OTMetric
5 Avenue Georges Bataille
60330 Le Plessis Belleville

LW8 Messages & Commandes

Version 1.00

Siège social: ALTO METERING SAS - 5 avenue Georges Bataille - 60330 Le Plessis Belleville, France
Téléphone: +33 3 60 80 00 10 - www.alto-metering.com - contact@alto-metering.com
Société par actions simplifiée (SAS) - Capital de 10 000 € - SIRET: 80177319300017
NAF-APE: 4791B - RCS/RM: B 801 773 193 - numéro EORI: FR80177319300017 - Numéro TVA: FR18801773193

Authors

Nom	Chapitres, Sections	Date Initiale	Visa
OM	Création	10-juil.-25	OM

History

Version	Initiales	Date	Description	Section
1.00	OM	10-juil.-25	Création	All

Table des matières

1. Description des messages

Tous les messages LoRaWAN™ sont composés de données de service et d'un ensemble d'octets, appelé « PAYLOAD », définis par le fabricant.

Dans le mode de fonctionnement standard, le capteur LW8 envoie des messages « non confirmés » (en utilisant la terminologie LoRaWAN™) à intervalles réguliers dépendant du paramètre « RF Period ». Par défaut, le capteur LW8 enverra un message « confirmé » tous les 12 messages pour vérifier la connexion avec le serveur distant (ceci correspond à deux fois par jour avec la périodicité radio par défaut de 60 minutes).

Ces messages sont envoyés sur le port FPORT 2 du serveur de réseau LoRaWAN™ et la partie PAYLOAD est définie comme suit :

Si le type de capteur < 16

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10
Type capteur		Statut	Valeur 1				Valeur 2 (optionnelle)			

avec:

- *Type capteur* – Entier SHORT 16 bits (Octet de poids faible en premier) représentant le type de capteur LW8 comme défini par le fabricant (ex. 1 = LW8.PC1 - Compteur d'impulsions 1 entrée).
- *Statut* – Une valeur composite bit à bit représentant l'état du capteur:
  - 128 – N/A (Non Applicable)
  - 64 – Erreur de Communication: le câble de connexion est coupé et le LW8 ne peut pas communiquer avec le compteur (pour les impulsions) ou avec son capteur physique (température, hygrométrie, etc.)
  - 32 – Erreur Capteur: Erreur générique (non définie)
  - 16 – Erreur temporaire: Erreur générique qui s'auto-corrige dans le temps (non définie)
  - 8 – Erreur Permanente: Erreur générique du capteur qui requiert une intervention (non définie)
  - 4 – Pile faible: La pile du capteur est faible
  - 2 – N/A
  - 1 – N/A
- *Valeur 1* – Entier LONG 32 bits (Octet de poids faible en premier) représentant la première valeur mesurée
- *Valeur 2* – Entier LONG 32 bits (Octet de poids faible en premier) représentant la seconde valeur mesurée si applicable. **Note:** Ce champ est omis si une seule valeur est mesurée par le capteur.

Si type de capteur >= 16

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Type capteur		Statut	Valeur 1		Valeur 2 (optionnelle)	

avec:

- *Type capteur* – Entier SHORT 16 bits (Octet de poids faible en premier) représentant le type de capteur LW8 comme défini par le fabricant (ex. 1 = LW8.PC1 - Compteur d'impulsions 1 entrée).
- *Statut* – Une valeur composite bit à bit représentant l'état du capteur:
  - 128 – N/A (Non Applicable)
  - 64 – Erreur de Communication: le câble de connexion est coupé et le LW8 ne peut pas communiquer avec le compteur (pour les impulsions) ou avec son capteur physique (température, hygrométrie, etc.)
  - 32 – Erreur Capteur: Erreur générique (non définie)
  - 16 – Erreur temporaire: Erreur générique qui s'auto-corrige dans le temps (non définie)
  - 8 – Erreur Permanente: Erreur générique du capteur qui requiert une intervention (non définie)
  - 4 – Pile faible: La pile du capteur est faible
  - 2 – N/A
  - 1 – N/A
- *Valeur 1* – Entier SHORT 16 bits (Octet de poids faible en premier) représentant la première valeur mesurée
- *Valeur 2* – Entier SHORT 16 bits (Octet de poids faible en premier) représentant la seconde valeur mesurée si applicable. **Note:** Ce champ est omis si une seule valeur est mesurée par le capteur.

Exemple de script de décodage:

```
function Decoder(bytes, port) {
  var decoded = {};
  if (port == 2) {
    decoded.deviceType = bytes[0] + (bytes[1] * 256);
    decoded.status = bytes[2];
    if (decoded.deviceType >= 16) {
      decoded.value1 = bytes[3] + (bytes[4] * 256);
      if (decoded.value1 >= 32768) decoded.value1 -= 65536;
      if (bytes.length > 5) {
        decoded.value2 = bytes[5] + (bytes[6] * 256);
        if (decoded.value2 >= 32768) decoded.value2 -= 65536;
      }
    } else {
      decoded.value1 = (bytes[3] + (bytes[4] * 256) + (bytes[5] * 256 * 256)  +
(bytes[6] * 256 * 256 *256));
      if (bytes.length > 9) {
        decoded.value2 = bytes[7] + (bytes[8] * 256) + (bytes[9] * 256 * 256)  +
(bytes[10] * 256 * 256 *256);
      }
    }
  }
  return decoded;
}
```

2. Messages de service

Le capteur LW8 répond à des commandes de service envoyées et reçues par le serveur de réseau LoRaWAN™ sur le port FPORT 3.

Message de service	Taille	Description
GetStatus = 0,	Octet	Etat du capteur
SetRFPeriod=1,	Short	Définit la périodicité en min. des envois de messages radio
GetRFPeriod=2,	Short	Renvoie la périodicité en min. des envois de messages radio
SetConfirmed=3,	Octet	Définit le nombre de messages 'non-confirmés' envoyés avant l'envoi de messages 'confirmés'
GetConfirmed=4,	Octet	Renvoie le nombre de messages 'non-confirmés' envoyés avant l'envoi de messages 'confirmés'
GetNbIndexes=5,	Octet	Renvoie le nombre d'index gérés par ce capteur
SetMaxRetries=6,	Octet	Définit le nombre maximum de ré-essais pour les messages LoRaWAN™ 'confirmés'
GetMaxRetries=7,	Octet	Renvoie le nombre maximum de ré-essais pour les messages LoRaWAN™ 'confirmés'
SetResetFlag=8,	Octet	Active ou désactive la fonction 'Reset' avec l'aimant
GetResetFlag=9,	Octet	Renvoie 1 si la fonction 'Reset' avec l'aimant est désactivée
GetBatteryLevel=10,	Short	Renvoie la tension de la pile en mV
GetMaxPayload=11,	Octet	Renvoie la taille maximale du PAYLOAD d'un message LoRaWAN™
SyncDateTime=12,	Aucun	Synchronise la Date et l'Heure avec le serveur distant
SetSyncDateTimeDelay=13,	Octet	Définit le délai (en jours de 0 à 48) entre deux demandes de synchronisation de la Date et l'Heure avec le serveur distant
GetSyncDateTimeDelay=14,	Octet	Renvoie le délai (en jours de 0 à 48) entre deux demandes de synchronisation de la Date et l'Heure avec le serveur distant
GetLastValues=16,	Aucun	Demande le renvoi des dernières valeurs dans le message standard
GetMaximumArchivedValues =17,	Short	Renvoie le nombre maximum de valeurs stockées dans l'historique
GetArchivedValues=18,	Voir 2.1	Demande l'envoi de valeurs stockées dans l'historique
SetPulseInOffset=22,	Long	Définit le nombre d'impulsions à ajouter à la valeur actuellement stockée. **Note:** Envoyer deux valeurs si le capteur a deux entrées.
SetOutputPulseLevel=23,	Octet	Définit le niveau de la sortie (pour le modèle switch uniquement)
GetOutputPulseLevel=24,	Aucun	Renvoie le niveau de la sortie (pour le modèle switch uniquement)
SetAlarmMode=25,	Octet	Active le mode alarme (pour le modèle state uniquement)
GetAlarmMode=26,	Aucun	Renvoie le mode alarme (pour le modèle state uniquement)
OpenOutput=30,	Octet	Ouvre le relais (pour le modèle switch uniquement)
CloseOutput=31,	Octet	Ferme le relais (pour le modèle switch uniquement)
GetOutputState=32,	Octet	Renvoie l'état du relais (ouvert/fermé) (pour le modèle switch uniquement)
ToggleOutput=33,	Short * 3	Bascule l'état du relais (fermé->ouvert) x fois (pour le modèle switch uniquement)
TriggerOutput=34,	Short, Octet, Long	Bascule l'état du relais (fermé->ouvert) suivant un nombre d'impulsions en entrée (pour le modèle switch uniquement)

FactoryReset=128,	Aucun	RàZ du capteur en mode usine -> **Arrêt de toute communication**
DeviceReset=129	Aucun	RàZ du capteur. Le capteur renvoie un 'join' au serveur distant
GetDeviceVersion=130,	Aucun	Renvoie la version actuelle du firmware du capteur

A chaque « Set » d'un paramètre reçu par le capteur, la valeur actualisée est renvoyée en réponse par ce dernier.

Le PAYLOAD d'un message de service a le format suivant :

Octet 0	Octet 1	...	Octet n
Commande	Valeurs		

La longueur du message dépend du type et du nombre de données transmises. Les valeurs de type entier 16 ou 32 bits sont émises avec l'octet de poids faible en premier.

NOTE: Certains messages de service ne sont disponibles que pour certaines versions du capteur LW8.

2.1. Message Get Archived Values

2.1.1. Requête

Octet 0	Octet 1	Octet 2	Octet 3
18	Début		Nombre

- *Début* – Représente l'enregistrement de départ requis: 0 = valeur la plus récente, 1 = valeur précédente, etc.
- Nombre – Nombre de valeurs à retourner.

Note: Si « début » est plus grand que le nombre de valeurs actuellement stockées, aucune valeur n'est retournée. Le nombre de valeurs retournées est automatiquement ajusté suivant le nombre d'enregistrements disponibles ou la taille maximale autorisée par la transmission LoRaWAN™.

2.1.2. Réponse

Si type de capteur < 16:

Octet 0	Octet 1	Octet 2	Octet 3	Octet 4	Octet 5	Octet 6	Octet 7	...	Octet n-3	Octet n-2	Octet n-1	Octet n
Commande	Type capteur		Statut	Valeur 1				...	Valeur n			

If Type de capteur >= 16:

Octet 0	Octet 1	Octet 2	Octet 3	Octet 4	Octet 5	...	Octet n-1	Octet n
Commande	Type capteur		Statut	Valeur 1		...	Valeur n	

Exemple de script de décodage d'un historique de valeurs:

```
function Decoder(bytes, port) {
  var decoded = {};  var i;
  if (port ==3) {
    decoded.function = bytes[0];
    if (bytes[0] == 18) {
      decoded.deviceType = bytes[1] + (bytes[2] * 256);
      decoded.status = bytes[3];  decoded.values = {};
      if (decoded.deviceType >= 16) {
        for (i=0; (i+4) < bytes.length; i+=2) {
          decoded.values[i/2] = bytes[4+i] + (bytes[5+i] * 256);
          if (decoded.values[i/2] >= 32768) decoded.values[i/2] -= 65536;
        }
      } else {
        for (i=0; (i+4) < bytes.length; i+=4)
          decoded.values[i/4] = bytes[4+i] + (bytes[5+i] * 256) + (bytes[6+i] *
256 * 256)  + (bytes[7+i] * 256 * 256 *256);
      }
  } else {
      decoded.result = bytes[1];
      if (bytes.length > 2) decoded.result += bytes[2] * 256;
      if (bytes.length > 4) decoded.result += (bytes[3] * 256 * 256) + (bytes[4]
* 256 * 256 * 256);
    }
  return decoded;
}
```