SYSTEMY WBUDOWANE

Sprawozdanie do zadań 1-5

Zadanie 3, 4 zostały wykonane samodzielnie.

Zadania 1, 2, 5 zostały wykonane zespołowo przez: Sebastian Celoch, Michał Chomczyk, Tymoteusz Łuczko, Konrad Banul

Spis treści

[Zadanie 1 2](#_Toc136099160)

[Opis zadania: 2](#_Toc136099161)

[Problemy podczas implementacji: 2](#_Toc136099162)

[Kod: 2](#_Toc136099163)

[Zadanie 2 7](#_Toc136099164)

[Opis zadania: 7](#_Toc136099165)

[Problemy podczas implementacji: 7](#_Toc136099166)

[Kod: 7](#_Toc136099167)

[Zadanie 3 12](#_Toc136099168)

[Opis zadania: 12](#_Toc136099169)

[Problemy podczas implementacji: 12](#_Toc136099170)

[Kod: 12](#_Toc136099171)

[Zadanie 4 19](#_Toc136099172)

[Opis zadania: 19](#_Toc136099173)

[Problemy podczas implementacji: 19](#_Toc136099174)

[Kod: 19](#_Toc136099175)

[Zadanie 5 27](#_Toc136099176)

[Opis zadania: 27](#_Toc136099177)

[Problemy podczas implementacji: 28](#_Toc136099178)

[Kod: 28](#_Toc136099179)

# Zadanie 1

## Opis zadania:

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

Przycisk S3–następny programPrzyciskS6–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-prawo8.Kolejka

9.6 bitowy generator liczb pseudolosowych oparty o konfigurację 1110011

## Problemy podczas implementacji:

Największy problem sprawiło zaimplementowanie 5, 6, 7, 8 zadania

## Kod:

#define FCY 4000000ULL

#include <stdio.h>

#include <stdlib.h>

#include <stddef.h>

#include <stdbool.h>

#include <libpic30.h>

#include "app.h"

#define ONE\_VOLT 310

#define ONE\_TENTH\_VOLT 31

#define ONE\_HUNDREDTH\_VOLT 3

void zad1**(** void **);**

void zad2**(** void **);**

void zad3**(** void **);**

void zad4**(** void **);**

void zad6**(** void **);**

void zad7**(** void **);**

void zad8**(** void **);**

void zad9**(** void **);**

void r\_program**(** int prog **);**

APP\_DATA appData **=** **{**

**.**messageLine1 **=** "Explorer 16 Demo" **,**

**.**messageLine2 **=** "Press S3 to cont" **,**

**.**messageTime **=** "Time 00: 00: 00 " **,**

**.**messageADC **=** " Pot = 0.00 Vdc "

**}** **;**

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Section: Main Entry Point

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int main **(** void **)**

**{**

SYS\_Initialize**();**

LED\_Enable**(**LED\_D3**);**

LED\_Enable**(**LED\_D4**);**

LED\_Enable**(**LED\_D5**);**

LED\_Enable**(**LED\_D6**);**

LED\_Enable**(**LED\_D7**);**

LED\_Enable**(**LED\_D8**);**

LED\_Enable**(**LED\_D9**);**

LED\_Enable**(**LED\_D10**);**

int prog **=** 1**;**

/\* Infinite Loop \*/

**while** **(** 1 **)**

**{**

delay**(**500**);**

**if(**BUTTON\_IsPressed**(**BUTTON\_S3**))**

**{**

prog**++;**

**}**

**else** **if** **(**BUTTON\_IsPressed**(**BUTTON\_S6**))**

**{**

prog**--;**

**}**

**if(**prog **>** 9**)**

**{**

prog **=** 1**;**

**}**

**else** **if** **(**prog **<** 1**)**

**{**

prog **=** 9**;**

**}**

LCD\_ClearScreen**();**

char txt**[**16**];**

sprintf**(**txt**,** "Zadanie %d"**,** prog**);**

LCD\_PutString**(**txt**,** **sizeof(**txt**)-**1**);**

r\_program**(**prog**);**

LATA **=** 0**;**

//LED\_On(LED\_D10);

delay**(**200**);**

**}** **;**

**}**

void delay**(**unsigned int ms**)**

**{**

unsigned int i**;**

unsigned char j**;**

**for** **(**i **=** 0**;** i **<** ms**;** i**++)** **{**

**for** **(**j **=** 0**;** j **<** 200**;** j**++)** **{**

Nop**();**

Nop**();**

Nop**();**

Nop**();**

Nop**();**

**}**

**}**

**}**

void zad1**(**void**)**

**{**

**for** **(**int i **=** 0**;**i**<=**256**;**i**++)**

**{**

LATA**=**i**;**

delay**(**500**);**

**if(**BUTTON\_IsPressed**(**BUTTON\_S3**)** **||** BUTTON\_IsPressed**(**BUTTON\_S6**))**

**{**

**break;**

**}**

**}**

**}**

void zad2**(**void**){**

**for** **(**int i**=**256**;** i**>**0**;** i**--)**

**{**

LATA **=** i**;**

delay**(**500**);**

**if(**BUTTON\_IsPressed**(**BUTTON\_S3**)** **||** BUTTON\_IsPressed**(**BUTTON\_S6**))**

**{**

**break;**

**}**

**}**

**}**

void zad3**(**void**)**

**{**

**for** **(**int i**=**0**;** i**<**256**;** i**++)**

**{**

LATA **=** i **^** **(**i **>>** 1**);;**

delay**(**500**);**

**if(**BUTTON\_IsPressed**(**BUTTON\_S3**)** **||** BUTTON\_IsPressed**(**BUTTON\_S6**))**

**{**

**break;**

**}**

**}**

**}**

void zad4**(**void**)**

**{**

**for** **(**int i**=**256**;** i**>**0**;** i**--)**

**{**

LATA **=** i **^** **(**i **>>** 1**);;**

delay**(**500**);**

**if(**BUTTON\_IsPressed**(**BUTTON\_S3**)** **||** BUTTON\_IsPressed**(**BUTTON\_S6**))**

**{**

**break;**

**}**

**}**

**}**

void zad5**(**void**){**

**for** **(**int i**=**0**;** i**<**145**;** i**+=**16**){**

**for** **(**int j**=**0**;** j**<**10**;** j**++)** **{**

LATA **=** i **+** j**;**

delay**(**500**);**

**if(**BUTTON\_IsPressed**(**BUTTON\_S3**)** **||** BUTTON\_IsPressed**(**BUTTON\_S6**))**

**{**

**break;**

**}**

**}**

**}**

**}**

void zad6**(**void**){**

**for** **(**int i**=**144**;** i**>=**0**;** i**-=**16**){**

**for** **(**int j**=**9**;** j**>=**0**;** j**--)** **{**

LATA **=** i **+** j**;**

delay**(**500**);**

**if(**BUTTON\_IsPressed**(**BUTTON\_S3**)** **||** BUTTON\_IsPressed**(**BUTTON\_S6**))**

**{**

**break;**

**}**

**}**

**}**

**}**

void zad7**(**void**)** **{**

int counter **=** 7**;**

**while** **(**1**)** **{**

**for** **(**int i **=** 0**;** i **<** 5**;** i**++)** **{**

LATA **=** counter**;**

counter **<<=** 1**;**

delay**(**500**);**

**}**

**for** **(**int j **=** 0**;** j **<** 5**;** j**++)** **{**

LATA **=** counter**;**

counter **>>=** 1**;**

delay**(**500**);**

**}**

**if(**BUTTON\_IsPressed**(**BUTTON\_S3**)** **||** BUTTON\_IsPressed**(**BUTTON\_S6**))** **{**

**break;**

**}**

**}**

**}**

void zad8**(**void**)**

**{**

int x **=** 7**;**

int bit8 **=** 1**;**

int bit8\_2 **=** 0**;**

**while(**1**){**

delay**(**300**);**

**for(**int i **=** 0**;** i **<** x**;** i**++){**

delay**(**300**);**

LATA **=** bit8 **+** bit8\_2**;**

bit8 **<<=** 1**;**

**}**

bit8\_2 **=** bit8\_2 **+** bit8**;**

delay**(**300**);**

LATA **=** bit8\_2**;**

bit8 **=** 1**;**

x**--;**

**if(**x **==** **-**1**){**

x **=** 7**;**

bit8\_2 **=** 0**;**

**}**

**if(**BUTTON\_IsPressed**(**BUTTON\_S3**)** **||** BUTTON\_IsPressed**(**BUTTON\_S6**))**

**{**

**break;**

**}**

**}**

**}**

void zad9**()**

**{**

**while(**1**){**

delay**(**400**);**

int i**,** j**;**

int num**[**64**];**

// tablic? od 0 do 63 (6-bit))

**for** **(**i **=** 0**;** i **<** 64**;** i**++)** **{**

num**[**i**]** **=** i**;**

**}**

srand**(**TMR1**);**

**for** **(**i **=** 63**;** i **>** 0**;** i**--)** **{**

j **=** rand**()** **%** **(**i **+** 1**);**

int temp **=** num**[**i**];**

num**[**i**]** **=** num**[**j**];**

num**[**j**]** **=** temp**;**

**}**

**for** **(**i **=** 0**;** i **<** 64**;** i**++)** **{**

LATA **=** num**[**i**];**

delay**(**400**);**

**}**

**if(**BUTTON\_IsPressed**(**BUTTON\_S3**)** **||** BUTTON\_IsPressed**(**BUTTON\_S6**))**

**{**

**break;**

**}**

**}**

**}**

void r\_program**(**int prog**)**

**{**

**switch(**prog**)**

**{**

**case** 1**:**

zad1**();**

**break;**

**case** 2**:**

zad2**();**

**break;**

**case** 3**:**

zad3**();**

**break;**

**case** 4**:**

zad4**();**

**break;**

**case** 5**:**

zad5**();**

**break;**

**case** 6**:**

zad6**();**

**break;**

**case** 7**:**

zad7**();**

**break;**

**case** 8**:**

zad8**();**

**break;**

**case** 9**:**

zad9**();**

**break;**

**default:**

zad1**();**

**break;**

**}**

**}**

# Zadanie 2

## Opis zadania:

W oparciu o program przykładowy oraz dokumentację czujnika temperatury TC1047A:

Alarm!

Jako źródło danych należy wziąć potencjometr czujnik temperatury TC 1047A, nastawę – wartość po

przekroczeniu której alarm załącza się – ustawić na temperaturę 25 stopni.

Po przekroczeniu wartości nastawy – przez pierwsze 3 sekundy mruga jedna dioda, a potem zapalone

są wszystkie.

Jeśli alarm jest załączony (mruganie, zapalone wszystkie diody) – to po spadku temperatury poniżej

nastawy alarmowej – alarm należy wyłączyć (przerwać mruganie, zgasić diody)

Wyłączenie alarmu – przycisk S6

## Problemy podczas implementacji:

Problem pojawił się w zmianie wartości potencjometru na stopnie Celsjusza.

## Kod:

#include <stdio.h>

#include <stdlib.h>

#include <stddef.h>

#include <stdbool.h>

#include "lcd.h";

#define FCY 4000000ULL

#include <libpic30.h>

#include "app.h"

#define ONE\_VOLT 310

#define ONE\_TENTH\_VOLT 31

#define ONE\_HUNDREDTH\_VOLT 3

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Section: File Scope Variables and Functions

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Update\_LCD **(** void **)** **;**

void SYS\_Initialize **(** void **)** **;**

extern void ConvertADCVoltage **(** unsigned int **)** **;**

extern void Hex2Dec **(** unsigned char **)** **;**

double temp **=** 0**;**

double voltage**;**

APP\_DATA appData **=** **{**

**.**messageLine1 **=** "Explorer 16 Demo" **,**

**.**messageLine2 **=** "Press S3 to cont" **,**

**.**messageTime **=** "Time 00: 00: 00 " **,**

**.**messageADC **=** " Pot = 0.00 Vdc "

**}** **;**

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Section: Main Entry Point

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void set\_alarm**(){**

**for(**int i **=** 0**;** i **<=** 3**;** i**++)**

**{**

i**++;**

LATA **=** 128**;**

\_\_delay\_ms**(**500**);**

LATA **=** 0**;**

\_\_delay\_ms**(**500**);**

**}**

LATA **=** 255**;**

**}**

int main **(** void **)**

**{**

LCD\_PutString **(** **(**char**\*)** **&**appData**.**messageLine1**[**0**]** **,** **sizeof** **(**appData**.**messageLine1 **)** **-** 1 **)** **;**

LCD\_PutString **(** **(**char**\*)** **&**appData**.**messageLine2**[**0**]** **,** **sizeof** **(**appData**.**messageLine2 **)** **-** 1 **)** **;**

LCD\_PutString **(** **(**char**\*)** **&**appData**.**messageTime**[**0**]** **,** **sizeof** **(**appData**.**messageTime **)** **-** 1 **)** **;**

LCD\_PutString **(** **(**char**\*)** **&**appData**.**messageADC**[**0**]** **,** **sizeof** **(**appData**.**messageADC **)** **-** 1 **)** **;**

SYS\_Initialize **(** **)** **;**

TIMER\_SetConfiguration **(** TIMER\_CONFIGURATION\_RTCC **)** **;**

ADC\_SetConfiguration **(** ADC\_CONFIGURATION\_AUTO\_SAMPLE\_CONVERT **)** **;**

int start **=** 0**;**

int zmiana **=** 0**;**

**while** **(** 1 **)**

**{**

**if(**BUTTON\_IsPressed **(** BUTTON\_S6 **))**

start **=** 0**;**

**if** **(** start **==** 0 **)**

**{**

**if(**BUTTON\_IsPressed **(** BUTTON\_S6 **))**

**{**

start **=** 1**;**

**break;**

**}**

appData**.**temp1**=**ADC\_Read10bit**(**ADC\_CHANNEL\_TEMPERATURE\_SENSOR**);**

ConvertADCVoltage**(**appData**.**temp1**);**

temp**=(((**appData**.**adones**-**'0'**)\***0**+(**appData**.**adtens**-**'0'**)\***0.1**+(**appData**.**adhunds**-**'0'**)\***0.01**)-**0.1**)\***100**-**46**;**

char a**[**20**];**

**if(**temp **>=** 28**)**

**{**

LCD\_PutString**(**"Alarm ON "**,**16**);**

sprintf**(**a**,**"Temp: %0.1f"**,**temp**);**

LCD\_PutString**(**a**,**10**);**

**if(**zmiana **==** 0**)**

**{**

set\_alarm**();**

zmiana **=** 1**;**

**}**

**}**

**else** **if(**temp **<** 28**)**

**{**

sprintf**(**a**,**"Temp: %0.1f"**,**temp**);**

LCD\_PutString**(**a**,**10**);**

LATA **=** 0**;**

zmiana **=** 0**;**

**}**

\_\_delay\_ms**(**500**);**

LCD\_ClearScreen**();**

**}**

**}** **;**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Update\_LCD **(** void **)**

**{**

LCD\_PutChar **(** 'T' **)** **;**

LCD\_PutChar **(** 'i' **)** **;**

LCD\_PutChar **(** 'm' **)** **;**

LCD\_PutChar **(** 'e' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** appData**.**tens **+** 0x30 **)** **;**

LCD\_PutChar **(** appData**.**ones **+** 0x30 **)** **;**

Hex2Dec **(** appData**.**minutes **)** **;**

LCD\_PutChar **(** ':' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** appData**.**tens **+** 0x30 **)** **;**

LCD\_PutChar **(** appData**.**ones **+** 0x30 **)** **;**

Hex2Dec **(** appData**.**seconds **)** **;**

LCD\_PutChar **(** ':' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** appData**.**tens **+** 0x30 **)** **;**

LCD\_PutChar **(** appData**.**ones **+** 0x30 **)** **;**

**}**

void ConvertADCVoltage **(** unsigned int adc\_conv\_data **)**

**{**

/\* reset values \*/

appData**.**adones **=** 0 **;**

appData**.**adtens **=** 0 **;**

appData**.**adhunds **=** 0 **;**

appData**.**adthous **=** 0 **;**

**while** **(** adc\_conv\_data **>** 0 **)**

**{**

/\* test for 1 volt or greater \*/

**if** **(** adc\_conv\_data **>** **(** ONE\_VOLT **-** 1 **)** **)**

**{**

/\* increment 1 volt counter \*/

appData**.**adones**++** **;**

/\* subtract 1 volt \*/

adc\_conv\_data **-=** ONE\_VOLT **;**

**}**

/\* test for 0.1 volt \*/

**else** **if** **(** adc\_conv\_data **>** **(** ONE\_TENTH\_VOLT **-** 1 **)** **)**

**{**

/\* increment tenths \*/

**if** **(** appData**.**adtens **<** 9 **)**

**{**

appData**.**adtens**++** **;**

**}**

**else**

**{**

/\* tenths has rolled over \*/

appData**.**adones**++** **;**

/\* so increment ones and reset tenths \*/

appData**.**adtens **=** 0 **;**

**}**

adc\_conv\_data **-=** ONE\_TENTH\_VOLT **;**

**}**

/\* test for 0.01 volt \*/

**else** **if** **(** adc\_conv\_data **>** **(** ONE\_HUNDREDTH\_VOLT **-** 1 **)** **)**

**{**

/\* increment hundredths \*/

**if** **(** appData**.**adhunds **<** 9 **)**

**{**

appData**.**adhunds**++** **;**

**}**

**else**

**{**

/\* hundredths has rolled over \*/

appData**.**adtens**++** **;**

/\* so increment tenths and reset hundredths \*/

appData**.**adhunds **=** 0 **;**

**}**

adc\_conv\_data **-=** ONE\_HUNDREDTH\_VOLT **;**

**}**

**else** **if** **(** adc\_conv\_data **<=** **(** ONE\_HUNDREDTH\_VOLT **-** 1 **)** **)**

**{**

appData**.**adthous**++** **;**

adc\_conv\_data **--** **;**

**}**

**}**

appData**.**adones **+=** 0x30 **;**

appData**.**adtens **+=** 0x30 **;**

appData**.**adhunds **+=** 0x30 **;**

appData**.**adthous **+=** 0x30 **;**

**}**

void Hex2Dec **(** unsigned char count **)**

**{**

/\* reset values \*/

appData**.**hunds **=** 0 **;**

appData**.**tens **=** 0 **;**

appData**.**ones **=** 0 **;**

**while** **(** count **>=** 10 **)**

**{**

**if** **(** count **>=** 200 **)**

**{**

count **-=** 200 **;**

appData**.**hunds **=** 0x02 **;**

**}**

**if** **(**count **>=** 100**)**

**{**

count **-=** 100 **;**

appData**.**hunds**++** **;**

**}**

**if** **(**count **>=** 10 **)**

**{**

count **-=** 10 **;**

appData**.**tens**++** **;**

**}**

**}**

appData**.**ones **=** count **;**

**}**

# Zadanie 3

## Opis zadania:

W oparciu o program przykładowy dla Explorer 16/32 oraz PIC24FJ128GA010.

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

Reklama!

Atrakcyjna, - temat dowolny

## Problemy podczas implementacji:

Jedyny problem napotkany podczas implementacji to pomysł na reklamę.

## Kod:

#include <stdio.h>

#include <stdlib.h>

#include <stddef.h>

#include <stdbool.h>

#include "app.h"

#define ONE\_VOLT 310

#define ONE\_TENTH\_VOLT 31

#define ONE\_HUNDREDTH\_VOLT 3

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Section: File Scope Variables and Functions

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Update\_LCD **(** void **)** **;**

void SYS\_Initialize **(** void **)** **;**

extern void ConvertADCVoltage **(** unsigned int **)** **;**

extern void Hex2Dec **(** unsigned char **)** **;**

APP\_DATA appData **=** **{**

**.**messageLine1 **=** "Explorer 16 Demo" **,**

**.**messageLine2 **=** "Press S3 to cont" **,**

**.**messageTime **=** "Time 00: 00: 00 " **,**

**.**messageADC **=** " Pot = 0.00 Vdc "

**}** **;**

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Section: Main Entry Point

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int main **(** void **)**

**{**

SYS\_Initialize**();**

//https://openlabpro.com/guide/custom-character-lcd-pic/?fbclid=IwAR0ZAMS0KPyWX2emjHpkgujIbLyBNNELObAcFTSeyhAVIXfTCTDZ2cxfwek

//serduszko ale nie wiem czy wyjdzie na zajeciach tescik

//unsigned char Pattern3[]= {0x00,0x00,0x0a,0x15,0x11,0x0a,0x04,0x00};

//unsigned char Pattern4[]= {0x00,0x00,0x0a,0x1f,0x1f,0x0e,0x04,0x00};

char pom**[**16**]=**" "**;**

char tablica**[**122**][**17**]={**

" "**,**

"P I Z Z A ! "**,**

" P I Z Z A ! "**,**

" P I Z Z A ! "**,**

" PIZZA! "**,**

" "**,**

" P"**,**

" P "**,**

" P I"**,**

" P I "**,**

" P I Z"**,**

" P I Z "**,**

" P I Z Z"**,**

" P I Z Z "**,**

" P I Z Z A"**,**

" P I Z Z A "**,**

" P I Z Z A !"**,**

" P I Z Z A ! "**,**

" P I Z Z A ! "**,**

"P I Z Z A ! "**,**

" P I Z Z A ! "**,**

" P I Z Z A ! "**,**

" PIZZA! "**,**

" "**,**

"K E B A B ! "**,**

" K E B A B ! "**,**

" K E B A B ! "**,**

" KEBAB! "**,**

" "**,**

" K"**,**

" K "**,**

" K E"**,**

" K E "**,**

" K E B"**,**

" K E B "**,**

" K E B A"**,**

" K E B A "**,**

" K E B A B"**,**

" K E B A B "**,**

" K E B A B !"**,**

" K E B A B ! "**,**

" K E B A B ! "**,**

"K E B A B ! "**,**

" K E B A B ! "**,**

" K E B A B ! "**,**

" KEBAB! "**,**

" "**,**

"F R Y T K I "**,**

" F R Y T K I "**,**

" F R Y T K I "**,**

" FRYTKI "**,**

" "**,**

" F"**,**

" F "**,**

" F R"**,**

" F R "**,**

" F R Y"**,**

" F R Y "**,**

" F R Y T"**,**

" F R Y T "**,**

" F R Y T K"**,**

" F R Y T K "**,**

" F R Y T K I"**,**

" F R Y T K I "**,**

" F R Y T K I "**,**

"F R Y T K I "**,**

" F R Y T K I "**,**

" F R Y T K I "**,**

" FRYTKI "**,**

" "**,**

" Z"**,**

" Z "**,**

" Z A"**,**

" ZA "**,**

" ZA P"**,**

" ZAP "**,**

" ZAP R"**,**

" ZAPR "**,**

" ZAPR A"**,**

" ZAPRA "**,**

" ZAPRA S"**,**

" ZAPRAS "**,**

" ZAPRAS Z"**,**

" ZAPRASZ "**,**

" ZAPRASZ A"**,**

" ZAPRASZA "**,**

" ZAPRASZA M"**,**

" ZAPRASZAM "**,**

" ZAPRASZAM Y"**,**

" ZAPRASZAMY "**,**

" ZAPRASZAMY !"**,**

" ZAPRASZAMY! "**,**

" "**,**

" ZAPRASZAMY! "**,**

" "**,**

" ZAPRASZAMY! "**,**

" "**,**

" ZAPRASZAMY! "**,**

" "**,**

" ZAPRASZAMY! "**,**

" ZAPRASZAMY! "**,**

" ZAPRASZAMY! "**,**

"ZAPRASZAMY! "**,**

"APRASZAMY! "**,**

"PRASZAMY! "**,**

"RASZAMY! "**,**

"ASZAMY! "**,**

"SZAMY! "**,**

"ZAMY! "**,**

"AMY! "**,**

"MY! "**,**

"Y! "**,**

"! "**,**

" ! ! ! ! ! "**,**

" "**,**

" ! ! ! ! ! ! ! !"**,**

" "**,**

" ! ! ! ! ! "**,**

" "**,**

" ! ! ! ! ! ! ! !"**,**

" "**,**

" ! ! ! ! ! ! ! !"**,**

**};**

/\* Infinite Loop \*/

**while** **(** 1 **)**

**{**

**for(**int i**=**0**;**i**<**122**;**i**++)**

**{**

LCD\_PutString **(** tablica**[**i**]** **,** **sizeof(**tablica**)** **)** **;**

LCD\_PutString**(**pom**,** **sizeof(**pom**));**

delay**(**200**);**

LCD\_ClearScreen**();**

**}**

**}** **;**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function:

void Update\_LCD( void )

Summary:

Function to update LCD

Description:

This function will update the time on the LCD

Precondition:

None.

Parameters:

None.

Returns:

None.

Remarks:

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void delay**(**unsigned int ms**)**

**{**

unsigned int i**;**

unsigned char j**;**

**for** **(**i **=** 0**;** i **<** ms**;** i**++)** **{**

**for** **(**j **=** 0**;** j **<** 200**;** j**++)** **{**

Nop**();**

Nop**();**

Nop**();**

Nop**();**

Nop**();**

**}**

**}**

**}**

void Update\_LCD **(** void **)**

**{**

LCD\_PutChar **(** 'T' **)** **;**

LCD\_PutChar **(** 'i' **)** **;**

LCD\_PutChar **(** 'm' **)** **;**

LCD\_PutChar **(** 'e' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** appData**.**tens **+** 0x30 **)** **;**

LCD\_PutChar **(** appData**.**ones **+** 0x30 **)** **;**

Hex2Dec **(** appData**.**minutes **)** **;**

LCD\_PutChar **(** ':' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** appData**.**tens **+** 0x30 **)** **;**

LCD\_PutChar **(** appData**.**ones **+** 0x30 **)** **;**

Hex2Dec **(** appData**.**seconds **)** **;**

LCD\_PutChar **(** ':' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** appData**.**tens **+** 0x30 **)** **;**

LCD\_PutChar **(** appData**.**ones **+** 0x30 **)** **;**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function:

void ConvertADCVoltage ( unsigned int )

Summary:

Function to convert ADC data into volts

Description:

This is the file for the Explorer 16 Dem that converts raw 10 bit ADC data

to volts suitable for the LCD display

Precondition:

None.

Parameters:

None.

Returns:

None.

Remarks:

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void ConvertADCVoltage **(** unsigned int adc\_conv\_data **)**

**{**

/\* reset values \*/

appData**.**adones **=** 0 **;**

appData**.**adtens **=** 0 **;**

appData**.**adhunds **=** 0 **;**

appData**.**adthous **=** 0 **;**

**while** **(** adc\_conv\_data **>** 0 **)**

**{**

/\* test for 1 volt or greater \*/

**if** **(** adc\_conv\_data **>** **(** ONE\_VOLT **-** 1 **)** **)**

**{**

/\* increment 1 volt counter \*/

appData**.**adones**++** **;**

/\* subtract 1 volt \*/

adc\_conv\_data **-=** ONE\_VOLT **;**

**}**

/\* test for 0.1 volt \*/

**else** **if** **(** adc\_conv\_data **>** **(** ONE\_TENTH\_VOLT **-** 1 **)** **)**

**{**

/\* increment tenths \*/

**if** **(** appData**.**adtens **<** 9 **)**

**{**

appData**.**adtens**++** **;**

**}**

**else**

**{**

/\* tenths has rolled over \*/

appData**.**adones**++** **;**

/\* so increment ones and reset tenths \*/

appData**.**adtens **=** 0 **;**

**}**

adc\_conv\_data **-=** ONE\_TENTH\_VOLT **;**

**}**

/\* test for 0.01 volt \*/

**else** **if** **(** adc\_conv\_data **>** **(** ONE\_HUNDREDTH\_VOLT **-** 1 **)** **)**

**{**

/\* increment hundredths \*/

**if** **(** appData**.**adhunds **<** 9 **)**

**{**

appData**.**adhunds**++** **;**

**}**

**else**

**{**

/\* hundredths has rolled over \*/

appData**.**adtens**++** **;**

/\* so increment tenths and reset hundredths \*/

appData**.**adhunds **=** 0 **;**

**}**

adc\_conv\_data **-=** ONE\_HUNDREDTH\_VOLT **;**

**}**

**else** **if** **(** adc\_conv\_data **<=** **(** ONE\_HUNDREDTH\_VOLT **-** 1 **)** **)**

**{**

appData**.**adthous**++** **;**

adc\_conv\_data **--** **;**

**}**

**}**

appData**.**adones **+=** 0x30 **;**

appData**.**adtens **+=** 0x30 **;**

appData**.**adhunds **+=** 0x30 **;**

appData**.**adthous **+=** 0x30 **;**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function:

void Hex2Dec ( unsigned char )

Summary:

Explorer 16 Demo Hex to Decimal Conversion File

Description:

This is the file for the Explorer 16 Dem that converts the hexadecimal data

into decimal format.

Precondition:

None.

Parameters:

None.

Returns:

None.

Remarks:

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Hex2Dec **(** unsigned char count **)**

**{**

/\* reset values \*/

appData**.**hunds **=** 0 **;**

appData**.**tens **=** 0 **;**

appData**.**ones **=** 0 **;**

**while** **(** count **>=** 10 **)**

**{**

**if** **(** count **>=** 200 **)**

**{**

count **-=** 200 **;**

appData**.**hunds **=** 0x02 **;**

**}**

**if** **(**count **>=** 100**)**

**{**

count **-=** 100 **;**

appData**.**hunds**++** **;**

**}**

**if** **(**count **>=** 10 **)**

**{**

count **-=** 10 **;**

appData**.**tens**++** **;**

**}**

**}**

appData**.**ones **=** count **;**

**}**

# Zadanie 4

## Opis zadania:

W oparciu o program przykładowy dla Explorer 16/32 oraz PIC24FJ128GA010.

Wszelkie komunikaty z zadania powinny być wyświetlone na wyświetlaczu LCD 16x2.

Kontroler kuchenki mikrofalowej.

S3 – wybór mocy 800W, 600W, 350W, 200W

S6 – dodanie czasu 30 s

S5 – Start/Stop

S4 – Reset

## Problemy podczas implementacji:

Przycisk S5 włączał się samemu, trzeba było zakomentować 4 linijki kodu w system.c xD.

## Kod:

#include <stdio.h>

#include <stdlib.h>

#include <stddef.h>

#include <stdbool.h>

#include "app.h"

#define ONE\_VOLT 310

#define ONE\_TENTH\_VOLT 31

#define ONE\_HUNDREDTH\_VOLT 3

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Section: File Scope Variables and Functions

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Update\_LCD **(** void **)** **;**

void SYS\_Initialize **(** void **)** **;**

extern void ConvertADCVoltage **(** unsigned int **)** **;**

extern void Hex2Dec **(** unsigned char **)** **;**

APP\_DATA appData **=** **{**

**.**messageLine1 **=** "Explorer 16 Demo" **,**

**.**messageLine2 **=** "Press S3 to cont" **,**

**.**messageTime **=** "Time 00: 00: 00 " **,**

**.**messageADC **=** " Pot = 0.00 Vdc "

**}** **;**

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Section: Main Entry Point

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void delay**(**unsigned int ms**)**

**{**

unsigned int i**;**

unsigned char j**;**

**for** **(**i **=** 0**;** i **<** ms**;** i**++)** **{**

**for** **(**j **=** 0**;** j **<** 200**;** j**++)** **{**

Nop**();**

Nop**();**

Nop**();**

Nop**();**

Nop**();**

**}**

**}**

**}**

int main **(** void **)**

**{**

/\* Call the System Intialize routine\*/

SYS\_Initialize **(** **)** **;**

/\* Display welcome message \*/

LCD\_PutString **(** **(**char**\*)** **&**appData**.**messageLine1**[**0**]** **,** **sizeof** **(**appData**.**messageLine1 **)** **-** 1 **)** **;**

LCD\_PutString **(** **(**char**\*)** **&**appData**.**messageLine2**[**0**]** **,** **sizeof** **(**appData**.**messageLine2 **)** **-** 1 **)** **;**

/\*Initialize Timer\*/

TIMER\_SetConfiguration **(** TIMER\_CONFIGURATION\_RTCC **)** **;**

/\*Initialize ADC configure it as 10 bit interrupt driven mode\*/

ADC\_SetConfiguration **(** ADC\_CONFIGURATION\_AUTO\_SAMPLE\_CONVERT **)** **;**

/\* Display initial ADC and Timer messages \*/

LCD\_PutString **(** **(**char**\*)** **&**appData**.**messageTime**[**0**]** **,** **sizeof** **(**appData**.**messageTime **)** **-** 1 **)** **;**

LCD\_PutString **(** **(**char**\*)** **&**appData**.**messageADC**[**0**]** **,** **sizeof** **(**appData**.**messageADC **)** **-** 1 **)** **;**

char moc**[**12**]=**"Moc: "**;**

char czas**[**11**]=**"Czas: "**;**

int czasWrt**=**0**;**

int mocWrt**=**200**;**

char mocWrtStr**[**3**]=**"000"**;**

char czasSek**[**2**]=**"00"**;**

char czasMin**[**2**]=**"00"**;**

void tostring**(**char str**[],** int num**)**

**{**

**if** **(**num**==**0**)**

**{**

str**[**0**]=**'0'**;**

str**[**1**]=**'0'**;**

**}**

**else{**

int numCp**,** i**,** rem**,** len **=** 0**,** n**;**

numCp**=**num**;**

n **=** num**;**

**while** **(**n **!=** 0**)**

**{**

len**++;**

n **/=** 10**;**

**}**

**if** **(**num**==**1**)**

**{**

len**++;**

**}**

**if** **(**num**==**2**)**

**{**

len**++;**

**}**

**if** **(**num**==**3**)**

**{**

len**++;**

**}**

**if** **(**num**==**4**)**

**{**

len**++;**

**}**

**if** **(**num**==**5**)**

**{**

len**++;**

**}**

**if** **(**num**==**6**)**

**{**

len**++;**

**}**

**if** **(**num**==**7**)**

**{**

len**++;**

**}**

**if** **(**num**==**8**)**

**{**

len**++;**

**}**

**if** **(**num**==**9**)**

**{**

len**++;**

**}**

**for** **(**i **=** 0**;** i **<** len**;** i**++)**

**{**

rem **=** numCp **%** 10**;**

numCp **=** numCp **/** 10**;**

str**[**len **-** **(**i **+** 1**)]** **=** rem **+** '0'**;**

**}**

**}**

**}**

/\* Infinite Loop \*/

**while** **(** 1 **)**

**{**

tostring**(**czasSek **,**czasWrt**%**60**);**

tostring**(**czasMin **,**czasWrt**/**60**);**

tostring**(**mocWrtStr **,**mocWrt**);**

LCD\_PutString **(** moc **,** **sizeof(**moc**)** **)** **;**

LCD\_PutString **(** mocWrtStr **,** **sizeof(**mocWrtStr**)** **)** **;**

LCD\_PutString **(** "W" **,** 1 **)** **;**

LCD\_PutString **(** czas **,** **sizeof(**czas**)** **)** **;**

LCD\_PutString **(** czasMin **,** **sizeof(**czasMin**)** **)** **;**

LCD\_PutString **(** ":" **,** 1 **)** **;**

LCD\_PutString **(** czasSek **,** **sizeof(**czasSek**)** **)** **;**

delay**(**500**);**

LCD\_ClearScreen**();**

**while((**BUTTON\_IsPressed**(**BUTTON\_S3**)))**

**{**

**if(**mocWrt**==**200**)**

**{**

mocWrt**=**350**;**

**break;**

**}**

**if(**mocWrt**==**350**)**

**{**

mocWrt**=**600**;**

**break;**

**}**

**if(**mocWrt**==**600**)**

**{**

mocWrt**=**800**;**

**break;**

**}**

**if(**mocWrt**==**800**)**

**{**

mocWrt**=**200**;**

**break;**

**}**

**}**

**while((**BUTTON\_IsPressed**(**BUTTON\_S6**)))**

**{**

czasWrt**+=**30**;**

delay**(**500**);**

**}**

**while((**BUTTON\_IsPressed**(**BUTTON\_S5**))&&(**czasWrt**>**0**))**

**{**

delay**(**1000**);**

**while(**czasWrt**!=**0**)**

**{**

**if((**BUTTON\_IsPressed**(**BUTTON\_S5**)))**

**{**

tostring**(**czasSek **,**czasWrt**%**60**);**

tostring**(**czasMin **,**czasWrt**/**60**);**

tostring**(**mocWrtStr **,**mocWrt**);**

LCD\_PutString **(** moc **,** **sizeof(**moc**)** **)** **;**

LCD\_PutString **(** mocWrtStr **,** **sizeof(**mocWrtStr**)** **)** **;**

LCD\_PutString **(** "W" **,** 1 **)** **;**

LCD\_PutString **(** czas **,** **sizeof(**czas**)** **)** **;**

LCD\_PutString **(** czasMin **,** **sizeof(**czasMin**)** **)** **;**

LCD\_PutString **(** ":" **,** 1 **)** **;**

LCD\_PutString **(** czasSek **,** **sizeof(**czasSek**)** **)** **;**

delay**(**1000**);**

**break;**

**}**

czasWrt**--;**

tostring**(**czasSek **,**czasWrt**%**60**);**

tostring**(**czasMin **,**czasWrt**/**60**);**

tostring**(**mocWrtStr **,**mocWrt**);**

LCD\_PutString **(** moc **,** **sizeof(**moc**)** **)** **;**

LCD\_PutString **(** mocWrtStr **,** **sizeof(**mocWrtStr**)** **)** **;**

LCD\_PutString **(** "W" **,** 1 **)** **;**

LCD\_PutString **(** czas **,** **sizeof(**czas**)** **)** **;**

LCD\_PutString **(** czasMin **,** **sizeof(**czasMin**)** **)** **;**

LCD\_PutString **(** ":" **,** 1 **)** **;**

LCD\_PutString **(** czasSek **,** **sizeof(**czasSek**)** **)** **;**

delay**(**1000**);**

LCD\_ClearScreen**();**

**}**

**}**

**while((**BUTTON\_IsPressed**(**BUTTON\_S4**)))**

**{**

czasWrt**=**0**;**

mocWrt**=**200**;**

delay**(**1000**);**

**}**

**}** **;**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function:

void Update\_LCD( void )

Summary:

Function to update LCD

Description:

This function will update the time on the LCD

Precondition:

None.

Parameters:

None.

Returns:

None.

Remarks:

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Update\_LCD **(** void **)**

**{**

LCD\_PutChar **(** 'T' **)** **;**

LCD\_PutChar **(** 'i' **)** **;**

LCD\_PutChar **(** 'm' **)** **;**

LCD\_PutChar **(** 'e' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** appData**.**tens **+** 0x30 **)** **;**

LCD\_PutChar **(** appData**.**ones **+** 0x30 **)** **;**

Hex2Dec **(** appData**.**minutes **)** **;**

LCD\_PutChar **(** ':' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** appData**.**tens **+** 0x30 **)** **;**

LCD\_PutChar **(** appData**.**ones **+** 0x30 **)** **;**

Hex2Dec **(** appData**.**seconds **)** **;**

LCD\_PutChar **(** ':' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** appData**.**tens **+** 0x30 **)** **;**

LCD\_PutChar **(** appData**.**ones **+** 0x30 **)** **;**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function:

void ConvertADCVoltage ( unsigned int )

Summary:

Function to convert ADC data into volts

Description:

This is the file for the Explorer 16 Dem that converts raw 10 bit ADC data

to volts suitable for the LCD display

Precondition:

None.

Parameters:

None.

Returns:

None.

Remarks:

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void ConvertADCVoltage **(** unsigned int adc\_conv\_data **)**

**{**

/\* reset values \*/

appData**.**adones **=** 0 **;**

appData**.**adtens **=** 0 **;**

appData**.**adhunds **=** 0 **;**

appData**.**adthous **=** 0 **;**

**while** **(** adc\_conv\_data **>** 0 **)**

**{**

/\* test for 1 volt or greater \*/

**if** **(** adc\_conv\_data **>** **(** ONE\_VOLT **-** 1 **)** **)**

**{**

/\* increment 1 volt counter \*/

appData**.**adones**++** **;**

/\* subtract 1 volt \*/

adc\_conv\_data **-=** ONE\_VOLT **;**

**}**

/\* test for 0.1 volt \*/

**else** **if** **(** adc\_conv\_data **>** **(** ONE\_TENTH\_VOLT **-** 1 **)** **)**

**{**

/\* increment tenths \*/

**if** **(** appData**.**adtens **<** 9 **)**

**{**

appData**.**adtens**++** **;**

**}**

**else**

**{**

/\* tenths has rolled over \*/

appData**.**adones**++** **;**

/\* so increment ones and reset tenths \*/

appData**.**adtens **=** 0 **;**

**}**

adc\_conv\_data **-=** ONE\_TENTH\_VOLT **;**

**}**

/\* test for 0.01 volt \*/

**else** **if** **(** adc\_conv\_data **>** **(** ONE\_HUNDREDTH\_VOLT **-** 1 **)** **)**

**{**

/\* increment hundredths \*/

**if** **(** appData**.**adhunds **<** 9 **)**

**{**

appData**.**adhunds**++** **;**

**}**

**else**

**{**

/\* hundredths has rolled over \*/

appData**.**adtens**++** **;**

/\* so increment tenths and reset hundredths \*/

appData**.**adhunds **=** 0 **;**

**}**

adc\_conv\_data **-=** ONE\_HUNDREDTH\_VOLT **;**

**}**

**else** **if** **(** adc\_conv\_data **<=** **(** ONE\_HUNDREDTH\_VOLT **-** 1 **)** **)**

**{**

appData**.**adthous**++** **;**

adc\_conv\_data **--** **;**

**}**

**}**

appData**.**adones **+=** 0x30 **;**

appData**.**adtens **+=** 0x30 **;**

appData**.**adhunds **+=** 0x30 **;**

appData**.**adthous **+=** 0x30 **;**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function:

void Hex2Dec ( unsigned char )

Summary:

Explorer 16 Demo Hex to Decimal Conversion File

Description:

This is the file for the Explorer 16 Dem that converts the hexadecimal data

into decimal format.

Precondition:

None.

Parameters:

None.

Returns:

None.

Remarks:

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Hex2Dec **(** unsigned char count **)**

**{**

/\* reset values \*/

appData**.**hunds **=** 0 **;**

appData**.**tens **=** 0 **;**

appData**.**ones **=** 0 **;**

**while** **(** count **>=** 10 **)**

**{**

**if** **(** count **>=** 200 **)**

**{**

count **-=** 200 **;**

appData**.**hunds **=** 0x02 **;**

**}**

**if** **(**count **>=** 100**)**

**{**

count **-=** 100 **;**

appData**.**hunds**++** **;**

**}**

**if** **(**count **>=** 10 **)**

**{**

count **-=** 10 **;**

appData**.**tens**++** **;**

**}**

**}**

appData**.**ones **=** count **;**

**}**

# Zadanie 5

## Opis zadania:

W oparciu o program przykładowy dla Explorer 16/32 oraz PIC24FJ128GA010.

Wszelkie komunikaty z zadania powinny byd wyświetlone na wyświetlaczu LCD 16x2.

Zegar szachowy dla dwóch graczy

odmierzający czas SD (Sudden Death – całkowity czas na wykonanie ruchów w partii). Naciśnięcie

przycisku pierwszego gracza powoduje odmierzanie czasu drugiego gracza, a naciśnięcie przycisku

drugiego gracza – odmierzanie czasu pierwszego. Gra toczy się do poddania partii, ustalenia remisu,

Mata – to są jednak zdarzenia, których nie odnotowujemy na zegarze. Zegar ma wyświetlid czas

każdego z graczy i w razie gdy któryś z graczy nie wykona ruchu w pozostałym mu czasie wyświetlid

komunikat, że gracz przegrał grę przez czas.

Nastawa czasu – na podstawie potencjometru P2, 3 ustawienia czasu: 5min, 3min, 1 min

S3 – pierwszy gracz

S6 – drugi gracz

## Problemy podczas implementacji:

Podczas implementacji nie napotkałem żadnych problemów.

## Kod:

#include <stdio.h>

#include <stdlib.h>

#include <stddef.h>

#include <stdbool.h>

#define FCY 4000000ULL

#include <libpic30.h>

#include "app.h"

#define ONE\_VOLT 310

#define ONE\_TENTH\_VOLT 31

#define ONE\_HUNDREDTH\_VOLT 3

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Section: File Scope Variables and Functions

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Update\_LCD **(** void **)** **;**

void SYS\_Initialize **(** void **)** **;**

extern void ConvertADCVoltage **(** unsigned int **)** **;**

extern void Hex2Dec **(** unsigned char **)** **;**

APP\_DATA appData **=** **{**

**.**messageLine1 **=** "Explorer 16 Demo" **,**

**.**messageLine2 **=** "Press S4 to cont" **,**

**.**messageTime **=** "Chess timer " **,**

**.**messageADC **=** " "

**}** **;**

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Section: Main Entry Point

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int main **(** void **)**

**{**

/\* Call the System Intialize routine\*/

SYS\_Initialize **(** **)** **;**

/\* Display welcome message \*/

LCD\_PutString **(** **(**char**\*)** **&**appData**.**messageLine1**[**0**]** **,** **sizeof** **(**appData**.**messageLine1 **)** **-** 1 **)** **;**

LCD\_PutString **(** **(**char**\*)** **&**appData**.**messageLine2**[**0**]** **,** **sizeof** **(**appData**.**messageLine2 **)** **-** 1 **)** **;**

/\* wait here until switch S3 is pressed \*/

**while** **(!**BUTTON\_IsPressed **(** BUTTON\_S4 **))** **;**

/\*Initialize Timer\*/

TIMER\_SetConfiguration **(** TIMER\_CONFIGURATION\_RTCC **)** **;**

/\*Initialize ADC configure it as 10 bit interrupt driven mode\*/

ADC\_SetConfiguration **(** ADC\_CONFIGURATION\_AUTO\_SAMPLE\_CONVERT **)** **;**

int time1 **=** 300**;**

bool count1 **=** **false;**

int time1M **=** 0**;**

int time1S **=** 0**;**

int time2 **=** 300**;**

bool count2 **=** **false;**

int time2M **=** 0**;**

int time2S **=** 0**;**

bool g1Lose **=** **false;**

bool g2Lose **=** **false;**

bool game **=** **false;**

unsigned int pot **=** 0**;**

/\* Infinite Loop \*/

**while** **(** 1 **)**

**{**

**if(!**g1Lose **&&** **!**g2Lose **){**

**if** **(** appData**.**adc\_lcd\_update **)**

**{**

**if(!**game**){**

ConvertADCVoltage **(** appData**.**temp1 **)** **;**

float w **=** **(**appData**.**adones**-**'0'**)** **+** **(**appData**.**adtens**-**'0'**)\***0.1f **+** **(**appData**.**adhunds**-**'0'**)** **\*** 0.01f**;**

float x **=** w**/**3.3**;**

int y **=** x **\*** 100**;**

**if(**y **<** 33**){**

time1 **=** 300**;**

time2 **=** 300**;**

**}**

**if(**y **<** 66 **&&** y **>** 33**){**

time1 **=** 180**;**

time2 **=** 180**;**

**}**

**if(**y **>** 66**){**

time1 **=** 60**;**

time2 **=** 60**;**

**}**

**}**

time1M **=** time1**/**60**;**

time1S **=** time1 **-** **(**time1M **\*** 60**);**

time2M **=** time2**/**60**;**

time2S **=** time2 **-** **(**time2M **\*** 60**);**

LCD\_PutChar **(** 'G' **)** **;**

LCD\_PutChar **(** 'r' **)** **;**

LCD\_PutChar **(** 'a' **)** **;**

LCD\_PutChar **(** 'c' **)** **;**

LCD\_PutChar **(** 'z' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** '1' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** 'G' **)** **;**

LCD\_PutChar **(** 'r' **)** **;**

LCD\_PutChar **(** 'a' **)** **;**

LCD\_PutChar **(** 'c' **)** **;**

LCD\_PutChar **(** 'z'**)** **;**

LCD\_PutChar **(** ' '**)** **;**

LCD\_PutChar **(** '2' **)** **;**

LCD\_PutChar **(** time1M **/** 10 **+** '0' **)** **;**

LCD\_PutChar **(** time1M **%** 10 **+** '0' **)** **;**

LCD\_PutChar **(** ':' **)** **;**

LCD\_PutChar **(** time1S**/**10 **+** '0' **)** **;**

LCD\_PutChar **(** time1S **%** 10 **+** '0' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** time2M**/**10 **+** '0' **)** **;**

LCD\_PutChar **(** time2M **%** 10 **+** '0' **)** **;**

LCD\_PutChar **(** ':' **)** **;**

LCD\_PutChar **(** time2S**/**10 **+** '0' **)** **;**

LCD\_PutChar **(** time2S **%** 10 **+** '0' **)** **;**

**if(**BUTTON\_IsPressed **(** BUTTON\_S3 **)){**

game **=** **true;**

count1 **=** **true;**

count2 **=** **false;**

**}**

**if(**count1 **&&** time2 **>** 0**){**

time2 **=** time2 **-** 1**;**

\_\_delay\_ms**(**1000**);**

**}**

**if(**BUTTON\_IsPressed **(** BUTTON\_S6 **)){**

game **=** **true;**

count1 **=** **false;**

count2 **=** **true;**

**}**

**if(**count2 **&&** time1 **>** 0**){**

time1 **=** time1 **-** 1**;**

\_\_delay\_ms**(**1000**);**

**}**

**}**

**if(**time1 **==** 0**){**

**if(!**g1Lose**){**

LCD\_ClearScreen**();**

LCD\_PutString**(**"Gracz1 przegral!"**,** 16**);**

LCD\_PutString**(**"\* Koniec czasu \*"**,** 16**);**

g1Lose**=true;**

**}**

**}**

**if(**time2 **==** 0**){**

**if(!**g2Lose**){**

LCD\_ClearScreen**();**

LCD\_PutString**(**"Gracz2 przegral!"**,** 16**);**

LCD\_PutString**(**"\* Koniec czasu \*"**,** 16**);**

g2Lose**=true;**

**}**

**}**

**}**

**}**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function:

void Update\_LCD( void )

Summary:

Function to update LCD

Description:

This function will update the time on the LCD

Precondition:

None.

Parameters:

None.

Returns:

None.

Remarks:

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Update\_LCD **(** void **)**

**{**

LCD\_PutChar **(** 'T' **)** **;**

LCD\_PutChar **(** 'i' **)** **;**

LCD\_PutChar **(** 'm' **)** **;**

LCD\_PutChar **(** 'e' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** appData**.**tens **+** 0x30 **)** **;**

LCD\_PutChar **(** appData**.**ones **+** 0x30 **)** **;**

Hex2Dec **(** appData**.**minutes **)** **;**

LCD\_PutChar **(** ':' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** appData**.**tens **+** 0x30 **)** **;**

LCD\_PutChar **(** appData**.**ones **+** 0x30 **)** **;**

Hex2Dec **(** appData**.**seconds **)** **;**

LCD\_PutChar **(** ':' **)** **;**

LCD\_PutChar **(** ' ' **)** **;**

LCD\_PutChar **(** appData**.**tens **+** 0x30 **)** **;**

LCD\_PutChar **(** appData**.**ones **+** 0x30 **)** **;**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function:

void ConvertADCVoltage ( unsigned int )

Summary:

Function to convert ADC data into volts

Description:

This is the file for the Explorer 16 Dem that converts raw 10 bit ADC data

to volts suitable for the LCD display

Precondition:

None.

Parameters:

None.

Returns:

None.

Remarks:

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void ConvertADCVoltage **(** unsigned int adc\_conv\_data **)**

**{**

/\* reset values \*/

appData**.**adones **=** 0 **;**

appData**.**adtens **=** 0 **;**

appData**.**adhunds **=** 0 **;**

appData**.**adthous **=** 0 **;**

**while** **(** adc\_conv\_data **>** 0 **)**

**{**

/\* test for 1 volt or greater \*/

**if** **(** adc\_conv\_data **>** **(** ONE\_VOLT **-** 1 **)** **)**

**{**

/\* increment 1 volt counter \*/

appData**.**adones**++** **;**

/\* subtract 1 volt \*/

adc\_conv\_data **-=** ONE\_VOLT **;**

**}**

/\* test for 0.1 volt \*/

**else** **if** **(** adc\_conv\_data **>** **(** ONE\_TENTH\_VOLT **-** 1 **)** **)**

**{**

/\* increment tenths \*/

**if** **(** appData**.**adtens **<** 9 **)**

**{**

appData**.**adtens**++** **;**

**}**

**else**

**{**

/\* tenths has rolled over \*/

appData**.**adones**++** **;**

/\* so increment ones and reset tenths \*/

appData**.**adtens **=** 0 **;**

**}**

adc\_conv\_data **-=** ONE\_TENTH\_VOLT **;**

**}**

/\* test for 0.01 volt \*/

**else** **if** **(** adc\_conv\_data **>** **(** ONE\_HUNDREDTH\_VOLT **-** 1 **)** **)**

**{**

/\* increment hundredths \*/

**if** **(** appData**.**adhunds **<** 9 **)**

**{**

appData**.**adhunds**++** **;**

**}**

**else**

**{**

/\* hundredths has rolled over \*/

appData**.**adtens**++** **;**

/\* so increment tenths and reset hundredths \*/

appData**.**adhunds **=** 0 **;**

**}**

adc\_conv\_data **-=** ONE\_HUNDREDTH\_VOLT **;**

**}**

**else** **if** **(** adc\_conv\_data **<=** **(** ONE\_HUNDREDTH\_VOLT **-** 1 **)** **)**

**{**

appData**.**adthous**++** **;**

adc\_conv\_data **--** **;**

**}**

**}**

appData**.**adones **+=** 0x30 **;**

appData**.**adtens **+=** 0x30 **;**

appData**.**adhunds **+=** 0x30 **;**

appData**.**adthous **+=** 0x30 **;**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function:

void Hex2Dec ( unsigned char )

Summary:

Explorer 16 Demo Hex to Decimal Conversion File

Description:

This is the file for the Explorer 16 Dem that converts the hexadecimal data

into decimal format.

Precondition:

None.

Parameters:

None.

Returns:

None.

Remarks:

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Hex2Dec **(** unsigned char count **)**

**{**

/\* reset values \*/

appData**.**hunds **=** 0 **;**

appData**.**tens **=** 0 **;**

appData**.**ones **=** 0 **;**

**while** **(** count **>=** 10 **)**

**{**

**if** **(** count **>=** 200 **)**

**{**

count **-=** 200 **;**

appData**.**hunds **=** 0x02 **;**

**}**

**if** **(**count **>=** 100**)**

**{**

count **-=** 100 **;**

appData**.**hunds**++** **;**

**}**

**if** **(**count **>=** 10 **)**

**{**

count **-=** 10 **;**

appData**.**tens**++** **;**

**}**

**}**

appData**.**ones **=** count **;**

**}**