

IPEK-Prüfstand

Entwickeln und Produzieren eines physischen
User-Interface zum Steuern eines Prüfstandes



Abschlussbericht der Kooperationsphase 2018/2019

Durchgeführt am Institut für Produktentwicklung am Karlsruher
Institut für Technologie KIT
Betreut von Kai Wolter

Philipp Huth
Am Teilacker 10
76275 Ettlingen

Paul Margos
Bocksdornweg 58
76149 Karlsruhe

Inhaltsverzeichnis

1	Einleitung	4
1.1	Rahmenbedingung.....	4
1.2	Aufgabenstellung	5
2	Planung	6
2.1	Erste Ideen	6
2.2	Modell Kontrollboy	6
2.3	Hardware	7
2.4	Software	11
3	Umsetzung	12
3.1	Eingaben	12
3.2	Display	13
4	Ausblick	15
5	Danksagung	15
6	Abbildungsverzeichnis	16

Abstract

The aim of this project was to create a control unit for a transmission test bench. This control unit should smoothen the working process at the test bench. It also should enable the user to adapt the unit to his or her own needs.

During this project an analogue user interface was developed, which can be integrated into an existing computer system. The interface is based on the common micro controller line "Arduino" to benefit from the existing support for the platform and to facilitate the control unit for others to upgrade it or build up on this project.

1 Einleitung

Das IPEK ist eine Forschungseinrichtung mit der Aufgabe der Entwicklung von wissenschaftlichen Produkten und Innovation mit dem Schwerpunkt auf Antriebssysteme und Mobilität. Ziel unseres Projekts war es eine intuitive Nutzerschnittstelle (UI) zur Bedienung eines Prüfstandes am Institut für Produktentwicklung (IPEK) des KITs zu konzeptionieren und einen Prototypen herzustellen. Die UI soll den im IPEK arbeitenden Ingenieuren die Arbeit an den Prüfständen erleichtern, indem die Bedienung der Schnittstelle intuitiver ist als die Bedienung mit der Oberfläche am Computer.

1.1 Rahmenbedingung

1.1.1 Prüfstand

Mit einem Prüfstand werden Komponenten auf Funktionsweise, Effizienz und Beständigkeit getestet. Er dient dazu mögliche Schwächen und Fehler eines Bauteils schon vor der Massenfertigung zu erkennen. Der in dieser Arbeit verwendete Prüfstand ist ein Getriebeprüfstand zur Testung von Autogetrieben.

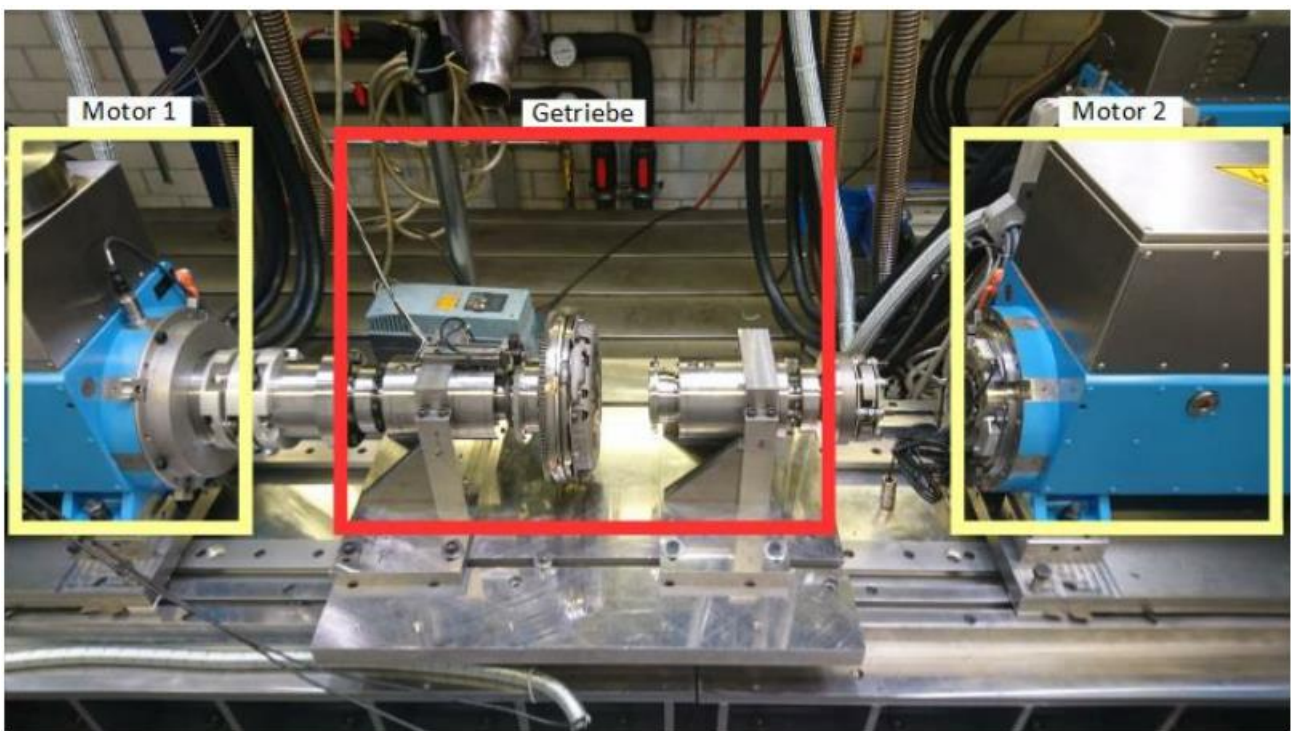


Abbildung 1: Getriebeprüfstand (eigene Abb.)

Der Getriebeprüfstand besteht aus:

- einem massiven Betonfundament, das gegen Schwingungen des Bodens gedämmt ist
Schwingungen aus dem Erdreich, ausgelöst durch zum Beispiel Bauarbeiten, würden die Messergebnisse der Schwingungsmessung verfälschen
- zwei Elektromotoren
ein Motor treibt das Getriebe an, der andere bremst es ab
- einer Einspannungsvorrichtung für die Testobjekte
sie kann für verschiedene Getriebeteile angepasst werden
- verschiedenen Sensoren
ein Schwingungssensor wird immer verwendet
andere Sensoren sind optional und werden je nach Bedarf am Prüfstand oder im Testobjekt eingesetzt

1.1.2 Funktionen des Prüfstands

Die Software des Prüfstands überprüft das Testobjekt auf Beständigkeit und Effizienz. Sie nutzt dabei eine Simulation des Straßenbetriebs eines Autos. Das zu testende Getriebe wird entweder mit einem festgelegten Drehmoment oder einer bestimmten Drehzahl angetrieben. Die zweite Größe ergibt sich jeweils aus der ersten und den Eigenschaften des Testobjekts sowie der Stärke der Bremsung durch den zweiten Motor.

Verschiedene vorgeschriebene Programme können abgespielt werden oder Werte in Echtzeit eingegeben werden.

1.2 Aufgabenstellung

Aufgabe dieses Projekts ist es die Arbeit an diesem Prüfstand mit einem User-Interface zu vereinfachen, intuitiver zu gestalten und zu beschleunigen. Das bedeutet das User-Interface soll lästige, sich wiederholende Arbeitsschritte aussparen, beispielsweise das Anklicken von Textfeldern zum Eingeben von Werten, ohne eine lange Einweisung und ohne Übung bedienbar sein und Zeit einsparen. Das User-Interface soll eine physische Steuerungseinheit darstellen, mit der die Grundfunktionen des Prüfstands angesteuert werden können. Es soll möglich sein den Prüfstand im Auge zu behalten und ihm gleichzeitig Befehle zu erteilen. Der Benutzer soll das Interface den eigenen Bedürfnissen anpassen können.

2 Planung

2.1 Erste Ideen

Ziel ist es eine eigene Steuerungseinheit zu bieten, die sich ohne Interaktion mit dem vorhandenen Desktop PC verwenden lässt, da sich dieser an einer ungünstigen Stelle vor dem Prüfstand befindet.

Um Funktionen abzurufen und Werte einzugeben ohne dabei das User-Interface ansehen zu müssen, erfolgt dies über analoge Knöpfe. Solche Knöpfe lassen sich im Gegensatz zu Knöpfen auf einem Touchscreen ertasten. Benutzt werden Druck- und Drehknöpfe, da sie anders als Schieber und Wippschalter immer in derselben Position sind und sich deshalb ohne Hinzusehen finden lassen.

Zur Ausgabe von Informationen wird ein Display verwendet, so dass der Benutzer nicht immer auf den Computerbildschirm schauen muss, um die Messwerte des Prüfstands zu sehen.

Über die Arbeit des UIs selbst geben zwei LEDs Auskunft. Mit ihnen lassen sich verschiedene Probleme diagnostizieren.

2.2 Modell Kontrollboy

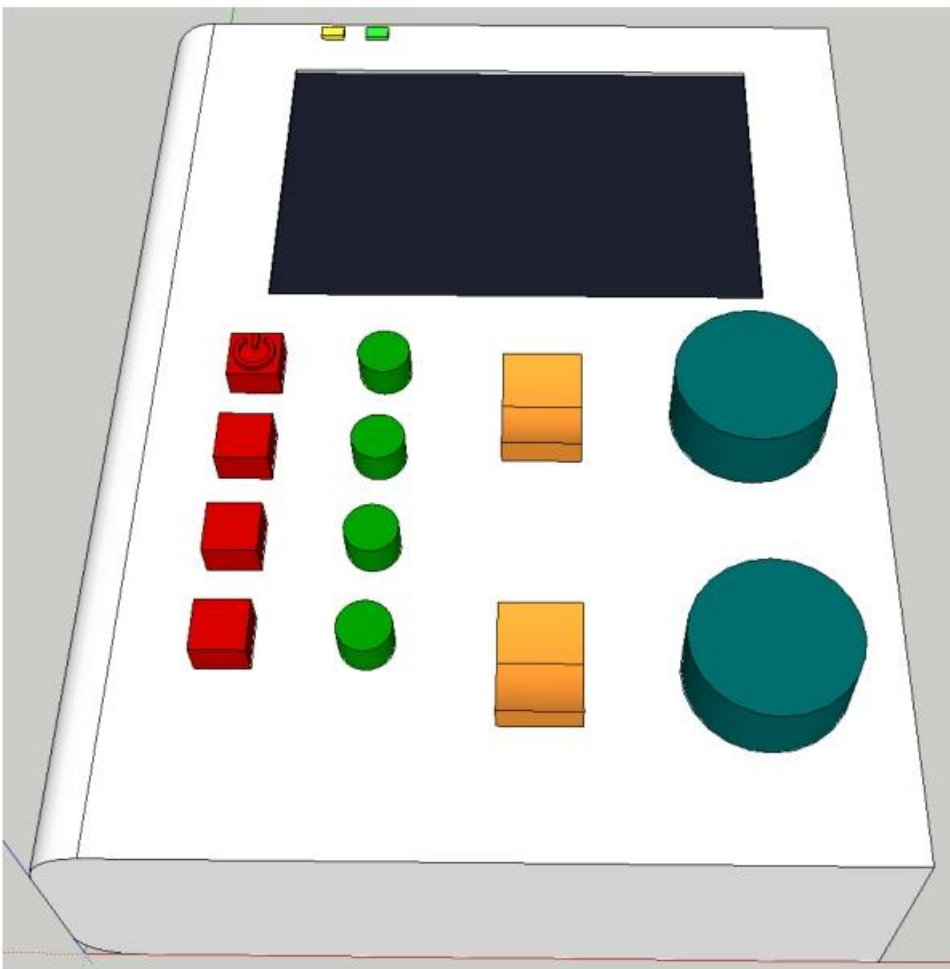


Abbildung 2: Kontrollboy (eigene Abb.)

Dieses 3D-Modell ist ein erster Entwurf für das User-Interface, in welches die obigen Überlegungen eingeflossen sind. Das Design basiert auf einem Game Boy, einer tragbaren Spielekonsolen der Firma Nintendo aus dem Jahr 1989. Das Vorbild zeichnet sich durch viel Platz für die Knöpfe und das oben liegende Display aus. Der Benutzer muss mit diesem Design den Blick weniger senken. Das User-Interface ist für Rechtshänder ausgelegt, kann aber auch von Linkshändern benutzt werden. Durch die Rundung an der linken Seite passt das User-Interface in eine menschliche Hand. Die Drehknöpfe können, dank ihrer Position am rechten Rand, bequem mit der rechten Hand erreicht werden. Die Wippschalter links der Drehknöpfe erlauben eine Mehrfachbelegung dieser Knöpfe. Mit Ändern der Schalterstellung ändert sich die Funktion. Links davon befinden sich zwei vertikale Reihen aus je vier Druckknöpfen. Sie sind am einfachsten mit der linken Hand zu bedienen. Zur Ausgabe von Informationen befinden sich in der linken oberen Ecke zwei LEDs.

2.3 Hardware

2.3.1 Drehimpulsgeber/ Potentiometer

Drehimpulsgeber und Potentiometer sind zwei Arten von Drehknöpfen. Potentiometer sind drehbare Widerstände, deren Widerstand abhängig von ihrer Position ist. Ein Potentiometer arbeitet mit Positionen zwischen 0° und höchstens 360° . Die Position lässt sich absolut bestimmen. Potentiometer sind stufenlos verstellbar, eine Änderung der Position kann laut theoretischen Berechnungen bis zu unter einem halben Grad kann mit den Messgeräten unseres Microcontrollers erfasst werden .

In einem Drehimpulsgeber dreht sich ein zahnradförmiger Kontakt über zwei punktförmige Kontakte.

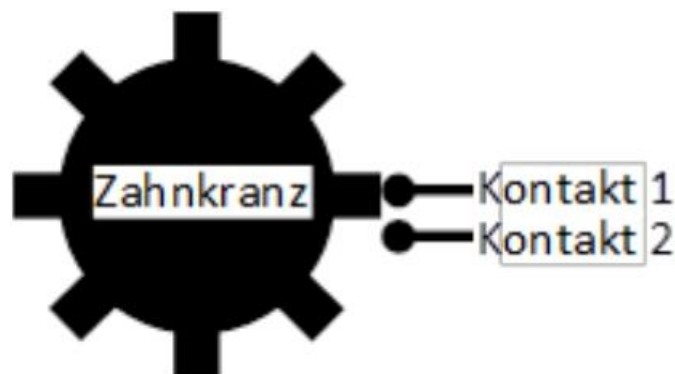


Abbildung 3: Drehimpulsgeber (eigene Abb.)

Bei einer Drehung des Drehimpulsgeber ist der zahnradförmige Kontakt abwechselnd mit den anderen verbunden. Dadurch kann die Weite und Drehrichtung bestimmt werden. Es kann keine absolute Position gemessen werden. Dafür kann der Drehimpulsgeber unendlich weit gedreht werden.

Es werden Drehimpulsgeber benutzt, da mit ihm Werte genauer eingestellt werden können. Sie können weiter gedreht werden, um mehr als 360°, als Potentiometer. Eine Viertelumdrehung muss deshalb nicht mindestens einem des Maximums entsprechen. So lässt sich ein Wert genauer einstellen. Das verbessert die User Experience, denn Menschen mögen keine unrunder Zahlen, wie beispielsweise 151,4 Nm anstelle von 150,0 Nm.

2.3.2 Drehimpulsgeber

Die zwei verwendeten Drehimpulsgeber haben je 32 Stufen auf einer Umdrehung. Sie rasten auf jeder Verstellstufe ein, was dem Benutzer eine Rückmeldung gibt. Sie können auch als Taster verwendet werden, wenn man sie nach unten drückt. Ihnen ist ein Vorwiderstand eingebaut, es wird keiner benötigt. Sie sind besonders robust und präzise.

2.3.3 Schalter /Taster

Es gibt zwei verschiedene Arten von Knöpfen, Schalter und Taster. Schalter behalten ihre Position nach dem Drücken bis zur erneuten Betätigung bei. Taster kehren zu ihrer Ausgangsposition zurück. Im User-Interface kommen acht Taster zum Einsatz. Taster sind in der benötigten Größenordnung weiter verbreitet als Schalter. Sie sind einfacher in ihrem Aufbau und günstiger. In Kombination mit Microcontrollern bieten Schalter keine Vorteile, ihre Funktionsweise kann softwareseitig mit Tastern nachgeahmt werden. Eine Gefahr beim Benutzen von Schaltern ist, dass sie beim Anschalten des Systems schon gedrückt sein können.

2.3.4 Wippschalter

Für die Wippschalter werden zwei Wippschalter genutzt, die über eine integrierte Lampe verfügen, welche eine Betätigung anzeigen kann. Wippschalter sind besondere Schalter, ihre Position kann ihnen angesehen werden. Die Lampe wird nicht verwendet. Die Wippschalter haben die Größe einer menschlichen Fingerkuppe. Sie sind angenehm zu bedienen.

2.3.5 Taster

Es werden zwei Arten Taster verwendet, jeweils vier Stück. Die verschiedenen Taster erfüllen verschiedene Funktionen, die runden dienen als Schalter, sie bleiben nach der ersten Betätigung bis zur nächsten gedrückt. Das wird durch Software ermöglicht.

Die runden Taster verfügen über ein Schraubgewinde, das den Einbau erleichtert. Sie sind schwer zu betätigen, weil sie tief gedrückt werden müssen. Sie verfügen über eine angenehm große Fläche zum Drücken.

Die quadratischen Taster sind kleiner und günstiger. Sie klicken bei einer Betätigung, was dem Benutzer eine Rückmeldung gibt. Sie erfordern mehr Kraft zum Drücken und haben eine kleine Fläche, was beim Benutzen tief in den Finger eindrückt. Aufsätze lösen bei dem User-Interface das Problem.

2.3.6 Display

Das Display, welches zum Einsatz kommt, ist ein 7 Zoll Touch Screen der Firma Nextion. Es verfügt über einen eigenen Microcontroller. Mit diesem ist es auch ohne den Arduino einsatzfähig. Nextion stellt bedienerfreundliche Software zur Verfügung, mit der das Layout auf dem Display designt und Funktionen, zum Beispiel ein Timer oder interaktive Textfelder, eingefügt werden können.

2.3.7 LED

Die verwendeten LEDs haben zwei verschiedene Farben, Orange und Gelb.

Sie benötigen eine Spannung von 5 V, welche der Arduino über die Digitalpins ausgeben kann.

2.3.8 Arduino

Der Arduino ist ein Microcontroller, welcher sich durch direkte Nutzungsmöglichkeit, eine auf ihn abgestimmte Programmiersprache und -umfeld und vielseitige Einsatzmöglichkeiten auszeichnet. Die Arduinoplattform ist weit verbreitet und wird durch verschiedene Drittanbieter unterstützt. Das von uns verwendete Modell Arduino DUE verfügt über erhöhte Rechenleistung von 84 Mhz und viele Aus- und Eingänge, darunter befinden sich bis zu 54 digitale und 12 analoge Pins, vier serielle Schnittstellen und zwei Micro-USB-Buchsen. Ersteres ist ausschlaggebend für die Nutzung. Dank der höheren Prozessorfrequenz können Werte in kleineren Intervallen eingelesen und verarbeitet werden. Für den Nutzer entsteht eine geringere Latenz beim Betreiben des Prüfstands.

2.3.9 Schaltung der Taster

An einigen Arduino Modellen ist nicht genug Platz für alle Knöpfe des UIs. Zu Beginn stand nicht fest welches Modell verwendet werden wird. Eine Schaltung kam zum Einsatz mit der theoretisch eine unbegrenzte Zahl an Tastern und Schaltern an einem Analogpin eingelesen werden kann.

Die Schaltung ist eine Spannungsteilerschaltung, eine solche wird auch Reihenschaltung genannt. Hierbei sind mehrere Widerstände hinter einander geschaltet. In Spannungsteilerschaltungen liegt an jedem Widerstand eine Teilspannung der Gesamtspannung an. Die Verhältnisse der Teilspannung sind gleich der Größenverhältnisse der Widerstände. In der verwendeten Schaltung wird die

Spannung, die an dem ersten Widerstand am Minuspol anliegt, gemessen. In Reihe zu diesem Widerstand sind die Knöpfe geschaltet. Zu jedem Knopf, Taster oder Schalter, ist ein Widerstand mit jeweils anderer Größe parallelgeschaltet. Wird ein Knopf gedrückt, wird der Widerstand überbrückt, die gemessene Spannung steigt. Abhängig von der Größe der Widerstände und Menge der gedrückten Knöpfe, steigt die Spannung bei jeder Kombination unterschiedlich stark. Der genaue Wert lässt sich errechnen. Dies wird im Folgenden als Beispiel durchgeführt.

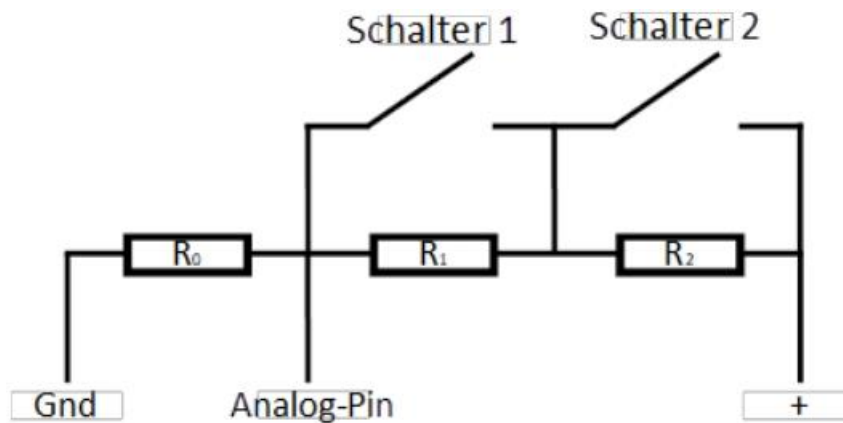


Abbildung 4: Schaltplan (eigene Abb.)

Beispielrechnung:

$$U_{ges} = 60 \text{ V}$$

$$R_0 = 1 \Omega$$

$$R_1 = 1 \Omega$$

$$R_2 = 2 \Omega$$

kein Schalter geschlossen:

$$U_{Pin} = U_{ges} \cdot R_0 : (R_0 + R_1 + R_2) = 15 \text{ V}$$

Schalter 1 geschlossen:

$$U_{Pin} = U_{ges} \cdot R_0 : (R_0 + R_2) = 20 \text{ V}$$

2.3.10 Schaltung

Die oben beschriebene Schaltung ist aufgrund der Übersichtlichkeit und des Einsparens der Isolierung auf eine Platine gelötet. Die Pins des Arduinos sind mit einer festgeklebten Steckleiste versehen. An deren hervorstehenden Pins sind die Kabel für die LEDs, die Drehimpulsgeber, das Display und die Platine verlötet. Diese Verbindung ist robuster als das Einstecken der Kabel in die Pins. Die Alternative, das Einkleben, verhindert eine Wiederverwendung des Boards.

2.3.11 Gehäuse

Für das Gehäuse waren fünf verschraubte Platten mit Aussparungen für Knöpfe und Display, sowie ein Rundstück aus der Aluminiumlegierung AlMg3 vorgesehen. Solche Platten waren nicht zeitnah erhältlich, deshalb wurde ein Prototyp aus Laser geschnittenem Holz gefertigt. Das Holz wurde lackiert, dass es nicht nach verbranntem Holz riecht. Die Holzplatten sind teilweise verklebt, die Rückseite ist verschraubt. Man kann sie zu Reparaturzwecken öffnen. Das Rundstück ist verklebt und aus PLA gedruckt.

2.3.12 Abdeckung der Knöpfe

Dass die Knöpfe einfacher und angenehmer zu betätigen sind, sind sie mit Aufsätzen versehen, die die Fläche zum Drücken oder Festhalten zum Drehen vergrößern.

Diese Aufsätze wurden aus Gründen der Einfachheit und Kosteneffizienz aus PLA gedruckt.

2.4 Software

2.4.1 Anforderungen

Mit der Software, die wir auf den Arduino und das Display aufgespielt haben, soll vor allem drei Aufgaben erfüllt werden. Die Knöpfe und das Touchpad auslesen, dem Prüfstand Sollwerte übertragen und Daten vom Prüfstand erhalten und anzeigen.

2.4.2 Trennung Display und Arduino

Zum Empfangen der Daten von dem Display auf den Arduino ist auf dem Arduino eine spezielle Bibliothek von Nöten. Diese Bibliothek ist mit dem verwendeten Arduino DUE nicht kompatibel. Deshalb ist es möglich Daten vom Arduino auf das Display zu übertragen, aber nicht umgekehrt. Das Display verfügt über einen eigenen Microcontroller, über den es sich selbst steuert. Ein umfassender Datenaustausch zwischen dem Arduino und dem Display ist dadurch nicht nötig.

2.4.3 Arduino IDE

Der Arduino wurde mit Hilfe des Arduino eigenen Programmumfelds programmiert und getestet. Es macht es einfach Bibliotheken einzubinden und das Programm zu übertragen.

2.4.4 Verbindung

Zur Kommunikation mit dem Prüfstand und dem Display wird eine serielle Verbindung genutzt. Die Verbindung zum Prüfstand geschieht über ein USB-Kabel. Bei der seriellen Kommunikation werden Daten über einen Leiter in einer vorbestimmten Rate von einem Gerät zum anderen gesendet. Die empfangbaren Befehle sind dabei im Programm des Empfängergeräts hinterlegt und werden abgeglichen. Auf einem anderen Leiter passiert das in die andere Richtung. Daten können gleichzeitig in beide Richtungen übermittelt werden. Dieses System ist fehlerunanfällig und einfach. Ein weiterer

Vorteil ist, dass über die genutzte USB-Schnittstelle Änderungen an der Software vorgenommen werden können. Auf diese Weise muss das Gehäuse nicht geöffnet werden.

2.4.5 Bibliotheken

Bibliotheken sind Teilprogramme die Funktionen beschreiben und zu bestehenden Programmen wie unseren hinzugefügt werden um die Funktionen nutzen zu können ohne sie neu schreiben zu müssen. Aufgrund der Probleme mit der Bibliothek zu dem Display greift unser Programm außer auf die üblichen Arduino Bibliotheken auf keine weiteren zurück. Wird zu einem späteren Zeitpunkt das Verbindungsproblem zum Display gelöst können auch Timerinterrupt Bibliotheken für den Arduino zum Einsatz kommen. So kann der Timer auf den Arduino verlegt und dort genutzt werden.

3 Umsetzung

3.1 Eingaben

3.1.1 Einlesen der Knöpfe

Nach dem in 2.3.9 beschriebenen Prinzip, zusehen in Abbildung 5, werden zwei Knöpfe je über einen Analogpin eingelesen. Analogpins messen die Spannung, welche an ihnen anliegt. Zur Verbesserung der Anpassungsmöglichkeiten des User-Interface an das spezielle Einsatzgebiet werden vier Taster virtuell zu Schaltern umgewandelt. Nach der ersten Betätigung werden sie im Programm als dauerhaft gedrückt betrachtet. Bei einer erneuten Betätigung wird dieser Zustand aufgehoben. Dadurch stehen dem Nutzer zwei sich verschieden verhaltene Knöpfe zur Verfügung.

3.1.2 Drehimpulsgeber

Die Drehimpulsgeber werden über die Digitalpins eingelesen. Besteht eine Verbindung zwischen einem punktförmigen Kontakt und dem drehbaren Kontakt, ist am zugehörigen Pin ein „High“. Wird ein Pin „High“ und der andere „Low“ lässt sich daraus auf eine Drehung des Drehimpulsgebers schließen, das Programm erkennt dieses Muster.

Die Position wird als Menge der zurückgelegten Drehschritte mit dem Uhrzeigersinn festgehalten, dreht man in die entgegengesetzte Richtung wird der Wert negativ.

3.1.3 Abfrage nach Befehlen vom Prüfstand

Der Arduino kennt zwei Befehle die er vom Prüfstand erhalten kann und auf die er reagiert. Die Datenstränge, die er vom Prüfstand erhält speichert der Arduino vorübergehend. Dann durchsucht der Arduino den Speicher nach einem Schlüsselwort, in diesem Fall „data“. Anhand der darauffolgenden Zeichen wird dem Arduino übermittelt, was zu tun ist. Entweder sendet er die Position der unbelegten Knöpfe und die Sollwerte für den Prüfstand oder er wartet darauf, dass Messdaten vom Prüfstand übertragen werden.

3.1.4 Einlesen von übermittelten Werten

Dass der Arduino die Messwerte zuordnen kann, werden sie durch eine charakteristische Zeichenfolge angekündigt. Danach wird das nächste Byte als Binärzahl eingelesen. Es werden die Istwerte des antreibenden und bremsenden Drehmoments, der Drehzahl und die gemessene Schnelle der vom Getriebe ausgehenden Schwingungen übertragen.

3.1.5 Doppelbelegung der Drehimpulsgeber

Neben jedem Drehimpulsgeber befindet sich ein Wippschalter. Je nach Schalterstellung hat der Drehimpulsgeber eine andere Funktion. Der Prüfstand treibt das zu testende Getriebe entweder mit einem bestimmten Drehmoment oder Drehzahl an. Mit Änderung der Schalterstellung des oberen Wippschalters, wechselt der obere Drehimpulsgeber zwischen diesen zwei Modi. Mit dem anderen Wippschalter kann über den zweiten Drehimpulsgeber entweder die Zeit für den Timer auf dem Display oder das Drehmoment für den bremsenden Motor festgelegt werden.

3.1.6 Standby Modus

Das User-Interface ist eine Ergänzung für das Kontrollsystem des Prüfstands und für dessen Betrieb optional, deshalb wird der Prüfstand auch ohne es in Gebrauch sein. Dass in diesem Fall das User-Interface keine Einstellungen am Prüfstand verändert, kann man es per Knopfdruck in einen Standby Modus versetzen. In diesem Modus wird das Display schwarz und die Kommunikation mit dem Prüfstand setzt aus.

3.1.7 Zurücksetzen der Drehimpulsgeber

Aufgrund der stufenlosen Verstellbarkeit des Drehimpulsgeber, ist es nicht erkennbar in welcher Position er sich befinden. Zur Rückkehr zum Ursprung wird die Position wieder auf null gesetzt, wenn der dafür vorgesehene Taster gedrückt wird. Zudem wird die Position zurückgesetzt, wenn die Funktionsweise des Drehimpulsgebers geändert wird. Letzteres ist ein Sicherheitsmechanismus, der versehentliche Einstellungen vermeiden soll.

3.2 Display

3.2.1 Ausgabe

Auf dem Display werden die Istwerte des antreibenden und bremsenden Drehmoments, die Drehzahl und die Schnelle der vom Getriebe ausgehenden Schwingung angezeigt. Diese Werte werden vom Prüfstand erfasst und an das User-Interface übertragen. Informationen über das User-Interface selbst werden dem Benutzer über zwei LEDs mitgeteilt. Mit der gelben LED wird angezeigt, dass sich das User-Interface im Arbeitsmodus befindet. Die orangefarbene LED zeigt an, dass Daten zwischen dem User-Interface und dem Prüfstand übertragen werden.

3.2.2 Design des Displays

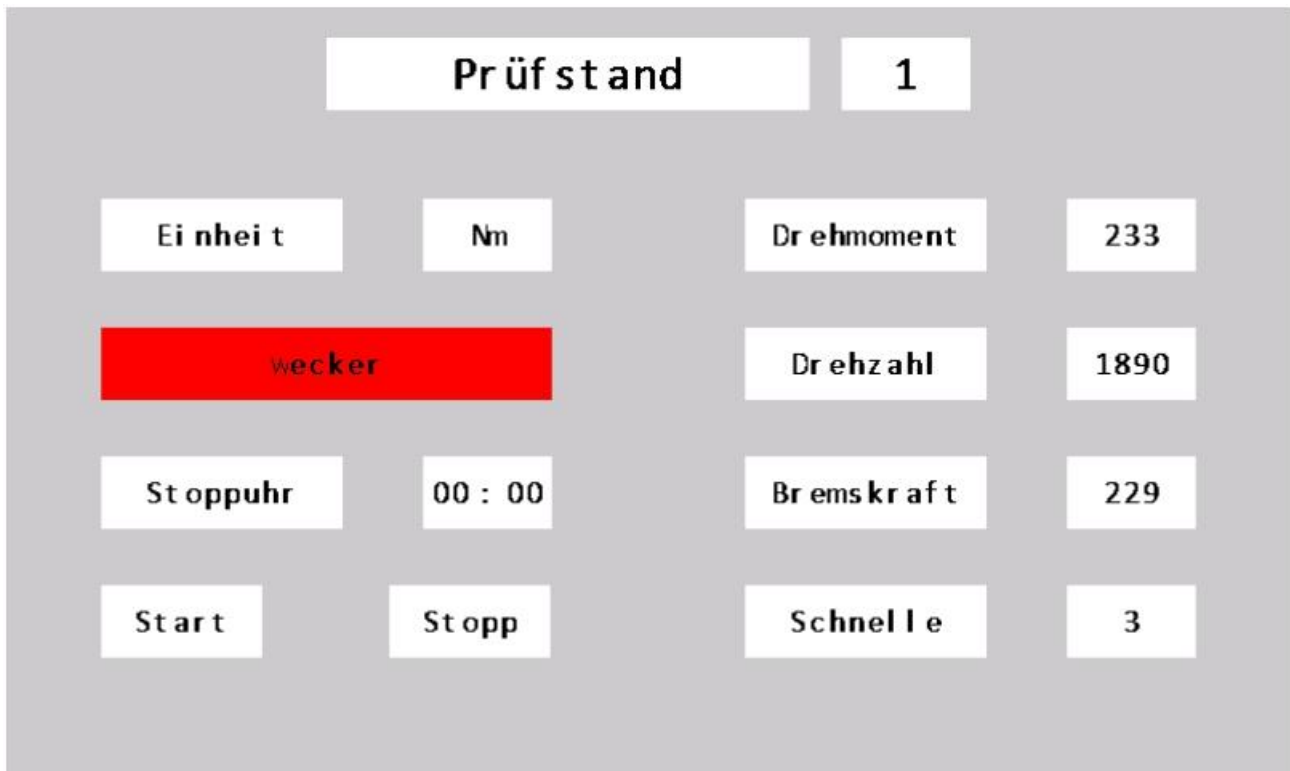


Abbildung 5: Displaylayout (eigene Abb.)

Abbildung 6 zeigt die Benutzeroberfläche auf dem Display.

Oben in der Mitte steht „Prüfstand“ mit der Nummer des Prüfstands, um allgemein zu zeigen worum es sich bei dem User-Interface handelt.

Darunter ist das Display zweigeteilt: Auf der rechten Seite befinden sich die Messwerte des Prüfstands, auf der linken Seite unten befindet sich der Timer. Darüber wird die Einheit eines ausgewählten Messwerts angezeigt. Die Einheiten stehen aus Platzgründen nicht bei den Messwerten.

Über der Zeit des Timers blinkt falls dieser abgelaufen ist ein rotes Textfeld mit dem Wort „Wecker“. Unter dem Timer befinden sich die dazugehörigen Knöpfe „Start“ und „Stopp“.

3.2.3 Funktionen

Diese beiden Knöpfe starten und stoppen den Timer. Die Zeit, von der runter gezählt wird, wird mit dem unteren Drehimpulsgeber eingegeben. Zur Zeitmessung wird der Timer des Displays genutzt. Da bei der Anzeige der Messwerte nicht genug Platz für die Einheiten ist und die Werte in einer leserlichen Größe dargestellt werden sollen, zeigt das Display die zur Größe gehörende Einheit auf Knopfdruck links an. Um dem Benutzer Klarheit über die Funktionen zu verschaffen wird der Wert, welcher mit dem oberen Drehimpulsgeber verstellt werden kann blau und für den unteren Drehimpulsgeber rot markiert.

4 Ausblick

Die Zielsetzungen des Projekts, ein User-Interface zur Verbesserung des Arbeitsprozesses am Geräteprüfstand, durch das Beschleunigen und Bieten von Individualisierungsmöglichkeiten, zu entwickeln, wurde erreicht.

Wichtige, erreichte Meilensteine waren ein den Arbeitsumstände am Prüfstand entsprechendes Design zu entwickeln, eine Auswahl an Funktionen zu treffen, Komponenten auszuwählen und das User-Interface zu fertigen und programmieren.

Das fertige User-Interface kann die rudimentären Funktionen eines Prüfstands ansteuern und kann von seinem Benutzer noch weiter optimiert werden.

Es gibt auch Verbesserungspotential an dem User-Interface, das bestand zukünftiger Projekte sein kann.

Die Verbindung zum Prüfstand ist noch nicht möglich, es fehlt eine geeignete Software auf empfangenden PC. Eine Liste und Erklärung der übertragbaren Befehle und Antworten liegt vor, wodurch es dem Anwender des User-Interface möglich ist die benötigte Software, auf das bestehende Betriebssystem zugeschnitten, zu schreiben. Im Anschluss an dieses Projekt soll die serielle Verbindung über USB, durch eine serielle Verbindung über ein Netzkabel ersetzt werden.

Das Gehäuse des User-Interface liegt noch als Prototyp vor, es ist nicht so robust und hochwertig wie die vorgesehene Aluminiumverkleidung. Das User-Interface kann aber auch im momentanen Zustand verwendet werden.

Unseren Anforderungen entsprechend sind sechs Taster des User-Interface nicht belegt, ihre Verwendung steht dem Nutzer frei zur Verfügung und kann im Programm des Prüfstandes codiert werden.

5 Danksagung

Wir möchten uns bei allen bedanken, die uns während des Projekts unterstützt haben.

Besonderer Dank geht an Kai Wolter, für die Ermöglichung des Projekts und das gesamte Ingenieurteam des IPEK für die produktive Zusammenarbeit.

Außerdem bedanken wir uns herzlichst bei Josephine Hector und Dr. h.c. Hans-Werner Hector für sechs Jahre Hector Seminar, in denen wir unsere Fähigkeiten im Bereich MINT und darüber hinaus ausprägen konnten.

Der letzte Dank geht an unsere Kursleiter Anke Richert und Dietmar Gruber, die uns in den vergangenen Jahren immer mit Rat und Tat zur Seite standen.

6 Abbildungsverzeichnis

Abbildung 1: Fertiges Modell (eigene Abb.)	1
Abbildung 2: Getriebeprüfstand (eigene Abb.)	4
Abbildung 3: Kontrollboy (eigene Abb.)	6
Abbildung 4: Drehimpulsgeber (eigene Abb.)	7
Abbildung 5: Schaltplan (eigene Abb.)	10
Abbildung 6: Displaylayout (eigene Abb.)	14

Selbstständigkeitserklärung

Hiermit versichern wir, dass ich die vorliegende Arbeit selbstständig und nur mit den angegebenen Hilfsmitteln angefertigt habe. Alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, wurden durch Angabe der Quellen als Entlehnungen kenntlich gemacht.

Paul Margos
27.09.2019
Karlsruhe

Philipp Huth
27.09.2019
Karlsruhe