Le système PERLETTA7 Annexes

Jean Claude Bergès

Le système PERLETTA7
Annexes

Jean Claude Bergès
UMR PRODIG / Université Paris 1 Panthéon-Sorbonne
zebulon@univ-paris1.fr

25 mars 2024

*Le présent document contient des informations détaillées sur les divers composants du système d'acquisition **perletta7** tel que conçu et testé dans le cadre d'une opération financée par la municipalité de Mandagout. Il est complémentaire du manuel de montage et d'exploitation (qu'il ne prétend pas être) et pourra être également utilisé comme référence lors de la production de vidéos ou d'autres supports pédagogiques.*

Description du contenu

Annexe 1 : La carte fond de panier
La fonction principale de cette carte est la gestion de l'alimentation électrique. Quelle soit fournie montée ou non, elle devra être testée complètement avant de poursuivre le montage pour éviter les comportements erratiques (voire la destruction) des composants électroniques.

Annexe 2 : Le micro-contrôleur Arduino-Nano
Le micro-contrôleur qui constitue le cœur du système perletta en est l'élément le plus complexe. Le décrire dépasserait le cadre de ce document. Quelques informations de base sont fournies pour aider à comprendre l'architecture du système. Son montage et son processus d'initialisation sont également décrits.

Annexe 3 : L'horloge
Ce système est conçu pour fournir des séries chronologiques et la référence temporelle sera fournie par un élément spécialisé décrit dans cette partie. Cet élément devra être initialisé lors du montage.

Annexe 4 : La carte SD
Les données sont enregistrées sur carte SD. Montage, test et utilisation du support de carte se trouve dans cette annexe. Le format des données est également indiqué de telle sorte que les données produites puissent être utilisées sans aucun logiciel spécifique.

Annexe 5 : L'émetteur radio
Cette partie contient des informations sur le montage de ce composant qui sera utilisé dans les prochaines versions de perletta.

Annexe 6 : La sonde de température
Ce composant permet de mesurer la température du sol. Il est fourni dans un conditionnement étanche et intègre sa calibration qui lui permet de fournir une mesure absolue de ce paramètre.

Annexe 7 : Le pluviomètre
La mesure de la pluviométrie repose sur une utilisation originale (à notre connaissance) d'un capteur de pression. Le principe de la mesure et le montage des composants associés sont décrits.

Annexe 8 : L'humidité du sol
L'humidité est mesurée indirectement par des capteurs très simples de résistivité électrique du sol. Leur montage et leur installation demande toutefois un certain soin pour obtenir des résultats utilisables.

Annexe 9 : Programme d'exploitation
Les montages électroniques précédemment décrits ne peuvent fonctionner que coordonnés par un programme. Le code source de ce programme est fourni dans cette annexe à titre d'information. Il est écrit en langage C mais, du fait de sa construction très linéaire, il n'est pas nécessaire de connaître ce langage de programmation pour comprendre ce programme. Bien évidemment écrire ou modifier un programme de ce type peuvent présenter des difficultés d'un autre ordre que son simple décodage.

Annexe 1 : La carte fond de panier

La carte fond de panier regroupe l'ensemble des connecteurs et quelques composants électroniques. Les éléments constituant Perletta, la station d'acquisition, seront reliés à cette carte et au processeur Arduino-Nano mais jamais directement entre eux. La fonction principale de cette carte est la gestion de l'alimentation électrique. Utilisant des piles, elle n'alimente l'ensemble du système que par intermittence afin d'étendre la durée de vie de sa batterie.

Elle est montée sur une carte de prototypage en circuit imprimé et sera (autant que possible) fournie assemblée avant le montage. Toutefois elle devra être testée préalablement et cette section décrit les procédures à mettre en œuvre. L'envers de cette carte (figure 1.a) en montre les connexions.

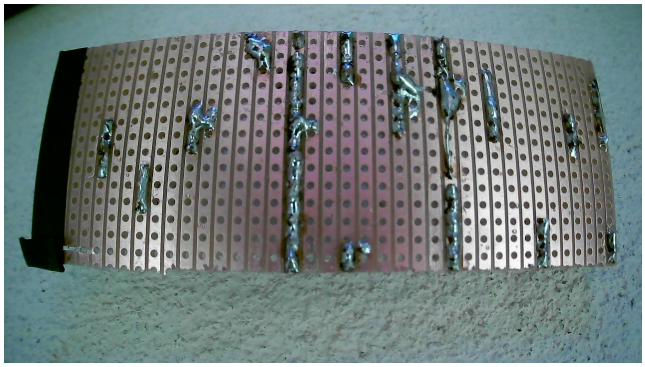

Figure 1.a : Envers du fond de panier

On remarquera que soudures et conducteurs se trouvent dans une seule direction (la plus petite). Dans la suite de ce texte, le terme ligne désignera une telle bande de connexion. Ainsi tous les éléments raccordés sur une même ligne seront connectés, leur position sur la ligne pourra varier sans influence sur le montage.

L'avers du fond panier apparaît sur la figure 1.b avant montage des divers éléments.

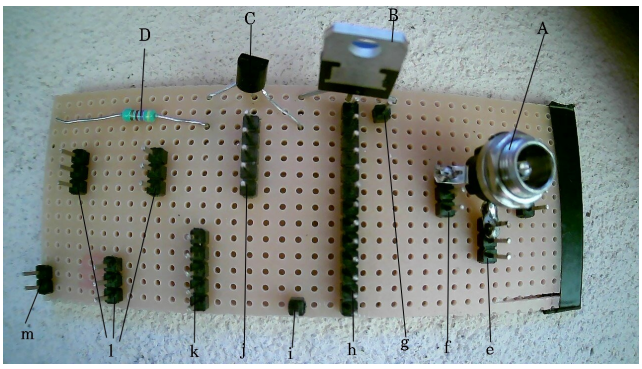

Figure 1.b : Fond de panier avant câblage

Sur cette carte sont soudés, outre les connecteurs, quelque composants électroniques. De droite à gauche figurent : une prise pour raccorder la batterie, un régulateur qui convertit la tension fournie par la batterie en une tension fixe de 5 V, un transistor fonctionnant comme un relais et une résistance (bleue sur ce montage) utilisée par la sonde de température. L'apparence des cartes pourra différer légérement mais la position relative des lignes sera identique. Sur cette figure les éléments suivants sont repérés :

A connecteur de la batterie
B régulateur 5 V

C transistor
D résistance
e neutre de la batterie
f tension positive de la batterie (6 V ou plus)
g relié à f
h barre de neutre (tous les composant doivent y être raccordé)
i sortie directe du régulateur (pour alimenter l'interrupteur périodique)
j commande du transistor et alimentation des composants analogiques (sondes)
k alimentation des autres composants
l trois connecteurs trois points (gauche, cente et droite)
m raccordement de la sonde de température

Le premier composant à intégrer sur cette carte sera l'interrupteur périodique TPL5110 (ne le monter qu'après la première étape du test). Ce composant à très faible consommation met en tension la broche DRV à intervalle de temps régulier et ne la stoppe que lorsqu'il reçoit un signal sur la broche DONE. L'intervalle est réglé par le potentiomètre X (figure 1.c) avec un tournevis fin. En butée le système est alimenté toutes les deux heures.

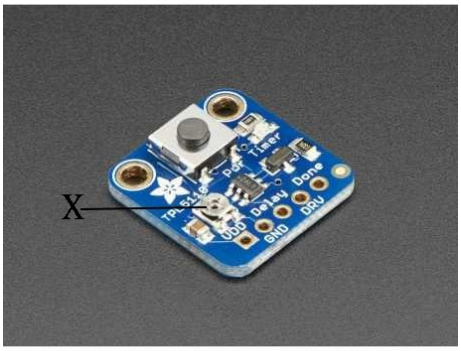

Figure 1.c : L'interrupteur périodique TPL5110

Le test de cette carte s'effectue en deux étape et nécessite l'utilisation d'un voltmètre. Ce test est important car une tension trop forte serait susceptible d'endommager les composants.

Première étape :

Raccorder **e** à **h** et **f** à **g**. Puis brancher la batterie sur **A**. Vérifier que la tension entre **h** et **i** est 5 V et que la tension entre **h** et **m** est 0 V. Lors de ces tests la sonde noire du voltmètre doit être en contact avec **h**.

Deuxième étape :

Intégrer l'interrupteur TPL5110 en raccordant sa broche VDD à **i**, sa broche GND à **h** et sa broche DRV à **j** (la broche DONE sera raccordée ultérieurement). Puis brancher la batterie. Un voyant vert doit alors s'allumer sur ce composant. Si ce voyant clignote le potentiomètre a été réglé dans le mauvais sens. La tension mesurée entre **h** et **m** doit être de 4.5 V et celle entre **h** et **j** de 5 V (les valeurs lues peuvent différer de quelques centièmes de Volts).

Lors des câblages, il est recommandé d'utiliser des fils noir ou brun pour le neutre et rouge ou orange pour la tension positive.

Annexe 2 : Le micro-contrôleur Arduino-Nano

Coordonnant l'action des composants du système, le micro-contrôleur Arduino-Nano en constitue le cœur. Un micro-contrôleur est un ordinateur à très faible consommation électrique, intégrant ses composants périphériques : mémoire, interfaces de conversion et de communication avec l'extérieur. Ces composants sont d'un coût très modéré mais leur puissance de calcul très limitée correspond à celle des premiers micro-ordinateurs des années 1980. Cette annexe est organisée en quatre parties : présentation de l'environnement de développement Arduino, description du composant Arduino-Nano, manipulation de l'interface et intégration de cet élément dans le système **Perletta**.

2.1 L'environnement de développement Arduino

Apparus au milieu des années 1970, les micro-contrôleurs sont depuis extensivement utilisés dans les petit automatismes (électroménager, outillage, dispositifs d'irrigation). Malgré leur simplicité leur emploi n'est pourtant pas simple. Leur capacité de calcul limitée ne permet pas de supporter un système d'exploitation comme Windows et ils ne peuvent être programmés que raccordés qu'à un autre ordinateur sur lequel sera installé un logiciel spécialisé permettant de communiquer avec le micro-contrôleur. Ce type de logiciel sera, dans la suite de ce texte, qualifié d'environnement de développement.

Estimant que les environnements de développements proposés dépassaient les capacités financières de leur institution, un groupe d'enseignant de l'institut de design d'Ivrea (Italie) s'est engagé à partir de 2005 dans la conception et la programmation d'Arduino, un environnement de développement pour micro-contrôleur. Cet environnement est ouvert et libre de droits, ce qui signifie qu'il peut être utilisé sans contrepartie financière. Progressivement, une communauté internationale s'est construite autour d'Arduino ajoutant des fonctions nouvelles et fédérant les utilisateurs.

Le système Arduino est constitué de deux composants. D'une part, un logiciel qui peut être gratuitement téléchargé sur les ordinateurs Windows, Mac-OS ou Linux et qui permet la programmation et la communication avec le micro-contrôleur et, d'autre part, une série de cartes électroniques, différant par leur capacité et leur format. Ces cartes se relient à l'ordinateur par un câble USB. La licence libre s'appliquant aussi à la conception des cartes, celles ci peuvent être produite sans rachat de droits ce qui contribue à maintenir leur coût modéré.

2.2 La carte Arduino-Nano

Parmi les diverses cartes du système Arduino le modèle Nano a été sélectionné pour sa compacité et sa faible consommation électrique. Un Arduino-Nano est représenté en figure 2.a ; du fait de l'intégration de multiples circuits périphériques dans le micro-contrôleur très peu de composants sont visibles sue cette carte. On remarquera la prise USB, ce micro-contrôleur (un ATMega328), un bouton poussoir pour la réinitialisation et quelques LED.

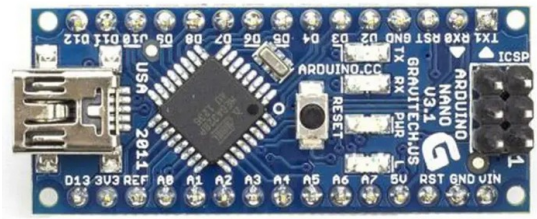

Figure 2.a : carte Arduino Nano

L'Arduino-Nano communique avec l'extérieur par 30 broches qui ont repérée par leur code inscrit sur la carte. La fonction de ces broches est la suivante :

VIN	Tension d'entrée non régulée
GND	Neutre
RST	Réinitialisation
5V	Tension d'entrée régulée en 5V
A0 à A7	Entrées analogiques, mesure des tensions en entrées
3V3	Sortie d'une tension en 3,3 V
D2 à D13	Entrées sortie digitales,
GND	Neutre
RST	Réinitialisation
RX0	Ligne d'entrée de la liaison série (dupliquée par la liaison USB)
TX1	Ligne de sortie de la liaison série (dupliquée par la liaison USB)

Il est à noter que quelques broches ont des fonctions secondaires comme A4 et A5 qui seront utilisées pour des liaisons spécialisées de certains composants.

Cette carte à deux modes de fonctionnement :

1. Connectées par la prise USB à un ordinateur, elle reçoit son alimentation électrique par cette même prise. Il est alors possible de téléverser un programme et d'interagir par un interface rudimentaire avec l'Arduino-Nano.

2. Non raccordée à un ordinateur, elle sera alimentée par la broche VIN ou 5V. Elle exécute alors le programme chargé préalablement en mode 1.

2.3 Interface de programmation

En cliquant sur le logo Arduino le programme permettant de programmer et tester le système se lance (figure 2.b).

Figure 2.b : Interface Arduino après le lancement

Ce logiciel peut être téléchargé gratuitement à partir du site https://www.arduino.cc et s'installe rapidement sur tous les systèmes d'exploitation : Windows, MacOs ou Linux. Son principal élément est un compilateur C qui convertit les programmes écrit dans ce langage en une suite d'instruction machine pour le processeur de la carte Arduino-Nano puis les transfère vers cette carte. Le C est un langage de programmation défini en 1978 et encore très répandu pour les calculs à hautes performances mais le décrire dépasserait le cadre de cette note ; la programmation dans ce langage n'est pas élémentaire et les limitations du processeur de l'Arduino-Nano la complexifie encore. Au demeurant, son étude ne serait guère utile pour le montage et l'exploitation d'un réseau de capteur, les programmes ne nécessitant pas ou peu d'interventions après leur mise au point. En revanche il est nécessaire de comprendre son rôle dans la configuration du système. Un point important est qu'il existe un lien étroit entre le raccordement des composants et le programme qui exploite ce montage. Pour ce motif un résumé du cablage doit figurer en commentaire en début de programme comme il apparaît sur la figure 2.b. La configuration de cette figure fait apparaître le début des deux programmes qui seront utilisés dans notre application : **perletta-init** et **perletta7**. Le premier programme sera utilisé lors du montage de l'appareil. Il permet de tester le fonctionnement du système en cours de montage, de lui affecter un code d'identification et de mettre à l'heure son horloge. Ce premier programme est interactif et fonctionne en connexion USB avec l'ordinateur supportant l'interface Arduino. En revanche le second programme qui est le programme d'exploitation doit s'exécuter de manière autonome. Ces moyens de communication avec l'extérieur étant la carte SD, d'une part, et la liaison radio, d'autre part. Ce programme, **perletta7,** devra être téléversé lorsque le montage aura été vérifiéet avant l'installation sur site.

Lors de la première activation de cet interface deux opérations doivent être effectuées : identifier le processeur et la prise USB sur lequel il est connecté et ouvrir les deux programmes qui seront utilisés. La première opération s'effectue à travers le menu **Tools** (ou **Outils**) de la barre supérieure. Il faudra sélectionner **board** puis **Arduino AVR** et enfin **Arduino Nano**. Dans la ligne suivante du menu **Processor ATmega 328P (old bootloader)**. Le sous menu **Port** indiquera l'emplacement de la connexion USB qu'il suffira de confirmer. Si ce sous menu n'est pas accessible la connexion entre le processeur Arduino et l'ordinateur est défectueuse. Le menu **File** (**Fichier**) de la barre supérieure permet d'effectuer la deuxième opération. Après avoir sélectionné **Open** (**Ouvrir**), un menu permettra de sélectionner **perletta-init** puis **perletta6** parmi les quelques programmes disponibles.

Ces opérations seront mémorisées ce qui permettra ensuite de disposer de ces sélections dès le lancement. Les seules fonctions utilisées seront **Upload** (**Téléverser**) dans le menu **Sketch** (**Croquis**) et **Serial Monitor** (**Moniteur série**) dans le menu **Tools** (**Outils**). La première fonction transfère un programme vers l'Arduino-Nano et la deuxième ouvre un interface d'échange rudimentaire (figure 2.c). Cet interface est en deux parties. La partie supérieure, réduite à une ligne, permet d'envoyer depuis le clavier vers l'Arduino et la partie inférieure affiche les messages en provenant.

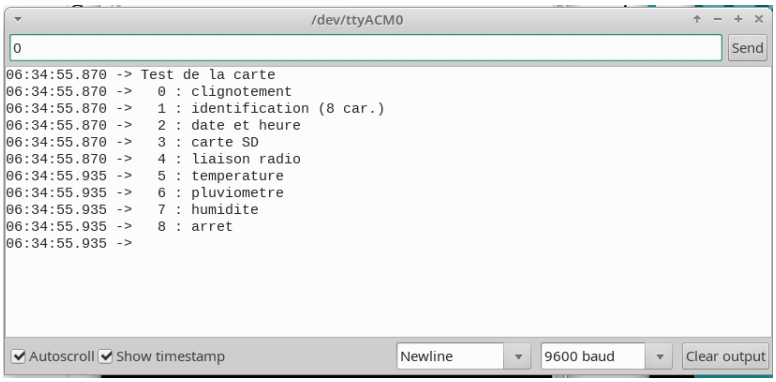

Figure 2.c : Le moniteur série

2.4 Montage et test

La carte Arduino-Nano sera raccordée au montage par trois connecteurs : un fil brun ou noir entre la broche **GND** et la barette **h**, un fil rouge ou orange entre la broche **5V** et la barette **k**, et un fil entre la broche **D3** et la broche **DONE** du TPL5110 précédemment installé (figure 1.b). Cette dernière connexion permettra au processeur d'informer l'interrupteur que les traitements sont terminés et que la tension peut être coupée.

La première opération à réaliser sera d'identifier le processeur (option 1 du menu de la figure 2.c). Cette identification sera utile pour repérer les mesures réalisées par la station. Elle comporte huit caractères et il est suggéré de la composer par des initiales en deux lettres suivies d'une date en six chiffres sur le modèle 240315 pour le 14 fêvrier 2024. Ce qui est très important est que chaque système ait une identification différente afin d'éviter les confusions lors de la concentration des données.

Annexe 3 : L'horloge

Les informations collectées ne pourraient pas être analysées si elles n'étaient pas associées à une date et une heure précise. Fournir cette information est le rôle du composant DS3231 qui sera l'horloge temps réel du système. De manière très similaire aux montres d'usage courant, ce composant est doté d'une pile permettant de maintenir l'heure même lorsque l'ensemble des autres composants est hors tension. Ce composant est représenté en figure 3.a, la support de pile se trouvant au verso.

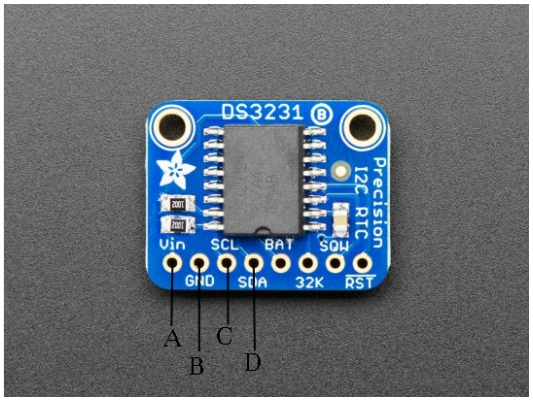

Figure 3.a : L'horloge DS3231

Seuls les quatre connecteurs de gauche devront être soudés sur des broches. En se rapportant à la figure 1.b, les broches d'alimentation A (Vin) et B (GND) seront respectivement reliées aux barrettes k et h. La broche **C** (SCL) sera reliée à la broche **A5** de l'Arduino-Nano et la broche **D** (SDA) à la broche **A4** (figure 3.b). On remarquera que pour ce composant les broches **A4** et **A5** ne fonctionnent pas comme des entrées analogiques mais en support d'un protocole de communication entre composants électronique, le bus I2C.

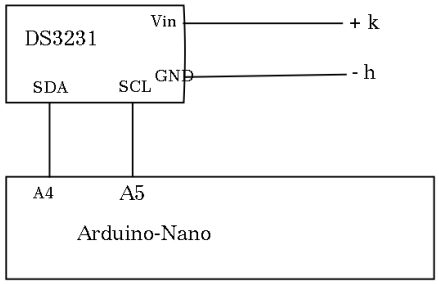

Figure 3.b : Cablage du composant

Le programme **perletta-init** permettra d'initialiser et de tester ce composant en utilisant l'option 2. Pour mettre à l'heure le DS3231 il faudra fournir successivement année, mois, jour, heure et minute. L'interface d'échange est assez primitif, pour entrer une information il faudra positionner le curseur dans la fenêtre supérieure d'une seule ligne. A l'exception de l'année toutes ces valeurs devront être fournies en deux chiffres (i.e 03 pour le mois de mars). Il est recommandé de régler l'horloge sur l'heure du temps universel, soit l'heure local moins un en hiver et l'heure locale moins deux en été.

Si l'horloge fonctionne correctement elle affichera, après l'initialisation, l'heure indiquée avec des intervalles de deux secondes (figure 3.c).

Figure 3.c : Test de l'horloge par perletta-init. Les données sont entrées dans la fenêtre supérieure (ligne terminée par le bouton **send***)*

Annexe 4 : La carte SD

Toutes les informations collectées par un système seront enregistrées sur ce même appareil. Cet enregistrement se fera sur une carte standard SD dans un format immédiatement lisible sous MS-Excel ou tout autre tableur. Le composant HW125 (figure 4.a) est un lecteur de carte SD compatible avec le système Arduino.

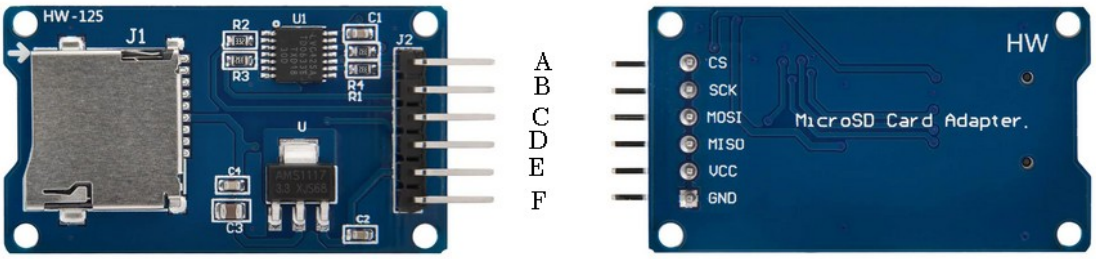

Figure 4.a : Lecteur de carte SD (avers et revers)

D'une manière très similaire à l'horloge DS3231, ce composant se connecte à l'Arduino-Nano par un protocole d'échange, le bus SPI. Le raccordement des broches entre ces deux éléments est donc imposé :

CS sur **D4**
SCK sur **D13**
MISO sur **D12**
MOSI sur **D11**

Toutefois une difficulté apparaît ; un autre composant, l'émetteur radio, utilise le même protocole d'échange et donc se raccorde aux mêmes broches. C'est pourquoi les connexions ne se feront pas directement mais en utilisant des connecteurs trois points intermédiaires. Aussi le câblage avec le fond de panier (figure 1.b) et l'Arduino-Nano sera réalisé selon la figure 4.b.

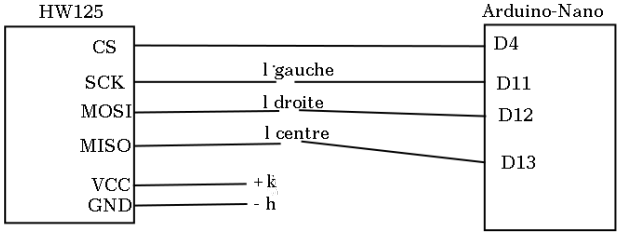

Figure 4.b : Raccordement du lecteur SD

Soit les connexions suivantes :

A	**D4**
B	**l** gauche
l gauche	**D13**
C	**l** droit
l droit	**D11**
D	**l** centre
l centre	**D12**
E	**k**
F	**h**

On s'assurera après avoir monté le HW125 qu'une carte SD est bien insérée dans le support. Lors du test de ce composant, le programme **perletta-init** crée un fichier **TESTTEST.TST** auquel il ajoute une ligne commençant par XXXXX, le lit et l'affiche sur l'écran.

Lorsque **perletta7**, le programme d'exploitation, est chargé les mesures effectuées sont enregistrées dans le fichier **PERLETTA.CSV**. Ce fichier, au format CSV, est donc compatible avec les suites bureautiques MS-Office ou LibreOffice. La figure 4.c montre un exemple de contenu de ce fichier.

AH24031	2024-03-20T16:52	TREF	235	PL	46	H2	1023	H1	1023	H0	910	TMP	170	PL	46	H2	1023	H1	1023	H0	909	TMP	170	PL	46	H2	1023	H1	1023	H0	911	TMP	170	PL	47	H2	1023	H1	1023	H0	910	TMP	170
AH24031	2024-03-20T16:52	TREF	266	PL	51	H2	1023	H1	1023	H0	996	TMP	170	PL	51	H2	1023	H1	1023	H0	1005	TMP	170	PL	52	H2	1023	H1	1023	H0	1017	TMP	170	PL	50	H2	1023	H1	1023	H0	1023	TMP	170
AH24031	2024-03-20T16:53	TREF	266	PL	51	H2	1023	H1	1023	H0	1023	TMP	171	PL	51	H2	1023	H1	1023	H0	1023	TMP	172	PL	51	H2	1023	H1	1023	H0	1023	TMP	173	PL	51	H2	1023	H1	1023	H0	1023	TMP	172
AH24031	2024-03-20T16:58	TREF	266	PL	51	H2	1023	H1	1023	H0	1023	TMP	180	PL	50	H2	1023	H1	1023	H0	1023	TMP	180	PL	51	H2	1023	H1	1023	H0	1023	TMP	180	PL	51	H2	1023	H1	1023	H0	1023	TMP	181
AH24031	2024-03-20T16:59	TREF	266	PL	51	H2	1023	H1	1023	H0	1023	TMP	180	PL	50	H2	1023	H1	1023	H0	1023	TMP	180	PL	52	H2	1023	H1	1023	H0	1023	TMP	181	PL	51	H2	1023	H1	1023	H0	1023	TMP	180
AH24031	2024-03-20T17:01	TREF	266	PL	51	H2	1023	H1	1023	H0	1023	TMP	178	PL	49	H2	1023	H1	1023	H0	1023	TMP	178	PL	51	H2	1023	H1	1023	H0	1023	TMP	178	PL	51	H2	1023	H1	1023	H0	1023	TMP	178
AH24031	2024-03-20T17:01	TREF	266	PL	50	H2	1023	H1	1023	H0	1023	TMP	177	PL	51	H2	1023	H1	1023	H0	1023	TMP	177	PL	51	H2	1023	H1	1023	H0	1023	TMP	177	PL	51	H2	1023	H1	1023	H0	1023	TMP	177
AH24031	2024-03-20T17:02	TREF	266	PL	51	H2	1023	H1	1023	H0	1023	TMP	177	PL	50	H2	1023	H1	1023	H0	1023	TMP	177	PL	50	H2	1023	H1	1023	H0	1023	TMP	177	PL	50	H2	1023	H1	1023	H0	1023	TMP	177
AH24031	2024-03-20T17:02	TREF	266	PL	51	H2	1023	H1	1023	H0	1023	TMP	177	PL	51	H2	1023	H1	1023	H0	1023	TMP	177	PL	51	H2	1023	H1	1023	H0	1023	TMP	177	PL	50	H2	1023	H1	1023	H0	1023	TMP	177

Figure 4.c : Fichier CSV créé par **perletta7**

Chaque ligne correspond à une séquence de mesure. En exploitation normale ces séquences seront espacées d'environ deux heures. Pour constituer ce fichier différentes séquence ont été générées en utilisant le bouton poussoir du composant **TPL5110** (décrit en annexe 1) ce qui explique que les heures de collecte soient beaucoup plus proches.

Le premier champ est le nom de l'appareil, ici **AH24031**. Celle ci est déterminée lors de l'initialisation du système par **perletta-init** (annexe 2). Il est important que le nom de chaque système soit différente. Le second champ indique la date et l'heure des mesures. Les champs suivants fonctionner en couple : l'identification de la mesure suivie de sa valeur. La signification de ces identification est :

TPL niveau de la batterie, une augmentation rapide de cette valeur signale des piles à changer
PL pression dans le pluviomètre, pour une sonde l'air libre une valeur entre 30 et 60
H2 mesure de résistivité de la sonde 2, entre 0 (conduction parfaite) et 1023 (isolation parfaite)
H1 mesure de résistivité de la sonde 1
H0 mesure de résistivité de la sonde 0
TMP température de la sonde thermique en dixième de degré

On remarquera que, à l'exception de TPL, ces mesures sont effectuées quatre fois. Ceci est fait pour s'assurer qu'elles sont stables et que les composants ne sont pas défectueux.

Annexe 5 : L'émetteur radio

L'émetteur radio est destiné à émettre en temps réel les informations enregistrées sur la carte SD. Bien qu'il ne soit pas utilisé par **perletta7**, la version actuelle du système d'acquisition, il est néanmoins prévu de le monter sur les systèmes de manière à ne pas avoir à modifier le matériel lors des prochaine version de **perletta**.

Cette fonction est fournie par le composant SX1278 qui utilise le protocole SPI comme le support de carte SD décrit en annexe 4. Aussi les connecteurs trois points l (figure 1.b) seront utilisés. Un autre point important est que ce composant fonctionne en 3,3 V et non en 5V. Aussi il ne prendra pas la tension positive du fond de panier mais de l'Arduino-Nano. Les raccordements nécessaires depuis ce composant sont les suivants :

Vin	**3.3 V**
GND	**h**
SCK	**l droit**
MISO	**l centre**
MOSI	**l gauche**
NSS	**D10**
RST	**D9**
D100	**D2**

Par ailleurs les raccordements suivants doivent exister entre la carte fond de panier et l'Arduino-Nano :

l droit	**D13**
l centre	**D12**
l gauche	**D11**

Annexe 6 : La sonde de température

La température de sol est un paramètre complémentaire de l'humidité et sa mesure sera réalisée par la sonde DS1820. Cette mesure, numérique, est précise au dixième de degré Celsius.

La sonde est fournie dans un conditionnement étanche et se termine par trois fils (figure 6.a). Les couleurs de ces fils indiquent leur fonction :

Jaune	signal
Noir	neutre
Rouge	tension positive (5 V)

Figure 6.a : La sonde DS1820B

Pour rester compatible avec les autres systèmes installés des connecteurs femelle seront soudés à l'extrémité de chacun de ces fils. Le neutre sera connecté à la barrette **h**, la tension positive à la barrette **k** et le signal à la barrette **m** (figure 1.b). De plus un connecteur sera posé entre cette même barrette **m** et la broche **D5** de l'Arduino-Nano. Ce câblage est représenté en figure 6.b.

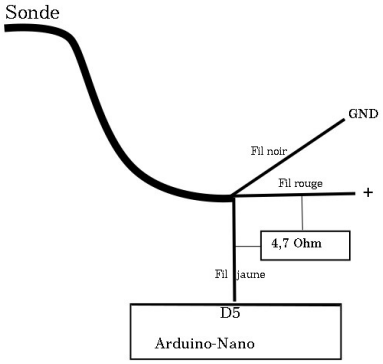

Figure 6.b : Câblage de la sonde thermique

Il est à remarquer que la sonde se raccorde à la broche D5 de l'Arduino-Nano en parallèle avec une résistance de 4,7 k-Ohm.

Le test de ce composant se fera par l'option 5 du programme **perletta-init**, la valeur renvoyée étant exprimée en dixième de degré.

Annexe 7 : Le pluviomètre

Les pluies intenses sont très variables dans l'espace et ceci d'autant plus que le terrain est accidenté. Aussi dans une zone où les dénivelés sont importants, comme sur la commune de Mandagout, une mesure locale de la précipitation est indispensable pour une interprétation correcte des résultats.

Pour notre système, cette fonction sera réalisée en mesurant les variations de pression dans une colonne d'eau. Cette méthode est compatible avec les mesures intermittentes (toutes les deux heures) et une alimentation par batterie, la diminution de pression due à l'évaporation dans les deux heure suivant une précipitation pouvant être considérée comme négligeable. Dans le montage du système **Perletta** (figure 7.a) cette colonne sera un tuyau vertical d'un mètre en PVC terminée par une bonde de vidange. Lors d'un événement pluvieux, la pluie pénétrera dans ce tuyau et le remplira d'autant plus que son intensité sera importante. A la base de la colonne, une durite sera reliée à une composant, intégré dans le boîtier de mesure qui en mesurera la pression et donc la hauteur d'eau dans le tube. Cette dispositif est très proche d'un pluviomètre traditionnel, la mesure d'une pression remplaçant celle de la lecture visuelle d'une échelle par un opérateur.

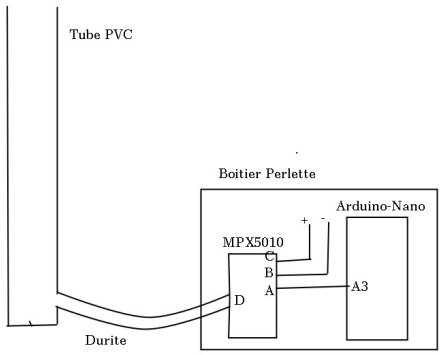

Figure 7.a : Schéma du pluviomètre

La mesure de pression sera réalisée par le composant MPX5010DP (figure 7.b). Ce composant, conçu pour les respirateurs médicaux, est capable de mesurer des différence de pression par rapport à la pression atmosphérique de 0 à 100 hPa avec une précision de 0,1 hPa. Ramené notre montage, il peut évaluer une hauteur d'eau entre 0 et 1 m avec une précision de 1 mm. Ces caractéristiques le rendent tout à fait adapté à l'estimation des précipitations.

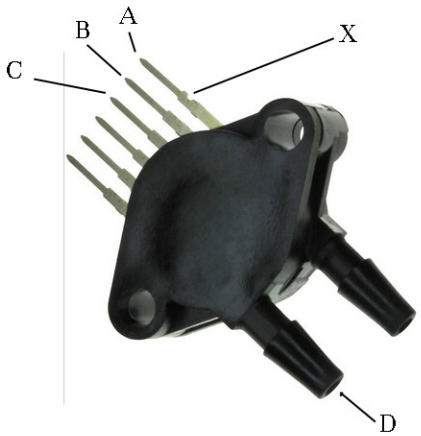

Figure 7.b : Le composant MPX5010DP

Avant tout montage, il faut repérer l'encoche **X** sur les broches du composant. La durite reliée à la base de la colonne devra être connectée en **D**, l'autre embout mesure la pression atmosphérique de référence ; il sera donc laissé libre dans le boîtier. L'affectation des broches métalliques est la suivante :

A sortie du composant une tension de 0 à 5 V
B neutre pour l'alimentation du composant
C tension positive pour l'alimentation du composant (5 V)

Sur le système **Perletta**, **A** sera connectée à la broche **A3** de l'Arduino-Nano, **B** à la barette **h** du fond de panier et **C** à la barette **j** de ce même fond de panier (figure 1.b). Il est important de raccorder ce connecteur sur **j** et non sur **k** parce que le composant MPX5010DP demande une intensié électrique faible mais ne tolère pas des tensions plus faibles que la tension nominale en alimentation.

Du fait de la forme des broches de ce composant, il est recommandé de souder les connecteurs pour éviter de faux-contacts.

Lors du test du pluviomètre par le programme **perletta-init**, les valeurs affichées doivent être entre 30 et 60 lorsque la durite est à l'air libre. S'il n'est est pas ainsi, il faut vérifier les connexions.

Lorsque **perletta7** est actif, les valeurs fournies par le pluviomètre seront interprétées par différence. Un phénomène pluvieux se traduit par une augmentation de la pression et donc de la valeur enregistrée, alors que l'évaporation sera associées à une décroissance beaucoup plus lente de cette valeur.

Lors de la première utilisation de l'eau devra être versée dans le pluviomètre de telle sorte que ladrite ne soit pas à l'air libre. Au bout de quelque mois d'utilisation ou après une pluie intense, il est recommandé de vider le pluviomètre pour le purger d'éventuels débris ou éviter que la colonne ne déborde.

Annexe 8 : L'humidité du sol

Il existe plusieurs méthodes pour mesurer l'humidité du sol : gravimétrie, sonde à neutrons, atténuation gamma, résistivité des sols. Pour maintenir le coût d'un système d'acquisition modéré et donc permettre le déploiement d'un réseau dense la méthode par résistivité des sols a été choisie. Celle ci se fonde sur une propriété des sols : être d'autant plus conducteurs que le sol est la teneur en eau est élevée. Elle mesure la tension entre deux électrodes enfoncée parallèlement dans le sol. Cette méthode simple présente toutefois plusieurs limitations. Il s'agit tout d'abord d'une mesure indirecte qui n'a qu'une signification relative en un site donné. La résistivité dépend d'autres paramètres tels que la nature du sol ou sa salinité. Par ailleurs la corrosion, éventuellement accélérée par l'effet électrolytique tend à modifier la réponse des capteurs dans le temps. Même si du fait de l'architecture **perletta** cet effet est modéré par le fait que les capteurs ne sont en tension que très peu de temps, il ne saurait être négligé sur de longues séries.

Le capteur d'humidité HW103 se compose de deux éléments (figure 8.a) reliés par **C** un câble à deux fils. A, le premier élément, est un pont de résistance qui mesure la conductivité de **B** , la sonde fichée dans la terre. L'élément **A** se trouve dans le boîtier perletta et l'élément **B** sera relié par un câble dont la longueur permettra une mesure à diverses profondeurs.

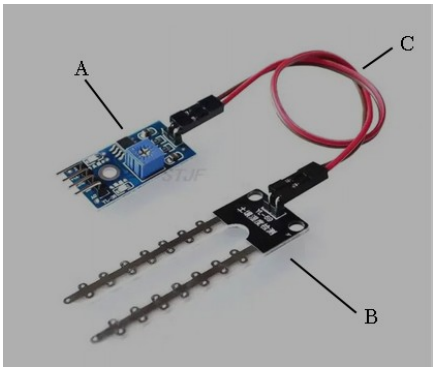

Figure 8.a : Sonde d'humidité HW103

Le système **perletta7** comporte trois sondes. L'une d'entre elles se trouve en surface et les deux autres, dotées de câbles d'une longueur comprise entre 50 cm et 1m, seront enterrées. La profondeur à laquelle ces sondes seront placées dépendra de la nature local du sol. Pour ce motif, il est important lors de l'installation de noter les profondeurs effectives. Lorsque la sonde est à l'air libre, sa résistivité est très élevée et la valeur renvoyée est alors de 1023. En revanche si les deux électrodes sont reliées par un conducteur métallique la résistance devient très faible et la valeur retournée est alors 0. Le montage des éléments se fera suivant le schéma de la figure 8.b.

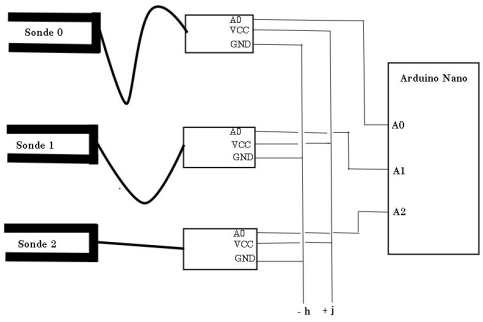

Figure 8.b : Raccordement des sondes d'humidité

Le connecteur **GND** de chaque sonde sera relié à la barrette **h** du fond de panier et le connecteur **VCC** à la barrette **j**. Pour leur part, les connecteurs **A0** des sondes 0, 1 et 2 seront respectivement reliées aux broches **A0**, **A1** et **A2** de l'Arduino-Nano. Sur chaque sonde le connecteur **D0** restera non connecté et le potentiomètre n'a pas à être réglé.

La fabrication du câble de connexion des sondes enterrées demande quelques précautions. D'une part, il devra être soudé du coté des électrodes (**B**) mais se terminera par des connecteurs amovibles du coté du pont de résistance (**A**). De la sorte le système pourra être retiré du terrain sans démonter les sondes. D'autre part, les raccordements sur la sonde devront être noyés dans une colle silicone (ou équivalente) qui augmentera la résistance mécanique de l'ensemble et isolera les soudures entre elle de telle sorte que la résistivité de l'ensemble ne dépende que des électrodes.

Annexe 9 : Programme d'exploitation

Cette annexe contient le programme d'exploitation perletta7. Ce programme est écrit en langage C qui est l'outil de développement de l'environnement Arduino. Dans cette version les données sont enregistrées sur carte SD, la carte radio (LoRa) n'est pas encore active.

```
/*
Gestion d'une station pluvio-humidite
J.C. Berges - janvier 2023
Temperature sol
JCB Mars 2024

TPL5110 --> D3(DONE)
DS1820  --> 5V GND D5(data) + resistance 5 KOhm D5-5V
HUM0   --> 5V GND A0
HUM1    --> 5V GND A1
HUM2    --> 5V GND A2
MPX5010 --> 5V DND A3
DS3231            --> 5V GND A4(SDA) A5(SCL)
SD      --> 5V GND D13(SCK) D12(MISO) D11(MOSI) D4(CS)
SX1278  --> 3.3V GND D13(SCK) D12(MISO) D11(MOSI) D10(NSS) D9(RST) D2(D100)
*/

/*
Les lignes suivantes sont utiles pour indiquer quelles bibliothèques vont être utilisées dans ce programme
*/
#include <EEPROM.h>
#include <DS323x.h>
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include <string.h>
#include <LoRa.h>
#include <OneWire.h>
#include <DallasTemperature.h>

Sd2Card card;

#define ONE_WIRE_BUS 5
#define HDR  "MAND00  "
#define TLR  "##"
#define FIC   "PERLETTA.CSV"
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
DS323x rtc;
File fifi ;
```

```
/*
Dans l'environnement Arduino le programme setup s'exécute une fois après chaque mise en tension. Pour le système perletta il contient l'ensemble des drectives nécessaires à l'acquisition et à l'enregistrement des données.
*/
void setup() {

  int k , pl , h2 , h1 , h0 , wawa , itmp ;
  char dtim[20] , fic[14] ;
  char lor[250] , *lol ;
  float tmp ;
/*
On commence par mettre en niveau bas la liaison avec le TPL5110 pour éviter qu'il ne coupe l'alimentation immédiatement.
*/
  pinMode(3,OUTPUT) ;
  digitalWrite(3,LOW) ;

/*
Démarrage de la carte SD
*/
  card.init(SPI_HALF_SPEED,4) ;
  SD.begin(4) ;
  lol = lor ;

/*
Lecture du nom de la station
*/

  for (k = 0 ; k < 8 ; k++)
    lol[k] = EEPROM.read(k) ;
  lol[8] = 0 ;

/*
Lecture de l'heure
*/

  Wire.begin();
  delay(2000);
  rtc.attach(Wire);
  sensors.begin();
  delay(2000) ;
  DateTime now = rtc.now();
```

```
  for (k = 0 ; k < 16 ; k++)
  dtim[k] = now.timestamp()[k] ;
  dtim[16] = 0 ;

  lol += 8 ; *lol = 32 ;
  for (k = 0 ; k < 16 ; k++)
    lol[k+1] = dtim[k] ;
  lol[17] = 0 ;
  lol += 17 ;

/*
Lecture de la tension de référence pour tester l'état des piles
*/

  ADMUX = 0x4F;
  delayMicroseconds(5);
  ADMUX = 0x4E;
  delayMicroseconds(200);
  ADCSRA |= (1 << ADEN);
  ADCSRA |= (1 << ADSC);
  while(ADCSRA & (1 << ADSC));
  wawa =  ADCL | (ADCH << 8);
  sprintf(lol," TREF %d %c",wawa,0) ;
  lol += strlen(lol) ;

/*
Quatre fois de suite sont mesurée : la pression dans la colonne (pluviomètre), la résistance des trois sondes et
la température de sol.
*/

  for (k = 0 ; k < 4 ; k++)
  {
    pl = analogRead(A3) ; delay(200) ;
    h2 = analogRead(A2) ; delay(200) ;
    h1 = analogRead(A1) ; delay(200) ;
    h0 = analogRead(A0) ; delay(200) ;
    sensors.requestTemperatures(); delay(200) ;
    tmp = sensors.getTempCByIndex(0);
    itmp = tmp * 10 ;
    sprintf(lol,"PL %d H2 %d H1 %d H0 %d TMP %d %c",
      pl,h2,h1,h0,itmp,0) ;
    lol += strlen(lol) ;

  }
```

```
/*
Toutes les informations collectéées sont enregistrées dans la carte SD (fichier PERLETTA.CSV)
*/
  fifi = SD.open("/") ;
  fifi.close() ;
  delay(200) ;
  fifi = SD.open(FIC,FILE_WRITE) ;
  fifi.println(lor) ;
  fifi.close() ;

/*
La ligne reliée au TPL5110 est montée pour mettre le système hors tension
*/

  digitalWrite(3,HIGH) ;
  delay(1000) ;
  digitalWrite(3,LOW) ;
}

/*
La section loop doit s'exécuter indéfiniment après setup. Ele est ici sans objet puisque le système est mis horstension en fin de setup
*/
void loop() {

}
```