Министерство образования и науки, молодежи и спорта Украины

Харьковский Национальный Университет Радиоэлектроники

Курсовая работа

Программно управляемый генератор линейно-нарастающего напряжения сверхнизкой частоты на микроконтроллере

Харьков 2011

**Введение**

Генераторы линейно – изменяющегося напряжения называют иногда генераторами развёртки, хотя этот термин не отражает их гораздо более широкого применения. Из области разверток заимствованы названия двух основных частей пилообразного импульса: прямой ход (главный, почти линейный участок t п ) и обратный ход (сравнительно короткий участок t о, форма которого обычно несущественна).

Пилообразное напряжение это такое напряжение, которое нарастает или спадает линейно в течение некоторого отрезка времени, называемого временем рабочего хода tо достигает первоначального значения. Такое напряжение используется устройствах сравнения, для горизонтальной развёртки электронного луча в электронно-лучевой трубке в других устройствах. Возврат луча в исходное положение должен происходить, возможно, быстрее, вследствие чего спадающий участок пилообразного напряжения должен иметь большую крутизну и малую продолжительность.

Пилообразные импульсы можно получить с помощью любого релаксатора: мультивибратора, одно вибратора или блокинг-генератора. Поэтому генераторы пилообразного напряжения составляют особый класс импульсных устройств и заслуживают специального рассмотрения. Генераторы линейно-изменяющегося напряжения являются широко известными устройствами импульсной техники. В данной курсовой работе будет рассмотрено создание генератора линейно-спадающего напряжения на МК ATMega8515.

**1. Обзор аналогичных устройств**

# Простейший генератор пилообразного напряжения (ГПН)

В простейшем случае, когда не требуется высокая линейность рабочего участка выходного напряжения, применяют заряд (рисунок 1.1,а) или разряд конденсатора через резистор R. После размыкания ключа Кл конденсатор заряжается по закону:

u=E(1-e -t/τ), где τ=RC.

Если во время рабочего хода использовать лишь начальный участок экспоненты, т.е. при tраб<<τ, или, другими словами, при Um<<E, можно считать u(t) при 0≤t≤tраб линейно изменяющимся напряжением. Учитывая, что iнач=E/R, а iкон=(E-Um)/R, находим коэффициент нелинейности:

ε=Um/E. (1.1)

Можно определить ε и по формуле

ε=Um/E=1-e-tраб/τ≈ tраб/τ.

Из (1.1) следует, что коэффициент нелинейности ε оказывается равным Um/E. Обычно это соотношение называется коэффициентом использования источника питания. При этом для получения достаточно малого значения ε приходится выбирать значение E во много раз большим амплитуды Um т.е. плохо использовать напряжение источника питания. Таким образом, простейшая схема с зарядом или разрядом конденсатора через резистор оказывается пригодной лишь при сравнительно невысокой линейности (примерно 10%).

Принципиальная схема простейшего ГПН с транзисторным ключом и соответствующие временные диаграммы напряжения приведены на рисунке 1.1б и 1.1в согласно[2].

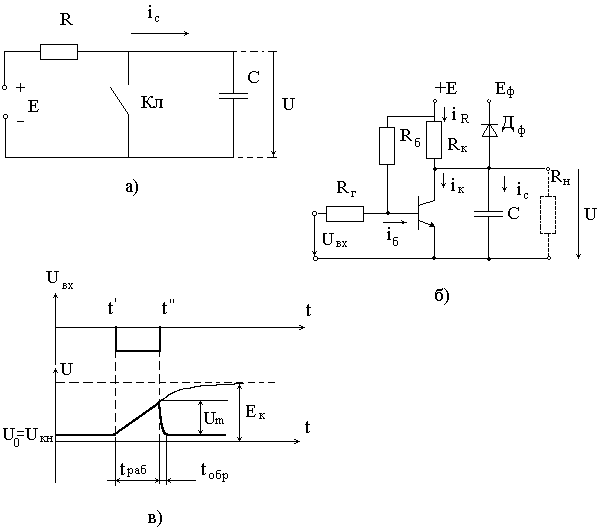


Рисунок 1.1-Принципиальная схема ГПН с транзисторным ключем и соответствующие временные диаграммы

# Классификация ГПН со стабилизаторами тока

Как уже указывалось выше, принцип действия схем генераторов пилообразного напряжения заключается в использовании заряда или разряда конденсатора во время рабочего хода через стабилизатор тока. Учитывая принципиальную общность почти всех применяемых на практике схем генераторов, целесообразно рассматривать их как варианты одной и той же схемы. При этом они отличаются друг от друга, главным образом, лишь способом создания напряжения в цепи стабилизатора тока. Согласно [2] по этому классификационному признаку различают следующие типы генераторов:

1). Генераторы, в которых стабилизатор тока реализован в виде отдельного структурного элемента со специальным источником напряжения Ест.

2). Генераторы, в которых источник напряжения Ест стабилизатора тока реализован в виде заряженного конденсатора. Необходимо отметить, что этот генератор по другому классификационному признаку часто относят к группе компенсационных устройств. Идея построения таких устройств основана на том, что стабилизация зарядного (или разрядного) тока конденсатора С может быть достигнута, если последовательно с ним включить источник, напряжение которого изменяется по тому же закону, что и на конденсаторе С, но имеет обратную полярность. Роль такого источника напряжения выполняет усилитель. В зависимости от способа включения усилителя различают схемы с положительной и отрицательной обратной связью.

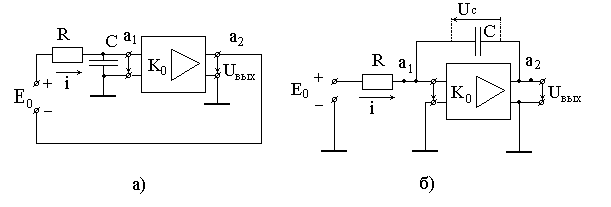


Рисунок 1.2- компенсационный генератор с ООС (а), с ПОС(б).

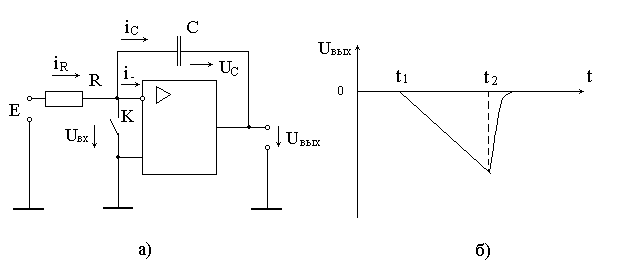
На рисунке 1.2а показан вариант функциональной схемы компенсационного генератора с положительной обратной связью (ПОС): если коэффициент усиления усилителя К0=+1, то повышение потенциала в точке а1 при заряде конденсатора С компенсируется точно таким же повышением потенциала в точке а2, и зарядный ток i останется неизменным. Конечно, в практических схемах вследствие того, что коэффициент усиления К0 не остается в процессе работы постоянным и точно равным 1, а так же в результате нестабильности других параметров схем наблюдается определенное не постоянство тока i и большее или меньшее значение коэффициента нелинейности напряжения на конденсаторе и выходного напряжения uвых. Генератор, реализующий функциональную схему на рисунке 1.2а, называют компенсационным генератором с ПОС.

3). Генераторы, в которых роль источника напряжения стабилизатора тока выполняет источник питания схемы. Такой генератор по другому классификационному признаку относится к компенсационным генераторам с ООС (отрицательной обратной связью); функциональная схема такого генератора показана на рисунке 1.2б.

# 1.3 Генераторы пилообразного напряжения на операционных усилителях

Интегрирующее включение операционного усилителя, обеспечивающего получение выходного напряжения, пропорционального интегралу от входного напряжения, предполагает включение конденсатора в цепь отрицательной обратной связи. Поэтому генераторы пилообразного напряжения на операционных усилителях строят по принципу генераторов с обратной связью, интегрирующих постоянное напряжения источника питания, которое для них является входным.

На рисунке 1.3а показана схема генератора пилообразного напряжения с интегрирующей RC-цепочкой, включенной в цепь отрицательной обратной связи операционного усилителя.



**2. Обоснование выбранного варианта технического решения**

Преимущества реализации генератора линейно нарастающего напряжения на микроконтроллере:

-компактность прибора;

-возможность программного управления устройством;

-простота схемного решения;

-точность формирования импульсов различной частоты;

-простое управление режимом работы с помощью клавиатуры;

-отображение частоты генерируемых импульсов на ЖКИ.

Схема разрабатываемого генератора приведена на чертеже РТ 468.524.008 ЭЗ. В схеме применяется микроконтроллер фирмы ATMEL – ATMega8515. Микросхема создана на основе ядра Mega, имеет в своём составе 8-разрядный и 16-разрядный таймеры для формирования временных задержек и 4 порта для ввода и вывода информации с внешних устройств.

Основные технические характеристики микроконтроллера:

-Форма входного сигнала -произвольная

-Максимальная амплитуда входного сигнала, 20В

-Дискретность отсчета частоты, 10Гц

-Время измерения, 100мс

-Период повторения измерений, 200 мс

-Напряжение питания, 5В

Цифровое значение амплитуды необходимо подавать на ЦАП, который преобразовывает его в аналоговую форму. В работе был использован параллельный восьмиразрядный ЦАП AD5601 фирмы Analog devices.

Регулировка амплитуды производится потенциометром R1, подключенным ко входу управления ЦАП. Так как нагрузка низвоомная( 50 ОМ), то для согласования сопротивления нагрузки с выходом устройства, применяется эмитерный повторитель на транзисторе 2N636. Для преобразования выходного сигнала микроконтроллера из уровня TTL в более мощный сигнал, а также для согласования устройства с нагрузкой (генератор работает на низкоомную нагрузку), используется выходной каскад. В качестве выходного каскада применён транзистор VT1. При появлении на выводе 6 импульса, он поступает через резистор R4 на базу VT1. В связи с тем, что величина импульса значительно меньше напряжения на эмиттере, т.е. база имеет менее положительный потенциал, транзистор открывается на время, равное длительности импульса. Резистор R4 ограничивает выходной сигнал микроконтроллера до значения, необходимого для нормальной работы транзистора VT1. В то время, когда открыт транзистор, в нагрузке протекает ток источника питания, т.ё. на нагрузке появляется импульс с амплитудой, равной амплитуде напряжения питания. Таким образом, осуществляется усиление последовательности импульсов. Так как выходом является коллектор, а такая схема включения имеет низкое выходное и достаточно высокое входное сопротивление, то этот каскад успешно выполняет функцию согласования выхода микроконтроллера с нагрузкой, так как по ТЗ генератор должен работать на низкоомную нагрузку.

Применены непроволочные тонкослойные металлодиэлектрические резисторы марки C2-23 постоянного сопротивления. У данных резисторов в качестве резистивного элемента используется тонкая плёнка слоя проводящего металла, поэтому они характеризуются повышенной теплостойкостью. Резисторы имеют невысокую стоимость.

Питание генератора осуществляется с помощью трансформатора, выпрямительного мостика и интегрального стабилизатора 78LR05.

Изменение частоты сигнала осуществляется матрицей кнопок 4х4.

Индикация частоты осуществляется алфавитно-цифровым ЖКИ с контроллером HD44780. Так как необходимо выводить не более 16ти символов, занимающих одну строку, был выбран символьный ЖКИ формата 16х1 WH1601B фирмы Winstar.

**3. Разработка принципиальной схемы устройства**

Разрабатываемое устройство должно генерировать линейно-нарастающее напряжение с дискретно-изменяющейся частотой и плавно-изменяющейся амплитудой. Принципиальная схема устройства приведена на чертеже РТ 468.524.008 ЭЗ.

Для дискретного изменения частоты будет использована матричная клавиатура 4х4, подключенная к порту D микроконтроллера. К кнопкам клавиатуры програмно подключены подтягивающие резисторы во избежание «обрыва» на входе. Подтяжка работает следующим образом: в ненажатом состоянии сопротивление между нулём и входом очень велико, и через резистор на входе создаётся потенциал, воспринимаемый МК как логическая единица. При нажатии картина меняется: теперь резистор - относительно бесконечное сопротивление, а на пине - потенциал нуля. Клавиатура будет обеспечивать изменение частоты от 2 Гц до 32 Гц с шагом 2 Гц. По умолчанию будет установлен режим генерации частоты 2Гц.

Цифровое значение амплитуды необходимо подавать на ЦАП, подключенный к порту С микроконтроллера.

Регулирование амплитуды производится переменным резистором R1, подключенным ко входу управления ЦАП.

Отображение генерируемой частоты в данном режиме будет осуществляться ЖКИ. Шина передачи данных на ЖКИ подключена к порту А микроконтроллера. Управляющие сигналы подключены к ножкам РВ0, РВ1, РВ2.

Питание генератора осуществляется с помощью трансформатора, выпрямительного мостика и интегрального стабилизатора 78LR05.

**4. Разработка алгоритма и программы функционирования устройства**

Алгоритм главной программы включает в себя следующие блоки:

-инициализация МК. В данном блоке осуществляется инициализация стека, настройка портов(на ввод/вывод данных), конфигурирование первого таймер-счетчика, настройка/разрешение прерываний, загрузка констант. Ниже приведен листинг инициализации МК:

|  |  |
| --- | --- |
| .include "D:\0B54~1\9E9E~1\VMLAB\include\m8515def.inc"  .def temp = r16  .def adr = r17  .def reg = r18  .def const0 = r19  .def const01 = r20  .def time = r21  .def lev =r22  .def Comand =r23  .def lcd =r24  .def x =r25  .equ E = 2  .equ RW = 1  .equ RS = 0  reset:  rjmp start  reti ; Addr $01  reti ; Addr $02  reti ; Addr $03  rjmp generation  reti ; Addr $05  reti ; Addr $06  reti ; Addr $07  reti ; Addr $08  reti ; Addr $09  reti ; Addr $0A  reti ; Addr $0B  reti ; Addr $0C  reti ; Addr $0D  reti ; Addr $0E  reti ; Addr $0F  reti ; Addr $10 | ; Program starts here after Reset  start: ldi temp,$02  out SPH,temp  ldi temp,$5F  out SPL,temp  ;====== настройка портов ===========  ldi temp,$F0  out DDRD,temp  ldi temp,$FF  out PORTD,temp  out DDRC,temp  out PORTC,temp  ;===== настройка Т1 и прерываний ====  ldi temp, $40  out TIMSK, temp  ldi temp, $09  out TCCR1B, temp  sei  ;======== загрузка констант ========  ldi const0,0  ldi const01,$01  ldi lev,$00  ldi x, $00  ;===== режим по умолчанию ========  ldi temp, high($1E78)  out OCR1AH, temp  ldi temp, low($1E78)  out OCR1AL, temp  ldi adr,0 |

-инициализация ЖКИ. В данном блоке реализуется инициализацию ЖКИ: определение направления отображения, выбор направления сдвига и настройка развертки в одну строку. Настройка каждого из параметров происходит по следующему алгоритму:

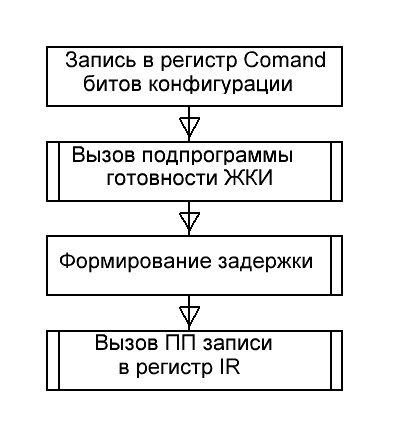


Рис. 4.1 — Алгоритм последовательности конфигурации ЖКИ

Листинг инициализации ЖКИ

|  |  |
| --- | --- |
| ldi Comand,$0C ;определение режима отображения  rcall gotov  rcall LCD\_Delay  rcall zap\_IR  ldi Comand,$06 ; выбор направления сдвига  rcall LCD\_Delay  rcall gotov | rcall LCD\_Delay  rcall zap\_IR  ldi Comand,$38 ;развертка в одну строку  rcall LCD\_Delay  rcall gotov  rcall LCD\_Delay  rcall zap\_IR |

-установка режима по умолчанию. Данный блок алгоритма включает в себя установку частоты по умолчанию, а также установку индикации по умолчанию. Листинг установки режима и индикации по умолчанию:

|  |  |
| --- | --- |
| ldi temp, high($1E78)  out OCR1AH, temp  ldi temp, low($1E78)  out OCR1AL, temp  ;=====Вывод надписи==============  ldi lcd,$46 ;F  rcall mm3  ldi lcd,$20  rcall mm3  ldi lcd,$3D ;=  rcall mm3 | ldi lcd,$20  rcall mm3  ldi lcd,$20  rcall mm3  ldi lcd,$32 ;2  rcall mm3  ldi lcd,$20  rcall mm3  ldi lcd,$48 ;H  rcall mm3  ldi lcd,$7A ;z |

После инициализации МК, ЖКИ и установки режима по умолчанию следует реализовать динамический опрос клавиатуры, который представляет собой следующую последовательность действий: группы кнопок условно разбиты на "линейки" и "колонки". Сначала программно на выходах PВ0-PВ3 выставляются единицы, а на P40 - ноль. При этом включена первая колонка, и осуществляется проверка нажатия кнопок именно этой колонки. Затем первая колонка отключается, и подключается следующая, и т.д.

Алгоритм опроса клавиатуры, который является частью главной программы, указан в чертеже РТ 468.524.008 П3, а ниже можно ознакомиться с листингом динамического опроса клавиатуры:

|  |  |
| --- | --- |
| forever:  ldi reg,$7F ;маска 01111111  out PORTD,reg  nop  nop  sbis PIND,0 ;проверка нажатия кнопки  rjmp r3;переход к метке нажатия кнопки  sbis PIND,1  rjmp r7  sbis PIND,2  rjmp r11  sbis PIND,3  rjmp r15  ldi reg,$BF ;маска 10111111  out PORTD,reg  nop  nop  sbis PIND,0  rjmp r2  sbis PIND,1  rjmp r6  sbis PIND,2  rjmp r10  sbis PIND,3  rjmp r14 | ldi reg,$DF ;маска 11011111  out PORTD,reg  nop  nop  sbis PIND,0  rjmp r1  sbis PIND,1  rjmp r5  sbis PIND,2  rjmp r9  sbis PIND,3  rjmp r13  ldi reg,$EF ;маска 11101111  out PORTD,reg  nop  nop  sbis PIND,0  rjmp r0  sbis PIND,1  rjmp r4  sbis PIND,2  rjmp r8  sbis PIND,3  rjmp r12  rjmp forever |

Как видно из блок-схемы алгоритма работы программы, в случае нажатия какой-либо кнопки, программа переходит на метку той кнопки, которая оказалась нажатой. Здесь с помощью первого таймер-счетчика устанавливается порог счета, который определяет частоту данного режима, а также вызывается подпрограмма замены символа, который отображает частоту данного режима. После чего программа возвращается обратно на метку forever, где снова начинается динамический опрос клавиатуры. Ниже приведен листинг блока обработки нажатий кнопок:

|  |  |
| --- | --- |
| r0:  ldi temp, high($1E78) ;2Гц  out OCR1AH, temp  ldi temp, low($1E78)  out OCR1AL, temp  ldi lcd, $32  rcall zamena1  rjmp forever  r1:  ldi temp, high($F40) ;4Гц  out OCR1AH, temp  ldi temp, low($F40)  out OCR1AL, temp  ldi lcd, $34  rcall zamena1  rjmp forever  r2:  ldi temp, high($A60) ;6 Гц  out OCR1AH, temp  ldi temp, low($A60)  out OCR1AL, temp  ldi lcd, $36  rcall zamena1  rjmp forever  r3:  ldi temp, high($7A0) ;8Гц  out OCR1AH, temp  ldi temp, low($7A0)  out OCR1AL, temp  ldi lcd, $38  rcall zamena1  rjmp forever  r4:  ldi temp, high($618) ;10Гц  out OCR1AH, temp  ldi temp, low($618)  out OCR1AL, temp  ldi lcd,$31  rcall zamena  ldi lcd,$30  rcall zamena1  rjmp forever  r5:  ldi temp, high($510) ;12Гц  out OCR1AH, temp  ldi temp, low($510)  out OCR1AL, temp  ldi lcd,$31  rcall zamena  ldi lcd,$32  rcall zamena1  rjmp forever  r6:  ldi temp, high($45C) ;14Гц  out OCR1AH, temp  ldi temp, low($45C)  out OCR1AL, temp  ldi lcd,$31  rcall zamena  ldi lcd,$34  rcall zamena1  rjmp forever  r7:  ldi temp, high($3D0) ;16Гц  out OCR1AH, temp  ldi temp, low($3D0)  out OCR1AL, temp  ldi lcd,$31  rcall zamena  ldi lcd,$36  rcall zamena1  rjmp forever  r8:  ldi temp, high($358) ;18Гц  out OCR1AH, temp  ldi temp, low($358)  out OCR1AL, temp  ldi lcd,$31  rcall zamena | ldi lcd,$38  rcall zamena1  rjmp forever  r9:  ldi temp, high($30C) ;20Гц  out OCR1AH, temp  ldi temp, low($30C)  out OCR1AL, temp  ldi lcd,$32  rcall zamena  ldi lcd,$30  rcall zamena1  rjmp forever  r10:  ldi temp, high($2C6) ;22Гц  out OCR1AH, temp  ldi temp, low($2C6)  out OCR1AL, temp  ldi lcd,$32  rcall zamena  ldi lcd,$32  rcall zamena1  rjmp forever  r11:  ldi temp, high($28B) ;24Гц  out OCR1AH, temp  ldi temp, low($28B)  out OCR1AL, temp  ldi lcd,$32  rcall zamena  ldi lcd,$34  rcall zamena1  rjmp forever  r12:  ldi temp, high($259) ;26Гц  out OCR1AH, temp  ldi temp, low($259)  out OCR1AL, temp  ldi lcd,$32  rcall zamena  ldi lcd,$36  rcall zamena1  rjmp forever  r13:  ldi temp, high($22E) ;28Гц  out OCR1AH, temp  ldi temp, low($22E)  out OCR1AL, temp  ldi lcd,$32  rcall zamena  ldi lcd,$38  rcall zamena1  rjmp forever  r14:  ldi temp, high($208) ;30Гц  out OCR1AH, temp  ldi temp, low($208)  out OCR1AL, temp  ldi lcd,$33  rcall zamena  ldi lcd,$30  rcall zamena1  rjmp forever  r15:  ldi temp, high($1E8) ;32Гц  out OCR1AH, temp  ldi temp, low($1E8)  out OCR1AL, temp  ldi lcd,$33  rcall zamena  ldi lcd,$32  rcall zamena1  rjmp forever |

Был полностью описан алгоритм основной программы. Далее будет приведен алгоритм подпрограммы генерации и приведен ее листинг.

Подпрограмма генерации вызывается по прерыванию от первого таймер-счетчика при нажатии кнопки. Прерывания предварительно разрешены в инициализации микроконтроллера:

|  |
| --- |
| ldi temp, $40 ;настройка прерываний от Т1  out TIMSK, temp  sei ; разрешение общих прерываний |

Генерация пилообразных импульсов была организована следующим образом: создана переменная, в которую изначально записано значение 0 и командой inc происходит увеличение значения переменной на 1, пока оно не достигнет 255. Далее сброс значения константы в ноль, и возобновление счета до порога. Блок-схема алгоритма подпрограммы генерации приведена на рис. 4.3.

Листинг подпрограммы генерации

|  |
| --- |
| generation:  out PORTC, lev  inc lev  reti |

Как видно подпрограмма, соответствующая такому алгоритму генерации линейно-нарастающего напряжения, является оптимальной с точки зрения кода и выполняется за минимальное количество тактов.

Весьма важным блоком программы является работа с ЖКИ.

При установке режима по умолчанию на ЖКИ выводится следующее:

«F = 2 Hz ». Далее при нажатии на кнопки изменяются лишь 4й и 5й символы, отображающие значение частоты, что происходит при помощи подпрограмм zamena и zamena1 соответственно. Для отображения на ЖКИ одного символа необходимо в переменную lcd записать значение символа, который необходимо отобразить, а далее проверить ЖКИ на готовность к общению и произвести запись в регистр DR. На рис. 4.4 приведен алгоритм вывода надписи на ЖКИ.

Рассмотрим алгоритм, который реализован в подпрограммах готовности и записи в регистры IR и DR. Для этого на рис. 4.5 приведена а блок-схема алгоритма подпрограммы готовности, на рис.4.6а подпрограмма записи в регистр DR, на рис. 4.6б подпрограмма записи в регистр IR.

Для осуществления работы с ЖКИ в каждой конкретной подпрограмме формируются необходимые в данном случае временные диаграммы :для случая записи в регистр IR и DR.

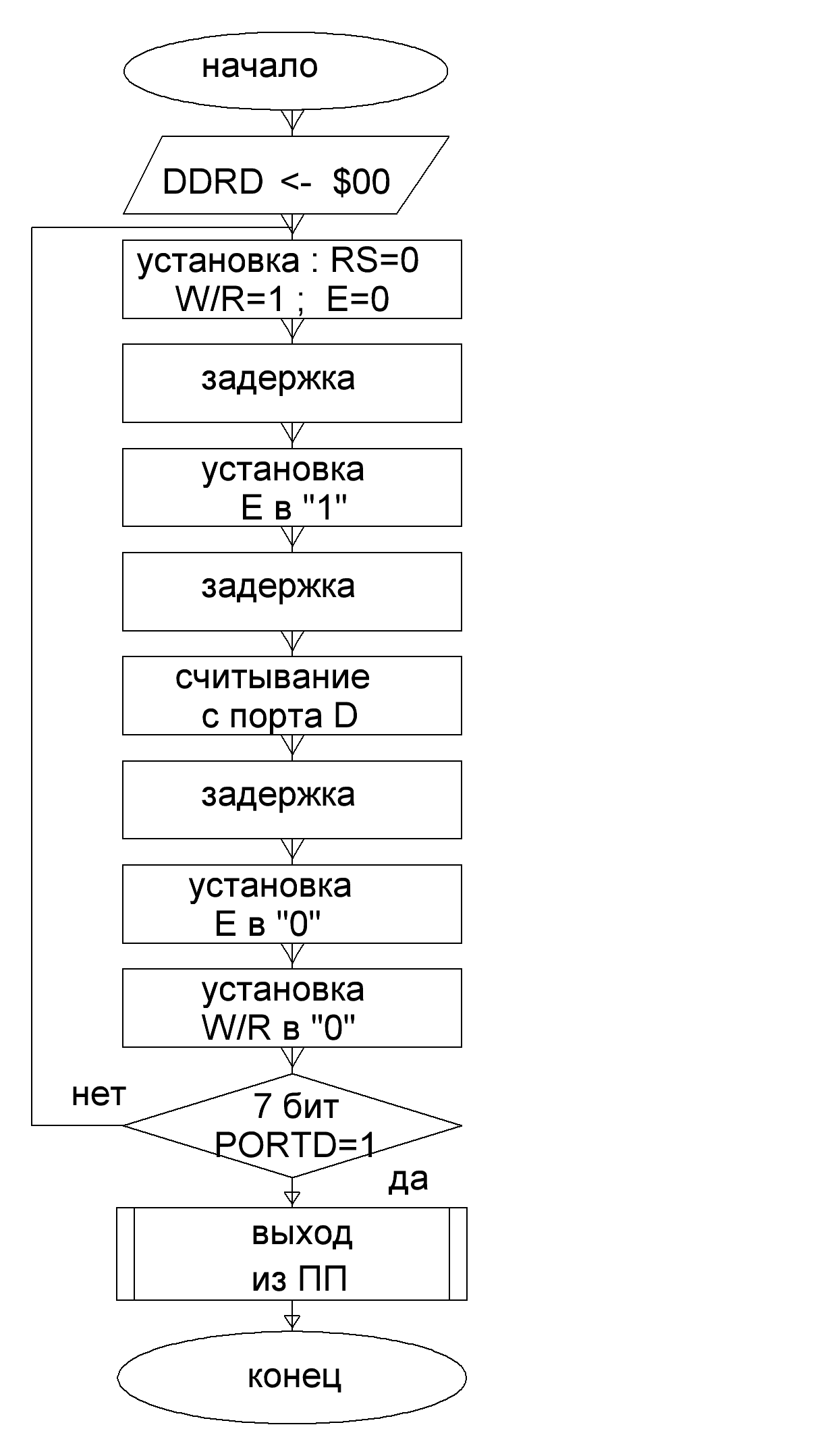


Рис. 4.5 – Блок-схема алгоритма проверки ЖКИ на готовность

Листинг подпрограммы готовности:

|  |  |
| --- | --- |
| gotov:  ldi temp, $00  out DDRB,temp  ldi temp, $FF  out PORTB,temp  out DDRA, temp  mm1: ldi temp,0b00000010  out PORTA, temp  rcall LCD\_Delay | sbi PORTA, E  rcall LCD\_Delay  in temp, PIND  rcall LCD\_Delay  cbi PORTA,E  rcall LCD\_Delay  cbi PORTA,RW  sbrs temp,7  ret  rjmp mm1 |

а). б).

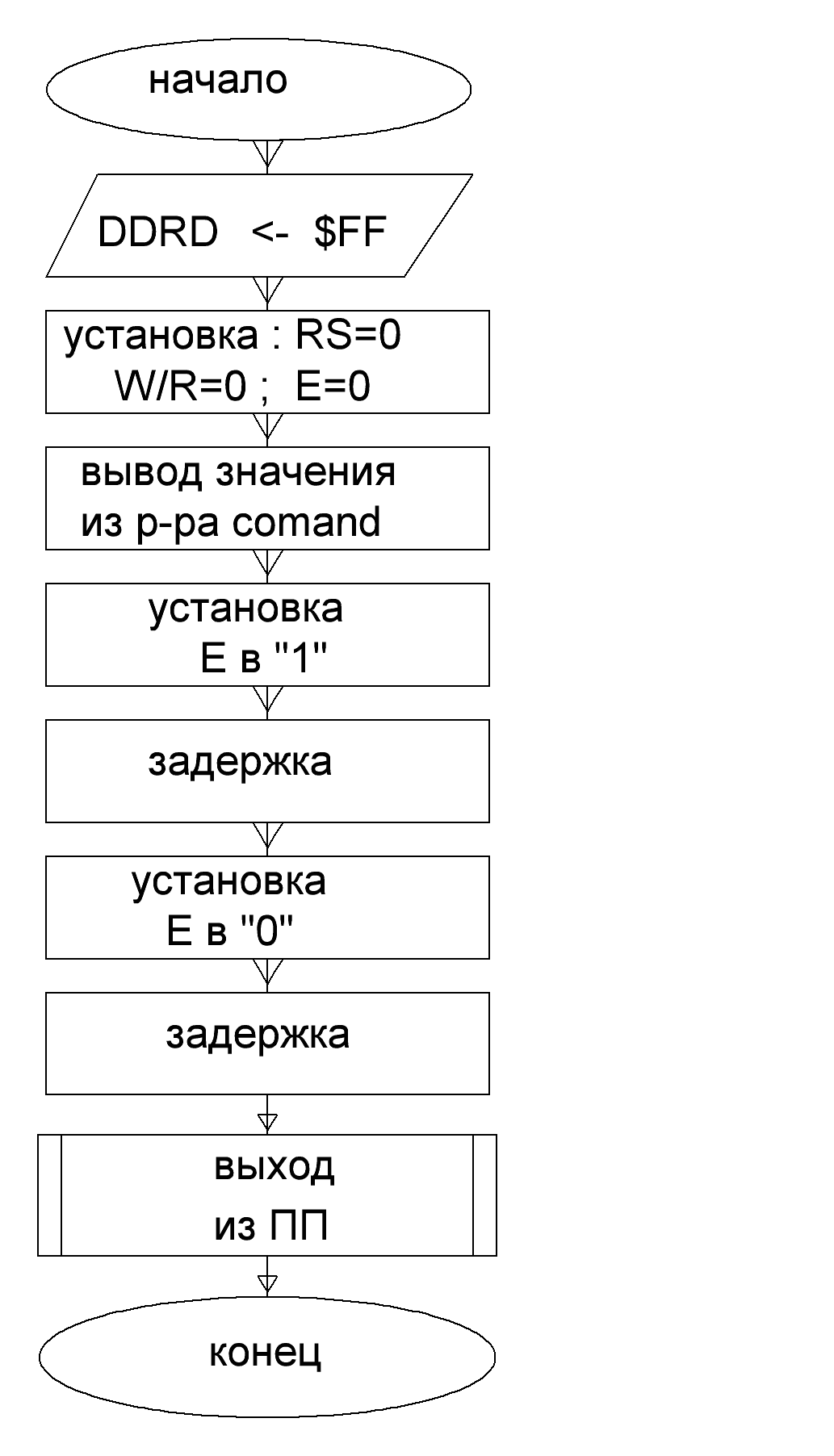
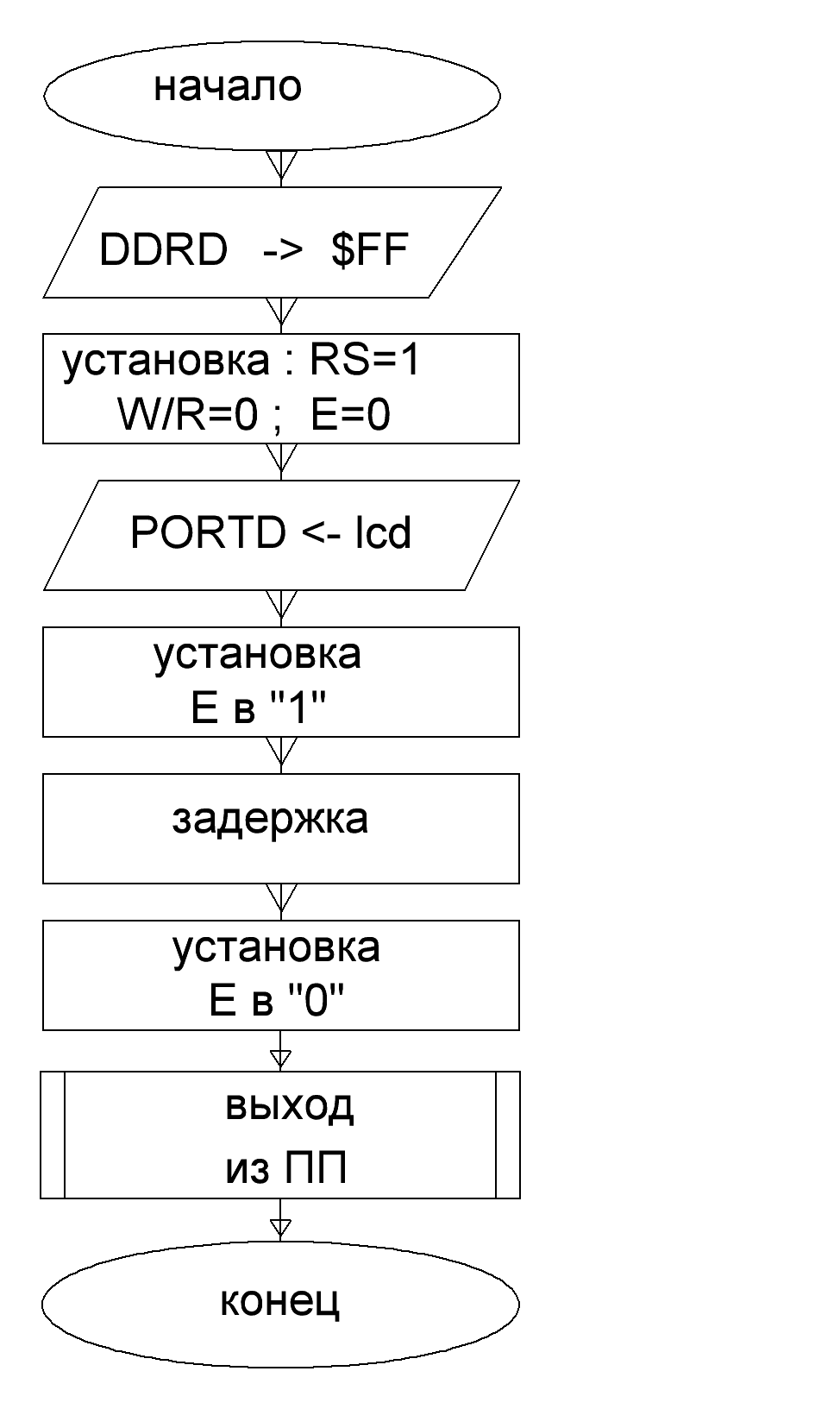


Рис. 4.6 – Блок-схема алгоритма подпрограмм записи в регистр IR(a) и записи в регистр DR(б)

Листинг подпрограммы записи в регистр IR

|  |  |
| --- | --- |
| zap\_IR:  ldi temp, $FF  out DDRD,temp  ldi temp, $00  out PORTA, temp | out PORTD, Comand  sbi PORTA,E  rcall LCD\_Delay  cbi PORTA,E  rcall LCD\_Delay  ret |

Листинг подпрограммы записи в регистр DR

|  |  |
| --- | --- |
| zap\_DR:  ldi temp, $FF  out DDRD, temp  ldi temp,0b00000001  out PORTA, temp  rcall LCD\_Delay | out PORTD, lcd  sbi PORTA,E  rcall LCD\_Delay  cbi PORTA,E  rcall LCD\_Delay  ret |

В подпрограммах замены указывается символ, который изменяется. Значение символа указывается непосредственно в блоке обработке нажатия кнопки.

В случае если изменяется один выводимый на ЖКИ знак, вызывается подпрограмма zamenа (единицы), в случае изменения десятков дополнительно вызывается подпрограмма zamena1.

Листинг подпрограммы замены единиц (zamenа) и десятков(zamena1):

|  |  |
| --- | --- |
| zamena:  ldi Comand, $84  mm2: rcall LCD\_Delay  rcall gotov  rcall LCD\_Delay  rcall zap\_IR  rcall LCD\_Delay | rcall gotov  rcall LCD\_Delay  rcall zap\_DR  ret  zamena1:  ldi Comand,$85  rjmp mm2 |

Как видно из листинга, необходимое время задержки реализуется подпрограммой задержки, использование которой является более оптимальной с точки зрения программного кода.

|  |
| --- |
| LCD\_Delay: ldi x,$04  L\_loop: dec x  brne L\_loop  ret |

Текст программы управления генератором линейно-нарастающего напряжения сверхнизкой частоты в полном объеме приведен в приложении А.

**5. Результаты эмуляции в программы в пакете VMLAB**

Ниже приведены результаты эмуляции в пакете VMLAB, которые представляют собой полученные значения генерируемых частот, сведенные в таблицу 5.1. В качестве примера приводятся окно осциллографа программы, и отображаемая при этом на ЖКИ частота для второго и шестого режимов.

Таблица 1 – Результаты эмуляции в пакете VMLAB

|  |  |  |
| --- | --- | --- |
| № кнопки | Задаваемая программно частота, Гц | Полученная в ходе эмуляции частота, Гц |
| 0 | 2.0 | 2.0 |
| 1 | 4.0 | 4.0 |
| 2 | 6.0 | 5.9 |
| 3 | 8.0 | 8.0 |
| 4 | 10.0 | 10.0 |
| 5 | 12.0 | 12.0 |
| 6 | 14.0 | 14.1 |
| 7 | 16.0 | 16.0 |
| 8 | 18.0 | 18.0 |
| 9 | 20.0 | 20.0 |
| 10 | 22.0 | 22.0 |
| 11 | 24.0 | 24.0 |
| 12 | 26.0 | 26.0 |
| 13 | 28.0 | 28.0 |
| 14 | 30.0 | 30.0 |
| 15 | 32.0 | 32.0 |

Отображаемое на ЖКИ значение частоты соответствует генерируемой в данный момент частоте для всех 16ти режимов. Соответствие кнопок частоте и полученная при анализе частота генерации показаны в таблице 5.1.

**6. Анализ временных соотношений и оценка погрешностей**

Расчет константы N, которая записывается в регистр сравнения первого таймер-счетчика выполняется по следующим соотношениям:

,

Где 0.25 мкс – время тактирования генератора при тактовой частоте f=4МГц;



Согласно техническому заданию расчет константы необходимо провести для шестнадцати режимов. Полученную константу N переведем в шестнадцатеричную систему. Соответственно ниже приведена таблица расчетов для 16ти частот:

Табл. 2 – Расчет коэффициента N в зависимости от частоты

|  |  |  |  |
| --- | --- | --- | --- |
| F, МГц | , мкс |  |  |
| 2 | 1960 | 7800 | 1E78 |
| 4 | 976 | 3904 | F40 |
| 6 | 664 | 2656 | A60 |
| 8 | 488 | 1952 | 7A0 |
| 10 | 390 | 1500 | 618 |
| 12 | 324 | 1296 | 510 |
| 14 | 279 | 1116 | 45C |
| 16 | 244 | 976 | 3D0 |
| 18 | 214 | 906 | 358 |
| 20 | 195 | 780 | 30C |
| 22 | 175 | 700 | 2BC |
| 24 | 162 | 648 | 288 |
| 26 | 148 | 592 | 250 |
| 28 | 139 | 556 | 22C |
| 30 | 128 | 512 | 200 |
| 32 | 122 | 488 | 1E8 |

Тот факт, что деление приводит к наличию более чем 4х знаков за запятой, и значения необходимо округлять до целых, говорит о том, что при генерации возможно возникновение погрешностей. Однако согласно эмуляции в пакете VMLAB погрешности генерации частоты не превышают 1%.

**Выводы**

В ходе выполнения данного курсового проекта был разработан алгоритм, спроектирована принципиальная схема и написана программа управления генератора линейно-нарастающего напряжения с дискретно-изменяющейся частотой с помощью матрицы кнопок и плавно-изменяющейся амплитудой.

Отображение генерируемой частоты осуществлялось с помощью ЖКИ.

После выполнения поставленной задачи были сделаны такие существенные выводы:

- проектирование и техническое выполнение генераторов линейно-нарастающего напряжения на базе микроконтроллеров является более оптимальным со схемотехнической точки зрения, с точки зрения управления и компактности устройства, а также точности получаемых значений;

- погрешности получаемые при генерации весьма малы, что свидетельствует о точности расчета;

- спроектированное устройство целиком и полностью соответствует техническому заданию.

Данное устройство может получить применение в сфере телевидения для горизонтальной развёртки электронного луча в электронно-лучевой трубке, а также широко применяться в импульсной технике.

**Приложение А**

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; BASIC .ASM template file for AVR

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.include "D:\0B54~1\9E9E~1\VMLAB\include\m8515def.inc"

; Define here the variables

; программа генератор пилообразный напряжение

.def temp = r16

.def adr = r17

.def reg = r18

.def const0 = r19

.def const01 = r20

.def time = r21

.def lev =r22

.def Comand =r23

.def lcd =r24

.def x =r25

.equ E = 2

.equ RW = 1

.equ RS = 0

; Define here Reset and interrupt vectors, if any

reset:

rjmp start

reti ; Addr $01

reti ; Addr $02

reti ; Addr $03

rjmp generation ; Addr $04

reti ; Addr $05

reti ; Addr $06 Use 'rjmp myVector'

reti ; Addr $07 to define a interrupt vector

reti ; Addr $08

reti ; Addr $09

reti ; Addr $0A

reti ; Addr $0B This is just an example

reti ; Addr $0C Not all MCUs have the same

reti ; Addr $0D number of interrupt vectors

reti ; Addr $0E

reti ; Addr $0F

reti ; Addr $10

; Program starts here after Reset

start: ldi temp,$02

out SPH,temp

ldi temp,$5F

out SPL,temp

;================= настройка портов ===================

ldi temp,$F0

out DDRD,temp

ldi temp,$FF

out PORTD,temp

out DDRC,temp

out PORTC,temp

;================ настройка Т1 и прерываний ==========

ldi temp, $40

out TIMSK, temp

ldi temp, $09

out TCCR1B, temp

sei

;================ загрузка констант ===================

ldi const0,0

ldi const01,$01

ldi lev,$00

ldi x, $00

;================ режим по умолчанию ==================

ldi temp, high($1E78)

out OCR1AH, temp

ldi temp, low($1E78)

out OCR1AL, temp

;================ инициализация ЖКИ ===================

ldi Comand,$0C ;1100 определение режима отображения

rcall gotov

rcall LCD\_Delay

rcall zap\_IR

ldi Comand,$06 ;110 выбор направления сдвига

rcall LCD\_Delay

rcall gotov

rcall LCD\_Delay

rcall zap\_IR

ldi Comand,$38 ;развертка в одну строку

rcall LCD\_Delay

rcall gotov

rcall LCD\_Delay

rcall zap\_IR

;================Вывод надписи============================

ldi lcd,$46 ;F

rcall mm3

ldi lcd,$20

rcall mm3

ldi lcd,$3D ;=

rcall mm3

ldi lcd,$20

rcall mm3

ldi lcd,$20

rcall mm3

ldi lcd,$32 ;2

rcall mm3

ldi lcd,$20

rcall mm3

ldi lcd,$48 ;H

rcall mm3

ldi lcd,$7A ;z

rcall mm3

;================последовательный опрос клавиатуры =============

forever:

ldi reg,$7F

out PORTD,reg

nop

nop

sbis PIND,0

rjmp r3

sbis PIND,1

rjmp r7

sbis PIND,2

rjmp r11

sbis PIND,3

rjmp r15

ldi reg,$BF

out PORTD,reg

nop

nop

sbis PIND,0

rjmp r2

sbis PIND,1

rjmp r6

sbis PIND,2

rjmp r10

sbis PIND,3

rjmp r14

ldi reg,$DF

out PORTD,reg

nop

nop

sbis PIND,0

rjmp r1

sbis PIND,1

rjmp r5

sbis PIND,2

rjmp r9

sbis PIND,3

rjmp r13

ldi reg,$EF

out PORTD,reg

nop

nop

sbis PIND,0

rjmp r0

sbis PIND,1

rjmp r4

sbis PIND,2

rjmp r8

sbis PIND,3

rjmp r12

rjmp forever

;=================== подпрограмма генерации ======================

generation:

out PORTC, lev

inc lev

reti

;================== метки перехода на кнопки =====================

r0:

ldi temp, high($1E78) ;2Гц

out OCR1AH, temp

ldi temp, low($1E78)

out OCR1AL, temp

ldi lcd, $32

rcall zamena1

rjmp forever

r1:

ldi temp, high($F40) ;4Гц

out OCR1AH, temp

ldi temp, low($F40)

out OCR1AL, temp

ldi lcd, $34

rcall zamena1

rjmp forever

r2:

ldi temp, high($A60) ;6 Гц

out OCR1AH, temp

ldi temp, low($A60)

out OCR1AL, temp

ldi lcd, $36

rcall zamena1

rjmp forever

r3:

ldi temp, high($7A0) ;8Гц

out OCR1AH, temp

ldi temp, low($7A0)

out OCR1AL, temp

ldi lcd, $38

rcall zamena1

rjmp forever

r4:

ldi temp, high($618) ;10Гц

out OCR1AH, temp

ldi temp, low($618)

out OCR1AL, temp

ldi lcd,$31

rcall zamena

ldi lcd,$30

rcall zamena1

rjmp forever

r5:

ldi temp, high($510) ;12Гц

out OCR1AH, temp

ldi temp, low($510)

out OCR1AL, temp

ldi lcd,$31

rcall zamena

ldi lcd,$32

rcall zamena1

rjmp forever

r6:

ldi temp, high($45C) ;14Гц

out OCR1AH, temp

ldi temp, low($45C)

out OCR1AL, temp

ldi lcd,$31

rcall zamena

ldi lcd,$34

rcall zamena1

rjmp forever

r7:

ldi temp, high($3D0) ;16Гц

out OCR1AH, temp

ldi temp, low($3D0)

out OCR1AL, temp

ldi lcd,$31

rcall zamena

ldi lcd,$36

rcall zamena1

rjmp forever

r8:

ldi temp, high($358) ;18Гц

out OCR1AH, temp

ldi temp, low($358)

out OCR1AL, temp

ldi lcd,$31

rcall zamena

ldi lcd,$38

rcall zamena1

rjmp forever

r9:

ldi temp, high($30C) ;20Гц

out OCR1AH, temp

ldi temp, low($30C)

out OCR1AL, temp

ldi lcd,$32

rcall zamena

ldi lcd,$30

rcall zamena1

rjmp forever

r10:

ldi temp, high($2C6) ;22Гц

out OCR1AH, temp

ldi temp, low($2C6)

out OCR1AL, temp

ldi lcd,$32

rcall zamena

ldi lcd,$32

rcall zamena1

rjmp forever

r11:

ldi temp, high($28B) ;24Гц

out OCR1AH, temp

ldi temp, low($28B)

out OCR1AL, temp

ldi lcd,$32

rcall zamena

ldi lcd,$34

rcall zamena1

rjmp forever

r12:

ldi temp, high($259) ;26Гц

out OCR1AH, temp

ldi temp, low($259)

out OCR1AL, temp

ldi lcd,$32

rcall zamena

ldi lcd,$36

rcall zamena1

rjmp forever

r13:

ldi temp, high($22E) ;28Гц

out OCR1AH, temp

ldi temp, low($22E)

out OCR1AL, temp

ldi lcd,$32

rcall zamena

ldi lcd,$38

rcall zamena1

rjmp forever

r14:

ldi temp, high($208) ;30Гц

out OCR1AH, temp

ldi temp, low($208)

out OCR1AL, temp

ldi lcd,$33

rcall zamena

ldi lcd,$30

rcall zamena1

rjmp forever

r15:

ldi temp, high($1E8) ;32Гц

out OCR1AH, temp

ldi temp, low($1E8)

out OCR1AL, temp

ldi lcd,$33

rcall zamena

ldi lcd,$32

rcall zamena1

rjmp forever

;================== работа с ЖКИ ==============

LCD\_Delay: ldi x,$04

L\_loop: dec x

brne L\_loop

ret

gotov:

ldi temp, $00

out DDRB,temp

ldi temp, $FF

out PORTB,temp

out DDRA, temp

mm1: ldi temp,0b00000010

out PORTA, temp

rcall LCD\_Delay

sbi PORTA, E

rcall LCD\_Delay

in temp, PIND

rcall LCD\_Delay

cbi PORTA,E

rcall LCD\_Delay

cbi PORTA,RW

sbrs temp,7

ret

rjmp mm1

zap\_IR:

ldi temp, $FF

out DDRD,temp

ldi temp, $00

out PORTA, temp

out PORTD, Comand

sbi PORTA,E

rcall LCD\_Delay

cbi PORTA,E

rcall LCD\_Delay

ret

zap\_DR:

ldi temp, $FF

out DDRD, temp

ldi temp,0b00000001

out PORTA, temp

rcall LCD\_Delay

out PORTD, lcd

sbi PORTA,E

rcall LCD\_Delay

cbi PORTA,E

rcall LCD\_Delay

ret

zamena:

ldi Comand, $84

mm2: rcall LCD\_Delay

rcall gotov

rcall LCD\_Delay

rcall zap\_IR

rcall LCD\_Delay

rcall gotov

rcall LCD\_Delay

rcall zap\_DR

ret

zamena1:

ldi Comand,$85

rjmp mm2

mm3:

rcall LCD\_Delay

rcall gotov

rcall LCD\_Delay

rcall zap\_DR

ret