COPEVUE

Cahier des charges
Système embarqué

Pierre-Yves David (H4213)

CDCSEv1.1 — 26 avril 2009 (LIVRABLE)

Table des matières

1 Introduction

1.1 Rappel du contexte

Il existe aujourd'hui de nombreux sites isolés et/ou difficiles d'accès qui nécessitent une surveillance et parfois des actions à distance. Ces sites se situent dans des espaces très différents tels que les citernes placées dans les forêts escarpées du pourtour méditerranéen, les réservoirs utilisés pour l'autonomie des chantiers dans le grand Nord mais aussi les personnes âgées qui se retrouvent souvent isolées.

Actuellement tous les contrôles et actions sont réalisés par un opérateur qui doit se déplacer sur le site. Il n'y a donc que très peu de réactivité, on ne peut effectuer un suivi fin des évolutions et des problèmes graves – par exemple la fuite d'un réservoir – ne peuvent pas être traités rapidement.

Étude COPEVUE L'objet de l'étude est la mise en place d'un système générique de surveillance et d'action à distance sur des sites isolés. Le système devra être évolutif, autonome et fiable.

1.2 Présentation du document

Ce document présente le cahier des charges du système embarqué au niveau des sites distants.

Ce système va assurer la surveillance et la maintenance de base des sites distants. Il sera également en communication avec les autres acteurs du système – système central, poste de gestion, intervenant – pour leur transmettre la valeur de ses capteurs et leur permettre de déclencher des opérations de maintenance complexe.

Dans le cadre du logiciel à développer, le cahier des charges sert de base à la rédaction des clauses contractuelles techniques, de qualité et de réception, à partir desquelles le

réalisateur proposera les spécifications fonctionnelles et non fonctionnelles du futur logiciel. Ce sont les besoins logiciels.

Nous exprimerons ces besoins en termes d'obligation de résultats, pas d'exigence de moyens. Ce document situe l'importance des fonctions du produit à développer pour l'application destinée au système de l'intervenant et donne leurs critères d'appréciation.

1.3 Documents applicables et de référence

Documents applicables

- Dossier de gestion de la documentation
- Dossier de spécification technique des besoins
- Dossier de faisabilité

Documents de référence

- Plan de référence d'un cahier des charges

2 Présentation des besoins du logiciel

2.1 Objectifs du logiciel

Le logiciel que nous développons doit s'assurer de la gestion courante et gérer les pannes simples des site distants. Il permet aussi une manipulation directe du système par les autres acteurs du système.

2.1.1 Traitement des affaires courantes

La première fonction du système embarqué consiste à s'éveiller à intervalles réguliers pour récupérer les valeurs de son ou ses capteurs et mettre à jour les positions de son ou ses potentiels actionneurs. Toutes ces tâches simples permettront au site distant d'avoir un fonctionnement beaucoup plus autonome et adapté que lorsqu'un intervenant devait effectuer ces opérations lui-même.

2.1.2 Gestion des imprévus

La détection des anomalies fait aussi partie du rôle du système embarqué. Les anomalies les moins graves devront être gérées automatiquement tandis que les plus problématiques seront simplement signalées à un intervenant humain *via* le traitement du système central.

2.1.3 Communication avec l'extérieur

Le système embarqué a besoin d'interagir avec les autres acteurs du système. Pour cela, il doit être capable de gérer plusieurs connectiques et protocoles différents pour faire face au maximum de situations possibles.

2.1.4 Action extérieure

Les autres agents du système doivent pouvoir intervenir précisément sur le système afin de récupérer des valeurs de capteurs, contrôler les actionneurs ou modifier la configuration du système.

2.2 Légèreté

Le système embarqué doit être le plus léger possible, que ce soit en puissance de calcul ou en empreinte mémoire. Ainsi, on peut l'utiliser sur la configuration la plus modeste possible afin de limiter la consommation d'énergie.

2.3 Autonomie

Vu la difficulté d'accès des sites distants, il est primordial que le système embarqué soit le plus autonome possible et qu'il soit capable de gérer le maximum de situations sans intervention humaine. Les événements environnementaux et les erreurs logicielles devront avoir le moins de conséquences possibles sur le reste du système. Lorsqu'une intervention humaine est nécessaire, on s'efforcera de permettre sa réalisation à distance.

3 Exigences fonctionnelles

3.1 Lecture des capteurs sur réseau CAN

Le système doit pouvoir récupérer les valeurs des différents capteurs qui sont connectés.

3.2 Contrôle des actionneurs sur réseau CAN

Le système doit pouvoir modifier la position des différents actionneurs qui sont connectés.

3.3 Automatismes courants

Le système doit pouvoir décider des positions de certains actionneurs en fonction de la valeur de certains capteurs ou d'éléments temporels.

3.4 Support de la veille

Le système doit être capable de se mettre en veille à la fin de chaque phase de routine et de se réveiller dans un délai assez court de manière régulière où à la demande.

3.5 Détection des anomalies

Des variations anormales des valeurs de capteur doivent être définies et détectées par le système embarqué, lorsque la résolution est possible sans intervention humaine. L'arrêt imprévu de certains systèmes doit aussi être détecté et signalé.

3.6 Gestion des anomalies

Les protocoles de gestion des anomalies simples doivent être définis et appliqués.

3.7 Gestion d'une connexion GPRS

Le système doit être capable de piloter une carte GPRS afin de se connecter au réseau de communication global.

3.8 Gestion du positionnement GPS

Le système doit être capable de piloter une carte GPS afin d'acquérir sa position. Même si dans le cas de la Norvège ce module n'est pas nécessaire, il doit néanmoins être prévu dans le système embarqué.

3.9 Gestion d'une connexion USB

Le système doit être capable de piloter une carte USB afin de se connecter au système de l'intervenant en cas de problèmes avec la connexion principale.

3.10 Transmission de données

Le système doit être capable de gérer les protocoles utilisés pour la communication avec les autres acteurs *via* les connectiques principales et de secours – détaillées ci-dessus. Le système doit être capable d'initier une connexion TCP/IP et d'envoyer des messages au serveur *via* le protocole SMTP.

3.11 Serveur SSH

Le système embarqué doit offrir la possibilité de se connecter *via* le protocole SSH aux autres acteurs du système général.

3.12 Outils de configuration

Une fois cette connexion établie, les acteurs du système doivent disposer d'outils permettant de modifier la configuration du système et de définir les règles de gestion d'anomalies et d'automatismes courants.

3.13 Outils de consultation et contrôle

Cette même connexion doit permettre de récupérer les valeurs des capteurs lues par le système et de contrôler les actionneurs du site distant.

4 Exigences non fonctionnelles

4.1 Légèreté

Le système embarqué doit être le plus léger possible, que ce soit en puissance de calcul ou en empreinte mémoire. Ainsi, on peut l'utiliser sur la configuration la plus modeste possible afin de limiter la consommation d'énergie.

4.2 Fiabilité

Les bibliothèques et logiciels utilisés devront être sélectionnés selon leur stabilité afin de garantir une grande fiabilité en fonctionnement. Les différentes sections du système devront être suffisamment isolées les unes des autres pour que la défaillance de l'une d'elles perturbe au minimum le fonctionnement des autres.

4.2.1 Modularité

Afin de réaliser les deux points ci-dessus, le système devra être le plus modulaire possible afin de garantir :

- La possibilité de supprimer toute section de code inutile pour un site distant – localisation GPRS, contrôle d'actionneurs pour les sites uniquement munis de capteurs, etc.
- L'indépendance des différents modules et donc la limitation des conséquences d'une défaillance d'un module.
- L'ajout simple de nouvelles fonctionnalités.

4.3 Performances

Le système n'a pas de contraintes de performances particulièrement rigides. La grande quantité d'entrées/sorties effectuée permet de se passer d'un ordonnancement préemptif.

4.3.1 Portabilité

Afin de pouvoir aisément choisir le matériel le plus économe en énergie par rapport au besoins de puissance réelle de l'application, on s'efforcera d'avoir une application portable sur le maximum d'architectures de microprocesseurs embarqués.

4.4 Sécurité

Les données devront être cryptées pour garantir la confidentialité des données – ce point peut sembler peu utile pour la Norvège mais prend tout son sens pour les informations médicales concernant des personnes âgées. De plus, on veillera à fournir des systèmes d'authentification fiables lors de toute communication grâce à l'algorithme RSA.

4.5 Normalisation

Le choix de bibliothèques et de logiciels respectant scrupuleusement les spécifications des protocoles utilisés est un élément-clef de la fiabilité des communications. On se référera particulièrement à :

RFC 793 pour TCP

RFC 821 pour SMTP

RFC 4251 pour SSH

5 Analyse des exigences

5.1 Critères d'analyse

Chacune des exigences fonctionnelles et non fonctionnelles est analysée et évaluée selon deux critères : sa nécessité et sa difficulté d'implémentation. Il lui est attribué deux notes sur 10, 0 signifiant par exemple la faisablité la plus basse et 10 l'importance la plus haute.

5.2 Analyse des exigences fonctionnelles

Fonction	Nécessité	Difficulté d'implémentation
Lecture des capteurs	9	3
Contrôle des actionneurs	9	3
Automatismes courants	6	8
Détection des anomalies	9	5
Gestion des anomalies	6	8
Gestion d'une connexion GPRS	7	6
Gestion du positionnement GPS	5	8
Gestion d'une connexion USB	7	7
Transmission de données	9	4
Serveur SSH	8	5
Outils de configuration	6	7
Outils de consultation et contrôle	8	2

FIGURE 1 – Grille d'analyse des exigences fonctionnelles

5.3 Analyse des exigences non fonctionnelles

Fonction	Nécessité	Difficulté d'implémentation
Légèreté	8	6
Fiabilité	8	5
Modularité	6	4
Performances	3	6
Portabilité	5	8
Sécurité	8	6
Normalisation	7	6

FIGURE 2 – Grille d'analyse des exigences non fonctionnelles

6 Mise en œuvre

6.1 Faisabilité

À la vue des analyses des exigences fonctionnelles et non fonctionnelles, en prenant en compte l'existant logiciel satisfaisant ou se rapprochant de nos contraintes de contexte, la faisabilité du logiciel pilotant le système de l'intervenant est approuvée.

6.2 Configuration cible

6.2.1 Matériel

Le système embarqué est destiné à une configuration modeste composée idéalement d'un processeur ARM à faible fréquence monté sur une carte mère capable de gérer l'énergie de ses composants en veille et se réveiller régulièrement, sur demande ou après un coupure de courant. La mémoire morte sera assurée par une disque à mémoire *flash* de faible

capacité pour limiter les risques de défaillance. La mémoire vive ne devrait pas dépasser quelque dizaines de mégaoctets.

6.2.2 Connectique

Les sites distants disposeront des connectiques suivantes :

une connectique USB pour une liaison filaire de secours avec les intervenants

une connectique GPS pour suivre la position du système si celui-ci est mobile – ce module est absent du système norvégien

une connectique GSM/GPRS pour la connectique sans fil directe avec le serveur central

6.3 Planning

Le planning de développement de notre application est donné ci-après. Il est prévisionnel et ne prend pas en compte les phases de déploiement ni celles de formation des techniciens. Ce planning est donné à titre indicatif dans un souci d'intégration au système global, il n'est pas à prendre comme contrainte de développement – notamment dans le cas d'une réponse à une appel d'offre.

Phase	Durée prévisionnelle
Développement des modules	5 mois
Intégration des modules et test en simulation	5 mois
Tests sur matériel réel et retour d'expériences	4 mois
Finalisation, éventuellement reprise de code posant des erreurs ou modification logiciel dans un souci d'améliorer l'ergonomie, le retour d'expérience nous indiquant les lacunes à combler	2 mois
Total	16 mois

FIGURE 3 – Planning prévisionnel de livraison du logiciel à destination du système embarqué