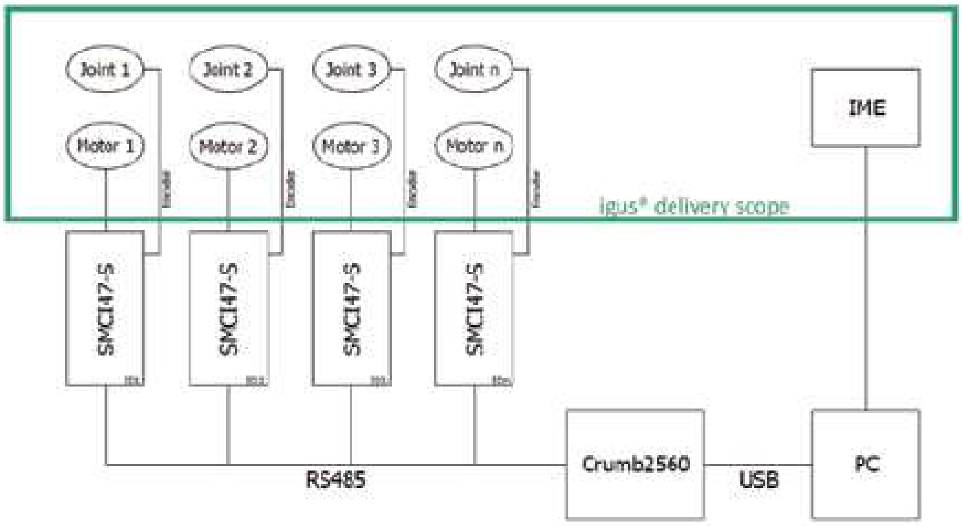
Preliminary documentation for the control for robolink® articulated arms for use with the igus IME Software (igus® motion editor)



Revision 01, Version 10.2013
Prepared by B.Eng. Felix Berger
fberger@igus.de
+49 – (0)2203 – 9649 - 7331

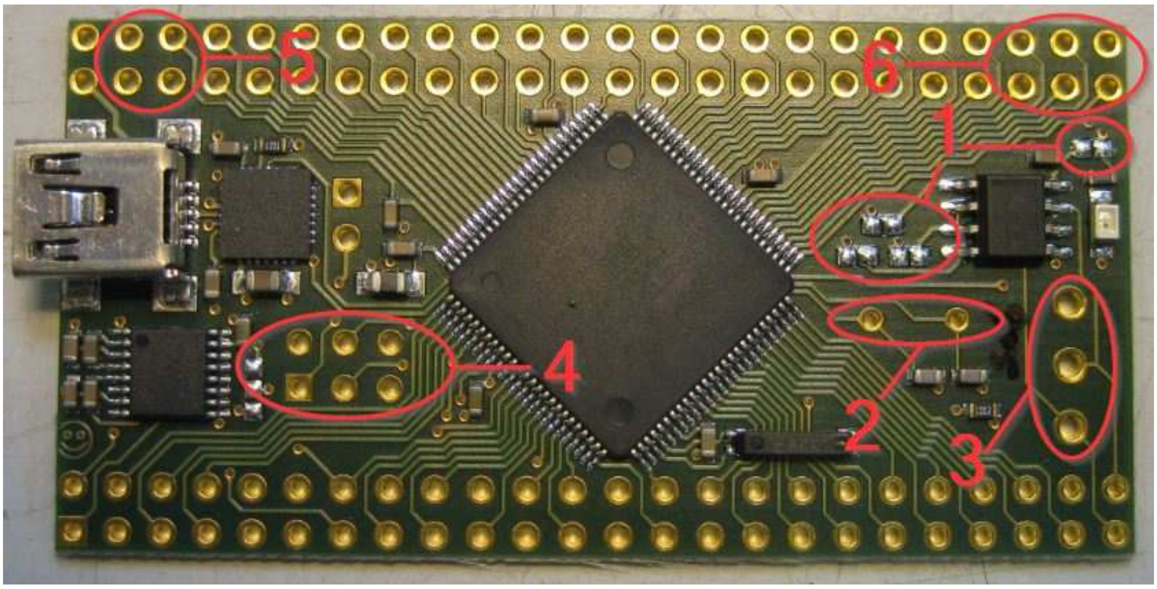
1a) Hardware parts list

Quantity	Part	Source	Order no.	Comment
1	robolink articulated arm	igus		
1-6	Nanotec SMCI47-S-2	Nanotec		RS485 Bus
1	RS485 Converter cable	Nanotec	ZK-RS485-USB	
1	Crumb2560 V1.1 AVR ATmega module	Chip45	crumb2560-1.1	16.000 MHz + headers
1	ATAVRISP-mkII programmable adapter	Chip45	avrisp2	
1	5V Power supply			
1	48V Power supply			approx. 5A per control
1	RS485 connector	Conrad	740389 – 05	
1	RS485 socket	Conrad	740631 – 05	
1	USB Cable A -> Mini B	Conrad	975416 – 05	
1m	Ribbon Cable	Conrad	601922 – 05	
1-6 + 1	D-Sub Ribbon cable connector	Conrad	711357 – 05	
2	D-Sub Ribbon cable socket	Conrad	711373 – 05	
2	D-Sub Solder cup connector	Conrad	742066 – 05	
1	D-Sub Solder cup socket	Conrad	742082 – 05	
4	120 Ω Resistor	Conrad	418145 – 05	
0.5m	5 x 0.34 mm² cable	igus	CF130.03.05.UL	
1	Optical coupler	Conrad	505454 – 05	
1-6	Motor cable	igus	CF.INI-P5-M12-BW-3	

1b) Software parts list

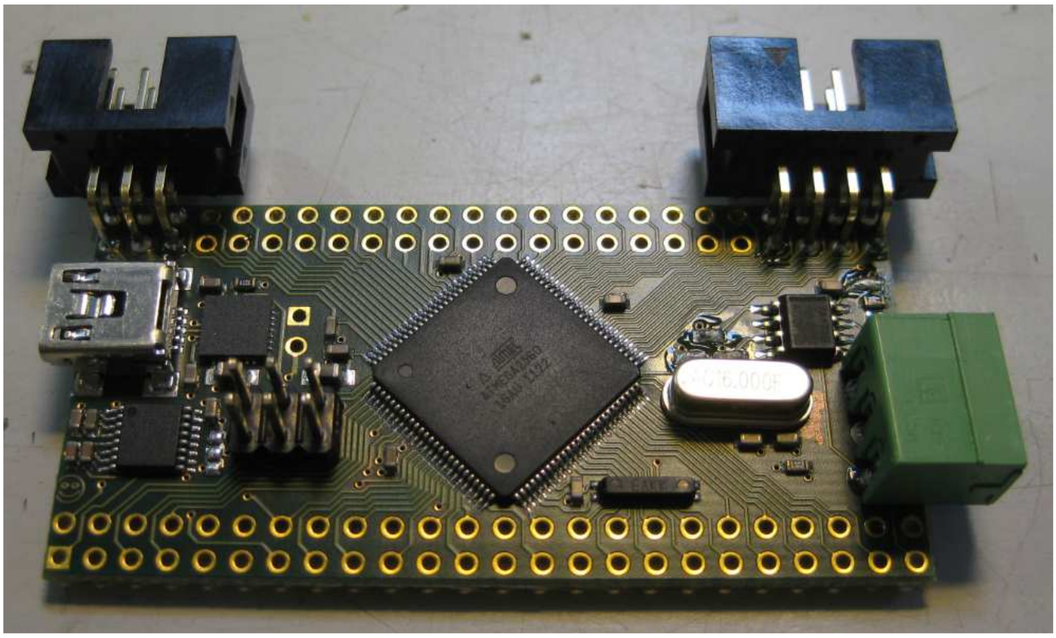
Programme	Current version	Source
IgusMotionEditor	v 2397	www.igus.de/robolink/software
CP210x VCP Driver	6.6.1	www.silabs.com
NanoPro	1.70.0.1	www.nanotec.de
Java-Programm NanoJEasy		www.igus.de/robolink/software

2a) Hardware configuration - Crumb2560



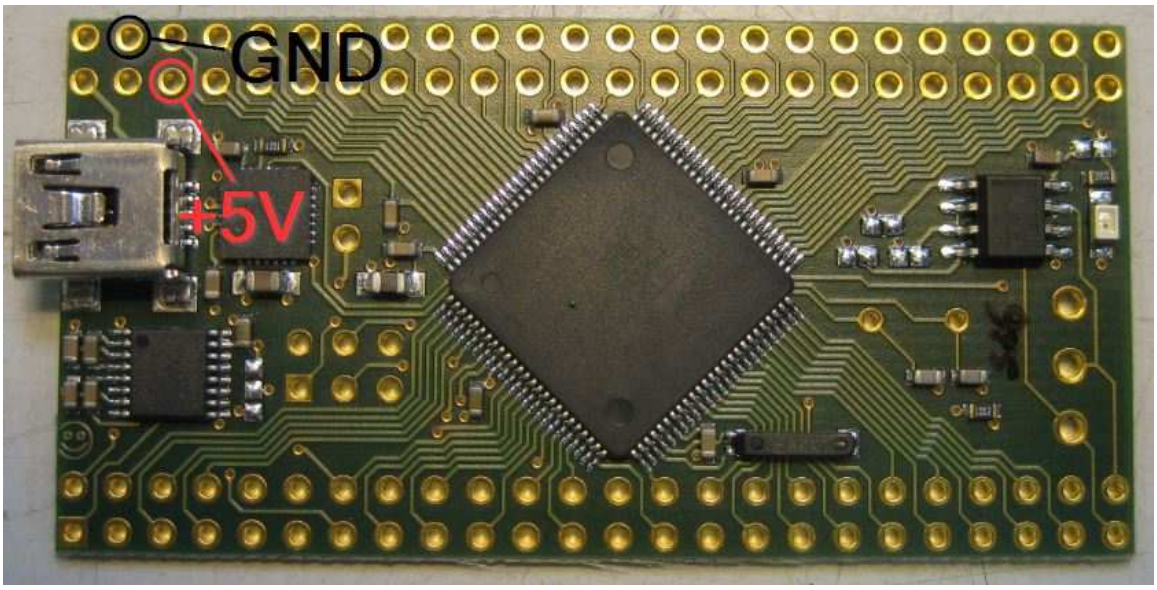
1. Bridge contacts (4x)
2. Quartz (16.000MHz)
3. RS485 connector (Conrad 740389-05)
4. Headers (6x2)
5. Headers (24x2 or shorter) (only the highlighted sections are used)
6. Headers (24x2 or shorter) (only the highlighted sections are used)

2b) Fully assembled circuit board

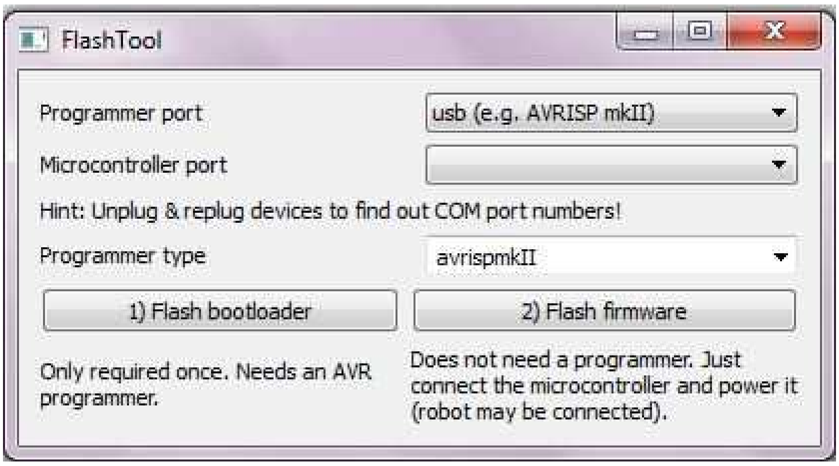


3a) Install boot loader

1. Connect AVR programming adapter to USB (PC/Laptop)
2. Use driver from the IgusMotionEditor directory (…/contrib/libusb)
   Restart the PC and press F8 during the boot process when driver signature problems occur. Then deactivate the driver signature. The deactivation is effective until the next restart.
3. Connect the programming adapter to 2x6 header (red side toward USB-Port)
4. Supply Crumb2560 with 5V DC



5. Execute Flashtool.exe in the IME directory
6. Configure as shown in the picture and flash boot loader


FlashTool

Programmer port: usb (e.g. AVRISP mkII)

Microcontroller port:

Hint: Unplug & replug devices to find out COM port numbers!

Programmer type: avrispmkII

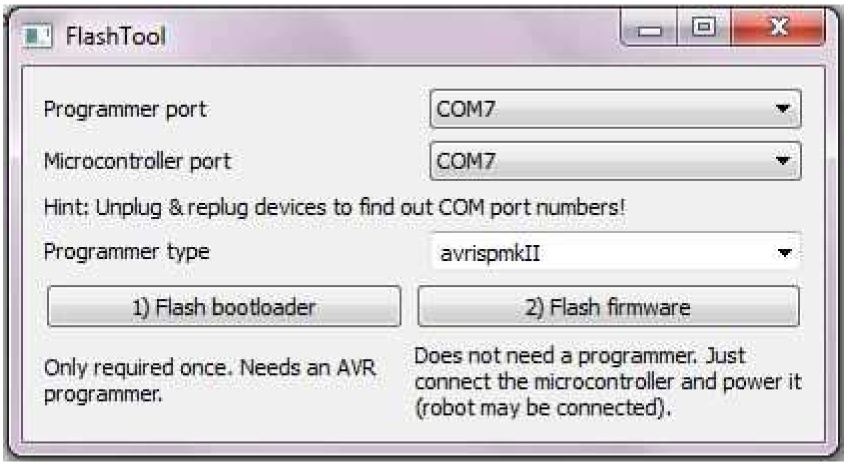
Only required once. Needs an AVR programmer.

Does not need a programmer. Just connect the microcontroller and power it (robot may be connected).

7. The microcontroller must blink constantly after a successful flash process

3b) Flash firmware

1. Restart PC to reactivate driver signature
2. Connect Crumb2560 with PC/Laptop with USB cable
3. Install CP210x VCP driver
4. Execute Flashtool.exe in the IME directory
5. Configure as shown in the picture and flash firmware



FlashTool

Programmer port: COM7

Microcontroller port: COM7

Hint: Unplug & replug devices to find out COM port numbers!

Programmer type: avrispmkII

1) Flash bootloader

Only required once. Needs an AVR programmer.

2) Flash firmware

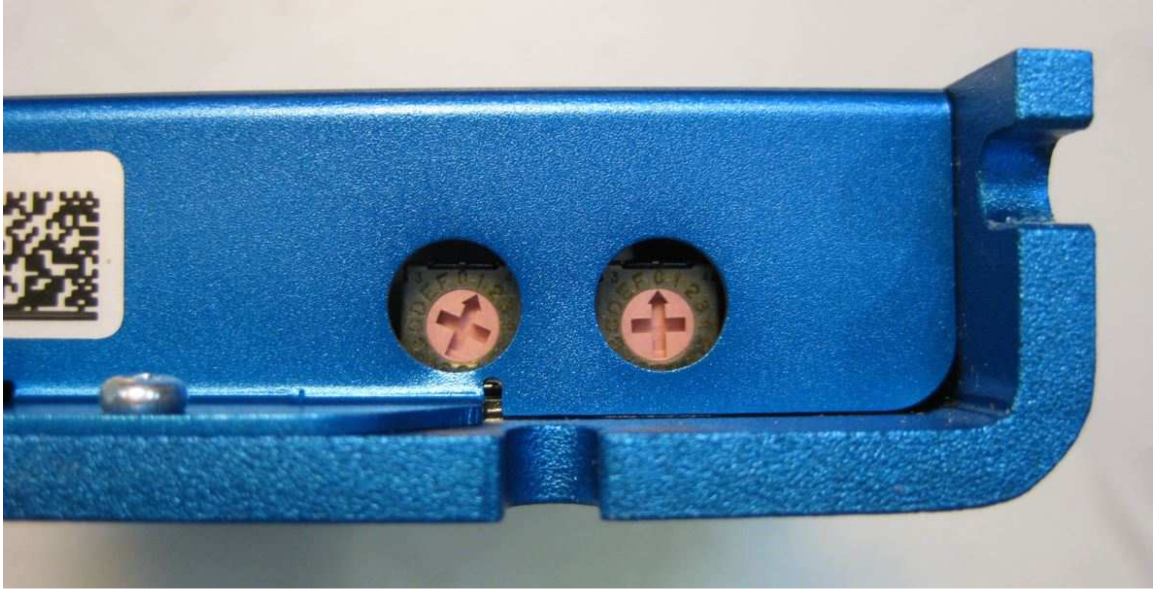
Does not need a programmer. Just connect the microcontroller and power it (robot may be connected).

6. Following a successful flash process, the microcontroller should blink briefly after power up

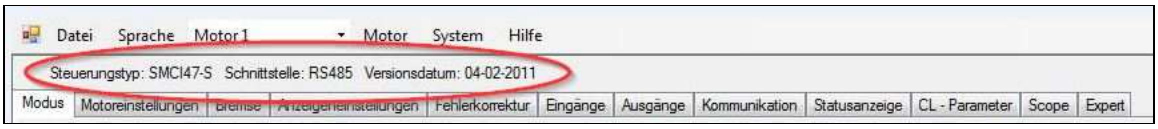
<u>The Crumb2560 microcontroller is now ready for use with the IgusMotionEditor!</u>

4) Nanotec SMCI47-S NanoPro configuration

1. Set motor address to "1" as shown



2. Using a RS485 converter cable, connect the control with PC/laptop
3. Supply control with 48V DC
4. Install and start NanoPro
5. Always deny the message "Read configuration from control"!
6. Select the COM interface of the RS485 converter under the "Communication" tab
7. Update firmware: System → Change Firmware → Select Firmware → RS485 / 04-02-2011
8. Verify successful update



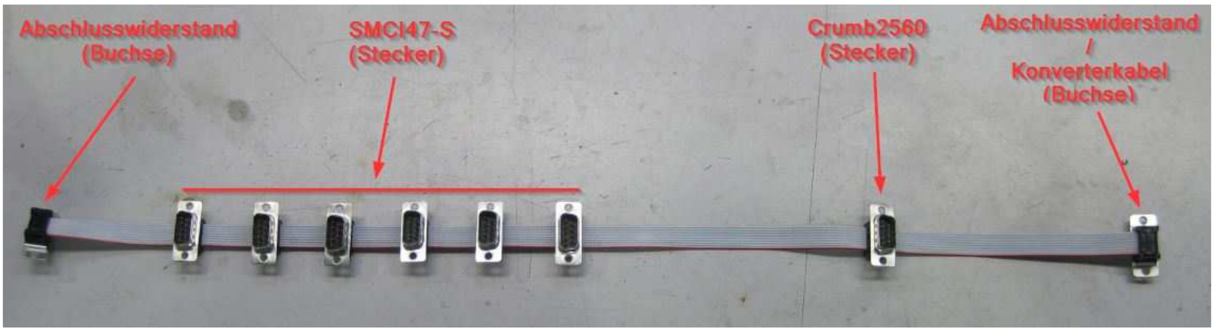
Steuerungstyp: SMCI47-S Schnittstelle: RS485 Versionsdatum: 04-02-2011

9. Reset the control to the default condition under the "Mode" tab
10. Restart control
11. Motor → delete motor
12. Motor → Add Motor → Address "1"
    (this step is necessary to reset all changed software settings)
13. Status display tab → activate autostart → save data → write configuration to control
    (briefly switch to mode tab if the autostart function is missing)
14. Close programme, turn off control, and repeat steps 1-14 for additional controls

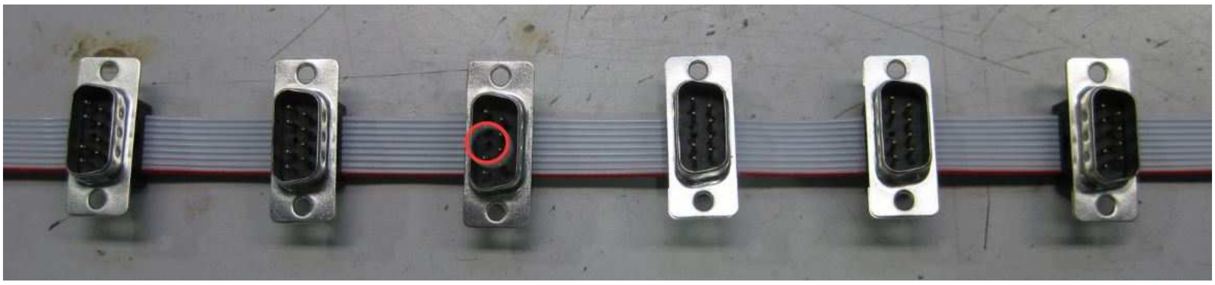
5) Prepare bus cable

Note: The bus cable shown here is a fast and cost-effective alternative to professional bus cables. We assume no guarantee for malfunctions and transmission errors!

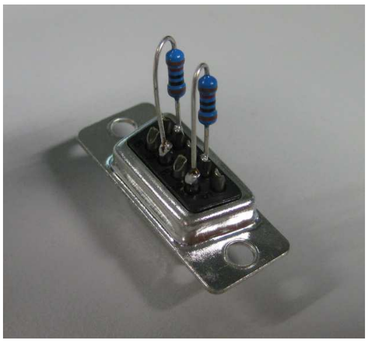
1. Harness the bus cable as shown. Pin 1 connector/plug always on the red core!
   1-6x SMCI47-S connector/s – depending on number of axis



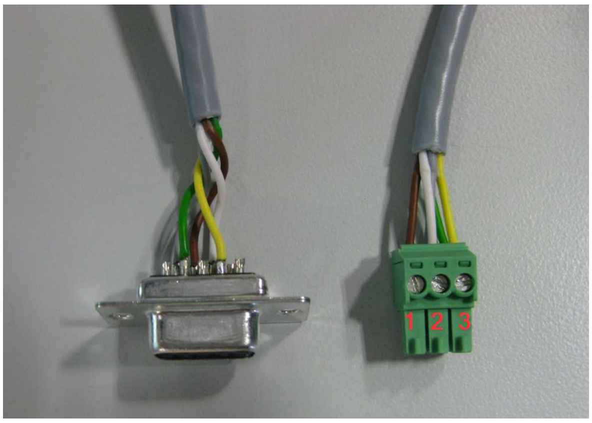
2. Remove pin 3 on all connectors with a needlenose plier. 5V line is not needed and can cause malfunctions.



3. Prepare terminating resistors: 120Ω resistor between pin 2+7 and 4+9
   Use D-Sub connector with solder cup (Conrad 742066-05).



4. Interface cable Crumb2560 (igus CF130.03.05.UL)
   Use D-Sub plug with solder cup (Conrad 742082-05).



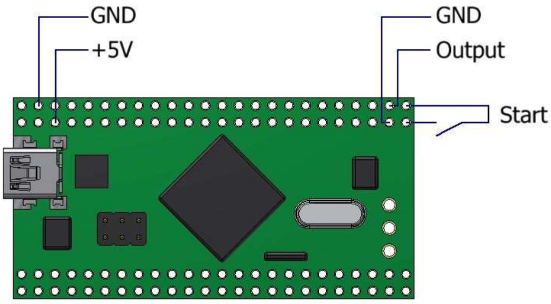
Pin D-Sub	Pin Crumb2560
2	3
4	3
7	2
8	1
9	2

6a) Nanotec SMCI47-S-2 device connection

Motor addresses 1-6 must be assigned before connecting all cables and the bus cable.

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* -
Analogue 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) Device connection Crumb2560



The Crumb2560 controller output is max. 20mA / 5V. We recommend using an optical coupler.

7) Configure Igus Motion Editor

1. Calibration file

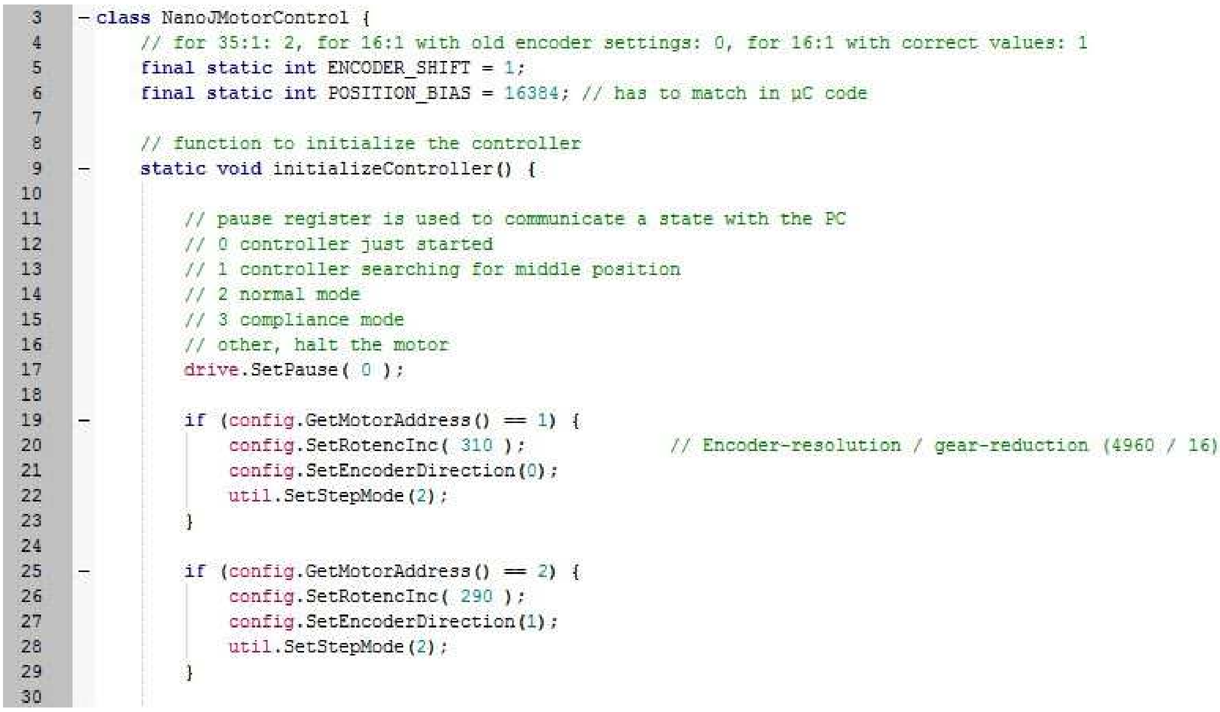
Example settings for a 2-axis RL-50-001 joint

```
[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 length
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 length
joystick_axis=0                   # Joystick axis
joystick_invert=0                 # Invert joystick axis
```

2. Java programme NanoJEasy

Example settings for a 2-axis RL-50-001 joint



```
class NanoJMotorControl {
    // for 35:1: 2, for 16:1 with old encoder settings: 0, for 16:1 with correct values: 1
    final static int ENCODER_SHIFT = 1;
    final static int POSITION_BIAS = 16384; // has to match in µC code

    // function to initialize the controller
    static void initializeController() {

        // pause register is used to communicate a state with the PC
        // 0 controller just started
        // 1 controller searching for middle position
        // 2 normal mode
        // 3 compliance mode
        // other, halt the motor
        drive.SetPause( 0 );

        if (config.GetMotorAddress() == 1) {
            config.SetRotencInc( 310 );          // Encoder-resolution / gear-reduction (4960 / 16)
            config.SetEncoderDirection(0);
            util.SetStepMode(2);
        }

        if (config.GetMotorAddress() == 2) {
            config.SetRotencInc( 290 );
            config.SetEncoderDirection(1);
            util.SetStepMode(2);
        }

```

- final static int ENCODER_SHIFT: Gearbox 16 = 1; Gearbox 35 = 2
- config.GetMotorAddress: Hardware address of the control
- config.SetRotencIng: Encoder resolution / Reduction gearing
  Encoder resolution RL-50 Pivot: 4960
  RL-50 Rotation: 4640
  RL-90 Pivot: 9920
  RL-90 Rotation: 9920
- config.SetEncoderDirection: Invert encoder rotating direction
  Use "1" when the arm slowly moves toward the dead stop during initialization
- util.SetStepMode: Use "2" – RL-90-BL1 Rotation NEMA34 Motor: “32”

- drive.SetCurrent: Use low motor power setting to begin
  NEMA17: Max 23% - 1.8A
  NEMA23: Max 56% - 4.2A
  NEMA34: Max 85% - 6.4A

3. Load Java parameters into the control

- COM port / baud rate (115200) / Enter motor address
- Compile programme
- Transfer programme
- Repeat for controls 1-6
- The Java programme was successfully loaded when the red LED on the control lights up steadily