МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский технический университет связи и информатики»

Кафедра «Многоканальные телекоммуникационные системы»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе по дисциплине
«Микропроцессорная техника в инфокоммуникациях»

Проектирование цифрового устройства на микроконтроллере. Клавиатура 4×4 и индикатор.

Выполнил:
студент группы БПАС2351
Садыкова Полина Игоревна

Проверил:
Шаврин С.С.

Москва 2026

ВВЕДЕНИЕ

В работе рассматривается устройство ввода и индикации на микроконтроллере MCS-51. Матричная клавиатура задает шестнадцатеричные параметры, которые последовательно выводятся на семисегментный светодиодный индикатор.

Цель работы:

1. Разработка и реализация на симуляторе процессоров МК51 схемы генератора в соответствии с индивидуальным заданием.

2. Разработка алгоритма работы генератора.

3. Разработка на языке ассемблера процессоров МК51 программы, реализующей работу генератора.

4. Отладка разработанной программы.

5. Оформление пояснительной записки.

1 ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Индивидуальное задание: схема № 4 (рисунок 1).

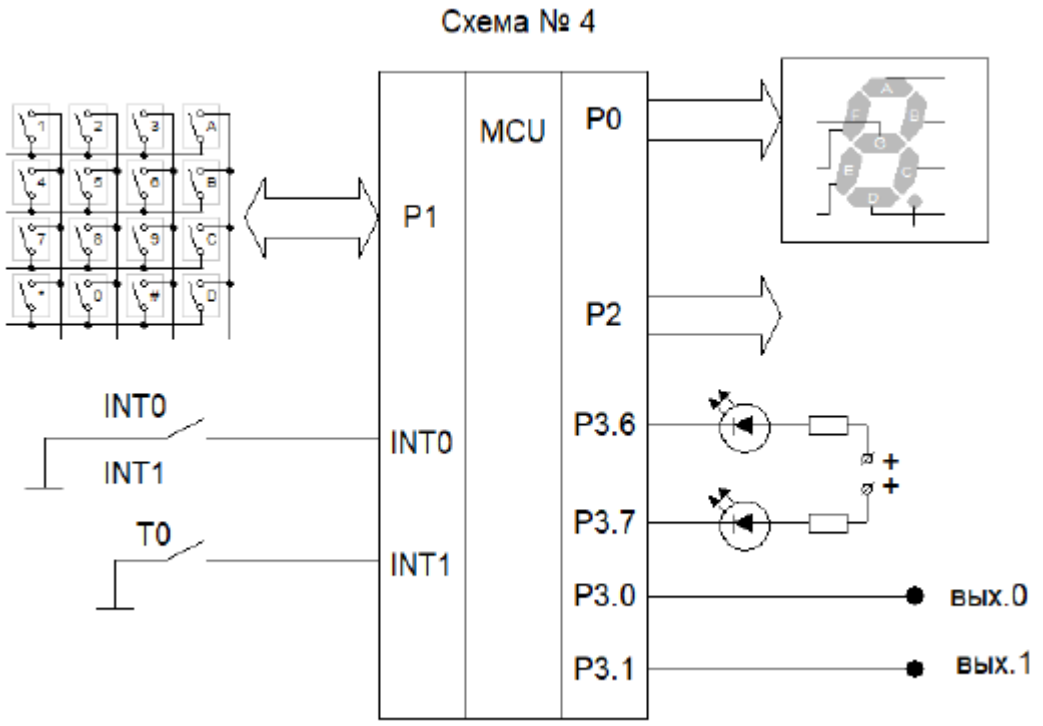


Рисунок 1 - Схема № 4 подключения внешних устройств к микроконтроллеру

Раскладка клавиатуры:

X1	–	P1.2	Y1	–	P1.7
X2	–	P1.3	Y2	–	P1.6
X3	–	P1.4	Y3	–	P1.1
X4	–	P1.5	Y4	–	P1.0

Раскладка индикатора:

A	–	P0.2	E	–	P0.7
B	–	P0.3	F	–	P0.6
C	–	P0.4	G	–	P0.1
D	–	P0.5	H	–	P0.0

Функциональность:

1. Ввод на клавиатуре значения T – длительности импульса (состояния высокого логического уровня) оно же и длительности паузы (состояния низкого логического уровня), в микросекундах, – одна или две цифры в шестнадцатеричной форме, – и его вывод на индикаторную панель. Примечание: для схемы 4 цифры выводятся на один индикатор последовательно.

2. Нажатие клавиши INT1 фиксирует это значение для дальнейшей генерации, фиксация индицируется зажиганием светодиода, подключенного к выводу P3.6.

3. Ввод на клавиатуре значения количества импульсов Nи (одна или две цифры в шестнадцатеричной форме), их вывод на индикаторную панель.

4. Нажатие клавиши INT0 фиксирует это значение и инициирует генерацию серии из Nи импульсов в порт вых.0.

5. В процессе генерации на индикаторную панель выводится текущее количество выведенных импульсов.

6. По окончании генерации светодиод, подключенный к выводу P3.6, гасится.

2 МИКРОКОНТРОЛЛЕРЫ СЕМЕЙСТВА MCS-51

2.1 Архитектура и выводы микроконтроллера

Выбран микроконтроллер AT89S52 семейства MCS-51: 8 Кбайт памяти программ, 256 байт ОЗУ, четыре 8-разрядных порта, три 16-разрядных таймера-счетчика, последовательный порт и шесть источников прерываний. Структура процессора приведена на рисунке 2.

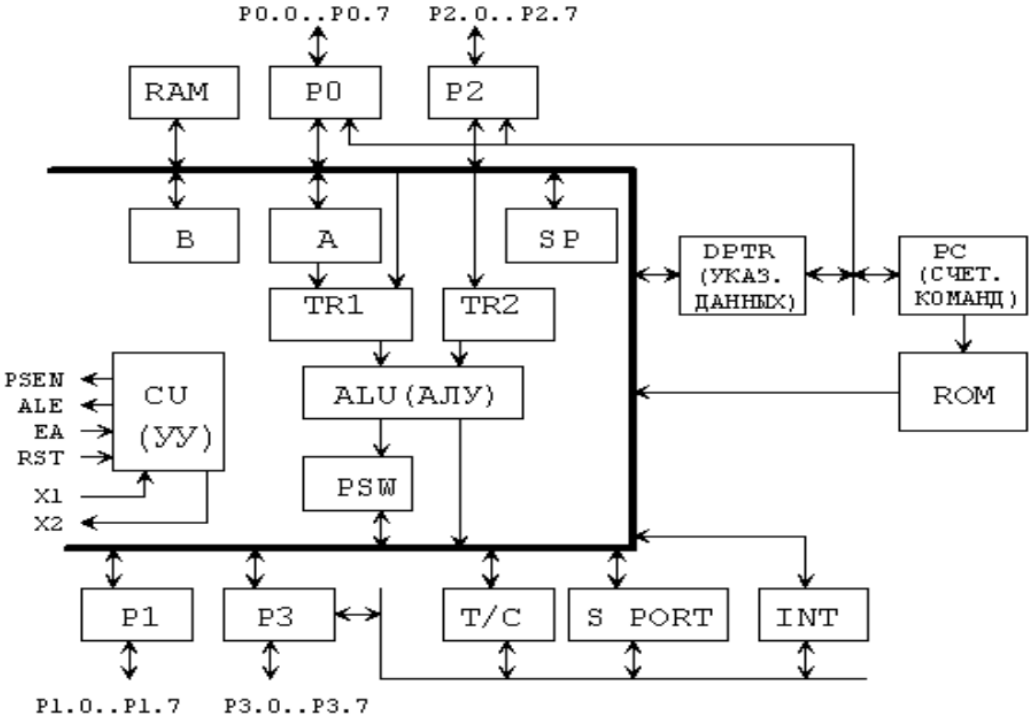


Рисунок 2 - Структура процессора MCS-51

Процессор содержит 8-разрядное АЛУ, аккумулятор A, регистры B и PSW, указатель стека SP, 16-разрядные регистры DPTR и PC. При частоте кварца 12 МГц машинный цикл длится 1 мкс и включает 12 тактов. Большинство команд занимает 1-2 машинных цикла, умножение и деление - 4.

Назначение выводов корпуса DIP-40 показано на рисунке 3. Порты P0-P3 занимают 32 вывода. Остальные используются для питания, сброса, подключения кварца и управления внешней памятью. В схеме включения на рисунке 4 установлены кварцевый резонатор, два конденсатора по 30 пФ и RC-цепь сброса. Вывод EA подключен к питанию.

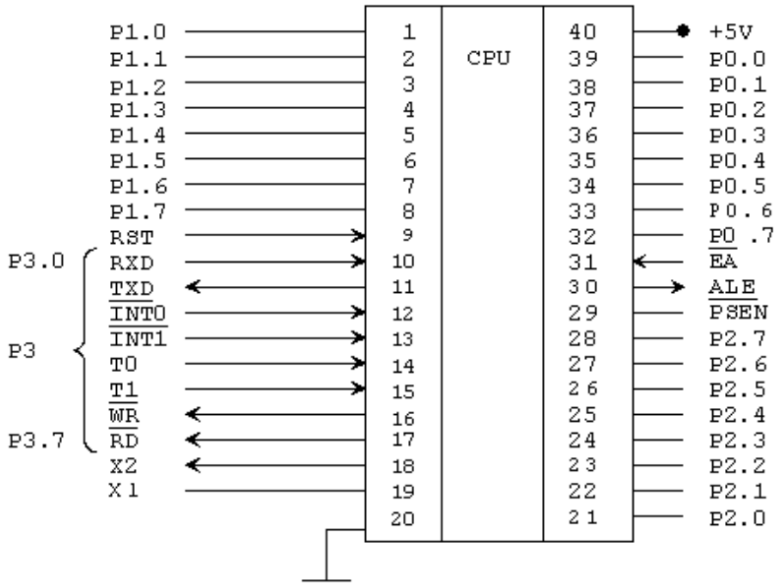


Рисунок 3 - Назначение выводов микроконтроллера

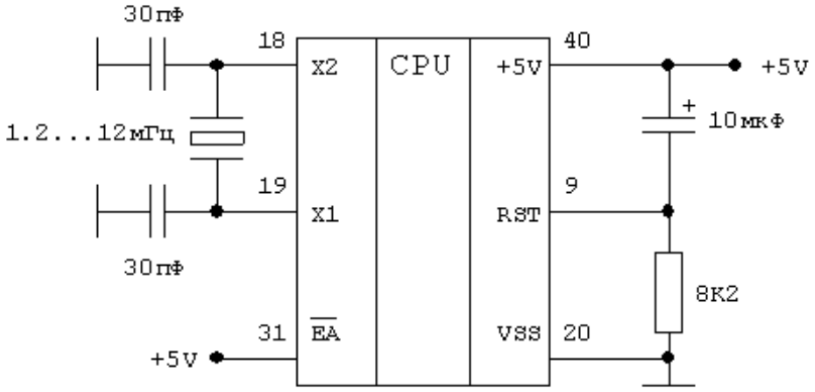


Рисунок 4 - Минимальная схема включения микроконтроллера

2.2 Организация памяти

Память программ и память данных в MCS-51 разделены (гарвардская архитектура). Из памяти программ константы читаются командой MOVC A, @A+DPTR, которая в работе используется для чтения таблиц масок и знакогенератора. Резидентное ОЗУ данных (рисунок 5) делится на области: 0-1Fh - четыре банка рабочих регистров R0-R7, 20h-2Fh - область прямо адресуемых битов (128 флагов), 30h-7Fh - ОЗУ общего назначения, область 80h-FFh содержит регистры специальных функций (порты, таймеры, регистры

прерываний) при прямой адресации и дополнительное ОЗУ при косвенной адресации.

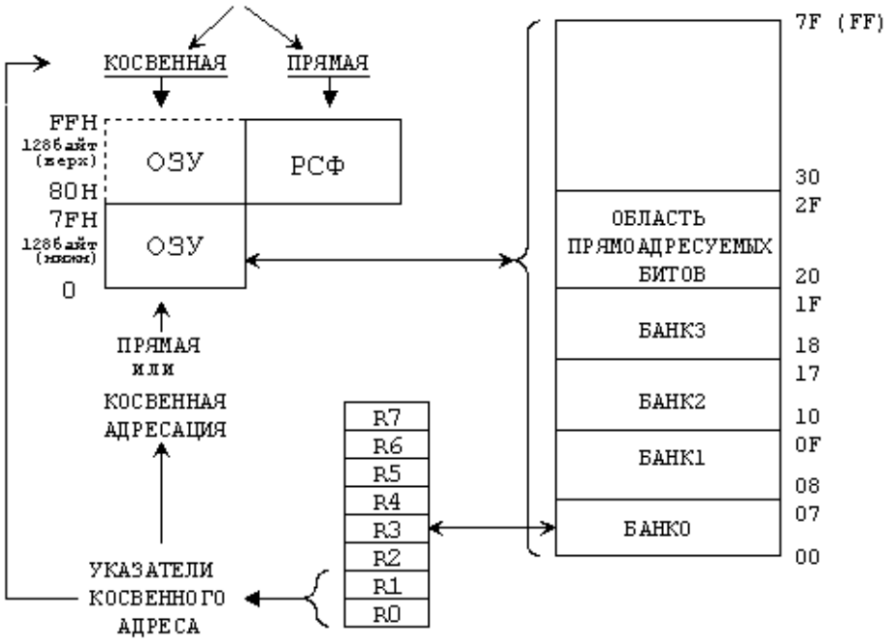


Рисунок 5 - Резидентное ОЗУ данных

В основной программе используется банк регистров 0, в обработчике таймера - банк 1, в обработчиках INT0 и INT1 - банки 2 и 3. Переключение выполняется битами RS1 и RS0 регистра PSW.

2.3 Параллельные порты ввода-вывода

Каждый вывод портов P0-P3 содержит защелку (D-триггер), выходной транзистор и входной буфер (рисунок 6). Порты P1, P2 и P3 квазидвунаправленные: в выходную схему встроен подтягивающий резистор, поэтому для использования вывода как входа в защелку записывается 1, после чего вывод может быть замкнут на общий провод внешней кнопкой. Порт P0 имеет выход с открытым стоком и при работе как порт общего назначения требует внешних подтягивающих резисторов. Выводы порта P3 имеют альтернативные функции: P3.0 - RXD, P3.1 - TXD, P3.2 - INT0, P3.3 - INT1, P3.4 - T0, P3.5 - T1, P3.6 - WR, P3.7 - RD.

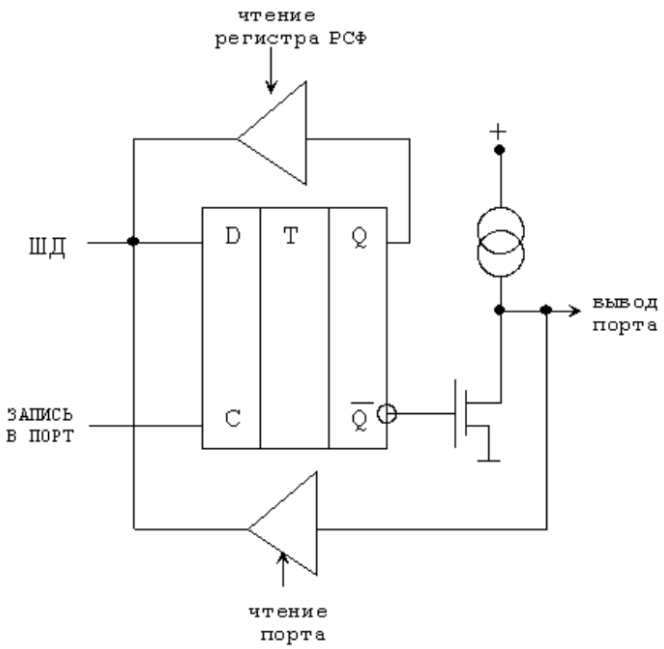


Рисунок 6 - Схема одной линии порта

Команды чтения порта (MOV A, P1, JB P3.2 и другие) читают состояние выводов, а команды типа «чтение-модификация-запись» (ANL, ORL, CPL, SETB, CLR) работают с защелкой. Нагрузочная способность вывода AT89S52 в состоянии 0 - до 10 мА, при этом суммарный ток порта P0 не должен превышать 26 мА, портов P1-P3 - 15 мА.

2.4 Таймеры-счетчики и система прерываний

Таймеры-счетчики T/C0 и T/C1 (рисунок 7) представляют собой 16-разрядные счетчики TH:TL. В режиме таймера они считают машинные циклы, в режиме счетчика - спады сигнала на входах T0, T1. Режим задается регистром TMOD: режим 0 - 13-разрядный счетчик, режим 1 - 16-разрядный, режим 2 - 8-разрядный с автоперезагрузкой, режим 3 - два 8-разрядных счетчика из T/C0. При переполнении устанавливается флаг TF0 (TF1) и формируется запрос прерывания.

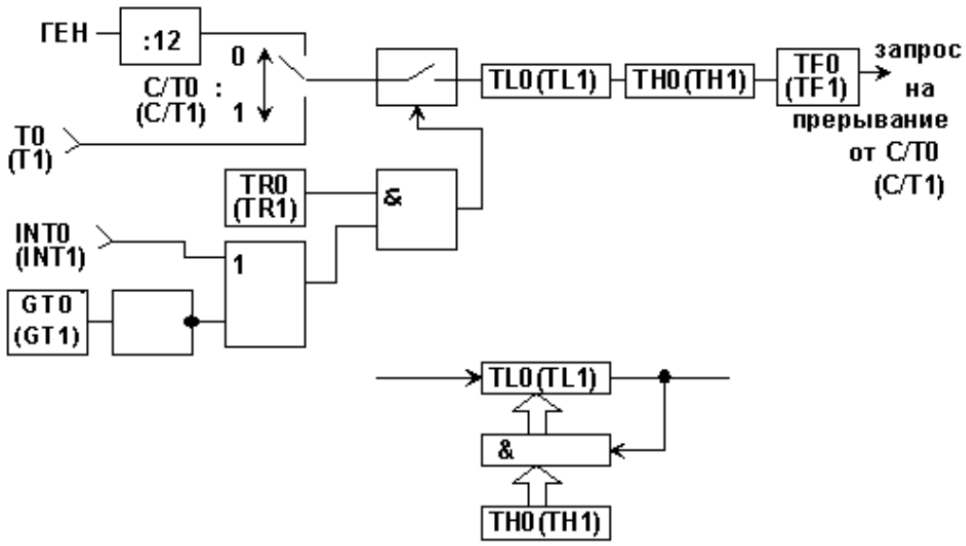


Рисунок 7 - Структура таймеров-счетчиков в режимах 0, 1 и 2

Источники прерываний, их флаги, адреса векторов и биты разрешения приведены в таблице 3. Общее разрешение прерываний задается битом EA регистра IE, приоритеты - регистром IP. Внешние прерывания INT0 и INT1 могут вызываться уровнем 0 или спадом сигнала (биты IT0, IT1 регистра TCON).

Таблица 3 - Источники прерываний MCS-51

Источник	**Флаг**	**Вектор**	**Разрешение**	**Приоритет**
INT0	IE0 = TCON.1	0003h	EX0 = IE.0	PX0 = IP.0
T/C0	TF0 = TCON.5	000Bh	ET0 = IE.1	PT0 = IP.1
INT1	IE1 = TCON.3	0013h	EX1 = IE.2	PX1 = IP.2
T/C1	TF1 = TCON.7	001Bh	ET1 = IE.3	PT1 = IP.3
Последовательный порт	RI, TI = SCON.0, 1	0023h	ES = IE.4	PS = IP.4

3 РАЗРАБОТКА СХЕМЫ УСТРОЙСТВА

Устройство выполняет следующие функции:

Согласно выданному индивидуальному заданию (Схема №4), спроектировано цифровое устройство на базе микроконтроллера КР1816ВЕ51, функционирующее в режиме генератора прямоугольных импульсов (меандра) с регулируемыми параметрами.

Модуль матричной клавиатуры 4x4: Линии строк подключены к выводам порта P1.2–P1.5 (X1–X4), линии столбцов — к P1.7, P1.6, P1.1, P1.2 (Y1–Y4). В ходе предпроектного анализа был выявлен аппаратный конфликт: пин P1.2 задействован одновременно как выход строки X1 и вход столбца Y4. В условиях симуляции это приводит к логическим гонкам. Для обеспечения стабильности системы линия столбца Y4 перенесена на свободный пин P1.0. Изолирующая маска столбцов пересчитана и составляет 0C3h (двоичное 11000011).

Модуль индикации и управления (Порты P0, P3):

• Порт P0: Подключен к семисегментному индикатору по карте: A→P0.2, B→P0.3, C→P0.4, D→P0.5, E→P0.7, F→P0.6, G→P0.1, H(DP)→P0.0. Активный уровень — логический ноль (0 / L). По заданию реализован последовательный вывод байта на один индикатор.

• Порт P3: Задействует пин P3.0 как цифровой выход генерации (вых.0), пин P3.6 как выход управления светодиодом фиксации параметров, а пины P3.2 (INT0) и P3.3 (INT1) как входы внешних команд управления.

Для подавления дребезга состояние клавиатуры проверяется с интервалом 1 мс. Оно принимается после 10 последовательных одинаковых результатов опроса. Алгоритм показан на рисунке 8.

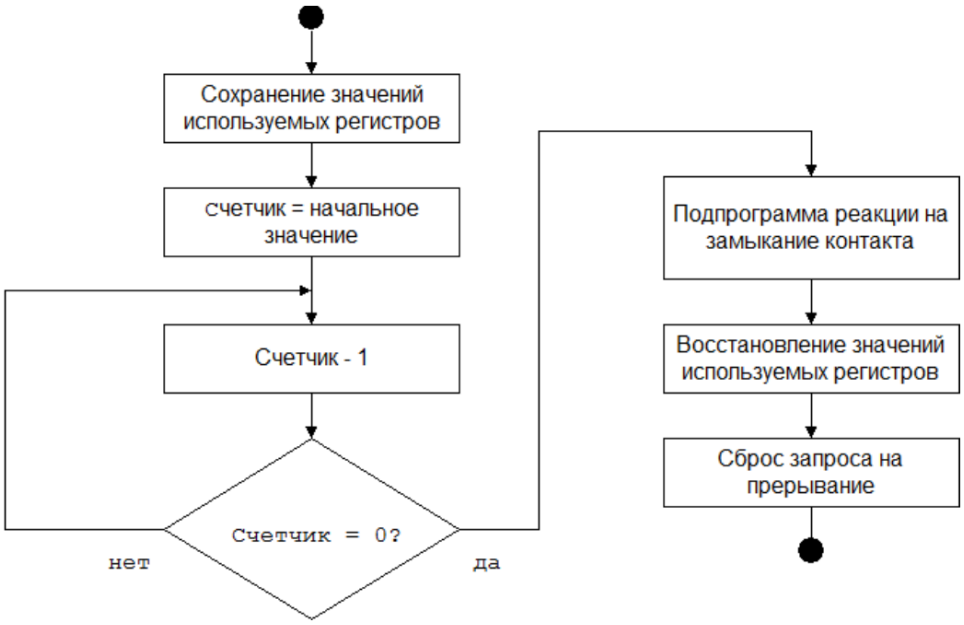


Рисунок 8 - Алгоритм подавления дребезга контакта со счетчиком

4 АЛГОРИТМ РАБОТЫ ПРОГРАММЫ

4.1 Распределение ресурсов

Распределение ресурсов микроконтроллера приведено в таблице 5.

Таблица 5 - Распределение ресурсов микроконтроллера

Ресурс	**Назначение**
30h (KEYCODE)	Ячейка ОЗУ для хранения вычисленного шестнадцатеричного кода нажатой клавиши.
31h (T_VAL)	Ячейка ОЗУ для хранения заданного пользователем значения длительности импульса \(T\).
32h (N_VAL)	Ячейка ОЗУ для хранения заданного количества импульсов серии Nn
33h (CUR_N)	Текущий счётчик фактически сгенерированных и выведенных в линию импульсов.
34h (TMP_INP)	Временный буфер для склеивания старшей и младшей тетрад при вводе двухзначного числа.
банк регистров 0	Используется для работы основного вычислительного ядра и управляющей логики (MAIN).
регистр R2	Счётчик текущей сканируемой строки матричной клавиатуры (значения 0..3).
регистр R3	Временный буфер для сохранения кода клавиши в подпрограмме ожидания WAIT_KEY.
регистр R4	Буфер для хранения вычисленного номера замкнутого столбца матрицы (значения 0..3).
регистры R5, R6, R7	Используются в качестве счётчиков-декрементов внутри функций времени DELAY_US и LONG_DELAY.
таймер T/C0	режим 1, прерывание каждую 1 мс
60h-7Fh	стек (SP = 5Fh)

4.2 Главная программа

«Графическая блок-схема главной программы отображает последовательно-иерархическую структуру управления генератором импульсов. На этапе инициализации происходит сброс регистров и гашение периферии. Ввод параметров T (длительность) и Nи (количество) реализован через вызовы подпрограммы ввода байта данных с клавиатуры. Блоки условного ветвления (ромбы проверки портов P3.3 и P3.2) играют роль аппаратных барьеров:

программа блокирует дальнейшее исполнение до тех пор, пока оператор физически не нажмет кнопку подтверждения. При фиксации времени T активируется исполнительный элемент включения светодиода готовности (P3.6 = 0). Замыкание линии стартового прерывания инициирует автоматический переход вычислительного ядра в замкнутый рабочий цикл генерации меандра GEN_LOOP с непрерывным обновлением знаков счётчика на индикаторе».

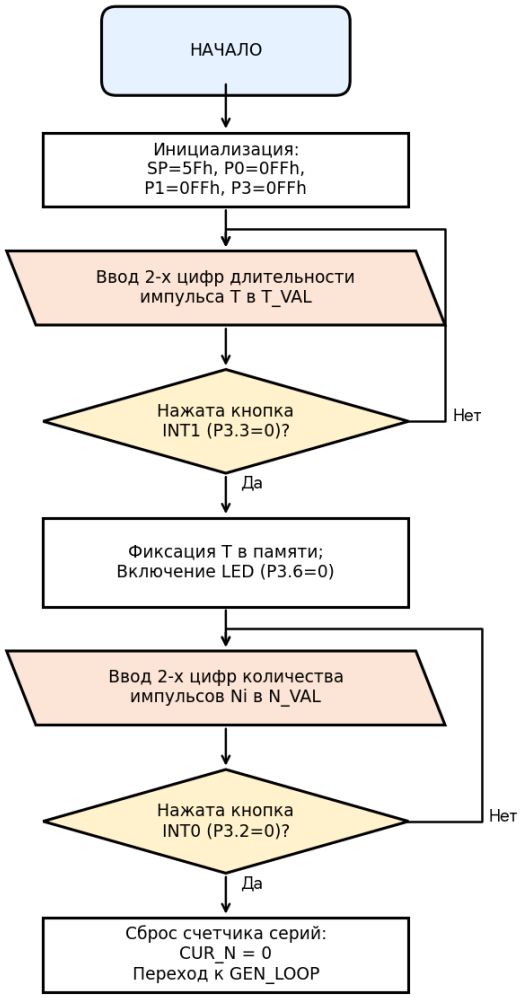


Рисунок 9 - Блок-схема главной программы

4.3 Подпрограмма

«Блок-схема подпрограммы KEYSCAN иллюстрирует алгоритм динамического поочередного опроса матричной структуры кнопок формата 4×4 методом "бегущего нуля". Внутри циклической структуры NEXTROW организовано последовательное смещение низкого логического уровня по строкам от 0 до 3

(регистр R2). Ключевой особенностью алгоритма является блок логического маскирования ANL A, #0C3h. Он изолирует биты входов столбцов (P1.7, P1.6, P1.1, P1.0), исключая влияние совмещенных линий связи. Если маска нарушена, условный блок переводит алгоритм на процедуру побитовой селекции замкнутой линии (блок FOUND) и математический расчёт итогового шестнадцатеричного индекса клавиши по формуле N = 4 x Строка + Столбец. Если за 4 итерации нажатий не обнаружено, подпрограмма возвращает в главный цикл флаг отсутствия ввода 0FFh» рисунок 10.

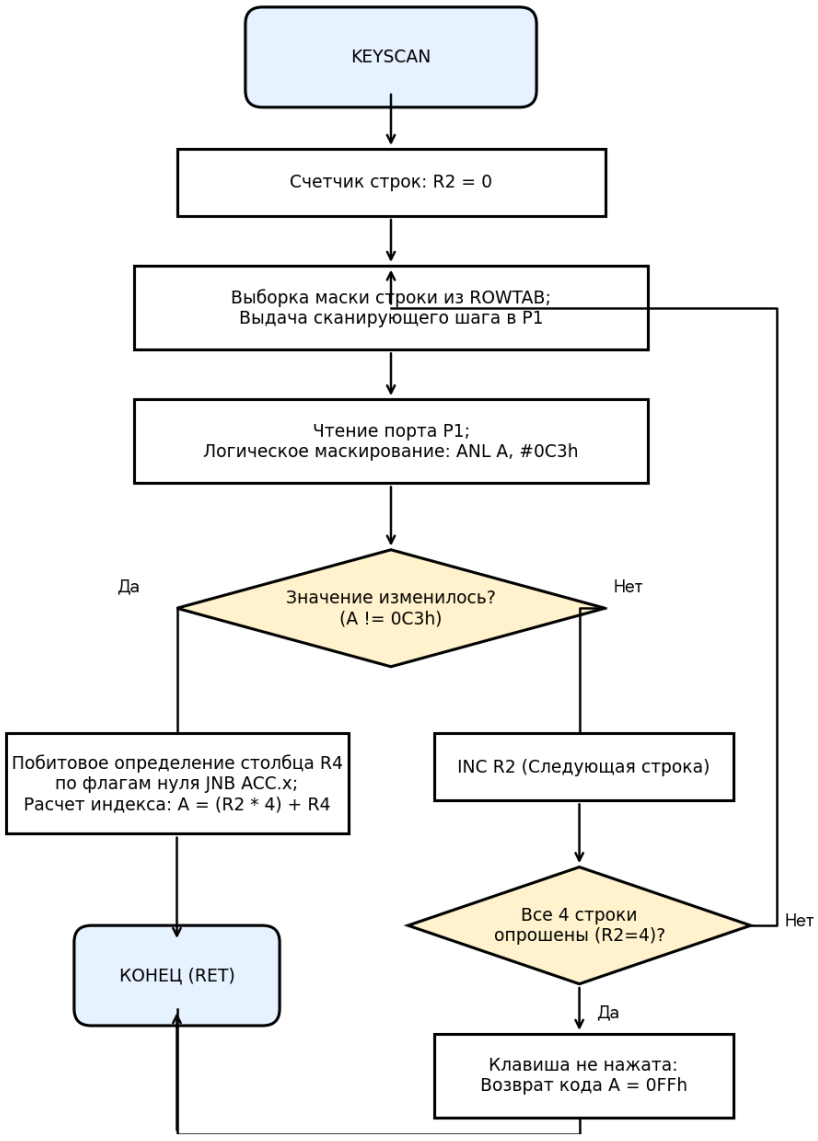


Рисунок 10 - Блок-схема подпрограммы опроса клавиатуры

5 ТЕКСТ ПРОГРАММЫ

Программа на ассемблере MCS-51 собрана в MCU 8051 IDE без ошибок и предупреждений.

Программа:

```
; --- Определение ячеек памяти ---
KEYCODE EQU 30h          ; Внутренний код нажатой клавиши
T_VAL   EQU 31h          ; Длительность импульса T (в микросекундах)
N_VAL   EQU 32h          ; Заданное количество импульсов Ni
CUR_N   EQU 33h          ; Текущий счетчик выведенных импульсов
TMP_INP EQU 34h          ; Буфер для ввода двухзначного числа

; --- Определение аппаратных пинов  ---
OUT_PIN EQU P3.0         ; Выход генерации импульсов (вых.0)
LED_PIN EQU P3.6         ; Светодиод фиксации параметров
BTN_INT0 EQU P3.2        ; Кнопка INT0 (Старт генерации)
BTN_INT1 EQU P3.3        ; Кнопка INT1 (Фиксация значения T)

      ORG 0000h
      LJMP INIT

      ORG 0030h
; --- Инициализация микроконтроллера ---
INIT:   MOV SP, #5Fh       ; Инициализация стека (область 60h-7Fh)
      MOV P0, #0FFh      ; Погасить семисегментный индикатор
      MOV P1, #0FFh      ; Порт клавиатуры в исходное состояние
      MOV P3, #0FFh      ; Погасить светодиоды, настроить входы кнопок
```

```
    CLR OUT_PIN        ; Исходное состояние выхода - 0

; ЭТАП 1: Ввод длительности импульса T

INPUT_T:
    ACALL INPUT_HEX_BYTE ; Считываем 1 или 2 цифры с клавиатуры
    MOV T_VAL, A       ; Сохраняем введенное значение длительности T

WAIT_INT1:
    JB BTN_INT1, WAIT_INT1 ; Ждем физического нажатия кнопки INT1 (P3.3)
    ACALL DEBOUNCE     ; Антидребезг для кнопки
    CLR LED_PIN        ; ЗАЖИГАЕМ светодиод P3.6 (параметры зафиксированы)

; ЭТАП 2: Ввод количества импульсов Ni

INPUT_NI:
    ACALL INPUT_HEX_BYTE ; Считываем число импульсов Ni с клавиатуры
    MOV N_VAL, A       ; Сохраняем количество импульсов

WAIT_INT0:
    JB BTN_INT0, WAIT_INT0 ; Ждем нажатия кнопки INT0 (P3.2) для старта
    ACALL DEBOUNCE     ; Антидребезг

; ЭТАП 3: Генерация серии импульсов меандра и динамический вывод

    MOV CUR_N, #0      ; Обнуляем текущий счетчик импульсов

GEN_LOOP:
    MOV A, CUR_N
    ACALL SHOW_BYTE    ; Выводим текущее число импульсов на индикатор

    ; --- Формирование высокого уровня (вых.0 = 1) ---
    SETB OUT_PIN
    MOV A, T_VAL
```

```
    ACALL DELAY_US      ; Задержка длительностью T

    ; --- Формирование низкого уровня (вых.0 = 0) ---
    CLR OUT_PIN
    MOV A, T_VAL
    ACALL DELAY_US      ; Пауза длительностью T

    INC CUR_N           ; Увеличиваем счетчик отправленных импульсов
    MOV A, CUR_N
    CJNE A, N_VAL, GEN_LOOP ; Повторяем, пока не выведем все Ni импульсов

    ; --- Окончание генерации серии ---
    MOV A, CUR_N
    ACALL SHOW_BYTE     ; Выводим финальное число импульсов
    SETB LED_PIN        ; ГАСИМ светодиод P3.6

    LJMP INIT           ; Перезапуск устройства к началу нового ввода

; ПОДПРОГРАММЫ ОБСЛУЖИВАНИЯ ПЕРИФЕРИИ

; --- Подпрограмма ввода байта (двух HEX-цифр последовательно) ---
INPUT_HEX_BYTE:
    ACALL WAIT_KEY      ; Ожидаем нажатия первой цифры
    MOV KEYCODE, A
    ACALL SHOW_DIGIT    ; Выводим первую цифру на панель
    MOV A, KEYCODE
    SWAP A              ; Сдвигаем в старшие 4 бита
    MOV TMP_INP, A      ; Сохраняем в буфер

    ACALL WAIT_KEY      ; Ожидаем нажатия второй цифры
    MOV KEYCODE, A
    ACALL SHOW_DIGIT    ; Выводим вторую цифру последовательно
    MOV A, KEYCODE
    ADD A, TMP_INP      ; Склеиваем обе цифры в один байт
```

```
    RET

; --- Ожидание отпускания и нажатия клавиши ---
WAIT_KEY:
    ACALL KEYSCAN
    CJNE A, #0FFh, WAIT_KEY ; Если кнопка уже была зажата — ждем отпускания
WK_LOOP:ACALL KEYSCAN
    CJNE A, #0FFh, WK_PRESS
    SJMP WK_LOOP
WK_PRESS:
    MOV KEYCODE, A      ; Сохраняем код нажатой клавиши
    ACALL DEBOUNCE      ; Пережидаем дребезг
    RET

; --- Подпрограмма последовательного вывода байта на один индикатор ---
SHOW_BYTE:
    MOV KEYCODE, A      ; Временно сохраняем байт в KEYCODE
    ANL A, #0F0h        ; Выделяем старшую цифру
    SWAP A
    ACALL SHOW_DIGIT    ; Выводим старшую цифру
    ACALL LONG_DELAY    ; Визуальная задержка между цифрами

    MOV A, KEYCODE
    ANL A, #0Fh         ; Выделяем младшую цифру
    ACALL SHOW_DIGIT    ; Выводим младшую цифру последовательно
    RET

; --- Вывод одной HEX-цифры (0..F) на индикатор ---
SHOW_DIGIT:
    MOV DPTR, #SEGTAB
    MOVC A, @A+DPTR
    MOV P0, A           ; Физический вывод маски палочек на индикатор P0
    RET
```

```
; --- Динамический опрос клавиатуры 4x4 (Оптимизированный под P1.0) ---
KEYSCAN:MOV R2, #0
        MOV DPTR, #ROWTAB
NEXTROW:MOV A, R2
        MOVC A, @A+DPTR
        MOV P1, A
        NOP
        MOV A, P1
        ANL A, #0C3h        ; Маска столбцов Y1..Y4 (P1.7, P1.6, P1.1, P1.0)
        CJNE A, #0C3h, FOUND
        INC R2
        CJNE R2, #4, NEXTROW
        MOV A, #0FFh
        RET
FOUND:  JNB ACC.7, C0
        JNB ACC.6, C1
        JNB ACC.1, C2
        JNB ACC.0, C3
        MOV A, #0FFh
        RET
C0:     MOV R4, #0
        SJMP CALC
C1:     MOV R4, #1
        SJMP CALC
C2:     MOV R4, #2
        SJMP CALC
C3:     MOV R4, #3
CALC:   MOV A, R2
        RL A
        RL A
        ADD A, R4
        RET

; --- Микросекундная регулируемая задержка длительности импульса T ---
```

```
DELAY_US:
      MOV R6, A          ; Загружаем значение T микросекунд
D_US_L: NOP              ; Каждая итерация при 12 МГц длится около 1-2 мкс
      DJNZ R6, D_US_L
      RET

; --- Визуальная задержка между последовательным выводом знаков ---
LONG_DELAY:
      MOV R7, #20
LD1:   MOV R6, #250
LD2:   DJNZ R6, LD2
      DJNZ R7, LD1
      RET

; --- Подпрограмма задержки антидребезга контактов ---
DEBOUNCE:
      MOV R7, #40
D1:    MOV R6, #250
D2:    DJNZ R6, D2
      DJNZ R7, D1
      RET

; --- Константы таблиц данных в памяти программ ---
ROWTAB: DB 0FBh, 0F7h, 0EFh, 0DFh
SEGTAB: DB 003h, 0E7h, 051h, 0C1h, 0A5h, 089h, 009h, 0E3h
      DB 001h, 081h, 021h, 00Dh, 01Bh, 045h, 019h, 039h

      END
```

6 ОТЛАДКА ПРОГРАММЫ В MCU 8051 IDE

В MCU 8051 IDE 1.4.9 создан проект polina2 для AT89S52 с частотой кварца 12 МГц. В проект добавлен файл pol.asm. Результат сборки показан на рисунке 12.

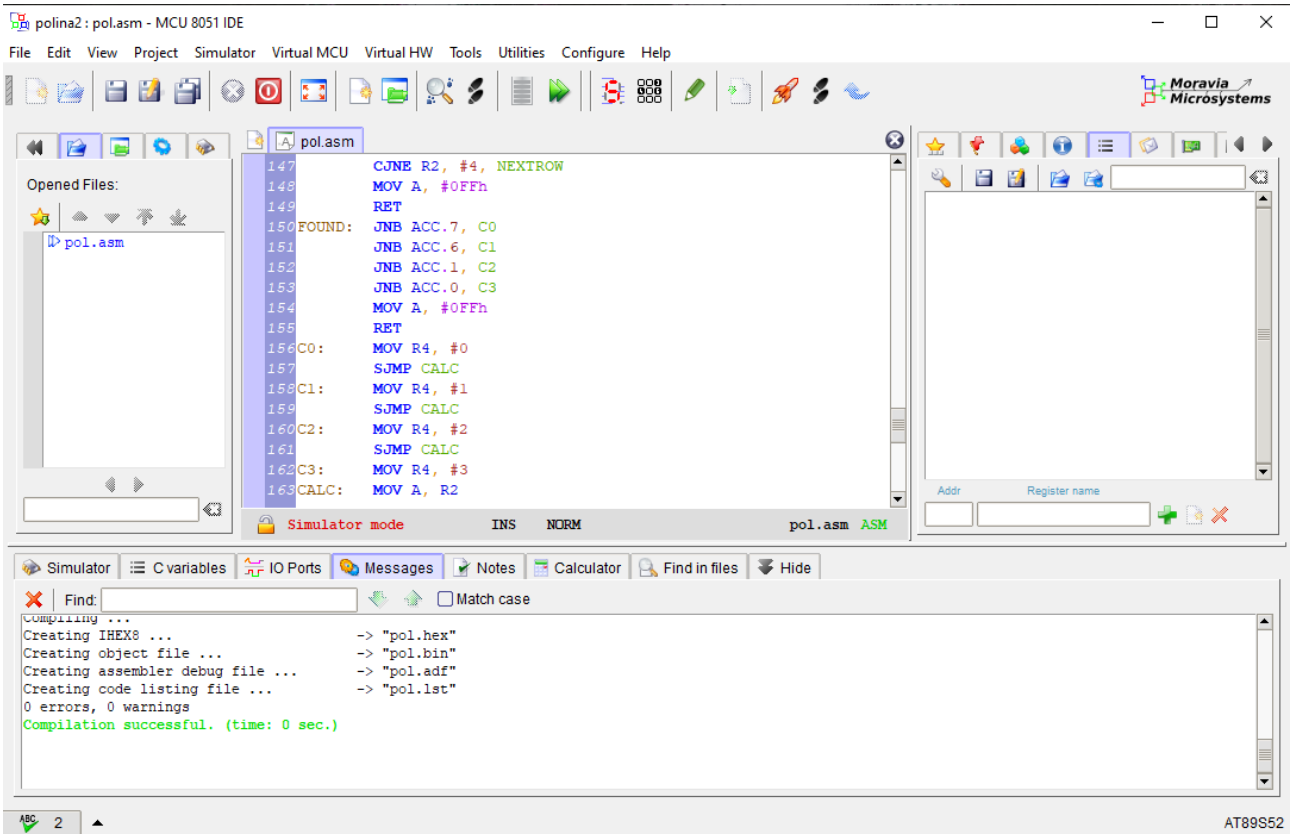


Рисунок 12 - Проект в среде MCU 8051 IDE после трансляции программы

В окне Virtual HW собрана индивидуальная схема №4: матричная клавиатура Matrix Keypad подключена к порту P1 по таблице 1 с оптимизированным переносом линии Y4 на пин P1.0. Семисегментный индикатор LED Display подключен к выводам порта P0 в соответствии с индивидуальной картой распределения (таблица 2) с назначением децимильной точки на вывод P0.0. Светодиодный индикатор зафиксирован на линии P3.6, а тактовые клавиши управления INT0 и INT1 привязаны к выводам прерываний P3.2 и P3.3 соответственно. Созданная аппаратная конфигурация виртуального стенда

сохранена в рабочем файле polina3.vhw. Проверка функционирования генератора в режиме реального времени приведена на рисунке 13.

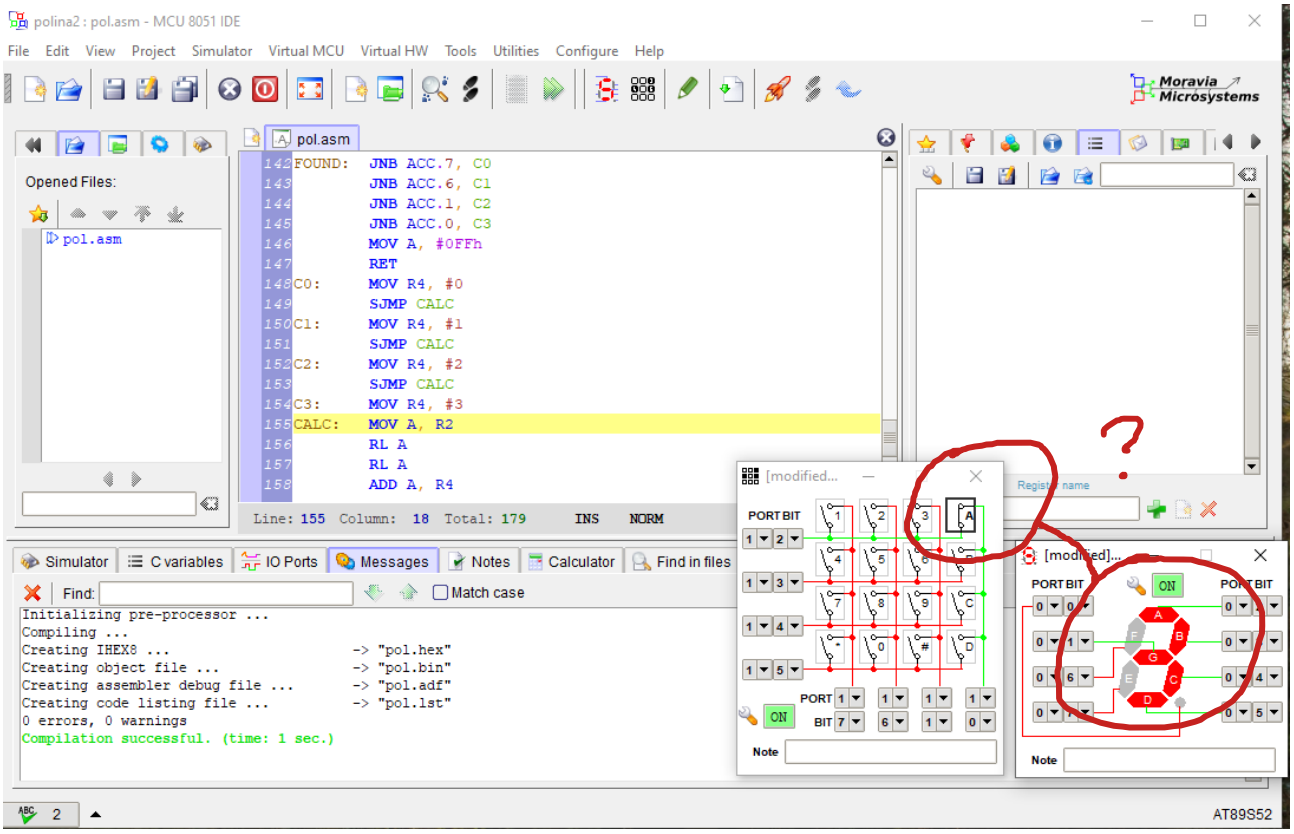
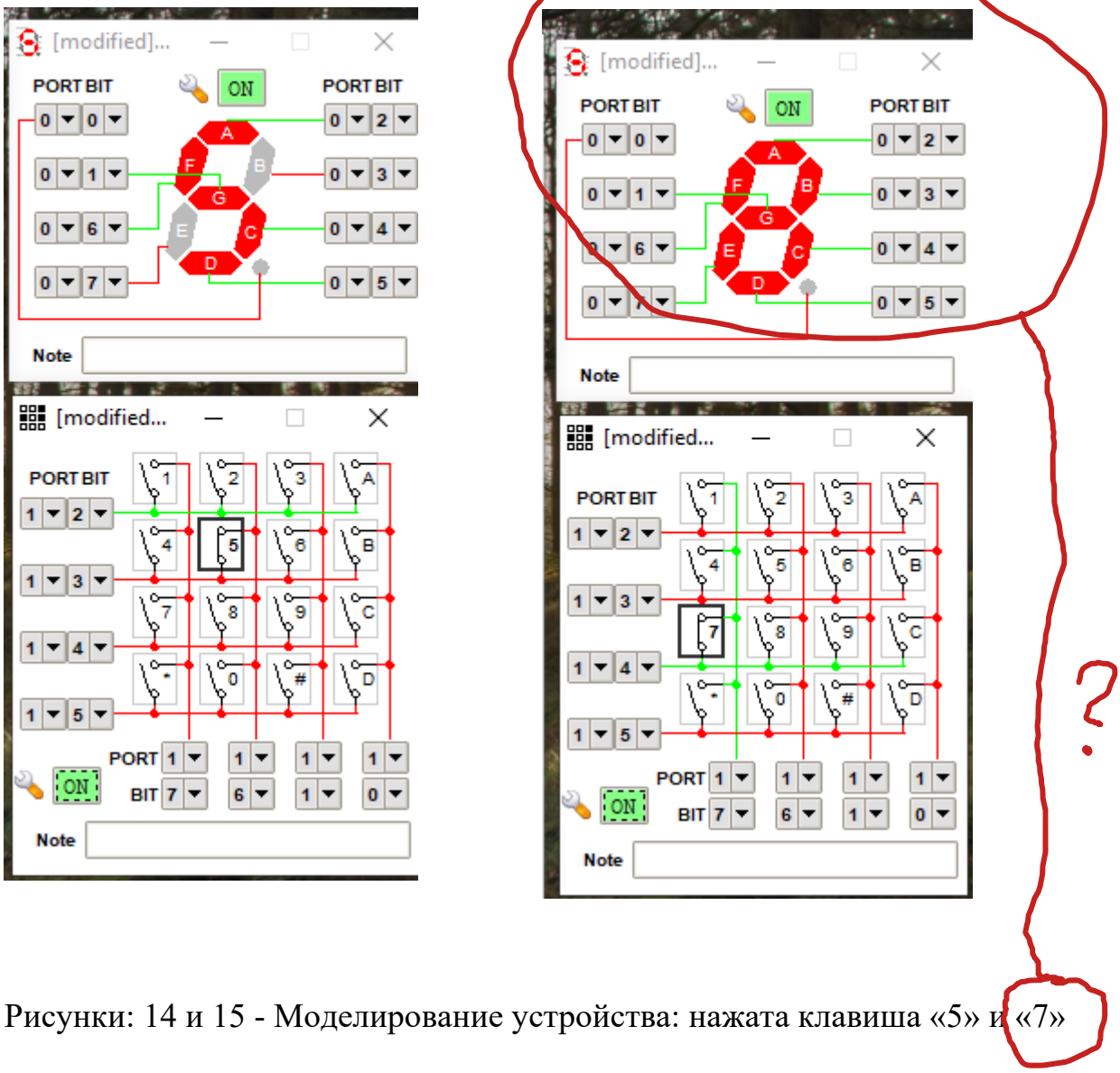


Рисунок 13 - Моделирование устройства: нажата клавиша «A»

Пример работоспособности представлены на рисунках 14 и 15:



Рисунки: 14 и 15 - Моделирование устройства: нажата клавиша «5» и «7»

ЗАКЛЮЧЕНИЕ

Разработано микропроцессорное устройство на базе микроконтроллера AT89S52 (архитектура MCS-51) с матричной клавиатурой 4×4 и одним семисегментным светодиодным индикатором. Аппаратное подключение периферии выполнено по схеме № 4 с индивидуальной раскладкой портов. В ходе проектирования устранён критический схемотехнический конфликт на совмещённой линии P1.2 путём переноса столбца Y4 на свободный пин P1.0 и расчёта изолирующей логической маски 0C3h.

Управляющая программа на языке Ассемблера реализует алгоритм работы регулируемого генератора меандрической последовательности прямоугольных импульсов. Ввод параметров длительности импульса T и количества импульсов Ni осуществляется последовательно с выводом символов на один индикатор. Программа опрашивает клавиатуру с подавлением дребезга контактов, обрабатывает внешние команды фиксации и старта на линиях INT1 и INT0, управляет контрольным светодиодом на выводе P3.6 и формирует заданную серию сигналов на цифровом выходе вых.0 (P3.0). В среде моделирования MCU 8051 IDE успешно проверены все режимы работы устройства, подтвердившие полную работоспособность разработанного программно-аппаратного комплекса.

СПИСОК ИСТОЧНИКОВ

1. Сташин, В. В. Проектирование цифровых устройств на однокристальных микроконтроллерах / В. В. Сташин, А. В. Урусов, О. Ф. Мологонцева. - Москва : Энергоатомиздат, 1990. - 224 с.

2. Соколов, Ю. П. Микроконтроллеры семейства MCS-51. Архитектура, программирование, отладка : учебное пособие / Ю. П. Соколов. - Рязань : РГРТА, 2002. - 56 с.

3. Фрунзе, А. В. Микроконтроллеры? Это же просто! Т. 1 / А. В. Фрунзе. - Москва : ИД СКИМЕН, 2002. - 336 с.

4. Фрунзе, А. В. Микроконтроллеры? Это же просто! Т. 2 / А. В. Фрунзе. - Москва : ИД СКИМЕН, 2002. - 392 с.

5. Фрунзе, А. В. Микроконтроллеры? Это же просто! Т. 3 / А. В. Фрунзе. - Москва : ИД СКИМЕН, 2003. - 224 с.

6. Шаврин, С. С. Микропроцессорная техника в инфокоммуникациях : материалы лекций / С. С. Шаврин. - Москва : МТУСИ, кафедра МТС, 2026.