Dobot Magician: Kurzzusammenfassung

Von der einfachen Queue bis zur Steuerung per Tastatur, COM und WLAN

Ausgangspunkt

Aus einzelnen Aufrufen von SetPTPCmd() wurde eine leicht lesbare Befehlsliste entwickelt. Queue-Indizes, Beschreibungen und Pausen werden automatisch verwaltet.

```
befehle = [
  ('fahre_zu', 180, 160, 50, 0, 'Fahre zu Punkt 1'),
  ('sauger_ein', 'Sauger einschalten', 1000),
  ('sauger_status', 'Status anzeigen', 0),
]
```

Wichtige Ergebnisse

Thema	Ergebnis
Befehlskette	Befehle werden in einer übersichtlichen Liste beschrieben und automatisch in die Dobot-Queue eingetragen.
Pausen	Standardpause oder individuelle Pause je Befehl. Die eigene Schnittstelle arbeitet in Millisekunden; SetWAITCmd() erhält Sekunden.
Anzeige	Der nächste Befehl wird vor seiner Ausführung in Thonny angezeigt.
Tastatur	p = Pause, w = Weiter, h = Halt, ? = Status.
Pause und Halt	Eine Pause kann fortgesetzt werden. Ein Halt beendet die Aufgabe endgültig und erfordert anschließend Kontrolle und Neueinrichtung.
Sauger	sauger_ein, sauger_aus und sauger_status wurden in die Befehlskette aufgenommen.

Aktuelle Steuerarchitektur

Tastatur, ESP32 über COM und ESP32 über WLAN arbeiten als getrennte Eingabethreads. Alle schreiben ausschließlich Steuerbefehle in eine gemeinsame Queue. Nur der Hauptthread greift auf die Dobot-API zu.

```
Tastatur-Thread ----\
ESP32-COM-Thread ----+--> Steuerbefehls-Queue --> Dobot-Hauptthread
ESP32-WLAN-Thread ---/
```

Letzte wichtige Dateien

```
befehlskette_v3_3.py
befehlskette_beispiel_v3.4.py
esp32_seriell_v1_0.py
esp32_wlan_v1_0.py
esp32_com_test.py
esp32_wlan_test.py
```

Stand: Die Python-Dateien wurden syntaktisch geprüft und teilweise simuliert. Die praktischen Tests mit Dobot und ESP32 stehen noch aus.