[Date]

Adrien BRITON

[nom de la société]

[Titre du document]

[Sous-titre du document]

Table des matières

[I. Description du projet 2](#_Toc498718061)

[II. Détail des fonctionnalités 3](#_Toc498718062)

[III. Répartition des tâches 4](#_Toc498718063)

# Description du projet

Le but de ce projet est de développer une application permettant de commander l’ouverture et la fermeture d’un portail, modélisés par le contrôle de deux servo-moteurs.

L’activation sera effectuée soit par une détection de mouvement par un capteur spécifique ou par action de l’utilisateur.

Une application Labview servira d’interface entre l’utilisateur et le portail.

Les principaux modules sont les suivants :

Le système détecte la présence d’un individu grâce au capteur, puis agit sur les servo-moteurs. Afin de garantir la sécurité de la personne traversant, un système de temporisation et de détection permanent est mis en place. Lorsqu’il n’y a plus d’obstacles, le système referme le portail.

Il existe aussi un système modélisant l’utilisation d’un interphone par utilisation d’un bouton poussoir : lorsque l’individu active le bouton poussoir, l’ouverture du portail est réalisée par retour Labview.

A l’ouverture ou à la fermeture du portail, l’afficheur et les LEDs sont utilisés afin de signifier l’état de fonctionnement du système.

# Détail des fonctionnalités

I2C

Port B

UART

Port A

SPI

Port A

SPI

Port A

SPI

Port A

PWM

Port C

UART

Port A

1. Présentation des fonctions
   1. Configuration de la carte ISEN

* Void init\_GPIO (void) : sert à configurer les différents périphériques de la carte ISEN
  1. Configuration des servo-moteurs :
* Void init\_Timer2 (void) : permet l’activation du timer sur la carte
* Void init\_PWM2 (void) : permet la configuration du PWM
  1. Configuration des afficheurs
* Void LED (void) :
* Void init\_digit(void) :
  1. Configuration du buzzer
* Void init\_Timer3 (void) : permet l’activation du timer sur la carte
* Void init\_PWM3 (void) : permet la configuration du PWM
  1. Configuration des boutons poussoir
* Void init\_BP (void) : permet l’activation des BP
* Void enable\_interrupt\_ext (void) : permet d’activer l’interruption sur les BP
  1. Configuration du capteur :
* Uint8 init\_Detection (
  1. dqg
* Void init\_UART (void) :

1. Précision des paramètres I/O

# Répartition des tâches

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Personnes** | *Jonathan* | *Rémy* | *Nour* | *Adrien* |
| **Tâches** | * PWM * Servo-moteurs * Boutons poussoirs | * UART * Labview | * I2C * Capteurs | * SPI * Afficheur * LEDs * Buzzer |

* Fonctions :
  + Ouverture du portail :
    - Détection présence
    - Activation ouverture
    - Clignoter LED
    - Détection ouverture complète
  + Fermeture du portail
    - Tempo
    - Activation fermeture
    - Détection obstacle
    - Clignoter LED
    - Détection fermeture complète
  + Détection de présence
    - Réception signal présence
    - Emission du signal
  + Appel réception
    - Réception signal appui
    - Emission du signal
  + Détection d’obstacles
    - Fonction d’interruption
    - Réception signal obstacle
    - Emission du signal

Clignotement des LED :

Void Clignoter\_Led(void)

{

//Clignotement des LED

Return null ;

}

Détection présence :

Int Detection\_presence()

{

//Détection

Return 0 ou 1 ;

}

Activation ouverture :

Int Ouverture()

{

//Ouverture

Return 0 ou 1 ;

}