|  |  |
| --- | --- |
| **Durée : 2 H 00**  **Objectif visé : O3 - Analyser l’organisation fonctionnelle et structurelle d’un produit**  **O2 - Identifier les éléments influents du développement d’un produit**  **Compétences : CO 2.1 CO 3.1**  **Connaissance visée : SA** **1.1.2 Communication technique**  **SA 3.1.4 Post traitement et analyse des résultats**  **SA** **4.1.2 Outils de représentation schématiques**    **Matériel nécessaire :** Poste informatique équipé d’internet et du logiciel de simulation Proteus ISIS | **Les badges de contrôle d'accès - Théo Norme** |

**Objectifs de l’activité :** L'objectif de cette activité est d'analyser les informations contenues dans le cahier des charges et de s'approprier les différents diagrammes SysML afin de préparer la réalisation matérielle du projet. En travail de groupe, chaque ilot doit effectuer des recherches afin d'analyser la réponse à apporter au problème posé.

* D'identifier et caractériser les fonctions et les constituants d’un produit ainsi que ses entrées/sorties à partir de diagrammes SysML
* D'interpréter les résultats issus de la simulation afin d'analyser le comportement du produit

# Présentation du PROJET : Étude de l'objet Serrure connectée

La serrure connectée se développe de plus en plus. Si la France a pris du retard dans ce domaine comparé aux pays comme la Corée du Sud ou les États-Unis, cela reste un marché en pleine évolution. Ainsi, une étude Coda Stratégie évalue le taux actuel d’équipement des ménages français à seulement 3% (2021), mais estime cependant la progression du marché européen à 122 millions d'euros en 2025.

Côté installateurs, la dernière étude de TBC Innovation sur le marché français de la porte d’entrée va dans le même sens : le nombre de professionnels ayant déclaré avoir posé une serrure connectée a triplé entre 2019 et 2020, passant de 4% (2019) à 11% (2020)(1).

Employé dans une Startup, on vous charge d'étudier et de développer une solution de contrôle d'accès de l'habitat avec la réalisation d'une serrure connectée. Celle-ci doit permettre le verrouillage de l'accès en local par clavier, puis par la suite de l'étude à distance via une application Web.

1. *Extrait du dossier Portes d’entrée de L’Echo de la baie n°140, d’Avril 2021*

# CAHIER DES CHARGES

La définition du cahier des charges est partiellement représentée par un ensemble de diagrammes SysML donnés sur les pages suivantes. Cette première étude ne prend en compte que l'accès au local par la partie clavier.

On vous demande d'étudier dans la troisième partie de ce sujet le fonctionnement et les éléments du produit en analysant les différents diagrammes SysML.

Diagramme d’expression du besoin :



Diagramme de cas d'utilisation (service rendu par le système) :



Diagramme de définition de blocs (organisation matérielle) :



Diagramme d'exigences (besoins des parties prenantes) :



Diagramme de séquence (opérations chronologiques réalisées par un acteur) :



Diagramme de bloc interne (structure interne du système) :



# Analyse du Cahier des charges

À partir des différents diagrammes SysML du produit, répondez aux questions suivantes :

**Q1** : De combien de chiffres est composé le code d'accès entré par l'utilisateur sur le clavier ? De combien de touches est constitué le clavier ?

Le code d'accès est composé de 4 chiffres. Le clavier possède 12 touches.

……………………………………………………………………………………………………………..

**Q2** : **Comment** doit réagir le système si l'utilisateur entre le mauvais code au clavier ?

La LED ROUGE doit clignoter et un son d'erreur doit être généré.

……………………………………………………………………………………………………………..

……………………………………………………………………………………………………………..

**Q3** : Quelles sont les informations transmises par le système au technicien ?

Le niveau de tension de la batterie, l'état du verrouillage actuel et le code d'accès autorisé

……………………………………………………………………………………………………………..

……………………………………………………………………………………………………………..

**Q4** : Combien de temps l'accès est-il possible si l'utilisateur entre le bon code au clavier ?

L'utilisateur a 5 sec pour accéder au bâtiment avant que l'accès ne soit de nouveau verrouillé

……………………………………………………………………………………………………………..

**Q5** : Citez tous les éléments matériels faisant partie de la chaine de puissance du système :

Batterie et Régulateur de tension (ALIMENTER), Relais (DISTRIBUER) et Gâche (CONVERTIR)

……………………………………………………………………………………………………………..

……………………………………………………………………………………………………………..

**Q6** : Quel est le rôle du relais dans le système ?

Il est utilisé comme interface de puissance entre la tension adaptée (12V) et la commande provenant de la carte Arduino (5V) afin de piloter la gâche électrique (12V/400mA)

……………………………………………………………………………………………………………..

……………………………………………………………………………………………………………..

**Q7** : Quelle est la tension nominale de la batterie Li-ion ? De combien d'éléments est-elle alors constituée ? Identifiez alors la tension de décharge maximale de la batterie ?

**Rappel** :

|  |  |  |  |
| --- | --- | --- | --- |
| **Technologie des éléments de batteries** | **Tension nominale d'un élément** | **Tension de décharge max par élément** | **Cycles de vie / Durée de vie max** |
| Lithium ion | 3,7 V | 2,5 V | 300 à 500 cycles / 3 ans |

……………………………………………………………………………………………………………..

La tension nominale de la batterie étant de 14,8V, elle possède donc 4 éléments de 3,7V.

La tension limite est donc de 4 x 2,5V = 10 V

……………………………………………………………………………………………………………..

**Q8** : Quel est le rôle de la gâche électrique dans ce produit ? Quelle est sa consommation en courant lorsqu'elle est activée ?

La gâche permet de verrouiller ou déverrouiller l'ouverture de la porte. Elle consomme 0,4 A lorsqu'elle est activée (diagramme de définition de blocs).

……………………………………………………………………………………………………………..

……………………………………………………………………………………………………………..

**Q9** : Combien de cycle d'ouverture peut-on avoir avant décharge de la batterie ? Le cahier des charges est-il respecté ?

**Données** : La batterie a une capacité de 2600mAh. La gâche consomme 400mA. La durée de consommation de courant est de 5 sec à chaque cycle d'ouverture.

* La batterie peut fournir un courant de 2,6 A x 3600s /5sec = **1 872A** par cycle d'ouverture
* Ce qui donne : 1 872A / 0,4A = **4 680** cycles d'ouverture avant décharge complète de la batterie.

Le cahier des charges impose 3650 cycles au minimum, donc la contrainte est respectée.

……………………………………………………………………………………………………………..

……………………………………………………………………………………………………………..

……………………………………………………………………………………………………………..

……………………………………………………………………………………………………………..

……………………………………………………………………………………………………………..

**Q10** : Vérifiez que la contrainte du cahier des charges est toujours vérifiée si l'on souhaite utiliser la batterie en toute sécurité (en évitant la décharge profonde…) ?

……………………………………………………………………………………………………………..

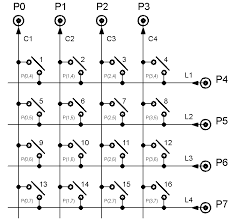
Sachant que par sécurité on ne descend pas en dessous de 20% de la charge maximale de la batterie, nous avons donc : 80 x 4 680 / 100 = **3 744** cycles d'ouverture, ce qui est toujours supérieur aux 3650 cycles imposés par le cahier des charges (soit 10 cycles par jour sur un an).

……………………………………………………………………………………………………………..

……………………………………………………………………………………………………………..

**Q11** : Complétez **les fonctions** de chaque bloc du système sur le diagramme de bloc interne de la page 4.

# Étude du fonctionnement d'un clavier

Un clavier est composé de boutons poussoirs intégrés comme une matrice de ligne et de colonnes :

Dans le système étudié, les colonnes **C2 à C0** du clavier sont **scrutées** toutes les 10ms, c'est à dire **qu'un** "**1**" logique **circule sur chaque colonne** toute les **10ms**



Lors de **l’appui d’une touche**, **une** des lignes **L3-L0** correspondant récupère le « **1** » Logique :



Le code binaire **MATRICIEL** ainsi créé par :

**L3 L2 L1 L0 C2 C1 C0**

sera alors transformé (**CODÉ**) en un **octet binaire** :

**D7 D6 D5 D4 D3 D2 D1 D0**

dont la valeur décimale représentera le **code ASCII de la touche appuyée**. Ex : Touche "3" ⇨ D7-D0 = %00110011

**♦ ETUDE PRELIMINAIRE :**

**Vérification des Prérequis** (Conversion Binaire 🡪 hexadécimale)

**Q12** : Donnez la conversion en hexadécimal des valeurs binaires suivantes **:**

**%10001111 :** $8F **% 10110110 :** $B6

**%11 0001 1100 :** $31C **%1011 :** $07

**Q13** : Complétez le tableau suivant correspondant aux différents codes matriciels du clavier en fonction de la touche appuyée

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **CODE MATRICIEL** | | | | | | | |  | **HEXA** |
| **Touches** |  | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |  |  |
| NC\* | **L3** | **L2** | **L1** | **L0** | **C2** | **C1** | **C0** |
|  | % | **0** | **0** | **1** | **0** | **0** | **0** | **1** | **0** | **$22** |
|  | % | **0** | **1** | **0** | **0** | **0** | **0** | **1** | **0** | **$42** |
|  | % | **0** | **0** | **1** | **0** | **0** | **1** | **0** | **0** | **$81** |
|  | % | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **1** | **$24** |
|  | % | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **1** | **$41** |
|  | % | **0** | **0** | **0** | **1** | **0** | **1** | **0** | **0** | **$14** |
|  | % | **0** | **1** | **0** | **0** | **0** | **1** | **0** | **0** | **$44** |
| Aucune | % | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** |  | **$04** |
| % | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** |  | **$02** |
| % | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** |  | **$01** |

*\*NC : non connecté*

**Q14** : En vous aidant du schéma de simulation ISIS "***SERRURE INTELLIGENTE.DSN***", indiquez pour chaque broche du circuit Arduino dans le tableau ci-dessous:

- le nom du signal connecté à la broche,

- le sens de transfert : en **entrée** (**IN**) OU en **sortie** (**OUT**)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Broches | 8 et 9 | 10 à 13 | A3 | A4 | A5 | 22 | 24 |
| Signal | EN – RS | LCDx | GACHE | ON | OFF | C0 | C1 |
| Sens | OUT | OUT | OUT | OUT | OUT | IN | IN |
|  |  |  |  |  |  |  |  |
| Broches | 26 | 28 | 30 | 32 | 34 | 1 | 3 |
| Signal | C2 | L0 | L1 | L2 | L3 | TXD | SOUND |
| Sens | IN | IN | IN | IN | IN | OUT | OUT |

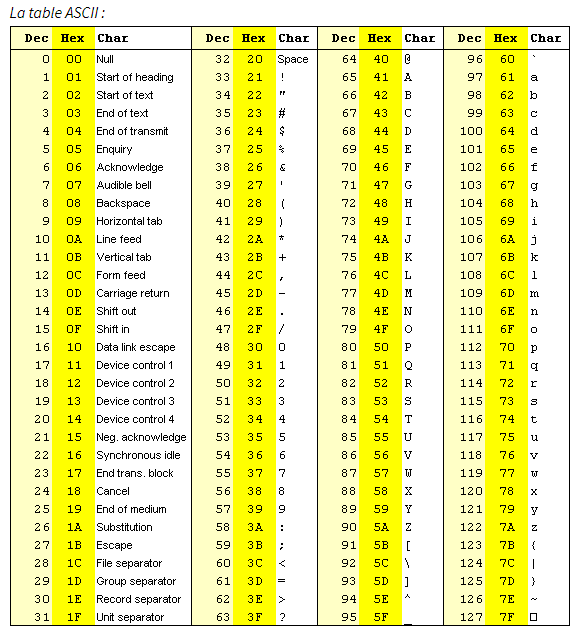
🖰 **Chargez** le programme "*clavier.ino.mega.hex*" dans le microcontrôleur ARDUINO Mega :

…\Ressources\Programmes\CLAVIER\**clavier.ino.mega.hex**

**Q15** : En vous aidant du **débogueur** du modèle de simulation ISIS, indiquez pour chaque touche enfoncée le code BIN, HEXA et DEC correspondant. En utilisant la ressource "Table ASCII", recherchez le caractère correspondant au code hexa ou décimal de la touche appuyée.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Touches** |  | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |  | **HEXA** | **DEC** | **Caractère ASCII** |
| 1 |  | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |  | $31 | 49 | 1 |
| 2 |  | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |  | $32 | 50 | 2 |
| 3 |  | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |  | $33 | 51 | 3 |
| 4 |  | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |  | $34 | 52 | 4 |
| 5 |  | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |  | $35 | 53 | 5 |
| 6 |  | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |  | $36 | 54 | 6 |
| 7 |  | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |  | $37 | 55 | 7 |
| 8 |  | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |  | $38 | 56 | 8 |
| 9 |  | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |  | $39 | 57 | 9 |
| \* |  | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |  | $2A | 42 | \* |
| 0 |  | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |  | $30 | 48 | 0 |
| # |  | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |  | $23 | 35 | # |

**Ressource** : Table ASCII



**Q16** : Que permet de faire à votre avis la librairie "Keypad" utilisée dans le programme "clavier.ino" ?

.............................................................................................................................................

Cette librairie permet de convertir le code matriciel du clavier (C3-C2-…L0) en un code ASCII correspondant à la touche appuyée.

.............................................................................................................................................

.............................................................................................................................................

**Q17** : Comment, à partir de la **valeur** **hexadécimale** de la touche appuyée nommée "key", retrouver la valeur **numérique** **correspondant** au caractère ASCII nommée "**valDEC**"?

Exemple : touche **3** -> Valeur de Key en ASCII : **$33** -> valDEC = **3**

.............................................................................................................................................

Touche **1** -> valeur de Key en Hexa = 0x3**1** -> valDEC = **1**

Touche **9** -> Hexa = 0x3**9** -> valDEC = **9**

Il faut donc retirer à chaque fois la valeur hexa 0x30 (48) à la valeur contenue dans "*key*"

Donc : valDEC = *key* – 0x30

.............................................................................................................................................

.............................................................................................................................................

.............................................................................................................................................

.............................................................................................................................................