

Projektvorstellung

→ Stationsübersicht



Quickstart-Anleitung für den Mitsubishi RV-2AJ

1. Inbetriebnahme des Roboterarms

1. Ciro Studio am PC starten.
2. Power-Schalter am Controller auf ON schalten.
3. Power-Schalter an der I/O-Box auf ON schalten.
4. Schlüsselschalter am Controller auf AUTO (Ext.) stellen.

2. Neues Projekt in CIROS Studio erstellen

1. In Ciro Studio:
 - Datei → Neu → Mitsubishi Projekt
 - Projektname & Programmname vergeben (Hier eigener Name) → Weiter
 - Roboterauswahl: RV-2AJ
 - E/A-Karte: 1, Hände: 1, Zusatzachsen: Keine
 - Programmiersprache: MELFA BASIC IV

3. Verbindung konfigurieren

1. Im RCI-Explorer (links unten):
 - Rechtsklick auf „Verbindung“ → Eigenschaften
 - COM-Port einstellen (Baud 9600, Parität Gerade etc. beibehalten)
 - In unserem Fall COM-Port 1 !
2. Erneut Rechtsklick auf „Verbindung“ → „aufbauen“
 - „Zustand“ sollte „Verbindung zum Roboter hergestellt“ anzeigen
 - Bei Warnmeldung → Roboter ggf. referenzieren (Achsen neu anlernen)

4. Programm erstellen (ohne Code)

1. Im Programmierfenster (oben rechts) die gewünschten Befehle eingeben (z. B. MOV P1, MOV P2 etc.).
2. Speichern, damit das Projekt gesichert ist.

5. Positionen anlernen (Teachen)

1. Oben rechts auf die zwei Pfeile klicken → „Online Teach-In“
2. Koordinatensystem wählen (z. B. „Welt“)
3. Geschwindigkeit & Inkrementalbewegung einstellen (z. B. 50 mm/s, 25 mm / 5°)
4. Achsen über X, Y, Z, A, B, C verfahren, bis die gewünschte Position erreicht ist
5. „Position übernehmen“ anklicken und als P1, P2, ... speichern

6. Programm und Positionen auf den Roboter übertragen

1. Im RCI-Explorer → Arbeitsplatz → Programme
 - Rechtsklick auf das Programm (z. B. .MB4) → „Download“
 - Rechtsklick auf die Positionsdatei (z. B. .POS) → „Download“
2. Im RCI-Explorer → RV-2AJ → Programme
 - Rechtsklick auf das gewünschte Programm → „Ausführen“ (z. B. Cycle-/Rep-/Debug-Modus)

Allgemeiner Bewegungsbefehl:

Im Laufe des Programms muss stets eine Fortlaufende Nummerierung verwendet werden. Siehe die pinken Zahlen vor jeder Zeile.

```
10 MOV P1 ' Fahre zu Position P1
```

Koordinatenpaar zum Verschieben einer Position anlegen:

```
20 DEF POS VECNEU
```

```
30 DEF POS AUXPOS
```

Wartebefehl DLY:

```
10 DLY 5
```

Lässt den Roboter an aktueller Position 5 Sekunden warten. Die Zahl kann in beliebigen maximal hundertstel Inkrementen geändert werden. Hierbei ist die Zahl durch ein „.“ Zu trennen, z.B.

```
10 DLY 5.05
```

VECNEU mit Koordinaten versehen um zu verschieben:

```
30 VECNEU = (+0.00,+26.40,+0.00,+0.00,+0.00)
```

Dieser Vektor verschiebt den Roboterarm von der aktuellen Position um 26.4mm in Y-Richtung. Um eine Verschiebung in Negativer Y-Richtung zu erzeugen, beim Errechnen der anzufahrenden Koordinaten ein „520 AUXPOS = P1 – VECXNEU“

verwenden!

Koordinatenpaar nach Verschiebung anfahren über AUXPOS:

Ebenfalls ist es empfehlenswert die Verfahr Bewegungen über eine AUXPOS durchzuführen um keine Veränderung der eigentlichen Positionskordinaten vorzunehmen.

```
520 AUXPOS = P1 + VECNEU ' Verschiebe um VECNEU
```

```
530 MOV AUXPOS
```

Nach erstmaligem Deklarieren von AUXPOS kann ab dem Moment über

```
520 AUXPOS = AUXPOS + VECNEU ' Verschiebe um VECNEU
```

```
530 MOV AUXPOS
```

Verfahren werden.

Siehe fertiges Programm mit Startpunkt über Weltkoordinaten im Online-Teach-In Tool angelegt und dann über 2 geänderte Positionen verfahren.

```
10 DEF POS AUXPOS
```

```
20 DEF POS VECNEU
```

```
30 DEF POS P1
```

```
40 VECNEU = (+0.00,+20.00,+0.00,+0.00,+0.00)
```

```
41 MOV P1
```

```
50 AUXPOS = P1 + VECNEU
```

```
60 MOV AUXPOS
```

```
61 DLY 2
```

```
70 AUXPOS = AUXPOS - VECNEU
```

```
80 MOV AUXPOS
```

```
90 DLY 2
```