

ВИМІРЮВАЛЬНІ СИСТЕМИ

УДК 536.532

ЗАСОБИ ВІДОБРАЖЕННЯ ВИМІРЮВАЛЬНОЇ ІНФОРМАЦІЇ У СЕРЕДОВИЩІ ПРОГРАМУВАННЯ QT CREATOR

© Андрій Озгович, Олена Тищенко, Тарас Домінюк, 2014

Національний університет “Львівська політехніка”, кафедра інформаційно-вимірювальних технологій,
вул. С. Бандери, 12, Львів-79013, Україна

Пропонується загальний огляд засобів, необхідних для створення інтерфейсу користувача віртуального вимірювального приладу, описано створення засобів відображення вимірювальної інформації як власних елементів управління в інтегрованому середовищі розроблення Qt Creator. Подано опис розроблених елементів відображення інформації та наведено фрагменти програм.

Предлагается общий обзор средств, которые необходимы для создания интерфейса виртуального измерительного прибора, описано создание средств отображения измерительной информации как собственных элементов управления в интегрированной среде разработки Qt Creator. Представлено описание разработанных элементов отображения информации и приведены фрагменты программ.

A general overview of tools that are needed to create the user interface of the virtual meter, described the creation of display measurement information as their own controls, an integrated development environment Qt Creator. The description of the developed elements display are pieces of information and applications.

Вступ. Сьогодні практично неможливо уявити собі програму, що не має інтерфейсу користувача. У статті пропонується використання бібліотек Qt, що має низку переваг. Працівник фірми Nokia та засновник KDE Маттіас Еттріх доволі влучно назвав ці переваги: «Qt робить платформонезалежне програмування реальністю, не обмежуючи ваш вибір і творчі можливості. Qt пропонує користувачам те, що вони хочуть: швидкі програми, що виглядають та працюють належно. Qt пропонує розробникам програм те, чого вони бажають: середовище, що дозволяє писати менше коду, створюючи при цьому більше. Завдяки цьому програмування стає цікавішим, і при цьому неважливо, є воно комерційним чи проектом з відкритим вихідним кодом (Open Source)».

Інтегроване середовище розроблення Qt Creator – це набір інструментів, що об’єднані в одному додатку.

Практично будь-яка програма має графічний інтерфейс користувача. Віджети (widgets) – це «будівельний матеріал» для його створення. Віджет – це не тільки область, що відображається на екрані, а компонент, здатний виконувати різні дії, наприклад реагувати на сигнали чи події, що надходять, чи відправляти сигнали іншим віджетам. Qt надає повний арсенал віджетів: від кнопок меню до діалогових вікон,

які необхідні для створення професійних програм. Проте виникає завдання створення власних віджетів, оскільки стандартного набору недостатньо для створення віртуальних вимірювальних приладів

Актуальність дослідження. Аналіз сучасного стану та тенденцій розвитку вимірювальної техніки, засобів вимірювань свідчить про те, що поряд з розробленням та удосконаленням традиційних вимірювальних приладів все більшого значення набуває новий напрям, а саме розроблення віртуальних вимірювальних приладів.

Основним елементом для людини, яка використовує віртуальні вимірювальні прилади, є інтерфейс користувача. Він, як правило, імітує передню панель традиційного приладу, що містить елементи керування, засоби введення та відображення інформації, функціональні графічні зображення (схеми, фото тощо).

Проаналізувавши всі наявні та доступні засоби для Qt Creator з відкритим вихідним кодом, можна зробити висновок про малу кількість та невпорядкованість засобів відображення вимірювальної інформації, які можна використати для розроблення віртуальних вимірювальних приладів. Саме тому постає завдання створення нових, структурованих, доступних, віджетів-індикаторів з широким спектром властивостей та універсальністю.

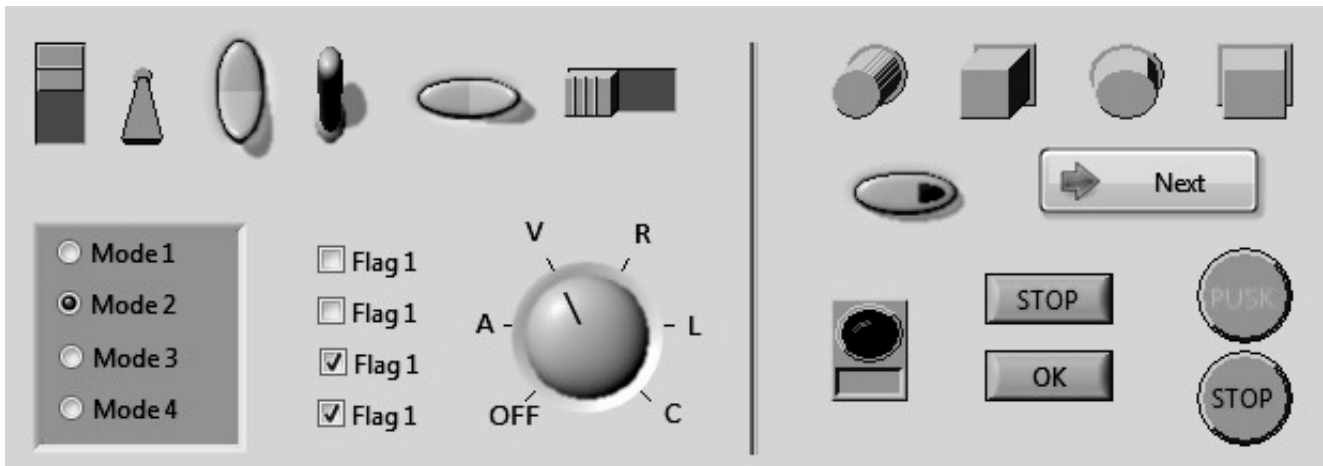


Рис. 1. Засоби керування режимами роботи віртуальних приладів

Мета роботи. Узагальнення відомостей про засоби відображення вимірювальної інформації та засоби керування. Створення універсальних аналогових та цифрових індикаторів, на основі яких можна будувати різноманітні віртуальні вимірювальні прилади та системи, розширення кількості доступних засобів відображення вимірювальної інформації для Qt Creator з відкритим вихідним кодом.

Виклад основного матеріалу. Панель користувача віртуального вимірювального приладу, як і реального приладу, складається із сукупності елементів керування та засобів відображення вимірювальної інформації (індикаторів) (рис. 1). Елементи керування імітують типові засоби управління, які є у будь-якого вимірювального приладу, наприклад, кнопки та перемикачі. Елементи керування дозволяють користувачу вводити дані. Індикатори відображають вихідні дані, що є результатом виконання програми. Тобто:

Елементи керування = дані, що вводить користувач = джерело даних.

Індикатори = дані, що виводяться користувачу = приймачі даних.



Рис. 2. Елементи введення числової інформації

Для створення віртуальних вимірювальних приладів найчастіше використовують такі елементи керування:

Перемикач – електронний компонент, який може розмикати електричне коло, вимикаючи струм або перемикаючи його з одного провідника на інший. Найпоширеніші типи перемикачів: перемикач клавішний, кнопковий вимикач, перемикач галетний, перемикач у корпусі DIP. Контакти можуть бути або нормально розімкнені, або нормально замкнені.

Перемикач – один з найважливіших елементів віртуальних вимірювальних приладів, що найчастіше використовується. Він являє собою компонент віртуального пристрою, який може перебувати в одному із двох станів: увімкнено (on) і вимкнено (off). Перемикач повинен надавати користувачу, щонайменше, вибір однієї з двох альтернатив. Перевага перемикачів полягає в тому, що всі опції користувачу видно відразу.

Кнопка – кнопка належить до засобів введення інформації в людино-машинних інтерфейсах шляхом формування електричного сигналу в різних пристроях методом замикання чи/та розмикання двох або більше електричних контактів.

Два (три, у випадку перемикання) контакти, що комутують сигнальні лінії в процесі натиснення кнопки, називають контактною групою. Кнопка може мати як одну, так і декілька контактних груп – нормально розімкнених, нормально замкнених чи перемикальних – у довільній їх комбінації.

Кнопка, залежно від наявності прикладеного до неї впливу, має два положення – «натиснуто» і «відпущено». Залежно від стану, у якому кнопка залишиться після зняття впливу на неї, кнопки поділяються на кнопки фіксованого і нефіксованого типу.

Кнопку можна побачити на будь-якому вимірювальному приладі, оскільки це елемент керування, що використовується дуже часто, як правило, для

виконання певної операції після натискання на неї. Для створення віртуальних вимірювальних приладів використовують такі варіанти кнопок, що натискаються:

- Звичайна кнопка – кнопка, що після відпускання завжди повертається у вихідне положення.
- Вимикач – кнопка, що може перебувати у двох станах: натисненому і ненабитеному, які відповідають положенню «увімкнено» чи «вимкнено».

Більшість вимірювальних приладів надають цілий ряд настроювань, що дають змогу змінити поведінку приладу. Для цього може бути корисним елемент **прапорець**, який дозволяє користувачу вибрати відразу декілька опцій. Прапорець складається з маленького прямокутника і може містити інформаційний текст чи картинку. Після натискання на прапорець у прямокутнику з'являється відмітка. Цей елемент керування встановлюється в положення «увімкнено» чи «вимкнено», і є, за логікою дій, кнопкою-вимикачем. Але, на відміну від інших, прапорець може мати ще і третій стан – невизначений.

Повзунок – компонент, призначений для введення чисел у вказаному діапазоні. Його аналогом в електронній техніці є змінний резистор.

Повзунок дає змогу комфортно здійснювати настроювання деяких параметрів віртуальних вимірювальних приладів, наприклад, гучність акустичного сигналу тощо. Цей елемент керування має позначки, які дають користувачу візуально чіткіше уявлення про крок та місце знаходження.

Поле числового введення даних є основою для введення та редагування даних – тексту та чисел – користувачем.

Засоби відображення вимірювальної інформації, які використовуються під час створення віртуальних вимірювальних приладів.

Одиничний індикатор складається з одного елемента відображення і призначений переважно для подання інформації у вигляді точки або іншої геометричної фігури.

Стрілковий індикатор – це засіб відображення інформації для забезпечення оператора відомостями про хід і спрямованість змін, а також кількісними характеристиками параметрів, що контролюються, у зоні допустимих значень.

Основними перевагами стрілкових індикаторів є простота, зручність і легкість в експлуатації. Незважаючи на певні недоліки – відображення тільки одного параметра, порівняно велику площину шкали – стрілкові індикатори широко використовуються, оскільки оператори дуже легко пристосовуються до них.

Ефективність сприйняття інформації з таких індикаторів залежить від характеристик окремих їх елементів: шкали, стрілки, оцифровки, розмітки.

Цифровий індикатор – пристрій для відображення числового значення у цифровому вигляді. Має фіксований набір елементів відображення (сегментів), які розташовані або довільно, або згруповані по декілька цифр.

За допомогою цього засобу відображення інформації на віртуальному вимірювальному приладі відображаються цілі числа. Цифровий індикатор можна увімкнути в режимі відображення двійкової, вісімкової, десяткової та шістнадцяткової систем числення, що неможливо для аналогового індикатора.

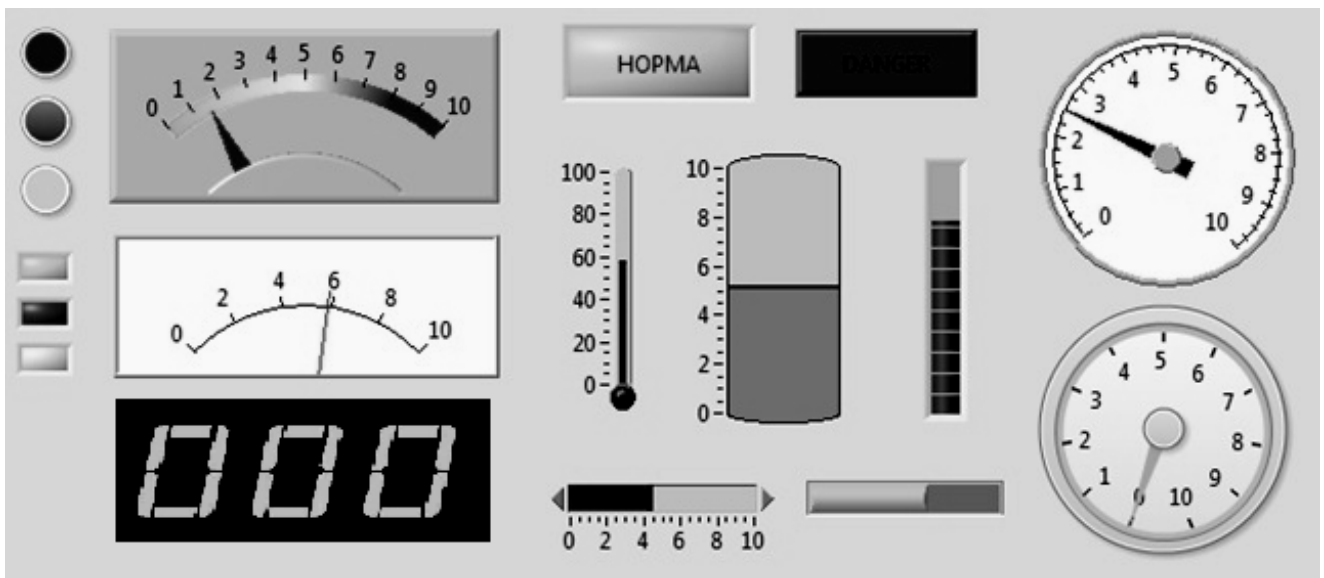


Рис. 3. Засоби відображення вимірювальної інформації

Шкальний індикатор – вид знаковитезувального індикатора. Призначений для відображення інформації у вигляді рівнів чи значень величини.

Дає змогу приблизно оцінити рівень сигналу за кількістю чи положенням елементів індикації на дискретній шкалі. Виконується у вигляді набору одиничних індикаторів (елементів шкали), розташованих вздовж прямого чи криволінійного відрізка. Індикація рівня сигналу на такому пристрої відображається увімкненням одного чи впорядкованих разом елементів. У першому випадку індикатор показує рівень сигналу положенням одного елемента; що далі від початку шкали, то більший рівень (шкала без заповнення). У другому випадку рівень сигналу відображається увімкненням всіх елементів від початку шкали до діючого значення (шкала із заповненням). Щоб більше елементів має шкала, то точніше можна оцінити рівень сигналу.

Шкальні індикатори застосовуються:

- для позначення амплітуди звукового сигналу в звуковідтворювальних пристроях;
- як шкала налаштування у радіоприймачах;
- для побудови графіків зміни величини тиску чи температури в погодних станціях;
- в цифрових мультиметрах для оперативної зміни рівня сигналу в аналоговому вигляді;
- індикатор заряду акумуляторної батареї в мобільних пристроях.

Елементи відображення не беруть активної участі у діях користувача і використовуються для інформування про те, що відбувається. Це може бути як текстова (надписи), так і графічна інформація (картинки, графіка тощо).

У сучасному програмуванні програми складаються з готових програмних блоків (як електронні прилади з мікросхем). Процес програмування полягає в налаштуванні параметрів блока, передаванні даних у блок та отриманні результатів роботи цього блока. Такі блоки називаються класами компонентів. Компоненти, які відображаються на екрані, називаються віджетами. За допомогою віджетів програміст розробляє інтерфейс користувача, перетягуючи їх на форму.

У базовій версії Qt Creator вбудовані віджети універсальні для всіх віджет-програм, але Qt є відкритою системою, яка дає змогу додавати будь-які елементи, які можна створити самостійно чи завантажити із сайту. Для кожної галузі науки і техніки можна створити спеціалізовані класи компонентів та віджетів, зокрема і для вимірювальної техніки.

Основою графічних засобів, що використовуються в Qt для створення віджетів, є клас QPainter. Цей клас може використовуватися для рисування геометричних фігур (точок, ліній, прямокутників, еліпсів, дуг, сегментів і секторів кола, багатокутників і кривих Без'є), а також піксельних карт, зображень і тексту. Крім того, QPainter підтримує такі функції, як згладжування з'єднань ліній (antialiasing) під час рисування фігур і букв у тексті, альфа-змішування (alpha blending), плавний перехід кольорів (gradient filling). QPainter також підтримує лінійні перетворення, такі як зміщення, повертання, обрізання і масштабування. Змінюючи реалізації функції QWidget :: paintEvent (), можна створювати власні віджети і здійснювати повний контроль над їх зовнішнім виглядом.

Трьома основними параметрами налаштування QPainter є перо, пензель і шрифт. Перо використовується для відображення прямих ліній і контурів фігур. Воно має колір, товщину, стиль лінії, стиль закінчення лінії та стиль з'єднання ліній. Пензель – це шаблон, який використовується для заповнення геометричних фігур. Він зазвичай має колір і стиль, але може також містити текстуру (піксельну карту, повторювану нескінченно) або кольорний градієнт. Шрифт має багато атрибутів, зокрема назву і розмір. Ці налаштування можна в будь-який час модифікувати за допомогою функцій setPen (), setBrush () і setFont ().

Клас QPainterPath може визначати довільні фігури векторної графіки, з'єднуючи один з одним основні графічні елементи: прямі лінії, еліпси, багатокутники, дуги, криві Без'є та інші. Такі ланцюжки є завершеними елементарними малюнками в тому сенсі, що будь-яка фігура або будь-яка комбінація фігур може бути подана у вигляді деякої послідовності графічних елементів QPainter.

У сучасних програмах градієнтні заповнення графічних фігур є популярною альтернативою однорідному заповненню. Кольорові градієнти ґрунтуються на інтерполяції кольорів, що забезпечує згладжені переходи між двома або більше кольорами. Вони часто застосовуються для отримання ефекту тримірності зображення. Qt підтримує три типи кольорних градієнтів: лінійний, конічний і радіальний. Лінійні градієнти визначаються двома контрольними точками і послідовністю «колірних відміток» на лінії, що з'єднує ці точки. Радіальні градієнти визначаються центральною точкою (xс, ус), радіусом r і точкою фокусу (xf, yf), яка доповнює колірні мітки. Центральна точка і радіус визначають коло. Зміна кольору поширюється у всі боки від точки фокусу, яка може збігатися з

центральною точкою або ні. Конічні градієнти визначаються центральною точкою (x_c , y_c) і кутом α . Зміна кольору поширюється навколо центральної точки подібно до переміщення секундної стрілки годинника.

У будь-який момент можливо зберегти в стеку стан QPainter за допомогою функції `save()`, і відновити його пізніше, викликаючи функцію `restore()`.

У координатній системі QPainter, що використовується за замовчуванням, точка (0, 0) розміщена в лівому верхньому кутку області відображення; значення координати x збільшується у разі переміщення вправо, а значення координати y збільшується у разі переміщення вниз. Область відображення є довільним прямокутником, заданим у фізичних координатах. Вікно є таким самим прямокутником, але в логічних координатах. Під час рисування задають координати точок у логічній системі координат. Ці координати, за допомогою лінійного алгебраїчного перетворення, переводяться у фізичну систему координат на основі налаштувань зв'язку «вікно – область відображення». Механізм «вікно – область відображення» зручно застосовувати для створення програмного коду, який не залежить від розміру або роздільної здатності пристрою виведення зображення.

Універсальна матриця перетворення використовується як доповнення до перетворення «вікно – область відображення». Вона дає змогу переміщувати початок координат, змінювати масштаб, повертати і обрізати графічні елементи. Зручно здійснювати перетворення, застосовуючи відповідні функції класу QPainter: `translateO`, `scale()`, `rotateO` і `shear()`.

На рис. 4 зображено засоби відображення інформації, що створили автори, працюючи за тематикою статті.

Підключення віджета-індикатора у вигляді плагіна потребує створення окремої бібліотеки, яку Qt Designer міг би завантажувати під час своєї роботи і створювати з її допомогою екземпляри віджета. За такого підходу на формі, під час її редагування і попереднього перегляду, відобразатиметься реальний віджет.

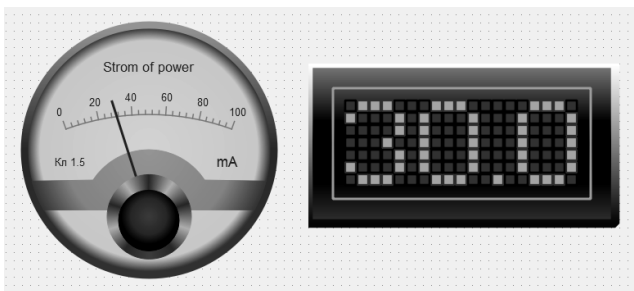


Рис. 4. Аналоговий та цифровий індикатори, створені в Qt Creator

Насамперед необхідно створити клас-нащадок від `QWidgetPlugin` і перекрити деякі віртуальні функції. Весь код можна розмістити у тих самих файлах з вихідними текстами або ж створити файли плагіна окремо.

Файл заголовка плагіна:

```
#ifndef ANALOGINDPLUGIN_H
#define ANALOGINDPLUGIN_H

#include <QDesignerCustomWidgetInterface>

class AnalogIndPlugin : public QObject, public
QDesignerCustomWidgetInterface
{
    Q_OBJECT
    Q_INTERFACES(QDesignerCustomWidgetInterface)

public:
    AnalogIndPlugin(QObject *parent = 0);

    bool isContainer() const;
    bool isInitialized() const;
    QIcon icon() const;
    QString domXml() const;
    QString group() const;
    QString includeFile() const;
    QString name() const;
    QString toolTip() const;
    QString whatsThis() const;
    QWidget *createWidget(QWidget *parent);
    void initialize(QDesignerFormEditorInterface *core);
```

private:

```
    bool m_initialized;
};
```

Клас `AnalogIndPlugin` є свого роду «фабрикою», яка виготовляє і випускає примірники віджета `AnalogInd`. Функції плагіна використовує середовище Qt Designer для створення примірників класу та отримання необхідної інформації.

```
QStringList AnalogIndPlugin::keys() const
{
    return QStringList() << "AnalogInd";
}
```

Функція `keys()` повертає список віджетів, які «випускає» плагін-фабрика. Цей плагін «випускає» тільки один віджет `AnalogInd`.

```
QWidget *AnalogIndPlugin::createWidget(QWidget
*parent)
{
```

```
return new AnalogInd(parent);
}
```

Функцію `createWidget()` викликає Qt Designer, коли необхідно створити примірник віджета. Перший аргумент – ім'я класу віджета. Він ігнорується, оскільки плагін `AnalogIndPlugin` обслуговує тільки один клас.

```
QString AnalogIndPlugin::includeFile() const
{
    return "analogind.h";
}
```

Функція `includeFile()` повертає ім'я файла заголовка віджета, який надає плагін.

```
bool AnalogIndPlugin::isContainer() const
{
    return false;
}
```

Функція `isContainer()` повертає `true`, якщо у віджет входять інші віджети, інакше – `false`. Наприклад, у `QFrame` можуть входити інші віджети. В цьому випадку повертається `false`, оскільки недоцільно робити з `AnalogInd` контейнер для інших віджетів. Тобто будь-який віджет може бути площадкою для розміщення інших віджетів, але Qt Designer не надає такої можливості, якщо `isContainer()` повертає `false`.

```
QString AnalogIndPlugin::group() const
{
    return QLatin1String("Indykators");
}
```

Функція `group()` повертає ім'я секції палітри компонентів, в якій буде розміщено віджет. Якщо такої секції немає, вона буде створена автоматично.

```
QIcon AnalogIndPlugin::icon() const
{
    return QIcon(QLatin1String(":/analog.png"));
}
```

Функція `icon()` повертає іконку для палітри компонентів.

```
QString AnalogIndPlugin::toolTip() const
{
    return QLatin1String("AnalogInd");
}
```

Функція `toolTip()` повертає текст підказки, яка з'являється, якщо навести курсор миші на іконку віджета в палітрі компонентів.

```
QString AnalogIndPlugin::whatsThis() const
{
    return QLatin1String("AnalogInd ");
}
```

Функція `whatsThis()` повертає текст, який з'являється на запит «What's This?» Qt Designer-a.

Файл з вихідним текстом плагіна повинен завершуватись викликом макроса `Q_EXPORT_PLUGIN()`.

Файл `.pro` для збірки плагіна виглядає так:

```
TEMPLATE = lib
CONFIG += plugin debug_and_release
HEADERS = analogindplugin.h
SOURCES = analogindplugin.cpp
RESOURCES = icons.qrc
QTDIR_build: DESTDIR =
$$QT_BUILD_TREE/plugins/designer
```

Змінна `QTDIR` містить шлях до каталогу, в який встановлено бібліотеку Qt. Під час компіляції плагіна командою `make` чи `make` він автоматично встановлюється у каталог `plugins` Qt Designer-a.

Після компіляції плагіна створений віджет аналогового індикатора використовується в Qt Designer, як і вбудовані віджети Qt.

Висновок. Qt Creator містить потужну бібліотеку графічних функцій, за допомогою якої можливо створювати інтерфейси користувача віртуальних вимірювальних приладів нового покоління. Qt Creator є відкритою системою, що створюється на засадах Open Source, в якій передбачено створення загальнодоступних програмних фрагментів окремими користувачами. У відкритих джерелах програмних кодів недостатньо якісних засобів відображення вимірювальної інформації.

Описані в статті технології розроблення і використання засобів відображення вимірювальної інформації можуть успішно використовуватись для програмування віртуальних приладів з розвиненим інтерфейсом користувача.

1. Шлее М. *Qt 4.8. Профессиональное программирование на C++*. – СПб.: БХВ-Петербург, 2012. – 912 с.: ил. – (В подлиннике). 2. Клименко Б. В. *Компьютерная аппаратура, аппаратура керування, запобіжники. Терміни, тлумачення, коментарі: навч. посіб.* – Х.: Талант, 2008. – 228 с. 3. Лисицын Б.Л. *Низковольтные индикаторы: справочник*. – М.: Радио и связь, 1985. – С. 63–66, 95–101, 108–123, 132–135. 4. Забродин Ю.С. *Промышленная электроника: учебник для вузов*. – М.: Высшая школа, 1982. – 496 с. 5. Гульятеев А.К., Машин В.А. *Проектирование и дизайн пользовательского интерфейса / А.К. Гульятеев, В.А. Машин*. – СПб.: Корона-Принт, 2010. – 350 с. 6. Головач В.В. *Дизайн пользовательского интерфейса. Искусство мыть слона / В.В. Головач*, 2010. – 94 с. – Режим доступа: <http://www.usethics.ru>, свободный.