
УДК 621.396.6

О.Ю. МАЛИЙ, О.Ю. ФАРАФОНОВ, О.А. ДЕМ'ЯНЕНКО

МЕТОД ОПИСУ ІНЖЕНЕРНИХ ЗАДАЧ ЗА ДОПОМОГОЮ УНІВЕРСАЛЬНОЇ МОВИ ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ

Пропонується метод, що дозволяє проводити опис інженерної задачі у вигляді програми на УМП для подальшої трансляції з метою автоматизованої генерації тексту програми обраного мікроконтролера. Метод дозволяє не прив'язуватися до окремих фірм виробників периферійних пристроїв та мікроконтролерів, що в свою чергу дозволяє в підсумку отримати оптимальне схемотехнічне рішення інженерної задачі з мінімальними витратами для виробника.

1. Вступ

Для створення кінцевого пристрою і написання програми керуючого мікроконтролера першим етапом є осмислення інженерного завдання і опис структури пристрою. Для цього інженер, який є користувачем системи автоматизації програмування, повинен першочергово осмислити, який кінцевий результат він хоче побачити при проектуванні пристрою. Процес створення радіоелектронної системи на основі мікроконтролера, з точки зору осмислення принципів її роботи та опису для системи автоматизації проектування, можна розбити на три основних етапи:

1. Визначення вхідних і вихідних пристроїв, які будуть обмінюватися інформацією з кінцевим користувачем системи і навколишнім середовищем.
2. Визначення структури складових частин мікроконтролерної системи.
3. Визначення принципів взаємодії складових частин та загального алгоритму роботи системи.

2. Класифікація пристроїв, що підключаються до мікроконтролерів

Для визначення вхідних і вихідних пристроїв в першу чергу необхідно класифікувати периферійні пристрої, що підключаються до мікроконтролерів та можуть бути виконані у вигляді як зовнішніх пристроїв, так і вбудованої в мікроконтролер периферії. Класифікація пристроїв за принципами організації зв'язку з мікроконтролером, а також по взаємодії з користувачем і навколишнім середовищем наведена на рис. 1.

Як видно з рис. 1, пристрої, що підключаються до мікроконтролерів, можна розділити на вхідні, вихідні та пристрої двостороннього зв'язку.

Вхідні пристрої здатні лише передавати будь-яку інформацію мікроконтролеру для подальшої обробки. До них можна віднести всі види датчиків, пристрої введення, такі як кнопки, клавіатура, а також компаратори, АЦП, ЦАП (у разі підключення їх виходів до мікроконтролера).

До вихідних відносяться пристрої, здатні тільки виводити дані. До них відносяться різні типи індикаторів (світлодіоди, рідкокристалічні індикатори, семисегментні індикатори, дисплеї), виконавчі пристрої (електромеханічні, електричні), АЦП, ЦАП, регістри (у разі підключення їх входів до мікроконтролера).

До пристроїв двостороннього зв'язку можна віднести різні види зовнішньої та вбудованої пам'яті й інші пристрої, які дозволяють проводити обмін даними з мікроконтролером за стандартними шинами обміну [1].

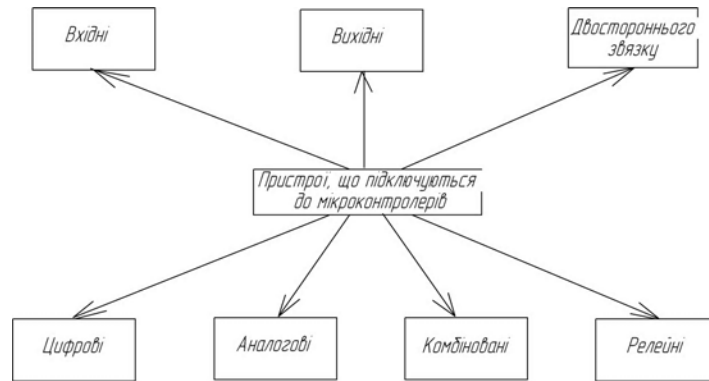


Рис. 1. Класифікація пристроїв, що підключаються до мікроконтролера

3. Опис периферійних пристроїв за допомогою універсальної мови програмування мікроконтролерів

За допомогою запропонованих в [2-6] методів опису структури мікроконтролерів і периферійних пристроїв, що входять до його складу, можна описувати будь-який тип периферійних пристроїв, які підключаються до різних мікроконтролерів. В кінцевому програмному комплексі для цих цілей використовується файл, що містить докладний опис принципів роботи периферійних пристроїв *device.dat*. Тому користувач/інженер може в програмному комплексі вибирати зі списку необхідний йому тип пристрою, а в разі необхідності конкретний пристрій для подальшого його підключення до мікроконтролера. Завдяки відкритій архітектурі файлу опису пристроїв, а також наявності вбудованих інструментів створення і редагування опису пристроїв, існує можливість додавання як нових типів пристроїв, так і нових конкретних пристроїв, що випускаються різними виробниками.

Для вирішення завдання реалізації першого етапу процесу створення радіоелектронної системи на основі мікроконтролера, з точки зору осмислення принципів її роботи та опису для системи автоматизації проектування, тобто визначення вхідних і вихідних пристроїв, які будуть обмінюватися інформацією з кінцевим користувачем системи і навколишнім середовищем, у програмному комплексі передбачена можливість вибору і додавання пристроїв у проєкт. З точки зору програмної реалізації дана процедура відбувається шляхом додавання в проєкт нового об'єкта, що має певну архітектуру, властивості та функції для управління ним і реалізації взаємодії з мікроконтролером.

Наприклад, термопара, яка за запропонованою класифікацією відноситься до вхідного аналогового пристрою, описується так:

Thermocouple:

Type, commands, tmin, Udt, tmax, gisteresise, dt, gisk, pins.

Після додавання в проєкт цей об'єкт має такі функції для процедури управління ним і реалізації взаємодії з мікроконтролером: *overload* (true, false) – вихід за верхню межу t° ; *notload* (true, false) – вихід за нижню межу t° ; *work* (true, false) – режим нормальної роботи *readtemp* – функція читання t° .

На другому етапі користувач / інженер вказує структуру взаємодії запропонованих пристроїв з мікроконтролером. З математичної точки зору це відбувається як присвоєння доступними для даного об'єкта параметрами і функціями певних значень, які необхідні для подальшого вибору мікроконтролера і створення блоків ініціалізації пристроїв у коді програми мікроконтролера, а також включення переривань.

Як приклад розглянемо підключення конкретного типу термопари (не задаючись конкретно моделлю даного пристрою для більшої гнучкості використання моделей в кінцевій мікроконтролерній системі) до мікроконтролера для введення з навколишнього середовища значення температури, яка змінюється в діапазоні від 0 до 100°C. При цьому будемо вважати, що коефіцієнт термоЕРС даного типу термопари дорівнює 50мкВ/°С, при мінімальній вимірюваній температурі (тобто при 0) вихідна напруга має значення 100мВ, похибку вимірювання приймемо 0,1 °С. Для спрощення завдання будемо вважати, що ефект гістерезиса для даної термопари відсутній. У цьому випадку об'єкт thermocouple буде мати такі значення параметрів:

```
Type = analog
commands = no
tmin = 0
Udt = 50(мкВ/°С)
tmax = 100
gisteresise = none
u0 = 100(мВ)
dt = 0.1
gisk = 0
pins = 2
```

На третьому етапі процесу створення радіоелектронної системи на основі мікроконтролера, з точки зору осмислення принципів її роботи та опису, для системи автоматизації проектування користувач / інженер визначає принципи взаємодії складових частин та загальний алгоритм роботи системи.

4. Приклади реалізації введення інженерних задач в систему автоматизації програмування

Пропонується для опису функціонування розроблюваної мікроконтролерної системи використовувати описану в [2-6] універсальну алгоритмічну мову програмування мікроконтролерів шляхом вказання логічних взаємозв'язків між об'єктами, прив'язаними до конкретних пристроїв підключеним користувачем на попередніх етапах планування завдання. Опис проводиться за допомогою встановлення чіткого взаємозв'язку вхідних і вихідних пристроїв з використанням, як посередника, мікроконтролера. Взаємозв'язок описується шляхом встановлення умовних зв'язків між об'єктами з допомогою, в першу чергу, оператора умови (if) і оператора присвоєння (=), а також за допомогою мінімального набору функцій і процедур універсальної алгоритмічної мови програмування мікроконтролерів. Як приклад розглянемо опис двох мікроконтролерних систем, в яких використовуються різні види вхідних і вихідних пристроїв.

Перша система, яка пропонується до розгляду як приклад, термометр з індикацією температури на текстовому однорядковому монохромному дисплеї. Система складається з чотирьох основних елементів:

- термопари (властивості, параметри і функції якої розглянуті вище);
- кнопки включення / виключення, яка описана у вигляді вхідного цифрового релейного пристрою з єдиним параметром pin_in, дозволяє встановити в одиницю або скинути в нуль вхід порту мікроконтролера;
- монохромного однорядкового восьмисимвольного текстового дисплею, об'єкта типу display, який описаний у вигляді вихідного цифрового паралельного пристрою;
- безпосередньо керуючого мікроконтролера.

При цьому вважаємо, що система вимірює температуру від 0 до 100°C з кроком 0,5 °С за допомогою термопари. Значення температури виводиться на дисплей у вигляді числа з комою, тобто у форматі від 1 до 3-х цифр до коми, сама кома, одна цифра після коми і значок градуса Цельсія, який складається з двох символів "o" і "C". Кнопка використовується для включення і виключення розроблюваної системи.

У цьому випадку алгоритм роботи даної мікроконтролерної системи матиме такий вигляд (рис. 2):

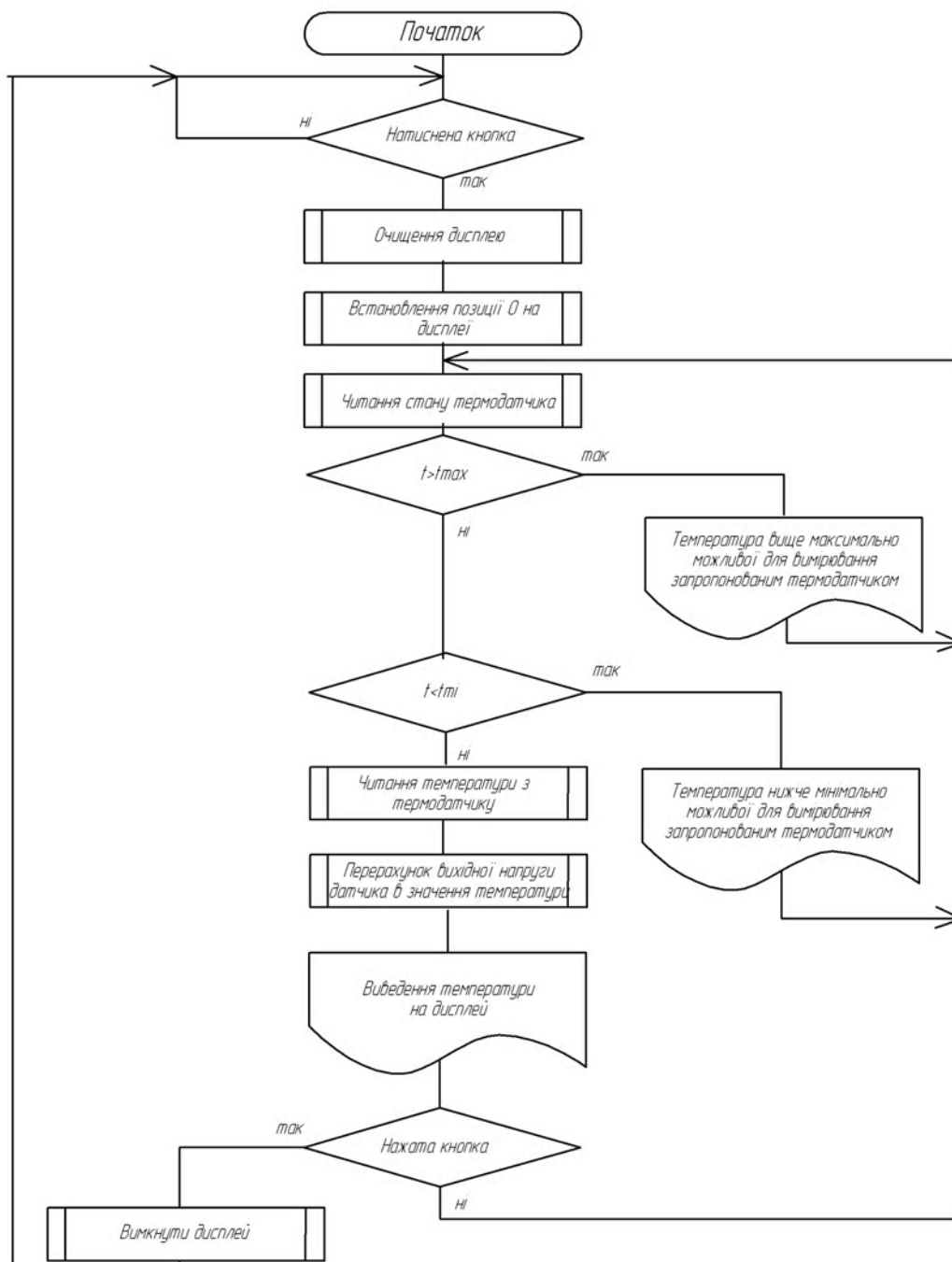


Рис. 2. Алгоритм роботи термометра з індикацією температури на текстовому однорядковому монохромному дисплеї

На універсальній алгоритмічній мові програмування (УМП) даний алгоритм роботи пристрою з урахуванням введених пристроїв у вигляді об'єктів описаний в табл. 1.

З прикладу видно, що інженеру немає необхідності вникати в тонкощі перетворення аналогового сигналу з виходу термопари в значення температури, а також в особливості системи команд управління дисплеєм, а можна використовувати тільки функції об'єктів *thermocouple* і *display*.

Друга система, яку ми розглянемо як приклад, являє собою електронний замок, який працює з використанням електронних смарткарт і управляє електронним замком і двома світлодіодами (червоним і зеленим), які індикують стан замка.

Система складається з п'яти основних елементів.

Таблиця 1

```

m2: if sw1.swon = true then
    pause(500)
endif
    if sw1.swof = true then
        display1.positionset = 0
m1: if thermocouple1.work then
    if thermocouple1.overload = true then
        display1.positionset = 7
        display1.load text = 'temp > max'
    goto m1
    endif
    if thermocouple1.notload = true than
display1.positionset = 7
        display1.load text = 'temp < min'
        goto m1
    endif
    s = thermocouple1.readtemp
temp = S/termocople1.Udt
display1.positionset = 6
If temp < 100 then
Display1.setsimbol = ' '
endif
t= int(temp/100)
if t <>0 then display1.setsimbol = simb(t)
endif
    display1.position set = 5
If temp < 10 then
display1.setsimbol = ' '
endif
temp = temp-t*100
t=int(temp/10)
if t >=10 then
display1.setsimbol =simb(t)
endif
display1.positionset = 4
temp = temp - t*10
t = int (temp)
display1.setsimbol = simb (t)
display1.positionset = 3
display1.setsimbol = ' '
display1.positionset =2
temp=temp - t
t=int(temp*10)
display1.setsimbol = simb (t)
display1.positionset = 1
display1.setsimbol = 'o'
display1.positionset = 0
display1.setsimbol = 'C'
endif
if sw1.swon=true then
display1.dispon=false
pause(500)
go to m2
endif
goto m1

```

– картоприймача / роз'єму для підключення смарткартки з контактом, що визначає наявність картки (в даному випадку незважаючи на пасивність цього пристрою, він описується як цифровий послідовний пристрій двостороннього зв'язку);

– двох світлодіодів, які описані як вихідні цифрові релейні пристрої з єдиним параметром pin_out, що дозволяє вмикатися, якщо вихід порту мікроконтролера дорівнює одиниці, і вимикатися, якщо вихід порту дорівнює нулю;

– електромеханічного замка (в електронній схемі буде виконуватися тільки роз'єм для його підключення), який працює як реле і отже описаний у вигляді вихідного цифрового релейного пристрою єдиним параметром pin_out (зелений світлодіод за бажанням інженера можна запаралелити із замком);

– енергонезалежної пам'яті, що зберігає набір дамів пам'яті дозволених карт, яка описана як послідовний пристрій двостороннього зв'язку, об'єкта типу eeprom (пам'ять може бути реалізована як всередині мікроконтролера, так і зовні у вигляді окремої мікросхеми);

– безпосередньо керуючого мікроконтролера.

При цьому вважаємо, що в початковому стані запалюється червоний світлодіод, система в разі наявності картки в роз'ємі (це визначається наявністю сигналу на контакті картоприймача) виробляє читання дампа пам'яті картки, а також реалізує алгоритм аутентифікації карти challenge / response і порівнює зчитаний дамп з дозволеними дампами для доступу, які зберігаються в незалежній пам'яті. Якщо дамп, зчитаний з карти, збігається з яким-небудь дампом, що зберігаються в пам'яті, і карта пройшла алгоритм аутентифікації, то червоний світлодіод гасне, а загоряється зелений і разом з тим подається сигнал на вхід електромеханічного замка, який після цього відкривається. Після відкриття система відміряє 10 секунд, потім замок закривається, зелений світлодіод гасне і спалахує червоний, і система переходить в початковий стан.

У цьому випадку алгоритм роботи даної мікроконтролерної системи матиме такий вигляд (рис. 3).

На універсальній алгоритмічній мові програмування даний алгоритм роботи пристрою з урахуванням введених пристроїв у вигляді об'єктів описаний в табл. 2.

З цього прикладу також видно, що інженеру немає необхідності вникати в тонкощі обміну даними між пам'яттю і мікроконтролером, а також процесом роботи мікроконтролера зі смарткарти, а просто використовуються функції об'єктів eeprom і smartcard.

Таблица 2

```

lock1.pin_out=false
vs1.pin_out=true
vs2.pin_out=false
m:8
if cardreader1.cardset=true then
  cardreader1.reset
  byte=0
m2:if bit<128 then
  bit=0
  byte=byte+1
  card=dump[byte]=0
m1: if <8 then

card_dump[byte]=card_dump[byte]*2+cardreader1.read
  bit=bit+1
  goto m1
endif
  goto m2
endif
  num=0
  eeprom1.setadd=0
m4: if num<16 then
  byte=0
  num=num+1
m3: byte=byte<128 then
  byte=byte+1
  ee_dump[num,byte]=eeprom1.readword
  ee_prom1.setadd=eeprom1.getadd+1
  goto m3
endif
  goto m4
endif
m5: if num<16 then
  byte=0
  num=num+1
m5: if byte<128 then
  byte=byte+1
  if card_dump[byte]<>ee_dump[num,byte] then
  goto m5
endif
  goto m6
endif
  goto m7
end if
  goto m8
m7: byte=0
  cardreader1.next
m14: if byte<14 then
  byte=byte+1
  bit=0
m13: if bit<8 then
  t=cardreader1.read
  if byte=13 then
  if byte=3 then

```

```

cardreader1.write
  ch[1]=08
  ch[2]=12
  ch[3]=19
  ch[4]=81
  need[1]=22
  need[2]=06
  need[3]=19
  need[4]=41
  i=0
m10: if i<4 then
  j=0
  i=i+1
m9: if j<8 then
  t=ch[i] and 128
  cardreader1.pin[io]=t
  ch[i]=ch[i]*2
  j=j+1
  goto m9
endif
  goto m10
endif
  i=0
m12: if i<4 then
  j=0
  i=i+1
  resp[i]=0
m11: if j<8 then
  cardreader1.write
  t=cardreader1.pin[io]
  resp[i]=resp[i]*2+t
  j=j+1
  goto m11
endif
  goto m12
endif
  end if
  goto m14
end if
  i=0
m15: if i<4 then
  i=i+1
  if resp[i]<>need[i] then
  goto m8
endif
  goto m15
endif
  lock1.pin_out=true
  vs1.pin_out=false
vs2.pin_out=true
pause(10000)
goto m8

```

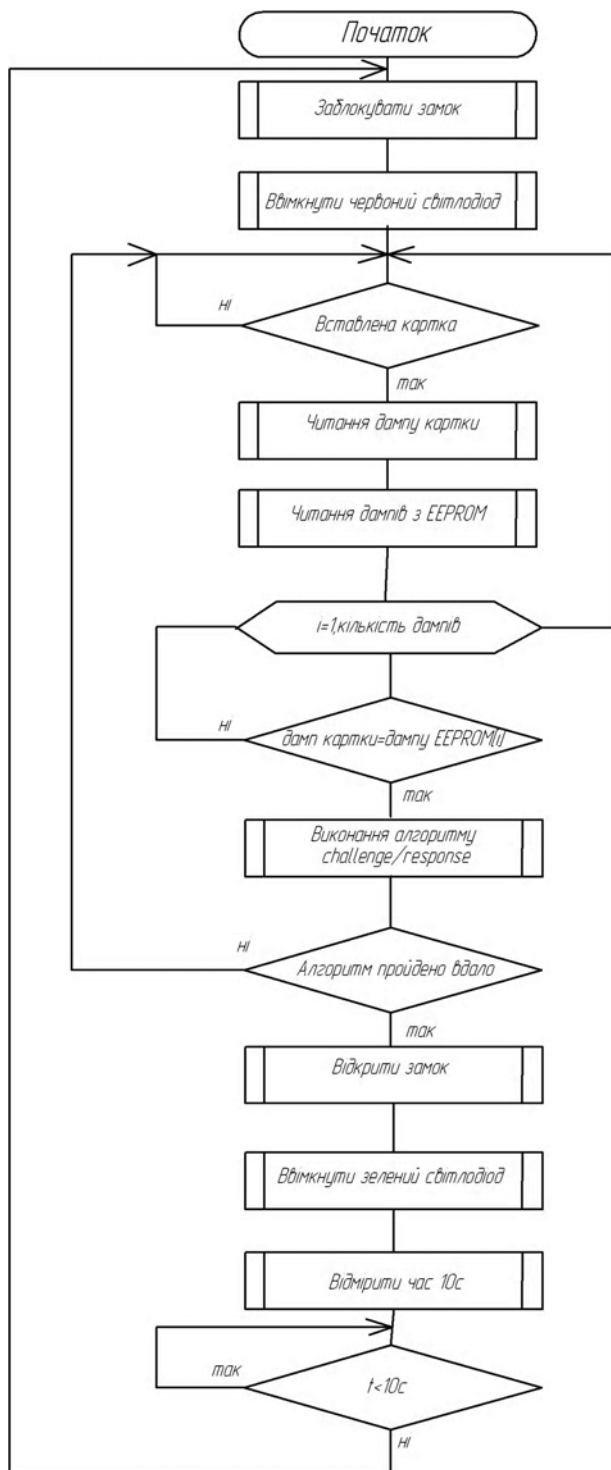


Рис. 3. Алгоритм роботи електронного замка

5. Висновки

Запропонований метод дозволяє проводити опис інженерної задачі у вигляді програми на УМПІ, що дає можливість далі шляхом трансляції проводити автоматизовану генерацію тексту програми обраного мікроконтролера. На відміну від раніше запропонованих методів дозволяє не прив'язуватися до окремих фірм виробників периферійних пристроїв та мікроконтролерів. Завдяки цьому в підсумку можна отримати оптимальне схемотехнічне рішення інженерної задачі з мінімальними витратами для виробника.

Список літератури: 1. Сташин В.В. и др. Проектирование цифровых устройств на однокристальных микроконтроллерах / В.В. Сташин, А.В. Урусов, О.Ф. Мологонцева. М.: Энергоатомиздат, 1990. 223с. 2. Vladimir Krischuk, Alexandr Maly. Development of methods of designing of means of the automated creation of the programs of microcontrollers in difficult electronic systems // "The experience of designing and application of CAD Systems in Microelectronic", Lviv, Ukraine, 2005. P. 279-280. 3. Кришук В.М., Малий О.Ю. Універсальна алгоритмічна мова програмування мікроконтролерів //Тези доповідей Міжнародної науково-практичної конференції "Сучасні проблеми і досягнення в галузі радіоелектроніки, телекомунікацій та інформаційних технологій". Запоріжжя, 2006. С. 152-153. 4. Alexandr Maly, Volodimir Krischuk The dynamic description of system of instructions of microcontrollers // CADSM-2007, Polyana, 21-25 Feb.2007. 5. Малий О.Ю. Універсальна алгоритмічна мова програмування мікроконтролерів/ В.М. Кришук, О.Ю. Малий, О.Ю. Воропай// Радіоелектроніка. Інформатика. Управління. 2009. №2. С. 115-120. 6. Малий А.Ю. Математическое обеспечение автоматизации разработки и проверки программ микроконтроллеров /А.Ю. Воропай, В.Н.Кришук, А.Ю. Малий //Радіотехніка. 2009. №3. С. 165-169. 7. Малий О.Ю. Розробка методів універсалізації програмування мікроконтролерів різних сімейств/ А.Ю. Малий //Радіотехніка. 2009. №4. С. 210-225.

Надійшла до редколегії 21.11.2011

Малий Олександр Юрійович, асистент каф. конструювання и технологии производства

радіоапаратури Запорізьського національного технічного університету. Адреса: Україна, 69063, Запоріжжя, вул.Жуковського, 64, тел. роб. (061)7698252, e-mail: docsasha@mail.ru.

Фарафонов Олексій Юрійович, канд.техн.наук, доцент кафедри конструювання та технології виробництва радіоапаратури Запорізьського національного технічного університету. Адреса: Україна, 69063, Запоріжжя, вул.Жуковського, 64, тел. роб. (061)7698252, e-mail: farafon@zntu.edu.ua

Демьяненко Олексій Анатолійович, студент магістратури гр.РП-116 Запорізьського національного технічного університету. Адреса: Україна, 69063, Запоріжжя, вул. Жуковського, 64.