

Application Note MotorCtrl Projekt

Anwendungshinweise und Einarbeitungshilfen

Tobias Wurm

Dieses Dokument beschreibt die, im Rahmen des MotorCtrl Projekt, verwendete Hard- und Software, deren Installation, Einstellung und Benutzung, sowie Funktionsweise und soll dem Leser eine schnelle Einarbeitung in die Thematik ermöglichen.

Inhaltsverzeichnis

Einführung	3
Komponenten- und Worterklärungen.....	3
Hardware	3
Software	4
Aufbau	4
Code Composer Studio	6
T.I. Kit Software	7
Initialisierung	8
CAN Kommunikation des T.I. Kit.....	11
Steuerungs-PC Software.....	13
MotorCtrl.....	13
Programmablauf MotorCtrl.....	13
CAN Repository.....	23
Linux Treiberinstallation für CAN Interface.....	23
PCAN View	24
Motoren starten und stoppen.....	24
Eigenschaften der CAN Kommunikation	25

Einführung

Dieses Dokument soll dem Leser Hilfestellung bei der Einarbeitung in das, im Wintersemester 2012/13 im Rahmen des Masterstudiengangs AUT, durchgeführte Projekt mit dem Namen MotorCtrl bieten. Im Rahmen dieses Projekts wurde auf einer Motoren Entwicklungsplattform von Texas Instruments eine Feldorientierte Regelung implementiert, die in der Lage ist über einen CAN Bus zu kommunizieren. Im Weiteren wurde ein, auf Linux basierendes, Programm entwickelt, welches als User Interface verschiedene Möglichkeiten bietet einen, mit der Entwicklungsplattform, verbundenen Motor zu steuern.

Im Folgenden finden sich Beschreibungen der verwendeten bzw. erstellten Hard- und Software, deren Installation, Einstellung und Benutzung, sowie Erläuterungen zu der Funktionsweise der einzelnen Module und deren Initialisierung.

Ein weiteres Dokument mit dem Namen *Implementierung einer Feldorientierten Regelung und Bestimmung der Maschinenparameter*, welches ebenfalls im Rahmen des MotorCtrl Projekts erstellt wurde, beschäftigt sich mit den theoretischen Grundlagen der Regelung. Desweiteren wird ein in Matlab/Simulink erstelltes Modell des Systems erläutert.

Komponenten- und Worterklärungen

Im folgenden Abschnitt sollen zunächst die wichtigsten Komponenten und Begriffe des Projekts erläutert werden.

Hardware

Steuerungs-PC: Auf diesem PC läuft das Programm *MotorCtrl* welches Funktionen zur Ansteuerung des T.I. Kits über einen CAN Bus bietet. An den Steuerungs-PC wird der PCAN-USB-Adapter angeschlossen.

T.I. Kit: Eine von Texas Instruments angebotene Entwicklungsplattform die Leistungselektronik und Rechenleistung zur Ansteuerung von Motoren bereitstellt. Detaillierte Informationen zur Ausstattung und Bedienung des Kits finden sich im Ordner Dokumente\T.I. Kit in den Dateien *HVMotorCtrl+PFC_HWGuide* und *HVMotorCtrl+PFCKit_HowToRunGuide*.

Motorensatz: Besteht aus zwei identische Asynchronmaschinen mit Käfigläufern die sich gegenüber stehen und mit einer ausgerichteten Welle verbunden sind. Siehe Abbildung 5.

PCAN-USB-ADAPTER: Lässt sich an den USB Port des Steuerungs-PC anschließen. Der 7-polige D-Sub Stecker dient als Anschluss an den CAN Bus. Weiterführende Informationen zum Anschluss des Adapters finden sich im Ordner Dokumente\CAN in den Dateien *PCAN-USB_UserMan_eng* bzw. *PCAN-USB_UserMan_deu*.

CAN Kabel: Überträgt das CAN-HIGH und CAN-LOW Signal aus deren Informationen sich die CAN Nachrichten ergeben.

Drehimpulsgeber: In einem der Motoren ist ein Drehimpulsgeber der Firma Thalheim verbaut. Ein passendes Datenblatt zu dem Gerät findet sich im Ordner Dokumente\Klemmenplan. Weitere Informationen zur Funktionsweise finden sich auf dem Drehimpulsgeber selbst.

USB A/B Kabel: Stellt die Verbindung zum Code Composer Studio her und wird nur im Entwicklungsstadium benötigt.

15V-Netzteil: Versorgt die 15V Komponenten im T.I. Kit.

Netzspannung: Während der Entwicklungsphase empfiehlt es sich das T.I. Kit an einen Stelltransformator anzuschließen um die Zwischenkreisspannung im T.I. Kit einzustellen. Dies bietet die Möglichkeit Systemänderungen auf sicheren Spannungsleveln zu testen und die Spannungsversorgung bei Bedarf schnell zu unterbrechen.

Software

HVACICAN: Das Programm in dem die Feldorientierte Regelung und die CAN Kommunikation realisiert ist. HVACICAN läuft auf der Piccolo F2803x controlCard welche im T.I. Kit verbaut wurde. Aufgrund von Abhängigkeiten zu unterlagerten Softwarekomponenten konnten einige Dateien nicht mit HVACICAN_<...> benannt werden und somit HVACI_Sensored_<...>.

MotorCtrl: Das Programm läuft auf dem Steuerungs-PC und bietet dem Benutzer ein Interface um das T.I. Kit, bzw. den daran angeschlossenen Motor, zu steuern und zu überwachen.

Code Composer Studio: Eine auf Eclipse basierende Entwicklungsumgebung von Texas Instruments welche benutzt wurde um im Laufe des Projekts das HVACICAN Programm zu entwickeln.

Aufbau

Die beiliegenden Dokumente *HVMotorCtrl+PFC_HWGuide* und *HVMotorCtrl+PFCKit_HowToRunGuide* im Ordner Dokumente\T.I. Kit geben Informationen zum Aufbau und zur Verkabelung des T.I. Kits. Auch die Abschnitte über das richtig setzen der Jumper und des Bananenkabels innerhalb des T.I. Kits sollten beachtet werden.

Abbildung 3 zeigt die richtige Ausrichtung des CAN Kabels, sowie der 15V Spannungsversorgung und des USB Kabels. Abbildung 4 zeigt den Stecker des Drehimpulsgebers. Weitere Informationen zum Drehimpulsgeber und dessen Anschlüssen finden sich im Ordner Dokumente\Klemmenplan in der Datei Motoranschlußklemmenplan.

Abbildung 1 zeigt das Motorklemmbrett sowie die Klemmleiste. In Abbildung 2 ist selbige Klemmleiste lediglich vergrößert, so dass die Belegung einfacher zu erkennen ist. Ein Anschlussplan der Klemmleiste findet sich im Ordner Dokumente\Klemmenplan.

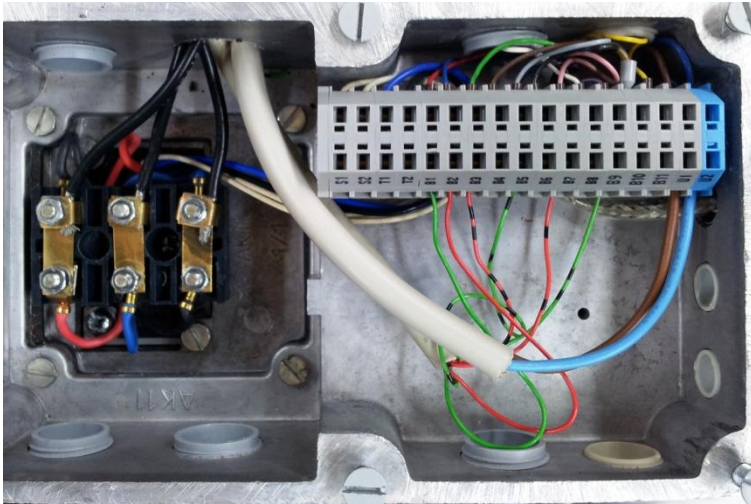


Abbildung 1: Motorklemmbrett und Klemmleiste

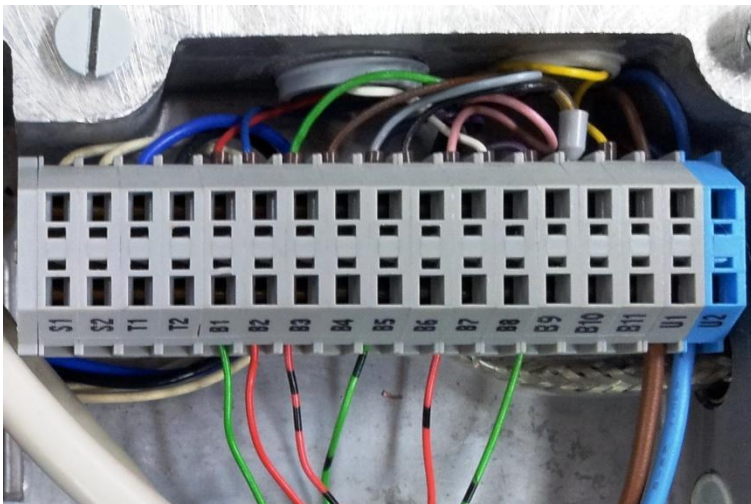


Abbildung 2: Klemmleiste



Abbildung 3: CAN-, 15V- und USB-Anschluss am T.I. Kit

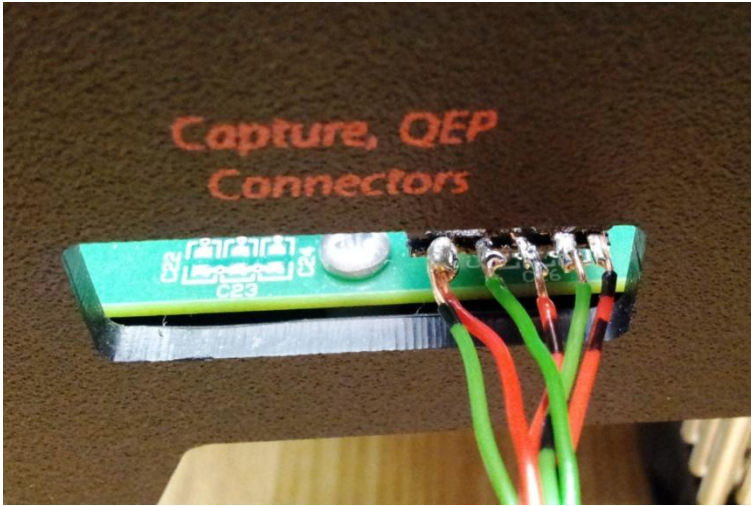


Abbildung 4: Drehimpulsgeber-Anschluss am T.I. Kit

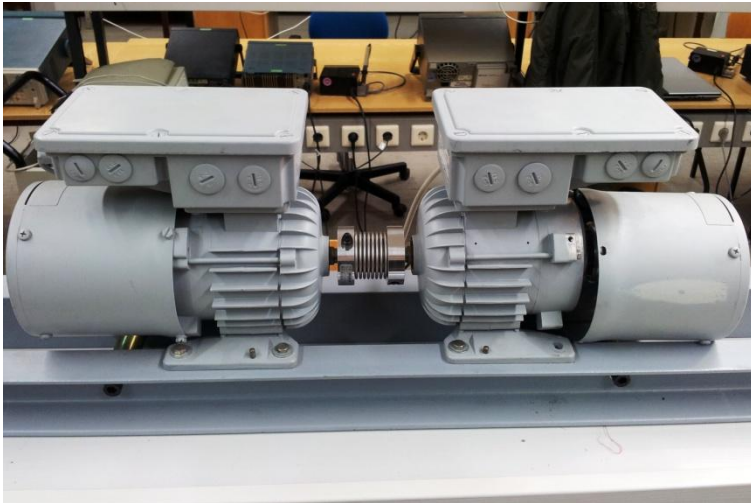


Abbildung 5: Motorensatz

Code Composer Studio

Das Code Composer Studio ist eine von Texas Instruments gelieferte Entwicklungsumgebung die innerhalb des Projekts in Version 4 genutzt wurde.

In dem Dokument *HVMotorCtrl+PFCKit_HowToRunGuide* findet sich ab Seite 9 eine Anleitung zur Installation und Einstellung von Code Composer Studio.

Die Schritte 1 bis 11 sind allgemeingültig und sollten in dieser Reihenfolge abgearbeitet werden, unabhängig davon welches Projekt letztendlich geladen wird. Die Schritte 12 bis 14 beziehen sich speziell auf T.I. Beispielprojekte. Schritte 15 bis 19 zeigen wie sich der Debugger starten lässt und sind wiederum für sämtliche Projekte gültig. Auch die Schritte 20 und 22 bis 25 sind allgemeingültig.

Das HVACICAN Projekt, welches im Verlauf des MotorCtrl Projekts entwickelt wurde, findet sich im beiliegenden Ordner Software. Dieser Ordner (HVACICAN) muss in das Verzeichnis

...\ti\controlSUITE\development_kits\HVMotorCtrl+PfcKit_v1.7 kopiert werden um im Schritt 8, oder nach Beendigung der Installation, geladen zu werden.

T.I. Kit Software

Der Algorithmus zur Regelung der Asynchronmaschine läuft auf der Piccolo F2803x controlCard die in das T.I. Kit gesteckt wird. Zusätzlich zur Berechnung des reinen Regelalgorithmus muss sich der Controller um die Verwaltung der CAN Kommunikation kümmern. Beide Funktionalitäten sind in dem beiliegenden Software Projekt HVACICAN für das Code Composer Studio enthalten.

Der Programmcode zur Regelung der Asynchronmaschine ohne CAN Kommunikation wurde aus einem Texas Instruments Beispielprojekt, welches zusammen mit dem T.I. Kit ausgeliefert wird, entnommen und modifiziert. Dieses Beispielprojekt, mit dem Namen HVACI_Sensored findet sich nach der Installation des T.I. Programms controlSUITE im Systempfad

...\ti\controlSUITE\development_kits\HVMotorCtrl+PfcKit_v1.7.

Zur Inbetriebnahme des Projekts liefert T.I. eine Step-by-Step Anleitung die den Benutzer durch die einzelnen Funktionen der Software führt und diese inkrementell, durch verschiedene Level, zur vollen Funktionalität zusammenbaut. Es wird empfohlen dieses Dokument als Einarbeitung in das hier vorgestellte Projekt, als auch in die Feldorientierte Regelung (FOR) allgemein zu nutzen. Die Anleitung findet sich im Ordner Dokumente\FOC in dem Dokument *Sensored FOC of ACI* oder nach Installation von controlSUITE im lokalen Verzeichnis unter

...\ti\controlSUITE\development_kits\HVMotorCtrl+PfcKit_v1.7\HVACI_Sensored\~Docs. Die dort vorhandenen Beschreibungen liefern einen guten Überblick über die Arbeitsweise der FOR in der Software.

Zusätzlich zu oben genanntem finden sich in den Dateien des HVACICAN Projekts zahlreiche Anmerkungen, Hinweise und Erklärungen die im Laufe des MotorCtrl Projekts in Form von Kommentaren dem Code beigelegt wurden.

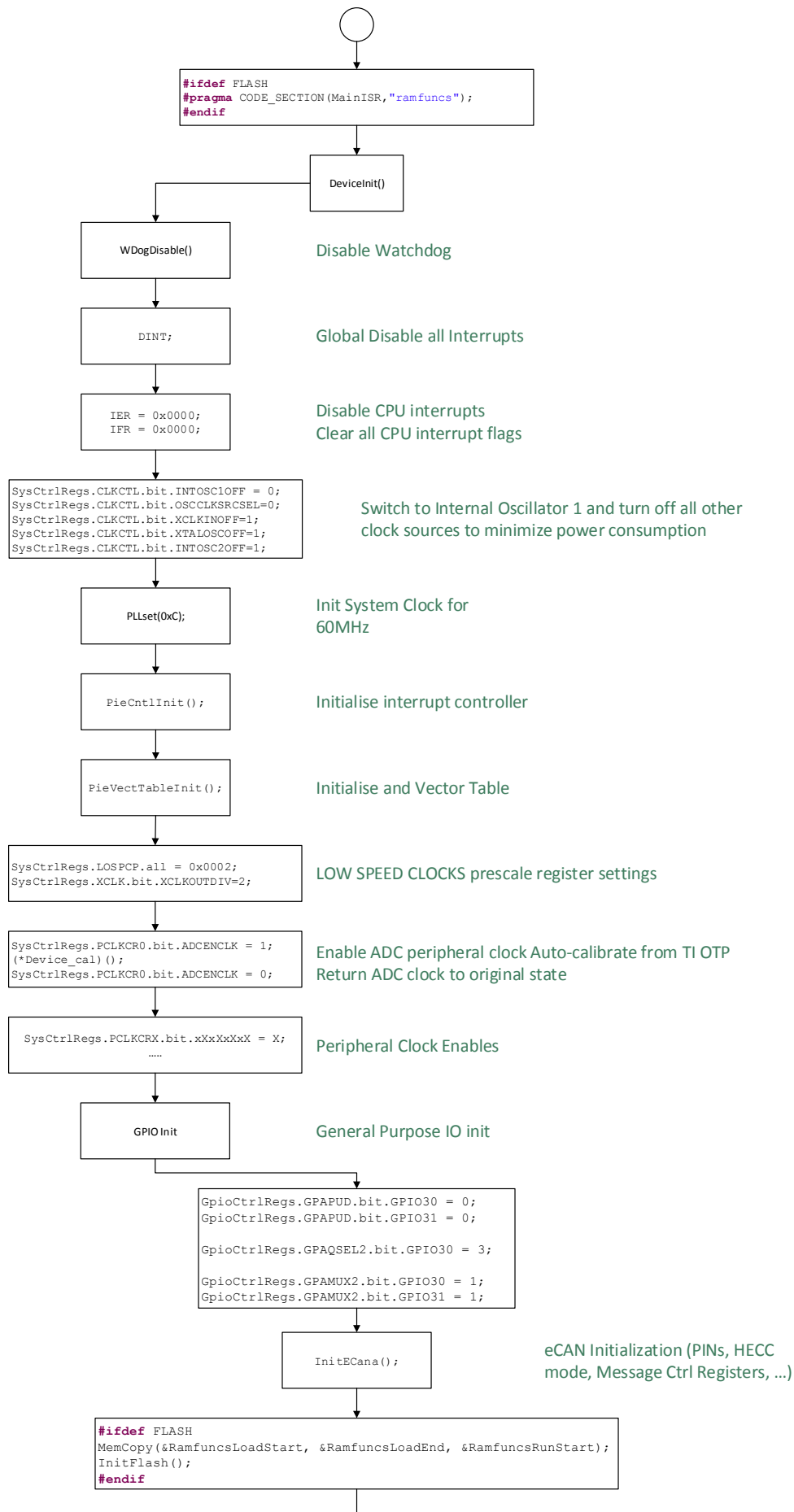
Die genannten Quellen liefern lediglich Informationen zur Software der FOR, jedoch nicht zur CAN Kommunikation. Diese soll deshalb im Abschnitt CAN Kommunikation des T.I. Kit noch genauer betrachtet werden.

Der Ordner Dokumente\HVACICAN Doxygen enthält eine Dokumentation des HVACICAN Programms welche mit dem Tool Doxygen erstellt wurde. In Form einer HTML Seite werden hier umfangreiche Informationen zum Quellcode bereitgestellt. Zum Öffnen der Dokumentation muss eine der Dateien mit Endung .html in einem Internetbrowser geöffnet werden.

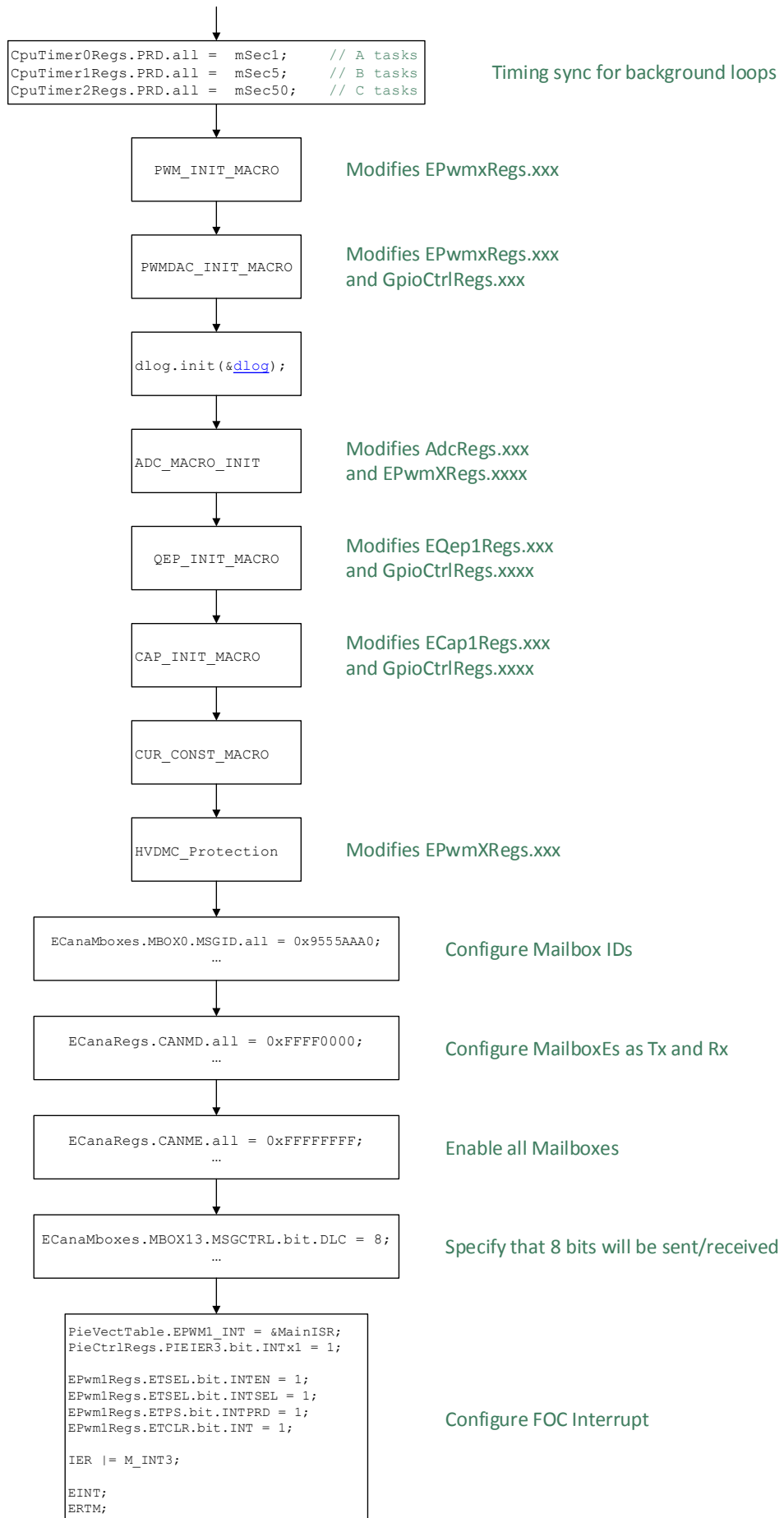
Initialisierung

Die korrekte Initialisierung des Prozessors ist der Erste und nicht zu unterschätzende Schritt bevor das eigentliche Benutzerprogramm ablaufen kann. Da Fehler in der Initialisierung häufig zu schwer nachvollziehenden Problemen führen, soll im folgenden Diagramm der Ablauf dieser Initialisierungen gezeigt werden.

Application Note MotorCtrl



Application Note MotorCtrl



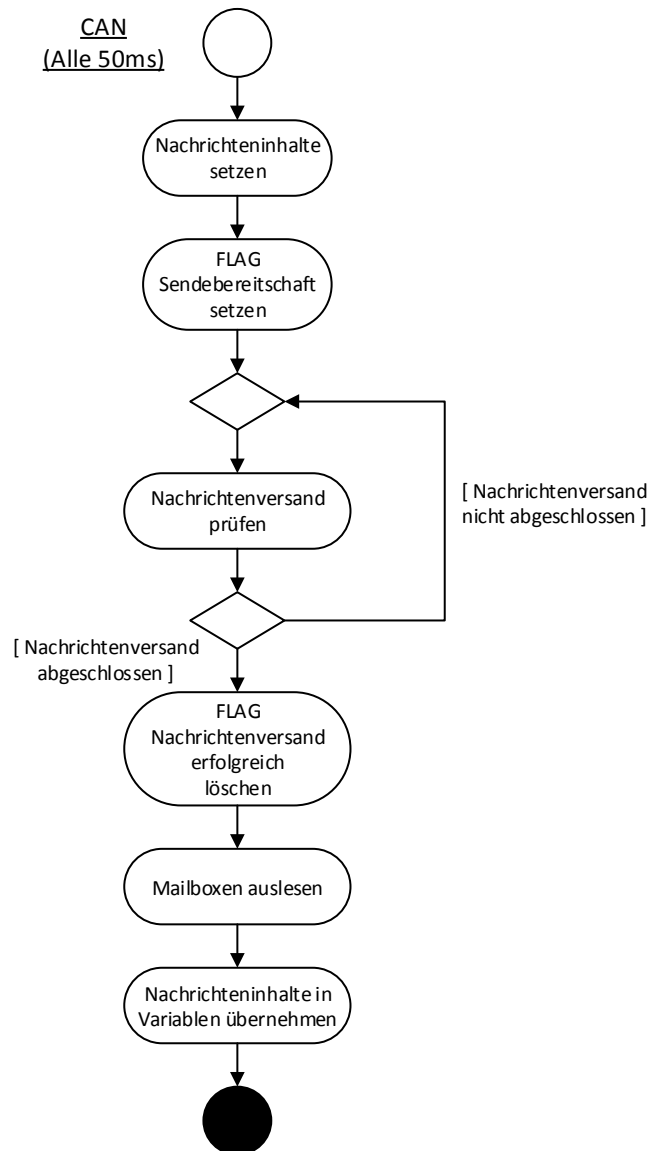
CAN Kommunikation des T.I. Kit

Die Feldorientierte Regelung stellt hohe Ansprüche an den Determinismus des Programms. Es ist also wichtig, dass die Programmabschnitte zur Berechnung der Regelung in äquidistanten Zeitabschnitten Rechenzeit bekommen. Um dies sicher zu stellen verwendet HVACICAN einen Timer-gesteuerten Interrupt in dem die neuen Sollwerte berechnet werden.

Beim jetzigen Projektstand sind, im Gegensatz zur FOR, bei der CAN Kommunikation kaum zeitliche Begrenzungen vorhanden. Gemeint ist hiermit die Zeit in der beispielsweise ein neuer Geschwindigkeitssollwert empfangen wird, oder die aktuelle Geschwindigkeit an den Steuerungs-PC versendet wurde, nicht die Zeitbedingungen die sich auf der Kommunikationshardware Ebene ergeben und durch die jeweiligen CAN Interfaces eingehalten werden müssen.

Aufgrund dieser Tatsache wird im HVACICAN eine Art Scheduling System verwendet, welches in regelmäßigen Abständen die Funktionen zur CAN Kommunikation ausführt. Durch eine Kombination von interruptgesteuerter Bearbeitung, für die FOR, und Rechenzeit-Scheduling, für CAN, kann den zeitlichen Anforderungen der verschiedenen Programmelemente Rechnung getragen werden.

Der Ablauf der CAN Kommunikation ist in folgendem Ablaufdiagramm dargestellt.



Steuerungs-PC Software

MotorCtrl

MotorCtrl ist neben dem Projektnamen auch der Name eines C basierenden Programms, welches im Rahmen des Projekts auf Ubuntu 12.10 geschrieben wurde um die Steuerung des T.I. Kit über den CAN Bus zu vereinfachen. Im Programm lässt sich ein Drehzahlsollwert vorgeben und die aktuelle Drehzahl des Motors auslesen, außerdem können die Parameter der PID Regler angezeigt werden. Das Programm läuft auf dem Steuerungs-PC.

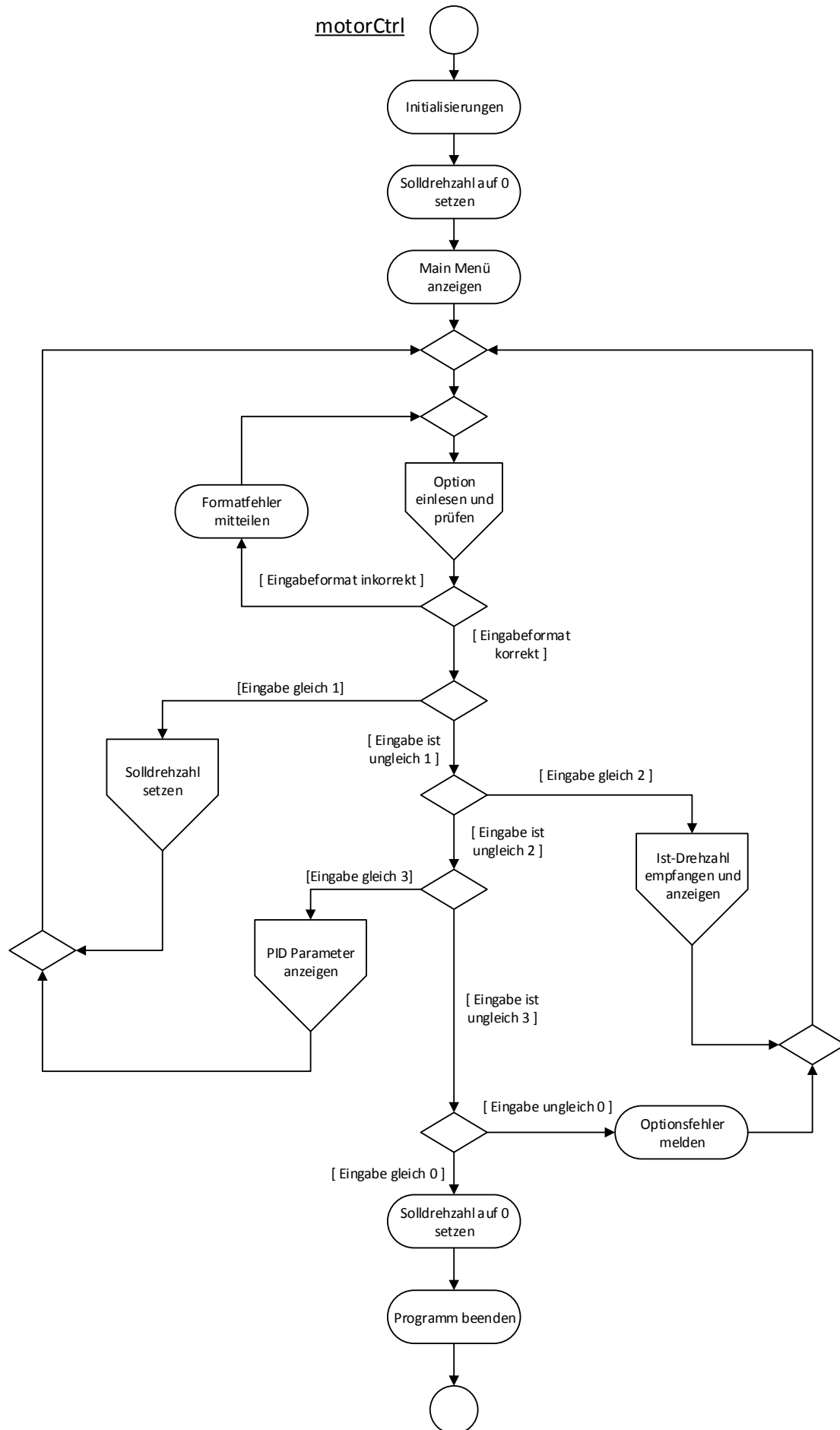
Das Programm MotorCtrl findet sich im beiliegenden Ordner *Software*. Es empfiehlt sich die Source Files auf dem eingesetzten Steuerungs-PC zu kompilieren. Dies kann mit Hilfe des Befehls *make* geschehen.

Starten lässt sich das Programm nach einem Wechsel in das entsprechende Verzeichnis mit dem Befehl *./MotorCtrl*. Vor dem Start des Programms muss der PCAN-USB-Adapter installiert und gestartet werden. Informationen hierzu finden sich im Abschnitt Linux Treiberinstallation für CAN Interface.

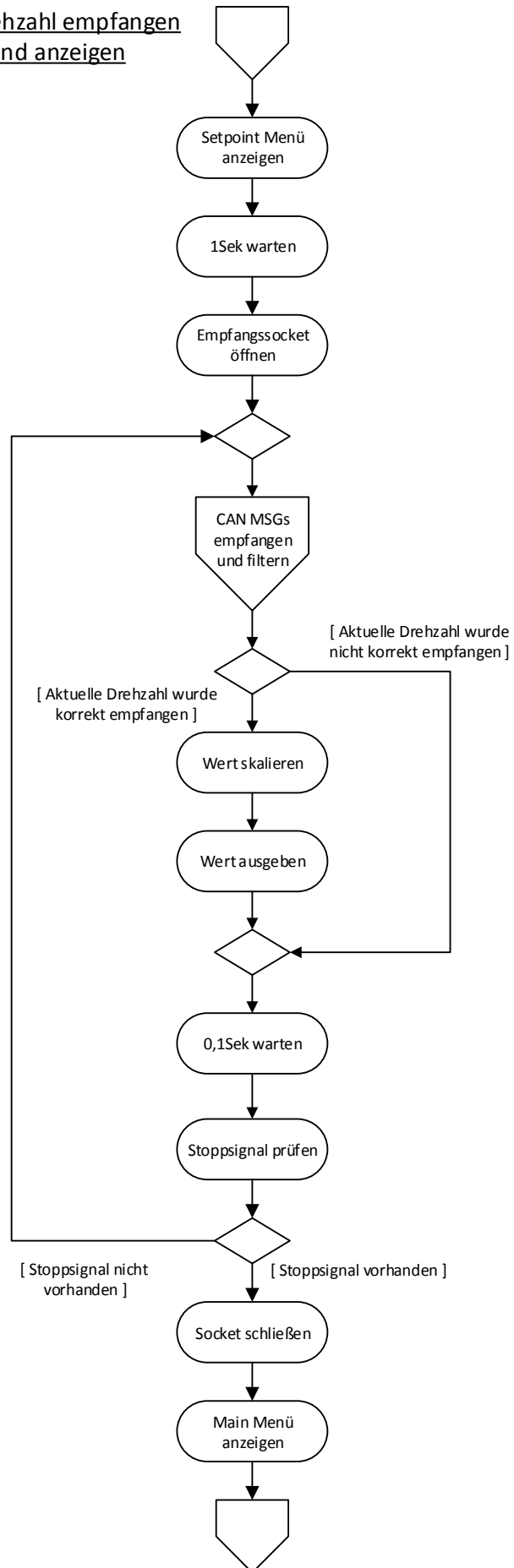
Programmablauf MotorCtrl

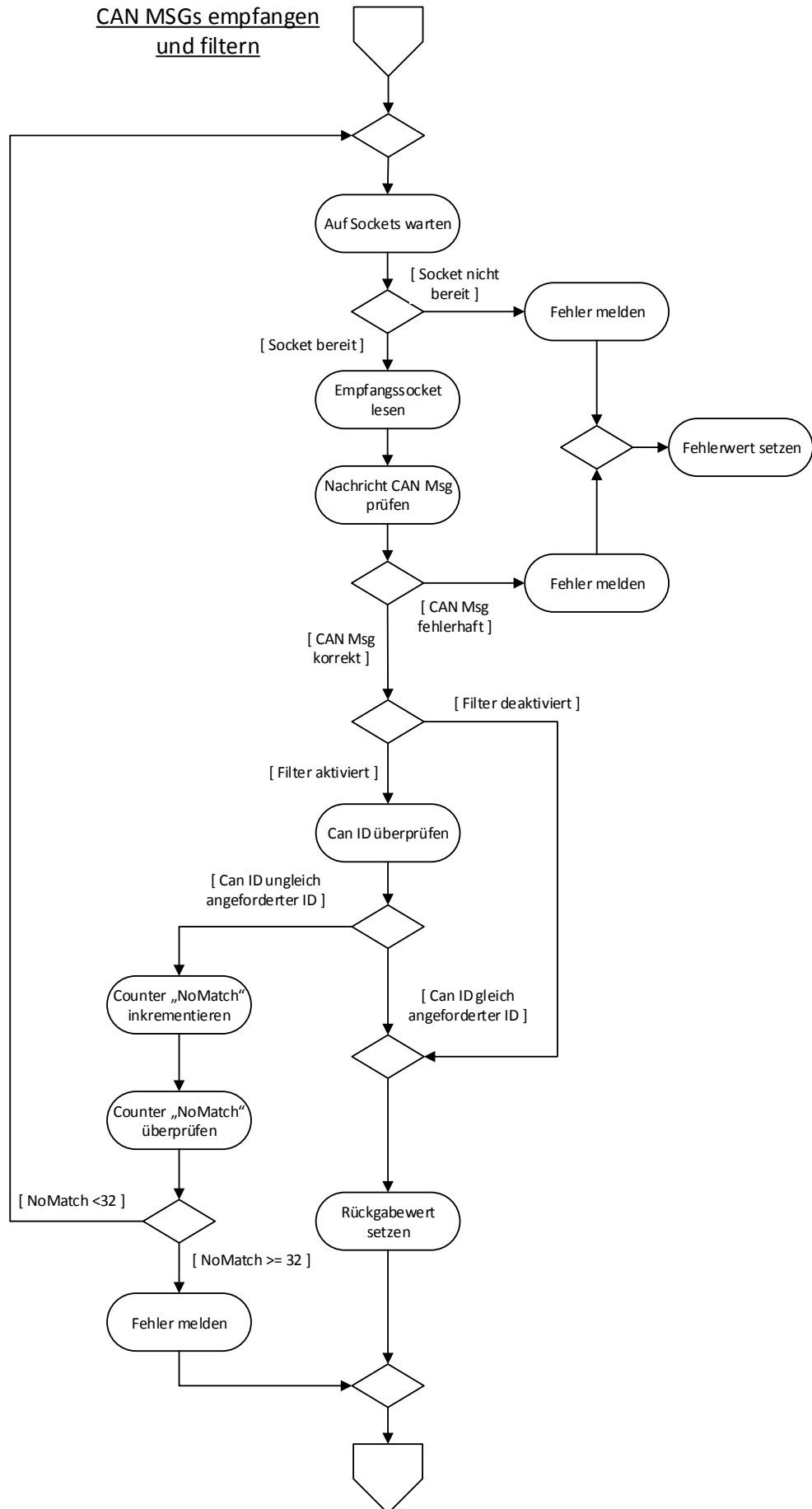
Bei der Entwicklung des Programmcodes wurde auf eine hohe Wiederverwendbarkeit geachtet. So sind beispielsweise Sende- und Empfangsfunktionen sehr allgemein und damit flexibel gehalten. Die nachfolgenden Aktivitätsdiagramme beschreiben die wesentlichen Programmschritte und sollen Hilfestellung bei der Einarbeitung und Weiterentwicklung des Programms bieten.

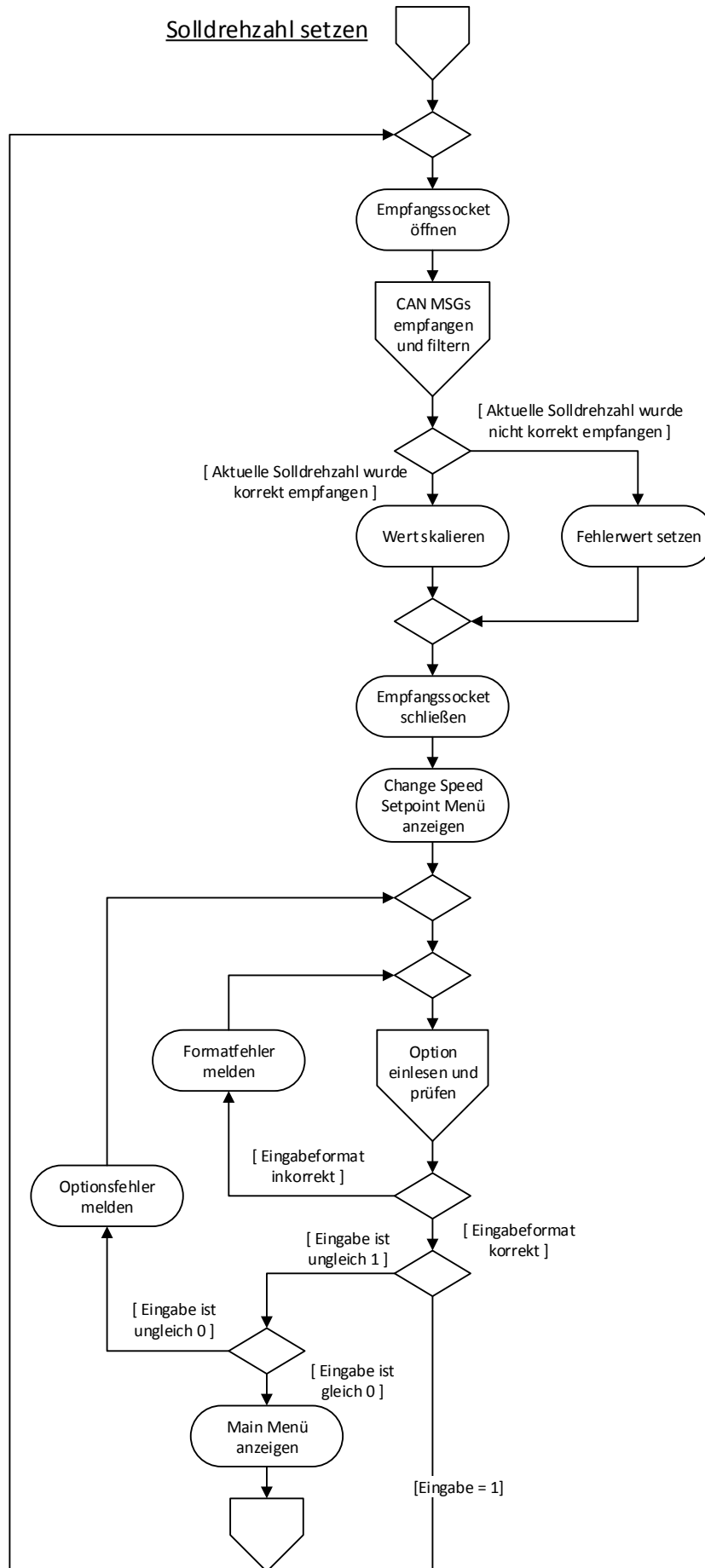
Neben den, an dieser Stelle vorhandenen, Informationen, finden sich zahlreichen Anmerkungen und Erklärungen in Form von Kommentaren in den Source- und Header Files des Projekts.

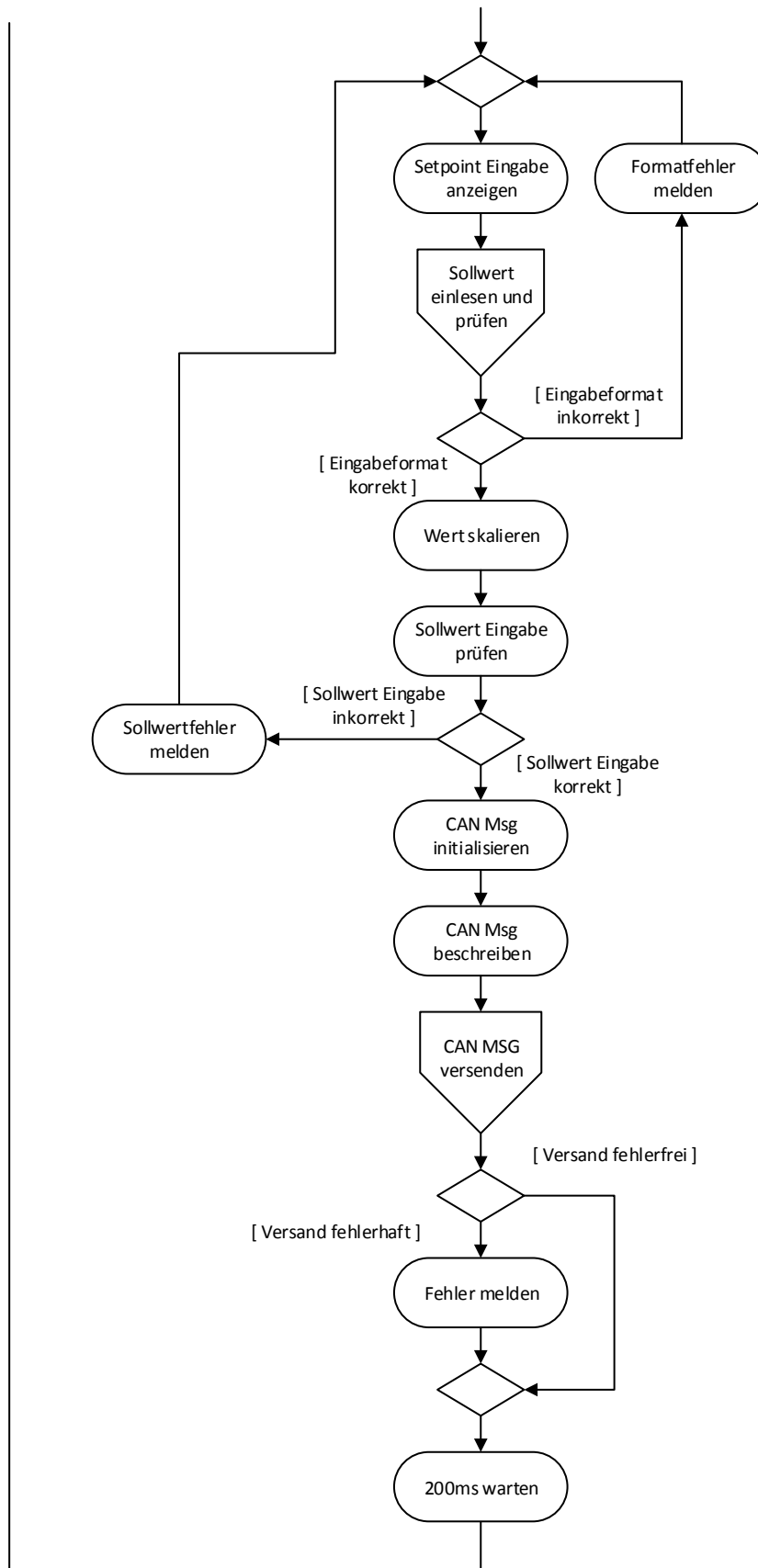


Ist-Drehzahl empfangen
und anzeigen

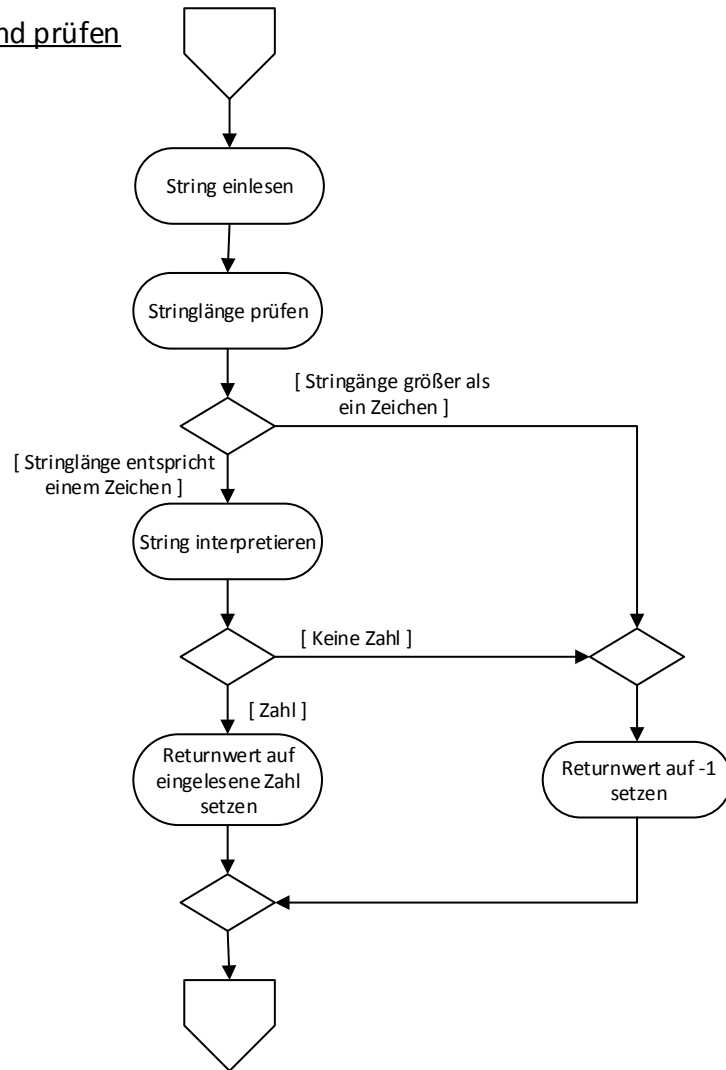




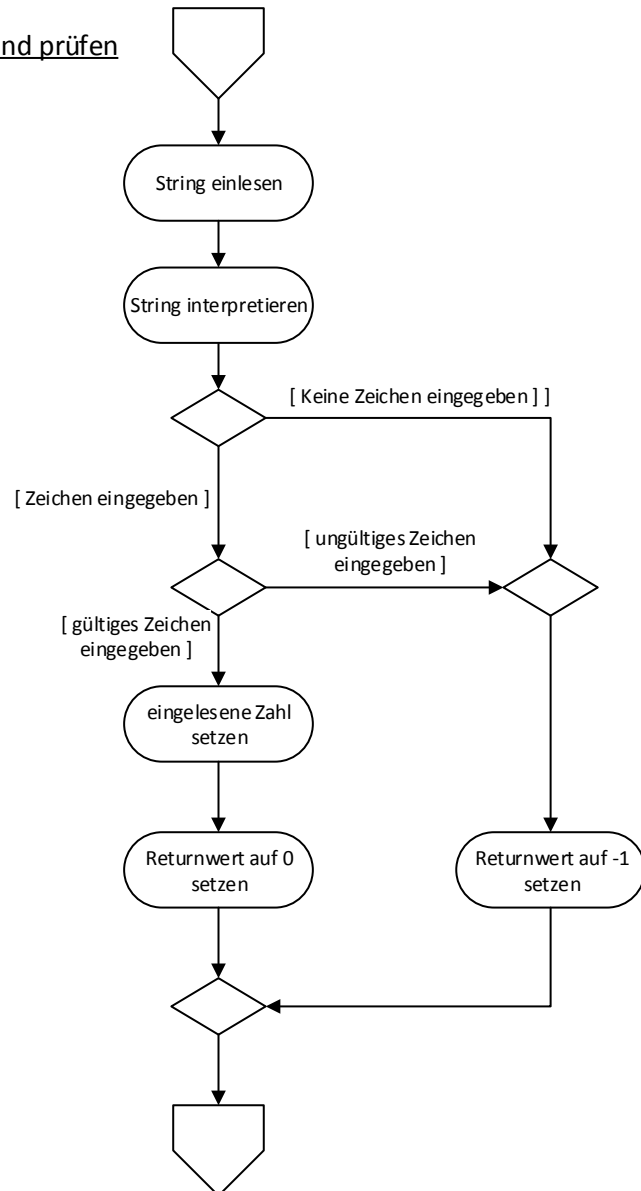




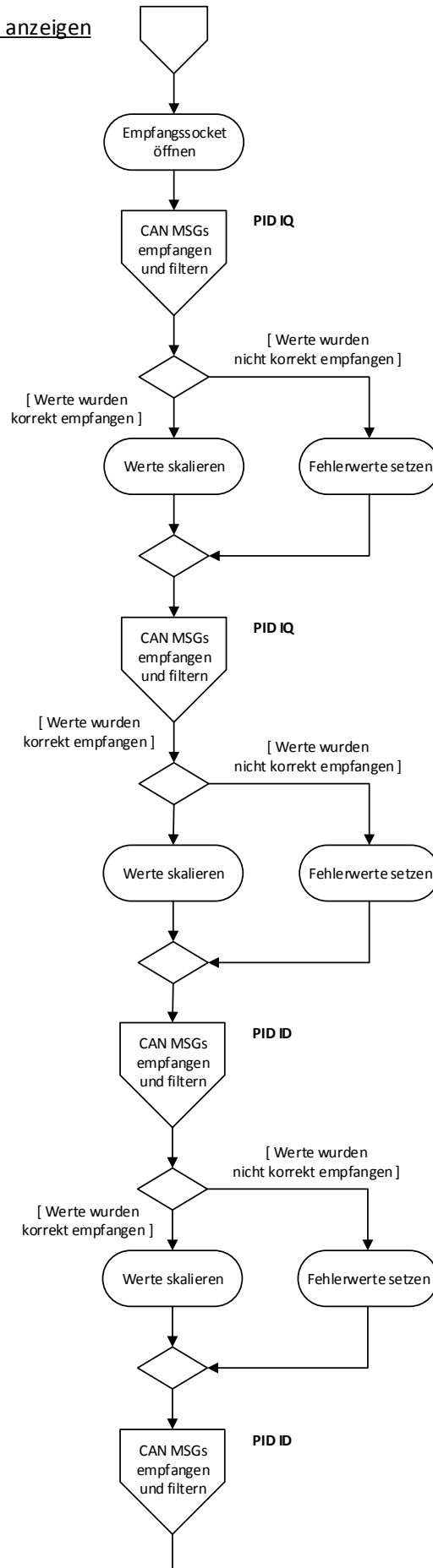
Option einlesen und prüfen



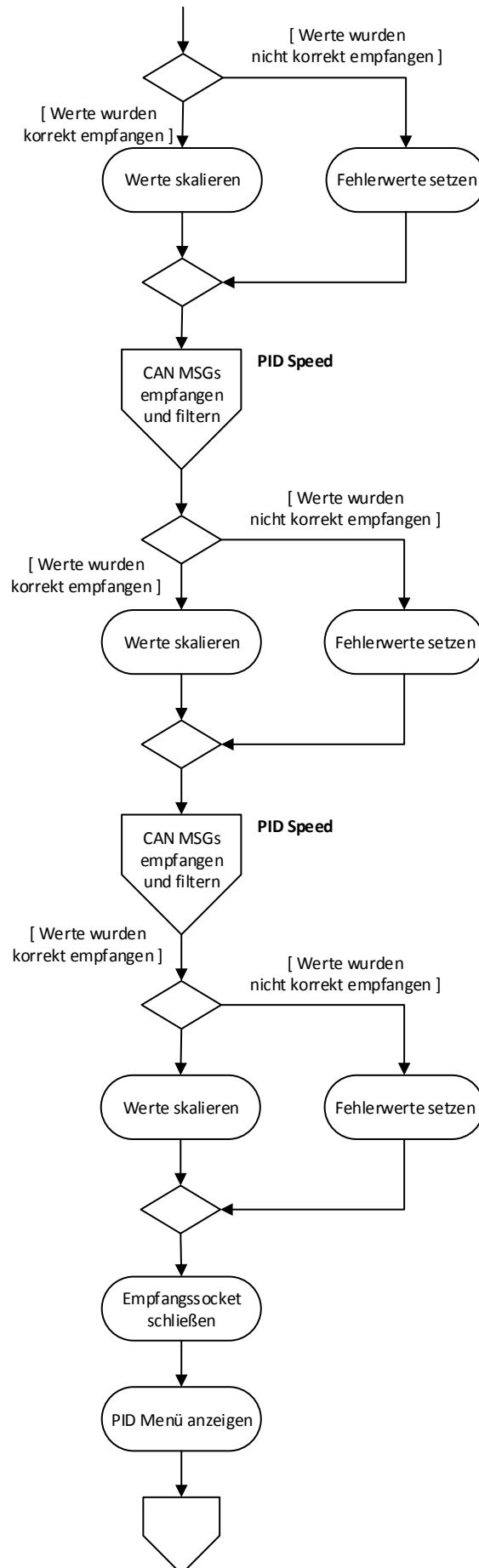
Sollwert einlesen und prüfen



PID Parameter anzeigen



Application Note MotorCtrl



CAN Repository

Teile des Programms MotorCtrl basieren auf einem Open Source CAN Bus Projekt der Volkswagen AG, welches verschiedene Programme zur Parametrierung und Nutzung von CAN Netzwerken bietet. Das Projekt kann über ein GIT Repository bezogen werden. Der entsprechende Befehl lautet *git clone git://anonscm.debian.org/collab-maint/can-utils.git*

Nach dem Durchlauf des Befehls findet sich eine Kopie des Projektordners am angegebenen Speicherort. Der Ordner ist zum Bau eines .deb Pakets vorbereitet, die jetzige Version scheint jedoch fehlerhaft zu sein da entsprechende Programme zum Paketbau sich mit Fehlern beenden. Es kann jedoch der Befehl *make* ausgeführt werden mit dem man alle relevanten Source- und Header Files erhält, die sich anschließend ausprobieren und modifizieren lassen. Der reduzierte Funktionsumfang dieser Programme kann die Einarbeitung in die CAN Kommunikation des Steuerungs-PCs vereinfachen.

Linux Treiberinstallation für CAN Interface

Um als Teilnehmer im CAN Bus integriert zu werden benötigt der verwendete Steuerungs-PC eine entsprechende Hardwareschnittschelle. Hierzu wird ein PCAN-USB-Adapter der Firma Peak verwendet (<http://www.peak-system.com/PCAN-USB.199.0.html>) der per USB Schnittstelle angeschlossen wird.

Bevor das Gerät verwendet werden kann, muss der entsprechende Treiber installiert werden. Abhängig vom Betriebssystem kann eine manuelle Installation relativ aufwendig werden. Da die benötigten Treiber ab der Linux Kernel Version 3.4 bereits mitgeliefert werden wird empfohlen den Kernel des Steuerungs-PC, falls notwendig, auf eine Version ≥ 3.4 zu aktualisieren.

Nach der Aktualisierung des Kernels wird der PCAN-USB-Adapter bei Anschluss an den Steuerungs-PC erkannt und die benötigten Treibermodule geladen. Zur Überprüfung lässt sich durch den Befehl *lsmod* eine Liste der geladenen Module anzeigen. Nach Anschluss des PCAN-USB-Adapters sollten sich in der Liste die Einträge *can_dev* und *peak_usb* finden. Weitere Informationen zur Treiberinstallation finden sich im Ordner Dokumente\CAN in der Datei *PCAN Driver for Linux_eng_7.1* oder auf der PEAK Linux Homepage (<http://www.peak-system.com/fileadmin/media/linux/index.htm#download>).

Der PCAN-USB-Adapter muss vor der Verwendung (also auch vor dem Start von MotorCtrl) aktiviert werden. Dies geschieht über den Befehl. *ip link set can0 up type can bitrate 1000000* . Die Zahl am Ende entspricht einer Übertragungsrate von 1Mbit/s. Diese Einstellung ist kompatibel zu MotorCtrl.

Soll der PCAN-USB-Adapter deaktiviert werden, muss der Befehl *ip link set can0 down* verwendet werden. Beide Befehle benötigen Root Rechte.

PCAN View

PCAN View ist eine von Peak System entwickelte Software mit der sich CAN Nachrichten anzeigen lassen die mit Hilfe eines PCAN-USB-Adapters empfangen werden. Zum Empfang von Nachrichten des T.I. Kits muss die Bit rate auf 1MBit/s eingestellt werden. Filter können entsprechend den im Abschnitt *Eigenschaften der CAN Kommunikation* dargestellten Identifiers gesetzt werden. PCAN View kann über die Homepage des Herstellers (www.peak-system.com) bezogen werden und kann bei der Entwicklung bzw. Modifikation des Programms sehr hilfreich sein.

Motoren starten und stoppen

Vor Ausführung der unten folgenden Schritte sollte das Code Composer Studio installiert und die Treiber des PCAN-USB-Adapters vorhanden sein.

System starten

Schritt 1: System, wie im Abschnitt Aufbau beschrieben, aufbauen.

Schritt 2: HVACICAN, wie im Abschnitt Code Composer Studio beschrieben, starten.

Schritt 3 (optional): Mit PCAN View den korrekten Versand von CAN Nachrichten prüfen. Hierbei können, wie im Abschnitt PCAN View beschrieben, CAN Nachrichten aus dem T.I. Kit aufgezeichnet werden.

Schritt 4: PCAN-USB-Adapter, falls noch nicht geschehen, anschließen und aktivieren (*ip link set can0 up type can bitrate 1000000*).

Schritt 5: MotorCtrl, wie im Abschnitt MotorCtrl, beschrieben starten.

Schritt 6: 230V Energieversorgung zum T.I. Kit herstellen.

System stoppen

Schritt 7: MotorCtrl beenden.

Schritt 8: 230V Energieversorgung zum T.I. Kit unterbrechen.

Eigenschaften der CAN Kommunikation

Die Kommunikationsgeschwindigkeit zwischen T.I. Kit und Steuerungs-PC beträgt 1MBit/s. Bei kurzen Leitungslängen sind, trotz der hohen Geschwindigkeit, keine Leitungsabschlüsse notwendig. Die untenstehende Tabelle gibt einen Überblick über die CAN Nachrichten die zwischen T.I. Kit und Steuerungs-PC ausgetauscht werden.

Identifizier	Übertragungsrichtung		Byte [0...3]	Byte [4...7]
	T.I. Kit	Steuerungs-PC		
0#1555AAA0	→		Aktuelle Geschwindigkeit	Neuer Geschwindigkeits Sollwert
0#1555AAA1	→		Life Sign	0x01.AB.45.DE
0#1555AAA2	→		id_pid - Kp	id_pid - Ki
0#1555AAA3	→		id_pid - Kd	id_pid - Kc
0#1555AAA4	→		iq_pid - Kp	iq_pid - Ki
0#1555AAA5	→		iq_pid - Kd	iq_pid - Kc
0#1555AAA6	→		speed_pid - Kp	speed_pid - Ki
0#1555AAA7	→		speed_pid - Kd	speed_pid - Kc
0#1555AAA0	←		Aktueller Geschwindigkeits Sollwert	Control Word