

Asynchronmaschine Datenerfassung und -auswertung

Das Programm fährt eine ASM hoch, misst Strom und Spannung und gibt den Osanna Kreis der ASM aus. Es gibt eine Einstellung bei dem das Programm die Maschine in Stern-Dreieck hochfährt Der erste Punkt ist in den ersten 100 ms, während die Maschine anläuft Der zweite Punkt ist nachdem die Maschine hochgefahren ist Der dritte Punkt ist während die Maschine hochfährt

Momentanes Programm

- die .h ist einigermaßen annehmbar
- die .cpp funktioniert keineswegs und ist unbrauchbar
- beste Lösung wäre das Projekt neu anzulegen und die readme zu kopieren

Klasse asm:

public:

init:

- liest wichtige Parameter ein (Modus, Frequenz)
 - Modus:
 - 'stardelta' Stern-Dreieck Anlauf
 - 'startup' normale Messung, fährt Maschine nach Messung herunter
- initialisiert Modus und Maschine
 - startet Maschine
 - ruft readData auf

processData:

- ruft splitSine, getPosWave und getParams auf

getResult:

- gibt die Werte in die Konsole aus
- (gibt die Werte in eine Textdatei aus)

private:

outParams (Struktur?)

- beinhaltet Strom und Phasenverschiebung der drei Punkte (U ist egal bei Kreis)

inParams (Liste von impData)

- wird als Pointer verwendet um immer auf den gleichen Speicherplatz zuzugreifen
- falls die Liste zu lang in txt oder so schreiben

readData:

- liest Daten von den Messgeräten ein
 - misst nicht kontinuierlich um Speicher zu verringern (alle 500 ms für 100 ms oder so)
- formatiert Daten als Liste von Strukturen (impData)
- speichert Daten als Liste

impData (struct)

- hat Strom (double), Spannung (double) und Zeitstempel(uint)

splitSine

- unterteilt Sinus in drei Zeitfenster
 - Zeitfenster 1: Ganz am Anfang
 - Zeitfenster 2: Irgendwann mittig (möglicherweise berechnen um genauer Mitte zu finden)
 - Zeitfenster 3: Am Ende wenn die Maschine hochgefahren ist
- mit jedem Zeitfenster wird getPosWave aufgerufen

getPosWave

- Liste von Strukturen als Param
- schaut wann >0
- schaut ob steigt
- bei Strom ~ 0 ist die Welle vorbei
- return max in dem Zeitfenster und deltaT vom Nulldurchgang

getParams

- kriegt Daten von getPosWave
- rechnet Phasenverschiebung (Delta Nulldurchgang) mit f aus
- berechnet Effektivwert aus Peak
- return Effektivwert I , U , Φ