



WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI I INFORMATYKI
INSTYTUT AUTOMATYKI
KIERUNEK AUTOMATYKA I ROBOTYKA
STUDIA NIESTACJONARNE I STOPNIA

LABORATORIUM - GRUPA L1

Technika Mikroprocesorowa

ĆWICZENIE 4.

Program realizujący sterowanie diodami LED poprzez komunikację UART

Wykonali:

Adam Bunzel
Olaf Karch

Prowadząca:

MGR INŻ. Andreas Kowol

TERMIN ODDANIA: : 6.06.2020

1. Opis działania programu

Napisany przez nas program w założeniu realizuje sterowanie diodami LED poprzez komunikację UART. Początkowo w terminalu wyświetla nam się komunikat odnośnie wybrania diody lub obu diod na których chcemy wykonać operację załączenia lub wyłączenia. Dioda jest wybierana poprzez podanie odpowiedniego znaku „r” dla diody czerwonej, „g” dla diody zielonej oraz „d” dla obu diod. Potwierdzenie wyboru odbywa się poprzez wciśnięcia klawisza „Enter” na naszej klawiaturze. Następnie wyświetla się komunikat odnoszący się do wyboru operacji, wyłącz(„f”) lub załącz(„t”). Gdy wpisujemy błędy znalazł lub cyfrę ujrzymy stosowny komunikat, że popełniliśmy błąd.

2. Kod programu.

```
#include <msp430.h>
#define LedRed      BIT0           // LedRed LED -> P1.0
#define LedGreen    BIT6           // LedGreen LED -> P1.6

void printChar(char c)
{
    while (!(IFG2&UCA0TXIFG)); // Check if TX is ongoing
    UCA0TXBUF = c;              // TX -> Received Char
}

void print(char *text)
{
    unsigned int i = 0;
    while(text[i] != '\0')
    {
        while (!(IFG2&UCA0TXIFG)); // Check if TX is ongoing
        UCA0TXBUF = text[i];       // TX -> Received Char + 1
        i++;
    }
}

int main(void)
{
    WDTCTL = WDTPW | WDTHOLD; // stop watchdog timer
    char ktoraDioda; // Definicja zmiennej char o nazwie "ktoraDioda"
    char offCZYon;

    if (CALBC1_1MHZ==0xFF) // Check if calibration constant erased
    {
        while(1); // do not load program
    }
}
```

```

DCOCTL = 0; // Select lowest DCO settings
BCSCTL1 = CALBC1_1MHZ; // Set DCO to 1 MHz
DCOCTL = CALDCO_1MHZ;

P1SEL = BIT1 + BIT2 ; // Select UART RX/TX function on P1.1,P1.2
P1SEL2 = BIT1 + BIT2;

UCA0CTL1 |= UCSSEL_2; // UART Clock -> SMCLK
UCA0BR0 = 104; // Baud Rate Setting for 1MHz 9600
UCA0BR1 = 0; // Baud Rate Setting for 1MHz 9600
UCA0MCTL = UCBRS_1; // Modulation Setting for 1MHz 9600
UCA0CTL1 &= ~UCSWRST; // Initialize UART Module

ADC10AE0 |= BIT5; // P1.5 ADC option select
ADC10CTL1 = INCH_5; // ADC Channel -> 1 (P1.1)
ADC10CTL0 = SREF_0 + ADC10SHT_3 + ADC10ON; // Ref -> Vcc, 64 CLK S&H

// konfiguracja pinow od LED
P1DIR |= LedRed+LedGreen; // Ustawienie ledow jako wyjscia
P1OUT &= ~LedRed; // wyŁ,Ä...cz led czerwony
P1OUT &= ~LedGreen; // wyŁ,Ä...cz led zielony

while(1)
{
    print("Wpisz na którym ledzie chcesz wykonać operacje\n\r");
    print("r= dioda czerwona, g= dioda zielona, d=obie diody na raz\n\r");

    ADC10CTL0 |= ENC + ADC10SC; // Sampling and conversion start
    while(!(UC0IFG & UCA0RXIFG)); // Wait for conversion to end
    ktoraDioda = UCA0RXBUF;
    UCA0RXBUF = 0;
    printChar(ktoraDioda);

    print("\n\r Wpisz \"t\" jesli chcesz włączyć, lub \"f\" by wyłączyć.\n\r");

    while(!(UC0IFG & UCA0RXIFG)); // Wait for conversion to end
    offCZYon = UCA0RXBUF;
    UCA0RXBUF = 0;
    printChar(offCZYon);
    print("\n\r");

    if(offCZYon != 't')
    {
        if(offCZYon != 'f')
        {
            print("Wybrano błędną operację na diodzie\n\r");
        }
    }

    switch(ktoraDioda)
    {
        case 'r':
            if(offCZYon == 't')
            {
                P1OUT &= LedRed;
            }

            if(offCZYon == 'f')
            {
                P1OUT &= ~LedRed;
            }
        . . .
    }
}

```

```
break;

case 'g':
    if(offCZYon == 't')
    {
        P1OUT &= LedGreen;
    }

    if(offCZYon == 'f')
    {
        P1OUT &= ~LedGreen;
    }

break;

case 'd':
    if(offCZYon == 't')
    {
        P1OUT &= LedGreen;
        P1OUT &= LedRed;
    }

    if(offCZYon == 'f')
    {
        P1OUT &= ~LedGreen;
        P1OUT &= ~LedRed;
    }

break;
default:
    print("Podano błędny znak\n\r");
break;

}

}
```