

Projektarbeit

Programmierung einer Universellen Infrarot-Fernbedienung mit einem Arduino UNO

Vorgelegt am: Datum

Von: Josia Rudolph, Nils Klotz, Martin Görner

Studiengang: Technische Informatik

Studienrichtung: Technik

Matrikelnummer: s4004602, s4004395, s4004485

Praxispartner: BA-Glauchau

Gutachter: Prof. A. Munke

Inhaltsverzeichnis

Abbildungsverzeichnis	III
Abkürzungsverzeichnis	IV
1. Zielsetzung	1
2. Beschreibung der Module	2
2.1. Infrarot Sender	2
2.2. Infrarot Empfänger	2
2.3. LCD-Display	3
2.4. Drehschalter (Rotary Encoder)	4
2.5. Arduino UNO	4
3. Funktionsweise der Module	6
3.1. Infrarot Sender	6
3.2. Infrarot Empfänger	6
3.3. LCD-Display	7
3.4. Drehschalter (Rotary Encoder)	7
4. Die Universelle Infrarot-Fernbedienung	9
5. Der Aufbau	12
6. Installationshinweise	15
7. Test und Fazit	16
8. Quellen	17

Abbildungsverzeichnis

Abbildung 1 , LED, funktionsweise (Pinlight, 2024)	2
Abbildung 2 , Fotodiode, funktionsweise (Digi Key, 2022)	3
Abbildung 3 , Display LCM1602A (Elegoo, 2024)	3
Abbildung 4 , Rotary encoder, Keyes KY-040, Funktionsweise (Elegoo, 2024)	4
Abbildung 5 , Arduino UNO (Elegoo, 2024)	5
Abbildung 6 , Klassendiagramm der Zielerwartung	10
Abbildung 7 , Ablaufdiagramm des Hauptsystems	11
Abbildung 8 , Technischer Aufbau als frizing Sketch	12
Abbildung 9 , Technischer Aufbau als Blockschaltbild	12
Abbildung 10 , Tatsächlicher Aufbau auf Steckbrett, Hauptmenü	14
Abbildung 11 , Tatsächlicher Aufbau auf Steckbrett, Empfangsmenü	14

Abkürzungsverzeichnis

USB	Universeller Serieller Bus
VCC	Common Collector Voltage (Versorgungsspannung)
GND	Ground (Masse)
LCD	Liquid Crystal Display
DVD	Digital Video Disc
NEC	Infrarot Protokoll von „Now Renesas“
SW	Switch (Schalter)
PWM	Pulse Width Modulation (Pulsweitenmodulation)
MISO	Master-In-Slave-Out (SPI-Bus Bezeichner)
MOSI	Master-Out-Slave-In (SPI-Bus Bezeichner)
SPI	Serial Peripheral Interface
IDE	Integrierte Entwicklungsumgebung

1. Zielsetzung

Von einfachen Steckdosen, welche den Stromverbrauch messen und Deckenlampen, die mit dem Smartphone bedienbar sind, bis hin zu Kühlschränken, die den aktuellen Inhalt per Foto mit dem Smartphone kommunizieren können und Warnmeldungen ausgeben, wenn keine Butter mehr da ist. Das Thema rund um die Hausautomatisierung nimmt immer größere Umfänge an. Es werden immer neuere Techniken entwickelt und moderne Protokolle genutzt, um diese riesige Bandbreite an Informationen schnell genug und bestenfalls gebündelt übertragen zu können. Doch nicht jeder Haushalt umfasst ein solches modernes Spektrum an Haushaltsgeräten. Noch immer finden sich Geräte in den eigenen vier Wänden wieder, welche aus einer Epoche stammen, wo die Universalfernbedienung noch als der heißbegehrteste Vernetzer des 20sten Jahrhunderts galt. Die Übertragung von Signalen über infrarot ist schon längst verjährt aufgrund der Inflexibilität. Das Thema dieser Arbeit soll daraufhin Zielen, diesen Abgrund zu überwinden und das zuverlässige und altbewährte Protokoll wieder in die Moderne zurückrufen. Die **Programmierung einer Universellen Infrarot-Fernbedienung mit einem Arduino UNO** soll ein Meilenstein legen zur Verknüpfung von Smarthome- und überholter Infrarottechnik. Jedoch soll dieses System auch autark funktionieren und beschränkt sich daher zunächst auf nur eine Plattform mit, im Angesicht einer weitgegriffenen Vernetzung, rudimentären Elementen.

Das universelle Infrarotsystem soll korrekte Signale an Geräte senden und zum Einprogrammieren von Befehlen Signale einer Fernbedienung empfangen und diese abspeichern.

Dies kann man in verschiedene Module unterteilen. Zum einen wird für das Senden von infrarot Signalen ein Schaltkreis inklusive der richtigen Implementierung der Komponente auf dem Arduino benötigt. Ebenso muss für den Empfänger die Elektronik sowie die Implementierung übereinstimmen.

Neben dem Sensor und Aktor Duo soll ein Display zur Visualisierung der Steuerung bereitstehen. Die Anzeige soll eine Statusanzeige und -rückmeldung für den Endverbraucher sein und eine Auswahl von Menüs darstellen.

Bedient wird der Aufbau über einen Rotary Encoder, der Einfachheit halber hier als Drehschalter bezeichnet. Der Drehschalter hat von der Form her Ähnlichkeit mit einem Potentiometer lässt sich jedoch in Stufen endlos in zwei Richtungen drehen. Zusätzlich wird beim Drücken auf den Drehschalter ein Knopf bedient. Dadurch soll eine Navigation durch das Menü und Auswahl von Funktionalitäten realisiert werden.

Das ganze Projekt wird auf einem Arduino UNO verwirklicht. Der UNO bietet eine Plattform mit einem Mikroprozessor, welcher sich frei über USB-Schnittstelle Programmieren lässt. Der Arduino stellt für den Controller einfach zu bedienende Eingabe- und Ausgabeschnittstellen zur Verfügung.

2. Beschreibung der Module

2.1. Infrarot Sender

Der Infrarot-Sender ist ein elektronisches Bauteil, das dazu dient, Infrarotlichtsignale zu erzeugen und auszustrahlen.

Funktionsweise: Der Sender arbeitet auf der Basis von Infrarot-LEDs (Light Emitting Diodes), die bei Stromdurchfluss Infrarotlicht emittieren. Durch das Modulieren des Stroms können verschiedene Infrarotcodes erzeugt werden, die von anderen Geräten wie Fernsehern oder Klimaanlage interpretiert werden können.

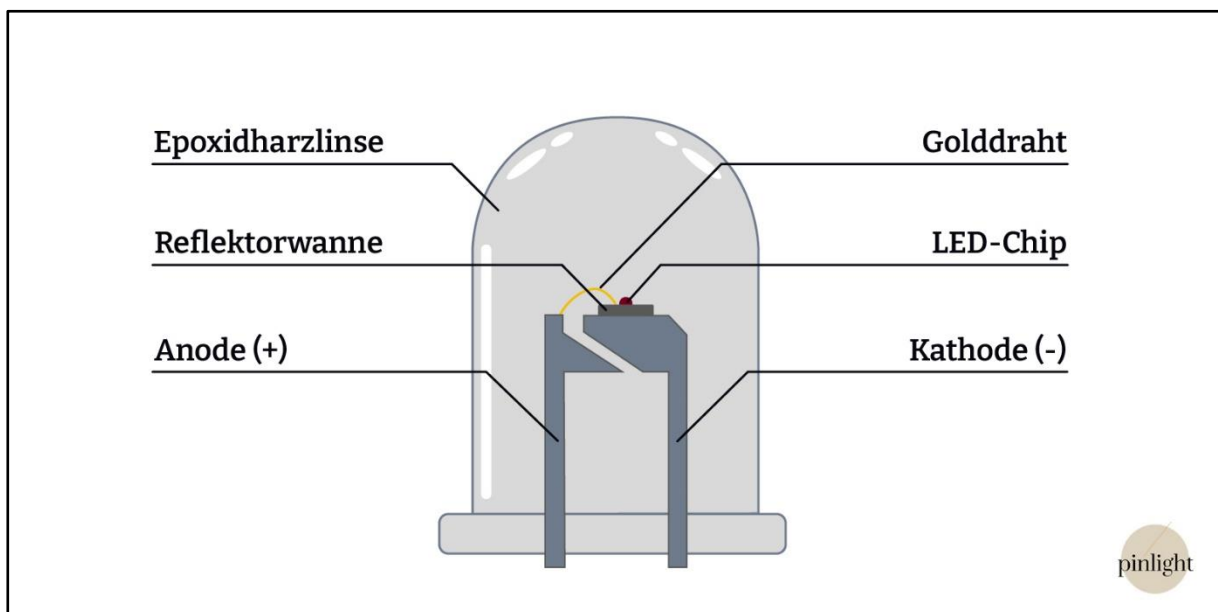


Abbildung 1, LED, funktionsweise (Pinlight, 2024)

Spezifikationen: Typischerweise hat der Infrarot-Sender drei Pins: einen für die Versorgungsspannung (VCC), einen für die Masse (GND) und einen für das Signal (SIGNAL), über den die modulierten Infrarotsignale gesendet werden.

2.2. Infrarot Empfänger

Der Infrarot-Empfänger ist ein Bauteil, das dazu dient, Infrarotlichtsignale zu empfangen und in elektrische Signale umzuwandeln.

Funktionsweise: Der Empfänger enthält einen Fotodetektor, der auf Infrarotlicht reagiert und eine Fotodiode oder einen Fototransistor enthält. Wenn Infrarotlicht auf die Diode trifft, erzeugt sie einen elektrischen Strom, der proportional zur Intensität

des empfangenen Lichts ist. Dieses elektrische Signal wird dann dem Mikrocontroller zur weiteren Verarbeitung übergeben.

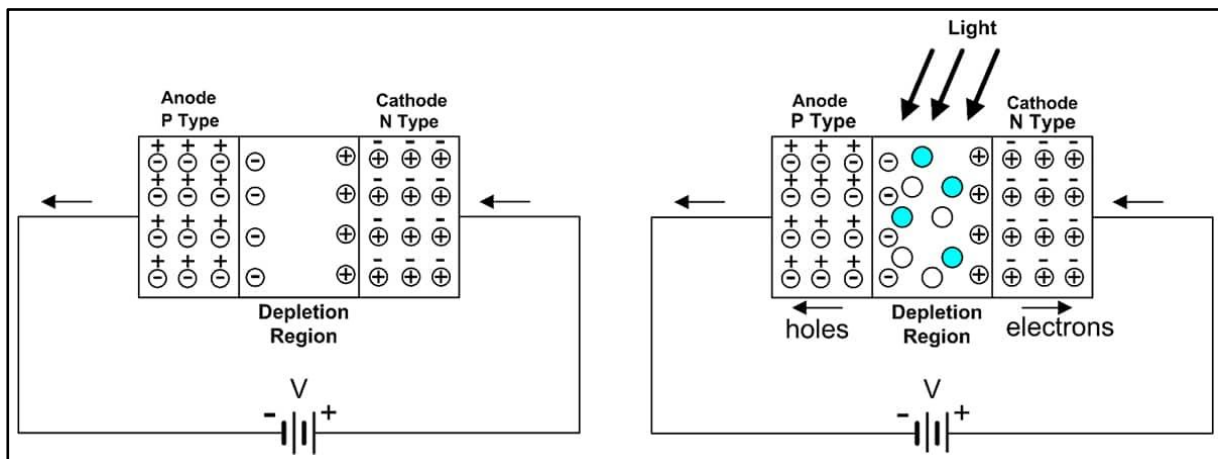


Abbildung 2, Fotodiode, funktionsweise (Digi Key, 2022)

Spezifikationen: Auch der Infrarot-Empfänger hat typischerweise drei Pins: einen für die Versorgungsspannung (VCC), einen für die Masse (GND) und einen Ausgangspin (SIGNAL), über den das empfangene Infrarotsignal an den Mikrocontroller weitergeleitet wird.

2.3. LCD-Display

Ein LCD-Display (Liquid Crystal Display) ist eine Anzeigevorrichtung, die es ermöglicht, Texte und einfache Grafiken darzustellen.

Funktionsweise: Das Display besteht aus einer Matrix von Pixeln, die durch die Anordnung von Flüssigkristallen gesteuert werden. Jedes Pixel kann individuell ein- oder ausgeschaltet werden, um Text oder Grafiken anzuzeigen. Das LCM1602A-Display ist ein gängiges 2x16-Zeichen-LCD, was bedeutet, dass es 2 Zeilen mit jeweils 16 Zeichen darstellen kann.

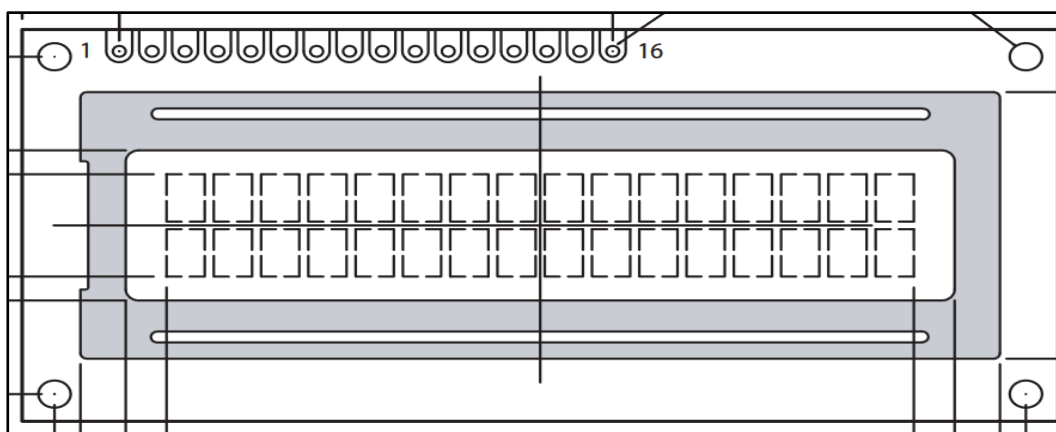


Abbildung 3, Display LCM1602A (Elegoo, 2024)

Spezifikationen: Das LCM1602A-Display hat üblicherweise 16 Pins, die für die Spannungsversorgung, die Datenübertragung und die Steuerung des Displays verwendet werden. Es benötigt eine Betriebsspannung von etwa 5 Volt und kann über einen parallelen oder seriellen Datenbus angesteuert werden.

2.4. Drehschalter (Rotary Encoder)

Ein Drehschalter, auch Rotary Encoder genannt, ist ein Eingabegerät, das verwendet wird, um eine rotierende Bewegung in digitale Signale umzuwandeln.

Funktionsweise: Der KY-040 Rotary Encoder enthält zwei Schalter, die durch Drehung des Encoders betätigt werden. Diese Schalter erzeugen digitale Impulse, wenn der Encoder gedreht wird. Darüber hinaus verfügt der Encoder über einen Taster, der gedrückt werden kann, um ein weiteres Signal zu erzeugen.

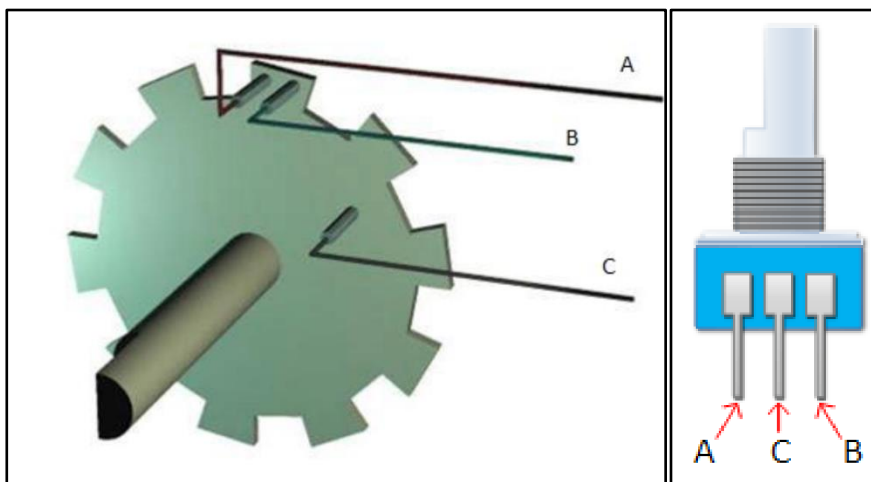


Abbildung 4, Rotary encoder, Keyes KY-040, Funktionsweise (Elegoo, 2024)

Spezifikationen: Der Keyes KY-040 Rotary Encoder hat in der Regel drei Pins: zwei Pins (A und B) für die rechts- bzw. linksdrehende Bewegung und einen Pin (SW) für den Taster. Die Pins werden verwendet, um die Drehrichtung und den Tastendruck an den Mikrocontroller zu übermitteln.

2.5. Arduino UNO

Der Arduino UNO ist ein Open-Source-Mikrocontroller-Board, das auf dem ATmega328P-Chip basiert und eine einfache und flexible Plattform für die Entwicklung von Embedded-Systemen bietet.

Funktionsweise: Der Arduino UNO verfügt über eine Vielzahl von digitalen und analogen Ein- und Ausgangspins, die für die Verbindung mit Sensoren, Aktoren und anderen elektronischen Bauteilen verwendet werden können. Das Board enthält einen Mikrocontroller, der mit einer Arduino-Entwicklungsumgebung programmiert wird, um verschiedene Aufgaben auszuführen.

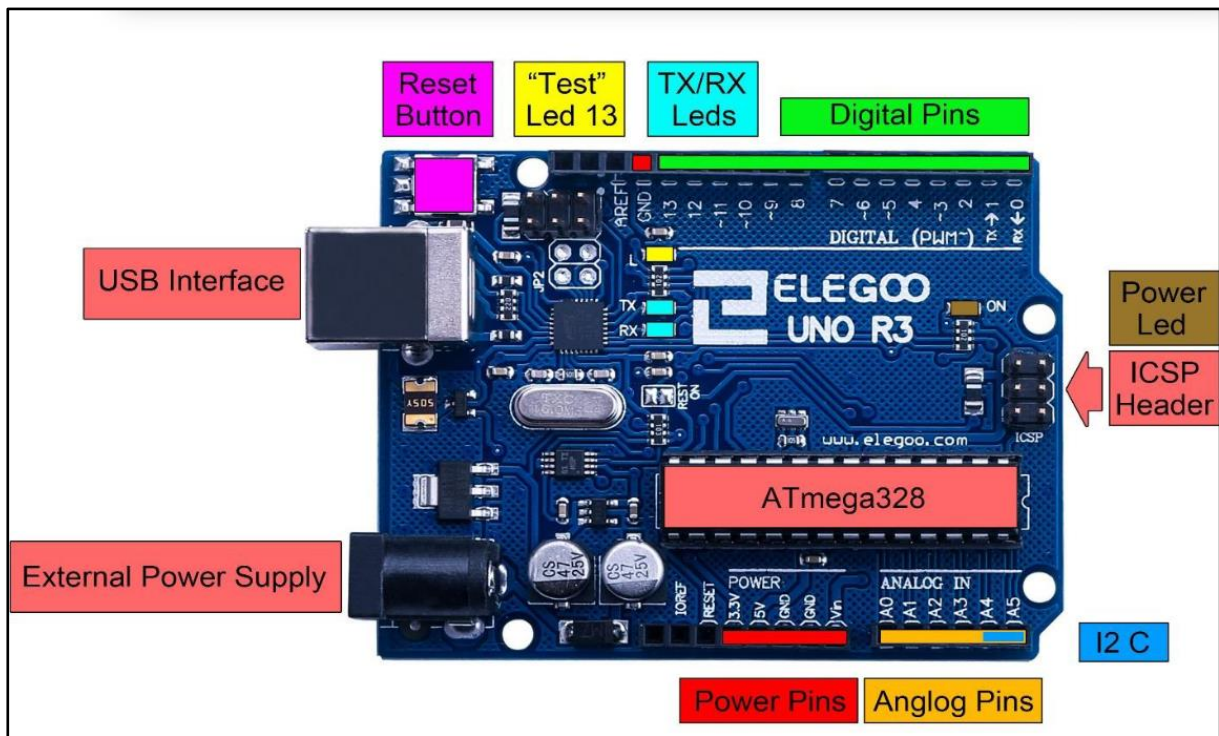


Abbildung 5, Arduino UNO (Elegoo, 2024)

Spezifikationen: Der Arduino UNO verfügt über 14 digitale Ein-/Ausgangspins (von denen 6 PWM-Ausgänge sind), 6 analoge Eingangspins, eine 16-MHz-Keramikresonator, einen USB-Anschluss, eine Stromversorgungsbuchse, einen ICSP-Header und einen Reset-Taster. Er wird in der Regel mit einer Betriebsspannung von 5 Volt betrieben und kann über USB oder eine externe Stromquelle mit Strom versorgt werden.

3. Funktionsweise der Module

3.1. Infrarot Sender

Der Infrarot-Sender erzeugt Infrarotlichtsignale, die von anderen Geräten wie Fernsehern, DVD-Playern oder Klimaanlage interpretiert werden können. Dies geschieht durch das Modulieren des Stroms, der durch die Infrarot-LED fließt. Die Modulation erfolgt gemäß einem bestimmten Protokoll, das von den meisten Infrarot-Fernbedienungen unterstützt wird, wie z.B. dem NEC- oder dem Sony-Protokoll. Dabei werden digitale Daten in Form von Ein- und Ausschaltimpulsen auf die Infrarot-LED übertragen, wodurch verschiedene Befehle codiert werden.

Um den Infrarot-Sender anzusteuern, müssen die entsprechenden digitalen Pins des Arduino UNO verwendet werden. Es gibt verschiedene Bibliotheken, die die Erzeugung von Infrarotsignalen vereinfachen, wie z.B. die "IRremote" Bibliothek für Arduino. Diese Bibliothek enthält Funktionen zum Senden von Infrarotsignalen gemäß verschiedenen Protokollen und erleichtert somit die Integration des Infrarot-Senders in das Arduino-Projekt.

Bei der Verwendung des Infrarot-Senders sind bestimmte Protokolle zu beachten, um sicherzustellen, dass die gesendeten Signale von den empfangenden Geräten korrekt interpretiert werden. Außerdem ist darauf zu achten, dass die Infrarot-LED richtig ausgerichtet ist und sich in der Sichtlinie des zu steuernden Geräts befindet.

Für die Ansteuerung des Infrarot-Senders kann die "IRremote" Bibliothek für Arduino verwendet werden. Diese Bibliothek bietet eine Reihe von Funktionen zum Senden von Infrarotsignalen gemäß verschiedenen Protokollen und macht die Integration des Infrarot-Senders in das Arduino-Projekt einfach und effizient.

3.2. Infrarot Empfänger

Der Infrarot-Empfänger empfängt Infrarotlichtsignale von einer Fernbedienung oder einem anderen Infrarotsender und wandelt sie in elektrische Signale um, die vom Arduino UNO verarbeitet werden können. Der Empfänger enthält einen Fotodetektor, der auf die Wellenlänge des Infrarotlichts abgestimmt ist, sowie einen integrierten Verstärker, um das empfangene Signal zu verstärken.

Um den Infrarot-Empfänger anzusteuern, wird der Ausgangspin des Empfängers mit einem digitalen Pin des Arduino UNO verbunden. Die Bibliothek "IRremote" für Arduino bietet Funktionen zum Decodieren der empfangenen Infrarotsignale und ermöglicht es

dem Arduino, die gesendeten Befehle zu interpretieren und entsprechend zu reagieren.

Beim Einsatz des Infrarot-Empfängers ist darauf zu achten, dass er richtig ausgerichtet ist und das Infrarotsignal von der Fernbedienung oder dem Sender empfangen kann. Es können verschiedene Protokolle verwendet werden, daher ist es wichtig sicherzustellen, dass der Empfänger das verwendete Protokoll unterstützt.

Die "IRremote" Bibliothek für Arduino bietet Funktionen zum Decodieren von Infrarotsignalen und erleichtert somit die Integration des Infrarot-Empfängers in das Arduino-Projekt.

3.3. LCD-Display

Das LCD-Display dient zur Anzeige von Texten und einfachen Grafiken. Es besteht aus einer Matrix von Pixeln, die durch die Anordnung von Flüssigkristallen gesteuert werden. Jedes Pixel kann individuell ein- oder ausgeschaltet werden, um Text oder Grafiken darzustellen. Das LCM1602A-Display ist ein 2x16-Zeichen-LCD, was bedeutet, dass es zwei Zeilen mit jeweils 16 Zeichen darstellen kann.

Das LCD-Display wird über einen parallelen oder seriellen Datenbus mit dem Arduino UNO verbunden. Um Texte oder Grafiken auf dem Display darzustellen, werden entsprechende Befehle an das Display gesendet, die die gewünschten Zeichen oder Symbole aktivieren oder deaktivieren.

Das LCM1602A-Display hat eine feste Auflösung von 2x16 Zeichen, daher können maximal 32 Zeichen gleichzeitig angezeigt werden. Die Darstellung von Grafiken ist aufgrund der begrenzten Auflösung und des Zeichensatzes des Displays begrenzt.

Für die Ansteuerung des LCD-Displays können verschiedene Bibliotheken für Arduino verwendet werden, wie z.B. die "LiquidCrystal" Bibliothek. Diese Bibliothek erleichtert die Initialisierung und Verwendung des LCD-Displays und bietet Funktionen zum Schreiben von Texten und Steuern der Anzeige.

3.4. Drehschalter (Rotary Encoder)

Ein Drehschalter, auch als Rotary Encoder bekannt, wandelt die rotierende Bewegung eines Schalters in digitale Signale um. Der Keyes KY-040 Rotary Encoder besteht aus einem drehbaren Knopf und zwei Schaltern, die beim Drehen des Encoders betätigt werden. Diese Schalter erzeugen digitale Impulse, wenn der Encoder gedreht wird, und ein weiterer Schalter erzeugt ein Signal, wenn der Knopf gedrückt wird.

Der Keyes KY-040 Rotary Encoder wird über seine drei Pins (A, B und SW) mit dem Arduino UNO verbunden. Die Pins A und B senden Impulse, wenn der Encoder gedreht wird, und der Pin SW sendet ein Signal, wenn der Knopf gedrückt wird. Durch Zählen der Impulse auf den Pins A und B und Auswerten des Signals auf dem Pin SW kann die Drehrichtung und der Tastendruck des Encoders erkannt werden.

Bei der Verwendung des Rotary Encoders ist zu beachten, dass er eine mechanische Komponente ist und daher eine gewisse Entprellung (Debouncing) erfordern kann, um zuverlässige Signale zu liefern. Außerdem kann die Anzahl der Impulse pro Umdrehung je nach Encoder variieren, daher ist es wichtig, die Impulse entsprechend zu zählen und zu interpretieren.

Für die Ansteuerung des Rotary Encoders gibt es verschiedene Bibliotheken für Arduino, die die Interaktion mit dem Encoder vereinfachen können. Eine häufig verwendete Bibliothek ist beispielsweise die "RotaryEncoder" Bibliothek, die Funktionen zum Zählen der Impulse und Erkennen von Tastendrücken bereitstellt. Diese Bibliothek macht es einfach, den Rotary Encoder in das Arduino-Projekt zu integrieren und seine Funktionen zu nutzen.

4. Die Universelle Infrarot-Fernbedienung

Die Logik der Hauptanwendung für die Universelle Infrarotfernbedienung entsteht aus der Zusammensetzung der einzelnen Module heraus. Zu den physischen Aktoren und Sensoren kommen Grunddefinitionen und eine Klasse zum Realisieren der Menüführung hinzu. Das unter Abbildung 6 aufgeführte Klassendiagramm veranschaulicht die modulare Zusammensetzung der Zielanwendung. Für jedes unter drittens aufgeführtes Modul steht eine Klasse bereit.

Die Sensor-Aktor Klasse beinhaltet die in den Testanwendungen erprobten Implementationen und Funktionen zum Senden und Empfangen von infrarot Signalen. Die Membervariablen „IRrecv“ und „IRsend“ sind Objekte der aus einer Infrarotbibliothek von Arduino bereitgestellten Klassen für die Sensor und Aktor Elektronik. Das Senden erfolgt auf die übergebenen Parameter an die „Send()“ Funktion. Die „Receive()“ Funktion wartet auf den Eingang eines Infrarotsignals und gibt die empfangen Parameter zurück.

Die Drehschalterklasse verfügt über nur eine einzige Funktion. Dem „Checkup()“ wird ein Zeiger einer Funktion übergeben, ein sogenannter „Callback“, welche nur aufgerufen wird, wenn eine Änderung an der Eingabeelektronik feststellbar ist. Ebenfalls wird die Information, auf welche Art der Drehschalter bewegt wurde über die Enumeration „Menü_Titel“ mit übergeben.

Die Displayklasse kann zwei Zeichenketten, welche an die „Text_aktualisieren()“ Funktion übergeben werden untereinander auf dem LCD-Display darstellen.

Die Menüklasse speichert die Inhalte der Menüs in Form von Arrays. Jedes Untermenü wird von einem separaten Array verkörpert. Ein Arrayelement besteht hierbei aus der „Menü_Eintrag“ Struktur. Über die Funktion „Get_Entry()“ erhält man einen Zeiger auf den aktuell ausgewählten Menüeintrag. Mittels „Manipulate_Entry()“ lässt sich ein Eintrag überschreiben. Die „Check_Index()“ Funktion ist da, um zu überprüfen, ob ein Index noch innerhalb der Grenzen des aktuellen Menü-Arrays ist. Diese Funktion ist wichtig, da C++ keinen falschen Indexzugriff erkennen kann. Das hätte unkontrollierte Speicherzugriffe zu Folge, die das Programm zu Absturz bringen würden. Die Struktur eines Menüeintrages besteht aus dem Namen als Zeichenkette, einem Datenobjekt von „IR_Data“ und einem Element der Enumeration „Menü_Titel“. Der Title gibt an, ob sich hinter dem Eintrag eine Funktion oder ein weiteres Menü verbirgt und wenn ja, welches.

Diese Module sind als Objekte im Hauptsystem implementiert.

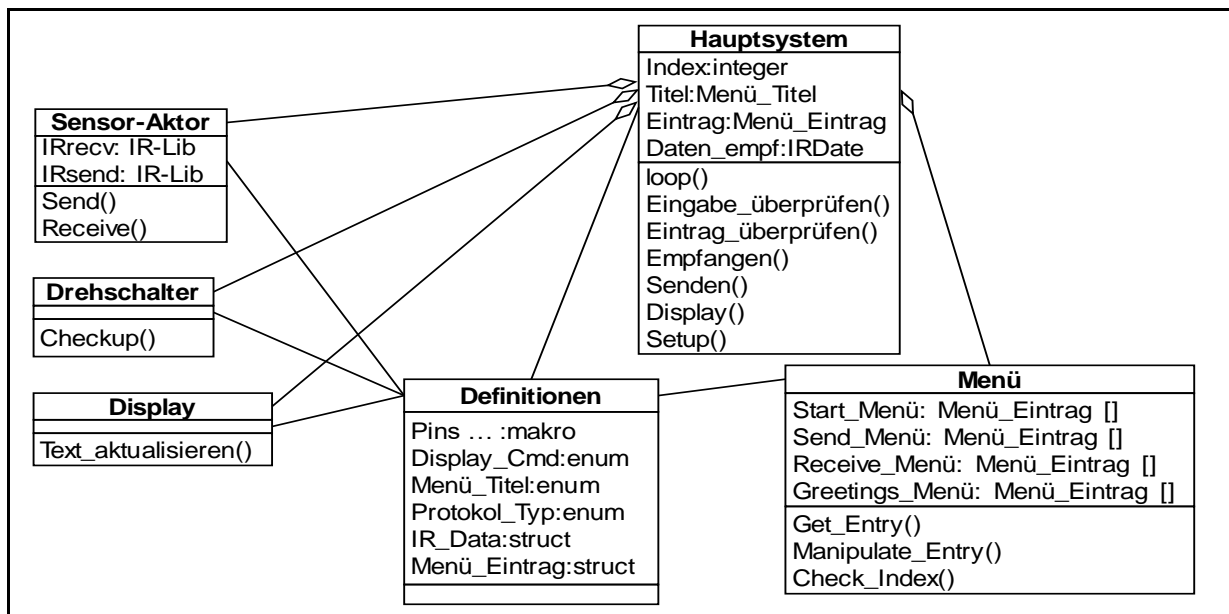


Abbildung 6, Klassendiagramm der Zielanwendung

Das Hauptsystem bildet das Herzstück der Anwendung. Im Ablaufdiagramm von Abbildung 7 sind die inneren Routinen verbildlicht. Diese laufen in Dauerschleifen, wartend auf eine Eingabe. Die Art der Eingabe und der Inhalt des ausgewählten Menüeintrages bestimmen die ausgeführte Funktionalität.

Noch vor Beginn der Hauptfunktionen werden im „Setup()“ alle benötigten Objekte initialisiert und Startwerte festgelegt. Es wird im Begrüßungsmenü ab Index null begonnen. Der Index beschreibt, an welcher Stelle des aktuellen Menüarrays man sich befindet. Auf dem Display wird bei Aktualisierung der aktuelle und der nachfolgende Menüeintrag über den Index und das ausgewählte Array ermittelt und dargestellt.

Anschließend startet das System in der Hauptschleife und wartet auf die Eingabe durch den Drehschalter technisch als „Rotary-Encoder“ bezeichnet. Das Drehen des Encoders inkrementiert oder dekrementiert den aktuellen Anzeigeindex. Woraufhin die anzeige erneuert wird.

Beim Drücken des Encoders und somit Bestätigen, wird über den Index und den Titel des aktuellen Menüs der aktuelle Menüeintrag ermittelt. Das Überprüfen des Eintrages legt offen, ob ein weiteres Menü geöffnet oder eine Funktion ausgeführt werden soll. Der Menüwechsel wird durch Angabe des neuen Menütitels und den Index null in die Wege geleitet.

Beim Ausführen einer Funktion könnte es sich, je nach dem in welchem Menü man sich befindet, um das Empfangen von Daten handeln. Das Einläuten der Empfangsschleife lässt das System auf Eingang von Signalen warten und hält diese bereit.

Sind gerade Daten empfangen worden, wird bei der Auswahl aus dem Sende-Menü nichts gesendet, sondern der Eintrag mit den Empfangenen Daten überschrieben.

Wurden zuvor keine Daten empfangen, so wird beim Auswählen aus dem Sende Menü ein infrarotes Signal gesendet.

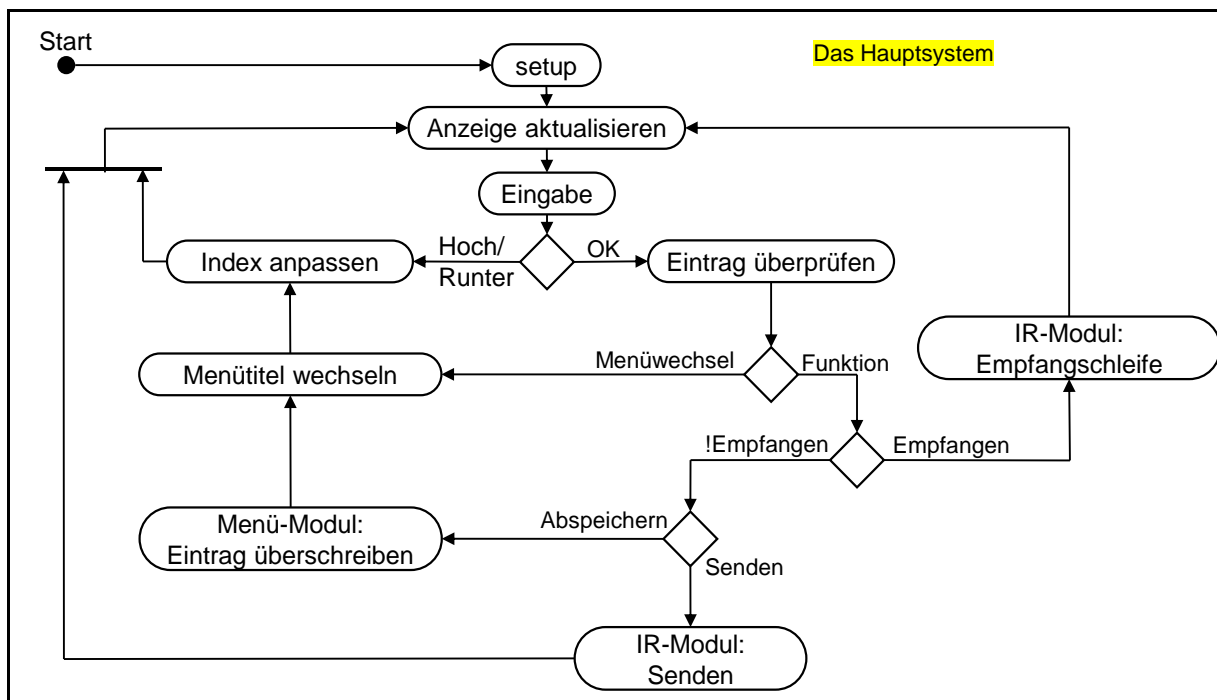


Abbildung 7, Ablaufdiagramm des Hauptsystems

5. Der Aufbau

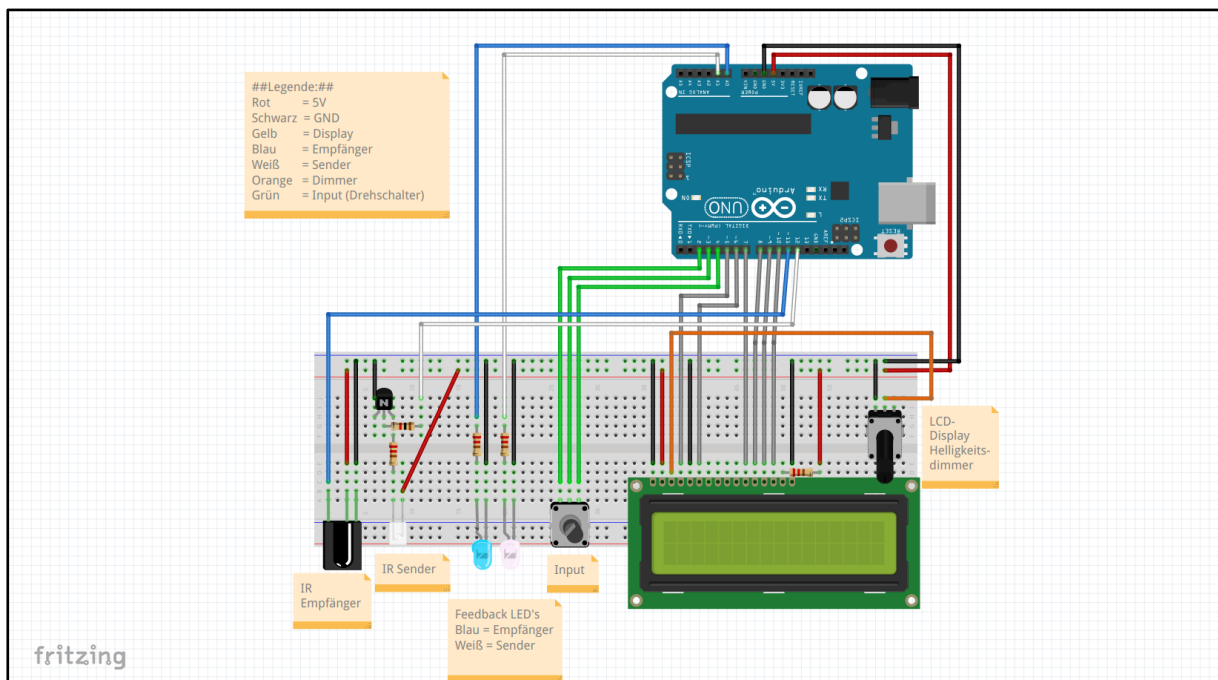


Abbildung 8, Technischer Aufbau als fritzing Sketch

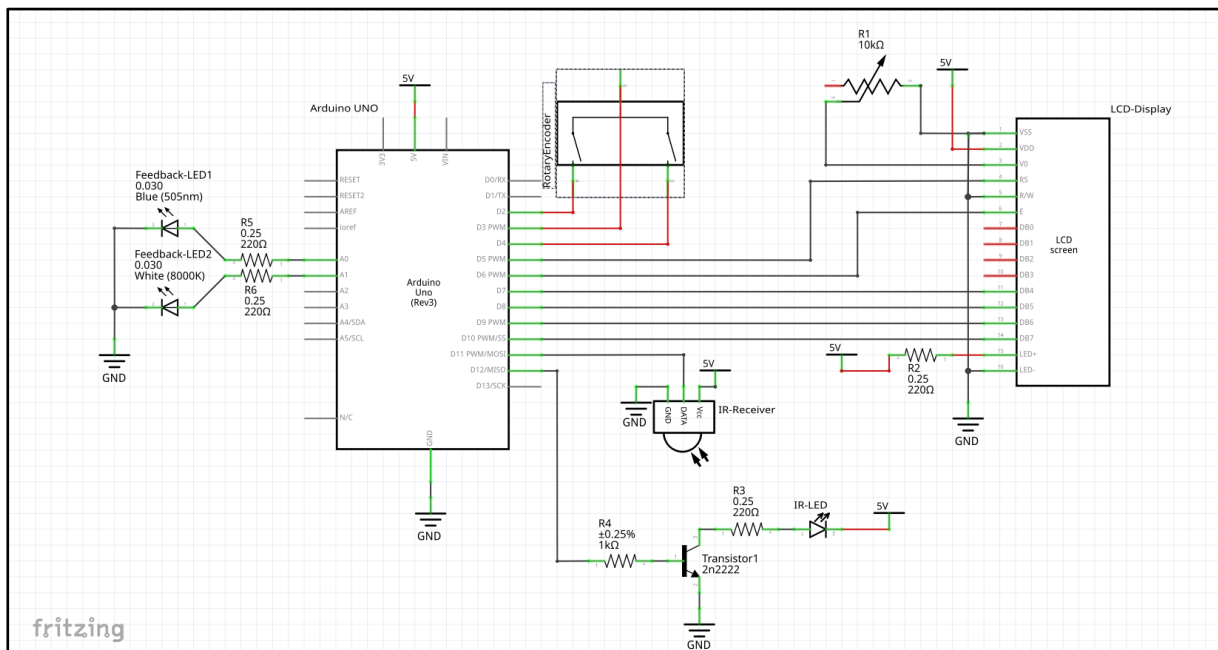


Abbildung 9, Technischer Aufbau als Blockschaltbild

Das LCD-Display ist folgendermaßen verkabelt.:

D5 (PWM - Pulse Width Modulation): Dieser Pin ist mit dem RS-Pin (Register Select) des LCD-Displays verbunden. Durch Anlegen eines HIGH- oder LOW-Signals kann zwischen dem Senden von Daten (HIGH) und Befehlen (LOW) an das Display umgeschaltet werden.

D6 (PWM) – verbunden mit dem E-Pin (Enable) des LCD-Displays. Durch Pulsen dieses Pins kann das Display aktiviert werden, um Daten oder Befehle zu verarbeiten.
D7-D10: Diese Pins sind mit den Datenpins (DB4-DB7) des LCD-Displays verbunden. Über diese Pins werden die Daten zum Anzeigen von Zeichen oder Steuerbefehlen an das Display gesendet.

D9 (PWM): Dieser Pin ist mit dem Backlight-Pin des LCD-Displays verbunden. Durch Anlegen eines PWM-Signals kann die Helligkeit der Hintergrundbeleuchtung gesteuert werden.

Verkabelung Sender: D12 (MISO - Master In Slave Out): Dieser Pin wird verwendet, um den IR-Sender anzusteuern. Er sendet Infrarotsignale an andere Geräte und ermöglicht es dem Arduino Uno, Befehle oder Daten an diese Geräte zu senden.

Verkabelung Empfänger: D11 (PWM/MOSI - Master Out Slave In): Der IR-Empfänger ist mit diesem Pin verbunden, der sowohl PWM- als auch MOSI-Funktionen bietet. Er empfängt Infrarotsignale von einer Fernbedienung oder einem anderen IR-Sender und wandelt sie in digitale Signale um, die vom Arduino Uno verarbeitet werden können.

Verkabelung Drehregler: D2, D3 (PWM) und D4: Diese Pins des Arduino Uno werden verwendet, um den RotaryEncoder anzusteuern. Sie ermöglichen die Erfassung der Drehbewegung und -richtung sowie die Bestimmung der Drehgeschwindigkeit.

Verkabelung Feedback LED's: A0 und A1 (Analogeingänge): Diese Pins sind mit den Feedback-LEDs verbunden, die dem Benutzer visuelles Feedback geben. Die Helligkeit der LEDs kann durch Anpassen der Spannung an den Analogeingängen variiert werden.

Auf den nachfolgenden Abbildungen 10 und 11 ist die für unser Beispiel verwendete Schaltung zu sehen.

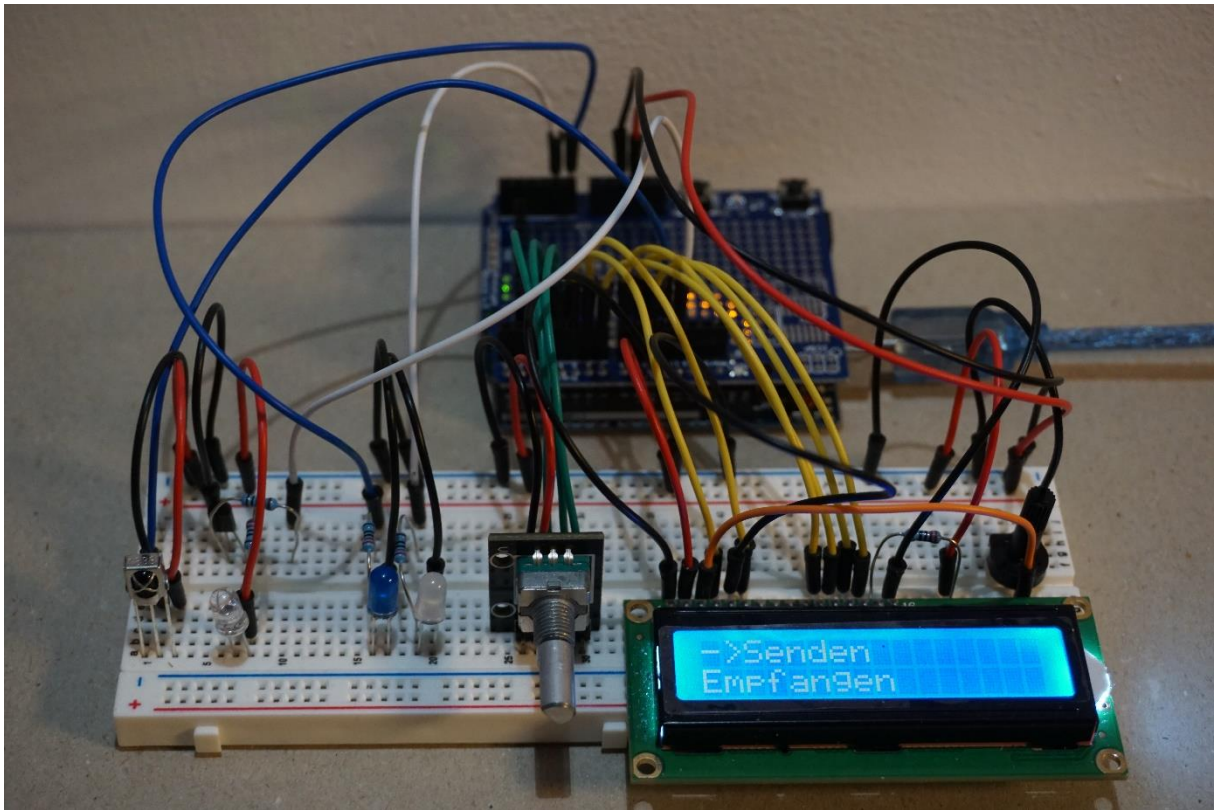


Abbildung 10, Tatsächlicher Aufbau auf Steckbrett, Hauptmenü

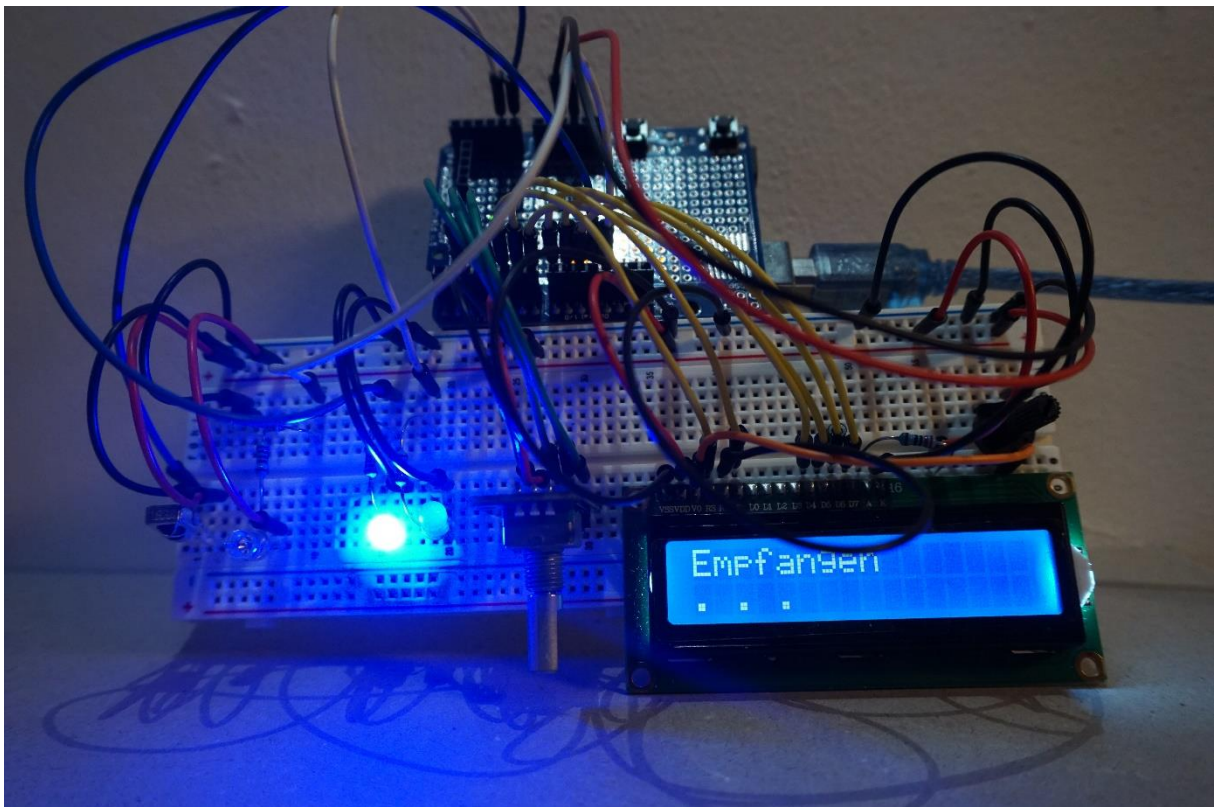


Abbildung 11, Tatsächlicher Aufbau auf Steckbrett, Empfangsmenü

6. Installationshinweise

In unserem GitHub (https://github.com/TIStudent420/Arduino_IR_Remote) befindet sich das komplette Projekt. Im Ordner „Quellcode/_Merged/base_system/“ sind die Quellcodedateien von den einzelnen Klassen, Definitionen und das Hauptsystem zu finden. Zum Kompilieren müssen zuvor die benötigten Bibliotheken in die Arduino IDE eingebunden werden. Hinweise dazu gibt es unter „Quellcode/Bibliotheken_hinzufügen.txt“. Alle verwendeten Bibliotheken sind im Ordner „Quellcode/Bibliotheken“ hinterlegt.

7. Test und Fazit

Im Umfang dieser Arbeit wurde erfolgreich die dynamische Bedienmöglichkeit von infrarotgesteuerten Geräten mit vor und während der Laufzeit programmierbarer Befehle implementiert. Die Zielgeräte lassen sich in einer Testumgebung wie gewünscht ansteuern.

Das Brettmuster mit dem Arduino ist nur etwas unpraktisch in der Handhabung. Im praktischen Gebrauch muss man ebenfalls feststellen, dass die Sendeleistung der infrarot LED zu wünschen übriglässt. Ein Problem, dass mit einer verstärkenden Beschaltung, eine anderen LED oder eine Linse eine Lösung gefunden hätte. Abgesehen von diesen Umgebungsfaktoren funktioniert das System zuverlässig.

Der Modulare Aufbau erleichtert ein Austauschen oder Ersetzen von Komponenten sehr. Die Eingabeanimation des Drehschalters könnte durch einen Raspberry Pi ersetzt werden, welcher seine Befehle von einem Server erhält. Damit stünde einer großflächigeren Vernetzung nichts mehr im Wege.

Abgekapselt von einem Smarthome Framework funktioniert die IR-Steuerung jedoch auch autark. Dank einer umfangreichen Bibliothek mit Funktionen zum Senden von Infrarot Signalen kann auch ein Großteil von kommerziell verwendeten Infrarotprotokollen bedient werden. Mit den richtigen Befehlen gelingt somit jede Kommunikation.

8. Quellen

Digi Key. (11. 01 2022). Fotodiode. *Die Grundlagen zu Fotodioden und Fototransistoren und ihrer Anwendung*. Abgerufen am 04. 04 2024 von <https://www.digikey.ch/de/articles/the-basics-of-photodiodes-and-phototransistors-and-how-to-apply-them>

Elegoo. (04. 04 2024). Das Umfangreichste Starter Kit - Elegoo UNO R3. (v1.0.17.7.10). China. Datenblättersammlung von CD. Kit unter [ELEGOO.com](https://www.elegoo.com) auffindbar

Pinlight. (04. 04 2024). LED einfach erklärt - Definition, Funktion und mehr. Abgerufen am 04. 04 2024 von https://pinlight.eu/e/led/?utm_referrer=https%3A%2F%2Fwww.google.com%2F