Panneau Leds

VIG Elena / MATA Alexis

Présentation du projet:

Le projet que nous devions réaliser consiste à afficher des messages sur un panneau leds selon 3 scénarios donnés :

1. Un utilisateur programmeur peut accéder, par la console de l'OS raspbian, au code source afin de le modifier et d'interagir avec le panneau LED.
2. Un utilisateur confirmé peut entrer le message qu'il souhaite afficher sur le panneau LED en l'entrant en argument dans la console de l'OS raspbian.
3. Un utilisateur lambda peut accéder à une page web qui lui permet de saisir un message et de l'afficher en appuyant sur « RUN ».

Répartition des tâches:

Elena :

- Construction de la boîte du panneau
- Diagramme de déploiement et diagramme d'activité
- Réalisation des connexions / soudures / câbles
- Installation / Test de la chaîne de compilation
- Coder / Tester / Debugger
- Modifier le programme afin d'exploiter les arguments de la fonction principale

Alexis :

- Installation / configuration de Raspbian Jessie
- Réalisation des connexions / soudures / câbles
- Mesure et relevé des signaux éléctriques (horloge)
- Installer une solution matérielle/logicielle
- Installer le serveur web

Matériels et logiciels 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
- Monia
- CodeBlocks
- Serveur Web
- Redmine

Installation / Configuration Raspbian Jessie

- Nous avons dézippé le fichier compressé '2021-01-22-raspbian-jessie-8.zip', puis installé l'utilitaire 'md5check_setup.exe', puis copié le fichier qu'on vient de décompresser. Enfin, nous avons installé puis formaté l'image Raspbian sur la carte SD.
  raspbian.PNG

- On commence ensuite la configuration en vérifiant si l'adresse IP est bien connectée sur le bon réseau, on regarde si le système est bien à jour, que le service ssh est activé et que la librairie wiringPI soit bien installée sur notre raspberry.
  Voici le raspberry installé et branché :
  branchement_raspberry.jpg

Diagramme de déploiement des cas d'utilisations et leurs scénarios

diagramme.PNG

Réalisation des connexions / soudures / câbles et test

- La soudure des composants de la carte mère de notre Raspbian Jessie étant déjà réalisée, il nous a fallu réaliser les connexions des câbles K1 et K2, puis les brancher correctement.

- Ensuite, pour les tester, il a fallu faire 2 leds (une rouge, une verte) avec une résistance, et ensuite les brancher sur la précédente monture avec la borne négative connectée au GND, et la borne négative connectée sur le GPIO 22 pour la led rouge et le GPIO 17 pour la led verte.
  Si les leds s'allument correctement et réagissent aux instructions données dans la console avec le GPIO, le test des connexion peut donc être validé.

Construction de la boîte du panneau leds

- En premier lieu, pour démarrer ce mini projet, on a du construire une boîte en carton et polystyrène refermable permettant au panneau leds de tenir debout.

PanneauLeds.jpg

Mesure et relevé des signaux électriques (horloge)

- 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

-~~o~~scilloscope.PNG

Installation / Test de la chaîne de compilation

- Installation de la chaîne de compilation Gnu :

gnu.PNG

- Création d'un exécutable qui ne fonctionne pas sous windows destiné au raspberry.
  Pour transférer ce fichier sur le Raspberry, nous utilisons le client SSH de l'utilitaire SmarTTY fourni avec la chaîne de compilation:

smartty.PNG

- Il faut sauvegarder la clé publique fournie par SmarTTY.

- Ensuite, nous avons testé le projet 'hello' en mode debug . Il faut lancer le serveur gdb sur le raspberry avec la commande suivante : 'gdbserver :1234bonjour'

bonjour.PNG

- On lance alors le client gdb grâce a la commande 'arm-linux-gnueabihf-gdb.exe bonjour', et on peut observer la réponse suivante sur SmarTTY :

gdb1234.PNG

- Place ensuite à la configuration de codeblocks

console.PNG

- Enfin, le projet "bonjour" peut être compilé.

compiler.PNG

- On exécute le binaire de notre projet et le message 'Hello world!' s'affiche. On vient de réaliser notre premier projet avec l'IDE de CodeBlocks.

*Maintenant on va faire une compilation croisée donc il faut changer les paramètres.

croisée.PNG
*On édite le fichier source bonjour.c et on le compile, si les paramètres sont bons l'IDE nous envoie le message 0 errors, 0 warnings.

*L'exécutable bonjour.exe se trouve dans le dossier de notre projet qu'on transfère dans la raspberry .

smarttypi.PNG
*On lui autorise l'exécution avec la commande chmod +x bonjour.exe et on l'exécute avec ./bonjour.exe. Le message qu'on vient de programmer s'affiche dans la console du raspberry.

*Ensuite on va mettre à jour le répertoire présent sur notre machine Windows en exécutant le script UpdateSysroot.bat.

*On a fini les test de la chaîne de compilation, on peut faire notre programme pour la solution finale.

Coder / Tester / Debugger

- Tout d'abord, nous avons dû tester pas à pas plusieurs programmes donnés pour comprendre les différentes fonctions permettant de faire fonctionner les leds.

- Ensuite, on a réalisé un programme permettant de faire apparaître le message "AAAA" sur le panneau grâce à un tableau 1D. Puis, une fois le programme réalisé et testé, avec un tableau 2D.
  Voici en exemple le programme avec le tableau 2D :

tableau2D.png

- Enfin, une fois que nous avions compris le fonctionnement du programme avec le tableau 2D et les différentes fonctions, on a pu créer un programme affichant "Bonne année 2021"

- Une fois toutes ces étapes réalisées, nous avons pu réfléchir au programme final permettant d'afficher le message tapé dans la console.

Installation du serveur web

- Un serveur web doit être installé (avec Apache2) pour l'utilisateur lambda. Grâce à l'application ainsi créée, celui-ci n'a plus qu'à taper le message qu'il veut pour l'afficher sur le panneau leds :

apache2.PNG

- Ensuite, il convient d'installer php avec la commande 'sudo apt-get install php5 libapache2-mod-php5 -y', puis un répertoire dans '/var/www/ sudo mkdir.
  Cette page s'affiche alors, lorsque tout est installé :

apache2.1.PNG

- Copie des fichiers dans le répertoire 'panneauLeds' de la raspberry :

indexphp.PNG

- Test de l'installation dans le serveur :

siteweb.PNG

- Modification du chemin de l’exécutable dans le fichier «runApp.php».

runAppphp.PNG

- Enfin, il faut rajouter les droits au groupe «www-data» dans le fichier /etc/sudoers.

droits.PNG

- Toutes ces étapes réalisées avec succès, le serveur web est correctement créé et peut être mis à la disposition du client.

Files

siteweb.PNG	13.7 KB	20/02/2021	Alexis Mata
Code.png	98.2 KB	20/02/2021	Alexis Mata
siteweb.PNG	17.2 KB	20/02/2021	Alexis Mata
Code.png	98.2 KB	20/02/2021	Alexis Mata
ProgrammeAAAA.jpg	466 KB	20/02/2021	Alexis Mata
ProgrammeAAAA.jpg	191 KB	20/02/2021	Alexis Mata
raspbian.PNG	91.3 KB	20/02/2021	Alexis Mata
branchement_raspberry.jpg	142 KB	20/02/2021	Alexis Mata
branchement_raspberry.jpg	77.7 KB	20/02/2021	Alexis Mata
raspbian.PNG	71.1 KB	20/02/2021	Alexis Mata
apache2.PNG	259 KB	21/02/2021	Alexis Mata
apache2.1.PNG	102 KB	21/02/2021	Alexis Mata
index.php.PNG	5.59 KB	21/02/2021	Alexis Mata
runApp.php.PNG	32 KB	21/02/2021	Alexis Mata
droits.PNG	35.8 KB	21/02/2021	Alexis Mata
apache2.PNG	259 KB	21/02/2021	Alexis Mata
apache2.1.PNG	102 KB	21/02/2021	Alexis Mata
diagramme.PNG	28.4 KB	21/02/2021	Alexis Mata
diagramme.PNG	28 KB	21/02/2021	Alexis Mata
PanneauLeds.jpg	143 KB	21/02/2021	Alexis Mata
indexphp.PNG	5.59 KB	21/02/2021	Alexis Mata
runAppphp.PNG	62.8 KB	21/02/2021	Alexis Mata
indexphp.PNG	33.4 KB	21/02/2021	Alexis Mata
gnu.PNG	165 KB	21/02/2021	Alexis Mata
gnu.PNG	165 KB	21/02/2021	Alexis Mata
smartty.PNG	169 KB	21/02/2021	Alexis Mata
bonjour.PNG	71.8 KB	21/02/2021	Alexis Mata
gdb1234.PNG	201 KB	21/02/2021	Alexis Mata
compiler.PNG	111 KB	21/02/2021	Alexis Mata
console.PNG	133 KB	21/02/2021	Alexis Mata
compiler.PNG	111 KB	21/02/2021	Alexis Mata
croisée.PNG	117 KB	21/02/2021	Alexis Mata
smarttypi.PNG	142 KB	21/02/2021	Alexis Mata
oscilloscope	2.16 MB	21/02/2021	Alexis Mata
oscilloscope.PNG	620 KB	21/02/2021	Alexis Mata
tableau2D.PNG	26.1 KB	09/03/2021	Elena Vig