



6^η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ
ΓΙΑ ΤΟ ΜΑΘΗΜΑ "Εργαστήριο Μικροϋπολογιστών"
5^η Εργ. Άσκ. στον Μικροελεγκτή AVR
Συνδιαστική/Επαναληπτική άσκηση – Εφαρμογή Internet of Things (στο easyAVR6)
Εξέταση – Επίδειξη: Τετάρτη 15/1/2020

Έκθεση: Κυριακή 19/1/2020

Σύντομη εισαγωγή στο Internet of Things

Η ραγδαία ανάπτυξη του Internet of Things (IoT) έχει επιφέρει σημαντική αύξηση των συσκευών που είναι συνδεδεμένες στο διαδίκτυο. Ο αριθμός αυτός εξακολουθεί να αυξάνεται με ταχύτατους ρυθμούς. Συνέπεια αυτού είναι η παραγωγή τεράστιου όγκου πληροφοριών προς επεξεργασία.

Τι είναι οι Συσκευές IoT:

Ως συσκευές IoT ορίζουμε κάθε είδους φυσική συσκευή, οχήματα, οικιακές συσκευές και άλλα αντικείμενα, στα οποία υπάρχουν ενσωματωμένα αισθητήρες, λογισμικό, ενεργοποιητές και σύνδεση στο διαδίκτυο, που τους επιτρέπει να επικοινωνούν και να ανταλλάσσουν δεδομένα είτε μεταξύ τους, είτε με το υπόλοιπο διαδίκτυο. Αυτές οι συσκευές έχουν περιορισμένους πόρους σε ό,τι αφορά την cpu, τη μνήμη και τις δικτυακές τους ικανότητες.

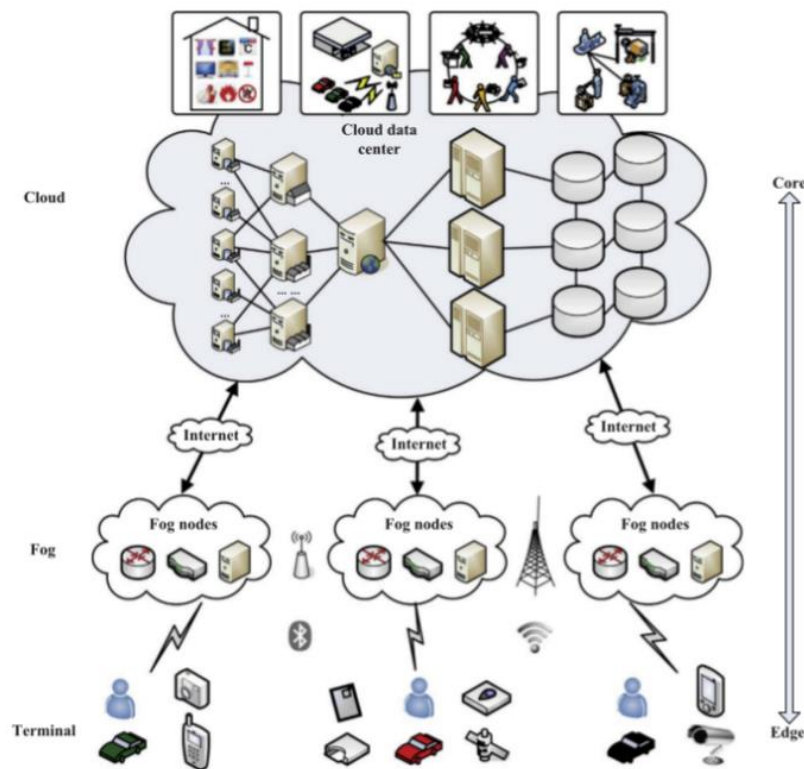
Τι είναι IoT Gateway:

Καθώς δισεκατομύρια συσκευές συνδέονται πλέον στο δίκτυο, το «Gateway» χρειάζεται προκειμένου να συνδέει υπο-δίκτυα διαφορετικών αρχιτεκτονικών και σε διαφορετικά περιβάλλοντα. Στην κλασική δομή του IoT χρησιμοποιείται ως διαμεσολαβητής/γέφυρα μεταξύ των τελικών συσκευών και του Cloud διαμορφώνοντας κατάλληλα τα δεδομένα που στέλνονται και λαμβάνονται. Τα Gateways συνδέουν επίσης στο Cloud υπο-δίκτυα που στο εσωτερικό τους μπορεί να χρησιμοποιούν διαφορετικά πρωτόκολλα επικοινωνίας. Τα Gateways εποπτεύουν την διαδικασία επικοινωνίας εντός και εκτός του υπο-δικτύου στο οποίο βρίσκονται. Μπορούν να εκτελούν διεργασίες ελέγχου, ασφάλειας, φιλτραρίσματος των δεδομένων καθώς επίσης να κάνουν σε χαμηλότερο επίπεδο επεξεργασία των δεδομένων που λαμβάνουν (πιο κοντά στις συσκευές και όχι στο Cloud). Τέλος, μεγάλος αριθμός δεδομένων που λαμβάνονται από τους αισθητήρες των υπο-δικτύων IoT, χρειάζεται το Gateway ως σύστημα συντονισμού και ελέγχου της μεταφοράς τους όχι μόνο από μία συσκευή στο Cloud αλλά και από συσκευή σε συσκευή.

Edge / Fog Computing:

Ωστόσο, τα τελευταία χρόνια, λόγω του γεγονότος ότι οι υποδομές του Cloud είναι γεωγραφικά κεντροποιημένες και απομακρυσμένες από τους χρήστες, οι εφαρμογές που απαιτούν επεξεργασία σε πραγματικό χρόνο, με χαμηλό latency αδυνατούν να εκτελεστούν επιτυχώς. Αυτό συμβαίνει λόγω μεγάλου χρόνου μεταφοράς των δεδομένων, συμφόρησης του δικτύου και υποβάθμισης της ποιότητας. Επιπλέον, σε εφαρμογές που απαιτείται ιδιωτικότητα των δεδομένων (π.χ. ιατρικές εφαρμογές), αυτή δε μπορεί να διασφαλιστεί απόλυτα σε απομακρυσμένες υποδομές. Αυτοί οι λόγοι οδήγησαν στη δημιουργία του Edge και του Fog Computing, όπου οι απαιτούμενοι υπολογισμοί γίνονται κοντά στις IoT συσκευές. Το σχήμα της αρχιτεκτονικής αυτής παρουσιάζεται στην Εικόνα 1. Έτσι οι

χρήστες ανεξαρτητοποιούνται από τις υποδομές του Cloud και, ταυτόχρονα, ενισχύεται η προστασία των προσωπικών τους δεδομένων.



Εικόνα 1 Αρχιτεκτονική Fog και Edge Computing¹

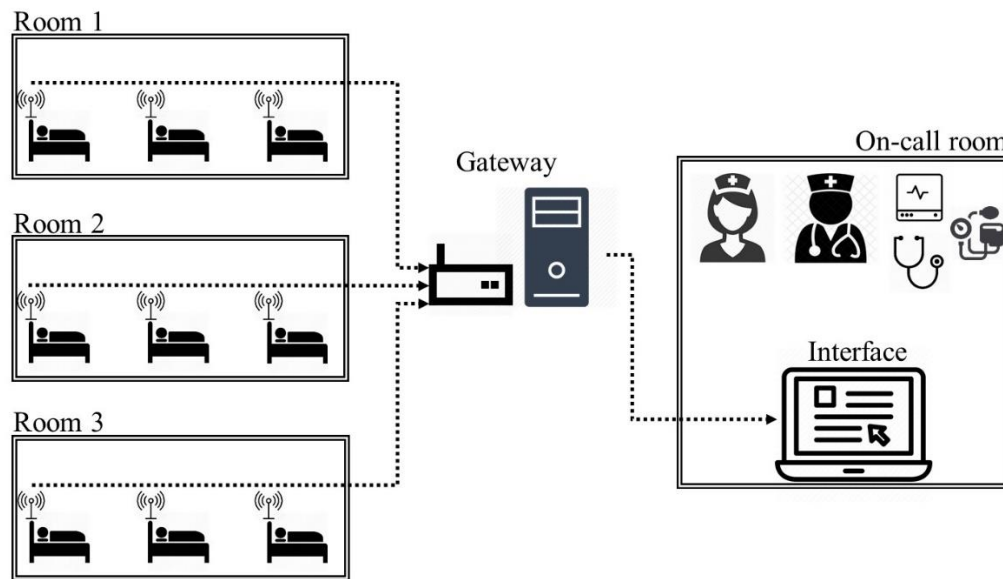
Από τη μία πλευρά, οι συσκευές IoT βρίσκονται κοντά στους καθημερινούς χρήστες και πιο συγκεκριμένα, στο terminal layer της αρχιτεκτονικής που φαίνεται στην Εικόνα 1. Από την άλλη πλευρά ο Edge Node (Gateways του Edge Computing, τα οποία συνεισφέρουν στην ανάπτυξη υπηρεσιών Edge και στη παροχή υπολογιστικών, αποθηκευτικών και δικτυακών πόρων στις συσκευές IoT) βρίσκεται στη διεπαφή του Fog layer και του terminal layer. Επιπροσθέτως, οι συσκευές IoT είναι συνδεδεμένες, είτε ενσύρματα, είτε ασύρματα, με το Gateway κόμβο και μπορούν να επικοινωνούν και να ανταλλάσσουν δεδομένα με εκείνον.

Η αρχιτεκτονική του συστήματος που θα μελετήσουμε στα πλαίσια της παρούσας εργασίας αποτελείται από ένα σύνολο n IoT συσκευών (easyAVR πλακέτες) και μίας συσκευής Gateway. Δεδομένα από αισθητήρες συνδεδεμένους στις συσκευές μπορούν να σταλούν προς επεξεργασία στο Gateway, ενώ τον Gateway μπορεί να στέλνει δεδομένα πίσω στις συσκευές.

Περιγραφή της εφαρμογής

Στην παρούσα άσκηση θα μελετηθεί μια εφαρμογή νοσοκομείου. Η Εικόνα 2 δίνει το γενικό σχήμα της εφαρμογής. Το κάθε κρεβάτι (IoT node) έχει συνδεδεμένους αισθητήρες στον ασθενή, τα δεδομένα των οποίων στέλνονται στο Gateway. Το Gateway επεξεργάζεται τα δεδομένα και είτε στέλνει κάποια απάντηση στο IoT node προκειμένου να γίνει κάποια αυτόματη λειτουργία, είτε στέλνει τα δεδομένα στο δωμάτιο των νοσηλευτών και των γιατρών. Τα δεδομένα παρουσιάζονται σε μια οθόνη (Interface) ούτως ώστε να μελετηθούν από το προσωπικό του νοσοκομείου προκειμένου να προβεί στις κινήσεις που απαιτούνται.

¹ P. Hu, S. Dhelim, H. Ning, and T. Qiu. Survey on fog computing: architecture, key technologies, applications and open issues. Journal of Network and Computer Applications, 2017



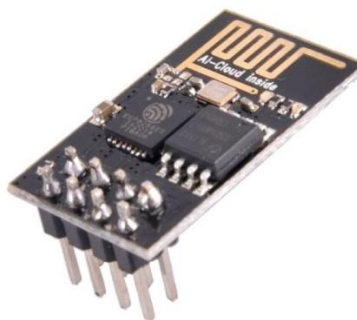
Εικόνα 2 Η εφαρμογή IoT που θα μελετηθεί στην παρούσα άσκηση

Η κάθε συσκευή IoT που βρίσκεται σε κάθε κρεβάτι θα είναι μια πλακέτα easyAVR, το πρόγραμμα της οποίας θα γραφτεί από την κάθε ομάδα. Ο αριθμός της ομάδας θα είναι ο αριθμός του κρεβατιού. Για λόγους απλότητας ο αισθητήρας θα είναι ο αισθητήρας θερμοκρασίας που μελετήθηκε στην 4^η Εργαστηριακή άσκηση.

Το Gateway θα είναι ήδη προγραμματισμένο και το Interface του δωματίου προσωπικού του νοσοκομείου θα παρουσιάζεται στον προβολέα του εργαστηρίου.

Ο πομποδέκτης WiFi ESP8266:

Ο πομποδέκτης Wi-Fi που θα χρησιμοποιήσουμε για την παραπάνω εφαρμογή είναι ο ESP8266 (Εικόνα 3). Πρόκειται για έναν προγραμματίσιμο πομποδέκτη ο οποίος έχει προγραμματιστεί να δέχεται εντολές μέσω σειριακής και να στέλνει αντίστοιχη απάντηση.



Εικόνα 3 Ο πομποδέκτης WiFi ESP8266

Συγκεκριμένα, οι εντολές που δέχεται φαίνονται στον παρακάτω πίνακα:

command	usage
baudrate: "Your baudrate"	Resets the Serial to the required speed
configuration	Reports your current configuration
connect	Uses the provided SSID and Password to connect to the WiFi network

debug: "true or false"	Every command will send extra debug information if set to true
help	Lists all commands
hostIP: "Your HostIP"	Your Host's IP address
password: "Your Password"	Password to be used for the connection
payload: [Your payload]	The payload to post to the Host. Must be in brackets
ready: "true or false"	Your Team's status
ssid: "Your SSID"	SSID to be used for the connection
teamname: "Your Teamname"	Your Team's name
transmit	Transmits the specified payload

Το payload που δέχεται ο server πρέπει να είναι σε μορφή json και συγκεκριμένα η εντολή πρέπει να είναι στην μορφή:

```
payload: [{"name": "Temperature", "value": 36}]
```

Το όνομα πρέπει να είναι οπωσδήποτε *Temperature* και η τιμή αριθμός. Σε όποια εντολή υπάρχουν τα "" πρέπει να τα στέλνεται. Κάθε εντολή πρέπει να τελειώνει με τον χαρακτήρα αλλαγής γραμμής \n. Τα ssid, password, hostIP και baudrate(9600) έχουν ήδη ρυθμιστεί για εσάς.

Όταν το ESP8266 είναι συνδεδεμένο στην πλακέτα **τα jumper από την UART πρέπει να είναι αποσυνδεδεμένα.**

Για τα τρία πρώτα ερωτήματα θα χρησιμοποιήσετε μόνο την UART και θα προσομοιώνετε εσείς τις απαντήσεις του ESP8266 (που θα έρχονται από το Gateway).

ΜΟΝΟ στο τελευταίο ερώτημα και αφού έχετε βεβαιωθεί για την ορθή λειτουργία των προηγούμενων βημάτων θα συνδέσετε το ESP8266 στην πλακέτα.

Τα ζητούμενα της 6ης εργαστηριακής άσκησης

Η 6^η άσκηση αποτελεί στην ουσία επανάληψη των όσων αναπτύξατε στις προηγούμενες ασκήσεις. Πιο συγκεκριμένα θα χρησιμοποιήσετε γνώση από την Άσκηση 5 (σειριακή επικοινωνία), από την Άσκηση 4 (έλεγχος αισθητήρα θερμοκρασίας), καθώς και την χρήση πληκτρολογίου και οθόνης από την Άσκηση 3.

Ζήτημα 6α) Να γραφεί πρόγραμμα για τον ATmega16 στο σύστημα EasyAvr6 το οποίο να στέλνει στην UART κατάλληλη εντολή για να ρυθμιστεί το όνομα της ομάδας σας στο ESP8266. Στη συνέχεια, θα διαβάζει από την UART την απάντηση του ESP8266 και αν αυτή είναι "Success" θα απεικονίζει στην LCD το μήνυμα *1.Success* ενώ αν είναι "Fail" θα τυπώνει στην LCD το μήνυμα *1.Fail*. Επαναλάβετε το ίδιο για την εντολή connect με μηνύματα *2.Success* και *2.Fail* αντίστοιχα. Μπορείτε να προσθέσετε μια μικρή καθυστέρηση πριν την αποστολή της δεύτερης εντολής για να προλάβετε να δείτε τα μηνύματα στην LCD.

Ζήτημα 6β) Επεκτείνετε το παραπάνω πρόγραμμα ώστε να διαβάζει μια μέτρηση από τον αισθητήρα θερμοκρασίας DS1820 και αφού την μετατρέψει στην κατάλληλη μορφή για να ρυθμιστεί το payload να το στέλνει στην UART. Ομοίως με πριν, θα διαβάζει την απάντηση από την UART και θα τυπώνει είτε *3.Success* είτε *3.Fail*.

Ζήτημα 6γ) Επεκτείνετε το παραπάνω πρόγραμμα ώστε να ελέγχει εάν πατήθηκε στο πληκτρολόγιο το πλήκτρο που αντιστοιχεί στο τελευταίο ψηφίο της ομάδας σας. Εάν πατήθηκε να στέλνει στην UART κατάλληλη εντολή για να ρυθμιστεί η κατάσταση της ομάδας σας σε έτοιμη για εξέταση. Αν πατήθηκε οποιοδήποτε άλλο πλήκτρο θεωρείτε ότι η ομάδα σας δεν είναι έτοιμη για εξέταση. Το πρόγραμμα σας **ΔΕΝ** πρέπει να περιμένει να πατηθεί κάποιο πλήκτρο. Ομοίως με πριν, θα διαβάζετε την απάντηση από την UART και θα τυπώνετε είτε *4.Success* είτε *4.Fail*.

Ζήτημα 6δ) Επεκτείνετε το παραπάνω πρόγραμμα ώστε να στέλνει στην UART κατάλληλη εντολή για να την αποστολή του payload στο Gateway. Να τυπώνετε στην οθόνη την απάντηση του server. Το πρόγραμμα πρέπει να επαναλαμβάνει την λειτουργία από το ερώτημα β μέχρι και το δ.

Για όλα τα ερωτήματα μπορείτε να δουλέψετε σε γλώσσα της επιλογής σας C ή assembly (προτείνεται για δική σας ευκολία η C καθώς όλες οι συναρτήσεις που θα χρησιμοποιήσετε είναι έτοιμες από τις προηγούμενες ασκήσεις).