Implementation of coherent IEEE 802.15.4 receiver on software defined radio platform

Uroš Pešović, Dušan Gliech, Peter Planinšič, Zoran Stamenković, Siniša Randjić

***Abstract* — Coherent receivers provide better demodulation characteristics then non-coherent receivers especially with higher bit error probability signals. They require synchronization with received signal using preamble which is transmitted at beginning of IEEE 802.15.4 packet. Software defined radio doesn't allow adjustment of local oscillator frequency using Costas Loop circuitry, so synchronization need to be performed by software synchronization in baseband. This paper represents implementation of coherent IEEE 802.15.4 receiver on software defined radio.**

***Keywords* — Coherent receiver, IEEE 802.15.4, Software defined radio**

I. Introduction

Coherent receivers provide better demodulation properties in comparison with non-coherent receiver for signals with higher bit error probability. In order to successfully receive useful information, any receiver needs to be synchronized with transmitter. The synchronization is important part of any digital receiver, and its role is to enable lock to incoming data transmitted from appropriate wireless transmitter. Synchronization is done at following levels in receiver:

1. Carrier recovery
2. Symbol timing recovery
3. Frame synchronization

In order to successfully demodulate received signals, any receiver needs to operate on same carrier frequency as local oscillator in transmitter. Even when they are set to same center frequency, frequency of local oscillators can vary up to ±40 ppm for IEEE 802.15.4 compliant transceivers [1]. Also, if receiver or transmitter is mobile, frequency can shift due Doppler effect.

Recent introduction of SDR (Software Defined Radio) platforms enabled researchers a fast way of prototyping radio communication equipment. Digital carrier synchronization is usually the only option in case of software defined radio platforms, which allow only processing of the baseband signals. Prior works present implementation of IEEE 802.15.4 non-coherent receivers on SDR in papers [2], [3] and [4].

Non-coherent receivers, require just accurate frequency adjustment to successfully demodulate received signals. Such IEEE 802.15.4 receivers use differential decoding by tracking phase difference in order to decode received information. Differential decoding doesn't require phase synchronization to recover modulated symbols, since phase offset is canceled by subtraction of phase of two successive symbols. Synchronization is only required on symbol timing level. Such receiver structures are quite simple but their disadvantage is higher error probability introduced by demodulation process itself. Previous work

Coherent receivers, requires phase and frequency synchronization in order to successfully demodulate received signals. If receiver operates on same frequency as transmitter there is constant carrier phase error $\Delta\varphi$ which corresponds to a rotation of the received constellation. Additionally, if receiver doesn't operate on exact carrier frequency, constant carrier frequency error $\Delta\omega$ corresponds to time-varying rotation of the received constellation.

$$\overline{A_k} = e^{j(\Delta\omega \cdot kT + \Delta\varphi)} A_k$$

Traditionally, carrier synchronization is performed using analog or hybrid analog/digital solutions by adjusting phase and frequency of the receiver's local oscillator using phase-locked loops (PLLs) coupled with Costas loop [5]

Digital solutions perform carrier synchronization after demodulation. Such solutions, allow use of free-running local oscillators for demodulation and frequency conversion. Constant carrier frequency and phase error are estimated after demodulation and if the parameters can be estimated with sufficient accuracy, these effects can easily be compensated by a complex multiplier in baseband processing, prior to detection. Digital carrier synchronization provide better jitter performance and/or tracking speed, as well as reduced power consumption and cost in comparison to traditional analog or hybrid solutions.

In this paper, we describe the implementation of IEEE 802.15.4 coherent receiver for 2.4 GHz band on a software defined radio platform.

II. IEEE 802.15.4 wireless transceivers

IEEE 802.15.4 standard is designed for low power, low cost battery operated devices, which are used in wireless sensor network applications. This standard defines physical and medium access layer. Physical layer provides

Uroš M. Pešović and Siniša S. Randjić are with the Faculty of Technical Sciences Čačak, University of Kragujevac, Svetog Save 65, 32000 Čačak, Serbia (tel: +38132302721; e-mail: uros.pesovic@ftn.kg.ac.rs, sinisa.randjic@ftn.kg.ac.rs).

Dušan Gliech and Peter Planinšič are with the Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova 17, 2000 Maribor, Slovenia (tel: +38622207128; e-mail: dusan.gleich@mb.si, peter.planinsic@mb.si).

Zoran Stamenković is with the IHP, Im Technologiepark 25, 15236 Frankfurt (Oder), Germany (tel: +493355625726; e-mail: stamenko@ihp-microelectronics.com)

several frequency bands, where 2.4 GHz ISM band, with 11 channels, is most widely used. IEEE 802.15.4 compliant transceivers employ transmitter and receiver on the same chip. Typical functional structure of such transceiver is separated into transmission and reception part (Fig. 1).

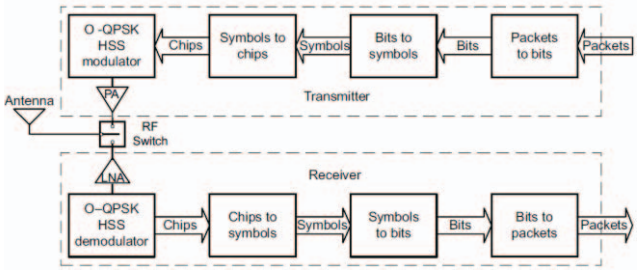

Fig. 1. Structure of IEEE 802.15.4 transceiver

Packet which needs to be transmitted is partitioned into groups of four bits, which are referred as symbol words. Each symbol word is spreaded using Direct-sequence spread spectrum (DSSS) signal spreading technique into appropriate orthogonal sequence of 32 chips, as shown in Table 1. Data rate of 2.4 GHz band is 250 kbps or 2 Mchips/s. Predetermined chip sequences are chosen in order to be maximally dissimilar from each other. Chip sequences are divided into two groups of sequences (C_0 to C_7 and C_8 to C_{15}), whose sequences are created by right cyclic shifting of their respective previous chip sequences for four chip positions. Chip's sequences of the other group are created through inversion of odd-indexed chip values of the respective chip sequence from the opposing group (C_0 to C_8; C_1 to C_9; . . . ; C_7 to C_{15}).

TABLE 1: IEEE 802.15.4 SYMBOL TO CHIP MAPPING

Symbol	*Chip sequence*	*Symbol*	*Chip sequence*
S_0 (0x0)	C_0 (0x744AC39B)	S_8 (0x8)	C_8 (0xDEE06931)
S_1 (0x1)	C_1 (0x44AC39B7)	S_9 (0x9)	C_9 (0xEE06931D)
S_2 (0x2)	C_2 (0x4AC39B74)	S_{10} (0xA)	C_{10} (0xE06931DE)
S_3 (0x3)	C_3 (0xAC39B744)	S_{11} (0xB)	C_{11} (0x06931DEE)
S_4 (0x4)	C_4 (0xC39B744A)	S_{12} (0xC)	C_{12} (0x6931DEE0)
S_5 (0x5)	C_5 (0x39B744AC)	S_{13} (0xD)	C_{13} (0x931DEE06)
S_6 (0x6)	C_6 (0x9B744AC3)	S_{14} (0xE)	C_{14} (0x31DEE069)
S_7 (0x7)	C_7 (0xB744AC39)	S_{15} (0xF)	C_{15} (0x1DEE0693)

Chips are separated into even and odd index chips in order to be modulated using two orthogonal phases. Even indexed chips are modulated onto in-phase I, while the odd-indexed chips are modulated onto quadrature phase Q. Both phases are shaped by half sine pulses, chips representing logical one are shape by positive half sine pulse, while chips representing logical zero are shaped by negative half sine pulse. Quadrature Q phase is delayed by one chip period and added to I phase which results in continuous phase modulation, known as Offset QPSK with the Half Sine pulse Shaping (O-QPSK HSS) modulation. Modulated baseband signal is modulated on the 2.4 GHz carrier, amplified and transmitted by one of the antennas.

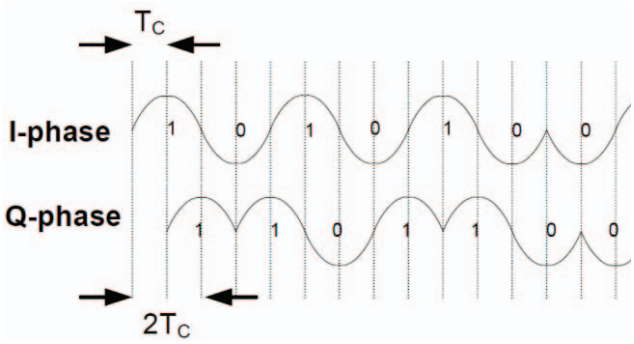

Fig. 2. O-QPSK HSS modulation

Format of physical layer frame, shown in Fig. 3, consists of synchronization header SHR, packet header PHR and PHY payload. Synchronization header is used for receiver synchronization and it consists of Preamble field and Start-Frame-Delimiter. Preamble represents a fixed pattern of symbol zero S_0, which is transmitted eight times on which receiver needs to lock on. Start-Frame-Delimiter marks start of packet and contain two symbol words S_{10} and S_7 ($A7_{HEX}$). Packet header defines length of PHY payload in bytes, which can be from 0 to 127 bytes long.

		Octets		
		1		variable
Preamble	SFD	Frame length (7 bits)	Reserved (1 bit)	PSDU
SHR		PHR		PHY payload

Fig. 3. Format of IEEE 802.15.4 physical frame

Architecture of IEEE 802.15.4 receiver is much complex then transmitter. During reception, wireless signals received by an antenna, are amplified and processed by an analog front-end. After processing, they are brought to O-QPSK HSS demodulator, where information is extracted in a form of digital chips. Depending on the quality of the received wireless signal some of the chips can be decoded incorrectly. When received chip sequence is formed, it's cross-correlated to predefined chip sequences, and symbol of most similar sequence if fed to next stage. Symbol words are translated to bits, and finally, bits are grouped into packets, which are forwarded to the upper protocol layer.

III. SOFTWARE RADIO PLATFORM

SDR is reconfigurable radio platform which uses software to perform some of the signal processing in a receiver and transmitter. It enables creation of physical layer algorithms with live interaction with host computers. It can be used for transmission and reception of signals, monitoring spectrum of received signal, measuring bit error rate (BER) and etc. It's usually based on programmable FPGA chip coupled with analog front end which modulates and demodulates radio signals. FPGA enables reconfigurability as well as high processing power since all computations can be carried out by hardware.

National Instruments USRP-2921 [6], is software programmable radio platform designed for wireless communications in 2.4 GHz ISM band. A simplified block diagram of NI USRP-2921 is presented in Fig. 4. USRP-2921 has two antennas which can be selected using internal

radio switch. Received radio signals, are amplified and mixed with matching RF signal in order to extract baseband signal. Baseband signal is composed of I/Q components which are filtered and sampled by twin 100 MS/s, 14-bit analog-to-digital converter (ADC). The digitized I/Q data is down converted to user specified sample rate and passed to the host computer over a standard Gigabit Ethernet connection. Maximum sampling rate is 25 MS/s in full resolution mode, or 50 MS/s in 8-bit resolution mode. During transmission, I/Q baseband component generated by the host computer are fed to a USRP-2921 over Gigabit Ethernet. The USRP hardware interpolates the incoming signal to 400 MS/s using a digital up conversion (DUC) process and then converts the signal to analog with a dual-channel, 16-bit digital-to-analog converter (DAC). The resulting analog signal is then mixed up with carrier and transmitted by antenna. The NI USRP-2921 connects to a host computer via Gigabit Ethernet Interface using appropriate USRP driver and can be integrated into Matlab and Labview applications.

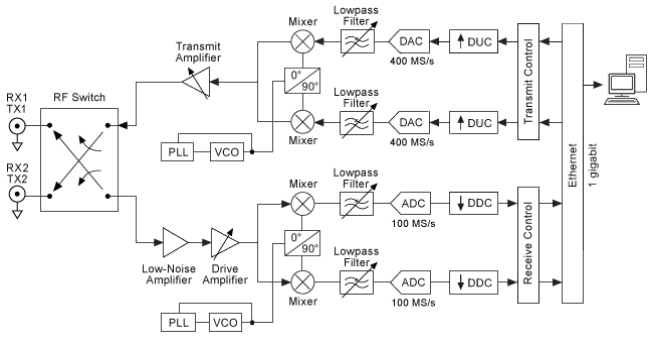

Fig. 4. NI USRP-2921 system block diagram

The USRP driver software provides functions for hardware/software configuration with tools for opening/closing sessions and performing read/write operations on the software defined radio platform. Using USRP driver, live signals are transmitted to host computer, where they can be processed. Since huge quantity of received information only slight data processing can be performed by computer in order to enable real-time operation. NI-USRP also enables that some of user defined computations can be carried on FPGA inside software defined radio in order to enable real time operation.

IV. Implementation and Results

Coherent receiver is developed as MATLAB application, with I/Q baseband signal fed by USRP 2921 SDR. Developed experiment consists of one commercial IEEE 802.15.4 transmitter which transmits random packets and coherent receiver implemented on SDR. SDR receives I/Q data sampled at 20 MS/s with 16 bit resolution, where each symbol is represented by 20 samples. Received samples are transferred via Gigabit Ethernet interface to MATLAB application on host computer. Detection of new packet can be determined either by monitoring channel energy level or by detection of preamble symbol S_0. Received I/Q signals presented in Fig. 5. are not synchronized to expect I/Q signals of preamble symbol S_0. If decoding is performed without synchronization, preamble symbol S_0, will be decoded with large number of chip errors. Expected and decoded preamble symbol S_0 chip sequences are represented by blue and red chip streams in Fig. 5.

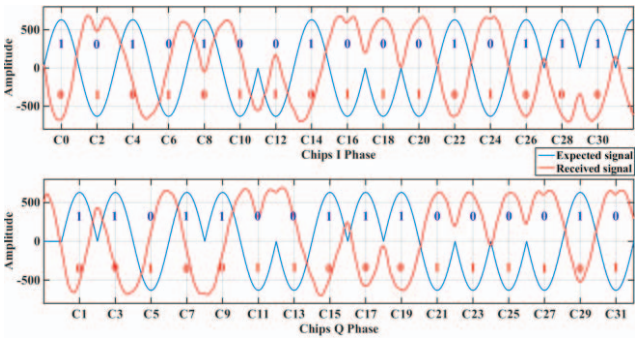

Fig. 5. I/Q diagram of expected and unsynchronized received signal

Timing synchronization is achieved by detection of start of preamble symbol S_0 in received signal by cross correlation phase of received signal with phase of expected preamble symbol sequence S_0 which is shown in Fig. 6.

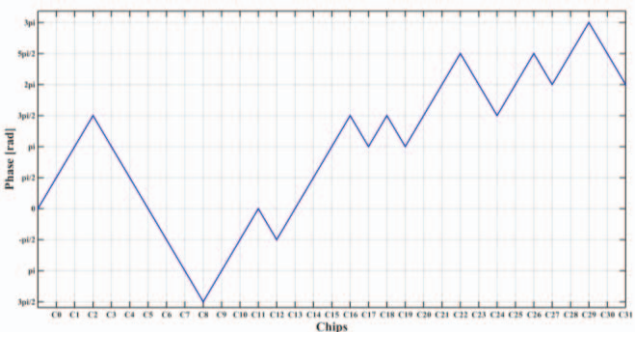

Fig. 6. Phase of preamble symbol S_0

If phase of received signal is compared to phase of expected preamble sequence, as shown in Fig 7., it's clear that receiver is not synchronized to incoming signal, both by frequency and phase. When phase of the received signal is subtracted from phase of expected preamble symbol sequence S_0, we can see that phase offset follows linear function, where coefficients $\Delta\varphi$ and $\Delta\omega$ can be are extracted using polynomial approximation. We can see that time-varying frequency drifts of local oscillator's don't affect that much linearity of phase difference between transmitter and SDR.

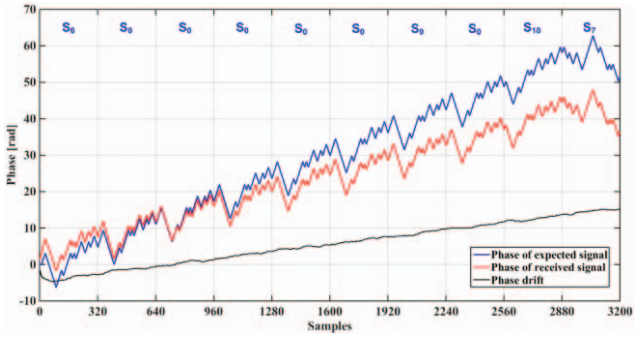

Fig. 7. Phase drift of received signal compared to expected signal

Using Δφ and Δω coefficients received signal is synchronized by multiplication with complex conjugate in order to remove phase drift.

$$A_k = \overline{A_k}$$

Satisfactory synchronization is achieved even for first preamble symbol, as shown in Fig. 8., so it's successfully decoded without any chip error. After first iteration of synchronization, other preamble symbols are successively subtracted from phase of expected preamble symbol S_0, each fine tuning Δφ and Δω parameters in order to fully synchronize on the received signal.

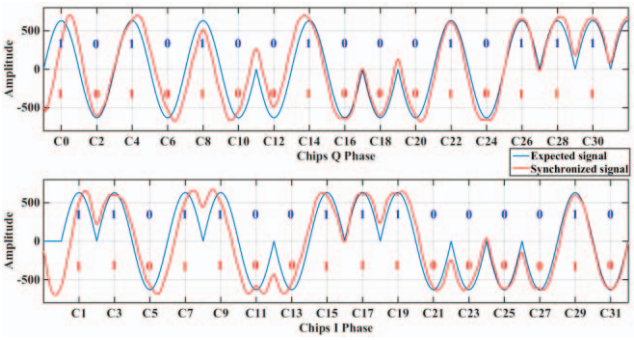

Figure. 8. I/Q diagram of first synchronized preamble symbol

Since all computation is performed on host computer, such implementation of coherent receiver cannot operate in real time, since block of data is firstly received as whole and then processed. Real time operation of such coherent receiver can be achieved if all computation is performed by FPGA logic on SDR platform, which will be focus of our further work.

V. Conclusion

This paper represents implementation of IEEE 802.15.4 coherent receiver in 2.4 GHz band on software defined radio platform. Presented digital synchronization is much simpler solution compared to complex analog synchronization solutions which employ Costas Loop. Presented digital synchronization solution can be implemented on FPGA on SDR platform in order to achieve real time operation of IEEE 802.15.4 coherent receiver.

Acknowledgment

The work presented in this paper achieved thanks to technical support from Laboratory of Signal Processing and Remote Control, form University of Maribor, Slovenia and funded by grant TR32043 for the period 2011-2015, by the Ministry of Education and Science of the Republic of Serbia.

References

[1] IEEE P802.15 Working Group, "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)". *IEEE Computer Society*. New York, USA, 2003

[2] U. Pesovic, D. Gliech, P. Planinsic, Z. Stamenkovic, S. Randjic, "Implementation of IEEE 802.15.4 transceiver on software defined radio platform", *TELFOR 2012*, Belgrade, Serbia, 2012, pp 376-379.

[3] A. Di Stefano, G. Fiscelli, C.G. Giaconia, "An FPGA-Based Software Defined Radio Platform for the 2.4GHz ISM Band", *Research in Microelectronics and Electronics,* 2006, pp 73-76,

[4] S. Knauth, "Implementation of an IEEE 802.15.4 Transceiver with a Software-defined Radio setup", *Lucerne University of Applied Sciences*,

[5] A. Massouri, T. Risset, "FPGA-based Implementation of Multiple PHY Layers of IEEE 802.15.4 Targeting SDR Platform, SDR-WInnComm", *Wireless Innovation Forum*, 2014

[6] NI, "NI USRP-2920, NI USRP-2921, Universal Software Radio Peripherals", National Instruments