



УДК 621.62

## ДОСЛІДЖЕННЯ РЕЖИМІВ РОБОТИ МІКРОПРОЦЕСОРНИХ СИСТЕМ

Студ. І.В. Мосур, гр. МгАк-17

Науковий керівник Л.П. Голубєв

Київський національний університет технологій та дизайну

**Мета і завдання.** Метою є збільшення часу автономної роботи мікропроцесора за допомогою режиму енергозбереження і технології переривань.

Завдання – розробити технічну частину системи, розробити алгоритм і програмну частину системи енергозбереження мікропроцесорної техніки.

**Об'єкт дослідження.** Режими енергозбереження мікропроцесорних систем, що були розглянуті на прикладі 8-розрядного мікроконтролера АТmega8.

**Методи та засоби дослідження.** При вивченні об'єкта дослідження використовувалися такі основні методи і способи: аналітичний, системний та програмно-технічний, експериментальний метод, вимірювання, порівняння, моделювання та аналіз. Також проводили розрахунки для визначення ефективності даного дослідження.

### **Наукова новизна та практичне значення отриманих результатів.**

Вперше розроблено програмно-технічний комплекс контролю енерговитрат мікропроцесорних систем за допомогою режимів енергозбереження та системи переривань. Таке технічне рішення дозволить збільшити час автономної роботи мікропроцесорної системи.

**Результати дослідження.** Розглянемо чому взагалі мікроконтролер взагалі споживає енергію. В мікроконтролері є багато-багато транзисторів, що працюють в так званому ключовому режимі (по типу звичайної кнопки). Основне споживання потужності йде в момент перемикання ключа, тобто чим менше перемикаць - тим менше енергоспоживання. У звичайному режимі роботи сам чіп на 8 МГц споживає приблизно 10,2 мА. У звичайній же пальчиковій батарейці - 1700-3000 мАг. Виходить, що мікросхема буде працювати від 17 годин до 12 діб. А правильну "якісну" батарейку потрібно пошукати. У плоскій літєвої батареї, яку ми використовували - взагалі 210 мАг. Такий "звичайний" режим роботи абсолютно не підходить для автономних пристроїв, таких як дистанційні вимикачі, пристрої охорони, годинники і метеостанції, і інші.

Є кілька способів знизити енергоспоживання мікроконтролера: зменшити частоту осцилятора, вимкнути осцилятор, зменшити кількість перемикаючих вентелей, відключивши ядро і/або додаткові генератори і таймери, відключити аналогову периферію (вимагає багато енергії), перевести які тільки можна ніжки мікроконтролера в Z-стан.

Частина цих способів автоматично виконується при включенні режиму «сон». Розглянемо їх докладніше. Режим Idle, Power-Down, Power-save, Standby, Extended Standby. Режими Idle і Power-Down зустрічаються найбільш часто. Цікавим для нас буде режим ADC Noise Reduction. Цей режим зупиняє процесор, але залишає працювати АЦП, систему зовнішніх переривань, інтерфейс TWI, сторожевий таймер і таймер / лічильник<sup>2</sup>; відповідно, зупиняються тільки годинник I / O, CPU і FLASH. Все це призначено для того, щоб зменшити внутрішні перешкоди (що виникають при перемиканні транзисторів самої мікросхеми) при роботі АЦП - так виходять вимірювання більш високої точності. На рис.1 зображений режими роботи програми.



Рисунок 1 – Робота програми в сплячому та штатному режимах

Виходить, що в більшості режимів мікросхема взагалі не розуміє, що була в сплячому режимі: тільки виконала останню команду перед sleep, і одразу виконується переривання яке просипає пристрій. За роботу зі сплячими режимами відповідає регістр MCUCR.

Нам будуть потрібні біти SM0, SM1 I SM2, якщо він є – щоб задати тип режиму сну. Так, наприклад, в ATmega8 режими сну задаються так:

SM2	SM1	SM0	Sleep Mode
0	0	0	Idle
0	0	1	ADC Noise Reduction
0	1	0	Power-down
0	1	1	Power-save
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Standby

Рисунок 2 – Визначення режимів сну в ATmega8

Після визначення з типом режиму залишається тільки вирішити його - встановити біт SE в 1. Ну і залишиться перевести в потрібний момент мікропроцесор в сон - командою asm sleep. Як приклад було написано програму для atmega8, у якій на виведення INT0 (PD2) є кнопка, а на PB0 є світлодіод, при натисканні на кнопку - світлодіод горить декілька секунд і йде в сплячий режим до наступного натискання.

Програма працює і в режимі Idle, і в режимі Power-down. Але при цьому в процесі написання і налагодження було з'ясовано, що в datasheet не просто так написано, що в режимі Power-down схема довго прокидається. Спочатку переривання було встановлено по низхідному фронту, при переході контакту INT0 з 1 на 0 (ISC01 = 1, ISC00 = 0). В режимі Idle все працює чудово, а ось в режимі Power-Down фронт змінювався до того, як мікросхема прокинеться - відповідно, ніякої обробки переривання не відбувалося. Взагалі, треба мати на увазі що в режимах, де схема довго повертається в повноцінний робочий стан, джерело переривання повинно бути активним від пробудження до повноцінного робочого стану.

**Висновки.** В результаті проведених досліджень була розроблена система яка дозволяє в режимі Idle знизити споживання до 5 мА, в режимі Power-down - до 1 мкА. Завдяки використанню енергозберігаючих режимів та переривань досягнуто значне зменшення енерговитрат в роботі мікропроцесорних систем.

**Ключові слова:** мікропроцесорна система, мікропроцесор, режим енергозбереження, переривання.