

В.В. Гуревич

**ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ПРОГРАМНОГО УПРАВЛІННЯ
ПОГОДЖУЮЧИМ ПРИСТРОЄМ ПУЛЬТА ПЧПУ “ЕЛЕКТРОНІКА НЦ-31”***(Представлено доктором технічних наук, професором Самотокіним Б.Б.)*

Розглянуті можливі шляхи реалізації програмного інтерфейсу взаємодії з пультом ПЧПУ. Наведено конкретний приклад побудови даного інтерфейсу засобами середовища візуального програмування Delphi.

Для відпрацювання моторних навичок, необхідних для роботи на верстатах з ЧПУ, є доцільним застосування макета справжнього пульта, якого підключено до програмного імітаційного комплексу за допомогою пристрою спряження [1].

З точки зору програмування, під Windows існують два основних способи підключення будь-якого зовнішнього пристрою до комп'ютера: через драйвер пристрою та за допомогою прямого звернення до відповідних системних ресурсів (у даному випадку до портів вводу/виводу паралельного інтерфейсу). Перший варіант характеризується наступними особливостями: він вимагає наявності спеціального програмного забезпечення (Driver Development Kit – DDK), глибокого знання принципів функціонування Windows на системному рівні. При цьому драйвер необхідно встановлювати на кожному комп'ютері, де використовується тренажер. Другий варіант підключення постає більш переважним, оскільки при програмуванні паралельного порту не виникає особливих ускладнень; стає можливим максимально швидко відпрацьовувати події від зовнішнього пристрою, що важливо з урахуванням інтенсивного використання системних ресурсів для тривимірного моделювання. Крім того, зводиться до мінімуму розмір інсталяційного пакета і не потребує додаткових дій при встановленні імітаційного комплексу на новий комп'ютер.

Оскільки програмна частина тренажера створюється у середовищі візуального програмування Delphi, то представляється зручним створення користувацького компонента, котрий буде містити всі необхідні для конкретного ПЧПУ властивості. Це надає можливість при створенні кінцевого програмного забезпечення абстрагуватися від особливостей взаємодії з пристроєм, що підключається. Користувач буде мати справу з подіями та властивостями даного компонента, які є характерними для цього пристрою в цілому. В нашому випадку це події натискання та відпускання клавіш, а також властивості стану відповідних цифрових індикаторів або світлодіодів. Отже, компонент, що розробляється, повинен мати два обробника подій (доцільно інформацію про натиснуту клавішу передавати як аргумент) та чотири властивості (три властивості для кожної групи цифрових індикаторів і одну – для всіх світлодіодів). Є сенс організувати взаємодію з цифровими індикаторами наступним чином: при встановленні значення властивості для відповідної групи індикаторів компонент бере на себе реалізацію взаємодії з кожним з індикаторів групи окремо. Іншими словами, значення властивості, що присвоюється, автоматично відображається відповідною групою індикаторів. Для світлодіодів найбільш зручною формою представлення є властивість-масив (array-property) з індексом, який означає номер індикатора, і властивістю логічного типу: горить (true), не горить (false). Для цього попередньо всі індикатори нумеруються у зручному для користувача порядку – за їх розташуванням на пульті.

Враховуючи, що в пульті ПЧПУ “Електроніка НЦ-31” не використовуються переривання, то при натисканні клавіш на ньому у Windows не виникають відповідні події. В цьому типі ПЧПУ робота організована шляхом періодичного опиту відповідних реєстрів пульта [2].

Таким чином, аналогічний спосіб взаємодії необхідно реалізувати під Windows. Для цього треба створити в Delphi компонент, який використовує можливості таймерного оповіщення. Найбільш близьким за своїми функціональними можливостями є компонент TTimer.

При цьому існує два шляхи створення нового компонента [3]: розширення існуючого або створення абсолютно нового. Враховуючи особливості, притаманні компоненту TTimer (включення таймера відбувається при одночасному виконанні двох умов: властивість Enable встановлено в True та присутній обробник подій onTimer), було обрано другий спосіб створення нового компонента. За основу було взято висхідний програмний код компонента TTimer, що знаходиться в модулі **extctrls.pas**. У цьому коді було змінено умову запуску таймера (достатнім є встановлення властивості Enable в True), змінено значення за замовчуванням властивості Interval (10 мс), розширена процедура обробки Windows-повідомлення WM_TIMER для обслуговування програмно-апаратного інтерфейсу з пультом і додані вищевказані події та властивості:

```
TBoard = class(TComponent)
private
  ...
public
  ...
  procedure AKK(b:boolean);
  procedure Reset;
  property LED[n:ledRange]:boolean read GetLed write SetLed;
  property DIGIT[n:digitsRange]:digitRange read GetDigit write SetDigit;
  property FEED:feedRange read GetFeed write SetFeed;
  property FRAME :FrameRange read GetFrame write SetFrame;
  property DISPLAY:DisplayRange read GetDisplay write SetDisplay;
published
  ...
  property OnKeyDown: TVVSKeyDownEvent read FOnKeyDown write SetOnKeyDown;
  property OnKeyUp : TVVSKeyUpEvent read FOnKeyUp write SetOnKeyUp;
end;
```

Зокрема, властивості OnKeyDown та OnKeyUp призначені для обробки подій, що відбуваються при натисканні і відпусканні клавіш на пульті ПЧПУ, властивість LED керує світлодіодами, властивість DIGIT дозволяє за необхідністю взаємодіяти з кожним окремим цифровим індикатором, властивості FEED, FRAME та DISPLAY керують відповідними групами цифрових індикаторів: подачі, номери кадру та переміщення інструмента. Дані властивості дозволяють не тільки встановити значення відповідних індикаторів, але й одержувати інформацію про їх стан. Для зручності також є доступною низка функцій для безпосереднього керування пультом: AKK – встановлення індикатора "Батарея заряджена", Reset – перевід пульта в початковий стан тощо.

Через те, що у мові Delphi Object Pascal відсутні команди роботи з портами, ця можливість була реалізована за допомогою асемблерних вставок. Нижче наведено приклад встановлення відповідної тетради для зчитування даних з пульта (керування мультиплексором [1]):

```
procedure TBoard.StrobeMux(b:boolean);
var a:byte;
begin
```

```
asm
  mov dx,ctrl
  in al,dx
  mov a,al
end;
if b then
  asm
    mov dx,ctrl
    mov al,a
    or al,4
    out dx,al
  end
else
  asm
    mov dx,ctrl
    mov al,a
    and al,0FBh
    out dx,al
  end;
delay;
end;
```

Таким чином, створення користувацького компонента істотно спрощує взаємодію нових імітаційних програм, що розробляються, орієнтованих на використання даного пульта. При роботі з іншим типом ПЧПУ програма не потребує змін: необхідно скорегувати тільки внутрішню реалізацію компонента TBoard, приділяючи увагу тільки особливостям конкретного пульта.

ЛІТЕРАТУРА:

1. Організація підключення пульта ПЧПУ "Електроніка НЦ-31" до персонального комп'ютера типу IBM / В.В. Гуревич, В.В. Сидоренко // Вісник ЖІТІ, 1997. – № 6. – С. 172–174.
2. Устройства числового программного управления "Электроника НЦ-31". Инструкция по эксплуатации, 1983. – 217 с.
3. Конопка Р. Создание оригинальных компонентов в среде Delphi / Пер. с англ. Рэй Конопка. – К.:НИПФ, "ДиаСофт Лтд.", 1996. – 512 с.

ГУРЕВИЧ Віктор Вікторович – аспірант ЖІТІ.

Наукові інтереси:

- сучасні системи навчання;
- комп'ютерні технології та обчислювальна техніка.