|  |  |
| --- | --- |
| Znalezione obrazy dla zapytania politechnika opolska | **Politechnika Opolska**  **L A B O R A T O R I U M** |

|  |  |
| --- | --- |
| Przedmiot: | **Technika Mikroprocesorowa** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Kierunek studiów: | **Automatyka i Robotyka** | | Rok studiów: | | **III** |
| Specjalność: | **-** | |  | | |
| Semestr: | **Vl** | Rok akademicki: | | ***2019/2020*** | |

|  |  |  |
| --- | --- | --- |
| *Nr ćwiczenia:* | **1** |  |

|  |
| --- |
| *Temat ćwiczenia:* |
| Zapoznanie się z układem MSP430G2 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Ćwiczenie wykonali:* | | | | | |
| *Nazwisko:* | | *Imię:* | *Nazwisko:* | | *Imię:* |
| **1.** | **Halek** | **Krzysztof** | **2.** | **Heisig** | **Henryk** |
| **3.** | **Pryszcz** | **Krzysztof** | **4.** |  |  |

|  |  |  |
| --- | --- | --- |
| *Uwagi:* | *Data:* | *Ocena za sprawozdanie:* |
|  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Termin zajęć: | | | | | |
| Data: | 7.03.2020 | Dzień tygodnia: | Sobota | Godzina: | 15:40 |

1. **Cel ćwiczenia**

Celem ćwiczenia było zapoznanie się ze środowiskiem programistycznym Code Composer Studio, które służy do tworzenia aplikacji dla wbudowanych procesorów Texas Instruments. Elementem wykonawczym programu jest mikrokontroler MSP430G2553. Program został napisany z użyciem języka programowania C.

1. **Zakres ćwiczenia**

Naszym zadaniem było napisanie programu, który ma polegać na obsłudze wejść oraz wyjść układu MSP430G2553. Program jest realizowany w nieskończonej pętli while. Polega on na zmianie kombinacji zapalonych diod LED po wciśnięciu przycisku. Każdorazowe wciśnięcie przycisku powoduje zwiększenie wartości zmiennej „a”   
o jeden. Jeżeli zmienna ta ma zdefiniowaną jakąś wartość na płytce zostają zaświecone diody o konfiguracji odpowiadającej wartości tej zmiennej. Liczba możliwych kombinacji wynosi 7. W celu poprawienia działania programu zastosowano instrukcje „for” aby nadać opóźnienie, które powoduje inkrementacje zmiennej „a” w taki sposób aby przy jednym naciśnięciu przycisku warunek był wykonywany tylko raz. Jeżeli wartość zmiennej jest większa lub równa 8 następuje zmiana jej wartości na 1.

1. **Kod programu**
2. **#include** <msp430.h>
3. **#define** SW BIT3 // definiujemy zmienną SW która

będzie

odpowiadała za Bit3. BIT3 odpowaida za fizyczny pin P1.3 na płytce prototypowej

1. **#define** RED BIT3 // Zdefiniujemy bit3 jako kolor

czerwony diody RGB

1. **#define** GRN BIT1 // Zdefiniujemy bit1 jako kolor zielony

diody RGB

1. **#define** BLU BIT5 // Zdefiniujemy bit5 jako kolor

niebieski diody RGB

1. **unsigned** **int** i,a; // rezerwujemy zmienną "i" oraz "a"jako

unsigned integer

1. **void** **main**(**void**) {
2. WDTCTL = WDTPW | WDTHOLD; // Zatrzymanie zegara watchdog'a
3. P2DIR |= (RED+GRN+BLU); // Ustawienie wyjść na pinach P2.3

(Czerwona LED), P2.1 (Zielona LED), P2.5(Niebieska LED)

1. P1DIR &= ~SW; // Ustawienie wejścia na pinie P1.3

(SW2)

1. P1REN |= SW; // włączenie rezoystora podciągającego
2. P1OUT |= SW; // ustawienie rezystora podciągającego

jako pull-up

1. a=1; // zapisanie wartości 1 dla zmiennej "a"
2. **while**(1){ // rozpoczęcie nieskończonej pętli

programu, który przy każdym naciśnięciu przycisku SW2 inkrementuje wartość zmiennej "a"

1. **if**(!(P1IN & SW)){ // wykonanie kodu jeźeli przycisk

SW2 jest naciśnięty

1. a=a+1; // inkrementacja zmiennej "a"
2. **for**(i = 0; i<10000; i++); // opóźnienie, aby przy jednym naciśnięciu

inkrementacja zmiennej "a" była wykonana jeden raz

1. }
2. **if**(a==1){ // warunek, który jest spełniony jeśli

zmienna "a" jest równa 1

1. P2OUT |= GRN;
2. P2OUT |= RED; // ustawienie wyjść dla odpowiednich pinów

do których jest podłączona dioda RGB

1. P2OUT &= ~BLU; // w tej konfiguracji zaświecony jest kolor

czerwony i zielony

1. }
2. **if**(a==2){ // warunek, który jest spełniony jeśli

zmienna "a" jest równa 2

1. P2OUT &= ~GRN;
2. P2OUT |= RED; // ustawienie wyjść dla odpowiednich pinów

do których jest podłączona dioda RGB

1. P2OUT |= BLU; // w tej konfiguracji zaświecony jest kolor

niebieski i zielony

1. }
2. **if**(a==3){ // warunek, który jest spełniony jeśli

zmienna "a" jest równa 3

1. P2OUT |= GRN;
2. P2OUT |= RED; // ustawienie wyjść dla odpowiednich pinów

do których jest podłączona dioda RGB

1. P2OUT |= BLU; // w tej konfiguracji zaświecony jest kolor

czerwony, zielony i niebieski

1. }
2. **if**(a==4){ // warunek, który jest spełniony jeśli

zmienna "a" jest równa 4

1. P2OUT |= GRN;
2. P2OUT &= ~RED; // ustawienie wyjść dla odpowiednich pinów

do których jest podłączona dioda RGB

1. P2OUT |= BLU; // w tej konfiguracji zaświecony jest kolor

zielony i niebieski

1. }
2. **if**(a==5){ // warunek, który jest spełniony jeśli

zmienna "a" jest równa 5

1. P2OUT |= GRN;
2. P2OUT &= ~RED; // ustawienie wyjść dla odpowiednich pinów

do których jest podłączona dioda RGB

1. P2OUT &= ~BLU; // w tej konfiguracji zaświecony jest kolor

zielony

1. }
2. **if**(a==6){ // warunek, który jest spełniony jeśli

zmienna "a" jest równa 6

1. P2OUT &= ~GRN;
2. P2OUT |= RED; // ustawienie wyjść dla odpowiednich

pinów do których jest podłączona dioda RGB

1. P2OUT &= ~BLU; // w tej konfiguracji zaświecony jest

Kolor czerwony

1. }
2. **if**(a==7){ // warunek, który jest spełniony jeśli

zmienna "a" jest równa 7

1. P2OUT &= ~GRN;
2. P2OUT &= ~RED; // ustawienie wyjść dla odpowiednich

pinów

do których jest podłączona dioda RGB

1. P2OUT |= BLU; // w tej konfiguracji zaświecony jest

kolor niebieski

1. }
2. **if**(a>=8){ // Jeśli zmienna "a" jest większa lub

równa 8

1. a = 1; // następuje ustawienie wartości zmiennej

"a" na 1

1. }
2. }
3. **Wnioski**

Za pomocą środowiska Code Composer Studio można z łatwością nawiązać komunikację z mikroprocesorem MSP430G2553 i wgrać programy napisane  
w języku „C”. Napisany pierwszy program do obsługi GPIO nie sprawił problemu,  
a uruchomiony kod działa poprawnie. Dzięki wbudowanym rezystorom podciągającym „Pull Up” można było skonfigurować przycisk jako sygnał wejściowy. Skonfigurowanie odpowiednich pinów jako wyjście pozwalało na zapalanie odpowiednich kolorów wbudowanej diody RGB. Natknęliśmy się jednak na przeszkodę typu zbyt szybką inkrementację zmiennej „a” przez co dołożyliśmy opóźnienie zrealizowanie na pętli „for”.