

Arkadiusz Kulesza

Sprawozdanie z ćwiczeń

Systemy Wbudowane

Osoby , które pomogły mi przy zadaniach :

Jacek Gross -osoba , z którą pracowałem w duecie .Wspólnie opracowaliśmy kod do zadania pierwszego .

Wojtek Pakoca - pomógł w zaimplementowaniu kodu do wyświetlania własnych znaków na wyświetlaczu lcd .

Zadanie 1

Treść zadania:

Program przetwarzający cyklicznie 9 podprogramów (następny program po 9 to 1, a poprzedni względem 1 to 9)

Przycisk RD13 – następny program

Przycisk RD6 – poprzedni program

Podprogramy:

1. 8 bitowy licznik binarny zliczający w górę (0...255)
2. 8 bitowy licznik binarny zliczający w dół (255...0)
3. 8 bitowy licznik w kodzie Graya zliczający w górę (repr. 0...255)
4. 8 bitowy licznik w kodzie Graya zliczający w dół (repr. 255...0)
5. 2x4 bitowy licznik w kodzie BCD zliczający w górę (0...99)
6. 2x4 bitowy licznik w kodzie BCD zliczający w dół (99...0)
7. 3 bitowy wężyk poruszający się lewo-prawo
8. Kolejka
9. 6 bitowy generator liczb pseudolosowych oparty o konfigurację 1110011

problemy podczas implementacji:

Najwięcej problemu sprawiło mi w pierwszym zadaniu zapoznanie się z działaniem mikrokontrolera. Następnym problemem ,który napotkałem było przypomnienie działań na bitach i ich implementacja w kodzie.

Kod zadania: plik newmain1.c w folderze kod_źródłowy

Źródła pomocne przy rozwiązywaniu zadań: Wikipedia, materiały z ćwiczeń i wykładów

Zadanie 2

Treść zadania:

W oparciu o materiał *potencjometr3.c*

Alarm!

Jako źródło danych należy wziąć potencjometr P1, nastawę – wartość po przekroczeniu której alarm załącza się – ustawić na połowę zakresu potencjometru.

Po przekroczeniu wartości nastawy – przez pierwsze 5 sekund mruga jedna dioda, a potem zapalone są wszystkie.

Jeśli alarm jest załączony (mruganie, zapalone wszystkie diody) – to po ustawieniu potencjometru poniżej nastawy alarmowej – alarm należy wyłączyć (przerwać mruganie, zgasić diody)

Wyłączenie alarmu – przycisk RB3

problemy podczas implementacji:

Najwięcej problemu sprawił czujnik ,którego odczyty oscylowały wokół danej wartości przez co ciężko było sterować diodami.

Kod zadania: plik newmain2.c w folderze kod_źródłowy

Źródła pomocne przy rozwiązywaniu zadań: Wikipedia, materiały z ćwiczeń i wykładów

Zadanie 3

Treść zadania:

W oparciu o plik *elsidi.c* oraz <http://wmii.uwm.edu.pl/~zajac/sw/hd44780.png>

Wszelkie komunikaty z zadań powinny być wyświetlone na wyświetlaczu LCD 16x2

Reklama!

Atrakcyjna, - temat dowolny

problemy podczas implementacji:

Największym problemem okazało się zaimplementowanie własnego znaczka na wyświetlaczu.

Kod zadania: plik newmain3.c w folderze kod_źródłowy

Źródła pomocne przy rozwiązywaniu zadań: Wikipedia, materiały z ćwiczeń i wykładów

Zadanie 4

Treść zadania:

W oparciu o plik picsimlab/elsidi.c oraz <http://wmii.uwm.edu.pl/~zajac/sw/hd44780.png>

Wszelkie komunikaty z zadań powinny być wyświetlone na wyświetlaczu LCD 16x2

Kontroler kuchenki mikrofalowej.

RB5 – wybór mocy 800W, 600W, 350W, 200W

RB4 – dodanie czasu 1min

RB3 – dodanie czasu 10s

RB2 – Start/Stop

RB1 – Reset

problemy podczas implementacji:

W danym zadaniu nie miałem żadnych problemów z względu na wcześniejsze zadania , które przygotowały mnie do napisania kodu w tym zadaniu .

Kod zadania: plik newmain4.c w folderze kod_źródłowy

Źródła pomocne przy rozwiązywaniu zadań: materiały z ćwiczeń i wykładów