

Zeitraum des Projekts	03.07.2024 – 12.09.2024
Vertiefungsaufgabe	Ampelsteuerung
Gruppen-Teilnehmer	Markus Seitz, Jeff Kurzbach, Luca Vierlinger, Fabian Schäfer

Dokumentation

im Fach „Künstliche Intelligenz“ für das Zusatzprojekt einer „Ampelsteuerung“ im Rahmen des Projektumfangs des OpenBot an der Technikerschule in Neumarkt i.d.OPf.

Ampelsteuerung mit Microcontroller



Abbildung 1: Bauteile Ampelsteuerung

Vorgelegt von: Schäfer Fabian, Kurzbach Jeff, Vierlinger Luca, Seitz Markus

Erstellt am: 12.09.2024

Projektbetreuer: Weber Dominik, Meier Christian

Inhaltsverzeichnis

1. Projektbeschreibung OpenBot	3
2. Projektbeschreibung Ampelsteuerung.....	4
3. Anforderungen an die Ampelsteuerung.....	5
4. Zeitmanagement und Aufgabenteilung.....	6
5. Entwicklungswerkzeuge	7
5.1 Hardware	7
5.2 Software	7
6. Projektdurchführung.....	8
6.1 Konstruktion des Ampelgehäuses	8
6.2 Konstruktion des Steuerkastens	9
6.3 Erstellen des Schaltplans	9
6.4 Verdrahtung der Ampelsteuerung	10
6.5 Programmierung des Arduino Mega 2560	11
6.6 Programmierung der Android App	13
7. Nutzen der Smartphone-App	15
8. Ergebnisse	16
8.1 Diskussion und Ausblick	16
8.2 Erweiterungsmöglichkeiten.....	16
9. Fazit	17
10. Literaturverzeichnis.....	18
11. Abbildungsverzeichnis	19
12. Anhang	20

1. Projektbeschreibung OpenBot

„OpenBot“ ist ein Forschungsprojekt von Intel, das kostengünstige Roboterlösungen für den Bildungs- und Forschungsbereich entwickelt. Es verwandelt herkömmliche Android-Smartphones in Roboter, indem es diese auf einem einfachen, fahrbaren Untersatz (ähnlich einem Mini-RC-Auto) montiert. Die Rechenleistung und die Sensoren des Smartphones, wie Kamera und Gyroskop, werden genutzt, um die Umgebung wahrzunehmen und autonome Fahrfunktionen zu ermöglichen. Ziel ist es, Robotik und künstliche Intelligenz mit minimalen Kosten zugänglich zu machen und offene Software- sowie Hardwareplattformen bereitzustellen.

Im Rahmen einer Projektarbeit an der Technikerschule in Neumarkt i.d.OPf. im Teilbereich Elektrotechnik wurde, nach dem Vorbild des „OpenBot“ von Intel, ein autonom fahrender Roboter entwickelt, dessen neuronales Netz darauf trainiert wurde einen vorgegebenen Rundkurs automatisch abzufahren.

Im Anschluss an dieses Projekt wurde unserem Team die Zusatzaufgabe übergeben eine Ampel mit zugehöriger Steuerungselektronik für die Teststrecke zu entwickeln.

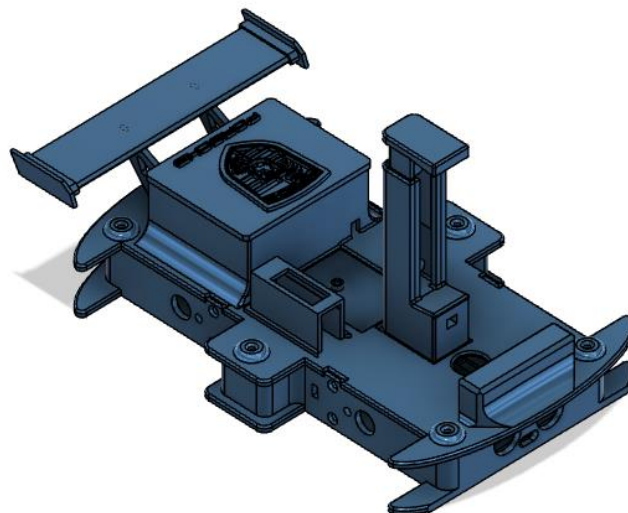


Abbildung 2: Chassis OpenBot

2. Projektbeschreibung Ampelsteuerung

Im Rahmen des Projekts „OpenBot“ wurde unserem Team die Zusatzaufgabe übertragen, eine Ampelsteuerung für die Teststrecke des „OpenBot“ zu entwerfen. Die Ampeln, sollen zukünftig als weiterer Projektteil des „OpenBot“ fungieren und über dessen neuronales Netz erkannt und ausgewertet werden. So soll der „OpenBot“ die Ampeln als Referenzobjekte verwenden, welche über sein neuronales Netz erkannt werden, sodass dieser dann bei einer rot leuchtenden Ampel stoppt bzw. bei einer grün leuchtenden Ampel weiterfährt. Folgende Dokumentation gibt Aufschluss über den Projekt- ablauf und ist zudem als detaillierte Betriebsanleitung der Ampelsteuerung zu sehen.

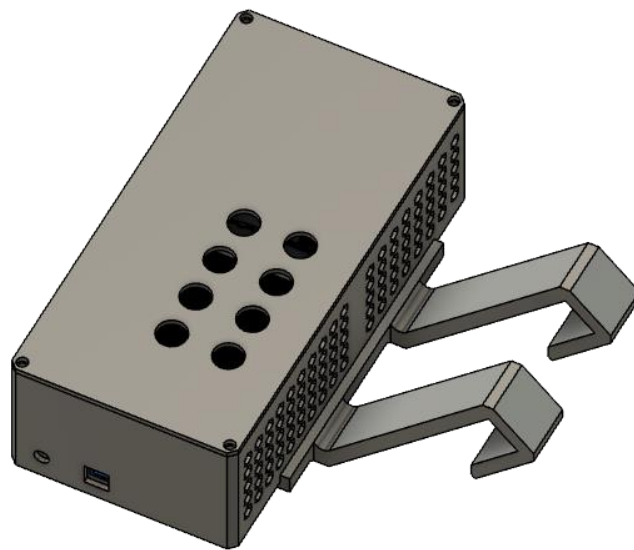


Abbildung 3: Steuerungskasten leer

3. Anforderungen an die Ampelsteuerung

Zu Beginn des Projekts wurden die wichtigsten Anforderungen / Eigenschaften der Ampelsteuerung festgelegt. Dieser erste Planungsschritt ist essenziell und liefert den Grundbaustein für einen guten und reibungslosen Projektablauf. Die Anforderungen, sowie deren Umsetzung wurden in einer Tabelle festgehalten, welche im Folgenden aufgeführt ist.

Anforderung	Umsetzung
Das Lichtfeld der Ampeln soll einen Durchmesser von mindestens 20mm aufweisen.	Verwendung einer Kreisrunden Anordnung an LED's, die die geforderte Größe aufweist.
Das Lichtfeld der Ampeln soll in verschiedenen Farben leuchten können.	Verwendung von WS2812B-RGB-LED's.
Die Ampeln sollen über einen feste Spannungsversorgung verfügen und einfach austauschbar / versetzbar sein.	Die Ampeln werden über verpolungssichere Steckverbinder über ein Kabel mit Spannung versorgt.
Die Ampeln sollen manuell über Taster zwischen den Farben Rot und Grün geschaltet werden können.	In den Steuerungskasten für die Ampelsteuerung werden Drucktaster eingebaut, über die die Ampeln gesteuert werden können.
Die Ampeln sollen aus der Ferne zwischen den Farben Rot und Grün geschaltet werden können.	Es wird ein Bluetoothmodul vom Typ HC-05 im Steuerungskasten verbaut, über das Daten vom Smartphone auf den Microcontroller übertragen werden können. Zusätzlich wird eine App entwickelt, mit Hilfe welcher das Bluetoothmodul und das Smartphone gekoppelt werden und so die Ampeln über das Smartphone gesteuert werden können.
Die Ampelsteuerung soll leicht skalierbar und erweiterbar sein.	Für die Steuerung der Ampeln wird ein Arduino Mega 2560 verwendet.
Es soll möglich sein mit den Ampeln eine Kreuzung mit vier Zufahrten zu steuern.	Es werden vier Prototypen des Ampelelements gebaut.

4. Zeitmanagement und Aufgabenteilung

Um den zeitlichen Ablauf des Projekts, sowie die Aufgabenteilung planen zu können wurde ein Ablaufplan für das Zusatzprojekt erstellt. In diesem sind Meilensteine / Teilziele des Projekts, ebenso wie der zeitliche Rahmen, in welchem die Teilziele zu erreichen sind, festgehalten. Ein weiteres Feld in der Tabelle zeigt, welche Aufgaben von welchen Teammitgliedern übernommen wurden. Auch der aktuelle stand der Arbeiten ist in einem weiteren Tabellenfeld dargestellt. Zum Zeitpunkt der Niederschrift dieser Dokumentation sind bereits alle anderen Aufgaben abgeschlossen, weshalb die Tabellenfelder hier mit „100%“ gekennzeichnet sind. Der Zeitablaufplan liegt dem Anhang dieser Dokumentation bei.

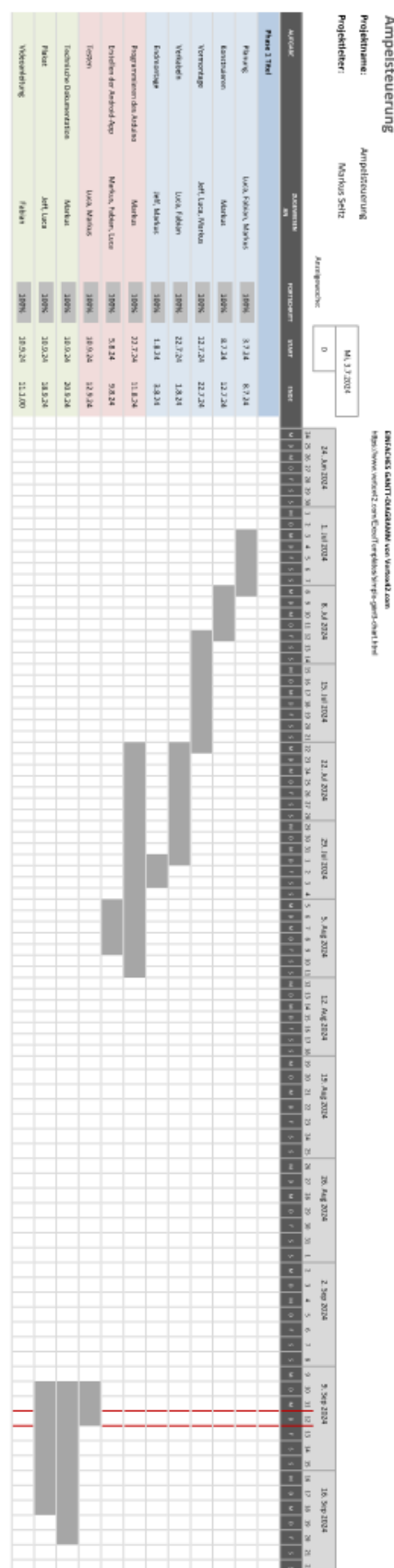


Abbildung 4: Zeitablaufplan Ampelsteuerung

5. Entwicklungswerkzeuge

Als Entwicklungswerkzeuge werden Software oder Tools, die in einer technischen Arbeit genutzt werden, um die Entwicklung, Analyse oder Optimierung technischer Systeme zu unterstützen, bezeichnet. Dazu gehören z.B. Editoren, Compiler, Debugger, aber auch Maschinen, wie 3D-Drucker oder Fräsen. Sie erleichtern den Entwicklungsprozess, automatisieren Aufgaben und helfen bei der Umsetzung technischer Konzepte in funktionierende Lösungen.

Im Folgenden sind alle Entwicklungswerkzeuge, die zur Umsetzung der Ampelsteuerung verwendet wurden aufgeführt.

5.1 Hardware

Werkzeuge	Maschinen
<ul style="list-style-type: none"> ▪ LötKolben ▪ div. Schraubendreher 	<ul style="list-style-type: none"> ▪ 3D-Drucker ▪ PC ▪ Notebook ▪ Smartphone

Bauteile	Verbrauchsmaterial
<ul style="list-style-type: none"> ▪ Arduino Mega 2560 ▪ HC-05 Bluetooth-Shield ▪ 4pol.-Steckverbinder ▪ WS2812B-RGB LED-Ring ▪ Drucktaster 1 x NO ▪ Klemmenblock Mega 2560 ▪ Netzteil 5V/3A ▪ DC-Buchse 5,5mm x 2,1mm ▪ Sicherungshalter 5x20mm ▪ Wago Klemmen 2002 	<ul style="list-style-type: none"> ▪ Einzeladerleitung 1mm² blau ▪ Einzeladerleitung 1mm² blauweiß ▪ Filament PLA-Basic ▪ Aderendhülsen 1mm² ▪ Kunststoffkleber ▪ div. Schrauben ▪ Lötzinn ▪ div. Schrumpfschläuche

5.2 Software

Programmierung	Konstruktion / Zeichnen	3D-Druck
<ul style="list-style-type: none"> ▪ Arduino IDE ▪ MIT AppInventor 	<ul style="list-style-type: none"> ▪ Autodesk Fusion 360 ▪ ProfiCAD 	<ul style="list-style-type: none"> ▪ BambuStudio

6. Projektdurchführung

Nachdem die wichtigsten Anforderungen an die Ampelsteuerung, sowie die zur Umsetzung der Anforderungen nötigen Maßnahmen festgelegt wurden, wurde im Folgenden ein Prototyp der Ampelsteuerung entwickelt.

6.1 Konstruktion des Ampelgehäuses

Für die Konstruktion des Ampelgehäuses wurde das CAD-Zeichenprogramm „Fusion 360“ von Autodesk verwendet.¹

Das Ampelgehäuse selbst setzt sich aus vier Einzelteilen zusammen. Bei der Konstruktion des Gehäuses wurde darauf geachtet, dass die WS2812B-RGB-LED, sowie der 4-polige Steckverbinder möglichst einfach und ohne großen Aufwand in das Gehäuse eingebaut werden können.

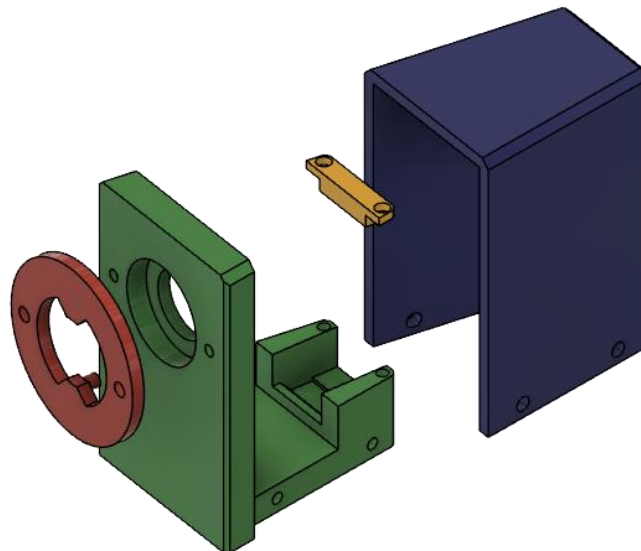


Abbildung 5: Explosionsdarstellung Ampelgehäuse

¹ vgl. Autodesk Fusion kostenlos herunterladen | Kostenlose Testversion | Autodesk (o. D.): [online] <https://www.autodesk.com/de/products/fusion-360/free-trial?msoclid=10e30f849123653629981b16904864aa> [abgerufen am 12.09.2024].

6.2 Konstruktion des Steuerkastens

Der Steuerkasten für die Ampelsteuerung wurde ebenfalls mit der Software „Fusion 360“ erstellt.

Hier galt es den Arduino Mega 2560, die Drucktasten, die Klemmen sowie den Sicherungshalter und vier der 4-poligen Steckverbinder in das Gehäuse zu integrieren. Um ein Aufstauen von warmer Luft im Gehäuse während des Betriebs zu vermeiden wurde dieses zusätzlich mit Lüftungsschlitzen versehen.

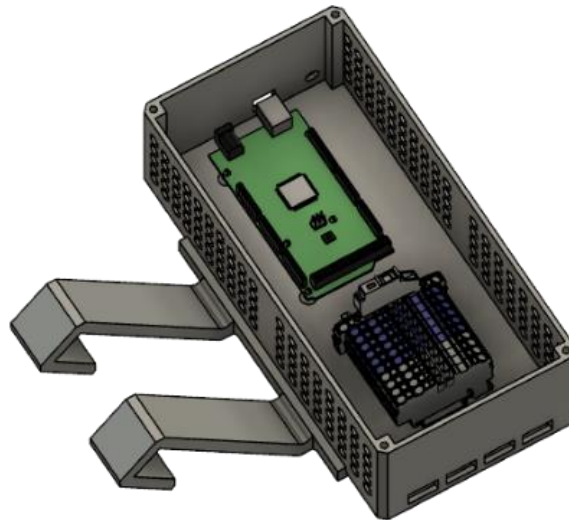


Abbildung 6: Steuerungskasten ohne Deckel

Um spätere Änderungen am Programm der Anlage einfach auf den Arduino übertragen zu können, wurde dieser so im Gehäuse platziert, dass dessen USB-Buchse ohne öffnen des Gehäusedeckels erreicht werden kann.

6.3 Erstellen des Schaltplans

Bevor der Hardwareaufbau der Ampelsteuerung verdrahtet werden konnte musste ein Schaltplan erstellt werden. Hierfür wurde das kostenlose Softwareprogramm „ProfiCAD“ verwendet.²



Abbildung 7: ProfiCAD Logo

Der Schaltplan kann dem Anhang dieser Dokumentation entnommen werden.

² vgl. Electrical CAD Software - ProfiCAD (o. D.): [online] <https://www.profi-cad.com/#:~:text=The%20best%20CAD%20for%20electrical%20and> [abgerufen am 12.09.2024].

6.4 Verdrahtung der Ampelsteuerung

Die Verdrahtung der Ampelsteuerung erfolgte unter Einhaltung aller geltenden Normen und Richtlinien der Elektrotechnik!

Nachdem der Schaltplan für vorliegendes Projekt erstellt war, konnte mit der Verdrahtung der Hardware begonnen werden.

Achtung:

Der Arduino Mega 2560 verfügt über interne „Pullup“-Widerstände, die hier für die Eingänge, auf die die Schaltdrähte der Drucktaster geführt wurden, aktiviert sind.³

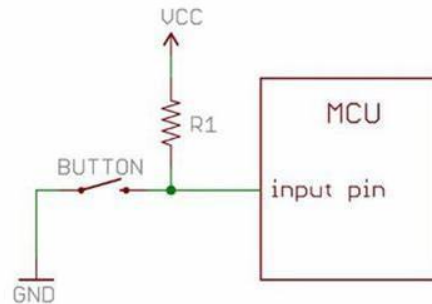


Abbildung 8: Pullup Resistor

Das heißt, um einen Wechsel des Signalpegels an diesen Pin's des Arduino zu erzeugen muss das 0V-Potential auf genannte Pin's gelegt werden. Die Drucktaster schalten also das 0V-Potential der Anlage auf die Eingänge des Arduino und **nicht** das 5V-Signal!



Abbildung 9: Steuerungskasten verdrahtet

³ vgl. Digital Input Pull-Up resistor (o. D.): www.arduino.cc, [online] <https://www.arduino.cc/en/Tutorial/DigitalInputPullup> [abgerufen am 12.09.2024].

6.5 Programmierung des Arduino Mega 2560

Zur Programmierung des Mega 2560 wurde die von Arduino eigens dafür bereitgestellte Software „Arduino IDE“ verwendet.⁴ Die Logik des Microcontrollers kann einfach in der Programmiersprache C entwickelt werden, wobei die „Arduino IDE“ den großen Vorteil bietet, dass zur Ansteuerung der Pin's des Controllers bereits eingebettete Funktionen vorliegen, was das Programmieren erheblich vereinfacht.

Für die Ansteuerung der WS2812B-RGB-LED's wird die Bibliothek „FastLED“ von „Daniel Garcia“ mit der Version 3.7.1 verwendet. Diese kann in der IDE einfach unter dem Reiter „Bibliotheken“ installiert werden.⁵

Um die Helligkeit der LED's anzupassen kann in Zeile 41 des Programmcodes der Wert vom Typ „Integer“ zwischen 0 und 255 variabel eingestellt werden. 255 entspricht dabei der Leuchtstärke 100%.

```
31 // define variables for the WS2812 RGB LED
32
33 #define RGB_PIN_1    5           // data pin for traffic light 1
34 #define RGB_PIN_2    6           // data pin for traffic light 2
35 #define RGB_PIN_3    7           // data pin for traffic light 3
36 #define RGB_PIN_4    8           // data pin for traffic light 4
37 #define RGB_LED_NUM_1 7           // 8 LEDs [0...7]
38 #define RGB_LED_NUM_2 7           // 8 LEDs [0...7]
39 #define RGB_LED_NUM_3 7           // 8 LEDs [0...7]
40 #define RGB_LED_NUM_4 7           // 8 LEDs [0...7]
41 #define BRIGHTNESS    50         // brightness range [0..255]
42 #define CHIP_SET      WS2812B    // types of RGB LEDs
43 #define COLOR_CODE    GRB        // sequence of colors in data stream
```

Abbildung 10: Programmcode Helligkeit

⁴ vgl. Software (o. D.): Arduino, [online] <https://www.arduino.cc/en/software> [abgerufen am 12.09.2024].

⁵ vgl. FastLED - Arduino reference (o. D.): www.arduino.cc, [online] <https://www.arduino.cc/reference/en/libraries/fastled/> [abgerufen am 12.09.2024].

Das Programm ist so geschrieben, dass die LED's, je nachdem welche der Drucktas-
ten betätigt wird, die Farben Rot oder Grün annehmen. Um die Farben der LED's, die
diese bei einem entsprechenden Tastendruck annehmen zu ändern, müssen die
Übergabewerte der Funktion „CRGB()“ (hier in Zeile 63) verändert werden.

Dies ist für jede LED einzeln vorzunehmen!

```
59 void switch_A1_green(){  
60  
61     for (int i = 0; i < RGB_LED_NUM_1; i++) {  
62  
63         LED_ARR_1[i] = CRGB(0, 255, 0);    // set to green  
64         FastLED.show();  
65  
66     }  
67 }
```

Abbildung 11: Programmcode Farbe

Die Funktion „CRGB()“ erwartet drei Werte vom Typ „Integer“ als Übergabeparameter. Der erste Wert gibt die Stärke der Lichtfarbe Rot, der zweite der Lichtfarbe Grün und der dritte der Lichtfarbe Blau an.

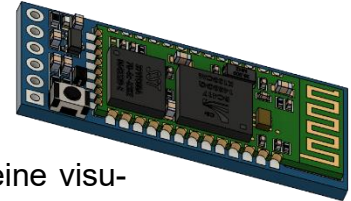
Die Werte können variabel zwischen 0 und 255 eingestellt werden.

Wollte man also eine rein blau leuchtende LED, so würde die Funktion wie folgt aussehen: **CRGB(0, 0, 255);**

6.6 Programmierung der Android App

Um die Ampeln auch aus der Ferne steuern zu können wurde im Vorfeld unter 3 der Lösungsansatz einer Smartphone-App in Verbindung mit dem Bluetoothmodul HC-05 festgelegt.⁶ Die Smartphone-App soll über Schaltflächen verfügen, mit denen die LED's der Ampeln gesteuert werden können. Bei Betätigung einer Schaltfläche in der App soll ein Signal an das Bluetoothmodul gesendet werden, welches dieses dann über die serielle Schnittstelle an den Arduino weiterleitet.

Abbildung 12: HC-05 Bluetoothmodul



Zum erstellen der Smartphone-App wurde in diesem Projekt der „MIT-AppInventor“ verwendet.⁷ Der „MIT-AppInventor“ ist eine visuelle Entwicklungsumgebung, die es ermöglicht, ohne Programmierkenntnisse mobile Apps für Android zu erstellen. Mit einer einfachen „Drag-and-Drop“-Oberfläche können Benutzer visuelle Bausteine nutzen, um Funktionen und Logik für ihre Apps zu erstellen.

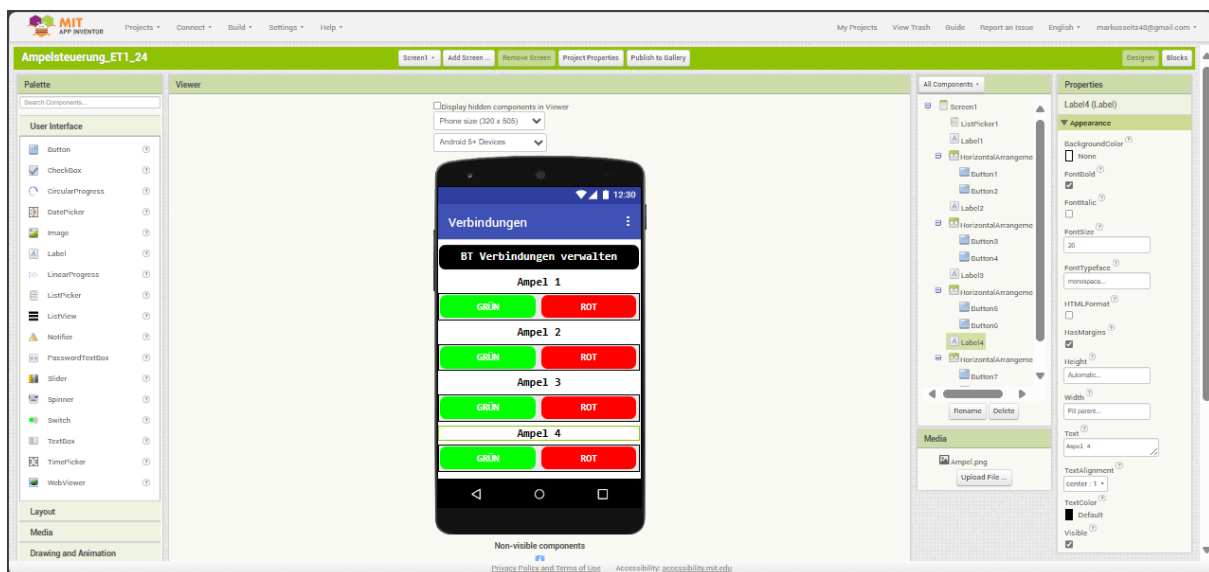


Abbildung 13: MIT-AppInventor Appview

⁶ vgl. Ewald, Wolfgang (2023): HC-05 und HC-06 Bluetooth Module, Wolles Elektronikliste, [online] <https://wolles-elektronikkiste.de/hc-05-und-hc-06-bluetooth-module#:~:text=Detail-> lierte%20und%20schrittweise%20Anleitung%20f%C3%BCr%20HC-05 [abgerufen am 12.09.2024].

⁷ vgl. MIT App Inventor (o. D.): appinventor.mit.edu, [online] <https://appinventor.mit.edu/#:~:text=MIT%20App%20Inventor.%20YouTube%20Channel.%20News> [abgerufen am 12.09.2024].

Nachdem die App im „MIT-AppInventor“ erstellt wurde kann eine „.apk“-Datei erzeugt werden, mit Hilfe welcher die App dann auf dem Smartphone installiert werden kann. Die „.apk“-Datei ist im Anhang dieser Dokumentation enthalten.

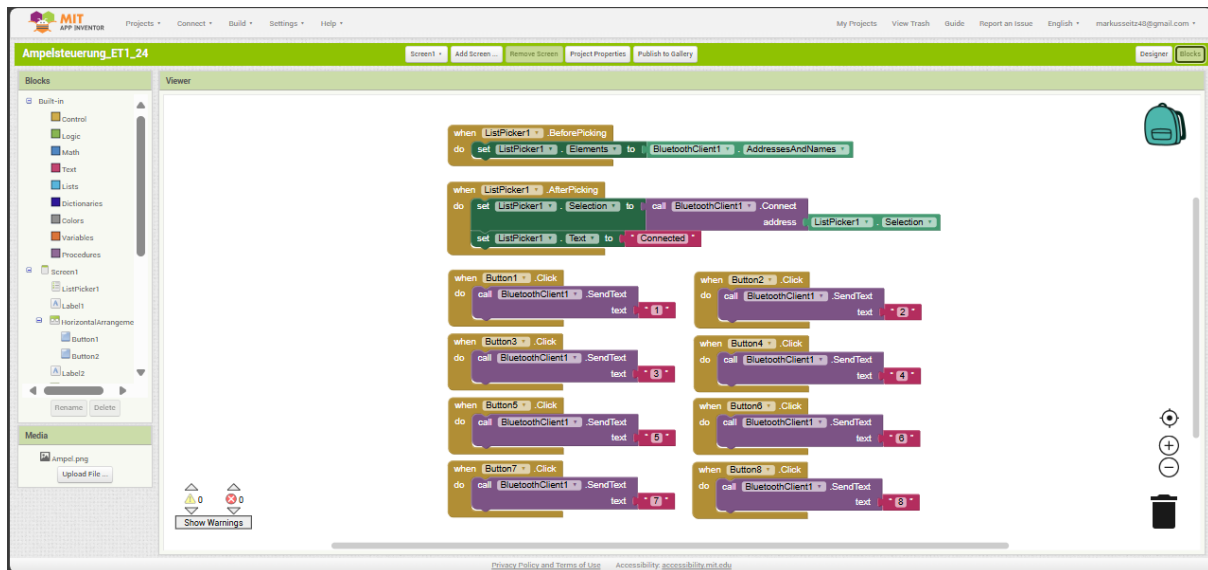


Abbildung 14: MIT-AppInventor Programmview

Wird eine der Schaltflächen in der App betätigt, so wird eine entsprechende Variable vom Typ „String“ über die serielle Schnittstelle an den Arduino geschickt.

Nehmen wir als Beispiel den Fall, für das Betätigen der Schaltfläche von „Ampel-1-grün“. Bei Betätigung der Schaltfläche wird eine Variable vom Typ „String“ mit der payload „1“ über die serielle Schnittstelle an den Arduino geschickt. Dieser empfängt die Variable und wertet diese aus.

Achtung:

Der String mit dem Variablenwert „1“ entspricht laut ASCII-Tabelle der Variable 49 vom Typ „Integer“. Dies ist bei der Auswertung im Programm, wie in folgender Abbildung in Zeile 349 gezeigt, zu beachten.

```
347 ✓ switch (incoming_value_int) {
348
349     case 49:                // 49 is ASCII Code for "1"
350         switch_A1_green();
351         Serial.println("switched A1 to green!");
352         break;
```

Abbildung 15: Programmcode String-Auswertung

7. Nutzen der Smartphone-App

Um die Ampeln über die Smartphone-App steuern zu können muss diese zuerst mit Hilfe der dem Anhang beiliegenden „.apk“-Datei auf dem entsprechenden Smartphone installiert werden. Hierzu die „.apk“-Datei in einen Dateiordner auf dem Smartphone kopieren. Durch klicken auf die Datei wird die Installation gestartet.

Hinweis:

Es kann sein, dass die Option „Installation aus unbekannten Quellen“ auf dem Smartphone deaktiviert ist. Um die Installation durchführen zu können muss diese aktiviert sein.

Um sich in der App mit dem Bluetoothmodul verbinden zu können, muss dieses zuerst mit dem Smartphone gekoppelt werden. Hierzu einfach Bluetooth auf dem Smartphone aktivieren und auf das HC-05 Modul klicken.

➔ (PIN: 1234)

Nach erfolgreicher Kopplung kann die App geöffnet werden. Um sich nun in der App mit dem Bluetoothmodul zu verbinden drückt man einfach auf die Schaltfläche „BT Verbindungen verwalten“ und klickt dann anschließend auf das HC-05-Modul. Die App verbindet sich nun mit dem Modul. Ist die Verbindung erfolgreich, schließt sich das Fenster und auf der Schaltfläche wird nun der Text „connected“ angezeigt. Nun kann durch drücken der Schaltflächen jede Ampel entsprechend gesteuert werden.

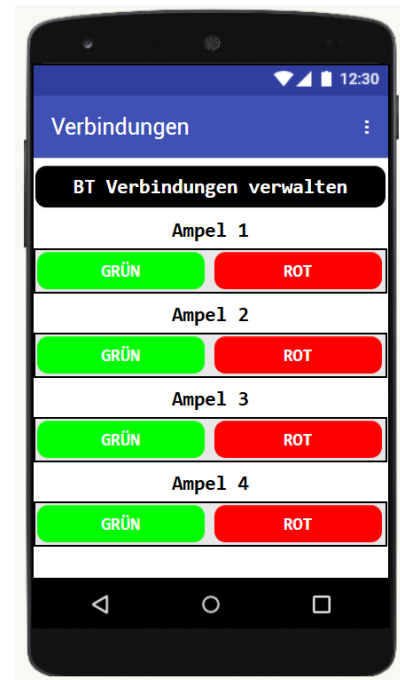


Abbildung 16: Mobile-App

8. Ergebnisse

8.1 Diskussion und Ausblick

Im Rahmen des Projekts „Ampelsteuerung“ konnten alle festgelegten Anforderungen erfolgreich umgesetzt werden. Die Ampelprototypen wurden konstruiert, verdrahtet, programmiert und auf ihre Funktionsfähigkeit hin getestet. Über Drucktaster und eine Smartphone-App lassen sich die Ampeln zwischen den Farben Rot und Grün umschalten. Die Bluetooth-Kommunikation zwischen der App und dem Arduino Mega 2560 funktioniert fehlerfrei, was eine Fernsteuerung ermöglicht. Sie ist allerdings in ihrer Reichweite begrenzt.

Beim Praxistest der Ampelsteuerung fiel auf, dass das Datensignal zum Umschalten der Farben, das vom Arduino an die WS2812B-LED's gesendet wird, von diesen nicht immer sicher erkannt wird. So kam es beim Erproben der Ampeln vor, dass diese bei Tastendruck nicht die gewünschte Farbe annahmen oder gar nicht reagierten.

Dies ist der Leitungslänge der Verbindung zwischen Steuerungskasten und Ampeln geschuldet. Bei der Verwendung kürzerer Leiter trat dieses Problem nicht auf.

Zur Behebung dieses Problems kann beispielsweise eine weitere WS2812B-LED in die Mitte jeder Leiterstrecke zwischen Steuerkasten und Ampel geschaltet werden. Diese dient somit als Signalverstärker. Ebenso kann ein SK6812-LED-Signalverstärker verwendet werden, der einfach hinter die Ausgänge des Arduino geschaltet wird.

Beide Lösungsansätze wurden im Rahmen dieser Arbeit aufgrund des begrenzten Zeitumfangs nicht mehr behandelt und können in späteren Projekten als Zusatzaufgaben für andere Gruppen dienen.

8.2 Erweiterungsmöglichkeiten

Die Ampelsteuerung für das „OpenBot“-Projekt bietet zahlreiche Ansätze für zukünftige Erweiterungen. Eine mögliche Weiterentwicklung wäre die Integration weiterer Verkehrszeichen wie etwa Fußgängerampeln, oder einer Brücke, die sich heben und senken lässt. Auch die Nutzung von Echtzeitdaten, etwa durch die Einbindung von Sensoren für Verkehrsfluss könnte die Steuerung dynamischer gestalten. Darüber hinaus könnte die Kommunikation zwischen mehreren Ampeln und dem „OpenBot“ durch die Implementierung eines Netzwerks verbessert werden, um synchronisierte Steuerungen an Kreuzungen zu ermöglichen.

9. Fazit

Das Zusatzprojekt „Ampelsteuerung“ im Rahmen des „OpenBot“-Projekts war ein erfolgreicher Schritt zur Erweiterung der „OpenBot“-Testumgebung an der Technikerschule in Neumarkt i.d.OPf.. Durch die Implementierung einer modularen, einfach skalierbaren Ampelsteuerung konnten wir eine zuverlässige Basis für die Erkennung und Auswertung von Ampelsignalen durch den „OpenBot“ schaffen. Die Kombination aus manuellem und ferngesteuertem Betrieb bietet Flexibilität im Testbetrieb. Zukünftige Erweiterungen, wie die Integration komplexerer Verkehrssteuerungen, sind durch die skalierbare Architektur problemlos realisierbar.

Das Projekt bedarf allerdings noch weiterer Optimierungsmaßnahmen. Wie unter 8.1 beschrieben erkennen die WS2812B-LED's das Datensignal des Arduino nicht immer sicher, was zu Fehlfunktionen im Betrieb der Ampeln führt. Hier wurden unter 8.1 bereits mögliche Lösungsansätze genannt.

10. Literaturverzeichnis

-
 appinventor.mit.edu. (kein Datum). Abgerufen am 12. 09 2024 von MIT App Inventor:
<https://appinventor.mit.edu/#:~:text=MIT%20App%20Inventor.%20YouTube%20Channel.%20News>

- Ewald, W. (. (3. 12 2023). Wolles Elektronikliste. Abgerufen am 12. 09 2024 von HC-05 und HC-06 Bluetooth Module: <https://wolles-elektronikliste.de/hc-05-und-hc-06-bluetooth-module#:~:text=Detaillierte%20und%20schrittweise%20Anleitung%20f%C3%BCr%20HC-05>

- www.arduino.cc. (kein Datum). Abgerufen am 12. 09 2024 von Digital Input Pull-Up resistor: <https://www.arduino.cc/en/Tutorial/DigitalInputPullup>

- www.arduino.cc. (kein Datum). Abgerufen am 12. 09 2024 von Software: <https://www.arduino.cc/en/software>

- www.arduino.cc. (kein Datum). Abgerufen am 12. 09 2024 von FastLED - Arduino reference: <https://www.arduino.cc/reference/en/libraries/fastled/>

- www.autodesk.com. (kein Datum). Abgerufen am 12. 09 2024 von Autodesk Fusion kostenlos herunterladen | Kostenlose Testversion | Autodesk: <https://www.autodesk.com/de/products/fusion-360/free-trial?msocid=10e30f849123653629981b16904864aa>

- www.proficad.com. (kein Datum). Abgerufen am 12. 09 2024 von Electrical CAD Software - ProfiCAD: <https://www.proficad.com/#:~:text=The%20best%20CAD%20for%20electrical%20and>

11. Abbildungsverzeichnis

Abbildung 1: Bauteile Ampelsteuerung	1
[Eigene Aufnahme, Markus Seitz, 12.09.2024]	
Abbildung 2: Chassis OpenBot.....	3
[Eigene Aufnahme, Markus Seitz, 12.09.2024]	
Abbildung 3: Steuerungskasten leer	4
[Eigene Aufnahme, Markus Seitz, 12.09.2024]	
Abbildung 4: Zeitablaufplan Ampelsteuerung	6
[Eigene Aufnahme, Markus Seitz, 12.09.2024]	
Abbildung 5: Explosionsdarstellung Ampelgehäuse	8
[Eigene Aufnahme, Markus Seitz, 12.09.2024]	
Abbildung 6: Steuerungskasten ohne Deckel	9
[Eigene Aufnahme, Markus Seitz, 12.09.2024]	
Abbildung 7: ProfiCAD Logo.....	9
Link To Us - ProfiCAD	
Abbildung 8: Pullup Resistor.....	10
Pull-up Resistors Explained: Arduino Guide to Stable Signal Inputs (duino4projects.com)	
Abbildung 9: Steuerungskasten verdrahtet.....	10
[Eigene Aufnahme, Markus Seitz, 12.09.2024]	
Abbildung 10: Programmcode Helligkeit.....	11
[Eigene Aufnahme, Markus Seitz, 12.09.2024]	
Abbildung 11: Programmcode Farbe	12
[Eigene Aufnahme, Markus Seitz, 12.09.2024]	
Abbildung 12: HC-05 Bluetoothmodul	13
[Eigene Aufnahme, Markus Seitz, 12.09.2024]	
Abbildung 13: MIT-AppInventor Appview	13
[Eigene Aufnahme, Markus Seitz, 12.09.2024]	
Abbildung 14: MIT-AppInventor Programmview	14
[Eigene Aufnahme, Markus Seitz, 12.09.2024]	
Abbildung 15: Programmcode String-Auswertung.....	14
[Eigene Aufnahme, Markus Seitz, 12.09.2024]	
Abbildung 16: Mobile-App.....	15
[Eigene Aufnahme, Markus Seitz, 12.09.2024]	

12. Anhang

- Schaltplan der Ampelsteuerung
- CAD-Modell Ampelgehäuse (.step, .f3d)
- CAD-Modell Steuerungskasten (.step, .f3d)
- CAD-Modell Haltebügel (.step, .f3d)
- Android-App (.apk)
- Arduino-Programmcode (.ino)
- Zeitablaufplan
- Materialliste / Kostenaufstellung
- Plakat
- Videoanleitung

Der Anhang wird in gedruckter Form, sowie in einem GitHub-Repository bereitgestellt!

user: Markus-Seitz-OpenBot24

repo: Markus-Seitz-OpenBot24/Ampelsteuerung_ET2

https: https://github.com/Markus-Seitz-OpenBot24/Ampelsteuerung_ET2.git