
Rapport de conception

Projet IoT : moniteur de plantes

Pauline Béranger
ISMIN 2A
GI

Bilel Betka
ISMIN 2A
GI

Janvier 2025

Table des matières

1	Introduction	3
2	Architecture du système de surveillance des plantes et de leur irrigation	3
3	Mise en œuvre du suivi de l'état des plantes et de leur irrigation	4
3.1	Environnement de travail	4
3.2	BME680 : surveillance des constantes environnementales	5
3.3	Détecteur de niveau d'eau	5
3.4	Buzzer : signalement du manque d'eau	6
3.5	The Things Network : transmission des données	6
3.6	Datacake : visualisation des données	7
4	Test du module	7
5	Synthèse et ouverture	8
	Bibliographie	8
	Table des figures	8
	Table des listages	9
	Annexes	9

1 Introduction

Qu'il s'agisse de fleurs ou de plantes en pot, les humains depuis toujours décorent et habillent leurs espaces de vies avec des plantes d'intérieur. On définit comme plantes d'intérieur « des plantes cultivées en pot, jardinière ou mur végétalisé, destinées à l'ornement des domiciles, locaux de bureaux, entreprises, halls d'expositions, etc. » [1] Selon une étude de marché de Businesscoot réalisée en 2024, « [les] Français montrent un intérêt croissant pour les plantes d'intérieur. » [2] Le contexte posé, la problématique à laquelle nous aimerions répondre est la suivante : comment assurer un suivi efficace de l'état des plantes et une irrigation adaptée à leurs besoins ?

Afin de répondre à cette problématique nous avons décidé de mettre en place un module. Le module doit pouvoir réaliser plusieurs fonctions. Placé à côté de la plante, il doit pouvoir acquérir des données sur la température, la qualité de l'air ambiant, le taux d'humidité de l'air puisque ce sont de bons indicateurs sur l'environnement d'une plante [3]; il doit également indiquer un faible niveau d'eau quand le capteur correspondant est placé dans le réservoir. L'ensemble de ces données doit pouvoir être transmis du module vers un serveur The Things Network (TTN) *via* le protocole LoRaWAN, et visualisé sur Datacake.

Nous verrons dans un premier temps la conception de l'ensemble du système à partir du matériel fourni, puis plus spécifiquement l'intégration des différents sous-blocs, à savoir premièrement le capteur multi-fonctions Bosch BME680, ensuite le capteur de niveau d'eau et enfin la mise en place de la visualisation Datacake.

Ce module IoT a été réalisé à deux par les auteurs de ce rapport, dans le cadre de l'unité pédagogique « Projet IOT » dispensée en deuxième année du cursus ISMIN. Les connaissances théoriques utilisées ont été dispensées par les cours des professeurs Kharbouche et Marques. Les multiples circuits imprimés ont été conçus et réalisés par les équipes de l'École.

2 Architecture du système de surveillance des plantes et de leur irrigation

Le module se présente sous la forme d'un montage de plusieurs circuits imprimés sur lesquels sont placés les composants, les modules externes et les extensions de broches du microcontrôleur. En voici un synoptique :

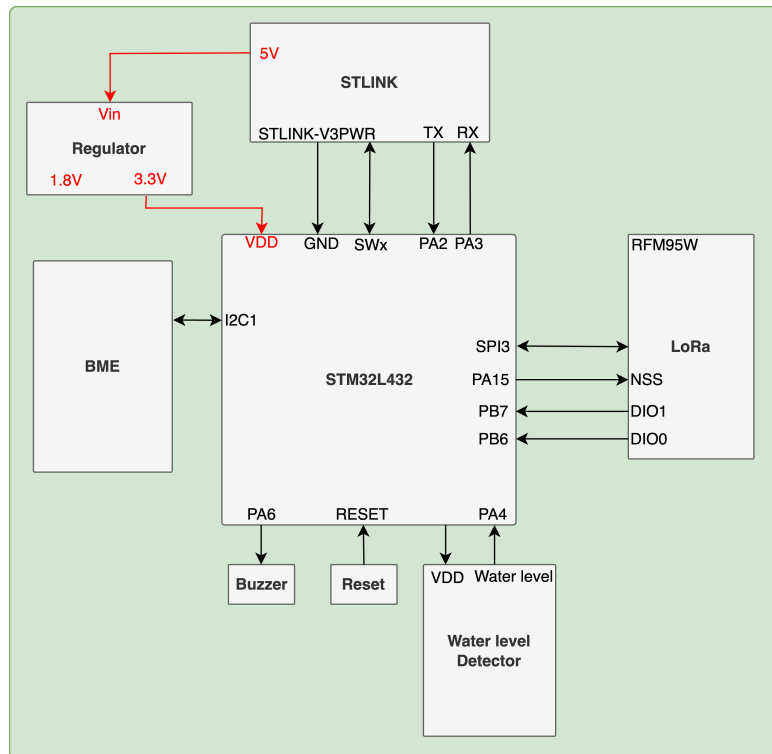


FIG. 1 : Synoptique du module

Voici la liste des composants utilisés :

- circuits imprimés du projet OCASS
 - microcontrôleur STM32L432KCU6
 - capteur multi-fonctions BME680
 - carte LoRa avec *transceiver*¹ RFM95W-868S2
 - ...
- unité de *debug* STLINK-V3PWR
- antenne LoRa
- capteur de niveau d'eau
- *buzzer*

La passerelle utilisée pour communiquer avec le serveur TTN est une Dragino LG308.

Le premier capteur est le BME680. Il communique avec la carte ST *via* un bus I²C. Le second est le capteur de niveau d'eau. Il est alimenté *via* la broche VDD (3,3 V) et l'acquisition des données se fait avec l'ADC par la broche PA4.

L'antenne est branchée sur la carte LoRa, elle-même pilotée par le microcontrôleur *via* un bus SPI, et permet la communication avec la *gateway*. Les données sont envoyées sur un serveur TTN et visualisées sur Datacake.

3 Mise en œuvre du suivi de l'état des plantes et de leur irrigation

3.1 Environnement de travail

L'architecture du module est du type « microcontrôleur + *transceiver*. » Le *transceiver* prend en charge toute la partie physique du protocole. La gestion se fait elle par le code C exécuté par le microcontrôleur.

¹Émetteur et récepteur.

Maintenir un coût faible étant essentiel, on implémente une trame LoRa *open source* par la librairie LoRa MAC In C (LMIC) et un formatage *via* la librairie Cayenne Low Power Payload (LPP).

Le réseau LoRa utilisé dans l'application est hybride : les *gateways* sont à charge de l'utilisateur, mais les données sont envoyées et visualisées dans le *cloud*. Pour notre application, le *network server* est TTN. Il met à disposition une offre gratuite qui convient amplement à notre usage.

L'*application server* est Datacake. Il permet de visualiser les données sous la forme de jolis graphiques.

3.2 BME680 : surveillance des constantes environnementales

Le capteur de Bosch est multi-fonctions. Il permet l'acquisition de trois données suivantes :

- la température de l'air
- le taux d'humidité relative de l'air
- le score IAQ de l'air

Les données passent par le périphérique I2C1 du microcontrôleur. L'acquisition des données se fait grâce à la librairie BME68x fournie par Bosch.

Plus le capteur sera proche de la plante, plus les mesures seront pertinentes.

3.3 Détecteur de niveau d'eau

Une des méthodes pour assurer l'arrosage de ses plantes pendant une absence est celle dite par capillarité. Un réservoir d'eau est relié par un fil au terreau de la plante. Par capillarité donc, l'eau va circuler le long du fil et passer petit à petit dans la terre. Le détecteur de niveau d'eau est à placer dans ce réservoir.



FIG. 2 : Exemple d'arrosage par capillarité²

Le détecteur est un circuit imprimé avec des électrodes sur sa longueur. Il possède trois broches : une alimentation,³ une masse, et une pour les données. On relie cette dernière au périphérique ADC1 du microcontrôleur sur le canal 9. L'ADC est piloté par le *timer* 6 pour une acquisition chaque seconde.

²Source : <https://communaute.carrefour.fr/app/730/channel/cote-jardin-2936/details/comment-fabriquer-son-systeme-dirrigatio-35709347>^o

³Dans notre configuration, le capteur de niveau d'eau est constamment alimenté. Dans l'idéal, cette alimentation devrait se faire ponctuellement à chaque fois que l'on a besoin du capteur, afin d'en limiter l'usure.

Les données reçues montent jusqu'à la valeur 1000 lorsque le capteur est totalement immergé. Or, lors de l'envoi, dans la fonction `cayenne_lpp_add_analog_output()`, cette valeur 1000 est multipliée par 100 puis stockée dans une variable de type `int16_t`, qui prend des valeurs allant de $-32\,768$ à $+32\,767$. 100 000 provoque un *overflow* et donc un résultat erroné. Pour pallier ce problème, nous avons divisé par 10 la donnée envoyée dans la fonction. On a une perte en précision, mais ce n'est pas dérangeant dans notre cas d'usage car nous cherchons seulement à détecter le dépassement d'une valeur seuil.

Il faut également noter que lors de l'échelonnage des valeurs du détecteur en fonction du niveau de l'eau en millimètres, nous avons constaté qu'elles n'étaient pas du tout linéaires. En effet, les 4 premiers millimètres font monter très rapidement la valeur à 600, tandis que les 33 millimètres suivants permettent d'atteindre progressivement la valeur maximale.

3.4 Buzzer : signalement du manque d'eau

Le *buzzer* est branché sur GPIO. Il permet une indication sonore lorsque le niveau d'eau a été jugé bas, c'est à dire en dessous de 600. L'absence de linéarité dans les valeurs de sortie du détecteur de niveau d'eau n'a pas été un obstacle, puisqu'il suffit de relever la valeur pour laquelle une alerte est souhaitée. Nous avons considéré qu'il était nécessaire d'alerter l'utilisateur lorsque le niveau d'eau était d'environ 4 millimètres.

La comparaison entre la valeur de niveau d'eau mesuré et la valeur seuil pour laquelle le *buzzer* sonne, s'effectue lors de la prise de la mesure dans l'ADC, donc toutes les secondes. Cette comparaison toutes les secondes est très pratique pour attester du bon fonctionnement du capteur lors du prototypage. Mais dans une situation plus proche d'un réel cas d'utilisation, nous pouvons imaginer faire une mesure toutes les heures par exemple.

3.5 The Things Network : transmission des données

Afin de permettre à l'utilisateur de suivre l'état de sa (ou ses) plantes en temps réel, le choix a été fait d'envoyer les données sur un serveur *cloud*. Nous rappelons que le système est à usage domestique et se doit de rester abordable et peu énergivore.

Dans un réseau suivant le protocole LoRaWAN, les échanges de données suivent une architecture bien définie. Comme évoqué dans la partie 3.1, [□] le microcontrôleur est relié à un *transceiver* LoRA, qui permet d'envoyer des données jusqu'à la passerelle vers un serveur *cloud* TTN. Dans notre configuration, nous avons un *device* LoRa de classe A ; ce dernier a une phase d'écoute après chacune de ses transmissions, sans contraintes de latence sur la réception des messages, afin de permettre un fonctionnement très peu énergivore et adaptable à la plupart des modules.

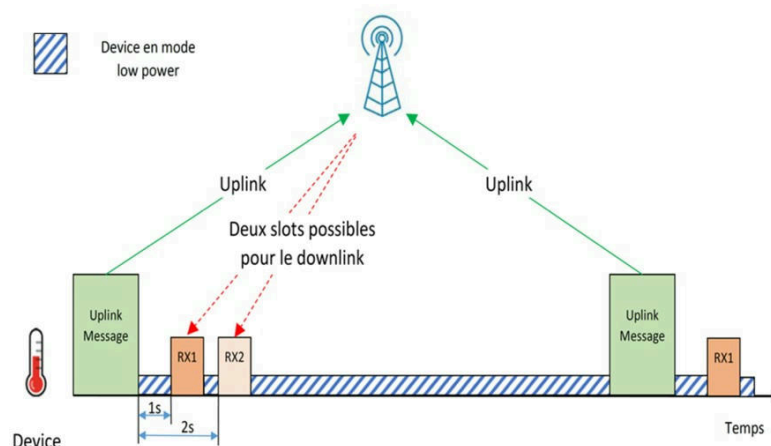


FIG. 3 : Fonctionnement du LoRa pour un *device* de classe A [4]

Le *device* LoRa envoie donc une trame (*uplink*) avec les données des capteurs (température, humidité, score IAQ et niveau d'eau) formatées en une Cayenne LPP. Les données sont envoyées par ondes radio sur la bande 868 MHz⁴. La passerelle LoRa, qui est en écoute permanente, reçoit cette trame *uplink* et agit comme un relais afin de les transmettre à son tour au *network server* TTN via Internet.

Le serveur TTN s'occupe ensuite du routage des données. Dans un premier temps il vérifie l'intégrité des messages grâce aux clés de sécurité *Network Session Key* (ou NwSkey) pour l'authentification réseau et *Application Session Key* (ou AppSKey) pour le chiffrement des données applicatives. Par la suite, TTN va envoyer les données au serveur applicatif pour la visualisation des données, Datacake.

3.6 Datacake : visualisation des données

La visualisation des données pour permettre une interface entre l'utilisateur et le module. La plateforme Datacake permet de configurer facilement et gratuitement la visualisation de données au format Cayenne LPP avec TTN. En configurant correctement Datacake, nous visualisons ces dernières très facilement. L'affichage du tableau de bord de Datacake est personnalisable et donc adaptable pour n'importe quel utilisateur. Voici le nôtre :



FIG. 4 : Visualisation sur Datacake

4 Test du module

Nous pouvons donc lire sur la figure précédente[□] une série de données acquises lors de notre soutenance le lundi 27 janvier 2025. Les trois premières mesures (température, score IAQ et humidité) sont relativement stables puisque nous nous trouvons dans une salle de classe sur une période totale d'une heure. Les seules variations que nous pouvons constater sont à 8:22, 8:40 et 9:15. En effet à ces instants, nous avons soufflé sur le capteur afin de montrer son bon fonctionnement. Souffler sur le capteur a provoqué une hausse légère de la température, une hausse du score IAQ et de l'humidité. De plus, le capteur de niveau d'eau n'était pas dans l'eau jusqu'à 8:35, puis il a été retiré de l'eau à 8:45 et 9:15. Cela se retranscrit parfaitement sur le graphique ci-dessus. Une ligne rouge permet d'identifier à partir de quelle valeur le *buzzer* sonne. Cette ligne rouge correspond à un niveau d'eau très bas, puisque les valeurs du capteur

⁴L'onde se déplaçant à la vitesse de la lumière, nous pouvons retrouver la longueur d'onde $\frac{300 \cdot 10^6}{868 \cdot 10^6} = 0,34$ m, en divisant par 2 on trouve la demi-longueur d'onde, qui correspond également à la longueur de l'antenne présente sur notre dispositif : 17 cm.

ne sont pas linéaires. Le comportement des courbes est cohérent avec la réalité. De plus, nous affichons à leur droite la dernière valeur mesurée pour chacune des courbes.

Ainsi la visualisation des données se fait en temps réel toutes les minutes. Cette résolution assez précise pour détecter une ouverture de porte séparant l'extérieur de l'intérieur et donc voir un creux de température ou une variation d'humidité par exemple. En ce qui concerne le niveau d'eau, selon le nombre de plantes et la taille du contenant de l'eau, le *buzzer* se mettra à sonner tous les trois jours voire deux semaines en été.⁵ Donc cette résolution temporelle est bien assez précise pour suivre l'évolution de l'eau.

5 Synthèse et ouverture

Nous avons donc réalisé un système permettant la visualisation à distances de données nécessaires à la surveillance de l'état de la plante grâce à l'association de deux capteurs et de LoRaWAN. La solution se base sur les offres gratuites de TTN et de Datacake, elle est donc adaptée à l'usage finalement assez basique qui en est fait. La problématique est résolue.

Cependant il est à noter que le type de détecteur de niveau d'eau que l'on utilise est connu pour s'user très rapidement. Pour un usage long terme, il serait donc judicieux d'optimiser le code afin de limiter son usure⁶ ainsi que l'éventuelle acquisition de remplacements. Le système de signalement sonore mis en place pourrait également être revu ; on pourrait imaginer une LED allumée ou un envoi automatique d'*e-mail* par exemple. Si l'utilisateur dispose d'une carte SIM supplémentaire avec un forfait téléphonique, il pourrait même s'envoyer un SMS moyennant le bon module GSM. Certaines fonctionnalités de Datacake permettent également l'envoi de SMS, mais cela nécessite un abonnement payant. On peut également noter l'absence de la luminosité parmi les données acquises, qui est pourtant essentielle pour une bonne santé des plantes. Autant de pistes d'améliorations pour le module.

Bibliographie

- [1] Wikipédia, « Plante d'intérieur ». Consulté le: 29 janvier 2025. [En ligne]. Disponible sur: https://fr.wikipedia.org/wiki/Plante_d'int%C3%A9rieur^o
- [2] Businesscoot, « Le marché des plantes d'intérieur - France », 2024. [En ligne]. Disponible sur: <https://www.businesscoot.com/fr/etude/le-marche-des-plantes-d-interieur-france>^o
- [3] D. K. Ryu *et al.*, « Control of temperature, humidity, and CO₂ concentration in small-sized experimental plant factory », *Acta Horticulturae*, n° 1037, p. 477-484, 2013.
- [4] A. Marques, *Projet IOT*. École des Mines de Saint-Étienne, 2024.

Table des figures

Fig. 1: Synoptique du module	3
Fig. 2: Exemple d'arrosage par capillarité ⁷	5
Fig. 3: Fonctionnement du LoRa pour un <i>device</i> de classe A [4]	6
Fig. 4: Visualisation sur Datacake	7

⁵Chiffres estimés par l'expérience personnelle de Pauline lorsqu'elle part en vacances et irrigue ses plantes par capillarité.

⁶Comme spécifié dans la partie 3.3,[□] il est préférable de n'alimenter que ponctuellement le capteur le temps de la mesure. Nous avons tenté de brancher son alimentation sur une sortie GPIO, mais il s'avère que la tension délivrée était trop faible. Nous nous sommes donc contentés de l'alimenter directement avec VDD.

⁷Source : <https://communaute.carrefour.fr/app/730/channel/cote-jardin-2936/details/comment-fabriquer-son-systeme-dirrigatio-35709347>^o

Table des listages

Annexes

Voici un lien vers le dépôt GitHub du projet : <https://github.com/senaalem/projet-iot>^o.