Auslesen und Anzeigen von kontinuierlichen zeitdiskreten Signalen im Hinblick auf Signale mit hohen Frequenzen

Autor: Richard Wieditz

Matrikelnummer: 10251959

Inhalt

1. Vorwort
2. Programmierung
   1. Vorwort
   2. Signalgenerierung
   3. Signalgewinnung
   4. Representation im Speicher
   5. Diagrammisierung
   6. Echtzeitfähigkeit
3. Problematiken
4. Schlusswort

1. Vorwort

Programm in C

1. Ziele
   1. Schnittstelle für Analyseprogramme
   2. Signale in Form von Schwingungen generieren
   3. simulierte Signalgewinnung
   4. effiziente Repräsentation im Speicher für spätere Diagrammisierung und Analysen
   5. Signal anzeigen

im folgenden ‚Wafer‘

2.1. Programmierung - Vorwort

Dependencies

* 1. Mac OS
  2. GCC Compiler

-> C Programm

* 1. (WAF BuildTool

wegen GTK)

* 1. GnuPlot

AquaTerm

* 1. Konzeptionierung möglichst modular
  2. Echtzeitfähigkeit unter Mac OS

Mach RealTimeScheduling Library

2.2. Programmierung - Signalgenerierung

synthesizer.h

Aufgabe

- Cosinusschwingungen beliebiger Frequenz erzeugen

- Generiertes Signal in CSV sichern um es weiterhin zur Signalgewinnung zu nutzen

- Simulation eines Signals mittels CSV Datei

* Funktionalität

- generieren der Schwingung

- konfigurierbar

- Signallänge

- Frequenz

- Phasenverschiebung

- Amplitude

- sichern als CSV

- simulation mittels CSV

- simulieren des Signals mit einstellbarer Aktualisierungsfrequenz

-> Berechenbarkeit

2.3. Programmierung - Representation im Speicher

window.h

- Aufgabe

- Signalrepresentation

- Schnittstelle für *diagram.h, analysor.h, filter.h*

- als Ringbuffer organisierter Speicherbereich

- statisch alloziert

- Beispielsignallänge: 50000 Samples

-> Abdeckung von Schwingungen bis 25 kHz

- Funktionalität

- nur push Funktion

- parallel laufende Start und Endpointer

- Zugriffszeit

-> O(1)

- auslesen in verschiedenen Auflösungen zur Abstraktion des Signals

- Beispielauflösung: 1 / 2000

- Zugriffszeit O(N \* Auflösung)

- Funktionen zur Rückgabe des Signals in Reihenfolge

- optimierte Zugriffszeit

-> einmalig doppelte Fenstergroße

- Fensterteil zur Repräsentation F1

- Fensterteil zur Optimierung F2

- kopieren von Teilsignal von Fensteranfang bis Endpointer

- einfügen des Teilsignals an Start von F2 mittels memcpy

-> Signal liegt in Reihe im Speicher

- Zugriffszeit eventuell O(1)

- insgesamt berechenbar

2.4. Programmierung - Diagrammisierung

diagram.h

- Aufgabe

- Diagrammisierung des Signals in Echtzeit

-> siehe Programmierung - Echtzeitfähigkeit

- verwendet ANSI C Bibliothek GnuPlot

-> mit AquaTerm (seit Mac OS)

* Funktionalität

- anzeigen des Signals mit einstellbarer Aktualisierungsfrequenz

- Zugriffszeit O(n)

- insgesamt berechenbar

2.5. Programmierung - Echtzeitfähigkeit

scheduler.h

-> Mach RealTimeScheduling Library

- Aufgabe

- Definition von Echtzeitfähigen pthreads

- startet die parallel Laufenden Teilprogramme in Echtzeit

3. Problematiken

* Zeitmangel
  + Diagrammaktualisierung nicht parametrisiert, sonder fest im Programmcode
* Simulation

4. Schlusswort

Unterschriften