



**ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ**  
**ΓΙΑ ΤΟ ΜΑΘΗΜΑ "Εργαστήριο Μικροϋπολογιστών"**  
**Bonus άσκηση – Εφαρμογή Internet of Things**

## **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:**

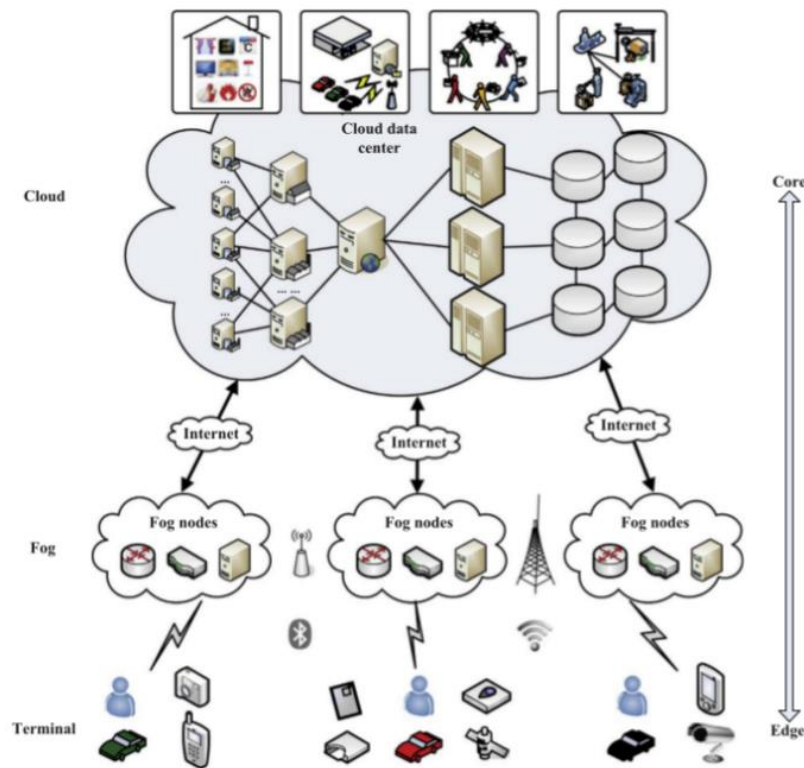
Ως συσκευές IoT ορίζουμε κάθε είδους φυσική συσκευή, οχήματα, οικιακές συσκευές και άλλα αντικείμενα, στα οποία υπάρχουν ενσωματωμένα αισθητήρες, λογισμικό, ενεργοποιητές και σύνδεση στο διαδίκτυο, που τους επιτρέπει να επικοινωνούν και να ανταλλάσσουν δεδομένα είτε μεταξύ τους, είτε με το υπόλοιπο διαδίκτυο. 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 Computing<sup>1</sup>

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.

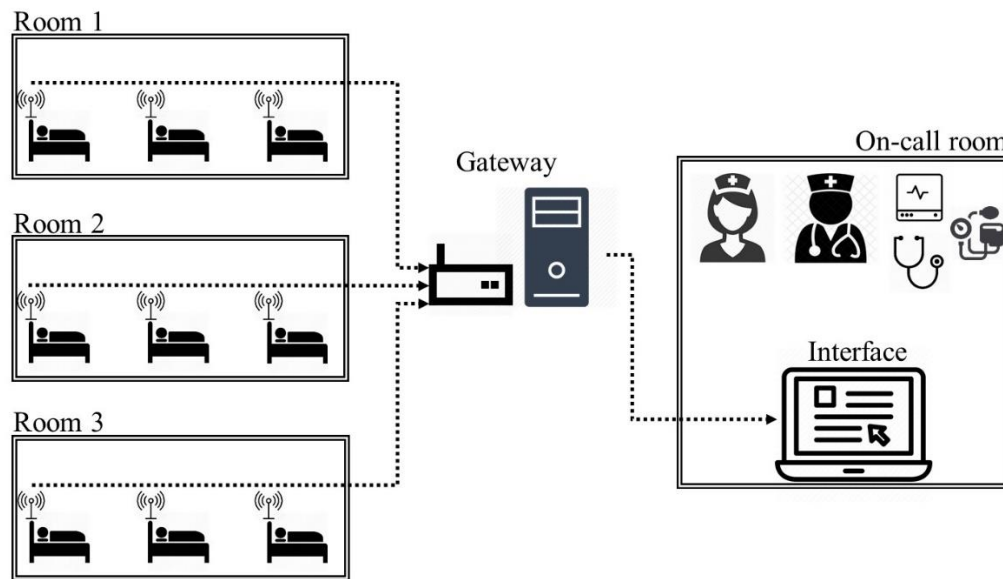
Η αρχιτεκτονική του συστήματος που θα μελετήσουμε στα πλαίσια της παρούσας εργασίας αποτελείται από ένα σύνολο  $n$  IoT συσκευών (easyAVR πλακέτες) και μίας συσκευής Gateway. Δεδομένα από αισθητήρες συνδεδεμένους στις συσκευές μπορούν να σταλούν προς επεξεργασία στο Gateway, ενώ τον Gateway μπορεί να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

## 1ο Μέρος

### 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.

<sup>1</sup> 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



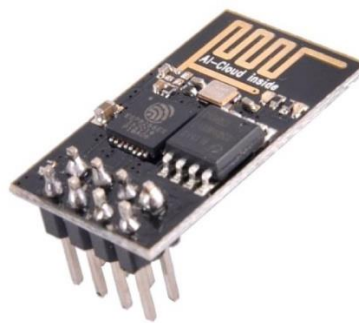
Εικόνα 2 Η εφαρμογή IoT που θα μελετηθεί στην παρούσα άσκηση

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<sup>η</sup> 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
<b>addSensor: "SensorName"</b>	Add a sensor Name
<b>APStart</b>	Uses the provided SSID and Password to create an Access Point
<b>baudrate: "Your baudrate"</b>	Resets the Serial to the required speed

<b>clientTransmit</b>	Sends all SensorValues to the connected server
<b>connect</b>	Uses the provided SSID and Password to connect to the WiFi network
<b>debug: "true or false"</b>	Every command will send extra debug information if set to true
<b>help</b>	Lists all commands
<b>getAllValues</b>	Sends the values of all Sensors
<b>getValue: "SensorName"</b>	Send the value of the specified Sensor
<b>hostIP: "Your HostIP"</b>	Your Host's IP address
<b>password: "Your Password"</b>	Password to be used for the connection
<b>restart</b>	Restarts the ESP
<b>sensorValue</b>	saves value to specified sensor. Values shouldn't contain = and ,
<b>ssid: "Your SSID"</b>	SSID to be used for the connection

Προκειμένου το ESP8266 να αναγνωρίσει ότι πρέπει να διαβάσει τα δεδομένα της σειριακής, οτιδήποτε στείλετε πρέπει να ξεκινάει με ESP: (π.χ. για σύνδεση θα στείλετε ESP:connect).

Έχουν προγραμματιστεί δύο πιθανές λειτουργίες:

- 1) Το ESP μπορεί να λειτουργήσει σαν Access Point, δηλαδή να δημιουργήσει ένα δίκτυο WiFi με τα SSID και Password που θα ορίσετε. Ταυτόχρονα, το ESP ξεκινάει έναν server στο δίκτυο αυτό.
- 2) Το ESP μπορεί να συνδεθεί στον server σαν client και να στέλνει αιτήματα στον server για να γίνει ανταλλαγή δεδομένων.

Προκειμένου να κατασκευάσετε ένα δίκτυο IoT κάποια από τα ESP που θα έχετε στην διάθεση σας θα χρησιμοποιούνται σαν Access Points και κάποια σαν clients. Μια καλή προσέγγιση είναι να ορίσετε αρκετά Access Points ώστε να καλύψετε την επιθυμητή έκταση και τα υπόλοιπα να είναι clients. Επιπλέον, ένας client μπορεί να συνδεθεί σε έναν server (σε ένα ESP) και μετά σε έναν άλλο server (άλλο ESP), κάτι που θα μπορούσατε να εκμεταλλευτείτε ώστε να προωθείτε σταδιακά τα δεδομένα από τους πιο απομακρυσμένους nodes προς τον Gateway.

Στην περίπτωση που το debug είναι false (κυρίως με false θα λειτουργείτε αφού τα μηνύματα μπορεί να είναι αρκετά μεγάλα για να τα εμφανίσετε στην LCD σε διαφορετική περίπτωση) τα ESP είναι προγραμματισμένα να απαντάνε "Success" όταν μια εντολή εκτελείται επιτυχώς και "Fail" σε διαφορετική περίπτωση. Εξαιρέση αποτελούν οι εντολές getValue και getAllValues οι οποίες επιστρέφουν την τιμή ενός ή όλων των αισθητήρων αντίστοιχα. Επιπλέον, στο ESP που λειτουργεί σαν Access Point θα επιστρέφεται το ServedClient σε περίπτωση που συνδέθηκε κάποιος client και πραγματοποιήσε ανταλλαγή δεδομένων με την εντολή clientTransmit.

Για να ζητήσει ο client την τιμή κάποιου αισθητήρα από τον server αρκεί να θέσετε την τιμή του αισθητήρα αυτού σε request.

Σε κάθε clientTransmit ο client στέλνει τις τιμές από όλους τους αισθητήρες και όσες δεν είναι request θα αντικαταστήσουν τις αντίστοιχες τιμές στον server. Ο server αντικαθιστά/στέλνει μόνο τιμές που αφορούν αισθητήρες που έχουν περαστεί με την εντολή addSensor. (πχ αν ο client έχει Sensor3=request ο server δεν θα στείλει την τιμή του αν δεν έχει καταχωρημένο τον Sensor3)

Προσέξτε επίσης ότι μπορεί να λάβετε clientTransmit οποιαδήποτε στιγμή αφού έχετε ξεκινήσει τον server καθώς δεν γνωρίζεται σε ποιο σημείο θα συνδεθεί ο client για να πραγματοποιήσει την ανταλλαγή.

Δώστε προσοχή στα εξής:

- 1) Κάθε εντολή που στέλνετε καθώς και κάθε απάντηση που λαμβάνετε θα έχει στο τέλος τον χαρακτήρα αλλαγής γραμμής '\n'.
- 2) Μερικές φορές οι client αποσυνδέονται οπότε να ελέγχετε κάθε φορά με την εντολή connect και μπορείτε σε περίπτωση που αποτύχει να δοκιμάζετε άλλη μια φορά (σε περίπτωση αποτυχίας υπάρχει timeout 20sec όπου το ESP δεν θα αποκρίνεται).
- 3) Μην ξεχάσετε να αλλάξετε το default ssid καθώς σε διαφορετική περίπτωση θα δημιουργηθούν πολλά ίδια δίκτυα στον χώρο και θα υπάρχει διένεξη.
- 4) Μπορείτε στην αρχή του προγράμματος σας να στέλνετε την εντολή restart για να σιγουρευτείτε ότι το ESP βρίσκεται στις default επιλογές του. Μετα την επανεκκίνηση το ESP τυπώνει μια γραμμή με κάποιες πληροφορίες και την γραμμή ESP8266: Waiting for command.

Αν για το project που επιλέξετε να υλοποιήσετε θεωρείτε ότι δεν σας καλύπτουν οι λειτουργίες που υπάρχουν ήδη στο ESP μπορείτε να μας στείλετε τον κώδικα του ESP με την αλλαγή που θέλετε να γίνει ώστε να προγραμματίσουμε ξεχωριστά κάποια ESP.

Default τιμές:

<b>baudrate</b>	9600
<b>debug</b>	false
<b>hostIP</b>	192.168.4.1
<b>password</b>	awesomePassword
<b>ssid</b>	awesomeAccessPoint

Σας δίνονται δύο παραδείγματα επικοινωνίας με το ESP:

SENDER	MESSAGE
AVR	ESP:addSensor: "Sensor1"
ESP	"Success"
AVR	ESP:sensorValue:"Sensor1"[128]
ESP	"Success"
AVR	ESP:getValue:"Sensor1"
ESP	Sensor1="128"
AVR	ESP:APStart
ESP	"Success"
ESP	ServedClient

Client:

AVR	ESP:addSensor: "Sensor1"
ESP	"Success"
AVR	ESP:sensorValue:"Sensor1"[request]
ESP	"Success"
AVR	ESP:connect
ESP	"Success"
AVR	ESP:clientTransmit
ESP	"Success"
AVR	ESP:getAllValues
ESP	Sensor1="128"

Σε όποια εντολή υπάρχουν τα "" πρέπει να τα στέλνετε. Κάθε εντολή που στέλνετε καθώς και κάθε απάντηση που λαμβάνετε θα έχει στο τέλος τον χαρακτήρα αλλαγής γραμμής '\n'.

**Μπορείτε να χρησιμοποιήσετε την UART και να προσομοιώνετε εσείς τις απαντήσεις του ESP8266 από τον υπολογιστή για να σιγουρευτείτε ότι λειτουργεί σωστά η σειριακή επικοινωνία. Σε αυτή την περίπτωση μην προσθέσετε το ESP: στην αρχή των μηνυμάτων από τον AVR γιατί το ESP8266 είναι ήδη συνδεδεμένο στην πλακέτα και θα στέλνει απαντήσεις.**

Για οποιαδήποτε εφαρμογή επιλέξετε θεωρείστε ότι έχετε στη διάθεση σας όλα τα περιφερειακά που χρησιμοποιήσατε και επιπλέον το θερμόμετρο DS1820. Απαραίτητη είναι η χρήση της σειριακής επικοινωνίας. Για το θερμόμετρο και την σειριακή θα βρείτε πληροφορίες στα pdf.

## Πιθανές εφαρμογές

### 1)Smart Traffic

Στόχος μας είναι να βελτιώσουμε την κυκλοφορία με τον έλεγχο των φωτεινών σηματοδοτών. Μια πιθανή υλοποίηση είναι η τοποθέτηση ενός Node σε κάθε σηματοδότη σε συνδυασμό με μια συσκευή που να μπορεί να ανιχνεύει τον αριθμό των

αυτοκινήτων (πχ μια κάμερα). Τα δεδομένα από πολλές διασταυρώσεις μπορούν να συγκεντρώνονται από κάποιον από τους Nodes και να προωθούνται με κάποιον τρόπο προς ένα Gateway υπεύθυνο για την ανάλυση των δεδομένων κίνησης και την εφαρμογή κάποιου αντίστοιχου αλγόριθμου για την ελάφρυνση της.

## 2)Έλεγχος υδάτων

Στόχος μας είναι να ελέγχουμε το βακτηριδιακό φορτίο του νερού σε παραλίες, λίμνες και ποτάμια. Γι αυτό απαιτείται η τοποθέτηση αισθητήρων σε κατάλληλες αποστάσεις. Μια πιθανή υλοποίηση είναι η τοποθέτηση Nodes που να διαβάζουν κατάλληλους αισθητήρες ανίχνευσης βακτηριδίων σε κατάλληλες αποστάσεις και η σταδιακή προώθηση των δεδομένων προς ένα Gateway.

## 3)Πυρκαγιές

Στόχος μας είναι η προφύλαξη των δασικών εκτάσεων από πυρκαγιές. Μια πιθανή υλοποίηση είναι η τοποθέτηση Nodes, σε κατάλληλες αποστάσεις, που να διαβάζουν αισθητήρες οι οποίοι θα πραγματοποιούν ανίχνευση καπνού ή και υψηλής θερμοκρασίας και θα προωθούν σταδιακά τα δεδομένα προς ένα Gateway υπεύθυνο για την ενεργοποίηση ενός συστήματος πυρόσβεσης.

## 4)Γεωργία

Στόχος μας είναι η αυτοματοποίηση της κάλυψης κάποιων αναγκών μιας καλλιέργειας. Μια πιθανή υλοποίηση είναι η τοποθέτηση Nodes σε κατάλληλες αποστάσεις, που θα διαβάζουν αρκετά δεδομένα που να μπορούν να δώσουν μια εικόνα σχετικά με το αν οι συνθήκες που επικρατούν ευνοούν την ανάπτυξη της καλλιέργειας (π.χ θερμοκρασία, υγρασία, pH εδάφους). Τα δεδομένα θα προωθούνται σταδιακά προς ένα Gateway υπεύθυνο για το πότισμα της καλλιέργειας και την ενημέρωση των διαχειριστών της καλλιέργειας σε περίπτωση αντίξοων συνθηκών.

Είστε ελεύθεροι να διαλέξετε κάποια από τις παραπάνω εφαρμογές, αν επιθυμείτε να την τροποποιήσετε, ή και να επιλέξετε μια διαφορετική δική σας εφαρμογή. Παραδοτέα της άσκησης θα είναι μια περιγραφή του προβλήματος που θέλετε να αντιμετωπίσετε, μια πρόταση επίλυσης του, και ένα proof of concept που θα περιέχει κώδικες. Προφανώς δεν υπάρχει απεριόριστη διαθεσιμότητα σε Nodes ή σε εξειδικευμένους αισθητήρες και γι αυτό για το proof of concept πρέπει να χρησιμοποιήσετε όσο το δυνατόν λιγότερο hardware μπορείτε και να προσομοιώσετε την λειτουργία των εξειδικευμένων αισθητήρων με τα περιφερειακά που έχετε διαθέσιμα. Πχ μπορείτε για το 1) να θεωρήσετε ότι ο αριθμός των αυτοκινήτων αντιστοιχεί στον αριθμό που διαβάσετε από τον ADC.

Σας υπενθυμίζονται οι περιορισμοί του προγράμματος απομακρυσμένης πρόσβασης:

- 1) Για να κάνετε reset στον ATMEGA16 χρησιμοποιείτε το push button RS που βρίσκεται αριστερά του PD7. Το Reset που βρίσκεται δίπλα από το Disconnect κάνει reset σε όλο το σύστημα σε περίπτωση που έχει κολλήσει.
- 2) Για να κάνετε Upload ένα αρχείο πρέπει να έχετε πατήσει πρώτα το connect και να φαίνεται ότι είναι Connected.
- 3) Για να ενεργοποιήσετε την ανάγνωση ενός LED πρέπει να κάνετε δεξί κλικ στο συγκεκριμένο LED. Για να την απενεργοποιήσετε κάνετε αριστερό κλικ.
- 4) Τα Push Buttons λειτουργούν με αριστερό κλικ (πατημένο = Λογικό 1) και αν θέλετε μπορείτε να κάνετε δεξί κλικ ώστε να μείνει πατημένο το συγκεκριμένο Push Button. Επανέρχεται με αριστερό κλικ.
- 5) Μπορείτε να χρησιμοποιήσετε τα LED της PORTB αλλά κανένα από τα Push Button ή μπορείτε να έχετε μονάχα 1 Push Button ενεργοποιημένο κάθε χρονική στιγμή και κανένα από τα LED.
- 6) Δεν μπορείτε για οποιαδήποτε PORT να χρησιμοποιείται και το LED και το Push Button.
- 7) Δεν μπορείτε να χρησιμοποιήσετε το keypad αν δεν το διαβάσει ο ATMEGA16 με την ρουτίνα scan\_keypad\_rising\_edge\_sim αλλιώς το κουμπί που πατήσατε θα παραμείνει κόκκινο και θα πρέπει να κάνετε reset στο σύστημα. Για να ενεργοποιήσετε την λειτουργία του πληκτρολογίου χρησιμοποιήστε τον διακόπτη ακριβώς από πάνω (1=ενεργοποιημένο)
- 8) Για να ενεργοποιήσετε την αποκωδικοποίηση των σημάτων της PORTD σαν να είναι συνδεδεμένη η οθόνη LCD χρησιμοποιήστε τον διακόπτη πάνω από την LCD (1=ενεργοποιημένη). Για να λειτουργήσει σωστά σιγουρευτείτε ότι ο ATMEGA16 έχει εκτελέσει την ρουτίνα lcd\_init ενώ η LCD είναι ενεργοποιημένη.
- 9) Η τιμή του slider για τον ADC (δεξιά πάνω στην πλακέτα) αποστέλλεται όταν αφήσετε το πλήκτρο του ποντικιού. Η τιμή δεν αντιπροσωπεύει κάτι.

## 2ο Μέρος

Όπως περιεγράφηκε στο προηγούμενο μέρος, για την επικοινωνία χρησιμοποιείται ο πομποδέκτης ESP8266, ο οποίος είναι ένας προγραμματίσιμος πομποδέκτης και έχει προγραμματιστεί να δέχεται εντολές μέσω σειριακής και να στέλνει αντίστοιχη απάντηση. Για αυτό το σκοπό έχει γραφεί πρόγραμμα που χρησιμοποιεί το πρωτόκολλο WiFi. Ζητείται επέκταση του προγράμματος ώστε να μπορεί να γίνει χρήση του πρωτόκολλου επικοινωνίας ESP-NOW για την επικοινωνία ανάμεσα στους πομποδέκτες.

## 3ο Μέρος

Στο μέρος αυτό σκοπός μας είναι η χρήση ενός raspberry pi για τον έλεγχο πρόσβασης σε έναν χώρο (πχ του εργαστηρίου). Για αυτό μπορεί να χρησιμοποιείται μια κάμερα για την ανάγνωση του QR code του πιστοποιητικού εμβολιασμού ή test και στη συνέχεια να γίνεται έλεγχος της εγκυρότητας του εγγράφου μέσω του gov.gr. Μετά την επιτυχή επαλήθευση του εγγράφου θα χρησιμοποιείται ένας αισθητήρας RFID για την ανάγνωση μια RFID κάρτας και θα ελέγχεται μέσα από μια βάση αν το συγκεκριμένο άτομο έχει πρόσβαση στον χώρο. Αν ναι, θα ανοίγεται μέσω ενός σερβομηχανισμού η πόρτα και αν όχι θα χτυπάει ένα κουδούνι στον χώρο (μπορεί να χρησιμοποιηθεί ένα buzzer). Στην περίπτωση που κάποιος δεν έχει κάρτα RFID θα υπάρχει και ένα button που θα ενεργοποιεί απευθείας το κουδούνι.