

MODULE 05

SÉANCE SYSTÈME 05

TP D'INFORMATIQUE

Durée 2h30

ANDROID : ENVOI DE LA TRAME TCP PERMETTANT DE DEMANDER L'ÉTAT DU VÉHICULE

BLOC DE COMPÉTENCES

U6 - VALORISATION DE LA DONNÉE ET CYBERSÉCURITÉ

COMPÉTENCE(S)

C08 - CODER

OBJECTIFS PÉDAGOGIQUES

Installation d'Android Studio et test de déploiement sur tablette. Codage de l'envoi d'une trame TCP permettant de demander la trame CAN au serveur

CONNAISSANCES ISSUES DU RÉFÉRENTIEL

- Langages de développement, de description, de création d'API et les IDE associés Niveau 3
- Programmation orientée objet Niveau 3

CONNAISSANCES OPÉRATIONNALISÉES

- Installer un environnement de développement et de test pour appareil mobile sous Android Niveau 3
- Coder une application pour Android utilisant les sockets
- Versionner un code Niveau 3

TP

Android Studio :

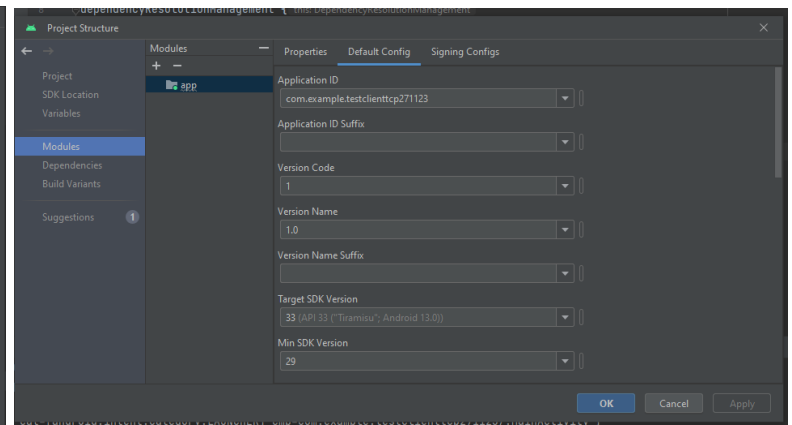
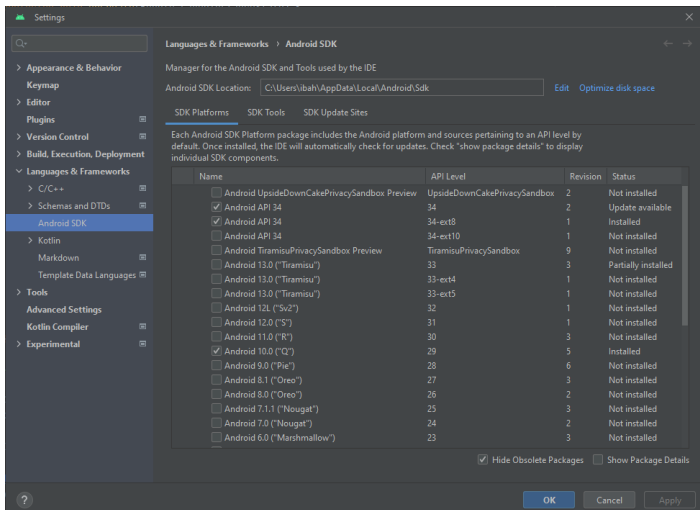
demande de la trame CAN au serveur TCP

Créer un projet AndroidStudio en native C++. Ajouter immédiatement dans le fichier AndroidManifest.xml :

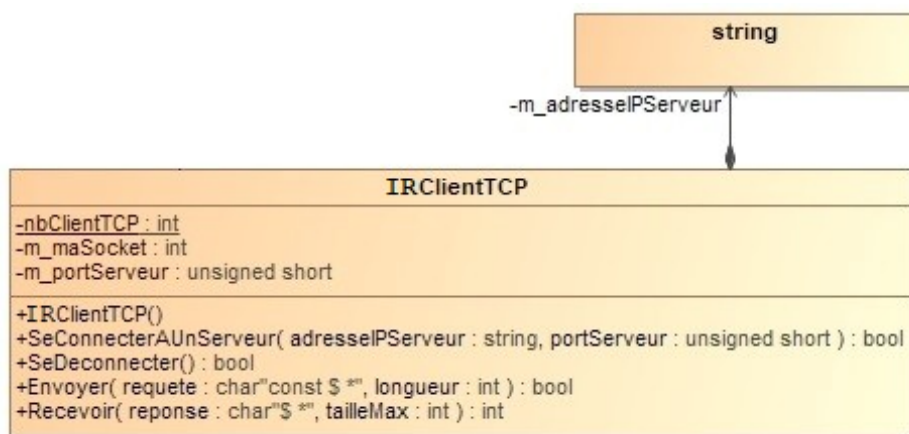
```
<uses-permission android:name="android.permission.INTERNET" />
```

Tester un premier déploiement sur tablette : « hello world ». Si un premier déploiement a été effectué avant d'ajouter les droits Internet, il faut désinstaller l'application sur la tablette.

Selon la version de la tablette utilisée, il faudra adapter les versions de SDK :



Intégrer la classe IRClientTCP, en se basant sur le module 3 : SS07 : **suivre à nouveau les vidéos**. Placer dans l'interface 2 boutons : 1 pour chacun des 2 identifiants CAN. Un TextView permettra d'afficher la trame CAN.



Dans MainActivity.java, ajouter 3 arguments à la méthode :

```
public native String stringFromJNI(String ip, String port, String idCAN);
```

Dans le code C++ (native-lib.cpp), les arguments deviennent :

```
extern "C" JNIEXPORT jstring JNICALL
Java_com_example_testclienttcp271123_MainActivity_stringFromJNI(
    JNIEnv* env,
    jobject /* this */, jstring ip, jstring port, jstring idCAN)
```

Pour convertir les variables JAVA dans le code C++, il faut utiliser :

```
std::string m_adresseIPServeur=(const char *)env->GetStringUTFChars(ip, NULL);
int m_portServeur=atoi(env->GetStringUTFChars(port, NULL));
```

En C++, envoyer l'identifiant CAN ("128" ou "0B6") au serveur, et afficher dans l'interface la réponse : suivre l'exemple de la séance SS07 du module 3.

Console d'AndroidStudio

Pour le débogage du code JAVA, utiliser :

```
Log.i("mon test log", "C3");
```

Pour le débogage du code C++, utilise :

```
syslog(LOG_CRIT, "information"); //nécessite #include <syslog.h>
```

Timer : envoi automatique

Qu'est ce qu'un timer en programmation ?

Ajouter un timer afin d'envoyer au serveur chaque seconde l'un puis l'autre des identifiants CAN.

```
ScheduledExecutorService timer = Executors.newSingleThreadScheduledExecutor();
timer.scheduleAtFixedRate(() -> {

}, 0, 1, TimeUnit.SECONDS);
```

Versionner le code complet.