**Pflichtenheft**

**Fahrzeug zur Infrastruktur Kollaboration**

1. **Zielbestimmungen**

**Ziel: autonome Funktion für Fahrzeug entwickeln. Beinhaltet: siehe Aufgabenstellung.**

MUSS Kriterien:

-Eine Verbindung zwischen einem Fahrzeug und einer Ampel über eine Streckenabschnittskontrolle muss herstellbar sein

-Ampel verteilt Informationen über Ampelzustand an das Fahrzeug

-Umsetzung mittels Mikrocontroller und geeigneter Netzwerktechnologie

-zwei Prototypen sollen realisiert werden

-Prototyp A basierend auf mbot Technologie

-Prototyp B basierend auf E-Scooter

-Erstellung eines Projketplans

-Zwischenpräsentation und Finale Präsentation

WUNSCH Kriterien:

-RTW kann sicher mit der Zentrale kommunizieren

-Zentrale ermittelt die Ankunftszeit des RTW und schaltet rechtzeitig die Ampel

-RTW kann sicher und ohne zusätzlichen Halt die Ampelkreuzung passieren

-Entlang des Weges des RTW´s kann eine grüne Welle geschaltet werden

ABGRENZUNGS Kriterien:

-kein komplexer Verkehr

-nur eine Ampel

-maximal ein Fahrzeug sowie RTW

1. **Einsatz**

Zielgruppen: Die Zielgruppe unseres Szenarios, sind alle Verkehrsteilnehmer die ein autonomes motorisiertes Fahrzeug mit Straßenzulassung besitzen und sich im Stadtverkehr (da wo es Ampeln gibt) bewegen.

Anwendungsbereiche: Da wo Ampeln sind und sich eine Geschwindigkeitsanpassung lohnt bzw. sich positiv auf die Energieeffizienz des Autos auswirkt.

Betriebsbedingungen: Eine Ampel und ein Verkehrsteilnehmer sollen über zwei Mikrocontroller (Constraint) miteinander kommunizieren und sich Daten, wie z.B. die Taktzyklen der Ampel durchgeben.

Physikalische Umgebung: In der Umwelt auf einer Straße mit Ampel

Betriebszeit: Die Betriebszeit ist gleich der Betriebszeit der Ampel.

Dies kommt auf die Ampel an.

Qualifik. d. Benutzergruppen: Die Benutzergruppe muss im Besitz einer gültigen Fahrerlaubnis sein (Amtlicher Führerschein).

1. **Produktübersicht**

**3.1 Stakeholder Analyse**

**Interne Stakeholder**

Projektleiterin: Christine Agethen

Mitarbeiter: Studierende

Christine Agethen

Daniel Münstermann

Christopher Beck

Johannes Schäfer

Rojda Takmaz

Melissa Sens

Alexander Wilms

Beratung: Professor Dr. Stefan Henkler

Eigentümer: Hochschule Hamm Lippstadt

Professor Dr. Stefan Henkler

Christine Agethen

Daniel Münstermann

Christopher Beck

Johannes Schäfer

Rojda Takmaz

Melissa Sens

Alexander Wilms

**Externe Stakeholder:**

Auftraggeber: Hochschule Hamm Lippstadt

Professor Dr. Stefan Henkler

Kunden: Studierende des Studiengangs Interaktionstechnik und Design

Studierende der Hochschule Hamm Lippstadt

Studierende anderer Hochschulen und Universitäten

Schüler/ Abiturienten

Professoren

Technikinteressierte Personen

Technikunternehmen

Geldgeber/

Lieferanten: über die Hochschule Hamm Lippstadt

Öffentlichkeit: Presse

Zeitung/Flyer

Hochschulwebsite

Öffentliche Blogbeiträge

Github

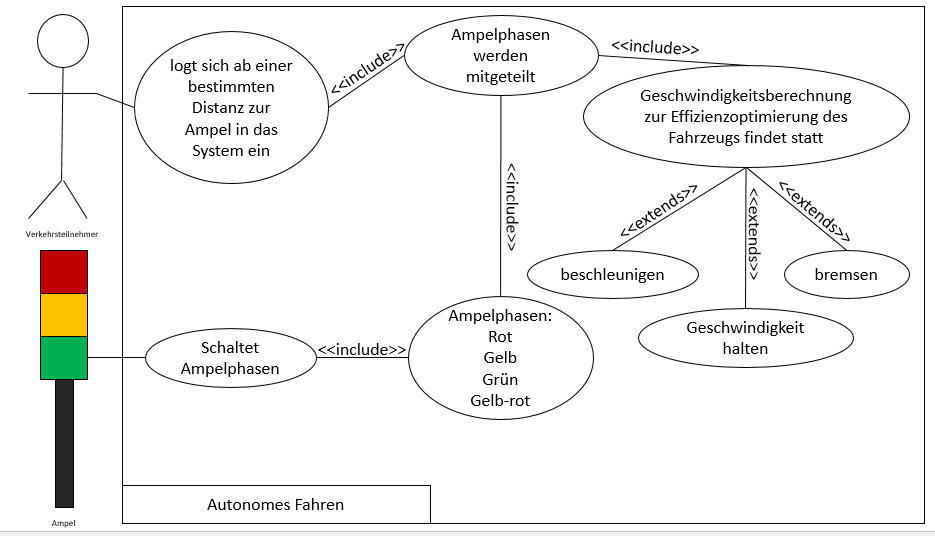
Präsentation

Vorstellung an der Hochschule Hamm Lippstadt z.B. Tag der offenen Tür

* 1. **Use-Case**

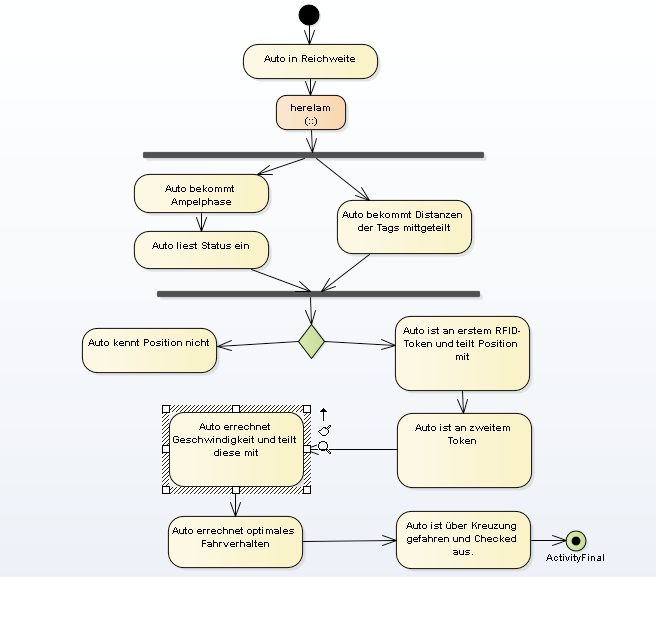
**Verkehrsteilnehmer: Fahrzeug oder Fahrer (sollte das Fahrzeug sein)**

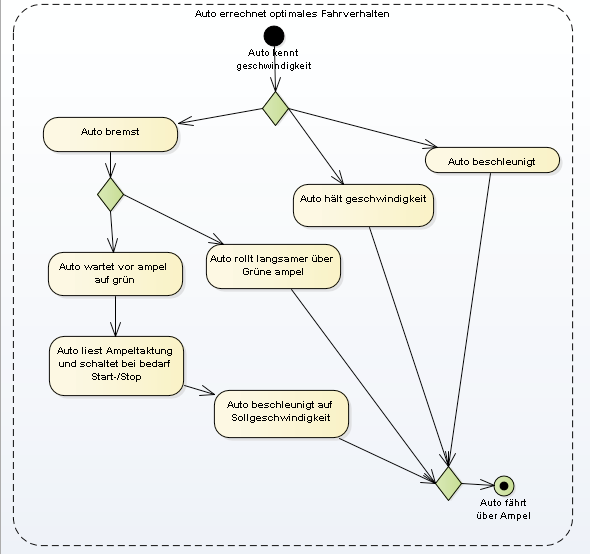
**Use Cases sollten keine Aktivitäten sein: „extern sichtbare“ Funktionen. Wichtig: Funktionen stehen im Vordergrund, nicht der konkrete Ablauf.**

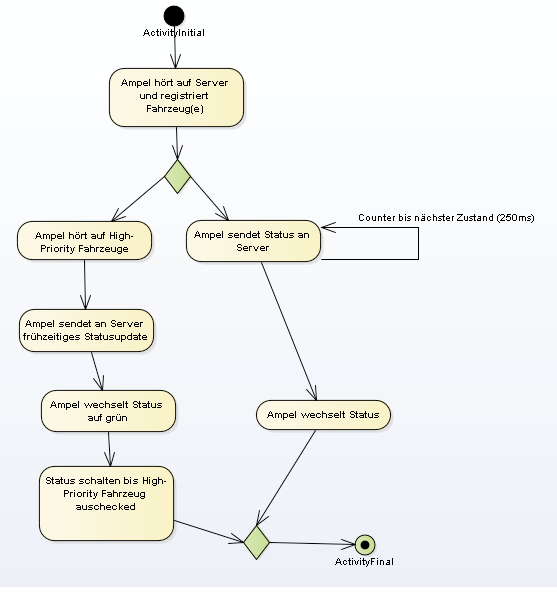
****

* 1. **Aktivitäten Diagramm**

**Jede Abbildung wird beschriftet und im Text (vorher) referenziert und beschrieben.**

****

****

****

1. **Funktionen**

Funktionsbeschreibung anhand des USE-Cases

F10 Ampel schaltet Ampelphasen

-Die Ampel ist eingeschaltet, schaltet die verschiedenen Ampelphasen und ist mit einer Kommunikationseinheit versehen

F20 Ampelphasen: Rot, Gelb-rot, Grün, Gelb

F30 logt sich ab einer bestimmen Distanz zur Ampel in das System ein

-sobald das Fahrzeug in einer relevanten Reichweite zur Ampel ist und eine Verbindung erfolgen kann, logt es sich in das System ein und kommuniziert mit der Ampel.

F40 Ampelphasen werden mitgeteilt

- Die Ampel sendet die zeitliche Abfolge der Ampelphasen an das Auto.

F50 Geschwindigkeitsoptimierung zur Effizienzsteigerung des Fahrzeugs findet statt

-Die Geschwindigkeitsoptimierung des Fahrzeugs findet durch Berechnung der Distanz zur Ampel, die dem Fahrzeug über die entsprechende Funktion mitgeteilt wird und der Zeiten der Ampelphasen statt.

F60 beschleunigen

-beschleunigt wird, wenn die Höchstzugelassene Geschwindigkeit auf der Straße noch nicht erreicht ist und es das Fahrzeug mit dieser Geschwindigkeit noch passend (aber schon knapp) über die Ampel schafft.

F70 Geschwindigkeit halten

-die Geschwindigkeit bleibt gleich, wenn das Auto ohne zu bremsen oder beschleunigen problemlos über die Ampel kommt.

F80 bremsen

-wenn gebremst werden muss, kommt es auf die Geschwindigkeitsminimierung an die vom System errechnet wurde. Im Besten Fall sollte das Auto seine Geschwindigkeit so verringern dass es ohne anzuhalten fahren kann, damit es nicht vor der roten Ampel steht.

Dies kann mit geringeren und höheren Geschwindigkeiten passieren, je nachdem wie weit das Fahrzeug von der Ampel entfernt ist.

1. **Leistungen**

L10 Ampel schaltet Ampelphasen

Bedingung: Ampel ist nicht intelligent und schaltet die ganze Zeit gleich.

L20 Ampelphasen: Rot, Gelb-rot, Grün, Gelb

Ausführungszeit: Rot 10sec

Gelb-rot 03sec

Grün 10sec

Gelb 03sec

L30 logt sich ab einer bestimmen Distanz zur Ampel in das System ein

Bedingung: Fahrzeug ist mit der Funktion ausgestattet sich in das System einzuloggen.

L40 Ampelphasen werden mitgeteilt

Bedingung: Fahrzeug kann mit Ampel kommunizieren.

Ausführungszeit: Ampel sollte die Phasen der Ampel möglichst schnell mitgeteilt bekommen, sodass das Fahrzeug früh reagieren kann und die größte Energieeffizienz besitzt.

L50 Geschwindigkeitsoptimierung zur Effizienzsteigerung des Fahrzeugs findet statt

Bedingung: Berechnung wurde richtig ausgeführt.

Ausführungszeit: Berechnung muss in Echtzeit stattfinden, damit sich das Auto direkt auf die berechnete Geschwindigkeit einstellen kann.

L60 beschleunigen

Ausführungszeit: Es sollte in Echtzeit beschleunigt werden, da die Berechnung mit einer bestimmten Distanz durchgeführt wurde und somit sofort beschleunigt werden muss.

L70 Geschwindigkeit halten

L80 bremsen

Ausführungszeit: Es sollte in Echtzeit gebremst werden, da die Berechnung mit einer bestimmten Distanz durchgeführt wurde und somit sofort gebremst werden muss.

1. **Qualitätsanforderungen**

Funktionalität: sehr gut

Zuverlässigkeit: sehr gut

Benutzbarkeit: gut

Effizienz: gut

Änderbarkeit: normal

Übertragbarkeit: normal

1. **Benutzeroberfläche**

B10 Ampel schaltet Ampelphasen

Zugriffsrechte: Programmierer kann Ampelphasen ändern

B20 Ampelphasen: Rot, Gelb-rot, Grün, Gelb

B30 Fahrzeug logt sich ab einer bestimmen Distanz zur Ampel in das System ein

Dialogstruktur: kommuniziert ab passender Reichweite mit der Ampel und ist somit im System eingeloggt.

Zugriffsrechte: Server, Programmierer, Benutzer besitzt keine Zugriffsrechte

B40 Ampelphasen werden mitgeteilt

Dialogstruktur: Ampel teilt Ampelphasen dem Fahrzeug mit.

Zugriffsrechte: Server, Ampel, Programmierer, Benutzer besitzt keine Zugriffsrechte

B50 Geschwindigkeitsoptimierung zur Effizienzsteigerung des Fahrzeugs finden

statt

Dialogstruktur: Fahrzeug kommuniziert im eigenen geschlossenen System

Zugriffsrechte: Programmierer, Benutzer besitzt keine Zugriffsrechte

B60 beschleunigen

Dialogstruktur: bekommt auszuführende Funktion vom Mikrocontroller mitgeteilt.

Zugriffsrechte: Programmierer

B70 Geschwindigkeit halten

Dialogstruktur: bekommt auszuführende Funktion vom Mikrocontroller mitgeteilt.

Zugriffsrechte: Programmierer

B80 bremsen

Dialogstruktur: bekommt auszuführende Funktion vom Mikrocontroller mitgeteilt.

Zugriffsrechte: Programmierer

1. **Sonstige Anforderungen**

Gesetze und Normen: für komplette Autonome Systeme gibt es noch keine Gesetzte und Normen

Revisionsfähigkeit: Muss vorhanden sein. Die Dokumentation soll so angefertigt werden das der aktuelle Stand des Systems direkt zu erkennen ist und anpassbar bzw. erweiterbar ist.

Sicherheitsanforderungen: Redundantes System, sollte irgendeine Komponente nicht mehr funktionieren, muss es trotzdem einen Weg geben die Sicherheit des Systems aufrecht zu erhalten.

Security, es darf kein Eindringen von außen möglich sein, niemand fremdes sollte auf das System, die Programme, Software zugreifen können und erst recht nicht ändern können.

-Passwortschutz: nur der Programmierer, die Firma, andere Zuständige Personen Firmen für das System sollten darauf zugreifen können.

Ein sicheres Passwort ist zwingend notwendig.

-Mitlaufen von Protokollen: Wichtige Daten des Systems, sollten dokumentiert werden. Es ist später gut zu Wissen mit welcher Effizienz und Sicherheit ein solches System betrieben wird.

-sichere Übertragungen: Übertragungen, Neuerungen sollten sicher und zuverlässig übertragen werden können.

Plattformunabhängigkeiten:

1. **Technisches Umfeld**

Software: MQTT, Programmiersprache

Hardware: R-FID Tags, 2 Rasperry-Pi´s

Randbedingungen:

Schnittstellen:

1. **Gliederung des Projekts**

Prototyp A mbot

Prototyp B E-Scooter

1. Implementierung des Fahrzeugs
2. Implementierung der Ampel
3. RFID-Tags ausgelesen bekommen, Distanz bestimmen
4. Kommunikationsprotokoll MQTT realisiert bekommen
5. Geschwindigkeitsanpassung des Fahrzeugs realisieren
6. Energieeffizienz, Kommunikation des Fahrzeugs realisieren
7. Prototyp der Ampel erstellen
8. Protokollieren des Projekts
9. Zwischenpräsentation, Finale Präsentation anlegen
10. **Ergänzungen**

Da das Projekt, ein autonomes Fahrzeug zu realisieren eine Auftrag von der HSHL ist, müssen wir bestimmte Anforderungen beachten. Die Bestellung der einzelnen Bauteile, Komponenten die wir benötigen müssen wir der Hochschule vorschlagen die sie dann bestellt oder diese Bestellung ablehnt.

Bei der Auswahl dieser Bauteile spielt die Funktionalität in erster Linie eine Rolle.

Das Projekt ist ein Semesterprojekt des 5ten Semesters und wird am Ende des Semesters Teil der Bewertung des Fachs Cyber-Physical Systems sein.

1. **Testfälle**

1.Test RFID Daten werden eingelesen, womit die Distanz des Fahrzeugs feststeht.

Dies kann mit einem Rechner der an den Rasberry-Pi angeschlossen ist kontrolliert werden.

2.Test Ampel überträgt mittels MQTT die Ampelphasen an den Rasberry-Pi.

Dies kann mit einem Rechner der an den Rasberry-Pi angeschlossen ist ausgelesen und somit kontrolliert werden.

3.Test Umsetzung des Projekts am mbot. Es wird geprüft ob die Berechnung richtig funktioniert, und der mbot je nach Distanz, das richtige tut (bremsen, beschleunigen, Geschwindigkeit halten).

Glossar: und / Systemmodell??

z.B. Streckenabschnittskontrolle???