
Relevé de température

Arduino - NAS

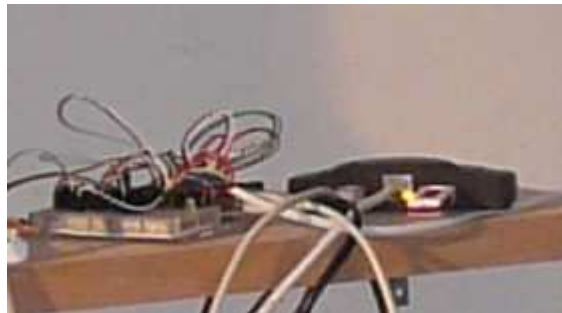
Emmanuel Pottier - 2011-11-02

Objectifs du projet

L'objectif de ce projet est de mettre en place une mesure de température à distance, avec enregistrement des valeurs en interne (data logger) et possibilité de lecture par internet.

Description

Du fait du matériel dont je disposai, j'ai choisi de mettre en place un système à base d'Arduino et de recycler un serveur USB-NAS (modèle 652 WLXKJ), habituellement utilisé pour accéder par le réseau à un disque usb.



Vue d'ensemble, la taille de la clef USB donne une indication de la taille réelle

L'idée est donc que :

1. L'arduino effectue la lecture du capteur
2. Puis il renvoie la valeur de température sur le port usb du boîtier NAS
3. Le boîtier NAS recopie la valeur reçue dans un fichier, sur une clef USB
4. Le fichier est ensuite disponible sur le NAS, soit dans le voisinage réseau, soit en connexion ssh/scp directement sur le NAS.

Les avantages du serveur NAS sont les suivants :

- accès natif au réseau
- prise en charge deux ports USB (l'un servira pour l'arduino, l'autre pour une clef usb)
- firmware alternatif disponible (Snake OS), permettant un accès complet - ssh - au système Linux embarqué
- le NAS permet par ailleurs d'alimenter l'ardunio via le port usb, il n'y aura donc qu'une seule alimentation, alimentation qui sera par ailleurs économe (quelque watts).

Pour ~30€, le NAS réunit donc les fonctions de data logger, de shield ethernet et d'alimentation. Il dispose par ailleurs d'une grosse réserve de puissance pour de futures évolutions (cadencé à 200Mhz, 256Mo de mémoire, possible de brancher des disques de plusieurs centaines de Go).

La mesure de temperature en elle même se fera par l'arduino, via des capteurs TMP36 branchés sur les entrées analogiques.

Matériel nécessaire / prérequis

Le matériel nécessaire est le suivant :

- Un boîtier USB NAS, modèle 652WLXKJ, disponible notamment chez dealextreme (<http://www.dealextreme.com/p/standalone-bittorrent-bt-client-upnp-usb-nas-ftp-samba-printer-sharing-network-lan-server-20383>)



- Une carte Arduino (Arduino Uno dans mon cas, mais n'importe quel clone devrait fonctionner)
- Un capteur TMP36, qui produit une tension proportionnelle à la temperature
- Une clef USB, un vieux modèle de 128Mo sera largement suffisant

Compter 30 € pour le boîtier nas, et a peu près autant pour l'arduino avec 2 ou 3 capteurs TMP36 et une vieille clef USB, soit un total de 60 €.

On suppose dans ce descriptif que le lecteur est familier de l'arduino, sait se connecter en ssh sur une machine et mettre jour un firmware.

Programmation

Arduino

Le programme de l'arduino est assez basique puisqu'il s'agit simplement d'effectuer la lecture du capteur, puis d'envoyer les valeurs sur le port série.

Le montage est détaillé sur le site de "ladyada" : <http://www.ladyada.net/learn/sensors/tmp36.html>

Le capteur TMP36 à pour avantage de permettre une mesure simple des temperatures usuelles en habitation, y compris pour les valeurs négatives. La formule qui donne la temperature est la suivante :
 $T^{\circ}\text{C} = (V-500)/10$

Dans lequel V est la tension à la borne du capteur, en mV.

Il est intéressant de noter que 1v en sortie du capteur correspond en pratique à 50°C. On peut considérer que l'on ne dépassera jamais cette temperature. On peut en conséquence fixer la référence analogique de l'Arduino sur "INTERNAL", soit 1.1 volts, pour obtenir une précision de lecture maximale.

La formule devient alors : $T^{\circ}\text{C} = (\text{analogRead}(\text{sensorPin}) * 1100 / 1024 - 500) / 10$

Remarque importante : du fait de la gestion des nombres a virgule flottante sur l'arduino, il est préférable d'effectuer les multiplications avant les division, et plus généralement, de limiter le nombre d'opérations. Enfin, il est bon de toujours ajouter ".0" aux entiers pour s'assurer que l'opération est effectuée sur des flottants, et non convertie en entiers.

La formule devient alors $T^{\circ}\text{C} = (\text{analogRead}(\text{sensorPin}) * 1.074 - 500.0) / 10.0$

En pratique, la référence interne n'est pas exactement de 1.1v. Dans mon cas, après étalonnage, la formule correcte sera : $T^{\circ}\text{C} = (\text{analogRef}(\text{sensorPin}) * 1.06157 - 500.0) / 10.0$

Le script final est donc :

```
void setup(){
  // Declaration entrees analogiques
  analogReference(INTERNAL);

  // Declaration de la liaison serie
  Serial.begin(9600);
}

float ReadTemp(int analogPin){
  // Lecture d'une temperature sur la pin specifiee
  float sensorTemp = (analogRead(analogPin) * 1.06157 - 500) / 10.0;
  return sensorTemp;
}

void loop(){
  float TmpInt = ReadTemp(0); // Le capteur Interieur est
connecté sur le port analog 0
  float TmpExt = ReadTemp(1); // Le capteur exterieur est
connecté sur le port analog 1

  // Ecriture des valeurs sur le port série
  Serial.print(TmpInt);
  Serial.print(";");
  Serial.print(TmpExt);

  // Attente avant de refaire une mesure
  delay(10000);
}
```

USB-NAS

Mise a jour du firmware : par défaut, le NAS est livré avec un firmware qui ne permet pas vraiment la personnalisation. Il convient donc de commencer par flasher le firmware pour installer "snake os", disponible sur google code (<http://code.google.com/p/snake-os/>). La procédure de mise a jour est détaillée sur le site du projet, et ne devrait pas poser de problème particulier.

Il faut ensuite s'assurer que le NAS reconnaît l'Arduino, et est capable de dialoguer avec lui. Pour les arduino équipés d'un chip usb ftdi (la plupart, sauf arduino Uno), la reconnaissance devrait être automatique. Pour l'arduino Uno, le chip USB a changé, ce qui impose d'utiliser un module spécifique, "cdc-acm" (voir plus loin la procédure de chargement du module).

Une fois l'arduino branché, il suffit de se connecter en ssh sur le nas, puis de taper "dmesg" pour voir si le système est bien reconnu, il devrait alors s'afficher quelque chose de semblable :

```
usb 2-2: new full speed USB device using str8100-ohci and address
2
usb 2-2: configuration #1 chosen from 1 choice
cdc_acm 2-2:1.0: ttyACM0: USB ACM device
```

Ici, l'arduino est reconnu comme ttyACM0 (la valeur peut être différente pour les autres versions d'arduino). Il sera donc accessible par /dev/ttyACM0.

S'il manque la dernière ligne (ou un équivalent), c'est que l'arduino n'est pas reconnu.

On peut alors voir ce que l'arduino envoie en tapant : `cat /dev/ttyACM0`

On devrait alors voir apparaître à chaque seconde les températures, sous un format comme :
21.45;16.39

Comme on souhaite envoyer le résultat directement dans un fichier, on va faire une redirection, couplée à un "nohup", pour que la redirection soit toujours active lorsque l'on se déconnecte :

```
nohup cat /dev/ttyACM0 >> /usb/sda1/data.log &
```

On peut voir en tapant "top" que le script continue de tourner en arrière plan.

Il suffit maintenant de lire le fichier "data.log" de la clef usb pour accéder aux données. On peut notamment le faire en se connectant en ssh, puis en tapant `cat /usb/sda1/data.log`, ou encore en configurant dans snake-os un partage de fichier sur la clef usb. On accède alors directement au fichier par le partage réseau.

A noter qu'il est intéressant d'insérer dans le fichier la date/heure à interface régulier.

Cela peut se faire assez simplement, en allant sur le NAS, puis éditer le fichier `/etc/cron.d/root`, et ajouter :

```
0 * * * * date +%F_%T >> /usb/sda1/data.log
```

Remarque : il est nécessaire de sauvegarder le fichier cron, puis la configuration sur le NAS (par l'interface web), puis de le relancer pour que la tâche soit prise en compte.

Cas de l'arduino Uno

L'arduino Uno utilise un chip usb différent des arduino précédents. De ce fait, il requiert un module supplémentaire pour être reconnu par le NAS.

Le module `cdc-acm.ko` est disponible en téléchargement (je l'ai généré à l'aide du SDK de snake-os : <http://pottier.emmanuel.free.fr/cdc-acm.ko>).

il suffit de le placer sur la clef usb, puis de le lancer à l'aide de la commande : `insmod /usb/sda1/cdc-acm.ko` (si tout se passe bien, la commande ne renvoie rien).

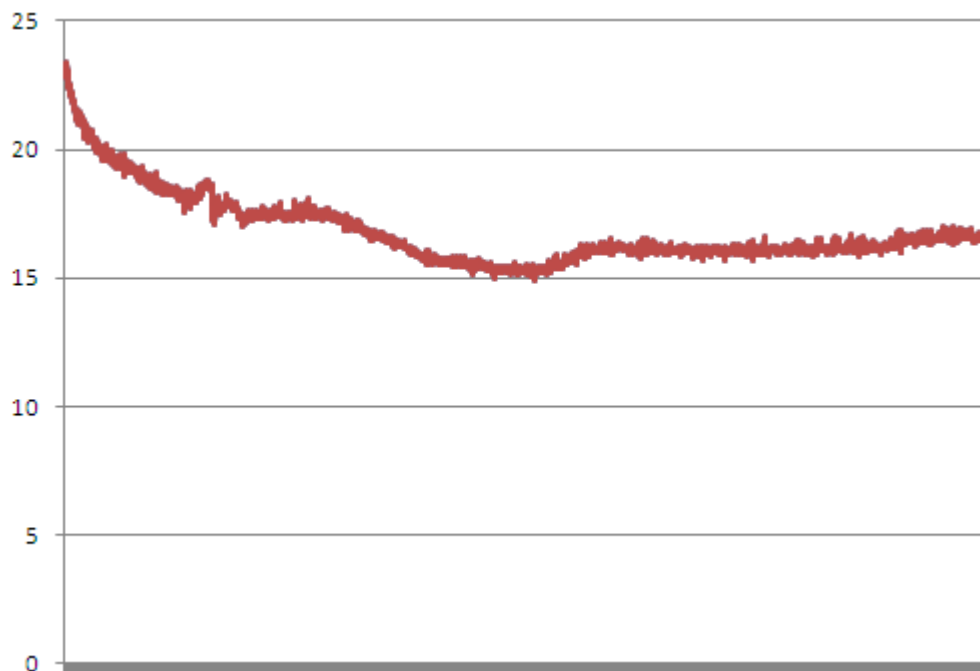
En pratique / par où commencer

Si vous êtes perdus parmi les choses à faire, voici le mode opératoire qui me semble le plus simple.

1. Commencer par l'arduino
 - a. Faire le montage sur l'arduino, avec le capteur TMP36
 - b. Faire le programme, et vérifier avec le moniteur série que vous recevez bien les valeurs à intervalle régulier
2. Sur le boîtier NAS
 - a. Flasher le firmware pour snake-os
 - b. Se connecter en ssh sur le NAS pour s'assurer que tout fonctionne correctement
3. Connecter l'arduino au NAS
 - a. Vérifier avec `dmesg` que l'arduino est bien reconnu
 - b. Vérifier avec `cat /dev/ttyACM0` que vous pouvez lire les valeurs
 - c. Mettre en place le "nohup" pour que les valeurs soient automatiquement recopiées dans la clef USB
 - d. Mettre en place la tâche "cron" pour avoir des timestamp dans le fichier de log

Aller plus loin

Ce projet est mon premier système "complet" réellement utilisable à base d'Arduino. Il me sert à visualiser la température dans une maison que je n'occupe pas à l'année.



*Résultat des mesures sur 3 jours, début novembre
(post traitement des données sous Excel)*

Il y a encore beaucoup de potentiel inexploité. Parmi les évolutions possibles, on pourrait citer :

- l'utilisation du serveur web intégré au NAS pour afficher la température, voire des graphiques, sur une page web
- le déclenchement d'actions à distance vers l'arduino, ce qui supposerait de pouvoir envoyer des données sur NAS vers l'arduino. C'est probablement possible, mais il faudrait le tester plus en détail
- l'envoi automatique d'email depuis le NAS (par exemple une "alerte gel"), c'est théoriquement possible, mais il semble que le NAS ne dispose pas par défaut d'une fonction d'envoi d'email.

Par ailleurs, il faut pour l'instant relancer le "nohup ..." (et le insmod le cas échéant) à chaque reboot du NAS. Il serait bon d'automatiser cette tâche

Liens

- Montage à base de TMP36 : <http://www.ladyada.net/learn/sensors/tmp36.html>
- Firmware snake os : <http://code.google.com/p/snake-os/>
- Boîtier NAS chez dealextreme (des boîtiers compatibles existent probablement) : <http://www.dealextreme.com/p/standalone-bittorrent-bt-client-upnp-usb-nas-ftp-samba-printer-sharing-network-lan-server-20383>
- Module arduino : chez n'importe quel revendeur Arduino. J'ai utilisé ce module plutôt pratique pour débiter : <http://www.oomlout.co.uk/prototyping-bundle-for-arduino-ardp-p-186.html>
- Le fichier cdc-acm.ko : <http://pottier.emmanuel.free.fr/cdc-acm.ko>

Remerciements

A Ricardo Gomes da Silva, pour m'avoir expliqué comment compiler le module cdc-acm
A Douglas Gazineu, pour avoir créé le firmware Snake-OS et le SDK
Et bien sûr à toute l'équipe Arduino pour avoir créé ce merveilleux outil.

Copie, modification et diffusion autorisées
