Electronics and Computer Science
Faculty of Physical Sciences and Engineering
University of Southampton

Author:
Boris Kolev
2nd May 2017

# A laboratory exercise on 802.15.4 communication between software defined radio (SDR) and XBee

Project supervisor:
Rob Maunder
Second examiner:
Geoff Merrett

A project report submitted for the award of
MEng Electrical and Electronic Engineering

## Abstract

The purpose of this project is to develop a fully worked lab experiment for third and fourth year students, as well as MSc students, learning about wireless communications in their university courses. The project uses the IEEE 802.15.4 wireless communications standard, implemented on a USRP 2922 (Universal Software Radio Peripheral) and XBee modules, using LabVIEW Communications as the programming environment. The IEEE standard implies low-cost, low-transmission rate and low-energy consumption, which nowadays are regarded as a priority for wireless communication devices. Using SDR (Software Defined Radio) to define this standard is an easy and convenient way to achieve a reliable communications system.

After completing this lab exercise, students will be exposed to an actual communication standard and will be able to see how standards are written and implemented in a real system. Moreover, students will be able to experience real-life O-QPSK transmission and test the reliability of the transceiver system by observing bit error rate (BER), constellation and eye graphs. Moreover, students will be able to analyze how different environments influence the performance of wireless communication systems. Finally, they will have the chance to implement the transceiver system by using actual RF modules, which are compliant with the 802.15.4 standard.

## Contents

## 1. Introduction

The IEEE 802.15.4 wireless communication standard [1] has been defined for low-power devices such as IoT (Internet of Things) [2] and XBee modules [3] and allows small electronic projects to easily implement wireless communications over short-distances.

The goal of this project is to create a fully worked lab experiment, which explains O-QPSK (Offset-Quadrature Phase Shift Keying) transmission [4] under the IEEE 802.15.4 wireless communication standard together with how an USRP-2922 [5] operates. The USRP acts as a SDR (Software Defined Radio) and allows users to easily create a transceiver system, which can be configured to follow the 802.15.4 standard. Most of the wireless communication courses offered in universities tend to focus on theory and sometimes students struggle to relate it to practice. In order to overcome this difficulty, SDR is implemented in this project, allowing students to get hands-on with wireless communications [6]. SDR offers the student a quick learning curve but exposes them to all the practical challenges of real-life communications. This project is meant to simplify the steps required to create a transceiver system by only looking at the PHY layer [1] of the 802.15.4 standard. Much of the wireless communications, nowadays, is dictated by standards because transmitters need to work with receivers from different manufacturers [7]. This is why, this project, uses the IEEE 802.15.4 standard and allows students to work with a standard and to see how it is implemented into practice. Furthermore, the project describes how an XBee2424 module works and tests its compatibility with an 802.15.4 compliant transceiver system developed by using LabVIEW Communications edition [8]. Moreover, most of the XBee papers propose the use of a microcontroller, which makes the implementation of the system more difficult. This project removes the need of a microcontroller by using an FTDI cable, which allows Serial to USB conversion. Finally, all of the work is incorporated in a lab exercise format, which will be aimed to be published in the IEEE Hands-On Lab Exchange [9].

Chapter 2 contains all the theoretical background connected to the 802.15.4 standard, USRP-2922, LabVIEW Communications and the XBee2424 module.

Chapter 3 is the planning section of the report, which contains what tasks have to be accomplished in order for the project to be successful together with all the risks regarding the project's failure. Moreover, it contains the time and project management techniques that are used in order to succeed backed up by a Gantt chart showing the flow of work.

Chapter 4 shows the implementation and testing of the actual transceiver together with the simulator of the system and the XBee compatibility testing.

Chapter 5 analyzes the obtained BER results and compares it to theoretical results to ensure correct system operations under the IEEE 802.15.4 communications standard.

Chapter 6 is a reflection section, which talks about on the work that has been carried out.

Chapter 7 contains ideas on what can be further implemented in the system to achieve a more advanced design and then concludes the project.

## 2. Background

In this part of the report the basic components of the project will be described as a theoretical interpretation. This section will describe the IEEE 802.15.4 wireless communication standard. First of all, the basics of the standard itself will be explained in section 2.1. Afterwards, a more in-depth analys will be taken into the O-QPSK PHY layer [1] of the standard, in section 2.3, together with all the necessary information regarding its different parts: PPDU (Physical Layer Convergence Protocol Data Unit) [10] format generation, bits-to-chip mapping [1], O-QPSK modulation and demodulation, and half-sine pulse shaping [11]. Furthermore, LabVIEW will be shown as a programming tool together with its essentials in section 2.4. The USRP will be described as a hardware part for the project's implementation in section 2.5. Lastly, the XBee24 module will be analyzed in section 2.6, for further use in the project.

### 2.1. Basics of the IEEE 802.15.4 standard

This standard is used to define the physical layer (PHY), which represents the radio transceiver, and medium access control (MAC) sublayer characteristics for low-data-rate wireless communications system [1]. The standard is based on the Open System Interconnection (OSI) model. This project focuses only on the PHY layer as specified in the introduction. The IEEE 802.15.4 network operates devices in the Personal Operating Space (POS), which have the same radio channel and form as the so-called LR-WPAN. These networks provide low-data rate, low-cost and short distance (around 15m radius) communication [12]. The standard also tries to tackle problems, which were not correctly solved by Bluetooth technology [13].

### 2.2.802.15.4 network topologies

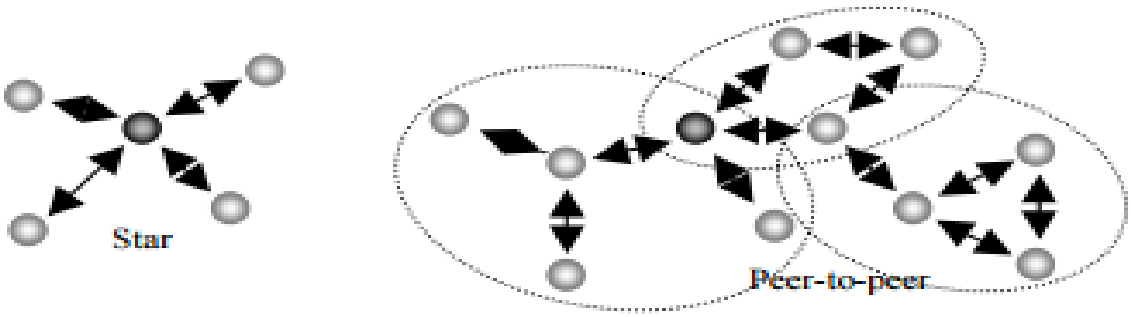


Figure 1. Star and peer-to-peer topology examples [14].

The 802.15.4 standard for LR-WPAN has two topologies - star topology and peer-to-peer topology [14] - which depend on the application requirements. The two topologies are shown in Figure 1. In the first topology, the star network, the devices communicate directly with an access point (AP) or the so-called central PAN coordinator which is the primary controller of the PAN [15]. There is also a PAN coordinator in the peer-to-peer topology, but it differs in that any device can communicate with any other device as long as they are inside each other's operating range. This makes the peer-to-peer topology suited for much more complex network formations such as mesh networking topology [1]. The devices in both of the topologies have either a function of an initiation point or a termination point for the networks.

### 2.3.PHY layer data transmission

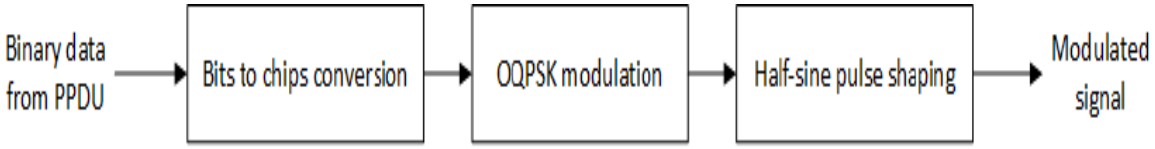


Figure 2. Generating the modulated PHY payload [16].

The whole procedure of transmitting data over the PHY layer is shown in Figure 2. In order to produce a modulated signal, the first step is to obtain the PPDU data packet with the only constraint that the PHY payload has to be bigger than 72 bits (9 octets) and no longer than 1016 bits (127 octets). This procedure is explained in section 2.3.1. Afterwards, these incoming bits are all mapped to chips following a fixed procedure, which will be explained in section 2.3.2. From this conversion, follows the O-QPSK modulation in section 2.3.3 which involves modulating the received chips into complex values in the I and Q phases. Furthermore, these complex values will be half-sine shaped (described in section 2.3.4) and then finally outputted to the receiving channel.

### 2.3.1. PPDU format generation

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

Figure 3. PPDU packet structure [16].

Whenever data is transmitted over the PHY layer, it is transformed into a PHY payload, which needs to be packaged into a packet structure for synchronization and detection. The PPDU packet structure is comprised of several fields and is shown in Figure 3. The structure is made in such a way that the leftmost field is received or transmitted first. All multiple octet (byte) fields are transmitted or received least significant octet first and each octet will be transmitted or received least significant (LSB) first. This transmission order should be applied when transferring data between the PHY and the MAC layer.
The PPDU packet contains the following [16]:

- Synchronization header (SHR), which is used for synchronizing and locking the receiving device to the bit stream. From Figure 3 it can be seen that it consists of the preamble field and SFD field.
  - The preamble is used for obtaining the chip and symbol synchronization within a message and has 4 octets each of which contains 8 bits which equal 32 bits of all zero bits.

| Bits: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

Figure 4. SFD field [1].

  - The SFD field guides the start and end frames of the sequence data and has an 8-bit length 0xA7 as shown in Figure 4, where Bit 0 is the least significant bit and is transmitted first.

- PHY header (PHR) containing all the frame length information. From Figure 3 it can be seen that the PHR consists of a frame length field and a reserved bit. The frame length field is 7 bits and is used to specify the total number of octets contained in the payload. This field is combined with the Reserved bit, which is usually a zero, to form the full PHR field. Suppose that the PHY payload is 64 (512 bits) octets long → convert 64 in binary → 0000001 (LSB first) → add the reserved bit 0 → obtain the PHR which is 00000010.
- A PHY payload (PSDU field), which is of variable length and carries the data of the PHY packet. As described in section 2.3 the length should be between 9 and 127 octets (72 bits and 1016 bits) [1]. The number of bits contained in this field must be divisible by 8 because it has to be an integer number of octets.

An example of a PPDU packet structure can be the following 00000000000000000000000000000000 (Preamble) 11100101 (SFD) 00000010 (8-bit PHR with a reserved 0-bit) 10...01 (PHY data packet with a length of 64 octets).

### 2.3.2. Bits to chip generation

| Data symbol | Chip values ($c_0$ $c_1$ … $c_{30}$ $c_{31}$) |
|---|---|
| 0 | 1 1 0 1 1 0 0 1 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 |
| 1 | 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 |
| 2 | 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 |
| 3 | 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 |
| 4 | 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 |
| 5 | 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 |
| 6 | 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 |
| 7 | 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 |
| 8 | 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 |
| 9 | 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 |
| 10 | 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 |
| 11 | 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 |
| 12 | 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 |
| 13 | 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 |
| 14 | 1 0 0 1 0 1 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 |
| 15 | 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 |

Figure 5. Symbol-to-chip mapping [1].

Firstly, the bits will be mapped to data symbols in the way that the 4 LSBs (b0, b1, b2, b3) of each octet are mapped to one data symbol, where b0 is the least significant bit, and then the 4 MSBs (b4, b5, b6, b7) are mapped to the next data symbol [1]. This is done by converting the binary numbers into decimals in the range 0-15, as seen in Figure 5. The LSBs are processed first and the MSBs are processed second. Afterwards, for the 2450 MHz band, every one of the data symbols will be mapped to a 32-chip PN sequence in such a way that the PN sequences are all related as shown in Figure 5. As a simple example if the bits are 1100 → converts to the data symbol 12 → generated chip sequence will be 00000111011110111000110010010110 .

### 2.3.3. O-QPSK modulation

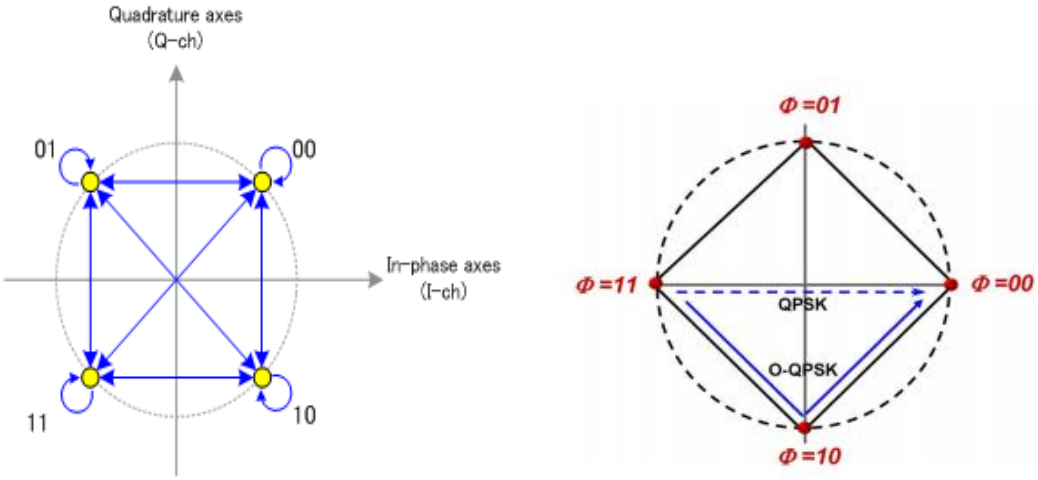


Figure 6. QPSK and O-QPSK constellation - 1) QPSK [17] 2) O-QPSK [18].

The O-QPSK is known as the Offset-Quadrature Phase Shift Keying and is an improved version of the standard QPSK [19]. The schemes are shown in Figure 6. In QPSK there is a series of bits which are converted into a series of complex numbers (e.g. 00 is converted into 1+i; 01 → -1+i; etc.) which are represented as constellation points. In this scheme transitions from any of the constellation points can be made to any of the other constellation points in successive steps due to the fact that both the I and Q components change at the same time which leads to 0 crossings, as shown in Figure 6 – 1). These 0 crossing lead to the signal changing from low to high and vice-versa, which leads to the need of a very linear amplifier. On the other hand, O-QPSK prevents these 0 crossings by changing the I and Q components one at a time and delaying the Q-phase by half a symbol period [20] which removes the sudden phase shift change as shown in Figure 6 – 2). In other words, O-QPSK only jumps $0^o$ and $\pm 90^o$ which leads to a constant signal amplitude (the amplitude follows the dotted line shown in

Figure 6 – 2)), compared to the QPSK signal amplitude, and removes the need of a linear amplifier.

The chips acquired from the bit-to-chip generation are modulated to the carrier using the O-QPSK. The even chips get modulated onto the in-phase and the odd chips go to the quadrature phase. Now, in order to form the offset, the Q-phase is delayed by $T_c$ with respect to the I-phase, where $T_c$ is the inverse of the chip rate which is 32 times the symbol rate [19]. This offset can be seen in Figure 7.

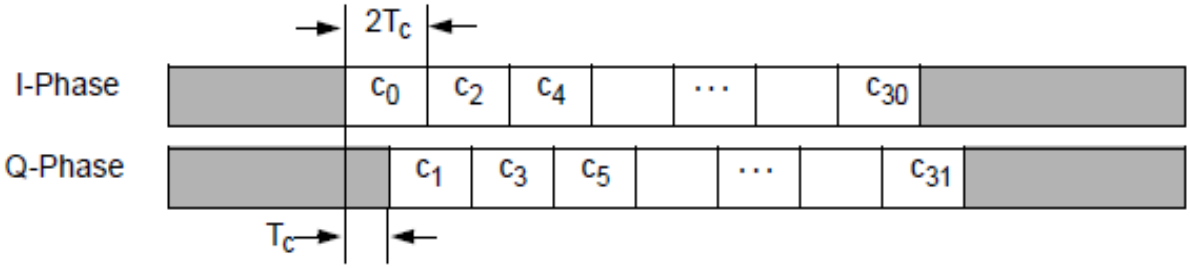


Figure 7. O-QPSK offset in the two phases [21].

An example for Figure 7 can be represented with the bit sequence 0000 which when transformed into a complex chip sequence will be → 11011001110000110101001000101110

OQPSK can be demodulated the same way QPSK is demodulated - by using coherent quadrature demodulation. The only difference here is that there is a delay $T_c$ introduced to the Q-phase with respect to the I-phase.

The O-QPSK PHY operates in 5 different frequency bands but for this project the one utilized is between 2400.0 – 2483.5 MHz which has 16 available channels, a data rate of 250 kbps and has a center frequency defined as [1]:

$$F_c = 2405 + 5(k - 11) \text{ in megahertz, for k=11,12,..26 (1)}$$

### 2.3.4. Half-sine pulse-shaping

In order to reduce the inter-symbol interference and to reduce the bandwidth of the modulated signal, half-sine pulse shaping is used which expands the rectangular pulse into a half-sine pulse or as mentioned in the previous section – delays the Q phase of the signal by half a symbol period leading to a constant signal amplitude. For the 2450 MHz band each chip is represented as follows [1]:

$$p(t) = \begin{cases} \sin \pi \frac{t}{2T_c}, 0 \le t \le 2T_c \\ 0, otherwise \end{cases} (2)$$

Figure 8 shows the half-sine pulse shaped baseband chip sequence which corresponds to the example in section 3.3.2.2:

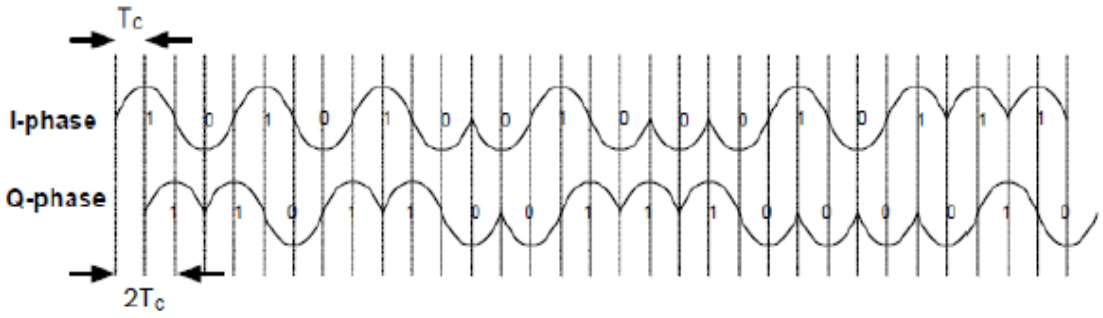


Figure 8. Half-sine pulse shaped chip sequence [1].

Complex-valued chip samples, $c_k$, have a discrete-time sequence and have the following continuous-time pulse shaped signal [1]:

$$y(t) = \sum_{k=-\infty}^{\infty} c_{2k} p(t - 2kT_c) + j c_{2k+1} p(t - 2kT_c - T_c) (3)$$

## 2.4.LabVIEW

LabVIEW is a graphical based programming environment which is specifically designed for engineers and scientists [22]. LabVIEW uses a data flow model instead of using the traditional text-based programming which enables fast learning and understanding of the program. The graphical interface allows the user to easily create all kinds of systems and virtual instruments (VI). Data acquisition and analysis can easily be performed by using all the different tools provided by the program which reduces working time. By using this graphical programming environment, the rate of development can be easily increased while keeping the functionality and stability of the system. This project will use LabVIEW Communications 1.1 which is a special edition of LabVIEW which is created to deal with communication-related projects.

LabVIEW consists of two panels:

- Block diagram - contains the graphical programming elements (2) and the functions palette (1) as shown in Figure 9.

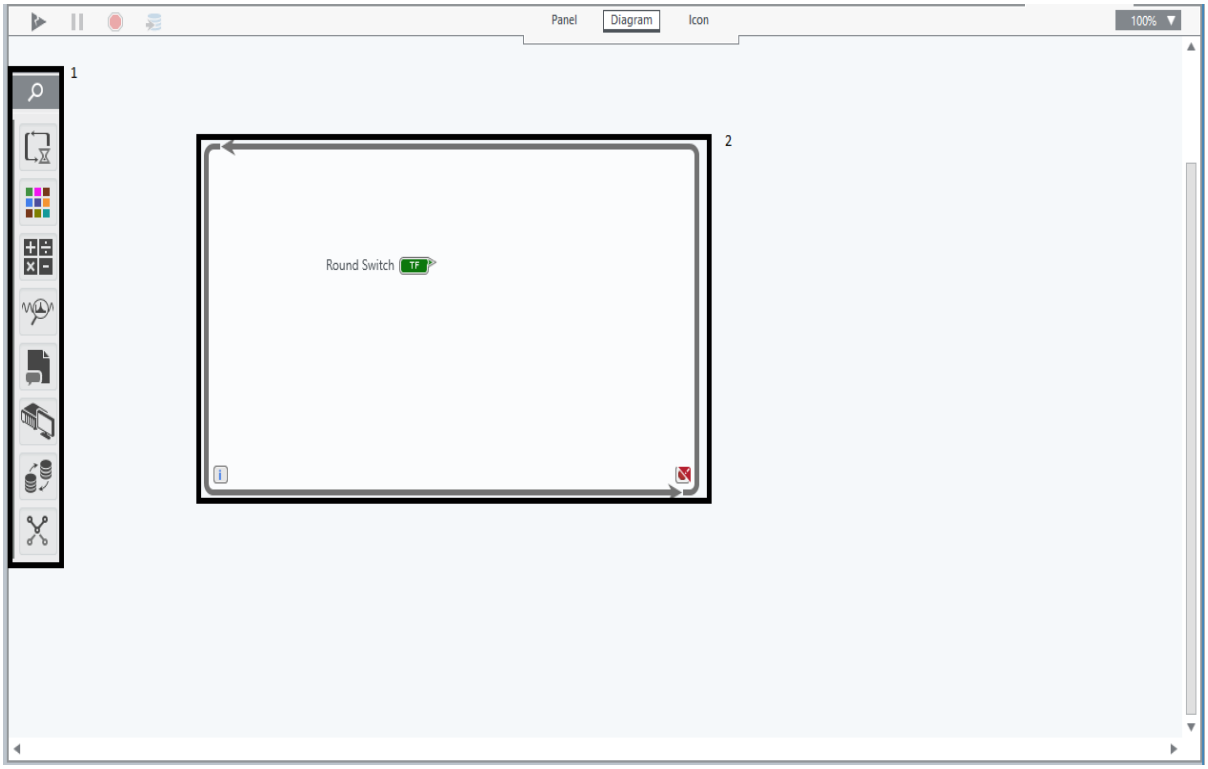


Figure 9. LabVIEW Block diagram.

- Front panel - contains the user interface controls (2) of the program and the controls palette (1) as shown in Figure 10.

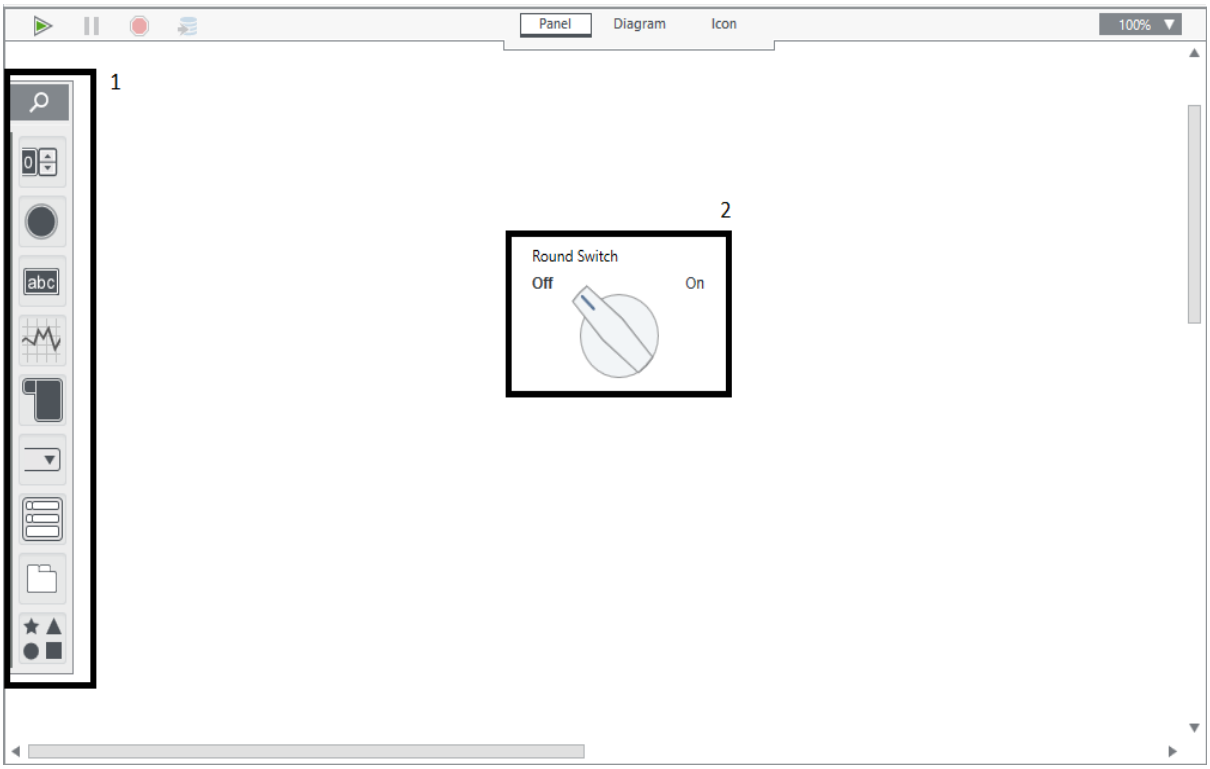


Figure 10. LabVIEW Front Panel.

## 2.5.USRP

The Universal Software Defined Radio is a member of the Software Defined Radio (SDR) family and used as a transceiver of RF signals [23]. Figure 11 shows the design of the module. It utilizes Analog to Digital and Digital to Analog conversion and allows computers to act as radio devices with high bandwidths. The devices consist of a motherboard which is represented as a high-speed FPGA including sub-boards which manage different frequency ranges from AM radio to Wi-Fi. These sub-boards transmit data to the PC using the antennas provided. As a whole, this device enables engineers and scientists to create innovating systems with minimal requirements. The USRP uses GNU Radio framework for PHY processing [23].


Figure 11. USRP 2922 module [24].

### 2.5.1. USRP Architecture

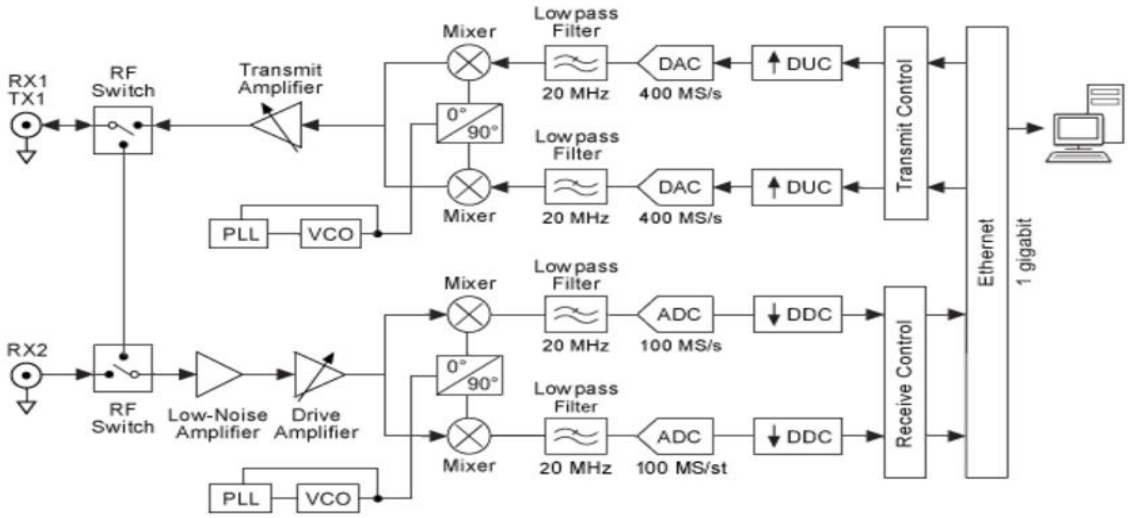


Figure 12. USRP architecture [24].

The USRP implies a common architecture for software-defined radios [25] as shown in Figure 12. It combines direct conversion analog front end with ADCs and DACs. The FPGA is used for the digital down conversion (DDC) and digital upconversion (DUC). When receiving the signal reaches a highly sensitive analog front end, which can receive very small signals, and digitizes it using DDC to In-Phase (I) and Quadrature (Q) baseband signals. From this point, ADC is carried out followed by a DDC which reduces the sampling rate and packetizes I and Q for transmission over an Ethernet cable. When transmitting from the host PC, where I and Q are generated and transferred over the Ethernet cable. Here a DUC sets up the DAC after which I-Q mixing occurs to directly up-convert the signals and produce the RF signal, which is amplified and transmitted [25].

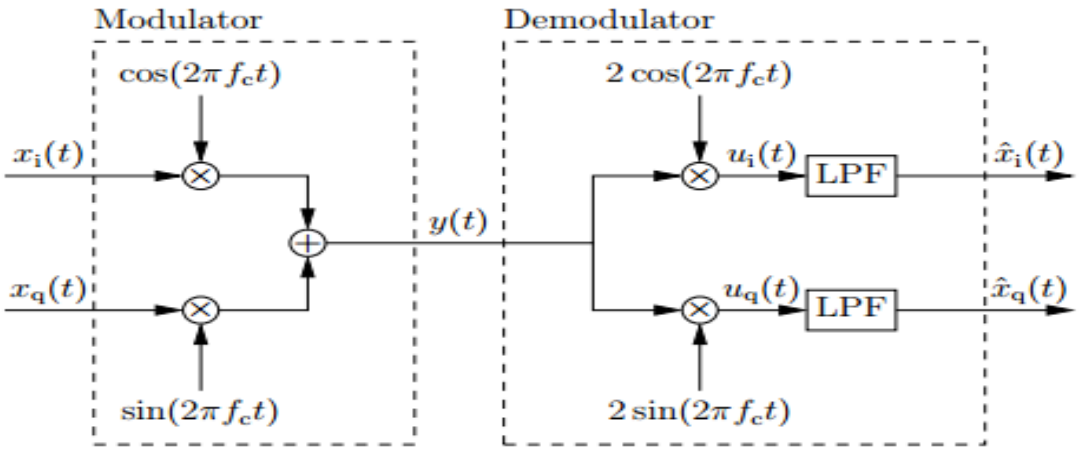


Figure 13. Signal Processing in a USRP device [26].

The USRP transceiver device follows a simple signal processing structure described in Figure 13. The modulated complex values is separated into I - $x_i(t)$ and Q - $x_q(t)$ phases → multiplied by $\cos(2\pi f_c t)$ and $\sin(2\pi f_c t)$ respectively and then summed up to obtain $y(t) = x_i(t)\cos(2\pi f_c t) + x_q(t)\sin(2\pi f_c t)$ → multiplied by $2\cos(2\pi f_c t)$ and $2\sin(2\pi f_c t)$ respectively, to obtain $u_i(t) = 2x_i(t)\cos(2\pi f_c t)\cos(2\pi f_c t) + 2x_q(t)\sin(2\pi f_c t)\cos(2\pi f_c t)$ and $u_q(t) = 2x_q(t)\sin(2\pi f_c t)\sin(2\pi f_c t) + 2x_i(t)\cos(2\pi f_c t)\sin(2\pi f_c t)$ → using the trigonometric identity $2\cos(x)\cos(x) = 1 + \cos(2x)$, $2\sin(x)\sin(x) = 1 - \cos(2x)$ and $2\cos(x)\sin(x) = \sin(2x)$ to obtain $u_i(t) = x_i(t) + x_i(t)\cos(4\pi f_c t) + x_q(t)\sin(4\pi f_c t)$ and $u_q(t) = x_q(t) - x_q(t)\cos(4\pi f_c t) + x_i(t)\sin(4\pi f_c t)$ → Low Pass Filter the signal in order to remove the out of band frequency components ($4\pi f_c t$) which leaves $\widehat{x_i(t)} = x_i(t)$ and $\widehat{x_q(t)} = x_q(t)$.

### 2.6.XBee24 module

The XBee24 module shown in Figure 14 to provides a cost-effective wireless connection between devices. It allows direct programming due to its onboard microprocessor and enables custom application development. The Serial Peripheral Interface (SPI) provides a high-speed interface and easily communicates with embedded micro-controllers which decrease costs and working time. This module is specially designed to incorporate the 802.15.4 standard which makes it a perfect RF device for the project. It has two modes of operation [27]:

- Transparent data mode - data is transmitted over-the-air to the receivers without being modified. It can be point-to-point communication or star.
- Application Programming Interface (API) - data is wrapped in a packet structure that takes care of all the communication steps.

The device is 802.15.4 compliant which means that it is capable of establishing a connection under this standard.

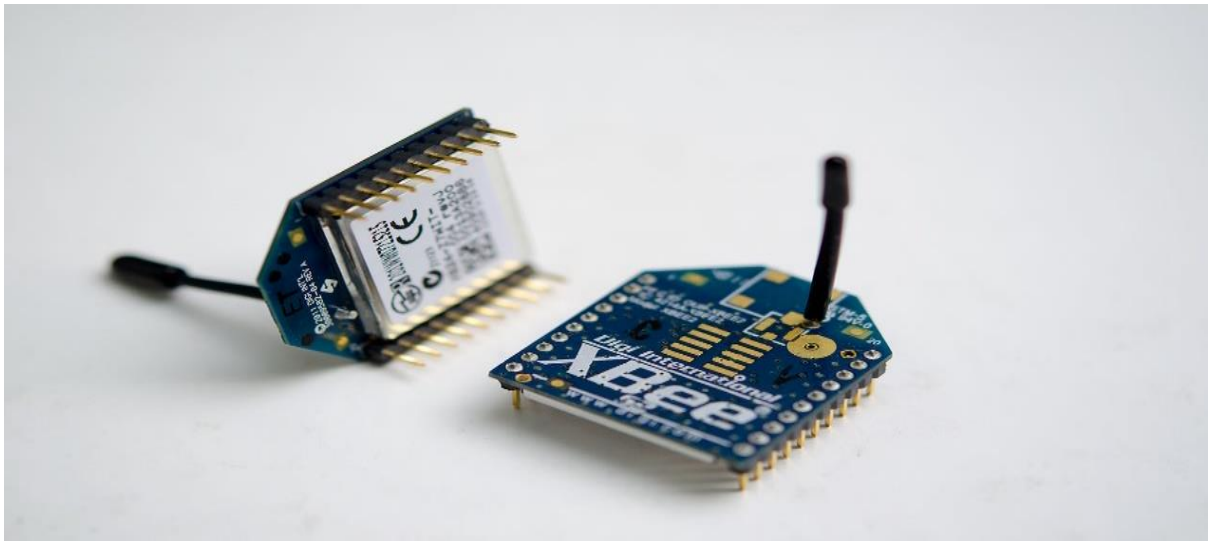


Figure 14. XBee24 module [28].

## 3. Planning

This chapter discusses the planning steps undertaken in order to produce a successful project. The project scope, success criteria and the project tasks are described in sections 4.1, 4.2 and 4.3. Sections 4.4 and 4.5 talk about the available resources and the risks connected with the project development, followed by a discussion on how the time spent on the project is managed in section 4.6, backed up by a Gantt chart in section 4.7.

### 3.1.Project scope:

The idea of this project is to develop a fully worked lab experiment based on the IEEE 802.15.4 wireless communications standard for students who study this material in university. After being finished the project is aimed to be published in the IEEEComSoc Hands-on Lab Exchange website. The project is entirely constructed in the LabVIEW programming environment and incorporates the use of a USRP 2922 pair and O-QPSK modulation, demodulation, and simulation. Due to the complexity of the standard, only the PHY layer will be focused on. Furthermore, XBee24 modules are included in the experiment and communication between them and the USRP is established so that the USRP transceiver systems detects the communication between the XBee modules. This project is based only around creating a software-defined radio without implementing a real telecommunications system.

### 3.2.Success criteria:

In order for this project to be successful the following criteria should be met:

- Create a fully working O-QPSK system that has a fully working transmitter and receiver and incorporates the 802.15.4 standard.
- Create a simulator of the system that works without the USRP and the XBee24 modules and aim to produce results which match the theoretical results of using the 802.15.4 standard and more precisely the O-QPSK modulation and demodulation.
- Characterize the BER performance of the O-QPSK used for the project and aim to produce values close to zero at sufficiently high SNR values.
- Manage to detect the communication between the XBee modules with the use of the USRP devices.
- Manage to set up the XBee24 modules to work with FTDI cables, without the use of a micro-controller.
- Create fully worked lab notes with questions for students together with examples and teacher notes.
- Manage to create a lab experiment worthy of publishing on the IEEEComSoc lab exchange website.
- Manage to meet final deadline and produce the 3,000-word interim report before Christmas and then finish the 10,000-word report before May.

### 3.3.Project tasks:

The whole project can be divided into the following sub-tasks, each of which will require different time-spans in order to be incorporated:

1. Learn how to program using the LabVIEW environment by undertaking LabVIEW programming courses organized by the University of Southampton.
2. Research the IEEE 802.15.4 standard (O-QPSK PHY module) together with the USRP and XBee24 modules by researching relevant scientific sources.
3. Create a PPDU format generator for the PHY payload together with bit-to-chip mapping.
4. Design a fully working O-QPSK modulator with PN spreading code which uses half-sine pulse shaping.
5. Design a demodulator for the complex value O-QPSK, which contains PPDU data locator using AWGN, correct signal synchronization, correct chip-to-bit demodulation used for producing the BER ratio, and correct waveform demodulation.
6. Create a simulator for the whole system so that the practical results can be compared to the theoretical ones.
7. Set up a connection between 2 XBee devices and detect the communication between them by using the USRP receiver module.
8. Test the whole system, compare results and debug the system if needed
9. Create fully worked lab notes for students and teachers which can be used in university.
10. Create a 10,000-word report combining all the work done.

### 3.4.Available resources:

Everything needed for the project is either provided by the university or is available to be purchased online with the provided individual project budget:

- The IEEE 802.15.4 standard which is available online.
- LabVIEW courses provided by the University of Southampton.
- USRP 2922 provided by the University of Southampton.
- XBee24 modules and adapters available to purchase online.

- Past dissertation papers connected to LabVIEW and the 802.15.4 standard available online – Chong You – "Implementation of transceiver system under IEEE 802.15.4 based on LabVIEW with USRP"
- QPSK LabVIEW template code provided by Dr. Rob Maunder

### 3.5.Risk assessment:

The following risk were outlined together with their impact, probability and solution:

- Computer may get stolen or damaged thus losing all work – High Impact – Low probability – Never leave personal belongings unattended and always keep back-ups.
- Insufficient resources for the project – High impact – Low probability – Carefully research all the success criteria components and have a clear overview of all the required steps in order to produce a fully-working system.
- Failure to produce a fully working O-QPSK transceiver system which correctly follows the IEEE 802.15.4 communication standard – High Impact – High Probability - Research past dissertation papers on the same topic and actively work with supervisor in order to debug all arising problems and keep up with the schedule.
- Failure to create an O-QPSK system with a BER performance of 0. This includes all the parts of the system - PPDU format generation, bits-to-chip mapping, and O-QPSK modulation and demodulation – High Impact – High Probability – Research past dissertation papers on the same topic and actively work with supervisor in order to debug all arising problems and keep up with the schedule.
- Failure to get the XBee24 modules to work with the USRP as these modules are extremely complex and require a lot of understanding before being able to incorporate it into the whole system – High Impact – High Probability – Thoroughly research scientific papers connected with XBee implementation and seek supervisor's help whenever issues occur.
- Failure to generate good lab notes – High Impact – Medium probability – Carefully look through lab notes from different universities so as to get a good idea of what a good lab consists of. Furthermore, carefully breakdown the project into different parts so as to be able to easily make lab exercises.

- Falling ill – Medium impact – Medium probability – Keep after personal health and schedule regular check-ups.
- Failure to meet deadlines – High Impact – Low probability – Carefully manage time by using different time management techniques such as Gantt charts.

### 3.6.Time management

The final deadline for the individual project is the 2nd of May which means that there are nearly 7 months in order to produce the whole design. In order to be time efficient a project schedule is designed and can be seen in section 4.7.

The first task undertaken is to get familiar with all the background material governing the project. As this is only research related, 1 week is appointed. In order to become efficient with LabVIEW special university organized courses are attended which have a length of 6 weeks. During this time the general aspects of O-QPSK modulation are programmed in the LabVIEW environment. First, the PPDU format generation and the bits-to-chips mapping are considered. For these activities 2 weeks are appointed. After completing these tasks, the O-QPSK modulation is implemented in a transmitter module which again has a 2-week duration. Afterwards, the receiver system is implemented which includes all of the O-QPSK demodulation components (PPDU location, synchronization, waveform demodulation and chip-to-bits conversion for obtaining BER ratio). As this task has the most components 4 weeks are appointed. When the transceiver is finished the simulator is created. For its completion 5 weeks are appointed from which 3 will be holiday weeks and no work will be carried out. A progress report is required showing the work done before the Christmas holidays. This task requires a lot of attention and 5 weeks are appointed, allowing plenty of time for revising and correcting mistakes. This report will be used as a draft of the original report which will be written throughout the whole project's time length – around 6 months. After the exam session at the end of January, the XBee implementation begins and continues for 6 weeks. When every aspect of the system is implemented, testing will begin for roughly 3 weeks which is enough time to compare the results with theoretical values and further improve the system. Finally, 2 weeks before the deadline of the project, fully-worked lab notes will be created which will include exercises for students and lab notes for the teaching staff.

### 3.7.Gantt chart:

A simple Gantt chart was generated in order to have an overview of how to proceed with the project.

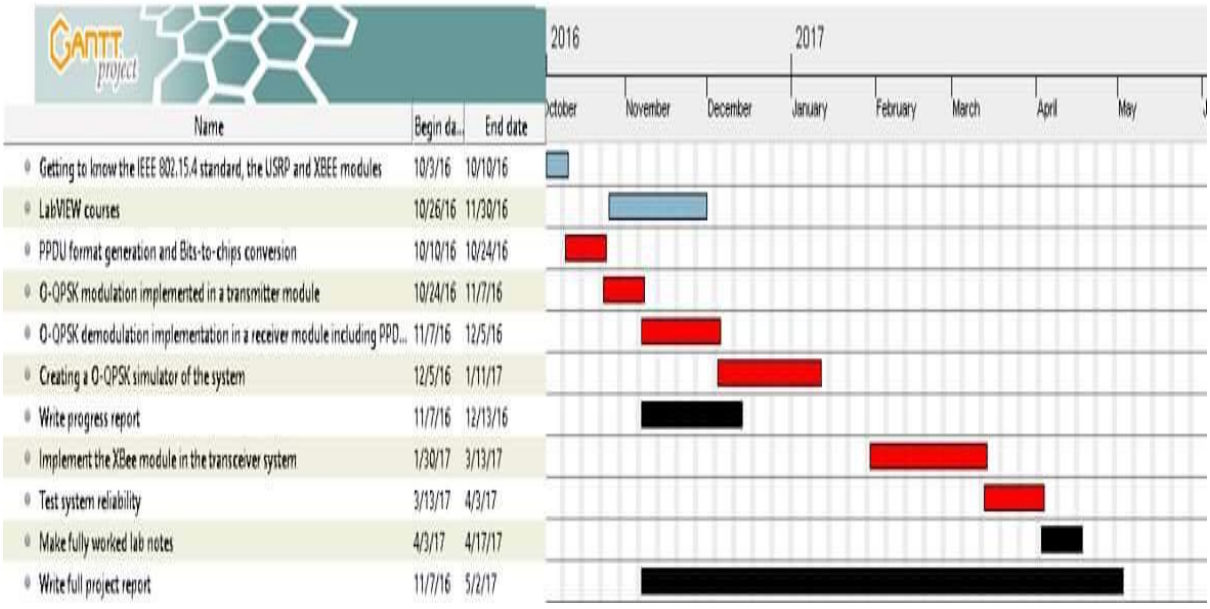


Figure 15. Gantt chart of the work process

## 4. Implementation and Testing

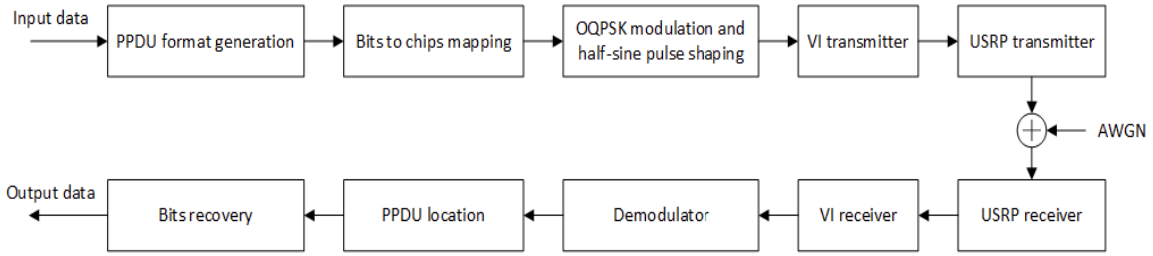


Figure 16. Transceiver schematic [29].

This section will talk about the implementation and testing of the transceiver system. Figure 16 shows the transceiver schematic. The transmitter consists of three main components – PPDU format generation, bits-to-chips mapping and O-QPSK modulation with half-sine pulse shaping, which transform the input data into a suitable form for the USRP transmitter. Sections 4.1 and 4.2 will talk about this in more detail. The receiver then recovers the data from the USRP receiver and the signal goes through O-QPSK demodulation, PPDU location and chips-to-bits recovery. Section 4.3 and 4.4 will explain this process more thoroughly.

## 4.1.O-QPSK transmitter module implementation and testing

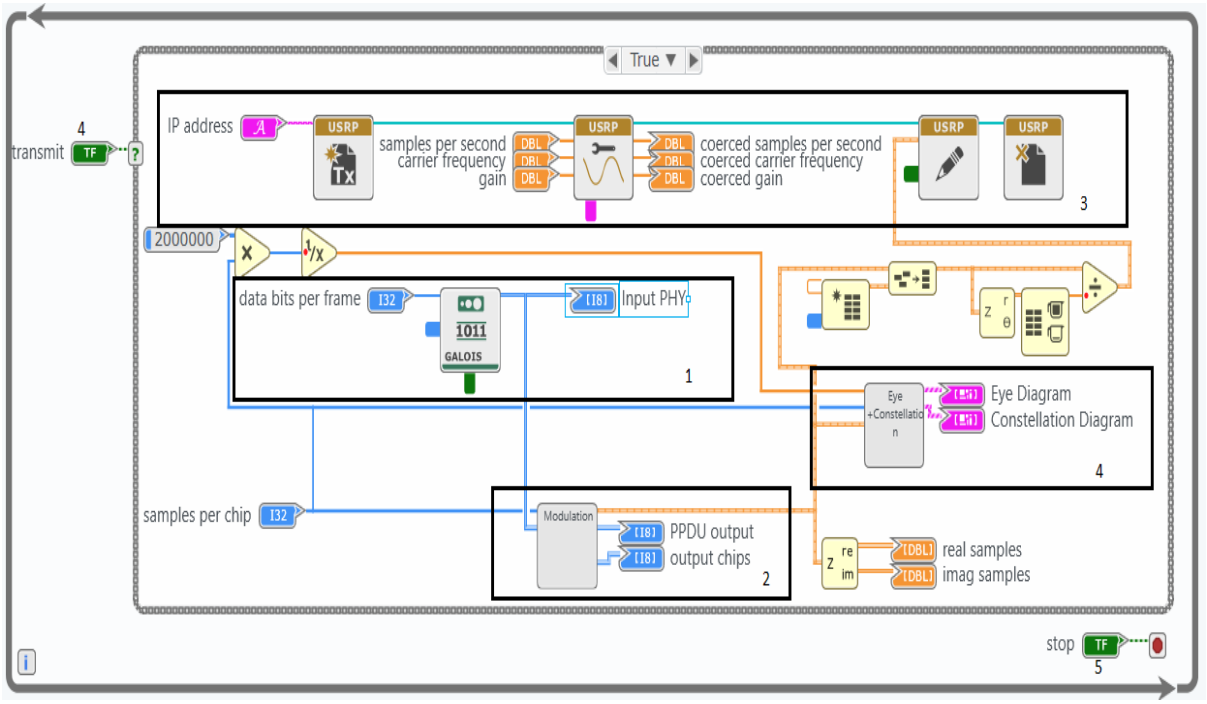


Figure 17. Transmitter VI block diagram.

Figure 17 shows the VI's block diagram. There are 4 components to be described – input generation functions (1), modulation VI (2), USRP settings functions (3) and eye and constellation diagram VI (4). These components will be described in sections 4.1.1 to 4.1.4. The transmit button (5) ensures new data to be transmitted every time it is pressed and the stop button (6) terminates the execution of the program.

### 4.1.1. Input generation


Figure 18. Front panel visualization of the input generation.

The first step to establishing the O-QPSK modulation is to generate random bits, which will be inputted to the modulation module of the VI. Figure 17 (1) shows this procedure. The random bits are generated by a function available in LabVIEW, which continuously generates random bits in the form of an array. The total bits per frame are inputted to the random bit generating function and then outputted on the front panel every time the transmit button is pressed as seen in Figure 18. These generate bits then are inputted into the modulation module.

### 4.1.2. O-QPSK modulation

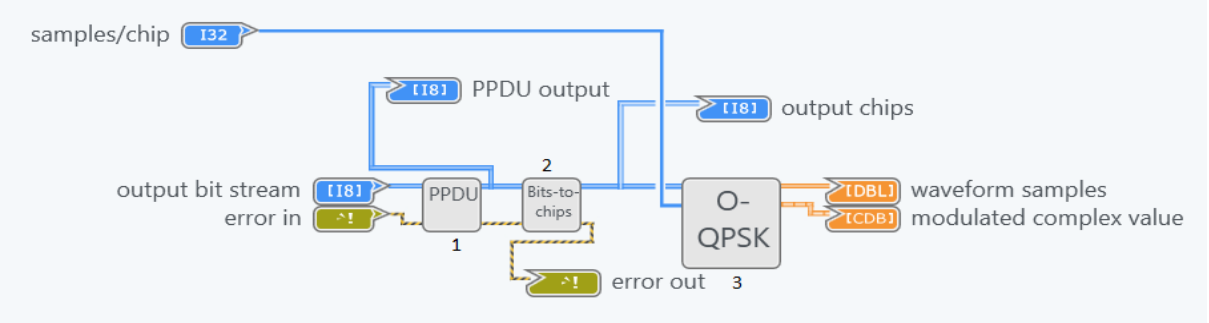


Figure 19. O-QPSK Modulation components.

The O-QPSK modulation module consists of 3 separate modules, as shown in Figure 19, which cover all the steps for producing PPDU format (1) then converting it to chips (2) and finally O-QPSK modulating (3) so as to obtain the modulated complex value and samples needed for transmission. Each individual step is described below. These steps follow the procedure shown in Figure 4.

#### 4.1.2.1. PPDU format generation

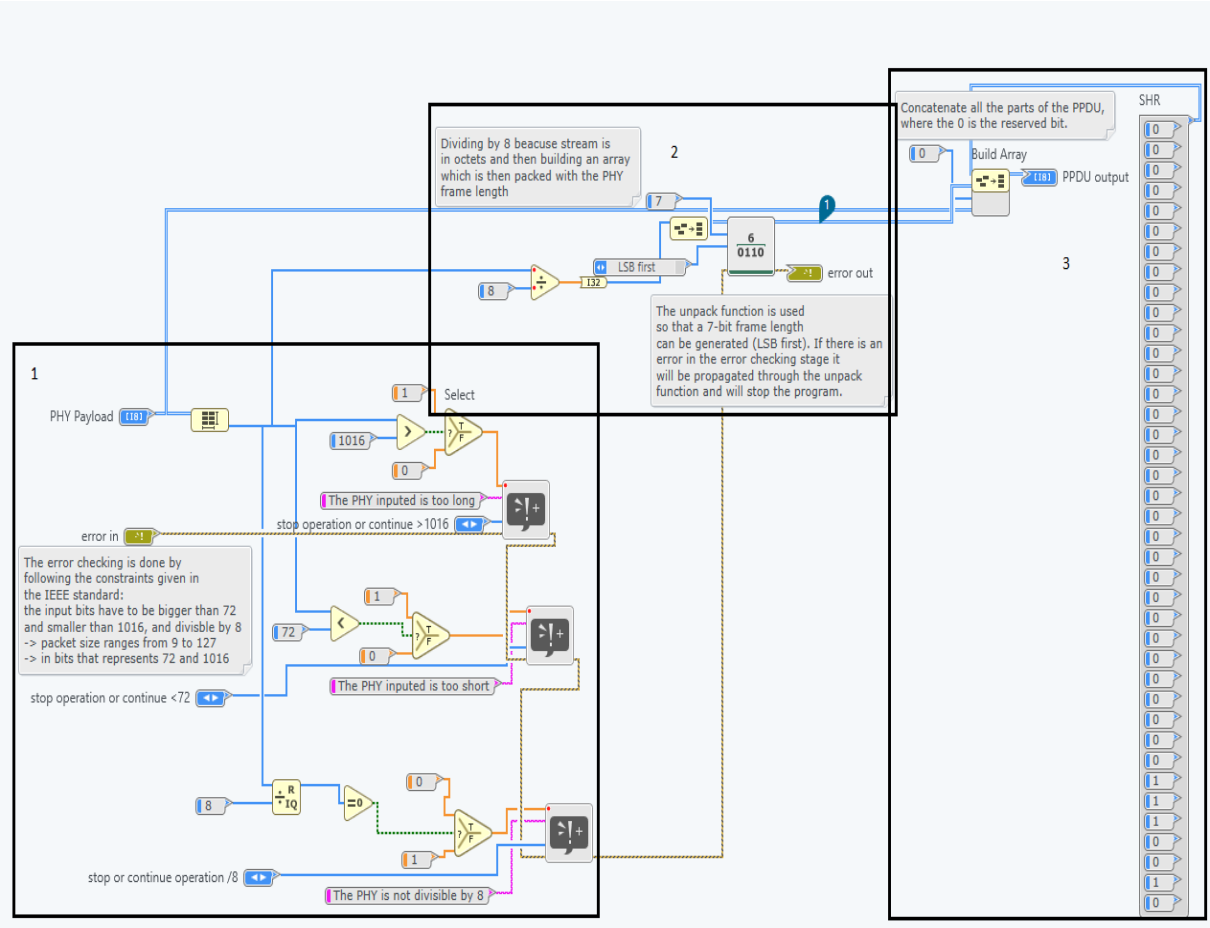


Figure 20. PPDU format generation.

The PPDU format generation function contains three parts – error checking (1), frame length generation (2) and PPDU generation (3) as shown in Figure 20.

#### 4.1.2.1.1. Error checking

Figure 20 (1) shows the error checking component for generating PPDU. The error checking part ensures that the incoming bits from the input generation in section 4.1.1. are following the IEEE 802.15.4 standard which specifies that the PHY input must be no larger than 1016 bits (127 octets), no smaller than 72 bits (9 octets) and must be divisible by 8 as described in the section 2.3. The error propagates throughout the whole PPDU format generation sequence and when the PHY does not meet the requirements, an error message is displayed.

When the inputted PHY is 1…1(1020 bits) the execution of the program is stopped and the following message is displayed:

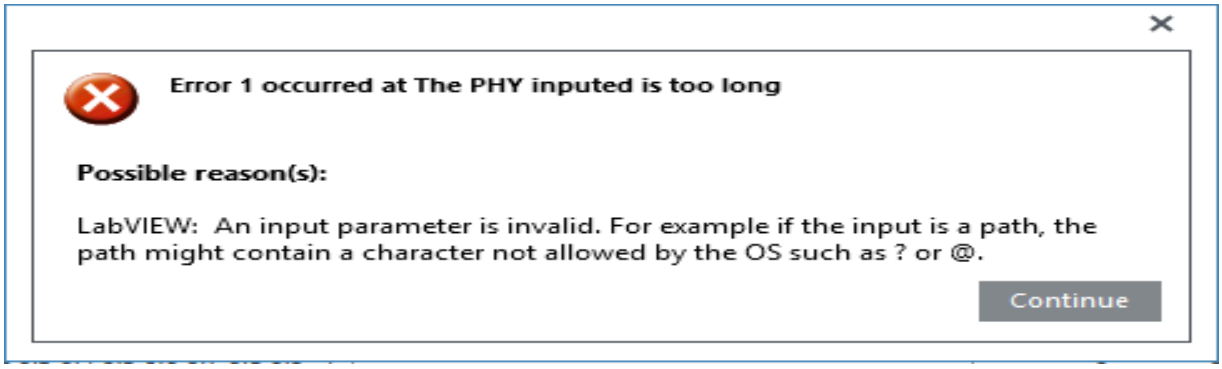


Figure 21. PHY payload is too long error message.

With a PHY input of 1…1 (64 bits) the VI stops execution:

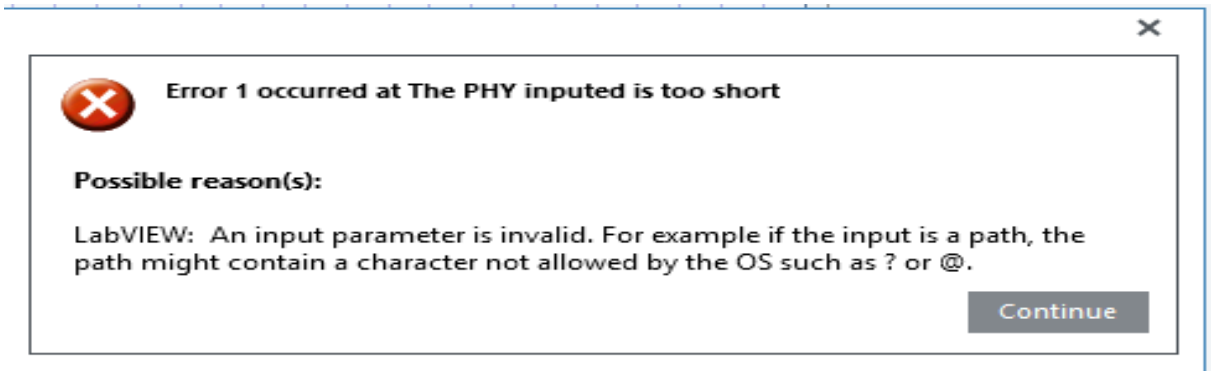


Figure 22. PHY payload is too short error message.

Inputting 1…1 (74 bits) leads to:

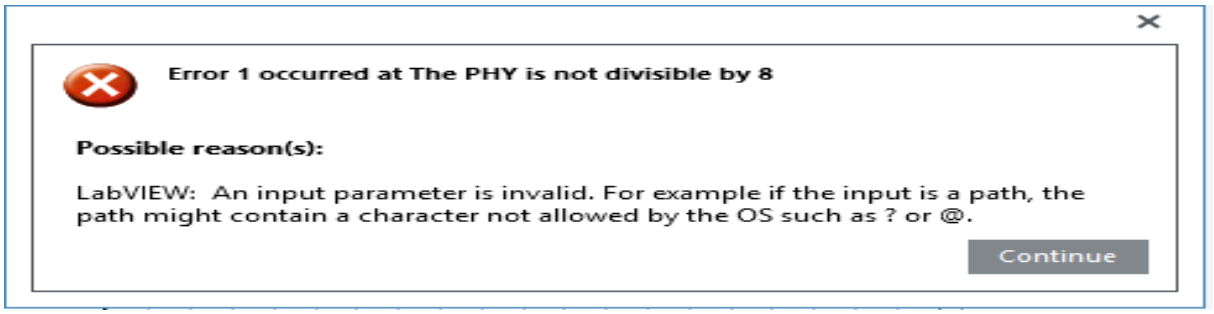


Figure 23. PHY payload is not divisible by 8.

#### 4.1.2.1.2. Frame length generation

Figure 20 (2) shows the frame length generation of the PPDU format. The PHY payload length is divided by 8 due to the fact that the frame length in octets as seen in Figure 2. An array is then created which has a length equal to the divided PHY payload length. The array is then unpacked to a 7-bit binary data format by using a LabVIEW unpack bit function with the LSB being formatted first. The output of this function is the generated PPDU frame length, which meets the requirements stated in section 2.3.1.

#### 4.1.2.1.3. PPDU output generation

As it can be seen from Figure 20 (3), the SHR which contains the Preamble (32-bits of 0's) and the SFD (1110010) are concatenated into an array together with the PHR, which contains the generated in section 4.1.2.1.2 frame length and a reserved bit which is always a 0, and the generated in section 4.1.1. PHY payload input (PSDU field) in order to form an array representing the PPDU format output.

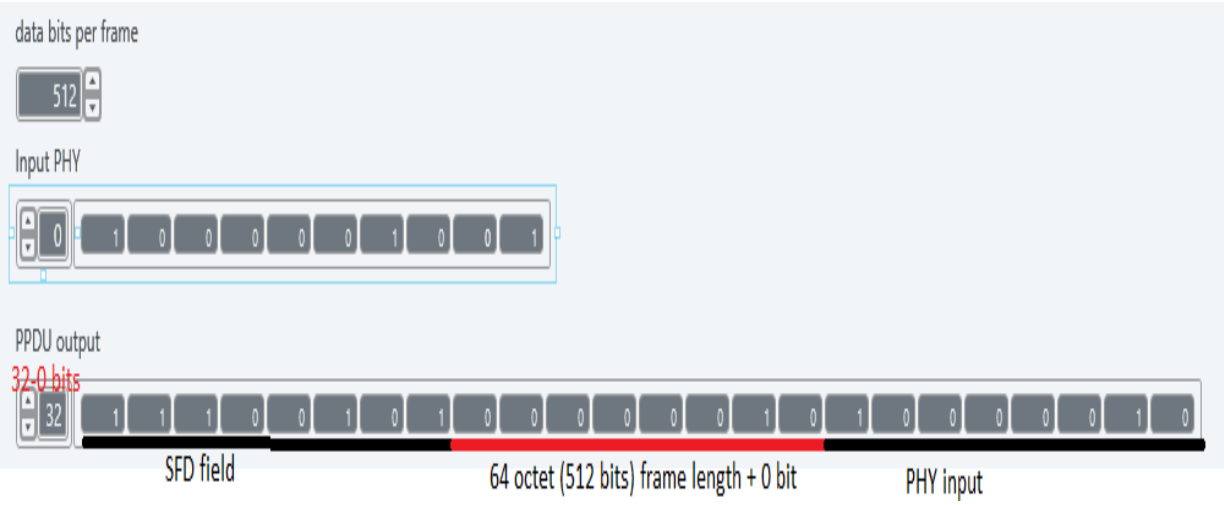


Figure 24. PPDU output generation front panel visualization.

When the example from section 2.3.1. is inputted, the correct PPDU sequence is generated where the first 32-bits are the Preamble field of all 0's as seen in Figure 24. This PPDU then goes as an input to the bits-to-chips module as shown in Figure 17.

#### 4.1.2.2. Bits-to-chips mapping

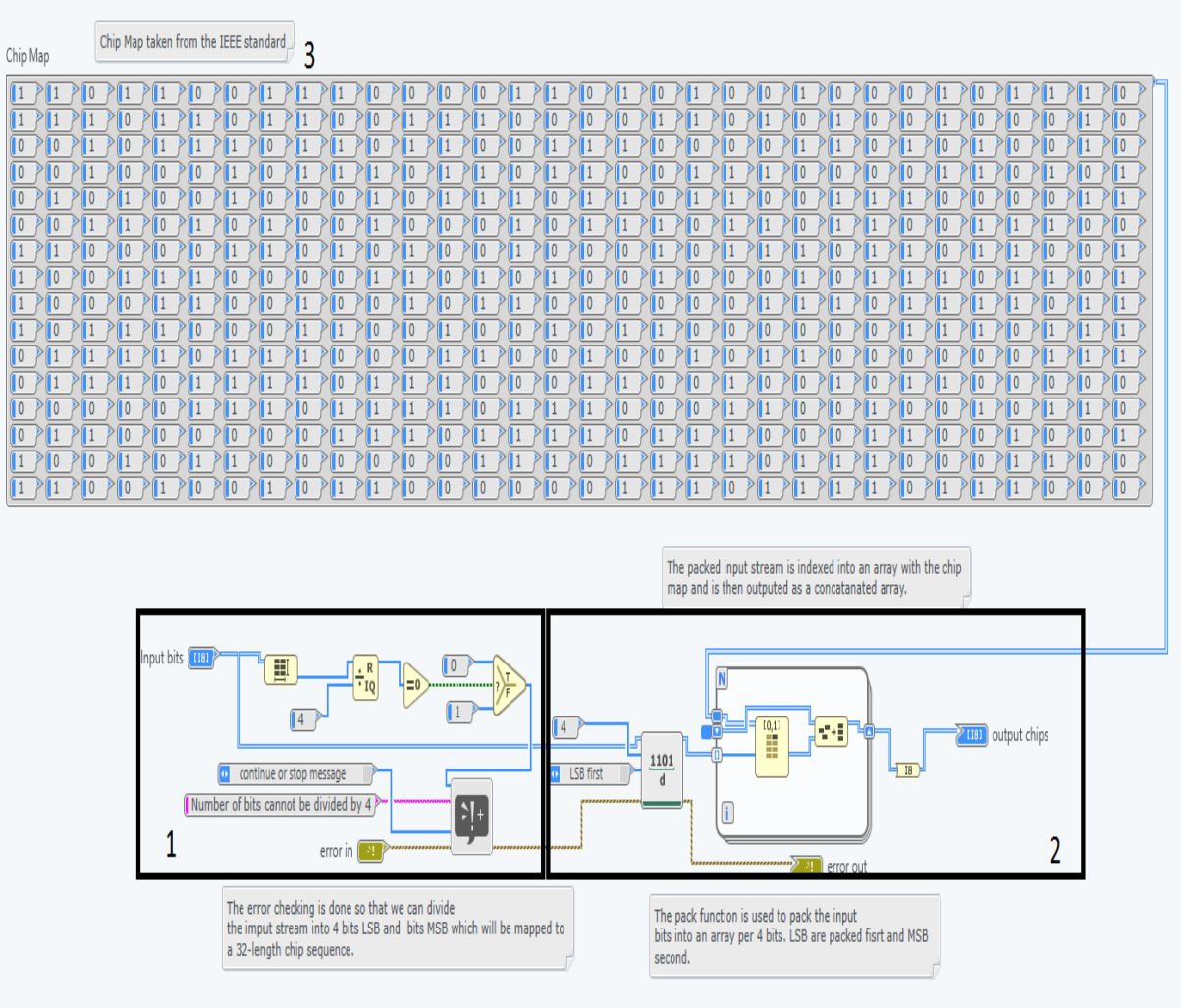


Figure 25. Bits-to-chips mapping block diagram.

The bits-to-chips mapping module contains 3 components – error-checking (1), bits-to-chip conversion (2) and the chip map (3) which the 32-length symbol chips specified in section 2.3.2. All of these components are described in the following sections.

##### 4.1.2.2.1. Error checking

The error checking procedure shown in Figure 25 (1) follows the same steps as the one described in section 4.1.2.1.1. The only difference here is that the size of the generated PPDU is checked to see if it is divisible by 4 because the chip mapping is done first to the 4 LSB and

then the 4 MSB. If the input, let's say 000001, is not divisible by 4 an error is displayed - Number of bits cannot be divided by 4. This can be seen in Figure 26.

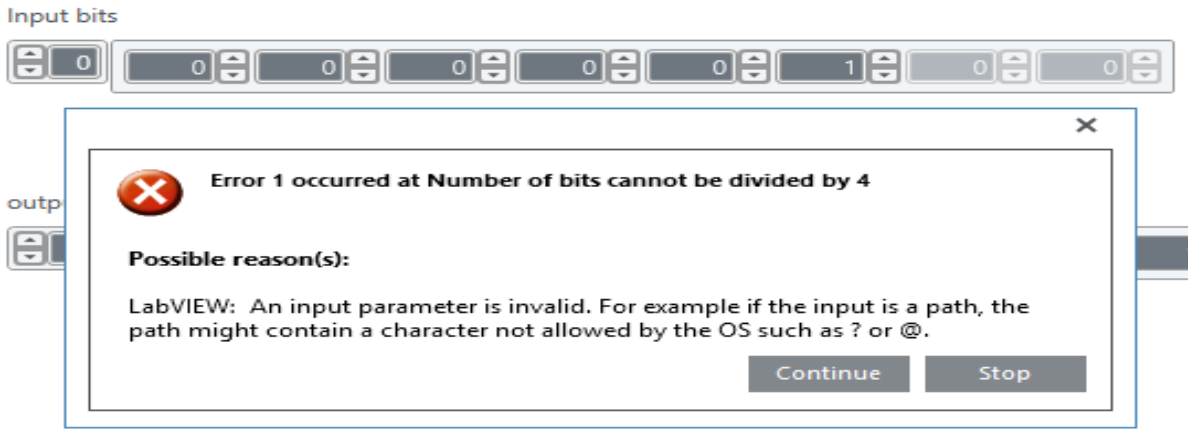


Figure 26. Number of bits cannot be divided by 4 error message.

#### 4.1.2.2.2. Actual Bits-to-chips mapping

Figure 25 (2) shows how the PPDU format is first packed to integers per 4 bits (LSB first) by using the LabVIEW's pack bits function. These integers are then packed in an array by using a for loop, which loops for each integer received. Afterwards this array is used to define the rows of the array indexing function which outputs one of the 16 different chip value sequences described in section 2.3.2 (3). Finally, the output of the array indexing function is concatenated with a 0 initialized shift register in order to produce the chip value output which is then inputted to the O-QPSK module as seen in Figure 17. When the data symbol 12 (1100) from Figure 5 is inputted the correct chip value is obtained as seen in Figure 27. The input here is in the form 0011 because the LSB goes first.

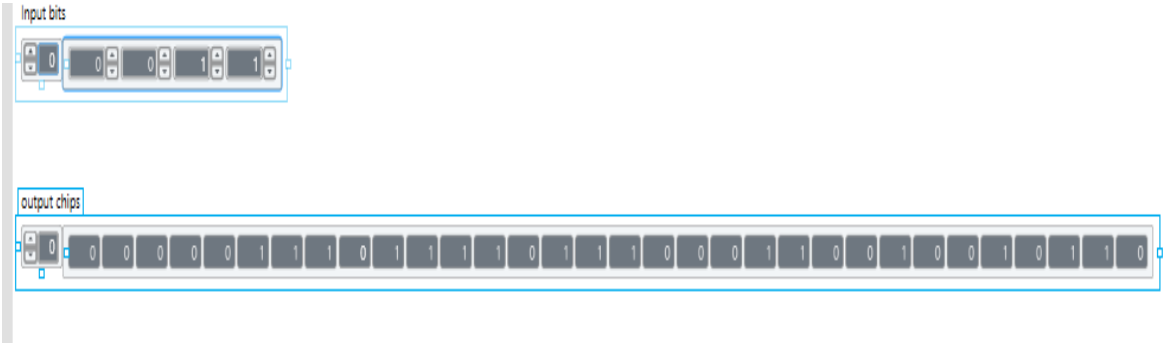


Figure 27. Correctly outputted chip sequence.

#### 4.1.2.3. Actual O-QPSK Modulation

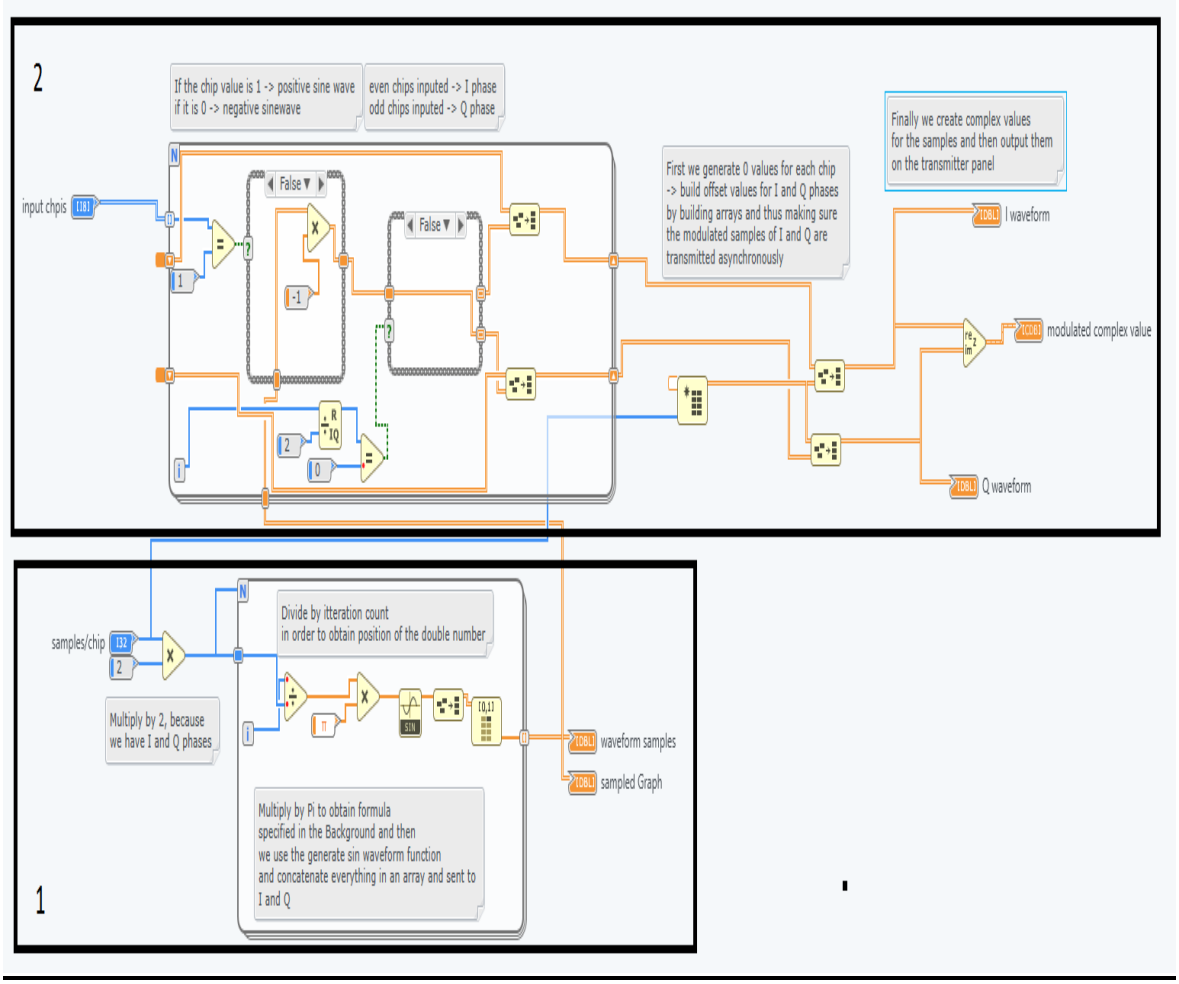


Figure 28. O-QPSK modulation module block diagram.

In order to achieve correct O-QPSK modulation as described in section 2.3.3, 3 steps are needed – half-sine waveform generation (1), I and Q phase shift separation and offset data length generation (2).

##### 4.1.2.3.1. Half-Sine waveform generation

Figure 28 (1) shows the procedure of generating a half-sine waveform which will be used as an individual template for the actual bits transmitted and modulated in section 4.1.2.3.2. The first step is to multiply the user defined samples/chip value by 2 in order to satisfy the transmission on both the I and Q phases. Afterwards, the ordinal position of the samples/chip is obtained by dividing the newly obtained value by the iteration count of the for loop. Multiplying by π defines the phase of the sampled sine waveform. This is then inputted to a Sine function, which computes the sine value and then sends the waveform to the I and Q phases as shown in Figure 28.

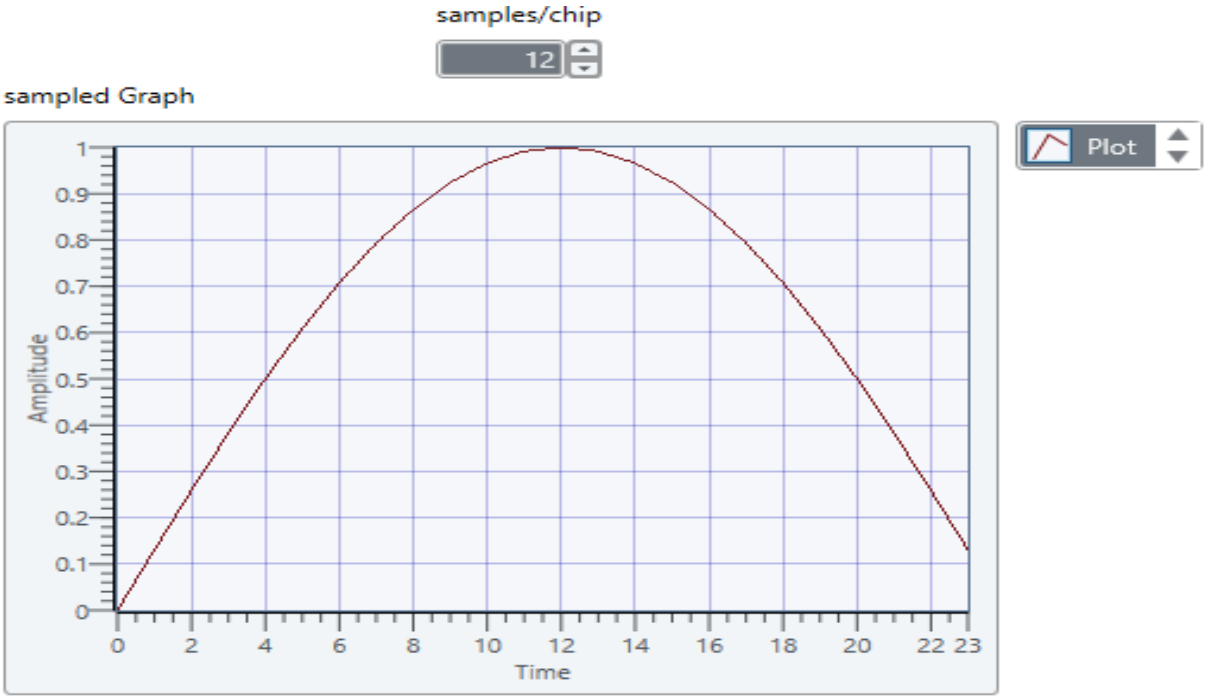


Figure 29. Generated half-sine waveform.

When inputting 12 samples/chip a perfect upper half-Sine waveform is obtained as seen in Figure 29. The gap, in the end, represents the connection point of the following waveform.

#### 4.1.2.3.2. I and Q phase shift and offset generation

Figure 28 (2) represents the actual I and Q modulation together with the offset generation needed in order to acquire the O-QPSK signal described in section 2.3.3. Firstly, the chips generated in section 4.1.2.2.2 are compared with either a 1 or a 0. From sections 2.3.3 and 2.3.4, it is clear that 1 represents the positive part of the Sine waveform and 0 represents the negative. These values are then assigned either to the negative part of the sine wave by multiplying it by -1 or to the positive part of the 0sine wave by multiplying it by 1. The number of inputted chips is then divided into odd and even values by using a quotient and remainder function where the even bits go the I phase and the odd bits go to the Q phase. Arrays are built for both the I and Q phases.

In order to obtain the offset between the I and Q phases a 0 valued array is initialized for each of the chip's transmissions. Afterwards, this array is concatenated with the arrays of I and Q, thus obtaining the I and Q waveforms. These waveforms are then transformed into a single complex value which then goes as an input to the Eye and Constellation diagram module, and the USRP settings functions as shown in Figure 17.

The obtained I and Q waveforms, when using the chip sequence given in section 2.3.3, are shown in Figure 30.

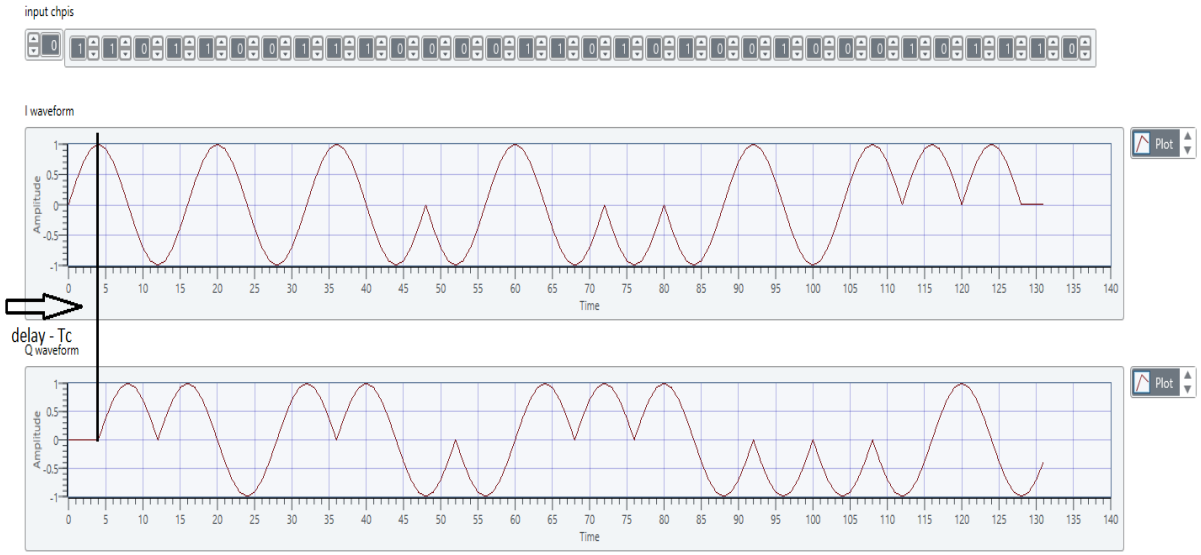


Figure 30. I and Q waveforms.

By comparing these results with the theoretical in sections 2.3.3 and 2.3.4, and Figure 8, it can be clearly seen that the modulator correctly generates half-sine waveforms, where the Q phase has a delay of Tc, where Tc is the inverse of the chip rate and follows the principles of O-QPSK modulation from the IEEE 802.15.4 standard.

### 4.1.3. Eye and Constellation Diagrams

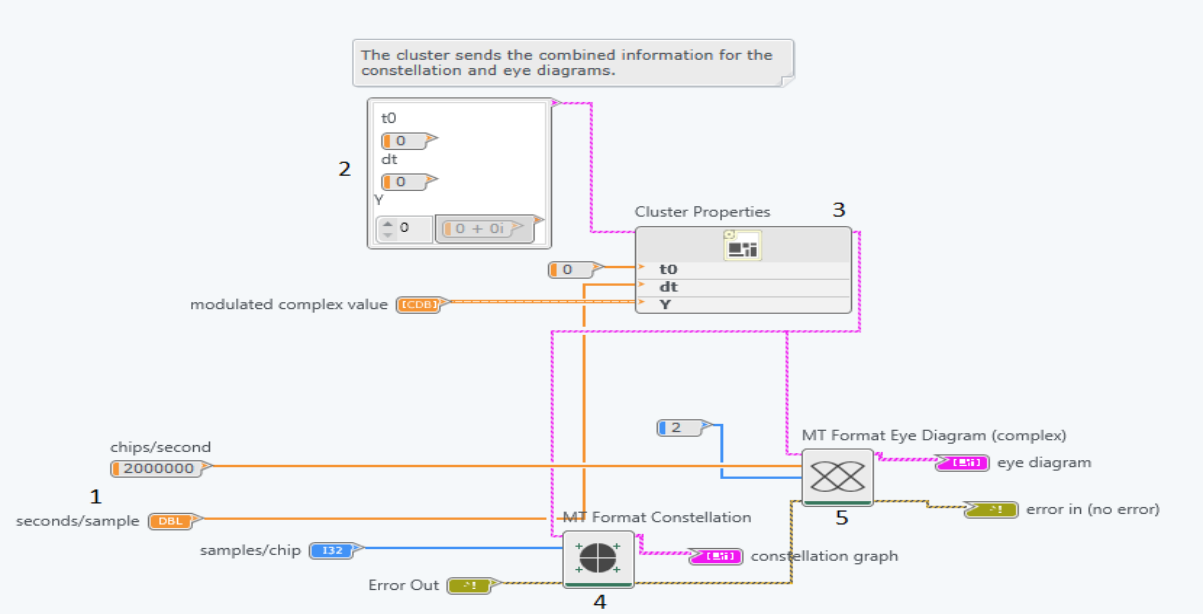


Figure 31. Eye and Constellation module block diagram.

This section of the transmitter contains five components as shown in Figure 31. In order to get the period between two samples, the sample rate, and its reciprocal must be obtained. This is done in Figure 17 by multiplying the chips per second and the samples per chip followed by a reciprocal function, which obtains the seconds per sample (1). The chips per second values are obtained from the IEEE 802.15.4 standard – 250 kbps (section 2.3.3) with every 4 bits of PPDU being converted to 32-length symbols leads to a chip rate of 2,000,000. This value must be constant meaning that $\frac{samples/second}{samples/chip} = 2\ 000\ 000$ . This value is then inputted in the cluster properties function (3) which defines the initial time, the time between two samples and sample amplitude of each sample initialized in the cluster (2). The data from the Cluster Properties function is used as an input to a constellation diagram function(4) and an eye diagram function (5), which plot the constellation and eye diagram of the incoming signal. With 2 samples/ chip inputted the following diagrams are plotted on the Front Panel:

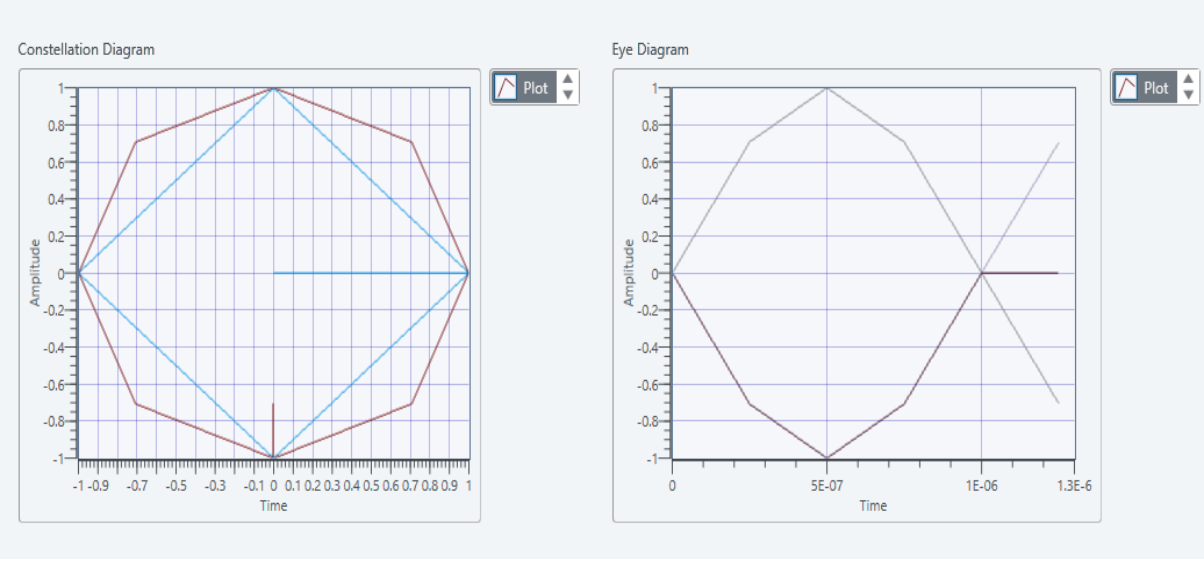


Figure 32. Eye and Constellation diagrams.

From these results, it can be seen that the constellation diagram correctly represents the O-QPSK modulated signal and that the transition between two diagonally opposite phases passes through an intermediate phase removing the ±180º phase shift as described in section 2.3.3. The eye diagram is a "perfect" open eye showing that the transmitted data follows the constellation circle.

### 4.1.4. Transmitter USRP settings

From Figure 17 the functions that set up the USRP can be seen. The first step is to create a transmission (Tx) session for the specified in the IP address device and output a session handle.

This session handle is then used as an input to a function which configures the sample rate, carrier frequency and gain of the USRP transmitter and again outputs a session handle for the next function, which writes the modulated complex values obtained in section 4.1.2.3.2 to the transmission channel. Finally, a close function is used to close the session handle to the specific design.

### 4.2.O-QPSK transmitter testing

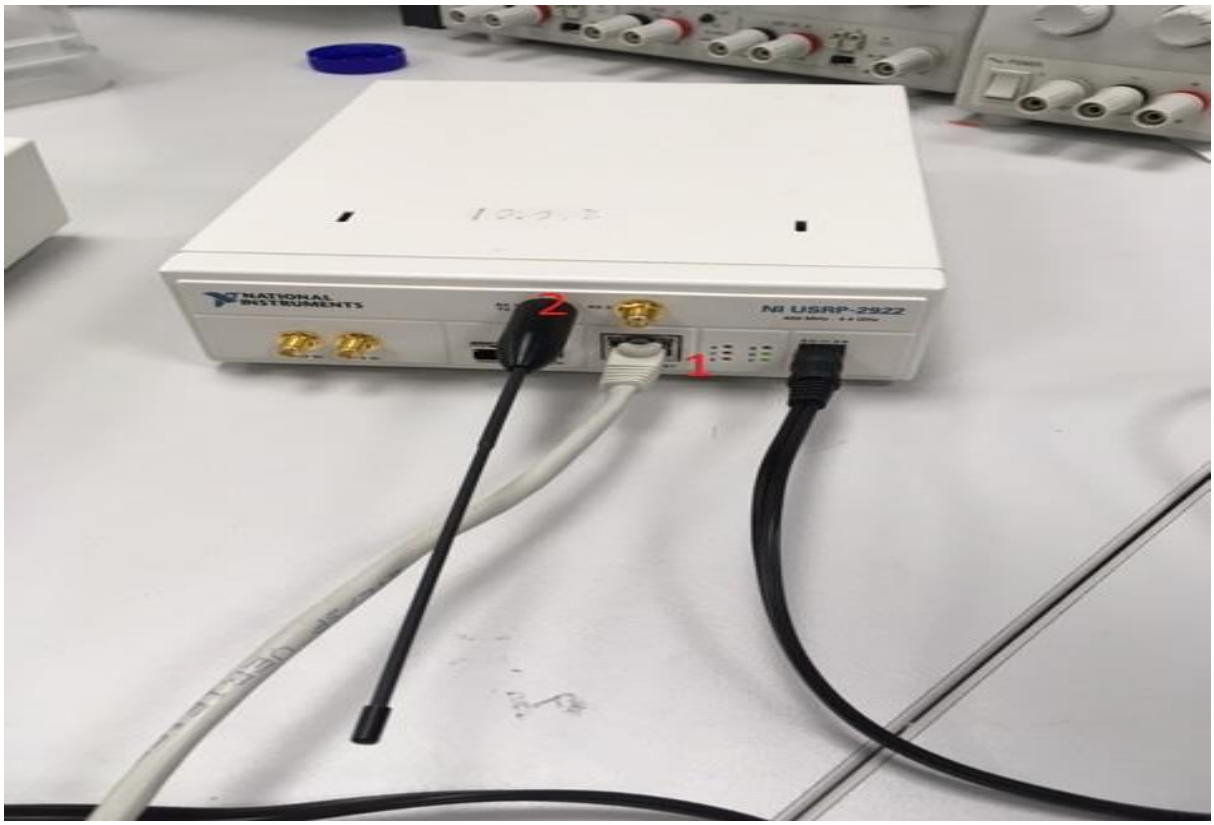


Figure 33. USRP-2922 transmitter

Figure 33 shows the hardware setup of the transmitter. An Ethernet cable (1) is connected to the PC and the USRP device. The device also has a Tx antenna (2) which is used to, later on, communicate with the receiver device.

On the front panel of the transmitter module, the carrier frequency is set to 2.405 GHz, samples/chip to 2, samples/second to 4 000 000 (in order to obtain a data rate of 250 kbps) and total bits as 512. Then the transmit button specified in section 5 is pressed the following data is observed:

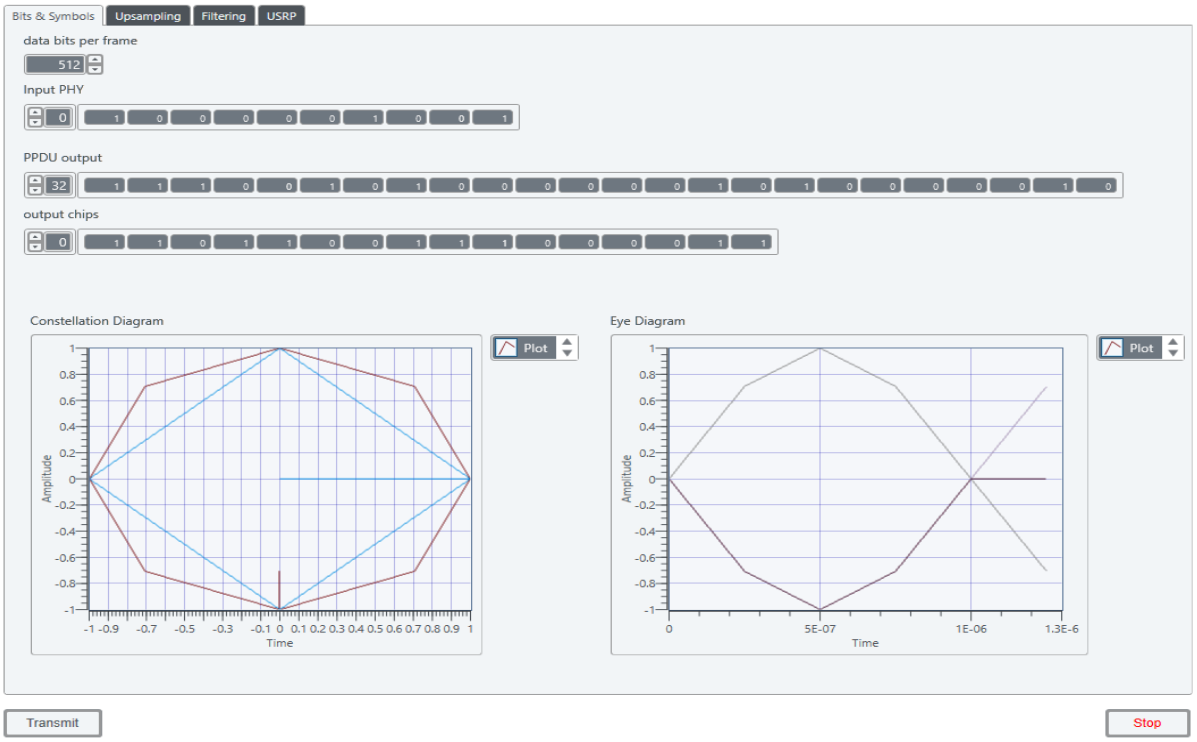


Figure 34. Front panel transmitter visualization.

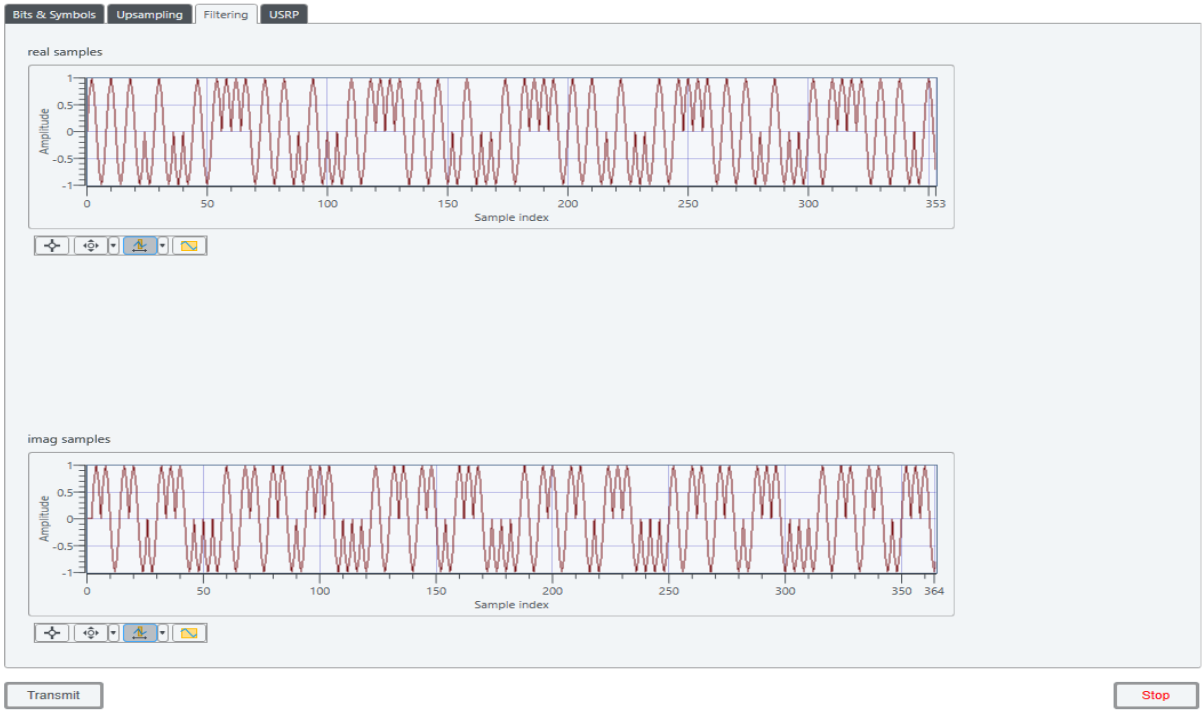


Figure 35. I and Q samples from the transmitter testing.

By comparing the results of these figures and the obtained data from the previous sections, it can be determined that the transmitter correctly modulates O-QPSK signals under the 802.15.4 standard.

### 4.3.O-QPSK receiver module implementation

This section will discuss the implementation of the receiver module VI together with all of its components

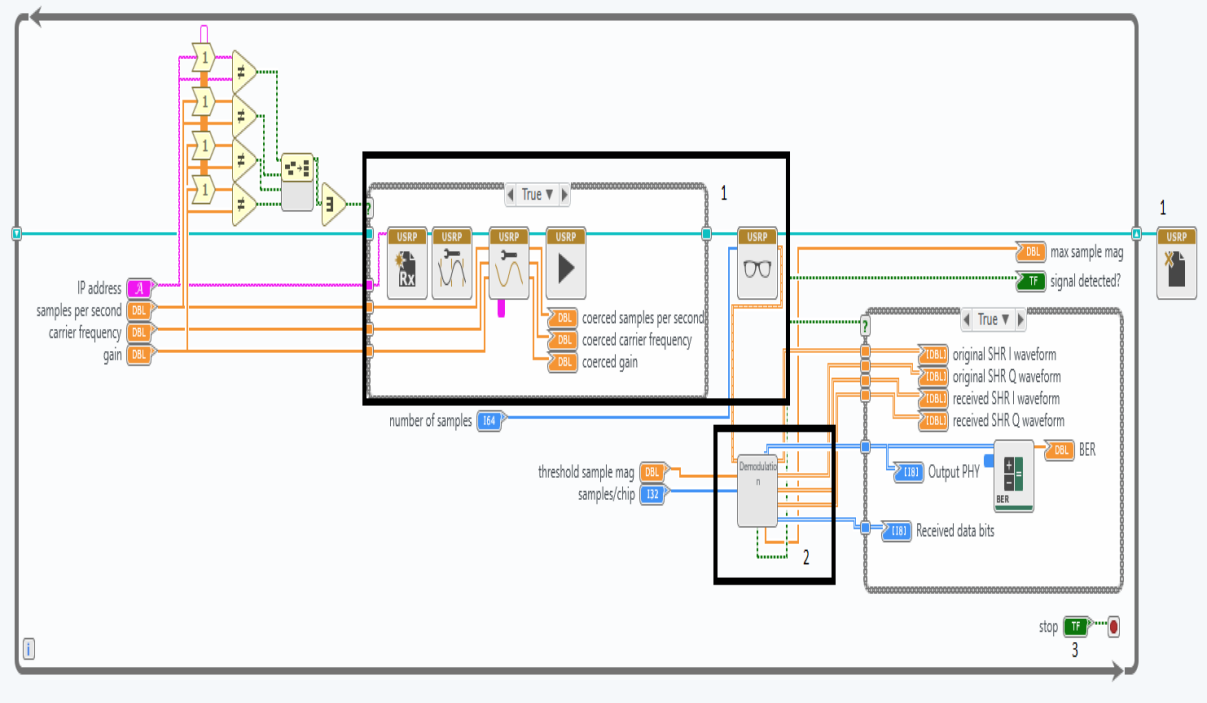


Figure 36. Receiver module block diagram.

Figure 36 shows the two main components of the receiver module, mainly the USRP settings (1) and the demodulation VI (2). This section will describe them in more detail throughout the next sub-sections. The stop button (3) terminates the program.

#### 4.3.1. USRP settings

The USRP settings shown in Figure 36 (1) have the similar structure as the ones explained in section 4.1.4. The difference here is that there is a special initiation function used to start the Rx acquisition together with a function, which fetches the session data of the transmitter, and outputs the data needed for the demodulation VI. The IP address of the USRP receiver should be specified for finding a connection on the PC. Furthermore, carrier frequency, samples per second and the gain of the received signal have to be defined.

### 4.3.2. O-QPSK demodulation

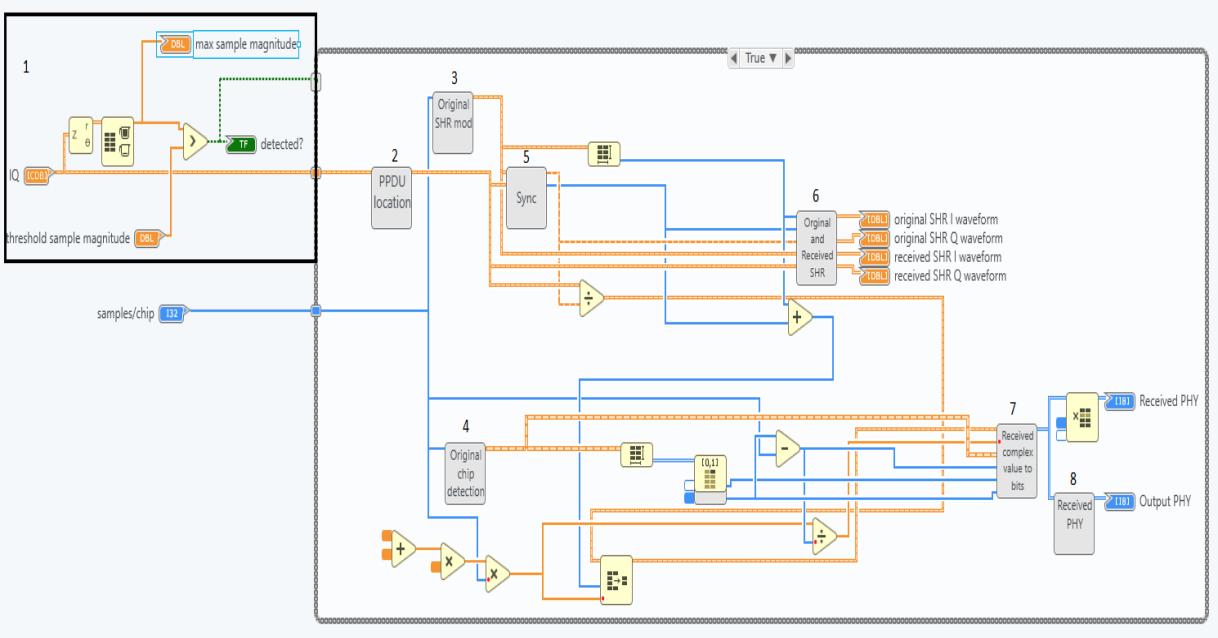


Figure 37. Demodulator block diagram.

Figure 37 shows that the demodulator consists of 8 elements – signal detection (1), PPDU location (2), original SHR modulation (3), original chip detection (4), synchronization (5), original and received SHR waveform generation (6), chips-to-bits conversion (7) and received PHY payload generation (8). The following sub-sections will describe the process of demodulation in more detail. These components will be tested in section 4.4 with the whole receiver system.

#### 4.3.2.1.Signal detection

As it can be seen in Figure 37 (1) the real part of incoming signal IQ is compared with a threshold sample magnitude value which is determined by the user so as to detect the actual signal. When the signal is detected a Boolean switch lights up on the front panel of the receiver.

#### 4.3.2.2.PPDU location

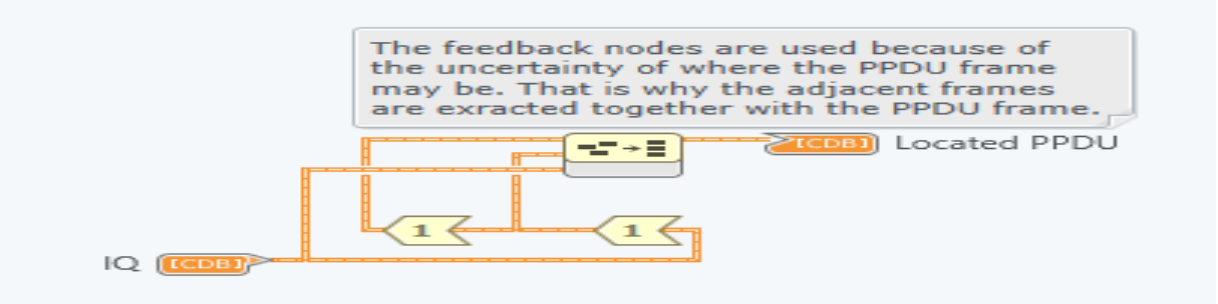


Figure 38. PPDU frame location.

After the actual signal is detected, the next step is to locate the PPDU frame. The problem with this detection is that when the frame is detected, it is not sure whether the whole PPDU data is in that frame. This is why feedback nodes are used, as shown in Figure 38, so as to extract the adjacent frames together with the actual frame in an array which then goes into the synchronization VI as shown in Figure 37 and is also used for obtaining the received bits later on.

#### 4.3.2.3.Original SHR modulation

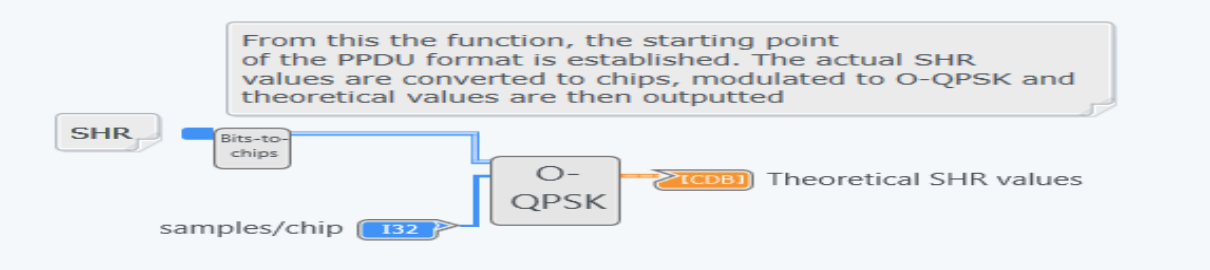


Figure 39. Original SHR modulation block diagram.

Figure 39 shows the procedure followed to obtain the theoretical values of the modulated SHR in order to obtain the start point of the PPDU format. The SHR values mentioned in section 2.3.1 are transformed to chips and then O-QPSK modulated in order to output the theoretical SHR values, which are then used to obtain the received SHR waveforms and later in used for the chips-to-bits conversion as shown in Figure 37.

#### 4.3.2.4.Original chips detection

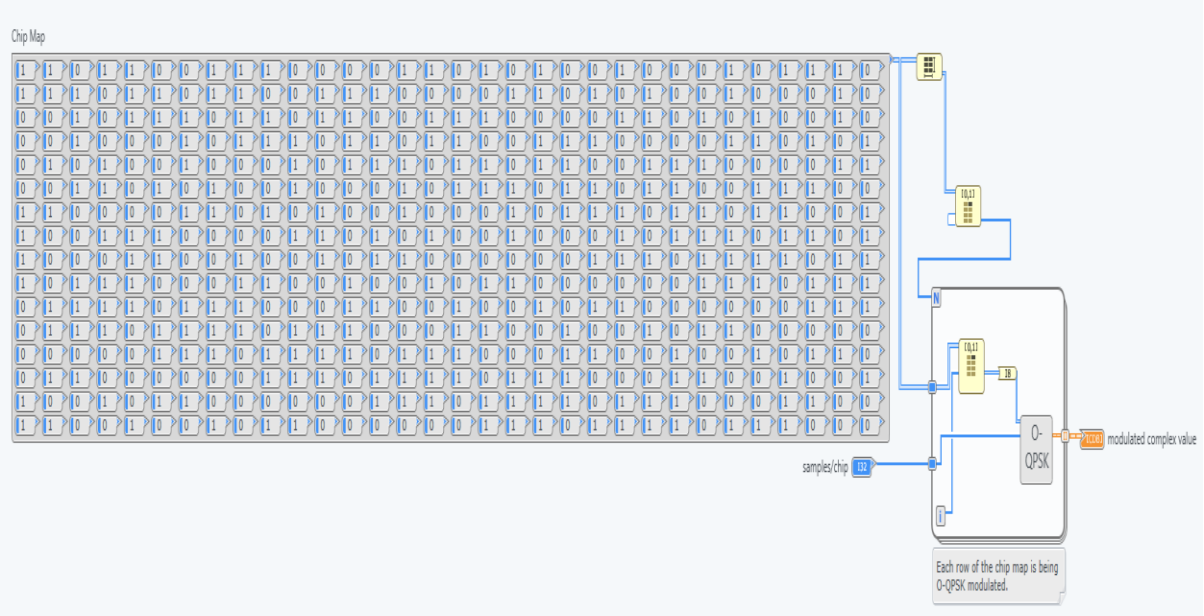


Figure 40. Original chips mapping modulation

It is not enough to only obtain the original SHR value, in order to output the PHY payload the received PPDU format must be restored in bits form. This is done by firstly obtaining the modulated chip matrix complex value as seen in Figure 40. The chips map is obtained by O-QPSK modulating every row of the chip map based on the chip mapping procedure explained in section 2.3.2.

#### 4.3.2.5.Signal Synchronization

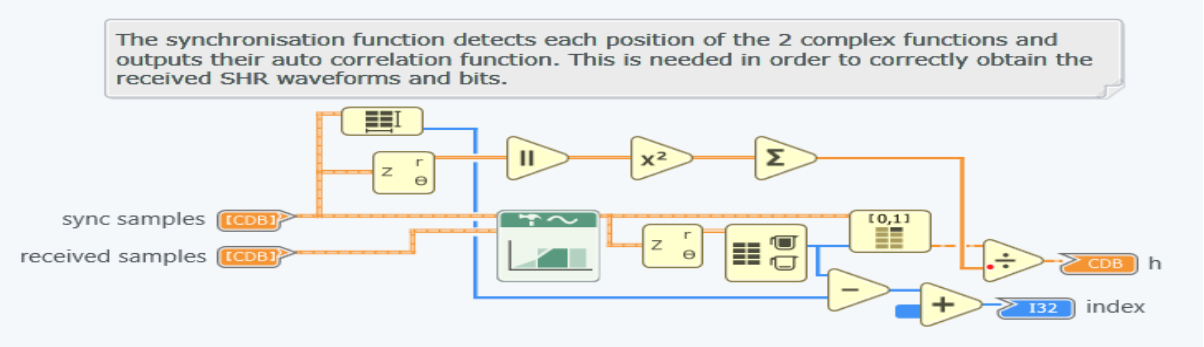


Figure 41. Synchronization block diagram.

As seen in Figure 41 the synchronization is used to obtain cross-correlation of the synchronized SHR values obtained in section 4.3.2.3 and the signal incoming from the transmitter by detecting each of their positions. This value (h) is then used in obtaining the received signal waveforms and the actual bits transmitted as seen in Figure 37.

#### 4.3.2.6.Original and Received SHR waveform generation

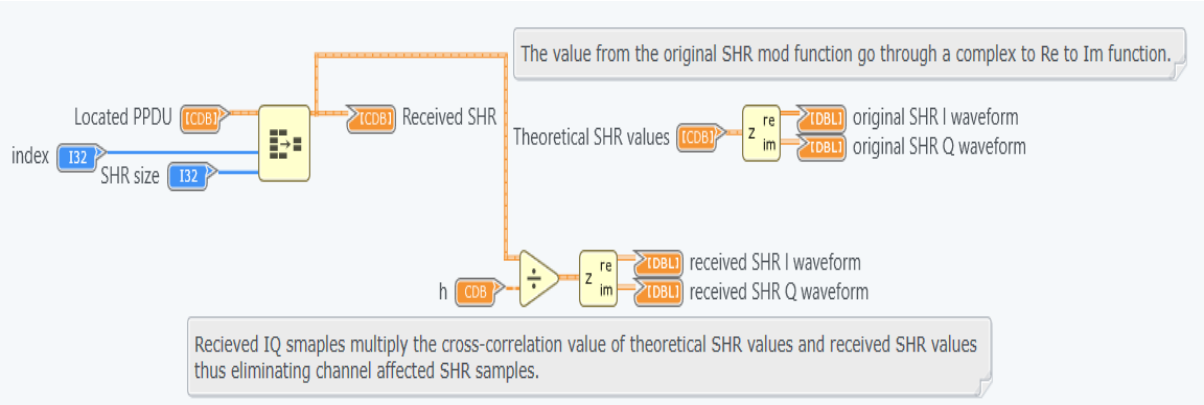


Figure 42. Original and received SHR waveform generation block diagram.

As it can be seen in Figure 42 the theoretical SHR values (section 4.3.2.3) are used to obtain the original SHR waveform both in the I and Q phases. In order to obtain the received SHR

waveforms, the received IQ samples are divided by the cross-correlation value, obtained in section 4.3.2.5, which eliminates the rotation of the samples caused by the channel. The waveforms are then outputted on the UI of the front panel. These components will be tested in section 4.4 in order to determine the accuracy of the system.

#### 4.3.2.7.Chips-to-bits conversion

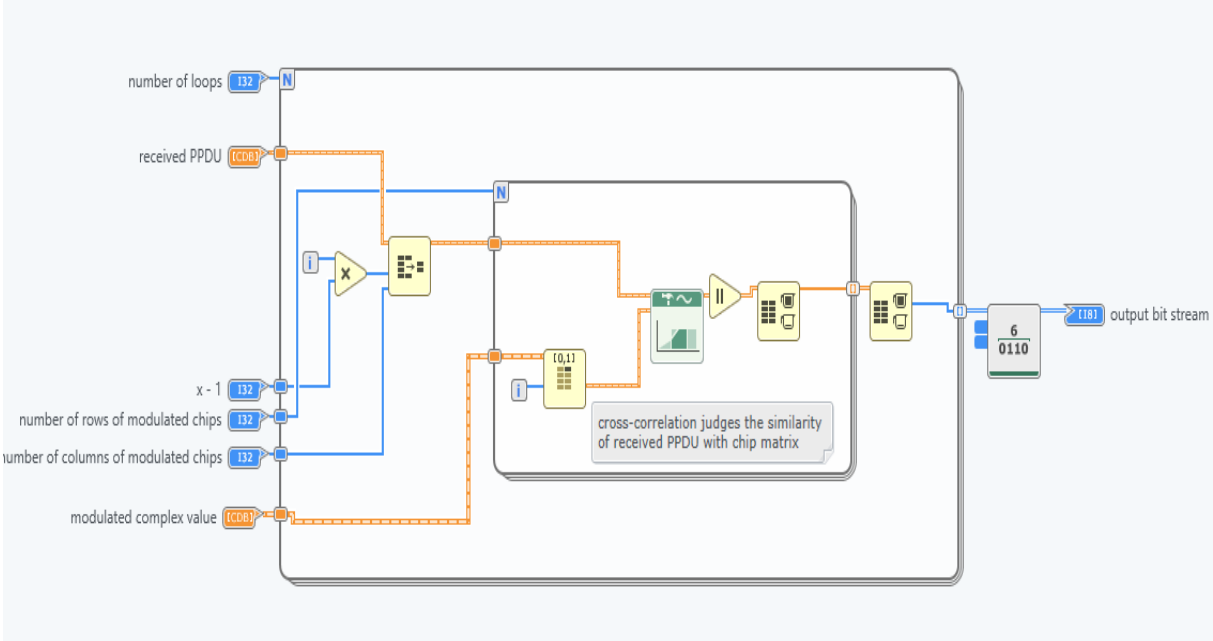


Figure 43. Chips-to-bits conversion block diagram.

As seen in Figure 43 this section consists of a cross-correlation function, which judges the similarity of the received PPDU and the chip matrix complex value, obtained in section 4.2.3.4. This is done so that the modulated PPDU values can be successfully converted into a bits form thus being able to display the transmitted bits together with the transmitted PHY payload. This output bit stream is then inputted into the Received PHY payload generation VI as shown in Figure 37.

#### 4.3.2.8.Received PHY payload generation

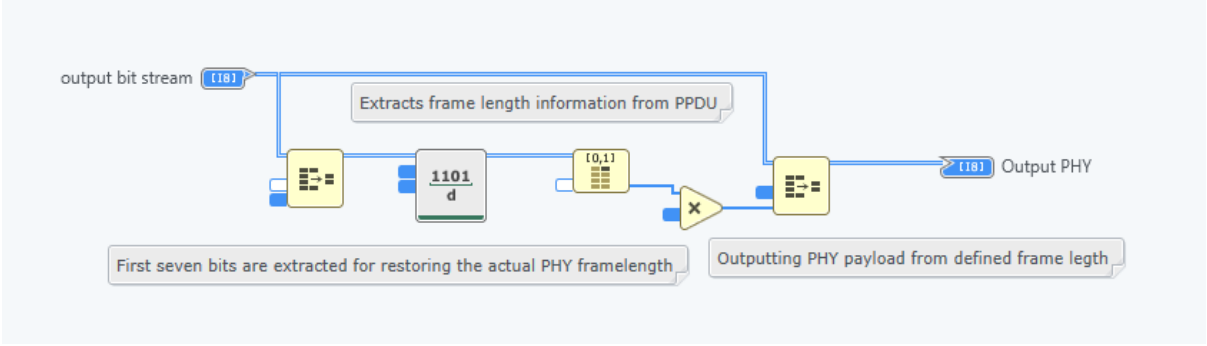


Figure 44. Received PHY payload generation function block diagram.

As it can be seen in Figure 44 the received PHY payload generation function is used to extract the actual frame length information from the PPDU format so that the correct form of the PHY payload values can be obtained. This is done by extracting the first seven-length bits (LSB first) of the output bit stream form section 4.3.2.7 and then outputting an array which is actually the PHY payload with the newly defined frame length. This PHY payload is also used to calculate the BER of the system later on.

### 4.4.O-QPSK receiver module testing

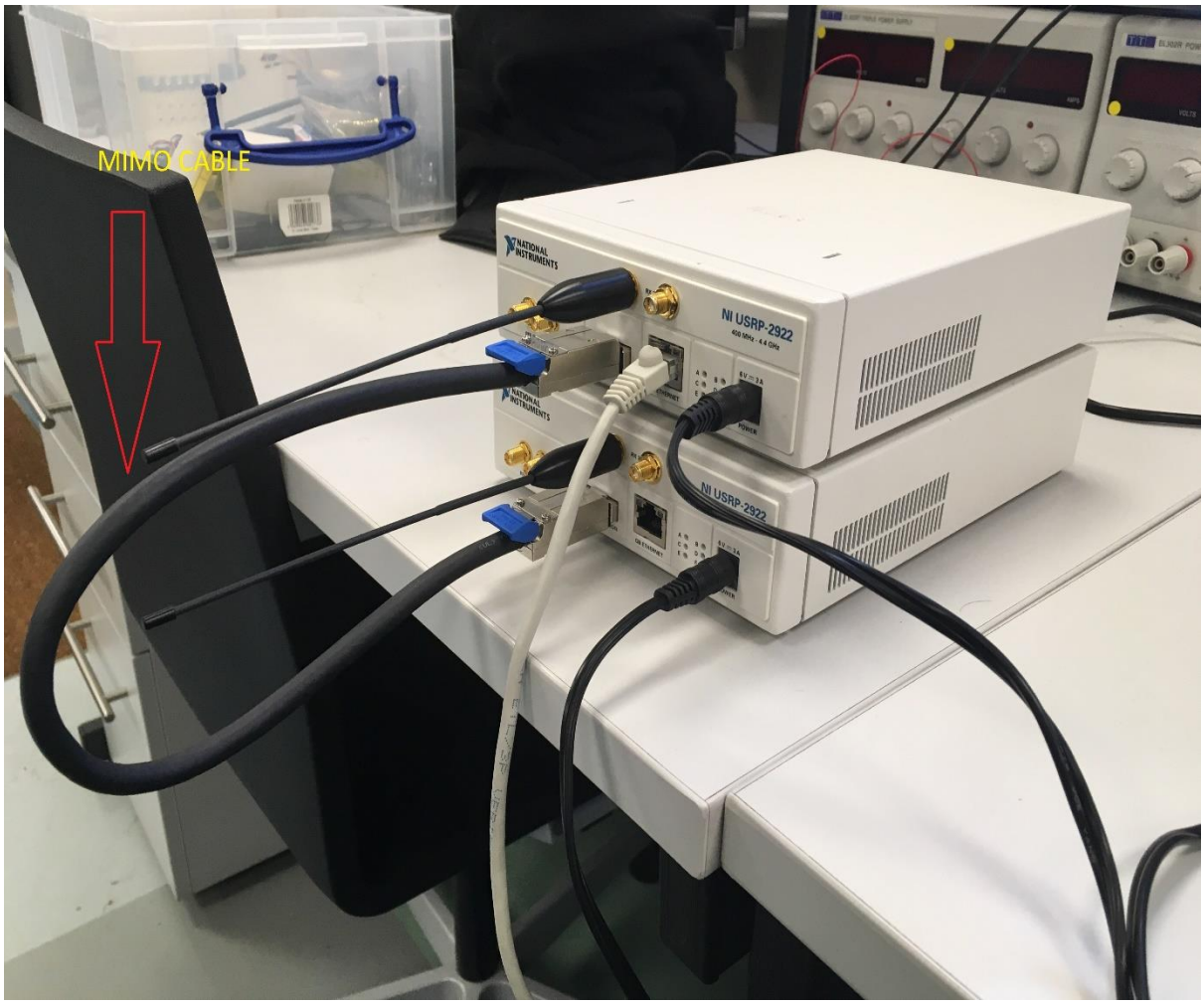


Figure 45. Receiver system set up.

Figure 45 shows how the receiver system is tested where the set up is the same as the one shown in Figure 33 with a difference that here a second USRP device is used as a receiver. The MIMO (Multiple Input Multiple Output) cable is used to share the Ethernet connection between the devices.

The frequency of the receiver is set to 2.405 GHz, samples/chip to 2, the samples/second to 4 000 000 which responds to the settings applied to the transmitter in section 5.2 in order to obtain a bit rate of 250 kbps. The sample threshold is set to 0.001.


Figure 46. Front panel of receiver module.

By inspecting Figure 56 it can be seen that the receiver correctly demodulates the transmitted data. The signal detected ? (1) Boolean lights up when the data is transmitted which shows that the input detection described in section 4.3.2.1 works as described. It can be seen that the waveforms of both the I and Q phases of the received SHR are similar to the original samples which proves that the function described in sections 4.3.2.3, 4.3.2.5 and 4.3.2.6 function correctly. They are not exactly identical due to the fact that the transceiver is being subject to changing channel environments. When the input and output PHY (3) and  are compared it can be seen that the values are identical and match the input PHY in Figure 34, which shows that the chip-to-bit conversion in the demodulator works. Finally, by observing the BER result (4), which is 0, it can be concluded that the receiver can successfully receive signals from the USRP transceiver device.

## 4.5.O-QPSK simulator implementation and testing

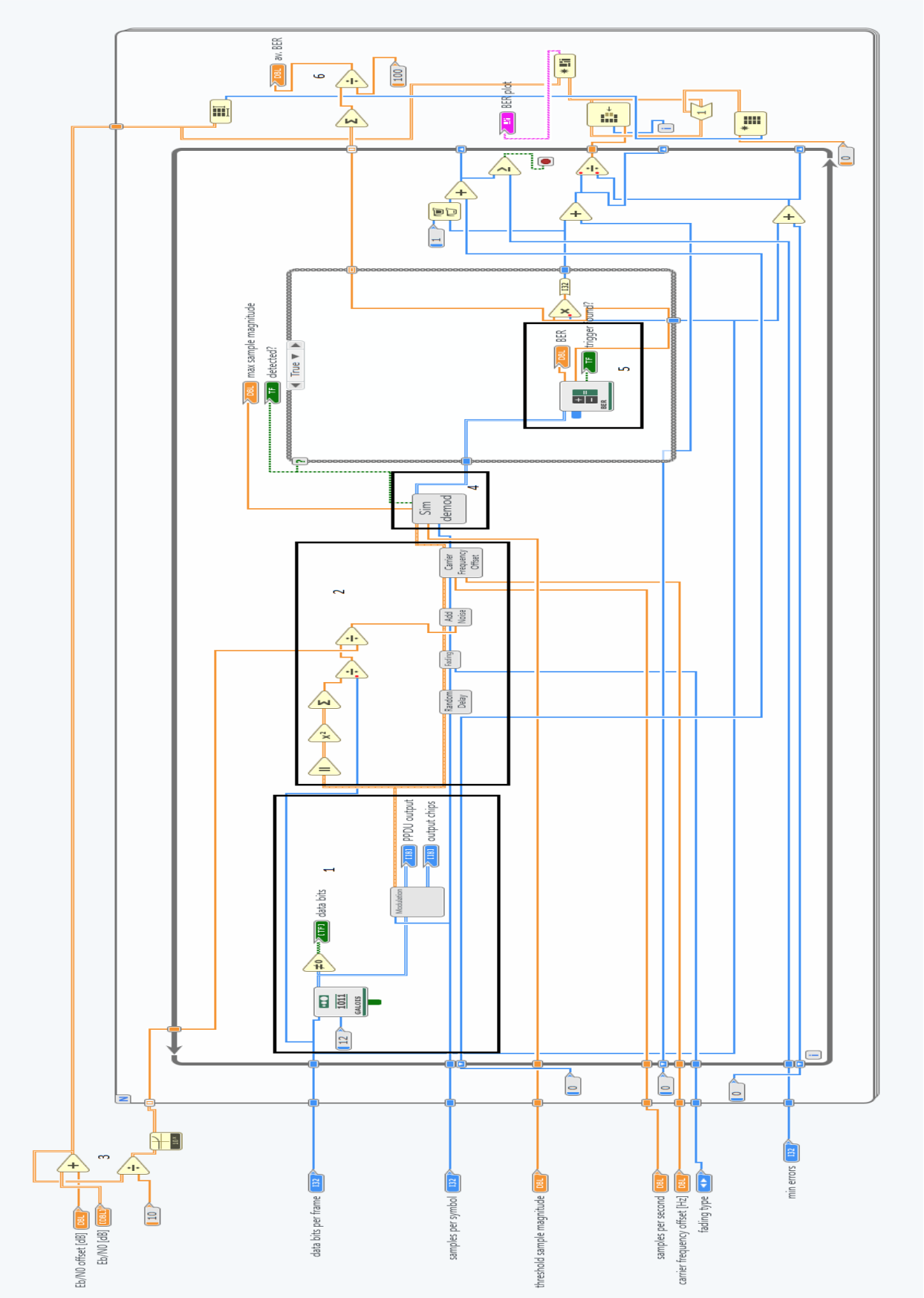


Figure 47. Simulator block diagram.

This section will look at the design and testing of the simulator module. As seen in Figure 47 the simulator is based on the design of the transceiver system without the use of USRP devices. Flat fading channel, AWGN mixing and a random delay are included into the design. The modulator and demodulator of the transceiver are copied in the simulator in order to make sure that the simulation is of the real hardware implementation. This simulator is designed to achieve theoretical bit error rates (BER) with different channel parameters so that the reliability of the designed system under the IEEE 802.15.4 can be proved. Section 4.5.1 talks about the design process and section 4.5.2 tests the simulator.

### 4.5.1. Simulator design

As it can be seen from Figure 47 the input generation and modulation (1) are the same as the ones shown in section 4.1. Here the data bits have to be between 72 and 1016 to meet the requirements stated in section 3.1. The 2nd part of the simulator is the actual simulation of the channel. First, it adds a random delay to the system, followed by fading function, which mixes the received waveform with the generated fading channel (Rayleigh fading). Afterwards, the signal goes through AWGN function. This function adds noise to the waveform based on the equation $N0 = \sum|s_i|^2 \div \frac{Eb}{N0}$ where $\sum|s_i|^2 = Eb \rightarrow N0 = Eb \times \frac{N0}{Eb} = N0$ where $\frac{Eb}{N0}$ (3) is defined by the user. This value is also used in the 6th element in order to calculate the average BER. The channel simulation also includes a carrier frequency offset function, which is used to test how much carrier frequency offset can the system sustain before the BER goes to 1. The demodulator of the simulator (4) is the same as the one described in section 4.3.2. The only difference is that the PPDU location from Figure 37 – (2) is removed in order to sample the frames one at a time to obtain a reasonable BER. Finally, the BER is calculated and a BER plot (5), based on a minimal number of error (5), is displayed on the front panel.

### 4.5.2. Simulator testing


Figure 48. Simulator testing.

As it can be seen from Figure 48 the PHY payload bits can generated and demodulated through different fading channels with AWGN. This means that the average BER value and the BER plot can be obtained in different channel conditions. In this case 512 bits (1) are selected with 2 samples/symbol (2) and 4 000 000 samples/second (3) in order to meet the 250 kbps data rate specified in section 2.3.3. The Eb/N0 (4) represents the SNR and is directly connected to the BER ratio. In this simulator an Eb/N0 offset (5) is used in order to simplify the UI. When the minimal errors (7) value is increased the BER plot becomes smoother and the average BER value more precise but the program takes longer to simulate. Different sample values, fading channels (9), carrier frequency offsets (8) and  are discussed in section 5.

### 4.6.XBee modules set-up and implementation

This section will discuss the steps undertaken to implement the XBee modules into the transceiver system.

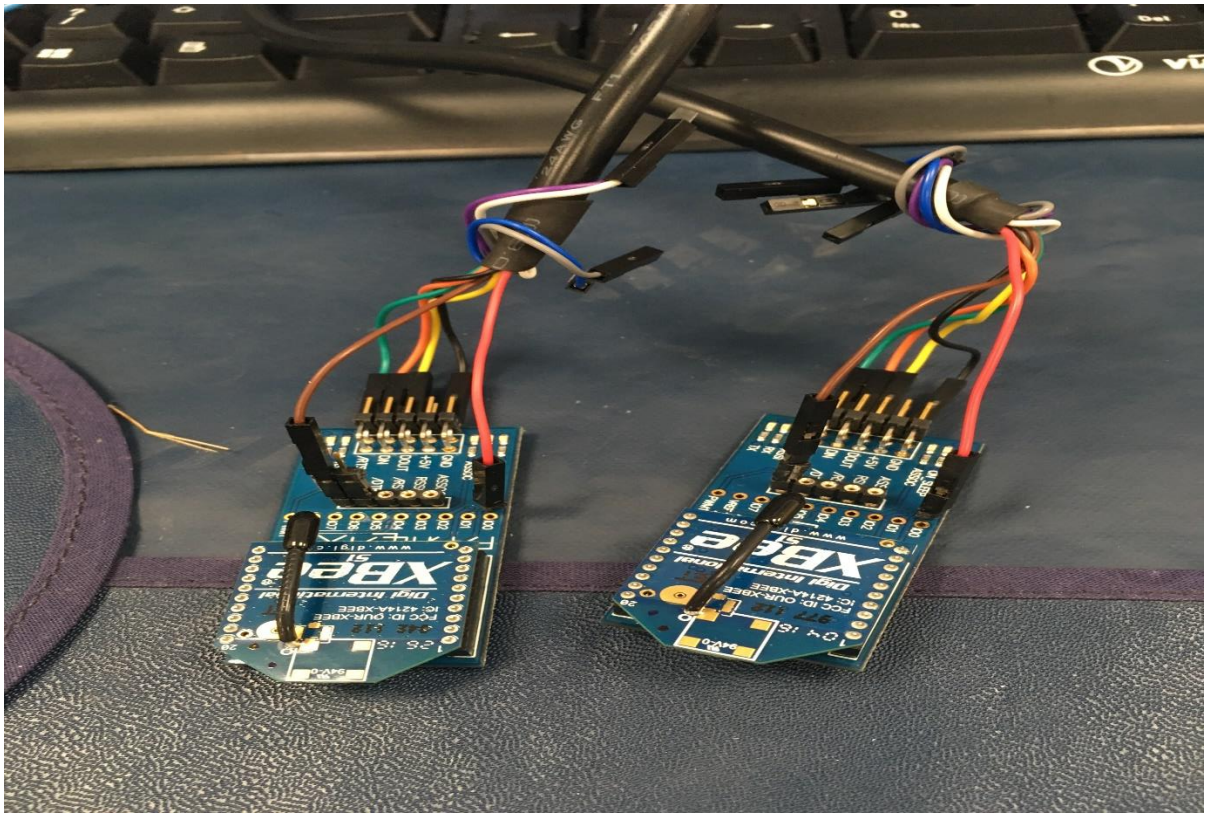

Figure 49. XBee connection to the PC using FTDI cables.

Figure 49 shows the set-up of the connection between the XBee devices and the PC. The modules are connected to the PC via FTDI cables (1). The FTDI cables have 10 wires for serial connection each serving a different purpose. For this project, only 6 of them are needed:

- GND (black) – Device ground supply pin.

- POWER (red) – Power output which can be customized to output +3.3V or +5V.
- RTS (green) – Request to send control output (Handshake signal).
- CTS (brown) – Clear to send control output (Handshake signal).
- TXD (orange) – Transmit asynchronous data.
- RXD (yellow) – Receive asynchronous data.

The XBees are connected to FTDI cables via adapters (2), which have the same pin set-up and serve as a convenient interface for the modules.

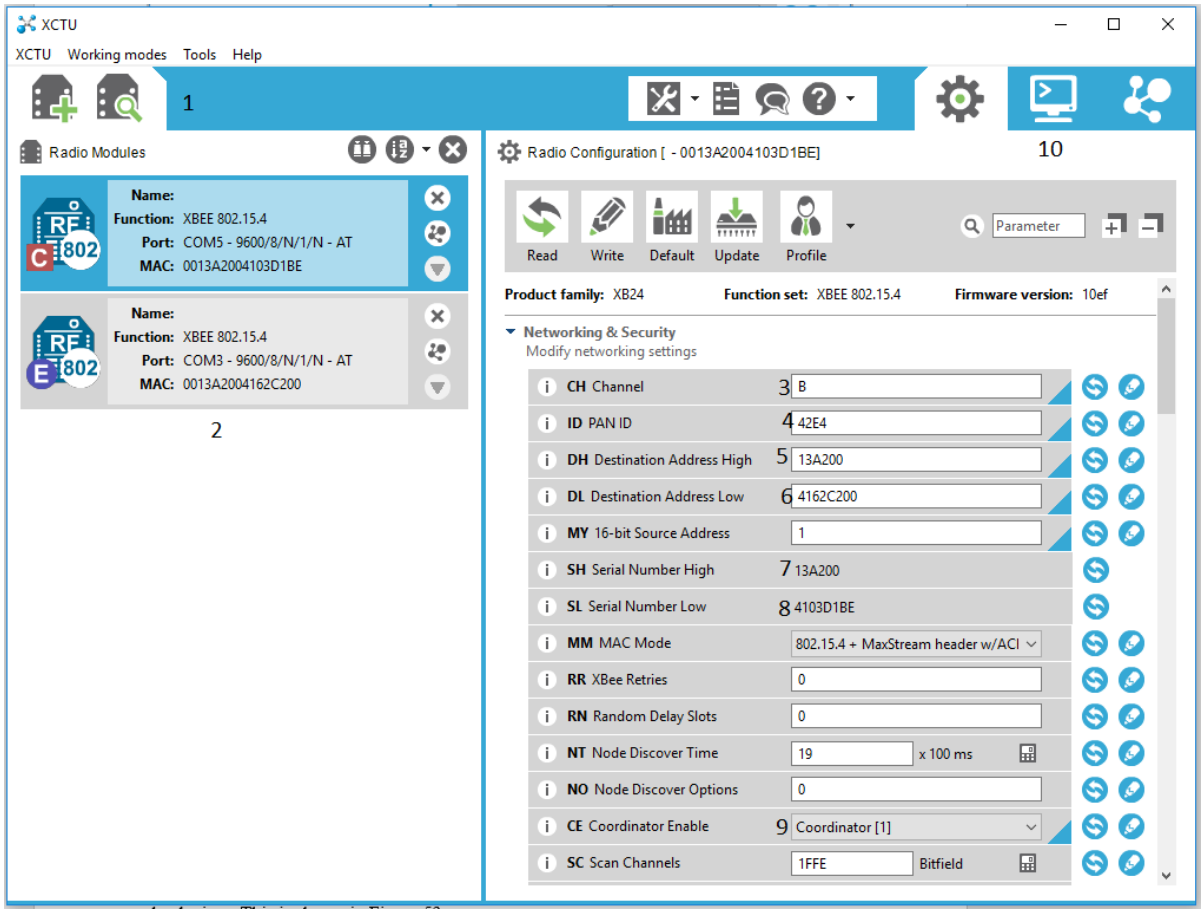


Figure 50. XCTU user interface.

In order to interface the XBee modules with the PC a free multi-platform application, called XCTU [30], is used. The program is specially designed for XBee modules and offers a simple-to-use graphical interface as shown in Figure 50. The first step for interfacing with the XBee modules is to detect the devices (1). When detected they appear as separate icons in the Radio Modules field (2). In order to establish a connection between the XBee's, their channels (3) and PAN (Personal Area Network) have to have the same value. Here the channel value goes from 0x0B to 0x1A (11-26) which represent the channel numbering discussed in section 2.3.3. Furthermore, the destination address DH and DL (5 and 6) on one of the XBees have to be

identical to the serial numbers SH and SL (7 and 8) on the other XBee and vice-versa. In order to meet the 802.15.4 network topology requirements described in section 2.2, one of the XBee modules is set to be a PAN coordinator (9) in order to establish a star topology. Finally, the console working mode (10) is entered so that data can be inputted and transmitted between the devices. This is shown in Figures 51 and 52.

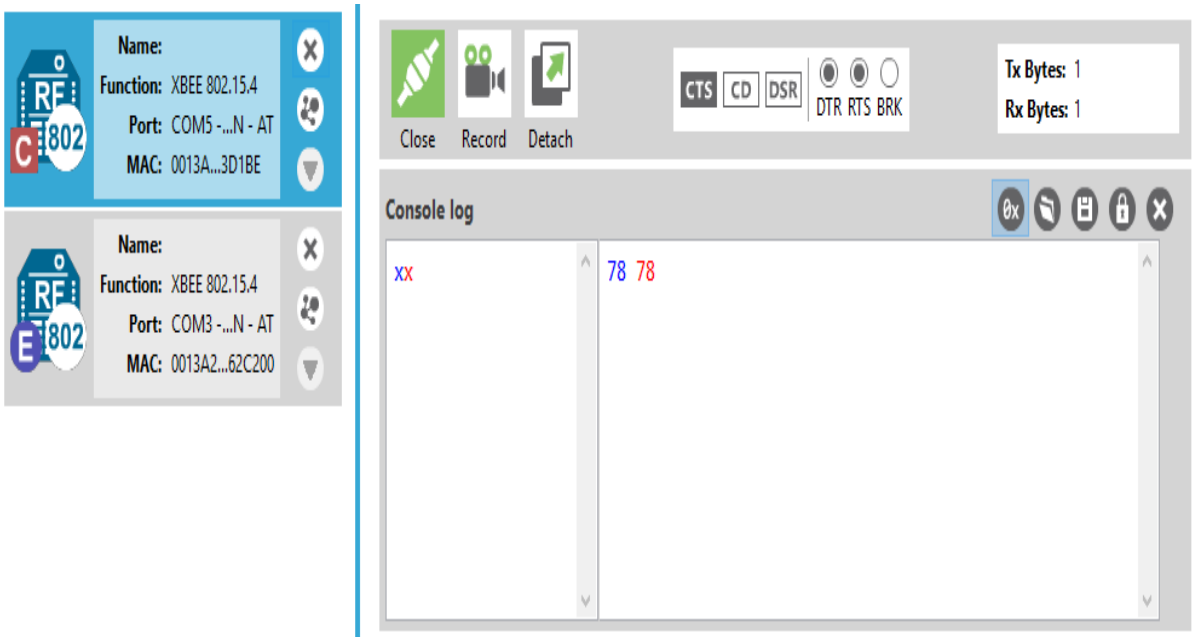


Figure 51. Coordinator sending data (blue) and receiving data (red).

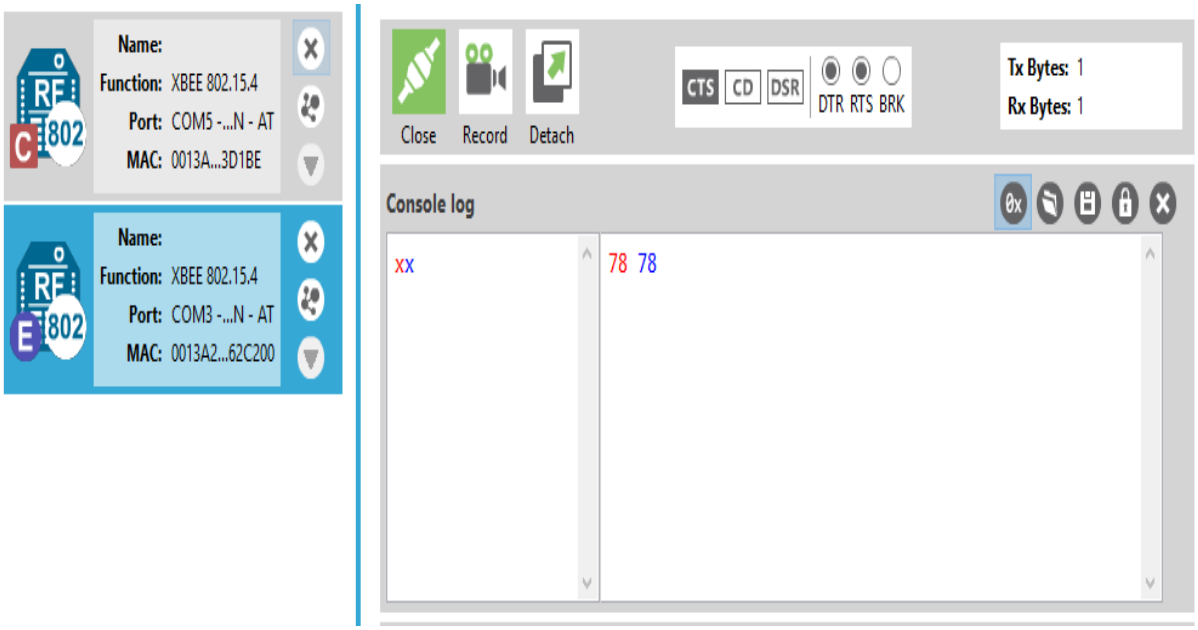


Figure 52. End device receiving data (red) and sending data (blue).

### 4.7.Transmission detection between the USRP receiver and the XBee modules

This section will describe how the receiver module described in section 4.3 detects data the transmission data from the communication of the two XBees. The receiver module is configured in the same way as described in section 4.4. The carrier frequency is set to 2.405 (equation 1 in section 2.3.3) in order to match the XBee channel used, which in this case is B (11 in decimal).

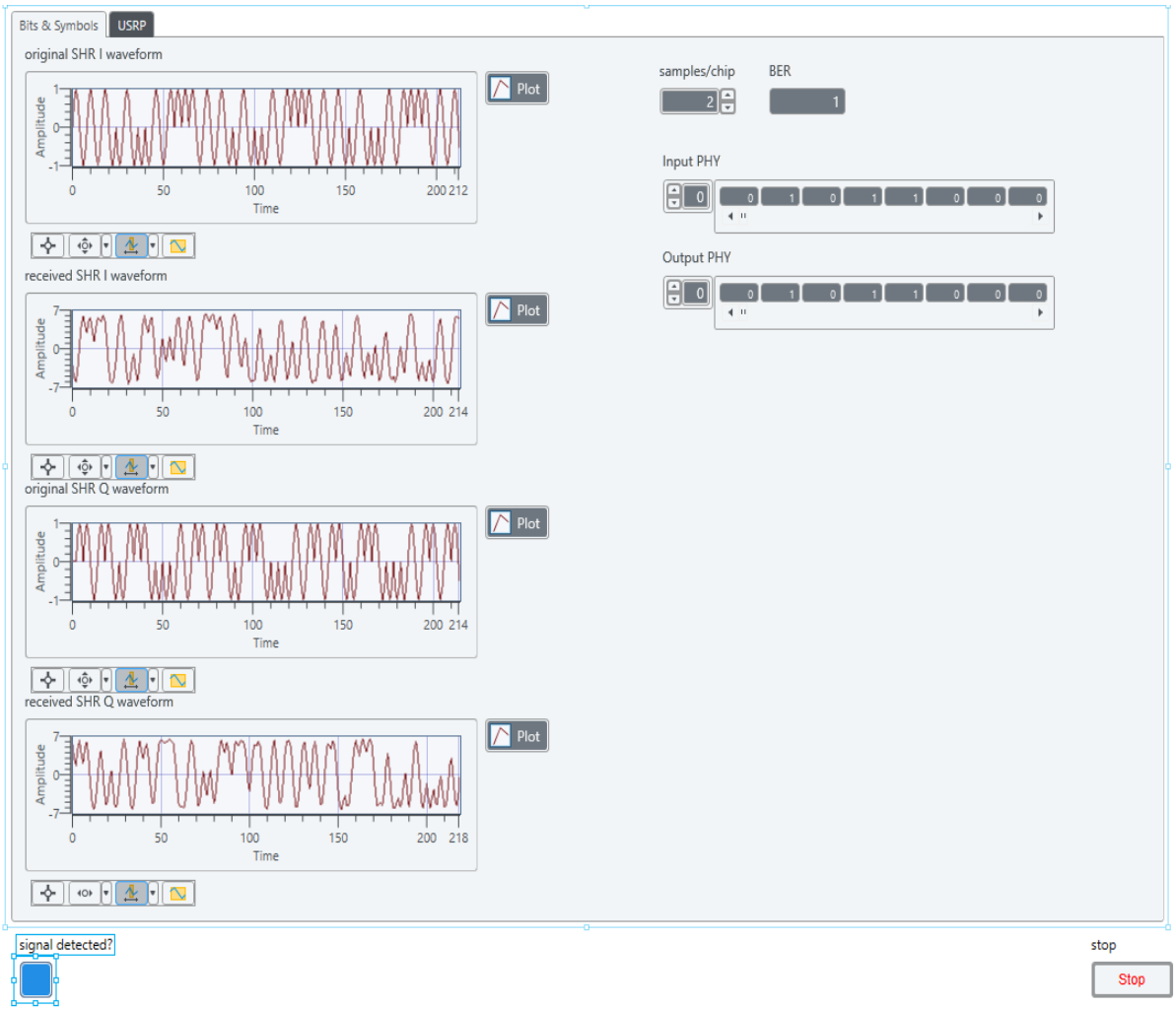


Figure 53. Receiver detection of the communication between the XBee devices.

By inspecting Figure 53 and comparing it with Figure 46, it can be seen that the receiver module successfully detects the communication between the two XBees from Figures 51 and 52. The signal detection button lights up and waveform samples are obtained. However, the data obtained is not the correct representation of the transmission due to the fact that the carrier offset frequency between the USRP device and the XBees is too high which leads to high BER results as explained in section 5.3.

## 5. BER Results

This section presents BER results obtained from the simulator described in section 5.5, which tests the transceiver system designed in section 4, under the IEEE 802.15.4 standard with different flat fading channels, sample rates, frequency offsets and SNR values. All of these parameters can be defined on the front panel of the simulator as described in section 4.5.2. The BER plots and values obtained with all these parameters can be compared with academic literature, thus guaranteeing the reliability of the designed system.

### 5.1.BER results with different flat fading channel conditions

Two different Rayleigh fading conditions are observed, namely block narrowband fading and uncorrelated fading, where Rayleigh fading is model of a propagation environment on a RF signal.

512 bits are selected with 2 samples/symbol and 4 000 000 samples/second and the SNR is set between 10 and 20 dB. BER performance is obtained on BER versus Eb/N0 (dB).

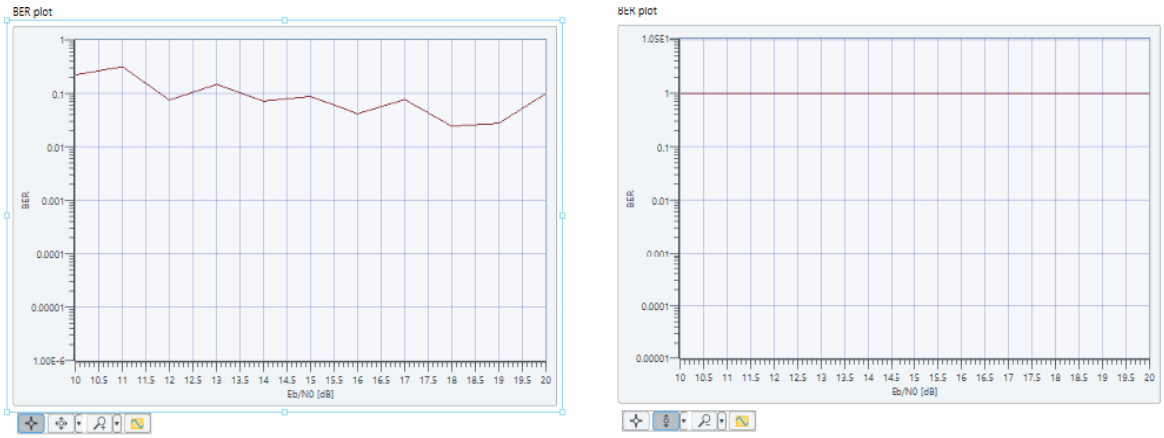


Figure 54. Simulator BER results obtained for narrowband (1) and uncorrelated fading (2).

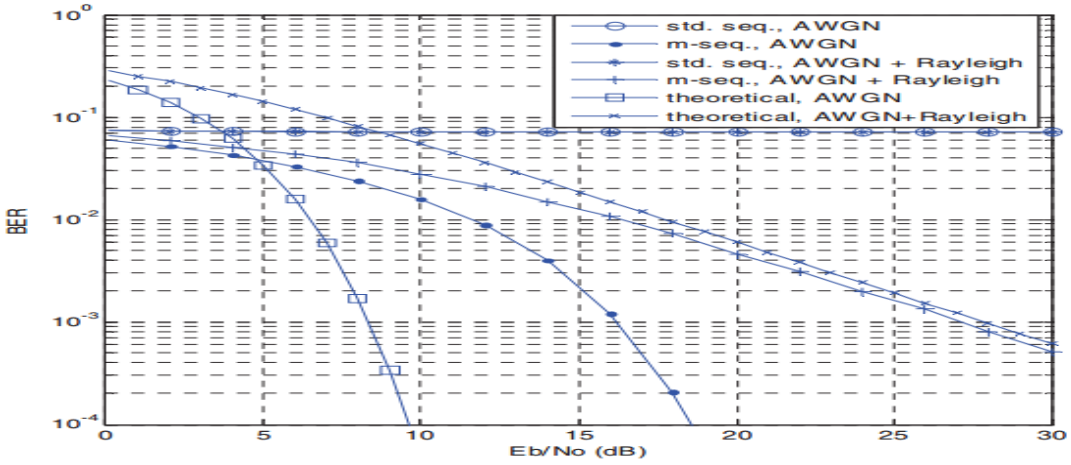


Figure 55. BER values of IEEE 802.15.4 [31]

By comparing Figure 54 and 55 it can be seen that the simulation results are similar with the theoretical values for a BER plot, which proves that the module correctly calculates the BER under different channel fading profiles.

### 5.2.BER plots under different SNR ratios

In this case BER plots are obtained with the SNR going from 0 dB to 15 dB. No fading is chosen and 512 bits are selected with 2 samples/symbol and 4 000 000 samples/second.

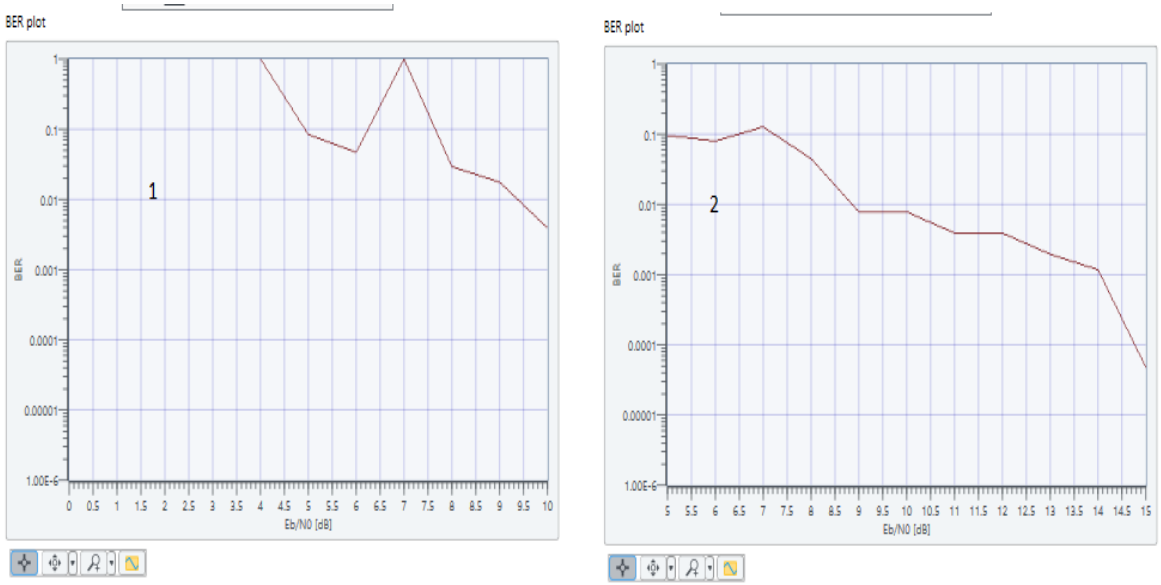


Figure 56. Simulated BER plots with SNR ratios having values between 0-10 dB (1) and 5-15 dB (2).

Figure 56 shows that the BER decreases as the SNR increases. This is so because the higher the Eb/N0 ratio, the less the transmitted signal is affected by noise. This leads to a small error probability. By comparing these results with Figure 55 it can be seen that the simulation BER plots follow the same tendency of having a high BER for low SNR and decreasing as Eb/N0 increases.

### 5.3.BER plots with different frequency offsets

The carrier frequency offset represents the difference between transmitting and receiving frequency and when it increases over a certain value the BER worsens due to the fact the gap between transmitted and received signal becomes very large.

No fading is chosen and 512 bits are selected with 2 samples/symbol and 4 000 000 samples/second and the SNR is set between 10 and 20 dB. BER performance is obtained on BER versus Eb/N0 (dB). Frequency offsets of 2 000 Hz and 3 500 Hz are chosen.

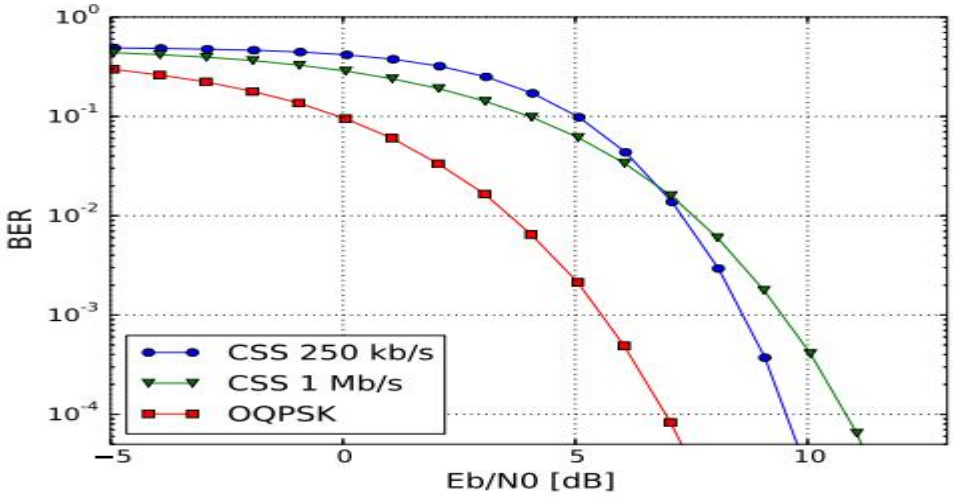


Figure 57. Theoretical BER value in AWGN for O-QPSK transmission [32].

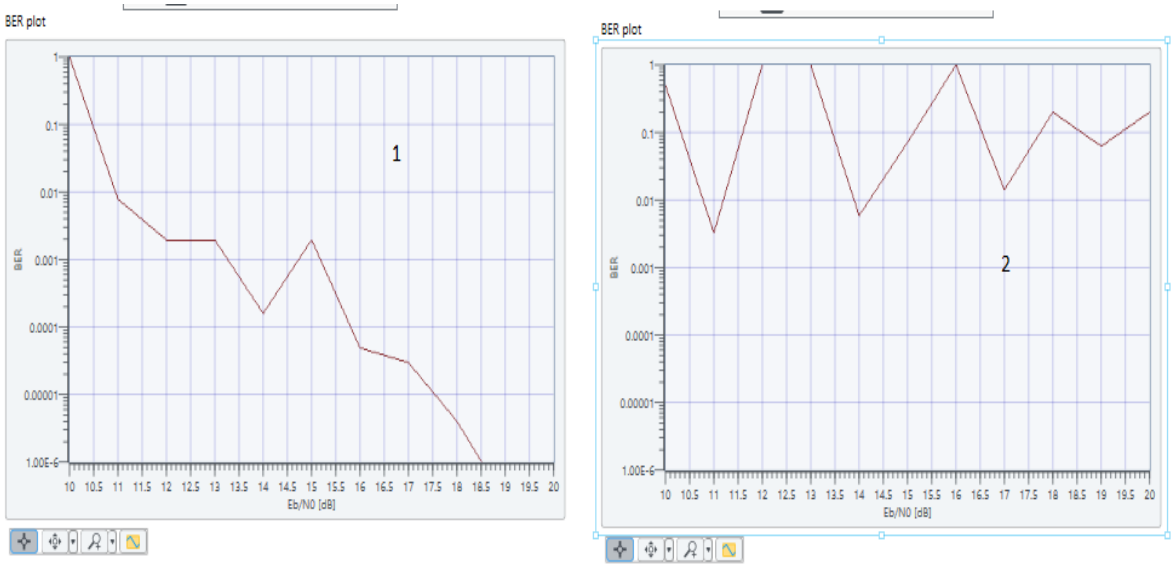


Figure 58. Simulated BER results for CFO of 2000 (1) and 3500 (2).

Figure 58 shows that as the CFO increases the BER performance worsens because the transmitted data is affected by the channel conditions more seriously. When compared with Figure 57 it can be seen that the simulated results produce worsened BER plots. This degrading BER performance can be observed in section 5.7 where the high CFO between the XBee and the USRP produce erroneous results.

### 5.4.BER plots with different sample values

According to the IEEE 802.15.4 standard the data rate for transmitting in the 2.4 GHz has to be 250 kbps. In order to obtain this the chip rate has to be set to 2 000 000 as described in section 4.1.3.

No fading is chosen and 512 bits are selected with 0 CFO and the SNR is set between 10 and 20 dB. The samples/second values are 4 000 000 and 8 000 000, paired with 2 and 4 samples/chip respectively. BER performance is obtained on BER versus Eb/N0 (dB).

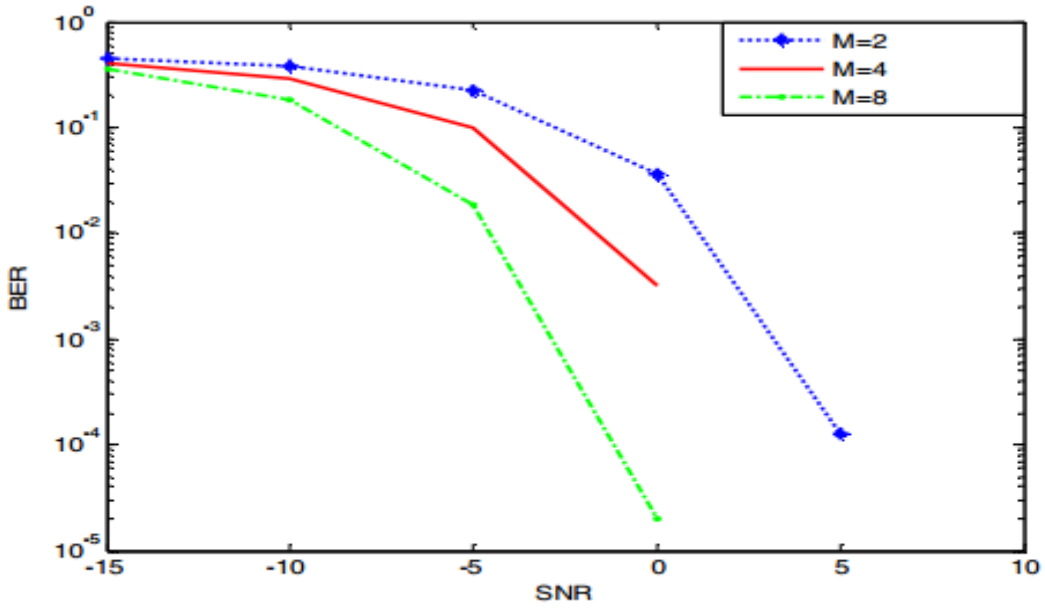


Figure 59. Theoretical BER plots for different number of samples [33].

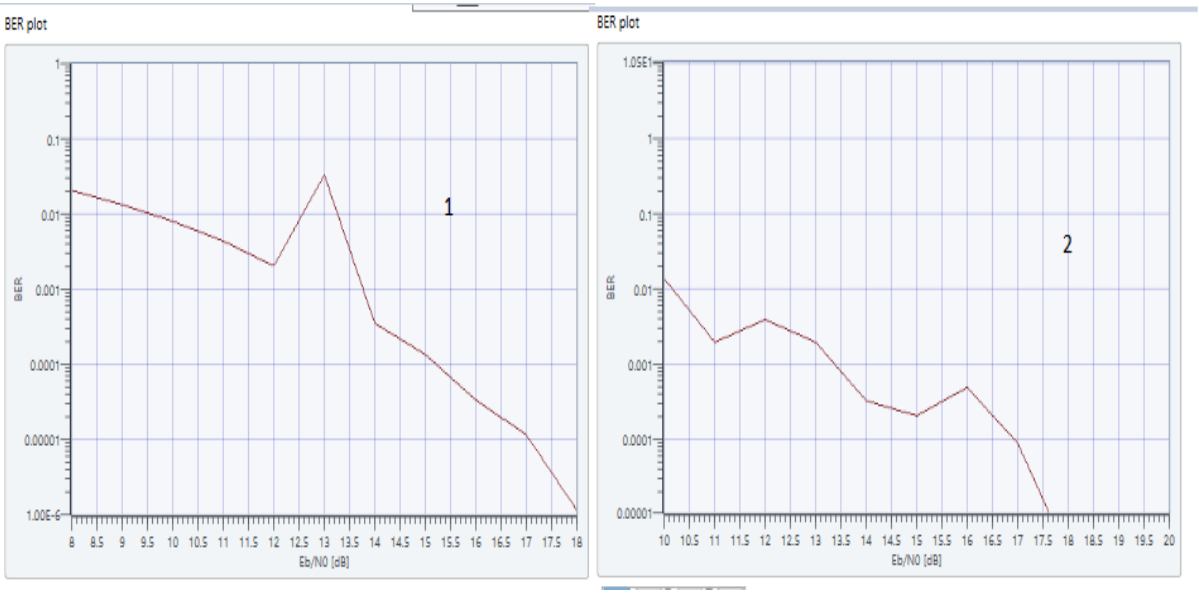


Figure 60. Simulated BER plots for 2 samples/chip (1) and 4 samples/chip (2).

From Figure 60 it can be seen that the more samples generated, the less the signal is affected by the channel conditions. This is so because when more samples are generated from the modulator, more complex values are received by the demodulator for a more precise restoring of the PHY payload. The BER plots from Figures 59 and 60 are close to identical and show the same tendency thus proving the correct operation of the system.

# 6. Reflection

This section will comment on how well the planning section (section 3) has been followed and if all of the task, criteria and time management tasks have been achieved.

## 6.1.Project scope and success criteria

The project successfully meets the scope specified in section 3.1 together with all of the success criteria listed in section 3.2. It creates a fully-worked lab experiment for students and is entirely based on the IEEE 802.15.4 standard. LabVIEW and the USRP modules have been successfully combined in order to create a transceiver system. Furthermore, the simulator module correctly proves the operation of the system by achieving results close to the theoretical ones. Finally, the detection of the communication between the XBee modules has been achieved.

## 6.2.Project tasks and resources

Every task listed in section 3.3 has been achieved and all of the resources have been used to their full extent.

## 6.3.Project risk assessment and time management

All of the proposed risks have been carefully noted and all difficulties encountered have been dealt with as planned. Debugging problematic software components has proven to be the most difficult aspect of the project and is something that the author will work on for his future engineering endeavors by carefully backtracking all of his work and focusing on each element rather than trying to debug the whole system. The schedule for the project, proposed in section 3.1, has been followed with little or no changes.

# 7. Conclusion

This project successfully implements a transceiver system under the IEEE 802.15.4 standard by using LabVIEW, USRP and XBee modules. The system carefully follows and implements the steps required to achieve O-QPSK modulation and demodulation, and then uses these components to create a transmitter and receiver which when combined with the USRP modules

creates the transceiver system. Furthermore, a simulator is created which tests the reliability of the system without the use of USRP modules and shows the BER performance of the system. Finally, the communication between two XBee modules is detected using the receiver module of the system.

### 7.1.Project value

The above described project is meant to serve as a lab exercise for university students learning about wireless communications. It provides explanations of each step required to build a communications system based on the IEEE 802.15.4 standard. Furthermore, it incorporates a simulator module, which can be used to prove the reliability of the designed transceiver system. What is more, it allows the user to test the system using actual RF modules and detect communication between them and the transceiver. Finally, this project can be used as a test platform for system designers who will use the 802.15.4 standard.

### 7.2.Future work

For the future development of the project, the MAC layer [1] of the IEEE 802.15.4 standard will be incorporated in the design as for now the design focuses only on the PHY layer. The MAC layer uses the information of the PHY layer so that it can complete the transmission between the devices.

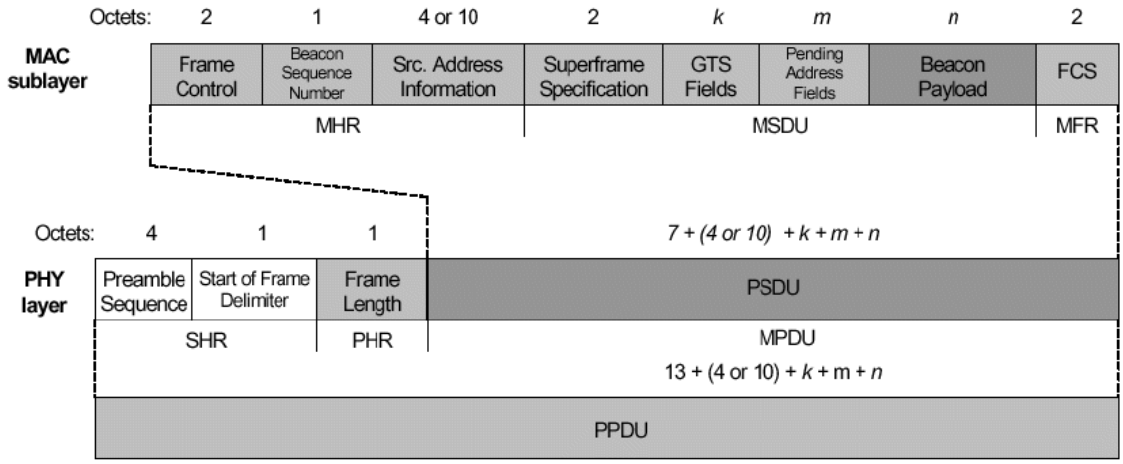


Figure 61. PHY and MAC layer of the 802.15.4 standard [1].

Figure 61 shows how the MAC layer is formed based on the PSDU field of the PHY layer. The future work for the project will change the components of the design in order to combine both PHY and MAC layer information thus establishing full wireless communication between the device.

## References:


[1] - IEEE Computer Society – "IEEE Standard for Low-Rate Wireless Networks" – IEEE Std. 802.15.4- 2015 Revision of IEEE Std. 802.15.4, Published 5 Dec 2015

[2] – Feng Xia, Laurence T. Yang, Lizhe Wang, Alexey Vinel – "Internet of Things" in the International Journal of Communication Systems, Published online in Wiley Online Library.

[3] – Digi International Inc –"XBee/XBee Pro RF Modules" – [Online] – Available: https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf

[4] – Yao-Hong Liu, Tsung-Hsien Lin – "A 3.5-mW 15-Mbps O-QPSK transmitter for real-time wireless medical imaging applications" in Custom Integrated Circuits Conference, 2008. (CICC 2008, IEEE), Published 17 Nov 2008, p-p 599-602

[5] – National Instruments – "NI USRP-292x/293x Datasheet" – [Online] – Available: http://www.ni.com/datasheet/pdf/en/ds-355

[6] – Marina Petrova, Andreas Achtzehn, Petri Mahonen – "System-Oriented Communications Engineering Curriculum: Teaching Design Concepts with SDR Platforms" in IEEE Communications Magazine, Vol: 52, Issue 5, 2014, p-p 202-209

[7] – Todor Cooklev – "Wireless Communication Stadards – A Study of IEEE 802.11, 802.15, and 802.16" in Standards Information Network, IEEE Press, 2004, p-p 1-5

[8] – National Instruments – "LabVIEW Communications System Design Suite Overview" – [Online] – Available: http://www.ni.com/white-paper/52502/en/

[9] – IEEE ComSoc – "IEEEComSoc Hans-On Lab Exchange" – [Online] – Available: http://labs.comsoc.org/

[10] – TechTarget – "802.11 PHY Layers" – p-p 214 – Available online: http://media.techtarget.com/searchMobileComputing/downloads/CWAP_ch8.pdf

[11] – Nam-Jin Oh and Sang-Gug Lee – "Building a 2.4-GHZ Radio Transceiver using 802.15.4" in IEEE Circuits and Devices Magazine Vol. 21 Issue 6, 2006

[12] - Bhumika, Dr. Mukesh Kumar, Dr. Sudhir Narayan Singh, Bhanuj Taneja – "Relative Study of Zig-Bee/IEEE 802.15.4 WPAN Performance under Different Topologies" in Advanced Computing & Communication Technologies (ACCT), Published 2015, p-p.415

[13] - Marina Petrova, Janne Riihijarvi, Petri Mahonen, Saverio Labella – "Performance Study of IEEE 802.15.4 Using Measurements and Simulations" - RWTH Aachen University, 2009

[14] – Ivan Howitt, Jose A. Gutierrez – "IEEE 802.15.4 Low Rate – Wireless Personal Area Network Coexistence Issues" in Wireless Communications and Networking, Published March 2003, p-p 1481-1486

[15] - Tuncer Baykas, Leonardo Goratti, Tinku Rasheed, Shuzo Kato – "On the Spectrum Efficiency of Mesh and Star Topology Wide Area Wireless Sensor Networks" in IEEE 25th International Symposium on Personal, Indoor and Mobile Radio Communications 2015, Published 2015, p-p 1819-1823

[16] - IEEE – "Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)" in IEEE Standard for Information Technology

[17] - Cdt21 – "Digital modulation: Phase Shift Keying (PSK)" – [Online] – Avaialble: http://www.cdt21.com/resources/Modulation/modulation_PSK.asp#p5

[18] – Yao-Hong Liu, Cheng-Lung Li, Tsung-Hsien Lin – "A 200-pj/b MUX-Based RF Transmitter for Implantable Multichannel Neural Recording" in IEEE Transactions on Microwave Theory and Techniques, Vol.57, NO. 10, Published Oct. 2009, p-p 2533-2541

[19] - Rafidah Ahmad, Othman Sidek, Wan Mohd Hafizi Wan Hassin, Shukri Korakkottil Kuhni Mohd – "Implementation of IEEE 802.15.4-Based OQPSK Pulse-Shaping Block on FPGA" in International Conference on Computer Applications and Industrial Electronics (ICCAIE 2011), p.459

[20] – Mansour Naslcheraghi – "Digital Modulation Schemes – O-QPSK" – Available online at: https://www.slideshare.net/mansournch/oqpsk

[21] - Naagesh S. Bhat, "Design and Implementation of IEEE 802.15.4 Mac Protocol on FPGA", Innovative Conference on Embedded Systems, Mobile Communication and Computing, 2011.

[22] - National Instruments – "LabVIEW System Design Software" – [Online] - Available: https://www.ni.com/labview/

[23] - Rahul Dhar, Gesley George, Amit Malani Peter Steenkiste – "Supporting Integrated MAC and PHY Software Development for the USRP SDR" in Networking Technologies for Software Defined Radio Networks, Published 2006, p-p. 68-77

[24] – National Instruments – "USRP-2922" – [Online] – Available: http://www.ni.com/en-gb/support/model.usrp-2922.html

[25] - National Instruments – "What is NI USRP Hardware?" – [Online] - Available: https://www.ni.com/white-paper/12985/en/

[26] – Rob Maunder – ELEC 3204 – "Wireless and Optical Communications Background Information on Amplitude Modulation" p-p 12 – [Online] – Available: https://secure.ecs.soton.ac.uk/notes/elec3204/rm/QAM_Notes.pdf

[27] – Digi International – XBEE S1 80.15.4 RF modules – [Online]- Available: https://www.digi.com/pdf/ds_xbeemultipointmodules.pdf

[28] – Wikipedia – [Online] – Available: https://en.wikipedia.org/wiki/XBee#/media/File:Pair_of_XBee_Series_2s_with_Whip_Antennas.jpg

[29] – Chong You – "Implementation of transceiver system under IEEE 802.15.4 based on LabVIEW with USRP", University of Southampton

[30] – Digi International – "XCTU" – [Online] – Available: https://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu

[31] – Shudong Fang, Stevan Berber, Akshya Swain, Saeed Ur Rehman – "A study on DSSS transceivers using O-QPSK modulation by IEEE 802.15.4 in AWGN and flat Rayleigh fading channels" in TENCON 2010-2010 IEEE Region 10 Conference, 2010

[32] – Felix Wunsch, Holger Jakel, Friedrich K. Jondral – "Performance Evaluation of IEEE 802.15.4 O-QPSK and CSS PHY in the Presence of Interference" in Vehicular Technology Conference (VTC Fall), 2015

[33] – M. Alnuiami, K. Shuaib, I. Jawhar – "Performance Evaluation of IEEE 802.15.4 Physical Layer Using MatLab/Simulink" in Innovations in Information Technology, 2006