**Содержание**

Введение 3

Теоретическая часть 4

Описание схемы 6

Описание программы 13

Заключение 34

Библиографический список 35

Приложения 36

**Введение**

Курсовой проект предназначен для приобретения практических навыков проектирования несложных микропроцессорных систем различного назначения. Проект базируется на теоретической части дисциплины «Организация ЭВМ и систем». Задание на курсовой проект выдается руководителем проекта.

Курсовой проект выполняется с целью закрепления знаний по курсу «Организация ЭВМ и систем» и развития навыков самостоятельного проектирования микропроцессорных систем различного назначения.

Задачами курсового проекта являются:

* практическое овладение методикой проектирования устройств;
* синтез функциональной схемы микропроцессорной системы на основе анализа исходных данных;
* получение навыков разработки аппаратного и программного обеспечения микропроцессорной системы;
* дальнейшее развитие навыков функционально-логического, схемотехнического и конструкторского проектирования, оформления и выпуска конструкторской документации в соответствии с ГОСТ.

Для решения перечисленных задач необходимы знания не только курса «Организация ЭВМ и систем», но и ряда смежных дисциплин, а также умение пользоваться нормативно-справочной информацией.

Одним из основных направлений научно-технического прогресса в настоящее время является развитие и широкое применение изделий микроэлектроники в промышленном производстве, в устройствах и системах управления самыми разнообразными объектами и процессами.

Одним из примеров являются микроконтроллеры, производимые фирмой Microchip Technology. Это семейство 8-разрядных микроконтроллеров отличается низкой ценой, низким энеpгопотpеблением и высокой скоpостью. Микроконтроллеры имеют встpоенное ЭППЗУ пpогpаммы, ОЗУ данных и выпускаются в 18 и 28 выводных коpпусах. Для изделий, пpогpамма котоpых может меняться, либо содеpжит какие-либо пеpеменные части, таблицы, паpаметpы калибpовки, ключи и т.д., выпускается электрически стираемый и пеpепpогpаммиpуемый микроконтpоллеp PIC16F84. Он также содержит электрически пеpепpогpаммиpуемое ПЗУ данных. Именно такой контpоллеp и будем использовать для разработки устройства ультразвукового измерения дальности.

**Теоретическая часть**

Работа устройства ультразвукового измерения дальности основывается на явлении распространения звуковых волн в воздушной среде и отражения их в процессе распространения от других сред (контролируемых тел).

Информация о расстоянии до контролируемого тела, точнее некоторой отражающей зоны, принадлежащей поверхности контролируемого тела, определяется временным запаздыванием принимаемого сигнала относительно излучаемого. Примерно таким же образом летучие мыши ориентируются в пространстве: они излучают вперед направленный пучок ультразвуковых колебаний и ловят отраженный сигнал. Звуковые волны распространяются в воздушной среде с определенной скоростью, поэтому по задержке прихода отраженного сигнала можно с достаточной степенью точности судить, на каком расстоянии находится тот предмет, который отразил звук.

Ультразвуковой дальномер производит измерение расстояния до контролируемого тела по схеме эхо-локации (см. рис 1).

Рис. 1. Схема эхо-локации.

Для измерения расстояний в воздушной среде используются пьезокерамические преобразователи (типа МУП-3 и МУП-4, произведенные “ЭЛПА” г. Зеленоград), работающие на 40 кГц частоте. Два пьезокерамических преобразователя (излучающий и приемный), подобранные так, чтобы резонансная частота излучения излучающего, совпадала с резонансной частотой приема приемного, образуют акустический блок.

Преимуществами использования таких преобразователей в воздушной среде являются: сравнительная простота излучения и приема колебаний, компактность приемоизлучающих элементов аппаратуры, высокая устойчивость к шумовому, химическому и оптическому загрязнению окружающей среды, возможность работы в агрессивных средах при высоких давлениях, возможность значительного удаления вторичной аппаратуры от места измерений, длительный срок службы, простота в использовании, сравнительно малая стоимость, практически мгновенная готовность к работе после включения, нечувствительность к электромагнитным помехам, высокая надежность, невосприимчивость органов слуха человека к ультразвуку используемой частоты (40КГц) и ряд других.

Примерами применения разрабатываемого ультразвукового дальномера могут служить: контроль дистанции между автотранспортом при его движении в условиях недостаточной видимости на небольших скоростях, измерение уровня заполнения резервуаров жидким веществом, уровня загрузки бункеров или кузовов автомобилей сыпучим или дробленым материалом, контроль размеров продукции, измерение дистанции от борта судна до причальной стенки и др.

**Описание принципиальной схемы**

Принципиальная электрическая схема проектируемого устройства представлена в приложении. Представленную схему можно разбить на 5 функциональных блоков:

1) блок питания;

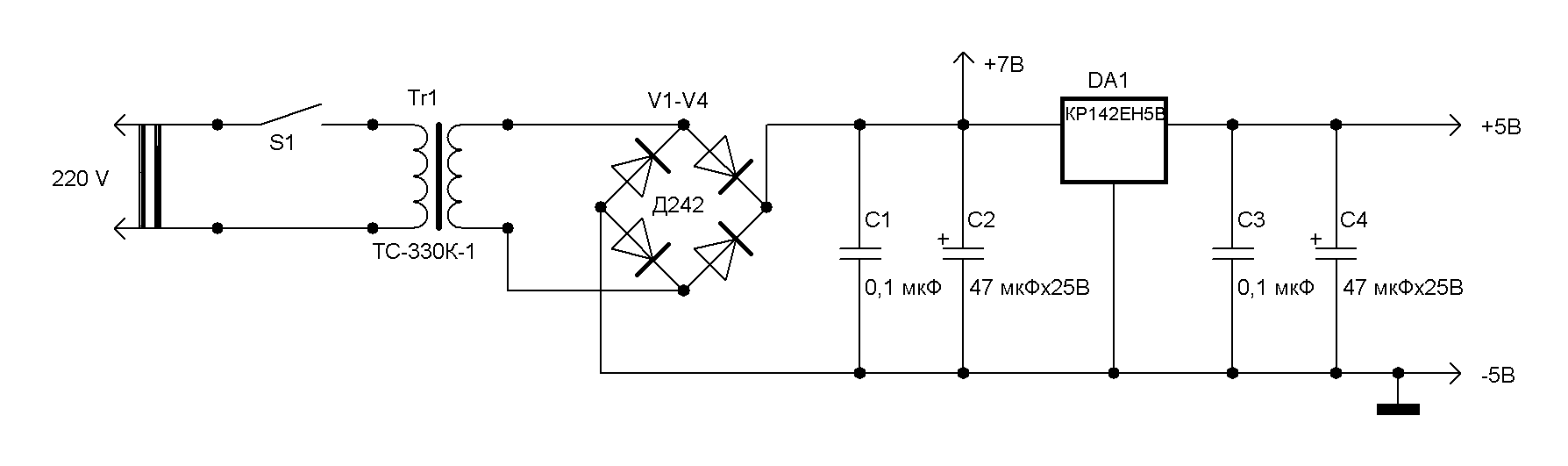
2) блок передатчика;

3) блок приемника;

4) блок индикации;

5) блок цифрового управления.

Рассмотрим порядок работы каждого из них.



TV1

Рис. 2. Блок питания.

Блок питания представлен на рис. 2. При включении сетевого выключателя S1 на первичную обмотку трансформатора TV1 поступает переменное напряжение величиной в 220В. Со вторичной обмотки трансформатора снимается пониженное до 7,5В переменное напряжение. После прохождения через диодный мост V1-V4 мы получаем выпрямленное, несглаженное напряжение величиной около 7В, т.к. существует некоторое небольшое падение напряжения на диодах. Пульсации полученного выпрямленного напряжения сглаживает электролитический конденсатор С2, а керамический конденсатор С1 предназначен для фильтрации высокочастотных сетевых помех. Затем напряжение стабилизируется при помощи интегрального стабилизатора напряжения DA1 и фильтруются высоко и низкочастотные помехи с помощью конденсаторов С3 и С4 соответственно. Диодный мост V1-V4 собран на кремниевых низкочастотных диодах допускающих напряжение до 100В при токе не более 10А. Интегральный стабилизатор напряжения DA1 (КР142ЕН5В) имеет следующие характеристики: Uвых=5В – выходное напряжение;

Iмакс=1,5А – максимальный ток нагрузки;

Pмакс=10Вт – максимальная мощность;

включение – плюсовое – тип подключения.

Данная схема блока питания является типовой.

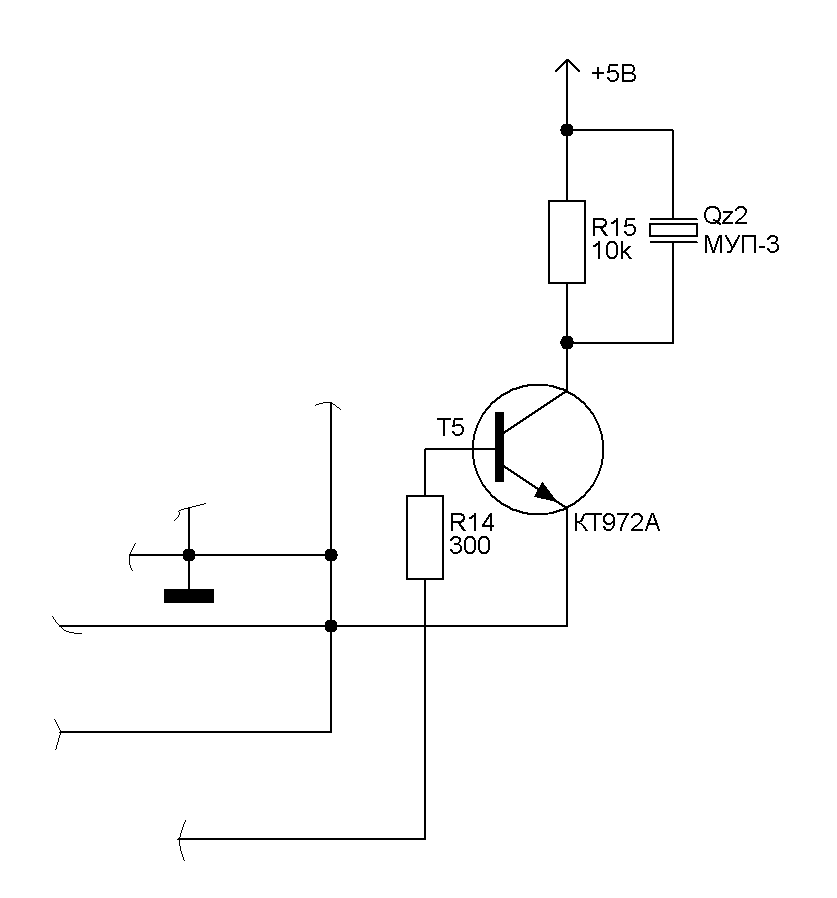


Рис. 3. Блок передатчика.

Блок передатчика представлен на рис. 3. Представленный блок выполнен по схеме усилителя с общим эмиттером, работающего в ключевом режиме. Ток на выводах микроконтроллера DD2 не должен превышать максимально разрешенный ток величиной в 20мА, при напряжении в 5В. Тогда по закону Ома: R=U/I или R=5/0,020=250Ом. Для R14 был выбран резистор номиналом в 300Ом для обеспечения более стабильной работы схемы. Резистор R15 служит для обеспечения работы транзистора Т5 по постоянному току, т.е. обеспечивает заряд паразитной емкости транзистора. Его номинал взят из типовой схемы подключения. В качестве излучателя Qz2 использован пьезокерамический ультразвуковой преобразователь МУП-3, (т.к. он обладают достаточно высокой эффективностью, по заверениям производителя) основные характеристики которого представлены в таблице 1.

Таблица 1. Характеристики ПКУП МУП-3

|  |  |
| --- | --- |
| Наименование параметра, единица измерения | Значение |
| Частота максимальной передачи, кГц | 37…45 |
| Звуковое давление на расстоянии 0,3 м при  Uвх=5В на частоте максимального излучения, дБ | 100 |
| Чувствительность на частоте максимального приема, мВ/Па | 20 |
| Ширина полосы излучения по уровню 0,5, кГц | 1,8 |
| Ширина полосы приема по уровню 0,5, кГц | 1,0 |
| Диаграмма направленности, Град |  |
| По уровню 0,7 макс. | 44 |
| По уровню 0,5 макс. | 74 |
| Емкость на частоте 1 кГц, пФ | 1300 |
| Входной импеданс на частоте максимального излучения, кОм | 0,5 |
| Предельное допустимое значение напряжения сигнала на входе, В | 12 |

Биполярные транзисторы типа n-p-n КТ972 используемые в схеме имеют следующие параметры:

- Uкбои=60В - максимально допустимое импульсное напряжение коллектор-база;

- Uкэои=60В - максимально допустимое импульсное напряжение коллектор-эмиттер;

- Iкmaxи=4000мА - максимально допустимый импульсный ток коллектора;

- Pкmaxт=8Вт - максимально допустимая постоянная рассеиваемая мощность коллектора с теплоотводом;

- h21э≥750 - статический коэффициент передачи тока биполярного транзистора в схеме с общим эмиттером;

- Iкбо≤1000мкА - обратный ток коллектора;

- fгр≥200МГц - граничная частота коэффициента передачи тока в схеме с общим эмиттером;

- Uкэн<1,5В - напряжение насыщения коллектор-эмиттер.

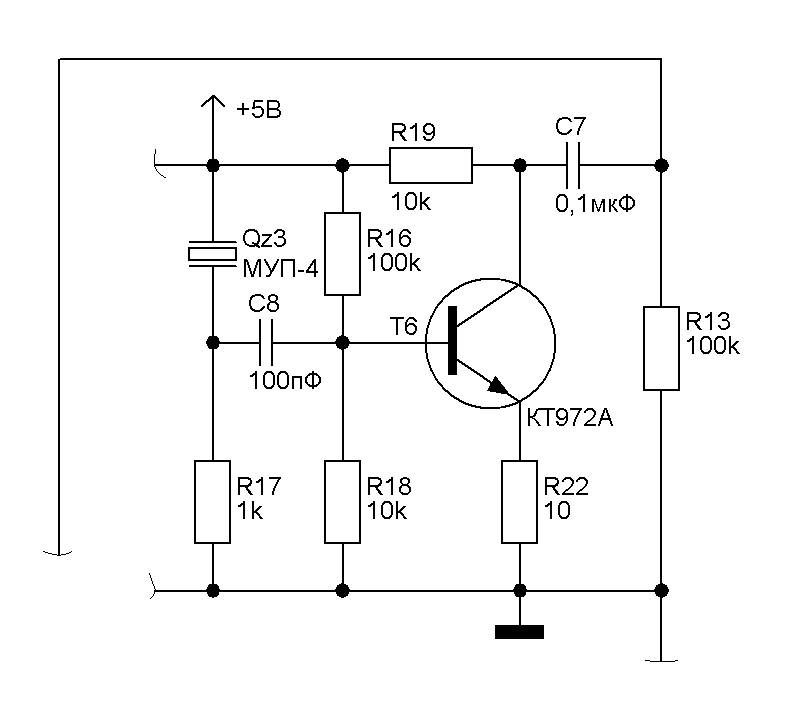


Рис. 4. Блок приемника.

Блок приемника изображен на рис. 4. Блок приемника выполнен по схеме усилителя с общим эмиттером. Максимальный коэффициент усиления для усилителя выполненного по схеме с общим эмиттером рассчитывается из соотношения резисторов R19 и R22. Т.е. 10000/10=1000. Резисторы R16 и R18 служат для стабилизации рабочей точки транзистора. Соотношение их номиналов определяет положение рабочей точки транзистора Т6. Резистор R13 подтягивает выход приемника к земле, когда нет сигнала с усилителя. Резистор R17 служит для установки режима чувствительности ультразвукового датчика Qz3. Конденсаторы С7 и С8 фильтруют постоянную составляющую. В качестве излучателя Qz3 использован пьезокерамический ультразвуковой преобразователь МУП-4, (т.к. он обладают достаточно высокой чувствительностью, по заверениям производителя) основные характеристики которого представлены в таблице 2.

Таблица 2. Характеристики ПКУП МУП-4

|  |  |
| --- | --- |
| Наименование параметра, единица измерения | Значение |
| Частота максимальной передачи, кГц | 37…45 |
| Звуковое давление на расстоянии 0,3 м при  Uвх=5В на частоте максимального излучения, дБ | 96 |
| Чувствительность на частоте максимального приема, мВ/Па | 30 |
| Ширина полосы излучения по уровню 0,5, кГц | 0,5 |
| Ширина полосы приема по уровню 0,5, кГц | 0,5 |
| Диаграмма направленности, Град |  |
| По уровню 0,7 макс. | 45 |
| По уровню 0,5 макс. | 70 |
| Емкость на частоте 1 кГц, пФ | 2500 |
| Входной импеданс на частоте максимального излучения, кОм | 0,2 |
| Предельное допустимое значение напряжения сигнала на входе, В | 12 |

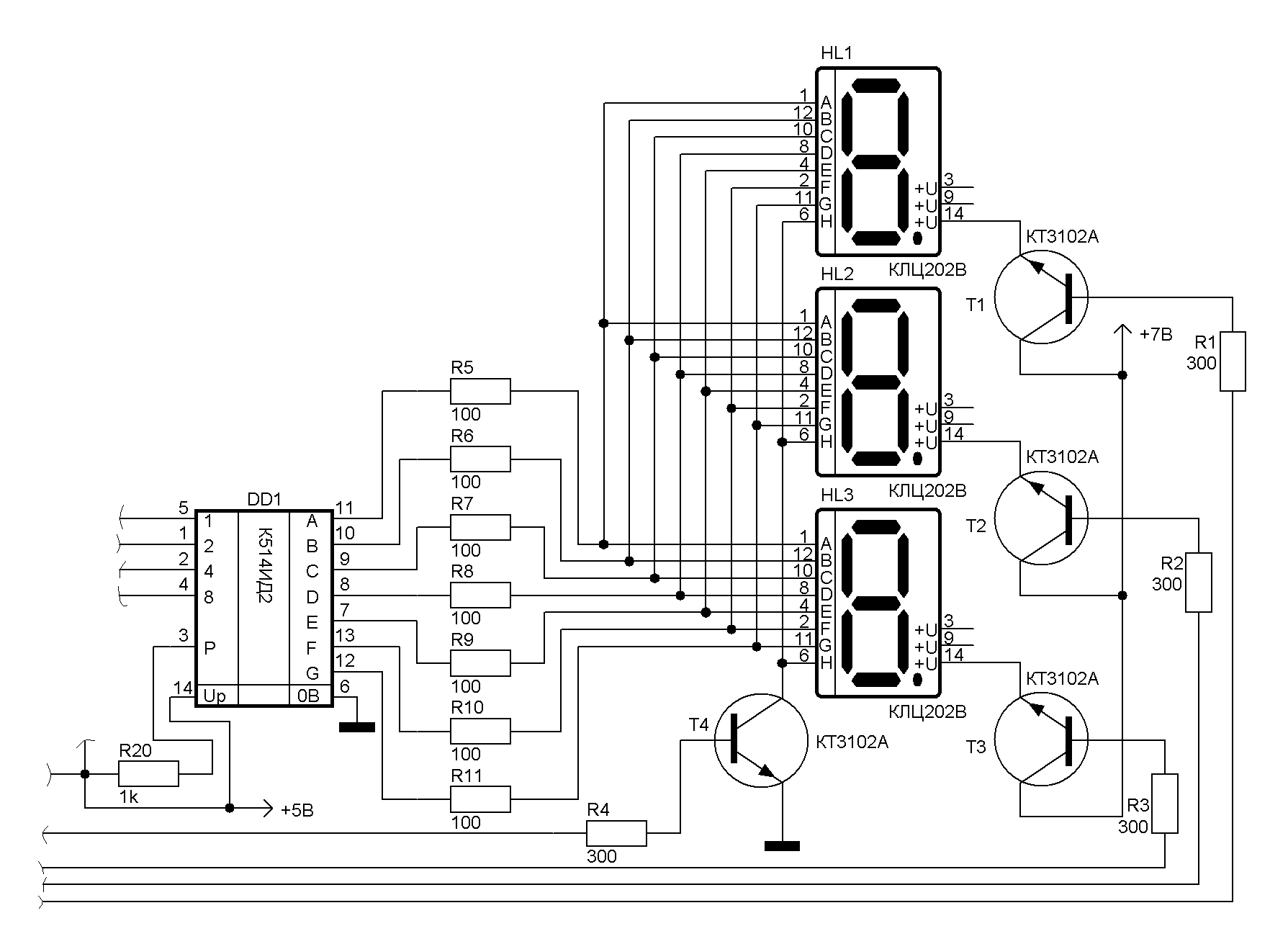


Рис. 5. Блок индикации.

Блок индикации представлен на рис. 5. DD1 – дешифратор семи сегментного индикатора. Он преобразует входной двоичный код в символы семи сегментного индикатора. Резистор R20 подтягивает вход 3 разрешения работы дешифратора к напряжению +5В. Резисторы R5-R11 служат для ограничения тока через сегменты индикаторов. Резисторы R1-R4 служат для ограничения тока через базы транзисторов Т1-Т4 и обеспечивают защиту выходов микроконтроллера. Транзисторы Т1-Т3 предназначены для коммутации питания подаваемого на индикаторы. Транзистор Т4 коммутирует включение точки на индикаторах. Светодиодные индикаторы служат для отображения информации. На входе питания индикатор имеет напряжение около 4,5В, получаемое за счет того, что транзисторы Т1-Т3 подключены по схеме эмиттерного повторителя. На переходе база-эмиттер имеет место небольшое, около 0,5В, падение напряжения, т.е. 5-0,5=4,5В (напряжение на входе питания индикатора). Ток через выводы микроконтроллера на должен превышать максимальный разрешенный ток величиной в 20мА, при напряжении в 5В. Тогда по закону Ома имеем: R=U/I или R=5/0,020=250Ом. Для R1-4 был выбран номинал в 300Ом для обеспечения более стабильной работы схемы. Согласно документации на индикаторы они должны иметь входное напряжение не более 2,5В при токе 20мА. Т.к. мы имеем на входе питания индикатора напряжение в 4,5В – следует его уменьшить. Лишние 2В гасятся с помощью резисторов R5-R11. Их номинал: 2В/0,020А=100Ом. В качестве HL1…HL3 выбраны индикаторы цифровые красного цвета свечения КЛЦ202В (арсенид-галлия-алюминия) в пластмассовом корпусе, состоящие из дискретных элементов, изготовленные по эпитаксиально-диффузионной технологий. Индикатор имеет 7 сегментов и децимальную точку, излучающие при подаче прямого тока. Различные комбинации элементов, обеспечиваемые внешней коммутацией, позволяют воспроизвести любую цифру от 0 до 9 и децимальную точку. Высота знака 20 мм. Параметры индикаторов и их принципиальная схема представлены ниже. Биполярные транзисторы типа n-p-n КТ3102 используемые в схеме имеют следующие параметры:

- Uкбои=50В - максимально допустимое импульсное напряжение коллектор-база;

- Uкэои=50В - максимально допустимое импульсное напряжение коллектор-эмиттер;

- Iкmaxи=200мА - максимально допустимый импульсный ток коллектора;

- Pкmaxт=0,25Вт - максимально допустимая постоянная рассеиваемая мощность коллектора с теплоотводом;

- h21э≥200 - статический коэффициент передачи тока биполярного транзистора в схеме с общим эмиттером;

- Iкбо≤0,05мкА - обратный ток коллектора;

- fгр≥150МГц - граничная частота коэффициента передачи тока в схеме с общим эмиттером;

Таблица 3. Основные электрические параметры КЛЦ202В

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование параметров, режим измерения,единица измерения | Буквенное  обозначение | Не менее | Не более |
| Средняя сила света элемента отображения при Iпр=20 мА , мкд | Iυ | 0,5 | . |
| Средняя сила света точки при Iпр=20 мА , мкд | Iυ | 0,07 | . |
| Постоянное прямое напряжение для элемента при Iпр=20 мА , В | Uпр | . | 2,5 |
| Постоянное прямое напряжение для точки при Iпр=20 мА , В | Uпр | . | 2,5 |
|  |  |  |  |

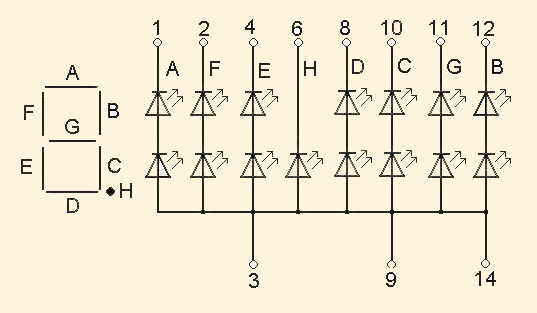


Рис. 6. Схема электрическая принципиальная КЛЦ202В.

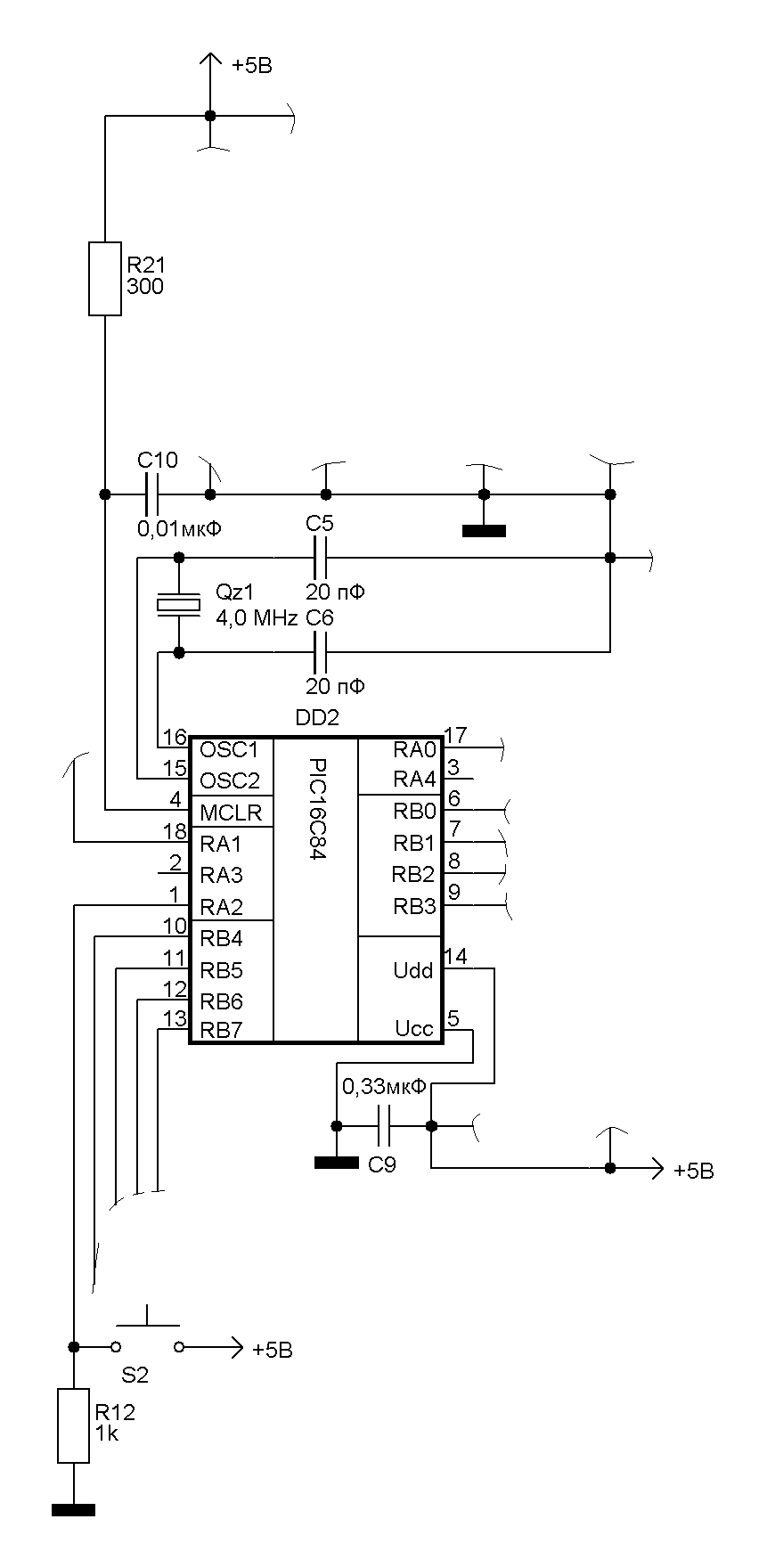


Рис.7. Блок цифрового управления.

Блок цифрового управления представлен на рис. 7. Блок цифрового управления представляет собой однокристальный микроконтроллер PIC16C84 имеющий подключения согласно технической документации производителя. Между входами питания микроконтроллера 14 и 5 ставится керамический конденсатор С9 обеспечивающий сглаживание пульсаций напряжения вызванных работой схемы. Ко входам OSC1 и OSC2 микроконтроллера подключен керамический резонатор QZ1 после которого установлены конденсаторы С5 и С6 необходимые для обеспечения правильного функционирования, согласно требованию производителя. Вход 1 микроконтроллера подтянут к земле через резистор R12 номиналом 1кОм, что является типовой защитой от помех в том случае, когда кнопка S2 разомкнута. Цепочка из резистора R21 и конденсатора С10 является типичной схемой подключения входа MCLR микроконтроллера (сброс памяти при включении питания) и обеспечивают необходимую задержку по времени для стабильного запуска микроконтроллера. Керамический резонатор QZ1 на 4.0 МГц имеет следующие параметры:

- резонансная частота,кГц: 4000;

- точность настройки (при 25 С) ,%: 0.3;

- температурный коэффициент (в диапазоне -20…+80С) ,%: 0.3;

- максимальное резонансное сопротивление ,Ом: 30;

- встроеный конденсатор,пФ: 30;

- рабочая температура,С: -20…+80.

**Описание программы**

Для удобства программа, приведенная в приложении, разбита на отдельные подпрограммы, которые взаимодействуют друг с другом согласно приведенной ниже обобщенной блок схеме программы.

Обобщенная блок-схема

начало

список директив

файл

слово CPU

описание переменных

исполняемая программа

основная программа

подпрограмма инициализации

подпрограмма индикации

подпрограмма вычисления

подпрограмма нажатия кнопки

подпрограмма передачи

подпрограмма ожидания

подпрограмма приема

Рис. 8. Обобщенная блок-схема.

Более подробно рассмотрим каждый блок приведенной на рис. 8. обобщенной блок схемы.

Список директив, файл, слово CPU

Данный блок соответствует приведенной ниже части программы.

;==============================================================

; программа для ультразвукового дальномера

;==============================================================

;список директив

list P=16C84 ;директива определяющая тип процессора

#include <P16C84.INC> ;файл, описывающий специфические переменные, соответствующие процессору

\_config b'00000000000001' ;слово конфигурации CPU

;==============================================================

С помощью директивы list указываем ассемблеру тип процессора. Подключаем соответствующий файл описания. Затем следует слово конфигурации CPU, которое при программировании микроконтроллера будет размещено по адресу расположенному за пределами пользовательской памяти программ. Слово конфигурации CPU содержит 14 бит. Биты 13-8 CP – защита программного кода (1=защита отключена, 0=защита установлена). Выбираем 0. Бит 7 DP – защита памяти данных EEPROM (1=защита отключена, 0=защита установлена). Выбираем 0. Биты 6-4 CP – защита программного кода (1=защита отключена, 0=защита установлена). Выбираем 0. Бит 3 PWRTE – разрешение задержки при включении питания (1=задержка отключена, 0=задержка установлена). Выбираем 0. Бит 2 WDTE – разрешение включения сторожевого таймера (1=сторожевой таймер включен, 0=сторожевой таймер отключен). Выбираем 0. Биты 1-0 FOSC1-FOSC0 – выбор режима тактового генератора (11=RC-генератор, 10=HS-резонатор, 01=XT-резонатор, 00=LP-резонатор). Выбираем 01 т.к. используем стандартный керамический резонатор 4МГц.

**Описание переменных**

Данный блок соответствует приведенной ниже части программы.

;==============================================================

;описание используемых переменных ;(назначение адресов ячеек для хранения переменных)

NUMIMP equ 0x0C ;рабочая переменная для подсчета числа импульсов

TIMER1 equ 0x0D ;рабочая переменная для подсчета времени 1

TIMER2 equ 0x0E ;рабочая переменная для подсчета времени 2

LAPSE equ 0x0F ;рабочая переменная для подсчета погрешности перевода времени

DIGIT1 equ 0x10 ;рабочая переменная индикатора дециметров

DIGIT2 equ 0x11 ;рабочая переменная индикатора метров

DIGIT3 equ 0x12 ;рабочая переменная индикатора декаметров

;==============================================================

В этом блоке описывается в каких ячейках ОЗУ (регистрах общего применения) будут хранится значения наших переменных. Назначение переменных понятно из комментариев приведенной выше части листинга программы.

**Исполняемая программа**

Данный блок соответствует приведенной ниже части программы.

;==============================================================

;исполняемая программа

org 0х000 ;вектор сброса процессора, начальный адрес

clrf PORTA ;очистили выходные защелки порта А

clrf PORTB ;и порта В

clrf TMR0 ;очистка таймера TMR0

bsf STATUS, RP0 ;включили банк 1

movlw b’00011110’ ;настроили на вывод линию RA0,

movwf PORTA ;остальные линии порта A на ввод

movlw b’00000000’ ;настроили на вывод все линии порта B

movwf PORTB ;RB0…RB7

bcf OPTION\_REG, 7 ;включили подтягивающие резисторы

bcf OPTION\_REG, 5 ;включили режим таймера для TMR0

bcf STATUS, RP0 ;включили банк 0

;==============================================================

Указываем адрес начала программ, вектор сброса процессора. Обнуляем значения в выходных защелках обеих портов. Обнуляем значение таймера. Затем установив в 1 бит RP0 регистра STATUS, мы получаем доступ к регистровому банку 1. Теперь обращаясь к регистрам PORTA и PORTB, мы обращаемся не к выходным защелкам, а к регистрам состояния этих портов, и настраиваем часть линий на ввод, а часть линий на вывод, что соответствует схеме приведенной в приложении. Используя бит 7 RBPU регистра OPTION\_REG включаем встроенную нагрузку порта B, для чего устанавливаем бит в 0. С помощью бита 5 T0CS выбираем источник тактирования для таймера TMR0. Для использования внутренней тактовой частоты CLKOUT, устанавливаем в 0 этот бит. В конце установив в 0 бит RP0 регистра STATUS, мы получаем доступ к регистровому банку 0.

**Основная программа main**

Данный блок соответствует приведенной ниже части программы.

;==============================================================

main ;основная программа

call initialization ;вызываем подпрограмму инициализации

main1 call indication ;вызываем подпрограмму индикации

goto pressing ;вызываем подпрограмму обработки нажатия кнопки

main2 call transfer ;вызываем подпрограмму передачи импульсов

main3 goto waiting ;вызываем подпрограмму ожидания отраженного сигнала

main4 goto reception ;вызываем подпрограмму приема отраженного сигнала

main5 call calculation ;вызываем подпрограмму вычисления расстояния

;==============================================================

В данной части используются всего два оператора перехода: оператор вызова подпрограммы call и оператор безусловного перехода goto. Для наглядности приведем блок-схему основной программы main, в которую добавлены условия и порядок взаимодействия рассматриваемых далее подпрограмм.

Блок-схема приведена на рис. 9.

начало

подпрограмма инициализации

подпрограмма индикации

подпрограмма вычисления

подпрограмма нажатия кнопки

подпрограмма передачи

подпрограмма ожидания

подпрограмма приема

кнопка нажата

да

нет

перепол-нение

да

нет

сигнал есть

да

нет

сигнал наш

нет

да

Рис. 9. Блок-схема основной программы main.

Подпрограмма инициализации initialization

Данный блок соответствует приведенной ниже части программы.

;==============================================================

initialization ;подпрограмма инициализации и начальных условий

**обнуляем значения используемых переменных**

clrf PORTA ;очистили выходные защелки порта A

clrf NUMIMP ;обнуляем значение NUMIMP

clrf TIMER1 ;обнуляем значение TIMER1

clrf NUMCH ;обнуляем значение NUMCH

clrf TIMER2 ;обнуляем значение TIMER2

clrf LAPSE ;обнуляем значение LAPSE

clrf DIGIT1 ;обнуляем значение DIGIT1

clrf DIGIT2 ;обнуляем значение DIGIT2

clrf DIGIT3 ;обнуляем значение DIGIT3

**;устанавливаем значения используемых переменных**

movlw d'5' ;помещаем значение "5" для NUMIMP в аккумулятор

movwf NUMIMP ;помещаем значение из аккумулятора в NUMIMP

movlw d'100' ;помещаем значение "100" для TIMER1 в аккумулятор

movwf TIMER1 ;помещаем значение из аккумулятора в TIMER1

movlw d'100' ;помещаем значение "100" для TIMER2 в аккумулятор

movwf TIMER2 ;помещаем значение из аккумулятора в TIMER2

movlw d'57' ;помещаем значение "57" для LAPSE в аккумулятор

movwf LAPSE ;помещаем значение из аккумулятора в LAPSE

return ;возврат из подпрограммы

;==============================================================

Здесь мы обнуляем используемые переменные, и устанавливаем для них необходимые значения. Количество передаваемых импульсов NUMIMP в сигнале принимаем равным 5 (согласовано с преподавателем). И т.д.

**Подпрограмма индикации indication**

Данный блок соответствует приведенной ниже части программы.

;==============================================================

indication ;подпрограмма индикации измеренного расстояния

;-----------------------------------------------------устанавливаем биты подключения индикаторов

bsf DIGIT1, 6 ;устанавливаем бит 6 переменной DIGIT1 в 1

bsf DIGIT2, 7 ;устанавливаем бит 7 (точка) переменной DIGIT2 в 1

bsf DIGIT2, 5 ;устанавливаем бит 5 переменной DIGIT2 в 1

bsf DIGIT3, 4 ;устанавливаем бит 4 переменной DIGIT3 в 1

;-----------------------------------------------------выводим значение DIGIT1 на индикатор

movf DIGIT1, 0 ;помещаем значение переменной DIGIT1 в аккумулятор

movwf PORTB ;выводим значение в порт В

**устанавливаем задержку отображения**

clrf TMR0 ;обнуляем таймер и

bcf INTCON, TOIF ;устанавливаем в 0 флаг прерывания по его переполнению

indc1 btfss INTCON, TOIF ;проверяем флаг прерывания по переполнению таймера

goto indc1 ;если флаг равен 0 (счетчик не переполнен)-выполняем проверку

;если флаг равен 1 (счетчик переполнен) - переходим к следующему

;индикатору

;-----------------------------------------------------выводим значение DIGIT2 на индикатор

movf DIGIT2, 0 ;помещаем значение переменной DIGIT1 в аккумулятор

movwf PORTB ;выводим значение в порт В

**устанавливаем задержку отображения**

clrf TMR0 ;обнуляем таймер и

bcf INTCON, TOIF ;устанавливаем в 0 флаг прерывания по его переполнению

indc2 btfss INTCON, TOIF ;проверяем флаг прерывания по переполнению таймера

goto indc2 ;если флаг равен 0 (счетчик не переполнен)-выполняем проверку

если флаг равен 1 (счетчик переполнен) - переходим к следующему индикатору

;-----------------------------------------------------выводим значение DIGIT3 на индикатор

movf DIGIT3, 0 ;помещаем значение переменной DIGIT3 в аккумулятор

movwf PORTB ;выводим значение в порт В

;устанавливаем задержку отображения

clrf TMR0 ;обнуляем таймер и

bcf INTCON, TOIF ;устанавливаем в 0 флаг прерывания по его переполнению

indc3 btfss INTCON, TOIF ;проверяем флаг прерывания по переполнению таймера

goto indc3 ;если флаг равен 0 (счетчик не переполнен)-выполняем проверку

;если флаг равен 1 (счетчик переполнен)-

;---------------------------------------------------------------------------------------------------------

return ;возврат из подпрограммы

;==============================================================

В первой части подпрограммы indication мы устанавливаем «биты точки и питания» индикаторов согласно схеме. В переменной DIGIT1 будет храниться значение дециметров от 0 до 9 т.к. используемые в схеме индикаторы являются семи сегментными с дополнительным выводом для десятичного разделителя – точки, в переменной DIGIT2 – значение метров, в переменной DIGIT3 – значение декаметров соответственно. Таким образом согласно схеме для отображения цифр на индикаторах через дешифратор используется всего четыре линии порта B: RB0, RB1, RB2, RB3 для передачи битов 0, 1, 2 и 3 соответственно. Вычисленные отображаемые значения будут храниться в переменных DIGIT1…3 поразрядно, для этого используется первые четыре бита каждой переменной. Линии RB4, RB5 и RB6 порта В мы используем для поочередной подачи питания на индикаторы. Линия RB4 на индикатор декаметров (отображение значения DIGIT3), RB5 на индикатор метров (отображение значения DIGIT2), RB6 на индикатор дециметров (отображение значения DIGIT1). Для этого мы используем следующие биты переменных: бит 6 - DIGIT1, бит 5 - DIGIT2 и бит 4 - DIGIT3 согласно схеме. Линия RB7 порта В используется для подачи сигнала на десятичную точку индикатора метров (переменная DIGIT2). Таким образом используется весь порт В и мы можем, заранее установив биты в переменных посылать их на порт В поочередно, соответственно поочередно подключая индикаторы. Для наглядности ниже приведен рисунок 10.

0

DIGIT1

1

0

0

x

x

x

x

7

6

5

4

3

2

1

0

биты

значение

1

DIGIT2

0

1

0

x

x

x

x

7

6

5

4

3

2

1

0

биты

значение

0

DIGIT3

0

0

1

x

x

x

x

7

6

5

4

3

2

1

0

биты

значение

RB7

PORT B

3

2

1

0

линии

на дешифратор

RB6

RB5

RB4

RB3

RB2

RB1

RB0

Дешифратор

4

2

1

5

вход

8

4

2

1

вывод

Рис. 10. Схема индикации.

Далее выводим значение DIGIT1 на порт В. Организуем задержку отображения на индикаторе при помощи таймера TMR0 на 256 мкс. Затем эту же процедуру производим для DIGIT2 и DIGIT3 отображая их значения на индикаторах.

**Подпрограмма нажатия кнопки pressing**

Данный блок соответствует приведенной ниже части программы.

;==============================================================

pressing ;подпрограмма обработки нажатия кнопки

btfss PORTA, 2 ;проверяем линию RA2 порта А (кнопку s2),

goto main1 ;если не нажата возвращаемся в программу (на indication)

goto main2 ;если нажата-возврат из подпрограммы

;==============================================================

При выполнении этой подпрограммы происходит тестирование присутствия сигнала (+5В) на линии RA2 порта A. Если на линию поступает напряжение (кнопка нажата) – выполняется команда перехода в основную программу на соответствующую метку (main2). В противном случае (кнопка не нажата) переход осуществляется на другую метку основной программы (main2).

**Подпрограмма передачи transfer**

Данный блок соответствует приведенной ниже части программы.

;==============================================================

transfer ;подпрограмма передачи импульсов

;---------------------------------------------------------------------------------------------------------

movlw d'5' ;помещаем значение "5" для NUMIMP в аккумулятор

movwf NUMIMP ;помещаем значение из аккумулятора в NUMIMP

;-----------------------------------------------------передача сигнала

tran1 bsf PORTA, 0 ;устанавливаем в 1 линию RA0 порта А

movlw d'249' ;помещаем значение "249" для таймера в аккумулятор

movwf TMR0 ;помещаем значение из аккумулятора в таймер

bcf INTCON, TOIF ;устанавливаем в 0 флаг прерывания по переполнению таймера

tran2 btfss INTCON, TOIF ;проверяем флаг прерывания по переполнению таймера

goto tran2 ;если флаг равен 0 (счетчик не переполнен)-выполняем проверку

;если флаг равен 1 (счетчик переполнен)-

;-----------------------------------------------------задержка сигнала

bcf PORTA, 0 ;устанавливаем в 0 линию RA0 порта А

movlw d'251' ;помещаем значение "251" для таймера в аккумулятор

movwf TMR0 ;помещаем значение из аккумулятора в таймер

bcf INTCON, TOIF ;устанавливаем в 0 флаг прерывания по переполнению таймера

tran3 btfss INTCON, TOIF ;проверяем флаг прерывания по переполнению таймера

goto tran3 ;если флаг равен 0 (счетчик не переполнен)-выполняем проверку

;если флаг равен 1 (счетчик переполнен)-

;-----------------------------------------------------считаем импульсы

decfsz NUMIMP, 1 ;уменьшаем зачение NUMIMP на "1" если NUMIMP не равен "0"

;(передано<5импульсов)-

goto tran1 ;переходим в начало подпрограммы на tran1 если NUMIMP равен "0"

;-----------------------------------------------------(передано 5 импульсов)-

movlw d'3' ;помещаем значение "3" для NUMIMP в аккумулятор

movwf NUMIMP ;помещаем значение из аккумулятора в NUMIMP

;---------------------------------------------------------------------------------------------------------

return ;возврат из подпрограммы

;==============================================================

Данная подпрограмма выводит на линию RA0 порта А микроконтроллера 5 импульсов частотой 40 кГц. Учитывая, что мы не подключаем предделитель для таймера-счетчика TMR0 и в схеме используется керамический резонатор частотой 4 МГц имеем тактовую частоту микроконтроллера равную 1 МГц, т.е. одна команда выполняется за 1 мкс. Следовательно один импульс частотой 40 кГц имеет продолжительность 1/40000=25 мкс. Т.е. мы должны сформировать один импульс за 25 мкс. Блок-схема данной подпрограммы представлена на рис. 11. Значения для таймера подобраны таким образом, чтобы сформировать импульс на линии продолжительностью около 25 мкс. В конце подпрограммы устанавливается новое значение для переменной NUMIMP, т.к. оно потребуется для работы другой подпрограммы.

начало

формируем сигнал заданной продолжительности

конец

импуль-сов 5?

да

нет

формируем задержку заданной продолжительности

считаем количество импульсов

Рис. 11. Блок-схема подпрограммы transfer.

**Подпрограмма ожидания waiting**

Данный блок соответствует приведенной ниже части программы.

;==============================================================

waiting ;подпрограмма ожидания отраженного сигнала

;-----------------------------------------------------задержка начала приема сигнала

movlw d'151' ;помещаем значение "151" для таймера в аккумулятор

movwf TMR0 ;помещаем значение из аккумулятора в таймер

bcf INTCON, TOIF ;устанавливаем в 0 флаг прерывания по переполнению таймера

wait1 btfss INTCON, TOIF ;проверяем флаг прерывания по переполнению таймера

goto wait1 ;если флаг равен 0 (счетчик не переполнен)-выполняем проверку

;если флаг равен 1 (счетчик переполнен)-

;-----------------------------------------------------начинаем счет времени

**ожидание приема (по 100 мкс)**

wait3 movlw d'161' ;помещаем значение "161" для таймера в аккумулятор

movwf TMR0 ;помещаем значение из аккумулятора в таймер

bcf INTCON, TOIF ;устанавливаем в 0 флаг прерывания по переполнению таймера

wait2 btfsc PORTA, 1 ;проверяем наличие сигнала на линии RA1 порта А

goto wait4 ;если сигнал есть-переходим на метку wait4

если сигнала нет-продолжаем счет времени

btfss INTCON, TOIF ;проверяем флаг прерывания по переполнению таймера

goto wait2 ;если флаг равен 0 (счетчик не

;-----------------------------------------------------переполнен)-выполняем проверку

decfsz TIMER1, 1 ;если флаг равен 1 (счетчик переполнен)-уменьшаем TIMER1 на 1

goto wait3 ;если TIMER1 не равен "0"

;-----------------------------------------------------продолжаем ожидание если TIMER1 равен "0"-

movlw d'100' ;помещаем значение "100" для TIMER1 в аккумулятор

movwf TIMER1 ;помещаем значение из аккумулятора в TIMER1

decfsz TIMER2, 1 ;уменьшаем значение TIMER2 на 1

goto wait3 ;если TIMER2 не равен "0"

;-----------------------------------------------------продолжаем ожидание

;если TIMER2 равен "0" (сигнал уже не вернется)-заполняем

;значения DIGIT1..3

movlw d'9' ;помещаем значение "9" для переменной DIGIT1 в аккумулятор

movwf DIGIT1 ;помещаем значение из аккумулятора в DIGIT1

movlw d'9' ;помещаем значение "9" для переменной ;DIGIT2 в аккумулятор

movwf DIGIT2 ;помещаем значение из аккумулятора в DIGIT2

movlw d'9' ;помещаем значение "9" для переменной ;DIGIT3 в аккумулятор

movwf DIGIT3 ;помещаем значение из аккумулятора в DIGIT3

;---------------------------------------------------------------------------------------------------------

goto main1 ;возврат из подпрограммы (сигнала нет)

wait4 goto main4 ;возврат из подпрограммы (сигнал есть)

;==============================================================

начало

необходимая задержка

запуск

таймера

TIMER1-1

TIMER1=100

TIMER2-1

сигнал есть?

да

нет

таймер досчитал

да

=0

нет

≠0

TIMER1

=0

≠0

TIMER2

DIGIT1…3=9

на indication

на reception

Рис. 12. Блок-схема подпрограммы waiting.

В начальной части подпрограммы производится небольшая задержка начала счета времени ожидания для того, чтобы произошло затухание помех вызванных излучением сигнала в предыдущей подпрограмме. Затем запускается таймер на 100 мкс. Т.к. мы будем считать промежутки времени по 100 мкс. Блок-схема подпрограммы представлена на рис. 12. Таким образом максимальный период ожидания составит 100\*100\*100=1000000 мкс, что составляет 1с. Принимая скорость распространения звука равной 331,4 м/с – это более чем достаточно, т.к. по условию задания максимальная измеряемая дальность устройства составит 99,9 метра.

**Подпрограмма приема reception**

Данный блок соответствует приведенной ниже части программы.

;==============================================================

reception ;подпрограмма приема отраженного сигнала

;---------------------------------------------------------------------------------------------------------

ожидание окончания сигнала

recp1 btfsс PORTA, 1 ;проверяем наличие сигнала на линии RA1 порта А

goto recp1 ;если сигнала есть-продолжаем ожидание

;-----------------------------------------------------если сигнала нет-начинаем проверку

recp6 movlw d'251' ;помещаем значение "251" для таймера в аккумулятор

movwf TMR0 ;помещаем значение из аккумулятора в таймер

bcf INTCON, TOIF ;устанавливаем в 0 флаг прерывания по переполнению таймера

recp1 btfss INTCON, TOIF ;проверяем флаг переполнения таймера если таймер не переполнен-

btfsc PORTA, 1 ;проверяем отсутствие сигнала на линии RA1 порта А

;если таймер переполнен или есть

goto recp2 ;сигнал-идем на метку recp2 если таймер не переполнен и нет

goto recp1 ;сигнала-идем на метку recp1

;-----------------------------------------------------проверяем флаг переполнения таймера

recp2 btfss INTCON, TOIF ;если таймер не переполнен-

goto recp3 ;сигнал не наш-идем на метку recp3

;---------------------------------------------------------------------------------------------------------

movlw d'252' ;помещаем значение "252" для таймера в аккумулятор

movwf TMR0 ;помещаем значение из аккумулятора в таймер

bcf INTCON, TOIF ;устанавливаем в 0 флаг прерывания по переполнению таймера

recp4 btfss INTCON, TOIF ;проверяем флаг переполнения таймера если таймер не переполнен-

btfss PORTA, 1 ;проверяем наличие сигнала на линии RA1 порта А если таймер переполнен или нет

goto recp5 ;сигнала-идем на метку recp5 если таймер не переполнен и есть

goto recp4 ;сигнал-идем на метку recp4

;-----------------------------------------------------проверяем флаг переполнения таймера

recp5 btfss INTCON, TOIF ;если таймер не переполнен-

goto recp3 ;сигнал не наш-идем на метку recp3

;---------------------------------------------------------------------------------------------------------

decfsz NUMIMP, 1 ;уменьшаем значение NUMIMP на "1"

goto recp6 ;если NUMIMP не равен "0"-продолжаем проверку

goto main5 ;если NUMIMP равен "0"-возврат из подпрограммы

recp3 goto main3 ;продолжение ожидания сигнала

;==============================================================

Блок-схема данной подпрограммы представлена на рис. 13. Т.к. переход в подпрограмму осуществляется при наличии сигнала на линии RA1 порта А микроконтроллера, то сначала дожидаемся окончания сигнала. Затем производим тестирование отсутствия сигнала на протяжении определенного времени, если до истечения данного промежутка времени сигнал появился, то делаем вывод, что принят не наш сигнал и выходим из подпрограммы и продолжаем ожидание правильного сигнала в подпрограмме waiting. После этого производим тестирование наличия сигнала на протяжении определенного времени, если до истечения данного промежутка времени сигнал пропал, то делаем вывод, что принят не наш сигнал и выходим из подпрограммы и продолжаем ожидание правильного сигнала в подпрограмме waiting. Если же в течение определенного времени сигнала не было, а затем в течении определенного времени сигнал был, то делаем вывод, что был успешно произведен прием одного импульса нашего сигнала, уменьшаем счетчик принятых импульсов на 1 и возвращаемся на метку recp6 рассматриваемой подпрограммы и производим все действия по приему импульса сигнала еще раз. Прием импульсов проводится до тех пор пока счетчик импульсов NUMIMP не станет равен нулю, и мы можем считать, что приняли три импульса нашего сигнала. После этого переходим в подпрограмму calculation. Работа подпрограммы более понятна из приведенной блок-схемы.

начало

нет

сигнал есть?

на waiting

таймер досчитал

да

да

запускаем

таймер

нет

запускаем

таймер

таймер досчитал

нет

да

сигнала нет?

да

нет

нет

таймер досчитал

да

NUMIMP-1

таймер досчитал

нет

да

сигнала нет?

да

нет

нет

да

NUMIMP=0

на waiting

Рис. 13. Блок-схема подпрограммы reception.

**Подпрограмма вычисления calculation**

Данный блок соответствует приведенной ниже части программы.

;==============================================================

calculation ;подпрограмма вычисления расстояния инвертируем значения TIMER1 и

;-----------------------------------------------------TIMER2

movf TIMER1,0 ;записываем значение переменной TIMER1 в аккумулятор

clrf TIMER1 ;обнуляем значение TIMER1

sublw d'100' ;из числа 100 вычитаем значение аккумулятора

movwf TIMER1 ;записываем полученное значение в переменную TIMER1

movf TIMER2,0 ;записываем значение переменной TIMER2 в аккумулятор

clrf TIMER2 ;обнуляем значение TIMER2

sublw d'100' ;из числа 100 вычитаем значение аккумулятора

movwf TIMER2 ;записываем полученное значение в переменную TIMER2

;-----------------------------------------------------устанавливаем значения для LAPSE и DIGIT1..3

clrf LAPSE ;обнуляем значение LAPSE

movlw d'57' записываем в аккумулятор значение "57"

movwf LAPSE ;записываем значение аккумулятора в переменную LAPSE

clrf DIGIT1 ;обнуляем значение DIGIT1

clrf DIGIT2 ;обнуляем значение DIGIT2

clrf DIGIT3 ;обнуляем значение DIGIT3

;-----------------------------------------------------записываем в аккумулятор вычитаемое

calc3 movlw d'6' ;"6"

subwf TIMER1, 1 ;вычитаем из переменной TIMER1 значение аккумулятора

проверяем флаги регистра STATUS

btfss STATUS, C ;флаг Z-нулевого результата

btfsc STATUS, Z ;и флаг C-переноса

goto calc4 ;если флаги Z и C не равны 0

goto calc1 ;если Z=0 и C=0 значит результат отрицательный

;-----------------------------------------------------меняем значения DIGIT1…3 и LAPSE

calc4 incf DIGIT1, 1 ;увеличиваем переменную DIGIT1 на "1"

movlw d'10' ;записываем в аккумулятор значение "10"

subwf DIGIT1, 0 ;вычитаем значение аккумулятора из переменной DIGIT1

btfss STATUS, Z ;проверяем результат на "0"

goto calc2 если результат не "0" идем на метку

;-----------------------------------------------------calc2

decf DIGIT1, 1 ;если результат равен "0" уменьшаем переменную DIGIT1 на "1"

incf DIGIT2, 1 ;увеличиваем переменную DIGIT2 на "1" (переносим разряд)

movlw d'10' ;записываем в аккумулятор значение "10"

subwf DIGIT2, 0 ;вычитаем значение аккумулятора из переменной DIGIT2

btfss STATUS, Z ;проверяем результат на "0"

goto calc2 ;если результат не "0" идем на метку

;-----------------------------------------------------calc2

decf DIGIT2, 1 ;если результат равен "0" уменьшаем переменную DIGIT2 на "1"

incf DIGIT3, 1 ;увеличиваем переменную DIGIT3 на "1" (переносим разряд)

movlw d'10' ;записываем в аккумулятор значение "10"

subwf DIGIT3, 0 ;вычитаем значение аккумулятора из переменной DIGIT3

btfss STATUS, Z ;проверяем результат на "0"

goto calc2 ;если результат не "0" идем на метку calc2

decf DIGIT3, 1 ;если результат равен "0" уменьшаем переменную DIGIT3 на "1"

goto calc5 ;выходим из подпрограммы т.к. достигнуто значение 99,9

;-----------------------------------------------------корректируем погрешность перевода

calc2 decfz LAPSE, 1 ;уменьшаем значение LAPSE на 1

goto calc3 ;если переменная LAPSE не равна "0" - идем на метку calc3

movlw d'57' ;записываем в аккумулятор значение "57"

movwf LAPSE ;записываем значение аккумулятора в переменную LAPSE

movlw d'2' ;записываем в аккумулятор значение "2"

addwf TIMER1, 1 ;прибавляем к переменной TIMER1 значение аккумулятора

goto calc3 ;идем на метку calc3

;-----------------------------------------------------корректируем переменные TIMER1 и TIMER2

calc1 addwf TIMER1, 1 ;прибавляем к переменной TIMER1 значение аккумулятора

movlw d'100' ;записываем в аккумулятор значение "100"

addwf TIMER1, 1 ;прибавляем к переменной TIMER1 значение аккумулятора

decf TIMER2 ;уменьшаем значение переменной TIMER2 на "1"

проверяем флаги регистра STATUS

btfss STATUS, C ;флаг C-переноса

btfsc STATUS, Z ;и флаг Z-нулевого результата

goto calc4 ;если результат не отрицательный-продолжаем

если Z=0 и C=0 значит результат отрицательный-

;---------------------------------------------------------------------------------------------------------calc5 return ;возврат из подпрограммы

;==============================================================

Блок-схема данной подпрограммы представлена на рис. 14. Поясним некоторые моменты. Принимая скорость звука равной 331,4 м/с, имеем 331,4\*1000/1000000 мм/мкс или 0,3314 мм/мкс. Т.к. по условию задания точность измерений составляет 0,1 м, а измеренное нами время составляет двойную величину, то логично вести подсчет промежутками времени за которое ультразвук проходит 0,2 м. Т.е. 200/0,3314=603,5=(600+3,5) мкс. Каждые 600 мкс у нас накапливается погрешность в 3,5 мкс. 57\*3,5=199,5≈200. Т.е. каждый промежуток времени 57\*600 мкс у нас возникает погрешность в 200 мкс. Хотя она и составляет 1/3 от заданной точности мы, все равно будем ее учитывать. Принцип работы подпрограммы ясен из блок-схемы.

начало

вычисляем TIMER1…2

начальные значения

DIGIT1+1

TIMER1+6

TIMER1+100

TIMER2-1

TIMER1<0

да

нет

нет

DIGIT1

=10

LAPSE

=0

LAPSE-1

конец

TIMER1-6

TIMER2<0

да

нет

да

нет

да

LAPSE=57

TIMER1+2

DIGIT1-1

DIGIT2+1

DIGIT2

=10

да

нет

DIGIT2-1

DIGIT3+1

DIGIT3

=10

да

нет

DIGIT3-1

Рис. 14. Блок-схема подпрограммы calculation.**Заключение**

В процессе выполнения курсового проекта, было спроектировано устройство ультразвукового измерения дальности. В пояснительной записке представлено подробное описание схемы устройства, описание программного обеспечения.

В приложении представлена принципиальная схема разработанного устройства и приведен листинг программы.

В ходе выполнения курсового проекта были решены следующие задачи:

* синтез и разработка принципиальной схемы устройства;
* разработка программного обеспечения;
* разработка конструкторской документации проекта (пояснительная записка).

**Библиографический список**

1. В. С. Яценков «Микроконтроллеры Microchip. Практическое руководство».
2. Документация к контроллерам фирмы Microchip.

**Приложения**

;==============================================================

программа для ультразвукового дальномера

;==============================================================

список директив

list P=16C84 ;директива определяющая тип процессора

#include <P16C84.INC> ;файл, описывающий специфические переменные, соответствующие процессору

\_config b'00000000000001' ;слово конфигурации CPU

;==============================================================

описание используемых переменных (назначение адресов ячеекдля хранения переменных)

NUMIMP equ 0x0C ;рабочая переменная для подсчета числа импульсов

TIMER1 equ 0x0D ;рабочая переменная для подсчета времени 1

TIMER2 equ 0x0E ;рабочая переменная для подсчета времени 2

LAPSE equ 0x0F ;рабочая переменная для подсчетапогрешности перевода времени

DIGIT1 equ 0x10 ;рабочая переменная индикатора дециметров

DIGIT2 equ 0x11 ;рабочая переменная индикатора метров

DIGIT3 equ 0x12 ;рабочая переменная индикатора декаметров

;==============================================================

**исполняемая программа**

org 0х000 ;вектор сброса процессора, начальный адрес

clrf PORTA ;очистили выходные защелки порта А

clrf PORTB ;и порта В

clrf TMR0 ;очистка таймера TMR0

bsf STATUS, RP0 ;включили банк 1

movlw b’00011110’ ;настроили на вывод линию RA0,

movwf PORTA ;остальные линии порта A на ввод

movlw b’00000000’ ;настроили на вывод все линии порта B

movwf PORTB ;RB0…RB7

bcf OPTION\_REG, 7 ;включили подтягивающие резисторы

bcf OPTION\_REG, 5 ;включили режим таймера для TMR0

bcf STATUS, RP0 ;включили банк 0

;==============================================================

main ;основная программа

call initialization ;вызываем подпрограмму инициализации

main1 call indication ;вызываем подпрограмму индикации

goto pressing ;вызываем подпрограмму обработки нажатия кнопки

main2 call transfer ;вызываем подпрограмму передачи импульсов

main3 goto waiting ;вызываем подпрограмму ожидания отраженного сигнала

main4 goto reception ;вызываем подпрограмму приема отраженного сигнала

main5 call calculation ;вызываем подпрограмму вычисления расстояния

;==============================================================

initialization ;подпрограмма инициализации и начальных условий

обнуляем значения используемых переменных

clrf PORTA ;очистили выходные защелки порта A

clrf NUMIMP ;обнуляем значение NUMIMP

clrf TIMER1 ;обнуляем значение TIMER1

clrf NUMCH ;обнуляем значение NUMCH

clrf TIMER2 ;обнуляем значение TIMER2

clrf LAPSE ;обнуляем значение LAPSE

clrf DIGIT1 ;обнуляем значение DIGIT1

clrf DIGIT2 ;обнуляем значение DIGIT2

clrf DIGIT3 ;обнуляем значение DIGIT3

устанавливаем значения используемых переменных

movlw d'5' ;помещаем значение "5" для NUMIMP в аккумулятор

movwf NUMIMP ;помещаем значение из аккумулятора в NUMIMP

movlw d'100' ;помещаем значение "100" для TIMER1 в аккумулятор

movwf TIMER1 ;помещаем значение из аккумулятора в TIMER1

movlw d'100' ;помещаем значение "100" для TIMER2 в аккумулятор

movwf TIMER2 ;помещаем значение из аккумулятора в TIMER2

movlw d'57' ;помещаем значение "57" для LAPSE в аккумулятор

movwf LAPSE ;помещаем значение из аккумулятора в LAPSE

return ;возврат из подпрограммы

;==============================================================

indication ;подпрограмма индикации измеренного расстояния

;-----------------------------------------------------устанавливаем биты подключения индикаторов

bsf IGIT1, 6 ;устанавливаем бит 6 переменной DIGIT1 в 1

bsf DIGIT2, 7 ;устанавливаем бит 7 (точка) переменной DIGIT2 в 1

bsf DIGIT2, 5 ;устанавливаем бит 5 переменной DIGIT2 в 1

bsf DIGIT3, 4 ;устанавливаем бит 4 переменной DIGIT3 в 1

;-----------------------------------------------------выводим значение DIGIT1 на индикатор

movf DIGIT1, 0 ;помещаем значение переменной DIGIT1 в аккумулятор

movwf PORTB ;выводим значение в порт В

устанавливаем задержку отображения

clrf TMR0 обнуляем таймер и

bcf INTCON, TOIF ;устанавливаем в 0 флаг прерывания по его переполнению

indc1 btfss INTCON, TOIF ;проверяем флаг прерывания по переполнению таймера

goto indc1 если флаг равен 0 (счетчик не переполнен)-выполняем проверку если флаг равен 1 (счетчик переполнен) - переходим к следующему индикатору

;-----------------------------------------------------выводим значение DIGIT2 на индикатор

movf DIGIT2, 0 ;помещаем значение переменной DIGIT1 в аккумулятор

movwf PORTB ;выводим значение в порт В

устанавливаем задержку отображения

clrf TMR0 ;обнуляем таймер и

bcf INTCON, TOIF ;устанавливаем в 0 флаг прерывания по его переполнению

indc2 btfss INTCON, TOIF ;проверяем флаг прерывания по переполнению таймера

goto indc2 если флаг равен 0 (счетчик не переполнен)-выполняем проверку если флаг равен 1 (счетчик переполнен) - переходим к следующему индикатору

;-----------------------------------------------------выводим значение DIGIT3 на индикатор

movf DIGIT3, 0 ;помещаем значение переменной DIGIT3 в аккумулятор

movwf PORTB ;выводим значение в порт В

;устанавливаем задержку отображения

clrf TMR0 ;обнуляем таймер и

bcf INTCON, TOIF ;устанавливаем в 0 флаг прерывания по его переполнению

indc3 btfss NTCON, TOIF ;проверяем флаг прерывания по переполнению таймера

goto indc3 ;если флаг равен 0 (счетчик не переполнен)-выполняем проверку если флаг равен 1 (счетчик переполнен)-

;---------------------------------------------------------------------------------------------------------

return ;возврат из подпрограммы

;==============================================================

pressing ;подпрограмма обработки нажатия кнопки

btfss PORTA, 2 ;проверяем линию RA2 порта А (кнопку s2),

goto main1 ;если не нажата возвращаемся в программу (на indication)

goto main2 ;если нажата-возврат из подпрограммы

;==============================================================

transfer ;подпрограмма передачи импульсов

;---------------------------------------------------------------------------------------------------------

movlw d'5' ;помещаем значение "5" для NUMIMP в аккумулятор

movwf NUMIMP ;помещаем значение из аккумулятора в NUMIMP

;-----------------------------------------------------передача сигнала

tran1 bsf PORTA, 0 ;устанавливаем в 1 линию RA0 порта А

movlw d'249' ;помещаем значение "249" для таймера в аккумулятор

movwf TMR0 ;помещаем значение из аккумулятора в таймер

bcf INTCON, TOIF ;устанавливаем в 0 флаг прерывания по переполнению таймера

tran2 btfss INTCON, TOIF ;проверяем флаг прерывания по переполнению таймера

goto tran2 ;если флаг равен 0 (счетчик не переполнен)-выполняем проверку

если флаг равен 1 (счетчик переполнен)-

;-----------------------------------------------------задержка сигнала

bcf PORTA, 0 ;устанавливаем в 0 линию RA0 порта А

movlw d'251' ;помещаем значение "251" для таймера в аккумулятор

movwf TMR0 ;помещаем значение из аккумулятора в таймер

bcf INTCON, TOIF ;устанавливаем в 0 флаг прерывания по переполнению таймера

tran3 btfss INTCON, TOIF ;проверяем флаг прерывания по переполнению таймера

goto tran3 ;если флаг равен 0 (счетчик не переполнен)-выполняем проверку

если флаг равен 1 (счетчик переполнен)-

;-----------------------------------------------------считаем импульсы

decfsz NUMIMP, 1 ;уменьшаем зачение NUMIMP на "1"если NUMIMP не равен "0" (передано<5импульсов)-

goto tran1 ;переходим в начало подпрограммы на tran1 если NUMIMP равен "0"

;-----------------------------------------------------(передано 5 импульсов)-

movlw d'3' ;помещаем значение "3" для NUMIMP в аккумулятор

movwf NUMIMP ;помещаем значение из аккумулятора в NUMIMP

;---------------------------------------------------------------------------------------------------------

return ;возврат из подпрограммы

;==============================================================

waiting ;подпрограмма ожидания отраженного сигнала

;-----------------------------------------------------задержка начала приема сигнала

movlw d'151' ;помещаем значение "151" для таймера в аккумулятор

movwf TMR0 ;помещаем значение из аккумулятора в таймер

bcf INTCON, TOIF ;устанавливаем в 0 флаг прерывания по переполнению таймера

wait1 btfss INTCON, TOIF ;проверяем флаг прерывания по переполнению таймера

goto wait1 если флаг равен 0 (счетчик не переполнен)-выполняем проверку

если флаг равен 1 (счетчик переполнен)-

;-----------------------------------------------------начинаем счет времени

ожидание приема (по 100 мкс)

wait3 movlw d'161' ;помещаем значение "161" для таймера в аккумулятор

movwf TMR0 ;помещаем значение из аккумулятора в таймер

bcf INTCON, TOIF ;устанавливаем в 0 флаг прерывания по переполнению таймера

wait2 btfsc PORTA, 1 ;проверяем наличие сигнала на линии RA1 порта А

goto wait4 ;если сигнал есть-переходим на метку wait4

если сигнала нет-продолжаем счет времени

btfss INTCON, TOIF ;проверяем флаг прерывания по переполнению таймера

goto wait2 если флаг равен 0 (счетчик не

;-----------------------------------------------------переполнен)-выполняем проверку

decfsz TIMER1, 1 ;если флаг равен 1 (счетчик переполнен)-уменьшаем TIMER1 на 1

goto wait3 ;если TIMER1 не равен "0"

;-----------------------------------------------------продолжаем ожидание если TIMER1 равен "0"-

movlw d'100' ;помещаем значение "100" для TIMER1 в аккумулятор

movwf TIMER1 ;помещаем значение из аккумулятора в TIMER1

decfsz TIMER2, 1 ;уменьшаем значение TIMER2 на 1

goto wait3 ;если TIMER2 не равен "0"

;-----------------------------------------------------продолжаем ожидание

если TIMER2 равен "0" (сигнал уже не вернется)-заполняем ;значения DIGIT1..3

movlw d'9' ;помещаем значение "9" для переменной DIGIT1 в аккумулятор

movwf DIGIT1 ;помещаем значение из аккумулятора в DIGIT1

movlw d'9' ;помещаем значение "9" для переменной DIGIT2 в аккумулятор

movwf DIGIT2 ;помещаем значение из аккумулятора в DIGIT2

movlw d'9' ;помещаем значение "9" для переменной ;DIGIT3 в аккумулятор

movwf DIGIT3 ;помещаем значение из аккумулятора в DIGIT3

;---------------------------------------------------------------------------------------------------------

goto main1 ;возврат из подпрограммы (сигнала нет)

wait4 goto main4 ;возврат из подпрограммы (сигнал есть)

;==============================================================

reception ;подпрограмма приема отраженного сигнала

;---------------------------------------------------------------------------------------------------------

ожидание окончания сигнала

recp1 btfsс PORTA, 1 ;проверяем наличие сигнала на линии RA1 порта А

goto recp1 ;если сигнала есть-продолжаем ожидание

;-----------------------------------------------------если сигнала нет-начинаем проверку

recp6 movlw d'251' ;помещаем значение "251" для таймера в аккумулятор

movwf TMR0 ;помещаем значение из аккумулятора в таймер

bcf INTCON, TOIF ;устанавливаем в 0 флаг прерывания по переполнению таймера

recp1 btfss INTCON, TOIF ;проверяем флаг переполнения таймера

если таймер не переполнен-

btfsc PORTA, 1 ;проверяем отсутствие сигнала на линииRA1 порта А

если таймер переполнен или есть

goto recp2 ;сигнал-идем на метку recp2

;если таймер не переполнен и нет

goto recp1 ;сигнала-идем на метку recp1

;-----------------------------------------------------проверяем флаг переполнения таймера

recp2 btfss INTCON, TOIF ;если таймер не переполнен-

goto recp3 ;сигнал не наш-идем на метку recp3

;---------------------------------------------------------------------------------------------------------

movlw d'252' ;помещаем значение "252" для таймера в аккумулятор

movwf TMR0 помещаем значение из аккумулятора в таймер

bcf INTCON, TOIF ;устанавливаем в 0 флаг прерывания по переполнению таймера

recp4 btfss INTCON, TOIF ;проверяем флаг переполнения таймера если таймер не переполнен-

btfss PORTA, 1 ;проверяем наличие сигнала на линии RA1 порта А

если таймер переполнен или нет

goto recp5 ;сигнала-идем на метку recp5

;если таймер не переполнен и есть

goto recp4 ;сигнал-идем на метку recp4

;-----------------------------------------------------проверяем флаг переполнения таймера

recp5 btfss INTCON, TOIF ;если таймер не переполнен-

goto recp3 сигнал не наш-идем на метку recp3

;---------------------------------------------------------------------------------------------------------

decfsz NUMIMP, 1 ;уменьшаем значение NUMIMP на "1"

goto recp6 ;если NUMIMP не равен "0"-продолжаем проверку

goto main5 ;если NUMIMP равен "0"-возврат из подпрограммы

recp3 goto main3 ;продолжение ожидания сигнала

;==============================================================

calculation ;подпрограмма вычисления расстояния

;инвертируем значения TIMER1 и

;-----------------------------------------------------TIMER2

movf TIMER1,0 ;записываем значение переменной TIMER1 в аккумулятор

clrf TIMER1 ;обнуляем значение TIMER1

sublw d'100' ;из числа 100 вычитаем значение аккумулятора

movwf TIMER1 ;записываем полученное значение в переменную TIMER1

movf TIMER2,0 ;записываем значение переменной TIMER2 в аккумулятор

clrf TIMER2 ;обнуляем значение TIMER2

sublw d'100' ;из числа 100 вычитаем значение аккумулятора

movwf TIMER2 ;записываем полученное значение в переменную TIMER2

;-----------------------------------------------------устанавливаем значения для LAPSE и DIGIT1..3

clrf LAPSE ;обнуляем значение LAPSE

movlw d'57' ;записываем в аккумулятор значение "57"

movwf LAPSE ;записываем значение аккумулятора в переменную LAPSE

clrf DIGIT1 обнуляем значение DIGIT1

clrf DIGIT2 ;обнуляем значение DIGIT2

clrf DIGIT3 ;обнуляем значение DIGIT3

;-----------------------------------------------------записываем в аккумулятор вычитаемое

calc3 movlw d'6' ;"6"

subwf TIMER1, 1 ;вычитаем из переменной TIMER1 значение аккумулятора

проверяем флаги регистра STATUS

btfss STATUS, C ;флаг Z-нулевого результата

btfsc STATUS, Z ;и флаг C-переноса

goto calc4 ;если флаги Z и C не равны 0

goto calc1 ;если Z=0 и C=0 значит результат отрицательный

;-----------------------------------------------------меняем значения DIGIT1…3 и LAPSE

calc4 incf DIGIT1, 1 ;увеличиваем переменную DIGIT1 на "1"

movlw d'10' ;записываем в аккумулятор значение "10"

subwf DIGIT1, 0 ;вычитаем значение аккумулятора из переменной DIGIT1

btfss STATUS, Z ;проверяем результат на "0"

goto calc2 ;если результат не "0" идем на метку

;-----------------------------------------------------calc2

decf DIGIT1, 1 ;если результат равен "0" уменьшаем переменную DIGIT1 на "1"

incf DIGIT2, 1 ;увеличиваем переменную DIGIT2 на "1" (переносим разряд)

movlw d'10' ;записываем в аккумулятор значение "10"

subwf DIGIT2, 0 ;вычитаем значение аккумулятора из переменной DIGIT2

btfss STATUS, Z ;проверяем результат на "0"

goto calc2 ;если результат не "0" идем на метку

;-----------------------------------------------------calc2

decf DIGIT2, 1 ;если результат равен "0" уменьшаем переменную DIGIT2 на "1"

incf DIGIT3, 1 ;увеличиваем переменную DIGIT3 на "1" (переносим разряд)

movlw d'10' ;записываем в аккумулятор значение "10"

subwf DIGIT3, 0 ;вычитаем значение аккумулятора из переменной DIGIT3

btfss STATUS, Z ;проверяем результат на "0"

goto calc2 ;если результат не "0" идем на метку calc2

decf DIGIT3, 1 ;если результат равен "0" уменьшаем переменную DIGIT3 на "1"

goto calc5 выходим из подпрограммы т.к. достигнуто значение 99,9

;-----------------------------------------------------корректируем погрешность перевода

calc2 decfz LAPSE, 1 ;уменьшаем значение LAPSE на 1

goto calc3 ;если переменная LAPSE не равна "0"-идем на метку calc3

movlw d'57' ;записываем в аккумулятор значение "57"

movwf LAPSE ;записываем значение аккумулятора в переменную LAPSE

movlw d'2' ;записываем в аккумулятор значение "2"

addwf TIMER1, 1 ;прибавляем к переменной TIMER1 значение аккумулятора

goto calc3 ;идем на метку calc3

;-----------------------------------------------------корректируем переменные TIMER1 и TIMER2

calc1 addwf TIMER1, 1 ;прибавляем к переменной TIMER1 значение аккумулятора

movlw d'100' записываем в аккумулятор значение "100"

addwf TIMER1, 1 ;прибавляем к переменной TIMER1 значение аккумулятора

decf TIMER2 уменьшаем значение переменной TIMER2 на "1"

проверяем флаги регистра STATUS

btfss STATUS, C ;флаг C-переноса

btfsc STATUS, Z ;и флаг Z-нулевого результата

goto calc4 ;если результат не отрицательный-;продолжаем

если Z=0 и C=0 значит результат отрицательный-

;---------------------------------------------------------------------------------------------------------calc5 return возврат из подпрограммы

;==============================================================

end

;==============================================================