23/05/2018

METEOCAPTOR

Application affichant la température, l'humidité et une capture du temps extérieur.

Adam MEBARKI – Rayan MEHDI
LICENCE PROFESSIONNELLE INFORMATIQUE EMBARQUEE ET MOBILE

Description du projet

MétéoCaptor est une application qui offrira la possibilité aux utilisateurs de recevoir une photo de l'extérieur, la température et l'humidité sur son smartphone. L'utilisateur pourra ainsi être tenu au courant sur le temps qu'il fait à l'extérieur. Il aura aussi la possibilité d'accéder à une archive répertoriant chaque prise faite par la « station météo ».

Matériels nécessaires pour la station météo

- Un Raspberry Pi 3 : il servira de « station météo ». Il s'occupera donc de capturer une photo de l'extérieur et d'envoyer la température.
- Une Caméra Raspberry Pi 3 rev 1.3 pour la prise de photo.
- Un capteur de température/humidité Adafruit Si7021.
- Un smartphone Android avec un sdk > 19.
- Un écran LCD tactile pour accéder aux paramètres du Raspberry Pi 3.
- Un câble HDMI ou autres câbles permettant la liaison entre l'écran et le Raspberry Pi 3.
- Des câbles USB – micro USB pour connecter le Raspberry Pi3 et le smartphone.
- Un chargeur pour allumer le Raspberry Pi 3.

Technologies

- Android Things à installer sur le Raspberry Pi 3.
- Android Studio pour installer l'application smartphone s'il n'y a pas d'Apk.
- L'API Nearby Connection. Très récente et constamment en cours d'évolution.

Représentation graphique et plan de montage

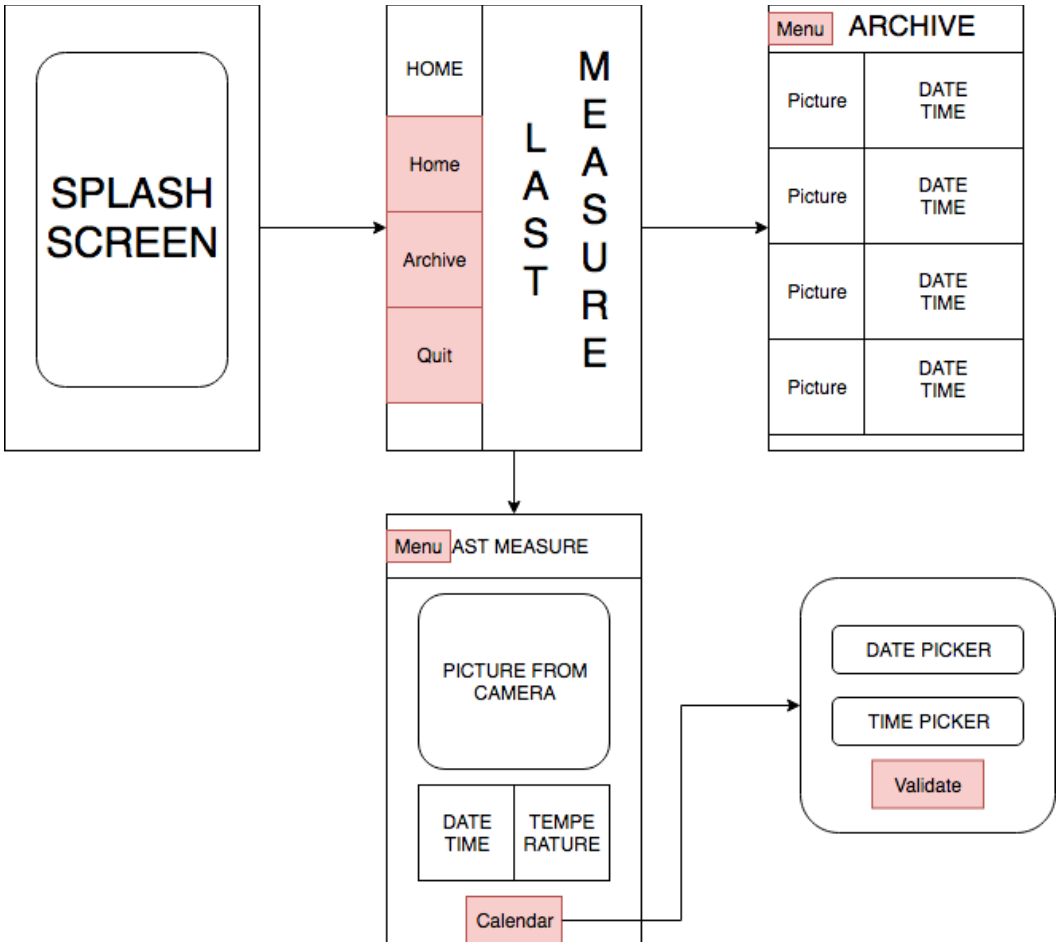
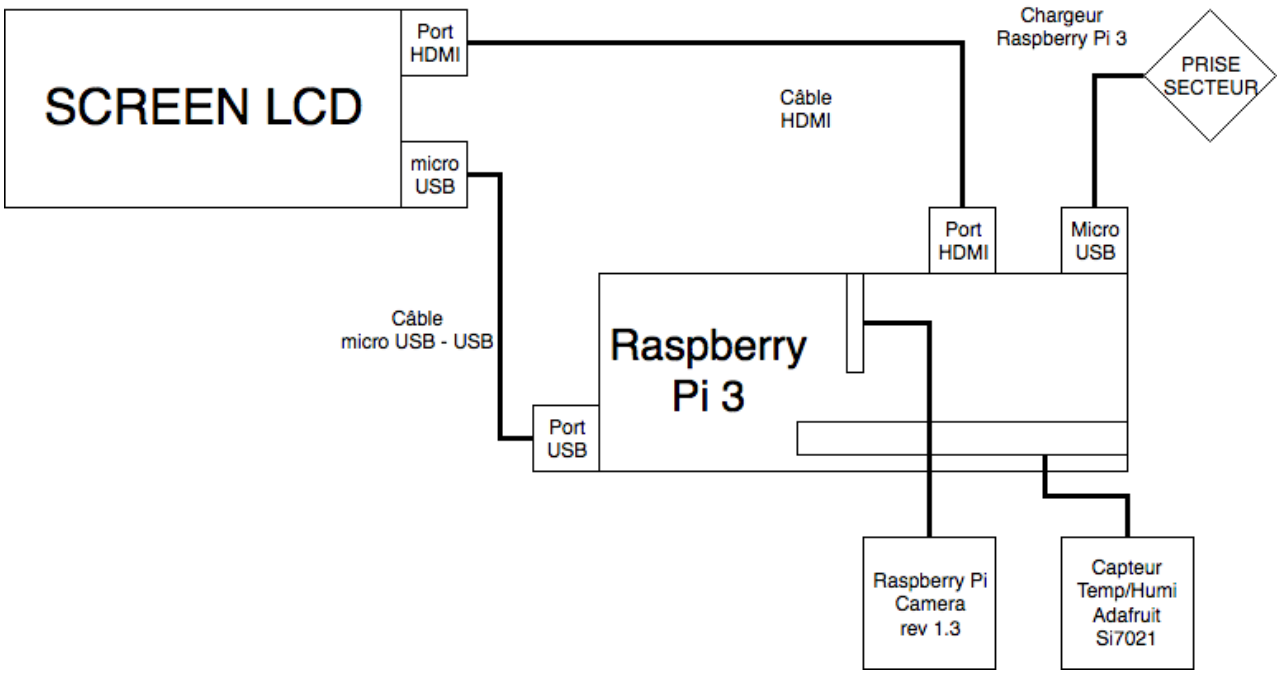
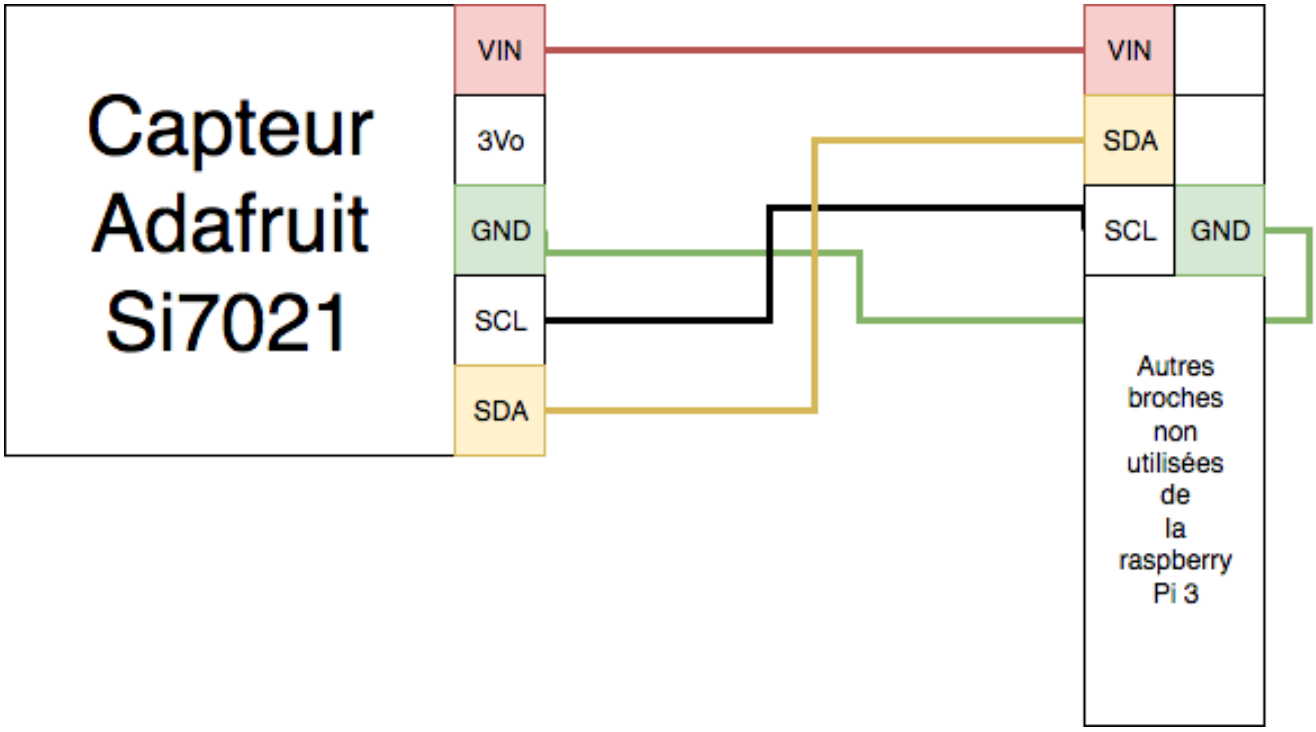


Schéma représentant la structure de l'application Android

Après l'apparition de l'écran d'accueil, l'utilisateur aura la dernière température prise par la « station météo ». Avec le menu burger, il aura la possibilité d'accéder à son archive contenant les précédentes prise du temps extérieur. La fonctionnalité supplémentaire qui peut être ajoutée est de pouvoir capturer le temps à intervalle de temps régulier et de pouvoir le modifier grâce à l'application. Pour l'instant il faut interagir manuellement avec le Raspberry Pi 3.



Plan de montage de la Raspberry Pi 3



Plan de montage du capteur Adafruit Si7021 sur la Raspberry Pi 3

Guide d'installation

Il faut tout d'abord suivre les instructions détaillées que l'on trouve sur le lien suivant : https://developer.android.com/things/hardware/Raspberrypi

Une fois toutes les étapes accompli correctement, AndroidThings est installé sur le Raspberry.

Il faut ensuite créer un projet Android Studio pour Android Things en se basant sur le guide suivant : https://developer.android.com/things/training/first-device/create-studio-project

A partir de là, pour compiler un projet sur le Raspberry, il faut se connecter sur le même réseau que ce dernier et utiliser la commande *adb connect AddressIpDuRaspberry.*

```
iutabgdin202-6:platform-tools iem$ ./adb connect 192.168.43.186
connected to 192.168.43.186:5555
```

Si le message « connected » apparaît, il est possible de « build » directement sur le Raspberry.

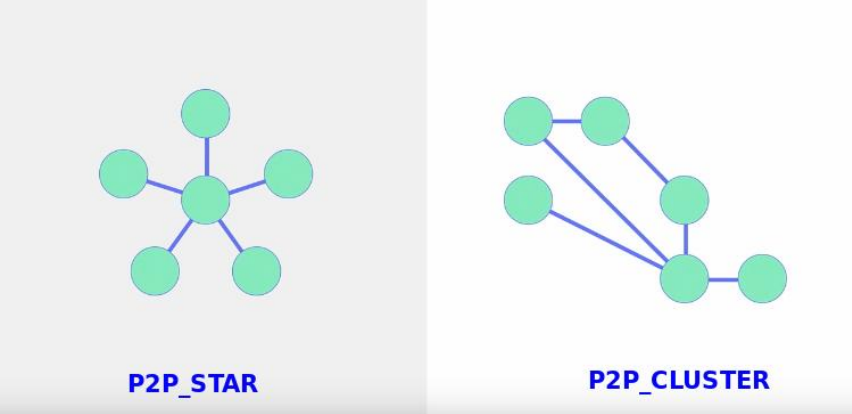
Si vous rencontrez des problèmes de permissions, il faut suivre les instructions suivantes pour donner les permissions au Raspberry : https://github.com/TilesOrganization/support/wiki/How-to-use-ADB-to-grant-permissions.

Connexion au Raspberry Pi 3

Afin d'établir une communication entre le Raspberry Pi 3 et le smartphone, il a été décidé d'opter pour l'API Nearby Connection. Très récente, cette API offre la possibilité de connecter, à l'aide de ressources supplémentaires, deux appareils Android entre elles grâce au Bluetooth et la Wifi. Nearby Connection est une API créée par Google qui propose une connexion réseau en peer to peer et d'échanger des données avec des appareils proches physiquement.

La rasperry Pi 3 support aisément Android Things. Android Things est une version allégée d'Android qui peut-être utilisée sur des appareils à faible consommation d'énergie. Il est notamment utilisé pour la connexion entre appareils, ce qui est le cas du projet. Son utilisation était essentielle.

Le principe de Nearby Connection est simple. Dans un premier temps il faut décider de la stratégie à adopter : P2P_STAR ou P2P_CLUSTER.



Pour le projet le P2P_STAR était le plus adapté. L'idée est que plusieurs smartphones puissent se connecter à la « station météo » et recevoir les données qu'elle possède.

Le fonctionnement de Nearby Connection avec la stratégie P2P_STAR demande deux rôles.

Le premier rôle est celui de **l'advertiser**. Il va avertir tous les objets qui sont connectés à lui qu'il possède des données et il va envoyer les mêmes données à tous les appareils sans distinction.

Le deuxième rôle est le **discover**. Il est celui qui se connecte à l'advertiser pour recevoir les données de celui-ci. Il n'a pas conscience de la présence des autres appareils connectés à l'advertiser.

Dans le projet la Raspberry Pi3 est **l'advertiser** et les smartphones sont les **discovers**.

Fonctionnement de l'application et de la station météo

En ce qui concerne le code de l'application Android Things, nous avons utilisé diverses librairies trouvées sur différents projets github (Voir Sitographie) pour la gestion de nos 2 capteurs (caméra et capteur température/humidité).

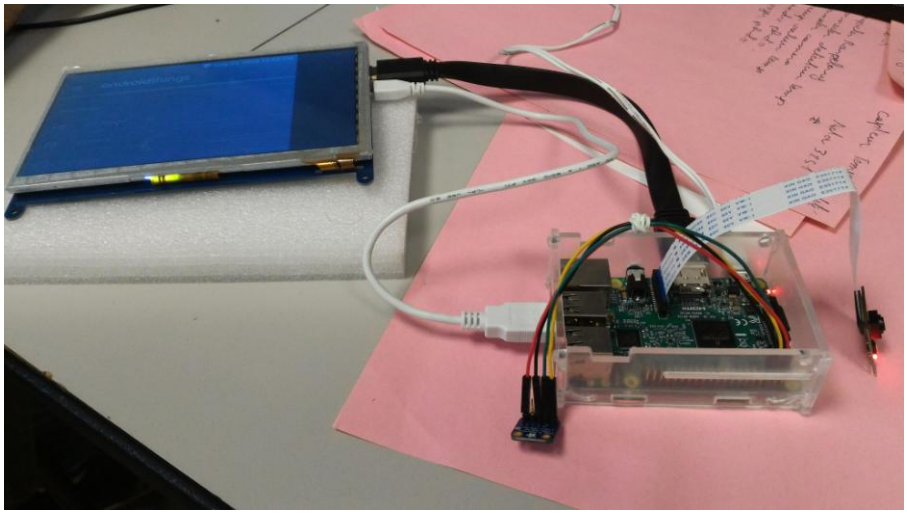
L'intégralité du code métier de l'application se trouve ensuite dans la Main Activity, les captures d'humidité et température se font en continu, pour enclencher la capture et l'envoi de photo, il suffit d'appuyer sur le bouton situé sur l'écran.

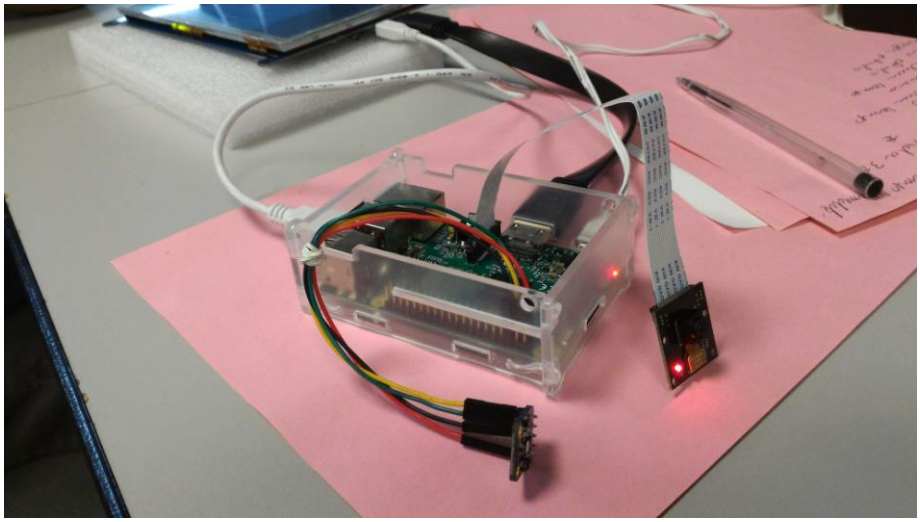
La photo est transformée en tableau de bytes puis transférée à l'appareil Android sous forme d'un fichier .txt avec Nearby Connection. Le nom du fichier contient la température et l'humidité.

De son côté, l'application reçoit donc le fichier txt. Dans un premier temps il y a une vérification sur le fichier. Par la suite on récupère la température et l'humidité sur le nom du fichier reçu puis on le lit afin de récupérer les données. Ces données sont stockées dans un tableau de bytes qui est convertir en Bitmap. Ce bitmap est ensuite affiché avec les outils propres à Android.

Nearby Connection étant récent, le code source pour préparer la connexion entre les appareils est dépassé. Il est recommandé de se documenter sur les documents officiels afin d'avoir l'écriture exacte de la connexion. Cependant la logique reste la même.

Il y a quelques crashs au niveau de l'application quand la connexion entre le Raspberry et le téléphone est perdu. La perte de connexion peut être du à plusieurs facteurs : Wifi, Bluetooth, Alimentation, etc … .





Photos de la station météo montée et en cours d'utilisation

Sitographie

Nearby Connection

https://developers.google.com/nearby/connections/overview

https://developers.google.com/nearby/connections/android/exchange-data

https://code.tutsplus.com/tutorials/google-play-services-using-the-nearby-connections-api--cms-24534

http://androidkt.com/nearby-connections-api-2-0/

Android Things

https://developer.android.com/things/

https://developer.android.com/things/hardware/Raspberrypi

Connexion Capteur Adafruit Si7021 avec Raspberry Pi 3

https://github.com/birdybix/Si7021Driver

Fragment et RecyclerView pour l'application

http://tutos-android-france.com/fragments/

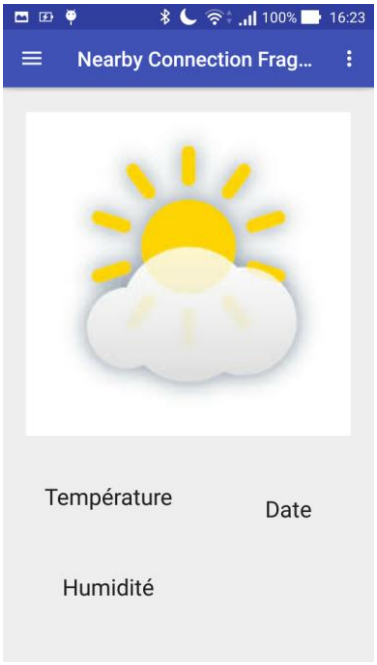
https://developer.android.com/guide/topics/ui/layout/recyclerview

http://tutos-android-france.com/material-design-recyclerview-et-cardview/

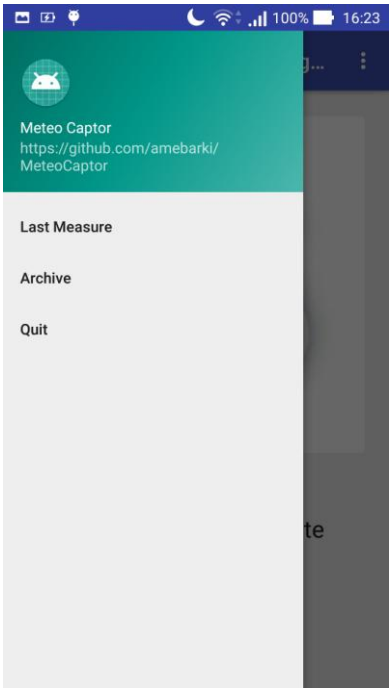
Annexes

Captures d’écran de l’application :

Splashcreen



Ecran d’accueil



Menu Burger

Archives Page