

Dossier ressource : serrure à reconnaissance faciale

1. Découverte du produit et de la problématique technique



Au fil des siècles, la serrurerie a connu une transformation radicale, passant de simples mécanismes métalliques à des dispositifs de haute technologie. Au départ, la sécurité reposait sur des clés traditionnelles, souvent

faciles à copier ou à déjouer, sans éliminer le risque de crochétage.



Avec l'avènement de la technologie numérique, des solutions plus sophistiquées permettent de répondre aux besoins croissants de protection des biens et des personnes.

Plusieurs solutions technologiques innovantes existent pour contrôler et faciliter l'accès à un bâtiment :

- Smartphone : l'utilisateur peut déverrouiller la porte via une application mobile utilisant Bluetooth ou Wi-Fi.
- Badge ou carte RFID : la serrure intègre un lecteur permettant l'ouverture par simple scan d'un badge ou d'une carte électromagnétique.
- Empreinte digitale : certains modèles utilisent l'authentification biométrique pour un niveau de sécurité élevé.
- Code numérique : un clavier permet l'entrée d'un code pour déverrouiller la porte.
- Reconnaissance faciale : comme pour l'empreinte digitale, elle utilise des caractéristiques physiques uniques.

Une serrure par reconnaissance faciale permet d'améliorer la sécurité, la convivialité et la gestion des accès, telles que :

- **Sécurité renforcée** : l'unicité du visage réduit considérablement le risque de duplication ou de piratage qui existe avec les clés traditionnelles.
- **Convivialité** : l'accès sans clé élimine le besoin de mémoriser des codes ou de porter des clés physiques, ce qui est particulièrement utile pour les personnes souvent distraites.
- **Gestion des accès** : la possibilité de programmer plusieurs visages autorisés facilite la gestion des entrées pour les membres de la famille ou les visiteurs temporaires (sans la nécessité d'avoir leurs empreintes).
- **Historique d'accès** : certains modèles enregistrent la date et l'heure à laquelle chaque personne accède à la propriété, augmentant ainsi la surveillance et le contrôle de l'accès.

La problématique est donc de trouver une solution pour réaliser une serrure à reconnaissance faciale à la fois fiable, économiquement accessible et simple d'utilisation, tout en assurant une accessibilité optimale même en cas de coupure d'alimentation secteur.

a. Chaîne d'information du système

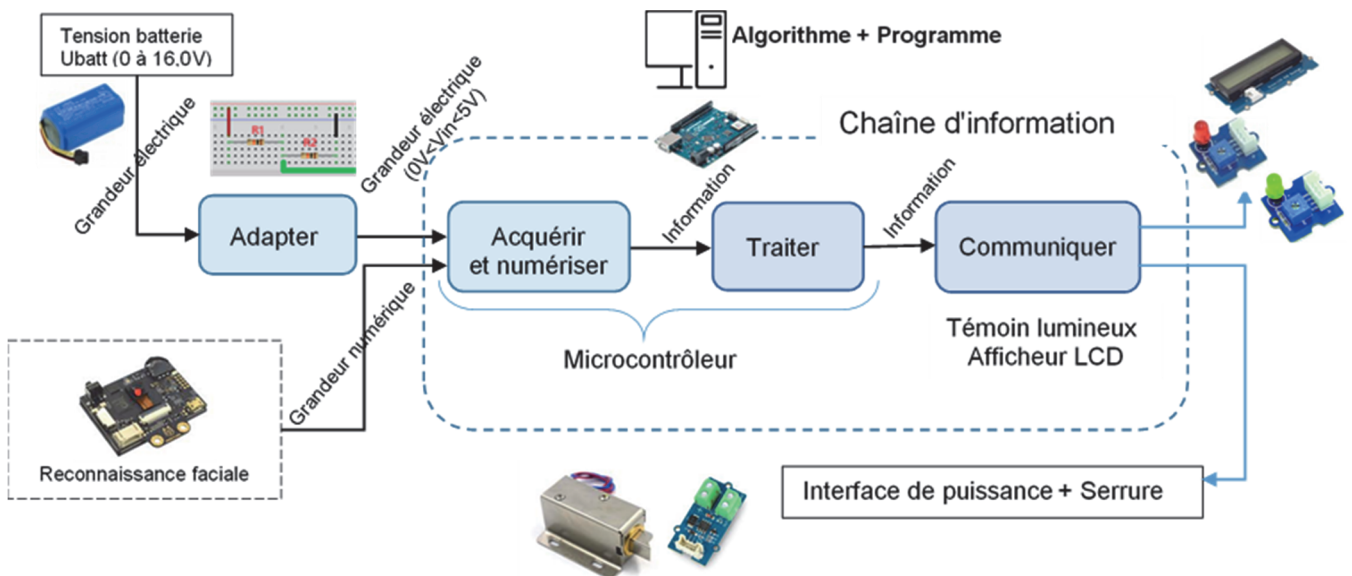


Figure 1: Chaîne d'information

b. Diagramme de blocs

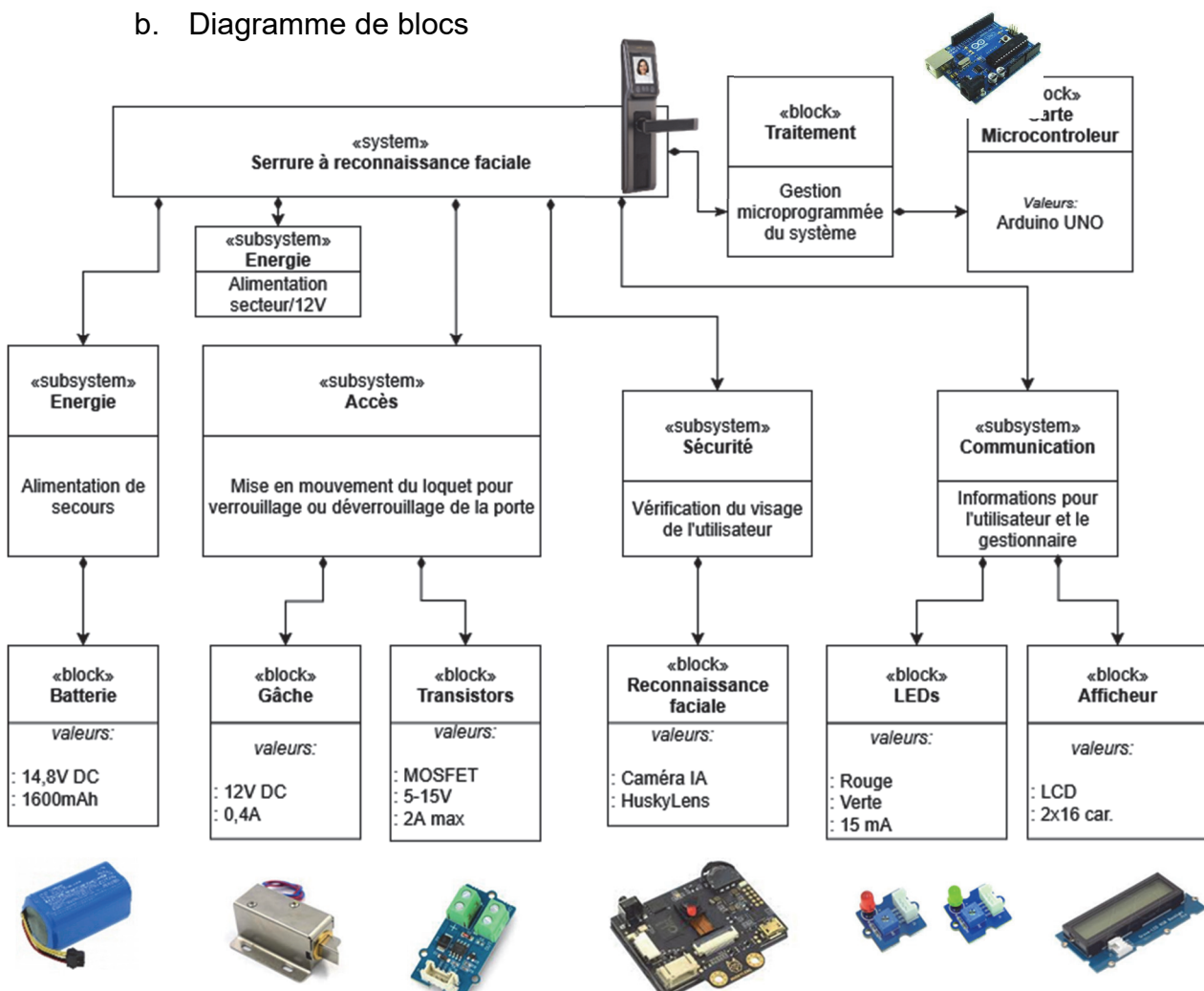


Figure 2: Diagramme de définition de bloc

c. Caractéristiques

Caméra IA HuskyLens :

- Alimentation : 3,3 V à 5,0 V
- Capteur : Omnivision OV640
- Ecran :
 - Définition : 320 × 240 pixels
- Dimension : 52 mm × 44,5 mm (2,05 mm × 1,75 mm)
- Protocoles de communication : UART ou I2C
- Fonctionnalités : reconnaissance de visage, suivi d'objets, reconnaissance d'objets, reconnaissance de couleurs, reconnaissance d'AprilTag, suivi de ligne

Serrure à solénoïde SL5520 :

- Alimentation : 12 V
- Consommation : 0,4 A
- Course : 10 mm
- Dimensions serrures : 55 × 23 × 28 mm

d. Cahier des charges

- Batterie de secours : Lithium – 14,8 V
- Résolution de l'écran minimum de 120 ppp afin d'identifier correctement le visage.
- Seuil de détection de batterie faible : 12 V
- Indice de protection de la caméra : IP 42

e. Schéma structurel partiel

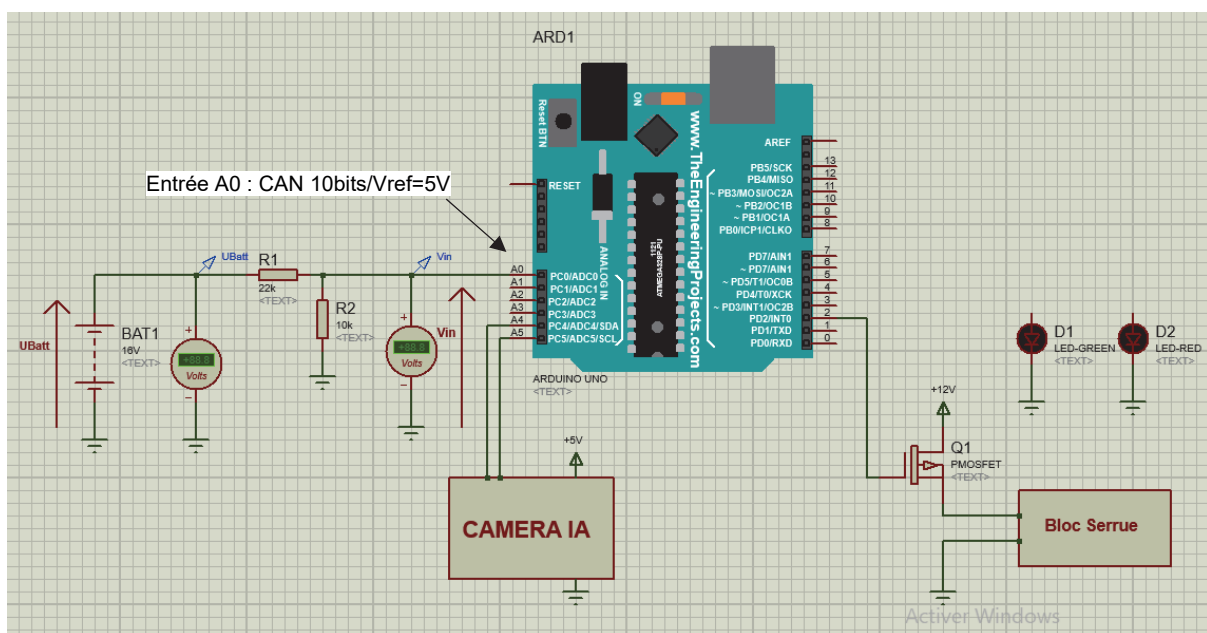


Figure 3: Schéma électronique

f. Algorithme

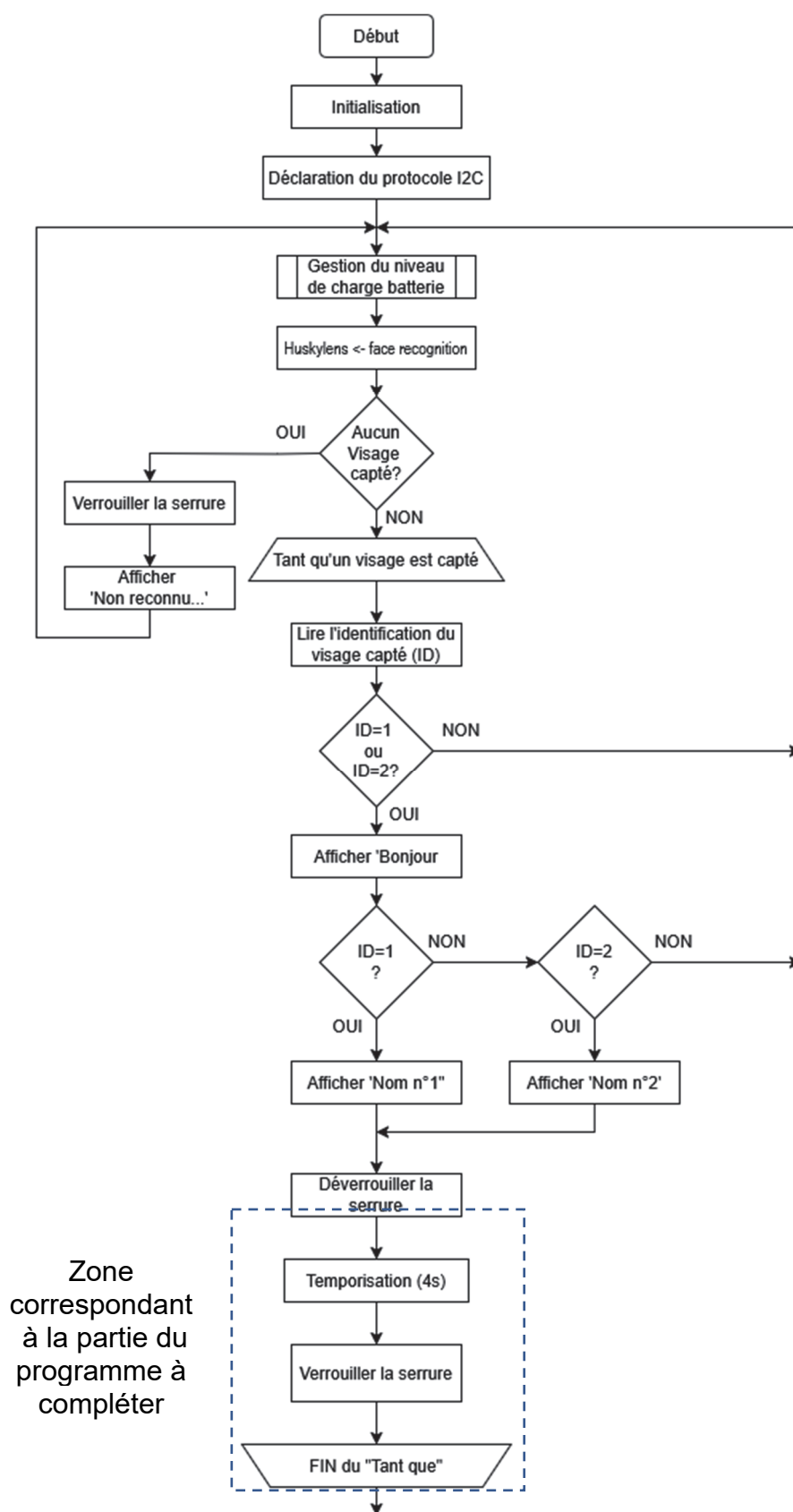


Figure 4 : Algorithme

g. Extrait du programme existant

/*Si la reconnaissance faciale est correcte (ID1 ou ID2), alors l'ouverture de la serrure est activée pendant 4 secondes. Une surveillance de l'état de charge de la batterie de secours : Batterie chargée: led verte allumée ----Batterie déchargée: led rouge allumée */

```
#define Batterie A0
#define SERRURE 2
#define LED_verte 3
#define LED_rouge 4
#include <Wire.h>      //bibliothèque pour communiquer en i2C avec l'afficheur
#include <rgb_lcd.h>    //bibliothèque de l'afficheur LCD
[...]
```

```
void setup() {
    Serial.begin(115200);
    pinMode(SERRURE, OUTPUT);
    pinMode(LED_verte, OUTPUT);
    pinMode(LED_rouge, OUTPUT);
[...]
```

```
void loop() {
    Niveau_Batterie();

    if (!huskylens.request()) Serial.println(F("Echec de la requete vers la caméra, vérifiez
la connexion!"));
[...]
```

```
while (huskylens.available())
{
    HUSKYLENSResult result = huskylens.read();
    int ID=result.ID;
    if(ID==1 || ID==2){ //Si ID1 ou ID2
        lcd.clear();      //Efface l'écran
        lcd.setCursor(0, 0);    //place le curseur en colonne0, ligne 0
        lcd.print("Bonjour");
        lcd.setCursor(0, 1);    //place le curseur en colonne0, ligne 1

        switch(ID){
            case 1: message="M. MARTIN"; Serial.println("M. MARTIN"); break;
            case 2: message="Mme FAIVRE"; Serial.println("Mme FAIVRE") ;break;
        }
        lcd.print(message);
    }
}
```

// Partie à compléter

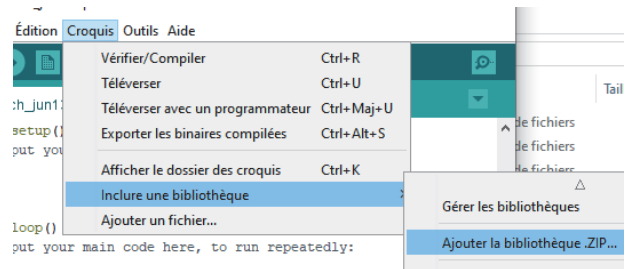
```
[...]
```

```
void Niveau_Batterie() { // détection du niveau de charge de la batterie de secours
    int N = analogRead(Batterie);
    if (N>600){ // Valeur de N à modifier pour répondre à l'exigence souhaitée.
        digitalWrite(LED_verte,HIGH);
        digitalWrite(LED_rouge,LOW);
    }
    else {
        digitalWrite(LED_verte,LOW);
        digitalWrite(LED_rouge,HIGH);
    }
}
```

2. Ajouter une bibliothèque dans l'IDE Arduino

Pour utiliser la caméra IA sous Arduino, il est nécessaire d'installer la librairie " **HUSKYLENS-master.zip**" :

- Ouvrir le programme dans l'IDE d'Arduino
- Cliquer sur "Croquis" puis sélectionner "Inclure une bibliothèque" et "Ajouter la bibliothèque .ZIP".
- Sélectionner le fichier de la bibliothèque présent dans le dossier « **Documents Expérimentation** »

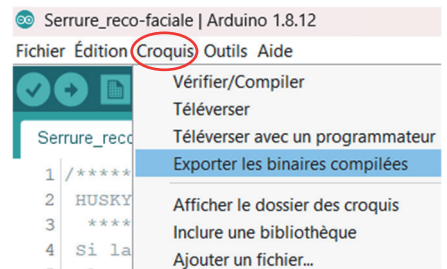



3. Procédure d'importation d'un programme Arduino dans le simulateur Proteus

Pour créer le fichier de simulation du microcontrôleur Arduino:

- Ouvrir le programme dans l'IDE d'Arduino
- Cliquer sur "Croquis" puis sélectionner "Exporter les binaires compilés".

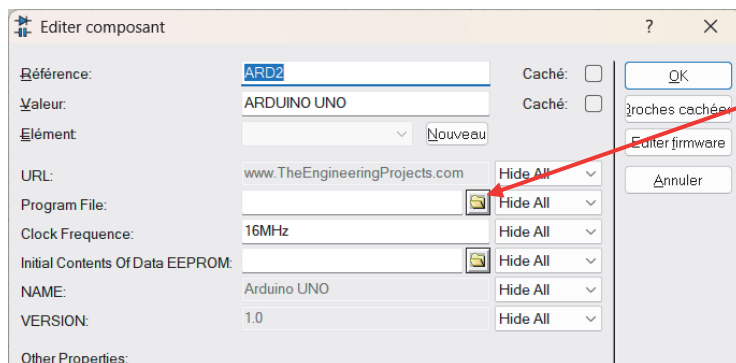
Un fichier avec l'extension **.hex** est ainsi généré dans le même dossier que le fichier programme.



Exemple:  **Simu_Niveau-Batterie.ino**

 **Simu_Niveau-Batterie.ino.standard.hex**

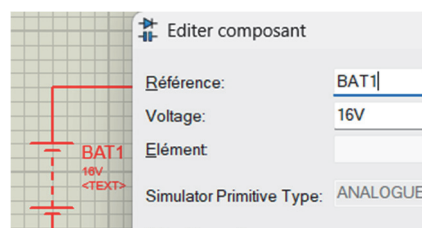
L'intégration du code source de la carte de développement Arduino sous Proteus se fait en double-cliquant sur la carte puis en cliquant sur l'icône de sélection de fichier. Il ne reste ensuite qu'à sélectionner le fichier HEX généré sous l'IDE d'Arduino.



Icone de sélection de fichier

Pour modifier un paramètre d'un composant, double-cliquer sur ce dernier puis modifier le paramètre souhaité.

Exemple:



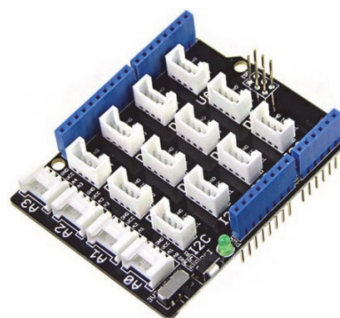
4. Expérimentation

Pour effectuer l'expérimentation, le matériel suivant est à disposition :

- une carte de développement Arduino ;
- un shield de connexion "Base" ;
- une caméra IA "HuskyLens" ;
- un module MOSFET Grove ;
- 2 LEDs Grove (Verte et Rouge) + 1 potentiomètre Grove (simulation de la tension Vin);
- 1 serrure solénoïde ;
- 1 Afficheur LCD 2x16 Grove.
- 1 pont diviseur de tension



Carte Arduino Uno R3



Shield base Grove



Caméra IA "HuskyLens"



Module MOSFET Grove



LEDs + Potentiomètre



Serrure solénoïde



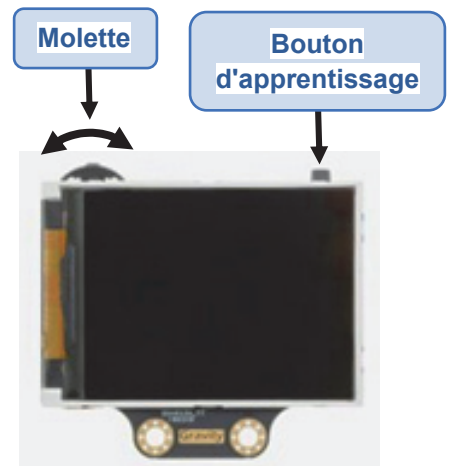
Afficheur LCD 2x16 Grove

Pont diviseur de tension

5. Configuration et apprentissage de la caméra

5.1 Phase d'initialisation

- Pivoter sur la molette de fonctions vers la droite jusqu'à ce que « **General Settings** » s'affiche.
- Sélectionner le menu en appuyant sur la molette de fonction.
- Pivoter sur la molette de fonctions vers la droite jusqu'à ce que l'option « **Factory Reset** » s'affiche.
- Appuyer sur la molette une première fois pour valider l'option « **Factory Reset** », puis une deuxième fois pour répondre « **Yes** ».



5.2 Reconnaissance multiple

- Sélectionner « **Face recognition** » avec la molette.
- Dans les options du menu « **Face recognition** », faire un appui long sur la molette pour entrer dans l'option « **Learn Multiple** ».
- Activer l'option « **Learn Multiple** » à l'aide de la molette (une barre blanche apparaît).
- A l'aide de la molette déplacer le curseur de gauche à droite (la barre devient bleue) et valider en appuyant sur la molette.
- Aller dans le menu « **Save & return** » (toujours avec la molette) et valider par « **yes** ».
- Revenir dans le menu « **Face recognition** » (si ce n'est pas déjà le cas)
- Pointer la caméra vers un visage. S'il est détecté, il est automatiquement encadré en blanc avec le mot « **Face** » sur l'écran.
- Pour apprendre un visage, pointer la croix visible au centre de l'écran à l'intérieur du cadre blanc, puis appuyez brièvement sur le bouton d'apprentissage (le bouton poussoir classique).
- Puis appuyer à nouveau pour valider avant la fin du compte à rebours. Une fenêtre « **Face : ID1** » doit apparaître.
- Répéter les étapes **g**, **h** et **i** pour l'apprentissage d'autres visages (**ID2**, ...).

