

Tournesol

Revue finale

Étudiant IR 1 → Rémy GONZALVES
Étudiant IR 2 → Adrien THIEL
Étudiant IR 3 → Nicolas DETILLEUX
Étudiant IR 4 → Grégory MOLL

Version 1.0



Sommaire

I-) Partie commune	4
I) Cahier des charges.....	4
II) Objectifs.....	7
A) Architecture matérielle du système.....	7
B) Contraintes techniques.....	7
C) Contraintes économiques.....	7
III) Mission du système.....	8
IV) Configuration d'exploitation.....	9
V) Diagrammes.....	10
VI) Description des fonctions.....	12
VII) Exigences.....	20
VIII) Le développement	20
IX) La documentation à produire	22
X) La gestion des modifications	22
II-) Contrat individuel Étudiant 1 (Gonzalves Rémy) : Module de gestion du suiveur solaire.....	23
I) Objectifs de l'étudiant 1.....	23
II) Recette minimale de l'étudiant 1 :.....	23
III) Matériel mis en oeuvre.....	25
IV) Diagramme interne de blocs.....	27
V) Fonctionnalités en charge.....	28
VI) Tâches à réaliser.....	29
VII) Tests unitaires.....	30
VIII) Diagrammes.....	31
IX) Décodage de la trame d'information du suiveur solaire.....	32
X) Planification :.....	33
III) Contrat individuel étudiant n°2 (Adrien Thiel) : Module de contrôle du régulateur de charge..	35
I) Objectifs :.....	35
II) Fonctionnalités en charge.....	36
III) Production attendue.....	36
IV) Recette minimale étudiant 2.....	37
V) Travail personnel de l'étudiant 2.....	37
VI) Tâches à réaliser.....	38
VII) Planification.....	40
VIII) Matériel.....	40
IX) Communication avec le Phocos-CX20 :.....	45
X) Partie logicielle :.....	50
XI) Test de unitaires :.....	56
XII) Manuel d'utilisation :.....	58
IV-) Contrat individuel étudiant n°3 (Nicolas Detilleux) : Module de gestion de la station météo...61	
I) Objectifs :.....	61
II) Fonctionnalités en charge :.....	61
III) Production attendue :.....	62
IV) Recette minimale étudiant 3 :.....	62
V) Présentation du travail de l'étudiant 3 :.....	62
VI) Tâches à réaliser :.....	63

VII) Planification :.....	65
VIII) Matériel :.....	66
IX) Diagramme interne de bloc :	69
X) Explications système :.....	71
XI) Diagramme de séquence :.....	73
XII) Partie IHM :.....	75
XIII) Tests de validation :.....	76
V-) Contrat individuel étudiant n°4 (Grégory Moll) : Module de contrôle de positionnement.....	78
I) Objectifs :	78
II) Fonctionnalités en charge de l'étudiant 4 :	78
III) Production attendue :.....	79
IV) Recette minimale étudiant 4 :.....	79
V) Tâches à réaliser par l'étudiant 4 :.....	82
VI) Planification :.....	84
VII) Ressource matérielle pour l'étudiant 4 :	84
IX) Diagramme IBD étudiant 4 :.....	86
XII) Suivi solaire :.....	93
XIII) Diagramme de classes :	96
XIV) Diagrammes de séquences :.....	98
XV) Test unitaires :	100
XVI) Annexe :	101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

I-) Partie commune

I) Cahier des charges

Présentation du projet :

On a besoin d'une alimentation électrique fiable pour alimenter les équipements nécessaires mais la présence d'un réseau électrique fonctionnel n'est pas toujours assurée dans les situations suivantes :

- installations isolées hors-réseau (phares, hôpitaux de campagne, ...)
- événements organisés en extérieur

Si aucun réseau n'est disponible, un générateur serait le choix logique pour générer de l'électricité. Cependant, les générateurs sont bruyants et très polluants. De plus, les directives européennes et nationales encouragent fortement la création d'unités locales de production d'énergies renouvelables car la dépendance des pays industrialisés à l'égard des combustibles fossiles, et des pays qui les produisent, induit aujourd'hui de véritables enjeux économiques et politiques.

Le solaire fait parti de ces énergies remises récemment au goût du jour. En effet, la théorie liée au photovoltaïque est ancienne puisqu'elle a été décrite par Hertz et Einstein au début du XXème siècle. L'énergie solaire est considérée comme inépuisable (compte tenu de la durée de vie du Soleil) et n'émet potentiellement aucune particule nocive (fabrication des panneaux pour la capter et des batteries pour la stocker mise à part). Elle est de plus disponible en grande quantité sur Terre. Mais le solaire est un secteur qui reste onéreux : environ 680 euros le mètre carré de panneau (posé avec onduleur et accessoires) pour une qualité moyenne (silicium amorphe ou monocristallin), ce qui le défavorise face à la concurrence d'autres énergies.

Outre les progrès faits dans la synthèse du silicium nécessaire aux cellules solaires, l'efficacité des panneaux solaires est limitée par les méthodes de fabrication. Ainsi, le rendement électrique maximal est de 15% pour des panneaux conventionnels.

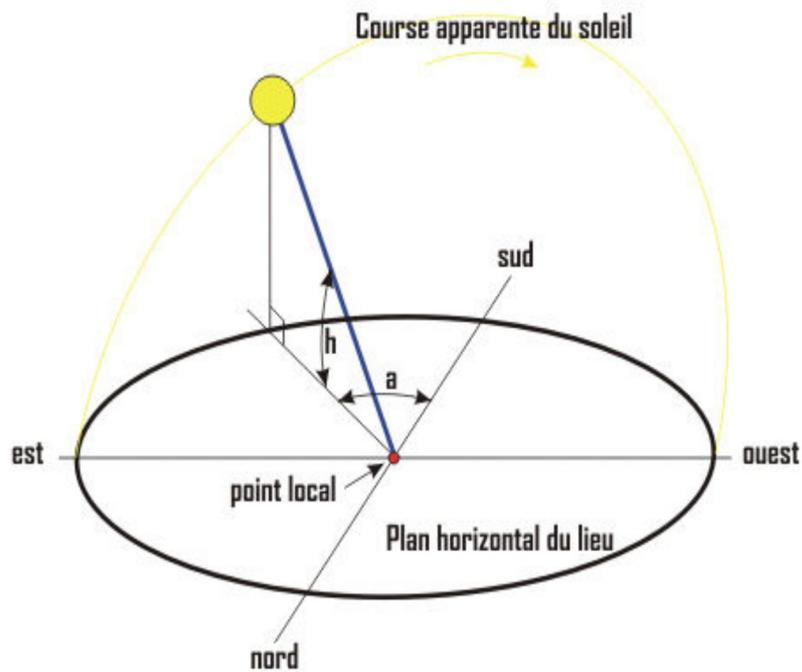
Il est possible d'atteindre 25% de rendement électrique mais avec des panneaux très onéreux, utilisés dans le domaine spatial.

Il ressort l'importance d'améliorer le rendement des installations solaires, pour exploiter au mieux les ressources potentielles. Celui-ci pourrait être augmenté de deux manières : la première consisterait à améliorer techniquement la cellule photovoltaïque, la seconde à optimiser l'angle d'éclairage du panneau en fonction de la position du soleil.

En retenant la seconde solution, il s'agira désormais d'appliquer le principe de l'héliotropisme **1** pour les panneaux solaires afin d'augmenter la production journalière d'électricité.

Tournesol	Version 2.1
Auteur : groupe	4/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

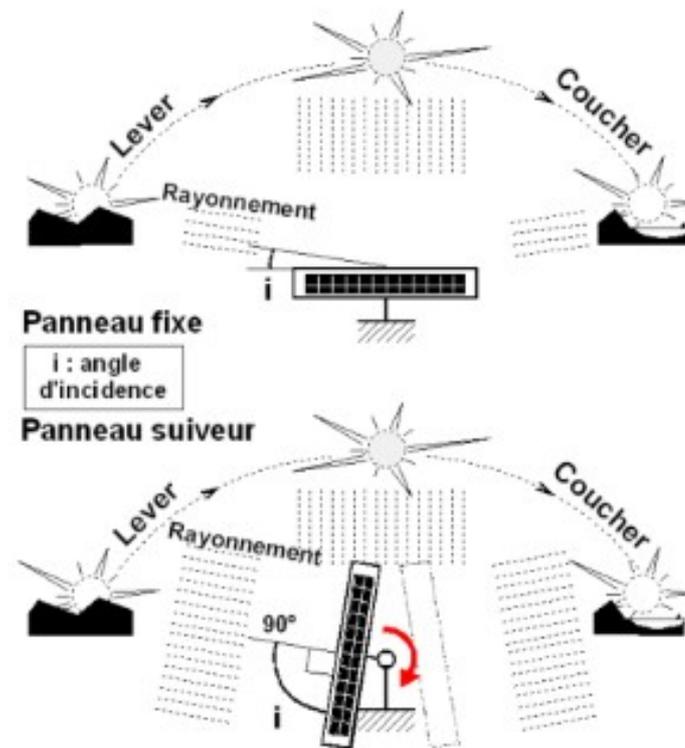


En effet, lorsqu'un panneau photovoltaïque est fixe par rapport au sol et orienté vers le sud (implantation en hémisphère nord), son rendement énergétique n'est pas constant au cours de la journée : en début et en fin de journée, le mauvais angle d'éclairement du panneau diminue le rendement de la production électrique.

Lorsqu'un système suiveur oriente le panneau solaire en le faisant pivoter en direction du soleil et en maintenant en permanence un angle d'incidence des rayons « i » idéal, voisin de 90° , la production électrique par mètre carré de panneau atteint alors son maximum.

Tournesol	Version 2.1
Auteur : groupe	5/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016



Le système de positionnement par rapport au soleil se fera idéalement selon 2 axes (l'azimut d'angle et l'élévation ou hauteur d'angle) et permettra :

- Une augmentation de la production des panneaux photovoltaïques de 30 à 40 % (si elle est réalisée sur 2 axes) ;
- Une réduction de la surface du parc solaire tout en gardant le même potentiel;
- Une réduction du temps de retour sur investissement (en moyenne amortissement du système en 4 ans).

Le système technique devra donc permettre de produire efficacement l'énergie électrique nécessaire pour alimenter des matériels portatifs ou satisfaire des besoins locaux en des lieux isolés.

Le développement de l'application répondra aux exigences des exploitants:

- simplicité d'utilisation,
- correspondre aux contraintes définies,
- réalisable dans un délai de 200 heures.

Tournesol	Version 2.1
Auteur : groupe	6/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

II) Objectifs

Il s'agit donc de réaliser un programme complet qui assure le fonctionnement autonome d'une installation photovoltaïque motorisé permettant d'optimiser la récolte d'énergie.

Le système devra :

- Orienter efficacement les panneaux solaires photovoltaïques en toute sécurité pour optimiser la récolte d'énergie
- Assurer la régulation de l'énergie et le contrôle sécurisé de la charge des batteries récupérer les informations de l'ensemble de l'installation et les stocker
- Signaler et journaliser les alarmes
- Communiquer avec l'utilisateur via un mini-écran tactile

A) Architecture matérielle du système

La station de production d'énergie électrique sera composée de panneaux photovoltaïques motorisés et des équipements nécessaires au stockage et à la régulation de l'énergie :

- au moins deux panneaux photovoltaïques de type mono-cristallin
- au moins deux batteries VRLA AGM
- un régulateur de charge
- un système de commande de positionnement 2 axes
- une station météo (Anémomètre / Girouette / Thermomètre / Capteur de luminosité)
- un capteur de positionnement solaire

Le système sera construit autour d'un nano-ordinateur sur lequel sera relié un mini-écran tactile.

B) Contraintes techniques

Le système fonctionnera en régime continu (24h/24) sauf arrêt pour entretien (changement d'éléments, extension) ou maintenance (prévention ou correction d'un défaut). Le pilotage des différents modules composant le système sera donc assuré en permanence.

C) Contraintes économiques

Le système sera maqueté et conservé par l'établissement pour servir de support lors des « Journées Portes Ouvertes ». Le budget matériel sera donc à la charge de l'établissement et ne devra pas excéder 2000 €.

Tournesol	Version 2.1
Auteur : groupe	7/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

III) Mission du système

Le système **Tournesol** devra remplir les missions suivantes pour exploiter l'installation :

- Transformer l'énergie solaire produite par les panneaux photovoltaïques en énergie électrique ;
- Réguler l'énergie qui est réalisée par le régulateur de charge qui protège les batteries contre les risques de surcharges et les décharges au-dessous du seuil minimum (décharge profonde) ;
- Stocker l'énergie qui est réalisée par le parc de stockage constitué de batteries. Le parc de stockage permet de déphaser la production de la consommation de l'énergie et de palier aussi à des pointes de puissance ponctuelles ;
- Distribuer et protéger.

L'application embarquée devra donc assurer :

- Le fonctionnement en mode manuel ou automatique ;
- La commande du positionnement des panneaux solaires en toute sécurité ;
- Le pilotage du régulateur de charge;
- Le paramétrage des consignes de protection contre le vent ;
- L'acquisition des données des modules (suiveur solaire, régulateur de charge, et station météo) ;
- La surveillance des différents modules ;
- La visualisation des états, des données et des alarmes sur mini-écran tactile ;
- L'archivage des états, des données et des alarmes dans une base de données .

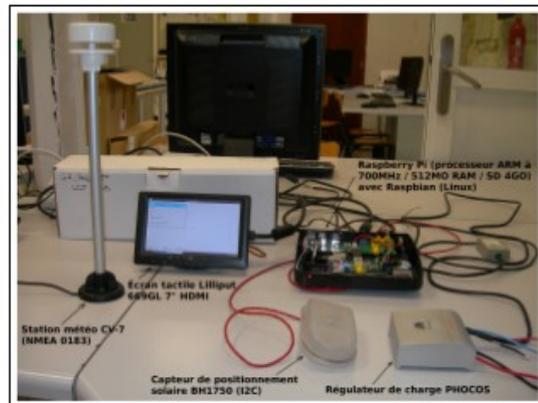
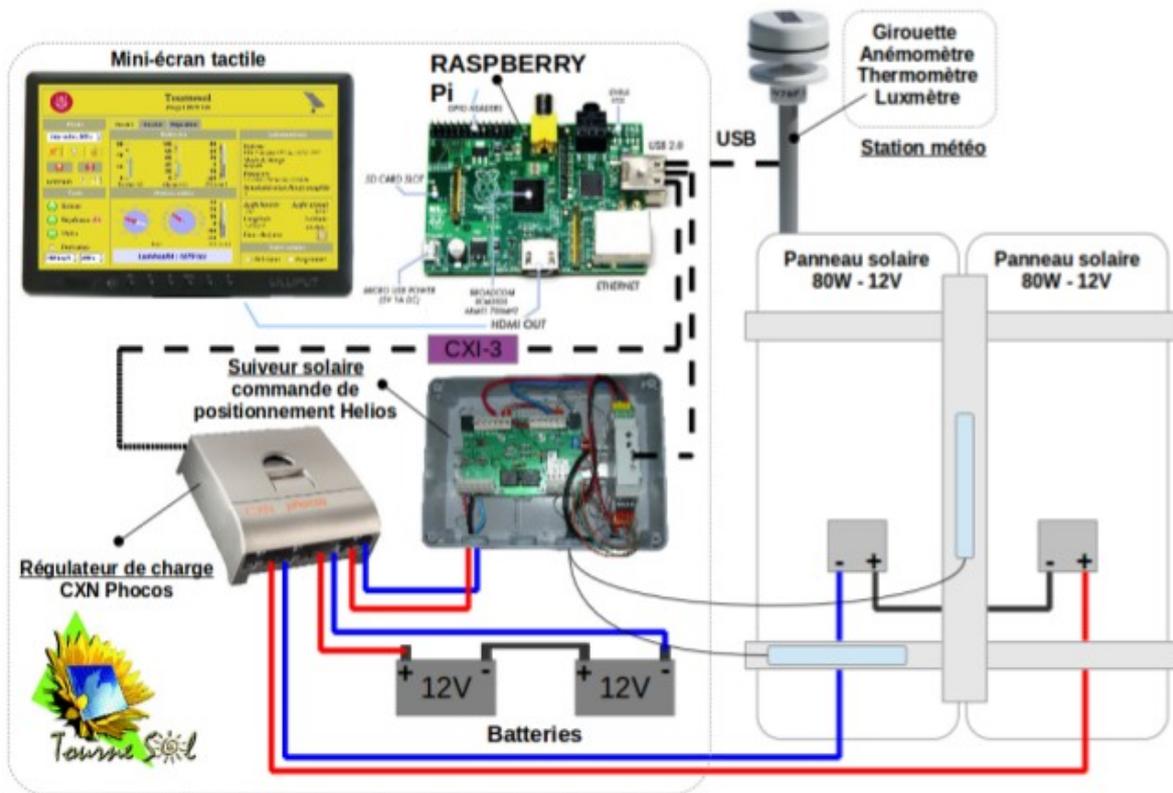
On distinguera les modules suivants :

- Suiveur solaire (panneaux photovoltaïque, système de commande de positionnement, des capteurs de positionnement solaire) ;
- Régulateur de charge (liaison avec les panneaux photovoltaïque, les batteries et les sorties ;
- Station météo (anémomètre, girouette, capteur de température, capteur de luminosité).

Tournesol	Version 2.1
Auteur : groupe	8/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

IV) Configuration d'exploitation



Tournesol	Version 2.1
Auteur : groupe	9/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

V) Diagrammes

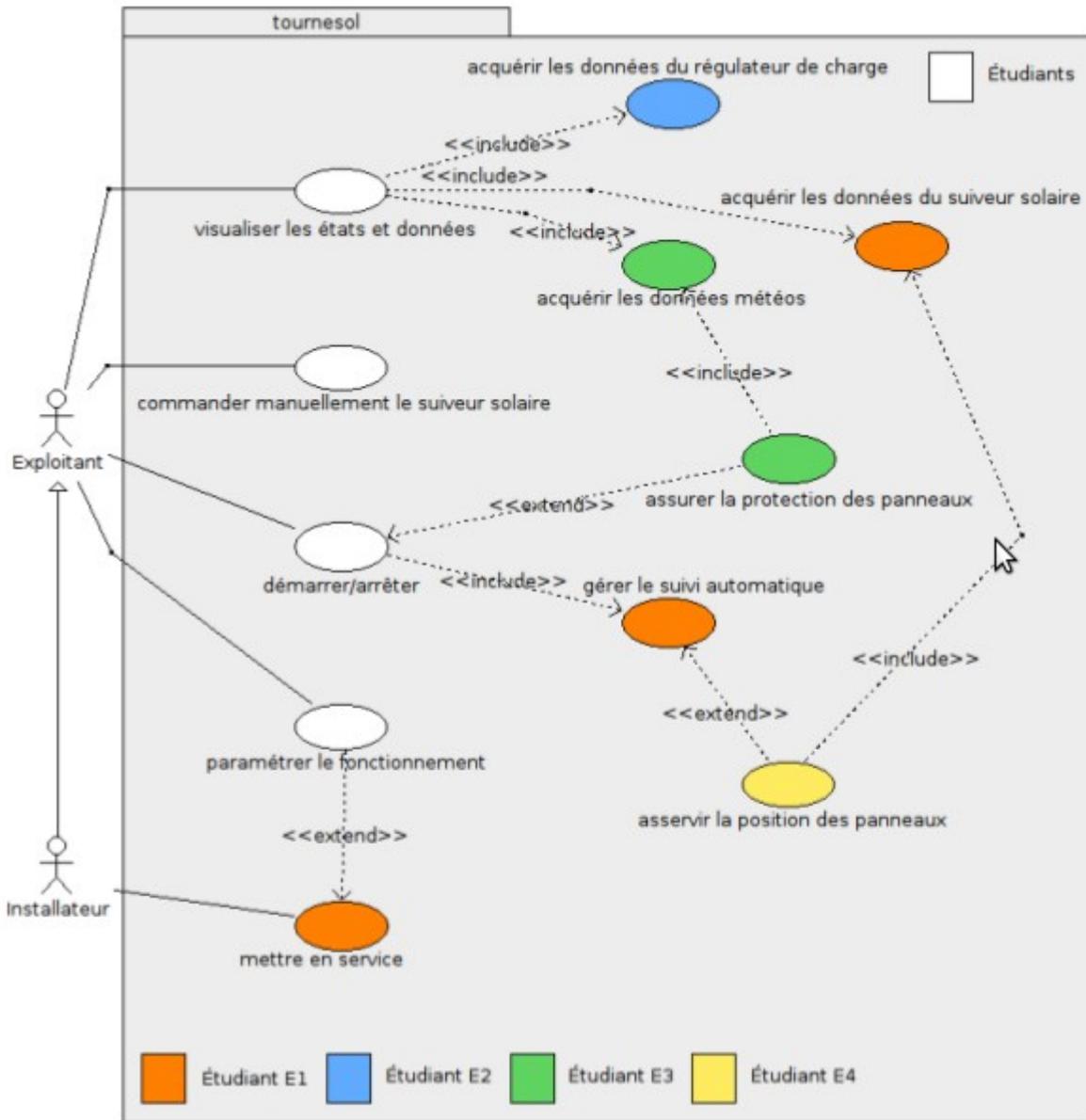


Diagramme de cas d'utilisations

Tournesol	Version 2.1
Auteur : groupe	10/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

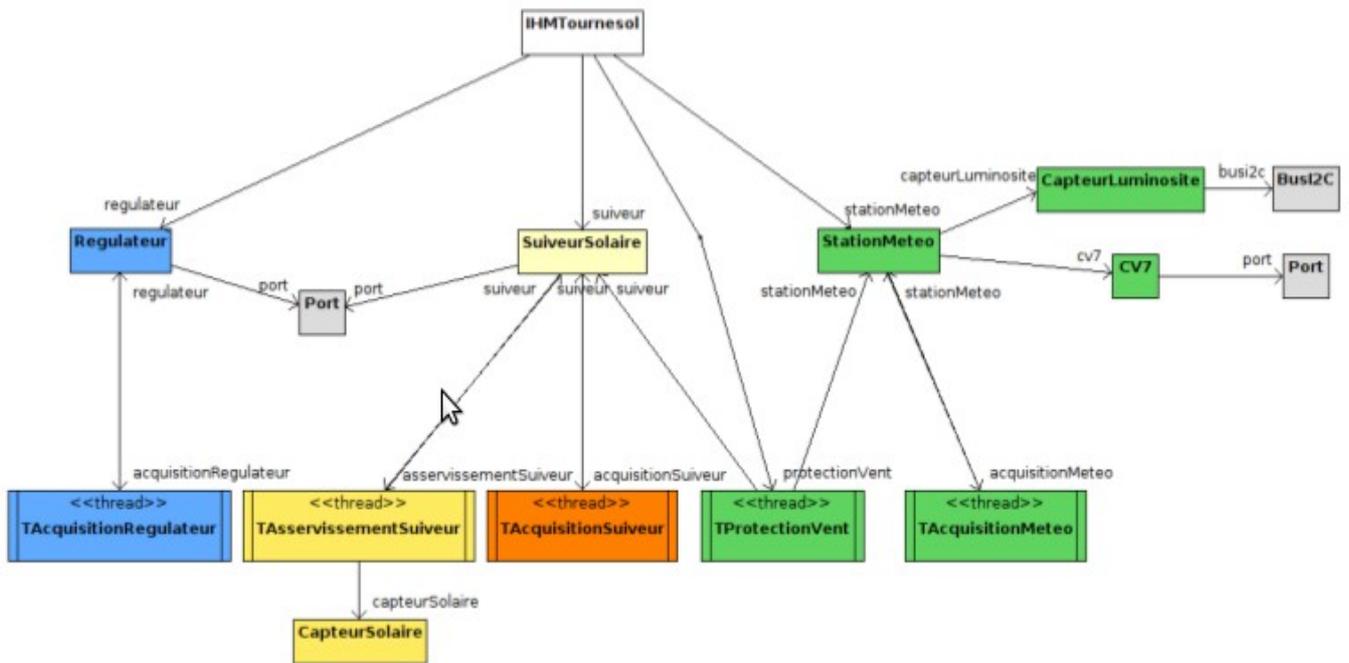


Diagramme de classes partiel

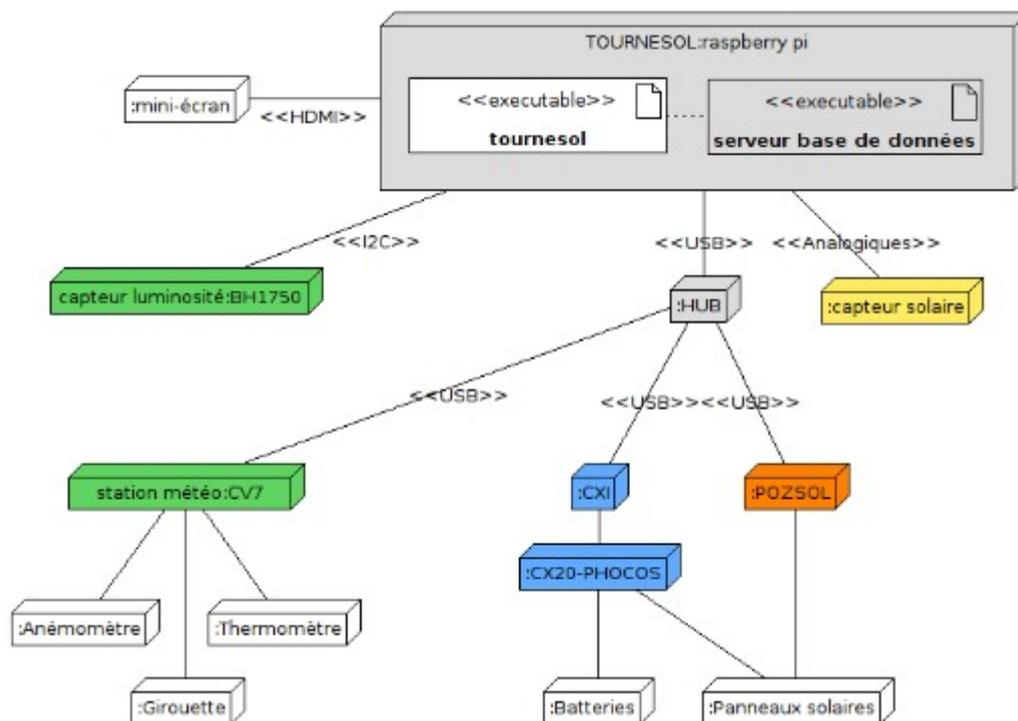


Diagramme de déploiement

Tournesol	Version 2.1
Auteur : groupe	11/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

VI) Description des fonctions

A) Fonction de mise en service

L'installateur commencera par régler l'heure et la date système.

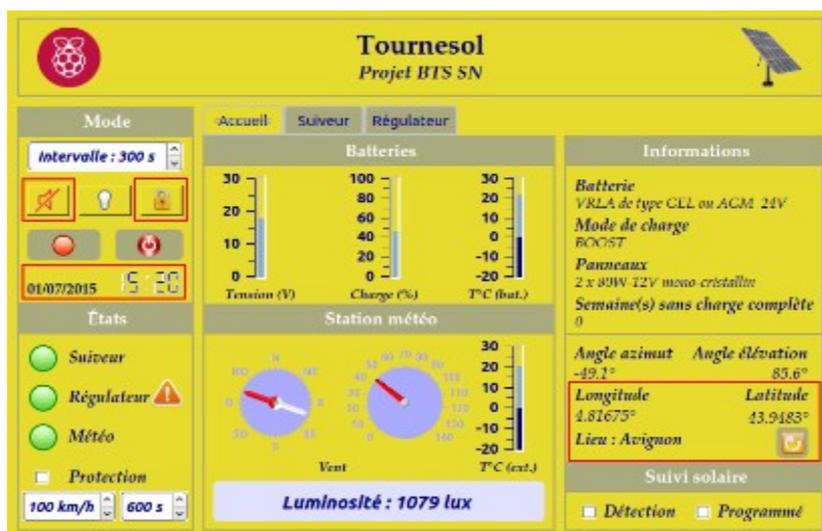
Avant de démarrer l'application, l'installateur devra aussi paramétrer le fichier de configuration en indiquant :

- la longitude, la latitude et le lieu de l'installation ;
- les noms des fichiers de périphériques associés aux modules « suiveur solaire », « régulateur de charge » et « station météo » ;
- les périodicités d'acquisition des différents modules.

L'installateur lancera ensuite l'application et vérifiera l'exactitude des informations (date/heure, longitude/latitude). Il initialisera l'installation :

- en enregistrant les données de localisation et d'horodatage ;
- en sélectionnant le type de batteries (au plomb à liquide électrolyte ou VRLA de type GEL 4 ou AGM) ;
- en activant ou désactivant le verrouillage de la programmation prévenant ainsi tout risque de modification accidentelle des paramètres ;
- en activant ou désactivant le signal sonore (buzzer) de niveau de charge de la batterie.

L'installateur pourra poursuivre en paramétrant l'installation ou laisser cette tâche à l'exploitant .



Exemple de mise en service par mini écran tactile

Tournesol	Version 2.1
Auteur : groupe	12/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

B) Fonction de régulation de l'énergie

Il s'agira de contrôler la régulation d'énergie de l'installation. Le régulateur protège la batterie de toute surcharge du champ solaire et de décharges trop importantes dues à la surconsommation des charges de sortie.

L'exploitant définira le type de batteries (au plomb à liquide électrolyte ou VRLA de type GEL ou AGM) à contrôler et la fonction "coupure charge faible" en sélectionnant un mode parmi ceux proposés. Le contrôleur dispose de cinq modes destinés à éviter que la batterie ne se décharge complètement (voir document constructeur).

Il pourra aussi activer ou désactiver :

- le verrouillage de la programmation
- le verrouillage du signal sonore (buzzer) de niveau de charge de la batterie
- la charge de sortie raccordée à son installation (mode jour/nuit).



Exemple d'IHM pour la régulation de l'énergie

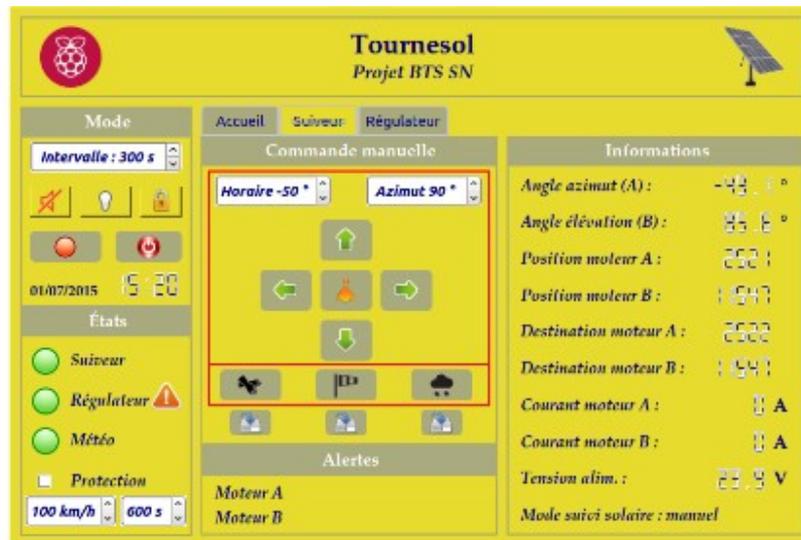
C) Fonction de commande manuelle d'orientation des panneaux

Par le mini écran tactile, l'utilisateur pourra commander manuellement l'orientation des panneaux soit :

- en fixant les valeurs des angles d'azimut et d'élévation ;
- en pilotant les axes suivant 4 directions ;
- en sélectionnant une position pré-programmée (nettoyage, nuit et neige).

Tournesol	Version 2.1
Auteur : groupe	13/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016



Exemple d'IHM permettant la commande manuelle d'orientation des panneaux solaires

D) Fonction de suivi automatique

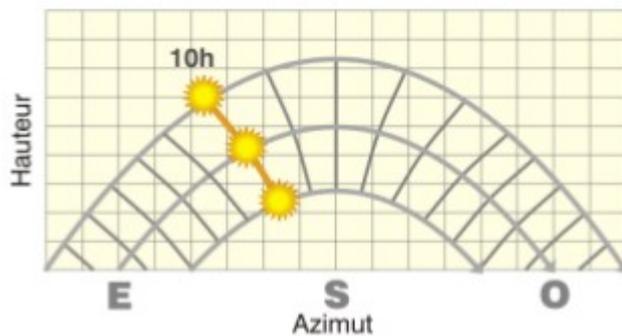
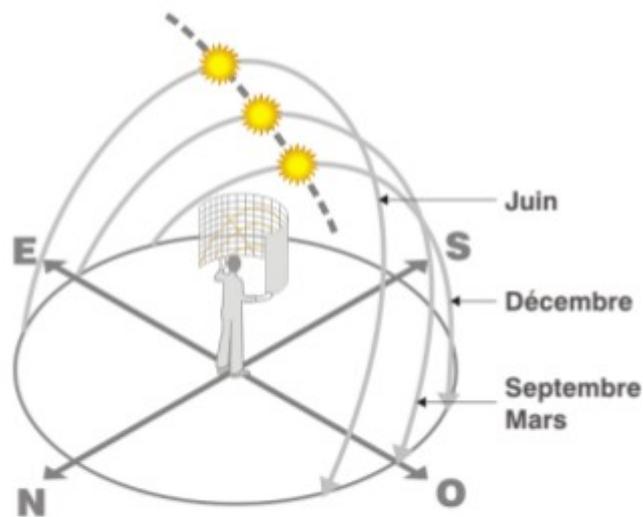
L'objectif est de s'orienter vers le soleil (perpendiculairement au rayonnement) tout au long de la journée, ce qui aura pour effet d'augmenter la production d'énergie de manière significative. On suivra le soleil en utilisant deux axes : en azimut (d'est en ouest, à mesure de l'avancée de la journée) et en hauteur (selon la saison et, de nouveau, l'avancée de la journée).

L'exploitant aura la possibilité de choisir le mode de positionnement du suiveur solaire : mode suivi automatique ou mode manuel.

Si l'exploitant a activé le suivi automatique, il devra préciser à quel intervalle de temps le suiveur solaire corrigera sa position pour suivre le soleil. Les valeurs possibles sont de 60 à 900 secondes (1-15 minutes).

Tournesol	Version 2.1
Auteur : groupe	14/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016



Le système offrira deux modes de suivi automatique :

- par un **système de positionnement astronomique temporel TdAPS** qui décrit de manière très juste le parcours du soleil dans le ciel en fonction du jour de l'année.

Ce système calcule la meilleure direction d'orientation en ne dépendant que de l'heure, de la date et de la latitude. Ce mode présente l'avantage de ne pas dépendre des conditions extérieures d'éclairement mais nécessite un système d'horloge.

- par un **système d'asservissement à l'aide d'un capteur solaire** (basé sur 4 photorésistances) qui détecte la position dans laquelle les panneaux solaires sont le mieux éclairés. Ce mode a pour avantage de pouvoir être mise en œuvre sans effectuer de calcul d'équations mais il est soumis aux conditions climatiques. Un ensoleillement diffus à cause d'une couverture nuageuse peut alors devenir très énergivore, la cellule tentant en permanence de trouver le maximum d'intensité lumineuse.

Tournesol	Version 2.1
Auteur : groupe	15/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

E) Fonction de protection contre le vent

L'exploitant pourra activer un mode de protection VENT pour ses panneaux.

Si le vent est trop fort, il pourrait endommager les panneaux. Par conséquent, il convient de déplacer les panneaux dans une position de sécurité sauvegardée (angles A et B normalement en position horizontale totale), lorsque la vitesse du vent dépasse le seuil de vitesse maximale pour la première fois. Après avoir été placé en position de sécurité, le suiveur doit attendre un temps minimum avant de revenir à un fonctionnement normal.

Si la vitesse du vent dépasse le seuil de vitesse à nouveau, une nouvelle période de temps de sécurité doit être respecté.

L'exploitant devra préciser les réglages :

- du seuil de vent en km/h ;
- du temps minimum d'attente en secondes ;
- de la position de sécurité (angles A et B en degrés 5) où le suiveur se positionne pour éviter d'endommager les panneaux.

L'ensemble des paramètres de protection seront stockés dans la base de données.

F) Fonction de surveillance des alarmes

Le système doit informer l'exploitant (alarme) en cas de problème :

- du suiveur solaire (erreurs de communication, de positionnement, de commandes d'axes, ...) ;
- du régulateur de charge (erreurs de communication, de surcharge, de surchauffe, ...) ;
- de la station météo (erreurs de communication, de mesures, ...) ;
- de la protection contre le vent (dépassement de seuil).

Les alarmes seront journalisées dans la base de données.

G) Fonction de paramétrage :

Le système dispose d'un mini-écran tactile permettant son paramétrage.

L'exploitant pourra aussi sélectionner :

- le verrouillage de la programmation
- le verrouillage du signal sonore (buzzer) de niveau de charge de la batterie
- le mode de suivi automatique (asservissement ou TdAPS) ;
- l'activation du mode de protection contre le vent ;
- l'activation du suivi automatique.

Tournesol	Version 2.1
Auteur : groupe	16/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

L'exploitant pourra donc régler :

- l'intervalle de mise à jour du positionnement en secondes ;
- la vitesse de vent maximale en km/h et le temps d'attente en secondes avant un retour à un fonctionnement normal ;
- les positions pré-programmées (nettoyage, vent, nuit et neige).

H) Fonction d'acquisition des données

Périodiquement, il sera réalisé une acquisition des données en provenance des différents modules :

- « suiveur solaire » :
 - les angles A et B en degrés
 - les positions A et B en pas
 - les destinations A et B en pas
 - les courants moteurs A et B en Ampères
 - la tension en Volts
 - le mode de suivi (manuel ou automatique)
- « régulateur de charge » :
 - la tension en Volts
 - le niveau de charge en %
 - la fin de tension de charge en Volts
 - le surplus d'énergie en %
 - les courants photovoltaïque et charge en Ampères
 - le niveau de charge le matin et le soir en %
 - les consommations (panneaux, charge) et excédant en Ampères/h
 - le type de batteries, le mode de charge, le nombre de semaines sans charge complète des batteries et la température
- « station météo » :
 - la vitesse en km/h et direction du vent
 - la température de l'air en degré Celsius
 - la luminosité en lux

Ces données seront ensuite affichées et archivées dans la base de données.

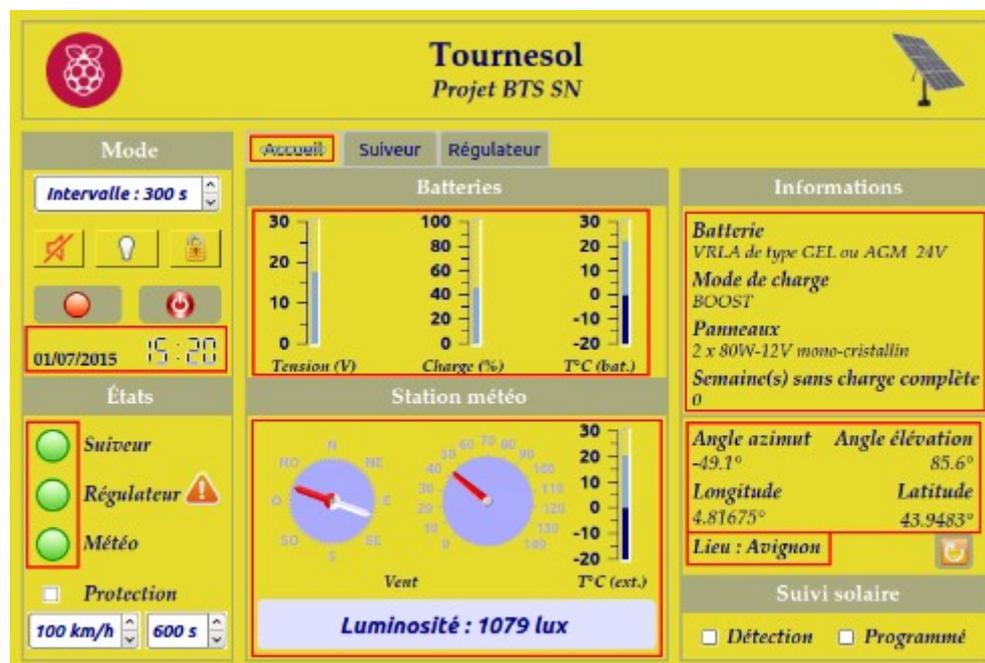
Tournesol	Version 2.1
Auteur : groupe	17/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

I) Fonction de visualisation

La visualisation se fera sur le mini-écran tactile. L'écran d'accueil permettra une visualisation des informations principales :

- La date et l'heure ;
- Les caractéristiques détaillées de l'installation (type de batteries, mode de charge, nombre de semaines sans charge complète des batteries, caractéristiques des panneaux photovoltaïques) ;
- La position des panneaux (angles en degrés d'azimut et d'élévation) ;
- Les informations sur le lieu de l'installation (longitude et latitude, nom du lieu) ;
- Les informations en provenance de la station météo (vitesse et direction du vent, température de l'air et luminosité en lux) ;
- Les informations en provenance du régulateur de charge (tension en Volts, charge en % et température des batteries en degré Celcius) ;
- Les états des modules ;
- L'intervalle de mise à jour du positionnement en secondes ;
- La vitesse de vent maximale en km/h et le temps d'attente en secondes avant un retour à un fonctionnement normal ;
- Les alarmes visuelles.

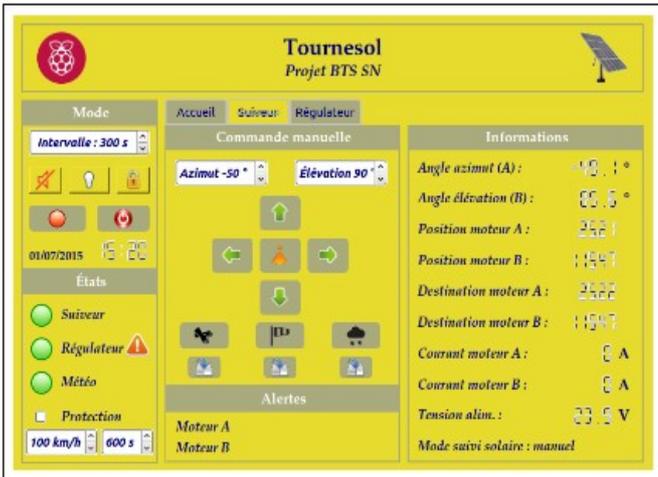


Exemple d'IHM assurant la visualisation des données

Tournesol	Version 2.1
Auteur : groupe	18/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Une visualisation détaillée des modules « suiveur solaire » et « régulateur de charge » sera aussi disponible :



L'IHM de l'écran du suiveur solaire



et celle du régulateur de charge

Tournesol	Version 2.1
Auteur : groupe	19/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

VII) Exigences

Le produit à réaliser devra répondre aux facteurs de qualité suivants :

<i>Facteurs liés à l'environnement d'exploitation et d'utilisation</i>	
Facteur	Signification
Couplage	capacité de liaison avec un autre logiciel
Efficacité	optimisation de l'utilisation des ressources
Maniabilité	facilité d'emploi pour l'utilisateur
Robustesse	conservation d'un fonctionnement conforme aux besoins exprimés, en présence d'événements non prévus ou non souhaités (arrêt normal, intempestif ou d'urgence)
Sécurité	disponibilité assurant la continuité des traitements

<i>Facteurs liés à l'environnement de maintenance et de suivi</i>	
Facteur	Signification
Adaptabilité	facilité de suppression, d'évolution de fonctionnalités existantes ou d'ajout de nouvelles fonctionnalités
Maintenabilité	facilité de localisation et de correction des erreurs résiduelles
Portabilité	minimisation des répercussions d'un changement d'environnement logiciel et matériel

VIII) Le développement

En ce qui concerne les exigences qualité du développement :

- le développement se fera de manière itérative et incrémentale (méthodologie UP) ;
- la modélisation du système devra être réalisée avec le langage de modélisation

UML 2 ;

- l'architecture du logiciel sera « orientée objet » et multitâche ;
- le système d'exploitation retenu est GNU Linux ;
- le langage de programmation à utiliser est le C++ ;
- l'IHM doit être développée avec l'API Qt 4 de Qt Development Frameworks,

filiale de Digia.

- l'implémentation des structures de données doit privilégier les structures de données de la STL ou équivalent dans Qt.
- le codage doit respecter le standard de codage C++ en cours dans la section ;
- la chaîne de production des exécutables doit être réalisée avec un gestionnaire de type make et le compilateur GNU g++ ;
- le gestionnaire de gestion de versions utilisé sera subversion ;

Tournesol	Version 2.1
Auteur : groupe	20/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

- le logiciel possédera une suite de tests unitaires cppunit ;
- la documentation du code sera générée à partir de doxygen ;
- la documentation du projet (README, TODO, Changelog et BUGS) respectera le langage de balisage Markdown ;
- le serveur de base de données est MySQL. Les requêtes seront exprimées en langage SQL.

Méthodologie	itérative et incrémentale (UP)
Modélisation	UML 2
Architecture logicielle	Orientée objet et multitâche
Système d'exploitation	GNU Linux
Environnement de développement	Qt Creator et Qt Designer
Compilateur	GNU g++/gcc
Débugueur	GNU gdb
Chaîne de fabrication	qmake et GNU make
API GUI	Qt
Atelier de génie logiciel	bouml
Plate-forme de tests unitaires	CppUnit
Logiciel de gestion de versions	subversion (client svn)

Le dépôt subversion sera organisé de la façon suivante :

- un répertoire trunk pour le développement courant,
- un répertoire tags pour le stockage des versions identifiées,
- un répertoire branches pour l'expérimentations, les tests et la correction des bugs.

Tournesol	Version 2.1
Auteur : groupe	21/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

IX) La documentation à produire

La suite bureautique libre LibreOffice ou Apache OpenOffice sera utilisée pour tous les documents “papier” et les diaporamas.

Chaque page du dossier technique devra être clairement identifiée par :

- un entête comprenant : le nom de l’établissement, la session du BTS et le nom du projet.
- un pied de page comprenant : l’identifiant du document, la date de mise à jour, le nom du ou des auteurs (les initiales sont acceptées), le numéro de page du nombre total de pages.

La documentation du projet sera rédigé en respectant le langage de balisage Markdown. On pourra ensuite la générer avec l’outil pandoc dans les formats ODT, PDF ou HTML. Elle comprendra les fichiers textes suivants :

- *README* : contient des informations sur le projet (nom, objectifs, date de début), l’équipe de développement (nom, prénom et courriel de chaque membre), l’version de l’application, la description détaillée des fonctionnalités (ce que fait l’application actuellement), la licence applicable et la procédure d’installation et les instructions d’exploitation ;
- *TODO* : (liste des tâches à réaliser) contient l’énumération de ce que l’équipe se fixe comme tâches à réaliser. Les éléments présents sont biffés une fois réalisés et se retrouveront dans le journal des améliorations et modifications apportées au programme ChangeLog ;
- *Changelog* : (journal des améliorations et modifications) contient l’énumération de ce que l’équipe a effectué comme travail sur le projet ;
- *BUGS* : contient la liste des défauts constatés non corrigés.

X) La gestion des modifications

Les modifications liées à des erreurs dans la demande du client (cahier des charges) ou des changements de stratégies liés à l’environnement ou les conditions financières ou des choix technologiques, devront faire l’objet d’une procédure de demande de modification clairement définie et approuvée par le responsable du projet et du client.

La procédure de gestion des modifications se déroule en trois temps :

1. établissement d’une demande de modification (remplir la "Fiche de demande de modification" fournie)
2. prise en compte de la modification (validée par un professeur responsable du projet)
3. réalisation de la modification

Un **avenant** sera alors joint au dossier technique.

Tournesol	Version 2.1
Auteur : groupe	22/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

II-) Contrat individuel Étudiant 1 (Gonzalves Rémy) : Module de gestion du suiveur solaire

I) Objectifs de l'étudiant 1

L'étudiant 1 a la charge du "Module de gestion du suiveur solaire" qui permet d'acquérir des données, de les archiver dans une base de données et d'assurer l'orientation des panneaux manuellement (en fixant les angles d'azimut et d'élévation ou en dirigeant les quatre axes de direction) et automatiquement (dans le mode programmé TdAPS). Le mini écran tactile permettra de visualiser les informations principales du suiveur solaire et de paramétrer son fonctionnement.

Il devra donc :

- Assurer la mise en service de l'installation
- Orienter efficacement les panneaux en toute sécurité
- Commander manuellement les panneaux
- Surveiller les alarmes
- Paramétrer le fonctionnement
- Acquérir les données
- Visualiser les états et données
- Archiver les états, données et alarmes

II) Recette minimale de l'étudiant 1 :

- La date et l'heure sont affichées périodiquement ;
- L'intervalle de suivi automatique est pris en compte par le suiveur solaire et il est enregistré dans la base de données ;
- L'activation et la désactivation du suivi automatique est effective et le mode programmé TdAPS est pris en compte ;
- Les informations sur l'installation sont affichées et il est possible de réaliser la mise en service ;
- L'état du suiveur solaire est visible sur le mini-écran tactile ;
- Les angles d'azimut et d'élévation courants sont affichés périodiquement sur l'écran d'accueil ;
- Les données détaillées du suiveur solaire sont affichées périodiquement dans l'écran correspondant et archivés dans la base de données ;
- La commande manuelle d'orientation des panneaux est réalisable soit en fixant les angles d'azimut et d'élévation soit en dirigeant les quatre directions.

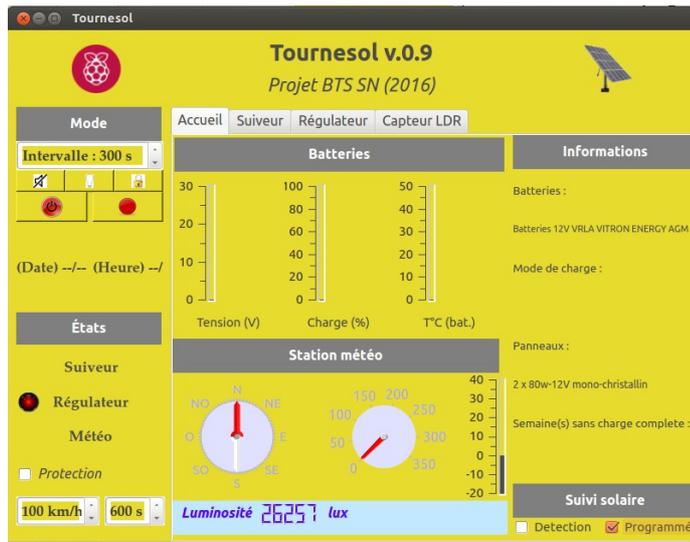
Tournesol	Version 2.1
Auteur : Rémy Gonzalves	23/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Production attendue :

- Une application informatique fonctionnelle ;
- Un modèle UML complet de la partie à développer ;
- Le code source commenté de l'application ;
- Les documentations prévues au paragraphe « Exigences sur la Livraison ».

Dans l'onglet accueil :



Dans l'onglet suiveur :



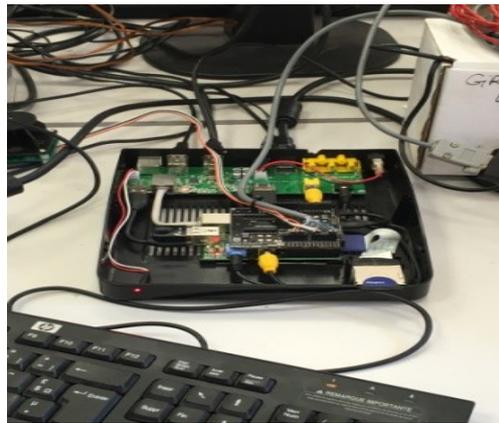
Tournesol	Version 2.1
Auteur : Rémy Gonzalves	24/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

III) Matériel mis en oeuvre

RPI (Raspberry Pi) :

Nano-ordinateur modèle B à processeur ARM1176JZF-S 700 Mhz équipé de 512 MO de RAM et d'une carte SD 4GO (Raspbian OS).



MINI-ECRAN :

Ecran tactile Lilliput 669GL-70NP/C/T 7° HDMI.



Tournesol	Version 2.1
Auteur : Rémy Gonzalves	25/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Suiveur Solaire (SM44M1V3P):

Suiveur solaire 2 axes SAT CONTROL équipé de deux panneaux 80W-12V mono-cristallin



Systeme de commande de positionnement (POZSOL) :

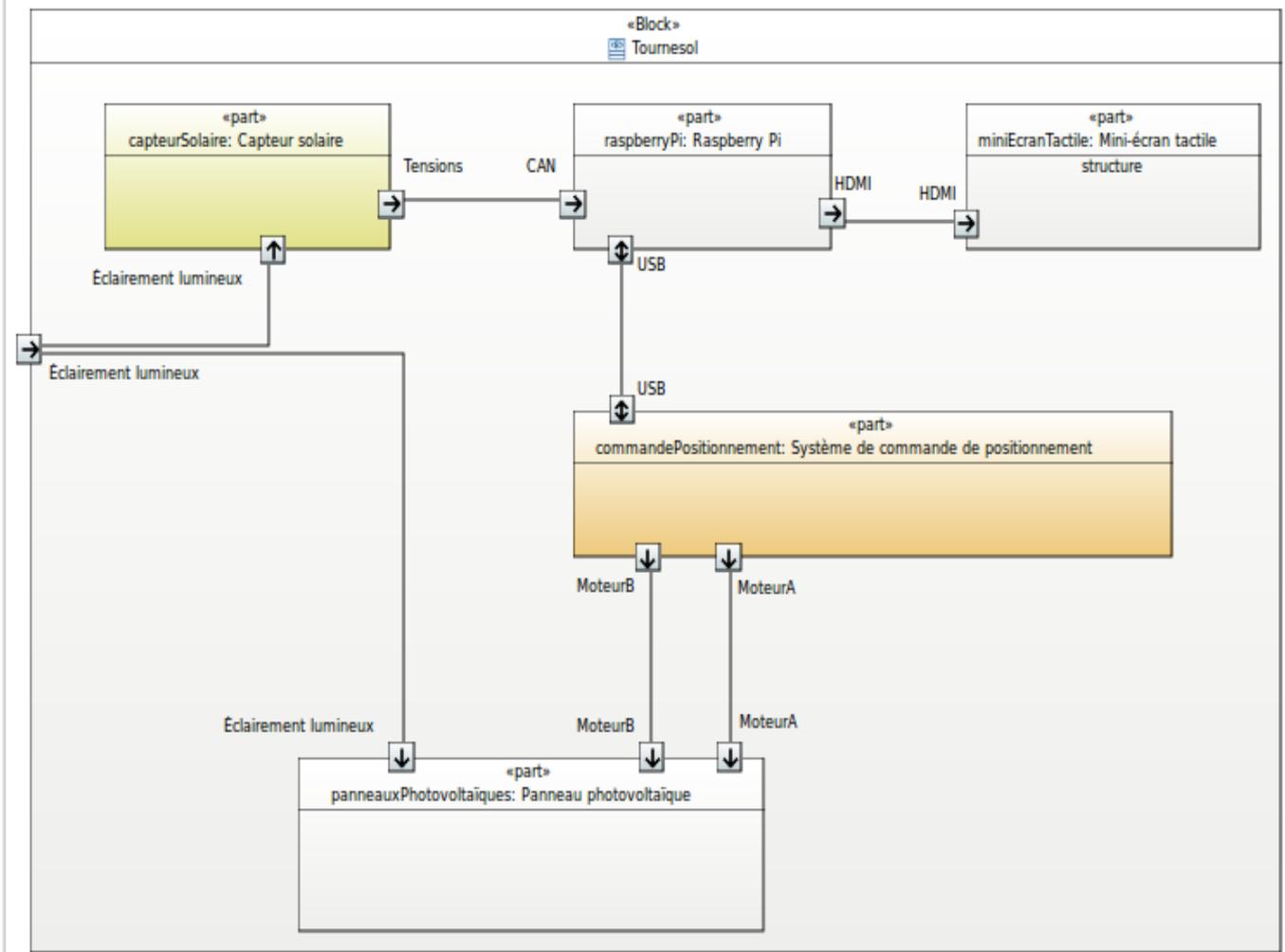
Système de commande de positionnement 2 axes avec interface USB (SM44M1V3P)

Tournesol	Version 2.1
Auteur : Rémy Gonzalves	26/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

IV) Diagramme interne de blocs

Diagramme interne du bloc Tournesol concernant le suiveur solaire :



Tournesol	Version 2.1
Auteur : Rémy Gonzalves	27/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

V) Functionalities in charge

Ref.	Fonction de service	Critère	Niveau	Flexibilité
FP-1	Assurer la mise en service	Localisation	Longitude/Latitude	F0
		Date/Heure	réglage automatique ou manuel	F0
		Attribution des ports	par fichier	F0
FP-3	Orienter efficacement les panneaux en toute sécurité	Mode de suivi automatique	mode TdAPS	F0
FP-4	Commander manuellement les panneaux	Par saisie	angles azimut et élévation (en degrés)	F0
		Par boutons	4 directions sous forme d'icônes	F0
FP-5	Surveiller les alarmes	Module à surveiller	suiveur solaire	F0
		Type d'alarme	visuelle	F0
FP-6	Paramétrer le fonctionnement	Sélection du mode de fonctionnement du suivi	manuel ou automatique	F0
		Sélection du mode de suivi automatique	2 modes (détection soleil ou TdAPS)	F0
		Réglage de l'intervalle de mise à jour du positionnement	en secondes (300 par défaut)	F0
FP-7	Acquérir les données	Périodicité	configurée par fichier	F0
		Module	suiveur solaire	F0
FP-8	Visualiser les états et données	Données	avec les unités	F0
		États	sous forme d'icône	F0
		Format horodatage	jj/mm/aaaa hh:mm	F0
FP-9	Archiver les états, données et alarmes	Base de données	SQL	F0
		Horodatage mesures	aaaa-mm-jj hh:mm:ss	F0

Tournesol	Version 2.1
Auteur : Rémy Gonzalves	28/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

VI) Tâches à réaliser

Tâches à réaliser	Production attendue	Estimation horaire
S'approprier le cahier des charges	Le cahier des charges est explicité. Les tâches à réaliser sont identifiées et les ressources sont définies.	16 h
Rechercher des solutions issues de l'innovation technologique pour le suivi automatique	Une étude comparative argumentée des différentes solutions est produite.	6 h (SCP)
Installer et configurer le système d'horloge du système	Un compte rendu est rédigé.	4 h (SCP)
Installer et configurer son environnement de développement	Le poste de développement est opérationnel.	2 h
Installer et raccorder le suiveur solaire	L'appareil fonctionne. La procédure d'installation a été respectée. Un compte rendu est rédigé.	4 h
Étudier et documenter les caractéristiques du suiveur solaire	Le suiveur solaire est caractérisé à partir de l'étude et des mesures réalisées.	10 h (SCP)
Relier et paramétrer l'interface de communication	L'interface de communication est correctement paramétrée et fonctionnelle. Un plan de câblage est réalisé. Le rapport de tests de mise en œuvre est rédigé.	4 h
Mettre en œuvre les programmes de test fournis	Le rapport de tests est renseigné.	12 h + 2 h (SCP)
Finaliser la modélisation UML du module	Les diagrammes UML (diagrammes de séquence « paramétrer le suivi automatique » et « piloter les panneaux », diagramme d'état du suivi automatique) sont élaborés et finalisés.	20 h
Produire la maquette de l'IHM du module	La maquette de l'IHM correspond aux exigences du cahier des charges.	8 h
Coder et tester les classes du module	Les classes <code>SuiveurSolaire</code> , <code>TAcquisitionSuiveur</code> et la structure <code>DonneesSuiveur</code> sont codées et valides. La classe <code>IHMTournesol</code> est complétée et valide.	40 h
Réaliser les tests unitaires	Les tests unitaires des classes <code>SuiveurSolaire</code> et <code>TAcquisitionSuiveur</code> sont écrits et archivés.	16 h
Faire la recette du module	Le cahier de recette du module est validé.	4 h + 4 h (SCP)
Intégrer en équipe l'application complète	L'application est intégrée et fonctionnelle.	4 h
Rédiger le dossier technique et les documents relatifs au projet	Le dossier est rédigé en respectant les exigences.	30 h
Produire un guide de mise en route et d'utilisation du module.	Un manuel est fourni.	4 h
Gérer la planification	Le planning prévisionnel est établi. Le planning est actualisé avec une mise en évidence des écarts par rapport au prévisionnel.	10 h
Total		200 heures

Tournesol	Version 2.1
Auteur : Rémy Gonzalves	29/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

VII) Tests unitaires

Méthode *acquerir()* :

Ce que la méthode doit faire	Ce que la méthode fait	Validation
Utilise la méthode recevoir() de la classe port pour recevoir une trame du suiveur solaire	Reçoit la trame du suiveur solaire	Oui
Envoyer la trame reçue a la méthode decoderTrame(QString reponseSuiveur)	Envoi la trame pour le décodage	Oui
Utiliser la méthode executer(QString requete) de la classe BaseDeDonnees pour stocker les informations reçues dans la base de données	N'envoie pas de requête	Non

Méthode *decoderTrame()* :

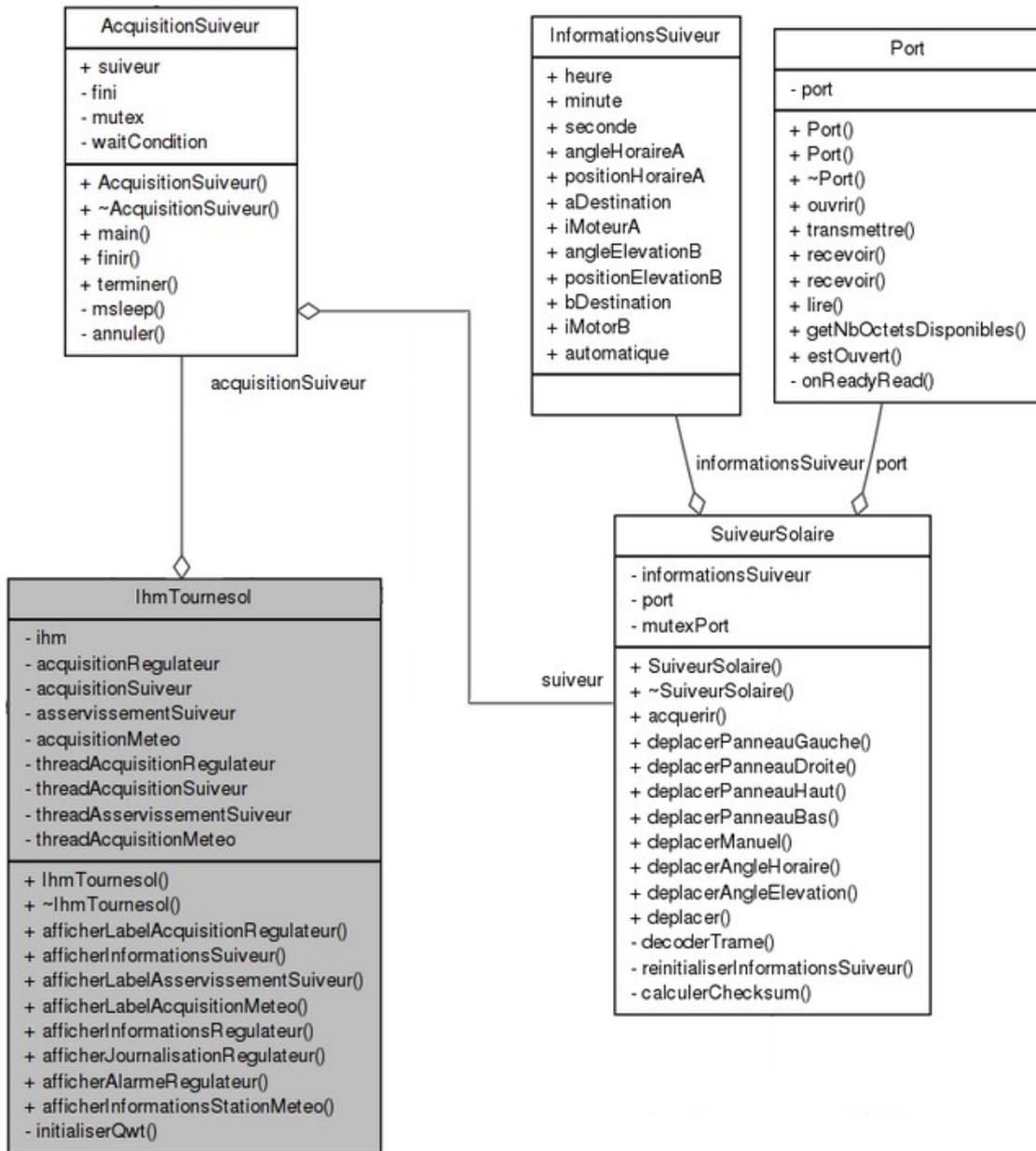
Ce que la méthode doit faire	Ce que la méthode fait	Validation
Utiliser la méthode split() de QList de QStringList sur la trame pour délimiter les informations de la trame	Utilise la méthode split() de QList de QStringList sur la trame	Oui
Détecter les index de la trame et trier les données pour afficher l'information souhaitée	Détecte les index de la trame et tri les données	Oui
Envoyer les données a l'IHM avec la méthode emit()	Envoi les données à l'IHM avec la méthode emit()	Oui

Tournesol	Version 2.1
Auteur : Rémy Gonzalves	30/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

VIII) Diagrammes

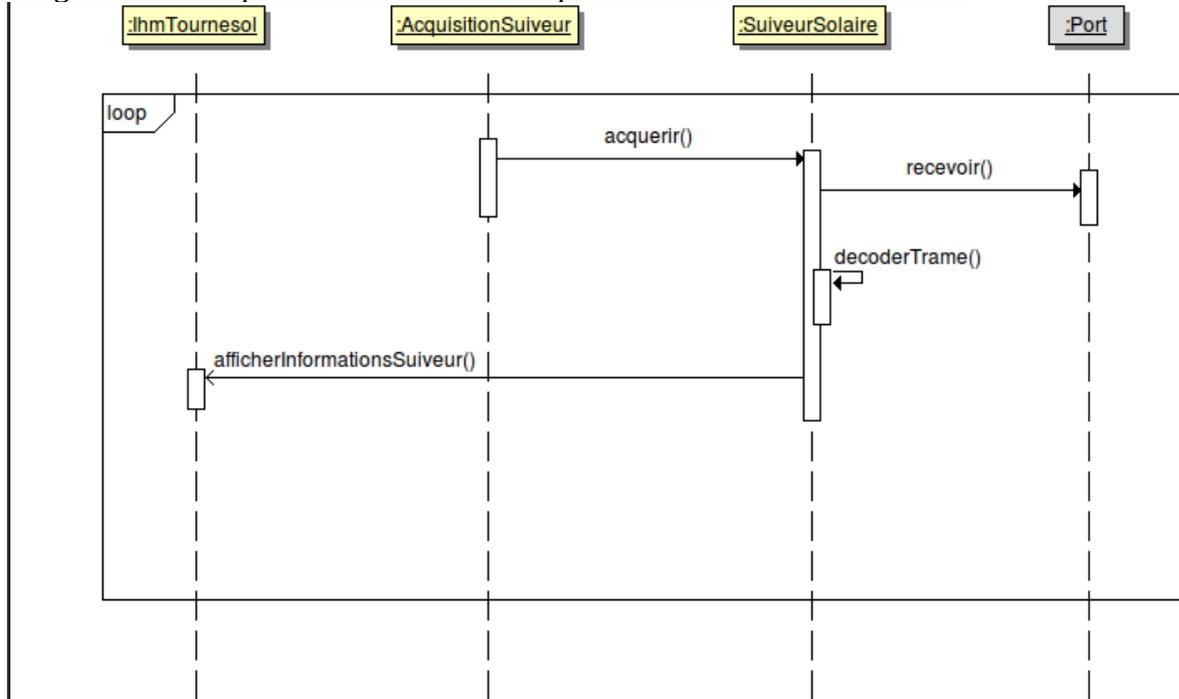
Diagramme de classe du suiveur solaire :



Tournesol	Version 2.1
Auteur : Rémy Gonzalves	31/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Diagramme de séquence du scénario « Acquérir les données du suiveur » :



IX) Décodage de la trame d'information du suiveur solaire

Lors de la réception des données du suiveur, que ce soit en hexadécimal ou en caractères, pour repérer les informations voulues, il faudra détecter le délimiteur qui signifie le début d'une information, puis l'index qui détermine quelle est l'information donnée. Le délimiteur de la trame du suiveur solaire est le caractère '\$' (0x24).

Exemple de réception de 3 informations :

\$ (09\$) 37\$ *01 → 0x24 0x28 0x30 0x39 0x24 0x29 0x33 0x37 0x24 0x2A 0x30 0x31

Ici on peut voir en vert des informations reçus en caractères (ASCII). Ensuite, on peut voir ici surligné que les différentes informations ont été transcrites en hexadécimal où chaque information est d'une couleur différente :

- En bleu : L'heure,
- En vert : Les minutes,
- En rouge : Les secondes.

Tournesol	Version 2.1
Auteur : Rémy Gonzalves	32/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

La donnée est donc découpée ainsi :

- Le délimiteur : '\$' (0x24)
- L'index (les différents index sont détaillés dans la documentation du protocole de communication) : ici 0x28, 0x29 et 0x2A
- Les valeurs correspondantes aux index (dans notre exemple : 09 pour l'heure, 37 pour les minutes et 01 pour les secondes)

X) Planification :

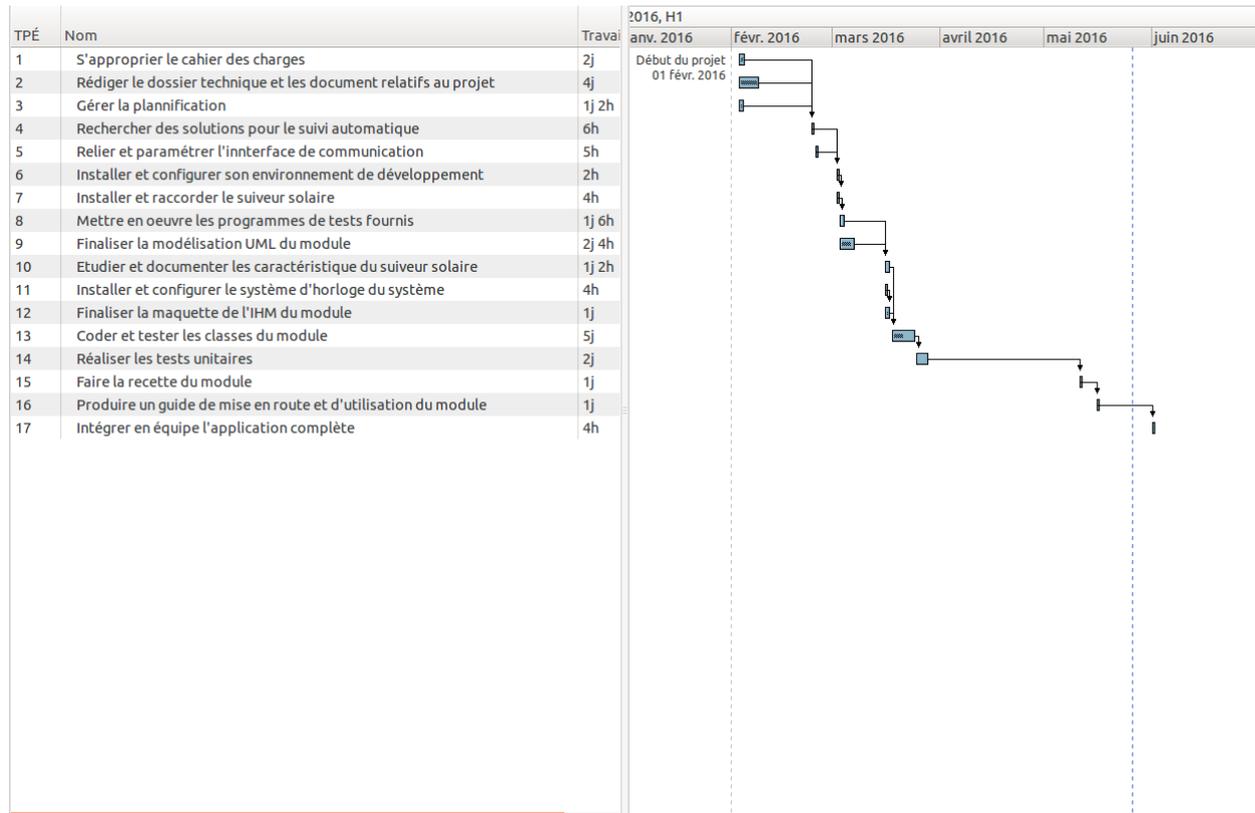
Planification initiale :

Tâches	Semaine																		Tot	
	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22		
S'approprier le cahier des charges	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	ET1	Tot
S'approprier le cahier des charges	10			4	2															16
Rechercher des solutions pour le suivi automatique				3		3														6
Installer et configurer son environnement de développement					2															2
Installer et raccorder le suiveur solaire					4															4
Étudier et documenter les caractéristiques du suiveur solaire							2	2	2			2	2							10
Relier et paramétrer l'interface de communication				4																4
Installer et configurer le système d'horloge du système							2	2												4
Mettre en œuvre les programmes de test fournis					6	8														14
Finaliser la modélisation UML du module					2	2	2	4	2			2	2	2	2					20
Produire la maquette de l'IHM du module							2	1				2			2				1	8
Coder et tester les classes du module							6	4	4			4	4	4	4	4	4	4	2	40
Réaliser les tests unitaires							1	2	2			2	2	2		2	2		1	16
Faire la recette du module															4				4	8
Intégrer en équipe l'application complète																			4	4
Rédiger le dossier technique et les documents relatifs au projet	1			2		2	1,5	2,5	3			2	2	2	2	4	4	2	2	30
Produire un guide de mise en route et d'utilisation du module.																2	2			4
Gérer la planification	2			2	0,5	0,5	0,5	0,5	1			0,5	0,5	0,5	0,5	0,5	0,5			10
Revue n°0					1															
Revue n°1									1											
Revue n°2															1					
Total																				200

Tournesol	Version 2.1
Auteur : Rémy Gonzalves	33/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Diagramme de Gantt :



Tournesol	Version 2.1
Auteur : Rémy Gonzalves	34/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

III) Contrat individuel étudiant n°2 (Adrien Thiel): Module de contrôle du régulateur de charge

I) Objectifs :

Vous intervenez en tant que technicien SN-IR pour “Concevoir une partie du système informatique Tournesol”.

Vous avez la charge du “**Module de contrôle du régulateur de charge**” qui permet :

- d’acquérir des données,
- de les archiver dans une base de données et
- de contrôler le régulateur de charge.

Un mini écran tactile permettra de visualiser les informations principales du module et de paramétrer son fonctionnement.



Tournesol	Version 2.1
Auteur : Adrien Thiel	35/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

II) Functionalities in charge

Ref.	Fonction de service	Critère	Niveau	Flexibilité
FP-2	Réguler l'énergie	Sortie	M/A	F2
FP-5	Surveiller les alarmes	Module à surveiller	régulateur de charge	F0
		Type d'alarme	visuelle	F0
FP-6	Paramétrer le fonctionnement	Choix des batteries	2 types	F2
		Fonction "coupure charge faible"	cinq modes	F2
		Programmation	verrou par l'écran tactile	F2
		Signal sonore (<i>buzzer</i>)	verrou par l'écran tactile	F2
FP-7	Acquérir les données	Périodicité	configurée par fichier	F0
		Module	régulateur de charge	F0
FP-8	Visualiser les états et données	Données	avec les unités	F0
		États	sous forme d'icône	F0
FP-9	Archiver les états, données et alarmes	Base de données	SQL	F0
		Horodatage mesures	aaaa-mm-jj hh:mm:ss	F0

III) Production attendue

- Une application informatique fonctionnelle ;
- Un modèle UML complet de la partie à développer ;
- Le code source commenté de l'application ;
- Les documentations prévues au paragraphe « Exigences sur la livraison »

Tournesol	Version 2.1
Auteur : Adrien Thiel	36/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

IV) Recette minimale étudiant 2

- ◆ Le verrouillage et déverrouillage de la programmation par clavier et du signal sonore sont pris en compte ;
- ◆ Les informations sur l'installation sont affichées ;
- ◆ L'état du régulateur de charge est visible sur le mini écran tactile ;
- ◆ Les valeurs de la tension, de la charge des batteries et de la température sont affichées périodiquement et graphiquement sur l'écran d'accueil ;
- ◆ Les données détaillées du régulateur de charge sont affichées périodiquement dans l'écran correspondant et archivées dans la base de données ;
- ◆ Les alertes signalées par le régulateur de charge sont affichées périodiquement dans l'écran correspondant et archivées dans la base de données ;
- ◆ Le choix du type de batteries et le mode de coupure sont paramétrables et fonctionnels ;
- ◆ La possibilité de quitter l'application avec succès.

V) Travail personnel de l'étudiant 2

Ma partie du projet consiste à créer la partie logicielle prenant en charge le régulateur de charge **PHOCOS-CX20** (voir présentation du matériel).

Cette partie logicielle consiste à envoyer des commandes au régulateur (trames 0x20 ou 0x21) pour ensuite recevoir des informations relevées par le régulateur puis les décoder et les traiter. Ces données seront ensuite affichées via l'IHM.

Mon travail consiste aussi à gérer la restitution de l'énergie, ainsi que des alarmes du régulateur en cas de décharge, de surcharge.

Tournesol	Version 2.1
Auteur : Adrien Thiel	37/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

VI) Tâches à réaliser

Tâches à réaliser	Production attendue	Estimation horaire
S'approprier le cahier des charges	Le cahier des charges est explicité. Les tâches à réaliser sont identifiées et les ressources sont définies.	10 h
Rechercher des solutions issues de l'innovation technologique pour le stockage d'énergie	Une étude comparative argumentée des différentes solutions est produite.	6 h (SCP)
Installer et configurer son environnement de développement	Le poste de développement est opérationnel.	1h
Installer et raccorder les appareils	Les appareils fonctionnent. La procédure d'installation a été respectée. Un compte rendu est rédigé.	4h
Étudier et documenter les caractéristiques des batteries	Les batteries sont caractérisées à partir de l'étude et des mesures réalisées.	10 h (SCP)
Relier et paramétrer l'interface de communication	L'interface de communication est correctement paramétrée et fonctionnelle. Un plan de câblage est réalisé. Le rapport de tests de mise en œuvre est rédigé.	4h
Mettre en œuvre les programmes de test fournis	Le rapport de tests est renseigné.	14 h + 4 h (SCP)
Finaliser la modélisation UML du module	Les diagrammes UML (diagrammes de séquence « paramétrer le régulateur de charge », diagrammes d'états correspondants) sont élaborés et finalisés.	20 h
Produire la maquette de l'IHM du	La maquette de l'IHM correspond	8h

Tournesol	Version 2.1
Auteur : Adrien Thiel	38/101

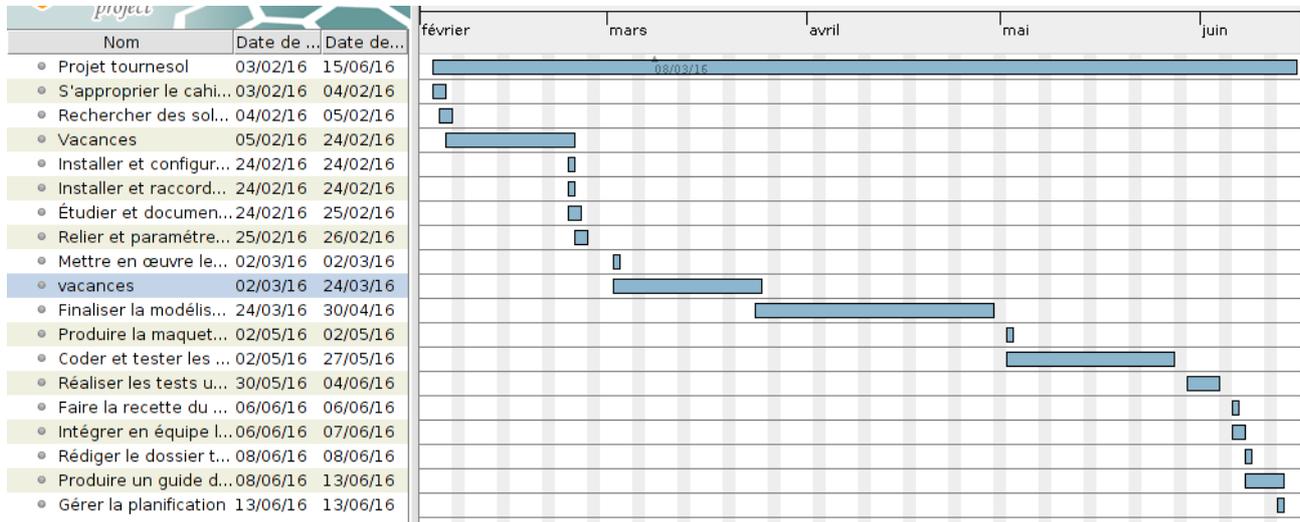
Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

module	aux exigences du cahier des charges.	
Coder et tester les classes du module	Les classes Regulateur, acquisitionRegulateur et la structure DonneesRegulateur sont codées et valides. La classe IHMTournesol est complétée et valide.	40 h
Réaliser les tests unitaires	Les tests unitaires des classes Regulateur et AcquisitionRegulateur sont écrits et archivés.	16 h
Faire la recette du module	Le cahier de recette du module est validé.	4 h + 4 h (SCP)
Intégrer en équipe l'application complète	L'application est intégrée et fonctionnelle.	4h
Rédiger le dossier technique et les documents relatifs au projet	Le dossier est rédigé en respectant les exigences.	30 h
Produire un guide de mise en route et d'utilisation du module.	Un manuel est fourni.	4h
Gérer la planification	Le planning prévisionnel est établi. Le planning est actualisé avec une mise en évidence des écarts par rapport au prévisionnel.	10 h
Total		200 heures

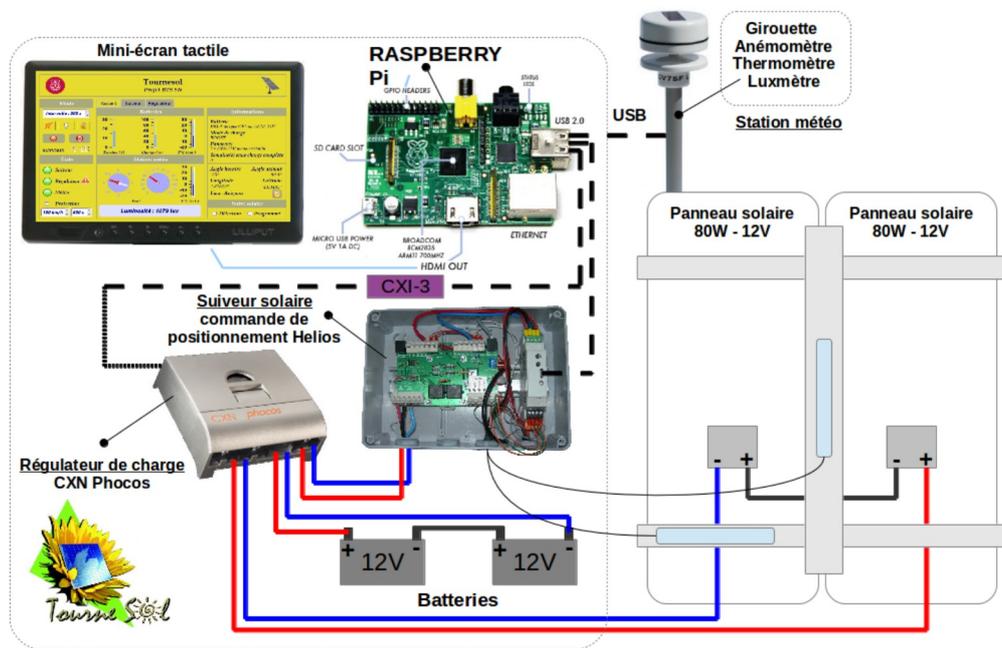
Tournesol	Version 2.1
Auteur : Adrien Thiel	39/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

VII) Planification



VIII) Matériel



Tournesol	Version 2.1
Auteur : Adrien Thiel	40/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Raspberry Pi modèle B :

Le Raspberry Pi est un nano-ordinateur mono-carte à processeur **ARM** conçu par le créateur de jeux vidéo David Braben, dans le cadre de sa fondation Raspberry Pi.

Cet ordinateur, qui a la taille d'une carte de crédit, est destiné à encourager l'apprentissage de la programmation informatique ; il permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux et des logiciels compatibles. Il est fourni nu (carte mère seule, sans boîtier, alimentation, clavier, souris ni écran) dans l'objectif de diminuer les coûts et de permettre l'utilisation de matériel de récupération.



Phocos-CX20 :

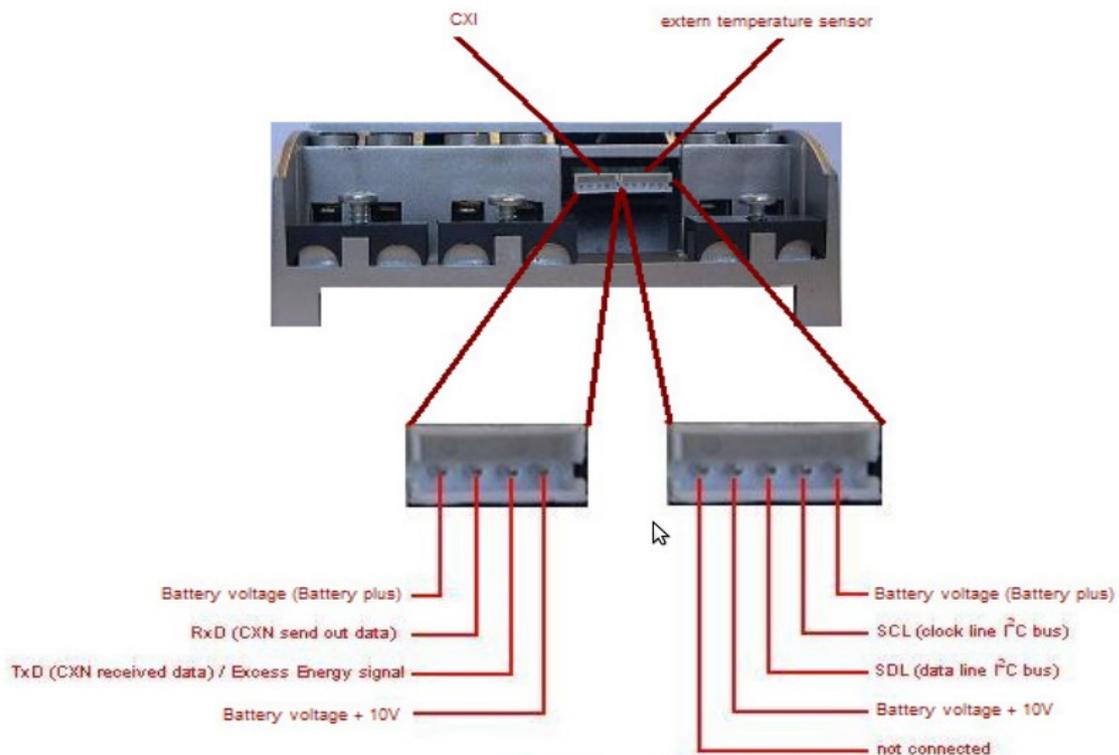
- Le contrôleur de charge phocos-CX20 protège du risque de surcharge par le générateur solaire, et décharge profonde par les charges. Les caractéristiques de chargement comportent plusieurs étapes qui incluent l'adaptation automatique à la température ambiante.
- Le contrôleur de charge s'ajuste automatiquement à la tension du système en 12V ou 24V.
- La touche sur la partie du dessus permet d'allumer ou d'éteindre le chargement.
- Le contrôleur de charge peut être programmé pour des applications d'éclairage.
- Le contrôleur fournit une sortie de contrôle pour les charges spéciales utilisant le surplus d'énergie.
- Il y a un interface série qui peut être utilisé avec un adaptateur de connexion optionnel (CXI pour le cas présent).
- Le contrôleur de charge est équipé de fonctions de sécurité et d'affichage.

Tournesol	Version 2.1
Auteur : Adrien Thiel	41/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016



Interface de communication CXI :



L'interface CXI est reliée à la Raspberry Pi par une liaison **USB**. Cette liaison USB sera pris en charge comme un port série virtuel par le système d'exploitation de la Raspberry PI.

Tournesol	Version 2.1
Auteur : Adrien Thiel	42/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Le paramétrage du port série sera le suivant :

- Débit : 9600 bits/s
- Données : 8 bits
- Parité : aucune
- Nombre de bits de Stop : 1

*Remarque : La communication avec le régulateur se fera via la classe **Port** fournie.*

Batteries :

2x Batteries AGM étanche (VRLA) 12V/14Ah de Victron Energy (41,82 euros TTC l'unité) montées en série.



VRLA est l'abréviation de « Valve Regulated Lead Acid », ce qui signifie que la batterie est étanche. Du gaz s'échappera par des soupapes de sécurité uniquement en cas de surcharge ou de défaillance d'éléments. Les batteries VRLA sont sans entretien à vie.

AGM est l'abréviation de « Absorbent Glass Mat ». Dans ces batteries, l'électrolyte est absorbé par capillarité dans une natte en fibre de verre placée entre les plaques. Les batteries AGM sont plus aptes à fournir des courants très élevés pendant de courtes durées (démarrage) que les batteries Gel.

Mini-écran :

Il prend en charge chaque entrée vidéo commune sur le marché (HDMI , DVI , VGA et vidéo composite) , et offre une variété de méthodes de montage utiles (VESA 75 , ¼ fil de BSW et le montage de fente verticale). En outre, il est alimenté par 12V , ce qui rend adaptable pour fonctionner sur secteur (adaptateur fourni) , des batteries , ainsi que d'être une solution dans un véhicule .

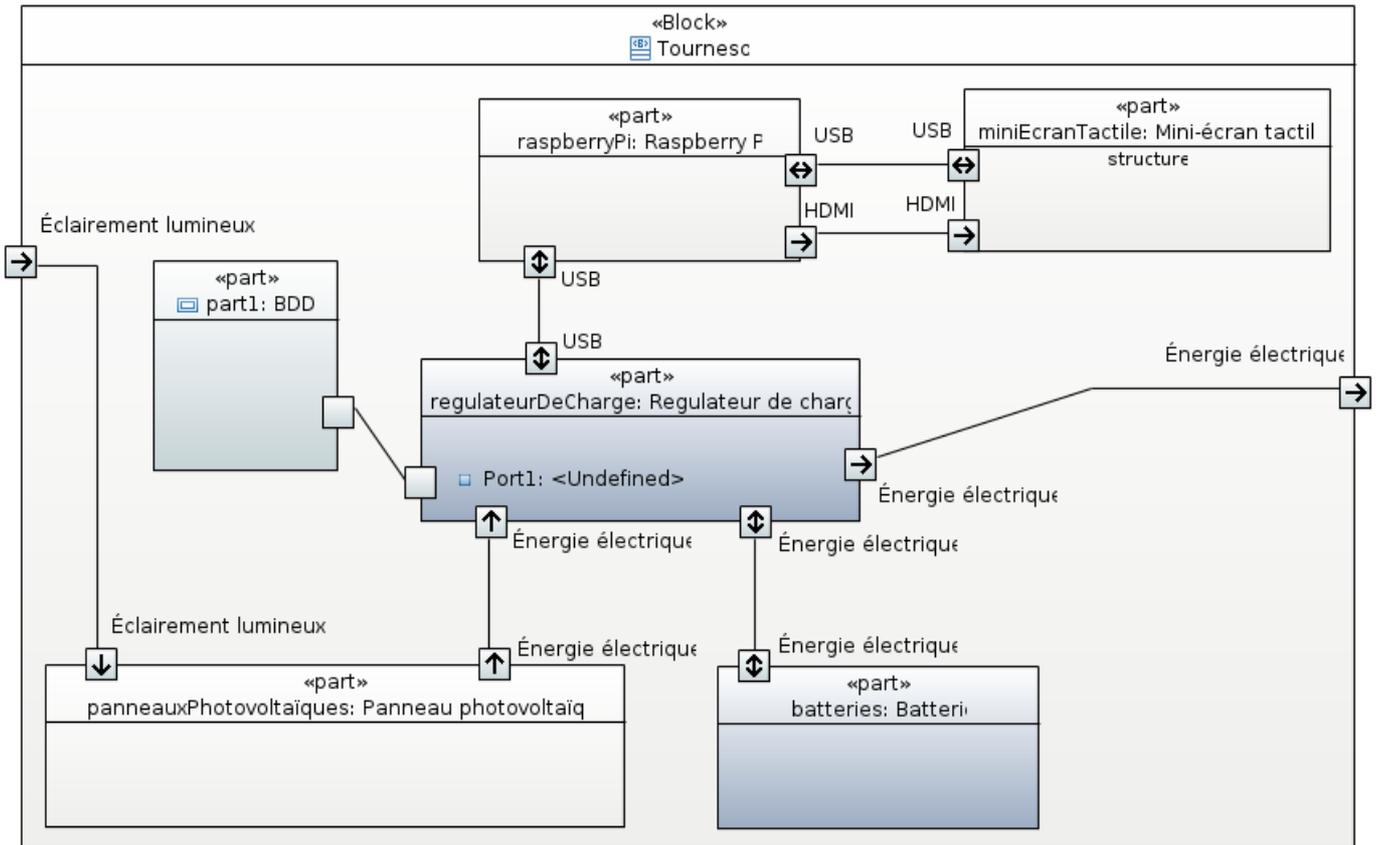


Tournesol	Version 2.1
Auteur : Adrien Thiel	43/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Diagramme de blocs internes :

Sur le diagramme IBD ci-dessous nous pouvons voir l'interaction entre les différents matériels (RPI, mini-écran, Phocos, batterie et panneaux photovoltaïques) :

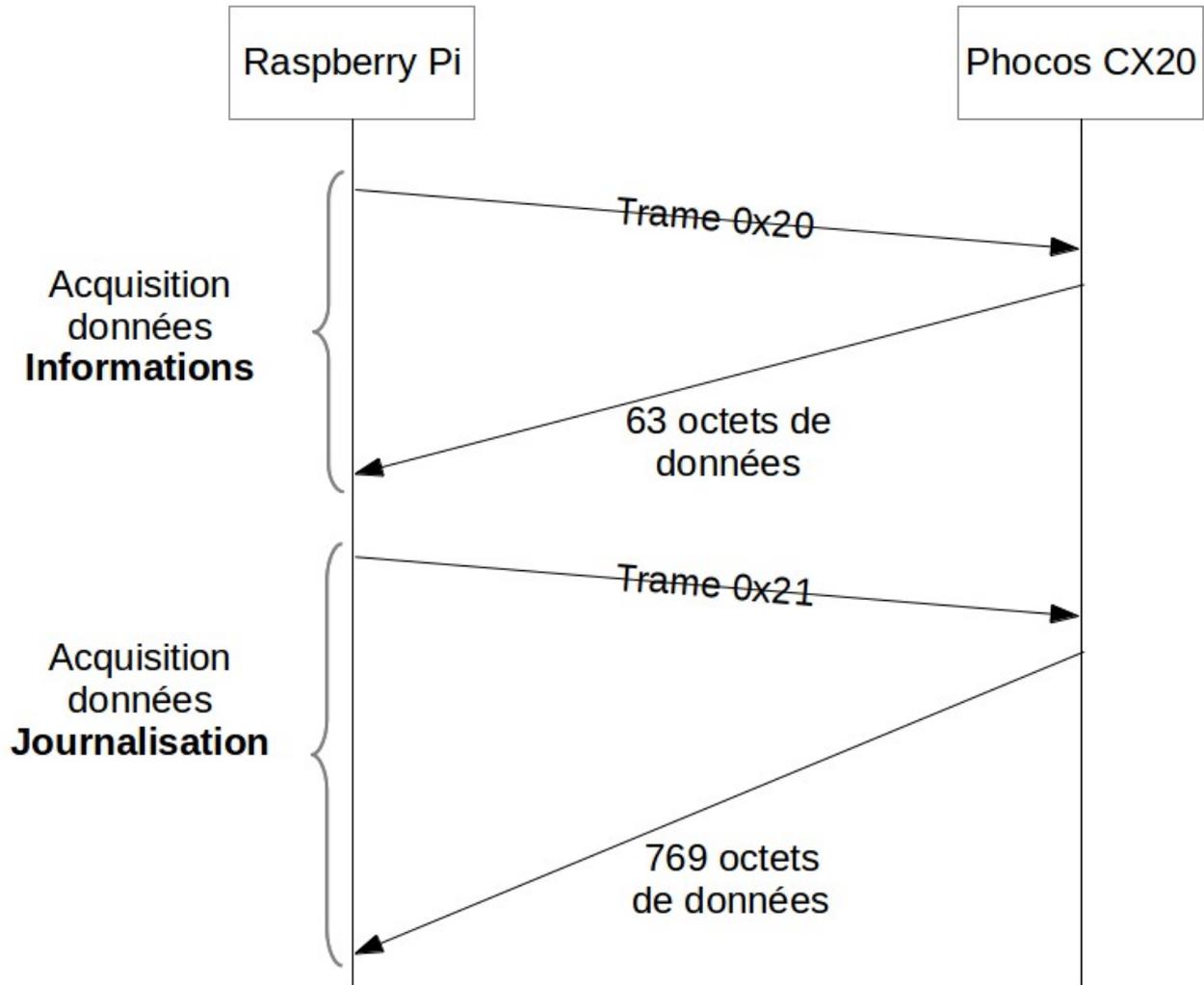


Tournesol	Version 2.1
Auteur : Adrien Thiel	44/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

IX) Communication avec le Phocos-CX20 :

La communication entre la raspberry et le phocos peut être représenté comme ceci :



Tournesol	Version 2.1
Auteur : Adrien Thiel	45/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Décodage des trames :

La trame d'information est toujours composé de caractères en ASCII qui représentent des nombres décimaux : ex : 30 36 34 → 0 6 4 } valeur en décimal correspondant à la version dans la trame d'information.

Les trame d'historique est toujours composé de caractères ASCII qui représentent des valeur en hexadécimal : ex : menu state : 30 30 39 → '0' '0' '9' => 0x09 valeur en hexadécimal du menu state de la trame de journalisation.

Les valeurs sont toujours séparé par le caractère « 0x20 » qui correspond à l'espace.

Le caractère ASCII qui est envoyer en premier correspond toujours au bit de poids fort. Dans la trame d'historique, une valeur peut être composé de deux, trois ou six parties. La partie 1 a toujours le bit de poids fort. Le premier caractère de la trame d'historique est « 0x21 » qui correspond à « ! ».

Remarque : le code ASCII de l'espace est 0x20 en hexadécimal soit 32 en décimal.

Trame d'Information :

Nous recevons la trame sous cette forme là :

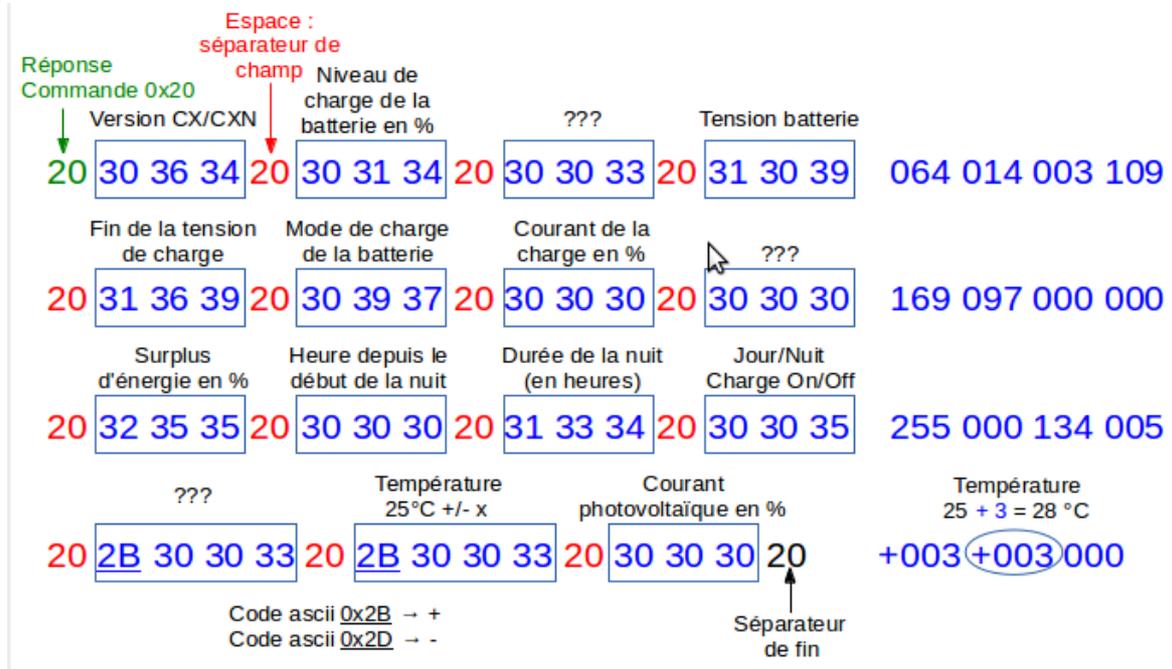
20 30 36 34 20 30 31 34 20 30 30 33 20 31 30 39	064 014 003 109
20 31 36 39 20 30 39 37 20 30 30 30 20 30 30 30	169 097 000 000
20 32 35 35 20 30 30 30 20 31 33 34 20 30 30 35	255 000 134 005
20 2B 30 30 33 20 2B 30 30 33 20 30 30 30 20	+003 +003 000

On peut voir sur la partie de droite, la trame en ASCII et la partie de gauche en hexadécimale. C'est donc la partie en hexadécimale que nous allons traiter.

Dans la partie en ASCII le délimiteur de champs est « 0x20 » ce qui correspond à l'espace dans la partie en hexadécimale. C'est grâce à ces délimiteurs de champs que l'on peut identifier les différents champs qui correspondent aux données que je doit décoder. C'est donc après l'identification des champs que le décodage peu commencer.

Tournesol	Version 2.1
Auteur : Adrien Thiel	46/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016



Après avoir identifié les différents champs il faut maintenant effectuer certain calculs pour avoir la bonne valeur comme par exemple pour la tension de la batterie (champ numéro 4 de la trame d'information) il faut effectuer le calcul suivant :

$$\text{valeur réelle} = ((0,032 * \text{valeur lu}) + 9) * 2$$

Tournesol	Version 2.1
Auteur : Adrien Thiel	47/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Trame de Journalisation :

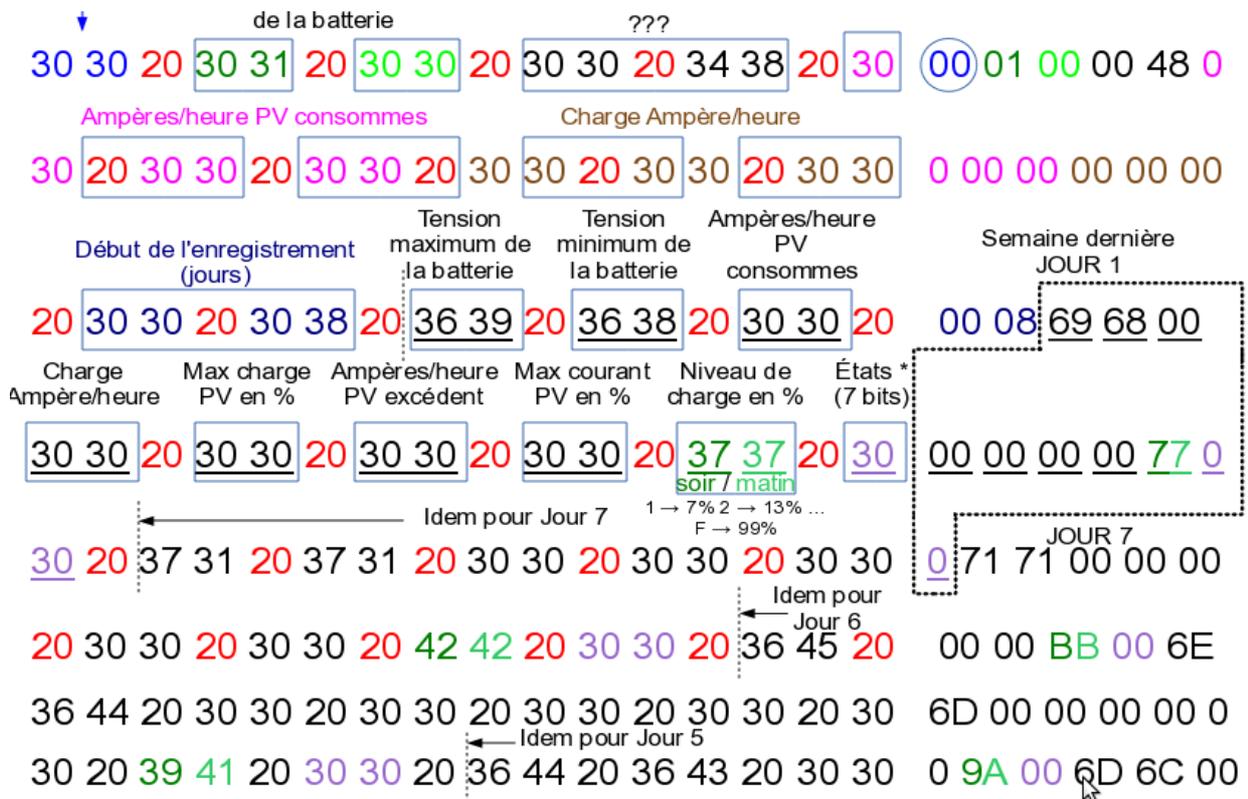
Dans le même principe que pour la trame d'information, La trame d'historique nous parvient comme ceci :

```

21 45 31 20 30 32 20 31 41 20 45 36 20 30 30 20 !E1 02 1A E6 00
30 30 20 46 41 20 39 30 20 46 46 20 46 46 20 30 00 FA 90 FF FF 0
32 20 30 32 20 30 32 20 36 38 20 36 45 20 46 46 2 02 02 68 6E FF
20 46 46 20 46 46 20 46 46 20 34 30 20 30 30 20 FF FF FF 40 00
30 30 20 30 31 20 30 30 20 30 30 20 34 38 20 30 00 01 00 00 48 0
30 20 30 30 20 30 30 20 30 30 20 30 30 20 30 30 0 00 00 00 00 00
20 30 30 20 30 38 20 36 39 20 36 38 20 30 30 20 00 08 69 68 00
30 30 20 30 30 20 30 30 20 30 30 20 37 37 20 30 00 00 00 00 77 0
30 20 37 31 20 37 31 20 30 30 20 30 30 20 30 30 0 71 71 00 00 00
20 30 30 20 30 30 20 42 42 20 30 30 20 36 45 20 00 00 BB 00 6E

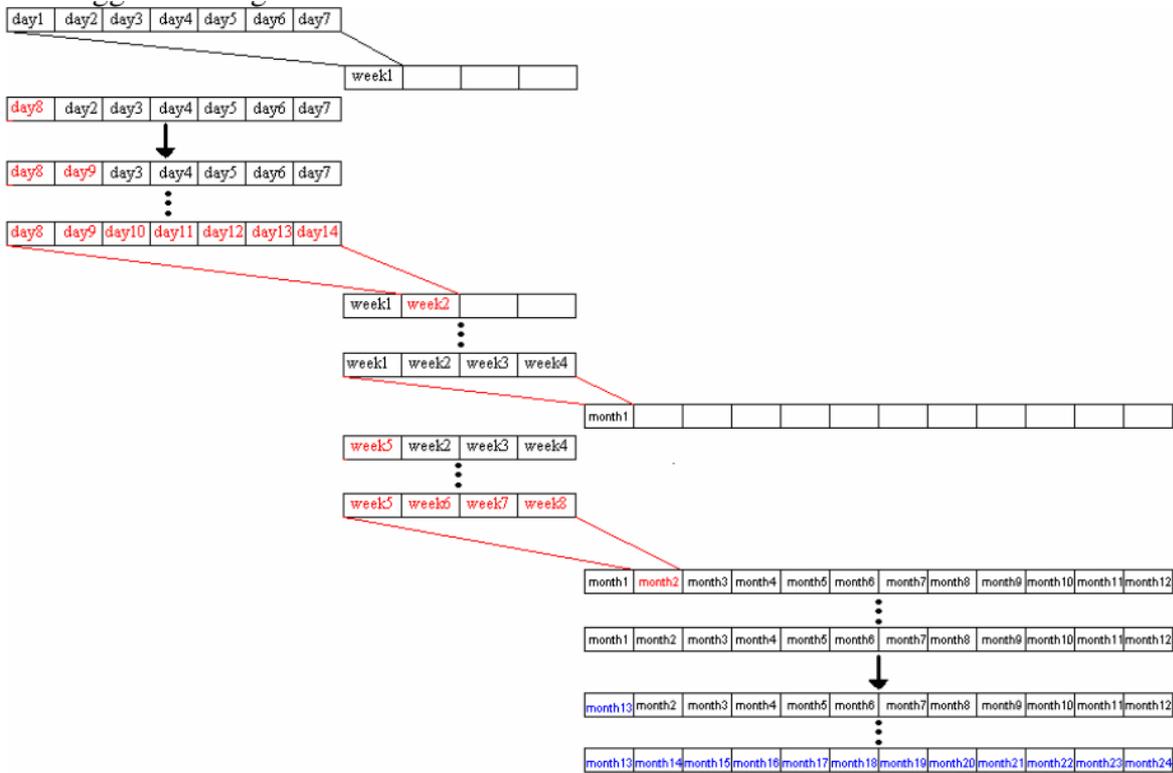
```

Nous devons donc identifier les différents champs de cette trame. Cela nous est permis grâce au caractère séparateur de champs qui est l'espace comme nous l'avons dit précédemment :



Tournesol	Version 2.1
Auteur : Adrien Thiel	48/101

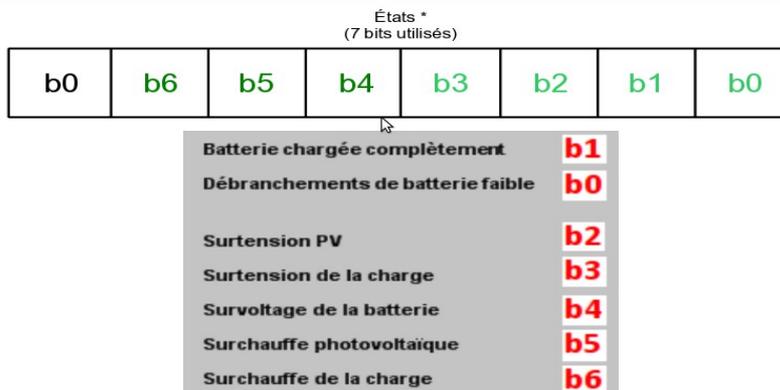
La journalisation peut être représentée de la façon suivante :



Les alertes ou « Etats » :

Le champs état est sur 7 bits, il y a 7 états différents. Pour identifier ces différents états nous faisons un masque : par exemple pour l'état b0 il faut que le champs état soit à 0000 0001 soit 0x01.

Ensuite il suffit de faire un masque sur chaque état pour savoir quelle(s) alerte(s) doit être déclencher.

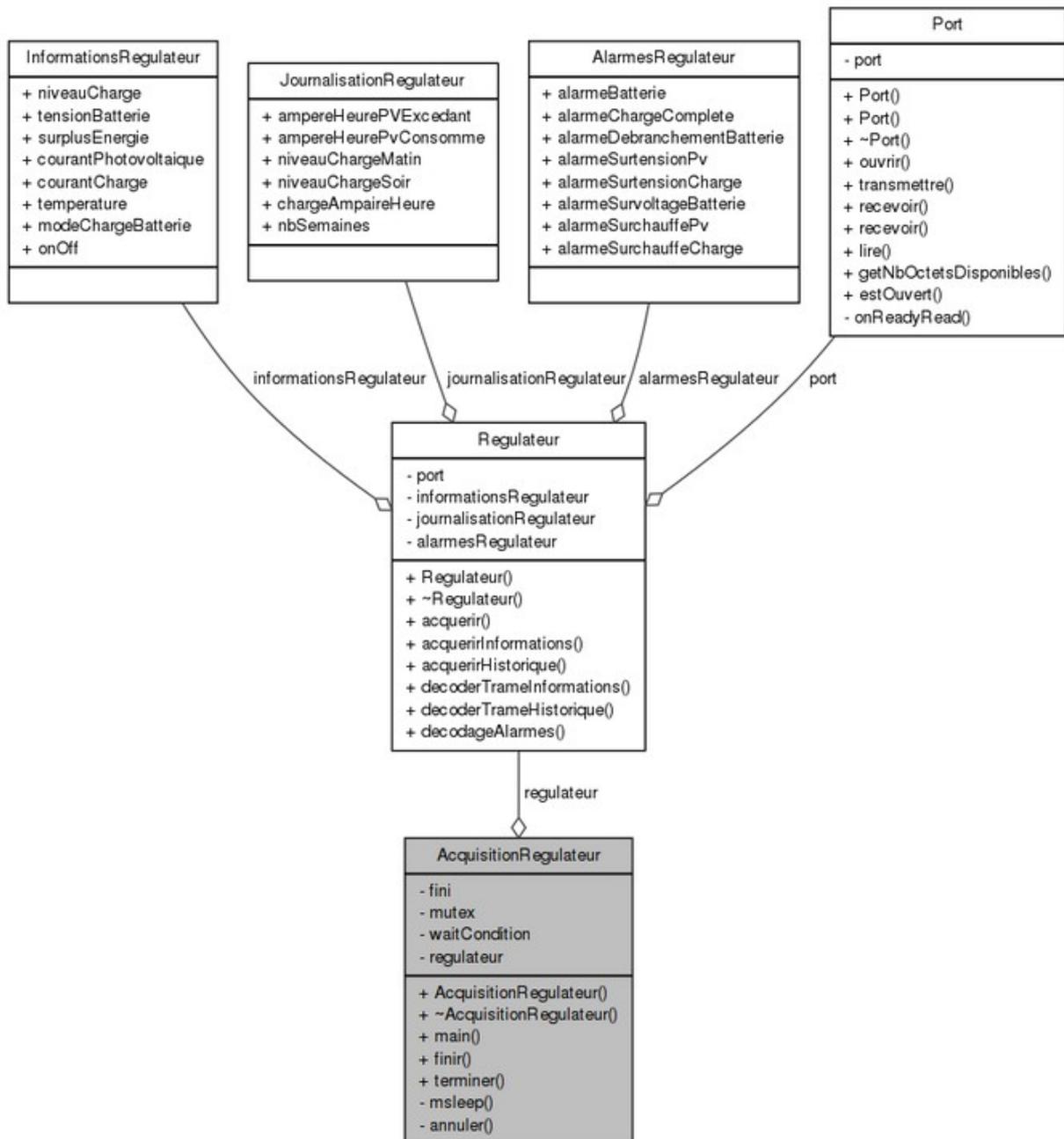


Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

X) Partie logicielle :

Ma partie logicielle est composée de deux classes :

- **AcquisitionRegulateur**
- **Regulateur**



Tournesol	Version 2.1
Auteur : Adrien Thiel	50/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Remarque : la classe **Regulateur** a besoin de la classe **Port** fournie pour communiquer avec le Phocos.

Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

AcquisitionMeteo	Classe qui assure la gestion de la station météo
AcquisitionRegulateur	Thread contenant toutes les données relatives au régulateur de charge
AcquisitionSuiveur	Gestion du suiveur solaire
AlarmesRegulateur	Regroupe les différents etats du regulateur
AsservissementSuiveur	Gestion du capteur solaire pour le suivi
BaseDeDonnees	Classe de gestion de base de données MySQL
BusI2C	Gestion du bus I2C pour les circuits LTC2309 et BH1750
CapteurLuminosite	Classe de gestion du capteur de luminosité
CapteurSolaire	Classe qui assure la détection de la position du soleil
CV7	Gestion de la récupération des phrases, découpe et stockage à partir de la station météo
IhmTournesol	Gestion de l'interface graphique
InformationsRegulateur	Regroupe les variable contenant les informations relatives a la trame d'Informations
InformationsStationMeteo	Regroupe les variables contenant les informations relatives à la trame d'information
InformationsSuiveur	Regroupe les données envoyées par le suiveur solaire
JournalisationRegulateur	Regroupe les variable contenant les informations relatives a la trame de Journalisation
Mesure	Regroupe les différentes informations relatives à une donnée
Port	Classe de gestion d'un port série
Regulateur	Gestion de l'acquisition et du traitement des données du régulateur de charge phocos CX20
StationMeteo	Gestion de la récupération des informations sur la CV7 et le capteur lumineux, affichage dans l'IHM e tstockage dans la base de données
SuiveurSolaire	Communication avec le système de positionnement

Tournesol	Version 2.1
Auteur : Adrien Thiel	51/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Scénario « Acquérir les données du régulateur de charge » :

La classe « **AcquisitionRegulateur** » appelle une méthode de la classe « **Regulateur** » cette méthode est « **acquérir()** » qui appelle deux méthodes de la classe régulateur:

-**acquérirInformations()** : cette méthode gère l'acquisition de la trame d'information :
 -Elle appelle la méthode **transmettre()** de la classe port pour émettre la requête « » (0x20) pour faire la demande de la trame d'information ;
 -Puis elle appelle la méthode **recevoir()** de la classe port afin de recevoir la trame en question ;
 -Ensuite elle appelle la méthode **decoderTrameInformations()** afin de traiter cette trame et d'en extraire puis de calculer les données qu'elle contient.

-**acquérirHistorique()** : cette méthode gère l'acquisition de la trame de journalisation :
 -Elle appelle la méthode **transmettre()** de la classe port pour émettre la requête « ! » (0x21) pour faire la demande de la trame de journalisation ;
 -Puis elle appelle la méthode **recevoir()** de la classe port afin de recevoir la trame en question ;
 -Ensuite elle appelle la méthode **decoderTrameHistorique()** afin de traiter cette trame et d'en extraire puis de calculer les données qu'elle contient ;
 -Elle appelle aussi la méthode **decodageAlarmes()** afin de traiter le champ « Etats » de la trame de journalisation qui ne contient que les données relatives au « alertes » du régulateur comme par exemple si la batterie est chargée complètement.

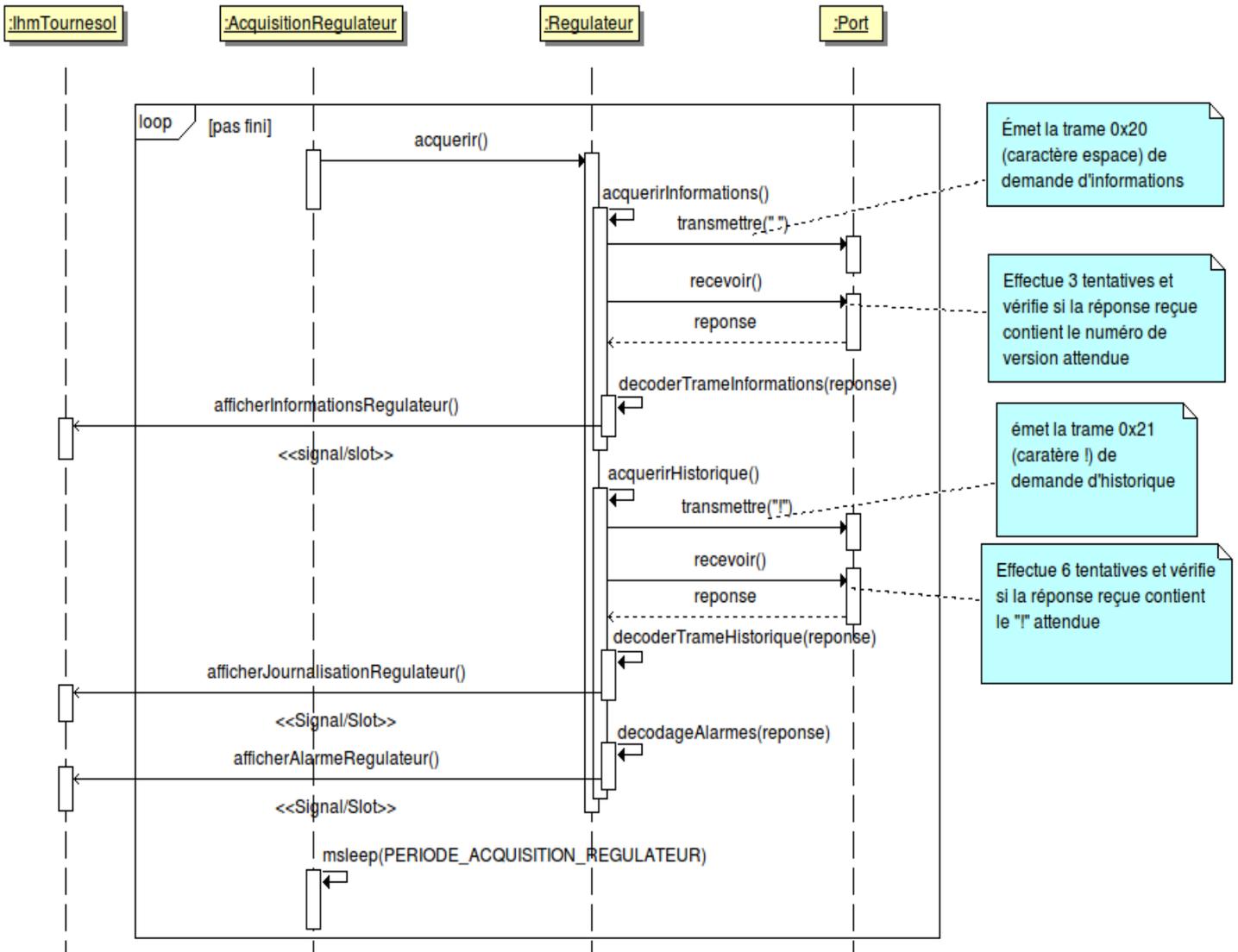
Les données recueillies par les méthodes de la classe *Regulateur* sont ensuite envoyés vers l'ihm par un système de signal/slot sous la forme de structures. Nous avons donc trois structures différentes :

-**InformationsRegulateur** : contient les données décodées par la méthode *decoderTrameInformations()* qui sont uniquement les données relatives à la trame 0x20.

-**JournalisationRegulateur** : contient les données décodées par la méthode *decodéTrameHistorique()* qui sont uniquement les données relatives à la trame 0x21.

-**AlarmesRegulateur** : contient les données décodées par la méthode *decodageAlarmes()* qui sont uniquement les données relatives aux alarmes qui sont contenues par le champ « etats » de la trame 0x21.

Tournesol	Version 2.1
Auteur : Adrien Thiel	52/101



Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

IHM :

Dans l'ihm, Nous retrouvons les valeurs décodé par nos méthodes, par exemple dans l'onglet « Régulateur » nous retrouvons les informations suivantes :



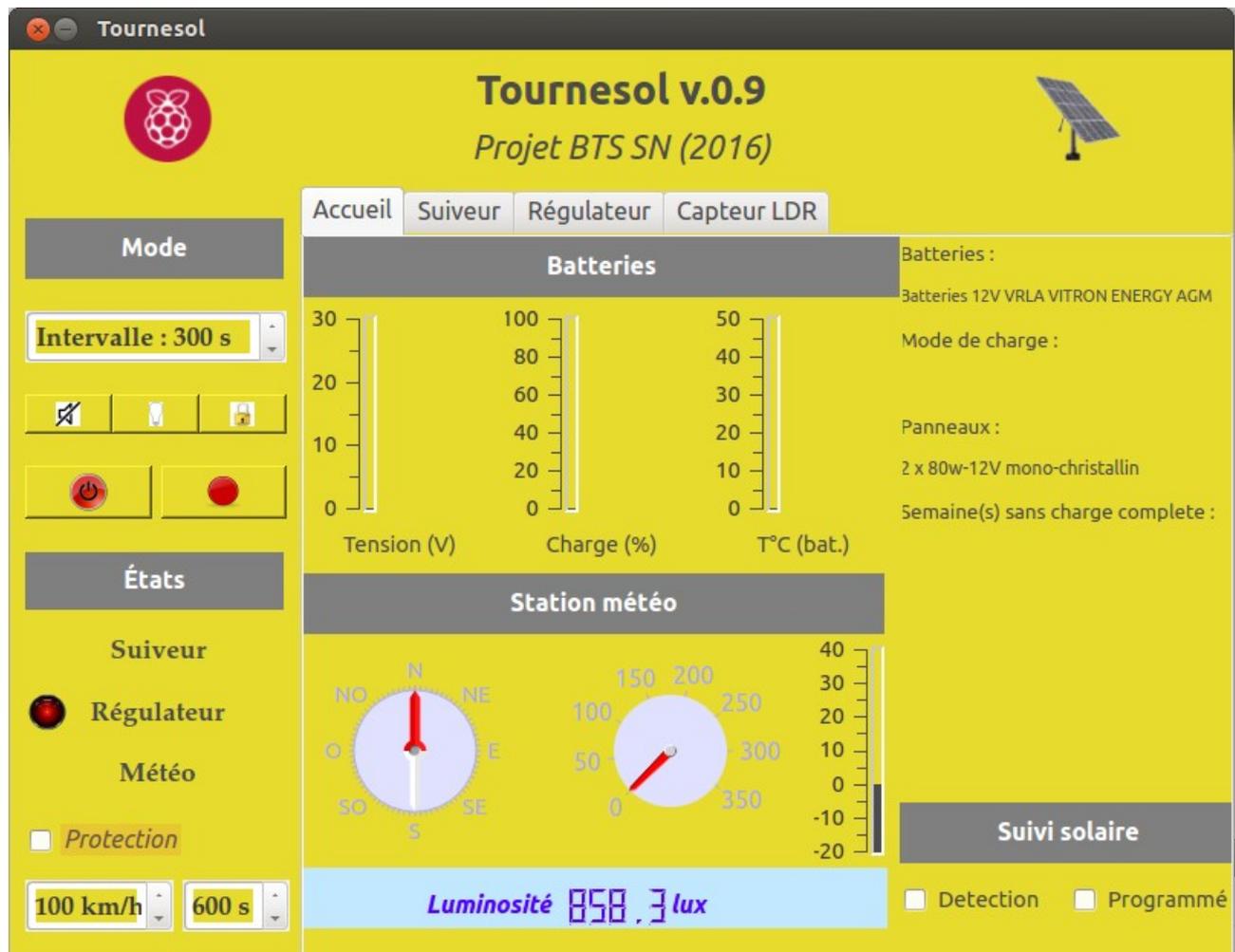
Dans l'onglet « Informations » on peut voir apparaître les données du régulateur de charge (La tension de la batterie, le niveau de charge, ...)

Dans l'onglet « Alerte » on peut voir les différentes alertes détecté par le régulateur de charge (voir manuel d'utilisation pour plus d'explications)

Tournesol	Version 2.1
Auteur : Adrien Thiel	54/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Les informations importantes sont affichées dans l'onglet d'accueil :



Dans l'onglet « Batteries » nous avons les trois informations les plus importantes :

- Tension des batteries ;
- Niveau de charge ;
- Température des batteries .

Et dans l'onglet en haut à droite toutes les informations nécessaires sur les batteries (type de batteries, mode de charge, ...).

Tournesol	Version 2.1
Auteur : Adrien Thiel	55/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

XI) Test de unitaires :

Méthode acquérir() :

Ce que la méthode doit faire	Ce que la méthode fait	Validation (oui/non)
Utiliser la méthode transmettre() de la classe port pour transmettre la commande 0x20.	Utilise la méthode transmettre() de la classe port pour transmettre la commande 0x20.	oui
Utiliser la méthode recevoir() de la classe port pour recevoir la trame 0x20 du phocos-cx20.	Utilise la méthode recevoir() de la classe port pour recevoir la trame du phocos-cx20.	oui
Utiliser la méthode decoderTrameInformation() et décoder la trame reçu.	Utilise la méthode decoderTrameInformation() et décode la trame reçu.	oui
Transmettre les informations vers le BDD.	non	non
Utiliser la méthode transmettre() de la classe port pour transmettre la commande 0x21.	Utiliser la méthode transmettre() de la classe port pour transmettre la commande 0x21.	oui
Utiliser la méthode recevoir() de la classe port pour recevoir la trame 0x21 du phocos-cx20.	Utilise la méthode recevoir() de la classe port pour recevoir la trame du phocos-cx20.	oui
Utiliser la méthode decoderTrameJournalisation() et décoder la trame reçu.	Utiliser la méthode decoderTrameJournalisation() et décoder la trame reçu.	oui
Transmettre les données de journalisation vers le BDD.	non	non

Tournesol	Version 2.1
Auteur : Adrien Thiel	56/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Méthode decoderTrameInformation() :

Ce que la méthode doit faire	Ce que la méthode fait	Validation (oui/non)
Identifier les différents champs de la trame d'information.	Identifie les différents champs de la trame d'information.	oui
Effectuer le traitement (décodage) des données recueillis.	Effectue le traitement (décodage) des données recueillis.	oui
Transmettre les données décodé à l'IHM.	Transmet les données décodé vers l'ihm	oui

Méthode decoderTrameJournalisation() :

Ce que la méthode doit faire	Ce que la méthode fait	Validation (oui/non)
Identifier les différents champs de la trame d'information.	Identifie les différents champs de la trame d'information.	oui
Effectuer le traitement de la journalisation hebdomadaire.	Effectue le traitement de la journalisation hebdomadaire.	oui
Effectuer le traitement de la journalisation mensuelle.	non	non
Effectuer le traitement de la journalisation annuelle.	non	non
Transmettre les données décodé à l'IHM.	Transmet les données décodé vers l'ihm	oui

Méthode decodageAlarmes() :

Ce que la méthode doit faire	Ce que la méthode fait	Validation (oui/non)
Récupérer le champs « etats » de la trame 0x21	Récupérer le champs « etats » de la trame 0x21	oui
Décoder les différentes alarmes	Décode les différentes alarmes	oui
Transmettre les données décodé à l'IHM.	Transmet les données décodé vers l'ihm	oui

Tournesol	Version 2.1
Auteur : Adrien Thiel	57/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

XII) Manuel d'utilisation :

Si dans l'ihm vous voyez apparaître un voyant rouge à coté de Régulateur dans l'onglet « Etat », cela veut dire que le logiciel n'a pas réussi à avoir une trame correcte au bout de 3 essais pour la trame d'information et au bout de 6 essais pour la trame de journalisation :



Si le voyant est vert cela veut dire que les trames sont reçu et traités normalement :



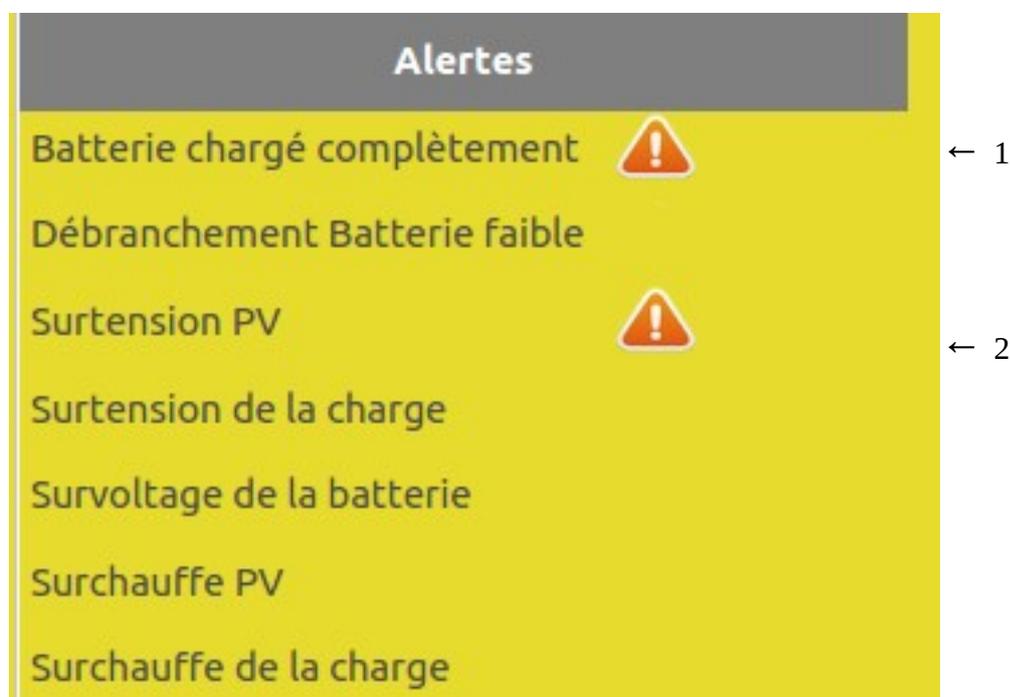
Tournesol	Version 2.1
Auteur : Adrien Thiel	58/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Ensuite pour les alertes, elles s'affiche dans l'onglet Régulateur, dans la partie « Alerte ».
Les différentes alertes sont afficher et si l'alerte est déclenché, en face de l'alerte s'affiche ceci :



par exemple dans le cas ci dessous :



Cela veut dire que :

- 1 : les batteries sont chargées complètement ;
- 2 : il y a une surtension des batteries.

Tournesol	Version 2.1
Auteur : Adrien Thiel	59/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Lorsqu'il y a au moins une alerte d'activé, une notification apparaît dans l'onglet « Etat » à droite de Régulateur avec le même logo que pour les alertes :



<= c'est à dire celui-ci.

Tournesol	Version 2.1
Auteur : Adrien Thiel	60/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

IV-) Contrat individuel étudiant n°3 (Nicolas Detilleux) : **Module de gestion de la station météo**

I) Objectifs :

Vous intervenez en tant que technicien SN-IR pour “Concevoir une partie du système informatique Tournesol”. Vous avez la charge du “**Module de gestion de la station météo**” qui permet d’acquérir des données, de les archiver dans une base de données et de communiquer avec la station météo. Le mini écran tactile permettra de visualiser les informations principales du module. Vous devez assurer la protection des panneaux contre le vent violent.

II) Fonctionnalités en charge :

Ref.	Fonction de service	Critère	Niveau	Flexibilité
FP-5	Surveiller les alarmes et protéger	Module à surveiller	station météo	F0
		Type d’alarme	visuelle	F0
		Périodicité de la surveillance	30 secondes	F0
		Protection automatique	contre le vent	F0
FP-6	Paramétrer le fonctionnement	Sélection de la protection contre le vent	par l’écran tactile	F0
		Réglage de la vitesse de vent maximale	en km/h (par défaut 90 km/h)	F0
		Réglage du temps d’attente avant retour normal	en s (par défaut 600 s)	F2
FP-7	Acquérir les données	Périodicité	configurée par fichier	F0
		Module	station météo	F0
FP-8	Visualiser les états et données	Données	avec les unités	F0
		États	sous forme d’icône	F0
		Format horodatage	jj/mm/aaaa hh:mm	F0
		Rafraîchement	1 seconde	F0
FP-9	Archiver les états, données et alarmes	Base de données	SQL	F0
		Horodatage mesures	aaaa-mm-jj hh:mm:ss	F0

Tournesol	Version 2.1
Auteur : Nicolas Detilleux	61/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

III) Production attendue :

- Une application informatique fonctionnelle
- Un modèle UML complet de la partie à développer
- Le code source commenté de l'application
- Les documentations prévues au paragraphe « Exigences sur la livraison »

IV) Recette minimale étudiant 3 :

- Les valeurs de la vitesse et direction du vent ainsi que la température de l'air et la luminosité sont affichées périodiquement et graphiquement sur l'écran d'accueil et archivées dans la base de données
- L'état de la station est visible sur le mini écran tactile
- Le réglage des consignes de protection contre les vents violents est réalisable et pris en compte
- La protection contre le vent violent est effective

V) Présentation du travail de l'étudiant 3 :

La partie du travail que je dois réaliser est la partie gestion de la météo. Je dois pouvoir recevoir et trier les trames venant de la station météo CV7, pour en extraire la température de l'air, ainsi que la direction et la force du vent.

D'autre part je dois pouvoir joindre le capteur de luminosité BH1750 sur le bus I2C, et de la même manière récupérer dans ces trames la luminosité ambiante. Ces données doivent être enregistrées dans la base de données, affichées sur l'IHM.

Je dois aussi pouvoir récupérer des informations de configuration depuis l'IHM pour la partie protection vent. Si la protection vent est activée, je dois donc aussi vérifier la force du vent et la comparer à la force entrée par l'utilisateur, et si celle-ci est supérieur placer les panneaux dans une position de sécurité.

Tournesol	Version 2.1
Auteur : Nicolas Dettleux	62/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

VI) Tâches à réaliser :

Tâches à réaliser	Production attendue	Estimation horaire
S'approprier le cahier des charges	Le cahier des charges est expliqué. Les tâches à réaliser sont identifiées et les ressources sont définies	16h
Rechercher des solutions issues de l'innovation technologique pour les stations météo	Une étude comparative argumentée des différentes solutions est produite.	6h (SCP)
Installer et configurer son environnement de développement	Le poste de développement est opérationnel.	2h
Installer et raccorder la station météo	Les appareils fonctionnent. La procédure d'installation a été respectée. Un compte rendu est rédigé	4h (SCP)
Relier et paramétrer les interfaces de communication	Les interfaces de communications sont correctement paramétrées et fonctionnelles. Un plan de câblage est réalisé. Le rapport de tests de mise en œuvre est rédigé.	4h
Connecter le capteur de luminosité	La communication avec le capteur est possible. Un plan de câblage est rédigé.	2h (SCP)
Etudier et documenter les caractéristiques des capteurs	Les capteurs sont caractérisés et des mesures réalisées.	6h (SCP)
Mettre en œuvre les programmes de test fournis	Le rapport de tests est renseigné	12h + 4h (SCP)
Finaliser la modélisation UML du module	Les diagrammes UML (diagrammes de séquence « régler la protection contre le vent » à compléter, diagramme d'état de la protection contre le vent) sont élaborés et finalisés.	20h
Produire la maquette de l'IHM du module	La maquette de l'IHM correspond aux exigences du cahier des charges.	12h
Coder et tester les classes du module	Les classes <i>StationMeteo</i> , <i>CV7</i> , <i>CapteurLuminosite</i> , <i>ProtectionVent</i> , <i>AcquisitionMeteo</i> et la structure <i>DonneesMeteo</i> sont codés et valides. La classe	40h

Tournesol	Version 2.1
Auteur : Nicolas Dettleux	63/101

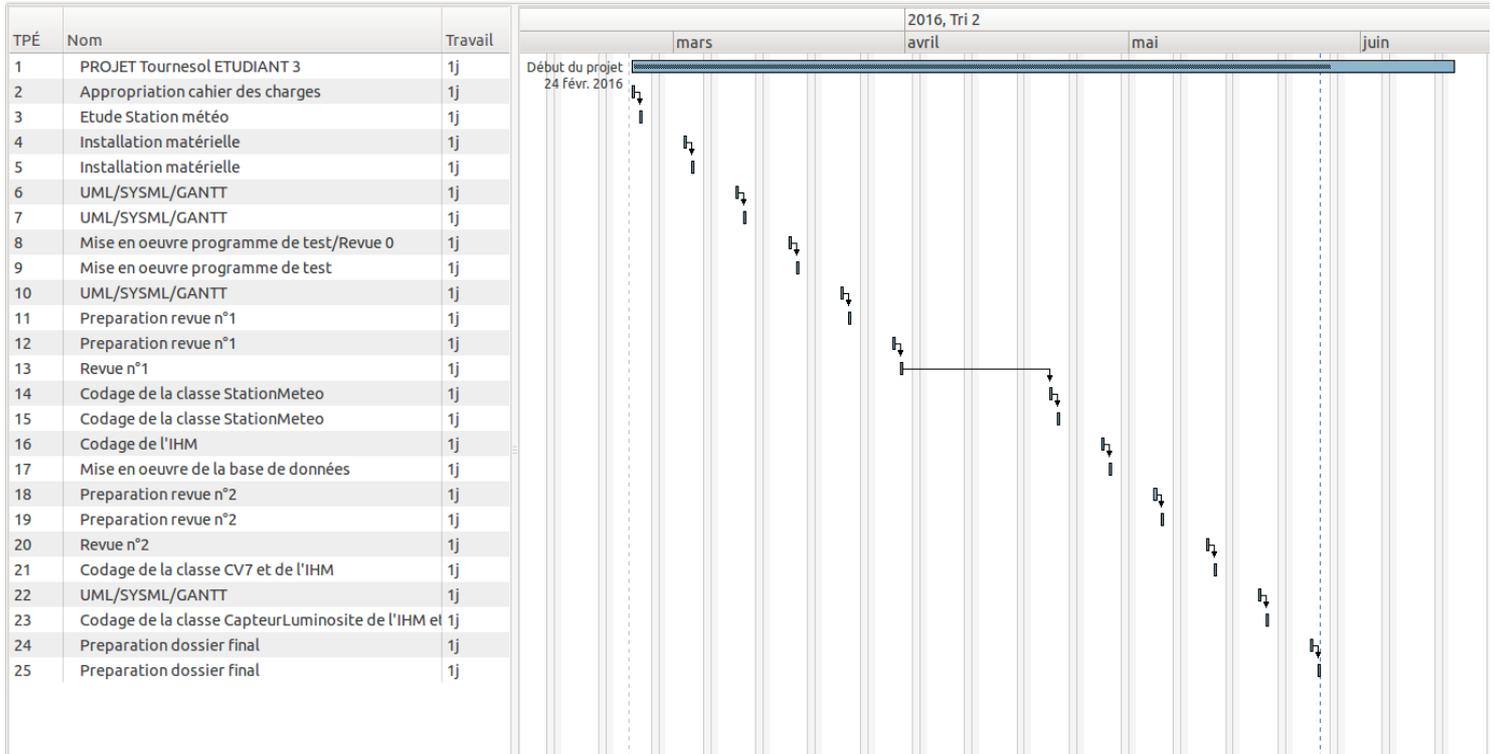
Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

	<i>IHM Tournesol</i> est complétée et valide.	
Réaliser les tests unitaires	Les tests unitaires des classes <i>StationMeteo</i> , <i>CV7</i> , <i>CapteurLuminosite</i> , <i>ProtectionVent</i> et <i>AcquisitionMeteo</i> sont écrits et archivés.	16h
Faire la recette du module	Le cahier de recette du module est validé.	4h + 4h (SCP)
Intégrer en équipe l'application complète	L'application est intégrée et fonctionnelle	4h
Rédiger le dossier technique et les documents relatifs au projet	Le dossier est rédigé en respectant les exigences	30h
Produire un guide de mise en route et d'utilisation du module	Un manuel est fourni.	10h
Gérer la planification	Le planning prévisionnel est établi. Le planning est actualisé avec une mise en évidence des écarts par rapport au prévisionnel.	10h
Total		200h

Tournesol	Version 2.1
Auteur : Nicolas Detilleux	64/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

VII) Planification :



Tournesol	Version 2.1
Auteur : Nicolas Detilleux	65/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

VIII) Matériel :



CV7 :

La CV7 est la station météo. Elle mesure la direction et la force du vent grâce à des ultrasons. Le protocole de communication est le protocole NMEA 0183, La station météo se relie à la raspberry pi par USB. Elle reçoit son alimentation par ce même USB, soit 5V.

Plage de mesures :

- Vitesse du vent de 0,9 à 144km/h
- Orientation relative du vent de 0 à 359°
- température de -15°C à +55°C

Caractéristiques du capteur :

- Résolution vitesse du vent : 0,1m/s
- Résolution direction du vent : 1°
- Précision de la vitesse du vent : $\pm 5\%$ de la vraie
- Précision de l'angle du vent : $\pm 3^\circ$ pour une vitesse supérieur à 7m/s
- Résolution température : 1°C
- Vitesse de transmission : 4800 bits/s
- Phrases NMEA 0183 utilisées :
 - Données vent – MWV – 2 fois pa seconde
 - Données température – XDR – 2 fois par seconde

Tournesol	Version 2.1
Auteur : Nicolas Detilleux	66/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

BH1750 :

Le capteur BH1750 est le capteur de luminosité. Il communique avec la raspberry via un bus I2C.

Plage de luminosité : 1 à 65535 lux

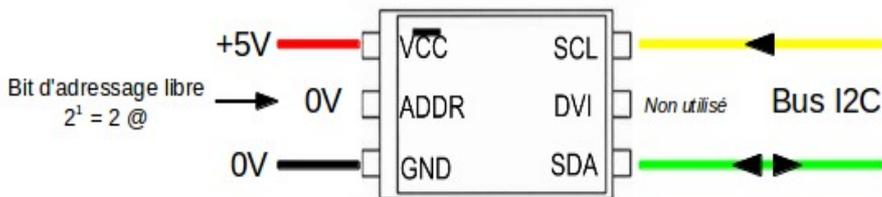
Luminosité convertie sur 16 bits

Mode de mesure :

- période 120ms / résolution 0,5lux
- période 120ms / résolution 1lux
- période 16ms / résolution 4lux



BH1750 Capteur luminosité I2C



Adresse I2C = 0x23

Plage de luminosité : 1 lux à 65535 lux
Pas : 0.5 lux à 4 lux (au choix)
Luminosité convertie sur 16 bits
3 modes de résolution au choix
Conversion : moins de 120 ms

Acquérir une mesure de luminosité

1. Lecture des données numériques

MSB ← lire 1 octet sur le bus I2C

LSB ← lire 1 octet sur le bus I2C

2. Conversion en lux

luminosite ← (MSB << 8) + LSB

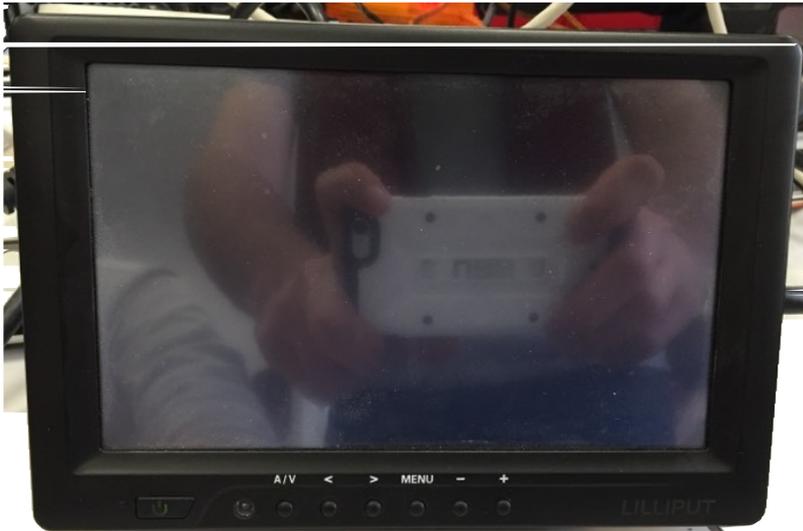
luminosite ← luminosite / 1,2

Tournesol	Version 2.1
Auteur : Nicolas Detilleux	67/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

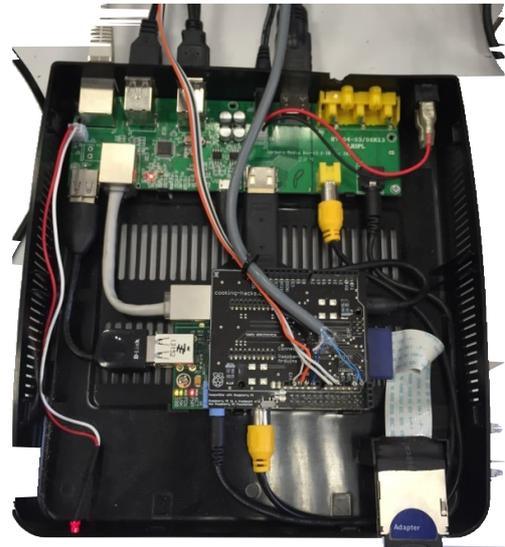
Mini écran :

Écran tactile Lilliput 669GL-70NP/C/T 7" HDMI



Raspberry pi :

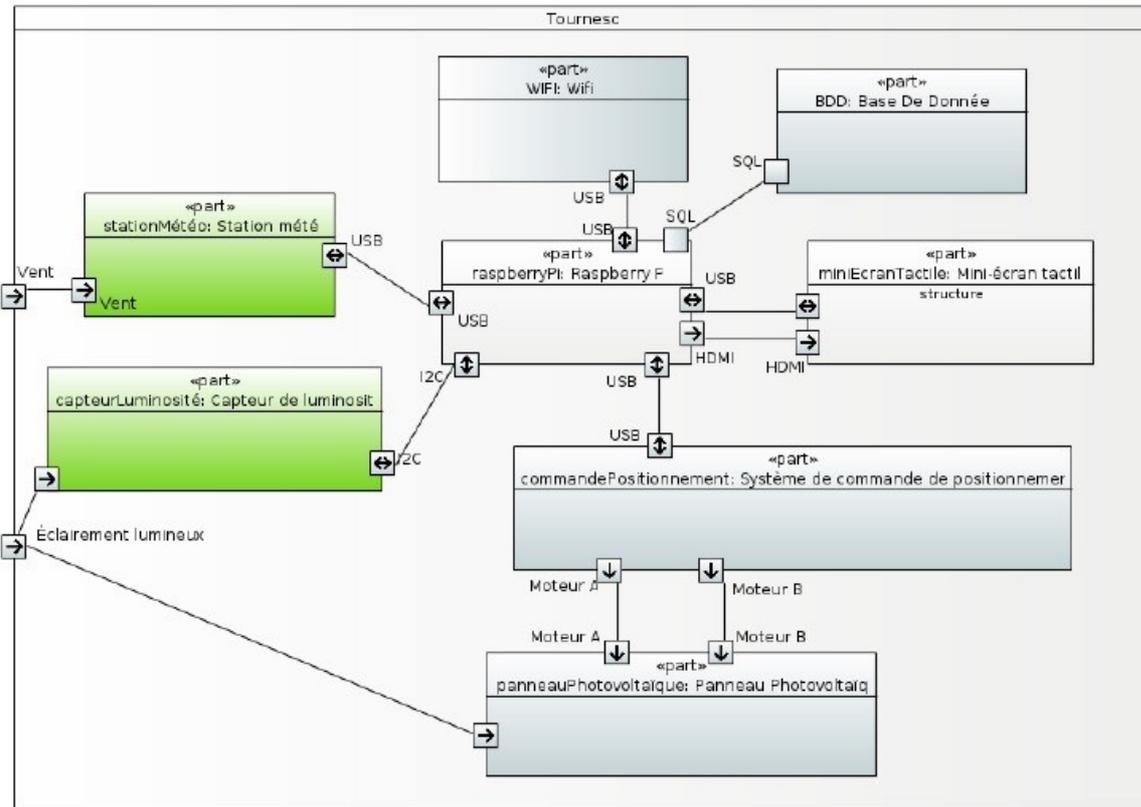
Nano-ordinateur modèle B à processeur ARM1176JZF-S 700 MHz équipé de 512MO de RAM et d'une carte SD 4GO (Raspbian OS)



Tournesol	Version 2.1
Auteur : Nicolas Detilleux	68/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

IX) Diagramme interne de bloc :



Tournesol	Version 2.1
Auteur : Nicolas Detilleux	69/101

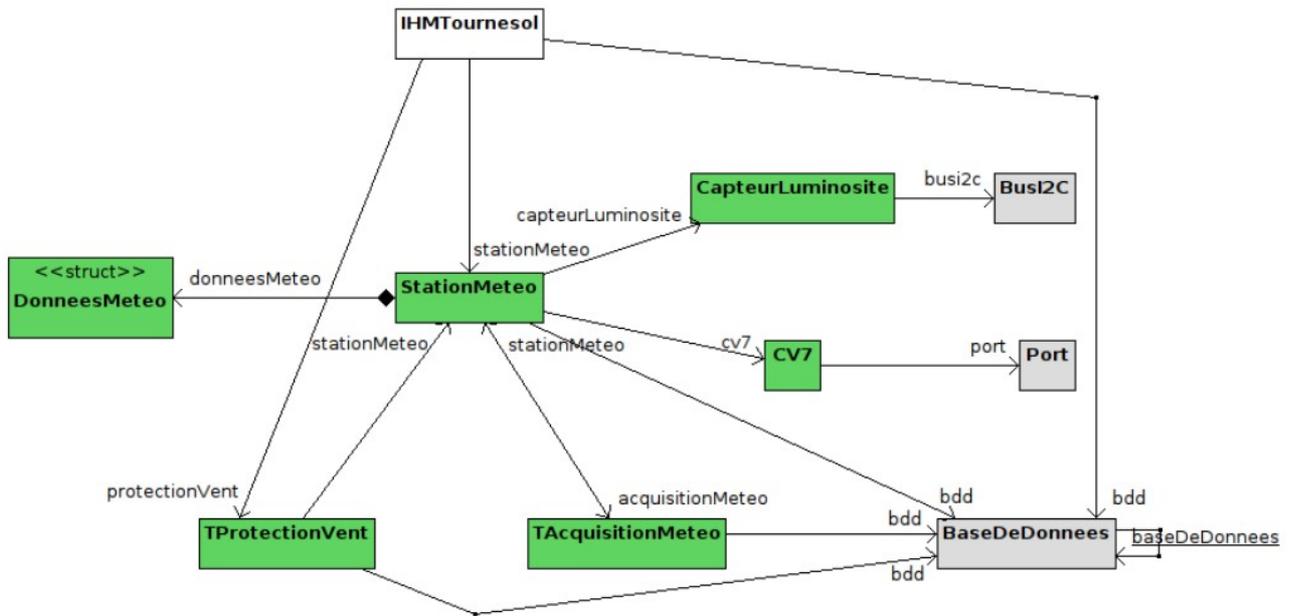
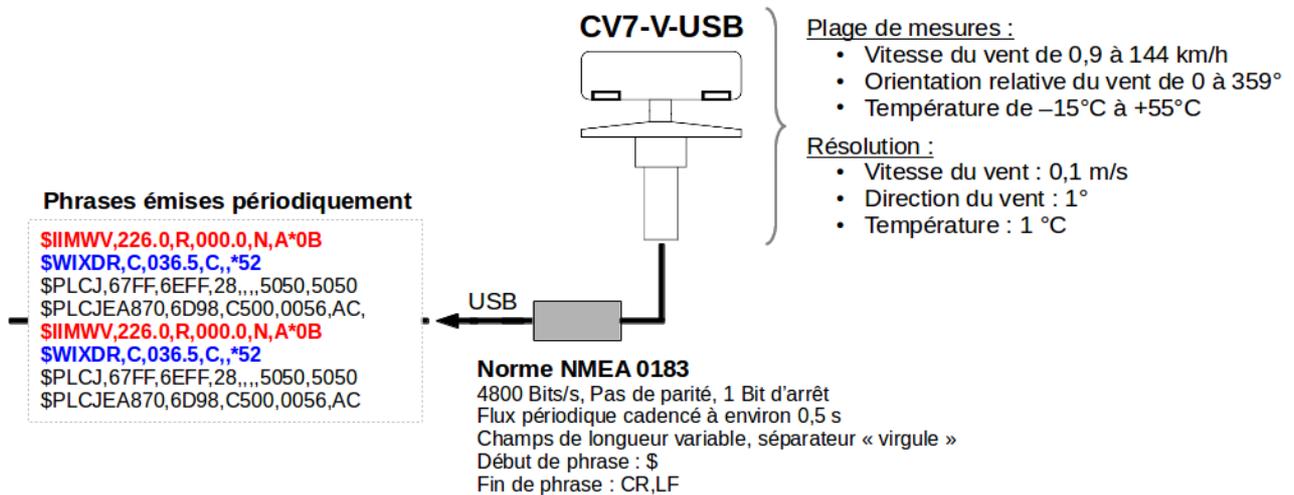


FIGURE 40 – Diagramme partiel des classes du module

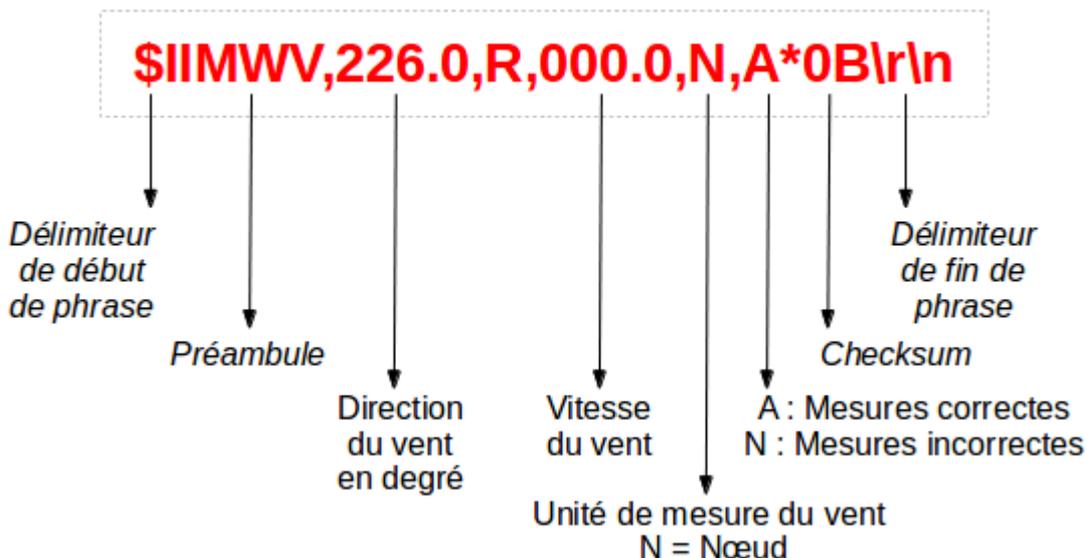
Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

X) Explications système :



La Station Météo CV7 envoie périodiquement des données. Nous nous intéressons aux trames de type « MWV » et « XDR » du protocole NMEA 0183 qui contiennent les informations que nous cherchons à obtenir.

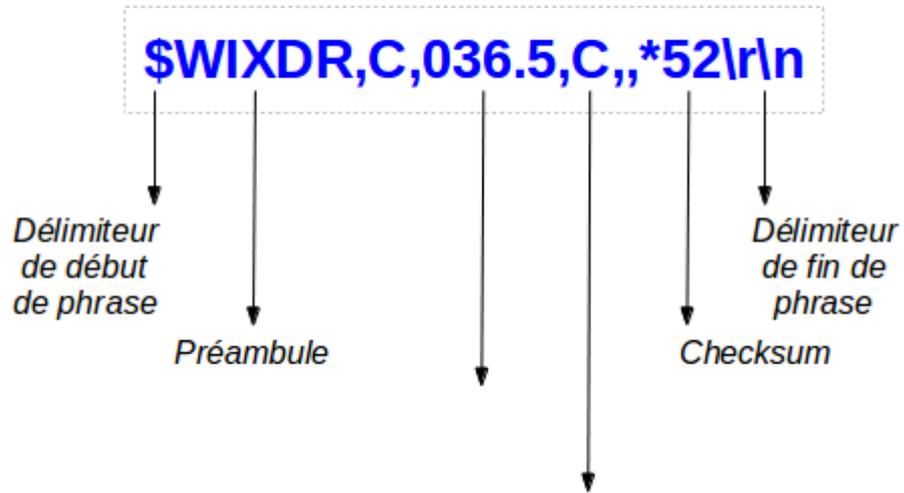
Phrase Vent



Tournesol	Version 2.1
Auteur : Nicolas Detilleux	71/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Phrase Température vent

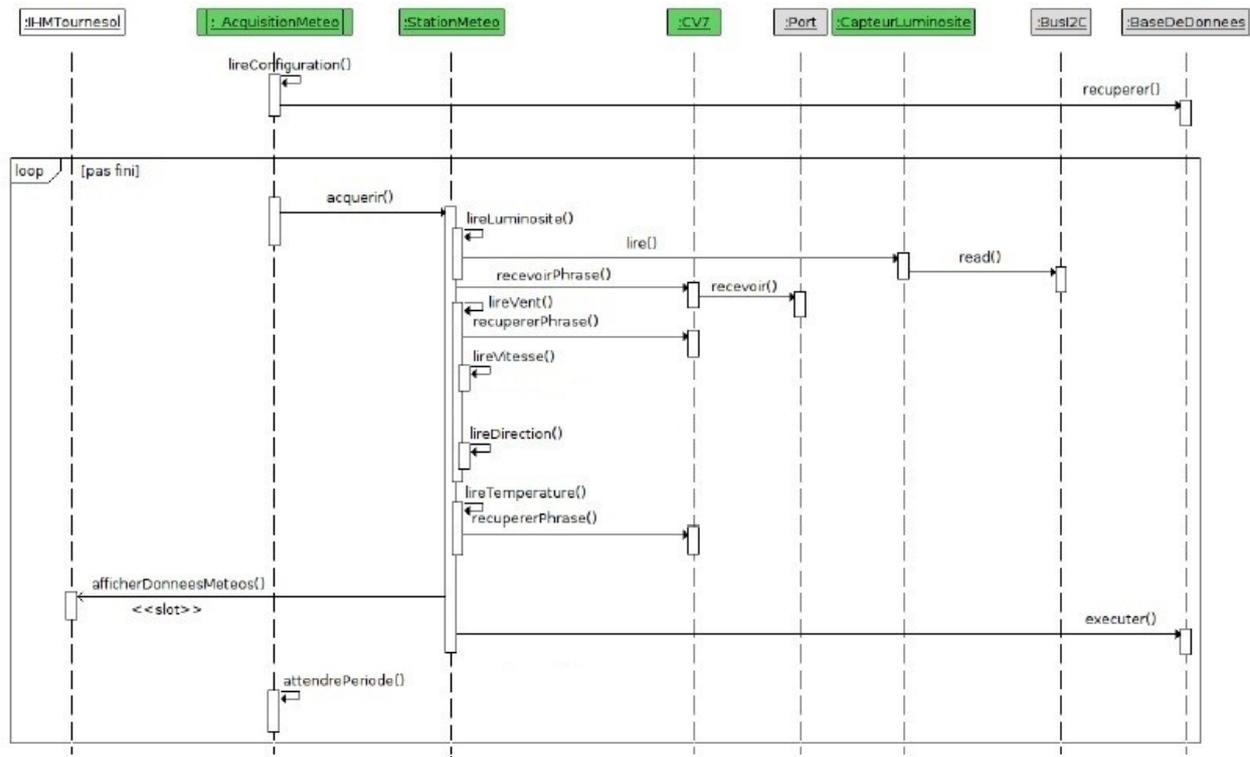


Température : 36,5 °C

Tournesol	Version 2.1
Auteur : Nicolas Detilleux	72/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

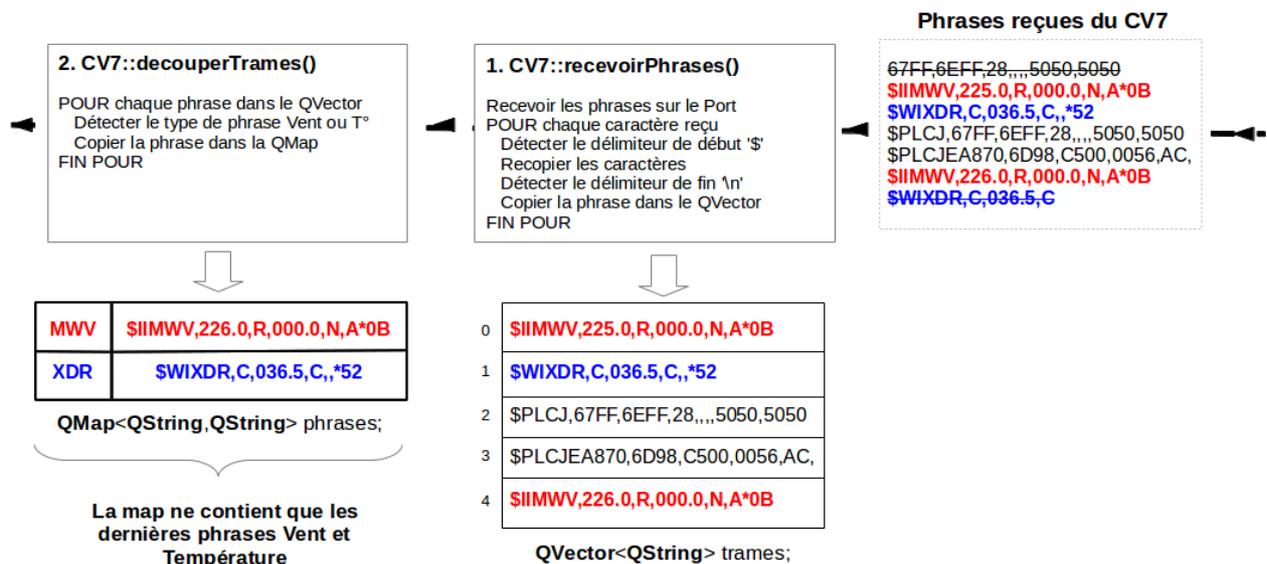
XI) Diagramme de séquence :



La classe **AcquisitionMeteo** est un *thread* qui réalise périodiquement (grâce à `attendrePeriode()`) l'acquisition de la luminosité et des données en provenance de la station météo CV7 et du capteur de luminosité BH1750 en appelant la méthode `acquerir()` de la classe **StationMeteo** :

- La méthode `lireLuminosite()` appelle la méthode `lire()` de la classe **CapteurLuminosite**, qui retourne un double, et le stocke dans une structure de type **InformationsStationMeteo**. La méthode `lire()` de la classe **CapteurLuminosite** utilise la classe **BusI2C** pour récupérer sous forme d'int l'information de luminosité comprise dans les trames. Puis elle le converti et retourne sous forme de double la luminosité.
- la méthode `recevoirPhrases()` de la classe **CV7** déclenche la réception des données, qui vont être découpées, triées, vérifiées, puis si une des trames est du type « MWV » ou « XDR » elle sera stockée dans la *map* de la classe **CV7**.

Tournesol	Version 2.1
Auteur : Nicolas Detilleux	73/101

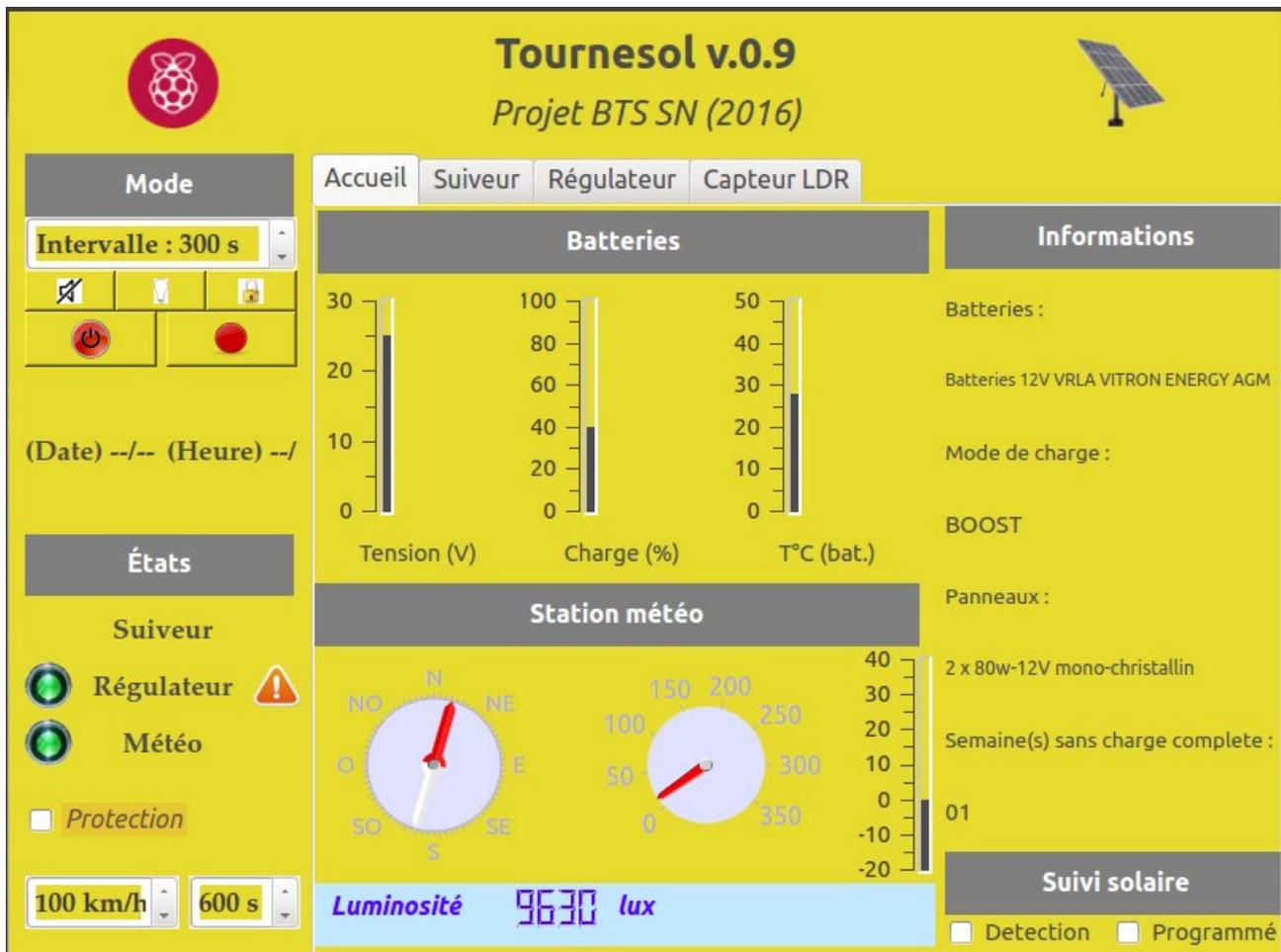


- la méthode lireVent() utilise la méthode recupererPhrase() de la classe StationMeteo pour récupérer une phrase du type « MWV », puis grâce aux méthodes lireVitesse() et lireDirection() elle extrait de cette trame la vitesse et la direction qu'elle stocke dans une structure de type **InformationsStationMeteo**.
- La méthode lireTemperature() ...

Une fois les données récupérées dans la structure de type **InformationsStationMeteo** le signal afficherDonneesMeteos() est émis pour assurer l'affichage dans l'IHM, et une requête est envoyée à la base de données

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

XII) Partie IHM :



Dans la partie IHM, je gère la partie Station météo, où je dois afficher la direction du vent, sa vitesse, la température qu'il fait, ainsi que l'affichage de la luminosité ambiante. Cet affichage se fait dans la partie « Accueil » de l'IHM.

De plus, dans la partie états, je dois pouvoir afficher à l'aide d'un voyant si la réception de la partie météo est effective. Et d'autre part, je dois pouvoir proposer le choix graphiquement d'activer ou non la protection contre le vent, et de rentrer les paramètres de vitesse maximale et de périodicité de vérification.

Tournesol	Version 2.1
Auteur : Nicolas Detilleux	75/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

XIII) Tests de validation :

Méthode acquérir() :

Ce que la méthode doit faire	Ce que la méthode fait	Validation (oui/non)
Appeler la méthode lireLuminosite() pour ajouter la luminosité dans la structure	Utilise la méthode lire luminosité pour ajouter la luminosité dans la structure	oui
Utiliser la méthode recevoirPhrases() pour remplir la map de la classe CV7	Utilise la méthode recevoirPhrases() pour remplir la map de la classe CV7	oui
Utiliser la méthode lireVent() pour ajouter dans la structure la direction et la vitesse du vent	Utilise la méthode lireVent() pour ajouter dans la structure la direction et la vitesse du vent	oui
Utiliser la méthode lireTemperature() pour ajouter la temperature dans la structure	Utilise la méthode lireTemperature() pour ajouter la temperature dans la structure	oui
Stocker les données vers la base de donnée	Stocke les données vers la base de donnée	oui
Transmettre la structure vers l'IHM	Transmet la structure vers l'IHM	oui

Tournesol	Version 2.1
Auteur : Nicolas Detilleux	76/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Méthode lireVent()

Ce que la méthode doit faire	Ce que la méthode fait	Validation (oui/non)
Utiliser la méthode recupererPhrase() en passant le type MWV en paramètre et ainsi récupérer une trame de type MWV depuis la map de la classe CV7	Utilise la méthode recupererPhrase() en passant le type MWV en paramètre et ainsi récupère une trame de type MWV depuis la map de la classe CV7	oui
Utiliser la méthode lireVitesse() pour extraire la vitesse de la trame et la socker dans la structure	Utilise la méthode lireVitesse() pour extraire la vitesse de la trame et la socker dans la structure	oui
Utiliser la méthode lireDirection() pour extraire la direction de la trame et la socker dans la structure	Utilise la méthode lireDirection() pour extraire la direction de la trame et la socker dans la structure	oui

Tournesol	Version 2.1
Auteur : Nicolas Detilleux	77/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

V-) Contrat individuel étudiant n°4 (Grégory Moll) : Module de contrôle de positionnement

I) Objectifs :

Vous intervenez en tant que technicien SN-IR pour “Concevoir une partie du système informatique Tournesol”.

Vous avez la charge du “**Module de contrôle de positionnement**” qui permet d’assurer l’orientation des panneaux automatiquement dans le mode de détection de la position du soleil à partir d’un capteur solaire composé de 4 photorésistances. Par le mini écran tactile, l’exploitant pourra commander manuellement l’orientation des panneaux en sélectionnant une position pré-programmée (nettoyage, vent, nuit et neige).

II) Fonctionnalités en charge de l’étudiant 4 :

Ref.	Fonction de service	Critère	Niveau	Flexibilité
FP-3	Orienter efficacement les panneaux en toute sécurité	Mode de suivi automatique	mode détection soleil	F0
FP-4	Commander manuellement les panneaux	Par positions pré-programmées	3 maximum (nettoyage, vent, nuit et neige)	F2
FP-5	Surveiller les alarmes	Module à surveiller	capteur solaire	F0
		Type d’alarme	visuelle	F0
FP-6	Paramétrer le fonctionnement	Sélection du mode de fonctionnement du suivi	manuel ou automatique	F0
		Sélection du mode de suivi automatique	2 modes (détection soleil ou TdAPS)	F0
		Réglage de l’intervalle de mise à jour du positionnement	en secondes (300 par défaut)	F0
		Réglage des positions pré-programmées (nettoyage, vent, nuit et neige)	par l’écran tactile en mode manuel	F2
FP-9	Archiver les états, données et alarmes	Base de données	SQL	F0
		Horodatage mesures	aaaa-mm-jj hh:mm:ss	F0

Tournesol	Version 2.1
Auteur : Grégory Moll	78/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

III) Production attendue :

- Une application informatique fonctionnelle ;
- Un modèle UML complet de la partie à développer ;
- Le code source commenté de l'application ;
- Les documentations prévues au paragraphe « Exigences sur la livraison ».

IV) Recette minimale étudiant 4 :

- L'activation et la désactivation du suivi automatique est effective et le mode détection de la position de soleil est pris en compte ;
- Le suivi automatique par capteur solaire est fonctionnel ;
- L'intervalle de suivi automatique est pris en compte pour le suivi automatique ;
- L'état du suiveur solaire est visible sur le mini écran tactile ;
- La commande manuelle d'orientation des panneaux est réalisable en sélectionnant une position pré-programmée (nettoyage, vent, nuit et neige) ;
- Les positions pré-programmées (nettoyage, vent, nuit et neige) sont modifiables et enregistrées ;

Tournesol	Version 2.1
Auteur : Grégory Moll	79/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Tournesol v.0.9
Projet BTS SN (2016)

Accueil Suiveur Régulateur Capteur LDR

Batteries

Batteries :
Batteries 12V VRLA VITRON ENERGY AGM

Mode de charge :

Panneaux :
2 x 80W-12V mono-cristallin

Semaine(s) sans charge complete :

Tension (V) Charge (%) T°C (bat.)

Station météo

NO N NE E SE S SO

150 200 250 300 350

40 30 20 10 0 -10 -20

Suivi solaire

Detection Programmé

Mode

Intervalle : 300 s

États

Suiveur

Régulateur

Météo

Protection

100 km/h 600 s

Luminosité 32545 lux

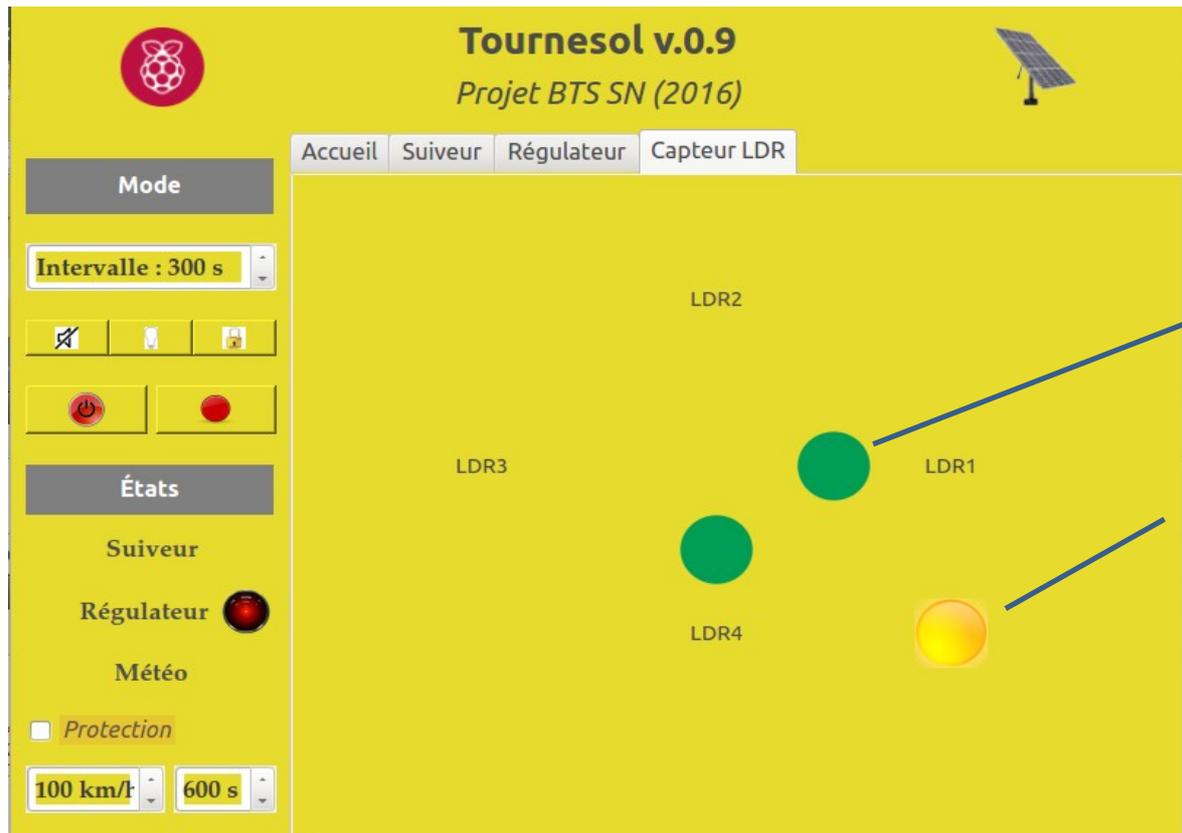
Réglage de l'intervalle périodique de commande des axes en mode automatique

Bouton de sélection du mode automatique/manuel

Affichage de l'état du suiveur

Tournesol	Version 2.1
Auteur : Grégory Moll	80/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016



LDR éclairée

Affiche la position du soleil par rapport aux LDR éclairées

Tournesol	Version 2.1
Auteur : Grégory Moll	81/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

V) Tâches à réaliser par l'étudiant 4 :

Tâches à réaliser	Production attendue	Estimation horaire
S'approprier le cahier des charges	Le cahier des charges est explicite. Les tâches à réaliser sont identifiées et les ressources sont définies.	16 h
Rechercher des solutions issues de l'innovation technologique pour les énergies solaires	Une étude comparative argumentée des différentes solutions est produite.	6 h (SCP)
Installer et configurer le système d'exploitation du nano ordinateur	Un compte rendu est rédigé.	4h
Installer et configurer le serveur de base de données	Un compte rendu est rédigé.	4h
Installer et configurer son environnement de développement	Le poste de développement est opérationnel.	2h
Installer et raccorder le capteur solaire	L'appareil fonctionne. La procédure d'installation a été respectée. Un compte rendu est rédigé.	4 h (SCP)
Étudier et documenter les caractéristiques des capteurs solaires	Les caractéristiques sont identifiées et des mesures sont réalisées.	6 h (SCP)
Mettre en œuvre les programmes de test fournis	Le rapport de tests est renseigné.	14 h + 4 h (SCP)
Finaliser la modélisation UML du module	Les diagrammes UML (diagrammes de séquence « définir des positions pré-programmées », « positionner les panneaux » et « enregistrer la nouvelle position », diagramme d'états du suivi automatique) sont élaborés et finalisés.	20 h

Tournesol	Version 2.1
Auteur : Grégory Moll	82/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

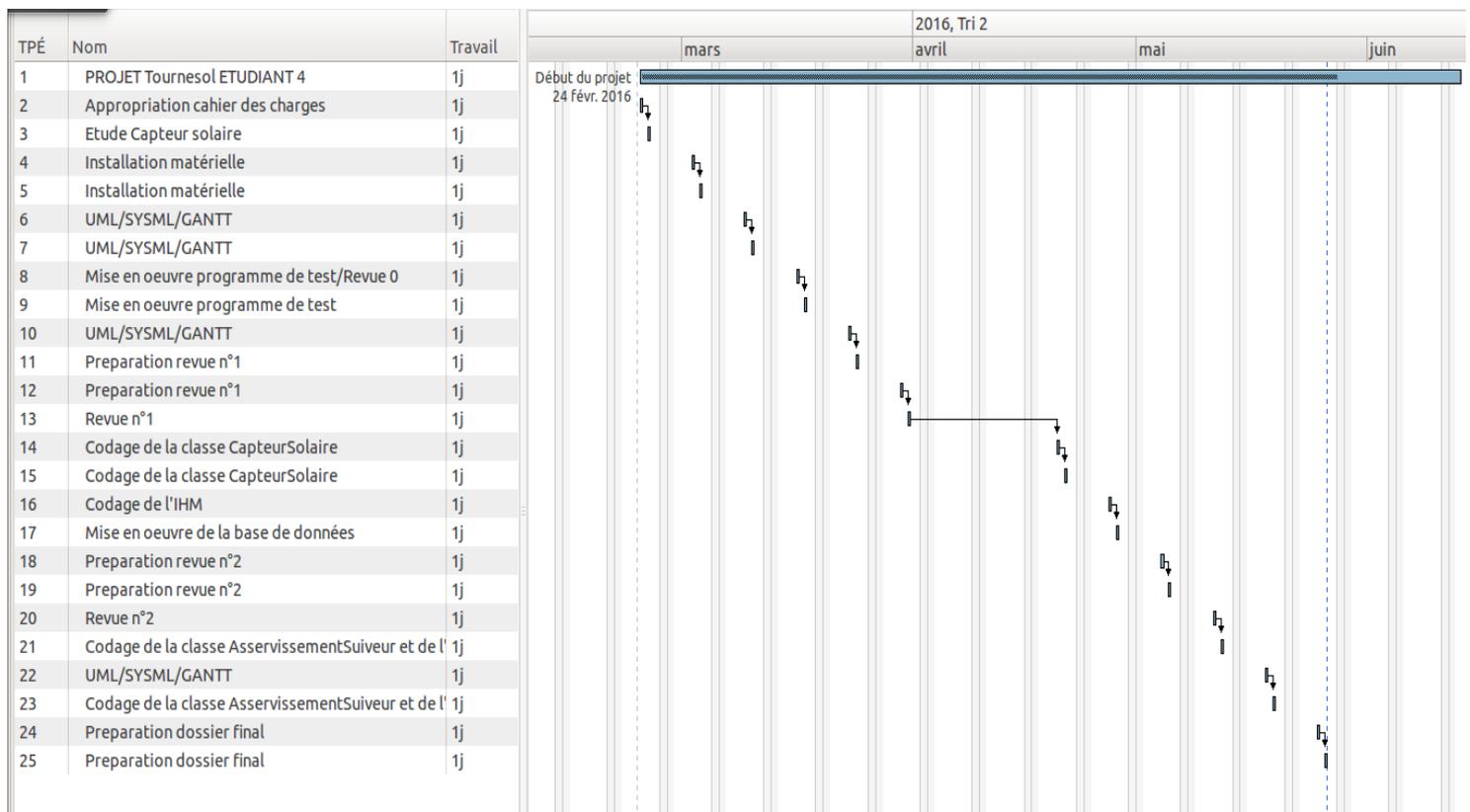
Produire la maquette de l'IHM du module	La maquette de l'IHM correspond aux exigences du cahier des charges.	8h
Coder et tester les classes du module	Les classes CapteurSolaire et TAsservissementSuiveur sont codées et valides. La classe SuiveurSolaire et IHMTournesol sont complétées et valides.	40 h
Réaliser les tests unitaires	Les tests unitaires des classes SuiveurSolaire, CapteurSolaire et TAsservissementSuiveur sont écrits et archivés.	16 h

Faire la recette du module	Le cahier de recette du module est validé.	4 h + 4 h (SCP)
Intégrer en équipe l'application complète	L'application est intégrée et fonctionnelle.	4h
Rédiger le dossier technique et les documents relatifs au projet	Le dossier est rédigé en respectant les exigences.	30 h
Produire un guide de mise en route et d'utilisation du module.	Un manuel est fourni.	4h
Gérer la planification	Le planning prévisionnel est établi. Le planning est actualisé avec une mise en évidence des écarts par rapport au prévisionnel.	10 h
Total		200 heures

Tournesol	Version 2.1
Auteur : Grégory Moll	83/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

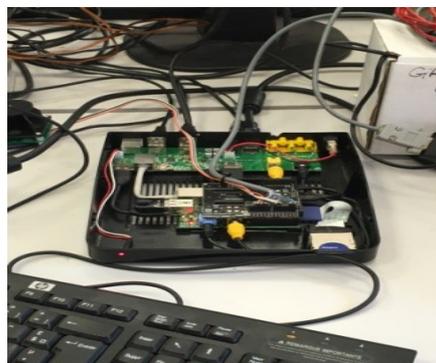
VI) Planification :



VII) Ressource matérielle pour l'étudiant 4 :

RPI :

Nano-ordinateur modèle B à processeur ARM1176JZF-S 700 Mhz équipé de 512 MO de RAM et d'une carte SD 4GO (Raspbian OS).



Tournesol	Version 2.1
Auteur : Grégory Moll	84/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

MINI-ECRAN :

Ecran tactile Lilliput 669GL-70NP/C/T 7° HDMI.



WIFI : Adaptateur USB Wifi.



LDR-MAX :

Capteur Solaire (4xLDR) avec dome PMMA.

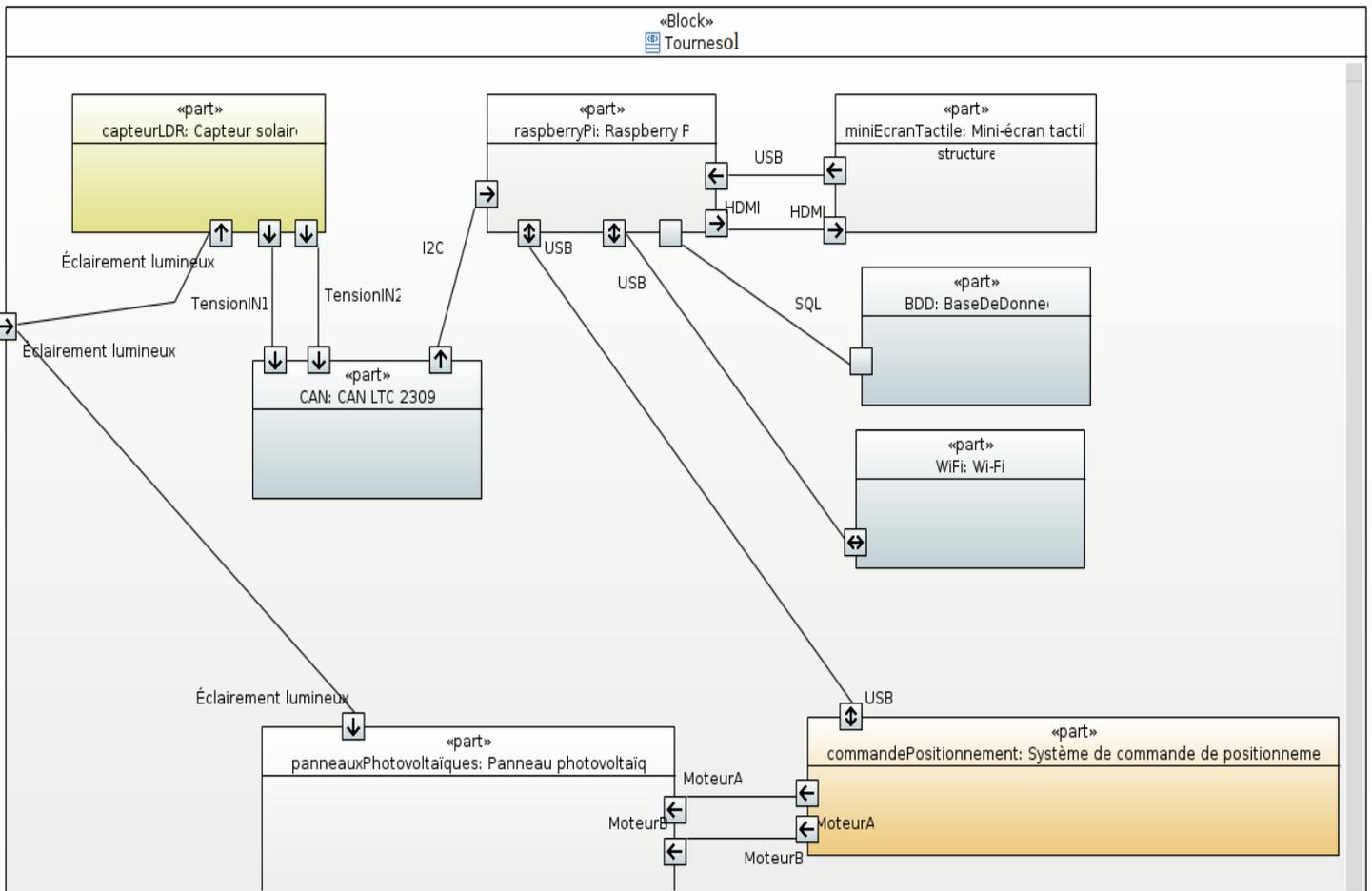
Une LDR (light-dependent resistor) est une photorésistance qui est un composant électronique dont la résistivité varie en fonction de la quantité de lumière incidente.



Tournesol	Version 2.1
Auteur : Grégory Moll	85/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

IX) Diagramme IBD étudiant 4 :



Ce diagramme de bloc permet d'expliquer plus clairement le fonctionnement du capteur solaire. Le capteur LDR reçoit un éclairage lumineux. Cet éclairage est envoyé à un convertisseur analogique numérique avec une tension IN1 et IN2. Une fois convertit la tension arrive en I2c à la raspberryPi qui peut envoyer, par connexion USB, comment déplacer les panneaux.

Tournesol	Version 2.1
Auteur : Grégory Moll	86/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

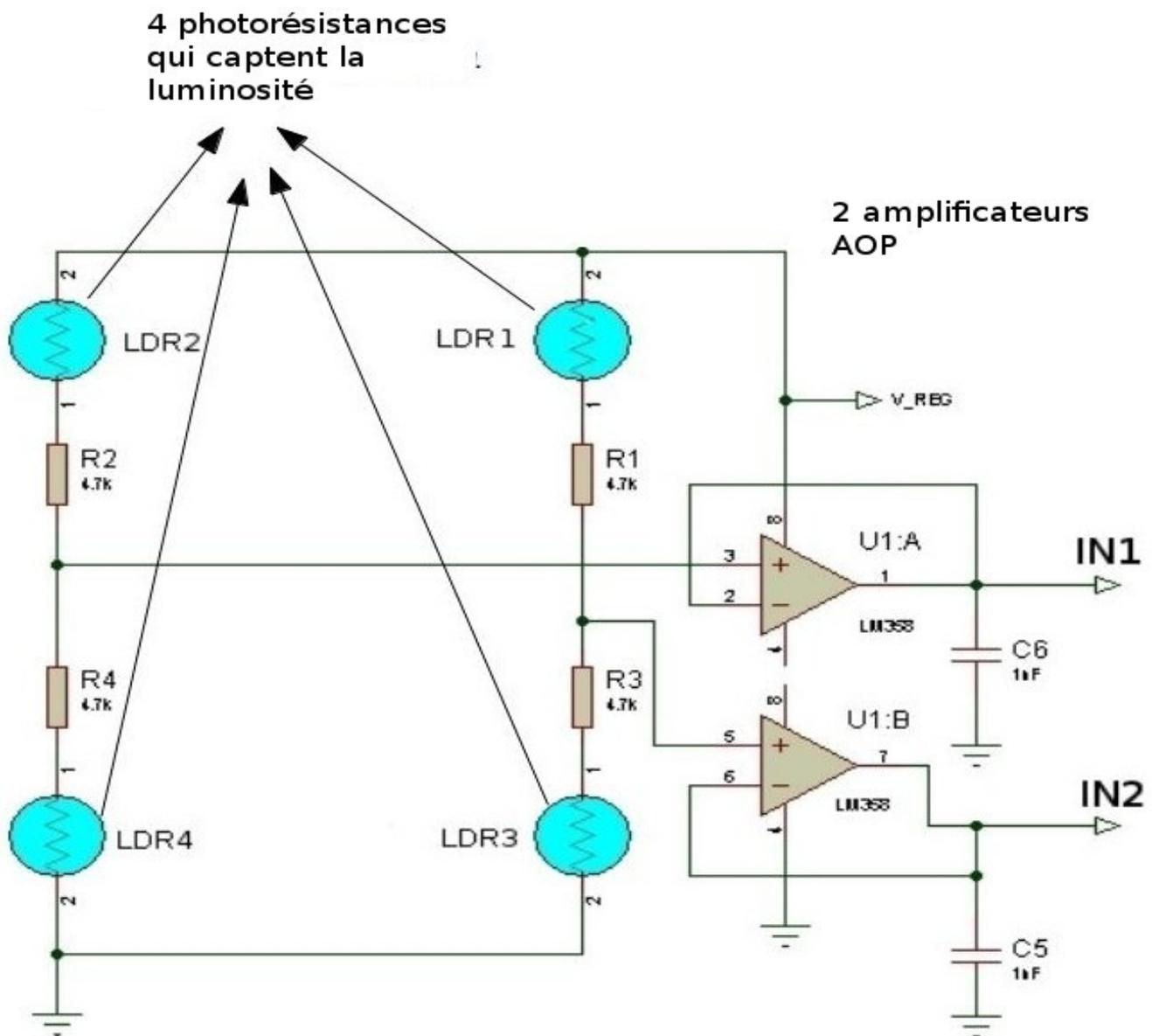
X) Ressource logicielle pour l'étudiant 4 :

Système d'exploitation du système	Raspbian (Linux raspberry 3.6.11+ minimum)
Système d'exploitation de développement	GNU Linux (Ubuntu 12.04.5 LTS)
Environnement de développement	Qt Creator 2.4.1 et Qt Designer 4.8.1
Compilateur	GNU g++/gcc version 4.6.3
Débugueur	GNU gdb 7.4
Chaîne de fabrication	qmake 2.01a et GNU make 3.81
API GUI	Qt 4.8.1 et Qwt version 5 ou 6
API Raspberry Pi pour <i>shield</i>	arduPI 2.08
Atelier de génie logiciel	bouml 4.23
Plate-forme de tests unitaires	CppUnit 1.12.1
Logiciel de gestion de versions	subversion (client svn 1.6.17)
Générateurs de documentation	Doxygen version 1.7.6.1 et pandoc 1.9.1.1
Gestionnaire de projet	Planner (version 0.14.5) ou gantter

Tournesol	Version 2.1
Auteur : Grégory Moll	87/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

XI) Détecter la position du soleil :



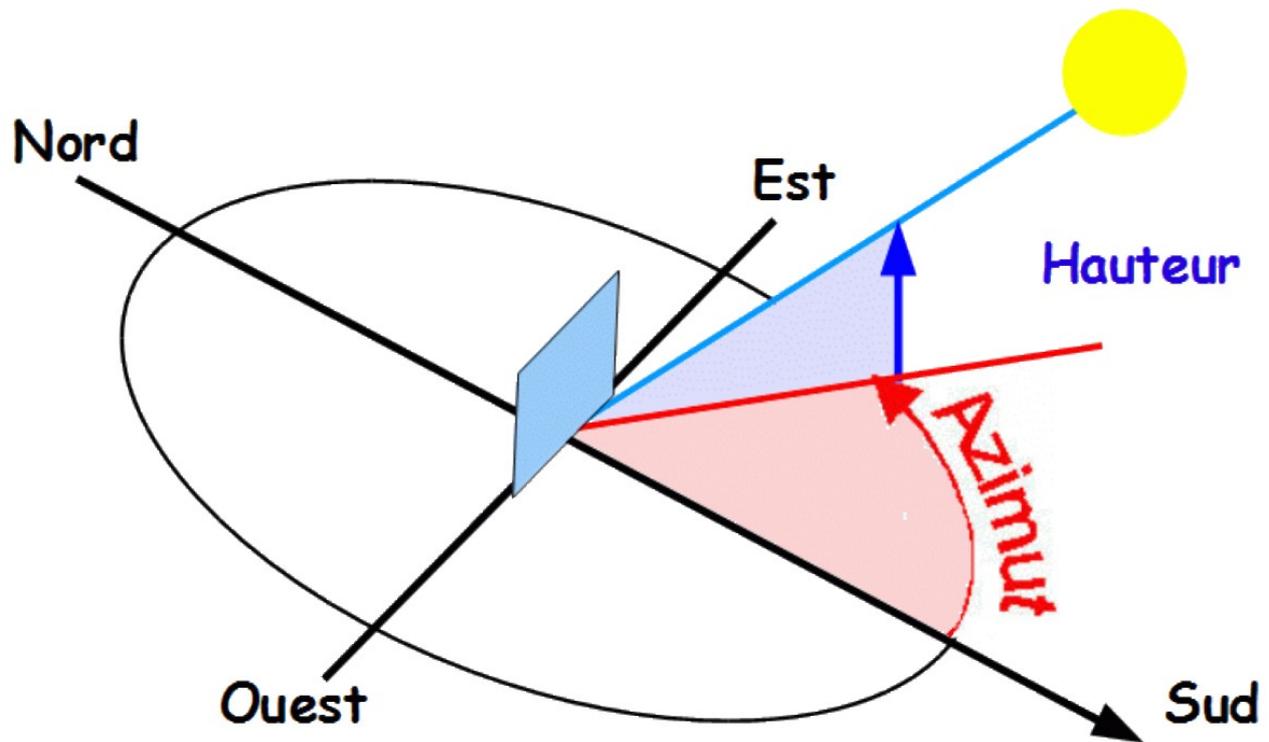
Ce capteur solaire est composé de 4 LDR (photorésistances) et permettra de déterminer l'azimut et la hauteur (élévation) du soleil.

Il est aussi composé de 2 amplificateur opérationnels (AOP en montage suiveur). Grâce à son

Tournesol	Version 2.1
Auteur : Grégory Moll	88/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

impédance d'entrée très importante et à sa faible impédance de sortie, le montage suiveur permet l'adaptation d'impédance vers le convertisseur analogique-numérique afin que celui-ci ne perturbe pas la détection des LDR.



Azimut solaire : Angle mesurée dans le sens des aiguilles d'une montre entre le point cardinal Sud (hémisphère nord) ou Nord (hémisphère sud) et la projection sur le plan horizontal local de la droite reliant la terre au soleil. L'angle est mesuré dans le sens des aiguilles d'une montre dans l'hémisphère nord et dans le sens contraire dans l'hémisphère sud.

Remarque : l'azimut solaire est négatif le matin (Est), nul ou égal à 180° à midi et positif l'après-midi (Ouest), sur tout le globe.

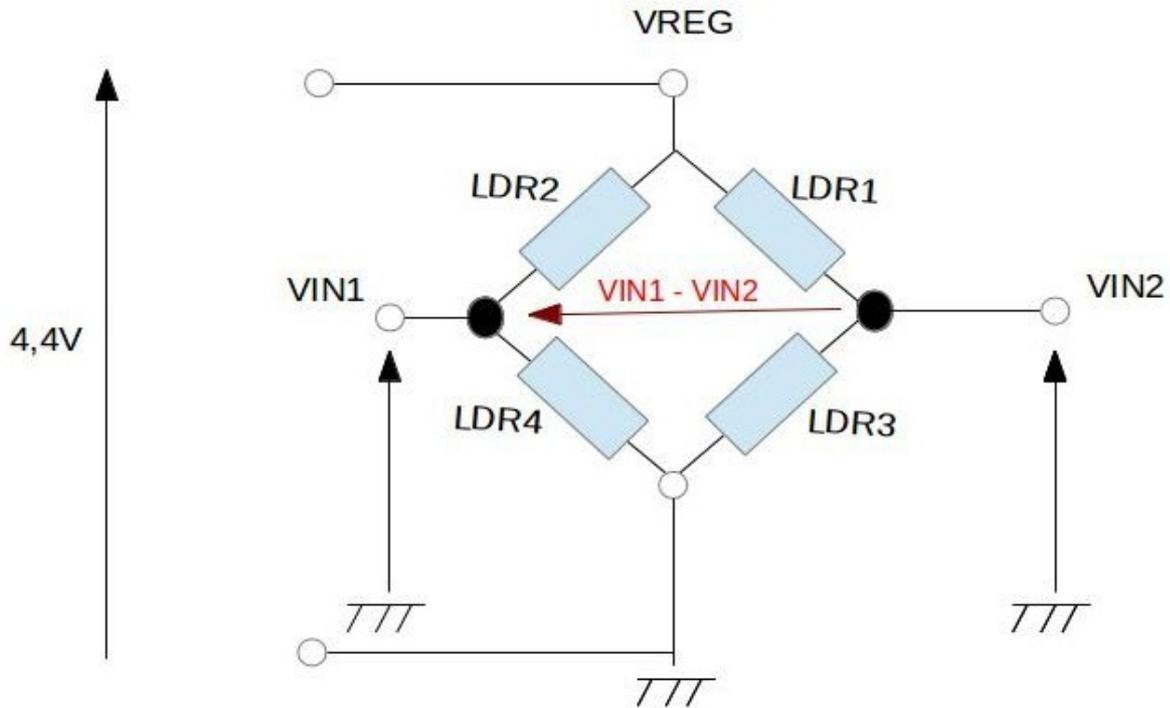
Hauteur solaire : La hauteur solaire est l'angle entre la droite joignant le centre du disque solaire au point d'observation et le plan horizontal passant par le point d'observation.

Le zénith est le point le plus élevé de l'hémisphère céleste, se trouvant directement à la verticale de l'observateur.

Tournesol	Version 2.1
Auteur : Grégory Moll	89/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Les 4 LDR sont reliés suivant le schéma ci-dessous :



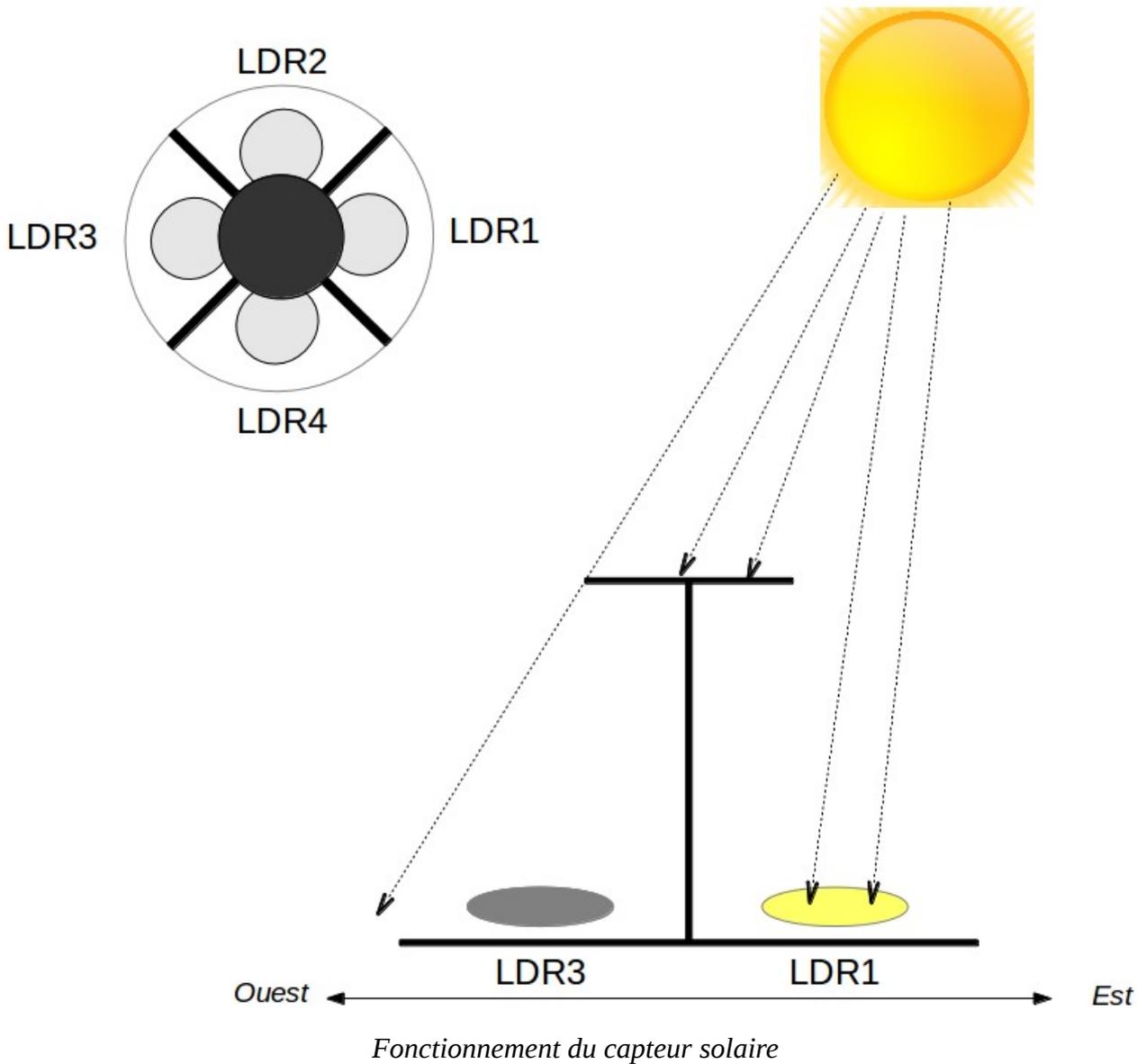
*Remarque : pour le schéma sur l'alimentation du capteur voir **Annexe 1**.*

LDR1 est orienté à l'est et LDR3 à l'ouest.
LDR2 est orienté au Nord et LDR4 au sud.

Tournesol	Version 2.1
Auteur : Grégory Moll	90/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Le schéma ci-dessous montre comment les LDR détectent la position du soleil.



Tournesol	Version 2.1
Auteur : Grégory Moll	91/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

A- A l'équilibre $(V_{IN1}-V_{IN2}) = 0V$.

Même éclairement sur les 4 photorésistances :

$$LDR1 = LDR2 = LDR3 = LDR4 \text{ donc}$$

$$V_{LDR1} = V_{LDR2} = V_{LDR3} = V_{LDR4} = V_{REG} / 2 = 2,2V$$

A l'équilibre, les 4 photorésistances sont éclairées de la même façon dans la lumière comme dans l'obscurité.

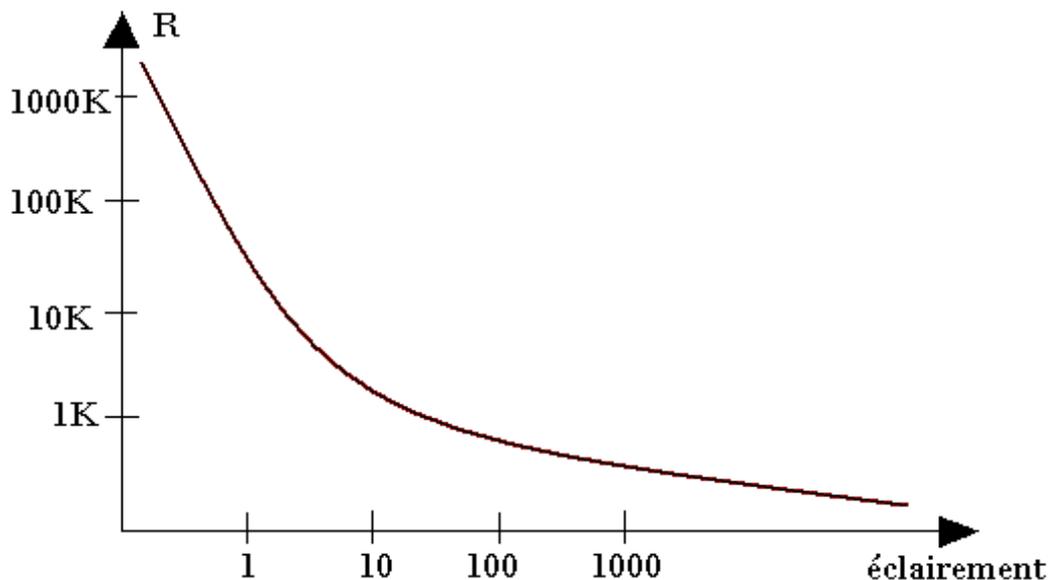
B- Déséquilibre.

- Au plus il y aura de l'éclairement au plus la valeur de la LDR diminuera.
- Au plus il y aura de l'obscurité au plus la valeur de la LDR augmentera.

Les tensions V_{IN1} et V_{IN2} vont donc varier en fonction de l'éclairement.

Le schéma ci-dessous nous montre la loi physique des photorésistance avec l'éclairement en lux et les résistances en ohms.

Exemple de variation de photorésistance LDR en fonction de l'éclairement



Tournesol	Version 2.1
Auteur : Grégory Moll	92/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

La tension du seuil de VIN1 et VIN2 est de 2,2 V (tension à l'équilibre).

Eclaircement	LDR2/LDR4	LDR1/LDR3	
LDRX	VIN1	VIN2	VIN1-VIN2
LDR1	Seuil	> Seuil	-
LDR2	> Seuil	Seuil	+
LDR3	Seuil	< Seuil	+
LDR4	< Seuil	Seuil	-

L'éclaircement de la LDR1 et de la LDR 3 permettra de déterminer l'azimut et la LDR 2 et la LDR4 la hauteur du soleil. Le tableau ci-dessus va donc permettre de déterminer la direction du soleil.

XII) Suivi solaire :

C- Tableau avec les mesures effectuées.

0 = LDR éteinte

1 = LDR allumée

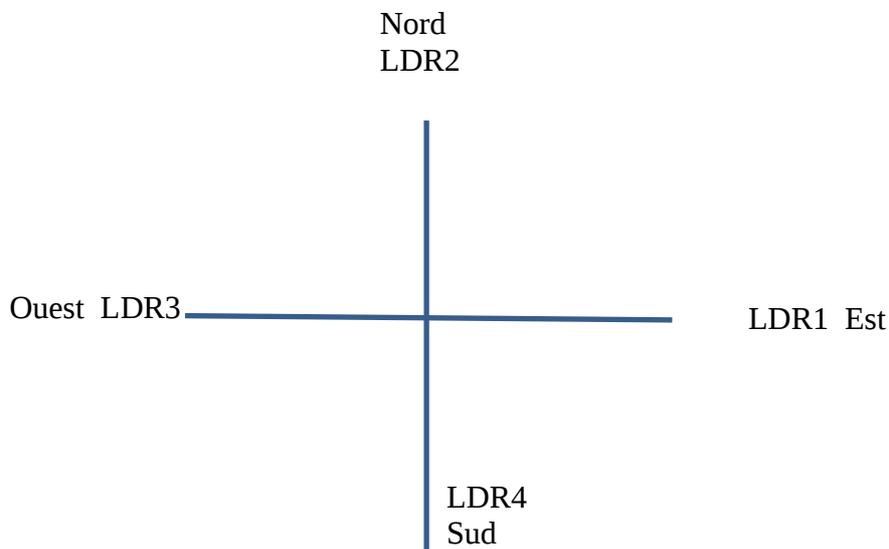
VIN1 et VIN2 en Volt

LDR1	LDR2	LDR3	LDR4	VIN1	VIN2	Direction du soleil
1	1	1	1	2,10	2,12	Centre
0	1	1	1	2,08	1,55	Ouest
1	0	1	1	1,8	2,1	Sud
1	1	0	1	2,10	2,69	Est
1	1	1	0	2,57	2,1	Nord
0	0	0	0	2,22	2,08	Centre
1	1	0	0	2,55	2,69	Nord-Est
0	0	1	1	2,09	1,56	Sud-Ouest
1	0	0	1	2,11	2,70	Sud-Est
0	1	1	0	2,58	2,11	Nord-Ouest

Tournesol	Version 2.1
Auteur : Grégory Moll	93/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Le capteur solaire étant positionné comme ceci sur le panneau :



Le panneau devra donc s'orienter de la manière suivante :

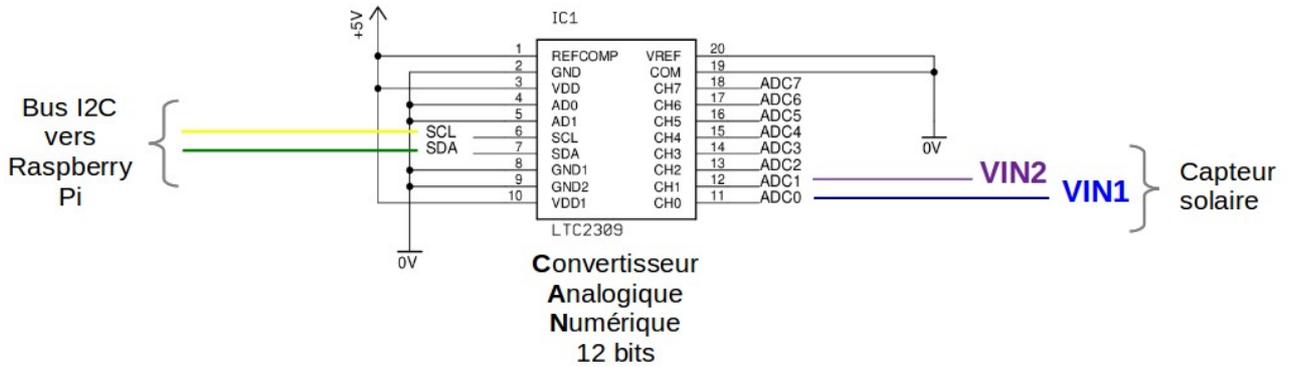
LDR éclairée ?	Déplacement du panneau
LDR 4 et 1	Sud-Est
LDR 2 et 1	Nord-Est
LDR 3 et 4	Sud-Ouest
LDR 3 et 2	Nord-Ouest
LDR 2	Nord
LDR 4	Sud
LDR 1	Est
LDR 3	Ouest
LDR 2, 3 et 4	Ouest
LDR 1, 3 et 4	Sud
LDR 1, 2 et 4	Est
LDR 1, 2 et 3	Nord

Remarque : le panneau se déplacera à partir d'un pas fixe défini par le suiveur solaire

Tournesol	Version 2.1
Auteur : Grégory Moll	94/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

D- Liaison vers le convertisseur Analogique Numérique



Capteur solaire

Moteur A : Le moteur A servira à bouger le panneau de l'Est à l'Ouest (Azimut)

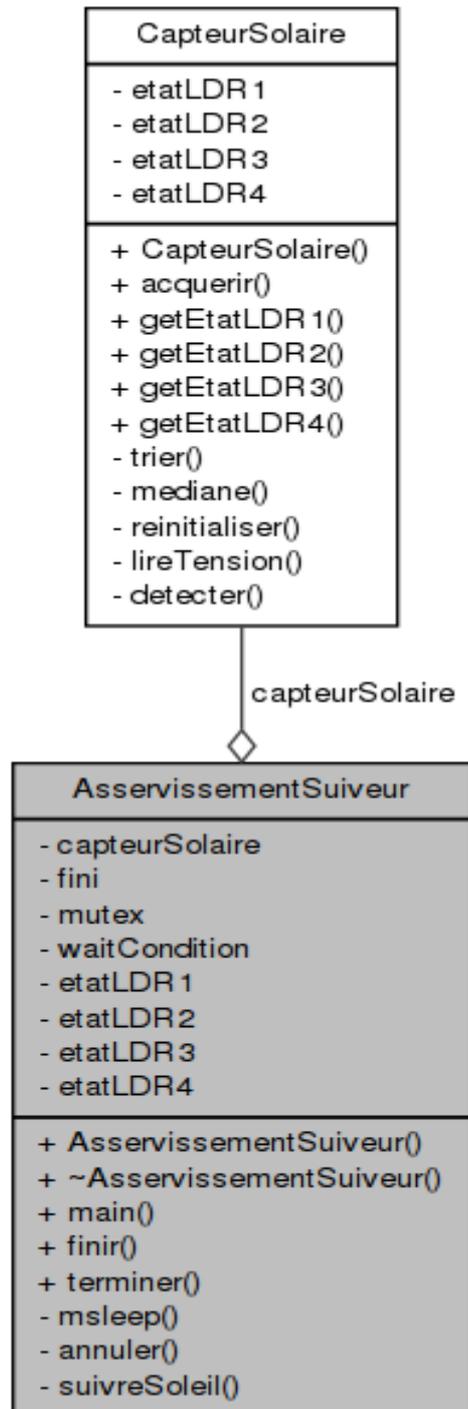


Moteur B : Le moteur B servira à bouger le panneau du Nord au Sud (Hauteur)

Tournesol	Version 2.1
Auteur : Grégory Moll	95/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

XIII) Diagramme de classes :



Tournesol	Version 2.1
Auteur : Grégory Moll	96/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Les classes **CapteurSolaire** et **AsservissementSuiveur** seront celles que je développe.

La classe **CapteurSolaire** va acquérir des données avec la méthode `acquérir()` qui va récupérer l'état des LDR avec les méthodes `getEtatLDR1()`, `getEtatLDR2()`, `getEtatLDR3()` et `getEtatLDR4()`.

Ces données sont des tensions, la méthode `trier()` va trier ces tensions pour ensuite les mettre dans un tableau et pouvoir détecter les 2 tensions IN1 et IN2.

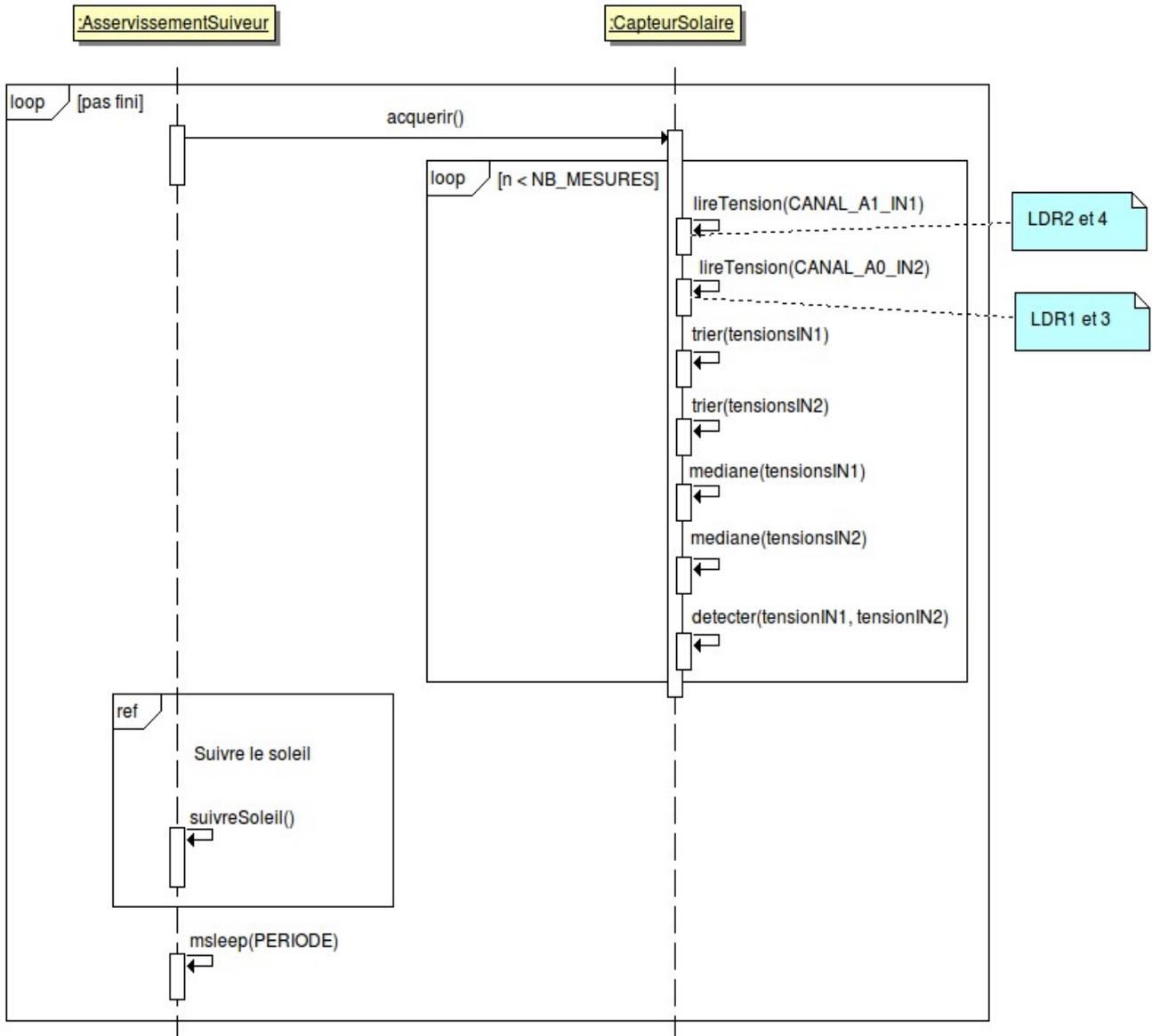
Une fois les tensions lues, cela m'indiquera l'état des LDR qu'on pourra par la suite fournir à la classe **AsservissementSuiveur**.

La classe **AsservissementSuiveur** va pouvoir ensuite envoyer comment suivre le soleil à la classe **AcquisitionSuiveur** avec la fonction `suivreSoleil()`.

Tournesol	Version 2.1
Auteur : Grégory Moll	97/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

XIV) Diagrammes de séquences :



Scénario «Détection du soleil»

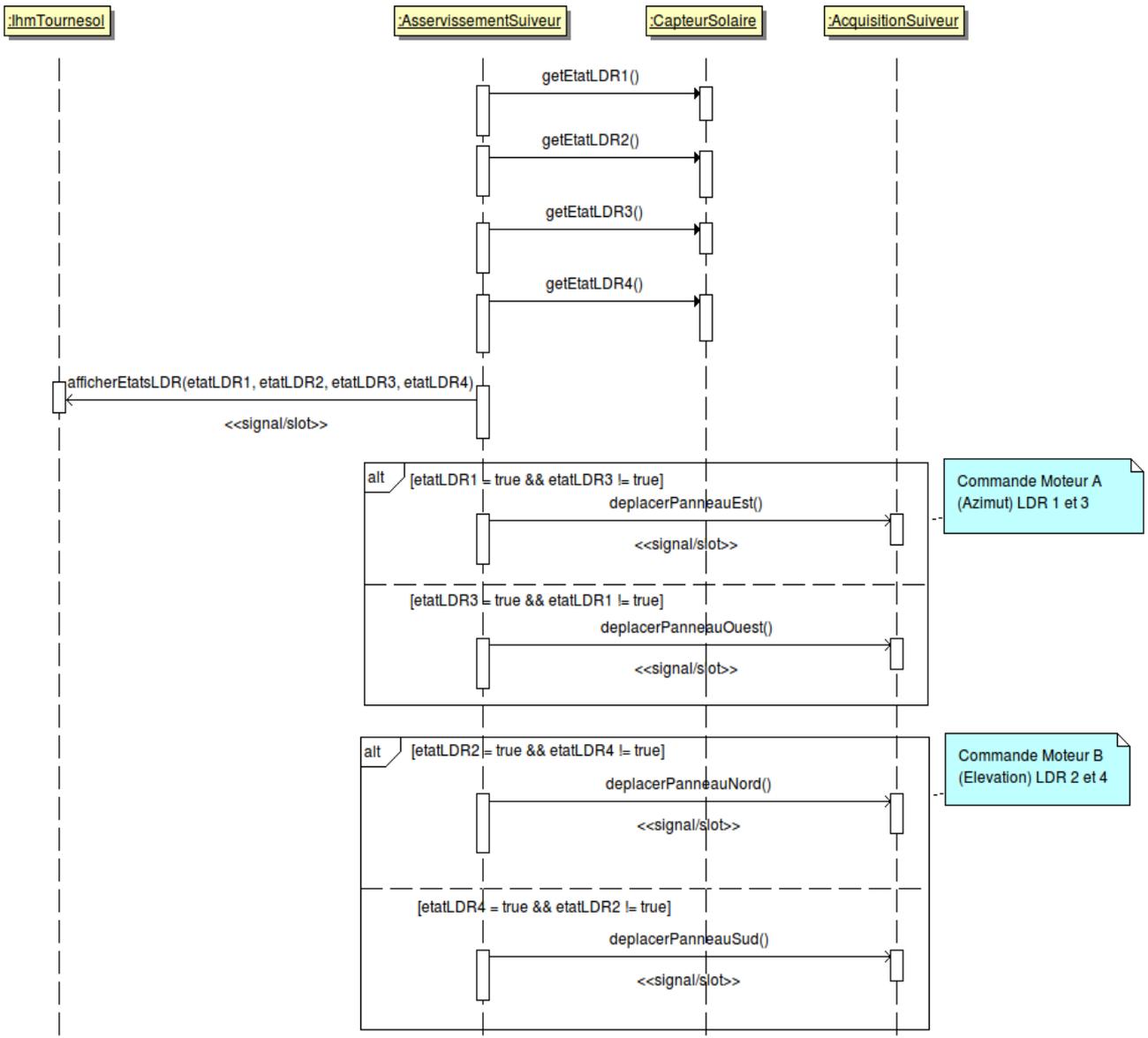
Ce diagramme de séquence montre comment le soleil va être détecté. La classe

Tournesol	Version 2.1
Auteur : Grégory Moll	98/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

AsservissementSuiveur va appeler la méthode **acquérir()** de la classe **CapteurSolaire**.

Cette méthode **acquérir()** va lire les tensions IN1 et IN2, les trier et donc par la suite détecter le soleil en les comparant à la tension de seuil.



Scénario « Orienter le panneau »

Tournesol	Version 2.1
Auteur : Grégory Moll	99/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

Ce diagramme de séquence montre comment le panneau va être orienté.

La classe **AsservissementSuiveur** va récupérer l'état des 4 LDR dans la classe **CapteurSolaire**.

AsservissementSuiveur va envoyer un signal à **IhmTournesol** qui sera l'état des LDR.

Suivant donc l'état des LDR, la classe **AsservissementSuiveur** va envoyer des signaux à la classe **AcquisitionSuiveur** qui donneront l'indication pour orienté le panneau (Au nord, au sud, à l'est ou à l'ouest).

XV) Test unitaires :

Méthode acquerir()

Ce que la méthode doit faire	Ce que la méthode fait	Validation (oui/non)
Lire les tensions IN1 et IN2	Elle lit la tension IN1 et la tension IN2	Oui
Trier les 2 tensions et les mettre dans un tableau	Elle trie les tensions et les mettre dans un tableau	Oui
Faire la médiane des tensions du tableau	Elle fait la médiane des tensions du tableau	Oui
Détecter 2 tensions (IN1 et IN2)	Elle détecte 2 tensions (IN1 et IN2)	Oui

Méthode suivreSoleil()

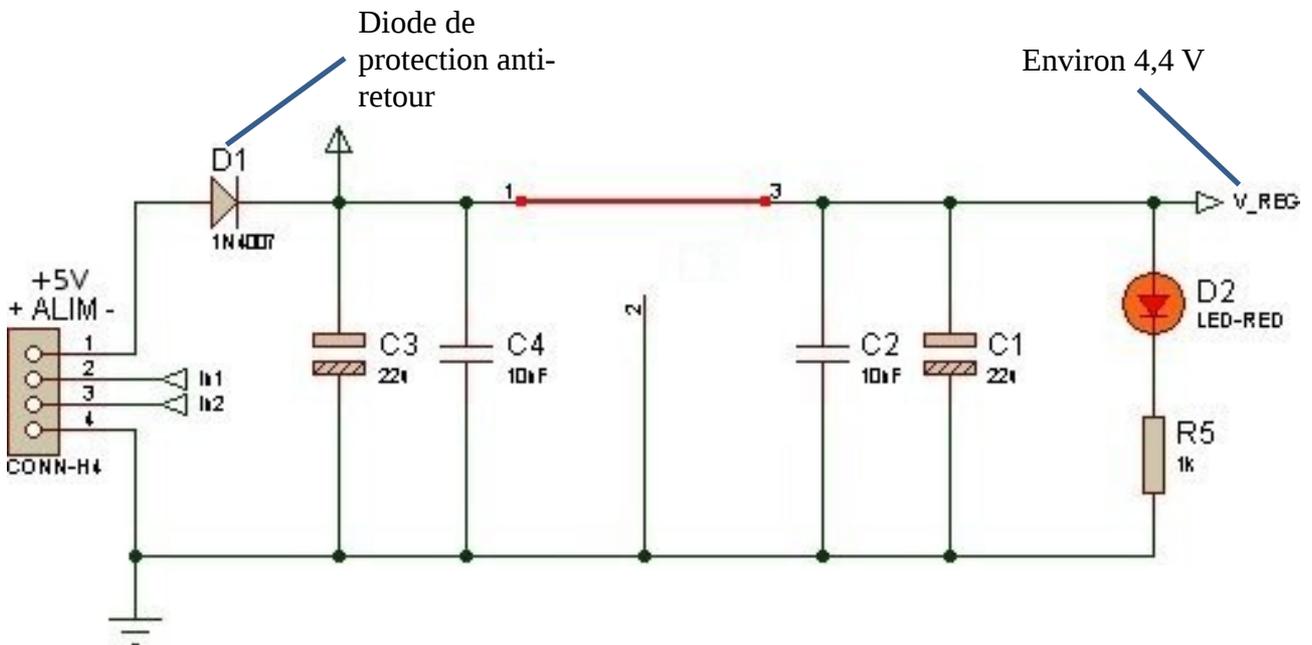
Ce que la méthode doit faire	Ce que la méthode fait	Validation (oui/non)
Récupérer l'état des LDR dans la classe CapteurSolaire	Elle récupère l'état des LDR dans la classe CapteurSolaire	Oui
Envoyer un signal à l'ihm pour afficher l'état des LDR	Elle envoie un signal à l'ihm pour afficher l'état des LDR	Oui
Vérifier si les 4 LDR sont face au soleil	Elle vérifie si les 4 LDR sont face au soleil	Oui
Envoyer un signal à l'ihm déplacerEst() si la LDR 1 est éclairée	Elle envoie un signal à l'ihm déplacerEst() si la LDR 1 est éclairée	Oui
Envoyer un signal à l'ihm	Elle envoie un signal à l'ihm	Oui

Tournesol	Version 2.1
Auteur : Grégory Moll	100/101

Tournesol	BTS SN
Lycée Saint Jean Baptiste de la Salle	2015-2016

deplacerOuest() si la LDR 3 est éclairée	deplacerOuest() si la LDR 3 est éclairée	
Envoyer un signal à l'ihm deplacerNord() si la LDR 2 est éclairée	Elle envoie un signal à l'ihm deplacerNord() si la LDR 2 est éclairée	Oui
Envoyer un signal à l'ihm deplacerSud() si la LDR 4 est éclairée	Elle envoie un signal à l'ihm deplacerSud() si la LDR 4 est éclairée	Oui

XVI) Annexe :



Annexe 1 : Schéma de l'alimentation du capteur

Ce schéma montre comment les capteurs sont alimentés.

Nous avons enlevé un régulateur 6 V car la carte est alimentée en +5V par la Raspberry Pi.

L'alimentation dispose d'une diode anti-retour qui provoque une chute de 0,6 V.

La tension V_Reg est donc d'environ 4,4 V.

Tournesol	Version 2.1
Auteur : Grégory Moll	101/101