

reprend ce code

```
int capteurIndex = str.indexOf("Td") + 2;
int temperatureIndex = str.indexOf("Y") + 1;

QString capteur = str.mid(capteurIndex, 2);
QString temperature = str.mid(temperatureIndex, 6);

qDebug() << "Capteur : " << capteur;
qDebug() << "Température : " << temperature;
ui->reponseServ->setText("Réponse du serveur : Température du capteur " +
capteur + ": " + temperature + "°C");
```

et fais en sorte que ça ne lise uniquement les données à partir du + ou -

ChatGPT

Pour extraire uniquement les données de température à partir du signe "+" ou "-" sans inclure le préfixe "Td" dans la chaîne, vous pouvez ajuster le code de la manière suivante :

cpp

Copy code

```
void ClientQt::onSocketReadyRead() { QByteArray data = socket-
>readAll(); QString str(data); qDebug() << "Réponse brute du serveur : "
<< str; ui->reponseServ->setText("Réponse du serveur : " + str); if
(str.startsWith("Td")) { // La réponse est de type "TdxX,YXX.XX" int
temperatureIndex = str.indexOf("+"); if (temperatureIndex == -1)
{ temperatureIndex = str.indexOf("-"); } if (temperatureIndex != -1)
{ QString temperature = str.mid(temperatureIndex); qDebug() <<
"Température : " << temperature; ui->reponseServ->setText("Réponse du
serveur : Température : " + temperature + "°C"); } } // Ajoutez des
conditions similaires pour les autres formats de réponse (Tf, Hr) si
nécessaire. }
```

Dans cet exemple, nous cherchons d'abord la position du signe "+" et, s'il n'est pas trouvé, la position du signe "-". Ensuite, nous extrayons la température à partir de cette position jusqu'à la fin de la chaîne. Cela devrait vous donner uniquement les données de température sans inclure le préfixe "Td".



```
j'ai donc ce code pour td en celcius : if (str.startsWith("Td")) {
// La réponse est de type "TdxX,YXX.XX"
```

```

int capteurIndex = str.indexOf("Td") + 2;
int temperatureIndex = str.indexOf("Y") + 5;

QString capteur = str.mid(capteurIndex, 2);
QString temperature = str.mid(temperatureIndex, 6);

qDebug() << "Capteur : " << capteur;
qDebug() << "Température : " << temperature;
ui->reponseServ->setText("Réponse du serveur : Température du capteur " +
capteur + ": " + temperature + "°C");
}

adapte le aux autres conditions qui sont comme ceci initialement : else if
(str.startsWith("Tf")) {
    // La réponse est de type "Tfxx,YXX.XX"
    // Extrait les données de température en degrés Fahrenheit
    QRegularExpression rx("Tf(\\d+),(Y[+-]\\d+\\.\\d+)");
    QRegularExpressionMatch match = rx.match(str);
    if (match.hasMatch()) {
        QString capteur = match.captured(1);
        QString temperature = match.captured(2);
        ui->reponseServ->setText("Réponse du serveur : Température du capteur "
+ capteur + ": " + temperature + "°F");
    }
} else if (str.startsWith("Hr")) {
    // La réponse est de type "Hrxx,XX.XX"
    // Extrait les données d'hygrométrie
    QRegularExpression rx("Hr(\\d+),(\\d+\\.\\d+)");
    QRegularExpressionMatch match = rx.match(str);
    if (match.hasMatch()) {
        QString capteur = match.captured(1);
        QString hygrometrie = match.captured(2);
        ui->reponseServ->setText("Réponse du serveur : Hygrométrie du capteur "
+ capteur + ": " + hygrometrie + "%");
    }
}
}

```

ChatGPT

Vous pouvez adapter le code pour les autres conditions en suivant le modèle que vous avez utilisé pour "Td". Voici comment vous pouvez le faire pour "Tf" et "Hr" :

cpp

Copy code

```

if (str.startsWith("Tf")) { // La réponse est de type "Tfxx,YXX.XX" int
capteurIndex = str.indexOf("Tf") + 2; int temperatureIndex =

```

```

str.indexOf("Y") + 5; QString capteur = str.mid(capteurIndex, 2);
QString temperature = str.mid(temperatureIndex, 6); qDebug() <<
"Capteur : " << capteur; qDebug() << "Température : " << temperature;
ui->reponseServ->setText("Réponse du serveur : Température du capteur "
+ capteur + ": " + temperature + "°F"); } else if
(str.startsWith("Hr")) { // La réponse est de type "Hrxx,XX.XX" int
capteurIndex = str.indexOf("Hr") + 2; int hygrometrieIndex =
str.indexOf("Y") + 5; QString capteur = str.mid(capteurIndex, 2);
QString hygrometrie = str.mid(hygrometrieIndex, 6); qDebug() <<
"Capteur : " << capteur; qDebug() << "Hygrométrie : " << hygrometrie;
ui->reponseServ->setText("Réponse du serveur : Hygrométrie du capteur "
+ capteur + ": " + hygrometrie + "%"); }

```

Ce code suit la même logique que celui pour "Td" en cherchant l'indice de départ pour le numéro du capteur et pour la température ou l'hygrométrie, puis en extrayant les données correspondantes. Assurez-vous que les indices utilisés correspondent à la structure réelle de vos réponses du serveur pour "Tf" et "Hr".