

Der Cache zwischen Backend und Model...

Das Ziel:

Von allen Sensoren bzw. Messkonfig-Eingangskanälen mehr oder weniger gleichzeitig aufgetretene Messwerte finden!

Annahme:

Messwerte können als gleichzeitig betrachtet werden, wenn sie in einem Intervall einer zuvor festgelegten maximalen Länge "period" auftreten.

Idee:

Zeitstrahl wird in Intervalle der Länge period aufgeteilt. Der Perioden-Wert ist groß genug, dass in praktisch allen Fällen von jedem Eingangskanal mindestens ein Messwert (MSet) erfasst wird.

Unsicherheit:

Mit welcher Verzögerung diese MSets im Cache landen, ist unsicher. Es kann jedoch eine Reihenfolgetreue bei der Übertragung angenommen werden.

Das Ziel, "gleichzeitige" Messwerte von allen Eingangskanälen zu finden, kann folgendermaßen erreicht werden:

(Grundvoraussetzung: Alle eingetroffenen MSet-Tripel werden nach Zeitstempel sortiert.)

1. Initiale Synchronisierung mit dem Messwertefluss des Raspberry Pi:

- Es wird in regelmäßigen Abständen (die in etwa der Größenordnung der Periode entsprechen) das früheste Intervall der Länge period gesucht, das MSets von allen In-Channels enthält, und bei dem der Mittelwert von $\min(t)$ und $\max(t)$ der Mitte des Intervalls entspricht.
- Intervallauswertung:
 - Für jeden Messkanal wird der Mittelwert aller vorliegenden Messwerte berechnet.
 - Die Mittelwerte werden in ein Array vom Typ `MSet[]` gepackt und der angemeldeten `IMRunListener`-Implementierung mitgeteilt.
 - Alle Zeitstempel, die im gefundenen Intervall oder davor liegen, werden aus dem Cache gelöscht.

Die obere Grenze des gefundenen Intervalls ist Ausgangspunkt für das Raster aller weiteren gleichlangen Intervalle auf dem Zeitstrahl.

2. Per `TimerTask` wird dann in regelmäßigen Abständen (die z. B. der Hälfte von period entsprechen) die folgenden Routine ausgeführt:

- Sobald von jedem Messkanal der Messkonfig ein Messwert oberhalb des aktuell im Fokus liegenden Intervalls vorliegt:
 - Falls von jedem Messkanal mindestens ein Messwert im Intervall vorliegt:
 - Für jeden Messkanal wird der Mittelwert aller vorliegenden Messwerte berechnet.
 - Die Mittelwerte werden in ein Array vom Typ MSet[] gepackt und der angemeldeten IMRunListener-Implementierung mitgeteilt.
 - Alle Zeitstempel, die im gefundenen Intervall oder davor liegen, werden aus dem Cache gelöscht.
 - Else-Fall:
 - Alle MSets aus dem im Fokus liegenden Intervall werden gelöscht, und der Fokus wird auf das nächste Intervall aus dem Raster gelegt.

(Aufgrund der angenommenen Reihenfolgetreue bei der Messwertübertragung kann davon ausgegangen werden, dass für das im Fokus befindliche Intervall keine weiteren Messwerte mehr eintreffen. Die Lücke im Messdatenstrom wird das Model bzw. der Benutzer an der Gui bemerken.)
- Falls die nach "sobald" genannte Bedingung länger als {timeout} nicht aufgetreten ist, werden alle Intervalle vor (now - [time-out]) aufgetreten sind gelöscht, d. h. alle MSets aus den Intervallen werden gelöscht, und über das IMRunListener wird eine entsprechende Benachrichtigung weitergegeben. Anschließend wird Fokus auf das unterste Zeitintervall verschoben, für das noch kein Timeout aufgetreten ist.