МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»

(БГТУ им. В.Г.Шухова)

**Лабораторная работа №2**

дисциплина «Программирование микроконтроллеров»

Выполнил: студент группы ВТ-31

Ковалёв И. Д.

Проверил: Гольцов Ю. А.

Белгород 2020

**Цель работы:** изучить принципы работы таймеров в микроконтроллере семейства MCS51.

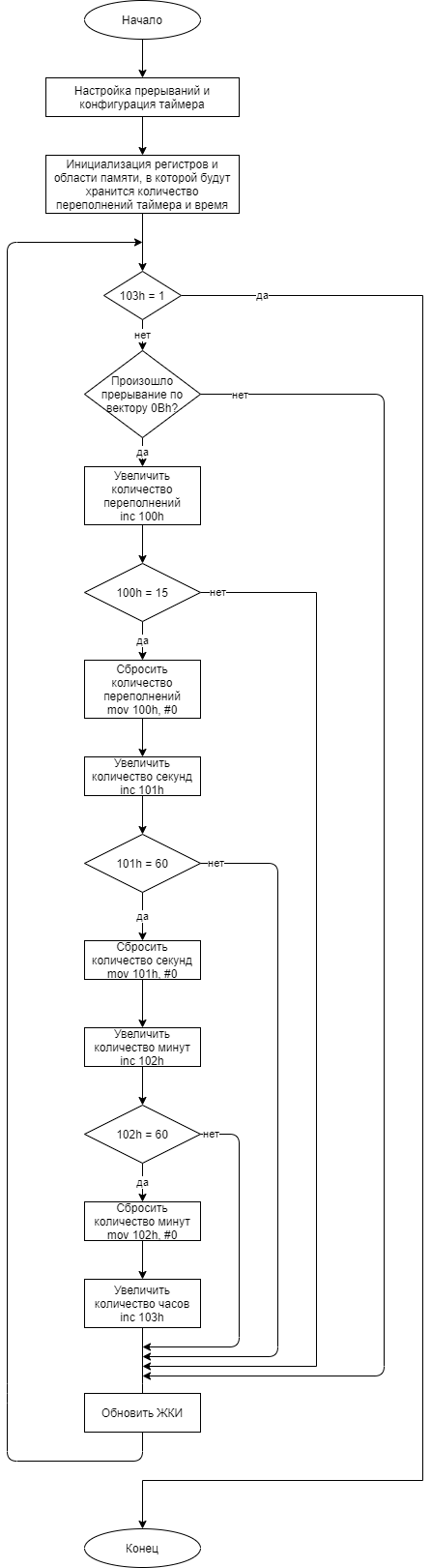
Задание к работе:

1. Создать программу, которая будет выводить на ЖКИ время в виде часы/минуты, секунды же реализовать миганием точки на ЖКИ.

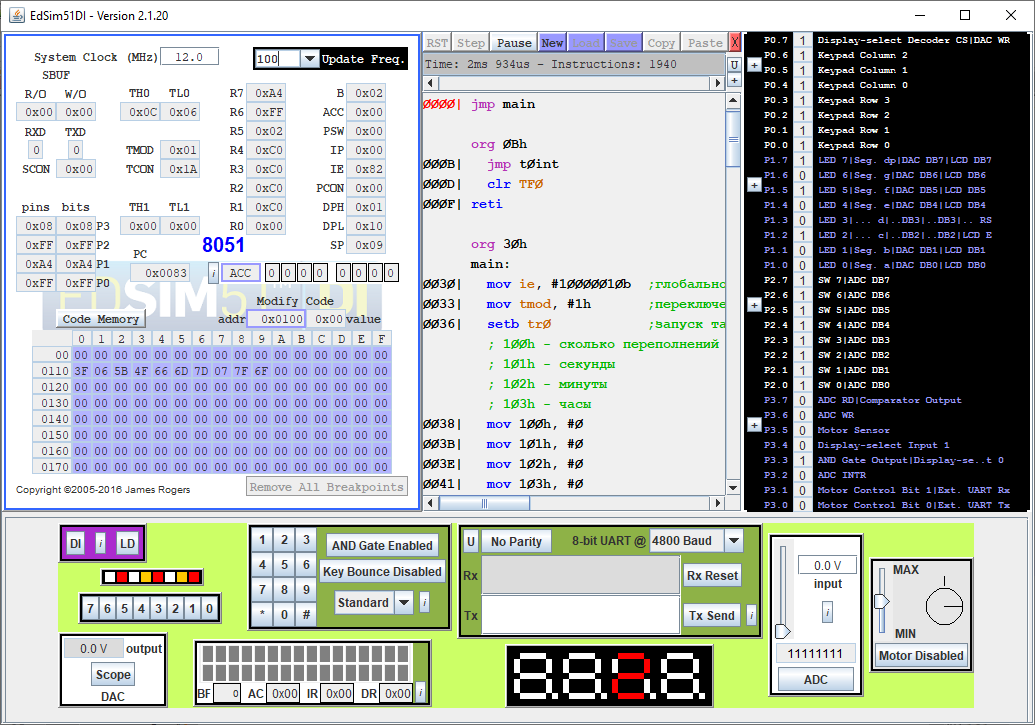
Для реализации данного задания будем использовать таймер 0 в первом режиме. По каждому прерыванию будем увеличивать счетчик переполнений, как только будет накоплено 15 таких переполнений, будем увеличивать счетчик секунд и т. д. Во время, когда прерывание таймера 0 не вызывается, будем обновлять ЖКИ.

В качестве счетчиков переполнений, секунд, минут и часов будем использовать память программы по адресам 0100h-0104h, состояние индикаторов – в регистрах R0-R4.

**Блок-схема разработанной программы:**



**Скриншоты разработанной программы:**



Поскольку обновление ЖКИ происходит посегментно, уловить момент, когда все индикаторы будут гореть невозможно. Если бы данная программа работала на реальном стенде, а не симуляторе, то выглядело бы это как и на любых «реальных» часах.

Вывод: в процессе выполнения были изучены принципы работы таймеров в микроконтроллере семейства MCS51.

**Листинг программы:**

org 00h

jmp main

org 0Bh

jmp t0int

clr TF0

reti

org 30h

main:

mov ie, #10000010b ;глобальное разрешение прерываний и прерывание таймера 0

mov tmod, #1h ;переключение таймера 0 в 1 режим

setb tr0 ;запуск таймера 0

; 100h - сколько переполнений произошло r0 - флаг мигалки

; 101h - секунды r1 - состояние индикатора минуты 1; r2 - состояние индикатора минуты 2

; 102h - минуты

; 103h - часы r3 - состояние индикатора часа 1; r4 - состояние индикатора часа 1

mov 100h, #0

mov 101h, #0

mov 102h, #0

mov 103h, #0

; инициализируем регистры р0 - р4

mov a, #0

call getSegmentFromA

; Поскольку считаем с нуля, повторное вычисление не требуется, все регистры (r1 - r4 ) забиваем нулями

mov 01h, r7 ; Радуемся жизни, потому что из регистра в регистр пересылать нельзя

mov 02h, r7

mov 03h, r7

mov 04h, r7

setb p0.7 ;включаем декодер

mov r5, #0

wait:

call updateLED ; и обновляем ЛЕД, пока не надоест

jmp wait

updateLED:

; Обновляем поочередно каждый индикатор

firstSegment:

cjne r5, #0, secondSegment

;выбираем первый индикатор

mov a, #0

anl a, #0e7h

orl a, #0h

mov p3, a

; Получаем сегмент, который надо вывести

mov a, 102h

call getSegmentFromA

; И выводим его

mov p1, r6

; Увеличивая счетчик текущего индикатора

jmp incCounter

; Аналогично для остальных трех индикаторов

secondSegment:

cjne r5, #1, thirdSegment

mov a, #0

anl a, #0e7h

orl a, #008h

mov p3, a

mov a, 102h

call getSegmentFromA

mov p1, r7

jmp incCounter

thirdSegment:

cjne r5, #2, forthSegment

mov a, #0

anl a, #0e7h

orl a, #010h

mov p3, a

mov a, 103h

call getSegmentFromA

mov p1, r6

jmp incCounter

forthSegment:

cjne r5, #3, incCounter

mov a, #0

anl a, #0e7h

orl a, #018h

mov p3, a

mov a, 103h

call getSegmentFromA

mov p1, r7

jmp resetCounter

; Обнуляем счетчик индикаторов

resetCounter:

mov r5, #0

jmp firstSegment ; И возвращаемся к первому индикатору

; Увеличиваем счетчик, и возвращаемся к первому, т.к. точно не знаем, какой обновляем

incCounter:

inc r5

jmp firstSegment

ret

t0int:

inc 100h

mov a, 100h ; Пересылаем в аккумулятор значение счетчика переполнений для сравнения

;Увеличиваем счетчик переполнений, 15 переполнений - 1 секунда

cjne a, #2, checkSec ; если не равно 15 - пропускаем

mov 100h, #0h ; иначе обнуляем счетчик переполнений

inc 101h ; и увеличиваем счетчик секунд

checkSec:

mov a, 101h ; Пересылаем в аккумулятор значение счетчика секунд для сравнения

;Проверяем секунды, если их 59 - обнуляем и увеличиваем минуты

cjne a, #59, checkMin ; если не равно 59 - пропускаем

mov 101h, #0h ; иначе обнуляем счетчик секунд

inc 102h ; и увеличиваем счетчик минут

checkMin:

mov a, 102h ; Пересылаем в аккумулятор значение счетчика минут для сравнения

;Проверяем минуты, если их 59 - обнуляем и увеличиваем часы

cjne a, #59, skip ; если не равно 59 - пропускаем

mov 102h, #0h ; иначе обнуляем счетчик минут

inc 103h ; и увеличиваем счетчик часов

skip:

reti

getSegmentFromA: ; В А загружено число

mov B, #10

div AB ; Делим на 10, чтобы узнать, из каких цифр состоит

mov DPTR, #0110h ; По адресу 0110h - таблица сегментов

movc a, @a+dptr ; Получаем адрес нужного сегмента

cpl a

mov r6, a ; Аналогично

mov a, b

movc a, @a+dptr

cpl a

mov r7, a

mov a, #0

ret

; Таблица, в которой храним сегмент. 0 - 3Fh, 1 - 06h итд

org 0110h

db 3Fh, 06H, 5Bh, 4Fh, 66h, 6Dh, 7Dh, 07h, 7Fh, 6Fh