

## Dossier ressources : Contrôle d'accès dans les établissements de santé

### 1. Découverte du produit et de la problématique technique

Les établissements de santé sont assez souvent confrontés aux vols de médicaments, aux violences à l'égard des personnels ou des patients.



Des solutions électroniques sont alors utilisées pour améliorer le contrôle d'accès. Concrètement, les clés sont remplacées par des cartes électroniques RFID.

Néanmoins, ces solutions ne prennent pas en compte les contaminations liées à l'état de santé des personnels soignants.

Une des mesures permettant de détecter les cas d'infection consiste à surveiller la température corporelle.

**Objectif de l'étude :** intégrer à une solution existante de contrôle d'accès par RFID ou magnétique, un système de contrôle de la température corporelle sans contact permettant de détecter les utilisateurs présentant une éventuelle infection.

### Principe de fonctionnement

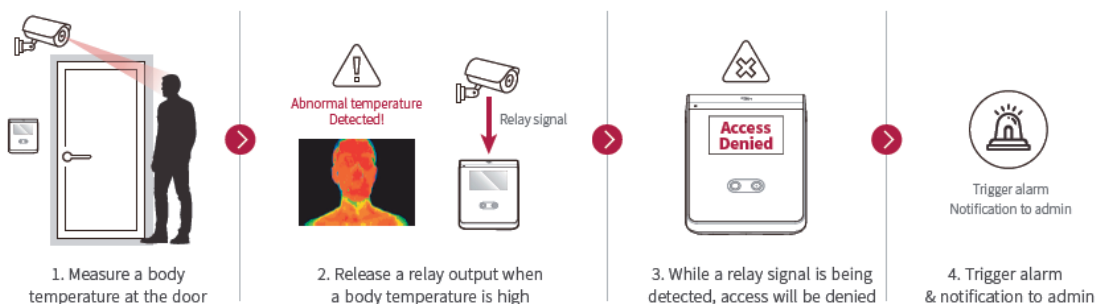
L'accès aux services sensibles (patients fragiles, salles d'opération...) doit dépendre non seulement de la validité du code de la carte RFID mais aussi de la température corporelle de l'utilisateur.

Si le capteur détecte une température corporelle supérieure à la température normale, l'accès est interdit, même si le code de la carte présentée est valide.

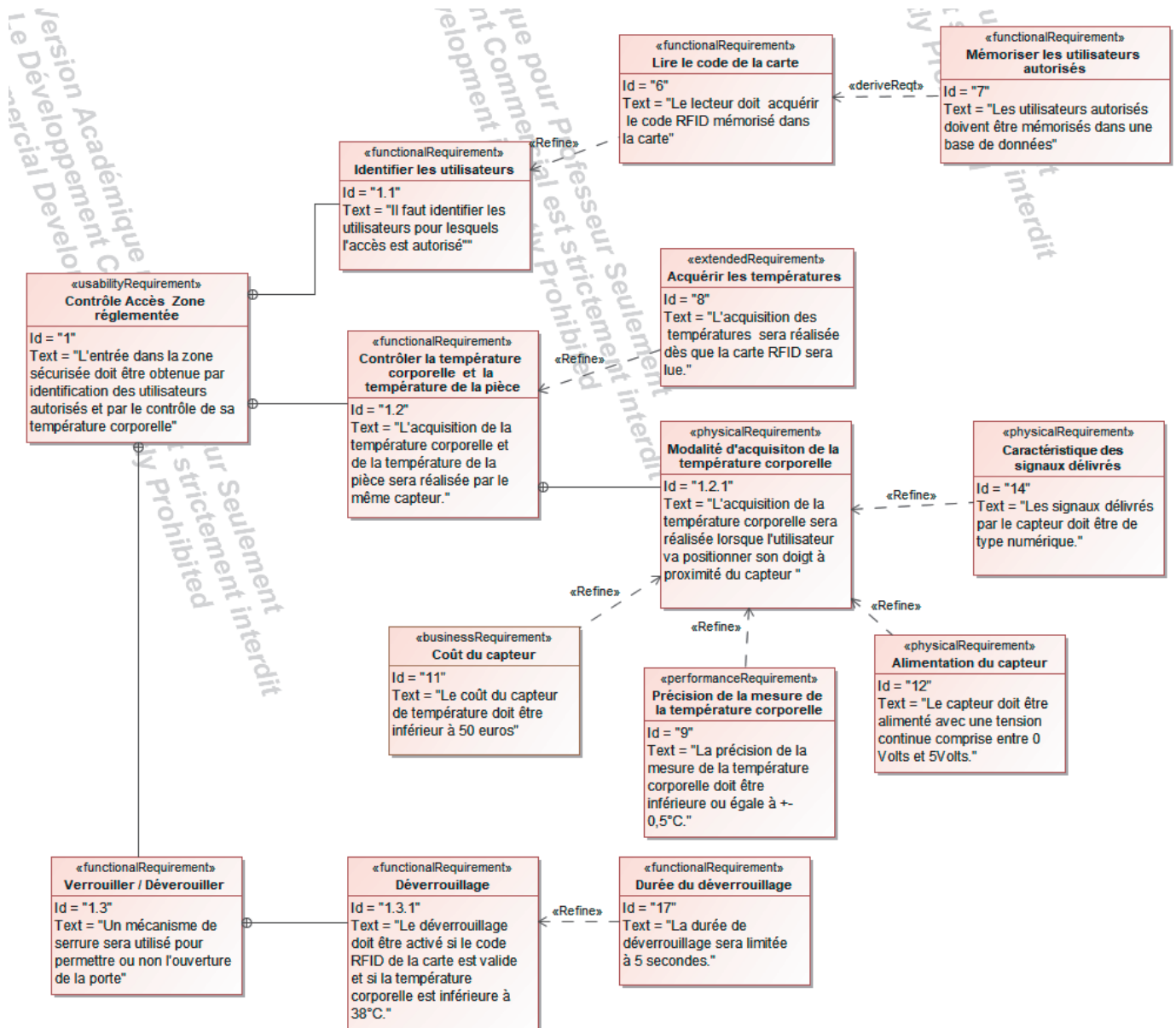
L'Organisation mondiale de la santé (OMS) définit la température corporelle normale :

- Température  $\leq 37,3$  °C, température normale
- Température de 37,4 à 37,9 °C, faible fièvre
- Température  $\geq 38$  °C, forte fièvre

#### Operation example



## a. Diagramme des exigences



## b. Fonctionnement attendu

Le code du tag RFID reçu par le capteur RFID est transmis à la carte ARDUINO MEGA qui commande une LED rouge et une LED verte ainsi qu'un servomoteur qui autorise l'accès ou non.

Si le code et la température corporelle sont validés alors, la LED verte s'allume, le servomoteur est actionné afin d'autoriser l'accès. Le verrouillage est automatique après une durée préétablie, la LED verte s'éteint.

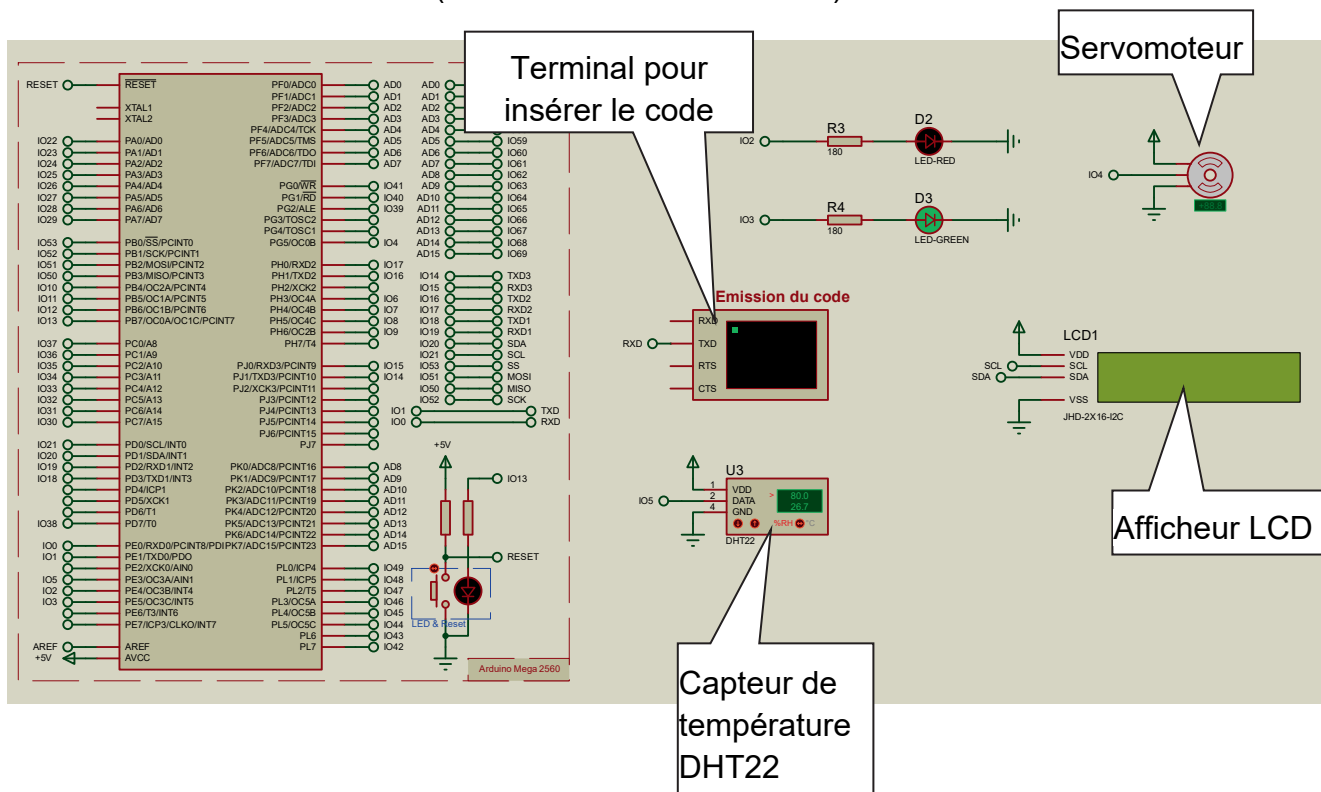
Si le code est invalide, la LED rouge s'allume quelques secondes puis s'éteint, le système de sécurisation reste verrouillé.

Un afficheur LCD indique l'état du fonctionnement.

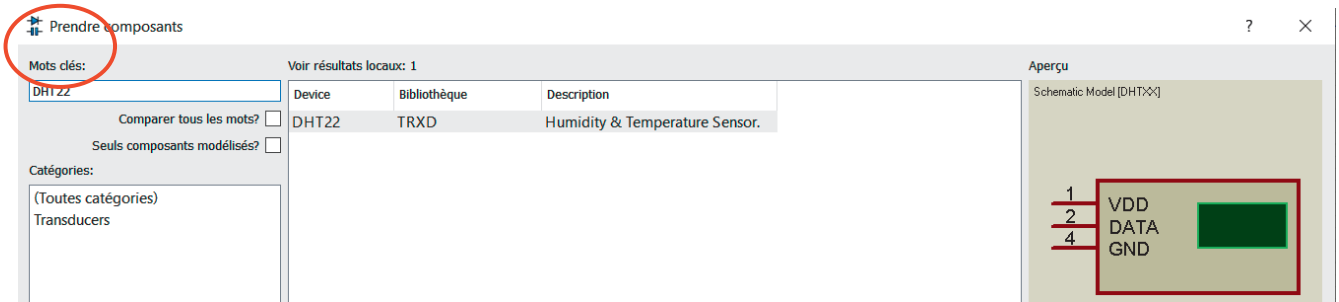
## 2. Simulation

Le capteur DHT22 est utilisé pour la simulation.

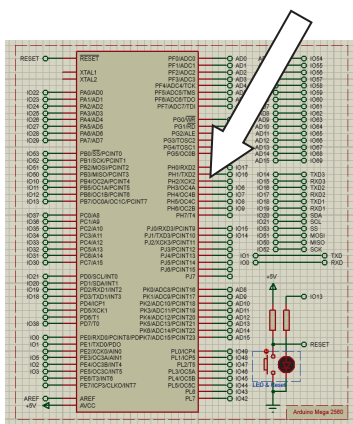
a. Schéma structurel (simulation PROTEUS 8.13) :



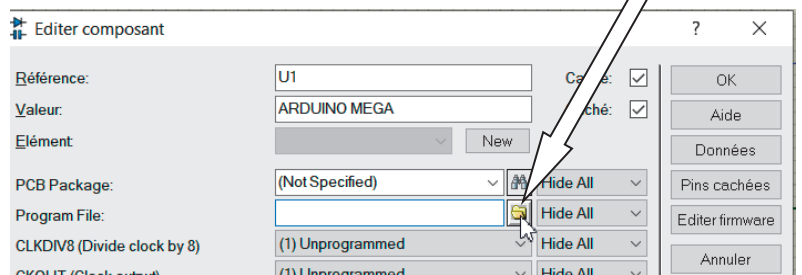
b. Réalisation de la simulation



L'intégration du code source de la carte de développement Arduino sous ISIS se fait en double-cliquant sur la carte puis en cliquant sur l'icône de sélection de fichier.



Ensuite sélectionner le fichier HEX généré sous l'IDE d'Arduino.



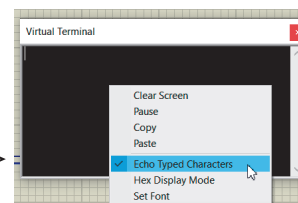
Dans le simulateur le code RFID est transmis via le terminal selon le protocole suivant :

1. Copier seulement le code valide à partir du programme ARDUINO

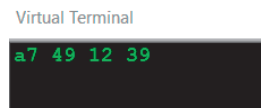
```
/* traitement du tag RFID */
if ((uid == "a7 49 12 39") && (temp
```

2. Lancer la simulation PROTEUS
3. Dans la fenêtre du terminal qui apparaît

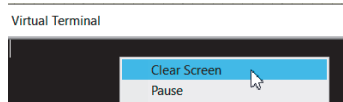
- a. La configurer (clic bouton droit souris)
- b. Coller le code précédent



- c. Résultat

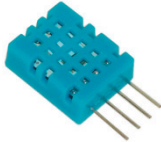





- d. Effacer l'écran



**3. Conception**

## Document 1 : capteurs de température

Désignation	Référence	Précision	Prix TTC	Alimentation	Protocole de communication
Capteur température/humidité 	DHT11	$\pm 2^{\circ}\text{C}$	2,90 €	3 à 5 V	1-Wire / Signal numérique
Convertisseur de température infrarouge 	TM 65	$\pm 0,2^{\circ}\text{C}$	107,61€	+24 V ca/cc	Signal analogique 0 V – 10 V ou 4...20 mA
Capteur de température infrarouge sans contact 	MLX90614 GY-906	$\pm 0,5^{\circ}\text{C}$	48€	2,6 V à 3,3 V	I2C
Module caméra thermique 	PIM387	$\pm 1^{\circ}\text{C}$	79,90 €	3 V à 6 V	I2C

## 4. Expérimentation

## Document 2 : opérateurs

## Comparaison d'opérateurs

Nom de l'opérateur	Opérateur	Description	Exemple
Égal	==	Vérifie si la valeur des deux opérands est égale ou non, si oui alors la condition devient vraie.	(A == B) ce n'est pas vrai
N'est pas égal	!=	Vérifie si la valeur des deux opérands est égale ou non, si les valeurs ne sont pas égales alors cela devient vrai.	(A != B) c'est vrai
Inférieur à	<	Vérifie si la valeur de l'opérande gauche est moins élevée que celle de droite, si oui alors la condition devient vraie.	(A < B) c'est vrai
Supérieur à	>	Vérifie si la valeur de l'opérande gauche est plus élevée que celle de droite, si oui alors la condition devient vraie.	(A > B) ce n'est pas vrai
Inférieur à ou égal	<=	Vérifie si la valeur de l'opérande de gauche est moins élevée ou égale que l'opérande de droite, si oui alors la condition devient vraie.	(A <= B) c'est vrai
Supérieur à ou égal	>=	Vérifie que la valeur de l'opérande de gauche est plus élevée ou égale à celle de droite, si oui alors la condition devient vraie.	(A >= B) ce n'est pas vrai

## Opérateurs booléens

Nom de l'opérateur	Opérateur	Description	Exemple
ET	&&	Appelé opérateur ET logique ou porte logique ET. Si les deux opérands ne valent pas zéro alors la condition devient vraie. Sinon, la condition est fausse.	(A && B) c'est vrai
OU		Appelé opérateur OU logique ou porte logique OU. Si chacun des deux opérands n'est pas zéro alors la condition devient vraie.	(A    B) c'est vrai
NON	!	Appelé opérateur NON logique ou porte logique NON. Permet d'inverser l'état logique d'un opérande. Si la condition de l'opérande est vraie alors l'opérateur NON logique placé devant celui-ci inversera le résultat : ce sera faux.	!(A && B) c'est faux

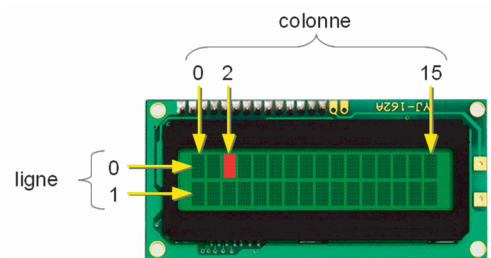
## Document 3 : utilisation de l'écran LCD

## Utiliser un écran LCD

Un **écran LCD** (Liquid Crystal Display ou écran à Cristaux Liquides en français) permet d'afficher des informations provenant d'une carte Arduino sans utiliser d'ordinateur.

Nous utiliserons un **16x2**, c'est à dire 16 colonnes par 2 lignes.

Les fonctions à utiliser :

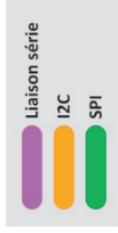


**lcd.print(variable ou "chaîne de caractère");** pour afficher une valeur ou une chaîne de caractère.

**lcd.setCursor(col, ligne);** sert à placer le curseur à un certain endroit de l'écran. Elle accepte donc deux paramètres à savoir la colonne et la ligne où le curseur doit se mettre.

Ex : `lcd.setCursor(0, 0)` place le curseur en début de la 1ère ligne.

Document 4 : schéma de câblage



# Microcontrôleur

