

Фаззі-модель може бути створена на базі Mamdani або Sugeno [4]. В моделі Mamdani вихід знаходиться після процесу дефаззифікації. В цьому випадку можна визначити кількість множин, вид функцій і метод дефаззифікації. В моделі Sugeno виникає проблема визначення лінійної вихідної функції і вона не дає необхідної точності.

Створення робастних засобів контролю за технічним станом породоруйнівного інструменту базується на використанні цифрової обробки вимірюваної інформації безпосередньо в первинному вимірювальному контролері, що дає переваги в експлуатації і дозволяє одним приладом замінити декілька первинних пристроїв контрольно-вимірюваної техніки.

1. Чигур І.І. Аналіз задачі алгоритмізації контролю технічного стану породоруйнівного інструменту // Розвідка і розробка нафтових і газових родовищ. Серія: Технічна кібернетика та електрифікація об'єктів паливно-енергетичного комплексу. – 2001, Вип. 37 (7). – С. 114-119.
2. Семенцов Г.Н., Горбійчук М.І., Чигур І.І. Математичний аналіз критеріїв відпрацювання доліт // Нафтова і газова промисловість. – 2001, № 6. – С. 25 – 28.
3. Лукас В.А. Теорія автоматического управління: Учеб. для ВУЗов. – М.: Недра, 1990. – 416 с.
4. Kästner W., Fenske A., Hampel R. Improvement of the robustness of Model-based. 5. Measuring Methods using Fuzzy Logic // World Scientific, Proceedings of the 3rd International FLINS Workshop. - Antwerp., Belgium, 1998. – pp. 129-142.

УДК 681.3

ВИКОРИСТАННЯ АНАЛОГОВИХ КОДЕКІВ ДЛЯ НАУКОВИХ ДОСЛІДЖЕНЬ

© Євчук О.В., Ровінський В.А., Стрілецький Ю.Й., 2006

Івано-Франківський національний технічний університет нафти і газу

Розглянуто способи використання аналогових кодеків, для вимірювання змінних аналогових сигналів низької частоти. Наведено схемні рішення вимірювальних пристроїв і програмна реалізація типових функцій, що застосовуються при роботі зі звуковими платами персональних комп'ютерів типу IBM PC в яких встановлені такі кодеки

При проведенні експериментальних та науково-дослідних робіт, пов'язаних з діагностикою або автоматизацією об'єктів нафтогазового комплексу, нерідко виникає потреба вводу вимірюваних даних в ЕОМ для подальшої їх обробки за допомогою спеціалізованого програмного забезпечення. На початкових етапах експериментів, які стосуються оптимального вибору інформативних параметрів об'єкта діагностування або управління, досліднику необхідно визначитись з розрядністю блоку аналогово-цифрового перетворення (АЦП), кількістю його робочих каналів, швидкістю А/Ц перетворення тощо. Ця інформація необхідна для правильного вибору АЦП і впливає на структуру інформаційно-вимірювальної системи. Однак зробити правильні висновки про параметри робочих сигналів можна тільки зробивши попередні вимірювання. Як правило, в такому разі

вибір АЦП залежить в першу чергу від досвіду проектувальника і вибір здійснюється із значним запасом як по розрядності, так і по швидкодії. Це негативно позначається на собівартості розроблюваної системи.

На сучасному етапі світова промисловість випускає значну кількість АЦП як у вигляді інтегральних схем (найбільш поширені виробники: Analog Devices, Maxim, Texas Instruments, Linear Technology), так і у вигляді модулів збору інформації (Siemens, National Instruments). Останні незважаючи на значну вартість, найбільш широко застосовуються в промислових організаціях, оскільки їх використання потребує знань тільки системотехніки та програмування і не потребує досконального знання схемотехніки прецизійних та вимірювальних схем. Власна розробка модулів збору інформації з розрядністю $n > 14$ в загальному випадку є складним завданням, яке може

виявитись не під силу розробнику без попереднього досвіду проектування вимірювальних та аналогово-цифрових схем великої розрядності.

Оскільки результати вимірювання практично завжди обробляються за допомогою ЕОМ, постає задача передачі даних від АЦП в реальному часі. Сучасні операційні системи, під управлінням яких працюють ЕОМ, висувають ряд вимог до написання драйверів зовнішніх пристроїв.

Частковим винятком в цьому випадку є використання інтерфейсів COM та LPT, які легко програмуються, однак перші мають обмежену для деяких випадків (наприклад, для віброакустики) швидкість обміну даними на рівні ≈ 10 кБайт/с, а другі поступово зникають, оскільки на заміну їм приходить магістраль USB. Слід зауважити, що наявні в продажу перетворювачі USB інтерфейсу, наприклад, перехідні пристрої „USB/ LPT” коректно працюють тільки з вузлами поблокового обміну даними і бітової роботи із регістрами порту не підтримують. Крім того, в багатозадачних системах, якими є сучасні ОС, забезпечити роботу зовнішнього вузла під керівництвом центрального процесора в потрібний момент часу практично неможливо, оскільки час роботи процесора ділиться між великою кількістю задач.

Практично всі сучасні ПЕОМ обладнані власними модулями АЦП/ЦАП, які використовуються для вводу і відтворення звукової інформації. Їх можна успішно використати для вводу і виводу дослідницької аналогової інформації, оминувши вищезгадані труднощі. Це зумовлено тим, що високоякісне звуковідтворення потребує дотримання всіх вимог схемотехнічного проектування, включаючи добір компонентів, розробки топології друкованих плат із забезпеченням мінімальних перехресних зв'язків та написання драйверів, що гарантують синхронність та неперервність передачі потоків даних. Розробка звукових пристроїв здійснюється висококваліфікованими фахівцями, а виготовлені в умовах масового виробництва звукозаписуючі і відтворюючі системи мають суттєво меншу вартість, ніж аналогічні промислові пристрої.

Існує значна кількість програм, призначених для професійного звукозапису та звуковідтворення (WaveLab, SpectraLAB, Nuendo, Cubase, SoundForge, Adobe Audition, CoolEdit), які дозволяють здійснювати запис та попередню обробку даних в реальному часі.

З 1997 р. звукові плати ПЕОМ здебільшого виконують на типових мікросхемах, які називаються звуковими кодеками типу АС'97[1]. Типовий кодек містить двоканальний 16-ти розрядний дельта-сігма АЦП і двоканальний 16-ти розрядний дельта-сігма ЦАП[2].

Наявність двох каналів із синхронним високоточним 16-ти розрядним перетворенням дає можливість легко вимірювати величину сигналів та співвідношення між ними. При цьому зрозуміло, що якість використаних в звуковій платі елементів (насамперед конденсаторів) суттєво залежить від ціни. Для особливо якісних перетворень використовують зовнішні звукові плати. За рахунок віддаленості від джерела завад (цифрової ЕОМ) забезпечується 24 розрядне перетворення.

Розглянемо обмеження, які виникають при використанні звукових плат у наукових дослідженнях.

Мікросхема кодека живиться однополярним живленням, здебільшого 5В. Аналогові сигнали від зовнішніх джерел, як правило, двохполярні. Тому для відтворення розмаху сигналу на входи кодека подають напругу зміщення, рівну половині діапазону вхідної напруги, утворюючи таким чином віртуальний нуль. Тому при відсутності вхідного сигналу з кодека надходить код, рівний половині діапазону. Для захисту цієї напруги зміщення від зовнішнього впливу на звукових платах встановлюють розділюючі конденсатори. Таким чином за допомогою звукової плати неможливо визначити рівень постійної та інфранизькочастотної напруги. Гарантується перетворення кодеком сигналів в діапазоні від 20Гц до 20кГц[2].

Допустимий діапазон вхідних напруг кодека не стандартизується і здебільшого становить 1В середньоквадратичної напруги по лінійному входу. Для захисту від можливої більшої вхідної напруги на вході встановлюють резисторний подільник на 2 (рис.1). Тому вхідний опір звукової плати становить близько 8кОм.

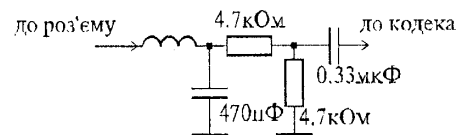


Рис. 1. Типовий вигляд вхідної ланки звукової плати

Стабільність параметрів кодека від температури не обумовлюється. Опорна напруга кодека, яка визначає величину рівня квантування АЦП, має значення від 2,1В до 2,5В[3]. Тому для різних мікросхем кодека виміряні значення будуть відрізнятися.

Висока точність перетворення вимагає продуманого вибору вхідних ліній, їх довжини, способу екранування.

Основний недолік використання звукової плати як модуль АЦП – це наявність нижньої частоти смуги пропускання. В залежності від типу

звукової плати нижня частота знаходиться в діапазоні. $f_H \approx 5 \dots 15$ Гц. Вимірювання за допомогою звукової плати постійних сигналів без додаткових пристроїв неможливе. Однак, використовуючи найпростіші пристрої перетворення частот – амплітудні модулятори, можна легко усунути цей недолік.

Часто при дослідженні треба вимірювати температуру. При температурах до 400°C для цього використовують термоопори. Вимірювання опору

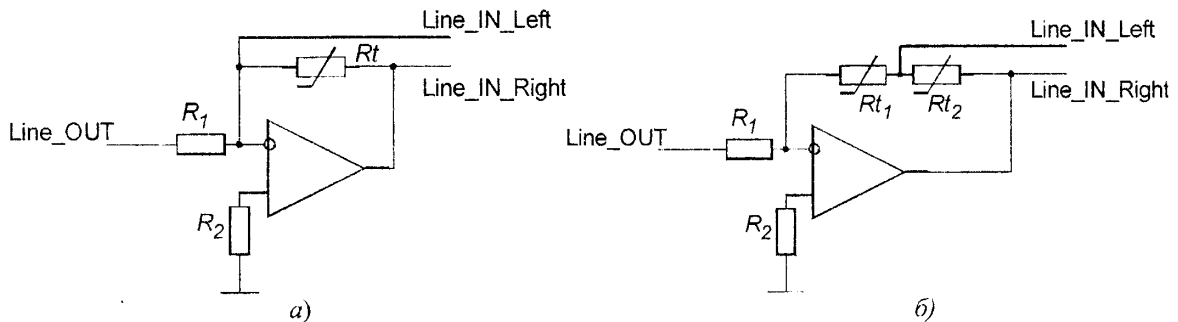


Рис. 2. Використання перетворювача напруга-струм при вимірюванні температури за допомогою звукової плати

Якщо використати прецизійний операційний підсилювач із малим вхідним струмом, то можна нехтувати вхідним струмом зміщення і вимірювати напругу на двох терморезисторах (рис.2,б). Оскільки на інверсному вході операційного підсилювача буде віртуальний нуль, лівий канал АЦП вимірюватиме напругу на $Rt1$, а правий на двох опорах $Rt1$ та $Rt2$. В такій схемі не компенсується вплив з'єднувальних провідників, тому вимірювати температуру можна тільки на невеликій віддалі.

Для вимірювання тиску зручно використовувати інтегральні датчики абсолютного

тиску і перепаду тиску. Доступними і недорогими є датчики типу MPX фірми Motorola. Серед них є звичайні тензорезистивні мости. Датчики із мостом потребують джерела струму, а вихідний сигнал таких датчиків – диференційна напруга розбалансу. Якщо скористатися додатковим інтегральним інструментальним підсилювачем $DA3$, схему (рис.2,а) можна модифікувати (рис.3). Додовнивши схему терморезистором із інструментальним підсилювачем $DA2$, отримаємо можливість вимірювати тиск і температуру одночасно.

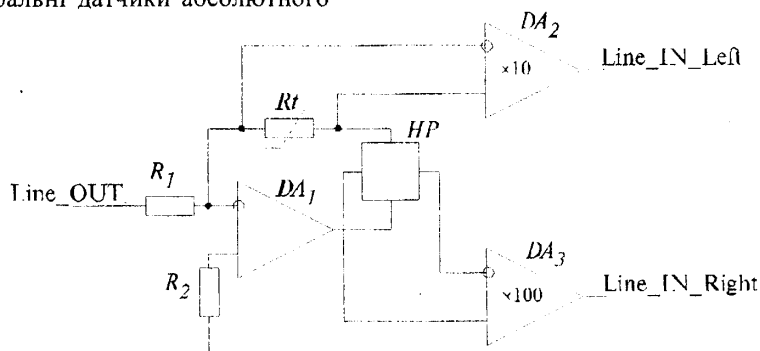


Рис. 3. Схема вимірювання температури та тиску за допомогою звукової плати

В більшості випадків джерелом опорного сигналу може бути не ЦАП кодека, а зріцева напруга. При цьому слід перед початком експериментів провести калібрування АЦП кодека

звукової плати, наприклад, за допомогою схеми, приведеної на рис.4.

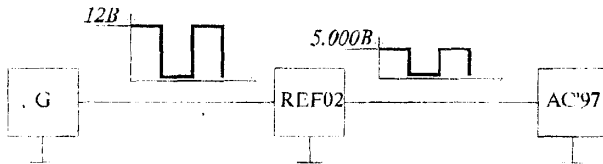


Рис. 4. Структурна схема калібрування звукової плати

Принцип дії схеми на рис.4 полягає у формуванні еталонних прямокутних імпульсів з амплітудою $U_{вих} = 5,000\text{В}$, які формуються за допомогою прецизійного джерела опорної напруги (наприклад, REF02), на вхід якого подаються імпульси від кварцового генератора напругою $U_{qx} \approx 12\text{В}$. Вимірювання еталонних прямокутних імпульсів за допомогою звукової плати і порівняння їх із відомими значеннями дозволяє скоректувати коефіцієнт підсилення звукової плати в залежності від температури. Перед початком калібрування та проведення вимірювань слід забезпечити вирівнювання температур всіх компонентів звукової плати шляхом прогріву вимірювального пристрою протягом не менше 30хв. Приєднання джерел сигналів до звукової плати слід здійснювати за допомогою окремих провідників, що значно зменшить рівень перехресних завад. Деякі звукові плати мають окремі роз'єми для кожного із сигналів, що ще більше зменшує рівень перехресних завад.

Головною перевагою використання звукової плати ПЕОМ є велика кількість програмного забезпечення для роботи з нею, а також простота використання її у власних проектах. Для управління звуковою платою передбачений механізм API-функцій ОС Windows.

Запис даних та відтворення можна здійснювати за допомогою як типових програм, так і за допомогою спеціально розробленого програмного забезпечення. Приклад функції ініціалізації звукової плати мовою Object Pascal в середовищі Borland Delphi:

```
function TForm1.InitSB:Boolean;
var header:PWaveHdr;
    tmp:integer;
    tim:double;
begin
    if scan_level then
scan_busy:=true;
    InitSB:=true;

Freq:=SBFreq[RadioFvib.ItemIndex]; //
Вибір частоти дискретизації з ряду
//визначення формату
new(WaveFormatExt);
```

```
with WaveFormatExt^ do begin
    wFormatTag:=
WAVE_FORMAT_PCM; // Формат даних
Wave PCM
    cbSize:=1;
    if scan_level then
nChannels:=2 else
nChannels:=Mono.ItemIndex+1;
    wBitsPerSample:=16;
    nSamplesPerSec:=Round(Freq);

nBlockAlign:=nChannels*(wBitsPerSample
div 8);

nAvgBytesPerSec:=nBlockAlign*nSamples
PerSec;
end;

tmp:=waveInOpen(nil,0,WaveFormatExt,0
,0,WAVE_FORMAT_QUERY);
if tmp<>0 then InitSB:=false;
//Визначення хендла
HwaveIn:=new(PHwaveIn);

tmp:=waveInOpen(HWaveIn,0,WaveFormatE
xt,Form1.handle,0,CALLBACK_WINDOW);
if tmp<>0 then begin

Application.MessageBox('Error
creating handle',

'InitSB',MB_ICONINFORMATION);
    InitSB:=false;
end;
//Додаємо заголовок
header:=new(PWaveHDR);
with header^ do begin
    lpdata:=pointer(PResSound);
    if scan_level then tim:=scan_tim
else tim:=Maxtime;

dwbufferlength:=Ceil(tim*WaveFormatE
t^.nAvgBytesPerSec);
    dwbytesrecorded:=0;

dwUser:=0;dwflags:=0;dwloops:=0;
end;

tmp:=waveInPrepareHeader(HWaveIn^,Hea
der,sizeof(TWavehdr));
if tmp<>0 then begin

Application.MessageBox('Error
creating header',

'InitSB',MB_ICONINFORMATION);
    InitSB:=false;
```

```

end;
//Додаємо буфер, куди звукова
плата буде записувати дані
tmp:=waveInAddBuffer(HWaveIn^,Header,
sizeof(TWaveHdr));
if tmp<>0 then begin
Application.MessageBox('Error adding
buffer',
'InitSB',MB_ICONINFORMATION);
InitSB:=false;
end;
//Тепер стартуємо
if waveInStart(HwaveIn^)<>0
then InitSB:=false else InitSB:=true;
end;

```

Важливим моментом при проведенні вимірювань за допомогою звукових плат є встановлення необхідного коефіцієнта підсилення вхідних або вихідних аналогових сигналів. Ця операція здійснюється за допомогою мікшера операційної системи, який є складовою частиною драйвера звукової плати і керує підсилювачами кодека. Коефіцієнти підсилення можуть бути виставленими з типовою роздільною здатністю 65536 відносних одиниць і забезпечуватимуть регулювання підсилення в межах ± 20 дБ. В багатозадачній системі коефіцієнт підсилення слід виставляти перед кожним вимірюванням. Механізм API – функцій ОС Windows дозволяє змінювати рівні системного мікшера, який взаємодіє з конкретним, в загальному випадку, довільним драйвером звукової плати. Наступний приклад функції, яка встановлює повзунки “Line Level” та “Record Level”, дозволяє реалізувати встановлення цих повзунків в довільні допустимі значення. Приклад реалізований мовою Object Pascal в середовищі Borland Delphi:

```

Function
TForm1.SetLevels(LineLevel,RecordLevel:WORD):boolean;
var
rc: MMRESULT;
// Код повернення .
hMix: HMIXER;
// Хендл мікшера що використовується
в
// API функціях мікшера.
mxc: MIXERCONTROL;
// Містить управляючі дані мікшера.
mxl: MIXERLINE;
// Містить лінійні дані мікшера
mxlc: MIXERLINECONTROLS;

```

```

mxcd:
TMIXERCONTROLDETAILS ;
i,k: UINT;
details: array [0..1] of
integer;
begin
SetLevels:=false;
rc:=mixerGetID(0,
k,MIXER_OBJECTF_WAVEIN);
rc:= mixerOpen(@hMix,
k,0,0,MIXER_OBJECTF_WAVEIN);
if (MMSYSERR_NOERROR <> rc)
then EXIT; // Couldn't open the
mixer.
ZeroMemory(@mxl,sizeof(mxl));
mxl.cbStruct:=sizeof(mxl);
mxl.dwComponentType:=MIXERLINE_COMPON
ENTTYPE_DST_WAVEIN;
ZeroMemory(@mxlc,
sizeof(mxlc));
mxlc.cbStruct := sizeof(mxlc);
mxlc.cbmxctrl := sizeof(mxc);
mxlc.pamxctrl := @mxc;
ZeroMemory(@mxc, sizeof(mxc));
mxc.cbStruct := sizeof(mxc);
mxc.fdwControl:=0;
ZeroMemory(@mxcd,
sizeof(mxcd));
mxcd.cbStruct:=sizeof(mxcd);
mxcd.cChannels:=1; mxcd.cbDetails:=4;
mxcd.paDetails:=@details;
// Рівень запису
rc:= mixerGetLineInfo(hMix,
@mxl,MIXER_GETLINEINFOF_COMPONENTTYPE
);
if (MMSYSERR_NOERROR <> rc)
then EXIT;
mxlc.dwLineID := mxl.dwLineID;
mxlc.cControls :=
mxl.cControls;
mxlc.dwControlType
:=MIXERCONTROL_CONTROLTYPE_VOLUME;
// Вибір лінії "Line In"
k:=mxl.cConnections;
for i:=0 to k do
begin
mxl.dwSource:=i;
rc:=mixerGetLineInfo(hMix,
@mxl, MIXER_GETLINEINFOF_SOURCE);
if
mxl.dwComponentType=MIXERLINE_COMPONE
NTTYPE_SRC_LINE then break;
if (MMSYSERR_NOERROR <> rc)
then EXIT;

```

```

end;

// Лінійний рівень
mxlc.cControls :=
mxl.cControls;;
mxlc.dwLineID:=mxl.dwLineID;
mxlc.dwControlType
:=MIXERCONTROL_CONTROLTYPE_VOLUME;
rc :=
mixerGetLineControls (hMix, @mxlc,
MIXER_GETLINECONTROLSF_ONEBYTYPE);
if rc<>0 then exit;
mxcd.cChannels:=mxl.cChannels;
mxcd.dwControlID:=
mxl.dwControlID;
rc :=
mixerGetControlDetails (hMix,
@mxc, MIXER_GETCONTROLDETAILSF_VALUE)
;
if rc<>0 then exit;
details[0]:=LineLevel;
details[1]:=LineLevel;
rc :=
mixerSetControlDetails (hMix,

```

```

@mxc, MIXER_GETCONTROLDETAILSF_VALUE)
;
if rc<>0 then exit;
rc:=MixerClose (hMix);
SetLevels:=true;
end;

```

Використовуючи вихідні лінії ЦАП кодека для формування багаточастотних сигналів, наприклад, DTFM, можна формувати дискретні вихідні сигнали. Дискретними сигналами можна перемикаєти вимірювані сигнали, збільшуючи їх кількість.

Таким чином використання звукової плати ПЕОМ дає дослідникам зручний механізм для проведення вимірювань різноманітних фізичних величин на початковому етапі досліджень.

1. Гук М. *Аппаратные средства IBM PC. Энциклопедия. 3-е изд. – СПб.: Питер, 2006. – 1027с.*
 2. AC'97 *SoundMAX® Codec AD1885. www.analog.com*
 3. *Low sampling rate operation for burr-brown audio data converters and codecs. Martin R., Kawai H. AB-148. www.ti.com*