



LoRa2HA

MLE31

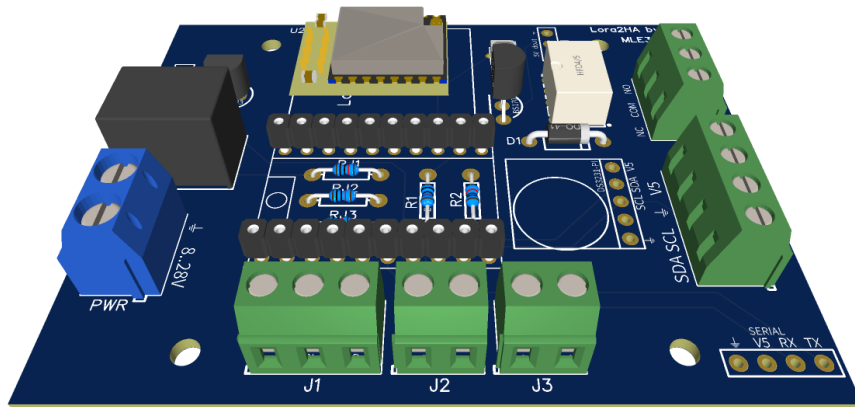


Table des matières

1	Description.....	3
2	Caractéristiques techniques.....	3
2.1	Entrées capteurs.....	4
2.2	Sortie contact sec.....	4
2.3	Alimentation.....	4
2.4	Module LoRa.....	4
2.5	LED.....	4
2.6	RTC.....	5
2.7	UPDI.....	5
3	Montage de la carte.....	5
3.1	Schéma.....	6
3.2	Nomenclatures.....	7
4	Programmation.....	8
5	Utilisation.....	8

2.1 Entrées capteurs

La carte comporte 5 connecteurs à vis, permettant de brancher les différents capteurs.

- **J1** : capteur binaire ou analogique
- **J2, J3** : capteur binaire ou analogique, avec alimentation 3.3 V
- **J4** : capteur I2C
- **J5** : alimentation 5 V pour capteurs.

Les entrées J1, J2 et J3 sont équipées d'un « pont résistif » pour certain capteur analogiques, les valeurs des résistances sont de 4.7KΩ.

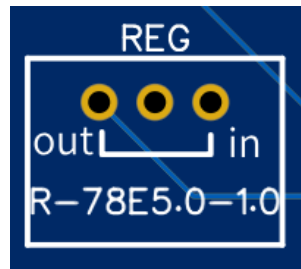
2.2 Sortie contact sec

Le connecteur **OUT1** fourni un contact sec, il peut aussi permettre de piloter certains équipements avec une très faible puissance : < 1 A.

2.3 Alimentation

Le module est alimenté par le connecteur **PWR**.

Si l'alimentation est directement du 5 VDC, il n'est pas utile de monter le régulateur **REG**, mais il faut faire un « shunt » en reliant les deux pastilles indiquées sur le PCB (out et in) comme indiqué ci-dessous.



Lorsque le régulateur **REG** est monté, il accepte une tension d'entrée comprise entre 8 et 28 VDC.

☒ Ce régulateur ne peut fournir que 1 A, il est donc conseillé de faire un bilan énergétique des tous les équipements qui pourraient être connectés au module : capteur, actionneur, led, rtc.

2.4 Module LoRa

Le module LoRa est un RA-02, en 433MHz, il est conseillé de lui connecter une antenne adaptée à la distance souhaitée ; généralement une antenne supérieure à 3 dBi permet de couvrir une distance d'au moins 500m.

2.5 LED

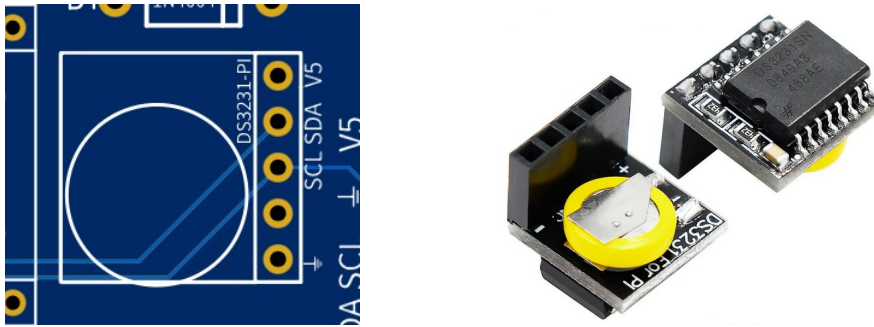
La led **LED1** permet de contrôler le fonctionnement du module. C'est une led adressable de type WS2812 et

il est possible d'étendre le nombre de led en utilisant le connecteur **H1**.

⊠ Attention à ne pas dépasser la limitation en courant imposé par le régulateur. Si l'alimentation est directement du 5V, il faudra dimensionner correctement celle-ci.

2.6 RTC

Le connecteur **DS3231-PI** permet de monter une horloge temps réel (RTC) afin que le module puisse être autonome lors d'automatisation basée sur l'horodatage.



→ Astuce : ce même connecteur peut aussi être utilisé pour gérer un afficheur I2C, ou tout autre dispositif sur bus I2C.

2.7 UPDI

C'est le connecteur qui permet la programmation du module, référez vous au chapitre « Programmation ».

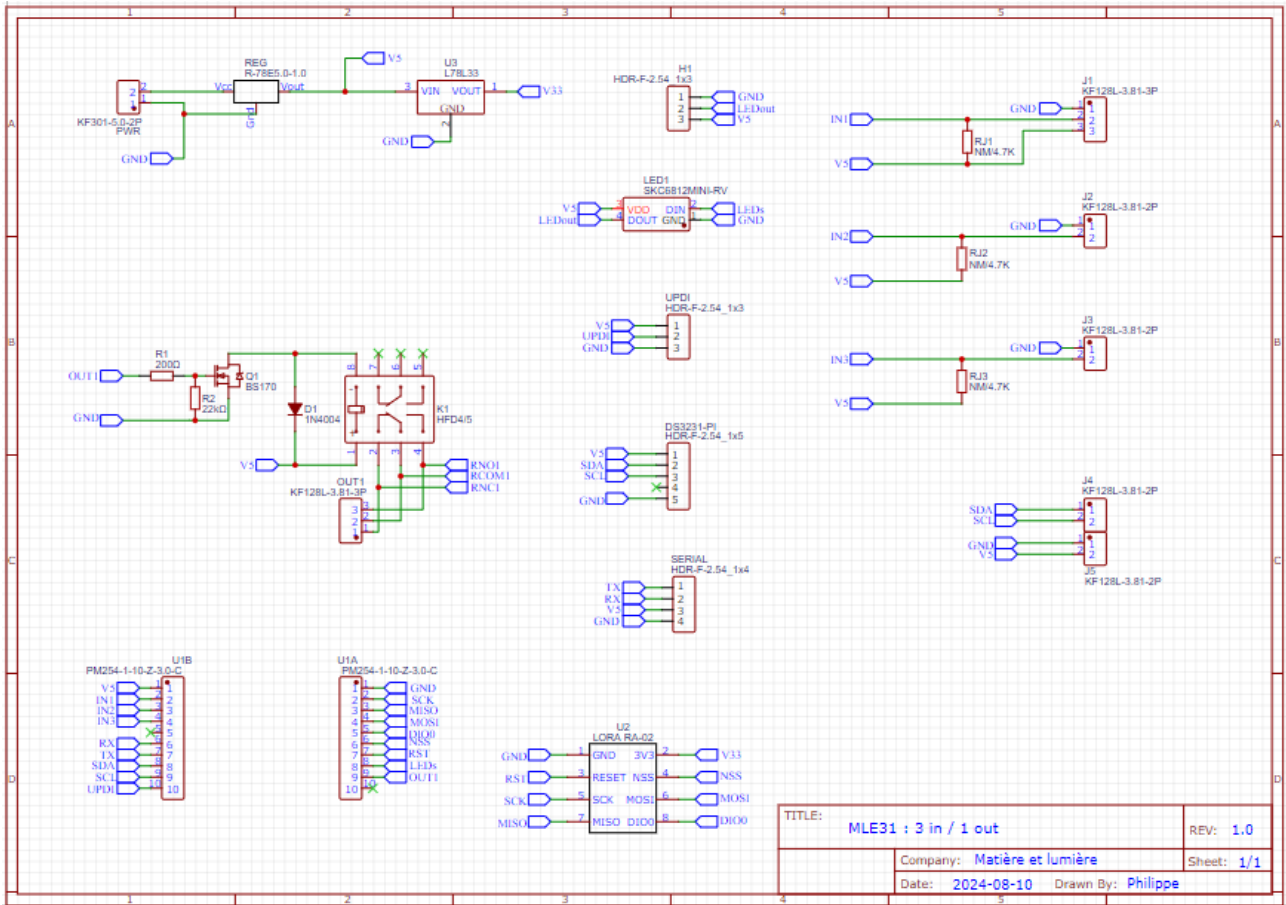
⊠ Il faut débrancher l'alimentation principale lors de la programmation.

3 Montage de la carte

Il est conseillé de souder en premier les composants CMS puis ensuite les traversants.

⊠ La carte processeur MLT3216 peut être soudée directement mais il est conseillé de la mettre sur support afin de faciliter la programmation de celle-ci lorsque le montage est complet et en production.

3.1 Schéma



3.2 Nomenclatures

Repère	Valeur/référence	Qté	Commentaire
D1	1N4004	1	Ou équivalent
DS3231-PI	Pin header femelle 2.54 1x5	1	
J1,OUT1	KF128L-3.81-3P	2	
J2,J3,J4,J5	KF128L-3.81-2P	4	
K1	HFD4/5	1	
LED1	SKC6812MINI-RV	1	
REG	R-78E5.0-1.0	1	Version 0.5A possible
PWR	KF301-5.0-2P	1	
Q1	BS170	1	
R1	200Ω	1	
R2	22kΩ	1	
RJ1, RJ2, RJ3	4.7kΩ	3	Adaptable selon besoin
U1A,U1B	Pin header Femelle 2.54 1x10	2	
U3	L78L33	1	
U2	Module LoRa RA-02 (connecteur 2x4)	1	

4 Programmation

La programmation utilise « MegaTinyCore », un projet Github prévu pour tous les microprocesseurs de la gamme Mega/Tiny de la série 0/1/2.

<https://github.com/SpenceKonde/megaTinyCore>

Dans le projet on trouve toute la documentation afin de transformer un simple Arduino (Uno, Nano) en programmeur UPDI.

Toutes les bibliothèques utiles sont incluses dans le package MegaTinyCore ou l'IDE Arduino, sauf la librairie dédiée à LoRa2HA : **Radiolink** (version > 2.0 obligatoire).

<https://github.com/PM04290/RadioLink>

☒ Si vous avez installé une ancienne version (< 2.0), il faut la re-télécharger.

La bibliothèque **MLiotElements** permet de gérer facilement tous les capteurs/actionneurs des modules. Elle n'est pas indispensable si vous créez votre propre programme, mais obligatoire si vous recompilez les firmwares par défaut.

Depuis la version 1.0 des HUB, ces derniers permettent la programmation des ATtiny3216 à l'aide du connecteur UPDI, voir le chapitre « Programmation UPDI » de la documentation principale.

5 Utilisation

Avec les Firmwares fournis dans le projet, lors de la première utilisation, la carte MLT3216 va s'initialiser avec des valeurs par défaut :

- Numéro (UID) = 40
- Numéro du Hub (HUID) = 0

Pour appairer le module avec le HUB, il faut :

1. maintenir le bouton CFG lors de la mise sous tension ; la LED va clignoter (1Hz)
2. relâcher le bouton CFG avant 10s (si vous tenez appuyé durant plus de 10s, les données internes vont être ré-initialisées)
3. la LED va s'allumer durant la transmission des informations vers le HUB (il doit être en mode « Active pairing »)
4. après transmission, la LED clignote lentement
5. lors de la fin du paramétrage par le HUB (Finish Pairing), le module passe en mode « normal », la LED s'éteint.