ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТТЕЛЕКОММУНИКАЦИЙ
И ИНФОРМАТИКИ»

РАСЧЁТНО-ГРАФИЧЕСКАЯ РАБОТА

по дисциплине “Микропроцессорные системы”

Выполнил студент Киреев Илья Анатольевич
Ф.И.О.

Группы ИВ-121

Работу принял _____ Коновалов А. С.
подпись

Защищена _____ Оценка _____

Новосибирск – 2024

Содержание

Задание

Необходимо реализовать сигнальный таймер/счетчик с выводом на семисегментный индикатор. Данный таймер должен отсчитать введенное с клавиатуры время и по окончании отсчета издать звуковой сигнал. На каждый отсчет на семисегментном индикаторе должно изменяться показание счётчика цифр.

Выбор компонентов

В соответствии с заданием, выбраны следующие компоненты:

- Микроконтроллерная плата, которая выполняет функцию центрального управляющего устройства – Arduino Uno R3.
- Клавиатура 4x4, по нажатию на деления которой производится старт отсчёта.
- Семисегментный индикатор для отрисовки текущего значения счётчика.
- Пьезоэлемент – тривиальный генератор звуковых сигналов. Необходим для воспроизведения звука по окончанию отображения цифр в конце работы таймера.
- Малая макетная плата, необходима для лаконичного размещения вышеупомянутых элементов.
- Набор резисторов. Необходимы для подавления избыточного напряжения. Подключаются к индикаторам.
- Набор проводов для обеспечения связности элементов.

Разработка схемы устройства

Для разработки схемы устройства необходимо связать компоненты воедино:

- Семисегментный индикатор (Рис. 1):

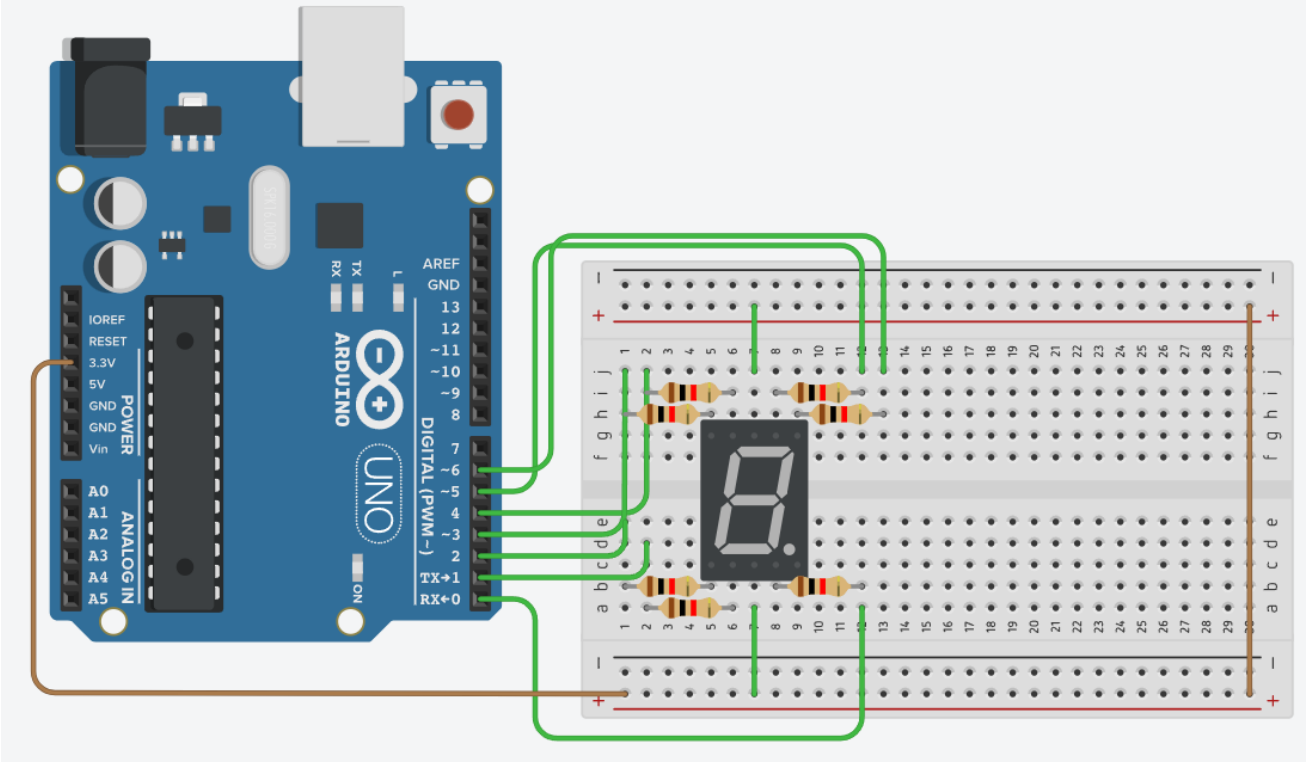

Рис. 1 – Постановка семисегментного индикатора

Выходы C, D, E, G, F, A, B (ножки) индикатора подключаются к PORTD номерам 0-6. На выходы типа “common” в свою очередь подаётся напряжение в 3.3V. Для подавления избыточного напряжения используются 7 резисторов.

- Пьезоэлемент (Рис. 2).

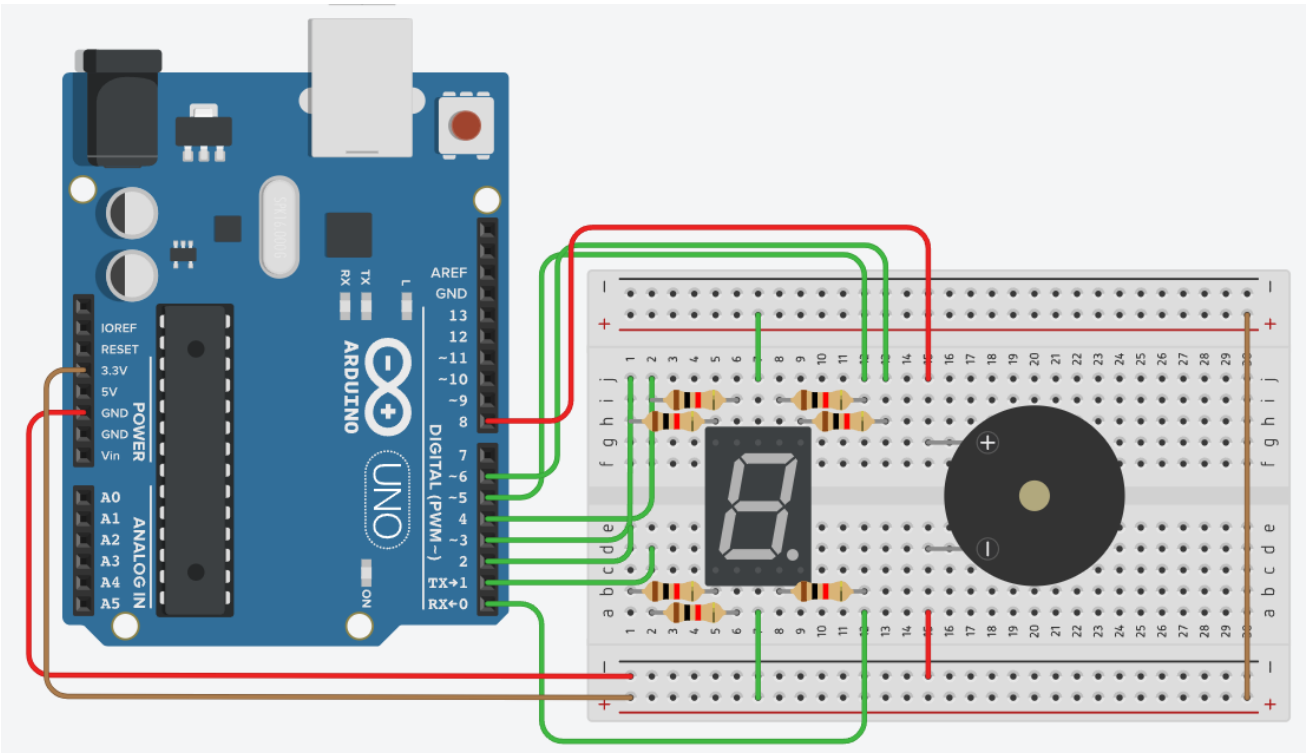

Рис. 2 – Постановка пьезоэлемента.

Пьезоэлемент подключается с контакта “+” – к PORTB-0, с контакта “-” – к GND.

- Клавиатура 4x4 (keypad) (Рис. 3)

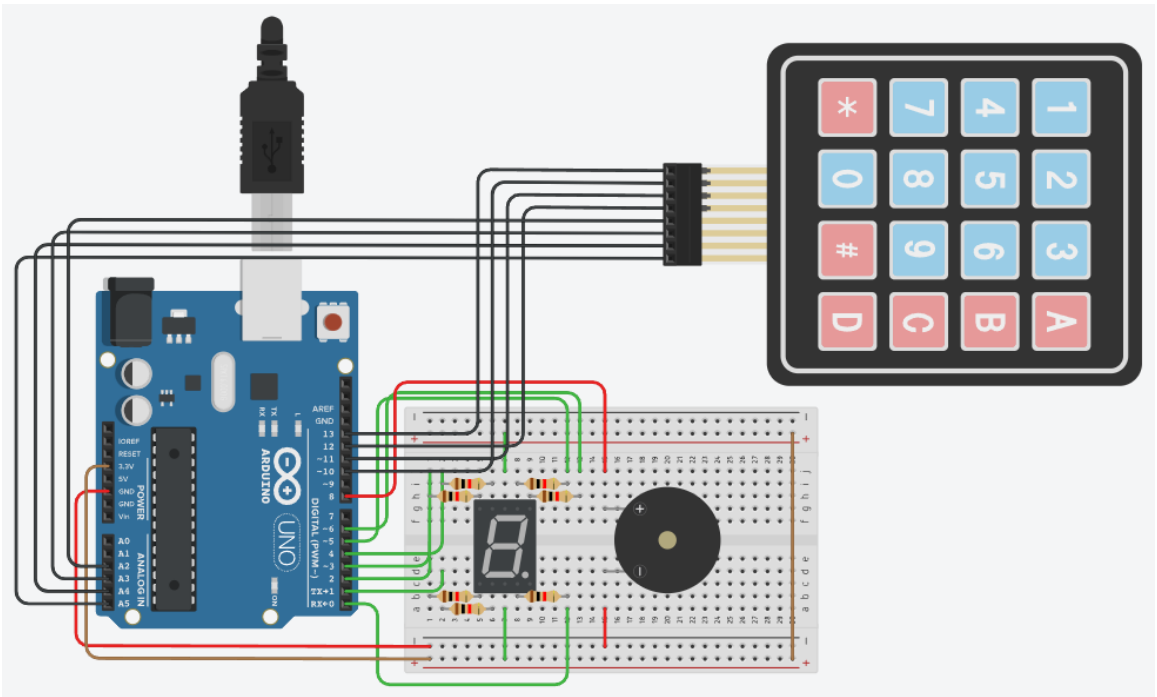

Рис. 3 – Постановка клавиатуры.

“Строки” 1-4 клавиатуры подключаются к PORTB 10-13.

“Столбцы” 1-4 клавиатура подключаются к A2-A5 соответственно.

Разработка управляющей программы

- Глобальные объекты

```
1  const int frequency = 1000;
2  const int timeNote = 200;
3  const int sec = 1000;
4
5  const int num[11][7] = {
6    {0,0,0,1,0,0,0}, // 0
7    {0,1,1,1,1,1,0}, // 1
8    {1,0,0,0,1,0,0}, // 2
9    {0,0,1,0,1,0,0}, // 3
10   {0,1,1,0,0,1,0}, // 4
11   {0,0,1,0,0,0,1}, // 5
12   {0,0,0,0,0,0,1}, // 6
13   {0,1,1,1,1,0,0}, // 7
14   {0,0,0,0,0,0,0}, // 8
15   {0,0,1,0,0,0,0}  // 9
16 };
17
18 const uint8_t ports[] = {0b11011111, 0b11101111, 0b11110111,
19 0b11111011};
20
21 const uint8_t pins[] = {0b00000100, 0b00001000, 0b00010000,
22 0b00100000};
23
24 const int numbers[3][4] = {{1, INT32_MIN, 7, 4}, {2, 0, 8, 5}, {3,
25 INT32_MIN, 9, 6}};
```

- frequency – предназначен для рассчёта полупериода звучания пьезоэлемента.
- timeNote – количество миллисекунд, которое будет звучать нота на пьезоэлементе.
- sec – количество миллисекунд в секунде.
- num – массив битовых представлений показа делений на семисегментном индикаторе.
- ports – массив “масок” для установления состояния портов.
- pins – массив “масок” для установления состояния пинов.
- numbers – массив, описывающий активные элементоы на клавиатуре.

- Установка начального состояния портов.

```
1 DDRB &= 0; PORTC &= 0;
2 DDRD  |= 0b11111111;
3 DDRB  |= 0b00000001;
4 DDRB  |= 0b00111100;
5 PORTC |= 0b00111100;
```

- Функция segReset.

```
1 void segReset() { PORTD &= 0; }
```

Предназначена для сброса PORTB в исходное состояние.

- Функция displayCurrentNum.

```
1 void displayCurrentNum(int numb) {
2   segReset();
3   for (int x = 0; x != 7; ++x) PORTD |= (num[numb][x] << x);
4 }
```

Предназначена для отображения текущей цифры на семисегментном индикаторе. В деления индикатора побитово поступают значения из массива num.

- Функция decrease.

```
1 void decrease(int numb)
2 {
3   for (int x = numb; x >= 0; x--) {
4     displayCurrentNum(x);
5     _delay_ms(sec);
6   }
7 }
```

Предназначена для посекундного счёта от указанного числа до нуля. Функция принимает число и посекундно в цикле отсчитывает значения от заданного числа до нуля, отображая их на индикаторе и сопровождая секундной задержкой.

- Функция sound.

```
1 void sound() {
2   uint16_t half_period = F_CPU / 8 / frequency / 4;
3   for (int j = 0; j != 5; ++j) {
4     for (int i = 0; i != timeNote; ++i) {
5       PORTB |= (1 << PB0); _delay_us(half_period);
6       PORTB &= ~(1 << PB0); _delay_us(half_period);
7     }
8     _delay_ms(500);
9   }
10 }
```

Предназначена для проигрывания звукового сигнала. После рассчёта полупериода, в цикле происходит воспроизведение сигнала, а затем – задержка. Таким образом, достигается эффект попеременного звучания.

- Основной цикл программы.

```
1   while (1) {
2     _delay_ms(10);
3
4     for (int i = 0; i != 3; ++i) {
5       for (int j = 0; j != 4; ++j) {
6         PORTB &= 0b11000011;
7         PORTB |= ports[j];
8
9         if (!(PINC & pins[i])) {
10          if (numbers[i][j] != INT32_MIN)
11          {
12            decrease(numbers[i][j]);
13            segReset();
14            sound();
15          }
16        }
17      }
18    }
19  }
```

Циклы на строках 4 и 5 предназначены для “обхода” массива numbers, а точнее, его столбцов и строк соответственно. На каждой итерации проверяется условия нажатия одной из кнопок клавиатуры, а затем, при определении её функциональности – производится обратный отсчёт, а затем – проигрывание мелодии.

Тестирование разработанной программы.

Ожидаемые результаты тестирования:

- После нажатия одной из функциональных кнопок клавиатуры (0, ..., 10) происходит вывод данного числа и всех, входящих в область [0, n], где n – значения числа, описывающее нажатую кнопку посекундно.
- По истечении данного времени воспроизводится звуковой сигнал.
- По истечении воспроизведения звукового сигнала схема снова готова к работе – вышеописанным действиям. Иными словами – входит в режим ожидания.

В ходе тестирования все пункты в ожидаемых результатах тестирования отработали корректно, без сбоев. Пример нажатия цифры “5” (Рис. 4-9).

Рис. 4-9 – Иллюстрация работы схемы.

Заключение

В ходе выполнения работы была сконструирована схема, описывающая сигнальный таймер, используя микроконтроллерную плату Arduino Uno R3, клавиатуру 4x4, пьезоэлемент, семисегментного индикатор, семь резисторов, малой макетной платы и набора проводов.

Была разработана управляющая программа, описывающая принцип работы сигнального таймера. Программа считывает введенное с клавиатуры число, производит отсчет от данного числа до нуля, после чего воспроизводит звуковой сигнал.

Полученное устройство протестировано. Необходимая функциональность достигнута.

Пиложение

```
#include <avr/io.h>
#include <util/delay.h>

const int frequency = 1000;
const int timeNote = 200;
const int sec = 1000;

const int num[11][7] = {
  {0,0,0,1,0,0,0}, // 0
  {0,1,1,1,1,1,0}, // 1
  {1,0,0,0,1,0,0}, // 2
  {0,0,1,0,1,0,0}, // 3
  {0,1,1,0,0,1,0}, // 4
  {0,0,1,0,0,0,1}, // 5
  {0,0,0,0,0,0,1}, // 6
  {0,1,1,1,1,0,0}, // 7
  {0,0,0,0,0,0,0}, // 8
  {0,0,1,0,0,0,0}  // 9
};

const uint8_t ports[] = {0b11011111, 0b11101111, 0b11110111,
0b11111011};

const uint8_t pins[] = {0b00000100, 0b00001000, 0b00010000,
0b00100000};

const int numbers[3][4] = {{1, INT32_MIN, 7, 4}, {2, 0, 8, 5}, {3,
INT32_MIN, 9, 6}};

void segReset() { PORTD &= 0; }

void displayCurrentNum(int numb) {
  segReset();
  for (int x = 0; x != 7; ++x) PORTD |= (num[numb][x] << x);
}

void decrease(int numb)
{
  for (int x = numb; x >= 0; x--) {
    displayCurrentNum(x);
    _delay_ms(sec);
  }
}

void sound() {
  uint16_t half_period = F_CPU / 8 / frequency / 4;
```

```
47    for (int j = 0; j != 5; ++j) {
48        for (int i = 0; i != timeNote; ++i) {
49        PORTB |= (1 << PB0); _delay_us(half_period);
50        PORTB &= ~(1 << PB0); _delay_us(half_period);
51        }
52      _delay_ms(500);
53    }
54  }
55
56  int main() {
57    DDRB &= 0; PORTC &= 0;
58    DDRD  |= 0b11111111;
59    DDRB  |= 0b00000001;
60    DDRB  |= 0b00111100;
61    PORTC |= 0b00111100;
62
63    while (1) {
64      _delay_ms(10);
65
66      for (int i = 0; i != 3; ++i) {
67        for (int j = 0; j != 4; ++j) {
68          PORTB &= 0b11000011;
69          PORTB |= ports[j];
70
71          if (!(PINC & pins[i])) {
72            if (numbers[i][j] != INT32_MIN)
73            {
74              decrease(numbers[i][j]);
75              segReset();
76              sound();
77            }
78          }
79        }
80      }
81    }
82  }
```