



Fachbereich 3: Mathematik und Informatik

## Bachelorarbeit

zur Erlangung des akademischen Grades Bachelor of Science

# Entwicklung einer Anwendung zur Echtzeit-Visualisierung von CAN-Bus-Daten landwirtschaftlicher Maschinen

vorgelegt von

*Colin Weber*  
2840825  
*colin@uni-bremen.de*

Erstgutachter: *Prof. Dr. Udo Frese*  
Zweitgutachterin: *Dr. Hui Shi*

14. Februar 2020



# Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation für diese Arbeit . . . . .	1
1.2	Ziel der Arbeit . . . . .	2
1.3	Hypothesen der Arbeit . . . . .	2
2	Stand der Technik	5
2.1	CAN-Bus Systeme . . . . .	5
2.2	Displaysysteme von Traktoren . . . . .	7
2.2.1	Ältere Systeme . . . . .	7
2.2.2	Aktuelle Systeme . . . . .	8
2.3	AGCO Smart Connect . . . . .	10
2.4	Systeme von Drittanbietern . . . . .	12
2.5	Eigene Arbeiten: PolyCAN . . . . .	12
2.6	Zusammenfassung des Standes der Technik . . . . .	13
3	Methodik	15
3.1	Versuchsaufbau . . . . .	15
3.2	Versuchsdurchführung . . . . .	16
4	Praktischer Teil	19
4.1	CAN-Bus Adapter . . . . .	19
4.2	Planung der Anwendung . . . . .	21
4.3	Das Programm . . . . .	25
5	Ergebnisse	33
5.1	Integrationstest des Systems . . . . .	33
5.2	Online Benutzerumfrage . . . . .	37
6	Diskussion und Fazit	41
6.1	Diskussion der Ergebnisse . . . . .	41
6.2	Fazit . . . . .	44
	Literaturverzeichnis	53



# 1 Einleitung

## 1.1 Motivation für diese Arbeit

Die Idee dieser Arbeit ist es zu prüfen, ob eine günstige und unkomplizierte Möglichkeit, Traktoren mit einem digitalen Ausgabegerät nachzurüsten, für Landwirte interessant und effizienzsteigernd ist. Immer mehr Traktoren verfügen heutzutage ab Werk über Bildschirme, die den Fahrer mit wichtigen Informationen über die Maschine versorgen oder auch eine Interaktion mit der Maschine ermöglichen. Zu den typischen Informationen zählen, wie man es aus dem modernen PKW kennt, Motordrehzahl, Geschwindigkeit des Fahrzeugs, Temperaturen sämtlicher Betriebsflüssigkeiten, Spritverbrauch, aber auch Daten wie Motorlast, Reifenschlupf oder traktorspezifische Daten wie Zapfwellendrehzahl oder Hydraulikdruck. Die stetig voranschreitende Technik bringt neben der Komplexität der Bedienung solcher Maschinen auch eine immer größer werdende Effektivität und Genauigkeit der Arbeiten.

Ermöglicht hat dies unter anderem die Einführung des CAN-Bus Systems mit standardisiertem Netzwerkprotokoll, SAE J1939 um das Jahr 1994 [Heck u. a. 1999]. Der CAN-Bus ist das zentrale Nervensystem des Traktors, worüber alle Steuergeräte, elektronische Bauteile und digitale Anzeigen miteinander kommunizieren. Dieses Nervensystem hat frei zugängliche Schnittstellen, mit Lese- und Schreibrecht. Dies ist unter anderem für die Wartung, Fehlerbehebung oder Softwareupdates durch einen Monteur mit speziellem Gerät erforderlich. Solche Schnittstellen sind vergleichbar mit dem OBD2 Anschluss in einem PKW und lassen sich ebenfalls dazu nutzen, um SAE J1939 spezialisierte Geräte mit dem CAN-Bus System des Traktors zu verbinden. Solch ein Gerät, welches frei auf dem Markt verfügbar ist, kann mittels Endgerät wie Smartphone, Tablet oder Laptop per Wifi oder Bluetooth gekoppelt werden, um anschließend mittels zugehöriger Anwendung eine Auswertung der verfügbaren Informationen im CAN-Bus System vorzunehmen.

Dieses System soll in dieser Arbeit entwickelt werden, um Nutzern, auch ohne Kenntnis über die Funktionsweise des CAN-Bus Systems, es zu ermöglichen ein Android Smart-device mit dem Traktor, welcher über ein CAN-Bus System verfügt, zu verbinden und als Ausgabegerät zu nutzen. Angezeigt werden sollen alle im CAN-Bus verfügbaren Daten, die für den Bediener einen Nutzen haben, in einem grafisch strukturiertem und klarem Design.

Gerade kleine oder ältere Traktoren verfügen ab Werk meist nicht über eingebaute Bildschirme, allerdings oftmals über ein CAN-Bus System. In diesem System sind jedoch alle Daten, die zur Anzeige auf einem solchen Bildschirm verwendet werden, vorhanden und können ausgelesen werden. Dieser Vorteil soll in dieser Arbeit genutzt werden.

## 1.2 Ziel der Arbeit

Das Ziel dieser Arbeit ist es, kleine und alte Traktoren dem digitalen Standard heutiger Großtraktoren anzupassen. Damit soll eine Effizienzsteigerung und eine Arbeitserleichterung erzielt werden. Dies soll kostengünstig, ohne technische Veränderungen und einfach einsetzbar sein. Angestrebgt wird ein "*Plug and Play*" System, welches lediglich die Installation einer Applikation auf einem Android Smartdevice und das Verbinden eines CAN-Bus Dongles mit dem Traktor erfordert. Mittels dieses Systems sollen alle im CAN-Bus System verfügbaren Daten erfasst und in Echtzeit grafisch auf einem Smartdevice angezeigt werden. Zusätzlich zu der Ausgabe in Echtzeit sollen alle Daten, die während der Arbeit auf dem Feld gesammelt werden, abgespeichert und anschließend für eine Auswertung der erbrachten Arbeit verwendet werden können. Hierzu zählt zum Beispiel eine Auswertung der Bodenbeschaffenheit des Feldes, mittels Reifenschlupf, Motorlast oder Spritverbrauch, gepaart mit Ortspunkten des Feldes, welche in Koordinatenform abgespeichert werden. Diese gesammelten Daten sollen sich im Anschluss sowohl in Tabellenform als auch in Kartenform darstellen lassen. In der Kartenansicht sollen ortsabhängige Maschinenparameter angezeigt werden.

Dieses System soll Landwirten im Anschluss vorgestellt werden und mit einer Benutzerumfrage ermittelt werden, ob ein Nutzen und eine Nachrüstung vorstellbar wäre und ob mit der Nachrüstung eine Effizienzsteigerung und eine Steigerung der Bedienerfreundlichkeit denkbar ist.

## 1.3 Hypothesen der Arbeit

Folgende drei Hypothesen ergeben sich für diese Arbeit:

- Maschinenrelevante Parameter sind in älteren Landmaschinen über den CAN-Bus verfügbar und ermöglichen es, das oben beschriebene System kostengünstig zu entwickeln. (Verfügbarkeit und Machbarkeit)
- Die Belastung des Fahrers kann durch vereinfachte visuelle Kontrolle der Maschine gesenkt werden.

- Die Belastung der Umwelt kann durch effizienteres Arbeiten gesenkt werden, indem visuelles Feedback, über den Zustand von Feld und Maschine, über ein Tablet angezeigt wird.

Der Punkt Verfügbarkeit spielt in dieser Arbeit eine große Rolle und war die Motivation, solch ein zuvor genanntes System zu entwickeln. In einer vorherigen Arbeit [PolyCAN 2019], die sich mit der Problembehandlung und Manipulation des CAN-Buses in Traktoren beschäftigt hat, wurde deutlich, dass relevante Maschinenparameter für Effizienzsteigerung und Entlastung des Fahrers im CAN-Bus der Maschine verfügbar und mit einfachen Mitteln auslesbar sind. Aus dieser Erkenntnis entstand die Motivation für diese Arbeit.

Auch in der Landwirtschaft spielt der schonende Umgang mit der Umwelt eine immer größere Rolle. Das Minimieren von Abgasemissionen, bodenschonende Bearbeitung der Felder und das genaue Nutzen von Ressourcen, wird nicht zuletzt durch immer stärker werdende Kommunikation zwischen Fahrer und Maschine, mittels digitaler Technik, möglich. So fahren Traktoren mittels GPS gestützter Lenkung heutzutage Zentimeter genau von allein, die Aussaat wird nur dort in den Boden gebracht, wo es erforderlich ist und der Traktor passt seine Motordrehzahl, Gangwahl und Geschwindigkeit entsprechend den Wünschen des Bedieners genau an.

Das Thema der Entlastung des Fahrers findet sich in allen Bereichen der Fahrzeugindustrie wieder. Sowohl in der Nutzfahrzeugbranche, als auch in der PKW Branche. Nicht selten wird während der Erntezeit oftmals über 12 Stunden auf dem Traktor verbracht, um das Erntegut zum richtigen Zeitpunkt einzulagern. Hier können kleine Assistenzsysteme bereits zur Entlastung des Fahrers beitragen.



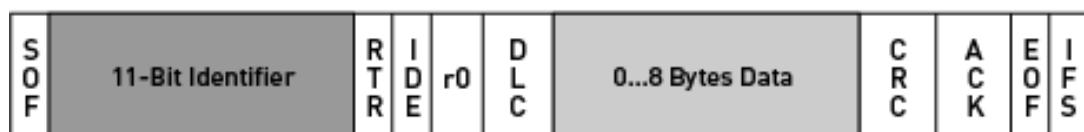
## 2 Stand der Technik

### 2.1 CAN-Bus Systeme

Das *Controller Area Network*, kurz *CAN*, welches 1983 von der Robert Bosch GmbH entwickelt und 1986 von der *Society of Automotive Engineers*, kurz *SAE*, als gebündeltes System zum Versenden von Informationen zwischen Steuereinheiten in Automobilen vorgestellt wurde [McFall und Fisher 2016], brachte einen neuen Standard in der Elektronik von Autos.

Dieses System wurde 1991 überarbeitet und als *CAN 2.0* publiziert [Robert Bosch GmbH 1991]. *CAN 2.0* ermöglichte sowohl ein Identifiersegment mit 11-Bit (*Base frame format*, CAN 2.0A) als auch mit 29-Bit (*Extended frame format*, CAN 2.0B) und dient noch heutzutage als Grundlage für die CAN Architektur [Umehara und Shishido 2019] (Abb. 2.1). 2012 erfolgte das letzte Update des CAN-Busses. Es wurde das *Controller Area Network - Flexible Data Rate*, kurz *CAN-FD* eingeführt und durch den ISO 11898 Standard definiert [ISO 11898-1:2015(E) 2015]. Dies ermöglicht eine flexible Anpassung an andere Protokolle mit einer Bitrate größer als 1 MBit/s und Übermittlung von Daten größer als 8 Byte pro Frame. [Hartwich und Bosch 2012 und SAE 2016].

**Standard CAN (V2.0A): 11-Bit Identifier**



**Extended CAN (V2.0B): 29-Bit Identifier**

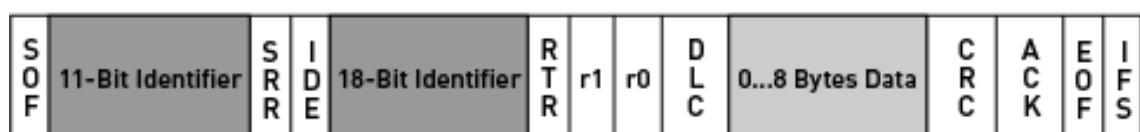


Abbildung 2.1: 11-Bit und 29-Bit Identifier CAN Frame [Eigene Grafik, angelehnt an: Lun u. a. 2010]



Abbildung 2.2: 29-Bit CAN Identifier nach SAE J1939 [Vector 2018]

Der CAN-Bus verbindet mittels zweier Kabeln alle Steuergeräte in einem Fahrzeug. Während an einem Kabel, welches als CAN-Low bezeichnet wird und die logische 1 repräsentiert, eine geringe Spannung anliegt, liegt an dem anderen Kabel, CAN-High, das die logische 0 repräsentiert, eine höhere Spannung an [Steve Corrigan 2002]. Eine über den CAN-Bus gesendete Nachricht wird als Frame bezeichnet und besteht aus mehreren Teilen. Jeder Frame beginnt mit dem Identifier welcher aus 11 oder 29 Bit besteht (Abb. 2.1). Der Standard SAE J1939 [SAE 2018] welcher für Heavy Duty Vehicles gilt, der auch Grundlage dieser Arbeit ist, da dieser ebenfalls Traktoren einschließt, definiert einen 29-Bit Identifier. Hiervon beschreiben 3 Bit die Priorität, 18 Bit die Parameter Gruppen Nummer, kurz PGN, und 8 Bit die Adresse der sendenden Steuereinheit (Abb. 2.2). Neben dem Identifier besteht ein CAN Frame ebenfalls aus einem Datensegment, welches bis zu acht Bytes an Nutzdaten übermitteln kann. Dieses enthält die eigentliche Nachricht, wie z.B. Motordrehzahl, Geschwindigkeit oder Spritverbrauch.

Jedes moderne Fahrzeug, ob Personenkraftwagen oder Nutzfahrzeug, verfügt heutzutage über einen sogenannten Diagnoseport. Dieser dient in erster Linie dazu, Mechanikern Zugriff auf die Steuergeräte des Fahrzeugs zu ermöglichen und mit spezieller Diagnosesoftware, die in der Regel herstellerspezifisch ist, den Fehlerspeicher des Fahrzeugs auszulesen, um Probleme festzustellen. Trotz des zuvor erwähnten Standards, welcher für die Kommunikation in dem Nachrichtensystem eines Fahrzeugs benutzt wird, gibt es Datenframes, die von diesem nicht definiert sind. Diese sind meist spezielle Fehlercodes. Diese werden von jedem Hersteller unterschiedlich definiert und werden *Proprietary PGN's* genannt (*herstellerspezifisch definierbare spezifische PGN*) [US Patent No. 10.256.969 B1 2019]. Dies führt dazu, dass trotz Standard für jeden Fahrzeugherrsteller ein extra Diagnosegerät benötigt wird, die oftmals bis zu 15.000€ kosten.

Eine weitere Nutzungsmöglichkeit des Diagnoseports ist das Auslesen von standardisierten Datenframes mittels eines weitaus günstigeren Adapters als mittels eines herstellerspezifisches Diagnosegeräts.

In folgender Abbildung (Abb. 2.3) ist ein solcher Diagnoseport sichtbar. Bei einem Deutsch 9-Pin Stecker liegt an Pin B eine dauerhafte Spannungsversorgung an, so wird auch bei ausgeschalteter Zündung des Fahrzeuges der eingesteckte Adapter mit Strom versorgt. An

Pin no.	Allocation
A	ECU_GND
B	Unswitched Power <sup>a</sup>
C	Tractor Bus CAN_H
D	Tractor Bus CAN_L
E	Not Specified <sup>b</sup>
F	Not Specified <sup>c</sup>
G	Not Specified <sup>c</sup>
H	Implement Bus CAN_H
J	Implement Bus CAN_L

<sup>a</sup>A direct connection to positive battery power through a 10A fuse.  
<sup>b</sup>Used for the shield of an SAE J1939 network in an SAE diagnostic connector.  
<sup>c</sup>Used for SAE J1708 network in an SAE diagnostic connector.

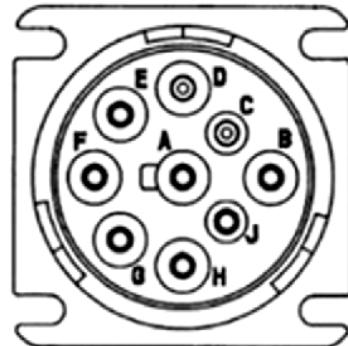


Abbildung 2.3: Deutsch HD10-9-1939 J1939 Diagnoseport Pinbelegung [Marx u. a. 2015]

Pin C und D liegen die Anschlüsse CAN-High und CAN-Low über die der J1939 Standard kommuniziert wird.

## 2.2 Displaysysteme von Traktoren

Seit der Einführung der CAN-Bus Systeme in Traktoren besteht die Möglichkeit, diese ab Werk mit eingebautem Display zu bestellen. Die Einstellungsmöglichkeiten und Anzeigeeoptionen solcher Bildschirme sind von Hersteller zu Hersteller ähnlich, lediglich die Darstellungsform variiert teils stark. In diesem Fall werden die Systeme des Herstellers *John Deere* vorgestellt, da während der Entwicklung dieser Arbeit, am BioResource & Agricultural Engineering Departement an der California Politechnic State University, Traktoren dieses Herstellers als Testfahrzeuge zur Verfügung standen.

### 2.2.1 Ältere Systeme

Im Jahr 2007 lieferte John Deere zwei neue Traktor Baureihen, den 6030 Premium und 7030 Premium, standardmäßig mit Display aus [John Deere 2006]. Das Farbdisplay, das *Command Center* genannt wird, ist mit dem CAN-Bus des Traktors verbunden und bietet die Möglichkeit Einstellungen am Traktor vorzunehmen und Maschinenparameter auszulesen. Dies soll durch einfache Bedienung den Bedienkomfort und die Produktivität erhöhen. Der Bildschirm verfügt über mehrere Screens, die man sich individuell anlegen und anzeigen lassen kann. Die Screens können mittels der Knöpfe oberhalb des Bildschirms durchgeschaltet werden, spezielle Eingaben werden mit dem Drehrad rechts vom Bildschirm und den 15 weiteren Knöpfen vorgenommen (Abb. 2.4 und Abb. 2.5).

Das *Command Center* bietet die Möglichkeit folgende Parameter anzuzeigen (sofern die Maschine damit ausgerüstet ist) [John Deere 2013]:

- Zapfwellendrehzahl
- Kühlflüssigkeitstemperatur
- Motoröldruck
- Betriebsstunden
- Motordrehzahl
- Maximale Hubhöhe der Heck-/Fronthydraulik
- Hydrauliköltemperatur
- Uhrzeit
- Batteriespannung
- Spritlevel
- Spritverbrauch pro Stunde
- Spritverbrauch pro Flächeneinheit
- Geschwindigkeit
- Reifenschlupf in %
- Bearbeitete Fläche
- Flächenleistung pro Stunde
- Getriebeeinstellungen
- Intelligent Power Management

### 2.2.2 Aktuelle Systeme

Seit 2013 vertreibt und rüstet John Deere Traktoren mit dem Gen 4 Command Center aus. Dieses gibt es in mehreren Displaygrößen, von 21,3cm bis 25,4cm, fest verbaut oder mobil Abb. 2.6. Die Bildschirme der 4. Generation verfügen zusätzlich zu den schon bekannten Knöpfen und Drehrad ebenfalls über eine Touchfunktion. Alle Geräte funktionieren herstellerübergreifend und ermöglichen einen schnellen Wechsel zwischen verschiedenen Maschinen unterschiedlicher Hersteller [John Deere 2020b]. Wie bei dem oben vorgestellten Command Center, können auch hier Screens individuell eingerichtet werden und zwischen diesen mit einer von Tablets bekannten Wisch-Geste hin und her gewechselt werden. Ebenfalls hinzu gekommen ist eine Kartenansicht. Sofern der Traktor über GPS verfügt können Fahrtwege geplant, angelegt und angezeigt werden. Durch die große Menge an Einstellungsmöglichkeiten verfügen die Bildschirme über eine Hilfefunktion,



Abbildung 2.4: Bildschirm eines Command Center in einem John Deere 6030 Premium [John Deere 2016]



Abbildung 2.5: Gesamtübersicht eines Command Center in einem John Deere 6030 Premium [John Deere 2013]

die kontextbasiert dem Nutzer kurz vermittelt, welche Aktionen vorgenommen werden können.

Eine große Neuerung gegenüber der vorherigen Generation ist der Fernzugriff. Dieser ermöglicht eine gemeinsame Bildschirmnutzung von mehreren Personen und soll zum Beispiel dabei helfen, unerfahrene Bediener durch erfahrenes Personal, welches sich an einem anderen Standort befindet, zu unterstützen, oder eine einfache Überwachung für die Zeitplanungen in Lohnunternehmen sicherzustellen.

Des Weiteren bietet diese Generation die Möglichkeit der Dokumentation und der draht-



Abbildung 2.6: Command Center 4. Generation [John Deere 2020a]

losen Datenübertragung, um zum Beispiel Dokumentationsdaten auszutauschen. Die mobilen Displays bieten zudem die Option die Bildschirmfläche zu verschieben und mehr Parameter ohne Umschalten der Screens sichtbar zu machen.

## 2.3 AGCO Smart Connect

Ein weiteres System, was in dieser Arbeit kurz vorgestellt werden soll, ist das Smart Connect von AGCO (auch als Fendt Smart Connect bekannt). AGCO (Allis-Gleaner Corporation) ist ein weltweiter Landmaschinenhersteller mit Hauptsitz in Georgia, USA und hat unter anderem folgende Tochterfirmen:

- Fendt
- Challenger
- Massey Ferguson
- Valtra

Dieses System wurde zusammen mit dem Erscheinen des neuen Traktor Fendt 942 Vario Anfang Juli 2019 vorgestellt.

Besonders interessant macht dieses System zum einen der Zeitpunkt des Erscheinens, denn dieser war parallel zu der Entwicklung der Applikation für diese Arbeit, zum Anderen

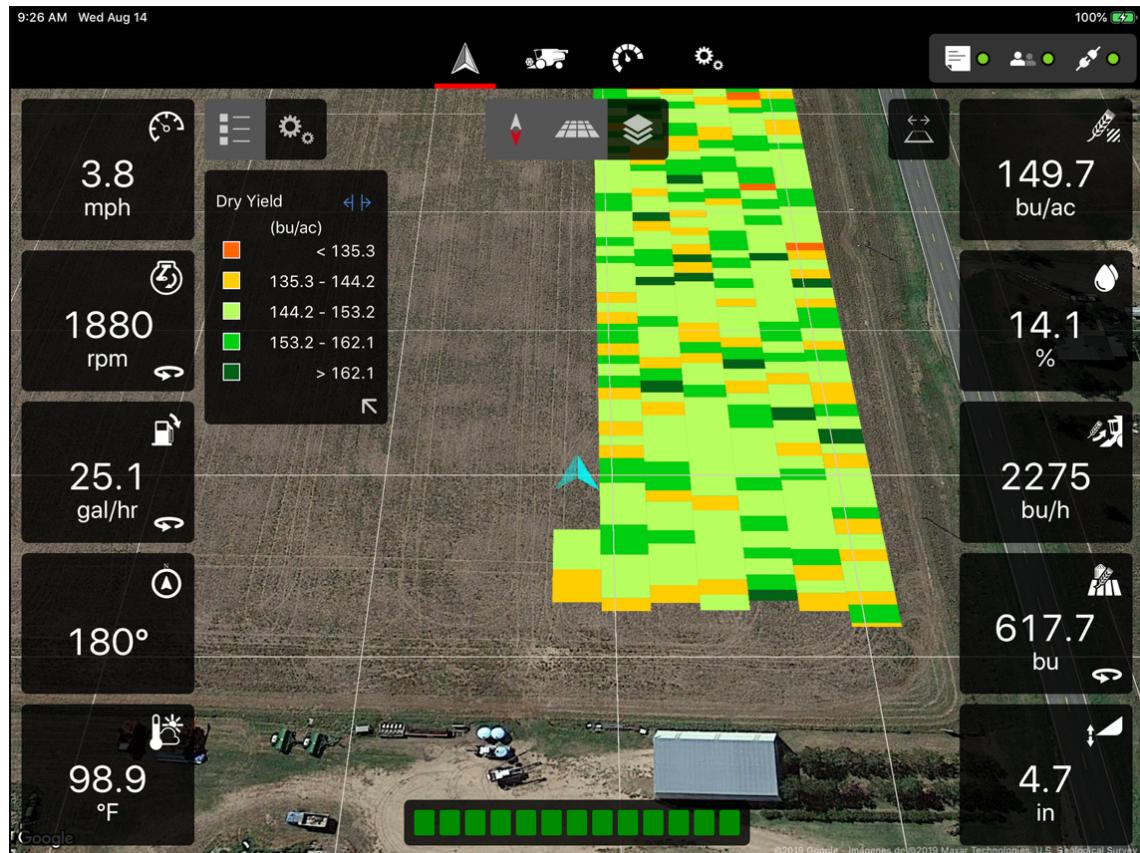


Abbildung 2.7: AGCO Smart Connect [Fuse 2019]

die starke optische und auch technische Ähnlichkeit beider Apps (Abb. 2.7). Die App ist speziell für iOS Endgeräte entwickelt und kommuniziert per WLAN Schnittstelle von einem im Traktor fest verbauten *Accessory Control Module*, kurz ACM, mit diesem. Es ist somit kein zusätzlicher Adapter, der mit dem CAN-Bus verbunden wird, erforderlich. Dies ermöglicht die Echtzeit-Visualisierung von Maschinenleistungsparametern auf dem Endgerät. Beworben wird dieses System als: *Die zusätzliche Anzeigefläche bietet eine kostengünstige Variante für extra Visualisierungsfläche und kann individuell je nach Kundenbedürfnis angepasst werden*. Diese Nachrüstmöglichkeit ermöglicht es somit Anzeigen von dem fest verbauten Bildschirm des Traktor zu übernehmen, diesen damit visuell zu entlasten und für andere Anzeigen nutzbar zu machen.

Neben den Maschinenleistungsparametern wird ebenfalls eine Echtzeitabdeckung auf einem Satellitenbild angezeigt, diese wird durch Markierungen gewünschter Parameter, wie z.B. Motordrehzahl, Schlupf, Kraftstoff/ AdBlue-Verbrauch, Motorlast und Hubkraft ergänzt. [Fuse 2019, Fendt 2019]

## 2.4 Systeme von Drittanbietern

### FieldView

Ein weiteres System ist *Climate FieldView*, dieses wird von The Climate Corporation in enger Zusammenarbeit mit AGCO entwickelt. Der von FieldView entwickelte *in-cab logger* wird in den CAN-Bus Diagnoseport der landwirtschaftlichen Maschine gesteckt und mit einem iPad per Bluetooth verbunden. Dies ermöglicht das Sammeln und Visualisieren von Maschinen- und Felddaten in Echtzeit während der Fahrt, wobei sich dieses System im Gegensatz zu Smart Connect weniger auf Maschinenparameter und stärker auf Kartierungen der Feldparameter konzentriert. Dieses System unterstützt dabei nur Anbaugeräte für die Aussaat und Düngung der Hersteller John Deere und Case. Zusätzlich können ausgewählte Parameter per Internet abgegriffen werden, um eine Koordinierung der Maschinen in einem Betrieb zu vereinfachen. Ebenfalls können manuell Daten in eine Cloud geladen werden. Der Anschaffungspreis eines solchen *in-cab loggers*, welcher herstellerübergreifend funktioniert, liegt in Nordamerika bei 299\$. Die dazu notwendige App wird kostenlos im App-Store als Download angeboten. Verschiedene Abo Modelle, die mehr Features bieten, werden ebenfalls angeboten. [The Climate Corporation 2018]

## 2.5 Eigene Arbeiten: PolyCAN

PolyCAN ist ein über ein Jahr entwickeltes System von Studenten der California Polytechnic State University, an dem der Autor ebenfalls mitgewirkt hat. Während der Entwicklung von PolyCAN konnte der Autor sehr viel über Traktoren und dessen CAN-Bus Technik lernen. Dieses Wissen diente als Grundlage und führte zu dieser Arbeit.

In Zusammenarbeit mit der Firma *iFixIt* wurde das System PolyCAN mit dem Ziel entwickelt, es Landwirten eigenständig zu ermöglichen, Softwareprobleme am Traktor zu beheben. Folgendes Problem sollte behandelt werden:

Ein ortsansässiger Farmer mit einem Kettentraktor hatte Probleme mit einem Sensor, der die Kettenspannung überwacht. Dieser Sensor hat einen Fehler ausgegeben und die Software des Traktors hat die Weiterfahrt der Maschine verhindert um weitere Schäden zu vermeiden. Den Sensor konnte der Farmer eigenständig für unter 100\$ austauschen, die Fehlermeldung war jedoch weiterhin in der Software gespeichert und hat den Traktor weiter lahmgelegt. Um dies zu reparieren musste der Farmer einen Mechaniker aus der nächsten Vertragswerkstatt mit einem Diagnosegerät, welches ca. 7000-8000\$ kostet, kommen lassen, um den Fehler zu löschen. Dies hat den Farmer mehrere 100\$ gekostet.

PolyCAN ist sowohl ein Lese- als auch ein Schreibgerät und ermöglicht das Überschreiben von solchen Fehlern in der Software, sodass ein softwarebedingter Stillstand der Maschine verhindert wird. Es ist zwar nicht möglich einen Fehlercode zu löschen, allerdings diesen

Fehlercode durch einen anderen Code, der in sehr hoher Frequenz an die Steuergeräte mittels PolyCAN geschickt wird, zu überspielen. Einen Manipulationsvorgang des Tachometers eines Traktors mittels PolyCAN zeigt das Video auf der Website von PolyCAN. Bei ausgeschaltetem Motor konnte die Anzeige der Drehzahl über den Tachometer des Traktors verändert werden. [PolyCAN 2019]

## 2.6 Zusammenfassung des Standes der Technik

Zusammenfassend kann gesagt werden, dass es bereits Systeme gibt, welche die Visualisierung von Maschinenparametern in Echtzeit ermöglichen. Diese Systeme sind allerdings nicht kostengünstig nachrüstbar oder nicht herstellerunabhängig.

Das Gen4 von John Deere bietet zwar keine kostengünstige Nachrüstmöglichkeit, kommt dem in dieser Arbeit angestrebtem System dennoch am nächsten. Es lässt sich mittels ISO-Bus, einer weiteren standardisierten Schnittstelle, die ebenfalls den J1939 Standard nutzt, mit allen Traktoren unterschiedlicher Hersteller, die so eine Schnittstelle besitzen, verbinden. Es ermöglicht die Anzeige vieler Maschinenparameter und weitere Optionen, wie z.B. das Anzeigen einer Karte.

Das Smart Connect von AGCO lässt sich nur mittels eines iOS Gerätes und einer Maschine, die über ein integriertes ACM verfügt, nutzen und zielt damit auf Verwendung bei modernen Maschinen ab.

Climate FieldView verwendet einen CAN-Bus Datenlogger, so wie das System in dieser Arbeit, konzentriert sich allerdings auf die Auswertung und Visualisierung von Maschinendaten der Anbaugeräte und ist nur mit zwei Herstellern kompatibel. Ebenfalls ist ein iOS Gerät notwendig.

Es besteht damit ein Potential für ein kostengünstiges, herstellerunabhängiges System zur Echtzeit-Visualisierung von den im CAN-Bus verfügbaren Maschinenparametern, das im Folgenden entwickelt werden soll.



# 3 Methodik

Wie in der Einleitung bereits erläutert, soll diese Arbeit drei Hypothesen abdecken.

- Die Verfügbar- und Machbarkeit ein solches System kostengünstig zu entwickeln und dieses nicht maschinengebunden einsetzbar zu machen.
- Die Belastung der Umwelt durch visuelles Feedback von Feld und Maschine über ein Tablet, was effizienteres Arbeiten ermöglicht, zu senken.
- Die Belastung des Fahrers durch vereinfachte visuelle Kontrolle der Maschine zu senken.

## 3.1 Versuchsaufbau

Für diese Arbeit soll ein System, bestehend aus Software, Android Tablet und CAN-Bus Adapter entwickelt werden.

Begleitet wurde die Entwicklung des Systems durch einen Experten, der über langjährige Erfahrung als Produktentwickler in der landwirtschaftlichen Industrie, mit den Schwerpunkten der Entwicklung großer landwirtschaftlicher Maschinen, Produkt- und Werkzeug-Innovationen, Prozessentwicklung und Automation verfügte.

### Verfügbarkeit und Machbarkeit

Für die Versuchsdurchführung der Verfügbarkeit und Machbarkeit soll ein System entwickelt werden, welches folgende Kriterien erfüllt und zusätzliche Voraussetzungen gegeben sein müssen:

- Es ist zu prüfen, ob die Traktoren, die zur Verfügung stehen und über ein CAN-Bus verfügen, einen für den Fahrer zugänglichen Diagnoseport haben, mit dem man Zugriff auf den J1939 Standard des Traktor erhält.
- Die auf dem Markt erhältlichen Adapter, die für dieses System notwendig sind, und an einen CAN-Bus Diagnoseport angeschlossen werden, dürfen nicht hersteller-, baujahr- oder traktorspezifisch sein.

- Es soll ein Adapter, welcher sich an den Diagnoseport des Traktors anschließen lässt und die Ausgabe des CAN-Bus interpretieren kann, mit einem Android Tablet verbunden werden.
- Das Interface der Anwendung auf dem Tablet soll für die interne Prüfung der Machbarkeit genutzt werden.
- Alle, nach Absprache mit dem Experten, für den Fahrer als wichtig gewertete und verfügbaren Parameter, die über den CAN-Bus des Diagnoseports auslesbar sind, sollen gesammelt und angezeigt werden. Hierzu ist kein grafisches Interface notwendig, eine Textausgabe ist ausreichend.
- Alle erforderlichen Parameter der Maschine müssen über den CAN-Bus ausgelesen werden können.
- Die verfügbaren, auslesbaren Daten der Maschine sind nicht hersteller-, baujahr- oder traktorspezifisch.
- Des Weiteren muss geprüft werden ob der Einsatz von dieser Technik ohne Einschränkungen nutzbar ist (denkbar wäre hier zum Beispiel ein Blockieren des Fahrbetriebs bei angeschlossenen Geräten am Diagnoseport).

#### Entlastung von Umwelt und Fahrer

Um die Entlastung von Umwelt und Fahrer überprüfen zu können, muss die Meinung von Dritten gesammelt werden. Hierfür ist ein funktionsfähiger Prototyp notwendig.

Der Prototyp soll während der Nutzerumfrage dazu dienen, außenstehenden Personen einen Eindruck von der geplanten Anwendung zu vermitteln, um eine Überprüfung der Hypothesen zu ermöglichen.

Für die Online Umfrage ist die Funktionstüchtigkeit des Systems nicht erforderlich. Hier soll das theoretisch erarbeitete Konzept validiert werden. Die Umfrage soll möglichst kurz und leicht verständlich gehalten werden, aber einen genauen Einblick in das System vermitteln und den Teilnehmern die Möglichkeit für individuelle Antworten bieten. Die Umfrage soll für Personen, die in einer Verbindung zur Landwirtschaft stehen zugänglich und durchführbar sein.

## 3.2 Versuchsdurchführung

### Verfügbarkeit und Machbarkeit

Um die Verfügbarkeit von CAN-Bus-Daten und die Machbarkeit eines solchen Systems zu prüfen, sollen mit dem zusammengestellten System und dem entwickelten Interface an mehreren Traktoren Tests durchgeführt werden. Alle Daten müssen gesammelt werden. Um zu testen, ob und welche Parameter aus einer Maschine ausgelesen werden können,

Hersteller und Modell	Baujahr	Besonderheiten
John Deere 5055E	2014	keine Kabine
John Deere 5525N	2009	nachgerüstete Elektroniksysteme
John Deere 5525 High Crop	2006	keine Kabine
John Deere 7330 Premium	2008	Bordcomputer
John Deere 8520T	2003	Raupenfahrwerk

Tabelle 3.1: Auflistung aller zur Verfügung stehenden Traktoren

muss diese gestartet und gefahren werden, um einen Realbetrieb zu simulieren. Zusätzlich müssen die Zapfwellen angeschaltet werden, damit überprüft werden kann, ob eine Zapfwellendrehzahl auslesbar ist. Der Hersteller, das Baujahr und die Baureihe des Traktors muss zusätzlich notiert werden. Falls der Traktor bereits über einen Bordcomputer verfügt, müssen hier die auslesbaren Daten des Bordcomputers mit den auslesbaren Daten des selbst entwickelten Systems verglichen werden und alle Differenzen notiert werden.

#### Entlastung von Umwelt und Fahrer

Die Onlineumfrage wird mittels Google Forms erstellt und in verschiedenen landwirtschaftlichen Foren geteilt. Den Teilnehmern soll ein kurzer Einstieg in das Thema, inklusive Bild des Systems in Benutzung, vermittelt werden, um es den Teilnehmern zu ermöglichen, die Fragen mit einem guten Verständnis des Systems zu beantworten. Die Nutzerumfrage ist in sieben Kapitel unterteilt.

1. Datenschutzerklärung
2. Einführung in das Thema: Anwenderumfrage zur Echtzeit-Visualisierung von Maschinenparametern mittels App. Es wird das Problem geschildert, dass viele kleine und ältere Traktoren über keine Bildschirme oder Bordcomputer verfügen um Fahrer mit aktuellen Maschinenparametern zu versorgen. Anschließend wird kurz die Funktionsweise des hier entwickelten Systems erklärt und das durch den CAN-Bus ein einfaches Nachrüsten möglich ist.
3. Personenbezogene Fragen und der Bezug zur Landwirtschaft sollen erörtern, in welcher Beziehung die Person zur Landwirtschaft steht, in welcher Altersgruppe der Teilnehmer ist und ob dieser Traktor fährt.
4. (Wenn in dem vorherigen Kapitel bestätigt wurde, dass die Person Traktor fährt) Grundlegende Fragen zum Bedienen von Traktoren: In welcher Häufigkeit die Person einen Traktor bedient, ob sie eventuell mit Displays oder Bordcomputern vertraut ist und ob der Traktor, den die Person bedient, über einen CAN-Bus verfügt und somit eine Nachrüstmöglichkeit bietet (Dies wird über das Alter des Traktors abgefragt).

5. Vorstellung der App, speziell des Arbeitsscreens: Alle angezeigten Maschinenparameter und Funktionen werden kurz erklärt und die grafische Benutzeroberfläche durch einen Screenshot dargestellt.
6. Abfrage der Nützlichkeit des zuvor gezeigten Arbeitsscreens. Bei einem negativen Ergebnis (die Anwendung ist störend oder nicht nützlich), wird zum Ende der Umfrage gesprungen.
7. (Wenn in dem vorherigen Kapitel die Frage der Nützlichkeit mit einer positiven Antwort bewertet wurde)  
Abfrage welche auf dem Arbeitsscreen sichtbaren Maschinenparameter des Systems nützlich sind, zu welchem Preis eine Aufrüstmöglichkeit interessant wäre und offene Frage nach weiteren Funktionswünschen.
8. Weitere Anmerkungen und Vorschläge

# 4 Praktischer Teil

## 4.1 CAN-Bus Adapter

### Entscheidung für den Adapter

Als Hardware für diese Arbeit werden zwei Dinge benötigt, zum Einen ein visuelles Ausgabegerät in Form eines Tablets und zum anderen ein Adapter, welcher das Tablet mit Informationen aus dem CAN-Bus des Traktors speist. Bei der Wahl des Tablets entschied sich der Autor bewusst gegen den Marktführer Apple mit 60% Marktanteil entschieden, da eine Entwicklung von iOS Apps auf einem Windows PC nicht unterstützt wird und während dieser Arbeit nur ein solcher PC zur Verfügung stand. Es wurde sich deshalb für ein Android Tablet, welches den zweitgrößten Marktanteil von 39% ausmacht entschieden (Stand: Februar 2020). [statCounter 2019]

Adapter, welche den CAN-Bus eines Traktors über den Diagnoseport auslesen und interpretieren können, gibt es auf dem Markt ab ca. 100€. Bei vielen dieser Adapter war die Beschreibung mangelhaft und es war nicht ersichtlich, welche Features diese Geräte mit sich bringen. Ebenfalls konnte bei vielen Geräten nicht festgestellt werden, ob der Hersteller eine Bibliothek zum Entwickeln einer Android Applikation zur Verfügung stellt. Für die Adapter, NEXIQ Blue-Link Mini und Digi Wireless Vehicle Bus Adapter (WVA), konnten die genauen Daten ermittelt und die zur Verfügung stehende Android Bibliothek begutachtet werden. Beide Geräte werden direkt in den Diagnoseport des Traktors gesteckt und lassen sich mittels Bluetooth bzw. Wifi mit einem Smartdevice verbinden. Folgende Tabelle (Tabelle 4.1) vergleicht beide Geräte im Detail. [NEXIQ-Technologies 2020 & Digi-International 2020]

Beide Adapter sind für das in dieser Arbeit angestrebte System einsetzbar und haben jeweils ihre Vorteile. Aufgrund der deutlich übersichtlicheren Webpräsenz, mehr verfügbarem Speicher, mehr Features, den umfangreichen Bedienungsanleitungen und Guidelines des Adapters, einer zur Verfügung stehenden Demo Applikation und einer 40 Seiten langen Android-Bibliotheken-Anleitung wurde sich für den Adapter des Herstellers Digi International Incorporated entschieden, um den Zeitaufwand für das Einrichten und Aneignen der Software für den Adapter möglichst gering zu halten.

Adapter	NEXIQ Blue-Link Mini	Digi WVA
Steckverbinder	9-Pin Deutsch Stecker	
Kompatible Betriebssysteme	Android und iOS Software verfügbar	
Interpretierbare Standards	J1939 und J1708 Standard	
Verbindung	Bluetooth	Wifi
Arbeitsspeicher	576KB RAM	64MB RAM
Speicher	2,5MB Flash	128MB Flash, davon 20MB frei nutzbar für z.B. Python Skripte
Weiteres		Web-Interface
		USB Port
		Schlummer Funktion
		Buzzer, per Software definierbar
		3 Achsen Beschleunigungssensor
Verfügbare Bibliotheken	Android und iOS Bibliothek	Android Bibliothek
Dokumentation	Geringe Dokumentation	Sehr ausführliche Schritt für Schritt Dokumentation

Tabelle 4.1: Vergleich des NEXIQ Blue-Link Mini mit dem Digi WVA Adapter



Abbildung 4.1: Digi Wireless Vehicle Bus Adapter (WVA) [eigenes Bild]

### Anschließen und Verbinden mit dem Adapter

Um den Wireless Vehicle Bus Adapter (WVA) in Betrieb zu nehmen, muss dieser mit Strom versorgt werden. Dies kann am einfachsten über den Diagnoseport im Traktor erfolgen, da dieser auch bei ausgeschalteter Zündung immer mit 12 bzw. 24 Volt versorgt wird. Die genaue Position des Diagnoseports im Traktor ist von Hersteller und Baureihe abhängig. In den meisten Fällen befindet sich der Anschluss in Sitznähe. Mögliche Positionen hierfür sind direkt hinter dem Sitz, unter einer abschraubbaren Seitenverkleidung, in oder in der Nähe von einem Sicherungskasten in der Kabine, hinter abschraubbaren Blenden im Fußbereich oder an sonstigen Positionen in der Kabine. Das Stecksystem des Diagnoseports ist verpolungssicher, sodass der WVA nur in einer Position in diesen gesteckt werden kann. Durch Drehen des Verschlussrings im Uhrzeigersinn wird der WVA gegen Vibrationen und ein Herausfallen aus dem Diagnoseport gesichert. Sobald der Adapter mit Strom versorgt wird, bestätigt dies die rechte LED (*PWR*). Der Adapter kann nun in den Wifi Einstellungen eines Tablet oder PC gefunden und mit einem Passwort verbunden werden. Ist das Gerät (Tablet oder PC) mit dem Adapter verbunden, kann man über das lokale Web Interface verschiedene Einstellungen vornehmen, welche für den Betrieb aber nicht zwingend notwendig sind. Das Web Interface bietet unter anderem die Einstellung des Power Managements, welches regelt, ab wann sich der Adapter abschaltet und wieder einschaltet. Das Einschalten kann zum Beispiel bei Bewegung, durch Knopfdruck, in bestimmten Abständen oder sobald die Lichtmaschine des Traktors Strom produziert geschehen.

Des Weiteren können Wifi-Einstellungen vorgenommen werden. Da nicht alle Tablets über mobiles Internet verfügen und während des Betriebes mit dem WVA über Wifi keine zweite Verbindung zu einem Smartphone, welches einen Hotspot für das Tablet bereitstellt, möglich ist, bietet der Adapter zudem die Möglichkeit des Infrastruktur-Netzes und fungiert als Access Point. Somit lässt sich der WVA mit einem von einem Smartphone zur Verfügung gestellten Hotspot verbinden und versorgt somit alle Geräte, die mit dem Adapter verbunden sind, mit mobilem Internet. Dies kann bei Datenaustausch mit anderen Endgeräten oder Hauptrechnern von Lohnunternehmen ein Vorteil sein.

## 4.2 Planung der Anwendung

Bevor die App entwickelt wurde, wurden alle späteren Funktionen und gestalterischen Aspekte der App mit einem Experten erörtert.

Um dazu alle möglichen Ausgabeparameter des zuvor vorgestellten Adapters zu berücksichtigen, wurde ein kurzes Programm geschrieben, welches eine Liste mit allen abfragbaren Parametern des Tools wiedergibt. Dabei ist zu berücksichtigen, dass Sattelzugmaschinen über den gleichen Standard (J1939) laufen und die meisten Auslesetools an diese

angepasst sind und nicht alle in der folgenden Liste stehenden Parameter aus einem Traktor auslesbar sind.

```
TotalDistance_J1587_130, EngineDesiredOperatingSpeed,  
TotalDistance_HOS, IgnitionSwitchStatus, VehicleSpeed_J1587_130,  
EngineOilPressure, BatteryPotential, TotalDistance_1, TripFuelUsed,  
VehicleSpeed_33, TotalDistance_0, TotalDistance_40, TotalDistance_3,  
HiResTotalDistance_17, Throttle, SeatBelt, BarometricPressure,  
AccelPedalPosition, PTOStatus, VehicleSpeed_J1587_128,  
TotalDistance_17, VehicleSpeed_J1587_144, HiResTotalDistance_40,  
VehicleSpeed_J1587_142, EngineSpeed, VehicleSpeed_J1587_140,  
TotalDistance_J1587_128, TripEngineHours, AccelPedalLowIdle,  
VehicleSpeed_40, ParkingBrake, TachographVehicleSpeed,  
TotalEngineHours, VehicleSpeed, HiResTotalDistance_33, TotalFuelUsed  
, HiResTotalDistance_3, EngineManifoldPressure, EngineCoolantTemp,  
HiResTotalDistance_1, HinoFuelRate, HiResTotalDistance_0,  
AmbientAirTemp, VehicleSpeed_3, TotalDistanceMiles_HOS,  
VehicleSpeed_0, EnginePercentLoadAtCurrentSpeed, VehicleSpeed_1,  
VehicleSpeed_17, TotalDistance_23, TransOilTemp, TransRequestedGear,  
FuelLevel, TransOutputShaftSpeed, CruiseControlStatus,  
HiResTotalDistance, TotalDistance, TotalDistance_J1587_142,  
TotalDistance_J1587_140, HiResTotalDistance_23,  
CruiseControlSetSpeed, HiResTripDistance,  
TransTorqueConverterLockupEngaged, TripDistance, VehicleSpeed_23,  
TotalDistance_33, FuelRate, FuelEconomy, TotalDistance_J1587_144,  
TransCurrentGear, TransTorqueConverterOilOutletTemp,  
HydraulicPressure, ATCControlStatus
```

Aus dieser Liste sind folgende Parameter in der Bewertung mit dem Experten, als für den Fahrer relevant eingestuft worden:

- **EngineOilPressure**, Motoröldruck
- **PTOStatus**, dies gibt den aktuellen Status der Zapfwelle(n) an, ob diese ein- oder ausgeschaltet sind
- **EngineSpeed**, Motordrehzahl
- **TotalEngineHours**, gibt die Betriebsstunden an
- **VehicleSpeed**, Geschwindigkeit des Fahrzeuges, gemessen über Raddrehzahlsensor
- **TotalFuelUsed**, der insgesamt verbrauchte Kraftstoff
- **EngineCoolantTemp**, Temperatur der Kühlflüssigkeit
- **AmbientAirTemp**, Umgebungstemperatur

- **EnginePercentLoadAtCurrentSpeed**, Aktuelle Motorlast [%]
- **TransOilTemp**, Getriebeöltemperatur [F]
- **FuelLevel**, Kraftstofftankinhalt [%]
- **TransOutputShaftSpeed**, Geschwindigkeit des Fahrzeuges gemessen über Drehzahlsensor am Getriebeausgang
- **TotalDistance**, die insgesamt zurückgelegte Strecke
- **FuelRate**, aktueller Kraftstoffverbrauch
- **HydraulicPressure**, Hydrauliköldruck

Zusätzlich soll die App die aktuelle Position auf einem Satellitenbild darstellen können und GPS Informationen mit auslesbaren Parametern des Traktors sinnvoll verbinden. Bei dem Gestalten einer grafischen Benutzeroberfläche, welche die zuvor genannten Kriterien erfüllt, sind weitere Aspekte zu berücksichtigen.

Das Anzeigegerät (Tablet) ist unter Umständen weiter von der Person entfernt als im normalen Tabletgebrauch, was zum Beispiel mangelnden Montagepunkten in einer Traktorkabine geschuldet sein kann. Weiterhin ist zu berücksichtigen, dass Traktoren über eine andere Federungstechnik als PKW verfügen. So besitzen viele Modelle keine gefederte Vorderachse, stattdessen ist jedoch der Sitz luftgefedert, was dafür sorgt, dass sich der Fahrer, bei unebener Fahrbahn oder auf dem Feld, mit dem Sitz sehr stark auf und ab bewegt. Dies führt dazu, dass eine feinfühlige Bedienung und Ablesen des Tablets stark erschwert sein kann. Dies führt zu der Anforderung, dass der Fahrer so wenig wie möglich mit der App interagieren muss und somit alle Parameter auf einen Blick sichtbar sind.

Bevor die App programmiert wurde, wurde ein Mockup erstellt, um eine geeignete Darstellungsform für die oben genannten Kriterien zu finden. Zum Erstellen eines Mockups wurde das Programm *InVision Studio* verwendet. Dieses Programm ermöglicht es, durch einfache Drag&Drop Technik interaktive Screens zu erstellen und diese miteinander zu verknüpfen. Mit der dazugehörigen App dieses Programms lässt sich das erstellte Mockup direkt auf dem Tablet anzeigen.

Im Folgenden (Abb. 4.2 und Abb. 4.3) sind zwei Bildschirme als Mockup zu sehen. Der eine Bildschirm zeigt die Anwendung für den deutschen Markt mit metrischem System und der andere Bildschirm die Anwendung für den amerikanischen Markt mit imperialen System. Der eine Bildschirm zeigt zusätzlich zu den Parametern des Traktors ein Satellitenbild, in das die aktuelle Position und der gefahrene Fahrtweg farblich abhängig von der Motorlast dargestellt wird. Der zweite Bildschirm zeigt den zeitlichen Verlauf der ausgewählten und grün hinterlegten Parameter als Graphen an. Zusätzlich wurde als weitere Visualisierungsmöglichkeit eine farbliche Drehmomentkurve in den unteren Bereich des Bildschirms gezeichnet. Dieser zweite Bildschirm wurde in Absprache mit dem Experten als nicht relevant bewertet, da eine Auswertung der angezeigten Graphen

durch den Fahrer während der Fahrt nicht erfolgen könnte und nicht aussagekräftig ist. Somit wurde zusammen mit dem Experten beschlossen, die Anwendung entsprechend des ersten Mockup (Abb. 4.2) zu entwickeln.

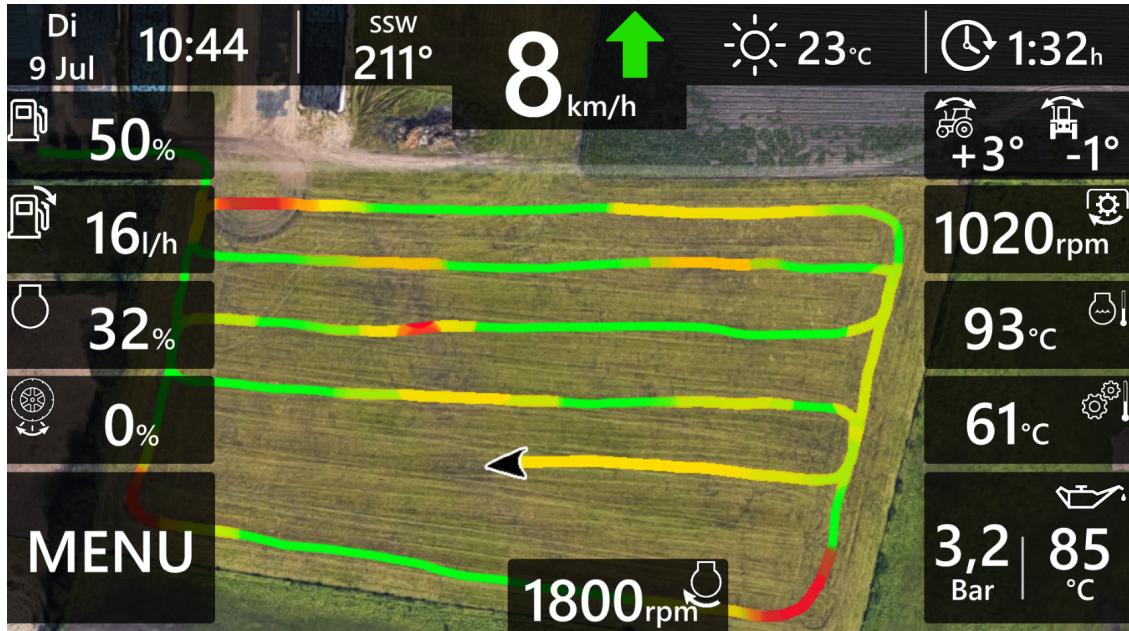


Abbildung 4.2: Mockup Bildschirm 1

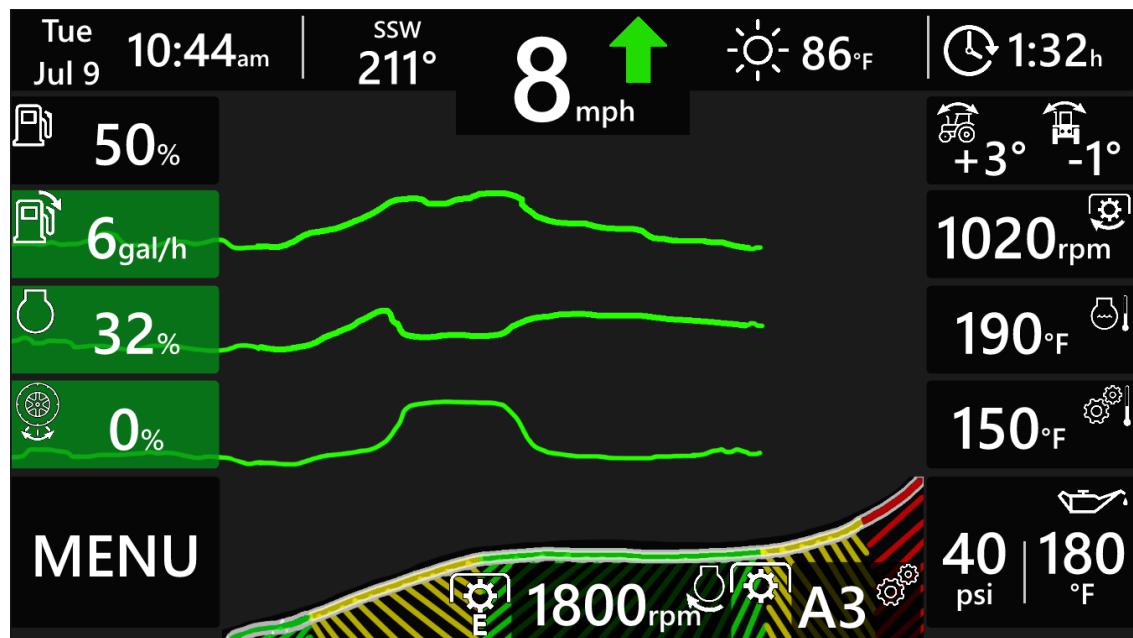


Abbildung 4.3: Mockup Bildschirm 2

## 4.3 Das Programm

Zum Schreiben der Android Applikation wurde das Programm Android Studio und die Programmiersprache Java verwendet. Dabei wurde sich gegen die weitere mögliche Programmiersprache *Kotlin* entschieden, um auf die im Verlauf des Studium erlernten Java Kenntnisse zurückgreifen zu können. Im Folgenden werden einige wichtige Funktionen und Besonderheiten des Quellcodes erläutert.

### Abfragen und Anzeigen von Maschinenparametern

Die Hauptfunktion der Applikation ist das Anzeigen von Maschinenparametern in Echtzeit. Hierzu wird eine Variable des Typs `TextView` zur Anzeige, eine Methode `VehicleDataListener` zum Empfangen von Maschinenparametern und eine Methode `subscription` für den entsprechenden Datenparameter benötigt.

Zuerst wird der `TextView`, welcher für das Anzeigen der Parameter an einer zuvor definierten Stelle auf dem Tablet zuständig ist, einer dem auszulesenden Wert entsprechend benannten Variable definiert. Da die Applikation sowohl für das metrische, als auch das imperiale System entwickelt wurde, wird speziell zwischen dem Wert `vTransTemp` und der Einheit `uTransTemp` unterschieden.

Im nächsten Schritt wird ein `vehicleDataListener` für den gewünschten Parameter gesetzt, welcher den Parameter auf dem Screen aktualisiert, sobald über den event Channel ein neuer Wert empfangen wird.

Im letzten Schritt wird die *Subscription* für den Parameter gesetzt und das gewünschte Aktualisierungsintervall definiert. Der Wert für das Intervall wurde in Abhängigkeit des Parameters der Subscription gewählt, so ist dieser bei Motordrehzahl und Fahrzeuggeschwindigkeit kleinstmöglich gewählt und bei Temperaturen von Betriebsstoffen im 15 Sekunden Intervall.

```
final TextView vTransTemp = (TextView) findViewById(R.id.value_transmissionTemp);
wva.setVehicleDataListener ("TransOilTemp", new VehicleDataListener() {
    @Override
360    public void onEvent(VehicleDataEvent event) {
        VehicleDataResponse response = event.getResponse();
        if (metric){
            vTransTemp.setText( setPrecisionTo0 (tempInMetric(response.getValue())));
        } else {
365            vTransTemp.setText( setPrecisionTo0 (response.getValue()));
        }
    }
});
wva.subscribeToVehicleData("TransOilTemp", 15);
```

Auf die Funktionen `setPrecisionTo0()`, `tempInMetric()` und den Wert `metric` wird auf den folgenden Seiten eingegangen.

### Genauigkeit der Anzeigeparameter

Um alle ausgelesenen Anzeigeparameter mit einer dem Parameter angemessenen Genauigkeit darstellen zu können, werden über die Funktionen `setPrecisionTo1()` und `setPrecisionTo0()` zum Beispiel die Temperaturen auf eine ganze Zahl aufgerundet. Distanzen werden auf eine Nachkommastelle aufgerundet.

```
1080 public String setPrecisionTo1 (Double value){
1081     if (!Double.isNaN(value)){
1082         value = BigDecimal.valueOf(value)
1083             .setScale (1, RoundingMode.HALF_UP)
1084             .doubleValue ();
1085         return Double.toString (value );
1086     } else {
1087         return "NaN";
1088     }
1089 }
```

### Metrisches und imperiales System

Bei Maschinen die mit dem imperialen System laufen und diese Einheiten verwenden, werden ebenfalls alle Werte, die über den WVA ausgelesen werden, in imperialer Größe dargestellt. Um hier eine Anpassung an Betriebe oder Fahrer einfach vornehmen zu können, welche das metrische System nutzen, gibt es die Funktion, die komplette App von dem imperialen System auf das metrische System umzurechnen.

Dies beginnt bereits bei der Anzeige von Datum und Uhrzeit. Während im imperialen System die Uhrzeit im 12 Stunden Format und das Datum mit *Monat - Tag - Jahr* angezeigt wird, wird im metrischen System das 24 Stunden Format und die Darstellung von *Tag - Monat - Jahr* genutzt. Der folgende Quellcodeausschnitt zeigt diese Umstellung zwischen metrisch und imperial.

```
260     String month =
261         new SimpleDateFormat("MMM", Locale.ENGLISH).format(calendar.getTime());
262     String dayInt =
263         new SimpleDateFormat("d", Locale.ENGLISH).format(calendar.getTime());
264     String monthDayInt;
265     if (metric){
266         time =
267             new SimpleDateFormat("HH:mm", Locale.ENGLISH).format(calendar.getTime());
268         monthDayInt = dayInt + " " + month;
```

```

} else {
    time =
        new SimpleDateFormat("hh:mm", Locale.ENGLISH).format(calendar.getTime());
    timeUnit =
        new SimpleDateFormat("a", Locale.ENGLISH).format(calendar.getTime());
    monthDayInt = month + " " + dayInt;
}

```

Die Variable `metric` des Typs `boolean` lässt sich in einem extra Menü in der App über eine Checkbox auswählen. Das Umrechnen von imperialen Einheiten zu metrischen Einheiten erfolgt auf zwei unterschiedliche Arten. Das Umrechnen von Temperaturen erfolgt über eine extra Funktion, `tempInMetric()`.

```

1100 public double tempInMetric(Double value){
    double metricTemp = (value - 32) / 1.8;
    return metricTemp;
}

```

Die zweite Möglichkeit, die genutzt wird, um imperiale Werte auf metrische Werte umzurechnen, ist das einfache Multiplizieren mit dem entsprechenden Faktor. Der durch den WVA ausgelesene Parameter wird grundsätzlich mit einer Variable namens `metricNAME` multipliziert. Diese Variable, z.B. `metricPressure` beträgt bei ausgewähltem imperialen System `metricPressure = 1` und bei ausgewähltem metrischem System `metricPressure = 0.06895`. Dies erfolgt in einer extra Methode `metric()`, in der ebenfalls alle angezeigten Einheiten von imperial auf metrisch umgestellt werden.

```

1295 if (metric){
    metricVelocity = 1.609;
    metricPressure = 0.06895;
    metricFuelRate = 3.78541;
    metricDistance = 1.61;
    metricDistanceString = "km";
    unitAmbientTemp.setText(" C ");
    unitCoolantTemp.setText(" C ");
    unitTransTemp.setText(" C ");
    unitOilTemp.setText(" C ");
    unitOilPress . setText ("bar");
    unitFuelRate . setText ("l/h");
    unitVelocity . setText ("km/h");
}
else { ...

```

### Farblicher Fahrtwegverlauf auf der Karte

Wie im ersten Bildschirm des Mockups zu sehen, soll der zurückgelegte Weg des Traktors farblich in Abhängigkeit von zum Beispiel der Motorlast auf der Karte dargestellt werden. Dabei sollen negative (schlechte) Werte rot und positive (gute) Werte grün dargestellt werden. Die Motorlast wird hierbei durch den WVA von dem Traktor in [%] ausgelesen, wobei eine möglichst geringe Motorlast als gut definiert wird. Die Umrechnung von [%] auf den entsprechenden Farbwert erfolgt mittels der Funktion `valueToColor()`

```

730 public String valueToColor( Integer srcValue){
    Integer destR;
    Integer destG;
    Integer destB = 0;
    if (srcValue <= 50){
        destR = Math.round(5.1f * srcValue );
        destG = Math.round(255f);
    } else {
        destR = Math.round(255f);
        destG = Math.round(255f - (5.1 f * (srcValue - 50)));
    }
    String destRString = Integer.toHexString(destR);
    destRString = ("00" + destRString ).substring (destRString .length ());
    String destGString = Integer.toHexString(destG);
    destGString = ("00" + destGString ).substring (destGString .length ());
    String destBString = Integer.toHexString(destB );
    destBString = ("00" + destBString ).substring (destBString .length ());
    String destColorString =destRString + destGString + destBString ;
    return destColorString ;
}

```

Nachdem die einzelnen Farbwerte berechnet und jeweils zu einem String als Abbild des Integers zur Basis 16 umgewandelt wurde, werden diese zu einem String zusammengefasst. Dieser wird in der Funktion `public void addPath()` zu einem Integer Basis 16 zurückgewandelt und im Anschluss durch folgende Zeile verwendet, um einen Bitmap, welcher auf der Karte angezeigt wird, mit der Funktion `colorize()` zu färben. Diese Funktion geht Pixel für Pixel durch den Bitmap und färbt diesen nach der zuvor errechneten Farbe.

```

696     .icon(BitmapDescriptorFactory .fromBitmap(colorize(dot, desColorInt )))

```

```

1 public Bitmap colorize(Bitmap srcBmp, int dstColor) {
2     int width = srcBmp.getWidth();
3     int height = srcBmp.getHeight();
4     float srcHSV[] = new float [3];
5     float dstHSV[] = new float [3];
6     Bitmap dstBitmap =
7         Bitmap.createBitmap(width, height, Bitmap.Config.ARGB_8888);
8     for (int row = 0; row < height ; row++) {
9         for (int col = 0; col < width; col++) {
10            int pixel = srcBmp.getPixel(col, row);
11            int alpha = Color.alpha( pixel );
12            Color.colorToHSV(pixel, srcHSV);
13            Color.colorToHSV(dstColor, dstHSV);
14            dstHSV[2] = srcHSV[2];
15            dstBitmap.setPixel (col, row, Color.HSVToColor(alpha, dstHSV));
16        }
17    }
18    return dstBitmap;
19 }
```

### Aufnahme

Damit der Fahrer die Möglichkeit hat alle gesammelten Daten zu speichern, wurde eine Aufnahmefunktion implementiert. Die Idee dabei ist, dass der Fahrer für jedes Feld, auf dem er arbeitet, eine separate Aufnahme starten kann. Wenn die Aufnahmetaste gedrückt wird, wird nach dem Feldnamen gefragt auf dem sich die Maschine befindet. Nachdem dies geschehen ist, werden in einem JSON Object mehrere Werte gespeichert und dieses JSON Object einem JSON Array hinzugefügt. Dieses JSON Array wird einmalig durch die Funktion `startRecord()` einer JSON Datei mit allen Startparametern hinzugefügt und im Anschluss mit einem vordefinierten Intervall durch die Funktion `writeToCurrentField()` fortlaufend aktualisiert und abgespeichert.

```

1 if (recording){
2     try {
3         JSONObject jo = new JSONObject();
4         jo.put("latLng", currLatLng);
5         jo.put("engineLoad", currentEngineLoad);
6         jo.put("tireSlip", currentTireSlip );
7         jo.put("engineRpm", currentEngineRpm);
8         trackListJA.put(jo );
9     } catch (Exception e){
10         e.printStackTrace ();
11 }
```

```

        }
        if ( saveIntervall >= 100){
            writeToCurrentField ();
            saveIntervall = 0;
    }else {
        saveIntervall++;
}

```

```

1160 public void startRecord ( final String field , Button recordButton){
1161     try {
1162         String currentDate =
1163             new SimpleDateFormat("dd.MM.yyyy--HH:mm").format(new Date());
1164         final File fileJson = new File ("path.json");
1165         String strFileJson = getStringFromFile ( fileJson . toString ());
1166         final JSONObject previousJsonObj = new JSONObject(strFileJson );
1167         final JSONObject jo= new JSONObject();
1168         final JSONArray ja = new JSONArray();
1169         currentField = field ;
1170
1171         jo.put("fieldName", field);
1172         jo.put("startDate", currentDate);
1173         jo.put("startEngineHours", currentEngineHours);
1174         jo.put("startDistance", currentDistance);
1175         jo.put("startFuelLevel", currentFuelLevel);
1176         jo.put("trackList", ja);
1177
1178         previousJsonObj.put( field , jo );
1179
1180         JSONObject currentJsonObject = previousJsonObj . put( field , jo );
1181
1182         writeJsonFile ( fileJson , currentJsonObject . toString ());
1183         recordButton . setText ("recording" + "_click_to_stop");
1184         recordButton . setMaxLines(2);
1185         recordButton . setBackgroundColor(Color.parseColor ("#8000FF00"));
1186         recording = true;
1187     }catch (Exception e){
1188         e. printStackTrace ();
1189     }
1190 }

```

Mit einem erneuten Klick auf den *Record-Button*, werden durch die Funktion **writeToCurrentField()** alle aktuellen Parameter als Endwerte gespeichert, um somit

eine Zusammenfassung der gesamten zurückgelegten Strecke, verbrauchtem Treibstoff und Arbeitszeit zu ermöglichen.

#### Berechnung des Reifenschlupfs

Um den Reifenschlupf zu berechnen, wird in der Funktion `calcTireSlip()` die über das Tablet festgestellte GPS Geschwindigkeit mit der über den WVA aus dem Traktor ausgelesenen Fahrzeuggeschwindigkeit verglichen und zu einem Prozentwert umgerechnet.

```
820 public Double calcTireSlip (Double vehicleSpeed){  
    Double tireSlip = vehicleSpeed - currentGpsSpeed;  
    tireSlip = tireSlip * 100;  
    tireSlip = tireSlip / vehicleSpeed;  
    currentTireSlip = tireSlip ;  
    if (Double.isNaN( tireSlip )){  
        tireSlip = 0.0;  
    }  
    return tireSlip ;  
825 }
```

#### Dokumentation der täglichen Arbeitszeit der Maschine

Um eine Kontrolle über die gesamte Arbeitszeit pro Tag der Maschine zu bekommen, werden die Betriebsstunden der Maschine in der selben JSON Datei gespeichert, in der auch die Fahrtwege gespeichert werden. Dies geschieht einmalig beim Starten der Applikation. Um bei mehrfachem Start der Applikation an einem Tag ein mehrfaches Schreiben der Betriebsstunden in die Datei zu verhindern, prüft die Funktion `checkEngineUptimeDay()`, ob an dem Tag die Betriebsstunden bereits in die Datei geschrieben wurden. Ist dies nicht der Fall, werden, mit den Funktionen `setEngineUptimeDay()` und `setTotalDistanceDay()`, die ausgelesenen Werte in der Datei abgespeichert. Die Tageswerte können im Anschluss durch die Funktionen `calcUptime()` und `calcUpDistance()` berechnet werden.



# 5 Ergebnisse

## 5.1 Integrationstest des Systems

### Userinterface

Das Ergebnis der grafischen Benutzeroberfläche kommt dem Mockup sehr nahe. Es wurde sich dafür entschieden, die Karte auf der gesamten Bildschirmfläche anzuzeigen. Dies soll den maximalen Überblick über die aktuelle Position ermöglichen. Überlagert wird diese Karte entlang des Bildschirmrandes von mehreren Kacheln in Schwarz mit einer mittelstarken Transparenz. Diese Kacheln bieten die visuelle Unterteilung zwischen den jeweiligen Anzeigeparametern und eine Abhebung von der Karte. Jeder Anzeigeparameter oder jede Parametergruppe, die sich auf einer Kachel befindet ist durch ein parameterbezogenes Symbol gekennzeichnet. Bei der Anzeige der Werte auf den Kacheln wurde darauf geachtet, dass die angezeigten Werte eine deutlich größere Schriftgröße aufweisen, als die zu diesen Werten gehörigen Einheiten. Auf dem ersten Screenshot (siehe Abb. 5.1) ist das grafische Userinterface in metrischer Darstellung zu sehen, ohne eine Verbindung zwischen Tablet und WVA. Hier wurde ein GPS Genauigkeitstest durchgeführt und dokumentiert. Mit dem Tablet in der Hand wurde ein Weg auf einem ca. 1,5m breiten Bürgersteig abgelaufen, dabei wurde über eine Testfunktion eine zufällige Motorlast simuliert und den entsprechenden Koordinaten zugewiesen. Gestartet wurde dieser Versuch in dem zu sehenden Gebäude, weswegen die ersten ermittelten Koordinaten eine hohe Abweichung aufweisen. Die Überprüfung der Genauigkeit außerhalb des Gebäudes ergab eine Abweichung unter einem Meter. Dies wurde zusammen mit dem Experten als für die geplante Anwendung völlig ausreichend gewertet. Auch die Darstellung der ortsabhängigen Motorlast lässt sich mit dem entwickelten System gut ablesen.

In dem zweiten Screenshot (Abb. 5.2) wurde selbiger Test über eine längere Wegstrecke durchgeführt, durch die Verwendung des Tablets in einem Auto. Auch hier erzielte das Tablet in Verbindung mit der Anwendung eine gut verwendbare GPS Genauigkeit. Ebenfalls ist der einige Stunden zuvor und oben erklärte vorherige gespeicherte GPS Test nach mehrfachem Neustart der Anwendung auf der Karte weiterhin zu sehen.

Ein Test der Anwendung in Verbindung mit einem laufenden Traktor ist in dem nachfolgenden Screenshot (Abb. 5.3) zu sehen. In diesem Fall wird dort keine Karte angezeigt.

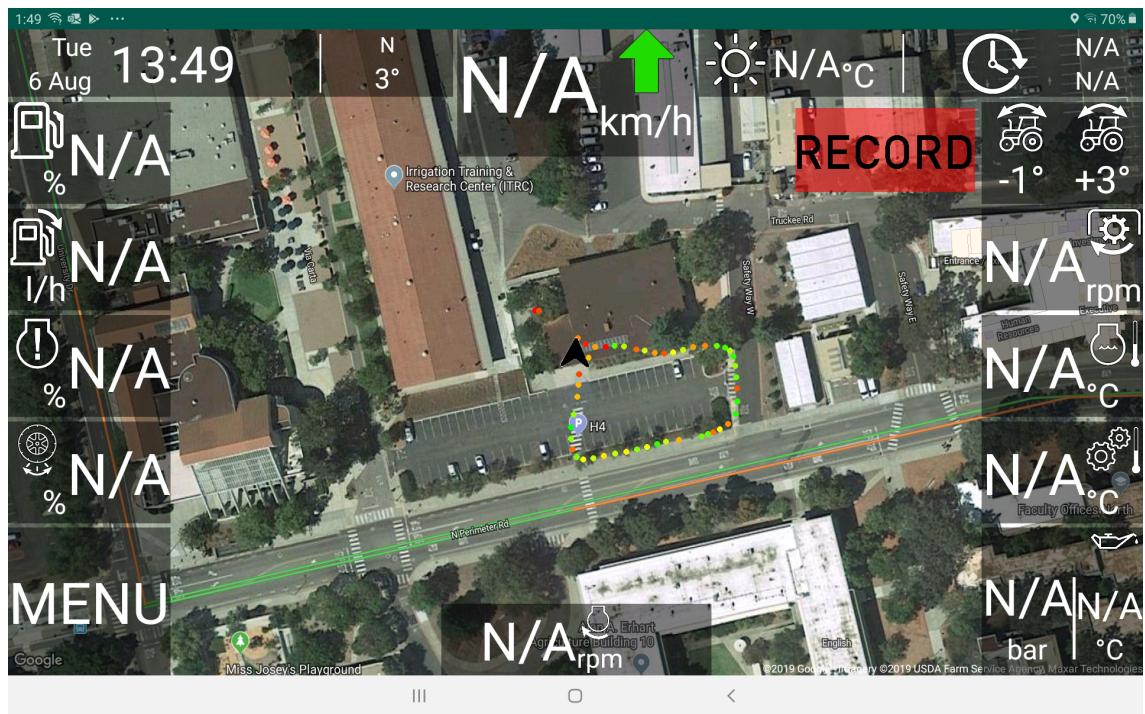


Abbildung 5.1: Test der GPS Genauigkeit

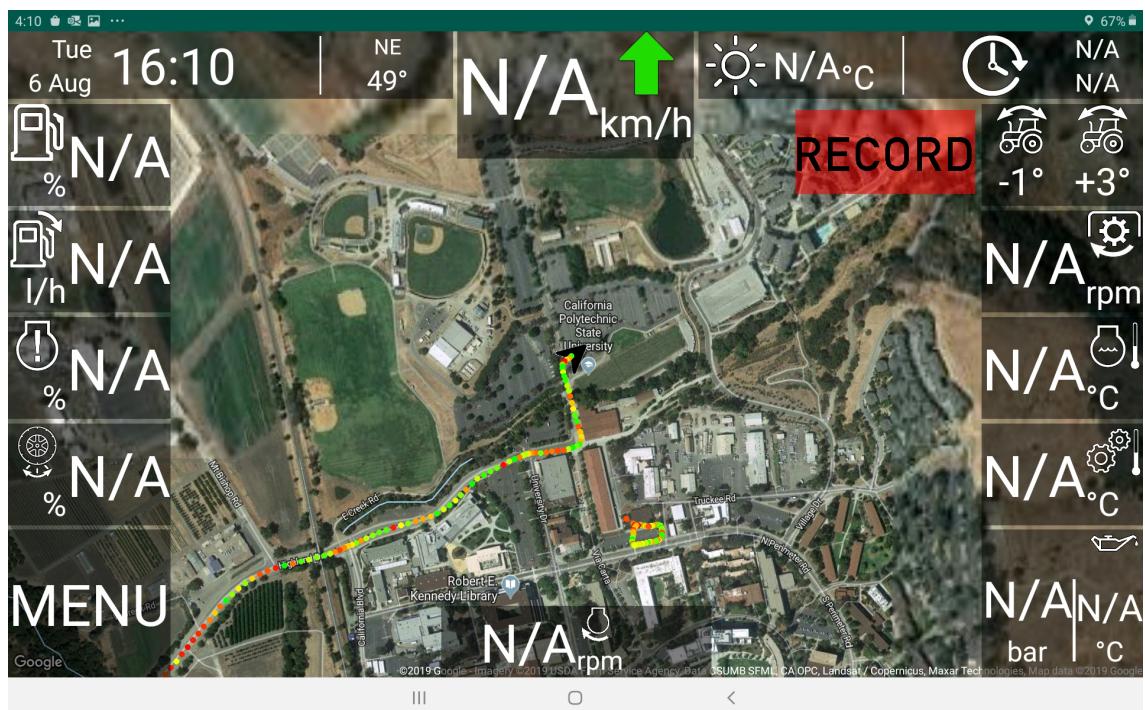


Abbildung 5.2: Darstellung von Fahrtweg und Motorlast

Dies liegt daran, dass das Tablet direkt mit dem WVA verbunden ist und keinen eigenständigen Zugriff auf ein mobiles Datennetz bietet. Bei solch einem Tablet, was nicht über diese Möglichkeit verfügt, empfiehlt es sich, wie bereits bei der Vorstellung des Adapters (siehe Abschnitt 4.1) beschrieben, den WVA als Access Point zu nutzen. Der Adapter kann mit einem Smartphone, welches einen Hotspot zur Verfügung stellt, verbunden werden und somit auch das Tablet mit einem Internetzugang versorgen und das Laden der Karte ermöglichen.

Zudem ist hier die imperiale Benutzeroberfläche zu sehen, mit dem imperialen Standard für Datum und Zeit und den imperialen Maßeinheiten.

Wie zu erkennen, wurde dieser Test vor der Implementierung der `setPrecisionToX` Funktionen durchgeführt und war Auslöser für diese. Sowohl bei dem Füllstand des Kraftstofftanks als auch der Anzeige des Motoröldrucks wird die Notwendigkeit deutlich, da, wie in der Abbildung (Abb. 5.3) zu sehen, diese Werte mit einer Genauigkeit von mehreren Nachkommastellen angezeigt werden, was nicht erforderlich ist. Aber auch weitere Parameter, wie zum Beispiel die Betriebsstunden am aktuellen Tag, oben rechts zu sehen, oder die Motordrehzahl, unten mittig zu sehen, nutzen die Funktionen.

Der *Record-Button* welcher eine Aufzeichnung der Fahrtwege und weiteren Parameter ermöglicht, fragt nach Betätigen nach einem Namen für die Aufzeichnung (Feldname), wird während der Aufzeichnung grün und zeigt den Text *recording click to stop* an.

Die Anzeige für die Motordrehzahl, für welche das Abfrageintervall auf die größtmögliche Frequenz gesetzt wurde, zeigt die Veränderung der Drehzahl flüssig und ohne Verzögerung an.

#### Funktionsfähigkeit abhängig vom Traktor

Es lässt sich sagen, dass die Applikation so funktioniert, wie es geplant war. Es sind lediglich zwei Probleme aufgetreten, auf die im Folgenden eingegangen wird.

Während der Entwicklung der Applikation in Amerika, standen insgesamt fünf Traktoren der Marke John Deere zur Verfügung (siehe Tabelle 3.1). Sowohl große als auch kleine Traktoren und Traktoren mit und ohne Kabine. An zwei dieser Traktoren, beide Traktoren der selben Modellreihe (John Deere 5525) konnte der WVA keinerlei Daten empfangen und keine CAN-Bus Aktivität feststellen. Eine Abhängigkeit dieses Problems ließ sich bis auf die Modellreihe nicht feststellen. Traktoren früherer und späterer Baujahre ließen sich auslesen, ebenfalls technisch geringerwertig und höherwertig ausgestattete Traktoren. Auch ein intensiver Kontakt mit den Entwicklern des WVA und das Upgrade auf eine nicht veröffentlichte Firmware des WVA erbrachte nicht den gewünschten Erfolg. Diese Firmware wurde erst kurz vor Abreise aus Amerika auf den WVA gespielt, da ein anschließendes erneutes Downgraden nicht möglich war, die Entwickler eine Funktionsfähigkeit der Firmware nicht garantieren konnten und der Autor die Entwicklung des Projektes nicht gefährden wollte.

Durch den Kontakt mit den Entwicklern, konnte erfahren werden, dass der WVA in erster

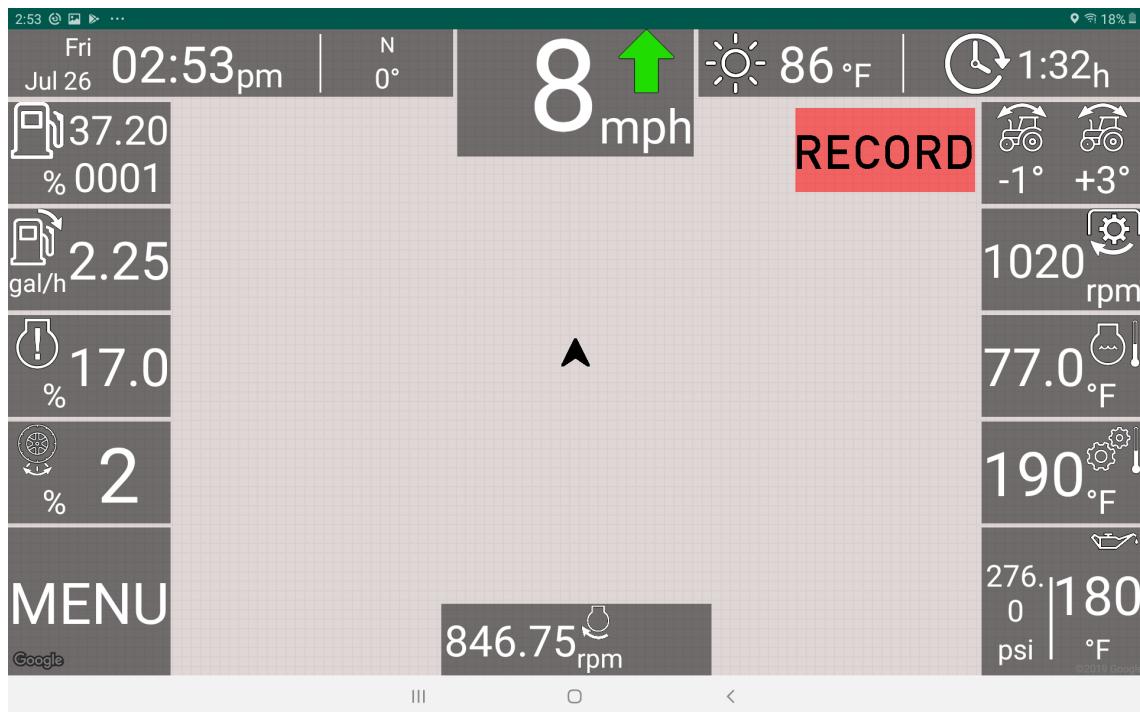


Abbildung 5.3: Test Echtzeitanzeige von Traktorparametern

Linie für Semi-Trucks, welche den gleichen Standard wie Traktoren verwenden, entwickelt wird und dieser auch ausschließlich an diesen Trucks getestet wird. Die Entwickler verfügten über keine Erfahrung des WVA in Kombination mit Traktoren und konnten somit das Problem nicht benennen.

An den auslesbaren Traktoren konnte festgestellt werden, dass einige Parameter, wie zum Beispiel die *AmbientAirTemp* (Umgebungstemperatur), über den WVA nicht ausgelesen werden konnten. Obwohl dieser als auslesbarer Parameter durch den WVA aufgelistet wurde (siehe Abschnitt 4.2) und der Traktor in der Lage war, diesen Wert zu ermitteln, konnte über den Adapter kein Wert festgestellt werden. Verifiziert wurde die Funktionsfähigkeit traktorseitig bei dem größten zur Verfügung stehenden Traktor, einem John Deere 7330 Premium, welcher über den zuvor vorgestellten Command Center Bildschirm verfügt (siehe Abb. 2.4 und Abb. 2.5) und die *AmbientAirTemp* anzeigen kann. Dieses Problem betraf zwei Parameter, die laut Experten relevant für den Fahrer sind und somit auch für die Anwendung. Hierbei handelt es sich zum einen um die bereits erwähnte Umgebungstemperatur und zum anderen um den Hydraulikdruck des Traktors. Beide Werte, die nachweisbar dem Traktor bekannt sind, konnten durch den Adapter nicht ausgelesen werden.

## 5.2 Online Benutzerumfrage

Die Online Benutzerumfrage wurde mit Google Forms (siehe Abschnitt 3.2) durchgeführt und über vier Landwirtschaftliche Foren durch einen Post verbreitet. Dieser Post wurde bis zum 26.01.2020 insgesamt 2548 mal angeschaut und es haben insgesamt 85 Personen an der Umfrage teilgenommen.

Im Folgenden werden die Auswertungen der einzelnen Fragen dargestellt.

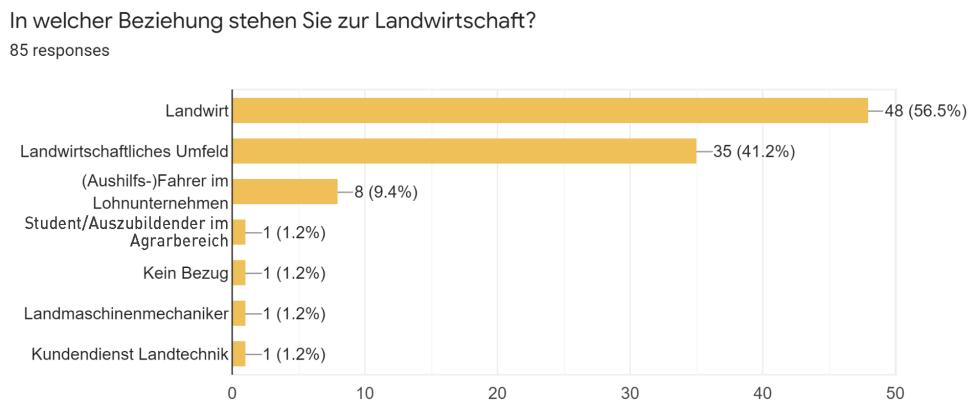


Abbildung 5.4: Benutzerumfrage: Beziehung zur Landwirtschaft

Die Teilnehmer der Benutzerumfrage sind zum größten Teil Landwirte oder Personen mit landwirtschaftlichem Umfeld (siehe Abb. 5.4). Lediglich eine Person ohne Bezug zur Landwirtschaft hat zusätzlich an der Umfrage teilgenommen.

Wie alt sind Sie?

85 responses

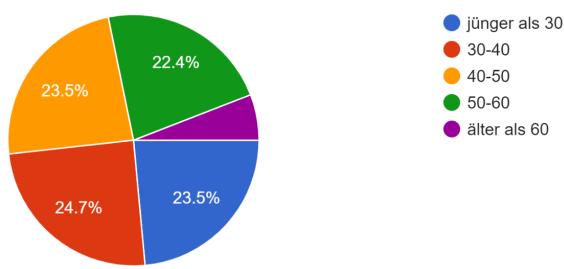


Abbildung 5.5: Benutzerumfrage: Alter der Teilnehmer

In der Benutzerumfrage sind alle Altersgruppen vertreten (siehe Abb. 5.5). Diese sind sehr gleichmäßig verteilt, bis auf Personen über 60 Jahre, diese machen hier die Minderheit mit nur 5,9% aus.

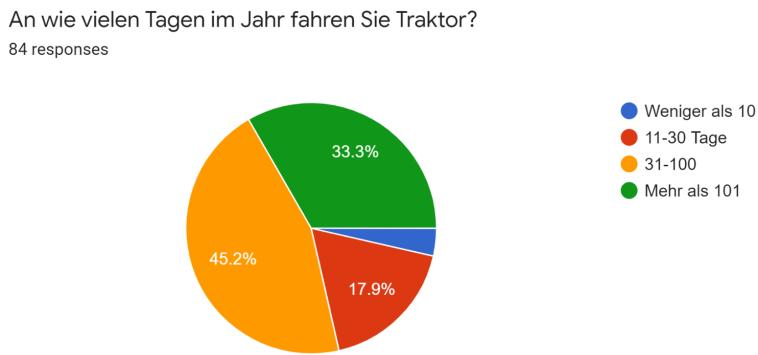


Abbildung 5.6: Benutzerumfrage: Bedienung von Traktoren in Tagen pro Jahr

84 der insgesamt 85 Teilnehmer haben angegeben Traktor zu fahren.  
Fast die Hälfte der Teilnehmer fährt im Jahr zwischen 31 und 100 Tagen Traktor, 1/3 der Teilnehmer mehr als 101 Tage im Jahr, der Rest weniger als 31 Tage (siehe Abb. 5.6).

Die Befragung ergab, dass 70,2% (59) der Teilnehmer einen Traktor fahren, der Baujahr 1995 oder neuer ist. 29,8% (25) Teilnehmer bedienen einen Traktor älter als Baujahr 1995. 78,6% (66) der Teilnehmer besitzen einen Traktor ohne Bildschirm, 21,4% (18) besitzen im Gegenzug einen Traktor mit Bildschirm.

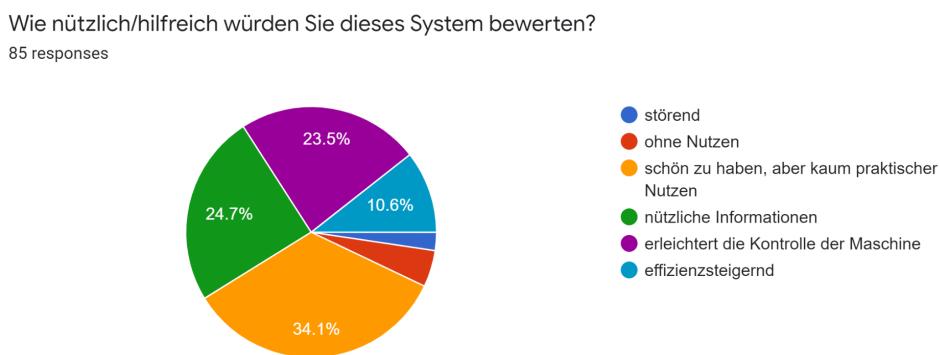


Abbildung 5.7: Benutzerumfrage: Nützlichkeit der gezeigten Anwendung

Die Bewertung der Nützlichkeit der Anwendung fällt sehr unterschiedlich aus, dennoch bewertet die Mehrheit die Anwendung als mindestens nützlich, bis hin zu effizienzsteigernd (siehe Abb. 5.7).

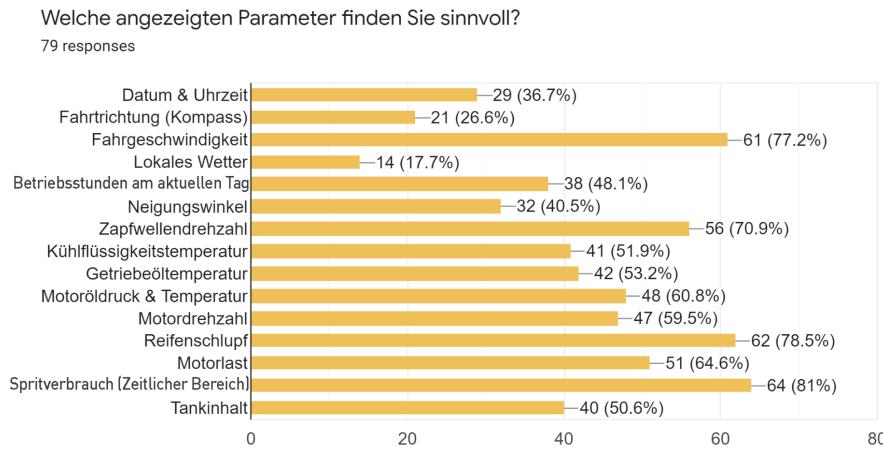


Abbildung 5.8: Benutzerumfrage: Sinnvolle Parameter

Einige Parameter, die in der Anwendung in Echtzeit angezeigt werden, werden von Teilnehmern als deutlich sinnvoller bewertet als andere Parameter. Die drei sinnvollsten Parameter sind: Kraftstoffverbrauch über einen einstellbaren Zeitraum bewertet, Reifenschlupf und Fahrgeschwindigkeit. Am wenigsten nützlich sind: Lokales Wetter, Fahrtrichtung (Kompass) und Datum und Uhrzeit (siehe Abb. 5.8).

Zu welchem Preis könnten Sie sich vorstellen so ein System nachzurüsten?

79 responses

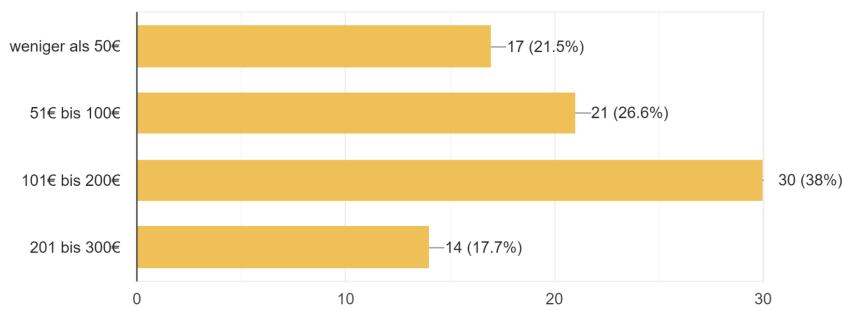


Abbildung 5.9: Benutzerumfrage: Vorstellbarer Preis des vorgestellten Systems

Die Mehrheit der Teilnehmer könnte sich vorstellen, ein solches System zu einem Preis von 101€ - 200€ als Nachrüstmöglichkeit für ihre Maschinen anzuschaffen (siehe Abb. 5.9).

In einem Freitextfeld der Umfrage konnten die Teilnehmer weitere gewünschte Funktionen nennen. Diese waren:

- Belastung der Hydraulik
- Spurführung und Parallelfahrhilfe
- Hektarzähler
- Warngrenzen z.B. für die Öldrucküberwachung, mit individuell einstellbaren Grenzwerten
- Wartungsintervalle
- Zuordnung zu einzelnen Schlägen, ggf. mit einer art Tätigkeitsberichten (z.B. x Stunden gegrubbert)
- Auswertung Monat/Jahr, Berechnung der Auslastung und Leistungsbedarf der Maschine (z.B. zur Auswahl der als nächstes anzuschaffenden Maschine)
- Export der Daten an Büro PC
- Auftragsmanagement

Weitere sonstige Vorschläge wurden von den Teilnehmern über ein abschließendes Kommentarfeld genannt:

- Mehrfaches Lob für die Idee und Umsetzung
- Die Applikation darf nicht in Konkurrenz zu bereits am Traktor vorhandenen Diagnosesystemen stehen.
- Ideal wäre ein Baukasten, mit dem man sich selbst den Bildschirm zusammenstellen kann.
- Eine System-Variante die Parameter nicht nur über CAN-Bus, sondern auch über Sensoren erfasst.

# 6 Diskussion und Fazit

## 6.1 Diskussion der Ergebnisse

### Ergebnisse der Anwendung

Die erzielten Ergebnisse mit der Anwendung sind aus Sicht des Entwicklers und auch des Experten sehr zufriedenstellend, da die grundsätzliche Idee und Funktionsweise dargestellt, erwiesen und evaluiert werden konnte. Ebenfalls konnte die erste Hypothese, dass es möglich ist, solch ein System kostengünstig zu entwickeln und maschinenungebunden einsetzbar zu machen, zum Teil bestätigt werden. Wenn ein Android Tablet und eine passende Tablethalterung für den Traktor zur Verfügung steht, belaufen sich die Anschaffungskosten für das hier vorgestellte System auf etwa 250€. Der grundsätzliche maschinenungebundene Einsatz konnte getestet und ebenfalls bestätigt werden.

Bevor man die Entwicklung des vorgestellten Systems fortführt, sollten zunächst die zwei angesprochenen Probleme behandelt werden (siehe Abschnitt 5.1). Hierzu muss ermittelt werden, bei welchen Maschinen Fehler 2 (keine Parameter auslesbar) auftritt um den tatsächlichen Grund für diesen zu ermitteln. Um dies durchzuführen, benötigt man verschiedene Maschinen unterschiedlicher Hersteller, Baujahre und Baureihen. Es besteht die Möglichkeit, dass dieser Fehler mit dem aktuell verwendeten CAN-Bus Adapter nicht behoben werden kann und somit auf einen anderen Adapter gewechselt werden muss. Dies könnte eine Reimplementierung großer Teile des Quellcodes mit sich bringen, weshalb dieser Fehler genau analysiert werden muss, bevor eine Weiterentwicklung der Software erfolgt. Es besteht ebenfalls die Möglichkeit, dass dieses Problem auf der Seite des Traktors vorliegt und hier ein Defekt in der Elektronik herrscht. An einem der beiden nicht funktionierenden Traktoren wurde bereits für Studienzwecke viel an der Elektronik verändert. Die Funktionsfähigkeit der Elektronik der Traktoren ließe sich allerdings relativ einfach mit einem vom Hersteller zertifizierten Diagnosegeräts bestätigen oder ausschließen.

Gleicher sollte im Anschluss mit Fehler 1 (bestimmte Parameter sind nicht auslesbar) passieren, da die Lösung zu diesem Problem auch zu der Verwendung eines anderen CAN-Bus Adapters führen kann.

Ebenfalls wären weitere realitätsnahe Tests bei mehrstündiger Feldarbeit hilfreich, um das grafische Userinterface bei unterschiedlichen Lichtverhältnissen zu überprüfen. Viele

Tablets neigen zu einer starken Spiegelung der einfallenden Sonnenstrahlen. Solche Tests ermöglichen es außerdem potentiell weitere Fehler in der Software festzustellen. Ebenso kann eine Auswertung der Dokumentation der Maschinenparametern, welche über einen längeren Zeitraum durchgeführt wurde, vorgenommen werden. Auch kann ermittelt werden ob eine Arbeitserleichterung, Beeinträchtigung oder Ablenkung des Fahrers eintritt.

### Ergebnisse der Benutzerumfrage

An der Benutzerumfrage haben insgesamt 85 Personen teilgenommen, die entweder selbst Landwirte oder Personen mit landwirtschaftlichem Umfeld sind. Es wurden in gleichmäßiger Verteilung Teilnehmer jeder Altersklasse erreicht. 78,5% der Befragten bedienen regelmäßig (mindestens 31 Tage im Jahr) einen Traktor. 70,2% der Teilnehmer bedienen einen Traktor des Baujahrs 1995 oder neuer und somit bereits über einen CAN-Bus verfügt. Von diesen besitzen lediglich 30,51% einen integrierten Bildschirm. Dies bestätigt die Motivation dieser Arbeit, eine günstige Nachrüstmöglichkeit für Traktoren mit CAN-Bus zu entwickeln. Die Auswertung der Nützlichkeit ergab, dass 41,2% aller Teilnehmer die vorgestellte Anwendung höchstens mit einem minimalen Nutzen bewerten würden, die Mehrheit (58,8%) sieht eine Nützlichkeit bis hin zu einer Effizienzsteigerung in der Anwendung. Betrachtet man das Ergebnis, welche der implementierten Parameter als sinnvoll erachtet wurden, lässt sich feststellen, dass Parameter, die durch andere Geräte abrufbar sind, wie das Wetter oder die Uhrzeit, eine sehr geringe Relevanz aufweisen. Ebenfalls werden Parameter, welche in den meisten Fällen auch bei Traktoren ohne Bildschirm angezeigt werden oder beim Betrieb nicht dauerhaft überprüft werden müssen, wie zum Beispiel Tankinhalt, Kühlflüssigkeitstemperatur oder Motordrehzahl, als weniger relevant betrachtet als Parameter, die eine ständige Überprüfung erfordern, wie zum Beispiel Fahrgeschwindigkeit und Zapfwellendrehzahl, oder Parameter, die eine spezielle Berechnung erfordern wie der Spritverbrauch in einem zeitlich einstellbaren Bereich oder der Reifenschlupf.

Besonders überraschend war das Ergebnis nach der Frage des vorstellbaren Anschaffungspreises, hier wären 55,7% bereit mehr 101€ in die Nachrüstung der vorgestellten Anwendung zu investieren.

Ebenfalls lassen sich dank der Benutzerumfrage die Hypothesen bezüglich der Senkung der Umweltbelastung und der Minimierung der Belastung für den Fahrer teilweise bestätigen. Eine Effizienzsteigerung der Maschinen, welche 10,6% der Teilnehmer mit der Anwendung erwarten, kann zu einer Minimierung der Umweltbelastung führen. Zusätzlich zu diesen 10,6% gehen 23,5% der Teilnehmer von einer erleichterten Kontrolle der Maschinen durch diese Anwendung aus. Womit auch die Hypothese der Minimierung der Fahrerbelastung teilweise bestätigt werden kann. – Die Bestätigung wird als *teilweise* betrachtet, da nicht alle Teilnehmer die Anwendung als erleichternd oder effizienzsteigernd bewertet haben.

Das Ergebnis der Benutzerumfrage, gerade unter Berücksichtigung der sehr gleichmäßig verteilten Altersgruppen, ist besser ausgefallen als erwartet, es wurde nicht mit so ei-

nem starken Interesse und Akzeptanz eines solchen Systems gerechnet. Dank der vielen individuellen Kommentare lässt sich vermuten, dass mit einer Weiterentwicklung der Anwendung die Zahl 58,8% Nutzen sehender Teilnehmer weiter gesteigert werden könnte. Großes Interesse besteht hier offenbar in der genaueren Dokumentation und Auswertung gesammelter Werte um Maschinenparameter in Abhängigkeit mit Art und Ortschaft der Tätigkeit genauer in Bezug setzen zu können und ein wirtschaftlicheres Arbeiten zu ermöglichen.

Ebenfalls könnte das Erweitern der Anwendung auf die Auswertung von Diagnose-Fehlercodes eine Steigerung des Nutzenfaktors mit sich bringen. Der in dieser Arbeit vorgestellte verwendete CAN-Bus Adapter bietet diese Möglichkeit bereits und es wäre somit nur eine Erweiterung der Anwendung softwareseitig nötig.

## 6.2 Fazit

Mit dieser Arbeit konnte gezeigt werden, dass landwirtschaftliche Maschinen, die über einen CAN-Bus (J1939) verfügen, auf einfache Weise mit einem System zur Echtzeit-Visualisierung von Maschinenparametern nachgerüstet werden können. Dies eröffnet die Möglichkeit sowohl ältere Maschinen, die über keinen Bildschirm verfügen, technologisch mit neuen Funktionen aufzurüsten, als auch neuere Maschinen, die bereits einen Bildschirm besitzen, mit einem weiteren Bildschirm zu ergänzen.

Die Verwendung des J1939 Standards erlaubt zudem grundsätzlich die Nutzung des vorgestellten Systems auf Maschinen verschiedener Baujahre und Hersteller. Zusammen mit dem ebenfalls standardisierten Diagnoseport begünstigt dies sogar den unkomplizierten Wechsel des Systems zwischen verschiedenen Maschinen. Die zusätzliche Verwendung von durch die Systemkomponente Tablet nutzbarer Technologien, wie das tabletinterne GPS oder mobile Datenübertragung, bringen der Anwendung einen weiteren Nutzen.

58,8% der Teilnehmer der Benutzerumfrage bestätigen bereits eine Nützlichkeit der aktuellen Anwendung. 70,7% der Teilnehmer, die über einen Traktor mit CAN-Bus verfügen, haben angegeben, dass ihre Maschine nicht mit einem Bildschirm ausgerüstet ist. Damit ergibt sich ein erhebliches Nachrüstpotential, und für 38% der Teilnehmer ist ein Nachrüstpreis von 100-200 € vorstellbar.

Weiterer Mehrwert für die Anwendung könnte durch das direkte Auslesen von Fehlercodes erzeugt werden. Zusätzliches Potential lässt sich vermuten, wenn Funktionen wie z.B. das Speichern und die Auswertung gesammelter Daten über einen langen Zeitraum oder die Fernüberwachung der Maschinen für einen großen Betrieb ergänzt würden. Die Kombination der Anwendung mit Cloud-Diensten könnte zu einer erleichterten und teilweise automatisierten Datenverwaltung führen.

Die Arbeit hat gezeigt, dass es einfach möglich ist externe Geräte an einen CAN-Bus anzuschließen und Daten auszulesen. Allerdings ist es genau so einfach möglich, Daten in den CAN-Bus einzuspeisen und damit Einfluss auf die Maschine zu nehmen. Dies hat der Autor im Rahmen des PolyCAN Projektes [PolyCAN 2019] selber demonstriert indem er den Tachometer eines Traktor manipuliert hat. Auch andere Autoren weisen auf diese Risiken hin [Burakova u. a. 2016].

# Anhang



# Schriftliche Versicherung

Hiermit versichere ich, dass ich meine Bachelorarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen sind, habe ich als solche kenntlich gemacht.

Bremen, 14. Februar 2020

---

Colin Weber



# Abbildungsverzeichnis

2.1	11-Bit und 29-Bit Identifier CAN Frame [Eigene Grafik, angelehnt an: Lun u. a. 2010] . . . . .	5
2.2	29-Bit CAN Identifier nach SAE J1939 [Vector 2018] . . . . .	6
2.3	Deutsch HD10-9-1939 J1939 Diagnoseport Pinbelegung [Marx u. a. 2015]	7
2.4	Bildschirm eines Command Center in einem John Deere 6030 Premium [John Deere 2016] . . . . .	9
2.5	Gesamtübersicht eines Command Center in einem John Deere 6030 Pre- mium [John Deere 2013] . . . . .	9
2.6	Command Center 4. Generation [John Deere 2020a] . . . . .	10
2.7	AGCO Smart Connect [Fuse 2019] . . . . .	11
4.1	Digi Wireless Vehicle Bus Adapter (WVA) [eigenes Bild] . . . . .	20
4.2	Mockup Bildschirm 1 . . . . .	24
4.3	Mockup Bildschirm 2 . . . . .	24
5.1	Test der GPS Genauigkeit . . . . .	34
5.2	Darstellung von Fahrtweg und Motorlast . . . . .	34
5.3	Test Echtzeitanzeige von Traktorparametern . . . . .	36
5.4	Benutzerumfrage: Beziehung zur Landwirtschaft . . . . .	37
5.5	Benutzerumfrage: Alter der Teilnehmer . . . . .	37
5.6	Benutzerumfrage: Bedienung von Traktoren in Tagen pro Jahr . . . . .	38
5.7	Benutzerumfrage: Nützlichkeit der gezeigten Anwendung . . . . .	38
5.8	Benutzerumfrage: Sinnvolle Parameter . . . . .	39
5.9	Benutzerumfrage: Vorstellbarer Preis des vorgestellten Systems . . . . .	39



# Tabellenverzeichnis

3.1	Auflistung aller zur Verfügung stehenden Traktoren . . . . .	17
4.1	Vergleich des NEXIQ Blue-Link Mini mit dem Digi WVA Adapter . . . .	20



# Literaturverzeichnis

- Burakova, Yelizaveta, Bill Hass, Leif Millar und Andre Weimerskirch, Hrsg. (2016). *Truck Hacking: An Experimental Analysis of the SAE J1939 Standard* (siehe S. 44).
- Digi-International (2020). *Wireless Vehicle Bus Adapter (WVA)*. URL: <https://www.digi.com/products/networking/gateways/wireless-vehicle-bus-adapter#overview> (besucht am 09.02.2020) (siehe S. 19).
- Fendt (2019). *Smart Farming: Fendt Connect - Smart Connect*. URL: <https://www.fendt.com/at/smart-farming-fendt-connect-smart-connect> (besucht am 31.12.2019) (siehe S. 11).
- Fuse (2019). *Smart Connect: Mapping and machine performance visualization app*. URL: <https://www.fusesmartfarming.com/products/smartconnect/> (besucht am 31.12.2019) (siehe S. 11).
- Hartwich, Florian und Robert Bosch, Hrsg. (2012). *CAN with Flexible Data-Rate* (siehe S. 5).
- Heck, Edward T., John Kitzerow und Tony Caravella, Hrsg. (1999). *Large Scale Application of J-1939 CAN* (siehe S. 1).
- ISO 11898-1:2015(E) (2015). *Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical signalling*. Standard. International Organization for Standardization (siehe S. 5).
- John Deere (2006). *Neu: Premium Traktoren 6030 von 110 bis 180 PS*. URL: [http://www.schweizer-eiken.ch/pdf/brochure\\_6030\\_premium\\_de.pdf](http://www.schweizer-eiken.ch/pdf/brochure_6030_premium_de.pdf) (besucht am 11.02.2020) (siehe S. 7).
- John Deere (2013). *John Deere Sales Manual: Controls, CommandCenter*. URL: [https://salesmanual.deere.com/sales/salesmanual/en\\_NA/tractors/2012/feature/operator\\_station\\_and\\_controls/6030p\\_7030p/6030p\\_7030p\\_commandcenter.html](https://salesmanual.deere.com/sales/salesmanual/en_NA/tractors/2012/feature/operator_station_and_controls/6030p_7030p/6030p_7030p_commandcenter.html) (besucht am 06.01.2020) (siehe S. 8, 9).
- John Deere (2016). *Loadable Language Manual*. URL: [https://www.deere.de/assets/pdfs/region-2/parts-and-service/stellar-support/de/Loadable\\_Language\\_Manual\\_MY16-DE.pdf](https://www.deere.de/assets/pdfs/region-2/parts-and-service/stellar-support/de/Loadable_Language_Manual_MY16-DE.pdf) (besucht am 11.02.2020) (siehe S. 9).
- John Deere (2020a). *Gen4*. URL: <https://www.deere.com/en/technology-products/precision-ag-technology/guidance/gen-4-premium-activation/> (besucht am 11.02.2020) (siehe S. 10).

- John Deere (2020b). *John Deere Technologien Für Die Präzisionslandwirtschaft*. URL: <https://www.deere.de/assets/publications/index.html?id=b52d9bb8#1> (besucht am 06.01.2020) (siehe S. 8).
- Lun, Wei, Chee Kyun Ng, Borhanuddin Mohd Ali, Nor Ali, Fakhrul Noordin und F.Z. Rokhani (2010). „Review of Researches in Controller Area Networks Evolution and Applications“. In: *Proceedings of the Asia-Pacific Advanced Network* 30. DOI: 10.7125/APAN.30.3 (siehe S. 5).
- Marx, Samuel E., Joe D. Luck, Roger M. Hoy, Santosh K. Pitla, Erin E. Blankenship und Matthew J. Darr (2015). „Validation of machine CAN bus J1939 fuel rate accuracy using Nebraska Tractor Test Laboratory fuel rate data“. In: *Computers and Electronics in Agriculture* 118, S. 179–185 (siehe S. 7).
- McFall, Kevin und Timothy D. Fisher, Hrsg. (2016). *A Beginner's Guide to Controller Area Network Bus Access in Modern Vehicles* (siehe S. 5).
- NEXIQ-Technologies (2020). *NEXIQ Blue-Link2122 Mini*. URL: <https://www.nexiq.com/Product/Detail/126015> (besucht am 09.02.2020) (siehe S. 19).
- PolyCAN (2019). *Hacking Tractors for Repair*. URL: <https://tractorhacking.github.io/> (besucht am 31.12.2019) (siehe S. 3, 13, 44).
- Robert Bosch GmbH (1991). *CAN Specification version 2.0*. Techn. Ber. Robert Bosch GmbH (siehe S. 5).
- SAE (2016). *High-Speed CAN (HSC) for Vehicle Applications at 500 kbps with CAN FD Data at 5 Mbps*. Standard. Society of Automotive Engineers (siehe S. 5).
- SAE (2018). *Serial Control and Communications Heavy Duty Vehicle Network*. Standard. Society of Automotive Engineers (siehe S. 6).
- statCounter (2019). *Tablet Operating System Market Share Worldwide*. URL: <https://gs.statcounter.com/os-market-share/tablet/worldwide> (besucht am 14.02.2020) (siehe S. 19).
- Steve Corrigan, Texas Instruments, Hrsg. (2002). *Introduction to the Controller Area Network (CAN)* (siehe S. 6).
- The Climate Corporation (2018). *Die Plattform für digitale Landwirtschaft zur datengestützten Ertragsoptimierung*. URL: <https://climatefieldview.de/> (besucht am 31.12.2019) (siehe S. 12).
- Umehara, Daisuke und Takeyuki Shishido (2019). „Controller Area Network and Its Reduced Wiring Technology“. In: *IEICE Transactions* 102-B, S. 1248–1262 (siehe S. 5).
- US Patent No. 10.256.969 B1, Hrsg. (2019). *SCRAMBLING DATA FOR USE WITH VEHICLE COMMUNICATION PROTOCOL*. Ricardo Castano Salinas und John Forrest Smith. The Heil Co. (siehe S. 6).
- Vector (2018). *Lösungen für Ihre SAE J1939-Vernetzung*. URL: <https://www.vector.com/de/de/know-how/technologien/protokolle/sae-j1939/> (besucht am 11.02.2020) (siehe S. 6).