



CHEAP'EAU

SOLUTIONS INNOVANTES A BAS COUT POUR LE SUIVI DES SYSTEMES DE GESTION DES EAUX URBAINES

Durée du projet : 2020-2022



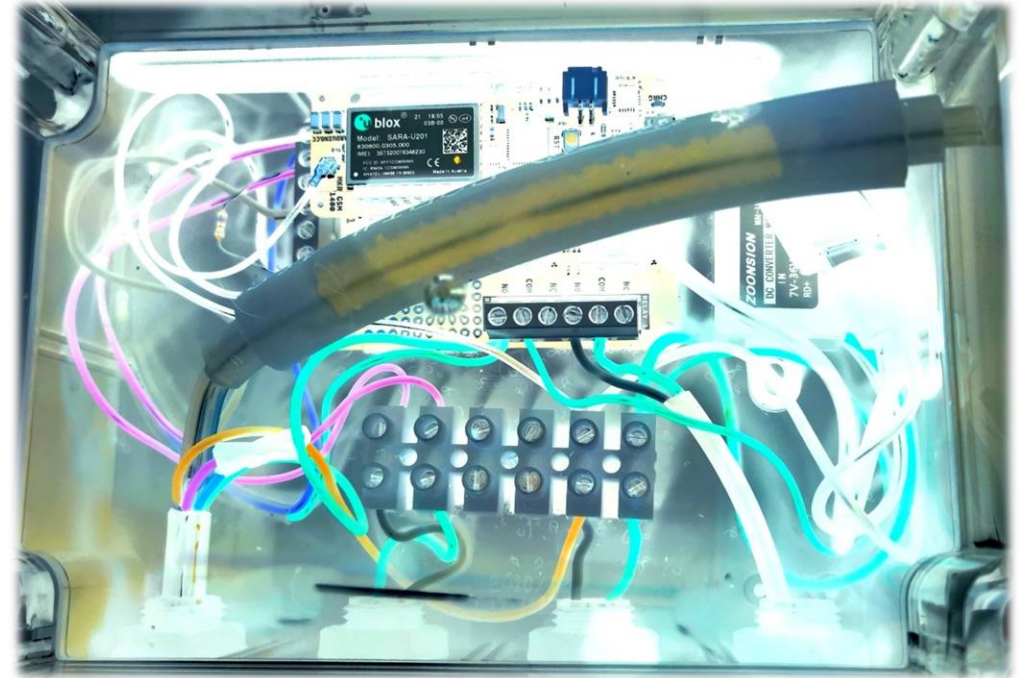
Sommaire

- Le projet
- Choix et développement technique
 - Mise en situation
 - Gestion des données
 - Conclusions

Le projet

Projet Cheap'Eau : Objectifs à atteindre

- Abordable
- Robuste et discret
- Autonome
- Performances
- Connectés
- Simple à construire et à prendre en main



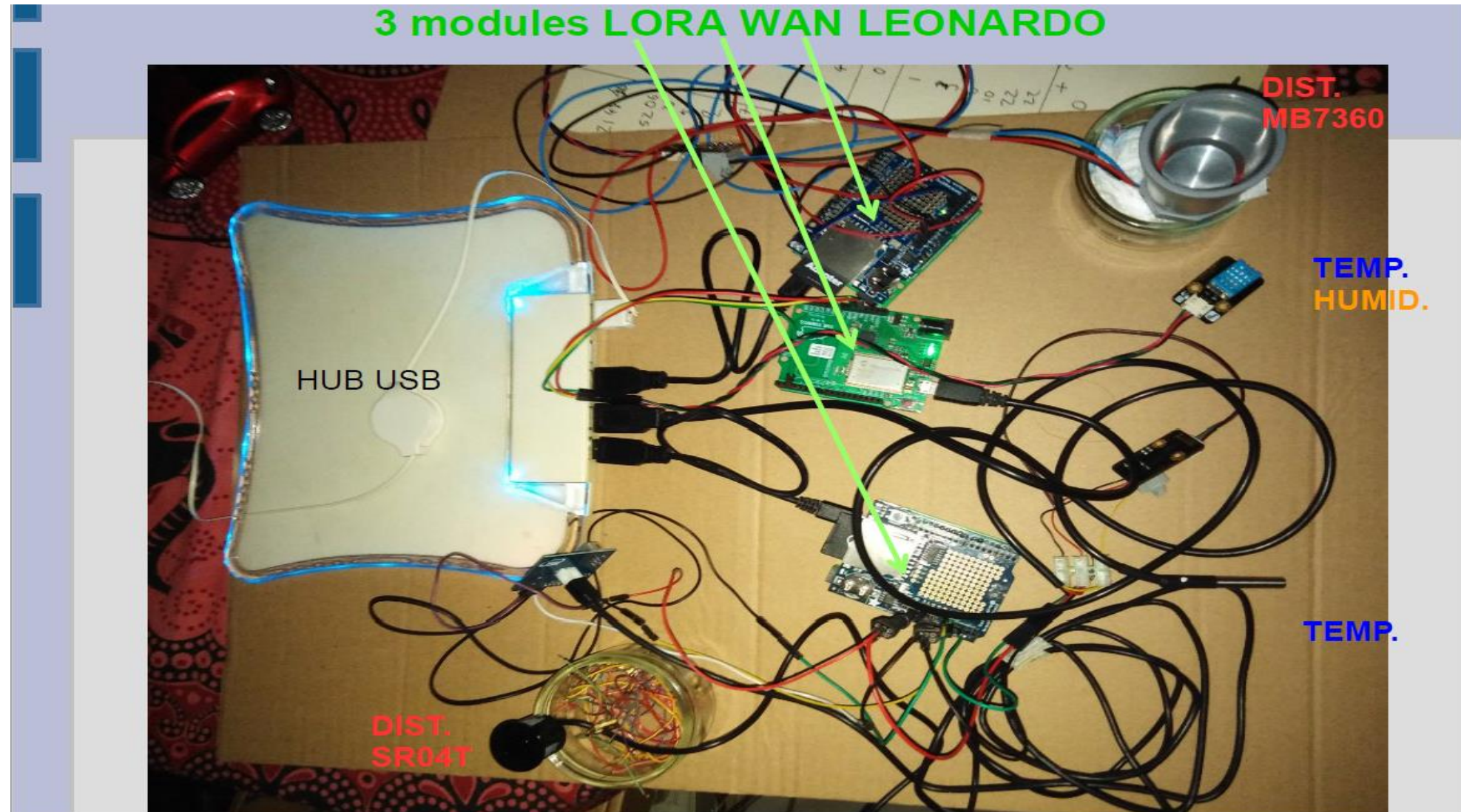
Projet Cheap'Eau : cahier des charges

- Utiliser des capteurs bon marché (< 10-15 € / unité)
- Fabriqués en série pour l'industrie, les transports ou la maison, et immédiatement disponibles dans le commerce
- Panneaux solaires/batterie
- Encapsulage simple
- Les adapter aux matériels informatiques et logiciels libres des communautés d'utilisateurs Arduino[®]
- Réseau IoT : Lora ou SigFox IoT
- Définir les besoins avec les gestionnaires, BE et citoyens (associations) et assurer le transfert de ces systèmes

Les étapes du projet

- **Choix des variables à mesurer** : quels besoins, pour quels objectifs? collecte des attentes des opérationnels, choix des systèmes (AEGIR)
- **Choix des systèmes et leurs développements** des solutions techniques à bas coûts les plus adaptées aux objectifs
- **Tests en milieux contrôlés** (laboratoire, bungalow OTHU)
- **Évaluation sur le terrain** en comparaison avec les systèmes traditionnels
- **Transfert des résultats** et valorisation opérationnelle

Choix des variables à mesurer



Choix et développement technique

Lora ou Sigfox?

Technology	LoRa	Sigfox
Bandwidth	very low	
Range	few kms!	
Frequency	unlicensed	
Frequencies [Mhz]	415, 868, 902, 915, 923	868, 902, 920, 923
Antenna / gateway	public / private	private
Coverage	crowdsourced	one company / country
Subscription	free	15-30 € / year
Power consumption	very low	
Complexity (programming)	medium	
Peer-to-peer communication	possible	impossible
Two-ways communication	yes but very limited	
Security	medium	medium or high?

Choix du type de carte Arduino Lora

TTN-UN-868



Leonardo

Microcontrôleur : Atmega 32U4

Mémoire flash 32ko

Horloge: 16 Mhz

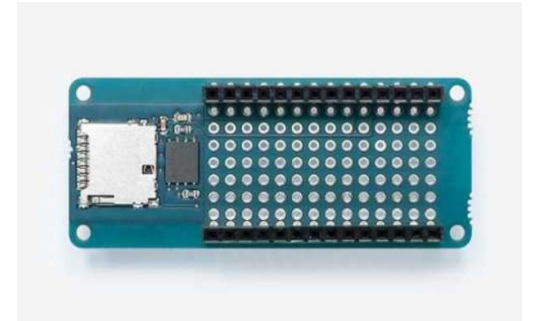
Mkr WAN 1300 (connectivité Lora) Arduino



Microcontrôleur : Atmel SAM D21

Mémoire flash : 256 ko

Horloge : 48 Mhz



Choix du capteur de mesure de distance

Sensor: JSN-SR04T



Plage de mesure : de 25cm à 4.5m
Précision: 0.5cm

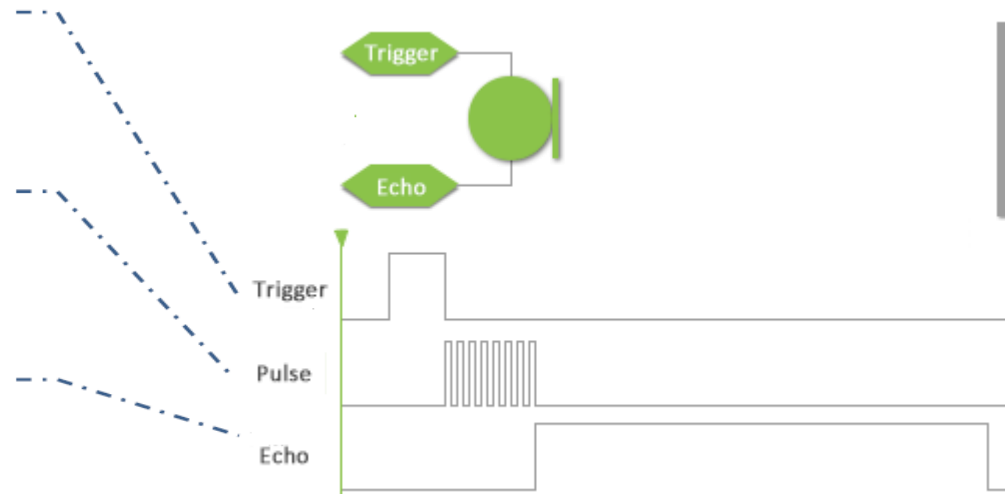
Principe de la mesure

Ultrasonic ranging principle

First send out a 20 μ S pulse to emit sonic burst

module automatically send eight 40khz square wave

ECHO = 1 for the duration of the sound wave to come and go



Distance [m] = duration [μ s] / 2 * C [m/s] / 1E6 (Panda *et al.*, 2016)

$$C_{approx} = (331.296 + \theta * 0.606) * \left(1 + (RH * 9.604 * 10^{0.032 * (\theta - 0.004 * \theta^2) - 6})\right)$$

θ in $^{\circ}$ C [0,50]

RH in % [0,100]

Ex: 8000 μ s @ 20 $^{\circ}$ C / 50%

$C_{approx} = 344.06$ m/s

Distance = 1.376 m

Capteur de distance (Programme)

```
// CAPTEUR DISTANCE SR04T
valeurTemp = rtc.getTemperature();
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// FIN DU SIGNAL ACOUSTIQUE
unsigned long duration = pulseIn(echoPin, HIGH);
vitesseSon = 331.5 + 0.6 * valeurTemp ;
// vitesseSonCorrigeeHumidite = 331.4 + 0.6 * valeurTemp + 0.0124 * humidity;
Distancel = duration * vitesseSon / 20000 ;

// BOUCLE DE MOYENNAGE DES VALEURS DE TEMPERATURE ET DE DISTANCE
for (int i = 0; i < 15; i++) {
    valeurTemp = (valeurTemp + rtc.getTemperature()) / 2;
    Distancel = (Distancel + (duration * vitesseSon / 20000)) / 2;
    delay(1000);
}
```

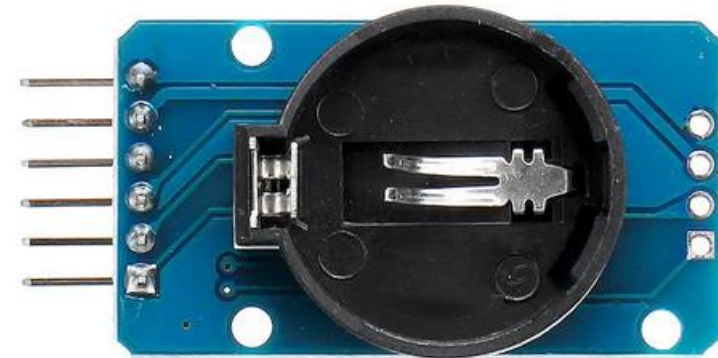
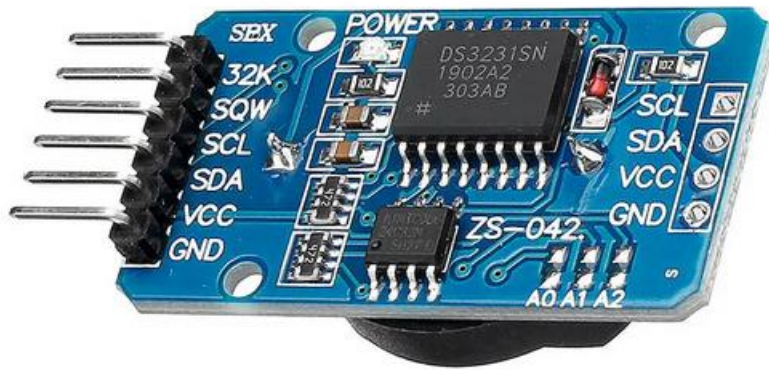
Variable d'impulsion > 10microsec

Influence humidité négligeable

Moyenne sur 15 échantillons

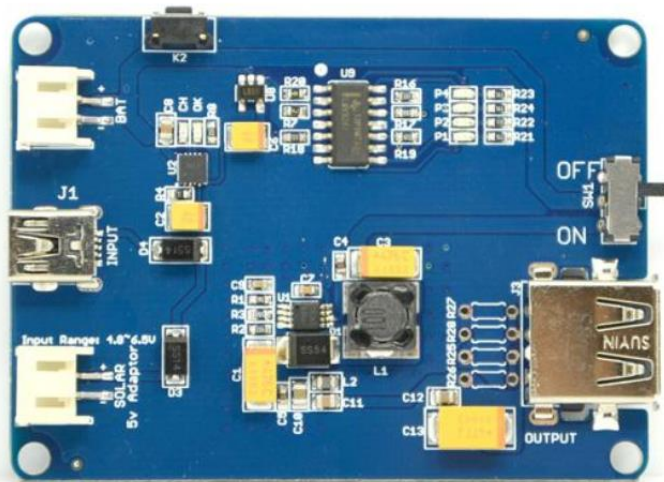
Échantillonnage toutes les secondes

Real Time Clock + Capteur de température



Module RTC DS3231 avec capteur de température
Compensée pour les dérives lié à la température

Alimentation électrique



Carte LiPo Rider Pro 106990008

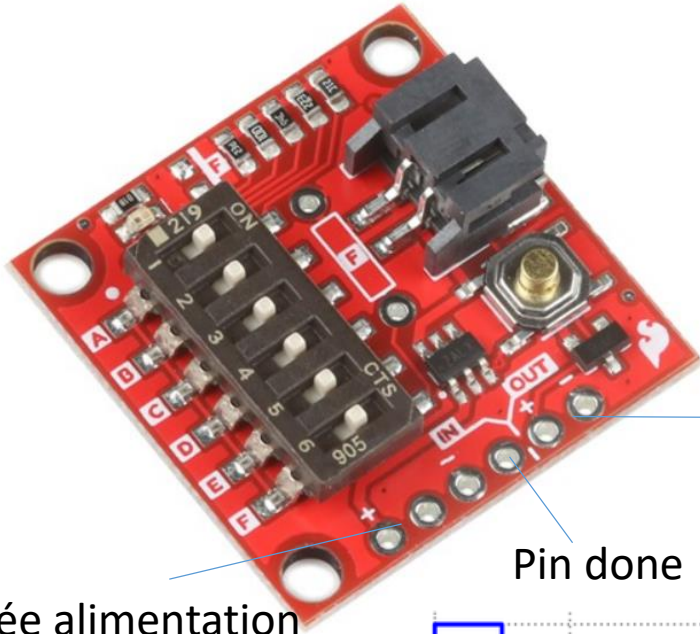


Cellule solaire SOL3W
5,5 V/540 mA - 160 x 138 mm



Batterie LiPo 3.8v 5100mAh

Nano Timer TPL5110



Entrée alimentation électrique

Pin done

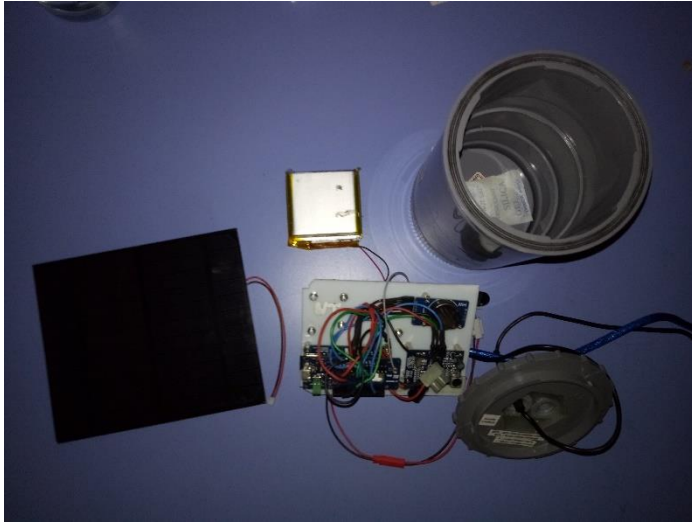
Sortie alimentation électrique



Temps réglable par commutation de switch (2 sec à 15 min)
 Consommation électrique à l'arrêt : 35nA pour 2.5v
 Précision temporelle : 1%

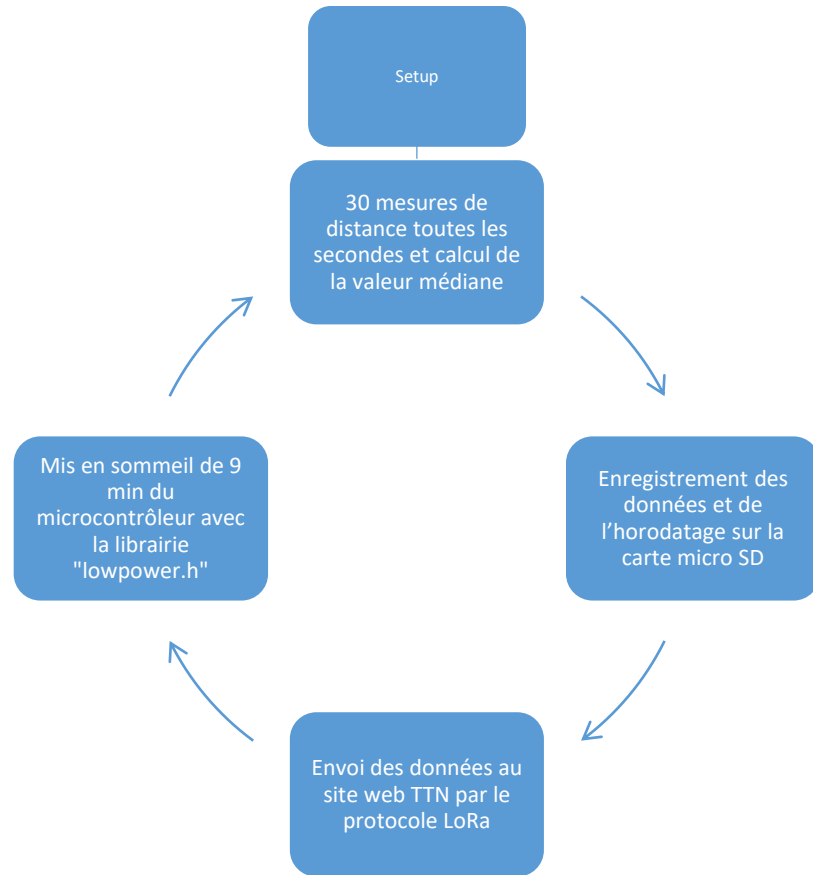
Timer	Switch Combo	Resistance
2-3 s	A+B+C+D+E	7.579 kΩ
3-4 s	A+B+C+D	7.933 kΩ
4 s	A+B+C	8.470 kΩ
5 s	A+B+E	8.844 kΩ
6 s	A+B	9.329 kΩ
10 s	A+C+D+E	11.563 kΩ
~12 s	A+C+D	12.407 kΩ
~13 s	A+C+E	12.742 kΩ
~15 s	A+D+E	13.225 kΩ
~18 s	A+C	13.774 kΩ
~19 s	B+C+D+E	14.243 kΩ
20 s	A+D	14.341 kΩ
~22 s	A+E	14.790 kΩ
~25 s	B+C+D	15.546 kΩ
~28 s	B+C+E	16.075 kΩ
32 s	B+D+E	16.852 kΩ
35 s	B+C	17.754 kΩ
40 s	B+D	18.707 kΩ
~45 s	B+E	19.479 kΩ
~5 min	C+D+E	40.400 kΩ
8 min	C+D	52.995 kΩ
~12 min	C+E	59.694 kΩ
15 min	D+E	72.033 kΩ

Prototype final

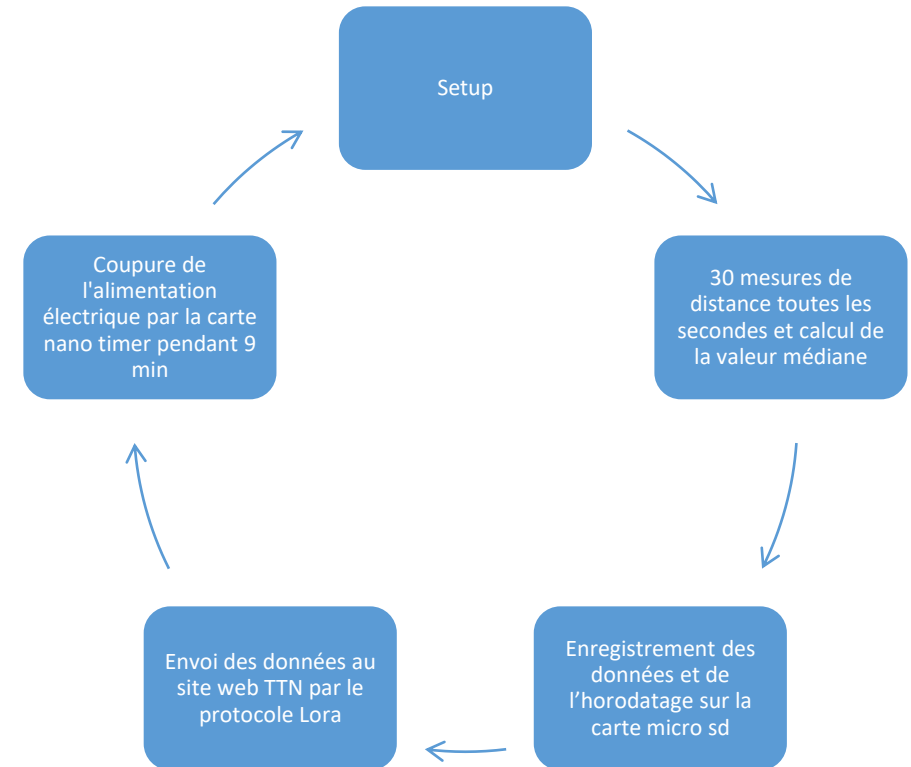


Consommation en mode standby :
Sans nano timer : 28mA
Avec nano timer : 3mA

Les programmes arduino



Sans nano timer



Avec nano timer

Coût total

• Mkr wan 1300	40.98 euro TTC
• Shield micro SD	21.90 euro TTC
• Capteur ultrason	13.51 euro TTC
• RTC DS3231	3.99 euro TTC
• Batterie LiPo 3.8v 5100mAh	29.95 euro TTC
• Cellule solaire 3Wc	19.00 euro TTC
• Carte lipo Rider Pro	15.60 euro TTC
• Encapsulage	10.50 euro TTC
Total: 155.43 euro TTC	

Mise en situation

Site d'essai



Site de Chassieu



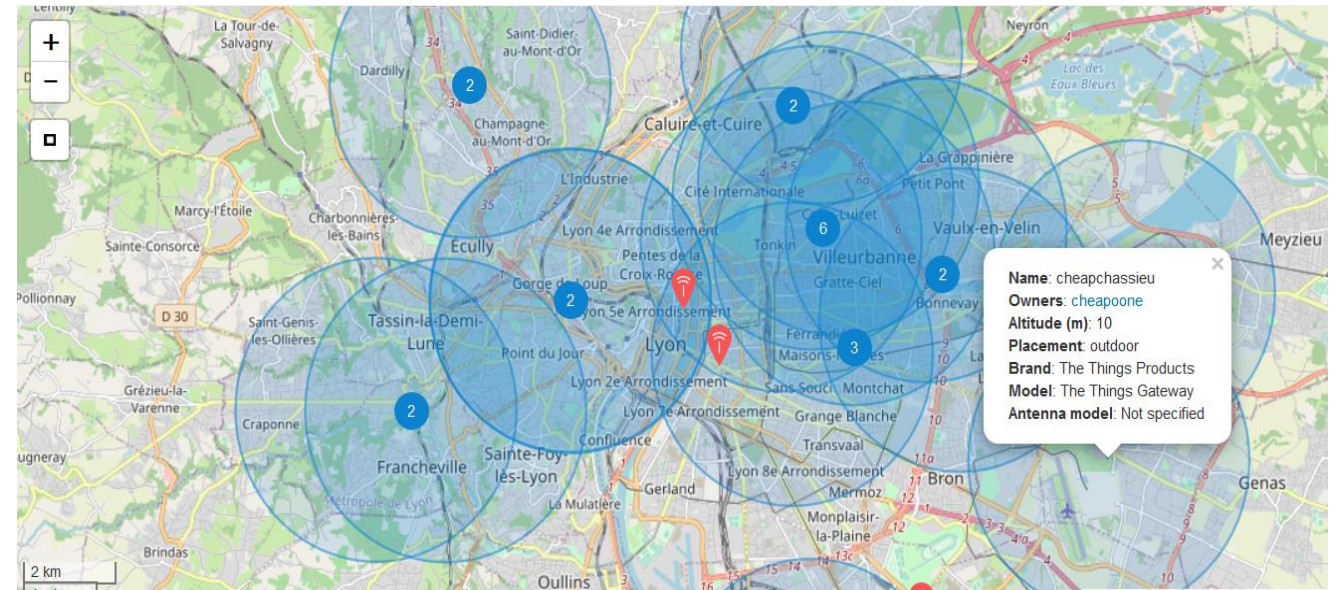
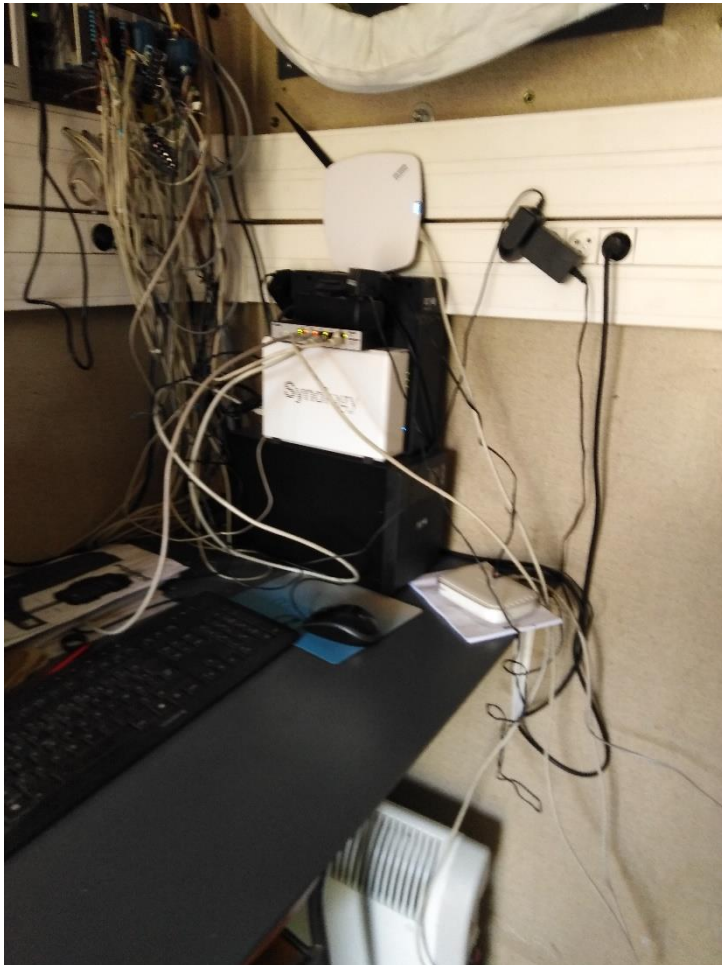
Prototype sans nano timer



Prototype avec nano timer



Gateway TTN-GW-868



Coût de la gateway : 301.42 euro HT (Farnell)

Gestion des données

Site web The Things Network

Filters: uplink downlink activation ack error

time	counter	port						
▼ 14:56:57		0						
▲ 14:56:57	0	3	retry confirmed	payload: 02 3F 00 80 01 83	Bat: 3.87	Dist: 128	Temp: 5.75	
⚡ 14:56:47				dev addr: 26 01 57 58	app eui: 70 B3D57E D002 E6 08	dev eui: A8 610A 32 374D 81 01		
▼ 14:46:53		0						
▲ 14:46:52	0	3	retry confirmed	payload: 02 71 00 80 01 82	Bat: 3.86	Dist: 128	Temp: 6.25	
⚡ 14:46:42				dev addr: 26 01 26 38	app eui: 70 B3D57E D002 E6 08	dev eui: A8 610A 32 374D 81 01		
▼ 14:36:55		0						
▲ 14:36:54	0	3	retry confirmed	payload: 02 71 00 80 01 83	Bat: 3.87	Dist: 128	Temp: 6.25	
⚡ 14:36:44				dev addr: 26 01 6D 51	app eui: 70 B3D57E D002 E6 08	dev eui: A8 610A 32 374D 81 01		
▼ 14:26:43		0						
▲ 14:26:43	×	×	historical	payload: 02 8A 00 80 01 83	Bat: 3.87	Dist: 128	Temp: 6.5	
▲ 14:26:43	0	3	retry confirmed	payload: 02 8A 00 80 01 83	Bat: 3.87	Dist: 128	Temp: 6.5	
⚡ 14:26:32				dev addr: 26 01 63 BD	app eui: 70 B3D57E D002 E6 08	dev eui: A8 610A 32 374D 81 01		

```
uint16_t temperature = valeurTemp * 100;
// uint16_t distanceSR04T = Distance1;
uint16_t distanceSR04T = m;
uint16_t Vbatterie = TensionBat * 100;
```

```
// ENVOI DES DONNEES SUR RESEAU LORAWAN
int err;
byte payload[6];
payload[0] = highByte(temperature);
payload[1] = lowByte(temperature);
payload[2] = highByte(distanceSR04T);
payload[3] = lowByte(distanceSR04T);
payload[4] = highByte(Vbatterie);
payload[5] = lowByte(Vbatterie);
```

```
modem.setPort(3);
modem.beginPacket();
modem.write(payload, sizeof(payload));
err = modem.endPacket(true);
modem.sleep();
```

Transfert des données sur google sheet

cheapomkr1300one

Fichier Édition Affichage Insertion Format Données Outils Modules complémentaires Aic

100% Lecture seule


	A	B	C	D	E	F	G	H
1	Date.UTC	Device_No	Distance_m	Tem	Battery_	RSSI		
5632	01/12/2020 10:55:33	8101	125	4.25	3.91	-101		
5633	01/12/2020 11:05:38	8101	126	4.50	4.00	-99		
5634	01/12/2020 11:15:43	8101	126	5.25	3.89	-99		
5635	01/12/2020 11:25:48	8101	126	5.75	3.85	-97		
5636	01/12/2020 11:35:53	8101	127	5.75	3.81	-97		
5637	01/12/2020 11:45:58	8101	127	5.75	3.87	-96		
5638	01/12/2020 11:56:02	8101	127	6.00	3.87	-100		
5639	01/12/2020 12:06:14	8101	127	6.25	3.87	-102		
5640	01/12/2020 12:16:11	8101	127	6.50	3.92	-103		
5641	01/12/2020 12:26:17	8101	127	6.00	4.01	-103		
5642	01/12/2020 12:36:20	8101	128	6.00	3.87	-117		
5643	01/12/2020 12:56:44	8101	128	6.25	3.89	-112		
5644	01/12/2020 13:06:49	8101	128	6.50	3.88	-112		
5645	01/12/2020 13:16:39	8101	128	6.50	3.87	-111		
5646	01/12/2020 13:26:44	8101	128	6.50	3.87	-113		
5647	01/12/2020 13:36:56	8101	128	6.25	3.87	-113		
5648	01/12/2020 13:46:54	8101	128	6.25	3.86	-107		
5649	01/12/2020 13:56:59	8101	128	5.75	3.87	-105		
5650	01/12/2020 14:07:03	8101	128	5.50	3.86	-103		
5651	01/12/2020 14:17:08	8101	128	5.25	3.85	-109		
5652	01/12/2020 14:27:12	8101	128	5.00	3.84	-99		
5653	01/12/2020 14:37:17	8101	128	4.50	3.83	-105		

Ajoutez 1000 lignes de plus en bas.

INTEGRATION OVERVIEW

Process ID cheapomkr1300threeulintegration

Status ● Running

Platform  HTTP Integration (v2.6.0) [documentation](#)

Author The Things Industries B.V.

Description Sends uplink data to an endpoint and receives downlink data over HTTP.

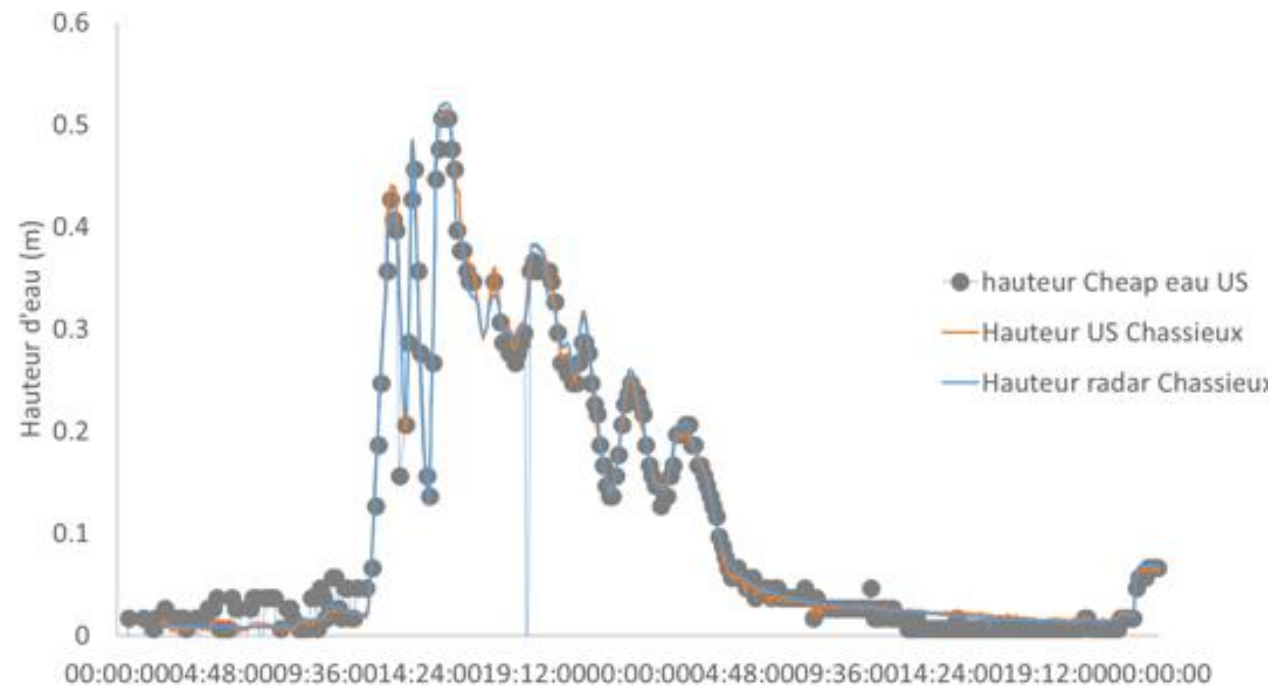
SETTINGS

Access Key

The access key used for downlink

default key devices messages

Résultat



Mesure effectuée le 2 octobre 2020 lors du passage de la tempête Alex

Conclusions

- Le prototype avec le nano timer est plus fiable dans son fonctionnement que celui qui n'en n'est pas équipé.
- L'autonomie électrique sans panneau solaire, est de 3 semaines pour le prototype avec nano timer contre 4 jours pour le prototype qui n'en est pas équipé.
- Les résultats de mesure restent proche des mesures réalisées avec du matériel professionnel, mais il y a quelques aberrations à corriger.
- La connexion via lorawan vers le cloud TTN donne de bon résultat, mais il y a quelques pertes de données et l'intégration http vers google sheet (provisoire) fonctionne bien.
- Dans le futur, on pourrait envisager d'utiliser l'outil Node Red pour faire transiter nos données sur un serveur local.



CHEAP'EAU
SOLUTIONS INNOVANTES
BAS CÔÛT - LIBRES - CONNECTÉES
[SUIVI | EAUX PLUVIALES]

CHEAP'EAU

SOLUTIONS INNOVANTES A BAS COUT
POUR LE SUIVI DES SYSTEMES DE GESTION DES
EAUX URBAINES

Merci pour votre attention!

Des questions?