

1. Avant propos	1
2. Plugins nécessaires.....	1
1. Subversive	1
2. CDT	1
3. AVR Eclipse	1
3. Configuration du SVN.....	2
1. Ajout du répertoire SVN	2
2. Création du projet	3
1. Partie informatique	7
2. Partie électronique	7
4. Utilisation du SVN	8
5. Conclusion	9

Avant propos

Eclipse est un IDE fonctionnant sur toutes les plateformes ayant du Java. Nous utiliserons **tous** Eclipse pour le développement afin, entre autres, de pouvoir s'entraider en cas de besoin.

Vous devez donc installer Eclipse (c'est gratuit et Hadopi n'est pas encore en place, donc pas de soucis). Pour windows, allez donc sur <http://www.eclipse.org/downloads/>. Pour Linux, il est préférable d'utiliser les paquets de votre distro, ça sera bien plus simple.

Plugins nécessaires

Subversive

Subversive est un plugin SVN pour Eclipse. Ainsi tout notre code sera stocké sur [Google Code](#) et tout le monde y aura accès. Rendez-vous [ici](#) pour les instructions d'installation de Subversive. Je vous avoue que c'est probablement la partie la plus prise de tête.

CDT

Cette sous-section ne concerne que l'équipe Electronique puisque nous coderons beaucoup en C.

Si vous avez Eclipse 3.5, il devrait déjà être inclus (à en croire [ce lien](#)). Sinon, il faut installer le plugin de développement C et C++ pour Eclipse.

Pour cela, ouvrez Eclipse, et cliquez sur "Help", puis "Software Updates/Find and Install". Ensuite, cliquez sur "Search for new software to install" puis "Next". Enfin, cliquez sur "New remote site" et entrer l'URL suivante avant de valider: <http://download.eclipse.org/tools/cdt/releases/ganymede/>

Il faut ensuite naviguer dans la liste des plugins proposer et sélectionner le paquet CDT Development (ou quelque chose comme ça, au pire, montrez-moi je saurais vous guider).

AVR Eclipse

Une fois de plus, cette sous-section ne concerne que l'équipe Electronique, et plus particulièrement que les développeurs du code pour microcontrôleurs. Toujours afin

d'harmoniser le développement, je propose d'utiliser le plugin [AVR pour Eclipse](#) plutôt que Arduino. Rien ne vous empêche d'utiliser Arduino **à condition de bien mettre à jour le code via SVN** dès qu'une modification a lieu. Une fois de plus, cette étape est bien plus aisée si vous utilisez Eclipse.

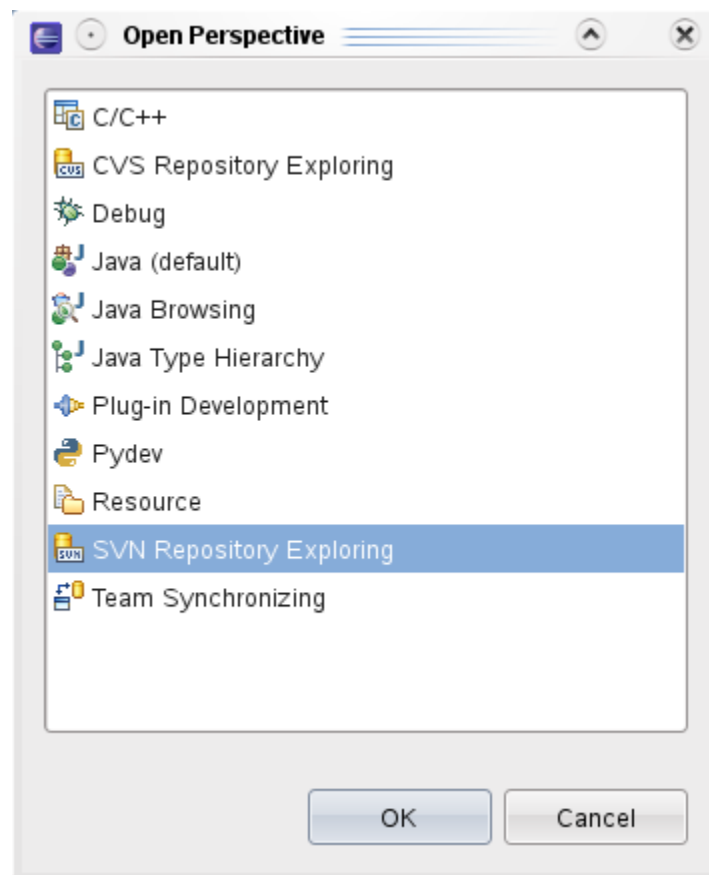
Commencez par installer AVRDUDE ou un produit équivalent permettant de communiquer avec des microcontrôleurs AVR. Sous Linux, je vous conseille d'utiliser votre gestionnaire de paquets et de chercher pour "avr". Sur OpenSuSE, il y avait deux paquets à installer. Pour installer AVR Eclipse, il faut procéder de façon analogue à l'installation de CDT (pour Eclipse < 3.5): ajouter l'URL suivante dans les sites de MàJ. <http://avr-eclipse.sourceforge.net/updatesite/>

Configuration du SVN

Cette étape concerne tout le monde.

Ajout du répertoire SVN

Pour configurer le SVN, il faut commencer par ajouter le serveur SVN à la liste de vos SVN disponibles. Pour cela, commencez par ouvrir Eclipse. Allez ensuite dans "Window" puis "Open Perspective" et cliquez sur "Other". Sélectionnez "SVN Explorer".



Puis, dans la marge de droite, qui est probablement vide, faites un clic droit (ou Alt+Click

sous Mac) et cliquez "Repository location" dans le menu "New". Entrez l'URL suivante avant de valider:

`https://sensitive-home.googlecode.com/svn`

Entrez aussi votre nom de compte Google (seulement la partie avant l'arobase) et votre mot de passe lié au projet que vous trouverez [ici](#).

New Repository Location

Enter Repository Location Information

Define the SVN repository location information. You can specify additional settings for proxy and svn+ssh, https connections.

General Advanced SSH Settings SSL Settings

URL: `https://sensitive-home.googlecode.com/svn` Browse...

Label

☒ Use the repository URL as the label

☐ Use a custom label:

Authentication

User: `christopher.rabotin`

Password:

☒ Save password

⚠ Saved secret data is stored on your computer in a file that's difficult, but not impossible, for an intruder to read.

Show Credentials For: `<Repository Location>`

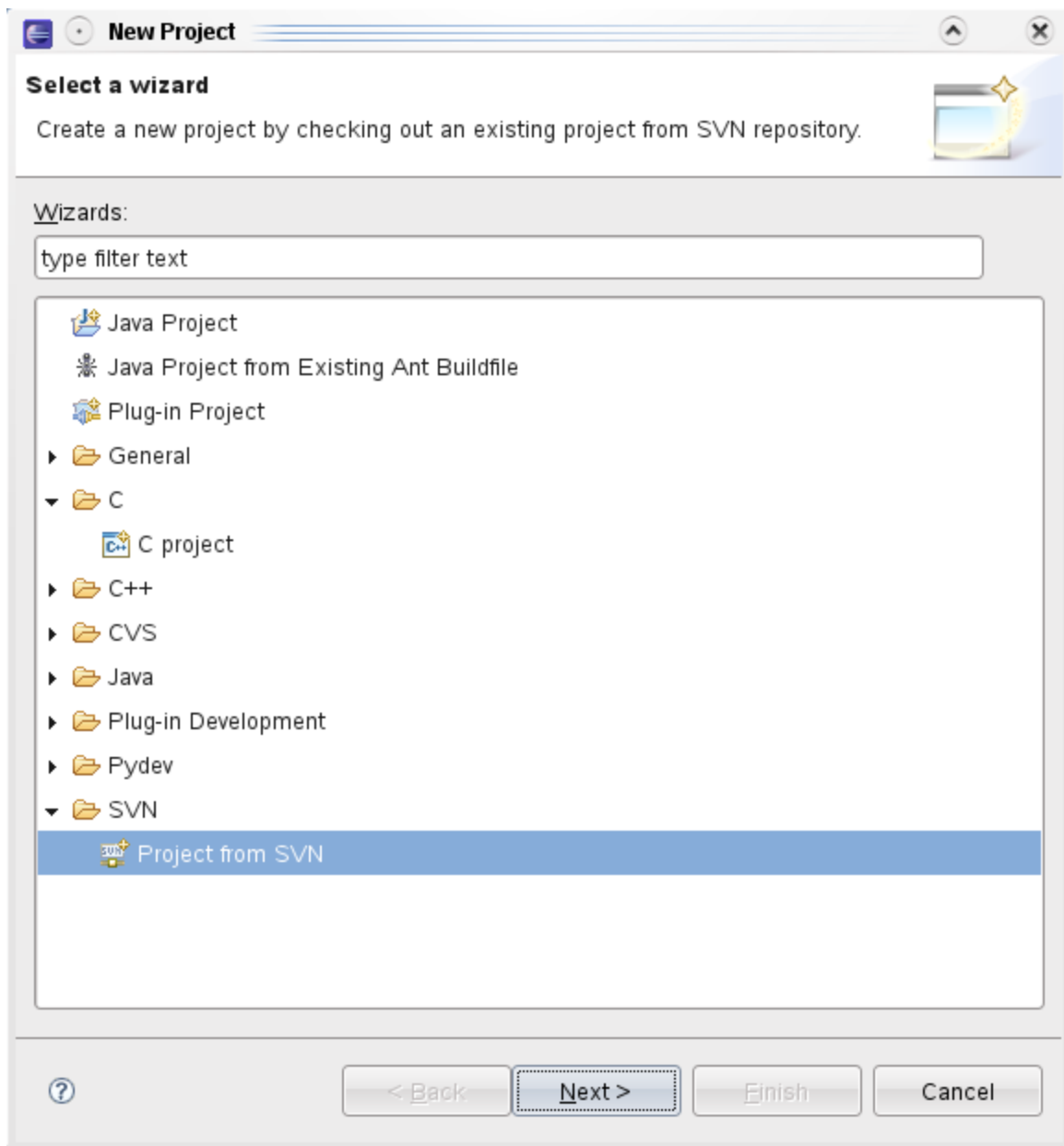
☒ Validate Repository Location on finish Reset Changes

Finish Cancel

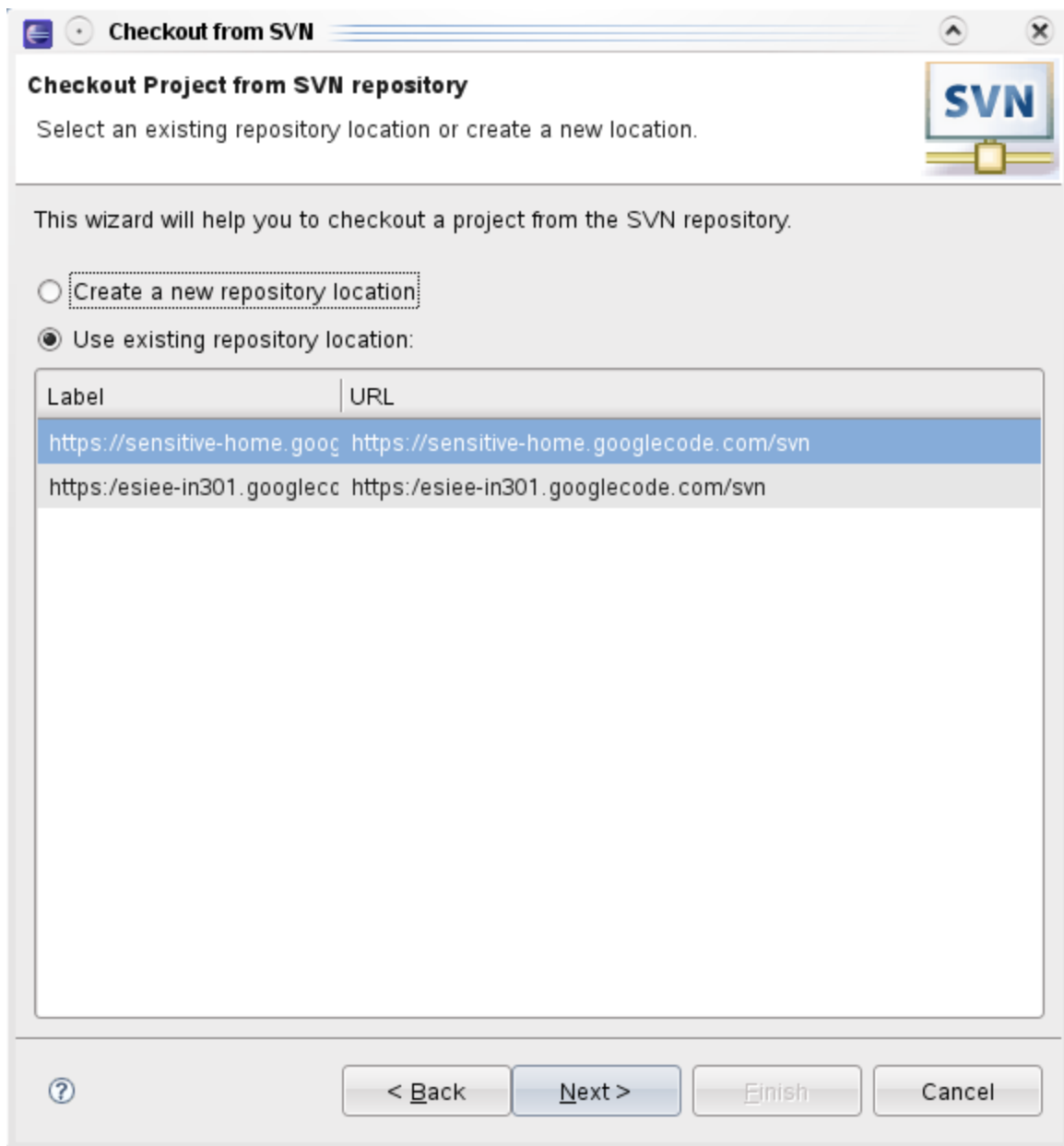
Vérifiez que vous n'avez pas oublié ni le **s** après *http* ni le */svn* et cliquez sur "Finish".

Création du projet

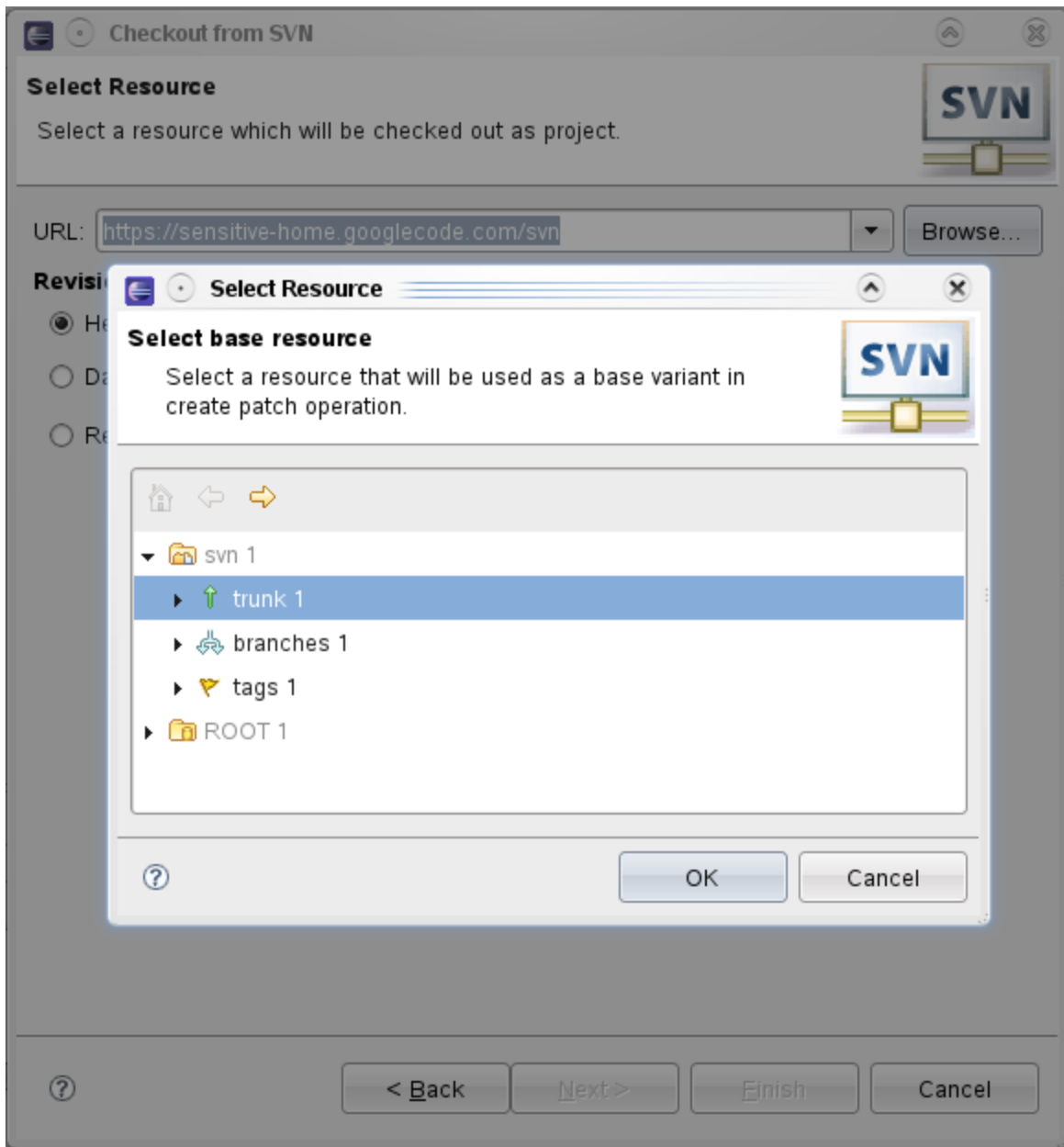
Dans cette étape, ça se corse, j'y ai donc mis beaucoup d'impressions d'écran. Commencez par faire Ctrl+N pour créer un nouveau projet de type "Autre". Sélectionnez ensuite "Project from SVN".



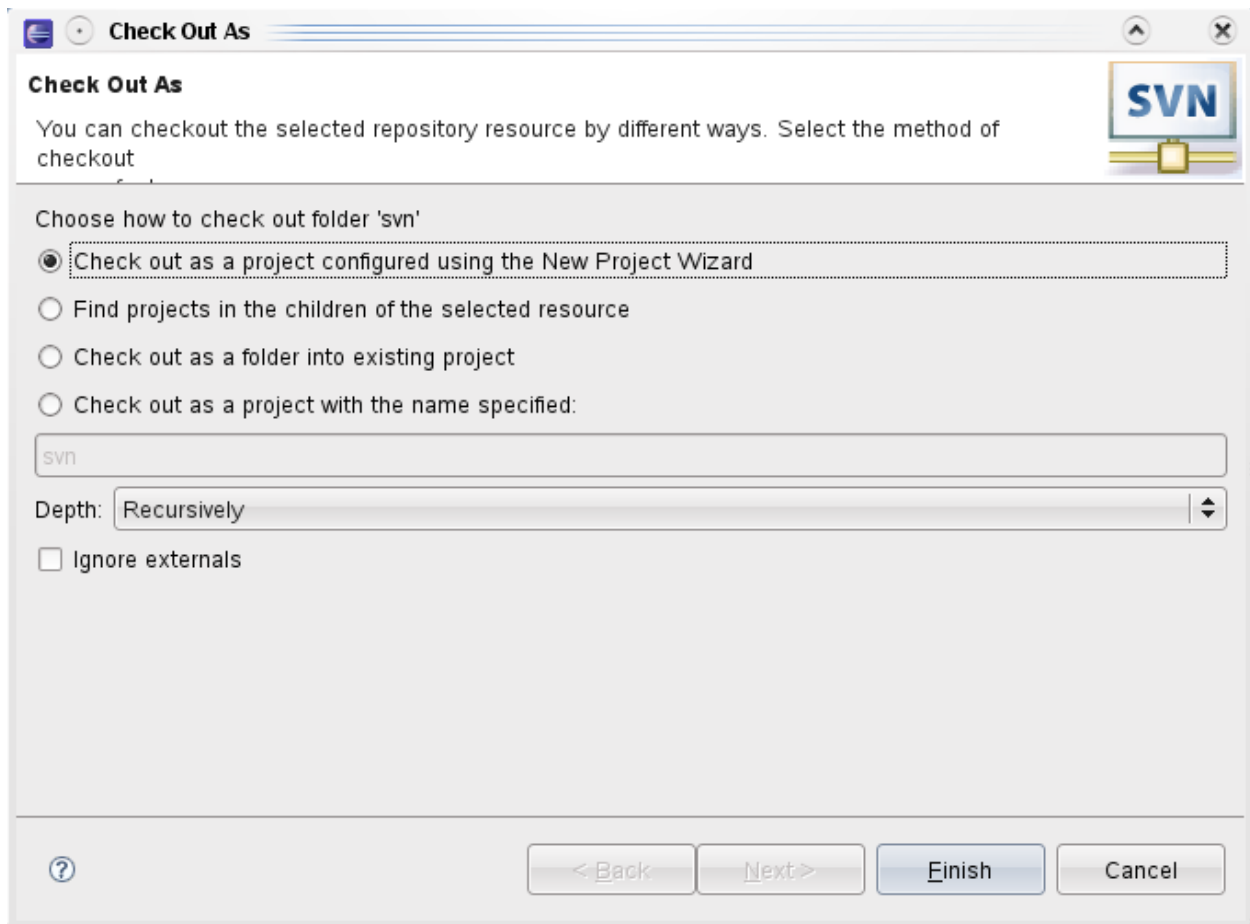
Cliquez sur "Next" et sélectionnez le répertoire SVN que vous venez d'ajouter.



Cliquez sur "Next" puis "Browse" (ou "parcourir" en anglais). Sélectionnez alors **trunk** (et rien d'autre!).



Après avoir bien sélectionné **trunk**, cliquez sur OK puis sur "Finish". Vous vous trouvez alors dans la fenêtre "Check out as". Sélectionnez "Check out as a project configured using the new project wizard" (ça devrait être sélectionné par défaut).



Cliquez ensuite sur "Finish".

Partie informatique

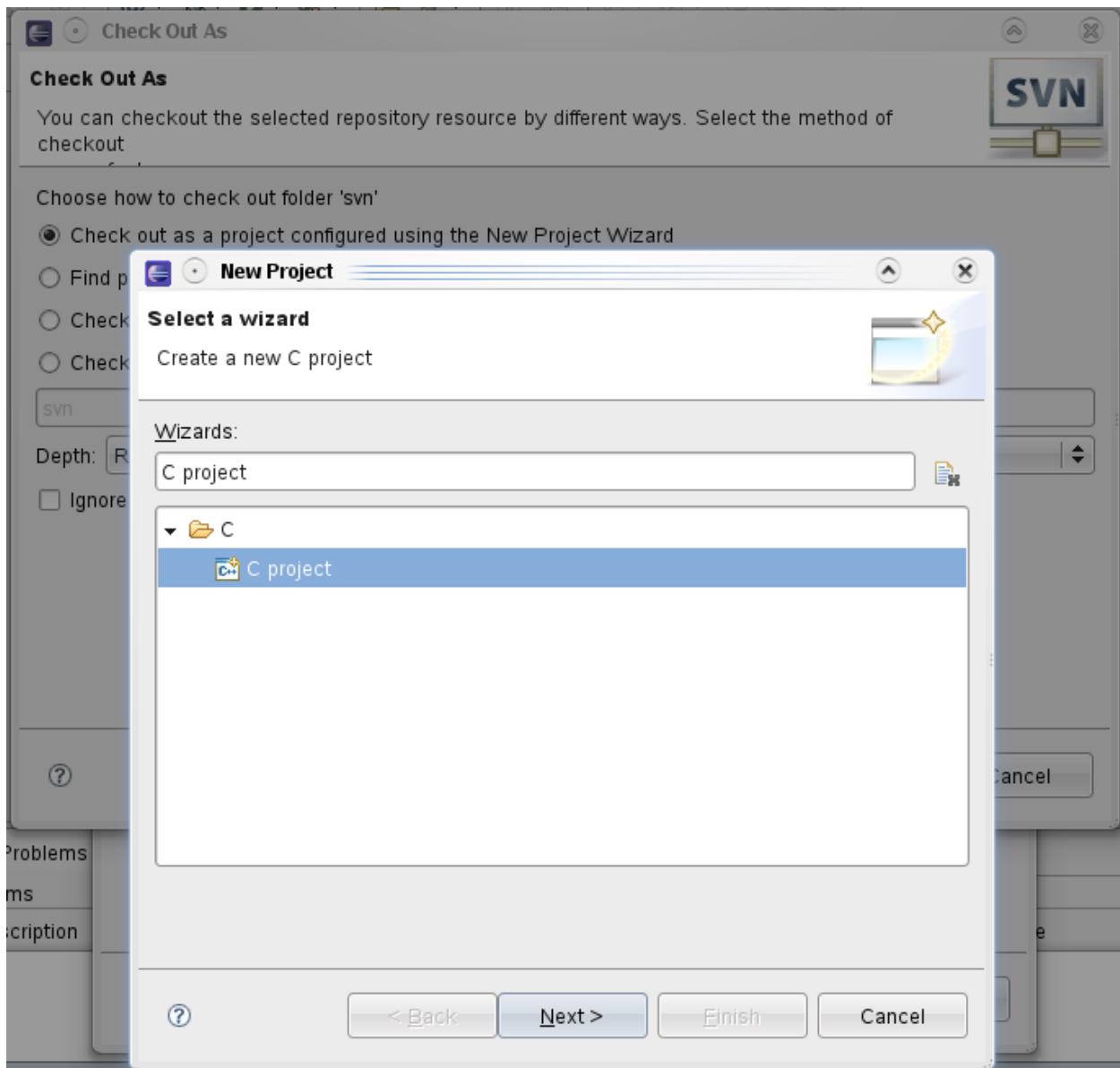
Après avoir cliqué sur "Finish" à l'étape précédente, une nouvelle fenêtre s'ouvre pour proposant de créer un nouveau projet. Là, il faut sélectionner "Java project", puisque vous allez développer en Java.

Cliquez sur "Next" et enfin remplissez le nom du projet (Sensitive Home me parait bien).

Partie électronique

Ce qui suit ne concerne que les développeurs des microcontrolleurs. Si vous n'avez pas encore installé le plugin AVR pour Eclipse, il serait temps de le faire, vous en aurez besoin.

Après avoir cliqué sur "Finish" à l'étape précédente, une nouvelle fenêtre s'ouvre pour proposant de créer un nouveau projet. Là, il faut sélectionner "C project":



Cliquez sur "Next" et enfin remplissez le nom du projet (Sensitive Home me paraît bien). Pour terminer, prenez garde à bien sélectionner "AVR Cross Target Application" et cliquez sur "Finish".

Utilisation du SVN

Ca y est, tout devrait être opérationnel. Voyons comment utiliser le SVN avec Eclipse.

Lorsque vous voulez créer un nouveau fichier, il faut impérativement le faire dans le répertoire "trunk", nulle part d'autre!

Lorsque vous souhaitez mettre à jour vos fichiers ou dossiers, faites un clic droit dessus puis allez dans le menu "Team" et cliquez sur "Update". Cela mettra vos fichiers et dossier à jour

par rapport à ce qui est sur le serveur. Il est fort recommandé de faire cette étape dès l'ouverture de Eclipse et sur le dossier principal.

Pour mettre à jour le serveur avec des fichiers/dossiers que vous avez modifié ou créé, aller dans le menu Team puis "Comit". N'oubliez pas de mettre un commentaire (utile de préférence). Pensez à mettre à jour vos fichier (Update) avant de mettre à jour le serveur (Comit) et vérifiez que ça compile.

Il se peut qu'il y ait des conflits de modification de fichiers. Dans ce cas, il est préférable de prévenir le développeur ayant codé le fichier en question. Si vous êtes sûr que vos modifications n'affecteront pas le reste du code, mettez quand même à jour le serveur.

Conclusion

J'espère que cette documentation vous sera utile. N'hésitez pas à me poser des questions.