

ESP32-CAM GRBL CNC Controller Documentation

Overview

This firmware transforms an ESP32-CAM module into a powerful CNC controller with integrated camera capabilities. The system communicates with an Arduino running GRBL firmware over serial communication, providing a complete web-based interface for machine control, camera monitoring, and automated scanning operations.

System Architecture

Hardware Components

- **ESP32-CAM Module:** Primary controller with integrated camera
- **Arduino Board (Uno/Nano/Mega):** Running GRBL firmware for CNC control
- **SD Card:** For storing captured images and scan data
- **WiFi Network:** For remote access and control

Communication Flow

Web Browser ↔ ESP32-CAM Web Server (Port 80)



ESP32-CAM Stream Server (Port 81)



ESP32-CAM ↔ Arduino (UART Serial @ 115200 baud)



GRBL Firmware



Stepper Motors/Drivers

Setup Instructions

Prerequisites

- ESP32-CAM module with camera
- Arduino board with GRBL firmware installed
- SD card formatted as FAT32

- WiFi network credentials

Configuration (Before Upload)

Edit these parameters in the code (if needed):

```
// HTTP Server Ports
const uint16_t HTTP_PORT = 80;      // Web interface
const uint16_t STREAM_PORT = 81;    // Camera stream

// GRBL Communication
const uint32_t GRBL_BAUD = 115200; // GRBL baud rate
```

Wifi Setup

Use Serial terminal to connect to wifi, steps are as follow:

- Send **menu** for Main Menu

```
=====
Main Menu
=====
1. WiFi Configuration
- Configure WiFi networks
2. Credentials Management
- Manage stored networks
3. System Status
- Show system information
4. Exit Menu
- Return to command mode
=====
Timeout: 179 seconds
Press '0' to exit menu
=====
```

- Select option 1

```
Select option (1-4): 1

=====
WiFi Configuration
=====
1. Scan Networks
- Scan for available networks
2. Connect to Network
- Connect to WiFi network
3. Auto Connect
- Auto-connect to stored networks
4. Check Connection
- Check current WiFi status
5. Back to Main Menu
- Return to main menu
=====
Timeout: 179 seconds
Press '0' to exit menu
=====
```

- Select 1 to scan networks

```
Select option (1-5): 1

=====
 WiFi Network Scan
=====
 Available networks:
=====
 1. HUAWEI Y5 2017      RSSI: -62  [SECURED]
=====
```

- Select your network, add password

```
Select network to connect to (0 to cancel)
> 1

=====
 Connect to WiFi Network
=====

Enter WiFi password (leave blank for open network)
> *****

Connecting to HUAWEI Y5 2017...
Attempting to connect to: HUAWEI Y5 2017
✓ Connected to HUAWEI Y5 2017 successfully
IP Address: 192.168.43.99
✓ Connection successful!
```

Hardware Connections

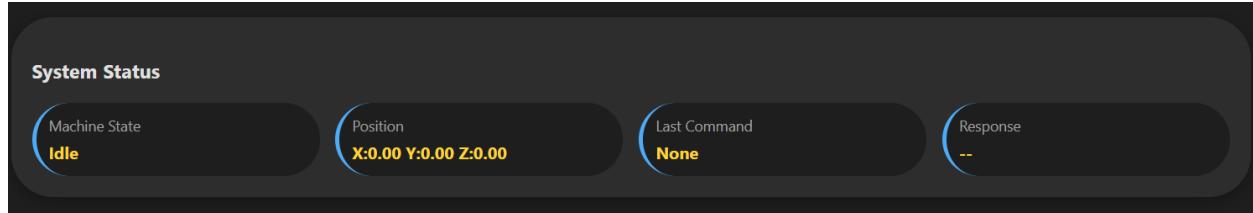
Connect ESP32-CAM to Arduino GRBL:

- ESP32 TX (GPIO1) → Arduino RX
- ESP32 RX (GPIO3) → Arduino TX
- GND → GND

Web Interface Documentation

Main Controls Tab

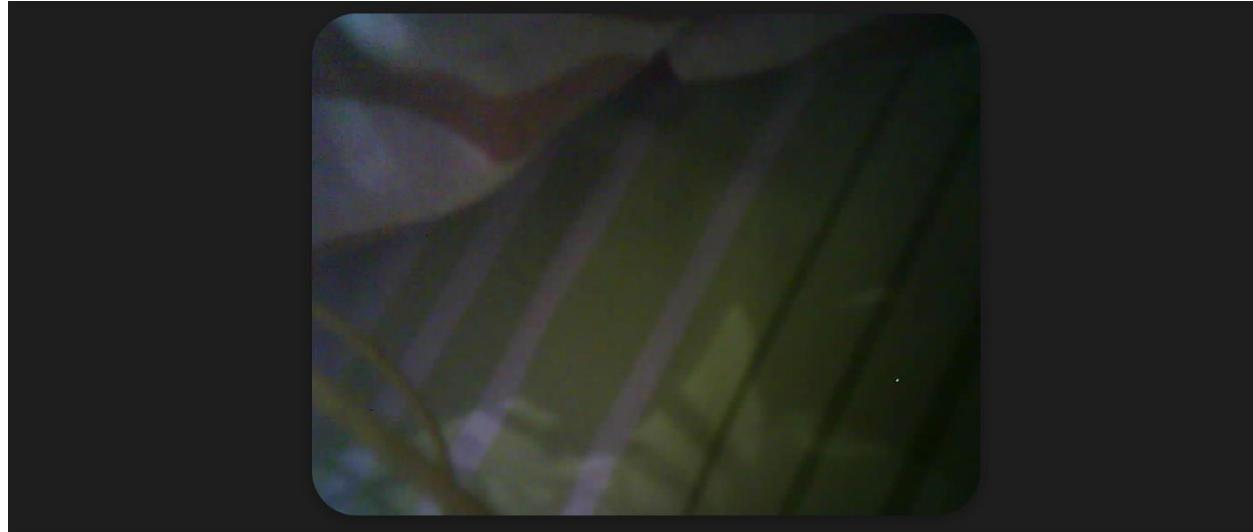
System Status Section



Displays real-time machine information:

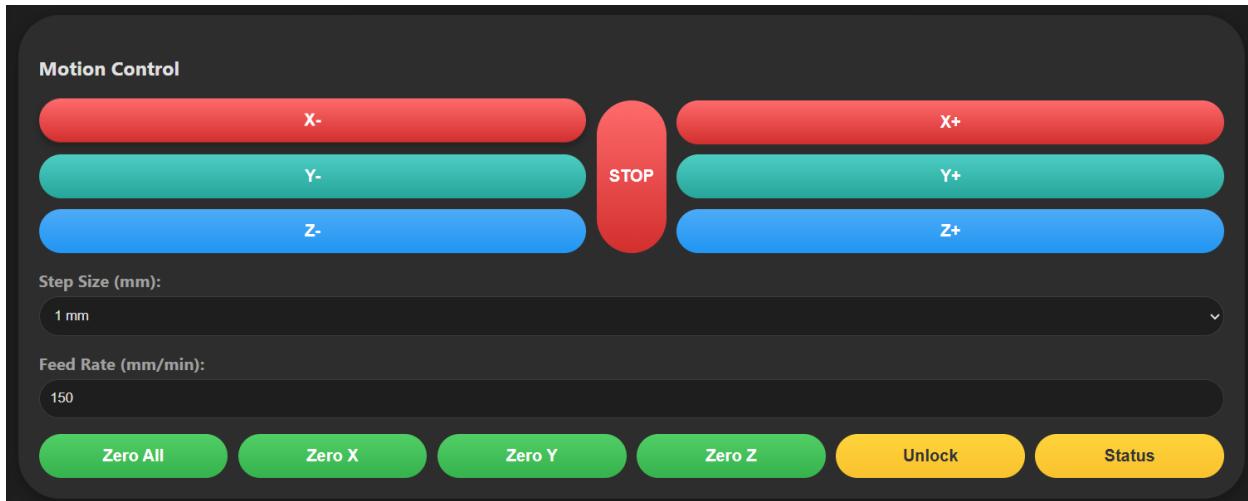
- **Machine State:** Idle, Run, Hold, Alarm, etc.
- **Position:** Current coordinates (X, Y, Z) in work position
- **Last Command:** Most recent command sent to GRBL
- **Response:** GRBL's response to commands

Camera Feed



- Live MJPEG stream from ESP32-CAM camera
- Automatically loads from [http://\[esp-ip\]:81/stream](http://[esp-ip]:81/stream)
- Resolution and quality adjustable via Camera Settings tab

Motion Control



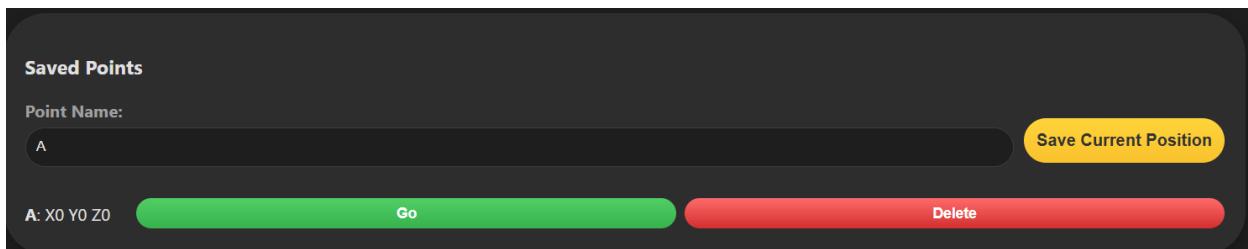
Jogging Controls:

- Axis buttons (X+, X-, Y+, Y-, Z+, Z-) for manual movement
- **Step Size:** Configurable movement increments (0.1mm to 100mm)
- **Feed Rate:** Movement speed in mm/min (10-5000)

Quick Actions:

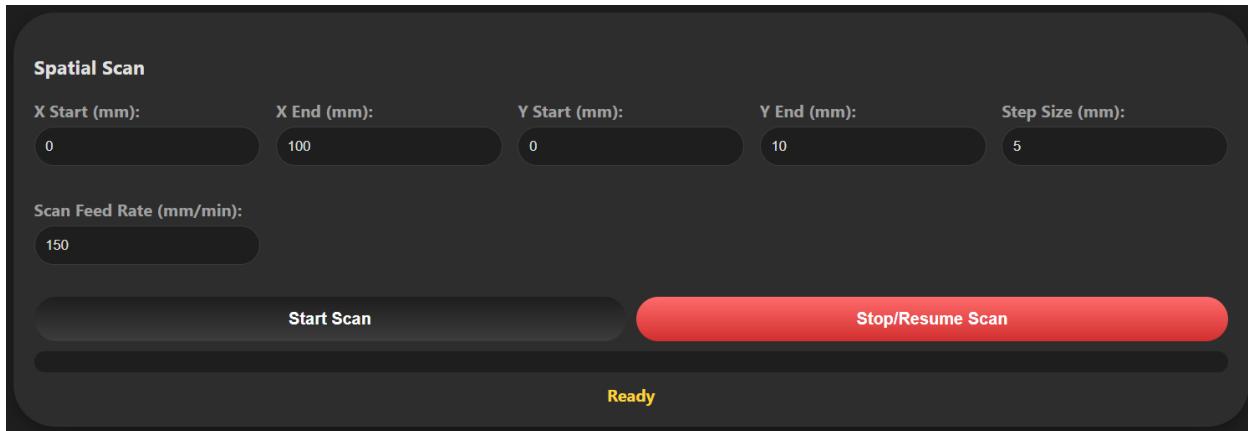
- **Zero All:** Sets all axes to zero position (G10 L20 P0 X0 Y0 Z0)
- **Zero X/Y/Z:** Sets individual axis to zero
- **Unlock:** Removes GRBL alarm state (\$X)
- **Status:** Refreshes machine status

Saved Points



- **Save Current Position:** Stores current coordinates with custom name
- **Go To Point:** Moves machine to saved position
- **Delete Point:** Removes saved position
- Points persist across sessions and are stored in ESP32 memory

Spatial Scan

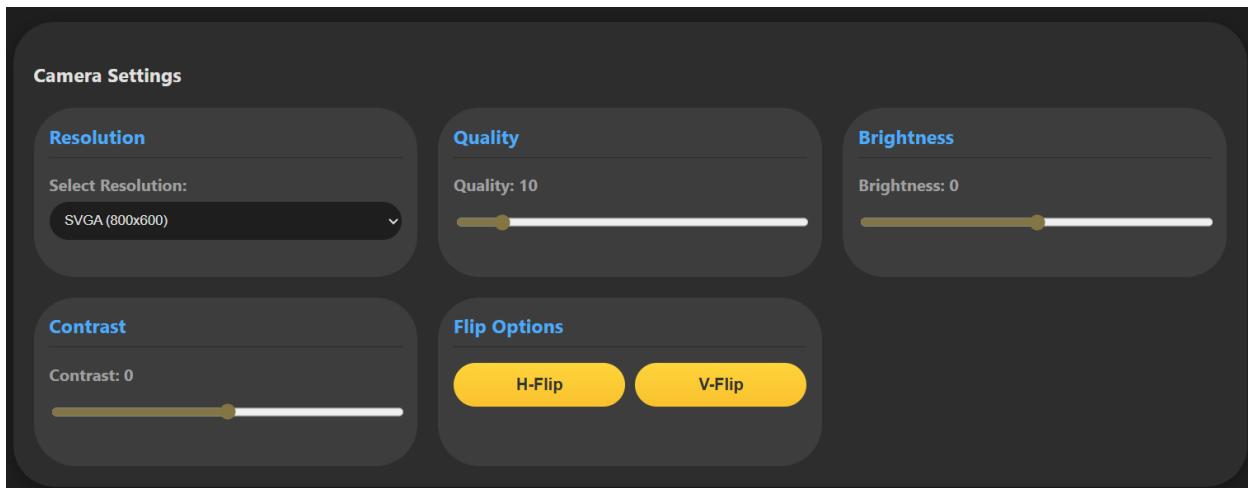


Automated grid scanning with camera capture:

- **X/Y Start/End:** Define scan area boundaries
- **Step Size:** Distance between capture points
- **Scan Feed Rate:** Movement speed during scanning
- **Progress Bar:** Visual scan progress indicator
- **Stop/Resume:** Pause and resume scanning operations

Images are saved to SD card in /scan_[number] folders with sequential numbering.

Camera Settings Tab



Resolution

- QVGA (320×240)

- VGA (640×480)
- SVGA (800×600) - Default
- XGA (1024×768)
- HD (1280×720)
- UXGA (1600×1200)

Quality

- Adjustable JPEG quality (5-63)
- Lower values = better quality, larger files
- Higher values = worse quality, smaller files

Brightness & Contrast

- Range: -2 to +2
- Real-time adjustment of camera sensor parameters

Flip Options

- **H-Flip:** Horizontal mirror
- **V-Flip:** Vertical flip

Motion Settings Tab



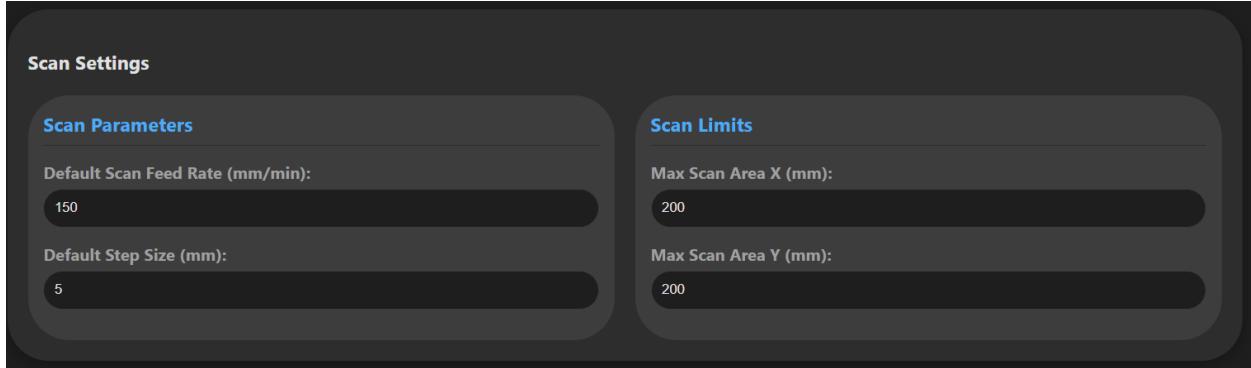
Default Feed Rates

- Sets default movement speed for all operations
- Range: 10-5000 mm/min

Step Increments

- **Small Step:** Fine movement adjustment (0.01-1mm)
- **Large Step:** Coarse movement adjustment (1-100mm)

Scan Settings Tab



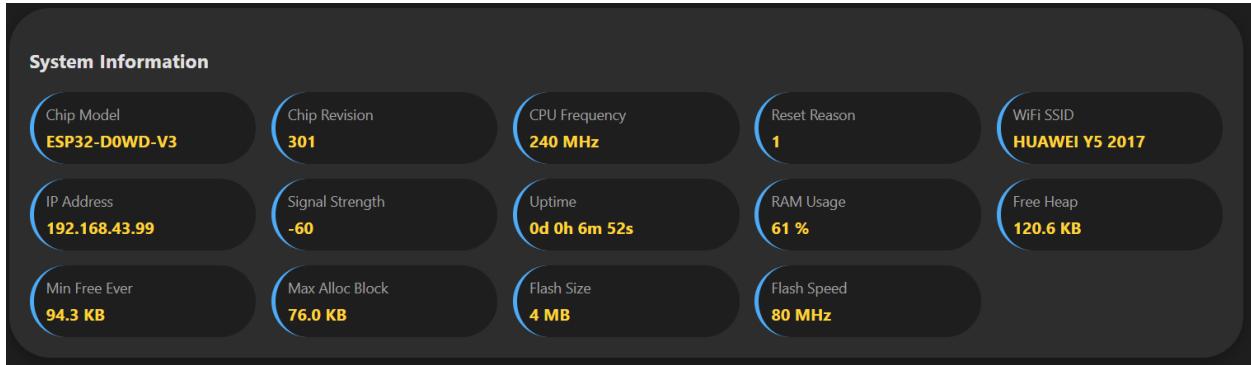
Scan Parameters

- **Default Scan Feed Rate:** Preset speed for scanning operations
- **Default Step Size:** Preset distance between scan points

Scan Limits

- **Max Scan Area X/Y:** Defines maximum allowable scan dimensions
- Prevents accidental large movements

System Info Tab



Hardware Information

- **Chip Model:** ESP32 variant identifier
- **Chip Revision:** Silicon revision
- **CPU Frequency:** Current processor speed

- **Reset Reason:** Boot cause analysis

Network Information

- **WiFi SSID:** Connected network name
- **IP Address:** ESP32's assigned IP
- **Signal Strength:** RSSI in dBm
- **Uptime:** System runtime duration

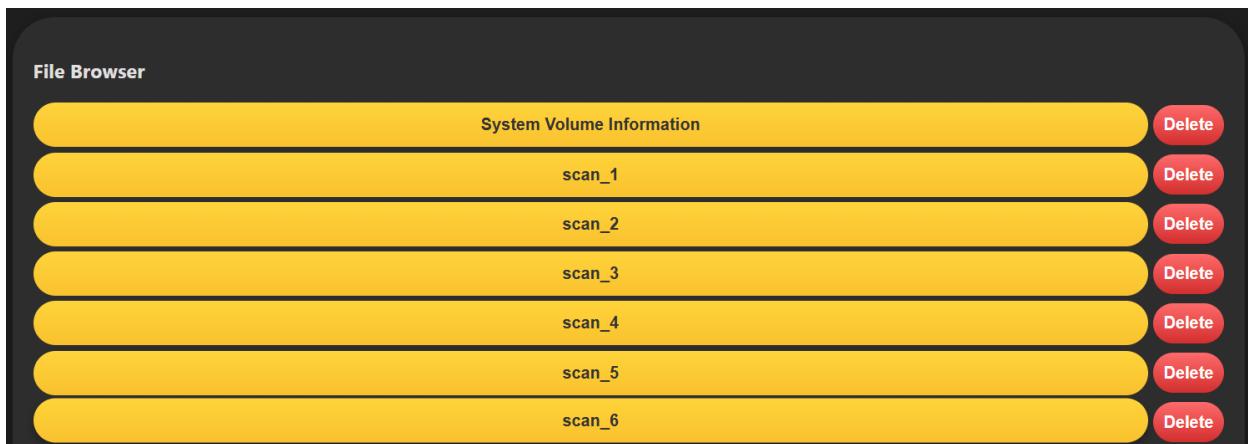
Memory & Performance

- **RAM Usage:** Current heap utilization percentage
- **Free Heap:** Available memory in KB
- **Min Free Ever:** Lowest memory point since boot
- **Max Alloc Block:** Largest contiguous free memory block

Storage

- **Flash Size:** Total onboard flash memory
- **Flash Speed:** Memory clock frequency

File Browser Tab



Directory Navigation

- Browse scan folders (/scan_1, /scan_2, etc.)
- Navigate through folder hierarchy
- Parent directory access

File Management

- **Download:** Retrieve captured images
- **Delete Folder:** Remove entire scan directories with confirmation
- Files displayed with visual buttons for easy access

API Documentation

Core GRBL Commands

Send Command to GRBL

GET /cmd?c=[command]

Parameters:

- c: URL-encoded GRBL command (required)

Example:

```
curl "http://192.168.1.100/cmd?c=G0%20X10%20Y20%20F500"
```

Response: ok or error message

Get Machine Status

GET /status

Response (JSON):

```
{  
  "state": "Idle",  
  "position": "X:10.00 Y:20.00 Z:0.00"  
}
```

Example:

```
curl http://192.168.1.100/status
```

Camera Control APIs

Set Camera Parameters

GET /camera?[parameter]=[value]

Parameters:

- res: Resolution (3=QVGA, 5=VGA, 6=SVGA, 7=XGA, 8=HD, 10=UXGA)
- quality: JPEG quality (5-63)

- brightness: Brightness level (-2 to +2)
- contrast: Contrast level (-2 to +2)
- hmirror: Horizontal mirror (0=off, 1=on)
- vflip: Vertical flip (0=off, 1=on)

Examples:

```
# Set resolution to VGA
curl "http://192.168.1.100/camera?res=5"

# Set brightness to +1
curl "http://192.168.1.100/camera?brightness=1"

# Enable horizontal flip
curl "http://192.168.1.100/camera?hmirror=1"
```

Spatial Scan APIs

Start Scan

GET

/startscan?xstart=[value]&xend=[value]&ystart=[value]¥d=[value]&step=[value]&feed=[value]

Parameters:

- xstart, xend: X-axis scan boundaries (mm)
- ystart, yend: Y-axis scan boundaries (mm)
- step: Distance between points (mm)
- feed: Scan movement speed (mm/min, optional)

Example:

```
curl
"http://192.168.1.100/startscan?xstart=0&xend=100&ystart=0&yend=50&step=5&feed=150"
```

Response: OK or error message

Stop/Resume Scan

GET /stopscan

GET /resumescan

Examples:

```
# Pause scan
curl http://192.168.1.100/stopscan

# Resume paused scan
curl http://192.168.1.100/resumescan
```

Get Scan Status

GET /scanstatus

Response (JSON):

```
{
  "status": "Scanning...",
  "progress": 45
}
```

Example:

```
curl http://192.168.1.100/scanstatus
```

System Information APIs

Get IP Address

GET /ip

Example:

```
curl http://192.168.1.100/ip
```

Response: 192.168.1.100

Get Detailed System Info

GET /systeminfo

Response (JSON): Comprehensive system metrics including memory, storage, network, and hardware details.

Example:

```
curl http://192.168.1.100/systeminfo
```

File Management APIs

List Files in Directory

GET /files?dir=[path]

Parameters:

- dir: Directory path (optional, defaults to root)

Response (JSON):

```
{
  "parent": "/",
  "folders": ["/scan_1", "/scan_2"],
  "files": ["/scan_1/img_00001.jpg"]
}
```

Example:

```
curl "http://192.168.1.100/files?dir=/scan_1"
```

Download File

GET /download?file=[path]

Parameters:

- file: Full path to file (must start with /scan_ and end with .jpg)

Example:

```
curl -O "http://192.168.1.100/download?file=/scan_1/img_00001.jpg"
```

Delete Folder

GET /deletefolder?folder=[path]

Parameters:

- folder: Full folder path (must start with /scan_)

Example:

```
curl "http://192.168.1.100/deletefolder?folder=/scan_1"
```

Response: Folder deleted or error message

Position Management APIs

Save Current Position

GET /savepoint?name=[name]

Parameters:

- name: Position name (required)

Example:

```
curl "http://192.168.1.100/savepoint?name=Home_Position"
```

Get Saved Points

GET /getpoints

Response (JSON):

```
[  
  {  
    "name": "Home_Position",  
    "x": 0.000,  
    "y": 0.000,  
    "z": 0.000  
  }  
]
```

Example:

```
curl http://192.168.1.100/getpoints
```

Delete Saved Point

GET /deletepoint?name=[name]

Parameters:

- name: Position name to delete (required)

Example:

```
curl "http://192.168.1.100/deletepoint?name=Home_Position"
```

Move to Saved Point

GET /gotopoint?name=[name]

Parameters:

- name: Position name to move to (required)

Example:

```
curl "http://192.168.1.100/gotopoint?name=Home_Position"
```

Set Default Feed Rate

GET /setdefaultfeed?rate=[value]

Parameters:

- rate: Default feed rate in mm/min (required)

Example:

```
curl "http://192.168.1.100/setdefaultfeed?rate=200"
```