Version

1

Maturaprojekt 2023

Sebastian Haider, Philip Jessner, Simon Frauenschuh  
Betreuer: Dipl. Ing. Karl Heinz Steiner

Ball On Plate

Maturaprojekt 2023

Ball On Plate

© Sebastian Haider, Philip Jessner, Simon Frauenschuh

HTBLuVA Salzburg

Itzlinger Hauptstraße 30

5020 Salzburg

Kontakt: simon\_frauenschuh@icloud.com

I n h a l t s v e r z e i c h n i s

[Einleitung 1](#_Toc83495608)

[Grundansatz 1](#_Toc83495609)

[Messtechniken 1](#_Toc83495610)

[Benutzerinteraktionen 1](#_Toc83495611)

[C – Code (Logik) 3](#_Toc83495612)

[Wahl der Programmiersprache 3](#_Toc83495613)

[Grundkonzept Berechnung 3](#_Toc83495614)

[Web-Applikation (Back- & Frontend) 4](#_Toc83495615)

[Wahl der Softwaresysteme 4](#_Toc83495616)

[Mechanische Konstruktion 5](#_Toc83495617)

[Wahl der Fertigungsmethode 5](#_Toc83495618)

Kapitel

1

# Einleitung

Das Maturaprojekt “Ball On Plate“ lässt eine Kugel vordefinierte Wege über eine Oberfläche rollen und ermöglicht punktgenaue Bewegung des Körpers.

Z

iel des Projekts ist es, eine Kugel, die auf einer ebenen Oberfläche mit bekanntem Reibungskoeffizienten liegt, automatisiert entsprechend der Nutzereingabe zu bewegen, bzw. ihre Bewegungen zu berechnen und eine dementsprechende Bewegung der Oberfläche (im Weiteren meist Platte genannt) einzuleiten, um den vordefinierten Pfad (Kreis, stabilisieren auf einem Punkt, …) zu folgen.

icon key

1. Valuable information
2. Test your knowledge
3. Keyboard exercise
4. Workbook review

HIER EIN FOTO DES PROJEKTS EINFÜGEN!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Grundansatz

Während der Entwicklung wurden verschiedenste Möglichkeiten der Aufenthalts- und Bewegungsberechnung der Kugel genutzt. Generell galt jedoch, die Position der Kugel nicht nur auszulesen und somit der Realität immer „hinterherzuhinken“ (siehe [Grundkonzept\_Berechnung](#Grundkonzept_Berechnung)), sondern zu berechnen und die Sensordaten der Touchfolie nur zur Bestätigung zu nutzen. Gleiches Prinzip wurde bei der Positionsfeststellung der Platte gewählt. Hierbei wird auf den im Servocontroller gespeicherten Wert zugegriffen, und nur bei einer zu großen Abweichung des vom Gyroskop weitergegebenen Wertes Maßnahmen zur Korrektur eingeleitet, bzw. das Programm abgebrochen zur manuellen Fehlerauswertung.

Messtechniken

In diesem Kapitel wird erläutert, welche Messmethoden realisierbar wären; Vor-, sowie Nachteile erörtert und ein detaillierter Einblick in die gewählte Methode gegeben.

Benutzerinteraktionen

Bei dem Projekt wurde vollständig auf physische Komponenten zur Ein- und Ausgabe wie beispielsweise Taster, kabelgebundenes Display, … verzichtet. Hier wurde mit Java EE, PostgreSQL und WildFly auf ein vollkommen Cloud- bzw. Enterprise orientiertes System gesetzt, das ortsungebundene Steuerung sowie zusätzlichen Projektumfang und somit Professionalität gewährt. Zudem wurde so die einfache Erweiterbarkeit gewährleistet, wobei bei steigendem Funktionsumfang keinerlei Änderungen der Hardware vonnöten sind. Da die Steuerung der Kugel automatisiert ist, wird dem Nutzer lediglich ein vordefinierter Funktionsumfang geliefert, aus dessen verschiedenen vorgegebenen Bewegungsabläufen wählen kann. Zudem wird eine 2D Simulation über den Browser ausgegeben, um auch nicht vor Ort anwesenden Benutzern ein Live-Feedback zu ermöglichen.

Kapitel

2

# C – Code (Logik)

Dieser Teil der Dokumentation befasst sich vorwiegend mit der Realisierung der Berechnung und Ausführung der Bewegungen, sowie mit messtechnisch relevanten Problemen und Messverfahren.

D

a die Berechnung der Kugellaufbahn vollständig auf einem Raspberry Pi durchgeführt werden, wurde, wie im Kapitel [Wahl\_der\_Programmiersprache](#Wahl_der_Programmiersprache) näher beschrieben, für alle Berechnung auf das hardwareschonende C gesetzt. Dies umfasst alle Eingaben über den I2C Bus, die Ausgabe über I2C an die Motoren sowie das Hosten eines WildFly Servers für die Web-Applikation (siehe [Web\_Applikation](#Web_Applikation)).

Im weiteren Verlauf dieses Kapitels wird ebenfalls noch ein kurzer Einblick in die Gründe für die Wahl bestimmter Bus-Systeme und Auswertungsmethoden gegeben. Dies dient jedoch nur zur Erläuterung getroffener Entscheidungen aus softwaretechnischer Sicht zu den Messmethoden, die unter [Messtechniken](#Messtechniken) näher erörtert wurden.

Wahl der Programmiersprache

Die ersten Prototypen des Programms wurden der Einfachheit halber mit Python bewerkstelligt. Die Programmstruktur wurde jedoch bald zu komplex und hardwareintensiv, weshalb auf das ressourcenarme C gewechselt wurde, welches es auch ermöglichte, hardwarenäher zu programmieren, was insbesondere der Auswertung der Gyroskopdaten zugutekam, bei denen nun Bitshifting verwendet werden konnte (siehe Funktion „readWord2c“ (lib/gyroscope.h)). Die Verbindung zu den einzelnen Komponenten via I2C konnte ebenso problemlos wie rasch realisiert werden. Der fehlende integrierte Funktionsumfang war für dieses Projekt nicht wirklich von Bedeutung, da der Hauptteil aus Berechnungen besteht, die im Funktionsumfang mitgeliefert sind. Fehlende Funktionen / Bibliotheken wurden aus dem Internet gedownloadet (pca9685.h) oder selbst geschrieben (alle restlichen).

Datenerfassung / Messtechnik

Die Datenerfassung beruht auf der Annahme, dass im Regelfall keine Fehler auftreten. Dies spart während der Datenverarbeitung Zeit, da die Motoren schon beginnen können, sich in die gewünschte Position zu bewegen.

Kapitel

3

Grundkonzept Berechnung

Kugelposition wird vorausberechnet, Motoren werden bewegt, Touchfolie bestätigt Berechnung (wie bei Gyroskop)

Kapitel

3

# Web-Applikation (Back- & Frontend)

Dieser Teil der Dokumentation befasst sich vorwiegend mit der Realisierung der Benutzerinteraktion bzw. Visualisierung über eine Java EE Plattform, sowie mit der Datenbankstruktur und -Realisierung.

t

o save time in the future, print a copy of this document. Click **Print** on the **File** menu, and press ENTER to receive all eight pages of examples and instructions. With the printed document in hand, position yourself in normal view to see the style names next to each paragraph. Scroll through the document, and write the style names next to the paragraphs (press CTRL+HOME to reposition yourself at the beginning of the document).

To create a drop cap for the lead paragraph, like the example above, select the letter T, and then type a new letter.

Wahl der Softwaresysteme

The “icon key” at left was produced by using the Heading 8 style for the words “icon key” and the List Bullet 5 style for the text below—which uses a Wingdings symbol for the bullet character. To change the bullet symbol, click **Bullets and Numbering** on the **Format** menu. Click **Modify**, and then click the **Bullet** button. Select a new symbol, and then click **OK** twice.

# Mechanische Konstruktion

Dieser Teil der Dokumentation befasst sich ausschließlich mit der Realisierung und den mechanischen Problemen der für das Projekt unbedingt notwendigen mechanischen Grundlage.

t

o save time in the future, print a copy of this document. Click **Print** on the **File** menu, and press ENTER to receive all eight pages of examples and instructions. With the printed document in hand, position yourself in normal view to see the style names next to each paragraph. Scroll through the document, and write the style names next to the paragraphs (press CTRL+HOME to reposition yourself at the beginning of the document).

To create a drop cap for the lead paragraph, like the example above, select the letter T, and then type a new letter.

Wahl der Fertigungsmethode

The “icon key” at left was produced by using the Heading 8 style for the words “icon key” and the List Bullet 5 style for the text below—which uses a Wingdings symbol for the bullet character. To change the bullet symbol, click **Bullets and Numbering** on the **Format** menu. Click **Modify**, and then click the **Bullet** button. Select a new symbol, and then click **OK** twice.