

TUC

ΠΛΗ501 – εργασία #1

Εργασία στο TinyOS

Χατζηδήμος Νικόλαος - 2020030188

Κρητικάκης Μάριος - 2020030213

1/12/2024

Περιεχόμενα

Σύγκριση με τις Απαιτήσεις TAG:.....	3
Αρχιτεκτονική module	3
Περιγραφή.....	3
Tasks.....	4
Calculate Data	4
Αποστολή Δεδομένων.....	5
sendAvgDataTask() & SendMaxDataTask()	5
Events.....	6
Boot.booted().....	6
RadioControl.startDone().....	6
RoutingMsgTimer.fired().....	6
EpochTimer.fired()	6
LostTaskTimer.fired()	6
Παρατηρήσεις	6
Κατανομή εργασίας	6
Αποτελέσματα	7
Τοπολογία Δέντρου	7
2 Εποχές από MAX	11
Τυχαία επιλογή εντολής κατά την εκκίνηση.....	15
Μέγεθος παραθύρων και σταθερών	16
Stop μετά από 15 δευτερόλεπτα	17
Καινούργιες βοηθητικές συναρτήσεις.....	18

Διαφορές

1. Αφαίρεση μη χρησιμοποιούμενων διεπαφών
 - Leds, Led0Timer, Led1Timer, Led2Timer
 - NotifyAMSend, NotifyAMPacket, NotifyPacket
 - NotifyReceive
 - NotifySendQueue, NotifyReceiveQueue
 - SerialControl, SerialAMSend, SerialAMPacket, SerialPacket, SerialReceive
2. Προσθήκη νέων διεπαφών
 - DataMaxPacket, DataMaxAMSend, DataMaxAMPacket, DataMaxSendQueue, DataMaxReceiveQueue
 - DataAvgPacket, DataAvgAMPacket, DataAvgAMSend, DataAvgReceive, DataAvgSendQueue, DataAvgReceiveQueue
 - RandomGenerator, GeneratorSeed
3. Αλλαγές στις μεταβλητές
 - Προστέθηκαν νέες μεταβλητές και αφαιρέθηκαν κάποιες παλιές:
 - Προστέθηκαν οι μεταβλητές radioMessageMaxSendPkt, radioMessageAvgSendPkt, serialPkt, serialRecPkt
 - Αφαιρέθηκαν οι μεταβλητές roundCounter
4. Αλλαγές στις συναρτήσεις
 - Προστέθηκαν νέες συναρτήσεις και αφαιρέθηκαν κάποιες παλιές:
 - Προστέθηκαν οι συναρτήσεις generateRandomValue, initializeSensorValue, updateSensorValue, rootResults
 - Αφαιρέθηκαν οι συναρτήσεις που σχετίζονται με τις διεπαφές που αφαιρέθηκαν
5. Αλλαγές στα γεγονότα (events)
 - Προστέθηκαν τα γεγονότα DataMaxAMSend.sendDone, DataAvgAMSend.sendDone, RoutingAMSend.sendDone, DataMaxReceive.receive, DataAvgReceive.receive
 - Αφαιρέθηκαν τα γεγονότα που σχετίζονται με τις διεπαφές που αφαιρέθηκαν
6. Αλλαγές στις εργασίες (tasks)
 - Προστέθηκαν νέες εργασίες και αφαιρέθηκαν κάποιες παλιές:
 - Προστέθηκαν οι εργασίες sendMaxDataTask, sendAvgDataTask
 - Αφαιρέθηκαν οι εργασίες που σχετίζονται με τις διεπαφές που αφαιρέθηκαν
7. Αλλαγές στη λογική του προγράμματος
 - Έγιναν αλλαγές στη λογική του προγράμματος για να υποστηριχθούν οι νέες λειτουργίες:
 - Προστέθηκε η λογική για την αποστολή και λήψη μηνυμάτων DataMax και DataAvg
 - Προστέθηκε η λογική για την ενημέρωση των τιμών αισθητήρων και τον υπολογισμό των αποτελεσμάτων

Συνοπτικά

- Οι αλλαγές που έγιναν για να μεταβούμε από το SRTreeC_original.nc στο SRTreeC.nc περιλαμβάνουν την αφαίρεση μη χρησιμοποιούμενων διεπαφών, την προσθήκη νέων διεπαφών, την προσθήκη νέων μεταβλητών και συναρτήσεων, την προσθήκη νέων γεγονότων και εργασιών, και την αλλαγή της λογικής του προγράμματος για να υποστηριχθούν οι νέες λειτουργίες.

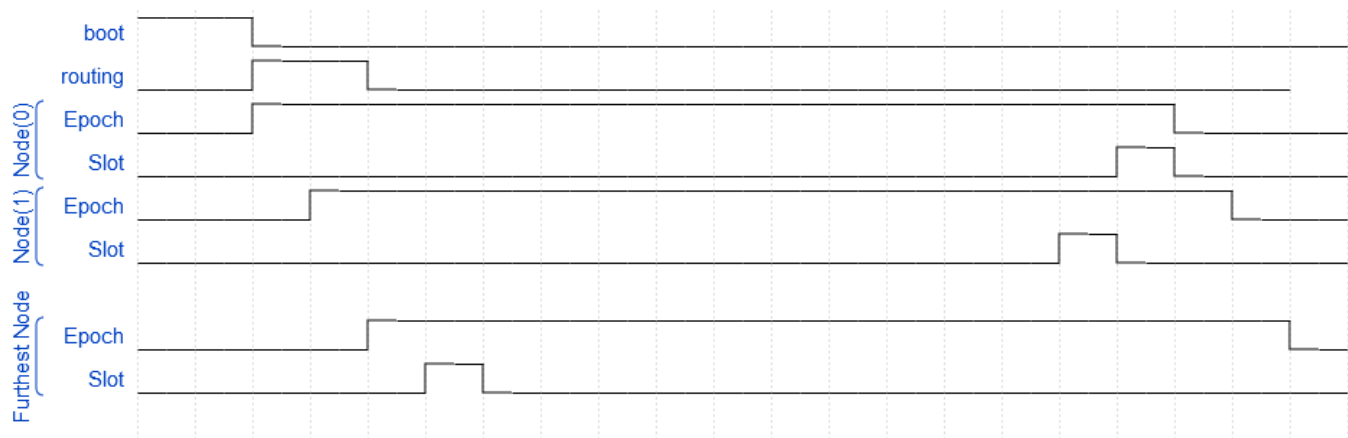
Σύγκριση με τις Απαιτήσεις TAG:

Απαίτηση TAG	Υλοποίηση στο Κώδικα	Συνάφεια
Συγκέντρωση Δεδομένων	DataMax, DataAvg, ουρές αποστολής/λήψης	✓ Υποστηρίζεται
Δομή Δέντρου Δρομολόγησης	Μεταβλητές parentNode, depth	✓ Συμβατό
Χρονοδρομολόγηση	Χρονομετρητής EpochTimer	✓ Συνάδει
Διαχείριση Ενέργειας/Απώλειας	Ουρές αποστολής και λήψης	✓ Υλοποιημένη
Μηχανισμοί Δρομολόγησης	Αποστολή και λήψη μέσω DataMaxAMSend και Receive	✓ Σχετικό

Αρχιτεκτονική module

Timers :

- RoutingMsgTimer : Τρέχει το routing
- SlotTimer : Υπεύθυνο για το παράθυρο δειγματοληψίας, αποστολής και παραλαβής μηνυμάτων
- EpochTimer : Υπεύθυνο για τον διατήρηση της εποχής



Περιγραφή

Κάθε αρχή εποχής διαφέρει ελάχιστα από κόμβο σε κόμβο αναλόγως με το βάθος του κόμβου λόγο του χρόνου μετάδοσης των μηνυμάτων.

Αφού οι κόμβοι λάβουν το μήνυμα δρομολόγησης, και άρα το βάθος τους και τον πατέρα τους, θέτουν το παράθυρο δειγματοληψίας τους ανάλογα με το βάθος τους σύμφωνα με την παρακάτω συνάρτηση.

$$WindowStart = (EpochWindow - StartOffset) - CommandTimeConstant \cdot (Depth)$$

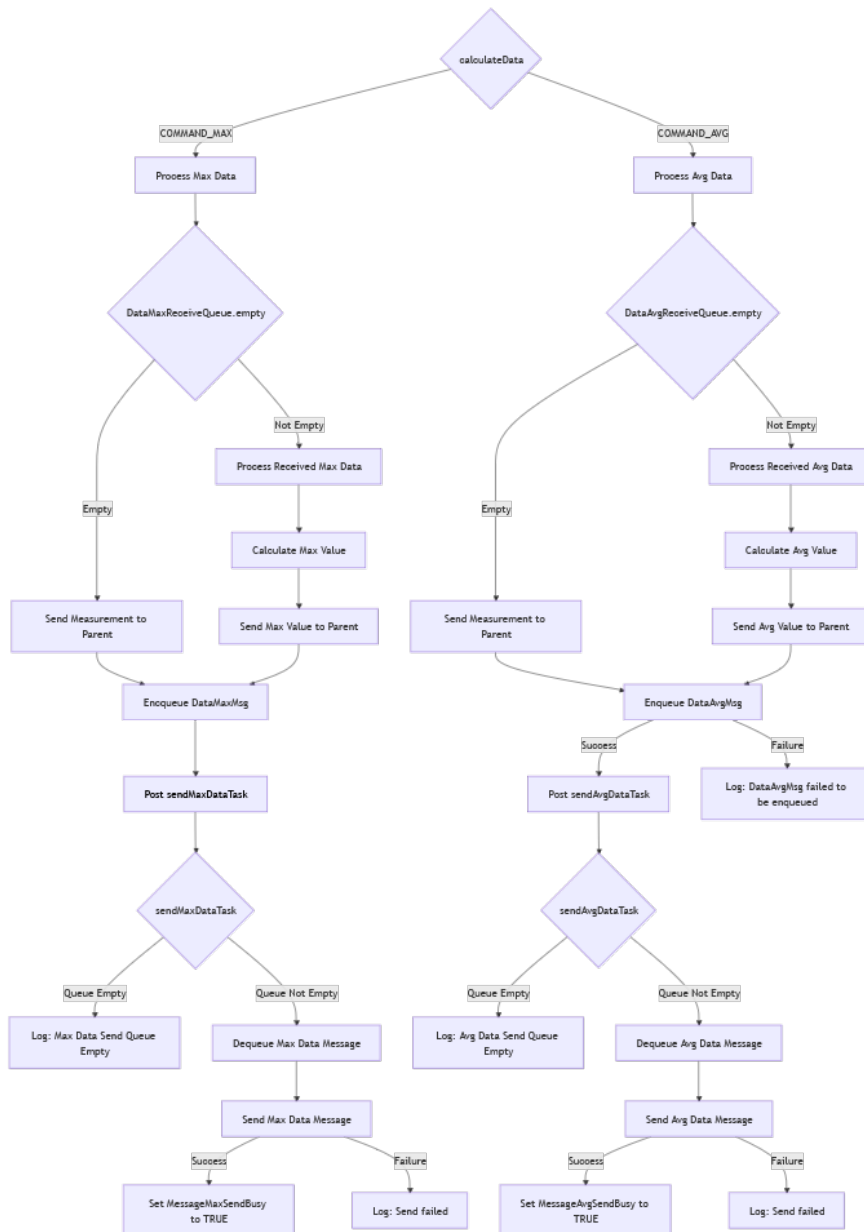
Οι κόμβοι που βρίσκονται σε χαμηλότερο βάθος ξεκινούν το παράθυρο δειγματοληψίας του πρώτοι, στέλνοντας τα δεδομένα τους στον πατέρα τους, όπως προδιαγράφετε στον TAG.

Queues

- RoutingQueue : Δεν τροποποιήθηκε.
- DataMaxQueue : Δρομολογούνται τα μηνύματα της εντολής MAX.
- DataAvgQueue : Δρομολογούνται τα μηνύματα της εντολής AVG.

Tasks

Calculate Data



Η συνάρτηση CalculateData() καλείτε στην αρχή του παραθύρου μέτρησης. Αφού κληθεί, καλεί την updateSensorValue όπου και λαμβάνει την καινούργια μέτρηση. Αν μέχρι τότε έχει λάβει δεδομένα από κάποιο παιδί, τα επεξεργάζεται αναλόγως την εντολή, και τα στέλνει στον πατέρα. Αν δεν έχει λάβει μήνυμα από κάποιο παιδί, στέλνει αποκλειστικά την μέτρησή του.

Τα μηνύματα που στέλνουμε είναι τα εξής :

```
typedef nx_struct DataAvgMsg{
```

```
    nx_uint16_t Sum;
```

```
    nx_uint8_t Count;
```

```
} DataAvgMsg;
```

```
typedef nx_struct DataMaxMsg{
```

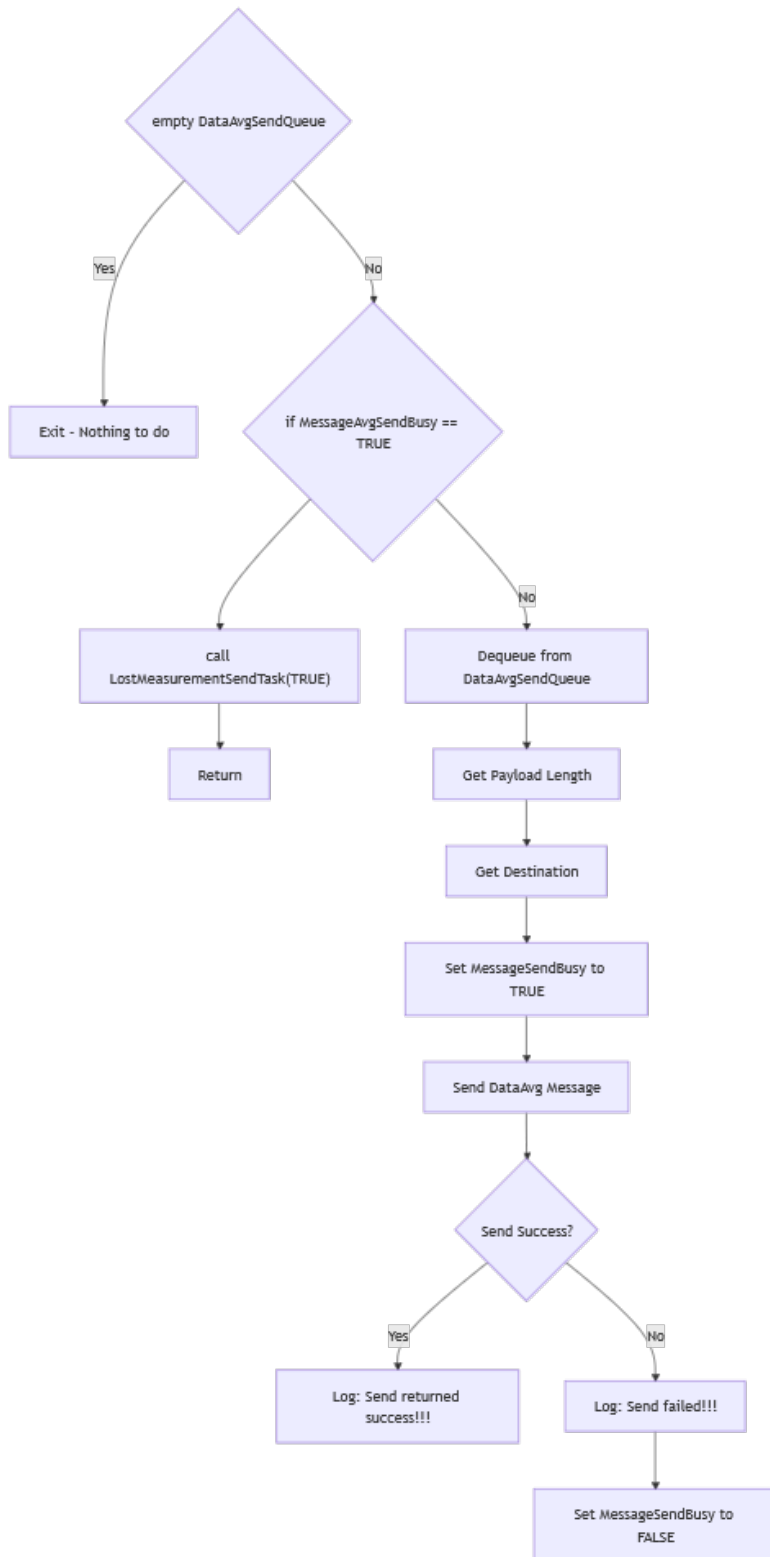
```
    nx_uint8_t data;
```

```
} DataMaxMsg;
```

1. Εφόσον οι τιμές μέτρησης περιορίζονται στο διάστημα [1..50], 6-bit ($2^6 = 64$) αρκούν για να στείλουμε τα δεδομένα.

2. Ο μέγιστος αριθμός κόμβων στο δέντρο μπορεί να είναι μονάχα 255 (uint8_t max).

Αποστολή Δεδομένων



`sendAvgDataTask()` & `SendMaxDataTask()`

Η αποστολή δεδομένων γίνεται με δύο Tasks -- `sendAvgDataTask` & `sendMaxDataTask` -- τα οποία είναι απολύτως όμοια. Για αυτόν τον λόγο αναφέρουμε μονάχα την μία από τις δύο, την `sendAvgDataTask`.

Πρώτα τα sanity checks, όπου ελέγχουμε εάν το queue είναι άδειο, όπου και επιστρέφουμε αφού δεν έχουμε κάτι να στείλουμε. Και ελέγχουμε επίσης εάν η γραμμή είναι κατειλημμένη, στην οποία περίπτωση κρατάμε τα δεδομένα για να τα στείλουμε αργότερα.

Αφού περάσαμε τα sanity checks, εξάγουμε το μήνυμα από το queue και το στέλνουμε με το `DataAvgAMSend`.

Events

Boot.booted()

Κατά την εκκίνηση ξεκινάμε τον [ασύρματο πομποδέκτη](#), αρχικοποιούμε τις μεταβλητές του κόμβου αναλόγως με το ID τους και τέλος αρχικοποιούμε την μέτρηση του αισθητήρα για να σύμφωνα με τις απαιτήσεις της εργασίας.

RadioControl.startDone()

Αν βρισκόμαστε στον κόμβο ρίζας, ξεκινάμε τον RoutingMsgTimer κάνοντας oneshot μερικά millis αργότερα.

RoutingMsgTimer.fired()

Έχει μείνει σχετικά απaráλλακτο από το αρχικό. Η μόνη κύρια διαφορά βρίσκεται στο ότι το Routing τρέχει μονάχα μία φορά.

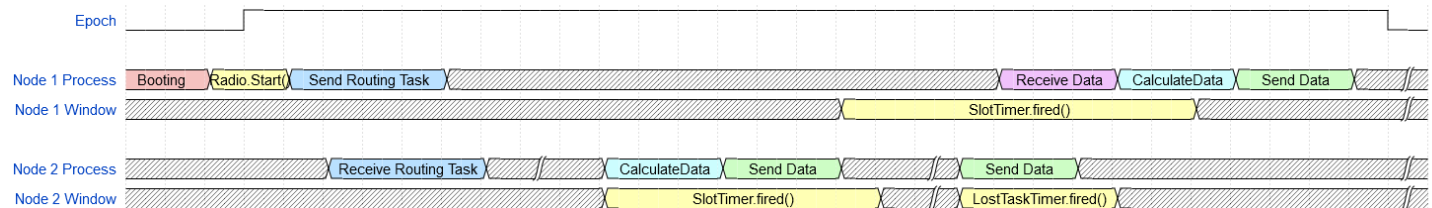
EpochTimer.fired()

Στην αρχή της κάθε εποχής καλούμε την [CalculateData\(\)](#).

LostTaskTimer.fired()

Έγινε μια προσπάθεια να ανακτηθούν και να ανασταθούν τα χαμένα μηνύματα όμως είχαμε μονάχα μερική επιτυχία.

Ι Παράδειγμα χρονισμών event με χαμένο μήνυμα.



Παρατηρήσεις

Κατά την προσομοίωση χανόντουσαν ένα σημαντικό κομμάτι μηνυμάτων και ακόμη και με εκτενής δοκιμές παραθύρων και χρονισμών δεν καταφέραμε να ανακτήσουμε τα δεδομένα.

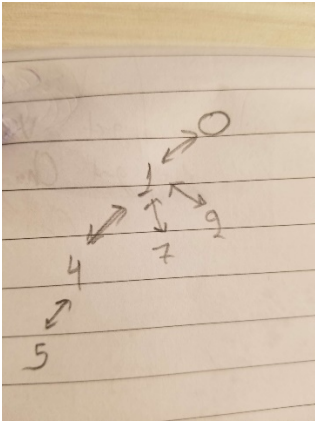
Παρατηρήθηκαν επίσης ίδια αποτελέσματα μεταξύ δύο η περισσότερων προσομοιώσεων.

Τέλος, παρατηρήθηκαν σποραδικά σφάλματα όπου κάποιος κόμβος δεν εκκινούσε ή εκκινούσε κάποιος που βρισκόταν εκτός εμβέλειας. Κάποια σφάλματα από αυτά βρέθηκαν να οφείλονται στους χρονισμούς των Timers και διορθώθηκαν. Άλλα ήταν εξαιρετικά σπάνια για να γίνει αποσφαλμάτωση.

Κατανομή εργασίας

Εργαστήκαμε και οι δύο παράλληλα σε κάθε κομμάτι.

Τοπολογία Δέντρου



2 Εποχές από AVG

TicksPerSecond : 10000000000

```

0 1 -50.0
1 0 -50.0
1 4 -50.0
4 1 -50.0
1 7 -50.0
7 1 -50.0
1 2 -50.0
2 1 -50.0
4 5 -50.0
5 4 -50.0
  
```

```

0:0:10.000000000 DEBUG (0): Booted
0:0:10.000000000 DEBUG (0): curdepth = 0 , parentID= 0
0:0:10.000000000 DEBUG (0): Command selected is AVG
0:0:10.000000000 DEBUG (1): Booted
0:0:10.000000000 DEBUG (1): curdepth = 255 , parentID= 65535
0:0:10.000000000 DEBUG (2): Booted
0:0:10.000000000 DEBUG (2): curdepth = 255 , parentID= 65535
0:0:10.000000000 DEBUG (3): Booted
0:0:10.000000000 DEBUG (3): curdepth = 255 , parentID= 65535
0:0:10.000000000 DEBUG (4): Booted
0:0:10.000000000 DEBUG (4): curdepth = 255 , parentID= 65535
0:0:10.000000000 DEBUG (5): Booted
0:0:10.000000000 DEBUG (5): curdepth = 255 , parentID= 65535
0:0:10.000000000 DEBUG (6): Booted
0:0:10.000000000 DEBUG (6): curdepth = 255 , parentID= 65535
0:0:10.000000000 DEBUG (7): Booted
0:0:10.000000000 DEBUG (7): curdepth = 255 , parentID= 65535
0:0:10.000000000 DEBUG (8): Booted
0:0:10.000000000 DEBUG (8): curdepth = 255 , parentID= 65535
  
```


0:0:10.000000000 DEBUG (9): Booted
0:0:10.000000000 DEBUG (9): curdepth = 255 , parentID= 65535
0:0:10.000000010 DEBUG (0): Radio initialized successfully!!!
0:0:10.000000010 DEBUG (1): Radio initialized successfully!!!
0:0:10.000000010 DEBUG (2): Radio initialized successfully!!!
0:0:10.000000010 DEBUG (3): Radio initialized successfully!!!
0:0:10.000000010 DEBUG (4): Radio initialized successfully!!!
0:0:10.000000010 DEBUG (5): Radio initialized successfully!!!
0:0:10.000000010 DEBUG (6): Radio initialized successfully!!!
0:0:10.000000010 DEBUG (7): Radio initialized successfully!!!
0:0:10.000000010 DEBUG (8): Radio initialized successfully!!!
0:0:10.000000010 DEBUG (9): Radio initialized successfully!!!
0:0:10.488281260 DEBUG (0): RoutingMsgTimer fired! radioBusy = False
0:0:10.488281260 DEBUG (0):

0:0:10.488281260 DEBUG (0): ##### ROUTING #####
0:0:10.488281260 DEBUG (0): #####
0:0:10.488281260 DEBUG (0): Sending RoutingMsg...
0:0:10.488281260 DEBUG (0): RoutingMsg enqueued successfully in SendingQueue!!!
0:0:10.488281290 DEBUG (0): startEpoch()
0:0:10.488281290 DEBUG (0): startEpoch(): Started Epoch Timer. Offset Milli = 40000
0:0:10.510742197 DEBUG (1): RoutingMsgTimer fired! radioBusy = False
0:0:10.510742197 DEBUG (1): Sending RoutingMsg...
0:0:10.510742197 DEBUG (1): RoutingMsg enqueued successfully in SendingQueue!!!
0:0:10.510742227 DEBUG (1): startEpoch()
0:0:10.510742227 DEBUG (1): startEpoch(): Started Epoch Timer. Offset Milli = 39820
0:0:10.533203135 DEBUG (2): RoutingMsgTimer fired! radioBusy = False
0:0:10.533203135 DEBUG (2): Sending RoutingMsg...
0:0:10.533203135 DEBUG (2): RoutingMsg enqueued successfully in SendingQueue!!!
0:0:10.533203135 DEBUG (4): RoutingMsgTimer fired! radioBusy = False
0:0:10.533203135 DEBUG (4): Sending RoutingMsg...
0:0:10.533203135 DEBUG (4): RoutingMsg enqueued successfully in SendingQueue!!!
0:0:10.533203135 DEBUG (7): RoutingMsgTimer fired! radioBusy = False
0:0:10.533203135 DEBUG (7): Sending RoutingMsg...
0:0:10.533203135 DEBUG (7): RoutingMsg enqueued successfully in SendingQueue!!!
0:0:10.533203165 DEBUG (2): startEpoch()
0:0:10.533203165 DEBUG (2): startEpoch(): Started Epoch Timer. Offset Milli = 39640
0:0:10.533203165 DEBUG (4): startEpoch()
0:0:10.533203165 DEBUG (4): startEpoch(): Started Epoch Timer. Offset Milli = 39640
0:0:10.533203165 DEBUG (7): startEpoch()
0:0:10.533203165 DEBUG (7): startEpoch(): Started Epoch Timer. Offset Milli = 39640
0:0:10.561523448 DEBUG (5): RoutingMsgTimer fired! radioBusy = False
0:0:10.561523448 DEBUG (5): Sending RoutingMsg...
0:0:10.561523448 DEBUG (5): RoutingMsg enqueued successfully in SendingQueue!!!
0:0:10.561523478 DEBUG (5): startEpoch()
0:0:10.561523478 DEBUG (5): startEpoch(): Started Epoch Timer. Offset Milli = 39460

0:0:48.413085948 DEBUG (5): SlotTimer.fired()
0:0:48.413085958 DEBUG (5): calculateData()
0:0:48.413085958 DEBUG (5): calculateData(): No Data Received!!!
0:0:48.413085958 DEBUG (5): calculateData(): Sending Measurement [36] to Parent 4
0:0:48.413085958 DEBUG (5): calculateData(): DataAvgMsg enqueued in SendingAvgQueue successfully!!!
0:0:48.413085958 DEBUG (5): updateSensorValue(): New Measurement = 28
0:0:48.413085978 DEBUG (5): sendAvgDataTask()
0:0:48.413085978 DEBUG (5): sendAvgDataTask(): Send returned success!!!
0:0:48.417800916 DEBUG (4): DataAvg packet received!!! from 5
0:0:48.417968762 DEBUG (5): A Data Avg package sent... True
0:0:48.510742198 DEBUG (5): LostTaskTimer.fired()
0:0:48.560546885 DEBUG (2): SlotTimer.fired()
0:0:48.560546885 DEBUG (4): SlotTimer.fired()
0:0:48.560546885 DEBUG (7): SlotTimer.fired()
0:0:48.560546895 DEBUG (2): calculateData()
0:0:48.560546895 DEBUG (2): calculateData(): No Data Received!!!
0:0:48.560546895 DEBUG (2): calculateData(): Sending Measurement [28] to Parent 1
0:0:48.560546895 DEBUG (2): calculateData(): DataAvgMsg enqueued in SendingAvgQueue successfully!!!
0:0:48.560546895 DEBUG (2): updateSensorValue(): New Measurement = 34
0:0:48.560546895 DEBUG (4): calculateData()
0:0:48.560546895 DEBUG (4): calculateData(): Data Received from 5: ChildCount = 1 : Sum = 36
0:0:48.560546895 DEBUG (4): calculateData(): DataAvgMsg enqueued in SendingQueue successfully!!!
0:0:48.560546895 DEBUG (4): updateSensorValue(): New Measurement = 41
0:0:48.560546895 DEBUG (7): calculateData()
0:0:48.560546895 DEBUG (7): calculateData(): No Data Received!!!
0:0:48.560546895 DEBUG (7): calculateData(): Sending Measurement [43] to Parent 1
0:0:48.560546895 DEBUG (7): calculateData(): DataAvgMsg enqueued in SendingAvgQueue successfully!!!
0:0:48.560546895 DEBUG (7): updateSensorValue(): New Measurement = 50
0:0:48.560546915 DEBUG (2): sendAvgDataTask()
0:0:48.560546915 DEBUG (2): sendAvgDataTask(): Send returned success!!!
0:0:48.560546915 DEBUG (4): sendAvgDataTask()
0:0:48.560546915 DEBUG (4): sendAvgDataTask(): Send returned success!!!
0:0:48.560546915 DEBUG (7): sendAvgDataTask()
0:0:48.560546915 DEBUG (7): sendAvgDataTask(): Send returned success!!!
0:0:48.564178486 DEBUG (1): DataAvg packet received!!! from 7
0:0:48.564346332 DEBUG (7): A Data Avg package sent... True
0:0:48.567749021 DEBUG (1): DataAvg packet received!!! from 2
0:0:48.567916867 DEBUG (2): A Data Avg package sent... True
0:0:48.569076528 DEBUG (1): DataAvg packet received!!! from 4
0:0:48.569244374 DEBUG (4): A Data Avg package sent... True
0:0:48.658203135 DEBUG (2): LostTaskTimer.fired()
0:0:48.658203135 DEBUG (4): LostTaskTimer.fired()
0:0:48.658203135 DEBUG (7): LostTaskTimer.fired()
0:0:48.713867197 DEBUG (1): SlotTimer.fired()
0:0:48.713867207 DEBUG (1): calculateData()
0:0:48.713867207 DEBUG (1): calculateData(): Data Received from 7: ChildCount = 1 : Sum = 43

0:0:48.713867207 DEBUG (1): calculateData(): Data Received from 2: ChildCount = 1 : Sum = 28
0:0:48.713867207 DEBUG (1): calculateData(): Data Received from 4: ChildCount = 2 : Sum = 80
0:0:48.713867207 DEBUG (1): calculateData(): DataAvgMsg enqueued in SendingQueue successfully!!!
0:0:48.713867207 DEBUG (1): updateSensorValue(): New Measurement = 35
0:0:48.713867227 DEBUG (1): sendAvgDataTask()
0:0:48.713867227 DEBUG (1): sendAvgDataTask(): Send returned success!!!
0:0:48.721343990 DEBUG (0): DataAvg packet received!!! from 1
0:0:48.721511836 DEBUG (1): A Data Avg package sent... True
0:0:48.811523447 DEBUG (1): LostTaskTimer.fired()
0:0:48.867187510 DEBUG (0): SlotTimer.fired()
0:0:48.867187520 DEBUG (0): calculateData()
0:0:48.867187520 DEBUG (0): calculateData(): Data Received from 1: ChildCount = 5 : Sum = 188
0:0:48.867187520 DEBUG (0): calculateData(): Sending Results to PC
0:0:48.867187520 DEBUG (0):
 rootResults(): AVG = 37.000000

0:0:48.867187520 DEBUG (0): updateSensorValue(): New Measurement = 31
0:0:49.062500010 DEBUG (0): Epoch: 1
0:0:49.062500010 DEBUG (1): Epoch: 1
0:0:49.062500010 DEBUG (2): Epoch: 1
0:0:49.062500010 DEBUG (4): Epoch: 1
0:0:49.062500010 DEBUG (5): Epoch: 1
0:0:49.062500010 DEBUG (7): Epoch: 1
0:1:26.914062510 DEBUG (5): SlotTimer.fired()
0:1:26.914062520 DEBUG (5): calculateData()
0:1:26.914062520 DEBUG (5): calculateData(): No Data Received!!!
0:1:26.914062520 DEBUG (5): calculateData(): Sending Measurement [28] to Parent 4
0:1:26.914062520 DEBUG (5): calculateData(): DataAvgMsg enqueued in SendingAvgQueue successfully!!!
0:1:26.914062520 DEBUG (5): updateSensorValue(): New Measurement = 31
0:1:26.914062540 DEBUG (5): sendAvgDataTask()
0:1:26.914062540 DEBUG (5): sendAvgDataTask(): Send returned success!!!
0:1:26.919189463 DEBUG (4): DataAvg packet received!!! from 5
0:1:26.919357309 DEBUG (5): A Data Avg package sent... True
0:1:27.011718760 DEBUG (5): LostTaskTimer.fired()
0:1:27.089843760 DEBUG (2): SlotTimer.fired()
0:1:27.089843760 DEBUG (4): SlotTimer.fired()
0:1:27.089843760 DEBUG (7): SlotTimer.fired()
0:1:27.089843770 DEBUG (2): calculateData()
0:1:27.089843770 DEBUG (2): calculateData(): No Data Received!!!
0:1:27.089843770 DEBUG (2): calculateData(): Sending Measurement [34] to Parent 1
0:1:27.089843770 DEBUG (2): calculateData(): DataAvgMsg enqueued in SendingAvgQueue successfully!!!
0:1:27.089843770 DEBUG (2): updateSensorValue(): New Measurement = 39
0:1:27.089843770 DEBUG (4): calculateData()
0:1:27.089843770 DEBUG (4): calculateData(): Data Received from 5: ChildCount = 1 : Sum = 28
0:1:27.089843770 DEBUG (4): calculateData(): DataAvgMsg enqueued in SendingQueue successfully!!!
0:1:27.089843770 DEBUG (4): updateSensorValue(): New Measurement = 31

```

0:1:27.089843770 DEBUG (7): calculateData()
0:1:27.089843770 DEBUG (7): calculateData(): No Data Received!!!
0:1:27.089843770 DEBUG (7): calculateData(): Sending Measurement [50] to Parent 1
0:1:27.089843770 DEBUG (7): calculateData(): DataAvgMsg enqueued in SendingAvgQueue successfully!!!
0:1:27.089843770 DEBUG (7): updateSensorValue(): New Measurement = 43
0:1:27.089843790 DEBUG (2): sendAvgDataTask()
0:1:27.089843790 DEBUG (2): sendAvgDataTask(): Send returned success!!!
0:1:27.089843790 DEBUG (4): sendAvgDataTask()
0:1:27.089843790 DEBUG (4): sendAvgDataTask(): Send returned success!!!
0:1:27.089843790 DEBUG (7): sendAvgDataTask()
0:1:27.089843790 DEBUG (7): sendAvgDataTask(): Send returned success!!!
0:1:27.093307515 DEBUG (2): A Data Avg package sent... True
0:1:27.093536396 DEBUG (7): A Data Avg package sent... True
0:1:27.094238296 DEBUG (4): A Data Avg package sent... True
0:1:27.187500010 DEBUG (2): LostTaskTimer.fired()
0:1:27.187500010 DEBUG (4): LostTaskTimer.fired()
0:1:27.187500010 DEBUG (7): LostTaskTimer.fired()
0:1:27.265625010 DEBUG (1): SlotTimer.fired()
0:1:27.265625020 DEBUG (1): calculateData()
0:1:27.265625020 DEBUG (1): calculateData(): No Data Received!!!
0:1:27.265625020 DEBUG (1): calculateData(): Sending Measurement [35] to Parent 0
0:1:27.265625020 DEBUG (1): calculateData(): DataAvgMsg enqueued in SendingAvgQueue successfully!!!
0:1:27.265625020 DEBUG (1): updateSensorValue(): New Measurement = 42
0:1:27.265625040 DEBUG (1): sendAvgDataTask()
0:1:27.265625040 DEBUG (1): sendAvgDataTask(): Send returned success!!!
0:1:27.268371606 DEBUG (0): DataAvg packet received!!!! from 1
0:1:27.268539451 DEBUG (1): A Data Avg package sent... True
0:1:27.363281260 DEBUG (1): LostTaskTimer.fired()
0:1:27.441406260 DEBUG (0): SlotTimer.fired()
0:1:27.441406270 DEBUG (0): calculateData()
0:1:27.441406270 DEBUG (0): calculateData(): Data Received from 1: ChildCount = 1 : Sum = 35
0:1:27.441406270 DEBUG (0): calculateData(): Sending Results to PC
0:1:27.441406270 DEBUG (0):
    rootResults(): AVG = 33.000000

0:1:27.441406270 DEBUG (0): updateSensorValue(): New Measurement = 37

```

2 Εποχές από MAX

TicksPerSecond : 10000000000

```

0 1 -50.0
1 0 -50.0
1 4 -50.0
4 1 -50.0
1 7 -50.0
7 1 -50.0

```

```
1 2 -50.0
2 1 -50.0
4 5 -50.0
5 4 -50.0
0:0:10.000000000 DEBUG (0): Booted
0:0:10.000000000 DEBUG (0): curdepth = 0 , parentID= 0
0:0:10.000000000 DEBUG (0): Command selected is MAX
0:0:10.000000000 DEBUG (1): Booted
0:0:10.000000000 DEBUG (1): curdepth = 255 , parentID= 65535
0:0:10.000000000 DEBUG (2): Booted
0:0:10.000000000 DEBUG (2): curdepth = 255 , parentID= 65535
0:0:10.000000000 DEBUG (3): Booted
0:0:10.000000000 DEBUG (3): curdepth = 255 , parentID= 65535
0:0:10.000000000 DEBUG (4): Booted
0:0:10.000000000 DEBUG (4): curdepth = 255 , parentID= 65535
0:0:10.000000000 DEBUG (5): Booted
0:0:10.000000000 DEBUG (5): curdepth = 255 , parentID= 65535
0:0:10.000000000 DEBUG (6): Booted
0:0:10.000000000 DEBUG (6): curdepth = 255 , parentID= 65535
0:0:10.000000000 DEBUG (7): Booted
0:0:10.000000000 DEBUG (7): curdepth = 255 , parentID= 65535
0:0:10.000000000 DEBUG (8): Booted
0:0:10.000000000 DEBUG (8): curdepth = 255 , parentID= 65535
0:0:10.000000000 DEBUG (9): Booted
0:0:10.000000000 DEBUG (9): curdepth = 255 , parentID= 65535
0:0:10.000000010 DEBUG (0): Radio initialized successfully!!!
0:0:10.000000010 DEBUG (1): Radio initialized successfully!!!
0:0:10.000000010 DEBUG (2): Radio initialized successfully!!!
0:0:10.000000010 DEBUG (3): Radio initialized successfully!!!
0:0:10.000000010 DEBUG (4): Radio initialized successfully!!!
0:0:10.000000010 DEBUG (5): Radio initialized successfully!!!
0:0:10.000000010 DEBUG (6): Radio initialized successfully!!!
0:0:10.000000010 DEBUG (7): Radio initialized successfully!!!
0:0:10.000000010 DEBUG (8): Radio initialized successfully!!!
0:0:10.000000010 DEBUG (9): Radio initialized successfully!!!
0:0:10.488281260 DEBUG (0): RoutingMsgTimer fired! radioBusy = False
0:0:10.488281260 DEBUG (0):
#####
0:0:10.488281260 DEBUG (0): ##### ROUTING #####
0:0:10.488281260 DEBUG (0): #####
0:0:10.488281260 DEBUG (0): Sending RoutingMsg...
0:0:10.488281260 DEBUG (0): RoutingMsg enqueued successfully in SendingQueue!!!
0:0:10.488281290 DEBUG (0): startEpoch()
0:0:10.488281290 DEBUG (0): startEpoch(): Started Epoch Timer. Offset Milli = 40000
0:0:10.510742197 DEBUG (1): RoutingMsgTimer fired! radioBusy = False
0:0:10.510742197 DEBUG (1): Sending RoutingMsg...
```

0:0:10.510742197 DEBUG (1): RoutingMsg enqueued successfully in SendingQueue!!!
0:0:10.510742227 DEBUG (1): startEpoch()
0:0:10.510742227 DEBUG (1): startEpoch(): Started Epoch Timer. Offset Milli = 39870
0:0:10.539062510 DEBUG (2): RoutingMsgTimer fired! radioBusy = False
0:0:10.539062510 DEBUG (2): Sending RoutingMsg...
0:0:10.539062510 DEBUG (2): RoutingMsg enqueued successfully in SendingQueue!!!
0:0:10.539062510 DEBUG (4): RoutingMsgTimer fired! radioBusy = False
0:0:10.539062510 DEBUG (4): Sending RoutingMsg...
0:0:10.539062510 DEBUG (4): RoutingMsg enqueued successfully in SendingQueue!!!
0:0:10.539062510 DEBUG (7): RoutingMsgTimer fired! radioBusy = False
0:0:10.539062510 DEBUG (7): Sending RoutingMsg...
0:0:10.539062510 DEBUG (7): RoutingMsg enqueued successfully in SendingQueue!!!
0:0:10.539062540 DEBUG (2): startEpoch()
0:0:10.539062540 DEBUG (2): startEpoch(): Started Epoch Timer. Offset Milli = 39740
0:0:10.539062540 DEBUG (4): startEpoch()
0:0:10.539062540 DEBUG (4): startEpoch(): Started Epoch Timer. Offset Milli = 39740
0:0:10.539062540 DEBUG (7): startEpoch()
0:0:10.539062540 DEBUG (7): startEpoch(): Started Epoch Timer. Offset Milli = 39740
0:0:10.569335948 DEBUG (5): RoutingMsgTimer fired! radioBusy = False
0:0:10.569335948 DEBUG (5): Sending RoutingMsg...
0:0:10.569335948 DEBUG (5): RoutingMsg enqueued successfully in SendingQueue!!!
0:0:10.569335978 DEBUG (5): startEpoch()
0:0:10.569335978 DEBUG (5): startEpoch(): Started Epoch Timer. Offset Milli = 39610
0:0:48.567382823 DEBUG (5): SlotTimer.fired()
0:0:48.567382833 DEBUG (5): calculateData()
0:0:48.567382833 DEBUG (5): calculateData(): No Data Received!!!
0:0:48.567382833 DEBUG (5): calculateData(): Sending Measurement [36] to Parent 4
0:0:48.567382833 DEBUG (5): calculateData(): DataMaxMsg enqueued in SendingMaxQueue successfully!!!
0:0:48.567382833 DEBUG (5): updateSensorValue(): New Measurement = 28
0:0:48.567382853 DEBUG (5): sendMaxDataTask(): Send returned success!!!
0:0:48.570129419 DEBUG (4): DataMax packet received!!! from 5
0:0:48.570129419 DEBUG (4): ### RoutingReceive.receive() end #####
0:0:48.570297264 DEBUG (5): A Data Max package sent... True
0:0:48.664062510 DEBUG (2): SlotTimer.fired()
0:0:48.664062510 DEBUG (4): SlotTimer.fired()
0:0:48.664062510 DEBUG (7): SlotTimer.fired()
0:0:48.664062520 DEBUG (2): calculateData()
0:0:48.664062520 DEBUG (2): calculateData(): No Data Received!!!
0:0:48.664062520 DEBUG (2): calculateData(): Sending Measurement [28] to Parent 1
0:0:48.664062520 DEBUG (2): calculateData(): DataMaxMsg enqueued in SendingMaxQueue successfully!!!
0:0:48.664062520 DEBUG (2): updateSensorValue(): New Measurement = 34
0:0:48.664062520 DEBUG (4): calculateData()
0:0:48.664062520 DEBUG (4): calculateData(): Data Received from 5: Value = 36
0:0:48.664062520 DEBUG (4): calculateData(): DataMaxMsg enqueued in SendingQueue successfully!!!
0:0:48.664062520 DEBUG (4): updateSensorValue(): New Measurement = 41
0:0:48.664062520 DEBUG (7): calculateData()

0:0:48.664062520 DEBUG (7): calculateData(): No Data Received!!!
0:0:48.664062520 DEBUG (7): calculateData(): Sending Measurement [43] to Parent 1
0:0:48.664062520 DEBUG (7): calculateData(): DataMaxMsg enqueued in SendingMaxQueue successfully!!!
0:0:48.664062520 DEBUG (7): updateSensorValue(): New Measurement = 50
0:0:48.664062540 DEBUG (2): sendMaxDataTask(): Send returned success!!!
0:0:48.664062540 DEBUG (4): sendMaxDataTask(): Send returned success!!!
0:0:48.664062540 DEBUG (7): sendMaxDataTask(): Send returned success!!!
0:0:48.665039073 DEBUG (5): LostTaskTimer.fired()
0:0:48.668365493 DEBUG (1): DataMax packet received!!! from 2
0:0:48.668365493 DEBUG (1): ### RoutingReceive.receive() end #####
0:0:48.668533339 DEBUG (2): A Data Max package sent... True
0:0:48.672729482 DEBUG (1): DataMax packet received!!! from 7
0:0:48.672729482 DEBUG (1): ### RoutingReceive.receive() end #####
0:0:48.672897328 DEBUG (7): A Data Max package sent... True
0:0:48.673446640 DEBUG (1): DataMax packet received!!! from 4
0:0:48.673446640 DEBUG (1): ### RoutingReceive.receive() end #####
0:0:48.673614486 DEBUG (4): A Data Max package sent... True
0:0:48.761718760 DEBUG (2): LostTaskTimer.fired()
0:0:48.761718760 DEBUG (4): LostTaskTimer.fired()
0:0:48.761718760 DEBUG (7): LostTaskTimer.fired()
0:0:48.762695322 DEBUG (1): SlotTimer.fired()
0:0:48.762695332 DEBUG (1): calculateData()
0:0:48.762695332 DEBUG (1): calculateData(): Data Received from 2: Value = 28
0:0:48.762695332 DEBUG (1): calculateData(): Data Received from 7: Value = 43
0:0:48.762695332 DEBUG (1): calculateData(): Data Received from 4: Value = 44
0:0:48.762695332 DEBUG (1): calculateData(): DataMaxMsg enqueued in SendingQueue successfully!!!
0:0:48.762695332 DEBUG (1): updateSensorValue(): New Measurement = 35
0:0:48.762695352 DEBUG (1): sendMaxDataTask(): Send returned success!!!
0:0:48.763946566 DEBUG (0): DataMax packet received!!! from 1
0:0:48.763946566 DEBUG (0): ### RoutingReceive.receive() end #####
0:0:48.764114411 DEBUG (1): A Data Max package sent... True
0:0:48.860351572 DEBUG (1): LostTaskTimer.fired()
0:0:48.867187510 DEBUG (0): SlotTimer.fired()
0:0:48.867187520 DEBUG (0): calculateData()
0:0:48.867187520 DEBUG (0): calculateData(): Data Received from 1: Value = 44
0:0:48.867187520 DEBUG (0): calculateData(): Sending Results to PC
0:0:48.867187520 DEBUG (0):
 rootResults(): MAX = 44

0:0:48.867187520 DEBUG (0): updateSensorValue(): New Measurement = 31

Τυχαία επιλογή εντολής κατά την εκκίνηση.

```
event void Boot.booted(){
    dbg("Boot", "Booted\n");

    // Start the radio control interface to enable communication
    call RadioControl.start();

    epochCounter = 0;

    // Initialize the random number generator
    call GeneratorSeed.init(time(NULL)+TOS_NODE_ID);

    // Radio is ready for sending
    setRoutingSendBusy(FALSE);

    if(TOS_NODE_ID==0) { // Root node initialization
        curdepth=0;
        parentID=0;
        dbg("Boot", "curdepth = %d , parentID= %d \n", curdepth , parentID);

        // Randomly select the command to run (MAX or AVG)
        if (call RandomGenerator.rand32() % 2 + 1 == 1) {
            dbg("Boot", "Command selected is MAX\n");
            COMMAND_TO_RUN = COMMAND_MAX;
        } else {
            dbg("Boot", "Command selected is AVG\n");
            COMMAND_TO_RUN = COMMAND_AVG;
        }
    }
}
```


Μέγεθος παραθύρων και σταθερών

```
enum{  
    SENDER_QUEUE_SIZE=100,  
    RECEIVER_QUEUE_SIZE=100,  
    AM_SIMPLEROUTINGTREEMSG=22,  
    AM_ROUTINGMSG=22,  
    AM_MAXMSG=24,  
    AM_AVGMSG=25,  
    SEND_CHECK_MILLIS=70000,  
    MAX_DEPTH=200,  
    EPOCH_PERIOD_MILLI= 40*1000,  
    MAX_STEP= 130,  
    AVG_STEP= 180,  
    TIMER_FAST_PERIOD=20,  
    LOST_TASK_PERIOD=100,  
    OFFSET_FIX = 700  
};
```

Stop μετά από 15 δευτερόλεπτα

```
event void EpochTimer.fired() {
    epochCounter++;

    // Stop the experiment after 15 epochs
    if(epochCounter == 15) {
        call RadioControl.stop();
        call EpochTimer.stop();
        return;
    }

    dbg("SRTreeC", "Epoch: %d\n", epochCounter);

    // Start the window timer of the node
    call SlotTimer.startOneShot(offset - OFFSET_FIX);
}
```

Καινούργιες βοηθητικές συναρτήσεις

```
uint16_t generateRandomValue(uint16_t min, uint16_t max) {
    return (rand() % (max - min + 1)) + min;
}

/**
 * Function to initialize the sensor value
 */
void initializeSensorValue() {
    measurement = generateRandomValue(1, 50);
}

/**
 * Function to update the sensor value
 */
void updateSensorValue() {
    int16_t minValue = measurement - (measurement * 30 / 100);
    int16_t maxValue = measurement + (measurement * 30 / 100);

    // Ensure the value stays within 1 and 50
    if (minValue < 1) minValue = 1;
    if (maxValue > 50) maxValue = 50;

    measurement = generateRandomValue(minValue, maxValue);

    dbg("SRTreeC", "updateSensorValue(): New Measurement = %d\n", measurement);
}
```