Panneau_Leds_MathieuR_Lucas

Sommaire

Introduction

Présentation du Projet

Réalisation d'un projet permettant d'afficher un message déroulant sur un panneaux leds 8x32, 3 scénario sont envisager
1) Un utilisateur avancé 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 intermédiaire 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 ».

Cahier des charges

Diagramme d'utilisation simplifié

8vow.png

Diagramme de déploiement

x9w4.png

Liste du matériel

Lors de ce mini-projet nous avons eu besoin de:
1 Panneau LED 8x32
2 Connecteur HE10
1m de Nappe
Alimentation PC 230V
Fer a souder
1 Oscilloscope
1 Voltmètre
1 Raspberry Pi b+
1 Câble Ethernet
1 Adaptateur HDMI vers DVI;
1 Câble HDMI
Modules borniers
Module Pi Face
1 Alimentation Pi
1 Carte Micro SD 16Go avec un adaptateur USB/SD.

Liste des logiciels

Lors de ce mini-projet, nous avons eu besoin de:
- CodeBlocks;
- SmarTTY;
- StarUML;
- Apache2.

Répartition des taches

Mathieu:

- Diagramme des cas d'utilisation/écrire les scénarios associés
- Installation/Configuration Raspbian Jessie
- Fabrication d'un boîtier pour le panneaux leds
- Diagramme de déploiement
- Diagramme d’activités
- Installation de la toolchain
- Test d'un programme en compilation croisée

- Fonctionnement des panneaux (Registre à décalage)
- Programmes de test
- Prototypage
- Codage intégration

Lucas:

- Fabrication d'un boîtier pour le panneaux leds
- Réalisation matérielle et tests connexions
- Comprendre l'interconnexion des différents matériels
- Tester les sorties GPIO avec 2 leds connectés
- Test d'un programme en compilation croisée
- Relevé des signaux (Horloge à partir d'un timer logiciel)
- Fonctionnement des panneaux (Registre à décalage)
- Programmes de test
- Prototypage
- Codage intégration
- Intégration serveur web

Création du Panneau LED

Construction du panneau

Création de la boite du panneau

Création des câbles

Câblage au panneau et Raspberry

Schéma de connexion du Raspberry au panneau
 mu1t.png
Alimentation du panneau
 ojpg.png

Installation des logiciels

Code::Blocks

Compilateur ARM sur Windows

SmarTTY (Client SSH & SCP)

Apashe Serveur Web

Mise en forme

Pour la mise en forme de la page, nous utilisons le code html suivant:

 sin4.png

Installation & configuration de Raspbian

Développement

Développement du programme

Scénario avec utilisateur avancé

Dans ce scénario, l'utilisateur aura accès au code source directement afin de modifier le message qui s'affichera sur le panneau LED.
Pour cela nous avons utilisé le **C**/**C++**:

 7kq8.png

Scénario avec utilisateur intermédiaire

Dans ce scénario, l'utilisateur intermédiaire a accès au terminal du raspberry, et peut choisir le texte affiché en le mettant en

argument de la commande d'exécution du programme. Pour cela, nous avons ajouté le programme suivant, permettant de récupéré les arguments saisies dans la console, et de les mettre sous forme de chaine de caractères, au programme précédent.

rsuc.png

Scénario avec utilisateur lambda

Dans ce scénario, notre client pourra choisir directement sur le service web la phrase qu'il voudra afficher sur le panneau. Le programme C/C++ est le même que dans le scénario précédent.
Pour que le client puisse choisir, nous avons intégré un **programme php** à notre site web lui permettant d'exécuter le programme à distance, avec en argument le texte choisis par l'utilisateur sur un champ sur le site.

zeh5.png