

Лабораторне заняття 4

Переривання таймера

Мета: ознайомитися з принципами роботи переривань. Отримати практичні навички по організації переривань та структури програми. Використовуючи наробки, реалізувати багатофункціональні часи.

Теоретичні відомості

Переривання (англ. Interrupt) – сигнал від програмного або апаратного забезпечення, повідомляє процесору про настання якої-небудь події, що вимагає негайної уваги. Переривання сповіщає процесор про настання високопріоритетної події, що вимагає переривання поточного коду, що виконується процесором. Процесор відповідає припиненням своєї поточної активності, зберігаючи свій стан і виконуючи функцію, звану оброблювачем переривання (або програмою обробки переривання), яка реагує на подію і обслуговує його, після чого повертає управління в перерваний код.

Arduino Leonardo має 37 переривання, а саме: переривання таймера-лічильника, АЦП, UART, зовнішні переривання, компаратора тощо.

Переривання організовані у вигляді таблиці векторів переривання:

| Vector No. | Program Address ⁽²⁾ | Source | Interrupt Definition |
|------------|--------------------------------|--------------|---|
| 1 | \$0000 ⁽¹⁾ | RESET | External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset |
| 2 | \$0002 | INT0 | External Interrupt Request 0 |
| 3 | \$0004 | INT1 | External Interrupt Request 1 |
| 4 | \$0006 | INT2 | External Interrupt Request 2 |
| 5 | \$0008 | INT3 | External Interrupt Request 3 |
| 6 | \$000A | Reserved | Reserved |
| 7 | \$000C | Reserved | Reserved |
| 8 | \$000E | INT6 | External Interrupt Request 6 |
| 9 | \$0010 | Reserved | Reserved |
| 10 | \$0012 | PCINT0 | Pin Change Interrupt Request 0 |
| 11 | \$0014 | USB General | USB General Interrupt request |
| 12 | \$0016 | USB Endpoint | USB Endpoint Interrupt request |
| 13 | \$0018 | WDT | Watchdog Time-out Interrupt |
| 14 | \$001A | Reserved | Reserved |
| 15 | \$001C | Reserved | Reserved |
| 16 | \$001E | Reserved | Reserved |
| 17 | \$0020 | TIMER1 CAPT | Timer/Counter1 Capture Event |

Рисунок 4.1 – Частина таблиці переривань

Приклад:

Лістинг 4.1 – Приклад роботи з перериванням таймеру

```
volatile short c_250ms = 0;
volatile bool flag_250ms = false;
volatile short c_400ms = 0;
volatile bool flag_400ms = false;
```

```
short led1 = 10;
short led2 = 11;
short led3 = 12;
short led4 = 13;
```

```
ISR(TIMER1_COMPA_vect) // обробчик прерывания. Вызывается каждую миллисекунду
{
    c_250ms++;
```

```

if(c_250ms == 250) { // Выполняем каждые 250 мс
    flag_250ms = true;
    c_250ms = 0;
}
c_400ms++;
if(c_400ms == 400) { // Выполняем каждые 400 мс
    flag_400ms = true;
    c_400ms = 0;
}

}

void initTimer() { // инициализация Timer1
    cli(); // отключить глобальные прерывания
    TCNT1 = 0;
    TCCR1A = 0; // установить регистр управления A в 0
    TCCR1B = 0;
    TCCR1B |= (1 << WGM12); // включить CTC режим
    TCCR1B |= (0b001 << CS10); // Установить биты на коэффициент деления

    OCR1A = 15999; // установка регистра совпадения
    TIMSK1 |= (1 << OCIE1A); // включить прерывание по совпадению таймера

    sei(); // включить глобальные прерывания
}

void setup() {
    // put your setup code here, to run once:
    pinMode(led1, OUTPUT);
    pinMode(led2, OUTPUT);
    pinMode(led3, OUTPUT);
    pinMode(led4, OUTPUT);
    digitalWrite(led1, HIGH);
    digitalWrite(led2, HIGH);
    digitalWrite(led3, HIGH);
    digitalWrite(led4, HIGH);

    initTimer();
}

void loop() {
    if(flag_250ms) { // Выполняем каждые 250 мс
        flag_250ms = false;

        digitalWrite(led4, LOW);
    }
    if(flag_400ms) { // Выполняем каждые 400 мс
        flag_400ms = false;

        digitalWrite(led4, HIGH);
    }
}

```

Зміст роботи

Завдання 1: Ввести приклад, що представлено у лістингу 4.1. Розібратися з роботою програми.

Завдання 2: Реалізувати додаток з наступними функціями:

- відображати на семисегментному індикаторі час, температуру, вологість;
- тривалість відображення часу – 15с;
- тривалість відображення температури – 5с;
- тривалість відображення вологості – 5с;
- введення с клавіатури поточного часу.