Αναφορά 1ης Εργασίας Ενσωματωμένων Συστημάτων

Άρης Ελευθέριος Παπαγγέλης – ΑΕΜ: 8883

Το πρόγραμμα υλοποιήθηκε με τέσσερις συναρτήσεις:

- main
- samplingWithoutTimeStamps
- samplingWithTimestamps
- handleAlarm

Ενώ χρειάστηκε και μία global μεταβλητή flag, η οποία αρχικά ισούται με 1.

Main

Αρχικά, θέτουμε τα t=7200 και dt=0.1 με βάση την εκφώνηση, και καλούμε διαδοχικά τις δύο συναρτήσεις που λαμβάνουν τα timestamps, με ορίσματα τα t και dt. Πριν την κλήση των συναρτήσεων βέβαια, έχουμε ορίσει με την εντολή signal την συνάρτηση handleAlarm ως τον χειριστή του σήματος SIGALRM, σε περίπτωση που καταφθάσει τέτοιο σήμα (θα καταστεί σαφές παρακάτω).

HandleAlarm

Ο handler του σήματος SIGALRM, το μόνο που κάνει είναι να θέσει την καθολική μεταβλητή flag ίση με 0.

SamplingWithoutTimeStamps

Ορίζουμε το μέγιστο μέγεθος του πίνακα δειγματοληψίας, και δεσμεύουμε και άλλες 10 επιπλέον θέσεις για σιγουριά. Ύστερα, ορίζουμε τα structs στα οποία θα αποθηκευτούν το κάθε timestamp και ο χρόνος που η διεργασία κοιμάται.
Ύστερα, η λογική έχει ως εξής: Ορίζουμε ενα alarm για χρόνο t (=7200), και στη συνέχεια δειγματοληπτούμε όσο το flag συνεχίζει να είναι 1. Μετά από την λήψη κάθε δείγματος, η διεργασία κοιμάται για χρόνο dt (=0.1). Αυτή η διαδικασία συνεχίζεται, μέχρι να παρέλθει το διάστημα t. Όταν αυτό συμβεί, καλείται η handleAlarm σαν interrupt, και θέτει το καθολικό flag ίσο με 0. Επομένως, το loop δειγματοληψίας σταματάει πια να εκτελείται, εφόσον το flag έγινε 0.
Τέλος, αποθηκεύουμε τα timestamps σε ένα αρχείο, με τίτλο “withoutTimestamps.txt”, για περαιτέρω επεξεργασία σε MATLAB και εξαγωγή διαγραμμάτων και στατιστικών.

SamplingWithTimeStamps

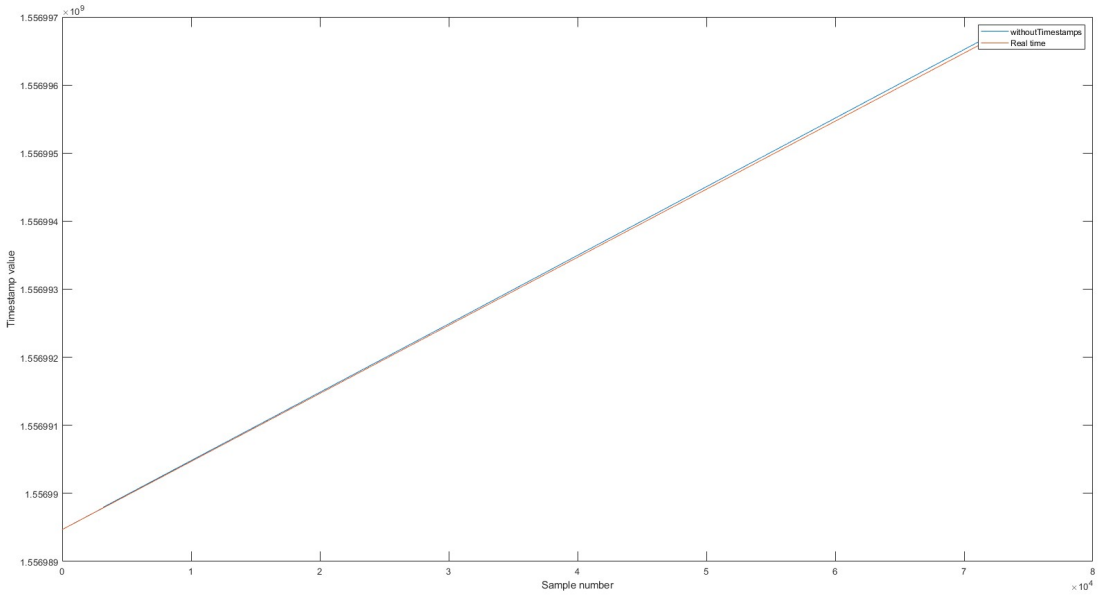
Η λογική είναι ακριβώς η ίδια με την προηγούμενη συνάρτηση, η μόνη διαφορά είναι ότι, αυτή τη φορά η διεργασία δεν κοιμάται για σταθερό χρόνο, αλλά υπολογίζει πόσο λιγότερο από 0.1s πρέπει να κοιμηθεί με βάση τα προηγούμενα timestamps, ώστε να έχουμε ορθή απόσταση ανάμεσα στα δείγματα, ακόμα και για μεγάλους χρόνους δειγματοληψίας. Δηλαδή, προσπαθεί να καταπολεμήσει το drift των τιμών.
Ουσιαστικά, προστίθεται η εξής γραμμή στο loop δειγματοληψίας:

sleepTime.tv_nsec=(long int)(1000000000*dt-(timeStamps[i]-timeStamps[0]-0.1*i)*1000000000);

Ύστερα, τα αποτελέσματα αποθηκεύονται σε αρχείο με τίτλο “withTimestamps.txt”

Διαγράμματα και στατιστικές μετρήσεις

Υλοποίηση Α – Χωρίς χρήση των timestamps

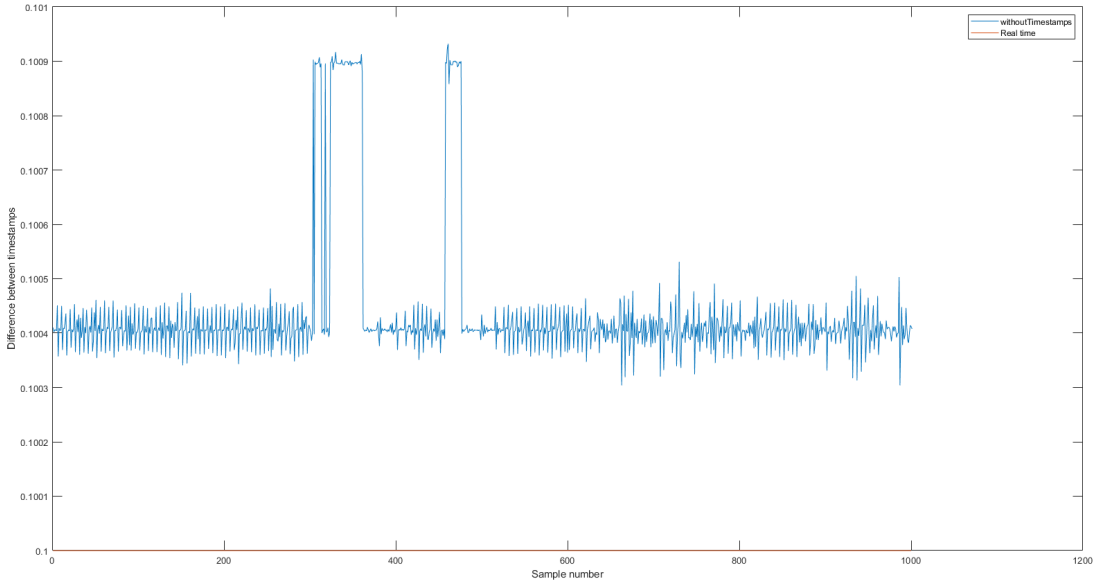


Παρατηρούμε πως τα timestamps έχουν μια αρκετά καλή ακρίβεια σε σχέση με τον πραγματικό χρόνο στην αρχή, η οποία χαλάει όμως όσο περισσότερο διαρκεί η δειγματοληψία. Επίσης, ο αριθμός των δειγμάτων είναι 71542, αντί για 72000 που θα έπρεπε να ήταν κανονικά.
Παραθέτω και τις στατιστικές μετρήσεις για τις αποστάσεις ανάμεσα στα δείγματα:

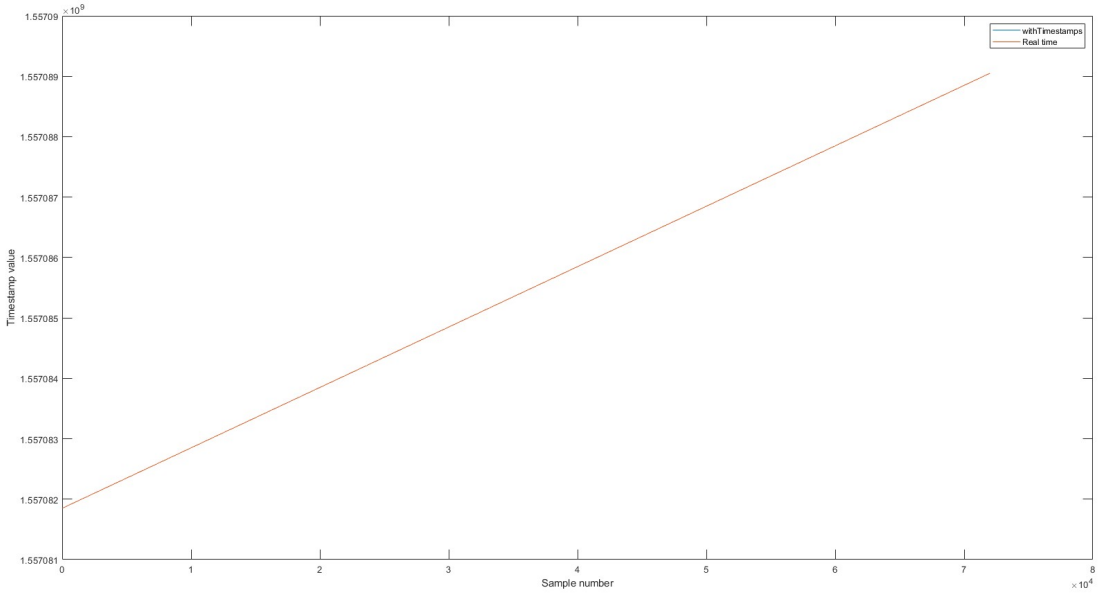
A_max	7.9017
A_mean	0.1007
A_median	0.1004
A_min	0.1000
A_std	0.0292

Παρατηρούμε πως σε κάποιο δείγμα υπήρξε πρόβλημα, εξού και η υπερβολικά μεγάλη τιμή για το max. Κατά τα άλλα, οι αποστάσεις είναι αρκετά κοντά στο 0.1, που είναι το ζητούμενο.

Τέλος,παρατίθεται και ένα στιγμιότυπο των αποστάσεων, σε σχέση με το 0.1 που είναι το ζητούμενο:



Υλοποίηση Β – Με χρήση των timestamps

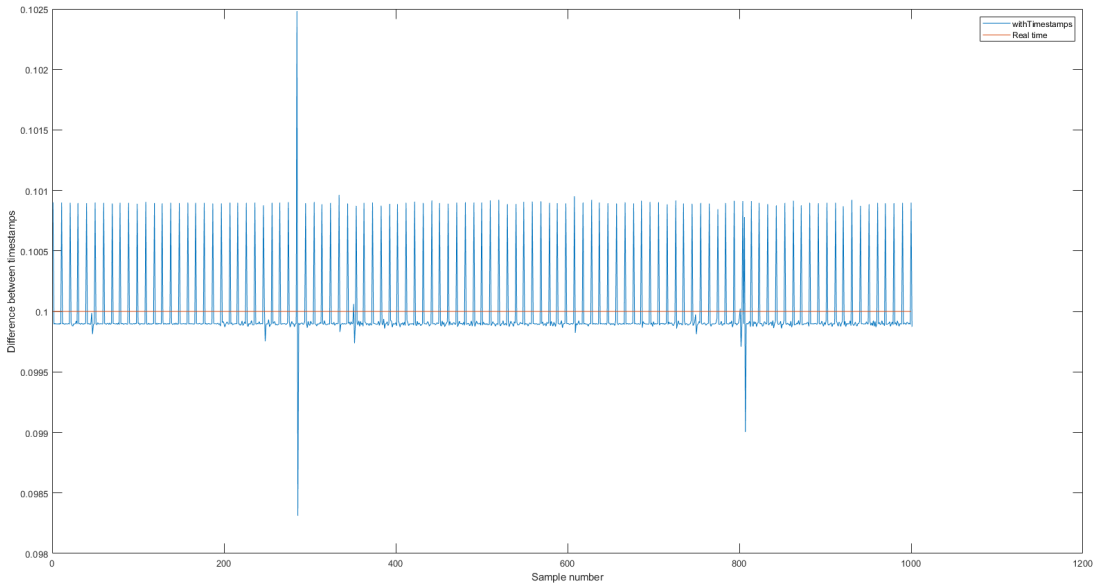


Παρατηρούμε πως τα timestamps έχουν άριστη ακρίβεια σε σχέση με τον πραγματικό χρόνο, αφού τα γραφήματα του πραγματικού χρόνου και της υλοποίησης withTimestamps ταυτίζονται. Επίσης, ο αριθμός των δειγμάτων είναι 72000, ακριβώς όσα θα έπρεπε να είναι κανονικά.
Παραθέτω και τις στατιστικές μετρήσεις για τις αποστάσεις ανάμεσα στα δείγματα:

B_max	0.1025
B_mean	0.1000
B_median	0.0999
B_min	0.0976
B_std	3.0296e-04

Παρατηρούμε πως τα αντίστοιχα στατιστικά είναι πολύ καλύτερα για αυτή την υλοποίηση, αφού το mean είναι ακριβώς στο 0.1, ενώ και το min και max δεν απέχουν πολύ από την πρότυπη τιμή. Επιπλέον, και η τυπική απόκλιση ειναι αρκετά μικρή.

Τέλος,παρατίθεται και ένα στιγμιότυπο των αποστάσεων, σε σχέση με το 0.1 που είναι το ζητούμενο:



Βλέπουμε πως η απόσταση μεταξύ των δειγμάτων σε αυτή την περίπτωση είναι πολύ καλύτερη σε σχέση με την πρώτη υλοποίηση.

Σχόλια και παρατηρήσεις

Η δεύτερη υλοποίηση, ουσιαστικά έχει έναν μηχανισμό ανάδρασης (feedback) και για αυτό καταφέρνει να επιτύχει καλύτερα αποτελέσματα.

Σε κάθε περίπτωση, μπορούμε να παρατηρήσουμε από τα στατιστικά που βγάλαμε πως υπάρχουν διακυμάνσεις απο δείγμα σε δείγμα, και στις δύο υλοποιήσεις. Αυτό συμβαίνει επειδή το πρόγραμμα μας έτρεξε σε πολυδιεργασιακό λειτουργικό σύστημα (Ubuntu) και όχι σε κάποιο λειτουργικό σύστημα πραγματικού χρόνου. Σε ένα λειτουργικό σύστημα πραγματικού χρόνου, η εκτέλεση ενός προγράμματος θα πρέπει να είναι προβλέψιμη και επαναλήψιμη. Αυτό δυστυχώς δεν είναι εφικτό στα λειτουργικά συστήματα γενικής χρήσης που χρησιμοποιούνται στους προσωπικούς υπολογιστές.

Όλα τα στατιστικά και τα γραφήματα παρήχθησαν μέσω επεξεργασίας των αρχείων “withTimestamps.txt” και “withoutTimestamps.txt” με MATLAB.

Ο κώδικας της εργασίας, το MATLAB script και τα αρχεία .txt βρίσκονται στο ακόλουθο repository του Github:
https://github.com/ArisPapangelis/Real-Time-Embedded-Systems-ECE