Δίκτυα Αισθητήρων

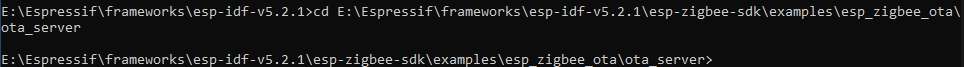
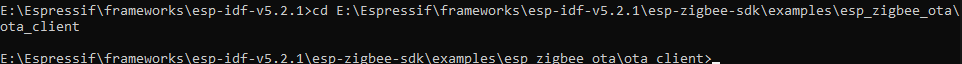
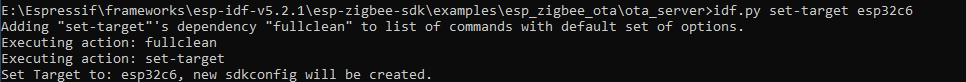
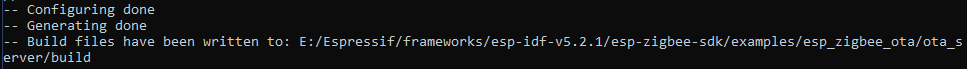
Όνομα: Παναγιώτης

Επίθετο: Τικταπανίδης

1. **Setup Συνδεση και Eπικοινωνια του Client-Server.**
   1. **Setup.**

Αρχικα χρησιμοποιησαμε εναν σταθερό υπολογιστη και ενα φορητο Laptop. Συνδέσαμε στον σταθερο υπολογιστη το ενα esp32c6 και στο λαπτοπ το αλλο.Θεσαμε οτι το esp32c6 που ειναι συνδεδεμενο στον σταθερο υπολογιστή θα ειναι ο Server ενω στο λαπτοπ θα ειναι ο Client.

* 1. **Connection.**
     1. A computer screen with a black screen

        Description automatically generated**Download Espressif-IDE & sdk.**Αποφασισαμε να κατεβασουμε το latest-version(5.3.0) του espressif διοτι οπως θα δειτε και παρακάτω στην αναφορά ηταν το μοναδικο version το οποιο δεν ειχε κανενα προβλημα στις αλλαγες του κωδικα που καναμε για τα ερωτηματα.  
          
        Μετα το περας της εγκαταστασεις του espressif ανοιξαμε το ΕSP-IDE 5.2-CMD(Δεν γνωριζω γιατι το αναφερει ως 5.2 εμεις εχουμε κατεβασει το 5.3)  
        Πριν ξεκινησουμε με τα υπολοιπα επρεπε να κατεβασουμε και στα δυο τον φακελο sdk.Επομενως με βαση τον αρχειο WSN\_Assignment που μας εχει ανατεθει χρησιμοποιησαμε την εντολη :  
          
        **git clone** [**https://github.com/espressif/esp-zigbee-sdk.git**](https://github.com/espressif/esp-zigbee-sdk.git)
     2. **Ρυθμίσεις παραμέτρων Client-Server.**Πριν απο τις ρυθμισεις θα πρεπει να παμε στον καταληλλο φακελο τον οποιον βρισκονται τα αρχεια ωστε να ορισουμε κιολας με αυτον τον τροπο ποιο esp32c6 θα παρει τον ρολο σαν Client και ποιο σαν Server.  
        Η εντολες που χρησιμοποιησαμε ειναι οι:μετα απο αυτο επρεπε να ορισουμε τον στοχο του επεξεργαστη χρησιμοποιωντας την εντολη  
         **idf.py set-target esp32c6**  
        η ιδια διαδικασια εχει γινει και για τον client.  
        Πριν δοκιμασουμε να δουμε εαν ο client και server εχουν επικοινωνια μεταξυ τους θα επρεπε να διαγραψουμε την NVRAM.Αυτη την διαδικασια την εχουμε εκτελεσεις μετα απο καθε αλλαγη που καναμε στον κωδικα ωστε να μην υπαρχει προηγουμενες πληροφοριες.Το command που χρησιμοποιησαμε ηταν το:  
          
          
          
          
        A computer screen shot of a black screen

        Description automatically generatedidf.py -p COM3 erase-flash  
          
          
        για να βρουμε την θεση του port ανατρεξαμε στο device manager για να δουμε σε ποιο port ειναι συνδεδεμενο το esp32c6.
     3. A screenshot of a computer program

        Description automatically generated**Eπικοινωνία.**Για να δουμε την σειριακη εξοδο εκτελεσαμε και οι δυο την εντολη : idf.py -p PORT flash monitor. Εκτελωντας και οι δυο ταυτοχρονα αυτη την εντολη περιμενα να δουμε εαν εχουν επικοινωνια τα δυο esp μεταξυ τους , επομενως θα επρεπε να βλεπουμε πακετα να μεταφερονται κατι που το επιτυχαμε.  
        Εδω εχουμε ηδη κανει σετ το delay μας να ειναι 1 δευτερολεπτο οπως μας ζητηθηκε σε παρακατω ερωτημα.(Στο ερωτημα εξηγω πως βαλαμε το delay.)

1. **Ερωτησεις.**
   1. Τι ακριβως κανει η συναρτηση esp\_zb\_app\_signal\_handler στο client και στον server. Περιγραψτε τις διαδικασιες.  
        
      **Απαντηση: Στο Client:** Η λειτουργια αυτη καλειται καθε φορα που λαμβανεται ενα σημα εφαρμογης Zigbee(πρωτοκολλο ασυρματης επικοινωνιας).Επεξεργαζεται 3 τυπους συματων και την κατασταση.  
      Οι 3 τροποι σηματων ειναι :  
      ESP\_ZB\_BDB\_SIGNAL\_DEVICE\_FIRST\_START / REBOOT / STEERING

Αναλυτικα βηματα στην λειτουργια:  
1. Εξαγει τον τυπο σηματος(sig\_type) και την κατασταση σφαλματος(err\_status)απο την δομη esp\_zb\_app\_singal\_t  
Διαχειριση Switch-Case:  
 1.Επιτυχη σηματα συνδεσης:  
 Καταγραφει πληροφοριες δικτυου αν το σημα υποδεικνυει επιτυχη συνδεση στο δικτυο

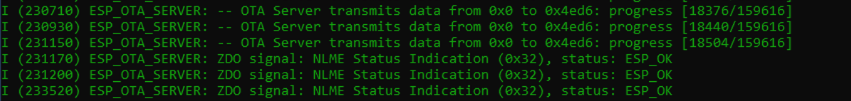
2.Αποτυχημενα σηματα συνδεσης:

Καταγραφει την αποτυχια κατευθυνσης του δικτυου και προγραμματιζει μια ανακληση για να ξεκινησει ξανα την κατευθυνση του δικτυου.  
3.Προεπιλεγμενη περιπτωση:  
Καταγραφει την αποτυχια κατευθυνσης του δικτυου και προγραμματιζει μια ανακληση για να ξεκινησει ξανα την κατευθυνση του δικτυου.  
  
Παρεχει ανατροφοδοτηση και διαχειριζεται επαναληψεις σε περιπτωση αρχικων αποτυχιων διαμορφωσης δικτυου.  
**Στο server:** Στο server η συναρτηση χειριζεται συνηθως αλλα ειδη σηματων που σχετιζονται με τη λειτουργια του συστηματος οπως αιτηματα αναβαθμισης κωδικα η διαχειριση καταχωρισεων συσκευων.Εξυπηρετει την αναγκη για διαχειριση και απαντηση σε μηνυματα του Zigbee στις εφαρμογες,εξασφαλιζοντας ετσι τη σταθερη λειτουργια του δικτυου.

* 1. Τι ακριβώς κάνει η συνάρτηση esp\_zb\_task στον client και στον server. Περιγράψτε τις διαδικασίες.

**Απαντηση:  
Στο Client:   
1.**Δημιουργει και διαμορφώνει την αρχικη ρυθμιση του Zigbee stack ως συσκευη τερματικου(end-device) μεσω της esp\_zb\_init.  
2.Χρησιμοποιει τη δομη esp\_zb\_cfg\_t για να καθορισει τις παραμετρους δικτυου.  
3.Δημιουργει ενα ΟΤΑ client cluster και καθοριζει τις παραμετρους του, οπως η εκδοση αρχειου, ο κωδικος κατασκευαστη και ο τυπος εικονας.  
4.Προσθετει τα αναγκαια attributes στο ΟΤΑ client cluster.  
5.Δημιουργει μια λιστα cluster και προσθετει το OTA client cluster σε αυτην.  
6.Δημιουργει μια λιστα endpoint και προσθετει τη λιστα cluster σε αυτην καθοριζοντας την ταυτοτητα του endpoint.  
7.Eγγραφει τη συσκευη με τη λιστα endpoint στο Zigbee stack.  
8.Εγγραφει εναν χειριστη ενεργειων για τη διαχειριση ενεργειων ΟΤΑ.  
9.Καθοριζει το συνολο καναλιων για το δικτυο.  
10.Ξεκιναει το Zigbee stack   
11.Εισαγει την κυρια loop και διαχειριζεται τη ροη των γεγονοτων και των σηματων και των γεγονοτων.  
**Στο Server:  
1.** Αρχικοποιει το Zigbee stack  
2.Δημιουργει και Διαμορφωνει το ΟΤΑ server Cluster  
3.Διασφαλιζει και διαχειριζει τις συσκευες για την ασφαλη ομαλη λειτουργια του δικτυου.

* 1. A screen shot of a computer screen

     Description automatically generatedA screen shot of a computer code

     Description automatically generatedΓια να προσθεσουμε καθυστερηση 1 δευτερολεπτου στο κωδικα συμβουλευτικα προηγουμενο μαθημα (ΙοΤ) και αυτο που χρειαστηκε να αλλαξω στον κωδικα μου ειναι να βαλω τα παρακατω:  
     και μεσα στην static void esp\_zb\_task(void \*pvParameters) το παρακατω   
     Για την υλοποιηση της ισχυης του σηματος :  
     Πρωτα επρεπε να κανω include το API εαν και δεν χρειαζοταν γιατι εχει γινει ηδη include μεσα στο header file του server.  
     Το επομενω βημα ηταν να παω git\_hub που μας εχει δωθει(ΑPI) και να βρω στην γραμμη 527 το πως γινεται set η tx\_power.Επομενως πηγα στο κωδικα του server και προσθεσα:   
     Αρχικα χωρις απο αυτη την προσθηκη το σημα εφτανε σε εμβελεια περιπου ηταν στα 30 μετρα.Η διαδικασια ηταν ως εξης ο Server ηταν σταθερος και καθε 2 δευτερολεπτα ξεκινουσαμε και πηγαιναμε τον client ενα μετρο πιο μακρια μεχρι να δουμε εαν θα διακοπει το σημα.Το σημα διακοπηκε περιπου απο τα 30-50 μετρα.   
     Οταν ξανα ηρθε πιο κοντα ο client επανηλθε και συνεχισε να στελνει τα πακετα.  
     Ξανατρεξαμε την διαδικασια αλλα αυτη την φορα αλλαξαμε την ισχυ απο το IEEE802154\_TXPOWER\_VALUE\_MΑΧ στο IEEE802154\_TXPOWER\_VALUE\_MIN και απαναλαβαμε την ιδια διαδικασια με πριν.  
     Αυτο που παρατηρησαμε ειναι οτι το σημα χαθηκε σε πολυ εμβελεια το πολυ 10 μετρα και επανηλθε παλι ξανα οταν μπηκαμε ξανα σε αυτη την εμβελεια.  
       
       
     Οι αλλαγες που καναμε στον κωδικα για να επιτυχουμε αυτο ηταν :  
     #define IEEE802154\_TXPOWER\_VALUE\_MIN 1  
     Εμεις δοκιμασαμε να κανουμε αυτο διοτι το VALUE\_MAX δεν ηταν εμφανες σε καποιο header folder που κοιταξαμε και επομενως για να καταφεραμουν να υλοποιησουμε το ερωτημα αποφασισαμε να το κανουμε define στην αρχη για να κανει bypass ολα τα υπολοιπα που δεχεται ωστε να μας δωσει αποτελεσμα κατι το οποιο οντως λειτουργησε.  
       
     Για να προσθεσουμε συναγερμο συμβουλευτηκαμε το API επομενως μπηκαμε στον κωδικα μας και προσθεσαμε την παρακατω void:  
       
     A blue background with yellow text

     Description automatically generated  
       
       
     Μετα απο αυτη την αλλαγη παρατηρησαμε οτι καθε φορα που θα χανετε το σημα θα μας εμφανιζει το μυνημα που εχουμε θεση εμεις .   
       
       
       
     A screen shot of a computer

     Description automatically generated  
       
     Τελος ξανα αλλαξαμε την IEEE802154\_TXPOWER\_VALUE\_MIN 1 και παρατηρησαμε οτι η εμβελεια του σηματος εχει μεγαλωση παλι και αυτη την φορα εφτασε περιπου τα 40 μετρα.
  2. Για να καταφερουμε να υλοποιησουμε την φυγη του client απο το δικτυο το μονο που ειχαμε να κανουμε ειναι να μπουμε μεσα στο header file esp\_ota\_client και να αλλαξουμε την τιμη : #define ED\_AGING\_TIMEOUT ESP\_ZB\_ED\_AGING\_TIMEOUT\_256MIN /\* End device ages time \*/  
       
     αφου καναμε τις αλλαγες ειδαμε οτι ξεκιναει κανονικα να κανει συνδεση και μετα απο 10 δευτερολεπτα σταματαει η συνδεση μεχρι που μετα απο λιγο χρονο ξαναξεκιναει να στελνει κανονικα πακετα   
     A screen shot of a computer

     Description automatically generated