**Systemy Wbudowane**

Sprawozdanie do zadań 1-3

**Sprawozdanie przygotował:**

Michał Chomczyk 162385

Zadania 1 i 2 zostały wykonane zespołowo przez: Michał Chomczk, Tymoteusz Łuczko, Piotr Dondalski, Paweł Deptuła, Igor Ozga.

Zadanie 3 zostało wykonane samodzielnie przez Michała Chomczyka.

Spis treści

[Zadanie 1 1](#_Toc135130176)

[Opis zadania: 1](#_Toc135130177)

[Problemy podczas implementacji: 2](#_Toc135130178)

[Kod: 2](#_Toc135130179)

[Zadanie 2 6](#_Toc135130180)

[Opis zadania: 6](#_Toc135130181)

[Problemy podczas implementacji: 7](#_Toc135130182)

[Kod: 7](#_Toc135130183)

[Zadanie 3 11](#_Toc135130184)

[Opis zadania: 11](#_Toc135130185)

[Problemy z implementacją: 11](#_Toc135130186)

[Kod: 11](#_Toc135130187)

# 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 switcha pozwalającego na przełączanie się między podprogramami.

## Kod:

#include <stdio.h>

#include <stdlib.h>

#include <stddef.h>

#include <stdbool.h>

#include <xc.h>

#include "app.h"

#define ONE\_VOLT 310

#define ONE\_TENTH\_VOLT 31

#define ONE\_HUNDREDTH\_VOLT 3

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 "

**}** **;**

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 grey**(**int value**)** **{**

**return** value **^** **(**value **>>** 1**);**

**}**

int bcd**(**int value**)** **{**

**return** **((**value**/**10**)** **<<** 4 **|** **(**value**%**10**));**

**}**

void swap**(**int**\*** selected\_subfunction**)** **{**

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

**(\***selected\_subfunction**)** **+=** 1**;**

**if** **((\***selected\_subfunction**)** **==** 10**)** **{**

**(\***selected\_subfunction**)** **=** 1**;**

**}**

**}**

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

**(\***selected\_subfunction**)** **-=** 1**;**

**if** **((\***selected\_subfunction**)** **==** 0**){**

**(\***selected\_subfunction**)** **=** 9**;**

**}**

**}**

**}**

unsigned int xorshift32**(**unsigned long seed**)** **{**

int config **=** 115**;**

unsigned long x **=** seed**;**

x **\*=** config**;**

x **^=** x **<<** 13**;**

x **^=** x **<<** 5**;**

**return** x**;**

**}**

unsigned long generator**(**unsigned long seed**)** **{**

int config **=** 115**;**

unsigned long x **=** seed**;**

x **\*=** config**;**

x **^=** x **<<** 13**;**

x **^=** x **<<** 5**;**

**return** x**;**

**}**

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

**{**

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

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

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

unsigned char later **=** 0**;**

int running\_prog **=** 1**;**

unsigned long seed **=** 21337**;**

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

**{**

// Zadanie 1

later **=** 0**;**

**while(**running\_prog **==** 1**){**

LCD\_ClearScreen**();**

char str**[**20**];**

sprintf**(**str**,** "program %d "**,** running\_prog**);**

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

LATA **=** later**;**

later**++;**

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

**if** **(**later **==** 255**)** **{**

later **=** 0**;** // reset

**}**

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

swap**(&**running\_prog**);**

**}**

**}**

// Zad 2

later **=** 255**;**

**while(**running\_prog **==** 2**){**

LCD\_ClearScreen**();**

char str**[**20**];**

sprintf**(**str**,** "program %d "**,** running\_prog**);**

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

LATA **=** later**;**

later**--;**

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

**if** **(**later **==** 0**)** **{**

later **=** 255**;** // reset

**}**

delay**(**500**);** // czas na zmiane przyc

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

swap**(&**running\_prog**);**

**}**

**}**

// Zad 3

later **=** 0**;**

**while(**running\_prog **==** 3**){**

LCD\_ClearScreen**();**

char str**[**20**];**

sprintf**(**str**,** "program %d"**,** running\_prog**);**

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

LATA **=** grey**(**later**);**

later**++;**

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

**if** **(**later **==** 255**)** **{**

later **=** 0**;** // reset

**}**

delay**(**500**);** // czas na zmiane przyc

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

swap**(&**running\_prog**);**

**}**

**}**

// Zad 4

later **=** 255**;**

**while(**running\_prog **==** 4**){**

LCD\_ClearScreen**();**

char str**[**20**];**

sprintf**(**str**,** "program %d"**,** running\_prog**);**

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

LATA **=** grey**(**later**);**

later**--;**

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

**if** **(**later **==** 0**)** **{**

later **=** 255**;** // reset

**}**

delay**(**500**);** // czas na zmiane przyc

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

swap**(&**running\_prog**);**

**}**

**}**

// z5

later **=** 0**;**

**while(**running\_prog **==** 5**){**

LCD\_ClearScreen**();**

char str**[**20**];**

sprintf**(**str**,** "program %d"**,** running\_prog**);**

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

LATA **=** bcd**(**later**);**

later**++;**

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

**if** **(**later **==** 99**)** **{**

later **=** 0**;**

**}**

delay**(**500**);** // czas na zmiane przyc

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

swap**(&**running\_prog**);**

**}**

**}**

// z6

later **=** 99**;**

**while(**running\_prog **==** 6**){**

LCD\_ClearScreen**();**

char str**[**20**];**

sprintf**(**str**,** "program %d"**,** running\_prog**);**

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

LATA **=** bcd**(**later**);**

later**--;**

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

**if** **(**later **==** 0**)** **{**

later **=** 99**;**

**}**

delay**(**500**);** // czas na zmiane przyc

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

swap**(&**running\_prog**);**

**}**

**}**

// z7

later **=** 7**;**

**while(**running\_prog **==** 7**){**

LCD\_ClearScreen**();**

char str**[**20**];**

sprintf**(**str**,** "program %d"**,** running\_prog**);**

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

int i **=** 0**;**

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

LATA **=** later**;**

later **<<=** 1**;**

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

**}**

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

LATA **=** later**;**

later **>>=** 1**;**

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

**}**

delay**(**500**);** // czas na zmiane przyc

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

swap**(&**running\_prog**);**

**}**

**}**

// z8

int ledy\_turned\_on **=** 7**;**

int leds\_on **=** 1**;**

int leds\_queued **=** 0**;**

**while(**running\_prog **==** 8**){**

LCD\_ClearScreen**();**

char str**[**20**];**

sprintf**(**str**,** "program %d"**,** running\_prog**);**

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

**for(**int i **=** 0**;** i **<** ledy\_turned\_on**;** i**++){**

LATA **=** leds\_on **+** leds\_queued**;**

leds\_on **<<=** 1**;**

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

**}**

leds\_queued **+=** leds\_on**;**

LATA **=** leds\_queued**;**

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

leds\_on **=** 1**;**

ledy\_turned\_on**--;**

**if(**ledy\_turned\_on **==** **-**1**){**

ledy\_turned\_on **=** 7**;**

leds\_queued **=** 0**;**

**}**

delay**(**500**);** // czas na zmiane przyc

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

swap**(&**running\_prog**);**

**}**

**}**

// z9

**while(**running\_prog **==** 9**){**

LCD\_ClearScreen**();**

char str**[**20**];**

sprintf**(**str**,** "program %d"**,** running\_prog**);**

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

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

seed **=** generator**(**seed**);**

int result **=** **(**int**)** **(**seed **%** 64**);**

LATA **=** result**;**

delay**(**500**);** // czas na zmiane przyc

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

swap**(&**running\_prog**);**

**}**

**}**

**}**

**}**

# 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 oraz sam czujnik temperatury działał niespójnie.

## Kod:

#include <stdio.h>

#include <stdlib.h>

#include <stddef.h>

#include <stdbool.h>

#include "app.h"

#include <p24fj128ga010.h> //z zad 1

#define ONE\_VOLT 310

#define ONE\_TENTH\_VOLT 31

#define ONE\_HUNDREDTH\_VOLT 3

#define SCALE 308L // z zad 1

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 "

**};**

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 alarmActive **=** 0**;**

void triggerAlarm**(**int threshold**)** **{**

int loopCounter**;**

**if** **(**appData**.**adhunds **-** '0' **>** threshold**)** **{**

**if** **(**alarmActive **==** 0**)** **{**

**for** **(**loopCounter **=** 0**;** loopCounter **<=** 10**;** loopCounter**++)** **{**

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

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

**if** **(**appData**.**adhunds **-** '0' **<=** threshold**)** **{**

**continue;**

**}**

LED\_On**(**LED\_D9**);**

delay**(**150**);**

LED\_Off**(**LED\_D9**);**

delay**(**150**);**

**}**

**if** **(**appData**.**adhunds **-** '0' **>** threshold**)** **{**

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

**}**

alarmActive **=** 1**;**

**}**

**}** **else** **{**

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

alarmActive **=** 0**;**

**}**

**}**

void task2**()** **{**

float baseTemp **=** 25**;**

**if** **(**BUTTON\_IsPressed**(**BUTTON\_S4**))** **{**

triggerAlarm**(-**1**);**

**}**

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

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

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

float temperature **=** **(**voltage **-** 0.1**)\***100 **-** 46**;**

char str**[**20**];** //size of the number

sprintf**(**str**,** "temperatura %g stopni"**,** temperature**);**

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

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

**if** **(**baseTemp **<=** temperature**)** **{**

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

**if** **(**baseTemp **>** temperature**)** **{**

**break;**

**}**

LED\_On**(**LED\_D3**);**

delay**(**150**);**

LED\_Off**(**LED\_D3**);**

delay**(**150**);**

**}**

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

**}** **else** **{**

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

**}**

**}** **else** **{**

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

delay**(**3000**);**

**}**

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

LCD\_ClearScreen**();**

appData**.**adc\_lcd\_update **=** 0**;**

**}**

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

SYS\_Initialize**();**

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

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

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

task2**();**

**};**

**}**

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 zadao powinny byd wyświetlone na wyświetlaczu LCD 16x2.

Reklama!

Atrakcyjna, - temat dowolny

## Problemy z implementacją:

Osoba pisząca to sprawozdanie jest mało kreatywna więc największą trudność sprawiło wymyślenie tematu reklamy.

## 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

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 "

**}** **;**

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 **(** **)** **;**

/\* Infinite Loop \*/

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

**{**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

delay**(**2000**);**

LCD\_ClearScreen**();**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

delay**(**2000**);**

LCD\_ClearScreen**();**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

delay**(**2000**);**

LCD\_ClearScreen**();**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

delay**(**2000**);**

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 **;**

**}**