International Journal of Electrical and Computer Engineering (IJECE)
Vol. 7, No. 6, December 2017, pp. 3484 – 3491
ISSN: 2088-8708, DOI: 10.11591/ijece.v7i6.pp3484-3491

Design and Implementation of an Embedded System for Software Defined Radio

A. E. Abdelkareem[1], Saad Mohammed Saleh[2], and Ammar D. Jasim[3]
[1,3]College of Information Engineering, Al- Nahrain University, Baghdad, Iraq
[2]College of Engineering, Diyala University, Diyala, Iraq

Article Info

Article history:
Received: Mar 20, 2017
Revised: Jun 18, 2017
Accepted: Jul 8, 2017

Keyword:
DSP
Embedded system
Receiver
Synchronization

ABSTRACT

In this paper, developing high performance software for demanding real-time embedded systems is proposed. This software-based design will enable the software engineers and system architects in emerging technology areas like 5G Wireless and Software Defined Networking (SDN) to build their algorithms. An ADSP-21364 floating point SHARC Digital Signal Processor (DSP) running at 333 MHz is adopted as a platform for an embedded system. To evaluate the proposed embedded system, an implementation of frame, symbol and carrier phase synchronization is presented as an application. Its performance is investigated with an on line Quadrature Phase Shift keying (QPSK) receiver. Obtained results show that the designed software is implemented successfully based on the SHARC DSP which can utilized efficiently for such algorithms. In addition, it is proven that the proposed embedded system is pragmatic and capable of dealing with the memory constraints and critical time issue due to a long length interleaved coded data utilized for channel coding.

Corresponding Author:
Dr Ammar E. Al-Qassab
College Of Information Engineering (COIE), Al- Nahrain University
Baghdad, Jadiria,P.O. Box 64004,Iraq
+964-7705802111
ammar.e@coie-nahrain.edu.iq

1. INTRODUCTION

Embedded systems have gained considerable attention in the last years. Nowadays, developing an embedded software is required everywhere specially with the advancements of networks technology which necessitates internet protocol (IP) for each device to derive the Internet of things (IoT). In terms of an embedded system, the authors in [1] develop an alarm system embedded in Altera DE0 FPGA and validated via an experiment test whereas in [2], a software framework is suggested to increase the speed of an embedded software. Selecting the most appropriate DSP processor and tackling a real-time signal is an important issue. Programmable DSP is more flexible, of a lower cost, and a higher speed than other processors, so it has become the best solution for many communication, medical, and industrial products because traditional microprocessors are inappropriate for such applications.

SHARC has been improved by using separate memories for data and instruction. In addition, it includes a high speed I/O controller to support Direct Memory Access (DMA). Furthermore, [3] mentioned that SHARC uses shadow registers for all the CPUs registers. They are used to accomplish the interrupt quickly by moving the entire register contents to these registers in a single clock cycle. Signal processing functions, such as Viterbi decoding, can be implemented using DSPs. For example, Analog Devices, TigerSHARC ADSP-101S and SHARC ADSP-21065L can be used in the baseband modem implementation. The first of these manipulates the Viterbi Decoder in 0.86 MIPS and 1024-point complex FFT in 32.75 s, and has been used as a multiprocessor structure by [4] with FPGA to implement an OFDM underwater acoustic communication system. The second manipulates 1024-point FFT in 0.274 ms. In addition, TMS320C6416 is designed for 3rd

Generation Partnership Project (3GPP) turbo code and is capable of decoding up to 12 Mbps (6 Iterations) [5].

In [6], a practical description of the design choices and hardware implementation details based on Spartan3 xc3s2000 FPGA required to build an efficient symbol synchronizer has been shown. The implementation is suggested to be applied for a short-range, underwater FSK acoustic modem. Furthermore, in [7] the transmitter has been implemented with multiple DSPs of type ADSP-TS101s and FPGA as the logical control. It has been proven experimentally that the signal transmitter satisfies requirements of signal transmission for OFDM in real-time underwater acoustic communication.

The paper contribution is to introduce a DSP-based embedded system for software defined radio (SDR). This embedded software is suitable for coherent receivers which are working in an online mode. By observing the performance of the system, the time's disposition of the frame is assigned to each each stage on the receiver accordingly. This shows that the proposed embedded receiver is performing adequately. Furthermore, in the proposed design, the reception of incoming frames and their processing phases are interfered without using the external memory. The remainder of the paper is organized as follows. In Section 2., the description of the proposed system is presented. Section 3. introduces the experimental results. Finally, conclusions are drawn in Section 4.

2. PROPOSED SYSTEM

Input bit sequence are encoded to have an immunity against channel errors. A simple code rate-$1/2$ nonsystematic convolutional (NSC) code and constraint length $K = 5$ is selected as a channel code. In order to permute the data, the encoder output is interleaved. The interleaver is consequently randomize error. Interleaved bits are transmitted using quadrature phase shift keying (QPSK) with a carrier frequency of 10 kHz and a symbol rate of 4 ksps.

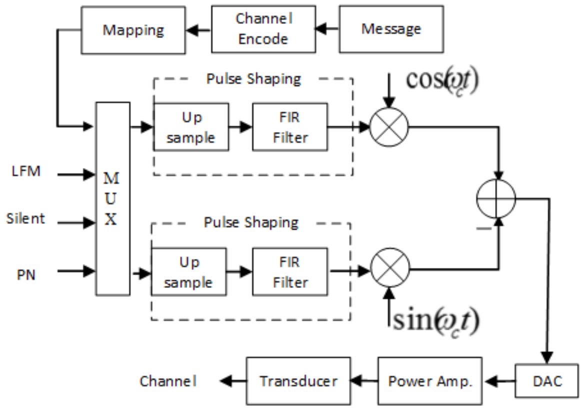


Figure 1. Block diagram of the transmitting system

2.1. Transmitter

The block diagram of the transmitting system is shown in Fig.1. In this system, we suppose that the input message is converted into bit stream to be transmitted to the other site using wireless channel. Fig.2 shows the frame structure, which is composed linear frequency modulation (LFM) of 10 ms that can be utilized for frame synchronization, silent period of 12.5 ms was used to mitigate interference between the LFM and the training sequence due to the multipath. In addition, Pseudo-noise (PN) sequence is transmitted to initiate carrier phase recovery in the second stage of synchronization. Fig.3 shows the real-time transmission of the frame presented in Fig.2. This figure depicts the entire frame structure which contains two silent periods, LFM and data.

2.2. Receiver

The receiver structure is illustrated in Fig.4. The received frame can be represented as:

$$y(k) = r(k) + w(k) = x(k) \otimes h(k) + w(k), \tag{1}$$

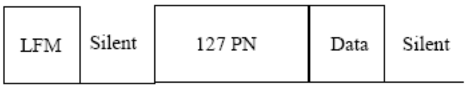


Figure 2. Transmitter frame structure

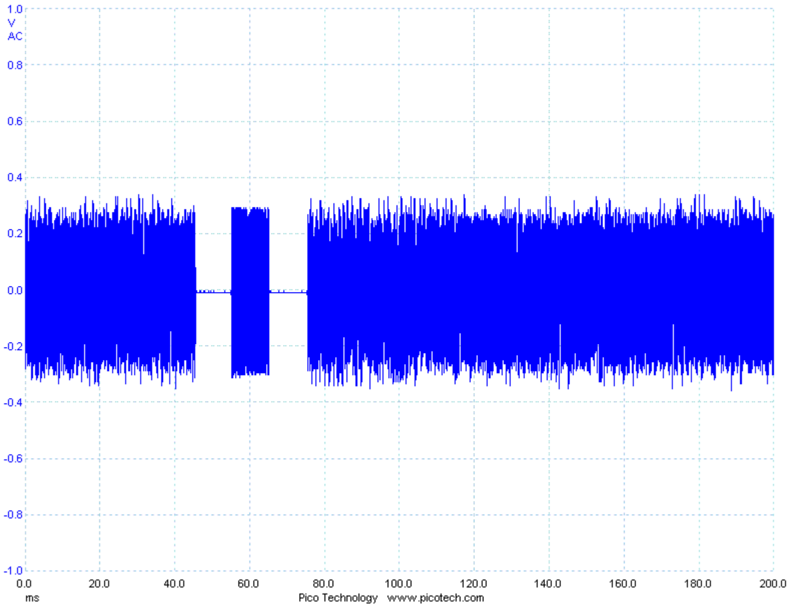


Figure 3. On-line transmitted signal

where $y(k)$ is the received signal at k time index, $h(k)$ represents the channel impulse response which is given as:

$$h(k) = \sum_{l=0}^{L-1} h_l(k)\delta(\tau_l), \tag{2}$$

where L is the number of taps, τ_l is the delay spread associated with the l-th tap and $h_l(k)$ is the complex-valued channel fading coefficient of the l-th tap, $w(k)$ represents the additive white Gaussian noise (AWGN) samples, $\otimes$ denotes the circular convolution operation. The transmitted samples $x(k)$ are convolved with $h(k)$ and the received passband signal is given as:

$$r(k) = \sum_{l=0}^{L-1} h(l)x((k-l))_N, \; k = 0 \ldots N-1. \tag{3}$$

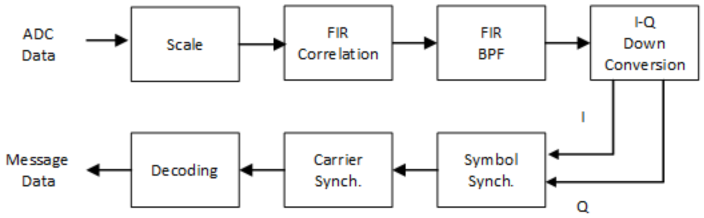


Figure 4. Block diagram of the receiver

In this paper, two stages of the synchronization are considered on the receiver side as an application for the proposed embedded system. Therefore, the decoder is implemented to to be used later. In the decoding stage, an optimized version of Veterbi decoder which is called BCJR is considered. Refer to [8] for more details on BCJR. Once the synchronization is achieved, a reliable data can be delivered to the decoding stage.

A. Packet synchronization

The received signal $r(k)$, is sampled in the analog-to-digital converter (ADC) and scaled. The LFM detection process which represents the header of the frame structure shown in Fig.2 is performed by comparing the threshold value with the cross correlation of the received signal $r(k)$ and known LFM t_n; the start of the packet. Equation 4 shows how to calculate the cross correlation [9]. The value of k that corresponds to maximum absolute value of the cross correlation is the packet timing estimate. If the cross correlation is greater than the threshold value, the frame synchronization is achieved.

$$\hat{t_s} = arg\ \underset{k}{max} \left| \sum_{n=0}^{C_l-1} r_{k+n} t_n^* \right| \tag{4}$$

In Equation 4, the length C_l of the cross correlation determines the performance of the algorithm. Larger values improve performance, however it increases the amount of computation required. In hardware implementations, FIR-correlation is adopted. This filter is conducting LFM with its flipped version. Fig.5 depicts the real-time cross correlation detection of the LFM peak signal. This signal is then filtered in the frequency band $[f_c - R/2, f_c + R/2]$ where R is the data rate. The sampling rate is chosen to be an integer multiple of carrier frequencyf_c for the sake of simple data manipulation.

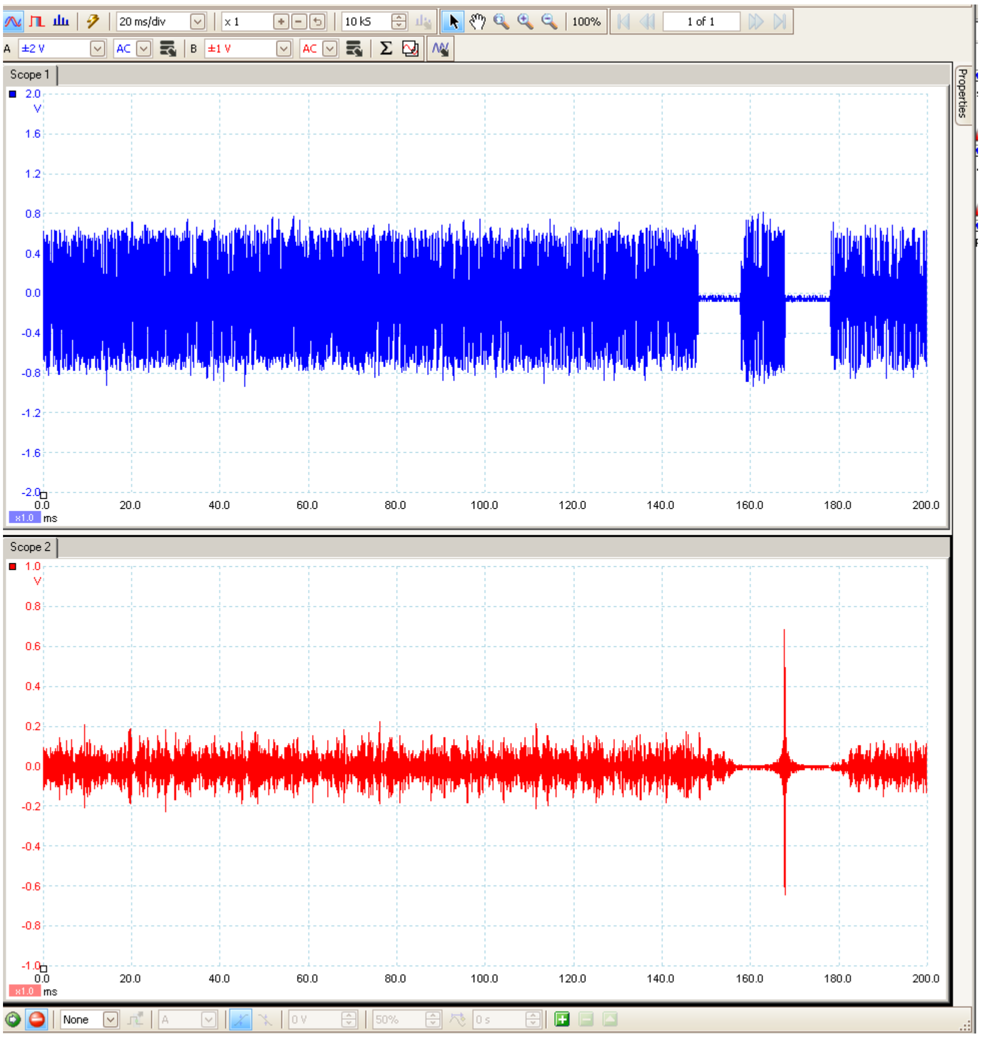


Figure 5. Peak detection of the received signal

B. Symbol Synchronization

Early-late based timing recovery is adopted. This symbol synchronization algorithm generates its error based on samples that are early and late compared to the ideal sampling point. We use a buffer of length $N = 12$ to store the matched filter output and measuring the energy in the left (early) and right (late) half of the buffer as:

$$E_{early} = \sum_{n=0}^{5}(x_I[n])^2 + (x_Q[n])^2, E_{late} = \sum_{n=6}^{11}(x_I[n])^2 + (x_Q[n])^2 \tag{5}$$

It is well known that the early and late samples are at different amplitudes. By comparing the amplitudes of the early and late samples, the timing error is generated. To eliminate this error, it is required to use a technique such as early-late synchronization which will produce better results and maintain perfect symbol timing. A delay line of one symbol time T_s is created and the total energy in the early and late samples will be compared. The sample to be used for later processing is the sample that lies in the middle of the early and late samples. Symbol timing is then adjusted in order to maintain approximately equal energy in the two halves then the center of the delay line corresponds to the optimal sampling point.

C. Carrier synchronization

It is noticed that I-Q constellation has a varying phase offset due to a carrier frequency and its phase mismatch between the transmitted and local carrier. Thus carrier recovery is necessitated for coherent receivers [10]. Decision-Directed carrier phase recovery via Costas loop is utilized as shown in Fig.6 and embedded in the DSP platform at the receiver side.

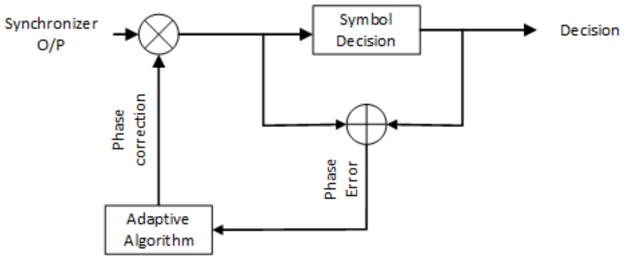


Figure 6. Decision-directed carrier phase loop

Fig.6 is represented in algorithm 1 which is implemented on the receiver side. In this algorithm, the adaptive step size is of importance in terms of convergence period. It must be varied to get satisfactory results and it is recommended to start with 0.01.

```
Result: Phase correction
initialization: Adjust scaling of I and Q symbol values to be close to 1.0;
Set Initially phase correction in the adaptive algorithm to 1.0+j0;
while no phase correction do
    Find $y_{I,Q}(n) = x(n) \times c(n)$; where $x(n)$, $c(n)$ is the synchronizer O/P and phase correction,
    respectively;
    Make decision on $y_{I,Q}(n)$ to obtain $\Re\{d(n)\}, \Im\{d(n)\}$;
    Calculate the phase error $e_{I,Q}(n) = d_{I,Q}(n) - y_{I,Q}(n)$;
    Update phase correction in the adaptive algorithm;
    Vary the adaptive step size $\mu$;
    where, the operator $\Re$, $\Im$ represents the real and imaginary part, respectively.
end
```

Algorithm 1: Decision-directed carrier recovery algorithm via costas loop for QPSK

2.3. DSP Implementation

Von Neumann architecture uses single memory for both data and instruction, however, this type of architecture [11] dissipate more power than conventional DSP architecture. As we have multistage synchronization in the proposed receiver, it is important avoid any wait state in the system. This architecture is offered by Super Harvard Architecture (SHARC), where the address lines of data and instructions are split.

In the proposed system, the ADSP-21364 SHARC EZ-KIT LITE has been selected. It consists of a 333 MHz SHARC DSP with an audio codec which provides 2 24-bit ADC inputs and 8 24-bit DAC outputs at a maximum sampling rate of 96kHz. It also provides serial peripheral interface (SPI) link. To obtain maximum processor utilization, double buffering technique is used. The SHARC DSP has a direct memory access (DMA) coprocessor, which reads/writes a block of data from/to memory while the processor core works on another block. The operating system was Visual DSP++. ADC data, in 24 bits signed integer format ∓ 8388608, must first be converted to floating point representation.

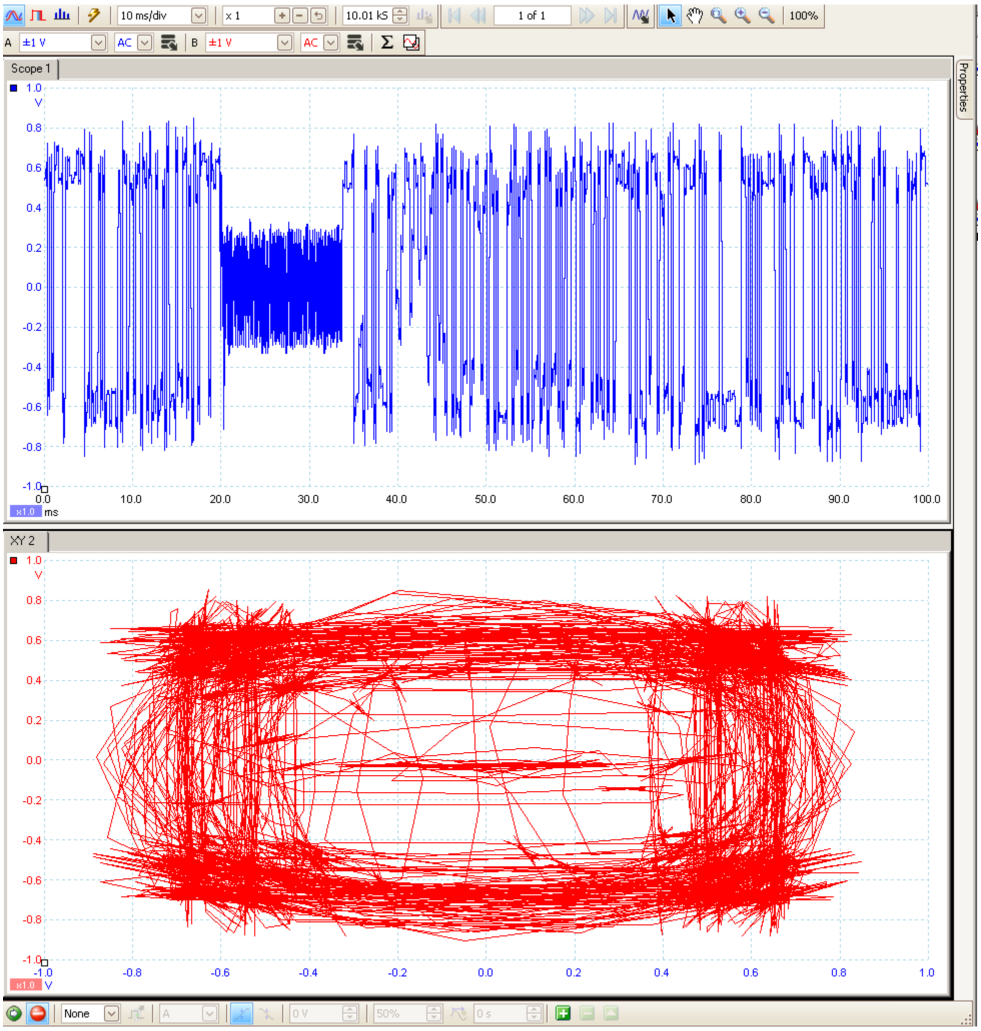
3. EXPERIMENTAL RESULTS

The proposed system is investigated and emulated in the lab with a wire link between transmitter and receiver in order to calibrate the system. In Fig.7, it can be seen on the upper plot that there is a cut in the reception represented by the concentrated area of the QPSK signal. This concentrated area visualizes the state of the signal when the microprocessor comes out of work in real time and consequently affects the constellation outputs as shown in the lower plot which is the eye diagram of the received QPSK signal. This is due to a long execution time at the receiver side especially on the decoding stage, where the time is critical and consequently the DSP ran out of the real time mode. Such type of run time error is subtle and can not be detected quickly. Thus, in favor of Fig.7, the problem has been identified and manipulated by adopting a piplining and double buffering through an interrupt programming.

Fig.8, depicts obtained real time constellation of the proposed embedded QPSK receiver. The adaptive step size μ was 0.005 and zero error rate. It is shown from this figure that the the embedded system for multistage synchronization was achieved. An interesting point in these requirements is that the onchip memory of size 3Mbit is determined, which is provided by the utilized SHARC processor (including interleaver) equivalent to $21N_f$ symbols. In order to evaluate the proposed system, obtained result, in terms of memory utilization, is compared with [12] and shows that our system outperforms their system in one block memory requirement, where they used $22N_f$. It is worth mentioning that there was insufficient memory space while buffering the entire frame in both transmitter and receiver. Data memory (DM) space and program memory (PM) in cooperation with heap are exploited to tackle these constraints of memory. In terms of the number of operations, Table 1 demonstrates the number of operations (multiply, divide, add, subtract) of each stage of the synchronization, but the read and write operations has not been considered.

Table 1. Receiver operations

Receiver stages	No.Operations
BPF+LPF	179
Synchronization	388



10 ms/div x 1 10.01 kS 1 of 1 100%

A ±1 V AC B ±1 V AC

None A 0 V 50% 0 s

Figure 7. Out of real-time reception

4. CONCLUSION

The focus of this paper was on the design and implementation of an embedded system for software defined radio using SHARC DSP. The proposed embedded system has been investigated online through an implementation of QPSK synchronization schemes and convolutional decoding. It was assessed in the laboratory to calibrate the operation. This paper presented a technique to tackle both the effect of critical time and memory constraints issues. In conclusion, the pipelining and double buffering is useful to gain processing time and should be considered. The interleaver length is crucial in selecting the DSP memory where it requires to buffer the whole frame. Obtained results show that the implementation is robust and working online successfully and can be considered in many embedded systems.

REFERENCES

[1] A. Zakwan, et al.,"Implementation of Algorithm for Vehicle Anti-Collision Alert System in FPGA,"*International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 7, pp. 775-783, April. 2017.

[2] M. Abdurohman and A. Sasongko,"Software for Simplifying Embedded System Design Based on Event-Driven Method,"*International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 5, pp. 491-502, Jun. 2015.

[3] A. Devices, *LaTeX – Embedded Processor and DSP Selection Guide*, Analog Devices, 2005.

[4] Yan, Z., Huang, J. and He, C., "Implementation of an OFDM underwater acoustic communication system

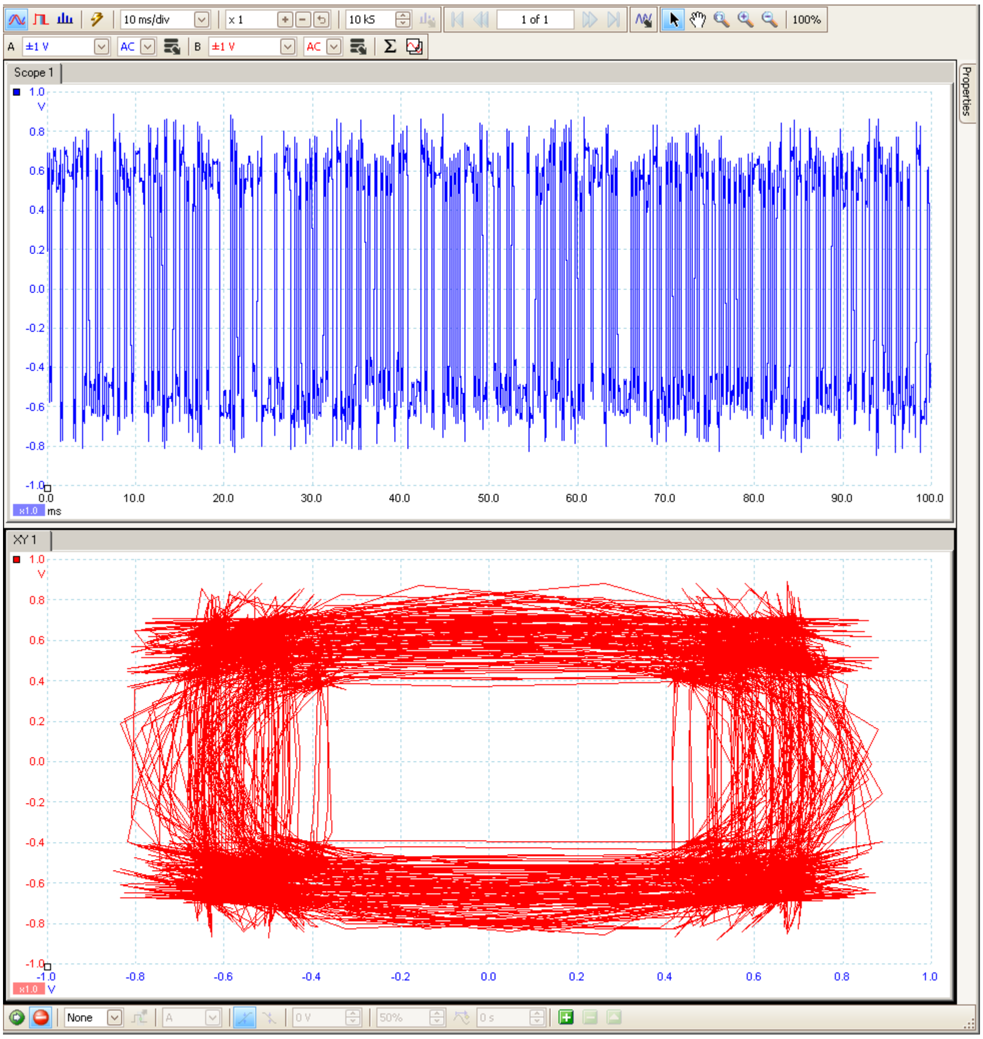


Figure 8. On-line synchronized reception of the QPSK signal

on an underwater vehicle with multiprocessor structure," *Frontiers of Electrical and Electronic Engineering*, vol. 2, pp. 151-155, 2007.

[5] M.R. Soleymani, Yingzi Gao and U. Vilaipornsawai, *LaTeX – Turbo Coding for satellite and wireless Communications*, Springer Netherlands, 2002.

[6] Ying Li, et al., "Hardware Implementation of Symbol Synchronization for Underwater FSK,"*Sensor Networks, Ubiquitous, and Trustworthy Computing, 2010. SUTC IEEE 2010*, 2010, pp. 82- 88, 2010.

[7] L. Kaizhuo, et al., "Design and implementation of underwater OFDM acoustic communication transmitter," in *Audio, Language and Image Processing Conference, 2008. ICALIP 2008. International Conference*, 2008, pp. 609-613.

[8] L.Bhal et al.,"Optimal decoding of linear codes for minimizing symbol error rate,"*IEEE Transactions on Info. Theory*, Vol. 20, pp. 284-287, Mar. 1974.

[9] J. Heiskala and J. Terry *LaTeX –OFDM Wireless LANs: A theoretical and Practical guide*, SAMS, ISBN: 0672321572.

[10] J. G. Proakis and M. Salehi *LaTeX –Digital Communications*, McGraw-Hill, Fifth Edition, 2008.

[11] W. H. Park, M. H. Sunwoo, and S. K. Oh, "Efficient DSP architecture for Viterbi decoding with small trace back latency,"*Asia Pacific on Circuits and Systems Conference, IEEE 2004*, 2004, pp. 2813-2818 , 2004.

[12] Or, Y., Kutz G., Chass A., Gubeskys A., Pollak E.,"Iterative decoding algorithms for real time software implementation in wireless communication systems," *IEEE Conf.,VTS*,pp. 1884 - 1888, vol.3, 2001.