

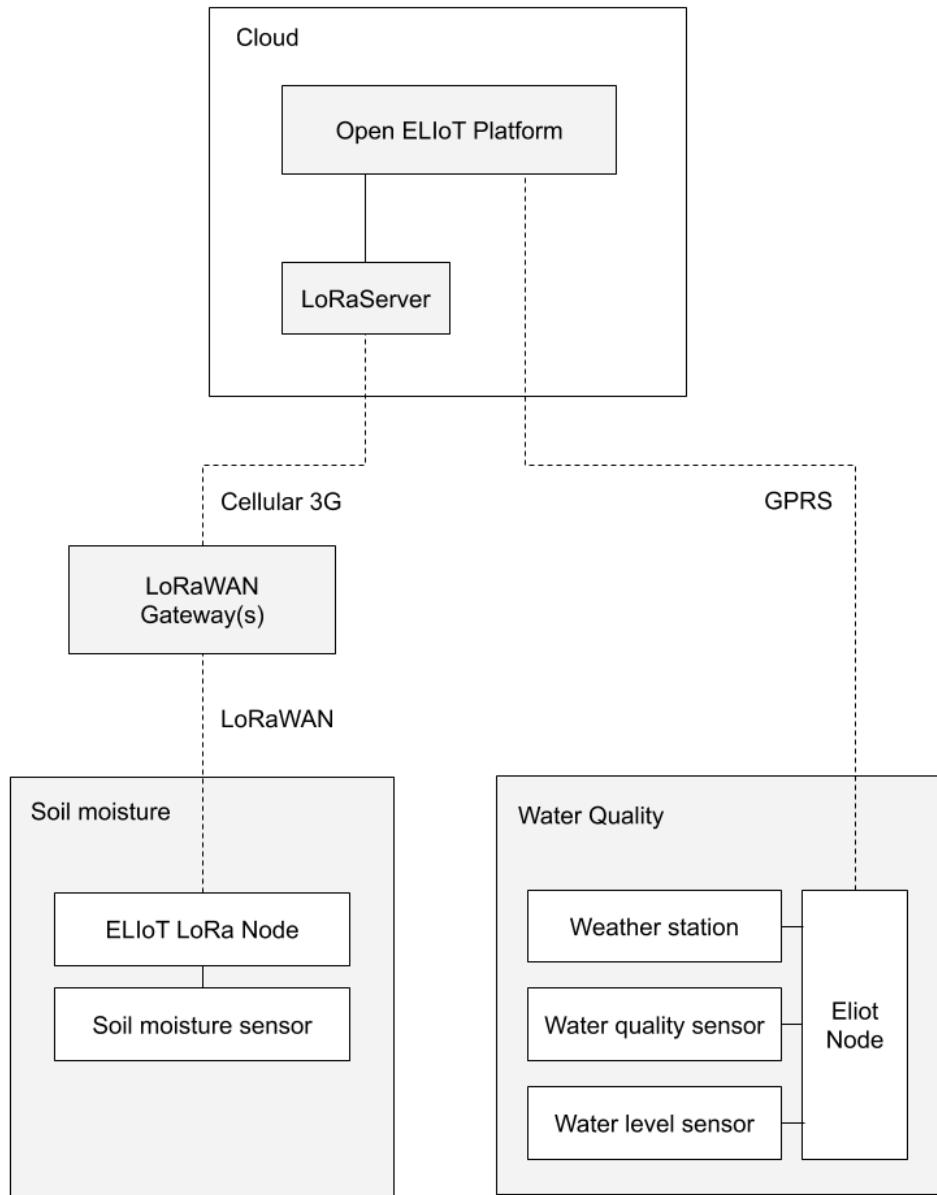
Τεχνικός οδηγός ανάπτυξης αισθητήρων μέτρησης περιβαλλοντικών παραμέτρων - 1η έκδοση

Αρχιτεκτονική λυσης - εξοπλισμος	4
ELIoT Node (v1)	6
Κεντρικός μικροελεγκτής και βασική πλακέτα	6
Επικοινωνίες GSM	8
Πρωτοκόλλα επικοινωνιας με αισθητηρες SDI-12	9
Αισθητήρας μέτρησης ποιότητας υδάτων	10
Αισθητήρας μέτρησης στάθμης	11
Μετεωρολογικός σταθμός	13
Ασυρματα πρωτοκόλλα επικοινωνιας με πλατφορμα	16
Διαχείριση ενέργειας	16
Περίβλημα	17
Βάση στήριξης πλακετών	17
ELIoT LoRa Node	19
Κεντρικος μικροελεγκτης	19
Αισθητήρας μέτρησης υγρασίας	20
LoRaWAN Gateway	20
Ρύθμιση του gateway	21
Ρύθμιση LoRa Server	22
Διαχείριση ενέργειας	28
Περίβλημα	28
Διάγραμμα συνδεσης - συναρμολόγηση	29

TTGo TFox	30
DFRobot Solar Manager	30
SIM7000	30
Arduino Nano 5V	30
Level Shifter	30
DS2312 RTC	30
BME280	31
Υλισμικό (Firmware)	31
Εκκίνηση	31
Κύκλος Λειτουργίας	31
Επικοινωνία με την πλατφόρμα	32
Μέτρηση παραμέτρων αισθητήρων	33
Μνήμη και διαχείριση αρχείων	33
Καταγραφή συμβάντων (Logging)	33
Ώρα	34
FOTA (Firmware Over The Air)	34
Αρχικοποίηση συσκευής	34
Αισθητήρας ποιότητας εδάφους AquaTROLL 400	34
Προγραμματισμός firmware	36
Σύνδεση με πλατφόρμα	40
Πρόσθεση της συσκευής στην πλατφόρμα	40
Ρυθμίσεις σύνδεσης στο υλισμικό	42
Εύρεση διεύθυνσης MAC ενός node	43
Απομακρυσμένος έλεγχος / ενημέρωση λογισμικού (FOTA)	43
Απομακρυσμένος έλεγχος	43
Απομακρυσμένη ενημέρωση λογισμικού (FOTA)	45
Testing / debugging	46

'Εκδοση debug / release	46
Logs	46
Προβλήματα, επόμενα βήματα	48
Σφαλματα στις επικοινωνιες GSM	48
Παράρτημα προδιαγραφών εξαρτημάτων (datasheets)	50

Αρχιτεκτονική λύσης - εξοπλισμός



Σχήμα: Αρχιτεκτονική λύσης

Το σύστημα αποτελείται από 2 τυπους σταθμων (sensor nodes) οι οποίοι εγκαθίστανται σε διαφορετικά πεδία ανάλογα με τις παραμέτρους για τις οποίες προορίζονται:

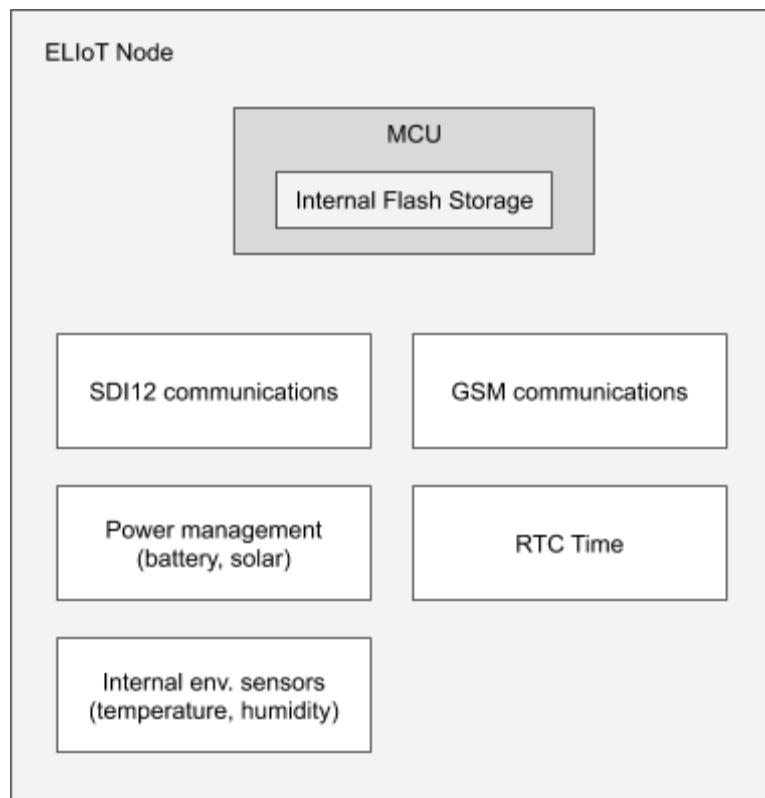
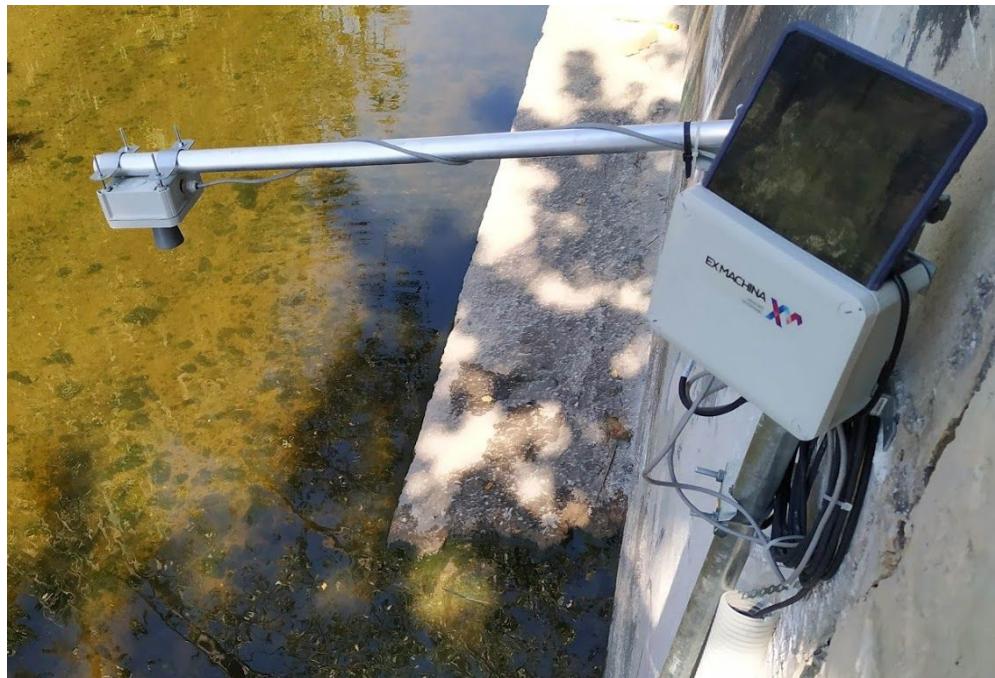
- Σταθμός μέτρησης ποιότητας υδάτων.

- Σταθμός μέτρησης υγρασίας εδάφους

Οι σταθμοί λειτουργούν αυτόνομα λαμβάνοντας μετρήσεις από τους αισθητήρες ανά τακτά χρονικά διαστήματα και τις αποστέλλουν ασύρματα με τελικό προορισμό την cloud πλατφόρμα στην οποία αποθηκεύονται.

Η επικοινωνία του σταθμού μέτρησης ποιότητας υδάτων επιτυγχάνεται μέσω GSM - GPRS οπότε και απαιτείται μια κάρτα SIM ανα σταθμό. Στην περίπτωση των σταθμών μέτρησης υγρασίας εδάφους, λόγω του γεγονότος ότι τοποθετούνται μαζικά στο πεδίο, για να διατηρηθεί το κόστος της κατασκευής και διατήρησης χαμηλό, επικοινωνούν με την πλατφόρμα με την τεχνολογία LoRaWAN χωρις να χρειαζεται SIM καρτα (GPRS). Αντί αυτού, απαιτείται ενα LoRaWAN gateway για να προωθεί τα δεδομένα των σταθμών μέσω του δικτύου κινητής τηλεφωνίας, στην πλατφόρμα. Κάθε LoRaWAN gateway μπορεί να εξυπηρετεί χιλιάδες σταθμούς αρκεί να βρίσκονται εντός της εμβέλειας του, και ολα μεσα απο μια μονο SIM κάρτα.

Eliot Node (v2)

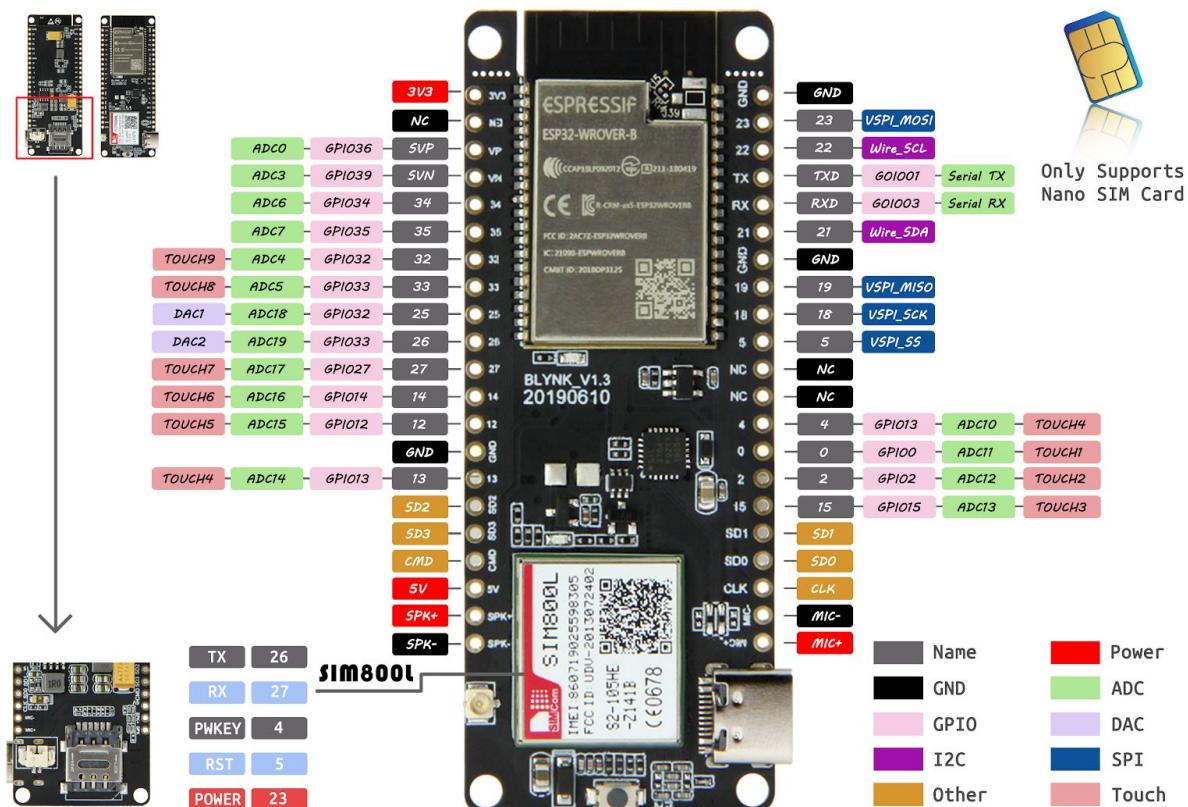


Σχήμα: Αρχιτεκτονική EIIoT Node

Το EIIoT node αποτελεί κεντρικό δομικό στοιχείο του κάθε σταθμού, στον οποίο συνδέονται όλοι οι αισθητήρες των οποίων τα δεδομένα καταγράφει, αποθηκεύει και αποστέλλει στην πλατφόρμα. Αποτελείται από κυκλώματα που του επιτρέπουν να επικοινωνεί μέσω ποικίλων πρωτοκόλλων με τους αισθητήρες, κυκλώματα διαχείρισης ενέργειας για αυτόνομη λειτουργία με φόρτιση των μπαταριών από τον ήλιο και υποστήριξη τεχνολογιών ασύρματης επικοινωνίας για αποστολή των δεδομένων.

Κεντρικός μικροελεγκτής και βασική πλακέτα

Στην προηγούμενη έκδοση υλικού χρησιμοποιήθηκε η πλακέτα μικροελεγκτή TTGo TFox η οποία όμως υστερούσε σε σημεία κρίσιμα για την λειτουργία του συστήματος πράγμα που την κατέστησε ακατάλληλη, και συνεπώς αντικαταστάθηκε με το TCall, το οποίο είναι του ιδίου κατασκευαστή. Το TCall περιέχει ενσωματωμένο το κύκλωμα SIM800 το οποίο είναι υπεύθυνο για την επικοινωνία με το δίκτυο κινητής τηλεφωνίας (GSM module) και έτσι αντικαθιστά ταυτόχρονα και το εξωτερικό GSM module SIM7000 της προηγούμενης έκδοσης μειώνοντας έτσι το συνολικό κόστος και αντικαθιστώντας 2 πλακέτες με μία.



Σχήμα: Πλακέτα TCall με το διάγραμμα συνδέσεων (pinout)

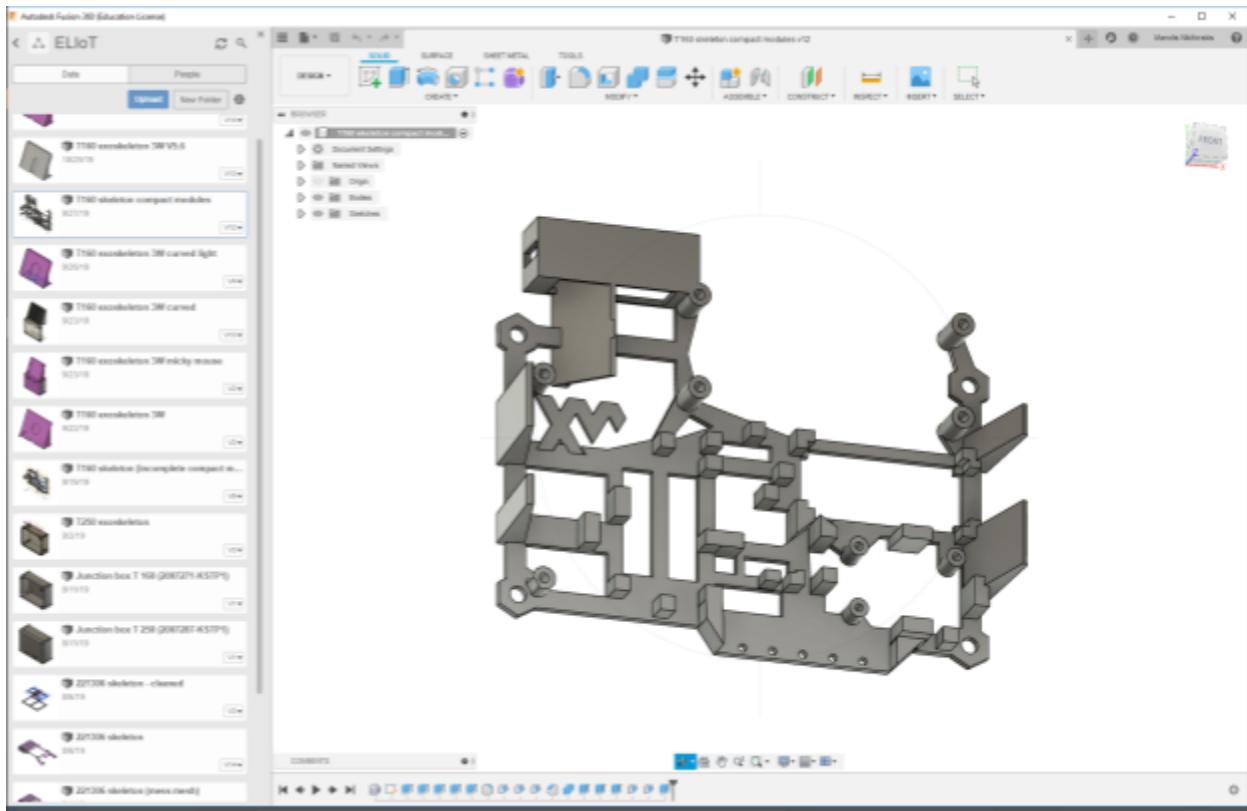
Hardware Specifications	
Chipset	ESPRESSIF-ESP32 240MHz Xtensa® dual-core 32-bit LX6 microprocessor
FLASH	QSPI flash 4MB / PSRAM 8MB
SRAM	520 kB SRAM
Button	Reset
USB to TTL	CP2104
Modular interface	UART, SPI, SDIO, I2C, LED PWM, TV PWM, I2S, IRGPIO, capacitor touch sensor, ADC, DACLNA pre-amplifier
On-board clock	40MHz crystal oscillator
Working voltage	2.7V-3.6V
Working current	About 70mA
Sleep current	About 300uA
SIM card	Only supports Nano SIM card
Working temperature range	-40°C ~ +85°C
Size & Weight	78.83mm*28.92mm*8.06mm(11.77g)
Power Supply Specifications	
Power Supply	USB 5V/1A
Charging current	500mA
Battery	3.7V lithium battery
JST Connector	2Pin 1.25mm
USB	Type-C
Wi-Fi	
Standard	FCC/CE-RED/IC/TELEC/KCC/SRRC/NCC
Protocol	802.11 b/g/n(802.11n, speed up to150Mbps)A-MPDU and A-MSDU

	polymerization, support 0.4μS Protection interval
Frequency range	2.4GHz~2.5GHz(2400M~2483.5M)
Transmit Power	22dBm
Communication distance	300m
bluetooth	
Protocol	meet bluetooth v4.2BR/EDR and BLE standard
Radio frequency	with -97dBm sensitivity NZIF receiver Class-1,Class-2&Class-3 emitter AFH
Audio frequency	CVSD&SBC audio frequency
Software specifications	
Wi-Fi Mode	Station/SoftAP/SoftAP+Station/P2P
Security mechanism	WPA/WPA2/WPA2-Enterprise/WPS
Encryption Type	AES/RSA/ECC/SHA
Firmware upgrade	UART download/OTA(Through network/host to download and write firmware)
Software Development	Support cloud server development /SDK for user firmware development
Networking protocol	IPv4, IPv6, SSL, TCP/UDP/HTTP/FTP/MQTT
User Configuration	AT + Instruction set, cloud server, android/iOSapp
OS	FreeRTOS

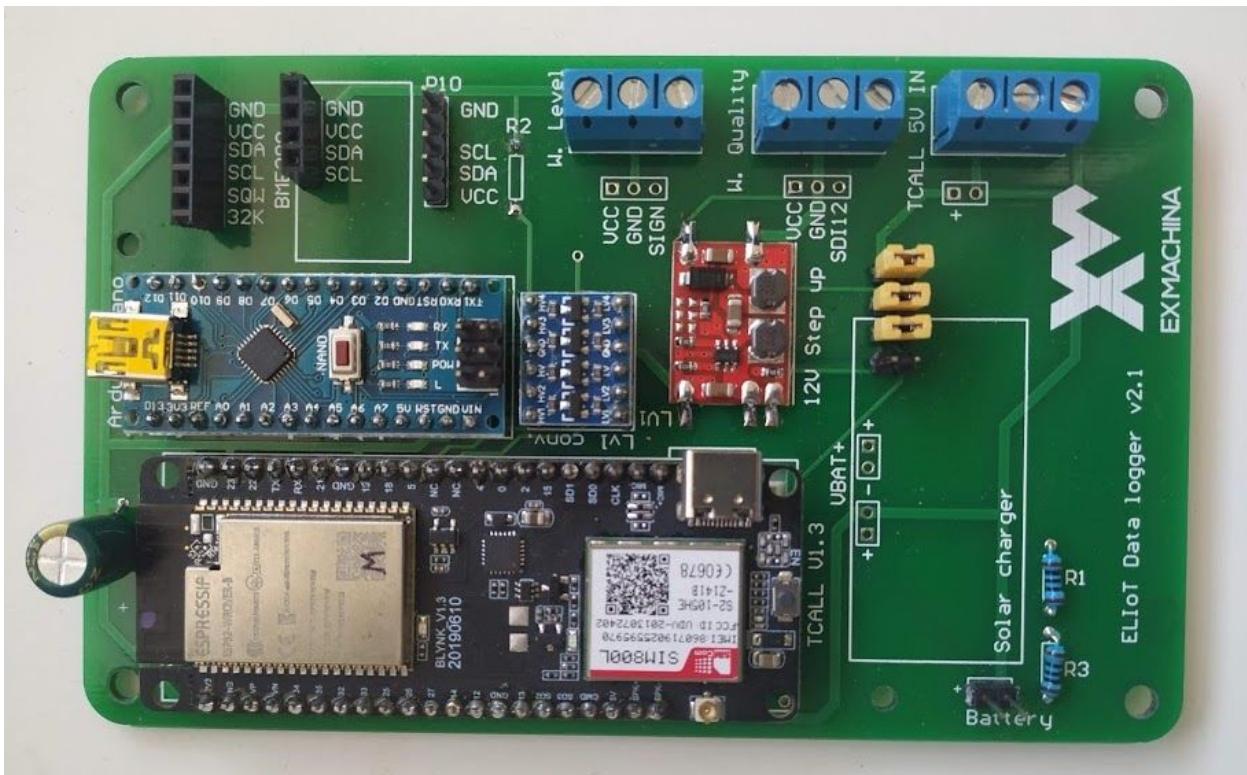
Πίνακας: Χαρακτηριστικά TTGo TCall

Επιπλέον σχεδιάστηκε κύκλωμα-πλακέτα στήριξης (carrier PCB) όπου τοποθετούνται όλες οι υποπλακέτες (modules) διευκολύνοντας κατά πολύ τη διαδικασία της συναρμολόγησης και την αξιοπιστία του συστήματος. Η πλακέτα βιδώνεται απευθείας μέσα στο περίβλημα και έτσι καταργείται η 3D εκτυπωμένη πλαστική βάση στήριξης για τα Modules και το κουτί OBO T160.

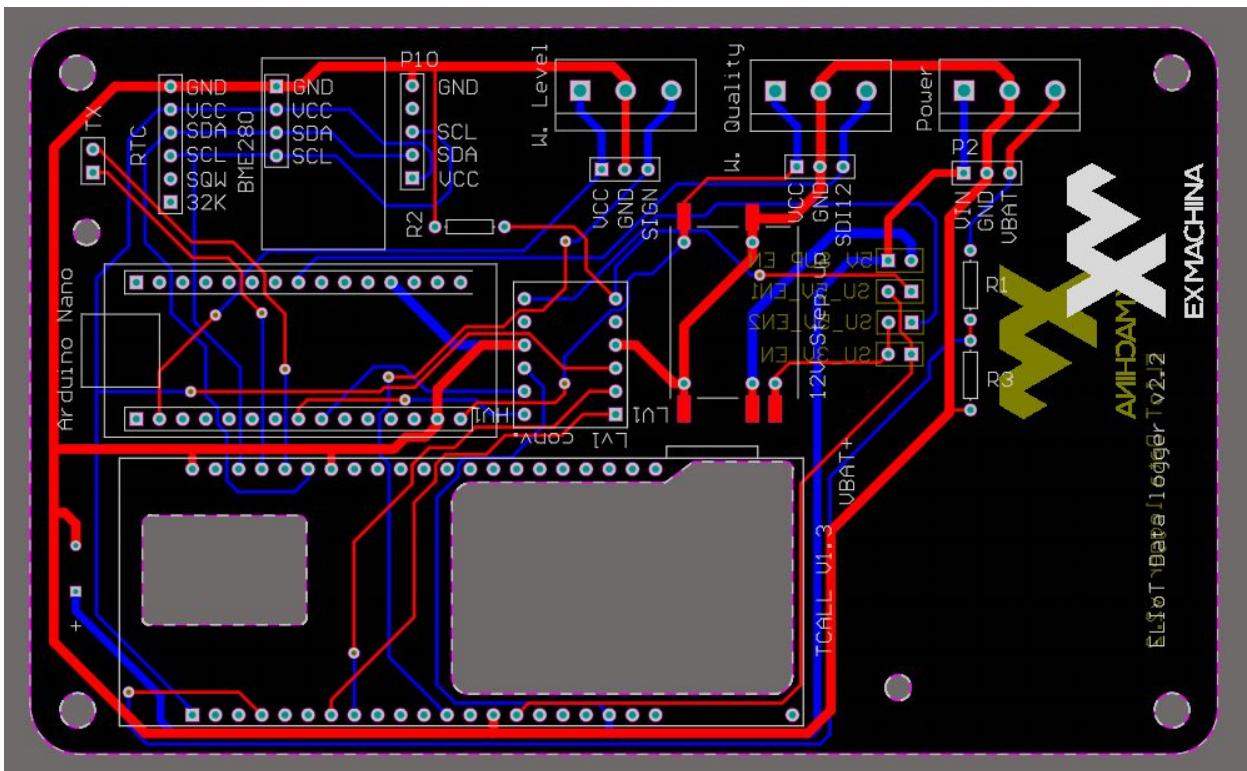




Σχήμα: 3D σκελετος για modules

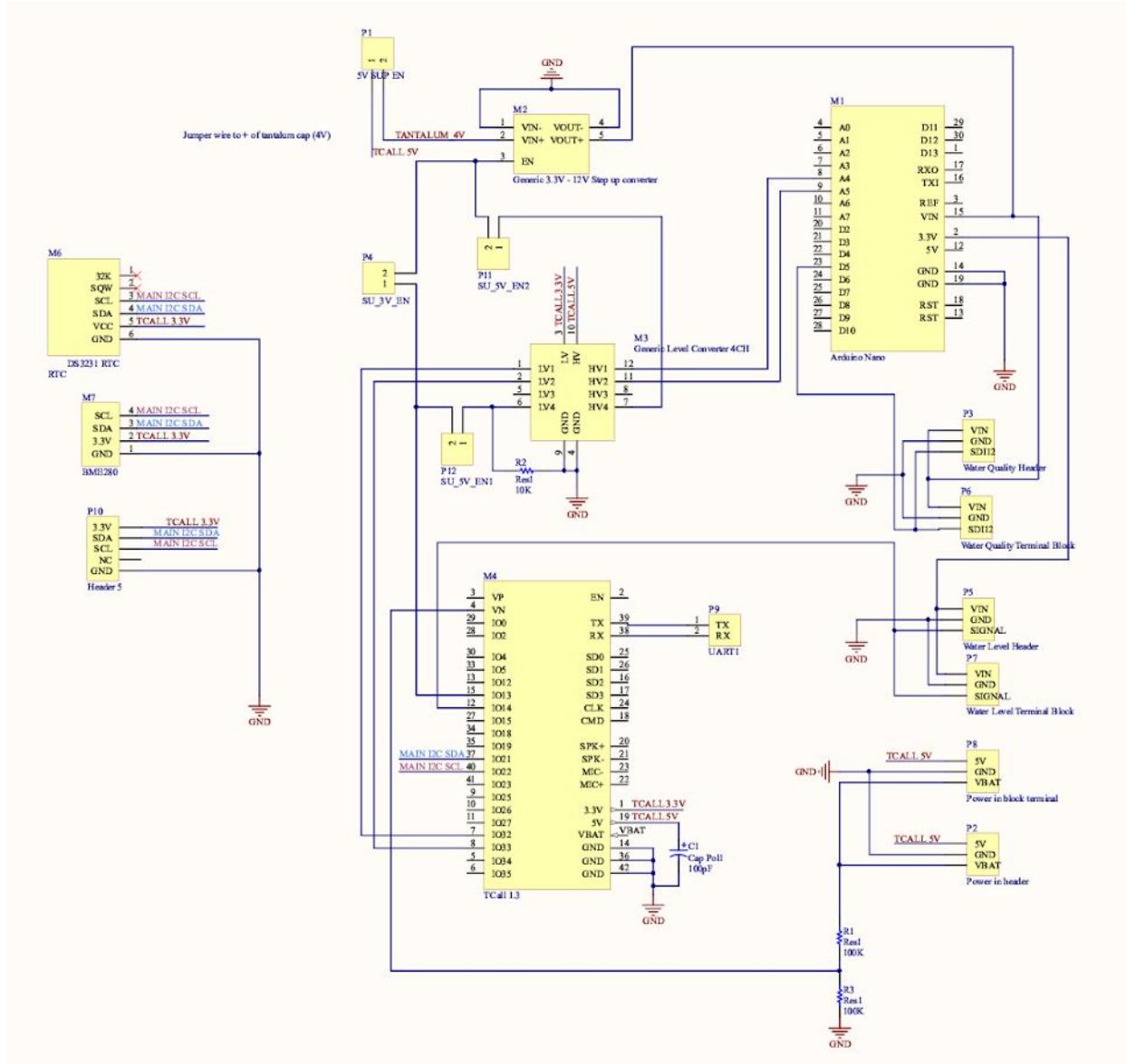


Σχήμα: Πλακέτα τυπωμένου κυκλώματος (PCB)



Σχήμα: Πλακέτα Node

Διάγραμμα σύνδεσης - συναρμολόγηση



Σχήμα: Σχηματικό Node

Επικοινωνίες GSM

Λόγω της μετάβασης της κεντρικής πλακέτας από TFox σε TCall, δεν υφίσταται η ανάγκη για εξωτερικό GSM module αφού το TCall περιέχει ενσωματωμένο module με παρόμοια χαρακτηριστικά και καλύπτει πλήρως τις ανάγκες της κατασκευής. Έτσι το εξωτερικό module SIM7000 που βρισκόταν σε ανεξάρτητη πλακέτα, αντικαταστάθηκε με το SIM800 που

λόγω της εγγύτητας του στον κεντρικό μικροεπεξεργαστή καθώς και της ανάλογης θωράκισης του τελευταίου, εξάλειψε τον θόρυβο στις επικοινωνίες και έτσι βελτιώθηκαν αισθητά οι επιδόσεις και η αξιοπιστία του συστήματος συνολικά.

Μετα από εκτεταμένη μελετη τεχνικων χαρακτηριστικων των παρακατω οικογενειων απο cellular modems:

- SIMCom SIM800 series
- SIMCom SIM900 series
- SIMCom LTE / NB-IoT / HSPA series
- AI-Thinker A6, A6C, A7, A20
- u-blox 2G / 3G / 4G / LTE series
- Sequans Monarch LTE Cat M1/NB1
- Quectel BG96, M95

Επιλεχθηκε το SIMCom **SIM7000** CAT-M1/NB-IoT Module ως η χρυση τομη δυνατοτητων και κοστους, το οποιο υποστηρίζει GSM, NB-IoT, LTE-CATM1 και GPS.



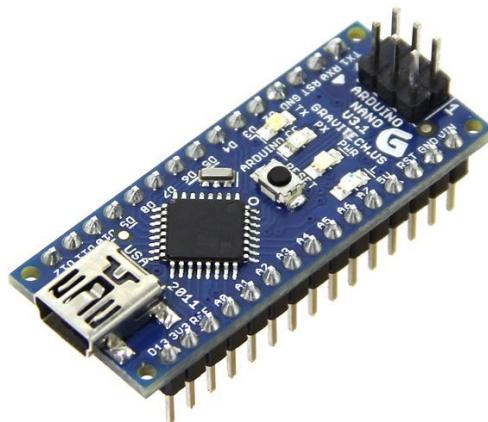
SIM7000 modules που χρησιμοποιηθήκαν στην αναπτυξή του ELIoT node v1

Πρωτοκόλλα επικοινωνίας με αισθητήρες SDI-12

Το SDI-12 είναι ένα ασύγχρονο σειριακό πρωτόκολλο και χρησιμοποιείται από τους περισσότερους (industrial) περιβαλλοντικούς αισθητήρες της αγοράς για την ενσύρματη επικοινωνία με ένα data logger με ή χωρις επικοινωνίες. Για την υλοποίηση του απαιτούνται

μόνο 3 αγωγοί (2 για τροφοδοσία και 1 κανάλι επικοινωνίας) και είναι σχετικά απλό στη χρήση του.

Στο ELIoT Node η υλοποίηση του πρωτοκόλλου γίνεται με τη βοήθεια της ανοιχτού κώδικα βιβλιοθήκης EnviroDIY η οποία τρέχει σε ένα Arduino Nano. Το Arduino Nano συνδέεται με τον κεντρικό μικροελεγκτή ESP32 μέσω διαύλου I2C και λειτουργεί ως ένας διαφανής διαμεσολαβητής που προωθεί τα δεδομένα αυτούσια από το ESP32 στον αισθητήρα μέσω του διαύλου SDI-12 και αντίστροφα. Η χρήση ενός Arduino Nano ως διαμεσολαβητή είναι απαραίτητη γιατί δεν υπάρχει διαθέσιμη υλοποίηση του πρωτοκόλλου SDI-12 συμβατή με ESP32.



Arduino nano, SDI-12 adapter

Ο κύκλος επικοινωνίας με τους αισθητήρες είναι αρκετά απλός. Αρχικά αποστέλλεται στον αισθητήρα εντολή έναρξης μέτρησης στην οποία ο αισθητήρας επιστρέφει ως απάντηση τον αριθμό των παραμέτρων που θα μετρήσει και τον χρόνο που θα διαρκέσει αυτή η μέτρηση. Ο κεντρικός μικροελεγκτής, αφού περάσει το χρονικό αυτό όριο, αποστέλλει εντολή ανάγνωσης των δεδομένων (αποτελεσμάτων) των μετρήσεων και λαμβάνει την ανάλογη απάντηση.

Αισθητήρας μέτρησης ποιότητας υδάτων

Ο πολυαισθητήρας ποιότητας υδάτων που υλοποιείται είναι ο Aqua TROLL 400 της In-situ και έχει την ικανότητα να μετράει εως 12 διαφορετικές παραμέτρους από 6 αισθητήρες. Είναι αρκετά χαμηλής κατανάλωσης (~16-75mA κατά τη διάρκεια της μέτρησης) ώστε να συνδέεται σε συστήματα υποστηριζόμενα από απλές μπαταρίες και επικοινωνεί με SDI-12.

Τροφοδοτείται με 8 - 36V, συνεπώς για να συνδεθεί στο σύστημα χρησιμοποιείται ένας step up converter που μετατρέπει την τάση του συστήματος (3.3V) σε 12V.

Πριν την πρώτη χρήση του αισθητήρα απαιτείται η αρχικοποίηση του με σύνδεση σε ηλεκτρονικό υπολογιστή μέσω του ειδικού μετατροπέα USB SDI-12 του κατασκευαστή και την εφαρμογή "Win-Situ". Η αρχικοποίηση αφορά την ενημέρωση του λογισμικού του στην τελευταία έκδοση και στην επιλογή των παραμέτρων που επιθυμούμε να μετρήσουμε, καθώς και τη σειρά με την οποία αυτοί οι παράμετροι θα επιστρέφονται όταν γίνεται ανάγνωση των αποτελεσμάτων των μετρήσεων στον αισθητήρα αμέσως μετά την ολοκλήρωση της μέτρησης. Το βήμα αυτό είναι απαραίτητο αφού η σειρά των παραμέτρων πρέπει να είναι συγκεκριμένη επειδή τα δεδομένα αποτελούνται μόνο από μια σειρά αριθμών χωρίς κάποια άλλη πληροφορία που αφορά την αντιστοίχιση τους στις παραμέτρους που περιγράφουν.



Σχήμα: Πολυαισθητήρας AquaTROLL 400

Αισθητήρας μέτρησης στάθμης

Η μέτρηση στάθμης νερού επιτυγχάνεται με τη χρήση του πομπού/δέκτη υπερήχων MB7066 XL-MaxSonar-WRL1 της MaxBotix. Τροφοδοτείται με 3.3V και η ανάγνωση της μέτρησης από αυτόν γίνεται μέσω αναλογικού καναλιού (PWM) όπου ο αισθητήρας παράγει παλμό σταθερής συχνότητας με πλάτος ανάλογο της απόστασης του από την επιφάνεια του νερού (58μS / cm). Το σήμα αυτό καταγράφεται από τον μικροελεγκτή και από το πλάτος του συμπεραίνεται το αποτέλεσμα. Λόγω της ψηφιακής φύσης του σήματος είναι λιγότερο επιρρεπές σε εξωτερικές παρεμβολές.

Σε προηγούμενη έκδοση ο αισθητήρας επικοινωνούσε με τον μικροελεγκτή αναλογικά, παράγοντας τάση ανάλογη της απόστασης του από τη στάθμη του νερού. Η μέθοδος αυτή ενώ λειτουργούσε φυσιολογικά, συχνά λόγω παρεμβολών από εξωτερικούς παράγοντες (ηλεκτρομαγνητική ακτινοβολία, θόρυβος στη γραμμή λόγω ασταθούς παροχής ενέργειας κ.α.) εισήγαγε σφάλματα στη μέτρηση με τη στάθμη να ταλαντεύεται συχνά γύρω από το πραγματικό νούμερο.



Σχήμα: Αισθητήρας στάθμης υπερηχων MaxBotix MB7066



Σχήμα: Αισθητήρας στάθμης σε αδιαβροχό κουτί

MB7066 Specifications:

Resolution of 1 cm
10Hz read rate
42kHz Ultrasonic sensor measures distance to objects

RoHS Compliant
Read from all 3 select sensor outputs: Analog Voltage, Serial, Pulse Width
Operates from 3.0-5.5V
Low 3.4mA average current requirement
Small, light weight module
Designed for easy integration into your project or product
Operational Temperature from -40°C to +70°C (-40°F to +160°F)
Real-time automatic calibration (voltage, humidity, ambient noise)
Firmware filtering for better noise tolerance and clutter rejection
200,000+ Hours Mean Time Between Failure
Weather Resistant (IP67), optional Chemical Resistant F-Option
Matches standard electrical 3/4-inch PVC pipe fittings for easy mounting (3/4" National Pipe Thread Straight)
Our longest range outdoor, weather resistant, ultrasonic sensor
Maximum range of 1068 cm (450 inches)
Long range, narrow detection zone

Μετεωρολογικός σταθμός

Fine Offset WH2900 / Ambient Weather WS-2902A είναι η πιό οικονομική επιλογή την οποία θα τροποποιήσουμε ώστε να συνδεθεί με το ELIoT Node και να στελνει μετρησεις στο ELIoT cloud.



WS-2902A external sensors all-in-one



Outdoor measurement specifications

Temperature range: -40~+60°C

Resolution: 0.1°C

Measuring range rel. humidity: 10%~99%

Accuracy : +/- 5%
Rain volume display: 0 – 9999.9mm
Resolution: 0.3mm (if rain volume < 1000mm)
1mm (if rain volume > 1000mm)
Wind speed: 0-50m/s (0~100mph)
Accuracy: +/- 1m/s (wind speed < 5m/s)
+/-10% (wind speed > 5m/s)
Wind direction: 0 to 359 degree
Light: 0-400k Lux
Accuracy : +/-15%

Measuring internally every 16 sec

Ο Meter Group Atmos 41 χρησιμοποιείται ως σταθμός αναφοράς ο οποίος προσφέρει υψηλή ακρίβεια και ανοιχτό πρωτόκολλο επικοινωνίας (SDI-12).



Metergroup Atmos 41

Measurements specifications:

Solar radiation
Range: 0 to 1750 W/m²
Resolution: 1 W/m²
Accuracy: ± 5% of measurement typical

Precipitation
Range: 0 to 400 mm/h
Resolution: 0.017 mm
Accuracy: ± 5% of measurement from 0 to 50 mm/h

VAPOR PRESSURE
Range
0 to 47 kPa
Resolution
0.01 kPa
Accuracy
Varies with temperature and humidity, ±0.2 kPa typical below 40 °C

RELATIVE HUMIDITY
Range
0 to 100% RH
Resolution
0.1% RH
Accuracy
Varies with temperature and humidity, ±3% RH typical

Air temperature
Range: -50 to 60 °C
Resolution: 0.1 °C
Accuracy: ± 0.6 °C

Humidity sensor temperature
Range: -40 to 50 °C
Resolution: 0.1 °C
Accuracy: ± 1.0 °C

Barometric pressure
Range: 50 to 110 kPa
Resolution: 0.01 kPa
Accuracy: ± 0.1 kPa from -10 to 50 °C, ± 0.5 kPa from -40 to 60 °C

Horizontal wind speed
Range: 0 to 30 m/s
Resolution: 0.01 m/s
Accuracy: the greater of 0.3 m/s or 3% of measurement

Wind gust
Range: 0 to 30 m/s
Resolution: 0.01 m/s
Accuracy: the greater of 0.3 m/s or 3% of measurement

Wind direction

Range: 0° to 359°

Resolution: 1°

Accuracy: ± 5°

Tilt

Range: -90° to +90°

Resolution: 0.1°

Accuracy: ± 1°

Lightning strike count

Range: 0 to 65,535 strikes

Resolution: 1 strike

Accuracy: variable with distance, >25% detection at <10km typical

Lightning average distance

Range: 0 to 40 km

Resolution: 3 km

Accuracy: variable

Ασυρματα πρωτοκόλλα επικοινωνιας με πλατφορμα

Αν και η πλατφορμα υποστηριζει MQTT / COAP το HTTP πρωτοκολλο ειναι αρκετο για τις ανάγκες των nodes.

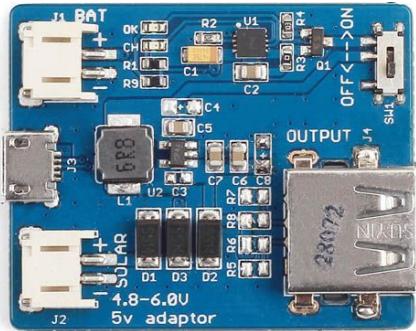
Διαχείριση ενέργειας

Οι μπαταρίες λιθίου 18650 προσφερουν μια καλη σχεση κοστους (~5€) διαθεσιμοτητας και χωρητικοτητας (~2500 - 3000 mAh)

Αναλογα το μεγεθος του ηλιακού πάνελ (2 ή 4/5 Watt) αντιστοιχεί και το πληθος των μπαταριων (2 ή 4) και με αυτο το τροπο εχουμε δυο επιλογες διαθεσιμης ενεργειας για τα nodes ανάλογα με τις ανάγκες των περιφερειακών αισθητήρων.

Κατά τις δοκιμές το ενσωματωμένο κύκλωμα διαχείρισης ενέργειας IP5306 του TCall παρουσίασε λειτουργικά προβλήματα που το καθιστούν ακατάλληλο για αυτόνομη λειτουργία. Συγκεκριμένα στην περίπτωση που τα αποθέματα ενέργειας της μπαταρίας εξαντληθούν πλήρως και σε μετέπειτα χρονικό διάστημα επανέλθουν σε λειτουργικά επίπεδα λόγω ηλιακής φόρτισης, απαιτείται ανθρώπινη επέμβαση για να γίνει επανεκκίνηση του χειροκίνητα, ειδάλλως η παροχή ενέργειας μένει ανενεργή παρόλο που υπάρχει αρκετή. Για αυτό το λόγο είναι ακατάλληλο για αυτόνομη λειτουργία σε απομακρυσμένες περιοχές όπου δεν υπάρχει η ευχέρεια ανθρώπινης παρέμβασης. Για την αποφυγή αυτής της κατάστασης το σύστημα διαχείρισης ενέργειας του TCall αντικαταστάθηκε με αντίστοιχο εξωτερικό module και συγκεκριμένα με το Seeed LiPo Rider, το οποίο διαθέτει ηλιακή φόρτιση και παράγει σταθερή τάση 5V στην έξοδό του. Το LiPo Rider συνδέεται απευθείας

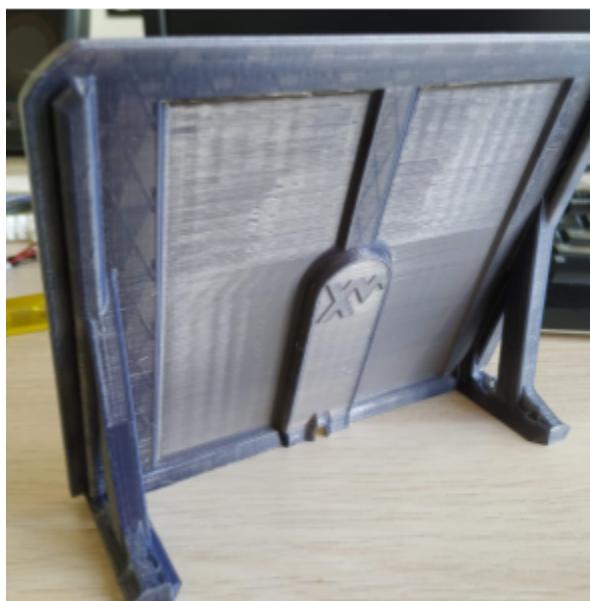
στη γραμμή των 5V του TCall και έτσι προσπερνάει τελείως το δικό του κύκλωμα διαχείρισης ενέργειας.



Σχήμα: LiPo Rider

Περίβλημα

Το κουτί με ενσωματωμένο ηλιακό πάνελ 2W ανεβάζει υψηλες θερμοκρασίες το καλοκαίρι σε πλήρη ηλιοφάνεια (π.χ. 60C). Για αυτό το λόγο αποφασίσαμε να μεταφέρουμε το ηλιακό πάνελ έξω από το κουτί, σχεδιάζοντας μια ειδική θήκη που προσαρμόζεται σε ενα οικονομικό αδιάβροχο κουτί, την οποία και εκτυπώνουμε σε ειδικό 3D printer με PETG filament. Το νέο solar panel είναι μεγαλυτερο 3W.



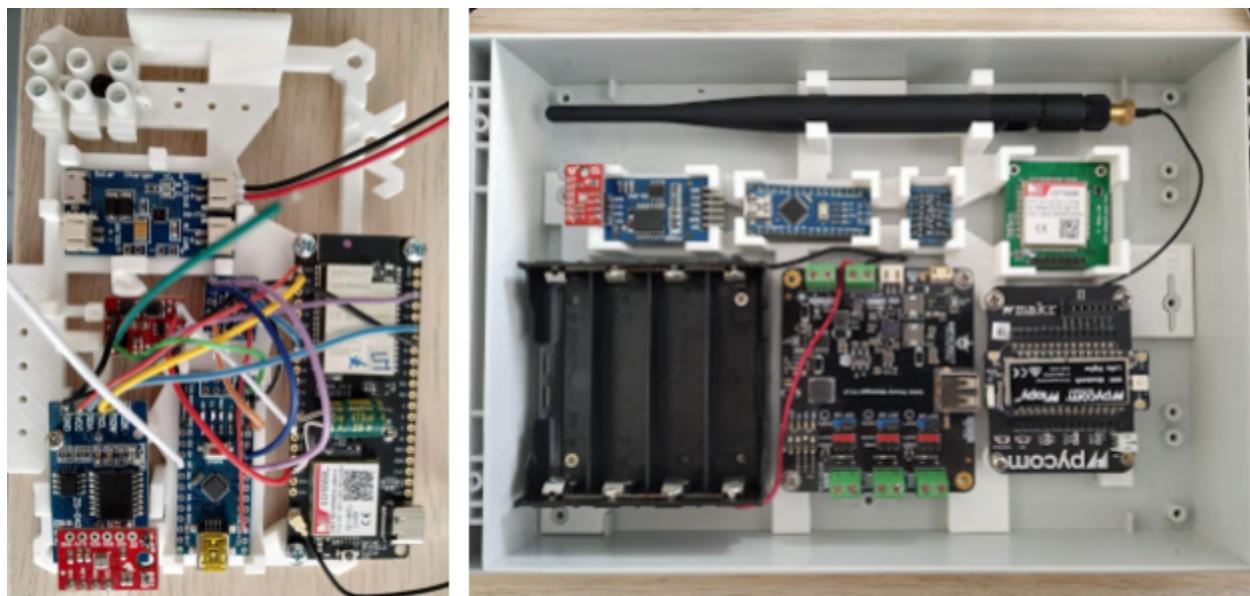
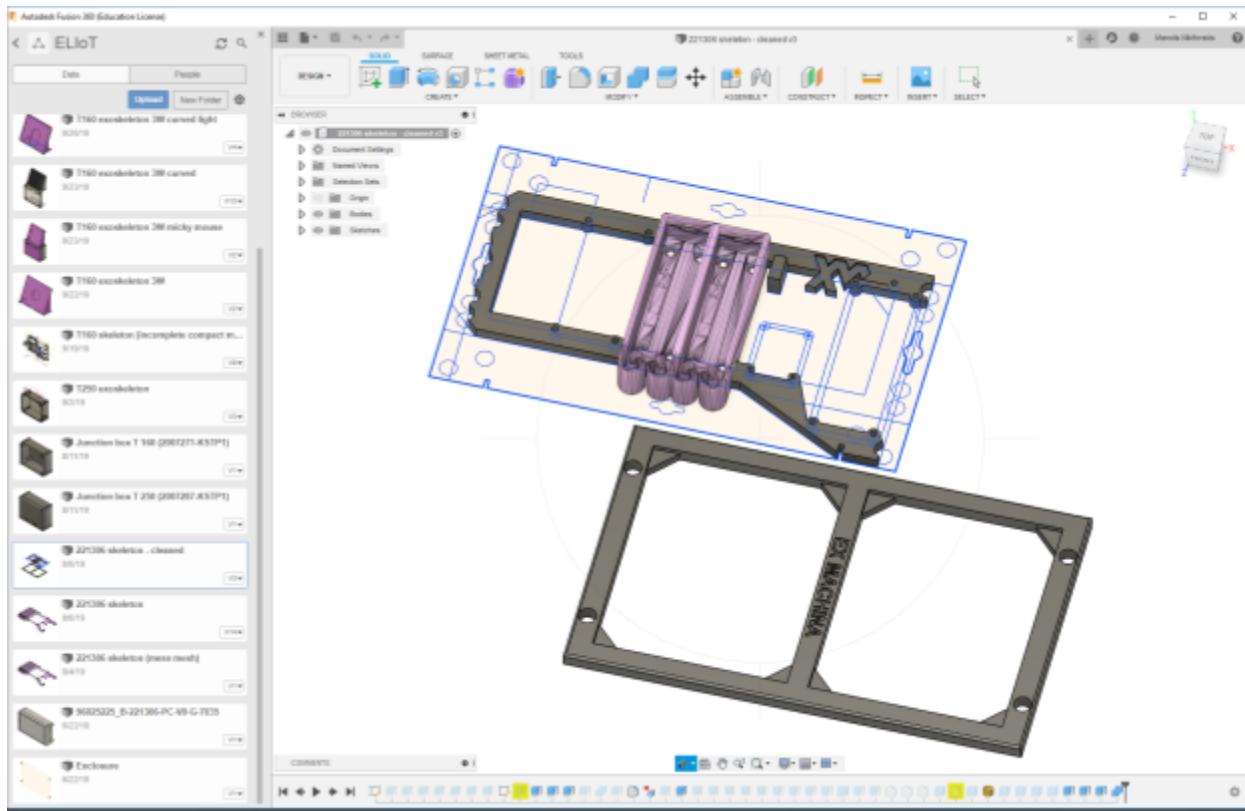
Σχήμα: Bopla vs OBO T160
+ 3D printed εξωτερικος σκελετος solar panel για OBO T160



Σχήμα: Τελικό κουτί OBO T160 με 3D printed 3W solar panel

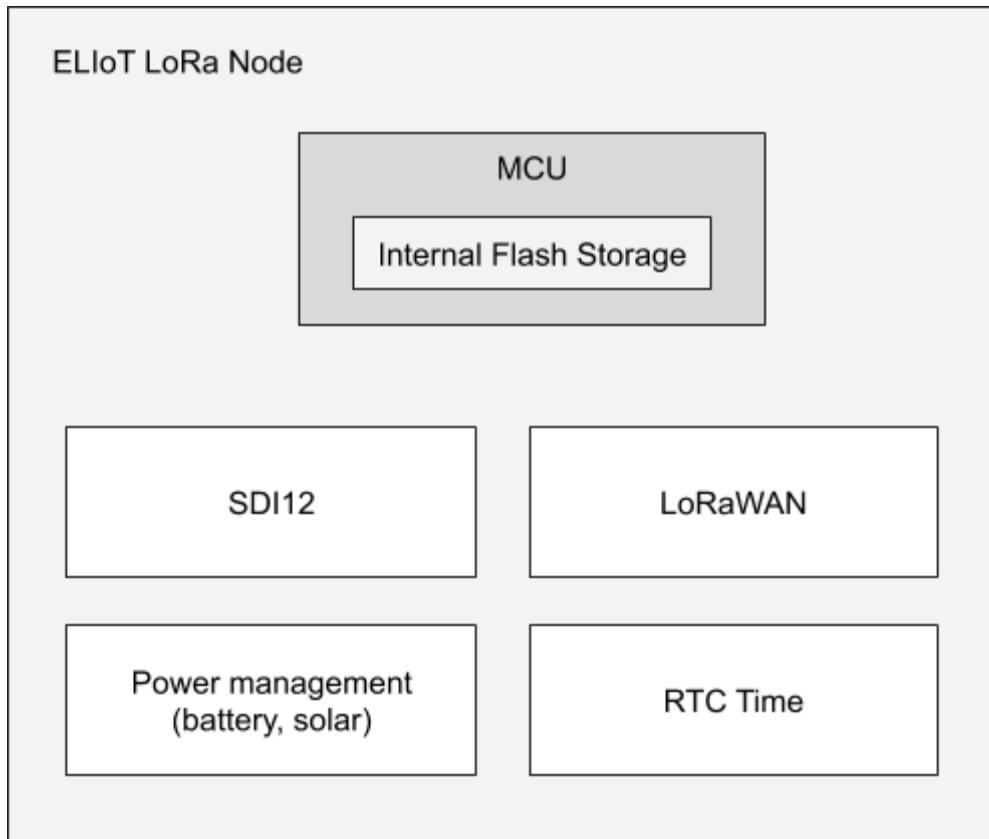
Βάση στήριξης πλακετών

Για να στηριχθούν οι επιμέρους πλακέτες της κατασκευής μέσα στο κουτί, σχεδιάστηκε πλαστική βάση με οπές στήριξης για την κάθε πλακέτα, η οποία εκτυπώθηκε σε υλικό PLA σε τρισδιάστατο εκτυπωτή. Η σχεδίαση έγινε με τη βοήθεια λογισμικού CAD (Fusion 360) και χρησιμοποιήθηκε 3D μοντέλο του κατασκευαστή του περιβλήματος ώστε να σχεδιαστεί στις κατάλληλες διαστάσεις και να δημιουργηθούν οι οπές για τη στήριξη της μέσα στο κουτί.



Σχήμα: 3D printed βάσεις στήριξης με τις πλακέτες τοποθετημένες

ELIoT LoRa Node

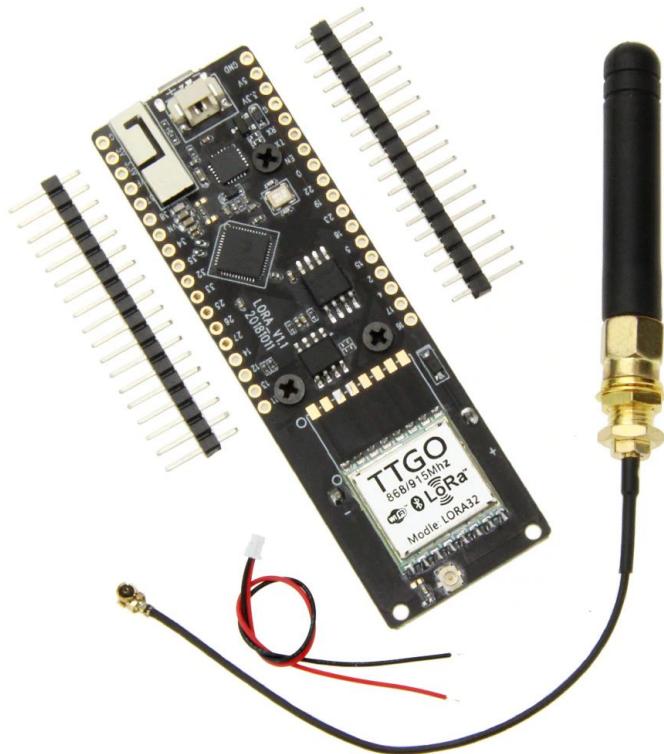


Σχήμα: Αρχιτεκτονική ELIoT LoRa Node

Το ELIoT LoRa Node είναι μια απλούστερη εκδοχή του ELIoT Node και χρησιμοποιεί το πρωτόκολλο επικοινωνίας LoRaWAN που είναι κατάλληλο για εφαρμογές που απαιτούν τοποθέτηση μεγάλου πλήθους κόμβων στην οποία περίπτωση η αποστολή των δεδομένων μέσω του δικτύου κινητής τηλεφωνίας θα ήταν διαχειριστικά δύσκολη και κοστοβόρα αφού για κάθε κόμβο απαιτείται μια κάρτα SIM και συνδρομή. Το πρωτόκολλο LoRaWAN είναι περιοριστικό ως προς τον όγκο δεδομένων που μπορούν να αποσταλούν (μερικές δεκάδες bytes) καθώς και στη συχνότητα των αποστολών, συνεπώς το ELIoT LoRa Node προορίζεται για σύνδεση με αισθητήρες των οποίων τα δεδομένα είναι μικρού όγκου πράγμα που το κάνει κατάλληλο για την εφαρμογή της μέτρησης υγρασίας εδάφους. Υποστηρίζει τη σύνδεση αισθητήρα SDI-12 του οποίου τα δεδομένα διαβάζει ανα τακτά χρονικά διαστήματα και αποστέλλει άμεσα χωρίς αποθήκευση στην εσωτερική μνήμη.

Κεντρικος μικροελεγκτης

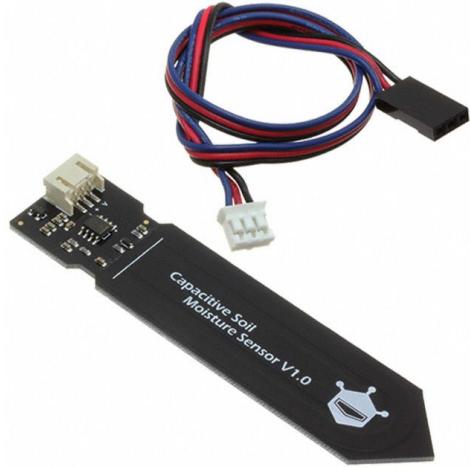
Όπως και στην περίπτωση του ELIoT Node χρησιμοποιειται η ίδια πλακέτα μικροελεγκτή TTGO TFox για λόγους συμβατότητας και απλοποίησης.



Σχήμα: TTGO TFox

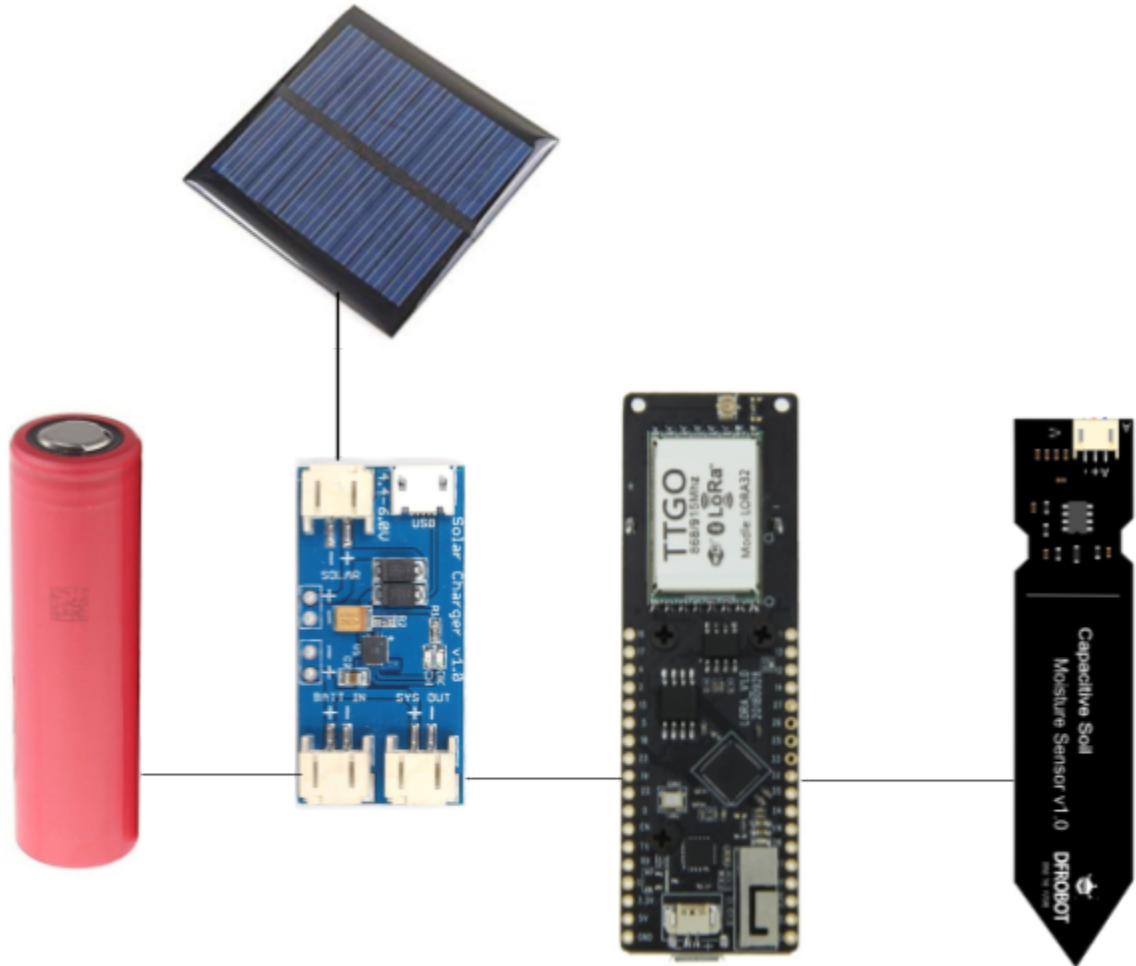
Αισθητήρας μέτρησης υγρασίας

Η μέτρηση της υγρασίας εδάφους επιτυγχάνεται με τον αισθητήρα "Capacitive Soil Moisture Sensor" της DFRobot ο οποίος χρησιμοποιεί την πυκνωτική μέθοδο για την μέτρηση της υγρασίας που τον περιβάλλει και παράγει στην έξοδο τάση ανάλογη της υγρασίας στο εύρος 1.2 - 2.5V, η οποία καταγράφεται από τον μικροελεγκτή.



Σχήμα: Αισθητήρας μέτρησης υγρασίας εδάφους

Διάγραμμα σύνδεσης - συναρμολόγηση



Σχήμα: Διάγραμμα σύνδεσης

LoRaWAN Gateway (RAK)

Για τη λήψη των δεδομένων από το EIIoT LoRa Node απαιτείται η ύπαρξη ενός gateway το οποίο θα προωθεί στο διαδίκτυο τα δεδομένα που θα λαμβάνει ασύρματα μέσω του πρωτοκόλλου LoRaWAN. Η σύνδεσή στο διαδίκτυο επιτυγχάνεται μέσω καλωδίου Ethernet ενώ σε δυσπρόσιτες περιοχές όπου η ενσύρματη σύνδεση δεν είναι δυνατή, μπορεί να συνδεθεί με 3G modem και έτσι να αποκτήσει πρόσβαση στο διαδίκτυο μέσω του δικτύου κινητής τηλεφωνίας.

Το RAK Outdoor Gateway παρέχει φιλικό διαχειριστικό περιβάλλον που είναι προσβάσιμο από οποιονδήποτε φυλλομετρητή ιστού (browser) με ενσωματωμένο LoRaWAN network server σε αντιθεση με το Lorix One που χρησιμοποιηθηκε στην εκδοση v1 παραδοτέο D2.1.

Η διεύθυνση IP που θα ανατεθεί στο Gateway όταν συνδεθεί στο δίκτυο δεν είναι γνωστή εκ των προτέρων και για αυτό ο διαχειριστής θα πρέπει να την ανακαλύψει με κάποιο τρόπο (π.χ. nmap, port scanning κλπ). Στρέφοντας το φυλλομετρητή στη διεύθυνση αυτή ενδέχεται να εμφανιστεί μήνυμα με προειδοποίηση για την ασφάλεια του ιστότοπου, όπου ο διαχειριστής θα πρέπει να κάνει τις απαραίτητες ρυθμίσεις για να παρακάμψει αυτόν τον περιορισμό και να αποκτήσει πρόσβαση στον ιστότοπο. Μετά από αυτα τα βήματα θα πρέπει να εμφανίζεται η αρχική σελίδα του RAK Gateway.



Στην αρχική σελίδα απαιτείται σύνδεση με το όνομα και τον κωδικό διαχειριστή, όπου τα προ-ρυθμισμένα στοιχεία χρήστη είναι username: **root** και password: **root**.

Για την ενεργοποίηση του ενσωματωμένου στο gateway LoRa server, απαιτείται η ενεργοποίηση του από το μενού LoRa Gateway -> LoRa Packet Forwarder. Εκεί στο tab General Setup πρέπει να καθοριστούν σωστά τα 2 παρακάτω πεδία:

- Protocol: πρέπει να επιλεγεί το **Built-in LoRa Server**
- Frequency Plan -> Region: **EU863-870**

Στη συνέχεια επιλέται το κουμπί “Save & Apply” ώστε να αποθηκευτούν και να ενεργοποιηθούν οι αλλαγές.

Στο μενού LoRa Network Server -> Application εμφανίζονται οι ενεργές εφαρμογές και επιλέγοντας μια από αυτές εμφανίζονται οι συσκευές που ανήκουν σε αυτή. Για την προώθηση των δεδομένων στο Thingsboard αρκεί να δημιουργηθεί ένα καινούριο Integration, επιλέγοντας Integrations -> Create. Σε αυτό το σημείο επιλέγεται το **Thingsboard.io** από το dropdown στην επιλογή Integration kind και στο πεδίο Thingsboard server συμπληρώνεται η διεύθυνση του.

Όπως αναφέρεται και στη σημείωση κάτω από τη διεύθυνση του Thingsboard server, κάθε device που δημιουργείται θα πρέπει να συνοδεύεται και από μια μεταβλητή με το όνομα **ThingsBoardAccessToken** και τιμή το **Access Token** του αντίστοιχου device στο Thingsboard server.

Διαχείριση ενέργειας

Όπως και στο ELIoT Node, οι μπαταρίες 18650 είναι πλέον κατάλληλες για τη συγκεκριμένη εφαρμογή και λόγω της χαμηλής κατανάλωσης που χαρακτηρίζει το LoRaWAN σε συνδυασμό με τη κατάλληλη διαχείριση στο υλισμικό, με μία μόνο μπαταρία αυτού του τύπου, το ELIoT LoRa Node μπορεί να λειτουργεί για πολύ μεγάλα χρονικά διαστήματα με μία μόνο μπαταρία, ακόμα κι όταν η ηλιοφάνεια είναι περιορισμένη. Όταν υπάρχει ηλιοφάνεια η μπαταρία φορτίζει μέσω των ενσωματωμένων φωτοβολταϊκών και αν αυτή είναι ικανοποιητική, μια μόνο μέρα φόρτισης μπορεί να προσφέρει εβδομάδες έως και μήνες αυτονομίας.

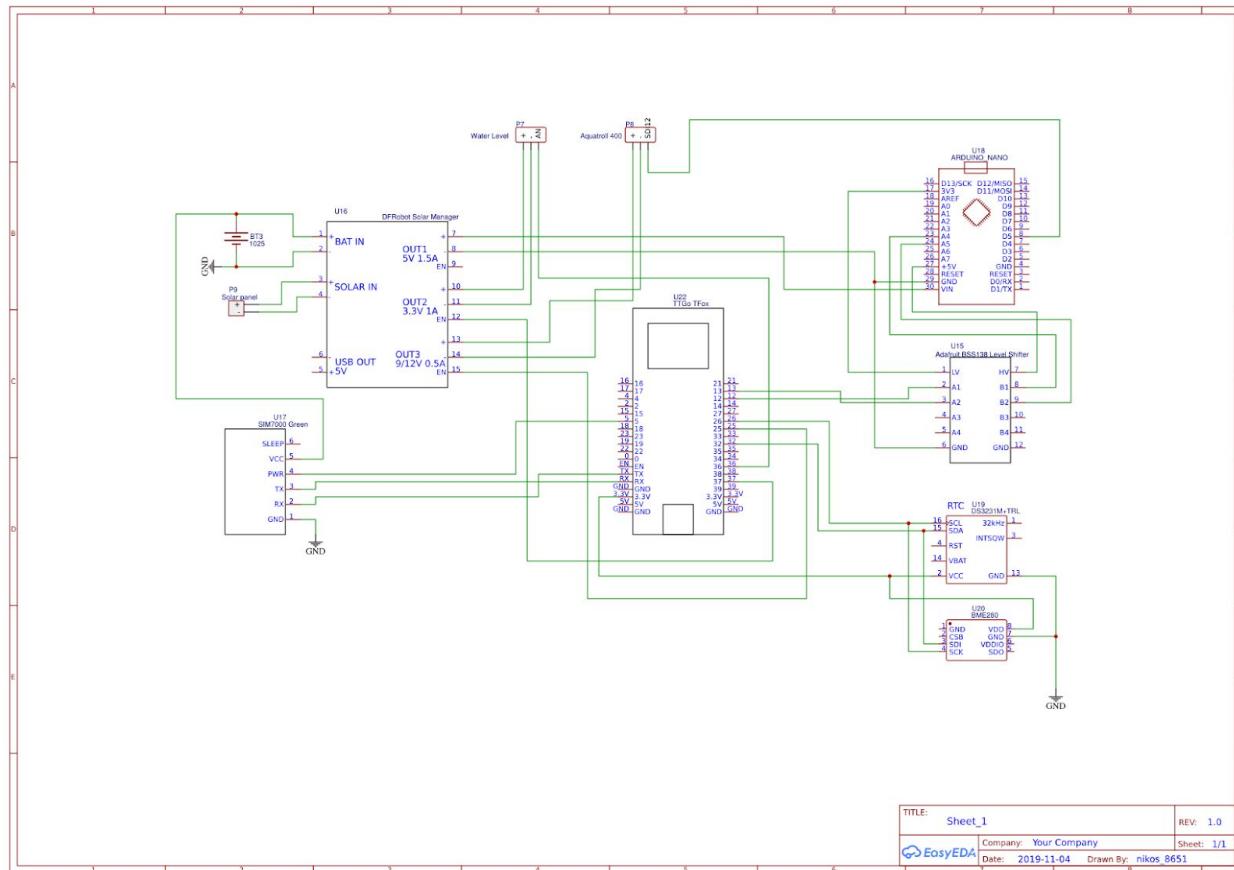
Περίβλημα

Λόγω της απλότητας του ELIoT LoRa Node αλλά και χαμηλής απαίτησης σε ενέργεια, το σύνολο των εξαρτημάτων μπορεί να χωρέσει σε μικρό κουτί χαμηλού κόστους στην επιφάνεια του οποίου μπορεί να τοποθετηθεί φωτοβολταϊκό ισχύος 0,5W.



Σχήμα: Περίβλημα ELIoT LoRa Node

Διάγραμμα συνδεσης - συναρμολόγηση



Σχήμα: Πλήρες σχηματικό κατασκευής

Η συναρμολόγηση της κατασκευής αποτελείται από σειρά πλακετών (modules), κάποιες από τις οποίες έχουν βιοηθητικό ρόλο. Παρακάτω αναλύεται η σύνδεση και η χρήση τους.

TTGo TFox

Η πλακέτα του κεντρικού μικροελεγκτή η οποία ελέγχει όλες τις βιοηθητικές.

DFRobot Solar Manager

Υπεύθυνο για την φόρτιση των μπαταριών από ηλιακή ενέργεια και την παραγωγή όλων των τάσεων που χρειάζονται για να λειτουργίσουν τα επιμέρους modules μέσω των 3 εξόδων που διαθέτει (3,3V, 5V, 12V).

Η καθεμία από τις εξόδους ξεχωριστά μπορεί να ανοιγοκλείνει από τον κεντρικό μικροελεγκτή και χρησιμοποιούνται ως εξής:

- 3,3V - Τροφοδοσία του αισθητήρα στάθμης νερού
- 5V - Τροφοδοσία του Arduino Nano που χρησιμοποιείται ως SDI-12 adapter.
- 12V - Τροφοδοσία του αισθητήρα ποιότητας νερού

SIM7000

Χρησιμοποιείται για τη σύνδεση στο δίκτυο κινητής τηλεφωνίας και την αποστολή των δεδομένων. Επικοινωνεί με τον κεντρικό μικροελεγκτή μέσω διαύλου UART και η τροφοδοσία του προέρχεται απευθείας από τη μπαταρία αφού διαθέτει δικό της κύκλωμα διαχείρισης ενέργειας. Μπορεί να χρησιμοποιηθεί οποιοδήποτε SIM7000 module της αγοράς.

Arduino Nano 5V

Τυπική πλακέτα μικροελεγκτή Arduino Nano στην έκδοση των 5V. Στη θέση της μπορεί να χρησιμοποιηθεί και το Arduino Pro Mini 5V.

Βρίσκεται μεταξύ της κεντρικής πλακέτας TFox και του αισθητήρα ποιότητας υδάτων και είναι υπεύθυνο για την υλοποίηση του πρωτοκόλλου SDI 12 για την άντληση δεδομένων από τον αισθητήρα και έπειτα την αποστολή του στη πλακέτα κεντρικού μικροελεγκτή μέσω διαύλου I2C.

Level Shifter

Μεταφράζει τα logic levels του κεντρικού μικροελεγκτή ESP32 από τα 3.3V στα 5V του Arduino Nano ώστε να μπορούν να επικοινωνούν μέσω του διαύλου I2C. Μπορεί να χρησιμοποιηθεί οποιοδήποτε αμφίδρομο level shifter της αγοράς.

DS2312 RTC

Πλακέτα χρονομέτρησης που λειτουργεί ως η βασική πηγή ώρας για το σύστημα. Μπορεί να χρησιμοποιηθεί οποιοδήποτε πλακέτα που είναι βασισμένη στο κύκλωμα DS2312.

BME280

Πλακέτα αισθητήρα εσωτερικών συνθηκών του περιβλήματος (θερμοκρασία, υγρασία, πίεση). Η ύπαρξή του είναι προαιρετική και βοηθάει στην αποσφαλμάτωση.

Υλισμικό (Firmware)

Το υλισμικό είναι ανεπτυγμένο για την πλατφόρμα Arduino και κάνει χρήση μιας σειράς ανοιχτών βιβλιοθηκών για τη λειτουργία των διάφορων μερών του και την υλοποίηση των πρωτοκόλλων που χρησιμοποιεί.

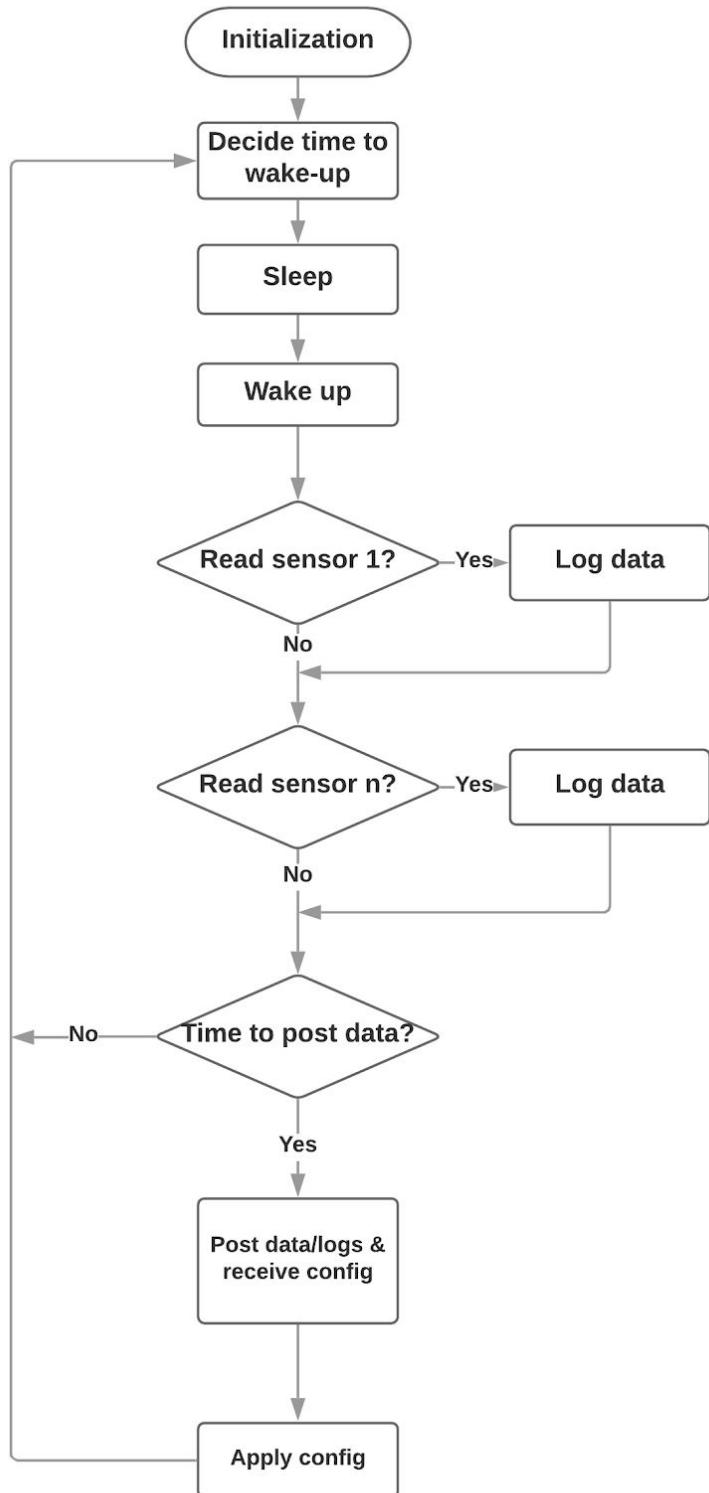
Εκκίνηση

Κατά την εκκίνηση το node αρχικοποιεί όλες τις παραμέτρους του και κάνει απόπειρα συγχρονισμού του RTC μέσω GPRS (NTP). Αφού ολοκληρωθεί η αρχικοποίηση, το node μπαίνει στον φυσιολογικό κύκλο λειτουργίας.

Κύκλος Λειτουργίας

Τον περισσότερο χρόνο του κύκλου λειτουργίας, το node βρίσκεται σε λειτουργία ύπνου προκειμένου να ελαχιστοποιήσει την κατανάλωση ενέργειας και ξυπνάει μόνο σύμφωνα με το πρόγραμμα αφύπνισης που καθορίζει τα χρονικά διαστήματα στα οποία θα πρέπει να εκτελεστεί η διαδικασία μέτρησης και της επικοινωνίας με την πλατφόρμα. Το χρονικά διαστήματα αυτά καθορίζονται ξεχωριστά για την καθεμία διαδικασία από τον χρήστη της πλατφόρμας.

Μετά από την κάθε αφύπνιση ελέγχεται ο λόγος της αφύπνισης και εκτελείται η ανάλογη διαδικασία (πχ. μέτρηση αισθητήρων) και αφού αυτοί/αυτές ολοκληρωθούν, υπολογίζεται ο χρόνος ως την επόμενη αφύπνιση και το node ξαναμπαίνει σε λειτουργία ύπνου για το διάστημα που υπολογίστηκε.



Σχήμα: Κύκλος λειτουργίας

Επικοινωνία με την πλατφόρμα

Ανα καθορισμένα χρονικά διαστήματα εκτελείται η επικοινωνία με την πλατφόρμα μέσω του δικτύου κινητής τηλεφωνίας. Κατα τη διαδικασία αυτή αποστέλλονται τα καταγραφηθέντα από τους αισθητήρες δεδομένα, καθώς και καταγραφές (logs) από τη λειτουργία της συσκευής που περιλαμβάνουν πληροφορίες για τη λειτουργία της (μπαταρία, εσωτερική θερμοκρασία) καθώς και τυχόν σφάλματα κατα τη λειτουργία που μπορούν να βοηθήσουν στην αποσφαλμάτωση. Σε περίπτωση αποτυχίας αποστολής, τα δεδομένα παραμένουν για αποστολή στην επόμενη επικοινωνία με την πλατφόρμα.

Ταυτόχρονα λαμβάνονται από την πλατφόρμα δεδομένα απομακρυσμένου ελέγχου μέσω των οποίων ο χρήστης έχει τη δυνατότητα να εκτελέσει:

- Αλλαγή των χρονικών διαστημάτων της μέτρησης και της αποστολής
- Επανεκκίνηση της συσκευής
- Διαμόρφωση της εσωτερικής μνήμης αποθήκευσης
- Ενημέρωση υλισμικού εξ αποστάσεως (FOTA)

Μέτρηση παραμέτρων αισθητήρων

Κατά τη διαδικασία μέτρησης των αισθητήρων, οι αισθητήρες ενεργοποιούνται και εκτελούν διαδικασίες μέτρησης. Τα αποτελέσματα της μέτρησης επιστρέφονται από τους αισθητήρες μαζί με CRC checksum (όπου υπάρχει η δυνατότητα) και επιβεβαιώνεται η εγκυρότητα τους για αποφυγή αποθήκευσης κατεστραμμένων δεδομένων λόγω πιθανού προβλήματος κατά την επικοινωνία με τον αισθητήρα. Τα δεδομένα αποθηκεύονται στην εσωτερική μνήμη μαζί με το αντίστοιχο CRC checksum ώστε να ελέγχονται για την εγκυρότητά τους και πριν την αποστολή τους ώστε να εντοπιστεί πιθανή καταστροφή των δεδομένων από προβλήματα της μνήμης.

Μνήμη και διαχείριση αρχείων

Το ESP32 διαθέτει μνήμη flash η οποία χρησιμοποιείται για την αποθήκευση του ίδιου του υλισμικού αλλα και των δεδομένων του node, και είναι χωρισμένη σε partitions που προορίζονται για διαφορετική χρήση το καθε. Ένα από τα partitions χρησιμοποιείται ως αποθήκη δεδομένων με το σύστημα αρχείων SPIFFS, το οποίο προσφέρει βασική διαχείριση αρχείων με υποστήριξη wear levelling για επέκταση της ζωής της.

Τα δεδομένα των τρεχουσών ρυθμίσεων του node αποθηκεύονται σε ξεχωριστό partition κάνοντας χρήση του Parameters API του Arduino. Ο διαχωρισμός των ρυθμίσεων της συσκευής από το σύστημα αρχείων προστατεύει την ορθή λειτουργία της συσκευής σε περίπτωση σφαλμάτων στο τελευταίο. Έτσι σε περίπτωση καταστροφής του συστήματος

αρχείων μπορεί να εκτελεστεί διαμόρφωση (format) του partition και να συνεχιστεί κανονικά η λειτουργία.

Καταγραφή συμβάντων (Logging)

Καθ' όλη τη διάρκεια εκτέλεσης του υλισμικού διαγνωστικά, σημαντικά συμβάντα και σφάλματα καταγράφονται από το σύστημα καταγραφής (logs) στο σύστημα αρχείων. Τα δεδομένα καταγραφής αποτελούνται από την ώρα του συμβάντος, τον κωδικό σφάλματος και σχετικά με αυτόν δεδομένα. Οι καταγραφές αποστέλλονται στην πλατφόρμα κατα την διαδικασία επικοινωνίας με αυτή και βοηθούν στην παρακολούθηση της λειτουργίας του node αλλα και στην διάγνωση σφαλμάτων αν αυτά προκύψουν.

Ώρα

Για την ορθή λειτουργία του συστήματος είναι απαραίτητο να υπάρχει καταγραφή της ώρας. Για τον σκοπό αυτό χρησιμοποιείται εξωτερικό RTC (Real Time Clock) module, που αναλαμβάνει την καταγραφή του χρόνου, με βοηθητική παροχή ενέργειας μέσω μπαταρίας ρολογιού, το οποίο συγχρονίζεται κατά την εκκίνηση του node μέσω του πρωτοκόλλου NTP. Επιπλέον σε κάθε αφύπνιση του node για εκτέλεση οποιασδήποτε διαδικασίας, γίνεται έλεγχος της ώρας για τυχόν μηδενισμό της (π.χ. λόγω προβληματικής βοηθητικής μπαταρίας) στην οποία περίπτωση γίνεται συγχρονισμός μέσω NTP εκ νέου.

Τέλος, αν παρόλες τις ενέργειες δεν υπάρχει η έγκυρη ώρα στο σύστημα, οι καταγραφές των αισθητήρων αναβάλλονται για να αποφευχθούν καταγραφές με μη έγκυρη ώρα.

FOTA (Firmware Over The Air)

Για την διόρθωση σφαλμάτων ή επέκταση των λειτουργιών του node, είναι δυνατή η αποστολή νέου υλισμικού εξ αποστάσεως το οποίο γράφεται στη μνήμη του node και ξεκινάει να λειτουργεί άμεσα.

Αρχικά ο χρήστης αναπτύσσει το νέο λογισμικό και το κάνει compile στον υπολογιστή του. Το αποτέλεσμα του compilation είναι ένα binary αρχείο το οποίο ο χρήστης πρέπει να ανεβάσει σε κάποιον server ώστε αυτό να είναι προσβάσιμο μέσω internet. Έπειτα αφού συμπληρώσει στη πλατφόρμα τη διεύθυνση του αρχείου αυτού μαζί με το αντίστοιχο md5 checksum, δίνει απομακρυσμένα εντολή στο node να εκτελέσει τη διαδικασία FOTA. Όταν το node επικοινωνήσει με την πλατφόρμα, θα λάβει το καινούριο υλισμικό και θα το αποθηκεύσει σε δεύτερο partition που υπάρχει ειδικά για την αποθήκευση υλισμικού και θα προχωρήσει σε επανεκκίνηση του συστήματος και εκτέλεση από το partition αυτό.

Το node πλεον εκτελεί τη νέα έκδοση του υλισμικού, όμως για να μπορέσει να συνεχίσει θα πρέπει να περάσει επιτυχώς το OTA self test αμέσως μετά την πρώτη εκκίνησή του. Κατα το self test γίνεται απόπειρα λήψης δεδομένων από την πλατφόρμα ως έλεγχος για να

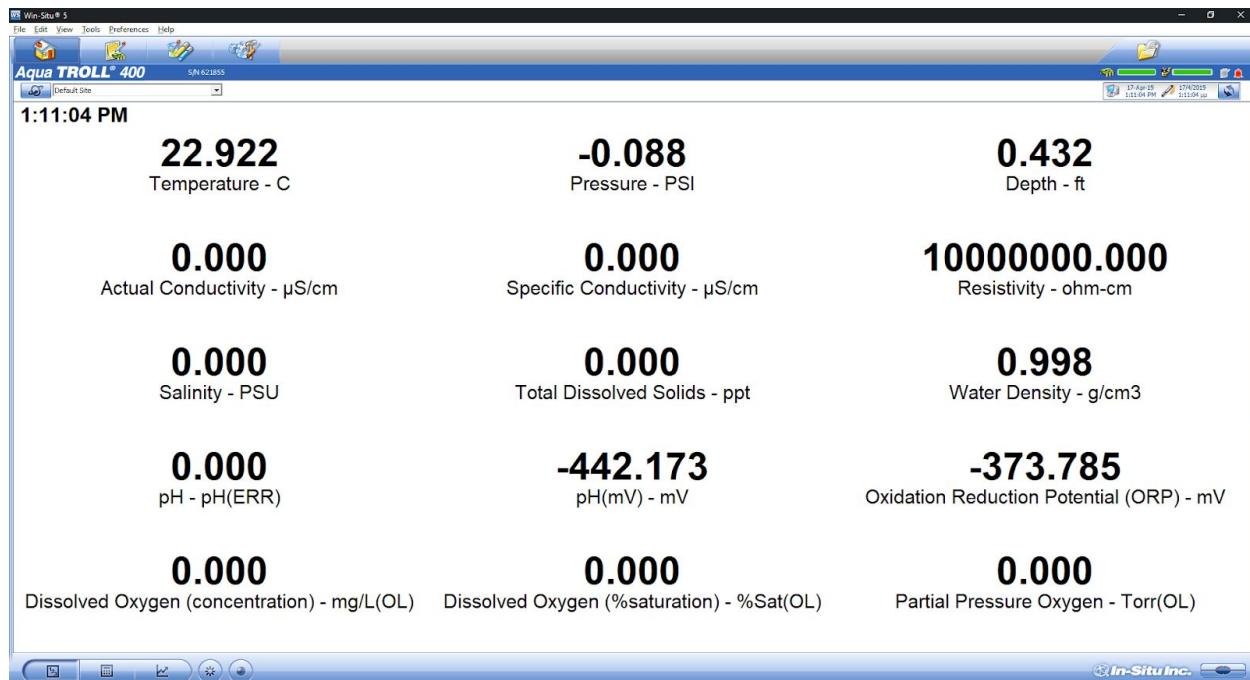
διαπιστωθεί αν το καινούριο λογισμικό μπορεί να επικοινωνεί ακόμα με την πλατφόρμα έτσι ώστε σε περίπτωση που το τελευταίο είναι προβληματικό, να υπάρχει τουλάχιστον η δυνατότητα να εκτελεστεί ξανά ο απομακρυσμένος έλεγχος ώστε να του δοθεί εκ νέου η εντολή για επόμενο λογισμικό. Σε περίπτωση που το self test αποτύχει, το λογισμικό επαναφέρεται στην προηγούμενη έκδοση και γίνεται άμεσα επανεκκίνηση σε αυτό. Η επαναφορά στην προηγούμενη έκδοση είναι δυνατή λόγω της ύπαρξης 2 partitions υλισμικού, έτσι ανα πάσα στιγμή στην μνήμη υπάρχουν 2 εκδόσεις του.

Αρχικοποίηση συσκευής

Αισθητήρας ποιότητας εδάφους AquaTROLL 400

Όπως προαναφέρθηκε, ο AquaTROLL απαιτεί κάποια προρύθμιση για να εξάγει τα κατάλληλα δεδομένα και με τη σωστή σειρά ώστε να μπορούν να αναγνωστούν από το Node. Η ρύθμιση του AquaTROLL γίνεται με τη σύνδεση σε υπολογιστή με τη βοήθεια του ειδικού μετατροπέα USB - SDI-12 της ίδιας εταιρείας και του λογισμικού Win-Situ που υπάρχει στην επίσημη ιστοσελίδα της.

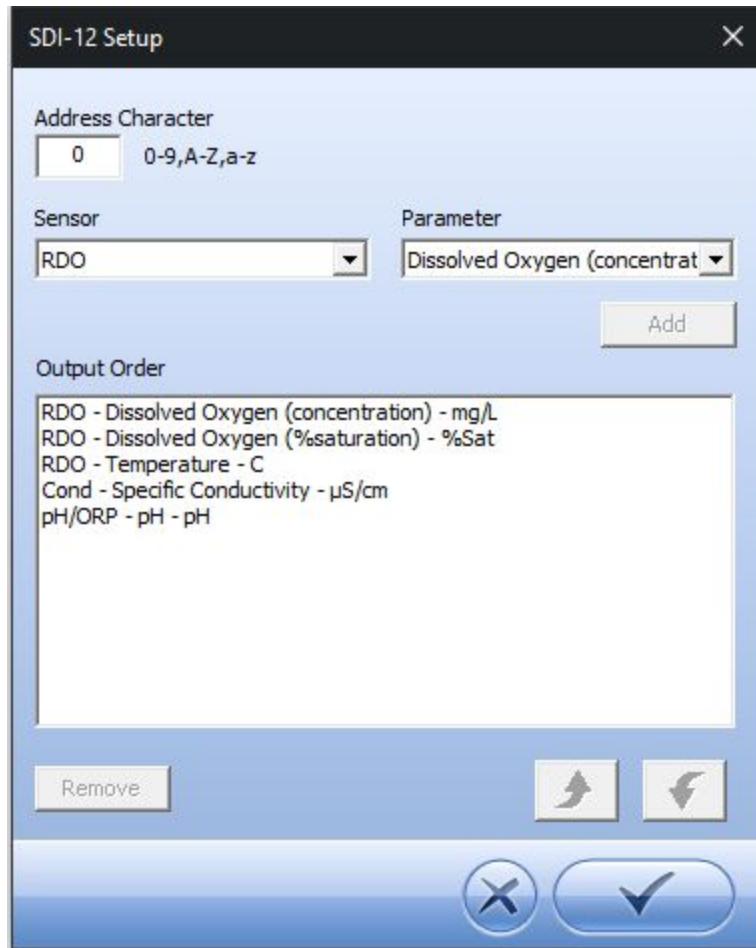
Εκκινώντας την εφαρμογή, αν η σύνδεση με τον αισθητήρα ήταν επιτυχής εμφανίζεται η αρχική οθόνη όπου και φαίνονται οι τρέχουσες τιμές των παραμέτρων όπως φαίνεται στο σχήμα.



Σχήμα: Αρχική οθόνη Win-Situ με τον αισθητήρα συνδεδεμένο

Κατά την πρώτη εκκίνηση ενδέχεται ο χρήστης να ερωτηθεί εαν επιθυμεί να προγραμματιστεί ο αισθητήρας στην τελευταία έκδοση του λογισμικού. Προτείνεται η αναβάθμιση να εκτελεστεί.

Από το μενού ρύθμισης παραμέτρων του αισθητήρα, επιλέγονται οι παράμετροι και ορίζεται η σειρά τους όπως στο παρακάτω σχήμα. Η ρύθμιση αυτή αφορά τη σειρά με την οποία ο αισθητήρας θα επιστρέψει τις παραμέτρους αυτές όταν ερωτάται από το Node και συνεπώς είναι απαραίτητο ο αισθητήρας να αποστέλλει τις τιμές με τη σειρά που αναμένει το Node ώστε να μπορούν να αντιστοιχιθούν στα σωστά πεδία.



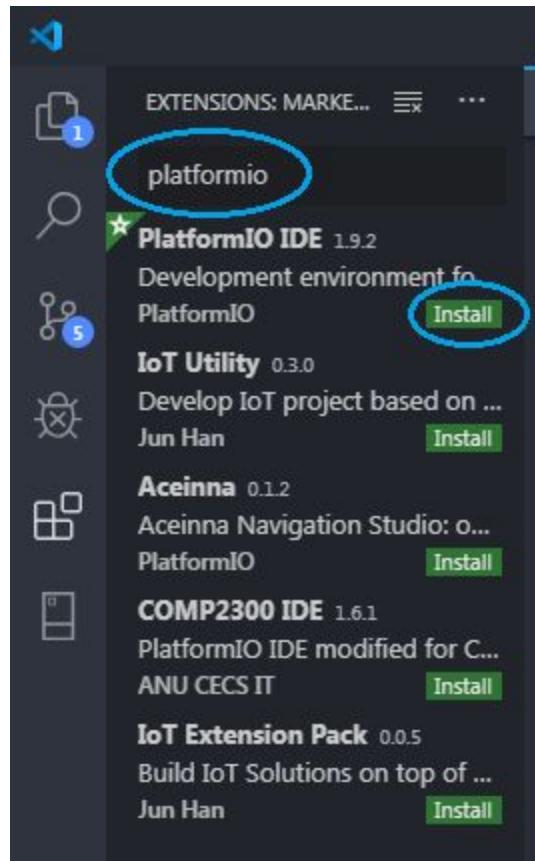
Σχήμα: Παράμετροι αισθητήρα και η σειρά που πρέπει να έχουν

Προγραμματισμός firmware

Για να προγραμματιστεί το υλισμικό πρέπει αρχικά να μεταγλωτιστεί σε δυαδικό αρχείο το οποίο θα εγγραφεί στη μνήμη του μικροελεγκτή. Η διαδικασία αυτή μπορεί να εκτελεστεί με διάφορα λογισμικά, στην παρούσα περίπτωση χρησιμοποιήθηκε το περιβάλλον ανάπτυξης

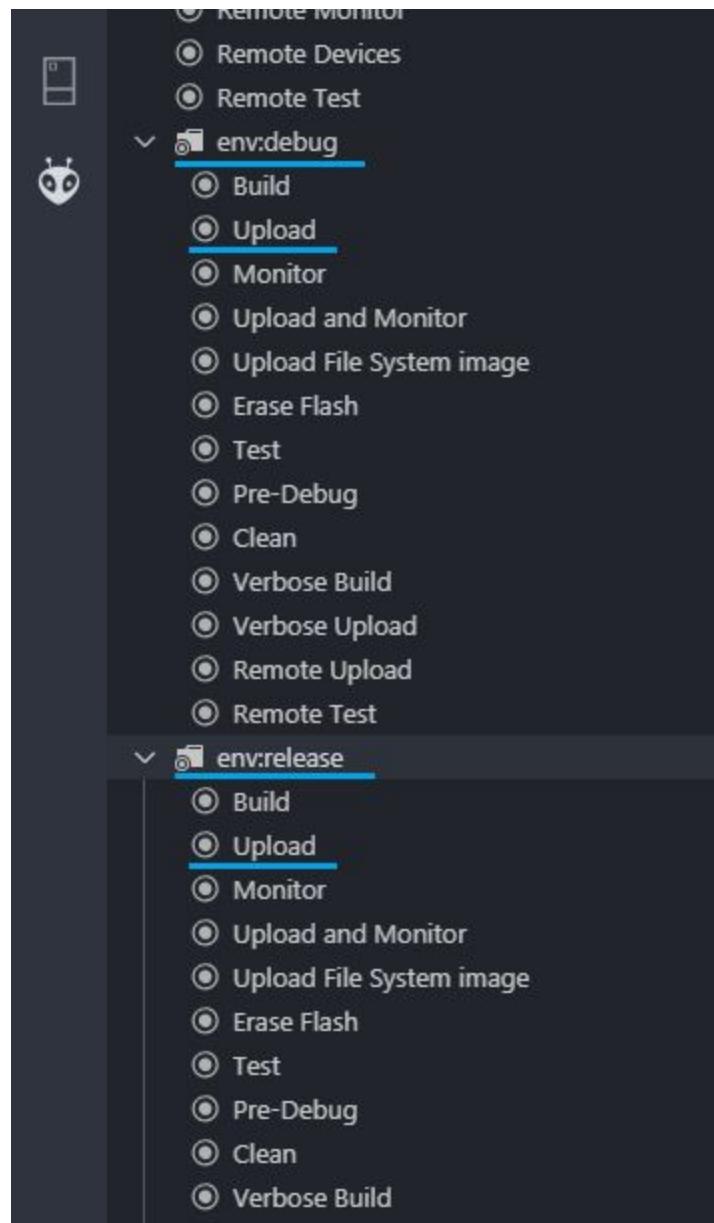
“Visual Studio Code” (ή VSCode) με το πρόσθετο “PlatformIO” το οποίο προσθέτει τη δυνατότητα προγραμματισμού μικροελεγκτών. Ο συνδυασμός των 2 αυτών ανοικτών λογισμικών αποτελεί σύνηθισμένη λύση στην ανάπτυξη εφαρμογών για την πλατφόρμα Arduino και όχι μόνο.

Αρχικά εγκαθίσταται το VSCode από την επίσημη σελίδα της εφαρμογής (<https://code.visualstudio.com/>). Όταν ολοκληρωθεί η εγκατάσταση και αφού ανοίξει η εφαρμογή, μέσω της μπάρας “Extensions” (View -> Extensions) αναζητείται και εγκαθίσταται το “PlatformIO”.



Σχήμα: Εγκατάσταση PlatformIO IDE από τη μπάρα Extensions του VSCode

Μετά την εγκατάσταση απαιτείται επανεκκίνηση του VSCode, η οποία όταν πραγματοποιηθεί και εφόσον η εγκατάσταση ήταν επιτυχής, θα εμφανιστεί επιπλέον εικονίδιο στην βασική μπάρα εργαλείων. Πατώντας το εμφανίζονται όλες οι διαθέσιμες εντολές της καινούριας προσθήκης.



Σχήμα: Εικονίδιο και μενού PlatformIO

Με εγκατεστημένα πλέον τα απαιτούμενα λογισμικά, ο κώδικας του υλισμικού πρέπει να προστεθεί ως project. Πατώντας στο το βασικό μενού File -> Open folder”, επιλέγεται ο φάκελος στον οποίον βρίσκεται ο κώδικας του υλισμικού όπως κατέβηκε από το αποθετήριο (repository). Το αποθετήριο περιλαμβάνει όλα τα αρχεία που χρειάζεται για να αναγνωριστεί ως project από το PlatformIO και να μεταγλωτιστεί στο κατάλληλο δυαδικό αρχείο.

Συνδέοντας την πλακέτα μικροελεγκτή στον υπολογιστή μέσω της θύρας USB, είναι έτοιμη για προγραμματισμό. Πριν από αυτό όμως θα πρέπει να οριστεί το όνομα της σειριακής θύρας στην οποία είναι συνδεδεμένη, ρύθμιση η οποία βρίσκεται καταχωρημένη στο αρχείο “platformio.ini” του project. Το όνομα της θύρας ορίζεται στη μεταβλητή “upload_port” στην ενότητα “[common]”.

```
11 [board_config]
12 ;name = wipy3
13 name = tcall
14 ;name = feather
15 ;name = wipy
16
17 [platformio]
18 env_default = debug
19
20 [common]
21 ; platform = espressif32@1.7.0
22 platform = espressif32
23 board = esp-wrover-kit
24 framework = arduino
25 monitor_speed = 115200
26 upload_speed = 921600
27 upload_port = /dev/ttyUSB0
28
```

Σχήμα: Ορίζοντας τη θύρα προγραμματισμού

Τέλος, μέσω του μενού “PlatformIO” της βασικής μπάρας, επιλέγοντας την επιλογή “Upload” γίνεται η μεταγλώττιση και έπειτα η εγγραφή/προγραμματισμός στη μνήμη του μικροελεγκτή.

Τέλος επιλέγοντας την κατάλληλη επιλογή “Upload” από τις ενότητες **env:debug** ή **env:release** μεταγλωτίζεται και εγγράφεται/προγραμματίζεται στον μικροελεγκτή η ανάλογη έκδοση και διαφοροποιούνται ως εξής:

- **Debug** - Κατάλληλη όταν το node προορίζεται για δοκιμαστική λειτουργία η αποσφαλμάτωση. Σε αυτή τη λειτουργία η συσκευή τυπώνει πληροφορίες λειτουργίας και αποσφαλμάτωσης στη σειριακή θύρα ενώ ταυτόχρονα εφαρμόζει ειδικές ρυθμίσεις που διευκολύνουν τον χρήστη στο να εντοπίσει τυχόν προβλήματα.
- **Release** - Έκδοση κατάλληλη για την κανονική λειτουργία της συσκευής όταν αυτή προορίζεται για τοποθέτηση στο πεδίο.

Εαν επιλεχθεί η επιλογή “upload” χωρίς να προσδιοριστεί κάποια από τις 2 εκδόσεις, θα προγραμματιστεί από προεπιλογή η έκδοση debug.

Κατά τη διάρκεια της διαδικασίας, η πρόοδος εμφανίζεται στην κονσόλα στο κάτω μέρος της οθόνης.

The screenshot shows a terminal window with the following output:

```
PROBLEMS 60 OUTPUT DEBUG CONSOLE TERMINAL

Writing at 0x00038000... (78 %)
Writing at 0x0003C000... (85 %)
Writing at 0x00040000... (92 %)
Writing at 0x00044000... (100 %)
Wrote 457712 bytes (223500 compressed) at 0x00010000 in 4.4 seconds (effective 827.1 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
===== [SUCCESS] Took 32.94 seconds =====

Environment Status Duration
-----
debug SUCCESS 00:00:32.939
release IGNORED
win IGNORED
=====
===== 1 succeeded in 00:00:32.939 =====

Terminal will be reused by tasks, press any key to close it.
```

Σχήμα: Επιτυχής ολοκλήρωση της διαδικασίας όπως φαίνεται στην κονσόλα

Σύνδεση με πλατφόρμα

Για να επικοινωνεί η συσκευή με την πλατφόρμα Thingsboard απαιτούνται ενέργειες και στον πίνακα διαχείρισης της πλατφόρμας αλλά και στο αρχείο ρυθμίσεων του υλισμικού. Αναλυτικότερα, για να αναγνωρίζεται η συσκευή από την πλατφόρμα, αρχικά πρέπει να καταχωρηθεί ως συσκευή σε αυτή και το μοναδικό κλειδί (access token) της καινούριας αυτής καταχώρησης να προστεθεί στο υλισμικό το οποίο έπειτα μεταγλωτίζεται και προγραμματίζεται στον μικροελεγκτή. Με τις ενέργειες αυτές κάθε επικοινωνία της συγκεκριμένης συσκευής με την πλατφόρμα θα προωθείται στην αντίστοιχη καταχώρηση συσκευής σε αυτή, με την αντίστοιχη να επιτυγχάνεται μέσω του μοναδικού κλειδιού. Τέλος πρέπει να οριστεί η διεύθυνση και η θύρα από την οποία είναι προσβάσιμη η πλατφόρμα.

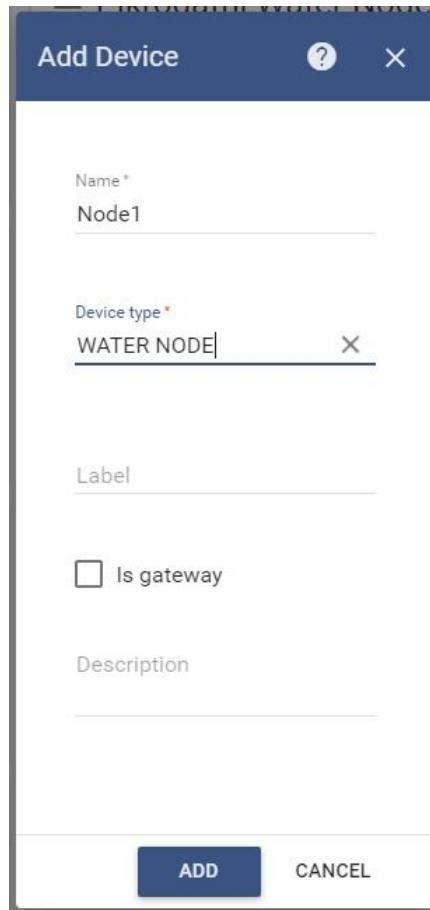
Πρόσθεση της συσκευής στην πλατφόρμα

Ανοίγοντας την ενότητα "Devices" εμφανίζονται οι ήδη καταχωρημένες συσκευές. Η προσθήκη νέας συσκευής ξεκινάει πατώντας το εικονίδιο "+" στην κάτω δεξιά γωνία της οθόνης και συμπληρώνοντας τα παρακάτω πεδία:

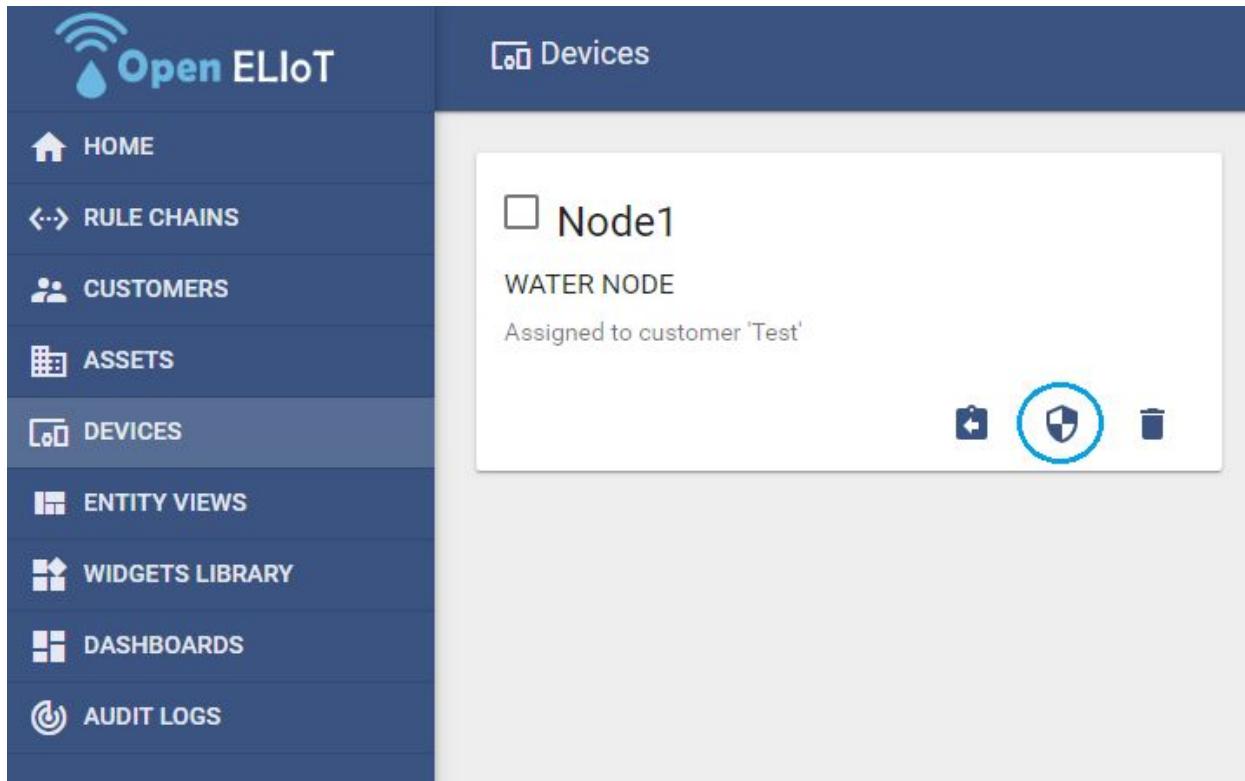
- **Name** - Το όνομα της συσκευής για ευκολότερη αναγνώριση από τον χρήστη της πλατφορμάς. Μπορεί να συμπληρωθεί με οτιδήποτε επιθυμεί ο χρήστης.

- Device type - Ότιος της συσκευής ομαδοποιεί όμοιες συσκευές κάτω από ένα όνομα, το οποίο μετέπειτα μπορεί να βοηθήσει στην εκτέλεση ενεργειών που αφορούν ολόκληρη την ομάδα συσκευών (π.χ. οπτικοποίηση των δεδομένων όλων των συσκευών που ανήκουν σε μια ομάδα ταυτόχρονα). Αν δεν υπάρχουν ήδη τύποι συσκευών προς επιλογή σε αυτό το πεδίο, πληκτρολογώντας ένα όνομα αυτή η ομάδα θα δημιουργηθεί και θα είναι διαθέσιμη προς επιλογή κατα την δημιουργία επόμενης συσκευής.

Πατώντας “Add” η συσκευή δημιουργείται και η διαδικασία ολοκληρώνεται.



Σχήμα: Παράθυρο δημιουργίας νέας συσκευής



Σχήμα: Ενότητα “Devices” με την καινούρια συσκευή

Πατώντας το εικονίδιο “Device credentials” όπως φαίνεται στο σχήμα, εμφανίζεται το μοναδικό αναγνωριστικό της συσκευής (device token) που θα καταχωρηθεί στο υλισμικό και θα χρησιμοποιείται από αυτή για πιστοποίηση όταν θα επικοινωνεί με την πλατφόρμα.

Ρυθμίσεις σύνδεσης στο υλισμικό

Οι ρυθμίσεις που αφορούν τη συνδεσιμότητα του node στην πλατφόρμα βρίσκονται στο αρχείο **include/credentials.h**, το οποίο πρέπει να ενημερωθεί με τις κατάλληλες ρυθμίσεις πριν προγραμματιστεί ο μικροελεγκτής. Είναι χωρισμένο σε 2 ενότητες, **Debug** και **Release**, η καθεμία από τις οποίες έχει το δικό της σετ των ίδιων ρυθμίσεων που εφαρμόζονται ανάλογα με το ποια έκδοση επιλέχθηκε να μεταγλωτιστεί. Με αυτόν τον τρόπο σε περίπτωση δοκιμαστικής λειτουργίας το node μπορεί να προγραμματιστεί με την έκδοση debug και έτσι να επικοινωνεί με διαφορετική εγκατάσταση της πλατφόρμας από ότι οταν μπει στο πεδίο σε κανονική λειτουργία (έκδοση release).

Σε αυτό το αρχείο συμπληρώνονται τα στοιχεία για το σύνολο των nodes που θα χρησιμοποιηθούν και οχι για ένα μονο node. Κατά την εκκίνηση, το node αντλεί τα στοιχεία του με βάση την αντιστοίχιση που έγινε σε αυτό το αρχείο με τη διεύθυνση MAC του και έτσι γνωρίζει ποια στοιχεία του αντιστοιχούν. Με αυτόν τον τρόπο όταν υπάρχει πλήθος nodes, αποφεύγεται ο προγραμματισμός διαφορετικού υλισμικού στο καθένα, αντιθέτως όλα προγραμματίζονται με το ίδιο και η αντιστοίχιση γίνεται δυναμικά.

Για να προστεθεί ένα node στο υλισμικό, προστίθεται μια καταχώρηση στη μεταβλητή πίνακα **DEVICE_DESCRIPTOROS** όπως φαίνεται στο το παράδειγμα, συμπληρώνοντας τα παρακάτω στοιχεία με την ίδια σειρά:

- Διεύθυνση MAC - Η μοναδική διεύθυνση MAC του συγκεκριμένου node
- APN - Το APN για συνδεσιμότητα στο GPRS όπως το ορίζει ο πάροχος της κάρτας SIM
- Device token - Το μοναδικό κλειδί που δίνει το thingsboard για τη συσκευή που δημιουργήθηκε και αντιστοιχείται στο συγκεκριμένο node.

```
11  const DeviceDescriptor DEVICE_DESCRIPTOROS[] =
12  {
13      {"11:22:33:44:55:66", "internet", "Yihbnbiu&9634Faf2334"}, // Node 1
14      {"77:88:99:11:22:33", "internet", "KDAv409Urduwh87Hufas"}, // Node 2
15      {"44:55:66:77:88:99", "internet", "JUi8H7899yfoiOA8afa3"} // Node 3
16  };
17
```

Σχήμα: Παράδειγμα ρυθμίσεων για 3 nodes

Αν κάποιο node προγραμματιστεί με το υλισμικό ενώ προηγουμένως η διεύθυνση MAC του δεν έχει καταχωρηθεί στο αρχείο αυτό, το node θα τερματίζει τη λειτουργία του κατα την εκκίνηση με ανάλογο σφάλμα στην κονσόλα.

Τέλος στις μεταβλητές **TB_SERVER** και **TB_PORT** συμπληρώνεται η διεύθυνση και η θύρα από την οποία είναι προσβάσιμη η πλατφόρμα Thingsboard.

Εύρεση διεύθυνσης MAC ενός node

Για να μπορεί να συμπληρωθεί το παραπάνω αρχείο με τις διευθύνσεις MAC των nodes, θα πρέπει ο χρήστης να αντλήσει αυτή τη πληροφορία από κάθε node ξεχωριστά, αφού η διεύθυνση είναι μοναδική για το καθένα. Αυτό γίνεται προγραμματίζοντας το node αρχικά χωρίς να προστεθεί στο αρχείο αυτό, έτσι κατα την εκκίνηση θα τερματίζεται η λειτουργία με σφάλμα στην κονσόλα: "Device not recognized!". Ταυτόχρονα θα τυπώνεται και η διεύθυνση MAC της συσκευής η οποία έπειτα μπορεί να συμπληρωθεί στο αρχείο αυτό.

Εναλλακτικά όταν υπάρχουν πολλά nodes, μπορεί να βρεθεί η διεύθυνση MAC με τη βοήθεια του εργαλείου **esp-tool** το οποίο το οποίο συνοδεύει το PlatformIO, τρέχοντας το στην κονσόλα με τον εξής τρόπο:

```
python esptool.py read_mac
```

Η οποία τυπώνει στην κονσόλα πλήθος πληροφοριών μεταξύ αυτών και της διεύθυνσης MAC, όπως φαίνεται στο σχήμα:

```
esptool.py v2.6
Found 2 serial ports
Serial port COM19
Connecting....
Detecting chip type... ESP32
Chip is ESP32-PICO-D4 (revision 1)
Features: WiFi, BT, Dual Core, Embedded Flash, Coding Scheme None
MAC: d8:a0:1d:40:58:b0
Uploading stub...
Running stub...
Stub running...
MAC: d8:a0:1d:40:58:b0
Hard resetting via RTS pin...
```

Σχήμα: Έξοδος εργαλείου esp-tool

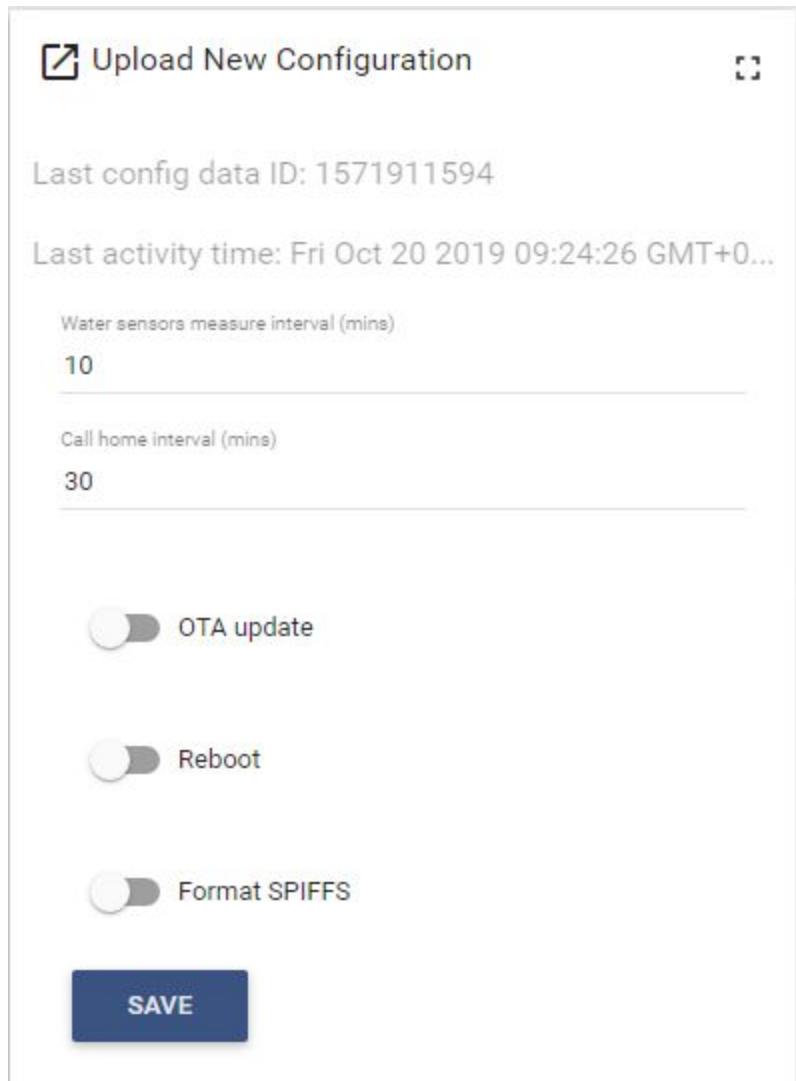
Απομακρυσμένος έλεγχος / ενημέρωση λογισμικού (FOTA)

Το node ανά τακτά χρονικά διαστήματα επανέρχεται από τη λειτουργία ύπνου στην οποία βρίσκεται για να εκτελέσει διάφορες διαδικασίες. Μια από αυτές τις διαδικασίες είναι η επικοινωνία με την πλατφόρμα κατα την οποία εφόσον το επιθυμεί ο χρήστης, μπορεί να λάβει νέες ρυθμίσεις/εντολές και να τις εφαρμόσει, αναφέροντας το αποτέλεσμα πίσω στην πλατφόρμα.

Απομακρυσμένος έλεγχος

Ο έλεγχος των παραμέτρων/εντολών που μπορούν να ρυθμιστούν εξ αποστάσεως κατα την επικοινωνία γίνεται μέσω ειδικού Thingsboard widget που φτιάχτηκε ειδικά για αυτόν τον σκοπό. Αφού προστεθεί το Device Management widget και συνδεθεί στη συσκευή, ο χρήστης μπορεί να ορίζει τις παρακάτω εντολές/παραμέτρους:

- Sensors measure interval - Ρυθμός σε λεπτά με τον οποίον θα ξυπνάει το node για να πάρει μετρήσεις από του αισθητήρες
- Call home interval - Ρυθμός σε λεπτά με τον οποίον θα ξυπνάει το node για να επικοινωνήσει με την πλατφόρμα (αποστολή δεδομένων, λήψη νέων ρυθμίσεων/εντολών και εφαρμογή)
- OTA Update - Εντολή για λήψη νέου λογισμικού, εφαρμογή του και εκκίνηση σε αυτό
- Reboot - Εντολή για επανεκκίνηση του node
- Format SPIFFS - Εντολή για διαμόρφωση (format) της εσωτερικής μνήμης flash



Σχήμα: Device management widget

Πατώντας “Save” οι ρυθμίσεις αποθηκεύονται και θα ληφθούν από το node στη επόμενη επικοινωνία του με την πλατφόρμα. Αφού οι ρυθμίσεις ληφθούν και εφαρμοστούν από τη συσκευή, τα αποτελέσματά τους μπορούν να προβληθούν από την καταγραφή (log).

Απομακρυσμένη ενημέρωση λογισμικού (FOTA)

Για να πραγματοποιηθεί απομακρυσμένη ενημέρωση λογισμικού πρέπει το δυαδικό παράγωγο της μεταγλώττισης (firmware binary) να είναι προσβάσιμο μέσω δημόσιου URL. Ενεργοποιώντας την επιλογή “OTA Update” στο Device Management widget, εμφανίζονται επιπλέον πεδία που απαιτούνται για τη διαδικασία:

- Target firmware version - Η έκδοση του νέου λογισμικού προς εφαρμογή. Σε περίπτωση που η τρέχουσα έκδοση είναι ίδια, η διαδικασία αναβάλλεται.

- Firmware binary URL - Η διεύθυνση από την οποία το αρχείο λογισμικού είναι προσβάσιμο
- Firmware MD5 - Το MD5 checksum του αρχείου λογισμικού. Αφού ληφθεί το αρχείο από το node, υπολογίζεται το checksum και αν δεν συμπίπτει με την τιμή αυτού του πεδίου, θεωρείται ότι το αρχείο πιθανόν είναι κατεστραμμένο και για λόγους ασφαλείας η διαδικασία αναβάλλεται.

Upload New Configuration ::

Last config data ID: 1571911594

Last activity time: Fri Oct 20 2019 9:53:44 GMT+0300 (Easter...)

Target firmware version 101	32 / 32
Firmware binary URL http://myfileserver.com/files/firmware_v101.bin	
Firmware MD5 7435792a30bfaca4b561a882b85cf19	
Note: Call home interval (mins) 10	
Call home interval (mins) 30	
<input checked="" type="checkbox"/> OTA update <input type="checkbox"/> Reboot <input type="checkbox"/> Format SPIFFS	

SAVE

Σχήμα: Πεδία FOTA στο widget

Στην επόμενη επικοινωνία του node με την πλατφόρμα, θα ληφθεί η εντολή ενημέρωσης λογισμικού και η διαδικασία θα ξεκινήσει. Για την επιτυχία ή αποτυχία της διαδικασίας ο χρήστης θα ενημερωθεί αμέσως μετά το πέρας της διαδικασίας μέσω των logs που θα αποστείλει το node.

Testing / debugging

Σε περίπτωση προβλημάτων αλλά και κατά τη διαδικασία της ανάπτυξης, είναι χρήσιμο να υπάρχουν επιπλέον λειτουργίες ή διαγνωστικά εργαλεία τα οποία θα βοηθήσουν στην γρήγορη αποσφαλμάτωση του υλισμικού.

Έκδοση debug / release

Στο PlatformIO project έχουν οριστεί σετ ρυθμίσεων μεταγλώττισης οι οποίες ενεργοποιούνται ανάλογα τη εντολή build ή upload που θα χρησιμοποιηθεί από το μενού PlatformIO της βασικής μπάρας. Κάνοντας μεταγλώττιση του υλισμικού στην έκδοση "debug" ενεργοποιείται η έξοδος μηνυμάτων αποσφαλμάτωσης στην σειριακή κονσόλα, τα οποία μπορεί να προβάλλει ο χρήστης μέσω σύνδεσης USB και κατάλληλης εφαρμογής (serial monitor). Τα μηνύματα αυτά περιλαμβάνουν την τρέχουσα κατάσταση της συσκευής αλλά και τυχόν σφάλματα που προέκυψαν, σε πραγματικό χρόνο.

Ταυτόχρονα, κάνοντας μεταγλώττιση του κώδικα σε έκδοση debug, χρησιμοποιούνται τα αντίστοιχα στοιχεία αναγνώρισης και σύνδεσης του node στο αρχείο credentials.h. Με αυτόν τον τρόπο όταν η συσκευή προορίζεται για δοκιμαστική χρήση, γίνεται χρήση διαφορετικών στοιχείων σύνδεσης και έτσι μπορεί να χρησιμοποιηθεί μια δοκιμαστική εγκατάσταση της πλατφόρμας.

Logs

Σε όλη τη ροή της λειτουργίας του, το node καταγράφει μια σειρά από συμβάντα και σφάλματα τα οποία αποθηκεύονται στην εσωτερική μνήμη και αποστέλλονται στην πλατφόρμα κατά την διαδικασία αποστολής δεδομένων, προκειμένου να κάνουν γνωστή την τρέχουσα κατάσταση της συσκευής, να αναφέρουν τυχόν σφάλματα ή το αποτέλεσμα μιας διαδικασίας που ζήτησε ο χρήστης εξ αποστάσεως (π.χ. αλλαγή ρυθμού μέτρησης). Η μορφή των καταγραφών είναι πολύ απλή λόγω των περιορισμών μνήμης και επικοινωνιών (GPRS), και περιλαμβάνει έναν κωδικό που αντιστοιχεί στο συμβάν, τον ακριβή χρόνο του συμβάντος καθώς και κάποια επιπλέον στοιχεία που βοηθούν στην περιγραφή του (metadata). Οι καταγραφές μπορούν να προβληθούν από το αντίστοιχο widget στην πλατφόρμα, όπου οι καταγραφές εμφανίζονται με την μορφή κειμένου.

Logs		
Realtime - last 30 days		
Timestamp	Log	Raw
2019-09-25 01:32:58	Sensor data submitted (Total entries: 3 - CRC failures: 0)	5,0
2019-09-25 01:32:46	GSM: RSSI (-54)	101,-54,0
2019-09-25 01:32:46	GSM battery gauge (Voltage: 4059 mV - Pct: 82%)	18,4059,82
2019-09-25 01:32:26	Waking up	13,3,0
2019-09-25 01:32:22	File system space (Used: 1506 B - Free: 1372970 B)	47,1506,1372970
2019-09-25 01:32:22	Internal env. sensor (Pressure: 1011hPa - Alt: 11m)	17,1011,11
2019-09-25 01:32:22	Internal env. sensor (Temperature: 15C - Rel. Humidity: 57%)	16,15,57
2019-09-25 01:32:21	Battery gauge (Voltage: 4035 mV - Pct: 100%)	15,4035,100
2019-09-25 01:32:21	Calling home	3,0,0
2019-09-25 01:22:43	Waking up	13,1,0
2019-09-25 01:22:37	Going to sleep	12,11,0
2019-09-25 01:12:58	Waking up	13,1,0

Σχήμα: Widget καταγραφών (logs)

Επιλεγμένα από τα συμβάντα που λαμβάνονται από το node καταγράφονται και ως χρονοσειρές οι οποίες μπορούν να προστεθούν σε ανάλογα widgets και να οπτικοποιηθούν σε γραφήματα. Τα ονόματα αυτών των χρονοσειρών ξεκινάνε με "d_" (diagnostics) και είναι οι εξής:

- d_boot - Πραγματοποιήθηκε εκκίνηση της συσκευής
- d_sd_total_rec - Πραγματοποιήθηκε αποστολή εγγραφών των αισθητήρων
- d_sd_crc - Υπήρξε σφάλμα κατά τον έλεγχο ακεραιότητας (crc check) (τιμή: αριθμός εγγραφών που απέτυχαν τον έλεγχο)
- d_sd_total_req - Πραγματοποιήθηκε αποστολή εγγραφών των αισθητήρων (τιμή: αριθμός http request που εκτελέστηκαν)
- d_sd_failed_req -

Χρονοσειρά	Περιγραφή	Τιμή
d_boot	Πραγματοποιήθηκε εκκίνηση της συσκευής	Άνευ σημασίας
d_sd_total_rec	Ολοκληρώθηκε η αποστολή εγγραφών των αισθητήρων	Συνολικός αριθμός εγγραφών προς αποστολή
d_sd_crc	Παρουσιάστηκαν σφάλματα κατά τον έλεγχο	Αριθμός εγγραφών που απέτυχαν τον έλεγχο (και συνεπώς δεν εστάλησαν)

	ακεραιότητας (CRC check) εγγραφών πριν την αποστολή	
d_sd_total_rec	Ολοκληρώθηκε η αποστολή εγγραφών των αισθητήρων	Αριθμός των HTTP requests που εκτελέστηκε για την αποστολή.
d_sd_failed_req	Ολοκληρώθηκε η αποστολή εγγραφών των αισθητήρων	Αριθμός των HTTP requests που απέτυχαν
d_tm_awake_s	Η συσκευή μπήκε σε λειτουργία ύπνου.	Χρόνος που βρισκόταν το node σε λειτουργία από τότε που επανήλθε από τη λειτουργία ύπνου (δευτερόλεπτα)
d_wakeup	Η συσκευή επανήλθε από τη λειτουργία ύπνου.	Άνευ σημασίας
d_ntp_fail	Ο συγχρονισμός της ώρας μέσω NTP απέτυχε.	Άνευ σημασίας
d_bat_mv	Κατάσταση μπαταρίας	Τάση σε mV.
d_bat_p	Κατάσταση μπαταρίας	Ποσοστό
d_temp	Εσωτερική θερμοκρασία συστήματος	Θερμοκρασία σε βαθμούς κελσίου
d_hum	Εσωτερική υγρασία συστήματος	Σχετική υγρασία ποσοστιαία (%)
d_gsm_conn_fail	Αποτυχία σύνδεσης στο δίκτυο GSM	Άνευ σημασίας
d_gsm_rssi	Ισχύς σήματος	Ισχύς σε dBm
d_gprs_conn_fail	Αποτυχία σύνδεσης GPRS	Άνευ σημασίας

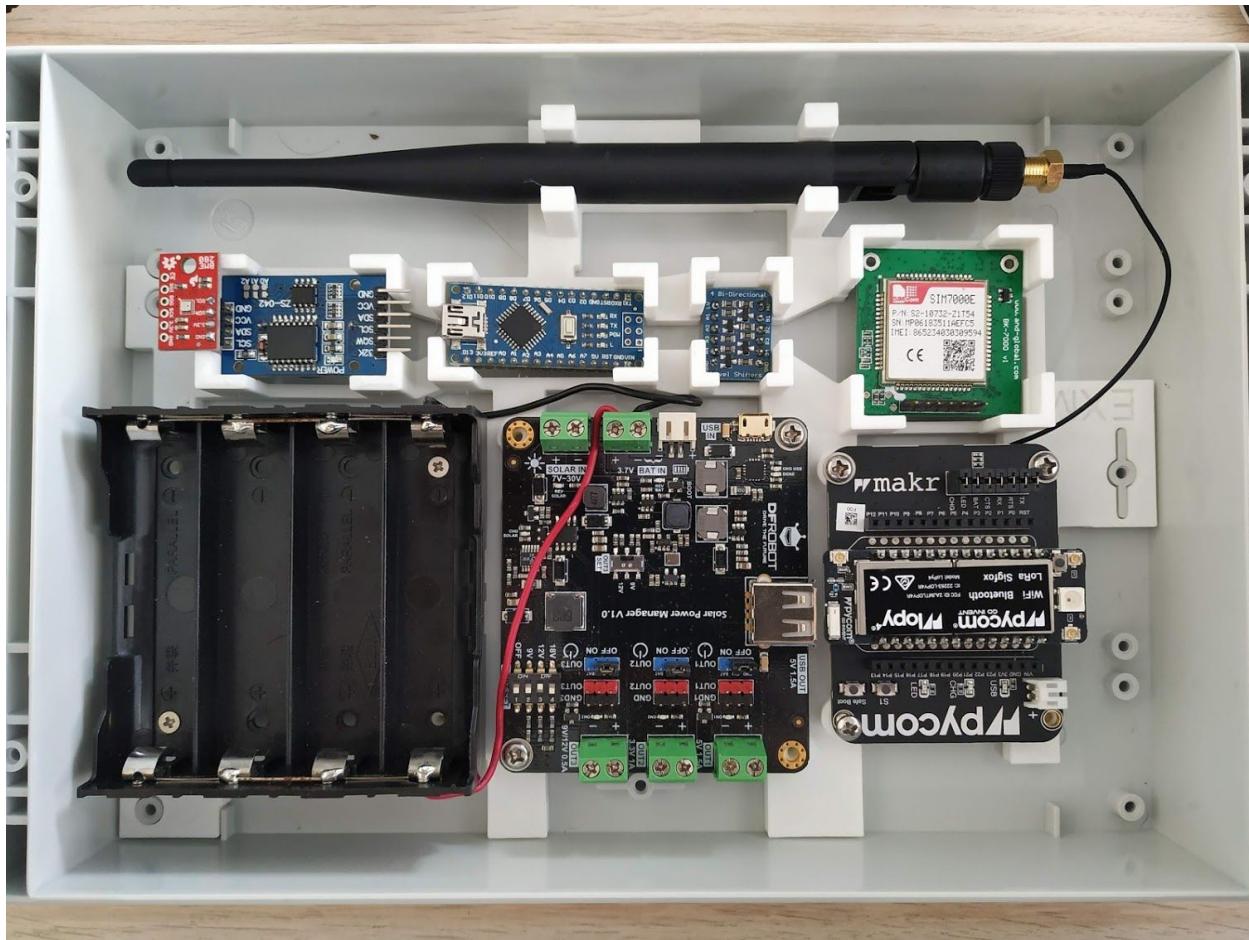
Προβλήματα, επόμενα βήματα

Σφαλματα στις επικοινωνιες GSM

Στην πορεια των δοκιμων του πρωτοτυπου κυκλωματος ELIoT Node v1 προεκυψε οτι υπηρχαν πολλα σφαλματα / θόρυβος στις UART επικοινωνιες μεταξυ ESP32 + SIM7000. Η χρηση ομοαξονικου καλωδιου δεν αποτελεσε ικανοποιητικη λυση, και μετα απο εκτεταμενες δοκιμες καταληξαμε στο συμπερασμα οτι το προβλημα ηταν στο γεγονος οτι ο

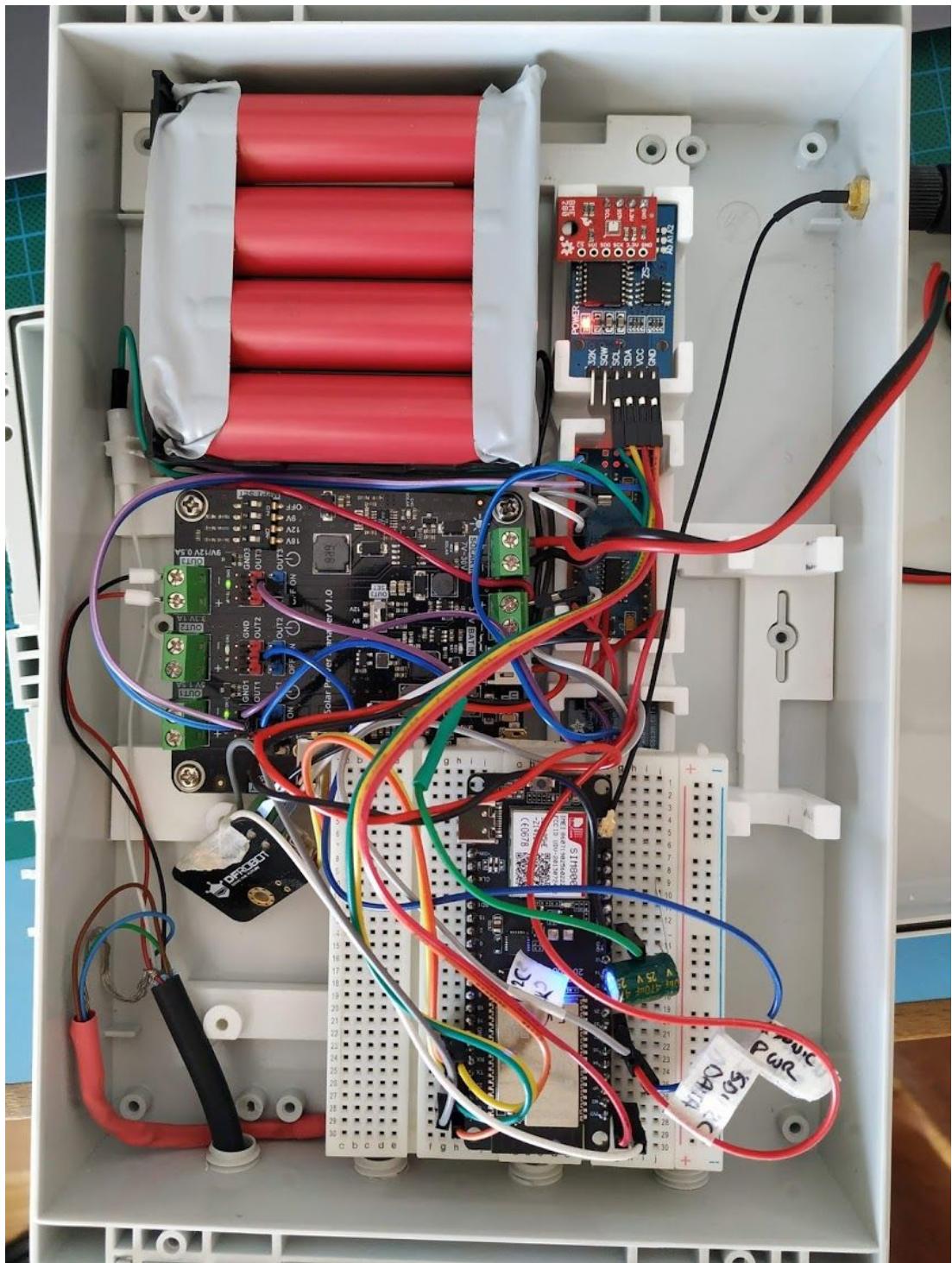
μικροελεγκτής ESP32 και το κύκλωμα που το υποστηρίζει δεν είναι θωρακισμένα στις ηλεκτρομαγνητικές παρεμβολές, αρα δεχεται παρεμβολες από το GSM module κατα την διαρκεια της λειτουργιας των GSM επικοινωνιων.

Αυτο το σημαντικο προβλημα καθιστα ακαταλληλο το TFOX για GSM επικοινωνιες και για αυτο εγιναν διαφορες δοκιμες με διαφορετικες κεντρικες πλακετες ESP32 που ειχαν το chip θωρακισμενο, οπως το PYCOM LOPY4.



ELIoT Node v1.1 pycom lopy4 + SIM 7000

Το PYCOM LOPY4 έχει καλύτερη συμπεριφορά, σταθερότητα στις επικοινωνίες GSM αλλά ανεβαζει το κοστος των υλικων (€30 + €30) οποτε η αναζητηση για εναλλακτικες πλακετες με ESP32 συνεχιστηκε. Τον Ιουλιο του 2019 η TTGO εβγαλε το TCALL μια νεα πλακετα ESP32 με ενσωματωμενες επικοινωνιες GSM (*SIM800) σε ιδιαίτερα χαμηλο κοστος (€10), το οποίο και αποτελει το πρώτο ELIoT Node που λειτουργησε αυτόνομα και ασταμάτητα για ~1 μήνα, οπότε αποφασίστηκε η έκδοση v2 του node να βασιστει σε αυτό. (βλεπε D2.4)



EIIoT Node v1.2 TTGO TCALL SIM800

Παράτημα προδιαγραφών εξαρτημάτων (datasheets)

Στον φάκελο appendix υπάρχουν τα αντίστοιχα datasheets των modules που χρησιμοποιήθηκαν.