

CFPT

Installation du menu

Partie Raspberry

Groupe Bugs Destroyer
Octobre 2021

Table des matières

| | |
|------------------------------------------------------------------------|---|
| I. Introduction | 1 |
| II. Matériel requis..... | 1 |
| III. Câblage..... | 2 |
| IV. Installation..... | 2 |
| 1 ^{er} étape : Activer l'interface I2C sur le raspberry | 2 |
| 2 ^{ème} étape : Installer les librairies..... | 2 |
| 3 ^{ème} étape : Configuration..... | 3 |
| 4 ^{ème} étape : lancement du programme..... | 3 |
| 5 ^{ème} étape (facultatif) : automatiser le lancement..... | 3 |
| V. Informations supplémentaires : | 4 |

I. Introduction

Ceci est un projet annexe des bornes d'arcades initialement prévu pour la cité des métiers 2021. Il a été réalisé par l'Ecole Entreprise 2021-2022. En reprenant le développement du menu de sélection de jeu installé sur les bornes d'arcades, nous avons prévu une solution de minuteur afin de ne plus pouvoir jouer indéfiniment sans laisser sa place à d'autres joueurs.

Afin de visualiser ce minuteur et voir combien de temps de jeu il nous reste, un Raspberry Pi contrôle un affichage 7 segments. Ce Raspberry communique avec le PC de la borne d'arcade via une connexion Ethernet (et grâce au protocole HTTP).

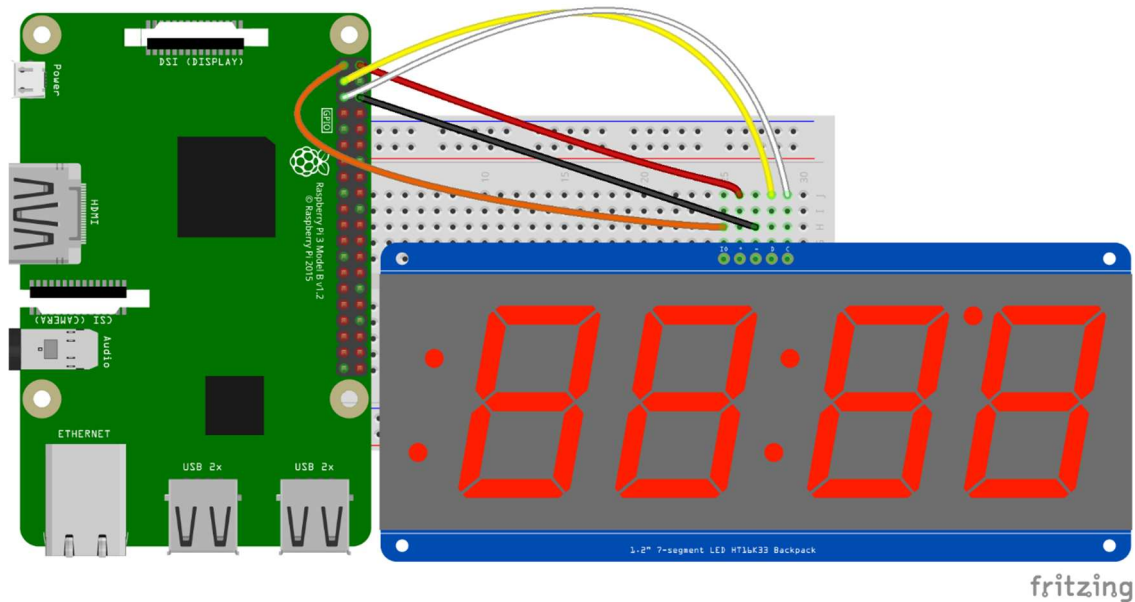
II. Matériel requis

- 1x Raspberry Pi (possédant des pins GPIO)
- 1x [Adafruit 1.2" 7-segment display avec HT16K33](#)
- 5 Câbles et une breadboard
- 1x connexion Ethernet (wifi ou filaire, selon le modèle de Raspberry)

III. Câblage

L'affichage dispose de 5 pins à câbler :

- 2 alimentations (5V et 3.3V) → fils rouge et orange
 - 1 cathode → fil noir
 - 2 transferts de donnée avec le bus I2C → fils jaune et blanc
- Ces deux derniers se connectent sur le GPIO2 et le GPIO3 du raspberry



Plus d'informations sur le montage :

- <https://learn.adafruit.com/adafruit-led-backpack/1-2-inch-7-segment-backpack-assembly>
- <https://learn.adafruit.com/adafruit-led-backpack/python-wiring-and-setup-d74df15e-c55c-487a-acce-a905497ef9db>

IV. Installation

Premièrement, installer un OS (de préférence Raspberry Pi OS) sur votre raspberry. Vous devez également posséder python 3.0 ou supérieur.

1^{er} étape : Activer l'interface I2C sur le raspberry

Dans votre terminal, entrez la commande `sudo raspi-config` puis naviguez dans le menu.

Interface Options → P5 I2C → Yes

2^{ème} étape : Installer les librairies

Entrez les commandes suivantes :

```
sudo apt install python3-pip
```

```
sudo pip3 install adafruit-circuitpython-ht16k33
```

```
sudo apt install python3-pil
```

3^{ème} étape : Configuration

Une fois que vous avez récupéré le code source, vous devrez vérifier et modifier certaines valeurs.

1. Adresse I2C :

Le plus important est l'adresse pour l'interface I2C, qui est renseignée au début du programme. Celle-ci peut être différente, il vous faut donc la vérifier à chaque nouvelle installation.

```
pi@borne1-rasp:~/Desktop/timer $ sudo i2cdetect -y 1
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  70  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

Taper la commande `sudo i2cdetect -y 1` et regarder le chiffre dans le tableau.

Dans le code, modifier la configuration comme cela : `address=0x<votre adresse>`

Exemple du résultat final : `display = BigSeg7x4(i2c, address=0x70)`

Si vous oubliez cette étape, le script risque de ne pas se lancer et de générer une erreur.

2. Nombre de minutes du timer :

Vous devez configurer le nombre de minutes. Il faudra changer la valeur de la constante "DEFAULT_TIME". L'exemple ci-contre représente un timer de 14 minutes et 35 secondes.

```
DEFAULT_TIME = [1,4,3,5]
```

4^{ème} étape : lancement du programme

Il ne vous reste plus qu'à lancer le script. Pour cela, entrez la commande `sudo python3 serv.py`. N'oubliez pas de noter l'adresse IP du raspberry, elle vous sera utile lors de l'installation du menu sur les bornes d'arcades.

5^{ème} étape (facultatif) : automatiser le lancement

Vous pouvez utiliser des Cron Job afin d'automatiser le lancement du script python au démarrage du raspberry. Alors, vous n'aurez qu'à démarrer l'OS pour que le serveur soit déjà prêt à fonctionner sans avoir besoin d'effectuer la 4^{ème} étape.

Premièrement, déplacez le fichier du script dans le dossier `/bin` de Linux.

Ensuite, pour créer le cron vous devez entrer la commande `sudo crontab -e`. Un fichier de configuration s'ouvrira dans votre éditeur de texte (vim, nano, etc...). Tout à la fin de ce fichier, vous devez écrire la ligne suivante : `@reboot python /bin/serv.py &`.

Pour tester que cela a fonctionné, redémarrez et envoyez une requête au serveur web. S'il répond sans que vous l'ayez vous-même lancé, cela veut dire que votre installation est terminée.

V. Informations supplémentaires :

Pour récupérer le code : [GitHub](#)

Product Owner : Katia MOTA & Francisco GARCIA

Si vous avez besoin de plus d'informations, n'hésitez pas à contacter jeremie.arcdc@eduge.ch.