



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 НУЛЕВОЙ ТАЙМЕР»

Методические указания к выполнению лабораторной работы

Санкт - Петербург  
2009

Богураев М.В., Кисляков С.В. «ПЕРИФЕРИЙНЫЙ МОДУЛЬ НУЛЕВОЙ ТАЙМЕР». Методические указания к выполнению лабораторной работы №6(8). СПб: ГОУ «Лицей при СПбГУТ», 2009.

## ЛАБОРАТОРНАЯ РАБОТА №6 «ПЕРИФЕРИЙНЫЙ МОДУЛЬ НУЛЕВОЙ ТАЙМЕР»

### Цель работы

Научиться включать периферийные модули. Ознакомиться с работой периферийного модуля нулевой 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. Модуль `TIMER0` может настраиваться и выполнять функцию счётчика или таймера. Счётчик `TIMER0` производит подсчёт каждого нарастания (или спада) напряжения на ножке `RA4/T0CKI`. Если сделать так, что нарастание или спад напряжения будут происходить с равным периодом, то счётчик станет таймером.

О включении и выборе режима можно прочитать в технической документации на микроконтроллер в файле, который доступен на сайте производителя (файл 39582b.pdf). Выбор режима работы нулевого таймера осуществляется битом `T0CS` в регистре `OPTION_REG`. Если бит `T0CS` сброшен, то модуль `TIMER0` работает в режиме таймера. И содержимое регистра `TMR0` будет увеличиваться на единицу на каждом машинном цикле. То есть источником тактовых импульсов будет служить основной генератор микроконтроллера. К таймеру можно подключить предделитель. Предделитель позволяет увеличивать период (делить «частоту») импульсов, которые изменяют содержимое регистра `TIMER0`. Коэффициент деления устанавливается битами `<PS2:PS0>` в регистре `OPTION_REG`. Нулевой таймер начинает счёт, если `T0CS=0` (или когда на его вход поступают тактовые импульсы)

В этой лабораторной работе `TIMER0` конфигурируется следующим образом: источник тактовых импульсов – основной генератор микроконтроллера (`T0CS=0`). Предделитель не подключен к `TIMER0` (`PSA=1`).

Чтобы пользоваться нулевым таймером его нужно настроить, запустить и разрешить прерывание от него, установив бит `TMR0IE` в регистре `INTCON`.

По прерыванию от нулевого таймера катод одного, а затем другого семисегментного индикатора будет соединяться через `PORTC` с общим проводом источника напряжения. Когда подключается один индикатор, выводится одна цифра, а когда подключается другой индикатор, выводится другая цифра. Каждое число отображается лишь часть времени. Поскольку человеческий глаз не замечает быстрой смены образов, то создаётся впечатление, что индикаторы отображают одновременно, хотя на самом деле они горят по очереди. Такой способ получил название динамической индикации.

### Задание

Откройте файл 39582b.pdf, изучите раздел `TIMER0`. Заполните таблицу включения и настройки нулевого таймера. Создайте проект `Project8`, и откомпилируйте текст программы `Project8`. Запустите программу в симуляторе MPLAB IDE и промоделируйте работу программы в окне `Watch`, а переключение каналов `RC0`, `RC1` в `Logic Analyzer`.

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

### Порядок выполнения

Откройте файл 39582b.pdf, изучите раздел TIMER0. Заполните таблицу включения и настройки нулевого таймера (табл. 1).

Табл. 1. Включение и настройка нулевого 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	INTEN	RBIEN	T0IF	INTF	RBIF	0000 000x	0000 000u
OPTION_REG	1, 3	81h, 181h	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
TMR0	0, 2	01h, 101h	Регистр модуля Timer0								xxxx xxxx	uuuu uuuu
			0	0	0	0	0	0	0	0		

Примечание: x = неизвестно, u = не изменяется.

В папке Projects создайте папку Project8, в нее скопируйте файл Project8. Запустите MPLAB IDE и создайте проект Project8. 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 RC0, RC1 нужно выделить добавляемый канал и нажать кнопку Add (рис. 3). Выберите каналы RC0 и RC1 (Рис. 4) и нажмите кнопку ОК. Теперь в окне Logic Analyzer будет отображаться состояние выводов RC0 и RC1 (Рис. 5). Чтобы симуляция происходила в удобном для наблюдения темпе, выберем время одного шага в Animate режиме 1 миллисекунду, и поставим галочку напротив Enable Realtime watch updates (Рис. 6). Кнопкой Animate запустим симулятор (рис. 7). После начала симуляции Logic Analyzer будет отображать состояние выводов RC0 и RC1 (Рис. 8). Изменения регистров PORTC, PORTD и TMR0 будут видны и в окне Watch (Рис. 9).

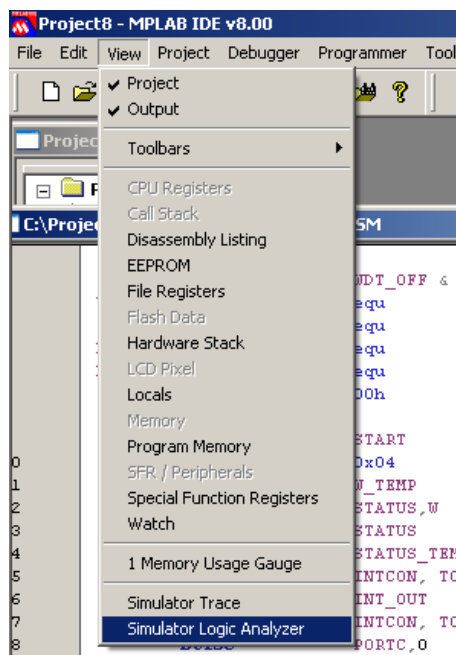


Рис. 1. Запуск логического анализатора.

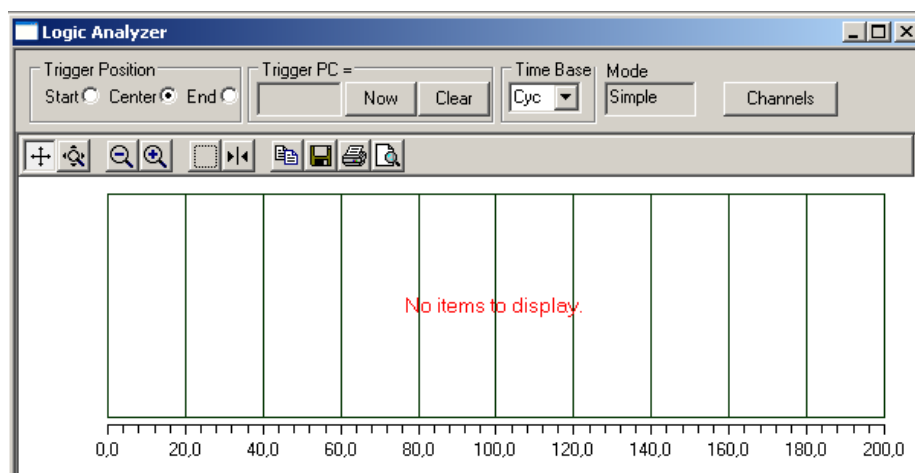


Рис. 2. Вид окна логического анализатора.

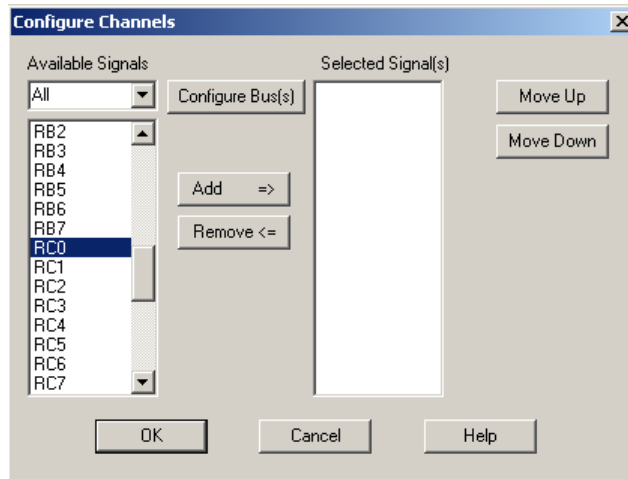


Рис. 3. Выбор каналов.

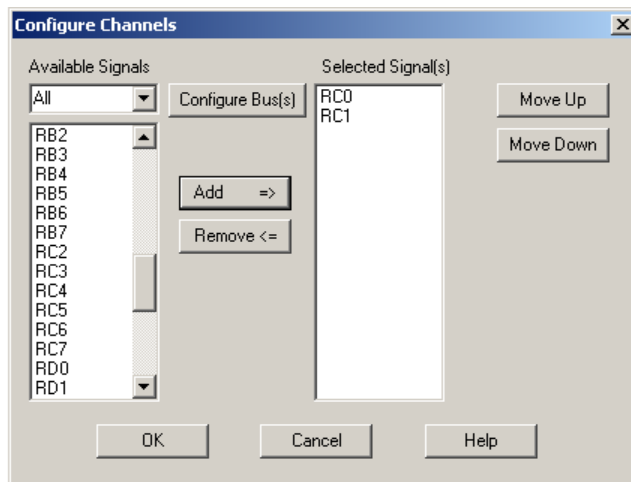


Рис. 4. Вид окна Configure Channels после выбора RC0 и RC1.

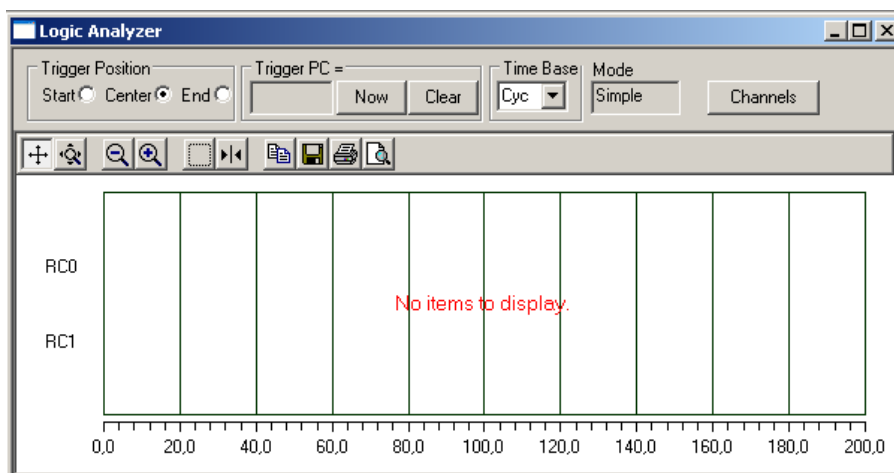


Рис. 5. Вид окна Logic Analyzer после настройки RC0 и RC1.

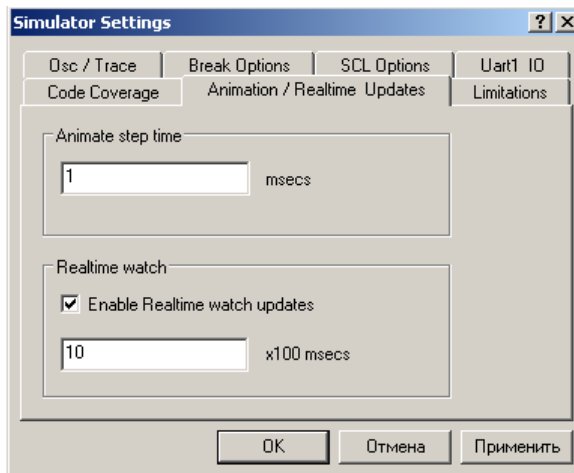


Рис. 6. Выбор времени одного шага в Animate режиме.

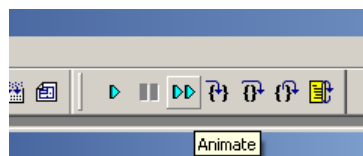


Рис. 7. Кнопка Animate.

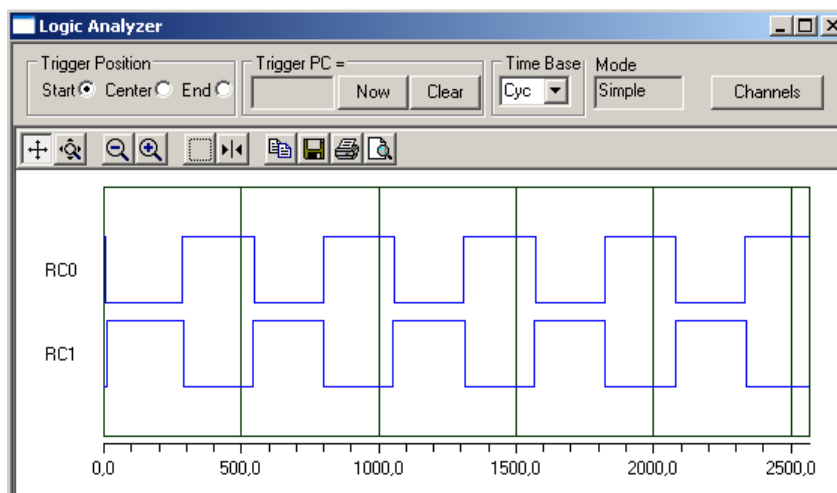


Рис. 8. Вид окна Logic Analyzer в процессе симуляции.

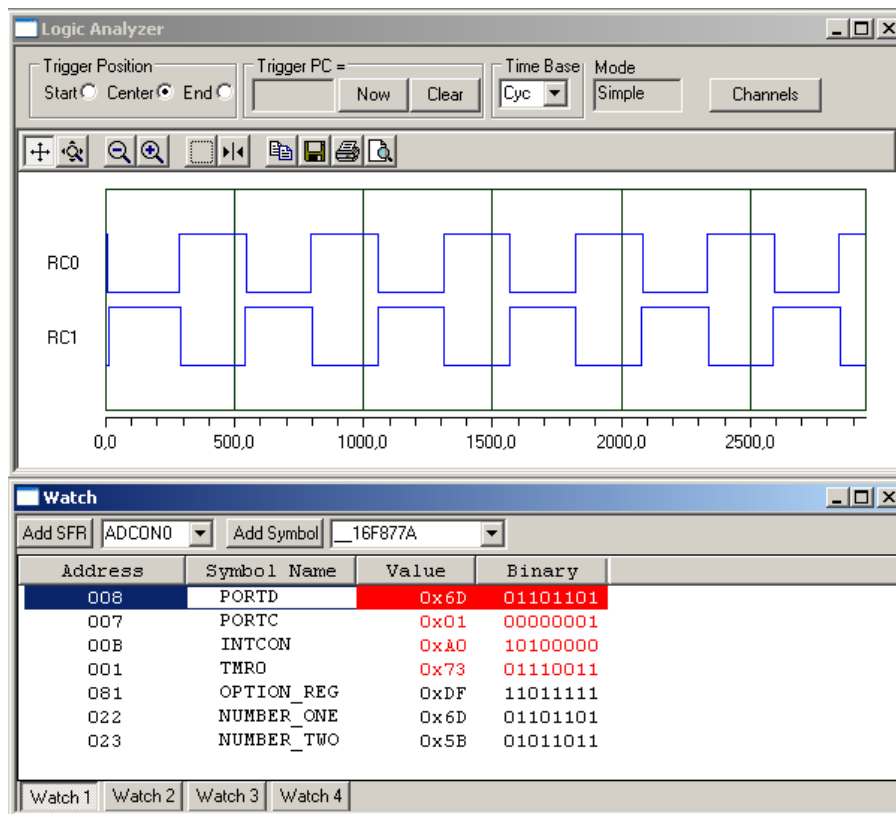


Рис. 9. Окна Watch и Logic Analyzer отображающие регистры.

После того, как будет получено несколько периодов сигнала в окне Logic Analyzer, остановите симулятор кнопкой Halt (рис. 10).

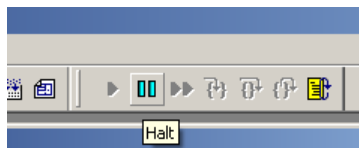


Рис. 10. Кнопка Halt для остановки симулятора.

Проанализируйте полученный результат: постарайтесь понять, какое число и когда будет подано на PORTD; какие при этом уровни будут на ножках RC0 и RC1. Предскажите результат работы программы на макете.

После симуляции соберите схему рис. 16 на лабораторном макете как показано на рис. 17. После этого в среде разработки выберите имеющийся у вас отладчик, например PICkit 2 (рис. 11).

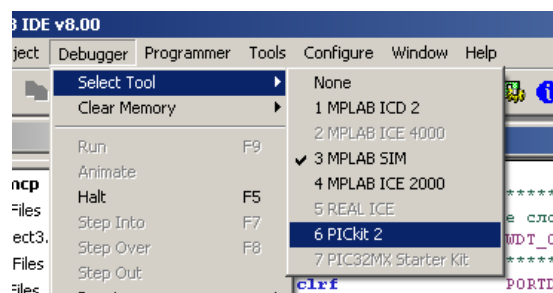


Рис. 11. Выбор отладчика в среде разработки.



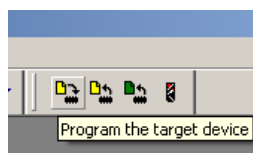


Рис. 12. Кнопка для программирования лабораторного макета.

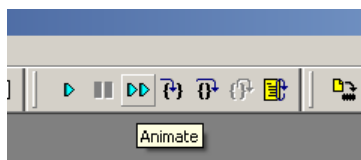


Рис. 13. Кнопка для запуска программы в режиме Animate.

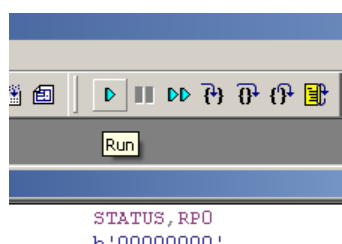


Рис. 14. Кнопка для запуска программы в режиме Run.

Запрограммируйте лабораторный макет (рис. 12). Запустите программу в режиме Animate (рис. 13). Затем запустите программу в режиме RUN (рис. 14) на макете и предъявите 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. 15. Принципиальная электрическая схема для выполнения лабораторной работы изображена на рис. 16. На рис. 17 показаны соединения на плате лабораторного макета.

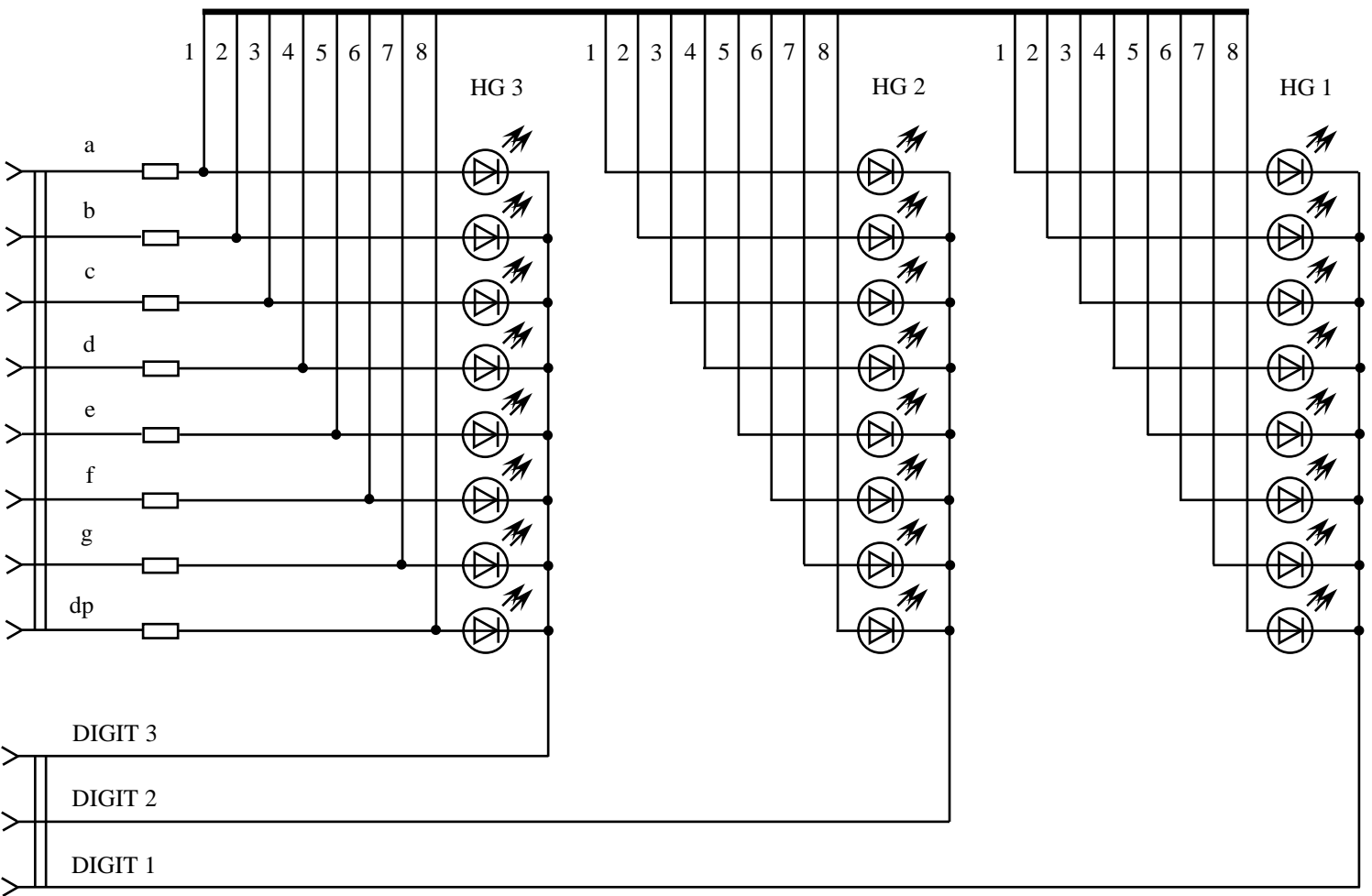


Рис. 15. Принципиальная электрическая схема семисегментных индикаторов.

DD1 PIC16F877A

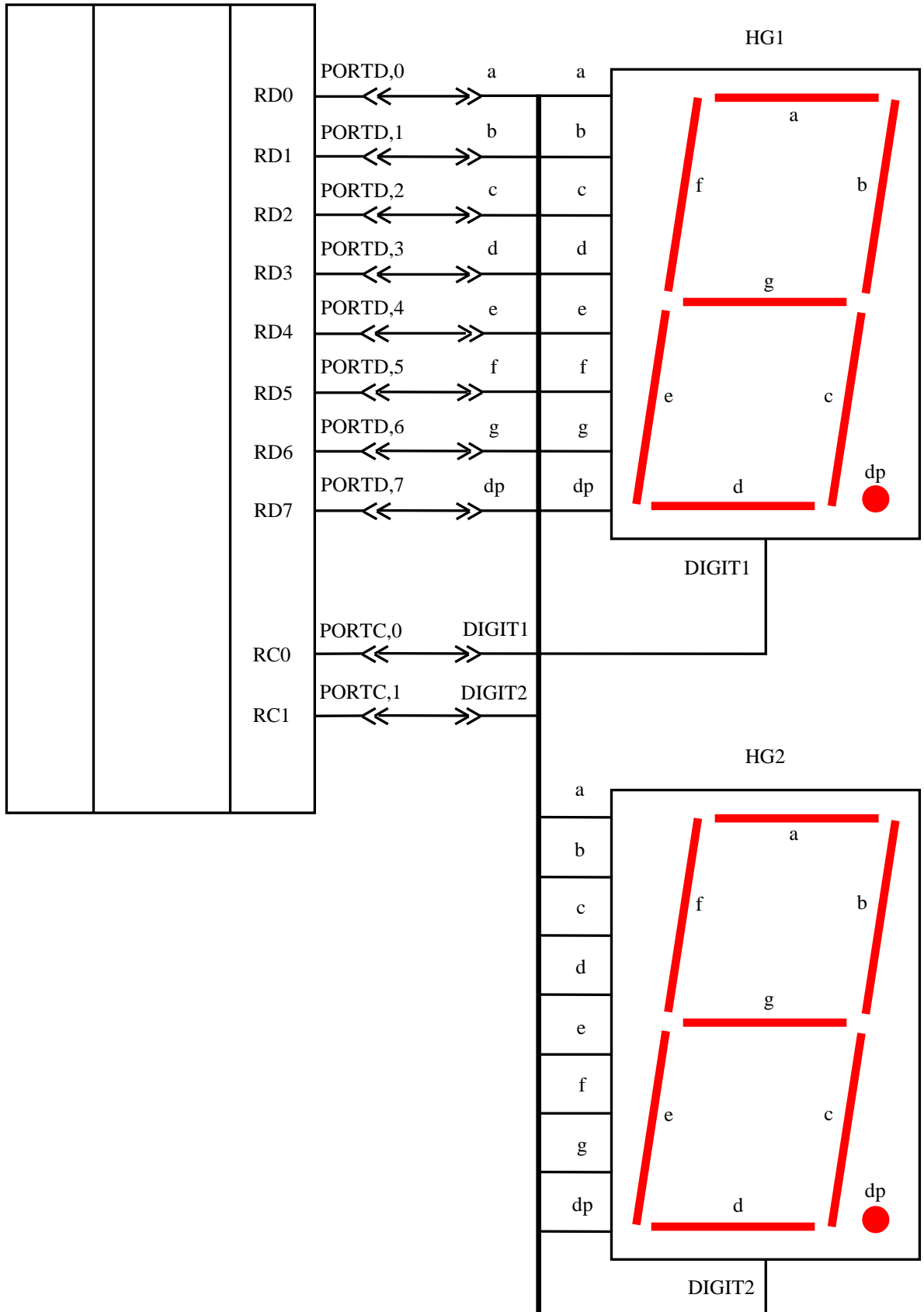


Рис. 16. Принципиальная электрическая схема для выполнения лабораторной работы.

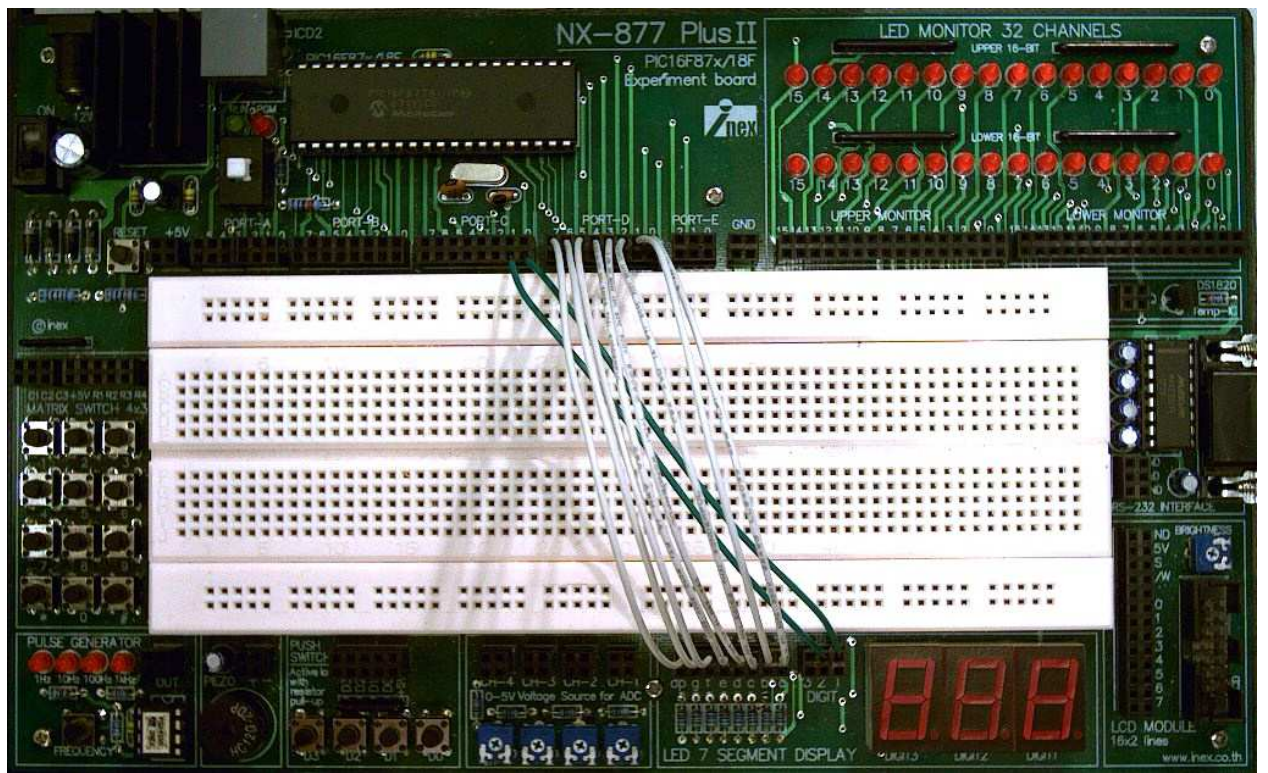


Рис. 17. Схема, собранная на лабораторном макете.

## Программное обеспечение

Алгоритм программы изображён на рис. 18а и 18б.

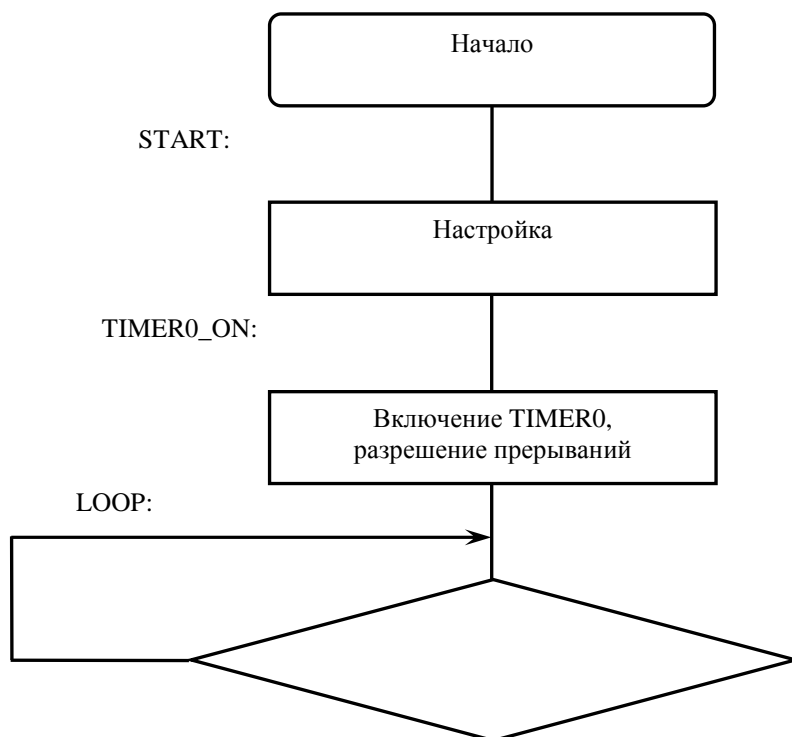


Рис. 18 а) Алгоритм основной программы.

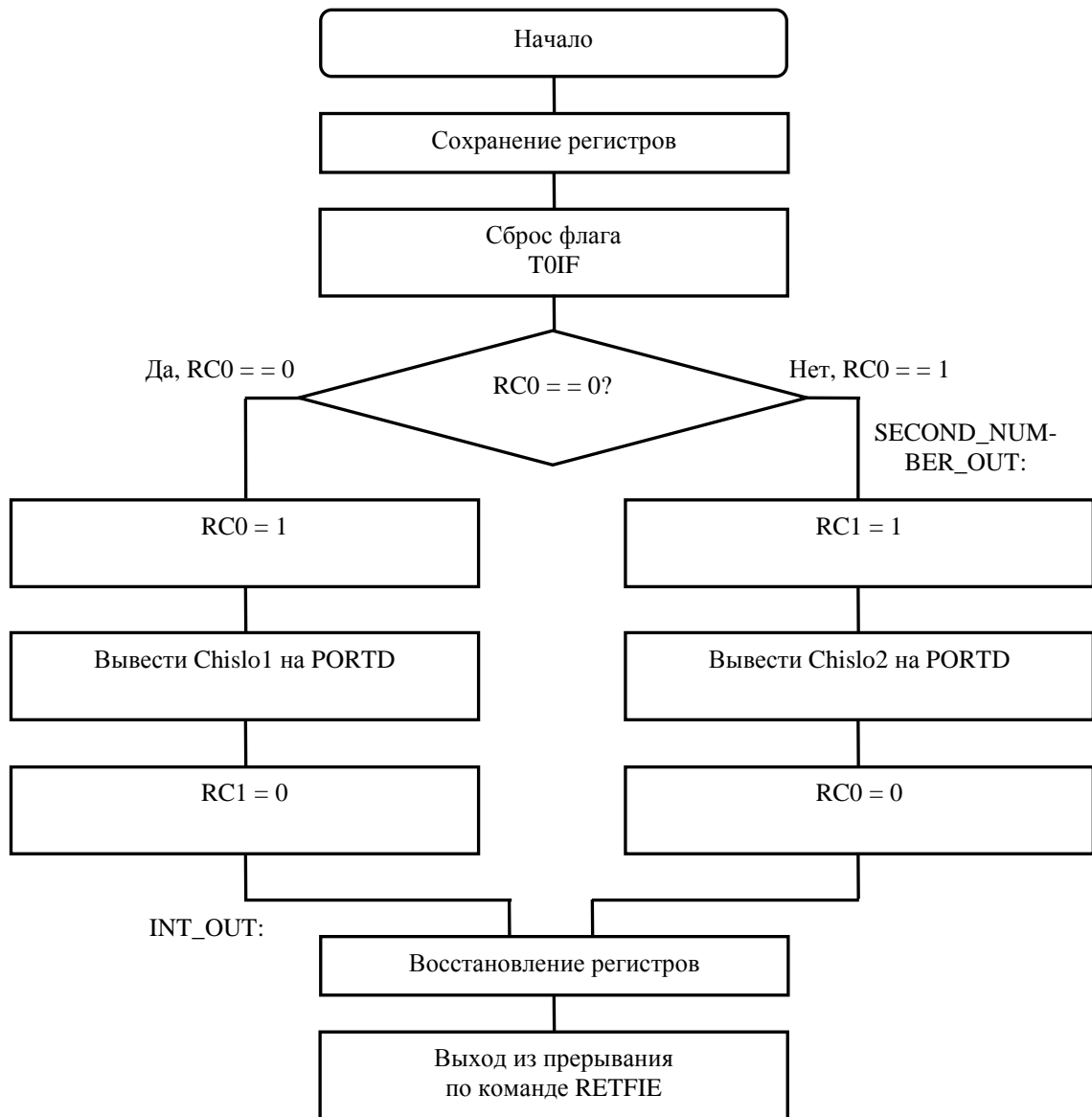


Рис. 18 б) Алгоритм обработчика прерывания.

Текст файла Project8.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
org         00h
nop
goto       START
org         0x04
movwf     W_TEMP
swapf    STATUS,W
clrf     STATUS
movwf    STATUS_TEMP
btfss   INTCON, T0IF
goto    INT_OUT
bcf     INTCON, T0IF
btfsc   PORTC,0
goto    SECOND_NUMBER_OUT
bsf     PORTC,0
movf    NUMBER_ONE,W
movwf   PORTD
bcf     PORTC,1
goto    INT_OUT
SECOND_NUMBER_OUT:
bsf     PORTC,1
movf    NUMBER_TWO,W
movwf   PORTD
bcf     PORTC,0
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
movwf   TRISC
bcf     STATUS,RP0
bsf     PORTC,1
movlw   b'01101101'
movwf   NUMBER_ONE
movlw   b'01011011'
movwf   NUMBER_TWO
TIMER0_ON:
clrf    TMR0;
```

```
        bsf      STATUS, RP0
        bcf      OPTION_REG, T0CS
        bcf      STATUS, RP0
        bsf      INTCON, T0IE
        bsf      INTCON, GIE

loop:
        goto    loop
        end
```

### Индивидуальные задания

Выведите три разных символа или цифры на семисегментные индикаторы в режиме динамической индикации.

Напишите программу для подсчёта числа нажатий кнопки в диапазоне от 0 до 24 нажатий.

### Контрольные вопросы

1. Что такое периферийные модули? Назовите несколько периферийных модулей.
2. Что такое таймеры?
3. Где можно прочитать о включении и выборе режима работы нулевого таймера?
4. Как пользоваться нулевым таймером?
5. Что такое динамическая индикация?
6. Прокомментируйте заполненную таблицу включения и настройки нулевого таймера.



Оглавление:

ЛАБОРАТОРНАЯ РАБОТА №6 «ПЕРИФЕРИЙНЫЙ МОДУЛЬ НУЛЕВОЙ ТАЙМЕР».	3
Цель работы .....	3
Теоретические основы .....	3
Задание.....	3
Порядок выполнения.....	4
Аппаратное обеспечение .....	9
Программное обеспечение.....	13
Индивидуальные задания .....	16
Контрольные вопросы.....	16
Список рисунков:	
Рис. 1. Запуск логического анализатора.....	5
Рис. 2. Вид окна логического анализатора.....	5
Рис. 3. Выбор каналов. ....	6
Рис. 4. Вид окна Configure Channels после выбора RC0 и RC1. ....	6
Рис. 5. Вид окна Logic Analyzer после настройки RC0 и RC1. ....	6
Рис. 6. Выбор времени одного шага в Animate режиме.....	7
Рис. 7. Кнопка Animate.....	7
Рис. 8. Вид окна Logic Analyzer в процессе симуляции. ....	7
Рис. 9. Окна Watch и Logic Analyzer отображающие регистры.....	8
Рис. 10. Кнопка Halt для остановки симулятора. ....	8
Рис. 11. Выбор отладчика в среде разработки. ....	8
Рис. 12. Кнопка для программирования лабораторного макета. ....	9
Рис. 13. Кнопка для запуска программы в режиме Animate.....	9
Рис. 14. Кнопка для запуска программы в режиме Run.....	9
Рис. 15. Принципиальная электрическая схема семисегментных индикаторов. ....	10
Рис. 16. Принципиальная электрическая схема для выполнения лабораторной работы. ....	11
Рис. 17. Схема, собранная на лабораторном макете. ....	12
Рис. 18 а) Алгоритм основной программы.....	13
Рис. 18 б) Алгоритм обработчика прерывания.....	14

