Tee-Thermometer

Zielsetzung:

Jeder kennt das Problem: Direkt nach dem Aufbrühen ist der Tee zu heiß zum Trinken, also lässt man ihn auskühlen. Dabei verliert man ihn jedoch schnell aus den Augen, und denkt erst an ihn, wenn er bereits zu kalt ist. Um diesem Problem entgegenzuwirken habe ich mir folgendes Hilfsmittel ausgedacht: Ein Gerät, welches die Temperatur des Tees misst, und die verbleibende Wartezeit berechnet bis eine gewünschte Temperatur erreicht ist. So hat man die verbleibende Zeit immer im Blick, und muss nicht immer die Temperatur erfühlen. Außerdem lässt sich leicht ein Alarm implementieren, falls gewünscht.

Theorie / Funktionsweise:

Es gibt verschiedene Herangehensweisen, um dieses Problem zu lösen. Am einfachsten ist es, den Temperaturverlauf eines typischen Tees zu messen, und daraus die benötigte Zeit zum Abkühlen abzuleiten. Bei der gleichen Anfangstemperatur, Umgebungstemperatur und Tasse dauert es immer gleich lang, bis der Tee die gewünschte Temperatur erreicht hat. Dies könnte man als open-loop Vorgehensweise bezeichnen.

Eine aufwändigere Methode ist es, die Temperatur in regelmäßigen Abständen zu messen, und daraus den Zeitpunkt für das Erreichen einer gewünschten Temperatur zu extrapolieren. Damit ist die Vorhersage nicht von eingegebenen Parametern wie der Umgebungstemperatur oder der Tasse abhängig. Außerdem wird dadurch die veränderte Abkühlzeit durch eine Verringerung der Flüssigkeitsmenge und dem Durchmischen der Flüssigkeit berücksichtigt.

Implementierung:

Da komplexe Berechnungen vorgenommen werden müssen, ist die Verwendung eines Microcontrollers unerlässlich. Hier wurde der Arduino Nano verwendet. Im Nachhinein betrachtet, wäre ein leistungsfähigerer Chip besser geeignet, aber für die Funktion reicht der Nano aus. Als Temperatursensor wurde der DS18B20 verwendet. Dies ist ein digitaler Temperatursensor, welcher mit nur einem Pin verbunden werden muss, und die Temperatur bereits digital übermittelt. Dadurch sind keine weiteren Bauteile notwendig, um die Temperatur auszulesen. Ein weiterer Vorteil ist die hohe Genauigkeit der Temperaturmessung, welche mit +- 0.2 °C angegeben wird.

Als Berechnungsgrundlage für das Auskühlen einer Flüssigkeit gilt eine Exponentialfunktion mit Offset.

$$T(t) = T_0 * \ e^{-k*(t-t')} + T_{Umgebung}$$

Wobei T_0 die Anfangstemperatur, k die Abkühlrate, t' einen Zeitversatz und$T_{Umgebung}$ die Umgebungstemperatur darstellen. Da die Umgebungstemperatur, sowie die Anfangstemperatur nicht aus den Messwerten ersichtlich sind müssen sie als freie Parameter in der Modellfunktion berücksichtigt werden.

Um einen Fit der Temperaturmesswerte an diese Exponentialfunktion durchzuführen, wende ich einen Trick an. Vernachlässigen wir zunächst die Umgebungstemperatur. Angenommen sie ist bekannt, kann sie von den Messwerten subtrahiert werden. Dies vereinfacht die Formel. Jetzt kann man durch logarithmieren der Gleichung einen linearen Zusammenhang zwischen dem Logarithmus der Messwerte und der Abkühlkonstanten k erhalten. Hiermit ist es möglich eine lineare Regression durchzuführen, was sehr wenig Rechenaufwand im Vergleich zu einer nichtlinearen Optimierung

erfordert. Da der Arduino Nano geringe Ressourcen aufweist, ist dies hier notwendig. Die verwendeten Formeln lauten

[Hier Formeln]

Damit können die Parameter T_0 und k bestimmt werden, was die Berechnung des Zeitpunkts des Erreichens der Trinktemperatur ermöglicht.

[Hier Formel]

Da die Umgebungstemperatur nicht bestimmt werden kann, muss die vorhergehende Berechnung für mehrere Werte für $T_{Umgebung}$ erfolgen, und daraus per brute-force Optimierung der am besten passende Wert bestimmt werden.

Fertigstellung:

Um das Gerät bequem nutzen zu können, wurde ein 64x32 OLED Display zur Anzeige der Temperatur und verbleibenden Zeit eingesetzt. Zusätzlich wird die verbleibende Zeit bis der Tee zu kalt ist angezeigt, damit man ihn nicht zu lange stehen lässt. Um einen mobilen Einsatz zu gewährleisten wurde ein Lithium-Ionen-Akku zusammen mit einem Lade- und Schutzschaltung basierend auf dem [Chipnummer] Chip sowie ein Boost-Converter verwendet. Dies ermöglicht ein Laden des Akkus über micro USB. Anschließend habe ich ein passendes Gehäuse entworfen und mit einem 3D-Drucker hergestellt.