

Diplomarbeit 2002

DA – ES – 122

Embedded Systems

Autoren:

Claudio Foscan
Thomas Rickenmann

Betreuer:

Erwin Brändle

Datum: 10.12.2002

Hochschule Rapperswil

1 Abstract

In der ersten und zweiten Semesterarbeit von Herrn Rickenmann entstand ein Spinningbike, welches anstelle der üblichen mechanischen Bremse einen 3-phasigen Strom-Generator besitzt. Das Ziel dieser Diplomarbeit ist es, für dieses Trainingsvelo eine Laststeuerung und eine Trainingsdatenbank zu erstellen. In einer ersten Version der Laststeuerung wird eine Glühlampenmatrix eingesetzt, welche von einem kleinen Embedded System auf der Basis des MSP430 Mikrocontrollers der Firma Texas Instruments gesteuert wird. Der Trainierende hat über ein kleines GUI auf einem grafischen LCD-Display Einfluss auf die gewünschte Trainingsart und der Sollleistung. Neben der Ist-Leistung sind auch die aktuellen Werte von Puls und Trittkadenz jederzeit abrufbar. Eine zusätzliche Funktion gibt dem Trainierenden den Überblick über seine momentane Trainingseinheit. Das Embedded System ist in Verbindung zu einem Hostrechner, welcher alle Informationen über die Personen und ihren Trainingsdaten aufzeichnet und archiviert. Auf dem Hostrechner können nach jeder Trainingseinheit die Resultate mit den vorhergehenden verglichen und so ein Trainingserfolg abgeschätzt werden.



Abb 1 : DrillControlCenter for Spinningbike

2 Inhaltsverzeichnis

1	ABSTRACT	2
2	INHALTSVERZEICHNIS	3
3	EINFÜHRUNG	5
3.1	AUFGABENSTELLUNG	5
3.2	ABLAUF	6
3.3	BERICHT	6
3.4	TERMINE	7
3.5	BESONDERES	7
3.6	BEWERTUNG	7
3.7	ORGANISATORISCHES	7
4	PROJEKTPLANUNG	8
4.1	ZEITPLANUNG	8
4.2	GROBPFLICHTENHEFT	8
5	SYSTEMBESCHREIBUNG	10
5.1	BLOCKSCHALTBILDER	10
5.1.1	SPINNING POWERBACK VELO	10
5.1.2	DRILL CONTROL CENTER	11
5.1.3	NETZTEIL UND SENKE	12
5.1.4	PC - ARCHIV	12
5.2	INBETRIEBNAHME	13
6	SPINNING BIKE	16
7	DCC-NETZTEIL & LEISTUNGSINTERFACE	17
8	DRILL CONTROL CENTER	20
8.1	MSP430F149	22
8.1.1	BASIC CLOCK SYSTEM	23
8.1.2	PORTBELEGUNG	25
8.2	PULSEMPFÄNGER	26
8.2.1	FUNKTIONSWEISE DES SENDERGÜRTELS	26
8.2.2	VERSTÄRKER MIT TIEFPASS	26
8.2.3	KOMPARATOR	27
8.2.4	ASTABILER KIPPSCHWINGER	28
8.3	LEISTUNGSMESSUNG	29
8.4	TASTER UND ENCODER	30

8.5	KOMMUNIKATION.....	31
8.6	DISPLAY	33
9	JAVA GRAPHICAL USER INTERFACE.....	35
9.1	ANALYSE	35
9.2	ANFORDERUNGEN	35
9.3	DESIGN.....	36
9.4	FUNKTIONALITÄT	37
9.4.1	NEW USER.....	38
9.4.2	FREE TRAINING	40
9.4.3	CONCONITEST	42
9.5	ENTWICKLUNGSUMGEBUNG	45
9.5.1	VORBEREITUNG / INSTALLATION.....	45
9.5.2	BESONDERHEITEN	45
9.5.3	ARCHIV	46
10	EMBEDDED SOFTWARE.....	48
10.1	STATEMACHINE	48
10.1.1	STATE 0 WELCOME	49
10.1.2	STATE 1 LOGIN.....	49
10.1.3	STATE 2 CONFIRM NAME	49
10.1.4	STATE 3 CHOOSE MODE	49
10.1.5	STATE 4 PREPARE FOR TRAINING	50
10.1.6	STATE 5 FREE TRAINING	50
10.1.7	STATE 6 CHOOSE TRAINING PROGRAMM	50
10.1.8	STATE 7 TRAINING PROGRAMM	51
10.1.9	STATE 8 CONFIRM TEST	51
10.1.10	STATE 9 TEST	51
10.2	MODULE	52
10.2.1	I2C LIBRARY.....	52
10.2.2	T6963C-DISPLAYCONTROLLER LIBRARY.....	52
10.2.3	PCF8591 A/D-WANDLER LIBRARY	53
10.2.4	PCF8574 PORTEXPANDER LIBRARY	53
10.2.5	UART LIBRARY	53
10.2.6	POWERCONTROL LIBRARY.....	54
10.2.7	VIEW LIBRARY	54
10.3	ENTWICKLUNGSUMGEBUNG	55
10.3.1	INSTALLATION.....	55
10.3.2	NEUES PROJEKT	55
10.3.3	PROBLEME.....	58
11	DANKSAGUNG.....	58
12	FAZIT DER DIPLOMARBEIT	59
13	ABBILDUNGSVERZEICHNIS	60

3 Einführung

In zwei Semesterarbeiten wurde im Fachbereich Energietechnik ein Spinning-Bike entwickelt, welches die aufgewendete Muskelarbeit in elektrische Energie umwandelt. Das Trainingsgerät arbeitet nach einem neuartigen Powerback-Konzept, welches sogar die Möglichkeit einer Einspeisung ins Niederspannungsnetz vorsieht.

Der aktuelle Stand der Arbeiten ist so, dass mit dem integrierten Scheibenläufergenerator die eingebrachte Muskelenergie in eine elektrische 3-Phasen Wechselspannung gewandelt wird. Die eigentliche Rückspeisung ins Niederspannungsnetz wird ev. in späteren Studien- oder Diplomarbeiten weiterverfolgt.

Beim Einsatz moderner Spinning-Bikes in Fitness-Centern ist die gezielte Trainingskontrolle ein zentraler Erfolgsfaktor. Dafür sind hochwertige Kontroll- und Steuergeräte erforderlich, welche es ermöglichen, die Trainingseinheiten gezielt zu kontrollieren und Trainingserfolg und Fortschritt an den Traininginstruktor weiterzugeben.

3.1 Aufgabenstellung

Für das im Rahmen der erwähnten Studienarbeiten entwickelte Spinning-Bike soll ein *Drill Control Center* entwickelt werden, welches erlaubt, Trainingseinheiten gezielt zu steuern und zu koordinieren.

Neben den eigentlichen Kontroll- und Steuerfunktionen ist ein spezielles Augenmerk auf plausible und einfache Bedienbarkeit zu richten. Das *Drill Control Center* soll in der Lage sein, die anfallenden Daten ganzer Trainingseinheiten zu speichern und bei Bedarf über eine geeignete Schnittstelle an die Kontrolleinheit des Traininginstruktors zu übertragen. Über die Kontrolleinheit soll der Instruktor in der Lage sein, Trainings zu definieren, überwachen und auszuwerten.

Die Diplomarbeit umfasst unter anderem die folgenden Schwerpunkte:

- Erarbeiten des Gesamtkonzeptes
- Verfassen eines Pflichtenheftes
- Festlegen des Funktionsumfanges
- Definieren des Bedienkonzeptes und HMI-Design (Human Machine Interface) Entwicklung
- Aufbau und Inbetriebnahme des eigentlichen *Drill Control Centers* sowie der benötigten Hard- und Software
- Implementation einer einfachen Kontrolleinheit für den Traininginstruktor auf einem Standard-PC

Um die Anforderungen an das *Drill Control Center* gezielt definieren und abstecken zu können, ist bei Bedarf Kontakt mit externen Fachspezialisten aufzunehmen.

3.2 **Ablauf**

Zu Beginn der Arbeit ist ein Projektplan zu erarbeiten, um die Arbeit zeitlich zu strukturieren. Weiter ist ein Grobpflichtenheft zu erstellen, welches von allen beteiligten Parteien zu unterzeichnen ist.

Die Arbeiten sollen innerhalb des Projektteams geeignet aufgeteilt werden; die Aufteilung ist im Bericht entsprechend festzuhalten.

Weitere Einzelheiten werden an den regelmässigen Besprechungen festgelegt; wichtige Entscheide sind als Kurzprotokolle dem abschliessenden Bericht (im Anhang) beizufügen.

3.3 **Bericht**

Über die Diplomarbeit ist ein Bericht zu verfassen, dessen Textteil (ohne Anhang) 60 Seiten nicht übersteigen soll. Im Bericht sollen alle wesentlichen gemachten Überlegungen, Abklärungen, Berechnungen und Untersuchungen detailliert (in Text und Bild) dokumentiert werden.

Der Bericht muss gut leserlich geschrieben und übersichtlich gegliedert sein. Er soll mindestens die folgenden Kapitel umfassen: Inhaltsverzeichnis, allgemeine (für nicht Fachleute) verständliche Einleitung, Aufgabenstellung (Original), Zusammenfassung, kommentierter Zeitplan (Soll) und Arbeitsfortschritt (Ist) sowie die Aufteilung der Arbeiten. Der Aufbau des übrigen Teils des Berichtes mit der Analyse der Aufgabenstellung und Lösungsmöglichkeiten, der Konzept- und Realisierungsbeschreibung, sowie der zugrunde liegenden theoretischen Betrachtung ist der Aufgabe entsprechend zu gestalten. Zum Bericht gehören auch eine Systembeschreibung, Angaben darüber wie das System erstellt werden muss, sowie eine kommentierte Auflistung der erstellten Verzeichnisse und Daten.

Der Bericht ist in 2 Papier-Exemplaren, sowie in elektronischer Form (Word Dokument) abzugeben.

Alle anfallenden Daten, wie auch der Bericht sind gemäss separater Richtlinie zu archivieren.

3.4 Termine

Mo.	21.10.2002	KW 43/02	Beginn der Diplomarbeit
Di.	10.12.2002 / 24 Uhr	KW 50/02	Ende der Diplomarbeit, Abgabe des Berichtes
Mi.	11.12.2002	KW 50/02	Präsentation der Diplomarbeit für Examinatoren, Fachexperten
Do.	12.12.2002		
Fr.	13.12.2002	KW 50/02	Ausstellung der Diplomarbeiten
Do.	16.01.2003	KW 03/03	Mündliche Diplomprüfung

Im übrigen gelten die Termine der Abteilung Elektrotechnik und individuelle Vereinbarungen.

3.5 Besonderes

Aus dem Bericht muss klar hervorgehen wie die Aufgabenteilung innerhalb des Teams vorgenommen wurde und wer für welche Teile des Berichtes verantwortlich ist.

Besondere Anschaffungen wie auch Arbeiten an Drittpersonen (Assistenten, Lehrlinge, usw.) müssen vorgängig abgesprochen und rechtzeitig koordiniert werden.

Als Energie-Senke kann u.U. auch eine einfache Schaltung mit sequentiell zugeschalteten ohmschen Lasten (z.B. Glühlampen) verwendet werden.

3.6 Bewertung

Bewertet werden: Arbeits-Methodik, Technischer Inhalt, Resultat, Bericht, Präsentation

3.7 Organisatorisches

Betreuung der Arbeit:	E. Brändle
Industriepartner:	(Labor Energietechnik, Prof. H.Prechtl)
Betreuung des Labors:	R. Brunner / C. Naef
Zeitaufwand nach Studienführer:	ca. 7½ Wochen
Arbeitsplatz:	CT-Labor 6.007
Besprechungen:	nach Vereinbarung

4 Projektplanung

4.1 Zeitplanung

Um möglichst effizient Arbeiten zu können, haben wir die Aufgaben unseren Fachgebieten entsprechend aufgeteilt. Herr Rickenmann ist für die sportlichen Aspekte, sowie das Interface für den Betreuer der Trainingsperson verantwortlich. Ausserdem konstruiert er den Pulsempfänger. Die Aufgaben von Herrn Foscan sind die Entwicklung der Drill Control Center Soft- und Hardware sowie die Leistungssenke. Folgender Zeitplan veranschaulicht die geplanten Termine der einzelnen Komponenten.

Herr Rickenmann Woche:	43	44	45	46	47	48	49	50
Hardware Pulsempfänger								
Software Interface								
Herr Foscan	43	44	45	46	47	48	49	50
Hardware Senke								
Hardware DrillControlCenter								
Software DrillControlCenter								

Abb 2 : Zeitplanung

4.2 Grobpflichtenheft

Wie aus der Aufgabenstellung zu entnehmen ist, soll ein HMI und eine Software zum Spinning Powerback Trainingsvelo entwickelt werden. Dies beinhaltet folgende Punkte :

- Das HMI soll intuitiv und einfach zu bedienen sein.
- Es soll möglichst genau Aufschluss über die Trainingseinheit bezüglich Leistung geben.
- Die Trainingsperson muss jederzeit ihre individuelle Leistung berücksichtigen und das Trainingsgerät danach anpassen können.
- Die Betreuungsperson muss jederzeit die Leistungsdaten des Trainierenden im Auge haben können.
- Nach jeder Trainingseinheit soll der Trainierende und sein Betreuer die Trainingseinheit besprechen und mit anderen Trainingseinheiten vergleichen können.

Massnahmen zur Einhaltung der geforderten Punkte:

1. Am Trainingsvelo wird die Kontrolleinheit auf dem Lenker so befestigt, das der Trainierende die Einheit gut erreichen kann.
2. Die verwendete Anzeige ist sehr kontrastreich und genügend gross.
3. Die Menüs der Kontrolleinheit sind einfach und mit guter Verständlichkeit beschriftet. Als Systemsprache benutzt das Drill Control Center Englisch.
4. Die Software auf dem Überwachungsrechner ist selbsterklärend und ohne grosse Schnörkel.
5. Es findet immer eine Kommunikation zwischen Drill Control Center und Hostrechner statt. Dadurch hat der Betreuer immer die aktuellen Daten des Trainierenden.

Zur Verfügung stehen folgende Mittel :

1. MSP430F149 Hardware von Olimex
2. C-Compiler von Quadravox
3. Borland JBuilder 5
4. Platinen Layouttool von OrCad
5. Schema Editor von OrCad
6. Microsoft Office für die Dokumentation
7. zwei Arbeitsplätze im Embedded-System-Labor

Für allfällige Ausgaben ist Hr. Erwin Brändle zu kontaktieren.

5 Systembeschreibung

5.1 Blockschaltbilder

Das ganze System ist aus folgenden Komponenten aufgebaut :

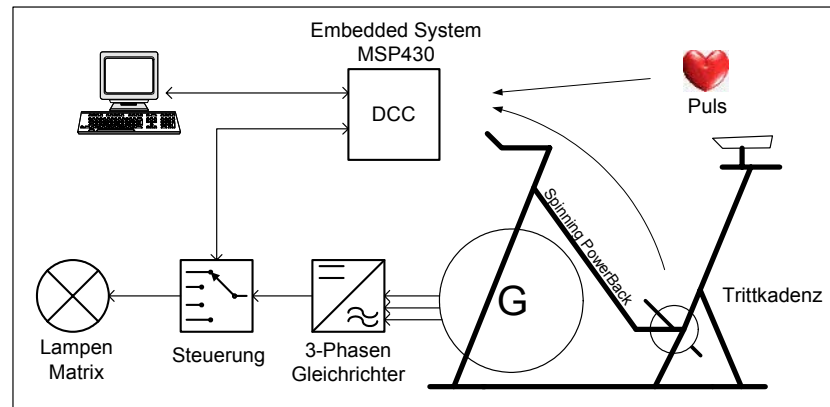


Abb 3 : Blockschaltbild Total

Physikalisch sind vier Systeme vorhanden.

- Das Spinning Powerbackvelo mit Generator und Gleichrichter
- das Drill Control Center, montiert auf dem Spinning Powerback Velo
- Eine Senke mit Netzteil, Laststeuerung und Leistungsmessung
- Ein PC mit der gesamten Trainingsdatenbank.

5.1.1 Spinning Powerback Velo

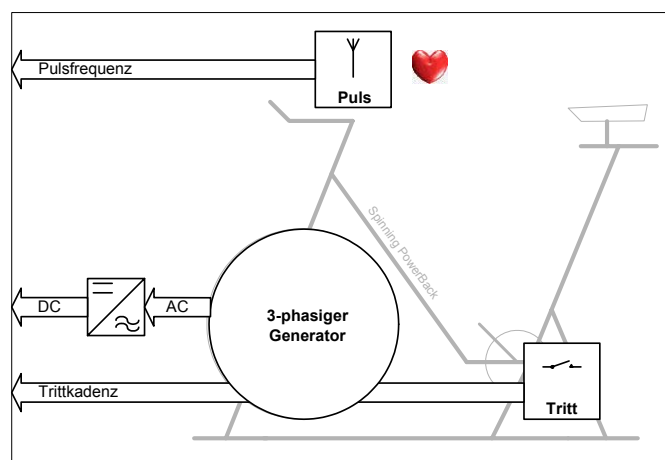


Abb 4 : Blockschaltbild Spinning Powerback

Das Spinning Powerback Velo liefert den Trittkadenzimpuls, die Pulsimpulse und vom Generator die gleichgerichtete ungeglättete Spannung.

5.1.2 Drill Control Center

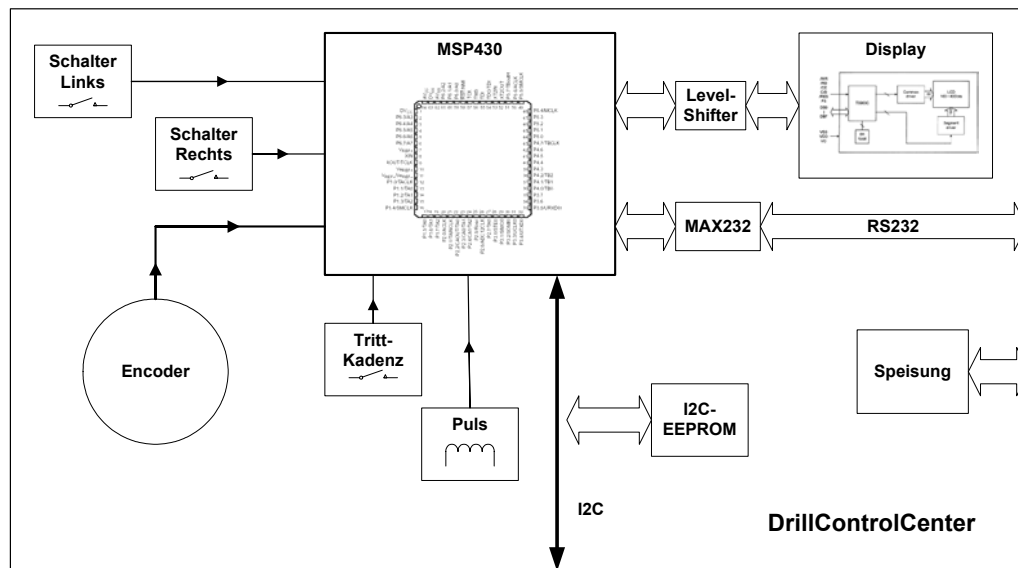


Abb 5 : Blockschaltbild Drill Control Center

Die zentralen HMI-Komponenten befinden sich alle im Drill Control Center. Das Display, die Taster und der Encoder stellen die Kommunikation zwischen Mensch und Controller sicher. Die Steuerung des ganzen Systems erfolgt via I2C-Bus. Der Datenaustausch mit dem PC - Archiv erfolgt über eine serielle Schnittstelle.

5.1.3 Netzteil und Senke

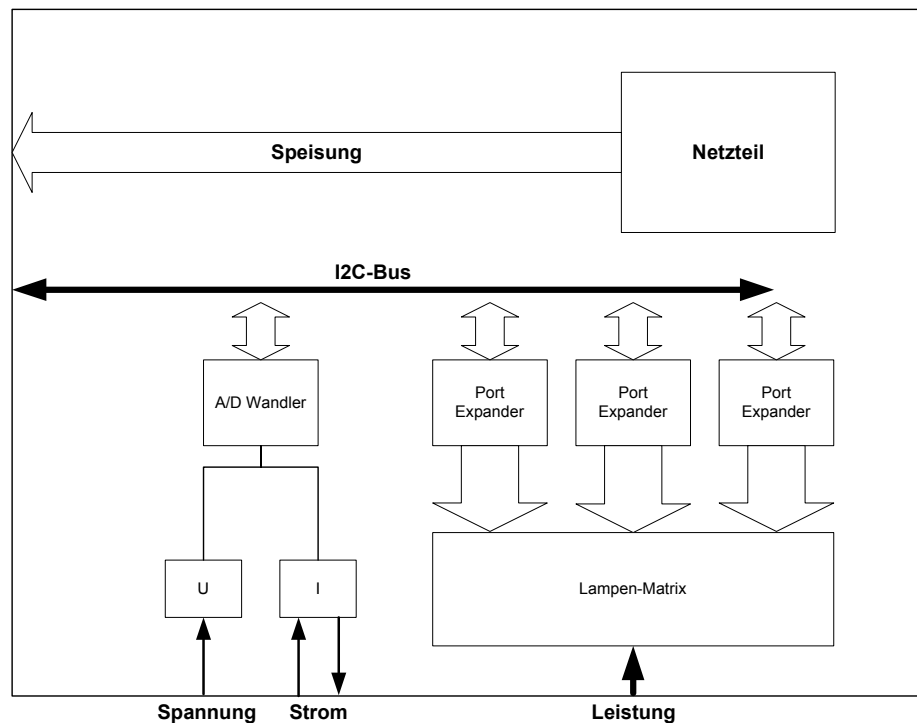


Abb 6 : Blockschaltbild Netzteil und Senke

Das Netzteil liefert die Versorgungsspannungen für das ganze Drill Control Center. Benötigt werden 5 Volt für das Display, 3.3 Volt für den MSP430F149 und das I2C-EEPROM. Die +12 Volt und -12 Volt werden für die Empfänger der Trittkadenz und des Pulses benötigt. Die Leistungsmessung erfolgt mittels Spannungs - und Strommessung der Zuleitungen der Senke. Die Spannung wird heruntergeteilt und geglättet vom I2C-AD-Wandler eingelesen. Mit einer Stromzange wird aus dem Strom eine zum Strom proportionale Spannung erzeugt und ebenfalls mit dem AD-Wandler eingelesen. Strom mal Spannung ergibt die Momentanleistung. Die Lampenmatrix wird mittels I2C Portexpandern angesteuert.

5.1.4 PC - Archiv

Als PC-Archiv kommt ein handelsüblicher Personal Computer mit Windows als Betriebssystem zum Einsatz. Dadurch, dass die Archivsoftware auf Java basiert, wäre auch ein Einsatz auf anderen Plattformen wie Apple Macintosh oder Unix Workstations denkbar. Die Auswertung der Trainingsdaten erfolgt aber via Microsoft Excel.

5.2 Inbetriebnahme

Das Spinning Powerback Velo sollte auf einer ebenen Fläche mit genügend Platz rundherum aufgestellt werden. Es ist auf die richtige Höhe des Sattels und des Lenkers zu achten. Das Drill Control Center wird auf dem Lenker befestigt.

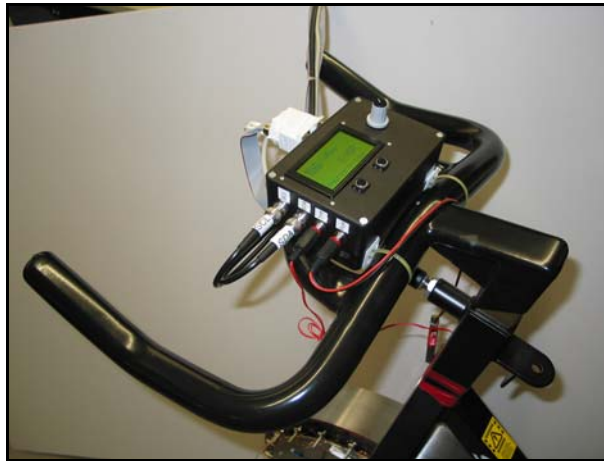


Abb 7 : Installation Befestigung DCC am Lenker

Zuerst verbindet man den Pulsempfänger und den Trittkadenzgeber (Reedkontakt an den Pedalen) mit dem DCC.

Danach folgen die I2C Leitungen SDA und SCL. Beim Prototypen sind diese Leitungen separat und abgeschirmt geführt um Störungen zu vermeiden.

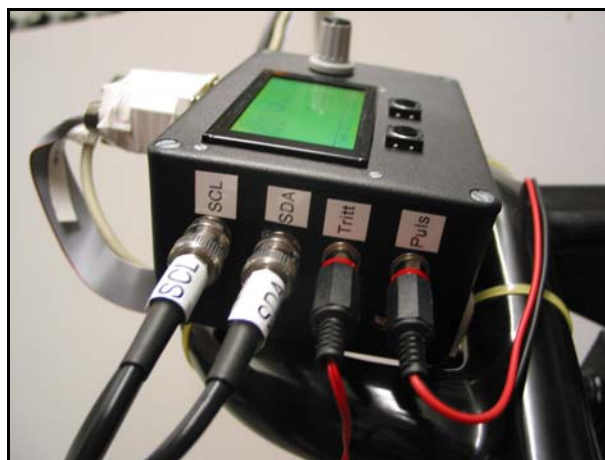


Abb 8 : Installation Anschlüsse DCC

Nun fehlen noch die Speisung und die Verbindung zum PC Archiv.

Die Powerleitungen werden mit der Senke verbunden.

Wichtig : auf die Polarität achten!!!

Im ausgeschalteten Zustand liegt die Maximalleistung der Lampenmatrix an den Poweranschlüssen.

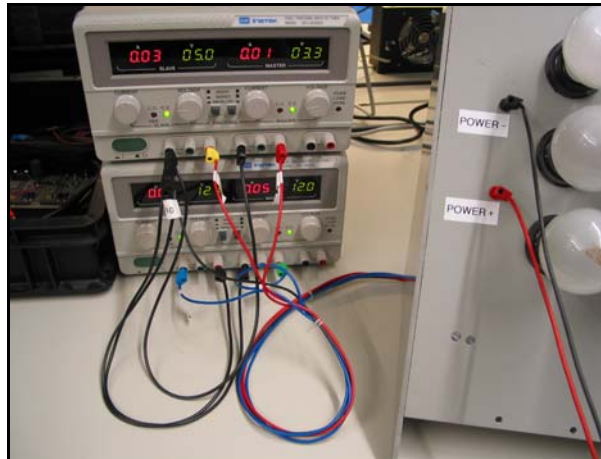


Abb 9 : Installation Anschluss der Speisung

Danach die Speisung der Senke anschliessen und einschalten.

Wichtig : auf die Polarität und Spannung achten!!!

Nun sollte auf dem Display des DCC der Willkommensbildschirm angezeigt werden. Falls dies nicht der Fall ist, den Resettaster betätigen und /oder die Speisung der Senke und des DCC überprüfen.

Damit die Leistungsberechnung korrekt arbeitet, die Stromzange einschalten (auf 100mV/A Bereich) und auf Null Volt abgleichen!



Abb 10 : Installation Einschalten und Abgleich der Stromzange

PC-Archiv anschliessen und Programm starten. Eventuell muss vorher noch die Java Virtual Machine installiert werden. Auf dem PC sollte ebenfalls Microsoft Excel 2000 oder höher installiert sein.

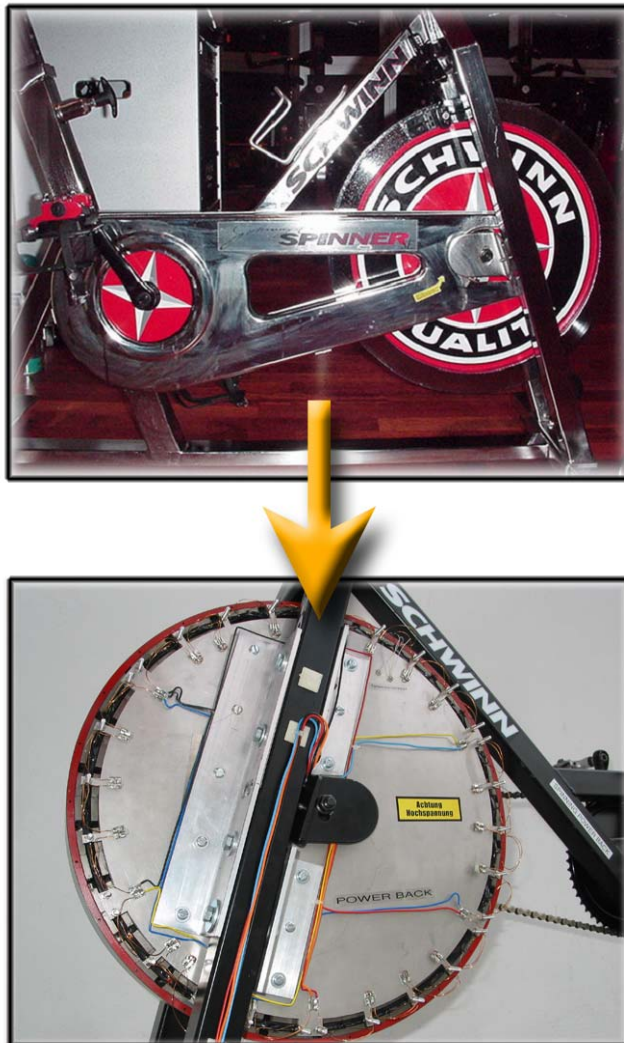


Abb 11 : Software für Betreuer

Danach ist das Spinning Powerback Trainingsvelo einsatzbereit. Viel Spass!

6 Spinning Bike

Ein Spinningbike ist eine spezielle Form eines Velo Hometrainers, der sich durch einen stärkeren Rahmen und eine Schwungmasse, die durch eine Kette angetrieben wird, von den gewöhnlichen Trainingsgeräten unterscheidet.



Ersetzt man nun diese ca. 25 kg schwere Schwungmasse durch einen Generator, hat man bereits ein Kleinst-Kraftwerk konzipiert. Anstatt die geleistete Arbeit ungenutzt zu verheizen, kann sie sinnvoll in hochwertige elektrische Energie umgewandelt und ggf. in das öffentliche Stromversorgungsnetz eingespiesen werden.

Während den ersten beiden Semesterarbeiten 2001-2002 im Fach Energietechnik entstand ein voll funktionsfähiger Generator, der eine Leistung von max. 700 Watt abgeben kann.

Abb 12 : Spinning Powerback

Die 3-phasige Ausgangsspannung variiert je nach Trittfrequenz zwischen 120V – 650V. Je nach dem zugeschalteten Widerstand ergibt sich die gewünschte Bremskraft.

7 DCC-Netzteil & Leistungsinterface

Beim Training auf dem Spinning Powerback Velo entsteht elektrische Arbeit. Um diese Arbeit wegzubringen, benutzen wir eine Lampenmatrix



Abb 13 : Lampenmatrix

mit welcher die Leistung in 5-Watt-Schritten einstellbar ist. Die Lampenmatrix wird mittels Relais und I2C-Portexpandern angesteuert. Auf der Platine befinden sich noch zwei StepDown DC/DC Wandler zur Erzeugung von 5 bzw. 3.3 Volt aus der 12 Volt Versorgungsspannung der Relais. Das Drill Control Center benötigt ± 12 Volt, 5 Volt und 3.3 Volt. Vom DCC-Netzteil führt ein Kabel mit allen erforderlichen Leitungen für die Speisung und den I2C-Bus zum Drill Control Center. Die Pinbelegung des DSUB9 Steckers auf der DCC-Netzteil Platine ist folgendermassen:

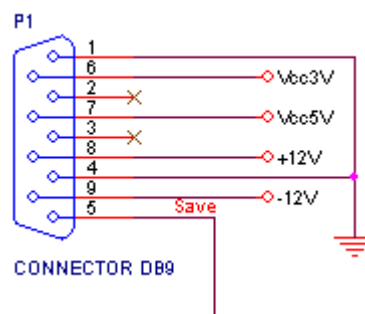


Abb 14 : Powerconnector

Die Save-Leitung ist für das Notausrelais gedacht. Sobald dieses Relais abfällt, fallen auch alle anderen Relais ab und das Spinningvelo wird mit der maximalen Leistung der Leistungsstufe gebremst. Dies ist nötig, da bei einem Spinningvelo kein Freilauf eingebaut ist, wie dies bei normalen Strassenvelos der Fall ist.

Aufgrund der variablen Spannung am Spinningvelo (0-600 Volt, abhängig von der Trittkadenz) muss die Leistungsstufe auch auf Überspannung reagieren können. Es sind drei Spannungsbereiche definiert: 0-200 Volt, 200-400 Volt und 400-600 Volt. Im ersten Spannungsbereich von 0-200 Volt reicht eine Glühlampe aus um die entsprechend Leistung durch Parallelschaltung mehrerer solcher zu erreichen. Im zweiten Spannungsbereich darf nie eine Glühlampe alleine an der Quellenspannung des Velos hängen, da sie sonst zerstört wird. Deshalb sind nur immer Serieschaltungen von mindestens zwei Lampen erlaubt. Im dritten Spannungsbereich sind nur Serieschaltungen von drei Lampen zugelassen.

Die Lampen sind wie folgt verdrahtet:

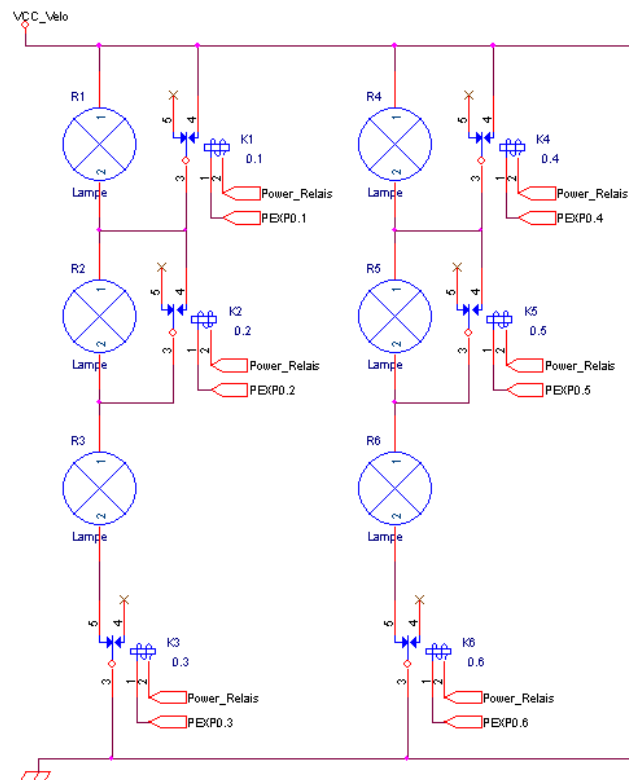


Abb 15 : Lampenarray

Relais K3 schaltet den ganzen Strang ein oder aus. Wenn Relais K1 und K2 eingeschaltet sind, überbrücken sie die Lampen R1 und R2. Das heisst, es ist nur die Lampe R3 an der Versorgungsspannung angeschlossen. Es darf also nur bis 200 Volt an der Versorgung liegen. Steigt die Spannung, fällt das Relais K2 ab und es liegen zwei Lampen in Serie an der Versorgungsspannung. Steigt die Spannung über 400 Volt, fällt auch noch das Relais K1 ab und es liegen alle drei Lampen in Serie. Geschaltet werden die Relais von PCF8574 Portexpandern für I2C. Die Bausteine können 8 Bit ausgeben und einlesen. Sie sind über die **I2C Adressen 0,1,2** ansprechbar.

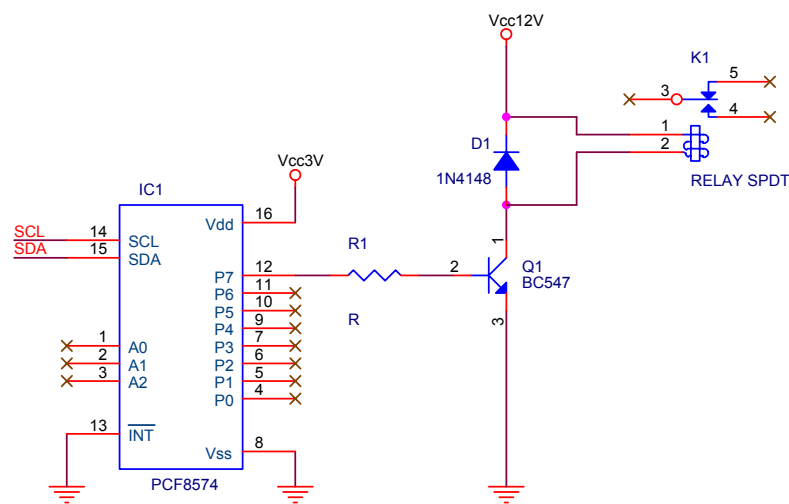


Abb 16 : PortExpander

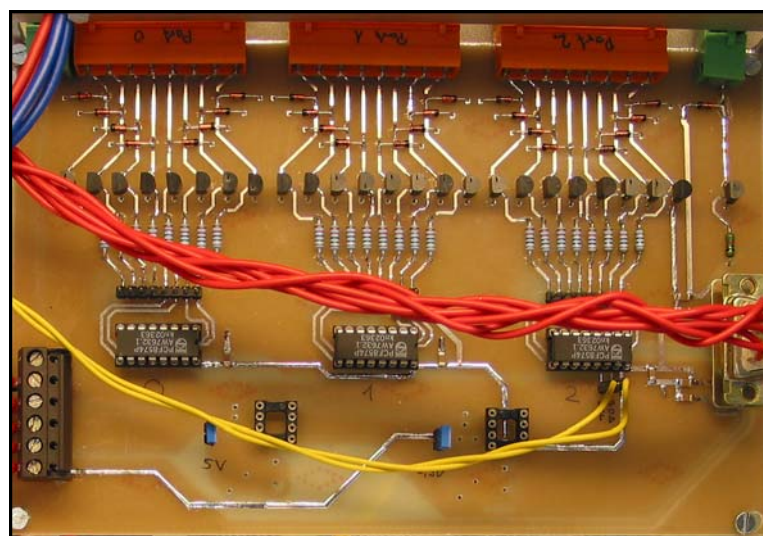


Abb 17 : PortExpander Hardware

8 Drill Control Center

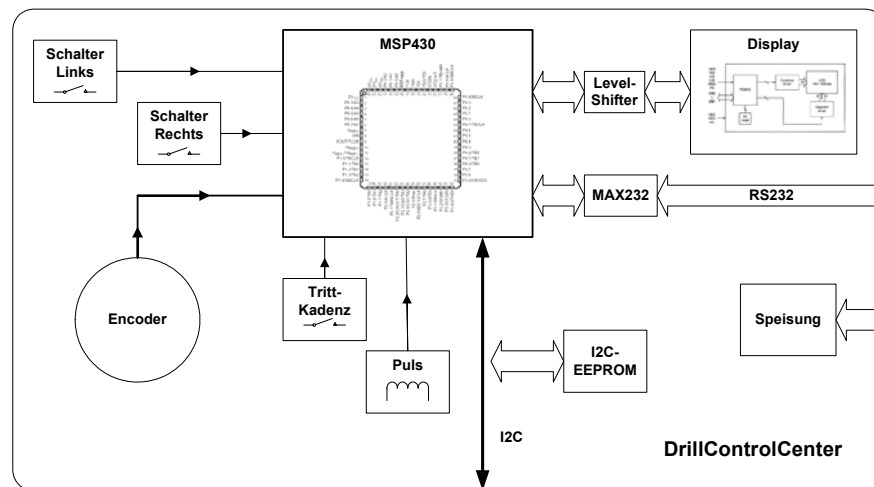


Abb 18 : Blockschaltbild Drill Control Center

Ein kleines auf dem Spinning Powerback Velo montiertes Gehäuse stellt den Platz für das komplette Drill Control Center zur Verfügung. Es beinhaltet folgende Komponenten :

1. Microcontroller MSP430F149
2. Pulsempfänger
3. Trittkadenzempfänger
4. Taster & Encoder
5. I2C Speicher
6. RS232 Treiber
7. Display Treiber

Alle diese Komponenten sind auf der eigens dafür gefertigten Platine untergebracht.

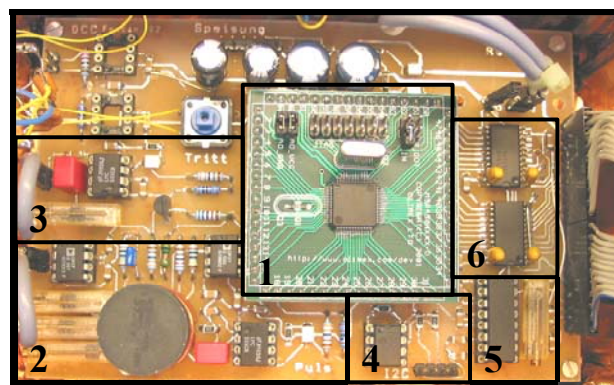


Abb 19 : Drill Control Center offen

Bei dem Microcontroller handelt es sich nicht um den reinen Chip, sondern um eine kleine Platine mit Quarz und Anschlüssen für JTAG und den kompletten Ports des MSP430F149. Der Pulsempfänger wandelt die Signale eines Polar - Sende - Gürtels in Pulse um, welche durch den MSP430 ausgewertet werden können. Dasselbe gilt für den Trittkadenzempfänger. Während einer Trainingseinheit werden alle Werte im internen Speicher abgelegt. Da der MSP430 nur über 2KB RAM verfügt, ist ein externer I2C Speicher vorhanden um die Werte aufzunehmen. Für die Kommunikation mit dem Hostrechner wird eine serielle Schnittstelle nach RS232 Standard verwendet. Da der MSP430 Microcontroller mit 3,3 Volt Betriebsspannung läuft, müssen die Pegel sowohl für die RS232 als auch für das Display angepasst werden. Diese Aufgabe übernehmen die beiden Bausteine MAX3232 für RS232 und 74LVC4245 für das Display.

Der Trainierende hat die Möglichkeit über das Display, den Encoder und die Taster mit dem Drill Control Center zu interagieren. Die beiden Taster sind so angeordnet, dass sie als Soft-Taster programmiert werden können. Das bedeutet, dass auf dem Display in der untersten Zeile die jeweilige Funktion für die Taster eingeblendet werden kann. Dasselbe gilt für den Encoder. Je nach dem in welchem Menu man sich befindet, übernimmt der Encoder eine andere Funktion. Näheres dazu ist im Kapitel über die Embedded Software zu finden.



Abb 20 : Frontplatte Drill Control Center

8.1 MSP430F149

Als zentrale Recheneinheit kommt ein MSP430F149 Microcontroller von Texas Instrument zum Einsatz. Microcontroller heisst, dass der Baustein bereits Peripherie eingebaut hat. Er besitzt zum Beispiel A/D-Wandler, zwei UARTs, Timer usw... Das Blockschaltbild veranschaulicht die interne Struktur des MSP340F149.

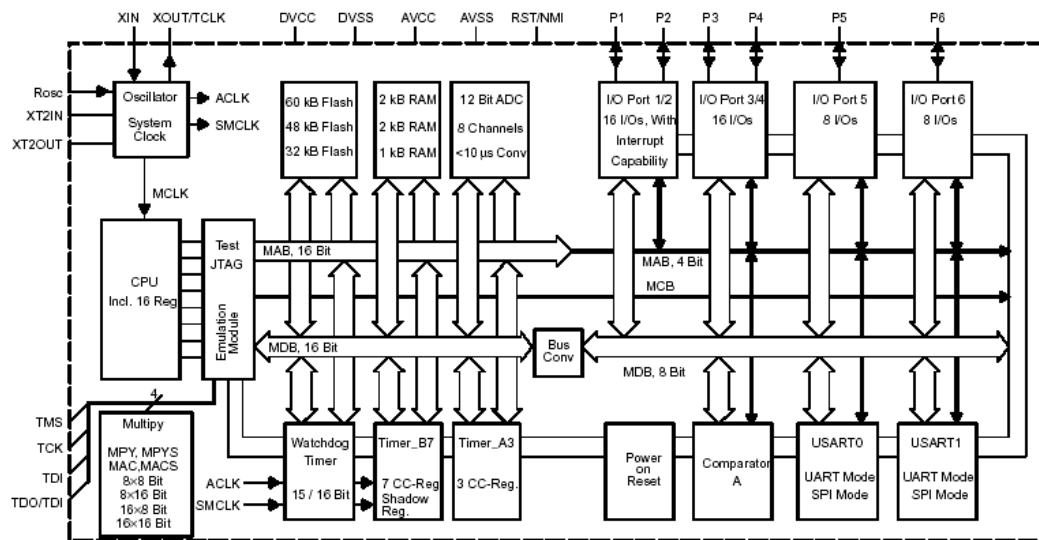


Abb 21 : Blockschaltbild MSP430F149

Im **Drill Control Center** kommen 5 von 6 Ports, eine UART, beide Timer und der Watchdog zum Einsatz. Der Controller besitzt 2KB RAM und 60KB Flash Speicher, welcher über die JTAG-Schnittstelle beschrieben werden kann. Das Besondere am Controller ist, dass kein Daten- und Adressbus nach aussen geführt wird. Für die Displaysteuerung müssen deswegen zwei Ports zweckentfremdet werden. Speziell für batteriebetriebene Geräte verfügt der Controller über vier verschiedene Low – Power - Modi, welche den Strombedarf drastisch senken. Auch im Normalbetrieb benötigt der MSP430 sehr wenig Strom (max 250uA bei 1MHz). Im DCC benutzt er aber keinen dieser Stromsparmodes. Es stehen mehrere Compiler für den MSP430 zur Verfügung. Für dieses Projekt wurde die Entwicklungsumgebung AQ430 von Quadravox eingesetzt.

8.1.1 Basic Clock System

Der MSP430F149 besitzt ein spezielles Taktsystem. Er verfügt über 3 Oszillatoren und 3 interne Taktsignale.

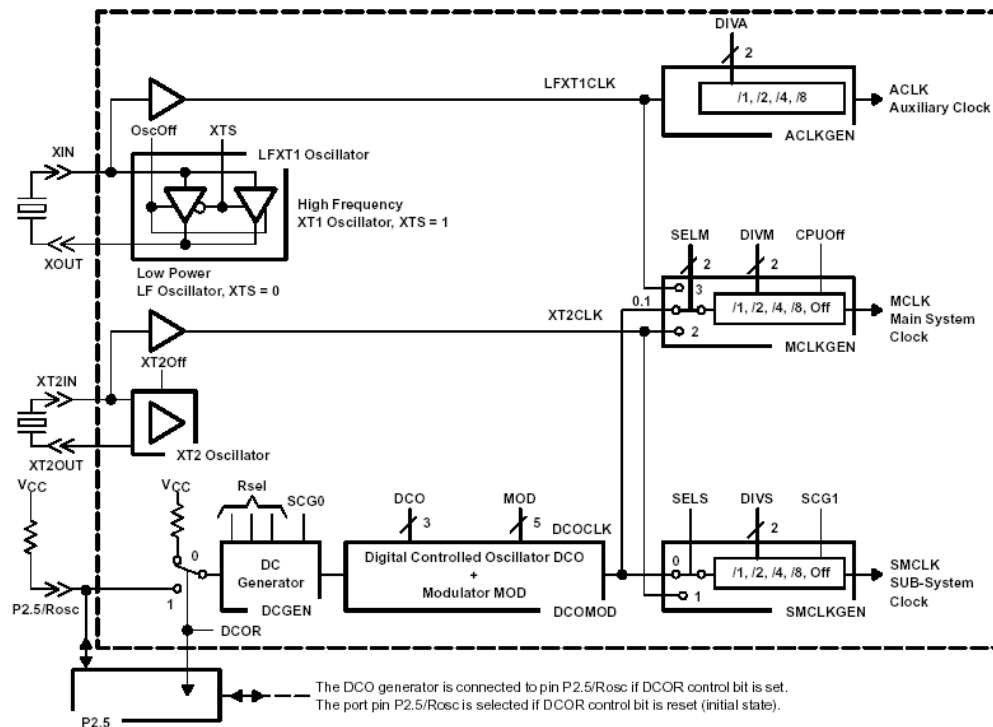


Abb 22 : Blockschaftbild BasicClockSystem

LFXT1 Oszillator

Der Lowpower Oszillator des MSP430. Er kann entweder mit einem Uhrenquarz im Niederfrequenzbereich (32KHz) oder mit einem normalen Quarz im Hochfrequenzbereich (455kHz – 8MHz) betrieben werden.

XT2 Oszillator

Kann nur für den Hochfrequenzbereich genutzt werden.

DCOszillator

Der Digital Controlled Oszillator des MSP430 ist im Chip selbst untergebracht und kann via Software eingestellt werden. Die Frequenz liegt zwischen 0 Hz und ca. 5MHz je nach Speisespannung.

ACLK	Auxillary Clock gespeisen immer vom LFXT1 Oszillator. Wird unter anderem für die UART benötigt.
MCLK	Main System Clock kann von allen drei Oszillatoren gespeisen werden. Wird für die CPU benötigt.
SMCLK	SUB System Clock kann von XT2 und DC Oszillator gespeisen werden. Wird für interne Komponenten wie AD-Wandler, UART oder Watchdog benötigt.

Alle drei Signale können je auf einen Pin des MSP430 gelegt werden und damit weiter verwendet werden für externe Komponenten.

Im **Drill Control Center** kommt ein ACLK von 32,768 kHz und ein MCLK/SMCLK von 4 MHz (mittels externem Quarz) zum Einsatz.

8.1.2 Portbelegung

Die Ports des MSP430F149 sind im Drill Control Center folgendermassen belegt:

P1.0–P1.2	Increment Encoder & Encoder Switch
P1.3, P1.4	Taster
P1.5	Trittkadenzempfänger
P1.6	Pulseempfänger
P2.0, 2.1	I2C Interface
P3.4, P3.5	RS232 Schnittstelle
P4.0–P4.7	Datenbus für Display
P5.0–P5.3	Steuerbus für Display
P6.0	Debug-Pin

Die restlichen Port-Pins stehen noch für Erweiterungen zur Verfügung. Port 1 ist Interrupt fähig. Das bedeutet, dass jedes Drücken eines Tasters, drehen des Increment Encoder, ein Impulsempfang vom Puls- oder Trittkadenzempfänger, ein Interrupt auslöst. Dies ist nötig, da für die Berechnung der Leistung, der Steuerung der Last und die Anzeige der Daten so viel Rechenzeit wie möglich zu Verfügung stehen muss. Puls und Trittkadenz sind zeitkritisch, da bei jedem Impuls die vergangene Zeit seit dem letzten Impuls gemessen wird. Port 2 wäre auch Interrupt tauglich, jedoch benötigt der I2C-Bus kein Interrupteingang. Port 3.4 und 3.5 sind mit der internen UART verbunden. Sie können direkt an den RS232 Treiberbaustein angeschlossen werden. Port 4 und 5 arbeiten als normale parallele Ports, wobei Port 5 immer als Ausgang geschaltet ist, während Port 4 bidirektional arbeitet. Port 6 wurde nur zum Debuggen der Embedded Software benutzt.

Programmiert wird der MSP430F149 via JTAG Schnittstelle. Eine genaue Beschreibung der JTAG Schnittstelle findet sich im Anhang.

8.2 *Pulsempfänger*

Die meistverbreiteten Pulsuhren auf dem Markt, die auch fast von jedem Sportcenter verwendet werden, sind die der Firma POLAR. Aufgrund dieser Tatsache haben wir uns entschieden, auch bei unserer Controleinheit einen Empfänger einzusetzen, der auf die Frequenz von POLAR-Sendern abgestimmt ist.

8.2.1 Funktionsweise des Sendergürtels

Bei jedem Herzschlag wird ein kleiner Spannungsimpuls an die Herzmuskulatur abgegeben, damit sich diese zusammenzieht. Die Sendergürtel der Firma POLAR messen diese Spannungsimpulse in Form einer Schrittspannung auf dem Oberkörper des Athleten.

Die Herzimpulse werden auf ein sinusförmiges Signal der Frequenz von ca. 5.2kHz moduliert. Auf dem Markt existieren im Moment drei verschiedene Modelle von POLAR-Sendern, die pro Herzschlag ein bis max. drei aufeinanderfolgende Impulse aussenden, welche exponential abfallen. Der aufgebaute Empfänger ist ausgelegt für alle Sender – Modelle.

Für die Signalverarbeitung sind insgesamt drei Stufen erforderlich, welche im Folgenden genauer beschrieben werden.

8.2.2 Verstärker mit Tiefpass

Über einen LC-Schwingkreis werden die Signale vom Sender empfangen und durch einen nichtinvertierenden Verstärker auf einen Amplitudenpegel von etwa 40mV verstärkt. Die Antenne wird mit der folgenden Formel auf eine Resonanzfrequenz von 5.2kHz abgestimmt.

$$f_{\text{Resonanz}} = \frac{1}{2\eta \cdot \sqrt{L \cdot C}}$$

Der Verstärkungsfaktor kann mit R24 und R29 abgestimmt werden.

$$A = \frac{R24 + R29}{R29}$$

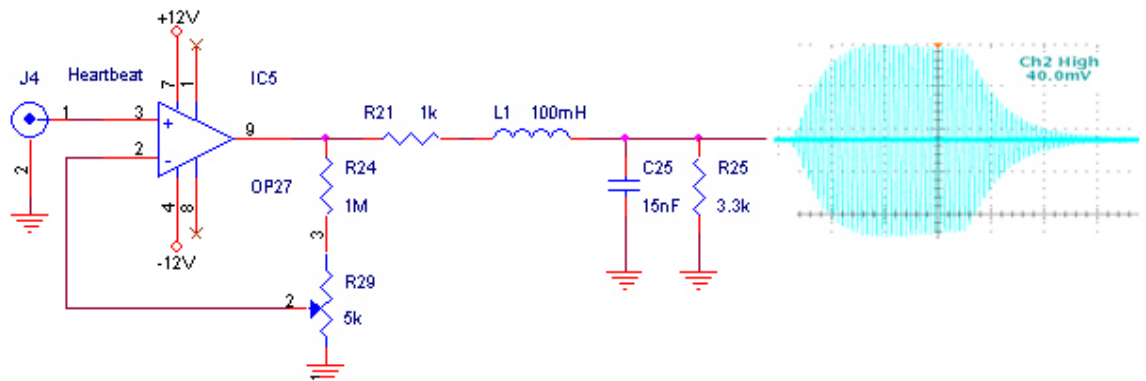


Abb 23 : Pulsempfänger mit Verstärker und Tiefpass

8.2.3 Komparator

Die empfangenen Spannungsamplituden sind je nach der Distanz Sender – Empfänger verschieden. Mit Hilfe eines Komparators werden alle Impulse auf den gleichen Wert gesetzt(11.2V). Mit R22 kann die Schwellspannung eingestellt werden.

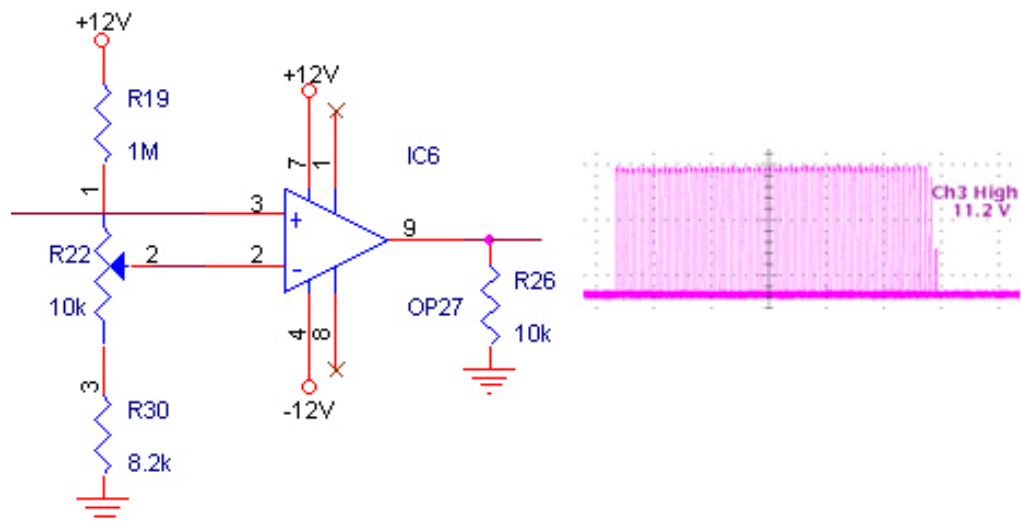


Abb 24 : Komparator

8.2.4 Astabiler Kippschwinger

Damit das Signal im MSP430 einfach weiterzuverarbeiten ist, muss sichergestellt werden, dass am Ausgang des Empfängers pro Herzschlag nur ein einziger Impuls liegt. Dies wird mit einem astabilen Kippschwinger erreicht. Wird dieser einmal angeregt schaltet er den Ausgang für eine bestimmte Zeit t auf High.

$$t = 1.1 \cdot R_{18} \cdot C_{24}$$

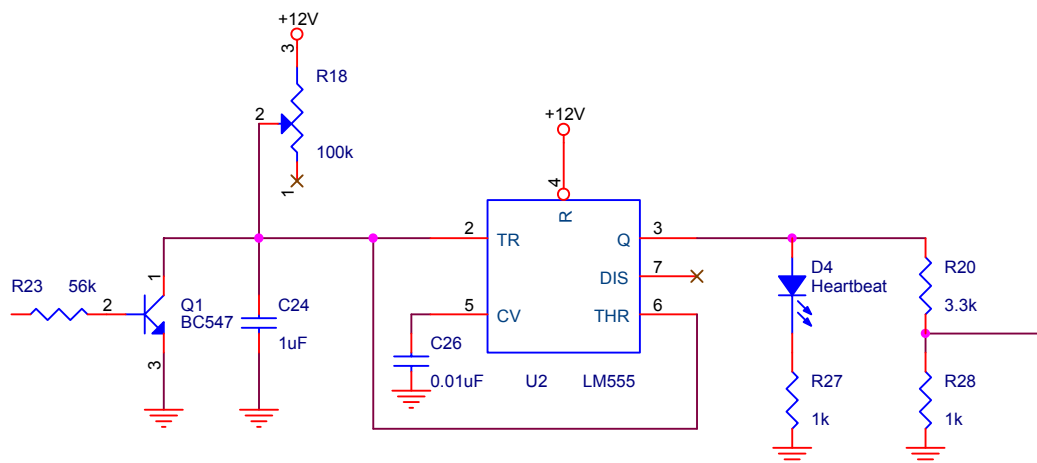


Abb 25 : Astabiler Kippschwinger, Monoflop

Die untenstehende Grafik zeigt zusammenfassend die Signale nach den jeweiligen Stufen. Der Ausgangspegel liegt auf einem konstanten Wert von 12V. Die Impulsdauer kann wie oben gezeigt eingestellt werden. Mit einer Dauer von 250ms könnte man also theoretisch Puls-Signale bis 240 Herzschläge pro Minute verarbeiten. Extremwerte vom menschlichen Herz liegen zwischen 30 – 220 Schlägen.

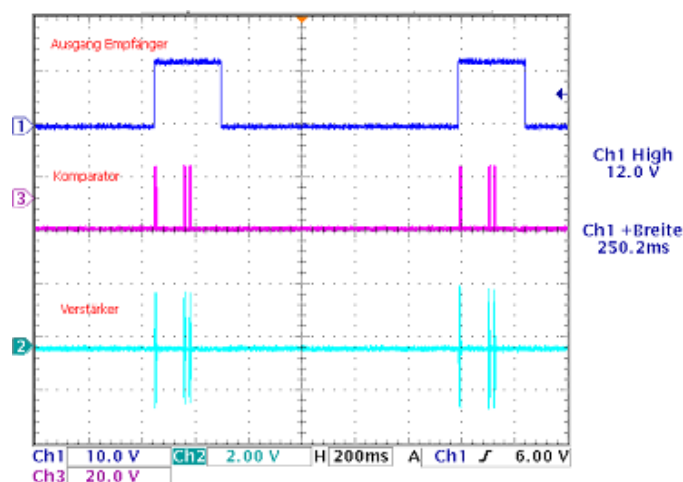


Abb 26 : Signale im Pulsempfänger

8.3 Leistungsmessung

Für die Berechnung der erbrachten Leistung und Arbeit werden zwei Messwerte benötigt, nämlich Strom und Spannung. Die 3-phasige Spannung am Spinningbike wird über einen Brückengleichrichter gleichgerichtet. Damit ist es einfacher die Spannung und den Strom zu messen. Mittels 1000:1 Teiler wird die Spannung auf 0-600 mV heruntergeteilt und eine aktive Stromzange liefert 100mV/A. Diese Werte werden mittels I2C-ADWandler an das Drill Control Center übermittelt. Die MSP Internen ADWandler sind sehr empfindlich auf Spannungsspitzen und dergleichen.

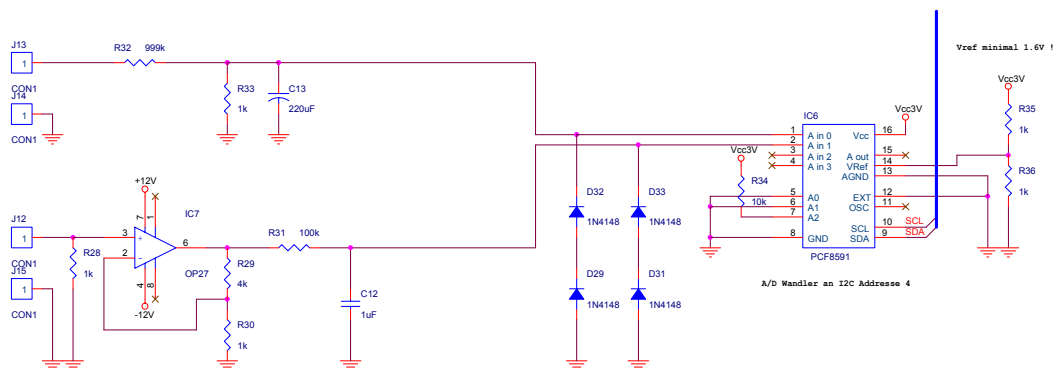


Abb 27 : AD-Wandler PCF8591

Der Baustein PCF8591 besitzt vier analoge Eingänge und einen Analogausgang. Über den Anschluss Vref kann die Referenzspannung eingespeist werden. Der ausgelesene Wert ist ein Verhältnis zu dieser Referenzspannung: z.B. 256 ist die Referenzspannung, 128 die Halbe - und 64 ein Viertel Referenzspannung. Der Baustein besitzt die I2C-Adresse 4.

8.4 Taster und Encoder

Für die Bedienung des DCC sind nur drei Komponenten erforderlich. Zwei unterhalb des Displays angeordnete Taster und ein seitlich angeordneter Encoder. Die Taster sind je mittig unter der linken und rechten Hälfte des Displays angebracht. Mittels Software können ihnen verschiedene Funktionen zugeordnet werden. Die entsprechende Funktion kann dann in der untersten Zeile des Display dargestellt werden. Der Encoder wird ebenfalls für unterschiedliche Funktionen genutzt. Zum Beispiel kann am Anfang beim Login mittels Encoder die Loginnummer eingestellt werden. Während dem Training kann die Sollleistung mittels Encoder manipuliert werden. Die Taster und der Encoder sind alle am Port 1 des MSP430 angeschlossen. Eine entsprechende Interrupt Service Routine stattet sie mit den verschiedenen Funktionen aus. Hardwareseitig sieht das wie folgt aus:

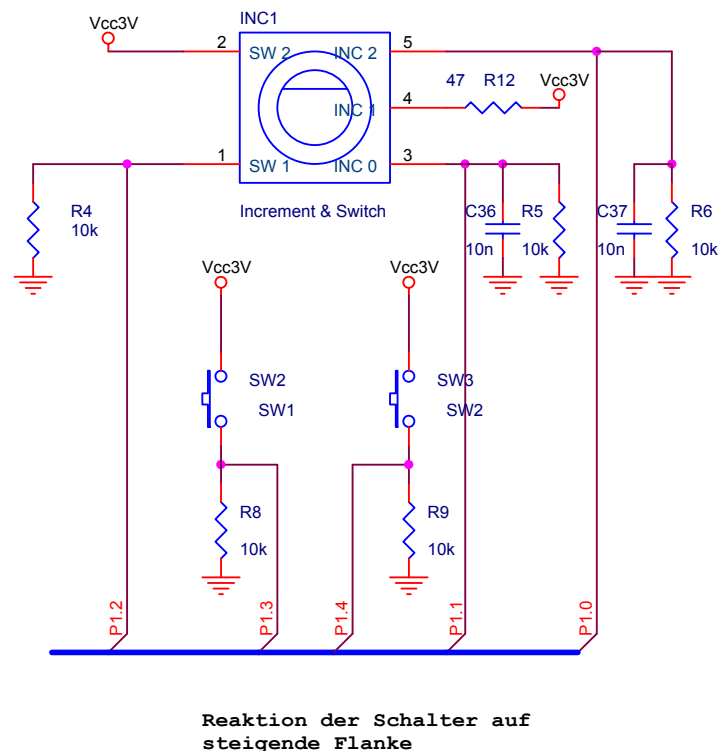


Abb 28 : Schema Taster & Encoder

Der Encoder verfügt ebenfalls über einen Taster, welcher zum Bestätigen der Eingaben genutzt werden kann. Auf dem Prozessorboard im DCC befindet sich noch ein Reset-Switch für den MSP430.

8.5 Kommunikation

Damit das DCC mit dem Hostrechner kommunizieren kann, benötigt es eine Schnittstelle. In der ursprünglichen Planung ist ein Bus für mehrere DCC vorgesehen. Aus Zeitgründen ist nur eine RS232 Schnittstelle implementiert. Denkbar wären eine RS485 oder eine Ethernet Schnittstelle. Die Pegelwandlung der 3.3Volt Pegel der MSP430 UART übernimmt ein MAX3232 Baustein, welcher die 3.3Volt Pegel auf ± 12 Volt anhebt. Die Beschaltung des Bausteins ist sehr einfach:

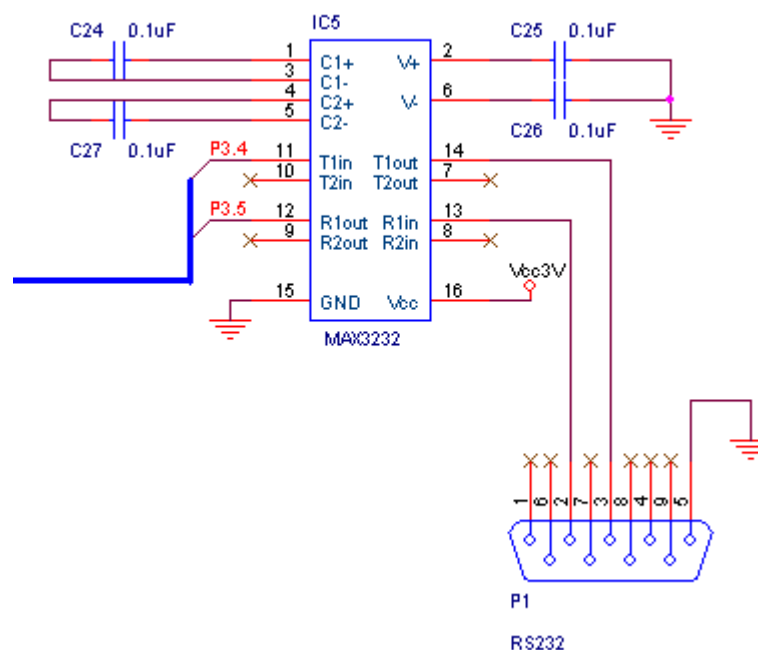


Abb 29 : Schema RS232 Treiber

Da kein Hardware Handshaking benötigt wird, sind nur die zwei Leitungen RX und TX verbunden.

Die Kommunikation erfolgt mittels rudimentärem Protokoll, welches nur über 16 verschiedene Pakete verfügt. Diese sind:

Master	CMD	SIZE	Befehl
PC / DCC	0000	xxxx	Reset der Kommunikation
PC / DCC	0001	0100	senden der Loginnummer
PC	0010	1111	senden des Namens
DCC	0011	0110	senden der Ist - Daten
PC	0100	0100	senden der Soll - Daten
DCC	0101	0000	Start freies Training
PC	0111	0000	Start Test
PC / DCC	1000	0000	Pause
PC / DCC	1001	0000	Stopp
PC	1010	0000	Nummer ungültig
PC	1011	1111	Textmessage an DCC
DCC	1100	0000	Init Test

Abb 30 : Kommandos für RS232 Kommunikation

Die Login Transaktion :

- DCC sendet die Loginnummer an den Host.
- Host sendet die Nummer und den entsprechenden Namen an DCC.

Die Trainingstransaktion :

- DCC sendet das „Start freies Training“ Command.
- DCC sendet alle 2 Sekunden ein Ist-Daten-Paket inkl. Nummer.
- DCC kann während des Trainings „Pause“ senden.
- DCC sendet am Schluss des Trainings „Stop“.

Der Aufbau der Datenpakete ist im Anhang dokumentiert.

8.6 Display

Als Anzeige kommt ein Display der Firma Marelcom zum Einsatz. Es besitzt 8KB RAM und kann 160 x 80 Pixel darstellen. Da das Display mit min. 4.5 Volt betrieben werden muss, müssen die Bussignale vom MSP430 von dessen 3.3 Volt Logik angepasst werden. Zwei Bausteine des Typs 74LVX4245 reichen, um den kompletten Bus bidirektional an den MSP430 anzukoppeln. Folgendes Bild verdeutlicht dies:

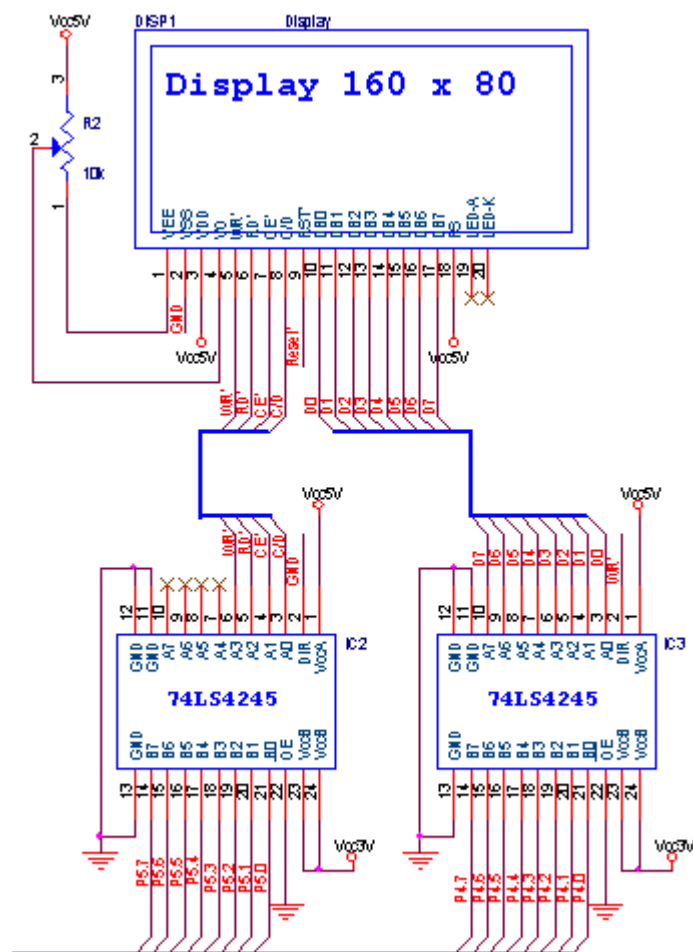


Abb 31 : Schema Display & Pegelwandler

Mittels Write - Signal wird die Richtung des Datenbusses IC3 bestimmt. IC2 ist immer in Richtung B -> A, da die Steuersignale immer Richtung Display gehen. Poti R2 ist für die Kontrasteinstellung zuständig.

Die 8 KB RAM sind wie folgt aufgeteilt:

0x0000	G R A P H I C 1
0x09FF	
0x0A00	G R A P H I C 2
0x13FF	
0x1400	G R A P H I C 3
0x1DFF	
0x1E00	T E X T
0x1F0D	
0x1F0E	F R E E & C G
0x1FFF	

Abb 32 : Memory-Map Display-RAM

Im Grafikbereich 1 werden alle grafischen Elemente der Standard-Ausgaben gespeichert und dargestellt. Im Grafikbereich 2 wird während der Trainingsphase die Leistungskurve aufgezeichnet. Im Grafikbereich 3 die Pulscurve. Der Textbereich ist für die Darstellung aller Textelemente der gleiche. Mit dieser Aufteilung sind noch 243 Bytes frei. In diesen 243 Bytes ist aber noch das interne Charakter Generator ROM integriert. Der restliche Platz könnte noch für eigene Zeichen genutzt werden.

9 JAVA Graphical User Interface

9.1 Analyse

Den Instruktoren von Spinninglektionen ist es oft nicht möglich alle Trainierenden optimal zu kontrollieren und zu betreuen. Es wäre also von Vorteil, wenn der Leiter zu jeder Zeit zentral und einfach Zugriff hat auf alle aktuellen Werte von jedem Einzelnen. Zudem ist es für den Trainierenden oft interessant, die erbrachten Daten im Nachhinein zu analysieren und mit früheren Einheiten zu vergleichen. Das PC Interface hat also die Aufgabe die gesamten Trainings- und User-Daten zu speichern und zu verwalten. Dazu ist eine einfach zu bedienende Benutzeroberfläche wünschenswert.

Wir haben uns entschieden die Daten über die serielle Schnittstelle RS 232 zu übertragen. Die Datenübertragung sollte in beide Richtungen möglich sein, damit z.B. auch ein Fitnessstest möglich ist, der über den PC gesteuert wird und die Leistungswerte vorgibt. Für die JAVA – Entwicklungsumgebung wurde der JBuilder 5 eingesetzt. Für die genaue Installation wird auf das Kapitel 9.5.1 verwiesen.

9.2 Anforderungen

Alle User sollten in einer Datenbank erfasst werden und eine zugehörige persönliche Loginnummer erhalten. Mit ihr sollte es möglich sein, sich über das Drill Control Center (DCC) einzuloggen und den richtigen Namen vom PC zurück aufs Display zu erhalten. Wir haben uns entschieden die Daten wie Zeit, Puls Trittfrequenz und Leistung alle 2 Sekunden auf den PC zu übertragen und zu speichern. In dieser Arbeit beschränken wir uns darauf, nur einen User auf einmal zu verwalten. In einer späteren Anwendung sollte es mit diesem Konzept trotzdem möglich sein, verschiedene Benutzer in Echtzeit gleichzeitig zu kontrollieren und zu überwachen. Daher wird den eigentlich gesendeten Daten jeweils ein Code angefügt, welcher neben der Art der übertragenen Daten auch über Benutzer Aufschluss gibt. Das folgende Diagramm zeigt die Kommandos für die RS232 Kommunikation.

M a s t e r	C M D	S I Z E	B e f e h l
P C / D C C	0 0 0 0	x x x x	Reset der K o m m u n i k a t i o n
P C / D C C	0 0 0 1	0 1 0 0	s e n d e n d e r L o g i n n u m m e r
P C	0 0 1 0	1 1 1 1	s e n d e n d e s N a m e n s
D C C	0 0 1 1	0 1 1 0	s e n d e n d e r I s t - D a t e n
P C	0 1 0 0	0 1 0 0	s e n d e n d e r S o l l - D a t e n
D C C	0 1 0 1	0 0 0 0	S t a r t f r e i e s T r a i n i n g
P C	0 1 1 1	0 0 0 0	S t a r t T e s t
P C / D C C	1 0 0 0	0 0 0 0	P a u s e
P C / D C C	1 0 0 1	0 0 0 0	S t o p p
P C	1 0 1 0	0 0 0 0	N u m m e r u n g ü l t i g
P C	1 0 1 1	1 1 1 1	T e x t m e s s a g e a n D C C
D C C	1 1 0 0	0 0 0 0	I n i t T e s t
D C C	1 1 0 1	0 0 0 0	C o n f i r m p o w e r

Abb 33 : Kommandos für RS232 Kommunikation

9.3 Design

Das Herzstück des Programms stellt die Klasse Interface dar, in welcher zusätzlich die Klassen sendeListener und commListener implementiert sind, welche automatisch aufgerufen werden, wenn am seriellen Port Daten anstehen oder gesendet werden müssen. Um mit dem Port kommunizieren zu können muss das Package javax.comm.* installiert werden. Die genaue Installations-Anleitung ist im Kapitel 9.5.1 zu finden. Der Code für die grafische Oberfläche ist weitgehend getrennt vom eigentlichen Programm. Die grafischen Teile befinden sich in den Klassen Grafik, TrainingGrafik und ArchivGrafik

9.4 Funktionalität

Wie anfangs erwähnt sollte die Benutzeroberfläche von der Bedienung her so einfach als möglich gestaltet werden und dennoch alle wichtigen Funktionen enthalten. Der Benutzer hat grundsätzlich die Auswahl zwischen drei verschiedenen Hauptfunktionen.

1. Training (Daten vom Spinningbike erfassen – Regelung über DCC)
2. New User (neue User haben sich am PC einmalig zu registrieren)
3. Archiv (alle Daten können im Archiv angesehen werden)

Beim Start des JAVA Programms erscheint die eigentliche Startseite, welche den Ausgangspunkt zu allen weiteren Funktionen darstellt.



Abb 34 : Java GUI Willkommens - Bildschirm

9.4.1 New User

Möchte eine Person auf dem Spinningbike trainieren, so hat sie sich ein einziges Mal am PC zu registrieren. Durch das Anwählen von ‚New User‘ gelangt sie auf die untenstehende Seite. Der vollständige Name ist im Feld einzutragen. Die Eingabe muss mit ‚Registry‘ bestätigt werden.



Abb 35 : Java GUI Benutzer registrieren

Alle User werden im zentralen File ‚UserDaten.csv‘ gespeichert und verwaltet. Beim Betätigen des ‚Registry‘ Buttons wird jedem neuen User automatisch die tiefste noch freie Loginnummer zugeteilt. Zusätzlich wird ein Ordner mit dem Namen des Users erstellt, welche sich im Verzeichnis D:\Drill_Control\Users befinden. Alle persönlichen Trainingsdaten werden später in diesem Ordner abgelegt. Die nachfolgende Darstellung illustriert den Ablauf grafisch.



Abb 36 : Java GUI Bestätigung Registration

Der Benutzer wird aufgefordert seine persönliche Loginnummer zu notieren, damit er sich zukünftig mit dieser auf dem Spinningbike einloggen kann. Das ‚Message – Fenster‘ verschwindet erst nach Betätigung des OK Buttons. Mit ‚again‘ kann direkt ein weiterer Benutzer eingegeben werden. Mit ‚ok->exit‘ gelangt man zurück zum Auswahlmenu.

9.4.2 Free Training

Mit dem Betätigen des ‚Training‘ – Buttons öffnet der PC den seriellen Port ‚COM1‘. Er ist nun bereit Daten zu senden und zu empfangen. Der PC verweilt in dieser Position bis sich am DCC ein Benutzer einloggt. Die Kontrolle über den PC wird von diesem Moment an ausschliesslich vom DCC gesteuert. Bis ein freies Training möglich ist laufen folgende Schritte ab:

1. Benutzer loggt sich mit persönlicher Nummer ein und wartet auf seinen Namen. Wird die Nummer 0000 gesendet reagiert der PC nicht. Damit ist ein Training ohne Datenaufzeichnung möglich.
2. PC sucht zur gesendeten Nummer den richtigen Namen im File UserDaten.csv.
3. PC sendet den Namen als String zurück zum DCC, welches den Vornamen grafisch auf dem Display darstellt.
4. Der Benutzer kann zwischen 3 Trainingsprogrammen entscheiden.
 - Free Training
 - Trainingsprogramm
 - Test

Wird der Modus „Free Training“ gewählt, zeichnet der PC die Trainingsdaten alle 2 Sekunden auf und speichert sie im bereits persönlich erstellten Ordner. Der File-Name trägt die Art des Trainings sowie das aktuelle Datum mit Zeit, z.B. Training_2._Dezember_2002_13.10Uhr.csv. Zusätzlich werden die aktuellen Daten auf der PC Oberfläche grafisch dargestellt.

5. Die Aufzeichnung der Daten erfolgt so lange, bis der Benutzer das Training unterbricht.

Der gesamte Ablauf wird auf der nächsten Seite grafisch dargestellt.

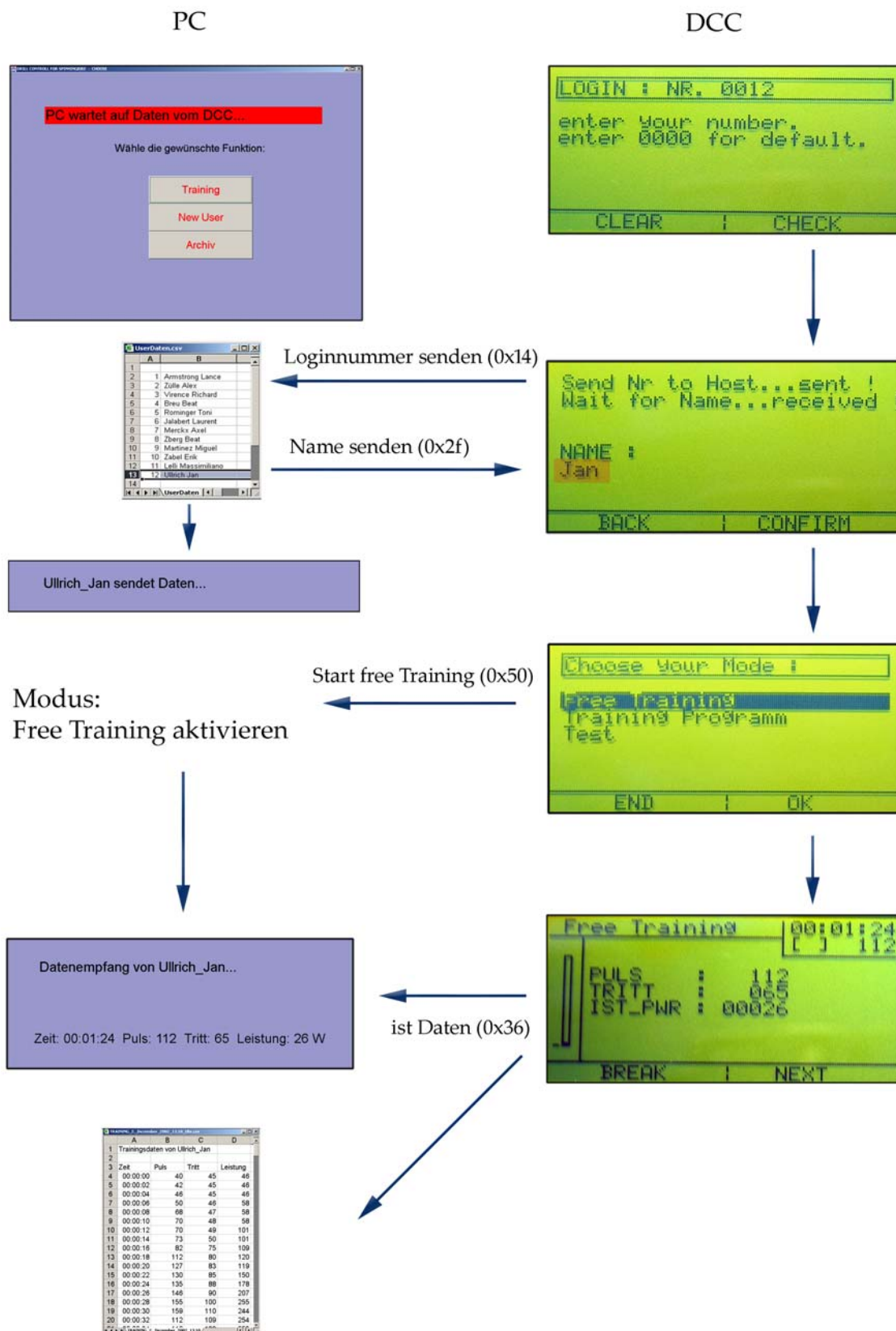


Abb 37 : Java GUI Ablauf "Free Training"

9.4.3 Conconitest

Herr Conconi und seine Mitarbeiter stellten 1982 anlässlich des Weltkongresses für Sportmedizin ein Testverfahren zur Ermittlung der anaeroben Schwelle bei Läufern und Radfahrern vor. Der Test basiert auf dem Phänomen, dass sich die Herzfrequenz in weiten Bereichen linear zur erbrachten Leistung verhält, bei weiter steigender Belastung jedoch abflacht. Dieser Abknick wird auch Deflektionspunkt genannt. Nach Conconi stimmt die dabei erreichte Herzfrequenz mit der anaeroben Schwelle überein.

Was ist die Anerobe Schwelle?

Die Energiebereitstellung beim Radfahren kann sowohl mit ausreichend Sauerstoff (aerob), als auch unter Sauerstoffmangel (anaerob) ablaufen. Je schneller die Bewegungsgeschwindigkeit, um so mehr Sauerstoff wird zur Energieverbrennung benötigt. Irgendwann kann der Organismus nicht mehr ausreichend Sauerstoff aufnehmen. Es wird fortan Energie unter Sauerstoffmangel freigesetzt. Diese Belastung kann nur über eine sehr begrenzte Zeitdauer durchgeführt werden. Der Übergangsbereich vom Zustand der ausreichenden Sauerstoffversorgung (aerobe) zum Sauerstoffmangel (anaerob) wird als aerob-anaerobe Schwelle bezeichnet. Für einen Stundenweltrekord ist die Fahrgeschwindigkeit im Bereich der aerob-anaeroben Schwelle von grosser Wichtigkeit. Der Stundenweltrekord wird in den ersten 50 Minuten exakt auf dieser Schwelle gefahren, die restlichen 10 Minuten leicht im anaeroben Bereich. Dieser Übergangsbereich lässt sich am einfachsten mit einem Conconitest auf einem Spinningbike oder Hometrainer ermitteln. Die Herzfrequenz liegt bei 90-95% von H_{fmax} .

Testablauf:

Nach einer Aufwärmphase mit 25W / 50W wird der eigentliche Test gestartet. Die Belastung wird alle 3 Minuten um 25 Watt erhöht. Um ein erfolgreiches Testergebnis zu erzielen, sollte die Trittfrequenz während dem ganzen Ablauf auf ca. 80U/min gehalten werden. Der Test ist beendet, wenn die Person ihren Maximalpuls erreicht hat. Die Daten von Puls und Leistung werden nachher auf dem Computer ausgewertet.

Programmablauf:

Sobald der Benutzer den ‚Test‘ Modus mit ‚OK‘ bestätigt, wird das Kommando ‚init Test‘ (0xb0) vom DCC an den PC gesendet. Der PC schaltet in den Conconitest – Modus. Sobald dieser bereit ist, sendet er ein ‚Start Test‘ (0x70) an das DCC. Somit weiss das DCC, dass der PC bereit ist und gibt dem PC ebenfalls ein ‚Start Test(0x70)‘ zurück. Von diesem Moment an übernimmt der PC die volle Kontrolle über den gesamten Testablauf. Der Trainierende hat nun keine Möglichkeit mehr die Last manuell einzustellen, damit der Conconitest erfolgreich ablaufen kann.

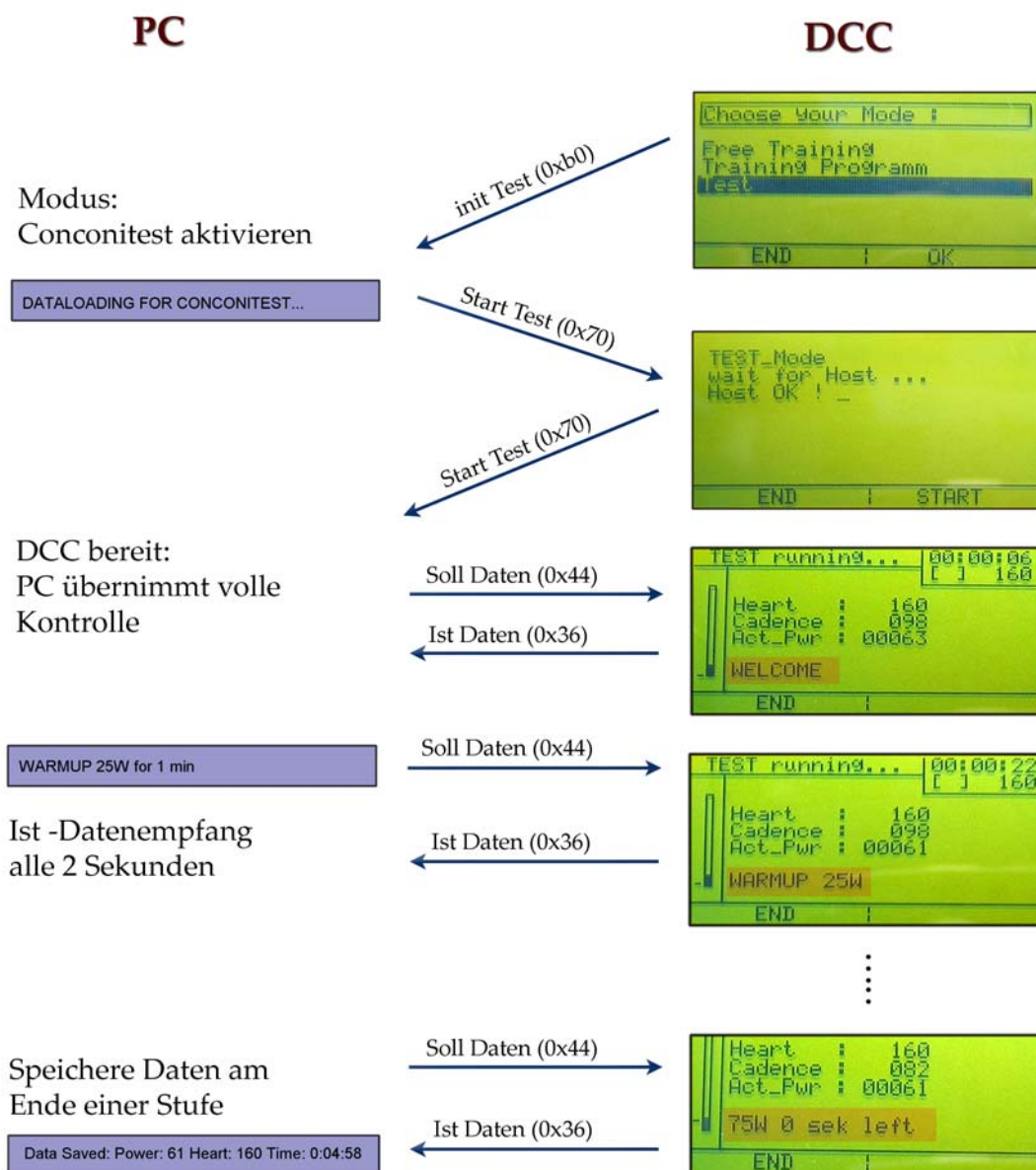


Abb 38 : Java GUI Ablauf "Conconi Test"

Über das Display wird der Testkandidat fortlaufend über den Testverlauf informiert. Wie anfangs erwähnt, erhöht sich die Leistung alle 3 Minuten um 25 Watt. Am Ende einer Stufe werden die erbrachten Leistungs - Daten auf dem PC gespeichert. Damit der Test aussagekräftig wird, ist es sinnvoll einen Durchschnittswert der erbrachten Leistung zu speichern. Gegen Ende einer Stufe werden 50 Leistungswerte zwischengespeichert und anschliessend gemittelt. Weil der menschliche Kreislauf etwas verzögert auf eine Anstrengung reagiert, werden nur 4 Pulswert am Ende einer Stufe aufgenommen und gemittelt. Nach Ablauf einer Einheit werden die Daten im persönlichen Ordner mit dem Namen zB. CONCONI_3.Dezember_2002_12.34Uhr.csv gespeichert und auf dem PC-Bildschirm kurz angezeigt.

Fehlererkennung:

Im Programm sind einige Methoden zur Fehlererkennung implementiert. Damit wird sichergestellt, dass der Test am Schluss erfolgreich ausgewertet werden kann und möglichst keine verfälschten Daten gespeichert werden. Die Fehler-Anzeige erfolgt ungeachtet dessen, was zuvor auf der Zeile ausgegeben wurde – Fehlermeldungen haben 1. Priorität. Folgende Erkennungen sind aktiv:

1. Der Test beginnt nur, wenn während den ersten 10 Sekunden durchgehend ein Pulssignal von über 30 Schläge/Minute empfangen wird. Auf dem Display erscheint ‚STOP – NO HEART RATE!‘



2. Fällt der Puls während dem Testablauf unter 40 Schläge / Minute, deutet dies auf einen Empfangsfehler hin. Auf dem Display erscheint: „CHECK PULS!!“



3. Wird die Soll – Trittfrequenz von 80U/min nicht eingehalten, erscheint bei einer Kadenz von >75 und <85 die Meldung: ‚HOLD 80U/MIN‘ im 2 Sekunden Takt wechselnd mit der momentanen Ist-Kadenz.



9.5 Entwicklungsumgebung

9.5.1 Vorbereitung / Installation

Die Installation des JBuilders erfordert keine grosse Mühe. Um mit JAVA mit dem seriellen Port kommunizieren zu können muss das Paket java.comm installiert werden. Das gesamte JAVA Communication API kann von der SUN-Homepage www.java.sun.com heruntergeladen werden.

Das Programm speichert alle User – Odner / Files unter folgendem Verzeichnis: D:\Drill_Control\Users - Dieser Pfad ist zu Beginn anzulegen, oder im Programm entsprechend anzupassen.

9.5.2 Besonderheiten

JAVA stellt eine Vielzahl von bereits vordefinierten Paketen mit ihren Klassen zur Verfügung, welche direkt verwendet werden können. Einige Schwierigkeiten ergaben sich durch die Fehlerbehandlung mit Exceptions. Ausnahmen sind Fehler, die den normalen Programmablauf unterbrechen, wie zB das Lesen und Schreiben eines Files. Die Exceptions können mit Hilfe des ‚catch‘ – Kommandos abgefangen werden.

```
try {  
    // Code, der evtl. Exceptions erzeugt  
} catch (Exceptions1 e1) {  
    // Fehler 1 behandeln  
} catch (Exception2 e2) {  
    // Fehler 2 behandeln
```


9.5.3 Archiv

Wie der Name schon sagt, können im Archiv alle gespeicherten Trainingsdaten angesehen werden.

Wird im Auswahlménü der 'Archiv' Button betätigt, öffnet sich automatisch das Programm Excel. Nach einem „Welcome - Fenster“ öffnet sich der Ordner mit den Usern, wo alle Trainings- und Test-Daten gespeichert sind.

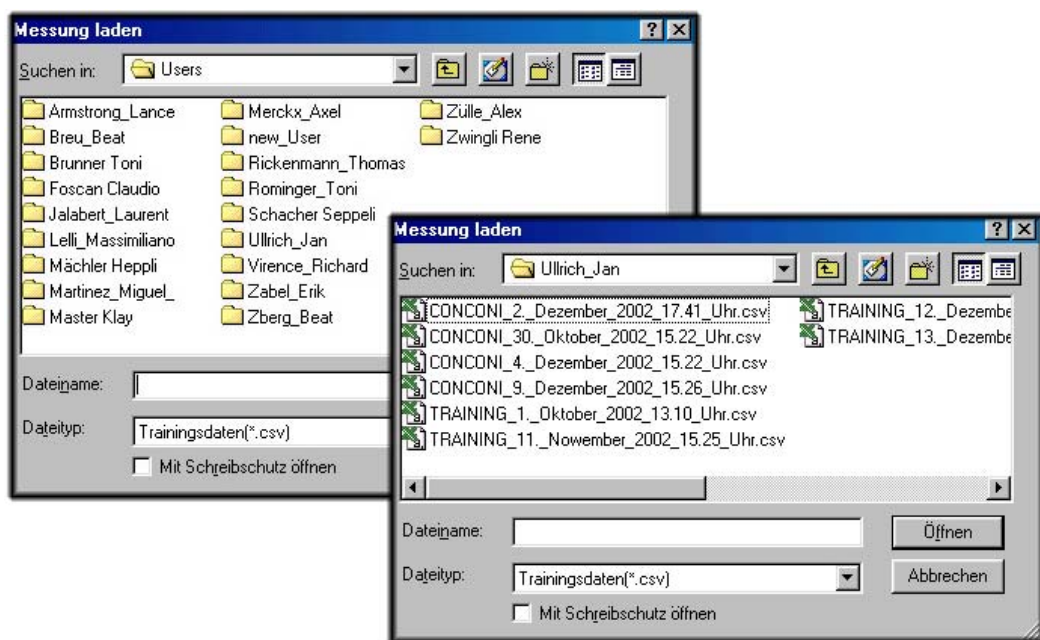


Abb 39 : Auswertung der Trainings – und Testdaten

Mit einem speziell definierten Excel - Makro lässt sich jede Trainingseinheit ohne zusätzliche Schritte automatisch darstellen. Wie das nächste Bild zeigt ist der selbst durchgeführte Conconitest erfolgreich verlaufen. In der x - Achse sind die Leistungswerte und in der y-Achse die zugehörige Pulsfrequenz aufgetragen. Der Knick, oder auch Deflektionspunkt genannt, ist aus der Grafik gut ersichtlich. Die Anerobe Schwelle liegt bei der Testperson also bei einer Pulsfrequenz von ca. 172 Schläge pro Minute und 218 Watt Leistung. Mit diesen Angaben sind nun alle Intensitätsstufen für diese Person bestimmt. Unter Pulskontrolle ist ein späteres Training optimal möglich.

Wird ein Trainingsfile geöffnet, wird die zugehörige Kurve mit Angaben von Puls, Trittfrequenz und Leistung in Abhängigkeit der Trainingszeit aufgezeichnet.

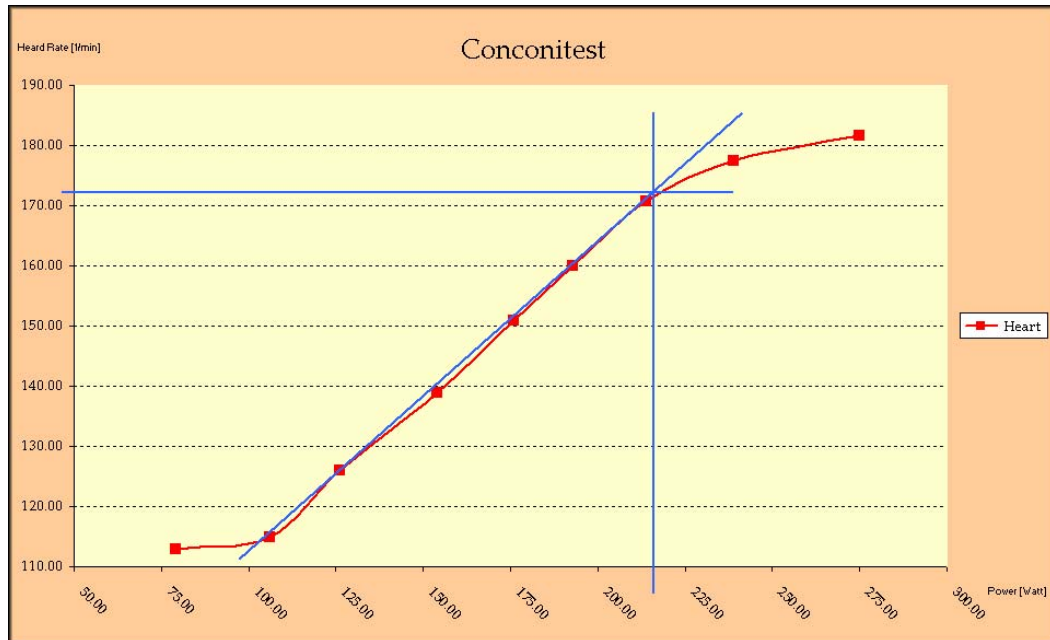


Abb 40 : Excel Auswertung Conconi Test

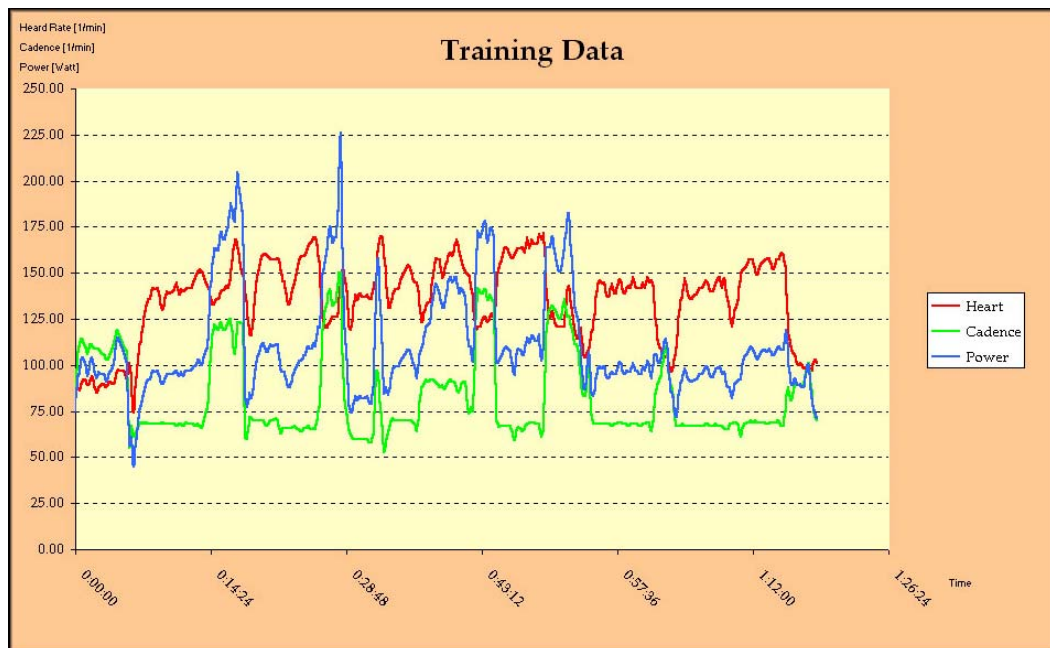


Abb 41 : Excel Auswertung Trainingseinheit

10 Embedded Software

Die Software für den MSP430F149 wurde vollständig in C geschrieben. Aufgrund des Speichermangels wird kein Betriebssystem eingesetzt. Statt dessen ist die komplette Ablaufsteuerung in C mittels einer StateMachine implementiert. Für die Komponentensteuerung stehen verschiedene Module zur Verfügung. Jede Peripheriekomponente ist einzeln als Modul implementiert worden.

Für das ganze Projekt kam die Entwicklungsumgebung AQ430 der Firma Quadravox zum Einsatz. Die Umgebung besteht aus einem Projektmanager, einem Codeeditor, Compiler Assembler Linker und einem Debugger.

10.1 StateMachine

Im main() Programm kommt neben der ganzen Interruptsteuerung eine StateMachine zum Einsatz. In ihr ist die ganze Ablaufsteuerung implementiert. Wenn der Benutzer z.B. vom Welcome Bildschirm auf Login wechselt, wechselt er den State. Je nach State sind verschiedene Operationen möglich.

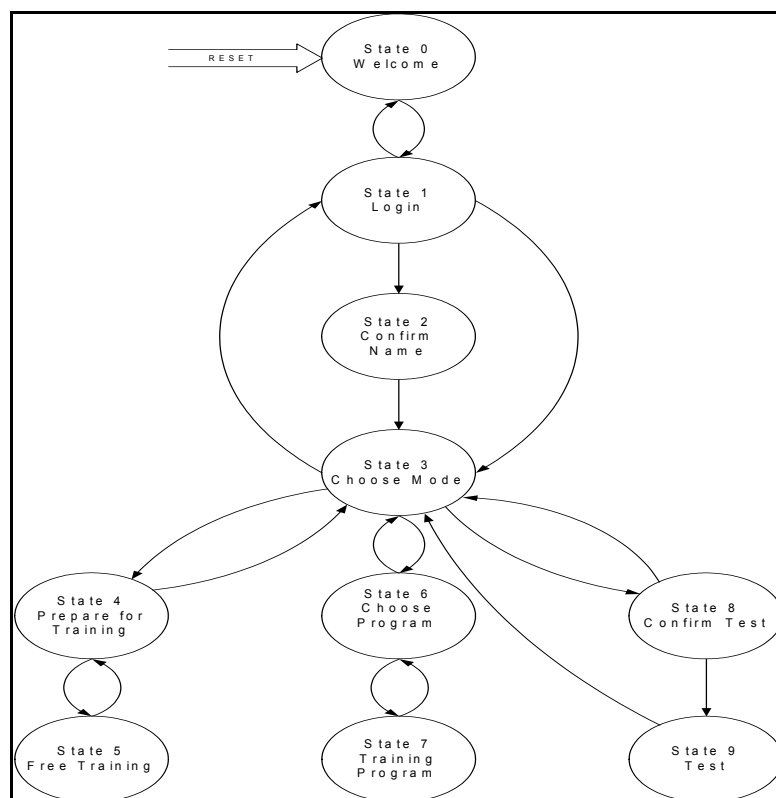


Abb 42 : StateMachine

10.1.1 State 0 Welcome

Reset State - zeigt den Willkommensbildschirm.

Encoder : keine Funktion

Taster 0 : keine Funktion

Taster 1 : Login

10.1.2 State 1 Login

Login Bildschirm. Über den Encoder kann die eigene Nummer eingegeben werden. Mittels Cancel-Taste kann die Nummer gelöscht und nochmals eingegeben werden. Wird innerhalb von 20 Sekunden kein Confirm ausgelöst, wechselt das DCC automatisch (mit Hilfe des Watchdogs) in den State 0.

Encoder : ändert die Zahlen der Nummer

Taster 0 : Cancel-Taster, löscht die Nummer

Taster 1 : Confirm-Taster, startet Anfrage an Host PC

10.1.3 State 2 Confirm Name

Abfrage der Nummer auf Host PC. Ist die Nummer vorhanden, sendet der PC den Namen des Trainierenden zurück an das DCC. Falls keine Nummer eingegeben worden ist, sprich „0000“, wechselt das DCC automatisch in State 3. Empfängt das DCC einen Namen, zeigt es den Namen an und verlangt nach einer Bestätigung. Wird der Name bestätigt, wechselt das DCC in den State 3.

Encoder : keine Funktion

Taster 0 : zurück zum Login State 1

Taster 1 : Bestätigung des Namens, wechseln in State 3

10.1.4 State 3 Choose Mode

Auswahl der Trainingsmethode. Zur Auswahl stehen: „Free Training“, „Training Programm“ oder „Test“. „Free Training“ führt zum State 4, „Training Programm“ zum State 6 und „Test“ zum State 8. Beendet man die Auswahl, folgt State 1 Login.

Encoder : Auswahl des Menüpunktes

Taster 0 : Ende der Auswahl, State 1 Login

Taster 1 : Bestätigung der Auswahl, State 4, State 6 oder State 8

10.1.5 State 4 Prepare for Training

Warte State, Trainings -Velo befindet sich in Wartestellung und gibt das Training frei, wenn Start gewählt wird. Falls von einem „Free Training“ in den Warte-State gewechselt wird, fährt die Senke bis zur Maximalleistung hoch und bremst so die Schwungmasse. Wird das Training beendet, wechselt das DCC in den State 3. Im State 4 ist der Timer in Pausenstellung und läuft erst weiter wenn in State 5 gewechselt wird.

Encoder : keine Funktion

Taster 0 : Ende des „Free Training“, State 3

Taster 1 : Start des „Free Training“, State 5

10.1.6 State 5 Free Training

„Free Training“ State. In diesem State läuft der Timer, die Spannungsregelung, die Leistungsregelung via Encoder, die Puls - und Trittempfänger. Das DCC sendet alle 2 Sekunden die Ist-Daten an den Host PC. Es kann zwischen 3 Fenstern hin und her gewechselt werden. Im ersten Fenster werden alle Daten im Textmodus angezeigt. Die Ist- und Soll-Leistungen werden immer angezeigt. Im „Puls Curve“- Fenster wird ein Verlauf des Pulses über 3 Minuten und 20 Sekunden angezeigt. Sobald die Kurve voll ist, fängt sie von vorne an. Im „Power Curve“ - Fenster wird ein Verlauf der Ist-Leistung angezeigt. Mit dem Encoder kann jederzeit die Soll-Leistung beeinflusst werden. Je grösser die Soll-Leistung, desto grösser das Moment an den Pedalen. Wird in den State 4 gewechselt, bremst das Velo automatisch ab.

Encoder : Einstellung der Soll-Leistung

Taster 0 : beendet das Training, State 4

Taster 1 : wechselt zwischen den Fenstern

10.1.7 State 6 Choose Training Programm

Auswahl des Trainingsprogramms. Mittels Encoder kann hier das Trainingsprogramm ausgewählt werden. Je nach Auswahl werden im darauf folgenden Training die Sollwerte der Leistung anders gesetzt. Falls die Auswahl beendet wird, wechselt das DCC in State 3. Ist die Auswahl getroffen und bestätigt, folgt State 7.

Encoder : Auswahl des Trainingsprogramms

Taster 0 : beendet das Auswahlmenu

Taster 1 : startet das ausgewählte Trainingsprogramm

10.1.8 State 7 Training Programm

Ist analog zum State 4, ausser dass nicht mehr mittels Encoder die Soll-Leistung geändert werden kann. Während dem Trainingsprogramm wechselt das DCC selbständig die Soll-Leistung wie es das ausgewählte Programm vorschreibt. „Power Curve“ und „Puls Curve“ stehen weiterhin zur Verfügung.

Encoder : keine Funktion

Taster 0 : beendet das Trainingsprogramm : State 6

Taster 1 : wechselt zwischen den Ansichten

10.1.9 State 8 Confirm Test

In diesem State wartet das DCC auf die Bestätigung des Host PC. Sobald der Host PC sich gemeldet hat kann der Test beginnen.

Encoder : keine Funktion

Taster 0 : beendet die Abfrage

Taster 1 : startet den Test

10.1.10 State 9 Test

Beim Test übernimmt der Host PC einen Teil der Kontrolle über das DCC. Er kann die Soll-Leistung vorgeben und Textnachrichten auf dem Display anzeigen. Dadurch ist es möglich individuelle Trainingsprogramme oder Leistungstests zu fahren. Es ist nur ein Fenster vorhanden in dem die aktuellen Daten angezeigt werden.

Encoder : keine Funktion

Taster 0 : beendet den Test

Taster 1 : keine Funktion

10.2 **Module**

10.2.1 I2C Library

Bibliothek für die Steuerung des I2C Buses an Port 2 des MSP430F149

Funktionen :

- void i2cinit(void); // initialisierung
- unsigned char i2csend(unsigned char); // sendet daten
- unsigned char i2creceive(unsigned char*, unsigned char); // empfängt daten
- void i2cstart(void); // start kondition
- void i2cstop(void); // stopp kondition

10.2.2 T6963C-Displaycontroller Library

Bibliothek für die Ansteuerung des T6963 Controllers vom Display.

Funktionen :

- unsigned char WriteXY(unsigned char* string, unsigned char x, unsigned char y, unsigned char groesse, short texthome); //schreibt string auf koordinaten x,y
- unsigned char ClearText(short texthome); //löscht ganzes textfeld
- unsigned char ClearLine(unsigned char zeile, short texthome); //löscht text zeile
- unsigned char SoftKey(unsigned char* sktext, unsigned char nr, short texthome, short graphichome); //schreibt den text für die softkeys.
- unsigned char ClearGraphic(short graphichome); //löscht grafikbereich
- unsigned char ClearGraphicLine(unsigned char line, short graphichome); //löscht eine linie im grafikbereich 1
- unsigned char HLine(unsigned char x, unsigned char y, unsigned char length, short graphichome); //zeichnet horizontale Linie
- unsigned char VLine(unsigned char x, unsigned char y, unsigned char laenge, short graphichome); //zeichnet vertikale Linie
- unsigned char SetPixel(unsigned char x, unsigned char y, short graphichome); //zeichnet einen pixel
- unsigned char ClearPixel(unsigned char x, unsigned char y, short graphichome); //löscht einen pixel
- unsigned char ClearScreen(short texthome, short graphichome); //löscht den bildschirm

Zusätzlich stehen weitere Funktionen für die Konfiguration des Displays zur Verfügung. Siehe Anhang.

10.2.3 PCF8591 A/D-Wandler Library

Bibliothek stellt Funktion zum Einlesen eines Analogwertes zur Verfügung.

Funktion :

- `unsigned char GetAnalog(unsigned short*, unsigned char);` // liefert analogwert (8Bit) von i2c-adc

10.2.4 PCF8574 Portexpander Library

Bibliothek zur Ansteuerung der Portexpander PCF8574 in der Senke.

Funktionen:

- `unsigned char setrelais(unsigned char, unsigned char);` // setzt ausgang via i2c-bus
- `unsigned char resetpexp(unsigned char);` // setzt ausgang auf 0x00
- `unsigned char resetallpexp(void);` // setzt alle ausgänge auf 0
- `unsigned char setrelaisonly(unsigned char, unsigned char);` // setzt ein einzelnes relais und lässt die anderen wie sie sind
- `unsigned char resetrelaisonly(unsigned char, unsigned char);` // löscht ein einzelnes relais und lässt die anderen wie sie sind

10.2.5 UART Library

Bibliothek mit den Funktionen zur Kommunikation mit dem Host PC

Funktionen:

- `void RSSend(unsigned char zeichen);` //sendet zeichen universal
- `void RSSendNummer(char* t_nummer);` //sendet nummer an host => login
- `void RSSendData(char* t_nummer, short t_zeit, char t_puls, char ttritt, char t_leistung);` //sendet ist-daten an host
- `void RSSendConfirmPower(char* t_nummer);` //bestätigung der soll leistung von host

10.2.6 PowerControl Library

Bibliothek mit den Funktionen zur Kontrolle der Senke

Funktionen :

- unsigned char SetPower(unsigned char leistung); // stellt die gewünschte leistung ein
- unsigned char SetVoltage(unsigned char spannung); // stellt den gewünschten spannungsbereich ein
- void BreakPower(unsigned char leistung); // fährt die leistung kontinuierlich hoch, und stellt nach time x nominalleistung wieder her, zum bremsen

10.2.7 View Library

Bibliothek mit den Funktionen zur Darstellung der Menustrukturen und der Kurven während Trainingsfahrten

Funktionen :

- unsigned char UpdatePowerIndicator(void); //veraendert powerbalken auf allen views
- unsigned char UpdateTimerView(void); //veraendert die timeranzeige auf allen views
- unsigned char UpdatePulsView(void); //update pulsfrequenz
- unsigned char UpdateTrittView(void); //update trittkadenz
- unsigned char UpdatePowerViwe(void); //update ist_power
- unsigned char UpdatePowerCurve(unsigned short time, unsigned short power); //update leistungskurve
- unsigned char UpdatePulsCurve(unsigned short time, unsigned char puls); //update pulskurve
- unsigned char UpdateFrameView(void); //update gitter geruest
- void Init_Logo(unsigned char x, unsigned char y); //schreibt logo in den speicher

10.3 *Entwicklungsumgebung*

10.3.1 Installation

Die Installation ist unproblematisch. Einfach das auf der CD enthaltene Zip-File entpacken und „Setup.exe“ ausführen. Beim ersten Start muss noch die Lizenznummer eingegeben werden. Danach installiert die Software noch einen Parallelport Treiber und nach einem Neustart funktioniert alles.

10.3.2 Neues Projekt

1. File , New

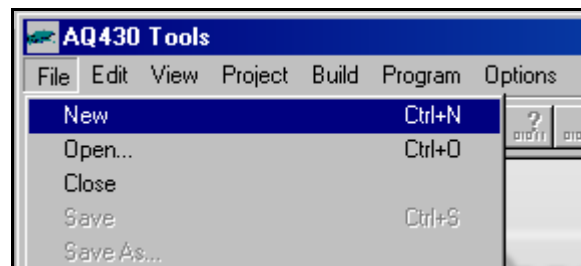


Abb 43 : AQ430 File,New

2. New Projectfile

- Project File auswählen und bestätigen.

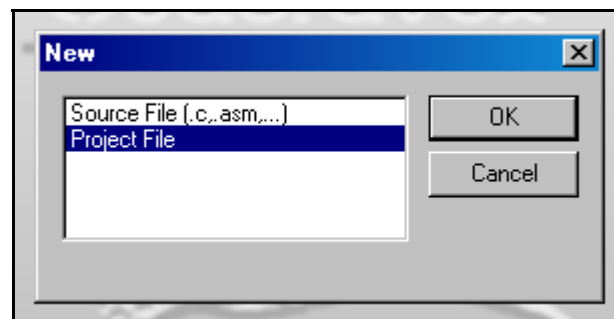


Abb 44 : AQ430 New Project File

3. Projekteinstellung

- Name und Pfad wählen, den richtigen Chip auswählen.

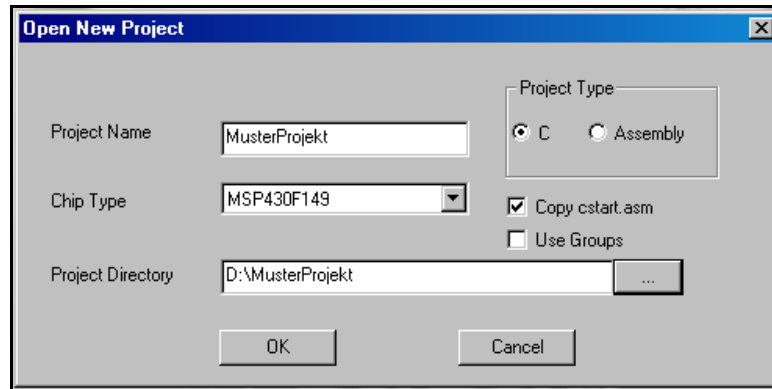


Abb 45 : AQ430 Projektsettings

4. Projektmanager erscheint

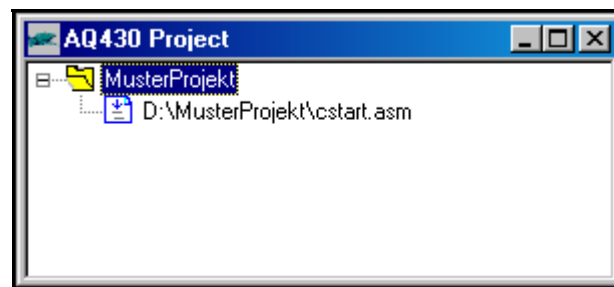


Abb 46 : AQ430 Projektmanager

4. Dateien hinzufügen oder neue Dateien erstellen.

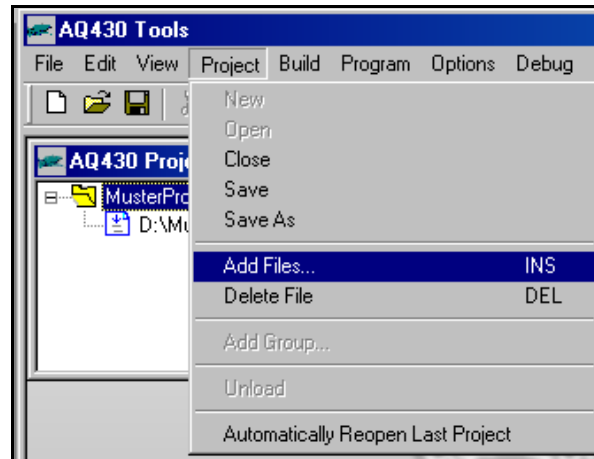


Abb 47 : AQ430 AddFiles

6. Projekt compilieren und übertragen in MSP430

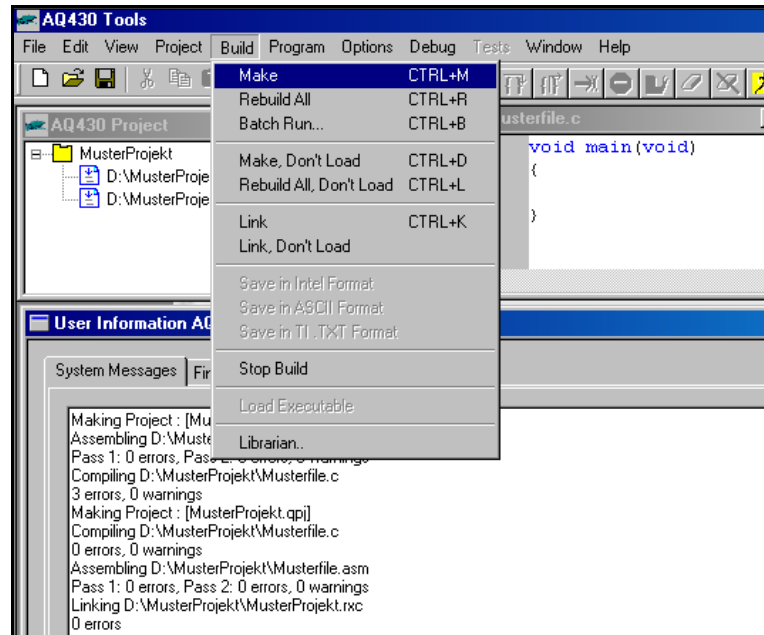


Abb 48 : AQ430 Compile and Transfer

10.3.3 Probleme

1. Der Code Editor hat eine schlechte farbige Kennzeichnung der Syntax. Es empfiehlt sich einen externen Editor (z.B. TextPad oder UltraEdit) einzusetzen.
2. Wird während dem Debuggen des Programms ein Hardware-Reset am Target durchgeführt, stürzt die ganze Entwicklungsumgebung ab.
3. Mehrdimensionale Felder können während dem Debuggen nicht im Watch-Window gesichtet werden.

11 Danksagung

Wir bedanken uns für die selbstlose Unterstützung bei folgender Personen:

- | | |
|-------------------------|----------------------------|
| • Herr Erwin Brändle | Betreuer der Diplomarbeit |
| • Herr Caspar Naef | Chef ES-Labor |
| • Herr Remo Sennhauser | Lehrling ES-Labor |
| • Herr Ryffel Rudolf | Werkstattchef Maschinenbau |
| • Herr Kurt Schoch | Mitarbeiter EAT-Labor |
| • Frau Susi Rickenmann | Lektorin |
| • Herr Hanspeter Foscan | Lektor |

12 Fazit der Diplomarbeit

Wir haben erreicht was wir uns vorgenommen hatten. Sicher gibt es noch viele Punkte an denen man noch Verbesserungen anbringen könnte, aber in 7 Wochen fehlte schlicht die Zeit dafür. Die Senke, welche wir auch noch konstruieren musste verschlang zu viel Zeit. Am Ende entstand ein funktionierendes Trainingsgerät inkl. Software für einen Fitnessstudio.

Die Arbeit gestaltete sich sehr Spannend. Wie verbinde ich Hochspannung und Niederspannung, wie steuere ich eine Last, welche Bedienelemente sind am geeignetsten um während eines Trainings Einstellungen vorzunehmen usw. Solche Fragen musste ich mich stellen. Dadurch erhielt ich aber Einblicke in Themen die mir nicht so vertraut waren. Man lernt nie aus! Ich denke aber, genau dass macht einen Ingenieur aus, sich auch Aufgaben und Situationen die er nicht kennt erfolgreich zu stellen. Mein Fazit: es ist nicht einfach ein funktionierendes HMI (in Funktion und Benutzerfreundlichkeit) zu konstruieren. Eine gute Projektplanung am Anfang ist Gold Wert. Die Arbeit hat viel Spass gemacht, war aber leider in der Zeit zu kurz bemessen.



Hätte mir jemand vor einigen Monaten gesagt, dass ich jemals ein JAVA-Programm schreiben werde, hätte ich ihn wohl für verrückt gehalten. Im Bereich Programmieren, geschweige den von JAVA, hatte ich bis zu Beginn dieser Arbeit sehr wenig Erfahrung. Umso schwieriger gestaltete sich die Einarbeitung in den ersten Wochen der Diplomarbeit. Meine beiden erfolgreichen Semesterarbeiten im Fach Energietechnik haben mich dazu ermutigt, auch diese Herausforderung anzunehmen. Das Zusammenspiel von verschiedener Hard – und Software, aber auch der sportliche Aspekt hat mich an dieser Arbeit sehr fasziniert. Der Weg zum Ziel war steinig – umso glücklicher bin ich, es geschafft zu haben!



13 Abbildungsverzeichnis

<i>Abb 1 : DrillControlCenter for Spinningbike.....</i>	<i>2</i>
<i>Abb 2 : Zeitplanung</i>	<i>8</i>
<i>Abb 3 : Blockschaltbild Total.....</i>	<i>10</i>
<i>Abb 4 : Blockschaltbild Spinning Powerback.....</i>	<i>10</i>
<i>Abb 5 : Blockschaltbild Drill Control Center.....</i>	<i>11</i>
<i>Abb 6 : Blockschaltbild Netzteil und Senke.....</i>	<i>12</i>
<i>Abb 7 : Installation Befestigung DCC am Lenker.....</i>	<i>13</i>
<i>Abb 8 : Installation Anschlüsse DCC.....</i>	<i>13</i>
<i>Abb 9 : Installation Anschluss der Speisung.....</i>	<i>14</i>
<i>Abb 10 : Installation Einschalten und Abgleich der Stromzange</i>	<i>15</i>
<i>Abb 11 : Software für Betreuer.....</i>	<i>15</i>
<i>Abb 12 : Spinning Powerback</i>	<i>16</i>
<i>Abb 13 : Lampenmatrix</i>	<i>17</i>
<i>Abb 14 : Powerconnector.....</i>	<i>17</i>
<i>Abb 15 : Lampenarray</i>	<i>18</i>
<i>Abb 16 : PortExpander</i>	<i>19</i>
<i>Abb 17 : PortExpander Hardware.....</i>	<i>19</i>
<i>Abb 18 : Blockschaltbild Drill Control Center.....</i>	<i>20</i>
<i>Abb 19 : Drill Control Center offen.....</i>	<i>20</i>
<i>Abb 20 : Frontplatte Drill Control Center.....</i>	<i>21</i>
<i>Abb 21 : Blockschaltbild MSP430F149</i>	<i>22</i>
<i>Abb 22 : Blockschaltbild BasicClockSystem</i>	<i>23</i>
<i>Abb 23 : Pulsempfänger mit Verstärker und Tiefpass.....</i>	<i>27</i>
<i>Abb 24 : Komparator.....</i>	<i>27</i>
<i>Abb 25 : Astabiler Kippschwinger, Monoflop</i>	<i>28</i>
<i>Abb 26 : Signale im Pulsempfänger</i>	<i>28</i>
<i>Abb 27 : AD-Wandler PCF8591</i>	<i>29</i>
<i>Abb 28 : Schema Taster & Encoder</i>	<i>30</i>
<i>Abb 29 : Schema RS232 Treiber.....</i>	<i>31</i>
<i>Abb 30 : Kommandos für RS232 Kommunikation</i>	<i>32</i>
<i>Abb 31 : Schema Display & Pegelwandler.....</i>	<i>33</i>
<i>Abb 32 : Memory-Map Display-RAM.....</i>	<i>34</i>
<i>Abb 33 : Kommandos für RS232 Kommunikation</i>	<i>36</i>
<i>Abb 34 : Java GUI Willkommens Bildschirm.....</i>	<i>37</i>
<i>Abb 35 : Java GUI Benutzer registrieren.....</i>	<i>38</i>
<i>Abb 36 : Java GUI Bestätigung Registration.....</i>	<i>39</i>
<i>Abb 37 : Java GUI Ablauf "Free Training".....</i>	<i>41</i>
<i>Abb 38 : Java GUI Ablauf "Conconi Test"</i>	<i>43</i>
<i>Abb 39 : Auswertung der Trainings – und Testdaten</i>	<i>46</i>
<i>Abb 40 : Excel Auswertung Conconi Test.....</i>	<i>47</i>
<i>Abb 41 : Excel Auswertung Trainingseinheit</i>	<i>47</i>
<i>Abb 42 : StateMachine.....</i>	<i>48</i>
<i>Abb 43 : AQ430 File,New</i>	<i>55</i>
<i>Abb 44 : AQ430 New Project File.....</i>	<i>55</i>
<i>Abb 45 : AQ430 Projektsettings.....</i>	<i>56</i>
<i>Abb 46 : AQ430 Projektmanager.....</i>	<i>56</i>
<i>Abb 47 : AQ430 AddFiles</i>	<i>57</i>
<i>Abb 48 : AQ430 Compile and Transfer.....</i>	<i>57</i>