



Белорусский национальный  
технический университет

Международный институт  
дистанционного образования

МИДО  
БНТУ



**В.Ф.АЛЕКСЕЕВ**  
**В.И.ЛАКИН**

# **ПРОЕКТИРОВАНИЕ**

## **АППАРАТНО-ПРОГРАММНЫХ**

## **ВЫЧИСЛИТЕЛЬНЫХ СРЕДСТВ**



**Лабораторный практикум**

**БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
МЕЖДУНАРОДНЫЙ ИНСТИТУТ ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ**  
Кафедра информационных систем и технологий

**В.Ф.АЛЕКСЕЕВ, В.И.ЛАКИН**

## **ПРОЕКТИРОВАНИЕ АППАРАТНО- ПРОГРАММНЫХ ВЫЧИСЛИТЕЛЬНЫХ СРЕДСТВ**

Лабораторный практикум  
для студентов специальности 1-40 01 01  
«Программное обеспечение информационных технологий»  
и 1-40 01 02 «Информационные системы и технологии»

Минск БНТУ 2011

УДК 681.322 (075.8)  
ББК 32.973.26-02 я73  
А 47

*Рецензенты:* кафедра автоматизированных систем управления производством Белорусского государственного аграрно-технического университета (заведующий кафедрой кандидат технических наук, доцент О.Ч.Ролич); заведующий кафедрой автоматизированных информационных систем Минского института управления, доктор технических наук, профессор В.И.Курмашев

**А 47**

**Алексеев, В.Ф.**

Проектирование аппаратно-программных вычислительных средств : лаб. практикум для студ. спец. «Программное обеспечение информационных технологий» и «Информационные системы и технологии» / В.Ф. Алексеев, В.И. Лакин. – Минск : БНТУ, 2011. – 20 с.: ил. [Электронный ресурс]

ISBN

В лабораторном практикуме описаны пять работ, позволяющие студентам освоить программирование микроконтроллеров.

Пособие предназначено для студентов, магистрантов и аспирантов высших учебных заведений, инженеров-программистов, занимающимся вопросами проектирования аппаратно-программных вычислительных средств.

УДК 681.322 (075.8)  
ББК 32.973.26-02 я73

© Алексеев В.Ф., Лакин В.И., 2011

© Белорусский национальный технический университет, 2011

ISBN

## СОДЕРЖАНИЕ

|   |    |
|---|----|
| <b>ЛАБОРАТОРНАЯ РАБОТА № 1</b>                          |    |
| Разработка и программирование таймера.....              | 4  |
| <b>ЛАБОРАТОРНАЯ РАБОТА № 2</b>                          |    |
| Аналого-цифровые и цифро-аналоговые преобразователи.... | 9  |
| <b>ЛАБОРАТОРНАЯ РАБОТА № 3</b>                          |    |
| Датчики температур.....                                 | 14 |
| <b>ЛАБОРАТОРНАЯ РАБОТА № 4</b>                          |    |
| Оптические датчики.....                                 | 16 |
| <b>ЛАБОРАТОРНАЯ РАБОТА № 5</b>                          |    |
| Датчики перемещения.....                                | 18 |

## ЛАБОРАТОРНАЯ РАБОТА № 1

# РАЗРАБОТКА И ПРОГРАММИРОВАНИЕ ТАЙМЕРА

### *1.1. Цель работы*

Спроектировать таймер с частотой 800 Гц. Точность таймера: 0.001.

### *1.2. Краткие теоретические сведения*

Иногда бывает необходимо замерять промежутки времени, меньшие 55 мс (это стандартная скорость BIOS-таймера). Например, в играх скорость синхронизируется именно по таймеру. Таким образом, если скорость должна быть, скажем, 24 кадра в секунду, необходимо убыстрять таймер (например, до скорости 140 Гц, как это сделано в DOOM-е). Чем быстрее "тикает" таймер, тем с большей точностью можно замерять время. Есть и другие приложения программирования таймера, например, работа с PC-SPEAKER-ом (со встроенным динамиком) в wave-режиме (скорость таймера тогда достигает величин порядка 10-20 кГц).

### *1.3. Механизм работы таймера 8254*

Программируемый интервальный таймер типа Intel 8254 устроен довольно сложно. Он имеет 3 так называемых канала, которые работают одновременно и "тикают" с разной скоростью. Например, 0-й канал вызывает генерацию прерывания 8, 1-й используется DMA, а 2-й — для генерации звука. Каждый канал может работать в нескольких режимах. Но вся эта информация практически не представляет ценности. Рассмотрим, как же работает таймер. Он имеет кварцевый генератор (или часы), которые "тикают" ровно 119318 раз в секунду. Нулевой канал, который мы будем рассматривать, имеет так называемый счетчик. Этот счетчик возрастает на 1 каждый "тик" часов. Когда он достигает некоторого граничного значения (которое можно установить программно), он сбрасывается, и генерируется прерывание 8. BIOS устанавливает это граничное значение в 0, что для таймера обозначает 65536. То есть каждый 65536-й "тик" часов генерируется прерывание 8, оно "срабатывает" примерно  $119318/65536=18.2$  раз в секунду.

Если мы хотим установить свое граничное значение cnt, можно воспользоваться командами

```
outportb(0x43, 6); /* channel state */
outportb(0x40, (char)cnt); /* low byte */
outportb(0x40, (char)(cnt>>8)); /* hi byte */
```

Запустим подобную программу на выполнение, скажем, со значением cnt=65536/8. Когда она отработает, мы увидим странную вещь: часы в Нор-тоне станут идти в 8 раз быстрее! Это происходит оттого, что после выхода

из программы необходимо установить старый счетчик, который устанавливал BIOS, то есть 0. Однако и это не решит до конца проблему. Ведь на период работы программы часы все равно будут идти в 8 раз быстрее, то есть после выхода из программы собьется время. Этому можно избежать, если написать "заплату" на 8-е прерывание таким образом, чтобы BIOS-обработчик вызывался не каждый раз, а лишь каждый восьмой. При этом в остальные семь раз необходимо посылать в 20h-й порт значение 20h, чтобы разрешить следующие прерывания от таймера (если этого не сделать, прерывание 8 вызовется только один раз).

Широкое распространение получила таймерная интегральная схема типа 555(исправить). Эта микросхема работает в двух режимах: ждущего мультивибратора и режиме несинхронного мультивибратора (генератора прямоугольных колебаний).

#### ***1.4. Программируемые таймеры***

Интегральные схемы таймеров и счётчиков — однокристалльные компоненты — обычно жёстко монтируют в соответствии с требованиями конкретных приложений. Помимо них существуют ИС программируемых счётчиков и таймеров. Одним из таких приборов, который допускает перепрограммирование в соответствии с требованиями различных приложений, является программируемый интервальный таймер типа Intel 8253.

ИС 8253 содержит 3 независимых 16-разрядных счётчика, выполняющих счёт в обратном направлении. Каждый из счётчиков имеет свои собственные линии синхросигнала, выхода (заём) и стробирования (разрешения). Предварительная установка позволяет задавать двоичный или двоично-десятичный алгоритм счёта, причём каждый из счётчиков может работать в одном из шести различных режимов:

0 — по достижении таймером предельного показания на его выходе устанавливается низкий уровень. Подача низкого уровня на вход стробирования останавливает процесс счёта, а высокого уровня — разрешает его;

1 — это режим программируемого ждущего мультивибратора. На выходе низкий уровень поддерживается в течение всего времени, пока идёт процесс счёта синхроимпульсов. Фронт импульсов на выводе стробирования может как разрешать счёт, так и осуществлять сброс счётчика после поступления последнего синхроимпульса;

2 — данный режим позволяет получать последовательность импульсов. Выход характеризуется высоким уровнем в течение процесса счёта синхроимпульсов и приобретает низкий уровень на один синхрпереход после достижения предельного показания. Процесс повторяется многократно, пока на выводе стробирования существует высокий уровень. Низкий уровень на выводе стробирования блокирует счёт и вызывает появление на выходе высокого уровня, если он там не установился ранее;

3 — этот режим сходен с режимом 2, однако, при его использовании на выходе формируется волна прямоугольной формы. Если показание счётчика  $n$  чётное, выход имеет высокий уровень в течение  $n/2$  синхрпериодов и низ-

кий — также в течение  $n/2$  синхропериодов. Если показание счётчика нечётное, выход имеет высокий уровень в течение  $(n+1)/2$ , а низкий —  $(n-1)/2$  синхропериодов. Стробирование выполняется так же, как в режиме 2;

4 — выход в этом режиме, как правило, имеет высокий уровень; низкий уровень на выходе действует в течение одного синхропериода после достижения предельного показания. Низкий уровень на выводе стробирования блокирует процесс счёта, а высокий — разрешает счёт;

5 — выход здесь такой же, как и в режиме 4, и отличается от последнего лишь тем, что разрешение на счёт даётся по фронту импульса стробирования.

Частота следования синхроимпульсов может быть в пределах до 2 МГц, и поскольку нет необходимости в выборе коэффициента заполнения последовательности синхроимпульсов, равным 50%, таймер 8253 может использоваться в качестве счётчика событий. Запуск может выполняться как аппаратным, так и программируемым путём. Выходы таймера можно использовать для возбуждения прерываний, их можно опрашивать или снимать с них синхросигналы для тактирования других устройств; т.о., эта схемная реализация обладает достаточно высокой степенью гибкости. С помощью программных средств пользователь может задавать режим работы и загружать в счётчик начальное показание.

Под управлением функции `timer()` установленная на макетной плате ИС 8253 выдаёт временные метки с определённой частотой. В этой функции задаются число периодов между метками и используемая единица времени. Значение единицы измерения, равное 's', 'm', или 'u', служит для выбора длительности периода в с, мс или мкс соответственно. Длительность в мкс округляется до значения, ближайшего к величине, которая кратна 3,3543 мкс.

Так по команде `timer(5,'m')` счётчик выдаёт последовательность импульсов с периодом повторения 5 мс.

Приведем пример на Си, иллюстрирующий работу с таймером

```
unsigned BIOTimerSpeed=1;
unsigned TimerFreq=(unsigned) (1193181L/65536L);
void interrupt (*SvInt08)(void)=NULL;
void Set8254Counter(unsigned cnt)
{ long l=cnt;
  if(!cnt) l=65536L; /* если 0, то на самом деле 65536 */
  BIOTimerSpeed=(unsigned) (65536L/l);
  outportb(0x43, 6);
  outportb(0x40, (char)cnt);
  outportb(0x40, (char) (cnt>>8));
}
void interrupt NewInt08(void)
{ static cnt=0;
  cnt++; /* увеличить счетчик пропущенных тиков */
  /* если пора вызывать обработчик BIOS...*/
  if(cnt>=BIOTimerSpeed) { cnt=0; SvInt08(); }
  /* иначе разрешить следующие прерывания */
  else outportb(0x20, 0x20);
}
```

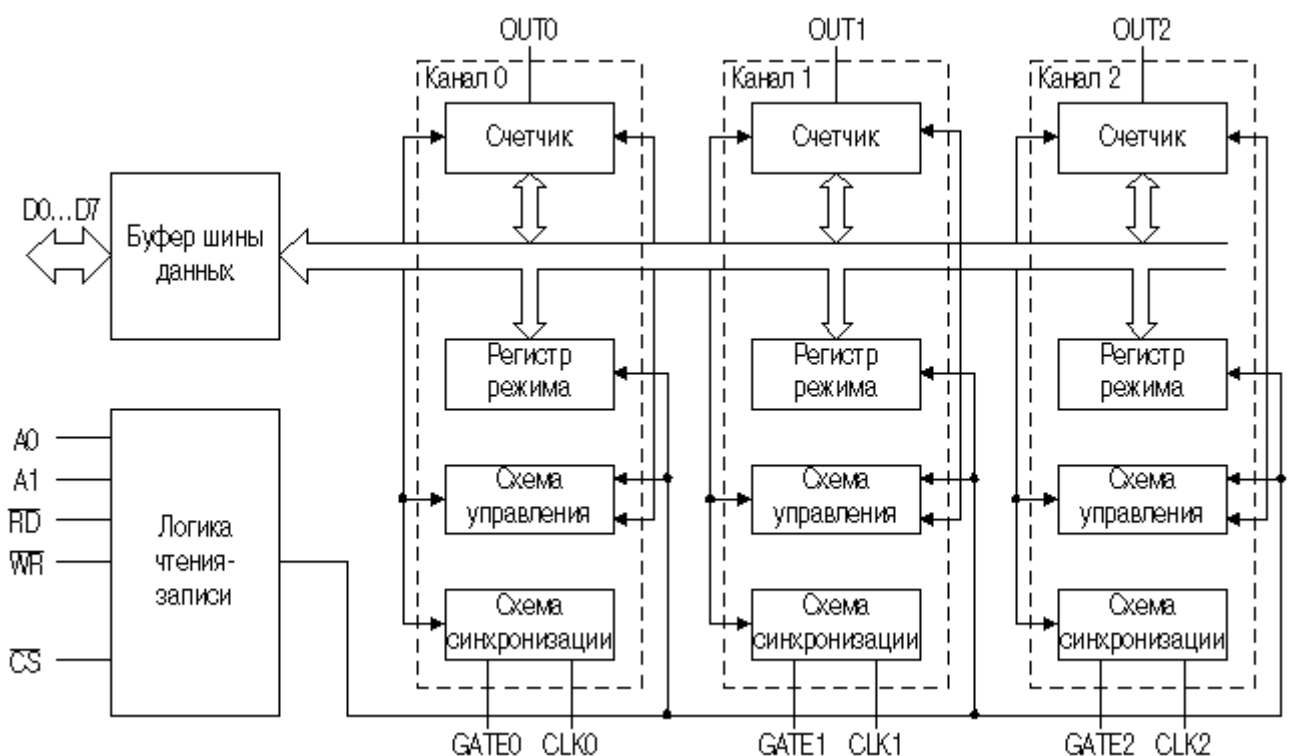
```

void DeactivateTimer(void); /* предварительное описание */
int SetTimer(unsigned cnt)
{ /* если передается 0, то отключить нашу
  процедуру обработки */
  if(!cnt)
  { Set8254Counter(0);
    /* отключить от прерывания */
    if(SvInt08) setvect(8,SvInt08);
    return 0;
  }
  TimerFreq=1193181L/cnt;
  Set8254Counter(cnt);
  SvInt08=getvect(8); setvect(8,NewInt08);
  atexit(DeactivateTimer);
  return 1;
}
void SetTimerFreq(unsigned freq)
{ SetTimer((unsigned) (1193181L/freq)); }
void DeactivateTimer(void)
{ SetTimer(0); }

```

Для инициализации таймера нужно из `main()` вызвать либо `SetTimer()` с граничным значением в параметре, либо (что лучше использовать при очень большой частоте таймера, чтобы избежать ошибок округления), функцию `SetTimerFreq()` с параметром частоты таймера в Гц.

### 1.5. Схема программируемого таймера КР580ВИ53





## **1.6. Порядок выполнения работы**

1. Изучить теоретическую часть.
2. Выполнить программирование таймера КР580ВИ53.
4. Выполнить следующее индивидуальное задание: разработать аппаратную и программную часть таймера, подключенного к шине с частотой 1300. Таймер считает 1/100 долю секунды. При изменении частоты ошибка составляет 1/1000 секунды. Тестирование провести на анимационной модели.

*Пояснение для выполнения индивидуального задания.*

Одним из примеров приборов, которые допускают перепрограммирование в соответствии с требованиями различных приложений, является программируемый интервальный таймер типа Intel 8253. Он содержит три независимых 16-разрядных счетчика, выполняющих счет в обратном направлении. Каждый из счетчиков имеет свои собственные линии синхросигнала, выхода (заем) и стробирования (разрешения). Предварительная установка позволяет задавать двоичный или двоично-десятичный алгоритм счета, причем каждый из счетчиков может работать в одном из шести различных режимов:

Режим 0. По достижению таймером предельного показания на его выходе устанавливается низкий уровень. Подача низкого уровня на вход стробирования останавливает процесс счета, а высокого уровня – разрешает его.

Режим 1. Это режим программируемого ждущего мультивибратора. На выходе низкий уровень поддерживается в течение всего времени, пока идет процесс счета синхроимпульсов. Фронт импульса на выводе стробирования может, как разрешать, так и осуществлять сброс счетчика после поступления последнего синхроимпульса.

Режим 2. Данный режим позволяет получить последовательность импульсов. Выход характеризуется высоким уровнем в течение процесса счета синхроимпульсов и приобретает низкий уровень на один синхропериод после достижения предельного показания. Процесс повторяется многократно, пока на выводе стробирования существует высокий уровень. Низкий уровень на выводе стробирования блокирует счет и вызывает появление на выходе высокого уровня, если он там не устанавливался ранее.

Режим 3. Этот режим сходен с режимом 2, но при его использовании на выходе формируется волна прямоугольной формы.

Режим 4. Выход в этом режиме, как правило, имеет высокий уровень; низкий уровень на выходе действует в течение одного синхропериода после достижения предельного показания. Низкий уровень на выводе стробирования блокирует процесс счета, а высокий - разрешает счет.

Режим 5. Выход здесь такой же, как и в режиме 4, и отличается от последнего лишь тем, что разрешение на счет дается по фронту импульса стробирования.

3. Оформить отчет и защитить работу.

## **1.7. Содержание отчета**

1. Цель работы.
2. Теоретическая часть.

3. Описание порядка выполнения работы и полученные результаты.

### ***1.8. Контрольные вопросы***

1. Назначение таймеров?
2. Какие типы таймеров Вы знаете? В чем их преимущества и недостатки, отличительные параметры?
3. Какие языки программирования можно использовать для программирования таймеров?

## **ЛАБОРАТОРНАЯ РАБОТА № 2**

### **АНАЛОГО-ЦИФРОВЫЕ И ЦИФРО-АНАЛОГОВЫЕ ПРЕОБРАЗОВАТЕЛИ**

#### ***1.1. Цель работы***

Спроектировать аналогово-цифровой преобразователь (АЦП) и цифро-аналоговый преобразователь (ЦАП) и запрограммировать процесс передачи данных из АЦП в ЦАП.

#### ***1.2. Исходные данные***

Входной сигнал — затухающие колебания на определенном интервале (фиксированные значения). Преобразование аналоговой величины в цифровую форму (таблица; 2 схемы: АЦП и ЦАП; выходной график — программа преобразования).

#### ***1.3. Краткие теоретические сведения***

Преимущества обработки информации и осуществления функций управления с использованием цифровых методов становятся все более очевидными. Однако данные, которые мы получаем из реального мира, обычно представлены в аналоговой форме. Необходимый аналого-цифровой интерфейс обеспечивает система сбора данных. Она преобразует исходные данные от одного или нескольких измерительных преобразователей в выходной сигнал, пригодный для цифровой обработки; преобразование осуществляется с помощью таких компонентов, как усилители, фильтры, схемы выборки – хранения, мультиплексоры и аналого-цифровые преобразователи (АЦП).

**Принцип работы АЦП.** Аналого-цифровое преобразование по существу является операцией, устанавливающей отношение двух величин. Входной аналоговый сигнал  $v_i$  преобразуется в дробь  $x$  путем сопоставления его значения с уровнем опорного сигнала  $V_r$ . Цифровой сигнал преобразователя есть кодовое представление этой дроби. Если выходной код преобразователя

является  $n$ -разрядным, то число дискретных выходных уровней равно  $2^n$ . Для взаимно-однозначного соответствия диапазон изменения входного сигнала должен быть разбит на такое же число уровней. Каждый квант (величина интервала) такого разбиения представляет собой значение аналоговой величины, на которое отличаются уровни входного сигнала, представляемые двумя соседними кодовыми комбинациями. Этот квант называют также величиной младшего значащего разряда (МЗР). Таким образом,

$$Q = \text{МЗР} = \text{ПД} / 2^n,$$

где  $Q$  – квант, МЗР – аналоговый эквивалент младшего значащего разряда, ПД – полный диапазон изменения входного аналогового сигнала.

Все аналоговые величины внутри заданного интервала разбиения представляются одним и тем же цифровым кодом, которому обычно ставят в соответствие значение аналоговой переменной в средней точке интервала, называемое пороговым уровнем. Тот факт, что входной сигнал может отличаться от порогового уровня на величину, достигающую  $\pm \frac{1}{2}$  МЗР, не отличается при этом по кодовому представлению, означает, что любому процессу аналого-цифрового преобразования присуща неопределенность (погрешность) дискретизации, равная  $\pm \frac{1}{2}$  МЗР. Ее влияние можно только уменьшить, увеличивая число разрядов в выходном коде преобразователя.

**Погрешности преобразователя.** Характеристики реальных преобразователей по ряду параметров могут отличаться от идеальных характеристик (аналогичных идеальной характеристике). Передаточная характеристика преобразователя может быть сдвинута относительно идеальной характеристики. Эта погрешность «смещения» или «установки нуля» определяется как значение аналоговой величины, при которой характеристика пересекает ось входных напряжений.

В АЦП проявляются два типа нелинейности – интегральная и дифференциальная. Интегральная нелинейность определяется максимальным отклонением передаточной характеристики от идеальной прямолинейной характеристики при нулевых значениях погрешностей смещения и усиления. Дифференциальная нелинейность – это отклонение величины одного из квантов от его идеального аналогового значения.

**Разрешение преобразователя.** Этот наиболее важный параметр преобразователя определяется как минимальная величина изменения аналогового напряжения на входе АЦП, вызывающая изменение выходного кода на один МЗР. Значение этого параметра обычно указывается в расчете на идеальный преобразователь и поэтому скорее отражает возможности преобразователя, чем его реальные рабочие характеристики. Разрешение может задаваться в процентах от полного диапазона или числом разрядов преобразователя.

Сигналы АЦ преобразователей бывают:

- аналоговый входной;

- аналоговый опорный;
- выходной;
- управляющий.

Аналого-цифровые преобразователи (АЦП) являются устройствами, которые принимают входные аналоговые сигналы и генерируют соответствующие им цифровые сигналы, пригодные для обработки микропроцессорами и другими цифровыми устройствами.

Принципиально не исключена возможность непосредственного преобразования различных физических величин в цифровую форму, однако эту задачу удастся решить лишь в редких случаях из-за сложности таких преобразователей. Поэтому в настоящее время наиболее рациональным признается способ преобразования различных по физической природе величин сначала в функционально связанные с ними электрические, а затем уже с помощью преобразователей напряжение-код — в цифровые. Именно эти преобразователи имеют обычно в виду, когда говорят об АЦП.

Процедура аналого-цифрового преобразования непрерывных сигналов, которую реализуют с помощью АЦП, представляет собой преобразование непрерывной функции времени  $U(t)$ , описывающей исходный сигнал, в последовательность чисел  $\{U'(t_j)\}, j = 0, 1, 2, \dots$ , отнесенных к некоторым фиксированным моментам времени. Эту процедуру можно разделить на две самостоятельные операции. Первая из них называется дискретизацией и состоит в преобразовании непрерывной функции времени  $U(t)$  в непрерывную последовательность  $\{U(t_j)\}$ . Вторая называется квантованием и состоит в преобразовании непрерывной последовательности в дискретную  $\{U'(t_j)\}$ .

Цифро-аналоговый преобразователь (ЦАП) предназначен для преобразования числа, определенного, как правило, в виде двоичного кода, в напряжение или ток, пропорциональные значению цифрового кода.

При выполнении работы необходимо получить следующие результаты (рис. 1...5).

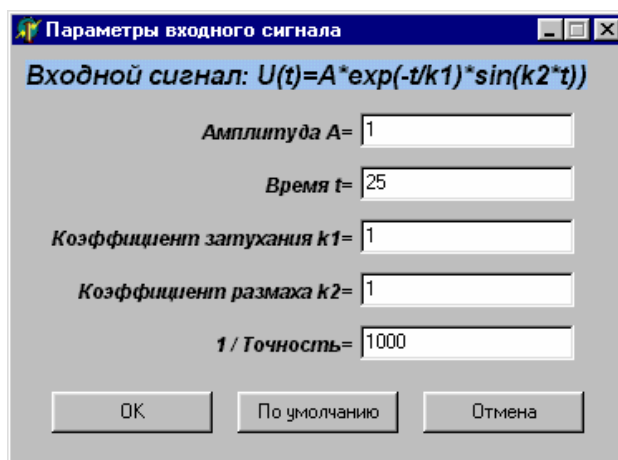


Рисунок 1 – Результаты работы программы

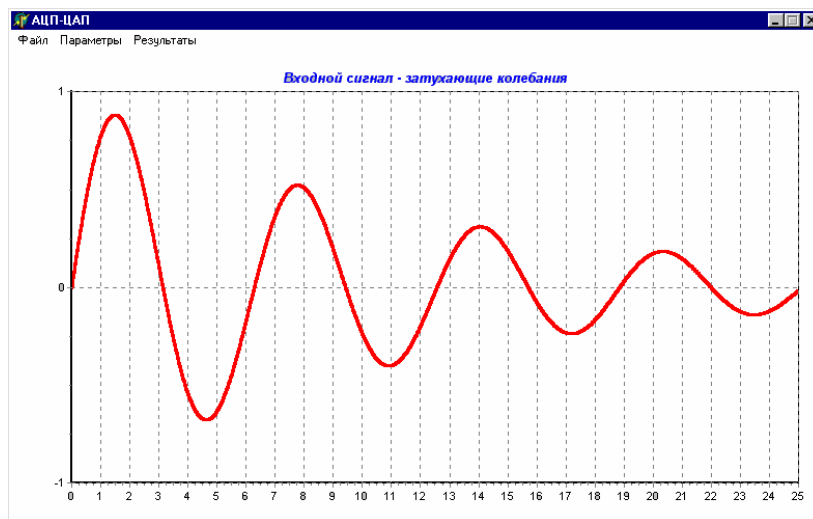


Рисунок 2 – График входного сигнала

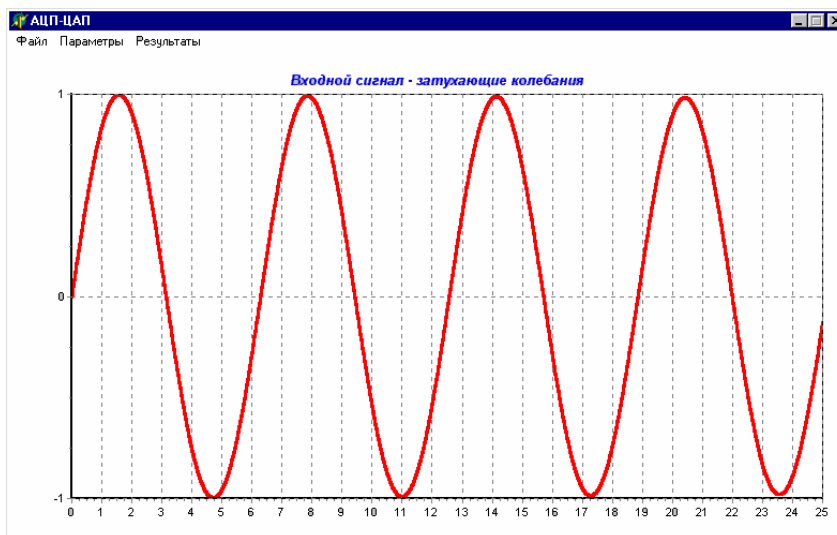


Рисунок 3 – График выходного сигнала в конце процесса моделирования

Рисунок 4 – Представление полученных результатов в табличном виде

| Результаты |             | Результаты |             | Результаты |             |
|------------|-------------|------------|-------------|------------|-------------|
| Время t    | Сигнал U(t) | Время t    | Сигнал U(t) | Время t    | Сигнал U(t) |
| 0.45       | 0.435       | 9          | 0.409       | 17.55      | -0.949      |
| 0.9        | 0.783       | 9.45       | -0.025      | 18         | -0.74       |
| 1.35       | 0.975       | 9.9        | -0.454      | 18.45      | -0.383      |
| 1.8        | 0.972       | 10.35      | -0.792      | 18.9       | 0.05        |
| 2.25       | 0.777       | 10.8       | -0.972      | 19.35      | 0.472       |
| 2.7        | 0.426       | 11.25      | -0.959      | 19.8       | 0.8         |
| 3.15       | -0.008      | 11.7       | -0.755      | 20.25      | 0.969       |
| 3.6        | -0.441      | 12.15      | -0.4        | 20.7       | 0.945       |
| 4.05       | -0.786      | 12.6       | 0.033       | 21.15      | 0.732       |
| 4.5        | -0.974      | 13.05      | 0.46        | 21.6       | 0.374       |
| 4.95       | -0.968      | 13.5       | 0.795       | 22.05      | -0.058      |
| 5.4        | -0.769      | 13.95      | 0.971       | 22.5       | -0.478      |
| 5.85       | -0.418      | 14.4       | 0.954       | 22.95      | -0.803      |
| 6.3        | 0.017       | 14.85      | 0.747       | 23.4       | -0.968      |
| 6.75       | 0.448       | 15.3       | 0.392       | 23.85      | -0.94       |
| 7.2        | 0.789       | 15.75      | -0.041      | 24.3       | -0.725      |
| 7.65       | 0.973       | 16.2       | -0.466      | 24.75      | -0.366      |
| 8.1        | 0.963       | 16.65      | -0.798      |            |             |
| 8.55       | 0.762       | 17.1       | -0.97       |            |             |

#### ***1.4. Порядок выполнения работы***

1. Изучить теоретическую часть.
2. Спроектировать аналогово-цифровой преобразователь (АЦП) и цифро-аналоговый преобразователь (ЦАП) с учетом исходных данных.
3. Осуществить программирование процесс передачи данных из АЦП в ЦАП при входном сигнале — затухающие колебания на определенном интервале (фиксированные значения).
4. Реализовать преобразование аналоговой величины в цифровую форму.

#### ***1.5. Содержание отчета***

1. Цель работы.
2. Теоретическая часть.
3. Описание порядка выполнения работы и полученные результаты.

#### ***1.6. Контрольные вопросы***

1. Обоснуйте выбор языка программирования.
2. Какие функции выбранного Вами языка программирования Вы использовали при реализации задачи?
3. Какие иные языки программирования можно было использовать для решения данной задачи?

## ЛАБОРАТОРНАЯ РАБОТА № 3

### ДАТЧИКИ ТЕМПЕРАТУР

#### 1.1. Цель работы

Изучить принцип работы, научиться калибровать и использовать транзисторный термометр.

#### 1.2. Исходные данные

Разработать программу для калибровки и использования транзисторного термометра. Диапазон изменения температуры: минус 20... +40 °С.

#### 1.3. Краткие теоретические сведения

Транзистор является хорошим датчиком температуры. При фиксированном токе коллектора напряжение база – эмиттер транзистора линейным образом зависит от температуры, т.е. так же, как и прямое напряжение на диоде. На рис. 1 представлена схема термометра с использованием в качестве датчика транзистора MTS105.

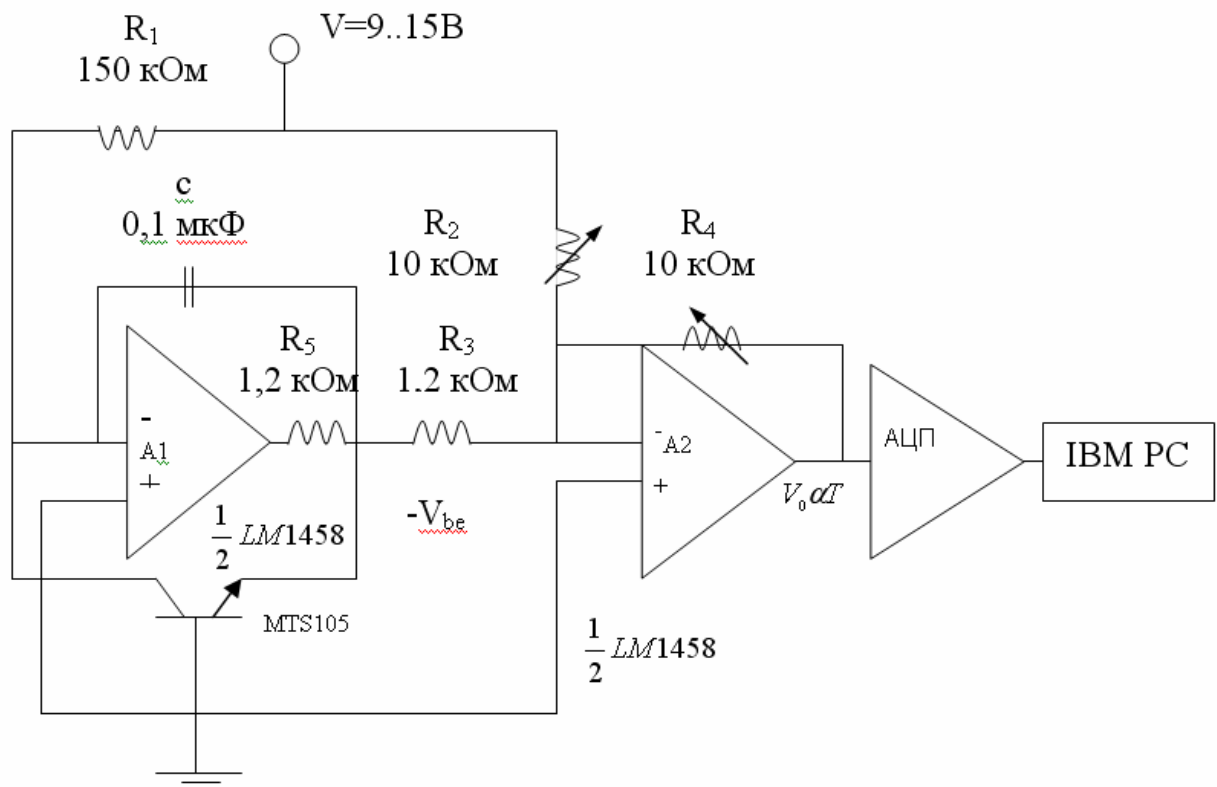


Рисунок 1 – Схема термометра с использованием в качестве датчика транзистора MTS105

Резистор  $R_1$  определяет коллекторный ток транзисторного датчика. Этот резистор должен обладать высокой стабильностью и низким темпера-

турным коэффициентом сопротивления. Операционные усилители  $A_1$  и  $A_2$  должны иметь малый дрейф. На выходе ОУ  $A_1$  будет действовать напряжение  $-V_{BE}$ . Это напряжение усиливается ОУ  $A_2$ . С помощью потенциометра  $R_4$  регулируется коэффициент усиления этого ОУ для поддержания выходного сигнала в пределах границ насыщения. Если в схему ввести потенциометр  $R_2$ , то выходной сигнал можно будет откалибровать к  $v_0 = 0$  при  $T = 0^\circ\text{C}$ . Если же калибровка осуществляется программно, то этот потенциометр не нужен. Элементы  $R_5$  и  $C$  предотвращают самовозбуждение схемы. Напряжение питания  $+V$  должно быть очень стабильным.

Для калибровки транзисторный датчик погружается в ледяную ванну, и с помощью потенциометра  $R_2$  устанавливается напряжение  $v_0 = 0$ , что соответствует индикации температуры в градусах Цельсия. Точность этого термометра  $\pm 0,01^\circ\text{C}$  в интервале температур от  $-50$  до  $+125^\circ\text{C}$ . Потенциометр  $R_4$  – регулятор усиления схемы (величины выходного напряжения). Калибровка в тройной точке воды и использование прецизионной схемы обеспечивают точность  $\pm 0,01^\circ\text{C}$  с учетом нелинейности и долговременной нестабильности.

#### ***1.4. Порядок выполнения работы***

1. Изучить теоретическую часть.
2. Разработать программу для калибровки и использования транзисторного термометра.

#### ***1.5. Содержание отчета***

1. Цель работы.
2. Теоретическая часть.
3. Описание порядка выполнения работы и полученные результаты.

#### ***1.6. Контрольные вопросы***

1. Обоснуйте выбор языка программирования.
2. Какие функции выбранного Вами языка программирования Вы использовали при реализации задачи?
3. Какие иные языки программирования можно было использовать для решения данной задачи?



## ЛАБОРАТОРНАЯ РАБОТА № 4

### ОПТИЧЕСКИЕ ДАТЧИКИ

#### *1.1. Цель работы*

Приобрести практические навыки по разработке программ для измерения освещенности с использованием оптического датчика.

#### *1.2. Исходные данные*

Разработать программу для измерителя освещенности с использованием оптического датчика.

#### *1.3. Краткие теоретические сведения*

*Измерение характеристик излучения.* Источники света имеют различное пространственное распределение генерируемого ими излучения. Одни излучают по всем направлениям, излучение других сосредоточено в узком конусе (пучке). Для описания пространственных и энергетических характеристик излучения источников света нужна некоторая система единиц. Как правило, используют стандартные фотометрические (относящиеся только к видимому диапазону длин волн) и радиометрические (относящиеся ко всему спектру) единицы системы СИ. Переход от «излучательных» единиц к «световым» осуществляется с помощью весовых коэффициентов, которые находятся по кривой спектральной чувствительности «идеального человеческого глаза».

*Оптические датчики.* Оптические датчики принято разбивать на 2 класса:

- детекторы фотонов (фотодетекторы);
- приемники теплового излучения (ИК - датчики).

Выходной сигнал детектора фотонов определяется эффектами взаимодействия фотонов с материалом детектора. Эти детекторы имеют малую постоянную времени (порядка нескольких микросекунд или меньше), более узкую характеристику спектральной чувствительности по сравнению с приемниками теплового излучения и более высокую, чем у последних, удельную детектирующую способность. Детекторы фотонов чувствительны к фотонам с энергиями, превышающими характерную внутреннюю энергию или ширину энергетической щели для материала детектора; их чувствительность в расчете на один ватт падающего излучения линейно возрастает с увеличением длины волны вплоть до критической пороговой длины волны.

В отличие от детекторов фотонов приемники теплового излучения в идеальном случае имеют плоскую характеристику спектральной чувствительности и воспринимают падающую на них энергию. Заметим, однако, что фильтрующие свойства материалов окон могут существенно повлиять на вид этой характеристики. Приемники теплового излучения характеризуются постоянной времени порядка нескольких миллисекунд и применяются главным образом в ИК–диапазоне. Они часто используются вместе с оптическими мо-

дулями с целью коррекции дрейфа нуля. Чувствительность приемника теплового излучения, помещенного в откаченную камеру, улучшается в 5-10 раз по сравнению с его чувствительностью на воздухе, причем в таких условиях меньше проявляются различные свойства возмущения, связанные, например, с движением воздуха.

*Параметры оптических датчиков:*

- функция спектральной чувствительности;
- постоянная времени;
- поле обзора;
- рассеиваемая мощность;
- интегральная чувствительность или просто чувствительность;
- эквивалентная мощность шума;
- удельная детектирующая способность или удельная способность по обнаружению излучения:

$$D^* = \frac{(A\Delta f)^{\frac{1}{2}}}{P_N} \left( \text{см} \cdot \text{Гц}^{\frac{1}{2}} \cdot \text{Вт}^{-1} \right),$$

где  $A$  – площадь детектора и  $\Delta f$  – ширина полосы пропускания усилителя в измерительном тракте. Этот параметр измеряется на определенной длине волны в предположении полусферического поля обзора (поля сбора излучения);

- квантовый выход – число фотоэлектронов, возникающих при поглощении каждого падающего фотона;
- темновой ток – остаточный ток, протекающий в отсутствие падающего излучения.

#### ***1.4. Порядок выполнения работы***

1. Изучить теоретическую часть.
2. Разработать программу для измерителя освещенности с использованием оптического датчика.

#### ***1.5. Содержание отчета***

1. Цель работы.
2. Теоретическая часть.
3. Описание порядка выполнения работы и полученные результаты.

#### ***1.6. Контрольные вопросы***

1. Обоснуйте выбор языка программирования.
2. Какие функции выбранного Вами языка программирования Вы использовали при реализации задачи?
3. Какие иные языки программирования можно было использовать для решения данной задачи?

## ЛАБОРАТОРНАЯ РАБОТА № 5

### ДАТЧИКИ ПЕРЕМЕЩЕНИЯ

#### 1.1. Цель работы

Приобрести практические навыки по разработке программ для считывания информации от контактной ручки управления с использованием датчиков перемещения.

#### 1.2. Исходные данные

Разработать программу для управления положением курсора и считывания информации от контактной ручки управления.

#### 1.3. Краткие теоретические сведения

Измерительные преобразования перемещения широко используются во многих областях техники, особенно в промышленности. Приведем примеры типичных преобразователей перемещения:

- тензодатчики (резистивные, свободные, приклеиваемые, полупроводниковые, эластичные резистивные, мостовые схемы);
- линейные дифференциальные преобразователи (трансформаторы);
- пьезоэлектрические;

*Усиление заряда.* Более распространенным методом обработки сигнала от пьезоэлектрического преобразователя является метод усиления заряда. Можно сказать (при некоторых ограничениях), что при использовании усилителя заряда АЧХ системы преобразователь – усилитель на низких частотах определяется только компонентами  $R$  и  $C$  цепи обратной связи усилителя, а полная чувствительность системы не зависит от емкостей преобразователя и кабеля. В схеме, изображенной на рис. 1, инвертирующей вход ОУ на полевых транзисторах – виртуальная земля.

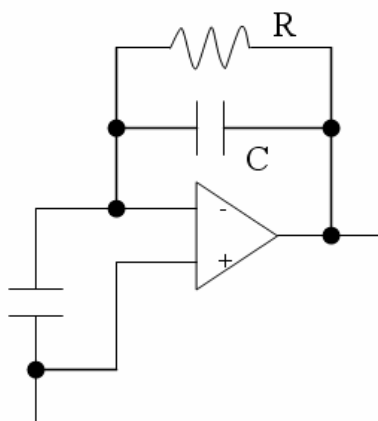


Рисунок 1 – Усилитель заряда

Это означает, что напряжение на сопротивлении утечки и суммарной емкости преобразователя и кабеля по существу равно нулю. Таким образом,

можно использовать длинные соединенные кабели без какого-либо влияния на чувствительность или АЧХ системы. Весь генерируемый заряд накапливается на конденсаторе обратной связи, а выходное напряжение схемы равно взятому с обратным знаком напряжению на этом конденсаторе.

$$v_0 = -v = \frac{-1}{C} \int_0^t \left( \frac{Kdx}{dt} \right) dt = \frac{-Kx}{C}$$

Может показаться, что  $v_0$  пропорционально  $x$  даже на нулевой частоте (при  $x=const$ ). На самом деле это не так. Параллельно конденсатору должен быть обязательно включен резистор с большим сопротивлением. В противном случае входной ток смещения ОУ будет вызывать временной дрейф выходного напряжения, и в конце концов ОУ войдет в насыщение. Однако добавление этого резистора превращает схему в активный фильтр верхних частот с постоянной времени  $RC$ . Нижняя граница частоты такого фильтра  $f = \frac{1}{2\pi RC}$ , так что схема не работает при нулевой частоте.

Заметим, что уменьшение  $C$  приводит к повышению чувствительности, но при этом также возрастает нижняя граничная частота. Кроме того, при увеличении входной емкости возрастает шум усиления заряда, хотя это касается низкоуровневых и низкочастотных измерений. Таким образом, налицо взаимосвязь чувствительности и АЧХ системы.

В лабораторной работе необходимо исследовать перемещение датчиков на примере распознавания перемещения карандаша (мыши) на экране. Таблица распознавания кривых представлена на рис. 2.

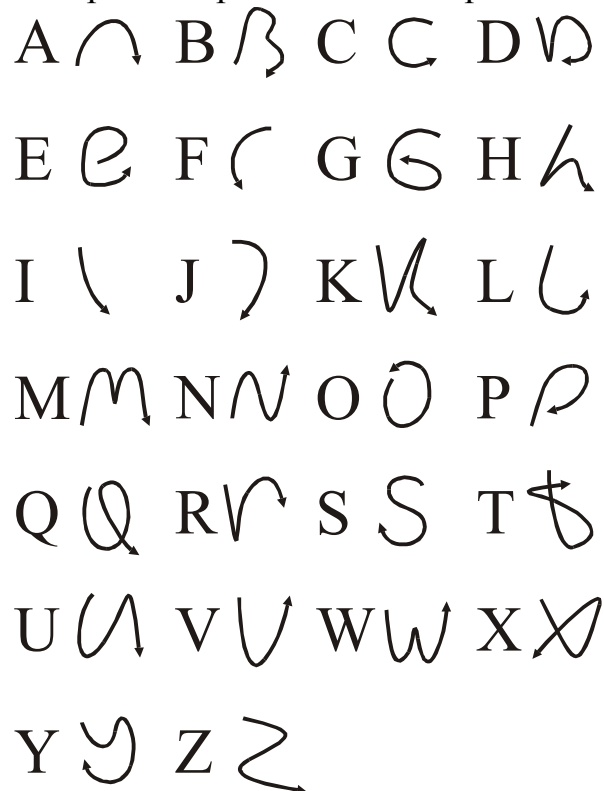


Рисунок 2 – Таблица распознавания кривых

#### ***1.4. Порядок выполнения работы***

1. Изучить теоретическую часть.
2. Разработать программу для управления положением курсора и считывания информации от контактной ручки управления.

#### ***1.5. Содержание отчета***

1. Цель работы.
2. Теоретическая часть.
3. Описание порядка выполнения работы и полученные результаты.

#### ***1.6. Контрольные вопросы***

1. Обоснуйте выбор языка программирования.
2. Какие функции выбранного Вами языка программирования Вы использовали при реализации задачи?
3. Какие иные языки программирования можно было использовать для решения данной задачи?