



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.

«ЦИКЛЫ»

Методические указания к выполнению лабораторной работы

Санкт - Петербург
2009

Богураев М.В., Кисляков С.В. «ЦИКЛЫ». Методические указания к выполнению лабораторной работы №2(4). СПб: ГОУ «Лицей при СПбГУТ», 2009.

ЛАБОРАТОРНАЯ РАБОТА №2 «ЦИКЛЫ»

Цель работы

Научиться составлять циклические программы на ассемблере. Понять зависимость времени выполнения команд от частоты тактового генератора. Освоить моделирование в MPLAB IDE. Овладеть навыками программирования PIC микроконтроллеров.

Теоретические основы

Микроконтроллер выполняет программу, которая состоит из команд. Для выполнения каждой команды требуется время. Длительность выполнения команды задаётся частотой тактового генератора. Четыре периода тактового генератора составляют один машинный цикл. Большинство команд микроконтроллер PIC16F877A выполняет за один машинный цикл. За два машинных цикла выполняются команды безусловных переходов и команды условных переходов, если выполняется условие перехода.

Зависимость времени выполнения одного машинного цикла от частоты кварцевого резонатора описывается следующей формулой:

$$T_{CY} = 4 \cdot \frac{1}{F},$$

где T_{CY} время (период) выполнения одного машинного цикла в секундах, F частота кварцевого резонатора в герцах. Для оценки времени выполнения программы нужно знать количество команд и длительность выполнения каждой команды (учитывая, что команды условных переходов выполняются за два машинных цикла, если выполнено условие перехода). Умножив время выполнения команд на количество команд можно рассчитать время выполнения программы.

Для организации циклов используют команды условных и безусловных переходов. Команда безусловного перехода `goto` будет отправлять микроконтроллер к началу цикла, а по команде условного перехода `decfsz` из ячейки-счётчика будет вычитаться единица. Как только содержимое ячейки счётчика сравняется с нулём, команда безусловного перехода будет пропущена и цикл закончится.

Задание

Заставьте светодиод моргать в режиме `Animate`. Создайте проект, откомпилируйте программу `Project4`. Запустите программу в симуляторе MPLAB IDE. Откройте окно симулятора `Watch`. Поставьте точки останова и промоделируйте время выполнения программы при помощи окна `Stopwatch`. Рассчитайте и реализуйте задержку на заданное время.

Подключите светодиод к порту микроконтроллера. Запрограммируйте микроконтроллер и продемонстрируйте работу программы на лабораторном макете в режиме `Animate`.

Порядок выполнения

На диске C:\ в папке Projects создайте папку Project4. В эту папку скопируйте файл Project4 и создайте проект с названием Project4. Откомпилируйте программу. Откройте окно Watch и настройте его для отображения регистров TRISD, PORTD, PCL, рабочего регистра W (WREG) и регистра COUNTER (рис. 1). Выберите симулятор MPLAB SIM (рис. 2). Выберите окно настройки симулятора (рис. 3). В появившемся окне выберите вкладку Animation/Realtime Updates (рис. 4) и настройте симулятор. Для этого в графе Animate step time установите значение 1000 мс, поставьте галочку напротив Enable Realtime watch updates и введите цифру 1 в графе x100 msecs. Запустите программу в режиме Animate (рис. 5). Обратите внимание на изменение содержимого регистров COUNTER и PORTD.

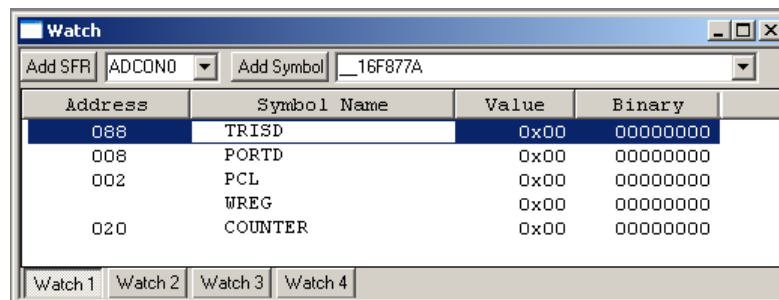


Рис. 1. Вид настроенного окна Watch.

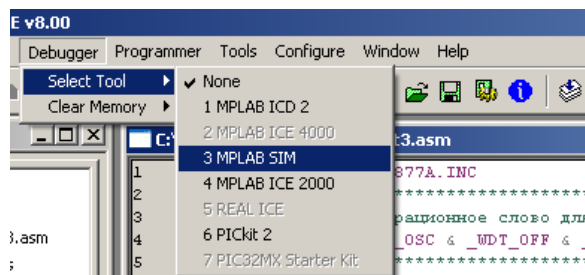


Рис. 2. Выбор симулятора MPLAB SIM.

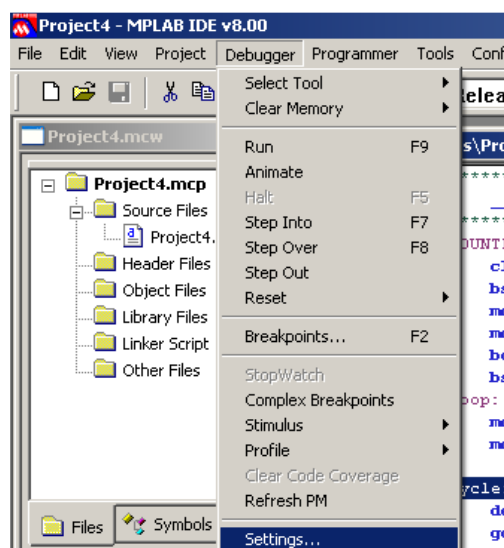


Рис. 3. Выбор окна настройки симулятора.

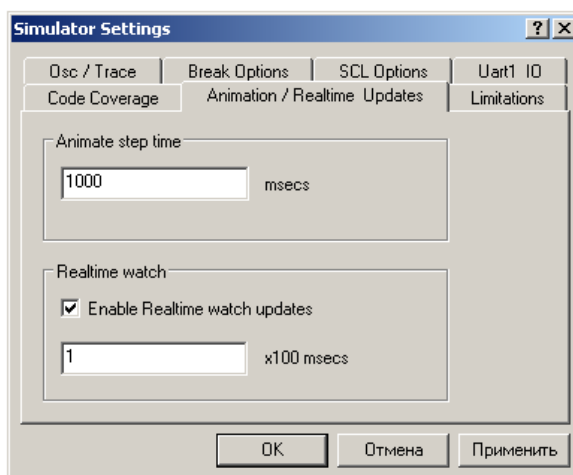


Рис. 4. Настройка симулятора.

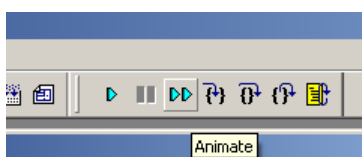


Рис. 5. Кнопка Animate.

Остановите симулятор кнопкой Halt (рис. 6). Перегрузите симулятор кнопкой Reset (рис. 7).

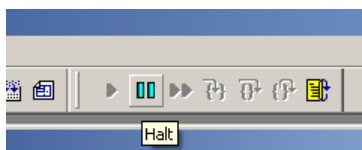


Рис. 6. Кнопка Halt для остановки симулятора.

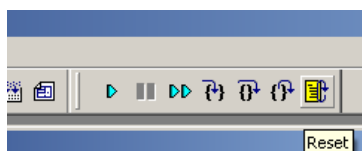


Рис. 7. Кнопка Reset для перезагрузки.

После перезагрузки поставьте точку останова напротив команды переключения светодиода. Точка останова появляется после двойного щелчка правой кнопкой мыши на строке содержащей команду (рис. 8). При необходимости убрать точку останова нужно дважды щёлкнуть правой кнопкой мыши на соответствующей строке.

```

2 ;*****
3 ;__CONFIG _HS_OSC & _WDT_OFF & _PWRTE_OFF & _CP_OFF & _BODEN_OFF & _LVP_OFF & _CPD_OFF
4 ;*****
5 COUNTER      equ 0x20      ; Указываем расположение регистра COUNTER
6 clrf        PORTD
7 bsf         STATUS,RPO    ; Переходим в 1-й банк памяти
8 movlw      b'11111110'
9 movwf      TRISD         ; Устанавливаем TRISD,0 на выход
10 bcf        STATUS,RPO    ; Возвращаемся в 0-й банк памяти
11 bsf        PORTD,0; Включаем светодиод
12 loop:
13 movlw      0x3
14 movwf      COUNTER      ; COUNTER = 0x03
15
16 cycle:
17 decfsz    COUNTER,F
18 goto     cycle
19 movlw      0x01
20 xorwf     PORTD,F; Переключаем светодиод
21 goto     loop
22 end
23

```

Рис. 8. Вид метки точки останова на 20-й строке.

Вызовите окно Stopwatch (рис. 9). Запустите программу. Выполнение программы будет приостановлено, когда программа дойдёт до точки останова. Зелёная стрелка, указывающая на следующую команду, будет находиться на точке останова. Теперь нужно сбросить показания окна Stopwatch кнопкой Zero (рис. 10). После того, как показания сброшены, нужно вновь запустить симулятор. Симуляция программы будет вновь приостановлена, когда та дойдет до точки останова. Проанализируйте показания окна Stopwatch. Рассчитайте число, которое вы поместите в регистр COUNTER для получения задержки, время которой указывает учитель.

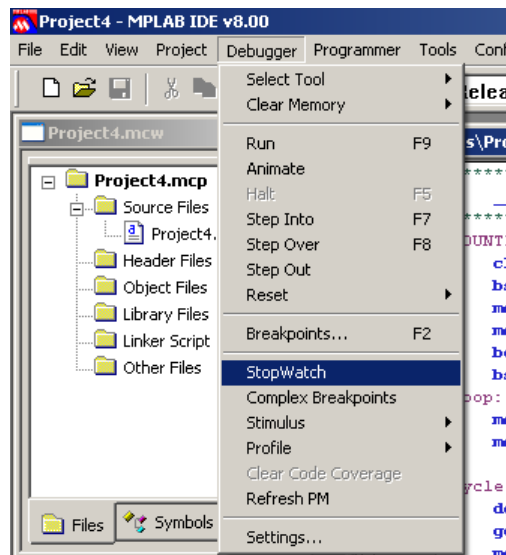


Рис. 9. Выбор окна Stopwatch.

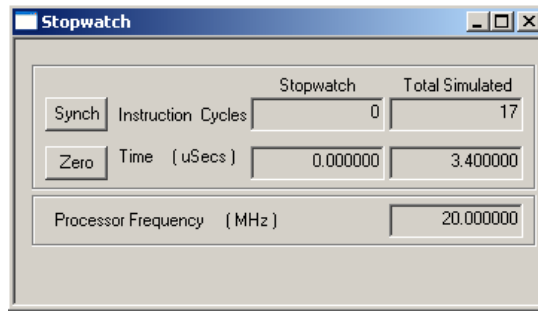


Рис. 10. Окно Stopwatch.

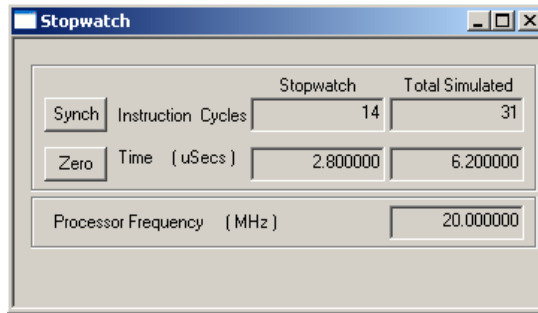


Рис. 11. Окно Stopwatch с новыми показаниями.

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. 12).

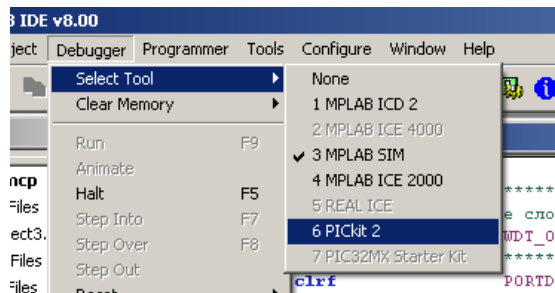


Рис. 12. Выбор отладчика в среде разработки.

Запрограммируйте лабораторный макет (рис. 14). Запустите программу в режиме Animate (рис. 15). В режиме Animate светодиод будет моргать.

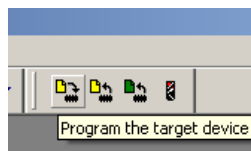


Рис. 14. Кнопка для программирования лабораторного макета.

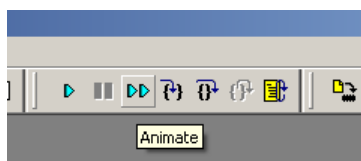


Рис. 15. Кнопка для запуска программы на макете в режиме Animate.

Аппаратное обеспечение

Аппаратное обеспечение аналогично предыдущей работе. Принципиальная электрическая схема изображена на рис. 16. Схема работает следующим образом: на микроконтроллер подаётся напряжение от блока питания. В процессе выполнения программы нулевой бит PORTD настраивается на выход. Как только в регистр PORTD,0 записывается единица, на 19 ножке микросхемы устанавливается потенциал +U. Через разъёмы и провод потенциал передаётся на балластный резистор R1, который нужен для ограничения тока через светодиод. Последовательно с резистором включен анод светодиода. Катод светодиода соединён с нулевым потенциалом. Если PORTD,0 = 1, тогда ток течёт по цепи +U, 11 ножка контроллера, внутренний коммутатор порта PORTD,0 ножка 19, разъём PORTD,0, провод, разъём LOWER MONITOR,0, балластный резистор R1, светодиод VD1, нулевой потенциал. Когда в регистр PORTD,0 записывается ноль, тогда на 19 ножке микросхемы устанавливается нулевой потенциал. Через разъёмы и провод нулевой потенциал передаётся на балластный резистор R1, и анод светодиода. Поскольку катод светодиода соединён с нулевым потенциалом и на анод подаётся нулевой потенциал, то разница потенциалов равна нулю. А так как разница потенциалов равна нулю, ток перестаёт течь и светодиод перестаёт излучать свет. В следующем цикле программы на светодиод опять подаётся единица, затем ноль и так далее.

Для расчёта времени задержки полезно знать, что на лабораторных макетах установлен кварцевый резонатор с частотой 20 МГц.

На рис. 17 показаны соединения на плате лабораторного макета.

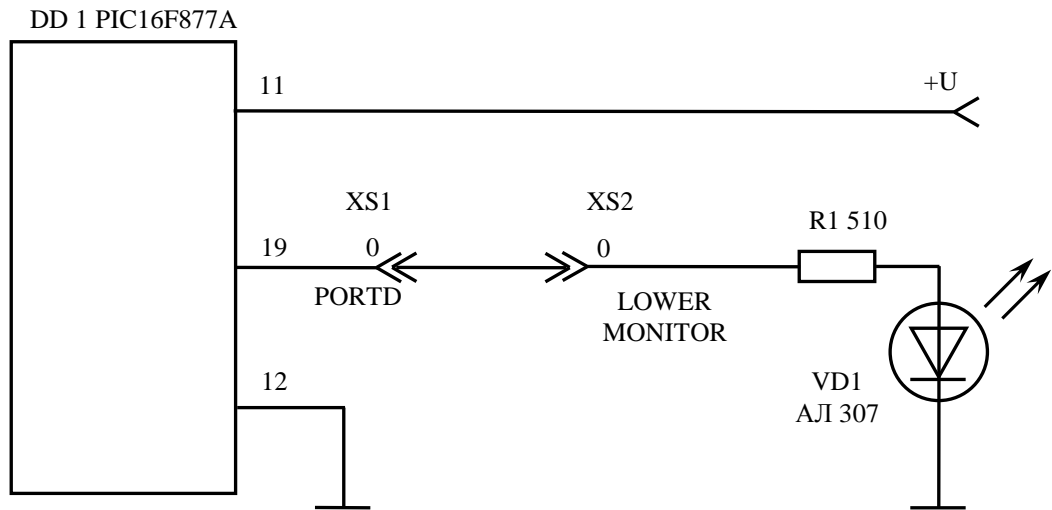


Рис. 16. Схема электрическая принципиальная.

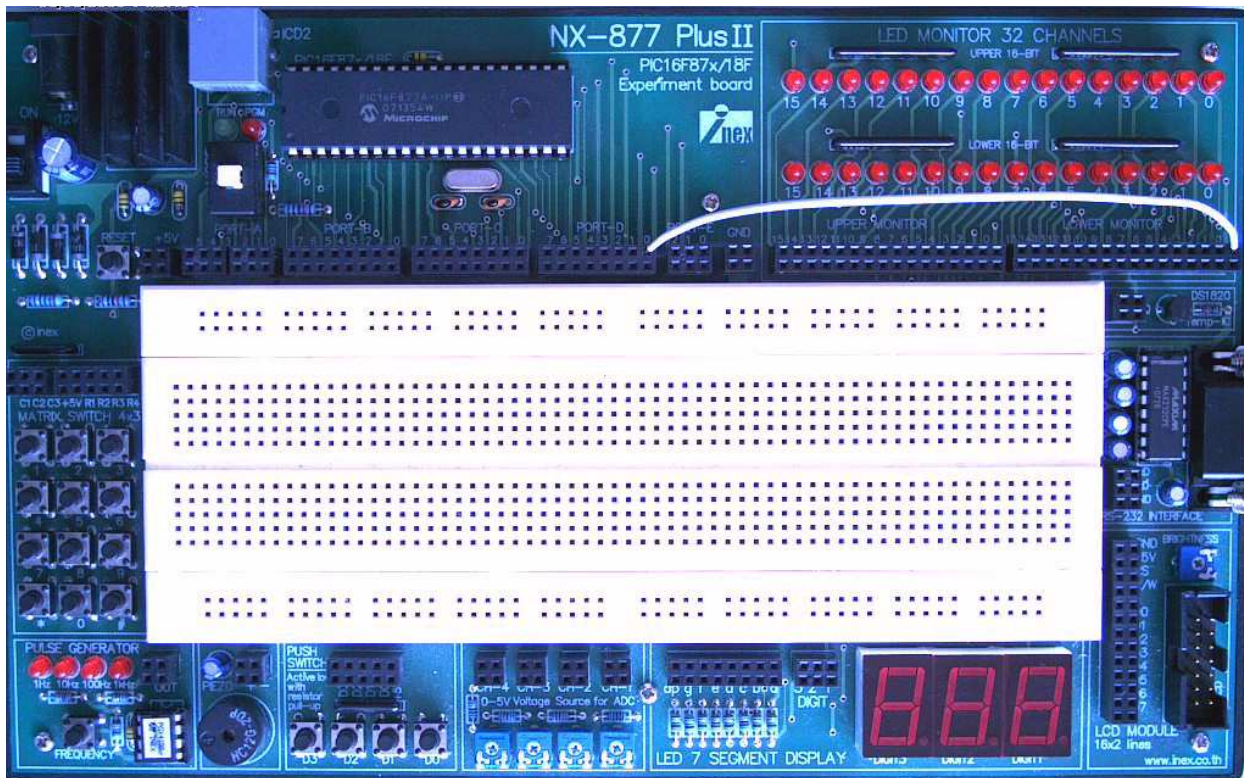


Рис. 17. Схема, собранная на лабораторном макете.

Программное обеспечение

Алгоритм программы изображён на рис. 18.

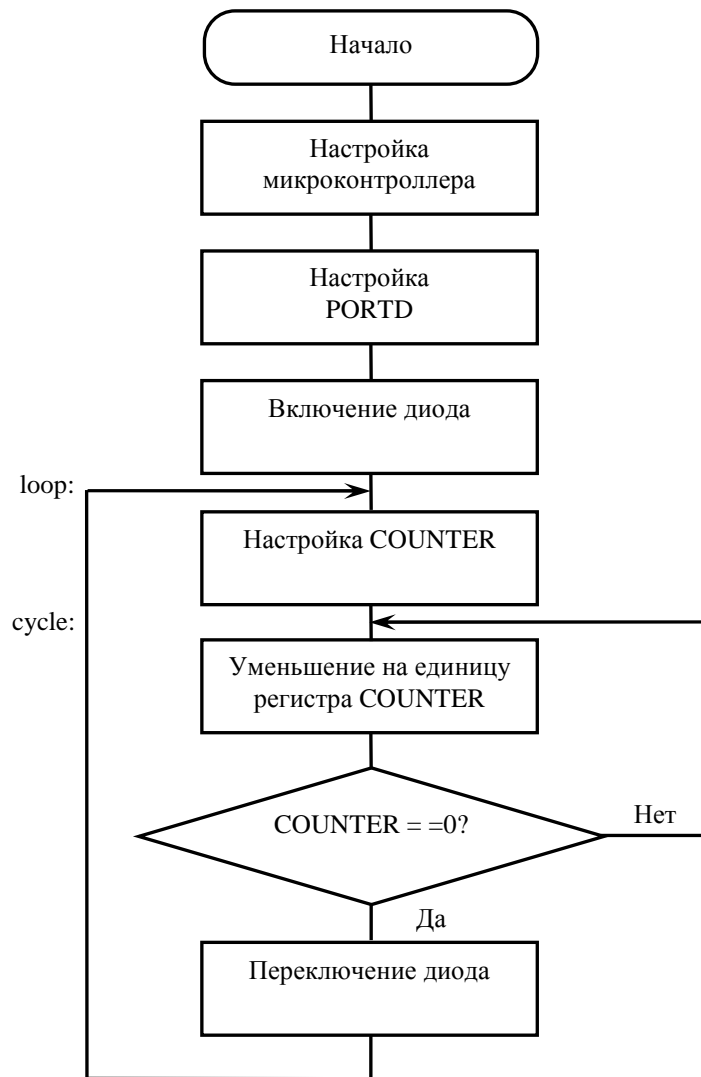


Рис. 18. Алгоритм программы Project4.

Текст файла Project4.ASM

```
*****
;
;   __CONFIG _HS_OSC & _WDT_OFF & _PWRTE_OFF & _CP_OFF &
;   _BODEN_OFF & _LVP_OFF & _CPD_OFF
;
;*****
COUNTER      equ  0x20; Указываем расположение регистра COUNTER
              clrf   PORTD
              bsf   STATUS,RP0; Переходим в 1-й банк памяти
              movlw b'11111110'
              movwf TRISD      ; Устанавливаем TRISD,0 на выход
              bcf   STATUS,RP0; Возвращаемся в 0-й банк памяти
              bsf   PORTD,0    ; Включаем светодиод

loop:
              movlw 0x3
              movwf COUNTER    ; COUNTER = 0x03

cycle:
              decfsz COUNTER,F
              goto  cycle
              movlw 0x01
              xorwf PORTD,F    ; Переключаем светодиод
              goto  loop
              end
```

Индивидуальные задания

Заставьте светодиод моргать в режиме RUN с заданным периодом.

Контрольные вопросы

1. Как зависит время выполнения машинного цикла от частоты кварцевого резонатора?
2. Как можно рассчитать время выполнения программы?
3. Какая из команд (decfsz или goto) является оператором условного перехода, а какая безусловного перехода?
4. Каков механизм работы оператора условного перехода (дать описание события, которое происходит, если условие выполнено и что происходит, если условие не выполнено)?
5. Объясните, как работают команды decfsz и incfsz?
6. Как реализовать задержку более 154 микросекунд?
7. Почему в режиме Animate светодиод моргает, а в режиме Run горит непрерывно?

Оглавление:	
ЛАБОРАТОРНАЯ РАБОТА №2 «ЦИКЛЫ»	3
Цель работы	3
Теоретические основы	3
Задание.....	3
Порядок выполнения.....	4
Аппаратное обеспечение	8
Программное обеспечение.....	10
Индивидуальные задания	11
Контрольные вопросы.....	11
Список рисунков:	
Рис. 1. Вид настроенного окна Watch.....	4
Рис. 2. Выбор симулятора MPLAB SIM.	4
Рис. 3. Выбор окна настройки симулятора.	4
Рис. 4. Настройка симулятора.	5
Рис. 5. Кнопка Animate.....	5
Рис. 6. Кнопка Halt для остановки симулятора.	5
Рис. 7. Кнопка Reset для перезагрузки.	5
Рис. 8. Вид точки останова.	6
Рис. 9. Выбор окна Stopwatch.....	6
Рис. 10. Окно Stopwatch.	7
Рис. 11. Окно Stopwatch с новыми показаниями.	7
Рис. 12. Выбор отладчика в среде разработки.	7
Рис. 14. Кнопка для программирования лабораторного макета.	7
Рис. 15. Кнопка для запуска программы на макете в режиме Animate.	7
Рис. 16. Схема электрическая принципиальная.	9
Рис. 17. Схема, собранная на лабораторном макете.	9
Рис. 18. Алгоритм программы Project4.	10

