
FICHE REALISATION

INTERLOCK ARDUINO CONTROLE PAR TELEPHONE

Titre : Interlock Arduino contrôlé par téléphone

Type de réalisation : Ensemble électronique

Durée estimée : 1 jour

Date de réalisation : 15/03/2024 Sébastien Schaub

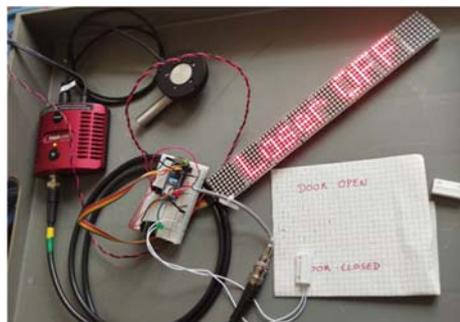
Fichiers associés : *FuncBLE.h*

FuncInit.h

Interlock_PIM.ino

Interlock_PIM.apk

Interlock_PIM.aia



Contact : Sébastien Schaub (sebastien.schaub@imev-mer.fr)

Objectif

L'objectif est de créer un système d'interlock pour une salle comportant un laser. Cela signifie qu'il s'agit d'un appareil permettant lorsque la porte du laboratoire est ouverte :

- De piloter la fermeture d'un shutter devant la source laser,
- D'assurer un affichage clair à l'extérieur du laboratoire
- De produire une alerte sonore

Mais afin de permettre une plus grande liberté et sécurité à l'expérimentateur, l'appareil est contrôlable depuis son portable via le Bluetooth.

Matériel

Description	Fournisseur	Reference	PU HT	Quantité	Total
Arduino nano ESP32	RS	268-6962	19	1	19
LED rouge 5mm	RS	228-5988	0.436	1	0.436
LED verte 5mm	RS	228-6004	0.198	1	0.198
Module buzzer actif GT1143	Gotronic	35239	2.5	1	2.5
Contact magnétique	Gotronic	26449	3.25	1	3.25
Connecteur BNC femelle	RS	144-0878	2.315	1	2.315
Plaque de connexion rapide	Gotronic	12312	4.5	1	4.5
Matrice LED (8x8)	Gotronic	35183	5.42	8	43.36
TOTAL					75.559

(* en cherchant, on peut trouver les matrices déjà soudées...)

Il faudra également :

- Une alimentation 5V (si utilisation de l'affichage LED au moins 1A et ne pas passer par l'alimentation de l'arduino qui pourra avoir du mal à alimenter)
- Un shutter pilotable en TTL (type SHB1T de Thorlabs).

Réalisation

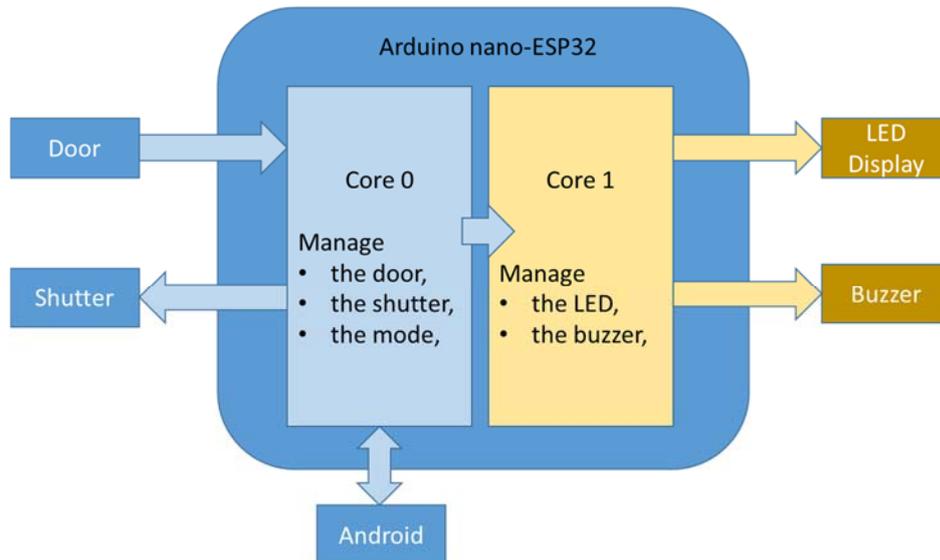
Modes de fonctionnement

L'interlock peut fonctionner avec 4 Modes :

- « SAFE » : dans ce cas, le shutter est TOUJOURS fermé.
- « WARNING » : le shutter est TOUJOURS ouvert. A utiliser avec précaution
- « AUTO » : le shutter ne s'ouvre que si la porte est fermée. A utiliser en général quand l'expérimentateur est dans la pièce
- « SINGLE » : le shutter reste ouvert pour UNE SEULE ouverture de porte. A la seconde ouverture, le système repasse en SAFE. Utile pour permettre à l'utilisateur de quitter la pièce en laissant l'expérience tourner. Cependant, sans réactivation du mode SINGLE, une personne rentrant dans la pièce stoppe l'expérience (mais sa sécurité est assurée).

Les modes sont contrôlés par le téléphone uniquement.

Schéma fonctionnel



L'arduino est codé via Arduino-IDE en s'appuyant sur plusieurs librairies :

- Parola⁽¹⁾ qui permet de gérer les affichages LED
- ArduinoBLE⁽²⁾ qui permet de gérer la communication Bluetooth BLE

¹ https://majicdesigns.github.io/MD_Parola/functions_func.html

² <https://www.arduino.cc/reference/en/libraries/arduinoBLE/>

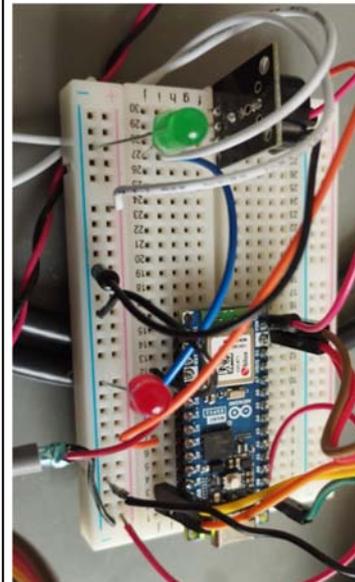
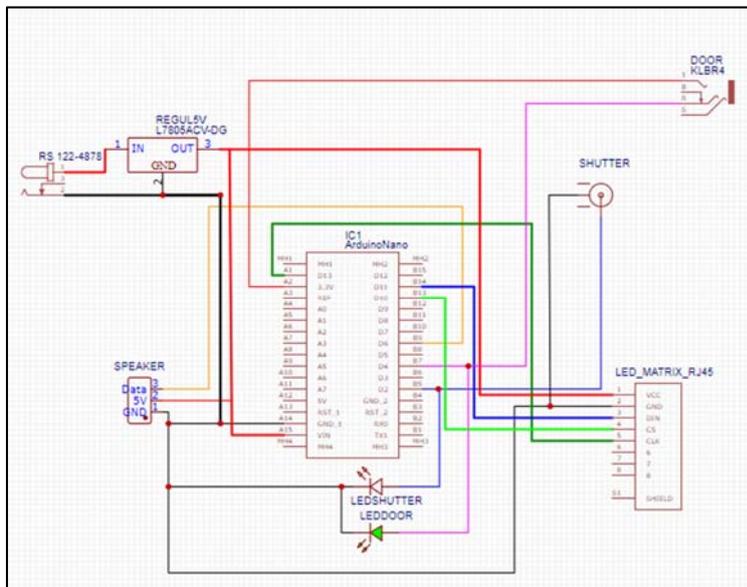
Pour les taches sur l'Arduino, on sépare les taches entre les 2 cœurs :

- Le Core 0 va gérer la sécurité, les actions qui ont besoin d'être faite rapidement (gestion du shutter en fonction de la porte et de l'android)
- Le Core 1 va gérer l'alarme et l'affichage qui n'ont pas besoin d'être aussi réactifs

Pour la communication Bluetooth, l'Arduino sera le périphérique auquel se connecte l'Android et qui fournira un Service « **ILServ** » avec 4 caractéristiques :

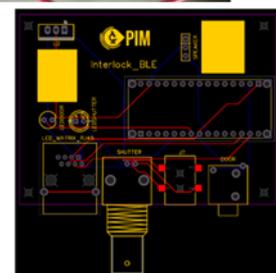
- **ModeChar** qui permet d'échanger la modification du Mode
- **DoorChar** qui permet d'informer le téléphone de l'état de la porte
- **ShutterChar** qui permet d'informer le téléphone de l'état du shutter
- **KeepChar** qui permet d'informer le téléphone de l'état du paramètre de maintien du Mode « Single »

Schéma électronique



(J'ai ajouter un régulateur de tension, mais c'est probablement pas nécessaire)

A terme, le système sera imprimé sur un PCB, ça n'est pas nécessaire non plus (mais c'est plus classe ^^)



Application Android

Ecrire sous MIT App Inventor⁽³⁾, un langage graphique assez facile d'accès.

Au lancement l'application va chercher continuellement un Bluetooth device « Interlock ». Une fois trouvé, il se connecte et permet la visualisation de l'état de l'Interlock.

L'interface ensuite permet de switcher entre les différents modes et de visualiser

- L'état de la porte
- L'état du shutter
- L'état du SingleKeep (quand il est vrai, on est en Mode « Single » et on peut ouvrir la porte sans couper le shutter)
- Les dernières actions effectuées (actuellement, seulement si l'android est connecté)

Mise en place

1. Installer sur Arduino-IDE le programme (bien penser à mettre dans un répertoire ayant le même nom que le .ino)



Interlock_PIM.ino



FuncInit.h



FuncBLE.h

2. Faire le montage électronique
3. Installer l'apk sur votre mobile/tablette Android



Interlock_PIM.apk

Pour ceux qui voudraient changer l'application (mettre le logo de votre labo par exemple...), ouvrir sur MIT App Inventor le fichier suivant :



Interlock_PIM.aia



³ <https://ai2.appinventor.mit.edu/>