Umgebungsbewerter

Ben Joel Schönbein
769772

Bauteile:

- 1 Arduino Uno
- 1 DHT22 Sensor
- 1 SSD1331 OLED Display
- 1 ARD MIC MODUL2
- 1 Plastikhülle

Idee

Die Idee meines Mikrocontroller Projektes war es, ein handliches Gerät zur Messung der Umgebungstemperatur, Luftfeuchtigkeit, sowie der Lautstärke zu bauen. Diese Daten werden im Mikrocontroller anhand der optimalen Temperatur [tmp], Luftfeuchtigkeit [hmd] und Lautstärke eines Schlafzimmers [snd] bewertet und dann über das OLED-Display ausgegeben. Dabei ist die Bewertung als dynamische Farbe erkennbar.

Planung

Anfangs wollte ich das Projekt mit einem Raspberry Pi Pico und Micropython umsetzen, was sich im späteren Verlauf als problematisch aufwies, weshalb ich auf den Arduino Uno mit C++ wechselte. Zur Messung der Temperatur und Luftfeuchtigkeit stand von Anfang an fest, dies mit dem DHT22 umzusetzen. Die farbliche Ausgabe sollte mittels LED-Streifen geschehen, dazu sollte es ein 16x2 LCD Display zur Ausgabe der Werte geben, was sich im späteren Verlauf ebenfalls zu dem SSD1331 OLED-Display änderte. Der Sound Sensor sollte als letztes eingebaut werden, da das Grundgerüst zur Messung der Temperatur und Luftfeuchtigkeit sowie der Ausgabe auf einem Display fertig sein sollte.

Umsetzung

Als erstes habe ich mich mit dem Raspberry Pi Pico, sowie Micropython vertraut gemacht. Die Implementation des DHT22 ging ohne Probleme, da dieser von Micropython selbst unterstützt wird. Jedoch gab es kein Package zur Steuerung von LED-Streifen, welches Micropython unterstützt, weshalb relativ zeitaufwendig ein Package gesucht werden musste. Hierbei fand ich das Micropython-Dotstar Repository, welches einwandfrei läuft. Der nächste Schritt war, das System mit einem Display zu verbinden. Anfangs versuchte ich es mit einem seriellen 16x2 LCD Display, wobei ich das SPI-Protokoll nutzte. Es war zwar möglich Strings an das Display zu schicken, welches diese auch ausgab, nur konnte das Display nicht zurückgesetzt werden, was aus dem Problem entsteht, dass man keine Commands von dem Mikrocontroller aus an das Display schicken konnte. In Micropython gibt es bisher ebenfalls kein Package, welches die Steuerung des Displays unterstützt. Daraufhin versuchte ich es mit einem OLED SSD1331 Display. Nach langer Recherche und Suche nach einem Package, mit welchem man dieses Display mittels Micropython steuern kann und nichts gefunden wurde, entschloss ich, dass Micropython von vielen Geräten noch nicht Unterstützt wird, weshalb der Mehraufwand zu hoch ist, weiter mit dem Raspberry Pi Pico

und Micropython zu arbeiten. Deshalb bin ich daraufhin auf den Arduino Uno gewechselt, wo es für jedes verwendete Gerät ein jeweiliges Package gibt.

Arduino Uno

Durch meine Vorerfahrung mit C++ war das Einfinden in den Arduino Uno simpel. Da dieser Mikrocontroller weit verbreitet ist, ist das Finden von benötigten Packages und Tutorials sehr einfach und benötigt wenig Aufwand. Ein Nachteil, der sich für mich mit dem Arduino Uno gegenüber des Raspberry Pi Pico ergab, war , dass der Arduino Uno nur einen SCL und SDA Pin besitzt, weshalb ich nur das OLED Display nutzen konnte, anstatt geplant das Display und ein LED-Streifen.

DHT22

Durch das DHT22 Package ist das Auslesen der Umgebungstemperatur und Luftfeuchtigkeit sehr leicht, man muss nur den jeweiligen Data-Pin auslesen. Mit dieser Komponente gab es keine Probleme.

SSD1331 OLED Display

Diese Komponente benötigt ebenfalls ein Package, welches im Arduino Uno gegeben ist. Die Benutzung ist sehr intuitiv, mit Commands, mit welchen man auf das Display Strings auf bestimmten Positionen, mit einer bestimmten Farbe und Textgröße, schreiben kann. Die einzige Komplikation war, dass die Farben nicht im RGB888 Format definiert werden, sondern im RGB565 Format. Dazu musste man die Farben mittels einer Funktion konvertieren, diese Dokumentation hat dabei geholfen.

ARD MIC MODUL2

Der Vorteil dieses Sound Sensors ist, dass man kein Package benötigt, um ihn zu benutzen. Die Nachteile sind jedoch, dass nicht die Umgebungslautstärke als dB gemessen wird, sondern dieser Sensor ab einem bestimmten Threshold aktiviert wird. Dazu ist diese Grenze nicht konstant, weshalb die Lautstärke, bei der dieser Sensor auslöst, von Zeit zu Zeit unterschiedlich ist.

Programmcode

Der Mikrocontroller misst alle 2 Sekunden die Pins, die mit den Sensoren verbunden sind. Ich habe ein Ratingsystem entwickelt, welches die Temperatur, Luftfeuchtigkeit und Lautstärke zuerst unabhängig voneinander bewertet, diese Werte der Sensoren werden auf dem Display ausgegeben, dabei ist die Farbe des jeweiligen Temperatur und Luftfeuchtigkeit Wertes dynamisch, was bedeutet, dass die Farbe abhängig von dem jeweiligen Wert

gegenüber der vordefinierten optimal Werte, gesetzt wird. Bei der Lautstärke wird entweder “loud” oder “quiet” ausgegeben, abhängig davon, ob der Sensor auslöst. Um die Gesamtwertung zu berechnen, wird den Werten noch abhängig von dem Bereich, in welchem sich diese befinden, ein Wert von 0 bis 2 zugeordnet, wobei 0 gleich gut, 1 gleich ok und 2 gleich schlecht ist. Bei der Lautstärke gibt es nur die Wertungen 0 oder 2. Die Bewertungen der Messwerte werden addiert, wodurch sich die Gesamtbewertung ergibt. Diese wird auf dem Display ebenfalls textuell und farblich dargestellt.

[tmp] optimale Zimmertemperatur, Umweltbundesamt
https://www.umweltbundesamt.de/umwelttipps-fuer-den-alltag/heizen-bauen/heizen-raumtemperatur#gewusst-wie

[hmd] optimale Luftfeuchtigkeit im Schlafzimmer, getAir
https://www.getair.eu/wissen/optimale-luftfeuchtigkeit-so-schuetzen-sie-gesundheit-und-gebaeude/#:~:text=Eine%20optimale%20Luftfeuchtigkeit%20sollte%20zwischen,die%20Luftfeuchtigkeit%20über%20Nacht%20kontinuierlich.

[snd] optimale Zimmerlautstärke, mietrecht.com
https://www.mietrecht.com/zimmerlautstaerke/#:~:text=So%20sollten%20für%20eine%20Zimmerlautstärke,30%20und%2040%20db%20liegen.