



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

**Konzeption und Implementierung einer IoT-basierten
Verkehrsstromanalyse zur Ermittlung des Mobilitätsbedarfs
im Technologiepark Berlin-Adlershof**

Masterarbeit

Name des Studiengangs

Wirtschaftsinformatik

Fachbereich 4

vorgelegt von

Oksana Tsurkan

Datum:

Berlin, 22.09.2020

Erstgutachter: Prof. Dr. Olga Willner

Zweitgutachter: M.Sc. Lukas Becker

Danksagung

An dieser Stelle möchte ich meinen besonderen Dank allen Personen aussprechen, die mich während der Bearbeitung der Masterarbeit unterstützt haben. In erster Linie gilt dieser Dank meinen Betreuern und Prüfern Frau Prof. Dr. Olga Willner und Herrn M.Sc. Lukas Becker, die sich stets Zeit genommen haben und mich mit ihrem fachlichen Wissen sowie wertvollem Feedback unterstützt haben.

Des Weiteren möchte ich mich besonders bei Herrn Prof. Dr.-Ing. Nicolas Lewkowicz, Herrn M. Eng Erik Zinger und Herrn Christopher Horz von der Beuth Hochschule bedanken, welche mir mit ihrem umfangreichen technischen Wissen stets sehr hilfreiche Hinweise gegeben haben und mich bei der Hardwareentwicklung unterstützt haben.

Ebenfalls möchte ich mich bei Herrn Prof. Dr. rer. nat. Peter Wagner und Dipl.-Ing. Sebastian Schulz vom Deutschen Zentrum für Luft- und Raumfahrt für die Ermöglichung der Verkehrserhebungen an der Mess- und Versuchsstrecke am Ernst-Ruska-Ufer in Adlershof bedanken. Ein besonderer Dank gilt ihnen für die Unterstützung bei der Gehäuseentwicklung, Montage sowie Inbetriebnahme von den Detektoren.

Weiterhin möchte ich mich herzlich bei der Firma WISTA Management GmbH und allen Mitarbeitern bedanken, die die Durchführung dieser Arbeit ermöglicht und unterstützt haben.

Kurzfassung

Das Ziel dieser Masterarbeit ist, aktuelle Technologien der Verkehrsstromanalyse miteinander zu vergleichen und deren Eignung zur Ermittlung des Mobilitätsbedarfs im Wissenschafts- und Wirtschaftsstandort Adlershof zu bewerten. Für die ausgewählten Technologien ist ein allgemeingültiges Konzept zu Erfassung und Klassifizierung der Verkehrsgruppen Fußgänger, Fahrradfahrer und motorisierter Individualverkehr zu erarbeiten und dieses prototypisch umzusetzen. Die prototypische Umsetzung soll die Hardware, deren wetterbeständiges Gehäuse sowie Integration ins vorhandene LoRaWAN Netz beinhalten. Die Präzision der Erhebungsergebnisse ist zu bewerten und zu interpretieren.

Um diese Forschungsfragen zu beantworten, ist zunächst eine Literaturrecherche durchgeführt worden, mit dem Ziel Technologien der automatisierten Verkehrsstromanalyse zu untersuchen und miteinander zu vergleichen. Mithilfe einer Nutzwertanalyse sind drei Verkehrserfassungssysteme für den Einsatz im Technologiepark Adlershof ausgewählt worden – jeweils eine für jede Verkehrsgruppe. Dabei wird der Druckdetektor für die Erfassung von Radfahrerzahlen auf einem Fahrradweg eingesetzt, der Radardetektor für die Erhebung der MIV-Mengen auf einer Fahrbahn und der Ultraschalldetektor für die Erkennung der Fußgänger auf einem Gehweg.

Es wird ein Konzept vorgestellt, das ein von der Verkehrsdatenerhebung bis zur Datenvisualisierung durchgehenden Verarbeitungsprozess beschreibt. Während das Konzept generisch gehalten worden ist, fand die prototypische Umsetzung im Technologiepark Adlershof statt und wurde an die ortspezifischen Besonderheiten angepasst.

Die Messergebnisse wurden unterschiedlich validiert. Die Radardetektorzahlen wurden mit erfassten Zahlen der Induktionsschleifen verglichen. Trotz großer absoluter Abweichungen lieferte der Grove-Radarsensor aussagekräftige Trendwerte. Ähnlich ist es mit dem Grove-Ultraschallsensor, der in der Reichweite bis zu 1 Meter gute Trendwerte liefert, aber ab 1 Meter Abstand ungenau wird. Der Grove-Drucksensor war auch nach der Anpassung des Spannungsoutputs nicht in der Lage ein über den angeschlossenen Gummischlauch fahrendes Fahrrad zu erkennen.

Um die Ergebnisse dieser Verkehrserhebungen genauer zu interpretieren, sollen Messungen über einen längeren Zeitraum durchgeführt werden. Weiterführende Forschung könnte sich mit anderen Technologien der Erfassung von Fahrradfahrern beschäftigen. Außerdem können unterschiedliche Verfahren des Data Minings für die Hochrechnung der Messergebnisse sowie Erhöhung deren Präzision eingesetzt werden.

Abstract

The aim of this master's thesis is to compare current technologies of traffic flow analysis and to evaluate their suitability for determining the mobility needs in the WISTA Science and Technology Park in Berlin-Adlershof. For selected technologies, a general concept had to be worked out, consisting of the guideline for detecting and classifying the traffic groups (pedestrians, cyclists and motorized individual traffic). This concept should be implemented through a prototype, which should consist of hardware, its weatherproof housing and integration into the existing LoRaWAN network. The precision of the measurement results had to be evaluated and discussed.

To answer these research questions, a literature research was carried out with the aim of examining and comparing technologies of automated traffic flow analysis. With the help of a cost-benefit analysis, three traffic recording systems were selected - one for each traffic group. The pressure detector is used to record the number of cyclists on the cycle paths, the radar detector to record motorized vehicles on the road and the ultrasound detector to measure the number of pedestrians on the sidewalks.

A concept that describes the process from traffic data collection to data visualization is presented. While the concept was kept generic, the prototypical implementation took place in the Technology Park in Berlin-Adlershof and was adapted to the site-specific features. The measurement results were validated differently. The radar detector results were compared with the numbers recorded by the induction loops. Despite large absolute deviations, the Grove radar sensor provided meaningful trend values. This was also the case with the Grove ultrasonic sensor, which provides good trend values within a range of up to one meter but became imprecise starting from one meter. The Grove pressure sensor was not able to detect a bicycle riding over the connected rubber hose, even after adjusting the voltage output of the sensor.

To interpret the results of these traffic surveys more accurately, measurements over a longer period are to be carried out. Further research could deal with other technologies for detecting cyclists. In addition, different data mining methods can be used to extrapolate the measurement results and increase their precision.

Inhaltsverzeichnis

Danksagung	I
Kurzfassung.....	II
Abstract	III
Inhaltsverzeichnis.....	IV
Abbildungsverzeichnis.....	VII
Tabellenverzeichnis.....	X
Abkürzungsverzeichnis.....	XI
1 Einleitung	1
1.1 Problemstellung und Motivation	1
1.2 Zielsetzung der Arbeit	2
1.3 Aufbau der Arbeit	3
2 Forschungsfragen.....	5
3 Theoretischer Hintergrund	6
3.1 Verkehrserhebungen	6
3.2 Verkehrszählungen	9
3.2.1 Grundsätzliche Verfahrensbeschreibung	9
3.2.2 Zählung des Kfz-Verkehrs.....	10
3.2.3 Zählung des Radverkehrs	11
3.2.4 Zählung des Fußgängerverkehrs	12
3.2.5 Datenaufbereitung, -dokumentation und -auswertung.....	12
3.3 Verkehrserfassungssysteme	15
3.3.1 Aufbau eines Verkehrserfassungssystems	15
3.3.2 Übersicht und Vergleichskriterien.....	15
3.3.3 Induktionsschleife.....	19
3.3.4 Magnetfelddetektor.....	21

3.3.5	Druckdetektor	22
3.3.6	Videoerfassung.....	24
3.3.7	Infrarotdetektor	26
3.3.8	Laserdetektor.....	28
3.3.9	Ultraschalldetektor	30
3.3.10	Radardetektor	31
3.3.11	Kombinationsdetektor	33
3.3.12	Wi-Fi-, Bluetooth-Detektoren	34
3.3.13	Weitere Erfassungssysteme	35
3.3.14	Zusammenfassende Bewertung	36
4	Konzept.....	38
4.1	Grundlagen.....	38
4.1.1	Erhebungsanlass	38
4.1.2	Erhebungsziele	38
4.1.3	Aussagegüte.....	38
4.1.4	Datenschutz.....	39
4.2	Erhebungsvorbereitung	40
4.2.1	Auswahl der Verkehrserfassungstechnologie	40
4.2.2	Übertragungstechnologie.....	43
4.2.3	Zeit und Dauer	46
4.2.4	Räumliche Abgrenzung.....	46
4.3	Vorbereitung der Datenerhebung	48
4.3.1	Montage & Inbetriebnahme.....	48
4.3.2	Witterungsschutz	49
4.4	Datenaufnahme und -übertragung.....	52
4.4.1	Datenaufnahme	52
4.4.2	Datenbereitstellung	53
4.5	Datenaufbereitung und -analyse	55
4.5.1	Datenaufbereitung	55

4.5.2	Datenvisualisierung	57
5	Umsetzung	59
5.1	Allgemein	59
5.1.1	Standort.....	59
5.1.2	Stromversorgung.....	60
5.1.3	Schutz vor Vandalismus.....	60
5.2	Hardware	61
5.2.1	IoT Octopus.....	61
5.2.2	STM.....	62
5.2.3	Sensoren.....	62
5.2.4	Gehäuse.....	66
5.2.5	Montage & Inbetriebnahme	69
5.3	Software.....	73
5.3.1	Firmware IoT-Octopus.....	73
5.3.2	Firmware STM.....	78
5.3.3	thingsHub	80
5.3.4	Datenvisualisierung	86
6	Ergebnisse	91
6.1	Durchgeführte Tests	91
6.2	Bewertung der Präzision	94
7	Zusammenfassung und Ausblick	98
Literaturverzeichnis.....		102
Anhang		105

Abbildungsverzeichnis

Abbildung 1: Gliederung der Verkehrsdatenerhebung	7
Abbildung 2: Tagesganglinie des Radverkehrs in Dresden	14
Abbildung 3: Grundprozesse eines Verkehrsdetektors	15
Abbildung 4: Induktionsschleife	19
Abbildung 5: Magnetfelddetektor	21
Abbildung 6: Druckschlauchsystem	22
Abbildung 7: Videoerfassung	24
Abbildung 8: Infrarotdetektor	26
Abbildung 9: Laserdetektoren	28
Abbildung 10: Ultraschalldetektor	30
Abbildung 11: Radardetektor	31
Abbildung 12: Kombinationsdetektor	33
Abbildung 13: Konzept Verkehrserfassungssysteme Adlershof	42
Abbildung 14: Datenübertragung per LoRaWAN	44
Abbildung 15: Architektur einer ttn-Anwendung	44
Abbildung 16: Architektur des thingsHub	45
Abbildung 17: Neuralgische Knotenpunkte für Verkehrsstromanalyse	47
Abbildung 18: Montage Ultraschalldetektor	48
Abbildung 19: Montage Ultraschalldetektor, Variante 2	48
Abbildung 20: Ausrichtung Ultraschallsensor	48
Abbildung 21: Montage Druckdetektor	49
Abbildung 22: Montage Radardetektor	49
Abbildung 23: Konzept Gehäuse Fußgängerzähler	50
Abbildung 24: Konzept Gehäuse Fahrradzähler	50
Abbildung 25: Konzept Gehäuse MIV-Zähler	51
Abbildung 26: Datenaufnahme Fußgängerzähler	52
Abbildung 27: Schema Datenbereitstellung	54
Abbildung 28: DLR-Befragung: Fahrtaufkommen pro Modus; Modal Split	55
Abbildung 29: Visualisierung Fahrtaufkommen	57
Abbildung 30: Visualisierung Fahraufkommen relativ	57
Abbildung 31: Visualisierung Modal Split	58
Abbildung 32: Visualisierung Akkustand	58
Abbildung 33: DLR Messbrücke	59
Abbildung 34: IoT Octopus Development Board	61
Abbildung 35: LoRaWAN-Modul von tnn am IoT Octopus	61

Abbildung 36: STM32L0 Discovery Kit.....	62
Abbildung 37: Utraschallsensor	63
Abbildung 38: Elektronik Fußgängerzähler	63
Abbildung 39: Drucksensor.....	64
Abbildung 40: Elektronik Fahrradzähler	64
Abbildung 41: Radarsensor	65
Abbildung 42: Elektronik MIV-Zähler	65
Abbildung 43: Wandgehäuse.....	66
Abbildung 44: Gehäuse Ultraschalldetektor Vorderseite	66
Abbildung 45: Gehäuse Ultraschalldetektor Stromkabelzugang	67
Abbildung 46: Gehäuse Druckdetektor	67
Abbildung 47: 3D-gedrucktes Gehäuse für den Grove-Radarsensor.....	68
Abbildung 48: Gehäuse für Elektronik des Radardetektors	68
Abbildung 49: Montage Fußgängerzähler	69
Abbildung 50: Montage Fußgängerzähler 2.....	69
Abbildung 51: Montage Fahrradzähler.....	70
Abbildung 52: Montage Fahrradzäher 2.....	70
Abbildung 53: Montage MIV-Zähler überkopf	71
Abbildung 54: Montage MIV-Zähler überkopf 2	71
Abbildung 55: Montage MIV-Zähler seitlich	72
Abbildung 56: ArduBlock-Umgebung	73
Abbildung 57: Sensoren-Block.....	74
Abbildung 58: ArduBlock für Fußgängerzähler	74
Abbildung 59: Timer-Flussdiagramm	76
Abbildung 60: STM Codestruktur.....	78
Abbildung 61: thingsHub-Integration.....	80
Abbildung 62: Device bei thingsHub anlegen	80
Abbildung 63: Projektinitialisierung	81
Abbildung 64: Driverliste aktualisieren.....	84
Abbildung 65: Driver zuweisen	84
Abbildung 66: Datentabelle erstellen	85
Abbildung 68: REST-API Datenexport	85
Abbildung 67: REST-API Response	85
Abbildung 69: Export CSV	86
Abbildung 70: Dashboard Verkehrsstromanalyse	86
Abbildung 71: Graph-Panel.....	87
Abbildung 72: Graph-Panel Datenabfrage	87

Abbildung 73. Graph-Panel Schwellenwerte	88
Abbildung 74: Gauge-Panel	88
Abbildung 75: Gauge-Panel Datenabfrage.....	88
Abbildung 76: Gauge-Panel Schwellenwerte	89
Abbildung 77: Bar Gauge-Panel Datenabfrage	89
Abbildung 78: Bar Gauge-Panel Schwellenwerte.....	90
Abbildung 79: Rohwerte des Fahrradzählers	92
Abbildung 80: MIV-Tagesganglinie Radardetektor vs. Induktionsschleife, Montage überkopf.....	93
Abbildung 81: MIV-Tagesganglinie Radardetektor vs. Induktionsschleife, Montage seitlich.....	94
Abbildung 82: MIV-Zähler prozentuale Abweichung (Montage überkopf)	95
Abbildung 83: MIV-Zähler prozentuale Abweichung (Montage seitlich)	95
Abbildung 84: MIV-Zähler Anstiegsvergleich (Montage überkopf).....	96
Abbildung 85: MIV-Zähler Anstiegsvergleich (Montage seitlich).....	96

Tabellenverzeichnis

Tabelle 1: Einteilung der Kfz in Grundklassen	10
Tabelle 2: Erläuterung zu verkehrstechnischen Kenngrößen	16
Tabelle 3: Kriterien für Detektorvergleich.....	17
Tabelle 4: Übersicht Verkehrserfassungssysteme	18
Tabelle 5: Erfassung von Verkehrsteilnehmern	36
Tabelle 6: Zuordnung von verkehrstechnischen Kenngrößen zu Verkehrserfassungssystemen.....	36
Tabelle 7: Vergleich der Verkehrserfassungssysteme	37
Tabelle 8: Gewichtung der Auswahlkriterien.....	41
Tabelle 9: Ergebnisse der Nutzwertanalyse.....	42
Tabelle 10: Neuralgische Knotenpunkte	47
Tabelle 11: Eigenschaften von Grove Ultrasonic Ranger	63
Tabelle 12: Eigenschaften von Grove Pressure Sensor	64
Tabelle 13: Eigenschaften von Grove Doppler Radar.....	65
Tabelle 14: Testergebnisse Fußgängerzähler	91

Abkürzungsverzeichnis

3D	dreidimensional
CW	Continious Wave
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DTV	durchschnittliche tägliche Verkehrsstärke
DTV _w	durchschnittliche werktägliche Verkehrsstärke
FGSV	Forschungsgesellschaft für das Straßen- und Verkehrswesen
FMCW	Frequency Modulated Continious Wave
MAC	Media-Access-Control-Adresse
MCU	Multipoint Control Unit
MIV	motorisierter Individualverkehr
MSV	maßgebliche stündliche Verkehrsstärke
ÖPNV	öffentlicher Personennahverkehr
PIR	Passiv-Infrarot-Sensor
ToF	Time-of-Flight

1 Einleitung

1.1 Problemstellung und Motivation

Der Standort Berlin Adlershof gehört zu einem der größten Wissenschafts- und Technologieparks Deutschlands und zum größten Medienstandort Berlins. Circa 22.000 (Stand: 2019) Personen arbeiten in Unternehmen und Einrichtungen, die im Technologiepark Adlershof ansässig sind. Sie gelangen zu Fuß, mit dem Fahrrad, mit dem Auto oder mit dem öffentlichen Personennahverkehr (ÖPNV) zum Standort. Außerdem befindet sich im Zentrum des Wissenschafts- und Technologieparks Universitäten, die von ca. 6.500 Studierenden besucht werden. (*Berlin Adlershof: Daten und Fakten 2019*)

Perspektivisch wird der Standort weiterhin wachsen und bis zum Jahr 2030 wird eine Verdopplung der Anzahl der Mitarbeiter erwartet. Weiterhin gehen die Prognosen von circa 10.000 Studenten und 11.000 Anwohnern aus. (Krajzewicz et al. 2018, S. 6)

Die Infrastruktur sowie die Mobilität müssen dabei mitwachsen, sodass weiterhin eine gute Anbindung gewährleistet werden kann. Aus diesen Gründen hat die WISTA Management GmbH (WISTA) und Ihre Tochterfirma WISTA.Plan GmbH das Deutsche Zentrum für Luft- und Raumfahrt (DLR) beauftragt, eine Studie zur Verkehrsprognose (Zieljahr 2030) durchzuführen. Ergänzend steht die Verkehrsprognose 2030 der LK Argus zur Verfügung, welche durch WISTA.Plan beauftragt wurde.

Die WISTA und das DLR haben im November 2018 das aktuelle Mobilitätsverhalten im Standort durch eine Umfrage untersucht. Dabei wurden die 1.200 Studienteilnehmer aller Nutzergruppen gefragt, welche Verkehrsmittel sie nutzen, um zum Standort Adlershof zu kommen sowie zu welchen Uhrzeiten dieses geschieht. Die Ergebnisse dieser Umfrage nutzte das DLR für die Validierung der darauffolgenden Simulationen der Nachfrage und des Verkehrszustands. Die Simulationsergebnisse zeigen, dass der Technologiepark Adlershof keine außerordentlichen qualitativen Unterschiede im Verkehrsverhalten erwarten soll. Die Anteile der einzelnen Verkehrsteilnehmer bleiben somit nicht verändert. Im Gegensatz dazu ist quantitativ eine signifikante Steigerung der Standortbesucher aufgezeigt und somit eine erhebliche Erhöhung des Verkehrsaufkommens vorausgesagt, die zu einer Verkehrsnetzüberlastung führen kann. Die hochgerechnete Anzahl an Verkehrsteilnehmer war so hoch, dass sie nicht vollständig in die Simulation eingespielt werden konnten. Dabei würden sich die Verlustzeiten verdoppeln. Laut DLR würde es in der Realität nicht dazu kommen, weil die Besucher sich für Alternativen entscheiden. (Krajzewicz et al. 2018)

LK Argus GmbH hat im Jahr 2017 eine Studie durchgeführt, um die Leistungsfähigkeit der relevantesten Knotenpunkte für das Jahr 2030 vorherzusagen. Dabei wurden die Bauvorhaben der WISTA.Plan GmbH berücksichtigt sowie die Einflüsse, die daraus resultieren. Die Ergebnisse der Studie stellen, analog zur DLR-Studie, eine stark zunehmende Verkehrsmengen dar. Dabei wäre laut Prognose die Grenze der Leistungsfähigkeit einiger Knotenpunkte im Jahr 2030 erreicht. (Lotz und Windmüller 2018)

Um diese Auswirkung in der Realität zu vermeiden, wurden in Jahren 2018 und 2019 in Zusammenarbeit mit LK Argus GmbH Maßnahmen für jeden kritischen Knotenpunkt entwickelt, deren Realisierung die zukünftige hochgerechnete Auslastung um 28% reduzieren kann. (Reimann und Stolz 2019)

Die WISTA Management GmbH arbeitet ständig an einer Optimierung der Verkehrsinfrastruktur. Um die Wirksamkeit der eingeführten Maßnahmen zu ermitteln, soll im Technologiepark künftig eine Messmethode entwickelt werden, mithilfe deren die Verkehrsdaten in Echtzeit erfasst werden.

1.2 Zielsetzung der Arbeit

Allgemeines Ziel der vorliegenden Arbeit ist es, aktuelle Technologien der IoT-basierten Verkehrsstromanalyse für die Ermittlung des Mobilitätsbedarfs zu untersuchen und zu bewerten. Der Schwerpunkt soll auf der Erfassung des Fahrtaukommens in Abhängigkeit vom genutzten Verkehrsmittel liegen. Dabei soll es zwischen motorisiertem Individualverkehr (MIV), Radfahrer und Fußgänger unterschieden werden. Die Messungen sollen kontinuierlich durchführbar sein, damit die Möglichkeit besteht, eine Verteilung über den jeweils gewünschten Zeitraum (Tag, Woche, Monat, Jahr) zu erhalten. Im Einzelnen sollen folgende Ziele verfolgt werden.

Ziele im Kontext der theoretischen Analyse:

- Recherche der aktuellen Methoden der Verkehrsstromanalyse mit Realisierbarkeit im IoT-Kontext
- Herleitung von Kennzahlen sowie qualitativen Kriterien an Technologien der Verkehrsstromanalyse
- Bewertung und Vergleich ausgewählter Technologien der Verkehrserfassung

Mithilfe der theoretisch gewonnenen Ergebnisse soll eine Übersicht zu verfügbaren Technologien geschafft sowie deren Stärken und Schwächen identifiziert werden. Anhand der Bewertung und des Vergleichs dieser Technologien soll es künftig möglich sein, die passende Methode in Abhängigkeit des Anwendungsfalls auszuwählen.

Ziele im Kontext des praktischen Anteils:

- Definition und Priorisierung der Anforderungen an die Technologien in Zusammenarbeit mit der WISTA Management GmbH
- Auswahl der geeigneten Technologien für den Einsatz im Technologiepark Adlershof
- Erarbeitung eines Konzeptes zur Erfassung der Verkehrsströme folgender Verkehrsteilnehmer: MIV, Radfahrer, Fußgänger
- Prototypische Umsetzung des Konzeptes an festgelegten neuralgischen Verkehrsknotenpunkten inkl. Hardware und deren Gehäuse, Integration ins vorhandene LoRaWAN-Netz sowie Visualisierung der erfassten Daten
- Validierung der durchgeföhrten Verkehrsstromanalyse

Das Ergebnis des praktischen Teils soll eine solide Datenbasis schaffen, um Entscheidungen für die Optimierung der Mobilität im Technologiepark Adlershof zu treffen. Aufgrund dieser Daten soll es möglich sein, Defizite im Mobilitätsbereich zu erkennen sowie im Standort umgesetzte Maßnahmen auf ihre Wirksamkeit zu überprüfen. Außerdem soll die entwickelte Technologie auf andere Standorte der WISTA mit vergleichbaren Herausforderungen und Zielen übertragbar sein.

1.3 Aufbau der Arbeit

Zum Erreichen der Ziele wird die Arbeit in 7 Kapitel gegliedert, deren Inhalte im Folgenden zusammenfassend vorgestellt werden.

Kapitel 2 beinhaltet die Forschungsfragen, welche die vorliegende Masterarbeit zu beantworten bezweckt.

In Kapitel 3 wird der theoretische Hintergrund des Gesamtkomplexes Verkehrsstromanalyse erläutert. Hierfür wird zunächst die Begrifflichkeit und Arten der Verkehrserhebung definiert. Als Nächstes wird detailliert auf Verkehrszählung eingegangen, da sie den Schwerpunkt dieser Masterarbeit bildet. Dabei werden die Besonderheiten der Verkehrsgruppen Fußgänger, Fahrradfahrer und motorisierten Fahrzeuge im Kontext der Verkehrszählung dargestellt. Im nächsten wesentlichen Teil des theoretischen Hintergrunds stehen die aktuellen Technologien, die der automatischen Verkehrserhebung dienen, im Fokus. Zunächst wird deren Aufbau beschrieben und Vergleichskriterien definiert, anhand deren sie später bewertet werden. Es werden Eigenschaften und Funktionsweise, Art der Installation, Vor- und Nachteile sowie Erfassungsmöglichkeit von Verkehrsteilnehmern von 10

Verkehrserfassungssystemen beleuchtet. Ein komplexer Vergleich dieser Technologien anhand bereits definierter Kriterien rundet das Kapitel ab.

Kapitel 4 beinhaltet das Konzept zur Verkehrsstromanalyse, welches alle notwendigen Schritte zur Auswahl, Entwicklung und Einführung von Verkehrserfassungssystemen erläutert. Die Auswahl der geeigneten Technologien erfolgt mithilfe der Nutzwertanalyse, die in Zusammenarbeit mit WISTA Management GmbH durchgeführt wird. Daraufhin folgen weitere Empfehlungen zur Erhebungsvorbereitung wie Übertragungstechnologie sowie zeitliche und räumliche Abgrenzung der Datenerhebung. Als Nächstes werden Anweisungen zur Montage sowie Witterungsschutz gegeben. Abschließend wird das Datenkonzept mit Prozessen der Datenaufnahme, -bereitstellung, -aufbereitung und -visualisierung erläutert.

Die prototypische Umsetzung dieses Konzeptes wird in Kapitel 5 beschrieben. Zuerst wird auf allgemeine Aspekte wie Standort, Stromversorgung und Schutz vor Vandalismus eingegangen. Darauffolgend werden die Hardware- und Softwarekomponenten beschrieben. Dabei veranschaulichen zahlreiche Abbildungen die fertigen Detektoren, Hardwaremontage am Messstandort sowie Datenvisualisierungen.

In Kapitel 6 wird erläutert, wie die IoT-Geräte getestet wurden. Daraufhin werden die Testszenarien definiert und deren Ergebnisse vorgestellt. Anhand der Testergebnisse wird die Präzision ermittelt. Die Datenauswertungen aus dem Visualisierungstool runden das Ergebniskapitel ab.

Das letzte Kapitel beinhaltet Zusammenfassung der im Laufe dieser Arbeit gewonnenen Ergebnisse und Erkenntnisse. Abschließend wird einen Ausblick auf potenzielle Erweiterungen der eingeführten Technologien und deren Einsatz zur Ermittlung des Mobilitätsbedarfs gegeben.

2 Forschungsfragen

Diese Abschlussarbeit bezweckt die Beantwortung folgender Forschungsfragen:

- Welche technologischen Möglichkeiten gibt es, Verkehrsströme in Echtzeit automatisiert zu erfassen?
- Welche Möglichkeiten gibt es, unterschiedliche Verkehrsarten automatisiert voneinander zu unterscheiden?
- Wie kann diese Technologie am Standort Adlershof und dessen LoRaWAN-Netz integriert werden?
- Wie präzise sind die erfassten Daten sowie Ergebnisse der Klassifizierung?

3 Theoretischer Hintergrund

3.1 Verkehrserhebungen

Die Verkehrsdatenerfassung ist kein Selbstzweck, sondern dient der Lösung mehrerer Fragestellungen. Zum einen unterstützt sie die kurzfristige Verkehrsbeeinflussung und liefert aktuelle Zustandsinformationen über verschiedene Kommunikationsmittel. Zum anderen stellt sie wichtige statistische Daten bereit, die für die Planung und Bewertung von Verkehrsmanagementmaßnahmen notwendig sind, und unterstützt somit die langfristige Verkehrsplanung. (FGSV 2019, S. 9)

FGSV (2012a, S. 29) definiert Verkehrserhebung als „Gewinnung von Daten eines bestehenden Verkehrszustands“. Außer Anzahl der Verkehrsteilnehmer, Geschwindigkeit, Aufenthaltsdauer (quantitative Merkmale) können verschiedene qualitativen Merkmale wie Zweck der Fahrt, benutztes Verkehrsmittel u.a. erhoben werden. Diese Daten bilden eine Grundlage für ein effektives Verkehrsmanagement, das sich auf eine optimale gegenseitige Abstimmung der Verkehrsnachfrage und dem Angebot an Verkehrssystemen richtet (FGSV 2012a, S. 122).

Vor der Erhebung soll das konkrete *Untersuchungsziel* formuliert werden. Dieses Ziel kann aus Gebieten wie Verkehrsstatistik, Verkehrsmanagement, Marktforschung etc. stammen und sich auf Erfassung unterschiedlichsten *Kenngrößen* richten, die ebenso vorab ausgewählt werden sollen. Außerdem muss die erforderliche *Genauigkeit* der benötigten Daten definiert werden. Als nächstes soll die *Größe des Untersuchungsgebiets* bestimmt werden. Anschließend wird die Entscheidung über die *Erhebungsmethode* getroffen. Die etablierten Erhebungsmethoden werden im Folgenden beschrieben. Zu definieren sind die *Dauer* und *Zeitpunkt der Erhebungen* sowie *Länge der Zeitintervalle*. (FGSV 2012b, S. 9)

Grundsätzlich wird im Verkehrswesen zwischen verkehrstechnischen Erhebungen (*Zählungen*) und verkehrsverhaltensbezogenen Erhebungen (*Beobachtungen* und *Befragungen*) unterschieden (Steierwald et al. 2005, S. 80). Im Folgenden werden die jeweiligen Gruppen der Verkehrsdatenerhebung erläutert.

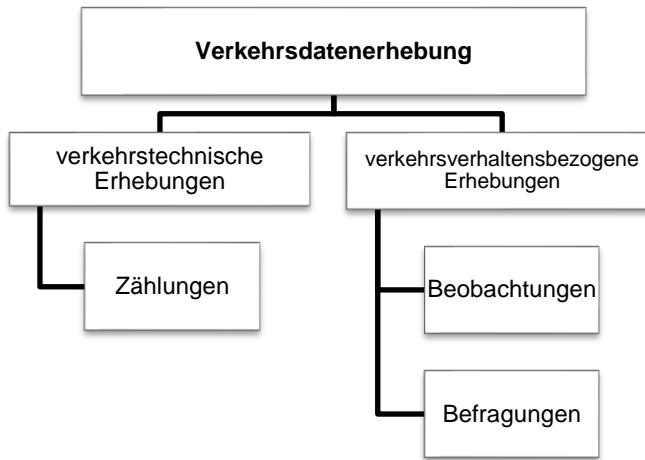


Abbildung 1: Gliederung der Verkehrsdatenerhebung

Durch Verkehrszählungen werden „[...] Ortsveränderungen von Personen und Gütern auf den Verkehrswegen eines Untersuchungsraums erfasst“ (FGSV 2012b, S. 13). Die Verkehrszählungen werden in *Objekt-, Querschnittszählungen und Stromerhebungen* aufgeteilt. Bei Objektzählungen werden Personen und/oder Fahrzeuge erfasst, die sich während eines definierten Zeitraums in einem abgegrenzten Bereich befinden, z.B. Fußgänger auf einem Fußgängerabschnitt oder parkende Fahrzeuge. *Querschnittszählungen* konzentrieren sich auf Zählungen von Personen und/oder Fahrzeugen, die während eines Zeitabschnittes einen Zählquerschnitt passieren. *Stromerhebungen* liefern Informationen über die Verteilung der Verkehrsströme an einem Knotenpunkt oder in einem größeren Erhebungsgebiet. Dabei wird eine nach Geh- und/oder Fahrtrichtung getrennte Querschnittszählung durchgeführt bzw. werden „[...] an allen relevanten Zu- und Ausfahrtstraßen alle ein- und ausströmende Verkehre erfasst“ (Steierwald et al. 2005, S. 83).

Beobachtungen werden durchgeführt, um äußere Merkmale sowie sichtbare Verhaltensweisen im Straßenraum zu erfassen. Dabei können Informationen über Fahrtgründe und -ziele der Verkehrsteilnehmer gesammelt werden. Die Datenerfassung erfolgt nichtkommunikativ, d.h. ohne Mitwirkung der Zielpersonen. Die Beobachtungsverfahren beschränken sich meist auf die visuelle Erfassung durch Bild- und Videoaufnahmen. Diese Verfahren finden Anwendung in Verkehrssicherheit, um die situationsbezogene Einhaltung der Verkehrsregeln zu überwachen, Konflikte sowie Abstand-, Geschwindigkeits- und Beschleunigungsverhalten zu analysieren. Außerdem werden Beobachtungen zur Analyse komplexer Verkehrssituationen sowie Verkehrsverhaltens eingesetzt. (FGSV 2012b, S. 56)

Befragungen werden kommunikativ durchgeführt und liefern Mobilitätskennziefern sowie personenbezogene Verhaltensdaten. Da Befragungen zu verkehrsverhaltensbezogenen Erhebungen gehören, können damit außer eigentlichen Verkehrszahlen Rahmenbedingungen und Entscheidungshintergründe erfasst

werden. (FGSV 2012b, S. 63) Es werden mehrere Befragungsarten nach dem Ort bzw. dem Adressaten unterschieden. Darunter fallen die *Befragungen im Verkehrsnetz*, die an ausgewählten Verkehrspunkten (z.B. auf Parkplätzen, in öffentlichen Verkehrsmitteln) mit dem Ziel durchgeführt werden, Quell- und Zielangaben sowie Fahrtzweck zu erfassen. Bei *Befragungen im Haushalt* werden zufällig ausgewählte Personen bzw. Haushalte mithilfe standardisierter Fragebogen mündlich, schriftlich oder telefonisch über ihre Verkehrsaktivitäten befragt. *Befragungen am Aktivitätsort* werden als Interviews am Ort einer Aktivität, z.B. im Einkaufszentrum, Spielplatz usw. durchgeführt. Das Ziel dabei kann Ermittlung der Gesamtzahl aller Personen am Aktivitätsort, Identifikation des Aufenthaltsziels, Ermittlung des durchschnittlichen Besucherdauer u.a. sein. Diese Befragungen schaffen die Datenbasis für komplexe Fragestellungen, wie z.B. Freizeit- oder Kundenverhalten sowie Akzeptanz von eingeführten Maßnahmen. (FGSV 2012b, S. 77)

Die vorliegende Arbeit beschäftigt sich ausschließlich mit automatischen Verkehrszählungen. Deren Anwendungsgebiete sowie grundsätzliche Verfahrensbeschreibung werden im Folgenden beschrieben.

3.2 Verkehrszählungen

3.2.1 Grundsätzliche Verfahrensbeschreibung

Die Verkehrszählungen liefern Informationen über räumliche und zeitliche Verteilung der Verkehrsströme an Knotenpunkten und Querschnitten. Durch Zählungen können keine Quelle-Ziel-Beziehungen bzw. Wegezwecken ermittelt werden. (FGSV 2012b, S. 25)

Der erste Schritt in Vorbereitung auf Verkehrszählungen ist die Definition bzw. Differenzierung der Raumeinheiten. Diese hängen von Erhebungszielen ab und können unterschiedliche Fläche haben. Beispiele für solche Raumeinheiten sind Haltestellen bei Netzuntersuchungen, Verkehrszellen bei Untersuchungen zur Verkehrsberuhigung sowie Grundstücke bzw. Parkplätze bei Parkraumuntersuchungen. (FGSV 2012b, S. 26)

Als Nächstes werden die Zählzeiten definiert, die sich ebenso nach Erhebungszielen richten. Dabei wird versucht, „[...] solche Monate, Wochentage und Tageszeiten zu nutzen, die eine Erfassung eines repräsentativen Abbildes des Verkehrsgeschehens gewährleisten“ (Steierwald et al. 2005, S. 85). Günstige Erhebungszeiten liegen in den Zeiträumen von März bis Oktober in den Wochen, die keine besonderen Ereignisse wie Ferien und Feiertage beinhalten. An Wochentagen zwischen Montag und Donnerstag ähneln sich die Erhebungszahlen in der Vergangenheit am meisten. Als Tageszeitintervalle haben sich folgende Zeitspannen etabliert:

- Tagesverkehr: 6:00 - 22:00 Uhr
- Vormittagsverkehr: 6:00 - 10:00 Uhr
- Nachmittagsverkehr: 15:00 - 19:00 Uhr
- Nachtverkehr: 22:00 - 6:00 Uhr

Eine der häufigsten Anwendungsfälle bei Verkehrszählungen ist die Ermittlung der Verkehrsbelastung. Die Verkehrsbelastung kann sich auf unterschiedliche Zählzeiten richten. So können bei entsprechend ausgewählter Zeiteinheit von einer Stunde, einem Tag oder einem Werktag *maßgebliche stündliche Verkehrsstärke* (MSV), *durchschnittliche tägliche Verkehrsstärke* (DTV) sowie *durchschnittliche werktägliche Verkehrsstärke* (DTV_w) ermittelt werden. Sollte anhand der Verkehrszählungen eine Hochrechnung stattfinden, müssen bei der Wahl des Zeitintervalls je nach Hochrechnungsverfahren mehrere Faktoren berücksichtigt werden. So fordern viele Hochrechnungsverfahren nicht nur Daten der Durchschnittstage, sondern auch Ergebnisse der Zählungen von Wochenend- sowie Feiertagen. (Steierwald et al. 2005, S. 85; FGSV 2012b, S. 26)

Zur Gewährleistung der guten Erhebungsqualität sind vor Beginn der automatischen Zählungen ausgiebige Gerätetests sowie während der Zählung regelmäßige Kontrollen durchzuführen. Für die korrekte Zuordnung und Auswertung der Zählergebnisse sollen während der Messung Informationen wie Ort, Erhebungszeit und andere Besonderheiten, die für die Interpretation der Ergebnisse essenziell sind, erfasst werden. Diese Metadaten sollen in einer Auswertedatei bzw. in einer Datenbank gespeichert werden. Vor Beginn der Auswertung ist eine Plausibilitätsprüfung durchzuführen, indem Fehler ausgeschlossen und ggf. Daten korrigiert werden. Die korrigierten Daten sind mit einer entsprechenden Kennzeichnung in der Auswertedatei bzw. Datenbank zu versehen. Nach der Datenauswertung müssen die Ergebnisse auf die Erfüllung der Erhebungsziele und anderer Anforderungen geprüft werden. (FGSV 2012b, S. 28–29)

3.2.2 Zählung des Kfz-Verkehrs

Zählungen des Kfz-Verkehrs werden häufig in Kombination mit Fahrzeugklassifizierungen durchgeführt. Die BASt (2012) gliedert motorisierte Fahrzeuge in neun Grundklassen (siehe Tabelle 1).

Tabelle 1: Einteilung der Kfz in Grundklassen

Quelle: in Anlehnung an BASt 2012, S. 141

Grundklasse	Kurzbezeichnung	Erläuterung
Motorräder	Krad	Motorräder ggf. mit Beiwagen, aber keine Fahrräder, keine Mofas
Pkw	Pkw	Pkw vom Kleinwagen bis zur Großraum-Limousine einschließlich der Offroad-Fahrzeuge
Lieferwagen	Lfw	Lieferwagen \leq 3,5 t zul. GG (Gesamtgewicht)
Pkw mit Anhänger	PkwA	Pkw und Lfw \leq 3,5 t zul. GG mit Anhänger
Lkw ohne Anhänger	Lkw	Lkw $>$ 3,5 t zul. GG ohne Anhänger
Lkw mit Anhänger	LkwA	Lkw $>$ 3,5 t zul. GG mit Anhänger
Sattelkraftfahrzeuge	Sattel-Kfz	Alle Sattelkraftfahrzeuge
Busse	Bus	Fahrzeuge mit mehr als 9 Sitzplätzen zur Personenbeförderung
Nicht klassifizierbare Fahrzeuge	Nk Kfz	Alle Kfz, bei denen die Fahrzeugart nicht bestimmbar war oder keiner der Klassen gehört

Das Kfz-Verkehr zeichnet sich durch eine asymmetrische Verteilung in Abhängigkeit der Fahrtrichtung und der Fahrtzeit. Aus diesem Grund wird eine richtungsgtrennte Erfassung bevorzugt, die eine ausreichende Aussagekraft und höhere

Messgenauigkeit liefert. Vor allem die Geschwindigkeiten können so genauer erfasst werden. Die Ausnahme sind die schmalen oder schwach belastete Straßen, auf denen eine gemeinsame Erfassung für beide Fahrtrichtungen ausreicht. (FGSV 2010, S. 9)

Die zeitliche Differenzierung der Fahrzeugzahlen und -merkmale hängt vom Einsatzzweck ab und kann von „[...] Erfassung einzelnen Fahrzeuge mit genauer zeitlicher Zuordnung des Einzelfahrzeuges bis hin zu deren zeitlichen Aggregierung zu Minuten-, Stunden- oder Tageswerten“ (FGSV 2010, S. 9) variieren. In der Regel wird in der Verkehrsplanung das Stundenintervall als Grundlage für die Messungen gewählt. Bei Knotenpunktzählungen, insbesondere bei Erfassung von Geschwindigkeiten und Zeitlücken werden kürzere Intervalle (5-, 10- oder 15-Minuten) bevorzugt. Die technischen Eigenschaften der Erfassungsgeräte (Speicherkapazität, Einsatzdauer usw.) sollen auf die Auswahl des Zeitintervalls ausgerichtet werden. (FGSV 2010, S. 9)

3.2.3 Zählung des Radverkehrs

Der Radverkehr weist im Vergleich zum motorisierten Verkehr eigene Bewegungsmuster auf und zeichnet sich durch eine höhere Flexibilität aus. Außerdem sind die Radverkehrsteilnehmer empfindlicher gegen Witterungseinflüsse und agieren in vielen Fällen spontan und unvorhersehbar. Das Verhalten der Radfahrer hängt stark von der vorhandenen Infrastruktur ab. Die Radverkehrsanlagen können entweder auf der Fahrbahn (Radfahrstreifen) oder auf Gehwegniveau aufgebaut sein. Da oft eine Kombination von beiden möglich ist, sind die Radfahrer „[...] legal oder gesetzwidrig, im gesamten Straßenraum zu finden“ (FGSV 2010, S. 47). Sobald die Radverkehrsanlagen abwesend sind, fahren die Radfahrer zusammen mit MIV oder Fußgängern im Mischverkehr. Aufgrund der bauartbedingten Besonderheiten können Fahrräder oft von anderen sich bewegenden Objekten schwer unterschieden werden. Besonders die Fahrräder mit geringer Breite, geringem Gewicht und geringem Metallanteil sind problematisch zu erfassen. Aus diesen Gründen ist die automatische Erfassung des Radverkehrs mit Herausforderungen verbunden. Vor Beginn der Erfassung sind die radverkehrsspezifischen Besonderheiten sowie örtlichen Umstände zu analysieren. (FGSV 2010, S. 46–47; Zweibrücken et al. 2005, S. 23–33)

Bei der Erfassung des Radverkehrs sind in der Regel zwei Kennwerte von Bedeutung: *Menge je Zeiteinheit* und *Fahrtrichtung*. Die Geschwindigkeitsmessung ist dabei nicht relevant. Die automatische Zählung ist mithilfe unterschiedlicher Detektoren möglich. Die Detektoren sowie deren Eignung für Radverkehrszählungen werden im Abschnitt 3.3 beschrieben. Viele Geräte versprechen eine gleichzeitige Erhebung des Kfz- und Radverkehrs. In der Realität ist solche Erfassung wegen der sehr unterschiedlich

ausgeprägten fahrzeugspezifischen Merkmale problematisch. Für die Zählung des Radverkehrs ist eine andere Geräteanordnung sowie feinere Einstellungen der Geräteempfindlichkeit erforderlich. Aus diesem Grund wird empfohlen „[...] den Kfz-Verkehr und Radverkehr mit zwei Systemen getrennt voneinander zu erfassen“ (FGSV 2010, S. 47). Die Zählung des Radverkehrs soll dabei auf Radverkehrsanlagen (z.B. Radweg, Radfahrstreifen) stattfinden.

3.2.4 Zählung des Fußgängerverkehrs

Der Fußgängerverkehr unterscheidet sich vom motorisierten Verkehr, ähnlich wie der Radverkehr, durch größere Flexibilität und erhöhte Witterungsabhängigkeit. Diese Verkehrsgruppe neigt zu spontanen Richtungswechseln sowie zu häufiger Nichteinhaltung der Verkehrsregeln. „Zählungen des Fußgängerverkehrs dienen vorrangig der rein mengenmäßigen Erfassung der Fußgänger an einem Querschnitt [...]“ (FGSV 2012b, S. 25). Da der Fußgängerverkehr stark vom Witterungsbedingungen abhängt, sind die Wetterdaten während der Fußgängerzählung gemeinsam mit Fußgängerzahlen zu erfassen. Die automatischen Zählverfahren ermitteln die Anzahl der Personen, die sich während eines bestimmten Zeitraums in einem definierten Querschnitt befanden bzw. den definierten Querschnitt überquert haben. Als Messstandort eignen sich für dichten Fußgängerverkehr die Treppenanlagen, da die Passierenden sich dort in übersichtlicher Form teilen und besser detektiert werden können. Außerdem kann der Zählquerschnitt während des Erhebungszeitraums durch künstliche Hindernisse in mehrere Abschnitte geteilt werden. (FGSV 2012b, S. 30)

3.2.5 Datenaufbereitung, -dokumentation und -auswertung

Die Datenaufbereitung basiert auf dem Zweck der Verkehrsdatenerhebung. In vielen Fällen ist es sinnvoll nicht nur den aktuellen Zustand zu erfassen, sondern Erhebungen über längeren Zeitraum durchzuführen. Statistisch gesehen ist ein Ergebnis umso repräsentativer, je länger und häufiger eine Erhebung stattfindet. Die über einen langen Zeitraum erhobenen Daten können später für Erstellung von Vergleichen benutzt werden. Für die aussagekräftige Datenauswertung ist neben der Datenqualität, die Qualität der Aufbereitung und Dokumentation entscheidend. Das bedeutet, dass nicht nur reine Erhebungsdaten (Anzahl, Geschwindigkeit usw.), sondern auch Erhebungsumstände – Metadaten wie Datum, Ort usw. zu erfassen sind. Diese Daten sind in einem Datensatz zu speichern, sodass langfristig Zuordnung und Vergleichbarkeit sichergestellt werden. (FGSV 2012b, S. 45)

Vor Beginn der Zählung ist festzulegen, wie die Daten verarbeitet werden. Diese Entscheidung muss für die spätere Nachvollziehbarkeit dokumentiert werden. Als nächstes ist die Entscheidung über eine passende Übertragungstechnologie zu treffen. Die Daten können per Kabel oder per Datenpunkt übertragen werden. Dabei muss der Datenverlust vermieden werden z.B. durch ein im Detektor eingebauten Ringspeicher. Er sorgt dafür, dass die Daten so lange lokal gespeichert bleiben, bis sie durch neu erfasste Daten überschrieben werden. (FGSV 2010, S. 52)

Um Datenübertragungsfehler sowie mögliche Detektorstörungen und -ausfälle zu erkennen, ist die Durchführung der Plausibilitätsprüfungen notwendig. Dabei ist außerdem auf Unregelmäßigkeiten des Verkehrsablaufs (Bauarbeiten, Fahrstreifensperrung usw.) zu achten, da sie einen Einfluss auf die Datenplausibilität haben können. (FGSV 2010, S. 52)

Die Fehlerquellen sind bei unterschiedlichen Zählverfahren jeweils anders. Aus diesem Grund sind die Plausibilitätsprüfungen unterschiedlich komplex. Als Erstens wird in der Regel eine automatische Vollständigkeitsprüfung durchgeführt, um fehlende Datensätze auszuschließen. Die inhaltlichen Prüfungen können nur bedingt automatisch erfolgen. Dabei wird meist die Meinung von Experten gefragt. (FGSV 2012b, S. 45)

Bei automatischen Zählverfahren können unterschiedlich große Fehler auftreten. Aus diesem Grund ist es sinnvoll zu definieren, welche Fehlertoleranz akzeptiert werden kann. Dabei ist zu beachten, dass das tägliche Verkehrsaufkommen natürlichen Schwankungen unterliegt. Bei Zählungen im Fußgänger- und Radverkehr werden Vollständigkeit der Datenerhebung sowie komplettete Erfassung während des Erhebungszeitraums (Geräteausfall, Unfälle usw.) automatisch kontrolliert. Außerdem findet zur Plausibilitätsprüfung einen Abgleich mit Nachbarzählstellen statt. (FGSV 2012b, S. 45)

Bei Zählungen des Kfz-Verkehrs sind zusätzlich die Verkehrsstärke je Richtung auszuwerten. Stark unsymmetrische oder unfeasible Verkehrsstärke deutet auf Erhebungsfehler hin. Sie werden durch falsche Zählungen an einer oder mehreren Zählstellen, systematische Fehler durch falschen Gerätetestandort bzw. fehlerhafte Kalibrierung der Messgeräte verursacht. Bei Kfz-Klassifizierungen ist der Schwerverkehrsanteil zu überprüfen. Soll dieser zu hoch bzw. zu niedrig sein, ist von einer falschen Einteilung der Fahrzeugklassen auszugehen. Dies kann am Auswerteprogramm, aber auch an den Zählgeräten liegen. (FGSV 2012b, S. 46)

Nach Aufbereitung der Messdaten sowie Plausibilitätsprüfungen liegen die Messergebnisse als Einzeldaten oder aggregiert zu definierten Intervallwerten vor. Die Auswerteprogramme analysieren die Ergebnisse und ermitteln Kennwerte, die sie

tabellarisch zusammenführen und meist in Form von Ganglinien darstellen. (FGSV 2010, S. 53) (siehe Abbildung 2). Sie beinhaltet die stündliche Verkehrsstärke des Radverkehrs in Form von Absolutwerten an drei Standorten in Dresden.

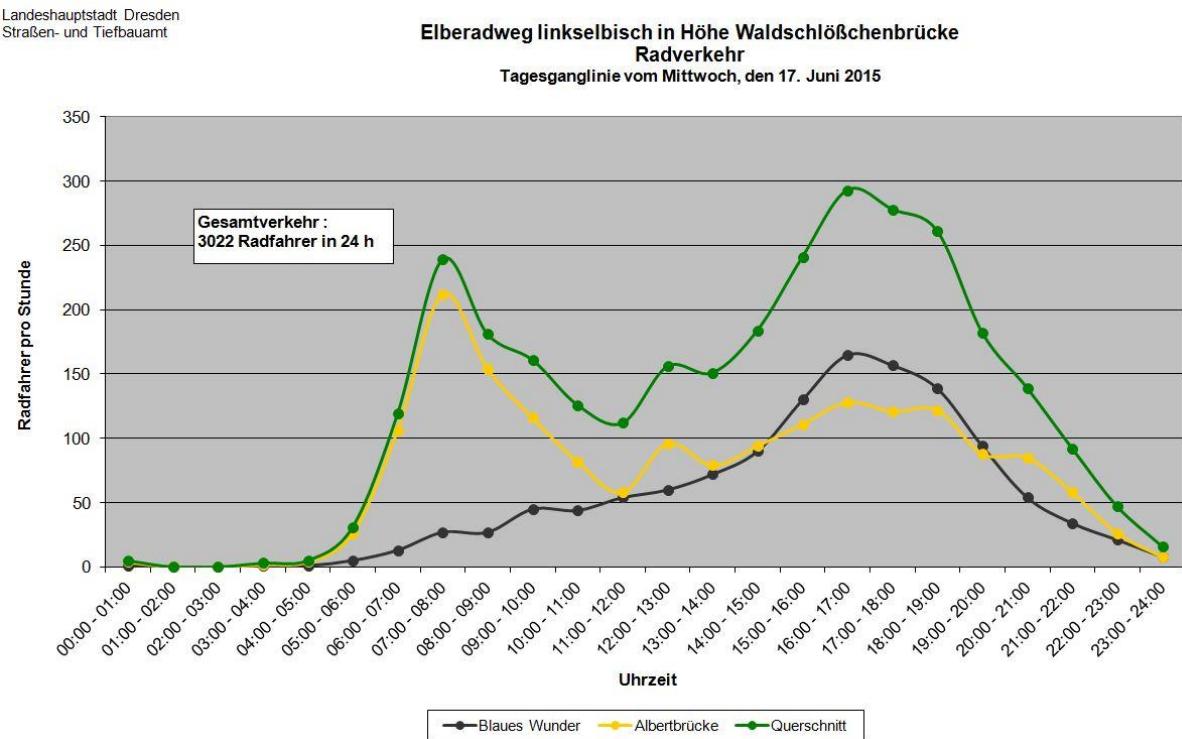


Abbildung 2: Tagesganglinie des Radverkehrs in Dresden

Quelle: Landeshauptstadt Dresden, Straßen- und Tiefbauamt 2015

Der Erhebungsprozess ist detailliert zu dokumentieren. Dies sorgt für Vergleichbarkeit der Verkehrserhebungen und ermöglicht deren objektive Beurteilung. Somit ist die Prozessdokumentation eine Grundvoraussetzung zur Bewertung der Qualität der erhobenen Daten. (FGSV 2012b, S. 48)

3.3 Verkehrserfassungssysteme

3.3.1 Aufbau eines Verkehrserfassungssystems

Automatische Verkehrserfassungssysteme werden als Verkehrsdetektoren bezeichnet. Die FGSV (2012a, S. 124) definiert einen Detektor als „Einrichtung zur Erfassung der Anwesenheit von Fahrzeugen oder Fußgängern sowie des Bewegungsverhaltens, der Abmessungen oder der Arten von Fahrzeugen“. Ein Detektor besteht generell aus einer Erfassungseinheit, Kabel- oder Funkverbindungen und einer Auswerteeinheit.

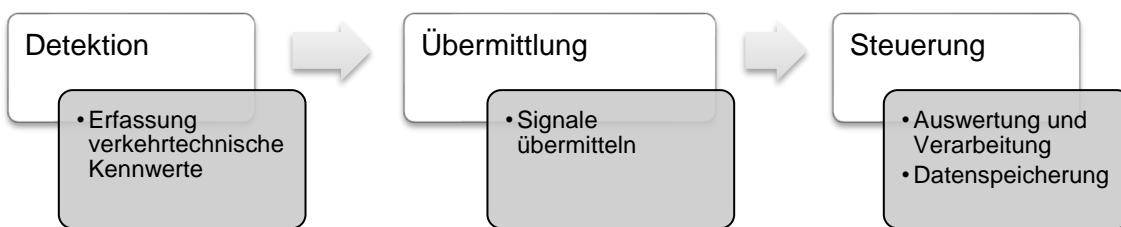


Abbildung 3: Grundprozesse eines Verkehrsdetektors

Quelle: in Anlehnung an Bütler et al. 2010, S. 9

Für die Datenaufnahme bzw. Detektion ist in einem automatischen Verkehrserfassungssystem ein Sensor zuständig. Sensoren erfassen die physikalischen Eigenschaften eines Objektes in der Umgebung wie z.B. Druck, Leitfähigkeit, Reflexion usw. sowie deren Änderungen und wandeln diese in elektrische Signale um (FGSV 2019, S. 10). Sie werden über Kabel- oder Funkverbindungen an die Auswerteeinheit übermittelt. Sie konvertiert die elektrischen Signale in Verkehrsparameter wie Fahrzeugpräsenz, Geschwindigkeit, Gewicht usw. (VDC 2007, S. 3.1)

Der Hauptunterschied zwischen Verkehrserfassungssystemen besteht in der Aufgabeteilung von Detektion, Übermittlung und Steuerung der Daten. Bei einigen Systemen findet die Detektion und Steuerung im selben Gerät statt, während bei anderen Systemen die detektierten Werte an eine externe Steuerungseinheit übermittelt werden. (Bütler et al. 2010, S. 9)

3.3.2 Übersicht und Vergleichskriterien

Die jüngsten Fortschritte in den Bereichen Sensorik, drahtloser Kommunikation und des maschinellen Lernens förderten die Entwicklung zahlreicher innovativer Verkehrserfassungssysteme. Während die neuen Technologien eine höhere Messgenauigkeit ermöglichen, variieren deren Eigenschaften stark und stellen somit

eine Herausforderung für die Auswahl einer optimalen Technologie in Abhängigkeit des Anwendungsfalls dar. (Won 2019)

Bütler et al. (2010) benutzen Systemwahlmatrizen als Methode zur Suche nach einem geeigneten Verkehrserfassungssystem. Dabei wird jedes System „[...] auf die Einsetzbarkeit für unterschiedliche Verkehrsteilnehmer sowie über deren Erfassungsfunktionen und Nutzungseinschränkungen geprüft“ (Bütler et al. 2010, S. 35). Dieser Prozess erfolgt in vier Schritten:

1. Einsetzbarkeit der Verkehrserfassungssysteme für unterschiedliche Verkehrsregelungssysteme (Lichtsignalanlage, Tunnelsteuerung, Rampenwirtschaft und Verkehrsdatenerhebung) analysieren
2. Einsetzbarkeit für unterschiedliche Verkehrsteilnehmer (*Fußgänger, Zweiradfahrzeuge, Motorfahrzeuge*) untersuchen
3. Erfassungsfunktionen in jeweiliger Verkehrsteilnehmergruppe (*Fußgänger, Fahrzeuge*) analysieren
4. Einsatzmöglichkeit der Erfassungssysteme hinsichtlich unterschiedlicher Kriterien bewerten

Da die Erfassungsfunktionen je nach Verkehrsteilnehmer variieren, werden sie auf die für den Fußverkehr und den fahrenden Verkehr unterteilt. Die Gruppe Fahrzeuge enthält sowohl den MIV als auch den Radverkehr. Folgende Erfassungsfunktionen werden betrachtet:

Tabelle 2: Erläuterung zu verkehrstechnischen Kenngrößen

Quelle: in Anlehnung an FGSV 2010, S. 7

Kategorie	Kenngröße	Erläuterung
Fahrzeuge	Fahrzeugmenge	Anzahl der Fahrzeuge je Zeiteinheit an einem Querschnitt
	Fahrtrichtung	Erkennung der Fahrtrichtung des sich bewegenden Fahrzeugs
	Anwesenheit im Stau/Stand	Erkennung sich stauenden und stehenden Fahrzeuge
Fußgänger	Fußgängermenge	Anzahl der Fußgänger je Zeiteinheit an einem Querschnitt

Um die Unterschiede zwischen den einzelnen Verkehrsdatenerfassungssystemen darzustellen, wurden Vergleichskriterien definiert, die in Tabelle 3 aufgelistet und erläutert sind. Als Grundlage für die Definition dieser Vergleichskriterien dienen die Hinweise zu Detektionstechnologien im Straßenverkehr der FGSV (2019, S. 33–35).

Tabelle 3: Kriterien für Detektorvergleich

Quelle: in Anlehnung an FGSV 2019, S. 33–35

Kategorie	Kriterium	Erläuterung
Datenqualität	Genauigkeit	Grad der Übereinstimmung zwischen den erhobenen Daten und der Realität
	Zuverlässigkeit	Funktionsfähigkeit der Detektionstechnologie unter variierenden Randbedingungen
	Vollständigkeit	Grad der Abdeckung
Handhabung	Montage- & Installationsaufwand	zeitlicher und organisatorischer Aufwand, der für Montage und Einrichtung eines Detektionssystems anfällt
	Wartungsintensität	Häufigkeit & Umfang der notwendigen Wartungsarbeiten
	Energieverbrauch	Höhe des Stromverbrauchs
Erfassungsbereich	Reichweite	Größe des Erfassungsbereichs, welcher durch den Detektor abgedeckt wird
	Erfassung mehrerer Fahrstreifen	Möglichkeit eines einzelnen Detektors gleichzeitig mehrere Fahrstreifen zu überwachen
äußere Einflüsse	witterungsabhängige Einflüsse	Funktionsfähigkeit einer Messtechnologie bei folgenden Wetterbedingungen: <ul style="list-style-type: none"> • Niederschlag (Regen, Schnee) • Sonneneinstrahlung • Nebel, Rauch • Lichtverhältnisse • Temperaturschwankungen • Wind
	Diskretion	Unauffälligkeit einer Technologie im öffentlichen Raum
	Verschmutzung	Anfälligkeit der Bauteile einer Messtechnologie gegenüber der Verschmutzung
rechtliche Aspekte	Datenschutz	Sicherstellung der DGSVO-Konformität der Verkehrsdatenerhebung:
Wirtschaftlichkeit	Kosten	Höhe der Planungs-, Bau-, Betriebs- sowie Rückbaukosten

Die Detektoren werden generell „[...] je nach Einsatzzweck und örtlicher Gegebenheit in verschiedenen Positionen in Straßenraum angeordnet“ (FGSV 2010, S. 12). Sie können *in/unter*, *auf*, *neben* oder *über* der Fahrbahn montiert werden und sind in *Systeme im Belag* (in-roadway) und *Systeme über dem Belag* (over-roadway) untergliedert. (Klein et al. 2006a, S. 34)

Systeme im Belag werden entweder in die Fahrbahndecke eingebaut oder auf der Fahrbahnoberfläche befestigt. Systeme über dem Belag werden entweder über der Fahrbahn oder neben der Fahrbahn, mit einem ausreichenden Abstand von der nächsten Fahrspur versetzt. (Klein et al. 2006a, S. 34)

Die aktuellen Verkehrserfassungssysteme sind in Tabelle 4 in Abhängigkeit der Positionierung gegliedert und werden in folgenden Abschnitten detaillierter beschrieben.

Tabelle 4: Übersicht Verkehrserfassungssysteme

<i>Systeme im Belag</i>	<i>Systeme über dem Belag</i>
3.3.3 Induktionsschleife	3.3.6 Videoerfassung
3.3.4 Magnetfelddetektor	3.3.7 Infrarotdetektor
3.3.5 Druckdetektor	3.3.8 Laserdetektor 3.3.9 Ultraschalldetektor 3.3.10 Radardetektor 3.3.11 Kombinationsdetektor 3.3.12 Wi-Fi-, Bluetooth-Detektoren

3.3.3 Induktionsschleife

Eigenschaften und Funktionsweise

Induktionsschleifen sind Drahtschleifen oder Drahtspulen, die unter der Fahrbahn im Fahrbahnbetrag eingelassen sind (Schnieder und Becker 2007, S. 101).



Abbildung 4: Induktionsschleife

Quelle: Kurzmann 2015, S. 11

In den Spulen wird ein elektromagnetisches Feld erzeugt. Die Energie in diesen Feldern wird von der Geometrie- und Materialeigenschaften der Spule sowie der Windungszahl beeinflusst. Die Materialeigenschaften einer Induktionsschleife hängen von den Umgebungsbedingungen ab und ändern sich, wenn ein Fahrzeug die Induktionsschleife überfährt oder in dem von der Schleife umschlossenen Bereich anhält.

Die Änderungen der Materialeigenschaften

und damit verbundenen Änderungen des elektromagnetischen Felds werden erfasst, in ein Signal umgewandelt, welches an die Auswerteeinheit weitergeleitet wird. Jede Klasse von Fahrzeugen verursacht eine für diese Klasse charakteristische Form der Feldänderung. Durch Auswertung der Feldänderung lässt sich die Fahrzeugkategorie bestimmen. Außerdem kann durch eine entsprechende Positionierung mehrerer Induktionsschleifen die Fahrzeuggeschwindigkeit berechnet werden. (Schischmanow 2005, S. 6; Schnieder und Becker 2007, S. 102–103; Klein et al. 2006a, 1.11-1.12)

Installation

Eine oder mehrere Drahtschleifen werden in die gefrästen Langlöcher in die Fahrbahn eingebaut und anschließend vergossen. Die Induktionsschleifen werden fest fixiert, um unerwünschte Verschiebung zu vermeiden. (Bütler et al. 2010, S. 16) Die Installationstechniken unterscheiden sich durch die Behandlung der Ecken, in denen sich zwei Sägeschnitte überschneiden sowie durch Spleißtechniken und Dichtungsarten (Klein et al. 2006b, 5.5).

Vorteile:

- höchste Präzision der Fahrzeugzählung im Vergleich zu anderen häufig verwendeten Erfassungssystemen
- ausgereifte Technologie, viele Erfahrungswerte durch jahrelangen Einsatz
- unempfindlich gegenüber Wetterbedingungen und äußeren Störeinflüssen
- langfristiger Einsatz an Dauer- und Langzeitzählstellen möglich

- flexibles Design, das sich für eine Vielzahl der Anwendungsfälle eignet
- Detektion eines ruhenden und sich stauenden Verkehrs möglich

Nachteile:

- hoher organisatorische und finanzielle Aufwand bei der Installation, Reparatur- und Instandhaltungsarbeiten
- Detektion ausschließlich innerhalb des Bereichs möglich, wo die Schleife eingebaut ist
- lediglich Metallobjekte können detektiert werden
- Beeinflussung des Magnetfelds führt zu Störungen
- geringe Empfindlichkeit und dadurch Schwierigkeiten Fahrzeuge mit hoher Bodenfreiheit bzw. Kunststoffkarosserie zu detektieren

(Bütler et al. 2010, S. 18–19; Klein et al. 2006a, 1.8)

Erfassungsmöglichkeit von Verkehrsteilnehmern

Induktionsschleifen sind in der Lage motorisierte Fahrzeuge aller Kategorien zu detektieren. Die Detektion erfolgt unter der Bedingung, dass im Fahrzeug metallische Bauteile eingebaut sind. (Bütler et al. 2010, S. 18)

Für die Erfassung des Veloverkehrs eignen sich Induktionsschleifen nur dann, wenn sie in Radwegen eingebaut werden und unter der Bedingung, dass die Fahrradfahrer spurtreu fahren. Es gibt Systeme, die Fahrraddetektion auch im Mischverkehr ermöglichen. In diesem Fall muss die Empfindlichkeit der Induktionsschleife auf den Veloverkehr eingestellt werden. (Zweibrücken et al. 2005, S. 56)

Ausgehend von physikalischen Eigenschaften, können die Fußgängerzahlen mithilfe von Induktionsschleifen nicht erfasst werden (Zweibrücken et al. 2005, S. 56).

3.3.4 Magnetfelddetektor

Eigenschaften und Funktionsweise

Magnetfelddetektoren sind passive Detektoren, die das Vorhandensein eines Eisenmetallobjekts erkennen, indem sie die Veränderungen im stationären Erdmagnetfeld detektieren (Klein et al. 2006a, 1.13).



Abbildung 5: Magnetfelddetektor

Quelle: Rosim Magnetelddetektor

Der Hauptunterschied zwischen dem Magnetfeldsensor und der Induktionsschleife besteht darin, dass der Magnetfeldsensor „[...] das permanent bestehende Erdmagnetfeld als Grundlage der Messung verwendet [...]“, während Induktionsschleifen die Veränderung des künstlich erzeugten Magnetfelds erfassen (Bütler et al. 2010, S. 20).

Wenn ein Fahrzeug in die Erfassungszone eines Magnetfeldsensors eintritt, verursachen seine Energiefelder eine magnetische Anomalie - Veränderung des Erdmagnetfelds im Erfassungsgebiet. Anhand dieser Änderung kann der Sensor ein Fahrzeug detektieren und die Größe bestimmen. (Klein et al. 2006a, 1.13; Bütler et al. 2010, S. 20)

Zwei Arten von Magnetfeldsensoren werden zur Messung von Verkehrsstromparametern verwendet. Das zweiachsige Fluxgate-Magnetometer erkennt sowohl den ruhenden als auch den fließenden Verkehr während der Magnetdetektor nur sich bewegende Fahrzeuge detektiert. (Klein et al. 2006a, 2.56)

Installation

Magnetfeldsensoren können im Belag eingebaut oder auch seitlich des Fahrstreifens montiert werden (Bütler et al. 2010, S. 20). Häufig werden sie als Alternative zu Induktionsschleifen an den Orten angebracht, wo unbeschichteter Stahl vorhanden ist bzw. wo die Installation der Induktionsschleifen nicht zulässig ist. Brücken gehören zu den häufigen Einsatzorten von Magnetfeldsensoren. (Klein et al. 2006a, 2.57)

Vorteile:

- unempfindlich gegenüber Wetterbedingungen (Temperatur, Wind, Regen, Schnee, Eis, Verschmutzung)
- Einbau an für Induktionsschleifen ungeeigneten Stellen (z.B. Brücken)
- Einbau im sowie über dem Belag möglich
- Fahrstreifensperre während der Installation nicht zwingend erforderlich

Nachteile:

- Erfassung von ausschließlich ferromagnetischen Objekten
- hoher organisatorischer und finanzieller Aufwand bei Installation, Reparatur- und Instandhaltungsarbeiten
- längliche Erfassungsbereiche sind schwer abzudecken (mehrere Sensoren für die Erkennung des gesamten Bereiches notwendig)

(Klein et al. 2006a, 1.8; Bütler et al. 2010, S. 21–22)

Erfassungsmöglichkeit von Verkehrsteilnehmern

Magnetfeldsensoren können alle motorisierte Fahrzeugtypen erfassen. Dabei werden die ferromagnetischen Bauteile eines Fahrzeuges erkannt. Je weniger solche Bauteile ein Fahrzeug besitzt, desto schwieriger ist es für den Magnetfeldsensor ihn zu detektieren.

Aus diesem Grund ist die Erfassung des Veloverkehrs und der Fußgängerzahlen mithilfe der Magnetfeldsensoren sehr schwer realisierbar und findet in der Praxis nicht statt. (Bütler et al. 2010, S. 21)

3.3.5 Druckdetektor

Eigenschaften und Funktionsweise

Dieses Verkehrserfassungssystem besteht aus einem mit Luft gefüllten Schlauch und einem Druckausgleichgerät, an den der Schlauch angeschlossen ist. (Zweibrücken et al. 2005, S. 57). Die Druckschläuche werden senkrecht zur Verkehrsflussrichtung auf der Oberfläche der Fahrbahn befestigt (VDC 2007, 4.1).

Beim Überfahren erzeugt ein Fahrzeug im Schlauch einen Luftdruck, der entlang des Gummischlauchs weitergeleitet wird. Dieser Druckimpuls schließt einen Luftschalter und erzeugt somit ein elektrisches Signal, das an ein Erfassungsgerät gesendet wird. Die Datenwerte werden von einem integrierten Datenerfassungsgerät gespeichert. (VDC 2007, 4.1; Zweibrücken et al. 2005, S. 57)

Diese Technologie eignet sich für Kurzzeitverkehrszählungen und ist in der Lage Fahrzeuge nach Achszahl und Abstand zu klassifizieren. Sie zählt unter den



Abbildung 6: Druckschlauchsystem

Quelle: *Natural Rubber Road Tube*

bekannten Verkehrserfassungstechnologien zu einer der ältesten und wurde an mehreren Orten durch Induktionsschleifen und Over-Roadway-Sensoren ersetzt. (Klein et al. 2006a, 3.46)

Installation

Für die Befestigung der Druckschläuche auf der Fahrbahnoberfläche werden spezielle Nägel oder andere Metallbefestigungen benutzt. Besonders beim Einsatz auf Fahrradwegen ist besondere Vorsicht geboten, indem Metallvorrichtungen außerhalb der Fahrspur angebracht werden. Für die Fixierung kann auch Klebeband oder anderer passende Klebstoff benutzt werden. (FHWA 2016, 4.16)

Vorteile:

- schnelle Installation und Montage, einfache Instandhaltung
- kostengünstige, tragbare Low-Power-Technologie

Nachteile:

- Anfälligkeit gegenüber äußeren Störeinflüssen sowie Temperaturänderungen
- ungenaue Achszählung bei hohem LKW- und Busaufkommen
- begrenzte Lebensdauer der Druckschläuche wegen hoher Anfälligkeit für die Abnutzung durch vorbeifahrenden Verkehr

(Klein et al. 2006a, 3.46; VDC 2007, 4.1; Zweibrücken et al. 2005, S. 57)

Erfassungsmöglichkeit von Verkehrsteilnehmern

Der Druckschlauchsensor eignet sich für die Erfassung des rollenden Verkehrs und kann die Anzahl der Fahrzeuge sowie deren Fahrtrichtung detektieren. Für die Klassifizierung zwischen den einzelnen Fahrzeugarten ist diese Technologie eher ungeeignet. (VDC 2007, 4.1)

Laut FHWA (2016, 4.15-4.16) können die Druckschläuche für die Erfassung des Veloverkehrs eingesetzt werden. Dafür muss die Empfindlichkeit des Sensors entsprechend eingestellt werden. Obwohl diese Einstellung ohne großen Aufwand möglich ist, empfiehlt die FHWA die Messungen nur auf eigenen Spuren (Radweg, Radstreifen) durchzuführen. Um die Fahrtrichtung des Radfahrers zu detektieren, sollen zwei pneumatische Druckschläuche mit genügend Abstand parallel zueinander platziert werden. Die Fahrtrichtung kann aus dem Zeitpunkt der Erkennungsereignisse an jedem Schlauch abgeleitet werden.

Die Fußgängerzahlen können mithilfe der Druckschläuche nicht ermittelt werden. Oft werden aber die Druckschläuche mit Infrarotsensoren an Orten kombiniert, wo sowohl Radfahrer- als auch Fußgängerzählungen stattfinden sollen. (FHWA 2016, 4.15)

3.3.6 Videoerfassung

Eigenschaften und Funktionsweise



Die Verkehrserfassung mittels Videokamera gehört zu Over-Roadway-Technologien und ist am weitesten verbreitet. Videokameras erfassen und liefern umfassende Informationen für die Fahrzeugklassifizierung. Im Gegensatz zu In-Roadway-Systemen, bei denen mehrere Detektoren für die Erfassung mehrerer Fahrspuren erforderlich sind, kann eine einzige Kamera Verkehrsdaten mehrerer Spuren erfassen. (Won 2019, S. 9)

Abbildung 7: Videoerfassung

Quelle: Breithaupt

Ein Videoerfassungssystem besteht aus einer oder mehreren Kameras, einem mikroprozessorbasierten

Computer zum Digitalisieren und Analysieren der Aufnahmen sowie einer Software, die die Aufnahmen interpretiert und in Verkehrsflussdaten umwandelt. (Klein et al. 2006a, 1.14)

Heutzutage werden drei Arten der Videoerfassungssysteme eingesetzt: *Tripline*, *Closed-Loop-Tracking* und *Data Association Tracking*. Bei *Tripline*-Systemen definiert der Anwender eine begrenzte Anzahl linearer Erfassungszonen auf der Fahrbahn, die von der Videokamera überwacht wird. Das Erscheinen eines Objektes im Sichtfeld einer Videokamera bedeutet Änderung der Eigenschaften von den Pixeln, die sich in diesem Bereich befinden. Wenn ein Fahrzeug oder Fußgänger eine der definierten Zonen überquert, registriert die Software diese Pixeländerung und vergleicht den aktuellen Zustand mit dem vorherigen, in dem das Objekt abwesend war. *Tripline*-Systeme sind in der Lage Geschwindigkeiten zu schätzen. Dabei messen sie die Zeit, die ein detektiertes Fahrzeug benötigt, um eine Erfassungszone bekannter Länge zu durchqueren. (Klein et al. 2006a, 2.68)

Closed-Loop-Trackingsysteme erkennen Fahrzeuge entlang größerer Fahrbahnabschnitte und verfolgen sie kontinuierlich. Diese Systeme stellen zusätzliche Informationen über das Verhalten des Fahrzeuges bereit. Besonders wertvoll ist die Verfolgung der Fahrzeugbewegung von Spur zu Spur. Diese Daten können an Displays und Radios übertragen werden, um Fahrer in der Umgebung auf unberechenbares Verhalten des erfassten Fahrzeugs aufmerksam zu machen. (Klein et al. 2006a, 2.69)

Data-Association-Trackingsysteme identifizieren und verfolgen ein Fahrzeug oder eine Fahrzeuggruppe, sobald sie im Sichtfeld der Kamera erscheinen. Der Computer

erkennt Fahrzeuge, indem er nach verbundenen Pixelbereichen sucht. Diese Bereiche werden von Bild zu Bild beobachtet, um Verfolgungsdaten zu erstellen. Fahrtzeit und Informationen zum Ausgangs- sowie Zielort können ermittelt werden, indem die Fahrzeuge verfolgt werden, wenn sie vom Sichtfeld einer Kamera zum Sichtfeld der nächsten Kamera wechseln. (Klein et al. 2006a, 2.70)

Installation

Die Videoerfassungsgeräte werden meistens überkopf montiert und so eingestellt, dass sich der gewünschte Erfassungsbereich im Sichtfeld der Kamera befindet (Bütler et al. 2010, S. 26). Die Genauigkeit der Geschwindigkeitsmessung wird von der Anbringungshöhe einer Kamera beeinflusst. Je niedriger das Videogerät befestigt ist, desto größer ist der Fehler, da der Messfehler proportional zur Fahrzeughöhe geteilt durch die Kamerahöhe ist. Viele Hersteller geben einen maximalen Überwachungsbereich ihrer Videogeräte, der das Zehnfache der Anbringungshöhe beträgt. In Realität ist die Reichweite von Faktoren wie Straßenkonfiguration, Überlastungsgrad, Wetterbedingungen usw. abhängig und somit deutlich niedriger. Laut Klein et al. (2006a, 2.75-2.76) hängt die Reichweite explizit von der Höhe der Kameramontage, dem Abstand zum Fahrzeug und der Fahrzeughöhe ab.

Vorteile:

- Erfassung auf mehreren Fahrspuren bzw. in mehreren Erkennungszonen möglich
- Einfaches Hinzufügen und Ändern der Erfassungszonen
- geringer Installationsaufwand
- Eignung auf eine Vielzahl an Anwendungsfällen durch hohe Flexibilität
- alle Arten von Verkehrsteilnehmern detektierbar

Nachteile:

- hohe Anfälligkeit gegenüber Witterungseinflüssen
- eine zuverlässige Nachtüberwachung erfordert eine Straßenbeleuchtung
- bei Verschmutzung der optischen Sensoren fehlerhafte Erfassung
- Vandalismusgefahr
- teure Gewährleistung des Datenschutzes

(Bütler et al. 2010, S. 27; Klein et al. 2006a, 1.9; Zweibrücken et al. 2005, S. 59; Risse et al. 2018, S. 21)

Erfassungsmöglichkeit von Verkehrsteilnehmern

Videoerfassungssysteme eignen sich für die Erfassung Fahrzeuge aller Kategorien inkl. Veloverkehrs und bieten oft Klassifizierungsmöglichkeiten. Sie können große Verkehrsbelastungen erfassen und Informationen über die Anzahl, die Lage und die Bewegungsrichtung der Objekte liefern. (Zweibrücken et al. 2005, S. 59) Außerdem können die Videogeräte für die Erfassung der Fußgängerzahlen eingesetzt werden. Manche Systeme liefern Informationen über die zurückgelegten Strecken sowie die Aufenthaltsdauer der Passanten. (Risse et al. 2018, S. 21)

3.3.7 Infrarotdetektor

Eigenschaften und Funktionsweise

Infrarotdetektoren dienen zur Detektion von Wärmestrahlung im Infrarotbereich. Die Passiv-Infrarot-Sensoren (PIR) erfassen Wärmestrahlen, die von Straßen, Fahrzeugen oder Fußgängern abgegeben bzw. reflektiert werden. Selbst übertragen sie keine Energie und senden keine Signale, deswegen werden sie *Passiv-Sensoren* genannt. PIR-Sensoren erfassen die Temperatur der Fahrbahn sowie die der Objekte (der Fahrzeuge bzw. Fußgänger). Die Objekte werden aufgrund des Temperaturunterschiedes gegenüber dem Hintergrund (Fahrbahn) detektiert. (Bütler et al. 2010, S. 27–28; Klein et al. 2006a, 2.85)

Alle Infrarotdetektoren können langsame Temperaturschwankungen durch Tageszeiten, Sonneneinstrahlung oder Schatten kompensieren (FGSV 2019, S. 37).

Installation

Die Installation der Infrarotdetektoren erfolgt entweder über Kopf oder seitlich am Straßenrand. Das Gerät wird oft am Signalmast oder am Lichtmast entgegen oder in Fahrtrichtung des Verkehrsstroms angebracht. Der Anbringungsort soll so ausgewählt werden, dass eine direkte Sichtverbindung zwischen dem PIR-Sensor und der Erfassungszone besteht. Das Sichtfeld des Sensors darf nicht von Gebäuden oder anderen Objekten beeinträchtigt werden. Die Reichweite der PIR-Sensoren beträgt 20–50 Meter. Die Form des Detektionsbereich ist kreisrund oder elliptisch, wobei die Größe je nach Detektortyp zwischen 2 und 20 Metern für die Geräte, die über der Fahrbahn angebracht sind, variiert. Für die seitlich befestigten Sensoren kann eine



Abbildung 8: Infrarotdetektor

Quelle: Bächle 2012

Detektionszone 2 bis 3 Metern betragen. Um den Detektor auf die Erfassung eines einzigen Fahrstreifens zu beschränken, muss bei der Montage ein kleiner Öffnungswinkel eingestellt werden. (FGSV 2019, Anhang 10.2, Abschnitt 4.2.4)

Vorteile:

- niedriger Installations- und Wartungsaufwand
- Flexibilität bei Fahrstreifenänderung
- Erkennung aller Verkehrsteilnehmer möglich
- niedriger Stromverbrauch
- Diskretion und Anonymität der erfassten Objekte

Nachteile:

- Anfälligkeit gegenüber Witterungseinflüssen (Regen, Schnee, Nebel)
- Verschmutzung der Sensorabdeckung führt zu Fehlern
- Eingeschränkte Genauigkeit bei längerem Aufenthalt (> 1 Minute) im Erfassungsbereich bzw. zu niedriger (< 3 km/h) oder zu hoher (> 150 km/h) Geschwindigkeit
- nicht sinnvoll für Erfassung des stauenden Verkehrs bzw. parkenden Fahrzeuge
- freies Sichtfeld notwendig
- nebeneinander gehende bzw. mehrere Personen im Detektionsbereich werden als eine Person erfasst

(FGSV 2010, S. 20; FGSV 2019, Anhang 10.2, Abschnitt 4.2.4; Klein et al. 2006a, 1.9)

Erfassungsmöglichkeit von Verkehrsteilnehmern

Der Infrarotdetektor wird für Erfassung von Kraftfahrzeugen, Radfahrern und Fußgängern eingesetzt. Er kann die Fahrzeug- bzw. Fußgängerpräsenz messen sowie Verweildauer im Erfassungsbereich berechnen. Die Genauigkeit der Mengen- und Geschwindigkeitserfassung sowie Klassifizierung unterschiedlicher Fahrzeugarten ist eingeschränkt. Die Infrarotdetektoren eignen sich für die Analyse des Verkehrszustandes sowie dessen Änderung und werden für die Erfassung des zähflüssigen bzw. stockenden Verkehrs eingesetzt. (FGSV 2010, S. 20)

3.3.8 Laserdetektor

Eigenschaften und Funktionsweise

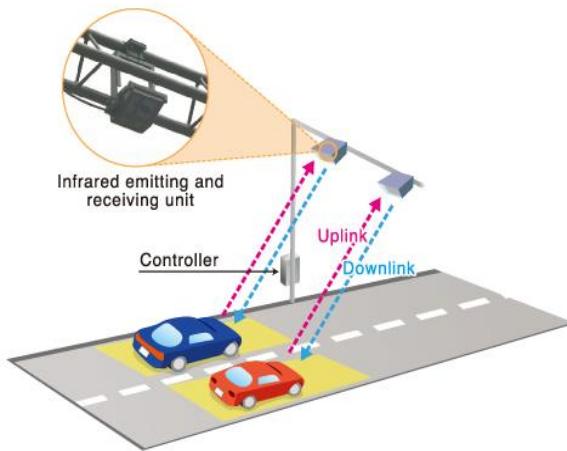


Abbildung 9: Laserdetektoren

Quelle: Sumitomo Electric

Die Laserdetektoren sind aktive Infrarotdetektoren und unterscheiden sich von Passiv-Infrarot-Detektoren darin, dass sie Energie im nahen Infrarotspektrum übertragen. Sie verfügen über zwei Optiksätze: Sende- und Empfangsoptik. Die Sendeoptik teilt den Laserdiodenausgang in zwei Strahlen auf. Die Empfangsoptik hat ein breites Sichtfeld, um die von Objekten reflektierte Energie zu sammeln. Die Sendung mehrerer Strahlen ermöglicht eine Geschwindigkeitsmessung, indem die Zeiten aufgezeichnet werden, zu denen das

Fahrzeug in den Erfassungsbereich eines jeden Strahls einfährt. Da der Abstand zwischen den gesendeten Strahlen bekannt ist, kann die Geschwindigkeit anhand der Division der Entfernung durch die Zeit berechnet werden. (Klein et al. 2006a, 2.83)

Die Laserdetektoren dienen der schnellen und präzisen Detektion von dreidimensionalen (3D) Objekten. Sie werden für 3D-Vermessungen sowie für Klassifizierung der detektierten Fahrzeuge in bis zu 28 Fahrzeugarten eingesetzt. Das Detektionsprinzip von Laserdetektoren basiert auf dem „Time-of-Flight“-Verfahren (ToF). Dabei sendet der Laserdetektor aktiv die Laserstrahlen im Infrarot-Spektrum. Diese Strahlen werden an Objekten im Erfassungsbereich reflektiert. Die Strahllaufzeit wird registriert und die Scanebene wird in unterschiedlichen Winkeln und mit variierteren Scanfrequenzen gemessen. So können die 3D-Punktwolken erstellt werden, die nach dem Ausfiltern von Ausreißern zu Linioprofilen von 3D-Objekten werden. Anhand der Linioprofilen werden Höhe, Breite und Länge der detektierten Objekte berechnet. Für die Fahrzeugklassifizierung werden diese Werte mit den Abmessungen der Fahrzeugmustern verglichen und somit wird das erfasste Fahrzeug der entsprechenden Fahrzeugklasse zugeordnet. (FGSV 2019, S. 28–29)

Installation

Die Installation der Laserdetektoren soll über dem Fahrstreifen erfolgen. Sollte die Detektion über zwei Fahrstreifen erfolgen, kann das Gerät mittig über zwei Fahrstreifen angebracht werden. In diesem Fall wird die Erfassungsgenauigkeit geringer. Laserdetektoren sollten in einer Höhe von 5,5 bis 10 Meter über Fahrbahnoberfläche

montiert werden. Dabei soll ähnlich wie bei PIR-Detektoren ein freies Sichtfeld zwischen dem Sensor und der Erfassungszone bestehen. (FGSV 2010, Anhang 10.2, Abschnitt 4.2.8)

Vorteile:

- relativ witterungsunabhängig
- Einsatz eines Detektors für mehrere Fahrstreifen möglich
- Erfassung 3D-Profile der detektierten Objekte
- hohe Detektionsrate und -genauigkeit
- Erkennung aller Verkehrsteilnehmer möglich

Nachteile:

- Anfälligkeit gegenüber dichtem Nebel bzw. starkem Niederschlag
- Installation an einem Mast bzw. an einer Verkehrszeichenbrücke
- freies Sichtfeld notwendig
- Verschmutzung der Sensorabdeckung
- hoher Energieverbrauch

(FGSV 2019, Anhang 10.2, Abschnitt 4.2.8; Klein et al. 2006a, 1.9; Bütler et al. 2010, S. 30)

Erfassungsmöglichkeit von Verkehrsteilnehmern

Die Laserdetektoren können sowohl Fahrzeuge als auch Fußgänger detektieren und eignen sich somit für eine Vielzahl der Anwendungen. Sie ermitteln Anzahl, Geschwindigkeit und 3D-Angaben zum Objekt und sind in der Lage das Verkehrsverhalten zu erkennen. Sie können große Verkehrsmengen in kurzer Zeit erfassen. Alle Detektoren ermöglichen richtungsabhängige Verkehrserfassung. Die Software kann an die Anwendungssituation angepasst werden. (Zweibrücken et al. 2005, S. 60; Risse et al. 2018, S. 20)

3.3.9 Ultraschalldetektor

Eigenschaften und Funktionsweise

Ultraschallgeräte sind aktive Detektoren, die in regelmäßigen Abständen Schallwellen mit einer Frequenz zwischen 16 und 35 kHz senden. Die Frequenz der Schallübertragung liegt somit meist über dem von Menschen hörbaren Bereich. Wenn diese Wellen ein Objekt treffen, werden sie in Form eines Echos reflektiert, welches vom Empfänger registriert wird. Anhand der Laufzeit der Schallimpulse berechnet der Detektor den Abstand zwischen ihm und dem erfassten Objekt. Die Reichweite dieser Sensoren liegt bei maximal 8 Metern. (FGSV 2010, S. 21)

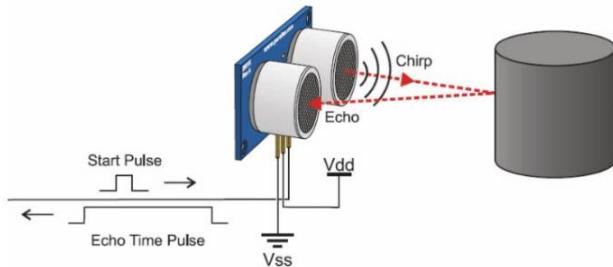


Abbildung 10: Ultraschalldetektor

Quelle: Wong 2017

Mithilfe dieser Technologie können Fahrzeugpräsenz, deren Höhe (bei Installation über Kopf) und Geschwindigkeit erfasst werden. Anhand der Höhe kann eine Unterscheidung nach Fahrzeugkategorien durchgeführt werden. Die Genauigkeit der Fahrzeugklassifizierung mit einem Ultraschalldetektor ist im Vergleich zu anderen Technologien gering und kommt selten im Einsatz. Am häufigsten werden Ultraschalldetektoren für die Überwachung von Stellplätzen in Parkbauten eingesetzt. Außerdem ist der Ultraschallsensor ein wichtiger Teil des sogenannten Kombinationsdetektors (siehe Abschnitt 3.3.11). (FGSV 2010, S. 20–21)

Installation

Die Ultraschalldetektoren können über der Fahrbahn installiert werden. In diesem Fall sind besteht die Möglichkeit, die Höhe eines Fahrzeugs zu messen und somit eine grobe Klassifizierung zwischen den Fahrzeugarten durchführen. Der Detektor wird über der Fahrstreifenmitte z.B. an einem Lichtmast angebracht, wo eine kontinuierliche Stromversorgung sichergestellt werden kann. Seltener werden die Ultraschallsensoren seitlich der Fahrspur montiert, wobei die Frequenzänderung erfasst und so die Geschwindigkeit gemessen werden kann. (FGSV 2010, S. 20–21)

Vorteile:

- genaue Abstandsmessung und Objekterkennung
- Erfassung mehrerer Spuren möglich

Nachteile:

- niedrige Erfassungsgenauigkeit (außer beim Einsatz in den Parkhäusern)
- niedrige Reichweite
- Anfälligkeit gegenüber lauten Nebengeräuschen sowie Temperaturänderungen, starkem Wind, Nebel, Schnee
- freies Sichtfeld notwendig

(FGSV 2010, S. 20–21; Klein et al. 2006a, 1.9)

Erfassungsmöglichkeit von Verkehrsteilnehmern

Die Ultraschalldetektoren sind in der Lage alle Objekte zu erkennen, die sich in der Reichweite des Sensors befinden. Mit einer geeigneten Installation und angepassten Einstellungen können diese Detektoren Fahrzeuge des MIV, Fahrradfahrer sowie Fußgänger detektieren. Die Unterscheidung zwischen den Verkehrsteilnehmern ist mit Ultraschalltechnologie nur bedingt möglich. Aus diesem Grund wird empfohlen die Detektoren auf abgetrennten Wegen zu installieren, wo nur eine Gruppe der Verkehrsteilnehmer unterwegs ist. Das bedeutet, dass die Erfassung des Veloverkehrs ausschließlich auf Fahrradwegen und Detektion der Fußgänger ausschließlich auf Gehwegen möglich sind. Dabei ist es wichtig, dass die Erfassungsbereich des Sensors der Breite des Weges entspricht (falls seitlich angebracht), damit nur Verkehrsteilnehmer einer Gruppe erfasst werden. (Risse et al. 2018, S. 16)

3.3.10 Radardetektor

Eigenschaften und Funktionsweise

Abbildung 11: Radardetektor

Quelle: Radargeräte auf der A6 2019

Radar ist eine Vorrichtung zum Senden elektromagnetischer Signale und zum Empfangen der Echos von detektierten Objekten innerhalb seines Erfassungsbereichs. Radargeräte ermöglichen eine Erfassung der Geschwindigkeit, Entfernung sowie Form und Größe der Objekte. Es gibt diverse Radar-Technologien, die unterschiedliche Funktionsweisen haben und somit unterschiedliche Verkehrskenngrößen erfassen können. (Klein et al. 2006a, 1.15-1.17)

In der Verkehrstechnik kommen Dopplerradardetektoren im Einsatz. Sie strahlen Signale einer bestimmten Frequenz kontinuierlich aus und werden aus diesem Grund

als Dauersender oder CW-Radare (Continuous Wave) bezeichnet. Wenn ein Objekt sich im Erfassungsbereich bewegt, erzeugt es eine Frequenzänderung. Das Radargerät empfängt diese Änderung und kann darauf basierend die Geschwindigkeit des Objektes berechnen. Die Frequenzänderung wird lediglich von sich bewegenden Objekten (Mindestgeschwindigkeit 1-8 km/h) verursacht – die ruhenden Objekte werden nicht erfasst. (FGSV 2019, S. 38, 2010, S. 22–24; Hering und Schönfelder 2018, S. 280–281)

Um die Detektion von stehenden oder sich langsam bewegenden Objekten zu ermöglichen, wurden modulierte CW-Radare entwickelt. Die FMCW-Radare (Frequency Modulated Continuous Wave) können im Gegensatz zu CW-Radaren eine absolute Entfernung mithilfe der Frequenzmodulation ermitteln. Viele Detektoren können beide Technologien gleichzeitig verwenden und ermöglichen somit die Erfassung von Objekten unterschiedlicher Geschwindigkeit. (FGSV 2019, S. 38)

Installation

Die Radardetektoren können sowohl seitlich als auch über der Fahrbahn montiert werden. Bei der Montage soll der Messwinkel genau ausgerechnet bzw. eingehalten werden, um Genauigkeitsverlust bei Geschwindigkeitsmessung zu vermeiden.

Seitenradare eignen sich für den Einsatz an Einbahnstraßen, denn so kann keine Abschattungen der in Gegenrichtung fahrenden Fahrzeuge entstehen. Der Einsatz von Seitenradaren auf mehrspurigen Fahrbahnen ist nur dann sinnvoll, wenn kein großes Verkehrsaufkommen zu erwarten ist und hohe Fehlertoleranz vorhanden sind. Die Seitenradare können an Pfosten von Verkehrsschildern befestigt werden. Die ideale Anbringungshöhe liegt bei 1-1,5 Meter über der Fahrbahn. (FGSV 2010, S. 22–24)

Bei der Installation über der Fahrbahn ist eine mehrspurige Detektion möglich. Bei Anbringung mehrerer Radargeräte nebeneinander ist darauf zu achten, dass keine gegenseitigen Störungen entstehen. Die elliptischen Erfassungszonen dürfen sich nicht überschneiden und die Frequenzen, die von den Radargeräten gesendet werden, sollen sich voneinander deutlich unterscheiden. (FGSV 2010, S. 22–24)

Vorteile:

- präzise Geschwindigkeitsmessung
- Unempfindlichkeit gegenüber Wetter- sowie Umgebungsbedingungen
- Detektion aller Verkehrsteilnehmer möglich (bei Kombination von CW und FMCW)
- geringer Installations- und Wartungsaufwand
- große Reichweite (20-50 Meter)

Nachteile:

- nur eine Detektionszone pro Detektor möglich
- bei CW-Technologie nur Erfassung von bewegenden Objekten möglich
- Einsatz in Kurven nicht empfehlenswert
- freies Sichtfeld notwendig

(FGSV 2019, Anhang 10.2, Abschnitt 4.2.6; Klein et al. 2006a, 1.8)

Erfassungsmöglichkeit von Verkehrsteilnehmern

Mit Radarsensoren sind Erfassungen aller Verkehrsteilnehmer realisierbar. Sie liefern Information bezüglich Menge und Geschwindigkeit der erfassten Objekte und können sowohl permanent als auch periodisch eingesetzt werden. Radargeräte können anhand der Frequenz die Entfernung zum Fahrzeug oder Fußgänger messen und somit die Bewegungsrichtung ermitteln. (Risse et al. 2018, S. 14; FGSV 2019, S. 38)

3.3.11 KombinationsdetektorEigenschaften und Funktionsweise

In Kombinationsdetektoren sind verschiedene Sensoren integriert mit dem Ziel, die Vorteile der einzelnen Technologien zu verbinden und deren Nachteile auszugleichen. Am häufigsten vereinigen die Kombinationsdetektoren Radar-, Ultraschall- und PIR-Sensoren. Dabei erfüllt jeder Sensor eine bestimmte Aufgabe und ist für die Erfassung spezieller Merkmale zuständig. So erfasst der Radarsensor die Geschwindigkeit, der Ultraschallsensor ermittelt die Höhenprofile und sorgt für die genaue Objektzählung sowie -klassifizierung, während der PIR-Sensor die Objektlänge detektiert und dafür sorgt, dass nur Objekte in einer bestimmten Erfassungszone erkannt werden. (FGSV 2010, S. 28–29, 2019, S. 38)



Abbildung 12: Kombinationsdetektor

Quelle: Xtralis ASIM

Installation

In Abhängigkeit des Anwendungsfalls kann die Montage direkt über der Fahrbahn erfolgen – für die spurselektive Erfassung, oder seitlich – für die Erfassung der Objekte nur einer Fahrspur. Die auf dem Markt angebotenen Systeme werden ausschließlich überkopf installiert. Das Gerät soll über die Fahrbahnmitte auf Lichtmasten oder

Schilderbrücken montiert und über Kabelverbindungen mit der Speichereinheit verbunden werden. Um die erwartete Messgenauigkeit zu erreichen, muss bei der Einrichtung des Geräts der vorgegebene Anstellwinkel des Radarsensors eingestellt werden. (FGSV 2010, S. 28, 2019, S. 38)

Vorteile:

- hohe Verfügbarkeit durch Kombination der Vorteile eingesetzter Sensortechnologien
- Unempfindlich gegenüber Umgebungsbedingungen
- hohe Genauigkeit und Zuverlässigkeit
- hochwertige Klassifizierung der Fahrzeugklassen (bis zu 8 Klassen)
- Erfassung des Verkehrs in zu- und abfließender Richtung (Wechsel des Betriebsmodus notwendig)

Nachteile:

- Überkopfmontage-Infrastruktur notwendig
- Fehleranfälligkeit in Stop-and-Go-Verkehr
- Infrastruktur für Datenabfrage- und -speicherung notwendig

(FGSV 2019, Anhang 10.2, Abschnitt 4.2.7; FGSV 2010, S. 28)

3.3.12 Wi-Fi-, Bluetooth-Detektoren

Bluetooth- und Wi-Fi-Detektoren werden zur Erkennung mobiler Endgeräte (Smartphones, Tablets usw.), die sich in Fahrzeugen befinden bzw. von Fußgängern mitgetragen werden, eingesetzt. Es werden lediglich Geräte identifiziert, deren Bluetooth bzw. Wi-Fi-Schnittstelle aktiviert ist. Beim Einsatz zweier Bluetooth-Detektoren (mit reichlich Abstand zwischen ihnen) können Fahrtzeiten, Fahrzeuggeschwindigkeiten sowie Verlustzeiten gemessen werden. Diese Technologie ermöglicht eine eindeutige Identifizierung des Verkehrsteilnehmers unabhängig vom genutzten Verkehrsmittel. Dies ist aufgrund der Geräteadresse (Media-Access-Control-Adresse, kurz: MAC) möglich, die im Moment der Detektion ausgelesen wird. Somit ist die Erfassung der Quelle-Ziel-Beziehung sowie Fahrzeug- bzw. Fußgängerverfolgung möglich. Da über die MAC-Adresse eine Person eindeutig identifizierbar ist, gilt diese Information als personenbezogene Daten. Aus diesem Grund soll Einsatz der Bluetooth- bzw. Wi-Fi-gestützten Technologien laut FGSV (2019, S. 41) „[...] nur nach Absprache mit dem örtlichen Datenschutzbeauftragten oder dem Landesbeauftragten für den Datenschutz durchzuführen.“ Die Messergebnisse der Wi-Fi- und Bluetooth-Detektoren sind ungenau, was mehrere

Gründe hat. Zum einen, haben die Verkehrsteilnehmer keine mobilen Geräte bei sich oder sie sind ausgeschaltet. Zum anderen, kann ein Verkehrsteilnehmer mehrere mobilen Geräte mit aktiven Wi-Fi- oder Bluetooth-Schnittstellen mitführen. Außerdem können bei Fahrzeugmessungen der Distanzfehler zwischen Detektor und dem mobilen Gerät sowie fehlende zeitliche Synchronisation zwischen den zwei Detektoren zu unplausiblen Ergebnissen beitragen. (FGSV 2019, S. 40–41, Anhang 10.2, Abschnitt 4.2.12; Risse et al. 2018, S. 22)

3.3.13 Weitere Erfassungssysteme

Es gibt eine Vielzahl von Verkehrserfassungssystemen, die sich aktuell in Entwicklung befinden oder sich nur für einen Einsatz in besonderen Situationen eignen. Diese Technologien werden in diesem Abschnitt beschrieben.

Lichtschränken finden den Einsatz im Rahmen der Verkehrsüberwachung und eignen sich zur Geschwindigkeitsmessung. Ein Lichtschranksystem besteht aus zwei Lichtschränken, die quer zur Fahrtrichtung angebracht sind. Jede Lichtschanke besteht aus einem Lichtempfänger und einem Lichtsender. Wenn ein Objekt die Erfassungszone des Detektors betritt, wird das Lichtweg unterbrochen. Diese Unterbrechung wird vom Sensor registriert. Da der Abstand zwischen den zwei Lichtschränken bekannt ist, wird anhand der Zeit, die ein Fahrzeug für diesen Weg benötigt hat, die Fahrtgeschwindigkeit ermittelt. (FGSV 2010, S. 18–19)

3D-Flächendetektor kann genaue Objektmessungen auf sehr großen Flächen durchführen. Es ist möglich „[...] mit nur einem Sensor eine Kreisfläche von bis zu 160 m präzise zu vermessen“ (FGSV 2019, S. 39) Die detektierte Fläche kann mithilfe der Kombination mehrerer Einheiten skaliert werden. Die 3D-Flächendetektoren werden für die Überwachung von Parkplätzen eingesetzt, um Parklücken zu ermitteln sowie unzulässig genutzte Parkflächen zu identifizieren. (FGSV 2019, S. 39)

Fahrzeugseitige Erfassungssysteme werden seit Beginn des 21. Jahrhunderts immer beliebter. Die Grundidee dieser Erfassungsart besteht an Datensammlung der Fahrzeuge, in denen die entsprechenden Messgeräte installiert sind. Die dabei gesammelten Daten werden FCD (Floating Car Data) genannt. Die neueren Fahrzeuge sind mit unterschiedlichen Technologien ausgestattet, z.B. Navigationssystem, GPS-Modul, Mobilfunk usw. Die GPS-Satelliten können solche Geräte lokalisieren, deren Daten aufgezeichnet und an die Serverzentrale übertragen werden. Da noch nicht alle Fahrzeuge mit solchen Systemen ausgestattet sind, können Unter- bzw. Überrepräsentierungen auftreten und Ergebnisse beeinflussen (z.B., wenn auf einer Strecke mehr LKWs als PKWs mit FCD unterwegs sind, wird die Durchschnittsgeschwindigkeit zu niedrig berechnet). (FGSV 2019, S. 41)

3.3.14 Zusammenfassende Bewertung

Dieser Abschnitt beinhaltet die Vergleiche der einzelnen Technologien anhand deren Einsetzbarkeit für unterschiedliche Verkehrsteilnehmer (s. Tabelle 5), deren Erfassungsfunktionen (s. Tabelle 6) sowie anhand bereits definierter Kriterien (s. Tabelle 7). Die Eignung der jeweiligen Technologie wird mit -, 0 und + bewertet, was für *nicht geeignet*, *bedingt geeignet* und *gut geeignet* steht.

Tabelle 5: Erfassung von Verkehrsteilnehmern

Quelle: in Anlehnung an FGSV 2010, S. 7, Bütler et al. 2010, S. 37

Verkehrsteilnehmer	Verkehrserfassungssysteme									
	Induktionsschleife	Magnetfelddetektor	Druckdetektor	Videoerfassung	Infrarotdetektor	Laserdetektor	Ultraschalldetektor	Radardetektor	Kombinationsdetektor	Wi-Fi-, Bluetooth-Detektoren
Fußgänger	-	-	-	+	+	+	+	+	+	0
Radfahrer	0	-	+	+	+	+	+	+	+	0
MIV	+	+	+	+	+	+	+	+	+	0

Tabelle 6: Zuordnung von verkehrstechnischen Kenngrößen zu Verkehrserfassungssystemen

Quelle: FGSV 2019, S. 7

Kenngröße	Verkehrserfassungssysteme									
	Induktionsschleife	Magnetfelddetektor	Druckdetektor	Videoerfassung	Infrarotdetektor	Laserdetektor	Ultraschalldetektor	Radardetektor	Kombinationsdetektor	Wi-Fi-, Bluetooth-Detektoren
Fahrzeugmenge	+	+	+	+	+	+	+	+	+	+
Fahrtrichtung	+	+	+	+	0	+	+	+	+	-
Anwesenheit im Stau/Stand	0	0	-	+	-	+	0	-	0	+
Fußgängermenge	-	-	-	+	+	+	+	+	+	0

Die Erläuterungen zu der untenstehenden Tabelle sind im Anhang A zu finden.

Tabelle 7: Vergleich der Verkehrserfassungssysteme

Quelle: in Anlehnung an FGSV 2019 , Bütler et al., S. 39

Vergleichskriterien	Verkehrserfassungssysteme									
	Induktionsschleife	Magnetfelddetektor	Druckdetektor	Videoerfassung	Infrarotdetektor	Laserdetektor	Ultraschalldetektor	Radar detektor	Kombinationsdetektor	Wi-Fi-, Bluetooth-Detektoren
Genauigkeit	+	+	+	0	0	0	0	+	+	0
Zuverlässigkeit	+	+	0	+	0	+	0	0	0	0
Vollständigkeit	0	0	0	0	0	+	+	0	+	-
Montage- & Installationsaufwand	-	-	+	+	+	+	+	+	+	+
Wartungsintensität	+	+	+	-	0	0	0	0	0	-
Stromversorgung	+	+	+	-	+	-	+	+	0	-
Reichweite	-	-	-	+	+	+	0	+	0	+
Erfassung mehrerer Fahrstreifen	-	-	-	+	-	+	-	-	+	-
witterungsabhängige Einflüsse	+	+	+	-	-	0	0	0	0	0
Diskretion	+	+	0	0	0	0	0	0	0	0
Verschmutzung	+	+	+	-	0	-	0	0	0	0
Datenschutz	+	+	+	-	+	+	+	+	+	-
Kosten	-	-	0	-	0	-	+	+	0	-

4 Konzept

4.1 Grundlagen

4.1.1 Erhebungsanlass

Das folgende Konzept eignet sich insbesondere für Gebiete, die zukünftig (in 5-10 Jahren) mit erhöhtem Standortwachstum sowie der damit verbundenen Verkehrsüberlastung rechnen. Infolgedessen planen diese Standorte diverse Mobilitätsmaßnahmen mit dem Ziel, die Überlastung der kritischen Knotenpunkte zu vermeiden.

Um die Wirksamkeit und Angemessenheit der eingeführten Mobilitätsmaßnahmen zu überprüfen, ist die Veränderung der Mobilität zu erfassen. Um diese Mobilitätsveränderung festzustellen, soll deren aktueller Status ermittelt werden. Diese Analyse ist sowohl vor Beginn als auch nach der Umsetzung der Maßnahmen durchzuführen. Für die korrekte Interpretation der Ergebnisse ist deren Vergleichbarkeit entscheidend.

Aus diesem Anlass wird ein Konzept der Verkehrsstromanalyse zur systematischen Ermittlung des Mobilitätsbedarfs entwickelt.

4.1.2 Erhebungsziele

Folgende Ziele werden in Rahmen der Entwicklung dieses Konzeptes verfolgt:

- Entwicklung einer kostengünstigen, mobilen und skalierbaren Lösung zur automatisierten Erfassung des Verkehrsaufkommens
- Ermittlung der Verteilung des Transportaufkommens auf folgende Verkehrsteilnehmer: MIV, Radfahrer, Fußgänger
- Schaffung einer Datenbasis für die Wirksamkeitsüberprüfung der eingeführten Verkehrsmanagement-Maßnahmen sowie für die Optimierung der Mobilität

4.1.3 Aussagegüte

Da dieses Konzept eine kostengünstige Verkehrsstromanalyse bezweckt, ist die erwartete absolute Messgenauigkeit geringer als die von professionellen Verkehrserfassungsgeräten. Aus diesem Grund werden die Messergebnisse nicht absolut (z.B. zwischen 8:00 und 8:15 Uhr wurden *fünfzehn* Fußgänger erfasst),

sondern relativ (z.B. zwischen 8:00 und 8:15 Uhr war das Fußgängeraufkommen *gering*) dargestellt. Solch eine Ergebnisinterpretation erhöht die Aussagegüte und sorgt dafür, dass die verhältnismäßige Verkehrsverteilung möglichst vergleichbar abgebildet wird.

4.1.4 Datenschutz

Grundsätzlich ist bei jeder Verkehrserhebung ein Datenschutzkonzept mit einem Datenschutzbeauftragten zu erarbeiten bzw. mit ihm abzustimmen. Die folgenden Aussagen basieren auf der rechtlichen Grundlage der EU-Datenschutzgrundverordnung (EU-DSGVO) des Jahres 2018. Die Bestimmungen dieser Verordnung gelten laut Art. 2 Abs. 1 DSGVO für die automatisierte Verarbeitung personenbezogener Daten.

Dabei sind die personenbezogenen Daten nach Art. 4 Nr. 1 DSGVO „[...] alle Informationen, die sich auf eine identifizierte oder identifizierbare natürliche Person beziehen“. Die Personen sind mithilfe der Zuordnung zu einem Namen, Kennnummer, Standort und anderen Merkmalen identifizierbar. Alle Daten, die einer Person zugeordnet werden können, gelten als personenbezogen. Zu den personenbezogenen Daten, die im Rahmen der Verkehrserhebungen relevant sind, zählen Alter, Geschlecht, Wohnort, Pkw-Besitz, Kfz-Kennzeichen, Telefonnummer, Gesichter, Firmenzugehörigkeit (FGSV 2012b, S. 102).

Unter „Verarbeitung“ ist nach Art. 4 Nr. 2 DSGVO jeder ausgeführte Vorgang im Zusammenhang mit personenbezogenen Daten, wie Erheben, Erfassen, Speichern, Verwenden usw. zu verstehen.

Beim Einsatz der automatischen Verkehrserhebungsmethoden ist vor Beginn der Erhebung zu definieren, ob eine Verarbeitung der personenbezogenen Daten der erfassten Verkehrsteilnehmer stattfindet. Dies ist z.B. der Fall bei Videobeobachtungen sowie automatischen Kennzeichenerfassungen. Auch der Einsatz der Wi-Fi- und Bluetooth-Detektoren ist mit Datenschutzkonzept zu versehen. Dabei ist zu beachten, dass die MAC-Adressen unter Umständen als personenbezogene Daten angesehen werden können.

Die technischen Lösungen, die im Rahmen dieses Konzeptes beschrieben werden, konzentrieren sich auf verkehrstechnische Zählungen. Sie sind in der Lage lediglich Präsenz der Objekte festzustellen ohne Rückschlüsse auf die Personen zu ziehen, die diese Objekterkennung verursacht haben.

4.2 Erhebungsvorbereitung

4.2.1 Auswahl der Verkehrserfassungstechnologie

Für die systematische Auswahl von geeigneten Technologien der Verkehrsdatenerhebung wird eine Nutzwertanalyse durchgeführt. Dabei wird eine breite Vielfalt von technologischem Alternativen analysiert, um sie entsprechend den Wünschen und Anforderungen des Entscheidungsträgers zu priorisieren.

Dittmer (1995) definiert Nutzwertanalyse als eine effektive Planungsmethode, die der systematischen Entscheidungsvorbereitung durch Bewertung und Auswahl optimaler Lösungen dient. Diese Methode eignet sich besonders für komplexe Fragestellungen, deren Gesamtwert (Gesamtnutzen) aus mehreren Teilen besteht, die bei Entscheidungsfindung zu betrachten sind. Dabei können sowohl objektive als auch subjektive Informationen erfasst werden. (Dittmer 1995, S. 43)

Die Analyse erfolgt anhand der im Vorfeld definierten entscheidungsrelevanten Größen - Kriterien (s. Abschnitt 3.3.2). Die Kriteriendefinition ist vor Priorisierung durchzuführen, um die Objektivität der Nutzwertanalyse zu gewährleisten. Dabei sind die Technologien entsprechend dem Erfüllungsgrad der definierten Kriterien zu bewerten. Da die meisten Kriterien qualitativ sind, ist für den Vergleich eine Wertetabelle zu definieren, die den Erfüllungsgrad abbildet. Im Abschnitt 3.3.14 wurde die Eignung der jeweiligen Technologie mit -, 0 und + bewertet, was für *nicht geeignet*, *bedingt geeignet* und *gut geeignet* steht. Im Rahmen der Nutzwertanalyse wird folgendes Bewertungssystem angewendet: 0 – nicht geeignet, 1 – bedingt geeignet und 2 – gut geeignet.

Als Nächstes ist die Priorisierung durchzuführen. Dies geschieht durch Gewichtung der Kriterien von 1% bis 100%. Dabei soll die Summe aller Gewichtungen 100% betragen. Die Gewichtung der WISTA Management GmbH ist in Tabelle 8 dargestellt.

Tabelle 8: Gewichtung der Auswahlkriterien

<i>Kriterien</i>	<i>Gewichtung</i>
Genauigkeit	10,00%
Zuverlässigkeit	10,00%
Vollständigkeit	10,00%
Montage- & Installationsaufwand	7,50%
Wartungsintensität	10,00%
Energieverbrauch	7,50%
Reichweite	2,50%
Erfassung mehrerer Fahrstreifen	2,00%
witterungsabhängige Einflüsse	7,50%
Diskretion	5,00%
Verschmutzung	6,00%
Datenschutz	10,00%
Kosten	12,00%
100,00%	

Als Nächstes werden die Ergebnisse durch Multiplikation der Punktwerte (0 bis 2) mit der Kriteriengewichtung und anschließender Addition der Ergebnisse für jede Alternative ermittelt. In Excel wird dafür die Funktion *SUMMENPRODUKT(Gewichtung-Array;Punktwerte-Array)* benutzt.

Für die bessere Vergleichbarkeit werden die Ergebnisse normiert. Das Endergebnis stellt dar, zu wie viel Prozent eine Technologie die Anforderungen erfüllt. Je höher dieser Wert ist, desto besser eignet sich die Technologie für die untersuchte Situation. Da die Erhebung der einzelnen Verkehrsteilnehmer getrennt erfolgen soll, werden drei Verkehrserfassungssysteme konzipiert. Laut der durchgeföhrten Nutzwertanalyse (s. Tabelle 9) eignen sich folgende Technologien für den Einsatz im Technologiepark Adlershof: Druckdetektor, Radardetektor sowie Ultraschalldetektor, da sie den höchsten Zielerreichungsgrad aufweisen.

Tabelle 9: Ergebnisse der Nutzwertanalyse

Kriterien	Gewichtung	Induktionsschleife	Magnetfelddetektor	Druckdetektor	Videoerfassung	Infrarotdetektor	Laserdetektor	Ultraschalldetektor	Radar detektor	Kombinationsdetektor	Wi-Fi, Bluetooth-Detektor
Genauigkeit	10,00%	2	2	2	1	1	1	1	2	2	1
Zuverlässigkeit	10,00%	2	2	1	2	1	2	1	1	1	1
Vollständigkeit	10,00%	1	1	1	1	1	2	2	1	2	0
Montage- & Installationsaufwand	7,50%	0	0	2	2	2	2	2	2	2	2
Wartungsintensität	10,00%	2	2	2	0	1	1	1	1	1	0
Energieverbrauch	7,50%	2	2	2	0	2	0	2	2	1	0
Reichweite	2,50%	0	0	0	2	2	2	1	2	1	2
Erfassung mehrerer Fahrstreifen	2,00%	0	0	0	2	0	2	0	0	2	0
witterungsabhängige Einflüsse	7,50%	2	2	2	0	0	1	1	1	1	1
Diskretion	5,00%	2	2	1	1	1	1	1	1	1	1
Verschmutzung	6,00%	2	2	2	0	1	0	1	1	1	1
Datenschutz	10,00%	2	2	2	0	2	2	2	2	2	0
Kosten	12,00%	0	0	1	0	1	0	2	2	1	0
	100,00%	1,42	1,42	1,54	0,69	1,18	1,17	1,45	1,48	1,40	0,59
		0,71	0,71	0,77	0,35	0,59	0,58	0,73	0,74	0,70	0,29

Dabei wird der Druckdetektor für die Erfassung von Radfahrerzahlen auf den Fahrradwegen eingesetzt, der Radar detektor für die Erhebung der MIV-Mengen auf der Fahrbahn und der Ultraschalldetektor für die Erkennung der Fußgänger auf den Gehwegen.

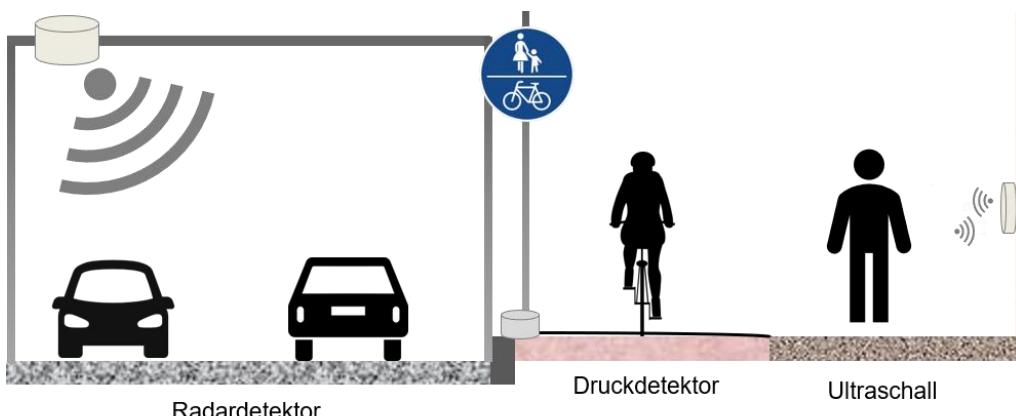


Abbildung 13: Konzept Verkehrserfassungssysteme Adlershof

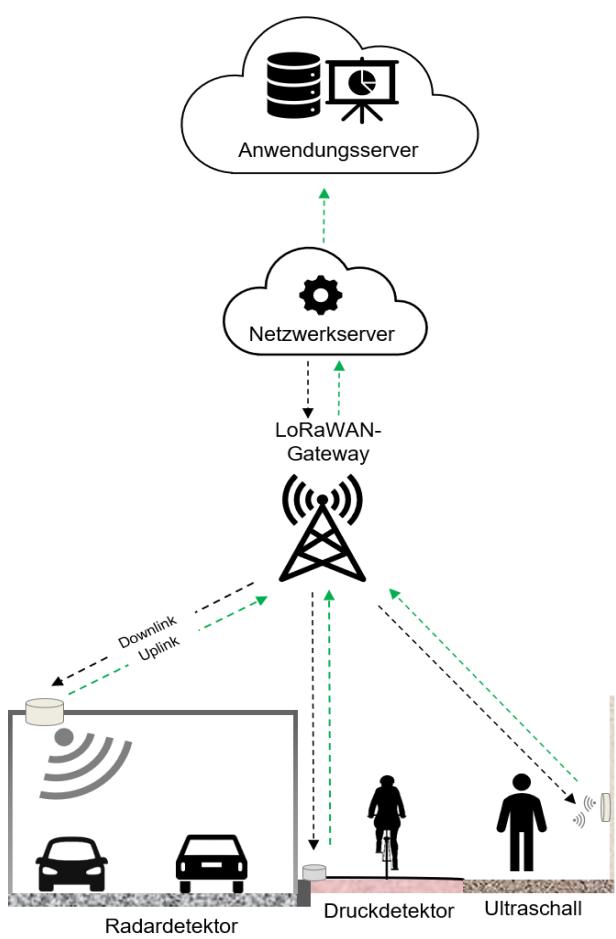
4.2.2 Übertragungstechnologie

Speziell für die Datenübertragung bei IoT-Lösungen wurde Low Power Wide Area Network (LPWAN-Netz) entwickelt. Ein LPWAN-Netz besteht aus mehreren Sensoren und Aktoren – Nodes, welche einen ständigen Datenaustausch mit einer Basisstation – Gateway betreiben. Die Nodes verfügen über eine Antenne, welche die Kommunikation auf niederfrequenten Funkwellen ermöglicht. Das Gateway überträgt die gesammelten Daten z.B. über das Internet an das Backendsystem und dient somit als Schnittstelle zwischen der reellen und der virtuellen Welt. (Linnemann et al. 2019, S. 11)

Es gibt mehrere Anbieter für LPWAN-Netze: LoRaWAN, Sig Fox, NB-IoT der Telekom u.a. Da jeder Anbieter eigene Vor- und Nachteile hat, sind die LPWAN-Netze anhand Kriterien aus unterschiedlichen Bereichen zu vergleichen. So ist bei der Auswahl auf *Infrastrukturanforderungen* (Reichweite, Durchdringung, Datenrate usw.), *Hardwareanforderungen* (Energieeffizienz, Ökosystem, Produktqualität), *IT-Anforderungen* (vertikale Integrationstiefe, IT-Sicherheit, Verfügbarkeit usw.) und *betriebswirtschaftliche Anforderungen* (Kosten, Lizenzmodell, horizontale Integrationstiefe) zu achten. Linnemann et al. (2019) vergleichen die Netze mehrerer Anbieter anhand dieser Eigenschaften und stellen eine Basis für die Entscheidung über die geeignete Technologie für die Datenübertragung dar. (Linnemann et al. 2019, S. 13–23)

LoRaWAN hebt sich von anderen LPWAN-Netzen durch das offene Lizenzmodell hervor. Das ermöglicht den Benutzern das Netz selbstständig aufzubauen. Durch diesen Ansatz verfügt LoRaWAN über das größte Ökosystem an Hardware sowie über hohe Mitgliederanzahl. LoRaWAN setzt auf international anerkannte Protokolle (M2M, CoAP und MQTT), was dessen Einsatz noch attraktiver macht. Außerdem lassen sich die Geräte ohne großen Aufwand ins Netz einbinden. (Linnemann et al. 2019, S. 18–23)

Der Technologiepark Adlershof verfügt über mehrere LoRaWAN-Gateways, wodurch die Netzbdeckung im ganzen Gebiet gewährleistet wird. Das Long Range Wide Area Network (LoRaWAN) zeichnet sich durch hohe Energieeffizienz und große Reichweite. In ländlichen Gebieten kann die Reichweite dieser Technologie bis 15 km betragen, während sie im urbanen Raum auf 2 bis 4 km reduziert ist. (Linnemann et al. 2019, S. 28)



Die LoRaWAN-Gateways übertragen die empfangenen Datenpakete an einen Netzwerkserver, welcher diese Daten anschließend an den Anwendungsserver weiterleitet.

Außerdem ist der Netzwerkserver für die Datenverarbeitung zuständig. Er entschlüsselt die empfangenen Nachrichten, entfernt die Dopplungen, sendet die Downlink-Nachrichten an die Endknoten (IoT-Geräte) und regelt alle Prozesse, die damit zusammenhängen.

Der Anwendungsserver enthält die eigentliche IoT-Anwendung. Er ist an den Netzwerkserver angebunden und realisiert die Datenverwaltung, Datenvisualisierung sowie Kommunikation zwischen unterschiedlichen Services.

Abbildung 14: Datenübertragung per LoRaWAN

Bei der Entwicklung der eigenen LoRaWAN-Architektur kann auf offene Netzwerke (z.B. The Things Network, kurz *ttn*) oder kommerzielle Lösungen (z.B. *thingsHub*) zugegriffen werden.

Im Folgenden wird die Vorgehensweise zum Aufbau einer *ttn*-basierten Architektur erläutert (siehe Abbildung 15).

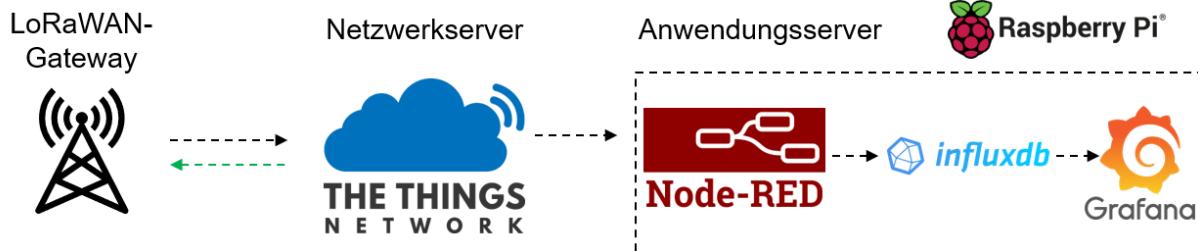


Abbildung 15: Architektur einer *ttn*-Anwendung

Quellen: The Things Network Logo; Node-RED Logo; influxDB Logo; Raspberry Pi Logo; Grafana Logo

Der Netzwerkserver wird von *The Things Network* zur Verfügung gestellt. In dieser cloudbasierten Anwendung soll jedes IoT-Gerät registriert werden, das mit dem Gateway kommuniziert. Jedem Gerät werden spezielle Schlüssel zugewiesen, die in

der Geräte-Firmware festzuhalten sind, sodass jeder Detektor eindeutig identifizierbar ist.

In der Anwendung erfolgt außerdem die Entschlüsselung des empfangenen Payloads. Das geschieht mithilfe eines Decoders, der direkt in der Anwendung programmiert werden kann. Die empfangenen und entschlüsselten Daten werden in tabellarischer Form angezeigt, jedoch nicht gespeichert.

Die Datenverwaltung wird vom Anwendungsserver übernommen. Er kann in verschiedenen Weisen aufgebaut werden und aus unterschiedlichen Elementen bestehen. Die in diesem Konzept vorgeschlagene Lösung besteht aus *NodeRED* – Werkzeug zum Auslesen der Daten aus der *ttn* und deren Weiterleitung an die *influxdb* – Datenbankmanagementsystem für die dauerhafte Datenspeicherung, die als Datenquelle für den Visualisierungstool *Grafana* dient. *Node-RED* und *influxdb* sollen kontinuierlich ausgeführt sein, damit die von IoT-Geräten gesendeten Daten lückenlos gespeichert werden können. Da dies mit hohem Ressourcen- und Energiebedarf verbunden ist, kann der Anwendungsserver auf einem *Raspberry Pi* – Einplatinencomputer betrieben werden. Die Installation sowie Konfiguration der einzelnen Anwendungen auf einem *Raspberry Pi*-Computer unterscheiden sich nur minimal von den gewöhnlichen, weswegen er sich besonders gut als Anwendungsserver für die IoT-Applikationen eignet.

Als Alternative kann auf die fertigen kommerziellen Lösungen zugegriffen werden.

Eine solcher Lösungen ist *thingsHub* – eine IoT-Plattform, die den Benutzern den Zugang zum Netzwerk- und Anwendungsserver zur Verfügung stellt (siehe Abbildung 16).

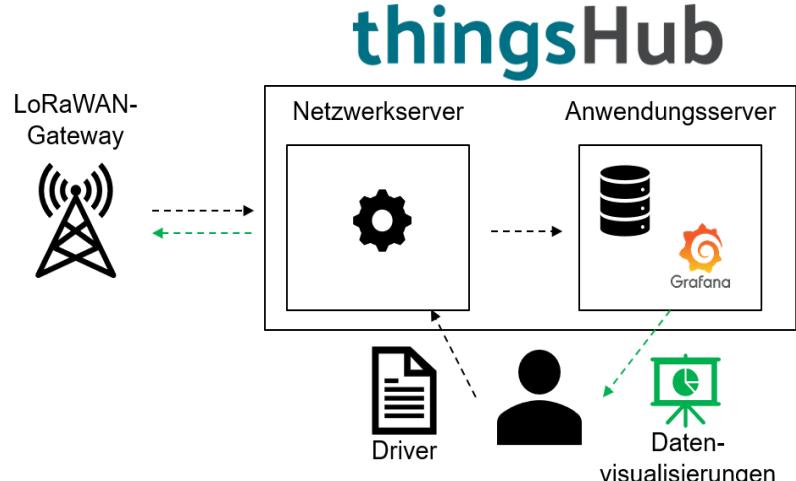


Abbildung 16: Architektur des thingsHub

Quellen: thingsHub Logo; Grafana Logo

Der Benutzer registriert seine Geräte auf dem Netzwerkserver und implementiert einen *Driver* - eine Anwendung, die die Datenstruktur des Payloads definiert. Im Gegenteil zum *ttn* hat der Benutzer keinen Zugang zu den einzelnen Bauteilen der Plattform. Mit dem Anwendungsserver interagiert der Benutzer durch Datenabfrage und Konfiguration der Visualisierungen in *Grafana*. Außerdem ermöglicht *thingsHub* den

Datenimport und -export und hat eine REST API-Schnittstelle zur Datenintegration in eigene Anwendungen.

Die im Rahmen dieses Konzeptes entwickelten IoT-Lösungen werden auf der Plattform *thingsHub* registriert und verwaltet.

4.2.3 Zeit und Dauer

Das vorliegende Konzept beabsichtigt einen *langfristigen* Einsatz von ausgewählten Verkehrserfassungstechnologien. Aus den erhobenen Daten ist die *durchschnittliche werktägliche Verkehrsstärke (DTV_w)* zu ermitteln.

Um die Verteilung des Verkehrsaufkommens über den Tag zu ermitteln, ist die von IoT-Geräten erfasste aggregierte Anzahl der Fahrzeuge (MIV), Radfahrer sowie Fußgänger *alle 15 Minuten* an den Netzwerk-Server zu übertragen. Nach der Übertragung soll die Zählung von Null angefangen werden.

Die automatischen Zählungen sind für den *Tagesverkehr an Wochentagen (6:00 bis 22:00 Uhr)* durchzuführen. Die 24/7-Erfassung kann bei Möglichkeit durchgeführt werden.

Es ist damit zu rechnen, dass die Datenerhebung *über mehrere Jahre* stattfinden soll, um den aktuellen Status des Mobilitätsverhaltes zu ermitteln, nach der Umsetzung von Maßnahmen zur Verbesserung des deren Effektivität zu quantifizieren und final die Zielerreichung der Maßnahme zu bewerten. Maßnahmen zur Verbesserung der Mobilität in Adlershof werden in einem gesonderten Projekt der WISTA Management GmbH erarbeitet und sind nicht Teil der vorliegenden Arbeit.

4.2.4 Räumliche Abgrenzung

Die in diesem Konzept beschriebenen IoT-Lösungen eignen sich ausschließlich für Standorte, wo eine klare räumliche Trennung zwischen den Verkehrsteilnehmern möglich ist. Die Entscheidung über Standorte der Anbringung von Verkehrszählgeräten ist abhängig von geplanten Mobilitätsmaßnahmen zu treffen.

Zunächst wurden für die künftige Geräteinstallation die Ein- sowie Ausfahrten aus dem Technologiepark Adlershof ausgewählt.



Abbildung 17: Neuralgische Knotenpunkte für Verkehrsstromanalyse

Die Verkehrsmengen auf diesen Kreuzungen wurden bei DLR Anfang 2019 für das Jahr 2030 simuliert. Um die Ergebnisse der Simulation mit der aktuellen Verkehrsmengenentwicklung zu vergleichen, sind die Verkehrsmessungen zuerst auf diesen Knotenpunkten durchzuführen. Dabei handelt es sich um folgende Knotenpunkte:

Tabelle 10: Neuralgische Knotenpunkte

Nr.	Standort / Kreuzung
K1	Adlergestell / Dörpfeldstraße (S Adlershof)
K2	Adlergestell / Köpenicker Straße / Glienicker Weg
K3	Ernst-Ruska-Ufer / Wegedorfstraße
K4	Groß-Berliner-Damm / Igo-Etrich-Straße / Herrmann-Dorner-Allee
K5	Köpenicker Straße / Ernst-Ruska-Ufer
K6	S Schöneweide

4.3 Vorbereitung der Datenerhebung

4.3.1 Montage & Inbetriebnahme

Fußgängerzähler

Der Ultraschalldetektor ist seitlich anzubringen. Dabei kann das Gerät an der Wand, am Laternen- oder Signalmast usw. montiert werden.

Die Installationshöhe sollte 130-150 cm betragen, um Erkennung von Fußgänger unabhängig von ihrer Größe sicherzustellen. Optimal ist es, wenn die Breite des Fußgängerweges die maximale Reichweite des Ultraschallsensors nicht überschreitet. In diesem Fall ist die Erkennungszone als der Bereich zwischen dem Sensor und der Grenze zwischen dem Fußgänger- und dem Fahrradweg definiert. Die Objekte, die außerhalb dieser Erkennungszone erfasst werden, sind nicht mitzuzählen.

Bei sehr breiten Fußgängerwegen können zwei Fußgängerzähler auf gegenüberliegenden Seiten des Fußgängerweges installiert werden (s. Abbildung 19). Dabei sollen die Erkennungszonen der einzelnen Sensoren voneinander getrennt werden und sich nicht überschneiden.

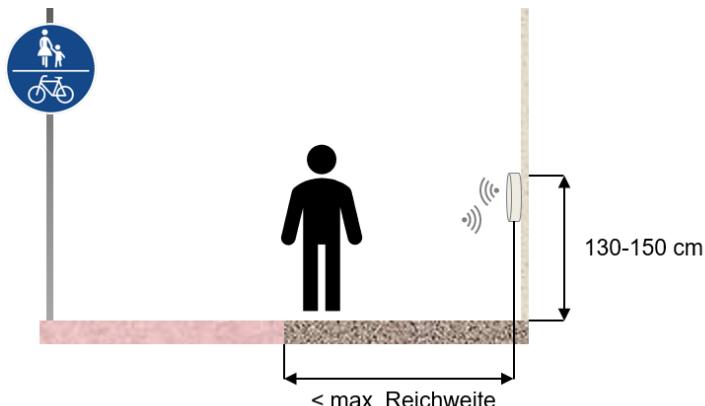


Abbildung 18: Montage Ultraschalldetektor

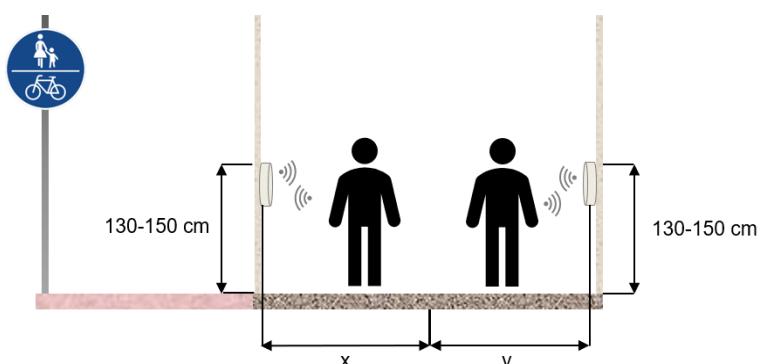


Abbildung 19: Montage Ultraschalldetektor, Variante 2

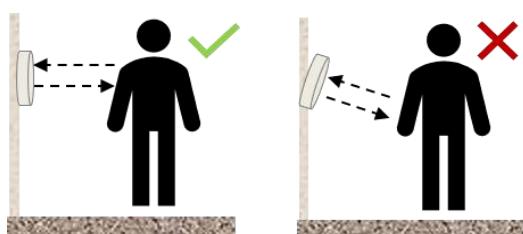


Abbildung 20: Ausrichtung Ultraschallsensor

Der Ultraschallsensor ist so zu auszurichten, dass die Signale parallel zur Gehwegoberfläche gesendet und empfangen werden (s. Abbildung 20).

Fahrradzähler

Der Druckdetektor besteht aus zwei Hauptbestandteilen: einer Elektronik-einheit und einem Druckschlauch.

Die Elektronikeinheit (Mikrokontroller, Drucksensor, Verbindungs- und Stromkabel) ist in ein Gehäuse einzubauen, welches in der unmittelbaren Nähe des Fahrradweges zu montieren ist.

Der Druckschlauch ist über die ganze Breite des Fahrradweges mit einem entsprechenden Klebeband an mehreren Stellen zu befestigen. Die Klebebandstreifen dürfen keinen Druck auf den Schlauch ausüben.

MIV-Zähler

Um die komplette Fahrbahn zu überwachen, ist der Radarsensor überkopf zu montieren. Die Installation auf einer Brücke (z.B. durch Befestigung am Brückengeländer) eignet sich dafür am besten.

Abhängig davon, welcher Radarsensor eingesetzt wird, ist zu unterscheiden, ob die Messung fahrspurgetrennt oder fahrbahn-übergreifend stattfindet. Für höhere Genauigkeiten ist die erste Variante zu bevorzugen. In diesem Fall soll für jede Fahrspur ein Radardetektor eingerichtet werden. Die Reichweite der Sensoren ist so einzustellen, dass die Erkennungszonen der benachbarten Radardetektoren sich nicht überschneiden.

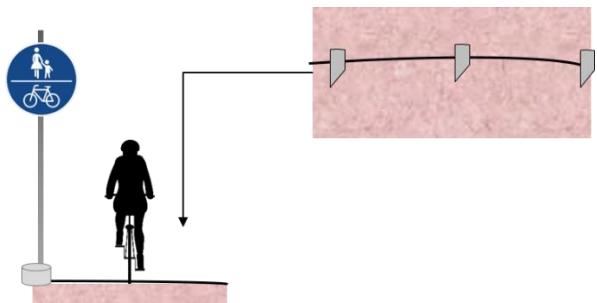


Abbildung 21: Montage Druckdetektor

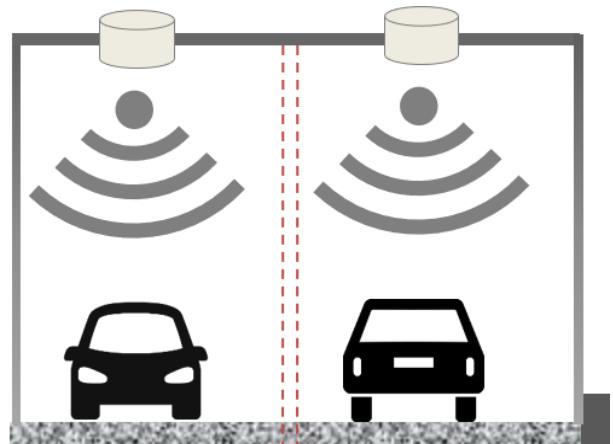


Abbildung 22: Montage Radardetektor

4.3.2 Witterungsschutz

Für den Schutz gegen Umwelteinflüsse wie Niederschlag, Staub, Feuchtigkeit usw. sind die Detektoren in ein Gehäuse einzubauen. Im Folgenden werden Gehäuse-Konzepte für die einzelnen Geräte dargestellt. Sie werden allgemein gehalten und sind beliebig anpassbar bzw. erweiterbar.

Fußgängerzähler

Für den Ultraschalldetektor eignet sich ein witterfestes Outdoor-Wandgehäuse. Dieses ist mit zwei runden Öffnungen für den Ultraschallsensor zu versehen, da für die Messung ein freies Sichtfeld erforderlich ist. Um Schutz vor Niederschlag zu gewährleisten, ist über den Öffnungen eine Überdachung zu montieren. Sollte sich die Stromquelle außerhalb des Gehäuses befinden, ist auf der passenden Stelle eine Öffnung für das Stromkabel einzubauen. Das Gehäuse soll über Befestigungsmöglichkeiten an der Wand verfügen. Die Größe des Gehäuses ist in Abhängigkeit der Geräteabmessungen zu wählen.

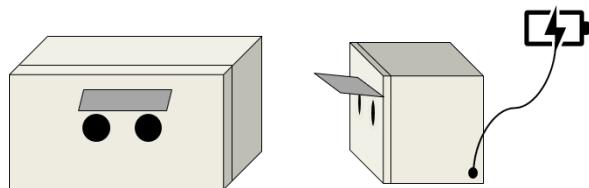


Abbildung 23: Konzept Gehäuse Fußgängerzähler

Fahrradzähler

Die Elektronikeinheit des Fahrradzählers soll sich in einem verschließbaren witterfesten Outdoor-Gehäuse befinden. Die Art des Gehäuses ist in Abhängigkeit der Anbringungsart zu wählen. Die Box soll über zwei Öffnungen verfügen: eine für den Stromanschluss, sobald das Gerät von einer externen Stromquelle versorgt wird und eine für den Druckschlauch, der über den Fahrradweg verlegt werden soll. Diese Öffnungen sind mit den Kabelverschraubungen auszustatten, die für eine Dichtung des Kabels bzw. des Schlauchs notwendig sind.

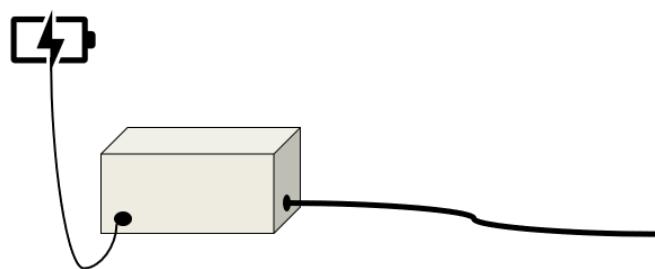


Abbildung 24: Konzept Gehäuse Fahrradzähler

MIV-Zähler

Da der Radarsensor direkt über einem bestimmten Fahrstreifen positioniert werden soll, kann er getrennt von der restlichen Elektronik in einem eigenen Gehäuse installiert werden. Entsprechend den Abmessungen und Eigenschaften des Sensors ist dieses witterfeste Radar-Gehäuse mit einer Öffnung für den Sensor sowie einer Kabelöffnung zu versehen. Die restliche Elektronik soll sich in einem eigenen Gehäuse

befinden. Es soll über zwei Kabelöffnungen verfügen, um die Verbindung mit dem Radarsensor und ggf. die externe Stromversorgung zu gewährleisten.

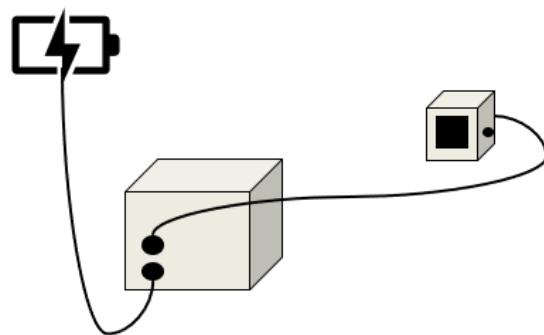


Abbildung 25: Konzept Gehäuse MIV-Zähler

4.4 Datenaufnahme und -übertragung

4.4.1 Datenaufnahme

Der allgemeine Prozess der Datenaufnahme ist vereinfacht in Abbildung 26 dargestellt.

Fußgängerzähler

Der Ultraschallsensor ermittelt den Abstand zum erkannten Objekt und liefert den Wert als Ergebnis der Messung. Um das Messergebnis zu interpretieren, ist ein Schwellenwert zu definieren. Dieser Schwellenwert repräsentiert die Grenze des Fußgängerweges. Alle Objekte, die sich hinter dieser Grenze befinden, werden nicht als Fußgänger gezählt. Jedes erkannte Objekt, das sich innerhalb der Erkennungszone befindet, erhöht den Fußgängerzähler um 1.

Die Fußgärtzahl ist nach Ablauf des 15-minütigen-Timers an das LoRaWAN-Gateway zu übertragen. Nach der erfolgreichen Übertragung wird der Timer neugestartet und die Zählung fängt bei Null an.

Fahrradzähler

Der Fahrradzähler misst den im Schlauch herrschenden Druck. Sobald ein Fahrrad über den Schlauch fährt, wird der Druck erhöht. Diese Änderung wird durch den Drucksensor aufgenommen.

Dafür soll zunächst der Normaldruck definiert werden. Wenn dieser überschritten wird, wird davon ausgegangen, dass in diesem Moment ein Fahrrad über dem Schlauch fuhr.

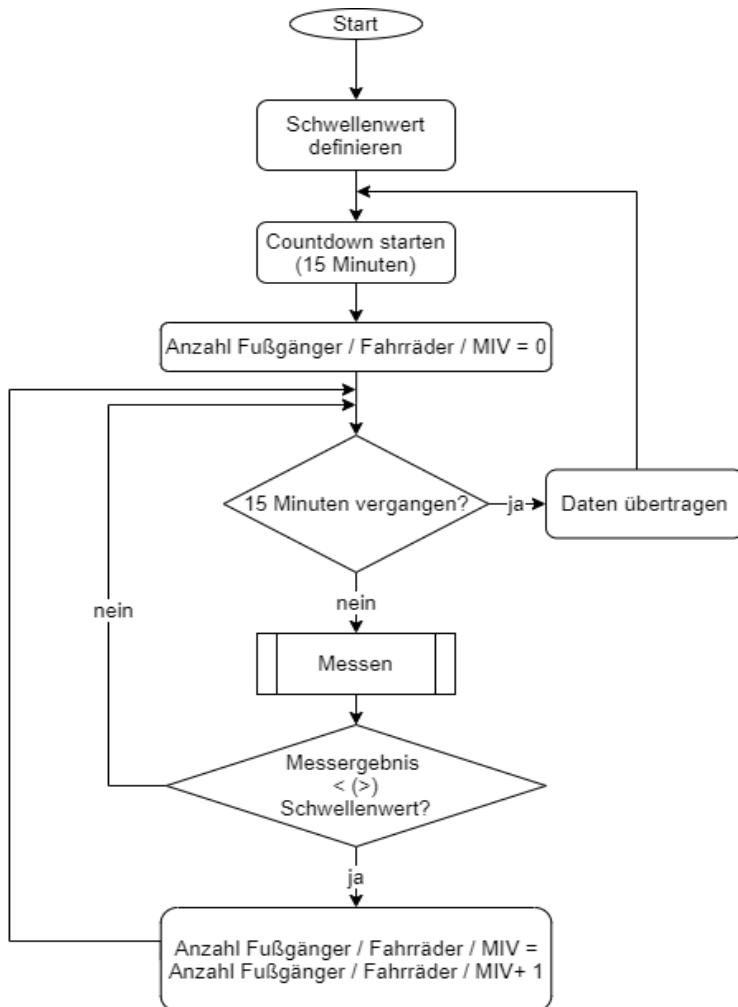


Abbildung 26: Datenaufnahme Fußgängerzähler

MIV-Zähler

Der Radarsensor erkennt die Geschwindigkeit der sich bewegenden Fahrzeuge. Es ist für jeden Standort eine Mindestgeschwindigkeit zu definieren, die ein sich bewegendes Objekt aufweisen muss, um als MIV klassifiziert zu werden. Sollte eine Überschreitung dieser Mindestgeschwindigkeit erfasst werden, ist die Anzahl der Fahrzeuge um „Eins“ zu erhöhen.

4.4.2 Datenbereitstellung

Die von Detektoren gesendete Daten sind in einer Datenbank zu speichern. Die Abbildung 27 stellt das Schema der Datenbereitstellung sowie die Datenstruktur dar.

Die IoT-Geräte sollen alle 15 Minuten die erfasste Anzahl der Fußgänger, Fahrräder oder MIV-Fahrzeuge übertragen.

Außerdem ist mit diesen Informationen die aktuelle Sensortemperatur zu senden. Sollte die Sensortemperatur außerhalb des zulässigen Bereichs (dieser ist im Voraus zu definieren, z.B. 20-50 °C) liegen, deutet das auf einen möglichen Defekt hin.

Die Anzahl, Temperatur sowie der vom Gateway übertragene Zeitstempel sind die Pflichtfelder, ohne die kein Tabelleneintrag möglich sein soll.

Da der Akkustand sich nicht alle 15 Minuten ändert, ist diese Information einmal täglich oder auf Nachfrage zu übertragen.

Zusätzlich zu den Gerätedaten soll es möglich sein die Umgebungsdaten zu speichern. Ein Benutzer soll in der Lage sein einen bestehenden Datensatz durch Zusatzinformationen wie Unfall, Bauarbeiten, Feiertag oder Unwetter zu ergänzen. Jede dieser Informationen entspricht einer booleschen Variablen, die standardmäßig auf *false* gesetzt wird. Meldet der Benutzer z.B. einen Unfall, sind alle künftige Datensätze mit *accident = true* zu versehen. Wenn der Unfall beendet ist, meldet der Benutzer dies im System und die Eigenschaft *accident* wird für die kommenden Datensätze auf *false* gesetzt.

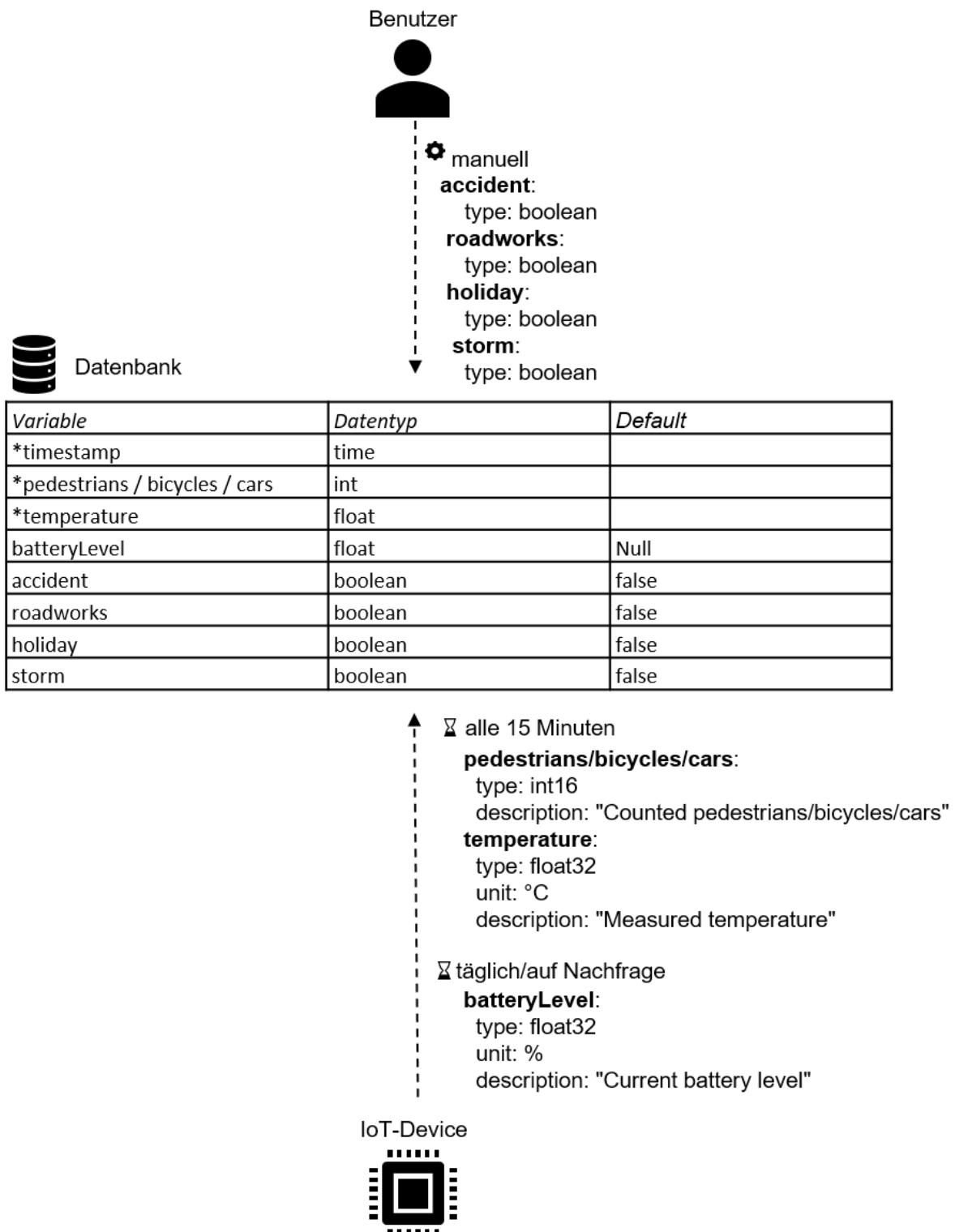


Abbildung 27: Schema Datenbereitstellung

4.5 Datenaufbereitung und -analyse

4.5.1 Datenaufbereitung

Um die Qualität der erhobenen Daten sowie deren Plausibilität sicherzustellen, sind folgende Prüfungen durchzuführen.

Automatische Vollständigkeitsprüfung

Hierbei ist sicherzustellen, dass die erhobenen Daten keine Lücken aufweisen. Diese Prüfung soll automatisch durchgeführt werden. Sollten lückenhafte Datensätze auftreten, sind die Gründe für die fehlenden Messungen zu untersuchen und zu dokumentieren.

Vergleich mit Ergebnissen der Befragung

Wurden im Vorfeld der Verkehrszählungen sonstige Verkehrserhebungen (z.B. Befragungen) durchgeführt, können diese zur Validierung von Ergebnissen der automatischen Zählungen genutzt werden. So wird die Plausibilität der Verkehrserfassung im Technologiepark Adlershof anhand der Ergebnisse von DLR-Befragung geprüft. Dabei werden keine absoluten Werte verglichen, sondern die Verläufe der Tagesganglinien sowie die Verteilung der Verkehrsmittelnutzung.

Dabei ist zu beachten, dass die Befragung im Jahr 2018 stattgefunden hat und wird langfristig an Aktualität verlieren. Die regelmäßige (z.B. jährliche) Durchführung solcher Befragungen kann eine hochwertige Plausibilitätskontrolle der automatischen Zählstellen gewährleisten.

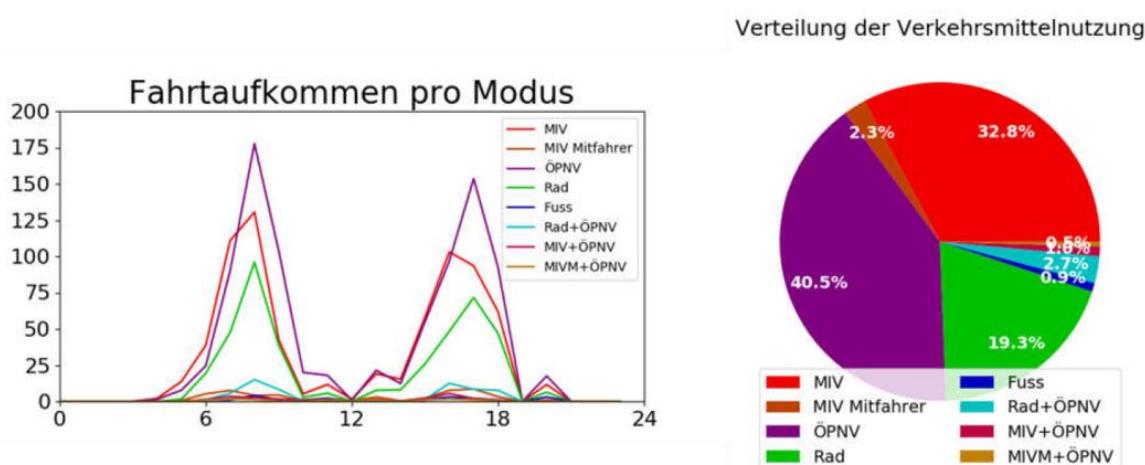


Abbildung 28: DLR-Befragung: Fahrtaufkommen pro Modus; Modal Split

Quelle: Krajzewicz et al. 2018, S. 18–20

Vergleich mit benachbarten Zählstellen

Sollten mehrere Zählstellen nebeneinander eingerichtet werden, ist es sinnvoll, deren Ergebnisse miteinander zu vergleichen. So können mögliche Defizite der einzelnen Geräte erkannt und Maßnahmen frühzeitig ergriffen werden.

Besonders wertvoll ist der Vergleich mit Zählstellen anderer Technologien (falls verfügbar). So können z.B. die Radar-Zählungen des MIV durch Zählungen der Induktionsschleifen oder Videoerfassungssysteme validiert werden.

Manuelle Erhebung

Eine weitere Möglichkeit zur Qualitätskontrolle ist die manuelle Erhebung, die parallel zur automatischen Erhebung durchzuführen ist. Dabei sind Zählblätter oder die geeignete Software-Lösungen zu benutzen. Wichtig ist, die genaue Erhebungszeit sowie den Bezug auf die kontrollierte Zählstelle zu dokumentieren.

Dokumentation der Erhebungsumstände

Außer der eigentlichen Erhebungszahlen sind die Erhebungsumstände zu dokumentieren. Dies wird bei der späteren Datenauswertung behilflich sein. Folgende Umstände sollen erfasst werden:

- Unfall
- Baustelle
- Feiertag
- Unwetter

Dabei sind immer die betroffenen Zählstellen anzugeben sowie die Uhrzeit und die Dauer dieser Umstände zu erfassen.

4.5.2 Datenvisualisierung

Die Daten sind mithilfe einer Software (z.B. Grafana) zu visualisieren. Die Visualisierungen sind möglichst interaktiv zu gestalten, sodass das Zeitfenster flexibel wählbar ist.

Für die Darstellung des Fahrtaufkommens eignet sich eine Graph-Visualisierung (s. Abbildung 29).

So werden die Verkehrsmengen (Y-Achse) der jeweiligen Verkehrsteilnehmer zu bestimmten Zeiten (X-Achse) angezeigt. Für höhere Aussagegüte sind die Verkehrsmengen der einzelnen Verkehrsteilnehmer getrennt darzustellen. Dabei sind keine absoluten Zahlen, sondern eine relative Auslastung abzubilden (gering/moderat/hoch). Die Schwellenwerte sind für jeden Verkehrsmodus separat zu definieren. Die Auslastung kann entsprechend farblich markiert werden (s. Abbildung 30)

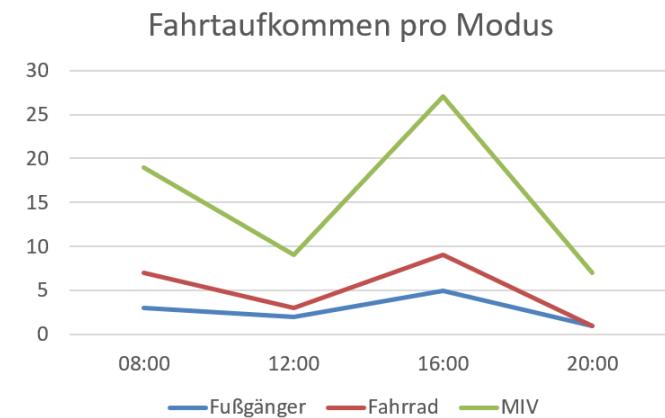


Abbildung 29: Visualisierung Fahrtaufkommen

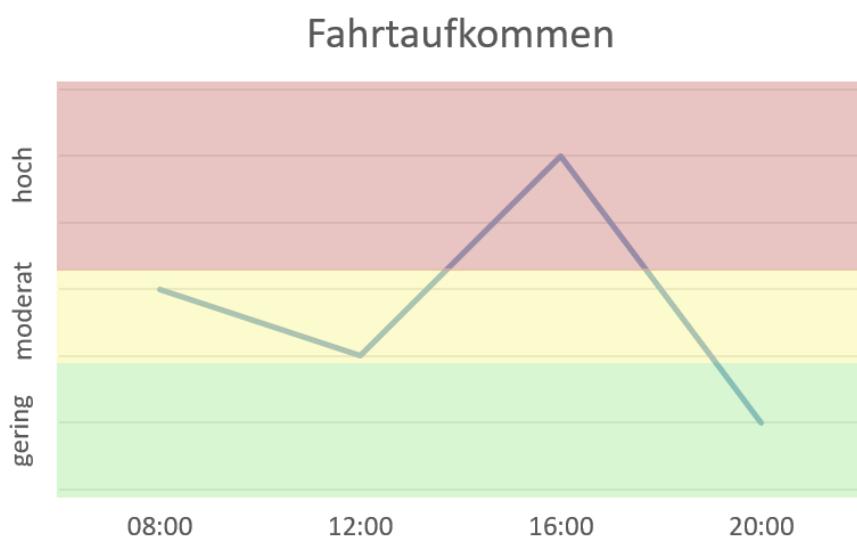


Abbildung 30: Visualisierung Fahrtaufkommen relativ

Als Nächstes ist die Verteilung des Fahrtaufkommens auf verschiedene Verkehrsmittel darzustellen (Modal Split). Diese Verteilung lässt sich am besten mithilfe eines Kreis-Diagramms abbilden.

Dabei sind die prozentuellen Anteile der Verkehrsmittel mit unterschiedlichen Farben hervorzuheben.

Sollte die Stromversorgung der Detektoren mithilfe von Akkus oder Batterien gewährleistet werden, ist deren aktueller Status regelmäßig zu überprüfen (s. Abbildung 32).

Dafür kann ein Balkendiagramm eingesetzt werden, das einen Überblick über den Akkustand aller Detektoren liefert. Für eine schnelle Auswertung können die einzelnen Balken farblich markiert werden. Bei einem Akkustand ab 20% bleibt der Balken grün, zwischen 10% und 20% soll der Balken gelb werden und der kritische Akkustand unter 10% ist mit der roten Farbe zu signalisieren. Dabei können die einzelnen Schwellenwerte angepasst werden.

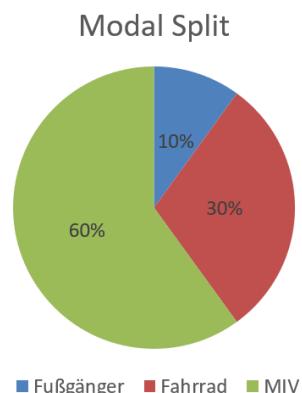


Abbildung 31: Visualisierung Modal Split

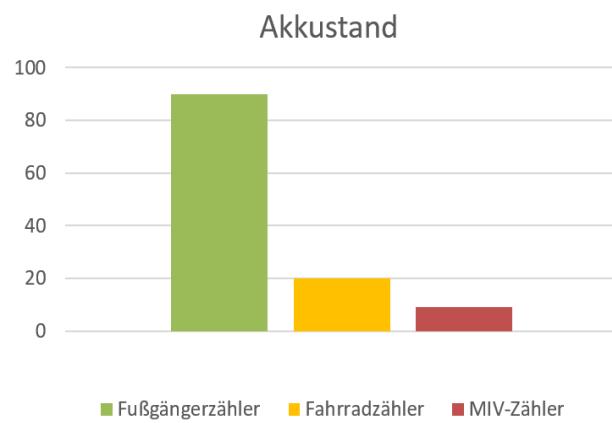


Abbildung 32: Visualisierung Akkustand

5 Umsetzung

5.1 Allgemein

5.1.1 Standort

Die prototypische Umsetzung des oben beschriebenen Konzeptes findet in Kooperation mit dem DLR, Institut für Verkehrssystemtechnik statt. Zum ersten Teststandort wurde eine der beiden Messbrücken am Ernst-Ruska-Ufer in Adlershof ausgewählt. Diese Messstation verfügt über Induktionsschleifen, Infrarotsensoren und Kameras, die MIV-Verkehrsdaten der vier Fahrstreifen erheben.



Abbildung 33: DLR Messbrücke

Der Standort eignet sich für den prototypischen Einsatz der IoT-Geräte aus folgenden Gründen:

- klare räumliche Trennung zwischen den Verkehrsteilnehmern (s. Abbildung 33)
- Integration ins vorhandene Sensornetz mit wenig Aufwand
- Nähe zum LoRaWAN-Gateway
- direkter Vergleich mit anderen Verkehrserfassungstechnologien möglich
- geringere Vandalismusgefahr, da keine Hauptstraße und wegen Präsenz anderer Verkehrserhebungstechnologien weniger auffällig

Die Messbrücke befindet sich in der unmittelbaren Autobahnnähe und ist für MIV- sowie Fahrradfahrer attraktiv. Fußgängerzahlen sind in diesem Gebiet unterrepräsentiert. Aus diesem Grund eignet sich der Standort nicht für die Ermittlung des Modal Splits. Hier werden die prototypischen Detektoren angebracht, um ihre Funktionalitäten umfassend zu testen und potenzielle Fehlerquellen zu identifizieren.

5.1.2 Stromversorgung

Für die prototypische Umsetzung kamen zwei Methoden der Stromversorgung infrage: mobil durch Powerbank oder stationär am Stromkasten bzw. direkt an der Messbrücke. Da die Messstation am Ernst-Ruska-Ufer über zahlreiche stationäre Stromquellen verfügt, wird der Radar- und Druckdetektor von der Stromquelle an der Brücke mit Elektroenergie versorgt. Der Ultraschalldetektor wird an eine Steckdose am Stromkasten angeschlossen, an dem er montiert ist.

5.1.3 Schutz vor Vandalismus

Die erste Maßnahme für Schutz vor Vandalismus ist die bereits beschriebene Standortwahl. Die Anzahl der Passanten in diesem Gebiet ist sehr gering, was die Vandalismusgefahr deutlich senkt.

Weitere Aspekte des Schutzes gegen Vandalismus sind das Gehäuse und die Montage. Dabei ist ein Gehäuse zu verwenden, welches robust gegenüber äußerer Einflüssen ist. Das Gehäuse ist fest am Stromkasten bzw. an der Messbrücke und möglichst unauffällig zu montieren, um zusätzlichen Schutz vor Diebstahl zu gewährleisten.

5.2 Hardware

5.2.1 IoT Octopus

Ein zentrales Element von Fußgänger- und Fahrradzähler ist der IoT Octopus – eine Arduino-kompatible Entwicklungsplatine (s. Abbildung 34).

Der Hauptbestandteil der Platine ist der Mikrokontroller ESP8266 – eine kostengünstige und hochintegrierte Wi-Fi Multipoint Control Unit (MCU), die speziell für IoT-Anwendungen entwickelt wurde. Er zeichnet sich durch hohe Haltbarkeit, Kompaktheit und energiesparende Architektur aus. (*ESP8266 von espressif*)

Im IoT-Octopus ist ein Umweltsensor von Bosch (BME280 oder BME680) integriert, der Temperatur, Luftdruck, Luftfeuchte und Luftqualität misst.

An den Octopus können über zwei Pinouts (analog und I2C) Sensoren angeschlossen werden. Außerdem ist in der Platine einen Steckplatz für Adafruit Featherwings verbaut, was eine Anbindung unterschiedlicher Erweiterungen (z.B. LoRaWAN-Modul von tnn) ermöglicht (s. Abbildung 35).



Abbildung 34: IoT Octopus Development Board

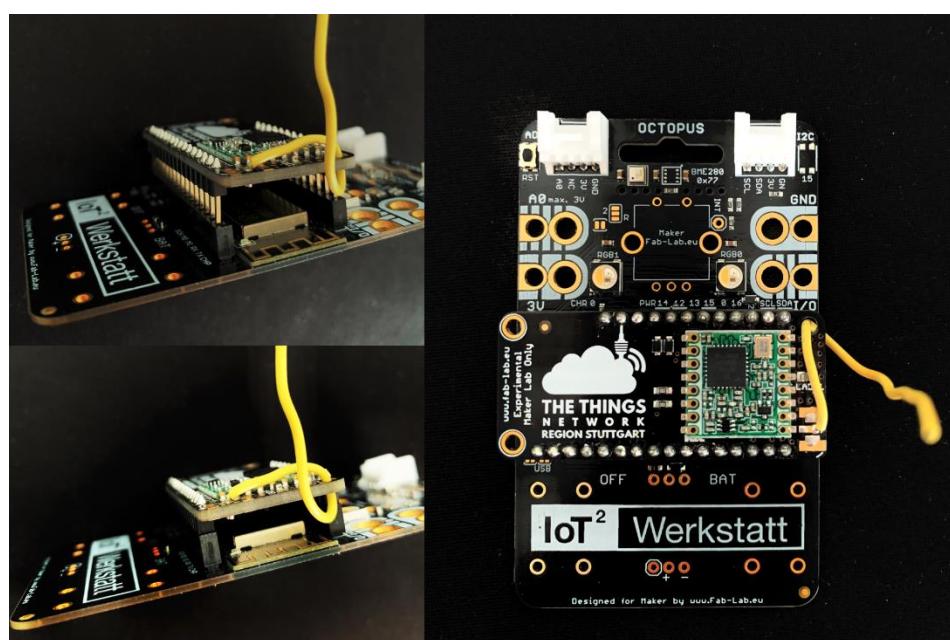


Abbildung 35: LoRaWAN-Modul von tnn am IoT Octopus

5.2.2 STM

Da der IoT-Octopus nicht über die UART-Schnittstelle verfügt, die für die Kommunikation mit dem Radarsensor erforderlich ist, wird für den MIV-Zähler das STM32L0 Discovery Kit benutzt.

Dieses Discovery Kit ist ein Entwicklungswerkzeug zum Erlernen und Entwickeln von Lösungen, die auf LoRa-, Sigfox u.a. Technologien basieren. Das Board enthält das All-in-One Modul CMWX1ZZABZ-091, welches mit einem STM32L072CZ-Mikrocontroller und einem SX1276-Transceiver betrieben wird. Der Transceiver verfügt über ein LoRa-Langstreckenmodem, welches eine hohe Störfestigkeit und geringen Stromverbrauch auszeichnet. Der Benutzer hat Zugriff auf alle Peripheriegeräte, u.a. auf UART.



Abbildung 36: STM32L0 Discovery Kit

Quelle: STM32L0 Overview

Der STM wird von Projektpartnern aus der Beuth Hochschule für unterschiedliche Anwendungen eingesetzt. Im Rahmen dieser Masterarbeit wird die Entwicklung von MIV-Zähler (sowohl von der Hardware als auch von der Software) von der Beuth Hochschule durchgeführt. Die Ergebnisse werden im Weiteren vorgestellt.

5.2.3 Sensoren

Für die Prototypenentwicklung wurden für alle drei Anwendungen die passenden Grove-Sensoren ausgewählt. Grove ist ein Open-Source, moduliertes und gebrauchsfertiges Toolset, was für Einfachheit und schnelles Prototyping optimiert ist. Der Hauptvorteil von Grove-Sensoren gegenüber traditionellen Elektronik-Bauteilen ist, dass sie über standardisierte Anschlüsse verfügen, von welchen viele direkt an den Octopus angeschlossen werden können. Da es von jedem Grove-Modul eine klare Dokumentation sowie einen Demo-Code gibt, sind die Detektoren nach dem

Anschluss in der Regel innerhalb von wenigen Minuten einsatzbereit. (*Grove Homepage*)

Ultraschallsensor

Der Ultrasonic Ranger von Grove misst die Impulsbreite eines Echosignals, die proportional zur gemessenen Entfernung ist. Er hat folgende Eigenschaften:

Tabelle 11: Eigenschaften von Grove Ultrasonic Ranger

Quelle: *Grove Ultrasonic Ranger*

Parameter	Wert / Wertebereich
Abmessungen	50mm x 25mm x 16mm
Gewicht	13 g
Messwinkel	15°
Messbereich	2-350 cm
Arbeitstemperatur	-10 bis ca. 60°
Betriebsspannung	3,2 - 5,2 V
Betriebsstrom	8 mA
Ultraschallfrequenz	40 kHz
Ausgang	PWM (I2C)



Abbildung 37: Ultraschallsensor

Quelle: *Grove Ultrasonic Ranger*

Der Ultraschallsensor wird über I2C an das Octopus-Board angeschlossen (s. Abbildung 38).

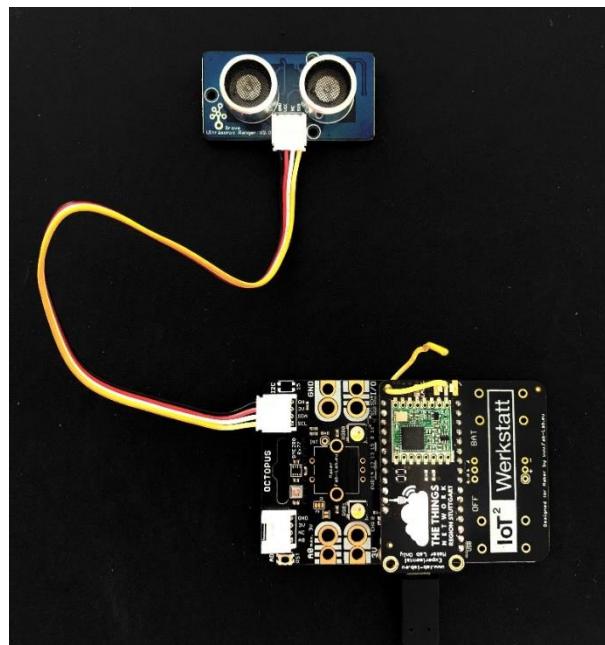


Abbildung 38: Elektronik Fußgängerzähler

Drucksensor

Der Silizium-Drucksensor MPX5700AP misst das Luftdruck im angeschlossenen Gummischlauch. Er hat folgende Eigenschaften:

Tabelle 12: Eigenschaften von Grove Pressure Sensor

Quelle: *Grove Integrated Pressure Sensor*

Parameter	Wert / Wertebereich
Abmessungen	49mm x 20mm x 5mm
Gewicht	10 g
Messbereich	15 kPa – 700 kPa
Arbeitstemperatur	-10 bis ca. 60°
Betriebsspannung	3,3 - 5 V
Ausgang	analog



Abbildung 39: Drucksensor

Quelle: *Grove Ultrasonic Ranger*

Der Sensor wird über den analogen Ausgang an Octopus angeschlossen (s. Abbildung 40).

Der Gummischlauch mit der Länge 5m, Innendurchmesser 3mm sowie Außendurchmesser 6mm wird luftdicht mit dem Sensor verbunden. Das andere Ende des Schlauchs wird luftdicht verschlossen.

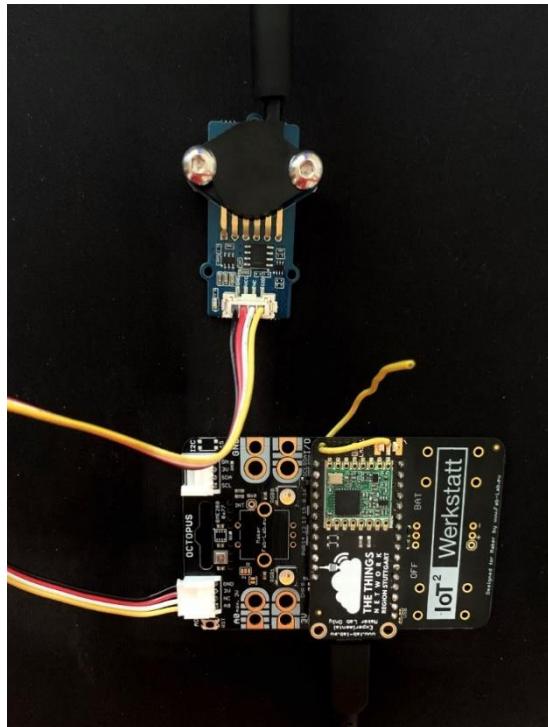


Abbildung 40: Elektronik Fahrradzähler

Radarsensor

Der Grove-Doppler-Radarsensor basiert auf einem 24-GHz-Radar-Transceiver BHT24LTR11 und wird von einem XMC1302-MCU angetrieben. Er hat folgende Eigenschaften:

Tabelle 13: Eigenschaften von Grove Doppler Radar

Quelle: *Grove Doppler Radar*

Parameter	Wert / Wertebereich
Abmessungen	40mm x 40mm
Gewicht	5 g
Erkennungsentfernung	10 m
Erkennungszone	65° horizontal 22° vertikal
Übertragungsfrequenz	24 GHz
Betriebsspannung	3,3 - 5 V
Kommunikationsschnittstelle	UART

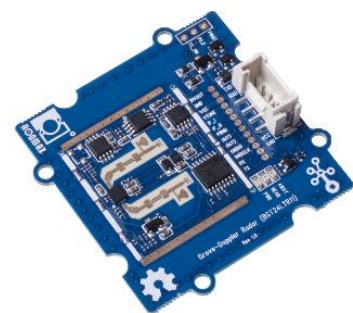


Abbildung 41: Radarsensor

Quelle: *Grove Doppler Radar*

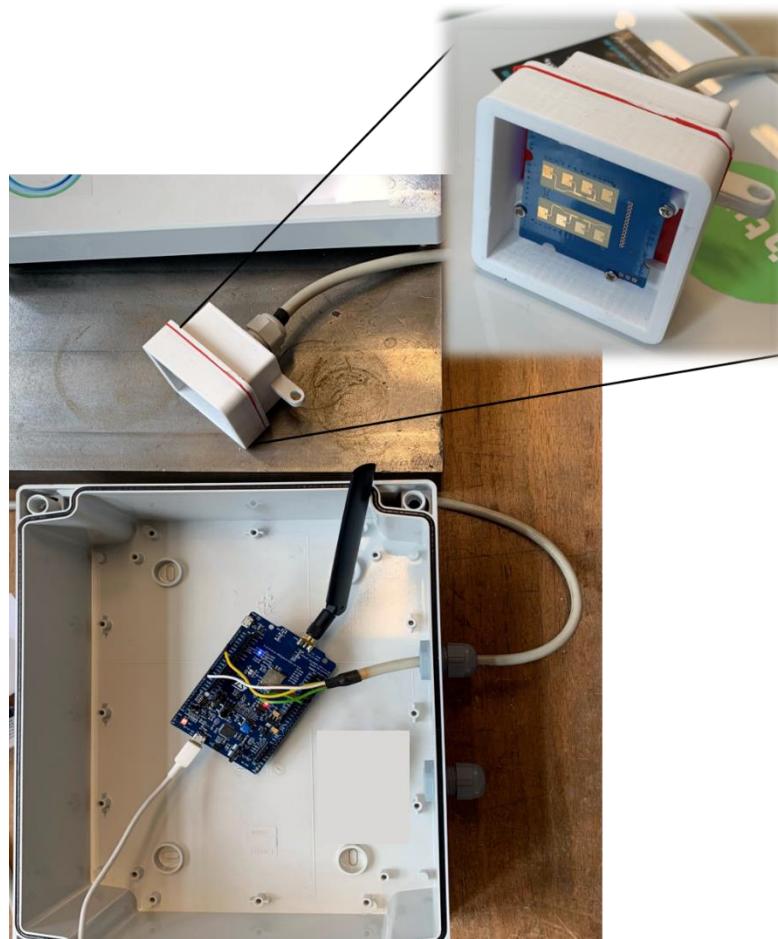


Abbildung 42: Elektronik MIV-Zähler

5.2.4 Gehäuse

Für Schutz vor äußeren Einflüssen wird für alle Detektoren ein Wandgehäuse Fibox TA292411 mit Abmessungen 289 x 239 x 107 mm verwendet. Das Gehäuse lässt sich schnell und ohne Verwendung spezieller Werkzeuge montieren.



Abbildung 43: Wandgehäuse Quelle: *Wandgehäuse Fibox*

Das Gehäuse wurde für jede Anwendung entsprechend angepasst. Für den Ultraschalldetektor wurde das Gehäuse mit Öffnungen und einer Überdachung versehen (s. Abbildung 44).

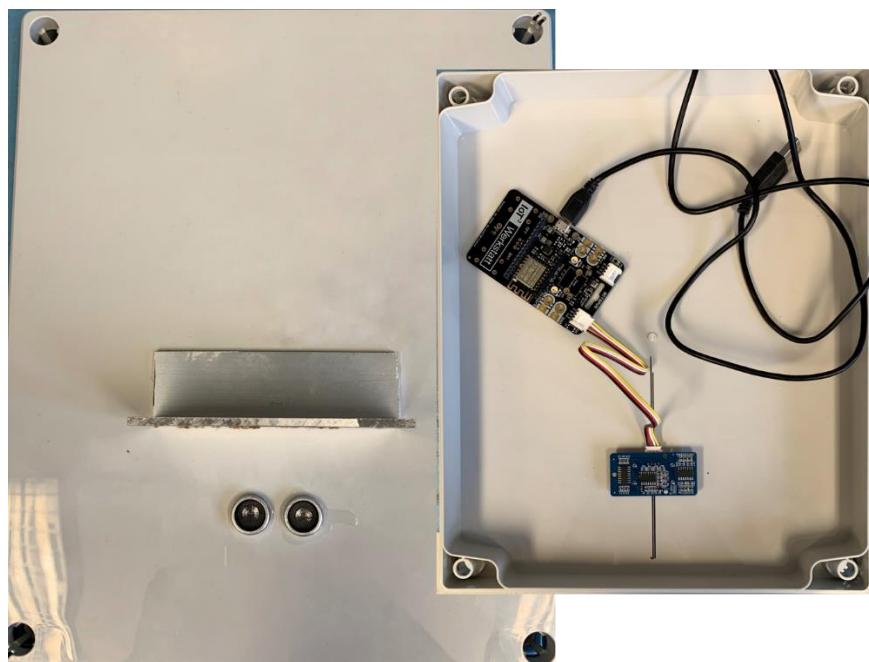


Abbildung 44: Gehäuse Ultraschalldetektor Vorderseite

Die Abbildung 45 stellt dar, wie das Stromkabel im Gehäuse verlegt wird. Es wird durch Kabelverschraubung eingebaut, was für eine Dichtung notwendig ist.



Abbildung 45: Gehäuse Ultraschalldetektor Stromkabelzugang

Das Gehäuse des Druckdetektors wurde mit zwei Öffnungen versehen: für das Stromkabel und für den Gummischlauch (s. Abbildung 46). Diese werden ebenso mit Kabelverschraubungen gesichert.

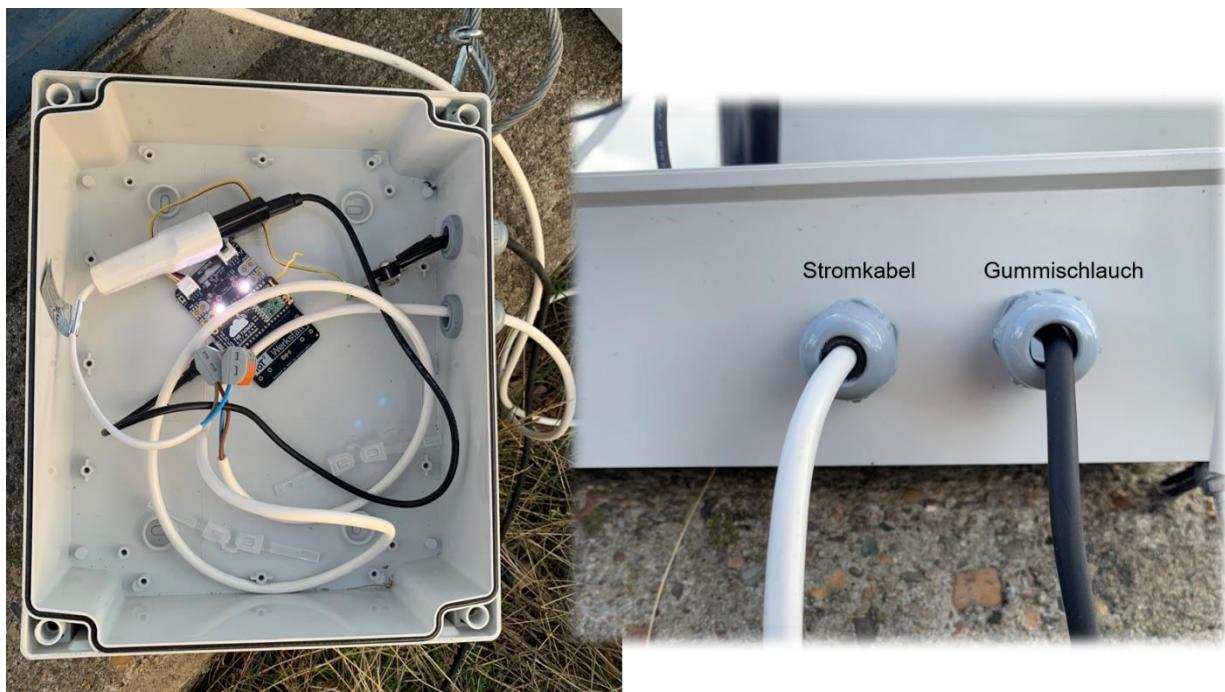


Abbildung 46: Gehäuse Druckdetektor

Für den Radardetektor wurden zwei Gehäuse vorbereitet. Das kleinere Gehäuse für den Grove-Radarsensor wurde mithilfe eines 3D-Druckers gefertigt und verfügt über eine Öffnung für das Verbindungskabel sowie einen Aufsatz zur Befestigung am Brückengelände. Der Sensor wird am Gehäuse fest verschraubt.



Abbildung 47: 3D-gedrucktes Gehäuse für den Grove-Radarsensor

Das Gehäuse für die restliche Elektronik hat zwei Öffnungen für ein Strom- sowie Verbindungskabel mit dem Radarsensor.



Abbildung 48: Gehäuse für Elektronik des Radardetektors

5.2.5 Montage & Inbetriebnahme

Fußgängerzähler

Die Montage des Ultraschalldetektors erfolgt an einem Stromkasten, welcher sich in der unmittelbaren Nähe des Fußgängerwegs befindet.



Abbildung 49: Montage Fußgängerzähler

Das Gehäuse wird mit zwei Metallstreifen an der inneren Türseite des Stromkastens befestigt. Dadurch wird verhindert, dass das Gerät abgenommen wird, ohne dass die Tür geöffnet wird.



Abbildung 50: Montage Fußgängerzähler 2

Fahrradzähler

Der Fahrradzähler wird mit einem Drahtseil mit der Brückenleiter verbunden, was Diebstahl verhindern sollte. Der Gummischlauch wird mit einem robusten wasserfesten Gewebe-Klebeband am Fahrradweg befestigt.



Abbildung 51: Montage Fahrradzähler



Abbildung 52: Montage Fahrradzäher 2

MIV-Zähler

Der Radardetektor wird für Testzwecken in zwei unterschiedlichen Weisen montiert. In beiden Fällen wird der Radarsensor so ausgerichtet, dass er in der Lage ist alle vier Fahrspuren zu überwachen.

Zuerst erfolgt die Installation überkopf am Brückengeländer. Dabei wird der Radarsensor in seinem Gehäuse an einem Aufsatz montiert, welcher mit Kabelbinder am Brückengeländer befestigt wird. Der Sensor ist dabei nach unten ausgerichtet und mit einem 2m UART-Kabel mit dem Board verbunden, welches sich im Gehäuse auf der Brücke befindet.

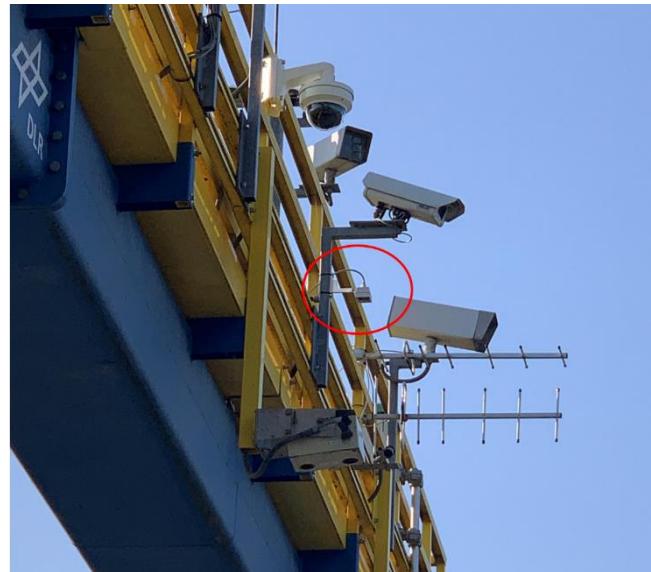


Abbildung 53: Montage MIV-Zähler überkopf



Abbildung 54: Montage MIV-Zähler überkopf 2

Für die seitliche Befestigung an Brückenleiter wurde ein längeres UART-Kabel (5m) benutzt. Die Befestigung des Sensors ist identisch zu der Überkopf-Montage. Der Sensor wurde so ausgerichtet, dass er Fahrzeuge auf allen vier Fahrspuren erkennen kann. Das Gehäuse mit der restlichen Elektronik befindet sich neben dem Druckdetektor auf dem Fundament und wird mit einem Drahtseil mit der Brückenleiter verbunden.



Abbildung 55: Montage MIV-Zähler seitlich

5.3 Software

5.3.1 Firmware IoT-Octopus

Die Programmierung des Octopus-Boards erfolgt in einer fertig konfigurierten Arduino-Umgebung, die speziell für Octopus entwickelt wurde – die IoT-Werkstatt (<https://www.umwelt-campus.de/forschung/projekte/iot-werkstatt>). Sie ermöglicht grafische Firmwarereprogrammierung mithilfe des ArduBlock - Werkzeugkastens mit Puzzleteilen (s. Abbildung 56). Die benötigten Blöcke, wie z.B. Sensoren, Kontrollstrukturen, Variablen usw. können ausgewählt und mit der Maus in die richtige Stelle gezogen werden. Es ist bereits vordefiniert, welche Blöcke miteinander zusammengefügt werden können. Das hilft dem Entwickler noch vor der Kompilierung Fehler zu vermeiden und sich an die richtige Programmietechnik zu halten.



Abbildung 56: ArduBlock-Umgebung

ArduBlock wird bei der Entwicklung der Firmware für Fußgänger- und Fahrradzähler verwendet. Mit seiner Hilfe werden die LoRaWAN-Protokolle definiert und die Sensortemperatur ausgelesen.

Für die Definition des LoRaWAN-Protokolls wurden im entsprechenden TTN-Block die *Device* und *App EUIs* sowie *App Key* eingetragen, die später einem angelegten Gerät in thingsHub zugewiesen werden. Außerdem wurde in diesem Block der *Port* definiert, über den die Daten übertragen und empfangen werden.

Um die Temperatur des Bosch-Umweltsensors auszulesen, wurde der Block Sensoren verwendet. Abhängig vom integrierten Sensor (BME280 oder BME680) wurde die entsprechende Variante ausgewählt und mit der Maus zum Feld 1 gezogen.

Die Abbildung 58 stellt das fertige ArduBlock-Programm für den Fußgängerzähler. Der Fahrradzähler besteht aus identischen Blocks und unterscheidet sich lediglich durch den Inhalt der jeweiligen Felder.



Abbildung 57: Sensoren-Block

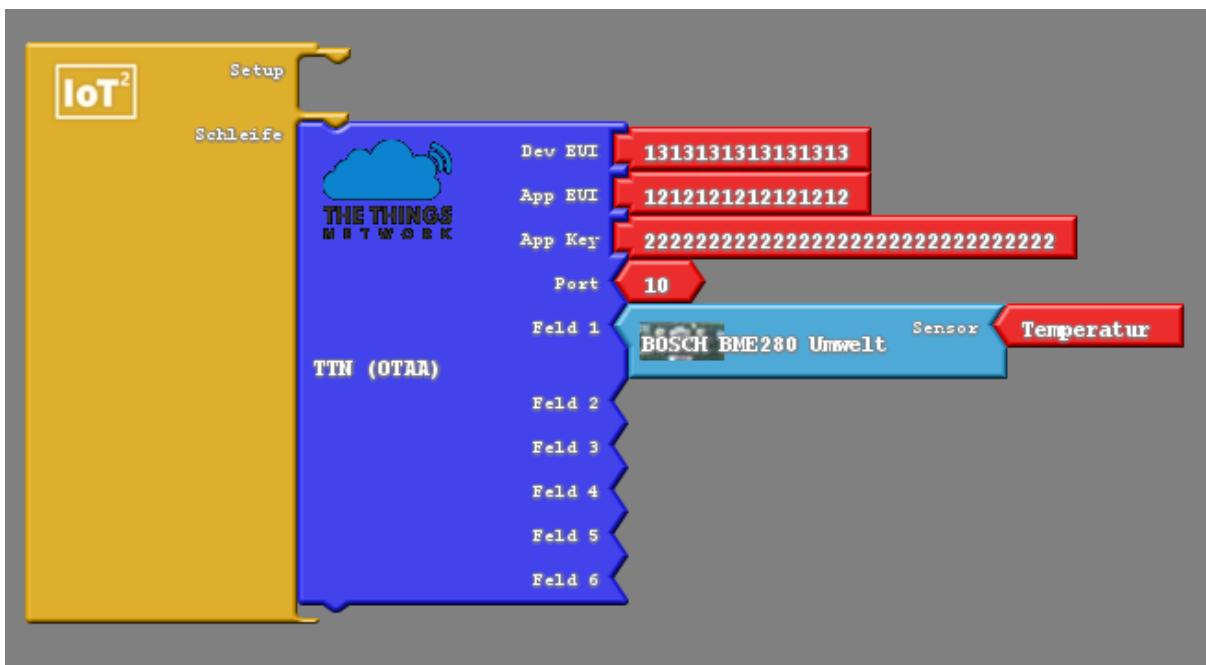


Abbildung 58: ArduBlock für Fußgängerzähler

Beim *Hochladen auf den Octopus* wird aus den ArduBlocks ein Quellcode generiert, welcher beliebig änderbar ist. Da nicht nur die Sensortemperatur, sondern auch Anzahl der Fußgänger bzw. Autofahrer übertragen werden soll, wurde die Firmware manuell erweitert.

Im Folgenden wird ein Codeausschnitt dargestellt, der die Datenaufnahme des Fußgängerzählers beinhaltet. Der automatisch generierte ArduBlock-Code wird in diesem Ausschnitt nicht dargestellt.

```

#include <lmic.h>
#include <hal/hal.h>
#include "Ultrasonic.h"
#include <Adafruit_BME680.h>
#include <Wire.h>

Ultrasonic ultrasonic(5);
const int distance = 300; // Abstand zur Grenze der Fußgängerzone
int pedestrianCounter = 0;

void setup(){ // Einmalige Initialisierung
    Serial.begin(115200);
    delay(5000);

    // Vorbereite BME680
    // Set up Bosch BME 680
    // Initialisiere LoRaWAN

    Wire.begin(); // Initialisiere den I2C-Bus
    if (Wire.status() != I2C_OK) Serial.println("Something wrong with I2C");

void loop() { // Kontinuierliche Wiederholung

    int RangeInCentimeters = 0;
    // es wird kontinuierlich gemessen
    RangeInCentimeters = ultrasonic.MeasureInCentimeters();

    /* wenn das erfasste Objekt sich innerhalb der Erkennungszone befindet
    und das Ergebnis der Messung ungleich Null bzw. 350 (maximaler Wert) ist,
    wird die Anzahl der Fußgänger um 1 erhöht */

    if(RangeInCentimeters < distance &&
        RangeInCentimeters != 0 && RangeInCentimeters != 350){
        pedestrianCounter++;
        Serial.println("FUßGÄNGER ERFASST! Anzahl der Fußgänger: ");
        Serial.println(pedestrianCounter);

        delay(1000); // Nachdem ein Fußgänger erfasst wird, 1 Sekunde nicht
                      // messen, um mehrfache Erfassung zu vermeiden
    }
    delay(500); // Messung alle 0,5 Sekunden
}
}

```

Um die erfasste Anzahl alle 15 Minuten senden zu können, wurde ein Timer implementiert. Nach Ablauf dieses Timers erfolgt die LoRaWAN-Datenübertragung. Da der Batteriezustand seltener, als die Anzahl der erfassten Objekte zu übertragen ist, wurde dafür ein separater Timer (24 Stunden) definiert.

Für die Implementierung des Timers wird die Arduino-Funktion *millis()* verwendet. Sie gibt die Anzahl der Millisekunden zurück, seitdem das aktuelle Programm gestartet wurde. Nach 50 Tagen geht diese Zahl wieder auf Null zurück. (*Arduino Funktionen 2020*)

Da es zwischendurch nicht möglich ist, *millis()* auf Null zu setzen, werden Hilfsvariablen definiert, die diese Funktion übernehmen. Jedes Mal nach Ablauf des 15-minütigen bzw. 24-stündigen Timers wird der aktuelle *millis*-Wert in der Hilfsvariable festgehalten.

Um beim nächsten Mal zu prüfen, ob der Timer abgelaufen ist, wird die Differenz zwischen aktuellem und dem in der Hilfsvariable gespeicherten *millis*-Wert mit dem Wunschintervall verglichen.

Solange die Differenz kleiner ist, soll weiter gemessen werden und sobald diese dem Wunschintervall erreicht hat, sollen die Daten an TTN übertragen werden.

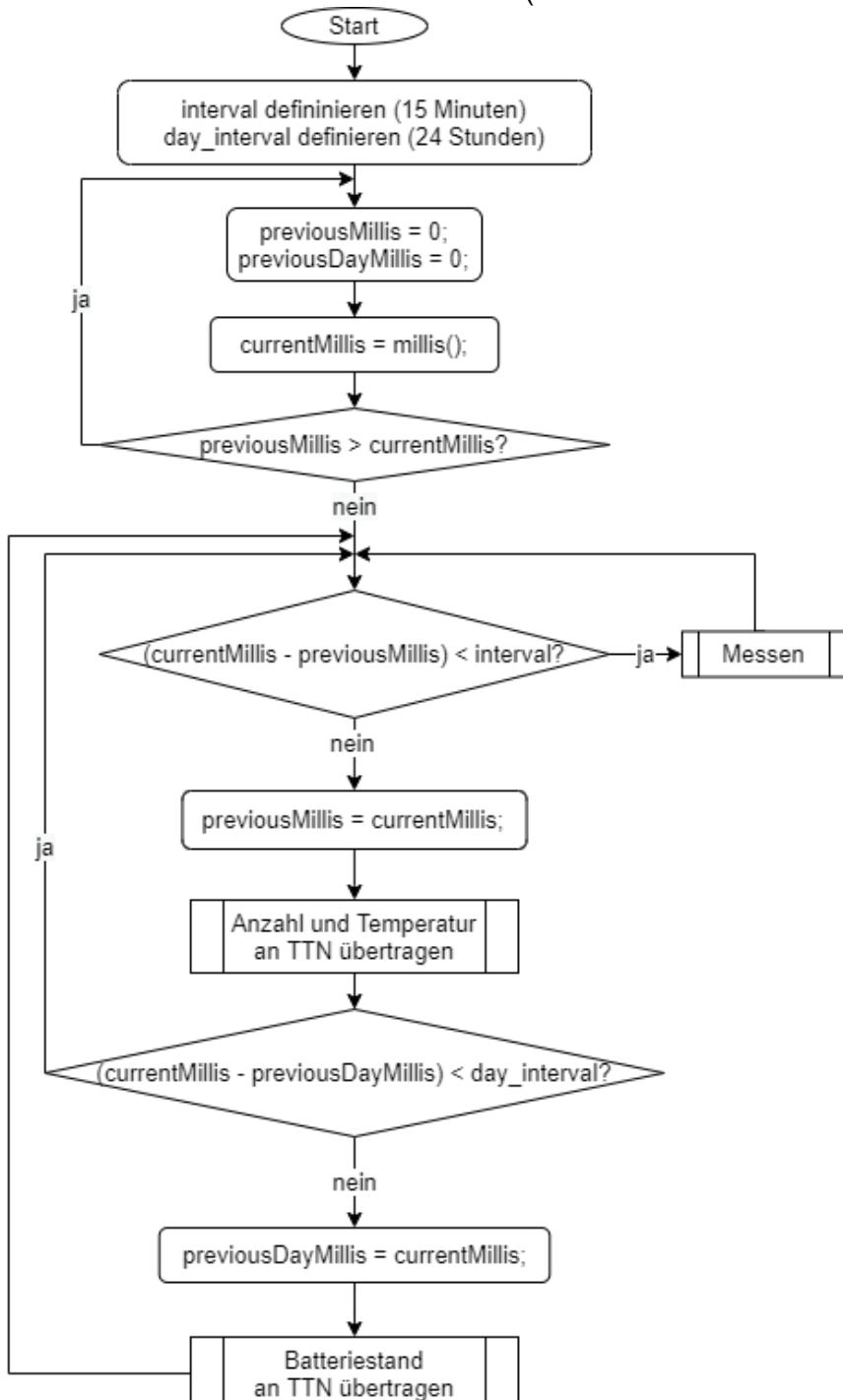


Abbildung 59: Timer-Flussdiagramm

```

const int interval = 900000; // 15 Minuten
const int day_interval = 86400000; // 24 Stunden
unsigned long previousMillis=0;
unsigned long previousDayMillis=0;
int temp = 0;
const int batterylevel = 100;
void setup(){ // Einmalige Initialisierung
    delay(5000);
    Serial.begin(115200);
    // Vorbereite BME680
    // Set up Bosch BME 680
    // Initialisiere LoRaWAN
    // Initialisiere den I2C-Bus

}
void loop() { // Kontinuierliche Wiederholung
    unsigned long currentMillis = millis();
    //Wenn 15 Minuten noch nicht abgelaufen sind
    if ((unsigned long)(currentMillis - previousMillis) < interval){
        //messen und prüfen, ob Fußgänger erkannt wird
    }
    //Wenn 15 Minuten bereits abgelaufen sind
    else {
        previousMillis = currentMillis; // Alle 15 Minuten die Zeit festhalten
        temp=round(boschBME680.readTemperature()); //Temperatur ermitteln
        {//Block sende Daten an TTN
        //Für die Übertragung der Anzahl sowie Temperatur wird Port 10 benutzt
        int port = 10;
        uint8_t measurement[2];
        measurement[0] = pedestrianCounter;
        measurement[1] = temp;
        // Daten senden
        }
        pedestrianCounter = 0;
        temp = 0;
        //Wenn 24 Stunden abgelaufen sind
        if ((unsigned long)(currentMillis-previousDayMillis) >= day_interval){
            previousDayMillis = currentMillis;
            //Block sende Daten an TTN
            {
            //Für die Übertragung des Batteriezustandes wird Port 11 benutzt
            int port = 11;
            uint8_t measurement[1];
            measurement[0] = batterylevel;
            // Daten senden
            }
        }
    }
}
}

```

5.3.2 Firmware STM

Dieser Abschnitt beinhaltet Codeausschnitte aus der Firmware für den Radardetektor, die vom Projektpartnern aus der Beuth Hochschule entwickelt wurde.

Die Entwicklung erfolgte in einer integrierten Entwicklungsumgebung STM32CubeIDE (<https://www.st.com/en/development-tools/stm32cubeide.html>).

Das ist eine C/C++ Plattform, welche Peripheriekonfiguration, Codegenerierung, Codekomplilierung und Debugging-Funktionen für STM32-Mikrocontroller und Mikroprozessoren ermöglicht.

Die Firmware besteht aus mehreren Komponenten. Abbildung 60 stellt eine Übersicht über ein Teil dieser Komponenten dar.

GBT24LTR11.c

Diese Komponente ist eine direkte Schnittstelle zwischen der Firmware und dem Sensor.

Die *receive()*-Funktion empfängt die Daten von der UART-Schnittstelle und erhöht den Counter für die Fahrzeuganzahl, falls die Mindestgeschwindigkeit überschritten wurde.

bsp.c

Hier erfolgt die Sensorinitialisierung

(Funktion *BSP_sensor_Init()*). Dabei wird der Bereich definiert, in dem die Geschwindigkeit gemessen wird (0-60km/h) sowie die gewünschte Empfindlichkeit des Sensors eingestellt.

Durch die Funktion *BSP_sensor_Read()* erfolgt die Kommunikation mit *GBT24LTR11.c*. Hier wird die *receive()*-Funktion abgerufen und die empfangenen Daten (Counter, Geschwindigkeit, Bewegungsrichtung) werden in einem externen Strukturkonstrukt *measure* gespeichert.



Abbildung 60: STM Codestruktur

main.c

Das ist die Hauptkomponente, welche für den Start des Programms zuständig ist. Sie definiert die Logik, nach der die Firmware funktionieren wird. Hier wird die LoRaWAN-Schnittstelle definiert und die Befehle von *bsp.c* aufgerufen.

Zuerst erfolgt die Sensorinitialisierung mittels *BSP_sensor_Init()*. Während der Timer läuft (15 Minuten), werden die Sensordaten mittels *BSP_sensor_Read()* ausgelesen. Wenn der Timer abgelaufen ist, wird die Funktion *Send()* aufgerufen, die in *main.c* deklariert wurde. Sie verschlüsselt die zu übertragende Anzahl durch Bitshifting und sendet diese an das LoRaWAN-Gateway. Nach dem Senden wird die Anzahl auf Null gesetzt und der Prozess beginnt von vorne an.

5.3.3 thingsHub

Die Integration der Applikationen in thingsHub-Plattform erfolgt nach folgendem Schema:



Abbildung 61: thingsHub-Integration

Device anlegen

Für jedes physische IoT-Gerät ist ein entsprechendes virtuelles Device bei thingsHub anzulegen. Dieser Prozess erfolgt in vier Schritten (s. Abbildung 62).

1.

Add new device

NAME / ID ACTIVATION KEYS CONNECT DRIVER

Name: PedestrianCounter
Device ID: pedestriancounter

1. Exit **Save & Next**

2.

Add new device

NAME / ID ACTIVATION KEYS CONNECT DRIVER

OTAA ABP

Device EUI (DevEUI): 13 13 13 13 13 13 13 13
Application EUI (AppEUI): 12 12 12 12 12 12 12 12
Application Key (AppKey): 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22

2. Exit **Save & Next**

3.

Add new device

NAME / ID ACTIVATION KEYS CONNECT DRIVER

Select the IoT Network to register and activate your device:
K-Tel LoRaWAN

3. Exit **Register Device & Next**

4.

Add new device

NAME / ID ACTIVATION KEYS CONNECT DRIVER

Assign a device driver
Select your Device Driver

4. Exit **Assign Driver & Finish**

Abbildung 62: Device bei thingsHub anlegen

Zunächst ist dem Gerät eine aussagekräftige Bezeichnung sowie eine eindeutige ID zuzuweisen. Als Nächstes sind die Schlüssel, die bereits in der Firmware definiert wurden, einzutragen. Danach ist das zugehörige LoRaWAN-Netzwerk zu wählen, damit das Gerät registriert und aktiviert werden kann. Der letzte Schritt ist das

Zuweisen des Gerätetreibers (auch Driver genannt). Da er implementiert werden soll, kann dieser Schritt zuerst übersprungen werden.

Driver implementieren & zuweisen

Gerätedriver dient als Übersetzer zwischen dem physischen Gerät und der IoT-Plattform. Er vermittelt dem thingsHub ein Verständnis dafür, wie die Daten, die ein Gerät sendet, interpretiert werden sollen.

Die Implementierung erfolgt mithilfe eines Driver Development Kit, welches die thingsHub Entwickler unter <https://gitlab.com/smartmakers/drivers/sdk> bereitstellen. Nach der Installation der Entwicklungsumgebung kann mit Implementierung begonnen werden. Die Dokumentation ist unter <https://gitlab.com/smartmakers/drivers/sdk/-/tree/master/docs> verfügbar.

Durch den Git-Bash-Befehl

\$drivers init golang wird im Ordner, der für den Driver erstellt wurde, ein golang-Projekt initialisiert.

Nach der erfolgreichen Initialisierung werden im Driver-Ordner fünf Dateien angelegt (s. Abbildung 63). Die Driver-implementierung erfolgt durch Konfiguration der .project-, main.go- sowie schema.yaml-Dateien.

Als Erstes sind die Treibermetadaten

einzupflegen. Dies erfolgt durch Erweiterung der .project-Datei. Die Metadaten sind allgemeine Informationen über den Driver, die die Kommunikation mit dem System (thingsHub) erleichtern. Außerdem helfen sie dem Benutzer schnell zu erkennen, welche Eigenschaften den Driver besitzt und für welche Anwendung er entwickelt wurde. Die Konfigurationsdaten für den Fußgängerzähler sehen wie folgt aus:

```
type: golang
metadata:
  name: driver-pedestrian-counter
  author: oksanatsurkan
  labels: []
  platform: "amd64"
  os: "linux"
```

```
MINGW64:/c/Users/oksan/Documents/driver
oksan@DESKTOP-2QHOUQG MINGW64 ~/Documents/driver
$ drivers init golang
go: creating new go.mod: module driver
go: gitlab.com/smartmakers/drivers/sdk upgrade => v1.2.12
Hey, golang driver engineer!
You are now ready to develop an awesome golang driver!
Make sure that you place all of your code in the file *main.go*.
You can build the driver with the command
  drivers build
Then you can use the command
  drivers decode --payload=<payload> --port=<port>
to check if driver correctly decodes your payload.
```

Abbildung 63: Projektinitialisierung

Als Nächstes ist ein Driver-Schema zu definieren. Das ist eine YAML oder JSON-Datei, die ausführliche Informationen über Daten beinhaltet, die vom Gerät übertragen werden. Ein Schema bildet ein Baum aus Eigenschaften (*properties*), die einen Typ haben und optional eine Beschreibung sowie eine Einheit haben können. Da thingsHub nur 64-bit-Variablen unterstützt, werden alle properties als int64-Variablen übertragen und gespeichert. Die Temperatur und Batteriestand werden zur Vereinfachung aufgerundet.

```
version: "3.0"
properties:
  pedestrians:
    type: int64
    description: "Counted pedestrians"
  temperature:
    type: int64
    unit: °C
    description: "Rounded sensor temperature"
  battery:
    type: int64
    unit: %
    description: "Current battery level (rounded)"
```

Der letzte Schritt ist die Implementierung der Uplink-Methode in *main.go*-Datei. Die nach der Projektinitialisierung automatisch erstellte Datei enthält bereits den Code, der erweitert werden soll. Die Methode *Uplink()* wird mit einer switch-case-Anweisung ergänzt, die Unterscheidung nach Ports beinhaltet. Sollte die Nachricht aus dem Port 10 kommen, ist die erste Variable des Payloads die Anzahl der Fußgänger und die zweite Variable ist die Sensortemperatur. Im Falle des Ports 11 ist lediglich der Batteriestand zu speichern.

```

package main

import (
    "os"
    "errors"
    "gitlab.com/smartmakers/drivers/sdk/go/driver/lvl3"
)

func main() {
    lvl3.Run(&driver{}, os.Args)
}

type driver struct {
    lvl3.DefaultDriver
}

func (drv *driver) Uplink(evt lvl3.UplinkEvent) (*lvl3.Response, error) {

    switch evt.Port{
    case 10:
        resp := lvl3.Response{
            Updates: []lvl3.Update{
                {
                    Time: evt.Time,
                    Properties: map[string]interface{}{
                        "pedestrians": evt.Payload[0],
                        "temperature": evt.Payload[1],
                    },
                },
            },
        }
        return &resp, nil
    case 11:
        resp := lvl3.Response{
            Updates: []lvl3.Update{
                {
                    Time: evt.Time,
                    Properties: map[string]interface{}{
                        "battery": evt.Payload[0],
                    },
                },
            },
        }
        return &resp, nil
    default:
        return nil, errors.New("Invalid port")
    }
}

```

Nach der Implementierung kann der Driver kompiliert und auf thingsHub hochgeladen werden. Dazu ist im Driver-Ordner der build-Befehl `$ drivers build` auszuführen. Nach dem erfolgreichen Kompilieren ist die Anmeldung im Driver-Registry erforderlich.

```
$ drivers login <server> <username> -p <password>
```

Der letzte Schritt ist das Hochladen des Drivers.

```
$ drivers push -t version
```

Nach dem Hochladen kann der Driver bei thingsHub abgerufen und dem Gerät zugewiesen werden.

Im Drivers-Menü soll zunächst die Driverliste aktualisiert werden.

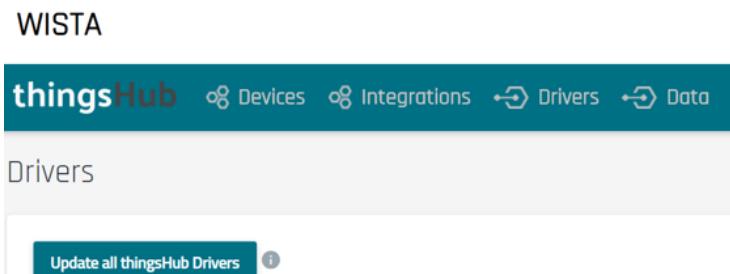


Abbildung 64: Driverliste aktualisieren

Wenn die Aktualisierung abgeschlossen ist, kann der Driver einem Gerät zugewiesen werden. Im Devices-Menü kann der gewünschte Driver aus der Liste ausgewählt werden. Anschließend ist die Auswahl mit *Assign* zu bestätigen und der Driver wird dem Gerät zugewiesen (s. Abbildung 65).

1.

DEVICE DRIVER

not set

oksanatsurkan/driver-pedestrian-counter:1.0.3

smartmakers/zenner-edc-lora:0.2.1

florianmerz/driver-demo:0.1.0

oksanatsurkan/driver-pedestrian-counter:1.0.3

smartmakers/generic-hex:1.0.1

smartmakers/elsys-ers-co2:3.1.2

smartmakers/elsys-ers:1.1.0

2.

DEVICE DRIVER

oksanatsurkan/driver-pedestrian-counter:1.0.3

Select your Device Driver

Assign

Abbildung 65: Driver zuweisen

Datentabelle erstellen

Standardmäßig werden die empfangenen Daten in temporären Tabellen, die automatisch angelegt werden, für sieben Tage gespeichert.

Damit die Daten bei thingsHub dauerhaft gespeichert werden können, soll für jede Anwendung eine Datentabelle erstellt werden.

Dabei ist die Tabellenname, Beschreibung und Device Label einzupflegen. Das erstellte Device Label soll anschließend dem Device zugewiesen werden.

Datenimport & -export

Sollten externe Daten in thingsHub importiert werden, gibt es dazu zwei Methoden: manuell und über REST API. Im Menüpunkt Data können CSV-Dateien importiert werden. Nach dem Import erscheinen sie als neue Datentabellen in der Liste. Auf die Dokumentation der REST-Schnittstelle kann von Homepage der thingsHub zugegriffen werden. Im Bereich *Data Tables* befinden sich die dazu benötigten Abfragen.

Der Datenexport vom thingsHub ist durch REST API mit der Abfrage [GET /api/v3/data/query](#) möglich. Dabei ist die gewünschte Abfrage als *query* zu übergeben. Um den Inhalt der Fußgängerzähler-Tabelle zu bekommen ist folgende Abfrage zu senden: `SELECT * FROM "autogen"."pedestrian-counters"`.

Server response	
Code	Details
200	<p>Response body</p> <pre>"2020-09-10T15:54:46.701Z", null, "pedestriancounter", null, "PedestrianCounter", 47, 29], ["2020-09-10T15:57:33.613Z", null, "pedestriancounter", null, "PedestrianCounter", 71, 29],</pre>

Abbildung 68: REST-API Response

Create Data table

Name
PedestrianCounter

Automatically generated name for time series measurement
pedestriancounter

Description
Table for storing pedestrian counter's data:
-pedestrians
-temperature
-battery

Device data label selector
Help: Select the label of the devices from which you want to store data
xPedestrian Counter

Abbildung 66: Datentabelle erstellen

GET /api/v3/data/query Query your IoT devices data.

Parameters

Name	Description
q * required string (query)	time series IoT device data query

```
SELECT * AS "pedestrians" FROM "autogen"
```

Execute Clear

Abbildung 67: REST-API Datenexport

Eine andere Methode die Daten zu exportieren, ist durch eine Export CSV-Funktion in Visualizer. Der Inhalt eines beliebigen Panels kann als CSV-Datei exportiert werden (s. Abbildung 69).

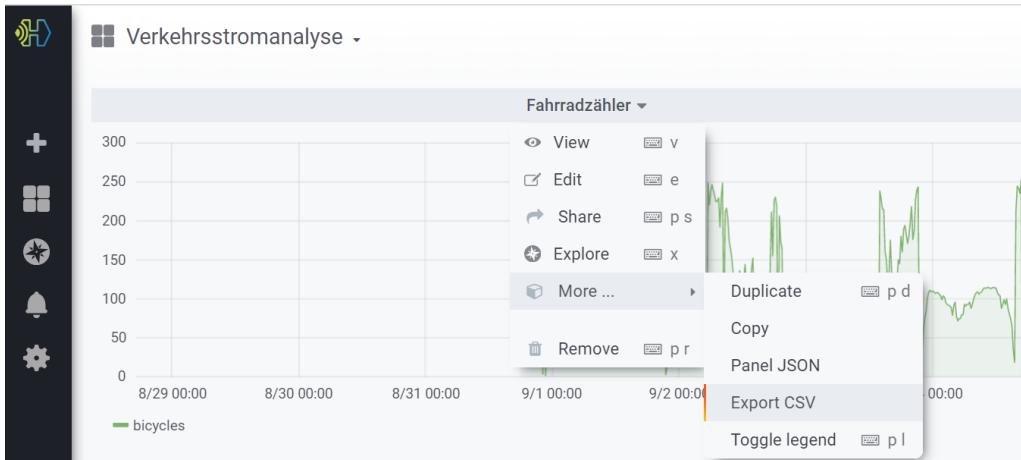


Abbildung 69: Export CSV

5.3.4 Datenvisualisierung

In thingsHub werden die Daten mithilfe von einem integrierten Tool, welches auf Grafana basiert, visualisiert. Als erstes wird ein Dashboard *Verkehrsstromanalyse* erstellt, welches alle Visualisierungen enthalten wird.

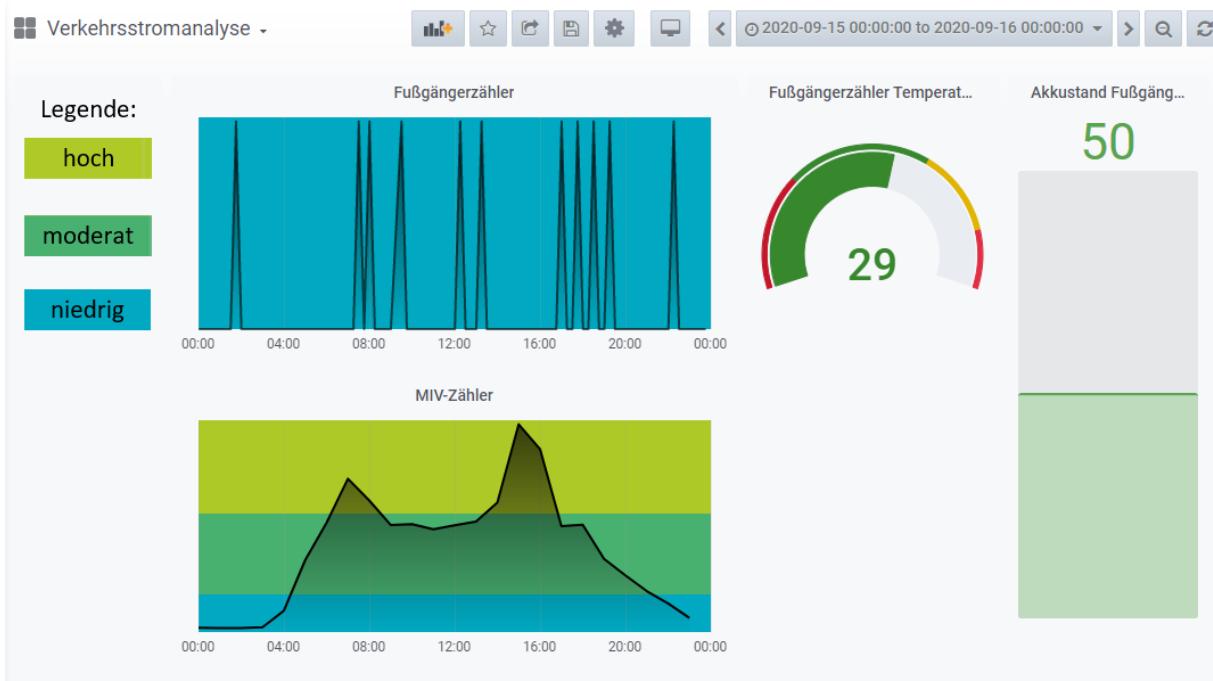


Abbildung 70: Dashboard Verkehrsstromanalyse

Drei unterschiedliche Visualisierungsarten (Panels) befinden sich auf diesem Dashboard: ein *Graph-Panel* zur Anzeige der erfassten Personenzahlen, ein *Gauge-Panel* zur Anzeige der Temperatur und ein *Bar-Panel* zur Anzeige des Akkustands.

Panel zur Anzeige der Detektortemperatur sowie ein Bar-Gauge-Panel, um den aktuellen Akkustand darzustellen. Im Folgenden werden die einzelnen Visualisierungen am Beispiel des Fußgängerzählers detailliert beschrieben.

Die Graph-Visualisierung stellt die kumulierte Anzahl der erfassten Personen (Y-Achse) über ausgewählte Zeit (X-Achse) dar.

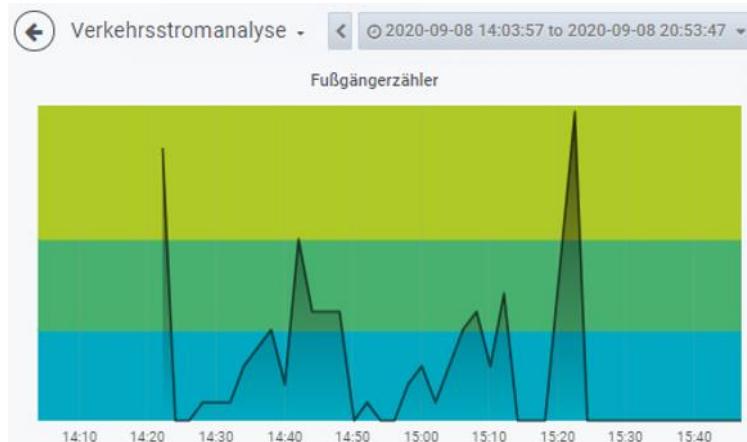


Abbildung 71: Graph-Panel

Die Daten werden aus der entsprechenden Tabelle ausgelesen. Die Datenabfrage ist in Abbildung 72 dargestellt.

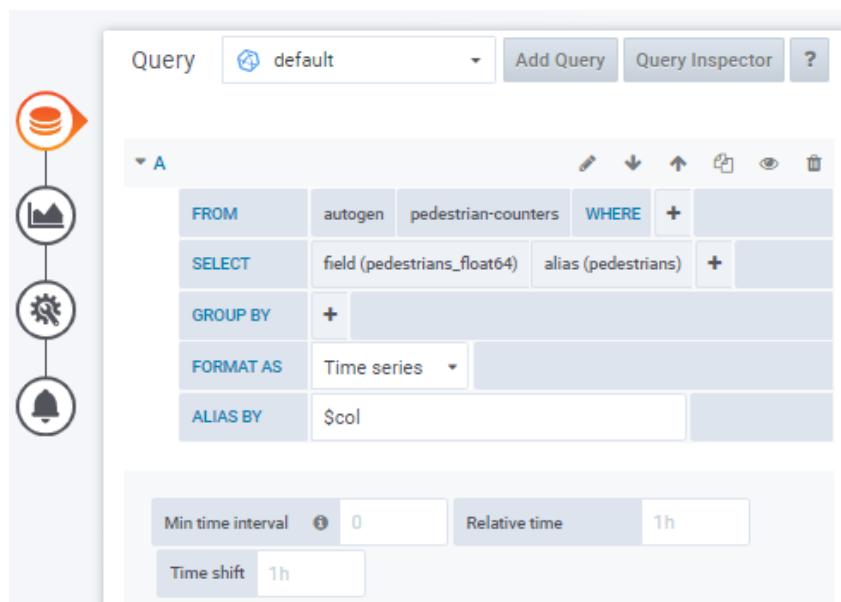


Abbildung 72: Graph-Panel Datenabfrage

- *autogen* weist bei thingsHub darauf hin, dass die Detektordaten aus einer Tabelle ausgelesen werden, wo sie langfristig gespeichert werden.
- *pedestrian-counters* ist die eindeutige Tabellen-ID.
- *field(pedestrian_float64)* – dieser Wert wird auf dem Grafen abgebildet.
- *alias(pedestrians)* – Bezeichnung für die selektierten Daten

Nachdem die richtigen Daten ausgelesen werden, kann die Ausgabe angepasst werden. Da die Visualisierung keine absoluten Werte beinhalten soll, werden für den Fußgänger folgende Schwellenwerte definiert:

Threshold	Comparison	Value	Color	Fill	Line	Y-Axis
T1	lt	5	custom	on (orange)	on (orange)	left
T2	gt	5	custom	on (orange)	on (green)	left
T3	gt	10	custom	on (orange)	on (green)	left

Abbildung 73. Graph-Panel Schwellenwerte

Dabei steht *lt* für *less than* (kleiner als...) und *gt* für *greater than* (größer als...). Die Bereiche werden farblich voneinander unterschieden. Wenn die Anzahl der erfassten Personen in 15 Minuten kleiner als 5 ist, wird die Auslastung in diesem Zeitraum als niedrig dargestellt. Zwischen 5 und 10 ist die Auslastung moderat und ab 10 ist diese hoch. Zur Temperaturanzeige wird die Gauge-Visualisierung verwendet.

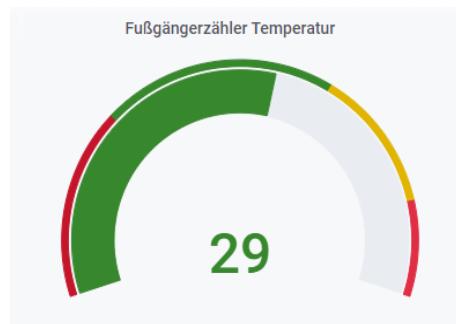


Abbildung 74: Gauge-Panel

Der letzte empfangene Wert wird als Zahl und zusätzlich die Zuordnung dieses Wertes auf einer farblichen Skala angezeigt. Das dient der schnellen Übersicht und der sofortigen Problemerkennung. Die Datenabfrage ist in Abbildung 75 dargestellt.

```
Query: default
▼ A
SELECT last("temperature_float64") FROM "autogen"."pedestrian-counters"
FORMAT AS Time series ALIAS BY Naming pattern
Min time interval 0 Relative time 1h Time shift 1h
```

Abbildung 75: Gauge-Panel Datenabfrage

- `last(„temperature_float64“)` – die letzte empfangene Temperatur
- `“autogen”.“pedestrian-counters”` – Tabelle, aus der die Daten ausgelesen werden.

Als Nächstes werden die Schwellenwerte definiert, um die Anzeige farblich zu gestalten.

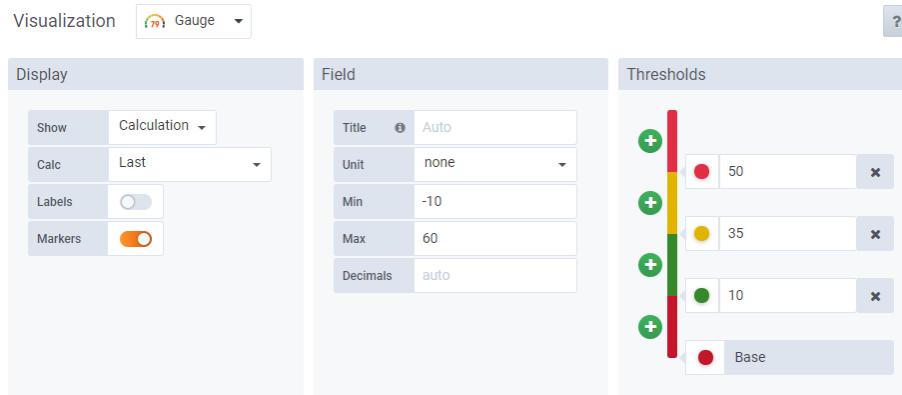


Abbildung 76: Gauge-Panel Schwellenwerte

Die mögliche Temperatur liegt zwischen -10 und 60°C. Dabei ist der Bereich zwischen 10 und 35°C als normal (grün) zu betrachten, während der zwischen 35 und 50°C als hoch (gelb) und über 50 bzw. unter 10°C als kritisch (rot) bezeichnet werden.

Künftig sollen die Detektoren mit Akkus/Batterien getrieben werden. Obwohl es zum aktuellen Zeitpunkt noch nicht geschieht, wird bereits eine Visualisierung konfiguriert, welche den Akkustand anzeigt. Aktuell sendet der Detektor immer der gleiche vorprogrammierte Wert. Um ihn auszulesen, wird folgende Datenabfrage ausgeführt:

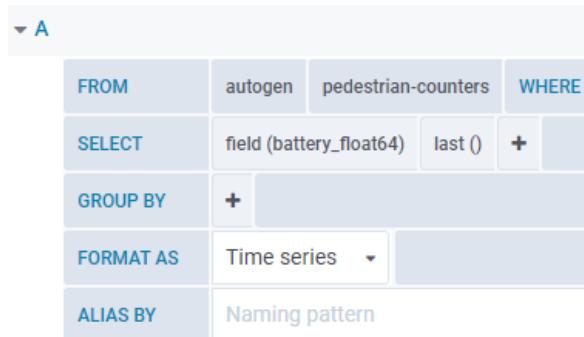


Abbildung 77: Bar Gauge-Panel Datenabfrage

Um die Anzeige farblich zu gestalten, ist der minimale (0%) sowie der maximale Wert (100%) zu definieren. Der kritische Akkustand (1-10%) wird mit der roten Farbe signalisiert, der niedrige (10-20%) mit der gelben und der Akkustand ab 21% wird grün markiert.

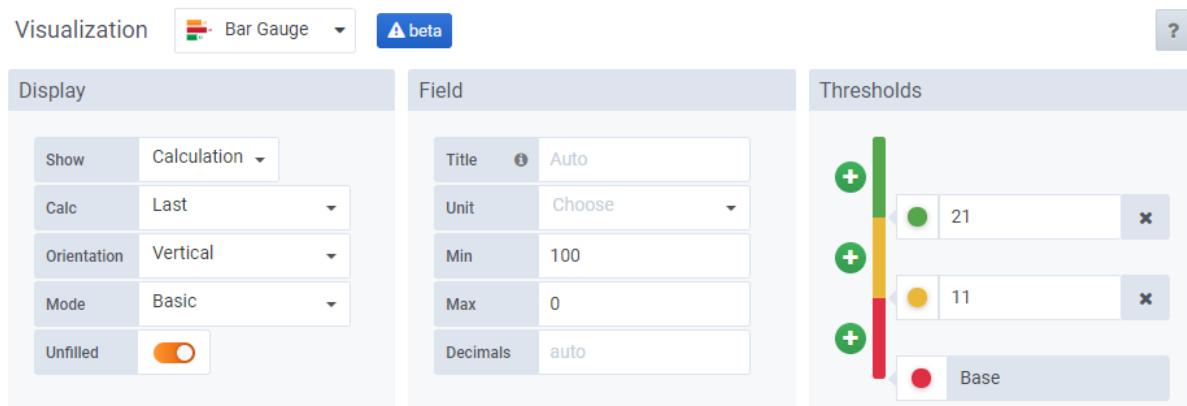


Abbildung 78: Bar Gauge-Panel Schwellenwerte

6 Ergebnisse

6.1 Durchgeführte Tests

Um die Funktionsweise der Detektoren zu überprüfen werden verschiedene Testszenarien durchgeführt. Im Folgenden werden Ziele, Durchführung sowie Ergebnisse dieser Tests erläutert.

Fußgängerzähler

Da die Fußgängerzahlen am Messstandort unterrepräsentiert sind, wurde ein fiktives Szenario durchgespielt. Das Ziel ist die Reichweite und Präzision des Sensors unter realen Bedingungen zu messen. Dabei werden Situationen modelliert, bei denen eine Person aus unterschiedlicher Entfernung (<0,5m, 0,5-1m, 1-1,5m und >1,5m) ein oder mehrere (5, 10 und 20) Male am Sensor vorbeiläuft. Jedes Szenario wird vier Mal durchgespielt und es wird ermittelt, wie oft der Sensor die Person erkannt hat.

Die Testergebnisse sind in Tabelle 14 dargestellt.

Tabelle 14: Testergebnisse Fußgängerzähler

Entfernung	<0,5m	0,5-1m	1-1,5m	>1,5m
Ziel (Anzahl der Erkennungen)	1	1	1	1
Versuch 1	0	0	0	0
Versuch 2	1	1	0	0
Versuch 3	1	0	0	0
Versuch 4	1	0	1	0
Ziel (Anzahl der Erkennungen)	5	5	5	5
Versuch 1	3	2	0	0
Versuch 2	4	3	1	0
Versuch 3	5	1	1	0
Versuch 4	2	3	1	0
Ziel (Anzahl der Erkennungen)	10	10	10	10
Versuch 1	10	5	7	0
Versuch 2	6	6	2	0
Versuch 3	6	3	1	0
Versuch 4	6	7	0	0
Ziel (Anzahl der Erkennungen)	20	20	20	20
Versuch 1	20	17	2	0
Versuch 2	18	13	5	0
Versuch 3	20	15	3	0
Versuch 4	17	14	1	0

Dabei werden sie in drei Gruppen geteilt, die sich farblich voneinander unterscheiden. Grün werden Ergebnisse markiert, die zu 100% mit dem Ziel übereinstimmen. Sollte

das Ergebnis vom Ziel abweichen, aber trotzdem einen richtigen Trend aufweisen (z.B. bei Zielanzahl 5 sind das 2 und 3, aber nicht 1 und 4, bei 10 – 6, 7, 8, 9, aber nicht 5 und kleiner), werden sie gelb markiert. Zum Schluss werden Null-Ergebnisse oder die, die keinen Trend aufweisen, rot markiert.

Fahrradzähler

Nach der Montage sollte zuerst der vorhandene Druck im Schlauch im Normalzustand gemessen werden. Da die eigentliche Umrechnung der Rohdaten in Druck keine Relevanz hat, wurden dabei ausschließlich Rohwerte analysiert. Der Rohwert im Normalzustand betrug 198. Der Druckdetektor wurde anschließend so programmiert, dass sobald dieser Wert überschritten wird, wird die Anzahl der Fahrradfahrer um eins erhöht.

Das Auftreten auf den Gummischlauch hat den Anstieg des Rohwertes bis auf 200 verursacht. Dabei hat sich der Rohwert beim Überfahren eines Fahrrads nicht geändert. Der Druckunterschied im Gummischlauch war nicht zu erkennen unabhängig davon, wie weit vom Sensor das Fahrrad gefahren ist. Die Abbildung 79 zeigt die Rohwerte des Fahrradzählers, die am 9.9.2020 gemessen wurden. Daraus ist erkennbar, dass die Rohwerte lediglich zwischen 0 und 9 Uhr einzeln auf 199 steigen, was durch äußere Faktoren wie Temperaturschwankungen zu erklären sind. Während des Tages wurde kein einziger Fahrradfahrer erkannt.

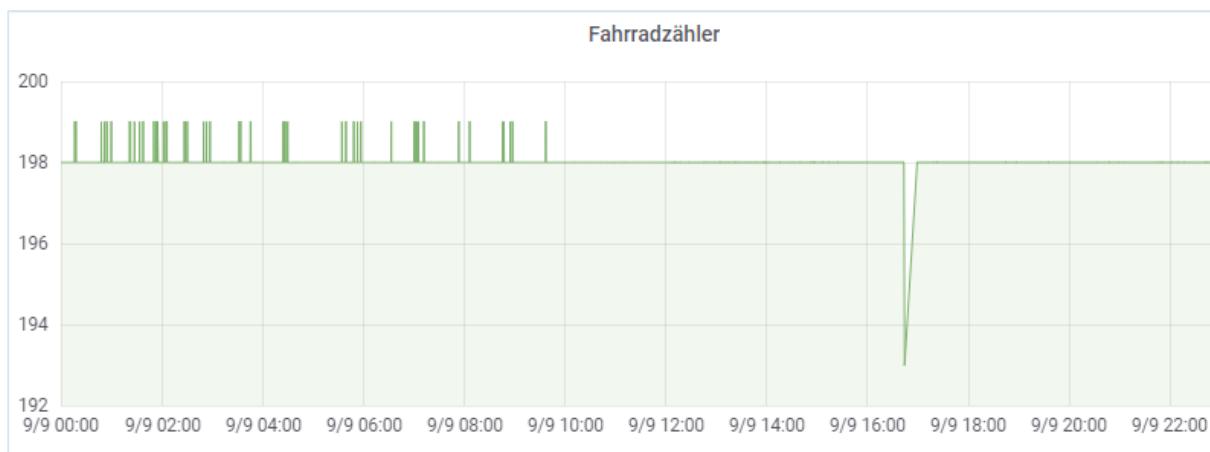


Abbildung 79: Rohwerte des Fahrradzählers

Aus diesem Grund wurde davon ausgegangen, dass der Impuls, der durch das Überfahren übertragen wird, für den Drucksensor zu klein ist. In Zusammenarbeit mit der Beuth Hochschule wurde der Spannungsoutput des Drucksensors so modifiziert, dass die kleinsten Druckänderungen erkennbar sein sollten. Der Schwellenwert wurde neu gesetzt und Testmessungen wurden durchgeführt. Die Fahrradfahrer wurden trotz Elektronikanpassungen nicht erkannt. Auch der Schlauchtausch hatte keinen Einfluss auf die Messergebnisse.

Aus den durchgeführten Tests folgt, dass der Grove-Drucksensor in Kombination mit einem Gummi- bzw. Silikonschlauch nicht für die Erfassung des Fahrradaufkommens geeignet ist.

MIV-Zähler

Da der Radardetektor direkt an der Messbrücke über den Induktionsschleifen montiert wurde, können dessen Ergebnisse durch die Schleifenergebnisse validiert werden. Die dazu benötigten Daten werden von DLR zur Verfügung gestellt. Das Ziel dieses Tests ist die Ermittlung der Präzision des Grove-Radarsensors und ggf. Auswahl der geeigneten Position für die Montage (überkopf oder seitlich).

Die erfassten Daten der beiden Technologien werden stündlich aggregiert und in einem Liniendiagramm gegenübergestellt.

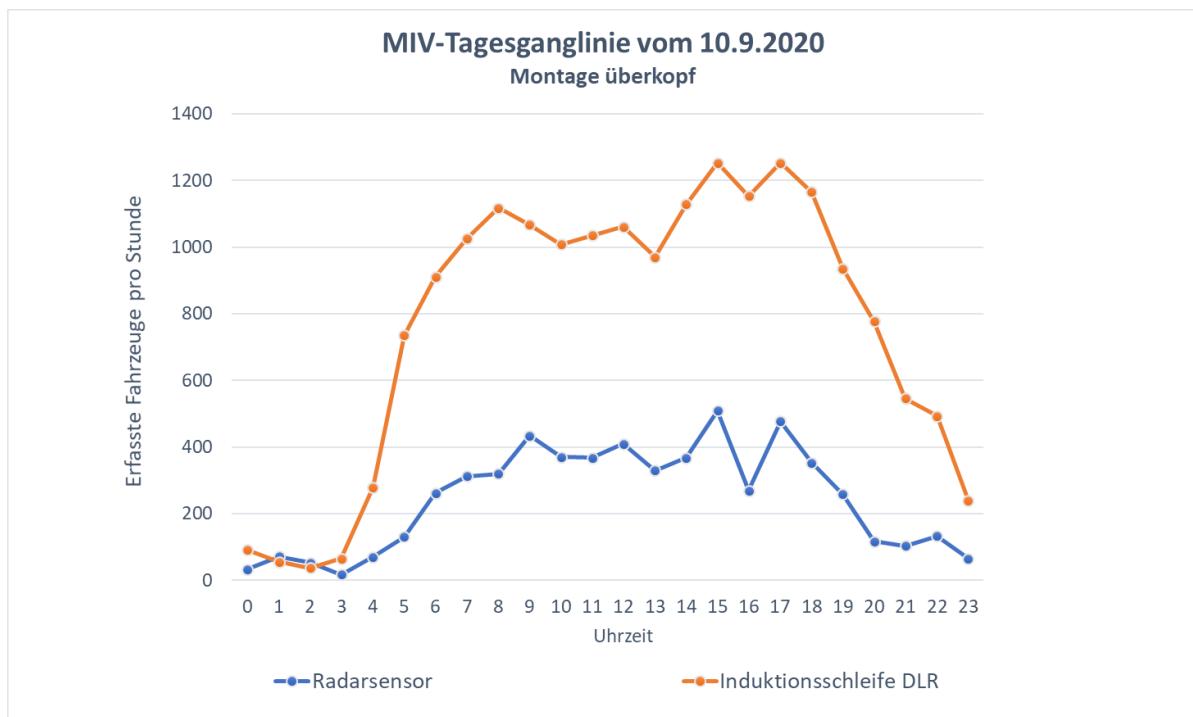


Abbildung 80: MIV-Tagesganglinie Radardetektor vs. Induktionsschleife, Montage überkopf

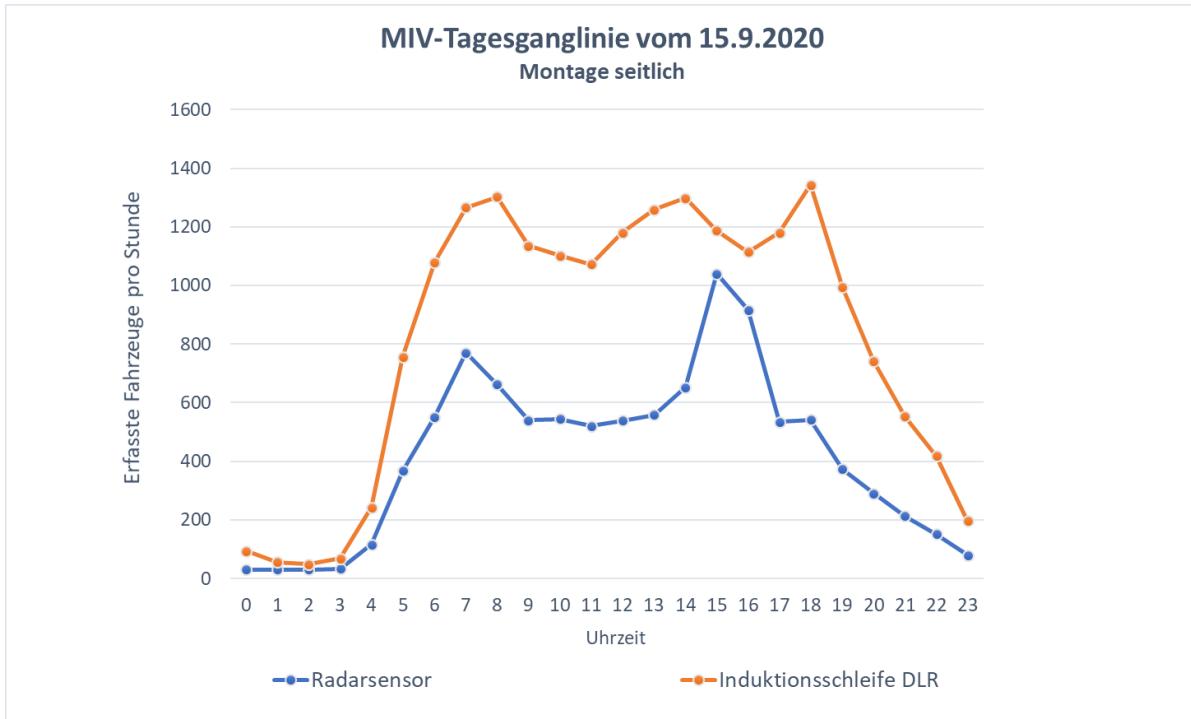


Abbildung 81: MIV-Tagesganglinie Radardetektor vs. Induktionsschleife, Montage seitlich

6.2 Bewertung der Präzision

Fußgängerzähler

Aus den Testergebnissen lassen sich folgende Schlussfolgerungen ziehen. Zum einen beträgt die Reichweite des Grove-Ultraschallsensors in realen Bedingungen *maximal 1 Meter*. Auch innerhalb dieser Reichweite sind die absoluten Messergebnisse nicht zu 100% zuverlässig. Nur 25% dieser Ergebnisse stimmen mit den Zielwerten überein. 18,75% der Ergebnisse sind nicht aussagekräftig. Die restlichen 56,25% weisen einen Trend auf und sind für die Auswertungen relevant.

Der Fußgängerzähler kann in Standorten angebracht werden, wo die Breite des Fußgängerweges bei maximal einem Meter liegt. Andernfalls können zwei Detektoren auf gegenüberliegenden Seiten des Gehweges, dessen Breite zwei Meter beträgt, montiert werden. Im Übrigen kann die Abdeckung der gewünschten Erkennungszone nicht gewährleistet werden.

MIV-Zähler

Die Testergebnisse weisen darauf hin, dass die Präzision des Grove-Radarsensors im Vergleich zu Induktionsschleifen sehr gering ist. Die durchschnittliche prozentuale Abweichung bei einer Montage überkopf liegt bei ca. 67%.

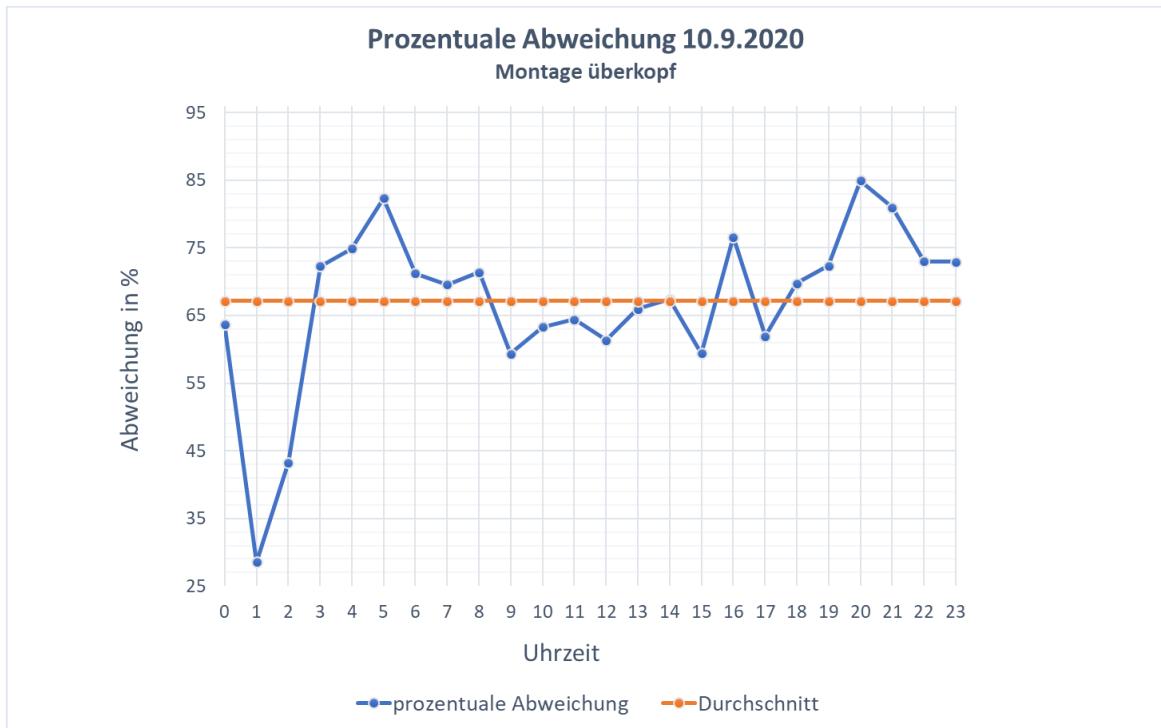


Abbildung 82: MIV-Zähler prozentuale Abweichung (Montage überkopf)

Die seitliche Montage verbessert die durchschnittliche prozentuale Abweichung der Ergebnisse um 17%.

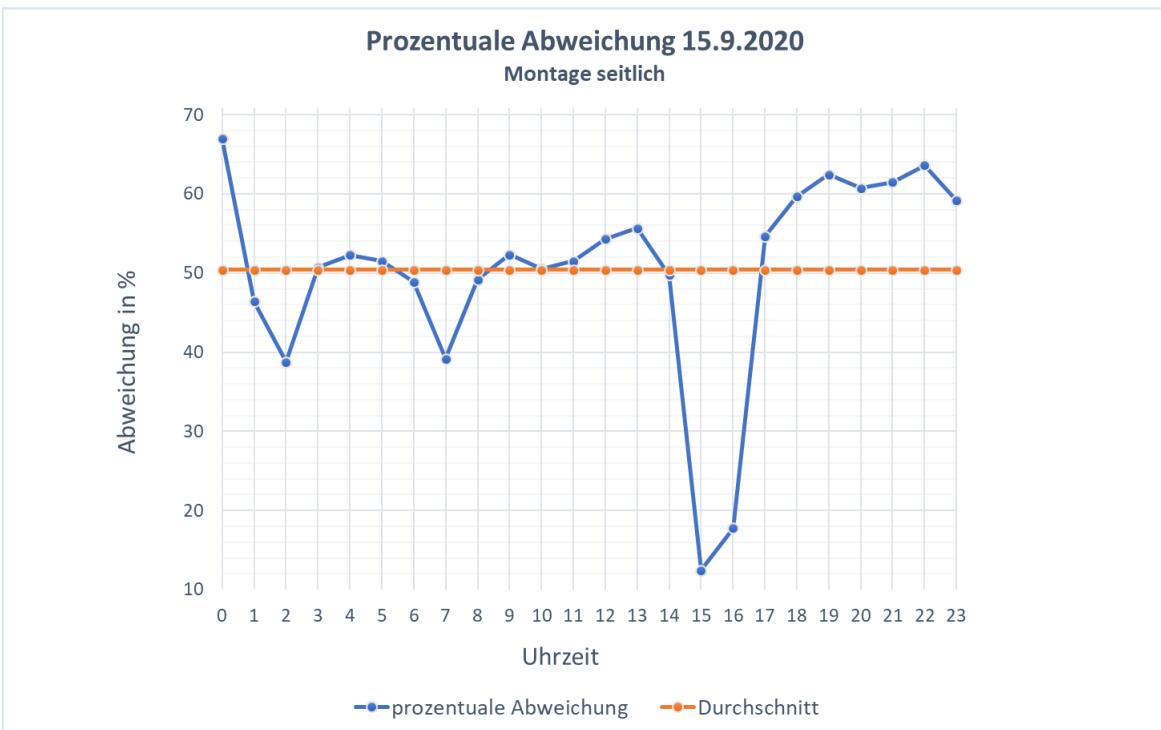


Abbildung 83: MIV-Zähler prozentuale Abweichung (Montage seitlich)

Da die Datenauswertung anhand relativen Werten durch Trendanalyse erfolgt, wird als Nächstes der Anstieg der beiden Kurven analysiert. Dabei wird zuerst die Differenz zwischen dem nächsten Datenpunkt und dem aktuellen gebildet. Im nächsten Schritt wird die Differenz auf dem aktuellen Wert normiert. Diese Berechnung wird für jeden Datenpunkt durchgeführt mit Ausnahme vom letzten, da er keinen Nachfolger hat.

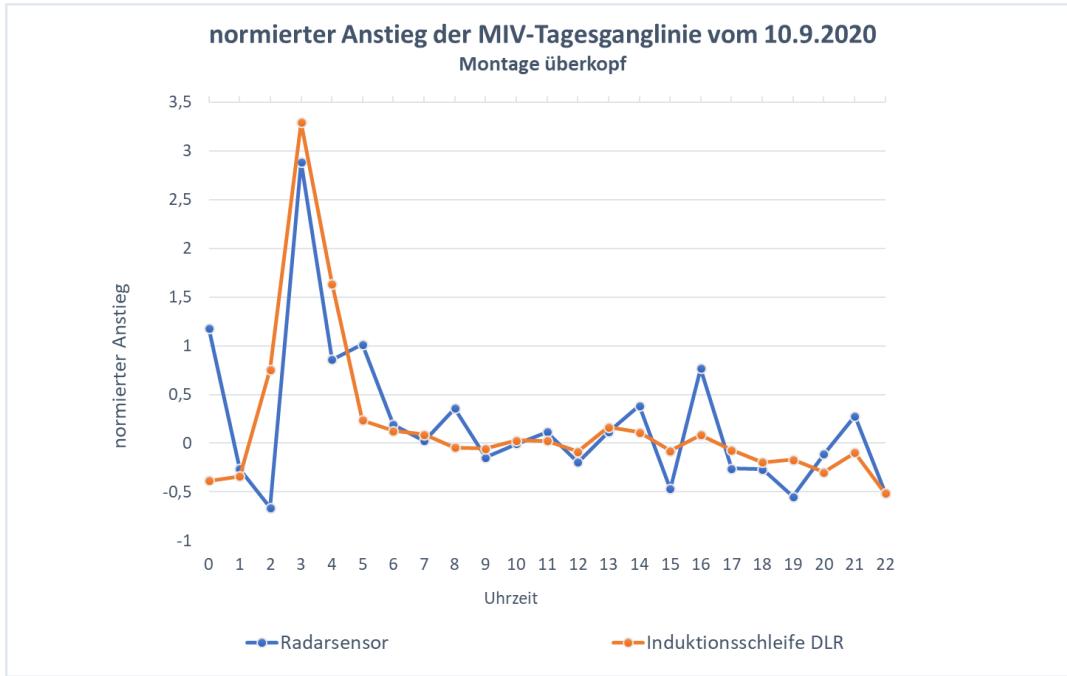


Abbildung 84: MIV-Zähler Anstiegsvergleich (Montage überkopf)

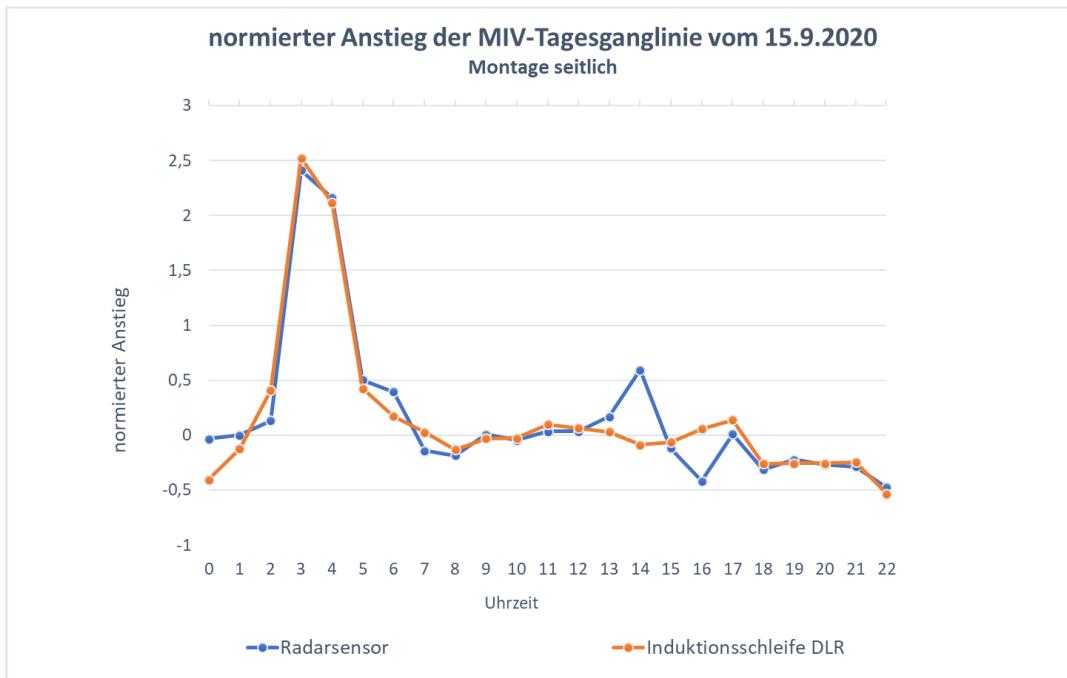


Abbildung 85: MIV-Zähler Anstiegsvergleich (Montage seitlich)

Auf beiden Diagrammen lässt die Ähnlichkeit der beiden Kurvenverläufe darauf schließen, dass der Radarsensor aussagekräftige Trendwerte liefert. Obwohl die absoluten Zahlen deutlich geringer, als die der Induktionsschleife sind, erkennt der Sensor Unterschiede zwischen geringer, mittlerer und hoher Auslastung.

Der Kurvenverlauf des Radarsensors in der seitlichen Lage ist an den meisten Stellen identisch mit dem der Induktionsschleifen. An Hauptverkehrszeiten und ggf. bei Stau (6-8 Uhr und 13-17 Uhr) hat der seitlich-montierte Radarsensor große Abweichung im Trendverlauf aufgewiesen. Er war in der Lage die steigende Anzahl des MIV zu erkennen, sobald sie aber hoch geblieben ist und indem sich Stau gebildet hat, konnte er die langsam bewegenden Fahrzeuge nicht mehr erkennen. Außerdem wurde für ihn die Sicht auf die weit liegenden Fahrspuren durch sich stauende Fahrzeuge blockiert. Sobald der Stau beendet wurde, hat sich der Trendverlauf des seitlich montiertes Radarsensors stabilisiert. Der überkopf-montierte Radarsensor reagierte weniger sensibel auf den Stau, hat aber im Übrigen höhere Abweichung im Trendverlauf als der seitlich-montierte.

Um die Aussage über die bessere Montageart zu treffen, sind Verkehrserhebungen über längere Zeit durchzuführen. Dabei wäre sinnvoll ein Radarsensor jeweils für eine oder maximal zwei Fahrspuren zu verwenden. So könnte die Stauempfindlichkeit des Detektors verringert werden. Die Radarergebnisse können mithilfe der Referenzdaten von Induktionsschleifen verbessert werden. Dafür können zahlreiche Methoden des Data Minings eingesetzt und miteinander verglichen werden. Sie können Zusammenhänge und Regelmäßigkeiten erkennen und somit die Radarergebnisse möglichst nah an die der Induktionsschleifen anpassen. Dies ist aber erst möglich, wenn eine ausreichende Datenbasis geschaffen wird. Dafür sind die Erhebungen 24/7 über mehrere Monate durchzuführen.

7 Zusammenfassung und Ausblick

Der Standortwachstum und somit stets zunehmende Verkehrsbelastung verschärft die bereits bestehende Notwendigkeit zur Optimierung der Verkehrsströme. Um Aussagen über den Mobilitätszustand treffen zu können, ist der Einsatz unterschiedlicher Technologien der automatischen Verkehrserfassung sinnvoll. All diese Technologien haben unterschiedliche Fähigkeiten und eignen sich weniger oder mehr für bestimmte Anwendungsgebiete. Der Einsatz mancher davon ist mit hohem organisatorischem und finanziellem Aufwand verbunden, was deren Verbreitung an mehreren Standorten verhindert. Das Ziel dieser Masterarbeit war ein günstiges mobiles Verkehrserfassungssystem für die Erfassung der Trends im Verkehrsfluss im Technologiepark Adlershof konzeptionell und prototypisch zu entwickeln.

Dabei sollte eine Klassifizierung nach drei Verkehrsgruppen (Fußgänger, Fahrradfahrer, MIV) stattfinden, um die Ermittlung des Modal Splits zu ermöglichen. Dafür wurden Besonderheiten bei Verkehrszählungen von jeder diesen Verkehrsgruppen analysiert. Da diese sehr unterschiedlich sind, ist es für ein einziges Gerät herausfordernd zwischen all diesen Besonderheiten zu unterscheiden. Aus diesem Grund wurde im Rahmen dieser Masterarbeit jede Verkehrsgruppe mit eigenem System getrennt erfasst.

Um eine Entscheidung über die passenden Verkehrserfassungssysteme zu treffen, wurde eine umfangreiche Literaturrecherche durchgeführt. Das Ziel dabei war die aktuellen Technologien der automatisierten Verkehrsstromanalyse zu untersuchen und miteinander zu vergleichen. Der Vergleich erfolgte anhand in der Literatur vorhandenen Kriterien. Besonders detailliert wurden Eigenschaften, Funktionsweise, Installation, Vor- und Nachteile sowie Erfassungsmöglichkeit von Verkehrsteilnehmern untersucht und beschrieben. Es wurden zehn Technologien untersucht, unter welchen neben klassischen Verkehrserhebungsmethoden wie Induktionsschleifen und Magnetfelddetektoren auch moderne Systeme der Videoerfassung sowie WiFi- und Bluetooth-Detektoren zu finden sind. Der Vergleich erfolgte stichpunktmaßig und wurde zum Schluss in +,0,- System kategorisiert, um die quantitative Vergleichbarkeit zu ermöglichen.

Als Nächstes wurden drei Verkehrserfassungssysteme für den Einsatz im Technologiepark Adlershof ausgewählt – jeweils eine für jede Verkehrsgruppe. Da bei der Auswahl der passenden Technologien mehrere Kriterien (sowohl quantitative als auch qualitative) zu betrachten waren, wurde dabei die Nutzwertanalyse als Methode zur systematischen Entscheidungsvorbereitung eingesetzt. Aus den Ergebnissen der Nutzwertanalyse folgte die Entscheidung über den Einsatz des Druckdetektors für die Erfassung von Radfahrerzahlen auf den Fahrradwegen, des Radardetektors für die

Erhebung der MIV-Mengen auf der Fahrbahn und des Ultraschalldetektors für die Erkennung der Fußgänger auf den Gehwegen. Dabei ist der Einsatz dieser Technologien ausschließlich an Standorten möglich, wo eine klare räumliche Trennung zwischen den Verkehrsteilnehmern vorhanden ist.

Als Übertragungstechnologie wurde LoRaWAN ausgewählt, die gegenüber anderen LPWAN-Netzen durch ihr offenes Lizenzmodell und hohen Maß an Flexibilität sowie großes Ökosystem an Hardware und Mitgliederanzahl überzeugt. Ihre Eignung im Mobilitätsbereich ist durch hohe Energieeffizienz und große Reichweite gegeben. Im Konzept wurden zwei Szenarien zur Integration der IoT-Anwendungen ins bereit bestehende LoRaWAN-Netz beschrieben. Zum einen wurde die Architektur einer IoT-Anwendung bei Verwendung eines offenen Netzwerkes (The Things Network) und zum anderen eines kommerziellen Netzwerkes (thingsHub) dargestellt.

Außerdem wurde im Konzept auf Montage und Witterungsschutz für die Detektoren eingegangen. Für jeden Detektor wurden Empfehlungen zur optimalen Installation mit schematischen Abbildungen erläutert. Zusätzlich wurden für Detektoren individuelle Gehäuse konzipiert, welche die Wetterbeständigkeit ermöglichen sollen. Das Konzept beinhaltet die beschriebenen Prozesse der Datenaufnahme und -übertragung, die in Form von Flussdiagrammen und schematischen Abbildungen dargestellt wurden. Darauffolgend wurden die zu übertragenen Payloads sowie Datenbankstrukturen definiert.

Anschließend wurden Empfehlungen zur Datenaufbereitung und -visualisierung gegeben. Diese sind sinnvoll, um die Qualität und der Plausibilität der erhobenen Daten sicherzustellen. Unter diesen zählen automatische Vollständigkeitsprüfung, Vergleich mit ggf. durchgeführten Befragungen oder benachbarten Zählstellen, manuelle Erhebung sowie Dokumentation der Erhebungsumstände. Außerdem wurden die Datenvisualisierungen entworfen, welche zur Datenauswertung dienen sollen. Da die erwartete absolute Messgenauigkeit der günstigen Sensoren deutlich geringer als die der professionellen Detektoren ist, wurde entschlossen die Ergebnisse als Trendverläufe darzustellen.

Das Konzept wurde in Kooperation mit dem Deutschen Zentrum für Luft- und Raumfahrt und der Beuth Hochschule umgesetzt. Die Umsetzung beinhaltete neben der Hardware- und Softwareentwicklung die Installation und Durchführung der Verkehrszählungen an der Mess- und Versuchsstrecke am Ernst-Ruska-Ufer in Adlershof. In diesem Standort ist eine klare räumliche Trennung zwischen den Verkehrsteilnehmern vorhanden und durch Nähe zum LoRaWAN-Gateway war die Integration ins vorhandene Sensornetz mit wenig Aufwand möglich. Außerdem bestand an diesem Standort Möglichkeit die Messergebnisse mit den Ergebnissen der

professionellen Detektoren des DLR zu vergleichen und somit deren Präzision zu ermitteln. Zudem war an der Messtrecke die Stromversorgung gewährleistet, was die Erhebungen über längere Zeit ermöglichte.

Das Ziel war für jede Verkehrsgruppe einen Verkehrserfassungsgerät zu bauen. Zuerst entstand aus einem an das IoT-Octopus-Board über I2C angeschlossenen Grove-Ultraschallsensor ein Fußgängerzähler, der an einem Stromkasten in direkter Nähe eines Fußgängerweges angebracht wurde. Seine Aufgabe bestand darin, den Abstand zum nächsten Objekt in seiner Erkennungszone zu messen und sobald er kleiner als gesetzter Schwellenwert war, den Fußgängercounter um eins zu erhöhen. Die Testergebnisse besagten, dass der Fußgängerzähler aussagekräftige Trendwerte in der Reichweite von einem Meter liefert. Sollte eine Person von weiterer Entfernung vorbeilaufen, konnte er sie nicht erkennen. Aus diesem Grund wurde die Empfehlung ausgesprochen auf breiteren Wegen zwei Fußgängerzähler gegenüber anbringen, um die Abdeckung der gewünschten Erkennungszone zu gewährleisten.

Der Fahrradzähler, bestehend aus einem an das IoT-Octopus-Board über AD angeschlossenen Grove-Drucksensor wurde an der Messbrücke in der unmittelbaren Nähe eines Fahrradwegs angebracht. An den Drucksensor wurde ein Gummischlauch angeknüpft, welcher über den Fahrradweg gelegt und mit Gewebe-Klebebänderstücken befestigt wurde. Das Ziel des Drucksensors war den Druckunterschied zu messen, der beim Überfahren des Gummischlauchs entsteht. Nach zahlreichen Tests und Justierungen, u.a. auch nach Anpassung des Spannungsoutputs musste festgestellt werden, dass der Grove-Drucksensor nicht in der Lage ist die minimalen Druckunterschiede, die von einem Fahrrad verursacht werden, zu erkennen. Aus diesem Grund können keine Datenauswertungen und weitere Aussagen über Präzision und Qualität der erhobenen Daten getroffen werden.

Der MIV-Zähler wurde von den Projektpartnern aus der Beuth Hochschule entwickelt. Der Grund dafür war, dass der ausgewählte Grove-Radarsensor über eine UART-Schnittstelle kommuniziert. Da diese Schnittstelle am IoT-Octopus nicht vorhanden war, wurde stattdessen das STM-Entwicklungsboard benutzt. Der Radarsensor wurde so konfiguriert, dass er alle Objekte, die sich in seiner Reichweite bewegen, erkennt und als MIV zählt. Bei der Montage des MIV-Zählers wurden zwei Optionen getestet. Zum einen die Installation überkopf an der Messbrücke und zum anderen eine seitliche Installation an Brückenleiter. Die Validierung der Radarerhebungen erfolgte durch Induktionsschleifen, die sich unter der Messbrücke befanden. Der Vergleich hat gezeigt, dass die durchschnittliche prozentuale Abweichung des Radarsensors bei Montage überkopf bei 67% und bei der seitlichen Montage bei 50% liegt. Da aber für die Datenauswertung nur die Trendwerte entscheidend sind, wurden die Verläufe der beiden Tagesganglinien analysiert. Der Vergleich hat ergeben, dass der überkopf-

montierter MIV-Zähler durchschnittlich größere Trendabweichung aufweist, aber weniger stauempfindlich, als der seitlich-montierter MIV-Zähler ist. Für die vollständige Interpretation und Vergleich der Ergebnisse sind Verkehrserhebungen über längere Zeit notwendig. Trotz absoluter Abweichungen ist der MIV-Zähler sowohl überkopf- als auch seitlich-montiert aussagekräftig genug, um zu Zwecken der Mobilitätsbedarfsanalyse verwendet zu werden.

Zusammenfassend können die kostengünstigen Grove-Sensoren in Kombination mit dem IoT-Octopus bzw. dem STM-Board für die Verkehrsstromanalyse in Adlershof eingesetzt werden. Die Präzision des Fußgänger- und des MIV-Zählers ist ausreichend, um die Trendwerte und Tageslinienverläufe in Zusammenhang mit künftiger Einführung der Mobilitätsmaßnahmen zu analysieren und Aussagen über Effektivität dieser Maßnahmen zu treffen. Zukünftige Forschung könnte an der Weiterentwicklung des Fahrradzählers anknüpfen, indem alternative Drucksensoren eingesetzt und getestet werden. Im Hinblick auf die Präzision kann es sinnvoll sein, eine Hochrechnung des tatsächlichen Verkehrsaufkommens durchzuführen, indem die Ergebnisse der Induktionsschleifen für den Radardetektor als Referenz genommen werden. Für den Fußgängerzähler könnten manuellen Ergebnisse oder Ergebnisse anderer Technologien, die eine höhere Präzision aufweisen und für die Erhebung der Fußgängerzahlen geeignet sind, als Referenz dienen. Für die Hochrechnung können unterschiedliche Verfahren der Data Mining eingesetzt werden.

Literaturverzeichnis

Arduino Funktionen. [online]

<https://www.arduino.cc/reference/de/language/functions/time/millis/>, zuletzt aktualisiert am 01.09.2020, zuletzt geprüft am 03.09.2020.

BASt (2012): *Technische Lieferbedingungen für Streckenstationen*. Hg. v. Bundesministerium für Verkehr, Bau und Stadtentwicklung.

Berlin Adlershof: Daten und Fakten. [online] <https://www.adlershof.de/adlershof-in-zahlen/>, zuletzt aktualisiert am 25.04.2019, zuletzt geprüft am 28.04.2020.

Bütler, R.; Reding, U.; Dieringer, P. (2010): *Funktionale Anforderungen an Verkehrserfassungssysteme im Zusammenhang mit Lichtsignalanlagen. Forschungsauftrag VSS 2007/303 auf Antrag des Schweizerischen Verbandes der Strassen und Verkehrs fachleute (VSS)*.

Dittmer, Gonde (1995): *Managen mit Methode. Instrumente für individuelle Lösungen*. Wiesbaden: Gabler Verlag.

ESP8266 von espressif. [online]

<https://www.espressif.com/en/products/socs/esp8266>, zuletzt geprüft am 25.08.2020.

FGSV (2010): *Hinweise zur kurzzeitigen automatischen Erfassung von Daten des Straßenverkehrs*. Ausg. 2010. Forschungsgesellschaft für Straßen- und Verkehrswesen. Köln (FGSV W1 - Wissensdokumente, FGSV-120).

FGSV (2012a): *Begriffsbestimmungen. Teil: Verkehrsplanung, Straßenentwurf und Straßenbetrieb*. Ausg. 2012. Forschungsgesellschaft für Straßen- und Verkehrswesen. Köln (FGSV).

FGSV (2012b): *Empfehlungen für Verkehrserhebungen. EVE*. Ausg. 2012. Forschungsgesellschaft für Straßen- und Verkehrswesen. Köln (FGSV R2 - Regelwerke, FGSV 125).

FGSV (2019): *Hinweise zu Detektionstechnologien im Straßenverkehr*. Ausgabe 2019. Forschungsgesellschaft für Straßen- und Verkehrswesen. Köln (FGSV W1 - Wissensdokumente).

FHWA (2016): *Traffic Monitoring Guide*. The Federal Highway Administration.

Grafana Logo. [online] Grafana.com, zuletzt geprüft am 10.08.2020.

Grove Doppler Radar. [online] <https://www.seeedstudio.com/Grove-Doppler-Radar-BGT24LTR11-p-4572.html>, zuletzt geprüft am 25.08.2020.

- Grove Homepage. [online] <https://www.seeedstudio.com/category/Grove-c-1003.html?p=2>, zuletzt geprüft am 25.08.2020.
- Grove Integrated Pressure Sensor. [online] <https://www.seeedstudio.com/Grove-Integrated-Pressure-Sensor-Kit-MPX5700AP-p-4295.html>, zuletzt geprüft am 25.08.2020.
- Grove Ultrasonic Ranger. [online] https://wiki.seeedstudio.com/Grove-Ultrasonic_Ranger/, zuletzt geprüft am 25.08.2020.
- Hering, Ekbert / Gert Schönfelder (2018): *Sensoren in Wissenschaft und Technik*. 2. Auflage. Wiesbaden: Springer Fachmedien Wiesbaden.
- influxDB Logo. [online] <https://influxdata.github.io/design.influxdata.com/branding-docs/img/influxdb/preview.svg>, zuletzt geprüft am 10.08.2020.
- Klein, Lawrence A.; Milis, Milton K.; Gibson, David R.P. (2006a): *Traffic Detector Handbook. Third Edition—Volume I*. U.S. Department of Transportation.
- Klein, Lawrence A.; Milis, Milton K.; Gibson, David R.P. (2006b): *Traffic Detector Handbook. Third Edition-Volume II*. U.S. Department of Transportation.
- Krajzewicz, Daniel; Heinrichs, Matthias; Wagner, Peter; Flötteröd Yun-Pang (2018): *Mobilität Johannisthal / Adlershof 2030. Abschlussbericht*.
- Landeshauptstadt Dresden, Straßen- und Tiefbauamt (2015): *Radverkehr pro Tag. Radverkehr auf dem Elberadweg links der Elbe am 17. Juni 2015 (Mittwoch, normaler Werktag)*. [online] <https://www.dresden.de/de/stadtraum/verkehr/radfuss/elberadweg/zahlen.php>, zuletzt geprüft am 25.05.2020.
- Linnemann, Marcel / Alexander Sommer / Ralf Leufkes (2019): *Einsatzpotentiale von LoRaWAN in der Energiewirtschaft. Praxisbuch zu Technik, Anwendung und regulatorischen Randbedingungen*. Wiesbaden: Springer Fachmedien Wiesbaden.
- Lotz, Tobias; Windmüller, Daniel (2018): *Berlin-Johannisthal/Adlershof Gebietskonkrete Verkehrsprognose 2030. Abschlussbericht (Teil 1)*.
- Node-RED Logo. [online] <https://nodered.org/about/resources/>, zuletzt geprüft am 10.08.2020.
- Raspberry Pi Logo. [online] <https://www.vtecs.de/wp-content/uploads/2019/08/RPi-Logo-Landscape-Reg-SCREEN-768x206.png>, zuletzt geprüft am 10.08.2020.
- Reimann, Alexander; Stolz, Simon (2019): *Berlin-Johannisthal/Adlershof Verfeinerung Verkehrsmengen und -verteilung sowie*

- Leistungsfähigkeitsbetrachtung zur Gebietskonkreten Verkehrsprognose 2030. Abschlussbericht (Teil 2).*
- Risse, Maude Luggen; Regli, Pascal; Leuba, Jenny; Bucheli, Dominik (2018):
Fussgänger zählen. Zählsysteme für den Fussverkehr und ihre Anwendung.
Zürich, revidierte Fassung. Hg. v. Fussverkehr Schweiz.
- Schischmanow, Adrian (2005): *Analyse des Straßenverkehrs mit verteilten opto-elektronischen Sensoren.* Humboldt-Universität zu Berlin, Berlin.
Mathematisch-Naturwissenschaftlichen Fakultät II.
- Schnieder, Eckehard; Becker, Uwe (Hg.) (2007): *Verkehrsleittechnik. Automatisierung des Straßen- und Schienenverkehrs.* Berlin, Heidelberg:
Springer-Verlag Berlin Heidelberg (VDI-Buch).
- Steierwald, Gerd; Künne, Hans Dieter; Vogt, Walter (Hg.) (2005):
Stadtverkehrsplanung. Grundlagen, Methoden, Ziele. 2., neu bearb. und erw.
Aufl. Berlin: Springer. [online] <http://lib.myilibrary.com/detail.asp?id=62752>.
- VDC (2007): *A Summary of Vehicle Detection and Surveillance Technologies used in Intelligent Transportation Systems.* Unter Mitarbeit von Luz Elena Y. Mimbela und Lawrence A. Klein. The Vehicle Detector Clearinghouse.
- Wandgehäuse Fibox. [online] <https://www.conrad.de/de/p/fibox-ta292411-wandgehaeuse-289-x-239-x-107-abs-lichtgrau-ral-7035-1-st-533233.html>, zuletzt geprüft am 31.08.2020.
- Won, Myounggyu (2019): *Intelligent Traffic Monitoring Systems for Vehicle Classification: A Survey.* [online] <http://arxiv.org/pdf/1910.04656v1.pdf>.
- Zweibrücken, Klaus; Sauter, Daniel; Schweizer, Thomas; Stäheli, Andreas; Beaujean, Katja (2005): *Erhebung des Fuss- und Veloverkehrs. SVI-Forschung 2001/503.* Schlussbericht.

Anhang

A Vergleich der Verkehrserfassungssysteme

Verkehrserfassungssysteme	Qualitätskriterien		
	Genauigkeit	Zuverlässigkeit	Vollständigkeit
Induktionsschleife	hohe Detektionsrate	Fahrzeugauflösung abhängig von Schleifenlänge	Nicht-Detektion von Fahrzeugen, die zwischen den Schleifensystemen fahren, Fahrzeuge mit geringem Metallanteil
Magnetfelddetektor	sehr hohe Detektionsrate; sehr genau bei zugewiesenen Fahrwegen	witterungs- und tageszeitunabhängig	Messfehler bei hohen Geschwindigkeiten; Nicht-Detektion von Fahrzeugen mit geringem Metallanteil
Druckdetektor	sehr hohe Detektionsrate	Zählung bei kleinen Abständen zwischen Fahrzeugen ungenau;	Nicht-Detektion beim gleichzeitigen Überfahren mehrere Fahrzeuge
Videoerfassung	hohe Detektionsrate	hohe Zuverlässigkeit	Unvollständige Daten können aufgrund der Beleuchtung sowie Witterungsabhängigkeit entstehen, aber auch im Stau
Infrarotdetektor	hohe Detektionsrate; genaue Erfassung von Fahrzeugen und Fußgängern; Abnahme der Genauigkeit bei zunehmender Aufenthaltsdauer von Fahrzeugen im Detektionsbereich	Zählung bei kleinen Abständen zwischen Fahrzeugen ungenau; bei seitlicher Montage Detektion der Fahrzeuge auf der Nachbarspur	bei Ausrichtung auf die Fahrspurmitte keine Erfassung seitlich fahrender Fahrräder; auf Gehwegen Gefahr der Nichterfassung der Randbereichen
Laserdetektor	Genauigkeit steigt mit abnehmender Geschwindigkeit der Objekte	sehr hoch	sehr hoch, annähernd 100% - Vollerfassung
Ultraschalldetektor	genaue Abstandsmessung	mittel	Unvollständigkeit nur wenn defekt

Radardetektor	hohe Detektionsrate; hohe Genauigkeit der Präsenzerkennung und der Geschwindigkeitsmessung	bei kleinen Fahrzeugabständen ungenau; bei seitlicher Montage Detektion der Fahrzeuge auf der Nachbarspur	bei Ausrichtung auf die Fahrspurmitte keine Erfassung seitlich fahrender Fahrräder; auf Gehwegen Gefahr der Nichterfassung der Randbereichen
Kombinationsdetektor	hohe Genauigkeit der Geschwindigkeitsmessung	geeignet für hohe Geschwindigkeiten; eingeschränkte Erfassung in Stop-and-Go-Situationen, bei Spurwechsel usw.	Sehr hoch aufgrund der Kombinationssensorik
Wi-Fi-, Bluetooth-Detektoren	mittel, abhängig vom Detektionsradius, Straßenkategorie und Fahrzeugtypen	Beeinträchtigung bei Überlappung der Erfassungsbereiche; Verzerrung durch Erfassung verschiedener Verkehrsmodi	Erfassung nur von Fahrzeugen/Fußgängern mit aktivierten mobilen Geräten (Erfassungsrate ca. 25-50%)

Vergleichskriterien	Verkehrserfassungssysteme									
	Induktionsschleife	Magnetfelddetektor	Druckdetektor	Videoeffassung	Infrarotdetektor	Laserdetektor	Ultraschalldetektor	Radardetektor	Kombinationsdetektor	WiFi-, Bluetooth-Detektoren
Montage- & Installationsaufwand	sehr aufwendig	aufwendig	einfach, schnell	einfach	einfach	einfach	einfach	einfach	einfach	einfach
Wartungsintensität	gering	sehr gering	gering	aufwendig	mäßig	mäßig	mäßig	mäßig	mäßig	aufwendig
Energieverbrauch	flexibel	gering		hoch	gering	hoch	gering	gering	mittel	hoch
Reichweite	je nach Schleifengröße	bis 1,5m	punkuelle Erfassung	bis 75m	20-50m	bis 150m	bis 8m	20-50m	wählbar	bis 100m
Erfassung mehrerer Fahrstreifen	nicht möglich	nicht möglich	nicht möglich	bis 2 FS	nicht möglich	bis 2 FS	nicht möglich	nicht möglich	Bis 2 FS	nicht möglich
Diskretion	sehr hoch	sehr hoch	mittel	mittel	mittel	mittel	mittel	mittel	mittel	mittel
Verschmutzung	nicht relevant	nicht relevant	nicht relevant	sehr anfällig	gering anfällig	anfällig	gering anfällig	gering anfällig	gering anfällig	gering anfällig
Datenschutz	nicht relevant	nicht relevant	nicht relevant	aufwendig	nicht relevant	nicht relevant	nicht relevant	nicht relevant	nicht relevant	aufwendig
Kosten	1.000€ bis 2.000€ + hohe Installationskosten	1.200€ bis 1.800€ + hohe Installationskosten	1.200€ bis 2.500€	mit Auswertesystem 4.000€ bis 7.500€	1.300€ bis 2.100€	6.000€ bis 11.000€	900€ bis 1.300€	900€ - 1.100€	1.300€ - 2.800€	Mit Auswertesystem 7.000€ bis 15.000€

B Code

B.a Fußgängerzähler

```
#include <lmic.h>
#include <hal/hal.h>
#include "Ultrasonic.h"
#include <Adafruit_BME680.h>
#include <Wire.h>

Ultrasonic ultrasonic(5);
const int interval = 900000; // 15 Minuten
const int day_interval = 86400000; // 24 Stunden
unsigned long previousMillis=0;
unsigned long previousDayMillis=0;
const int distance = 350; // Abstand zum nächsten Objekt
int pedestrianCounter = 0;
int temp = 0;
const int batterylevel = 50;

// BME680 Lib written by Limor Fried & Kevin Townsend for Adafruit Industries,
// http://www.adafruit.com/products/3660
Adafruit_BME680 boschBME680; // Objekt Bosch Umweltsensor
int boschBME680_ready = 0;

// LoRaWAN Copyright (c) 2015 Thomas Telkamp and Matthijs Kooijman
// (c) 2018 Terry Moore, MCCI
// https://github.com/mcci-catena/arduino-lmic
// ----- LoRa PinMapping FeatherWing Octopus
const lmic_pinmap lmic_pins = {
    .nss = 2,                                // Connected to pin D
    .rxtx = LMIC_UNUSED_PIN,                  // For placeholder only, Do not connect
    ed on RFM92/RFM95
    .rst = LMIC_UNUSED_PIN,                  // Needed on RFM92/RFM95? (probably not
) D0/GPIO16
    .dio = {
        15, 15, LMIC_UNUSED_PIN           }
};

static const u1_t PROGMEM DEVEUI[8]={
    0x13,0x13,0x13,0x13,0x13,0x13,0x13,0x13};
void os_getDevEui (u1_t* buf) {
    memcpy_P(buf, DEVEUI, 8);
}

static const u1_t PROGMEM APPEUI[8]={
```

```
0x12,0x12,0x12,0x12,0x12,0x12,0x12,0x12};  
void os_getArtEui (u1_t* buf) {  
    memcpy_P(buf, APPEUI, 8);  
}  
  
static const u1_t PROGMEM APPKEY[16]={  
    0x22, 0x22  
, 0x22, 0x22, 0x22};  
void os_getDevKey (u1_t* buf) {  
    memcpy_P(buf, APPKEY, 16);  
};  
  
volatile int LoRaWAN_Tx_Ready      = 0; // Merker für ACK  
  
int LoRaWAN_Rx_Payload = 0 ;  
// ----- LoRa Event  
void onEvent (ev_t ev) {  
    Serial.print(os_getTime());  
    Serial.print(": ");  
    switch(ev) {  
        case EV_SCAN_TIMEOUT:  
            Serial.println(F("EV_SCAN_TIMEOUT"));  
            break;  
        case EV_BEACON_FOUND:  
            Serial.println(F("EV_BEACON_FOUND"));  
            break;  
        case EV_BEACON_MISSED:  
            Serial.println(F("EV_BEACON_MISSED"));  
            break;  
        case EV_BEACON_TRACKED:  
            Serial.println(F("EV_BEACON_TRACKED"));  
            break;  
        case EV_JOINING:  
            Serial.println(F("EV_JOINING"));  
            break;  
        case EV_JOINED:  
            Serial.println(F("EV_JOINED"));  
            // Disable link check validation (automatically enabled  
            // during join, but not supported by TTN at this time).  
            LMIC_setLinkCheckMode(0);  
            break;  
        case EV_RFU1:  
            Serial.println(F("EV_RFU1"));  
            break;  
        case EV_JOIN_FAILED:  
            Serial.println(F("EV_JOIN_FAILED"));  
            break;  
        case EV_REJOIN_FAILED:  
            Serial.println(F("EV_REJOIN_FAILED"));  
    }  
}
```

```
        break;
        break;
    case EV_TXCOMPLETE:
        Serial.println(F("EV_TXCOMPLETE (includes waiting for RX windows)"));
        if (LMIC.txrxFlags & TXRX_ACK)
            Serial.println(F("Received ack"));
        if (LMIC.dataLen) {
            Serial.println(F("Received "));
            Serial.println(LMIC.dataLen);
            Serial.println(F(" bytes of payload"));
            LoRaWAN_Rx_Payload = 0;
            for (int i = 0;i<LMIC.dataLen;i++) {
                Serial.println(LMIC.frame[i+ LMIC.dataBeg],HEX);
                LoRaWAN_Rx_Payload = 256*LoRaWAN_Rx_Payload+LMIC.frame[i+ LMIC.dataBeg];
            };
        }
    }
    LoRaWAN_Tx_Ready = 1;
    // Schedule next transmission
    //os_setTimedCallback(&sendjob, os_getTime()+sec2osticks(TX_INTERVAL), do_send);
    break;
case EV_LOST_TSYNC:
    Serial.println(F("EV_LOST_TSYNC"));
    break;
case EV_RESET:
    Serial.println(F("EV_RESET"));
    break;
case EV_RXCOMPLETE:
    // data received in ping slot
    Serial.println(F("EV_RXCOMPLETE"));
    break;
case EV_LINK_DEAD:
    Serial.println(F("EV_LINK_DEAD"));
    break;
case EV_LINK_ALIVE:
    Serial.println(F("EV_LINK_ALIVE"));
    break;
case EV_TXSTART:
    Serial.println(F("EV_TXSTART"));
    break;
case EV_JOIN_TXCOMPLETE:
    Serial.println(F("EV_JOIN_TXCOMPLETE"));
    break;
default:
    Serial.println(F("Unknown event"));
    break;
}
}
```

```
void setup(){ // Einmalige Initialisierung
    Serial.begin(115200);
    delay(5000);

    // Vorbereite BME680
    boschBME680_ready = boschBME680.begin(118);

    if (boschBME680_ready == 0) {
        Serial.println("BME680 nicht vorhanden - der alte Octopus nutzt BME280, gg
f. Puzzleteile tauschen");
        while(1) delay(1);
    }

    if (!boschBME680.begin(118)) {
        Serial.println("Failed to communicate BME680");
        while (1) {
            delay(1);
        };
    }

    // Set up Bosch BME 680
    boschBME680.setTemperatureOversampling(BME680_OS_8X);
    boschBME680.setHumidityOversampling(BME680_OS_2X);
    boschBME680.setPressureOversampling(BME680_OS_4X);
    boschBME680.setIIRFilterSize(BME680_FILTER_SIZE_3);
    boschBME680.setGasHeater(320, 150); // 320*C for 150 ms

    // -- Initialisiere LoRaWAN
    os_init();           // LMIC LoRaWAN
    LMIC_reset();        // Reset the MAC state
    LMIC.txpow = 27;     // Maximum TX power
    LMIC.datarate=DR_SF12; // Long Range
    LMIC.rps = updr2rps(LMIC.datarate);

    Wire.begin(); // ---- Initialisiere den I2C-Bus
    if (Wire.status() != I2C_OK) Serial.println("Something wrong with I2C");
}

void loop() { // Kontinuierliche Wiederholung

    temp=round(boschBME680.readTemperature());
    unsigned long currentMillis = millis();

    //15 Minuten abgelaufen?
    if ((unsigned long)(currentMillis - previousMillis) < interval){      int Ran
geInCentimeters = 0;
```

```

RangeInCentimeters = ultrasonic.MeasureInCentimeters();
if(RangeInCentimeters < distance && RangeInCentimeters != 0 && RangeInCentimeters != 350){
    pedestrianCounter++;
    Serial.println("FUßGÄNGER ERFASST! Anzahl der Fußgänger: ");
    Serial.println(pedestrianCounter);
    Serial.println("Aktuelle Sensortemperatur: ");
    Serial.println(temp);
    delay(1000); // Nachdem ein Fußgänger erfasst wird, 1 Sekunde nicht
                  // messen, um mehrfache Erfassung zu vermeiden
}
delay(500); // Messung alle 0,5 Sekunden
}

else{
    if ((unsigned long)(currentMillis - previousDayMillis) >= day_interval){
        previousDayMillis = currentMillis;
        { //Block----- sende Daten an TTN
            int port = 11;
            uint8_t measurement[1];

            measurement[0] = batterylevel;

            // Check if there is not a current TX/RX job running
            //if (LMIC.opmode & OP_TXRXPEND) {
            if (LMIC.opmode & (1 << 7)) {
                Serial.println(F("OP_TXRXPEND, not sending"));
            }
            else {
                // Prepare upstream data transmission at the next possible time.
                LoRaWAN_Tx_Ready = 0; // Merker für ACK
                LMIC_setTxData2(port, measurement, sizeof(measurement), 0); // Sende

                Serial.println(F("Packet queued"));
                while(LoRaWAN_Tx_Ready==0) {
                    yield();
                    os_runloop_once();
                }; // Warte bis gesendet
            }
        } // Blockende
    }

    previousMillis = currentMillis; // Alle 15 Minuten die Zeit festhalten
    temp=round(boschBME680.readTemperature());
    Serial.println("Zeit ist um! Anzahl der Fußgänger der letzten 15 Minuten: ")
;
    Serial.println(pedestrianCounter);
    Serial.println("Aktuelle Sensortemperatur: ");
}

```

```
Serial.println(temp);

{ //Block----- sende Daten an TTN
    int port = 10;
    uint8_t measurement[2];

    measurement[0] = pedestrianCounter;
    measurement[1] = temp;

    // Check if there is not a current TX/RX job running
    //if (LMIC.opmode & OP_TXRPEND) {
    if (LMIC.opmode & (1 << 7)) {
        Serial.println(F("OP_TXRPEND, not sending"));
    }
    else {
        // Prepare upstream data transmission at the next possible time.
        LoRaWAN_Tx_Ready = 0; // Merker für ACK
        LMIC_setTxData2(port, measurement, sizeof(measurement), 0); // Sende

        Serial.println(F("Packet queued"));
        while(LoRaWAN_Tx_Ready==0) {
            yield();
            os_runloop_once();
        }; // Warte bis gesendet
    }
} // Blockende
pedestrianCounter = 0;
temp = 0;
}

}
```

B.b MIV-Zähler

bsp.c

```
/**  
 * *****  
 * @file      bsp.c  
 * @author    MCD Application Team  
 * @brief     manages the sensors on the application  
 * *****  
 *  
 * @attention  
 *  
 * <h2><center>&copy; Copyright (c) 2018 STMicroelectronics.  
 * All rights reserved.</center></h2>  
 *  
 * This software component is licensed by ST under Ultimate Liberty license  
 * SLA0044, the "License"; You may not use this file except in compliance with  
 * the License. You may obtain a copy of the License at:  
 *  
 *          www.st.com/SLA0044  
 *  
 * *****  
 *  
 */  
  
/* Includes -----  
*/  
#include <string.h>  
#include <stdlib.h>  
#include "hw.h"  
#include "timeServer.h"  
#include "bsp.h"  
#include "GBT24LTR11.h"  
#if defined(LRWAN_NS1)  
#include "lrwan_ns1_humidity.h"  
#include "lrwan_ns1_pressure.h"  
#include "lrwan_ns1_temperature.h"  
#else /* not LRWAN_NS1 */  
#if defined(SENSOR_ENABLED)  
#if defined (X_NUCLEO_IKS01A1)  
#warning "Do not forget to select X_NUCLEO_IKS01A1 files group instead of X_NU  
CLEO_IKS01A2"  
#include "x_nucleo_iks01a1_humidity.h"  
#include "x_nucleo_iks01a1_pressure.h"  
#include "x_nucleo_iks01a1_temperature.h"  
#else /* not X_NUCLEO_IKS01A1 */  
#include "x_nucleo_iks01a2_humidity.h"  
#include "x_nucleo_iks01a2_pressure.h"
```

```
#include "x_nucleo_iks01a2_temperature.h"
#ifndef /* X_NUCLEO_IKS01A1 */
#ifndef /* SENSOR_ENABLED */
#ifndef /* LRWAN_NS1 */

/* Private typedef -----*/
*/
extern measure_TypeDef measure;
extern UART_HandleTypeDef huart1;

/* Private define -----*/
*/
#define STSOP_LATITUDE ((float) 43.618622 )
#define STSOP_LONGITUDE ((float) 7.051415 )
#define MAX_GPS_POS ((int32_t) 8388607 ) // 2^23 - 1
/* Private macro -----*/
*/
/* Private variables -----*/
*/
/* Private function prototypes -----*/
*/
/* Exported functions -----*/
-*/

/* Private variables -----*/
*/
uint8_t t=0;

void BSP_sensor_Read(sensor_t *sensor_data)
{
    uint8_t i=0;

    if(t==0)
    {
        if(GBT24LTR11_init())
        {
            PRINTF("\n\nOK, THRESOLD SET\nThreshold: %d\n\n\n",measure.threshold);
        }
    }
}
```

```
        PRINTF("OK, SPEED RANGE SET\nSpeedmax: %d\nSpeedmin: %d\n\n",measure.speed_max,measure.speed_min);

    }

    t=1;
}

uint8_t state=receive(0);
/*if(state==0)
{
    PRINTF("Is not answering");
}*/
if(state==1)
{
    sensor_data->count=measure.count;
    sensor_data->speed=measure.speed;
    sensor_data->state=measure.state;
}

if(state==2)
{
    PRINTF("WRONG DATA!");
}
}

void  BSP_sensor_Init()
{
/* USER CODE BEGIN 6 */

    //INIT
    measure.huart=&huart1;
    measure.threshold=200;
    measure.speed_min=0;
    measure.speed_max=1700;

    //FUNCTIONS

    /* USER CODE END 6 */
}

***** (C) COPYRIGHT STMicroelectronics *****END OF FILE****
*/
```

```
/*
Copyright (C) Preis Ingenieurbüro GmbH

Licensed under the Apache License, Version 2.0 (the "License").
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
*/
```

GBT24LTR11.c

```
/*
 * GBT24LTR11.c
 *
 * Created on: Aug 25, 2020
 *      Author: Chris
 */
/*includes*/
#include "GBT24LTR11.h"
#include "stm32l0xx_hal.h"
#include "util_console.h"
//#include "stm32f3xx_hal.h"

/*
 * Structure Prototype
 *
 */
volatile measure_TypeDef measure;

/*void GBT24LTR11_init(measure_TypeDef measure)
{
    measure.huart;
}*/



uint8_t ready; //flag for being ready Command C1 (Receive)
uint8_t ready_1; //flag for being ready command C3 (SET SPEED)
uint8_t ready_2; //flag for being ready command C7 (SET THRESOLD)
uint8_t data[10] = {};//RX Buffer

uint8_t GBT24LTR11_init()
{
    if(setSpeed()&&setThreshold())return 1;

    return 0;
}
```

```
uint8_t receive(uint8_t appleav)
{
    uint8_t i=0;

    if(i==0)
    {
        HAL_UART_Receive_IT(measure.huart,data, 10);
        i=1;
    }

    if(ready==1)//before else
    {
        ready=0;
        //return 1; //Return 1 == Received
        HAL_Delay(WAIT); //TIME TO WAIT IN MS

        //Targetstate
        if(data[0]==GBT24LTR11_MESSAGE_HEAD)
        {
            if (data[1]==GBT24LTR11_SEND_ADDRESS)
            {
                if(data[2]==GBT24LTR11_COMMAND_GET_TARGET)
                {
                    if(appleav==0)
                    {
                        if(data[7]==2 || data[7]==1)
                        {
                            //x=1;
                            measure.count++;
                            PRINTF("COUNTER:%d\n",measure.count);
                            HAL_UART_Receive_IT(measure.huart,data, 10);
                            ready=0;
                            if(data[7]==2)measure.app++;
                            if(data[7]==1)measure.leave++;

                        }
                    }
                    if(appleav==1)
                    {
                        uint8_t x=0;
                        if(data[7]==2) x=1;

                        if(data[7]==1&&x==1)
                        {
                            x=0;
                            measure.count++;
                        }
                    }
                }
            }
        }
    }
}
```

```

        }
    }
    /*
    if(data[7]==1&&x==1)
    {
        x=0;
        count++;
        PRINTF("COUNTER:%d\n",count);
        measure->count=count;

    }/*
    measure.state= data[7];
    measure.speed=(data[5] * 256 + data[6]);
    return 1;

}
return 2; //WRONG DATA
}
return 2;
}
return 2;
}

return 0;
}

/*SET SPEED IN CM/S
Parameter: speed_min::minimal detected speed
            speed_max::maximal detected speed
Return: 0 NO/WRONG ANSWER
        1 READY

*/
uint8_t setSpeed()
{
    uint16_t len=0;
    unsigned char commandc3[11] = {0x55, 0x2A, 0xC3, 0x00, 0x06, 0x02, 0x09, 0
x01, 0x03, 0x01, 0x57};

    commandc3[5] = measure.speed_max/256;
    commandc3[6] = measure.speed_max%256;

    commandc3[7]=measure.speed_min/256;
    commandc3[8]=measure.speed_min%256;

    for(uint8_t i=0;i<9;i++) len+=commandc3[i];
    commandc3[9]=len/256;
    commandc3[10]=len%256;
}
```

```
ready_1=0;
uint8_t i=0;
do
{
    HAL_UART_Receive_IT(measure.huart,data, 11);
    HAL_UART_Transmit(measure.huart,commandc3,11,100);
    HAL_Delay(25);
}while(ready_1==0&&i<100);

if(ready_1==1)
{
    if(((data[5] * 256 + data[6]) == measure.speed_max) && ((data[7] * 256
+ data[8]) == measure.speed_min))
    {
        return 1; //OK
    }
}
return 0; //ERROR
}

uint8_t setThreshold()
{
    ready_2=0;
    int len = 0;
    unsigned char commandc7[11] = {0x55, 0x2A, 0xC7, 0x00, 0x06, 0x00, 0x00, 0
x04, 0x00, 0x01, 0x50};
    commandc7[5] = measure.threshold / 256 / 256 / 256 % 256;
    commandc7[6] = measure.threshold/ 256 / 256 % 256;
    commandc7[7] = measure.threshold / 256 % 256;
    commandc7[8] = measure.threshold % 256;
    for (int i = 0; i < 9; i++)
    {
        len += commandc7[i];
    }
    commandc7[9] = len / 256 % 256;
    commandc7[10] = len % 256;
    uint8_t i=0;
    do
    {
        HAL_UART_Receive_IT(measure.huart, data, 11);
        HAL_UART_Transmit(measure.huart, commandc7, 11, 100);
    }while(ready_2==0&&i<100);

    uint16_t thr = data[5] * 256 * 256 * 256 + data[6] * 256 * 256 + data[7] *
256 + data[8];
}
```

```
    if(ready_2==1&&thr==measure.threshold)
    {
        return 1; //OK
    }
    return 0; //ERROR
}

void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
    if(huart==measure.huart)
    {
        if(data[2]==0xC3)
        {
            ready_1=1;
            //PRINTF("OK, SPEED RANGE SET\n");
        }
        if(data[2]==0xC1)
        {
            //HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, RESET);
            ready=1;
        }
        if(data[2]==0xC7)
        {
            ready_2=1;
            //PRINTF("OK, THRESOLD SET\n");
        }
    }
    return;
}
```

Main.c

```
/**  
*****  
*  
* @file    main.c  
* @author  MCD Application Team  
* @brief   this is the main!  
*****  
*  
* @attention  
*  
* <h2><center>&copy; Copyright (c) 2018 STMicroelectronics.  
* All rights reserved.</center></h2>  
*  
* This software component is licensed by ST under Ultimate Liberty license  
* SLA0044, the "License"; You may not use this file except in compliance with  
* the License. You may obtain a copy of the License at:  
*           www.st.com/SLA0044  
*  
*****  
*  
*/  
  
/* Includes -----  
*/  
#include "hw.h"  
#include "low_power_manager.h"  
#include "lora.h"  
#include "bsp.h"  
#include "timeServer.h"  
#include "vcom.h"  
#include "version.h"  
#include "hw_rtc.h"  
#include "stm32l0xx_hal.h"  
// Sensoren  
#include "GBT24LTR11.h"  
  
#include "stdio.h"  
#include "string.h"  
  
/*  
 * Erik Zinger  
 * 10.08.2020  
 *  
 */
```

```
//uint16_t counter;

/* Private typedef -----
 */

sensor_t sensor_data;
UART_HandleTypeDef huart1;
extern measure_TypeDef measure;

RTC_DateTypeDef sDate;
RTC_TimeTypeDef sTime;
/* Private define -----
 */

#define LORAWAN_MAX_BAT    254

/*!
 * Defines the application data transmission duty cycle. 5s, value in [ms].
 */
#define APP_TX_DUTYCYLE          900000 //90000
    // xxx
/*!
 * LoRaWAN Adaptive Data Rate
 * @note Please note that when ADR is enabled the end-device should be static
 */
#define LORAWAN_ADR_STATE LORAWAN_ADR_ON
/*!
 * LoRaWAN Default data Rate Data Rate
 * @note Please note that LORAWAN_DEFAULT_DATA_RATE is used only when ADR is disabled
 */
#define LORAWAN_DEFAULT_DATA_RATE DR_0
/*!
 * LoRaWAN application port
 * @note do not use 224. It is reserved for certification
 */
#define LORAWAN_APP_PORT         10
/*!
 * LoRaWAN default endNode class port
 */
#define LORAWAN_DEFAULT_CLASS     CLASS_A
/*!
 * LoRaWAN default confirm state
 */
#define LORAWAN_DEFAULT_CONFIRM_MSG_STATE LORAWAN_UNCONFIRMED_MSG
/*!
 * User application data buffer size
```

```
/*
#define LORAWAN_APP_DATA_BUFF_SIZE           64
*/
/*!
 * User application data
 */
static uint8_t AppDataBuff[LORAWAN_APP_DATA_BUFF_SIZE];

/*!
 * User application data structure
 */
//static lora_AppData_t AppData={ AppDataBuff, 0 ,0 };
lora_AppData_t AppData = { AppDataBuff, 0, 0 };

/* Private macro -----
*/
/* Private function prototypes -----
*/
void MX_USART1_UART_Init(void);

/* call back when LoRa endNode has received a frame*/
static void LORA_RxData(lora_AppData_t *AppData);

/* call back when LoRa endNode has just joined*/
static void LORA_HasJoined(void);

/* call back when LoRa endNode has just switch the class*/
static void LORA_ConfirmClass(DeviceClass_t Class);

/* call back when server needs endNode to send a frame*/
static void LORA_TxNeeded(void);

/* callback to get the battery level in % of full charge (254 full charge, 0 no charge)*/
static uint8_t LORA_GetBatteryLevel(void);

/* LoRa endNode send request*/
static void Send(void *context);

/* start the tx process*/
static void LoraStartTx(TxEventType_t EventType);

/* tx timer callback function*/
static void OnTxTimerEvent(void *context);

/* tx timer callback function*/
static void LoraMacProcessNotify(void);
```

```
/* Private variables -----*/
*/
/* load Main call backs structure*/
static LoRaMainCallback_t LoRaMainCallbacks = { LORA_GetBatteryLevel,
    HW_GetTemperatureLevel,
    HW_GetUniqueId,
    HW_GetRandomSeed,
    LORA_RxData,
    LORA_HasJoined,
    LORA_ConfirmClass,
    LORA_TxNeeded,
    LoraMacProcessNotify
};

LoraFlagStatus LoraMacProcessRequest = LORA_RESET;
LoraFlagStatus AppProcessRequest = LORA_RESET;
/*!
 * Specifies the state of the application LED
 */
static uint8_t AppLedStateOn = RESET;

static TimerEvent_t TxTimer;

#ifndef USE_B_L072Z_LRWAN1
/*!
 * Timer to handle the application Tx Led to toggle
 */
static TimerEvent_t TxLedTimer;
static void OnTimerLedEvent(void *context);
#endif
/* !
 *Initialises the Lora Parameters
 */
static LoRaParam_t LoRaParamInit = {
    LORAWAN_ADR_STATE,
    LORAWAN_DEFAULT_DATA_RATE,
    LORAWAN_PUBLIC_NETWORK
};

uint8_t d=0;
uint8_t counter;
uint16_t len;
char uartBuff[100];

#define MAX_FILE_SIZE    1000
```

```
/* Private functions -----
*/

/** 
 * @brief Main program
 * @param None
 * @retval None
 */
int main(void)
{
    /* STM32 HAL library initialization*/
    HAL_Init();
    /* Configure the system clock*/
    SystemClock_Config();

    /* Configure the debug mode*/
    DBG_Init();

    /* Configure the hardware*/
    HW_Init();
    MX_USART1_UART_Init();

    /* USER CODE BEGIN 1 */

    /* MX_SPI2_Init(); */

    */

    BSP_LED_Init(LED_BLUE);

    // Disable Stand-by mode
    LPM_SetOffMode(LPM_APPLI_Id, LPM_Disable);

    PRINTF("EdgeCity V1.0 Build 31.08.2020\n");
    PRINTF("APP_VERSION= %02X.%02X.%02X.%02X\n", (uint8_t)(__APP_VERSION >> 24),
), (uint8_t)(__APP_VERSION >> 16), (uint8_t)(__APP_VERSION >> 8), (uint8_t)__APP_VERSION);
    PRINTF("MAC_VERSION= %02X.%02X.%02X.%02X\n", (uint8_t)(__LORA_MAC_VERSION
>> 24), (uint8_t)(__LORA_MAC_VERSION >> 16), (uint8_t)(__LORA_MAC_VERSION >> 8
), (uint8_t)__LORA_MAC_VERSION);

    // Configure the Lora Stack
    LORA_Init(&LoRaMainCallbacks, &LoRaParamInit);


```

```
LORA_Join();

LoraStartTx(TX_ON_TIMER) ;
BSP_sensor_Init(&sensor_data);

//STATIC_DEVICE_EUI
while (1)
{
    BSP_sensor_Read(&sensor_data);
    if (AppProcessRequest == LORA_SET)
    {
        /*reset notification flag*/
        AppProcessRequest = LORA_RESET;
        /*Send*/
        Send(NULL);
    }
    if (LoraMacProcessRequest == LORA_SET)
    {
        /*reset notification flag*/
        LoraMacProcessRequest = LORA_RESET;
        LoRaMacProcess();
    }
    /*If a flag is set at this point, mcu must not enter low power and must
    loop*/
    DISABLE_IRQ();

    /* if an interrupt has occurred after DISABLE_IRQ, it is kept pending
     * and cortex will not enter low power anyway */
    if ((LoraMacProcessRequest != LORA_SET) && (AppProcessRequest != LORA_SET))
    {
#ifndef LOW_POWER_DISABLE
        //LPM_EnterLowPower();
#endif
    }

    ENABLE_IRQ();

    /* USER CODE BEGIN 2 */
    /* USER CODE END 2 */
}
}

void LoraMacProcessNotify(void)
{
```

```
LoraMacProcessRequest = LORA_SET;
}

static void LORA_HasJoined(void)
{
#if( OVER_THE_AIR_ACTIVATION != 0 )
    PRINTF("JOINED\n");
#endif
    LORA_RequestClass(LORAWAN_DEFAULT_CLASS);
}

static void Send(void *context)
{

// xxx hier werden alle Sensoren eingelesen:
BSP_sensor_Read(&sensor_data);

SysTime_t stime = SysTimeGetMcuTime();

// Output:

// save_data(&sensor_data);

// können die Daten per Lora versendet werden?
if (LORA_JoinStatus() != LORA_SET)
{
    /*Not joined, try again later*/
    PRINTF("Not joined, try again later\n");
    LORA_Join();
    return;
}
else
    PRINTF("Lora joined\n");

BSP_LED_On(LED_BLUE);

TVL1(PRINTF("SEND REQUEST\n"));

#ifndef USE_B_L072Z_LRWAN1
    TimerInit(&TxLedTimer, OnTimerLedEvent);
    TimerSetValue(&TxLedTimer, 200);

```

```
LED_On(LED_RED1) ;

    TimerStart(&TxLedTimer);
#endif

AppData.Port = LORAWAN_APP_PORT;

counter++;
RTC_DateTypeDef sDate;
RTC_TimeTypeDef sTime;

HW_RTC_GetCalendarValue( &sDate, &sTime);

/*uint32_t hour = sTime.Hours;
uint32_t minutes = sTime.Minutes;
uint32_t seconds = sTime.Seconds;
PRINTF("LORA_send: Nr.%d %02d:%02d:%02d\n\n", counter, hour, minutes, seconds );
AppData.Buff[0] = (counter >> 8) & 0xFF;
AppData.Buff[1] = counter & 0xFF;
AppData.Buff[2] = hour;
AppData.Buff[3] = minutes;
AppData.Buff[4] = seconds;*/

PRINTF("COUNTER=%d\n",sensor_data.count);
AppData.Buff[0] = (sensor_data.count >> 24) & 0x000000FF;
AppData.Buff[1] = (sensor_data.count >> 16) & 0x000000FF;
AppData.Buff[2] = (sensor_data.count>> 8) & 0x000000FF;
AppData.Buff[3] = sensor_data.count & 0x000000FF;

AppData.BuffSize = 4;

LORA_send(&AppData, LORAWAN_DEFAULT_CONFIRM_MSG_STATE);
measure.count=0;
sensor_data.count=0;

/* USER CODE END 3 */
}

static void LORA_RxData(lora_AppData_t *AppData)
{
/* USER CODE BEGIN 4 */
```

```
PRINTF("PACKET RECEIVED ON PORT %d\n\r", AppData->Port);

switch (AppData->Port)
{
case 3:
    /*this port switches the class*/
    if (AppData->BuffSize == 1)
    {
        switch (AppData->Buff[0])
        {
        case 0:
        {
            LORA_RequestClass(CLASS_A);
            break;
        }
        case 1:
        {
            LORA_RequestClass(CLASS_B);
            break;
        }
        case 2:
        {
            LORA_RequestClass(CLASS_C);
            break;
        }
        default:
            break;
        }
    }
    break;
case LORAWAN_APP_PORT:
    if (AppData->BuffSize == 1)
    {
        AppLedStateOn = AppData->Buff[0] & 0x01;
        if (AppLedStateOn == RESET)
        {
            PRINTF("LED OFF\n");
            LED_Off(LED_BLUE) ;
        }
        else
        {
            PRINTF("LED ON\n");
            LED_On(LED_BLUE) ;
        }
    }
    break;
default:
    break;
}
```

```
/* USER CODE END 4 */
}

static void OnTxTimerEvent(void *context)
{
    /*Wait for next tx slot*/
    TimerStart(&TxTimer);

    AppProcessRequest = LORA_SET;
}

static void LoraStartTx(TxEventType_t EventType)
{
    if (EventType == TX_ON_TIMER)
    {
        /* send everytime timer elapses */
        TimerInit(&TxTimer, OnTxTimerEvent);
        TimerSetValue(&TxTimer, APP_TX_DUTYCYCLE);
        OnTxTimerEvent(NULL);
    }
    else
    {
        /* send everytime button is pushed */
        GPIO_InitTypeDef initStruct = {0};

        initStruct.Mode = GPIO_MODE_IT_RISING;
        initStruct.Pull = GPIO_PULLUP;
        initStruct.Speed = GPIO_SPEED_HIGH;

        HW_GPIO_Init(USER_BUTTON_GPIO_PORT, USER_BUTTON_PIN, &initStruct);
        HW_GPIO_SetIrq(USER_BUTTON_GPIO_PORT, USER_BUTTON_PIN, 0, Send);
    }
}

static void LORA_ConfirmClass(DeviceClass_t Class)
{
    PRINTF("switch to class %c done\n", "ABC"[Class]);

    /*Optionnal*/
    /*informs the server that switch has occurred ASAP*/
    AppData.BuffSize = 0;
    AppData.Port = LORAWAN_APP_PORT;

    LORA_send(&AppData, LORAWAN_UNCONFIRMED_MSG);
}

static void LORA_TxNeeded(void)
{
    AppData.BuffSize = 0;
```

```
    AppData.Port = LORAWAN_APP_PORT;

    LORA_send(&AppData, LORAWAN_UNCONFIRMED_MSG);
}

/***
 * @brief This function return the battery level
 * @param none
 * @retval the battery level 1 (very low) to 254 (fully charged)
 */
uint8_t LORA_GetBatteryLevel(void)
{
    uint16_t batteryLevelmV;
    uint8_t batteryLevel = 0;

    batteryLevelmV = HW_GetBatteryLevel();

    /* Convert batterey level from mV to linea scale: 1 (very low) to 254 (fully charged) */
    if (batteryLevelmV > VDD_BAT)
    {
        batteryLevel = LORAWAN_MAX_BAT;
    }
    else if (batteryLevelmV < VDD_MIN)
    {
        batteryLevel = 0;
    }
    else
    {
        batteryLevel = (((uint32_t)(batteryLevelmV - VDD_MIN) * LORAWAN_MAX_BA
T) / (VDD_BAT - VDD_MIN));
    }

    return batteryLevel;
}

#endif USE_B_L072Z_LRWN1
static void OnTimerLedEvent(void *context)
{
    LED_Off(LED_RED1) ;
}
#endif
```

```
***** (C) COPYRIGHT STMicroelectronics *****END OF FILE***  
*/
```

```
void MX_USART1_UART_Init(void)
{
    huart1.Instance = USART1;
    huart1.Init.BaudRate = 115200;
    huart1.Init.WordLength = UART_WORDLENGTH_8B;
    huart1.Init.StopBits = UART_STOPBITS_1;
    huart1.Init.Parity = UART_PARITY_NONE;
    huart1.Init.Mode = UART_MODE_TX_RX;
    huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart1.Init.OverSampling = UART_OVERSAMPLING_16;
    huart1.Init.OneBitSampling = UART_ONE_BIT_SAMPLE_DISABLE;
    huart1.AdvancedInit.AdvFeatureInit = UART_ADVFEATURE_NO_INIT;
    if (HAL_UART_Init(&huart1) != HAL_OK)
    {
        Error_Handler();
    }
}
```

```
/*
 ****
 *
 * @file    main.c
 * @author  MCD Application Team
 * @brief   this is the main!
 ****
 *
 * @attention
 *
 * <h2><center>&copy; Copyright (c) 2018 STMicroelectronics.
 * All rights reserved.</center></h2>
 *
 * This software component is licensed by ST under Ultimate Liberty license
 * SLA0044, the "License"; You may not use this file except in compliance with
 * the License. You may obtain a copy of the License at:
 *           www.st.com/SLA0044
 *
 ****
 *
 */
/* Includes -----
*/
#include "hw.h"
#include "low_power_manager.h"
#include "lora.h"
#include "bsp.h"
#include "timeServer.h"
#include "vcom.h"
#include "version.h"
#include "hw_rtc.h"
#include "stm32l0xx_hal.h"
// Sensoren
#include "GBT24LTR11.h"

#include "stdio.h"
#include "string.h"

/*
 *  Erik Zinger
 *  10.08.2020
 *
 */
//uint16_t counter;
```

Eidesstattliche Versicherung

Ich versichere hiermit, dass ich die vorliegende Masterthesis selbstständig und ohne fremde Hilfe angefertigt und keine andere als die angegebene Literatur benutzt habe. Alle von anderen Autoren wörtlich übernommenen Stellen wie auch die sich an die Gedankengänge anderer Autoren eng anlehnenden Ausführungen meiner Arbeit sind besonders gekennzeichnet. Diese Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Berlin, den 22. September 2020

A handwritten signature in black ink, appearing to read "K. P." followed by a stylized surname.