Transmission de données agricoles

Sujet n°23 — Corrigé

Question 1 *Décoder température et humidité*

```
    def decoder_temperature(self):
        self._temperature = (int(self._trame[16:28], 2) - 900) / 10

    def decoder_humidite(self):
        bloc = self._trame[28:36]
        if bloc == "10100000":          # cas particulier : 100 %
            self._humidite = 100
        else:                           # BCD : un chiffre décimal par quartet
            self._humidite = int(bloc[0:4], 2) * 10 + int(bloc[4:8], 2)

t = Transmission("00101010110010000100100011000110001011101")
assert t.get_temperature() == 26.4
assert t.get_humidite() == 62
```

Question 2 *Contrôle de parité*

```
    def est_valide(self):
        trame = self._trame
        blocs = [trame[0:8], trame[8:16], trame[16:28], trame[28:36]]
        for i in range(4):
            if str(blocs[i].count("1") % 2) != trame[36 + i]:
                return False
        return True
```

Question 3 *Analyse des trames reçues*

Le fichier contient 314 trames ; avec le seul contrôle de parité, 310 sont jugées valides et le graphique présente des valeurs aberrantes. Le fichier contient en effet deux trames de 41 bits et une de 38 bits. Les deux trames de 41 bits passent le contrôle par hasard (les bits lus en `trame[36:40]` ne sont pas les bits de contrôle) : l'une donne une humidité de 114 %, dont le quartet `1011` n'est pas un chiffre BCD. Une trame correcte en parité indique enfin −28,6 °C, une erreur de mesure d'après l'énoncé. Une trame trop courte pourrait aussi provoquer une erreur de conversion.

Question 4 *Un serveur robuste*

On complète `est_valide` : longueur et caractères d'abord, parité ensuite, puis valeurs physiquement possibles.

```
    def est_valide(self):
        trame = self._trame
        if len(trame) != 40 or trame.count("0") + trame.count("1") != 40:
            return False
        blocs = [trame[0:8], trame[8:16], trame[16:28], trame[28:36]]
        for i in range(4):
            if str(blocs[i].count("1") % 2) != trame[36 + i]:
                return False
        quartets = (int(trame[28:32], 2), int(trame[32:36], 2))
        if trame[28:36] != "10100000" and (quartets[0] > 9 or quartets[1] > 9):
            return False
        return self._temperature >= -10
```

Il reste 307 trames valides ; la courbe varie de 22,7 à 34,2 °C, sans pic aberrant.