C-language function class for signal processing at ADC testing

Ján Šaliga, Szabolc Csernok
Department of Electronics and Telecommunications, Technical University in Košice,
Park Komenského 13, SK-04120 Košice, Slovak Republic,
Phone: +421 55 6022866, Fax: +421 55 6323989, E-mail: jan.saliga@tuke.sk

Summary - *The paper presents some ideas and results from the attempt to create a comprehensive suite of data processing, C-language functions that enables simplifying the development of new, user oriented software for testing analog to digital converters (ADC) according the IEEE standards 1057, 1241 and DYNAD. The developed function class covers both dynamic and histogram based test methods. Nowadays, it is prepared in the form of instrument driver – function panel (*.fp) for the software development package LabWindows/CVI by National Instruments. All functions were developed, debugged and tested in this environment.*

The paper also contains some results from comparing the new developed function class used in a developed examples of end-user applications with test data processing Matlab software by Kollar and Markus [6].

Keywords: ADC testing, C language
LabWindows/CVI, IEEE Std. 1241, DYNAD

INTRODUCTION

Analog-to-digital converters (ADC) are very often used and widely spread component parts of many electronic devices and systems. Their development and applications are unthinkable without performance testing. The need to standardize the ADC test procedures lead to the preparation of new standards IEEE1241 [1] and DYNAD [2] that spread the application area and add some new ideas to the IEEE 1057 Std. [3]. Particularly DYNAD contains very detail mathematical descriptions of data processing procedures.

Along with the innovations in standardization process, the activities to develop software implementations of standardized test methods for widely implementation these procedures in practice were introduced. The first one [4] used the graphical programming environment LabVIEW and the second [5], [6] in Matlab environment. Both of them are built in the form of end user application, which process the data from ADC test in the form of a data file. This fact partially reduces their areas of possible applications, and excludes them from usage in special user-dependent applications e.g. testing more comprehensive systems where data processing is only a part of more complex software including data acquisition and control activities especially in real time applications.

DEVELOPED C-LANGUAGE DATA PROCCESING FUNCTION CLASS

C-language is one of most popular and widely spread programming languages. The most famous C-language based software development environments for general applications are Microsoft Visual C, Borland C++, Watcom C etc. Probably the most favorite and useful C-language based programming environment for development of applications in measurement area is Labwindows/CVI by National Instruments [7]. Its main advantages are

- Support by very enhanced measurement and data analysis oriented function libraries,
- Integrated data acquisition libraries
- System of instrument drivers
- Integration of User Interface Editor for very simple creating of Graphical User Interface

More over, the last versions of LabWindows/CVI come in form of Measurement Studio with support libraries that can be integrated into Visual Basic and Visual C++ environments [7].

It invoked the idea to create a C-language class of functions, which could contain the most of common calculations needed for analysis of a record from ADC test according to [1], [2], and particularly [3]. Such a library can be subsequently used at any software development, which consists of a part dedicated to such a signal processing, e.g. PC sound card testing [8], [9].

The actual working version (0.4) of developed and tested system of data processing functions comes particularly from [3]. Some overstrikes found in the available draft were corrected and the equations were compared with equal ones in [1] and [2]. The acquired results created the mathematical basis for developed system of functions. Coming from authors' experiences gained at practical ADC testing and from checking over the functions in test applications, the system was supplemented by some additional supporting procedures. The nowadays system – version 0.4 can be divided into 4 main function subclasses focused on:

- Histogram-based test procedures,
- Dynamic test procedures with data processing in time domain (fitting),
- Dynamic test procedures with data processing in frequency domain,
- Miscellaneous additional and support tasks useful for ADC testing software applications.

The complete list of implemented functions in version 0.4 is in table 1. The histogram-based test procedures subclass contains the functions for:

- Determining minimal needed number of recorded samples for required uncertainty of results,
- Processing recorded data such as computation of cumulative histogram, estimation of code transition levels, gain and offset, DNL and INL, as well as amplitude and DC value of testing sinewave.

The dynamic test procedure with data processing in time domain function subclass contains the functions for:

- Estimation of input test signal parameters by four or three parameters fitting procedure,
- Reconstruction of input sinewave and calculation of residuals,
- Calculation of SINAD and power spectrum distribution function (PSDF).

The function class for processing record in the frequency domain is divided furthermore on the subclasses for

- Coherent and
- Incoherent sampling

Both subclasses use the Fourier series that could be calculated from the record by the common function Fourier(.). Subclass for processing record acquired at coherent sampling contains such function as

- Estimation of noise floor, amplitude and DC value of input signal,
- Estimation of SINAD, SNR and PSDF

Subclass for processing record acquired at incoherent sampling contains such function as

- Estimation of amplitude and frequency of basic harmonic and noise floor,
- Estimation of SINAD, SNR and PSDF.

Practical experiences with the previous versions of the function library show a need and usefulness of some additional supporting functions that were grouped into a subclass "Miscellaneous", which contains functions for

- Conversion of arrays,
- Calculation of nearest coherent input test frequency,
- Calculation of minimal number of record length for dynamic testing.

Class/Panel Name:	Function Name:
Histogram Based Test	
Required Num. Of Samples for NL	ReqNoOfSampNL
Cumulative Histogram	CumHist
Determine Code TranLev From Sin	CompTranLevSin
Determine Nominal Trans Levels	DetNomTranLevels
Gain & Offset Least Squares Fit	GainOffsetLSF
Gain And Offset from Min-Max	GainOffsetMinMax
Gain And Offset from End Points	GainOffsetEndPoints
DNL From Trans Levels	DNLFromLevels
INL From Trans Levels	INLFromLevels
Amplitude And Offset Estimation	EstimateAmpDC
Miscellaneous	
Copy Integer Array To Double	CopyIntToDouble
Convert SINAD to ENOB	ConvSINADtoENOB
Nom Of Samples for Dynamic Test	NoOfSampDynTest
Data Processing In Time Domain	
Four Parameter fitting Method	FourParameterMethod
Fixed Frequency fitting Method	FixedFrequencyMethod
Convert Amp & Phase to Sinewave	ConvAmpPhase
Compute Residuals	CompResiduals
Compute SINAD	ComputeSINAD
Compute Phase For Samples	CompPhaseForSamp
Compute PSDF	CompPSDF
Data Processing In Freq Domain	
Fourier transformation	Fourier
With Coherent Sampling	
Comp Amp DC Freq Coherent Samp	CompAmpDCFreqCoherentSamp
NoiseFloor Eval Coherent Sampl	NoiseFloorEvalCoherentSamp
Calculate SINAD Coherent Sampl	CalcSINADCoherentSamp
Calculate SNR Coherent Sampling	CalcSNRCoherentSamp
Calculate THD Coherent Sampling	CalcTHDCoherentSamp
With Non Coherent Sampling	
Line Estimate	LineEstimate
Calculate correction Factor	CompFact
Noise Floor Eval NonCoher Samp	NoiseFloorEvalNonCoherentSamp
Calculate SINAD NonCoher Samp	CalcSINADNonCoherentSamp
Calculate SNR NonCoherent Samp	CalcSNRNonCoherentSamp
Calculate THD NonCoherent Samp	CalcTHDNonCoherentSamp

Tab. 1. List of data processing functions implemented in the library, ver. 0.4.

EXAMPLE OF APPLICATION OF THE DEVELOPED FUNCTION CLASS

The following example shows the typical applications of the developed functions in acquiring and processing a record from dynamic ADC test.

1. Test setup (frequency and record length)

```
NoOfSampFreq (NoOfBit, EstimDNL, &M);
SelectInputFrequency (M, fs, fd, &fi, &df);
```

2. Record processing in time domain

```
LineEstimate (sdata, M, 1, &freq);
freq= freq / M;
FixedFrequencyMetod (sdata, M, freq, x);
//alternative
FourParameterMetod (sdata, M, &freq, x);
ConvToAmpPhase (x[0], x[1], &Amp, &Ph);
CompResiduals (sdata, M, Amp, Ph, x[2], freq, NULL,
    &Erms);
ComputeSINAD (Amp, Erms, &SINAD);
ConvSINADtoENOB (SINAD, 1, &ENOB);
```

3. Alternative record processing in frequency domain (coherent sampling)

```
Fourier (sdata, M, y2);
CompAmpDCFreqCoherentSamp (y2, M, &J, &Amp, &dc);
NoiseFloorEvalCoherentSamp (y2, M, J, hmax, &NFI2);
CalcSINADCoherentSamp (NFI2, y2, M, J, &SINAD);
CalcSNRCoherentSamp (NFI2, y2, M, J, hmax, &SNR);
CalcTHDCoherentSamp (y2, M, J, hmax, &THD);
ConvSINADtoENOB (SINAD, 1, &ENOB);
```

4. Alternative record processing in frequency domain (incoherent sampling)

```
LineEstimate (sdata, M, TypeOfWinForFreqEst, &j);
ACDCEstimator (sdata, M, &pom, &dc);
LinEv1D (sdata, M, 1.0, -dc, temp);
ScaledWindow (temp, M, TypeOfWin, &w_c);
Fourier (temp, M, y2);
CompFact (TypeOfWin, M, Wc2);
NoiseFloorEvalNonCoherentSamp (y2, M, j, hmax,
    TypeOfWin, &NFI2);
CalcSINADNonCoherentSamp (NFI2, y2, M, j, hmax,
    TypeOfWin, w_c.enbw, Wc2, &SINAD);
ConvSINADtoENOB (SINAD, 1, &ENOB);
CalcTHDNonCoherentSamp (y2, M, j, hmax, Wc2, &THD);
CalcSNRNonCoherentSamp (NFI2, y2, M, j, hmax,
    TypeOfWin, w_c.enbw, Wc2, &SNR);
```

EXPERIMENTAL RESULTS AND COMPARISON

The developed library has been examined in many tests to debug and approve. Comparison of the results acquired from the library application on the record with the results obtained from Matlab software referred in [5] and [6] for the same records was chosen as the main criteria of correctness. Some achieved results of such testing are shown in Fig. 1 and Tab. 2.

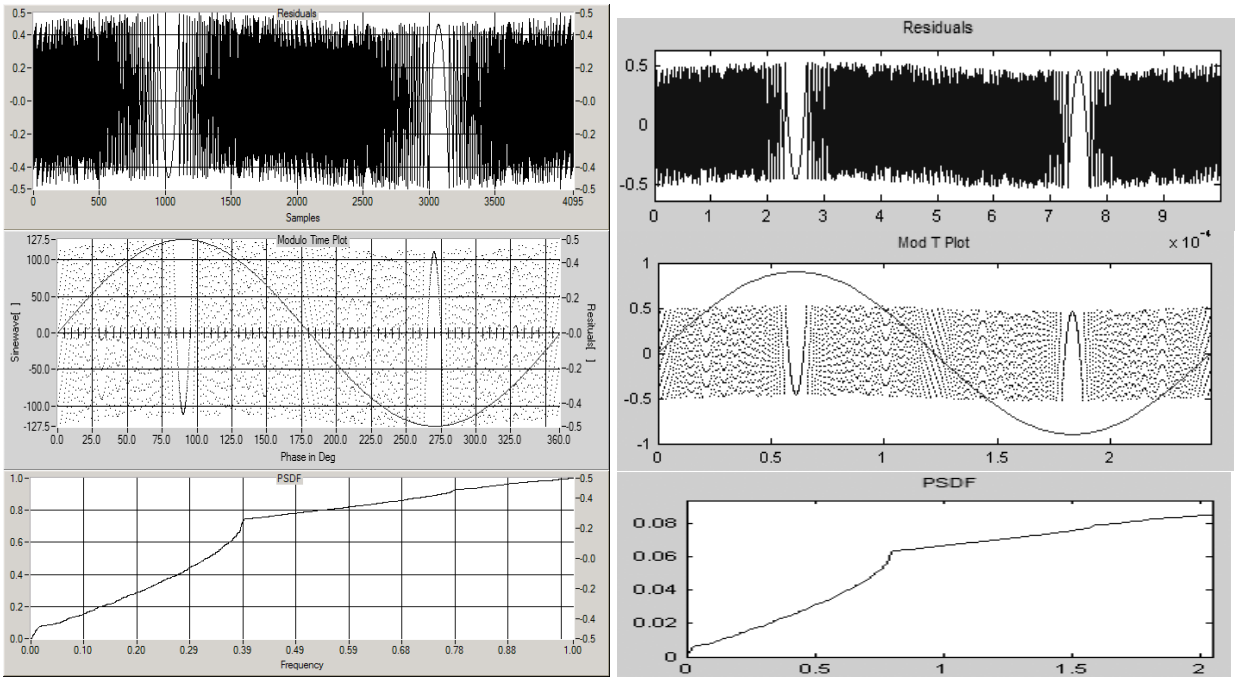

Fig. 1. Comparison example of graphical results obtaining from the developed library (left column) and from Matlab software [6] (right column), the record from file “noiseless8bit.dat“included in [6].

File with record	Results achieved by Matlab software							
	Ampl. [bit]	DC offset [10^{-3} bit]	Frequency	Phase[°]	ENOB	Amplitude [%FS]	Noise +distort	SINAD [dB]
Example00.dat	127	-1.24	107,2346	-51.4	8.03	99.2	0.283	50
Example01.dat	127	-9.09 .10^{-3}	100	-54	9.3	99.5	0.117	58
Example02.dat	127	-0.356	99	-54.4	8.02	99.2	0.284	50
Example03.dat	127	-0.94	101	-53.6	8.02	99.2	0.285	50
Example04.dat	127	-1.24	107.2346	-51.4	8.03	99.2	0.283	50
Example05.dat	127	20.1	107.2346	-15.4	8	99.2	0.288	50
File with record	Results achieved by software with the developed function class in time domain							
	Ampl. [bit]	DC offset [10^{-3} bit]	Frequency	Phase[°]	ENOB	Amplitude [%FS]	Noise +distort	SINAD [dB]
Example00.dat	127.01	-1.245	107.2346	-51.41	8.029	99.23	0.2831	50.03
Example01.dat	127.33	-9.0.10^{-3}	100	-54.0	9.301	99.48	0.1172	57.71
Example02.dat	127.01	-0.356	99	-54.36	8.022	99.23	0.2845	49.99
Example03.dat	127.01	-0.94	101	-53.64	8.022	99.23	0.2845	49.98
Example04.dat	127.01	-1.25	107.2346	-51.41	8.029	99.23	0.2831	50.03
Example05.dat	127.01	20.06	107.2346	-15.39	8.001	99.22	0.2886	49.86

File with record	Results achieved by software with the developed function class in frequency domain (windowed functions)					
	Frequency	THD[%]	ENOB	SNR [dB]	NFI	SINAD [dB]
Example00.dat	107.23376	0	8.017	49.78	0.00739	50.02
Example01.dat	100.00376	0.00017	9.354	160.62	0.21 .10^{-6}	58.07
Example02.dat	99.00702	0	8.078	49.80	0.007278	50.39
Example03.dat	101.00168	0	8.060	49.96	0.007243	50.28
Example04.dat	107.23379	0	8.0166	49.78	0.007396	50.020
Example05.dat	107.2339	0	8.1236	50.42	0.006869	50.664

Tab. 2. Comparison of some results achieved by the Matlab software [6] and an application using the developed function class.

CONCLUSIONS AND FUTURE WORK

The system of C-language data processing functions incorporated in the developed function class proved a good accordance achieved result in applications with the reference Matlab software. It seems to become a useful support tool for everybody who needs to develop an application in C-language and especially in LabWindows/CVI including data processing from ADC tests. More over it enable simply testify a new test and data processing methods with comparison to standardized ones. On the other hand, the developed functions system (version 0.4) does not cover all standardized test methods. The class absents such functions as estimation SFDR and intermodulation distortion, better calculation and record error handling, etc.

The authors intend to create the Internet homepage dedicated to the developed library on the server http://measur.fei.tuke.sk soon and to ask the members of EUPAS as well as all specialists willing to test the library to add comments and their opinion on the incorporating functions.

REFERENCES

[1] IEEE Std. 1241 “Standard for Terminology and test methods”, draft ver. VS022500

[2] DYNAD, draft ver. 3.3., 2001, http://www.fe.up.pt/~hsm/dynad

[3] IEEE Std. 1057-1994, “Standard for Digitizing Waveform Recorders”

[4] J. Blair: “Sine-fitting software for IEEE Standards 1057 and 1241” *IMTC’99. Proceeding*, Venice, Italy, May 1999. pp. 1504-6.

[5] J. Márkus, I. Kollár: “Standard framework fot IEEE-Std-1241 in Matlab”, *IMTC’01 Proceeding*, Budapest, Hungary, May 2001, , pp. 1847 – 1852

[6] J. Márkus, “ADC Test Data Evaluation Program for Matlab” http://www.mit.bme.hu/services/ieee/ADC-test

[7] Home page of CVI http://www.ni.com/cvi

[8] J. Šaliga, L. Michaeli,: „Software for metrological characterisation of PC sound cards“,*11th IMEKO TC- 4 Symposium and 6th EWADC, Proceedings*, Lisbon, Portugal, September 2001, pp. 33- 37.

[9] L. Michaeli: Modelling of AD interfaces, *Mercury-Smekal*, Košice, Slovakia, 2001, ISBN 80-968550-1-8