Panneau Leds

Stan Le Guen / Joris Pegon

Introduction au projet:

Nous avions pour but d'afficher un message sur un panneau led de 3 manières différentes représentées sous la forme de 3 scénarios que voici:

- Scénario 1: Un utilisateur programmeur ayant accès au code source modifie le message à afficher, puis exécute le programme
- Scénario 2:Un utilisateur expérimenté entre le message en argument lors de l'exécution du programme avec la console de l'OS RaspBian
- Scénario 3:Un utilisateur lambda utilise une interface web avec d'afficher le message simplement en appuyant sur un bouton "run"

Répartition des tâches:

Joris :

- Fabrication de l'enveloppe corporelle du panneau led
- Identification des nappes et connectiques
- Réalisation des connectiques (soudures, câblages)
- Installer une solution matérielle
- Assistance dans la recherche de solutions

Stan :

- Installation et configuration de Raspbian Jessie
- Installation et tests de la chaîne de compilation croisée
- Recherche de solutions (Coder / Tester / Debugger)
- Installer une solution logicielle
- Installation du serveur web

Bilan matériel et logiciel utilisés

Bilan matériel:

- 1 Panneau led
- 1 Bornier
- 2 Connecteurs HE10
- 2m de nappes
- une clé USB
- 1 Raspberry Pi
- Carte micro SD
- 1 Boîtier RPI
- Module wifi
- Module Pi face

Bilan Logiciel:

- Raspbian
- Smartty
- StarUML
- Notepad++
- CodeBlocks
- Serveur Web
- Redmine

Afin d'illustrer les scénarios présentés précédemment , voici deux diagramme résumant la situation:

StarUml1.PNG

StartUml2.PNG

Conception de l'enveloppe du panneau LED

- **Ci joint des photos de celle ci:**

Panneau_Leds.jpg

Panneau_Leds2.jpg

AlimPC.jpg

Arrière.jpg

ArrièreLeds.jpg

ArrièreLedsCablés.jpg

Mesure a l'oscilloscope

- **On a relevé la tension entre chaque borne des leds rouges et vertes grâce à un oscilloscope lorsqu'on faisait clignoter les leds (horloge). Voici ci-dessous le graphique obtenu : on peut bien remarquer le changement d'état des leds correspondant**

Oscillo.jpg

Partie 1 : Installation matérielle et logicielle

- Stan s'est occupé de la partie installation logicielle englobant l'installation / la configuration du RaspBian Jessie & l'installation / la configuration de l'environnement de développement.
- Joris, pendant ce temps la, s'est occupé de l'enveloppe corporelle du panneau LED et du câblage de la Raspberry

Raspberry.jpg

On a relevé la tension entre chaque borne des leds rouges et vertes grâce à un oscilloscope lorsqu'on faisait clignoter les leds (horloge). Voici ci-dessous le graphique obtenu : on peut bien remarquer le changement d'état des leds correspondant

K1K2.jpg

1.1 Installation du RaspBian Jessie

Il a tout d'abord fallu installer la chaine de compilation permettant alors de compiler les programmes futurs sont notre ordinateur , pour cela un logiciel nous était fourni : GNU toolchain

Toolchain.PNG

Cela a vite été suivi par le test de compilation d'un programme simple écris à travers l'outil Notepad++ et compilé dans l'invite de commande avec la commande " arm-linux-gnueabuhf-gcc.exe -g bonjour.c -o bonjour "

programmetest.PNG

compilation1.PNG

Cela a donc crée un fichier exécutable par la Raspberry et non par notre machine Windows.
Nous avons ensuite relevé l'adresse IP de notre Raspberry avec la commande : "ip a"

1.2 SmarTTY

SmarTTY est un client SSH fourni avec la chaine de compilation vu précédemment, il sert ainsi d'interface entre notre machine Windows et la Raspberry
Nous avons alors configuré une nouvelle connexion pour SmarTTY:

SmarTTY.PNG

Partie 2 : OS et Programmes tests

A cet instant , il était possible de "drag'n drop" notre exécutable dans le dossier prévu à cet effet.
Nous avons donc autorisé l'exécution de notre binaire avec la commande " chmod +x bonjour " puis exécuté celui ci avec la commande ./bonjour

bonjour.PNG

2.1 Code::Blocks

Code::Blocks est l'environnement de développement nous ayant permis de développer nos programmes de tests avant d'arriver aux solutions demandées.

codeblocks.PNG

codebonjour.PNG

2.2 Programmes tests

Les programmes tests les plus importants ont étés les suivants :

Le programme suivant a servi à déterminer si les entrées et sorties de la carte Raspberry fonctionnaient

blinkgpio.PNG

Le programme suivant est un extrait d'un programme permettant l'affichage en statique de 4 "A" sur le panneau LED, en utilisant des tableau 1D

aaaa.PNG

Ce dernier programme permet encore une fois l'affichage de 4 "a" mais cette fois ci défilants et utilisant un seul tableau 2D

aaaa2d.PNG

Partie 3 : Serveur Web

- Afin d'avoir la possibilité d'écrire un message a travers une interface web, il nous faut l'application "Apache2". Avec cette application , l'utilisateur lambda n'a plus qu'a taper le message voulu :

apache2.PNG

- Il conviendra ensuite d'installer php avec la commande : sudo apt-get install php5 libapache2-mod-php5 -y puis de créer un dossier "/var/www/" avec sudo mkdir.
  Ceci s'affiche alors :

versionphp.PNG

- On copie alors les ficher dans le répertoire "panneauLeds" de la Raspberry :

repertoire.PNG

- Test de l'installation dans le serveur :

panneauled.PNG

- Modification du chemin de l'exécutable dans le fichier "runApp.php".

cheminexe.PNG

- Ajout des droits au groupe "www-data" dans le ficher /etc/sudoers.

modifsudoers.PNG

- Lorsque toutes ces étapes sont effectuées avec succès, alors il est possible désormais d'envoyer un message à travers une interface web , le produit est donc délivrable à l'utilisateur lambda

Files

Panneau_Leds.jpg	119 KB	20/02/2021	Joris PEGON
PanneauLeds2.jpg	123 KB	20/02/2021	Joris PEGON
Arrière.jpg	58.5 KB	20/02/2021	Joris PEGON
ArrièreLeds.jpg	66.3 KB	20/02/2021	Joris PEGON
ArrièreLedsCablés.jpg	75.3 KB	20/02/2021	Joris PEGON
AllimPC.jpg	85.9 KB	20/02/2021	Joris PEGON
K1K2.jpg	90.1 KB	20/02/2021	Joris PEGON
Oscillo.jpg	84.5 KB	20/02/2021	Joris PEGON
Raspberry.jpg	64.6 KB	20/02/2021	Joris PEGON
StarUml1.PNG	64.1 KB	21/02/2021	Stan Le Guen
StarUml1.PNG	48.6 KB	21/02/2021	Stan Le Guen
StarUml1.PNG	64.1 KB	21/02/2021	Stan Le Guen
Toolchain.PNG	298 KB	21/02/2021	Stan Le Guen
programmetest.PNG	101 KB	21/02/2021	Stan Le Guen
compilation1.PNG	201 KB	21/02/2021	Stan Le Guen
SmarTTY.PNG	203 KB	21/02/2021	Stan Le Guen
bonjour.PNG	199 KB	21/02/2021	Stan Le Guen
blinkgpio.PNG	11.9 KB	21/02/2021	Stan Le Guen
aaaa.PNG	35 KB	21/02/2021	Stan Le Guen
aaaa2d.PNG	30.4 KB	21/02/2021	Stan Le Guen
codebonjour.PNG	53.4 KB	21/02/2021	Stan Le Guen
codeblocks.PNG	29.2 KB	21/02/2021	Stan Le Guen
apache2.PNG	259 KB	22/02/2021	Stan Le Guen
versionphp.PNG	101 KB	22/02/2021	Stan Le Guen
repertoire.PNG	30.6 KB	22/02/2021	Stan Le Guen
panneauled.PNG	9.5 KB	22/02/2021	Stan Le Guen

cheminexe.PNG	61 KB	22/02/2021	Stan Le Guen
modifsudoers.PNG	37.2 KB	22/02/2021	Stan Le Guen