



Deltaroboter Benutzerhandbuch

Autoren:

Dennis Roth TET21B

Mayar Hanhon TET21B

Betreuer:

Peter Steinert

Wichtige Daten:.....3

Wichtige Hinweise5

API-Schnittstellenbeschreibung.....6

Grundkonfiguration6

1. Laden der GCode-Dateien	6
2. Löschen einer GCode-Datei.....	6
3. Speichern eines GCode-Programms	7
4. Update von Einstellungen	7
5. Motorstop	8
6. Homing.....	8
7. Manuelle Steuerung der Koordinaten	9
8. Manuelle Steuerung der Greiferstärke.....	9
9. Ausführen eines Pick-and-Place-Programms	10
G-Code Übersicht	11
Allgemeine Bewegungsbefehle	11
G0 - Schnelle Positionierung (Rapid Positioning)	11
G1 - Lineare Interpolation (Linear Interpolation)	11
Kreisförmige Bewegungsbefehle	11
G2 - Kreisinterpolation im Uhrzeigersinn (Clockwise Circular Interpolation)	11
G3 - Kreisinterpolation gegen den Uhrzeigersinn (Counterclockwise Circular Interpolation)	12

Spezialbefehle	12
G4 - Wartezeit (Dwell Time)	12
G17, G18, G19 - Auswahl der Arbeitsebene (Plane Selection)	12
G28 - Fahrt zur Homeposition (Return to Home)	12
M100, M200, M300, M400 - steuern verschiedene Greifertypen.	13
Router Konfiguration	14
Probleme und Lösungen	14

Wichtige Daten:

Wlan:

Router Name:	admin
Router Password:	secret
Netzwerk SSID:	Deltarobot SSID
Netzwerk Password:	ChangeMe
SSH:	
IP:	deltarobot
Nutzername:	pi
Password:	raspberry

Wichtige Hinweise

1. Die sechs Schalter an den Motortreibern sind entscheidend für die Funktionalität und sollten daher nicht willkürlich betätigt werden. Für mehr Informationen lesen Sie bitte die Datenblätter des Motors.
2. Wenn das Schaltnetzteil der Motoren ausgeschaltet wird, muss es sich erst entladen, was dazu führen kann, dass die Motoren für eine kurze Zeit weiterhin Strom erhalten und dementsprechend weiterlaufen. Der Schalter des Netzteils sollte daher nicht als Notausschalter verwendet werden.
3. Im G-Code-Editor werden fehlerhafte Angaben gelb hervorgehoben, während der Text rot eingefärbt wird. Der G-Code kann auch bei Vorliegen von Fehlern gespeichert und ausgeführt werden. Dies sollte jedoch nur geschehen, wenn Sie die technischen Implikationen vollständig nachvollziehen können.
4. Bitte achten Sie darauf, dass niemand den Roboter bedient, während Sie den Greifer wechseln. Auch wenn der Knopf (deactivate Magnet) in den Einstellungen gedrückt gehalten wird, können andere Personen, die Zugriff auf das Netzwerk haben, den Roboter steuern!
- 5.

API-Schnittstellenbeschreibung

Grundkonfiguration

Base URL: `http://<hostname>:3010`

Protokolle: HTTP/HTTPS

Content-Type: `application/json`

Endpunkte

1. Laden der GCode-Dateien

URL: `/loadGCodeFiles`

Methode: GET

Beschreibung: Lädt alle GCode-Dateien aus einem spezifizierten Verzeichnis.

Antworten:

200 OK: Erfolgreich geladene Dateien.

Beispiel: `[{"fileName": "example.gcode", "content": "GCode content here..."}]`

500 Internal Server Error: Fehler beim Lesen der Dateien.

Beispiel: `{"error": "Fehler beim Lesen der Datei."}`

2. Löschen einer GCode-Datei

URL: `/deleteGCode`

Methode: DELETE

Query-Parameter:

name: Name der zu löschenden GCode-Datei.

Beschreibung: Löscht eine spezifizierte GCode-Datei.

Antworten:

200 OK: Datei erfolgreich gelöscht.

Beispiel: {"message": "Datei 'example.gcode' erfolgreich gelöscht."}

400 Bad Request: Kein Dateiname angegeben.

Beispiel: {"error": "Dateiname nicht angegeben"}

404 Not Found: Datei nicht gefunden.

Beispiel: {"error": "Datei 'example.gcode' nicht gefunden"}

3. Speichern eines GCode-Programms

URL: /gcode

Methode: POST

Body:

name: Name der Datei.

content: Inhalt des GCode.

Beschreibung: Speichert ein neues GCode-Programm.

Antworten:

200 OK: Programm erfolgreich gespeichert.

Beispiel: {"message": "Programm 'example.gcode' erfolgreich gespeichert."}

400 Bad Request: Ungültige Daten im Request.

Beispiel: {"error": "Ungültige Daten"}

4. Update von Einstellungen

URL: /updateSettings

Methode: POST

Body:

Beispiel: {"gripperMode": "parallelGripper", "motorSpeed": 75}

Beschreibung: Aktualisiert die Einstellungen des Roboters über MQTT.

Antworten:

200 OK: Einstellungen erfolgreich aktualisiert und publiziert.

Beispiel: {"message": "Einstellungen erfolgreich aktualisiert und publiziert"}

400 Bad Request: Keine Einstellungen gesendet.

Beispiel: {"error": "Keine Einstellungen gesendet"}

500 Internal Server Error: Fehler beim Publizieren der Einstellungen.

Beispiel: {"error": "Fehler beim Publizieren der Einstellungen"}

5. Motorstop

URL: /motors/stop

Methode: POST

Body:

stop: Boolean-Wert, der angibt, ob die Motoren gestoppt werden sollen.

Beschreibung: Sendet ein Signal, um alle Motoren zu stoppen.

Antworten:

200 OK: Motorstopp signalisiert.

Beispiel: {"message": "Motorstopp signalisiert."}

400 Bad Request: Ungültige Daten, erwartet true.

Beispiel: {"error": "Ungültige Daten, erwartet true"}

6. Homing

URL: /homing

Methode: POST

Body:

active: Boolean-Wert, der das Homing aktiviert.

Beschreibung: Aktiviert die Homing-Funktion des Roboters.

Antworten:

200 OK: Homing signalisiert.

Beispiel: {"message": "Homing signalisiert."}

400 Bad Request: Ungültige Daten, erwartet einen Boolean.

Beispiel: {"error": "Ungültige Daten, erwartet einen Boolean"}

7. Manuelle Steuerung der Koordinaten

URL: /manual/control/coordinates

Methode: POST

Body:

coordinates: Array von 4 Elementen, die die Koordinaten darstellen.

Beschreibung: Aktualisiert die Koordinaten für den manuellen Modus.

Antworten:

200 OK: Koordinaten erfolgreich aktualisiert.

Beispiel: {"message": "Koordinaten erfolgreich aktualisiert."}

`400 Bad Request`: Ungültige Koordinaten, erwartet ein Array von 4 Elementen.

Beispiel: {"error": "Ungültige Koordinaten, erwartet ein Array von 4 Elementen"}

8. Manuelle Steuerung der Greiferstärke

URL: /manual/control/gripper

Methode: POST

Body:

gripper: Numerischer Wert, der die Stärke des Greifers angibt.

Beschreibung: Aktualisiert die Greiferstärke für den manuellen Modus.

Antworten:

200 OK: Greifersteuerung erfolgreich aktualisiert.

Beispiel: {"message": "Greifersteuerung erfolgreich aktualisiert."}

400 Bad Request: Ungültige Greiferstärke, erwartet eine Zahl.

Beispiel: {"error": "Ungültige Greiferstärke, erwartet eine Zahl"}

9. Ausführen eines Pick-and-Place-Programms

URL: /pickandplace/program

Methode: POST

Body:

program: Zeichenkette, die den Namen des auszuführenden Programms angibt.

Beschreibung: Übermittelt den Namen eines auszuführenden Programms an den Roboter.

Antworten:

200 OK: Programm erfolgreich übermittelt.

Beispiel: {"message": "Programm erfolgreich übermittelt."}

400 Bad Request: Ungültiger Programmwert, erwartet eine Zeichenkette.

Beispiel: {"error": "Ungültiger Programmwert, erwartet eine Zeichenkette"}

WebSockets

Verbindung aufbauen: WebSockets verbinden sich auf demselben Port, auf dem der Server lauscht. Der WebSocket-Client empfängt

Zustandsaktualisierungen des Roboters via MQTT.

Events:

connection: Ein Client verbindet sich.

message: Nachrichten von MQTT, z.B. robot/state, werden an verbundene Clients gesendet.

close: Ein Client trennt die Verbindung.

error: Fehlerbehandlung für WebSocket-Verbindungen.

G-Code Übersicht

Allgemeine Bewegungsbefehle

G0 - Schnelle Positionierung (Rapid Positioning)

Bewegt das Werkzeug schnell zu einer angegebenen Position ohne Bearbeitung.

Beispiel: G0 X0 Y0 Z-280 A20 F80 – Schnelle Bewegung zur Koordinatenposition mit Greiferdrehung. (A20 -> Greiferdrehung von 20°; F80 -> 80% der zulässigen Geschwindigkeit)

G1 - Lineare Interpolation (Linear Interpolation)

Bewegt das Werkzeug linear zur angegebenen Position und führt dabei eine Bearbeitung aus.

Beispiel: G1 X0 Y0 Z-280 A20 F80 – Lineare Bewegung zur Koordinatenposition mit gleichmäßiger Greiferdrehung.

Kreisförmige Bewegungsbefehle

G2 - Kreisinterpolation im Uhrzeigersinn (Clockwise Circular Interpolation)

Bewegt das Werkzeug auf einem kreisförmigen Pfad im Uhrzeigersinn zur Zielposition.

Beispiel: G2 X20 Y0 I10 J0 A10 F20

X20 Y0: Diese Koordinaten definieren die Zielposition des Werkzeugs auf der XY-Ebene. In diesem Fall soll das Werkzeug zur X-Position 20 und zur Y-Position 0 bewegt werden.

I10 J0: Diese Parameter sind die Koordinaten des Kreismittelpunkts relativ zur aktuellen Position des Werkzeugs. I10 bedeutet, dass der Mittelpunkt des Kreises 10 Einheiten in X-Richtung von der aktuellen Position entfernt ist. J0 bedeutet, dass der Mittelpunkt des Kreises auf derselben Y-Position wie die aktuelle Position des Werkzeugs liegt.

G3 - Kreisinterpolation gegen den Uhrzeigersinn (Counterclockwise Circular Interpolation)

Führt eine kreisförmige Bewegung gegen den Uhrzeigersinn zur Zielposition aus.

Beispiel: G3 X20 Y0 I10 J0 A10 F20 – Gegenläufiger Kreisbogen zur gleichen Position mit Drehung.

Spezialbefehle

G4 - Wartezeit (Dwell Time)

Pausiert die Maschinenaktivität für eine festgelegte Zeit.

Beispiel: G4 P500 – Maschine wartet 500 Millisekunden.

G17, G18, G19 - Auswahl der Arbeitsebene (Plane Selection)

G17 – Auswahl der XY-Ebene.

G18 – Auswahl der YZ-Ebene.

G19 – Auswahl der ZX-Ebene.

G28 - Fahrt zur Homeposition (Return to Home)

Fahrt das Werkzeug in die anfängliche Ausgangsposition zurück.

Beispiel: G28 F80 – Rückkehr zur Homeposition mit spezifischer Geschwindigkeit.

Greifersteuerungsbefehle

M100, M200, M300, M400 - steuern verschiedene Greifertypen.

Parallelgreifer (M100): Einstellbar von S0 (geschlossen) bis S100 (geöffnet).

Magnetgreifer (M300): S0 schaltet aus, S1 schaltet ein.

Vakuum- und Compliant-Greifer (M400 und M200): S-1 aktiviert das Vakuum, S0 schaltet aus, S1 aktiviert den Druck.

Beispiel: M100 S50 – Aktiviert den Parallelgreifer und setzt ihn auf 50% der Kapazität.

Router Konfiguration

1. Aktualisierung des Betriebssystems:

SUDO APT-GET UPDATE

SUDO APT-GET FULL-UPGRADE

SUDO REBOOT

2. Installation von RaspAP:

CURL -SL HTTPS://INSTALL.RASPAP.COM | BASH

Die Konfiguration des Access Points erfolgte über die Web-Oberfläche von RaspAP, wo SSID, Sicherheitseinstellungen und weitere relevante Parameter angepasst wurden. (<https://github.com/RaspAP/raspap-webgui>)

Probleme und Lösungen

Während des Betriebs traten Probleme auf, bei denen das SSID des Access Points nur kurzzeitig sichtbar war und keine stabile Verbindung ermöglichte. Es stellte sich heraus, dass der Raspberry Pi versuchte, gleichzeitig eine Verbindung zu einem vorhandenen WLAN-Netzwerk herzustellen und den Access Point über dasselbe WLAN-Interface zu betreiben. Eine detaillierte Untersuchung wurde durchgeführt, wobei folgende diagnostische Befehle verwendet wurden:

iwconfig zur Überprüfung der drahtlosen Konfiguration

nmcli con show zur Anzeige aktiver Netzwerkverbindungen

Es wurde festgestellt, dass der Raspberry Pi (wlan0) automatisch mit einem vorkonfigurierten Netzwerk ("preconfigured") verbunden war. Die Lösung des Problems wurde durch das Löschen dieser Verbindung erreicht:

```
sudo nmcli con delete id "preconfigured"
```

Nach dieser Anpassung funktionierte der Access Point ohne Unterbrechungen und ermöglichte eine stabile und dauerhafte Verbindung zur Webanwendung des Delta-Roboters.