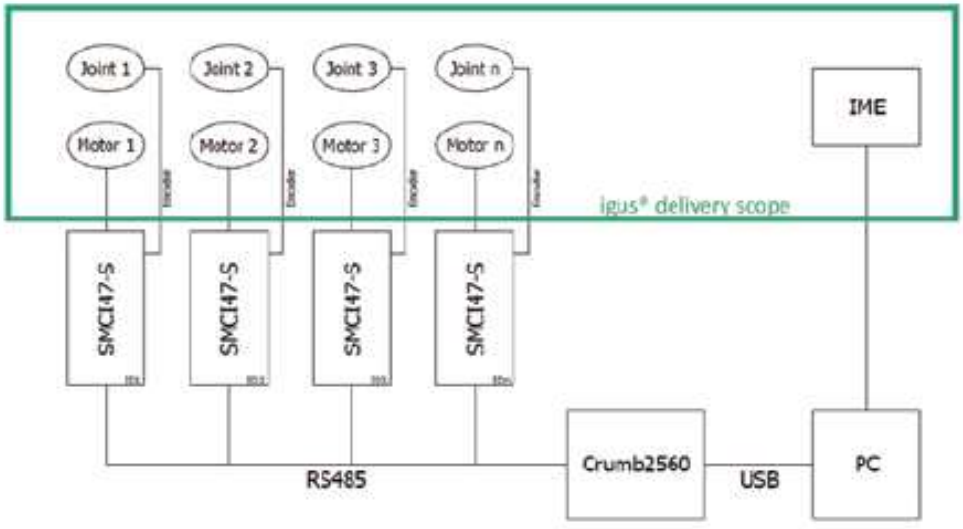
Vorläufige Dokumentation der Steuerung für robolink® Gelenkarme zur Verwendung der igus Software IME (igus® motion editor)



Version 01, Stand 10.2013
Erstellt von B.Eng. Felix Berger
fberger@igus.de
+49 – (0)2203 – 9649 - 7331

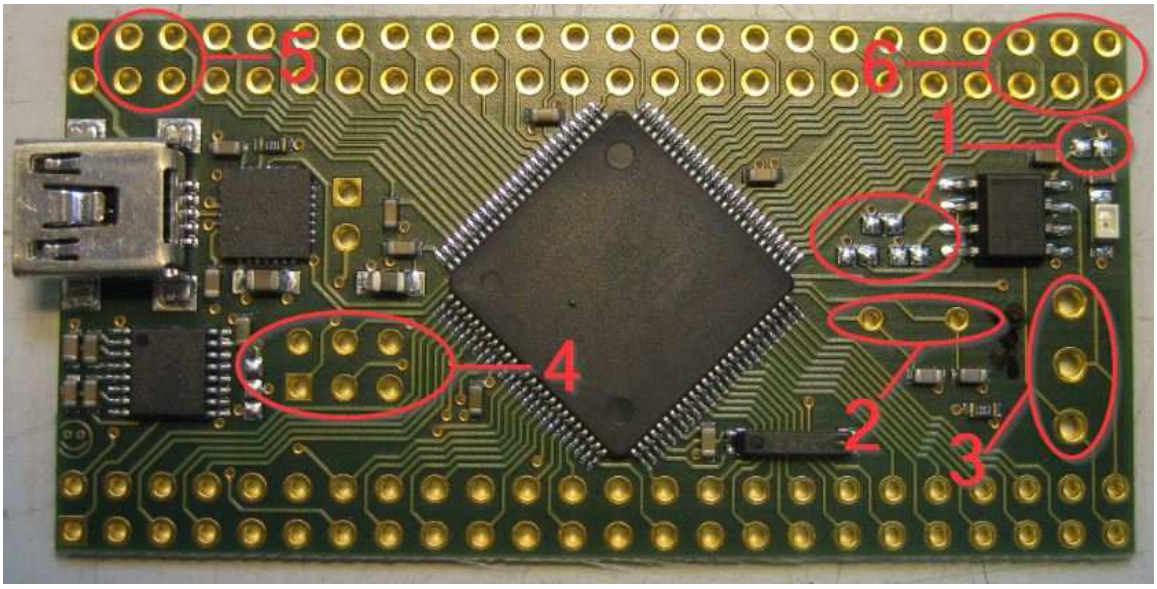
1a) Stückliste Hardware

Menge	Artikel	Bezugsquelle	Bestell-Nr.	Anmerkung
1	robolink Gelenkarm	igus		
1-6	Nanotec SMCI47-S-2	Nanotec		RS485 Bus
1	RS485 Konverterkabel	Nanotec	ZK-RS485-USB	
1	Crumb2560 V1.1 AVR ATmega Modul	Chip45	crumb2560-1.1	16.000 MHz + Stiftleisten
1	ATAVRISP-mkII Programmier Adapter	Chip45	avrisp2	
1	Stromversorgung 5V			
1	Stromversorgung 48V			ca. 5A pro Steuerung
1	RS485 Stecker	Conrad	740389 – 05	
1	RS485 Buchse	Conrad	740631 – 05	
1	USB Kabel A -> Mini B	Conrad	975416 – 05	
1m	Flachbandkabel	Conrad	601922 – 05	
1-6 + 1	D-Sub Stecker Flachkabel	Conrad	711357 – 05	
2	D-Sub Buchse Flachkabel	Conrad	711373 – 05	
2	D-Sub Stecker Lötkelch	Conrad	742066 – 05	
1	D-Sub Buchse Lötkelch	Conrad	742082 – 05	
4	Widerstand 120 Ω	Conrad	418145 – 05	
0,5m	Leitung 5 x 0,34 mm²	igus	CF130.03.05.UL	
1	Optokoppler	Conrad	505454 – 05	
1-6	Motorleitung	igus	CF.INI-P5-M12-BW-3	

1b) Stückliste Software

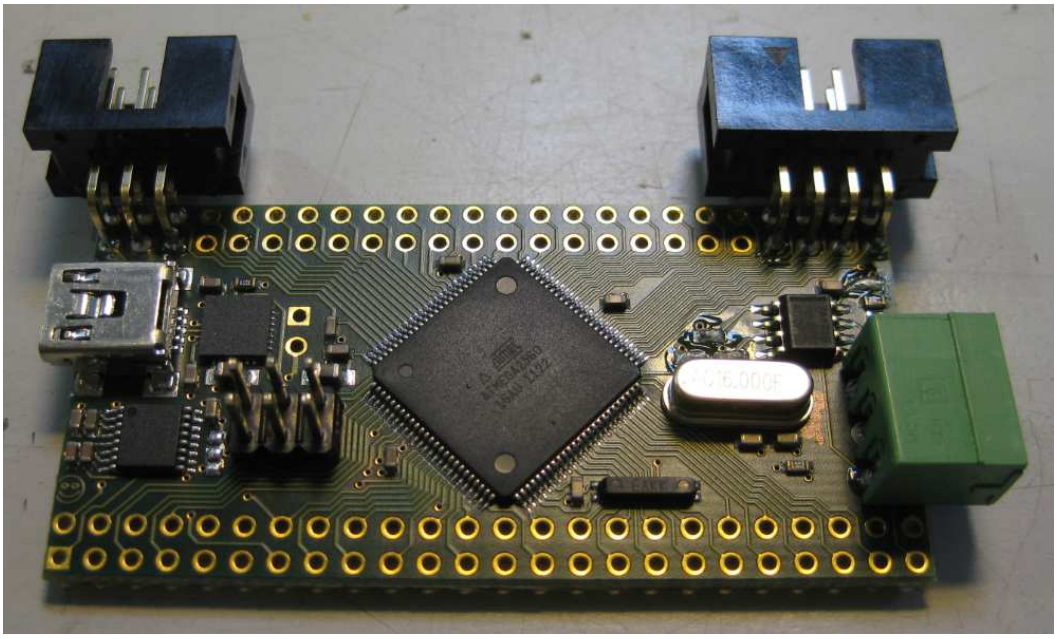
Programm	Aktuelle Version	Bezugsquelle
IgusMotionEditor	v 2397	www.igus.de/robolink/software
CP210x VCP Driver	6.6.1	www.silabs.com
NanoPro	1.70.0.1	www.nanotec.de
Java-Programm NanoJEasy		www.igus.de/robolink/software

2a) Hardwarekonfiguration Crumb2560



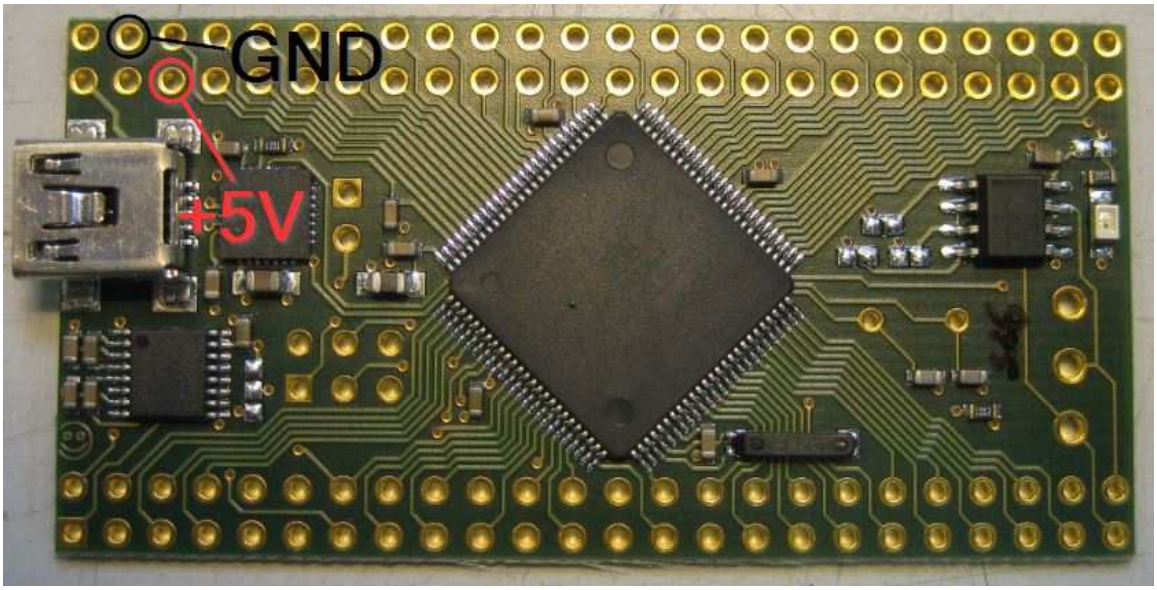
1. Kontakte brücken (4x)
2. Quarz (16.000MHz)
3. RS485 Stecker (Conrad 740389-05)
4. Stiftleiste (6x2)
5. Stiftleiste (24x2 o. kürzer) (nur der markierte Bereich wird verwendet)
6. Stiftleiste (24x2 o. kürzer) (nur der markierte Bereich wird verwendet)

2b) Fertig bestückte Platine



3a) Bootloader installieren

1. AVR Programmieradapter an USB (PC/Laptop) anschließen
2. Treiber aus dem IgusMotionEditor-Verzeichnis verwenden (…/contrib/libusb)
   Bei Treibersignatur-Problemen den PC neustarten und beim Bootvorgang F8 drücken. Dort die Treibersignatur deaktivieren. Die Deaktivierung ist bis zum nächsten Neustart wirksam.
3. Programmieradapter an 2x6 Stiftleiste anschließen (rote Seite Richtung USB-Port)
4. Crumb2560 mit 5V Gleichspannung versorgen



5. Flashtool.exe im IME Verzeichnis ausführen
6. Konfiguration wie Bild vornehmen und Bootloader flashen

7. Nach erfolgreichem Flashvorgang muss der Mikrocontroller dauerhaft blinken

Igus GmbH
Spicher Straße 1a – 51147 Köln
Tel: +49-(0)2203 / 9649-7331
eMail: FBerger@igus.de

3b) Firmware flashen

1. PC neustarten um Treibersignierung wieder zu aktivieren
2. Crumb2560 per USB Kabel mit PC/Laptop verbinden
3. CP210x VCP Treiber installieren
4. Flashtool.exe im IME Verzeichnis ausführen
5. Konfiguration wie Bild vornehmen und Firmware flashen

6. Nach erfolgreichem Flashvorgang muss der Microcontroller nach dem Einschalten kurz blinken

Der Crumb2560 Mikrocontroller ist nun für den Einsatz mit dem IgusMotionEditor einsatzbereit!

4) Konfiguration Nanotec SMCI47-S NanoPro

1. Motoradresse wie abgebildet auf „1“ stellen

2. Steuerung per RS485-Konverterkabel mit PC/Laptop verbinden
3. Steuerung mit 48V Gleichspannung versorgen
4. NanoPro installieren und starten
5. Meldung „Konfiguration aus Steuerung lesen“ immer verneinen!
6. Unter dem Reiter „Kommunikation“ die COM-Schnittstelle des RS485 Konverters wählen
7. Firmware aktualisieren: System → Firmware ändern → wähle Firmware → RS485 / 04-02-2011
8. Erfolgreiche Aktualisierung prüfen

9. Unter dem Reiter „Modus“ die Steuerung in den Auslieferungszustand zurücksetzen
10. Steuerung neustarten
11. Motor → Motor entfernen
12. Motor → Motor hinzufügen → Adresse „1“
    (dieser Schritt ist notwendig um alle geänderten Software-Einstellungen zurückzusetzen)

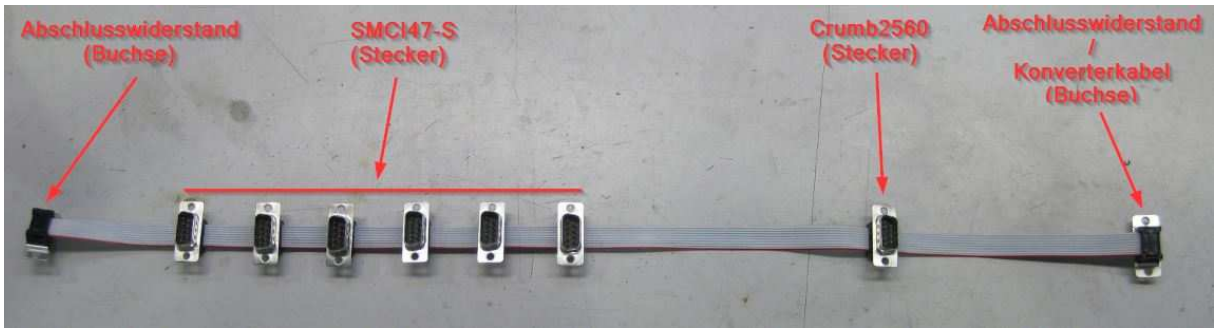
Igus GmbH
Spicher Straße 1a – 51147 Köln
Tel: +49-(0)2203 / 9649-7331
eMail: FBerger@igus.de

13. Reiter Statusanzeige → Autostart aktivieren → Daten speichern → Konfiguration in Steuerung schreiben
    (wenn die Autostart-Funktion fehlt, kurz zum Modus-Reiter wechseln)
14. Programm schließen und Steuerung ausschalten bzw. Schritte 1-14 für weitere Steuerungen wiederholen

5) Buskabel vorbereiten

Hinweis: Die hier aufgezeigte Busleitung ist eine schnelle und kostengünstige Alternative zu professionellen Busleitungen. Wir übernehmen keine Garantie bzgl. Störungen und Übertragungsfehler!

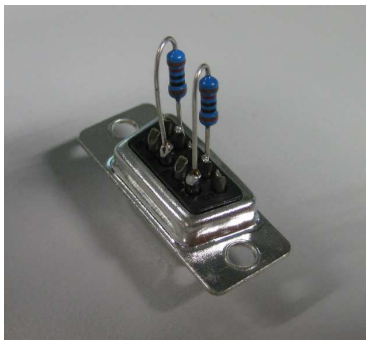
1. Buskabel wie abgebildet konfektionieren. Pin 1 Stecker/Buchse immer auf rote Ader!
   1-6x SMCI47-S Stecker – je nach Anzahl der Achsen



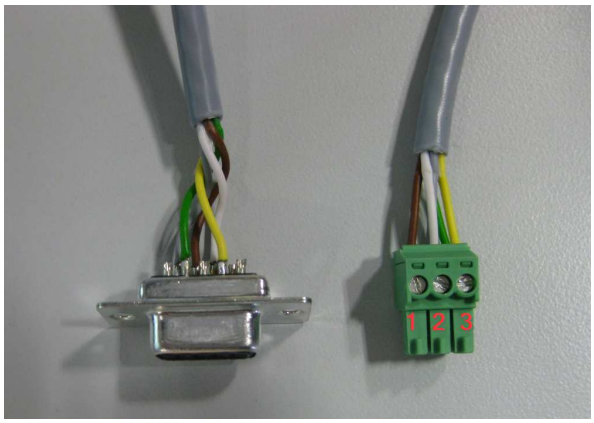
2. Pin 3 aller Stecker mit einer dünnen Zange entfernen. 5V Leitung wird nicht benötigt und kann Störungen verursachen.

Igus GmbH
Spicher Straße 1a – 51147 Köln
Tel: +49-(0)2203 / 9649-7331
eMail: FBerger@igus.de

3. Abschlusswiderstände vorbereiten: 120Ω Widerstand zwischen Pin 2+7 und 4+9
   D-Sub Stecker mit Lötkelch verwenden (Conrad 742066-05).



4. Verbindungsleitung Crumb2560 (igus CF130.03.05.UL)
   D-Sub Buchse mit Lötkelch verwenden (Conrad 742082-05).



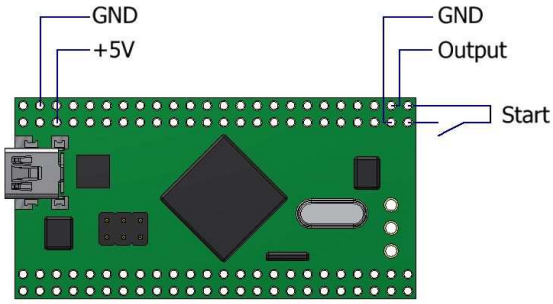
Pin D-Sub	Pin Crumb2560
2	3
4	3
7	2
8	1
9	2

6a) Geräteanschluss Nanotec SMCI47-S-2

Vor dem Anschluss aller Leitungen und des Buskabels müssen die Motoradressen 1-6 vergeben werden.

Input 1	- *NC* -
Input 2	- *NC* -
Input 3	- *NC* -
Input 4	- *NC* -
Input 5	- *NC* -
Input 6	- *NC* -
Signal GND	GND
Output 1	- *NC* -
Output 2	- *NC* -
Output 3	- *NC* -
Analog In	Robolink Hall-Sensor
GND	GND
Brake	- *NC* -
GND	- *NC* -
+5 V	Robolink +5V
Channel B	Robolink Channel B
Channel A	Robolink Channel A
Index	Robolink Index
GND	Robolink GND
Winding A	Motor A - white
Winding A\	Motor A\ - brown
Winding B\	Motor B\ - black
Winding B	Motor B - blue
UB 24-48 V	+48V
GND	GND

6b) Geräteanschluss Crumb2560



Der Ausgang des Crumb2560 Controllers gibt max. 20mA / 5V aus. Es wird die Verwendung eines Optokopplers empfohlen.

7) Igus Motion Editor konfigurieren

1. Calib-Datei

Beispiel-Einstellung für ein 2-Achs RL-50-001 Gelenk

```
[Joint0]
name=Pivoting                   # Displayed Name
type=X                          # Joint type (X = Pivoting / Z = Rotation)
address=1                       # Motor controller address
lower_limit=-1.5708             # Lower joint angle limit in radians
                                ( Pi/180*angle )
upper_limit=1.5708              # Upper joint angle limit in radians
                                ( Pi/180*angle )
offset=0.0                      # Joint offset in radians ( Pi/180*angle )
invert=0                        # Invert the axis (0 or 1)
encoder_steps_per_turn=6400     # 360/1,8*X*i   (X = 1 full-step / 2 half-step)
                                (i = gear reduction)
motor_steps_per_turn=6400       # 360/1,8*X*i   (X = 1 full-step / 2 half-step)
                                (i = gear reduction)
max_current=30                  # Current moving
hold_current=20                 # Current stop
length=0.10                     # Displayed lenght
joystick_axis=0                 # Joystick axis
joystick_invert=0               # Invert joystick axis

[Joint1]
name=Rotation                   # Displayed Name
type=Z                          # Joint type (X = Pivoting / Z = Rotation)
address=2                       # Motor controller address
lower_limit=-6.2832             # Lower joint angle limit in radians
                                ( Pi/180*angle )
upper_limit=6.2832              # Upper joint angle limit in radians
                                ( Pi/180*angle )
offset=0.0                      # Joint offset in radians ( Pi/180*angle )
invert=0                        # Invert the axis (0 or 1)
encoder_steps_per_turn=6400     # 360/1,8*X*i   (X = 1 full-step / 2 half-step)
                                (i = gear reduction)
motor_steps_per_turn=6400       # 360/1,8*X*i   (X = 1 full-step / 2 half-step)
                                (i = gear reduction)
max_current=30                  # Current moving
hold_current=20                 # Current stop
length=0.10                     # Displayed lenght
joystick_axis=0                 # Joystick axis
joystick_invert=0               # Invert joystick axis
```

2. Java-Programm NanoJEasy

Beispiel-Einstellung für ein 2-Achs RL-50-001 Gelenk

```
3   class NanoJMotorControl {
4       // for 35:1: 2, for 16:1 with old encoder settings: 0, for 16:1 with correct values: 1
5       final static int ENCODER_SHIFT = 1;
6       final static int POSITION_BIAS = 16384; // has to match in µC code
7
8       // function to initialize the controller
9       static void initializeController() {
10
11          // pause register is used to communicate a state with the PC
12          // 0 controller just started
13          // 1 controller searching for middle position
14          // 2 normal mode
15          // 3 compliance mode
16          // other, halt the motor
17          drive.SetPause( 0 );
18
19          if (config.GetMotorAddress() == 1) {
20              config.SetRotencInc( 310 );            // Encoder-resolution / gear-reduction (4960 / 16)
21              config.SetEncoderDirection(0);
22              util.SetStepMode(2);
23          }
24
25          if (config.GetMotorAddress() == 2) {
26              config.SetRotencInc( 290 );
27              config.SetEncoderDirection(1);
28              util.SetStepMode(2);
29          }
30
```

- final static int ENCODER_SHIFT: Getriebe 16 = 1; Getriebe 35 = 2
- config.GetMotorAddress: Hardware-Adresse der Steuerung
- config.SetRotencIng: Encoderauflösung / Getriebeuntersetzung
  Encoderauflösung RL-50 Schwenk: 4960
  RL-50 Rotation: 4640
  RL-90 Schwenk: 9920
  RL-90 Rotation: 9920
- config.SetEncoderDirection: Drehrichtung Encoder umkehren
  Fährt der Arm bei der Initialisierung langsam Richtung Anschlag, „1“ verwenden
- util.SetStepMode: „2“ verwenden – RL-90-BL1 Rotation NEMA34 Motor: „32“

- drive.SetCurrent: Motorstrom anfangs gering einstellen
  NEMA17: Max 23% - 1,8A
  NEMA23: Max 56% - 4,2A
  NEMA34: Max 85% - 6,4A

3. Java Parameter auf Steuerung laden

   - COM-Port / Baudrate (115200) / Motoradresse einstellen
   - Compile Program
   - Transfer Program
   - Für 1-6 Steuerung widerholen
   - Das Java Programm wurde erfolgreich übertragen, wenn die rote LED der Steuerung dauerhaft blinkt