

Plusieurs pistes sont envisagées pour Solioti.

I/ Un mur végétalisé d'intérieur (haut de gamme)

Le principe : un mur végétalisé placé en intérieur (mur, meuble, ...) avec un système automatisé dont l'objectif est d'entretenir un couvert végétal d'ornement ou d'alimentation (fleurs, herbes aromatiques, ...). Le système se veut le plus attractif avec possibilité d'implémenter une application ludique sur écran juste au niveau du boîtier du Raspberry.

Le matériel IoT :

- [\[lien\]](#) des lumières de croissance hydroponique MR16 à spectre complet => 2€ l'unité ;
- [\[lien\]](#) un capteur d'humidité au sol => 5€ ;
- [\[lien\]](#) un capteur de température et d'humidité de l'air => 3€ ;
- [\[lien\]](#) un contrôleur de niveau d'eau (plein/vide) => 10€ ;
- [\[lien\]](#) un Raspberry modèle B Pi 3 + adaptateur d'alimentation + boîtier ABS + dissipateurs de chaleur + écran tactile 3.5 pouces avec stylo => 65€ ;
- [\[lien\]](#) carte d'extension de module de panneau de commande de relais RPI à 6 canaux pour Raspberry Pi 3 => 20€.

Total : 105€.

Fonctionnement général : le dispositif est commandé par un Raspberry équipé d'une carte d'extension (composée de relais) pour pouvoir communiquer avec l'équipement électronique. Le présent dispositif part du principe que les études documentaires permettent de se passer d'un capteur de luminosité. Le programmeur des lampes UV définit un éclairage de base suivant la saisonnalité pour que les plantes sélectionnées puissent survivre sur le mur végétalisé. Reste le capteur d'humidité du substrat pour s'assurer que le mur ne dépasse pas le seuil de stress hydrique et que l'arrosage est bien effectif. Un contrôleur de niveau d'eau indique aussi quand la cuve est pleine ou vide. Enfin, un capteur indique la température et l'humidité de l'air. Ces données sont transmises au serveur du Raspberry qui sera en capacité de stocker les informations en réseau local via Node-Red (plate-forme de low-code en Javascript qui sert de cerveau au système d'automatisation). L'utilisation des équipements électroniques demandera sans doute d'utiliser du Python ou du C++ en langage de programmation. L'interaction avec ces données (affichages, commande de la lampe horticole, réglages) sera possible via l'écran tactile connecté au Raspberry. Tout l'aspect alerte, choix des programmes, interaction avec le mur, conseils sont gérés par l'application sur l'écran. Une connexion Wifi peut permettre de dupliquer l'application en ligne pour que la gestion puisse aussi se faire à distance.

II/ Un mur végétalisé d'intérieur (bas de gamme)

Le principe : un mur végétalisé placé en intérieur (mur, meuble, ...) avec un système automatisé dont l'objectif est d'entretenir un couvert végétal d'ornement ou d'alimentation (fleurs, herbes aromatiques, ...). Le système optimise au maximum l'équipement pour rester attractif tout en étant le moins cher possible.

Le matériel IoT :

- [\[lien\]](#) des lumières de croissance hydroponique MR16 à spectre complet => 2€ l'unité ;
- [\[lien\]](#) un capteur d'humidité au sol => 5€ ;
- [\[lien\]](#) un contrôleur de niveau d'eau (plein/vide) => 10€ ;
- [\[lien\]](#) carte ESP-WROOM-32D => 4€ ;
- [\[lien\]](#) un potentiomètre => 0,50€ ;

- [\[lien\]](#) un capteur de température et d'humidité de l'air => 3€ ;
- [\[lien\]](#) led 3 couleurs => 0,50€ ;
- [\[lien\]](#) buzzer => 0,50€ ;
- [\[lien\]](#) module 3G => 0,50€ ;
- [\[lien\]](#) un bouton poussoir => 0,50€.

Total : 26,50€ (le prix ne prend pas en compte le boîtier de la carte ni le programmeur pour allumer/éteindre les leds, donc prévoir plutôt 40€).

Fonctionnement général : Le présent dispositif part du principe que les études documentaires permettent de se passer d'un capteur de luminosité. Le programmeur des lampes UV définit un éclairage de base suivant la saisonnalité pour que les plantes sélectionnées puissent survivre sur le mur végétalisé. Un contrôleur de niveau d'eau indique quand la cuve est pleine ou vide. Le capteur d'humidité au sol sert de sécurité au cas où le seuil de stress hydrique serait dépassé, auquel cas, un arrosage est forcé. L'arrosage peut aussi être forcé avec un bouton poussoir. Les alertes sur le module peuvent être gérées par une led 3 couleurs ainsi qu'un buzzer. Le buzzer signale le dysfonctionnement et la couleur de led la nature du problème (arrosage, dysfonctionnement électronique, communication réseau). Un capteur de température et d'humidité peut servir à définir si l'environnement est propice au développement des plantes. Concernant la connectivité, Wifi ou 3G peuvent être sélectionnées suivant la préférence du particulier. Les informations sont envoyées sur un serveur en ligne sur une plateforme Node-Red. Une application peut être développée afin d'offrir une interface didactique, permettant de gérer à distance le mur végétalisé et de recevoir des alertes.

III/ Un mur végétalisé d'extérieur (gestion des eaux de pluie) pour un petit projet

Le principe : le mur permet d'alléger le traitement des eaux de pluie en métropole, en plus d'autres atouts comme l'agrément, la lutte contre les îlots de chaleurs, la biodiversité.

Le matériel IoT :

- [\[lien\]](#) carte ESP-WROOM-32D => 4€x2 ;
- [\[lien\]](#) un contrôleur de niveau d'eau (plein/vide) => 10€ ;
- [\[lien\]](#) un capteur d'humidité au sol => 5€ ;
- [\[lien\]](#) un capteur de température et d'humidité de l'air => 3€ ;
- [\[lien\]](#) un pluviomètre => 16€ ;
- [\[lien\]](#) un capteur de mesure de distance étanche par ultrasons => 3,50€ ;
- [\[lien\]](#) module 3G => 0,50€ ;
- [\[lien\]](#) un écran LCD I2C => 2€ ;
- [\[lien\]](#) un relais arduino pour faire tourner la pompe et le contrôleur de niveau d'eau => 1,50€ pièce.

Total : 49,5€ (Le prix tel qu'estimé est un minimum, car il faut voir derrière si le module du mur pourra être alimenté par prise ou s'il faudra implémenter une pile. De même, il faut prévoir le coût de conception pour des cartes électroniques personnalisées avec boîtier.)

Fonctionnement général : Deux modules permettent de faire tourner le système. Un module au niveau du mur végétalisé et un autre au niveau de la cuve. Le module du mur contrôle l'humidité du substrat en coton, le pluviomètre, la température et l'humidité de l'air. Le module de la cuve détaille la hauteur du niveau d'eau dans la cuve, peut déterminer quand la cuve est pleine ou vide, actionne la pompe et peut afficher les données d'alerte des deux modules via l'écran LCD. La communication réseau se fait en 3G, pour ne pas avoir à s'intégrer dans le réseau Wifi du client

(même si la possibilité existe). Un programme d'arrosage automatisé de base fait vivre le système. Si le seuil de stress hydrique du mur est dépassé, un arrosage est forcé. Le programme d'arrosage peut être plus ou moins intensif suivant que la cuve soit pleine ou vide, ou que l'API météo annonce une pluviométrie conséquente ou une sécheresse pour les prochains jours (sur 5 jours, c'est gratuit, mais ça peut aller au-delà de 30 jours). Si le capteur température détecte une température inférieure à 1 ou 0, le système d'arrosage s'interrompt. Si le pluviomètre détecte une forte pluie, le système d'arrosage se met en mode tamponnement des eaux de pluie. Le système peut être perfectionné si l'API météo détecte quel jour viendront les fortes pluies, pour vider la cuve au préalable. Les contrôleurs de niveau d'eau au niveau de la cuve servent de sécurité. Si la cuve est pleine, les électrovannes d'arrivée d'eau sont bloquées pour éviter l'inondation. Si elle est vide, la pompe est désactivée de base le temps d'alimenter le réservoir en eau de ville ou d'attendre la prochaine arrivée d'eaux de pluie. Le capteur de distance à ultrason transmet une donnée plus précise du niveau d'eau restant dans la cuve. Toutes les données sont transmises via 3G à un serveur en ligne Node-Red. Le client/paysagiste peut ainsi suivre l'état du mur avec les différents capteurs, et actionner à distance l'arrosage au besoin, ou forcer un programme d'arrosage donné.

IV/ Les autres solutions à développer

Parmi les autres solutions à développer :

- les murs végétalisés d'extérieur en gestion de pluie pour de grands projets. La complexité ou l'envergure du projet demanderait une plus grande fiabilité ainsi qu'une meilleure performance. Pour commencer, le mieux serait de se reposer sur des modules de Raspberry aillant fait leurs preuves, au moins pour le module de la cuve qui va centraliser toutes les données. Le module servirait de passerelles aux capteurs sur les murs et gérerait les équipements sensibles comme la pompe, grâce à des relais. Le coût serait bien sûr plus important que le dispositif précédent. Ceci dit, avec le temps, il peut être allégé grâce au R&D pour reproduire un système de Raspberry adapté aux problématiques IoT des murs végétalisés de A à Z, à un prix plus abordable. Ceci dit, il faudra du temps pour développer cette solution. L'avantage du Raspberry, c'est que le système pourra être autonome en local, sans forcément devoir être commandé à distance sur un serveur en ligne comme les précédents modèles.
- Les murs végétalisés d'extérieur en gestion des eaux grises. Ici, le défi posé appartient plus au domaine du hardware qu'à l'IoT en tant que tel. En effet, l'équipement électronique ne devrait pas vraiment différer. L'enjeu principal devrait d'utiliser ces eaux grises sans faire mourir le couvert végétal (par l'intégration de produits chimiques), d'être vigilants sur l'entretien du matériel (notamment des conduits d'eau ainsi que de la pulvérisation) et sur les nuisances qu'un tel système peut engendrer (odeur, etc).
- Les murs végétalisés en gestion des eaux de pluie sur les abris de bus. Là encore, l'équipement électronique ne devrait pas grandement changer, si ce n'est qu'il pourra être plus bas de gamme (pour la pompe, en particulier) dans la mesure où les besoins sont encore inférieurs à un petit projet de murs végétalisés d'extérieur. L'enjeu sera de protéger l'équipement des dégradations et vols dans des espaces couramment fréquentés par le public, ainsi que de faire vivre le couvert végétal en extérieur pour l'agrément. En cela, quelques problématiques du mur végétalisé d'intérieur peuvent revenir.