

```

1 Übergabestring - Syntax:
2
3 Trenner zwischen Blöcken: ;
4 Trenner zwischen Infos: ,
5 Marker für Beginn eines Blocks: #
6 Marker für Ende der Übergabe: ;;
7 Zahlen (größer als 1 Stelle): In Char-Array (=String) gecasteter Short bzw. Int, feste Länge der Zahlen: 16bit oder 32bit
8 Fließkommazahlen: Vorkommastelle: Int (als Char-Array), Nachkommastelle: Short (als Char-Array), durch '.' getrennt,
9 feste Länge der Zahlen: 16bit oder 32bit, bisher nur für Wartezeit
10
11 Block-IDs: M - Motor
12           L - Lampe
13           D - Digitaler Input <- notwendig für Anzeige im Web
14           A - Analoger Input <- notwendig für Anzeige im Web
15           S - Warten (Schlafen)
16           I - If (Verzweigung)
17           J - EndIf (Ende Verzweigung)
18           W - While (Schleife)
19           X - EndWhile (Ende Schleife)
20
21 Infos: Motor: Port-Nr (0..3), Richtung (0=Linkslauf 1=Rechtslauf), Drehzahl (0..8) (vgl. RoboPRO)
22         Lampe: Port-Nr (0..3), Helligkeit (0..7)
23         DIn: Port-Nr (0..7)
24         AIn: Port-Nr (0..7)
25         Warten: Wartezeit in Sekunden (XXXX.XXX) (Vorkommastellen (Sekunden) 32 bit unsigned int = 4x char (max 4.29mrd)),
26                 Nachkommastellen (ms) 16bit unsigned short = 2x char (max 3 Stellen), kleinste Zeiteinheit = 1ms)
27         If: Abzufragender Input (D, A), Input Port-Nr (0..7), Bedingung ('< > =' bei AIn, '=' bei DIn), Wert (16bit unsigned
28             short = 2x char bei AIn (AD-Wandler liefert 12bit 0..4095), 0/1 bei DIn), ID_I (16bit unsigned short) fortlaufend
29         EndIf: ID_I (16bit unsigned short) (zugehörig zum entsprechenden If, um Verschachtelung zu erlauben)
30         While: Art (Z, Zählschleife), Anzahl Durchläufe (16bit unsigned short), ID_W (16bit unsigned short) fortlaufend
31                Art (D,A Normal), Input Port-Nr (0..7), Bedingung ('< > =' bei AIn, '=' bei DIn), Wert (16bit unsigned short
32                    bei AIn (AD-Wandler liefert 12bit 0..4095), 0/1 bei DIn), ID_W (16bit unsigned short) fortlaufend
33         EndWhile: ID_W (16bit unsigned short) (zugehörig zum entsprechenden While, um Verschachtelung zu erlauben)
34 Bemerkung: ID_I werden durchgezählt (0..max Wert für short), ID_W werden separat durchgezählt (0..max Wert für short)
35
36 Für erste Testläufe ohne verschlüsselte int- und short-Werte (max Char-Größe: 0..255)
37 Infos: Motor: Port-Nr (0..3), Richtung (0=Linkslauf 1=Rechtslauf), Drehzahl (0..8) (vgl. RoboPRO)
38         Lampe: Port-Nr (0..3), Helligkeit (0..7)
39         DIn: Port-Nr (0..7)
40         AIn: Port-Nr (0..7)
41         Warten: Wartezeit in Sekunden (XXXX.XXX) (Vorkommastellen (Sekunden) 8bit unsigned char),
42                 Nachkommastellen (ms) 8bit unsigned char (max 2 Stellen), kleinste Zeiteinheit = 10ms)
43         If: Abzufragender Input (D, A), Input Port-Nr (0..7), Bedingung ('< > =' bei AIn, '=' bei DIn), Wert/16 (8bit unsigned
44             char bei AIn, siehe Bem. (AD-Wandler liefert 12bit 0..4095), 0/1 bei DIn), ID_I (8bit unsigned char) fortlaufend
45         EndIf: ID_I (8bit unsigned char) (zugehörig zum entsprechenden If, um Verschachtelung zu erlauben)
46         While: Art (Z, Zählschleife), Anzahl Durchläufe (8bit unsigned char), ID_W (8bit unsigned char) fortlaufend
47                Art (D,A Normal), Input Port-Nr (0..7), Bedingung ('< > =' bei AIn, '=' bei DIn), Wert/16 (8bit unsigned char
48                    bei AIn, siehe Bem. (AD-Wandler liefert 12bit 0..4095), 0/1 bei DIn), ID_W (8bit unsigned char) fortlaufend
49         EndWhile: ID_W (8bit unsigned char) (zugehörig zum entsprechenden While, um Verschachtelung zu erlauben)
50 Bemerkungen:
51 - ID_I werden durchgezählt (0..max Wert für char), ID_W werden separat durchgezählt (0..max Wert für char)
52 - Analogwerte werden in der HMI durch 16 geteilt, auf SW-Seite wieder mit 16 multipliziert (so bleibt der Wert innerhalb der
53   char-Größe. Der Genauigkeitsverlust sei für die Testläufe ausnahmsweise vernachlässigbar...
54
55 Beispiele für Blöcke:
56 Motor: #M,1,1,8; -> Block enthält Motor, Port = 1, Richtung = 1 = Rechtslauf, Geschwindigkeit = 8
57 Lampe: #L,2,4; -> Block enthält Lampe, Port = 2, Helligkeit = 4
58 DIn: #D,3; -> Digitaler Input, Port = 3
59 AIn: #A,3; -> Analoger Input, Port = 3
60 Warten: #S,'005','500'; -> Warten, Dauer = 5.5 Sekunden
61 If: #I,D,4,1,'0'; -> Wenn DIn mit Port-Nr 4 aktiv (Taster gedrückt = 1), ID_I = 0
62     dann nachfolgende Blöcke abarbeiten, sonst direkt zu EndIf mit ID_I = 0
63 If: #I,A,5,=,'128','1'; -> Wenn AIn mit Port-Nr 5 genau den Wert 128 hat (z.B. Rohwert Temperatursensor), ID_I = 1
64     dann nachfolgende Blöcke abarbeiten, sonst direkt zu EndIf mit ID_I = 1
65 EndIf: #J,'0'; -> EndIf, ID_I = 0
66 EndIf: #J,'1';
67 While: #W,Z,'5','3'; -> Solange bei Zählschleife der Wert >0, ID_W = 3
68     dann nachfolgende Blöcke abarbeiten, sonst direkt zu EndWhile mit ID_W = 3
69 While: #W,D,1,'0','4'; -> Solange DIn mit Port-Nr 1 nicht aktiv (Taster nicht gedrückt = 0), ID_W = 4
70     dann nachfolgende Blöcke abarbeiten, sonst direkt zu EndWhile mit ID_W = 4
71 While: #W,A,7,>,'32','5'; -> Solange AIn mit Port-Nr 7 den Wert >32 hat (z.B. Rohwert Temperatursensor), ID_W = 5
72     dann nachfolgende Blöcke abarbeiten, sonst direkt zu EndWhile mit ID_W = 5
73 EndWhile: #X,3; -> EndWhile, ID_W = 3

```

```
74 EndWhile: #X,4;
75 EndWhile: #X,5;
76
77 Beispiel für ein Programm, bei welchem der Roboter (1 Motor, Port 1, Rechtslauf, Geschwindigkeit 8) losfährt, alle 0,1 Sekunden den Taster abfragt und bei einem Tasterdruck (Digital,
Port 2) stehenbleibt:
78 #M,1,1,8;#W,D,2,0,'1';#S,'0'.'100';#X,'1';#M,1,1,0;;
79
80 Offene Punkte:
81 - Casten der Werte:
82     32bit int -> 4x char: char0 = zahl%256; zahl/=256; char1 = zahl%256; zahl/=256; etc. char3 = zahl%256;
83     4x char -> 32bit int: zahl = char0; zahl = zahl + 256*char1; zahl = zahl + 256*256*char2; zahl = zahl + 256*256*256*char3;
84 - Rückgabe an die HMI (via Container -> Aufbau?)
85     von HMI:     bool Start, Pause/Weiter, Stopp;
86     an HMI:      int [8] Analogwerte/Digitalwerte;
87                 bool Erfolgreich aufgebaut, Fehler im Programm, Programm beendet;
```