ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΔΙΑΔΙΚΤΥΑΚΟΥ ΡΟΛΟΓΙΟΥ ΜΕ ΧΡΗΣΗ ΤΟΥ ESP32»

Του Φοιτητή
Σκαρλατάκη Γεώργιου
Αρ. Μητρώου: 509558

Επιβλέπων
Γιακουμής Άγγελος
Βαθμίδα: Λέκτορας

Τίτλος Δ.Ε: Σχεδίαση και υλοποίηση διαδικτυακού ρολογιού με χρήση του ESP32

Κωδικός Δ.Ε: 20147

Ονοματεπώνυμο φοιτητή: Σκαρλατάκης Γεώργιος

Ονοματεπώνυμο εισηγητή: Γιακουμής Άγγελος

Ημερομηνία ανάληψης Δ.Ε: 05/05/2020

Ημερομηνία περάτωσης Δ.Ε: 24/06/2020

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Ευχαριστίες

Η παρούσα πτυχιακή εργασία με θέμα «Σχεδίαση διαδικτυακού ρολογιού με χρήση του ESP32» πραγματοποιήθηκε στο πλαίσιο της πτυχιακής εργασίας του τμήματος μηχανικών πληροφορικής και ηλεκτρονικών συστημάτων του Αλεξάνδρειου Τεχνολογικού Εκπαιδευτικού Ιδρύματος το έτος 2020.

Αρχικά θα ήθελα ευχαριστήσω την οικογένεια μου που στάθηκε στο πλευρό μου όλα αυτά τα χρόνια και μου έδωσαν σημαντικά εφόδια (οικονομικά και ψυχολογικά) να γίνω ένας καλός άνθρωπος και με βοήθησαν με κάθε τρόπο σε προβλήματα που συνάντησα κατά την διάρκεια της ζωής μου.

Θα ήθελα να ευχαριστήσω τον καθηγητή του τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων κ. Γιακουμή Άγγελο και τον καθηγητή του τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων κ. Μπάμιο Γεώργιο που μου έδωσαν την δυνατότητα να πραγματοποιήσω την πτυχιακή εργασία καθώς και την καθοδήγηση τους.

Επίσης θα ήθελα να ευχαριστήσω θερμά την κοπέλα μου για την στήριξη της με κάθε τρόπο και την πολύτιμή βοήθεια της καθώς και την υπομονή της για την περάτωση της πτυχιακής εργασίας.

Τέλος θα ήθελα να αφιερώσω την εργασία αυτή στην μνήμη του καλού μου φίλου και συναδέλφου Καριπίδη Γιώργη, που στηρίξαμε και βοηθήσαμε ο ένας τον άλλον κατά την διάρκεια των σπουδών μας, καθώς να εκφράσω και τις ευχαριστίες μου για την στήριξη που μου έδωσε η οικογένεια του.

Πρόλογος

Η πτυχιακή εργασία έχει ως αντικείμενο τον σχεδιασμό και την υλοποίηση ενός διαδικτυακού ρολογιού με την χρήση της ολοκληρωμένης πλακέτας ESP-32. Ο προγραμματισμός του ESP32 θα γίνει με την βοήθεια του προγράμματος ARDUINO IDE. Στην συνέχεια θα λάβει την ώρα από τον NTP (Network Time Protocol) Server με UDP και στην συνέχεια μεταδίδει την ημερομηνία και ώρα σε μια οθόνη LCD 16 * 2.

Η υλοποίηση της εργασίας βασίζεται στον προγραμματισμό του μικροεπεξεργαστή της ESP-WROOM-32, μέσω της αναπτυξιακής πλακέτας ESP32 DEVKIT DOIT V1.Επίσης θα χρησιμοποιηθεί οθόνη LCD 16 * 2 η οποία είναι συνδεδεμένη με I2C module. Στην παρακάτω εργασία θα αναπτυχθεί η διαδικασία μιας τέτοιας κατασκευής, οι δυνατότητες και τα χαρακτηριστικά του μικροεπεξεργαστή.

Περίληψη

Στόχος της πτυχιακή εργασία είναι η ανάπτυξη της ιστορίας του ESP32 και την εταιρία ESPRESSIF SYSTEMS που το κατασκεύασε. Γίνεται αναφορά στις δυνατότητες που έχει το module σε σχέση με το προηγούμενο μοντέλο το module ESP8266. Επίσης αναπτύσσονται οι δυνατότητες από τα τις διάφορες εκδόσεις των modules και τον αναπτυξιακών πλακετών που υπάρχουν με βάση το ESP32. Η κατασκευή του κυκλώματος γίνεται με την αναπτυξιακή πλακέτα ESP32-DEVKIT-V1, ενός I2C module και μιας LCD 16 * 2 οθόνης επίσης χρειάζεται δίκτυο Wi-Fi και η χρήση ηλεκτρονικού υπολογιστή για τον προγραμματισμό της αναπτυξιακής πλακέτας μέσω του προγράμματος Arduino IDE.

Abstract

The aim of this dissertation is to design and prototype a web-based clock utilizing an ESP32 development board and interfacing this with an LCD 16*2 screen. In the first chapter a product review is given in order to introduce the ESP32 development boards with reference to the manufacturing company, ESPRESSIF and its legacy. The reader can appreciate the various modules available along with their capabilities and enhancements of the newer versions in the market. The second chapter is dedicated to the design of the circuit and the specific hardware used for the prototype. Primarily, this hardware consists of the ESP32-DEVKIT-V1 development board, the I2C module and the 16 * 2 LCD screen. Essential for the programming of the ESP32 through the Arduino IDE were a WiFi network and of course the use of a PC. The final chapter describes the basic operation of the device along with potential applications, followed by the bibliography.

Περιεχόμενα

Λίστα Εικόνων

Λίστα πινάκων

ΚΕΦΑΛΑΙΟ 1

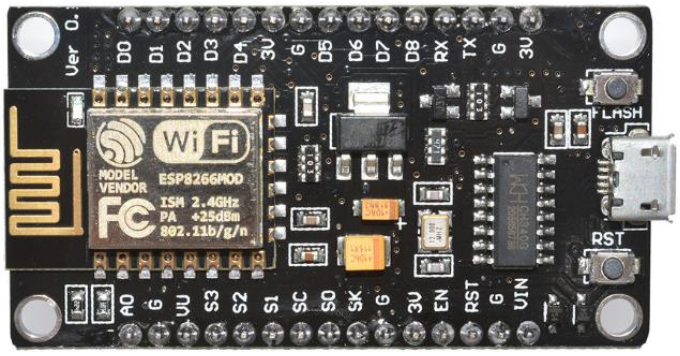
Στο κεφάλαιο αυτό γίνεται αναφορά στην ιστορία του ESP32, στα modules ESP32 όπως και στις αναπτυξιακές πλακέτες ESP32. Επίσης, αναλύονται οι δυνατότητες που έχει ο χρήστης και γίνεται επεξήγηση των pins από το module ESP32-WROOM-32

1.1 Η Ταυτότητα της πλακέτας ESP32

Το ESP32 είναι μια σειρά πλακετών χαμηλού κόστους (low-cost) και χαμηλής ενέργειας (low-power) οι οποίες περιέχουν ολοκληρωμένο κύκλωμα κεραίας Wi-Fi καθώς και dual-mode BLUETOOTH (BLUETOOTH CLASSIC και BLUETOOTH LOW ENERGY αλλιώς BLE). Ένα από τα βασικότερα πλεονεκτήματα του ESP32 είναι το γεγονός πως πρόκειται για ένα σύστημα ανοιχτού κώδικα και μπορεί να προγραμματιστεί σε αρκετές γλώσσες προγραμματισμού όπως C++, Python, JavaScript κλπ. Η κάθε γλώσσα περιέχει την αντίστοιχη βιβλιοθήκη και με την βοήθεια Η/Υ μπορεί να περαστεί ο κώδικας στο ESP32.

1.2 Ιστορία του ESP32

Τον Δεκέμβριο του 2013 η Εταιρεία ESPRESSIF SYSTEMS έβγαλε στην αγορά το πρώτο low-cost 2,4 GHz Wi-Fi SoC (System on a Chip) για tablet και set-top box (η αλλιώς tv-box). Μόλις σε διάστημα έξι μηνών από τον προηγούμενο SoC η εταιρεία έβγαλε στην αγορά το ESP8266 το οποίο ήταν ένα ολοκληρωμένο κύκλωμα βασισμένο στον επεξεργαστή L106 32-bit της εταιρίας Tensilica. Το ESP8266 χαρακτηρίζεται από πρωτοποριακά χαρακτηριστικά όπως το ότι μπορεί να χρησιμοποιήσει άμεσα το Wi-Fi με χαμηλό κόστος ενέργειας



Εικόνα 1.1 Ολοκληρωμένη πλακέτα ESP8266

Τον Σεπτέμβριο του 2016 η ESPRESSIF SYSTEMS κυκλοφορεί στην αγορά το ESP32 SoC το οποίο βασίζεται στον επεξεργαστή Xtensa LX6 microprocessor αλλά και σε ένα ULP (Ultra Low Power) co-processor.Ο ESP32 σε σχέση με το ESP8266 διαθέτει πιο γρήγορο επεξεργαστή (σε κάποιες εκδόσεις είναι διπύρηνος), BLUETOOTH και επίσης πολλούς νέους τύπους ολοκληρωμένων πλακετών.

Εικόνα 1.2 Ολοκληρωμένη πλακέτα ESP32 DEVKIT DOIT V1

1.3 Modules ESP32

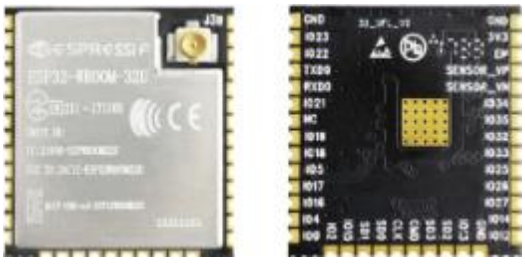
Η σειρά ESP32 κυκλοφορεί σε πολλές εκδόσεις και πλακέτες από τις οποίες η κάθε μια έχει και διαφορετικές δυνατότητες. Για αυτόν τον σκοπό έχουν δημιουργηθεί διαφορετικά modules ώστε να καλύψουν τις ανάγκες της κάθε πλακέτας.

ESP32-WROOM-32: Είναι το βασικό και το πιο κοινό module της οικογένειας από την σειρά ESP32. Επίσης ήταν το πρώτο που κυκλοφόρησε στην αγορά πάνω σε πλακέτες ESP32-DEVKIT και σε πλακέτες ESP32-WROVER-KIT V1 και V2.

Εικόνα 1.3-Module ESP32-WROOM-32

ESP32-WROOM-32D/ESP32-WROOM-32U: Ο σχεδιασμός αυτών των δύο modules είναι μικρότερος σε σχέση με αυτόν του ESP32-WROOM-32 με σκοπό να έχουν μεγαλύτερη flash memory και πιο δυνατή κεραία RF.

Εικόνα 1.4-Module ESP32-WROOM-32D



Εικόνα 1.5-Module ESP32-WROOM-32U

ESP32-SOLO-1: Το module αυτό είναι βασισμένο στο ESP32-WROOM-32D με τη μόνη διαφορα ότι έχει έναν πυρήνα στον επεξεργαστή και φτάνει εως τα 160Mhz.

Εικόνα 1.6-ModuleESP32-SOLO-1

ESP32-WROVER: Αυτή η σειρα από modules αναπτύχθηκε για να καλύψη τις ανάγκες της σειρας ESP-WROVER-KIT V3 και ESP-WROVER-KIT V4 αντικαθιστώντας το module του ESP32-WROOM-32 το όποιο χρησιμοποιούνταν για τα ESP-WROOVER KIT V1 και ESP-WROOVER KIT V2.

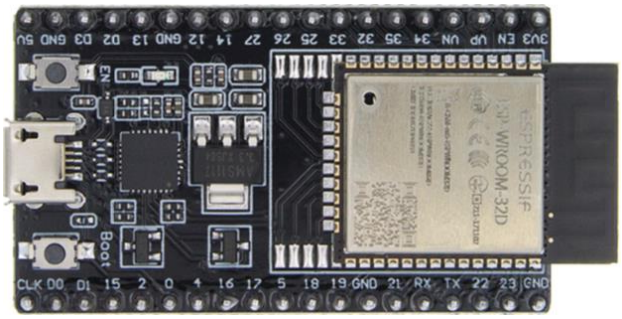
Εικόνα 1.7-Module ESP32-WROVER

1.4 Αναπτυξιακές πλακέτες ESP32

Οι σειρές των modules που προαναφερθήκαν προκειμένου να είναι εύκολες στη χρήση έχουν τοποθετηθεί πάνω σε αναπτυξιακές πλακέτες. Οι πλακέτες ESP32 έχουν αναπτυχθεί ανάλογα με τις δυνατότητες του κάθε module και μπορούν να καλύψουν διάφορες ανάγκες του χρήστη είτε αυτές είναι σε φυσικό μέγεθος της πλακέτας, είτε έχουν την δυνατότητα πολλών εισόδων-εξόδων, αλλά ακόμα και να έχουν επάνω τους κάμερα ή οθόνη.

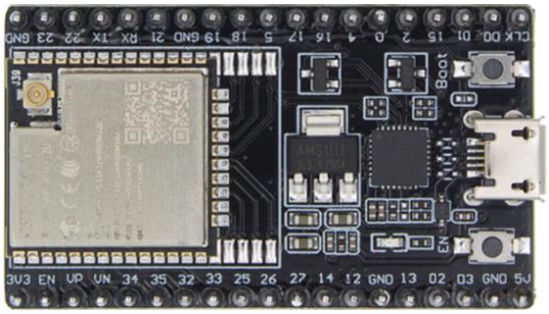
ESP32-DEVKIT-V4/32D: Αυτές οι δύο πλακέτες είναι οι πιο κοινές από τις σειρές των ESP32. Οι πλακέτες αυτές καλύπτουν επαρκώς όλες τις βασικές ανάγκες και είναι εύκολες στην χρήση. Η βασικότερη διαφορά των ESP32-DEVKIT-V4 και ESP32-DEVKITC-32D είναι τα modules. Πιο συγκεκριμένα η πρώτη είναι κατασκευασμένη με το ESP32-WROOM-32 και η δεύτερη με το ESP32-WROOM-32D.

Εικόνα 1.8Πλακέτα ESP32 DEVKIT V4



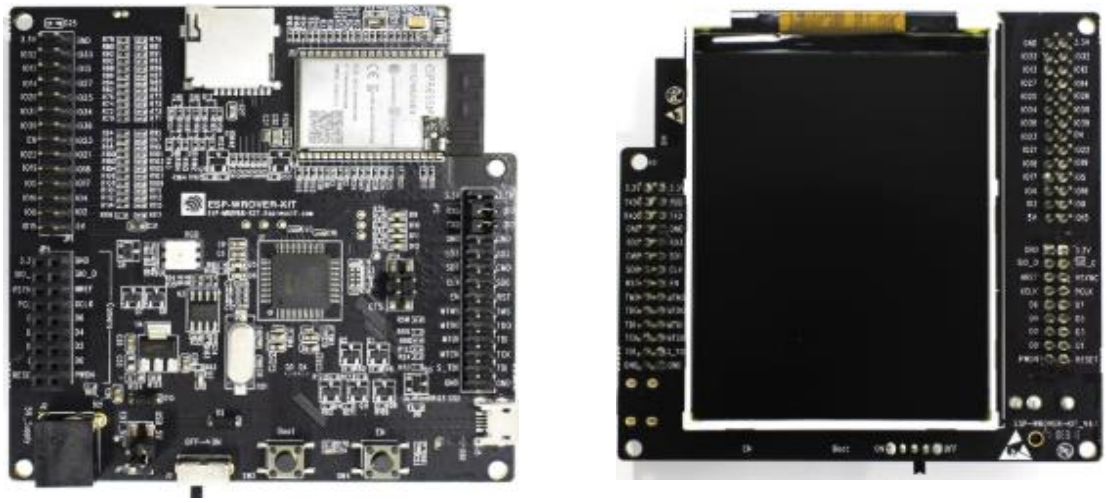
Εικόνα 1.9-Πλακέτα ESP32-DEVKITC-32D.

ESP32-DEVKITC-32U: Η Πλακέτα αυτή είναι μια μικρότερη εκδοχή της ESP32-DEVKITC-32D χωρίς το κύκλωμα της κεραίας. Αντί αυτού μπορεί να συνδεθεί εξωτερικού τύπου κεραία στην υποδοχή.



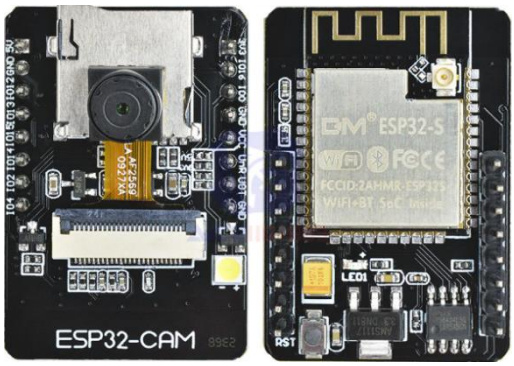
Εικόνα 1.10-ESP32-DEVKITC-32U.

ESP32-WROVER-KIT-V4.1:Η συγκεκριμένη πλακέτα χρησιμοποιεί το module ESP32-WROVER το οποίο τις δίνει περισσότερες δυνατότητες σε σχέση με τις υπόλοιπες σειρές καθώς έχει περισσότερες εξόδους-εισόδους, μεγαλύτερη μνήμη (memory flash) και επίσης οθόνη 3.2 ιντσών στο πίσω μέρος της πλακέτας.



Εικόνα 1.11-Πλακέτα ESP32-WROVER-KIT-V4.1

ESP32-CAM:Η πλακέτα αυτή είναι υλοποιημένη με το ESP32-SOLO-1 module και τα στοιχεία που την καθιστούν διαφορετική από τις υπόλοιπες πλακέτες είναι το γεγονός πως περιέχει κάμερα OV2640, θύρα για κάρτα SD ώστε να μπορεί ο χρήστης να αποθηκεύει φωτογραφίες, βίντεο και τέλος λίγες θύρες εισόδων-εξόδων για να μπορεί ο χρήστης να συνδέσει περιφερειακές συσκευές.



Εικόνα 1.12Πλακέτα ESP32-CAM

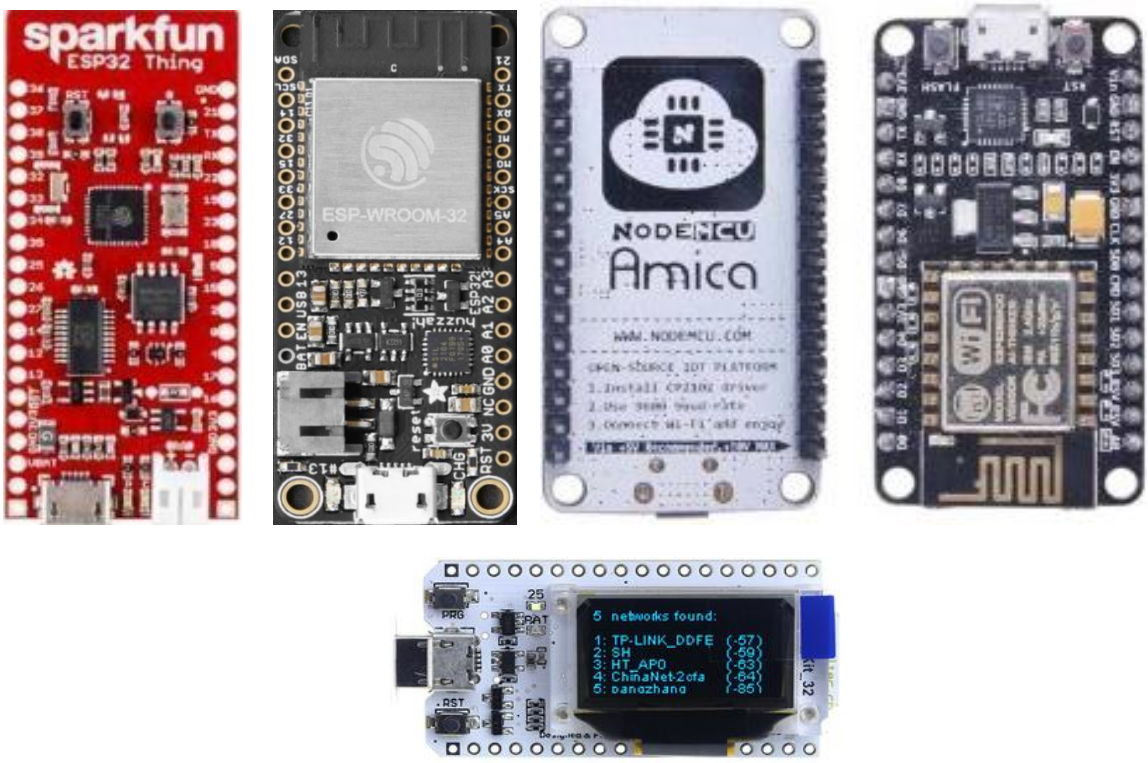
Φυσικά υπάρχουν εταιρείες οι οποίες αγοράζουν τα modules από την ESPRESSIF SYSTEMS με σκοπό να κατασκευάζουν δικές τους αναπτυξιακές πλακέτες σύμφωνα με τα πρότυπα της ESPRESSIF SYSTEMS ή ακόμα και με δικές τους πατέντες. Μερικές από αυτές τις εταιρίες είναι η SPARKFUN, ADAFRUIT, LOLIN, NODEMCU.Παρακάτω εμφανίζονται μερικές από τις πλακέτες που κατασκευάζουν οι συγκεκριμένες εταιρείες.

1.5 Πλεονεκτήματα του ESP32.

Στόχος της εταιρείας ESPRESSIF SYSTEMS είναι να κατασκευάζει πρωτότυπο εξοπλισμό που αφορά Wi-Fi, Bluetooth τα οποία ταυτόχρονα είναι φιλικά προς το περιβάλλον, οικονομικά και έχουν επίσης χαμηλό κόστος ενέργειας.

Σημαντική είναι η συνεισφορά και άλλων εταιρειών στην κατασκευή αναπτυξιακών πλακετών όπως:

- Η Netduino N3 από την εταιρία Wildness Labs.
- Η Photon από την εταιρία Particle.
- Η Feather HUZZAH από την εταιρία Adafruit.
- Η RedBoard Artemis από την εταιρία Sparkfun.
- Και η EFM32 Wonder Gecko από την εταιρία Silicon Labs.



Εικόνα 1.13-Διάφορες ESP32 πλακέτες από τρίτους κατασκευαστές

Αυτές οι αναπτυξιακές πλακέτες όπως και η ESP32-DEVKIT V4 έχουν ένα βασικό κοινό χαρακτηριστικό, το ότι είναι έτοιμες για χρήση καθώς δεν χρειάζονται εξωτερικά εξαρτήματα. Το μόνο που χρειάζεται είναι να συνδεθούν με τον ηλεκτρονικό υπολογιστή και τον κώδικα. Οι δυνατότητες του ESP32 σε συνδυασμό με την χαμηλή τιμή του τον κάνει να ξεχωρίζει από τα άλλα διότι περιέχει:

- Wi Fi.
- Dual Bluetooth .

- Αισθητήρα Θερμοκρασίας.
- Αισθητήρα μαγνητικού πεδίου.
- Και κεραία RF.

Πίνακας 1.1-Χαρακτηριστικά του ESP32

Κατηγορίες	Στοιχεία	Προδιαγραφές
Πιστοποιήσεις	Πιστοποίηση RF	FCC/CE-RED/IC/TELEC/KCC/SRRC/NCC
	Πιστοποίηση Wi-Fi	Wi-Fi- Alliance
	Certicitation Bluetooth	BQB
	Πράσινη πιστοποίηση	RoHS/REACH
Δοκιμή	Αξιοπιστία	HTOL/HTSL/uHAST/TCT/ESD
Wi-Fi	Πρωτόκολλα	802,11 b/g/n (802.11n έως 150 Mbps)
		Συγκέντρωση A-MPDU και A-MSDU
	Εύρος συχνοτήτων	2,4 GHz x 2,5 GHz
Bluetooth	Πρωτόκολλα	Προδιαγραφές Bluetooth v4.2 BR/EDR και BLE
	Ραδιόφωνο	Δέκτης NZIF με ευαισθησία –97 dBm
		Πομπός κλάσης 1, κατηγορίας 2 και κλάσης 3
		AFH
	Ήχου	CVSD και SBC
Hardware	Διασυνδέσεις λειτουργικής μονάδας	Κάρτα SD, uART, SPI, SDIO, I2C, LED PWM, Μηχανή PWM, I2S, IR, μετρητής σφυγμού, GPIO, αισθητήρας αφής, ADC, DAC
	Αισθητήρας on-chip	Hall sensor
	Ενσωματωμένος κρύσταλλος	40 MHz κρύσταλλος ταλάντωσης
	Integrated SPI flash	4 MB
	Τάση λειτουργίας/τροφοδοτικό	3,0 V ~ 3,6 V
	Ρεύμα λειτουργίας	80 mA
	Ελάχιστο ρεύμα από την παροχή	500 mA
	Συνιστόμενο εύρος θερμοκρασίας λειτουργίας	–40 °C ~ +85 °C
	Μέγεθος κατασκευής	(18,00 ± 0,10) mm × (25,50 ± 0,10) × mm (3,10 ± 0,10) mm
	Επίπεδο ευαισθησίας υγρασίας (MSL)	Επίπεδο 3

Πέρα από τις δυνατότητες που αναφέρθηκαν στον πίνακα-1 σημαντικό είναι να αναφερθεί ότι το ESP32

- Διαθέτει κρύσταλλο ταλάντωσης στα 40MHz.
- Η τάση λειτουργίας του είναι από 3V έως τα 3,6V.
- Το ρεύμα λειτουργίας είναι 80mA.
- Και τέλος η ευαισθησία υγρασίας το οποίο μπορεί να λειτουργεί το ESP32 είναι το επίπεδο 3 το οποίο αντιστοιχεί έως 168 ώρες στους 30 °C με 60%RH(

1.5.1 Επεξεργαστής

Ο επεξεργαστής του ESP32 αποτελείται από ένα ή δύο χαμηλής ενέργειας Xtesna 32-bit LX6 μικροεπεξεργαστών με τα παρακάτω χαρακτηριστικά:

- Οι διπύρηνοι επεξεργαστές φτάνουν τα 240GHz ενώ ο επεξεργαστής με έναν πυρήνα φτάνει έως 160GHz.
- Υποστηρίζει FPU(Floating Point Unit), μονάδα κινητής υποδιαστολής που χρησιμεύει για αριθμητικές πράξεις.
- Υποστηρίζει DSP (Digital Signal Processor) για την ανάλυση ψηφιακού σήματος.

Επίσης οι επεξεργαστές αυτοί διαθέτουν:

- Xtesna RAM/ROM κώδικα προστασίας για την χρήση του ESP32.
- Xtesna Local Memory το οποίο δίνει στον χρήστη εύκολη πρόσβαση στον εντοπισμό περιφερειακών συσκευών.
- JTAG λειτουργία η οποία βοηθάει για τον εντοπισμό των σφαλμάτων.

1.5.2Εσωτερική Μνήμη

Η εσωτερική μνήμη του ESP32 περιέχει:

- 448 KB ROM για εκκίνηση και βασικές λειτουργίες του συστήματος
- 520 KB on-chip SRAM για δεδομένα και οδηγίες
- 8 KB SRAM σε RTC, το οποίο ονομάζεται RTC FAST Memory και μπορεί να χρησιμοποιηθεί για αποθήκευση δεδομένων. Έχει πρόσβαση από την κύρια CPU κατά την εκκίνηση RTC από τη λειτουργία Deep-sleep.
- 8 KB SRAM σε RTC, το οποίο ονομάζεται RTC SLOW Memory και μπορεί να προσεγγιστεί από τον co-processor κατά τη διάρκεια της λειτουργία Deep-sleep.
- 1 Kbit του eFuse: 256 bit χρησιμοποιούνται για το σύστημα (διεύθυνση MAC και διαμόρφωση chip) και τα υπόλοιπα 768 bits προορίζονται για εφαρμογές πελατών, όπως κρυπτογράφηση flash και chip-ID.

1.5.3Μνήμη Flash και SRAM

Στην flash μνήμη αποθηκεύεται ο κώδικας του χρήστη. Πρέπει να σημειωθεί ότι αυτή η μνήμη είναι κομμάτι του επεξεργαστή και λειτουργεί σαν ROM (read only memory)κατά την εκτέλεση των εντολών. Ο χώρος της εξωτερικής μνήμης είναι 16MB.

Η SRAM βρίσκεται και αυτή μέσα στον επεξεργαστή και μπορεί να αποθηκεύσει έως 8MB. Η βασική της λειτουργία είναι να αποθηκεύει δεδομένα που μπορεί να συλλέξει από τον κώδικα του χρήστη.

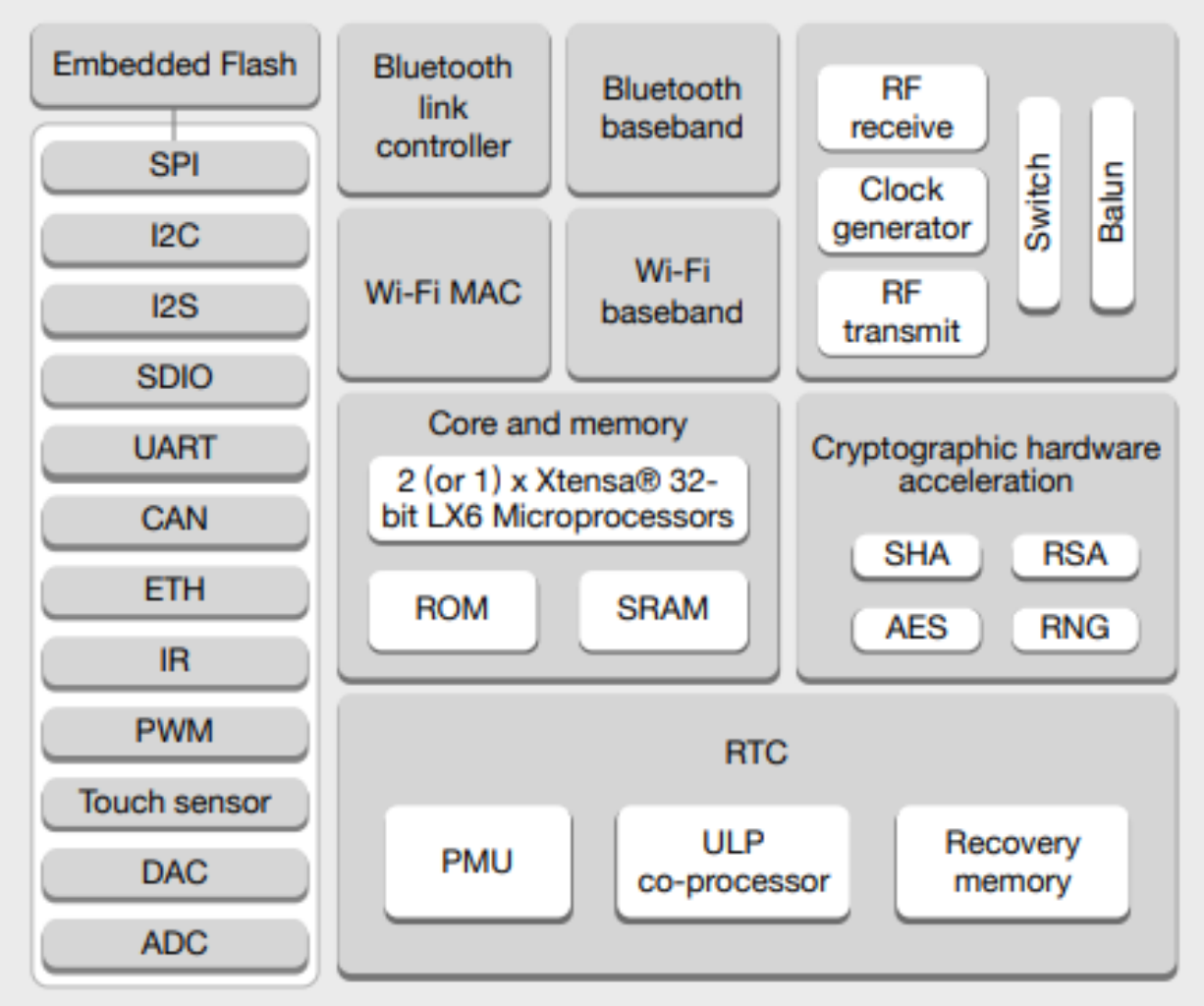
Ακόμα ένα πλεονέκτημα του ESP32 είναι ότι έχει δυνατότητά διασύνδεσης με άλλες συσκευές όπως:

- Οθόνες
- Ηχεία
- Κάρτες αποθήκευσης SD όπου δίνεται στον χρήστη να αποθηκεύσει δεδομένα ή και να τα αναπαράγει αυτά τα δεδομένα σε οθόνη η ηχείο
- LED, LED PWM (Pulse Width Modulation)
- IR
- Μοτέρ PWM
- Αισθητήρας σφυγμού
- Αισθητήρας αφής

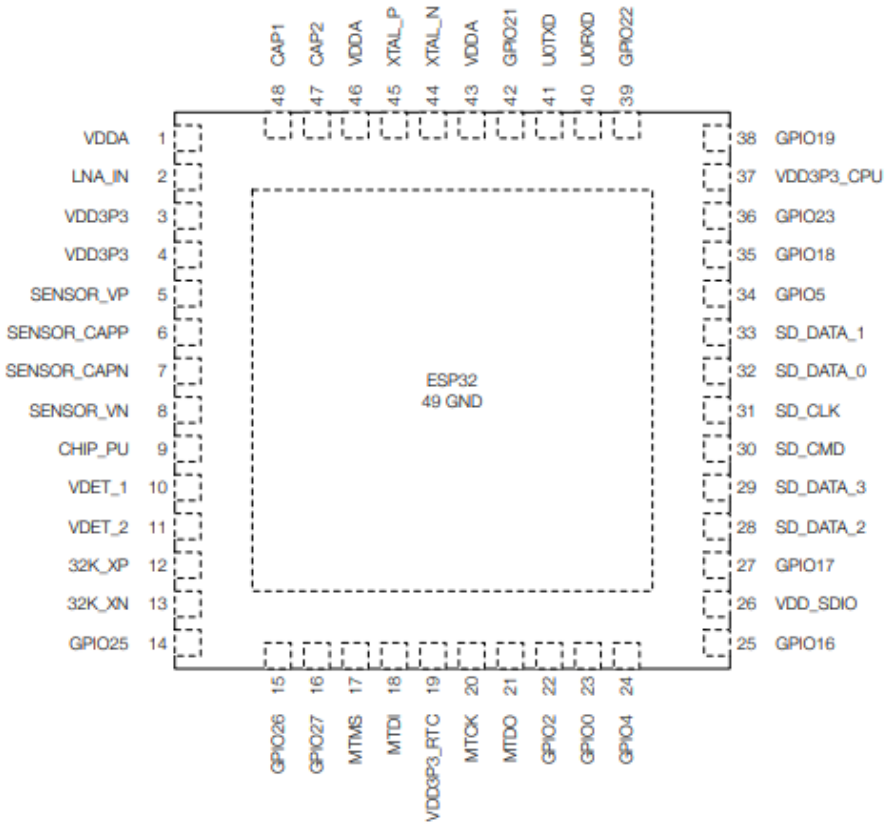
1.5.4 Επιπλέον δυνατότητες:

- SDIO (Secure Digital Input Output) αυτή η λειτουργία είναι ουσιαστικά μια ασφαλής επικοινωνία μεταξύ δύο συσκευών, από το master στο slave.
- uART (universal Asynchronous Receiver-Transmitter) πρόκειται για έναν τυπικό τρόπο διασύνδεσης δύο συσκευών μεταξύ τους εκτός δικτύου. Δεν χρειάζεται να γίνει κάποιος συγχρονισμός για την επικοινωνία μεταξύ των συσκευών. Μπορεί να συνδεθεί με ηλεκτρονικούς υπολογιστές.
- I2C (Inter-integrated Circuit) η διασύνδεση αυτή αφορά modules μόνο και όχι ηλεκτρονικούς υπολογιστές. Είναι αμφίδρομη η επικοινωνία μεταξύ των συσκευών αλλά μόνο μία από τις δύο μπορεί να στέλνει πληροφορίες την κάθε φορά. Επίσης χρειάζεται να γίνει συγχρονισμός μεταξύ των συσκευών για την επικοινωνία και μόνο η κύρια πλακέτα δίνει εντολές για παράδοση και παραλαβή πληροφοριών.
- SPI (Serial Peripheral Interface) η χρήση του και η λειτουργεία του είναι παρόμοια με αυτή του I2C με τη διαφορά ότι έχει την δυνατότητα να δώσει και να λάβει πληροφορίες ταυτόχρονα Έχει επίσης μεγαλύτερες ταχύτητες σε σχέση με το I2C. Η χρήση του είναι σε εφαρμογές οι οποίες αλλάζουν συνέχεια τα δεδομένα όπως ένα θερμόμετρο ή συσκευή μέτρησης πίεσης.
- I2S (Inter-integrated Circuit Sound) με αυτόν τον δίαυλο επικοινωνίας μπορεί ο χρήστης να συνδέσει συσκευές μετάδοσης ήχου (ηχεία) και να αναπαράγει ένα σήμα.
- GPIO (General Purpose Input / Output) είναι πύλες γενικής χρήσης.

Τέλος, διαθέτει την δυνατότητα να μετατρέπει τα σήματα ADC (Analog to Digital Converter) και DAC (Digital to Analog Converter).



Εικόνα 1.14-Διάγραμμα του ESP32



Εικόνα 1.15-Διάταξη των pin του ESP32

Πίνακας 1.2-Pin του ESP32

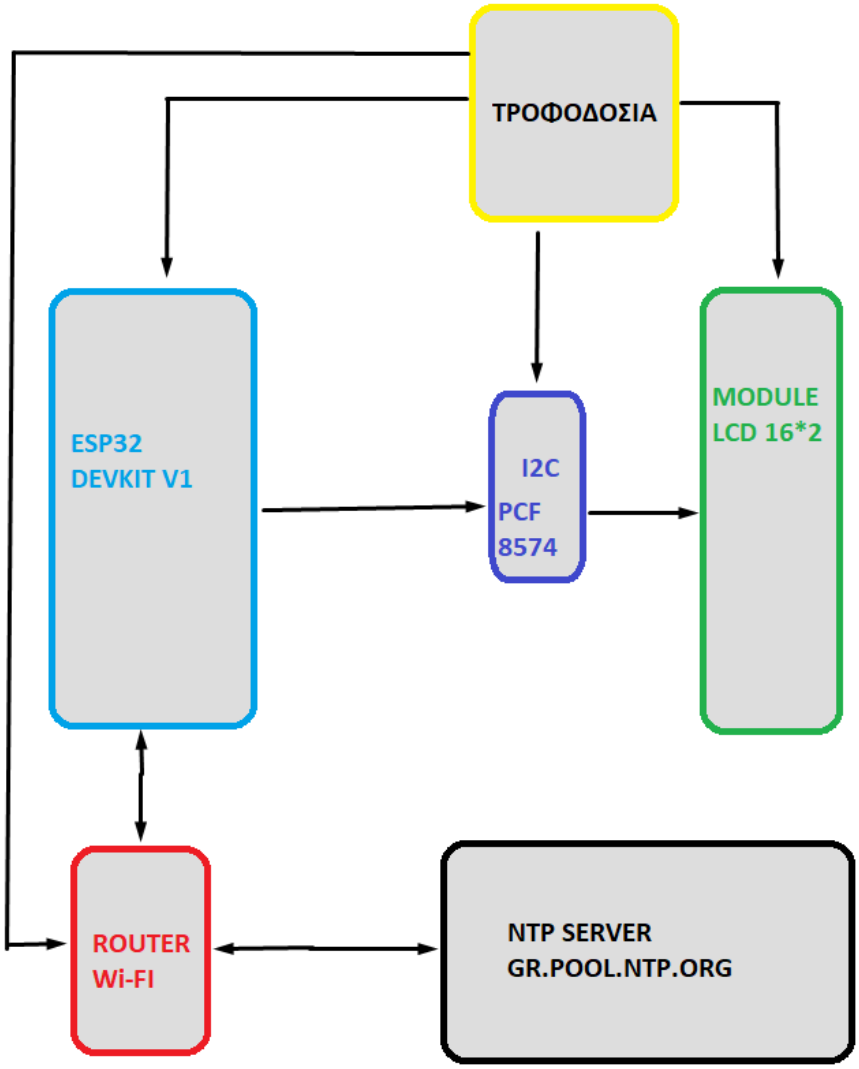
Όνομα	No.	Τύπος	Λειτουργια
Αναλωγικά Pins			
VDDA	1	P	Analog power supply (2.3 V ~ 3.6 V)
LNA_IN	2	I/O	RF input and output
VDD3P3	3	P	Analog power supply (2.3 V ~ 3.6 V
VDD3P3	4	P	Analog power supply (2.3 V ~ 3.6 V)
VDD3p3_RTC			
SENSOR_VP	5	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_CAPP	6	I	GPIO37, ADC1_CH1, RTC_GPIO1
SENSOR_CAPN	7	I	GPIO38, ADC1_CH2, RTC_GPIO2
SENSOR_VN	8	I	GPIO39, ADC1_CH3, RTC_GPIO3
CHIP_PU	9	I	High: On; enables the chip Low: Off; the chip powers off

			Note: Do not leave the CHIP_PU pin floating.
VDET_1	10	I	GPIO34, ADC1_CH6, RTC_GPIO4
VDET_2	11	I	GPIO35, ADC1_CH7, RTC_GPIO5
32K_XP	12	I/O	GPIO32, ADC1_CH4, RTC_GPIO9, TOUCH9, 32K_XP (32.768 kHz crystal oscillator input)
32K_XN	13	I/O	GPIO33, ADC1_CH5, RTC_GPIO8, TOUCH8, 32K_XN (32.768 kHz crystal oscillator output)
GPIO25	14	I/O	GPIO25, ADC2_CH8, RTC_GPIO6, DAC_1, EMAC_RXD0
GPIO26	15	I/O	GPIO26, ADC2_CH9, RTC_GPIO7, DAC_2, EMAC_RXD1
GPIO27	16	I/O	GPIO27, ADC2_CH7, RTC_GPIO17, TOUCH7, EMAC_RX_DV
MTMS	17	I/O	GPIO14, ADC2_CH6, RTC_GPIO16, TOUCH6, EMAC_TXD2, HSPICLK, HS2_CLK, SD_CLK, MTMS
MTDI	18	I/O	GPIO12, ADC2_CH5, RTC_GPIO15, TOUCH5, EMAC_TXD3, HSPIQ, HS2_DATA2, SD_DATA2, MTDI
VDD3P3_RTC	19	P	Input power supply for RTC IO (2.3 V ~ 3.6 V)
MTCK	20	I/O	GPIO13, ADC2_CH4, RTC_GPIO14, TOUCH4, EMAC_RX_ER, HSPID, HS2_DATA3, SD_DATA3, MTCK
MTDO	21	I/O	GPIO15, ADC2_CH3, RTC_GPIO13, TOUCH3, EMAC_RXD3, HSPICS0, HS2_CMD, SD_CMD, MTDO
GPIO2	22	I/O	GPIO2, ADC2_CH2, RTC_GPIO12, TOUCH2, HSPIWP, HS2_DATA0, SD_DATA0
GPIO0	23	I/O	GPIO0, ADC2_CH1, RTC_GPIO11, TOUCH1, EMAC_TX_CLK, CLK_OUT1
GPIO4	24	I/O	GPIO4, ADC2_CH0, RTC_GPIO10, TOUCH0, EMAC_TX_ER, HSPIHD, HS2_DATA1, SD_DATA1
VDD_SDIO			
GPIO16	25	P	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
VDD_SDIO	26	I/O	Output power supply: 1.8 V or the same voltage as VDD3P3_RTC
GPIO17	27	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
SD_DATA_2	28	I/O	GPIO9, HS1_DATA2, U1RXD, SD_DATA2, SPIHD
SD_DATA_3	29	I/O	GPIO10, HS1_DATA3, U1TXD, SD_DATA3, SPIWP
SD_CMD	30	I/O	GPIO11, HS1_CMD, U1RTS, SD_CMD, SPICS0
SD_CLK	31	I/O	GPIO6, HS1_CLK, U1CTS, SD_CLK, SPICLK
SD_DATA_0	32	I/O	GPIO7, HS1_DATA0, U2RTS, SD_DATA0, SPIQ
SD_DATA_1	33	I/O	GPIO8, HS1_DATA1, U2CTS, SD_DATA1, SPID
VDD3P3_CPU			
GPIO5	34	I/O	GPIO5, HS1_DATA6, VSPICS0, EMAC_RX_CLK
GPIO18	35	I/O	GPIO18, HS1_DATA7, VSPICLK

GPIO23	36	I/O	GPIO23, HS1_STROBE, VSPID
VDD3P3_CPU	37	P	Input power supply for CPU IO (1.8 V ~ 3.6 V)
GPIO19	38	I/O	GPIO19, U0CTS, VSPIQ, EMAC_TXD0
GPIO22	39	I/O	GPIO22, U0RTS, VSPIWP, EMAC_TXD1
U0RXD	40	I/O	GPIO3, U0RXD, CLK_OUT2
U0TXD	41	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
GPIO21	42	I/O	GPIO21, VSPIHD, EMAC_TX_EN
Αναλωγικά Pins			
VDDA	43	P	Analog power supply (2.3 V ~ 3.6 V)
VDDA	44	O	External crystal output
XTAL_P	45	I	External crystal input
VDDA	46	P	Analog power supply (2.3 V ~ 3.6 V)
CAP2	47	I	Connects to a 3 nF capacitor and 20 k Ω resistor in parallel to CAP1
CAP1	48	I	Connects to a 10 nF series capacitor to ground
GND	49	P	Ground

ΚΕΦΑΛΑΙΟ 2

Η πτυχιακή εργασία έχει σαν στόχο την κατασκευή ενός ηλεκτρονικού κυκλώματος με βάση την αναπτυξιακή πλακέτα ESP32-DEVKIT-V1. Το ESP32 συνδέεται με το δίκτυο του Wi-Fi και στην συνέχεια ‘’ζητάει’’ από το port 123 του NTP server την ώρα. Ο NTP Servers στέλνει με πρωτόκολλο UDP την ώρα και στην συνέχει με το module I2C την απεικονίζει στην οθόνη LCD 16 * 2.



Εικόνα 16-Διάγραμμα ροής κυκλώματος

2.1 Ανάλυση περιφερικών υλικών

Για την υλοποίηση του κυκλώματος χρειάζονται ένα module 16*2 LCD (Liquid Crystal Display) που απεικονίζει το σήμα που λαμβάνει και ένα module I2C το οποίο λειτουργεί ως αποκωδικοποιητής σήματος που λαμβάνει από το ESP32 και το μεταδίδει στην οθόνη LCD.

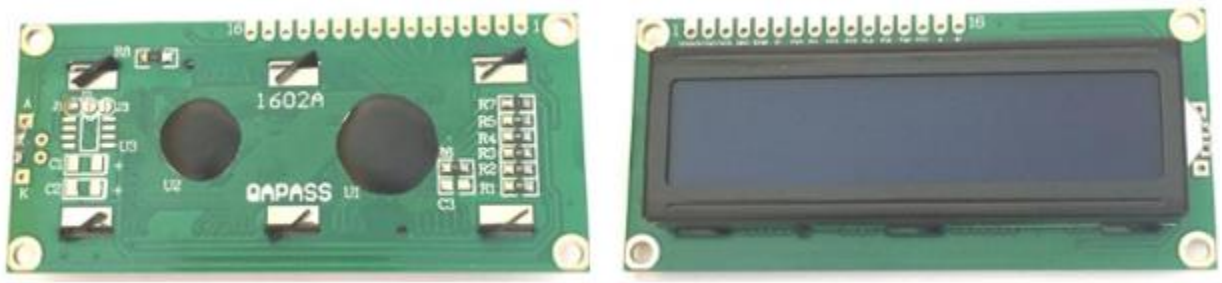
2.1.1 Module 16*2 LCD:

Τα modules αυτά χρησιμοποιούνται σε πολλές εφαρμογές, η πιο κοινή εφαρμογή που μπορεί να τα βρει κανείς είναι πάνω σε κομπιουτεράκια αριθμητικών πράξεων. Έχουν πολλές εκδόσεις όπως 8*1, 8*2, 10*2 κλπ. Το όνομα της έκδοσης του module αναφέρεται στον αριθμό των στηλών και σειρών που περιέχει η οθόνη. Ο πρώτος αριθμός αντιστοιχεί στις στήλες και ο δεύτερος στις σειρές. Για την πτυχιακή εργασία θα χρησιμοποιηθεί το module 16*2 LCD το οποίο χωράει 32 χαρακτήρες. Ο κάθε χαρακτήρας αποτελείται από 5 * 8 pixels. Στην εικόνα 2.2 αναπαρίσταται ένας χαρακτήρας του module.

Εικόνα 17-χαρακτήρας από module LCD

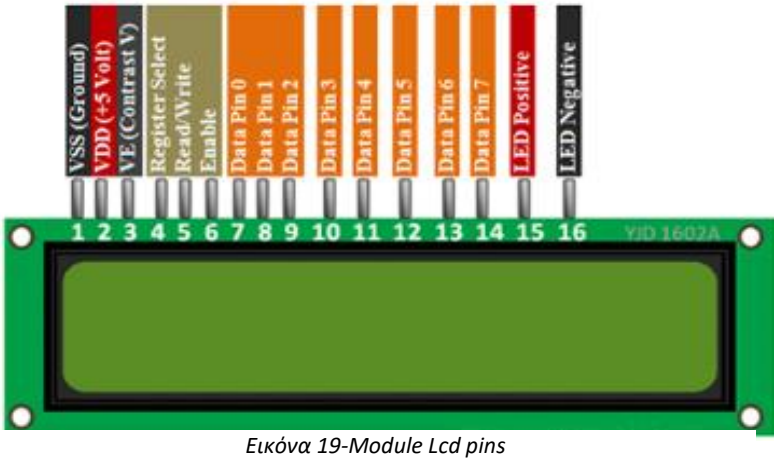
Ο χρήστης έχει την δυνατότητα μέσω του κώδικα να ενεργοποιήσει και να απενεργοποιήσει τα pixels του χαρακτήρα με απλές εντολές στον κώδικα. Για να μπορέσει ο χρήστης να δώσει στην οθόνη το σήμα και αυτή να το απεικονίσει αυτό που θέλει, χρησιμοποιείται το controller HD44780U το οποίο βρίσκεται στο πίσω μέρος του module στην θέση U1 όπως φαίνεται στην εικόνα 2.3. Τα χαρακτηριστικά του 16*2 LCD module:

- Η τάση λειτουργίας ξεκινάει από τα 4,7V έως τα 5,3V
- Η κατανάλωση ρεύματος είναι 1mA χωρίς τον οπίσθιο φωτισμό (backlight).
- Έχει την δυνατότητα εμφάνισης των αλφαριθμητικών χαρακτήρων.
- Μπορεί να εκτυπώσει 16 χαρακτήρες στην κάθε σειρά.
- Ο κάθε χαρακτήρας αποτελείται από 40 pixels (5 στήλες * 8 σειρές).



*Εικόνα 18-Module 16*2 LCD*

- Διατίθεται και με οπίσθιο φωτισμό χρώματος (backlight) πράσινο και μπλε.



Εικόνα 19-Module Lcd pins

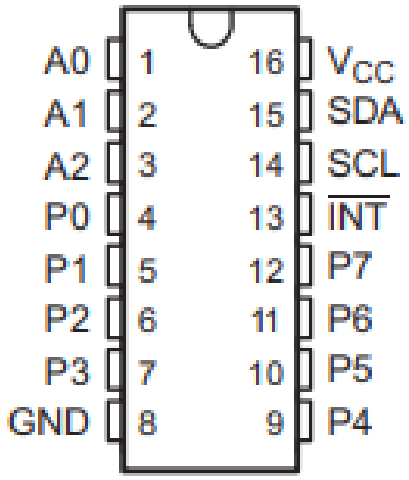
*Πίνακας 2.1-Pins 16*2 LCD Module*

Pin No.	Όνομα pin	Λειτουργεία
1	Vss (Ground)	GND (0V)
2	Vdd (+5 Volt)	Τάση τροφοδοσίας 5V.
3	VE (Contrast V)	Ρύθμιση της αντίθεσης. Με χρήση ενός ποτενσιόμετρου μπορούμε να ρυθμίσουμε την αντίθεση της οθόνης.
4	Register Select	
5	Read/Write	GND (0V) για να γράψουμε στην οθόνη. 5V για να διαβάσει από την οθόνη.
6	Enable	
7	Data Pin 0	8bit data pins
8	Data Pin 1	
9	Data Pin 2	
10	Data Pin 3	
11	Data Pin 4	
12	Data Pin 5	
13	Data Pin 6	
14	Data Pin 7	
15	LED Positive	Backlight VCC (5V)
16	LED Negative	Backlight GND (0V)

2.1.2 Module I2C:

Η κατασκευή του module I2C βασίζεται στο ολοκληρωμένο PCF8574. Το module I2C λειτουργεί σαν δίαυλος επικοινωνίας μεταξύ συσκευών οι οποίες έχουν τουλάχιστον ένα master και ένα slave. Θεωρητικά το I2C μπορεί να έχει ταυτόχρονα 128 συσκευές συνδεδεμένες με 7-bit address ή 1024 συσκευές όταν χρησιμοποιείται 10-bit address. Έτσι, η κάθε συσκευή έχει την δικιά της address (διεύθυνση) και με αυτόν τον τρόπο η master (κύρια) συσκευή διαλέγει με ποια συσκευή θα επικοινωνήσει. Αυτό επιτυγχάνεται με την σύνδεση δύο pin με τις άλλες συσκευές. Τα δύο αυτά pin ονομάζονται Serial Clock (SCL) και Serial DATA (SDA). Το SCL pin συγχρονίζει τα δεδομένα μεταξύ συσκευών και αυτός ο συγχρονισμός γίνεται μόνο από τις master συσκευές. Το SDA pin χρησιμεύει στην μεταφορά των πληροφοριών. Η ταχύτητα μεταφοράς των δεδομένων εξαρτάται από τα Ωμ δύο αντιστάσεων, οι οποίες είναι συνδεδεμένες σε μια πηγή τάσης 5V και τα άκρα σε κάθε ένα από τα pin SCL και SDA. Τα 2ΚΩ αντίστασης αντιστοιχούν στα 400kbps (kilobyte per second) ταχύτητας ενώ τα 10ΚΩ αντίστασης αντιστοιχούν στα 100kbps.

Όπως φαίνεται στην εικόνα 2.5 τα pins 15 και 14 είναι το SDA και το SCL, τα 16 και 8 είναι για την σύνδεση με την τάση και την γείωση και τέλος παρατηρούμε τα pins 5 έως 7 μαζί με τα 9 έως το 12 με το γράμμα «P». Στα pin 5-7 και 9-12 μπορούμε να συνδέσουμε άλλες περιφερειακές συσκευές ή να κάνουμε κάποιες άλλες λειτουργίες δίνοντας εντολή από την master συσκευή.

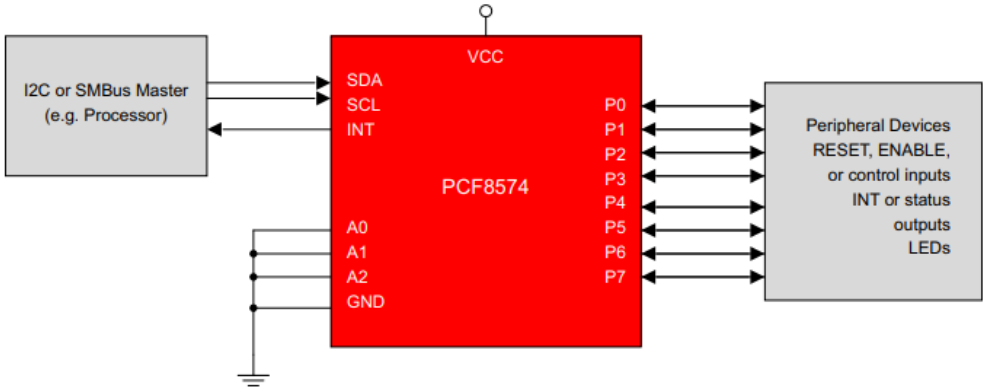


Εικόνα 20-Pins Layout PCF8574

Το ολοκληρωμένο PCF8574 για την σταθερή λειτουργεία του χρειάζεται 5V τάση DC, η κατανάλωση ρεύματος είναι 25mA και οι θερμοκρασίες λειτουργίας είναι από τους 40 °C έως τους 85 °C.

Πίνακας 2.2-pcf8574 πίνακας προτεινόμενων συνθηκών

		MIN	MAX	UNIT
V_{CC}	Supply voltage	2.5	6	V
V_{IH}	High-level input voltage	0.7 × V_{CC}	V_{CC} + 0.5	V
V_{IL}	Low-level input voltage	–0.5	0.3 × V_{CC}	V
I_{OH}	High-level output current		–1	mA
I_{OL}	Low-level output current		25	mA
T_A	Operating free-air temperature	–40	85	°C



*Εικόνα 21-PCF8574 συνδεσμολογία PCF8574 με οθόνη LCD 16 * 2*

2.2 Arduino IDE

Εκτός από τα υλικά τα οποία θα χρησιμοποιηθούν για την κατασκευή του κυκλώματος χρησιμοποιείται επιπλέον το Arduino IDE (Integrated Development Environment) το οποίο είναι ένα πρόγραμμα που είναι συμβατό με πολλά λειτουργικά συστήματα όπως Windows, macOS, Linux. Η γλώσσα προγραμματισμού του είναι η C και C++. Αρχικός στόχος του προγράμματος ήταν να προγραμματίζει αναπτυξιακές πλακέτες Arduino. Το περιβάλλον του προγράμματος Arduino IDE είναι ανοιχτού κώδικα δηλαδή ο χρήστης μπορεί να προσθέσει βιβλιοθήκες για να χρησιμοποιήσει περιφερειακά ηλεκτρονικά υλικά όπως οθόνες, κεραίες κλπ. Αυτές τις βιβλιοθήκες μπορεί να τις γράψει ο κάθε χρήστης. Το πρόγραμμα Arduino IDE μπορεί να χρησιμοποιηθεί και για προγραμματισμό άλλων αναπτυξιακών πλακετών πέρα των Arduino.

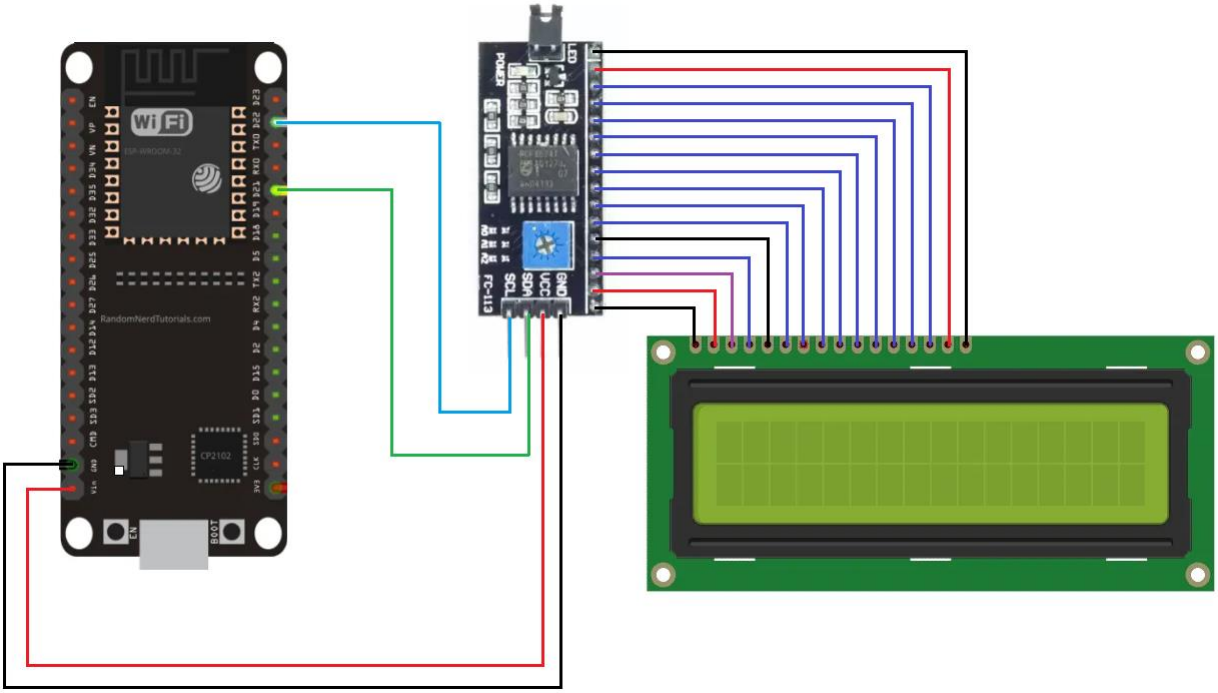
Εικόνα 22-Arduino IDE

Το περιβάλλον του Arduino IDE προσφέρει στον χρήστη:

- Εύκολη πρόσβαση και προσθήκη βιβλιοθηκών από τους developers και άλλους χρήστες
- Έτοιμα παραδείγματα για εκπαίδευση από διάφορες περιφερειακές συσκευές και αναπτυξιακές πλακέτες.
- Serial Monitor το οποίο μπορεί ο χρήστης να το αξιοποιήσει για να εμφανίζονται δεδομένα από τον κώδικα σε μια ψηφιακή οθόνη.

2.3 Σχεδίαση του κυκλώματος

Ο σχεδιασμός του κυκλώματος είναι απλός, αφού δεν χρειάζονται αλλά εξαρτήματα πέρα από αυτά που αναφέρθηκαν στο κεφάλαιο 2.1 και την αναπτυξιακή πλακέτα ESP32.



Εικόνα 23-Συνδεσμολογία του κυκλώματος.

Όπως φαίνεται και στην εικόνα 2.8 η αναπτυξιακή πλακέτα ESP32 τροφοδοτεί το I2C module και αυτό με την σειρά του την οθόνη 16*2 LCD. Το ESP32 συγχρονίζεται με το I2C module με την σύνδεση των pin D22 και του SCL αντίστοιχα με την μέθοδο που αναφέρθηκε στο κεφάλαιο 2.1. Οι πληροφορίες που μεταφέρει το ESP32 γίνεται μέσω του pin D21 στο pin SDA του I2C module. Στην συνέχεια το I2C αποκωδικοποιεί το σήμα και μεταφέρει τις πληροφορίες στο LCD module, όπου με την βοήθεια του HD44780U εμφανίζει το σήμα στην οθόνη σε αλφαριθμητική μορφή.

2.4 Εύρεση περιφερειακών συσκευών I2C

Στο κεφάλαιο 2.1 αναφέρθηκε ότι η επικοινωνία I2C μεταξύ των συσκευών γίνεται με την σύνδεση των SCL, SDA pins και ότι η κάθε συσκευή έχει ένα address. Για να μπορέσει να γίνει χρήση της οθόνης LCD θα χρειαστεί να δηλωθεί address (διεύθυνση) μέσα στον κώδικα. Παρακάτω είναι ο κώδικας για την εύρεση περιφερειακών συσκευών:

```
#include <Wire.h>

byte condition, address;

int devices;

void setup()

{

 Wire.begin();

 Serial.begin(115200);

 Serial.println("\nI2C Scanner");

}

void loop()

{

 Serial.println("Scanning for devices...");

 devices = 0;

 for(address = 1; address < 127; address++ )

 {

  Wire.beginTransmission(address);

  condition = Wire.endTransmission();

  if (condition == 0)

  {

   Serial.print("I2C device found at address 0x");

   if (address<16)
```

```
    {
      Serial.print("0");
    }
    Serial.println(address,HEX);
    devices++;
  }
  else if (condition==4)
  {
    Serial.print("Unknow error at address 0x");
    if (address<16)
    {
      Serial.print("0");
    }
    Serial.println(address,HEX);
              devices++;
  }
}
if (devices == 0)
{
  Serial.println("No I2C devices found\n");
}
```

```
 else
 {
  Serial.print("devices found:");
  Serial.println(devices);
 }
 delay(5000);
}
```

Παρακάτω αναλύονται οι εντολές του κώδικα:

```
#include <Wire.h>

byte condition, address;

int devices;
```

Έγινε δήλωση της βιβλιοθήκης που χρησιμοποιήθηκε στον κώδικα και δήλωση των μεταβλητών.

```
void setup()

{

 Wire.begin();

 Serial.begin(115200);

 Serial.println("\nI2C Scanner");

}
```

Σε αυτό το σημείο του κώδικα με την εντολή wire.begin() δηλώνει ότι θα χρησιμοποιήσει τα pin D21 ως SDA και D22 ως SCL. Στην συνέχεια με την εντολή Serial.begin() κάνει χρήση της ψηφιακής οθόνης και με την εντολή serial.println() εκτυπώνει στην ψηφιακή οθόνη «I2C Scanner».

```
void loop()

{

 Serial.println("Scanning for devices...");

 devices = 0;

 for(address = 1; address < 127; address++ )

 {

  Wire.beginTransmission(address);
```

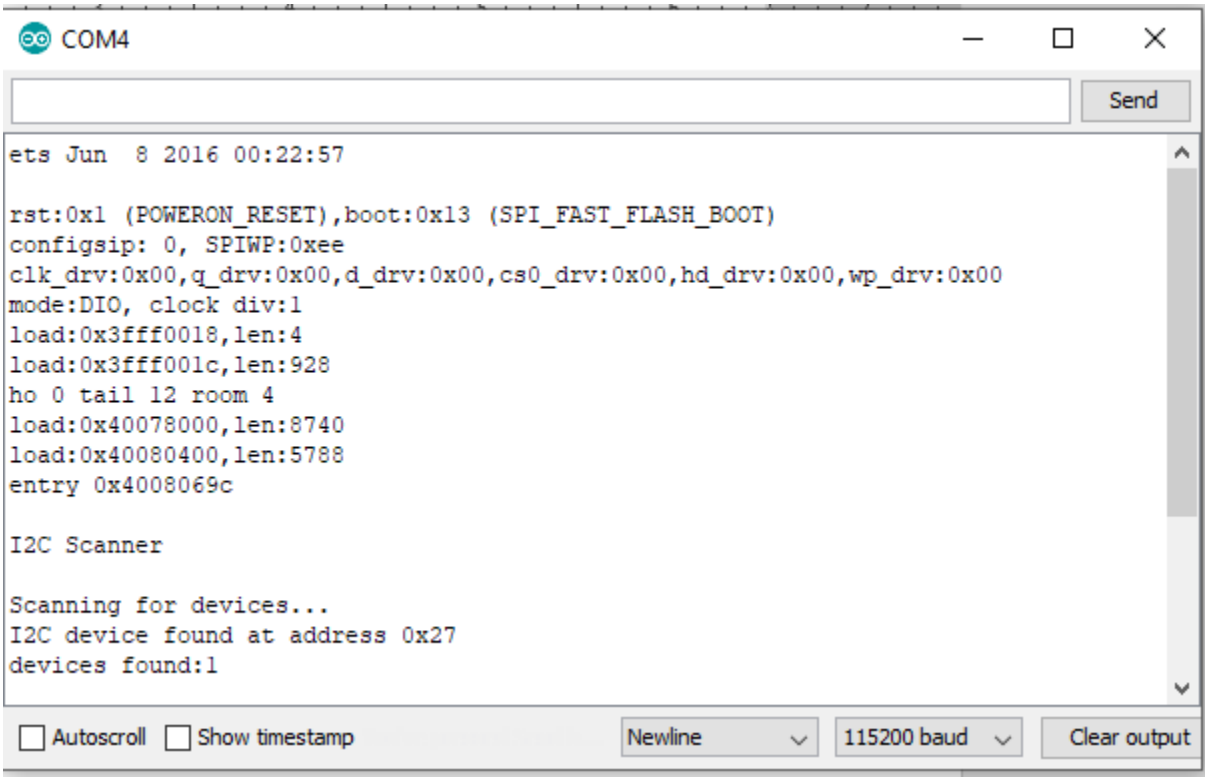
```
  condition = Wire.endTransmission();

  if (condition == 0)

  {

   Serial.print("I2C device found at address 0x");

   if (address<16)

   {

    Serial.print("0");

   }

   Serial.println(address,HEX);

   devices++;

  }

  else if (condition==4)

  {

   Serial.print("Unknow error at address 0x");

   if (address<16)

   {

    Serial.print("0");

   }

   Serial.println(address,HEX);

            devices++;

  }
```

```
 }

 if (devices == 0)

 {

  Serial.println("No I2C devices found\n");

 }

 else

 {

  Serial.println("devices found:");

  Serial.print(devices);

 }

 delay(5000);

}
```

Σε αυτό το κομμάτι του κώδικα με την εντολή void loop () ξεκινάει να γίνεται επανάληψη του κώδικα που συμπεριλαμβάνεται μέσα στα «{,}» μέχρι να σταματήσουμε να τροφοδοτούμε το κύκλωμα. Στην συνέχεια εκτυπώνει με την εντολή Serial.println () στην ψηφιακή οθόνη ότι ξεκινάει να ανιχνεύει για περιφερειακές slaves συσκευές με σύνδεση I2C και αρχικοποιεί την μεταβλητή Devices = 0. Στην επόμενη εντολή for() ορίζει την μεταβλητή address = 1,ορίζει την συνθήκη για να κάνει επαναλήψεις μέχρι η μεταβλητή address= 126 και κάθε φορά που το επαναλαμβάνει να κάνει το address=address+1. Η εντολή Wire.beginTransmission(address) χρησιμοποιείται για να ξεκινήσει την μετάδοση σήματος σε I2C slaves συσκευές με την διεύθυνση που έχει μέσα στην παρένθεση. Υπάρχουν πέντε καταστάσεις που μπορεί να πάρει η εντολή Wire.beginTransmission αυτές είναι:

- 0: επιτυχής επικοινωνία
- 1: μεγάλη πληροφορία για να μπορέσει να το μεταδώσει ο δίαυλος
- 2: στάλθηκε σήμα αλλά δεν επέστρεψε κανένα σήμα
- 3: όταν η συσκευή slave δεν έχει να στείλει άλλη πληροφορία
- 4: κάποιο άλλο πρόβλημα (χάθηκε η σύνδεση από την master συσκευή, κλπ).

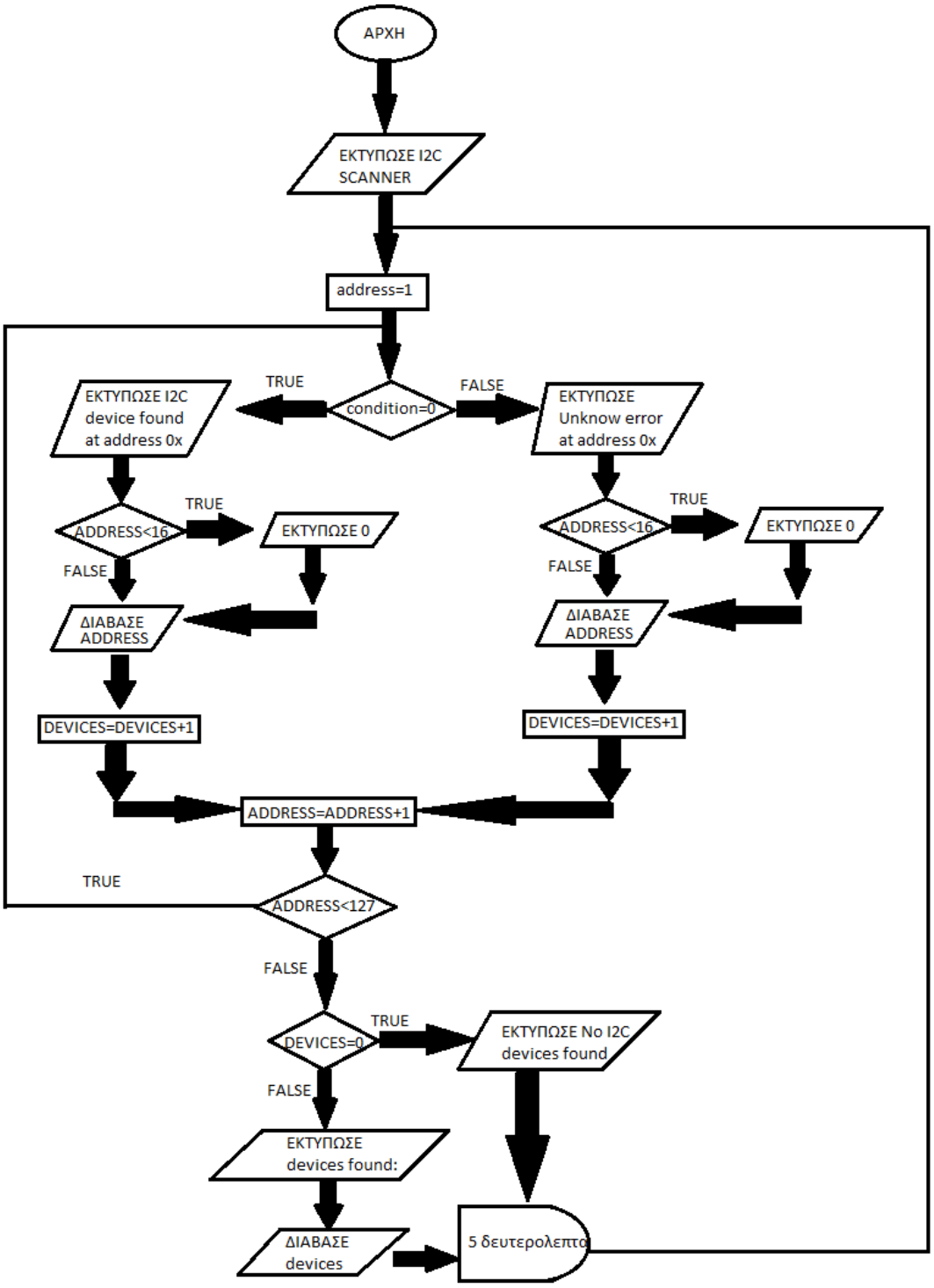
Αυτή την κατάσταση την αποθηκεύει στην μεταβλητή condition. Στην συνέχεια κάνει έλεγχο αν η μεταβλητή condition=0 και αν αληθεύει αυτή η συνθήκη ελέγχει αν η μεταβλητή address<16 τότε



Εικόνα 24-ψηφιακή οθόνη Arduino IDE

εκτυπώνει τον αριθμό 0. Στην συνέχεια εκτυπώνει την μεταβλητή address σε δεκαεξαδικό σύστημα και με την εντολή devices++ κάνει την αριθμητική πράξη devices=devices+1. Η επόμενη κατάσταση που μας ενδιαφέρει στο Wire.beginTransmission είναι όταν condition=4. Σε αυτή την κατάσταση βρέθηκε η συσκευή slave αλλά για κάποιον λόγο δεν γίνεται σωστά η επικοινωνία οπότε ακολουθεί ο ίδιος κώδικας και αν προκύψει πρόβλημα μας ενημερώνει σε ποια συσκευή. Οι άλλες καταστάσεις δεν έχουν κάποια χρήση για την εύρεση περιφερειακών συσκευών Αφού κάνει έλεγχο και τα 127 address εκτελεί έλεγχο αν η μεταβλητή devices=0. Αν ισχύει το πρόγραμμα ενημερώνει τον χρήστη εκτυπώνοντας στην ψηφιακή οθόνη πως δεν βρέθηκε κάποια συσκευή. Σε οποιαδήποτε άλλη κατάσταση ενημερώνει τον χρήστη εκτυπώνοντας στην ψηφιακή οθόνη πόσες περιφερειακές συσκευές βρέθηκαν από τον δίαυλο I2C. Στην συνέχεια περιμένει 5 δευτερόλεπτα και κάνει επανάληψη τις εντολές μέσα στο void loop() μέχρι να σταματήσει η τροφοδοσία ή να κάνει reset ο χρήστης.

Στην εικόνα 2.9 φαίνεται η διεύθηνση του module lcd που χρησιμοποιήται για την εργασία. Στην εικόνα 2.10 υπάρχει το μπλοκ διάγραμμα του προγράμματος εύρεσης περιφεριακών συσκευων με συνδεση I2C.



Εικόνα 25-Διάγραμμα ροής εύρεσης περιφερειακών συσκευών I2C

2.5 Πρωτόκολλα.

Τα πρωτόκολλα δίνουν στον χρήστη την δυνατότητα να ορίσει κάποιες διαδικασίες επικοινωνίας χωρίς να χρειάζεται να γνωρίζει λεπτομέρειες του υλικού των δικτύων και συνήθως είναι κατασκευασμένα να προστατεύουν από κακόβουλες επιθέσεις λογισμικού.

2.5.1 Πρωτόκολλο UDP (User Datagram Protocol).

Το πρωτόκολλο UDP είναι ένα από τα βασικά πρωτόκολλα που χρησιμοποιούνται στο διαδίκτυο. Μία εναλλακτική ονομασία του πρωτοκόλλου είναι Universal Datagram Protocol. Διάφορα προγράμματα χρησιμοποιούν το πρωτόκολλο UDP για την αποστολή σύντομων μηνυμάτων (γνωστών και ως segments) από τον έναν υπολογιστή στον άλλον μέσα σε ένα δίκτυο υπολογιστών. Ένα από τα κύρια χαρακτηρίστηκα του UDP είναι ότι δεν πρόκειται για αξιόπιστη μέθοδος επικοινωνίας. Εάν το δίκτυο έχει μεγάλο φόρτο τα πακέτα UDP που αποστέλλονται από έναν υπολογιστή μπορεί να φτάσουν με λάθος σειρά, διπλά η να μην φτάσουν και καθόλου. Ουσιαστικά η χρήση του γίνεται όταν η ταχύτητα είναι πιο σημαντική από την ακρίβεια της. Φυσικά υπάρχει αξιόπιστο πρωτόκολλο το οποίο έχει τους κατάλληλους μηχανισμούς ώστε να υπάρχει αξιοπιστία μεταξύ της επικοινωνίας των υπολογιστών και το πρωτόκολλο αυτό ονομάζεται TCP (Transmission Control Protocol). Η έλλειψη αυτών των μηχανισμών από το UDP το καθιστούν αρκετά πιο γρήγορο και αποτελεσματικό για εφαρμογές που δεν απαιτούν αξιόπιστη επικοινωνία.

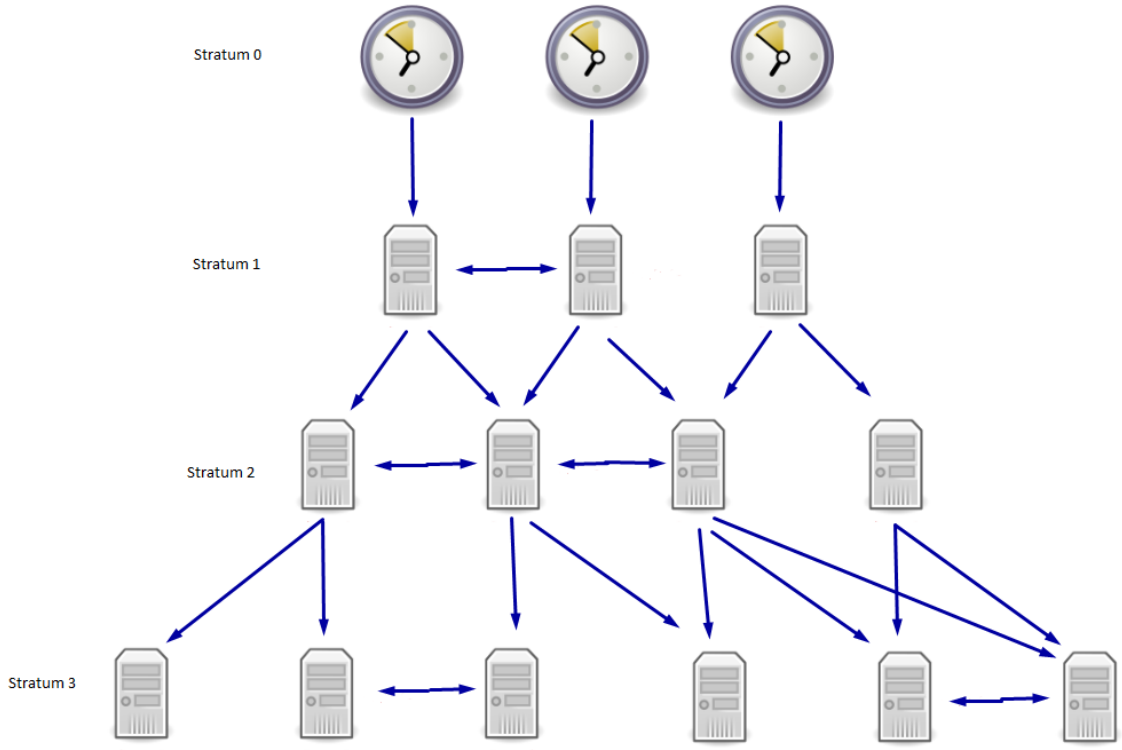
Εφαρμογές οι οποίες κάνουν χρήση του UDP είναι η μετάδοση εικόνας και ήχου (audio-video streaming). Σε αυτές τις εφαρμογές το ζητούμενο είναι να αποστέλλονται γρήγορα τα πακέτα ώστε να μην γίνεται διακοπή στην ροή του βίντεο ή του ήχου, παρόλο που υπάρχει πιθανότητα να χαθούν μερικά πακέτα Οι εφαρμογές αυτές διαθέτουν μηχανισμούς όπως π.χ σε περιπτώσεις που χαθεί κάποιο πακέτα να το διορθώσουν ώστε ο χρήστης που θα το παραλάβει να γίνει διακοπή ή να μην είναι εμφανής η αλλοίωση στην ροή της μετάδοσης εικόνας και ήχου. Γνωστές εφαρμογές που χρησιμοποιούν το UDP είναι το DNS (Domain Name System),VoIP (Voice over IP),IPTV και διάφορα παιχνίδια διαδικτύου ζωντανής ροής.

2.5.2 Πρωτόκολλο NTP (Network Time Protocol)

Το NTP (πρωτόκολλο δικτυακού χρόνου), είναι ένα πρωτόκολλο υπολογιστών. Το πρωτόκολλο αυτό χρησιμοποιείται για τον συγχρονισμό της ώρας από διαδικτυακές συσκευές με μια πηγή ώρας που λειτουργεί ως σημείο αναφοράς. Με το πρωτόκολλο NTP επιτυγχάνεται συγχρονισμός ονομαστικής ακρίβειας σε χιλιοστό του δευτερολέπτου σε τοπικά δίκτυα και για δίκτυα ευρείας περιοχής σε διαφορά χιλιοστού του δευτερολέπτου. Είναι σημαντικό να αναφερθεί ότι υπάρχουν παράμετροι οι οποίες προστατεύουν κακόβουλες επιθέσεις.

Ο συγχρονισμός της ώρας των συσκευών επιτυγχάνεται από εξυπηρετητές (clients) NTP που συνδέονται με τον server NTP και με αυτόν τον τρόπο οι δικτυακές συσκευές έχουν κοινό σημείο αναφοράς. Σε πολλές περιπτώσεις οι εξυπηρετητές συγχρονίζονται με άλλα ρολόγια αναφοράς και τελικά συγχρονίζονται με το UTC (Coordinal Universal Time). Η διαδικασία συγχρονισμού ονομάζεται Clock Strat και ακολουθεί την εξής πυραμίδα:

1. Stratum 0 ονομάζονται τα ατομικά ρολόγια (atomic clocks), είναι η πιο υψηλή μέτρηση ακρίβειας του χρόνου που υπάρχει μέχρι σήμερα. Για τον καθορισμό του διεθνούς ατομικού χρόνου χρησιμοποιείται ένα παγκόσμιο δίκτυο από 200 ατομικά ρολόγια που βρίσκονται σε πάνω από 50 εθνικά εργαστήρια. Αξίζει να σημειωθεί ότι το ατομικό ρολόι έχει ακρίβεια 0,0000000000000015 δευτερόλεπτα.
2. Stratum 1 είναι οι κύριοι πρωτεύοντες εξυπηρετητές (master clients) οι οποίοι συγχρονίζουν την ώρα με διαφορά λίγων μικροδευτερολέπτων από το Stratum 0. Οι master clients κάνουν ελέγχους μεταξύ τους ανά περιοχή για την ακρίβεια της ώρας αυτοί θεωρούνται και ως primary time servers.
3. Stratum 2 είναι οι υπολογιστές οι οποίοι συγχρονίζονται με τους Stratum 1. Πολλές φορές χρησιμοποιούν πάνω από έναν master client και κάνουν ελέγχους μεταξύ τους για τον έλεγχο αλλά και την ακρίβεια της ώρας.
4. Stratum 3 – 15 είναι οι υπολογιστές οι οποίοι ακολουθούν την ίδια διαδικασία με αυτήν του Stratum 2 αλλά πρόκειται για πιο απομακρυσμένο δίκτυο και μεσολαβεί άλλο κλιμάκιο.



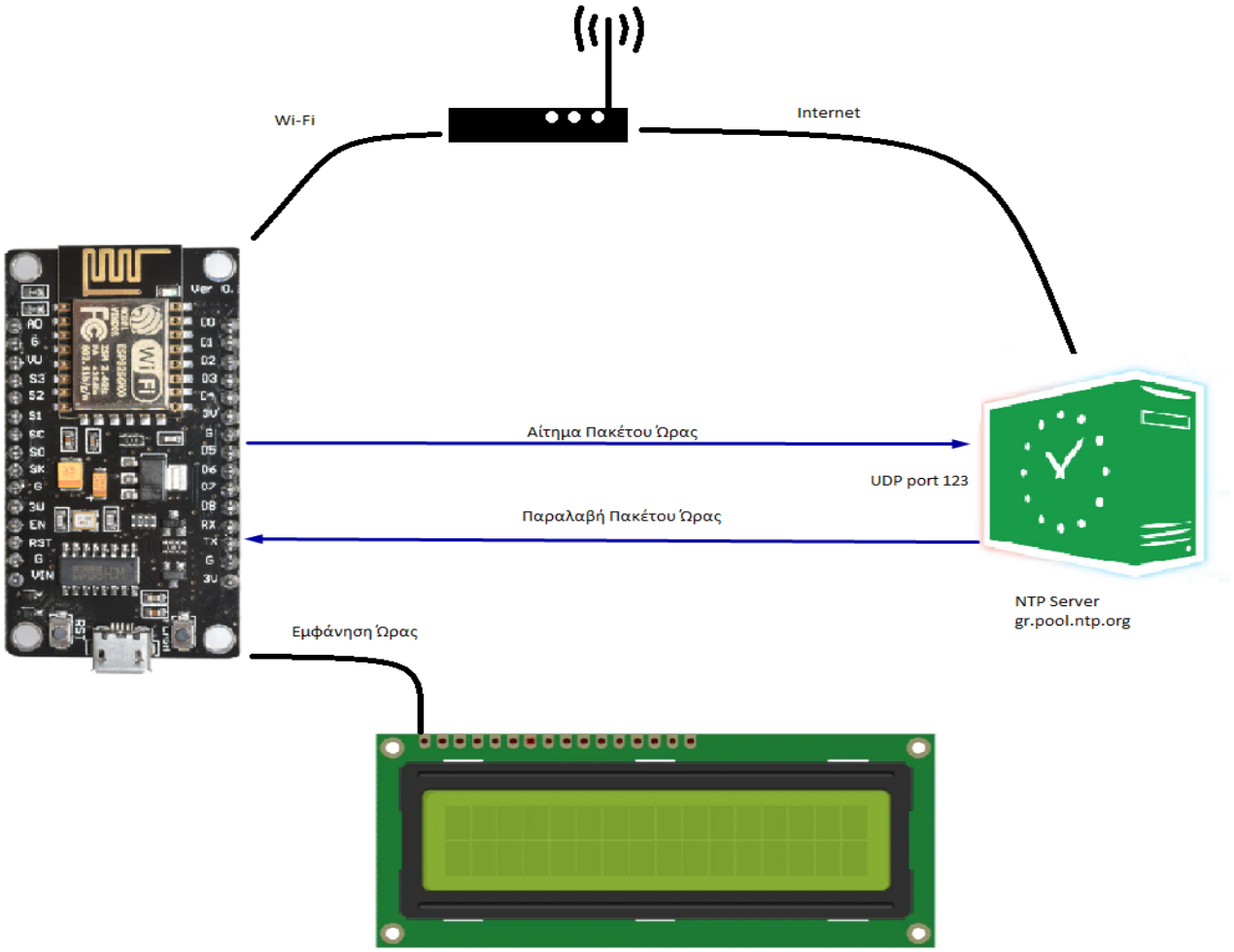
Εικόνα 26-Δομή NTP

Μπορεί να ξεπεράσει το Stratum 15 αλλά θεωρείται ότι είναι ασύγχρονο μετά από αυτό το κλιμάκιο λόγο του χρόνου που χρειάστηκε για να κάνει συγχρονισμό μέχρι το Stratum 15. Πρέπει να σημειωθεί ότι από

το Stratum 2 και μετά χρησιμοποιείται το πρωτόκολλο UDP για την επικοινωνία με την πόρτα (port) 123 του NTP server.

2.6 Ανάπτυξη κώδικα του ESP32 με την σύνδεση του NTP

Μέχρι τώρα έγινε επεξήγηση των εξαρτημάτων του κυκλώματος, τα πρωτόκολλα που χρησιμοποιούνται για την υλοποίηση της εργασίας αλλά και με την βοήθεια του κώδικα για την εύρεση της «θέσης» της οθόνης LCD με τον δίαυλο επικοινωνίας I2C.



Εικόνα 27-Διάγραμμα διαδικτυακού ρολογιού με το ESP32

Στη συνέχεια παρατίθεται ο κώδικας που χρησιμοποιείται από το ESP32.

```
#include <WiFi.h>
```

```
#include "time.h"

#include <LiquidCrystal_I2C.h>



LiquidCrystal_I2C lcd(0x27, 16, 2);



const char *ssid     = "YOUR Wi-Fi";

const char *password = "YOUR Wi-Fi PASSWORD";

const char* ntpServer = "gr.pool.ntp.org";

const long  gmtOffset_sec = 10800;

const int   daylightOffset_sec = 0;



void setup()

{

 Serial.begin(115200);



 lcd.init();

 lcd.backlight();

 lcd.setCursor(0, 0);



 Serial.print("Connecting to ");

 lcd.print("Connecting to ");
```

```
Serial.println(ssid);

lcd.setCursor(0, 1);

lcd.print(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED)

{

  delay(500);

  Serial.print(".");

  lcd.print(".");

}

Serial.println("");

lcd.clear();

Serial.println("WiFi connected.");

lcd.setCursor(0, 0);

lcd.print("WiFi connected.");

lcd.clear();

configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);

printLocalTime();



WiFi.disconnect(true);

WiFi.mode(WIFI_OFF);
```

```
}


void printLocalTime()
{
 struct tm timeinfo;
 if(!getLocalTime(&timeinfo))
 {
  Serial.println("Failed to obtain time");
  lcd.print("Failed to obtain time");
  return;
 }

 Serial.println(&timeinfo, "%A, %B %d %Y %H:%M:%S");
 lcd.setCursor(0, 0);
 lcd.print(&timeinfo, "Date %d/%m/%Y");
 lcd.setCursor(0, 1);
 lcd.print(&timeinfo, "Time  %H:%M:%S");
}
void loop()
{
 delay(1000);
```

```
  printLocalTime();

}
```

Όπως φαίνεται στον κώδικα χρησιμοποιούμε εκτός από την οθόνη LCD και το serial monitor που προσφέρει το πρόγραμμα Arduino IDE για έλεγχο ορθότητας της κατασκευής.

Οι βιβλιοθήκες οι οποίες χρησιμοποιούνται για σκοπό της κατασκευής είναι:

```
#include <WiFi.h>

#include "time.h"

#include <LiquidCrystal_I2C.h>
```

Με την παρακάτω εντολή δηλώνει ο χρήστης ότι στην διεύθυνση 0x27 υπάρχει η οθόνη LCD 16*2 και ο δίαυλος επικοινωνίας είναι I2C.

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

Παρακάτω ο χρήστης δηλώνει τις μεταβλητές που θα κάνει χρήση στο πρόγραμμα και τις ορίζει.

```
const char *ssid     = "YOUR Wi-Fi";

const char *password = "YOUR Wi-Fi PASSWORD";

const char* ntpServer = "gr.pool.ntp.org";

const long  gmtOffset_sec =10800;

const int   daylightOffset_sec = 0;
```

Το GMT (Greenwich Mean Time) αντιπροσωπεύει την ζώνη ώρας στην οποία η κάθε χώρα ανήκει. Το GMT έχει αντικατασταθεί από το UTC (Coordinated Universal Time) και ορίζει την Ελλάδα στο UTC+3. Λόγω χρήσης της συγκεκριμένης βιβλιοθήκης η εντολή ονομάζεται gmtOffset για τον ορισμό της ζώνης της χώρας. Οπότε, ο χρήστης ορίζει την ώρα σε δευτερόλεπτα δηλαδή 3600 seconds * (ζώνη ώρας) = gmtOffset_sec.

```
void setup()

{
```

Σε αυτό το σημείο του κώδικα το πρόγραμμα ξεκινάει να εκτελεί εντολές οι οποίες θα εκτελεστούν μόνο μία φορά.

```
 Serial.begin(115200);



 lcd.init();

 lcd.backlight();

 lcd.setCursor(0, 0);
```

Παραπάνω ο χρήστης κάνει έναρξη της ψηφιακής οθόνης καθώς και της LCD οθόνης η οποία είναι συνδεδεμένη μέσω του I2C module στα pins 21 και 22. Ενεργοποιεί το backlight της οθόνης και ορίζει την θέση που θα γράψει στην οθόνη.

```


 Serial.print("Connecting to ");

 lcd.print("Connecting to ");

 Serial.println(ssid);

 lcd.setCursor(0, 1);

 lcd.print(ssid);
```

Σε αυτό το σημείο ο κώδικας εκτυπώνει πως θα συνδεθεί στο δίκτυο που δήλωσε ο χρήστης από τις μεταβλητές.

Με την παρακάτω εντολή ο χρήστης δίνει εντολή στο ESP32 να συνδεθεί στο Wi-Fi δίκτυο με το ssid και password που όρισε στις μεταβλητές

```
WiFi.begin(ssid, password);
```

Στην συνέχεια ανά μισό δευτερόλεπτο κάνει έλεγχο αν συνδέθηκε το ESP32 με το Wi-Fi δίκτυο και κάθε φορά που κάνει τον έλεγχο και εφόσον δεν είναι συνδεδεμένο εκτυπώνει μια «.» στην ψηφιακή οθόνη αλλά και στην LCD δίπλα από το όνομα δικτύου.

```
while (WiFi.status() != WL_CONNECTED)

{

  delay(500);

  Serial.print(".");

  lcd.print(".");

}
```

Όταν γίνει η σύνδεση στο δίκτυο Wi-Fi, με τις επόμενες εντολές ο χρήστης αλλάζει σειρά στην ψηφιακή οθόνη και καθαρίζει την οθόνη LCD από χαρακτήρες που έχουν γραφτεί.

```
Serial.println("");

lcd.clear();
```

και δηλώνει στον χρήστη ότι έγινε σύνδεση στο δίκτυο Wi-Fi εκτυπώνοντας ότι συνδέθηκε

```
Serial.println("WiFi connected.");

lcd.setCursor(0, 0);

lcd.print("WiFi connected.");

lcd.clear();
```

παρακάτω διαμορφώνει τον κώδικα με τις ρυθμίσεις που έχει ορίσει ο χρήστης τις μεταβλητές και καλεί την συνάρτηση printLocalTime()

```
configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
```

```
 printLocalTime();
```

στην συνέχεια επειδή δεν χρειάζεται άλλο πια η χρήση του Wi-Fi μέσω του κώδικα γίνεται αποσύνδεση από το δίκτυο Wi-Fi και σταματάει την χρήση της κεραίας Wi-Fi για εξοικονόμηση ενέργειας.

```
 WiFi.disconnect(true);

 WiFi.mode(WIFI_OFF);

}
```

Επίσης σε αυτό το σημείο κλείνει η void setup

Στην συνέχεια ο κώδικας ανοίγει την συνάρτηση PrintLocalTime(). Σε αυτήν τη συνάρτηση, δημιουργεί μια δομή χρόνου (struct tm) την ονομάζει timeinfo και περιέχει όλες τις λεπτομέρειες για την ώρα (λεπτά, δευτερόλεπτα, ώρα, κλπ…).

```
void printLocalTime()

{

 struct tm timeinfo;
```

Στον παρακάτω πίνακα φαίνονται οι μεταβλητές που δημιουργούνται από την struct tm σε γλώσσα προγραμματισμού C

Πίνακας 2.3-Βάση δεδομένων struct tm

Όνομα	Τύπος μεταβλητής	Σημασία μεταβλητής	Εύρος
%S	int	Δευτερόλεπτα	0-60
%M	int	Λεπτά	0-59
%H	int	Ώρα	0-23
%d	int	Ημερομηνία	1-31
%m	int	Μήνας	0-11

%Y	int	Χρονολογία	
%A	Char	Όνομα ημέρας	
%B	Char	Όνομα μήνα	
%U	int	Ημέρα από την Κυριακή	0-6

Είναι σημαντικό να αναφερθεί ότι επειδή το πρωτόκολλο το οποίο χρησιμοποιείται είναι το UDP υπάρχει περίπτωση το πακέτο που ζητάει το ESP32 να μην το λάβει σε σωστή μορφή ή να μην είναι ολόκληρο ή και καθόλου οπότε σε αυτή την περίπτωση εκτυπώνεται η αποτυχία της λήψης της ώρας και σταματάει την χρήση της συνάρτησης printLocalTime()

```
if(!getLocalTime(&timeinfo))

{

  Serial.println("Failed to obtain time");

  lcd.print("Failed to obtain time");

  return;

}
```

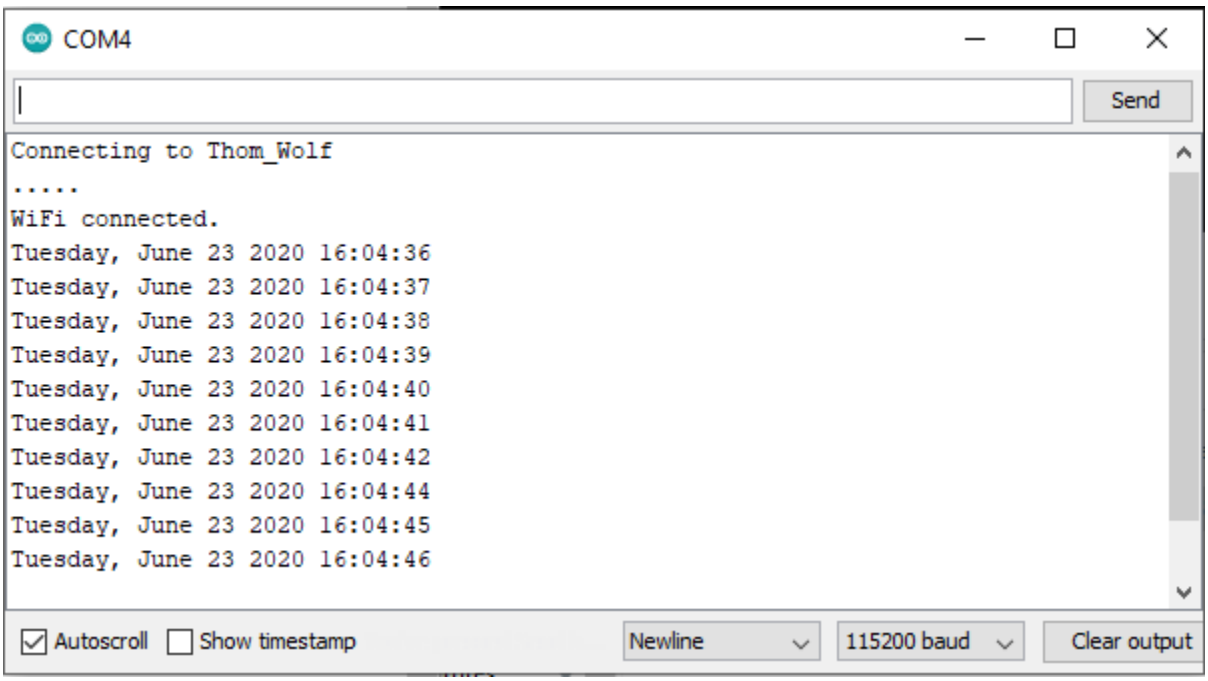
Στην περίπτωση που το πακέτο που ήρθε από τον NTP server είναι σωστό τότε εκτυπώνει την ώρα στην ψηφιακή οθόνη και στην LCD οθόνη και κλείνει η συνάρτηση printLocalTime()

```
 Serial.println(&timeinfo, "%A, %B %d %Y %H:%M:%S");

 lcd.setCursor(0, 0);

 lcd.print(&timeinfo, "Date %d/%m/%Y");

 lcd.setCursor(0, 1);

 lcd.print(&timeinfo, "Time  %H:%M:%S");

}
```

Τέλος μέσα στην συνάρτηση void loop όπου γίνεται η επανάληψη καλεί την συνάρτηση printLocalTime() και με καθυστέρηση 1 δευτερολέπτου κάνει επανάληψη.

```
void loop()

{

 delay(1000);

 printLocalTime();

}
```

Αν γίνει σωστή συνδεσμολογία του κυκλώματος, με αυτόν τον κώδικα θα εμφανιστεί στην ψηφιακή οθόνη

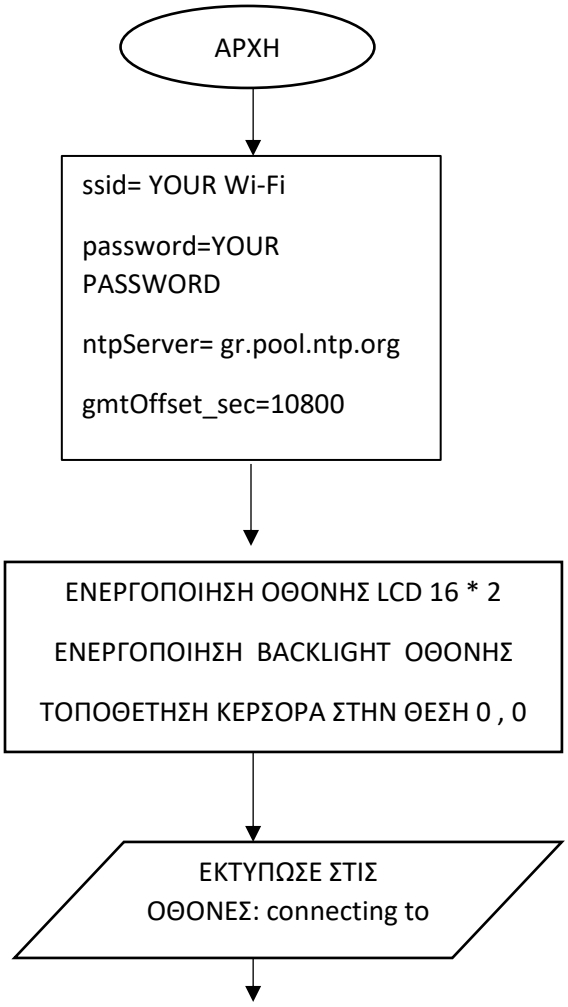


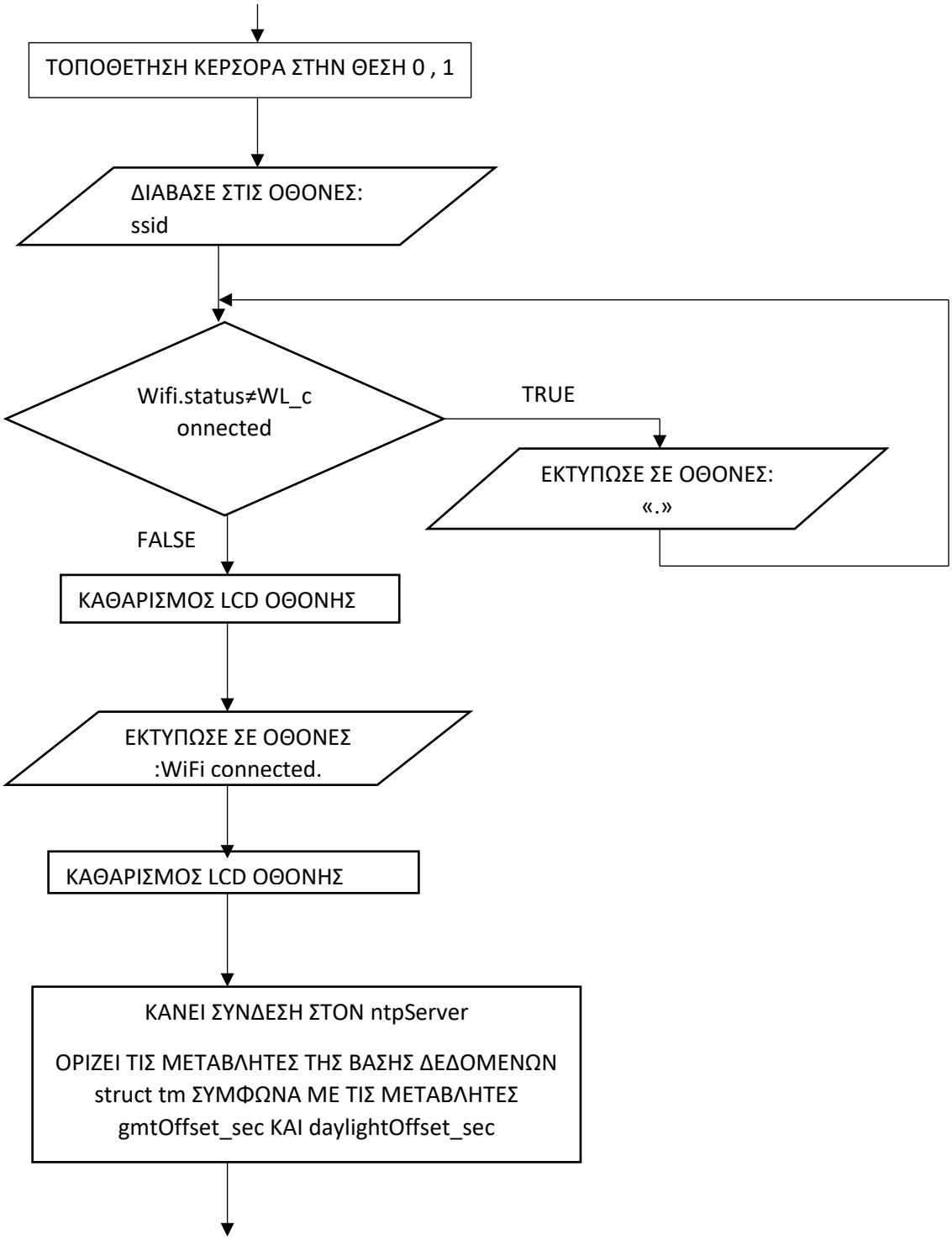
Εικόνα 28 Αποτέλεσμα ψηφιακής οθόνης

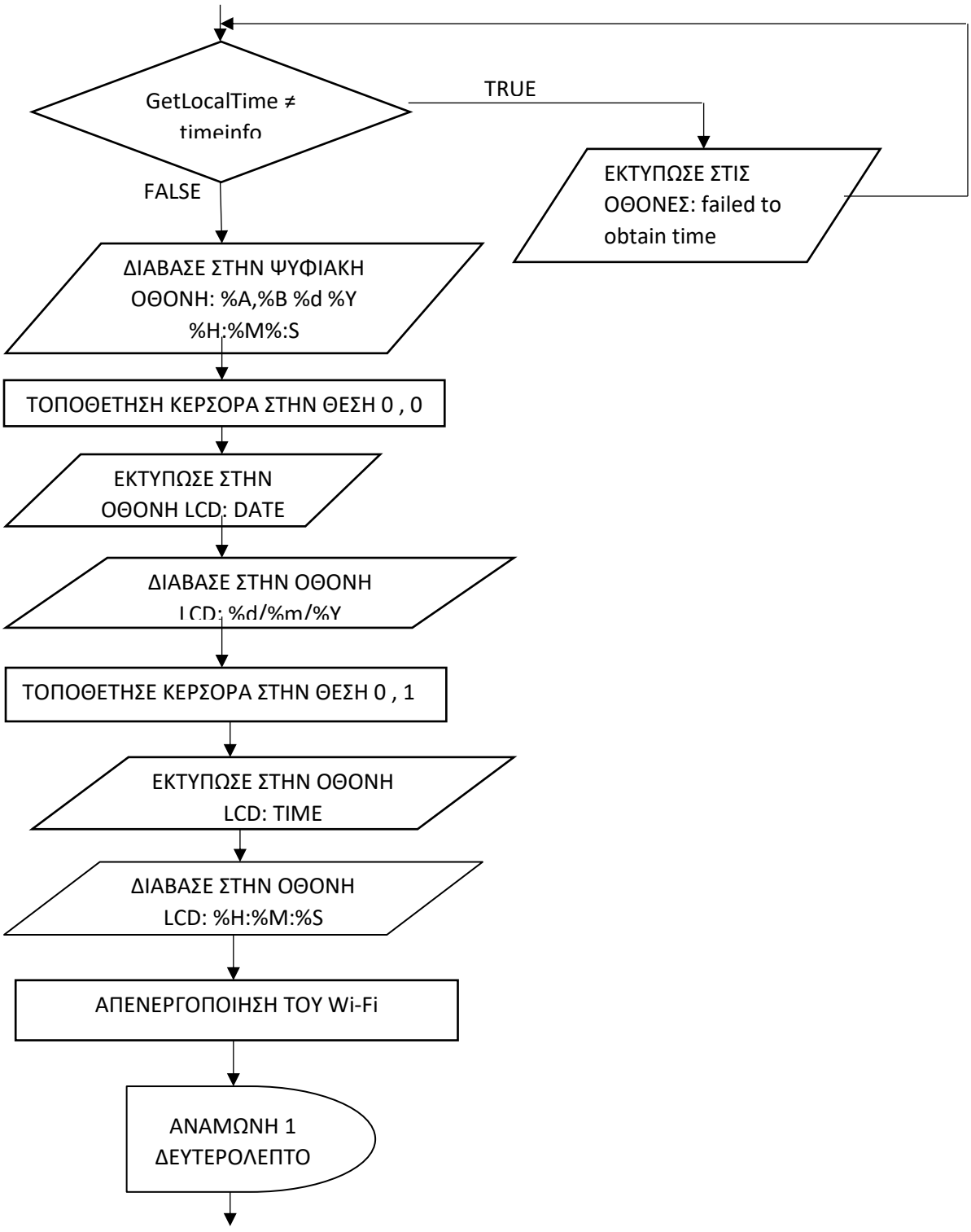
Ενώ στην οθόνη LCD θα εμφανιστεί η ώρα όπως φαίνεται στην εικόνα 2.14. και κάθε δευτερόλεπτο ανανεώνεται η ένδειξη της ώρας.

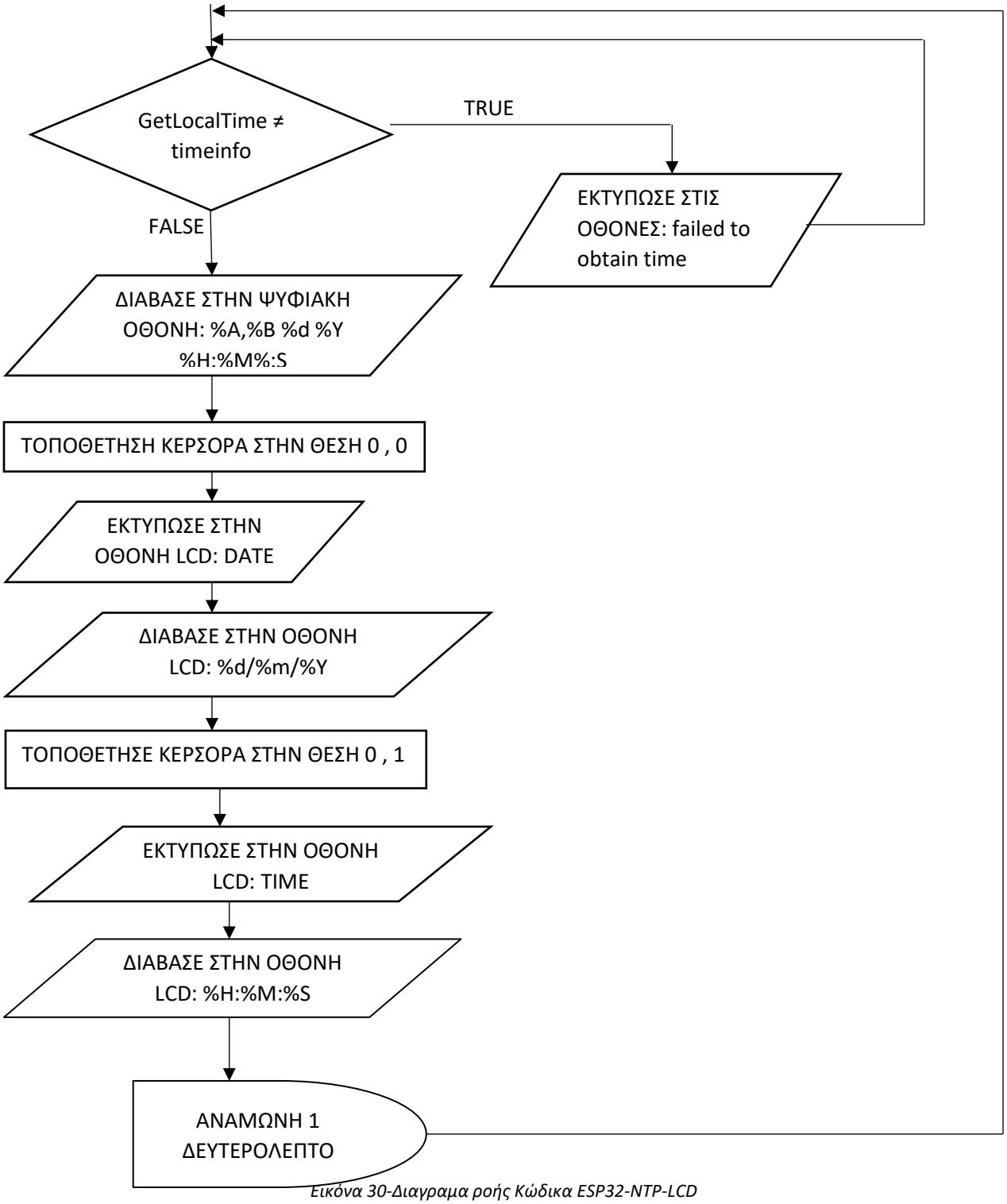
Εικόνα 29.14-Αποτέλεσμα LCD οθόνης

Παρακάτω στην εικόνα 2.15 είναι το διάγραμμα ροής του που δείχνει την δομή του κώδικα.




ΤΟΠΟΘΕΤΗΣΗ ΚΕΡΣΟΡΑ ΣΤΗΝ ΘΕΣΗ 0 , 1
ΔΙΑΒΑΣΕ ΣΤΙΣ ΟΘΟΝΕΣ:
ssid
Wifi.status≠WL_c
onnected
TRUE
ΕΚΤΥΠΩΣΕ ΣΕ ΟΘΟΝΕΣ:
«.»
FALSE
ΚΑΘΑΡΙΣΜΟΣ LCD ΟΘΟΝΗΣ
ΕΚΤΥΠΩΣΕ ΣΕ ΟΘΟΝΕΣ
:WiFi connected.
ΚΑΘΑΡΙΣΜΟΣ LCD ΟΘΟΝΗΣ
ΚΑΝΕΙ ΣΥΝΔΕΣΗ ΣΤΟΝ ntpServer
ΟΡΙΖΕΙ ΤΙΣ ΜΕΤΑΒΛΗΤΕΣ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ
struct tm ΣΥΜΦΩΝΑ ΜΕ ΤΙΣ ΜΕΤΑΒΛΗΤΕΣ
gmtOffset_sec ΚΑΙ daylightOffset_sec


GetLocalTime ≠ timeinfo
TRUE
ΕΚΤΥΠΩΣΕ ΣΤΙΣ ΟΘΟΝΕΣ: failed to obtain time
FALSE
ΔΙΑΒΑΣΕ ΣΤΗΝ ΨΥΦΙΑΚΗ ΟΘΟΝΗ: %A,%B %d %Y %H:%M%:S
ΤΟΠΟΘΕΤΗΣΗ ΚΕΡΣΟΡΑ ΣΤΗΝ ΘΕΣΗ 0 , 0
ΕΚΤΥΠΩΣΕ ΣΤΗΝ ΟΘΟΝΗ LCD: DATE
ΔΙΑΒΑΣΕ ΣΤΗΝ ΟΘΟΝΗ LCD: %d/%m/%Y
ΤΟΠΟΘΕΤΗΣΕ ΚΕΡΣΟΡΑ ΣΤΗΝ ΘΕΣΗ 0 , 1
ΕΚΤΥΠΩΣΕ ΣΤΗΝ ΟΘΟΝΗ LCD: TIME
ΔΙΑΒΑΣΕ ΣΤΗΝ ΟΘΟΝΗ LCD: %H:%M:%S
ΑΠΕΝΕΡΓΟΠΟΙΗΣΗ ΤΟΥ Wi-Fi
ΑΝΑΜΩΝΗ 1 ΔΕΥΤΕΡΟΛΕΠΤΟ



Εικόνα 30-Διαγραμα ροής Κώδικα ESP32-NTP-LCD

ΚΕΦΑΛΑΙΟ 3

3.1 Χρήση διαδικτυακού ρολογιού ESP32

Η χρήση του διαδικτυακού ρολογιού μπορεί να αξιοποιηθεί σε πολλές εφαρμογές στις οποίες χρειάζεται η μέτρηση της πραγματικής ώρας. Είναι αξιόπιστη εφαρμογή καθώς δέχεται την ώρα από τον NTP server και δεν εξαρτάται η μέτρηση από κάποια άλλη συσκευή. Έχει χαμηλή κατανάλωση ενέργειας, είναι εύχρηστο αφού χρειάζεται μόνο μια πηγή ενέργειας 5V όπως αυτές των κινητών τηλεφώνων.

Εφαρμογές που θα μπορούσε να γίνει χρήση του διαδικτυακού ρολογιού:

- Για δημιουργία Smarthome.
- Λήψη δειγμάτων.
- Ενεργοποίηση η απενεργοποίηση δικτύων ή συσκευών.
- Ενημέρωση της ώρας στον χρήστη ή άλλες συσκευές.

Η υλοποίηση της κατασκευής μπορεί να γίνει και σε άλλες οθόνες με άλλες διασυνδέσεις όπως αναφέρονται και στον πίνακα 1.1 με τη διαφορά ότι ο κώδικας θα πρέπει να προσαρμοστεί στην κάθε οθόνη όπως και στην αντίστοιχη καλωδίωση.

Βιβλιογραφικές αναφορές

Internet Site

[1] ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD. is a fabless semiconductor company, with headquarter in Shanghai Zhangjiang High-Tech Park. https://www.espressif.com/

[2] The open-source **Arduino** Software (**IDE**) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux https://www.arduino.cc/

[3] TI is a global semiconductor design & manufacturing company. Innovate with 80000+ analog ICs & embedded processors, software & largest sales/support staff. https://www.ti.com/

[4] SparkFun Electronics is an electronics retailer in Niwot, Colorado, United States. It manufactures and sells microcontroller development boards and breakout boards. https://www.sparkfun.com/

Βιβλία

[5] Internet of Things Projects with ESP32: Build exciting and powerful IoT projects using the all-new Espressif ESP32.

Data sheet

[6] ESPRESSIF SYSTEMS ESP32 Modules and Boards.

[7] Texas Instruments, pcf8574.

[8] SparkFun Electronics, LCD 16 * 2.