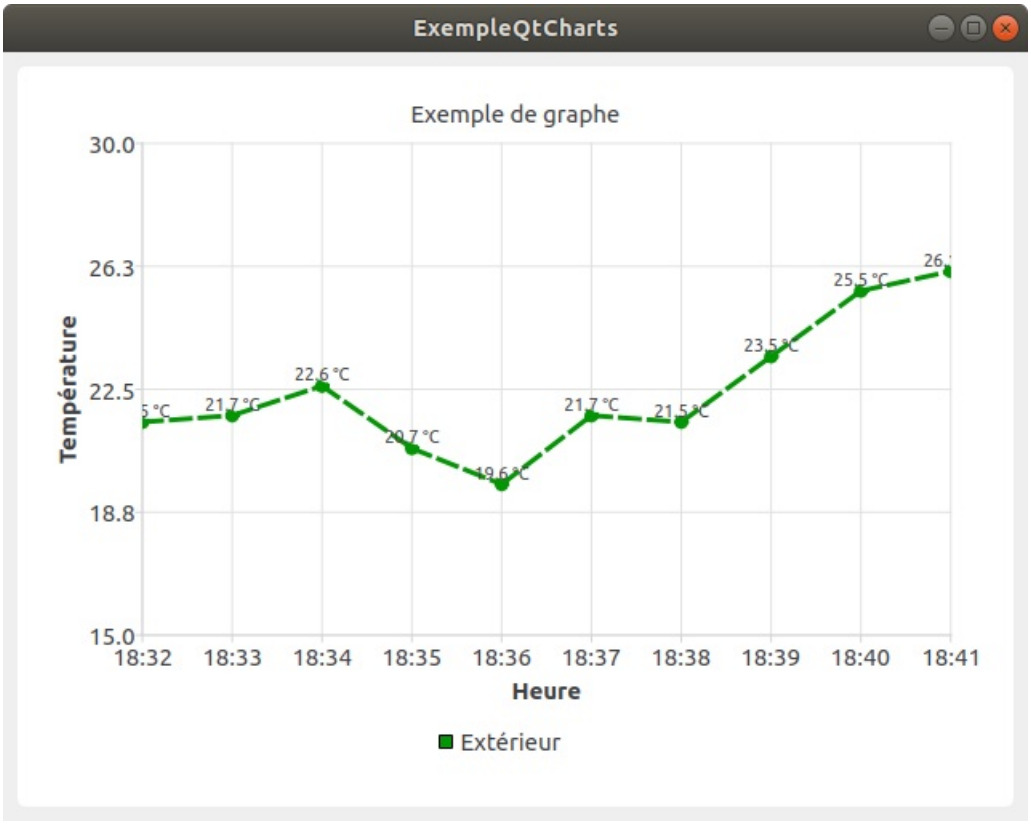
Qt : Des graphiques avec Qt Charts

> Version PDF de ce document : activite-qt-charts.pdf

Depuis la version 5.7, Qt fournit le module Qt Charts pour dessiner des graphiques.

On va présenter un exemple pour obtenir ceci :



API Qt Charts

Liens pour Qt Charts :

- Module Qt Charts
- Présentation

Pour utiliser le module `Qt Charts`, il faut l'activer, ainsi que le module `widgets`, dans le fichier de projet `.pro` :

```
...
QT += widgets charts
...
```

Remarque : Il faut utiliser `QApplication` (et non `QGuiApplication`) dans le fichier `main.cpp`.

On a créé un projet Qt widgets. Au final, le fichier `.pro` sera le suivant :

```
QT       += core gui charts

greaterThan(QT_MAJOR_VERSION, 4): QT += widgets

TARGET = ExempleQtCharts
TEMPLATE = app

SOURCES += \
        main.cpp \
        ihmgraphique.cpp

HEADERS += \
        ihmgraphique.h
```

Liens pour Qt Charts :

- Module Qt Charts
- Présentation
- Classes C++
- Chart
- ChartView
- LineSeries
- ValueAxis

L'affichage est basé sur un `ChartView`. On ajoute un `LineSeries` pour l'affichage de la courbe ainsi que 2 axes basés sur `ValueAxis`. Si l'on souhaite afficher l'horodatage, on utilisera un `DateTimeAxis` pour l'axe X.

La courbe `LineSeries` peut se personnaliser pour afficher les points de mesure (`setPointsVisible()`) et leurs valeurs (`setPointLabelsFormat()` et `setPointLabelsVisible()`). On lui donnera aussi un nom (`name`) pour qu'il s'affiche dans la légende du graphique. Le tracé se personnalise avec un `QPen`, par exemple avec `setStyle()`, on peut choisir le type de ligne, par exemple `Qt::DashLine` pour un affichage en pointillés.

Déclaration de la classe :

```
#ifndef IHMGRAPHIQUE_H
#define IHMGRAPHIQUE_H

#include <QtWidgets>
#include <QtCharts>

class IhmGraphique : public QMainWindow
{
    Q_OBJECT

private:
    QChartView *graphique; // un widget pour afficher un graphe
    QChart *graphe; // la représentation d'un graphe
    QLineSeries *courbe; // les données

public:
    IhmGraphique(QWidget *parent = 0);
    ~IhmGraphique();
};
```

```
#endif // IHMGRAPHIQUE_H
```

Définition de la classe :

```
#include "ihmgraphique.h"

IhmGraphique::IhmGraphique(QWidget *parent)
    : QMainWindow(parent)
{
    // Les données
    courbe = new QLineSeries();
    // Technique n°1
    // exemple avec un QValueAxis
    /*courbe->append(1, 21.5);
    courbe->append(2, 21.7);
    courbe->append(3, 22.6);
    courbe->append(4, 20.7);
    courbe->append(5, 19.6);
    courbe->append(6, 21.7);
    courbe->append(7, 21.5);
    courbe->append(8, 23.5);
    courbe->append(9, 25.5);
    courbe->append(10, 26.1);*/

    // exemple avec un QDateTimeAxis
    QDateTime debut = QDateTime::currentDateTime();
    QDateTime fin;
    courbe->append(debut.toMSecsSinceEpoch(), 21.5);
    fin = QDateTime::currentDateTime().addSecs(1*60);
    courbe->append(fin.toMSecsSinceEpoch(), 21.7);
    fin = QDateTime::currentDateTime().addSecs(2*60);
    courbe->append(fin.toMSecsSinceEpoch(), 22.6);
    fin = QDateTime::currentDateTime().addSecs(3*60);
    courbe->append(fin.toMSecsSinceEpoch(), 20.7);
    fin = QDateTime::currentDateTime().addSecs(4*60);
    courbe->append(fin.toMSecsSinceEpoch(), 19.6);
    fin = QDateTime::currentDateTime().addSecs(5*60);
    courbe->append(fin.toMSecsSinceEpoch(), 21.7);
    fin = QDateTime::currentDateTime().addSecs(6*60);
    courbe->append(fin.toMSecsSinceEpoch(), 21.5);
    fin = QDateTime::currentDateTime().addSecs(7*60);
    courbe->append(fin.toMSecsSinceEpoch(), 23.5);
    fin = QDateTime::currentDateTime().addSecs(8*60);
    courbe->append(fin.toMSecsSinceEpoch(), 25.5);
    fin = QDateTime::currentDateTime().addSecs(9*60);
    courbe->append(fin.toMSecsSinceEpoch(), 26.1);

    // Technique n°2
    // exemple avec un QValueAxis
    //*courbe << QPointF(1, 21.5) << QPointF(2, 21.7) << QPointF(3, 22.6) << QPointF(4, 20.7) <<
QPointF(5, 19.6) << QPointF(6, 21.7) << QPointF(7, 21.5) << QPointF(8, 23.5)<< QPointF(9, 25.5)<<
 QPointF(10, 26.1);
    courbe->setName(QString::fromUtf8("Extérieur"));
    QPen pen(0x059605);
    pen.setWidth(3);
    pen.setStyle(Qt::DashLine);
    courbe->setPen(pen);
```

```
    // Le graphe
    graphe = new QChart();
    graphe->addSeries(courbe);
    graphe->setTitle("Exemple de graphe");
    // Légende
    //graphe->legend()->hide();
    graphe->legend()->setAlignment(Qt::AlignBottom);

    // Les axes
    // Par défaut
    //graphe->createDefaultAxes();
    //graphe->axes(Qt::Horizontal).first()->setRange(0, 20);
    //graphe->axes(Qt::Vertical).first()->setRange(0, 10);

    // exemple avec un QValueAxis
    /*QValueAxis *axeX = new QValueAxis;
    axeX->setTickCount(10);
    axeX->setLabelFormat("%i");
    graphe->addAxis(axeX, Qt::AlignBottom);
    courbe->attachAxis(axeX);*/

    // exemple avec un QDateTimeAxis
    QDateTimeAxis *axeX = new QDateTimeAxis;
    axeX->setTickCount(10);
    axeX->setFormat("hh:mm");
    axeX->setTitleText("Heure");
    // Avec QDateTimeAxis, il faut définir min et max
    axeX->setMin(debut);
    axeX->setMax(fin);
    graphe->addAxis(axeX, Qt::AlignBottom);
    courbe->attachAxis(axeX);

    QValueAxis *axeY = new QValueAxis;
    axeY->setRange(15, 30);
    axeY->setLabelFormat("%.1f");
    axeY->setTitleText(QString::fromUtf8("Température"));
    graphe->addAxis(axeY, Qt::AlignLeft);
    courbe->setPointsVisible(true);
    courbe->setPointLabelsFormat("@yPoint °C");
    courbe->setPointLabelsVisible(true);
    courbe->attachAxis(axeY);

    // Le widget
    graphique = new QChartView(graphe);
    graphique->setRenderHint(QPainter::Antialiasing);

    setCentralWidget(graphique);
    resize(640, 480);
}

IhmGraphique::~IhmGraphique()
{
}
```

Lien : ExempleQtCharts.zip

http://tvaira.free.fr/