
Journal de bord *Simulateur* *afficheur Velleman K8101*

Elèves :

M. Alan DEVAUD

Enseignats :

M^{me} Anne TERRIER

M. Christophe

MARÉCHAL

Version 1.0 du
21 novembre 2016

Table des matières

1	Journal de bord DEVAUD	2
1.1	5 septembre 2016	2
1.2	12 septembre 2016	2
1.3	19 septembre 2016	2
1.4	26 septembre 2016	2
1.5	3 octobre 2016	2
1.6	10 octobre 2016	2
1.7	17 octobre 2016	3
1.8	31 octobre 2016	3
1.9	07 novembre 2016	3
1.10	14 novembre 2016	4
1.11	21 novembre 2016	4

1 Journal de bord DEVAUD

1.1 5 septembre 2016

- Présentation du cours
- Rédaction cahier des charges avec *Dylan WACKER*

1.2 12 septembre 2016

- Commencement de la rédaction du rapport

1.3 19 septembre 2016

- Rédaction du rapport sur sharelatex
- Recherche de l'utilisation des ports USB avec C#
 - L'USB n'est pas implémenté nativement en C#
 - Utilisation d'une *library* obligatoire
- Création d'un diagramme de classe pour créer l'affichage

1.4 26 septembre 2016

- Découpage de l'affichage en plusieurs classe respective
 - *SimulatorButton* (Bouton)
 - *SimulatorK8101* (L'afficheur, regroupe les classes qui le constitue)
 - *SimulatorLCD* (L'écran LCD de l'affichage)
- Implémentation des class *SimulatorButton*, *SimulatorK8101* et *SimulatorLCD*
- Décision commune de mettre en place une architecture MVC pour notre projet

1.5 3 octobre 2016

- Création du projet Visual Studio
- Création du *repository* sur github et partage avec *Dylan WACKER*
 - <https://github.com/Devaud/SimulateurAfficheurK8101>
- En sniffant les trames USB, j'ai remarqué que le Velleman communiquait en Serial et non avec le protocol USB
- Continuation du rapport

1.6 10 octobre 2016

- Continuation du rapport

- Insertion du menu au projet
- Renommage du projet Visual Studio en *SimulatorK8101* pour garder la même chart de nomage anglaise

1.7 17 octobre 2016

- Continuation du rapport
- Recherche des trams envoyé par le Velleman K8101
- Analyse de la DLL de la bibliothèque Velleman K8101
 - Utilisation du logiciel *.NET Reflector* pour lire la dll
 - Découverte de l'utilisation de la class *ManagementScope* et *ManagementClass*
- Propriétés du sérial port :
 - *Baudrate* : 9600
 - *Stopbits* : 1
 - *Databits* : 8
 - *Parity* : None
 - *Readtimeout* : 100
- *Device* identifiant pour la *Management* :
 - *PNPDeviceID* : VID_10CF&PID_8101
 - *Device_ID* : nom du serial port (ex. : COM1)
- *Management* aide :
 - <https://msdn.microsoft.com/en-us/library/ms143592%28v=vs.110%29.aspx>
 - <https://msdn.microsoft.com/en-us/library/system.management%28v=vs.110%29.aspx>
 - <https://msdn.microsoft.com/en-us/library/system.management.propertydatacollection%28v=vs.110%29.aspx>

1.8 31 octobre 2016

- Installation de *JetBrains dotPeek* pour décompiler les DLL. *.Net Reflector* ayant la license expirée (trial)
- Après de multitude teste, nous décidons de créer une classe POC¹ pour comprendre le mécanisme du sérial port

1.9 07 novembre 2016

- Récapitulation de ce que doit contenir le rapport par M.Maréchal

1. Proof Of Concept

- Correction du rapport
- Recherche d'information sur le WMI (Windows Management Instrumentation)
 - <http://dotnet.developpez.com/articles/wmi1/#>
- Demande d'aide à M.Maréchal. Suite à cette aide, une dll qui simule la .dll K8101 sera créée. L'utilisateur aura plus qu'à utiliser la dll de base ou la dll de simulation. L'utilisation des programme externes de Velleman ne sera pas utilisé par celle-ci.

1.10 14 novembre 2016

- Création de la solution du projet
- Création de 3 projet
 - *SimulatorDisplayerK8101* : Application qui simule l'affichage du Velleman K8101
 - *SimuK8101* : DLL de simulation (Simule K8101.DLL)
 - *TestProgram* : Programme test pour la communication et l'affichage
- Utilisation de TcpClient et TcpListener pour la connexion entre les deux applications

1.11 21 novembre 2016

- Problème de connexion rencontrer avec le TcpListener pour la lecture du Stream
- Suppression des TcpListener et TcpClient pour utiliser des sockets
- Connexion des sockets sur l'adresse Ip 127.0.0.1 et le port 500
- Passage de l'application en MVC
- Après concertation avec Dylan Wacker nous avons décider de séparer le travail en deux. Lui s'occupera du SimuK8101 et moi du SimulatorDisplayerK8101
- Création de la connexion entre les deux applications
- Création d'un thread pour recevoir des messages
- Suppression de code en trop
- Problème de Merge rencontrer. Des bouts de code se sont faufler où ils ne devaient pas
- Corection du merge
- Vérification de la documentation