



LoRa2HA

Documentation version 1.4

Introduction

LoRa2HA est un ensemble d'équipements permettant une interconnexion entre des capteurs/actionneurs éloignés et un serveur Home Assistant (via MQTT), par radio en utilisant des modules LoRa®; la portée peut aller jusqu'à plus de 500m selon la qualité des antennes.

Le HUB (basé sur ESP32), reçoit les données radio et les envoi à Home Assistant (H.A) au travers d'un serveur MQTT.

Le terme de HUB sera utilisé, et non Gateway, afin de ne pas créer de confusion avec les Gateway « LoRaWan ».

Le projet¹ contient les sources de tous les firmwares, les schémas des cartes ainsi que les « Gerber » afin de pouvoir fabriquer les PCB.

1 Ce projet est destiné à la communauté des « makers » et est fourni à titre informatif et éducatif. Il est fourni "tel quel", sans garantie d'aucune sorte, expresse ou implicite, y compris, mais sans s'y limiter, notamment concernant son fonctionnement, sa sécurité ou son adéquation à un usage particulier. L'utilisation de ce projet se fait à vos propres risques. L'auteur et les contributeurs ne pourront être tenus responsables des conséquences de l'utilisation de ce projet, des dommages directs, indirects, accessoires, ou consécutifs pouvant résulter de l'utilisation ou de l'incapacité d'utiliser le projet. L'assemblage, la modification ou l'utilisation des matériels et logiciels associés à ce projet impliquent des risques. Vous êtes seul responsable de l'utilisation que vous en faites, y compris des éventuels dommages matériels, blessures, ou autres incidents pouvant survenir. Important : Avant d'utiliser ce projet dans un environnement de production ou dans un cadre professionnel, veuillez vous assurer qu'il est conforme à vos exigences et aux réglementations en vigueur. Attention : Manipuler des composants électroniques, des outils, ou des matériaux peut être dangereux. Prenez toutes les précautions nécessaires, notamment l'utilisation d'équipements de protection, et suivez les réglementations locales en matière de sécurité.

Sommaire

Introduction.....	1
1 Les équipements.....	3
1.1 Les HUB.....	3
1.2 Les modules « capteurs/actionneurs ».....	4
2 Fonctionnement.....	5
3 Mise en route.....	6
3.1 Installation du HUB.....	6
3.1.1 Sélecteur WMODE.....	6
3.1.2 Wifi AP.....	6
3.1.3 Configuration réseau.....	7
3.1.4 Utilitaires.....	8
3.1.5 Mise à jour du Firmware.....	8
3.1.6 Mise à jour de fichiers.....	8
3.2 Installation des modules capteurs/actionneurs.....	8
3.2.1 Multiples modules identiques.....	9
3.2.2 Paramétrage des capteurs.....	10
3.2.3 Module MLD01 Linky.....	11
3.2.4 Module MLD02 Linky (Autonome).....	11
3.2.5 Module MLD3 (Piscine).....	11
3.2.6 Module MLE21 (2 in / 1 out).....	12
3.2.7 Module MLE31 (3 in / 1 out).....	12
3.2.8 Module MLE42 (4 in / 2 out).....	12
3.2.9 Module MLA30 (3 in / 0 out).....	12
3.3 Contrôle dans H.A.....	13
4 Montage des HUB.....	15
4.1 MLH1.....	15
4.2 MLH2.....	16
4.3 MLH3.....	16
4.4 MLH4.....	17
4.5 MLH5.....	18
5 Programmation.....	19
5.1 Prérequis.....	19
5.2 Programmer le HUB.....	19
5.2.1 Programmation simple de MLH1.....	19
5.2.2 Programmation simple de MLH2.....	19
5.2.3 Programmation simple de MLH3.....	19
5.2.4 Programmation simple de MLH4.....	20
5.2.5 Programmation simple de MLH5.....	20
5.2.6 Programmation avec l'IDE Arduino.....	20
5.3 Programmation UPDI.....	21
6 ANNEXE A : Nomenclatures.....	22
6.1 MLH1.....	22
6.2 MLH2.....	23
6.3 MLH3.....	24
6.4 MLH4.....	25
6.5 MLH5.....	25

1 Les équipements

La référence de chaque module adopte la terminologie suivante :

- **ML** : Module LoRa
- **H** : Hub, **D** : Dédié, **E** : alimentation Externe, **A** : alimentation Autonome
- un nombre indiquant un ordre de création (Ex : Hub et Dédié) ou un nombre d'entrées sorties (Ex 31 = 3 entrées / 1 sortie), éventuellement suivi d'un ordre de création ou d'une lettre (fonctionnalité particulière).

La nouvelle génération de modules est basée sur le microprocesseur ATtiny3216, une carte « fille » **MLT3216** équipe la majorité des modules de terrains, étant extractible elle permet une mise à jour firmware plus facilement.

1.1 Les HUB

Il existe 5 types de HUB selon le besoin, tous équipés d'un module LoRa :

- MLH1² : ESP32-S2 Mini, en wifi
- MLH2 : Olimex-POE-ISO, en Ethernet avec POE
- MLH3 : WT32-ETH01, en Ethernet
- MLH4 : ESP32-D1-Mini, en wifi
- MLH5 : TTGO_LoRa32 (pcb T3_v1.6)

Dans le cas du MLH3, il est tout à fait possible de le rendre « POE » en le connectant derrière un répartiteur POE de ce type, type USB-C/5v (coût inférieur à 10€) :



Le HUB dispose de son propre serveur web permettant le paramétrage de la connexion à H.A ainsi que l'identification des tous les modules sur le terrain.

Estimation de difficulté/prix

Modèle	Wifi	Ethernet	POE	DIY	Coût
MLH1	✓			★☆☆	★☆☆
MLH2	✓	✓	✓	★☆☆	★★★★
MLH3	✓	✓	indirect	★★★	★★★
MLH4	✓			★☆☆	★☆☆
MLH5	✓			★☆☆	★★★

DIY : difficulté de fabrication

Coût : <10€, <15€, >15€

² Le HUB MLH1 ne permet pas le téléchargement des firmware par UPDI

1.2 Les modules « capteurs/actionneurs »

Il existe plusieurs types de modules :

- **MLD01** : Relevé TIC Linky (alimenté par USB externe)
- **MLD02** : Relevé TIC Linky (alimenté par Linky) (**projet en cours**)
- **MLD03** : Gestion de piscine (hors chimie)
- **MLA30** : 3 entrées, autonome, très basse consommation, avec batterie et panneau solaire
- **MLE21** : 2 entrées / 1 sortie, alimenté en 8-28vdc (*remplacé par MLE31*)
- **MLE31** : 3 entrées / 1 relai, alimenté en 8-28vdc
- **MLE42** : 4 entrées / 2 relais ou 1 moteur (2 sens de rotation), alimenté en 8-28vdc

Chaque module possède un Firmware proposé par défaut mais il est possible d'adapter ceux-ci pour en modifier les fonctionnalités et/ou l'adapter aux capteurs connectés.

Estimation de difficulté/prix

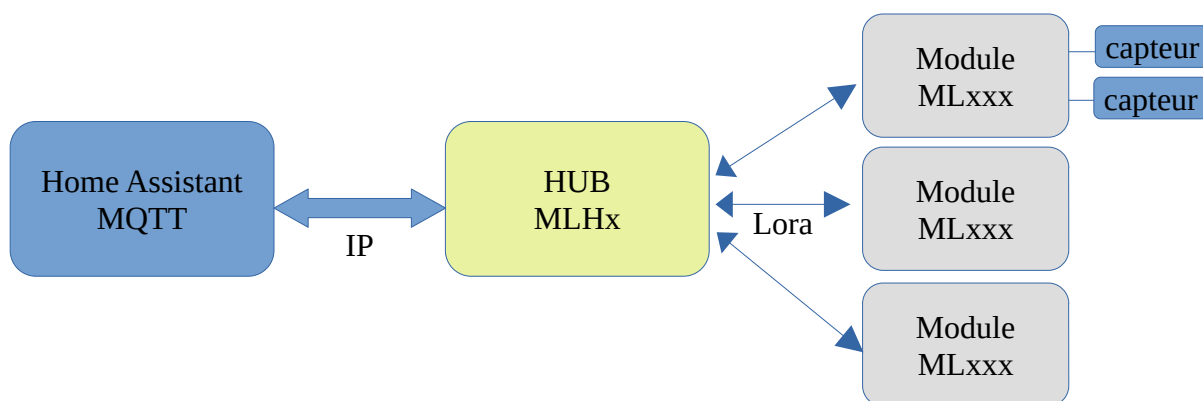
Modèle	DIY	Coût module	Coût externe ³
MLD01	★☆☆	★☆☆	☆☆☆
MLD03	★★☆	★★★	★★☆
MLA30	★★★	★★☆	★★☆
MLE31	★★☆	★★★	★☆☆
MLE42	★★☆	★★★	★★☆

DIY : difficulté de fabrication

Coût : <10€, <15€, >15€

³ Le coût externe est une indication du coût des différents capteurs/actionneurs les plus courants dans les applications pour lesquelles les modules sont utilisés.

2 Fonctionnement



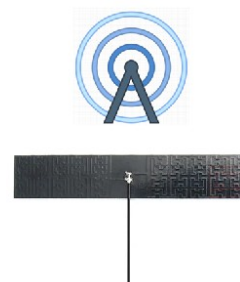
Chaque module (MLxxx) transmet les informations des capteurs (par LoRa) vers le HUB, celui-ci est connecté à H.A (via MQTT) par IP, et redirige les informations.

ATTENTION : le choix des antennes et leur position sont particulièrement importants.

Il est conseillé d'installer une antenne à gain important sur le HUB (> 3 dBi).

L'antenne à « ressort » des modules RA-01, ainsi que les antennes « bâton » (généralement avec connecteur SMA) doivent être utilisées à la verticale.

Les antennes IPX sur PCB, comme ci-contre (jonction au milieu), doivent être utilisées à la verticale et perpendiculaire à l'azimut du HUB.



Exemple de choix d'antenne que vous pouvez installer sur le HUB, avec un convertisseur IPX/SMA, sur un module RA-02.



3 Mise en route

Cela commence par l'installation et la configuration du HUB.

H.A doit avoir un serveur MQTT en état de fonctionnement.

3.1 Installation du HUB

➔ Pour la programmation du HUB, voir le chapitre « Programmation »

Le HUB une fois programmé doit être raccordé au réseau IP, identique à celui de H.A.

3.1.1 Sélecteur WMODE

Le sélecteur « WMODE » permet de choisir le mode de fonctionnement du réseau IP : Wifi STA, Wifi AP ou Ethernet.

- Wifi STA (Station) démarre le HUB en mode Wifi « classique », même si ce wifi n'est pas disponible, il restera alors en « défaut » (l'ancienne version passait automatiquement en mode AP, cela générerait des problèmes lors de coupure générale du secteur, quand le routeur wifi était long à démarrer, le HUB ne le trouvait pas et passait en mode AP),
- Wifi AP force le HUB à rester en mode AP (Access Point), même si les données wifi ont été renseignées,
- Ethernet (seulement sur MLH2 et MLH3) force le HUB à fonctionner en Ethernet, le Wifi n'est pas utilisé, câble connecté ou non.

➔ Pour les HUB wifi :

- si aucun paramétrage n'a jamais été effectué, il passe automatiquement en mode « Point d'accès » (AP).
- si un autre réseau wifi a déjà été paramétré et que vous souhaitez en changer parce qu'il n'est pas accessible, positionnez le sélecteur « WMODE » en position **AP**.

3.1.2 Wifi AP

Cherchez un réseau wifi nommé « lora2ha0 » et connectez-vous ; le mot de passe est : 12345678.

En mode AP, le HUB est à l'adresse IP 192.168.4.1 ou l'URL lora2ha0.local ; la page ci-dessous doit s'afficher.

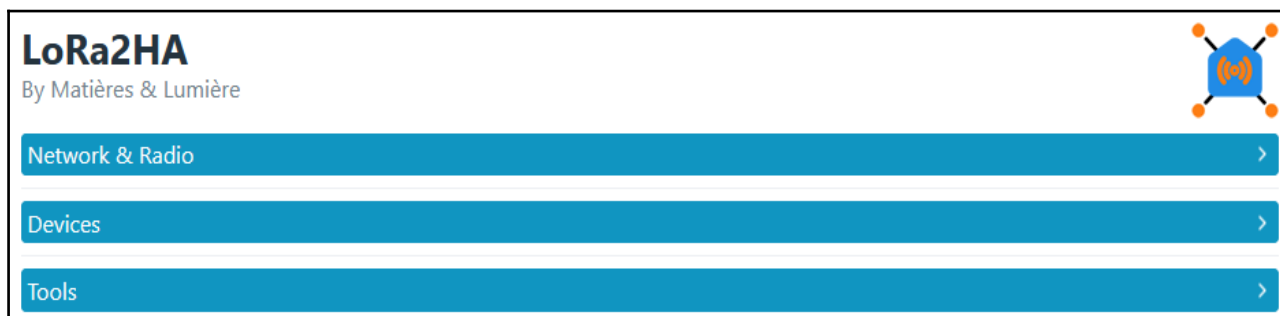
Avec un smartphone, scannez les QRcode suivants :

Connexion au wifi AP du HUB



Accès à la page web

❗ Si la page ne s'affiche pas, il est probable que le dossier « data », contenant le serveur web, n'ai pas été téléchargé correctement dans l'ESP32.



LoRa2HA
By Matières & Lumière

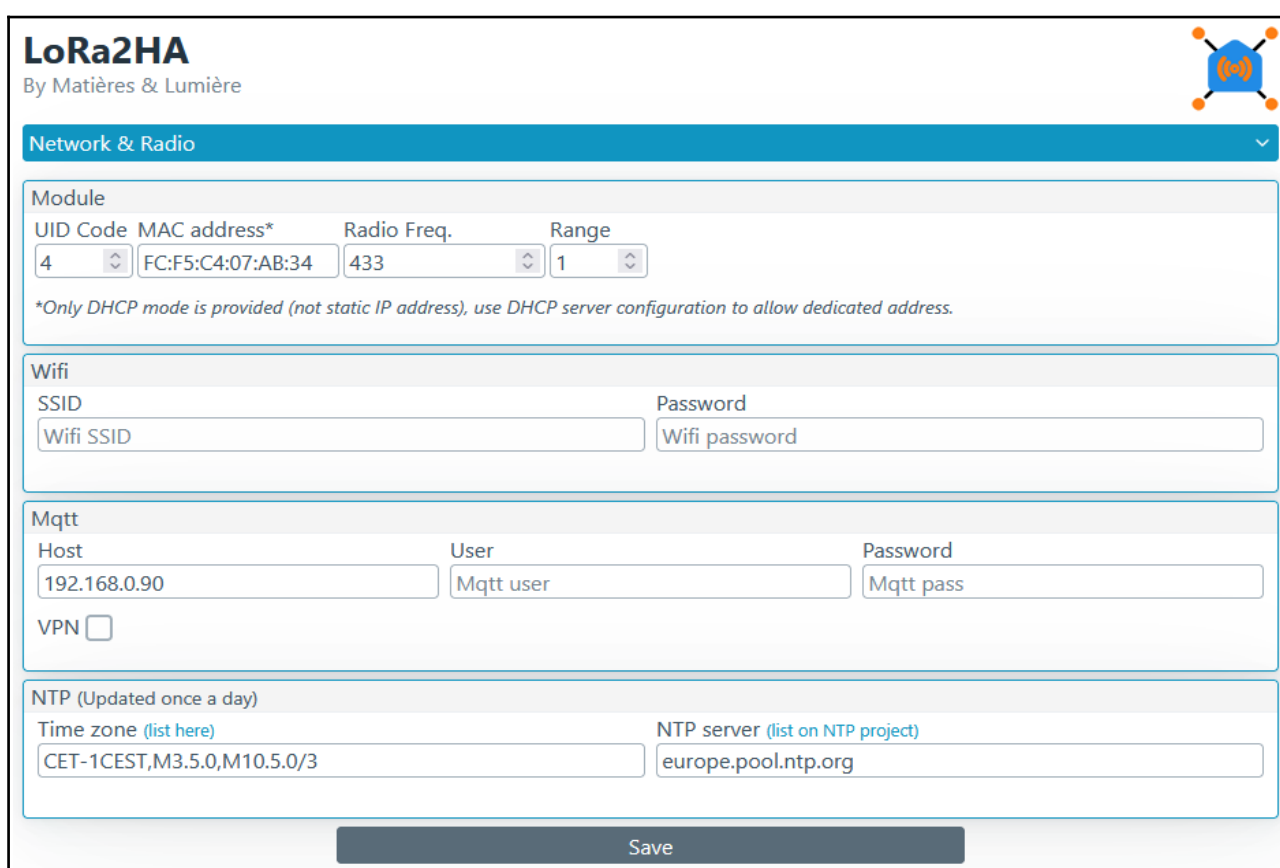
Network & Radio >

Devices >

Tools >

3.1.3 Configuration réseau

Ouvrez la zone « Network & Radio », renseignez les données demandées puis enregistrez.



LoRa2HA
By Matières & Lumière

Network & Radio

Module

UID Code	MAC address*	Radio Freq.	Range
4	FC:F5:C4:07:AB:34	433	1

*Only DHCP mode is provided (not static IP address), use DHCP server configuration to allow dedicated address.

Wifi

SSID	Password
Wifi SSID	Wifi password

Mqtt

Host	User	Password
192.168.0.90	Mqtt user	Mqtt pass

VPN ☐

NTP (Updated once a day)

Time zone (list here)	NTP server (list on NTP project)
CET-1CEST,M3.5.0,M10.5.0/3	europe.pool.ntp.org

Save

L'information de l'adresse MAC vous permet de pouvoir gérer votre serveur DHCP et affecter une adresse IP spécifique, le HUB ne fonctionne QUE avec un serveur DHCP, il n'est pas possible de lui attribuer une IP fixe.

Si vous envisagez de mettre plusieurs HUB sur le réseau, il faut indiquer un code unique à chacun dans « UID Code » (sinon laissez 0) ; il est conseillé de mettre « 1 » au premier HUB que vous paramétrez et laisser « 0 » pour le suivant.

Selon le modem LoRa qui est utilisé, vous devez indiquer la fréquence 433 ou 868 (MHz).

Dans « Mqtt host » indiquez uniquement l'adresse IP du serveur Mqtt (ou celle de H.A si Mqtt y est intégré en Add-on).

👉 Le port par défaut est le 1883 (pas de SSL). Une prochaine version permettra d'utiliser SSL.

Le configuration NTP sert au HUB pour « publier » l'heure vers les différents modules qui en ont besoin.

Une fois sauvegardé, il faut redémarrer le HUB, physiquement ou en cliquant sur le bouton « Reset » dans la zone « Tools », avant cela, positionnez le sélecteur « WMODE » sur **STA**.

En « production » l'adresse du HUB est « lora2ha0.local », ou l'adresse IP fournie par votre serveur DHCP.

Le HUB est prêt à recevoir les informations radio des « modules » et ainsi pouvoir les paramétrer.

3.1.4 Utilitaires

La zone « Tools » permet d'effectuer la maintenance du HUB ainsi que des mises à jours de fichiers, ou sauvegarder la configuration.

3.1.5 Mise à jour du Firmware

Le firmware peut être mis à jour par OTA (Other The Air), cliquant sur le bouton « Parcourir » de la zone « Update firmware », de sélectionner le fichier contenant le Firmware (avec un extension .bin), le nom contient en général le type d'ESP32 qui est utilisé, et enfin cliquer sur le bouton « Update ».

Le téléchargement prend entre 5 et 10 secondes, l'ESP32 redémarre et la page est web rechargée.

👉 Avant toute mise à jour du firmware, effectuez une sauvegarde de la configuration en cliquant sur le bouton « Download configuration file ».

3.1.6 Mise à jour de fichiers

Tous les fichiers du serveur web peuvent être mis à jour en cliquant sur le bouton « Parcourir » de la zone « Send file », de sélectionner le fichier dans le dossier « data », et enfin de cliquer sur le bouton « Send ».

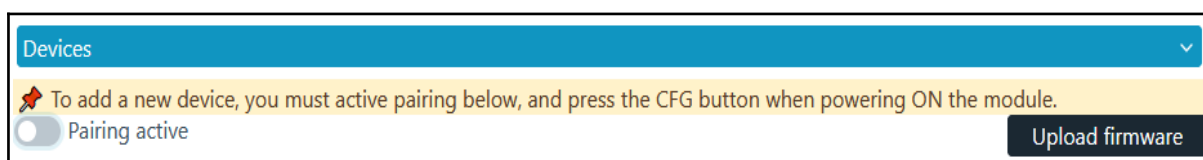
L'ESP32 ne redémarre pas mais la page web est rechargée.

Si vous avez envoyé un nouveau fichier de configuration, il faut redémarrer l'ESP32 pour qu'il soit pris en compte.

3.2 Installation des modules capteurs/actionneurs

➔ Pour la programmation des modules, voir le chapitre « Programmation »

Sur la page web du HUB, ouvrez la zone « Devices », si aucun module n'a encore été appairé, l'indication suivante est affichée.



Il faut mettre le HUB en mode « Pairing » (appairage) en activant le sélecteur « Pairing active », un bandeau orange apparaît.

Sur le module, appuyez sur le bouton « CFG » et mettez le sous tension. Si une led est présente, elle clignote tant que vous tenez le bouton appuyé. Relâchez le bouton pour passer le module en mode « Pairing ».



⚠ Attention, si vous tenez le bouton « CFG » appuyé plus de **10 secondes**, le module fera une remise à zéro complète de sa mémoire (Factory setting).

Chaque module doit avoir une adresse unique (définie par le firmware), lorsqu'il y a plusieurs modules identiques, il faut modifier l'adressage de ceux-ci :

1. affectez un nouveau numéro
2. cliquez sur le bouton « Change Nr »

Pour libérer le module de son mode « Pairing », vous devez cliquer sur le bouton « Finish pairing ».

➔ l'action « Finish pairing » n'a aucune incidence sur le HUB, elle libère simplement le module.

⚠ L'image ci-dessus correspond au firmware « mla30-mboxS », avec des contacts qui envoient leur état binaire (Binary sensor / State) ; dans le cas du firmware « mla30-mboxT » c'est un événement qui est généré (Trigger) et il n'est seulement visible **que** dans les automatisations.

3.2.1 Multiples modules identiques


Lorsque vous avez plusieurs modules identiques, par exemple 2 modules MLA30, ils doivent avoir les numéros 30 et 31, commencez toujours par le numéro le plus haut :

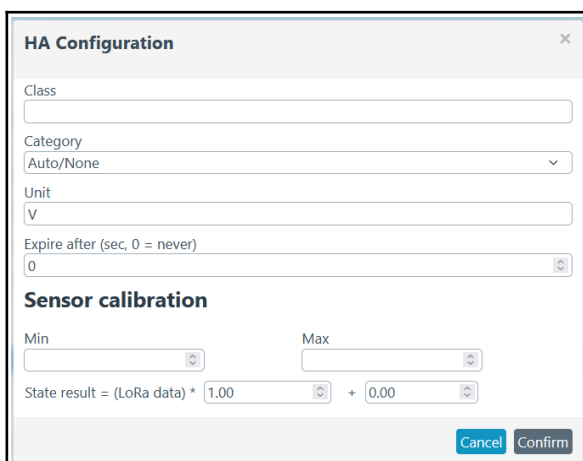
1. appairez le premier module

2. il affiche son numéro par défaut : 30
3. changez son adresse par 31
4. terminez le « pairing »
5. sauvegardez la configuration
6. appairez le deuxième module
7. il affiche son numéro par défaut : 30
8. gardez ce numéro
9. sauvegardez la configuration

3.2.2 Paramétrage des capteurs

- « **ID** » est le n° du capteur/actionneur, il est défini par le firmware de chaque module
- « **Label** » est le nom du capteur/actionneur qui apparaîtra dans H.A
- « **Data** » est le type de donnée transmise par le module, il est défini par le firmware

Cliquez ensuite sur le bouton  pour ouvrir la boîte de dialogue permettant la configuration du capteur/actionneur pour H.A.




- « **Class** » selon le type de capteur (sensor, binary_sensor, etc), il est possible d'avoir des classes différentes, vous trouvez les informations nécessaires sur la page web du site de Home Assistant : <https://www.home-assistant.io/docs/configuration/customizing-devices/#device-class>
- « **Category** » permet de classer le capteur en tant que **Mesure**, **Diagnostic** ou **Configuration**.
- « **Unit** » permet d'indiquer l'unité correspondant à la mesure du capteur
- « **Expire after** » est la durée pendant laquelle la mesure est valide, au-delà elle sera « indisponible » au sens H.A.
- « **Min** » et « **Max** » permettent de borner les mesures reçues. Il peut arriver qu'il y ai des erreurs de mesure (ou autre) qui aboutissent à une donnée qui soit hors de sa plage normale, par exemple une consommation électrique de plus de 100 KW, si vous avez un abonnement 9KWh, vous pouvez entrer 10 000 dans max, par exemple.
- « **State result** » permet d'appliquer des coefficients correcteur à une mesure brute ($Y = A \cdot X + B$), par exemple si une température est 1° en dessus de la température normale, entrez -1 dans le deuxième paramètre (B).

Dans la zone « Class », commencez à taper le début du texte si vous connaissez la classe à attribuer, ou faite 2 fois 1 clic pour afficher la liste complète.

Après avoir complété les différents paramètres, sauvegardez le paramétrage en cliquant sur « Confirm ».

Après avoir tout configuré, cliquez sur le bouton « Save ».

Il est possible de supprimer entièrement un module en cliquant sur le bouton .

3.2.3 Module MLD01 Linky

Le module est à l'adresse « 10 » par défaut (elle ne peut pas être changée, sauf à modifier le firmware et reprogrammer le module).

Le code source proposé, fonctionne en mode « Historique » par défaut, avec très peu de modification il est possible de le faire fonctionner en mode « standard ».

Le module ayant une forte consommation, il n'est pas possible d'utiliser l'alimentation fournie par le connecteur TIC du Linky (borne A), il devra donc avoir sa propre alimentation externe, type chargeur USB.

Connectez le module sur le connecteur TIC du linky : bornes I1 et I2.

Dans le Firmware par défaut (lora2ha-MLD01-Linky-ProMicro) vous disposez des informations ci-dessous :

ID	Label linky	Donnée
1	OPTARIF	BASE, HPHC/HCHP
2	BASE	Compteur (KWh)
3	HPHC	Compteur HC (KWh)
4	HCHP	Compteur HP (KWh)
5	PAPP	Puissance apparente (VA)
6	INST (monophasé) INST1 (triphasé)	Intensité instantanée 1 (A)
7	INST2 (triphasé)	Intensité instantanée 2 (A)
8	INST3 (triphasé)	Intensité instantanée 3 (A)

3.2.4 Module MLD02 Linky (Autonome)

Ce module n'est pas encore validé.

3.2.5 Module MLD3 (Piscine)

Le module « Piscine » est plus particulièrement adapté aux piscines hors-sol, il peut être positionné à l'endroit où est branché la pompe du filtre, dans un boîtier équipé de rail DIN.

La documentation détaillée se trouve dans le document [LoRa2HA-MLD03-readme.pdf](#).

3.2.6 Module MLE21 (2 in / 1 out)

Ce module est remplacé par le module MLE31.

3.2.7 Module MLE31 (3 in / 1 out)

Ce module est principalement destiné à la gestion d'un portail, mais sa polyvalence d'entrées/sortie permet de l'utiliser dans beaucoup d'autres projets.

La documentation détaillée se trouve dans le document [LoRa2HA-MLE31-readme.pdf](#).

3.2.8 Module MLE42 (4 in / 2 out)

Ce module est principalement destiné à la gestion d'une porte de garage ou d'un poulailler. Les 2 sorties peuvent être 2 relais **OU** 1 moteur (bidirectionnel), mais sa polyvalence d'entrées/sortie permet de l'utiliser dans beaucoup d'autres projets.

La documentation détaillée se trouve dans le document [LoRa2HA-MLE42-readme.pdf](#).

3.2.9 Module MLA30 (3 in / 0 out)

Ce module est principalement destiné à la gestion d'une boîte à lettre.

Étant autonome pour son alimentation, il peut être utilisé pour tous les projets de mesure à faible fréquence.

Sa consommation en veille est d'environ 5µA, soit une durée de vie de plus de 20 ans (sur le papier) ; cette durée est altérée par la consommation lors de son réveil, par le type de capteur qui lui est raccordé, et la fréquence à laquelle il est réveillé. La transmission LoRa consomme environ 60mA durant 100ms, par donnée à transférer.

La documentation détaillée se trouve dans le document [LoRa2HA-MLA30-readme.pdf](#).

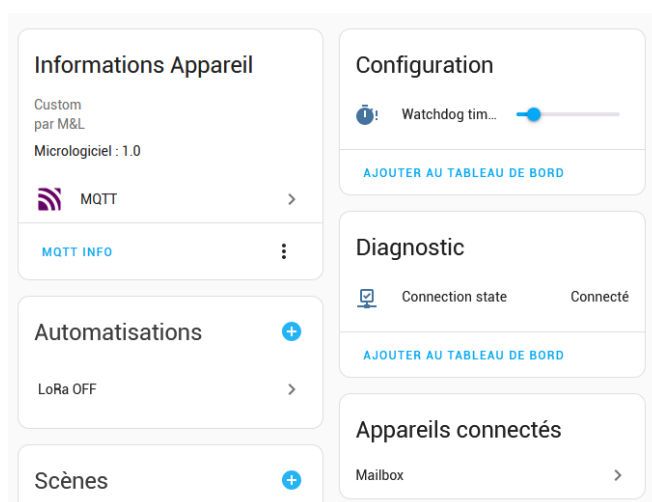
3.3 Contrôle dans H.A.

Dans H.A, cliquez sur « Paramètres » puis « Appareils et Services ».

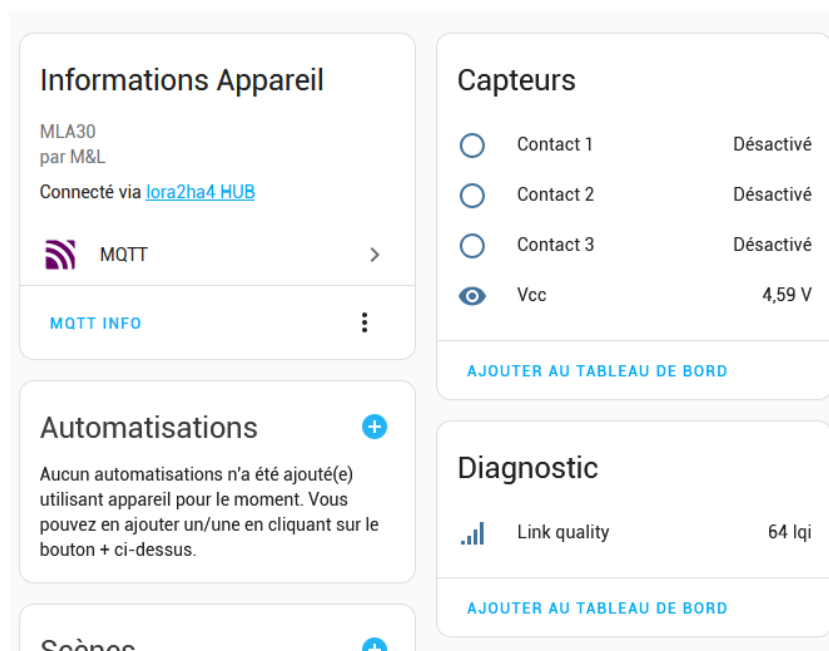
Dans le bloc « MQTT » cliquez sur le nombre d'appareils pour afficher la liste des appareils connus.

Le premier appareil trouvé concerne le HUB, « lora2ha0 HUB », il propose 2 entités :

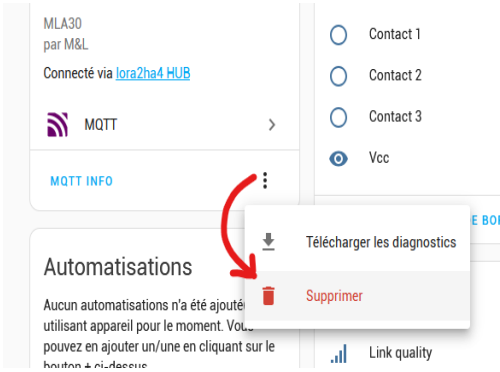
- **Watchdog timer**, période à laquelle le HUB envoie une information de « vie » à H.A
- **Connection state**, information de connexion du HUB.



Il propose la liste des tous les appareils qui lui sont connectés. En choisissant « Mailbox » vous arrivez sur cette page :



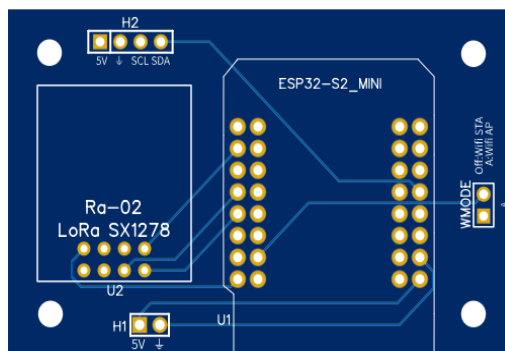
Si vous voyez apparaître des anciens capteurs ou contrôles (après changement de nom ou de type par exemple), il faut supprimer l'appareil en totalité. Si besoin un nouvel appareil sera re-crée lors du prochain démarrage du HUB.



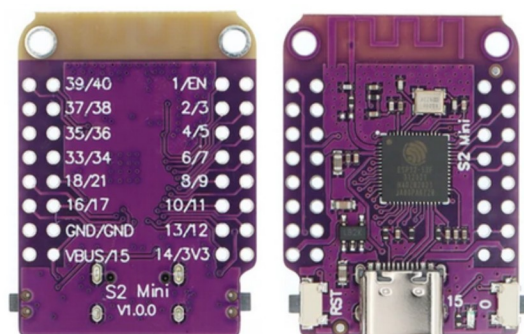
4 Montage des HUB

La liste complète des différents composants est fournie en Annexe A.

4.1 MLH1

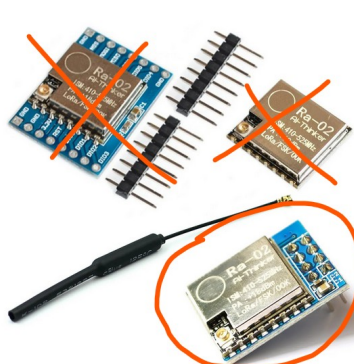


La carte MLH1 doit être équipée d'un ESP32-S2 Mini.



Le module LoRa conseillé est en 433MHz, RA-02 (connecteur IPX), en choisissant une antenne adaptée (> 3 Dbi).

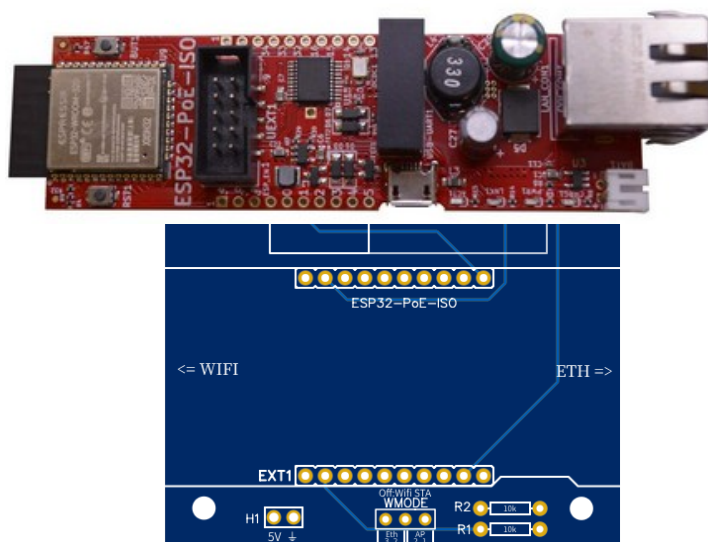
Attention, il existe plusieurs modèles de modules LoRa, il faut celui avec le connecteur DIP 2x4 pins.



L'alimentation de la carte se fait au travers du connecteur USB de l'ESP32 ou par le connecteur H1 en 5V.

4.2 MLH2

La carte MLH2 doit être équipée d'un ESP32-POE-ISO, Wifi à gauche, Ethernet à droite.

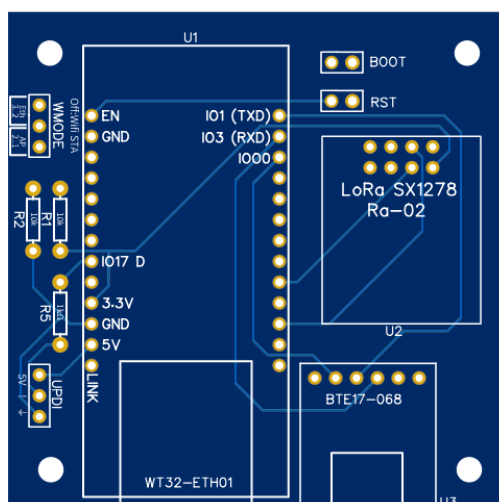


Il y a la possibilité d'utiliser la version « EA » avec une antenne IPX

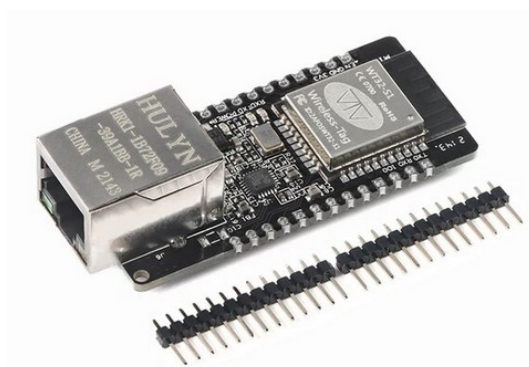
Le connecteur H1 peut être utilisé pour l'alimentation (5V) si le POE n'est pas disponible.

Le module LoRa est identique à la carte MLH1.

4.3 MLH3



La carte MLH3 doit être équipée d'un WT32-ETH01.



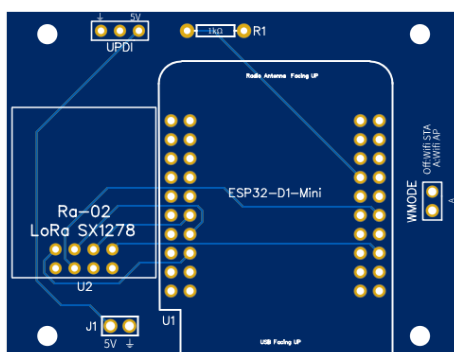
Le module LoRa est identique à la carte MLH1.

L'alimentation se fait au travers d'une mini carte USB (BTE17-06B), cette carte est un convertisseur USB /TTL CH340E, permettant la programmation du module.

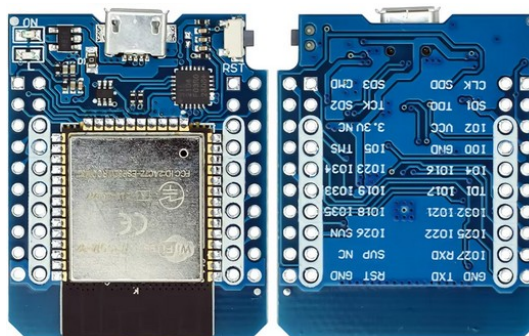


Pour passer le module en mode « programmation », il faut relier le Jumper « BOOT » et faire un « Reset » avec le Jumper « RST ».

4.4 MLH4



La carte MLH4 doit être équipée d'un ESP32-D1-Mini.



Le module LoRa est identique à la carte MLH1.

Le connecteur J1 peut être utilisé pour l'alimentation (5V).

4.5 MLH5



Il n'y a pas de PCB pour ce HUB, la carte inclue tous les modules.

WMode peut être géré en connectant un interrupteur sur l'entrée IO4.

5 Programmation

Ce chapitre sous-entend que vous maîtrisez la programmation d'un Arduino.

Plusieurs tutos traitent de ce sujet, cette documentation n'a pas vocation à entrer dans le détail de la programmation d'un Arduino.

5.1 Prérequis

La librairie « RadioLink » doit être installée en téléchargeant le fichier zip sur le GitHub « PM04290 » (<https://github.com/PM04290/RadioLink>) et en le décompressant dans le dossiers des librairies Arduino ;

Les différentes librairies utiles à la compilation sont détaillées dans le code source de chaque projet.

5.2 Programmer le HUB

Il existe une manière simple de programmer le HUB si vous ne souhaitez pas recompiler tout le programme ; il suffit d'utiliser les fichiers de commandes (.bat) du dossier « tools ». Ce dossier embarque tous les exécutables permettant d'effectuer la programmation sans IDE Arduino.

Il faut utiliser un câble USB équipé des fils de « data », pas simplement dédiée à la charge des appareils.

5.2.1 Programmation simple de MLH1

1. Appuyez sur le bouton « BOOT »
2. Mettez la carte sous tension en raccordant la prise USB
3. Relâchez le bouton « BOOT »
4. Dans le dossier « tools », exécutez le fichier de commande commande « MLH1_LOLIN-S2-Mini.bat »
5. Sélectionnez le port « COM » correspondant au connecteur USB
6. Le téléchargement se passe en 2 étapes : fichiers du serveur web, puis firmware
7. En fin de téléchargement, l'ESP32 ne fera pas de Reset, Faites un contact bref sur le bouton « RST ».

5.2.2 Programmation simple de MLH2

1. Appuyez sur le bouton « BUT1 »
2. Mettez la carte sous tension en raccordant la prise USB
3. Relâchez le bouton « BUT1 »
4. Dans le dossier « tools », exécutez le fichier de commande commande « MLH2_Olimex-POE_ISO.bat »
5. Sélectionnez le port « COM » correspondant au connecteur USB
6. Le téléchargement se passe en 2 étapes : fichiers du serveur web, puis firmware
7. En fin de téléchargement, l'ESP32 ne fera pas de Reset, Faites un contact bref sur le bouton « RST1 ».

5.2.3 Programmation simple de MLH3

1. Positionnez le Jumper « BOOT »

2. Mettez la carte sous tension en raccordant la prise USB
3. Enlevez le Jumper « BOOT »
4. Dans le dossier « tools », exécutez le fichier de commande commande « MLH3-WT32-ETH01.bat »
5. Sélectionnez le port « COM » correspondant au connecteur USB
6. Le téléchargement se passe en 2 étapes : fichiers du serveur web, puis firmware
7. En fin de téléchargement, l'ESP32 ne fera pas de Reset, Faites un contact bref sur « RST » à l'aide du Jumper.

5.2.4 Programmation simple de MLH4

1. Mettez la carte sous tension en raccordant la prise USB
2. Dans le dossier « tools », exécutez le fichier de commande commande « MLH3-WT32-ETH01.bat »
3. Sélectionnez le port « COM » correspondant au connecteur USB
4. Le téléchargement se passe en 2 étapes : fichiers du serveur web, puis firmware

5.2.5 Programmation simple de MLH5

5. Mettez la carte sous tension en raccordant la prise USB
6. Dans le dossier « tools », exécutez le fichier de commande commande « MLH5.ttgo-lora32-v21new.bat »
7. Sélectionnez le port « COM » correspondant au connecteur USB
8. Le téléchargement se passe en 2 étapes : fichiers du serveur web, puis firmware
9. En fin de téléchargement, l'ESP32 ne fera pas de Reset, Faites un contact bref sur le bouton « RST ».

5.2.6 Programmation avec l'IDE Arduino

Avec l'IDE Arduino, l'ESP32 doit être compilé avec le Core version $\geq 2.0.11$ (la version 3.x, n'a pas encore été validée), et installer les add-on pour télécharger le dossier « Data » dans la mémoire flash de l'ESP32.

⊠ Attention, pour l'ESP32-S2, le téléchargement du dossier « data » n'est, à ce jour, pas possible avec l'IDE (l'add-on ne sait pas le gérer) ; il faut donc utiliser le fichier de commande (dans « tools ») pour programmer celui-ci. Si vous adaptez le programme, il faudra « Exporter le binaire compilé » et le téléverser de la même manière.

Ouvrez le projet « lora2ha-HUB-x-y.ino », ou « x » et « y » sont les numéros de version.

Sélectionnez le « Type de carte » adapté au HUB que vous programmez :

- MLH1 : LOLIN S2 Mini
- MLH2 : Olimex-POE_ISO
- MLH3 : WT32-ETH01 Ethernet module
- MLH4 : WEMOS D1 MINI ESP32 (ou MH ET LIVE ESP32MiniKit)
- MLH5 : TTGO LoRa32 v2.1 (1.6.1)

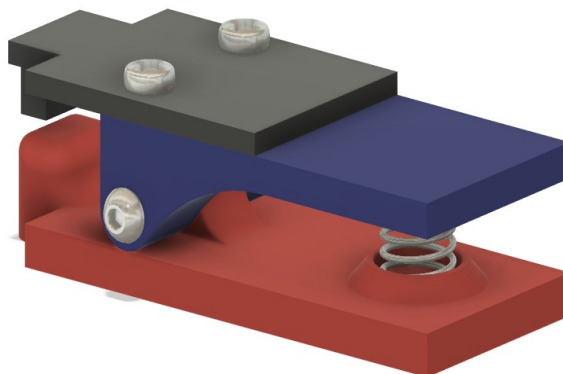
Si vous avez installé toutes les bibliothèques requises, vous pouvez compiler et télécharger le firmware.

Utilisez l'add-on « ESP32 Sketch Data Upload » pour téléverser le dossier « data » contenant le serveur web.

5.3 Programmation UPDI

Les HUB (à partir de la version 2.0, sauf MLH1) permettent de programmer tous les modules équipés d'un ATtiny3216, pour cela il suffit de connecter à 3 broches UPDI du HUB à celle de la carte MLT3216 ou du module, en respectant bien les signaux VCC, GND et UPDI.

Il est possible d'utiliser des fils Dupont si des connecteurs ont été montés ou une « pince » de programmation.



Le PCB de la carte MLT3216 et « sécable » et fourni un petit PCB permettant de fabriquer un pince de programmation(en noir) ; un fichier 3D est disponible afin de pouvoir imprimer les différents éléments de la pince (rouge et bleu), il faut ensuite :

- 2 vis M2x5 (pour fixer la carte au dessus)
- 1 vis M2x20 et son écrou (pour l'axe de la pince)
- 1 ressort de compression, diamètre 6mm longueur 10mm

Sur la page web du HUB, en bas à droite de la zone « Devices », cliquez sur le bouton « Upload device ».

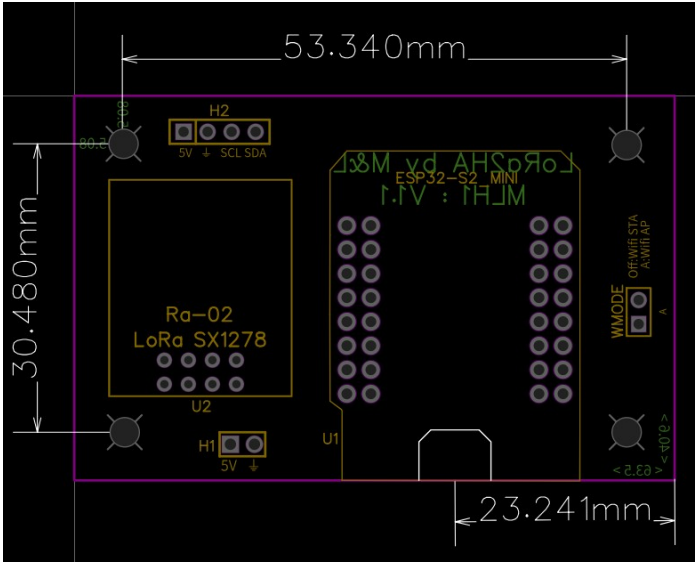
L'ESP32 peut contenir des firmware dans sa flash interne (Internal HUB) ou il est possible de récupérer des firmware sur le GitHub du projet (GitHub).

Clisez simplement sur le bouton « Upload » pour téléverser le firmware dans le module, après avoir connecté la pince UPDI

6 ANNEXE A : Nomenclatures

6.1 MLH1

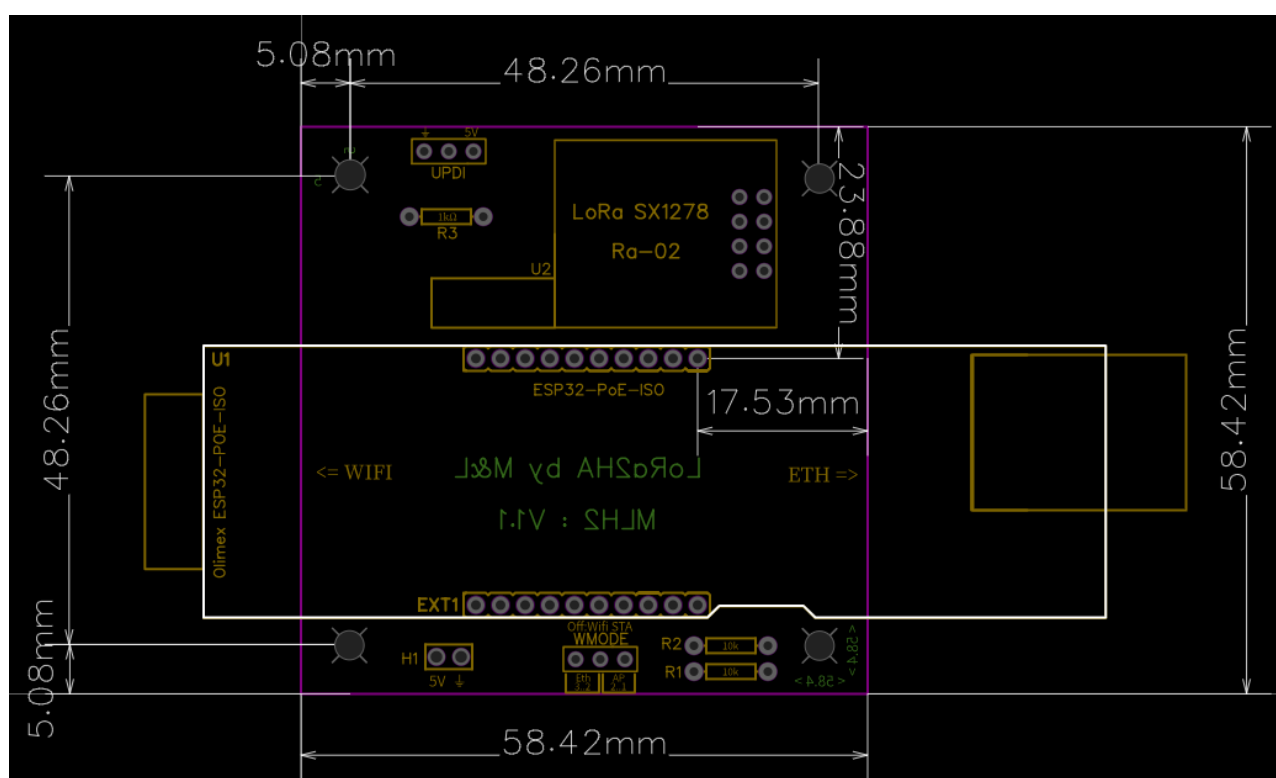
U1	ESP32-S2 Mini	
U2	RA-02	Module LoRa SX1278 433MHz
WMODE	Connecteur 1x2 2.54	



L'ESP32-S2 n'étant quasiment pas utilisé, cette carte n'est pas encore à jour de la dernière version, la gestion du WMODE.

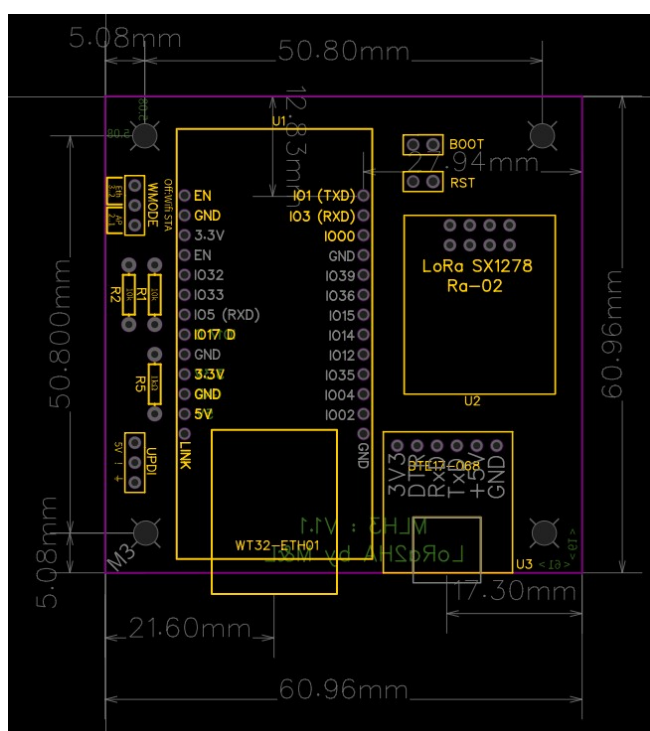
6.2 MLH2

U1	Olimex-POE-ISO	
U2	RA-02	Module LoRa SX1278 433MHz
R1, R2	10K Ω	
R3	1K Ω	
WMODE	Connecteur mâle 1x3 2.54	
H1	Connecteur femelle 1x2 5.54	optionnel
UPDI	Au choix selon le besoin de connexion	



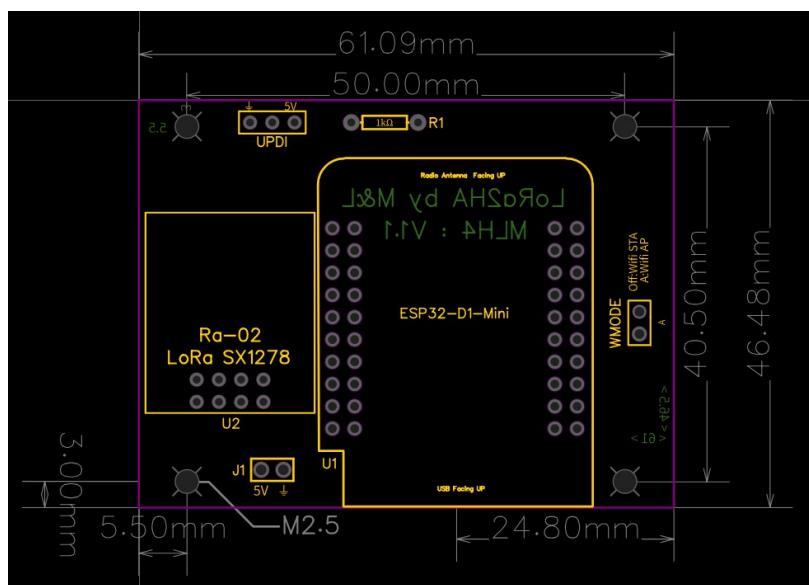
6.3 MLH3

U1	WT32-ETH01	
U2	RA-02	Module LoRa SX1278 433MHz
U3	Carte USB/TTL BTE17-06B	Alimentation / programmation
BOOT	Connecteur mâle 1x2 2.54	Jumper lors de la programmation
RST	Connecteur mâle 1x2 2.54	Jumper lors de la programmation
R1, R2	10K Ω	
R3	1K Ω	
WMODE	Connecteur mâle 1x3 2.54	
UPDI	Au choix selon le besoin de connexion	



6.4 MLH4

U1	ESP32-D1 Mini	
U2	RA-02	Module LoRa SX1278 433MHz
J1	Connecteur 1x2 2.54	optionnel
WMODE	Connecteur 1x2 2.54	
R1	1K Ω	
UPDI	Au choix selon le besoin de connexion	



6.5 MLH5

Il n'y a pas de PCB pour le module MLH5.

L'interrupteur WMode doit être connecté entre IO4 et GND.