



LoRa2HA

document version 0.3

LoRa2HA est un ensemble d'équipements permettant une interconnexion entre des capteurs éloignés et un serveur HomeAssistant (ou MQTT) par radio en utilisant des modules LoRa.

Le HUB, à base d'ESP32 (Wifi ou Ethernet), permet la collecte des données radio et les envois vers 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 projet contient les sources de tous les firmwares, les schémas des cartes ainsi que les « Gerber » afin de pouvoir les fabriquer.

Sommaire

| | |
|---|----|
| Les équipements..... | 3 |
| Le HUB..... | 3 |
| Les modules « capteurs »..... | 3 |
| Fonctionnement..... | 4 |
| Mise en route..... | 5 |
| Installation du HUB..... | 5 |
| Installation des modules capteurs..... | 7 |
| Module MLD01 Linky..... | 7 |
| Module MLE21 (2 in / 1 out)..... | 8 |
| Capteur sur J2..... | 8 |
| Capteur sur J3..... | 8 |
| Sortie relais sur J4..... | 8 |
| Module MLA30 (3 in / 0 out)..... | 9 |
| Capteur sur J1..... | 10 |
| Capteur sur J2 et J3..... | 10 |
| Alimentation..... | 10 |
| Paramétrage des capteurs dans le HUB..... | 11 |
| Contrôle dans H.A..... | 12 |
| Montage des cartes..... | 14 |
| MLH1..... | 14 |
| MLH2..... | 15 |
| MLH3..... | 16 |
| MLH4..... | 17 |
| MLD01..... | 18 |
| MLE21..... | 18 |
| MLA30..... | 19 |
| ML99..... | 21 |
| Programmation..... | 22 |
| Prérequis..... | 22 |
| Programmer le HUB..... | 22 |
| Programmer le module MLD01 (Linky)..... | 23 |
| Programmer les modules ATtiny84 (MLE21, MLA30, etc.)..... | 23 |
| Debug..... | 24 |
| Programmer le module MLE21..... | 24 |
| Programmer le module MLA30..... | 24 |
| ANNEXE A : Nomenclatures..... | 25 |
| MLH1..... | 26 |
| MLH2..... | 27 |
| MLH3..... | 28 |
| MLH4..... | 29 |
| MLD01..... | 30 |
| MLE21..... | 31 |
| MLA30 (v1.1)..... | 32 |
| MLA10 (projet futur)..... | 33 |
| ML99..... | 34 |
| ANNEXE B : Schémas..... | 35 |
| MLD01..... | 35 |
| MLE21..... | 35 |
| MLA30..... | 36 |
| MLA10..... | 36 |
| MLH1..... | 37 |
| MLH2..... | 37 |
| MLH3..... | 37 |
| MLH4..... | 38 |
| ML99..... | 39 |

Les équipements

Les références des modules adoptent la terminologie suivante :

- ML : Module LoRa (et non Matière & Lumière 😊)
- 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 21 = 2 entrées / 1 sortie), éventuellement suivi d'un ordre de création.

Le HUB

Il existe 4 types de HUB selon le besoin, tous équipés d'un module LoRa :

- MLH1 : Avec un ESP32-S2 Mini, en wifi
- MLH2 : Avec un Olimex-POE-ISO, en Ethernet avec POE
- MLH3 : Avec un WT32-ETH01, en Ethernet
- MLH4 : Avec un ESP32-D1-Mini, en wifi

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

Les modules « capteurs »

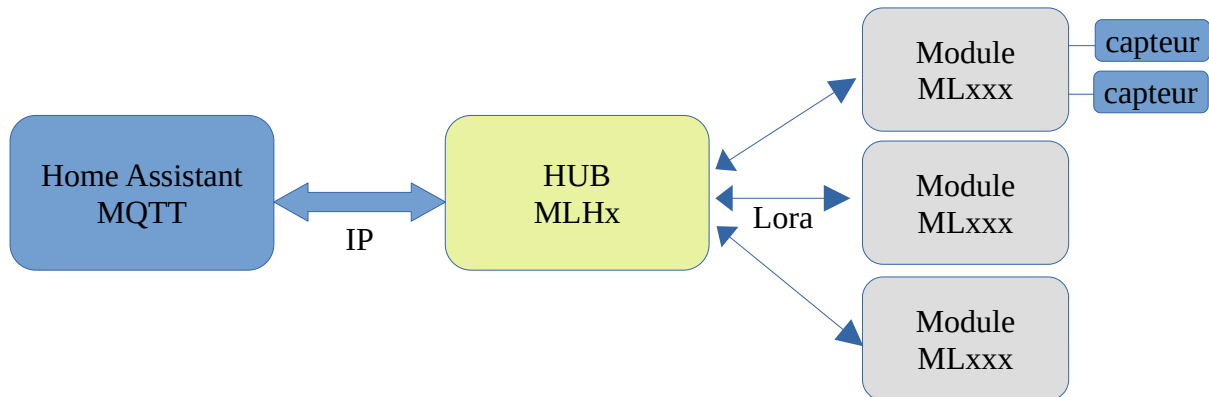
A ce jour il existe 3 types de modules, et 4 projets en cours :

- MLD01 : Relevé TIC Linky (alimenté par USB externe)
- MLD02 : Relevé TIC Linky (alimenté par Linky) (projet en cours)
- MLE21 : 2 entrées / 1 sortie, alimenté en 8-32vdc
- MLE12 : 1 entrée / 2 sorties, alimenté en 8-32vdc (projet en cours)
- MLA30 : 3 entrées, autonome, basse consommation, avec batterie et panneau solaire
- MLA10-1 : Télécommande 1 bouton (ou contact sec), sur pile CR2450 (projet en cours)
- MLA10-2 : 1 bouton (ou contact sec), sur pile CR2450 ou batterie (projet en cours)

Chacun d'eux possède un Firmware par défaut mais il est possible d'adapter celui-ci pour modifier les fonctionnalités de chaque module.

Un module codeur (ML99) permet de donner une adresse à chaque module (à base d'ATtiny84) lors de l'installation, sans avoir besoin de le reprogrammer.

Fonctionnement



Chaque module (MLxxx) transmet les informations des capteurs (par LoRa) vers le HUB, connecté à H.A. (MQTT) par IP, ce dernier va pouvoir rediriger 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 (> 2 dBi).

L'antenne ressort des module 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 plat, verticales et perpendiculaires 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 :
<https://fr.aliexpress.com/item/1005003818758184.html>



Mise en route

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

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

Installation du HUB

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

Le HUB une fois programmé (ou reçu prêt à l'emploi) doit être raccordé au réseau IP (identique à celui de H.A.).

Dans le cas d'un HUB wifi, ce dernier ne connaissant pas les informations de votre wifi, il est passé automatiquement en mode « Point d'accès ». Cherchez un réseau wifi nommé « lora2ha0 » et connectez-vous ; le mot de passe est : 12345678.


Le HUB est à l'adresse IP 192.168.4.1 ou l'URL lora2ha0.local ; la page ci-dessous doit s'afficher.

Oui, c'est en anglais... mais très simple 😊



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

LoRa2HA
 By Matières & Lumière



Network & Radio

Module

UID Code

MAC address

Radio Freq.

0

FC:F5:C4:07:AB:34

433

Wifi

SSID

Password

Wifi SSID

Wifi password

Mqtt

Host

User

Password

Mqtt host

Mqtt user

Mqtt pass

Save

L'information de l'adresse MAC vous permet de pouvoir gérer votre serveur DHCP et affecter une adresse IP spécifique.

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 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.

Une fois sauvegardé, il faut redémarrer le HUB, physiquement ou en cliquant sur le bouton « Reset » dans la zone « Tools ».

Tools

Update firmware 0.6

Select .bin file to upload

Parcourir... Aucun fichier sélectionné.

Update

Send file

Select file to send (config, images,...)

Parcourir... Aucun fichier sélectionné.

Send

Download configuration file

☐ Log LoRa packet

Reset

Le wifi du point d'accès du HUB devrait être perdu et si il n'y a pas de problème dans le paramétrage il a dû se connecter au wifi qui a été configuré, vous pouvez accéder à la même page web, via votre réseau, à l'adresse « lora2ha0.local ».

Le HUB est prêt à recevoir les informations radio des « modules capteurs ».

Installation des modules capteurs

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

Une fois programmé (ou reçu tout prêt), le module doit avoir une adresse unique.

Si vous n'avez qu'un seul module il n'y a rien à faire, il aura son adresse par défaut.

Pour affecter une adresse il faut connecter le boîtier « codeur » (ML99) sur le connecteur ICSP du module, choisir l'adresse à l'aide des 2 encodeurs rotatifs, par exemple pour l'adresse 31 il faut mettre 3 sur **10X** et 1 sur **1X**. Alimentez le module et celui-ci prendra directement l'adresse indiquée. Après quelques secondes vous pouvez débrancher le module et déconnecter la carte ML99.

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 référencées dans le tableau 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) |

Module MLE21 (2 in / 1 out)

Le module est à l'adresse « 20 » par défaut. Si vous installez plusieurs modules du même type, incrémentez l'adresse de chaque nouveau module (21, 22, etc.).

J1 permet d'alimenter le module en courant continu avec une tension comprise entre 7 et 35v.

J2 et J3 permettent de connecter les capteurs.

J4 permet de connecter la sortie du relais (NO ou NC).

Capteur sur J2

J2 permet de connecter plusieurs types de capteurs :

- numérique, de type DS18B20
- analogique
- binaire (contact)

Si un capteur DS18B20 est raccordée, il faut monter la résistance RJ2 (4,7k).

Si un capteur résistif est raccordé, et qu'il y a besoin d'établir un pont de résistance pour la mesure, il faut mettre en RJ2 la résistance à votre pont.

Si un capteur binaire est monté, il ne faut pas monter RJ2.

Capteur sur J3

J3 accepte les capteurs analogiques et binaires.

Si un capteur résistif est raccordé, et qu'il y a besoin d'établir un pont de résistance pour la mesure, il faut mettre en RJ3 la résistance à votre pont.

Si un capteur binaire est monté, il ne faut pas monter RJ3.

Sortie relais sur J4

Le relais HFD4/5 installé permet de gérer sans problème un contact sec (type ouverture de portail), si il doit agir sur un équipement actif, voici ses caractéristiques :

| CONTACT DATA | |
|------------------------------------|--|
| Contact arrangement | 2C |
| Contact resistance | 100mΩ max. (at 10mA 30mVDC) |
| Contact material | AgPd + Au plated, AgNi + Au plated |
| Contact rating (Res. load) | 2A 30VDC 0.5A 125VAC |
| Max. switching current | 2A |
| Max. switching voltage | 250VAC / 220VDC |
| Max. switching power | 62.5VA / 60W |
| Min. applicable load ¹⁾ | 10mV 10μA |
| Mechanical endurance | 1 x 10 ⁵ ops |
| Electrical endurance ²⁾ | 1 x 10 ⁵ ops (AgNi + Au plated, 0.5A 125VAC, Resistive load, at 40°C, 1s on 9s off) |

Notes: 1) Min. applicable load is reference value. Please perform the confirmation test with the actual load before production since reference value may change according to switching frequencies, environmental conditions and expected contact resistance and reliability.
2) Electric endurance data are collected in one pair CO contact test.

| COIL | | |
|------------------|---|-----------------|
| Coil power | Single side stable | See "COIL DATA" |
| | 1 coil latching | See "COIL DATA" |
| Temperature rise | 50K max. (At 1A load, 85°C environment) | |

| CHARACTERISTICS | | |
|--|-------------------------|-----------------------|
| Insulation resistance | | 1000MΩ (at 500VDC) |
| Dielectric strength | Between coil & contacts | 1600VAC 1min |
| | Between open contacts | 1000VAC 1min |
| | Between contact sets | 1800VAC 1min |
| Surge withstand voltage | | |
| Between open contacts (10/160μs) | | 1500VAC (FCC part 68) |
| Between coil & contacts (2/10μs) | | 2500VAC (Telecordia) |
| Operate time (Set time) | | 3ms max. |
| Release time (Reset time) | | 3ms max. |
| Ambient temperature | | -40°C to 85°C |
| Humidity | | 5% to 85% RH |
| Vibration resistance | | 10Hz to 55Hz 3.3mm DA |
| Shock resistance | Functional | 735m/s ² |
| | Destructive | 980m/s ² |
| Termination | | DIP, SMT |
| Unit weight | | Approx. 0.8g |
| Moisture sensitivity levels (Only for SMT type, JEDEC-STD-020) | | MSL 3 |
| Construction | | Plastic sealed |

Notes: 1) The data shown above are initial values.
2) UL insulation system: Class A

| SAFETY APPROVAL RATINGS | | |
|-------------------------|------------------|---|
| UL/CUL | AgPd + Au plated | 0.5A 125VAC at 70°C |
| | AgNi + Au plated | 1A 30VDC at 85°C 2A 30VDC at 40°C 0.5A 125VAC at 40°C |
| | AgPd + Au plated | 0.5A 125VAC at 85°C |
| TUV | AgPd + Au plated | 0.5A 125VAC at 85°C |
| | AgNi + Au plated | 1A 30VDC at 85°C 0.5A 125VAC at 85°C |

Notes: 1) All values unspecified are at room temperature.
2) Only typical loads are listed above. Other load specifications can be available upon request.

Le connecteur J4 est utilisé pour connecter la sortie relais, sur NO (Normally Open) ou NC (Normally Close).

Le Firmware par défaut (Iora2HA-MLE21-Portail.ino) permet la gestion d'un portail :

- J2 : contact sec de détection de portail ouvert
- J3 : bouton poussoir d'appel
- J4 : contact sec d'ouverture

Module MLA30 (3 in / 0 out)

Le module est à l'adresse « 50 » par défaut. Si vous installez plusieurs modules du même type, incrémentez l'adresse de chaque nouveau module (51, 52, etc.).

BAT permet de connecter une batterie 3.7v.

PV permet de connecter une cellule Photovoltaïque de 5V.

J4 permet de connecter la carte de gestion de charge batterie, type CRLD20MA.

J1 permet de connecter un capteur analogique, binaire ou de type DS18B20.

J2 permet de connecter un capteur analogique ou binaire.

J3 permet de connecter un capteur analogique ou binaire.

Capteur sur J1

J1 permet de connecter plusieurs types de capteurs :

- numérique, de type DS18B20
- analogique
- binaire (contact)

Si un capteur DS18B20 est raccordée, il faut monter la résistance RJ1 (4,7k).

Si un capteur résistif est raccordé, et qu'il y a besoin d'établir un pont de résistance pour la mesure, il faut mettre en RJ1 la résistance à votre pont.

Si un capteur binaire est monté, il ne faut pas monter RJ1.

Capteur sur J2 et J3

J2 et J3 acceptent les capteurs analogiques et binaires.

Si un capteur résistif est raccordé, et qu'il y a besoin d'établir un pont de résistance pour la mesure, il faut mettre en RJ2 (ou RJ3) la résistance à votre pont.

Si un capteur binaire est monté, il ne faut pas monter RJ2 (ou RJ3).

Alimentation

L'alimentation du module peut se faire à partir d'une simple batterie connectée sur BAT.

Si une cellule solaire est connectée sur PV, il faut monter un module CRLD20MA sur le connecteur J4 afin de gérer la charge.

⊗ Attention, la batterie doit avoir une protection, le module CRLD20MA ne gère pas la protection de charge/décharge. Généralement, les batteries comme celles ci-dessous sont équipées d'une protection



Paramétrage des capteurs dans le HUB

Ouvre la zone « Devices ».

The screenshot shows a web interface for managing devices. At the top is a blue header bar labeled 'Devices'. Below it, there are two device configuration cards, numbered 1 and 2.

Device 1:


- @: 1
- Name: Linky
- RL ver: 0
- ID 1: Label 'PuissApp', Data type 'Int'. HA configuration: Numeric sensor, apparent_power, VA, 0, 0, 10000.
- ID 2: Label 'Compteur', Data type 'Int'. HA configuration: Numeric sensor, energy, kWh, 0.
- Buttons: 'Add child' and a menu icon.

Device 2:

- @: 4
- Name: Portail
- RL ver: 0
- ID 1: Label 'Bouton', Data type 'Int'. HA configuration: Numeric sensor, 1, 0, 5.
- ID 3: Label 'Relais', Data type 'Int'. HA configuration: Switch, 0.
- Buttons: 'Add child' and a menu icon.

At the bottom right of the interface is a blue button labeled 'Add device'.

- « @ » est l'adresse par défaut du module ou celle que vous avez fixé avec le module ML99
- « Name » est le nom de l'appareil qui apparaîtra dans H.A.
- « RL ver » est le n° de version des paquets LoRa (laissez 0), cette donnée est en prévision de futures évolutions
- « 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 firmwares

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

The 'HA Configuration' dialog box is shown. It has a title bar with a close button (X). The configuration fields are:

- Type: Numeric sensor (dropdown)
- Class: apparent_power (text input)
- Unit: VA (text input)
- Expire after (sec, 0 = never): 0 (text input)
- Min: 0 (text input)
- Max: 10000 (text input)

At the bottom are two buttons: 'Cancel' (blue) and 'Confirm' (grey).

- « Type » permet de choisir dans une liste le type de capteur/actionneur dans H.A.
- « Class » selon le Type que vous avez choisi, 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>
- « Unit » permet d'indiquer l'unité correspondant à la mesure du capteur
- « Expire after » est la durée pendant laquelle la mesure est valide (indisponible au sens H.A.), cela peut être utile dans le cas d'un bouton poussoir ou vous souhaitez uniquement déclencher lors d'un appui ; vous allez déclencher l'automatisation à la réception de l'appui et en indiquant « 1 » dans « Expire after », le trigger (ou capteur) deviendra « indisponible ».
- « 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.

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.

The screenshot shows the 'HA Configuration' window. Under the 'Type' dropdown (set to 'Numeric sensor'), the 'Class' dropdown is open, displaying a list of options: 'volatile_organic_compounds', 'voltage', 'volume', 'water', 'weight', 'wind_speed', and 'U'. The 'voltage' option is currently selected and highlighted.

Pour ajouter un nouveau capteur/actionneur dans un module, clique sur le bouton « Add Child ».

Pour ajouter un nouveau module, cliquez sur le bouton « Add Device ».

Après avoir tout configuré, cliquez sur le bouton « Save all devices » et redémarrez le HUB (Physique ou « Tools / Reset ») pour recharger la nouvelle configuration.

☒ Pour supprimer un Module (Child), laissez la zone « Label » vide, il ne sera pas pris en compte lors de l'enregistrement.

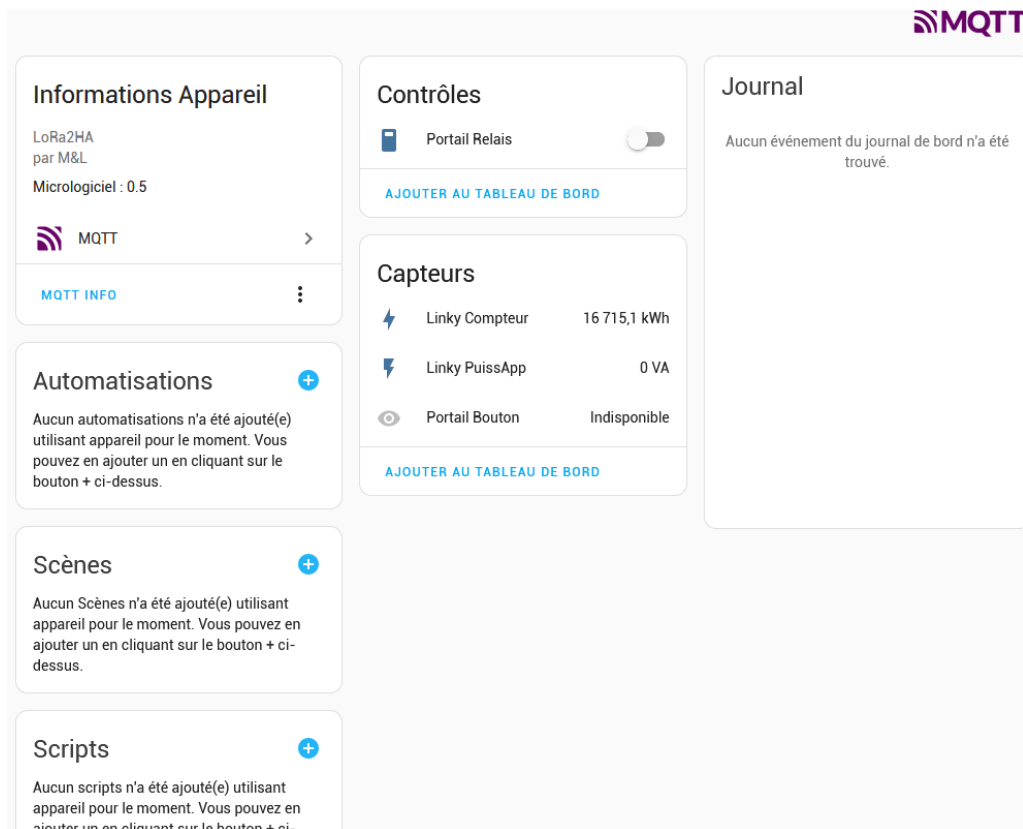
☒ Pour supprimer un « Device », si il n'y a plus de « Child », laissez la zone « Name » vide, il ne sera pas pris en compte lors de l'enregistrement.

Contrôle dans H.A.

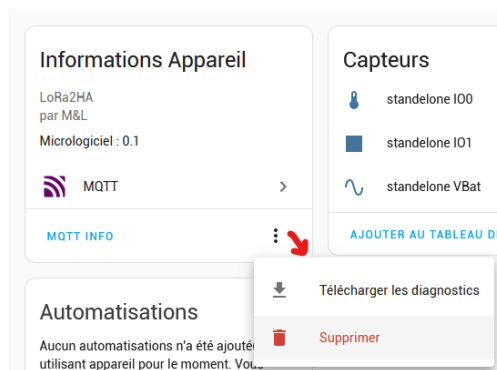
Dans H.A., cliquez sur « Paramètres » puis sur « Appareils et Services ».

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

Allez chercher « lora2ha0 » dans la liste et cliquez dessus, vous devriez obtenir une page dans ce style :



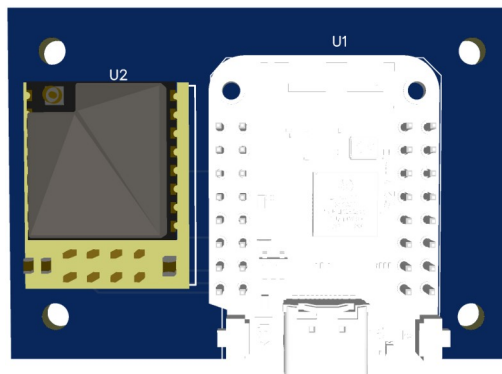
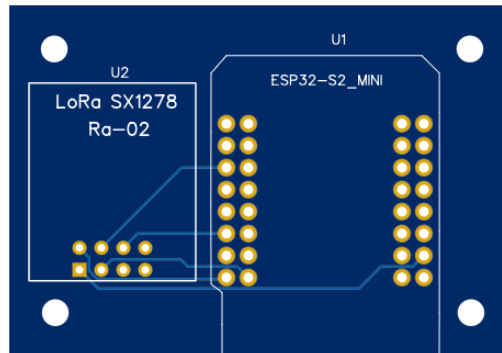
Si vous voyez apparaître des anciens capteurs ou contrôles (après changement de nom ou de type par exemple), il faut supprimer l'appareil en totalité, il sera re-crée lors du prochain redémarrage du HUB.



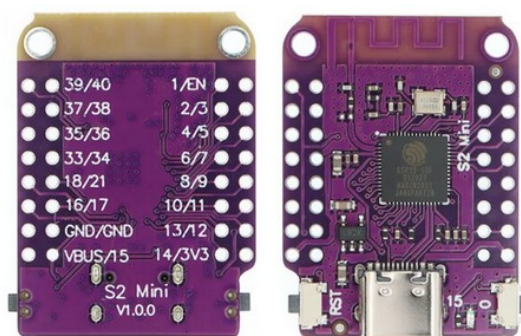
Montage des cartes

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

MLH1

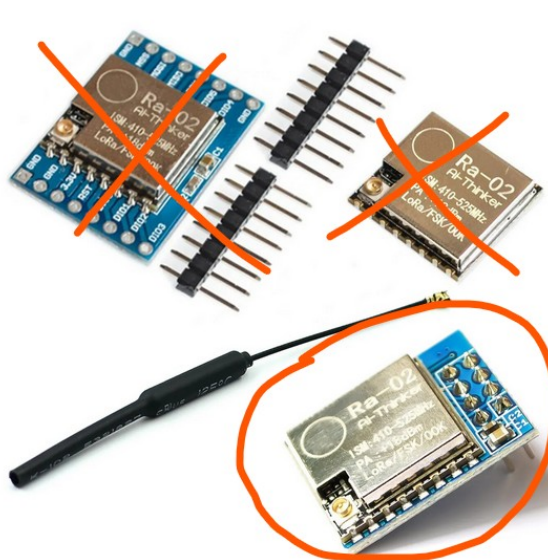


La carte MLH1 doit être équipée d'un ESP32-S2 Mini.



Le module LoRa conseillé est en 433MHz, RA-01 (antenne filaire) ou RA-02 (connecteur IPX). Il sera privilégié le RA-02 pour de la longue distance en choisissant une antenne adaptée (> 2 Dbi).

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



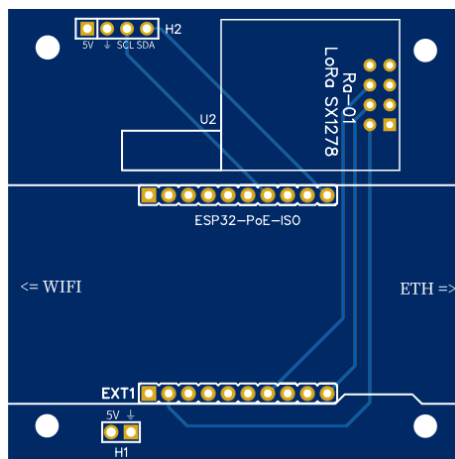
Exemples de lien de matériel :

RA-01 / RA-02 : <https://fr.aliexpress.com/item/1005001593805902.html>

ESP32-S2 Mini : <https://fr.aliexpress.com/item/1005004344359250.html>

L'alimentation de la carte se fait au travers du connecteur USB de l'ESP32 (transfo séparé conseillé).

MLH2



La carte MLH2 doit être équipée d'un ESP32-POE-ISO, connecteur USB vers le bas (H1).

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.

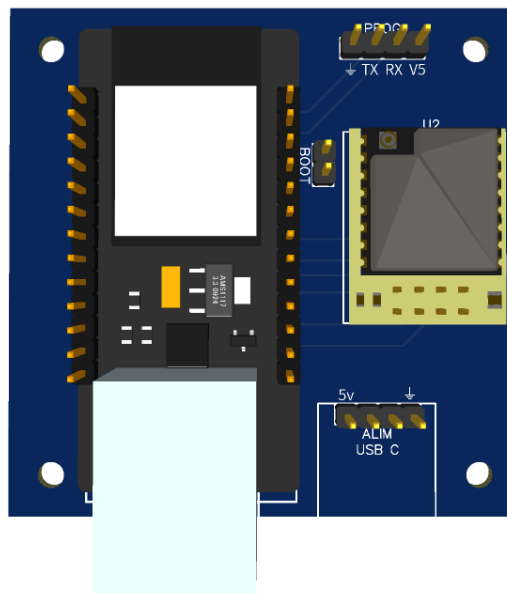
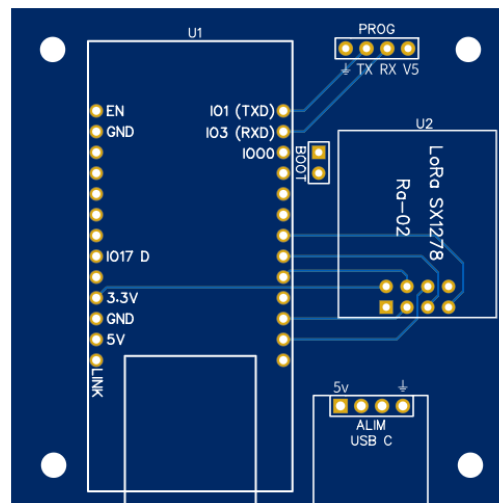
Le connecteur H2 peut être utilisé (après modification du firmware) pour connecter une extension I2C, par exemple un écran Oled.

Le module LoRa est identique à la carte MLH1.

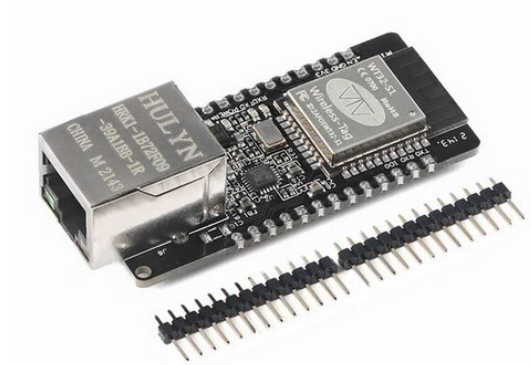
Exemple de lien d'achat : <https://www.olimex.com/Products/IoT/ESP32/ESP32-POE-ISO/open-source-hardware>



MLH3



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



Le module LoRa est identique à la carte MLH1.

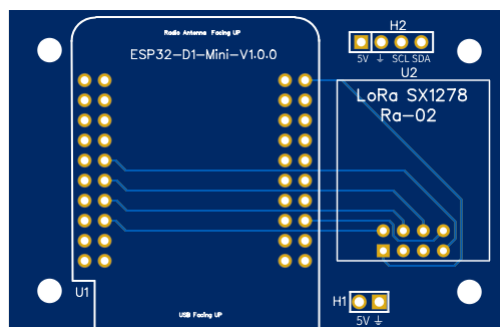
L'alimentation se fait au travers d'une mini carte USB dont on utilisera uniquement les pins +5V et GND. Celle-ci peut être soudée directement sur la carte en utilisant le modèle donné en exemple d'achat, ou en utilisant un modèle à visser sur le boîtier et en la raccordant au connecteur.

La programmation s'effectue uniquement au travers du connecteur « PROG » à l'aide de n'importe quel convertisseur USB/TTL (type FT232), il doit avoir les pin : +5v, GND, TX, RX.

Exemple de liens d'achat :

- WT32-ETH01 : <https://fr.aliexpress.com/item/1005002920674329.html>
- USB-C : <https://fr.aliexpress.com/item/1005002795893679.html>
- USB/TTL : <https://fr.aliexpress.com/item/1005005921832515.html>

MLH4



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



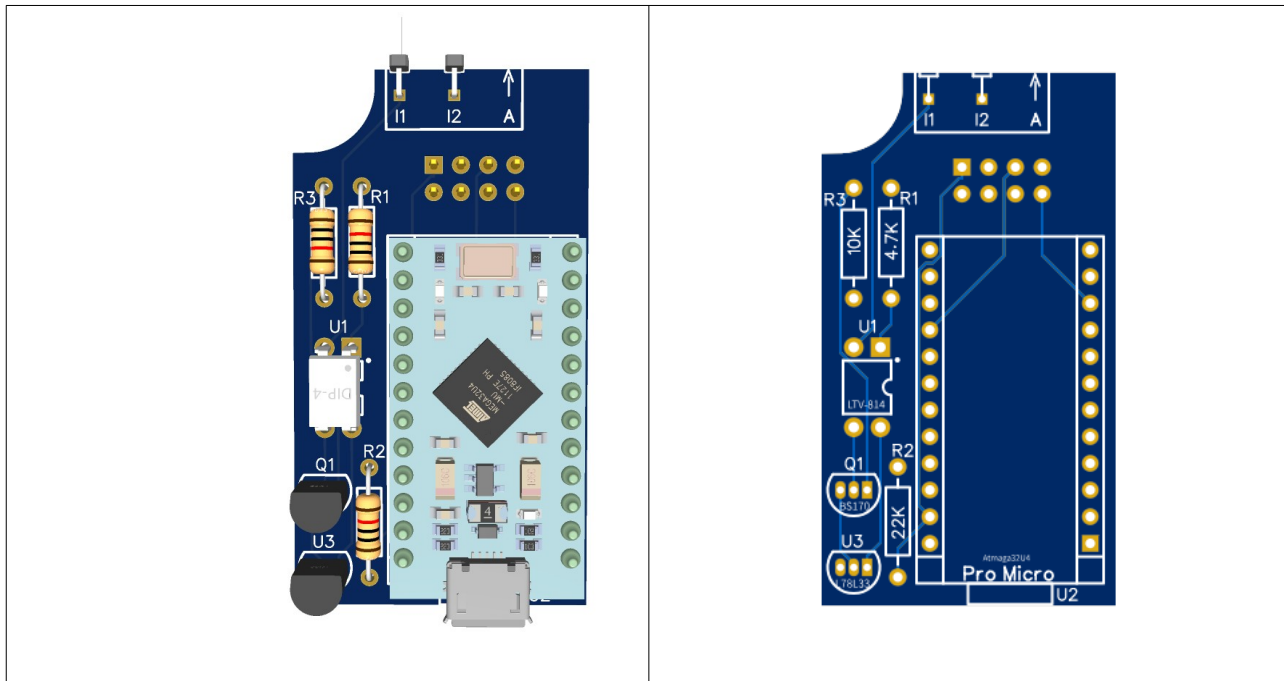
Le module LoRa est identique à la carte MLH1.

Le connecteur H1 peut être utilisé pour l'alimentation (5V).

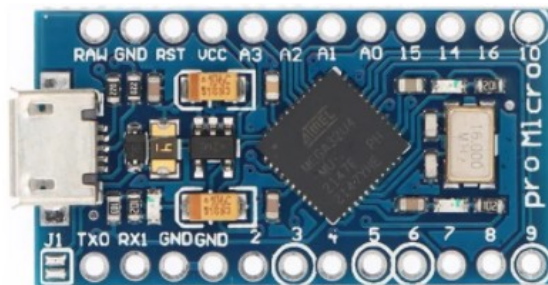
Le connecteur H2 peut être utilisé (après modification du firmware) pour connecter une extension I2C, par exemple un écran Oled.

Exemple de lien d'achat : <https://fr.aliexpress.com/item/1005001621844145.html>

MLD01



La carte MLD01 doit être équipée d'un Arduino Pro Micro (ATMega32U4).

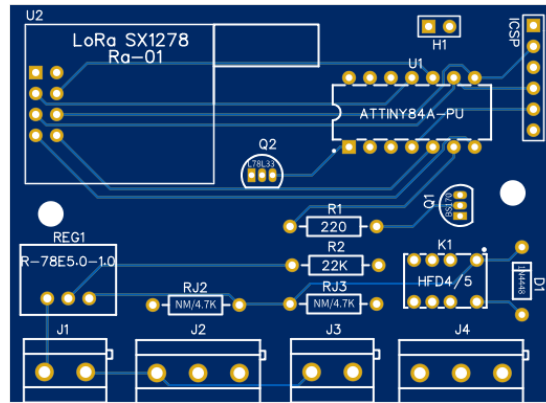


Le module LoRa est identique à la carte MLH1.

Exemple de liens d'achats : <https://fr.aliexpress.com/item/32846843498.html>

➔ Attention, sur la version v1.0 du PCB, il y a une petite erreur de routage, il faut relier les pin « 5 » et « RX » du ProMicro après avoir soudé celui-ci.

MLE21



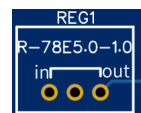
La carte MLE21 doit être équipée d'un ATtiny84-20PU.

Le module LoRa est identique à la carte MLH1.

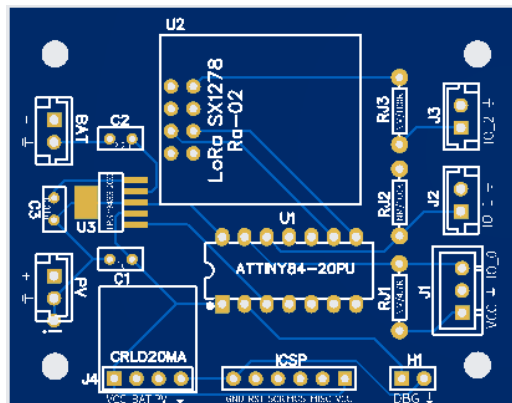
Exemple de liens d'achat :

- Convertisseur DC/DC : <https://www.digikey.fr/fr/products/detail/recom-power/R-78E5-0-1-0/4930585>
- Relais HFD4 5V : <https://fr.aliexpress.com/item/1005004845392306.html>

Il est possible d'alimenter directement la carte avec du 5V sur le connecteur J1, dans ce cas REG1 n'est pas monté mais il faudra relier les pastilles « in » et « out » de son connecteur .



MLA30



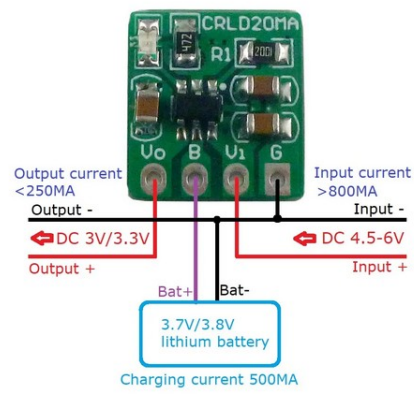
La carte MLA30 doit être équipée d'un ATtiny84-20PU.

Le module LoRa est identique à la carte MLH1.

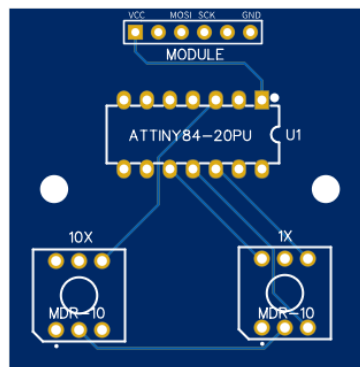
La carte « chargeur » est de type CRLD20MA, 3.3V ; exemple de lien d'achat :

<https://fr.aliexpress.com/item/1005005478025758.html>

1 Charge and Discharge

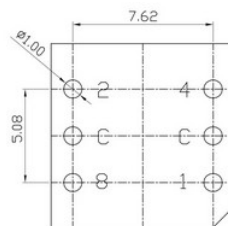


ML99



La carte ML99 doit être équipée d'un ATTiny84-20PU.

Les encodeurs 10 positions (DIP) doivent avoir les caractéristiques suivantes :



Exemple de lien d'achat : <https://fr.aliexpress.com/item/1005002489382650.html>

Programmation

Ce chapitre sous-entend que vous maîtrisez la programmation d'un Arduino, que ce soit les ATtiny ou les ESP32. Pour les ATtiny, il faut avoir un « programmeur » ou transformer un UNO (Nano) en Programmeur ISP.

Plusieurs tuto traitent de ces différents sujets si vous n'êtes pas un expert ; cherchez « ATtiny84 programmer », en anglais vous aurez plus de réponse.

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 dossier des librairies Arduino ; pour les ATtiny84 la librairie « tinySPY » doit être installée en téléchargeant le fichier zip sur le GitHub <https://github.com/JChristensen/tinySPI> et en le décompressant dans le dossier des librairies Arduino..

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

Programmer le HUB

L'ESP32 doit être compilé avec le Core version $\geq 2.0.11$ (pas la version 3 alpha, il reste trop de bug de compilation), et installer les add-on pour télécharger le dossier « Data » dans la mémoire flash de l'ESP32.

Lien à mettre dans « Préférences » de l'IDE pour avoir la liste des cartes ESP :

https://dl.espressif.com/dl/package_esp32_index.json

Vous devez avoir installé les librairies suivantes avec le gestionnaire de bibliothèque :

- ArduinoJson, version ≥ 7.0 (par Benoit Blanchon)
- AsyncTCP, version $\geq 1.1.4$ (par dvarel)
- PubSubClient

Vous devez installer les librairies suivantes en décompressant le fichier zip dans le dossier des librairies Arduino :

- <https://github.com/me-no-dev/ESPAsyncWebServer>

Ouvrez le projet « lora2ha-HUB.ino », dans le fichier « config.h » dé-commentez la ligne `#define USE_ETHERNET` si vous utilisez une carte Ethernet au lieu de Wifi (Olimex-POE ou WT32-ETH01).

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)

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

Programmer le module MLD01 (Linky)

Installez la librairie «TeleInfo» à l'aide du gestionnaire de bibliothèque.

Ouvrez le projet « lora2ha-MLD01-Linky », choisissez le type de carte « Arduino Leonardo ».

Compilez et téléchargez le firmware à l'aide du câble USB connecté à la carte.

Si vous souhaitez utiliser « DEBUG_SERIAL », il faut savoir que la liaison série USB du moniteur (dans l'IDE) ne fonctionne correctement que suite à un téléchargement ; en effet, un simple « reset » sur le ProMicro ne permettra pas d'avoir la liaison série ouverte correctement, il faut légèrement modifier le code source.

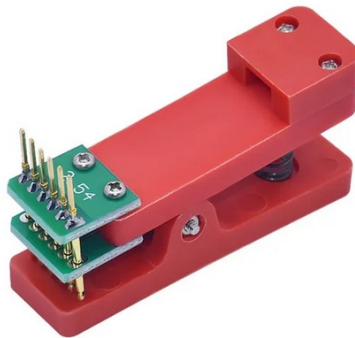
Programmer les modules ATtiny84 (MLE21, MLA30, etc.)

Pour programmer les cartes équipées d'ATtiny84, il vous faut un programmeur ISP ; il peut être fait en utilisant un simple Arduino UNO (ou Nano).

Il existe plusieurs tutoriels sur le sujet, choisissez celui qui correspond le mieux à votre cas de figure.

Les cartes sont équipées d'un connecteur ICSP, il est possible d'utiliser le type de pince ci-dessous pour programmer, dans ce cas il n'est pas utile de souder le connecteur.

⊗ Important : Il est impératif de débrancher l'alimentation de la carte durant la phase de programmation, la carte est alimentée par le programmeur.



Pour programmer toutes les cartes basées sur l'ATtiny84, il est conseillé d'utiliser le gestionnaire de cartes avec le modèle « Attiny Microcontollers ».



Si vous n'avez pas ce modèle de carte, ajoutez la ligne ci-dessous dans les « Préférences » de l'IDE puis redémarrez le.

https://raw.githubusercontent.com/damellis/attiny/ide-1.6.x-boards-manager/package_damellis_attiny_index.json

Debug

Les cartes à base d'ATtiny84 sont toutes équipées d'un connecteur H1, il permet de brancher provisoirement une LED (avec résistance 330 ohm) qui peut donner quelques informations sur le bon fonctionnement du firmware.
Exemple ci-contre de ce que j'utilise, protégé par une gaine thermo.



Programmer le module MLE21

L'ATtiny84 est prévu pour fonctionner avec son oscillateur interne à 8MHz ; après avoir connecté le programmeur et ouvert le projet « lora2ha-MLE21-Portail », sélectionnez le type de carte et le processeur ATtiny84 ; réglez la « Clock » sur « internal / 8MHz » puis gravez la séquence d'initialisation, après avoir vérifié que vous avez sélectionné le bon programmeur, généralement « Arduino as ISP ».

Ensuite il faut « Téléverser avec un programmeur ».

Pour les capteurs Binaires (contact), ne montez pas la résistance RJn (« n » est le connecteur utilisé par le contact).

Lors de mesure Analogique, adaptez la résistance RJn au pont diviseur que vous souhaitez (« n » est le connecteur utilisé par le capteur).

Programmer le module MLA30

L'ATtiny84 est prévu pour fonctionner avec son oscillateur interne à 1MHz afin d'économiser l'énergie de la batterie ; après avoir connecté le programmeur et ouvert le projet « lora2ha-MLA30 », sélectionnez le type de carte et le processeur ATtiny84 ; réglez la « Clock » sur « internal / 1MHz » puis gravez la séquence d'initialisation, après avoir vérifié que vous avez sélectionné le bon programmeur, généralement « Arduino as ISP ».

Ensuite il faut « Téléverser avec un programmeur ».

Utilisez les différents #define afin de configurer les types de capteurs que vous souhaitez connecter.

Pour le « réveil » par contact, celui-ci s'effectue sur front descendant (par mise à la masse), donc avec un contact N.O. (Normally Open).

Pour les capteurs Binaires (contact), ne montez pas la résistance RJn (« n » est le connecteur utilisé par le contact).

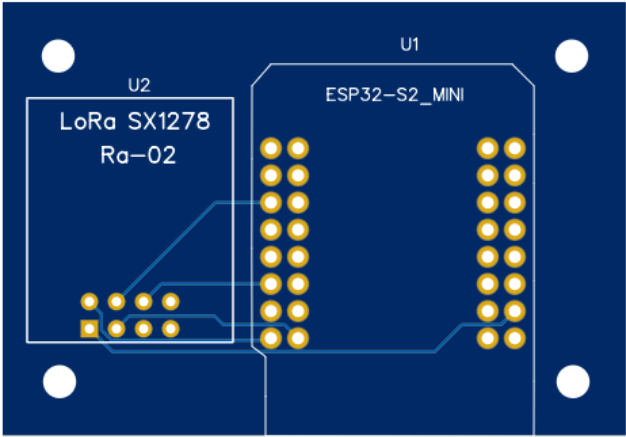
Lors de mesure Analogique, adaptez la résistance RJn au pont diviseur que vous souhaitez (« n » est le connecteur utilisé par le capteur).

ANNEXE A : Nomenclatures

A noter que toutes les dimensions nécessaires à la fabrication des boîtiers sont sérigraphiées au dos des cartes.

MLH1

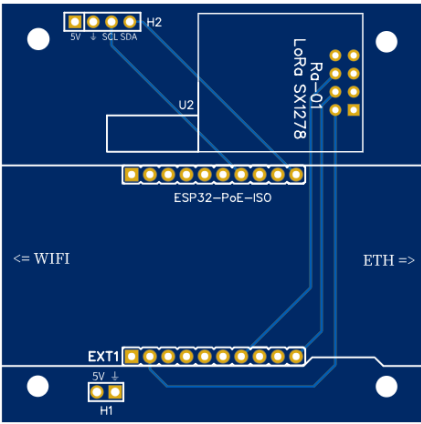
| | | |
|----|----------------|--|
| U1 | ESP32-S2 Mini | |
| U2 | RA-01 ou RA-02 | |



Taille : 58.4 x 40.6 mm

MLH2

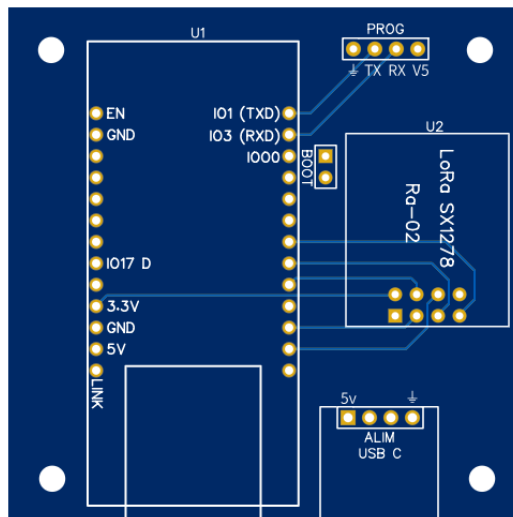
| | | |
|----|---------------------------|-----------|
| U1 | Olimex-POE-ISO | |
| U2 | RA-01 ou RA-02 | |
| H1 | Connecteur Dupont femelle | optionnel |
| H2 | Connecteur Dupont | optionnel |



Taille : 58.4 x 58.4 mm

MLH3

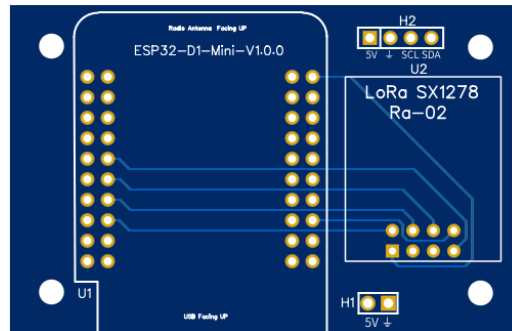
| | | |
|------------|---------------------------------|---------------------------------|
| U1 | WT32-ETH01 | |
| U2 | RA-01 ou RA-02 | |
| BOOT | Connecteur Dupont male | Jumper lors de la programmation |
| PROG | Connecteur Dupont | optionnel |
| ALIM USB C | Carte USB C / Connecteur Dupont | optionnel |



Taille : 61 x 61 mm

MLH4

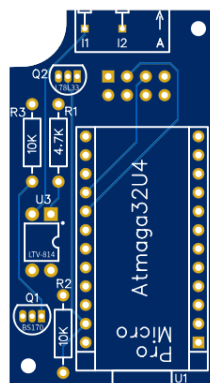
| | | |
|----|---------------------------|-----------|
| U1 | ESP32-D1 Mini | |
| U2 | RA-01 ou RA-02 | |
| H1 | Connecteur Dupont femelle | optionnel |
| H2 | Connecteur Dupont | optionnel |



Taille : 40.6 x 63.6 mm

MLD01

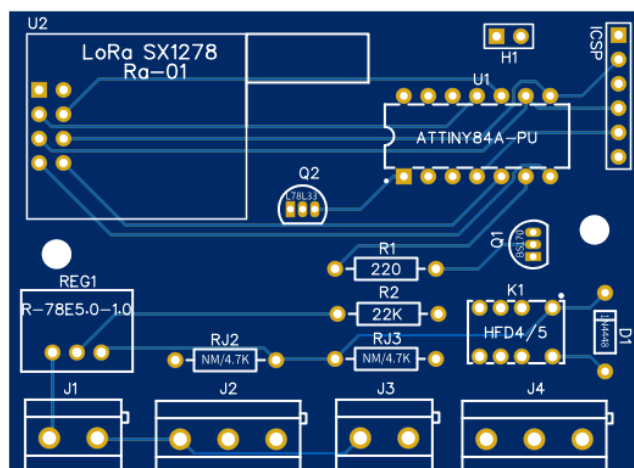
| | | |
|--------|---------------------------------------|------------------------|
| U1 | Arduino Pro Micro | 5v |
| U2 | RA-01 ou RA-02 | Au dos de la carte |
| U3 | LTV-814 | |
| Q1 | BS170 | Ou 2N7000, IRFZ44, etc |
| Q2 | L78L33 | |
| R1 | 4.7K | |
| R2, R3 | 10K | |
| I1, I2 | Dupont mâle coudé ou bornier à visser | Pas 5.08 |



Taille : 27.94 x 50.8 mm

MLE21

| | | |
|----------|--|------------------------|
| U1 | ATtiny84 | |
| U2 | RA-01 ou RA-02 | |
| D1 | 1N4448 | Ou équivalent |
| Q1 | BS170 | Ou 2N7000, IRFZ44, etc |
| Q2 | L78L33 | |
| R1 | 220 | |
| R2 | 22K | |
| RJ3 | 4.7K si DS18B20, ou valeur adaptée si pont | optionnel |
| RJ2 | valeur adaptée si pont | optionnel |
| K1 | HFD4 / 5v | |
| REG1 | R-78E5.0-1.0 | Optionnel si Alim 5V |
| J1,2,3,4 | Bornier à visser | Pas 5.08 |
| H1 | Connecteur Dupont femelle | Optionnel |
| ICSP | Dupont (ou pince) | Optionnel |

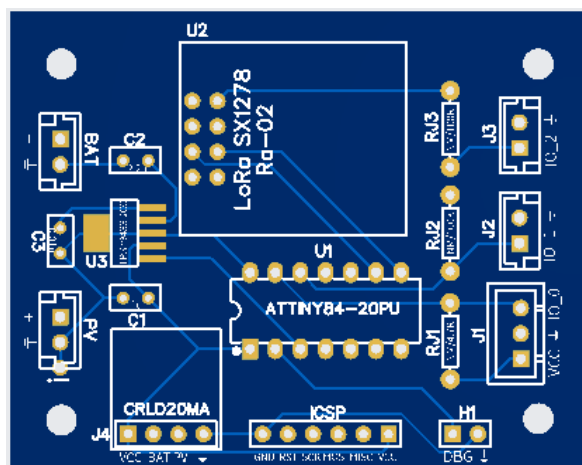


Taille : 66.2 x 48.26 mm

MLA30 (v1.1)

| | | |
|--------|--|------------------------------|
| U1 | ATtiny84 | 5v |
| U2 | RA-01 ou RA-02 | Au dos de la carte |
| U3 | TPS79433DCQ | |
| C1 | 1 μ F | optionnel si pas de parasite |
| C2 | 2.2 μ F | optionnel si pas de parasite |
| C3 | 0.01 μ F (10nF) | |
| RJ1 | 4.7K si DS18B20, ou valeur adaptée si pont | optionnel |
| RJ2, 3 | valeur adaptée si pont | optionnel |
| BAT | JST XH | Pas 2.54 |
| PV | JST XH (ou connecteur à visser*) | Pas 2.54 (ou 5.08) |
| J1,2,3 | Bornier à visser | Pas 5.08 |
| J4 | CRLD20MA 3.3V | |
| H1 | Connecteur Dupont femelle | optionnel |
| ICSP | Dupont mâle (ou pince) | optionnel |

* Le connecteur PV peut être un JST XH ou à visser 2.54, ou 5.08 en utilisant la pastille « ! »



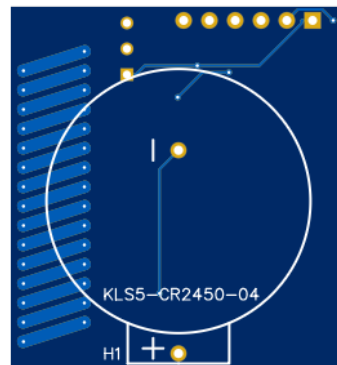
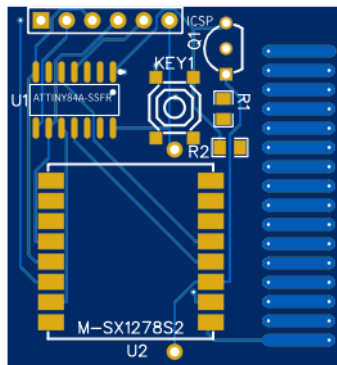
Taille : 58.4 x 45.7 mm

MLA10-1 (projet futur)

Projet de télécommande 1 bouton (ou contact sec)

Le point bloquant est l'antenne PCB : bonne forme et longueur

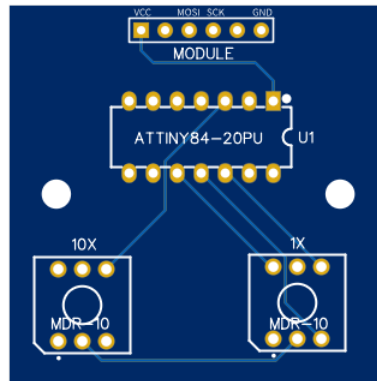
| | | |
|------|------------------------|--------------------------|
| U1 | ATtiny84-SSFR | SMD |
| U2 | SX1278S2 | SMD |
| Q1 | BS170 | Ou 2N7000, IRFZ44, etc |
| R1 | 220 | R0805 |
| RJ2 | 10K | R0805 |
| KEY1 | K2-3.6x6.1 | SMD (ou contact externe) |
| H1 | Socle pile CR2450 KLS5 | |
| ICSP | Dupont mâle (ou pince) | |



Taille : 33 x 35mm

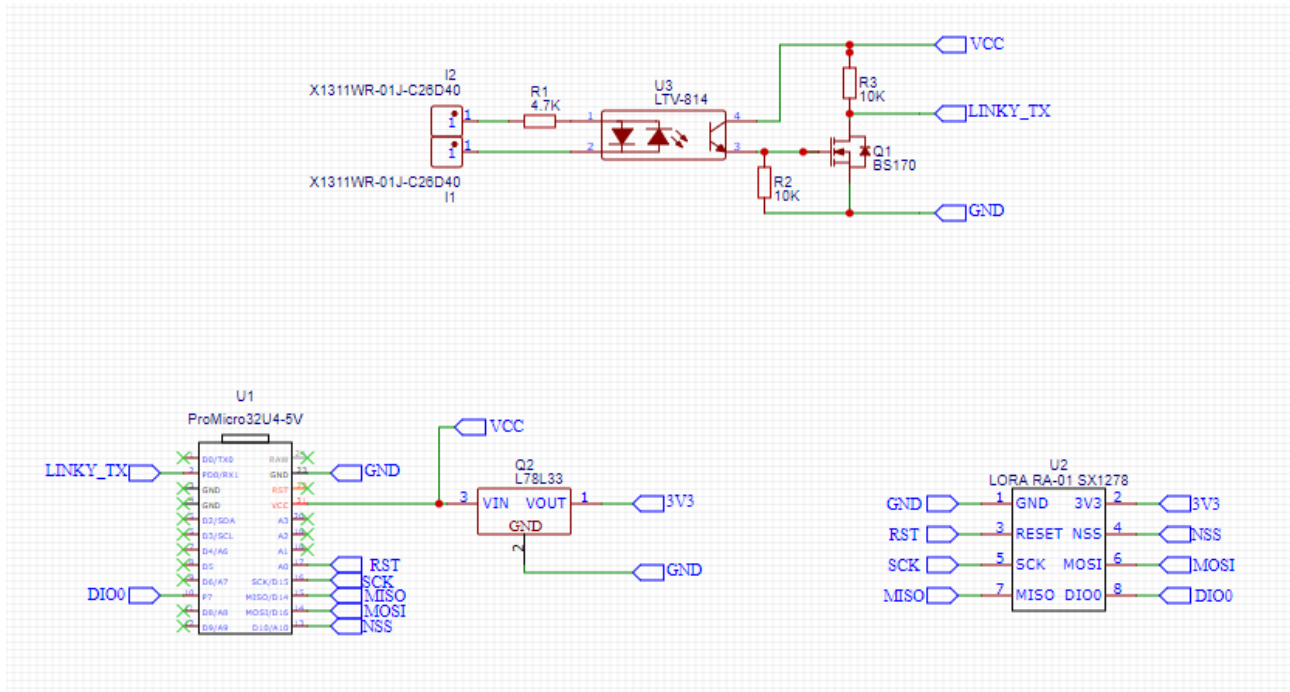
ML99

| | | |
|---------|-------------------------------------|-----|
| U1 | ATtiny84 | |
| 10X, 1X | Encodeur rotatif 0-9 (10 positions) | DIP |
| MODULE | Connecteur Dupont Mâle | |

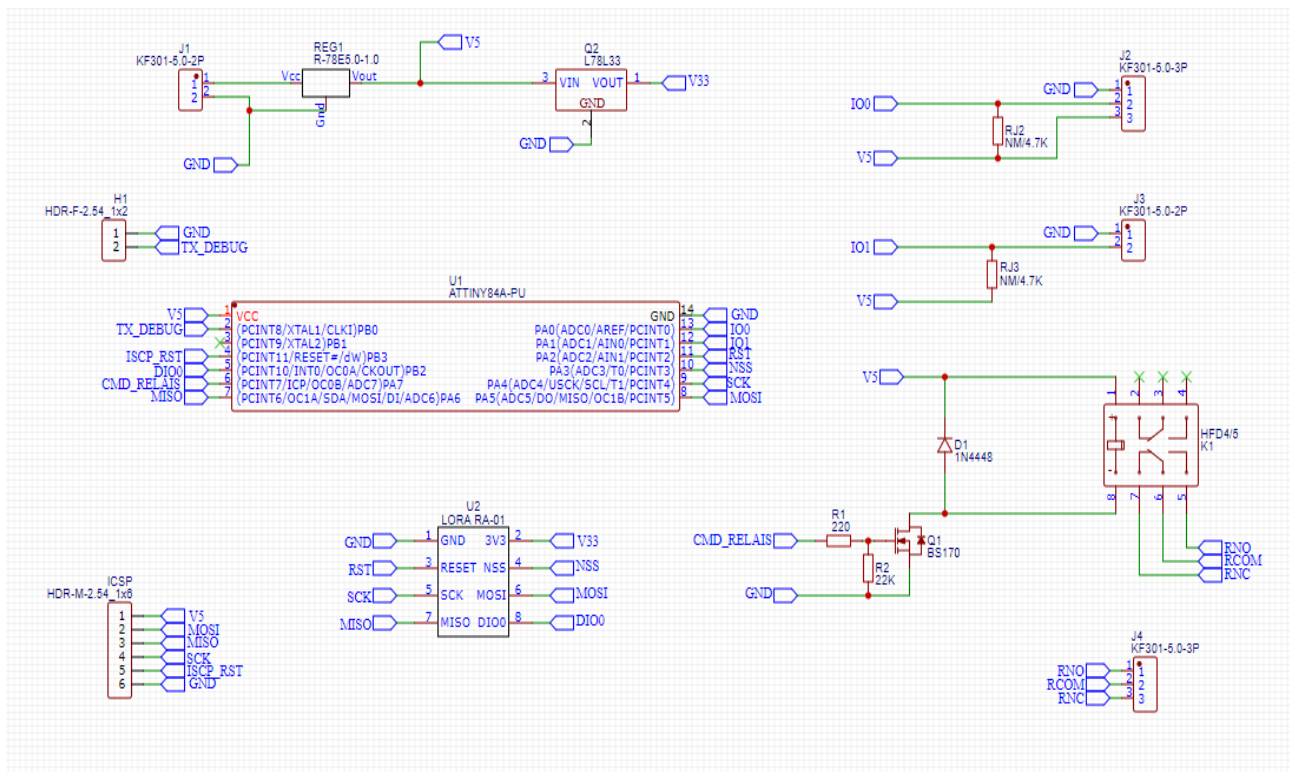


ANNEXE B : Schémas

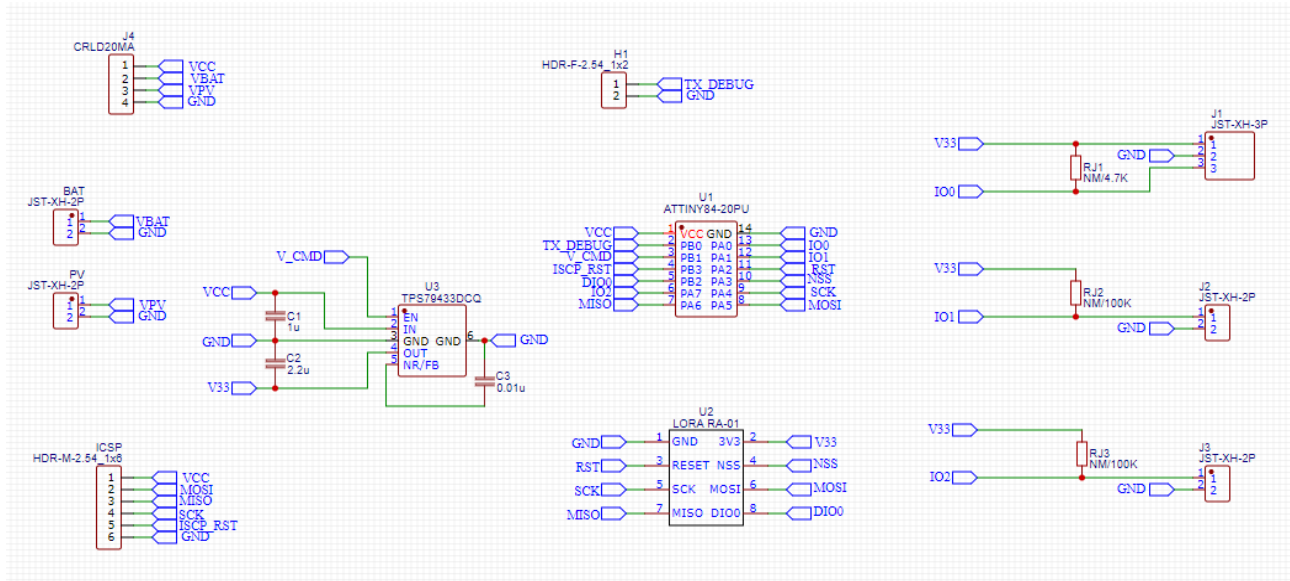
MLD01



MLE21



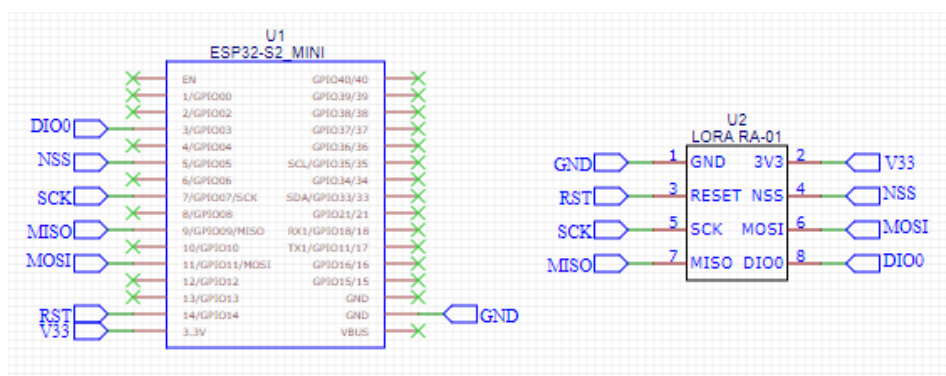
MLA30



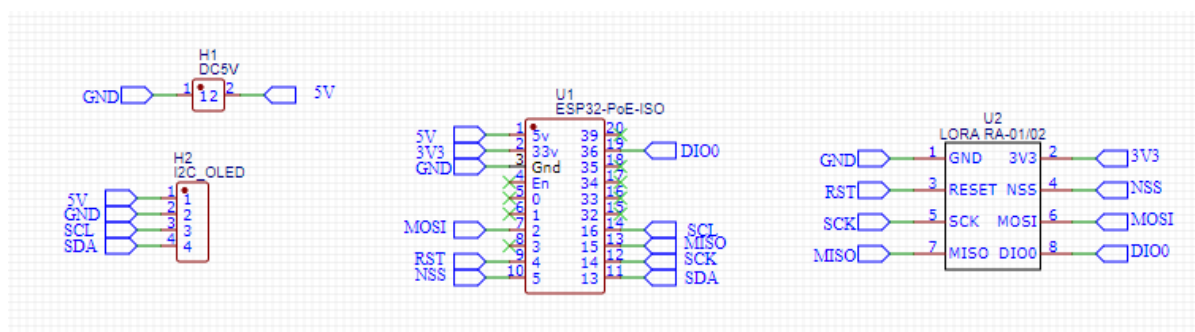
MLA10

Soon...

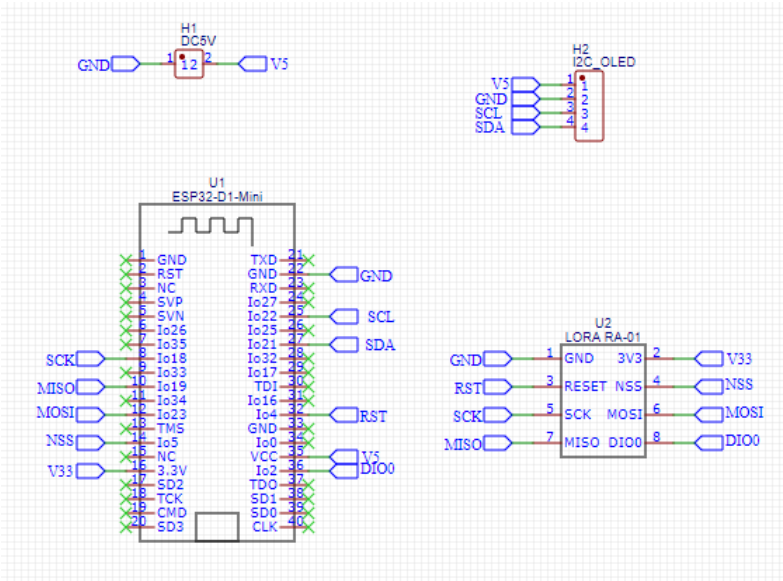
MLH1



MLH2



MLH4



ML99

