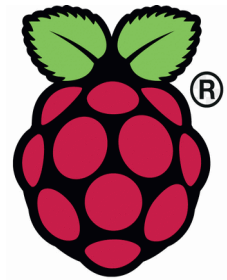


Premiers pas avec Raspberry Pi



Objectifs : Être capable d'installer le système d'exploitation sur un Raspberry Pi3, coder en C des applications utilisant les GPIOs et l'interface Piface Digital.

Date du document : février 2020

Mise à jour : février 2021 (Modif ordre parties), février 2022 HelloWorld.c

Version matérielle et logicielle : RaspberryPi 3 + Buster

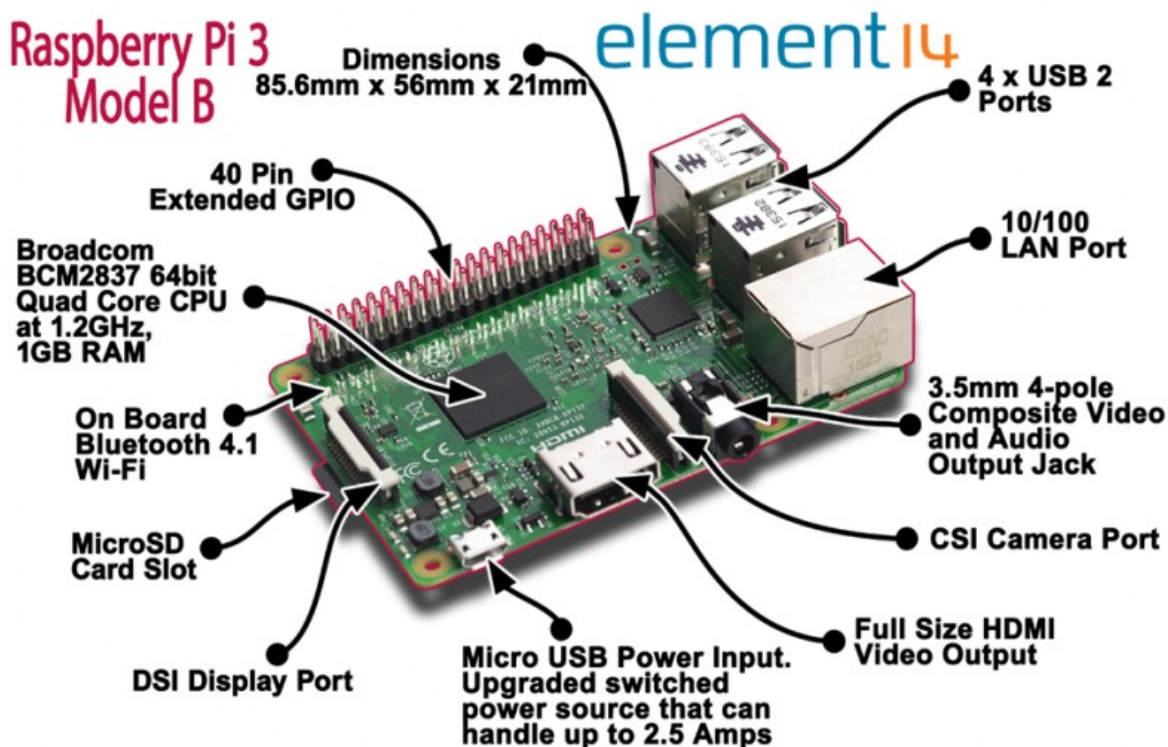
1. Présentation

Le Raspberry Pi

Le Raspberry Pi est un nano-ordinateur monocarte à processeur ARM de la taille d'une carte de crédit dédié essentiellement à des applications d'électronique/informatique embarquée. Il permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux (notamment Debian) et des logiciels compatibles.

Le prix d'une carte Raspberry Pi 3B+ en février 2020 est de l'ordre de 35€ TTC

La figure suivante présente quelques caractéristiques ainsi que les interfaces de la carte Raspberry Pi 3.



Le TP

Au cours de ce TP, vous allez prendre en main un Raspberry Pi, vous allez installer son système d'exploitation, développer des applications basiques en C utilisant les GPIOs et la carte d'interface Piface puis configurer le Raspberry Pi afin qu'une application soit exécutée au démarrage de manière autonome en mode Desktop puis en mode Console.

Dans ce TP, nous allons éviter au maximum d'utiliser les lignes de commande qui peuvent parfois décourager rapidement les débutants et travailler autant que possible en mode Desktop, sauf pour la dernière partie.

Matériel à votre disposition

- Carte Raspberry Pi 3
 - Alimentation pour Raspberry Pi 3 (5,1V 2,5A mini USB)
 - Carte SD vierge 8 Go ou 16 Go
 - Clavier USB
 - Souris USB
 - Moniteur HDMI ou Moniteur VGA équipé d'un convertisseur HDMI/VGA.
 - Carte d'interface Piface Digital2
 - fils, leds, resistances
- PC Windows avec logiciels : SD Card Formatter, Win 32 Disk Imager

L'accès au wifi est possible dans la salle de TP. Voir avec l'enseignant pour SSID et clé WPA.

Les activités à réaliser sont indiquées en gras et nommées A1, A2, etc.

2. Installation du système d'exploitation

Pour commencer à utiliser votre carte Raspberry Pi, il est nécessaire d'installer le système d'exploitation (sur votre PC, votre système d'exploitation est certainement Windows 10). Le système d'exploitation officiel du Raspberry Pi est Raspbian. Voici ce que nous trouvons sur la page d'accueil de raspbian.org :

Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that make your Raspberry Pi run. However, Raspbian provides more than a pure OS: it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi.

Debian est une distribution GNU/Linux non commerciale.

L'image permettant d'installer le système d'exploitation sur une carte SD est téléchargeable à cette adresse : <https://www.raspberrypi.org/downloads/raspbian/>

À ce jour, la version en cours de Raspbian est « Buster » qui se décline en trois possibilités :

Raspian Buster with desktop and recommended software

Raspian Buster with desktop

Raspian Buster lite

On travaillera dans un premier temps avec « Raspian Buster with desktop and recommended software », version la plus complète.

Vous allez maintenant effectuer les activités suivantes.

A1 Téléchargez ou récupérez l'image correspondante. Formatez la carte SD avec « SD Card Formatter ».

Attention : formatez la carte SD et pas votre clé USB ou le disque dur du PC ! Décompressez l'archive puis gravez l'image sur la carte SD en utilisant « Win 32 Disk Imager ».

Votre carte micro SD contient maintenant le système d'exploitation de votre Raspberry Pi. Vous allez maintenant faire tourner votre Raspberry.

A2 Raccordez sur votre carte Raspberry les périphériques suivants : clavier, souris, écran. Insérez la carte SD dans l'emplacement prévu puis connectez l'alimentation.

Le Raspberry Pi doit maintenant « booter ».

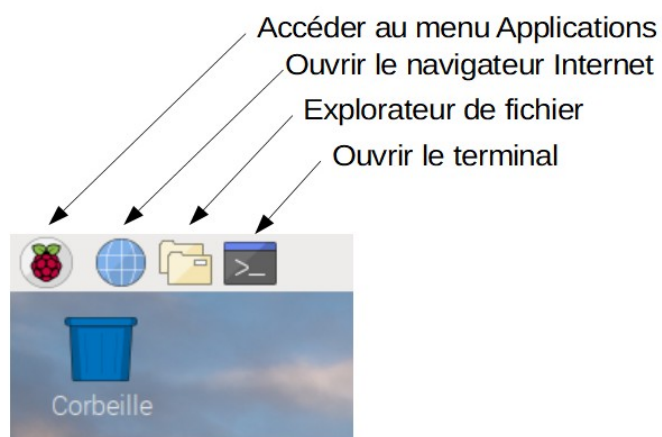
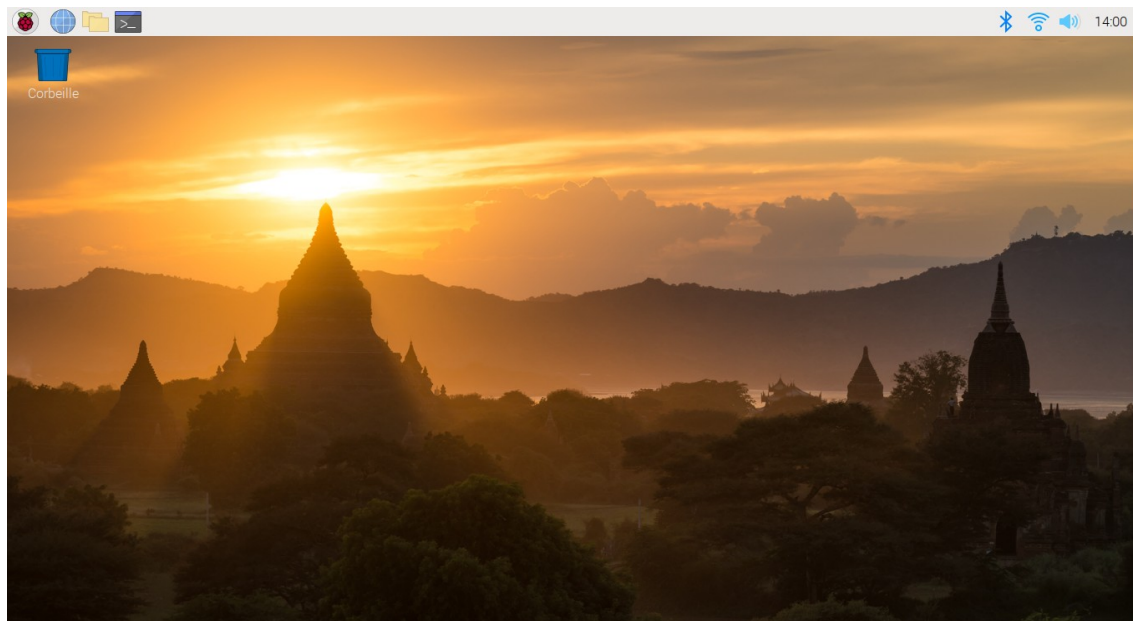
A3 Observez la procédure d'installation et répondez aux questions posées (pays, etc.), vous ne modifierez pas le mot de passe proposé, vous entrerez le code d'accès au wifi.

La procédure d'installation est maintenant terminée.

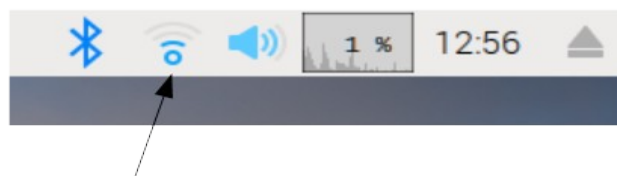
A4 Redémarrez votre Raspberry, vérifiez que l'heure affichée est correcte et que la connexion internet fonctionne bien en ouvrant le navigateur et en effectuant une recherche.

3. Premier contact avec Raspbian

Si tout c'est normalement bien passé, vous devez avoir sur votre écran quelque chose qui ressemble à ça.



Détail écran d'accueil, en haut à gauche



Indicateur champ Wifi, mais aussi accès au paramétrage (par clic droit) des accès réseaux wifi et filaire

Détail écran d'accueil en haut à droite

Pour installer des applications

Cliquer sur Framboise (Accéder au menu application), puis Préférences puis Add/remove software.

Pour modifier la configuration du Raspberry Pi

Cliquer sur Framboise, puis Préférences puis configuration du Raspberry Pi.

Pour ouvrir un terminal

Sous Debian, il est parfois utile d'ouvrir un terminal. C'est l'équivalent de l'invite de commande sous Windows.

Voici quelques commandes, document extrait de la ressource suivante :

<http://eduscol.education.fr/sti/sites/eduscol.education.fr/sti/files/ressources/pedagogiques/4346/4346-1-rpi-presentation.pdf>

LINUX mini (les commandes indispensables)

(<http://www.misfu.com/commandes-unix.html>)

- Raspberry Pi est un ordinateur fonctionnant sous le système d'exploitation LINUX-DEBIAN adapté à la Raspberry Pi sous le nom RASPBIAN, il est indispensable pour le configurer et l'utiliser de connaître quelques commandes Linux en mode console
- Une aide exhaustive pour chaque commande est disponible sous linux par :
man nom-de-la-commande

| | |
|--|--|
| sudo (substitute user do) | Entête donnant les droits de super utilisateur (root) Les programmes PYTHON doivent être lancés avec sudo |
| shutdown -r now | redémarrer |
| shutdown -h now | éteindre |
| ls | voir le contenu du dossier |
| Ls -l | voir le contenu du dossier et les droits sur les fichiers |
| pwd | afficher le nom du dossier courant |
| cd dossier | descendre dans l'arborescence |
| cd .. | remonter dans l'arborescence |
| nano nom_fichier | éditer un fichier |
| exit | se déconnecter e la machine |
| cp source destination | copier un fichier |
| mkdir nom | créer un dossier |
| mv source destination | déplacer un fichier/dossier |
| mv nom1 nom2 (dans le même dossier) | renommer un fichier |
| rm fichier et rm -r dossier | supprime un fichier ou un dossier |
| chmod +x nom_du_fichier | Rendre un fichier exécutable |
| ./nom_du_fichier | Exécuter un fichier |

- Pour éviter de taper à chaque fois **sudo**, passez en mode administrateur : **sudo -i**
passer en shell administrateur : **sudo -s** (déconseillé) pour sortir du mode shell : **exit**
- Linux utilise beaucoup les scripts, de petits fichiers de type texte qui permettent l'enchaînement de commandes systèmes. Il existe plusieurs types de scripts systèmes.
Les fichiers BASH sont très utilisés sur Raspberry Pi

Ping fonctionne aussi, comme sous l'invite de commande Windows à la différence qu'il faut en sortir par Ctrl C (le programme ping en permanence).

Ipconfig (invite de commande windows) devient ici ifconfig.

4. Coder en C

Pour coder en C, nous allons utiliser Geany. Pour accéder à Geany : Framboise puis Programmation puis Geany.

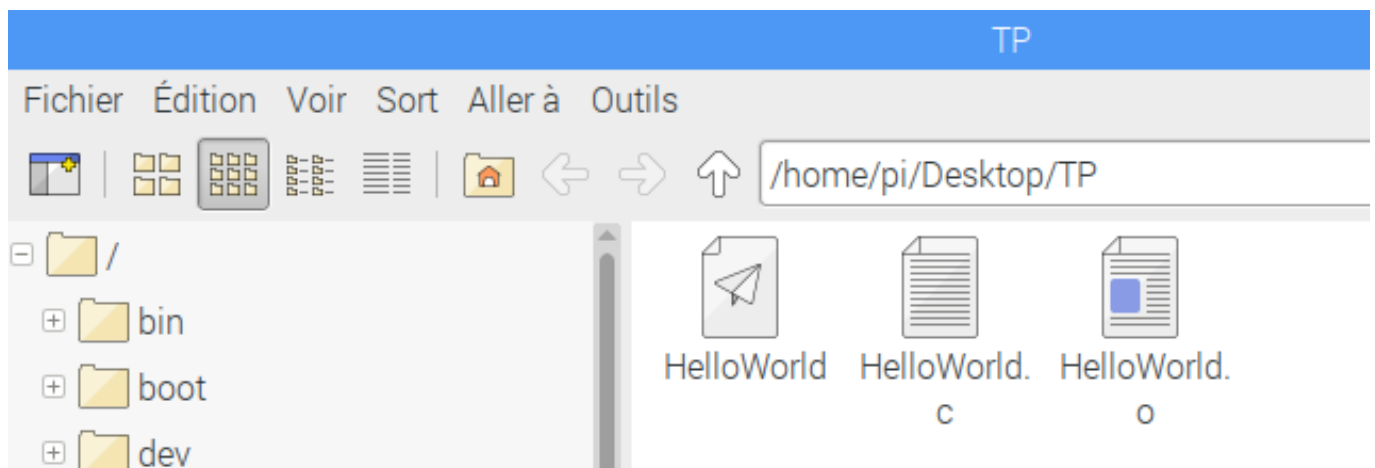
Le premier programme est un grand classique : Hello world dont on rappelle ici le code :

```
#include <stdio.h>

int main()
{
printf("Hello world ! \n");
return 0 ;
}
```

A5 Ouvrir Geany et cliquer sur Document>Définir le type de fichier > Langage de Programmation > Fichier source C. Puis saisir le programme Hello World. La coloration syntaxique doit être effective. L'enregistrer sous le nom HelloWorld.c dans un dossier TP que vous créerez sur le bureau. Cliquer sur Construire puis Compile. S'il n'y a pas d'erreurs, cliquer sur Build. S'il n'y a pas d'erreur vous pouvez cliquer sur Execute pour vérifier le bon fonctionnement du programme.

Inspecter votre répertoire de travail, dans celui-ci doivent maintenant figurer 3 fichiers.



Le fichier avec l'icône « avion en papier » est le fichier exécutable.

A6 En faisant un double-clic sur cette icône, le système vous propose « Exécuter » ou « Lancer dans un terminal ». Essayez ces deux options. Conclure.

A7 Ouvrir une fenêtre de terminal (en haut à gauche) puis déplacer vous à l'aide des commandes ls et cd jusqu'à votre répertoire de travail puis lancer votre exécutable (./HelloWorld). Conclure.

Vous savez maintenant programmer en C sur Raspberry, mais notre objectif n'est pas d'écrire des programmes « console » mais des applications embarquées interagissant avec l'extérieur.

Pour cela, nous allons utiliser les GPIOs.

5. Utiliser les GPIOs

Un des atouts du Raspberry Pi est le connecteur GPIO permettant d'interagir avec l'extérieur en y connectant des capteurs ou des actionneurs.

Sur le Raspberry 3, c'est un connecteur de 40 broches dont le brochage est donné ci-après.

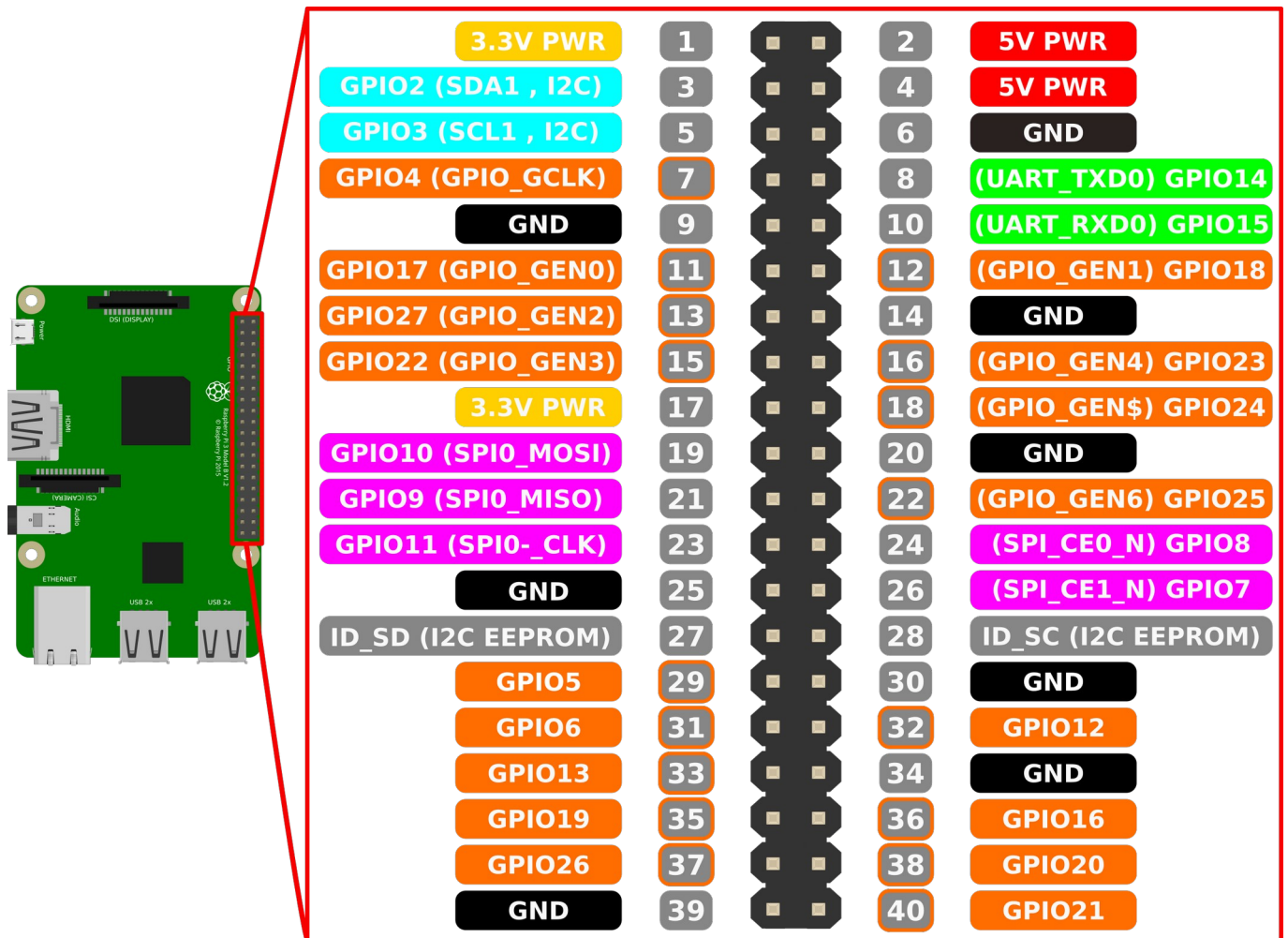


Image extraite du site suivant :

<https://www.framboise314.fr/scratch-raspberry-pi-composants/gpio/>

Quelques remarques :

Contrairement à Arduino ou mbed, il n'y a pas d'entrées analogiques : toutes les entrées-sorties sont numériques.

Les niveaux de tension sont (0V/3,3V). Ne pas mettre de signaux 5V sur les entrées : DANGER !

Pour exploiter les GPIOs, nous allons utiliser la bibliothèque WiringPi. Celle-ci est déjà installée par défaut. Pour aller plus loin :

<http://wiringpi.com/>

La plupart des manipulations effectuées ci-après sont fortement inspirées du document suivant :

<http://innovelectronique.fr/2013/06/09/raspberrypi-a-tout-faire/>

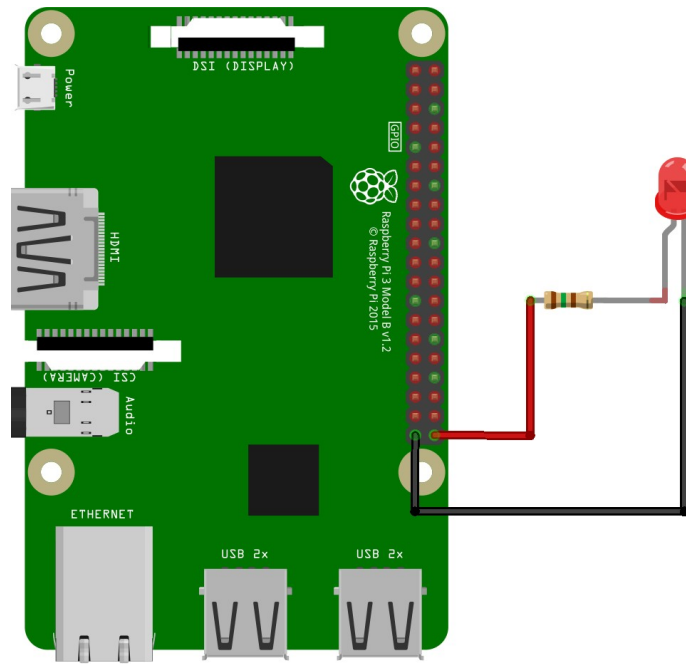
Notre premier programme appelé led va consister à faire clignoter une led connectée à une broche GPIO (encore un grand classique).

Le code est le suivant :

```
#include <stdio.h> // Fonction printf
#include <wiringPi.h> // Fonctions WiringPi

int main () // Fonction principale
{
printf ("Premier test - Clignotement DEL\nFin par Ctrl-C\n"); // Message d'accueil
wiringPiSetupGpio() ; // Choix de la numérotation des broches du RPi
pinMode (21, OUTPUT) ; // Choix mode Sortie de la broche GPIO 4
while (1) // Boucle infinie tant que 1 = 1
{
digitalWrite (21, HIGH) ; // DEL allumée
delay (500) ; // Attente 500ms
digitalWrite (21, LOW) ; // DEL éteinte
delay (500) ; // Attente 500ms
}
return 0 ; // Code retour
}
```

Le schéma de câblage est le suivant (respectez bien la polarité de la led !) :



Pour fonctionner correctement, il est nécessaire de modifier la commande de construction dans Geany. On y accède par Construire>Définir les commandes de construction.

Celle-ci doit être maintenant être la suivante :

gcc -o "%e" "%f" -lwiringPi -lwiringPiDev

Remarque : **-lwiringPiDev** est ici facultative mais sera nécessaire ultérieurement (pour Piface etc.).

A8 Saisir le programme donné, l'enregistrer sous led.c, modifier la commande de construction, effectuer le câblage et vérifier le bon fonctionnement.

A9 Fermer Geany et essayer d'exécuter le programme exécutable de plusieurs façons différentes (par double-clic, par le terminal, etc.).

A10 Question subsidiaire : calculer le courant dans la led.

Il est possible aussi de lire un bit en utilisant `digitalRead(pin)`. Cela est très proche d'Arduino.

Mais pour la suite de ce TP, nous allons utiliserons l'interface Piface qui permet d'éviter de connecter des composants volants sur le connecteur GPIO.

6. Lancer une application au démarrage

Application console

Pour plus d'informations, consulter éventuellement :

<https://forums.framboise314.fr/viewtopic.php?t=3363>

Toutes ces manipulations nous ont permis de développer des applications qui tournent sur le Raspberry. Toutefois dans une application embarquée et aboutie, il n'est pas nécessaire d'avoir un écran, un clavier, une souris, etc.

Nous allons voir ici comment lancer une application de manière autonome au démarrage du Raspberry.

Pour cela nous allons modifier le fichier nommé `rc.local` présent dans le répertoire `/etc`

Ce fichier ne peut pas être modifié avec Geany ou un éditeur graphique. Nous devons utiliser l'éditeur de texte `nano` accessible depuis un terminal.

Le premier programme que nous allons exécuter est le programme `led` (led clignotante).

A11 Copier votre programme exécutable (avion) dans le répertoire `/home/pi`. A l'aide de l'explorateur de fichier repérer et sélectionner le répertoire `/etc`. Vérifier la présence du fichier `rc.local`

A12 Toujours dans l'explorateur de fichier, effectuer un clic droit puis « Ouvrir dans un terminal ». Saisir la commande suivante `sudo nano rc.local`. Ceci a pour effet d'ouvrir l'éditeur de texte `nano`. Juste avant la ligne `exit 0`, saisir les lignes suivantes :

```
cd /home/pi
```

```
./led&
```

Puis sauver et sortir de l'éditeur (les commandes de l'éditeur sont indiquées en bas de la fenêtre `^X = Ctrl-X`).

A13 Rebooter votre Raspberry Pi et vérifier que le programme se lance bien au démarrage.

Application graphique

On peut aussi être amené à utiliser le raspberry pi en lançant une application graphique au démarrage (exemple : navigateur internet, traitement de texte, etc.)

Pour cela, la procédure est donnée ici :

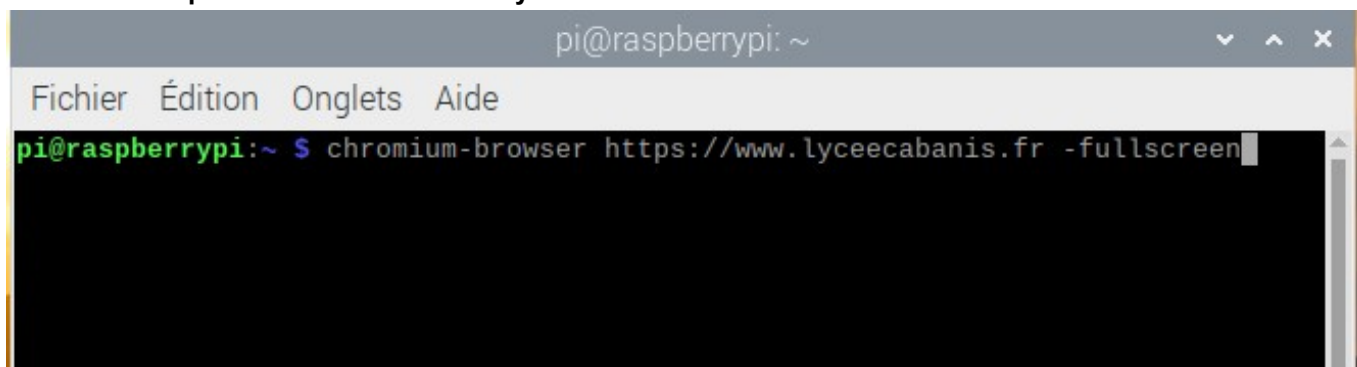
<http://www.siloged.fr/docs/raspberry/index.html?Lancerunprogrammeauboot.html>

Il faut modifier le fichier `/etc/xdg/lxsession/LXDE-pi/autostart` en rajoutant la ligne de commande qui permet de lancer l'application désirée.

Exemple : on désire qu'au démarrage de notre Raspberry Pi, le navigateur internet s'ouvre sur la page du lycée Cabanis (<https://www.lyceecabanis.fr>)

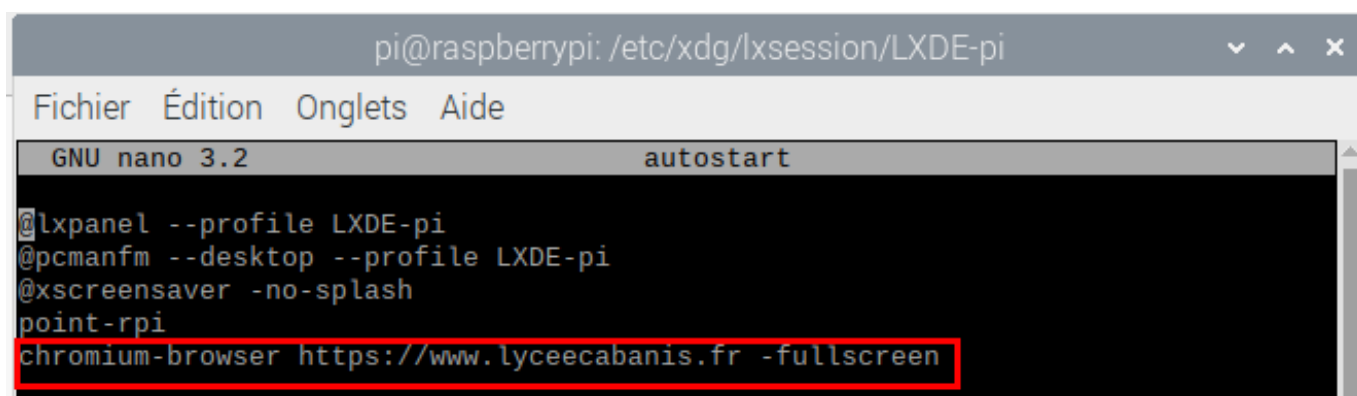
Préalablement, nous allons vérifier la validité de la ligne de commande :

A14 Ouvrir un terminal et saisir la ligne de commande ci-dessous. Vérifier que cela ouvre le navigateur internet en plein écran sur le site du lycée Cabanis.



```
pi@raspberrypi: ~
Fichier  Édition  Onglets  Aide
pi@raspberrypi:~ $ chromium-browser https://www.lyceecabanis.fr -fullscreen
```

A15 Si le test est concluant, vous pouvez modifier le fichier autostart conformément à la figure ci-dessous.

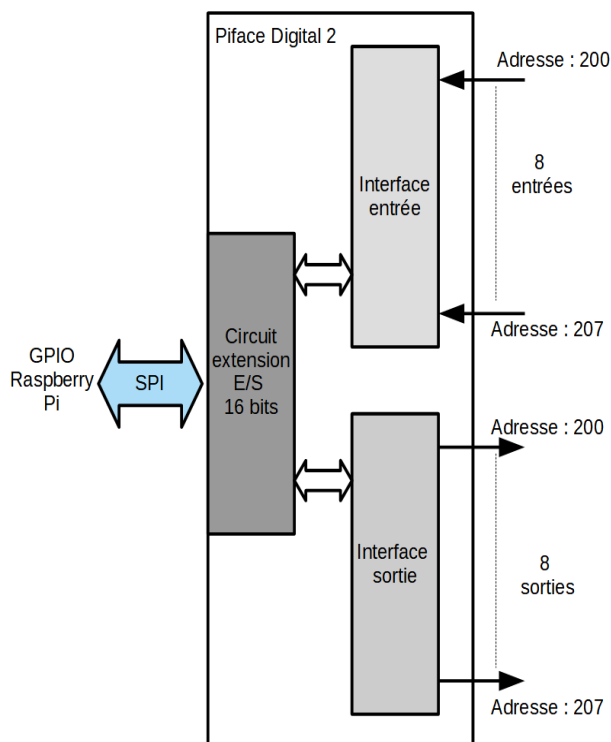


```
pi@raspberrypi: /etc/xdg/lxsession/LXDE-pi
Fichier  Édition  Onglets  Aide
GNU nano 3.2      autostart
@lxpanel --profile LXDE-pi
@pcmanfm --desktop --profile LXDE-pi
@xscreensaver -no-splash
point-rpi
chromium-browser https://www.lyceecabanis.fr -fullscreen
```

A16 Vérifier qu'au démarrage du Raspberry, le navigateur s'ouvre automatiquement en plein écran sur la page du lycée Cabanis.

7. Utiliser Piface

Un schéma structurel détaillé de la carte Piface Digital2 est donné en annexe. Le schéma de principe est le suivant :



La carte d'interface Piface Digital2 dialogue avec le Raspberry par l'intermédiaire de la liaison SPI.

Les 8 entrées sont notées de 0 à 7. Sur les 4 premières entrées sont placés des boutons poussoirs reliés à la masse. Les entrées peuvent être flottante, « pull-upées » ou « pull-downées », configurable par logiciel.

Les 8 sorties notées de 0 à 7 sont des sorties « collecteur ouvert » (équivalent à des interrupteurs à la masse, fermés ou ouverts). Sur chacune des 8 sorties sont connectées des leds qui permettent de visualiser l'état des sorties. Sur les deux premières sorties sont connectés des relais qui permettent de commuter des signaux à plus grande tension ou plus grande intensité.

Des cavaliers permettent de configurer la carte (adresse, alimentation, etc.). On ne touchera pas à ces cavaliers durant la manipulation.

Les entrées et les sorties sont accessibles sur un connecteur à vis.

Nous allons maintenant reprendre la led clignotante mais avec Piface.

A17 Éteindre proprement le Raspberry Pi, débrancher l'alimentation, enficher la carte Piface sur le Raspberry Pi puis rebrancher l'alimentation, attendre le redémarrage du système. Cliquer sur Framboise>Préférences>Configuration du Raspberry Pi> puis Onglet Interfaces et Activer SPI.

Nous allons utiliser l'exemple blink donné par WiringPi disponible à cette adresse :

<http://wiringpi.com/dev-lib/piface/>

Ce programme est aussi donné en Annexe 2, si vous n'arrivez pas à ouvrir le lien.

A18 Copier-coller le programme, le compiler et l'exécuter avec Geany.

A19 Observer quelle led clignote puis modifier le programme pour faire clignoter la première led, puis la dernière led. Que constatez-vous quand la première led clignote ?

A20 Bonus : si vous n'êtes pas trop en retard, écrivez un programme qui allume les 8 leds l'une après l'autre pendant 500 ms puis revient au début, en utilisant une boucle for.

On donne en Annexe 3 un programme qui permet de commander une led en fonction de l'appui sur un bouton-poussoir. Effectuer un « copier-coller » avec une clé usb ou avec un envoi par mail de ce programme (même si ce n'est pas très écoresponsable).

A21 Tester et valider le programme. Quel bouton-poussoir commande quelle led ?

A22 Écrire un programme qui allume les 4 premières leds en fonction de l'appui sur les 4 boutons poussoirs en utilisant une boucle for.

A23 Écrire un nouveau programme nommé bpOnOff.c qui permet d'allumer une led et de commander un relais lorsque on appuie sur un premier bouton-poussoir puis qui éteint la led et bascule le relais lorsque on appuie sur un second bouton-poussoir.

8. Raspberry pi en mode console

Les expérimentations décrites ici sont assez délicates. Cette partie est à faire en « bonus ».

Votre application étant maintenant stabilisée, nous allons utiliser une version plus légère du système d'exploitation. Il n'est pas nécessaire d'avoir un bureau pour lire un bouton poussoir et allumer une led ! L'environnement de travail sera moins convivial et un peu vintage !

A24 Préalablement à toute manipulation, sauvegardez vos fichiers exécutables sur une clé USB.

A25 Téléchargez ou récupérez l'image Raspbian Buster Lite. Formatez la carte SD avec « SD Card Formatter ». Attention : formatez la carte SD et pas votre clé USB ou le disque dur du PC ! Décompressez l'archive puis gravez l'image sur la carte SD en utilisant « Win 32 Disk Imager ». Installez la carte SD et démarrez !

Attention, le clavier est par défaut en anglais. On rappelle login : pi Password = raspberry (taper rqsberry).

A26 Après connexion, configurer le Raspberry en mode console avec l'instruction `sudo raspi-config` (attention au clavier qwerty). Configurer les paramètres régionaux (clavier en particulier), le wifi, le SSH, la liaison SPI.

Le fait de configurer le wifi et le SSH vous permettra de travailler sur le Raspberry à partir de votre pc Windows sous putty. Relever l'adresse ip du Raspberry avec `ifconfig`, mais vous pouvez continuer à travailler avec votre écran et votre clavier.

Il faut maintenant monter la clé USB. C'est parfois un peu difficile... suivre cette procédure :

https://wiki.debian-fr.xyz/Monter_une_cle_usb

Il faut créer un repertoire nommé cle par exemple dans le dossier mnt :

```
cd /      cd mnt      sudo mkdir cle
```

puis monter le disque. La commande `dmseg` permet de voir sous quel device elle est reconnue (sda1 ou sdb1 etc.)

```
puis      sudo mount /dev/sda1 /mnt/cle
```

Astuce : dir /mnt/cle permet de voir les dossier et fichiers présents sur la clé.

A27 Monter la clé USB puis copier le fichier exécutable dans /home/pi avec la commande `cp source destination`.

A28 Installer wiringPi selon la procédure donnée dans le document suivant :

<http://wiringpi.com/download-and-install/>

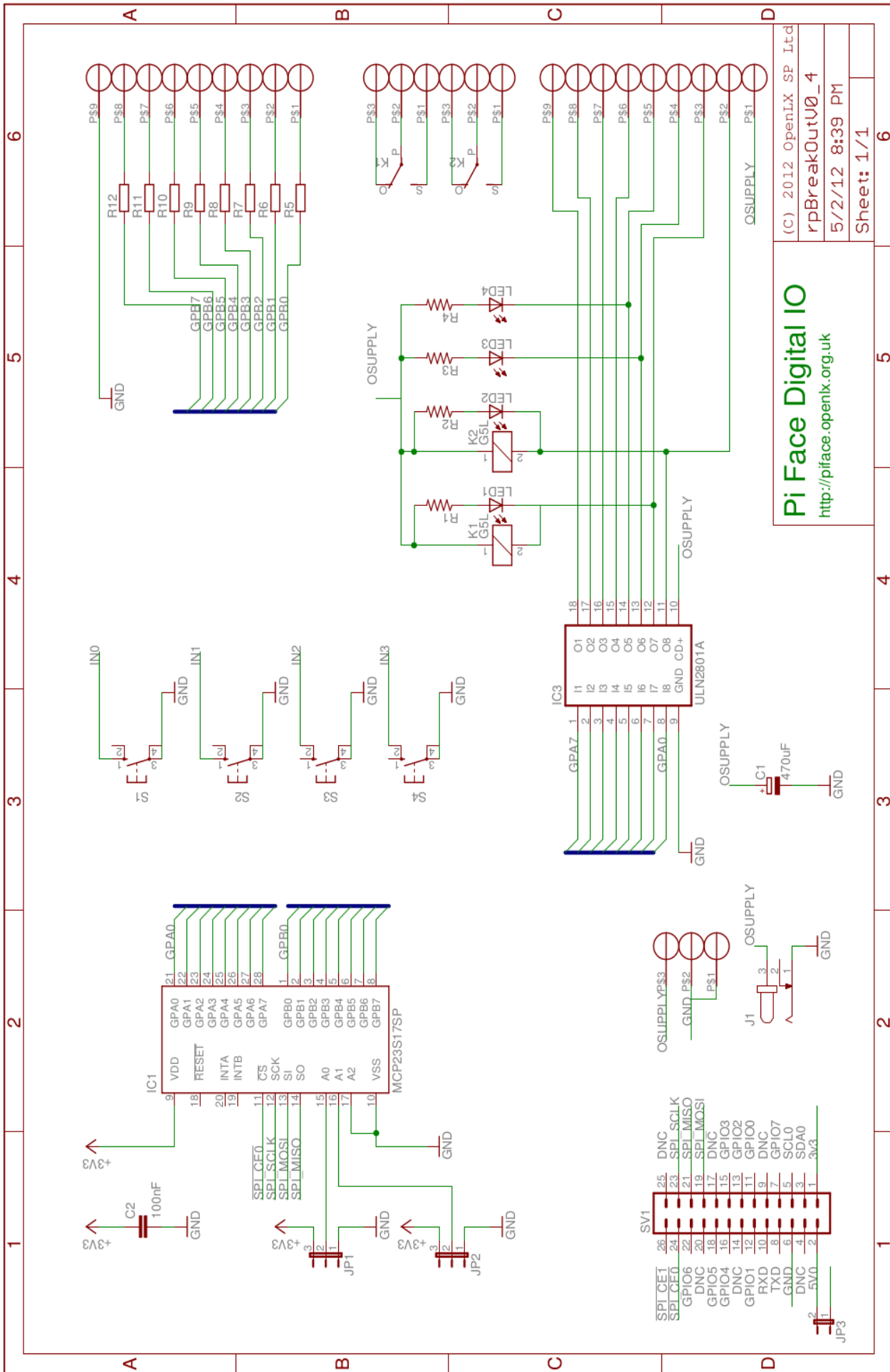
A29 Tester le fonctionnement de votre programme en le lançant dans /home/pi puis `./toto` (si votre programme exécutable s'appelle toto !).

A30 Modifier le fichier `rc.local` comme précédemment et c'est normalement terminé.

A31 Reboot and enjoy....

Bravo et félicitations si vous êtes arrivé jusqu'ici !

Annexe 1 : Schéma Piface Digital2



Annexe 2 : Blinking led Piface

```
/*
 * blink.c:
 *      Simple "blink" test for the PiFace interface board.
 *      ****
 */

#include <stdio.h>

#include <wiringPi.h>
#include <piFace.h>

// Use 200 as the pin-base for the PiFace board, and pick a pin
//      for the LED that's not connected to a relay

#define PIFACE 200
#define LED    (PIFACE+2)

int main (int argc, char *argv [])
{
    printf ("Raspberry Pi PiFace Blink\n") ;
    printf ("=====\n") ;

    // Always initialise wiringPi. Use wiringPiSys() if you don't need
    //      (or want) to run as root

    wiringPiSetupSys () ;

    // Setup the PiFace board

    piFaceSetup (PIFACE) ;

    for (;;)
    {
        digitalWrite (LED, HIGH) ; // On
        delay (500) ;               // mS
        digitalWrite (LED, LOW) ;   // Off
        delay (500) ;
    }

    return 0 ;
}
```

Annexe 3 : Lire une entrée et écrire une sortie avec PiFace

```
/*
 * bp_led
 * Lire une entrée (bouton poussoir) et écrire une sortie (led)
 ****
 */

#include <stdio.h>

#include <wiringPi.h>
#include <piFace.h>

// Use 200 as the pin-base for the PiFace board, and pick a pin

#define PIFACE 200
#define LED (PIFACE+1) //201
#define BP (PIFACE) //200

int main ()
{
    printf ("Raspberry Pi PiFace Lect/ecriture \n") ;
    printf ("=====\n") ;

    // Always initialise wiringPi. Use wiringPiSys() if you don't need
    // (or want) to run as root

    wiringPiSetupSys () ;

    // Setup the PiFace board

    piFaceSetup (PIFACE) ;
    pullUpDnControl(BP,PUD_UP); // pull-up du bouton-poussoir

    for (;;)
    { int etat_bp = digitalRead(BP) ;
      printf( "lecture BP = %d\n",etat_bp);
      if (etat_bp == 0)
      {
          digitalWrite (LED, HIGH) ; // On
      }
      else
      { //
          digitalWrite (LED, LOW) ; // Off
      }
    }

    return 0 ;
}
```