**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,

**Praxispartner:** BA-Glauchau

**Gutachter:** Prof. A. Munke

Themenblatt Unterschrieben hier einfügen

Inhaltsverzeichnis

**1. Grundidee/Zielsetzung/Erläuterung 1**

**2. Beschreibung der Module 2**

2.1. Infrarot Sender 2

2.2. Infrarot Empfänger 2

2.3. LCD-Display 2

2.4. Drehschalter (Rotary Encoder) 2

2.5. Arduino UNO 2

**3. Funktionsweise der Module 3**

3.1. Infrarot Sender 3

3.2. Infrarot Empfänger 3

3.3. LCD-Display 3

3.4. Drehschalter (Rotary Encoder) 3

**4. Die Universelle Infrarot-Fernbedienung 4**

**5. Der Aufbau 5**

**6. Test und Fazit 6**

**7. Quellen 7**

TODO: Format vom Inhaltsverzeichnis muss angepasst werden (Absätze & 1.Ebene fett)

1. Grundidee/Zielsetzung/Erläuterung

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 inclusive 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 Programmieren lässt. Der Arduino stellt für den Controller einfach zu bedienende Eingabe- und Ausgabeschnittstellen zur Verfügung.

Die Aufgabenstellung soll als erfüllt betrachtet werden, wenn sich Infrarotgeräte jeglicher Art damit dynamisch ansteuern lassen.

1. Beschreibung der Module

Kurze Beschreibung, Was ist das, wie funktioniert es (nur grob), Spezifikationen

Datenblätter unter: „Dokumentation/wichtige Datenblätter/“

* 1. Infrarot Sender

Ist halt ne LED…

* 1. Infrarot Empfänger

Datenblatt, hat 3 Pins, vielleicht ein wenig erklären, wofür die 3 Pins und wie das Signal aussieht, wenns raus kommt. Bzw. wie und warum da ein Signal raus kommt??

* 1. LCD-Display

Datenblatt: „Dokumentation/wichtige Datenblätter/LCD-Display.pdf“, Bezeichnung: LCM1602A, 2x16 Segmente, …, was noch so für die Darstellung wichtig ist

* 1. Drehschalter (Rotary Encoder)

Datenblatt: „Dokumentation/wichtige Datenblätter/Rotary-Encoder(Drehschalter).pdf“,

Keyes KY-040 Rotary Encoder. 3 Pins. 2 für rechts & links Drehbewegung, 1 für Knopf (runter drücken) … wie und warum?

* 1. Arduino UNO

Wie viele Pins, Analog/Digital, Spannungsversorgung, …allgemeines

1. Funktionsweise der Module

Beispielanwendungen liegen unter „/Quellcode/“

Vielleicht grob erklären, wie die Dinger funktionieren, welche Ansteuerung/Funktion wichtig ist, vielleicht auch die Bibliothek erwähnen

Bibliotheken und deren Quellen unter: „Quellcode/Bibliotheken/

* 1. Infrarot Sender

„/Quellcode/IR\_Remote/IR\_LED\_Test/“

Wie sende ich Signale, welche Besonderheiten sind zu beachten (Protokolle?)

Welche Bibliothek? -> „/Quellcode/Bibliothek/“

* 1. Infrarot Empfänger

„/Quellcode/IR\_Remote/IR\_Receiver\_Test/“

Wie empfange ich Signale, welche Besonderheiten sind zu beachten (Protokolle?)

Welche Bibliothek? -> „/Quellcode/Bibliothek/“

* 1. LCD-Display

„Quellcode/Display“

Wie stell ich sachen auf dem Display dar, welche Limits gibt es (2x16)

* 1. Drehschalter (Rotary Encoder)

„/Quellcode/Rotary\_Switch/Rotay\_Switch\_Test/“

…

1. 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 1 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 Signals 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 aufgerufen wird, sobald eine Änderung an der Eingabeelektronik feststellbar ist. Ebenfalls wird die Information, auf welche Art der Drehschalter bewegt wurde mit übergeben.

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

Die Menüklasse hat die Inhalte der Menüs in Form von Arrays aus der Eintragsstruktur der Definitionen. Über die Funktion „Get\_Entry()“ erhält man einen Zeiger auf den gewünschten 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. Die Struktur eines Menüeintrages besteht aus dem Name 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 1,** Klassendiagramm der Zielanwendung

Das Hauptsystem bildet das Herzstück der Anwendung. Im Ablaufdiagramm von Abbildung 2 sind die inneren Rutinen 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 beim 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 infrarot Signal gesendet.



**Abbildung 2**, Ablaufdiagramm des Hauptsystems

1. Der Aufbau

Bilder unter Dokumentation:

- Aufbau\_Brettmuster.png

- Aufbau\_Brettmuster-Schemtisch.png

- Bild von Unserem Brettmuster live muss noch gemacht werden!

Evtl kurz was zu den Pins und so …einfach die Bilder beschreiben

(Die Farben der Kabel sind einheitlich gewählt)

Bilder sind mit der Software „Fritzing“ entstanden (<https://fritzing.org>)

Fritzing Dokument liegt unter „/Schaltplan/IR\_Aufbau\_Sketch.fzz“

1. Test und Fazit

Sendeleistung lässt zu wünschen übrig.

Der ganze schnickschnak ist zimlich aufwendig und unhandlich

Das System funktioniert aber gut, Top Sache

Munke, Bitte geben sie und die 1

Ebnet den Weg das ganze in ein eigenes Smarthome-framework einzubinden

1. Quellen

!Auf jedenfall nicht ChatGPT (chat.openai.com)

Unter Reiter **Referenzen** und **Zitate und Literaturverzeichnis** auf **Quellen verwalten** und dann mit **Neu** eintragen.

* Das Quellenverzeichnis wird dann automatisch erstellt
* Quellen ganz einfach mit Zitat einfügen im Text oder als Fußnote hinzufügen