Цифровой выход

Цифровая версия датчика имеет индекс (D) и отличается тем, что в конструкцию датчика включена миниатюрная плата аналого-цифрового преобразователя (АЦП), преобразующая с высокой точностью и в реальном масштабе времени выходную информацию датчика в цифровую форму.

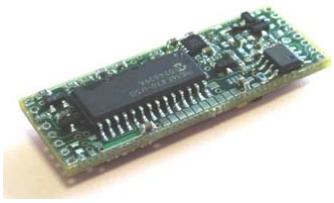
Аналого-цифровое преобразование заключается в преобразовании аналогового сигнала U(t) в последовательность дискретных значений U(iт), где т- период дискретности преобразования. Каждое значение U(iт) является средним значением сигнала U(t) на интервале т.

$$U(i\tau) = \frac{1}{\tau} \int_{(i-1)\tau}^{i\tau} U(t) \cdot dt$$

Цифровой выход позволяет транслировать выходные данные датчика современным вычислительным устройствам (ВУ) потребителя – персональным компьютерам, ноутбукам, бортовым вычислителям. В отличии от аналогового датчика цифровой датчик подключается к ВУ с помощью кабеля и не требует использования дополнительных преобразователей. Преобразователь выходных аналоговых данных датчика в цифровую форму и устройство трансляции цифровых данных включены в конструкцию датчика (плата АЦП).

Для использования цифрового датчика ВУ потребителя должно иметь порт RS232 или USB порт (эти порты имеются во всех современных ВУ). При использовании USB порта потребуется еще адаптер RS232/USB (USB-G)

Плата АЦП



Для преобразования выходных аналоговых данных датчика в цифровую форму разработана миниатюрная плата АЦП на базе 24-разрядного σ-Δ АЦП ADS1251 (Texas Instraments) и контролера на основе процессора PIC16F873. Плата выполняет следующие основные функции:

- преобразования выходного напряжения ВОГ - U(Ω) в последовательный 24-х разрядный код;
- преобразования в 10-и разрядный код аналоговых сигналов датчика температуры, датчика тока SLD, напряжения питания ВОГ, и контрольного сигнала (вспомогательные данные для коррекции и в технологических целях);
- передачи данных в ВУ потребителя в виде последовательного цифрового кода по стандарту RS232.

Стандарт RS232 обеспечивает надежную и помехозащищенную передачу цифровых данных. В стандарте RS232 вся информация передается в виде пакета (посылки) из нескольких последовательных байтов. Байт это восьмиразрядное двоичное число. Поэтому для передачи, например, 24 разрядного числа требуется три байта, а 10 разрядного – два байта. Алгоритм кодирования и расшифровки передаваемых данных определяется протоколом связи. Протокол связи также определяет:

- скорость передачи и параметры установки порта,
- частоту обновления передаваемых данных,
-методы контроля истинности принимаемой информации.

Расположение платы АЦП непосредственно в корпусе датчика позволило улучшить его характеристики, а именно:

- увеличить чувствительность за счет уменьшения паразитных наводок на аналоговые цепи,
- увеличить помехозащищенность и надежность передачи данных измерения угловой скорости,
- располагать датчик на достаточном удалении (до 3-х метров) от ВУ потребителя.

Протокол обмена (информация для программирования)

Параметры установки порта. **Восемь бит данных, один стоп бит, без контроля чётности**.

Скорость передачи данных по последовательному порту RS232 может составлять 115, 38 или 9 кбод. Скорость передачи данных для конкретного датчика устанавливается аппаратно и должна оговаривается при заказе датчика.

По умолчанию устанавливается скорость передачи 38 кбод. Передача данных о выходном напряжении датчика осуществляется контроллером АЦП в 24 разрядном формате (тремя байтами), а дополнительной информации в 16 разрядном формате (двумя байтами для каждой величины). За один сеанс связи по каналу RS232 передается посылка из восьми байт, включающих три байта данных точного канала и один байт данных одного из дополнительных данных, а также байт синхронизации, байт, содержащий номер байта грубого канала, и два байта контрольного слова.

Таблица 1 Структура данных ВОГ, передаваемых по интерфейсу RS232 .

Передаваемые Данные	Обозначение	Разряд ность	Соответствующий Аналоговый сигнал	Диапазон измененения	Цена младшего разряда кода
Нпапряжение на выходе ВОГ	UΩ	24	$U\Omega \cdot (2.5/2^{23})$ [В]	+2.5…-2.5	$2.5/2^{23}$ [В] (~0.298 мкВ)
Температура	T	16	$T\cdot (250/2^{15}-50)$ [°C]	+200…-300 °C	$250/2^{15}$ [°C]
Напряжение питания	U	16	$U\cdot (10/2^{15})$ [В]	+10…-10 В	$10/2^{15}$ [В]
Ток потребления	I	16	$I \cdot (0.25/2^{15})$ [А]	+0.25…-0.25 А	$0.25/2^{15}$ [А]
Контрольный сигнал	KS	16	$KS\cdot (2.5/2^{15})$ [В]	+2.5…-2.5 В	$2.5/2^{15}$ [В]
Резерв	R1	16	-	-	-
Резерв	R2	16	-	-	-
Резерв	R3	16	-	-	-
Резерв	R4	16	-	-	-

Данные резервных каналов R1…R4 передаются, но не используются.

Таблица 2. Структура данных в одном сеансе передачи (посылке).

Номер Байта	0	1	2	3	4	5	6	7
Передаваемые данные	Байт синхронизации (0xdd)	Напряжение на выходе ВОГ U(Ω), байты:			Счетчик посылок	Старший или младший Байт дополнительных данных	Контрольное Слово, байты:	
		младший	старший	средний			Старший	младший
Символ Байта	Sinx	UΩ(L)	UΩ(H)	UΩ(MDL)	Count	T,U,I,KS, R1...R4 (H/L)	CCL(H)	CCL(L)

Величина контрольного слова равна сумме значений байтов 1…5. Значение байта синхронизации 0xdd. Значение счетчика Count увеличивается на единицу в каждой следующей посылке.

Примечание:

В некоторых документах могут встречаться термины:

Rate – то же что и напряжение на выходе ВОГ. Счетчик медленных данных – то же, что и счетчик посылок.

Медленные данные – то же что и дополнительные данные.

Таблица 3. Полный цикл из 16 восьмибайтовых посылок, в которых 16 раз обновляются данные U(Ω) и один раз данные T, U, …R4.

Номер посылки	Байты							
	Sinx	UΩ (L)	UΩ (L)	UΩ (MDL)	Count	T,U,I,KS,R1...R4 (H/L)	CCL(H)	CCL(L)
0	0xdd	UΩ (L)	UΩ (H)	UΩ (MDL)	0x00	T(H)	CCL(H)	CCL(L)
1	0xdd	UΩ (L)	UΩ (H)	UΩ (MDL)	0x01	T(L)	CCL(H)	CCL(L)
2	0xdd	UΩ (L)	UΩ (H)	UΩ (MDL)	0x02	U(H)	CCL(H)	CCL(L)
3	0xdd	UΩ (L)	UΩ (H)	UΩ (MDL)	0x03	U(L)	CCL(H)	CCL(L)
4	0xdd	UΩ (L)	UΩ (H)	UΩ (MDL)	0x04	I(H)	CCL(H)	CCL(L)
5	0xdd	UΩ (L)	UΩ (H)	UΩ (MDL)	0x05	I(L)	CCL(H)	CCL(L)
6	0xdd	UΩ (L)	UΩ (H)	UΩ (MDL)	0x06	KS(H)	CCL(H)	CCL(L)
7	0xdd	UΩ (L)	UΩ (H)	UΩ (MDL)	0x07	KS(L)	CCL(H)	CCL(L)
…….								
14	0xdd	UΩ (L)	UΩ (H)	UΩ (MDL)	0x0E	R4(H)	CCL(H)	CCL(L)
15	0xdd	UΩ (L)	UΩ (H)	UΩ (MDL)	0x0F	R4(L)	CCL(H)	CCL(L)

Таким образом, частота обновления дополнительных данных в 16 раз меньше чем данных UΩ.

Скорость передачи данных по последовательному порту RS232 может составлять 115, 38 или 9 кбод:

– скорость передачи 38 кбод, частота обновления данных точного канала – 300 Гц, частота обновления дополнительных данных – 300/16 Гц;

– скорость передачи 115 кбод, частота обновления данных точного канала –1200 Гц, частота обновления дополнительных данных – 1200/16 Гц;

– скорость передачи 9 кбод, частота обновления данных точного канала – 70 Гц, частота обновления дополнительных данных – 70/16 Гц;

Запаздывание цифрового значения угловой скорости на входе аппаратуры потребителя.

Запаздывание цифрового значения угловой скорости на входе аппаратуры потребителя составляет приблизительно 1.5 периода (T) обновления данных точного канала, в точности равного периоду (т) дискретности преобразования аналогового сигнала (угловой скорости). Где T = 1/300 (либо 1/1200 или 1/70) секунды. Это следует из того, что

значение измеренной угловой скорости обычно относят к средине интервала измерения, а время передачи данных не больше периода T.

Контроль пропусков посылок (пропуска данных) при приеме данных.

Для контроля непрерывности приема посылок может быть использована информация, содержащаяся в счетчике Count – четвертый байт посылки. При отсутствии сбоев величина в счетчике увеличивается на единицу в каждой следующей посылке. Алгоритм проверки отсутствия пропусков информации может иметь следующий вид:

Count(i) – Count(i-1) = 1, если Count(i) > 0;
Count(i) – Count(i-1) + 16 = 1, если Count(i) = 0 ;

где i - номер последней посылки, Count(i) – значение счетчика Count, (четвертый байт посылки). Следует отметить, что контроль пропуска информации при измерении угловой скорости проводить не обязательно. При интегрировании сигнала ВОГ (при измерении углов поворота) пропуски информации искажают результат измерения угла.

Пример алгоритма (фрагмент программы) распаковки данных

/* Алгоритм (фрагмент программы) распаковки выходного сигнала ВОГ с цифровым выходом. Принимает и распаковывает посылку 8 байт в реальном масштабе времени:

Номер байта

0	1	2	3	4	5	6	7
0xdd	data(L)	data(H)	data(Midl)	Md(Count)	Md[](H/L)	CtrlSum(H)	CtrlSum(L)

0x00	MD[0](H)
0x01	MD[0](L)
0x02	MD[1](H)
0x03	MD[1](L)
.....	
0x0E	MD[7](H)
0x0F	MD[7](L).

Вычисляет значения выходного сигнала ВОГ-data, медленных данных md[0] – md[3], температуры, напряжения питания, тока, контрольного сигнала) и выставляет сигнал готовности данных - got_d. А также фиксирует ошибки, возникающие при приеме данных, а именно:
ошибку порта err_port, ошибку контрольной суммы ContrlSum_err, ошибку пропуска данных Count_err. В основной программе должны быть объявлены глобальные переменные:

```
    unsigned char b[9],a, err-port,got_d; int Contrl_sum1,Contrl_sum2; int md[16];
    unsigned int sinx=0,ContrlSum_err,Count_err;
    long int datc;
    int a;
*/

If( input(Bait))   // если новый байт принят.  Например: if(inp(0x3FA) & 0x04)
                        {

                        a= input(Bait);             // Чтение принятого байта . Например :   a=inp(0x3F8);
                        if (err_port)    err_port++;  // если ошибка порта (ошибка четности)
          if(sinx>0 & sinx<9)    // если синхронизация есть – осуществляется цикл приема данных;

                              // sinx – номер принятого байта

          {

          b[sinx]=a;

                sinx++;

            if(sinx = = 9)      //принят нулевой байт следующей посылки

          {
```

```
            if(b[8] == 0xdd)            //проверка синхронизации - проверка условия, что нулевой  байт

                                        //следующей посылки равен 0xdd

          {

Contrl_sum1=(int)b[1]+(int)b[2]+(int)b[3]+(int)b[4]+(int)b[5];    // проверка контрольной суммы
Contrl_sum2=(int)b[6]<<8 | (int)b[7];

if(Contrl_sum1 != Contrl_sum2)
       ContrlSam_err++;                                                  // фиксация ошибки контрольной суммы

datc=((long int)b[2]<<24 | (long int)b[3]<<16 | (long int) b[1]<<8)>>8; //  значение выходного сигнала ВОГ – U(Ω)

   sinx=1;

  hd=ld;       ld=b[5];   //запоминание предыдущего байта «медленных данных»  и прием очередного байта


 if (b[4]==0x01)
    md[0]=(int)hd<<8 | (int)ld;         //температура T
if (b[4]==0x03)
     md[1]=(int)hd<<8 | (int)ld;         //напряжение питания  U
 if (b[4]==0x05)
     md[2]=(int)hd<<8 | (int)ld;          //ток  I
  if (b[4]==0x07)
     md[3]=(int)hd<<8 | (int)ld;         //   Контрольный сигнал  KS
//При необходимости  аналогичным образом можно принять еще 4 значения резервных данных R1…R4

 if(b[4] – count1 ! = 1)       // контроль пропуска   данных по приращению счетчика Count «медленных  данных»
//данных»
              Count_err++;     //фиксация ошибки пропуска данных

  Count1=b[4];
 If(Count1=15 )
    Count1 =  -1;
 got_d=1;       //индикатор готовности данных – приема и распаковки 8 байтовой посылки. Обнуляется после чтения
данных основной программой
}
}
else       //если не 0xdd  при sinx=9. Сбой синхронизации

     sinx=0;
 if(!sinx)
b[0] = a;

if(b[0]= =0xdd)          //если синхронизация есть

 {
  sinx=1;      got_d=0;          datc=0;
}
else
sinx=0;
}
}
// END
*/  Данный алгоритм осуществляет пассивный контроль (фиксацию) ошибок при приеме и распаковке данных. То есть
флаг готовности данных выставляется и при наличии ошибок, а вопрос использования данных решается в основной
программе.*/
```