

УДК 004.353

Сгадов С. О.¹, Тур'янський Е.Д.²

¹ старш. викл. НУ "Запорізька політехніка"

² студ. гр. КНТ-516 НУ "Запорізька політехніка"

ОБ'ЄКТНО-ОРІЄНТОВАНИЙ ПІДХІД ДО ДИЗАЙНУ ІНТЕРФЕЙСІВ МІКРОПРОЦЕСОРНИХ ПРИСТРОЇВ

При розробці пристроїв на мікроконтролерах широко використовуються пристрої для виводу графічної інформації такі як LCD, TFT, OLED екрани з невеликою діагоналлю порядку 1" - 6". Незважаючи на невеликий розмір ці екрани представляють досить великі можливості по відображенню графічної й текстової інформації. Наприклад, 2.2" TFT екран на драйвері ILI9341 має вирішення 240x320 з можливістю виводу 16-бітного кольору [1]. Для таких екранів розроблені бібліотеки низького рівня, наприклад, на LCD Wiki [1], які дозволяють установлювати колір пікселів і малювати графічні примітиви. Так само нескладно знайти бібліотеки для створення й виводу растрових шрифтів [2]. Проте актуальною є завдання створення й підтримки зовнішнього вигляду екрана додатка в runtime, особливо якщо вивід того самого елемента екрана відбувається в різних позначках програми. Щоб зробити створення й підтримку екранів мікроконтролерного пристрою нами була розроблена система класів мовою C++ для вистави екрана (Малюнок 1). Базовим пристроєм був обраний мікропроцесор STM32F405VE.

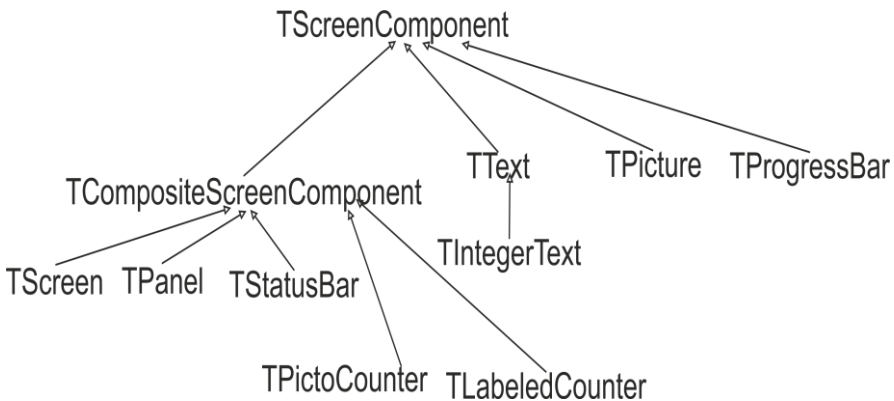


Рисунок 1 – Система класів для вистави екрана

Загальним предком для всіх класів є **TScreenComponent**. Він має координати, розміри, видимість і власника. Його основними методами є: віртуальний метод **Draw()**, який робить отрисовку компонент, віртуальний метод **Clear()**, який відповідає за очищення області компонента. Передбачається, що вони повинні бути перевизначені в спадкоємцях. Наприклад, його спадкоємці **TText**, **TPicture**, **TProgressBar** - відповідно відображають текст, картинку, прогресбар.

Наступним важливим класом є абстрактний клас складеного компонента **Tcompositescreencomponent**, спадкоємець якого може бути власником декількох інших компонентів. Для цього він має член **Components [MAX_SCREEN_COMPONENTS]** ; і метод **Add(TScreenComponent* Component)**. Більше того, якщо конструктор **TScreenComponent** ухвалює ненульовий показник на **Owner**, те він намагається додати свій екземляр у колекцію **TCompositeScreenComponent**. Це дозволяє конструювати екран або компоненти екрана на етапі конструювання класів. Наприклад, клас **TScreen** представляє абстракцію екрана і його конструктор викликає процедуру ініціалізації дисплея.

Наступним класом є **TPanel**. Основною ідеєю введення цього компонента є не тільки представлення частини екрана, яка володіє іншими компонентами, але й можливість миттєво перемикає вміст екрана, роблячи одну панель видимою, а іншу невидимою.

Наприклад наступний екран складається з рядка статусу, панелі **Panel1**, і панелі **Panel2**. **TPanel1** і **TPanel2** є спадкоємцями **TPanel** та припустимо володіють різними елементами виводу похідними від **TScreenComponent**.

Лістинг 1 – Опис класу TMyScreen

```
class TMyScreen:public TScreen
{
public:
    uint16_t bgcolor;
    TStatusBar1 StatusBar;
    TPanel1 Panel1;
    TPanel2 Panel2;
    TMyScreen(uint16_t bgcolor):TScreen(),
        StatusBar(this), Panel1(0,50,bcolor,this), Panel2(0,50,bcolor,this)
    {}
}
```

Тоді можна перемкнути вистави викликом `Panel1.Setvisible(false);`
`Panel1.Setvisible(true);`

Висновки. У даній роботі ми розробили об'єктний підхід для дизайну й виводу інформації для мікропроцесорних пристроїв. Цей підхід дуже близький до об'єктному підходу IDE типу Delphi, отже, можна розвинути його надалі в Visivig інструмент для дизайну інтерфейсів вбудованих пристроїв з використанням натівного коду на C++.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. 2.2inchSPI ModuleILI9341SKU:MSP2202[Електронний ресурс]–Режим доступу:http://www.lcdwiki.com/2.2inch_SPI_Module_ILI9341_SKU:MSP2202
2. Arduino і Android [Амперка / Вики] [Електронний ресурс] – Режим доступу: wiki.amperka.ru/android-и-bluetooth