**Rok. ak. 2019/2020   
grupa dziekańska: 4AiR2**

**termin realizowania laboratorium: czwartek 8:15**

**SPRAWOZDANIE Z LABORATORIUM**

**TECHNIKI MIKROPROCESOROWEJ**

**Autorzy sprawozdania:**

**1. Mateusz Papuga 222116**

**2. Adam Brząkała 222048**

imię nazwisko nr indeksu

**Data wykonania sprawozdania: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Warunkiem wpisania oceny laboratoryjnej do systemu elektronicznego jest wysłanie e-maila na adres: **andrzej.radecki @ p.lodz.pl** ze sprawozdaniem i kompletnymi kodami źródłowymi.

Temat e-mail **MUSI** mieć strukturę: TM\_dzień zajęć\_godzina zajęć\_nazwisko1\_nazwisko2

*gdzie nazwisko1 i nazwisko2 identyfikują autorów sprawozdania*

**Przykładowy** temat e-maila: TM\_wtorek\_12:15\_Radecki\_Rybicki

W wersji drukowanej sprawozdania należy zamieścić **załącznik:** Wydruk wysłanego e-maila.

**Wersja 0.3**

**Spis treści**

[**Programowanie systemów mikroprocesorowych w języku C.**](#_gjdgxs) **4**

[Charakterystyka zadań](#_30j0zll) 4

[Zadanie nr 1](#_1fob9te) 4

[Zadanie nr 2](#_3znysh7) 4

[Struktura opracowanego oprogramowania](#_2et92p0) 5

[Zadanie nr 1](#_tyjcwt) 5

[Zadanie nr 2](#_3dy6vkm) 6

[Kody źródłowe zaimplementowanych algorytmów](#_1t3h5sf) 7

[Zadanie nr 1](#_2s8eyo1) 7

[Zadanie nr 2](#_17dp8vu) 8

[**Programowanie systemów mikroprocesorowych z elementami asemblera.**](#_9znglab07iea) **9**

[Charakterystyka zadań](#_26in1rg) 9

[Zadanie nr 1](#_lnxbz9) 9

[Zadanie nr 2](#_35nkun2) 9

[Struktura opracowanego oprogramowania](#_1ksv4uv) 9

[Zadanie nr 1](#_44sinio) 9

[Zadanie nr 2](#_2jxsxqh) 9

[Kody źródłowe zaimplementowanych algorytmów](#_z337ya) 9

[Zadanie nr 1](#_3j2qqm3) 9

[Zadanie nr 2](#_1y810tw) 9

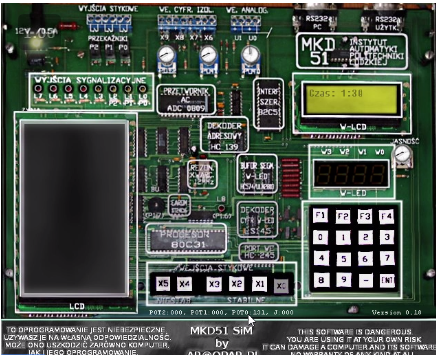
# Programowanie systemów **mikroprocesorowych** w języku C.

## Charakterystyka zadań

(Opis zadania, zdjęcia (zrzuty ekranu) działającej aplikacji w różnych stanach jej pracy)

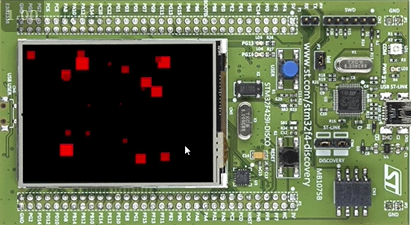
### Zadanie nr 1

MKD51 - Zrealizować minutnik cyfrowy na W-LCD (wskazujący minuty i sekundy) ustawiany POT0 (na wartość maksymalną 2:59) i uruchamiany przyciskiem X0.



### Zadanie nr 2

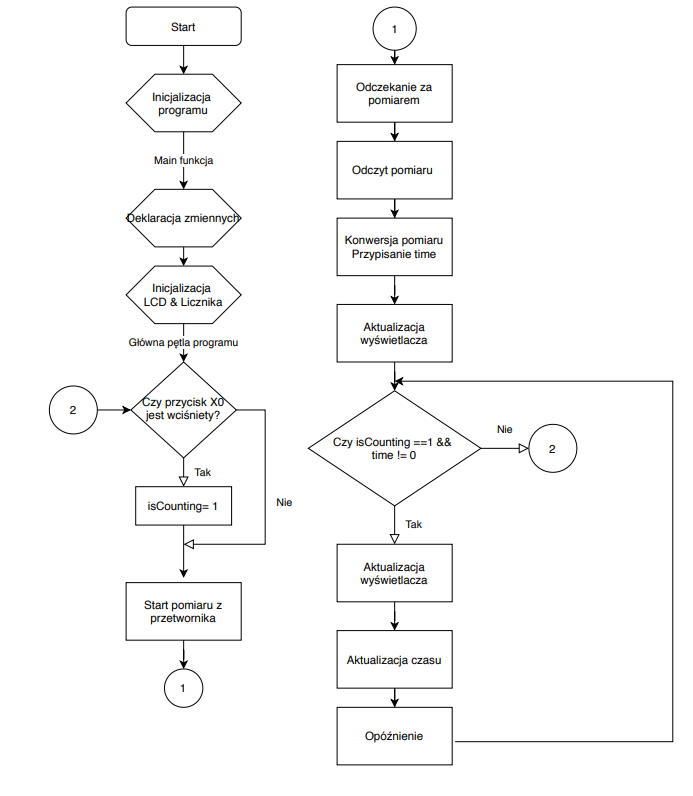
STM32 - Narysować 20 kwadratów na LCD o losowych współrzędnych położenia. Pierwszy kwadrat ma mieć bok o szerokości 21 pixeli i być dowolnego koloru o maksymalnym natężeniu składowych RGB.s Każdy kolejny kwadrat ma być tego samego koloru ale ciemniejszy o najmniejszą możliwą redukcję jasności i mniejszy o 1 piksel wzdłuż każdego boku.



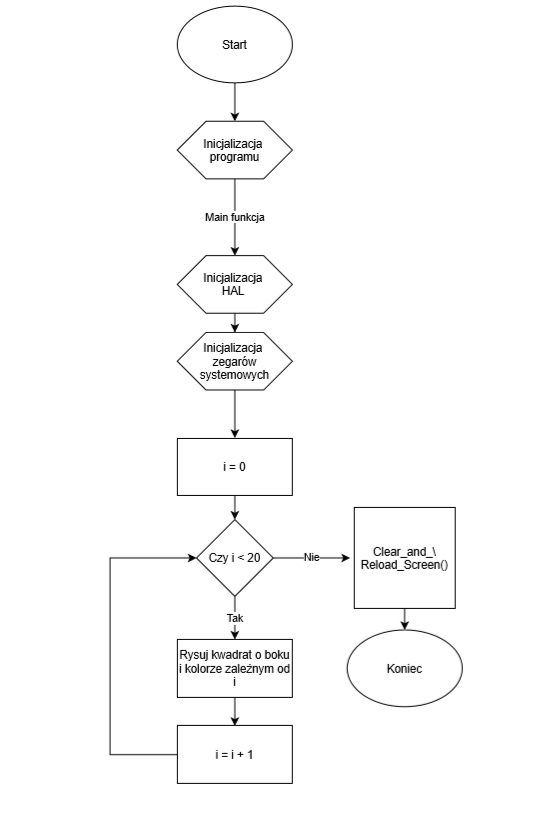
## Struktura opracowanego oprogramowania

(Schematy blokowe zaimplementowanych algorytmów)

### Zadanie nr 1



### Zadanie nr 2



## Kody źródłowe zaimplementowanych algorytmów

(Kod należy sformatować wielokolumnowo. Maksymalna objętość kodu: 1 strona na zadanie.).

### Zadanie nr 1

|  |  |
| --- | --- |
| #include <reg52.h>  #include <intrins.h>  #include "toolbox.h"  #include "lcd.h"  uint8 xdata PTWY \_at\_ 0x8008;  uint8 xdata PTAC \_at\_ 0x8000;  uint8 xdata POT0 \_at\_ 0x8005;  void init\_timer0()  {  TL0 = TH0 = 0; // war. poczatkowa  TMOD |= BV(0); // Licznik 0 w trybie 1    }  void delay(unsigned char sekundy, unsigned int milisekundy) // max do 1 minuty  {  unsigned ms = 0;  TL0 = -1000;  TH0 = (-1000) >> 8;  TR0 = 1; // Uruchomienie licznika    while(ms < (unsigned)(sekundy)\*1000 + milisekundy)  {  if(TF0)  {  ++ms;  TL0 = -1000 & 0xff;  TH0 = (-1000) >> 8;  TF0 = 0;  }  }    TR0 = 0;  }uint8 digitCount(uint8 number)  {  uint8 count = 1;    while(number /= 10)  ++count;    return count;  } | void numToStr(uint8 number, char \*outStr, uint8 outStrLen)  {  uint8 digits = digitCount(number);  for(outStrLen = outStrLen - 1; outStrLen >= digits; --outStrLen)  \*(outStr + outStrLen) = ' ';    while(digits--)  {  outStr[digits] = '0' + number % 10;  number /= 10;  }  }  void main(void)  {  unsigned int time;  char text[] = "Czas: 0:00";  uint8 isCounting = 0;    lcdInit();  init\_timer0();  while(1)  {  if(!(PTWY & BV(0)))  isCounting = 1;    POT0 = 0;  delay(0, 10);  time = (unsigned long)(PTAC)\*179/255;  numToStr(time/60, text+6, 1);  numToStr(time%60, text+8, 2);  lcdWriteText(text);    while(isCounting && time != 0)  {  numToStr(time/60, text+6, 1);  numToStr(time%60, text+8, 2);    lcdWriteText(text);    --time;  delay(1, 0);  }  }  } |

### Zadanie nr 2

|  |  |
| --- | --- |
| #include "main\_base.h"  #include <stdint.h>  #include <stdlib.h>  uint16\_t to\_pixel\_RGB565(uint8\_t r, uint8\_t g, uint8\_t b)  {  return ((uint16\_t)(r & 0x1f) << 11) | ((uint16\_t)(g & 0x3f) << 5) | (uint16\_t)(b & 0x1f);  }  void drawPixel(int x, int y, uint16\_t colour)  {  LCD\_BUF[x + y\*240] = colour;  }  void drawSquare(uint16\_t x, uint16\_t y, uint8\_t size, uint16\_t colour)  {  int pos\_x, pos\_y;  for(pos\_x = x; pos\_x <= x + size; ++pos\_x)  {  for(pos\_y = y; pos\_y <= y + size; ++pos\_y)  {  drawPixel(pos\_x, pos\_y, colour);  }  }  } | int main(void) {  HAL\_Init();  SystemClock\_Config();  for(uint8\_t i = 0; i < 20; ++i)  {  drawSquare(rand()%320, rand()%240,  21 - i, to\_pixel\_RGB565(0xff - i, 0, 0));  }  Clear\_And\_Reload\_Screen();  while (1)  {  }  } |

# 

# Programowanie systemów mikroprocesorowych z elementami asemblera.

## Charakterystyka zadań

(Opis zadania, zdjęcia (zrzuty ekranu) działającej aplikacji w różnych stanach jej pracy)

### Zadanie nr 1

............................................

### Zadanie nr 2

.............................................

## Struktura opracowanego oprogramowania

(Schematy blokowe zaimplementowanych algorytmów)

### Zadanie nr 1

............................................

### Zadanie nr 2

.............................................

## Kody źródłowe zaimplementowanych algorytmów

(Kod należy sformatować wielokolumnowo. Maksymalna objętość kodu: 1 strona na zadanie.).

### Zadanie nr 1

|  |  |
| --- | --- |
| #define .....  .............  ............. | .............  while (1) { .............  } |

### Zadanie nr 2

|  |  |
| --- | --- |
| #define .....  .............  ............. | .............  while (1) { .............  } |