Esp-Terminal Commands Documentation

September 29, 2025

Overview

Commands are case-sensitive and must match exactly, including spaces where specified. Responses are sent via UART, indicating success or failure. Unrecognized commands return "Command Not Found." Commands are grouped by category: System, SD Card, WiFi, HTTP/Network, BLE, and Classic Bluetooth. Each command includes its syntax, purpose, and typical output.

1 System Commands

Command Syntax	Description	Output
`sysinfo`	Retrieves system information (e.g., via a `sysinfo` function).	System info string (up to 512 bytes).

Table 1: System Commands

2 SD Card Commands

These commands require the SD card to be initialized using `enable sdcard`. File paths are relative to `/sdcard/`.

Command Syntax	Description	Output
`enable sdcard`	Initializes the SD card.	"SD card initialized successfully." or "Failed to initialize SD card."
`sdcardinfo`	Retrieves SD card information (e.g., capacity, type).	SD card info string (up to 512 bytes).
`ls`	Lists files and directories in the current directory.	File/directory list (up to 1024 bytes).
`mkdir <directory_name>`	Creates a new directory.	"Directory created." or "Failed to create directory."

rmdir <directory_name>	Removes the specified directory (must be empty).	"Directory removed." or "Failed to remove directory."
cd <directory_name>	Changes the current working directory.	"Directory changed." or "Failed to change directory."
pwd	Prints the current working directory path.	Directory path with newline.
rm <file_name>	Removes the specified file.	"File removed." or "Failed to remove file."
read <file_name>	Reads the content of the specified file.	File content with newline or error message.
write <file_name> <content>	Writes content to the specified file (overwrites if exists).	"File written." or "Failed to write file."
cp <source> <destination>	Copies a file or directory.	"Copy successful." or "Failed to copy."
mv <source> <destination>	Moves (renames) a file or directory.	"Move successful." or "Failed to move."

Table 2: SD Card Commands

3 WiFi Commands

These commands manage WiFi in STA, AP, APSTA, or NULL modes. Use `wifienable mode <mode>` first. Some require an active connection in STA or APSTA mode.

Command Syntax	Description	Output
wificonfig	Retrieves WiFi configuration details.	WiFi config string or "Failed to get WiFi config."
wifienable mode <mode>	Enables WiFi in sta, ap, apsta, or null mode.	"WiFi enabled in specified mode." or "Failed to enable WiFi: <error_code>" or "Invalid mode: null, sta, ap, apsta".
wifidisconnect	Disconnects from the current WiFi network (STA or APSTA mode).	"Disconnected from WiFi network." or "WiFi not in STA or APSTA mode."
wificonnect <SSID> <PASSWORD>	Connects to a WiFi access point (STA or APSTA mode).	"Trying to connect to <SSID>. Type wificonnect check for connection status."

`wificonnect check`	Checks WiFi connection status (STA or APSTA mode).	Connection status or "Not connected to any WiFi." or "WiFi not in STA or APSTA mode."
`wificreate <SSID> <PASSWORD> <CHANNEL>`	Creates a WiFi access point (AP or APSTA mode).	"Creating AP <SSID> on channel <CHANNEL>. Type wificreate check for status."
`wificreate check`	Retrieves access point status.	AP status string or "Failed to get AP status."
`wificreate disable`	Disables the access point (AP or APSTA mode).	"AP disabled." or "WiFi not in AP or APSTA mode."
`wifiscan` or `wifiscan update`	Scans WiFi networks and saves to `/sdcard/wifi_scan_output.txt` (STA or APSTA mode).	"Scan completed, results saved to /sd-card/wifi_scan_output.txt" or "Scan failed: <error_code>" or "WiFi not in STA or APSTA mode."
`wifiscan result`	Reads WiFi scan results from `/sdcard/wifi_scan_output.txt`.	Scan results with newline or error message.
`wifipktcap <type> <seconds> <filename>`	Captures WiFi packets (`beacon`, `probe`, `auth`, or `deauth`) for `seconds` and saves to `filename`.	"Capture completed, saved to file." or "Capture failed: <error_code>".

Table 3: WiFi Commands

4 HTTP/Network Commands

These require WiFi in STA or APSTA mode with an active connection. Files are saved to `/sdcard/`.

Command Syntax	Description	Output
`get <url> <file>`	Performs HTTP GET request and saves response to `file`.	"GET request successful, response saved to file." or "GET request failed: <error_code>" or "WiFi not connected in STA or APSTA mode."

3

`post <url>`
`"<post_data>"`
`<file>`
	Performs HTTP POST with quoted `post_data` and saves response to `file`.	"POST request successful, response saved to file." or "POST request failed: <error_code>" or "WiFi not connected in STA or APSTA mode." or "Invalid command format."

`upload <url>`
`<file>`
`[content_type]`
`[timeout_ms]`
`[response_file]`
	Uploads `file` to `url` via HTTP POST. Optional: `content_type` (default: `application/octet-stream`), `timeout_ms` (default: 10000), `response_file`. URL must start with `http://` or `https://`.	"File upload successful." or "File upload failed: <error_code>. Check URL, file, or network." or validation errors.

`download <url>` `<file>`	Downloads content from `url` via HTTP GET and saves to `file`.	"Download successful, file saved." or "Download failed: <error_code>" or "WiFi not connected in STA or APSTA mode."

Table 4: HTTP/Network Commands

5 BLE (Bluetooth Low Energy) Commands

Command Syntax	Description	Output
`bleenable`	Enables BLE functionality.	"BLE enabled successfully." or "Failed to enable BLE."
`bleconfig`	Retrieves BLE configuration details.	BLE config string (up to 128 bytes).
`blescan` `<filename.txt>`	Scans for BLE devices and saves results to `filename.txt` on SD card.	Success message or error message.
`bleadvstart` `<data>`	Starts BLE advertising with custom `data`.	"BLE advertising started." or "Failed to start BLE advertising."
`bleadvstop`	Stops BLE advertising.	"BLE advertising stopped." or "Failed to stop BLE advertising."

Table 5: BLE Commands

6 Classic Bluetooth Commands

Command Syntax	Description	Output
`blueenable`	Enables Classic Bluetooth functionality.	"Classic Bluetooth enabled successfully." or "Failed to enable Classic Bluetooth."
`blueconfig`	Retrieves Classic Bluetooth configuration details.	BT config string (up to 128 bytes).
`bluescan <filename.txt>`	Scans for Classic Bluetooth devices and saves to `filename.txt`.	Success message or error message.
`blueconnect <mac_address>`	Connects to a Bluetooth device with `mac_address`.	Connection status message or error message.
`bluedisconnect`	Disconnects from the current Bluetooth connection.	Disconnection status message or error message.
`bluecreate <Name> [PIN]`	Creates a Bluetooth server with `Name` and optional `PIN`.	Server creation status message or error message.
`bluecreate disable`	Disables the Bluetooth server.	Server disable status message or error message.
`bluesend <data>`	Sends `data` over the active Bluetooth connection.	Send status message or error message.
`bluesendnc <mac_address> <data>`	Sends `data` to `mac_address` (non-connected mode).	Send status message or error message or "No data provided."
`bluesendfile <filename.txt>`	Sends content of `filename.txt` over Bluetooth.	Send status message or error message.
`bluesendfilenc <mac_address> <filename.txt>`	Sends `filename.txt` to `mac_address` (non-connected mode).	Send status message or error message or "No filename provided."

Table 6: Classic Bluetooth Commands