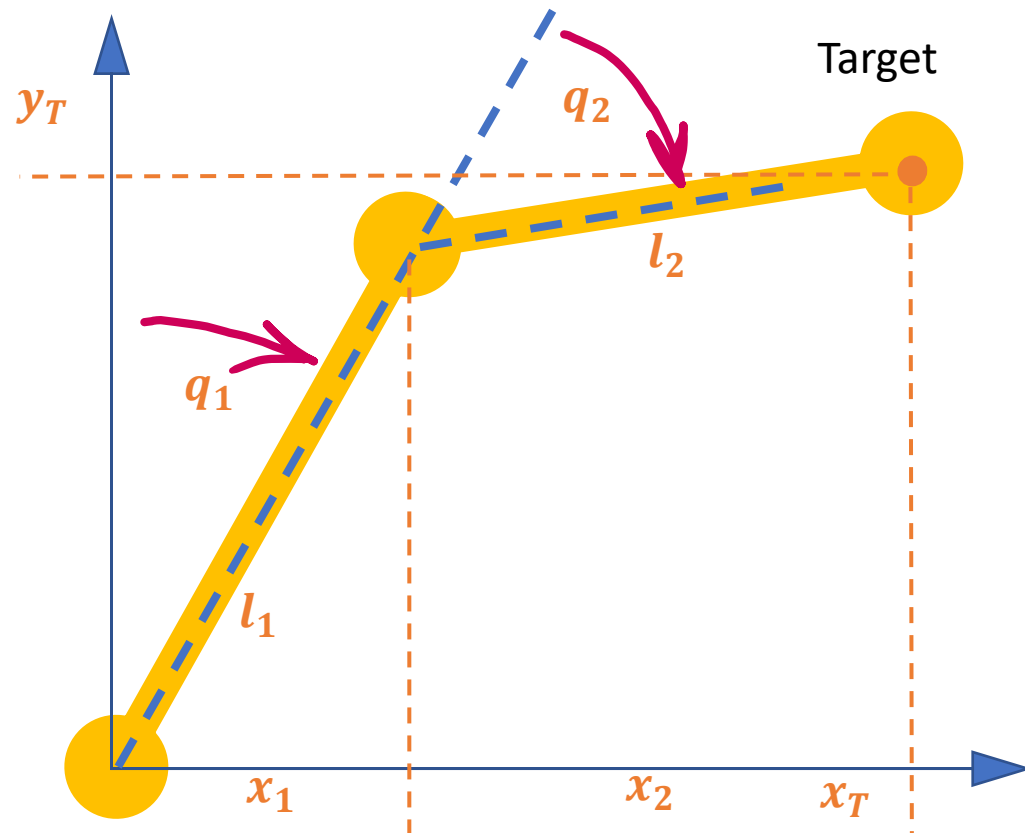


Einmal 80s und zurück

Roboterprogrammierung gestern und heute

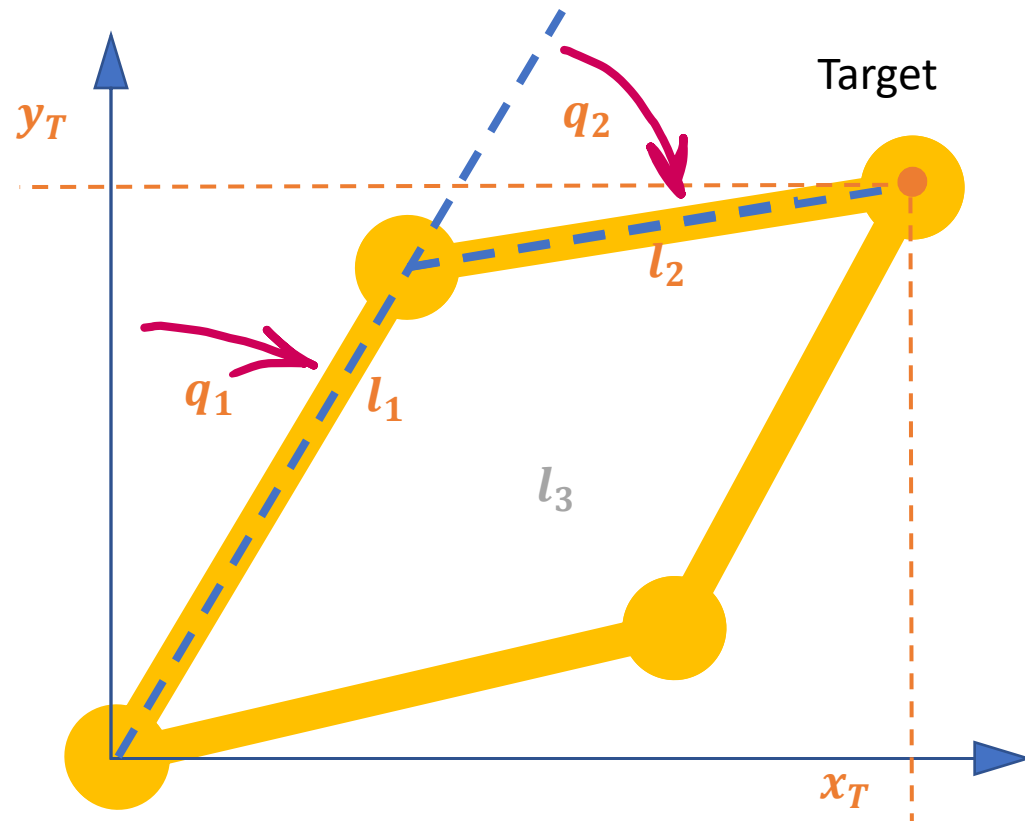
Forward Kinematics



$$q_1, q_2 \rightarrow x_T, y_T$$

$$\begin{aligned}x_t &= x_1 + x_2 \\x_t &= l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) \\y_t &= l_1 \cos(q_1) + l_2 \cos(q_1 + q_2)\end{aligned}$$

Inverse Kinematics



$$x_T, y_T \rightarrow q_1, q_2$$

Fahrbefehle



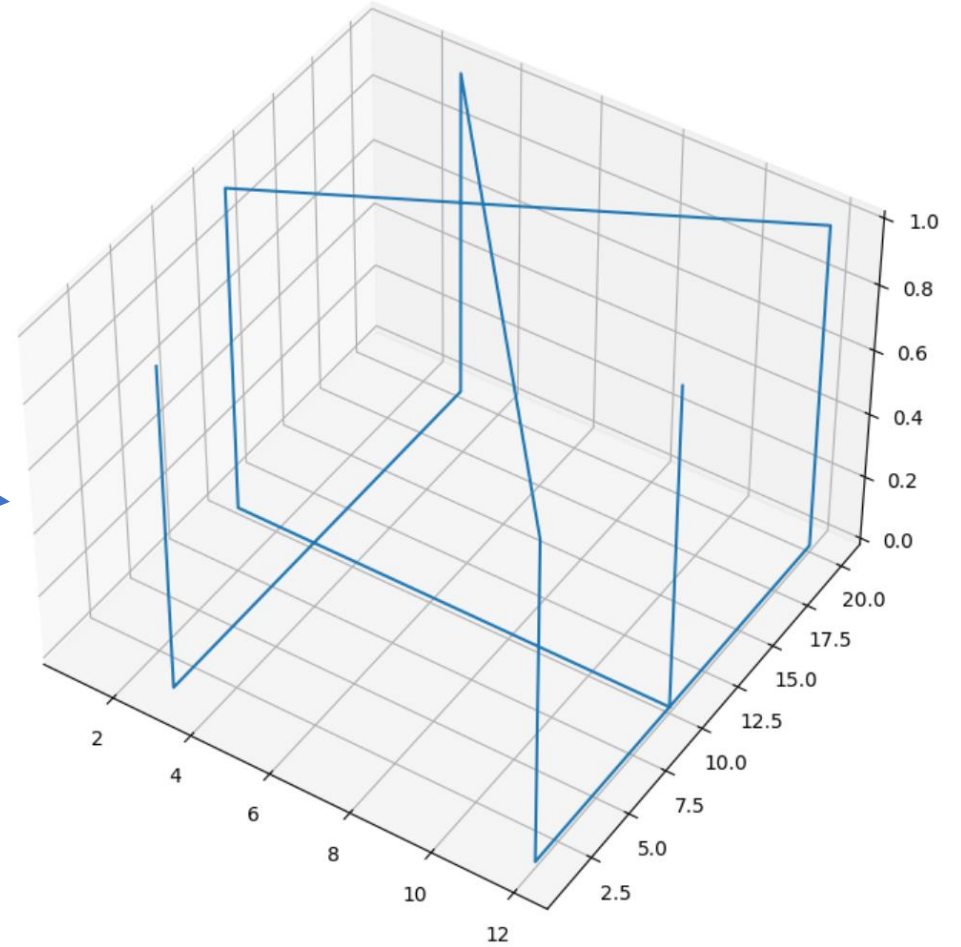
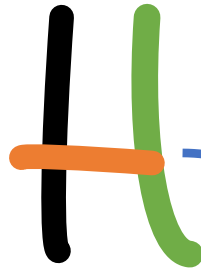
Case Study Weihnachtsmann

- 2,4 Mrd. Kinder auf der Welt
 - Beantwortet 1.000.000 Wunschzettel/Jahr
 - Traditionell wird eine handschriftliche Antwort gefordert
 - Durchschnittliche Schreibzeit pro Antwort 5 min (Gesamt: ~ 3.472 Tage)
- Lösung: Automatisierung!
- Projektstart: Oktober 1985

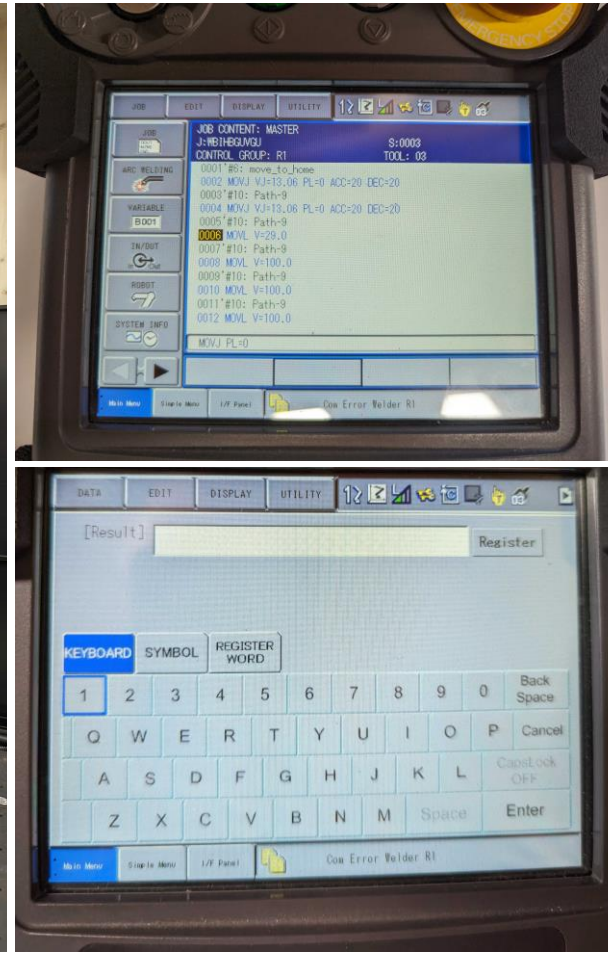
Handschrift zu Roboterpfad

Ho, ho, ho

H



YASKAWA Programmierung



YASKAWA Inform

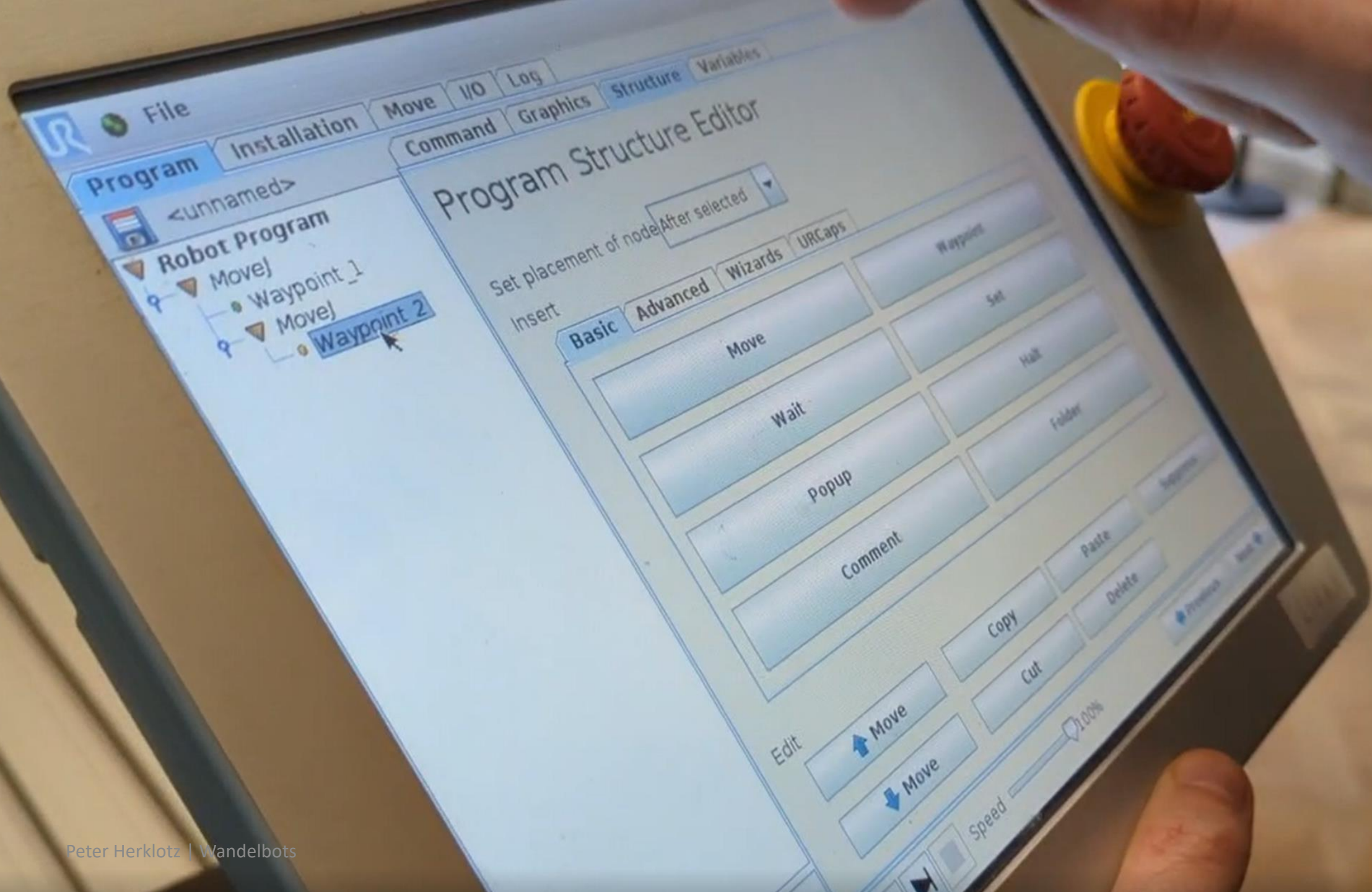


```

● ● ●
/JOB
//NAME JUG
//POS
///NPOS 11,0,0,0,0,0
///TOOL 3
///POSTYPE PULSE
///PULSE
C00000=-17025,-36269,-31638,1825,-95398,7116
C00001=-13340,2153,-3022,1414,-93340,5833
C00002=-13340,7308,15638,1453,-102172,5935
C00003=-5533,-30667,-13638,570,-103767,3225
C00004=-5533,-35705,-28915,555,-96955,3194
C00005=-4272,5732,2387,412,-94934,2754
C00006=-4272,13229,25982,429,-105601,2790
C00007=-23107,-11922,4890,2580,-105194,9395
C00008=-23107,-19046,-17018,2483,-95382,9194
C00009=-1761,-26178,-22655,132,-95929,1898
C00010=-1761,-19944,-3312,137,-104629,1907
//INST
///DATE 2022/09/19 10:37
///ATTR SC,RW
///GROUP1 RB1
NOP
MOVJ C00000 VJ=5.00
MOVL C00001 V=750.0
MOVL C00002 V=750.0
MOVL C00003 V=750.0
MOVL C00004 V=750.0
MOVL C00005 V=750.0
MOVL C00006 V=750.0
MOVL C00007 V=750.0
MOVL C00008 V=750.0
MOVL C00009 V=750.0
MOVL C00010 V=750.0
END
```


„Ein Cobot ist eine Vorrichtung und ein Verfahren zur direkten physischen Interaktion zwischen einer Person und einem von einem Computer gesteuerten Allzweckmanipulator“

- US Patent „Cobots“ (1997), J. Edward Colgate und Michael Peshkin



Belehrung



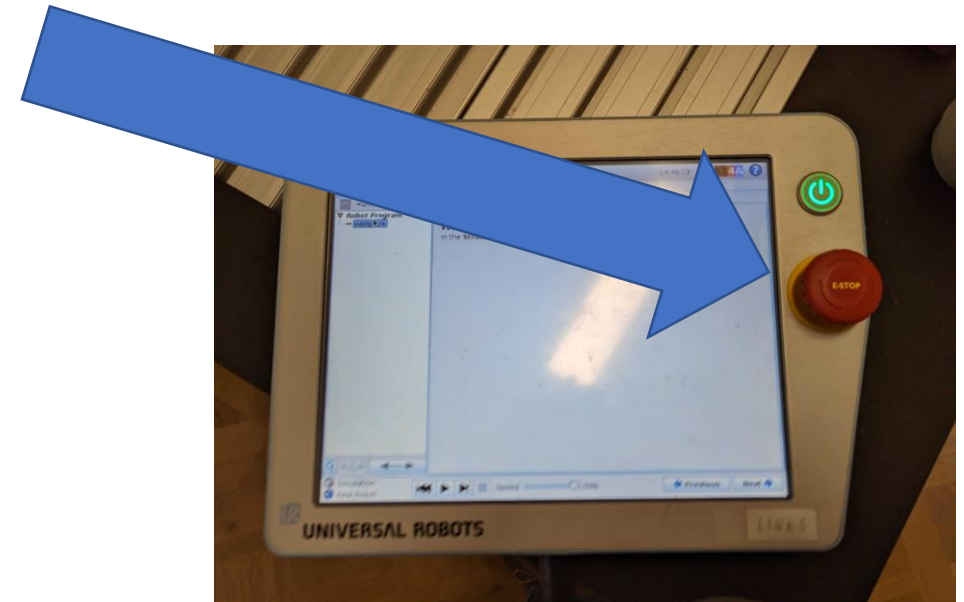
Immer jemand am „E-Stop“



Prüfen ob jemand oder etwas in der Nähe ist



Programm mit reduzierter Geschwindigkeit testen



Universal Robot

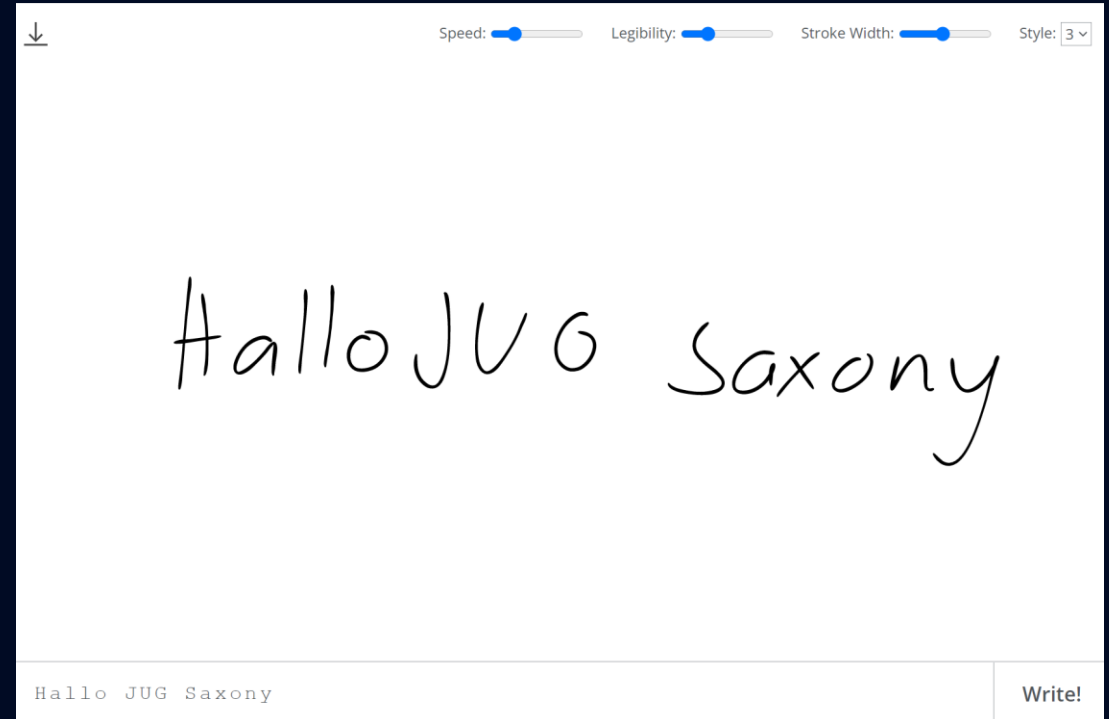
- Nutze URP und Freedrive um den Buchstaben „H“ einzulernen

```
def drawH():
    draw_z = 0.200
    # -30, -80, -120, -70, 90, -30 deg
    movej([-0.523599, -1.39626, -2.0944, -1.22173, 1.5708, -0.523599])
    # Move to first stroke
    movel(p[0.280, -0.290, 0.280, 2.2331, -2.2331, 0])
    # First stroke
    movel(p[0.280, -0.290, draw_z, 2.2331, -2.2331, 0])
    movel(p[0.200, -0.290, draw_z, 2.2331, -2.2331, 0])
    movel(p[0.200, -0.290, draw_z+0.01, 2.2331, -2.2331, 0])
    # Move to second stroke
    movel(p[0.280, -0.250, draw_z+0.01, 2.2331, -2.2331, 0])
    # second stroke
    movel(p[0.280, -0.250, draw_z, 2.2331, -2.2331, 0])
    movel(p[0.200, -0.250, draw_z, 2.2331, -2.2331, 0])
    movel(p[0.200, -0.250, draw_z+0.01, 2.2331, -2.2331, 0])
    # move to third stroke
    movel(p[0.240, -0.290, draw_z+0.01, 2.2331, -2.2331, 0])
    # third stroke
    movel(p[0.240, -0.290, draw_z, 2.2331, -2.2331, 0])
    movel(p[0.240, -0.250, draw_z, 2.2331, -2.2331, 0])
    movel(p[0.240, -0.250, draw_z+0.01, 2.2331, -2.2331, 0])

    # Move to home
    movej([-0.523599, -1.39626, -2.0944, -1.22173, 1.5708, -0.523599])
end
drawH()
```

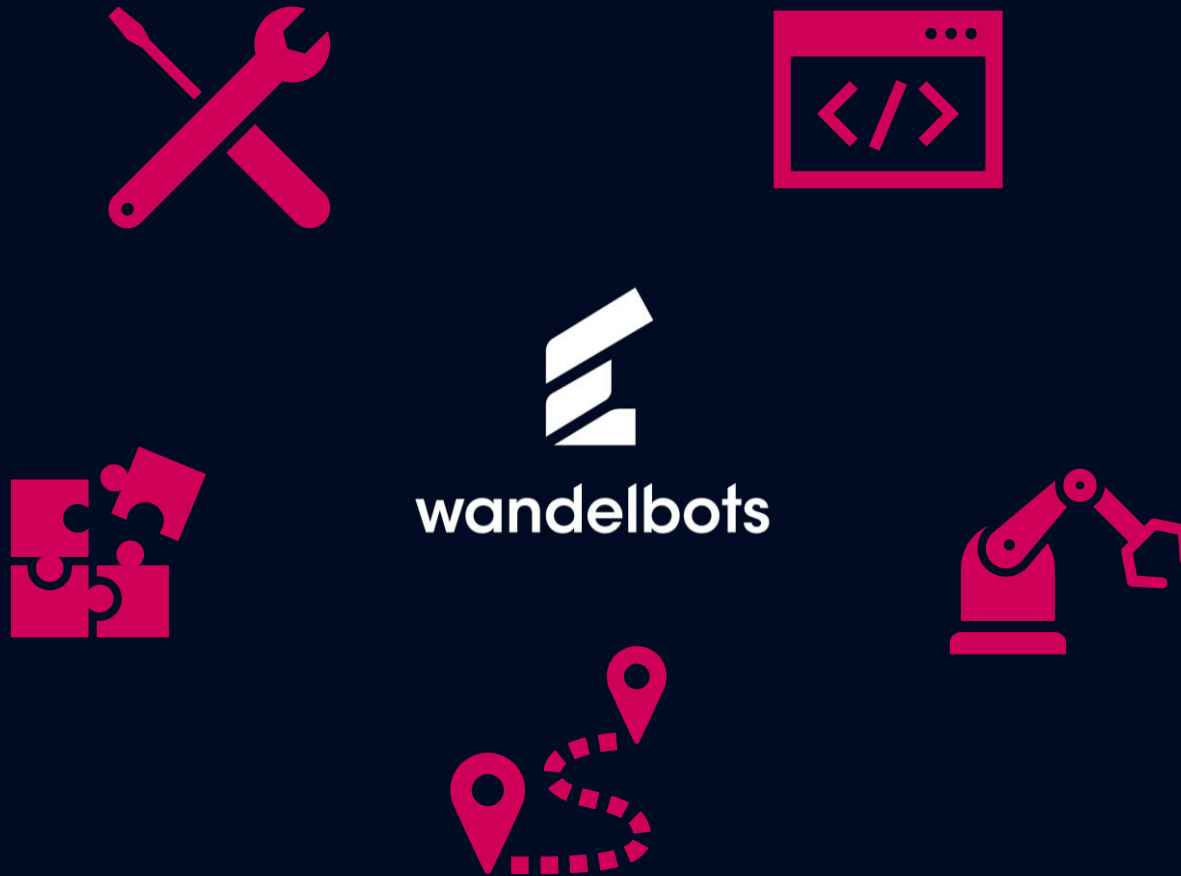
Weihnachtsmann Handschrift mit AI

- <https://github.com/sjvasquez/handwriting-synthesis>
- Erzeugt eine künstliche Handschrift
- Zeichen werden als einzelne Striche zurück gegeben
- Roboterpfad wird generiert



Screenshot: <https://www.calligrapher.ai/>

Wandelbots Developer Platform



Wandelbots Developer Platform

- Wandelscript
- Integration in Python
- Einfacher Zugriff auf Sensoren (z.B. Kameras)
- Integration von numpy, pandas, TensorFlow etc.

```
orientation_strategy := 'last'

home := [-450, -210, 200, -2, 2.5, 0]
text_strokes := create_letter_strokes("Hello JUG Saxony")
reference := [-520, -362, 9.8, 0, 0, 0]
paper_z := 0
air_z := 5

move via p2p() to home

move via p2p() to [-480, -350, 150, -2.22, 2.22, 0] :: [0, 0, 0, -0.5, 0, 0]
c := 10
for i := 0..<len(text_strokes):
    stroke_points := text_strokes[i]
    for j := 0..<len(stroke_points):
        point := stroke_points[j]
        x := point[0]
        y := -point[1]
        if j==0:
            move via line() to reference :: [x, y, air_z]

            move via line() to reference :: [x, y, paper_z]

            if j==len(stroke_points)-1:
                move via line() to reference :: [x, y, air_z]
        c := c-1
    if c == 0:
        sync
        c := 10
move via p2p() to home
sync
```