Министерство образования Республики Беларусь
Учреждение Образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра электронных вычислительных средств

Лабораторная работа №3
«Работа с матричной клавиатурой и семисегментными индикаторами на базе микроконтроллера»
Вариант №5

Выполнили:
ст. гр. 850702
Турко В. Д.
Маковский Р. А.

Проверил:
Санько Н. С.

Минск 2020

1 ЦЕЛЬ РАБОТЫ

Изучить принципы работы с матричной клавиатурой и семисегментым индикатором, их взаимодействие с микроконтроллером и моделирование в среде Proteus Design Suite.

2 ЗАДАНИЕ

Используя примитивы матричной клавиатуры (KEYPAD) и какого-либо варианта семисегментного индикатора (7SEG*) и 8 светодиодных индикатора в САПР Proteus, реализовать схему, зажигающую последовательно один светодиод, начиная с указанного на клавиатуре и до 8-го, при этом гася предыдущий. После достижения 8-го индикатора, осуществить аналогичное действие, но начиная с 8-го светодиода и до 1-го. Повторять пока устройство включено. Включение/выключение осуществлять с помощью переключателя. С помощью клавиатуры указывается, с какого светодиодного индикатора осуществляется старт работы системы, а также направление хода (справа налево, либо наоборот) с помощью дополнительных нецифровых кнопок. На семисегментный индикатор необходимо выводить номер текущего светодиода.

3 ХОД РАБОТЫ

3.1 Листинг программы

```
code    equ $F000
data    equ $0000
porta   equ $1000
portb   equ $1004
portc   equ $1003
porte   equ $100A
ddrc    equ $1007 ; DDRC - регистр настройки направления разрядов порта C:
tlfg2   equ $1025
tmsk2   equ $1024

    org data
init_PC     fcb $87 ; 10000111b
col_count   fcb $01 ; 00000001b
timer  fcb 0
start_led bsz 1
start_number bsz 1
direction bsz 1
```

```
    org code
begin:
    lds #$000F
    ldaa #%00000011
    staa tmsk2  ; настройка коэффициента деления частоты таймера
    ldaa init_PC ; инициализация битов 0 - 2, 7 порта C на выход
    staa ddrc    ; бит установлен в 0 - на вход, 1 - на выход
    ldab col_count ; загрузка начального значения счетчика колонок

init:
    ldaa #%10000000
    staa start_led
    ldaa #%00000001
    staa direction
    staa start_number
scan_button:
    ldaa porta
    anda #%00000001 ; проверяем, нажата ли кнопка
    beq scan_keys ; если не нажата, то проверяем клавиатуру
    jmp check_button ; иначе запускаем цикл

scan_keys:
    cmpb #%08 ; проверка на переполнение счечика
    bne check1A ; переход, если значение B не равно $08
    ldab col_count ; реинициализация если было переполнение

check1A:
    stab portc ; загрузка единицы, если не было переполнения в счетчике

    ldaa porte ; загрузка состояния порта E в акк. A
    anda #%0001 ; нажата ли кнопка в первой строке?
    beq check1B ; переход, если не первая строка
    tba ; перемещение счечика колонок в регистр A из B
    anda #%0100 ; нажата ли кнопка в первом столбце?
    beq check2A ; переход, если нет
    ldaa #%00001000 ; сохраняем цифру "2"
    staa start_number
    ldaa #%10000000 ; сохраняем первый диод
    staa start_led
    jmp end_scan ; переход в конец кода
check2A:
    tba
    anda #%0010 ; нажата ли кнопка во втором столбце?
    beq check3A ; переход, если нет
    ldaa #%00010000 ; сохраняем цифру "2"
    staa start_number
    ldaa #%01000000 ; сохраняем второй диод
    staa start_led
    jmp end_scan ; переход в конец кода
```

```
check3A:
    tba
    ldaa #%00011000 ; сохраняем цифру "3"
    staa start_number
    ldaa #%00100000 ; сохраняем третий диод
    staa start_led
    jmp end_scan

check1B:
    ldaa porte ; загрузка состояния порта E в акк. A
    anda #%0010 ; нажата ли кнопка во второй строке?
    beq check1C ; переход, если не вторая строка
    tba
    anda #%0100 ; нажата ли кнопка в первом столбце?
    beq check2B ; переход, если нет
    ldaa #%00100000 ; сохраняем цифру "4"
    staa start_number
    ldaa #%00010000 ; сохраняем четвертый диод
    staa start_led
    jmp end_scan
check2B:
    tba
    anda #%0010 ; нажата ли кнопка во втором столбце?
    beq check3B
    ldaa #%00101000 ; сохраняем цифру "5"
    staa start_number
    ldaa #%00001000 ; сохраняем пятый диод
    staa start_led
    jmp end_scan
check3B:
    tba
    ldaa #%00110000 ; сохраняем цифру "6"
    staa start_number
    ldaa #%00000100 ; сохраняем шестой диод
    staa start_led
    jmp end_scan

check1C:
    ldaa porte ; загрузка состояния порта E в акк. A
    anda #%0100 ; нажата ли кнопка в третьей строке?
    beq check1D ; переход, если не третья строка
    tba
    anda #%0100 ; нажата ли кнопка в первом столбце?
    beq check2C ; переход, если нет
    ldaa #%00111000 ; сохраняем цифру "7"
    staa start_number
    ldaa #%00000010 ; сохраняем седьмой диод
    staa start_led
    jmp end_scan
```

```
check2C:
    tba
    anda #%0010 ; нажата ли кнопка во втором столбце?
    beq check3C ; переход, если нет
    ldaa #%01000000 ; сохраняем цифру "8"
    staa start_number
    ldaa #%00000001 ; сохраняем восьмой диод
    staa start_led
    jmp end_scan
check3C:
    jmp end_scan

check1D:
    ldaa porte ; загрузка состояния порта E в акк. A
    anda #%1000 ; нажата ли кнопка в четвертой строке?
    beq end_scan ; переход, если не четвертая строка
    tba
    anda #%0100 ; нажата ли кнопка в первом столбце?
    beq check2D ; переход, если нет
    ldaa #%10000000 ; иначе задаем направление влево
    staa direction
    jmp end_scan ; продолжаем скан кнопок
check2D:
    tba
    anda #%0010 ; нажата ли кнопка во втором столбце?
    beq check3D ; переход, если нет
    jmp end_scan
check3D:
    tba
    ldaa #%00000001 ; задаем направление вправо
    staa direction
    jmp end_scan ; продолжаем скан кнопок

end_scan:
    aslb ; сдвиг регистра B влево
    jmp scan_button ; переход в начало цикла сканирования

check_button:
    bsr show ; вывод состояния
    ldaa porta ; Загружаем значение с порта A в акк. A
    anda #%00000001 ; Проверяем, нажата ли кнопка
    bne loop ; Если нажата, то запускаем цикл
    ldaa #%00000000 ; сохраняем цифру "0"
    staa porta ; сохраняем цифру "0"
    staa start_number
    ldaa #%00000000 ; выключаем диоды
    staa portb ; выключаем диоды
    staa start_led
    ldab col_count
```

```
    jmp init

loop:
    ldaa start_led ; загружаем из памяти диод
    cmpa #%00000001 ; проверяем конец движения вправо
    beq set_left ; меняем, если упёрлись
    cmpa #%10000000 ; иначе проверяем конец движения влево
    beq set_right ; меняем, если упёрлись
loop_next:
    ldaa direction ; загружаем из памяти направление
    cmpa #%00000001 ; проверяем направление вправо
    beq shift_right ; если вправо, то шаг вправо
    bra shift_left ; иначе шаг влево

set_right:
    ldaa #%00000001 ; задаем направление вправо
    staa direction
    bra loop_next
set_left:
    ldaa #%10000000 ; задаем направление влево
    staa direction
    bra loop_next

shift_right:
    ldaa start_number
    adda #%00001000 ; инкремент цифры
    lsrb ; сдвиг вправо
    bsr save_state ; сохраняем состояние
    bra check_button ; возврат к проверке кнопки

shift_left:
    ldaa start_number
    suba #%00001000 ; декремент цифры
    lslb ; сдвиг влево
    bsr save_state ; сохраняем состояние
    bra check_button ; возврат к проверке кнопки

save_state:
    staa start_number ; сохраняем цифру
    stab start_led ; сохраняем диод
    rts

show:
    ldaa start_number ; загружаем из памяти цифру
    ldab start_led ; загружаем из памяти диод
    staa porta ; выводим цифру
    stab portb ; выводим диод
    bsr delay
    rts
```

```
delay:
    ldaa #3 ; 0,524sec*3=1.5sec
    staa timer
wait
    ldaa tlfg2
    anda #%10000000
    beq wait
    staa tlfg2
    ldaa timer
    deca
    staa timer
    bne wait
    rts

    org $FFFE
    FDB begin
```

3.2 Результат работы программы

На скриншоте из САПР Proteus (рис. 3.1) представлен результат выполнения программы.

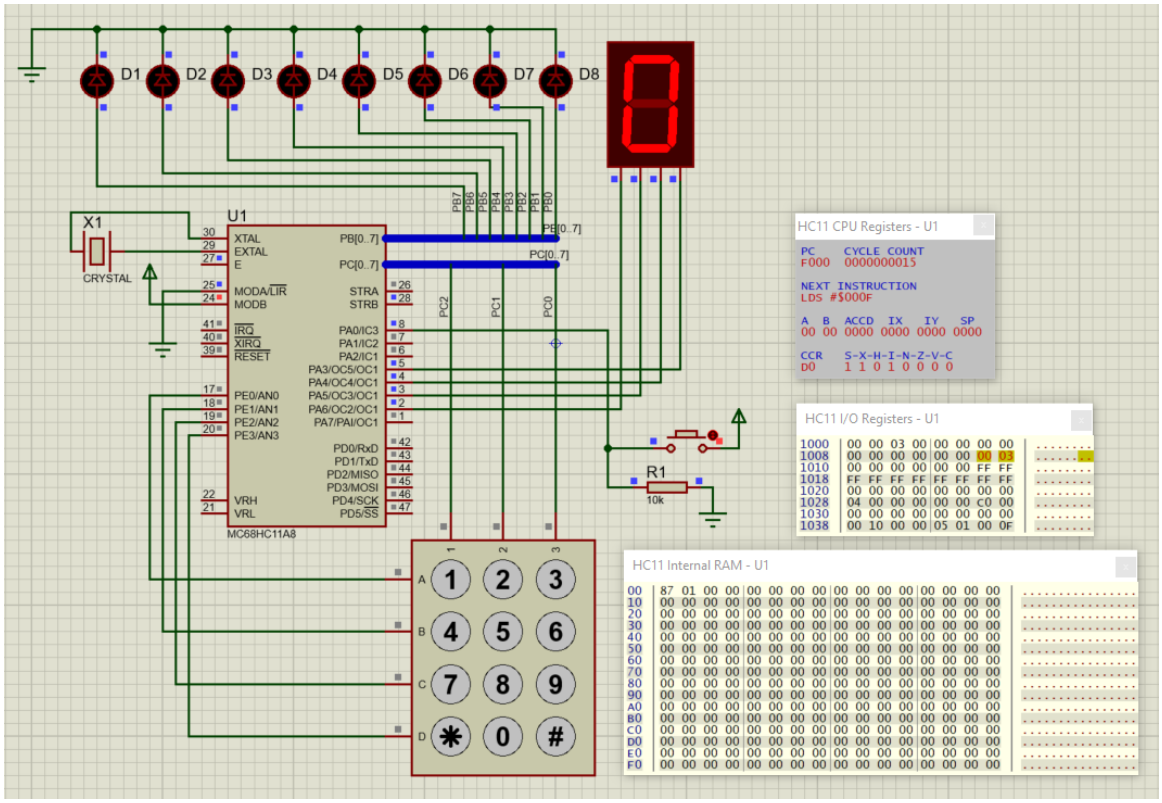

Рисунок 3.1 — Начальное состояние системы

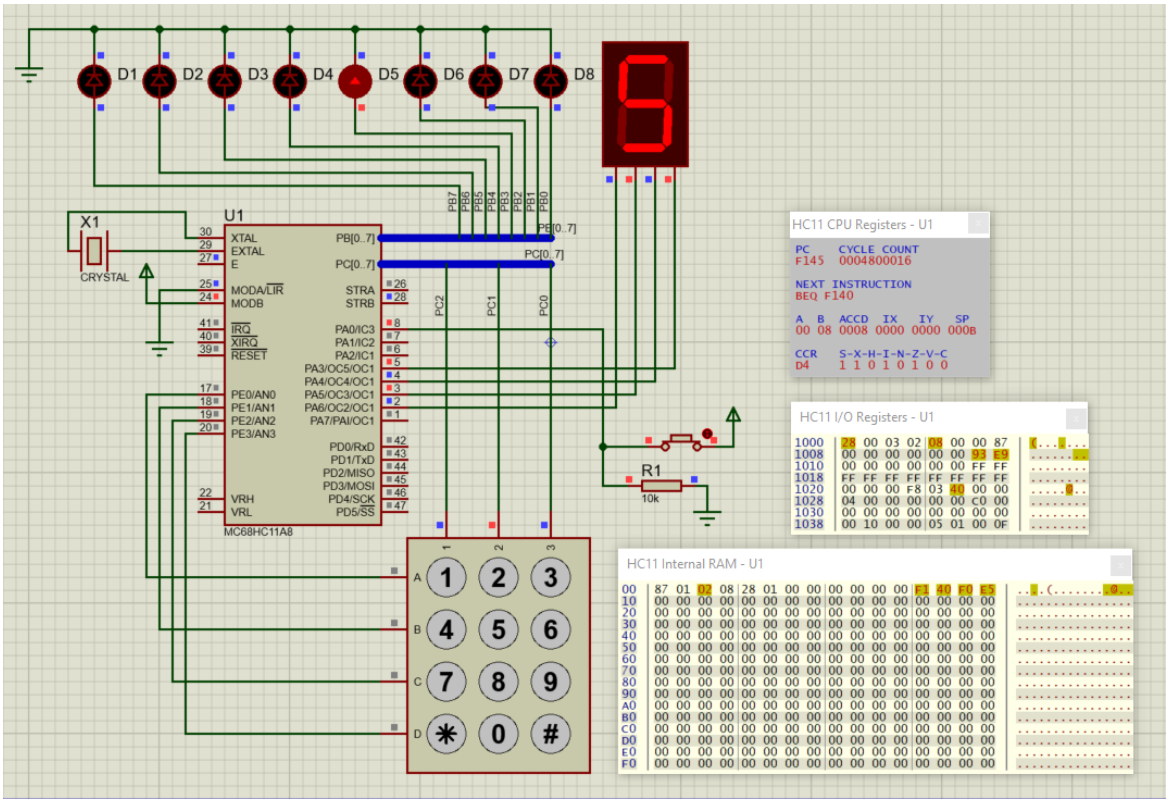

Рисунок 3.2 — Начало движения вправо с 5-го светодиода

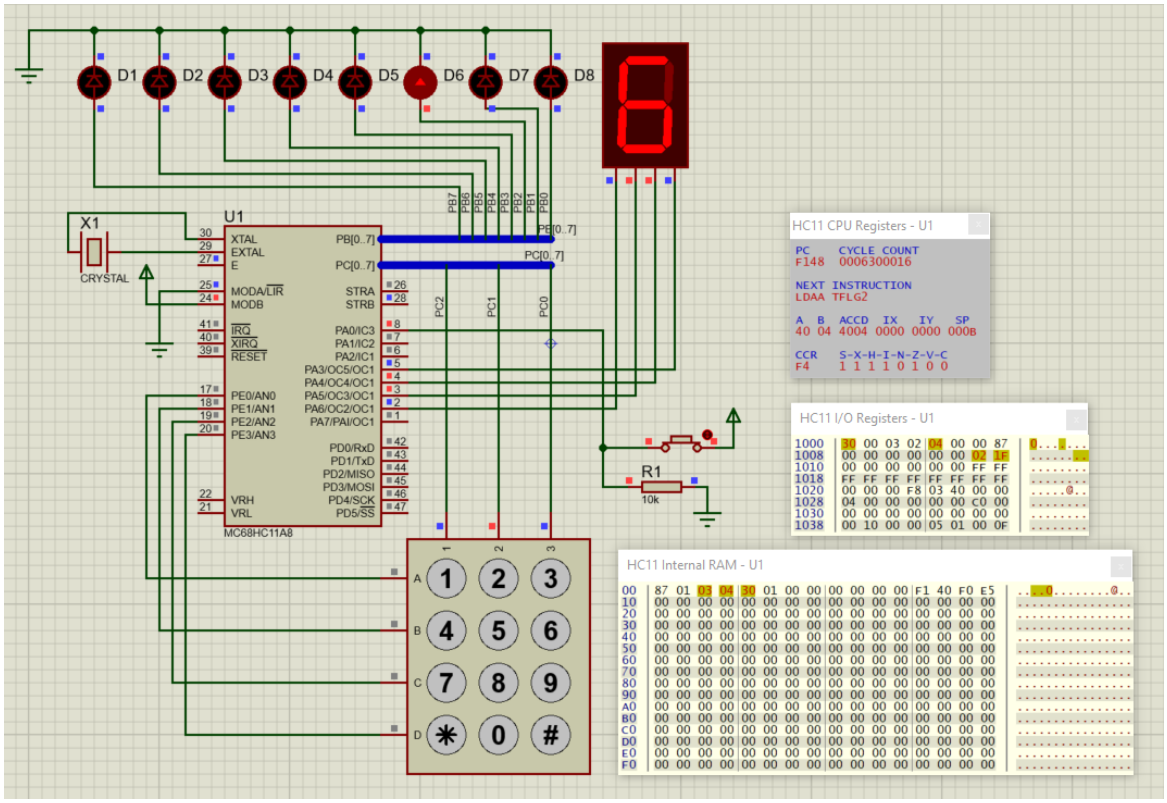

Рисунок 3.3 — Продолжение движения

4 ВЫВОД

При выполнении данной лабораторной работы мы ознакомились с принципами работы с матричной клавиатурой и семисегментным индикатором, а также с их взаимодействием с микроконтроллером. В результате чего была разработана программа, позволяющая поочередно включать и выключать светодиоды в заданном направлении на разработанной схеме.