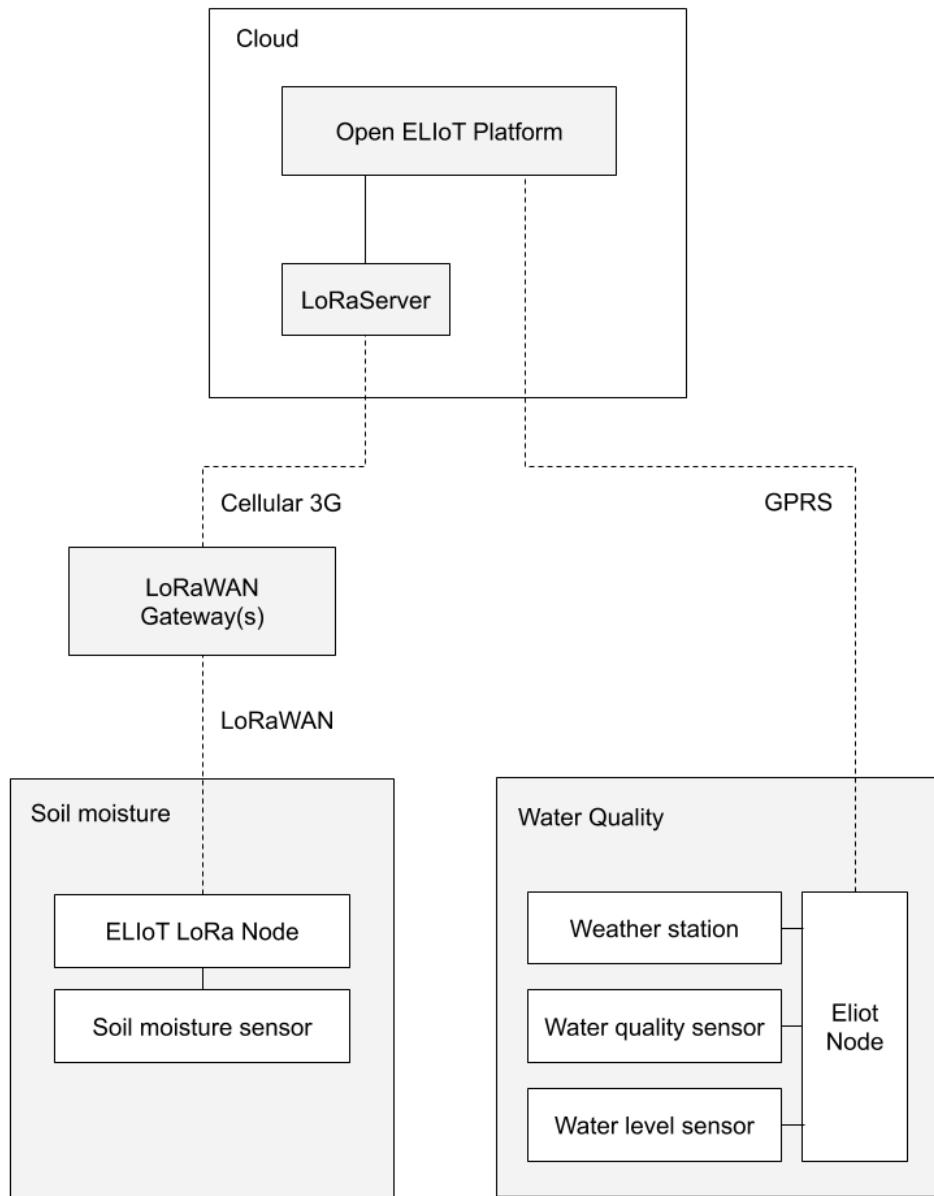


# Τεχνικός οδηγός ανάπτυξης αισθητήρων μέτρησης περιβαλλοντικών παραμέτρων - Τελική έκδοση

<b>Αρχιτεκτονική λυσης - εξοπλισμος</b>	<b>3</b>
ELIoT Node (v3)	5
Κεντρικός μικροελεγκτής και βασική πλακέτα	6
Μητρική πλακέτα (carrier board)	9
Επικοινωνίες	10
GSM	10
SDI-12	11
Αισθητήρες	11
Αισθητήρας μέτρησης ποιότητας υδάτων	11
Αισθητήρας μέτρησης στάθμης	12
Μετεωρολογικοί σταθμοί	14
MeterGroup Atmos41	14
Fineoffset WHxxx / AmbientWeather WS-2xxx	16
Επικοινωνία με πλατφόρμα	17
Διαχείριση ενέργειας	18
Περίβλημα / κουτί	18
ELIoT Soil Sensor	20
LoRaWAN Gateway	21
Ρύθμιση του gateway	22
Ρύθμιση LoRa Server	23
<b>Υλισμικό (Firmware)</b>	<b>30</b>
Εκκίνηση	30
Κύκλος Λειτουργίας	30
Επικοινωνία με την πλατφόρμα	32
Μέτρηση παραμέτρων αισθητήρων	32
Μνήμη και διαχείριση αρχείων	32
Καταγραφή συμβάντων (Logging)	33
Ώρα	33
FOTA (Firmware Over The Air)	33

<b>Αρχικοποίηση συσκευής</b>	<b>34</b>
Αισθητήρας ποιότητας εδάφους AquaTROLL 400, 500, 600	34
Προγραμματισμός firmware	35
Σύνδεση με πλατφόρμα	39
Πρόσθεση της συσκευής στην πλατφόρμα	39
Ρυθμίσεις σύνδεσης στο υλισμικό	41
Εύρεση διεύθυνσης MAC ενός node	42
<b>Απομακρυσμένος έλεγχος / ενημέρωση λογισμικού (FOTA)</b>	<b>43</b>
Απομακρυσμένος έλεγχος	43
Απομακρυσμένη ενημέρωση λογισμικού (FOTA)	44
Testing / debugging	46
Έκδοση debug / release	46
Logs	46
<b>Παράρτημα προδιαγραφών εξαρτημάτων (datasheets)</b>	<b>48</b>

# 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:

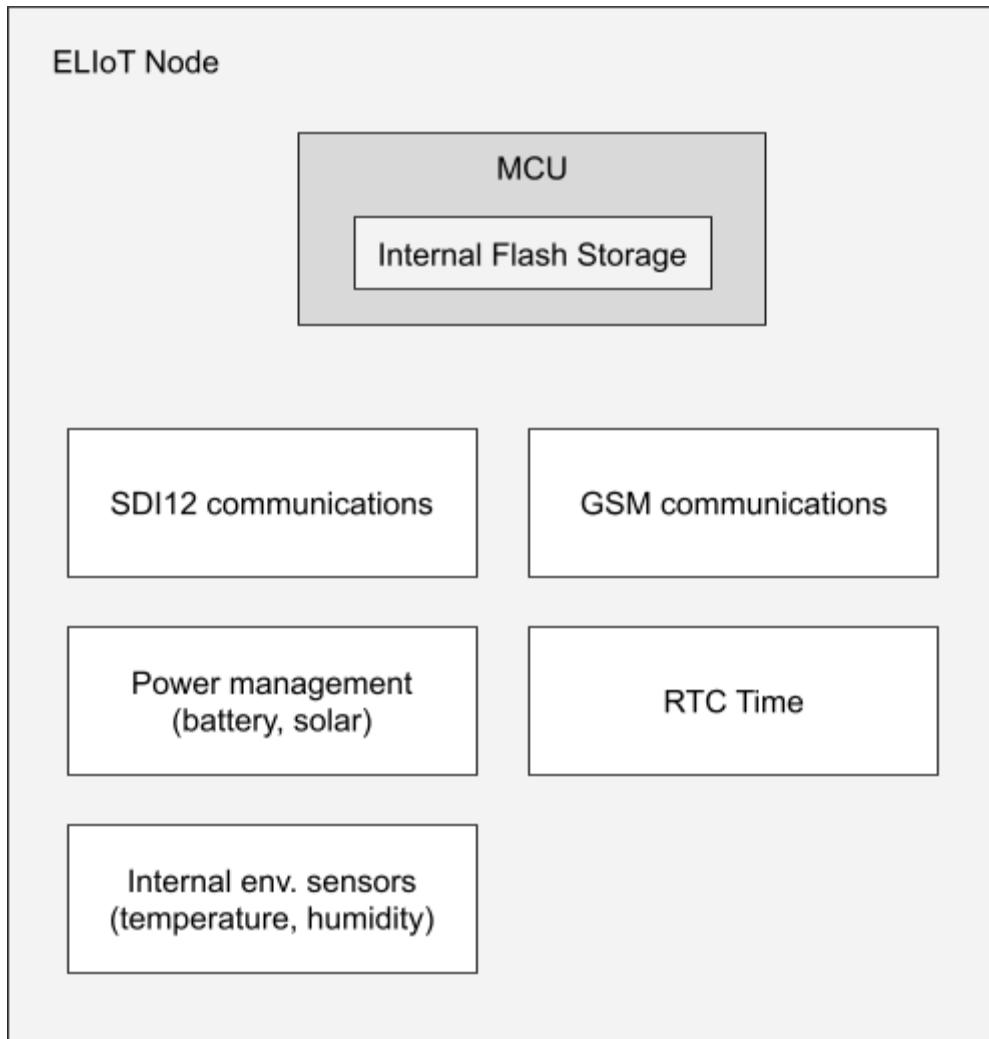
- Σταθμός μέτρησης ποιότητας υδάτων.
- Σταθμός μέτρησης υγρασίας εδάφους

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 / NBIoT 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 συντήρησης χαμηλό, επικοινωνούν με την πλατφόρμα με την τεχνολογία LoRaWAN χωρίς να χρειαζεται SIM κάρτα (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 (v3)





Σχήμα: Αρχιτεκτονική ELIoT Node

Το ELIoT 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.

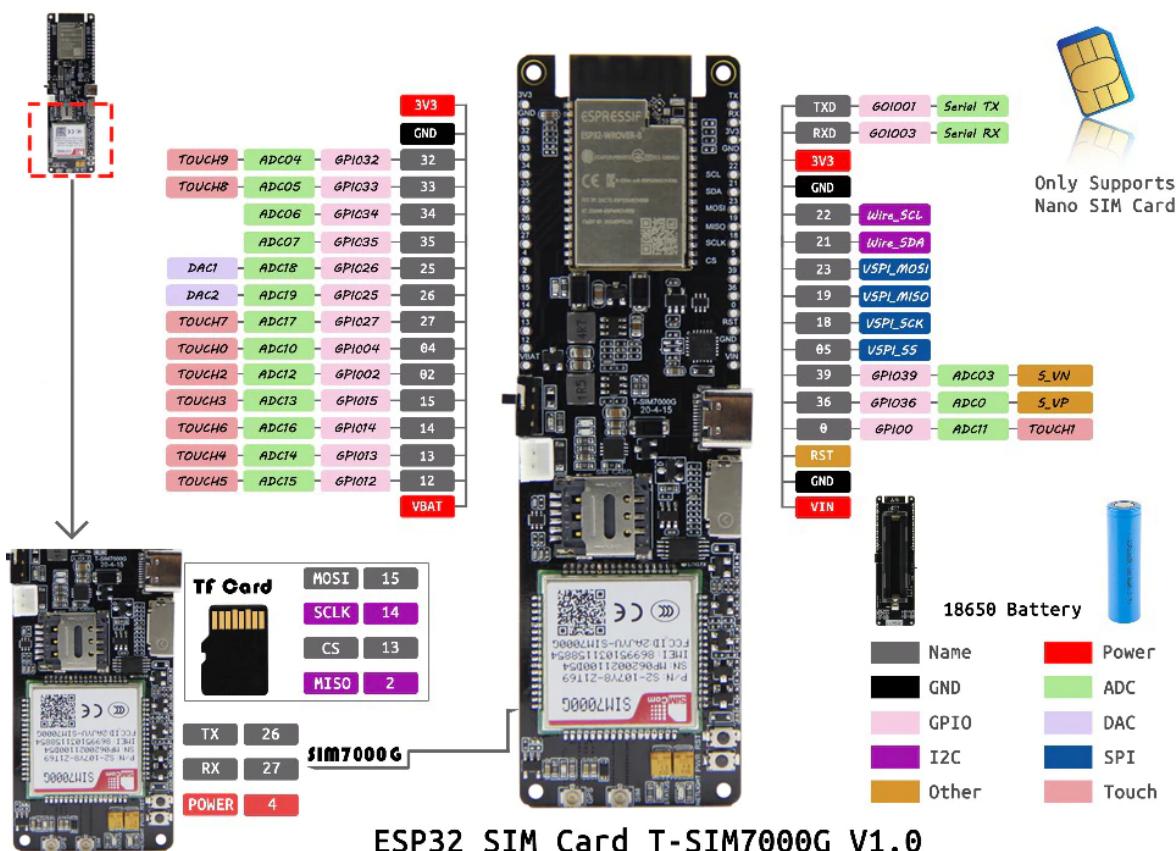
### Κεντρικός μικροελεγκτής και βασική πλακέτα

Βασικό κριτήριο στην επιλογή του μικροελεγκτή ήταν η υποστήριξη από την πλατφόρμα ανάπτυξης Arduino, πράγμα που το καθιστά προσιτό και εύκολα παραμετροποιήσιμο από την κοινότητα ανοιχτού υλικού και λογισμικού. Επιπλέον υπάρχει μεγάλος αριθμός βιβλιοθηκών για κάθε χρήση, έτσι οποιοσδήποτε μπορεί να παραμετροποιήσει το υλισμικό στις δικές του ανάγκες και να προσθέσει αισθητήρες ειδικά για τη χρήση στην οποία το προορίζει.

Τέλος, εφόσον το σύστημα προορίζεται για αυτόνομη λειτουργία με παροχή ενέργειας από μπαταρίες, η κατανάλωση ενέργειας θα πρέπει να διατηρείται στο ελάχιστο και θα πρέπει να υπάρχει δυνατότητα φόρτισης με ηλιακή ενέργεια.

Η πλατφόρμα ESP32 της Espressif τηρεί όλα τα παραπάνω κριτήρια και σε συνδυασμό με τις βιβλιοθήκες του Arduino αποτελεί ίσως το πιο δημοφιλές σύγχρονο οικοσύστημα ανάπτυξης Internet of Things εφαρμογών. Υπάρχει μεγάλο εύρος πλακετών ανάπτυξης στην αγορά, σε πολύ χαμηλό κόστος και σε πλήθος παραλλαγών σε ενσωματωμένα περιφερειακά όπως κυκλώματα επικοινωνίας LoRa, GSM, κυκλώματα φόρτισης, οθόνες κ.α.

Για την ανάπτυξη του ELIoT Node v3 χρησιμοποιείται η πλακέτα ανάπτυξης TSIM7000 της LilyGO η οποία είναι ανοιχτού κώδικα και ανοιχτού υλικού (open hardware) με προσιτό κόστος. Περιλαμβάνει κύκλωμα φόρτισης μπαταρίας με ηλιακή ενέργεια και USB,, GSM/GPRS/NBIoT module και χαμηλή κατανάλωση ενέργειας.



Σχήμα: Πλακέτα ανάπτυξης T-SIM7000G και τα περιφερειακά της

<b>Hardware Specifications</b>
Main chip: ESP32
Module: ESP32-WROVER-B,SIM7000G
Interface:USB,SD Card,Solar Input interface
Flash Memory: 4M Flash / 8M PSRAM
Button: Reset,Power switch
USB to TTL: CP2104
Car Clock: 40MHz
SIM Card Size: Nano SIM Card
SIM Antenna
GPS Antenna
Operating Temperature Range: -40-85 degrees
Product Size: 111mm * 37mm * 20mm
Product Weight: 50g
<b>Power Specifications</b>
Working Voltage: 5v input Solar input voltage range :4.4-6v Working Current: About 200mA Charging Current: 780mA Battery Model: 18650 Battery Holder Model: 18650 battery holder JST Connector: 2pin 2mm USB: TYPE-C 3.0
<b>Wi-Fi</b>
Standard :FCC/CE-RED/IC/TELEC/KCC/SRRC/NCC

Protocol : 802.11 b/g/n(802.11n, speed up to 150Mbps) 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
<b>Bluetooth</b>
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 Specification
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

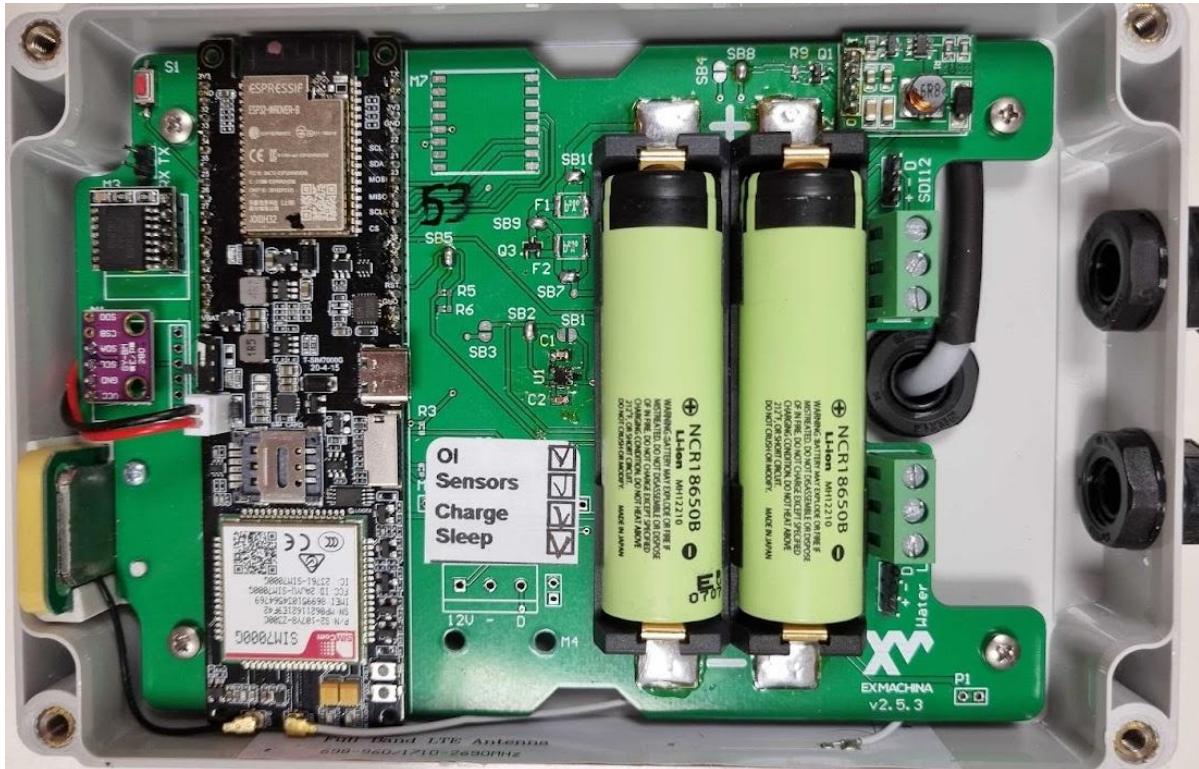
Πίνακας χαρακτηριστικών LilyGo TSIM7000

## Μητρική πλακέτα (carrier board)

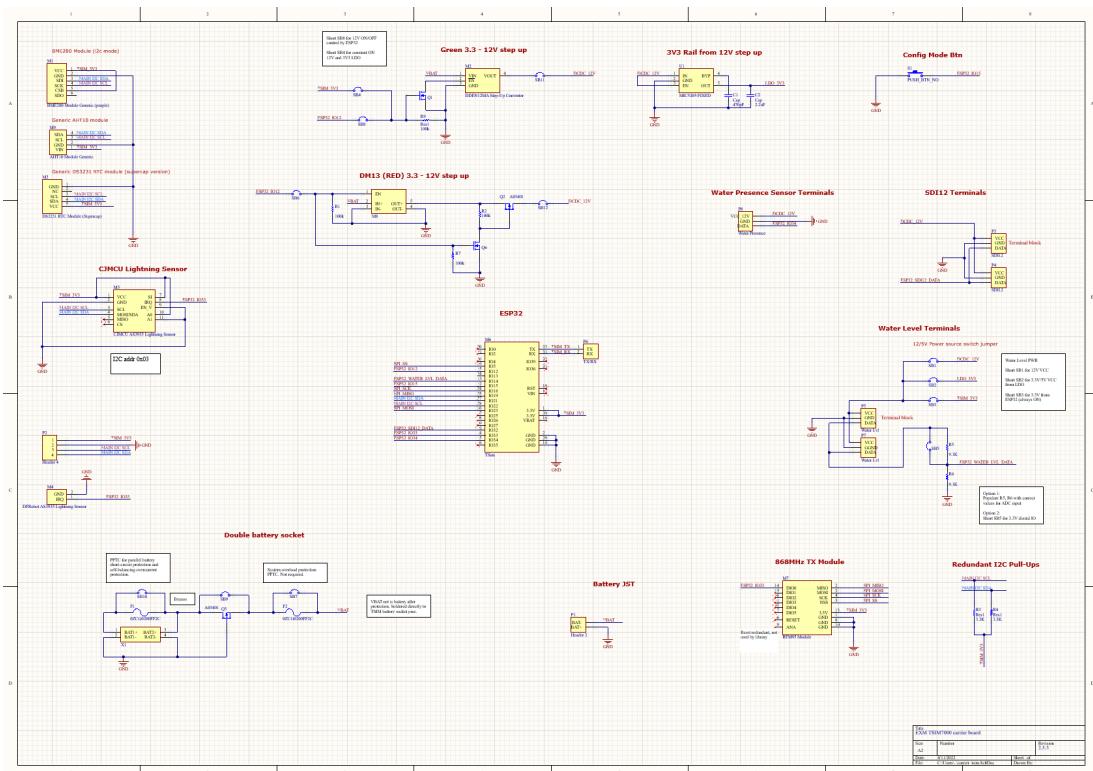
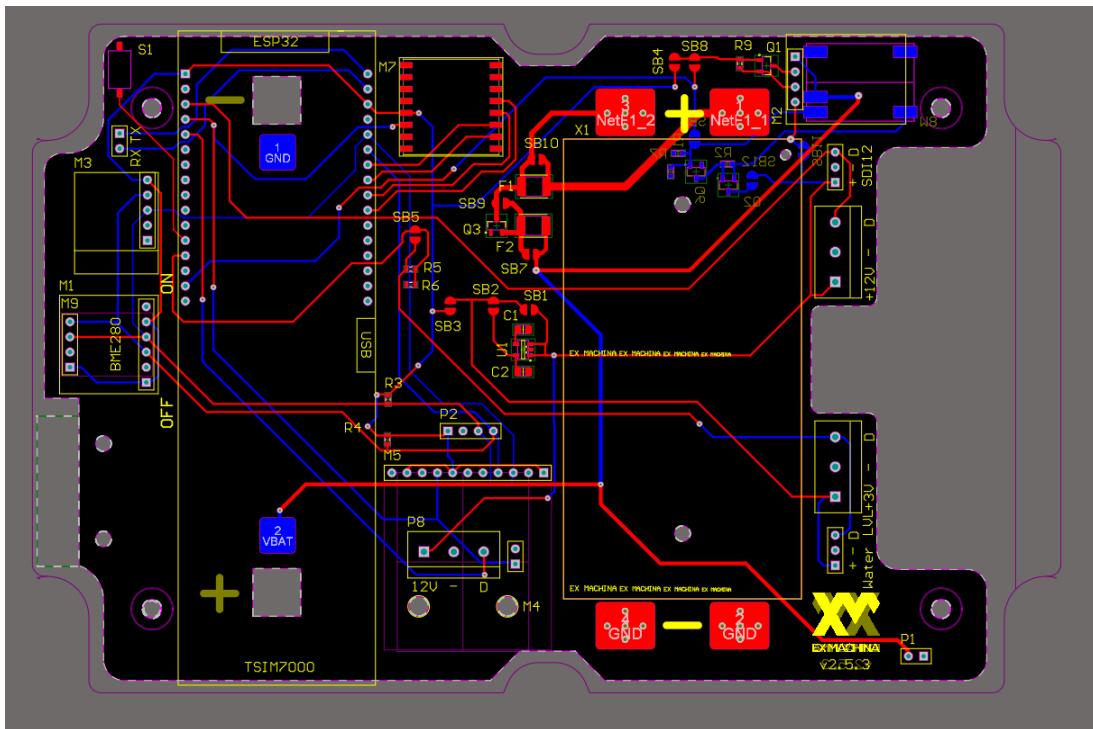
Για την υποστήριξη της βασικής πλακέτας ανάπτυξης TSIM7000, καθώς και την προσθήκη άλλων υποπλακετών (modules) για επιπλέον λειτουργικότητα, σχεδιάστηκε πλακέτα τυπωμένου κυκλώματος (PCB) η οποία έχει τον ρόλο της “πλακέτας φορέα” (carrier board). Η πλακέτα αυτή έχει κατάλληλο σχήμα για μέγιστη εκμετάλλευση του χώρου που υπάρχει στο κουτί της

κατασκευής καθώς και οπές στήριξης ευθυγραμμισμένες με τις βάσεις στήριξης στο κουτί με βίδες. Επιπλέον στην πλακέτα τοποθετούνται τα παρακάτω εξαρτήματα/υποπλακέτες:

- Αισθητήρας θερμοκρασίας, υγρασίας και πίεσης. Για παρακολούθηση των εσωτερικών συνθηκών θερμοκρασίας/υγρασίας του κουτιού και της ατμοσφαιρικής πίεσης.
- RTC module (κύκλωμα ρολογιού) με μπαταρία για διατήρηση της ώρας
- Πομποδέκτης LoRa
- Θήκη για 2 μπαταρίες τύπου 18650
- Υποδοχές τύπου κλέμας για συνδεση των εξωτερικών αισθητήρων
- Κυκλώματα σταθεροποίησης και τροφοδοσίας των εξωτερικών αισθητήρων



Σχήμα: Πλακέτα ELIoT Node v3



Σχήμα: Πλακέτα και σχηματικό ELIoT Node

## Επικοινωνίες

### GSM

Για την μεταφορά των δεδομένων στο cloud μέσω internet, χρησιμοποιείται το πλέον διαδεδομένο module της SimCom, SIM7000. Υποστηρίζει επικοινωνίες μέσω GSM/GPRS/NBIoT καθώς και δέκτη GPS. Η εγγραφή της πλακέτας στο δίκτυο κινητής τηλεφωνίας επιτυγχάνεται μέσω της κάρτας SIM η οποία τοποθετείται στην υποδοχή που βρίσκεται στην πλακέτα ανάπτυξης TSIM7000.



Σχήμα: SimCom SIM7000 module

### SDI-12

Το SDI-12 είναι ένα ασύγχρονο σειριακό πρωτόκολλο και χρησιμοποιείται από τους περισσότερους βιομηχανικούς 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.

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.

Για την υποστήριξη του πρωτοκόλλου SDI-12 από το ELIoT Node v3 αναπτύχθηκε ειδική βιβλιοθήκη.

## Αισθητήρες

Αισθητήρας μέτρησης ποιότητας υδάτων

Για την μέτρηση της ποιότητας υδάτων υποστηρίζονται οι αισθητήρες της σειράς AquaTROLL In-Situ, και συγκεκριμένα τα μοντέλα 400, 500, 600,

Περιλαμβάνουν την ικανότητα μέτρησης εύρους παραμέτρων από πολλαπλούς υπο-αισθητήρες που τοποθετούνται σε αυτούς. Είναι αρκετά χαμηλής κατανάλωσης (~16 - 300mA κατά τη διάρκεια της μέτρησης - ανάλογα τις παραμέτρους που απαιτούνται) ώστε να μπορούν να συνδέονται 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



Σχήμα: Πολυαισθητήρας AquaTROLL 500

#### 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.



Σχήμα: Αισθητήρας στάθμης υπερήχων MaxBotix MB7388



Σχήμα: Αισθητήρας στάθμης σε αδιαβροχό κουτί

### MB7388 Specifications

- Resolution of 1-mm
- 6Hz read rate
- Internal temperature compensation
- 42kHz ultrasonic sensor measures distance to objects
- RoHS Compliant
- Read from all 3 sensor outputs: Analog Voltage, Serial, Pulse Width
- Virtually no sensor dead zone, objects closer than 50cm typically range as 50cm
- Operates from 2.7-5.5V
- Low 2.9mA average current requirement
- Small, light weight module
- Designed for easy integration into your project or product
- Operational temperature -40°C to +65°C (-40°F to +149°F)
- Real-time automatic calibration (voltage, humidity, and ambient noise)
- Firmware filtering for excellent noise tolerance and clutter rejection
- Weather resistant (IP67), optional chemical resistant F-Option
- Matches standard electrical 3/4-inch PVC pipe fittings for easy mounting

(3/4-inch National Pipe Thread Straight)

- Long, narrow detection zone
- TTL Serial Output
- Our longest range, high-resolution outdoor, weather resistant, ultrasonic sensor
- Maximum range of 10 meters (394 inches)

## Μετεωρολογικοί σταθμοί

Για την συλλογή περιβαλλοντικών δεδομένων καιρού υποστηρίζεται η επικοινωνία και συλλογή δεδομένων από μετεωρολογικούς σταθμούς. Για λόγους κόστους υποστηρίζονται 2 τύποι σταθμών:

- MeterGroup Atmos41 - Υψηλού κόστους
- Fineoffset WHxxx / AmbientWeather WS-2xxx - Χαμηλού κόστους

### MeterGroup Atmos41

Ενσύρματος μετεωρολογικός σταθμός σχετικά υψηλού κόστους (~1000 ευρώ). Κατάλληλος για μετρήσεις υψηλής ακρίβειας και χρήση χωρίς ιδιαίτερη συντήρηση αφού δεν περιλαμβάνει κινούμενα μέρη (solid state). Η επικοινωνία με τον σταθμό επιτυγχάνεται με το πρωτόκολλο SDI-21 μέσω της ειδικής θύρας που υπάρχει στο EIIoT Node.



Σχήμα: MeterGroup Atmos41

## Measurements specifications:

**Solar radiation**  
Range: 0 to 1750 W/m<sup>2</sup>  
Resolution: 1 W/m<sup>2</sup>  
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

Πίνακας: Χαρακτηριστικά Atmos41

#### Fineoffset WHxxx / AmbientWeather WS-2xxx

Σειρά ασύρματων μετεωρολογικών σταθμών χαμηλού σχετικά κόστους (100 - 200 ευρώ), και αρκετά καλής ακρίβειας για τις περισσότερες χρήσεις. Η ανάγνωση των μετρήσεων του σταθμού επιτυγχάνεται με την ασύρματη λήψη των πακέτων που αποστέλλουν τακτικά και την αποκωδικοποίηση τους.



Σχήμα: Μετεωρολογικός σταθμός FineOffset / AmbientWeather

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

Πίνακας: Χαρακτηριστικά σταθμών Fineoffset WHxxx / AmbientWeather WS-2xxx

#### Επικοινωνία με πλατφόρμα

Η επικοινωνία των ELIoT nodes με την πλατφόρμα γίνεται με χρήση του πρωτοκόλλου HTTP.

#### Διαχείριση ενέργειας

Η βασική πηγή ενέργειας των ELIoT nodes αποτελείται από 2 μπαταρίες λιθίου τύπου 18650. Οι εν λόγω μπαταρίες προσφέρουν μια καλή σχέση κόστους (~5€), διαθεσιμότητας και χωρητικότητας (~2500 - 3000 mAh).

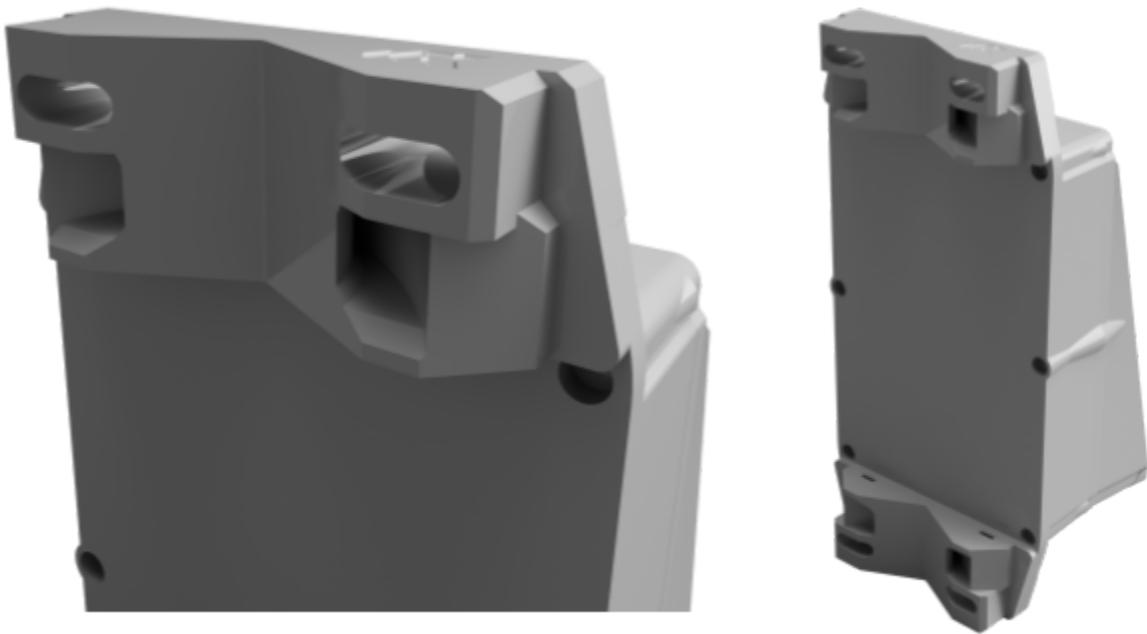
Οι μπαταρίες φορτίζουν από το ενσωματωμένο ηλιακό πάνελ ισχύος 2.5W.

#### Περίβλημα / κουτί

Λόγω του πεδίου χρήσης του ELIoT Node στο οποίο είναι συνηθισμένη έκθεση σε έντονα καιρικά φαινόμενα (βροχόπτωση, υψηλές θερμοκρασίες κλπ), απαιτείται κουτί με ειδικές προδιαγραφές ώστε να εξασφαλιστεί η απρόσκοπη λειτουργία και να μεγιστοποιηθεί ο χρόνος ζωής της κατασκευής. Για τον λόγο αυτό ως κουτί του ELIoT Node χρησιμοποιείται το **Hammond 1555hf17**, το υλικό του οποίου (πολυκαρβονικό) έχει αυξημένη αντοχή στον ήλιο ενώ ταυτόχρονα η κατασκευή του προσφέρει αδιαβροχοποίηση IP67.

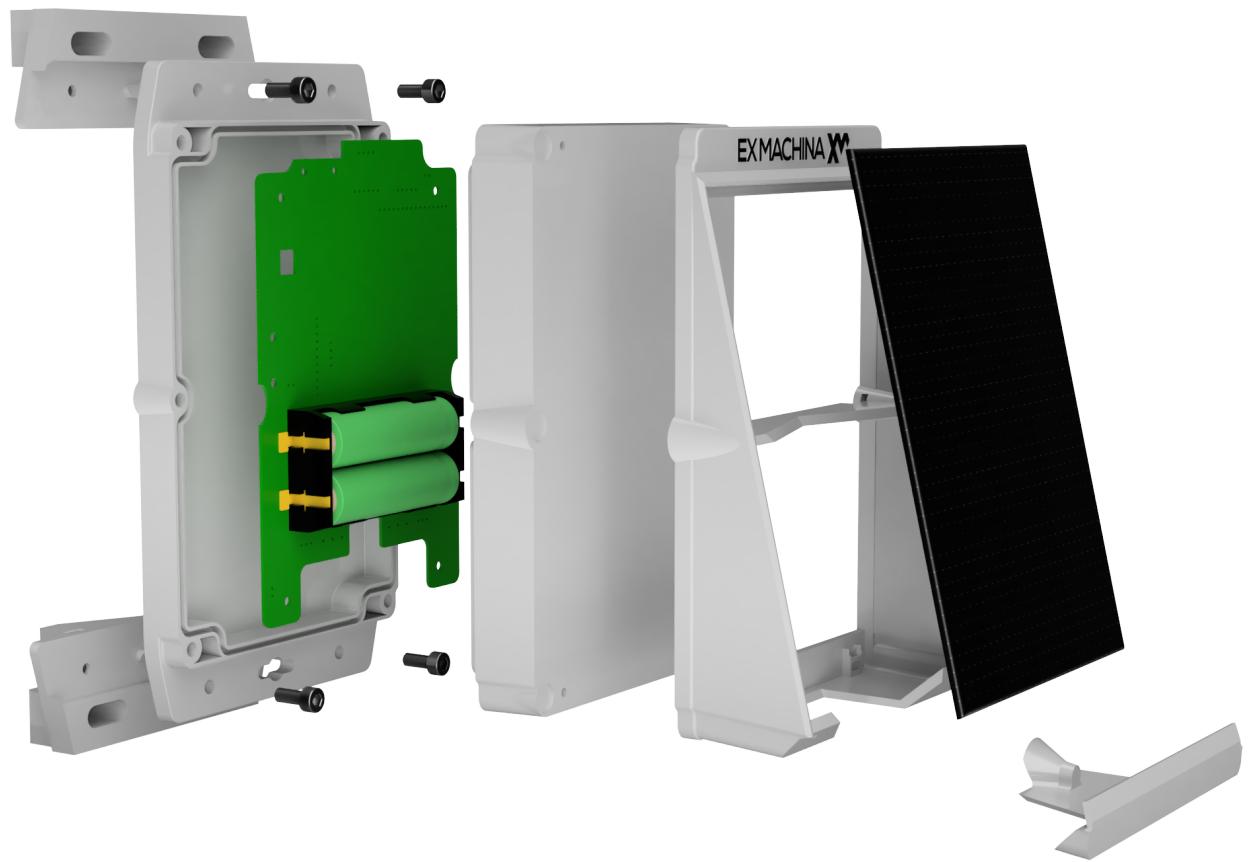
Η τυπωμένη πλακέτα του node προσαρμόστηκε με οπές ώστε να συμπίπτουν με αυτές που υπάρχουν στο εσωτερικό του κουτιού για να στηρίζεται σε αυτό με βίδες.

Για την στήριξη του κουτιού σε τοίχο ή οιστό, σχεδιάστηκαν ειδικά στηρίγματα τα οποία εκτυπώνονται σε 3D εκτυπωτή και βιδώνονται στο πίσω μέρος του κουτιού. Τα στηρίγματα αυτά φέρουν οπές για τοποθέτηση βιδών ή μεταλλικών/πλαστικών δεματικών.



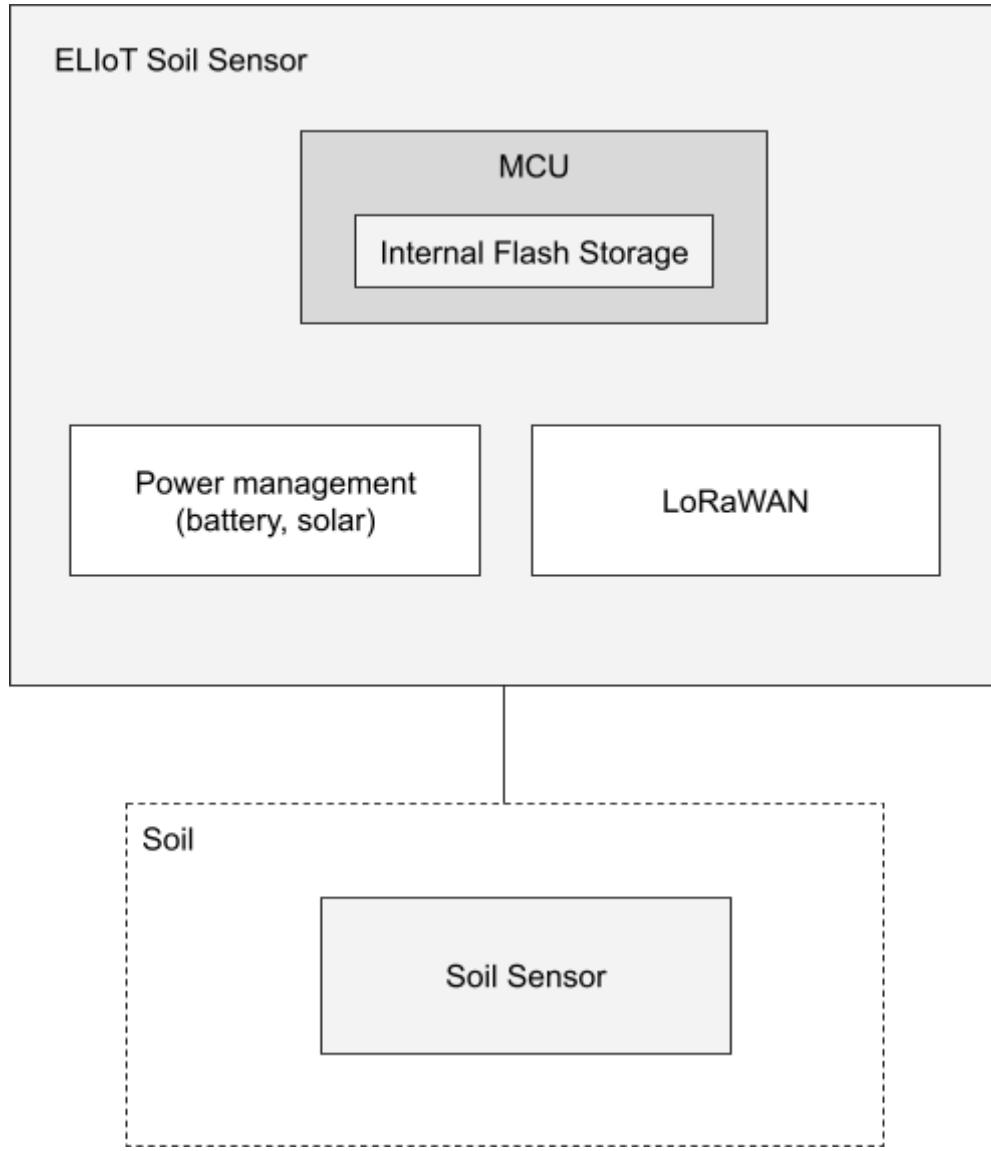
Σχήμα: 3D εκτυπωμένη βάση στήριξης κουτιού

Για την τοποθέτηση του ηλιακού πάνελ, σχεδιάστηκε βάση στην οποία τοποθετείται το πάνελ και η συνολική κατασκευή στηρίζεται πάνω στο κουτί με βίδες οι οποίες δεν καταλήγουν στο εσωτερικό του και συνεπώς δεν επηρεάζουν την αδιαβροχοποίησή του.



Σχήμα: Περίβλημα ELIoT node με τα επιμέρους τμήματά του

## ELIoT Soil Sensor



*Σχήμα: Αρχιτεκτονική ELIoT Sensor Node*

Ο αισθητήρας υγρασίας ELIoT Soil Sensor βασίζεται εξ ολοκλήρου στον ανοιχτού υλικού και λογισμικού αισθητήρα υγρασίας Dragino LSE01, με παραμετροποιημένο λογισμικό.

Αποτελείται από την κυρίως μονάδα που περιλαμβάνει τον μικροελεγκτή, τη μπαταρία, τον πομποδέκτη και τον αισθητήρα ο οποίος τοποθετείται σε συγκεκριμένο βάθος στο έδαφος.

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 Soil Sensor

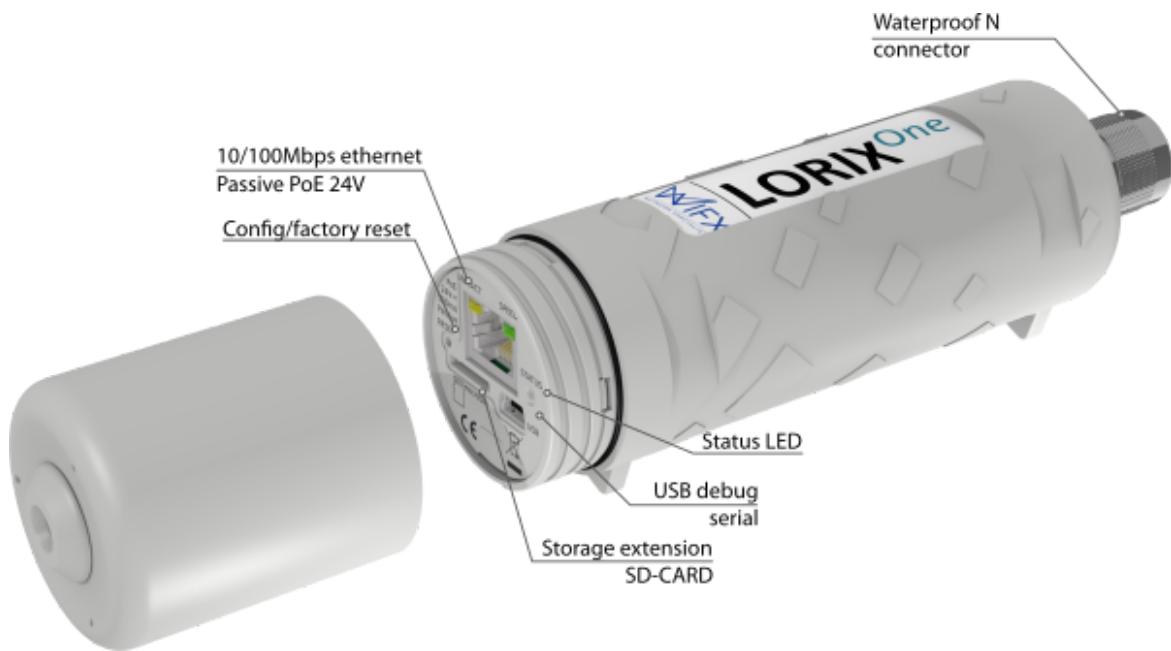
προσαρμόστηκε ώστε να αποστέλλει μια μέτρηση υγρασίας κάθε λίγα λεπτά με αποτέλεσμα η διάρκεια ζωής της μπαταρίας να είναι μεγαλύτερη του 1ος έτους.



## LoRaWAN Gateway

Για τη λήψη των δεδομένων από το EIIoT Soil Sensor απαιτείται η ύπαρξη ενός gateway το οποίο θα προωθεί στο διαδίκτυο τα δεδομένα που θα λαμβάνει ασύρματα μέσω του πρωτοκόλλου LoRaWAN. Μπορεί να χρησιμοποιηθεί οποιοδήποτε LoRaWAN gateway, και για τις ανάγκες αυτού του έργου επιλέχθηκε το Lorix One Gateway το οποίο προορίζεται για λειτουργία σε εξωτερικούς χώρους ενώ ταυτόχρονα είναι οικονομικά προσιτό. 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.

Τα πακέτα που λαμβάνει το gateway από τα EIIoT Soil Sensors τα προωθεί στο cloud και συγκεκριμένα στο LoRa Server. Ο LoRa Server είναι υπεύθυνος για την αποκωδικοποίηση των πακέτων LoRaWAN και την προώθηση τους στις αντίστοιχες εικονικές συσκευές στην πλατφόρμα Open EIIoT, όπου αποθηκεύονται.



Σχήμα: Lorix One gateway

### Ρύθμιση του gateway

Αρχικά πρέπει να ρυθμιστεί το gateway με τη διεύθυνση του LoRa Server και το σωστό packet forwarder που είναι συμβατό με το LoRa Server. Συνδεόμαστε με SSH και εκτελούμε τα παρακάτω εντολές:

```
sudo /etc/init.d/clouds-manager.sh stop
/etc/init.d/clouds-manager.sh configure
```

Ξεκινάει ο οδηγός ρύθμισης όπου απαντάμε "Yes" στο autostart και στην επιλογή του forwarder επιλέγουμε "packet-forwarder".

Στη συνέχεια δημιουργούμε αντίγραφο του αρχείου ρυθμίσεων "local\_conf.json" και το ανοίγουμε για επεξεργασία για να συμπληρώσουμε τα στοιχεία του gateway και τη διεύθυνση από την οποία είναι προσβάσιμος ο LoRa Server:

```
cd /opt/lorix/clouds/packet-forwarder/
cp local_conf.json local_conf.json.org
nano cp local_conf.json
```

Τα πεδία που πρέπει να συμπληρωθούν είναι τα εξής:

- **gateway\_ID** - Το μοναδικό αναγνωριστικό του gateway. Μπορεί να είναι τυχαία τιμή αρκεί η ίδια να συμπληρωθεί και κατά τη προσθήκη του gateway στο LoRa server.
- **server\_address** - Η διεύθυνση από την οποία είναι προσβάσιμος ο LoRa server

- serv\_port\_up/serv\_port\_down - Η θύρα πρόσβασης. Μπορεί να μείνει η προεπιλεγμένη τιμή
- serv\_enabled - Η τιμή παραμένει “true”

Προαιρετικά μπορούν να συμπληρωθούν και τα υπόλοιπα πεδία που κατα βάση περιέχουν πληροφοριακά στοιχεία για το gateway (γεωγραφικά στοιχεία, email διαχειριστή κ.α.)

```
{
  "gateway_conf": {
    "gateway_ID": "0000000000000000",
    "servers": [
      {
        "server_address": "server.address.com",
        "serv_port_up": 1700,
        "serv_port_down": 1700,
        "serv_enabled": true
      }
    ],
    "fake_gps": false,
    "ref_latitude": 0,
    "ref_longitude": 0,
    "ref_altitude": 0,
    "contact_email": "me@company",
    "description": "lorixone"
  }
}
```

Σχήμα: Αρχείο ρυθμίσεων local\_conf.json

Αποθηκεύουμε τις αλλαγές και πραγματοποιούμε επανεκκίνηση του gateway με την εντολή:

sudo reboot

### Ρύθμιση LoRa Server

Αρχικά πρέπει να προστεθεί το gateway ώστε να αναγνωρίζεται από το LoRa Server. Πριν προστεθεί ένα gateway, προηγείται η δημιουργία ενός “gateway profile” το οποίο λειτουργεί ως προφίλ (πρότυπο) όταν υπάρχει η ανάγκη προσθήκης πολλών gateways ίδιου τύπου. Επιλέγουμε το μενού **Gateway Profiles -> Create** και συμπληρώνουμε τα στοιχεία:

- Name - Όνομα που να περιγράφει το συγκεκριμένο τύπο gateway (π.χ. Lorix)
- Enabled Channels - Τα κανάλια που υποστηρίζει το gateway
- Network Server - Το προφίλ του server

## Gateway-profiles / Create

Name \*

LoRix One

A short name identifying the gateway-profile.

Enabled channels \*

0, 1, 2, 3, 4, 5, 6, 7

The channels active in this gateway-profile as specified in the LoRaWAN Regional Parameters specification. Separate channels by comma, e.g. 0, 1, 2. Extra channels must not be included in this list.

Network-server \*

Network Server

[ADD EXTRA CHANNEL](#) [CREATE GATEWAY-PROFILE](#)

Σχήμα: Φόρμα δημιουργίας gateway profile

Αφού δημιουργηθεί το προφίλ, μπορούμε να προχωρήσουμε στη προσθήκη του gateway στο μενού **Gateways -> Create** συμπληρώνοντας τα παρακάτω πεδία:

- Gateway name - Όνομα που περιγράφει το συγκεκριμένο gateway
- Gateway description - Περιγραφή
- Gateway ID - Το μοναδικό αναγνωριστικό του gateway που χρησιμοποιήθηκε και κατά τη συμπλήρωση του αρχείου ρυθμίσεων του gateway.
- Gateway profile - Το προφίλ που δημιουργήθηκε παραπάνω

Πατάμε “Create gateway profile” για να ολοκληρωθεί η διαδικασία.

## Gateways / Create

Gateway name \*

Lorix One #1

The name may only contain words, numbers and dashes.

Gateway description \*

Main gateway

Gateway ID \*

f9 99 8e 44 29 25 6c b6

MSB



Network-server \*

Network Server



Select the network-server to which the gateway will connect. When no network-servers are available in the dropdown, make sure a service-profile exists for this organization.

Gateway-profile

Lorix One



An optional gateway-profile which can be assigned to a gateway. This configuration can be used to automatically re-configure the gateway when LoRa Gateway Bridge is configured so that it manages the packet-forwarder configuration.



Gateway discovery enabled

When enabled (and LoRa Server is configured with the gateway discover feature enabled), the gateway will send out periodical pings to test its coverage by other gateways in the same network.

Σχήμα: Προσθήκη gateway

Για να αναγνωρίζονται τα EIIoT Sensor Nodes από τον LoRa Server θα πρέπει να προστεθούν σε αυτό ως συσκευές και να ρυθμιστούν με τα αντίστοιχα κλειδιά τα οποία χρησιμοποιούνται για ταυτοποίηση των συσκευών. Πακέτα που λαμβάνονται από τον LoRa Server και δεν συνοδεύονται από έγκυρα κλειδιά απορρίπτονται. Οι συσκευές στο LoRa Server ομαδοποιούνται σε μια οντότητα που ονομάζεται "application" η οποία περιλαμβάνει ρυθμίσεις που αφορούν το σύνολο των συσκευών που περιλαμβάνονται σε αυτή. Ταυτόχρονα για να προστεθούν συσκευές θα πρέπει πρώτα να δημιουργηθεί το προφίλ ("Device profile")

που τις περιγράφει. Συνεπώς για να προχωρήσουμε στην προσθήκη των συσκευών θα πρέπει πρώτα να δημιουργηθούν και οι δύο οντότητες (application και device profile).

Η δημιουργία ενός device profile πραγματοποιείται από το μενού Device-Profiles -> Create με την συμπλήρωση των παρακάτω στοιχείων.

- Device profile name - Όνομα που περιγράφει τον τύπο της συσκευής (πχ. ELIoT Sensor Node)
- Network server - O network server στον οποίο θα ανήκει αυτή η συσκευή
- LoRaWAN MAC version - Ορίζεται η τιμή "1.1.0"
- LoRaWAN Regional Parameters revision - Ορίζεται η τιμή "A"

Έπειτα στην καρτέλα ενεργοποιούμε το πεδίο "Device supports OTAA" και πατάμε "Create device profile" για να ολοκληρωθεί η διαδικασία.

**Device-profiles / Create**

GENERAL	JOIN (OTAA / ABP)	CLASS-B	CLASS-C	CODEC
<b>Device-profile name *</b> ELIoT LoRa Node				
A name to identify the device-profile.				
<b>Network-server *</b> Network Server				
The network-server on which this device-profile will be provisioned. After creating the device-profile, this value can't be changed.				
<b>LoRaWAN MAC version *</b> 1.1.0				
The LoRaWAN MAC version supported by the device.				
<b>LoRaWAN Regional Parameters revision *</b> A				
Revision of the Regional Parameters specification supported by the device.				
<b>Max EIRP *</b> 0				
Maximum EIRP supported by the device.				
<b>CREATE DEVICE-PROFILE</b>				

### Σχήμα: Προσθήκη device profile

Για τη δημιουργία application επιλέγεται το μενού Applications -> Create και συμπληρώνονται τα παρακάτω πεδία:

- Application name - Όνομα αντιπροσωπευτικό των συσκευών που θα περιέχει
- Application description - Συνοπτική περιγραφή

- Service profile - Service profile στο οποίο ανήκει το application

## Applications / Create

Application name \*

The name may only contain words, numbers and dashes.

Application description \*

Service-profile \*

The service-profile to which this application will be attached. Note that you can't change this value after the application has been created.

Payload codec

By defining a payload codec, LoRa App Server can encode and decode the binary device payload for you. **Important note:** they payload fields have moved to the device-profile. For backward-compatibility and migration, existing codec settings are still visible. Codec settings on the device-profile have priority over the application codec settings.

[CREATE APPLICATION](#)

Σχήμα: Δημιουργία application

Πατάμε “Create application” για να ολοκληρωθεί η διαδικασία και το νέο application προστίθεται στη λίστα με τα υπάρχοντα. Επιλέγουμε το νέο application και μεταφερόμαστε στη λίστα με τα devices που ανήκουν σε αυτό, η οποία σε αυτό το στάδιο είναι κενή.

Πριν προχωρήσουμε στη προσθήκη συσκευών πρέπει να γίνουν επιπλέον ρυθμίσεις οι οποίες αφορούν την προώθηση των δεδομένων από το LoRa Server στην IoT πλατφόρμα του Open ELIoT (integration). Από την καρτέλα “Integrations” επιλέγουμε “Create” και από τη λίστα “Integration kind” επιλέγουμε “Thingsboard.io”. Συμπληρώνουμε τη διεύθυνση από την οποία είναι προσβάσιμη πλατφόρμα του OpenELIoT και ολοκληρώνουμε πατώντας “Create integration”.

Integration kind \*

ThingsBoard.io

ThingsBoard.io integration configuration

ThingsBoard.io server \*

openeliot.exm.gr

Each device must have a 'ThingsBoardAccessToken' variable assigned. This access-token is generated by ThingsBoard.

**CREATE INTEGRATION**

### Σχήμα: Προσθήκη integration στο application

Τέλος, επιστρέφουμε στην καρτέλα “Devices” και πατάμε “Create” για να προσθέσουμε μια συσκευή ELIoT Sensor Node, συμπληρώνοντας τα παρακάτω πεδία:

- Device name - Το όνομα του συγκεκριμένου ELIoT Sensor Node
- Device description - Σύντομη περιγραφή της συσκευής
- Device EUI - Μοναδικό αναγνωριστικό της συσκευής. Πατώντας το εικονίδιο δημιουργείται μια τυχαία τιμή. Το αναγνωριστικό αυτό χρησιμοποιείται για την ταυτοποίηση της συσκευής και θα πρέπει να χρησιμοποιηθεί κατά τη ρύθμιση και τον προγραμματισμό του ELIoT Sensor Node που αντιπροσωπεύει.
- Device profile - Το device profile που δημιουργήθηκε παραπάνω.

GENERAL	VARIABLES	TAGS
Device name *		
eliot-lora-node-1		
The name may only contain words, numbers and dashes.		
Device description *		
Soil moisture sensor #1		
Device EUI *	c8 fe d2 3f 69 d3 bf 83	MSB C
Device-profile *	ELIoT LoRa Node	
<input type="checkbox"/> Disable frame-counter validation		
Note that disabling the frame-counter validation will compromise security as it enables people to perform replay-attacks.		
<a href="#">CREATE DEVICE</a>		

### Σχήμα: Δημιουργία συσκευής

Πατάμε “Create device” για να ολοκληρωθεί η διαδικασία. Η συσκευή δημιουργείται και μεταφέρομαστε αυτόματα στη φόρμα “Keys (OTAA)” όπου πρέπει να συμπληρωθούν τα κλειδιά ταυτοποίησής της. Τα κλειδιά αυτά θα πρέπει να είναι τα ίδια που χρησιμοποιήθηκαν κατά τον προγραμματισμό του συγκεκριμένου ELIoT Sensor Node. Σε περίπτωση που δεν έχουν δημιουργηθεί ακόμα κλειδιά, πατώντας το αντίστοιχο εικονίδιο  δημιουργείται νέο τυχαίο κλειδί το οποίο έπειτα μπορεί να προγραμματιστεί στο ELIoT Sensor Node.

DETAILS

CONFIGURATION

KEYS (OTAA)

ACTIVATION

DEVICE DATA

LOR&gt;

Network key (LoRaWAN 1.1) \*

cb 44 f9 67 02 c3 87 d7 a8 68 8b cc 6e 11 b5 fd

MSB



For LoRaWAN 1.1 devices. In case your device does not support LoRaWAN 1.1, update the device-profile first.

Application key (LoRaWAN 1.1) \*

ba 91 4b 97 ca a6 a6 f2 32 8a 1b d0 45 2f c4 02

MSB



For LoRaWAN 1.1 devices. In case your device does not support LoRaWAN 1.1, update the device-profile first.

[SET DEVICE-KEYS](#)

**Σχήμα: Προσθήκη κλειδιών στη συσκευή.**

Πατάμε "Set device keys" για να ολοκληρωθεί η διαδικασία και μεταφερόμαστε αυτόματα στη σελίδα του application που δημιουργήσαμε και τη λίστα των συσκευών που ανήκουν σε αυτή.

## 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

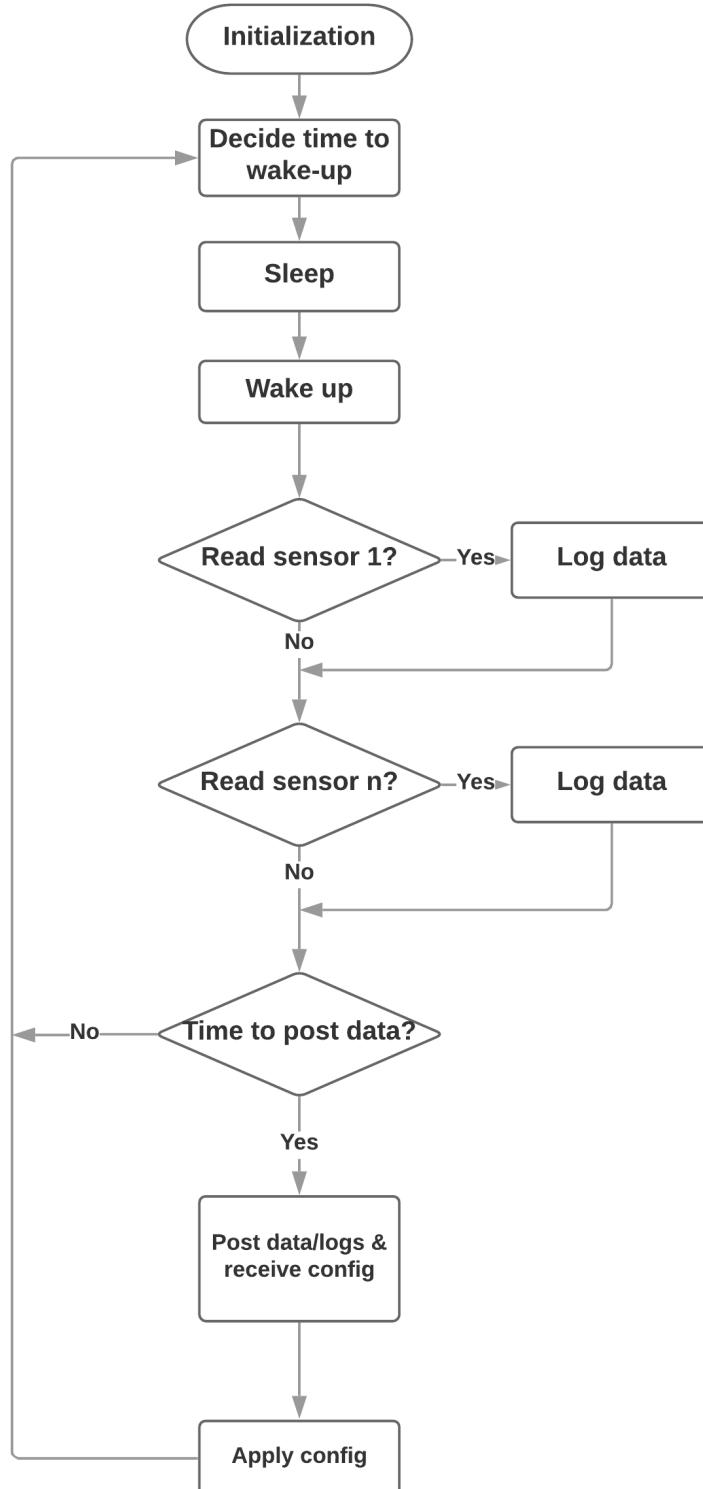
Κατά την εκκίνηση το node αρχικοποιεί όλες τις παραμέτρους του και κάνει απόπειρα συγχρονισμού του RTC μέσω GPRS (NTP). Αφού ολοκληρωθεί η αρχικοποίηση, το node μπαίνει στον φυσιολογικό κύκλο λειτουργίας.

## 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. Τα 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 πλέον 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

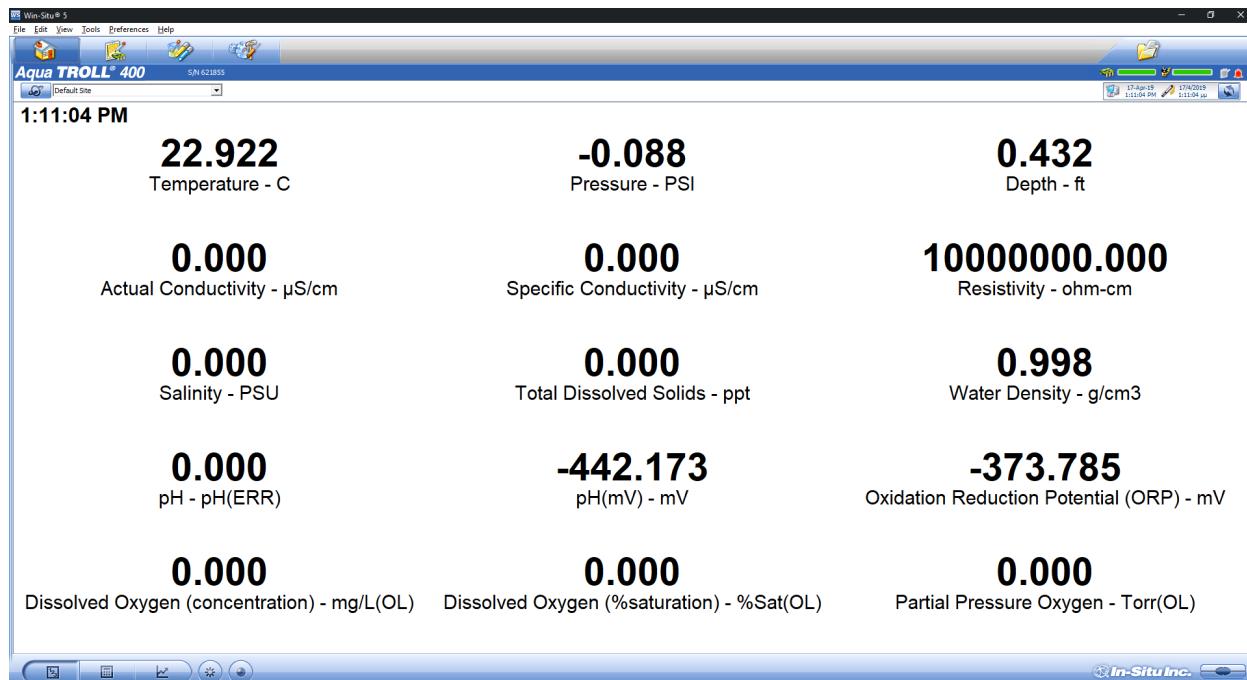
διαπιστωθεί αν το καινούριο λογισμικό μπορεί να επικοινωνεί ακόμα με την πλατφόρμα έτσι ώστε σε περίπτωση που το τελευταίο είναι προβληματικό, να υπάρχει τουλάχιστον η δυνατότητα να εκτελεστεί ξανά ο απομακρυσμένος έλεγχος ώστε να του δοθεί εκ νέου η εντολή για επόμενο λογισμικό. Σε περίπτωση που το 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, 500, 600

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.

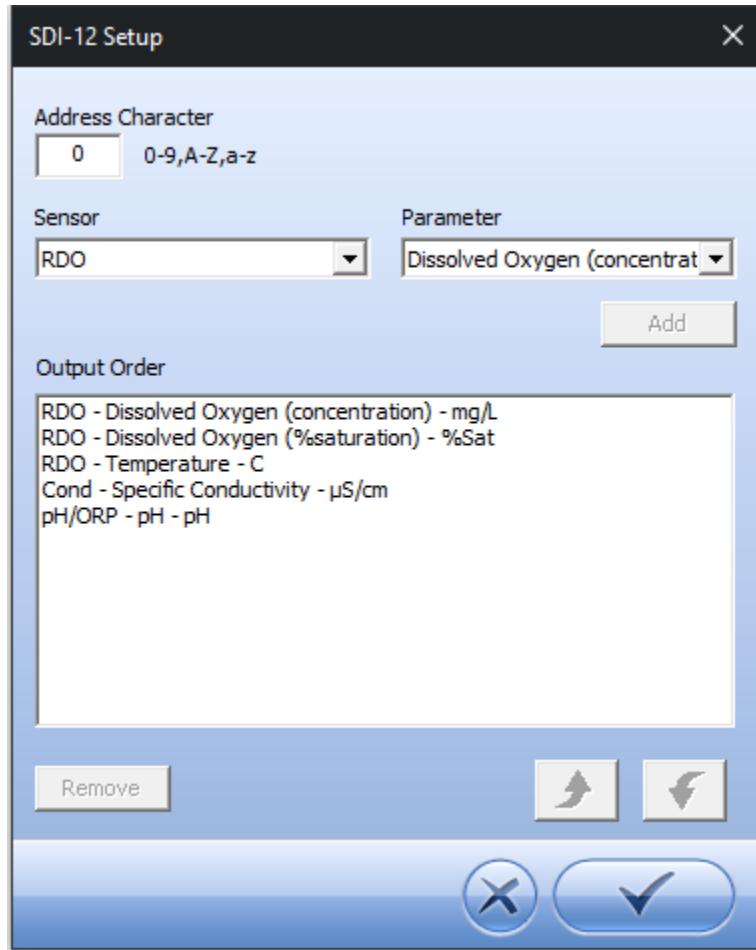
Εκκινώντας την εφαρμογή, αν η σύνδεση με τον αισθητήρα ήταν επιτυχής εμφανίζεται η αρχική οθόνη όπου και φαίνονται οι τρέχουσες τιμές των παραμέτρων όπως φαίνεται στο σχήμα.



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 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.



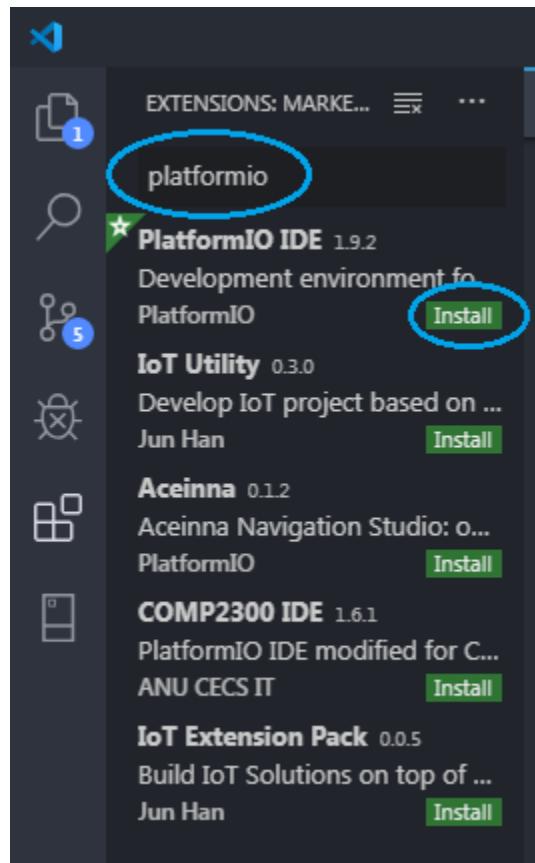
Σχήμα: Παράμετροι αισθητήρα και η σειρά που πρέπει να έχουν

## 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

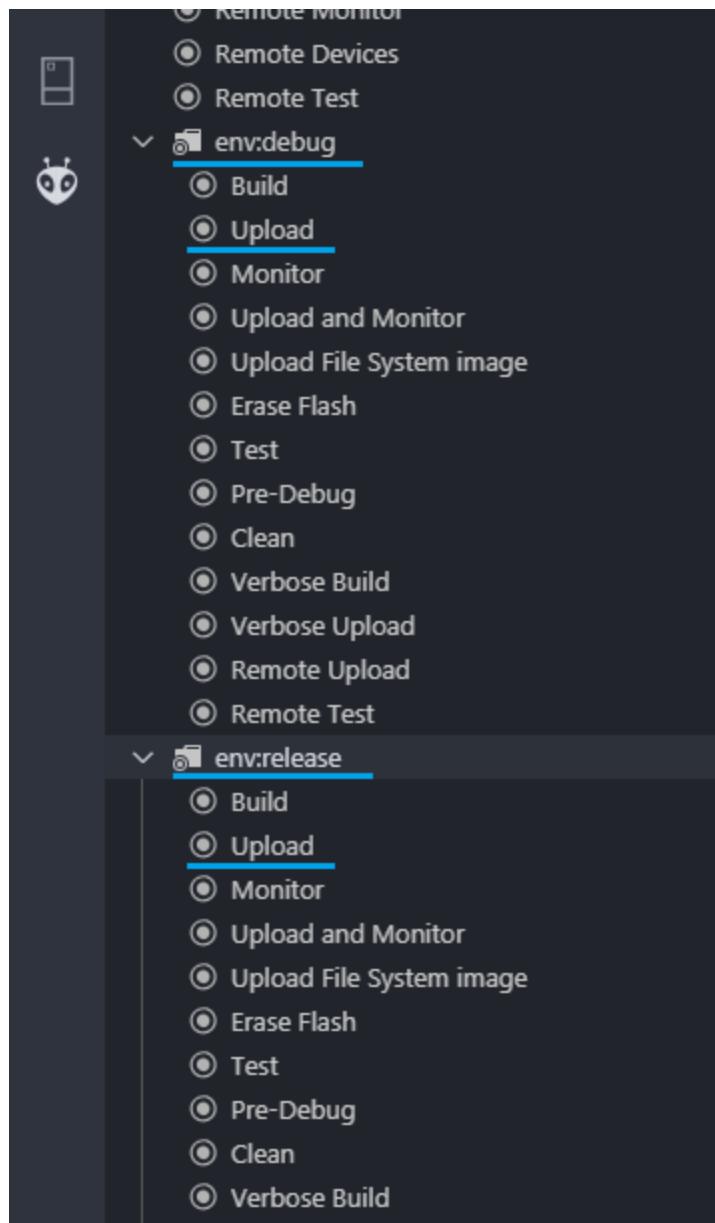
“Visual Studio Code” (ή VSCode) με το πρόσθετο “PlatformIO” το οποίο προσθέτει τη δυνατότητα προγραμματισμού μικροελεγκτών. Ο συνδυασμός των 2 αυτών ανοικτών λογισμικών αποτελεί σύνηθισμένη λύση στην ανάπτυξη εφαρμογών για την πλατφόρμα Arduino και όχι μόνο.

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 ή 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.

```
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...
=====
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 αντιστοίχιση 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.

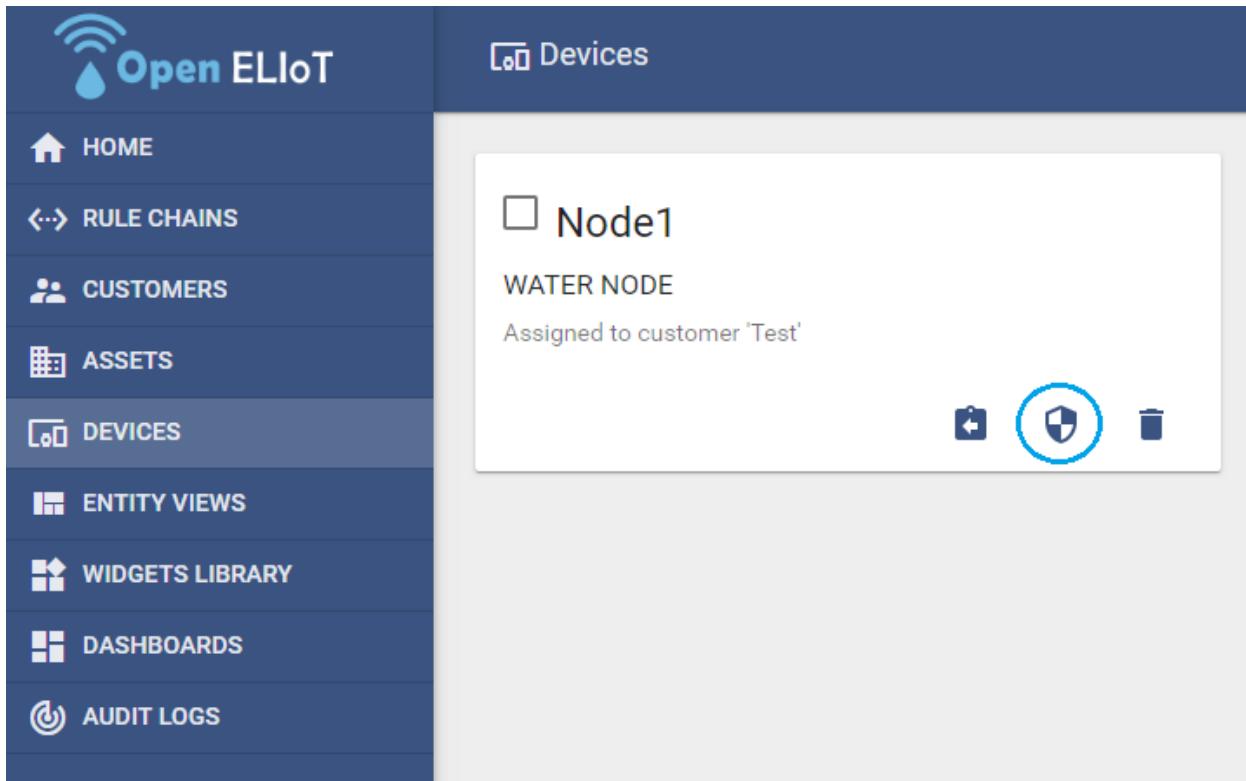
Πατώντας "Add" η συσκευή δημιουργείται και η διαδικασία ολοκληρώνεται.

The screenshot shows a modal dialog titled "Add Device". It contains the following fields:

- Name \***: Node1
- Device type \***: WATER NODE
- Label**: (empty)
- Is gateway**: (unchecked)

At the bottom of the dialog are two buttons: "ADD" (highlighted in blue) and "CANCEL".

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.

```
10
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

 Upload New Configuration

Last config data ID: 1571911594

Last activity time: Fri Oct 20 2019 09:24:26 GMT+0...

Water sensors measure interval (mins)

10

Call home interval (mins)

30

OTA update

Reboot

Format SPIFFS

**SAVE**

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
Firmware binary URL	<a href="http://myfileserver.com/files/firmware_v101.bin">http://myfileserver.com/files/firmware_v101.bin</a>
Firmware MD5	7435792a30bfaca4b561a882b85cf19
Water sensor measure interval (mins)	32 / 32
Call home interval (mins)	10
Water sensor measure interval (mins)	30
OTA update	<input checked="" type="checkbox"/>
Reboot	<input type="checkbox"/>
Format SPIFFS	<input type="checkbox"/>
<b>SAVE</b>	

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)	5,3,0
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

## Παράτημα προδιαγραφών εξαρτημάτων (datasheets)

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

