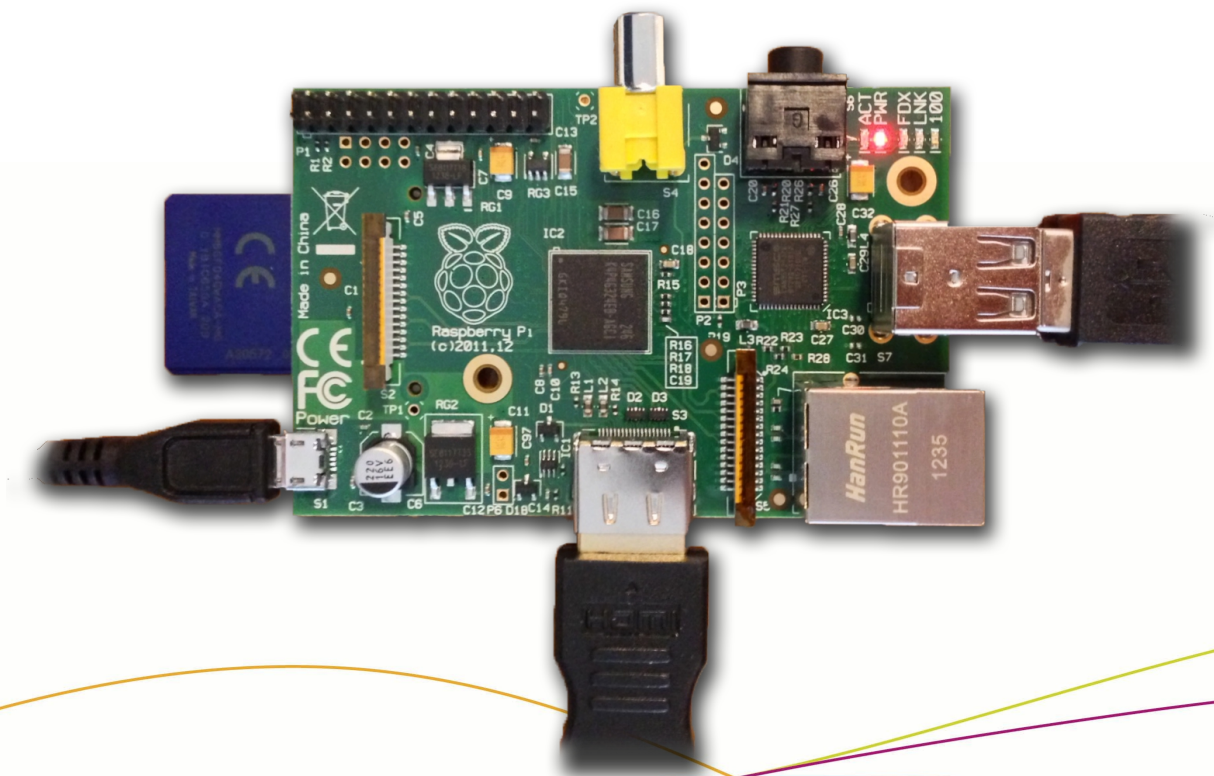


Mise en œuvre des cartes Raspberry Pi



Etudiants :

Hicham BELAHMER

Pierre BESSON

Merlin NIMIER-DAVID

Enseignant-responsable du projet :

Ludovic HENRIET

Date de remise du rapport :

14/06/2013

Référence du projet :

STPI/P6/2013 – 52

Intitulé du projet :

Mise en œuvre des cartes Raspberry Pi

Type de projet :

Expérimental

Objectifs du projet (10 lignes maxi) :

- **Découvrir la carte Raspberry Pi**
- **Réaliser une installation fonctionnelle (branchements, installation et configuration du système)**
- **Connecter la Raspberry Pi à Internet, installer et configurer un serveur web**
- **Interfacer la Raspberry Pi avec une carte Arduino**
- **Concevoir et réaliser un « réveil intelligent »**

Mots-clefs du projet (4 maxi) :

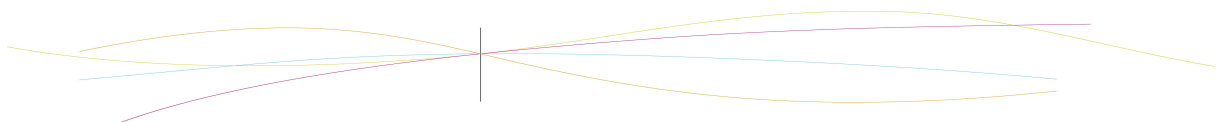
- **linux**
- **embarqué**
- **électronique**
- **web**

TABLE DES MATIÈRES

1. Introduction.....	5
2. Méthodologie / Organisation du travail.....	6
2.1. Travail réalisé en séances.....	6
2.2. Travail autonome et outils de collaboration.....	6
3. Travail réalisé et résultats.....	8
3.1. Mise en œuvre de la carte Raspberry Pi.....	8
3.1.1. Que peut-on faire avec une Raspberry Pi ?.....	8
3.1.2. Préparation de la Raspberry Pi.....	9
3.1.3. Matériel du projet.....	11
3.1.4. Préparation de la Raspberry Pi pour le projet réveil intelligent.....	12
3.2. Le projet « réveil intelligent ».....	13
3.2.1. Fonctionnalités principales.....	13
3.2.2. Technologies utilisées.....	13
3.2.3. Mode d'emploi.....	14
3.3. Interfaçage avec la carte Arduino.....	16
3.3.1. Réalisation d'un capteur de lumière avec une carte Arduino.....	17
3.3.2. Communication entre la carte Arduino et la Raspberry Pi.....	17
4. Conclusions et perspectives.....	19
4.1. Conclusion générale.....	19
4.2. Conclusions des membres du groupe.....	19
5. Bibliographie.....	20
6. Annexes.....	21
6.1. Fiche technique de la Raspberry Pi.....	21
6.2. Listings des programmes réalisés.....	22

NOTATIONS, ACRONYMES

- **RPI** : Raspberry Pi.
- **Arduino** : une carte similaire à la Raspberry Pi mais aux capacités plus limitées.
- **PHP** : PHP Hypertext Preprocessor (langage de programmation web, côté serveur).
- **HTML** : HyperText Markup Language (langage de description des pages web).
- **CSS** : Cascading StyleSheets (langage de mise en forme des pages web).
- **JavaScript** : langage de script web (côté client).
- **Bash** : langage de script des systèmes Unix.
- **GPIO** : General Purpose Input / Output (interface matérielle prenant la forme d'un simple pin, grâce à laquelle on peut lire et écrire des informations digitales).
- **Reddit** : site web composé de *boards* regroupant des actualités autour d'un thème.
- **Git** : système de gestion de versions (code et documents), utilisé également dans le département ASI.



1. INTRODUCTION

La Raspberry Pi, est un ordinateur de la taille d'une carte de crédit, qui a vu le jour il y a environ un an de cela. Depuis, plus d'un million d'exemplaires ont été vendus, devenant ainsi le gadget par excellence d'un grand nombre d'informaticiens et bricoleurs, notamment grâce à sa taille, son champ d'utilisation et son prix très avantageux de 35€.

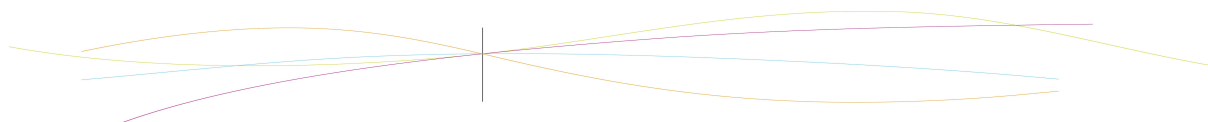
Aujourd'hui, amateurs d'informatique, bricoleurs, enseignants, étudiants, et fondamentalement toute personne aimant créer et réutiliser du matériel, ont trouvé de l'intérêt auprès de cette carte aux usages multiples. En effet, ce petit ordinateur a pu contrôler des robots, monter dans la haute atmosphère dans des ballons-sonde, devenant ainsi la pierre angulaire de n'importe quel gadget dont pouvait rêver l'esprit.

De plus, la Raspberry Pi est de plus en plus considérée comme l'outil ultime pour enseigner l'informatique aux enfants, si bien que Google a décidé de distribuer plus de 15 000 exemplaires dans les écoles anglaises.

Nous, nous sommes en France, et nous avons eu la chance d'en avoir une dans le cadre de notre projet P6, partie intégrante de notre formation d'ingénieur. Nous avons chacun choisi ce sujet pour des raisons similaires, même si les connaissances dans le domaine de l'informatique et de l'électronique au sein du groupe étaient hétérogènes, mais chacun a pu mettre sa patte dans ce projet.

De plus, étant donné l'aspect expérimental du projet, consistant principalement en la mise en œuvre de la carte Raspberry Pi, nous avons pu élargir notre champ de travail en concevant un « réveil intelligent », chose qui nous a permis d'intervenir dans les domaines de l'informatique, l'électronique et la gestion de projet.

Dans les pages qui suivent, nous nous intéresserons dans un premier temps à la méthodologie adoptée pendant ce projet. Ensuite, concernant le travail réalisé et les résultats obtenus, nous nous pencherons plus en détail sur les caractéristiques techniques de la carte Raspberry Pi, sur son utilisation principale ainsi que sur notre projet personnel de réalisation d'un réveil intelligent.



2. MÉTHODOLOGIE / ORGANISATION DU TRAVAIL

La répartition des tâches s'est effectuée comme suit :

Hicham Belahmer	Pierre Besson	Merlin Nimier-David
Installation et configuration du système	Installation et configuration du système	Installation et configuration du système
Installation et test de plusieurs outils	Installation et test de plusieurs outils	Installation et test de plusieurs outils
Installation et configuration du serveur web	Installation et configuration du serveur web	Installation et configuration du serveur web
Montage électronique	Montage électronique (soudure et vérifications)	Montage électronique (soudure)
	Programmation du logiciel Arduino pour détection du changement de luminosité	Programmation de l'interface de configuration et de l'affichage des informations
	Programmation du script de réveil	Réalisation de la vidéo de démonstration

2.1. Travail réalisé en séances

Nous nous sommes réunis (presque) chaque semaine pour une séance de travail d'une heure trente. Ces séances ont été particulièrement utiles durant la phase d'installation, lors de laquelle nous avons besoin d'intervenir sur le matériel et de tester de nombreuses choses.

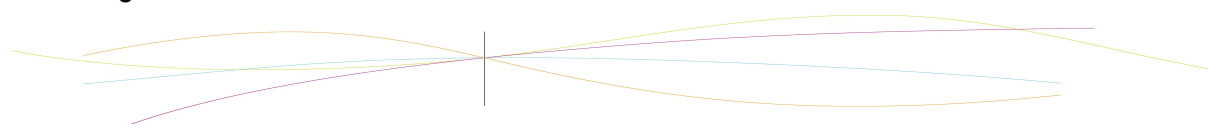
Ce fut également l'occasion de nous mettre d'accord sur la direction à emprunter, de définir plus précisément le projet « réveil intelligent » et de se renseigner sur plusieurs points techniques.

Enfin, c'est lors de ces séances que nous avons conçu et assemblé le montage électronique sur le *shield* Arduino, avec l'aide de M. Henriët.

2.2. Travail autonome et outils de collaboration

Mais de nombreuses heures de travail ont également été réalisées en hors de ces séances. En effet, toute la programmation et de nombreux tests ont été effectués chez nous, sans avoir accès au matériel. Nous avons compensé ce manque en reproduisant au mieux les conditions du projet sur notre équipement, soit grâce à une carte Raspberry Pi personnelle, soit en installant un système (distribution Linux, serveur Apache) similaire sur notre ordinateur.

Il fallait également se former sur plusieurs points techniques, comme par exemple la configuration du serveur web sur Raspberry Pi, ou bien l'utilisation des **GPIO** via script **bash**. Nous avons d'ailleurs eu le plaisir de constater que ces sujets sont bien documentés sur Internet grâce à une communauté forte.



De plus, une vidéo de présentation du projet, ayant pour vocation de présenter succinctement les fonctionnalités principales pendant l'oral de soutenance, a été tournée pendant nos temps libres.

Afin de collaborer de manière efficace, nous avons fait usage de trois outils :

- Les **e-mails** et **SMS** pour communiquer rapidement entre les séances
- Un dossier partagé **Google Drive** pour mettre en commun la rédaction du rapport et la préparation des diapositives de l'oral de soutenance
- Un **dépôt Git**, hébergé chez BitBucket, pour la synchronisation et l'archivage du code, avec gestion des versions

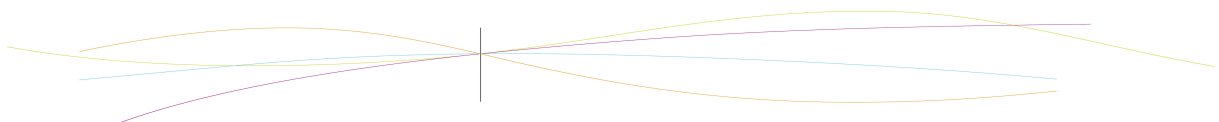


Figure 1: Bureau graphique sur la Raspberry Pi

Cependant, cet ordinateur a de très nombreux atouts, notamment son très faible coût, sa petite taille, sa faible consommation d'énergie, sa connectivité réseau et USB. Cela offre de nouvelles possibilités aux étudiants pour développer des projets innovants. De plus le système d'exploitation Linux disponible a été pour nous un grand avantage car nous avons déjà connaissance de cet environnement et de ses outils.

Il faut aussi noter que la Raspberry Pi, contrairement à une carte Arduino, est capable de lancer des programmes Linux classiques comme un navigateur web, un jeu 3D, voire même des applications serveur comme un serveur web, une plateforme de *cloud*, etc.

De plus elle dispose d'entrées et sorties digitales, sous la forme de pins GPIO (General Purpose Input Output). Ces derniers lui permettent de communiquer avec d'autres appareils ou circuits électroniques.



Figure 2: Schéma décrivant les 26 pins d'entrée ou sortie digitales de la Raspberry Pi

3.1.2. Préparation de la Raspberry Pi

La Raspberry Pi étant démunie de disque dur, une carte SD lui sert de mémoire de stockage principale. Il faut donc tout d'abord copier l'image disque du système d'exploitation de cette carte SD.

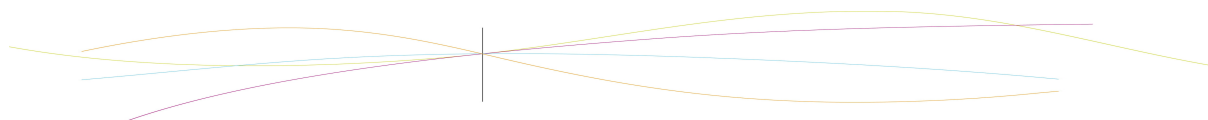
Procédure d'installation de la distribution Linux Raspbian « Wheezy » sur la carte SD (écriture brute des données de l'image disque) depuis un système Linux standard avec accès administrateur (`root`) :

1. On télécharge l'image sur [\[1\]](#).
2. On identifie le chemin d'accès à la carte SD en listant les partitions :

```
$ sudo fdisk -l
```

3. On écrit l'image disque sur la carte SD de manière brute avec la commande `dd`. Attention : il faut bien vérifier le chemin indiqué dans cette commande, sous peine de risquer d'effacer des données présentes sur l'ordinateur.

```
$ sudo dd bs=4M if=<chemin de l'image.img>
of=/dev/<identifiant de la carte sd>
```



4. Insérer la carte SD dans la Raspberry Pi.
5. Brancher l'écran, la souris, le clavier **puis** l'alimentation.
6. La Raspberry Pi démarre. Si aucune interface graphique n'est lancée, tapez simplement :

```
| $ startx
```

7. Par défaut, les identifiants sont les suivants :

- Nom d'utilisateur : pi
- Mot de passe : raspberry

Une fois le système correctement installé sur la carte SD, quelques étapes de configuration sont nécessaires.

Configuration générale de la Raspberry Pi :

La distribution Raspbian « Wheezy » propose une interface graphique pour choisir la langue, la disposition du clavier, le fuseau horaire, activer l'accès SSH (prise de contrôle par le réseau), etc. On y accède par la commande :

```
| $ raspi-config
```

Nous avons dû résoudre un problème que nous rencontrions avec notre écran. En effet, celui-ci était connecté via un câble HDMI → DVI et n'était pas reconnu au démarrage. Pour permettre la reconnaissance de notre écran il nous a fallu forcer l'affichage HDMI au démarrage. Pour cela on a modifié (en root) le fichier `/boot/config.txt` :

```
| $ sudo nano /boot/config.txt
```

Il faut dé-commenter la ligne `hdmi_force_hotplug` en retirant le `#` au début de la ligne.

Il nous a également fallu configurer la sortie audio : par défaut, le son est envoyé par la sortie HDMI. Si l'on souhaite plutôt brancher des enceintes ou des écouteurs sur la prise mini-jack, il faut utiliser cette commande dans le terminal :

```
| $ sudo amixer cset numid=3 1
```

Pour utiliser à nouveau la sortie son via HDMI, il suffira de taper :

```
| $ sudo amixer cset numid=3 2
```

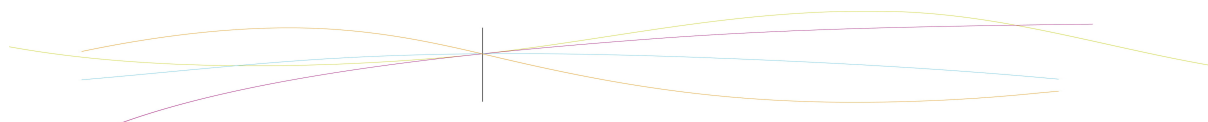
Connecter la Raspberry Pi à Internet

Il est possible de connecter la Raspberry Pi à Internet via le port Ethernet. Dans une situation classique, aucun réglage n'est nécessaire et la connexion est immédiatement utilisable. Seulement, dans les locaux de l'INSA, il est nécessaire de configurer le système pour qu'il utilise le proxy :

```
| $ export http_proxy='http://cachemad.insa-rouen.fr:3128'
```

Pour permettre à ces changements d'être effectifs à chaque redémarrage, on ajoute la ligne précédente dans le fichier `~/.bashrc`. Enfin, il faut régler le proxy pour l'installateur de paquets :

```
| $ echo "Acquire::http::Proxy \"http://cachemad.insa-rouen.fr:3128\";" > /etc/apt/apt.conf
```



Mettre à jour les paquets

Même si le système installé est récent, il est toujours important de mettre à jour les programmes et composants utilisés. Lorsque l'on est connecté à Internet, cela s'effectue très simplement à l'aide de ces deux commandes :

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

Changement de l'heure

La Raspberry perd l'heure à chaque redémarrage. En tant normal, une fois connectée à Internet, la Raspberry se met à l'heure grâce au protocole NTP. Seulement, le proxy de l'INSA bloque le port utilisé par NTP. Nous n'avons donc pas d'autre choix que de régler l'heure manuellement à chaque démarrage. Pour régler l'heure via le terminal :

```
$ sudo date -s "Apr 9 14:28"
```

Prendre des captures d'écran

Nous avons aussi souhaité pouvoir prendre des capture d'écran. Pour cela nous avons trouvé une solution simple : installer le paquet `scrot`.

```
$ sudo apt-get install scrot
```

Dès lors, pour prendre une capture d'écran depuis le terminal il suffit d'utiliser de taper :

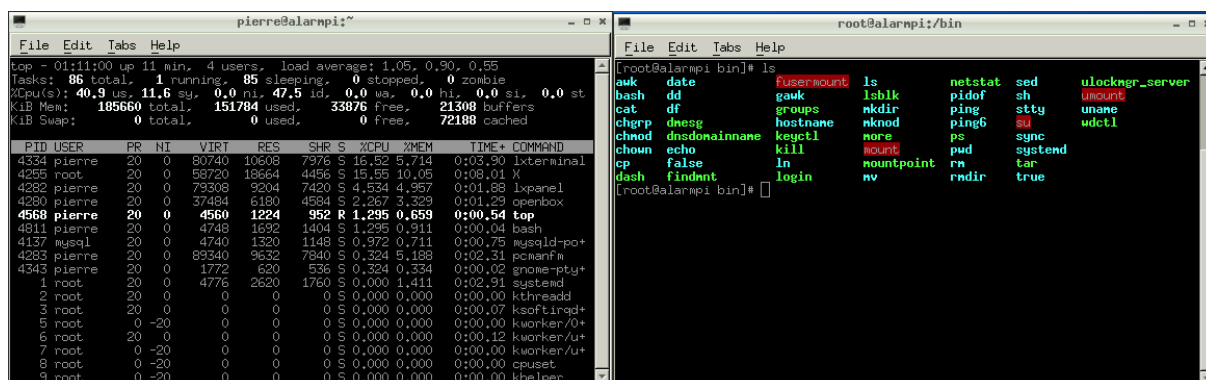
```
$ scrot /chemin/vers/le/fichier.png
```

Pour rendre cette procédure plus pratique, nous avons décidé d'assigner la commande ci-dessus à une touche du clavier (par exemple, la touche Impr. Écran). Il faut éditer le fichier `/home/pi/.config/openbox/lxde_rc.xml` :

```
$ nano /home/pi/.config/openbox/lxde_rc.xml
```

Puis pour que les changements soient immédiatement pris en compte :

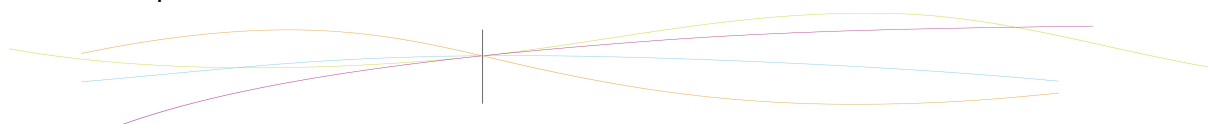
```
$ openbox --reconfigure
```



3.1.3. Matériel du projet

Pour le réveil :

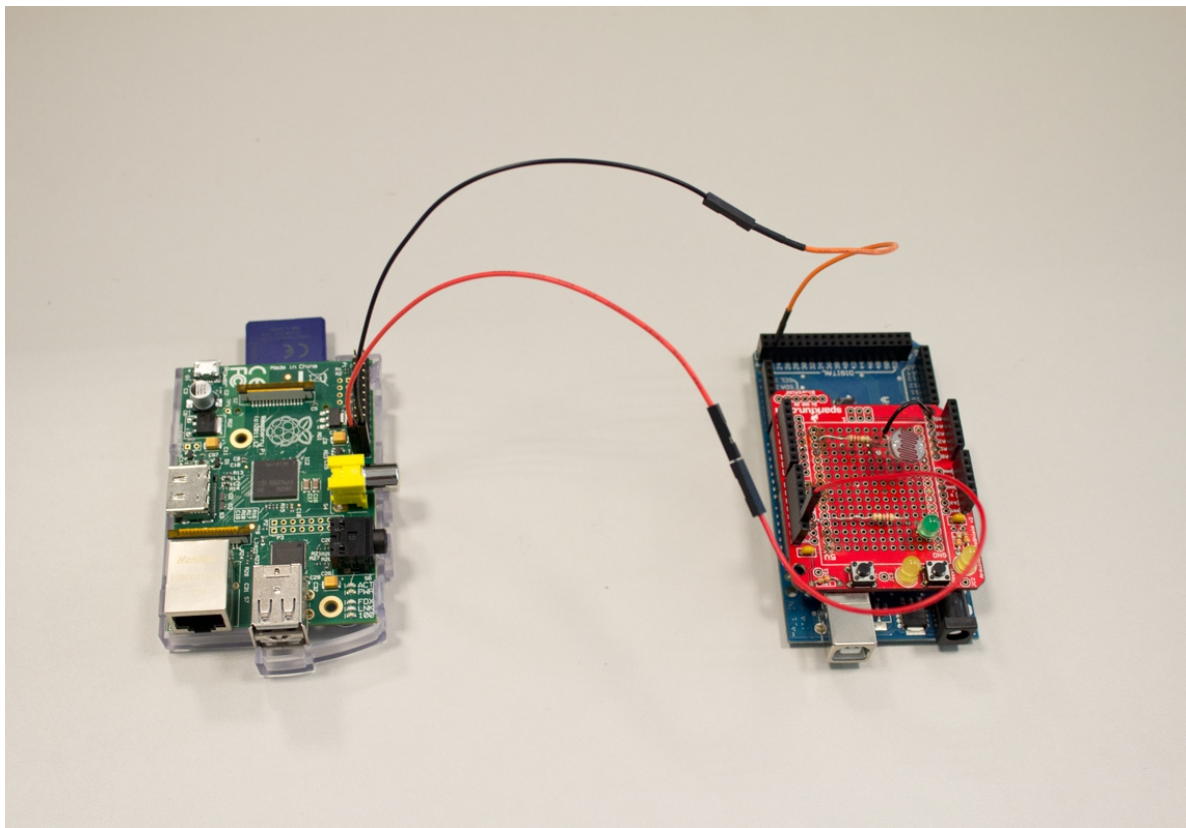
- Raspberry Pi
- Adaptateur 5V 1A



- Clavier et souris
- Câble ethernet pour la connexion Internet
- Adaptateur HDMI vers DVI
- Écran avec entrée DVI

Pour le capteur de lumière :

- Arduino Mega ATmega1280
- Une LED verte
- Une photorésistance



3.1.4. Préparation de la Raspberry Pi pour le projet réveil intelligent

Installation du serveur web [\[2\]](#)

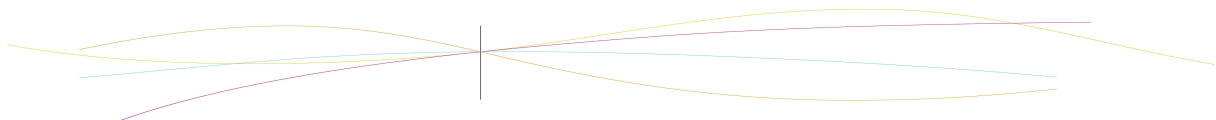
Dans le but d'héberger un site web pour notre projet de réveil intelligent nous avons installé un serveur Apache complet :

```
| $ sudo apt-get install apache2 php5 php5-mysql
```

Les services sont alors automatiquement créés et lancés au démarrage. Il suffit ensuite de placer les fichiers du site web dans le dossier : `/var/www/`

Montage des clés usb

Au cours de notre projet, il nous a été nécessaire de transférer des fichiers depuis une clé USB vers la Raspberry Pi. Cependant, cette opération nécessite de passer par la commande `mount` :



```
| $ sudo mount -a
```

Le contenu de la clé est alors rendu accessible dans `/media/[nom de la clé]`. Pour démonter la clé afin de la retirer en toute sécurité, il faudra taper :

```
| $ sudo umount -a
```

Il est possible de rendre le montage des clé USB automatique de la manière suivante :

```
| $ sudo bash
| $ apt-get install usbmount
| $ exit
```

3.2. Le projet « réveil intelligent »

Une fois les premières étapes d'installation et de configuration complétées, nous nous sommes lancés dans la réalisation d'un projet d'application de notre choix qui permet d'utiliser un bon nombre des fonctionnalités offertes par la Raspberry Pi. Il s'agit d'un « réveil intelligent ».

3.2.1. Fonctionnalités principales

Notre réveil est tout d'abord un réveil classique :

- Il est configurable à loisir (horaire au choix en fonction du jour de la semaine)
- Il joue un morceau de musique au choix

Mais il dispose également de plusieurs fonctionnalités innovantes et utiles :

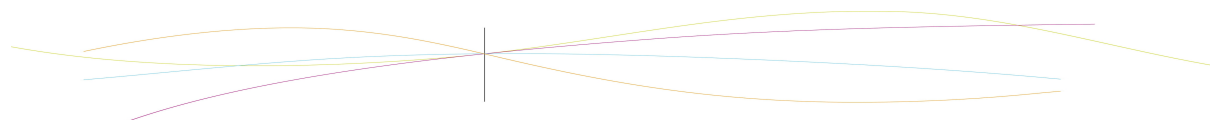
- Le réveil s'arrête automatiquement lorsque l'utilisateur est levé (grâce à une détection du changement de luminosité dans la chambre)
- Il affiche à l'écran des informations personnalisées et à jour :
 - L'heure et la date
 - Le temps restant avant le départ des trois prochains métros (à Rouen)
 - La météo
 - Les cinq gros titres de l'actualité mondiale (via Reddit)
- Le réveil annonce l'heure, la date et un message au choix à voix haute (grâce à la synthèse vocale)

Grâce à son interface web (détaillée plus loin), l'utilisateur n'a jamais besoin d'intervenir directement sur la Raspberry Pi : le matériel nécessaire au fonctionnement est donc minimal, et l'utilisation très simple.

3.2.2. Technologies utilisées

Le réveil intelligent fait appel à de nombreuses technologies, c'est pourquoi il était particulièrement adapté pour démontrer les larges capacités de la Raspberry Pi.

Tout d'abord, le programme principal est écrit script **bash**. Il se charge de jouer la musique, de communiquer avec la carte Arduino via les **GPIO** et, lorsque le réveil s'arrête, il déclenche l'affichage des informations matinales.



Ce script est lancé à la bonne heure par le programme **cron**, intégré à tous les systèmes Linux. Une fois correctement configuré (via la commande *crontab*), cron lance le ou les programmes demandés à l'heure voulue.

Cette configuration, justement, est réalisée via une **application web** hébergée sur la Raspberry Pi. En effet, nous avons installé et configuré un **serveur Apache**. L'utilisation des technologies web a l'avantage majeur d'offrir une interface homme-machine universelle, via un simple navigateur. L'interface de configuration, tout comme la récupération et l'affichage des informations matinales, sont programmés en **PHP**, **HTML5**, **CSS3** (Compass) et **Javascript** (jQuery).

Une attention particulière a été apportée à programmer la récupération, le traitement et l'affichage des informations sous la forme de modules indépendants. Ainsi, il sera facile dans le futur d'ajouter de nouveaux modules en se basant sur le modèle existant.

Enfin, la carte Arduino est programmée en une variante du langage **C**. Elle se charge principalement de détecter le niveau de luminosité de la pièce (son rôle est détaillé dans la partie 3.3).

3.2.3. *Mode d'emploi*

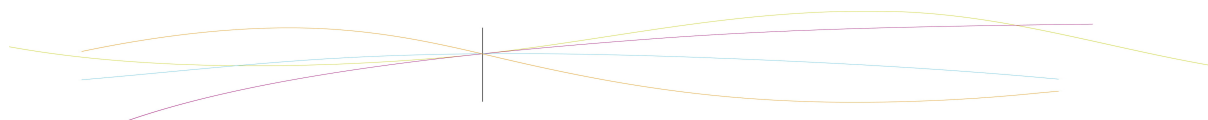
Matériel requis :

- Une Raspberry Pi
- Une carte SD contenant le système et les programmes
- Un écran et des enceintes
- Une carte Arduino munie d'un *shield* et du montage
- Une connexion à Internet
- Divers câbles pour connecter le tout

Il est à noter qu'il n'est pas nécessaire de disposer d'un clavier et d'une souris. En effet, toutes les interactions ont lieu via l'interface web.

Configuration :

Afin de choisir l'heure du réveil et de personnaliser les informations affichées le matin, un serveur web est lancé sur la Raspberry Pi. Une fois cette dernière connectée à Internet, on peut y accéder depuis n'importe quel appareil : un ordinateur, une tablette, un téléphone, etc. L'interface est optimisée pour mobile.



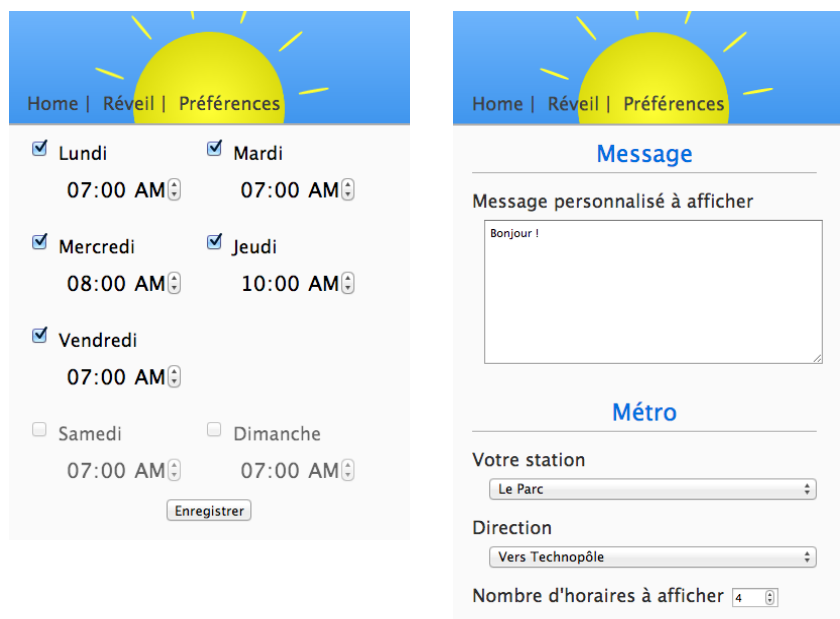


Figure 3: Vues de l'interface de configuration (mobile)

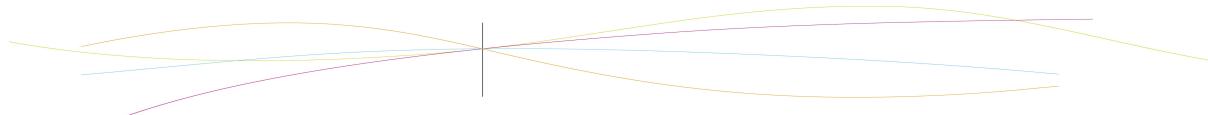
Les paramètres configurables sont :

- L'heure de réveil : chaque jour de la semaine peut avoir une heure différente, être activé ou désactivé
- Le message personnalisé à afficher et à lire « à voix haute » le matin (par synthèse vocale)
- Le métro : la station de départ et la direction souhaitée (le système est actuellement limité au métro de Rouen), ainsi que le nombre d'horaires à afficher
- La météo : la ville choisie parmi les villes disponibles (Rouen, Paris et Lyon)
- Les news : le thème des actualités à afficher depuis *Reddit* (par défaut, on affiche les nouvelles du monde)

Le choix de la musique s'effectue en plaçant le morceau à jouer dans un dossier spécifique sur la Raspberry Pi. Ceci peut être réalisé facilement via une connexion **SSH**.

Utilisation :

Une fois l'étape de configuration effectuée, le programme se charge de tout : la musique se lancera à l'heure demandée, et s'arrêtera automatiquement lorsque l'utilisateur allume la lumière. Les informations sont ensuite affichées automatiquement à l'écran, et se mettent à jour en temps réel.



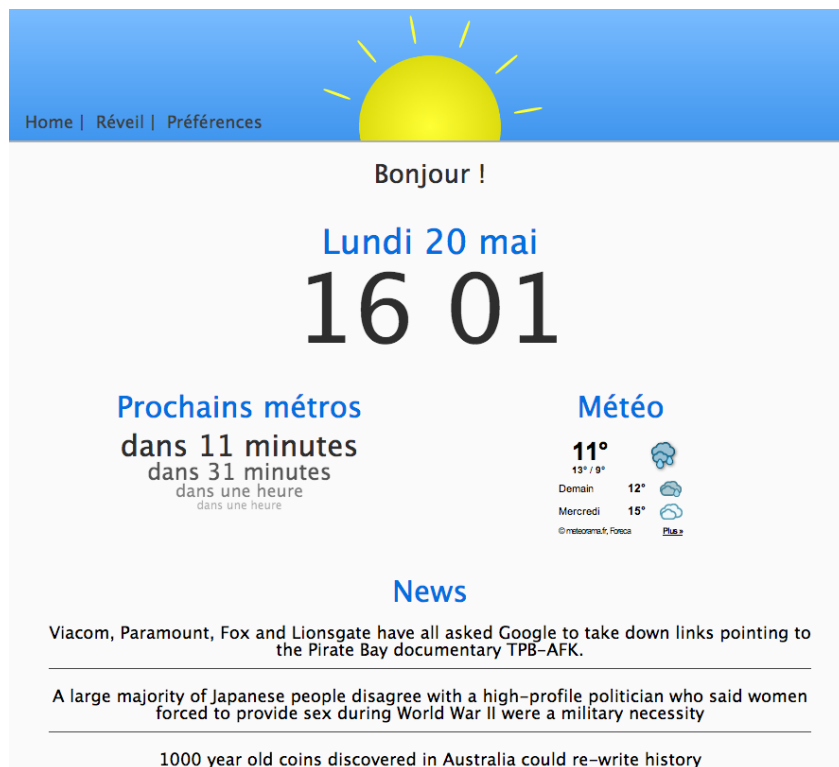
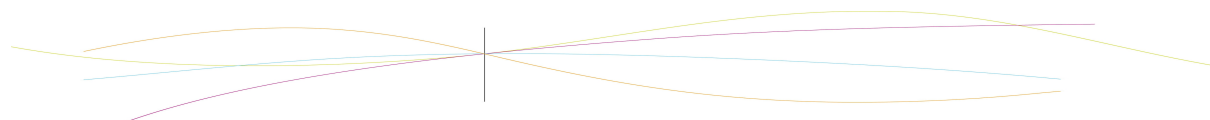


Figure 4: Vue de l'affichage des informations matinales

Note : Pour que l'arrêt automatique du réveil soit efficace, il est important que la pièce soit sombre pendant la nuit. Ainsi, lorsque l'utilisateur allume la lumière en se levant, la photorésistance pourra détecter un réel changement de luminosité.

3.3. Interfaçage avec la carte Arduino

Un autre aspect de notre projet a consisté à développer un capteur de lumière. Lorsque la sonnerie du réveil est déclenchée, elle continue jusqu'à ce qu'un seuil suffisant de lumière soit détectée. Typiquement, cela se produit quand l'utilisateur allume la lumière de sa chambre. Ainsi pour arrêter le réveil, l'utilisateur est obligé de se lever et d'allumer la lumière : cela nous a semblé être une bonne solution pour garantir que l'utilisateur ne se rendorme pas.



3.3.1. Réalisation d'un capteur de lumière avec une carte Arduino

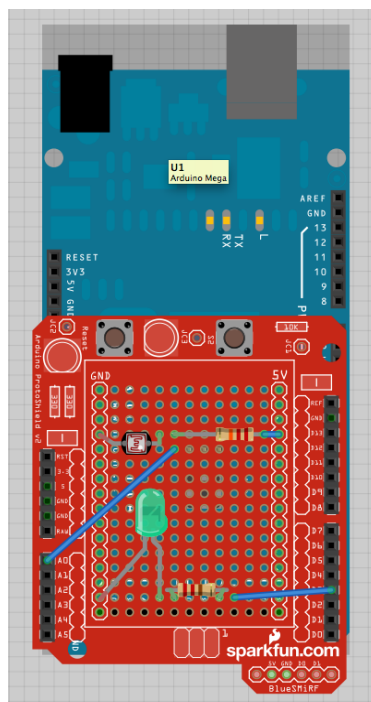


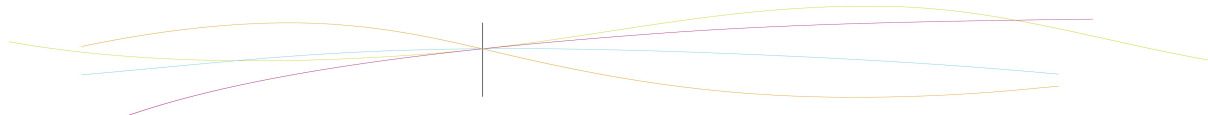
Figure 5: Capteur de lumière : schéma du montage électronique

Le montage de la carte Arduino est très simple : nous avons placé une LDR (photorésistance), la résistance de laquelle varie en fonction de l'intensité lumineuse reçue. Cela se traduit donc par une variation de la tension que nous recevons par l'entrée analogique A5. Après quelques essais, nous avons remarqué qu'une valeur supérieure à 200 (pour un maximum de 255) est un seuil satisfaisant.

Le programme chargé sur la Arduino [voir annexes] se contente donc de lire la valeur du port A5 à chaque seconde, puis décide si celle-ci correspond à une présence ou une absence de lumière. En conséquent, on allume ou on éteint la LED et on envoie l'information binaire correspondante à la Raspberry Pi par la sortie D22.

3.3.2. Communication entre la carte Arduino et la Raspberry Pi

Pour établir la communication entre les cartes Raspberry et Arduino, nous avons utilisé les **GPIO** de la Raspberry Pi [3]. En effet la Raspberry Pi est munie de 26 pin d'entrée / sortie. Mais cependant seulement 8 pin sont disponibles pour une utilisation classique (les GPIO 7,11,12,13,15,16,18 et 22).



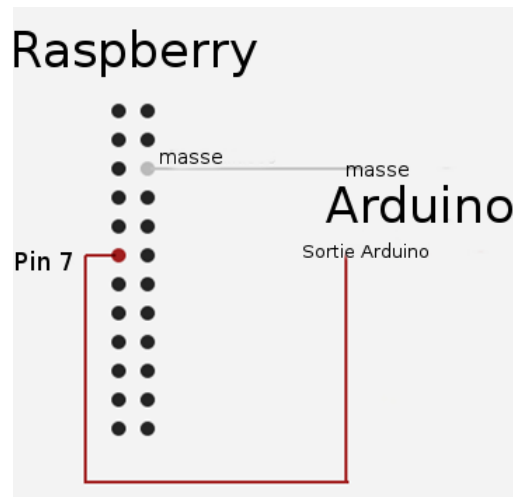


Figure 6: Schéma du montage de la communication Arduino - Raspberry

Nous avons décidé de relier la sortie digitale D22 de la Arduino au GPIO 7 de la Raspberry et la masse de la Raspberry (3ème PIN en partant du haut à droite) à celle de la Arduino. Une fois ces connexions effectuées, on est en mesure de lire une valeur digitale directement dans le terminal de la Raspberry en effectuant les étapes suivantes :

1. Ouvrir le GPIO 7 en entrée :

```
| echo "7" > /sys/class/gpio/export
| echo "in" > /sys/class/gpio/gpio7/direction
```

2. Lire la valeur digitale (affiche 0 ou 1) :

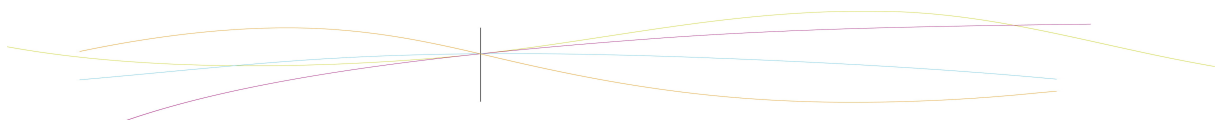
```
| cat /sys/class/gpio/gpio7/value
```

3. Fermer le GPIO lorsque l'on a terminé :

```
| echo "7" > /sys/class/gpio/unexport
```

Avec ce dispositif, nous avons réussi à interfacier deux appareils distinct et à établir une communication binaire entre eux. Bien que notre exemple soit extrêmement basique, c'est une démonstration des capacités de la Raspberry Pi dont on peut utiliser les GPIO pour communiquer avec d'autres appareils.

Par ailleurs, on peut imaginer plusieurs situations possibles d'utilisation des GPIO en relation avec une carte Arduino. Par exemple une situation où la Raspberry contrôle une ou plusieurs cartes Arduino ou inversement.



4. CONCLUSIONS ET PERSPECTIVES

4.1. Conclusion générale

Ce projet fut très enrichissant, tout d'abord par la variété de domaines qu'il met en jeu. En effet, de l'électronique aux langages de programmation de haut niveau, nous avons abordé, en concentré, tous les niveaux d'abstraction sur lesquels l'informatique repose. Le matériel utilisé est puissant, et offre de nombreuses opportunités. Les consignes données par M. Henriet nous laissant le champ libre, il était agréable de pouvoir imaginer n'importe quel projet d'application avec de tels moyens. De plus, nous sommes satisfaits de conclure ce projet en ayant produit un objet pouvant avoir une réelle utilité dans la vie de tous les jours.

En poursuivant le projet, les fonctionnalités suivantes pourraient être ajoutées au produit pour l'améliorer :

- Rendre l'arrêt automatique du réveil possible par une réelle détection de mouvement grâce au *shield* Arduino CMUcam4, utilisé par d'autres groupes de projet P6
- Travailler sur la compacité de l'objet, en concevant par exemple un boîtier rassemblant toutes les parties physiques
- Ajouter des fonctionnalités au réveil :
 - Rendre la configuration de la musique plus facile en l'intégrant à l'interface web déjà utilisée pour le reste des paramètres
 - Rendre le choix de la musique intelligent, en se basant par exemple sur la météo, l'heure du réveil ou d'autres critères

4.2. Conclusions des membres du groupe

Hicham Belhamer

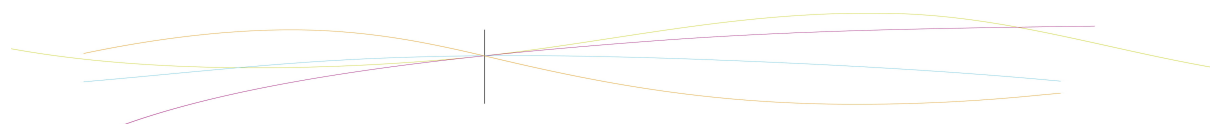
Pour ma part, j'ai trouvé le projet de mise en œuvre de la carte Raspberry Pi très intéressant car au-delà de l'aspect informatique du projet, nous avons aussi pu nous intéresser à l'aspect électronique, remémorant ainsi quelques cours de 1^{ère} année. De plus, étant le moins avancé des trois membres du groupe dans le domaine de l'informatique, ce travail m'a permis d'enrichir mes connaissances et de développer un esprit critique par rapport à l'utilisation d'un appareil et du champ d'utilisations que l'on peut en faire.

Pierre Besson

Ce projet a été très instructif. L'informatique embarquée est un domaine en plein essor et j'ai vraiment apprécié pouvoir expérimenter dans ce domaine et réaliser un tel projet. De plus, la liberté de choix du sujet a été un grand avantage pour nous forcer à être créatif et impliqué dans le projet. Bien que notre groupe était réduit à trois personnes, le travail d'équipe a été essentiel pour mener à bien le projet.

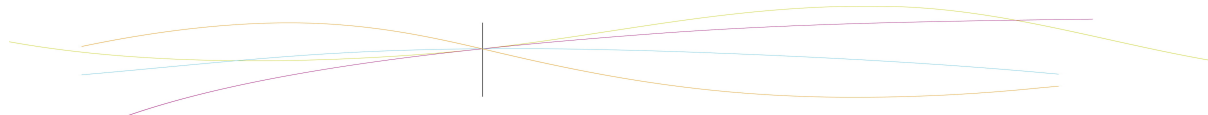
Merlin Nimier-David

Ce projet aura été pour moi l'occasion de découvrir de nouvelles technologies, et de les appliquer aussitôt à un projet concret. La plateforme utilisée étant extrêmement flexible, j'ai pris grand plaisir à imaginer puis mettre en œuvre des solutions techniques, sans limites imposées par le matériel. J'ai apprécié l'aspect « expérimental » de notre démarche.



5. BIBLIOGRAPHIE

- **[1]** Téléchargement des distributions Linux recommandées pour la Raspberry Pi
<http://www.raspberrypi.org/downloads>
[valide à la date du 29/05/13]
- **[2]** « Creating a LAMP server with the Raspberry Pi »
<http://www.penguintutor.com/linux/raspberrypi-webserver>
[valide à la date du 29/05/13]
- **[3]** « RPi Low-level peripherals »
http://elinux.org/RPi_Low-level_peripherals
[valide à la date du 29/05/13]



6. ANNEXES

6.1. Fiche technique de la Raspberry Pi

La Raspberry Pi est un ordinateur à très bas coût (30€) conçu par la fondation Raspberry Pi.

Cet ordinateur est équipé d'un processeur ARM Broadcom BCM2835 700 MHz, d'une puce graphique Broadcom VideoCore IV pour le décodage de vidéo et la 3D et de 256 ou 512 Mo de RAM.

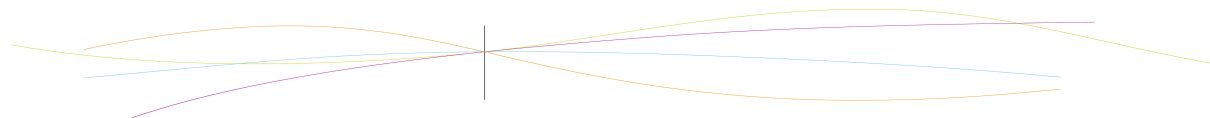
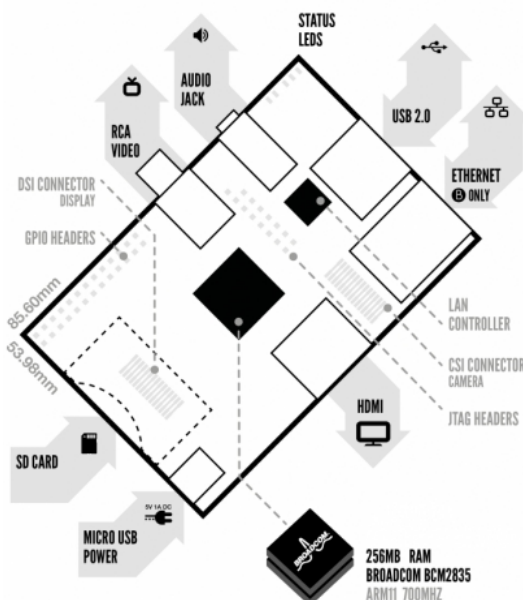
Connectique :

- 2 port USB 2.0 (extensible avec un hub usb)
- 1 port Ethernet
- 1 lecteur de carte SD (sert de disque dur principal)
- 1 sortie vidéo composite
- 1 sortie HDMI (vidéo et audio)
- 1 sortie jack 3.5mm

Caractéristiques :

- Alimentation minimum : 5V, 700 mA (3.5 W)
Il vaut mieux prendre une alimentation de 1A pour alimenter aussi le clavier et la souris
- Dimensions : 85.0 x 56.0 mm
- 26-pin analogiques (2x13) 2.54 mm

Raspberry Pi
Model A B



6.2. Listings des programmes réalisés

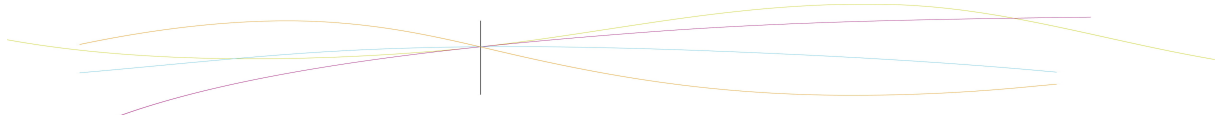
```
// Numéro des pins utilisés
int led = 13;
int ldr = A5;
int raspberry = 22;

void setup()
{
    pinMode(led, OUTPUT);
    pinMode(raspberry, OUTPUT);
    pinMode(ldr, INPUT);
}

void loop()
{
    // On attend une seconde avant de continuer
    delay(1000);

    int luminosite = analogRead(ldr);
    if(luminosite <200)
    {
        digitalWrite(raspberry,HIGH);
        digitalWrite(led, HIGH);
    }
    else
    {
        digitalWrite(raspberry,LOW);
        digitalWrite(led, LOW);
    }
}
```

Listing 1: Programme Arduino (détection de la luminosité)



```
#!/bin/bash

# À lancer en root pour avoir accès aux GPIO

# Initialisation
echo "7" > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio7/direction
gpio=$(cat /sys/class/gpio/gpio7/value)

# Choix de la musique à jouer par ordre aléatoire
ls musique/* |sort -R |tail -$N -n 1 |while read file; do
    alsaplayer $file&
done

# On annonce l'heure à voix haute (synthèse vocale)
heure=$(date +%H)
minute=$(date +%M)
echo "C'est l'heure de se réveiller !, il est $heure heures et $minute minutes"
espeak -v fr -s 200 "C'est l'heure de se réveiller !, il est $heure heures et $minute minutes"

# On continue tant que la lumière n'est pas allumée
while [ $gpio -eq 0 ]:
    do
        gpio=$(cat /sys/class/gpio/gpio7/value)
        sleep 10
    done

# Nettoyage
echo "7" > /sys/class/gpio/unexport
killall alsaplayer

# Affichage des informations via l'interface web
kiosk-browser http://127.0.0.1/P6-raspy
```

Listing 2: Script bash (réveil)

