



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

Кибернетика

---

Кафедра проблем управления

---

**КУРСОВОЙ ПРОЕКТ (РАБОТА)**

по дисциплине: NOPE

Тема курсового проекта (работы): генератор DTMF сигнала

Студент группы **TOP SECRET**

**NO NAME** \_\_\_\_\_

Руководитель курсового проекта (работы)

Трипольский П.Э. \_\_\_\_\_

Работа представлена к защите «\_\_\_» \_\_\_\_\_ 20\_\_ г.

Допущен к защите «\_\_\_» \_\_\_\_\_ 20\_\_ г.



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Кибернетика

Кафедра проблем управления

Утверждаю

Заведующий кафедрой \_\_\_\_\_

Подпись

ФИО

«\_\_» \_\_\_\_\_ 20\_\_ г.

**ЗАДАНИЕ**

**на выполнение курсового проекта (работы) по дисциплине**

«>>

Студент

Группа

**Тема:**

**Исходные данные:** \_\_\_\_\_

**Перечень вопросов, подлежащих обработке, и обязательного графического материала:** \_\_\_\_\_

\_\_\_\_\_

**Срок представления к защите курсового проекта (работы):** до «\_\_» \_\_\_\_\_ 20\_\_ г.

**Задание на курсовой проект (работу) выдал**

\_\_\_\_\_  
Подпись руководителя Ф.И.О. руководителя

«\_\_» \_\_\_\_\_ 20\_\_ г.

**Задание на курсовой проект (работу) получил**

\_\_\_\_\_  
Подпись обучающегося Ф.И.О. исполнителя

## Содержание

Техническое задание .....	4
Аннотация .....	4
Требования к функциональным характеристикам программы .....	4
Требования к надежности программы .....	5
Порядок контроля и приемки.....	5
Описание разработанного ПО.....	5
Общие сведения.....	5
Аппаратные средства .....	6
Использование микроконтроллера .....	7
Алгоритмы функционирования программного обеспечения .....	8
Генерирование синусоидального сигнала .....	9
Сложение разных частот для генерации DTMF сигнала .....	10
Создание таблицы преобразования .....	11
.....	12
Методика тестирования .....	12
Исходный код.....	14

## **Техническое задание**

### **Аннотация**

- Наименование программы: генератор DTMF сигнала
- Назначение программы: генерировать требуемый DTMF сигнал в соответствии с нажатой на клавиатуре кнопкой. Клавиатура состоит из 12 кнопок, числа от 0 до 9 и специальных символов: «\*», «#»
- Краткая характеристика программы и области её применения: двухтональный многочастотный аналоговый сигнал (Dual Tone Multi-Frequency) применяется при автоматической телефонной сигнализации между устройствами.
- Используемые при разработке компиляторы: AVRASM

### **Требования к функциональным характеристикам программы**

- Требования к составу выполняемых функций: должна быть разработана функция определения нажатой кнопки, функция настройки прерываний микроконтроллера и функция для подсчета скважности ШИМ сигнала в текущий момент времени для генерирования требуемого сигнала.
- Организация передачи входных, выходных данных в программу и из программы, соответственно: к ножкам микроконтроллера подсоединены 12 кнопок, на вход INT0 подается сигнал, свидетельствующий о нажатии одной из них. Выходными данными является сигнал, генерирующийся при помощи ШИМ на выходе OC0B
- Временные характеристики работы программы: всё время, пока нажата одна из кнопок, должен генерироваться сигнал DTMF, когда не нажата ни одна из кнопок, на выходе устройства должно быть действующее напряжение 2.5В

## **Требования к надежности программы**

Информация, определяющая степень контроля за входными и выходными данными, реакция на ошибочные ситуации: поскольку единовременное нажатие пользователем двух кнопок невозможно, но возможно последовательное нажатие и удержание более одной кнопки, программа должна обрабатывать такие ситуации и реагировать на то нажатие, которое было совершено раньше.

## **Порядок контроля и приемки**

Первым проверяющим работоспособность программы является разработчик самой программы. Ему предоставляется доступ к клавиатуре, состоящей из 12 кнопок, и осциллографу, на котором отображается генерируемый сигнал. Лишь убедившись с помощью среды моделирования в правильности работы разработанной программы, в разрешении ошибочных ситуаций (одновременно нажато более одной кнопки), разработчик снимает данные с осциллографа в виде графиков и помещает их в отчет в соответствующий раздел для экспертной оценки преподавателем.

## **Описание разработанного ПО**

### **Общие сведения**

- Наименование программы: генератор DTMF сигнала
- Функциональное назначение и сведения о функциональных ограничениях на применение:
- Программное и аппаратное обеспечение, необходимое для функционирования программы: для функционирования программы необходимы компоненты, описанные в разделе «Аппаратные средства». Данные компоненты могут быть собраны в реальной схеме либо в одной из программ, позволяющих моделировать работу микроконтроллера AVR Attiny2313, например, Proteus.

## Аппаратные средства

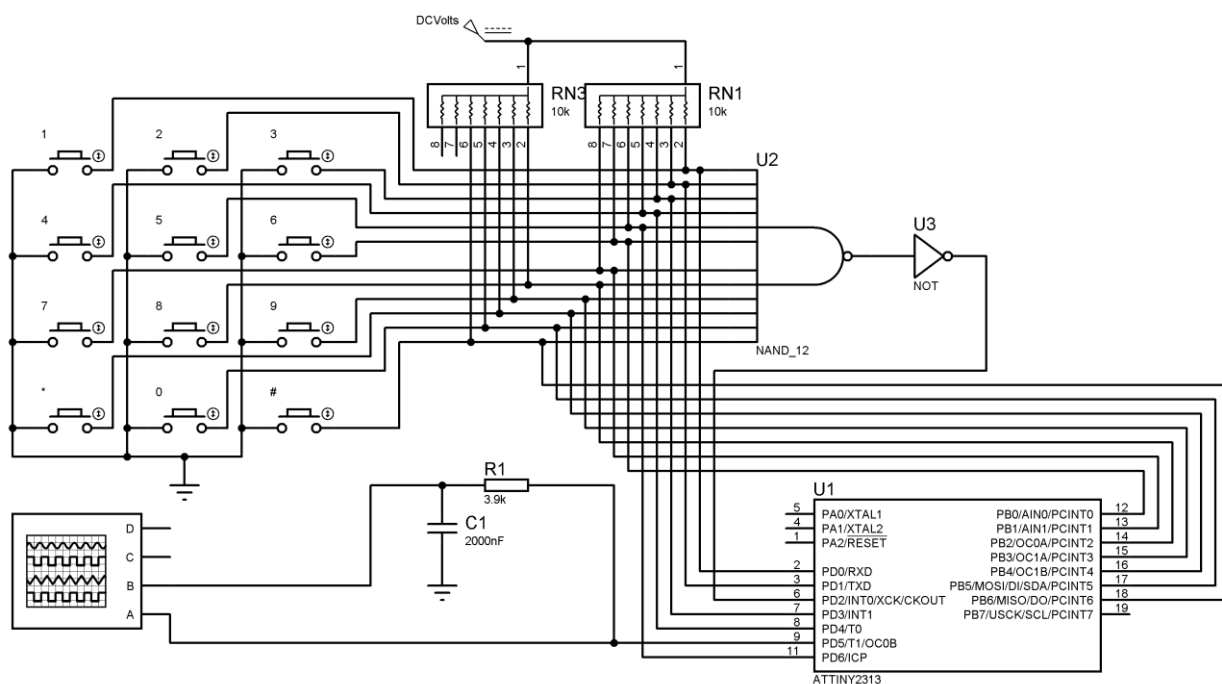


Рисунок 1. электрическая принципиальная схема устройства

Электрическая принципиальная схема устройства представленная на Рисунок 1 может быть условно разделена на несколько частей:

- Клавиатура и дополнительные компоненты (элементы «И-НЕ», «НЕ», резисторные сборки)
- Микроконтроллер Attiny2313
- Преобразователь ШИМ сигнала в действующее напряжением (резистор R1 и конденсатор C1)

Элемент «И-НЕ» на выходе изменяет значение с логического 0 на логическую 1 в том случае во время нажатия хотя бы одной из кнопок. Такой способ подключения клавиатуры позволяет настроить микроконтроллер таким образом, что опрос кнопок будет происходить только после нажатия одной из них. При ошибочном нажатии более одной кнопки, будем обработана та, что была нажата раньше.

Резисторные сборки необходимы для подключения кнопок к микроконтроллеру. Когда кнопки разомкнуты, выводы микроконтроллера через резисторные сборки подсоединены к питанию 5В. Когда кнопка нажата,

соответствующий вывод замыкается на землю. Таким образом, не существует ситуации, когда вывод микроконтроллера «подвешен» в воздухе, что в ряде случаев могло бы привести к наведению на нем напряжения.

Полный перечень используемых элементов представлен в таблице 1.

Таблица 1. Перечень элементов схемы

№	Название	Элемент	Описание
1	U1	Attiny2313	Микроконтроллер AVR компании Atmel
2	Button	Button	Кнопки клавиатуры
3	RN1-RN2	Резисторные сборки	Резисторные сборки. Сопротивление 10кОм, служат для «подтяжки» входов микроконтроллера к питанию 5В
4	R1	Резистор	Сопротивление 3.5кОм
5	C1	Конденсатор	2000nF, необходим для преобразования ШИМ сигнала в действующее напряжение для отображения на осциллографе
6	U2	Логический элемент «И-НЕ»	Когда нажата одна кнопка и более, на выходе элемента логическая 1, иначе – 0. Служит триггером для прерывания INT0
7	U3	Логический элемент «НЕ»	Инвертирует значение. Добавлен для более удобной настройки микроконтроллера

## Использование микроконтроллера

Таблица 2. Используемые выводы микроконтроллера

№	Порт	Направление	Описание
1	PD2	Вход	Изменение логического значения на этом выводе генерирует прерывание INT0
2	PD5	Выход	Выходной DTMF сигнал, генерируемый при помощи широтно-импульсной модуляции
3	PB0-PB6, PD0, PD1, PD3, PD4, PD6	Вход	Выводы подсоединены к кнопкам клавиатуры для возможности идентификации нажатой

Таблица 3. Список используемых периферийных устройств

№	Название	Описание
1	Таймер-счетчик	Служит для генерации DTMF сигнала
2	Внутренний тактовый генератор	Позволяет микроконтроллеру работать от внутреннего генератора частотой 8МГц.

Для генерирования на выходе PD5 синусоидального сигнала необходимо настроить таймер-счетчик на работу в режиме Phase Correct PWM и разрешить прерывания по переполнению, во время которых будет изменяться скважность (задается регистром OCR0B). Вывод PD5 задается как выход, при совпадении значений в регистре TCNT0 и OCR0B происходит изменение потенциала вывода. С логической 1 на логический 0 и наоборот в зависимости от того, инкрементируется значение в TCNT0 или декрементируется. Предделитель устанавливается равным 1.

Таблица 4. Регистры ввода вывода и их назначение

№	Регистр	Название	Назначение
1	R16	Temp	Вспомогательный регистр
2	R17	Temp2	Вспомогательный регистр
3	R18	Xlut_a	Указатель на элемент в массиве значений скважности ШИМ для синусоиды с частотой fa
4	R19	Xlut_b	Указатель на элемент в массиве значений скважности ШИМ для синусоиды с частотой fb
5	R20	Xsw_a	Ширина шага для генерации синусоиды частотой fa
6	R21	Xsw_b	Ширина шага для генерации синусоиды частотой fb
7	R22	Temp3	Вспомогательный регистр для промежуточных вычислений

### Алгоритмы функционирования программного обеспечения

Взаимодействие пользователя с клавиатурой (замыкание и размыкание контактов кнопок) приводит к изменению потенциала на выводе PD5, что приводит к срабатыванию прерывания INT0. В обработчике прерывания определяется, какое действие было совершено (нажали или отпустили кнопку). Если была нажата одна из кнопок, производится опрос клавиатуры



для определения этой кнопки. Если кнопку наоборот отпустили, то генерирование DTMF сигнала завершается, настройки таймер-счетчика сбрасываются к исходному состоянию.

### Генерирование синусоидального сигнала

Синусоидальный сигнал может генерироваться микроконтроллером при условии, что среднее значение напряжения, генерируемого ШИМ будет изменяться каждый период ШИМ. Соотношение между высоким и низким уровнями должно задаваться в соответствии с уровнем напряжения синусоидального сигнала при соответствующем времени. Данный процесс иллюстрируется на Рисунок 2. Исходные данные для ШИМ вычисляются для каждого её периода и записываются в таблицу преобразования.

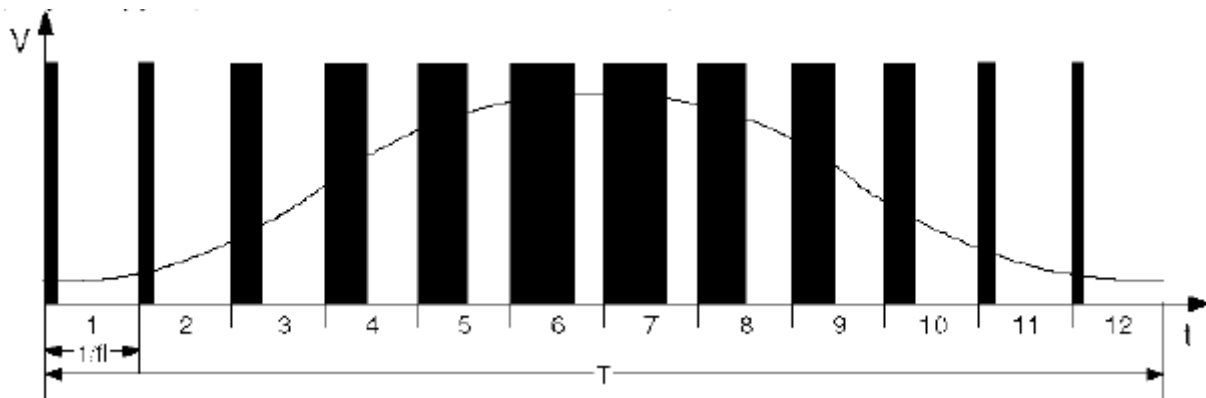


Рисунок 2. Изменение скважности ШИМ с каждым периодом ШИМ для генерирования синусоидального сигнала

Если из табличной выборки считывается каждое значение, то частота генерируемой синусоиды может быть посчитана по формуле (1):

$$f = \frac{f_l}{N_c} = \frac{f_{ck}/510}{N_c} \quad (1)$$

Где  $f_l$  – частота ШИМ ( $f_{ck} / 510$ )

$f_{ck}$  - частота тактового генератора

$N_c$  – количество выборок (=12 на Рисунок 2)

Если из табличной выборки считывать только каждое второе значение, то частота генерируемой синусоиды увеличится в два раза. Если каждое третье – в три раза и т.д. В общем случае шаг считывания  $X_{sw}$  определяет во сколько

раз частота синусоиды будет меньше, чем частота, посчитанная по формуле (1). Для определения шага  $X_{sw}$  используется формула (2):

$$X_{sw} = f \frac{N_c}{f_l} = \frac{f N_c 510}{f_{ck}} \quad (2)$$

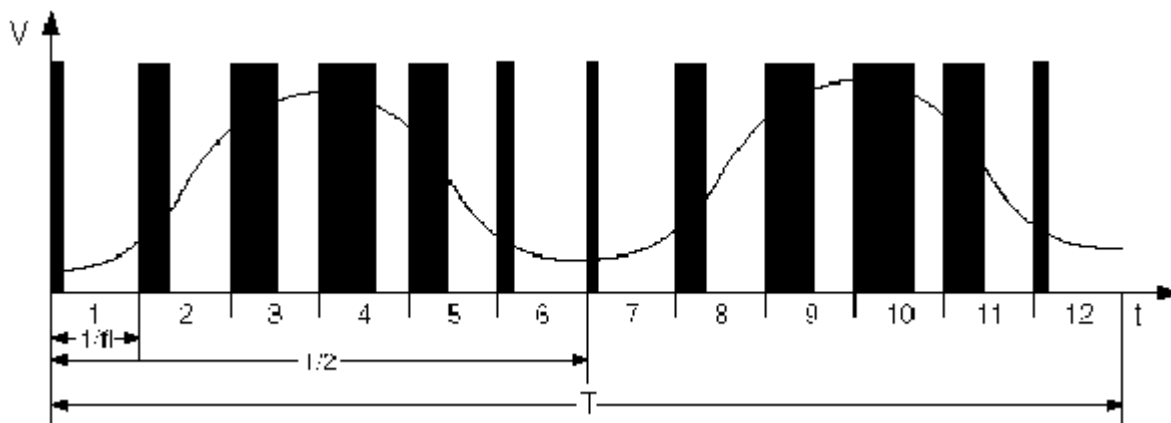


Рисунок 3. генерируемый синус при  $X_{sw} = 2$

Таким образом, для генерирования синусоидального сигнала заданной частоты на каждом периоде ШИМ необходимо считать значение из таблицы преобразования при помощи указателя и скопировать это значение в регистр OCR0B. Значение указателя на каждом периоде ШИМ может считаться по формуле (3):

$$X_{lut} = X'_{lut} + X_{sw} \quad (3)$$

$X'_{lut}$  – значение указателя на предыдущем шаге

$X_{lut}$  – текущее значение указателя

### Сложение разных частот для генерации DTMF сигнала

В общем виде DTMF сигнал может быть описан как сумма синусоидальных сигналов разных амплитуд и частот, ниже приведена таблица соответствия кнопок и частот.

Таблица 5. матрица формирования тонального сигнала

fb/fa	1209Гц ( $X_{sw} = 10$ )	1336Гц ( $X_{sw} = 11$ )	1633Гц ( $X_{sw} = 12$ )
697Гц ( $X_{sw} = 5$ )	1	2	3
770Гц ( $X_{sw} = 6$ )	4	5	6
852Гц ( $X_{sw} = 7$ )	7	8	9

941Гц ( $X_{sw} = 8$ )	*	0	#
------------------------	---	---	---

В общем виде формула может быть записана как:

$$f(t) = A_a \sin(2\pi f_a t) + A_b \sin(2\pi f_b t) \quad (4)$$

Где отношение амплитуд  $K = A_b/A_a$  исходных сигналов должно отвечать условию  $0.7 < K < 0.9$

С учетом уравнения (3), уравнение (4) может быть переписано в виде (5), а значение  $K$  принимается равным 0.75 для упрощения расчетов и замены операции деления побитовым сдвигом.

$$f(X_{lut}) = f(X_{luta}) + \frac{3}{4} f(X_{lutb}) \quad (5)$$

$F(X_{lut})$  – значение скважности ШИМ на каждом периоде, берется из таблицы преобразования.

$X_{lut}$  – указатель на значение в таблице преобразования.

### Создание таблицы преобразования

Для вычисления значения  $f(X_{lut})$  для последующей записи этих значений в память микроконтроллера используется формула (6):

$$f(x) = 63 + 63 \sin\left(\frac{2\pi x}{128}\right) \quad (6)$$

Результатом будет таблица, состоящая из чисел от 0 до 126. Выборка  $N_c = 128$ , каждое из значений задается 7 битами, сделано это из-за того, что результат сложения синусоид высокой и низкой частот имеет размер 1 байта.

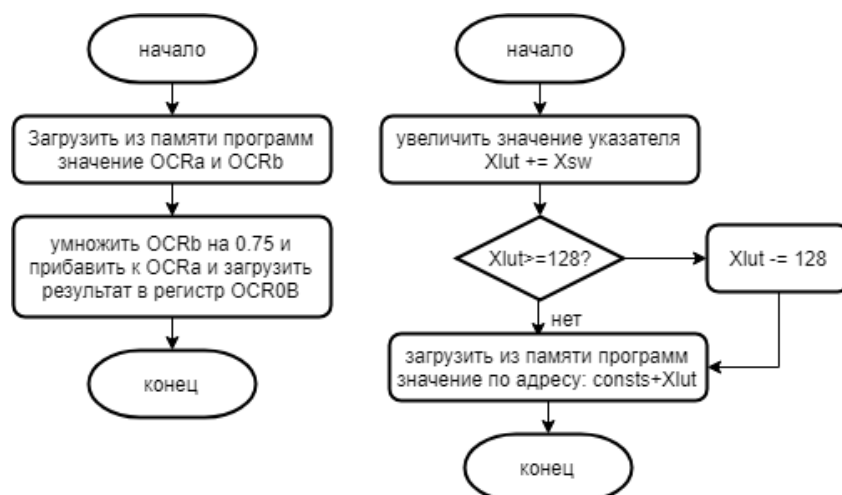


Рисунок 4. Обработчик прерывания по переполнению и функция получения значения из памяти программ

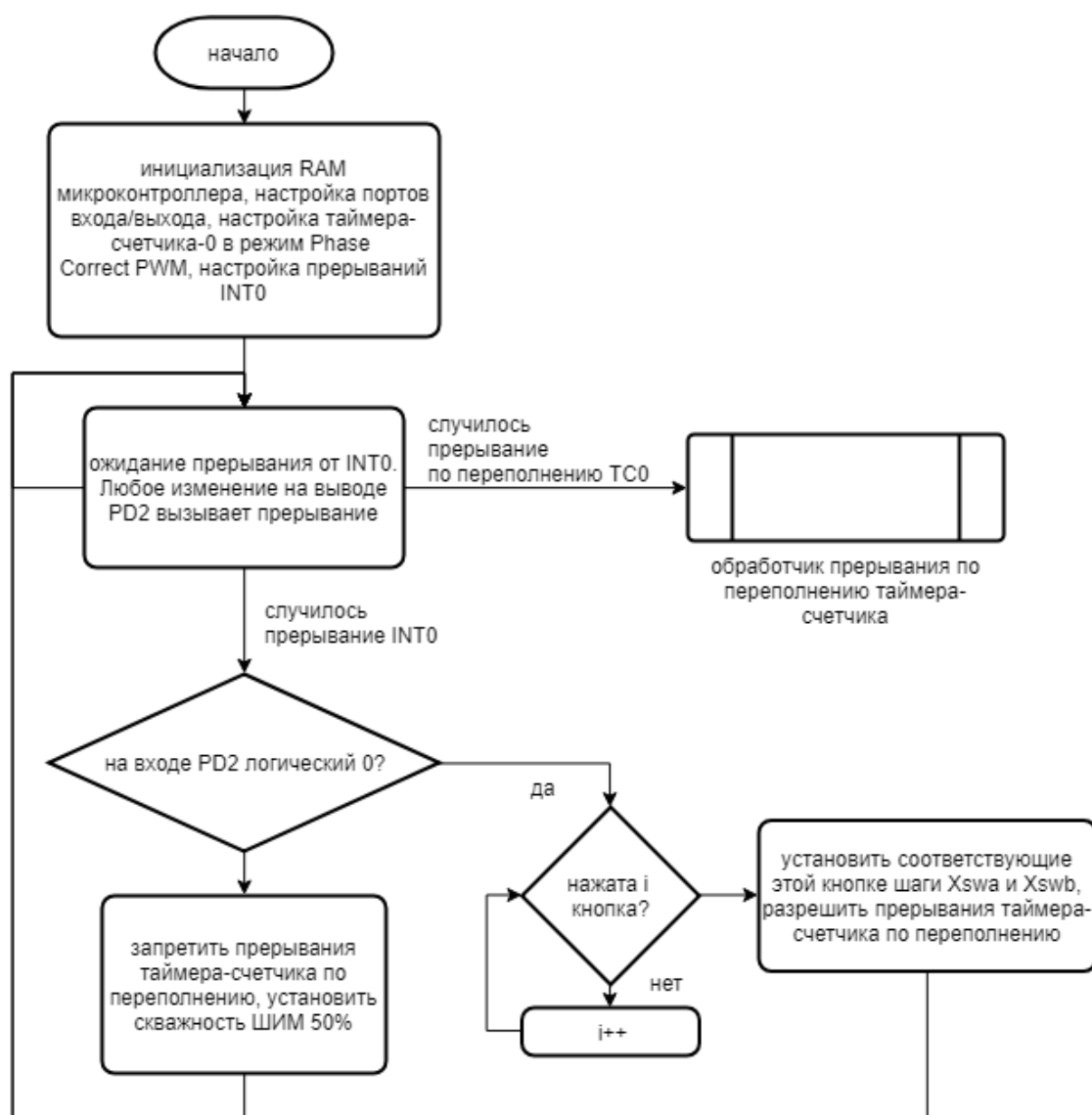


Рисунок 5. Алгоритм программы

### Методика тестирования

Во время проведения тестирования использовалась программы, позволяющие симулировать работу микроконтроллера и электрической-принципиальной схемы. Производились действия, которые могли произойти случайно или намеренно в ходе эксплуатации устройства, а именно: нажатие нескольких или всех кнопок одновременно, быстрое нажатие кнопки.

Каждое нажатие кнопки сопровождалось генерацией DTMF сигнала, соответствующего нажатой кнопке. Результат работы представлен на Рисунок 6 и Рисунок 7

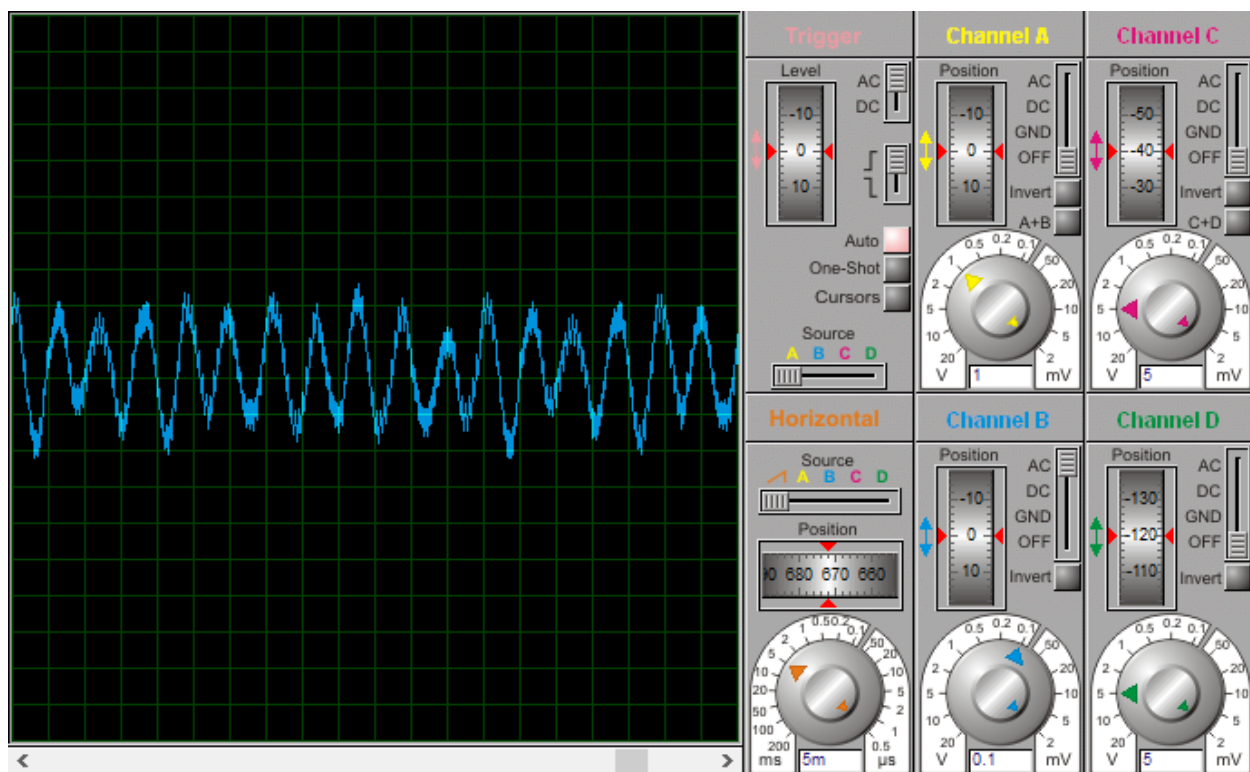


Рисунок 6. DTMF сигнал кнопки "5"

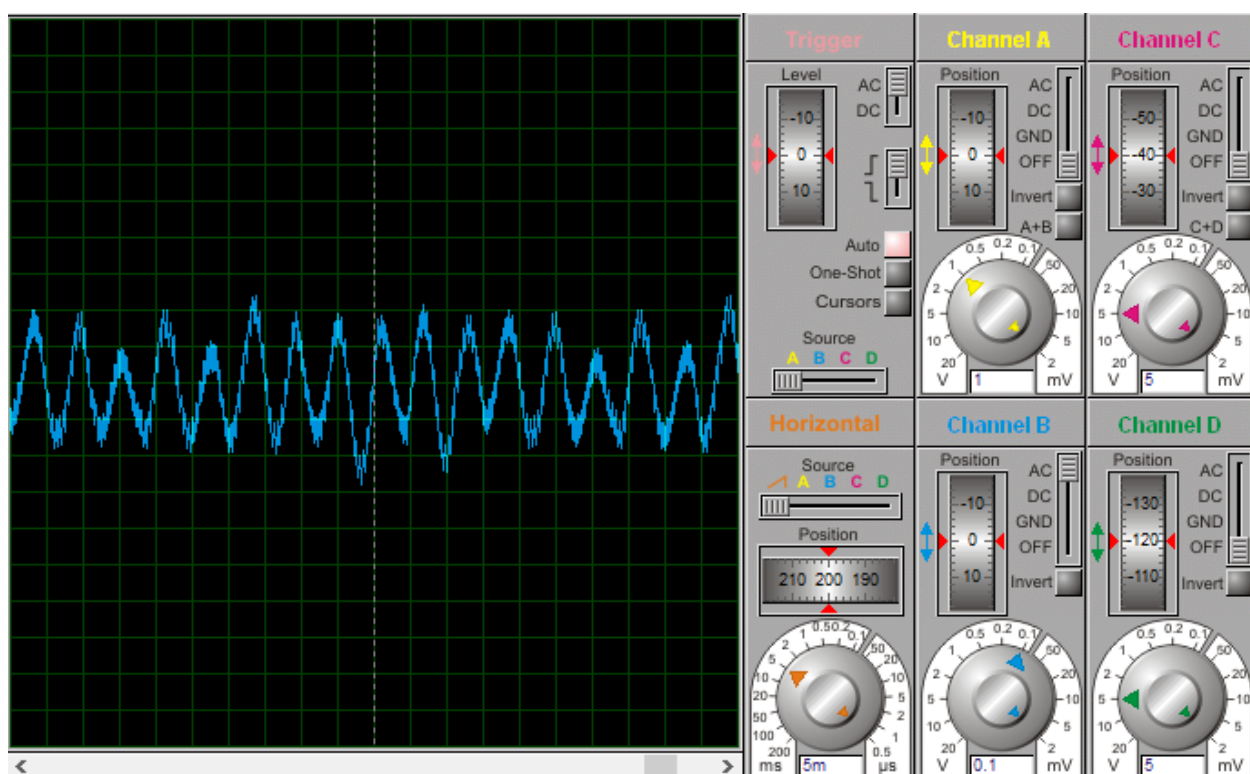


Рисунок 7. DTMF сигнал кнопки "\*"

Программа работает правильно, генерируемые сигналы для каждой из кнопок уникальны.

## Исходный код

```
;NAME: GENERATOR
;DATE 12.04.2019
.device Attiny2313

.def      temp = R16
.def      temp2 = R17
.def      XLut_a = R18      ; pointer on massive for a [0...127]
.def      XLut_b = R19      ; pointer on massive for b [0...127]
.def      Xsw_a = R20        ; step for a
.def      Xsw_b = R21        ; step for b
.def      temp3 = R22

.cseg
.org 0
rjmp      initialization
.org 0x0001
rjmp      INT0_Interr
.org 0x0006
rjmp      TC0_OVF

initialization:
    cli
    ldi          temp, LOW(RAMEND)
    out          SPL, temp

;IO PORTS
    ldi          temp, 0b00100000
    out          DDRD, temp
    clr          temp
    out          DDRB, temp

; prepare TC0 to work in PWM mode and enable OF interrupts
    ldi          temp, (1<<WGM00) | (1<<COM0B1) | (1<<COM0B0)
    out          TCCR0A, temp      ; phase correct PWM mode, TOP = 0xFF. Clear OCR0B on CM when up-counting, set
when down-counting
    ldi          temp, (1<<CS00)
    out          TCCR0B, temp      ; Prescaler = 1
    ldi          temp, (0<<TOIE0)
    out          TIMSK, temp      ; disable timer counter overflow

; set interrupt on INT0
    ldi          temp, (0<<ISC01) | (1<<ISC00)
    out          MCUCR, temp      ; any logical change of INT0 generates interrupt request
    ldi          temp, (1<<INT0)
    out          GIMSK, temp

    ldi          temp, 0x7F
    out          OCR0B, temp

    clr          Xsw_a
    clr          Xsw_b

    sei

loop:
    rjmp loop

TC0_OVF:
; some maths:
; OCR = f(Xlut_a) + 3/4 * f(Xlut_b)

    ldi          ZL, LOW(consts*2)
    ldi          ZH, HIGH(consts*2)

    ; get result in temp2
    rcall        getSample_a
    mov          temp, temp2      ; temp = f(Xlut_a)

    ldi          ZL, LOW(consts*2)
    ldi          ZH, HIGH(consts*2)
    ; get result in temp2
    rcall        getSample_b
    ; multiply temp2 on 3
    mov          temp3, temp2
    add          temp3, temp2
    add          temp3, temp2
    brcs        perenosWas      ; перейдет, если был перенос
```

```

cont:
    ; division on 4
    clc
    sbrs        temp3, 7
    sec
    ror         temp3
    lsr         temp3
    ; f(Xlut_a) + 3/4 f(Xlut_b)
    add         temp, temp3
    out         OCR0B, temp
    reti

perenosWas:
    sbr         temp3, 6
    rjmp        cont

; input: address in ZH:ZL
; output: result in temp2
getSample_a:
    ; Xlut_a = Xlut_a' + Xsw_a
    ; get data from table consts
    ; if Xlut_a >= 128 then Xlut_a = 0
    push        temp
    add         Xlut_a, Xsw_a
    cpi         Xlut_a, 128
    brsh        clearXa

cont1:
    add         ZL, Xlut_a
    clr         temp
    adc         ZH, temp

    lpm         temp2, Z
    pop         temp
    ret

clearXa:
    subi        Xlut_a, 128
    rjmp        cont1

; input: address in ZH:ZL
; output: result in temp2
getSample_b:
    ; Xlut_b = Xlut_b' + Xsw_b
    ; get data from table consts
    ; if Xlut_b > 128 then Xlut_b = 0
    push        temp
    add         Xlut_b, Xsw_b
    cpi         Xlut_b, 128
    brsh        clearXb

cont2:
    ; if Xlut_b >= 128 then Xlut_b = 0
    add         ZL, Xlut_b
    clr         temp
    adc         ZH, temp

    lpm         temp2, Z
    pop         temp
    ret

clearXb:
    subi        Xlut_b, 128
    rjmp        cont1

INT0_Interr:
    ; enable TOV interrupt request
    ; search pressed button
    ; if button pressed, set Xswa and Xswb
    ; Wave address in XH:XL

; enable TOV interrupt request
    sbic        PIND, 2
    rjmp        exit
    clr         temp
    out         TCNT0, temp
    ldi         temp, (1<<TOIE0)
    out         TIMSK, temp
    ; enable timer counter overflow

    clr         Xlut_a

```

```

        clr                Xlut_b

; search pressed button and set Xsw_a/Xsw_b
; произвести опрос кнопок, определить, какая нажата
; пропустить, если не нажата
        sbis               PIND, 0
        rjmp               Butt1
        sbis               PIND, 1
        rjmp               Butt2
        sbis               PIND, 3
        rjmp               Butt3
        sbis               PIND, 4
        rjmp               Butt4
        sbis               PIND, 5
        rjmp               Butt5 ; PINB
        sbis               PINB, 0
        rjmp               Butt6
        sbis               PINB, 1
        rjmp               Butt7
        sbis               PIND, 2
        rjmp               Butt8
        sbis               PIND, 3
        rjmp               Butt9
        sbis               PIND, 5
        rjmp               Butt0
        sbis               PIND, 4
        rjmp               Butt10
        sbis               PIND, 6
        rjmp               Butt11

        reti

; Table of Xsw:
; 1477Hz Xsw = 12 0x0C
; 1336Hz Xsw = 11 0x0B
; 1209Hz Xsw = 10 0x0A
; 941Hz  Xsw = 8  0x08
; 852Hz  Xsw = 7  0x07
; 770Hz  Xsw = 6  0x06
; 697Hz  Xsw = 5  0x05

Butt1:
        ldi                Xsw_a, 0x0A
        ldi                Xsw_b, 0x05
        reti

Butt2:
        ldi                Xsw_a, 0x0B
        ldi                Xsw_b, 0x05
        reti

Butt3:
        ldi                Xsw_a, 0x0C
        ldi                Xsw_b, 0x05
        reti

Butt4:
        ldi                Xsw_a, 0x0A
        ldi                Xsw_b, 0x06
        reti

Butt5:
        ldi                Xsw_a, 0x0B
        ldi                Xsw_b, 0x06
        reti

Butt6:
        ldi                Xsw_a, 0x0C
        ldi                Xsw_b, 0x06
        reti

Butt7:
        ldi                Xsw_a, 0x0A
        ldi                Xsw_b, 0x07
        reti

Butt8:
        ldi                Xsw_a, 0x0B
        ldi                Xsw_b, 0x07
        reti

Butt9:
        ldi                Xsw_a, 0x0C
        ldi                Xsw_b, 0x07
        reti

Butt0:
        ldi                Xsw_a, 0x0B

```



```

        ldi            Xsw_b, 0x08
        reti

Butt10:
        ldi            Xsw_a, 0x0A
        ldi            Xsw_b, 0x08
        reti

Butt11:
        ldi            Xsw_a, 0x0C
        ldi            Xsw_b, 0x08
        reti

exit:
        push          temp
        ldi            temp, (0<<TOIE0)
        out            TIMSK, temp                ; disable timer counter overflow
        ldi            temp, 0x7F
        out            OCR0B, temp
        pop            temp
        reti

.org 0x00F0
        consts: .DB 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 95, 98, 101, 103, 105, 108, 110, 112, 114, 115, 117, 119, 120, 121, 122, 123, 124, 125,
125, 126, 126, 126, 126, 126, 125, 125, 124, 123, 122, 121, 120, 119, 117, 115, 114, 112, 110, 108, 105, 103, 101, 98, 95, 93, 90, 87, 84, 81, 78,
75, 72, 69, 66, 63, 60, 57, 54, 51, 48, 45, 42, 39, 36, 33, 31, 28, 25, 23, 21, 18, 16, 14, 12, 11, 9, 7, 6, 5, 4, 3, 2, 1, 1, 0, 0, 0, 0, 0, 1, 1, 2, 3, 4, 5, 6,
7, 9, 11, 12, 14, 16, 18, 21, 23, 25, 28, 31, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60

```