

# ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – ГАБРОВО

*Катедра: КСТ*

*Дисциплина: ОИРВ*

## КУРСОВ ПРОЕКТ

*Тема: : Графичен конструктор на символи върху LCD-дисплей, синтезиращ кода на асемблерска програма, която ги изписва.*

*Графичният конструктор е тестван на микрокомпютър AVR Butterfly на фирма Atmel.*

*Кодът на графичния конструктор е разработен с Microsoft Visual C#. На машината, на която конструкторът се стартира, трябва задължително да бъде инсталиран .NET Framework 2.0.*

*Изготвил: инж. Джем Себахтин Риза, спец. КСТ, ОКС "Магистър", курс I, фак. No: 20506005*

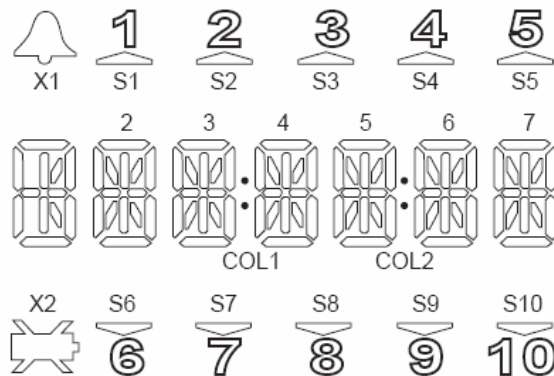
Дата: 09.4.2007 г.

Ръководител:

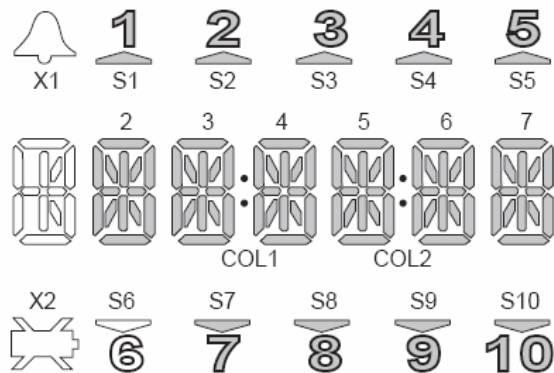
/ доц. д-р. Р. Райчев /

## STK502 LCD Дисплей

LCD дисплея който е монтиран на STK502 е илюстриран на фиг. 1. той се състои от седем символа и различни варианти фиксирани символи – числата от 1 до 10, звънец, символ за намаляваща батерия и стрелки за навигация.



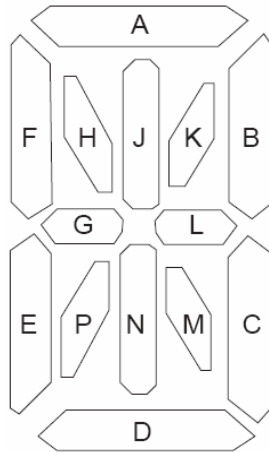
Фиг. 1 – Общ изглед на LCD Дисплей



Фиг. 2 – LCD сегментите които са свързани към STK502

LCD дисплея съдържа общо 120 сегмента които се контролират през 4 задни-равнини (back-planes) и 30 сегментни линии. Някои от сегментите на LCD дисплея не са свързани в стандартната конфигурация с която пристига STK502 (фиг. 2).

STK502 LCD има 6 подобни сегментни групи, като всяка една от тях е способна да визуализира символ. Всеки един елемент от LCD групата е рефериран към съответен бит. Всяка една група съдържа 14 отделни сегмента. Фигура 3 илюстрира една LCD група образуваща символ.



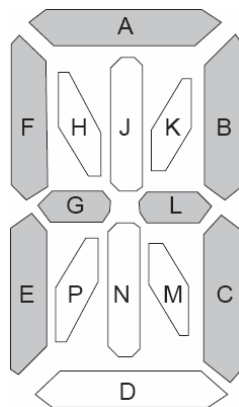
Фиг. 3 – Сегменти и реферирани букви от LCD дисплея.

### **Регистри за данни LCD (LCD Data Registers)**

LCD сегментите се управляват индивидуално от 20 регистъра за данни (LCDDR19:0). Не всички 20 регистъра за данни се използват само 16 от тях всъщност се използват в ATmega169. Всеки сегмент уникално се контролира спрямо вдигането или нулирането на бит от дадения регистър за данни.

### **Управление на LCD сегментите**

LCD сегментите се управляват от регистри за данни както е показано на фигура 4 за да се получи на екрана буквата „А” е необходимо да бъдат активни следните сегменти  $A=\{A, B, C, G, L, E, F\}$ .



Фиг. 4 – Илюстрация на буквата „А”

## Таблицы с регистри за данни манипулиращи символите върху LCD дисплея

Таблица 1 – Управляващи регистри от LCD дисплея за символ 1

Регистър	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
LCDDR0					1-K	-	-	1-A
LCDDR5					1-J	1-F	1-H	1-B
LCDDR10					1-L	1-E	1-G	1-C
LCDDR15					1-M	1-P	1-N	1-D

Таблица 2 – Управляващи регистри от LCD дисплея за символ 2

Регистър	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
LCDDR0	2-K	-	-	2-A				
LCDDR5	2-J	2-F	2-H	2-B				
LCDDR10	2-L	2-E	2-G	2-C				
LCDDR15	2-M	2-P	2-N	2-D				

Таблица 3 – Управляващи регистри от LCD дисплея за символ 3

Регистър	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
LCDDR1					3-K	-	-	3-A
LCDDR6					3-J	3-F	3-H	3-B
LCDDR11					3-L	3-E	3-G	3-C
LCDDR16					3-M	3-P	3-N	3-D

Таблица 4 – Управляващи регистри от LCD дисплея за символ 4

Регистър	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
LCDDR1	4-K	-	-	4-A				
LCDDR6	4-J	4-F	4-H	4-B				
LCDDR11	4-L	4-E	4-G	4-C				
LCDDR16	4-M	4-P	4-N	4-D				

Таблица 5 – Управляващи регистри от LCD дисплея за символ 5

Регистър	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
LCDDR2					5-K	-	-	5-A
LCDDR7					5-J	5-F	5-H	5-B
LCDDR12					5-L	5-E	5-G	5-C
LCDDR17					5-M	5-P	5-N	5-D

Таблица 6 – Управляващи регистри от LCD дисплея за символ 6

---

<b>Регистър</b>	<b>Бит 7</b>	<b>Бит 6</b>	<b>Бит 5</b>	<b>Бит 4</b>	<b>Бит 3</b>	<b>Бит 2</b>	<b>Бит 1</b>	<b>Бит 0</b>
<b>LCDDR2</b>	6-K	-	-	6-A				
<b>LCDDR7</b>	6-J	6-F	6-H	6-B				
<b>LCDDR12</b>	6-L	6-E	6-G	6-C				
<b>LCDDR17</b>	6-M	6-P	6-N	6-D				

---

# Разработка на програмна система за визуално редактиране на STK502 LCD Дисплей

## *Ръководство за програмиста*

LCD Редактора е разработен с помощта на Microsoft Visual C# и се нуждае да бъде инсталирам непременно .NET Framework 2.0 на машината към която се стартира апликацията.

Главния и клас на апликацията се казва `MainForm`, който се явява обвивка на стартиращия се редактор.

Използвани функции и тяхното описание

```
public MainForm( )
```

Конструктор на главната форма, в него се извършват някои инициализационни операции, като заделянето на памет за таблици с отмествания и маски.

```
private void VisualAsmCode( )
```

Функцията визуализира на екрана определен от диалога за визуализация на асемблер код. Кодът се генерира чрез алгоритъм, който използва директно зареждане със стойности на регистрите за данни. Обхождат се всички регистри за данни и тези които имат стойности различни от 0 се генерира специален код за тях образуван от две инструкции.

```
private void InitializeSymbolsRegisters( )
```

Функцията се използва за инициализиране на таблица с номера на регистри като резултат от функцията се получава следната редица с числа { 0, 0, 1, 1, 2, 2 }, тези числа не са случайни чрез тях могат да се сметнат кой регистър принадлежи за съответния сегмент.

```
private void InitializeRegistersOffsets( )
```

Използва се за инициализирането на отместването на регистрите спрямо всеки един буквен сегмент.

```
private void InitializeLEDsPos( )
```

---

Функцията се използва за инициализиране на битовите маски, които се използват при битови операции, за да се установи даден бит в единица.

```
private void MakeAllNullRegisters( )
```

Нулира всички регистри за данни.

```
void UpdateRegisters( )
```

Функцията инициализира наново всички регистри за данни като при срещане на светнат сегмент съответния бит от регистъра за данни се вдига.

```
private void VisualRegisters( )
```

Чрез тази функция се визуализират всички регистри на установеното на екрана място за тях. Като всеки един регистър се представя в десетична, шестнадесетична и двоична бройна система.

```
private void LoadBitmaps( )
```

Функцията се използва за инициализация на изображенията, които се използват за илюстрация на всеки един сегмент.

```
private void InitializeBitmaps( List<Bitmap> onRawBitmaps, List<Bitmap> offRawBitmaps )
```

Използва се при разграничаването на изображенията, които съответстват при светещ сегмент и при не светещ сегмент.

```
private void InitializeLEDS( )
```

Функцията се използва за инициализиране на символите и техните за всеки символ неговите сегменти.

```
private void UpdateAllSymbols( )
```

Функцията асоциира всеки един сегмент спрямо неговото визуално представяне на формата. При несъответствие на визуалните данни с вътрешните данни е необходимо да се използва функцията.

```
private bool GetSymbolAndLEDFromName( string strName, ref int rnSymbol, ref int rnLED )
```

Функцията има специално предназначение, поради причината, че прочита името на дадена контрола и след което се опитва да определи дали това име съответства на даден сегмент. Образуването на имената е следното:

---

**S[число]LED[буква]** където буквата уквзва кой сегмент се активира а числото кой символ.

```
private void UpdateVisual( PictureBox pictureBox, I_LED iLed )
```

Функцията се използва за опресняване на съответната контрола за изображения асоциирано спрямо съответния интерфейс за сегмент.

```
private void OnLEDClickEvent( object sender )
```

Функцията обработва събитие натискане на мишката върху даден сегмент и извършва операции по опресняване на екрана.

```
private interface I_LED
```

Частен интерфейс, който описва един сегмент.

```
void SwitchLED( )
```

Методът е публичен за интерфейс и се използва превключване от едно състояние на сегмента в друго.

```
EN_LED_STATE GetLEDState( )
```

Функцията връща като резултат състоянието на сегмента (включено/изключено).

```
int GetLetterNumber( )
```

Функцията връща като цяло число номера на сегмента, като всеки един уникален номер съответства на буква от символа.

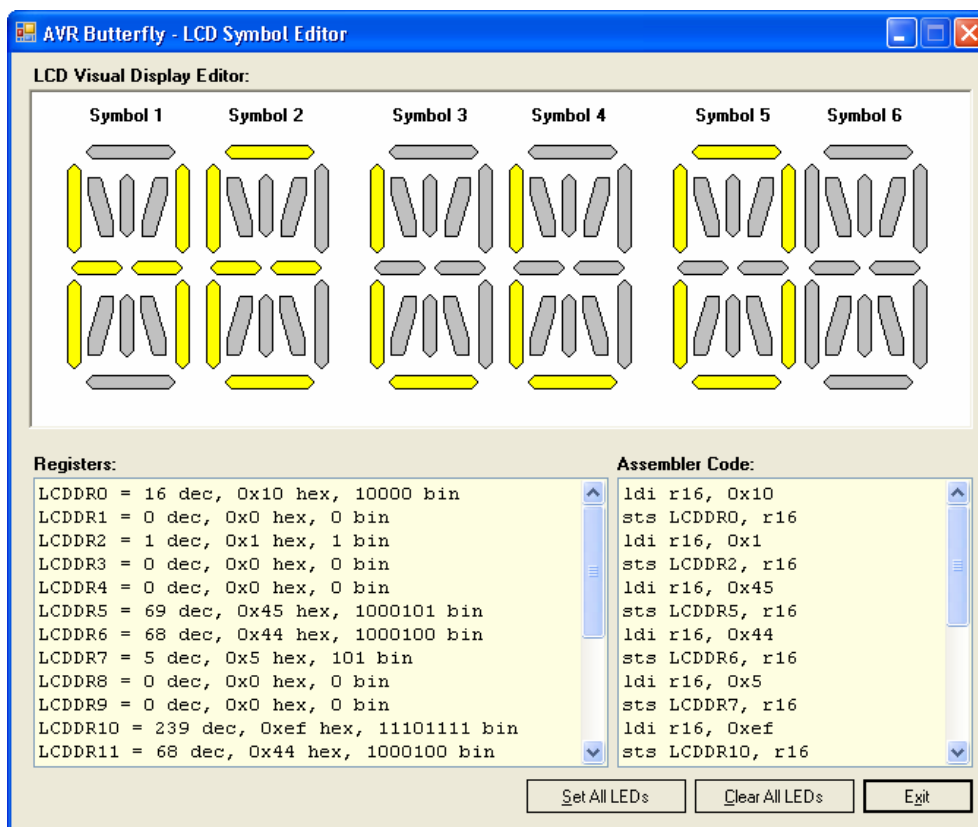
```
private class LCDDisplayLED : I_LED
```

Частния клас имплементира методите на интерфейса `I_LED`.

## ***Ръководство за потребителя***

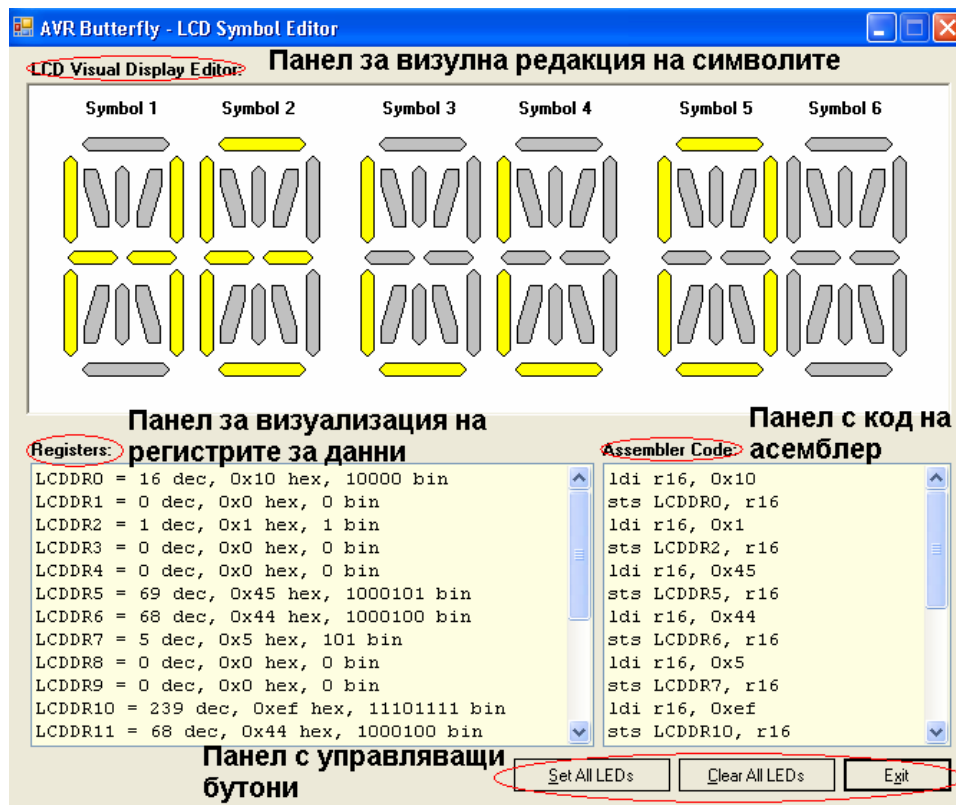
Общия потребителски интерфейс на LCD Редактора е показан на фигура 5.





Фиг. 5 - LCD Редактор

Забелязват се 4 категории части от потребителския интерфейс на LCD Редактора те са илюстрирани на фигура 6, и представляват визуален панел за редактиране на символите, панел показващ състоянието на регистрите за данни, панел за илюстрация на примерен код на асемблер и панел с бутони.



Фиг. 6 – Категории в потребителския интерфейс на LCD Редактора

Както беше споменато по-горе в текста, за да се стартира LCD Редактора е необходимо на машината да бъде инсталиран .NET Framework 2.0. .NET Framework 2.0 може да се намери и да се свали безплатно от интернет или да се инсталира от придружаващия диск този документ. В диска се съдържа документацията на STK502 сорс кода на LCD Редактора.

---

## **Използвана литература**

1. AVR065: LCD Driver for the STK502 and AVR Butterfly
2. STK502 User Guide

## ***Използвани уеб връзки***

1. <http://www.dwelch.com/avr/>

---

## Съдържание

STK502 LCD Дисплей.....	2
Регистри за данни LCD (LCD Data Registers).....	3
Управление на LCD сегментите .....	3
Таблицы с регистри за данни манипулиращи символите върху LCD дисплея .....	4
Разработка на програмна система за визуално редактиране на STK502 LCD Дисплей .....	6
Ръководство за програмиста .....	6
Ръководство за потребителя.....	8
Използвана литература .....	11
Използвани веб връзки .....	11