Proposed Interleaver Implementation with Arduino Micro Controller and its Performance vs LTE Interleaver

Amjad Ali Jassim* **Dr. Wael A.H.HADI****

**Department of electrical engineering/ University of Technology*

Abstract: The interleaver design take its role in parallel turbo code performance. In LTE "Long Term Evolution" the interleaver used with complex design. In this paper introduce design interleaver with simple mathematical model. And implement the design using Arduino UNO microcontroller as simple practical approach. The designed interleaver performance take its comparison with respect standard LTE interleaver using MATLAB. The proposed implementation show enhancement in system performance with three modulation types including QPSK, 16-QAM and 64-QAM which represent typical LTE modulation schemes.

1. Introduction:

The interleaver in its simple concept in communication applications represent treatment to burst error in transmitted signal over noisy channel. The error correcting codes take advantage of interleaver to improve overall system performance. Since the interleaver in its operation act as rearrange the input data sequence before encoding process. The encoded sequence after modulated transmitted over noisy channel. At receiver side a reciprocal operation starts by demodulate the received signal. At almost the received signal suffer from errors as single error or burst errors. The decoder for error correcting code well treat as possible the stand alone errors in received frame, but the burst errors represent more challenge to decoder. The interleaver comes here to break as possible this burst errors and convert it to single or standalone errors that decoder has more chance to correct them. At receiver side inverse operation take its role to restore the correct data sequence, such process so called deinterleaver process. In turbo code applications, the interleaver located in between two parallel convolutional encoders. And reciprocal operation taken at receiver side. The communication system shown in Figures 1 and 2.

There are several types of interleavers such as block interleaver [1], convolution interleaver [2], 2D interleaver [3], chaotic interleaver [4], but all share the same concept that reduce effect of burst errors on communication system. The interleaver design attends to achieve performance enhancement without adding more system complexity.

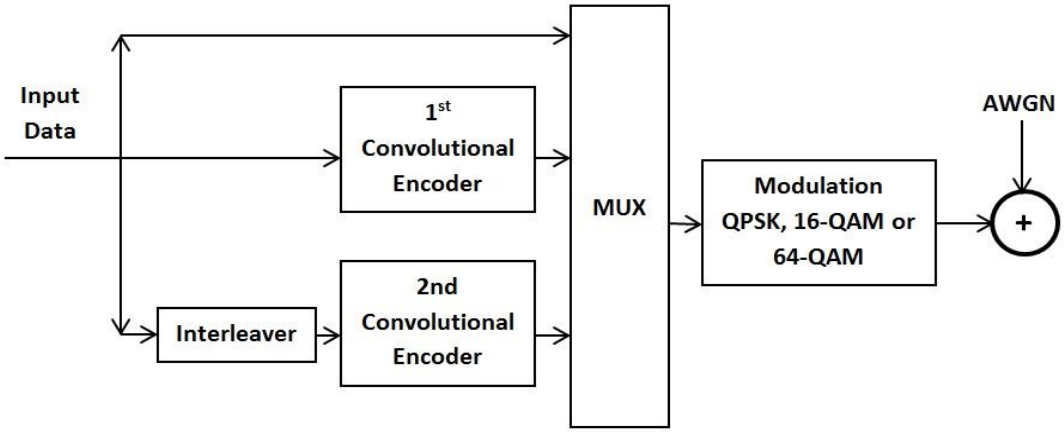

Figure 1. Turbo system transmitter with interleaver, with three possible modulation types as typical to LTE modulation schemes.

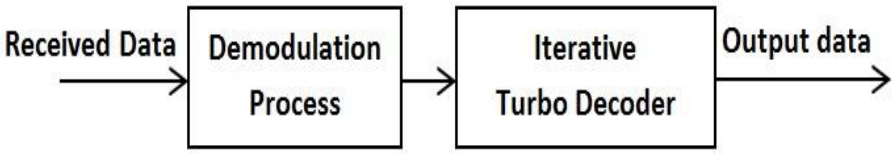

Figure 2. Turbo system receiver with iterative decoding.

2. LTE Interleaver:

3.

In LTE (Long Term Evolution) there are specific type of interleaver used as standard in LTE design, such interleaver known as QPP (Quadratic Primitive Polynomial). The LTE interleaver is based on a simple Quadratic Polynomial Permutation (QPP)

scheme. In its operation the interleaver permutes the indices of the input bits at encoder input. The relationship between the output index *p(i)* and the input index *i* is described by the following quadratic polynomial expression [5]:

$$p(i)=(f_1 \cdot i + f_2 \cdot i^2) \bmod (k) \ldots\ldots (1)$$

where *k* is the size of the input block and f_1 and f_2 are constants that depend on the value of *k*. The LTE allows 188 different values for the input block size *k*. The smallest block sizeis 40 and largest is 6144. These block sizes and the corresponding f_1 and f_2 constants aresummarized in Reference [6]. So in LTE interleaver QPP we need a predefined two values f_1 and f_2, such as storing 188 item in memory to recall and compute permutation index. This add more complexity to the system rather than direct calculation of permutation index.

4. Proposed interleaver:

The proposed interleaver mathematical model introduced as three equations as follows:

$$P(i) = \begin{cases} P(1) = 1 \\ P_{n+1}(i) = P_n(i) + step + 1, \text{ at } P_{n+1}(i) <= \text{Frame length} \\ P_{n+1}(i) = P(1) + 1 + \delta_n \quad , \text{at } P_{n+1}(i) > \text{Frame length} \\ \text{Update } \delta_{n+1} = \delta_n + 1 \quad , \delta_0 = 0 \quad \text{Initial at zero} \end{cases}$$

For example, illustrate its operation. Let input data of frame length of 14 and step size of 3, the permuted index calculated as follows:

P(1) = 1

P(2) = P(1) + (step =3) + 1 = 1 + 3 + 1 = 5

P(3) = P(2) + 3 + 1 = 5 + 3 + 1 = 9

P(4) = P(3) + 3 + 1 = 9 + 3 + 1 = 13

P(5) = P(4) + 3 + 1 = 13 + 3 + 1 = 17 which is > frame length of 14 then

P(5) = P(1) + 1 + 0 = 1 + 1 = 2 update $\delta_1 = 0 + 1 = 1$.

P(6) = P(5) + 3 + 1 = 2 + 3 + 1 = 6

P(7) = P(6) + 3 + 1 = 6 + 3 + 1 = 10

P(8) = P(7) + 3 + 1 = 10 + 3 + 1 = 14

P(9) = P(8) + 3 + 1 = 14 + 3 + 1 = 17 which is > frame length of 14 then

P(9) = P(1) + 1 + ($\delta_1 = 1$) = 1 + 1 + 1 = 3 update $\delta_2 = \delta_1 + 1 = 1 + 1 = 2$

P(10) = P(9) + 3 + 1 = 3 + 3 + 1 = 7

P(11) = P(10) + 3 + 1 = 7 + 3 + 1 = 11

P(12) = P(11) + 3 + 1 = 11 + 3 + 1 = 15 which is > frame length of 14 then

P(12) = P(1) + 1 + ($\delta_2 = 2$) = 1 + 1 + 2 = 4

P(13) = P(12) + 3 + 1 = 4 + 3 + 1 = 8

P(14) = P(13) + 3 + 1 = 8 + 3 + 1 = 12.

So we could summarize the permutation action compared with original data sequence as shown in Table 1.

Table 1. Step interleaver permuted sequence, frame length 14 and step of 3

Original sequence	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Permuted sequence	1	5	9	13	2	6	10	14	3	7	11	4	8	12

The resulted permuted sequence will be used by communication system encoder as reference to permute original binary data locations in input data frame. And

restore their locations again at receiver side to get right data sequence after final iterative decoding process.

5. Step interleaver implementation:

The LTE interleaver support different input data frames length, that starts from 40 up to 6144. So for hardware implementation of proposed step interleaver should be meat the ability of QPP interleaver that used as standard interleaver with LTE application. That leads to carful design in its algorithm to support such variable length.

In implementation chose simple Arduino UNO micro controller. Even it is limited with its capability in memory support as shown in Table 2 that describe technical specifications. The work successes in overcome such limits. And put the algorithm of step interleaver in practical use by programming Arduino UNO in manner not only work in fixed frame length but also support run the algorithm at each frame length (from 40 up to 6144) which match the ultimate frame length in LTE.

Table 2. Arduino UNO technical specification [٧].

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA

Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

To overcome memory limitation of Arduino UNO, SD card used with suitable connection with Arduino UNO. The connection between them listed in Table 3.

Table 3. SD card connected with Arduino UNO [8].

GND: ground (0V)
VCC: power supply (5V)
MISO (Master Input Slave Output) connected to Arduino UNO pin 12
MOSI (Master Output Slave Input) connected to Arduino UNO pin 11
SCK (Master Serial Clock) connected to Arduino UNO pin 13
CS (Slave Select) hint : you can connect this pin to any Arduino digital output

The Arduino UNO micro controller after program algorithm, it uses serial port to transmit data to user interface and save the calculated permuted sequence as text file in SD card. The stored information in text file will be exported to MATLAB, a full simulation function in m file that simulate communication system will import the stored information and use it to calculate system performance and plot the results as comparison between system using standard QPP interleaver and proposed step interleaver. the process illustrated in Figure 3.

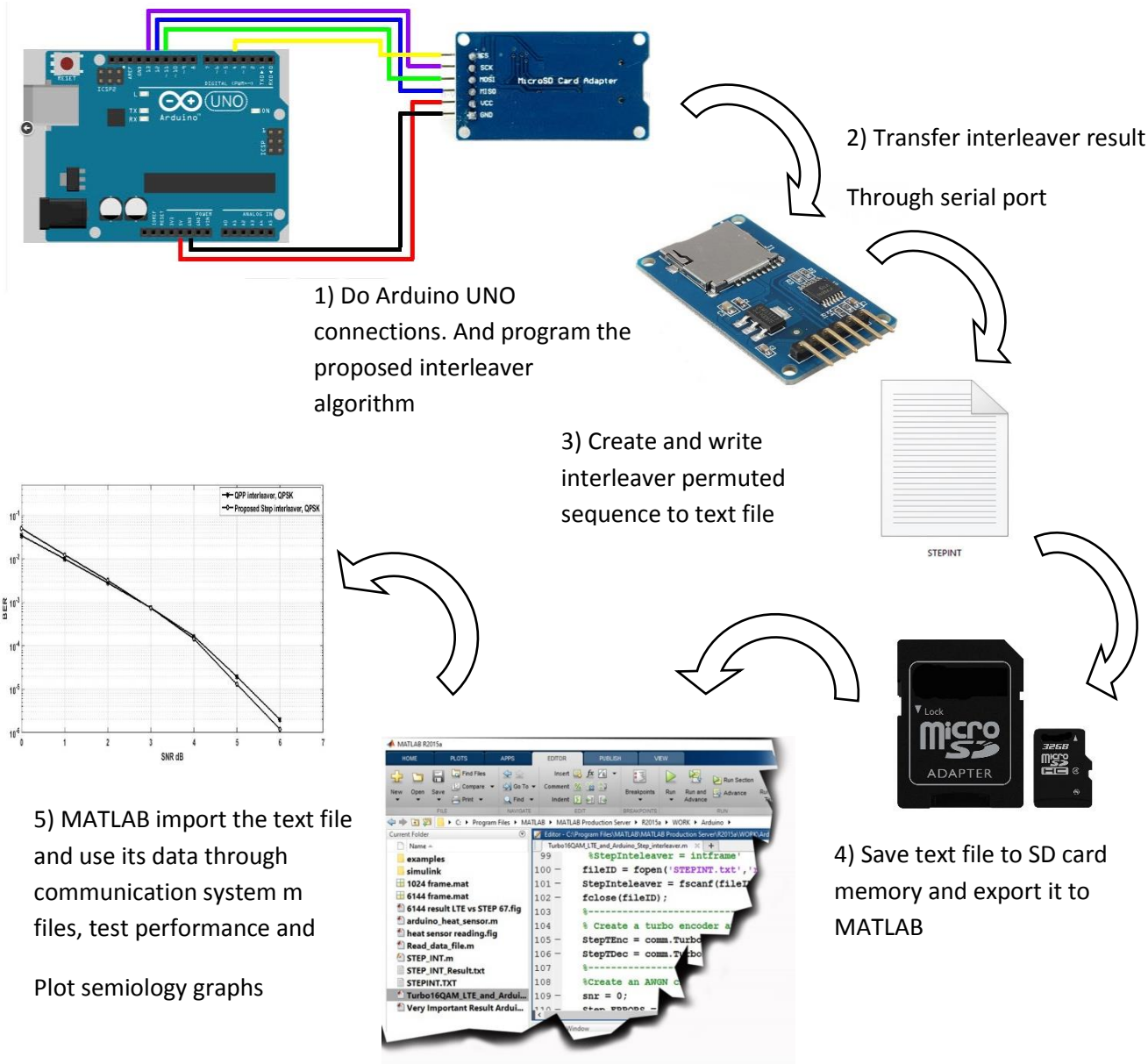

Figure 3. illustrate proposed step interleaver implementation and data process.

6. Simulation results

a communication system based on Parallel turbo code is considered with two types of interleavers. The system encoder consists of two parallel convolutional encoders of rate 1/2, with generating polynomial defined in octal by (5, 7) and maximum free distance of 5 and constraint length 3. The system receiver processes an iterative decoding, adjusted at fixed 4 iterations. the simulation study case changing between two types of interleavers first QPP at frame length 6144 the f_1 = 263 and f_2 = 480. This values of f_1 and f_2 adjusted as standard, where for each frame

length start from 40 to 6144 there are 188 standard values of f_1 and f_2.and second the proposed step interleaver at input frame length of max LTE limit 6144 with step size 67. The simulation takes three types of modulation schemes, and test each case individually for QPSK, 16-QAM and 64-QAM.

Figures 4,5 and 6 show simulation results.

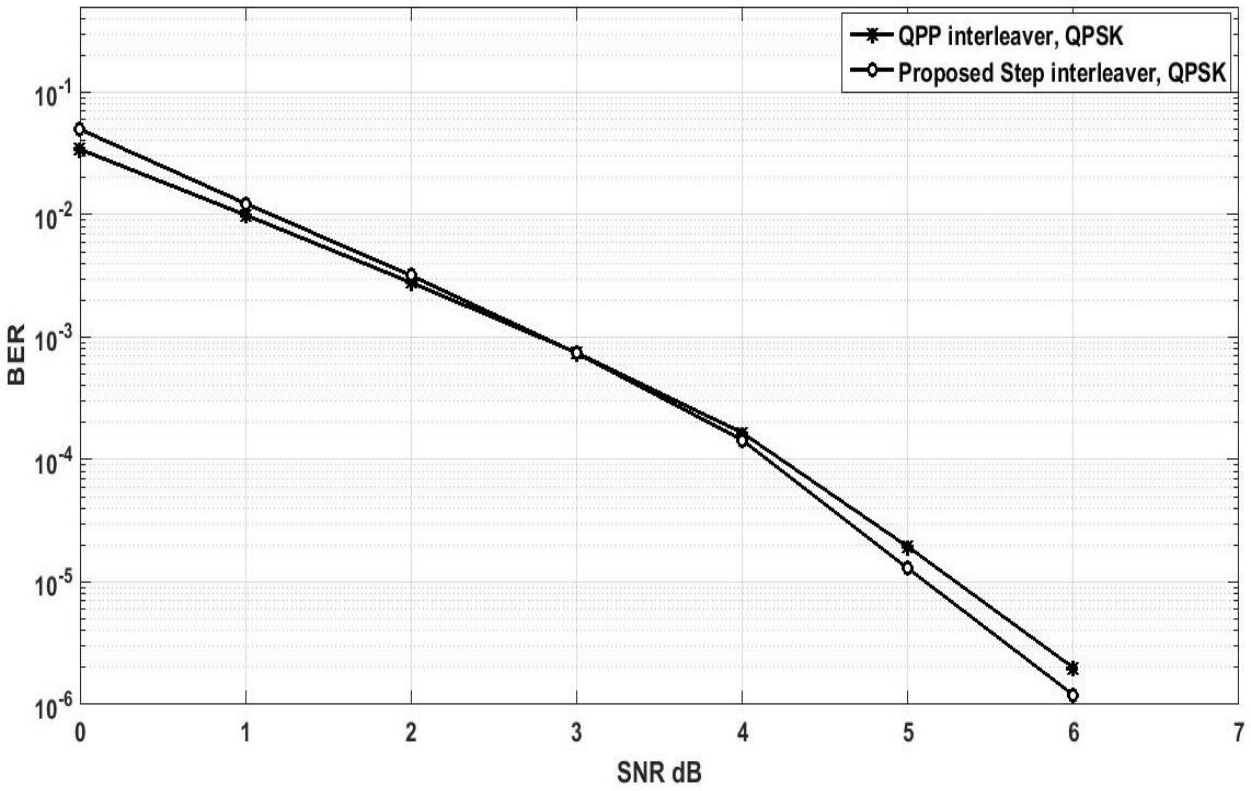

Figure 4. Parallel turbo code system, convolutional encoder rate 1/2 generator polynomial (5, 7), d_{free} = 5, constraint length =3. QPSK modulation. QPP interleaver vs proposed interleaver step size =67.

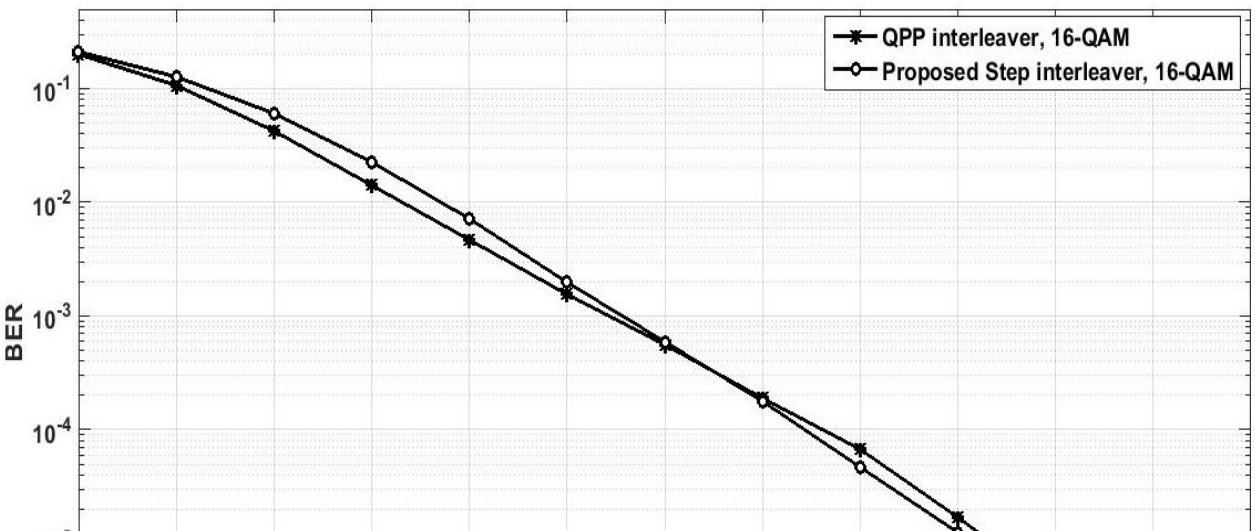

Figure 5. Parallel turbo code system, convolutional encoder rate 1/2 generator polynomial (5, 7), d_{free} = 5, constraint length =3. 16-QAM modulation. QPP interleaver vs proposed interleaver step size =67.

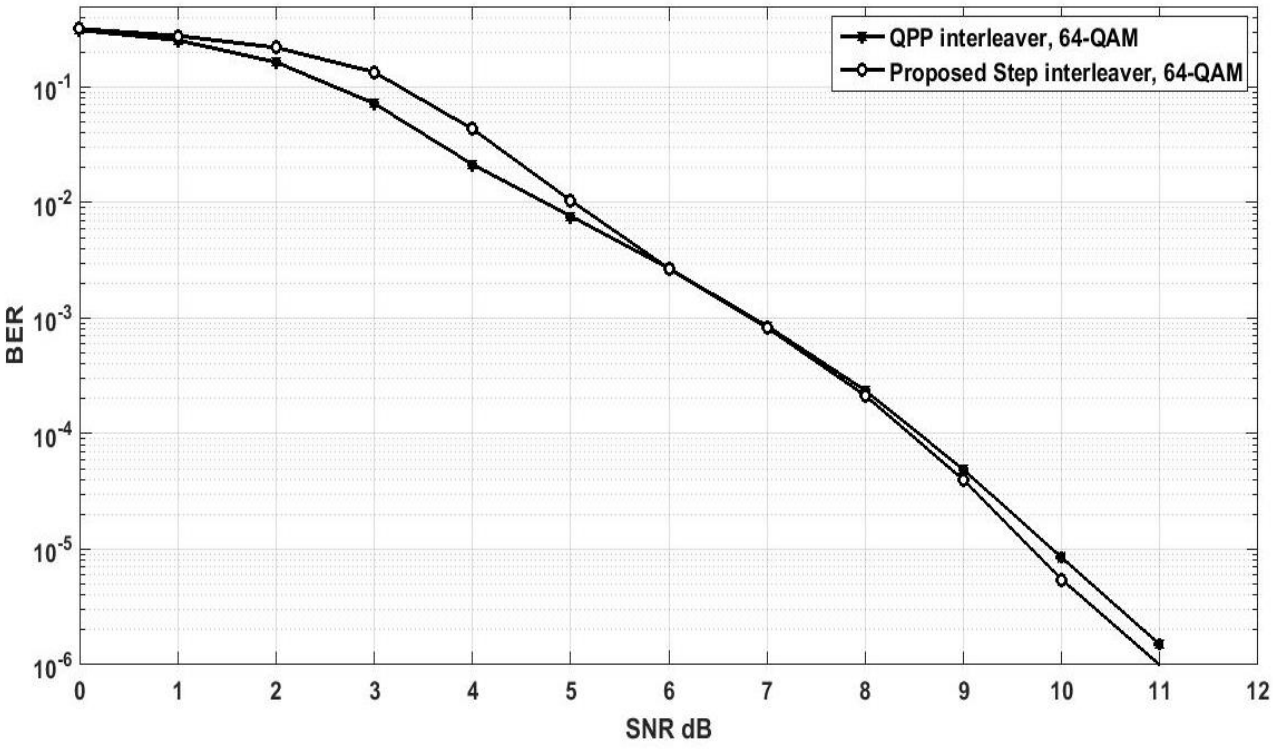

Figure 6. Parallel turbo code system, convolutional encoder rate 1/2 generator polynomial (5, 7), d_{free} = 5, constraint length = 3. 64-QAM modulation. QPP interleaver vs proposed interleaver step size = 67.

7. Conclusions

Implement proposed step interleaver using Arduino UNO with large frame length of 6144 and step size 67. Test the permutation result from implementation under MATLAB with communication system uses parallel turbo code and different modulation schemes. The comparison with standard QPP interleaver. Simulation results show improvement in system performance toward using the proposed step interleaver. also the proposed step interleaver not need stored values of f_1 and $f_2$rather than QPP interleaver. Which another point recorded to proposed step interleaver that reduce interleaver complexity.

8. References:

[1] Maria Kovaci, Horia G. Balta, Miranda M. Naforniţă, "The Performance of Interleavers Used with Turbo Code", Electronics and Telecommunications Faculty 2 Bd. V. Parvan, 1900 Timisoara, Romania.

[2] Sina Vafi and Tadeusz A. Wysocki," Application of convolutional interleavers in turbo codes with unequal error protection", Journal of Telecommunication and Information Technology. January 2006.

[3] P. Hanpinitsak and C. Charoenlarpnop parut, "2D Interleaver Design for Image Transmission over Severe Burst-Error Environment", International Journal of Future Computer and Communication, Vol. 2, No. 4, August 2013.

[4] Mohsen A. M. M. El-Bendary, Atef Abou El-Azm, Nawal El-Fishawy, Farid S. M. Al-Hosarey,Mostafa A. R. Eltokhy, Fathi E. Abd El-Samie, and H. B. Kazemian," An Efficient Chaotic Interleaver for Image Transmission over IEEE 802.15.4 Zigbee Network", Journal of Telecommunication and Information Technology. February 2011.

[5] Dr Houman Zarrinkoub, "Understanding LTE with MATLAB from Mathematical Modeling to Simulation and Prototyping", John Wiley & Sons, Ltd© 2014.

[6] A. Ghosh, R. Ratasuk, *Essentials of LTE and LTE-A*, Cambridge University Press, 2011.

[7] Arduino web site , https://www.arduino.cc/en/Main/ArduinoBoardUno. Access at 18-02-2017.

[8] Monocilindro web site, http://www.monocilindro.com/tag/catalex-microsd-adapter. Access at 18-0202017.