



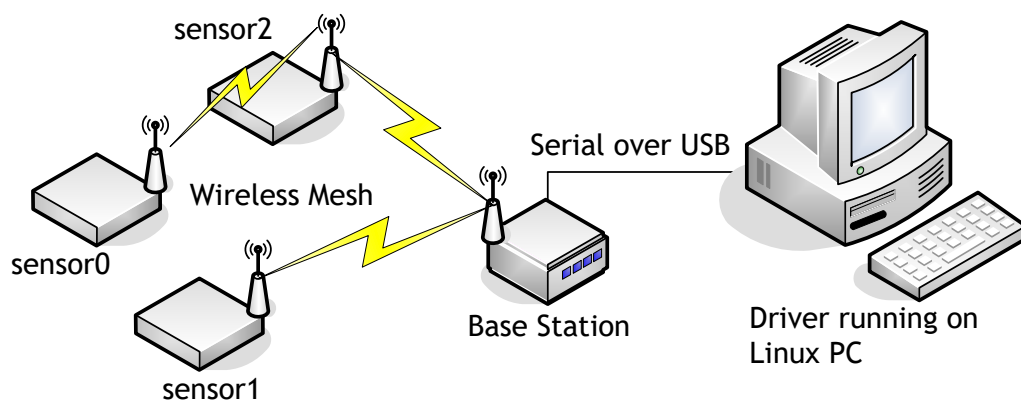
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ  
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

## Άσκηση 2η

# Οδηγός Ασύρματου Δικτύου Αισθητήρων στο λειτουργικό σύστημα Linux

### Εισαγωγή

Αντικείμενο της παρούσας εργαστηριακής άσκησης είναι η υλοποίηση ενός οδηγού συσκευής για ένα ασύρματο δίκτυο αισθητήρων κάτω από το λειτουργικό σύστημα Linux. Το συγκεκριμένο δίκτυο διαθέτει έναν αριθμό από αισθητήρες (ασύρματες κάρτες Crossbow MPR2400CA με αισθητήρες τάσης, θερμοκρασίας και φωτεινότητας MDA100CB) και ένα σταθμό βάσης (ασύρματη κάρτα MPR2400CA και διασύνδεση USB MIB520CB). Ο σταθμός βάσης συνδέεται μέσω USB με υπολογιστικό σύστημα Linux στο οποίο και θα εκτελείται ο ζητούμενος οδηγός συσκευής (ας τον ονομάσουμε Linux:TNG).

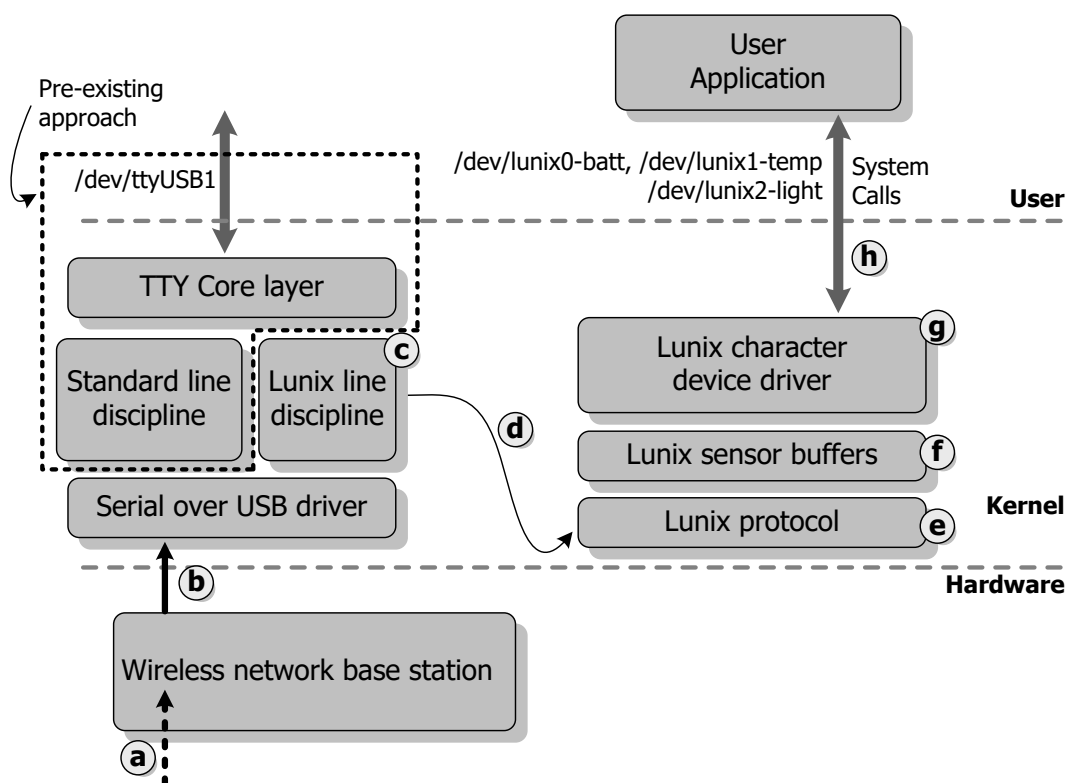


Σχήμα 1: Αρχιτεκτονική του υπό εξέταση συστήματος

Οι αισθητήρες αναφέρουν περιοδικά το αποτέλεσμα τριών διαφορετικών μετρήσεων: της τάσης της μπαταρίας που τους τροφοδοτεί, της θερμοκρασίας και της φωτεινότητας του χώρου όπου βρίσκονται. Τα δεδομένα μεταφέρονται μέσω ενός δικτύου mesh, από εναλλακτικές διαδρομές, έτσι ώστε το δίκτυο να προσαρμόζεται αυτόματα στην περίπτωση όπου ένας ή περισσότεροι αισθητήρες είναι εκτός της εμβέλειας του σταθμού βάσης (Σχήμα 1).

Ο σταθμός βάσης λαμβάνει πακέτα με δεδομένα μετρήσεων, τα οποία προωθεί μέσω διασύνδεσης USB στο υπολογιστικό σύστημα. Η διασύνδεση υλοποιείται με κύκλωμα Serial over USB, για το οποίο ήδη ο πυρήνας διαθέτει ενσωματωμένους

οδηγούς, οπότε τα δεδομένα όλων των μετρήσεων όλων των αισθητήρων εμφανίζονται σε μία εικονική σειριακή θύρα, /dev/ttyUSB1, όπως αναλύεται πιο διεξοδικά παρακάτω.



Σχήμα 2: Αρχιτεκτονική λογισμικού του υπό εξέταση συστήματος

## Ζητούμενα

Ζητείται η κατασκευή οδηγού συσκευής χαρακτήρων (character device driver), ο οποίος θα λαμβάνει τα δεδομένα των μετρήσεων από το δίκτυο αισθητήρων και θα τα εξάγει στο χώρο χρήστη σε διαφορετικές συσκευές, ανάλογα με το είδος της μέτρησης και τον αισθητήρα απ' όπου προέρχεται. Έτσι, αντί να υπάρχει ένα και μοναδικό /dev/ttyUSB1, θα υπάρχουν τρεις συσκευές για κάθε αισθητήρα: π.χ., για τον πρώτο αισθητήρα οι /dev/lunix0-batt, /dev/lunix0-temp και /dev/lunix0-light.

Το τελικό σύστημα (ήδη ανεπτυγμένος κώδικας που σας δίνεται σε συνδυασμό με κώδικα που σας ζητείται) θα λειτουργεί ως εξής: Τα δεδομένα των μετρήσεων αφού (a) λαμβάνονται από το σταθμό βάσης (b) θα προωθούνται μέσω USB στο υπολογιστικό σύστημα. Στη συνέχεια όμως δεν θα ακολουθούν τη συνήθη πορεία τους, προς το /dev/ttyUSB1 αλλά (c) θα παραλαμβάνονται από ένα φίλτρο, από τη διάταξη γραμμής (line discipline) του Linux, η οποία (d) θα τα προωθεί σε στρώμα (e) το οποίο ερμηνεύει το περιεχόμενο των πακέτων και αποθηκεύει τα δεδομένα των μετρήσεων σε διαφορετικούς buffers ανά αισθητήρα (f). Η ζητούμενη συσκευή χαρακτήρων (g), αναλόγως με το ποιο ειδικό αρχείο χρησιμοποιεί η εφαρμογή χρήστη, θα επιτρέπει την ανάκτηση (h) από τους buffers των δεδομένων που αντιστοιχούν σε κάθε περίπτωση.

Η υλοποίηση των τμημάτων (b), (c), (d), (e), (f) σας δίνεται έτοιμη, ως τμήματα κώδικα μαζί με τις κατάλληλες δομές δεδομένων που ορίζουν τις διεπαφές

(interfaces) ανάμεσά τους. Από εσάς ζητείται η υλοποίηση του **(g)** και ο έλεγχος **(h)** της σωστής λειτουργίας του συστήματος, μέσω ενός προγράμματος χρήστη.

Κατ' ελάχιστο, η συσκευή θα υποστηρίζει ανάκτηση των μετρήσεων με χρήση της κλήσης συστήματος `read()`, από πολλές διεργασίες ταυτόχρονα, οπότε απαιτείται κατάλληλος μηχανισμός ελέγχου πρόσβασης (κλειδώματα). Ανάμεσα σε διαδοχικές μετρήσεις, οπότε δεν υπάρχουν δεδομένα, ο οδηγός θα πρέπει να κοιμίζει τη διεργασία χρήστη, ώστε αυτή να μην καταναλώνει χρόνο CPU, όπως περιγράφεται στο συνοδευτικό φυλλάδιο. Προαιρετικά, μπορείτε να εξερευνήσετε τη δυνατότητα ανάκτησης των μετρήσεων χωρίς την εκτέλεση κλήσεων συστήματος, μέσω απεικονίσεων μνήμης ανάμεσα στον πυρήνα και το χώρο χρήστη (memory-mapped I/O). Στην περίπτωση αυτή, ο οδηγός σας χρειάζεται να υποστηρίξει και την κλήση συστήματος `mmap()`.

Σας προτείνουμε η υλοποίηση του **(g)** να γίνει σε δύο ανεξάρτητα στάδια, **(g1)** και **(g2)**, των οποίων τη λειτουργία θα μπορείτε να ελέγξετε χωριστά: το πρώτο στάδιο αφορά στην υλοποίηση των απαραίτητων system calls ώστε να είναι λειτουργική η συσκευή χαρακτήρων. Σε αυτό το στάδιο όμως, δεν είναι αναγκαίο να επιστρέφονται πραγματικές τιμές από τους sensor buffers, ούτε χρειάζεται η διεργασία χρήστη να κοιμάται έως ότου γίνουν διαθέσιμες νέες τιμές από τους αισθητήρες. Το σύστημα θα επιστρέφει πλασματικές, προκαθορισμένες τιμές για τα μετρούμενα μεγέθη. Στο δεύτερο στάδιο **(g2)** θα υλοποιήσετε την επικοινωνία με το τμήμα **(f)**, ώστε να λαμβάνετε πραγματικές τιμές από τους αισθητήρες και να τις επιστρέφετε στο χώρο χρήστη μόνο όταν αυτές έχουν ανανεωθεί.

Ο οδηγός θα αναπτυχθεί ως Linux kernel module σε εικονικές μηχανές, μία για κάθε εργαστηριακή ομάδα. Για το σκοπό αυτό σας δίνεται κατάλληλο σύστημα Debian εγκατεστημένο μέσα σε εικονική μηχανή, στο οποίο θα έχετε πλήρη δικαιώματα διαχείρισης.

Περισσότερες πληροφορίες για την οργάνωση του πυρήνα του Linux, την υποστήριξή του για οδηγούς συσκευών και τη σχεδίαση του ζητούμενου οδηγού μπορείτε να βρείτε στο συνοδευτικό φυλλάδιο «Οδηγός Ασύρματου Δικτύου Αισθητήρων στο Λειτουργικό Σύστημα Linux».