you are a GitHub, python, VS Code and small controller board expert, tinkerer or designer (or just smarter than me) then you already know this document. I knew *none* of those things, To dig inside the Meatloaf, I started lost, wandered around and eventually wanted to correct that. I learned by having fun.



THIS MANUAL IS BASED ON REAL-WORLD EXPERIENCE AND COMMUNITY CONTRIBUTIONS. FOR THE LATEST UPDATES AND INFORMATION, ALWAYS REFER TO THE OFFICIAL MEATLOAF WEBSITE.

Ah!!, you're using Meatloaf! That's the excellent ESP32-based device that provides Wi-Fi connectivity and file serving for Commodore computers. It's a really clever modern solution that emulates disk drives and provides network access.

For a comprehensive list of Meatloaf commands, you'll want to check the official documentation.

www.meatloaf.cc

[idolpx/meatloaf: A Commodore IEC serial multi-device emulator](https://github.com/idolpx/meatloaf)

For the most up-to-date documentation, check the [Meatloaf Wiki](#) on GitHub. The project is actively developed and new features are being added regularly!

APPROXIMATE Table of Contents - A Work In Progress

1. [Introduction](#)
 2. [Quick Start Guide](#)
 3. [Hardware Setup](#)
 4. [Wi-Fi Configuration](#)
 5. [Essential Commands](#)
 6. [File Management](#)
 7. [Firmware Updates](#)
 8. [Advanced Features](#)
 9. [Troubleshooting](#)
 10. [Command Reference](#)
 11. [Appendices](#)
-

INTRODUCTION

What is Meatloaf? - THEY SAY:

“Meatloaf is a modern IEC device that brings internet connectivity and advanced storage capabilities to vintage Commodore computers. It connects to your C64, C128, or VIC-20 via the serial port and provides access to thousands of programs, games, and utilities through Wi-Fi connectivity.”

Key Features:

- Internet connectivity for vintage Commodore computers
- Access to the massive Meatloaf online archive
- Local SD card storage support
- WebDAV file management
- Firmware updates via web browser
- Multiple Wi-Fi configuration methods

What You'll Need:

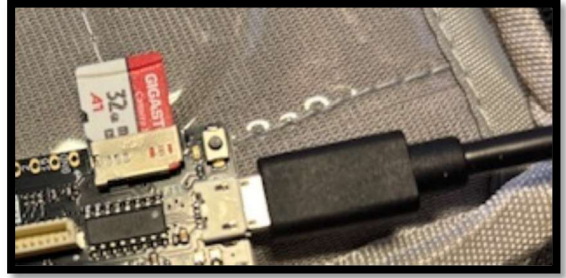
- Meatloaf device and case (mine flop around the desk uncased sometimes, not wise I suppose)
- Commodore 64, 128, or VIC-20 , Plus/4
- Wi-Fi network access
- USB power source (5V)
- MicroSD card (optional, up to 32GB recommended)
- Micro USB cable



QUICK START GUIDE

30-Second Setup -

1. Power down your Commodore computer
2. Insert SD card into Meatloaf (optional)
3. Connect Meatloaf to serial port on Commodore
4. Connect USB power to Meatloaf
5. Power on Commodore
6. Configure Wi-Fi: Type `LOAD"ML:WIFI",8` and press ENTER
 - MEATLOAF WIFI SSID CHANGER
 - ENTER WIFI SSID? YOUR SSID
 - ENTER WIFI PASSWORD? YOUR PASSWORD
 - SETSSID: YOUR SSID, YOUR PASSWORD
7. Test connection: Type `LOAD"ML:GOTD",8` for Game of the Day - Fingers Crossed "DIG DUG!"



 You're ready to explore!



HARDWARE SETUP

Physical Connection

⚠ Important: Always power down your Commodore before connecting or disconnecting devices!

Step-by-Step Connection:

1. Prepare the Meatloaf
 - Insert formatted SD card (if using local storage)
 - Ensure case is properly assembled
2. Connect to Commodore
 - Locate the serial port on the back of your Commodore
 - Gently insert Meatloaf device until firmly seated
 - Do not force the connection
3. Power the Meatloaf
 - Connect micro-USB cable to Meatloaf
 - Connect other end to USB power source:
 - USB wall adapter (5V, 1A minimum)
 - Computer USB port
 - Powered USB hub
 - Red LED should blink rapidly when powered
4. Power On
 - Turn on your Commodore computer
 - Meatloaf will initialize automatically

Visual Indicators

LED Pattern	Status
Rapid blinking red	Powered, no Wi-Fi connection
Slow blinking red	Connecting to Wi-Fi
Solid blue/green	Connected and ready
Off	No power or hardware issue



WI-FI CONFIGURATION

Meatloaf offers multiple methods to configure Wi-Fi. Choose the method that works best for you.

Method 1: Commodore Interface (Recommended)

This is the most straightforward method using your Commodore directly.

1. Access Wi-Fi Setup

2. **LOAD"ML:WIFI",88**

3. Enter Credentials

- Type your Wi-Fi network name (SSID)
- Type your password
- Important: Credentials are entered in PETSCII uppercase but stored as lowercase

```
enter wifi ssid? COM-COM
enter wifi password? COMMODORE
setssid:COM-COM,COMMODORE
```

4. For Mixed Case Passwords

- Use Shift + Commodore key to enable lowercase mode
- Your screen will show lowercase characters

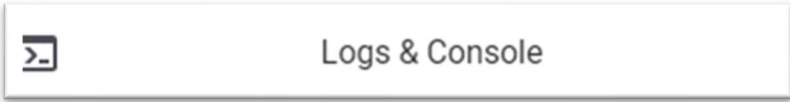
5. Save Settings

- Follow on-screen prompts to save configuration
- Meatloaf will attempt to connect automatically



ALTERNATE METHOD 1: WEB-BASED TERMINAL (ADVANCED)

Perfect for users comfortable with command-line interfaces. OR This can be done immediately after a flash firmware process.



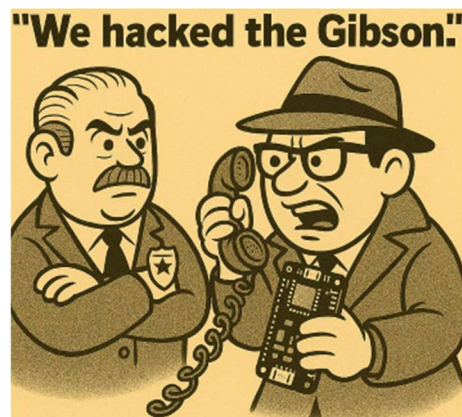
The website will open a serial terminal at 460800 baud. Do you need to know this? Maybe, if you try connecting with a serial terminal software like `tera term`.

I press the rest PB on the board at this point. I get a screen full of...

```
[17:10:53][lib/meatloaf/meatloaf.cpp:296] File(): source good rootfs[1][sd/.rom/dos1541]  
[17:10:53][lib/meatloaf/meatloaf.cpp:303] File(): target good rootfs[1][sd/.rom/dos1541]  
[17:10:53]^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  
[17:10:53][lib/device/iec/drive.h:141] setROM(): Drive ROM Loaded file[/sd/.rom/dos1541] size[0]  
[17:10:53]Attached Meatloaf device #30  
[17:10:53]Available heap: 3948608  
[17:10:53]Setup complete @ 562 (550ms)  
[17:10:53]WiFiManager::start() complete  
[17:10:53]MAC Address: 00:80:10:C0:05:DC  
[17:10:53]WIFI EVENT STA START
```

1. Flash Firmware via [meatloaf.cc](#)
2. Open Terminal using "Logs & Console" option
3. type:
4. Scan: and get a list available Wi-Fi SSID
5. Connect "your Wi-Fi SSID" "Your Wi-Fi password"
6. Reset to see boot screen and success

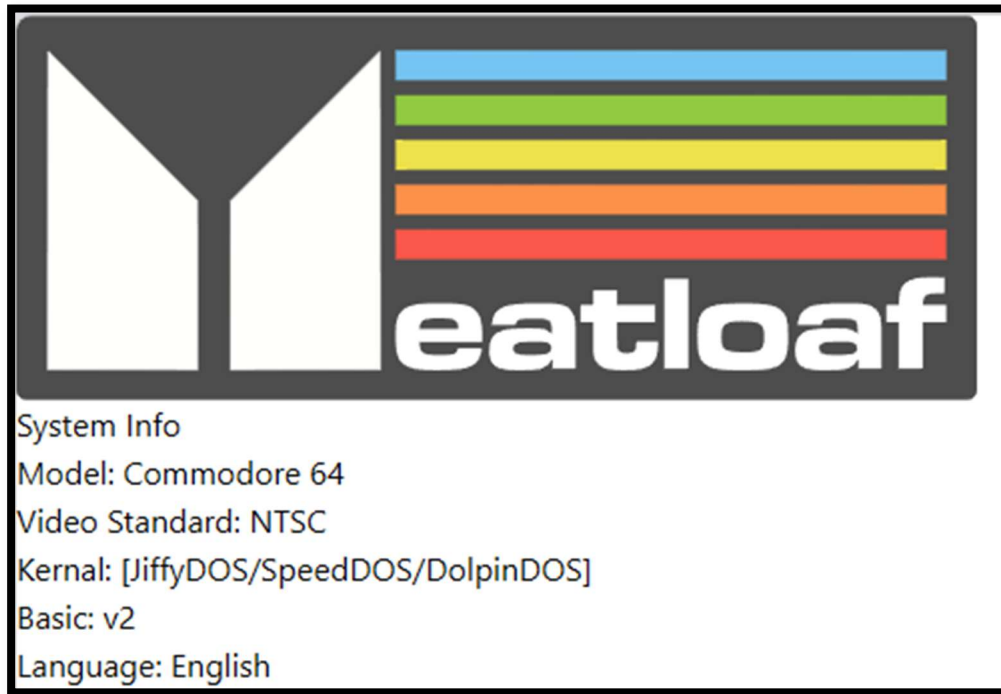
```
[17:49:48]meatloaf[/]# WIFI_EVENT_STA_CONNECTED
[17:49:51]IP_EVENT_STA_GOT_IP
[17:49:51]Obtained IP address: 192.168.1.22
[17:49:51]SNTP client start
[17:49:51]Starting web server on port 80
[17:49:51]WWW/WS/WebDAV Server Started!
[17:49:51]mDNS Service Started!
[17:49:51]TCP Server Started! (PORT 23)
```



MEATLOAF DEFAULT PORT 80 WEBPAGE - A TEST

[CONNECTION VIA WEB BROWSER ON A DEVICE WITHIN THE NETWORK]

In MY case I am 192.168.1.22. If you type this in a web browser



At the moment , this appears to be view only

MEATLOAF COMMANDS

NAVIGATION COMMANDS

Command	Description	Example
<code>LOAD"\$",8</code>	List current directory	Basic directory listing
<code>LOAD"CD:SD",8</code>	Change to SD directory	Access local storage
<code>LOAD"CD..",8</code>	Move up one directory	Navigate folders
<code>LOAD"CD/",8</code>	Alternate up directory	Same as above
<code>LOAD"CD//",8</code>	Alternate up directory	Same as above
<code>LOAD"CD+",8</code>	Back 1 Directory	Same as above
<code>LOAD"CD↑",8</code>	To Root and Clears URL, ROOT, Path	Same as above



QUICK ACCESS COMMANDS

Command	Description
LOAD"ML:GOTD",8	Game of the Day
LOAD"ML:GAME*",8	Random game from archive
LOAD"ML:ARCADE*",8	Random arcade game
LOAD"ML:LATEST",8	Recently added content
LOAD"ML:SEARCH",8	Search the archive

SYSTEM COMMANDS

Command	Description
LOAD"ML:CONFIG",8	Configuration utility
LOAD"ML:UPDATE",8	Update firmware
LOAD"ML:WIFI",8	Wi-Fi settings
LOAD"RESET",8	Reset/unmount disk image

FILE BROWSER COMMANDS

Command	Description
LOAD"ML:FB",8	Standard file browser
LOAD"ML:FB64",8	Enhanced file browser
LOAD"ML:MLB",8	Meatloaf browser



<https://github.com/idolpx/meatloaf/wiki/Using-Your-Meatloaf#traverse-your-file-system>

[Using Your Meatloaf · idolpx/meatloaf Wiki](#)

```
LOAD"CD:GAMES",8  = CD TO GAMES DIR
LOAD"CD//",8       = CD TO FLASH ROOT
                   OR URL ROOT
                   OR ARCHIVE ROOT
                   OR IMAGE ROOT
LOAD"CD←",8        = CD BACK ONE DIR
LOAD"CD↑",8        = CD TO FLASH ROOT
                   CLEARS URL
                   CLEARS ARCHIVE
                   CLEARS IMAGE
                   CLEARS PATH
```

File Management

Local SD Card Storage

Formatting SD Card:

- Use FAT32 format
- 32GB maximum recommended size
- Create folders for organization

Directory Structure Example:

SD:/

```
|— GAMES/  
|  |— ARCADE/  
|  |— STRATEGY/  
|  |— DEMOS/  
|— UTILITIES/  
|— IMAGES/  
|— PERSONAL/
```

WebDAV File Management

Access your Meatloaf's storage using WebDAV-compatible clients.

Recommended Client: Cyberduck

1. Download from <https://cyberduck.io/>
 2. Create new WebDAV connection
 3. Server: `http://[meatloaf-ip-address]`
 4. Port: 80
 5. Connect and manage files
-

ADVANCED FEATURES ARCHIVE SHORTCUTS

The Meatloaf archive includes thousands of programs accessible via short codes:

Popular Games:

- `LOAD"ML:JJ",8` - Jumpman Junior
- `LOAD"ML:SPYVSSPY",8` - Spy vs Spy
- `LOAD"ML:TAPPER",8` - Tapper
- `LOAD"ML:STARWARS",8` - Star Wars

Special Features:

- `LOAD"ML:ISS",8` - Track International Space Station
- `LOAD"ML:PTERM",8` - PETSCII Terminal
- `LOAD"ML:HIGHSORE",8` - High Score Archive

Profile System

Create a Meatloaf profile for personalized features:

1. Register: `LOAD"ML:REGISTER",8`
2. Login: `LOAD"ML:LOGIN",8`
3. Pair Device: `LOAD"ML:PAIR",8`
4. View Profile: `LOAD"ML:PROFILE",8`

Custom Mounting

Mount network resources or cloud storage:

`mount http://example.com/files/ drive_id`

TROUBLESHOOTING

Common Issues

Meatloaf Not Recognized:

- Check serial port connection
- Verify power LED is active
- Try different USB power source
- Reseat device connection

Wi-Fi Connection Fails:

- Verify network credentials
- Check signal strength
- Restart Meatloaf: `LOAD"ML:RESTART",8`
- Try different configuration method

Slow Loading:

- Check Wi-Fi signal strength
- Verify internet connection
- Try different archive mirror
- Clear browser cache if using web interface

SD Card Issues:

- Reformat SD card (FAT32)
- Check for physical damage
- Try smaller capacity card
- Ensure proper insertion

Error Messages

Error	Cause	Solution
"DEVICE NOT PRESENT"	Connection issue	Check cables and power
"FILE NOT FOUND"	Invalid path/file	Verify spelling and location
"NETWORK ERROR"	Wi-Fi problem	Check connection and credentials
"DISK FULL"	Storage full	Clean up SD card

Getting Help

Online Resources:

- Official site: <https://meatloaf.cc/>
- Community forums and Discord
- GitHub repository for technical issues

Diagnostic Commands:

- `LOAD"ML:WIFITEST",8` - Test Wi-Fi connectivity
 - `LOAD"ML:STRESSTEST",8` - Hardware stress test
 - Terminal: `sysinfo` - System information
 - Terminal: `meminfo` - Memory usage
-

TERMINAL COMMANDS

System Information

help # Command list
sysinfo # System info
meminfo # Memory usage
date # System time
restart # Reboot system

Network Operations

scan # Scan Wi-Fi networks
connect "ssid" "password" # Connect to Wi-Fi
ping hostname # Test connectivity
ipconfig # Network information
wget url filename # Download file

File Management

ls # List files
cd directory # Change directory
pwd # Current directory
cat filename # Display file
cp source dest # Copy file
mv old new # Move/rename file
rm filename # Delete file
mkdir dirname # Create directory
rmdir dirname # Remove directory

Hardware Control

led on/off # Control LED
pinMode pin mode # Set pin mode
digitalRead pin # Read digital pin
digitalWrite pin value # Write digital pin
analogRead pin # Read analog pin

APPENDICES

Appendix A: Supported File Formats

Format	Description	Usage
.PRG	Program files	Standard Commodore programs
.D64	Disk images	Full disk emulation
.T64	Tape images	Tape file collections
.P00	PC64 format	Cross-platform compatibility
.SEQ	Sequential files	Text and data files

Appendix B: Hardware Specifications

Meatloaf Device:

- Microcontroller: ESP32-based (Lolin D32 Pro)
- Storage: MicroSD slot (up to 32GB)
- Connectivity: Wi-Fi 802.11 b/g/n
- Power: 5V USB, ~200mA
- Interface: Commodore IEC serial port

Compatible Systems:

- Commodore 64 (all revisions)
- Commodore 128 (64 and 128 modes)
- VIC-20
- Plus/4 and C16 (with adapter)

Appendix C: USB Driver Installation

Windows:

1. Download CH340 drivers from manufacturer
2. Install with administrator privileges
3. Verify in Device Manager under "Ports"

macOS:

1. Install automatically via System Preferences
2. May require security approval in System Preferences

Linux:

1. Drivers typically included in kernel
2. Check with lsusb command
3. Add user to dialout group if needed

Appendix D: Troubleshooting Flowchart

PROBLEM: MEATLOAF NOT WORKING

- |— No power LED?
- | |— Check USB cable
- | |— Try different power source
- | |— Check device connections
- |— Power but no response?
- | |— Verify serial connection
- | |— Check Commodore power
- | |— Try different device
- |— Loads but no Wi-Fi?
- | |— Reconfigure Wi-Fi settings
- | |— Check network availability
- | |— Restart device
- |— Works locally but not online?
- | |— Test internet connection
- | |— Check firewall settings
- | |— Verify archive accessibility

Appendix E: Community Resources

Official Channels:

- Website: <https://meatloaf.cc/>
- Documentation: Available on main site
- Firmware releases: GitHub repository

Community Support:

- Discord server for real-time help
- Reddit communities: r/commodore, r/c64
- Facebook groups: Commodore enthusiasts
- YouTube channels: Demonstrations and tutorials

THIS MANUAL IS BASED ON REAL-WORLD EXPERIENCE AND COMMUNITY CONTRIBUTIONS. FOR THE LATEST UPDATES AND INFORMATION, ALWAYS REFER TO THE OFFICIAL MEATLOAF WEBSITE.

WIFI SETUP - PYTHON SCRIPT (FOR DEVELOPERS)

Ideal for automation or when other methods fail.

Prerequisites:

- Python installed on your computer
- pyserial library (pip install pyserial)

Script Usage:

```
import serial
```

```
import time
```

```
PORT = 'COM6' # Adjust for your system
```

```
BAUDRATE = 460800
```

```
ssid = "Your_Network_Name"
```

```
password = "Your_Password"
```

```
command = f'connect "{ssid}" "{password}"\n'.encode()
```

try:

```
with serial.Serial(PORT, BAUDRATE, timeout=2) as ser:
```

```
    print(f"✅ Connected to {PORT}")
```

```
    ser.write(command)
```

```
    time.sleep(1.0)
```

```
    response = ser.read(256)
```

```
    print(f"📡 Response:", response.decode(errors='ignore'))
```

except Exception as e:

```
    print(f"❌ Error: {e}")
```