



*Anforderungsspezifikation  
der Simulationskomponente  
des ARO1 (in Delphi)*

**Auto:** © by Thomas Gattinger (Carl-Benz-Schule Koblenz)

**Version:** 13.12.2009

# Inhaltsverzeichnis

1) Zielbestimmungen.....	3
2) Einsatz der Komponente.....	3
3) Funktionalität der Komponente.....	3
a) Elemente (Definitionsansicht).....	3
b) Schnittstellen.....	8
c) Visualisierung.....	10
4) Produktleistungen.....	10
5) Qualitätsanforderungen.....	10

## 1) Zielbestimmungen

Ziel dieses Projektes ist es eine abgeschlossene Komponente für Delphi zu entwickeln, die den ARO1 Roboterarm simuliert. Die Komponente kann dann mittels Drag&Drop von jedem Formular genutzt werden. Sie wird eine grafische Oberfläche haben, auf der der Roboterarm gezeichnet wird. Die Stellung des Armes hängt von den Daten ab, die über die Schnittstelle eingegeben werden können.

Optionales Ziel ist es die Visualisierung 3D-fähig zu programmieren, sodass der Arm rotieren kann und so die Stellungen der einzelnen Elemente besser zu erkennen sind. Für diese optionalen Elemente der Simulation sind auch einige optionale Schnittstellen genannt, die im zweiten Schritt realisiert werden.

## 2) Einsatz der Komponente

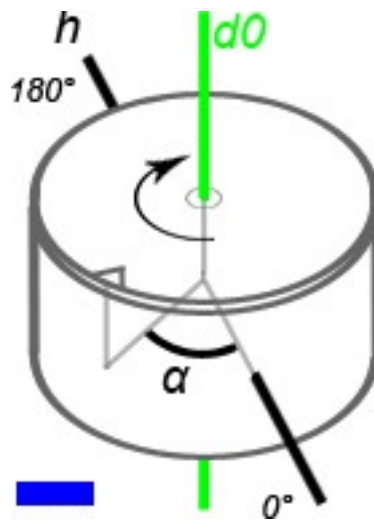
Die Komponente kann im ARO1 Konfigurationsprogramm als Vorschau dienen, sodass die Stellung des realen Roboterarms vorher im Konfigurationsprogramm angezeigt wird. Dies erleichtert die Bedienung, da beim Benutzen der Schieberegler eine „in time“ Vorschau erzeugt wird.

Ferner kann die Komponente zu Demonstrationszwecken im Unterricht verwendet werden bzw. zur Erläuterung des Verhaltens des Roboterarms bei verschiedenen Einstellungen.

## 3) Funktionalität der Komponente

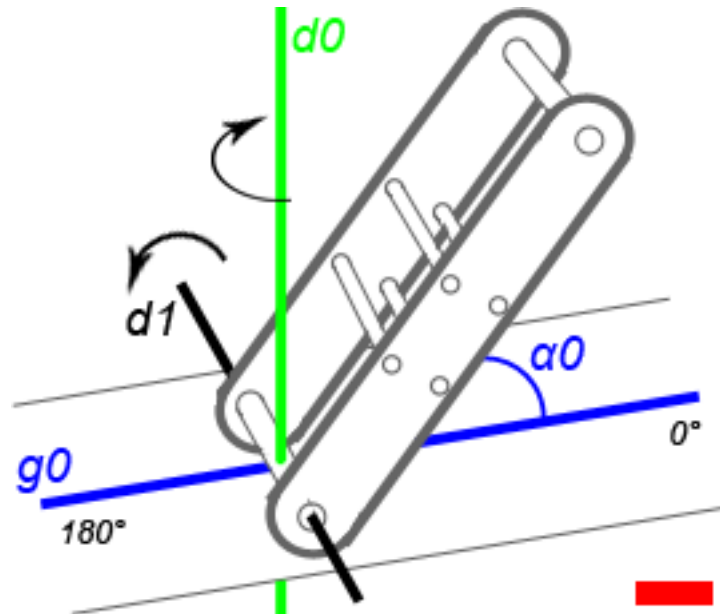
### a) Elemente (Definitionsansicht)

Element 0



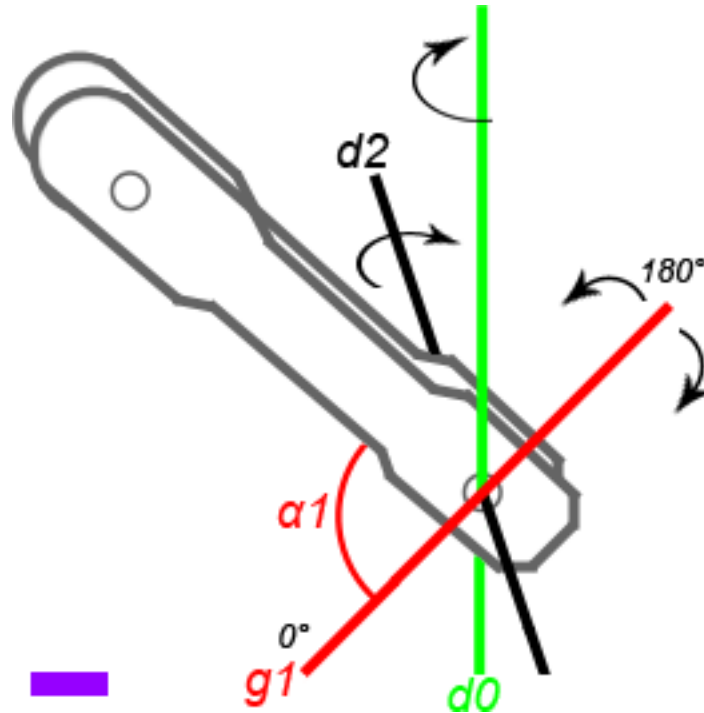
- **Rotationsachse  $d0$** : Um diese Achse dreht sich das Element horizontal.
- **Gerade  $h$** : Minimale bzw. maximale Drehposition bezogen auf die Einkerbung.
- **Winkel  $\alpha$** : Winkel um den sich das Element zwischen min. Winkel und max. Winkel dreht.
- **Blaue Markierung**: Die Markierung wird zur Verdeutlichung der Abhängigkeit vom nächsten Element zu diesem verwendet. Je nach Lage dieses Elements verändert sich auch die entsprechende Gerade im nächsten Element und die Winkel müssen angeglichen werden.

## Element 1



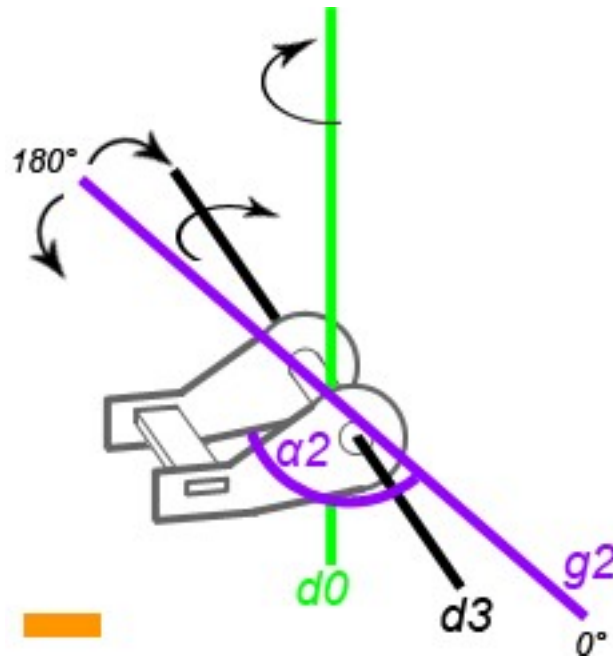
- **Rotationsachse  $d0$** : Um diese Achse dreht sich das Element horizontal abhängig von Element 0.
- **Rotationsachse  $d1$** : Um diese Achse dreht sich das Element vertikal.
- **Gerade  $g0$** : Basisachse auf den sich die Winkelangabe bezieht (abhängig von Element 0)
- **Winkel  $\alpha0$** : Winkel um den sich das Element zwischen min. Winkel und max. Winkel dreht.
- **Rote Markierung**: Die Markierung wird zur Verdeutlichung der Abhängigkeit vom nächsten Element zu diesem verwendet. Je nach Lage dieses Elements verändert sich auch die entsprechende Gerade im nächsten Element und die Winkel müssen angeglichen werden.

## Element 2



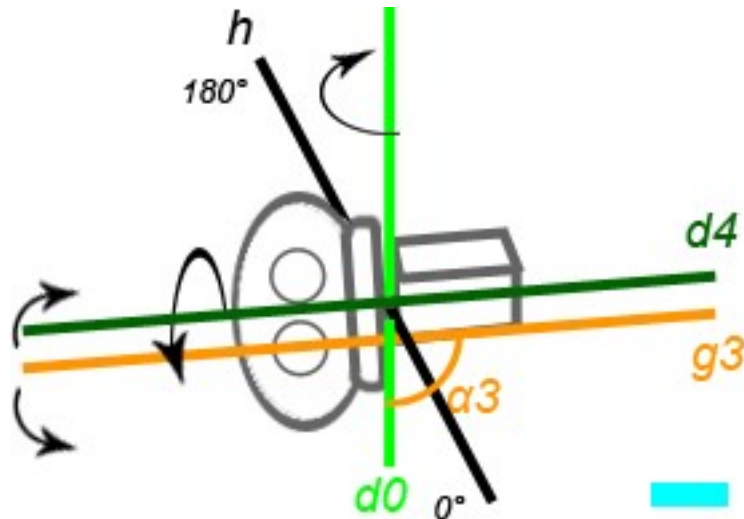
- **Rotationsachse  $d0$** : Um diese Achse dreht sich das Element horizontal abhängig von Element 0.
- **Rotationsachse  $d2$** : Um diese Achse dreht sich das Element vertikal.
- **Gerade  $g1$** : Basisachse auf den sich die Winkelangabe bezieht. Winkel dieser Achse hängt von Element 1 ab.
- **Winkel  $\alpha1$** : Winkel um den sich das Element zwischen min. Winkel und max. Winkel dreht.
- **Lila Markierung**: Die Markierung wird zur Verdeutlichung der Abhängigkeit vom nächsten Element zu diesem verwendet. Je nach Lage dieses Elements verändert sich auch die entsprechende Gerade im nächsten Element und die Winkel müssen angeglichen werden.

### Element 3



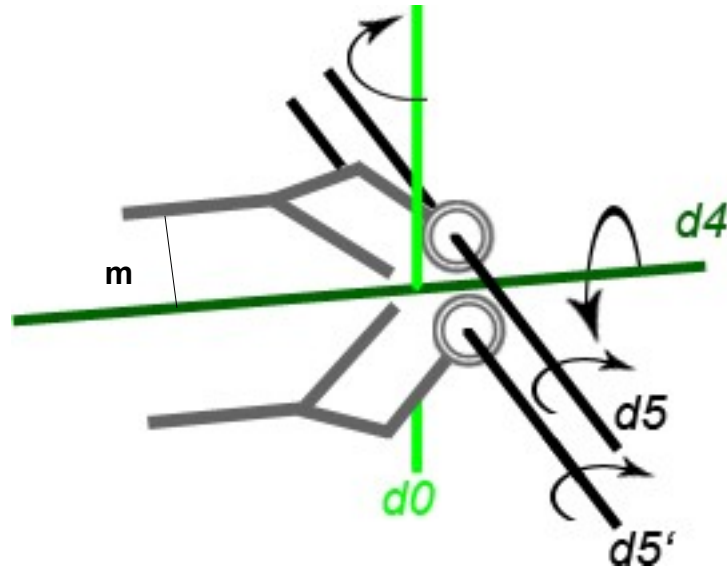
- **Rotationsachse  $d0$** : Um diese Achse dreht sich das Element horizontal abhängig von Element 0.
- **Rotationsachse  $d3$** : Um diese Achse dreht sich das Element vertikal.
- **Gerade  $g2$** : Basisachse auf den sich die Winkelangabe bezieht. Winkel dieser Achse hängt von Element 2 ab.
- **Winkel  $\alpha2$** : Winkel um den sich das Element zwischen min. Winkel und max. Winkel dreht.
- **Orangene Markierung**: Die Markierung wird zur Verdeutlichung der Abhängigkeit vom nächsten Element zu diesem verwendet. Je nach Lage dieses Elements verändert sich auch die entsprechende Gerade im nächsten Element und die Winkel müssen angeglichen werden.

## Element 4



- **Rotationsachse  $d0$** : Um diese Achse dreht sich das Element horizontal abhängig von Element 0.
- **Rotationsachse  $d4$** : Um diese Achse dreht sich das Element vertikal.
- **Gerade  $g3$** : Basisachse auf den sich die Winkelangabe bezieht. Winkel dieser Achse hängt von Element 3 ab.
- **Winkel  $\alpha3$** : Winkel um den sich das Element zwischen min. Winkel und max. Winkel dreht.
- **Orangene Markierung**: Die Markierung wird zur Verdeutlichung der Abhängigkeit vom nächsten Element zu diesem verwendet. Je nach Lage dieses Elements verändert sich auch die entsprechende Gerade im nächsten Element und die Winkel müssen angeglichen werden.

## Element 5



- **Rotationsachse  $d0$** : Um diese Achse dreht sich das Element horizontal abhängig von Element 0.
- **Rotationsachse  $d4$** : Um diese Achse dreht sich das Element vertikal.
- **Rotationsachse  $d5$** : Um diese Achse dreht sich das Element zwischen Minimaler und Maximaler Grenze (Wird in 100% angegeben)
- **Rotationsachse  $d5'$** : Um diese Achse dreht sich das Element. (zu  $d5$  entgegengesetzt)
- **Strecke  $m$** : Gibt an welche Strecke der Greifer maximal zurücklegen darf. (Einteilung von 0 bis 100) Greifer bewegt sich entsprechend der Drehung von  $d5$
- **$d5'$  und Greifer'**: Verhalten sich spiegelverkehrt zu  $d5$  und Greifer

## b) Schnittstellen

### Grundlegende Schnittstellen

Kompletter Arm:

- Geschwindigkeitsangabe in der die neue Roboterarmposition sich bewegen soll
- rote Kontur um das Element dessen Winkel sich verändert.

Element 0 (Servo 0):

- Minimaler Winkel:
- Maximaler Winkel:
- Position (über Schieberegler) von 0 bis 100

Element 1 (Servo 1+2):

- Minimaler Winkel:



- Maximaler Winkel:
- Position (über Schieberegler) von 0 bis 100

Element 2 (Servo 3):

- Minimaler Winkel:
- Maximaler Winkel:
- Position (über Schieberegler) von 0 bis 100

Element 3 (Servo 4):

- Minimaler Winkel:
- Maximaler Winkel:
- Position (über Schieberegler) von 0 bis 100

Element 4 (Servo 5):

- Minimaler Winkel:
- Maximaler Winkel:
- Position (über Schieberegler) von 0 bis 100

Element 5 (Servo 6):

- Minimaler Winkel:
- Maximaler Winkel:
- Position (über Schieberegler) von 0 bis 100

## **Optionale Schnittstellen**

Kompletter Arm:

- automatisches Rotieren (ja/nein)

Element 0 (Servo 0):

- Füllfarben (Oben, Seiten, Markierung)

Element 1 (Servo 1+2):

- Füllfarben (Oben, Seiten, Markierung)

Element 2 (Servo 3):

- Füllfarben (Oben, Seiten, Markierung)

Element 3 (Servo 4):

- Füllfarben (Oben, Seiten, Markierung)

Element 4 (Servo 5):

- Füllfarben (Oben, Seiten, Markierung)

Element 5 (Servo 6):

- Füllfarben (Oben, Seiten, Markierung)

### **c) Visualisierung**

Bei der 3D Darstellung wird die OpenGL Technologie verwendet.

Alle Elemente werden im 3D Koordinatensystem verwaltet.

Jedes Element besitzt ein eigenes Koordinatensystem in dem das Objekt gedreht bzw. verschoben werden kann.

## **4) Produktleistungen**

*Simulation:* Die Komponente soll die Funktionalität des Roboterarms originalgetreu nachbilden. Dabei gelten die gleichen Beschränkungen, die auch die Hardware einschränkt.

*Visualisierung:* Die visuelle Darstellung wird in 3D mit Rotationsmöglichkeit implementiert.

*Fehlerbehandlung:* Fehlerhafte Eingaben vom Benutzer werden automatisch korrigiert. Unterschreitet der Einstellungswert den Minimalgrenzwert so wird immer der Minimalgrenzwert verwendet. Überschreitet der Einstellungswert den Maximalgrenzwert, so wird immer der Maximalgrenzwert herangezogen. Zusätzlich wird ein roter Fehlertext angezeigt, der auf die automatische Korrektur hinweist.

## **5) Qualitätsanforderungen**

Auf Robustheit und Zuverlässigkeit wird größter Wert gelegt. Um dies zu erreichen wird defensive Programmierung in allen Prozeduren benutzt. Dadurch wird sichergestellt, dass in der Auswertungslogik der Prozeduren keine ungültigen Werte verwendet werden. Nach der vollständigen Fertigstellung der Pflicht- und Optionalelementen liegt der Schwerpunkt auch in der hohen Konfigurationsmöglichkeit und dem Detailreichtum, um den original Roboterarm möglichst detailgetreu zu simulieren.