**Table des matières**

[Introduction Générale 5](#_Toc101360015)

[Chapitre 1 : Concept général du projet 6](#_Toc101360016)

[**1.** **Introduction :** 6](#_Toc101360017)

[**2.** **Présentation de l’entreprise d’accueil** 6](#_Toc101360018)

[**2.1 test** 6](#_Toc101360019)

[**3.** **Présentation du projet** 6](#_Toc101360020)

[**4.** **Etude de l’existant** 7](#_Toc101360021)

[**5.** **Conclusion** 7](#_Toc101360022)

[Chapitre 2 : exigences et méthodologie de travail 7](#_Toc101360023)

[**1.** **Introduction** 7](#_Toc101360024)

[**2.** **Norme IEC-62 304** 7](#_Toc101360025)

[**3.** **Push-pull-commit (méthodologie de travail)** 7](#_Toc101360026)

[**4.** **Conclusion** 7](#_Toc101360027)

[Chapitre 3 : analyse et spécification de besoins 7](#_Toc101360028)

[**1.** **Introduction** 7](#_Toc101360029)

[**2.** **Spécification des besoins** 7](#_Toc101360030)

[**3.** **Architecture globale « Software –Hardware »** 8](#_Toc101360031)

[**3.1** **Hardware :** 8](#_Toc101360032)

[**3.2** **Software :** 8](#_Toc101360033)

[**4.** **Conclusion** 8](#_Toc101360034)

[Chapitre 4 : conception et réalisation 8](#_Toc101360035)

[**1.** **Introduction :** 8](#_Toc101360036)

[**2.** **Os KERNEL :** 8](#_Toc101360037)

[**2.1** **MiddleWare ( FreeRtos ) :** 8](#_Toc101360038)

[**3.** **Moteur pas à pas** 8](#_Toc101360039)

[**4.** **Interface homme machine  « IHM »** 8](#_Toc101360040)

[**5.** **Capteurs et mesure** 9](#_Toc101360041)

[**6.** **Connectivité** 9](#_Toc101360042)

[**7.** **Tache en cours de développement** 9](#_Toc101360043)

[**8.** **Conclusion** 9](#_Toc101360044)

[Conclusion générale 9](#_Toc101360045)

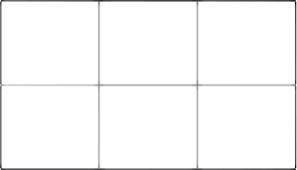
**Tables des figures :**

[Figure 1 2](file:///C:\Users\HP\Downloads\majdi-table.docx#_Toc101189233)

[Figure 2 856jdfjjfj5 6](file:///C:\Users\HP\Downloads\majdi-table.docx#_Toc101189234)

[Figure 3 : azizos 6](file:///C:\Users\HP\Downloads\majdi-table.docx#_Toc101189235)

[Figure 4 56565 6](file:///C:\Users\HP\Downloads\majdi-table.docx#_Toc101189236)



Figure

Liste des tableaux :

[Tableau 1 hhdhhdhh 7](#_Toc101189347)

# **Introduction Générale**

# **Chapitre 1 : Concept général du projet**

## **Introduction :**

## **Présentation de l’entreprise d’accueil**

### **2.1 test**

[place your text here]



Figure : azizos

## **Présentation du projet**

[place your text here]

Figure 56565

## **Etude de l’existant**

[place your text here ]

Tableau 1 hhdhhdhh

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

## **Conclusion**

[place your text here]

# **Chapitre 2 : exigences et méthodologie de travail**

## **Introduction**

[place your text here ]

## **Norme IEC-62 304**

[place your text here]

## **Push-pull-commit (méthodologie de travail)**

[place your text here ]

## **Conclusion**

[place your text here ]

# **Chapitre 3 : analyse et spécification de besoins**

## **Introduction**

[place your text here]

## **Spécification des besoins**

[place your text here ]

## **Architecture globale « Software –Hardware »**

[place your text here]

### **Hardware :**

### **Software :**

## **Conclusion**

[place your text here]

# **Chapitre 4 : conception et réalisation**

## **Introduction :**

## **Os KERNEL :**

### **MiddleWare ( FreeRTOS ) :**

##### **RTOS :**

Le système d’exploitation temps réel est un OS qui gère plusieurs tâches concurrentes selon leurs priorités, il est utilisé quand il y a des exigences temporelles sur les processus.

Le RTOS garantis la performance maximale du processeur et la mémoire ainsi que le fonctionnement sans erreur(error-free).

##### **FreeRTOS/CMSISV2**

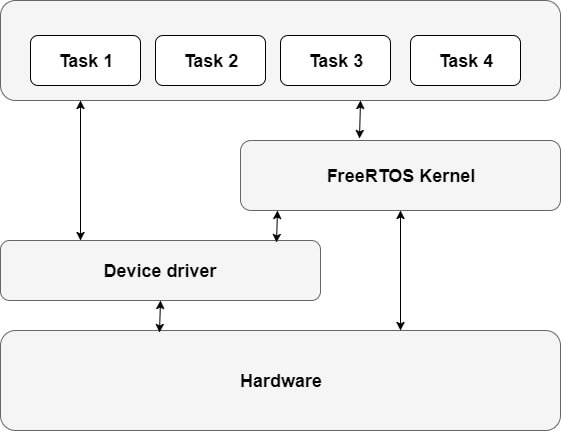
FreeRTOS est un système d’exploitation embarqué multitâches temps réel préemptif supporte actuellement 35 architectures. Il est aujourd’hui parmi les plus utilisés dans le marché des systèmes d'exploitation temps réel pour l’embarquée grâce à sa faible taille qui est de l’ordre de 4000 à 9000 octets.

Figure :Structure freeRTOS

Dans ce projet on a utilisé le CMSISV2 API comme une couche d’abstraction à FreeRTOS afin de faciliter la programmation et garantir un RTOS plus optimisé.

* 1. **Tâches Et Queues (File d’attente) :**

##### **Etat des tâches :**

Pour développer une application basée sur un OS, on décompose l’application en un ensemble de tâches. Dans FreeRTOS une tâche est fonction C contenant une boucle infinie et ne renvoie pas un résultat.

**void vATaskFunction( void \*pvParameters )**

**{**

**for( ; ; )**

**{**

**}**

**}**

Une tâche FreeRTOS peut se trouver dans l’un des états suivants :

**Prête (Ready)** : une tâche qui possède toutes les ressources nécessaires à son exécution. Elle lui manque seulement le processeur.

**Active (Running)** : Tâche en cours d’exécution, elle est actuellement en possession du processeur.

**Attente (Blocked)** : Tâche en attente d’un événement (queue de messages, sémaphores, timeout ...). Une fois l'événement arrivé, la tâche concernée repasse alors à l'état prêt.

**Suspendu (Suspended)** : tâche à l’état dormant ; elle ne fait pas partie de l’ensemble des tâches ordonnançables**.**

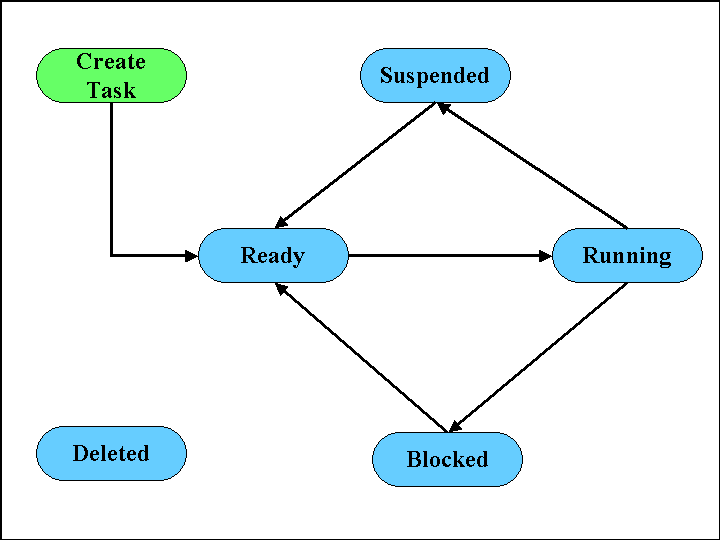


Figure :état des taches

* Tache 1 : stepperHandle :

Cette tache dispose la priorité maximale (osPriorityHigh) car elle gère le moteur pas à pas qui est le cœur du projet et tous les autres services fonctionnent en fonction de son état et de son avancement.

* Tache 2 : IHMHandle :

Cette tache dispose comme priorité (osPriorityAboveNormal), elle gère les flux de données entre l’interface homme machine et les autres taches.

* Tache 3 : ConnectivityHandle :
* Tache 4 : SensorsHandle
* Tache 5 : Battery\_manageHandle
* **Queues (c’est quoi) :**

## **Moteur pas à pas**

#### **Stepper Moteur Vs Servo :**

* 1. **Driver :**
  2. **Library ( l6474.C) :**
  3. **Interruption :**
  4. **Input / Output :**

## **Interface homme machine  « IHM »**

[place your text here ]

## **Capteurs et mesure**

[place your text here ]

## **Connectivité**

[place your text here ]

## **Tache en cours de développement**

[place your text here]

## **Conclusion**

[place your ext here ]

# **Conclusion générale**

[place your text here ]