

CHRU Lille



**Centre Hospitalier Régional
Universitaire de Lille**

2 Avenue Oscar Lambret

59000 Lille

COMPTE RENDU PROJET S8

Sujet : Amélioration de la traçabilité des médicaments en milieu hospitalier



Julian BONVILLE

François ROCHE

2^{ème} année cycle ingénieur IMA

(Informatique – Micro-électronique – Automatique)

Polytech Lille



**POLYTECH'
LILLE**

TABLE DES MATIERES

I – PRESENTATION DU PROJET	2
Définition du cahier des charges initial	2
Outils utilisés	2
Evolution du projet	2
II – PRESENTATION DE L'HÔPITAL	3
III – LES DATAMATRIX	3
Définition d'un Datamatrix	3
Décoder un Datamatrix	3
Utilisations	5
Méthode de screenshot	5
Méthode avec Motion	6
Méthode avec un programme issu de la foxcam	6
IV – L'EVOLUTION DE L'APPLICATION	6
Prototype 1 : application à une webcam	6
Application	6
Présentation de l'application	6
Retour des professionnels	7
Prototype 2 : application avec un tunnel d'images	7
Application	7
Prototype 3 : intégration des bases de données dans le prototype 1	8
Application	8
Présentation de l'application	8
Retour des professionnels	9
Conclusion	9
V – L'APPLICATION FINALE	10
Prototype 4 : saisie avec une douchette	10
Application	10
Modélisation 3D de la douchette	10
Déroulement d'un scan	12
A quoi servent les websockets ?	12
Présentation de l'application	12
VI – UN PEU DE CODE	14
Explications	14
Les fonctions DMTX	15
VII – POUR ALLER PLUS LOIN	16
Dernier bilan	16
Retour des professionnels	16
Les attentes futures	16
CONCLUSION	17
REMERCIEMENTS	18
SOURCES	18

I – PRESENTATION DU PROJET

DEFINITION DU CAHIER DES CHARGES INITIAL

L'objectif de ce projet est d'améliorer la traçabilité des médicaments au CHRU de Lille.

Les médicaments qui sont livrés à la pharmacie du CHRU de Lille sont tous répertoriés lors de la livraison. Ainsi, en cas de retrait de lot, il est possible de savoir immédiatement si un des médicaments a été stocké à la pharmacie et / ou dispensé. Actuellement, les données sont entrées à la main, avec tous les risques d'erreurs que cela comporte.

Le but de ce projet est de proposer une solution permettant d'automatiser ces différentes tâches. A partir d'un datamatrix présent sur chacune des boîtes, il est possible d'extraire les données utiles (n° CIP du médicament, date de péremption, n° de lot) et de les sauvegarder dans une base de données.

Il faudra aussi écrire une application web permettant d'accéder aux données au travers d'un navigateur internet avec outil de recherche.

Lors d'une alerte de retrait de lot par l'ANSM (Agence Nationale de Sécurité du Médicament), il est possible de recevoir un courriel (avec n° CIP et n° de lot concernés). Il conviendra de traiter automatiquement ce courriel afin de savoir si le lot a été livré à la pharmacie.

OUTILS UTILISES

Logiciels

Wamp server, Notepad++, PHPmyAdmin, Apache, Emacs, Okular

Langages

C, Java, PHP, HTML, CSS, SQL

Pour l'impression 3D

Solidworks, Cura 14.12

EVOLUTION DU PROJET

Le prototype 1 a été présenté le **25 février** à Polytech à une responsable du CHRU de Lille.

Le prototype 2 n'a pas été présenté car nous l'avons rapidement abandonné.

Le prototype 3 a été présenté le **11 mars** à la pharmacie du CHRU. Lors de la présentation se trouvaient la même responsable ainsi que des employés.

Le prototype 4 a été présenté le **22 avril** au CHRU à l'ensemble des responsables du service.

II - PRESENTATION DE L'HÔPITAL

Le CHRU de Lille utilise deux logiciels. Le premier, leur base de données est interne à l'hôpital, leur permet de rentrer leurs réceptions afin de connaître leurs stocks. Le second, GEF (Gestion Economique et Financière) est un logiciel privé, qui permet la gestion financière des médicaments pour l'hôpital. Il n'a aucun lien avec l'organisation des médicaments dans le stock de l'hôpital.

L'ANSM (Agence Nationale de Sécurité du Médicament) émet des alertes lorsqu'il y a un médicament à retirer du marché. Ces demandes doivent être rapidement traitées par l'hôpital, qui doit également effectuer une procédure avec les patients déjà dispensés de ce médicament.

III – LES DATAMATRIX

DEFINITION D'UN DATAMATRIX

Le datamatrix est un code barre bidirectionnel. Il est l'évolution du code barre traditionnel. Il présente l'avantage de contenir plus d'informations. Ces informations sont concentrées sur une matrice, qui peut être décodée par des appareils équipés d'un périphérique d'acquisition d'images (webcam, appareil photo ...). Les domaines d'application sont vastes (pharmacie, publicité, suivi de matériel, logistique ...)

Dans notre projet il représente les données nécessaires à effectuer la traçabilité du médicament.

DECODER UN DATAMATRIX

Afin de décoder les datamatrix nous avons commencé par utiliser l'utilitaire « libdmtx » sur <http://www.libdmtx.org/>

Une fois l'utilitaire installé sur nos ordinateurs nous avons décidé de le tester.

Pour cela nous avons :

- Pris une photo contenant un datamatrix ;
- Tapé la commande suivante dans le fichier où se situe la photo : `dmtxread -n nom_du_fichier`
- Comparé avec le résultat avec le site suivant : <http://boy.co.ua/decode.php>

Exemple :

Soit le datamatrix suivant :



Le résultat obtenu est le suivant : 010340093592455117150900103RLC50715

Dans ces informations nous avons le CIP13 numéro correspondant au nom du médicament, la DLU soit la date limite d'utilisation, et le numéro de lot du médicament.

Les documentations trouvées sur internet nous ont permis d'identifier ces éléments. La façon de lire le datamatrix issu d'un médicament est la suivante :

Entre 01 et 17 nous avons le CIP13 : 3400935924551

Entre le 17 et le 10 on a la Date de péremption : 150900 soit le : 00/09/2015 (00 : jour non spécifié)

Puis entre le 10 et le 91 on a le N° de lot : 3RLC50715 (attention, le 91 n'est pas toujours présent)

Nous avons créé un script permettant d'obtenir le CIP13, la DLU et le N° de lot :

```
<?php
$test='010340093592455117150900103RLC50715';
$length=strlen($test);

echo $test;
echo '<br>';
for ($i = 0; $i <= 12; $i++)
{
    $cip[$i]=$test[$i+3];
}
$j=5;
for ($i = 0; $i <= 5; $i++)
{
    $date[$i]=$test[$i+18];
}
$date1[0]=$date[4];
$date1[1]=$date[5];
$date1[2]=$date[2];
$date1[3]=$date[3];
$date1[4]=$date[0];
$date1[5]=$date[1];

$date[0]=$date1[0];
$date[1]=$date1[1];
$date[2]=$date1[2];
...
```

```
...
$date[3]=$date1[3];
$date[4]=$date1[4];
$date[5]=$date1[5];

$i=25;
while($i != $length-1)
{
    $i=$i+1;
    $nblot[$i]=$test[$i];
}
$cip13=implode($cip);
$date_fin=implode($date);
$numero_lot=implode($nblot);
echo "Nom du médicament :";
echo $cip13;
echo "<br>";
echo "Date de fin :";
echo $date_fin;
echo "<br>";
echo "Numéro de lot :";
echo $numero_lot;
echo "<br>";
?>
```

Résultat obtenu par l'utilisation du script :

010340093592455117150900103RLC50715
Nom du médicament :3400935924551
Date de fin :000915
Numéro de lot :3RLC50715

Nous avons remarqué que suivant la distance du Datamatrix et la luminosité ambiante, l'application pouvait être défaillante. Il faut également que le Datamatrix soit bien parallèle et face à la webcam, sinon sa forme carrée va être perçue plutôt comme un losange, et son déchiffrement sera alors impossible.

Les webcams empruntées règlent partiellement le problème de la luminosité, étant de bien meilleure qualité que nos caméras de PC portable. Il faut toutefois éviter de se trouver à contre-jour (même résultat qu'une prise de photo).

Pour la distance, l'essai de lecture sur une dizaine de médicaments nous a permis de trouver la distance optimale qui fonctionne toujours : entre 6 et 7 cm.



UTILISATIONS

Afin de pouvoir capturer un Datamatrix, nous avons choisi d'utiliser une webcam, ayant pour avantage d'être très économique.

L'objectif est qu'à travers la caméra, on décode le Datamatrix et on envoie ces informations sur un ordinateur.



Pour lire le flux vidéo de la webcam on a utilisé différentes méthodes :

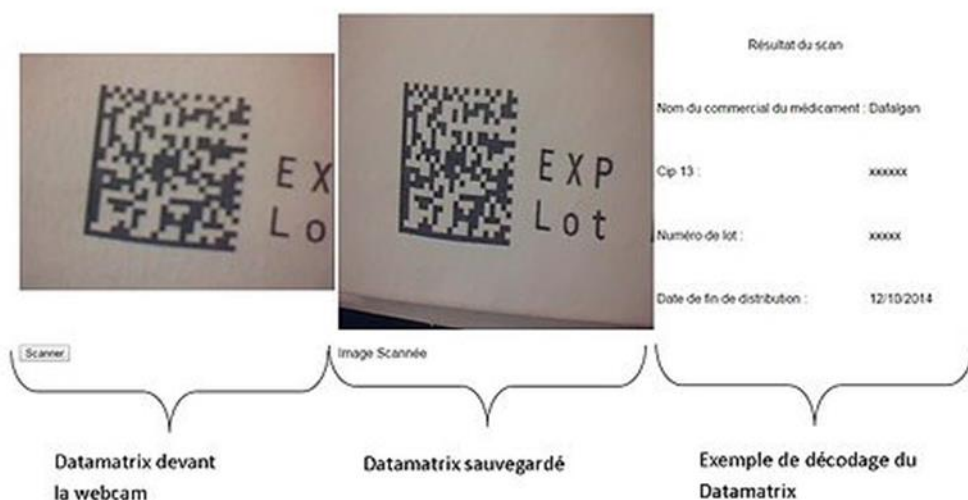
- la méthode de screenshot ;
- l'utilitaire Motion ;
- la méthode avec un programme issu de la foxcam.

Méthode de screenshot

Principe :

Quand l'utilisateur appuie sur le bouton "scanner" de notre page html, l'image se fige et s'enregistre sur l'ordinateur. Pour réaliser cela nous avons fait appel à des fonctions telles que *html2canvas* ainsi que des tutoriels internet que nous avons adapté pour l'application.

Résultats :



Méthode avec Motion

L'idée ici était de pouvoir lire le flux vidéo de 6 webcams en même temps, afin de pouvoir réaliser un convoyeur qui prendrait en photo le Datamatrix, quelle que soit la position du médicament. Cette solution a rapidement été abandonnée de par la longueur du temps d'acquisition. En effet l'application aurait été très lourde et aurait demandé beaucoup de mémoire à l'ordinateur.

Méthode avec un programme issu de la foxcam

Nous nous sommes donc tournés vers un programme en C issu de la foxcam pour pouvoir prendre une capture. Cette méthode, plus optimale, permet d'obtenir une meilleure image avec le Datamatrix de meilleure qualité.

C'est donc cette méthode qui a été retenue.



IV – L'EVOLUTION DE L'APPLICATION

PROTOTYPE 1 : APPLICATION A UNE WEBCAM

Application

Cahier des charges	Développement
Cahier des charges initial	Possibilité de réaliser un scan Capture des Datamatrix en fonctionnement Non prise en charge des bases de données

Présentation de l'application



Page d'accueil



Page d'accueil de l'application



Ajout d'un médicament

Sur cette dernière image on voit bien que si la distance requise entre la webcam et le Datamatrix n'est pas respectée, l'image n'est pas nette, et donc le décodage ne peut se faire.

Retour des professionnels

Ce premier prototype était très engageant pour eux.

Les modifications qu'ils souhaitaient sont les suivantes :

- Gérer le niveau du personnel (employé débutant - expérimenté - administrateur) ;
- Avoir une saisie manuelle (au cas où le Datamatrix soit inconnu de la base, ou pour un autre problème) ;
- Avoir une traçabilité de l'ajout effectué par le personnel (qui a fait quoi ? et quand ?)

Remarque :

Il est pour eux inutile d'afficher en continu le flux vidéo vu par la webcam sur la page de saisie. Nous avons donc convenu d'afficher une photo du médicament scanné à la place, afin de confirmer à l'utilisateur qu'il correspond à celui qu'il a scanné.

PROTOTYPE 2 : APPLICATION AVEC UN TUNNEL D'IMAGES

Application

Cahier des charges	Développement
Cahier des charges initial Rajout de plusieurs webcams pour pouvoir scanner un médicament sur un tapis (convoyeur)	Moniteur de contrôle avec motion

Nous avons constaté que l'utilitaire motion ne nous satisfait pas pour la capture du datamatrix. L'acquisition est beaucoup trop longue, et l'application serait lourde en termes de ressources.

→ Pour ces raisons, ce prototype a été abandonné.

PROTOTYPE 3 : INTEGRATION DES BASES DE DONNEES DANS LE PROTOTYPE 1

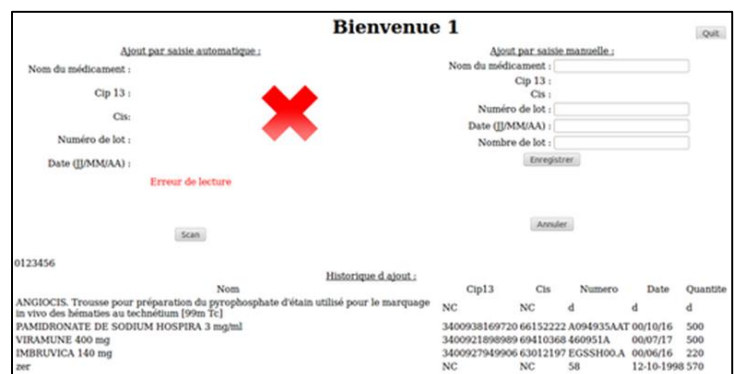
Application

Cahier des charges	Développement
<p>Cahier des charges initial</p> <p>Extraction des données du Datamatrix</p> <p>Gestion du niveau du personnel</p> <p>Demande d'une saisie manuelle</p> <p>Demande d'une traçabilité de l'ajout effectué par le personnel</p>	<p>Intégration des bases de données</p> <p>Capture des Datamatrix</p> <p>Réalisation d'ajout en mode semi-automatique</p> <p>Gestion du personnel</p> <p>Authentification unique</p> <p>Saisie manuelle</p> <p>Mise en place d'une traçabilité du personnel et des médicaments par historique</p> <p>Identification du CIS, CIP13 avec le nom du médicament</p>

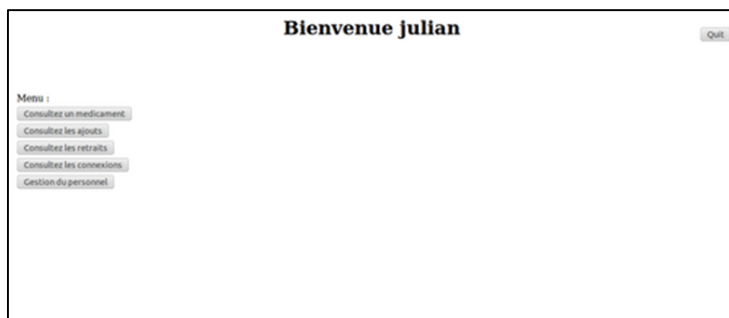
Présentation de l'application



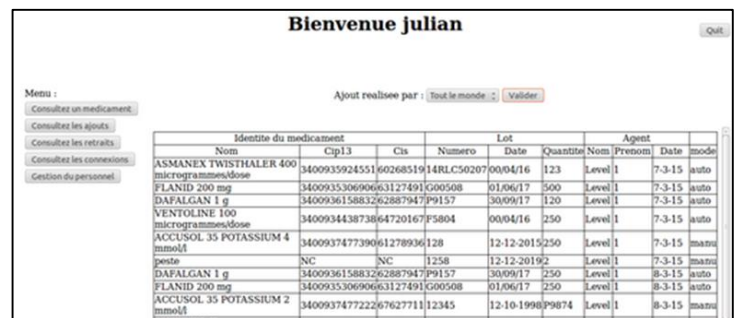
Page d'accueil



Page de connexion des opérateurs



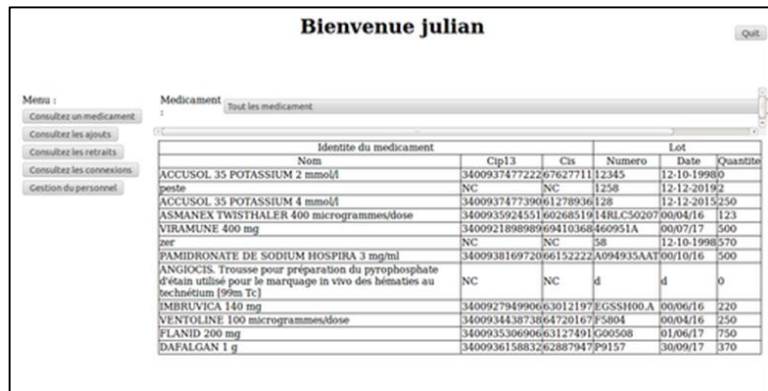
Page d'accueil de l'administrateur



Page de consultation des ajouts



Page d'accueil de l'administrateur



Page de consultation des médicaments en réserve



Page pour donner l'accès au personnel (uniquement sur un compte administrateur)

Retour des professionnels

Après la présentation du prototype 3 au CHRU de Lille, les modifications qu'ils souhaitent sont les suivantes :

- Donner la localisation prévue pour le médicament dans leur stock ;
- Imprimer sur une étiquette autocollante cette localisation, ainsi que le nom du médicament en petit (impression sur leur imprimante d'étiquettes Isiscoder PD41) ;
- Sur la page du scan, épurer au maximum : Nom, numéro et photo du médicament, date et localisation suffisent (visualiser le CIP13 et le CIS sur la page d'accueil est inutile) ;
- Faire deux niveaux de comptes sont suffisants, un compte administrateur et un compte pour chaque employé ;
- Améliorer la rapidité de l'acquisition du Datamatrix au maximum ;
- Pouvoir utiliser la webcam comme une douchette pour plus de facilité ;
- La création d'un convoyeur ne fait pas l'unanimité du personnel.

Conclusion

L'ensemble du personnel, présent pour ce premier prototype qui leur était présenté, était satisfait, mais quelques points ne leurs convenaient pas. Bilan :

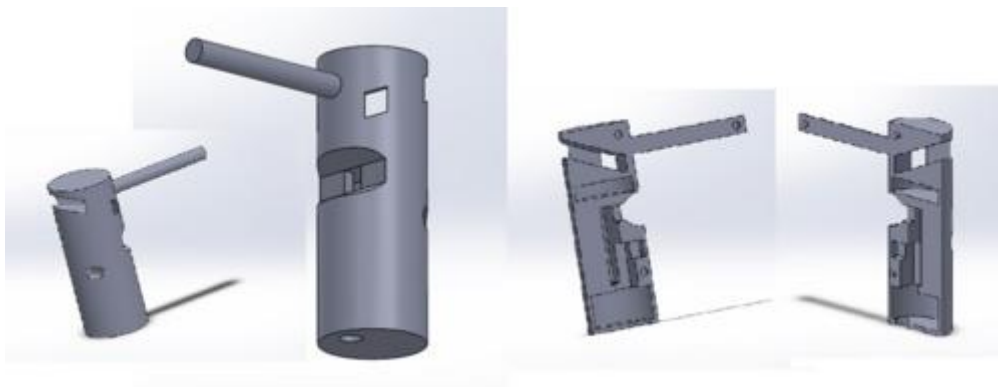
- Améliorer la rapidité de l'acquisition du Datamatrix.
- Créer une douchette avec la webcam.
- Epurer l'écran au maximum et ajouter la localisation.

PROTOTYPE 4 : SAISIE AVEC UNE DOUCHETTE

Application

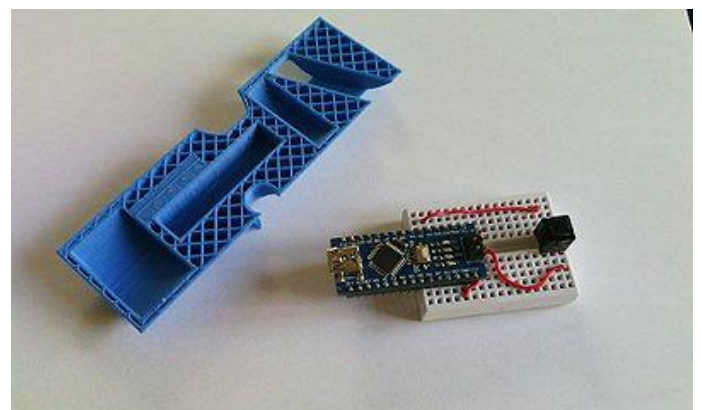
Cahier des charges	Développement
Cahier des charges initial Remplacement de l'alerte de l'ANSM par un outil de recherche Impression d'une étiquette comportant la localisation dans le stock Deux niveaux de comptes suffisent Minimiser les informations sur la page du scan Utilisation de la webcam comme une douchette	<i>L'application web est terminée. Elle communique avec le programme de lecture des Datamatrix via un serveur websocket</i>

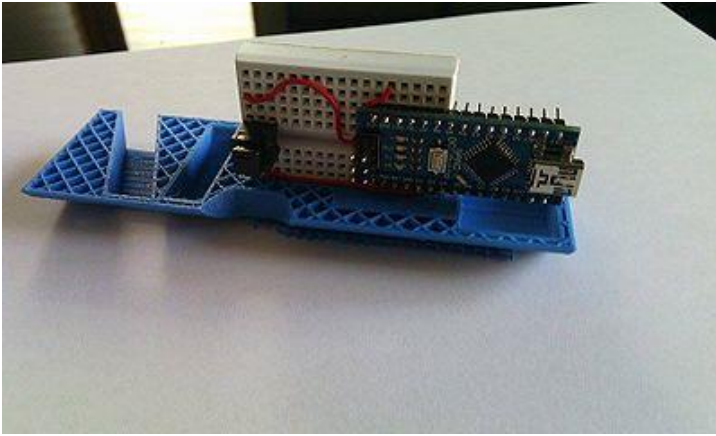
Modélisation 3D de la douchette



La webcam se place sur le dessus, et le socle dans la fente de la douchette. A l'intérieur de celle-ci se place la plaque d'essai sur laquelle se trouvent l'Arduino Nano et le bouton poussoir. Une tige a été prévue comme guide afin de respecter la distance optimale entre la webcam et le Datamatrix. Le bouton poussoir est accessible par le trou sous le guide. L'orifice sous la douchette sert à brancher l'Arduino. Enfin, les tétons permettent de fixer les deux coques ensemble, qui seront vissées dans le trou créé pour y placer une vis de serrage.

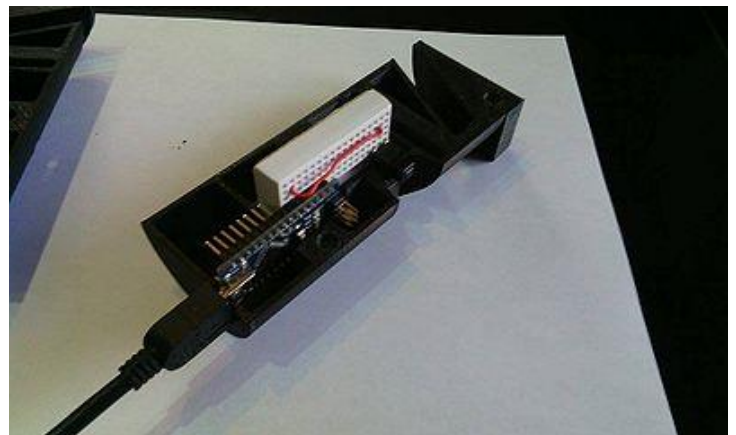
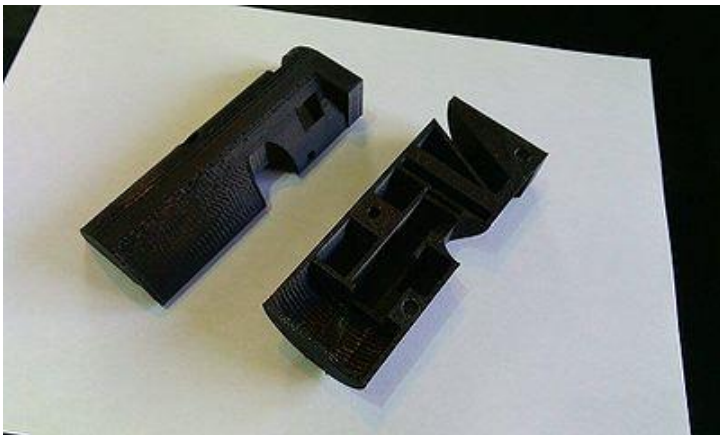
Résultat de la première impression (arrêtée à moitié de l'impression car la tige se décollait du plateau) :





On peut voir que l'Arduino rentre parfaitement dedans. Il faut cependant modifier le fichier .stl afin de supprimer la tige. Nous l'ajouterons plus tard différemment.

Résultat de l'impression du second prototype :

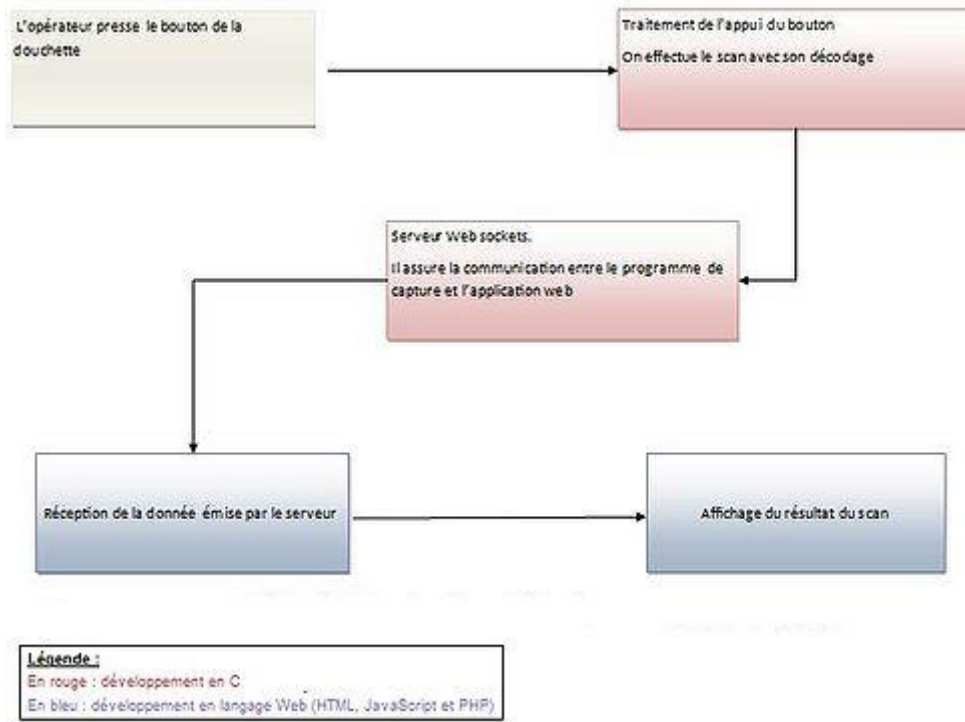


On observe que l'Arduino rentre toujours très bien dedans.

Avec la webcam et le câble USB branchés :



Déroulement d'un scan



- Le prototype fonctionne avec la douchette. Par un simple appui sur le bouton on est capable d'afficher le résultat dans une page web.

A quoi servent les websockets ?

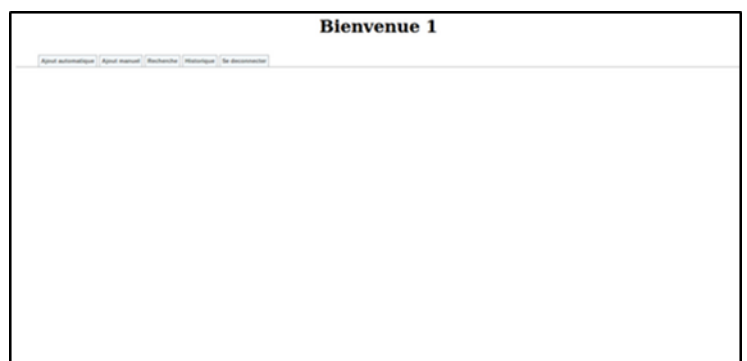
Ils permettent de faire très simplement de la communication synchrone dans une application, c'est-à-dire de la communication en temps réel. Plus imagé, cela permet de mettre en place l'équivalent d'un Chat sur le site.

Dans notre cas, l'appui sur le bouton de la douchette est directement prit en compte, et une fois le Datamatrix déchiffré, il est aussitôt affiché sur notre page web, sans avoir rechargé la page.

Présentation de l'application



Page d'accueil



Page d'accueil pour tous les opérateurs

Bienvenue 1

Ajout automatique | Ajout manuel | Recherche | Historique | Se déconnecter

Recherche :

Nom du médicament :

Numéro de lot :

Date (JJMM/AA) :

Page de recherche d'un médicament en réserve

Bienvenue francois

Ajout automatique | Ajout manuel | Recherche | Historique | Consulter les connexions | Gestion du personnel | Modifier les emplacements | Se déconnecter

Historique personnalisé Tout le monde

Level 1	12-3-15 9.41
Level 1	16-3-15 14.27
Level 1	23-3-15 13.52
Level 1	23-3-15 13.52
Level 1	23-3-15 13.56
Level 1	23-3-15 13.56
Level 1	23-3-15 13.58
Level 1	23-3-15 14.00
Level 1	23-3-15 14.04
Level 1	23-3-15 14.09
Level 1	23-3-15 14.09
Level 1	23-3-15 14.11
Level 1	23-3-15 14.11
Level 1	23-3-15 14.13
Level 1	23-3-15 14.14
Level 1	23-3-15 14.15
Level 1	23-3-15 14.16
Level 1	23-3-15 14.21
Level 1	23-3-15 14.28

Page de consultation des historiques de connexion

Bienvenue 1

Ajout automatique | Ajout manuel | Recherche | Historique | Se déconnecter

Ajout par saisie manuelle :

Nom du médicament :

Cip 13 :

Cls :

Numéro de lot :

Date (JJMM/AA) :

Nombre de lots :

Page permettant l'ajout manuel

Bienvenue 1

Ajout automatique | Ajout manuel | Recherche | Historique | Se déconnecter

Historique d'ajout :

Nom	Numero	Date	Quantite	Date d'ajout
znr	58	12-10-1998	570	25-3-15
IMBRUVICA 140 mg	EGSSH00A	00/06/16	220	11-3-15
VIRAMUNE 400 mg	460951A	00/07/17	500	11-3-15
PAMIDRONATE DE SODIUM HOSPIRA 3 mg/ml	A09495AAT	00/10/16	500	11-3-15
ANGIOKIS: Trousee pour préparation du pyrophosphate d'étain utilisé pour le marquage in vivo des métastases au technétium [99m Tc]				
ACCUSOL 35 POTASSIUM 2 mmol/l	12345	12-10-1998	P9874	8-3-15
FLANID 200 mg	G00508	01/06/17	250	8-3-15
DAFALGAN 1 g	P9157	30/09/17	250	8-3-15
paste	1258	12-12-2019	2	7-3-15
ACCUSOL 35 POTASSIUM 4 mmol/l	128	12-12-2015	250	7-3-15
VENTOLINE 100 microgrammes/dose	F5804	00/04/16	250	7-3-15
DAFALGAN 1 g	P9157	30/09/17	120	7-3-15
FLANID 200 mg	G00508	01/06/17	500	7-3-15
ASMANEX TWISTHALER 400 microgrammes/dose	14RLC50207	00/04/16	123	7-3-15

Page affichant l'historique d'ajouts des médicaments

Bienvenue julian

Ajout automatique | Ajout manuel | Recherche | Historique | Consulter les connexions | Gestion du personnel | Modifier les emplacements | Se déconnecter

Page d'accueil d'un administrateur

Bienvenue francois

Ajout automatique | Ajout manuel | Recherche | Historique | Consulter les connexions | Gestion du personnel | Modifier les emplacements | Se déconnecter

Autorisations

Utilisateur

roche francois

bonville julian

Level 1

Level 2

Operateur administrateur

Ajouter un agent

Nombre de personnes à ajouter :

Modifier un compte

roche francois

Page permettant la gestion du personnel (uniquement sur un compte administrateur)

UN PEU DE CODE

```
// Open serial port device
int serialOpen(char *device,int mode)
{
    int flags = (mode==SERIAL_READ?O_RDONLY:(mode==SERIAL_WRITE?O_WRONLY:O_RDWR));
    int fd = open(device,flags|O_NOCTTY|O_NONBLOCK);
    if (fd < 0) {perror(device); exit(-1);}
    return fd;
}
```

Pour que l'appui sur le bouton de la douchette soit pris en compte (presque) instantanément, nous avons utilisé l'option `O_NONBLOCK` lors de l'appel à la fonction d'ouverture du port série. Cette option permet une lecture non bloquante sur le port série (rafraichissement très rapide).

```
/* Fonction de lecture */
void lecture() {
    DmtxImage *img;
    DmtxDecode *dec;
    DmtxRegion *reg;
    DmtxMessage *msg;
    DmtxTime timeout;
    timeout.sec = time(NULL) + 2;
    timeout.usec = 0;
    int largeur,hauteur;
    unsigned char *image=NULL;
    valeur_datamatrix[0]='\0';
    initialisationQueue(Queue_CAPTURE,FAUX);
    envoyerMessage(Queue_COMMANDE_TYPE,COMMANDE_PHOTO,strlen(COMMANDE_PHOTO)+1);
    recevoirBinaire((char *)&largeur,sizeof(int));
    recevoirBinaire((char *)&hauteur,sizeof(int));
    if(largeur>0 && hauteur>0) {
        int taille=3*largeur*hauteur*sizeof(char);
        image=(unsigned char *)malloc(taille);
        recevoirBinaire((char *)image,taille);
        img = dmtxImageCreate(image,largeur,hauteur, DmtxPack24bppRGB);
        if(img != NULL) {
            dec = dmtxDecodeCreate(img, 1);
            if (dec != NULL) {
                reg = dmtxRegionFindNext(dec,&timeout);
                if (reg != NULL) {
                    msg = dmtxDecodeMatrixRegion(dec, reg, DmtxUndefined);
                    if (msg != NULL) {
                        strcpy(valeur_datamatrix, msg->output);
                        printf("%s \n",valeur_datamatrix); }
                    dmtxMessageDestroy(&msg);
                }
                else {printf("Datamatrix mal positionne\n");}
                dmtxRegionDestroy(&reg);
            }
            dmtxDecodeDestroy(&dec);
        }
        dmtxImageDestroy(&img);
    }
}
```

EXPLICATIONS

Sur la partie du code surlignée en jaune, la fonction lecture est le fruit d'une adaptation de deux codes issus de la communauté.

Cette partie est issue du code de la Foxcam créé par M. REDON, et la partie datamatrix est issue du site libdmtx. Elle permet de lire à la volée les images.

Afin de pouvoir agréger les deux codes il est important de bien connaître comment fonctionne les deux codes. Le code issu de M. REDON nous renvoie une image binaire. Le code issu d'internet a besoin de créer une image pour créer les régions de data matrix

On obtient alors l'image et sa taille. Une fois que l'on a ces paramètres on a juste à les intégrer dans la dernière ligne (`img = dmtxImageCreate(image,largeur,hauteur, DmtxPack24bppRGB);`).

LES FONCTIONS DMTX

dmtxDecodeCreate

Creates and initializes a new `DmtxDecode` struct, which designates the image to be scanned and initializes the scan grid pattern. This function must be called before any other scanning functions.

dmtxRegionFindNext

Searches every pixel location in a grid pattern looking for potential barcode regions. A `DmtxRegion` is returned whenever a potential barcode region is found, or if the final pixel location has been scanned. Subsequent calls to this function will resume the search where the previous call left off.

dmtxDecodeMatrixRegion

Extracts raw data from the barcode region and decodes the underlying message.

dmtxMessageDestroy

Releases memory held by a `DmtxMessage` struct. The complementary function, `dmtxMessageCreate()`, is automatically called by `dmtxDecodeMatrixRegion()` and therefore is not normally used by the calling program.

dmtxRegionDestroy

Releases memory held by a `DmtxRegion` struct. The complementary function, `dmtxRegionCreate()`, is automatically called by `dmtxRegionFindNext()` (actually `dmtxRegionScanPixel()`) and therefore is not normally used by the calling program.

dmtxDecodeDestroy

Releases memory held by a `DmtxDecode` struct. This is the complementary function to `dmtxDecodeCreate()`.

dmtxImageDestroy

Releases memory held by a `DmtxImage` struct, excluding the pixel array passed to `dmtxImageCreate()`. The calling program is responsible for releasing the pixel array memory, if required.

DERNIER BILAN

Retour des professionnels

Lors de l'entretien avec les responsables du service, nous avons pu présenter notre travail (le prototype 4 avec la douchette réalisée). Les conclusions ont été très bonnes, dans la mesure où ils ont dit que nous avons répondu au cahier des charges, en réalisant quelque chose de **fonctionnel** et de **simple à mettre en œuvre**. L'interface web leur convient parfaitement.

En ce qui concerne notre prototype de douchette, nous avons choisi une webcam et une impression 3D de la coque pour leur **faible coût**, mais il s'avère que l'achat d'une douchette n'est pas tellement un problème pour eux par rapport aux finances qu'ils ont. Pour rappel une douchette pour datamatrix coûte environ 300€, et n'en ayant besoin que d'une seule, cela reste abordable. Notre prototype leur a donc simplement permis de pouvoir constater **l'efficacité** et la **simplicité** de notre application.

Les attentes futures

Nous avons également beaucoup échangé sur leurs attentes futures. Lors de la réalisation de notre projet, nous avons traité la partie optimisation de l'acquisition des Datamatrix. Cependant, ils souhaiteraient faire la même chose avec leurs logiciels. Actuellement, ils sont obligés de rentrer deux fois les médicaments dans leurs deux logiciels : un est interne à l'hôpital, il leur permet de gérer leur stock ; et l'autre, GEF, est un logiciel privé qui permet de faire la gestion financière de leurs commandes.

Dans notre projet nous avons amélioré la partie acquisition sur leur propre logiciel en leur permettant de ne renseigner qu'une seule donnée (le nombre de lots). Il faudrait donc faire un parallèle avec leur second logiciel.

L'idée retenue est de garder en mémoire les données qui nous intéressent lors du scan, et de les renvoyer sur la fenêtre du logiciel privé par saisie sur clavier virtuel. On scanne, toute notre partie s'exécute, puis leur second logiciel attend les informations. Une partie des données préalablement scannées est renvoyée sur la fenêtre courante (qui est le second logiciel) grâce au clavier virtuel, et par succession de tabulations et d'envoi de la donnée souhaitée, les champs se remplissent un à un. Il restera à l'utilisateur à vérifier rapidement, puis appuyer sur entrée.

Cette méthode n'est certes pas très "propre", mais comme le logiciel est privé nous pensons qu'elle est la plus "simple" à mettre en œuvre. Notons que nous avons les droits de lecture sur les bases de données de ce logiciel privé.

Ils souhaiteraient également une fonctionnalité en plus, à savoir une **alerte automatique** pour indiquer les médicaments qui seront périmés dans moins de 3 mois. En effet ils doivent faire en sorte de les utiliser en priorité pour éviter les pertes. L'idéal serait de gérer le stock selon la **méthode FEFO** (First Expired, First Out).

CONCLUSION

Nous avons pu voir comment se passe une gestion de projet en milieu professionnel, en particulier le fait qu'un cahier des charges évolue sans cesse au cours d'un projet. En effet, à chaque démonstration le client peut s'imaginer plus facilement comment sera utilisé ce qu'il attend, et donc voir par lui-même ce qui pourrait poser problème lors de l'utilisation au quotidien. Une redéfinition du cahier des charges est donc à envisager à chaque présentation de l'avancée du travail.

De plus, nous avons également eu des changements à faire sans même avoir besoin de demander l'avis des clients. En effet, lorsque l'on s'aperçoit par nous même que le prototype ne pourra aboutir à une solution peu contraignante, il est inutile de persister dans la même direction (exemple avec l'utilisation de motion). Car en général, un petit inconvénient pour les développeurs en est un gros pour les clients, comme nous avons pu le constater avec la rapidité de notre système de scan automatique.

Pour conclure, ce projet S8 s'est bien déroulé, nous pensons avoir répondu correctement au cahier des charges final.

REMERCIEMENTS

Nous tenons à remercier toutes les personnes qui ont contribué, de près comme de loin, au succès de notre projet, et qui nous ont aidés à le faire évoluer.

Tout d'abord, nous tenons à remercier vivement Monsieur REDON pour le temps qu'il nous a consacré sur le développement de notre application, en particulier les problèmes liés à la partie des websockets ; ainsi que Monsieur BOE pour son accompagnement concernant la direction à prendre suite aux visites au CHRU.

Nous remercions également l'équipe de la pharmacie de l'hôpital de Lille, qui nous a apporté leur collaboration et leur expertise tout au long de ce projet. Ces visites nous ont permis de mieux cibler leurs besoins et d'orienter nos améliorations à entreprendre dans le projet.

Enfin, nous tenons à remercier les étudiants et le personnel du Fablab qui nous ont permis de réaliser notre douchette.

SOURCES

<http://www.libdmtx.org/>

<http://rex.plil.fr/>

<http://openclassrooms.com/>

<http://doc.ubuntu-fr.org/lamp>

<https://php.net/manual/fr/index.php>

<http://www.dilec.fr/document/Guide%20GS1%20DataMatrix%20ECC200.pdf>

<http://ansm.sante.fr/>

<http://boy.co.ua/decode.php>

http://libdmtx.sourceforge.com/documentation/0.7.2-1/dmtx_8h_25b68e5df52df4128907bf589598d129.html