

MODULE 05

SÉANCE SYSTÈME 04

TP D'INFORMATIQUE

Durée 2h30

EXTRACTION DES DONNÉES FEUX : CLASSE EN C++

BLOC DE COMPÉTENCES

U6 - VALORISATION DE LA DONNÉE ET CYBERSÉCURITÉ

COMPÉTENCE(S)

C08 - CODER

OBJECTIFS PÉDAGOGIQUES

Extraction de l'état des feux. Codage d'une classe en C++ permettant d'extraire les données CAN du combiné du véhicule

CONNAISSANCES ISSUES DU RÉFÉRENTIEL

- | | |
|--|----------|
| • Langages de modélisation d'application | Niveau 3 |
| • Programmation orientée objet | Niveau 3 |
| • Programmation réseau | Niveau 3 |

CONNAISSANCES OPÉRATIONNALISÉES

- | | |
|--|----------|
| • Écrire une classe C++ à partir d'un diagramme de classes | Niveau 3 |
| • Versionner un code | Niveau 3 |

TP

Extraction de l'état des feux

Pour extraire l'état des feux, il faut convertir la chaîne ASCII hexa en nombre entier comme précédemment. L'octet ciblé est le 5ème octet de la trame d'identifiant "128".

Quelles indices de la chaîne de caractères trameCAN sont concernés ?

Dans le programme principal, extraire le bon octet de la trame en appelant la méthode `ChaineHexaVersInt` :

```
unsigned char octet = ChaineHexaVersInt(trameCAN+...,2).
```

Afficher et vérifier la valeur de cet octet.

Il faut maintenant extraire chaque bit de l'octet. D'après la documentation constructeur (voir SS02), les feux de route sont représentés par le bit 5, les bits allant de 0 (lsb) à 7 (msb) :

L'instruction : `route = octet & (0x01 << 5);` permet de stocker le bit 5 de la variable 8 bits "octet" dans la variable booléenne "route". En effet si b5 est à 1, `route=true`, sinon `route=false` :

octet	=	b7	b6	b5	b4	b3	b2	b1	b0
<u>0x01<<5</u>	=	0	0	1	0	0	0	0	0
&	=	0	0	b5	0	0	0	0	0

Donner les instructions permettant de stocker l'état du clignotant droit (bit 2), du clignotant gauche (bit 1), de l'anti-brouillard (bit 3) et des feux de croisement (bit 6) :

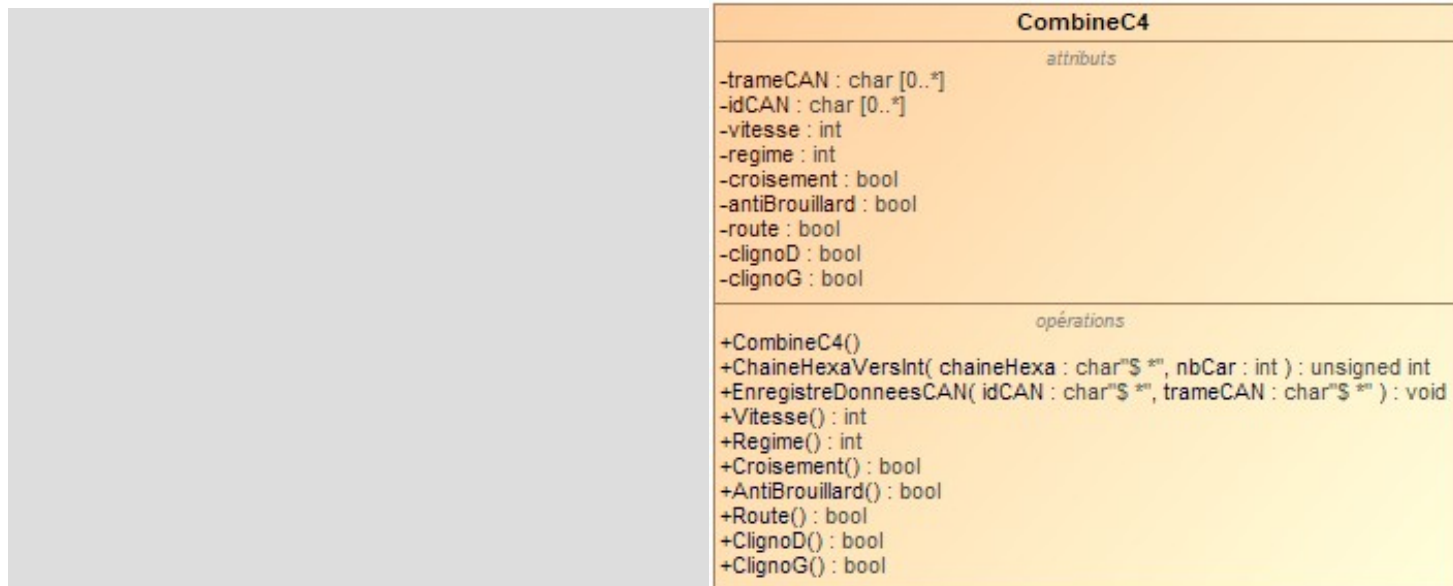
Modifier le programme principal en lui ajoutant l'extraction des feux puis l'affichage de leur état.
Votre code :

TP

Codage de la classe CombineC4

A partir du diagramme de classe donné, coder la déclaration de la classe dans CombineC4.h.

Votre code : (n'écrire que les attributs et méthodes relatives au moteur)



Codage de la définition des méthodes dans CombineC4.cpp :

Codage des méthodes d'accès

Les méthodes Vitesse(), Regime(), Route(), ClignoD(), ClignoG(), Croisement() et AntiBrouillard() sont chargées de retourner simplement l'attribut privé correspondant au nom de la méthode d'accès. Un exemple de code d'une des méthode d'accès :

```

// Exemple de code pour la méthode Vitesse()
int CombineC4::Vitesse() const {
    return vitesse;
}

```

Quelle fonction, en langage C, permet de comparer 2 chaînes de caractères ? Donner un exemple d'utilisation

```

// Exemple de code pour la fonction strcmp()
int main() {
    char s1[] = "Bonjour";
    char s2[] = "Bonjour";
    int result = strcmp(s1, s2);
    if (result == 0) {
        printf("Les chaînes sont égales.\n");
    } else {
        printf("Les chaînes sont différentes.\n");
    }
    return 0;
}

```

Codage de la méthode EnregistreDonneesCAN

En vous basant sur le programme principal codé précédemment : selon la valeur d'idCAN, décoder la vitesse, le régime ou bien l'état des feux en positionnant les valeurs dans les bons attributs.

Test de la classe CombineC4

Coder le programme principal en ajoutant la création (l'instanciation) d'un objet monCombine de la classe CombineC4 (au début du main). Après la réception de la trame, stocker la donnée dans l'objet CombinéC4 en appelant la méthode EnregistrerDonneeCAN. Il ne reste plus qu'à afficher les valeurs en appelant les méthodes d'accès Vitesse(), Regime(), Route()... Votre programme principal :

Bonus : scrutation automatique de l'état du véhicule

L'utilisateur ne saisie plus l'identifiant, mais ce dernier vaudra tantôt "0B6" tantôt "128". A chaque passage dans la boucle infinie du programme principal, si idCAN vaut "128" : copier "0B6" dans idCAN. Dans le cas contraire copier "128" dans idCAN. Quelque soit l'identifiant, calculer la vitesse et le régime, extraire l'état des feu, puis afficher toutes les informations du combiné. En fin de boucle attendre 1s (utiliser la fonction Sleep()). La fonction clrscr() permet d'effacer le contenu de la console.

Versionner le code complet.