

УДК 681.3.07

## РЕАЛІЗАЦІЯ СХЕМ СИСТЕМИ БАГАТОКОМПОНЕНТНОГО ДОЗУВАННЯ У PROTEUS

*Тейкін Д.В.*

*Харківський національний автомобільно-дорожній університет, Харків*

Можливість переходу до прогресивніших і економічно доцільніших методів автоматизованого управління пов'язана зі зміною технічної бази будівельного виробництва і комплектуванням його новітніми засобами мікропроцесорної техніки та програмами моделювання [1].

Для реалізації системи дозування матеріалів потрібно: промисловий контролер; модуль управління заслінками дозаторів; модуль збору даних. Для моделювання роботи запропонованої системи управління обрано симулятор Arduino та середовище Proteus. При багатокомпонентному дозуванні декілька дозаторів безперервної дії з автономним керуванням об'єднуються в систему, котра може функціонувати наступним чином.

Варіант 1. Незв'язне дозування декількох компонентів з продуктивністю, що підтримується у заданому постійному співвідношенні (рис. 1), здійснюється від загального для усіх дозаторів каналу. Сигнал від програмного задатчику, що визначає значення продуктивності всіх компонентів, подається для операцій множення на постійний коефіцієнт ( $0,2 \leq K_m \leq 2$ ) на аналого-множувальні пристрої АМП, число яких дорівнює числу дозаторів; значення співвідношення можна змінювати зміною коефіцієнтів  $K1 - K3$ .

Варіант 2. Функціональне дозування декількох компонентів з корекцією співвідношення та паралельним з'єднанням дозаторів. У такий спосіб дозатор 1 основного компоненту приймається провідним. На нього поступає сигнал завдання через функційний блок АМУ чи програмний задатчик. Цей же сигнал є вхідним для функціональних блоків відомих дозаторів 2 та 3, на котрі поступають сигнали, що коректують продуктивність кожного дозатору.

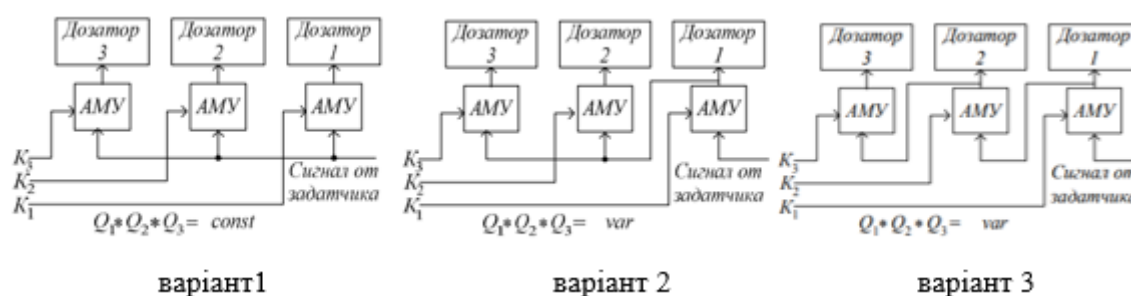


Рисунок 1 – Структурні схеми дозування компонентів

Варіант 3. Функціональне дозування декількох компонентів з корекцією співвідношення та послідовним з'єднанням дозаторів. У такий спосіб при трьохкомпонентному дозуванні дозатори 2 та 3 приймаються у якості відомих, що вмикаються послідовно з провідним дозатором 1 основного матеріалу.

Розглянемо моделювання процесу дозування за першим варіантом. Кількість дозаторів визначає кількість координат управління:  $X_{n+}$  - відкрити бункер  $n$ ; -  $X_{n-}$  - закрити бункер  $n$ . Для кожного кроку керування формуємо комбінацію керуючих сигналів  $X+$ ,  $X-$  і тривалість кроку  $T$ , с (таблиця 1).

Таблиця 1 – Сигнали управління дозаторами

Dir	X1+	X1-	X2+	X2-	X3+	X3-	T, с
0	0	0	0	0	0	0	1
1	1	0	0	0	0	0	3
2	0	1	0	0	0	0	1
3	0	0	1	0	0	0	9
4	0	0	0	1	0	0	1
5	0	0	0	0	1	0	11
6	0	0	0	0	0	1	1
7	0	0	0	0	0	0	7

При створенні програми визначаються змінні й оператори середовища програмування Arduino, необхідні для реалізації алгоритму функціонування мікроконтролера та формування керуючих дій (рис.2). Відповідно до алгоритму роботи створюється текст програми. У результаті компіляції програми одержуємо повідомлення про розмір коду для завантаження у пам'ять мікроконтролера. Середовище Proteus [4] дає можливість побудувати

схему мікроконтролерного пристрою (рис.3), зв'язати мікроконтролер з hex-файлом місткості пам'яті програм та виконати моделювання роботи пристрою за заданим алгоритмом.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);
//define engines
int x1=7;
int x2=6;
int x3=5;
int N=4;
int F=3;
int D=2;
float temperature=20;
void setup() {
  lcd.begin(16,2);
  pinMode(x1,OUTPUT);
  pinMode(x2,OUTPUT);
  pinMode(x3,OUTPUT);
  pinMode(N,OUTPUT);
  pinMode(F,OUTPUT);
  pinMode(D,OUTPUT);
}

  digitalWrite(X, HIGH);
  digitalWrite(Y, HIGH);
  digitalWrite(Z, HIGH);
delay(1000);
  digitalWrite(X, LOW);
  digitalWrite(Y, LOW);
  digitalWrite(Z, LOW);
  delay(1000);
  digitalWrite(Y, HIGH);
  digitalWrite(Z, HIGH);
delay(1000);
  digitalWrite(X, LOW);
  digitalWrite(Z, LOW);
  delay(1000);
```

Рисунок 2 – Вид програми й результаткомпілювання

При зміні рецептури багатокомпонентного дозування в автономному режимі може змінюватися алгоритм керуючих дій (програмнийскетч) в залежності від варіанту (схеми) керування дозаторами.

Сумісні програмні симулятори типу Proteus дають можливість перевірити логіку роботи пристрою управління на макеті.

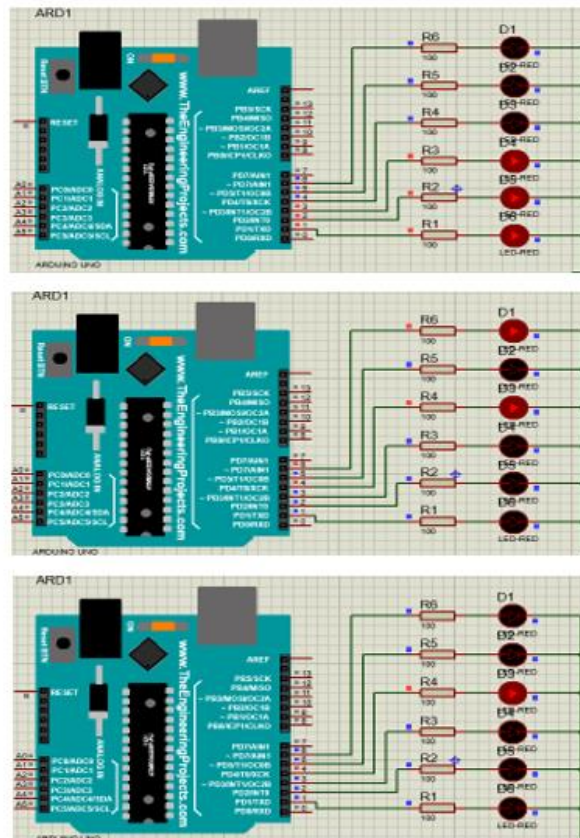


Рисунок. 3 – Результати моделювання

Моделювання у Proteus сприяє придбанню практичних навичок програмування й симуляції, за рахунок виконання попередньої обробки показань датчиків засобами Arduino, що підвищує гнучкість та можливість зміни апаратного забезпечення без зміни програмного коду.

### Література:

- [1] Плугіна Т.В. Інтелектуальна система контролю якості робочих процесів будівельно-дорожніх машин (БДМ) / Т.В. Плугіна, О.В. Єфименко. Вісник ХНАДУ. – 2019.– №. 87. – Т. 1 – С. 66-73. DOI: 10.30977/BUL.2219-5548.2019.87.0.66
- [2] Стеценко І.В. Моделювання систем: навч. посіб. [Електронний ресурс, текст] / І.В. Стеценко; М-во освіти і науки України, Черкас. держ. технол. ун-т. – Черкаси: ЧДТУ, 2010. – 399 с. ISBN 978-966-402-073-9
- [3] Proteus\_vsm\_ru.pdf . [Електронний ресурс]: [http:// proteus\\_vsm\\_ru.pdf. net/](http://proteus_vsm_ru.pdf.net/) (дата звернення 10.10.2020). — Назва з екрана.