

Übung: Calculating PI

Algorithmen:

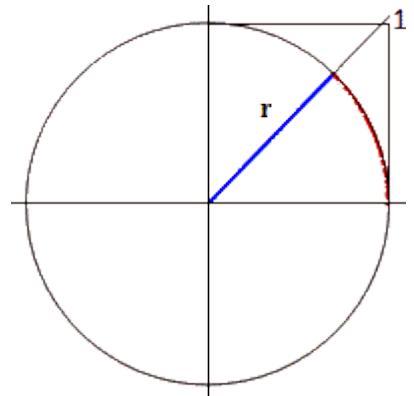
Es gibt diverse Algorithmen, wie man PI berechnen kann. Einige sind schneller als andere.

Ein sehr simpler, jedoch auch langsamer Ansatz ist folgender:

Wenn man in einem Quadrat mit der Seitenlänge 1 einen zufälligen Punkt wählt und dessen Distanz zur linken unteren Ecke berechnet, dann bekommt man entweder einen Wert grösser gleich oder kleiner als 1.

Nun besagt der Ansatz, dass das Verhältnis aller Punkte im Viertel-Kreis drin im Vergleich zu den Punkten im ganzen Quadrat einem Viertel PI entsprechen.

Mit einer genügend grossen Sample-Size kann man hiermit PI 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.

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots = \frac{\pi}{4}$$

Aufgabe:

- Realisiere die Leibniz-Reihen-Berechnung in einem Task.
- Wähle einen weiteren Algorithmus aus dem Internet.
- Realisiere den Algorithmus in einem weiteren Task.
- Schreibe einen Steuertask, der die zwei erstellten Tasks kontrolliert.

Dabei soll folgendes stets gegeben sein:

- Der aktuelle Wert soll stets gezeigt werden. Update alle 500ms
- Der Algorithmus wird mit einem Tastendruck gestartet und mit einem anderen Tastendruck gestoppt.
- Mit einer dritten Taste kann der Algorithmus zurückgesetzt werden.
- Mit der vierten Taste kann der Algorithmus umgestellt werden.
(zwischen Leibniz und dem zweiten Algorithmus)
- Die Kommunikation zwischen den Tasks muss Threadsafe sein. Es dürfen alle Kommunikationsmittel von FreeRTOS verwendet werden.
- Folgende Event-Bits könnte man beispielsweise verwenden:
 - o EventBit zum Starten des Algorithmus
 - o EventBit zum Stoppen des Algorithmus
 - o EventBit zum Zurücksetzen des Algorithmus
 - o EventBit für den Zustand des Kalkulationstask als Mitteilung für den Anzeige-Task
- Mindestens drei Tasks müssen existieren.
 - o Interface-Task für Buttonhandling und Display-Beschreiben
 - o Kalkulations-Task für Berechnung von PI mit Leibniz Reihe
 - o Kalkulations-Task für Berechnung von PI mit anderer Methode
- Erweitere das Programm mit einer Zeitmess-Funktion (verwende xTaskGetTickCount) und messe die Zeit, bis PI auf 5 Stellen hinter dem Komma stimmt. (Zeit auf dem Display mitlaufen lassen und beim Erreichen der Genauigkeit die Zeit berechnen. Die Berechnung von PI soll weitergehen.)
- Als Variablen für die Berechnungen sollten double-precision Datentypen verwendet werden, da sonst schnell Genauigkeit verloren geht.

Dokumentationsarbeiten:

Es soll ein kurzer Bericht zum Projekt verfasst werden. (ca. 8+ Seiten)

Darin enthalten sein soll folgendes:

- Erklärung des zusätzlich gewählten Algorithmus.
- Beschreibung der Tasks. Der Ablauf der Tasks und deren Funktionsweise muss ersichtlich sein. (z.B. Finite State Machine)
- Die verwendeten Kommunikationswege zwischen den Tasks müssen beschrieben werden.
- Die Resultate der Zeitmessung sollen aufgeführt und erklärt werden.
- Vergleiche die Geschwindigkeit von Leibniz gegenüber der anderen Methode
- Wage einen Rückschluss bezüglich der gemessenen Rechenleistung zur tatsächlichen Prozessorleistung zu machen.
- Die Softwareverwaltung GIT ist anzuwenden und online in einem Repository zu sichern.

Die Aufgabe muss bis zum 25.11.2025 erledigt sein und abgegeben werden. (Link auf Repository auf Github/Gitlab reicht. Bericht ebenfalls im Repository)

Die Übung wird benotet.

Gewertet werden folgende Punkte:

- Vollständigkeit gemäss Aufgabenstellung
- Einhalten der Vorgaben gemäss Aufgabenstellung
- Realisierung der zwei Algorithmen
- Realisierung des Steuertasks
- Realisierung des Zeit-Mess-Funktion
- Vollständigkeit der Dokumentation
- Korrekte und sinnvolle Nutzung von Git