Embedded Systems

Calculating Pi

Von **Erik Haubrich**

**Inhaltsverzeichnis:**

[Aufgabenstellung 3](#_Toc149156167)

[1 Algorithmen 4](#_Toc149156168)

[1.1 Leibniz-Reihe 4](#_Toc149156169)

[1.2 Nilakantha-Reihe 4](#_Toc149156170)

[2 Programm 5](#_Toc149156171)

[2.1 Aufbau 5](#_Toc149156172)

[2.2 Tasks 5](#_Toc149156173)

[2.3 Controller-/Interface-Task 5](#_Toc149156174)

[2.4 Buttontask 5](#_Toc149156175)

[2.5 Leibniz-Folge-Task 5](#_Toc149156176)

[2.6 Nilakantha-Folge-Taks 5](#_Toc149156177)

[3 Messbericht Thema – 3 6](#_Toc149156178)

[3.1 Kapitel Aufgabenstellung 6](#_Toc149156179)

[3.2 Kapitel Theorie 6](#_Toc149156180)

[3.3 Kapitel Experimente/Messungen 6](#_Toc149156181)

[3.4 Kapitel Diskussion der Messdaten 6](#_Toc149156182)

[Persönliches Fazit 7](#_Toc149156183)

[Literaturverzeichnis 8](#_Toc149156184)

[Abbildungsverzeichnis 9](#_Toc149156185)

[Anhang 10](#_Toc149156186)

# Aufgabenstellung

Als benotete Übung ist folgende Aufgabenstellung gegeben.

«Es gibt diverse Algorithmen, wie man PI berechnen kann. Einige sind schneller als andere. Ein sehr simpler, jedoch auch langsamer Ansatz ist folgender: Wenn man in einem Quadrat mit der Seitenlänge 1 zwei zufällige Punkte wählt und deren Distanz zur linken unteren Ecke berechnet, dann bekommt man entweder einen Wert über oder unter 1.

Nun besagt der Ansatz, dass das Verhältnis der Punkte im Viertel-Kreis drin im Vergleich zum ganzen Quadrat einem Viertel PI entspricht. Diese Tatsache kann man sich zum Vorteil nehmen und Punkte in diesem Bereich berechnen.

Einen anderen Weg zu PI stellen Reihen dar. Eine einfache Reihe für diesen Zweck ist die Leibniz-Reihe. Diese konvergiert, je weiter man sie berechnet, immer mehr gegen PI/4.»

Die Aufgabe besteht daraus die Leibniz-Reihe in einem Task zu berechnen. Ausserdem soll in einem zweiten Task einen anderen, selbst gewählten Algorithmus zu realisieren. Diese Tasks sollen durch einen Steuertask kontrolliert werden.

1. Der aktuelle Wert von Pi soll stetig auf dem Display des EduBoard angezeigt werden. Das Display muss alle 500ms aktualisiert werden.
2. Die Algorithmen müssen per Tastendruck gestartet werden können und mit einer anderen Taste gestoppt werden und mit einer dritten Taste zurückgesetzt werden. Die vierten Taste soll dazu dienen, zwischen den zwei Algorithmen hin und her zu schalten.

Die Kommunikation zwischen den Tasks kann entweder mit EventBits oder über TaskNotifications stattfinden.

Es müssen mindestens drei Tasks bestehen. Ein Task für das Buttonhandling und die Displayansteuerung, ein Task für die Berechnung mit der Leibniz-Folge und ein Task für den dritten Algorithmus.

Das Programm soll ausserdem mit eine Zeitmess-Funktion ausgestattet werden, um die Zeit zu messen, bis Pi auf fünf Nachkommastellen genau berechnet wurde. Dazu wird xTaskGetTickCount verwendet. Diese Zeit wird dann auf dem Display dargestellt. Die Berechnung von Pi soll aber weitergehen. Die Zeit soll die ganze Zeit mitlaufen.

# Algorithmen

## Leibniz-Reihe

Die Leibniz-Reihe ist eine Formel (Formel 1) zur Annäherung an die [Kreiszahl](https://de.wikipedia.org/wiki/Kreiszahl) Pi, die [Gottfried Wilhelm Leibniz](https://de.wikipedia.org/wiki/Gottfried_Wilhelm_Leibniz) (1646-1716) in den Jahren 1673–1676 entwickelte und 1682 in der Zeitschrift [Acta Eruditorum](https://de.wikipedia.org/wiki/Acta_Eruditorum) erstmals veröffentlichte. Sie lautet (1):

Formel 1: Leibniz-Folge

## Nilakantha-Reihe

Die Nilakantha-Reihe ist eine verbesserte, schnellere Formel (Formel 2)zur Annäherung an die Kreiszahl Pi. Die Formel geht auf den indischen Mathematiker und Astronomen Kelallur Nilakantha Somayaji (1444-1544). Witzigerweise berechnen die aufsummierten Brüche aber genau die Nachkommstellen von Pi, die 3 läuft gewissermaßen vorneweg. Sie kann auf verschieden Weise dargestellt werden (2), (3):

Formel 2: Nilakantha-Folge

Für den C-Code ist der erstere besser geeignet, da das Potenzieren viel Zeit benötigt, wie ich im späteren Verlauf feststellen musste.

Ich habe mich für den Nilakantha Algorithmus entschieden, da er einfach zu programmieren schien aber trotzdem schnell konvergiert.

# Programm

## Aufbau

Die Funktionsweise des Programms ist nach Finite State Machine aufgebaut. Es zeigt immer das Menü an, in dem man sich gerade befindet.

Als erstes wird ein Start-Bildschirm angezeigt, in welchem man die Auswahl zwischen mehreren Untermenüs hat. Die Auswahl erfolgt über die Tasten des EduBoard.

Im ersten Untermenü wird Pi angezeigt, das aus der Library math.h geholt wird. Mit einem weiteren Tastendruck kann man wieder zum Start-Bildschirm zurückkehren.

Im zweiten und dritten Untermenü kommt man in die Leibniz- bzw. Nilakantha-Folge-Menüs.

Diese sind grundsätzlich gleich aufgebaut.

Es wird angezeigt mit welcher taste gestartet, gestoppt, zurückgesetzt und der Algorithmus gewechselt, werden kann. In der ersten Zeile wird die aktuell konvergierte Zahl sowie die bis dahin vergangene Zeit angezeigt und in der zweiten Zeile wie viel Zeit nach dem Starten vergangen ist, bis Pi auf fünf Nachkommastellen genau konvergiert wurde.

## Tasks

Das Ganze Programm ist in vier Tasks aufgeteilt. Ein Task ist für die Ausgabe auf das Display sowie für die Steuerung der Berechnungstasks zuständig. Ein weiterer Task ist nur für das Buttonhandling und die anderen zwei sind je für ein Algorithmus sowie das Timing zuständig.

## Controller-/Interface-Task

## Buttontask

## Leibniz-Folge-Task

## Nilakantha-Folge-Taks

# Messbericht Thema – 3

## Kapitel Aufgabenstellung

## Kapitel Theorie

## Kapitel Experimente/Messungen

## Kapitel Diskussion der Messdaten

# Persönliches Fazit

Die Aufgabenstellung erschien mir anfangs nicht all zu kompliziert. Da ich aber keine Ahnung von Programmieren hatte, speziell nicht von Embedded Systems, erwies sich das aber als Trugschluss.

# Literaturverzeichnis

1. **Wikipeida.org. [Online] [Zitat vom: 21. 10 2023.] https://de.wikipedia.org/wiki/Leibniz-Reihe.**

**2. opengenus.org. [Online] [Zitat vom: 21. 10 2023.] https://iq.opengenus.org/different-ways-to-calculate-pi/.**

**3. 3.141592653589793238462643383279502884197169399375105820974944592.eu. [Online] [Zitat vom: 25. 10 2023.] https://3.141592653589793238462643383279502884197169399375105820974944592.eu/pi-berechnen-formeln-und-algorithmen/.**

# Abbildungsverzeichnis

Text

# Anhang