

# ΑΛΓΟΡΙΘΜΙΚΕΣ ΘΕΜΕΛΙΩΣΕΙΣ ΔΙΚΤΥΩΝ ΑΙΣΘΗΤΗΡΩΝ

End-to-end WSN Project 2023-2024

Αριθμός απόμων: 1-2

Ημερομηνία έναρξης - λήξης: 24/04/2024 - 20/06/2024

Βαθμολογία: 35% επί του τελικού βαθμού

Οι στόχοι της εργασίας είναι οι παρακάτω:

1. Η υλοποίηση ενός δικτύου αισθητήρων σε τοπολογία αστέρα
2. Αποθήκευση των δεδομένων των αισθητήρων σε πραγματικό χρόνο
3. Οπτικοποίηση των μετρήσεων από τους αισθητήρες σε πραγματικό χρόνο
4. Συλλογή δεδομένων μετρήσεων για εκπαίδευση ενός μοντέλου μηχανικής μάθησης

Εάν ενδιαφέρεστε για το project, μπορείτε να στείλετε ένα email στο [kyrgiannopou@gmail.com](mailto:kyrgiannopou@gmail.com) στο οποίο θα αναφέρετε:

- Όνομα και Επώνυμο
- Διεύθυνση ηλεκτρονικού ταχυδρομείου
- Όνομα χρήστη λογαριασμού στο Github
- Λειτουργικό σύστημα στο οποίο θα υλοποιήσετε το project (π.χ. Linux σε VM, MacOs, Native Linux installation, etc.)
- Θέμα μηνύματος: EndToEndWSL Project 2023

Github repository: <https://github.com/IoTLabUpatras/End-to-end-WSN-Project-2024>

## ΕΡΩΤΗΜΑΤΑ

**Ερώτημα 1** (Προγραμματισμός και εγκατάσταση των motes)

1. Βεβαιώστε την ορθή εγκατάσταση των εργαλείων (βλ. README.md) και τρέξτε το παράδειγμα hello-world (βλ. σχετική παρουσίαση για το Contiki-NG)
2. Χρησιμοποιώντας τα αρχεία πηγαίου κώδικα του παραδείγματος nullnet:
  - Ανακαλύψτε τη διεύθυνση MAC του κεντρικού κόμβου της τοπολογίας αστέρα
  - Τροποποιήστε τον κώδικα του αρχείου nullnet-unicast.c έτσι ώστε να στέλνει μηνύματα στο κεντρικό κόμβο (Hardcode τη διεύθυνση MAC)
  - Τροποποιήστε τον κώδικα των αρχέων πηγαίου κώδικα έτσι ώστε οι κόμβοι φύλλα να στέλνουν μετρήσεις κάθε 20 δευτερόλεπτα στον κεντρικό κόμβο και αυτές θα εκτυπώνονται στο terminal μέσω του κόμβου πατέρα.

Υποδείξεις:

- Φροντίστε τα δεδομένα κάθε εισερχόμενου μηνύματος να εκτυπώνονται στην ίδια γραμμή (για οπτικούς λόγους και θα είναι πιο εύκολο για το Ερώτημα 2)
- Για να στείλετε πολλά δεδομένα μέσω ενός μηνύματος, χρησιμοποιήστε την struct που σας δίνεται στο utils.h. Αν θέλετε να τροποποιήσετε τα δεδομένα που αποστέλονται, είστε ελεύθεροι να αλλάξετε τον κώδικα της struct όπως επιθυμείτε
- Οι αισθητήρες παράγουν δεδομένα σε τύπο δεδομένων integer. Χρησιμοποιήστε τις συναρτήσεις temperature\_int2double και humidity\_int2double για να τις μετατρέψετε σε τύπο δεδομένων double.

- Αν θέλετε να εκτυπώσετε αριθμούς κινητής υποδιαστολής στο terminal μέσω των motes χρησιμοποιήστε τη συνάρτηση float2str που σας δίνεται στο utils.h που σας δίνεται. (Μετατροπή σε string και μετά εκτύπωση)
- **Bonus (Προεραιτικό):** Τροποποιήστε τον κώδικα των πηγαίων αρχείων κώδικα για να δημιουργίσετε ένα πολύ απλό πρωτόκολλο επικοινωνίας μεταξύ των κόμβων. Ο σκοπός είναι να αυτοματοποιήσετε την ανακάλυψη της διεύθυνσης MAC του κεντικού κόμβου του δικτύου και την ανάθεση ID's στους κόμβους φύλλα.

Νοητικό παράδειγμα επικοινωνίας μέσω μηνυμάτων:

- Κόμβος πατέρας (nullnet-broadcast.c): I'm the parent node, my MAC address is 0010.0000.0000.0000
- Κόμβοι φύλλα (nullnet-unicast.c): Got your message, I'm a leaf node, my MAC address is 0102.0304.0506.0708
- Κόμβος πατέρας (nullnet-broadcast.c): Got your message leaf node, your ID is 1
- Κόμβος φύλλο (nullnet-unicast.c): Starting measurements transmission:  
ID: 1 Count: 1 Temperature: 25.3 C Humidity: 65.5%  
ID: 1 Count: 2 Temperature: 25.4 C Humidity: 65.2%  
...

## Ερώτημα 2 (Αποθήκευση δεδομένων σε βάση δεδομένων)

1. Εγκαταστείστε μία βάση δεδομένων της επιλογής σας (MySQL, MongoDB, ...)
2. Χρησιμοποιείστε τις παρακάτω εντολές:

```
mkfifo out_pipe
make TARGET=sky PORT=<your-device-usb-port> serialdump > out_pipe
```

για να δημιουργήσετε ένα named pipe, στο οποίο θα στέλνετε το output του κώδικα του κόμβου πατέρα.

3. Sanity Check: Ανάλογα με τη γλώσσα προγραμματισμού που θα χρησιμοποιήσετε στο backend (π.χ. Python, Java, PHP, ...), δημιουργίστε ένα αρχείο κώδικα το οποίο θα διαβάζει συνέχεια (ανά γραμμή) το named pipe και εκτυπώνει (σε δεύτερο terminal, όπου θα τρέχει το αρχείο κώδικα που δημιουργήσατε) το περιεχόμενο της γραμμής που διαβάζει.
4. Τροποποιήστε το αρχείο κώδικα που χρησιμοποιήσατε, έτσι ώστε από κάθε γραμμή δεδομένων που λαμβάνετε να πάρνετε τα απαραίτητα δεδομένα (π.χ. Timestamp, ID, Count, Temp, Humidity) και να τα αποθηκεύετε σε πραγματικό χρόνο στη βάση δεδομένων

### Υποδείξεις

- Αν χρησιμοποιείτε Linux (είτε native installation είτε VM), **φροντίστε να μην χρησιμοποιήσετε το gnome-terminal των Linux, αλλά εγκαταστείστε κάποιο άλλο (π.χ. terminator).** Ο λόγος είναι ότι δε θα μπορείτε αλλιώς να δείτε το output στο δεύτερο terminal σε πραγματικό χρόνο
- Όσον αφορά τα ερωτήματα 2 και 3, θα πρέπει στο δεύτερο terminal να εκτυπώνονται το output του κώδικα του κόμβου πατέρα (σαν να τρέχει στο πρώτο terminal)
- Φροντίστε τα μηνύματα που λαμβάνονται από τον κόμβο πατέρα να είναι σε κατάλληλη μορφή, έτσι ώστε να γίνει parse των γραμμών δεδομένων με εύκολο τρόπο

- Φροντίστε να επεξεργάζονται κατάλληλα οι γραμμές δεδομένων στο δεύτερο αρχέιο κώδικα, ώστε τα δεδομένα να αποθηκεύονται ανάλογα με τον τύπο δεδομένων που πρέπει και όχι σαν strings

### Ερώτημα 3 (Οπτικοποίηση δεδομένων)

Χρησιμοποιώντας πλατφόρμα της επιλογής σας (Προτείνεται Grafana ή υλοποίηση ενός απλού web app με όποιες τεχνολογίες θελήστε) υλοποιήστε μία εφαρμογή στην οποία θα φαίνονται real time οι μετρήσεις των αισθητήρων. Οι χρήστες θα μπορούν επίσης να ανακτούν ιστορικά δεδομένα για τους αισθητήρες.

### Ερώτημα 4 (Intelligence)

1. Θα πρέπει να μαζέψετε ένα επαρκές σύνολο δεδομένων μέσω των αισθητήρων (π.χ. 3 μέρες) το οποίο θα αποθηκευτεί στη βάση δεδομένων σας
2. Θα το χρησιμοποιήσετε για να εκπαιδεύσετε διάφορα μοντέλα τα οποία θα προβλέπουν τις επόμενες τιμές των μετρήσεων. Περισσότερες λεπτομέρειες υπάρχουν στο σχετικό notebook που έχει επισυναφθεί.

#### Υποδείξεις:

Στη προκειμένη φάση δε μας ενδιαφέρει η ακρίβεια του μοντέλου, αλλά μόνο η διαδικασία χρήσης των δεδομένων που συλλέγετε για το συγκεκριμένο σκοπό.

## ΑΝΑΦΟΡΑ

Η πρόοδος του project θα είναι άμεσα συνδεδεμένη με το VCS και κατ' επέκταση το GitHub. Για όλα τα ερωτήματα θα πρέπει να υπάρχει ο αντίστοιχος κώδικας στο repo που θα σας κοινοποιηθεί. Καλό θα ήταν να γράφονται σωστά commit messages ώστε να διευκολυνθεί η διαδικασία διόρθωσης. Επίσης, στο README.md θα πρέπει να υπάρχει μία συνοπτική αναφορά για τα βήματα που ακολουθήσατε σε κάθε ερώτημα καθώς και οδηγίες για την εγκατάσταση της εφαρμογής που δημιουργήσατε. Τέλος, για το ερώτημα 4 μπορείτε επίσης να γράψετε σχόλια μέσα στο notebook για τα βήματα που ακολουθήσατε.