

1. Структура и функционирование измерительно-вычислительной системы

Измерительно-вычислительная система (ИВС) представляет собой совокупность аппаратных и программных средств для автоматизации проведения физического эксперимента. Под автоматизацией подразумевается сбор, хранение и накопление информации, а также управление экспериментом и процессом измерения посредством единого (стандартизованного) средства связи в системе исследователь - физический процесс (объект). В качестве этого средства обычно используется *компьютер*. Подобная автоматизация предусматривает контроль над ведением эксперимента полностью программными методами, являющимися гибким инструментом в руках исследователя, который управляет ходом измерений, задавая только определённые команды компьютеру и получает результаты на экране дисплея или в распечатанном виде. Рассмотрим рис.1.1, на котором приведена структурная организация простейшей ИВС. Измерение требуемого параметра *физического объекта* осуществляется с помощью соответствующего *датчика* (например, температура измеряется с помощью термопары), при этом полученная величина обычно изменяется непрерывным образом в некотором диапазоне, т.е. сигнал датчика имеет аналоговую форму. Для последующей обработки сигнала его необходимо преобразовать в цифровую форму. Эта процедура осуществляется *аналого-цифровым преобразователем* (АЦП). Поскольку АЦП оперирует сигналами стандартной амплитуды (обычно $\pm 1В$), в то время, как амплитуда сигнала датчика может существенно изменяться в процессе измерений, необходимо использовать *согласующее устройство* (например, операционный усилитель), обеспечивающее приведение амплитуды сигнала к стандартизованной величине. Сигнал в цифровом виде поступает на вход *компьютера* (ПК).

Наряду с измерением некоторой физической величины, иногда необходимо внешнее воздействие на физический объект, например, изменить положение датчика параметра относительно объекта, или напрямую воздействовать на физический объект или окружающую его среду. Для реализации этой функции служит нижняя ветвь блок-схемы, изображённой на рис.1.1. Для управления *исполнительным устройством* (реле, шаговый двигатель, управляемый усилитель и т.п.) нужен аналоговый сигнал, в то время как от ПК можно получить только цифровой. Для преобразования из цифрового представления в аналоговое служит *цифро-аналоговый преобразователь* ЦАП. Он осуществляет операцию, противоположную действию АЦП. Аналоговый сигнал с выхода ЦАП поступает через согласующее устройство 2 на исполнительное, которое воздействует, например, на физический объект или датчик.

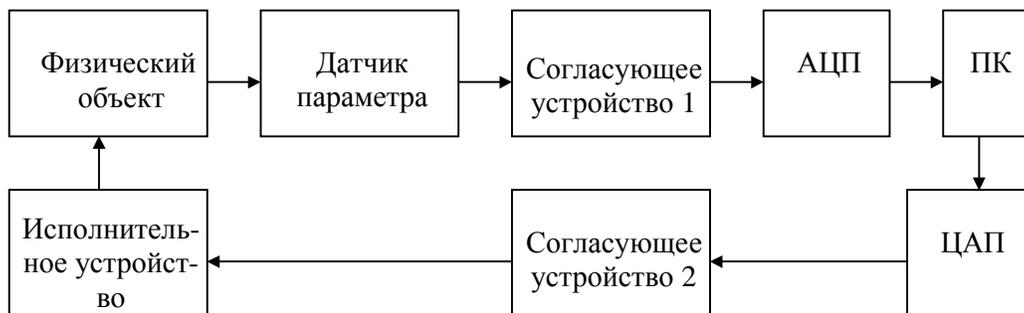


Рис. 1.1. Схема ИВС

1.1 Принцип преобразования аналоговой информации в цифровую

Основной принцип, положенный в основу работы АЦП - это переход от непрерывных величин к дискретным. В АЦП исходными непрерывными величинами, которые требуют преобразования, являются напряжение (ток) и время, поскольку любой аналоговый сигнал можно изобразить в виде непрерывного процесса во времени, т.е. как функцию $U(t)$, где U - напряжение; t - время. На выходе АЦП мы получаем набор дискретных значений кода (набор цифр) через определённые промежутки времени (рис.1.2). Весь диапазон допустимых входных напряжений АЦП квантуется на N фиксированных отсчётов, т.е. вся шкала напряжений делится на N равных частей и одна из этих частей называется шагом квантования Δh .

Например, при входном диапазоне напряжений от -1В до $+1\text{В}$ и при $N = 2^8$, шаг квантования $\Delta h = (|-1| + |1|)/2^8 \text{ В} = 0.0078 \text{ В}$. Таким образом, напряжение U на входе АЦП в какой-то момент времени может быть с точностью до значения Δh записано в виде:

$$U \approx A \cdot \Delta h,$$

где A - значение кода на выходе АЦП. Для оцифровки непрерывно изменяющегося сигнала преобразование входного напряжения идёт постоянно через определённые промежутки времени Δt (рис.1.2). Время Δt необходимо АЦП для преобразования и называется временем дискретизации. От значения Δt зависит максимально возможная частота (скорость изменения) входного сигнала для корректного преобразования. Время дискретизации зависит от внутреннего устройства АЦП и может быть от единиц наносекунд до единиц секунд.

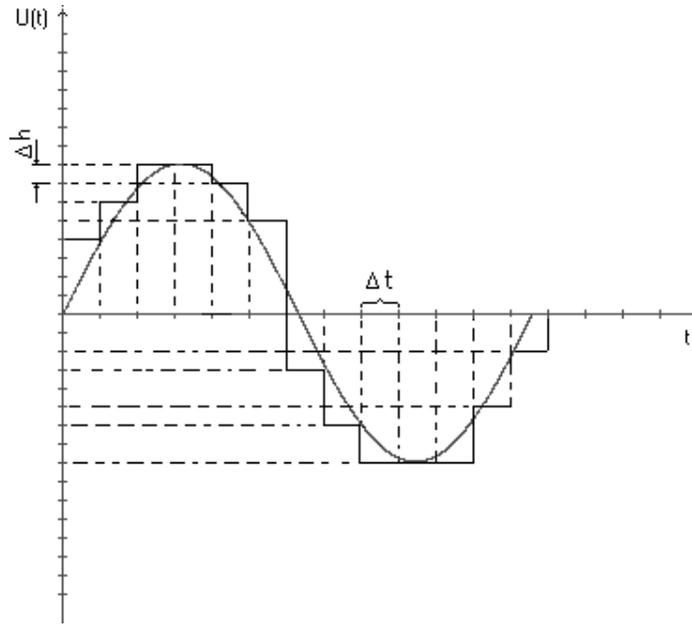


Рис.1.2. Иллюстрация принципа аналогово-цифрового преобразования.

Из рис. 1.2 следует, что исходный сигнал, изображаемый плавной кривой $U(t)$ после преобразования АЦП представляется ступенчатой линией, при этом, очевидно теряется часть информации о сигнале, поэтому ставится вопрос только о том, сколько информации допустимо потерять для сохранения достоверности результатов измерений. Потери обусловлены конечностью Δh и Δt (при Δh и $\Delta t \rightarrow 0$ переходим к непрерывному сигналу). Число N обычно лежит в диапазоне от 2^8 до 2^{16} и принимает фиксированные значения 2^8 , 2^{10} , 2^{12} , 2^{16} , причём диапазон входного напряжения может быть от -5В до $+5\text{В}$ или менее. При этом $\Delta h = 10/2^{10} \approx 1$ мВ. Необходимо учитывать, что диапазон входных напряжений АЦП строго фиксирован, и поэтому наименьшая погрешность при преобразовании будет у сигнала, близкого по амплитуде к верхней границе входного диапазона.

Для минимизации ошибок, связанных с дискретизацией во времени, необходимо, чтобы частота преобразования (число отсчетов в единицу времени) была гораздо больше максимальной частоты аналогового сигнала. По крайней мере, максимальная частота сигнала должна быть не менее чем в два раза меньше частоты дискретизации. В этом случае ещё возможно (теоретически) полностью восстановить исходный сигнал.

1.2 Элементы ИВС

Конкретная реализация ИВС в каждом случае своя, однако, можно выделить ряд общих для каждой системы функциональных узлов и параметров. Конструктивно ИВС обычно состоит из трёх основных частей: компьютера; интерфейсного блока (ИБ), содержащего большую часть устройства ИВС; совокупности датчиков параметров и исполнительных устройств. Состав последней части напрямую зависит от характера измерений и объекта исследования, т.е. от конкретной задачи. Датчиками параметра могут быть фотодиоды, фотоумножители, ПЗС матрицы, термопары, динамические головки и т.д. В качестве исполнительных устройств могут использоваться электромагниты, различные двигатели, реле, электронные ключи, нагреватели и т.д. Выбор компьютера также в какой-то мере зависит от эксперимента, но круг задач, решаемых при помощи одного типа компьютера, например IBM PC/AT, весьма широк и в большинстве случаев подходит любой из современных ПК. Интерфейсный блок является наиболее важной частью ИВС, т.к. осуществляет преобразование информации и исполнение команд при передаче в направлении объект → компьютер и компьютер → объект.

Интерфейсный блок обычно содержит АЦП, ЦАП и согласующие устройства (СУ). В отдельных ИБ может отсутствовать ЦАП или АЦП с соответствующими СУ, однако может случиться и так, что их окажется несколько. При дальнейшем описании будем опираться на модель ИБ с АЦП и ЦАП.

Основное назначение ИВС - автоматизация проведения эксперимента. При этом управление и контроль должны осуществляться программными средствами. Следует отметить, что ИБ является для компьютера внешним устройством (ВУ), поэтому и обращение с ним идёт как с периферийным узлом. ИБ содержит в своём составе совокупность регистров (ячеек) памяти, при обращении к которым и осуществляется весь обмен информацией между ИБ и компьютером. Обычно выбор нужного регистра осуществляется посредством взаимодействия с *портами* ввода-вывода.

Порты ввода-вывода

Порт представляет собой внешнее устройство, адресуемое микропроцессором. В ИВС порт представляет собой ячейку памяти (регистр) к которой можно обратиться, указав её адрес. Однако, в отличие от обычной памяти, в большинстве случаев эта ячейка может работать только на запись или только на считывание, т.е. обмен данными с ней однонаправлен. Под считыванием здесь понимается процесс, при котором данные от

внешнего устройства поступают по *шине данных* в процессор; запись - обратная операция - из процессора во внешнее устройство. Необходимо отметить, что по одному и тому же адресу порта могут находиться две независимые (раздельные!) ячейки, в одну из которых можно записывать, а из другой считывать, причем независимость их имеет принципиальное значение. Например, одна ячейка может содержать данные о состоянии системы (на считывание), а при записи во вторую - можно управлять движением шагового двигателя. Следует отметить, что существуют ИВС с двунаправленными портами, однако в данных лабораторных работах такая реализация не используется.

Процедура обращения к порту

Для того, чтобы обратиться к порту, необходимо знать его адрес, назначение и, в соответствие с последним, направление передачи данных при работе.

В данных лабораторных работах при написании программ используется пакет Delphi. В состав стандартных средств программирования среды Delphi, возможность прямого обращения к портам ввода-вывода не входит. Ввиду этого, для работы с портами используется специально написанный библиотечный модуль **ports**. Он содержит две процедуры, написанные на машинно-ориентированном языке программирования Ассемблер. Одна из этих процедур – **outport(address,data)** - предназначена для вывода (записи) заданного значения в порт с заданным адресом. При её вызове в порт с адресом **address** записывается значение переменной **data**. Другая процедура – **inport(address,data)** служит для ввода (считывания) данных из порта. В процессе выполнения эта процедура считывает данные в переменную **data** из порта с адресом **address**. Тип переменной, используемой в качестве **address** должен быть **word**, тип **data** – **byte**. Ниже приведено описание модуля **ports**.

```
unit ports;  
interface  
procedure outport(address :word;data :byte);  
procedure inport(address :word;var data :byte);
```

implementation

```
procedure outport(address :word;data :byte);  
begin  
  asm  
    mov dx,address
```

```

    mov al,data
    out dx,al
end
end;

procedure inport(address :word;var data :byte);
var regal : byte;
begin
    asm
        mov dx,address
        in al,dx
        mov regal,al
    end;
    data:=regal;
end;

end.

```

Этот модуль подключается к основной программе через меню Delphi. Для использования описанных процедур необходимо завести две переменные, например **a** и **b**, с типами **word** и **byte** соответственно. Значение первой использовать в качестве адреса; второй – для хранения вводимых/выводимых данных. Для вывода данных в порт необходимо: присвоить его адрес переменной **a**; установить **b**, равной записываемому значению; вызвать процедуру **outport(a,b)**. Для считывания данных из порта: присвоить адрес порта переменной **a**; вызвать процедуру **inport(a,b)**; в качестве данных использовать значение переменной **b** (процедура устанавливает её значение соответствующим образом). Переменные типа **word** могут принимать значения 0..65535; типа **byte** – 0..255. Необходимо иметь ввиду, что в используемых ИВК все порты однонаправлены, т.е. если записать, скажем, число **value** в порт с адресом **adr**, а затем считать из порта **adr**, то мы не получим исходного **value**.

Для последующей математической обработки результатов измерений необходимо полученные данные записать в текстовый файл. Ниже приведен фрагмент программы, обеспечивающий эту операцию.

{ переменные, задаваемые в разделе описания переменных соответствующей процедуры:

```

const
    k=500;    количество элементов выборки
var
    f: text;  текстовая файловая переменная}

```

.....

```

assign(f,'path_to_file\file_name');    {связать внешний файл file_name с
файловой переменной f }
rewrite(f);                            { создать и открыть файл }
for i:=1 to k do
    writeln(f,x[i]);                    { запись массива данных x[i] столбцом в файл }
close(f);    { закрыть файл }
.....

```

Регистры, обслуживающие АЦП

При работе с одноканальным АЦП существуют как минимум две информационные линии обмена: сигнала готовности АЦП и непосредственно данных. Сигнал готовности несёт информацию о состоянии АЦП в каждый момент времени. Необходимость его анализа при работе АЦП очевидна, т.к. если просто постоянно считывать данные, то наверняка получим повторную информацию, т.е. возникает ситуация, когда данные ещё не обновились, а АЦП уже снова опрашивается. Проще говоря, компьютер в этом случае работает гораздо быстрее системы сбора данных (ССД) АЦП и необходимо сделать задержку, чтобы «подождать» АЦП (время обращения компьютера к ИБ меньше времени дискретизации АЦП). Если сигнал готовности имеет значение 1(0) , это означает, что преобразование завершено и АЦП готов к считыванию, если 0(1), - то не готов. Следовательно, для считывания из АЦП необходимо наличие двух выделенных совокупностей портов: порт, адресующий регистр(ы) данных и порт регистра состояния АЦП. Кроме описанных двух информационных линий, при работе с АЦП может также использоваться сигнал запуска АЦП. Он необходим в тех системах, где АЦП работает не постоянно, а в режиме “старт-стоп”. По пришествии данного сигнала начинается процесс единичного преобразования, по завершении которого АЦП переходит в состояние ожидания следующего такого сигнала.

При исследовании закона распределения какой-нибудь физической величины в пространстве, например измерение температуры стержня по его длине, необходимо одновременно вводить информацию с нескольких датчиков. Для реализации подобной системы применяют многоканальные ССД на основе АЦП. Они отличаются от одноканальных ССД только количеством аналоговых входов, при этом каждый из них последовательно подключается ко входу единственного АЦП. Сигнал готовности ССД обычно несёт информацию о том , с каким из каналов оперирует АЦП в данный момент. Дополнительно ССД требует указания номера активного канала , а также, в ряде случаев, алгоритма подключения аналоговых входов к АЦП.

Регистры, обслуживающие датчики, исполнительные устройства и СУ

Датчики параметров обычно не подлежат управлению напрямую. Однако возможен контроль и взаимодействие с СУ. В качестве СУ в большинстве случаев используется усилитель с регулируемым коэффициентом усиления, который устанавливается программным образом. Изменение чувствительности необходимо для приведения входного сигнала к значениям, лежащим в пределах входного диапазона АЦП, так чтобы АЦП использовался оптимальным образом. В отдельных реализациях ИВС диапазон изменения коэффициента усиления может достигать 3-4 порядков. Применительно к многоканальным системам следует отметить, что количество усилителей в таком случае будет равно числу каналов и число изменяемых коэффициентов соответственно возрастёт. Для управления чувствительностью усилителя обычно используют два порта. Записью в один из них активизируют выбранный канал; используя второй - устанавливают чувствительность усилителя.

Для управления исполнительными внешними устройствами также отводится некоторая совокупность адресов портов ввода-вывода. Однако, структура этих портов зависит от конкретного экземпляра ИВС. Например, для управления электромагнитным реле достаточно выделить всего один бит, указывающий, включено оно или выключено. Рассмотрим более сложный пример управления шаговым двигателем (ШД), используемого при проведении одной из лабораторных работ. Ось шагового двигателя вращается по шагам, т.е. имеет конечный (минимальный) угол поворота.

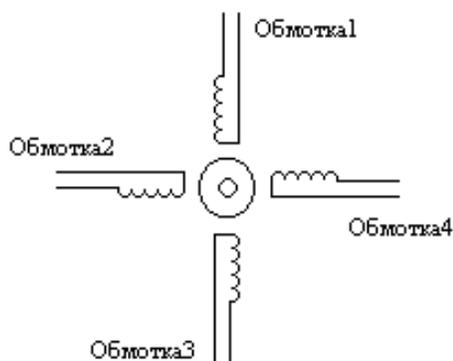


Рис.1.3. Схема шагового двигателя

Для того, чтобы двигатель сделал полный оборот необходимо сделать N шагов по $360/N$ градусов каждый. Следует отметить, что после каждого шага ось жёстко фиксируется и не имеет свободного вращения. Очевидно, что полную систему параметров для его управления составляет совокупность двух сигналов: направления и инициации шага (сигнал, по которому

делается один шаг). Алгоритм управления с использованием этих параметров таков: установил направление – сделал шаг и т.д.

Большинство шаговых двигателей можно условно представить как устройство с четырьмя (тремя) неподвижными обмотками, расположенными под прямым углом друг к другу, и подвижного якоря (см. рис.1.3). Каждая смежная пара обмоток может быть подключена к источнику питания, в то время как другая пара отключена.

Например, обмотки 1 и 2 подключены к источнику питания, а 3 и 4 – нет. Для того, чтобы повернуть ось двигателя на один шаг в заданную сторону, необходимо изменить порядок подключения обмоток. Повернём ось против часовой стрелки (по рис.3). Для этого необходимо сначала подключить обмотки 1 и 2 к источнику; 3 – 4 отключить; затем подать питание на обмотки 2-3 и отключить 1,4. Ось шагового двигателя сделает поворот на один шаг (минимальный угол) против часовой стрелки. Если обозначить подключённое состояние через 1, отключённое – через 0, и номеру обмотки сопоставлять положение 1 или 0 в записи вида xxxx, где каждое x принимает значения нуля или единицы (первое x слева направо – состояние 1 обмотки, второе x – второй и т.д.), то последовательность xxxx для вращения против часовой стрелки будет: 1100, 0110, 0011, 1001, 1100 ... Аналогичным образом можно организовать вращение по часовой стрелке. Последовательность подключения обмоток в этом случае будет, очевидно, следующей: 1100, 1001, 0011, 0110, 1100 ...

Из вышеприведённого примера следует, что для управления ШД требуется регистр порта на запись размером не менее 4 бит. Однако, если сделать преобразователь базисных параметров направления и инициации шага в управляющие последовательности xxxx, то потребуется только дву-битовый регистр.

2. Колебания шарика в вязкой среде

В работе исследуется зависимость от размера и скорости тела силы сопротивления, возникающей при колебательном движении сферического тела в воздухе. Эта задача является одной из простейших задач гидродинамики, иллюстрирующей ее основные положения. Исследование проводится сначала экспериментально, результаты измерений обрабатываются на компьютере с помощью программы, которую необходимо написать студенту. Затем на компьютере с помощью готовой программы исследуется математическая модель явления. Для физического анализа результатов измерений необходимо ознакомиться с разделом «Элементы гидродинамики».