

2018

# Convertisseur de bases pour des nombres entiers et réels, codés sur 32 bits

Carbonara Christian – CIN4B

ETML

18/05/2018

## Table des matières

1. Remerciements.....	3
2. Spécifications .....	3
2.1. Titre .....	3
2.2. Description .....	3
2.3. Matériel et logiciels à disposition .....	4
2.4. Prérequis .....	4
2.5. Cahier des charges .....	4
2.5.1. Objectifs et portée du projet (objectifs FFOR) .....	4
2.5.2. Caractéristiques des utilisateurs et impacts .....	4
2.5.3. Fonctionnalités requises (du point de vue de l'utilisateur) .....	4
2.5.4. Contraintes .....	5
2.5.5. Travail à réaliser par l'apprenti .....	5
2.5.6. Si le temps le permet .....	5
2.6. Les points suivants seront évalués .....	5
3. Planification initiale.....	6
4. Analyse.....	6
4.1. Opportunités .....	6
4.2. Document d'analyse et conception .....	6
4.2.1. Les conversions minimal/maximal avec 32 bits signés .....	6
4.2.2. Maquette graphique de manière générale .....	6
4.2.3. Maquette graphique pour les conversions.....	7
4.2.4. Conception du programme .....	9
4.3. Conception des tests.....	10
4.4. Planification détaillée .....	10
5. Réalisation .....	13
5.1. Dossier de réalisation .....	13
5.1.1. Logiciels installé / Utilisés .....	13
5.2. Modifications .....	13
6. Tests.....	13
6.1. Dossier des tests .....	13
7. Conclusion.....	13
7.1. Bilan des fonctionnalités demandées.....	13
7.2. Bilan de la planification.....	13
7.3. Bilan personnel.....	14
8. Divers .....	14

---

8.1. Journal de travail .....	14
8.2. Bibliographie .....	14
8.3. Webographie.....	14
9. Annexes .....	14

## 1. REMERCIEMENTS

## 2. SPÉCIFICATIONS

### 2.1. TITRE

Convertisseur de bases (binaire, octal, décimal et hexadécimal) pour des nombres entiers et réels (à virgules fixes), codés sur 32 bits

### 2.2. DESCRIPTION

Il s'agit d'implémenter une application, en C#, qui va permettre aux utilisateurs (des élèves et des enseignants), de saisir une base numérique (bases binaires, octales, décimales, hexadécimales, nombres à virgules fixes), un nombre et le programme affiche le même nombre dans les autres bases. De plus, le programme doit permettre d'accomplir des opérations élémentaires d'additions et soustractions sur des nombres entiers positifs et négatifs.

Le dispositif indiquera les détails des opérations, et se fera jusqu'à 32 bits signés pour les nombres binaires. En aucun cas l'utilisation de fonctions prédéfinies du langage ne sera acceptée. Les élèves doivent parfois résoudre des exercices, ou des situations, qui nécessitent des conversions de nombres d'une base à une autre. Ou encore, faire des additions ou des soustractions.

Par exemple, comment convertir un nombre réel à virgule fixe de décimal en hexadécimal (ou  $134.45^{10} \rightarrow ?.^{16}$ ).

Actuellement, à l'ETML, en informatique, un module nommé ELEOC-NUM pour électronique numérique, traite ce sujet. Les supports présentent de la théorie et des exercices. Toutefois, il conviendrait de pouvoir laisser les élèves résoudre des mêmes exercices, mais avec des données différentes, et de pouvoir vérifier leurs réponses de manière automatique.

L'idée est d'avoir une application qui permette aux élèves de s'entraîner pour les fonctionnalités suivantes, avec des mots de 32 bits :

- Conversions de nombres décimaux, binaires, octaux et hexadécimaux (entiers et/ou réels à virgules fixes) en nombres dans les 3 autres bases que la base courante.
- Additions et soustractions de 2 nombres binaires, octaux et hexadécimaux (entiers, positifs et/ou négatifs)
- Conversion d'un nombre réel à virgule fixe, décimal, positif ou négatif, en nombre binaire à virgule fixe et réciproquement.

Le candidat devra d'abord analyser les domaines d'applications du sujet pour chaque base considérée, afin que le tout soit cohérent. Par exemple,

avec 32 bits binaires signé, quel minima et quel maxima peut-on utiliser pour convertir en hexadécimal ?

### 2.3. MATÉRIEL ET LOGICIELS À DISPOSITION

1 ordinateur standard ETML, avec la structure habituelle.

### 2.4. PRÉREQUIS

Avoir suivi les modules ELEOC, et de programmation en C#.

### 2.5. CAHIER DES CHARGES

#### 2.5.1. Objectifs et portée du projet (objectifs FFOR)

**Force :**

- Possède déjà des idées concernant les possibilités pour coder le programme afin d'effectuer des conversions, additions et soustractions

**Faiblesse :**

- Difficulté à rédiger le rapport

**Opportunité :**

- Créer un programme qui sera repris par l'ETML pour effectuer des corrections, vérifications et aider les élèves ainsi que les enseignants

**Risque :**

- Manque de temps/connaissances

#### 2.5.2. Caractéristiques des utilisateurs et impacts

Les utilisateurs sont en 1<sup>ère</sup> année de l'ETML ainsi que très probablement des utilisateurs d'autres années, certains n'ont pas de connaissances poussées de l'informatique, il faut donc que le programme soit simple d'utilisation et intuitif pour tous.

Le programme peut aider les élèves à se corriger, à comprendre ce qu'ils ont fait de faux et comment ils doivent faire pour effectuer une conversion, addition ou soustraction.

#### 2.5.3. Fonctionnalités requises (du point de vue de l'utilisateur)

Conversions des nombres décimaux, binaires, octaux et hexadécimaux (entier et/ou réels à virgules fixes) en nombres dans les 3 autres bases que la base courante

Additions et soustractions de 2 nombres binaires, octaux et hexadécimaux (entiers, positifs et/ou négatifs)

Conversion d'un nombre réel à virgule fixe, décimal, positif ou négatif, en nombre binaire à virgule fixe et réciproquement.

Possibilité de voir le calcul entier, que ce soit pour la conversion, l'addition et la soustraction.

#### 2.5.4. Contraintes

Les fonctions prédéfinies ne sont pas autorisées et ne doivent en aucun cas être utilisées.

#### 2.5.5. Travail à réaliser par l'apprenti

L'apprenti doit créer un programme permettant aux élèves ainsi qu'aux enseignants de corriger les exercices d'ELEOC, le programme doit donc permettre la conversion de nombre positifs ou négatifs à virgule fixe entre différents formats (décimal, binaire, octal et hexadécimal), en plus de cela il doit être possible de soustraire et additionner des nombres entiers positifs et/ou négatifs dans différents formats (binaire, octal et hexadécimal), tout cela doit être simple d'utilisation pour chaque élève et enseignants de l'ETML en plus d'être intuitif.

Une planification doit être faite au début du projet avec un Gantt, un journal de travail doit aussi être fait et tenu à jour. Un rapport doit être fait et complet, il doit contenir l'intégralité du projet.

Tout le projet doit être mis sur un dépôt Git que l'élève pourra choisir par lui-même.

Une analyse du programme concernant l'application et ses fonctionnalités doit être faite, permettant de s'assurer de la cohérence du programme entre ses différentes fonctionnalités et utilisations.

#### 2.5.6. Si le temps le permet

Améliorer l'apparence du programme pour le rendre visuellement plus beau et plus intuitif. Permettre de mettre en fichier PDF le résultat de l'opération ainsi que le calcul effectué pour l'obtenir.

## 2.6. LES POINTS SUIVANTS SERONT ÉVALUÉS

La grille d'évaluation définit les critères généraux selon lesquels le travail du candidat sera évalué (documentation, journal de travail, respect des normes, qualité, ...).

En plus de cela, le travail sera évalué sur les 7 points spécifiques suivants (Point A14 à A20):

- Conversion de nombres entiers binaires signés dans les autres bases
- Conversion de nombres réels signés, à virgule fixe, dans les autres bases
- Conversion de nombres décimaux signés dans les autres bases

- Conversion de nombres décimaux signés, à virgule fixe, dans les autres bases
- Conversion de nombres hexadécimaux signés dans les autres bases
- Ergonomie intuitive et agréable pour tout type d'utilisateur de l'ETML
- La réalisation doit prendre en compte au moins un point significatif des bonnes pratiques en matière d'écoconception

### 3. PLANIFICATION INITIALE

La [planification initiale](#) se trouve en annexe.

## 4. ANALYSE

### 4.1. OPPORTUNITÉS

Approfondir ses connaissances en C#, opportunité de créer un programme qui sera utilisé en ELEOC, par les élèves de l'ETML en première année voir des années suivantes, ainsi que les enseignants qui pourront corriger des exercices.

### 4.2. DOCUMENT D'ANALYSE ET CONCEPTION

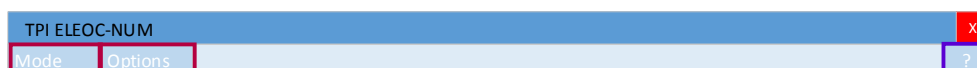
#### 4.2.1. Les conversions minimal/maximal avec 32 bits signés

	Décimal	Binaire	Octal	Hexadécimal
<b>Minimum</b>	-2147483648	1000000000000000 0000000000000000 Ou 1 0 <sup>x31</sup>	2000000000 Ou 2 0 <sup>x10</sup>	8000 0000 Ou 8 0 <sup>x7</sup>
<b>Maximum</b>	2147483647	0111111111111111 1111111111111111 Ou 0 1 <sup>x31</sup>	1777777777 Ou 1 7 <sup>x10</sup>	7FFF FFFF Ou 7 F <sup>x7</sup>

#### 4.2.2. Maquette graphique de manière générale

Les maquettes qui seront affichés représentent ce que le programme devait ressembler au début du projet, il est possible que ce ne soit pas le cas et que pour certaines raisons le produit final diffère de celles-ci.

Chaque interface possède la même barre d'action se trouvant tout en haut de l'interface du programme, comprenant **2 boutons avec une liste déroulante** et **1 bouton simple**. (Barre d'action affichée ci-dessous)



Le premier bouton avec une liste déroulante se nomme « **Mode** », cela permet à l'utilisateur de changer d'interface suivant l'opération voulu, il n'aura qu'à cliquer dessus pour qu'une liste apparaisse laissant le choix du mode pour convertir, additionner ou soustraire.

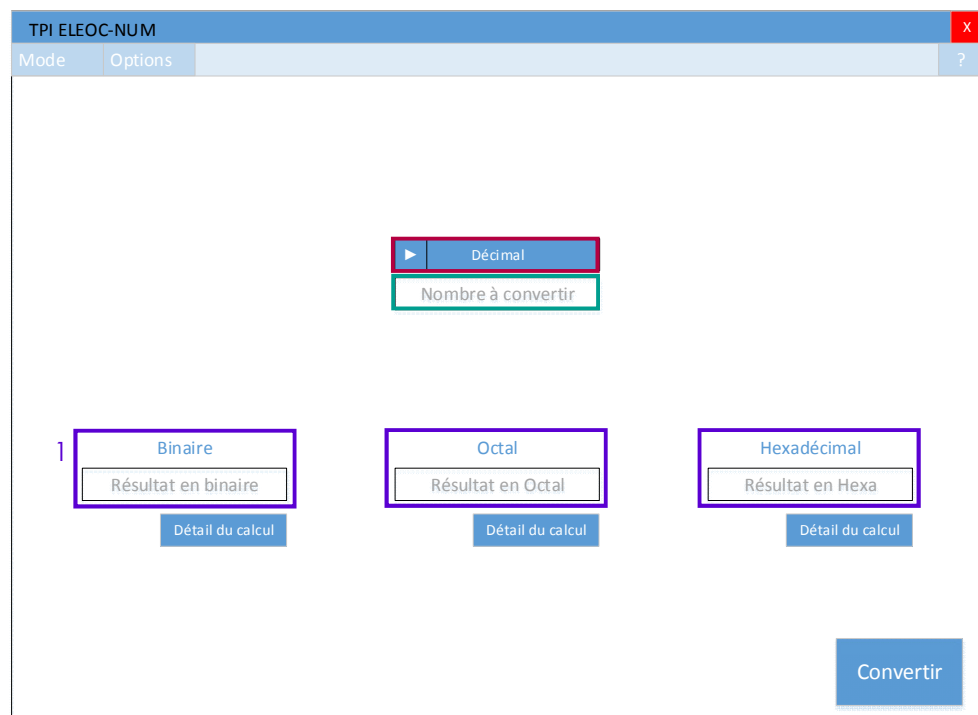
Le second bouton qui lui aussi est avec une liste déroulante nommé « Options » permettra à l'utilisateur de changer le format de son nombre, par exemple de choisir s'il veut effectuer la conversion voulu avec notamment un nombre à virgule ou non, il pourra aussi définir s'il s'agit d'un nombre signé ou non, cela permettra aux utilisateurs d'effectuer leurs opérations suivent ce qu'ils veulent et ce qu'ils ont appris, de plus cela évitera d'afficher des informations ou données qu'ils n'ont pas encore appris évitant ainsi de les embrouiller.

Un bouton d'aide « ? » sera disponible pour indiquer des informations concernant les nombres qu'ils peuvent entrer, le maximum et minimum possible, des aides concernant le manuel d'utilisation et de l'utilisation du programme.

#### 4.2.3. Maquette graphique pour les conversions

Pour la première maquette il s'agit de celle pour la conversion des nombres qui est l'interface qui s'affichera à l'utilisateur en ouvrant le programme.

Voici la maquette en question :



La maquette graphique de l'interface de conversion est présentée dans une fenêtre intitulée "TPI ELEOC-NUM". Elle comporte une barre de menu avec "Mode" et "Options", et un bouton d'aide "?". Le contenu principal est divisé en plusieurs sections :

- En haut à droite, un bouton "Dédimal" (surligné en rouge) et un champ "Nombre à convertir" (surligné en vert).
- En bas à gauche, une section "Binaire" (surlignée en violet) avec un bouton "Résultat en binaire" (surligné en violet) et un bouton "Détail du calcul" (surligné en violet). Une note (1) est placée à gauche de cette section.
- En bas au centre, une section "Octal" (surlignée en violet) avec un bouton "Résultat en Octal" (surligné en violet) et un bouton "Détail du calcul" (surligné en violet).
- En bas à droite, une section "Hexadécimal" (surlignée en violet) avec un bouton "Résultat en Hexa" (surligné en violet) et un bouton "Détail du calcul" (surligné en violet).
- En bas à droite, un bouton "Convertir" (surligné en bleu).

Concernant le nombre à convertir, il est possible de définir son format (binaire, décimal, octal ou hexadécimal) via la liste déroulante, en choisissant un format cela modifiera par la même occasion les autres champs où seront affichés les résultats. Par exemple si l'utilisateur ne veut plus convertir un nombre décimal mais binaire, lorsqu'il aura sélectionné le format binaire via la liste déroulante, à ce moment-là la zone de résultat en binaire (1) deviendra un résultat en décimal avec non pas le titre « Binaire » mais bien « Décimal ».



L'utilisateur entre la valeur qu'il veut contenir dans la zone de texte mis à sa disposition, cependant si l'utilisateur a choisi que le nombre à convertir est un nombre à virgule il y aura 2 zones de texte l'une à côté de l'autre, entre eux un point servira à les séparer en indiquant par la même occasion où se trouve la virgule, dans la première zone il s'agira du nombre se trouvant avant la virgule, la deuxième zone sera donc celle pour les nombres se trouvant après la virgule.

Voici ce à quoi ressemble l'interface lorsque l'utilisateur choisi de convertir un nombre à virgule :

Décimal

Nombre avant la virgule

.

Nombre après la virgule

Même si les calculs ne sont pas affichés de base il est cependant possible d'y avoir accès via le bouton « Détail du calcul »,

Seul les résultats de la conversion sont affichés et non les calculs pour y parvenir, cela dans le but de ne pas afficher beaucoup d'informations à l'écran évitant d'embrouiller l'utilisateur et par la même occasion que cela soit illisible.

Pour chaque conversion il est possible de voir le calcul, pour cela il suffit de cliquer sur détail du calcul. Voici un exemple de conversion de décimal à binaire :

Détail du calcul en binaire
✕

97/2 = 48
48/2 = 24
24/2 = 12
12/2 = 6
6/2 = 3
3/2 = 1
1/2 = 0

Reste 1
Reste 0
Reste 0
Reste 0
Reste 0
Reste 1
Reste 1

Tableau de puissance

$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
1	1	0	0	0	0	1

Résultat

97 = 1100001

↑

↓

Lire de bas en haut

Les étapes sont tous affichées, le « Reste » est affiché et une flèche partant de bas en haut indique dans quel sens il doit être lu, cela permet

de comprendre comment faire le calcul soi-même et de comprendre son erreur lors des corrections notamment, ainsi si l'utilisateur n'est pas à l'aise en ELEOC ou qu'il a de la peine, cela lui permettra de s'améliorer et de l'aider. Un tableau des puissances où sont entré les restes est visible, en dessous se trouve le résultat du calcul.

#### 4.2.4. Conception du programme

Le programme a besoin de pouvoir effectuer différentes opérations de calculs (conversion, addition et soustraction), afin d'éviter que cela nécessite trop de ressource ces opérations seront séparées, le programme possèdera des « modes », cela permettra de n'effectuer que l'opération voulu en affichant uniquement l'interface du « mode » choisi sans afficher l'interface des autres, pour changer de « mode » le programme possèdera une barre d'action avec un « Split Button », il s'agit d'un bouton avec une liste déroulante qui permettra de choisir le « mode » désiré pour soit convertir, additionner ou soustraire.

Dans cette même barre d'action il y aura encore un autre « Split Button », celui-ci servira à choisir différentes options comme la possibilité de définir notamment si l'on veut convertir des nombres avec virgule ou non, permettant aux élèves n'étant pas à l'aise en ELEOC de ne pas s'embrouiller à remplir de zéro ou de ne rien mettre dans la case après la virgule, donnant par la même occasion un résultat qui n'affichera aucune virgule évitant aussi de s'embrouiller et les erreurs de manipulation en entrant des chiffres après la virgule par erreur.

Pour la conversion il y aura donc comme vu au-dessus la possibilité de le faire sans virgule donnant uniquement des conversions avec des chiffres entiers, il n'y aura donc qu'une boîte de texte où l'utilisateur entrera la valeur qu'il veut convertir. Si l'utilisateur choisi d'effectuer une conversion avec virgule, une deuxième boîte de texte apparaîtra à droite de celle de base, un point s'affichera entre ces deux, il s'agit de la virgule pour ainsi les séparer en indiquant que la deuxième contiendra ce qu'il y a après la virgule, la séparation de ce qu'il y a avant la virgule et après permet d'éviter des erreurs du programme via notamment les regex qui seront utilisé pour vérifier le format des valeurs entrés ainsi que faciliter la conversion de ce qu'il y a avant la virgule et après.

Au sujet de la façon dont les conversions seront faites, des tableaux seront utilisés pour convertir, un algorithme sera mis en place pour afficher à l'utilisateur le calcul effectué et la démarche pour réussir à obtenir le résultat, le but est que l'utilisateur puisse obtenir le résultat avec les démarches nécessaire au calcul mais aussi qu'il puisse comprendre comment l'atteindre, cela lui permettra de se corriger, de plus s'il a des difficulté pour effectuer le calcul de lui-même il faut que le programme en affichant le calcul puisse le guider et l'aider à comprendre comment il doit faire.

Les additions et les soustractions utiliseront aussi des tableaux,

### 4.3. CONCEPTION DES TESTS

Test à effectuer	Ce qui est recherché
Faire des conversion, addition et soustraction avec des lettres ou autre caractères lorsque le format ne doit pas le permettre	<ol style="list-style-type: none"> <li>1. Vérifier s'il y a des erreurs ou crash du programme</li> <li>2. Empêcher une mauvaise manipulation</li> </ol>
Vérifier les limites possibles concernant la taille des calculs ou des nombres	<ol style="list-style-type: none"> <li>1. Vérifier s'il y a des erreurs ou crash du programme</li> <li>2. Limiter la taille maximal si besoin</li> </ol>
Fournir le manuel d'utilisation avec le programme à plusieurs personnes pour qu'elles utilisent le programme	<ol style="list-style-type: none"> <li>1. Vérifier la simplicité d'utilisation du programme</li> <li>2. S'assurer que le manuel d'utilisation soit clair et complet</li> </ol>
Tester sur différent PC n'ayant pas la même configuration	<ol style="list-style-type: none"> <li>1. Voir comment le programme réagis, s'il est bien compatible et qu'il n'y a aucune erreur</li> <li>2. Si ce n'est pas le cas il sera possible de soit corriger soit indiqué pourquoi il ne fonctionne pas</li> </ol>
Essayer de convertir, additionner et soustraire avec un nombre supérieur à 32 bits	<ol style="list-style-type: none"> <li>1. Voir si le programme empêche bien la manipulation lorsque la limite est dépassée</li> </ol>
Convertir, additionner et soustraire de petites valeurs jusqu'au maximum possible	<ol style="list-style-type: none"> <li>1. Vérifier si les fonctionnalités du programme fonctionnent correctement</li> <li>2. Vérifier si une petite valeur pose problème et inversement avec une grande valeur</li> </ol>
Effectuer les calculs avec nombres positifs et négatifs ainsi qu'avec virgule	<ol style="list-style-type: none"> <li>1. Vérifier le bon fonctionnement des calculs</li> </ol>

### 4.4. PLANIFICATION DÉTAILLÉE

La colonne concernant les liens et références de la planification n'a pas été reprise ci-dessous, la raison étant qu'il n'y a aucune information écrite sous celle-ci, de plus en retirant ces colonnes cela permet une meilleure visibilité.

Semaine 1		
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?
Lundi - 7 mai	12	Max. 12 (3h)
Entretien	2	Entretien concernant le TPI.
Planification initiale	8	Création de la planification initiale et mise à jour.
Journal de travail	2	Création du journal de travail et mise à jour.
Mercredi - 9 mai	27	Max. 27 (4h45)
Absence - Imprévu	15	Joutes sportives.
Planification initiale	4	Finalisation de la planification initiale.
Journal de travail	2	Mise à jour du journal de travail.
Rapport	4	Création du rapport, rédaction de la partie Analyse et Planification initiale, mise à jour avec les informations reçu.
Réalisation de croquis	2	Création de croquis avec Visio ou un autre logiciel pour le programme, emplacement des boutons, fonctionnalités, onglets, différentes pages,
Jeudi - 10 mai	24	Max. 24 (4h)
Congé - Vacances	24	Ascension.
Vendredi - 11 mai	27	Max. 27 (4h45)
Congé - Vacances	27	Pont avec l'Ascension.
Total semaine	90	Max. 90 (22h30)

Semaine 2		
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?
Lundi - 14 mai	12	Max. 12 (3h)
Réalisation de croquis	4	Finalisation des croquis du programme.
Rapport	6	Finalisation de la partie Analyse, planification initiale.
Entretien	2	Entretien avec M. Gruaz concernant l'avancement du projet.
Mercredi - 16 mai	27	Max. 27 (4h45)
Journal de travail	1	Mise à jour du journal de travail.
Recherche	8	Recherche concernant la conversion du décimal au binaire et inversement.
Coder le programme	12	Création du programme avec les norme ETML et des commentaires. Mise en place de l'interface pour convertir et préparation pour les futurs opérations (addition, soustraction, etc...). Ajout d'options pour choisir entre les conversions. Ajout de commentaire.
Coder pour convertir	6	Mettre en place les paramètres et le code pour convertir du décimal à binaire et inversement.
Jeudi - 17 mai	24	Max. 24 (4h)
Journal de travail	1	Mise à jour du journal de travail.
Recherche	8	Recherche concernant la conversion en octal et hexadécimale.
Coder pour convertir	8	Finalisation et correction du code pour convertir de décimal à binaire et inversement.
Coder le programme	7	Avancement dans le programme.
Vendredi - 18 mai	27	Max. 27 (4h45)
Journal de travail	1	Mise à jour du journal de travail.
Coder pour convertir	6	Finalisation de la conversion en octal et hexadécimale.
Coder le programme	8	Correction du code, vérification du bon fonctionnement et ajustement au norme ETML.
Réalisation de croquis	4	Création du croquis pour les additions.
Rapport	8	Avancement du rapport avec les éléments actuel.
Total semaine	90	Max. 90 (22h30)



Semaine 3		
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?
Lundi - 21 mai		Max. 12 (3h)
Congé - Vacances	12	Lundi de Pentecôte.
Mercredi - 23 mai		Max. 27 (4h45)
Journal de travail	1	
Entretien	2	Entretien avec M. Gruaz concernant l'avancement du projet.
Recherche	12	Recherche sur un moyen d'additionner 2 nombres binaire.
Coder le programme	4	Mise en place de l'interface pour additionner.
Coder pour additionner	8	Avancement dans l'addition avec 2 nombre binaire.
Jeudi - 24 mai		Max. 24 (4h)
Journal de travail	1	Mise à jour du journal de travail.
Recherche	9	Recherche pour additionner avec 2 nombres binaire, octaux et
Coder le programme	6	Avancement du code avec correction.
Coder pour additionner	8	Finalisation pour additionner avec 2 nombres binaires, octaux et hexadécimaux.
Vendredi - 25 mai		Max. 27 (4h45)
Journal de travail	1	Mise à jour du journal de travail.
Réalisation de croquis	4	Création du croquis pour soustraire.
Coder le programme	8	Correction du code si nécessaire et préparation de l'interface pour
Coder pour soustraire	4	Préparation pour soustraire 2 nombres binaires.
Tests	3	Test sur le bon fonctionnement du programme.
Rapport	7	Avancement du rapport avec les éléments actuels.
Total semaine	90	Max. 90 (22h30)

Semaine 4		
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?
Lundi - 28 mai		Max. 12 (3h)
Journal de travail	1	Mise à jour du journal de travail.
Entretien	2	Entretien avec M. Gruaz concernant l'avancement du projet.
Manuel d'utilisation	4	Création du manuel d'utilisation.
Recherche	5	Recherche pour soustraire 2 nombres binaires.
Mercredi - 30 mai		Max. 27 (4h45)
Journal de travail	1	Mise à jour du journal de travail.
Recherche	10	Recherche pour soustraire 2 nombres binaires, octaux et hexadécimaux.
Coder pour soustraire	10	Avancement du code pour soustraire 2 nombres binaires, octaux et
Coder le programme	6	Avancement dans le code avec correction.
Jeudi - 31 mai		Max. 24 (4h)
Journal de travail	1	Mise à jour du journal de travail.
Coder pour soustraire	10	Finalisation du code pour soustraire 2 nombres binaires, octaux et
Coder le programme	9	Correction, vérification du code et ajustement si nécessaire.
Tests	4	Vérification des normes de codage de l'ETML.
Vendredi - 1 juin		Max. 27 (4h45)
Journal de travail	1	Mise à jour du journal de travail.
Rapport	12	Avancement de le rapport avec les informations actuels.
Manuel d'utilisation	6	Vérification et correction du rapport.
Coder le programme	6	Avancement dans le manuel d'utilisation.
Tests	2	Vérification du code et correction si nécessaire.
Total semaine	90	Max. 90 (22h30)

Semaine 5		
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?
Lundi - 4 juin		Max. 12 (3h)
Journal de travail	1	Mise à jour du journal de travail.
Rapport	6	Avancement dans le rapport.
Manuel d'utilisation	3	Avancement du manuel d'utilisation.
Entretien	2	Entretien avec M. Gruaz concernant l'avancement du projet.
Mercredi - 6 juin		Max. 27 (4h45)
Journal de travail	3	Mise à jour du journal de travail avec correction si nécessaire.
Rapport	18	Finalisation du rapport.
Manuel d'utilisation	6	Finalisation du manuel d'utilisation.
Jeudi - 7 juin		Max. 24 (4h)
Journal de travail	4	Mise à jour du journal de travail avec correction si nécessaire.
Rapport	10	Vérification du rapport et correction si nécessaire.
Manuel d'utilisation	6	Vérification du manuel d'utilisation et correction si nécessaire.
Fin du projet	4	Vérification que tout soit en ordre, envoi du projet au chef de projet et
Vendredi - 8 juin		Max. 27 (4h45)
Fin du projet	27	Projet terminé et rendu.
Total semaine	90	Max. 90 (22h30)

## 5. RÉALISATION

### 5.1. DOSSIER DE RÉALISATION

#### 5.1.1. Logiciels installé / Utilisés

Microsoft Visual Studio 2017

### 5.2. MODIFICATIONS

## 6. TESTS

### 6.1. DOSSIER DES TESTS

## 7. CONCLUSION

### 7.1. BILAN DES FONCTIONNALITÉS DEMANDÉES

### 7.2. BILAN DE LA PLANIFICATION

### **7.3. BILAN PERSONNEL**

## **8. DIVERS**

### **8.1. JOURNAL DE TRAVAIL**

### **8.2. BIBLIOGRAPHIE**

### **8.3. WEBOGRAPHIE**

Inspiration pour l'interface de la conversion :

[http://www.groupeisf.net/automatismes/Numeration/Numeration\\_binaire/Ressources/Logique\\_combinatoire/html/01/28-ess0102006.htm](http://www.groupeisf.net/automatismes/Numeration/Numeration_binaire/Ressources/Logique_combinatoire/html/01/28-ess0102006.htm)

## **9. ANNEXES**

[Planification initiale](#)