



Ανοιχτή υποδομή Internet of Things για online υπηρεσίες περιβάλλοντος Open Internet of Things infrastructure for online environmental services, Open ELIoT

ΤΙΤΛΟΣ ΠΑΡΑΔΟΤΕΟΥ

2.1

“ Τεχνικός οδηγός ανάπτυξης αισθητήρων μέτρησης περιβαλλοντικών παραμέτρων - 1η έκδοση”

ΚΩΔΙΚΟΣ ΕΡΓΟΥ

T1EAK-01613

EXMACHINA

**WEATHER
ON DEMAND**



Αθήνα, 2019

Οι βιβλιογραφικές αναφορές στις εργασίες της παρούσας έκθεσης παρακαλούμε να γίνονται σύμφωνα με τον ακόλουθο τρόπο:

Ex Machina¹2019. Τεχνικός οδηγός ανάπτυξης αισθητήρων μέτρησης περιβαλλοντικών παραμέτρων - 1η έκδοση. Ανοιχτή υποδομή Internet of Things για online υπηρεσίες περιβάλλοντος (Open ELIoT) - 54 σελ.

¹ Ομάδα Έργου

ΠΕΡΙΕΧΟΜΕΝΑ

1. Αντικείμενο, Σκοπός και Ομάδα Έργου.....	5
1.1. Αντικείμενο και σκοπός της παρούσας μελέτης	5
1.2. Ομάδα έργου.....	5
2. Αρχιτεκτονική λύσης - εξοπλισμός.....	6
3. ELIoT Node (v1)	8
3.1. Κεντρικός μικροελεγκτής και βασική πλακέτα.....	9
3.2. Επικοινωνίες GSM.....	11
3.3. Πρωτόκολλα επικοινωνίας με αισθητήρες SDI-12.....	12
3.4. Αισθητήρας μέτρησης ποιότητας υδάτων	13
3.5. Αισθητήρας μέτρησης στάθμης.....	14
3.6. Μετεωρολογικός σταθμός.....	15
3.7. Ασύρματα πρωτόκολλα επικοινωνίας με πλατφόρμα	19
3.8. Διαχείριση ενέργειας	19
3.9. Περιβάλημα	20
3.10. Βάση στήριξης πλακετών	21
4. ELIoT LoRa Node	23
4.1. Κεντρικός μικροελεγκτής	23
4.2. Αισθητήρας μέτρησης υγρασίας.....	24
4.3. LoRaWAN Gateway	24
4.3.1 Ρύθμιση του gateway.....	25
4.4. Ρύθμιση LoRa Server	26
4.5. Διαχείριση ενέργειας	32
4.6. Περιβάλημα	32
5. Διάγραμμα συνδεσης - συναρμολόγηση	33
TTGo TFox	34
DFRobot Solar Manager	34
SIM7000	34
Arduino Nano 5V	34
Level Shifter.....	34
DS2312 RTC.....	34
BME280.....	35
6. Υλισμικό (Firmware).....	35
6.1. Εκκίνηση	35
6.2. Κύκλος Λειτουργίας.....	35
6.3. Επικοινωνία με την πλατφόρμα.....	36

6.4.	Μέτρηση παραμέτρων αισθητήρων	37
6.5.	Μνήμη και διαχείριση αρχείων	37
6.6.	Καταγραφή συμβάντων (Logging).....	37
6.7.	Ωρα	37
6.8.	FOTA (Firmware Over The Air).....	38
7.	Αρχικοποίηση συσκευής.....	38
7.1.	Αισθητήρας ποιότητας εδάφους AquaTROLL 400	38
7.2.	Προγραμματισμός firmware	40
7.3.	Σύνδεση με πλατφόρμα	44
7.4.	Πρόσθεση της συσκευής στην πλατφόρμα	44
7.5.	Ρυθμίσεις σύνδεσης στο υλισμικό.....	46
7.6.	Εύρεση διεύθυνσης MAC ενός node.....	46
7.7.	Απομακρυσμένος έλεγχος / ενημέρωση λογισμικού (FOTA)	47
7.7.1.	Απομακρυσμένος έλεγχος	47
7.7.2.	Απομακρυσμένη ενημέρωση λογισμικού (FOTA)	48
7.7.3.	Testing / debugging.....	49
7.7.4.	Έκδοση debug / release	49
7.7.5.	Logs.....	50
8.	Προβλήματα, επόμενα βήματα	51
8.1.	Σφάλματα στις επικοινωνίες GSM	51
	Παράρτημα προδιαγραφών εξαρτημάτων (datasheets)	54

1. Αντικείμενο, Σκοπός και Ομάδα Έργου

1.1. Αντικείμενο και σκοπός της παρούσας μελέτης

Η παρούσα τεχνική έκθεση συμπεριλαμβάνεται στην ενότητα εργασίας "ΠΕ 2: Ανάπτυξη έξυπνων αισθητήρων IoT και πλατφόρμας συλλογής, επεξεργασίας και οπτικοποίησης περιβαλλοντικών δεδομένων" και αποτελεί το παραδοτέο Π2.1 με τίτλο "Τεχνικός οδηγός ανάπτυξης αισθητήρων μέτρησης περιβαλλοντικών παραμέτρων - 1η έκδοση ". Στόχος της συγκεκριμένης τεχνικής έκθεσης είναι η καταγραφή των απαιτήσεων των χρηστών και των προδιαγραφών των έξυπνων αισθητήρων.

1.2. Ομάδα έργου

EXM

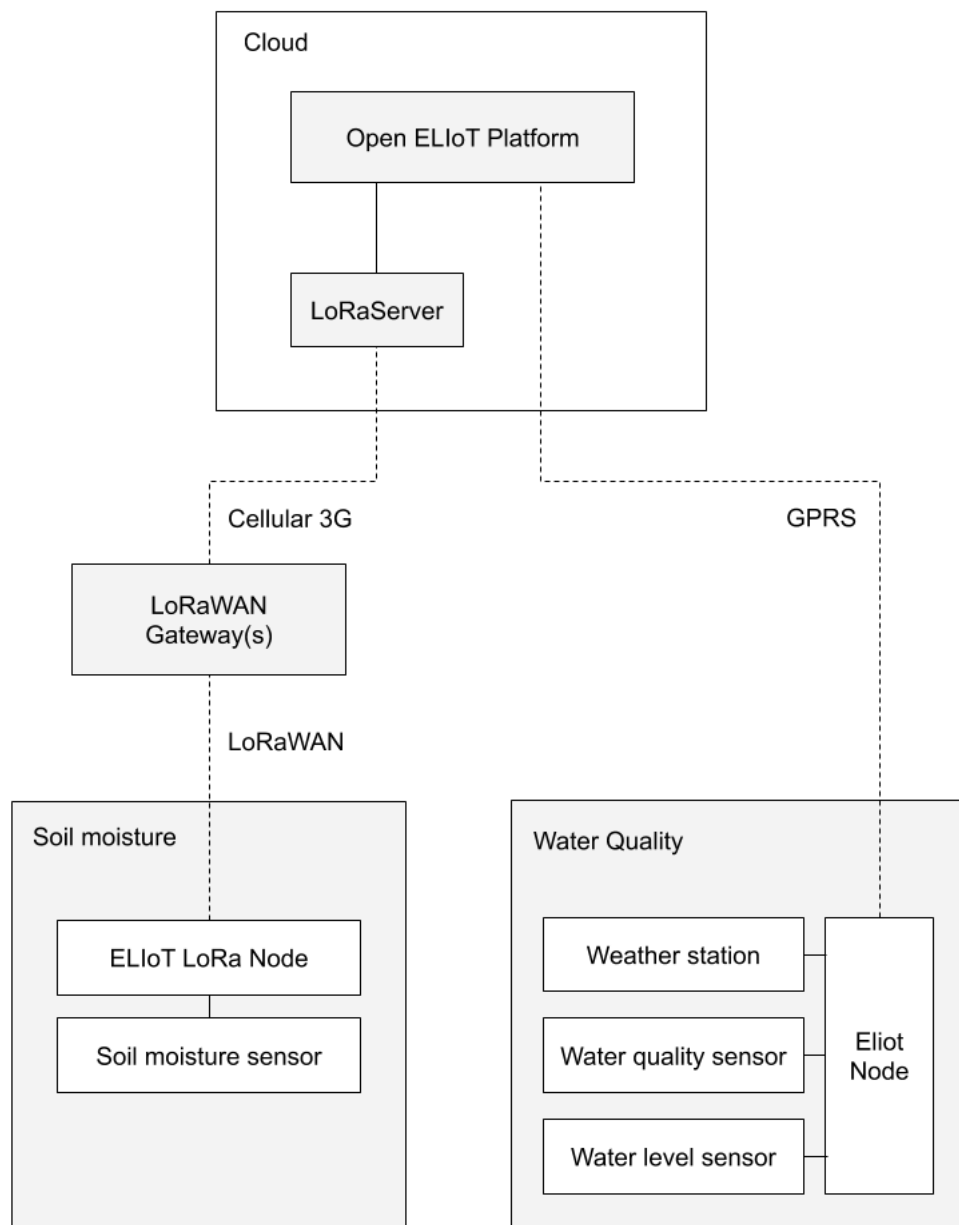
Βασίλειος Χρυσός, Μηχανικός Περιβάλλοντος, MSc, EXM

Εμμανουήλ Νικηφοράκης, Μηχανικός Ηλεκτρονικών Υπολογιστών, MSc, EXM

Ευστράτιος Θεοδώρου, Ηλεκτρολόγος Μηχανικός & Μηχανικός Υπολογιστών, MSc, EXM

Νικόλαος Τσιλιγκαρίδης, Μηχανικός Βιομηχανικού Σχεδιασμού, EXM

2. Αρχιτεκτονική λύσης - εξοπλισμός



Σχήμα 1: Αρχιτεκτονική λύσης

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

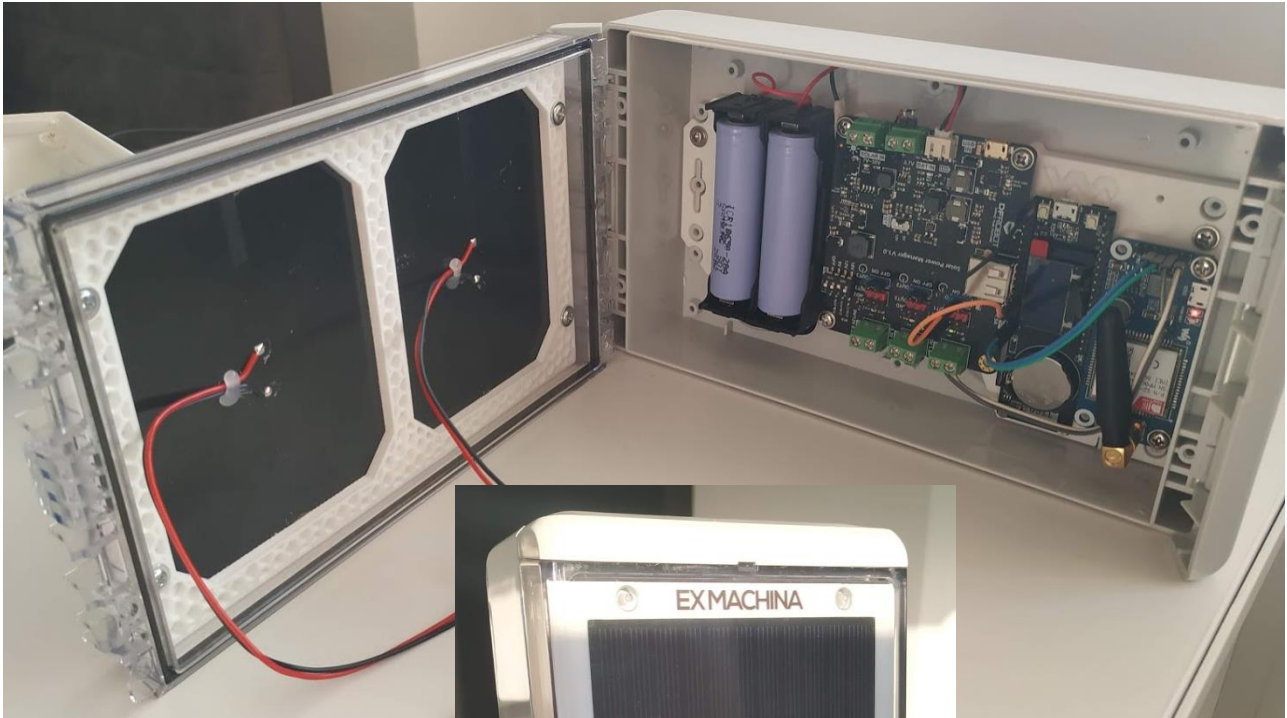
- Σταθμός μέτρησης ποιότητας υδάτων.
- Σταθμός μέτρησης υγρασίας εδάφους

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 χωρίς να χρειάζεται SIM κάρτα (GPRS). Αντί αυτού, απαιτείται ένα 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 όλα μέσα από μια μόνο SIM κάρτα.

3. ELIoT Node (v1)

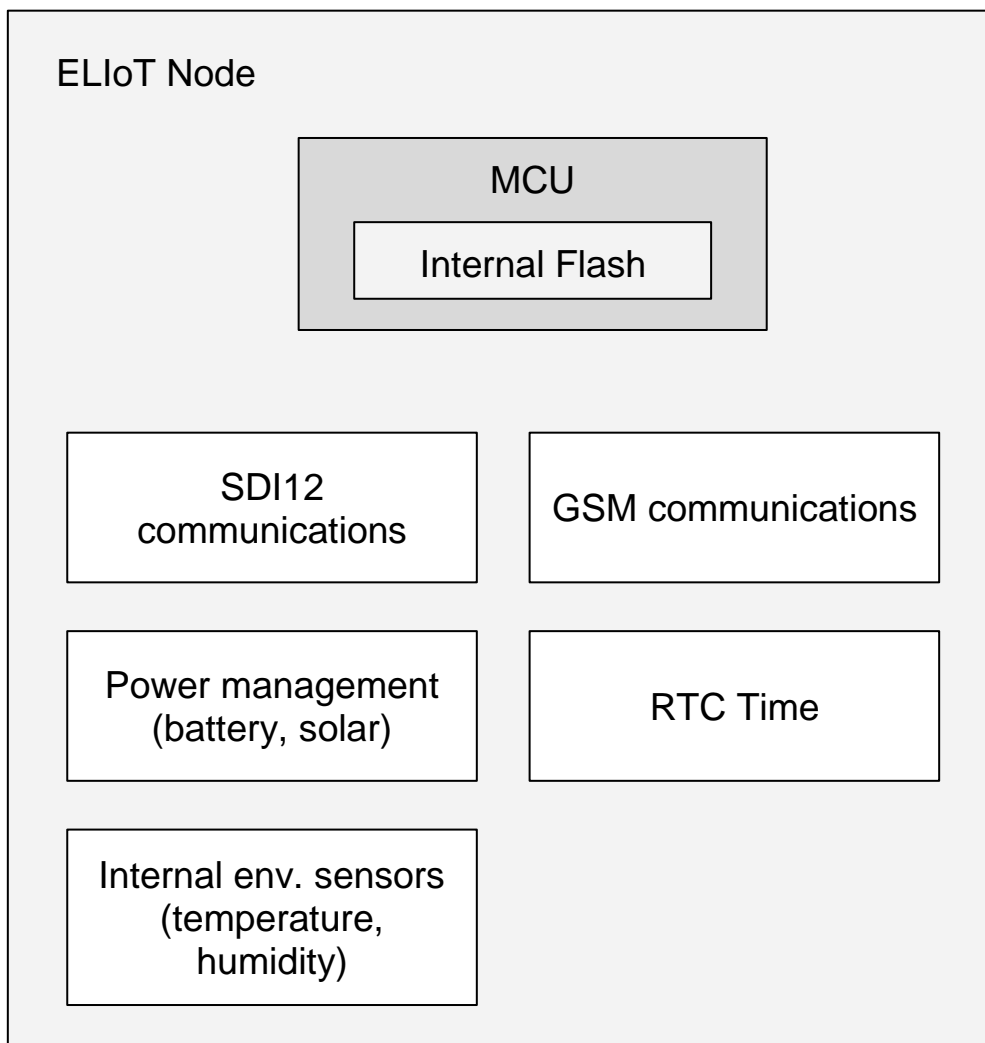


ELIoT Node V1

2W solar panel behind
transparent lid
Mounted via 3D
printed skeleton

2x 18650 Lithium
batteries
DF Robot Solar





Σχήμα 2: 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.

3.1. Κεντρικός μικροελεγκτής και βασική πλακέτα

Βασικό κριτήριο στην επιλογή του μικροελεγκτή είναι να υποστηρίζεται 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

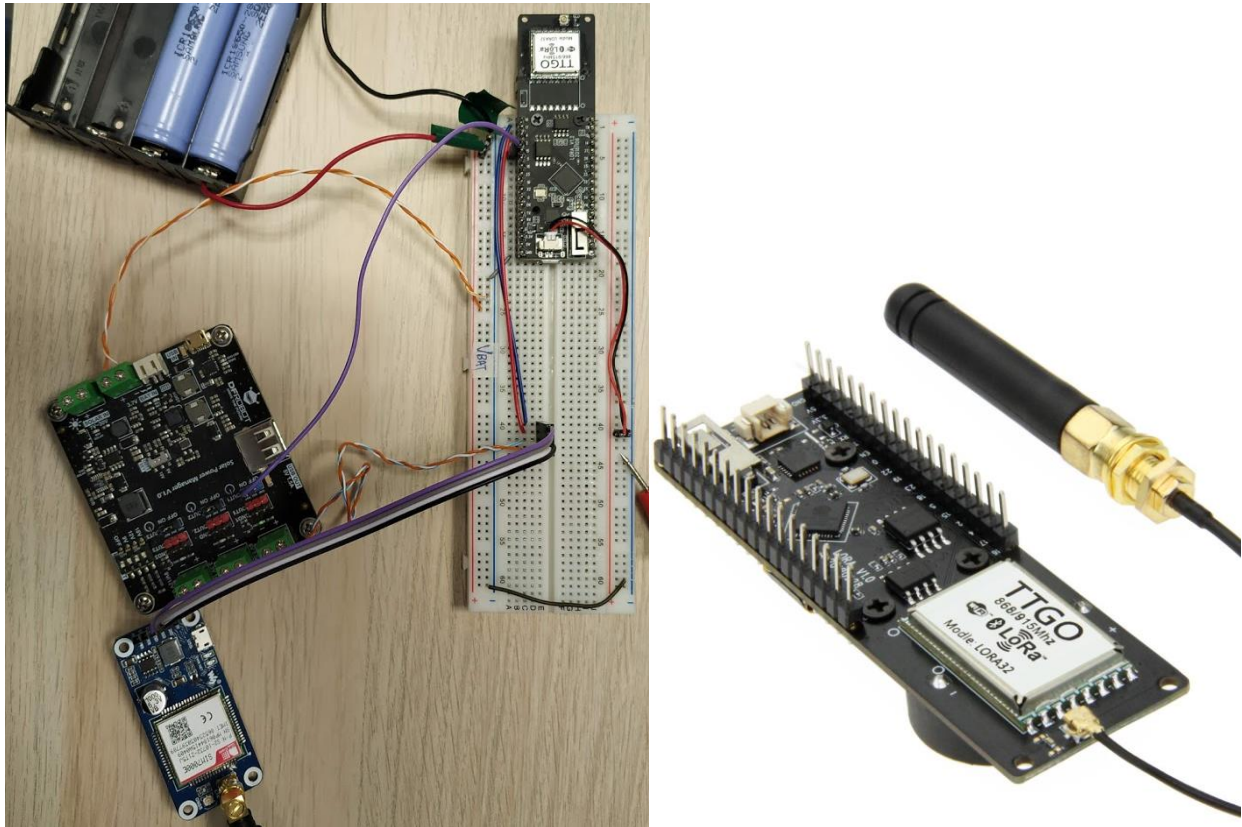
παρλλαγών σε ενσωματωμένα περιφερειακά όπως κυκλώματα επικοινωνίας LoRa, GSM, κυκλώματα φόρτισης, οθόνες κ.α.

Στον Πίνακα 1 αναλύονται κάποιες από τις πιο δημοφιλείς πλακέτες ανάπτυξης.

#	Name	Module	Charge current (mA)	Current idle Battery (mA)	Current deep sleep from battery (μA)	Approx price, USD
1	WMOS /18650	WROOM-32	530	125	15000	10
2	LOLIN32/Wemos	WROOM-32	210	43	125	8
3	TTGO	WROOM-32	94	45	172	6
4	TTGO Mini	ESP32 Chip	150	41	1500 (LED)	5
5	TTGO Pico	Pico Chip	160	42 (LED)	880 (LED)	9
6	LOLIN32 Pro	WROVER-32	160	38	1700	N/A
7	Wemos Lolín Lite	ESP32 Chip	164	40	800	5
8	Higrow	WROOM-32	530	109	23000	14
9	Wemos Pro	ESP32 Chip	250	43	790	9
10	FireBeetle	WROOM-32	250	51	53	19
11	Bare ESP	WROOM-32	N/A	43	8	N/A
20	TTGO ePaper	ESP32 Chip	260	50	10000 (w/o LED)	25
21	Wemos	WROOM-32	N/A	N/A	N/A	N/A
22	TTGO TS V1.2	ESP Chip	400	86	N/A	11
23	TTGO T4	ESP Chip	370	140	N/A	18
24	TTGO? Pro V2	ESP Chip	170	85	N/A	13
25	WiFi Kit32	ESP Chip	240		N/A	13
26	TTGO LoRa V2	ESP Chip	240	88	N/A	17
27	TTGO TFox	ESP Chip	500	40	900	29

Πίνακας 1: Πλακέτες ανάπτυξης ESP32

Από τις πλακέτες ανάπτυξης που εξετάστηκαν, η TTGO TFox είναι η πλησιέστερη στις απαιτήσεις με κύκλωμα φόρτισης μπαταρίας, χαμηλή κατανάλωση ενέργειας, εξωτερικό ρολόι RTC, ενώ το ενσωματωμένο LoRa module που χρησιμοποιείται στο ELIoT LoRa Node μειώνει το κόστος και απλοποιεί το κύκλωμα αφού δεν χρειάζεται να αγοραστεί μια επιπλέον εξωτερική πλακέτα. Αρα το TTGO TFox μπορεί να αποτελέσει την καρδιά και των δυο σταθμών (GSM ή LoRa) με την προσθήκη ενός GSM module στην περίπτωση του πρώτου. Επίσης είναι απαραίτητη η χρήση κάποιου solar power manager που θα φορτίζει τις μπαταρίες, καθώς και θα παρέχει ρεύμα υψηλότερης τάσης, απαραίτητο για τους περιφερειακούς αισθητήρες (5V & 12V).



Σχήμα 3: TTGO TFox prototype circuit

3.2. Επικοινωνίες GSM

Για την μεταφορά των δεδομένων στο cloud μέσω internet πρέπει να προστεθεί στο TTGO TFOX ένα GSM / GPRS module χαμηλού κόστους, χαμηλής ενεργειακής κατανάλωσης με ώριμη υποστήριξη από τις βιβλιοθήκες στο arduino οικοσύστημα.

Μετα απο εκτεταμένη μελέτη τεχνικών χαρακτηριστικών των παρακάτω οικογενειών απο cellular modems:

- SIMCom SIM800 series
- SIMCom SIM900 series
- SIMCom LTE / NBIoT / 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 χρυσή τομή δυνατοτήτων και κόστους, το οποίο υποστηρίζει GSM, NBIoT, LTE-CATM1 και GPS.

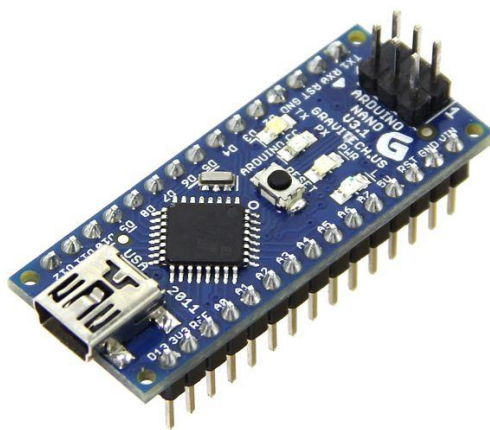


Σχήμα 4. SIM7000 modules που χρησιμοποιήθηκαν στην αναπτυξη του ELIoT node v1

3.3. Πρωτόκολλα 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 χωρίς 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.



Σχήμα 4. 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.

3.4. Αισθητήρας μέτρησης ποιότητας υδάτων

Ο πολυαισθητήρας ποιότητας υδάτων που υλοποιείται είναι ο Aqua TROLL 400 της In-situ και έχει την ικανότητα να μετράει έως 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 5: Πολυαισθητήρας AquaTROLL 400

3.5. 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 (ADC) όπου ο αισθητήρας παράγει τάση ανάλογη της απόστασης της επιφάνειας του νερού από τον αισθητήρα (για κάθε $\sim 3.2\text{mV}$ αντιστοιχούν 2cm απόστασης).



Σχήμα 6: Αισθητήρας στάθμης υπερήχων MaxBotix MB7066



Σχήμα 7: Αισθητήρας στάθμης σε αδιαβροχο 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

3.6. Μετεωρολογικός σταθμός

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



Σχήμα 8. WS-2902A external sensors all-in-one

Outdoor measurement specifications

Temperature range: $-40 \sim +60^{\circ}\text{C}$

Resolution: 0.1°C

Measuring range rel. humidity: $10\% \sim 99\%$

Accuracy : $\pm 5\%$

Rain volume display: $0 - 9999.9\text{mm}$

Resolution: 0.3mm (if rain volume $< 1000\text{mm}$)

1mm (if rain volume $> 1000\text{mm}$)

Wind speed: $0-50\text{m/s}$ ($0 \sim 100\text{mph}$)

Accuracy: $\pm 1\text{m/s}$ (wind speed $< 5\text{m/s}$)

$\pm 10\%$ (wind speed $> 5\text{m/s}$)

Wind direction: 0 to 359 degree

Light: $0-400\text{k Lux}$

Accuracy : $\pm 15\%$

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).



Σχήμα 9. Metergroup Atmos 41

Measurements specifications:

Solar radiation

Range: 0 to 1750 W/m²

Resolution: 1 W/m²

Accuracy: $\pm 5\%$ of measurement typical

Precipitation

Range: 0 to 400 mm/h

Resolution: 0.017 mm

Accuracy: $\pm 5\%$ 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, $\pm 3\%$ 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: $\pm 5^\circ$

Tilt

Range: -90° to +90°

Resolution: 0.1°

Accuracy: $\pm 1^\circ$

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

3.7. Ασύρματα πρωτόκολλα επικοινωνίας με πλατφόρμα

Αν και η πλατφόρμα υποστηρίζει MQTT / COAP, το HTTP πρωτόκολλο είναι αρκετό για τις ανάγκες των nodes.

3.8. Διαχείριση ενέργειας

Οι μπαταρίες λιθίου 18650 προσφέρουν μια καλή σχέση κόστους (~5€), διαθεσιμότητας και χωρητικότητας (~2500 - 3000 mAh)

Αναλογα το μέγεθος του ηλιακού πανελ (2 ή 4/5 Watt) αντιστοιχεί και το πληθος των μπαταριων (2 ή 4) και με αυτο το τροπο εχουμε δυο επιλογές διαθέσιμης ενέργειας για τα nodes ανάλογα με τις ανάγκες των περιφερειακών αισθητήρων.



ELIoT Node V1

2W solar panel behind
transparent lid
Mounted via 3D printed
skeleton

2x 18650 Lithium
batteries
DF Robot Solar
manager
TTGO TFOX ESP32
Waveshare SIM7000
NB IoT



3.9. Περίβλημα

Η απαίτηση είναι για αδιάβροχο οικονομικό κουτί, που θα χωράει τα εξαρτήματα, θα μπορεί να τοποθετηθεί σε ιστό και συγχρόνως θα επιτρέπει την ενσωμάτωση του solar panel ώστε ολη η μονάδα / Node να είναι ενιαία .



Bopla 231 x 125 x 40 mm
x 170 x 40 mm

vs

Bopla 271

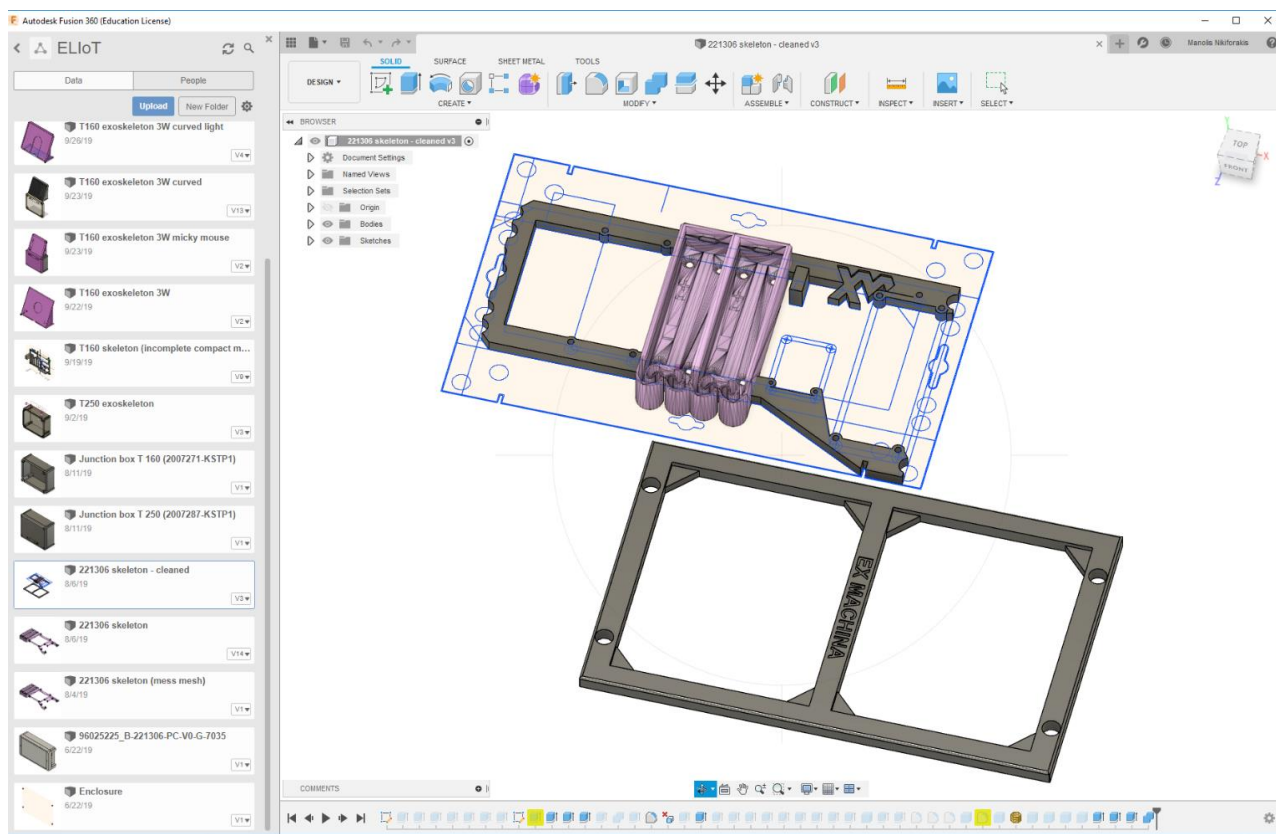
Σχήμα 10. Κουτιά Bopla

PC UL 94 V0 (material is suitable for outdoor use; f1-listing according to UL 746C)
PU seal, foamed;

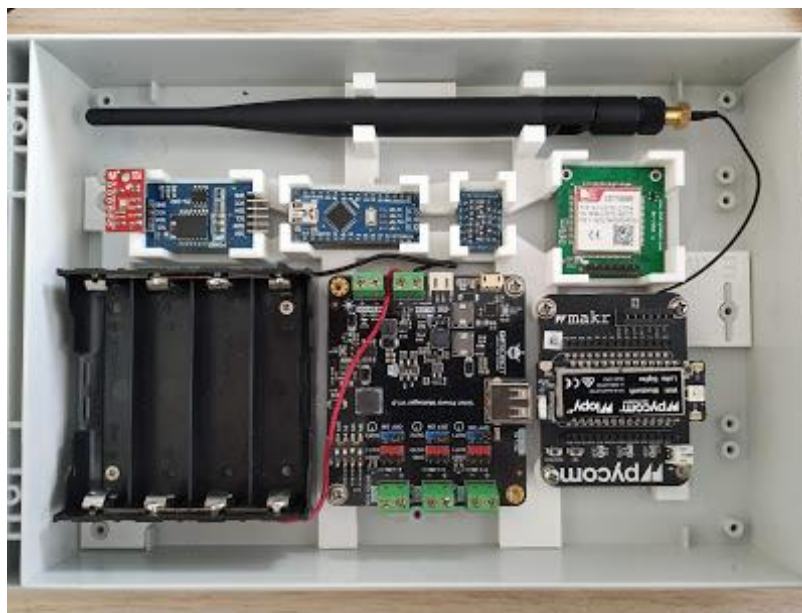
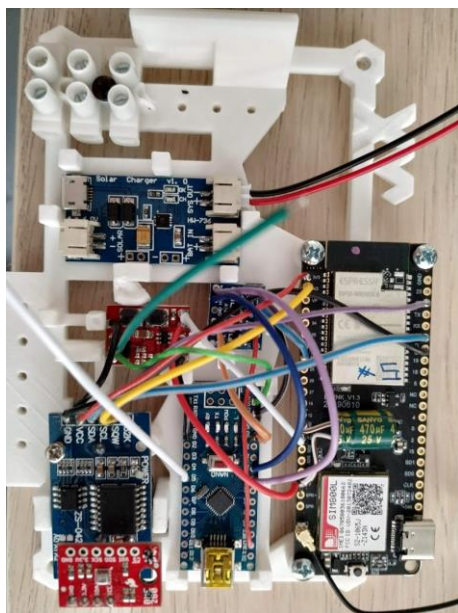
Το μικρό Bopla κουτί μπορεί να φιλοξενήσει φωτοβολταϊκά συνολικής ισχύος 2W ενώ το μεγάλο 5W, και ταυτόχρονα είναι πιο ευρύχωρο συνεπώς δίνει τη δυνατότητα χρήσης περισσότερων μπαταριών για μεγαλύτερη αυτονομία. Σε δοκιμαστική λειτουργία τα φωτοβολταϊκά και οι μπαταρίες που χωρούν στο μικρό κουτί κρίθηκαν ικανοποιητικά για την αυτόνομη λειτουργία του συστήματος, με τη λειτουργία σε συνθήκες μηδενικής ηλιοφάνειας να ξεπερνάει τις 2 εβδομάδες.

3.10. 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.

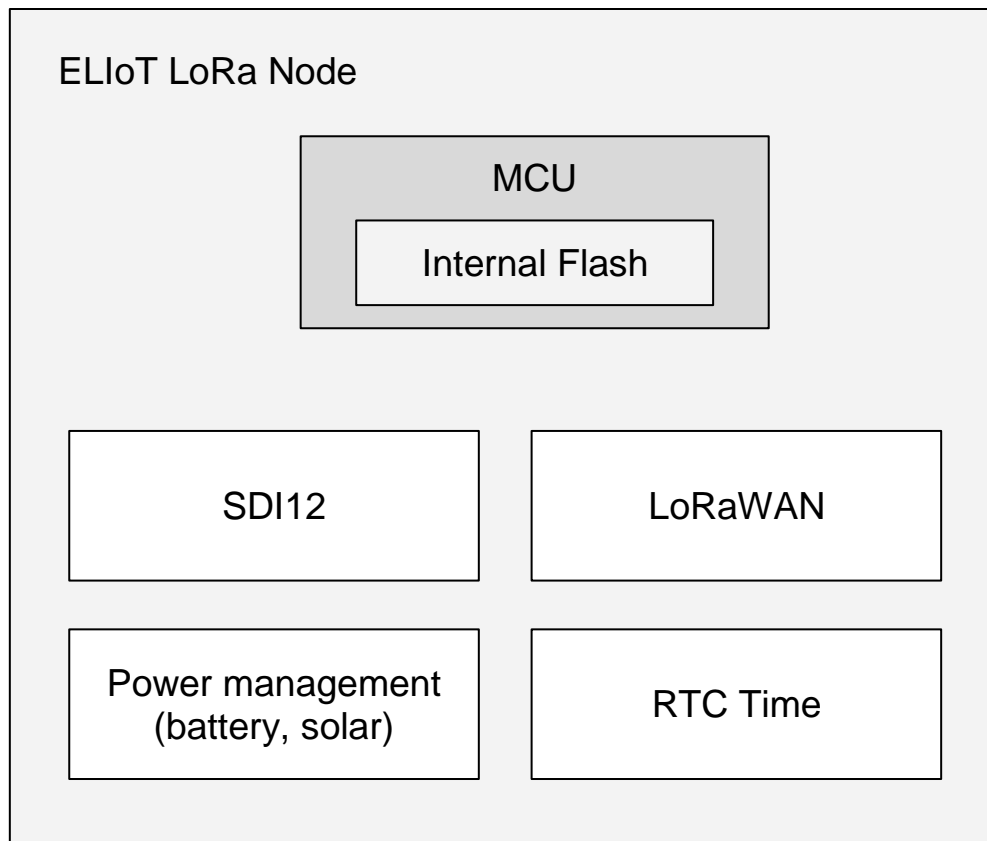


Σχήμα 12: Σχεδιασμός 3D printed βάσης στήριξης



Σχήμα 13: 3D printed βάσεις στήριξης με τις πλακέτες τοποθετημένες

4. ELIoT LoRa Node



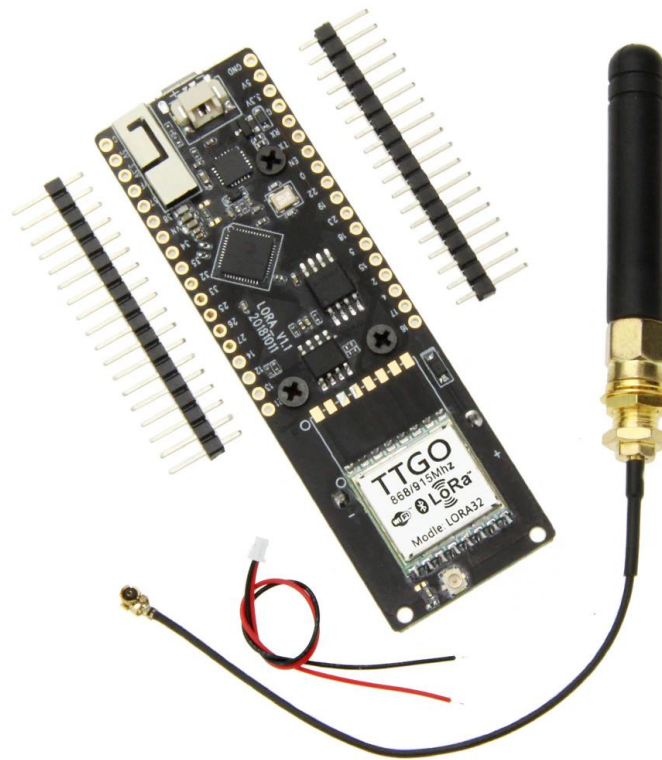
Σχήμα 14: Αρχιτεκτονική ELIoT LoRa Node

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.

4.1. Κεντρικός μικροελεγκτής

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



Σχήμα 15: TTGO T-Fox

4.2. 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.



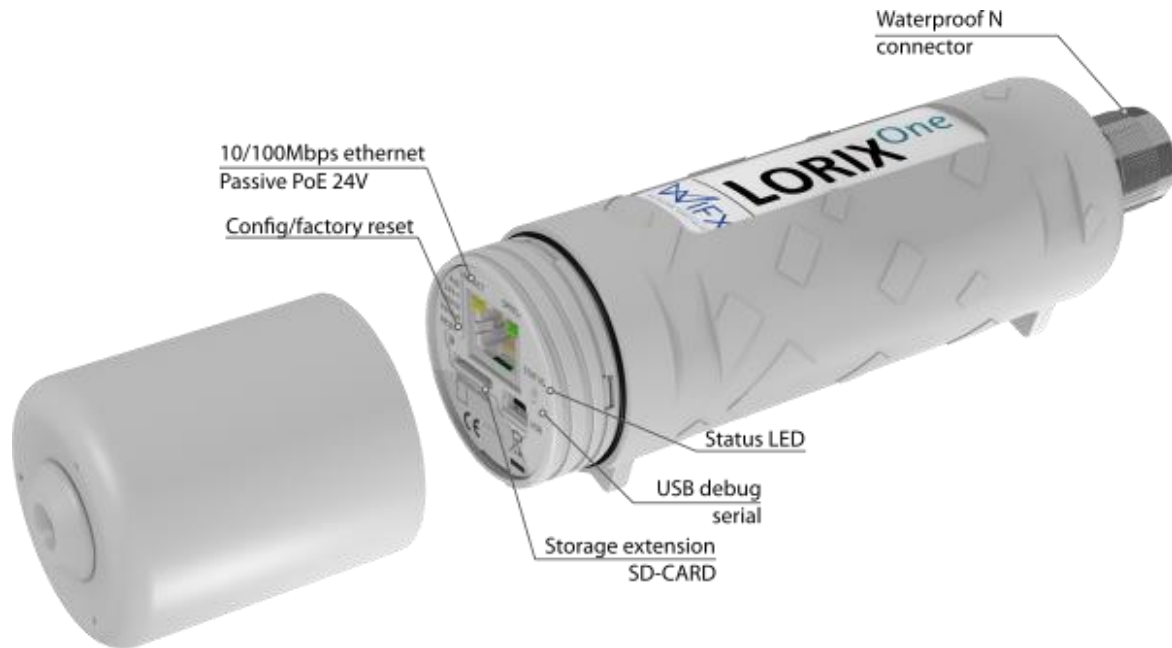
Σχήμα 16: Αισθητήρας μέτρησης υγρασίας εδάφους

4.3. LoRaWAN Gateway

Για τη λήψη των δεδομένων από το ELIoT LoRa Node 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 ELIoT LoRa Nodes τα προωθεί στο cloud και συγκεκριμένα στο LoRa Server. Ο LoRa Server είναι υπεύθυνος για την αποκωδικοποίηση των πακέτων LoRaWAN και την προώθησή τους στις αντίστοιχες εικονικές συσκευές στην πλατφόρμα Open ELIoT, όπου αποθηκεύονται.



Σχήμα 17: Lorix One gateway

4.3.1 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 18: Αρχείο ρυθμίσεων local_conf.json

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

```
sudo reboot
```

4.4. 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 sepecification. 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 19: 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 20: Προσθήκη gateway

Για να αναγνωρίζονται τα ELIoT LoRa 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 LoRa Node)

- Network server - Ο 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

Device-profile name *

ELIoT LoRa Node

A name to identify the device-profile.

Network-server *

Network Server

The network-server on which this device-profile will be provisioned. After creating the device-profile, this value can't be changed.

LoRaWAN MAC version *

1.1.0

The LoRaWAN MAC version supported by the device.

LoRaWAN Regional Parameters revision *

A

Revision of the Regional Parameters specification supported by the device.

Max EIRP *

0

Maximum EIRP supported by the device.

CREATE DEVICE-PROFILE

Σχήμα 21: 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 *

eliot-environmental-monitoring

The name may only contain words, numbers and dashes.

Application description *

Monitoring environmental parameters with ELIoT LoRa Nodes

Service-profile *

Environmental monitoring

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

None

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 22: 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 23: Προσθήκη integration στο application

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

- Device name - Το όνομα του συγκεκριμένου ELIoT LoRa 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 LoRa 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 

Device-profile *

ELIoT LoRa Node


☐ Disable frame-counter validation

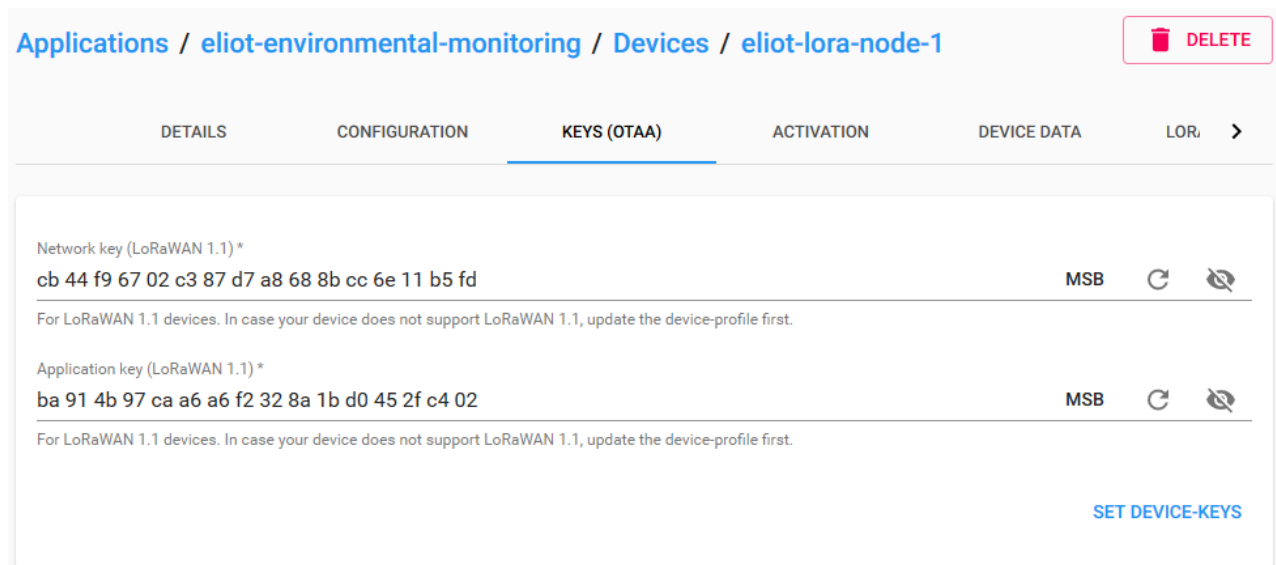
Note that disabling the frame-counter validation will compromise security as it enables people to perform replay-attacks.

CREATE DEVICE

Σχήμα 24: Δημιουργία συσκευής



Πατάμε “Create device” για να ολοκληρωθεί η διαδικασία. Η συσκευή δημιουργείται και μεταφερόμαστε 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 LoRa 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 LoRa Node.





Applications / eliot-environmental-monitoring / Devices / eliot-lora-node-1 DELETE

DETAILS CONFIGURATION **KEYS (OTAA)** ACTIVATION DEVICE DATA LOR >

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 25: 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.

4.5. 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 μια 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.

4.6. 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.

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

TTGo TFox

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

DFRobot Solar Manager

Υπεύθυνο για την φόρτιση των μπαταριών από ηλιακή ενέργεια και την παραγωγή όλων των τάσεων που χρειάζονται για να λειτουργήσουν 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.

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

6. 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.

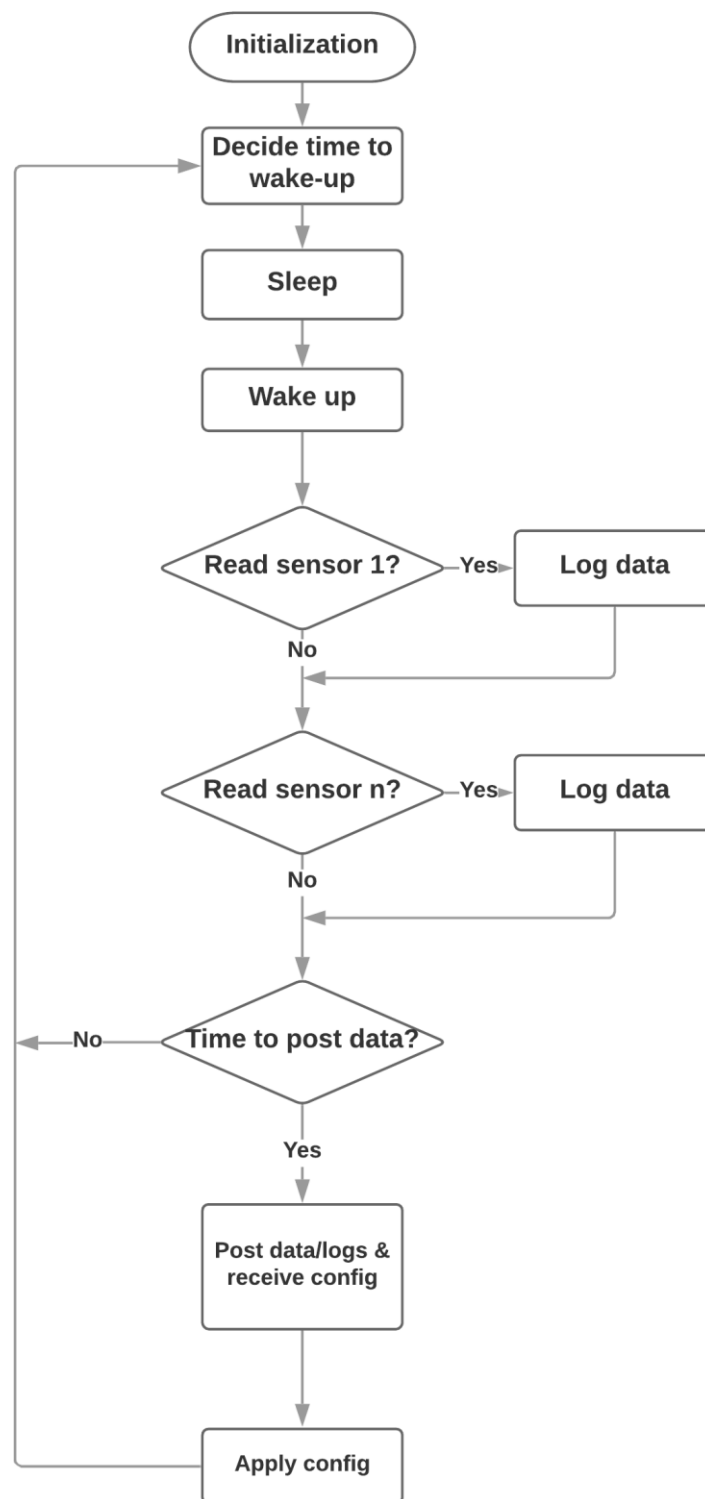
6.1. Εκκίνηση

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.

6.2. 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 28: Κύκλος λειτουργίας

6.3. 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 κατά 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)

6.4. 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.

6.5. Μνήμη και διαχείριση αρχείων

Το ESP32 διαθέτει μνήμη flash η οποία χρησιμοποιείται για την αποθήκευση του ίδιου του υλισμικού αλλά 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.

6.6. 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 κατά την διαδικασία επικοινωνίας με αυτή και βοηθούν στην παρακολούθηση της λειτουργίας του node αλλά και στην διάγνωση σφαλμάτων αν αυτά προκύψουν.

6.7. 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.

6.8. 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.

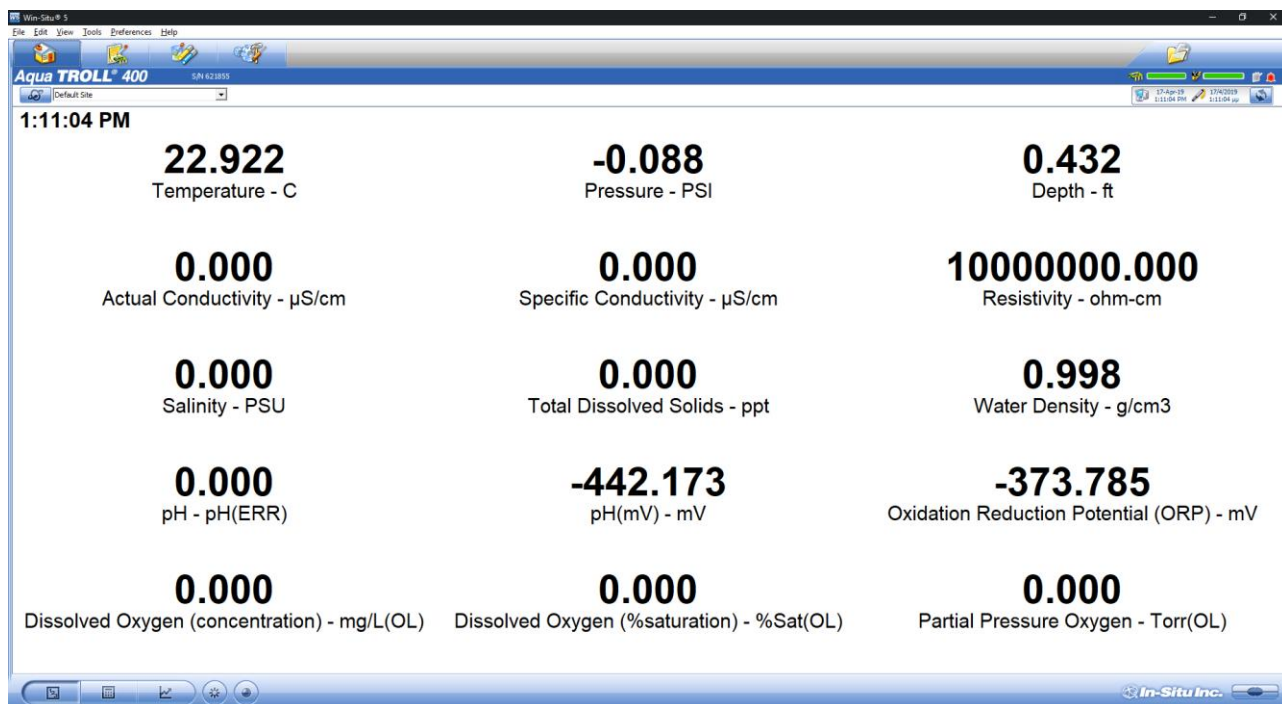
Το node πλέον εκτελεί τη νέα έκδοση του υλισμικού, όμως για να μπορέσει να συνεχίσει θα πρέπει να περάσει επιτυχώς το OTA self test αμέσως μετά την πρώτη εκκίνησή του. Κατά το self test γίνεται απόπειρα λήψης δεδομένων από την πλατφόρμα ως έλεγχος για να διαπιστωθεί αν το καινούριο λογισμικό μπορεί να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

7. Αρχικοποίηση συσκευής

7.1. 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.

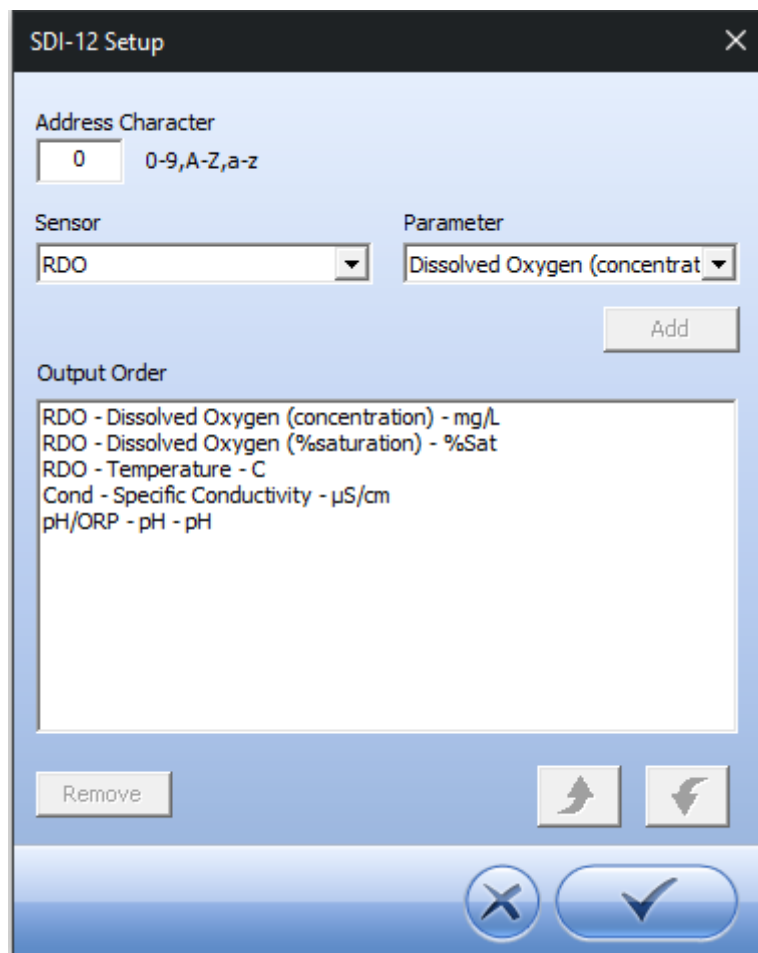
Εκκινώντας την εφαρμογή, αν η σύνδεση με τον αισθητήρα ήταν επιτυχής εμφανίζεται η αρχική οθόνη όπου και φαίνονται οι τρέχουσες τιμές των παραμέτρων όπως φαίνεται στο σχήμα.



Σχήμα 29: Αρχική οθόνη Win-Situ με τον αισθητήρα συνδεδεμένο

Κατά την πρώτη εκκίνηση ενδέχεται ο χρήστης να ερωτηθεί εάν 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 επιστρέφει 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 αντιστοιχισθούν στα σωστά πεδία.

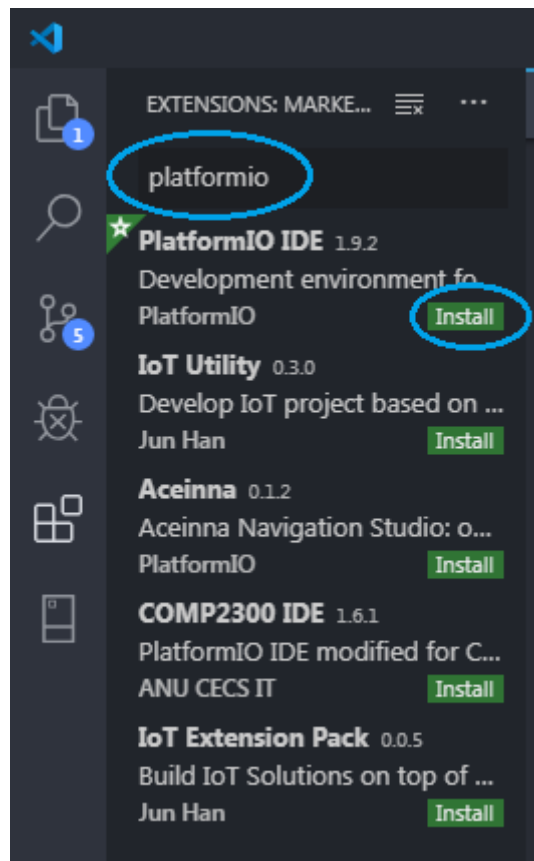


Σχήμα 30: Παράμετροι αισθητήρα και η σειρά που πρέπει να έχουν

7.2. Προγραμματισμός firmware

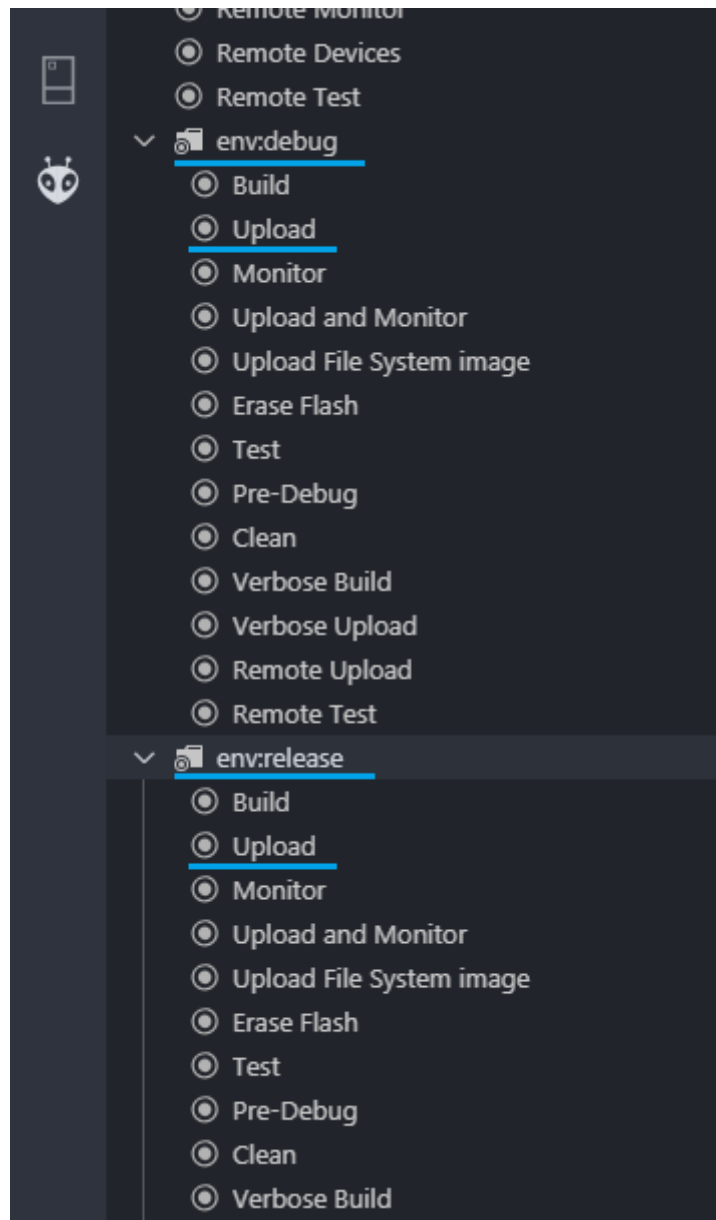
Για να προγραμματιστεί το υλισμικό πρέπει αρχικά να μεταγλωτιστεί σε δυαδικό αρχείο το οποίο θα εγγραφεί στη μνήμη του μικροελεγκτή. Η διαδικασία αυτή μπορεί να εκτελεστεί με διάφορα λογισμικά, στην παρούσα περίπτωση χρησιμοποιήθηκε το περιβάλλον ανάπτυξης “Visual Studio Code” (ή VSCode) με το πρόσθετο “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 31: 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 32: 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 33: 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** μεταγλωττίζεται 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 επιλεγθεί η επιλογή “upload” χωρίς να προσδιοριστεί κάποια από τις 2 εκδόσεις, θα προγραμματιστεί από προεπιλογή η έκδοση debug.

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

```
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 34: Επιτυχής ολοκλήρωση της διαδικασίας όπως φαίνεται στην κονσόλα

7.3. 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 μεταγλωττίζεται 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.

7.4. Πρόσθεση της συσκευής στην πλατφόρμα

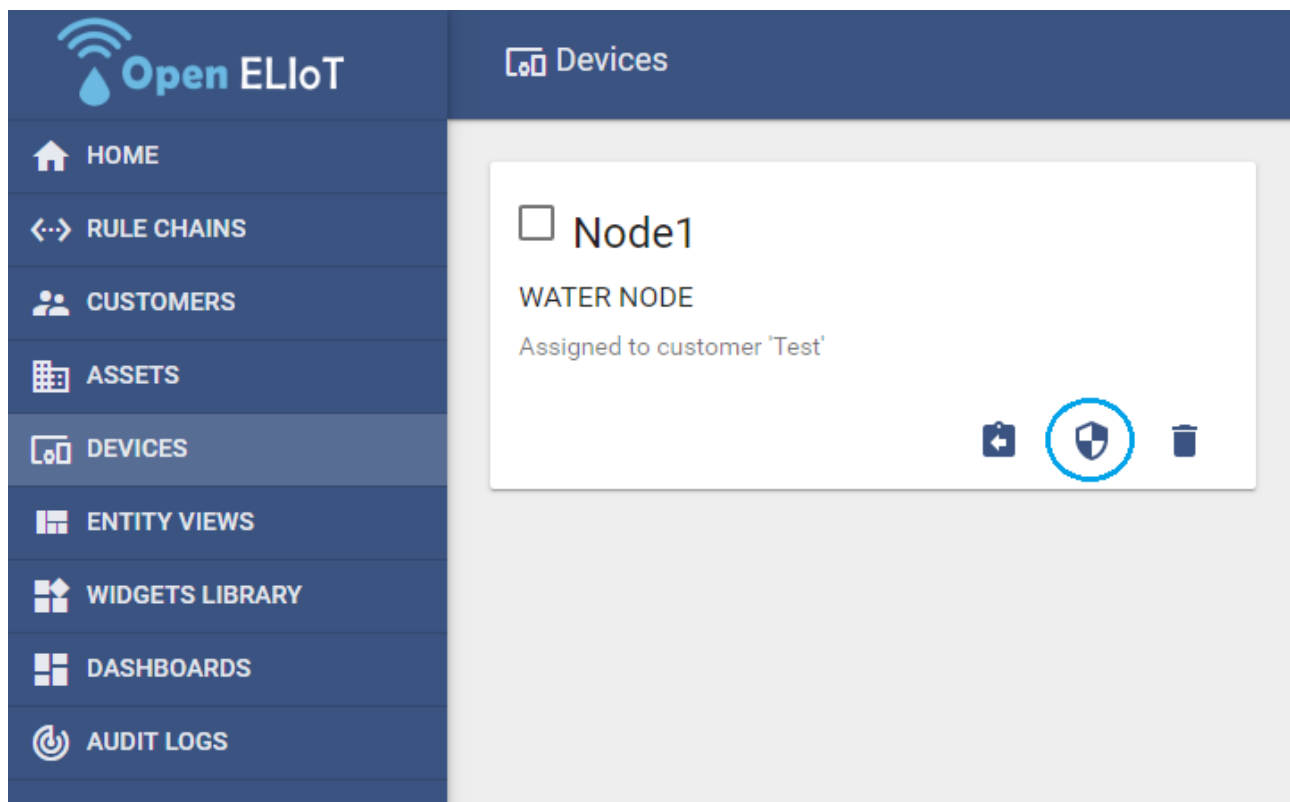
Ανοίγοντας την ενότητα “Devices” εμφανίζονται οι ήδη καταχωρημένες συσκευές. Η προσθήκη νέας συσκευής ξεκινάει πατώντας το εικονίδιο “+” στην κάτω δεξιά γωνία της οθόνης και συμπληρώνοντας τα παρακάτω πεδία:

- Name - Το όνομα της συσκευής για ευκολότερη αναγνώριση από τον χρήστη της πλατφορμας. Μπορεί να συμπληρωθεί με οτιδήποτε επιθυμεί ο χρήστης.
- Device type - Ό τύπος 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 κατά την δημιουργία επόμενης συσκευής.

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

The 'Add Device' form is a modal window with a dark blue header containing the title 'Add Device', a help icon, and a close icon. The form fields include: 'Name*' with the value 'Node1'; 'Device type*' with the value 'WATER NODE' and a clear button; 'Label' (empty); 'Is gateway' (checkbox, unchecked); and 'Description' (empty). At the bottom are 'ADD' and 'CANCEL' buttons.

Σχήμα 35: Παράθυρο δημιουργίας νέας συσκευής



Σχήμα 36: 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.

7.5. 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 όταν μπει στο πεδίο σε κανονική λειτουργία (έκδοση release).

Σε αυτό το αρχείο συμπληρώνονται τα στοιχεία για το σύνολο των nodes που θα χρησιμοποιηθούν και όχι για ένα μόνο 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_DESCRIPTOR** όπως φαίνεται στο 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_DESCRIPTOR[] =
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", "JUi8H7899yfoi0A8afa3"} // Node 3
16  };
17
```

Σχήμα 37: 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 κατά 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.

7.6. Εύρεση διεύθυνσης MAC ενός node

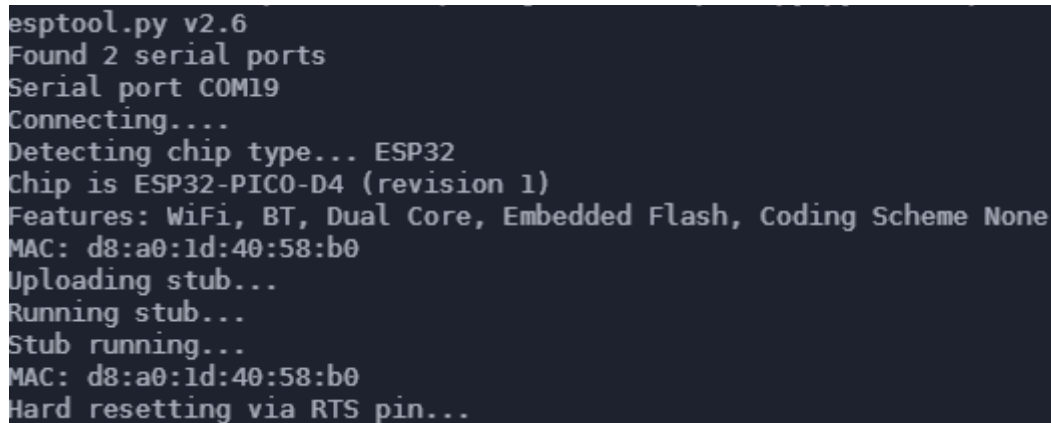
Για να μπορεί να συμπληρωθεί το παραπάνω αρχείο με τις διευθύνσεις MAC των nodes, θα πρέπει ο χρήστης να αντλήσει αυτή τη πληροφορία από κάθε node ξεχωριστά, αφού η διεύθυνση είναι μοναδική για το καθένα. Αυτό γίνεται προγραμματίζοντας το node αρχικά χωρίς να προστεθεί στο αρχείο αυτό, έτσι κατά την 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-PIC0-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 38: Έξοδος εργαλείου esp-tool

7.7. 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.

7.7. 1. 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 39: 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).

7.7. 2. 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

http://myfileserver.com/files/firmware_v101.bin

Firmware MD5

7435792a30bfaca4b561a882b85cfc19

Water sensor measure interval (mins)

10

Call home interval (mins)

30

OTA update

Reboot

Format SPIFFS

SAVE

Σχήμα 40: 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.

7.7.3. 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.

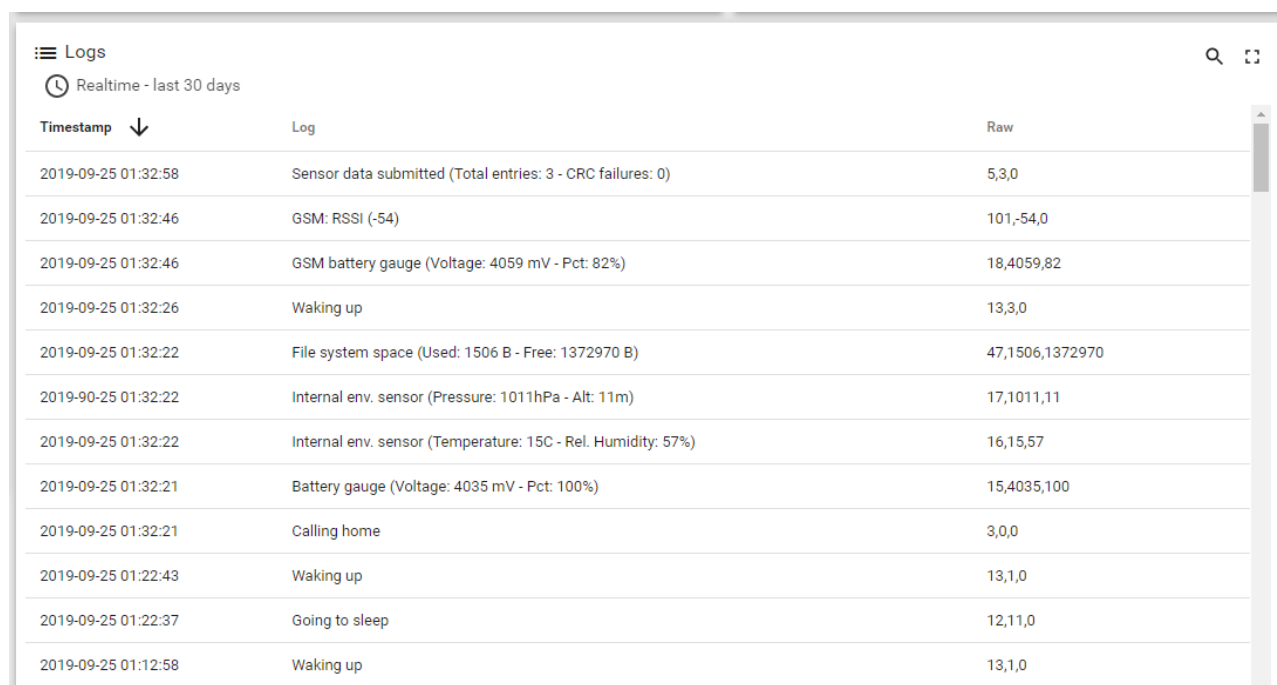
7.7.4. Έκδοση debug / release

Στο PlatformIO project έχουν οριστεί σεντ 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.

7.7.5. 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.



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 41: 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 ακεραιότητας (CRC check) εγγραφών πριν την αποστολή	Αριθμός εγγραφών που απέτυχαν τον έλεγχο (και συνεπώς δεν εστάλησαν)
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 απο 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

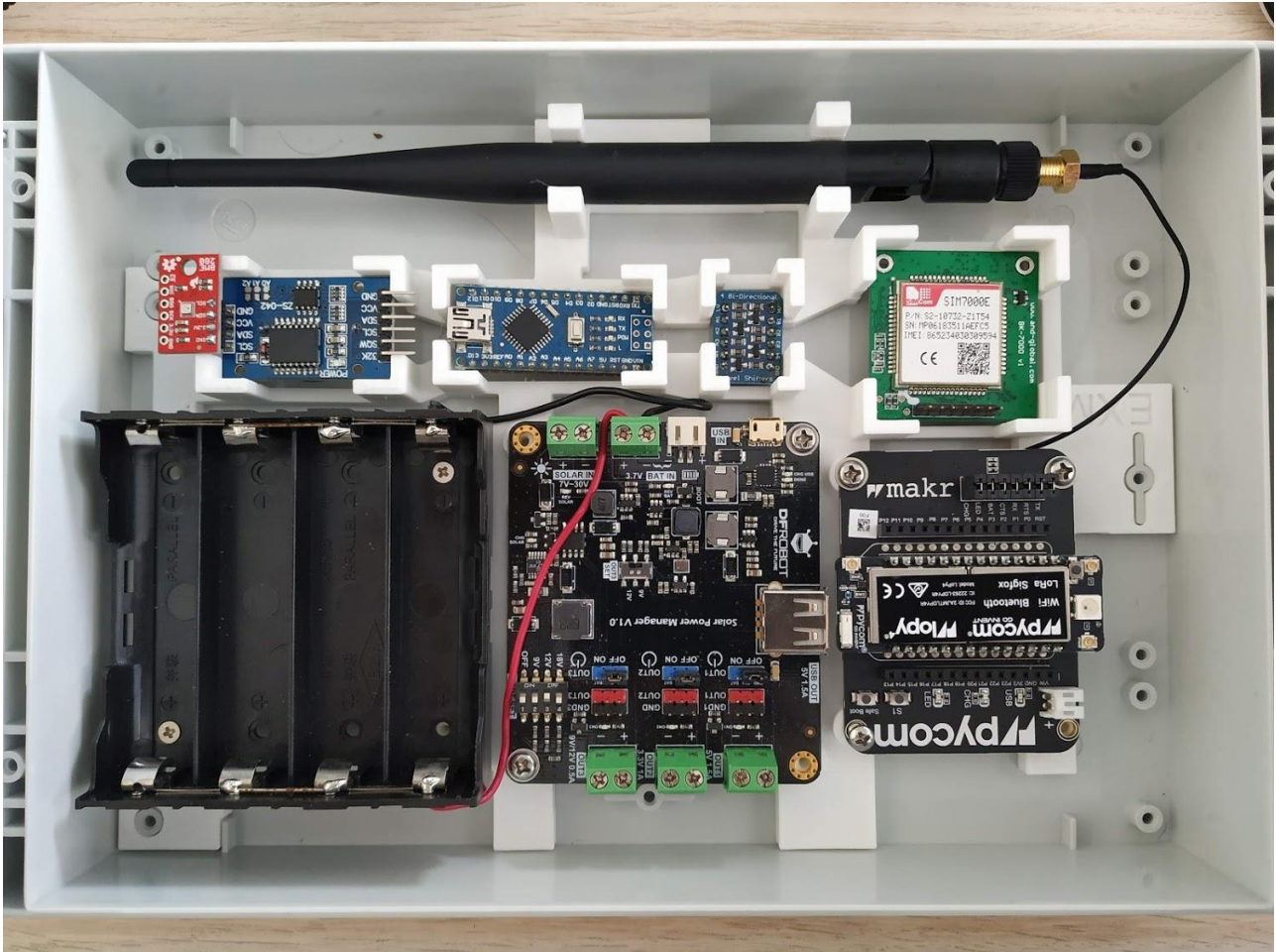
8. Προβλήματα, επόμενα βήματα

8.1. Σφάλματα στις επικοινωνίες GSM

Στην πορεία των δοκιμών του πρωτοτύπου κυκλώματος ELIoT Node v1 προέκυψε ότι υπήρχαν πολλά σφάλματα / θόρυβος στις UART επικοινωνίες μεταξύ ESP32 + SIM7000. Η χρήση ομοαξονικού καλωδίου δεν αποτέλεσε ικανοποιητική λύση, και μετά απο εκτεταμένες δοκιμές καταληξαμε στο συμπέρασμα ότι το πρόβλημα ήταν στο γεγονός ότι ο μικροελεγκτής ESP32 και το κύκλωμα που το υποστηρίζει δεν είναι

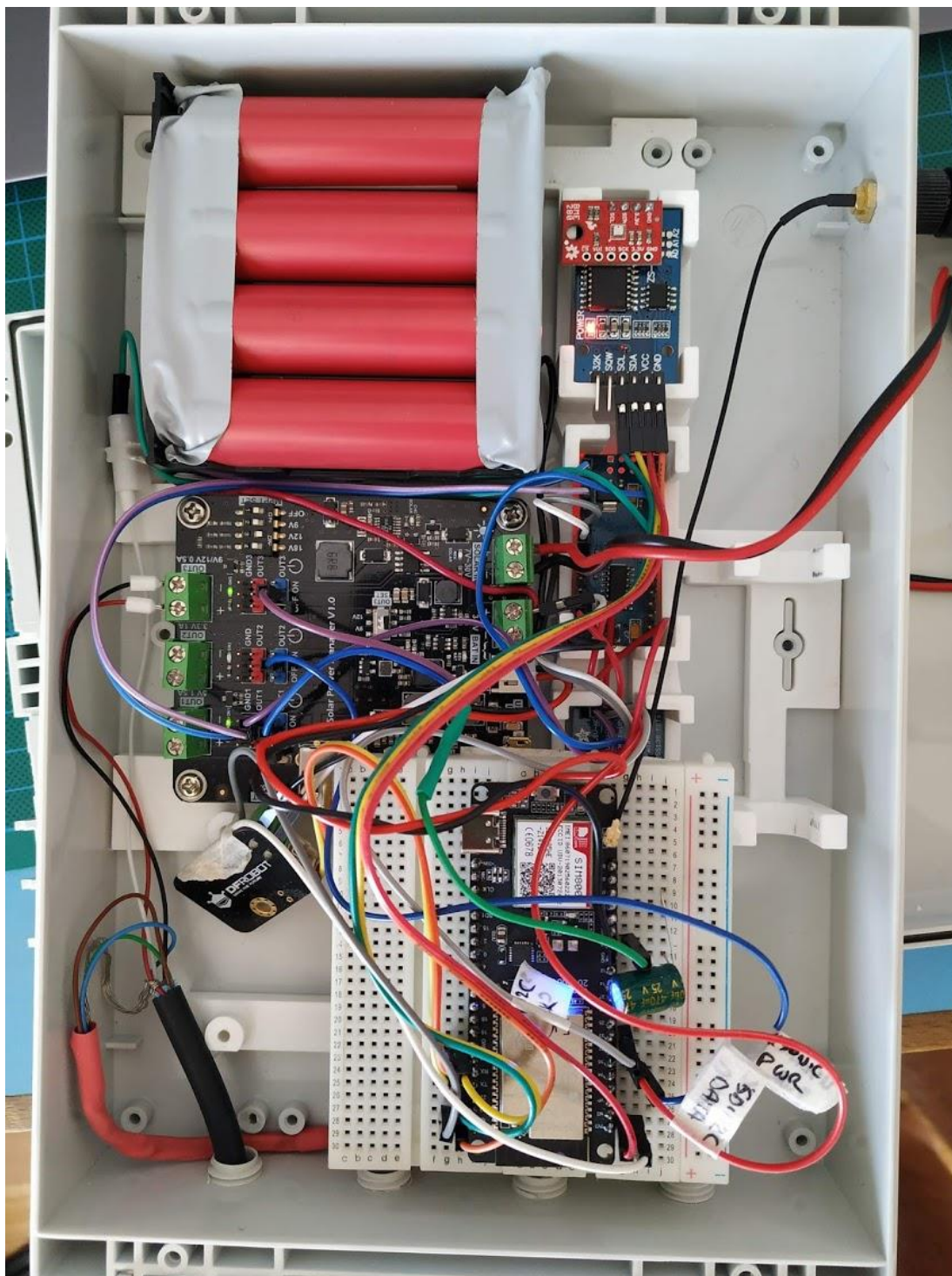
θωρακισμένα στις ηλεκτρομαγνητικές παρεμβολές, αρα δεχεται παρεμβολες απο το GSM module κατα την διαρκεια της λειτουργιας των GSM επικοινωνιων.

Αυτο το σημαντικό πρόβλημα καθιστά ακαταλλήλο το TFOX για GSM επικοινωνίες και για αυτό έγιναν διαφορές δοκιμές με διαφορετικές κεντρικές πλακέτες ESP32 που είχαν το chip θωρακισμένο, όπως το PYCOM LOPY4.



Σχήμα 42. 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 αντόνομα και ασταμάτητα για ~1 μήνα, οπότε αποφασίστηκε η έκδοση v2 του node να βασισκει σε αυτό. (βλεπε D2.4)



Σχήμα 43. ELIoT Node v1.2 TTGO TCALL SIM800

Παράρτημα προδιαγραφών εξαρτημάτων (datasheets)

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