Trier University
of Applied Sciences

HOCHSCHULE TRIER

University of Applied Sciences Trier

Bachelor Projektarbeit

Entwicklung und Implementierung eines Can-Interfaces zur präzisen Lageregelung eines Portalkrans

bachelor of engineering

Elektrotechnik
Informationstechnologie und Elektronik

verfasst von:
Torsten Zimmermann. [966352]

Teamprojekt mit:
Steave Dahm. [969471]

betreut von:
Prof. Dr. Ing. Matthias Scherer
Andreas Reis, M. Sc.

Abgabedatum:
13. Oktober 2020

Eidesstattliche Erklärung

Ich versichere, die Bachelor-Projektarbeit selbstständig und lediglich unter Benutzung der angegebenen Quellen und Hilfsmittel verfasst zu haben.

Ich erkläre weiterhin, dass die vorliegende Arbeit noch nicht im Rahmen eines anderen Prüfungsverfahrens eingereicht wurde.

_____

Ort, Datum

Kurzfassung

Im Rahmen einer Bachelor Projektarbeit wird für den Portalkran Versuch des Regelungstechnik Labors eine Ansteuerung über ein Can Interface Entwickelt und Implementiert. Dabei übernimmt eine Regeleinheit namens Unicontrol, welche auf einer Studentischen Master Arbeit aufbaut, die komplette Regelung des Versuchs.Zudem besitzt die Laufkatze eine Zusatzplatine mit mehreren Sensoren und Aktoren, die in meinem Teil der Arbeit noch programiertechnisch implementiert werden mussten. Das Can Interface hat die Aufgabe Unicontrol, das Motorsteuergerät und die Platine, welche ebenfalls auf einer Studentischen Arbeit besteht mit einander kommunizier fähig zu machen.Hierbei musste darauf geachtet werden, das das Motorsteuergerät mit Can-Open Kommuniziert. Unicontrol und die zusatz Platine kommunizieren über Can-Automotive. Dabei wurden auch verschiedene Sicherheitsmechanismen implementiert, sodass niemand zu schaden kommen kann.

Die Ansteuerung soll über einen externen Rechner von Zuhause aus für Studierende möglich gemacht werden. Dies hat den Vorteil, das Studierende in Zukunft das Regelungstechnik-Labor absolvieren können auch über das Wochenende hinweg. Der komplette Aufbau ist im Anhang nochmal durch ein Bild kenntlich gemacht worden.
Zum besseren Verständnis, wird auch nochmal kurz auf den Part meines Teampartners eingegangen, der sich um den kompletten mechanischen Aufbau des Portalkrans gekümmert hat.

Abstract

As part of a bachelor projekt for the control engineering laboratory, a control via a CAN interface is being developed and implemented for the gantry crane attempt. A control unit called Unicontrol, which is based on a student master's thesis, takes over the complete control of the experiment. In addition, the trolley has an additional circuit board with several sensors and actuators, which in my part of the work still had to be implemented in programming. The Can Interface has the task of making Unicontrol, the engine control unit and the circuit board, which is also based on a student work, capable of communicating with each other. It had to be ensured that the engine control unit communicates with Can-Open. Unicontrol and the additional circuit board communicate via Can-Automotive. Various security mechanisms have also been implemented so that nobody can be harmed.

The control should be made possible for students from home via an external computer. This has the advantage that in future students will be able to complete the control engineering laboratory over the weekend. The complete structure is shown again in the appendix by a picture.

Inhaltsverzeichnis

1 Einleitung

1.1 Aufgabenstellung

Die zentrale Aufgabe dieser Projektarbeit war die Vernetzung der Regeleinheit Unicontrol zusammen mit dem Motorsteuergerät und der eigens Entwickelten Platine.

Die Zusatz Platine welche auf einer studentischen Arbeit aufbaut besitzt Sensoren und Aktoren. Vorhandene Sensoren sind der Linearencoder, der Winkelencoder und der Hall Sensor. Vorhandene Aktoren sind der Hubmagnet und der Servomotor. Diese Peripherie Module der zusatz Platine sollten zudem initialisiert und getestet werden. Dabei war besonders dadrauf zu achten, das der Linearencoder über die komplette Distanz keine Striche des Codierfilms überspringt. Kommuniziert wurde wie bei Unicontrol über Can-Automotive. Can hat unter anderem den Vorteil, das auf Ereignisse sofort Reagiert werden kann, sofern die Message ID hoch priorisiert ist. Die Informationen über den Winkelencoder und den Linearencoder werden sobald die Laufkatze initialisiert ist, alle 10 ms übertragen. Damit kann man das System als echtzeitfähig bezeichnet werden.

Die Regeleinheit Unicontrol sollte über die Umgebung Controldesk welche auch im Automotive Bereich genutzt wird, steuerbar sein. Weiter unten ist diese Umgebung nochmal gezeigt und erklärt. Zudem wird der Regler in Software implementiert. Dies ist möglich indem in Simulink der Regler aufgebaut wird und in C-Code kompiliert wird und dem Unicontrol Code hinzugefügt wird.

Das Motorsteuergerät verwendet zur Kommunikation eine Can-Open verbindung. Dies machte es nicht möglich wie zuvor die Driverlib zu nutzen. Deshalb mussten hier 2 Funktionen integriert werden um die Can Nachrichten im gewünschten Format zu verpacken und zu entpacken. Es wurde gefordert, den Motor im Current Mode zu betreiben.

1.2 Aufbauende Arbeiten

Aufbauen tut diese Bachelor-Projektarbeit auf mehreren anderen Arbeiten. Einer dieser Arbeiten ist die Regeleinheit namens Unicontrol. Diese besitzt ein Can Interface zur Kommunikation mit den vorhandenen Knoten. Hierüber ist es möglich durch Software einen Regler zu implementieren.
Vorhanden war zudem, die Auswerte Elektronik, die auf der Laufkatze befestigt ist. Diese wurde als Platine realisiert und besitzt unter anderem den Linear-Winkelencoder, ein PWM interface und eine Ansteuerung eines Hubmagneten. Die Kommunikation über-

nahm hier Can-Automotive.
Außerdem war ein Motorsteuergerät vorhanden, bei dem die Grundlagen schon ausgearbeitet wurden und zur Verfügung standen. Hierbei war darauf zu achten, das Can-Open verwendet wurde.
Der komplette mechanische Aufbau wurde ebenfalls von einem Studenten übernommen.

1.3 Grundlagen

1.3.1 Betriebssystem

Jeweils Unicontrol sowie die Platine nutzen das Betriebssystem TI-RTOS[1] . Eine besonderheit ist dabei, das es sich um ein Echtzeitbetriebssystem handelt. Das bedeutet, das Code in einem bestimmten Zeitfenster auf jeden Fall ausgeführt wird. Speziell als Beispiel dieses Projektes ist zu nennen, das alle 10 ms die Winkel und Winkelencoder Werte angefragt werden und auch alle 10 ms zur Auswertung bereit stehen müssen um den weiteren Betrieb möglich zu machen.

1.3.2 Entwicklungsumgebung

ALs Entwicklungsumgebung kommt Code Composer Studio zum einsatz, welche speziell für die Entwicklung von Applikationen von TI Mikrocontroller entwickelt wurde. Unicontrol verwendet den Tiva? TM4C1294NCPDT Mikrocontroller. Die Zusatz Platine verwendet den Tiva? TM4C123GH6PM Mikrocontroller.

1.3.3 Sicherheit

Erst einmal ist die Einstellbare Position in ControlDesk für den Nutzer auf die vorhandene Strecke begrenzt. Dies macht es sofern alles ordnungsgemäß funktioniert unmöglich für den Nutzer über die Position hinaus zu fahren. Sollte es jedoch mal zum Problem kommen zum Beispiel durch mechanische Abnutzung des Kodierfilms, besitzt die Laufkatze Magnete an beiden enden als Endschalter. Diese sollen, sobald der Hall Sensor diese im initialisiertem Betrieb erkannt, die Platine komplett abschalten. Vorgebeugt werden soll dieser Fehler jedoch schon direkt von der Initialisierung. Diese fährt erst einmal die komplette Strecke ab und prüft die Funktion der beiden Magneten. Sind die Magneten erst einmal erkannt, wird noch geprüft ob keine Striche auf der Strecke verloren gegangen sind. Das Steuergerät schaltet automatisch ab, sobald das Can Kabel gekappt wird. Dies ist auch sehr sinnvoll, da nun keine Informationen mehr ankommen, das der Motor aufhören soll zu fahren. Diese Funktion wurde programmier technisch auch bei der Platine realisiert.
Solange diese Initialisierung nicht abgeschlossen ist, verfügt der Benutzer nicht über alle Funktionen von ControlDesk. Durch diese Initialisierung wird sicher gestellt, das die Magnete ordnungsgemäßig befestigt sind und später als Endschalter funktionieren. Zudem wird durch die Initialisierung bestätigt, das keine Streifen auf der Strecke Verloren gehen.

1.3.4 Simulink

Es ist geplant, den kompletten Regler in Simulink aufzubauen. Simulink erzeugt aus dem kompletten Modell C-Code und fügt diesen automatisch in Code Composer Studio hinzu. Hierbei war darauf zu achten, das man die Variablen, die später über ControlDesk verstellbar sind, extra eingepflegt werden.

1.3.5 Asap Editor

In Asap Editor ist es möglich eine .A2L Datei zu erstellen. Die .A2L Datei wird später von dem Programm Control Desk benötigt. In dieser Datei wird unter anderem eine Zuordnung zwischen dem symbolischen Variablennamen und zugehörigen Adressbereich getroffen. Dies soll es später möglich machen, die Variablen Werte während der Laufzeit zu verändern.

1.3.6 Control Desk

ControlDesk wird hier zusammen mit einer GUI dem Benutzer zur Verfügung gestellt. Von hier ist es möglich, Variablen Werte während der Laufzeit des Programm zu verändern. Variablen Werte konnten hier in der GUI über Verschiedene Komponenten verstellt werden. Möglich macht das verändern der Werte aber in wirklichkeit die XCP Verbindung. ControlDesk dient hier nur als GUI.

1.3.7 XCP

Eine XCP Verbindung wird benötigt um während der Laufzeit die verschiedenen Parameter zu Verändern oder auszulesen. Die XCP Verbindung läuft über den Can 1 Anschluss des Unicontrols. Bei XCP handelt es sich um ein universelles Mess- und Kalibrierprotokoll, welches komplett unabhängig vom verwendeten Netzwerktyp verwendet werden kann. Möglich ist zum Beispiel wie hier XCP on CAN oder aber auch XCP on USB usw.

1.3.8 Can

Bei Can handelt es sich um ein serielles Bussystem, das von Bosch entwickelt wurde. Bussysteme haben beispielsweise den Vorteil, dass mit ihnen ein Großteil an Kabelbäumen reduziert werden kann was Geld und Gewicht einspart. Große Verwendung findet Can unter anderem auch wegen diesen Vorteilen im Automotive Bereich.
Can arbeitet nach einem Multi Master Prinzip. Da bei diesem Prinzip, das gleichzeitige Senden von mehreren Stationen zu Kollisionen führen würde, benutzt der CAN-Bus ein prioritätengesteuertes Zugangsverfahren, das Arbitration. Es setzt sich immer die Nachricht durch, die mit der höchsten Priorität vergeben ist. Prioritäten müssen also mit bedacht von dem jeweiligen Entwickler vergeben werden, um schnell genug auf wichtige Ereignisse reagieren zu können. Die beiden zustände werden mit rezessiv(logisch 1) und dominant(logisch 0) bezeichnet. Der dominante Zustand setzt dich dabei immer gegen den rezessiven Zustand durch.

Zur eigentlichen Kommunikation, ist es ausreichend nur die CAN-HIGH und CAN-LOW Ader zu verwenden. Bei den beiden Signalen, handelt es sich um differenzielle Signale, die mit entgegen gesetzter Polarität übertragen werden. Dadurch wird eine hohe elektrische Störsicherheit erreicht, was ebenfalls den Can auszeichnet. Mit der hier verwendeten Can Verbindung ist eine Datenübertragung von 1Mbit/s möglich. Angegeben ist diese Übertragungsrate für eine Kabellänge von 40 Metern. Um eine optimale Kommunikation zu ermöglichen, muss noch eine Busterminierung vorgenommen werden. Terminiert wird hier an den enden des Netzwerkes jeweils mit 120 Ohm Widerständen. Ohne Terminierung würde es im Netzwerk zu Reflexionen kommen.

1.4 Herangehensweise

Zuerst einmal wurden die ganzen Grundlagen erarbeitet und hier Dokumentiert um dem Leser einen einfacheren einstieg in das Thema zu ermöglichen. Zudem wurde sich mit den aufbauenden Arbeiten beschäftigt.
Punkte die Sicherheitskritisch sind wurden erfasst und bewertet. Daraufhin wurde versucht diese durch die Software komplett zu verhindern.

1.4.1 Module

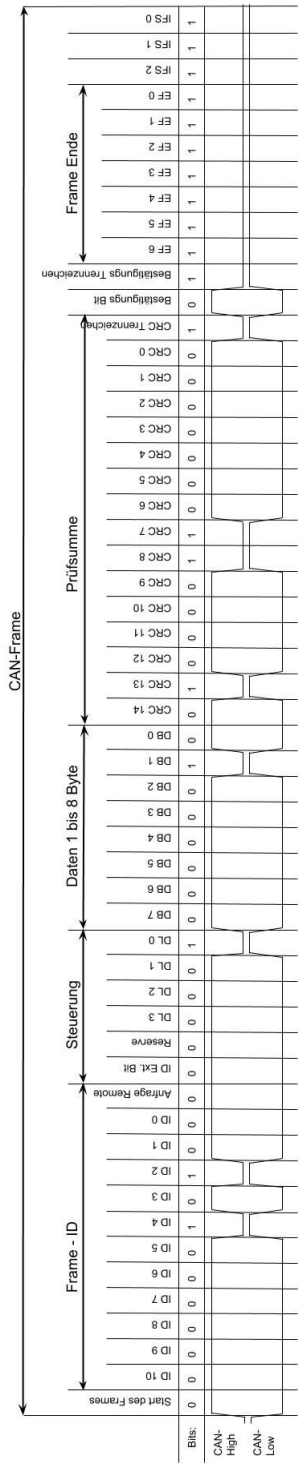
Alle verwendeten Module mussten erstmal komplett konfiguriert werden. Dazu entschied man sich hier eine vorhandene Driverlib zu verwenden. Diese musste natürlich erst mal an den entsprechenden Stellen sorgfältig gelesen und verstanden werden. Da die Driverlib auf benötigte Informationen von dem Datenblatt des Mikrocontrollers aufbaut, mussten zudem die entsprechenden Kapitel dieses nochmal zur Hand gezogen werden.

1.4.2 Haupt-Code

Hier musste sich erst einmal eine geeignete Initialisierung der Laufkatze überlegt werden, die die Funktionalität der Laufkatze sicher stellt. Erst danach konnte sich ein Algorithmus überlegt werden, der garantiert das alle 10ms die Daten von Linear-Winkelencoder zur Verfügung stehen. Zudem musste auf die Vergabe geeigneter ID Werte geachtet werden um so schnellst möglich auf Ereignisse zu reagieren.

1.4.3 GUI

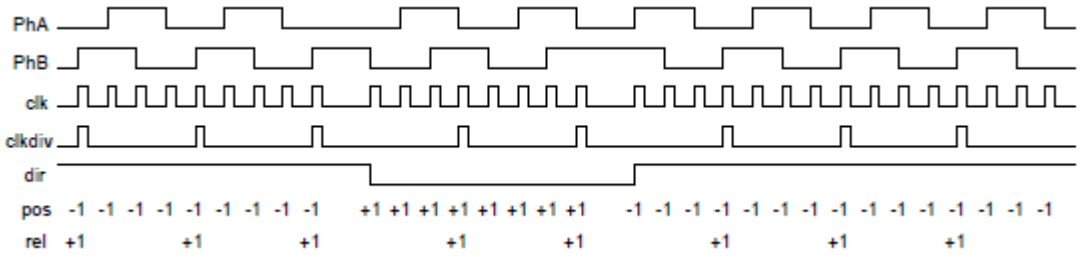
Es wurde sich eine geeignete GUI überlegt, die alle Funktionalitäten dem Benutzer zur Verfügung stellen soll. Für den Aufbau dieser Gui sei nochmal auf den Anhang verwiesen. Die Werte wurden hier schon begrenzt, sodass der User keine unzulässigen Werte einstellen kann bei denen jemand zu schaden kommen oder etwas kaputt gehen könnte.


CAN-Frame
Start des Frames
ID 10
ID 9
ID 8
ID 7
ID 6
ID 5
ID 4
ID 3
ID 2
ID 1
ID 0
Anfrage Remote
ID Ext. Bit
Reserve
DL 3
DL 2
DL 1
DL 0
DB 7
DB 6
DB 5
DB 4
DB 3
DB 2
DB 1
DB 0
CRC 14
CRC 13
CRC 12
CRC 11
CRC 10
CRC 9
CRC 8
CRC 7
CRC 6
CRC 5
CRC 4
CRC 3
CRC 2
CRC 1
CRC 0
CRC Trennzeichen
Bestätigungs Bit
Bestätigungs Trennzeichen
EF 6
EF 5
EF 4
EF 3
EF 2
EF 1
EF 0
IFS 2
IFS 1
IFS 0
Frame - ID
Steuerung
Daten 1 bis 8 Byte
Prüfsumme
Frame Ende
Bits:
CAN-High
CAN-Low

2 Initialisierung der Peripherie Module

2.0.1 QEI-Module

Die QEI ist ein Modul des Mikroprozessors, mit dem es möglich ist Positionsdaten und Geschwindigkeitsdaten zu erfassen. Die QEI benötigt dazu ein Quatratursignal, welches von einem Linear Encoder zur Verfügung gestellt wird. Die QEI zählt hierbei 4 Flanken pro Signalperiode des Kodierstreifens.



2.0.2 Hall Sensor

2.0.3 Hubmagnet

Ein Hubmagnet ist ein elektromagnetischer Aktor oder kurz Elektromagnet, der bei Bestromung ein Magnetfeld aufbaut. Durch dieses ist es möglich Gegenstände an eine Gewünschte Position zu transportieren. Die zusätzliche Masse am Pendel sorgt zudem für eine gewisse Trägheit, was natürlich die Regelung beeinflusst und den Lerneffekt der Studierenden erhöht.

2.0.4 Servomotor

Als Servomotor bezeichnet man eine spezielle Art der Elektromotoren, mit der es unter anderem möglich ist eine bestimmte Winkelposition zu erreichen. Innerhalb dieses Projektes, wurde der Servomotor über eine Pulsweitenmodulation(PWM) angesteuert.

2.0.5 Message Objekte

3 Algorithmus

3.1 Funktionsweise Haupt Code

3.2 Funktionsweise Sicherheitsmaßnahmen

4 Bedienungsanleitung

4.1 Software

5 Wartung

5.1 Epos

5.2 Software

5.3 Peak Adapter

5.4 Watchclass

6 Fazit

6.1 Aktueller Stand

In dem Projekt konnten alle gestellten Anforderungen erfolgreich umgesetzt werden. Es ist möglich die komplette Laufkatze an eine klar gewünschte Position zu Regeln. Möglich macht das der Regler der in Matlab implementiert wurde. Zudem konnten alle Sensoren und Aktoren erfolgreich in Betrieb genommen werden. Interessant zu sehen war hier, das die QEI keine Striche über die komplette Strecke verliert. Die Sicherheitsmechanismen wurden zudem auch erfolgreich Implementiert und getestet, sodass niemand zu schaden kommen kann. Zum besseren Verständnis wurde der C-Code auch nochmal sorgfältig mit Kommentaren versehen um später Änderungen schnell möglich zu machen. Da in Zukunft weiter mit dem Projekt gearbeitet werden wird, wurde ebenfalls ein Kapitel zur Wartung hinzugefügt, das dem Anwender möglichst einfach machen soll mögliche Fehlerquellen zu identifizieren.
In Zukunft könnten diese Informationen auch noch hilfreich für weitere Projekte sein wegen der guten Funktionsweise, weshalb auf eine detaillierte und Vollständige Ausarbeitung geachtet wurde.

6.2 Ausblick

Da es des öfteren Probleme bei dem Knoten gab der alle Transceiver im Netzwerk verbindet, wurde hierfür noch eine Platine in der Platinendesign Software Eagle designt. Diese sollte in Zukunft bestellt werden um einen optimalen Ablauf zu gewährleisten.
Zudem wär es sinnvoll in der Gui eine farbige LED zu implementieren, die dem Benutzer zeigt, ob die komplette Verbindung zur Laufkatze noch besteht oder ob diese beispielsweise erst mal noch initialisiert werden muss. Des weiteren sind noch die optimalen Parameter zu bestimmen um die Regelung perfekt zu machen.

7 Anhang

Abbildungsverzeichnis

Literaturverzeichnis