



ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
СРЕДНЕГО (ПОЛНОГО) ОБЩЕГО ОБРАЗОВАНИЯ

ЛИЦЕЙ ПРИ СПБГУТ

Вендор-ориентированный учебный курс в системе
«Старшая профильно-профессиональная школа-ВУЗ-Работодатель»:
«Программирование микроконтроллеров Microchip»

Богураев М.В., Кисляков С.В.

«ПЕРИФЕРИЙНЫЙ МОДУЛЬ ПЕРВЫЙ ТАЙМЕР»

Методические указания к выполнению лабораторной работы

Санкт - Петербург
2009

Богураев М.В., Кисляков С.В. «ПЕРИФЕРИЙНЫЙ МОДУЛЬ ПЕРВЫЙ ТАЙМЕР». Методические указания к выполнению лабораторной работы №7(9). СПб: ГОУ «Лицей при СПбГУТ», 2009.

ЛАБОРАТОРНАЯ РАБОТА №7 «ПЕРИФЕРИЙНЫЙ МОДУЛЬ ПЕРВЫЙ ТАЙМЕР»

Цель работы

Научиться включать периферийные модули. Ознакомиться с работой периферийного модуля первый таймер. Освоить моделирование этого модуля в MPLAB IDE. Овладеть навыками программирования таймеров PIC микроконтроллеров.

Теоретические основы

Периферийные модули – это устройства, которые располагаются на одном кристалле с ядром микроконтроллера. Каждый периферийный модуль выполняет одну или несколько функций. Периферийные модули могут быть подключены к ядру микроконтроллера и к ножкам корпуса микроконтроллера. Настройка и подключение этих модулей производится программно.

Таймеры – это устройства, при помощи которых измеряют время. Принципиальной разницы между счётчиком и таймером нет. Счётчик производит подсчёт каждого нарастания напряжения на ножке микроконтроллера. Если сделать так, что нарастания напряжения будут происходить с равным периодом, то счётчик станет таймером.

Модуль TIMER1 может выполнять функцию счётчика или функцию таймера. Настройка первого таймера осуществляется в регистре T1CON. В модуле имеется делитель. Делитель позволяет увеличивать период (делить «частоту») импульсов, которые изменяют содержимое регистров TMR1H и TMR1L. Коэффициент деления устанавливается битами <T1CKPS1:T1CKPS0> в регистре T1CON.

Модуль TIMER1 является шестнадцатиразрядным и отображается в память данных как две ячейки памяти – TMR1H и TMR1L это отличие от нулевого таймера позволяет отсчитывать более длительные интервалы времени. Первый таймер может работать с импульсами, которые поступают на вход RC0/T1OSO/T1CKI. В отличие от нулевого таймера модуль первого таймера имеет цепь тактового генератора, к которой можно подключить кварцевый резонатор. Эта цепь рассчитана на подключение «часового кварца» - кварцевого резонатора с резонансной частотой 32768 Герц. Резонатор подключается между выводами T1OSI и T1OSO (RC1 и RC0 соответственно) и снабжается двумя конденсаторами. Если бит T1OSCEN в регистре T1CON установлен, то тактовый генератор модуля TIMER1 включается. После включения генератора, программист должен сделать программную задержку (паузу) для того, чтобы генератор запустился. Если бит TMR1ON установлен, содержимое регистра TMR1L будет увеличиваться на один на каждом импульсе, который придёт на вход RC0/T1OSO/T1CKI (если делитель установлен в положение 1:1). То есть источником тактовых импульсов будет служить генератор таймера, и таймер начнёт счёт, если TMR1ON=1.

Настройка прерываний от этого модуля осуществляется в регистрах INTCON, PIE1, а флаг прерывания устанавливается в регистре PIR1.

О включении и выборе режима можно прочитать в технической документации на микроконтроллер в файле, который доступен на сайте производителя (файл 39582b.pdf).

В этой лабораторной работе TIMER1 конфигурируется следующим образом: источник тактовых импульсов внешний (External clock) – генератор модуля TIMER1 (бит TMR1CS = 1). Импульсы внешнего генератора поступают на вход RC0/T1OSO/T1CKI. Синхронизации с тактовым генератором микроконтроллера нет (бит #T1SYNC = 1). Делитель имеет коэффициент деления 1:1. Это означает, что биты <T1CKPS1:T1CKPS0> в регистре T1CON должны иметь нулевые значения.

Чтобы пользоваться первым таймером, нужно сначала включить генератор (установить бит T1OSCEN в регистре T1CON). Затем настроить таймер – установить биты

#T1SYNC и TMR1CS, установить предделитель, записать значение 0x00 в TMR1L 0x80 в TMR1H для отсчёта секундного интервала и запустить счёт таймера (установить бит TMR1ON в регистре T1CON). Разрешить прерывание от него, установив биты TMR1IE в регистре PIE1 и биты PEIE и GIE в регистре INTCON.

Содержимое ячейки NUMBER_ONE будет увеличиваться на единицу раз в секунду по прерыванию от первого таймера. Чтобы добиться от первого таймера секундного интервала нужно после каждого прерывания от него записывать в регистр TMR1H число 0x80 – заполнять таймер наполовину, иначе прерывания будут происходить раз в две секунды. Это так, потому что максимальное число, которое можно разместить в двух регистрах первого таймера – это 0xFFFF. Оно же десятичное 65535 – или 2 раза (!) по 32768. А 32768 Герц – это частота кварцевого резонатора, который подключен к генератору первого таймера. Значит, за одну секунду будет совершено 32768 приращений, а потом будет выставлен флаг прерывания. Это как раз и требуется в нашей задаче.

Два семисегментных индикатора будут переключаться по прерыванию от нулевого таймера (как и в предыдущей работе, но подключены они будут к ножкам RC2, RC3). На один индикатор будет выводиться содержимое ячейки NUMBER_ONE, а на другой индикатор будет выводиться содержимое ячейки NUMBER_TWO.

Задание

Откройте файл 39582b.pdf, изучите раздел TIMER1. Заполните таблицу включения и настройки первого таймера. Создайте проект Project9, и откомпилируйте текст программы Project9. Запустите программу в симуляторе MPLAB IDE и промоделируйте работу программы в окне Watch, а каналы T1CKI, RC2, RC3 и PORTD в Logic Analyzer.

Подключите два семисегментных индикатора на лабораторном макете к порту PORTD микроконтроллера. Выводы DIGIT1 и DIGIT2 подключите к порту PORTC (RC2, RC3) микроконтроллера. Запрограммируйте микроконтроллер и продемонстрируйте работу программы на лабораторном макете.

Порядок выполнения

Откройте файл 39582b.pdf, изучите раздел TIMER1 – работа в режиме асинхронного счётчика. Заполните таблицу включения и настройки первого таймера (табл. 1).

Табл. 1. Включение и настройка первого таймера.

Название регистра	Номер банка	Адрес	Название, номер бита, значение бита.								Состояние после POR	Состояние после сброса
			7	6	5	4	3	2	1	0		
INTCON	0,1,2,3	0Bh, 8Bh, 10Bh, 18Bh	GIE	PEIE	T0IE	INTIE	RBIIE	T0IF	INTF	RBIIF	0000 000x	0000 000u
TICON	0	10h	-	-	T1CKPS1	T1CKPS0	T1OSCE	#T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
PIR1	0	0Ch	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
RIE1	1	8Ch	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
TMR1L	0	0Eh	Младший байт регистра модуля Timer1								xxxx xxxx	uuuu uuuu
TMR1H	0	0Fh	Старший байт регистра модуля Timer1								xxxx xxxx	uuuu uuuu

Примечание: x = неизвестно, u = не изменяется.

В папке Projects создайте папку Project9, в нее скопируйте файл Project9. Запустите MPLAB IDE и создайте проект Project9. Выберите отладчик MPLAB SIM. Затем из меню View откройте Simulator Logic Analyzer (рис. 1). В появившемся окне (рис. 2) нажмите кнопку Channels. Появится окно Configure Channels. Для отображения каналов T1CKI, RC2, RC3 нужно выделить добавляемый канал и нажать кнопку Add (рис. 3). Выберите каналы T1CKI, RC2, RC3 (Рис. 4). Теперь, для отображения содержимого регистра PORTD сформируем шину. Для этого в окне Configure Channels нажмем кнопку Configure Bus(s) и в появившемся окне Configure Bus (рис. 5) нажмём кнопку New Bus. Появится окно Bus Name. Введём туда название шины PORTD (рис. 6) и нажмём кнопку ОК. В окне Configure Bus выделим RD0 – RD7 зажав клавишу Shift, и нажмём кнопку Add. После этого шина будет сформирована (рис. 7), нажмём кнопку ОК. Остаётся только добавить шину для чего в окне Configure Channels выбираем PORTD и нажимаем кнопку Add (рис. 8) и затем кнопку ОК.

Теперь в окне Logic Analyzer будет отображаться состояние выводов T1CKI, RC2, RC3 и PORTD (Рис. 9).

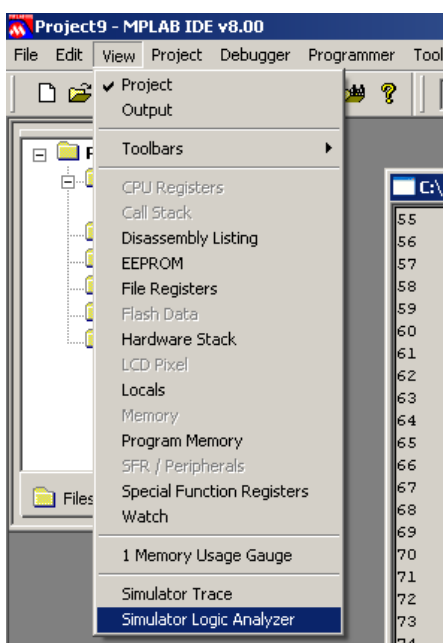


Рис. 1. Запуск логического анализатора.

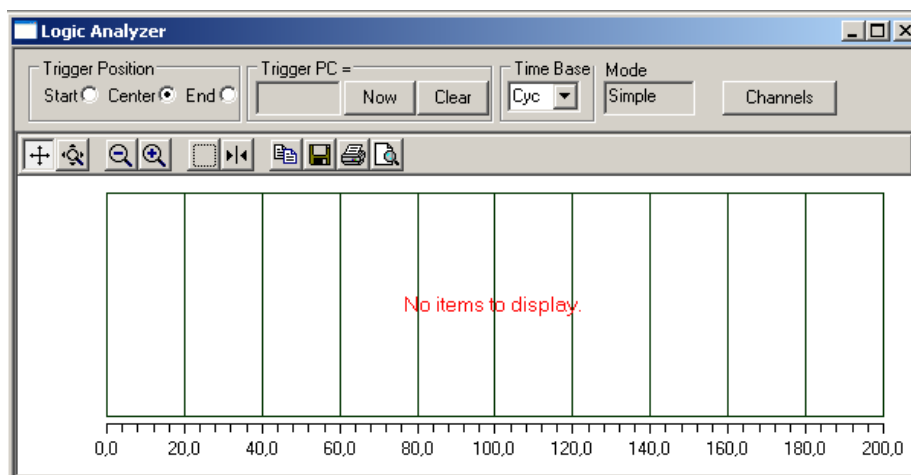


Рис. 2. Вид окна логического анализатора.

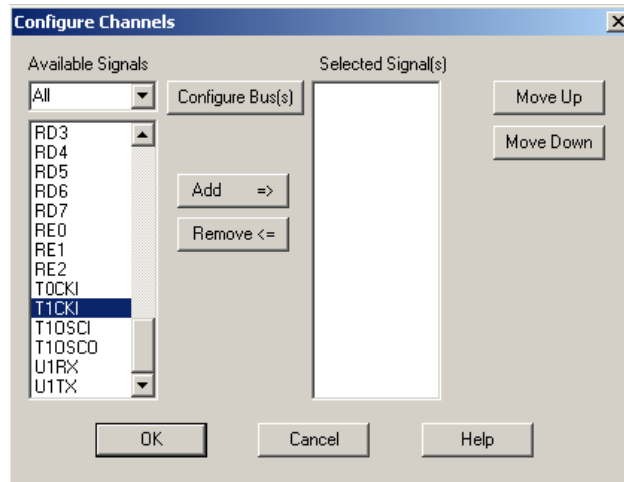


Рис. 3. Выбор каналов.

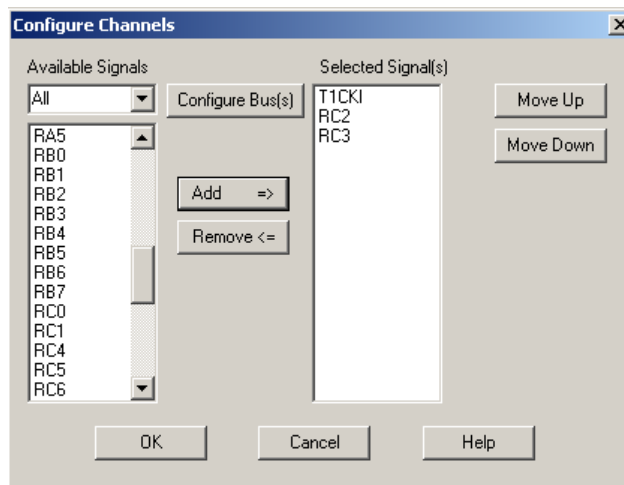


Рис. 4. Вид окна Configure Channels после выбора T1CKI, RC2 и RC3.

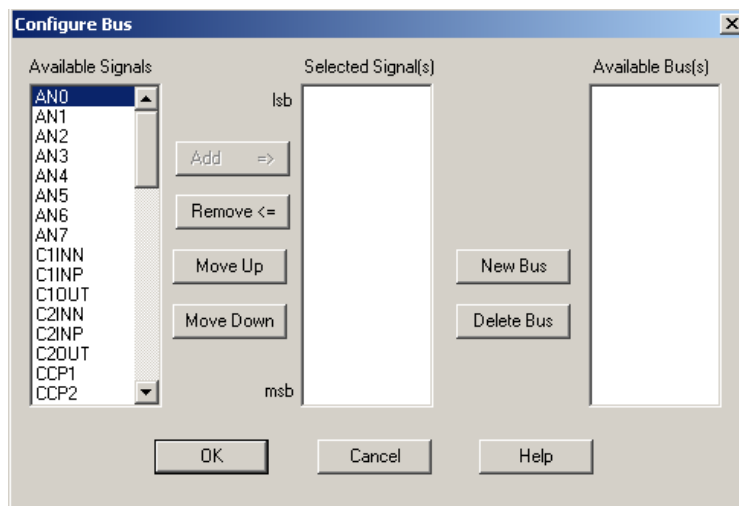


Рис. 5. Окно Configure Bus.

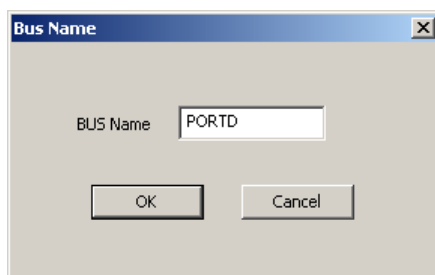


Рис. 6. Окна Bus Name для ввода названия шины.

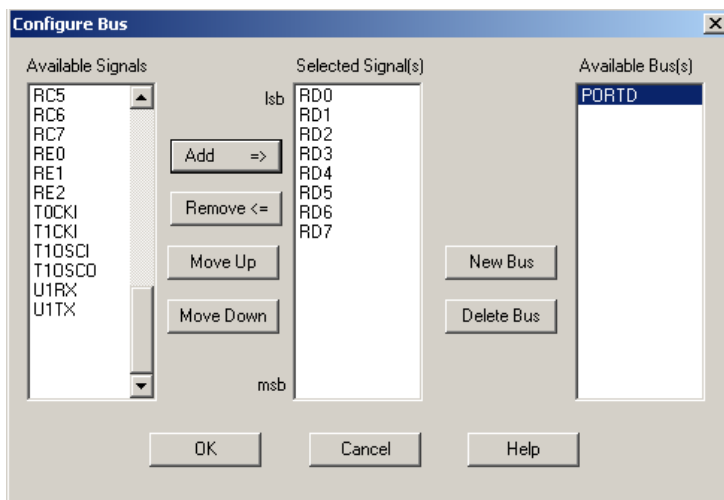


Рис. 7. Формирование шины PORTD.

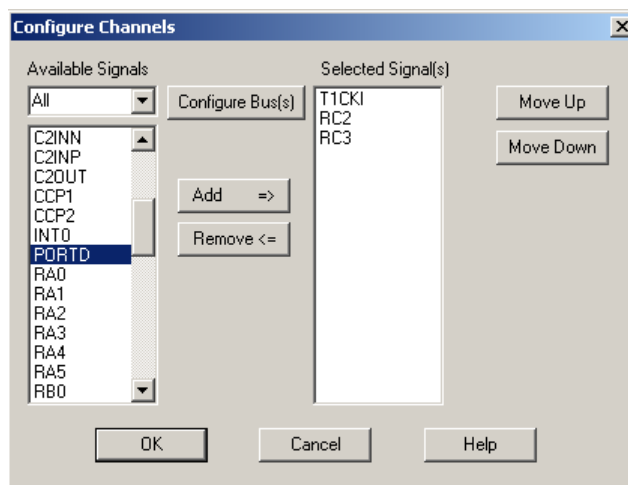


Рис. 8. Добавление сформированной шины PORTD.

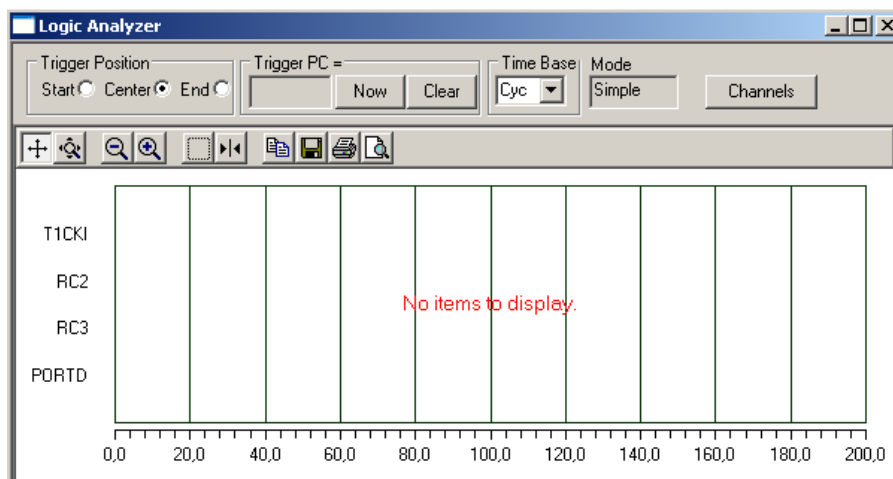


Рис. 9. Вид окна Logic Analyzer после настройки T1CKI, RC2, RC3 и PORTD.

Для симуляции работы модуля TIMER1 создадим файл стимулов (рис. 10). В появившемся окне стимулов выберем вкладку Clock Stimulus. В графе PIN выберем T1CKI (рис. 11). В графе Initial выберем значение Low – низкий уровень. В графах Low Cус и High Cус впишем единицы, в графе Begin выберем At Start, в графе End выберем Never (рис. 12). Эти установки не имеют абсолютно никакого отношения к реальному таймеру и выбраны исключительно исходя из соображений удобства симуляции, иначе симуляция займёт чрезмерно много времени. Рассмотрим значение этих установок: Initial – это значение, которое будет на ножке T1CKI в начале симуляции. Low Cус – длительность низкого уровня в машинных циклах. High Cус – длительность высокого уровня в машинных циклах. Begin – начало применения стимула. End – окончание применения стимула.

После того, как файл стимулов настроен, необходимо нажать кнопку Apply для применения текущих установок. В окне Output должно появиться сообщение об успешном применении стимулов. Если стимулы не применить, то симуляция не будет происходить.

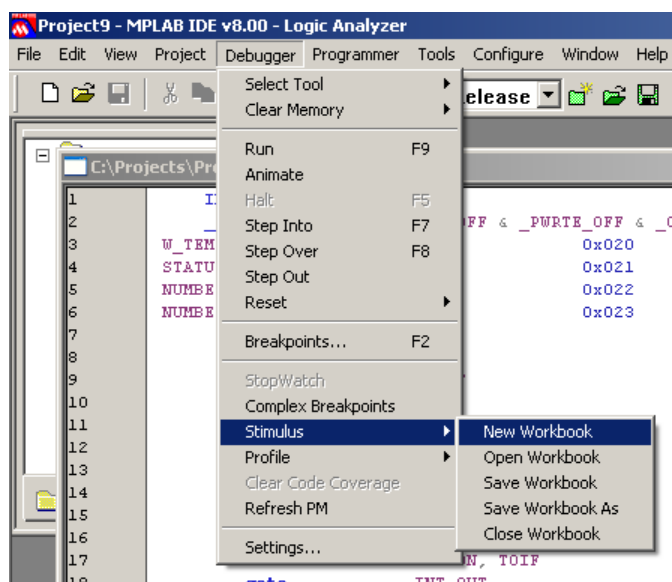


Рис. 10. Создание нового файла стимулов.

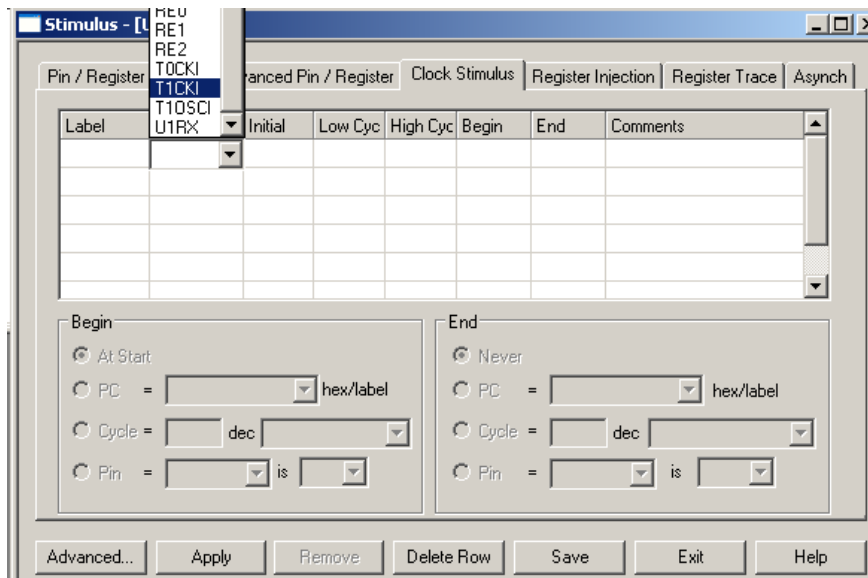


Рис. 11. Выбор симуляции напряжения на ножке T1CKI.

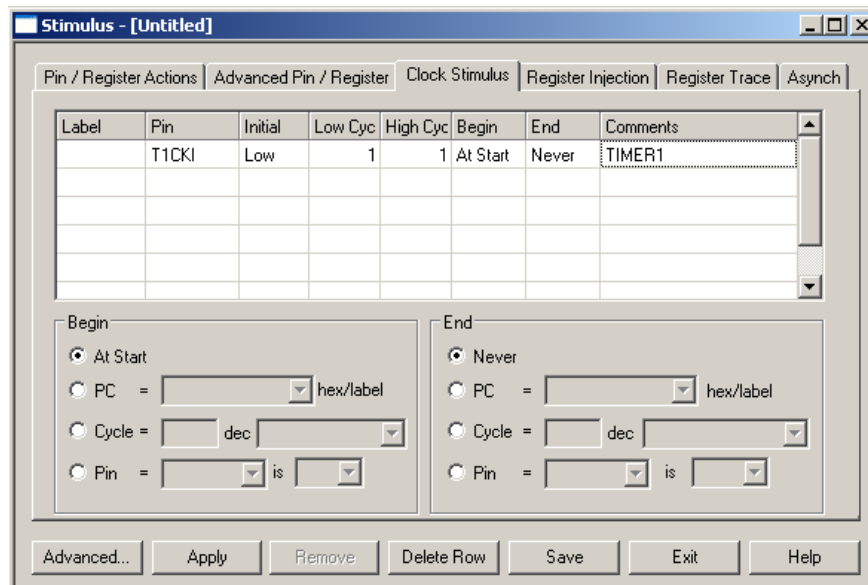


Рис. 12. Вид настроенного окна стимулов.

Чтобы симуляция происходила в удобном для наблюдения темпе (Debugger/Settings), выберем время одного шага в Animate режиме 1 миллисекунду, и поставим галочку напротив Enable Realtime watch updates (Рис. 14). Настроим окно Watch (View/Watch) для отображения регистров PORTC, PORTD, TMR1L, TMR1H и двух пользовательских регистров NUMBER_ONE и NUMBER_TWO (рис. 15). Кнопкой Animate запустим симулятор (рис. 16). После начала симуляции Logic Analyzer будет отображать состояние выводов T1CKI, RC2, RC3 и PORTD (Рис. 17). Изменения регистров PORTC, PORTD, TMR1H и TMR1H будут видны и в окне Watch.

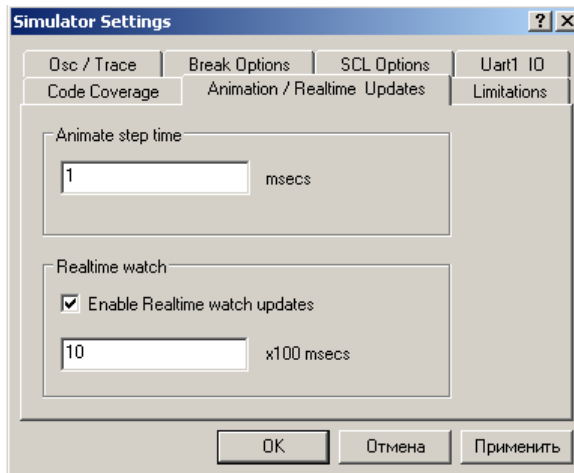


Рис. 14. Выбор времени одного шага в Animate режиме.

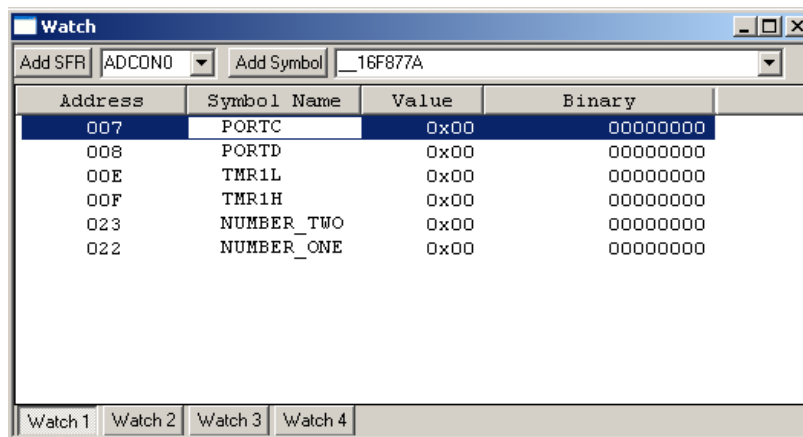


Рис. 15. Вид настроенного окна Watch.

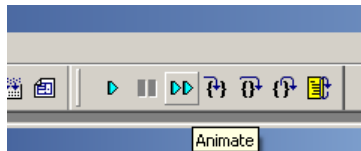


Рис. 16. Кнопка Animate.

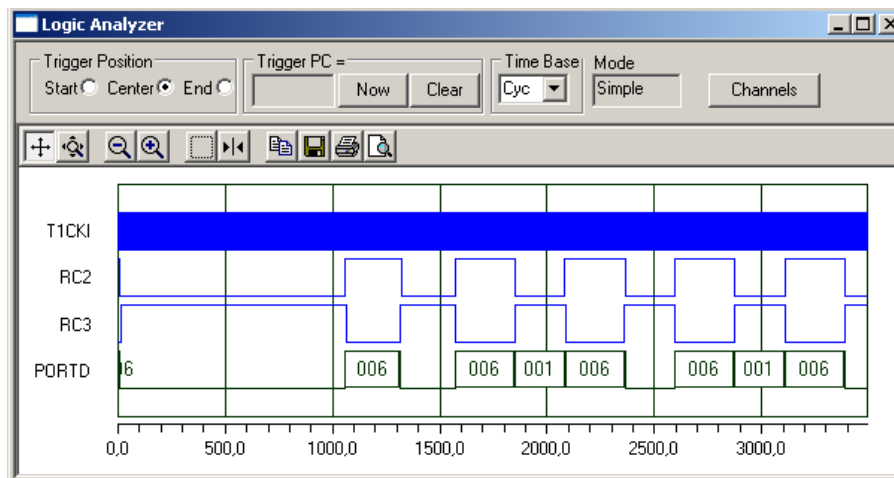


Рис. 17. Вид окна Logic Analyzer в процессе симуляции.

После того, как будет получено несколько периодов сигнала в окне Logic Analyzer, остановите симулятор кнопкой Halt (рис. 18).

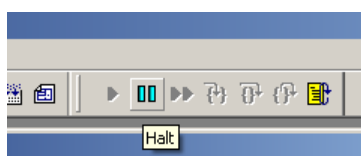


Рис. 18. Кнопка Halt для остановки симулятора.

Проанализируйте полученный результат: постарайтесь понять, что и когда происходит с пользовательскими регистрами NUMBER_ONE и NUMBER_TWO, какое число и когда будет подано на PORTD; какие при этом уровни будут на ножках RC2 и RC3. Предскажите результат работы программы на макете.

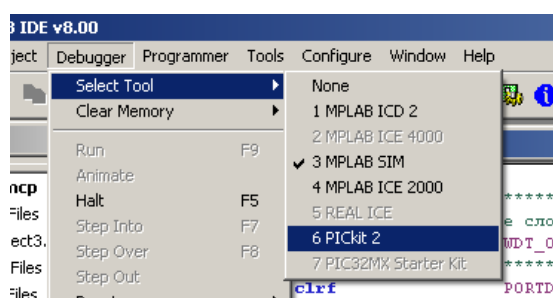


Рис. 19. Выбор отладчика в среде разработки.

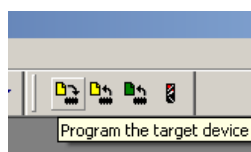


Рис. 20. Кнопка для программирования лабораторного макета.

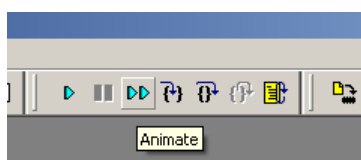


Рис. 21. Кнопка для запуска программы в режиме Animate.

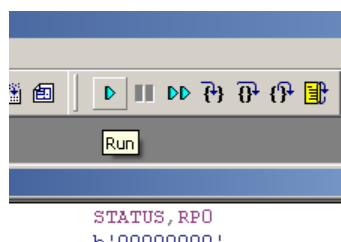


Рис. 22. Кнопка для запуска программы в режиме Run.

После симуляции измените текст программы: строки `movlw 0xFE` закомментируйте поставив `;` перед строкой, а строки `movlw 0x080` раскомментируйте убрав `;` После переделки программы соберите схему рис. 24 на лабораторном макете как показано на рис. 25. Когда схема собрана, выберите в среде разработки имеющийся у вас отладчик, например PICkit 2 (рис. 19). Запрограммируйте лабораторный макет (рис. 20). Запустите

программу в режиме Animate (рис. 21). Затем запустите программу в режиме RUN (рис. 22) на макете и предьявите результат.

Аппаратное обеспечение

В этой работе используется два семисегментных индикатора. Семисегментный индикатор состоит из восьми светодиодов и каждый из них работает как обычный светодиод. Только в семисегментном индикаторе светодиоды выполнены в виде полосок. Полоски имеют определённое положение на плоскости индикатора. Комбинируя горящие и не горящие светодиоды можно получать различные символы. Катоды у всех диодов объединены и подключаются к выводам PORTC. Принципиальная электрическая схема семисегментных индикаторов лабораторного макета изображена на рис. 23. Принципиальная электрическая схема для выполнения лабораторной работы изображена на рис. 24. На рис. 25 показаны соединения на плате лабораторного макета.

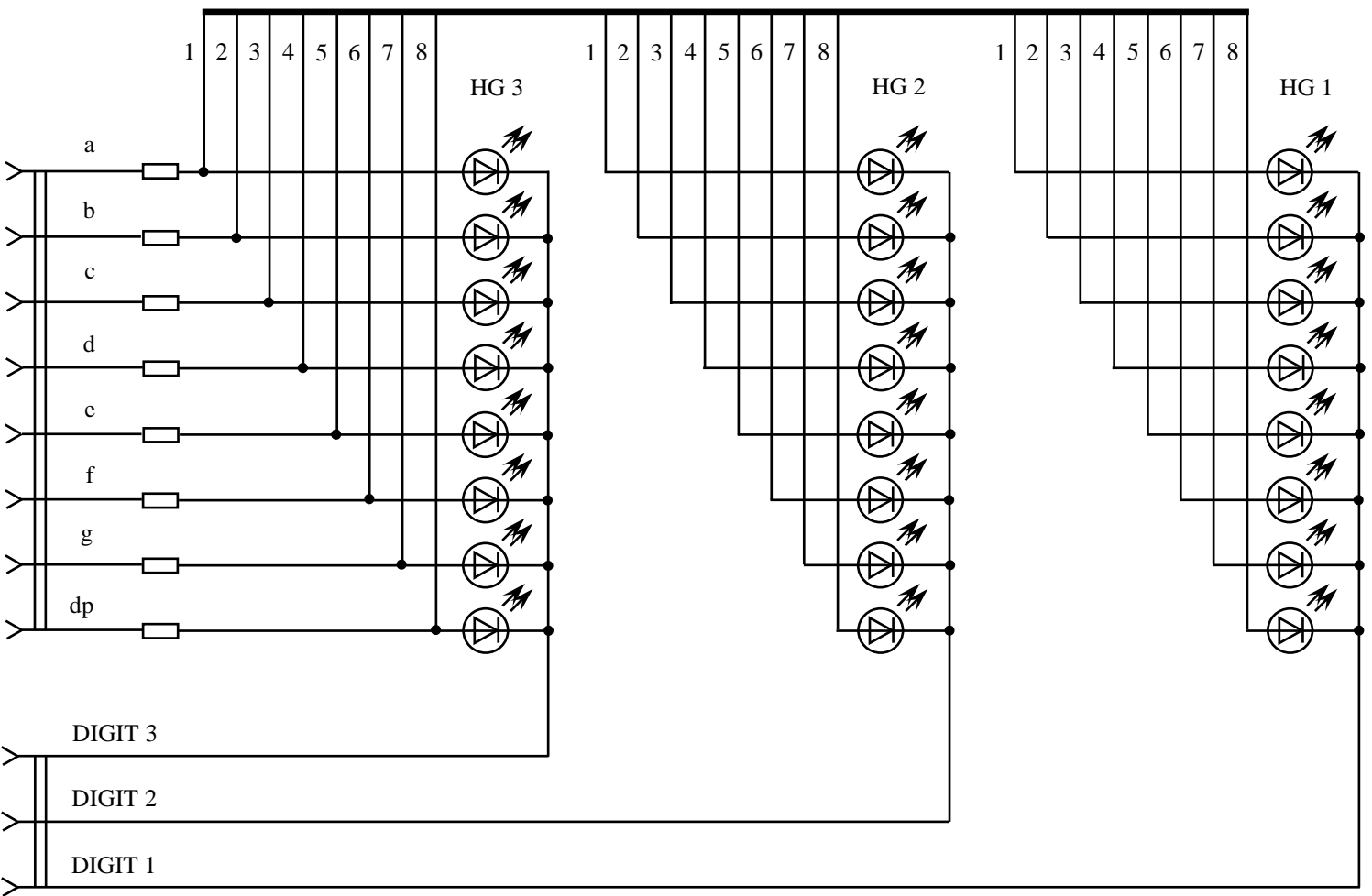


Рис. 23. Принципиальная электрическая схема семисегментных индикаторов.

DD1 PIC16F877A

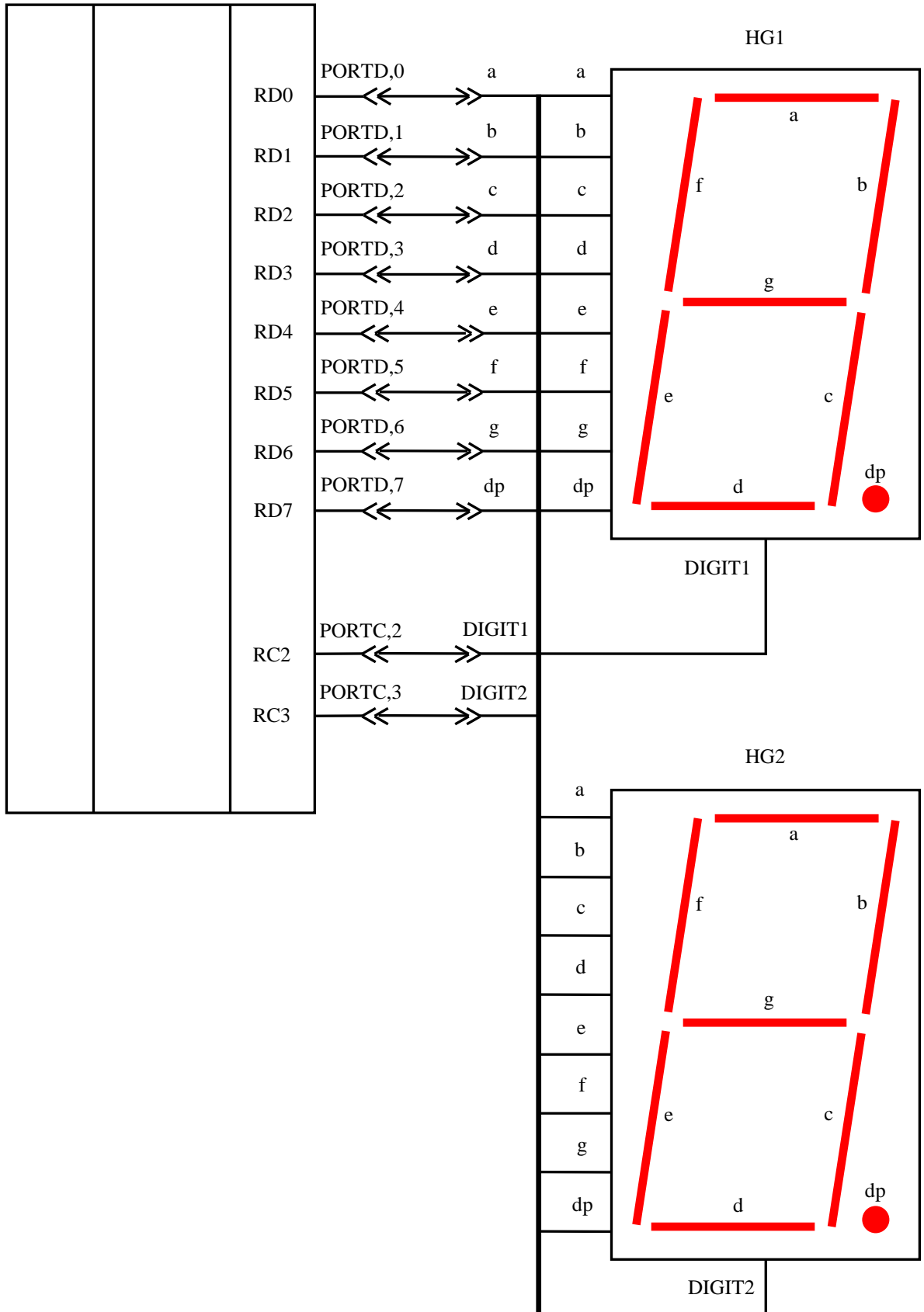


Рис. 24. Принципиальная электрическая схема для выполнения лабораторной работы.

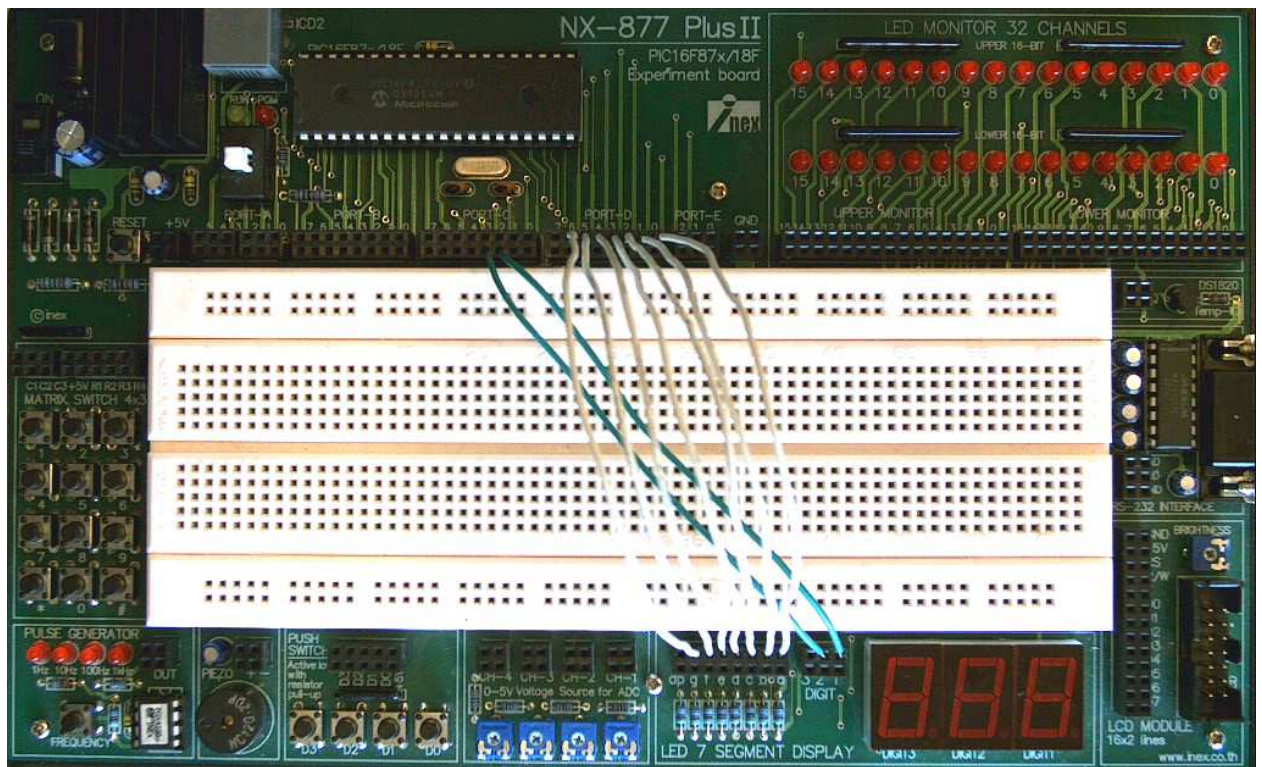


Рис. 25. Схема, собранная на лабораторном макете.

Программное обеспечение

Алгоритм программы изображён на рис. 26а и 26б.

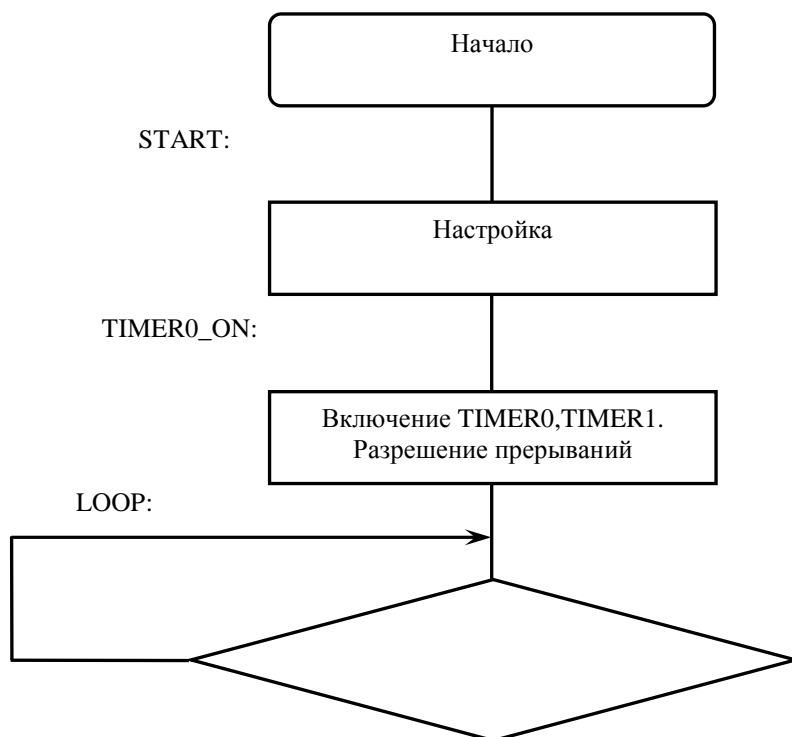


Рис. 26 а) Алгоритм основной программы.

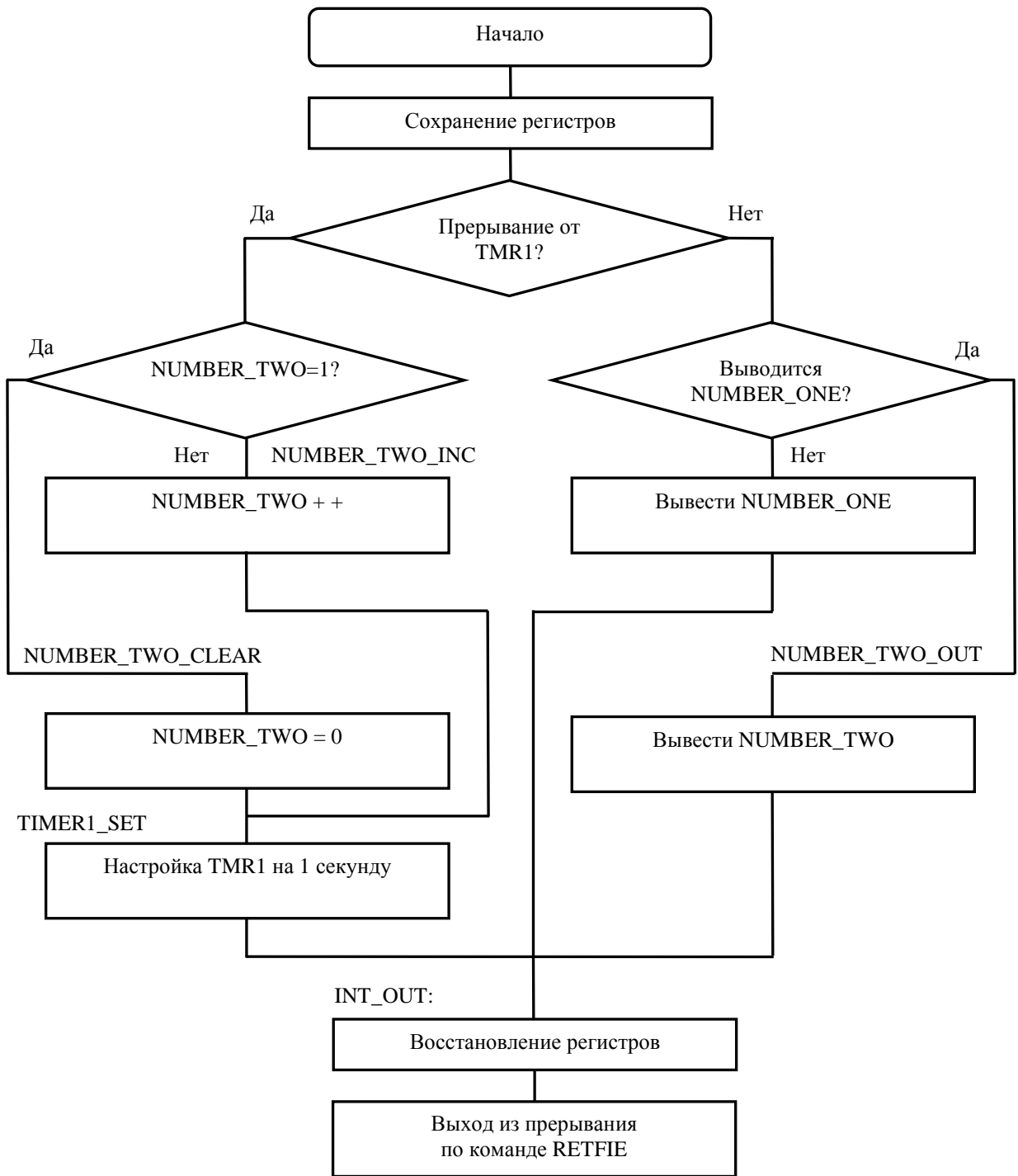


Рис. 26 б) Алгоритм обработчика прерывания.

Текст файла Project9.ASM

```
INCLUDE P16F877A.INC
__CONFIG _HS_OSC & _WDT_OFF & _PWRTE_OFF & _CP_OFF &
_BODEN_OFF & _LVP_OFF & _CPD_OFF & _DEBUG_ON
W_TEMP      equ 0x020
STATUS_TEMP equ 0x021
NUMBER_ONE  equ 0x022
NUMBER_TWO  equ 0x023
COUNTER     equ 0x024
org 00h
nop
goto START
org 0x04
movwf W_TEMP
swapf STATUS,W
clrf STATUS
movwf STATUS_TEMP
btfsc PIR1,TMR1IF
goto TMR1_INTERRUPT_STAFF
btfss INTCON, T0IF
goto INT_OUT
TMR0_INTERRUPT_STAFF:
bcf INTCON, T0IF
btfsc PORTC,2
goto NUMBER_TWO_OUT
bsf PORTC,2
movf NUMBER_ONE,W
movwf PORTD
bcf PORTC,3
goto INT_OUT
NUMBER_TWO_OUT:
bsf PORTC,3
movf NUMBER_TWO,W
movwf PORTD
bcf PORTC,2
goto INT_OUT
TMR1_INTERRUPT_STAFF:
bcf PIR1, TMR1IF
movlw 0x01
subwf NUMBER_TWO,W
btfsc STATUS,Z
goto NUMBER_TWO_CLEAR
goto NUMBER_TWO_INC
NUMBER_TWO_CLEAR:
clrf NUMBER_TWO
goto TIMER1_SET
NUMBER_TWO_INC
incf NUMBER_TWO,F
TIMER1_SET:
;movlw 0x080
movlw 0xFF
movwf TMR1H
```

```

INT_OUT:
    swapf STATUS_TEMP,W
    movwf STATUS
    swapf W_TEMP,F
    swapf W_TEMP,W
    retfie

START:
    clrf PORTC
    clrf PORTD
    bsf STATUS,RP0
    movlw 0x00
    movwf TRISD
    movlw b'00000011'
    movwf TRISC
    bcf STATUS,RP0
    bsf PORTC,3
    movlw b'00000110'
    movwf NUMBER_ONE
    movlw b'00000000'
    movwf NUMBER_TWO

TIMER1_ON:
    clrf TMR1L
    clrf TMR1H
    bsf T1CON, T1OSCEN
    movlw 0xFE
    movwf COUNTER

TMR1_LOOP:
    decfsz COUNTER
    goto TMR1_LOOP
    bsf T1CON, T1SYNC
    bsf T1CON, TMR1CS
    bsf T1CON, TMR1ON
;
    movlw 0x080
    movlw 0xFE
    movwf TMR1H

TIMER0_ON:
    clrf TMR0
    bsf STATUS, RP0
    bcf OPTION_REG,T0CS
    bcf STATUS, RP0

INTERRUPT_ENABLE:
    bsf INTCON, T0IE
    bsf INTCON, PEIE
    bsf STATUS, RP0
    bsf PIE1, TMR1IE
    bcf STATUS, RP0
    bsf INTCON, GIE

loop:
    goto loop
end

```

Индивидуальные задания

Напишите программу для подсчёта числа прерываний от первого таймера в диапазоне от 0 до 9. Одна из цифр показывает символ S.

Напишите программу для подсчёта числа прерываний от первого таймера в диапазоне от 0 до 59. Первая цифра показывает десятки, вторая цифра показывает единицы (секунд).

Контрольные вопросы

1. Назовите основные отличия нулевого таймера от первого?
2. Где можно прочитать о включении и выборе режима работы первого таймера?
3. Как пользоваться первым таймером?
4. Какие цифры надо записывать в регистры первого таймера для получения секундного интервала и почему?
5. Как сделать подсчёт интервалов времени вместо моргания второго индикатора?

Оглавление:

ЛАБОРАТОРНАЯ РАБОТА №7 «ПЕРИФЕРИЙНЫЙ МОДУЛЬ ПЕРВЫЙ ТАЙМЕР» ...	3
Цель работы	3
Теоретические основы	3
Задание.....	4
Порядок выполнения.....	5
Аппаратное обеспечение	13
Программное обеспечение.....	17
Индивидуальные задания	21
Контрольные вопросы.....	21
Список рисунков:	
Рис. 1. Запуск логического анализатора.....	6
Рис. 2. Вид окна логического анализатора.....	6
Рис. 3. Выбор каналов.	7
Рис. 4. Вид окна Configure Channels после выбора T1CKI, RC2 и RC3.....	7
Рис. 5. Окно Configure Bus.....	7
Рис. 6. Окна Bus Name для ввода названия шины.....	8
Рис. 7. Формирование шины PORTD.	8
Рис. 8. Добавление сформированной шины PORTD.	8
Рис. 9. Вид окна Logic Analyzer после настройки T1CKI, RC2, RC3 и PORTD.....	9
Рис. 10. Создание нового файла стимулов.	9
Рис. 11. Выбор симуляции напряжения на ножке T1CKI.....	10
Рис. 12. Вид настроенного окна стимулов.	10
Рис. 14. Выбор времени одного шага в Animate режиме.....	11
Рис. 15. Вид настроенного окна Watch.....	11
Рис. 16. Кнопка Animate.....	11
Рис. 17. Вид окна Logic Analyzer в процессе симуляции.	11
Рис. 18. Кнопка Halt для остановки симулятора.	12
Рис. 19. Выбор отладчика в среде разработки.	12
Рис. 20. Кнопка для программирования лабораторного макета.	12
Рис. 21. Кнопка для запуска программы в режиме Animate.....	12
Рис. 22. Кнопка для запуска программы в режиме Run.....	12
Рис. 23. Принципиальная электрическая схема семисегментных индикаторов.	14
Рис. 24. Принципиальная электрическая схема для выполнения лабораторной работы. ...	15
Рис. 25. Схема, собранная на лабораторном макете.	16
Рис. 26 а) Алгоритм основной программы.....	17
Рис. 26 б) Алгоритм обработчика прерывания.....	18

