

# RADEXPO

## Mesure de l'intérêt du public pour une œuvre exposée

L'idée de ce projet est né au cours de la dernière exposition de peinture à laquelle je participais et où je me demandais comment mesurer objectivement l'intérêt du public pour un tableau ou une œuvre par rapport aux autres tableaux ou aux autres œuvres exposées.

J'ai retenu plusieurs impératifs :

- mesure la plus fidèle possible de la durée de présence de visiteurs,
- pouvoir consulter les mesures effectuées en temps réel sur un écran,
- enregistrer les mesures dans un fichier pour pouvoir en faire une étude statistique ultérieure,
- horodater ces mesures.

Maîtrisant assez bien les microcontrôleurs Arduino, c'est donc leur usage que j'ai privilégié. J'ai essayé diverses approches de mesures avec le souci de ne pas utiliser de capteurs onéreux. Entre les détecteurs laser que je ne pouvais pas mettre en œuvre dans un lieu fréquenté par le public et les capteurs à ultrasons, j'ai finalement opté pour un système de détection infrarouge. Celui-ci donne des résultats constants et peu soumis aux mesures parasites comme ce fut le cas avec des mesures par ultrasons à une distance de plusieurs mètres.

Un autre impératif pour ce projet est la miniaturisation afin de pouvoir dissimuler facilement l'appareillage derrière un tableau par exemple.

Afin de répondre à cet objectif, j'ai retenu la solution de faire coopérer deux microcontrôleurs arduino de petite taille : des Arduino Nano. En effet, la taille du programme engendré par les impératifs ci-dessus est supérieur à la capacité de mémoire d'une seule carte Nano. Le choix aurait pu se porter aussi sur d'autres cartes miniatures Arduino (Micro ou Mini) mais je n'en disposait pas au moment de la mise en oeuvre de ce projet. Par ailleurs, les ports USB des Nano sont bien pratiques pour mettre à jour les programmes lors du développement et d'autre part la sortie simultanée de tensions de 5v et 3,3v est nécessaire dans ce montage. Ce dont ne dispose pas une carte Arduino Mini.

Pour dire vrai, je pensai au début pouvoir faire tenir le programme sur une seule carte Nano. Utopie !

### Les fonctions des cartes :

La première carte affectue la détection de présence de visiteurs et affiche les comptages sur un petit écran miniature. Elle demande à la seconde carte la date, l'heure et le n° d'appareil afin de les afficher et permettre à l'utilisateur de modifier la date et l'heure. Elle affiche aussi le numéro d'appareil, mais celui-ci, fourni par le fichier paramètres n'est pas modifiable.

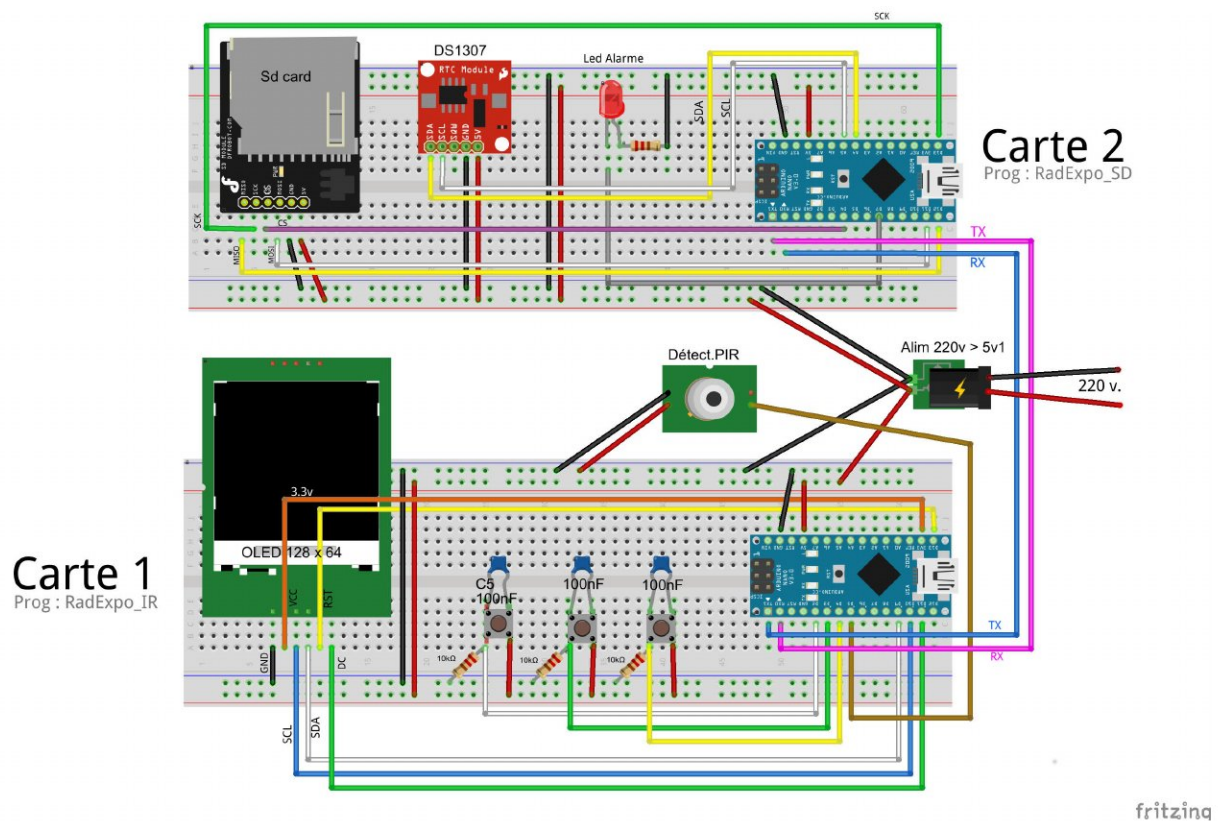
Elle est donc équipée d'un détecteur infrarouge, d'un écran Oled 64x128 et de 3 boutons pour le réglage de la date et de l'heure.

La seconde carte reçoit les données de mesures de la première carte et les enregistre dans un fichier. Elle commence à la mise sous tension par lire le contenu du fichier pour effectuer un comptage récapitulatif des données précédemment enregistrée et les envoyer à la première carte qui reprendra alors ses comptages là où elle s'est éventuellement arrêtée. Elle accède aussi à un fichier paramètre qui contient le numéro d'appareil. En l'absence de ce fichier sur la carte SD, l'appareil se voit automatiquement attribuer le n° 01.

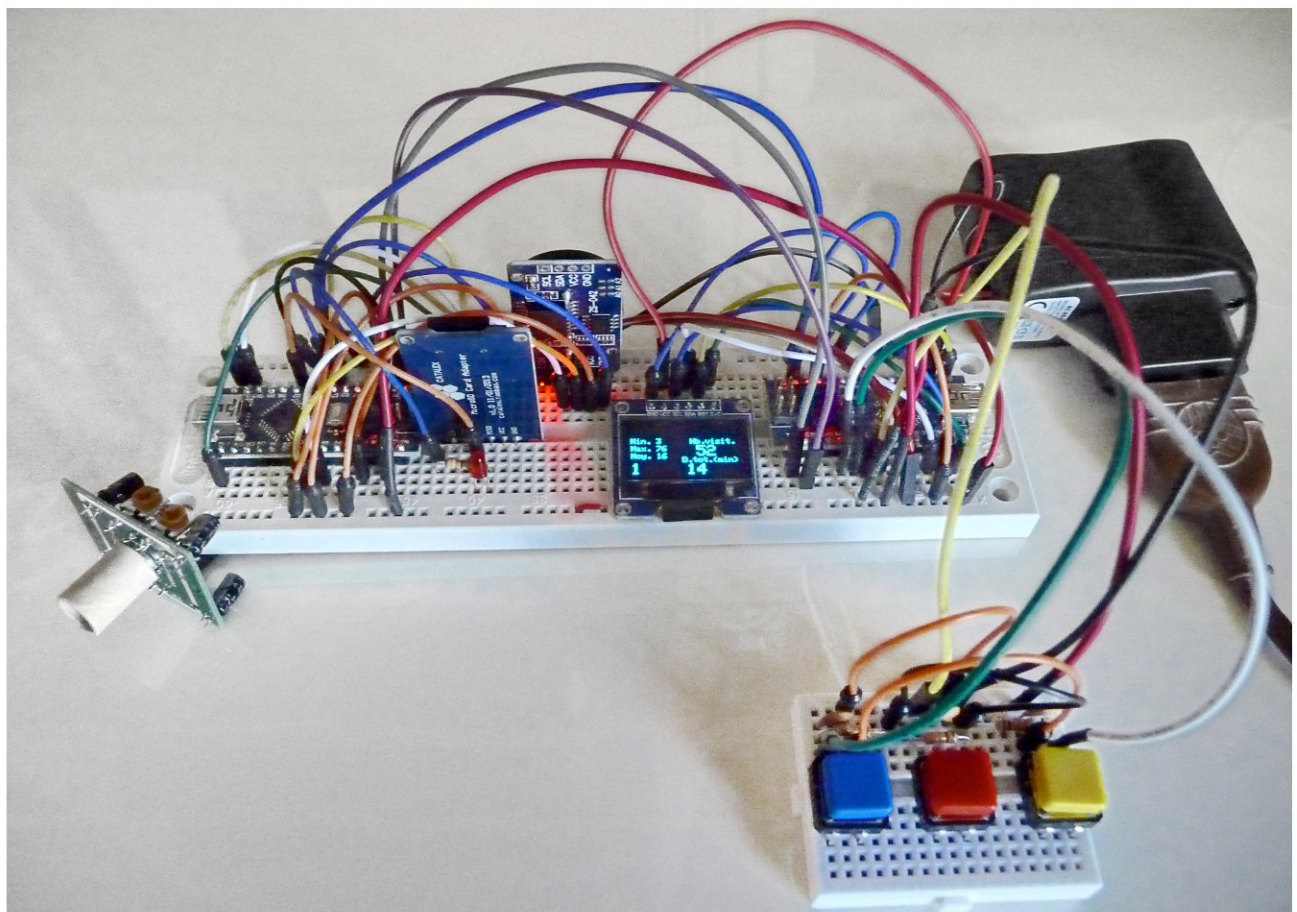
Cette seconde carte est donc équipée d'un module lecteur-enregistreur de carte SD et d'une horloge temps réel alimentée par batterie. Elle dispose aussi d'une led d'alarme pour signaler une défection de la carte SD.

Les deux cartes communiquent par leur port série.

**Le modèle théorique :**



**La réalité du prototype : (c'est un peu plus confus....mais c'est un proto de développement)**



## Méthodologie

Pour détecter la présence de visiteurs, l'utilisation d'un détecteur infrarouge passif (PIR) est la solution la moins onéreuse. Elle est basée sur une détection pyroélectrique qui détecte un changement de température (la présence d'une ou plusieurs personnes) mais ne permet pas une détection continue. Un capteur à détection thermoélectrique à distance aurait été plus judicieux, mais les coûts de ces capteurs sont beaucoup plus élevés.

J'ai donc pris un PIR dont j'ai remplacé la lentille de Fresnel par un petit tube afin de réduire le champ angulaire de détection.

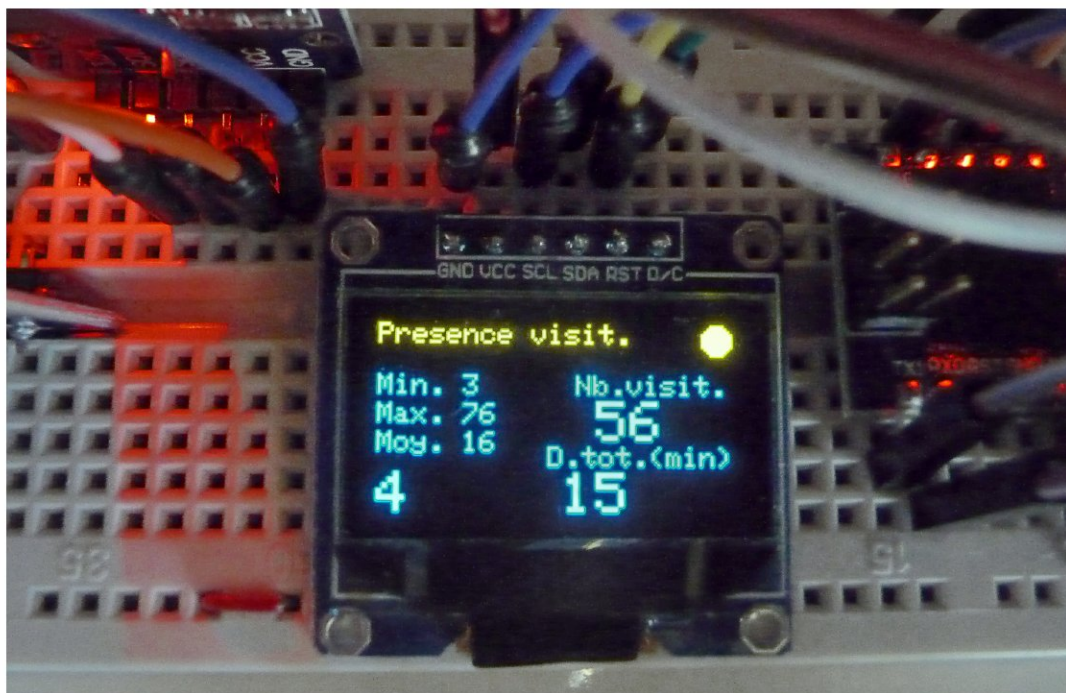
Ce détecteur envoie une tension s'il détecte un mouvement. Toutefois, il repasse à l'état bas (0v) de temps en temps même en présence d'un mouvement continu. Pour éviter d'interpréter ces passages à l'état bas de courte durée comme une absence de visiteur, on ignore ces phases à 0v si elles durent moins de 5 secondes.

On obtient en faisant ce lissage temporel une détection de présence qui n'est pas parfaite, mais qui donnera des résultats statistiques fiables par la fidélité dans ses mesures. On pourra ainsi déterminer l'intérêt que le public a porté à différentes œuvres équipées de ce système en étudiant les résultats comparatifs de celles-ci.

Les résultats des comptages affichés sur l'écran indiquent en temps réel :

- le nombre total de passages de visiteurs
- le temps total cumulé des visites depuis le début de l'utilisation du système (en minutes)
- la durée de visite la plus courte (en secondes)
- la durée de visite la plus longue (en secondes)
- la durée moyenne des visites (en secondes)
- le temps de la visite en cours d'enregistrement (en secondes) en bas de l'écran
- un message qu'une visite est en cours d'enregistrement
- un indicateur de l'activation du détecteur (spot jaune sur l'écran)
- le numéro d'appareil.

Pour éviter de brûler l'écran par un affichage permanent, l'affichage des mesures se fait dès le début d'une visite et ceci pendant une minute après la fin de celle-ci. Ensuite l'écran s'éteint automatiquement.



## Les échanges entre cartes

Comme dit précédemment, les deux cartes Arduino communiquent par leur port série à l'aide de messages répondant à un protocole bien défini. Celui-ci comprend un identifiant de 7 caractères en tête de message suivi des données spécifiques à chaque message et terminé par le signe « \$ » de fin de message.

A la mise sous tension, un message d'initialisation des compteurs (identifiant = « inittot ») est envoyé par la carte 1 vers la carte 2. Il est structuré de la façon suivante :

inittot/nb visit./durée totale/durée mini/durée maxi/durée moy/n° d'appareil\$

A la fin de chaque visite, un message (identifiant = « radexpo ») est envoyé par la carte 1 vers la carte 2. Ce message est structuré de la façon suivante :

radexpo/n° visiteur/durée de la visite en sec\$

A l'appui sur le bouton de demande d'affichage de la date et heure, le message : « demheur\$ » est envoyé par la carte 2. En réponse à celui-ci, la carte 1 renvoie la date et l'heure sous la forme :

affheur/hhmmss/JJMMAAAA/j\$

Si l'utilisateur met à jour la date ou l'heure ou le jour de la semaine, la carte 2 renvoie les données modifiées par le message :

regheur/hhmmss/JJMMAAAA/j\$

## La mise à jour de l'heure, de la date ou du jour de la semaine

Le bouton jaune demande l'affichage de la date/heure. Un second appui renvoie à l'affichage des mesures.

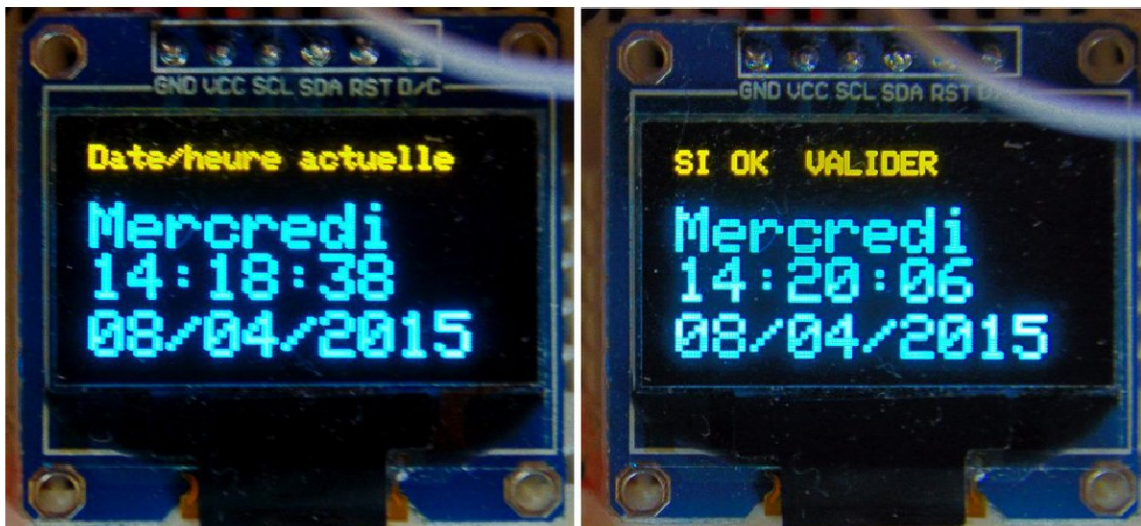
Lorsque la date/heure. est affichée, le bouton bleu déclenche le processus de modification.

Le bouton jaune sert alors à changer de champ de donnée (jour de la semaine > heures > minutes > secondes > jour > mois > année > validation).

Le bouton rouge sert à incrémenter la valeur du champ en cours de modification.

Le bouton bleu, lorsque la validation est suggérée, sert à valider les modifications.

Un message d'acquiescement est affiché lors de la validation des modifs ou si aucune modif n'a été détectée.



## Le fichier des données

A la réception d'un message «radexpo», la carte 2 complète des données de mesure de celle du numéro d'appareil ainsi que de la date et heure de réception et enregistre le tout dans le fichier de la carte SD sous forme d'un renregistrement fixe de 44 caractères mais au format CSV afin d'en faciliter l'importation dans un tableur. Il se présente sous la forme suivante :

radexpo;n°appareil;n°visiteur;durée(sec);AAAA;MM;JJ;hh:mm:ss;jour de la semaine (Dim=1)

### Exemple des données lues dans un tableur

radexpo	1	87	3	2015	4	8	13	53	53	4
radexpo	1	88	3	2015	4	8	13	54	48	4
radexpo	1	89	3	2015	4	8	13	55	43	4
radexpo	1	90	3	2015	4	8	13	55	56	4
radexpo	1	91	6	2015	4	8	13	57	31	4
radexpo	1	92	7	2015	4	8	13	57	47	4
radexpo	1	93	5	2015	4	8	13	58	18	4
radexpo	1	94	4	2015	4	8	13	59	8	4
radexpo	1	95	3	2015	4	8	13	59	30	4
radexpo	1	96	3	2015	4	8	14	0	2	4
radexpo	1	97	26	2015	4	8	14	0	56	4
radexpo	1	98	6	2015	4	8	14	1	12	4
radexpo	1	99	5	2015	4	8	14	1	33	4

Il est possible ensuite de faire une étude statistique des visites par numéro d'appareil (colonne 2) en fonction de l'heure, du jour de la semaine (colonne 11) ou sur une période donnée.

## Notes techniques

L'écran Oled sur la carte 1 est alimenté par la sortie 3,3 volt de la carte tandis que tous les autres composants sont alimentés en 5v. D'où l'intérêt d'une carte Arduino Nano qui possède ces deux voltages.

Pour que le détecteur infrarouge soit opérationnel, il faut attendre 30 secondes avant d'effectuer des mesures. Voilà pourquoi à la mise sous tension, une période dite de « calibrage » est décomptée avant le fonctionnement effectif de l'appareil. Celle-ci est explicitement indiquée sur l'écran.

Les programmes respectifs de chacune des deux cartes dépassent les 20 Mo mais c'est au niveau de la mémoire dynamique que c'est le plus limité (2Mo). De ce fait, toute modif du code pourrait entraîner des dysfonctionnements d'exécution si on augmente la taille ou le nombre des variables, celles-ci ayant été calculées au plus juste.

Le développement du code a été fait avec l'IDE Arduino V1.6.0.