


Arduino Python Multi-Capteur 2.4Ghz

Capteurs de températures, de pression, d'humidité sur un Arduino et envoyés par émetteur 2.4Ghz

 Difficulté Moyen

 Durée 1 heure(s)

 Catégories Électronique

 Coût 20 EUR (€)

Sommaire

Introduction

Étape 1 - Le montage

Étape 2 - Programmation Arduino

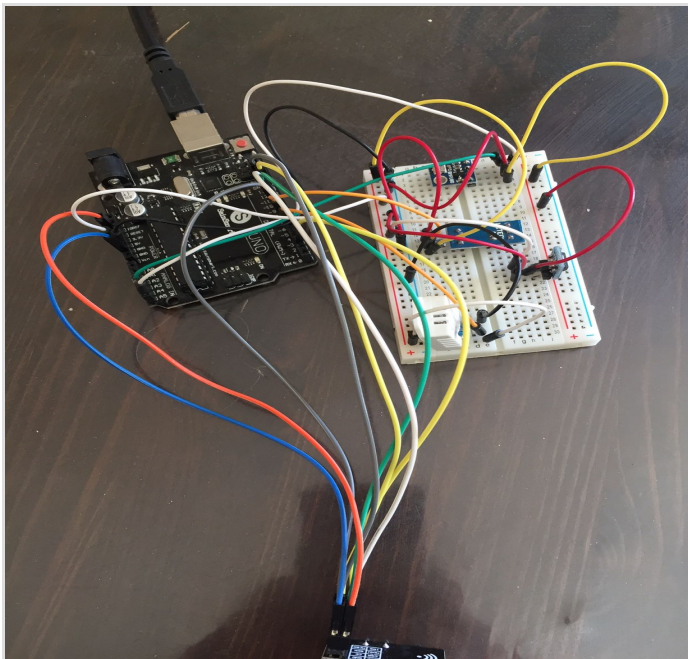
Étape 3 - La réception

Commentaires

Introduction

Avec ce tutoriel vous apprendrez à utiliser Arduino et des capteurs (température, pression et humidité) plus émission/réception en 2.4Ghz. Et en plus la réception sur un Raspberry pi en Python. L'idée m'est venue en voulant ajouter la pression et l'humidité à mon capteur de température extérieur.

Pour en savoir plus sur mes projets, visitez mon site.



Matériaux

- Un Raspberry moi c'est un PI Model B.
- Un Arduino uno
- 2 émetteurs/récepteurs en 2.4Ghz
- Un capteur pour la température extérieur : DS18B20
- Un capteur d'humidité (il fait aussi température) : DHT22
- Un capteur de pression (il faut aussi la température) : BMP180

Outils

- Un ordinateur pour la programmation Arduino
- Plaque d'essai et fil de connexion.

<https://github.com/Michelgard/Multi-capteur-Arduino-avec-emetteur-2.4Ghz>

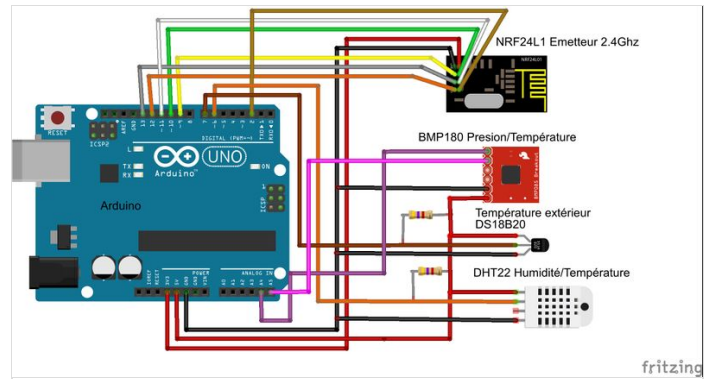
http://ouiaremakers.com/wp-content/uploads/formidable/capteur_multietNF24.ino

Étape 1 - Le montage

Un capteur de température intérieur, un capteur de température extérieur, un capteur d'humidité et un capteur de pression le tout connecté à un Arduino et les valeurs transférées par un émetteur en 2.4Ghz. Et pour la réception Raspberry, récepteur 2.4Ghz et du python. Voilà l'objet de ce petit tuto.

Pour le montage de l'émetteur voici le schéma. Rien de très compliqué mais il y a beaucoup de fil...

Bien penser aux résistances de 4.7K sur le récepteur de température et d'humidité. Et surtout attention l'émetteur 2.4Ghz fonctionne sous 3.3V.



Étape 2 - Programmation Arduino

Pour la programmation de l'Arduino il faut quelques biblio :

Le capteur de pression : Barometer, SPI et Wire . Je vous ajoute tous les liens sur mon Github :

- Le capteur d'humidité : DHT22
- Le capteur de température : DS18D20
- L'émetteur 2.4Ghz : RF24

Pour le programme de l'Arduino je vais pas le mettre ici en intégral mais vous pouvez le trouver sur mon Github.

Par contre quelques informations sur son fonctionnement. Les données à envoyer sont préparées avant l'envoi dans un tableau de char de cette façon :

On crée le char avec une valeur de départ : char SendPayload[32] = "RS1:";

Ensuite on ajoute le valeurs. Chaque valeur est précédé et suivie d'une lettre permettant à la réception son repérage. Par exemple pour la température :

```
strcat(SendPayload, "T"); dtostrf(txbuffer[0],2,2,tempstr); strcat(SendPayload,tempstr); strcat(SendPayload, "T");
```

Et ensuite le char complet est envoyé : bool ok = radio.write(&SendPayload,sizeof(SendPayload));

Mais comme il y a trop de donnée et que le nombre de caractère est au maximum de 32 il faut faire deux char envoyés à la suite. Chaque char sera repéré à la réception par son numéro "RS1" et "RS2".

Étape 3 - La réception

Pour la réception nous avons un montage plus simple. Un raspberry pi et un récepteur en 2.4Ghz. Il faut configurer le Raspberry Pi pour le récepteur tout est dans ce fichier pdf ci-joint : Config Raspberry et Récepteur 2.4Ghz.pdf

Et pour finir un fichier en Python pour la réception et l'enregistrement des données dans une base de données.

Pour la réception je lui fait faire deux boucles pour avoir mes deux char qui sont décortiqués pour à la fin des deux boucles avoir toutes les données qui sont, pour finir envoyer dans la base.

Le programme en Python : multi_capteurs-NF24.py. Pour fonctionner il lui faut un programme de config en xlm pour les données de connexion à la base : config

Et le sql pour créer la table dans votre base de donnée : Table SQL
Voilà. Bon bricolage. La suite sera un affichage sur mon DashScreen des données collectées.

Vous trouverez tous les documents et codes sur Github. Et d'autre info sur mon blog : www.blog-de-michel.fr

