

# Communications Numériques

## TP #2

### Représentation binaire d'une source d'informations

#### 1. Objectif du TP

L'objectif de ce deuxième TP est de pouvoir développer des schémas pour la visualisation d'une séquence binaire produite par plusieurs sources. Dans le cadre de ce TP, plusieurs sources sont testées notamment des sources binaires pures, des sources numériques, sources textuelles et sources analogiques. Pour chaque type de source, un schéma de visualisation doit être réalisé.

Les schémas à réaliser permettent de visualiser généralement une séquence binaire sur 8 bits. Cependant, dans certains cas, une visualisation sur un nombre variable de bits est envisagée pour apprécier la qualité de la représentation binaire.

#### 2. Visualisation d'une séquence binaire

Pour la visualisation d'une séquence binaire, deux schémas sont à réaliser :

- Schéma 1 : Séquence binaire déterministe.
- Schéma 2 : Séquence binaire aléatoire.

→ **Manipulation 1** : Réaliser le schéma de la simulation présenté dans Fig.2.1.

Pour cette manipulation, les blocs suivants sont utilisés pour la génération d'une séquence binaire déterministe :

- Variable : bit\_rate permet de fixer le nombre de bits émis par seconde. Dans ce cas, le débit est fixé à 1000 bits/s.
- Variable : k\_bits définit le nombre de bits à visualiser. Dans ce cas, k\_bits = 8 bits. Cependant, il est possible de changer la valeur selon les besoins de la simulation.
- Vector Source : permet de générer la séquence binaire déterministe en précisant les valeurs des bits dans un vecteur. Pour cette manipulation, une séquence de 16 bits doit être définie pour la représenter sur deux octets.
- Throttle : permet de limiter la consommation des ressources du processeur pour éviter une surchauffe en fixant Sample Rate. Dans ce cas, Sample Rate est fixé à k\_bits.

Les autres blocs ont été déjà utilisés dans le précédent TP.

Pour la visualisation de la séquence binaire sous forme d'amplitude, en double-cliquant sur le Bloc QT GUI Time Sink, appliquer les réglages suivants dans les onglets :

- General
  - Number of Points : k\_bits
  - Sample Rate : bit\_rate
- Config
  - Stem Plot -> Yes
  - Line 1 Color -> Blue
  - Line 1 Style -> Solid
  - Line 1 Marker -> Circle

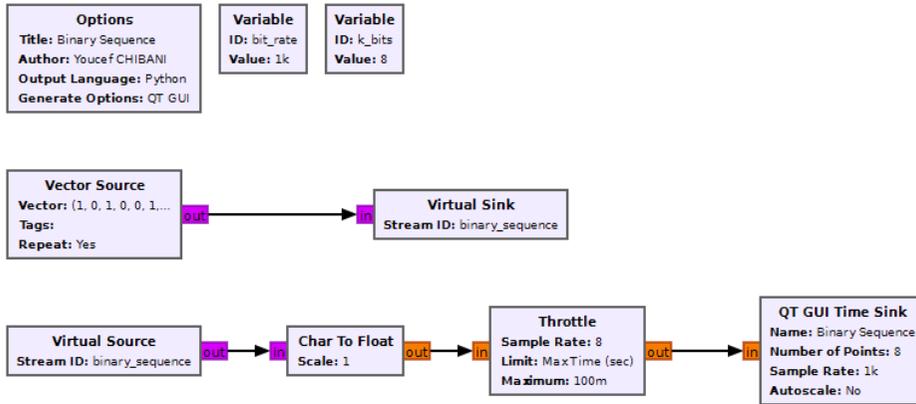


Fig. 2.1. Schéma de simulation d'une séquence binaire déterministe.

Question 2.1 : Interpréter le résultat obtenu.

→ **Manipulation 2** : Modifier le schéma de Fig. 2.1 en remplaçant le bloc Vector Source par Random Source avec les paramètres suivants :

- Minimum : 0
- Maximum : 2
- Num Samples : 1000000

Question 2.2 : Qu'observe-t-on ?

## 2. Visualisation d'une source numérique en binaire

Pour la visualisation binaire d'une source numérique, on se propose d'utiliser un signal numérique puis une image.

→ **Manipulation 3** : Réaliser le schéma de la simulation présenté dans Fig.2.2.

Pour cette manipulation, les blocs suivants sont utilisés pour la visualisation binaire d'une source numérique :

- File Source : permet de sélectionner un fichier numérique (Signal ou Image).
- Unpack K Bits : permet de décompacter un échantillon en un K bits. Dans notre cas, on le décompacte en utilisant la variable k\_bits fixée à 8 bits.

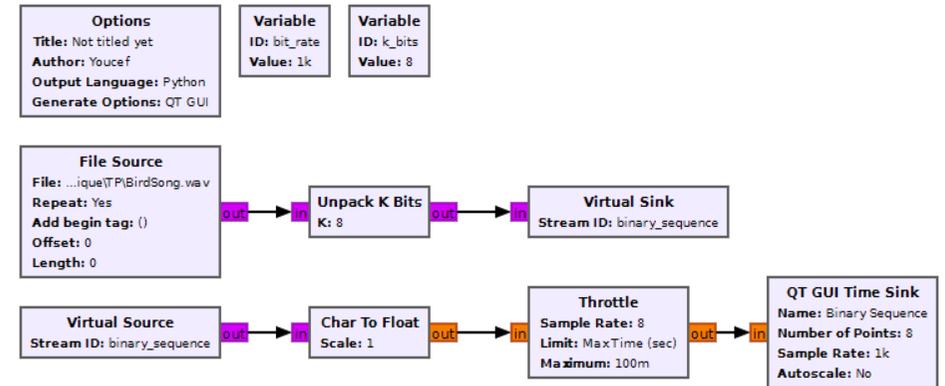


Fig. 2.2. Schéma de simulation d'une source numérique en binaire.

Question 2.3 : Qu'observe-t-on ?

## 3. Visualisation binaire d'une source composée de chiffres

Pour la visualisation binaire d'une source binaire composée de chiffres, deux schémas sont à réaliser :

- Schéma 1 : Séquence de chiffres déterministes.
- Schéma 2 : Séquence de chiffres aléatoires.

Pour la visualisation binaire d'une séquence de chiffres déterministes, on utilise le bloc Vector Source dans lequel on précise les chiffres à représenter en une séquence ID binaire.

→ **Manipulation 4** : Réaliser le schéma de la simulation présenté dans Fig.2.3.

Pour cette manipulation, les blocs suivants sont utilisés pour la visualisation binaire d'une source numérique :

- Variable : permet de définir la variable numbers dans un vecteur (0,1,2,3,4,5,6,7,8,9). Cette variable est utilisé dans le bloc Vector Source.
- Vector Source : permet de générer les chiffres en utilisant la variable numbers.

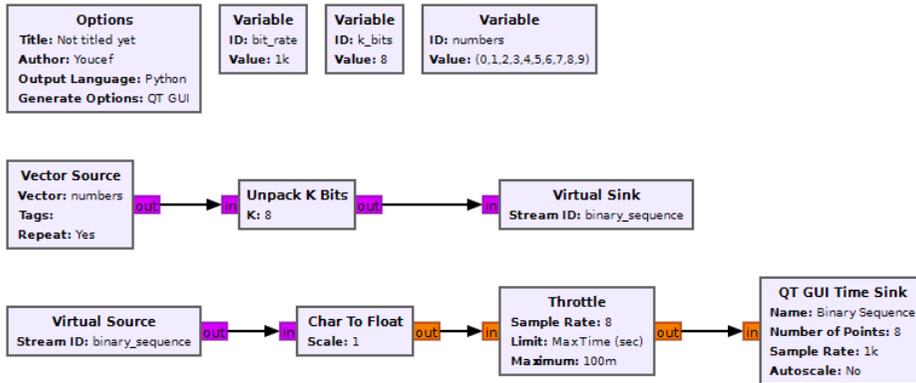


Fig. 2.3. Schéma de simulation d'une source en binaire composée de chiffres.

Question 2.4 : Qu'observe-t-on ?

→ **Manipulation 5** : Modifier le schéma de Fig. 2.3 en remplaçant le bloc Vector Source par Random Source avec les paramètres suivants :

- Minimum : 0
- Maximum : 256
- Num Samples : 1000000

Question 2.5 : Qu'observe-t-on ?

#### 4. Visualisation binaire d'un message textuel

Un message textuel peut être composé d'un mélange de chiffres, caractères de différentes langues ou emoji.

Pour la visualisation binaire d'un message textuel, on se propose de former un message en utilisant le protocole PDU (Protocol Data Unit) utilisé dans les réseaux informatiques.

PDU est un protocole qui permet de décrire la manière dont les informations sont emballées et transportées dans une chaîne de communication numérique notamment dans un réseau informatique.

Dans notre cas, PDU est utilisé pour représenter un message textuel pour l'envoyer dans une chaîne de communication numérique. Pour pouvoir visualiser le message en binaire, une transformation est nécessaire pour convertir la donnée au format PDU en format donnée en utilisant le bloc PDU to Stream.

→ **Manipulation 6** : Réaliser le schéma de la simulation présenté dans Fig.2.4.

Pour cette manipulation, les blocs suivants sont utilisés pour la visualisation binaire d'une source numérique :

- Variable : permet de définir le message textuel en déclarant la variable message dans une chaîne de caractères par exemple "0123456789".
- Periodic Message Source : permet de générer le message source défini par la variable message. Dans le paramètre Message PMT, on doit définir comment les caractères du message sont envoyés en utilisant la ligne de code en Python suivante :  
`pmt.cons(pmt.PMT_NIL, pmt.init_u8vector(len(message), [ord(c) for c in message]))`

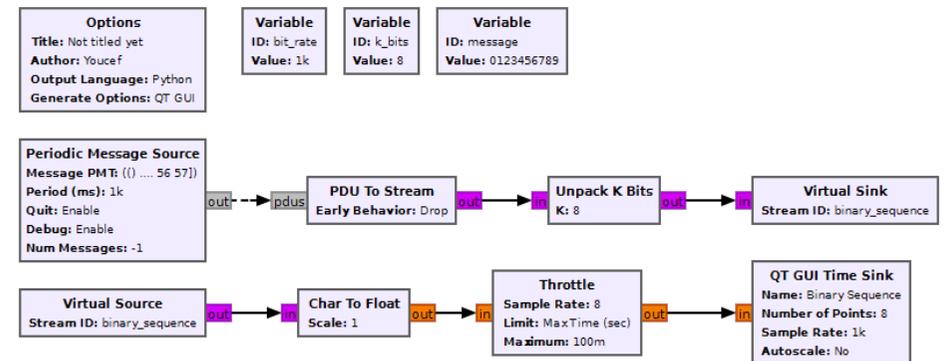


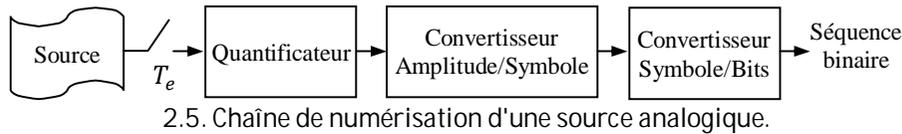
Fig. 2.4. Schéma de simulation d'une source textuelle en binaire.

Question 2.6 : Qu'observe-t-on ? Pourquoi la séquence binaire affichée dans cette manipulation n'est pas identique à la séquence binaire affichée par la manipulation 4 ?

Question 2.7 : Tester d'autres messages par exemple "Bonjour", "Hello", etc. Qu'observe-t-on ?

### 5. Visualisation binaire d'un signal analogique

Une chaîne de numérisation d'une source analogique est composée d'un échantillonneur de période d'échantillonnage  $T_e$ , d'un quantificateur pour convertir l'amplitude analogique en amplitude quantifiée, d'un convertisseur d'amplitude quantifiée en symbole et d'un convertisseur de symbole en bits. Fig. 2.5 illustre les différents blocs pour la numérisation d'une source analogique.

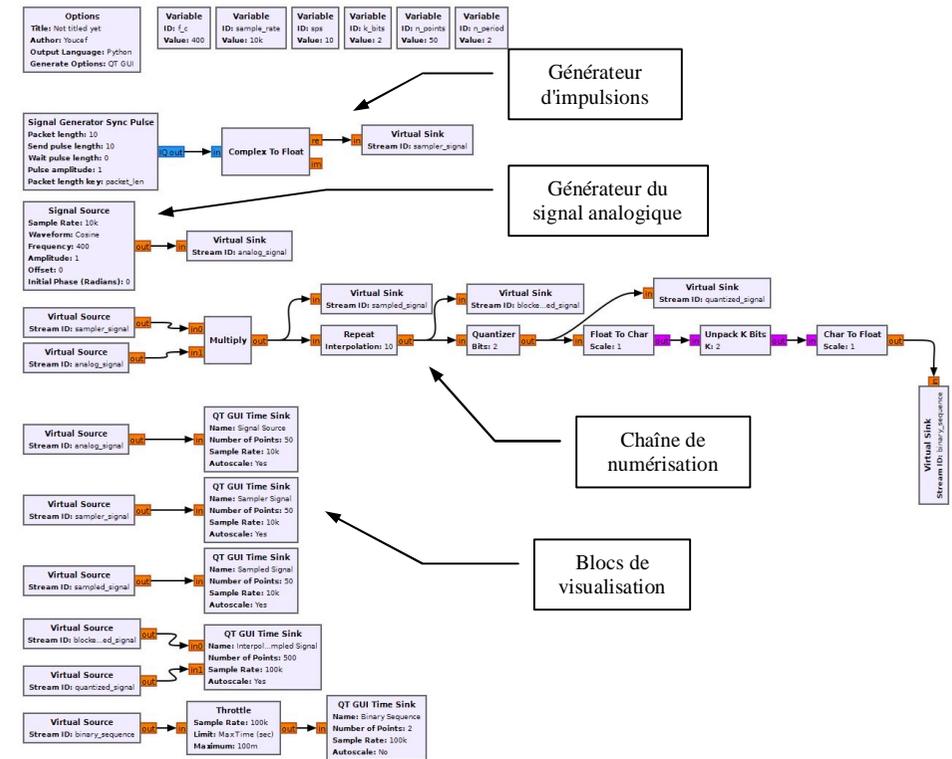


On se propose de réaliser le schéma de simulation en considérant un signal source de forme cosinus d'amplitude 1, réglé à une fréquence  $f_c = 400\text{Hz}$  et échantillonné à  $10000\text{Hz}$ .

→ **Manipulation 6** : Réaliser le schéma de la simulation présenté dans Fig.2.6.

Pour cette manipulation, les principaux blocs suivants sont utilisés pour la visualisation binaire d'une source analogique :

- Variable : 6 blocs de variable sont définis pour fixer la fréquence du signal ( $f_c$ ), la fréquence d'échantillonnage (sample\_rate), nombre d'échantillons par symbole (sps), nombre de bits pour représenter chaque symbole (k\_bits), nombre de points pour la visualisation du signal (n\_points) et le nombre de période pour la visualisation (n\_period).
- Signal Generator Sync Pulse : permet de générer les impulsions de forme Dirac. Les paramètres Packet length et Send Pulse doivent être réglés à sps.
- Repeat : permet de jouer le rôle d'un bloqueur en répétant la valeur de l'amplitude de l'échantillon par le bloc d'échantillonnage sps fois.
- Quantizer : permet de quantifier l'amplitude du signal en fixant le nombre bits du paramètre Bits à l'aide de la variable k\_bits.



2.6. Simulation d'une chaîne de numérisation d'une source analogique.

Question 2.8 : Quel est le rôle de la variable sps ?

Question 2.9 : Que se passe-t-il lorsqu'on change le nombre de bits de quantification ?