

Dobot Magician: Problem mit der Pause

Analyse des scheinbaren Stillstands in einer mit DobotDIIType aufgebauten Befehlskette

1. Ausgangssituation

Mehrere Bewegungsbefehle werden nacheinander in die Queue des Dobot eingetragen. Nach jeder Bewegung soll optional eine Pause folgen. Die Pausen werden mit `SetWAITCmd()` ebenfalls als Queue-Befehle eingetragen.

Beispiel für die verständliche Befehlsliste:

```
befehle = [  
    ('fahre_zu', 180, 160, 50, 0, 'Fahre zu Punkt 1'),  
    ('fahre_zu', 240, 140, 70, 0, 'Fahre zu Punkt 2', 2500),  
    ('fahre_zu', 200, 180, 50, 0, 'Fahre zu Punkt 3', 0),  
]
```

Die Zahl 2500 sollte dabei eine Pause von 2500 Millisekunden, also 2,5 Sekunden, bedeuten.

2. Beobachtetes Verhalten

Die Queue wurde korrekt aufgebaut. Der Dobot erreichte den ersten Bewegungsbefehl und gab dessen Queue-Index aus. Danach bewegte er sich nicht weiter. Es erschien keine Alarmanzeige am Dobot-Fuß und auch kein unmittelbarer Python-Fehler.

Queue-Index	Bedeutung
27	Erste Bewegung
28	Pause nach der ersten Bewegung
29	Zweite Bewegung
30	Pause nach der zweiten Bewegung
31	Dritte Bewegung

Der Queue-Index blieb beim ersten Bewegungsbefehl beziehungsweise unmittelbar danach stehen. Dadurch entstand zunächst der Eindruck, die Bewegung oder die Queue-Auswertung sei fehlerhaft.

3. Ursache

Die Ursache lag in der Einheit des Parameters von `SetWAITCmd()`. In der verwendeten Version von `DobotD11Type.py` erwartet die Python-Funktion die Wartezeit in **Sekunden**. Der Wrapper rechnet den übergebenen Wert intern selbst in Millisekunden um.

Die zunächst verwendete Zeile war deshalb falsch:

```
warteindex = dType.SetWAITCmd(api, pause_ms, isQueued=1)[0]
```

Bei `pause_ms = 1000` wurde nicht eine Sekunde, sondern eine Wartezeit von 1000 Sekunden eingetragen.

Übergebener Wert	Tatsächliche Wartezeit
1000	1000 Sekunden = 16 Minuten 40 Sekunden
2500	2500 Sekunden = 41 Minuten 40 Sekunden

4. Korrektur

Die verständliche Schnittstelle des eigenen Moduls soll weiterhin Millisekunden verwenden. Deshalb wird der Wert beim Aufruf von `SetWAITCmd()` durch 1000 geteilt:

```
warteindex = dType.SetWAITCmd(
    api,
    pause_ms / 1000.0,
    isQueued=1,
)[0]
```

Damit bleibt die Eingabe im eigenen Programm einfach und eindeutig:

```
STANDARD_PAUSE_MS = 1000    # 1 Sekunde
pause_ms = 2500             # 2,5 Sekunden
pause_ms = 0                # keine Pause
```

Ergebnis: Die Befehlsliste arbeitet weiterhin mit Millisekunden. Nur beim Übergang zur Dobot-API erfolgt die notwendige Umrechnung in Sekunden.

5. Merksatz

Die Einheit eines Parameters darf nicht allein aus dem Variablennamen abgeleitet werden. Entscheidend ist die konkrete Implementierung der verwendeten `DobotD11Type.py`. Bei `SetWAITCmd()` werden in dieser Version Sekunden übergeben, obwohl die interne Dobot-Struktur mit Millisekunden arbeitet.

Korrigierte Modulversion: **befehlskette.py Version 2.2**