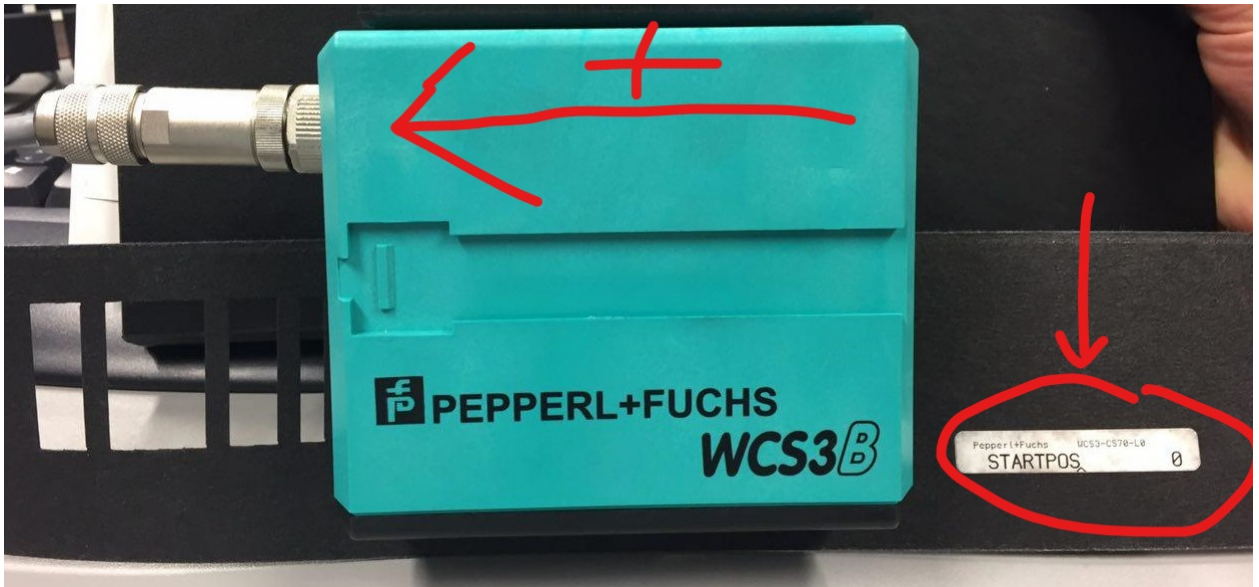


Pepperl und Fuchs Can Sensor einbinden

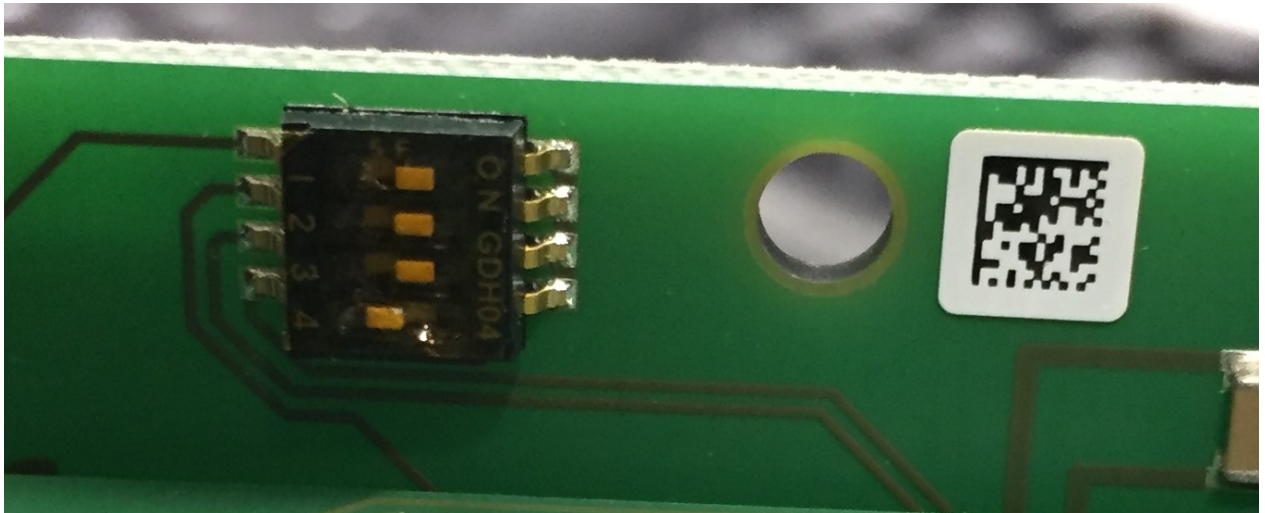
- Beide Can Sensoren („Lasco“ und „BOKU ILT“) wurden erfolgreich getestet am 09.02.2017 mit dem Projekt „CanSensorFunktioniert“.

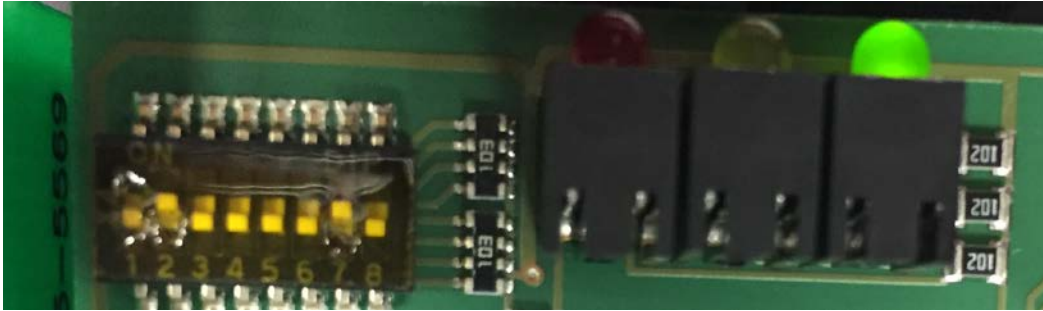
(Projekt liegt in „Dropbox\Projekt_RoboKran_Basisprogramm\05_BuR\CanSensor“)

- Der Sensor muss richtig herum montiert werden. Der Stecker schaut dabei in Positive Richtung, siehe Foto.



- DIP Schalter Einstellungen für bei diesen Tests wurden für beide Sensoren angeglichen und sind wie auf folgenden Abbildungen.

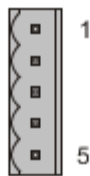




➤ Stecker am CAN Master Modul



Anschlussbelegung:



5-polige Steckerleiste

Anschlussbelegung		
Klemme	Bedeutung	
1	CAN _L	CAN Ground
2	CAN _L	CAN Low
3	SHLD	Schirm (Shield)
4	CAN _H	CAN High
5	NC	

➤ Stecker am Can Sensor

Elektrischer Anschluss

Der Anschluss erfolgt über einen 5-poligen M12-Stecker. Das Gegenstück des Steckverbinders gehört nicht zum Lieferumfang des Lesekopfes.

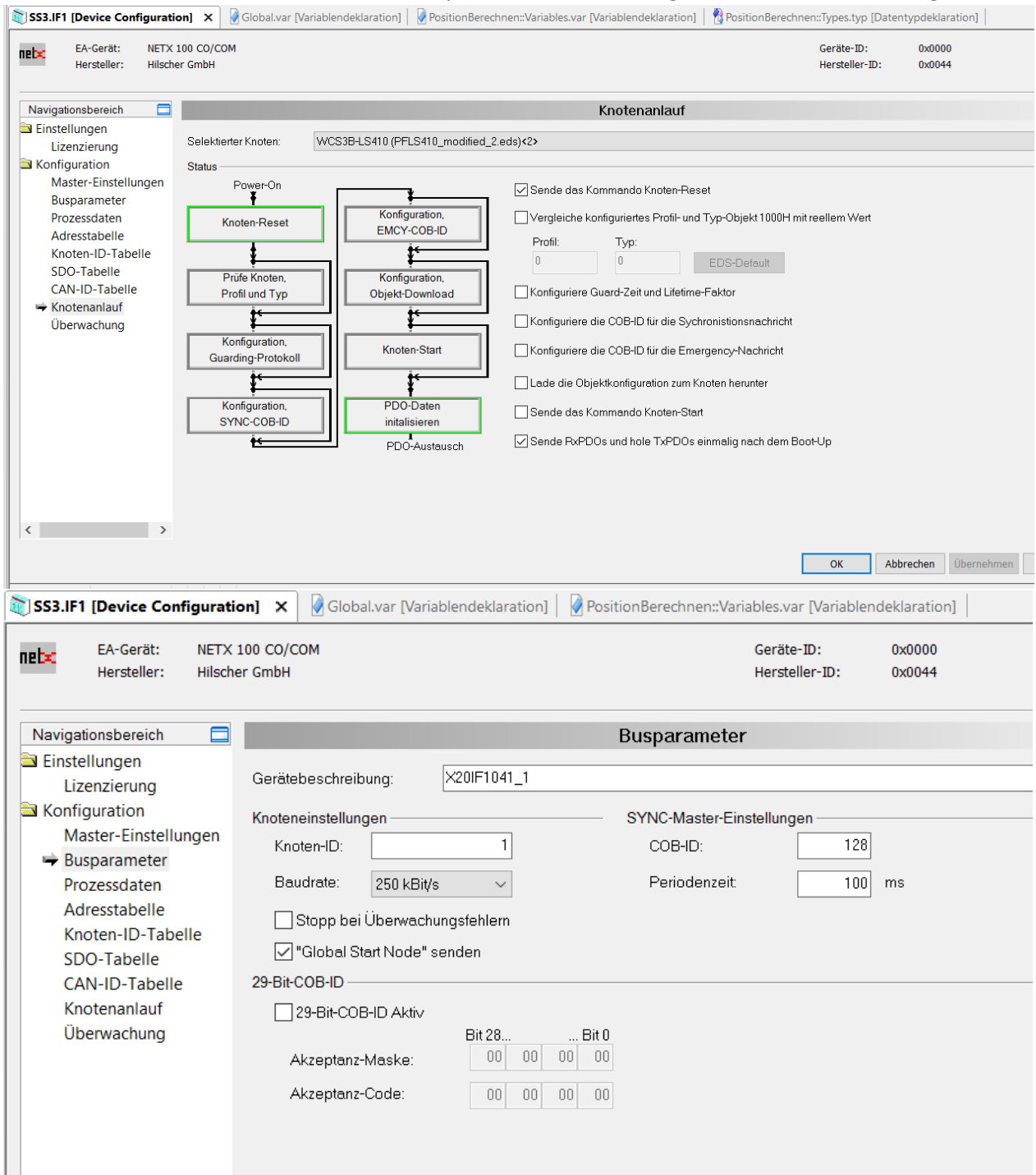
WCS3B, CANopen-Interface	
Pin	Benennung
1	n. c.
2	UB+
3	GND
4	CAN-H
5	CAN-L

Vollständige Vorgehensweise zur Einbindung in AS an die SPS, wie im Test ausgeführt

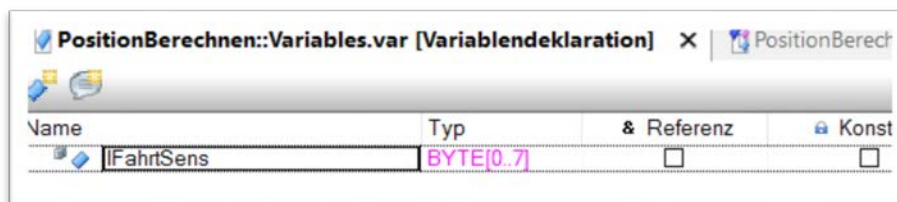
1. Neues Projekt, Hardware identifizieren
2. Falls EDS File vom can Sensor noch nicht hinzugefügt
 - a. "PFLS420_modified.eds" öffnen (das ist auch im BuR Ordner in /Dropbox/Basisprogramm hinterlegt) und kontrollieren ob der Wert der Zeile „FileName=.....“ dem Filename entspricht.
 - b. In AS → Extras/DTM Gerät hinzufügen
 - c. "PFLS420_modified.eds" auswählen
 - d. Jetzt DTM des CAN Masters markieren und auf Hardware hinzufügen
 - e. Doppelklick im Auswahlfenster auf den dort erschienenen Sensor.
3. Knotenpunkt auf 2 anpassen und kontrollieren, dass es wie auf Foto aussieht



4. Can Master Einstellen (Rechtsklick auf „CANopen DTM“ Gerätekonfiguration) wie auf Abbildung



5. Lokale Variable hinzufügen



6. Globale Variablen hinzufügen

Name	Typ	Konstante	Retain	Duplizierbar	Wert	Bezeichnung
posIst	REAL[0..8]	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0.8(0.0)	
SensorOut	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	TRUE	

posIst ist an das Robokran Projekt angepasst, nur das erste Datenfeld wird eigentlich benötigt.

SensorOut ist eine Flanke um anzuzeigen, dass außerhalb des Lineals ist.

7. Variable dem Modul zuweisen

Symbolname	Prozessvariable	Datentyp	Taskklasse
ModuleOk		BOOL	
2001sub00_TPD001_Data_to_SPS	PositionBe:IFahrtSens	OCTET[8]	Automatisch

8. C++ Programm, rechnet Wert aus und Setzt eine Flag „SensorOut“ wenn der Sensor aus dem Lineal ist oder ein anderer Fehler ist. Der posIst Wert bleibt dabei auf dem letzten erfolgreichen gelesenen Wert stehen.

```
// Wenn Sensor außerhalb des Lineals, dann SensorOut Flanke 1, sonst 0 und posIst[0] wird
berechnet
if (lFahrtSens[3] != 0 && lFahrtTest)

else
{
    //Can Absolut Sensor (Pepperl und Fuchs) Positionsberechnung
    posIst[0] = (lFahrtSens[2] + lFahrtSens[1] * 255 + lFahrtSens[0] * 65025 /*
255^2*/)/12.5;
    SensorOut = 0;
}

}
```

9. Draufspielen, funktioniert (hoffentlich ;))