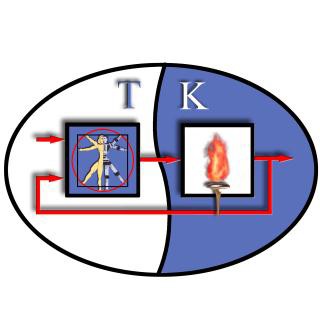
***МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ***



***ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ***

***ВЫСШЕГО ОБРАЗОВАНИЯ***

***«Белгородский Государственный Технологический Университет им. В.Г. Шухова» (БГТУ им. В.Г. Шухова)***

***ИЭИТУС***

***Кафедра Технической Кибернетики***

***Лабораторная работа № 2***

***дисциплина: «Микроконтроллеры в РТС»***

***тема: «Секундомер на микроконтроллере серии MCS51»***

***Выполнил:***

*студент группы МР-41 Рощук Р.Д.*

***Принял:***

*ст. преп. кафедры ТК*

*Гольцов Ю.А.*

*Белгород 2020*

**Выполнение**

Для подсчета прошедшего времени и вывода информации будем использовать прерывание от таймера 0 (16бит). При этом таймер 1 будем использовать для UART интерфейса (8бит с автозагрузкой значения из TH). Пусть скорость 9600бот:

9600=(2^smod\*Fosc)/(32\*12\*[256-TH])=31250/(256-TH)

TH1= #0FDh

В начальный момент выполняется настройка таймеров, последовательного порта и разрешается прерывание от таймера 0. После этого идет бесконечный цикл ожидания.

Срабатывание прерывания таймера 0 будет происходить раз в 10мс:

2^16-10000(11.059МГц/12тактов)=56320=0DC00h

TH0=0DCh; TL0=0h

Обработчик прерывания возвращает указанные выше значения TH и TL для поддержания 10мс периода опроса. Выводить данные будем в формате «12:34С», где 12 – прошедшие секунды, 34 – 340мс.

Для подсчета прошедшего времени будем использовать противоположное действие – сколько осталось времени (вызовов) до нуля. Поэтому в main мы загружаем

MOV 50h, #64h; 100 х10 – мс до обнуления;

MOV 51h, #3Ch; 60 секунд до обнуления.

В регистре 50h мы храним количество оставшихся 10мс интервалов до переполнения (и дальнейшего обнуления), а в 51h количество секунд до переполнения (обнуления) 60 секунд. Т.к. мы выводим только по одному символу, то нам при каждом вызове прерывания надо будет выделять из этих двух регистров десятки, единицы, десятые и сотые прошедшего времени. Так для секунд будем из прошедшего время выделять десятки как целое от деления на 10, а единицы как остаток от данного деления. ASCII коды полученных значений отправляем в параллельный порт. Прошедшее время получаем инверсией значения регистра 51h и вычитанием 195. Данное число обусловлено тем что в 51h значения будет меняться от 60 до 0, а при инверсии это будет от 195 до 255. Для приведения данного диапазона в 0-60 необходимо произвести коррекцию вычитанием 195. Для миллисекунд все аналогично, только значение коррекции 155.

После вывода уменьшаем количество оставшихся интервалов миллисекунд до наступления нуля используя DJNZ- декремент бита и переход на выход из прерывания если не нуль. При наступлении нуля нам необходимо произвести восстановление 100 интервалов и декремент для секунд. При 51h = 0 аналогично будет пред загружено 60 интервалов для секунд.

В конце происходит выход из прерывания.

**Блок схемы**

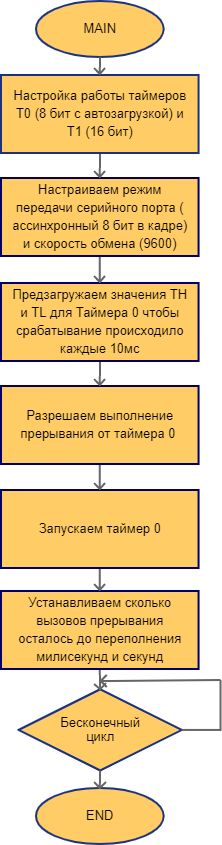


Рис 1. Main

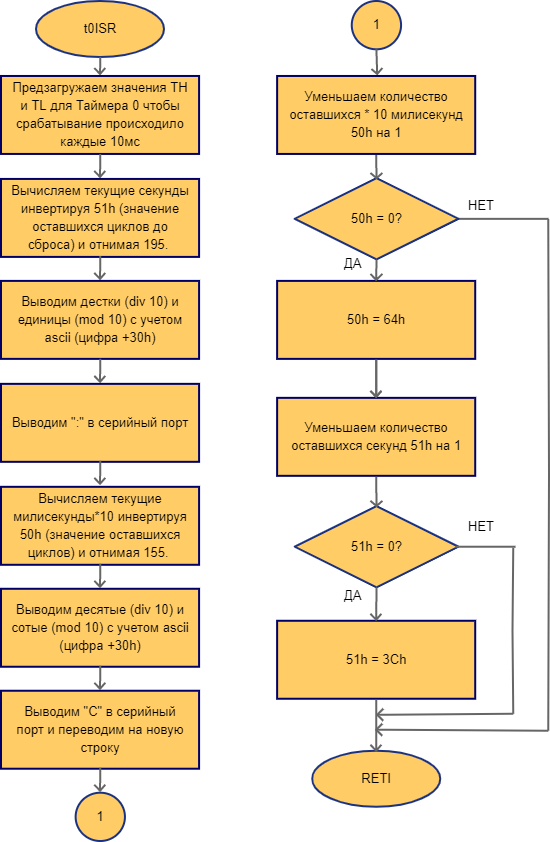


Рис 2. T0ISR (Прерывание от таймера 0)

**Пример работы программы**

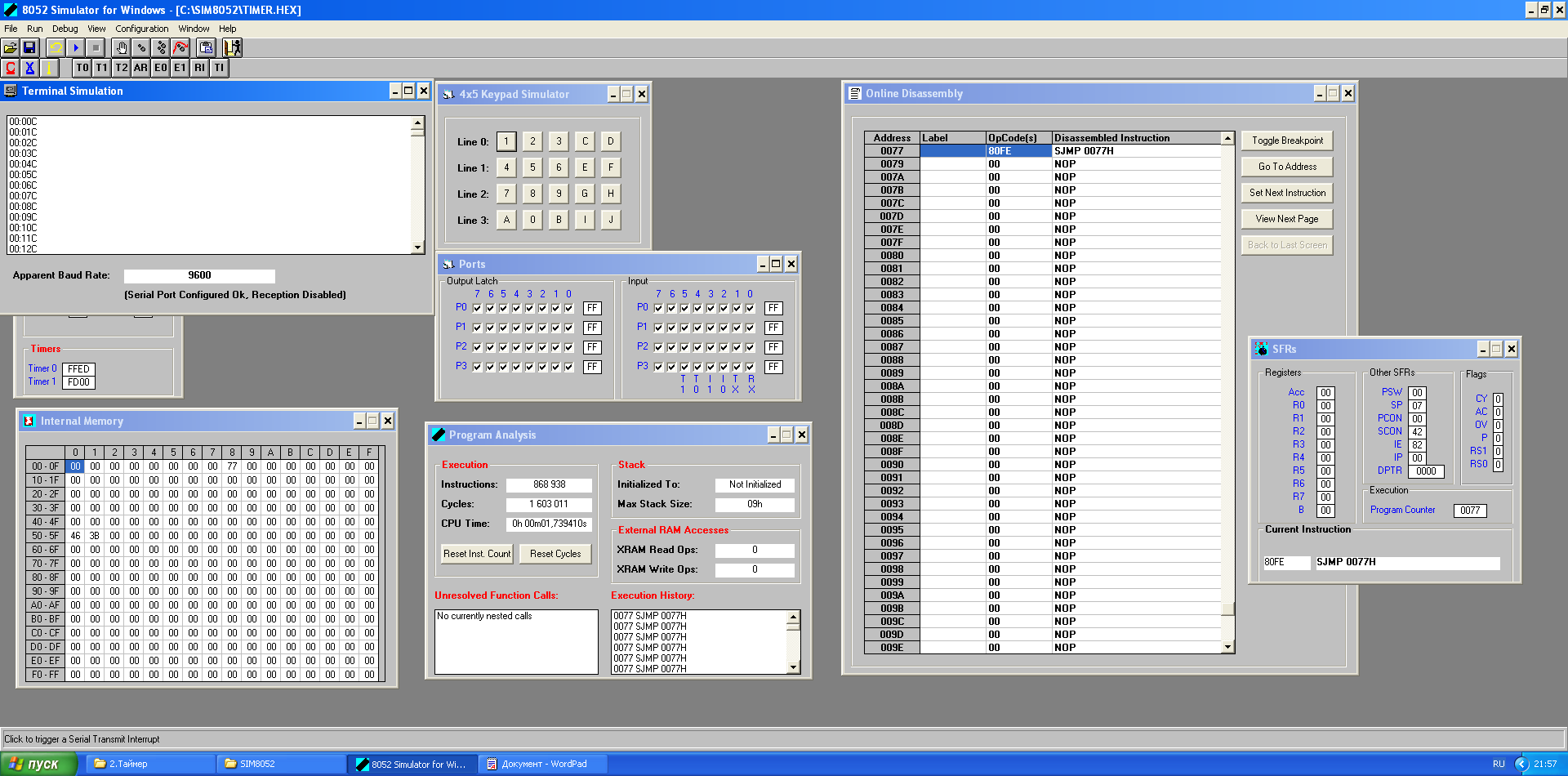


Рис 3. Старт работы секундомера

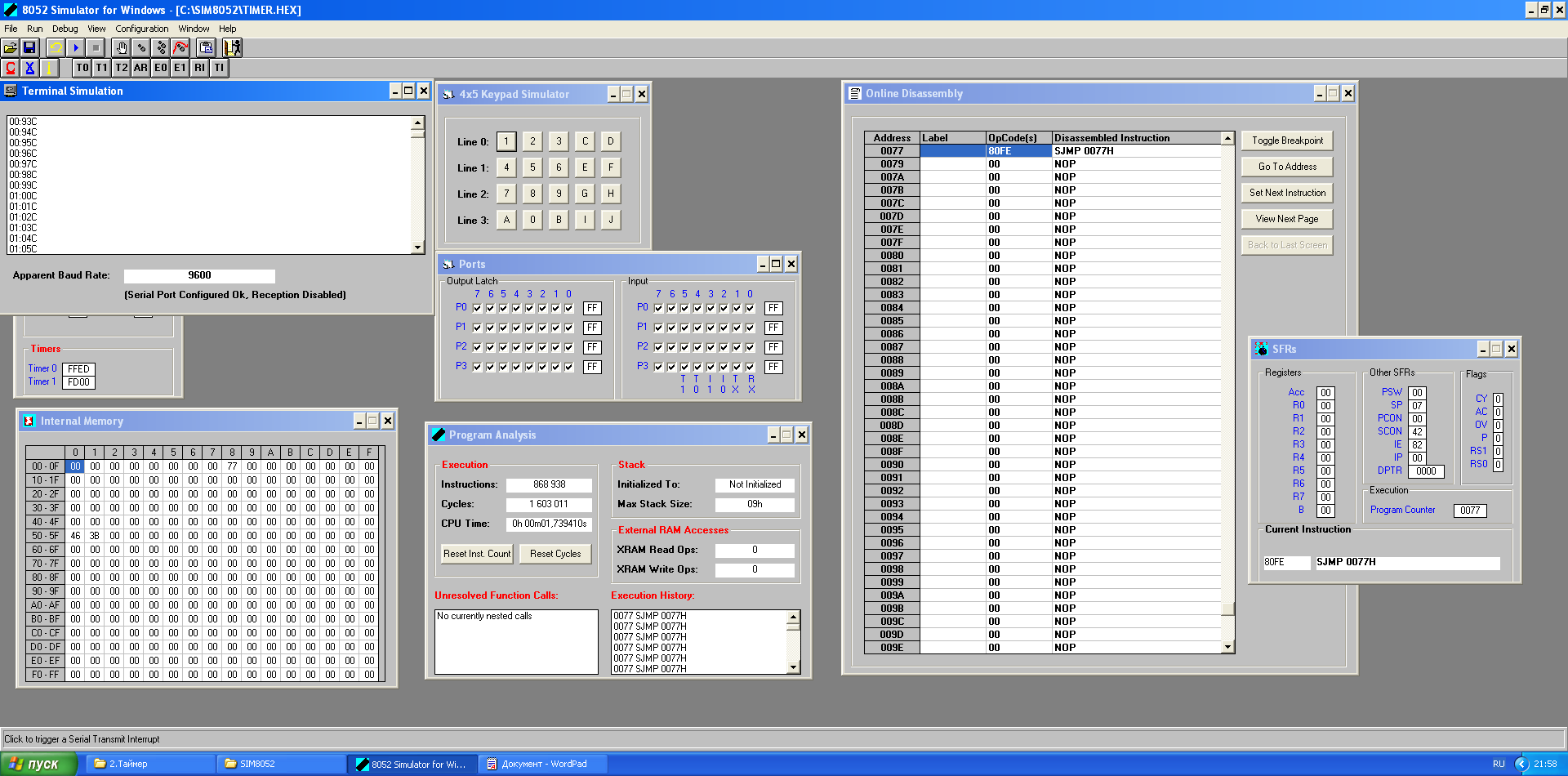


Рис 4. Начало второй секунды

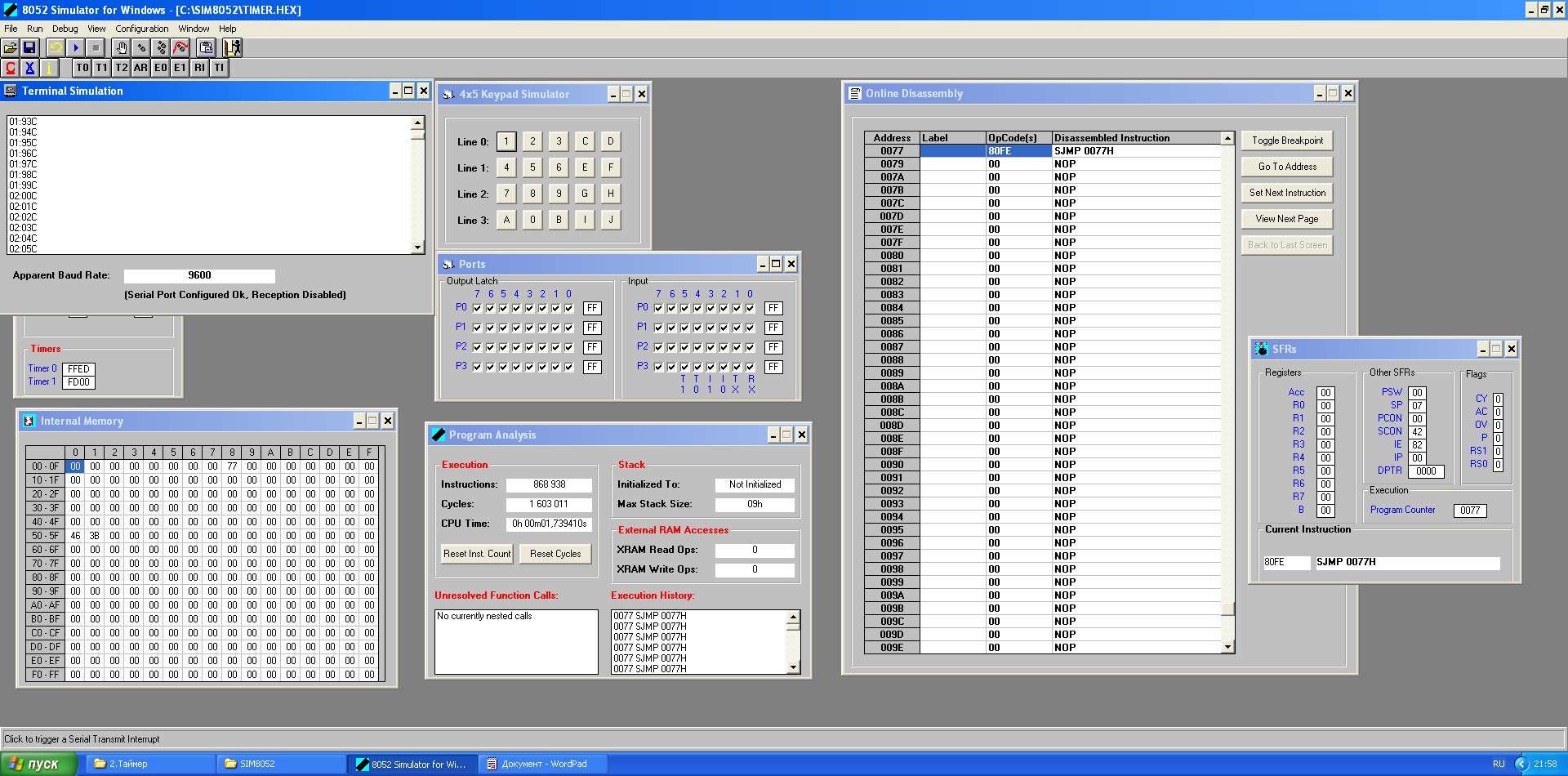


Рис 5. Окончание второй секунды

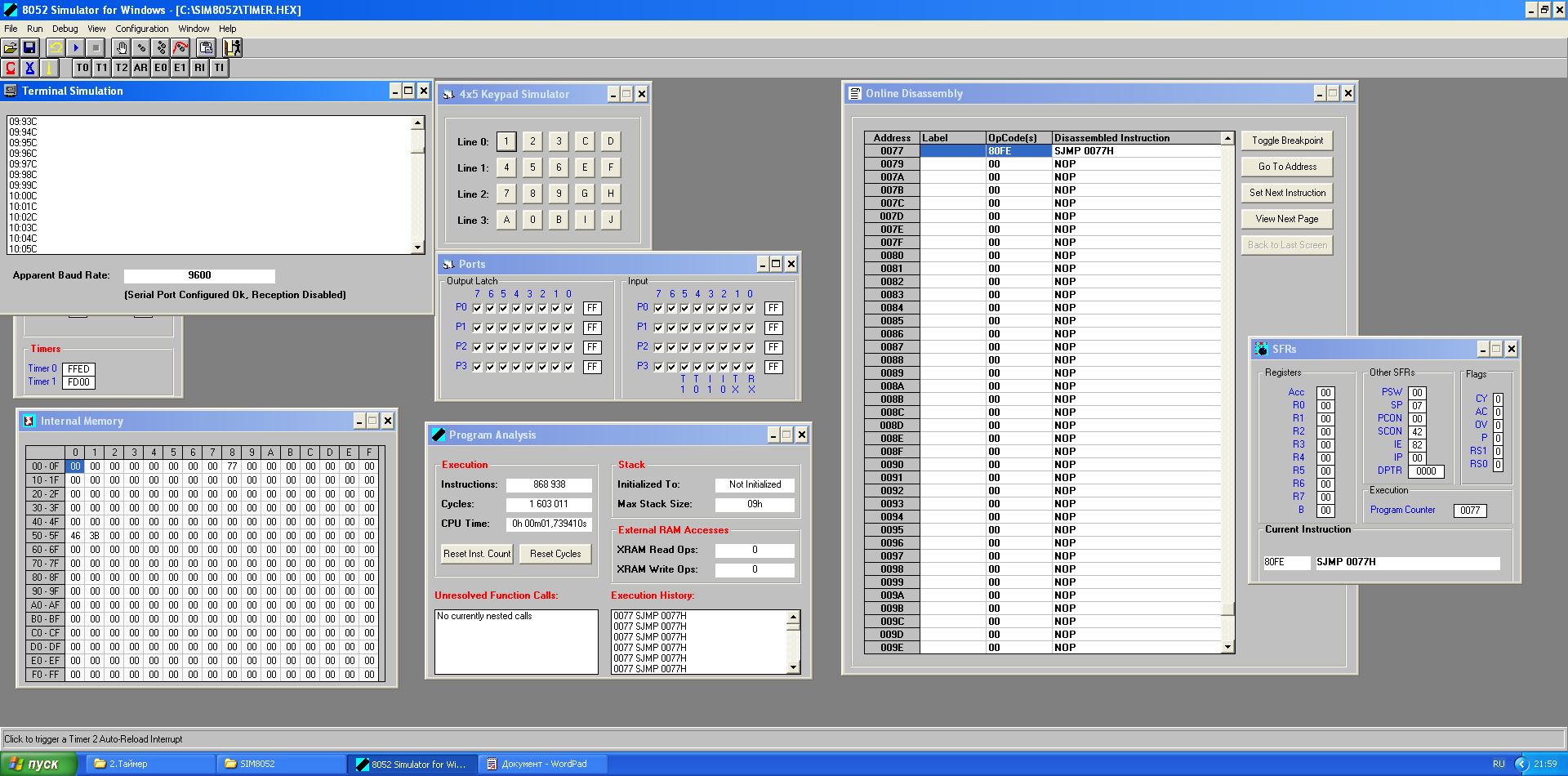


Рис 6. Переход через 10 секунд

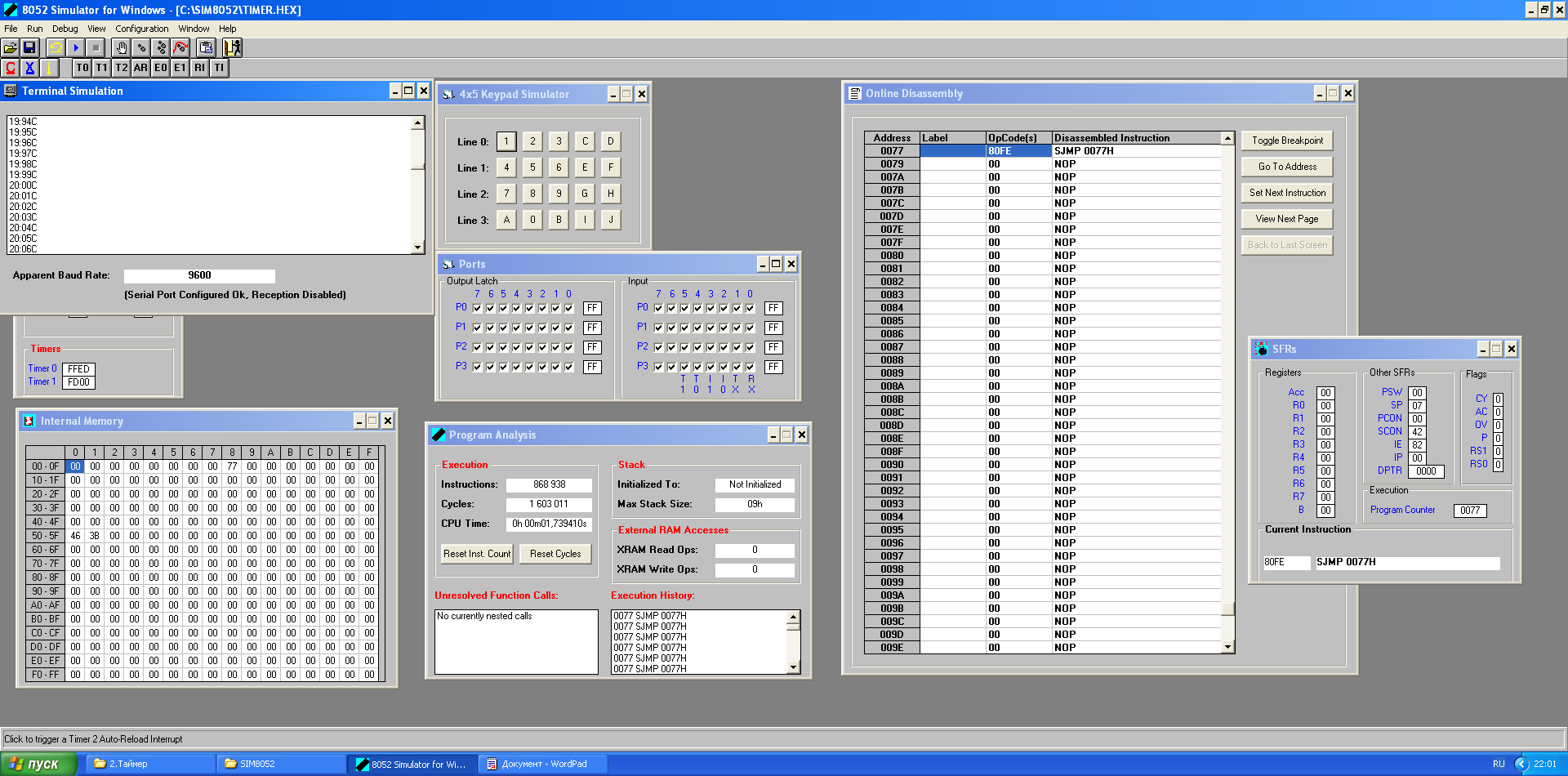


Рис 7. Переход через 20 секунд

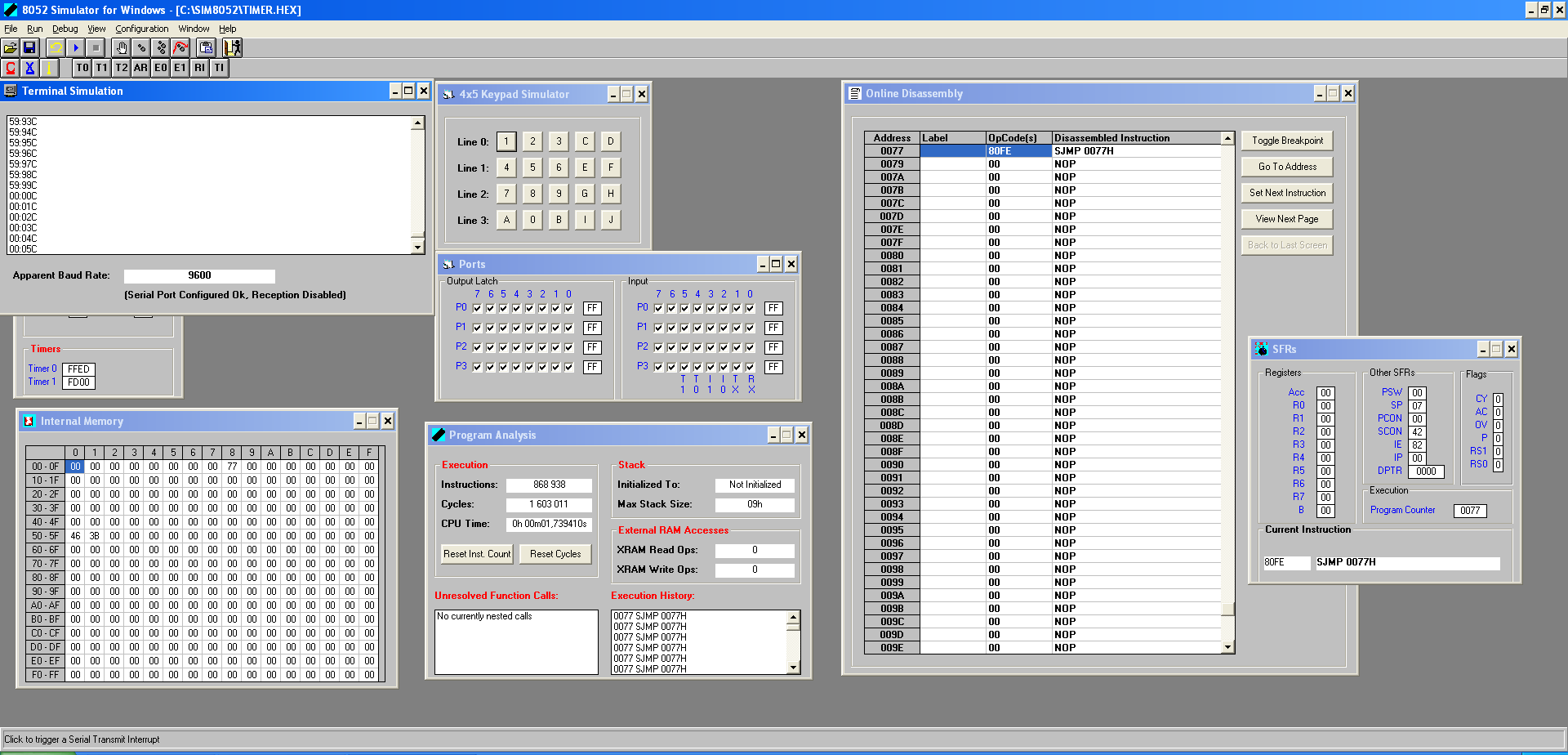


Рис 8. Переполнение секундомера

**Текст программы:**

ORG 0h

jmp main

t0ISR:

ORG 0Bh;Указываем где находится обработчик прерывания таймера 0

CLR TF0;Первой командой обработчика является команда очистки флага прерывания

MOV TH0,#0DCh;задаем смещение срабатывания таймера

MOV TL0,#00h

;Десятки

MOV A,51h

CPL A ;Инвертируем

SUBB A,#0C3h;195

MOV B, #0Ah

DIV AB

ADD A,#30h;Выводим цифру

MOV SBUF,A;Отправляем в порт

;Единицы

MOV A,B

ADD A,#30h;Выводим цифру

MOV SBUF,A;Отправляем в порт

MOV A,#3Ah;Выводим ":" между единицами и десятыми

MOV SBUF,A;Отправляем в порт

;Десятые

MOV A,50h

CPL A ;Инвертируем

SUBB A,#9Bh;155

MOV B, #0Ah

DIV AB

ADD A,#30h;Выводим цифру

MOV SBUF,A;Отправляем в порт

;Сотые

MOV A,B

ADD A,#30h;Выводим цифру

MOV SBUF,A;Отправляем в порт

MOV A,#43h;Выводим "C" в конце

MOV SBUF,A;Отправляем в порт

MOV A,#13

MOV SBUF,A

MOV A,#10

MOV SBUF,A

MOV A,#00h;Чтобы не ругался эмулятор

MOV B,#00h

DJNZ 50h,MS;Декремент прямоадресуемого байта х10мс и переход, если не нуль

;Если 0, то прошло 100 циклов по 10мс и нам надо востановить 100 циклов

MOV 50h, #64h

DJNZ 51h,MS;Уменьшаем на 1с и если не 0 переходим

MOV 51h, #3Ch;Когда переполнится 60 секунд сбрасываем

MS:

RETI;Выход из обработчика прерываний

main:

;T1 M1=1 M0=0 вывод в монитор порта

;T0 M1=0 M0=1 для подсчета времени/16бит

MOV TMOD,#00100001b

;передача с скоростью 9600=(2^smod\*Fosc)/(32\*12\*[256-TH])=31250/(256-TH)

MOV TH1,#0FDh

;1-й режим передачи (асинхронный 8 бит в кадре)

SETB SM1

CLR SM0

;обновление данных 100гц(10мс)

MOV TH0,#0DCh;2^16-10000(11.059/12)=56320=0DC00h

MOV TL0,#00h

;разрешаем прерывание от таймера 0

SETB ET0

;разрешаем выполнение прерываний

SETB EA

SETB TR0;запускаем таймер 0

MOV 50h, #64h;Настройка начальных значений секундомера

MOV 51h, #3Ch

jmp $;бесконечный цикл ожидания

END

**Листинг программы:**

MCS-51 MACRO ASSEMBLER TIMER 10/28/<0 PAGE 1

DOS 5.0 (038-N) MCS-51 MACRO ASSEMBLER, V2.3

OBJECT MODULE PLACED IN C:\SIM8052\TIMER.OBJ

ASSEMBLER INVOKED BY: C:\SIM8052\ASM51.EXE C:\SIM8052\TIMER.ASM

LOC OBJ LINE SOURCE

0000 1 ORG 0h

0000 02005B 2 jmp main

3 t0ISR:

000B 4 ORG 0Bh;Sj`g{b`el cde m`undhrq^ nap`anrwhj opep{b`mh^ r`ilep` 0

000B C28D 5 CLR TF0;Oepbni jnl`mdni nap`anrwhj` ^bk^erq^ jnl`md` nwhqrjh tk`c` opep{b`mh^

000D 758CDC 6 MOV TH0,#0DCh;g`d`el qleyemhe qp`a`r{b`mh^ r`ilep`

0010 758A00 7 MOV TL0,#00h

8 ;Deq^rjh

0013 E551 9 MOV A,51h

0015 F4 10 CPL A ;Hmbeprhpsel

0016 94C3 11 SUBB A,#0C3h;195

0018 75F00A 12 MOV B, #0Ah

001B 84 13 DIV AB

001C 2430 14 ADD A,#30h;B{bndhl vhtps

001E F599 15 MOV SBUF,A;Nrop`bk^el b onpr

16

17 ;Edhmhv{

0020 E5F0 18 MOV A,B

0022 2430 19 ADD A,#30h;B{bndhl vhtps

0024 F599 20 MOV SBUF,A;Nrop`bk^el b onpr

21

0026 743A 22 MOV A,#3Ah;B{bndhl ":" lefds edhmhv`lh h deq^r{lh

0028 F599 23 MOV SBUF,A;Nrop`bk^el b onpr

24

25 ;Deq^r{e

002A E550 26 MOV A,50h

002C F4 27 CPL A ;Hmbeprhpsel

002D 949B 28 SUBB A,#9Bh;155

002F 75F00A 29 MOV B, #0Ah

0032 84 30 DIV AB

0033 2430 31 ADD A,#30h;B{bndhl vhtps

0035 F599 32 MOV SBUF,A;Nrop`bk^el b onpr

33

34 ;Qnr{e

0037 E5F0 35 MOV A,B

0039 2430 36 ADD A,#30h;B{bndhl vhtps

003B F599 37 MOV SBUF,A;Nrop`bk^el b onpr

38

003D 7443 39 MOV A,#43h;B{bndhl "C" b jnmve

003F F599 40 MOV SBUF,A;Nrop`bk^el b onpr

0041 740D 41 MOV A,#13

0043 F599 42 MOV SBUF,A

0045 740A 43 MOV A,#10

0047 F599 44 MOV SBUF,A

0049 7400 45 MOV A,#00h;Wrna{ me psc`kq^ }lsk^rnp

004B 75F000 46 MOV B,#00h

004E D55009 47 DJNZ 50h,MS;Dejpelemr op^ln`dpeqselncn a`ir` u10lq h oepeund, eqkh me msk|

48 ;Eqkh 0, rn opnxkn 100 vhjknb on 10lq h m`l m`dn bnqr`mnbhr| 100 vhjknb

0051 755064 49 MOV 50h, #64h

0054 D55103 50 DJNZ 51h,MS;Slem|x`el m` 1q h eqkh me 0 oepeundhl

MCS-51 MACRO ASSEMBLER TIMER 10/28/<0 PAGE 2

LOC OBJ LINE SOURCE

0057 75513C 51 MOV 51h, #3Ch;Jncd` oepeonkmhrq^ 60 qejsmd qap`q{b`el

52 MS:

005A 32 53 RETI;B{und hg nap`anrwhj` opep{b`mhi

54 main:

55 ;T1 M1=1 M0=0 b{bnd b lnmhrnp onpr`

56 ;T0 M1=0 M0=1 dk^ ondqwer` bpelemh/16ahr

005B 758921 57 MOV TMOD,#00100001b

58 ;oeped`w` q qjnpnqr|~ 9600=(2^smod\*Fosc)/(32\*12\*[256-TH])=31250/(256-TH)

005E 758DFD 59 MOV TH1,#0FDh

60 ;1-i pefhl oeped`wh (`qhmupnmm{i 8 ahr b j`dpe)

0061 D29E 61 SETB SM1

0063 C29F 62 CLR SM0

63 ;namnbkemhe d`mm{u 100cv(10lq)

0065 758CDC 64 MOV TH0,#0DCh;2^16-10000(11.059/12)=56320=0DC00h

0068 758A00 65 MOV TL0,#00h

66 ;p`gpex`el opep{b`mhe nr r`ilep` 0

006B D2A9 67 SETB ET0

68 ;p`gpex`el b{onkmemhe opep{b`mhi

006D D2AF 69 SETB EA

006F D28C 70 SETB TR0;g`osqj`el r`ilep 0

0071 755064 71 MOV 50h, #64h;M`qrpnij` m`w`k|m{u gm`wemhi qejsmdnlep`

0074 75513C 72 MOV 51h, #3Ch

0077 80FE 73 jmp $;aeqjnmewm{i vhjk nfhd`mh^

74 END

MCS-51 MACRO ASSEMBLER TIMER 10/28/<0 PAGE 3

SYMBOL TABLE LISTING

------ ----- -------

N A M E T Y P E V A L U E A T T R I B U T E S

B. . . . . D ADDR 00F0H A

EA . . . . B ADDR 00A8H.7 A

ET0. . . . B ADDR 00A8H.1 A

MAIN . . . C ADDR 005BH A

MS . . . . C ADDR 005AH A

SBUF . . . D ADDR 0099H A

SM0. . . . B ADDR 0098H.7 A

SM1. . . . B ADDR 0098H.6 A

T0ISR. . . C ADDR 0003H A

TF0. . . . B ADDR 0088H.5 A

TH0. . . . D ADDR 008CH A

TH1. . . . D ADDR 008DH A

TL0. . . . D ADDR 008AH A

TMOD . . . D ADDR 0089H A

TR0. . . . B ADDR 0088H.4 A

REGISTER BANK(S) USED: 0

ASSEMBLY COMPLETE, NO ERRORS FOUND