

Dokumentation



Ubiquitous Computing

von Patrick Brinkert und Madita Dönni

Einleitung

Bei unserem Projekt handelt es sich um eine Art „Smart Home“ Vorrichtung. Diese Vorrichtung steuert die einzelnen Lichter, die sich im Zimmer oder gegebenenfalls in der Wohnung befinden.

Unser System soll dabei die Problematik eines Lichtschalters lösen. Da ein Lichtschalter immer auf einen bestimmten Ort beschränkt ist, verwendet unser System stattdessen ein akustisches Signal zum ein- und ausschalten der individuellen Leuchten. Des Weiteren birgt ein Lichtschalter hygienische Probleme, die unser System umgeht, indem es auf einer Gestensteuerung statt einer haptischen Bedienmöglichkeit basiert.

Problembeschreibung

Der Weg zum Lichtschalter bleibt durch die Klatschfunktion erspart. Durch 2 hintereinander folgende Töne, die identisch mit dem Klatschmuster sind, werden als Signal wahrgenommen.

Die Gestensteuerung hat den Vorteil, dass die Berührungen unnötig werden und innerhalb einer z.B. Firma keine Bakterien übertragen werden können. Die eigentliche Bedienung zum Ein-/Ausschalten war ein handygesteuertes System, jedoch wäre der Kontakt mit Bakterien nicht vermeidbar.

Überblick des Entwurfes

Die Realisierung fand mittels eines Arduinos statt. Zu den Komponenten gehören 2 Ultraschallsensoren, ein Mikrofon, ein Display und 3 LED Leuchten. Die 2 Ultraschallsensoren und das Mikrofon ermöglichen das Erfassen der Gesten. Der Bildschirm wird zum anzeigen der ausgewählten Lampe genutzt. Die LED Leuchten dienen als Beispiellampen.

Für die Gestensteuerung benutzen wir das Mikrofon und die Ultraschallsensoren. Ein Ultraschallsensor dient zum auswählen einer Lampe, welche auf dem LED-Bildschirm angezeigt wird. Das Ein- und Ausschalten der Lampe ist mittels des Mikrofons möglich.

Die wichtigsten Komponenten

Mikrofon

Bei unserem Mikrofon handelt es sich um das Sensor-Modul ST1146 Iduino. Es verfügt über einen analogen und einen digitalen Output, wobei wir den analogen verwenden. Unsere Implementierung verfügt über eine Methode spezifisch für die Funktion des Mikrofons. In dieser wird zunächst mit dem Befehl `analogRead(sensor)` der Output des Sensor-Moduls ausgelesen. Dieser Read-Befehl gibt einen Integer Wert zurück, der die Lautstärke des jeweiligen Zeitpunktes darstellt. Nun wird mit einer if-Bedingung überprüft ob der gespeicherte Wert ein lautes Geräusch (wie beispielsweise ein Klatschen) darstellt. Falls dies der Fall ist wird die Bedingung betreten. Nun wird mithilfe einer zählenden Integer-Variable überprüft ob das „Klatsch-Muster“ das richtige ist. In unserem Fall zwei Klatscher innerhalb einer Sekunde mindestens 100 Millisekunden getrennt voneinander. Dazu wird zunächst der erste Klatscher wahrgenommen und der Zeitpunkt des Klatschers festgehalten. Wenn bereits ein Klatscher wahrgenommen wurde wird überprüft ob der zweite Klatscher sich innerhalb des vorgegebenen Zeitrahmens befindet. Falls dies nicht der Fall ist werden unsere Variablen zurückgesetzt.

```
if (zustandSensor >= 35 || zustandSensor <= 22)
{
    if (counter == 0)
    {
        startPoint = millis();
        counter++;
    }

    else if (counter > 0 && 900 > millis()-startPoint && millis()-startPoint >= 100)
    {
        counter++;
    }

    else
    {
        counter = 0;
    }
}
```

Anschließend wird das Licht an oder aus geschaltet, falls der Zähler bei 2 ist. Dafür wird überprüft ob das Licht gerade eingeschaltet oder ausgeschaltet ist und das jeweils gegenteilige wird ausgeführt.

```
if (counter ==2)
{ if(zustandLight == false)
  { analogWrite(lampenauswahl, bright);
    zustandLight = true;}

  else if(zustandLight == true)
  { analogWrite(lampenauswahl, LOW);
    zustandLight = false;}

  counter = 0;
}
```

Der Aufruf dieser Methode befindet sich in einem `loop()` .

LED-Bildschirm

Bei unserem LED-Bildschirm handelt es sich um einen weiß-blauen LED-Baustein von Gleichmann Displays (GE-C1602B-TMI-JT/R). In unserem System agiert er als Anzeige der gerade aktiven und angewählten Lampe. In unserer Implementierung wird zunächst der der Bildschirm initialisiert. Hierbei wird die LiquidCrystal Library verwendet.

```
#include <LiquidCrystal.h>
LiquidCrystal anzeige(12, 11, 5, 4, 3, 2);
```

Und

```
anzeige.begin(16,2);
```

Anschließend wird wann immer eine Lampenumstellung in der Methode zur Lampen Wahl wahrgenommen wird das Display geupdated. Hier wird oben links auf dem Display der String der jeweils ausgewählten Lampe mithilfe eines print-Befehls angezeigt.

```
anzeige.setCursor(0,0);
anzeige.print("Lampe 1");
```

Ultraschall Sensoren

Als Sensoren benutzen wir das Ultraschall-Modul HC-SR04. Dieser Sensor verfügt über einen Triggereingang und einen Ausgang für das Messergebnis.

Für ein Rauschfreies Signal wird folgender Code benötigt:

```
digitalWrite(trigger2, LOW);  
  
delay(5);  
  
digitalWrite(trigger2, HIGH);  
  
delay(10);  
  
digitalWrite(trigger2, LOW);
```

Der Pin wird vor dem Auslesen ausgeschaltet.

```
dauer2 = pulseIn(echo2, HIGH);  
entfernung2 = (dauer2/2) * 0.03432;
```

Mit pulseIn wird die Zeit in Mikrosekunden bis der Schall zurückgekehrt ist gezählt. Als Nächstes wird das Ergebnis durch 2 geteilt und mit der Schallgeschwindigkeit multipliziert.

Für die Funktionen Lampenauswahl und Dimmen wurden die beiden Sensoren benötigt.

Die Funktion Lampenauswahl benutzt 3 Ebenen, die in unterschiedlichen Entfernungen definiert sind, um zwischen den Lampen auswählen zu können.

```
if (entfernung2 >= 1 && entfernung2 <= 8)  
  
if (entfernung2 >= 11 && entfernung2 <= 28)  
  
if (entfernung2 >= 21 && entfernung2 <= 38)
```

Für das Dimmen werden 5 Ebenen benutzt, damit 5 unterschiedliche Helligkeitsstufen ausgewählt werden können.

```
if (entfernung1 >= 0 && entfernung1 <= 10)  
  
if (entfernung1 >= 11 && entfernung1 <= 15)  
  
if (entfernung1 >= 16 && entfernung1 <= 21)  
  
if (entfernung1 >= 22 && entfernung1 <= 28)  
  
if (entfernung1 >= 29 && entfernung1 <= 35)
```

Die Helligkeitswerte der Lampen werden in der Schleife am Anfang abgerufen und am Ende gespeichert.

```
Bright = Bright1;  
  
Bright1 = Bright;
```

Durch die Methode analogWrite() wird mithilfe der Parameter der ausgewählten Lampe die gewünschte Helligkeit übergeben.

```
analogWrite(lampenauswahl, bright);
```

Die Gestalt

Die Gestalt dieses Prototyps ist eine Holzkiste, in welcher sich die gesamte Hardware befindet. Die Holzkiste besteht aus 6 Stücken, welche passend zugeschnitten wurden.

Es wurden 5 Bohrungen und 3 Aussägungen benötigt. Die Bohrungen sind für die LED Lampen, das Mikrofon, und die Stromversorgung. Die Aussägungen für die 2 Ultraschallsensoren und das Display.

Die Ultraschallsensoren und die LED Lampen befinden sich im oberen Teil, die Stromversorgung und das Mikrofon an der linken Seite und das Display im rechten unteren Teil der Kiste.

Evaluation

Das System wird mit einem Computer gestartet. Nachdem Start wird durch das Benutzen der Lampenauswahl, die gewünschte Lampe angesteuert und auf dem Display angezeigt.

Als nächstes fungiert das Mikrofon als Schalter, welcher mittels 2 mal Klatschen betätigt werden kann. Das Dimmen der ausgewählten und eingeschalteten Lampe ist mit den zweiten Ultraschallsensor durch das Variieren der Entfernung, mit einer Hand zum Sensor möglich.

Fazit

Alles in allem war das Projekt eine gute Erweiterung und Vertiefung unserer Fähigkeiten im Bereich Computervisualistik und Design. Beispielsweise haben wir gelernt, wie wir Hardware mit unserer Software richtig kombinieren. Und wie man Probleme, die nicht durch Software alleine lösbar sind, dennoch erarbeiten kann.

Wir haben viel über Sensoren und deren Auswertung gelernt. Allerdings haben wir auch die Schattenseiten des Arbeitens mit Hardware kennengelernt. Beispielsweise funktioniert unser Mikrofon-Sensor nicht wie gewünscht oder erwartet, da er keine ausreichende Geräuschkulisse aufnimmt. Dadurch ist unser System nur sehr schwer bedienbar geworden. All das hätte vermieden werden können durch einen hochwertigen, geräuschsensibleren Sensor.

Anhang

Bedienung unseres Systems

Zur Bedienung unseres Systems muss unbedingt eine Verbindung zum Computer bestehen. Dies ist erforderlich, zur optimalen Funktion unseres Systems, da es hin und wieder neu gestartet werden muss. Bei neustarten des Prototypen muss zunächst einmal die Hand über den „Lampen-Wahl“ Sensor gehalten werden um die Auswahl zu starten und damit auch den Bildschirm zu initialisieren. Anschließend kann durch auf und ab bewegen der Hand über dem „Lampen-Wahl“ Sensor durch die Lampen „geswitched“ werden. Die jeweilig ausgewählte Lampe wird dabei auf dem Display angezeigt. Eingestellt auf die gewünschte Lampe kann nun das Mikrofon als Schalter verwendet werden. Hierfür muss zwei Mal kurz und unmittelbar nacheinander in den Mikrofon-Sensor gepustet werden. So kann die ausgewählte Lampe an und auch wieder ausgeschaltet werden. Falls die Lampe nun gedimmt werden soll, kann dies durch den zweiten Ultraschall Sensor geschehen. Dieser ist genauso zu bedienen wie sein Gegenstück zur Lampen Auswahl. Nach dem ursprünglichen auswählen der Lampe können alle diese Aktionen in beliebiger Reihenfolge ausgeführt werden.