



CasaGuard

Nom de l'apprenti/-e : Thomas Muster

N° de l'apprenti/-e : 123456

Description générale du projet d'examen :

Il s'agit d'une application Java embarquée sur un contrôleur « phidget » indépendant et permettant la surveillance d'un point d'entrée d'une maison. A l'aide d'un capteur de mouvement et d'un capteur d'intensité sonore, l'accès au point d'entrée (par ex. une porte) sera surveillé en continu. Toute détection activera la lumière et les indésirables seront pris en photo grâce à une webcam. Sans attendre, les photos prises seront automatiquement déplacées par FTP sur un serveur centralisé pour éviter leur perte en cas de dégâts au matériel de surveillance et pour consultation en cas d'intrusion.

Domaine du TPI : (Cocher la case de couleur correspondante)

<input checked="" type="checkbox"/> Développement	<input type="checkbox"/> Réseau	<input type="checkbox"/> Systèmes
---	---------------------------------	-----------------------------------

Tracé d'activité de l'apprenti/-e durant la dernière année d'apprentissage :

- Travaux sur mandats internes ou partenaires externes – gestion de projet (module 306)
- Stage professionnel en entreprise
- Module interne S50 - Développer et implanter un système de sauvegarde et de restauration dans un environnement virtuel
- Module interne A30 - Concevoir et développer des applications mobiles

Coordonnées de l'apprenti/-e :

E-Mail : thomas.muster@gmail.com
Tél : 079 111 22 33
Prénom : Thomas
Nom : Muster
Adresse : Ecole Professionnelle
Rue des Peupliers 13
1700 Fribourg

Le supérieur professionnel

Jean Dujardin

Lieu et Date

Fribourg, le 26 janvier 2024

Table des matières

1	DONNÉE DU PROBLÈME	3
1.1	INTRODUCTION	3
1.2	PROBLÉMATIQUE RENCONTRÉE	3
2	FORMULATION DU MANDAT – ÉTAT DÉSIRÉ	4
2.1	DESCRIPTION ET OBJECTIFS DU PROJET	4
2.2	EXIGENCES FONCTIONNELLES	7
2.3	ANALYSE	8
2.4	CONCEPTION	8
2.5	RÉALISATION	8
3	INFRASTRUCTURE NÉCESSAIRE	9
4	CONNAISSANCES PRÉALABLES	9
5	TRAVAUX PRÉPARATOIRES	9
6	STANDARDS D'ENTREPRISE	9
7	DOCUMENTATION OBLIGATOIRE	10
8	CRITÈRES D'ÉVALUATION	11
9	DÉLAIS	11

1 Donnée du problème

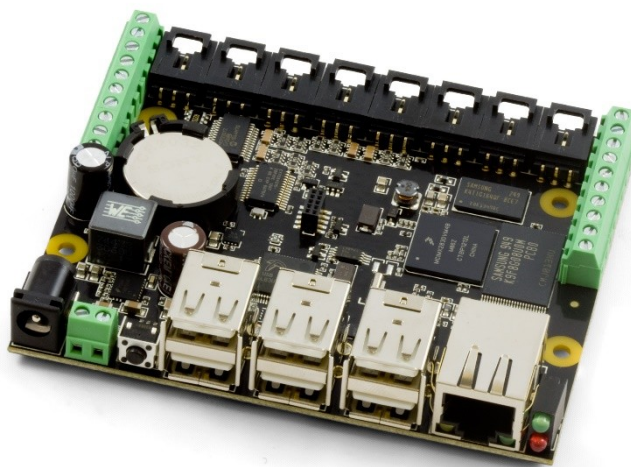
1.1 Introduction

Les phidgets (<http://www.phidgets.com>) sont des outils de prototypage très flexibles et simples d'emploi. Ils permettent de vérifier un concept, une idée, avant une éventuelle miniaturisation et production de masse.

Il s'agit de composants électroniques divers et variés (led, bouton, potentiomètre, senseur d'humidité, senseur de température, senseur de lumière, servo-moteur, accéléromètre 3D, ...) branchés à un contrôleur qui les gère.

Ce contrôleur est muni d'une interface USB, ce qui lui permet d'être branché à un ordinateur et de facilement être piloté via une librairie fournie, et ce, peu importe le langage utilisé.

Il existe également une version « indépendante » qui, à peine mise sous tension, se comporte réellement comme un PC indépendant.



Le contrôleur 1073 comporte un OS embarqué Unix/DEBIAN entièrement configurable, et ce de manière très confortable via des pages web fournies par son serveur web embarqué.

Par le même moyen, une application Java peut très facilement y être copiée/installée.

L'application peut aisément être démarrée manuellement à l'aide de SSH et, lorsque celle-ci fonctionne comme souhaité, elle peut être marquée d'un simple clic afin d'être automatiquement démarrée lors du boot de l'OS lors suite à une mise sous tension.

1.2 Problématique rencontrée

L'identification des personnes ayant illicitement pénétré dans des locaux reste bien souvent l'une des difficultés majeures lors d'une enquête. Et les solutions technologiques de surveillance sont bien fréquemment connues des concernés, souvent onéreuses et peu flexibles.



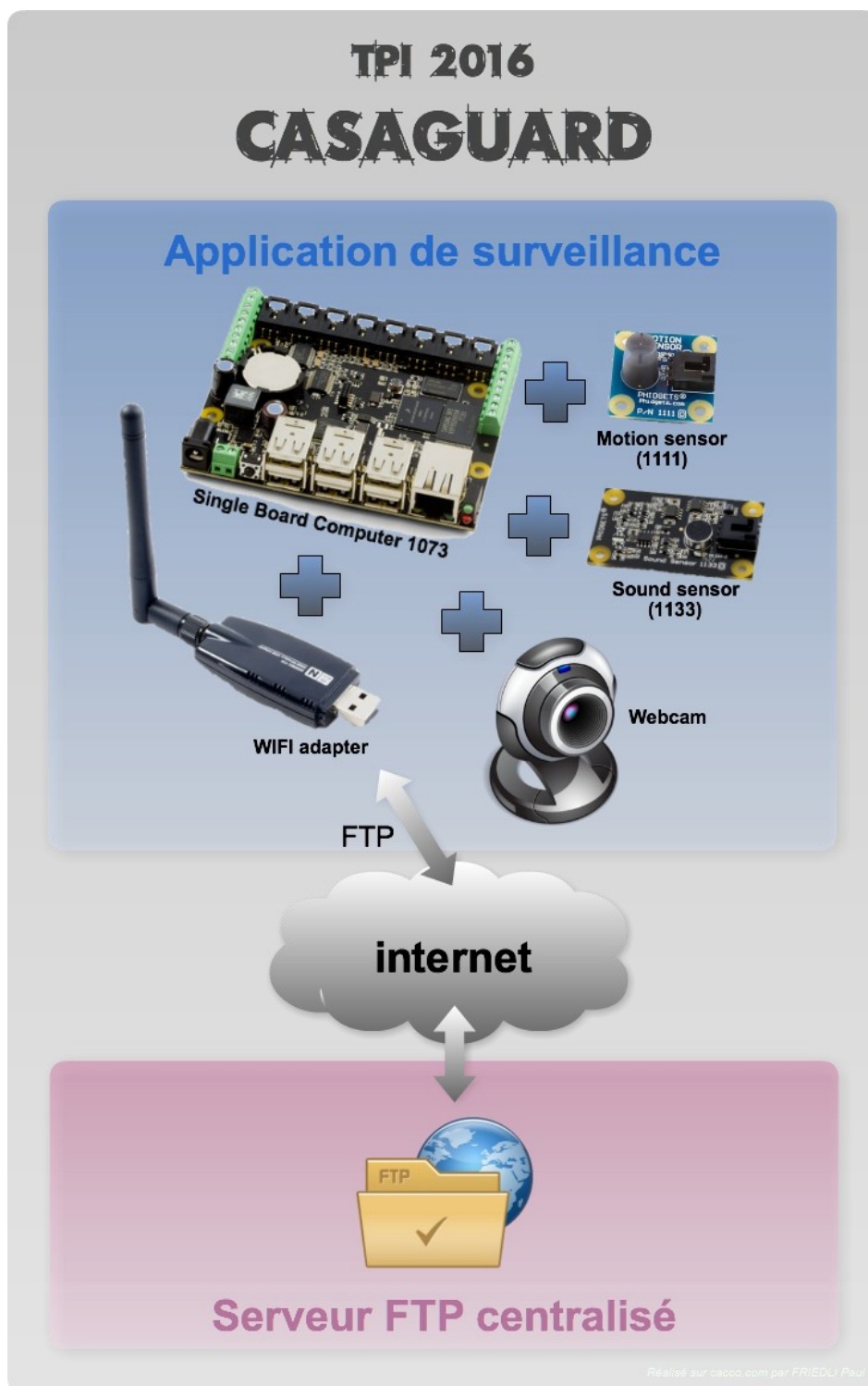
L'idée est ici de développer **un prototype** permettant de prendre « sur le fait » d'éventuels intrus de manière discrète tout en se prémunissant contre les éventuels dommages au matériel de surveillance.

2 Formulation du mandat – état désiré

2.1 Description et objectifs du projet

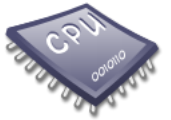
2.1.1 Schéma d'ensemble du projet

Le schéma ci-dessous donne une vue d'ensemble de la situation visée :



2.1.2 Partie applicative

Il s'agit de créer une application Java console qui pilotera le contrôleur phidgets branché aux composants utiles à ce projet.



Globalement, il faudra veiller à ce que le taux d'utilisation du CPU du SBC reste en moyenne dans des valeurs <75% (en mode **ALARM_ON**) et <25% (en mode surveillance, c.-à-d. **ALARM_OFF**).

Afin d'éviter de remplir les logs du SBC, il faudra également veiller à ce que l'affichage sur la console ne soit pas trop verbeux et qu'il soit clair, lisible et compréhensible.

L'application à réaliser se compose des 3 parties distinctes ci-dessous, opérant toutes en même temps (threads).

Surveillance

Cette partie ne se termine jamais.



Cette partie utilise les capteurs phidget de mouvement (1111) et de niveau sonore (1133) afin de détecter la présence d'une personne désirable ou non.



Les valeurs de seuil adéquates afin de permettre une bonne détection à l'aide de ces capteurs seront déterminées par l'essai et consignées dans un fichier de configuration unique.

Lorsque l'une ou l'autre valeur de seuil est dépassée, la situation **ALARM_ON** est activée. Pour que la situation repasse en mode **ALARM_OFF**, il faut que tous les capteurs retournent des valeurs sous les seuils durant plus de **ALARM_MIN_DURATION** secondes.

Identification

Cette partie dort la plupart du temps et ne se réveille qu'en cas de situation **ALARM_ON**.

Tant que la situation courante sera **ALARM_ON**, cette partie va successivement prendre des images de la webcam et les sauvegarder dans le dossier de l'application.

Après un temps configurable de **ALARM_LIGHT_DELAY_SEC** secondes, la lumière sera allumée. Placée à côté de la webcam, la lumière va attirer l'attention et augmenter les chances de capturer les visages des personnes présentes. Une valeur négative de **ALARM_LIGHT_DELAY_SEC** désactivera cette option.



Les images prises à l'aide de la webcam seront toujours nommées de la manière suivante :

Security event 2015.12.21@18:27:04.232.jpg

Pour simplifier la capture d'image avec la webcam sans avoir à utiliser opencv, elles seront directement téléchargées par http depuis le site web du SBC (<http://localhost:81/?action=stream>) qui les affiche en continu dans sa page web de configuration de la webcam.

Une fois que la situation courante ne sera plus **ALARM_ON**, la lumière sera éteinte avant de se remettre à dormir en attente d'une nouvelle situation **ALARM_ON**.

Sauvegarde



Cette partie va dormir la plupart du temps. Elle ne se réveillera qu'en cas de situation **ALARM_ON** ou toute les **FLUSH_FTPUPLOAD_EVERY_S** secondes.

Cette partie va commencer par dormir **ALARM_FTPUPLOAD_DELAY_MS** millisecondes avant de faire quoi que ce soit. Cela va augmenter les chances d'avoir une ou plusieurs images à traiter pour l'événement sécuritaire courant.

Cette partie va ensuite scanner le dossier de l'application à la recherche d'événements de sécurité (c'est-à-dire de photos prises par la partie « identification »).

S'il y a des événements de sécurité, cette partie se connectera au serveur FTP afin de toutes les uploader (les informations de connexion au serveur FTP **FTP_SERVER_ADDR**, **FTP_SERVER_USER**, **FTP_SERVER_PWD** et **FTP_SERVER_FOLDER** auront déjà été lues du fichier de configuration au démarrage de l'application). Les événements de sécurité correctement uploadés sur le serveur FTP seront supprimés localement afin qu'il y ait toujours de la place en suffisance sur le SBC.



2.1.3 Partie serveur FTP



Le but est de pouvoir démontrer la sauvegarde des événements de sécurité sur un serveur FTP distant.

L'apprenti créera un compte FTP sur nos serveurs www.emf-informatique.ch (haute disponibilité et accès dans et hors école).

2.2 Exigences fonctionnelles

Exigence fonctionnelle	Description	Priorité Haute Moyenne Basse	Temps estimé (jours)
Démarrage et planification	Lecture minutieuse de tous les documents Planification complète du projet selon modèle fourni Initiation du journal de travail selon modèle fourni	Haute	0.5 jour
Tests technologiques	Programme Java basique avec les quatre threads Surveillance, Identification, Sauvegarde et Purge, ne faisant que des <code>sout()</code> sur la console pour montrer ce qu'ils font. A l'aide d'un bouton, simulation de <code>ALARM_ON</code> dans Surveillance pour vérifier la bonne synchronisation/réveil de tous les threads.	Haute	0.5 jour
	Programme Java console qui utilise une librairie FTP afin de se connecter, s'identifier, changer de dossier, uploader un fichier, lister les fichiers présents, se déconnecter d'un serveur FTP.	Haute	0.5 jour
Analyse	Documentation d'analyse Production des diagrammes UML adéquats (diagramme des cas d'utilisation, diagramme d'activités, etc.) pour soutenir la documentation d'analyse	Haute	1 jour
Conception	Documentation de conception Production des diagrammes UML adéquats (de classes, de séquence, d'états, d'interactions, de navigation, etc.) pour soutenir la documentation de conception	Haute	1 jour
Implémentation	Implémentation de l'application Java Phidgets	Haute	4 jours
	Javadoc de qualité pour les workers, contrôleur et interfaces	Haute	0.5 jour
	Documentation d'implémentation	Haute	1 jour
Tests	Préparation d'un scénario de test Test complet de l'application selon le cahier des charges	Haute	0.5 jour
Documentation	Finalisation de la documentation projet	Haute	0.5 jour

2.3 Analyse

Cette phase de travail permet d'analyser les besoins tels qu'ils ont été décrits dans la donnée du problème, en les représentant par des diagrammes ou schémas adéquats.

Durant la phase d'analyse, l'apprenti/-e accomplit les tâches suivantes :

- Produire une planification complète du projet selon modèle fourni (c.f. §7)
- Initier un journal de travail selon modèle fourni (c.f. §7)
- Produire les représentations UML adéquates pour soutenir l'analyse
- Produire la documentation d'analyse (c.f. §7)

2.4 Conception

Durant la phase de conception, l'apprenti/-e accomplit les tâches suivantes :

- Conception de l'architecture des classes de l'application respectueuse du modèle MVC2
- Tous les diagrammes UML adéquats (diagramme des cas d'utilisation, diagramme d'activités, de classes, de séquences, d'états, d'interactions, de navigation de l'application, etc.)
- Développer l'interface homme-machine des deux applications
- Documentation de réalisation (c.f. §7)
- Tester le fonctionnement complet de l'application selon le cahier des charges
- Préparer une démonstration de qualité

2.5 Réalisation

Durant la phase de réalisation, l'apprenti/-e accomplit les tâches suivantes :

- Conception de l'architecture des classes de l'application respectueuse du modèle MVC2
- Tous les diagrammes UML adéquats (diagramme des cas d'utilisation, diagramme d'activités, de classes, de séquences, d'états, d'interactions, de navigation de l'application, ...)
- Développer l'interface homme-machine des deux applications
- Documentation de réalisation (cf. §7)
- Tester le fonctionnement complet de l'application selon le cahier des charges
- Préparer une démonstration de qualité

3 Infrastructure nécessaire

Pour la réalisation du projet, l'apprenti/-e disposera de l'infrastructure suivante :

- 1 ordinateur (à installer) avec :
 - MS-Windows 7 + Connection à Internet + Navigateur + MS-Office
 - Netbeans 8+
 - La librairie Java phidgets ainsi que l'ensemble des phidgets mentionnés
- 1 module « phidget single board 1073 »
- 1 senseur « phidget sound 1111 »
- 1 senseur « phidget motion 1113 »
- 1 adaptateur wifi
- 1 webcam Logitech

4 Connaissances préalables

Toutes les connaissances nécessaires à ce TPI ont été acquises dans le cadre des modules de sa formation.

5 Travaux préparatoires

Avant le début du projet et avec l'aide de son supérieur professionnel, l'apprenti aura préalablement :

- Pris connaissance du toolkit de base Java fourni avec les phidgets (<http://www.phidgets.com/>) et de sa philosophie de fonctionnement
- Pris connaissance des mesures de précaution utiles à garantir le bon fonctionnement de ces composants (électricité statique)
- Pris connaissance des capteurs phidgets N°1111 et N°1133, de la webcam, ainsi que du contrôleur 1018 et du SBC 1073
- Installé tous les packages Debian nécessaires et activé le support Java.
- Configuré la Webcam et le WIFI.
- Créé un compte FTP sur le serveur www.emf-informatique.ch de l'école.
- Testé une communication en FTP avec un serveur à l'aide d'une librairie java gratuite (par ex. [ftp4j](#))

Le module phidget, les senseurs, l'adaptateur wifi et la caméra web seront mis à disposition de l'apprenti pour cette phase préparatoire.

6 Standards d'entreprise

- UML comme langage de modélisation
- Java comme langage de programmation
- Architecture Pattern MVC2

7 Documentation obligatoire

A la fin du projet, l'apprenti/-e doit fournir les documents suivants :

Une planification

Cette planification doit être réalisée au début du projet avant toute autre action (selon modèle fourni). Elle décrit les étapes importantes du projet ainsi que la durée estimée correspondante. Elle doit être validée par le supérieur professionnel.

Un journal de travail

Ce document décrit les diverses étapes et activités liées au projet (selon modèle fourni).

Une documentation d'analyse

Ce document détermine les exigences et contraintes du projet et permet la justification des choix pour la réalisation du travail demandé. Ce document est composé de :

1. Résumé :
 - a) Synthèse de la documentation du mandant
 - b) Choix définitifs avec le mandant
2. Analyse UML avec les diagrammes qui seront nécessaires (cas d'utilisations, diagrammes d'activité, etc.)

Une documentation de réalisation

La documentation de réalisation a pour objectif de faciliter la maintenance et doit contenir les informations suivantes :

1. Conception :
 - a) Le ou les diagrammes de classes
 - b) Les diagrammes d'interactions
 - c) Le diagramme de navigation
2. Implémentation :
 - a) Code source/images/schémas/etc. des parties clé de l'application accompagné d'une description/explication des choix réalisés
 - b) L'intégralité du code source dûment commenté (projet NetBeans séparé)
 - c) Tests unitaires (si demandés) et fonctionnels de l'application
3. Remarques :
 - a) Problèmes rencontrés
 - b) Limites de la version
 - c) Améliorations possibles
4. Conclusions :
 - a) Commentaires personnels
 - b) Auto-évaluation
5. Résumé du rapport du TPI (voir critère PKOrg N°14)

Le Web Summary

Ce document a pour objectif de présenter le projet de manière succincte.

La remise de la documentation

La documentation est remise selon les instructions du manuel ICT-FR , partie C, point 1.9.1 « Périmètre du rapport, Tips ».

Le manuel est disponible dans PkOrg sous : Documents ➔ Documents pour tous :

<http://manuel.ict-fr.ch>

8 Critères d'évaluation

<http://criteres.ict-fr.ch>

Compétences professionnelles globales (selon la grille d'appréciation)

Résultat et efficience

1. **Solution / Produit** (selon la grille d'appréciation)
2. **Critères spécifiques au projet**
 - Analyse du problème (programmation) (159)
 - Conception - Architecture du logiciel (162)
 - Implémentation de solutions (programmation) (165)
 - Projet avec UML (235)
 - Satisfaction utilisateur : GUI, utilisation (146)
 - Respect des standards (ou standards d'entreprise) (227)
 - MVC programmation (249)
3. **Documentation** (selon la grille d'appréciation)
4. **Journal de travail** (selon la grille d'appréciation)

Présentation et entretien professionnel (selon la grille d'appréciation)

9 Délais

Le projet y compris la remise du projet aux experts se terminera le 7 juin 2024 à 17 heures. Le dernier délai pour la présentation du projet est fixé au 19 juin 2024 à 12:00.

La documentation est déposée sur PkOrg dans les temps.