

# enregistreur universel de données de jardinage

un pas vers l'Intelligence Artificielle au Jardin.

**Gamal Labib (Egypte)**

L'irrigation intelligente contrôlée par des capteurs, contribue à la conservation de l'eau, et aide à l'amélioration de la santé des plantes en situation de stress hydrique, du fait d'un programme d'arrosage inadapté. Un système d'irrigation intelligente, basée sur l'apprentissage machine, est un premier pas vers cette approche. Dans cet article, je présente un enregistreur de données constituant un premier élément du jardinage intelligent.

Source : Adobe Stock



L'automatisation de l'irrigation d'un jardin, soulage son propriétaire du souci quotidien de l'arrosage manuel. Malgré l'automatisation, si la distribution d'eau n'est pas adaptée, des zones décolorées peuvent apparaître dans les pelouses et les plantes peuvent présenter alors des signes de stress hydrique.

L'irrigation intelligente est un sujet sur lequel fabricants et chercheurs se sont focalisés depuis des décennies. La collecte des données par des capteurs, est un moyen magique permettant d'activer ou désactiver l'arrosage de façon adéquate et atténuer les variations des conditions de développement des plantations. Le capteur le plus utilisé pour cela est une sonde mesurant l'humidité du sol au niveau des racines des plantes. Des conditions de sécheresse du sol activent l'arrosage, jusqu'à ce que les racines soient correctement hydratées. [1].

L'Intelligence Artificielle (IA) donne une autre dimension au processus d'irrigation. Plusieurs approches sont possibles. Dans cet exemple [2], une caméra couleur ordinaire est utilisée, en conjonction avec des techniques d'apprentissage afin d'identifier si la plante est en situation de stress en raison d'un manque d'arrosage. Dans un autre exemple [3], l'apprentissage machine utilise les données de capteurs d'humidité, de température, et d'acidité (pH) [4], afin de prévoir les besoins en humidification du sol.

En général, les applications d'apprentissage machine (ML ou Machine Learning), se réfèrent à une grande quantité de données significatives pour un processus, afin de définir le modèle à appliquer. Des nouvelles données, jamais rencontrées précédemment, peuvent être injectées au modèle qui en retour fournit des informations importantes à leur sujet, et détermine les actions appropriées pouvant être prises. Les données utilisées dans les applications d'apprentissage machine peuvent se présenter sous différents formats, le format le plus populaire étant les valeurs séparées par des virgules (CSV ou Comma Separated Values).

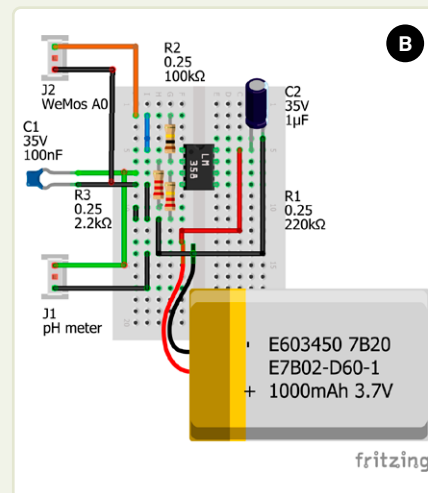


## Interfaçage d'un pH-mètre

L'acidité des sols varie lentement, selon des grandes périodes, des semaines et parfois de mois sont nécessaires pour constater des changements mesurables, en particulier si elle est modifiée par un jardinier. Ainsi, j'ai considéré que le suivi au jour le jour du pH était un objectif secondaire qui pouvait être accompli manuellement, ou être un complément à ce projet. En recherchant des capteurs de pH numériques disponibles sur le marché, je suis arrivé à la conclusion qu'ils étaient coûteux, particulièrement ceux pouvant être interfacés dans des projets utilisant un microcontrôleur Arduino.

Le capteur le plus économique disponible lors de la création de ce projet était le modèle manuel que je possédais déjà dans mon jardin, qui bizarrement, ne nécessite aucune pile pour fonctionner (voir **figure A**). Le capteur possède deux électrodes métalliques de matériaux différents et le sol agit comme un électrolyte. Une telle configuration constitue une pile électrochimique, qui est bien entendu relativement faible, mais

suffisamment puissante pour être mesurable par un multimètre analogique. En utilisant un amplificateur opérationnel et une petite pile, j'ai pu amplifier le signal afin de le diriger vers l'entrée analogique du chip WeMos, comme les autres capteurs (**figure B**).



Dans un projet d'Intelligence Artificielle pour le jardinage, il est nécessaire de se concentrer sur la collecte des données. C'est pour cela que j'ai prévu une grande variété de capteurs couvrant une large gamme de paramètres du jardinage.

Un enregistreur « universel » doit produire des fichiers de données CSV contenant les valeurs ambiantes de température et d'humidité, d'exposition au soleil, d'humidité du sol, de pH et d'évaluation du taux d'engrais et nutriments NPK (nitrogen-phosphorus-potassium fertility and nutrients), etc. Le volume d'eau atteignant le sol doit également être mesuré, permettant la détermination de l'efficacité des asperseurs utilisés par le système d'arrosage. Malheureusement, les capteurs de pH et de NPK sont trop coûteux pour pouvoir être expérimentés. J'ai cependant réussi à interfacé un capteur de pH économique afin de lui permettre de fonctionner avec des microcontrôleurs, comme le montre l'encadré « **Interfaçage d'un pH mètre** ».

## Réalisation du projet

L'enregistreur des données de culture est réalisé autour du module WeMos D1 Mini, basé sur un microcontrôleur ESP8266. Incluent les possibilités de connectivité Wifi, il supporte l'intégration de l'enregistreur au système de domotique (IoT) de mon logement. Son implémentation actuelle utilise quatre capteurs analogiques : un capteur de luminosité TEMA6000 permettant de déterminer l'ensoleillement des plantes, un capteur de pluie évaluant les averses de pluie (qui ne sont pas très fréquentes dans ma région), que je me prépare à utiliser

également pour déterminer la zone couverture des asperseurs, un capteur de niveau d'eau pour déterminer la pluviométrie ou la quantité d'eau délivrée par les asperseurs au point de mesure des données, et un capteur d'humidité du sol pour déterminer le degré d'humidité des racines. La configuration comprend un capteur de température et humidité DHT11, pour mesurer les conditions ambiantes au niveau de l'enregistreur. Il est important de noter que certains capteurs analogiques possèdent une sortie digitale qui fournit un niveau HAUT ou BAS, selon un seuil déterminé par un potentiomètre de réglage présent sur la carte. L'enregistreur est représenté sur la **figure 1**.

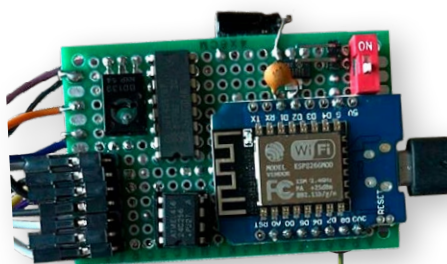
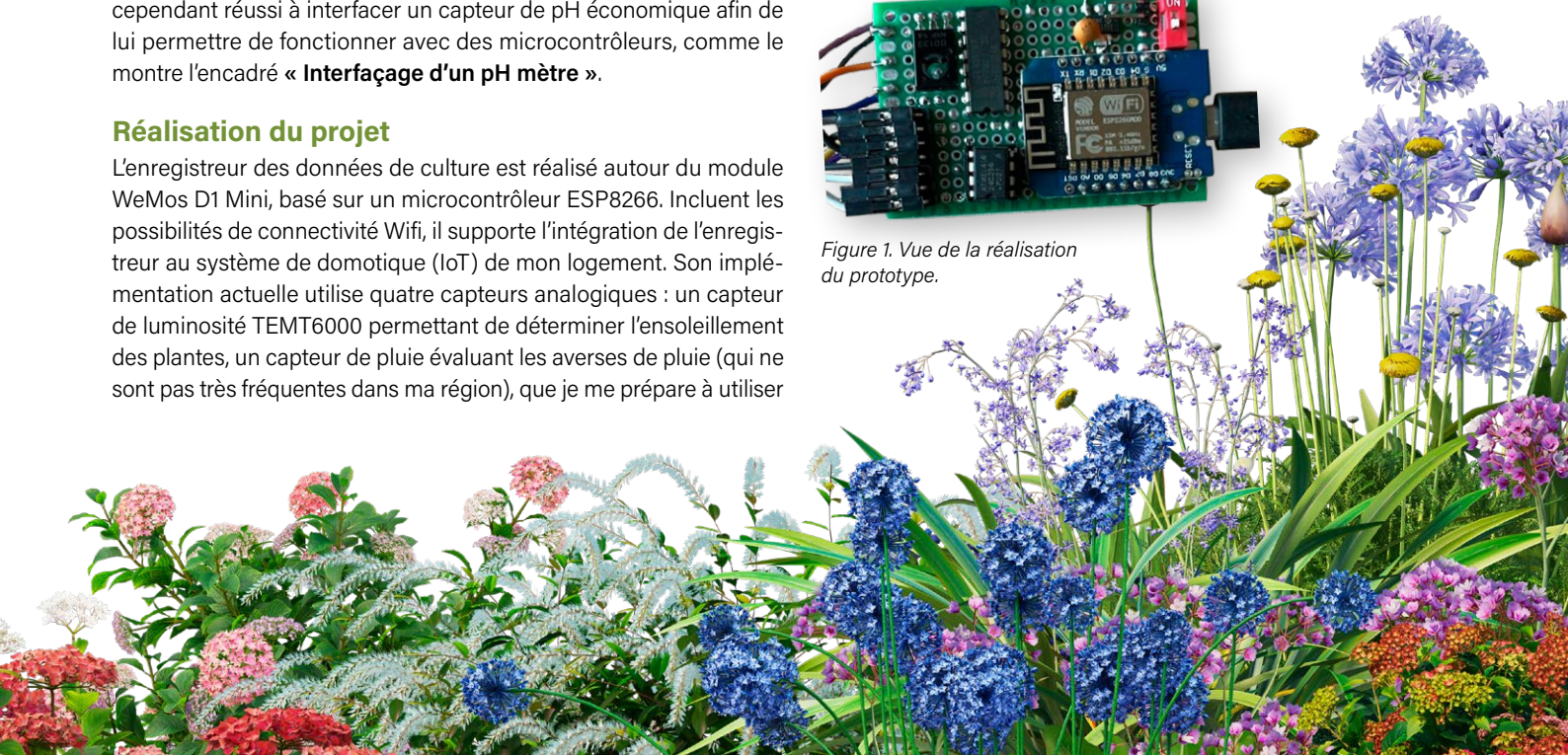


Figure 1. Vue de la réalisation du prototype.





L'ESP8266 ne possédant qu'une seule entrée analogique (A0), mais je dois connecter plusieurs capteurs analogiques. J'ai utilisé un chip multiplexeur analogique 4052, afin de permettre à l'enregistreur d'utiliser jusqu'à 8 capteurs. Le multiplexeur analogique utilise deux interfaces digitales GPIO ; les broches 16 (sortie D0) et 14 (D5) du Mini D1 sont utilisées à cette fin, permettant de sélectionner l'un des quatre capteurs afin de le relier à l'entrée analogique du microcontrôleur. L'enregistreur étant alimenté par une batterie, et prévu pour fonctionner sur une longue période, l'économie d'énergie est primordiale. Durant les périodes où l'enregistreur est désactivé, j'ai prévu de stopper le fonctionnement des capteurs en interrompant leurs connexions à la masse par un transistor NPN BD139. Pouvant supporter un courant de 1,5 A, sa base qui peut accepter une tension de 5 V, est pilotée directement par la broche GPIO 15 (D8) du D Mini. En fournissant un niveau HAUT au transistor, celui-ci est saturé, permettant le fonctionnement des capteurs. Dans le cas contraire, un niveau BAS permet d'économiser l'énergie consommée.

L'enregistreur devant être universel, j'ai choisi de délivrer les valeurs analogiques directement fournies par les capteurs, donnant ainsi à l'utilisateur, la possibilité de calibrer les capteurs et d'interpréter les valeurs le plus simplement possible. Le capteur de pluie est une exception, puisque je me base sur ses sorties analogique et numérique, comme je le décris plus loin. J'ai utilisé la broche GPIO13/D pour acquérir sa sortie numérique.

La **figure 2** montre le câblage des composants du projet. Les capteurs de pluie et d'humidité du sol sont totalement passifs, ils nécessitent la présence d'un module de contrôle pour fournir leurs données analogiques, alors que les autres capteurs fonctionnent de façon autonome. La liste complète des composants est fournie à la fin de l'article. La **figure 3** montre le câblage physique des composants du projet, avant leur installation dans le boîtier de l'enregistreur.

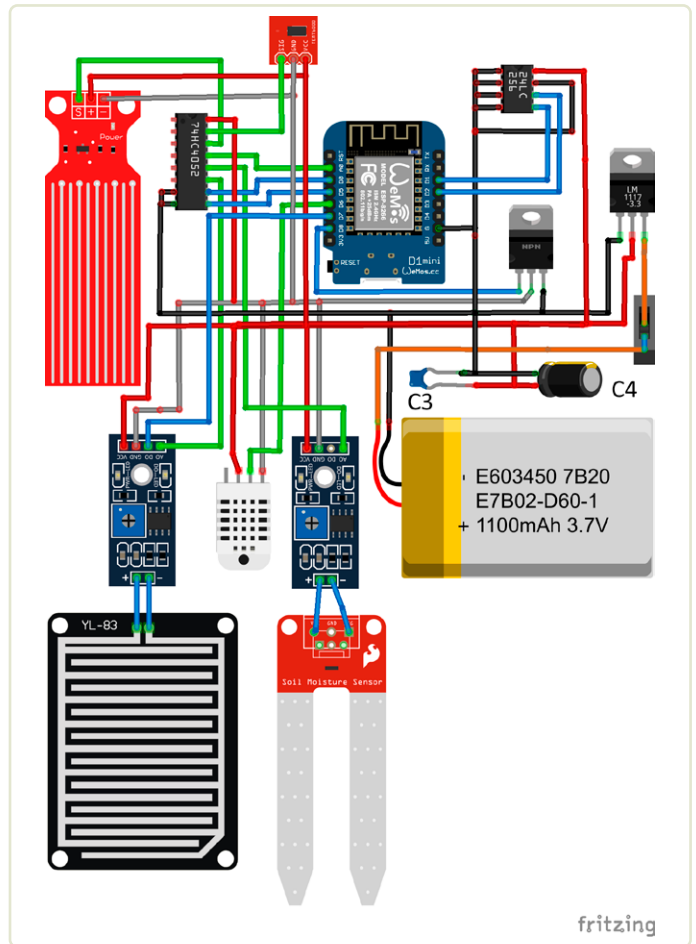


Figure 2. Câblage du circuit de l'enregistreur de jardinage.

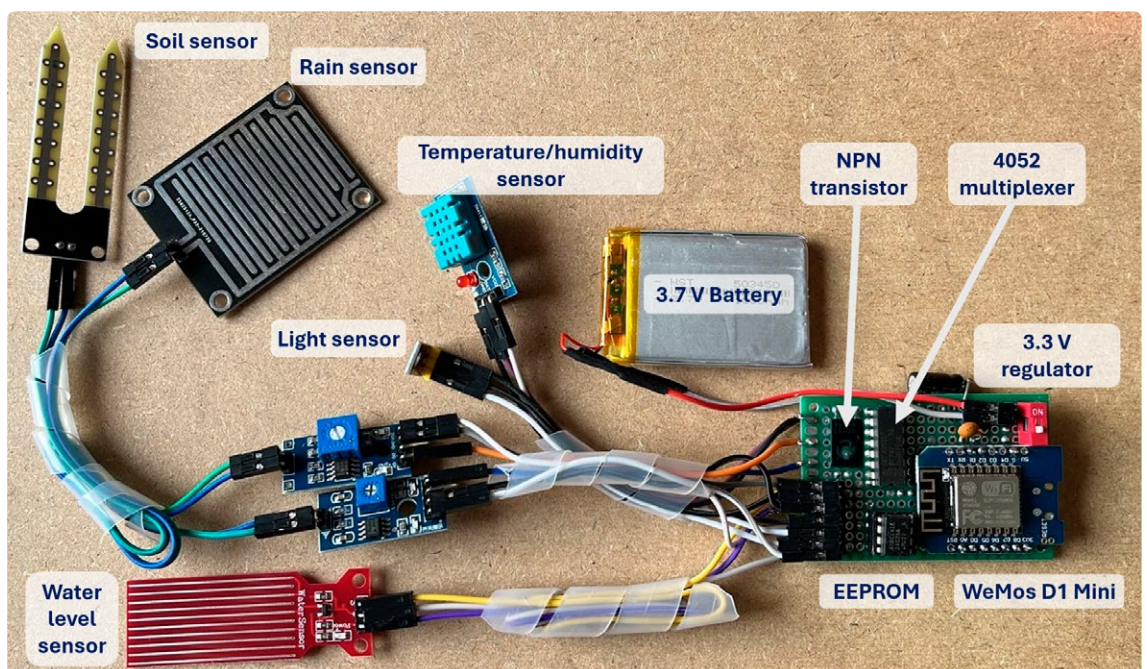


Figure 3. Ensemble des éléments reliés à l'extérieur du boîtier du projet.

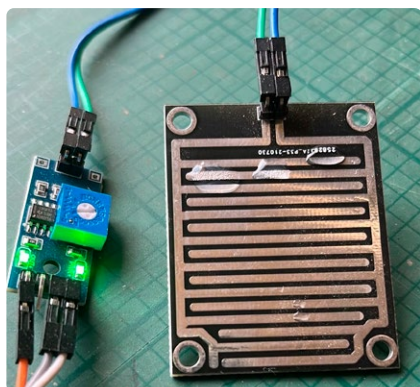
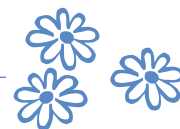


Figure 4. Le capteur de pluie.

13:55:15.169	-> Analog output: 538	→ New reading
13:55:16.187	-> Analog output: 538	} Readings induced by water viscosity
13:55:17.186	-> Analog output: 540	
13:55:18.190	-> Analog output: 538	
13:55:19.169	-> Analog output: 538	
13:55:20.165	-> Analog output: 538	
13:55:21.159	-> Analog output: 538	
13:55:22.187	-> Analog output: 537	
13:55:23.189	-> Analog output: 537	
13:55:24.182	-> Analog output: 537	

Figure 5. Interprétation des mesures délivrées par le capteur de pluie.

Après être mis en place, le capteur des paramètres d'ambiance est accessible par le fond du boîtier, afin de se situer le plus près possible du sol, tout en étant protégé de l'eau d'arrosage et de la pluie. Sur la partie haute de l'enregistreur, les capteurs de luminosité et de pluie sont exposés à l'environnement pour une meilleure acquisition de l'ensoleillement et de la quantité de pluie. Une trappe en verre est nécessaire pour isoler le capteur de lumière de l'extérieur. De façon identique aux capteurs d'eau et d'humidité du sol, ils sont reliés à l'enregistreur par un câble de 20 cm de long afin de faciliter leur positionnement. L'enregistreur est alimenté par une batterie au lithium de 3,7 V. Cette batterie a une capacité moyenne de 1100 mAh et ses dimensions lui permettent d'être facilement logée dans le boîtier du projet. Le module WeMos D1 fonctionnant sous une tension de 3,3 V, j'ai utilisé un régulateur de tension à faible chute (LDO) pour diminuer la tension de 0,4 V afin de respecter les spécifications de WeMos. J'ai choisi un régulateur prévu pour fournir une intensité de 800 mA permettant d'alimenter les composants de l'enregistreur. L'utilisateur devra choisir une batterie dont la capacité permet d'assurer le fonctionnement de l'enregistreur selon la durée souhaitée.

### Estimation de la quantité d'eau reçue

La quantité d'eau reçue par la zone considérée du jardin est un facteur important lorsqu'il convient de résoudre les problèmes de stress hydrique des plantes. J'ai utilisé un capteur de niveau d'eau pour déterminer cette quantité. C'est la méthode traditionnelle pour déterminer la quantité d'eau reçue, provenant de la pluie ou des arroseurs d'arrosage, dans un entonnoir alimentant un réservoir gradué. Le principe d'évaluation de ce dispositif simple est de considérer que la hauteur d'eau  $H$  dans le réservoir (par exemple cylindrique), dont la surface de base  $S$  est connue correspond au volume  $V = H \times S$ . Ainsi, lors de l'écriture du sketch Arduino, j'ai défini  $S$  par une constante et j'enregistre l'évolution du volume d'eau en fonction du temps selon la hauteur d'eau  $H$  indiquée par le capteur de hauteur d'eau.

Le capteur de pluie (figure 4) utilise une approche différente pour déterminer la quantité d'eau. La sonde utilise le même principe qu'un capteur de niveau d'eau, plus il y a d'eau reçue, plus la résistance du capteur diminue. Il possède différentes pistes métalliques qui lui permettent de déterminer la quantité de gouttes lui parvenant en un temps donné. Toutefois, l'accumulation de gouttes entre les pistes métalliques qui dépend de la viscosité de l'eau, est problématique.

Pour éviter ce problème, j'ai installé l'enregistreur en l'inclinant de façon à ce que les pistes métalliques du capteur soient orientées vers le sol afin de favoriser l'écoulement des gouttelettes et éviter de comptabiliser l'eau déjà prise en compte. Cela impose également la détection momentanée des gouttes d'eau au moment où elles atteignent le capteur et d'ignorer la détection des gouttes qui glissent sur la sonde en maintenant le contact. Les gouttelettes qui restent bloquées indéfiniment en bas des pistes du capteur constituent un problème supplémentaire.

C'est là que la sortie numérique du capteur vient à mon secours. En ajustant le potentiomètre de la figure précédente, cet effet peut être neutralisé, et la sortie numérique indique alors «NO RAIN» (pas de pluie, niveau HAUT). Il reste maintenant à résoudre le problème du ruissellement des gouttelettes sur les pistes métalliques. La mise en œuvre expérimentale du capteur de pluie montre de légères variations sur la sortie analogique du capteur. J'ai pris en compte ce phénomène en éliminant les valeurs subsistantes avant l'apparition d'une nouvelle valeur. Ce fonctionnement est illustré sur la figure 5, la sortie analogique d'une seule gouttelette indique une valeur de 538 et que les valeurs suivantes sont pratiquement identiques durant le ruissellement de la gouttelette provoqué par la gravité.

Lors de l'utilisation du capteur de pluie pour effectuer le réglage des arroseurs rotatifs, le système doit prendre en compte les deux catégories d'arroseurs utilisés pour les pelouses : les arroseurs fixes qui irriguent le gazon selon un cercle de rayon constant et les modèles rotatifs qui utilisent la force de l'eau pour provoquer leur rotation. Afin de prendre en considération les deux types d'arroseurs, le micrologiciel du microcontrôleur doit pouvoir distinguer entre la dispersion continue de l'eau et la dispersion intermittente.

### Utilisation des données enregistrées

L'enregistreur de Jardin est équipé d'une mémoire externe EEPROM de 32 ko pour la mémorisation des données captées pendant toute la durée de fonctionnement. Je l'ai installée sur un support DIP plutôt que de la souder, afin de permettre, si nécessaire, de la remplacer par un chip de capacité plus importante. La mémoire EEPROM interne de l'ESP8266 peut également être utilisée, mais sa capacité est limitée à 4 ko. De plus, l'écriture répétée de l'EEPROM interne aurait pour effet de réduire la durée de vie de l'ESP, c'est pourquoi je ne recommande pas de l'utiliser pour enregistrer les données.

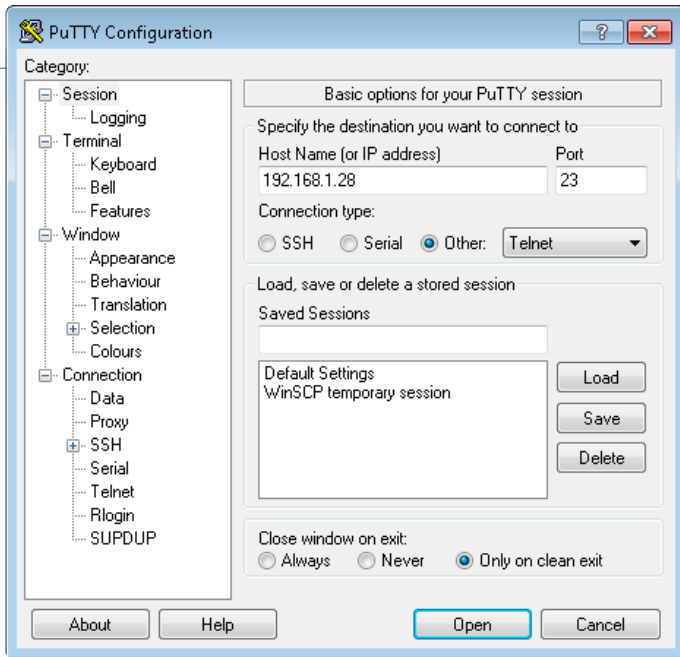


Figure 6. Configuration de PuTTY (1/2).

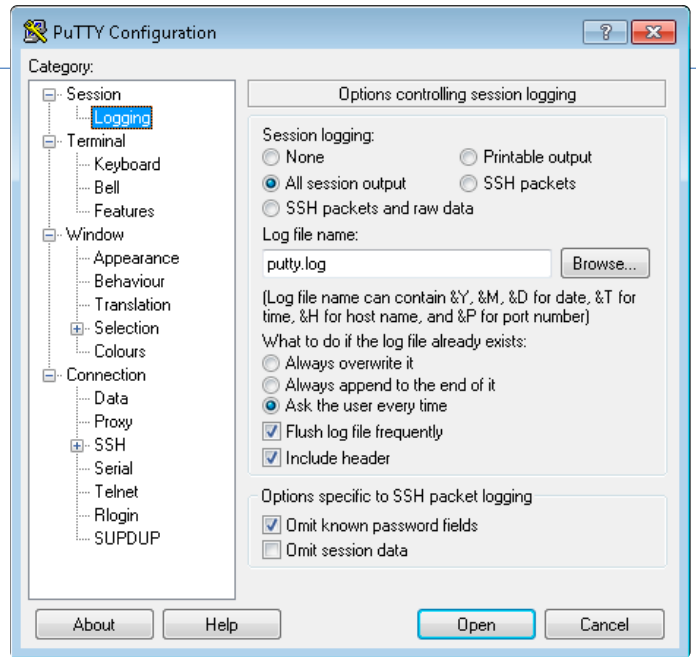


Figure 7. Configuration de PuTTY (2/2).

Chaque enregistrement des données recueillies est formaté selon le format CSV (valeurs séparées par des virgules). J'ai décidé que toutes les valeurs seraient représentées par des nombres entiers, cette application ne nécessitant pas la précision des nombres décimaux. C'est pourquoi le code transpose toutes les valeurs flottantes `Float` en entiers `Integer`, puis la fonction `String()` est utilisée pour les convertir en chaînes de caractères (`strVal`). L'appel en cascade des fonctions `strcat(strVal, ",")` est ensuite prévu pour insérer les virgules entre les différentes valeurs d'un enregistrement, les chaînes ainsi créées sont précédées par un en-tête contenant l'horodatage de l'enregistrement. Le projet ne comprend pas d'horloge RTC (Real Time Clock) pour maintenir une heure précise. Le dispositif se met à l'heure par consultation d'un serveur Internet NTP, lors de la réinitialisation intervenant à minuit chaque jour, afin d'obtenir les informations horaires. La fonction `millis()` est ensuite utilisée pour déterminer l'heure en cours et générer l'horodatage. La structure d'un enregistrement effectué est : *horodatage, température, humidité, éclairage, volume d'eau collecté, volume d'eau déversé, humidité du sol*.

Concernant la transmission à un ordinateur des valeurs enregistrées, il existe plusieurs approches possibles. Une des options peut être de disposer d'un serveur web s'exécutant sur l'enregistreur. Il est également possible de programmer le microcontrôleur pour transmettre les données à un PC en mode batch, après avoir été enregistrées dans la mémoire EEPROM. Un client Telnet tel que PuTTY peut être utilisé pour enregistrer les données dans un fichier CSV. Une troisième possibilité peut être de transmettre les enregistrements à un PC dès qu'une ligne de données CSV est créée. J'ai implémenté ce dernier mode d'opération dans la version courante de ce projet. La **figure 6** montre la fenêtre de configuration PuTTY sur le PC.

La procédure à suivre est assez simple : s'assurer que l'enregistreur de jardinage et le PC sont connectés au même réseau sans-fil, entrer l'adresse IP de l'enregistreur dans la zone *Host Name (nom de l'ordinateur hôte)*, définissez le port 23, puis choisissez le protocole Telnet depuis la fenêtre de saisie. Avant de cliquer sur *Open (Ouvrir)* pour lancer la transmission Telnet, il faut définir les paramètres de sauvegarde des données reçues de l'enregistreur. Cette étape est illustrée sur

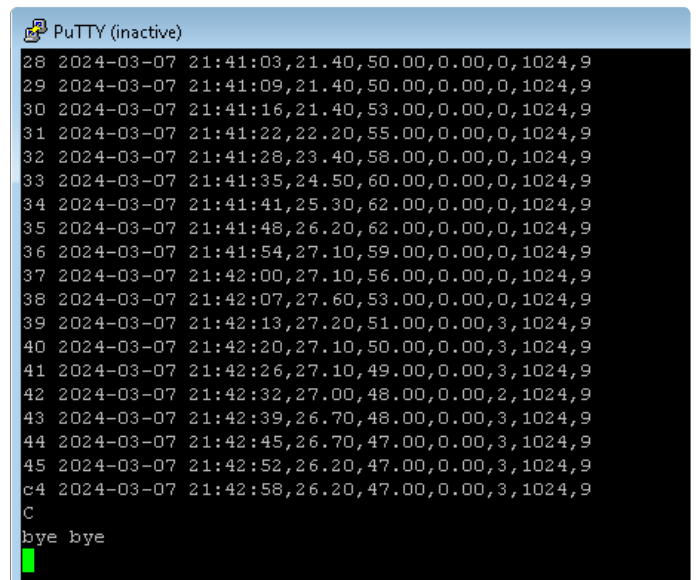
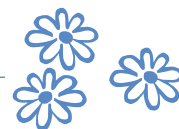


Figure 8. Fenêtre Telnet d'envoi des données au PC.

la **Figure 7**. Vous devez choisir *All session output (Envoi des données des sessions)* et *Ask the user every time (Demander à l'utilisateur à chaque transmission)* afin de ne pas réécrire un fichier log créé lors d'une session précédente. L'emplacement du fichier CSV destination doit alors être défini en utilisant le bouton *Browse (Parcourir)*. Vous pouvez alors cliquer sur *Open* pour démarrer le protocole Telnet. Après avoir été lancé, l'utilitaire Telnet débute la transmission des enregistrements en provenance de l'enregistreur, comme décrit sur la **figure 8**. Il est possible de mettre fin à la session de transmission en entrant « C » ; l'enregistreur retournera alors le message « bye bye » (Au revoir...) généré par la fonction `telnetAction()`. Les données des capteurs sont également disponibles sur le port série, au fur et à mesure qu'elles sont assemblées dans les enregistrements. Elles peuvent être lues par le Moniteur série de l'EDI Arduino selon la **figure 9**.





The screenshot shows the Arduino IDE Serial Monitor window with the title 'COM39'. It displays a series of sensor readings, each preceded by 'rain detected!'. The readings include Water Level (cm), Water Volume (cm3), Soil Moisture (unscaled and scaled), Humidity, Temperature (C and F), Heat index, Raw ADC data, Volts, Lux, Analog Input, and Digital Input. The data is formatted in a consistent, readable manner.

Figure 9. Données de sortie de chacun des capteurs dans le Moniteur série de l'IDE Arduino.

## Programme

La programmation du projet est réalisée en utilisant l'EDI Arduino. Le programme est composé du fichier principal *Garden.ino*, accompagné de cinq fichiers *.ino* auxiliaires. Chacun de ces fichiers comporte les fonctions pilotant un périphérique commun. *Timing.ino* contient les fonctions relatives au serveur d'horodatage NTP, *telnet-streaming.ino* concerne le formatage des enregistrements et leur envoi vers le PC, *analog-sensors.ino* comporte quatre fonctions de gestion des capteurs analogiques de l'ensoleillement, de la pluie, du volume d'eau et de l'humidité du sol. Enfin *digital-sensors.ino* contient une seule fonction qui fait l'acquisition de la température et de l'humidité ambiantes, *external-EEPROM.ino* contient les fonctions utilisées pour lire et écrire dans la mémoire EEPROM.

Ce style de codage facilite l'interprétation et la mise au point de chaque composante fonctionnelle du projet et améliore la réutilisabilité. Le **listage 1** comporte la section `loop()` du sketch ne comportant que dix instructions, dont six sont des appels aux fonctions décrites précédemment. La liste décrit également la section `setup()` dans laquelle je

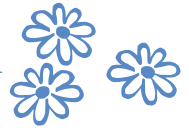


### Listage 1. Main loop.

```
void loop() {
  delay(2000);
  digitalWrite(SENSORS, HIGH); // enable sensors power
  delay(1000);
  senseAmbient(); // read ambient humidity & temperature
  senseLight(); // read ambient light intensity
  senseRain(); // read rain status
  senseWater(); // read irrigation water level
  senseSoil(); // read soil moisture level
  telnetAction(); // assemble logged data and send to telnet app
  digitalWrite(SENSORS, LOW); // disable sensors power
}

// this is part of setup()
pinMode(MUXA, OUTPUT); // part of 2-bit analog sensor selection address
pinMode(MUXB, OUTPUT); // part of 2-bit analog sensor selection address
pinMode(SENSORS, OUTPUT); // sensors GND control pin
pinMode(DHTPIN, INPUT_PULLUP); // ambient DHT sensor feed
pinMode(RAIN, INPUT); // rain status sensor digital feed
digitalWrite(SENSORS, LOW); // disable sensors power

// see [5] for the complete code
```



## Liste des composants

### Module Microcontrôleur

WeMos D1 Mini (ESP8266), <https://amazon.com/dp/B07W8ZQY62>

### Capteurs

Capteur de niveau d'eau, <https://amazon.com/dp/B09J2NK21Y>

Capteur de pluie, <https://amazon.com/dp/B01DK29K28>

Capteur d'humidité du sol, <https://amazon.com/dp/B083Q8DFHX>

Capteur d'ambiance DHT11, <https://amazon.com/dp/B0CCF2C2CF>

Capteur de luminosité TEMT6000, <https://amazon.com/dp/B00L8DW8L2>

### Divers

C3 = 100 nF, céramique

C4 = 470 µF, 10 V électrochimique

24LC256 EEPROM (Mouser)

LM1117MP-3.3-NOPB régulateur 3,3V (Mouser)

Transistor BD139 (Mouser)

Multiplexeur analogique CD4052BE (Mouser)

Batterie au lithium 1,100 mAh 3.7V, <https://amazon.com/dp/B06WRRQGR6>

Boîtier du projet 85×50×21 mm, <https://amazon.com/dp/B0B4VGGN6J>

Circuit Imprimé vierge double face FR-4 PCB 40×60 mm, <https://amazon.com/dp/B0968F3748>

contrôle le multiplexage des quatre sorties analogiques des capteurs, vers l'entrée analogique unique du microcontrôleur D1 Mini. Dans cette section, un port GPIO est dédié à la commande du transistor utilisé pour activer ou désactiver les capteurs afin d'économiser l'énergie. Le code complet se trouve en [5].

Il est important de remarquer que l'ESP8266 possède un convertisseur analogique-numérique (CAN) de résolution 10 bits, capable de transformer une valeur analogique d'entrée en une valeur numérique comprise entre 0 et 1023. De nombreux projets convertissent cette valeur en codage à 8 bits par des fonctions de transposition telle que `map(analogVal, 0, 1023, 0, 255)`, compressant cette valeur entre 0 et 255. J'ai évité de le faire afin de fournir les données des capteurs, le plus précisément possible.

### Remarques finales

L'irrigation intelligente, basée sur l'apprentissage machine nécessite un grand volume de données en provenance des capteurs afin de permettre de générer un modèle d'irrigation précis assurant le paramétrage de l'arrosage sous différentes conditions environnementales

que l'on retrouve dans un jardin. Une mise à jour du projet pourrait consister en l'enregistrement de l'état des plantes intéressantes, en utilisant une caméra ou d'autres types de capteurs, et l'ajout de ces données aux enregistrements effectués dans le cadre de ce projet. Cela permettrait de créer un modèle d'irrigation utilisant des techniques d'apprentissage machine avec supervision. ◀

VF : Jean Boyer — 230629-04



### À propos de l'auteur

Gamal Labib est un ingénieur passionné, enthousiasmé par les systèmes embarqués depuis plus de vingt ans, il est actuellement mentor (à [codementor.io](https://codementor.io)). Il est titulaire d'une maîtrise en engineering (Meng) et d'un Doctorat (PhD) en informatique. En plus d'écrire des articles techniques pour des magazines, il est professeur associé dans des Universités Égyptiennes et certifié Consultant en Informatique.

### Questions ou commentaires ?

Envoyez un courriel à l'auteur ([drgamallabib@yahoo.co.uk](mailto:drgamallabib@yahoo.co.uk)) ou contactez ([redaction@elektor.fr](mailto:redaction@elektor.fr))



### Produits

> **WeMos D1 mini Pro**  
[www.elektor.fr/19185](http://www.elektor.fr/19185)

> **Module ESP8266 ESP-01 Wi-Fi**  
[www.elektor.com/17326](http://www.elektor.com/17326)

> **Kit de capteurs SparkFun**  
[www.elektor.com/19620](http://www.elektor.com/19620)

## LIENS

[1] Exemple de projet d'irrigation : <https://circuitdigest.com/microcontroller-projects/automatic-irrigation-system-using-arduino-uno>

[2] Surveillance des plantations par l'IA : <https://smellslike.ml/posts/tf-microcontroller-challenge-droopthereitis/>

[3] Projet d'irrigation intelligente à base d'Arduino : <https://tinyurl.com/2kkbc2zx>

[4] Information sur le pH du sol : <https://esf.edu/eis/eis-soil-ph.php>

[5] Téléchargements : <https://elektormagazine.com/230629-01>