

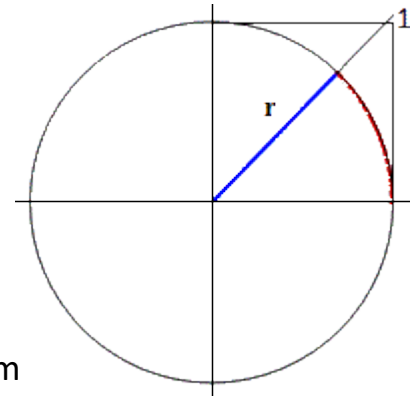
Übung: Calculating PI

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 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.

Ein anderer 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:

Wähle einen Algorithmus (Es darf auch ein anderer sein. Es finden sich diverse im Netz.)

Realisiere den Algorithmus in einem auf FreeRTOS basierenden Programm.

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.
- Die Kommunikation zwischen den Tasks soll mittels EventBits stattfinden.
 - EventBit zum Starten des Algorithmus
 - EventBit zum Stoppen des Algorithmus
 - EventBit zum Zurücksetzen des Algorithmus
 - EventBit für den Zustand des Kalkulationstask als Mitteilung für den Anzeige-Task
- Mindestens zwei Tasks müssen existieren.
 - Interface-Task für Buttonhandling und Display-Beschreiben
 - Kalkulations-Task für Berechnung von PI
- Erweitere das Programm mit einem Zeitmess-Hardware-Timer (wie in der Alarm-Clock Übung) und messe die Zeit, bis PI auf 5 Stellen hinter dem Komma stimmt. (Zeit auf dem Display mitlaufen lassen und bei Erreichen der Genauigkeit den Timer anhalten. Die Berechnung von PI soll weitergehen.)

Dokumentationsarbeiten:

Es soll ein kurzer Bericht zum Projekt verfasst werden. (ca. 2-3 Seiten)

Darin enthalten sein soll folgendes:

- Erklärung des gewählten Algorithmus.
- Beschreibung der Tasks. Der Ablauf der Tasks und deren Funktionsweise muss ersichtlich sein.
- Die EventBits und deren Aufgabe soll erklärt werden.
- Die Resultate der Zeitmessung sollen aufgeführt und erklärt werden.
- Wage einen Rückschluss von der gemessenen Rechenleistung auf die tatsächliche Prozessorleistung zu machen.
- Die Softwareverwaltung GIT ist anzuwenden und online in einem Repository zu sichern.

Die Aufgabe muss bis zum 9.4. erledigt sein und abgegeben werden.

(Link auf Repo reicht. Bericht ebenfalls im Repo)

Folgende Einstellungen müssen in den Projektoptionen vorgenommen werden:

AVR/GNU Linker

➔ *General: «Use vprintf library(-WL,-u,vfprintf)» aktivieren*

➔ *Miscellaneous: Other Linker Flags: «-lprintf_float»*

Damit wird die Unterstützung von Float und Double in sprintf aktiviert.

Die Float-Ausgabe des Display-Treibers reicht nicht um genügend Stellen nach dem Komma auszugeben.