

LABORATORIUM – OPTOELEKTRONIKA

Odczyt i wyświetlenie na LCD wielkości mierzonej czujnikiem z wyjściem analogowym

Celem ćwiczenia jest praktyczna nauka programowania mikrokontrolerów poprzez oprogramowanie uC Atmega32 do celu współpracy z czujnikiem temperatury z wyjściem analogowym. Student programuje m.in. komunikację z wyświetlaczem LCD i obsługę przetwornika analogowo-cyfrowego.

WYKONANIE ĆWICZENIA

- 1.1. W środowisku **AVR Studio** otworzyć umieszczony na pulpicie projekt o nazwie **LCDtemp**.
- 1.2. Z wykorzystaniem załączonych do instrukcji not katalogowych: (wyświetlacza LCD ze sterownikiem **HD44780U**, czujnika temperatury **LM35DZ** oraz mikrokontrolera **Atmega32**) uzupełnić we wskazanych kolorami żółtym, czerwonym i niebieskim miejscach program pokazany na następnej stronie, tak by zadziałał on zgodnie z założeniami, tj. odczytał wartość napięcia z czujnika, przeliczył ją na odpowiadającą temu napięciu temperaturę i wyświetlił ją na wyświetlaczu.
 - Każdy żółty znak zapytania należy zastąpić 0 albo 1, analizując bit po bicie odpowiednie rejestry z not katalogowych, i kierując się wskazówkami zawartymi w komentarzach (zielone fragmenty programu).
 - Czerwony znak zapytania jest bitem z rejestru ADCSRA mikrokontrolera **Atmega32** (strona 216 noty uC Atmega32) – nazwę bitu należy podać wprost.
 - Niebieski znak zapytania wynika z przekształcenia wzoru z 213 strony noty uC **Atmega32**, z uwzględnieniem przelicznika Temperatura <->Napięcie noty katalogowej czujnika temperatury **LM35DZ**.

Kluczowe fragmenty not katalogowych są zakreślone.

UWAGA:

Jeżeli ktoś ma inną koncepcję modyfikacji programu (np. poprzez odwoływanie się do poszczególnych bitów rejestrów), to oczywiście może ją realizować. Liczy się efekt końcowy..

```

//-----
// program główny
int main(void)
{
    // deklaracja zmiennej do przechowywania wyniku pomiaru
    float wynik;
    // deklaracja zmiennej do przechowywania wyniku pomiaru w postaci tekstowej
    char tekst[5];

    // konfiguracja we/wy portu D - pod wyświetlacz
    DDRD = 0xFC;

    // inicjalizacja i konfiguracja wyświetlacza
    lcd_init();
    write_command(0b????????); // Function set -> interfejs 4 bity, 1 linia, znak 5x10
    - nota katalogowa HD44780U
    write_command(0b????????); // Display on/off control -> display on, kursor
    niewidoczny - nota katalogowa HD44780U

    // inicjalizacja - Port A jako wejście
    DDRA = 0x00;

    while(1) // petla nieskończona
    {
        // czyść LCD
        write_command(0b????????); // z noty katalogowej HD44780U

        // ADMUX = ADC Multiplexer Selection Register
        ADMUX = 0b????????; // VREF=2.56 V, czujnik na ADC0 -> z noty katalogowej
        Atmegi32

        // ADCSRA = ADC Control and Status Register
        ADCSRA = 0b????????; // start przetwarzania ADC, preskaler częstotliwości = 128
        -> z noty katalogowej Atmegi32

        // pomiar z ADC
        while(? == 0) //czekaj na koniec konwersji -> z noty katalogowej Atmegi32
        {
        };

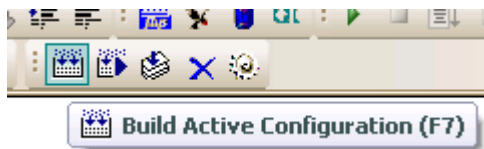
        // wynik przetwarzania jest zawsze przekazywany jako zmienna ADC:
        wynik=ADC; // pomiar = wartość z ADC

        // tekst na LCD
        write_text(" Temp: ");
        Vin=?; // przeliczenie napięcie z czujnika na temperaturę
        // funkcja dtostrf służy do konwersji zmiennej liczbowej na tekstową
        dtostrf(Vin,5,1,tekst);
        write_text(tekst);
        write_char(0b????????); // symbol "stopni" - nota katalogowa HD44780U
        write_text("C");

        _delay_ms(500); // opóźnienie = 0.5 sekundy.
    }
}

```

- 1.3. Po uzupełnieniu programu zgodnie z założeniami należy sprawdzić jego działanie. W tym celu należy go skompilować. Można to wykonać przyciskiem:



Jeżeli program skompilował się bezbłędnie należy go zgrać do mikrokontrolera. W tym celu:

- ✓ uruchomić program PonyProg,
- ✓ kolejno (rysunek poniżej):
 - ustawić rodzinę i typ używanego mikrokontrolera (1),
 - otworzyć plik programu (2), czyli plik *.hex utworzony w folderze projektu w trakcie wcześniejszej kompilacji,
 - wgrać go do mikrokontrolera (3).



- 1.4. Sprawdzić czy program działa zgodnie z założeniami (czyli na wyświetlaczu widoczna jest właściwa temperatura aktualizująca się co pół sekundy).
- 1.5. Jeśli powyższy punkt jest realizowany, program należy uzupełnić o fragment, który sprawi, że na wyświetlaczu po wartości temperatury pojawi się dodatkowo znak wskazujący, czy zmierzona temperatura jest niższa (<), wyższa (>), czy równa (=) 25°C, czyli np. dla zmierzonej temperatury otoczenia 26,2 °C na wyświetlaczu powinno się pojawić: „Temp: 26.2 °C >”. Można też dodać coś „od siebie”.

Ewentualna pomoc w punkcie 1.4:

```
if(i>5)
{
    to, co ma się stać jeśli i jest większe od 5
}
else
{
    to, co ma się stać jeśli nie
}
```

2. Literatura

- [1] Baranowski R.: *Wyświetlacze graficzne i alfanumeryczne w systemach mikroprocesorowych*, Wydawnictwo btc, Warszawa 2008
- [2] Hadam P.: *Projektowanie systemów mikroprocesorowych*, Wydawnictwo btc, Warszawa 2004
- [3] *Elektronika Praktyczna Plus - Displays*, Wydawnictwo AVT, Warszawa 2007
- [4] www.datasheetcatalog.com