

Kurzbeschreibung:

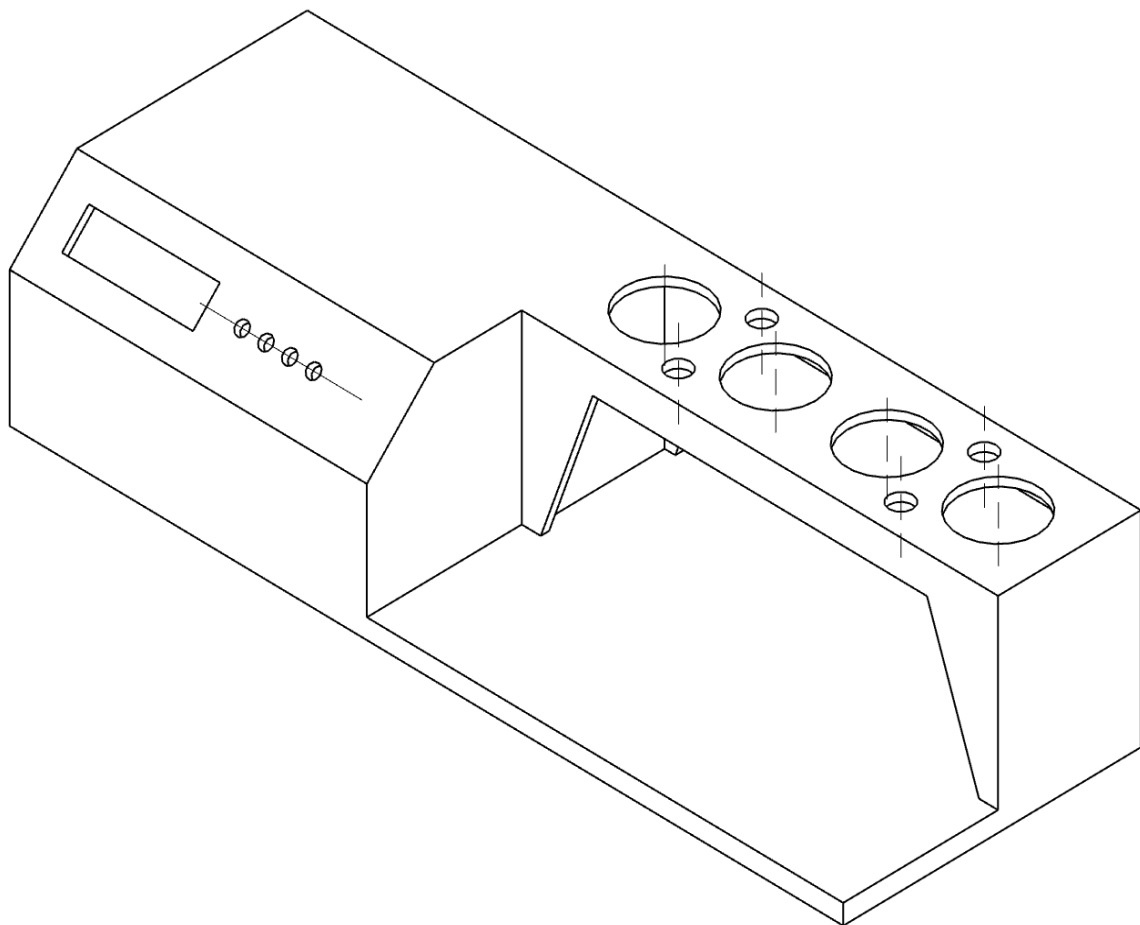
Skript basierender Mischautomat mit zufälliger Rezeptauswahl in Form eines kabellosen Würfels

Projekt Beschreibung:

CubicMixer ist ein Mischautomat basierend auf einer zufälligen Würfeingabe. Der Würfel ist kabellos mit dem Raspberry Pi verbunden und sendet die geworfenen Würfe automatisch. Basierend auf dem empfangenen Würfelresultat wird ein Rezept angemischt. Die Rezepte benutzen eine eigene entwickelte simple Programmiersprache mit einem eigenen Interpreter und einem eigenen Parser. Dies ermöglicht, die Rezepte und Zutaten von dem Python Code loszulösen und die Erweiterung des Mixers für zukünftige Nutzer zu erleichtern. Der Automat kann mit jeder Art von Zutat oder Rezept arbeiten, die mithilfe der zuvor genannten Programmiersprache beschrieben werden können.

Lernziele:

Ziel war es Python Grundlagen zu lernen und kompliziertere Themengebiete wie Objektorientierte Programmierung, Multiprocessing und Ausnutzung von bestimmten Designtricks wie z.B. lambda Funktionen, optionale Funktionsparameter oder einen geteilten Namespace zu meistern. Deswegen wurde überwiegend darauf verzichtet, vorgefertigte Module zu verwenden und eher das "Rad neu zu erfinden", um Übung in der Programmiersprache Python zu erlangen. Zusätzlich wurden grundlegende Erfahrungen in Informationsverarbeitung auf hardwarenaher Ebene gesammelt, sowohl in C als auch in Python.



Erste Entwurfsskizze

Anforderungen:

Der Würfel soll seine momentane Lage mithilfe eines Accelerometers bestimmen können und diese Werte dann über eine TCP Verbindung an den Raspberry Pi schicken. Die Verbindung des Würfels mit einem WLAN Netzwerk soll mithilfe dem Python Codes und einer seriellen Verbindung erfolgen, sodass der Code auf dem Würfel bei Änderung des WLAN Netzwerkes sich nicht verändert. Auf dem Raspberry Pi soll die TCP Verbindung auf einen eigenen Prozess laufen, um die Hauptschleife nicht durch den Verbindungsauf- und abbau zu verlangsamen. Der Empfänger muss ein Schließen der TCP Verbindung erkennen und versuchen einen neuen zu erstellen. Alle empfangenen Werte sollen dann innerhalb desselben Prozesses von den einzelnen Beschleunigungswerten des Accelerometers in eine Würfelseitenzahl umgewandelt werden. Außerdem muss erfasst werden, ob sich der Würfel in den letzten Sekunden bewegt hat.

Der Automat soll in zwei verschiedene Betriebsmodi benutzt werden können: Einmal soll rein nach Rezepten und das andere mal rein zufällig die Zutaten miteinander gemischt werden.

Die Rezeptausswahl soll zugrunde des Betriebsmodus und der Würfel-Eingabe immer eindeutig sein. Somit sollte eine Funktion entwickelt werden, die den Raum der möglichen Würfelwürfe auf den Raum der geladenen Rezepte abbildet.

Um eine Benutzung des Gerätes ohne Monitor und Tastatur zu ermöglichen, soll eine Benutzeroberfläche mit einem LCD Display und vier Eingabeknopfen (Enter, Escape, Left, Right) angebracht werden. Auf dem Display sollen Benachrichtigungen über den aktuellen Status des Geräts gezeigt werden, wie z.B. das aktuelle Würfelresultat oder das gewählte Rezept. Genauso soll aber auch eine Menüführung über diese Benutzereingabe möglich sein, um z.B. den Operationsmodus einzustellen, die Hardware zu testen oder ein bestimmtes Rezept von Hand auszuwählen.

Für die Skripte, die Rezepte und Zutaten beschreiben, soll ein endlicher deterministischer Automat zur Syntaxüberprüfung implementiert werden. Darüber hinaus sollen die Skripte auf semantische Fehler überprüft werden und gegebenenfalls bei Fehlern aussortiert werden. Nach der Übersetzung und Fehlerdetektion der Skripte, sollen diese in einer Datenstruktur abgelegt werden, sodass simple Verlinkungen wie in einer SQL Datenbank erzeugt werden können, um Mehrfachabspeicherung derselben Daten zu vermeiden.

Funktionalitäten, wie der Aufbau der Mischventile oder der Anschluss des Displays, sollen vom Benutzer definiert werden können. Somit soll die Python Funktionalität dieser Subsysteme auch in ein jeweils eigenes Modul ausgelagert werden, um die Anpassung an die dementsprechende Hardware für den Benutzer zu erleichtern. Des Weiteren sollen bestimmende Parameter immer als Variable geschrieben werden, um die Anpassbarkeit möglichst hochzuhalten.

Um die Entwicklung von weiteren Erweiterungen des Projektes, sei es Code-Erweiterungen oder Skript-Erweiterungen, möglichst effizient zu halten, soll das Programm unabhängig vom Betriebssystem ausgeführt werden können. Dies setzt voraus, dass jedmöglicher Code zur Hardware Initialisierung oder Benutzung losgelöst werden kann, um Kompatibilitätsanpassungen automatisch zu tätigen.

Zur weiteren Loslösung vom Betriebssystem sollen Startparameter angegeben werden, ob bestimmte Subsysteme emuliert werden sollen. Zwingend erforderlich hierbei ist die Emulation des LCD Displays, der Eingabeknopfe und des Würfels.