**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ **«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. В. Г. ШУХОВА»**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**ЛАБОРАТОРНАЯ РАБОТА №2**

**Дисциплина: Программирование микроконтроллеров**

**Тема: Использование семисегментного индикатора**

Выполнил: ст. группы ВТ-31

Подкопаев Антон Валерьевич

Проверил: ст. пр. кафедры ТК

Гольцов Юрий Александрович

**Белгород 2020**

**Цель работы:** получение навыков работы с внешними интерфейсами для создания часов на семисегментном индикаторе.

**Краткие теоретические сведения**

Индикатор называется семисегментным из-за того, что отображаемый символ строится из отдельных семи сегментов. Внутри корпуса такого индикатора находятся светодиоды, каждый из которых засвечивает свой сегмент.

Одноразрядные семисегментные индикаторы могут быть устроены на светодиодах, лампах тлеющего разряда, электровакуумные индикаторы (катодолюминесцентные, накаливаемые), нити накаливания[5], жидкие кристаллы и т. д. На больших табло наподобие цен на бензин могут применяться механические индикаторы, или блинкерные индикаторы, переключающиеся с помощью электромагнитов.

В обычном светодиодном индикаторе используется девять выводов: один идёт к катодам (минусам) всех сегментов, и остальные восемь — к аноду каждого из сегментов, включая точку. Эта схема называется «схема с общим катодом», существуют также схемы с общим анодом, где имеется общий анод (плюс) и индивидуально подключенные катоды. Часто делают не один, а два общих вывода на разных концах корпуса — это упрощает разводку, не увеличивая габаритов.

Многоразрядные индикаторы часто работают по динамическому принципу: выводы одноимённых сегментов всех разрядов соединены вместе. Чтобы выводить информацию на такой индикатор, управляющая микросхема должна циклически подавать ток на общие выводы всех разрядов, в то время как на выводы сегментов ток подаётся в зависимости от того, зажжён ли данный сегмент в данном разряде. Таким образом, чтобы получить десятиразрядный экран микрокалькулятора, нужны всего восемнадцать выводов (8 анодов и 10 катодов) — а не 81.

Существуют специальные микросхемы семисегментных дешифраторов, переводящие четырёхбитный код в его семисегментное представление. К примеру, отечественные (КР)514ид1 для индикаторов с общим катодом или (КР)514ид2 с общим анодом. Иногда дешифраторы встраивают прямо в индикатор. В настоящее время, в связи с широким распространением однокристальных микроконтроллеров с GPIO, семисегментные светодиодные индикаторы подключаются напрямую к выводам микроконтроллера.

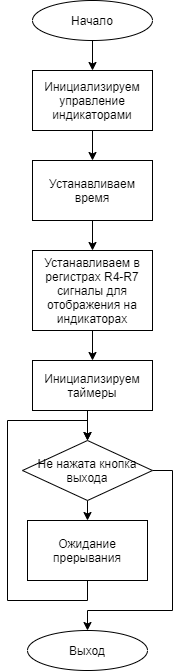
Изображение выглядит как объект, часы, подключен, маленький

Автоматически созданное описание

Задание к работе:

Создать программу, имитирующую работу часов с цифровой индикацией. Часы должны показывать часы и минуты; при этом каждую секунду мигает точка-разделитель между полями.

**Блок-схема разработанной программы**

****

**Код программы**

org 0h

jmp main

; Семисегментный индикатор работает так

; На P3.3, P3.4 сидит декодер , какой из индикаторов включить

; CS к нему идет с порта P0.7

; P1.0-1.7 это выходы abcdefg. с индикаторов

; В основном цикле постоянно обновляем индикатор

; В регистрах R4,5,6,7 храним состояние отдельных индикаторов

; По прерыванию обновляется значение регистров индикаторов

; В цикле постоянно показываем значение регистров

; Обновление индикаторов идет от таймера 1,

; часы отсчитываются таймером 0

; Банк регистров 0 отвечает за часы, банк регистров 1 - за индикацию

; Здесь располагается прерывание от таймера 0

org 00Bh

call timer0\_interrupt

clr TF0

reti

; Здесь располагается прерывание от таймера 1

org 001Bh

call refreshLed

clr TF1

reti

org 30h

main:

; Подготавливаем регистр R3,

; чтобы обновлялись светодиодные индикаторы

mov R3, #0h

; Никто не светится

mov R4, #0FFh

mov R5, #0FFh

mov R6, #0FFh

mov R7, #0FFh

; Начнем с времени 12:59

mov R2, #0

mov 20H, #0

mov 21H, #59

mov 22H, #12

; Часы

mov A, 22H

call num\_to\_seg

mov R6, 01h

mov R7, 00h

; Минуты

mov A, 21h

call num\_to\_seg

mov R4, 01h

mov R5, 00h

; Включаем декодер

setb P0.7

; Настраиваем таймер

; Режим работы таймера: M1 = 1, M0 = 0 (3) для таймера 1

; Автоперезагружаемый таймер

; M1 = 0, M0 = 1 (2) для таймера 0

; 16-битный таймер

mov tmod, #21h

; Глобальные прерывания - включены

setb EA

; Прерывания от таймера 1 - включены

setb ET1

; Прерывания от таймера 0 - включены

setb ET0

; Таймер будет срабатывать

; каждые 10 отсчетов

mov TH1, #00F0h

mov TL1, #00F0h

; Включить таймер 1

setb TR1

;запуск таймера 0

setb TR0

while\_lp:

nop

jmp while\_lp

refreshLed:

; В регистре R3 храним

; номер индикатора который сейчас обновляется

; Номер индикатора поступит на декодер в битах 3, 4

refreshLed0:

cjne R3, #0, refreshLed1

; Выбираем индикатор 0

mov A, P3

; Очищаем биты 3,4

anl A, #0E7h

; Устанавливаем как надо

orl A, #000h

mov P3, A

mov P1, R4

jmp inc\_led

refreshLed1:

cjne R3, #1h, refreshLed2

; Выбираем индикатор 1

mov A, P3

anl A, #0E7h

orl A, #008h

mov P3, A

mov p1, R5

jmp inc\_led

refreshLed2:

cjne R3, #2h, refreshLed3

; Выбираем индикатор 2

mov A, P3

anl A, #0E7h

orl A, #010h

mov P3, A

mov p1, R6

jmp inc\_led

refreshLed3:

cjne R3, #3h, loop\_led

; Выбираем индикатор 3

mov A, P3

anl A, #0E7h

orl A, #018h

mov P3, A

mov p1, R7

jmp inc\_led

loop\_led:

mov R3, #0h

jmp refreshLed0

inc\_led:

inc R3

ret

;R2 - количество переполнений таймера

;15 переполнений - 1 секунда

; Остальное - в ОЗУ

;20H - счетчик секунд (0-59)

;21H - счетчик минут (0-59)

;22H - счетчик часов (0-23)

timer0\_interrupt:

inc R2

; Тут должно быть 15!!!

cjne R2,#1,to\_sec

mov R2,#0h

inc 20H

; Для секунд - каждый раз меняем состояние бита точки

mov A, R6

xrl A, #080h

mov R6, A

to\_sec:

mov A, 20H

; Тут должно быть 59!!

cjne A,#2,to\_min

mov 20H,#0h

inc 21H

to\_min:

mov A, 21H

; Тут должно быть 59!!

cjne A,#59,to\_hr

mov 21H,#0h

inc 22H

to\_hr:

mov A, 22H

; Тут должно быть 23!!

cjne A,#23,to\_ret

mov 22H,#0h

to\_ret:

; Установка нужной индикации

mov A, 21H

call num\_to\_seg

; Регистры R5, R4 - минуты

mov R4, 01h

mov R5, 00h

mov A, 22H

call num\_to\_seg

; Регистры R7,R6 - часы

mov R6, 01h

mov R7, 00h

ret

; Преобразовываем число в аккумуляторе в значение, выводимое на порт

; индикатора. Кладем результат в пару регистров R0-R1

num\_to\_seg:

mov B, #10

div AB

; Теперь в A лежит результат деления

; а в B остаток от деления

; Берем число по адресу из A+100h

mov DPTR, #0100h

MOVC A,@A+DPTR

; Инвертируем его

cpl A

mov R0, A

; Получаем теперь знак для второй цифры

mov A, B

MOVC A,@A+DPTR

; Инвертируем его

cpl A

mov R1, A

ret

; Преобразование числа в сегмент

org 0100h

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

**Результаты работы программы на симуляторе EdSim51DI:**

**Изображение выглядит как снимок экрана, черный, сторона, сидит

Автоматически созданное описание**

**Изображение выглядит как снимок экрана, черный, сторона, сидит

Автоматически созданное описание**Изображение выглядит как снимок экрана, черный, сторона, сидит

Автоматически созданное описаниеИзображение выглядит как снимок экрана, черный, сторона, сидит

Автоматически созданное описание