 doi:10.20944/preprints202203.0114.v1

Article

Low Power EEG Data Encoding For Brain Neurostimulation Implants

Aikaterini Fragkou [1], Athanasios Kakarountas [2] and Vasileios Kokkinos [3]

[1] Department of Computer Science and Biomedical Informatics, University of Thessaly, Lamia, Greece ; aifragkou@uth.gr
[2] Department of Computer Science and Biomedical Informatics, University of Thessaly, Lamia, Greece ; kakarountas@uth.gr
[3] Department of Neurosurgery, Massachusetts General Hospital, Harvard Medical School, MA, USA; vasileios.kokkinos@mgh.harvard.edu

† This paper is an extended version of the paper presented in 2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM) [1]

Abstract: This research refers to the creation of a method of medical technology such as a device that can collect, encode, temporarily store, and transfer electroencephalographic signals from epileptic patients for study and evaluation. For applying the specific apparatus related to neurostimulation technologies such as Deep Brain Stimulation (DBS) and Responsive Neurostimulation (RNS) the approaches followed concern three basic levels. The first level is concerned with initial modelling and creating the template for the following two stages. Further on, the second level is based on developing code for programming integrated circuits and simulation with the appropriate software. The third and final stage is the most substantial, as the transmitter's construction is attempted at the evaluation level. In particular, more than one software and device are involved in this phase for the most realistic performance of the desired result. However, it is worth noting that the principal goal of this study is to achieve low consumption rates for the device's complete and smooth operation. As a result, this research aims to evolve this device so that it can send data wirelessly and simultaneously provide energy efficiency.

Keywords: epilepsy; Responsive Neurostimulation; wireless transmission; BLE; Delta encoding; low power

1. Introduction

Neurostimulation is described as the stimulation of brain activity by transmitting electrical pulses into the area of interest. It immediately targets the nerves and heals a variety of symptoms ranging from mild headaches to more complicated such as seizures [2]. In the case of epilepsy, three basic neurotransmission techniques are available: responsive, deep brain, and pneumogastric nerve stimulation.

Responsive Neurostimulation (RNS) is a medical device that captures electroencephalographic (EEG) data, diagnoses seizures, and stimulates the region that generates them. It was made by Neuropace [7] and approved by the American Food and Drug Administration in 2013 (U.S. Food and Drug Administration -FDA). In brief, it is a strategy of dealing with seizures as they occur, and it is suited to adult patients, who benefit from its methodology in addition to the pharmaceutical immunity they encounter.

The original basic implant consists of four electrodes put on the epileptogenic area. The electrodes are connected to the stimulator, which -when a seizure is detected- sends electric pulses to the nerves [4]. Additionally, the stimulator saves the signals and analyses them to identify seizures. The stimulator is implanted in the back and sides of the skull [5] and is carefully configured to respond to the needs of the implanted patient, such as the type of epilepsy and how it manifests itself. When the stimulus detects irregular brain electrical activity, it emits brief electrical pulses into the epileptogenic area for a few milliseconds [3]. Furthermore, each patient is provided with a laptop to which they submit their daily measurements through one telemetry device once a day, as well as a magnet

 doi:10.20944/preprints202203.0114.v1

with which they activate and deactivate the recording system when they feel they are going to have a seizure. Patients are asked to submit weekly data obtained on the computer to an online database where the doctor may analyse the signals [6] [7]. Depending on the doctor's remarks, the entire system can be reprogrammed until patients' everyday lives improve noticeably [6].

According to the American Association of Neurosurgeons, Deep Brain Stimulation (DBS) is a form of neurostimulation device used to treat seizures in the brain locations where they occur [8]. The stimulator, like in RNS systems, is a programmable device –implanted on the upper part of the chest– that delivers electrical pulses to regulate abnormal brain activity in an open loop mode. DBS, on the other hand, is employed when the location from which epileptic seizures develop is not as confined [9]. Although it is currently used to treat a wide range of focal seizures, it was first approved for temporal lobe stimulation [3].

Last but not least Vagus Nerve Stimulation (VNS) which comprises two sections, is given to individuals above the age of four in addition to their medicine. The first component is the stimulator, which, like DBS systems, is programmable and is inserted in the patient's subclavian region. If the patient experiences an upward seizure, they can utilize the device's provided magnet to trigger the suppression mechanism. The second component is the electrode, which wraps around the pneumogastric nerve at one end and is attached to the stimulator through a wire at the other. As a result, when a seizure occurs, the system is set to emit electrical pulses that suppress abnormal brain activity [3].

A main issue with neurostimulators is that they are powered by a battery that must be replaced when it runs out. This replacement will necessitate a recurrent surgical procedure on the patient, incurring significant costs for both the patient and the healthcare system. Furthermore, in order to send data from an RNS system to the patient's computer, the patient must scan the region of the implant once a day and wait for the wired transmission to complete [7]. As a result, low energy consumption, wireless transmission, and autonomy of these implants' devices are challenges that need to be investigated further.

As a result of the aforementioned requirements, we present the creation of a system for encoding and storing brain signals from brain neurostimulation devices in this paper. The captured data will subsequently be sent to the patient's PC or mobile device using a low-power communication protocol. Our system's designing parameter was chosen to be energy consumption because when low consumption is achieved, the time period before the neurostimulator battery has to be replaced may be substantially prolonged. The intention of this research is to transmit EEG data wirelessly and at a low power level.

2. Materials and Methods

2.1. Modules and design

The data obtained from the implanted device is closely tied to the data that will eventually be saved and delivered to the computer. The structure of these pieces must be preserved qualitatively and quantitatively for transmission. The information sent must be adjusted to fit particular data packets with low energy costs during transmission. Prior to packing, the information is compressed and encoded and then delivered over the protocol that links and communicates the different devices to achieve this energy reduction target. At this stage, the methods for structuring transmitting packages and the algorithms used to compress biosignals with minimal energy consumption will be examined.

2.1.1. EEG data

When the intracranial device is attached to the device described in this study, it receives data from the neurostimulator and sends it wirelessly to the central unit. We used PhysioNet, a library of physiological data that contains EEG recordings from 23 epileptic children admitted to Boston Children's Hospital, to duplicate this transmission strategy with actual EEG data. Each recording includes 9-42 one-hour sessions [10][11].

 doi:10.20944/preprints202203.0114.v1

For modelling in MATLAB, there was a random selection of a patient aged 11 years old as a first sample by whom its third one-hour measurement was employed. For 90 seconds of EEG data, the sampling rate was 256 Hz, resulting in a total of 23,040 samples [10][11][12]. Leads were chosen randomly, without any correlation of the areas where they are placed to the seizures arising, but the ability to be compressed for the best feasible way to transfer them.

2.1.2. Encoding algorithms

In 1949, a data encoding technique was presented based on the scientific work of Shannon and Fano. It rests on the idea that characters appearing in information streams are encoded based on how frequently they appear. The information stream is encoded into individual symbols, compressing it by computing the frequency or probability of their appearing in the stream for each of them. When these probabilities are determined, a list of all the symbols, along with their frequencies, is created.

The following step is the classification of the symbols in the list recorded, from the most likely to the least likely character, and dividing them into two categories. The most likely characters are in one group, and the least likely characters are in the other group. They are distributed so that the sum of the probabilities of the symbols in one group is not far from the corresponding sum of the probabilities in the other group. Furthermore, the group with the most likely characters is assigned bit 0, while the other is assigned bit 1. This process is continued until each group is divided into subgroups using the same probability criteria, resulting in a tree with a left child with a value of zero and a right child with a value of one, until there is only one symbol left in each subgroup [13][14].

A few years later, Huffman (1952) proposed a coding algorithm that, at first glance, appears to be similar to Shannon-Fano. This belief stems from the fact that both techniques depend on maintaining low entropy, which means that the more frequently a symbol appears in the coded message, the fewer digits are required to encode it. The coded information in each of its characters corresponds to a weight that reflects its frequency of occurrence. All symbols and their weights are set out to list the most likely to appear at one end and the least likely to occur at the other. Consequently, this technique entails creating a tree with a direction from the leaves to the root, repeated until the node containing the sum of all weights is calculated. The two lowest probabilities are adjusted together, and their sum is placed on a node, with children corresponding to the cumulative probabilities. The following two slighter probabilities are added up, and the new node and its children are added to the tree in construction [15].

In dictionaries-based algorithms, a new technique was introduced in 1980. Lempel-Ziv-Welch is a lossless compression method which uses string tables with corresponding encodings . At the beginning of each coding, a table is created that contains all of the symbols along with their codes and positions. As a result, each time a new string arrives, its appearance in the table is checked, which means that each sequence, along with the next item, is checked for its existence as a whole. Repeated sequences are detected, or new ones are added in this manner until the compressed data is complete [28].

Among the lossless encoding techniques that create dictionaries are methods based on sliding windows. These methods enable the return to previously encoded data and relate it to the data to be compressed. Based on this logic, the algorithms Lempel Ziv 77 (LZ77) and Lempel Ziv 78 (LZ78) produce codes with the relevant correlations as an output [17][18].

EEG data compression can be achieved using the Compressive sensing (CS) method in applications that record one-dimensional sparse signals. In this case, each signal is treated as a one-dimensional matrix of length N. When multiplied by a two-dimensional matrix MxN ($M << N$), it produces a one-dimensional signal Mx1. The generated signal appears to have less information, but this does not imply that essential data that would prevent the original signal from being retrieved has been lost [19][20]. Proper selection of the compressor matrix can improve energy consumption and reduce computational waste based on the design parameters of the current research system. The proposed matrices are

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2022 doi:10.20944/preprints202203.0114.v1

sparse and binary, with random, independent, and uniformly distributed coefficients that isolate different values of the original signal when multiplied by it [21].

Arithmetic Encoding is also a lossless compression method that generates codes by calculating the probability of information symbols for coding an entire string. These codes derive from the interval in which the expression is classified, restricted to [0,1] [29].

The SZ algorithm is a lossy compression technique that investigates predicting data based on neighbouring values. Those values can be predicted due to the adaptation to mathematical curves. The data is transformed into a one-dimensional linearisation and classified as predictable or unpredictable before coding. The compression process for unpredictable data involves the simple application of IEEE-754 coding and the additional study of the essential bits in the mantissa field for which repetitive patterns are encountered, ignoring the insignificant parts [22][23].

A brain signal can be normalised somehow, with the ultimate goal of limiting its values to a specific interval [a, b]. Many electrodes (multichannel) contribute to the reception of such a signal. The received channels can be organised in a two-dimensional array, with each line containing the samples taken from each channel based on the sampling frequency. In practice, a smoothing process occurs in the data array, which yields results with an integer and a decimal part (MCEEG - Multichannel EEG). Effectively, a smoothing process also appears in the data table from which results are obtained with an integer and a decimal part [24]. The Run Length Encoding (RLE) algorithm, which uses the information from the repeating symbols, compresses the integer part [27].

Algorithms that transform signals into different fields, such as the time and frequency domains, are used in various compression techniques. This transition from one state to another is accomplished using the Discrete Cosine Transform, which reduces information by expressing each signal's respective coefficients [25][26].

When a data sequence is available for encoding, calculating the differences between successive values and their coding is one method of limiting compressed information. In other words, the final result is a series of numbers that represent the interval between one value and the next. When the values are close together, there is a better chance that the calculated differences will be close to zero, requiring fewer bits for compression. This method can maintain data quality when compressed, mainly when applied to brain signals (lossless). Furthermore, its implementation does not necessitate complex and costly energy calculations because finding differences requires only one operation, subtraction. It is worth noting that it allows recording signal changes over time rather than the absolute width of the measurements for each period, which saves storage space and energy for subsequent transmission [27] [30]. Because the system that emerges from this study must temporarily store data and then wirelessly forward it to another network node, the encoding must reduce computing, storage, and transfer consumption. For all of the reasons stated above, Delta encoding is deemed most appropriate for this implementation.

2.1.3. System's memory

After compression, the data must be stored somewhere and retrieved before being sent to another device. The original RNS system wand does not permanently store the data but instead mediates its transfer to the central computer system. As a result, to implement a single storage unit that will allow data retrieval from previously stored data, buffers were used to hold the signals until they are transmitted to the BLE unit, from where they will be wirelessly transmitted. The buffers were implemented at the modelling and integrated circuit programming levels. To model in MATLAB how data is stored and read from memory, a First-in-First-Out (FIFO) buffer, which is a buffer based on the principle that whatever data is placed first is the first to leave, was simulated. A FIFO buffer is described by two pointers set at the beginning of the memory and indicates where the data is placed or removed. When pointers reach the end of the structure, they move to the beginning of the buffer to continue writing or reading, avoiding the overwriting of data that have not been transmitted and read yet [31].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2022 doi:10.20944/preprints202203.0114.v1

2.1.4. Bluetooth Low Energy - BLE

By implanting the stimulator as planned and activating its function, the following step gathers the data collected by the integrated electrodes. The rod is employed to initiate the transmission of signals via the RF communication protocol. The bar is then linked via a USB connection to a portable computer unit, where the data is kept for use in monitoring and assessing the patient. Many gadgets, including mobile phones, WIFI devices, satellite communications systems, and Bluetooth, employ electromagnetic radiation or electromagnetic radio waves to transmit and receive data wirelessly. It is separated into frequency bands (low, medium, and high) to make tuning the devices more accessible.

According to the extant literature, WIFI, Bluetooth, Bluetooth Low Energy, and Zigbee are the four primary protocols utilized for the communication of medical implant devices [32]. Zigbee is concerned with the formation of device networks and is used when several implants are in the same body, but in this study, only one implant is accessible [32][34]. Whereas Bluetooth Low Energy is the chosen technology for low-consumption applications. In particular, it assures up to ten times less consumption than WIFI [33].

Bluetooth Low Energy (BLE) is a communication protocol similar to Bluetooth Classic that operates at 2.4 GHz. This type of protocol can be found in various devices, including intelligent home automation systems, medical devices, fitness and navigation systems. These systems have one thing in common: they all transfer small amounts of data at slow speeds. Furthermore, they are divided into two parts. The first is concerned with a collection of sensors and other information recording devices (peripheral). The second is concerned with the computer systems (central) that process them (laptops, smartphones, PCs, and more). The peripheral part enters the 'sleep' mode more frequently to consume less energy.

To understand why BLE is used, the features that make it unique and necessary will be considered. Initially, low power consumption is a sufficient reason to use it, as there is a need for a wide range of devices with long battery life and durability. Due to the low cost of the modules, construction and installation are also kept to a minimum. Finally, there is a variety of open-access information on this technology that many devices have adopted. Therefore, in terms of consumption, cost, access to information, and compatibility, Bluetooth Low Energy becomes a competitive medium for providing small amounts of data transmission devices, where high transfer rates are not required. It should be mentioned that the assessment of the impact of wireless technology and radiation on patients' health is not the focus of this work [35].

2.1.5. System's design

This study aims to create a system for coding and transferring electroencephalographic signals recorded from the brain of epileptic patients while retaining the signal quality and conserving energy. As a result, meticulous design is necessary before implementation to avoid omissions, understand its functioning, and completely describe the device's constituent subsystems. First and foremost, the system's expected input must be determined. As shown in the diagram (Figure 1), this input consists of four channels of electroencephalographic signals; the amount of data corresponds to 90-second measurements, which in the already implemented neurostimulator result from the combination of observations 30 seconds before and 60 seconds after stimulation, respectively [12]. The data is then sent through an Analog-to-Digital converter, which is encoded and sent to the microcontroller unit (MCU).

In the first sub unit of the MCU the data is coded using the Delta technique, which calculates the differences between the samples. The compressed data is subsequently stored in system memory, where it is retrieved and transmitted whenever possible or requested. The communication is wireless and uses the BLE protocol, which divides the information into data packets stored and analysed on a host computer. The Control Unit supervises all of these modules, which synchronises them; when a connection signal from the BLE

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2022 doi:10.20944/preprints202203.0114.v1

module alerts the Control Unit that a connection has been established, the system functions (Figure 1).

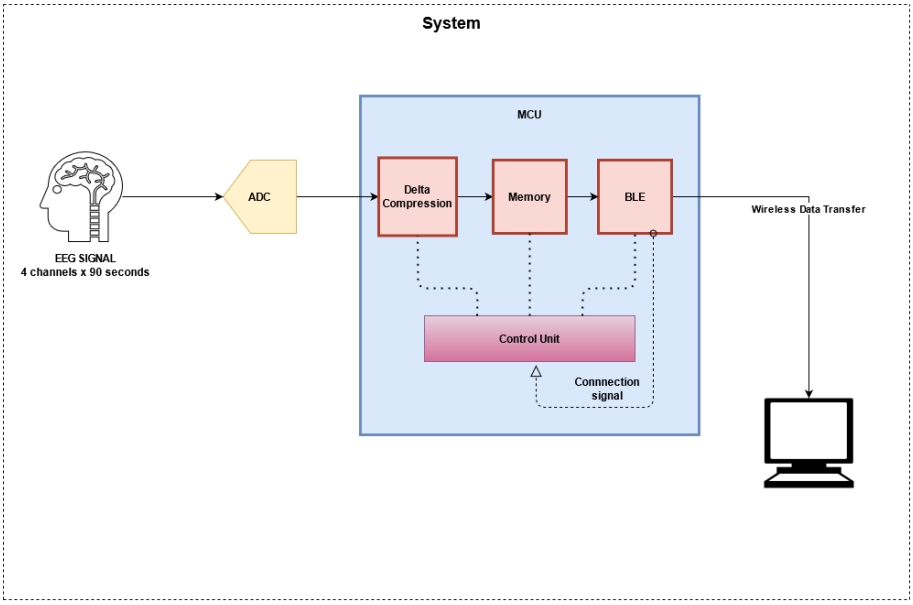

Figure 1. The modules of the system and the total design.

2.2. Modules' implementation

This research's implementation will take place on three levels. The first level concerns Matlab modelling for the preliminary evaluation of the proposed compression and how the signals will be structured. The second level of implementation involves the generation of integrated circuit code for the most efficient data storage and transmission, while the last and third effort involves the programming and control of the system's functioning with Arduino.

2.2.1. Modeling in MATLAB

In the initial simulation of the system for storing and transmitting brain signals, a function named delta_en was constructed in MATLAB (MATLAB R2021a) [36]. This function takes data as input and calculates the differences between successive values of the electroencephalographic signal in a loop of repetition. It then returns a one-dimensional compressed data array, transformed into the binary system to compute the necessary bits. The conversion is performed using the function decimal-to-binary (dec_2_bin), which determines the number of bits required to represent the integer and decimal parts and uses sequential divisions to calculate the binary form of the differences. The system determines the selection of all bits under consideration, and an equal distribution would be preferable (16 bits for integer and 16 bits for decimal).

A part of the modelling is shown in Listing 1. After reading the data file, four channels are chosen and transformed to binary form. The conversion occurs before encoding to quantify the transitions. The entire algorithm comprises successive phases such as signal reading and channel selection. These channels are then transformed to their binary form (each value is represented by 32 bits) to calculate the initial 0-to-0 (zeros_2_zeros), 1-to-0 (ones_2_zeros), 1-to-1 (ones_2_ones), and 0-to-1 (zeros_2_ones) transitions. Finally, the data is coded (delta_en) , and the new 0-to-0 (zeros_2_zeros) , 1-to-0 (ones_2_zeros), 1-to-1 (ones_2_ones), and 0-to-1 (zeros_2_ones) transition measurements are computed to compare them to the sets from the signal's decoded state.

Listing 1: MATLAB modeling of decimal to binary conversion and calculation of the total number of 0-to-1 transitions

```
clear all;
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2022 doi:10.20944/preprints202203.0114.v1


```
close all;
clc;
%% reading the eeg from edf file
file='D:\chb01_03.edf';
[record,hdr]=readEDF(file);

%% selection of 4 channels
%%with duration of 90sec(256x90=23,040 samples)
channel_1= record(21,23040:46080);
channel_2= record(22,23040:46080);
channel_3= record(23,23040:46080);
channel_4= record(20,23040:46080);
%% decimal to binary
bin_ch_1=[];
bin_ch_2=[];
bin_ch_3=[];
bin_ch_4=[];
for cols=1:length(channel_1)
out=[];
out= dec_2_bin(channel_1(cols));
bin_ch_1= [bin_ch_1; out'];
out= dec_2_bin(channel_2(cols));
bin_ch_2= [bin_ch_2; out'];
out= dec_2_bin(channel_3(cols));
bin_ch_3= [bin_ch_3; out'];
out= dec_2_bin(channel_4(cols));
bin_ch_4= [bin_ch_4; out'];
end
bin_ch_1=bin_ch_1';
bin_ch_2=bin_ch_2';
bin_ch_3=bin_ch_3';
bin_ch_4=bin_ch_4';
%% count 0-1 transitions
count1_z_2_o=zeros_2_ones(bin_ch_1)
count2_z_2_o=zeros_2_ones(bin_ch_2)
count3_z_2_o=zeros_2_ones(bin_ch_3)
count4_z_2_o=zeros_2_ones(bin_ch_4)
```

2.2.2. Memory in VHDL

The second step of implementation comprises the methods for programming integrated circuits and, in particular, the arrangement of the system's memory. A FIFO buffer was constructed in VHDL for this purpose, utilising the Quartus platform and ModelSim to write and simulate the code [37]. Some principles must be followed while designing a FIFO buffer to fit the transmitting device. Initially, the FIFO buffer entity was built, which contains the memory capacity and the size of each location in the memory, the clock, the reset variable, the variables to permit writing and reading (en_w, en_r), the data array, and the buffer (full, empty) control variables. Then the RAM implementation signals, the record counter (count), the pointers of the writing and reading locations (input, output), and the auxiliary variables for signalling the fullness and emptiness of the buffer (full_i, empty_i) were specified.

First of all, in the architecture, the reset control is conducted to initialise the counter and pointer values. These pointers increase every time data are read or written, and they are controlled to avoid overwriting. It and could be accomplished by using the signal named "full" which, when set to 1, prevents the addition of new values. When a new

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2022 doi:10.20944/preprints202203.0114.v1

element is introduced to the buffer, the counting signal is incremented by one. If an element is read, the counter is decremented by one, and new memory space is made available to write a new value. Furthermore, when the reading or writing pointer reaches the end of the buffer, it returns to the start point, creating a circular motion of the pointers. Listing 2 has the above requirements and is a part of the VHDL code.

Listing 2: VHDL code of the FIFO buffer

```
FIFO_IMPL: process (clk) is
begin
 if clk'event and clk='1'and clk'last_value='0'
 then
   if rst = '1' then
    count <= 0;
    input   <= 0;
    output  <= 0;
    data_read <= (others => '0');
   else

--reduction of count when a value is being read
   if (en_r = '1') then
     count <= count - 1;
   end if;
--writting process
   if (en_w = '1' and full_i = '0') then
     if input= RAM_L-1 then
       input<= 0;
     else
           input <= (input + 1);
     end if;
     ram(input) <= data;
     count <= count + 1;
     end if;

--reading pointer update
   if (en_r = '1' and empty_i = '0') then
     if output = RAM_L-1 then
       output <= 0;
     else
       data_read <= ram(output);
       output <= output + 1;

     end if;
    end if;
   end if;
 end if;
end process FIFO_IMPL;
full_i  <= '1' when count= RAM_L else '0';
empty_i <= '1' when count = 0       else '0';
full  <= full_i;
empty <= empty_i;
```

2.2.3. Data transmission with BLE

An Arduino platform was programmed to gather data from a computer, encode it, store it, and deliver it via the BLE communication protocol to demonstrate the system's

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2022 doi:10.20944/preprints202203.0114.v1

functioning. The Arduino model used was the Arduino Nano 33 BLE [38] because it already had the BLE module. To enable Arduino to send data, each value must be divided from 32 bits of information in the form of four bytes. When new data arrives on the computer's serial connection, the Arduino reads it in groups of four bytes.

After transferring a data packet, the Arduino gets a signal indicating that information is available (Serial. available ()> 0). As a result, it begins receiving this data (Serial. read ()) [83], saving it in a struct that takes quadruple Bytes, which it joins (fourByteMerge) to produce the 32-bit long integer that represents the same bits as the EEG signal's equivalent floating-point element. The integers are then saved in a FIFO buffer (buff.push ()), and when it is full (buff.size () equals a specific amount), data extraction (buff.pop ()) is enabled. The RingBuf.h library contains functions for this type of FIFO buffer. The large numbers that depart the buffer are broken down into 4-byte clusters and wirelessly sent via the BLE (valueLevelChar.write value ()) protocol. Simultaneously, a link has been formed between the Arduino platform and another device (central) that supports the communication protocol. In our case, the central device was a mobile phone with the "nRF Connect" [40] application installed. In relation to what we refer to in this paragraph is included in the section of the code below.

Listing 3: Arduino code for the evaluation of the transmission with BLE protocol

```
void updatevalueLevel() {

 byte b1,b2,b3,b4;
 int num1,num2,num3,num4;
 if (!buff.isFull()){
 for (int i=0;i<64;i++){
  if (Serial.available()) {
   combined.parts.firstByte = Serial.read();
   while(Serial.available()==0){
   }
   combined.parts.secondByte = Serial.read();
   while(Serial.available()==0){
   }
   combined.parts.thirdByte = Serial.read();
   while(Serial.available()==0){
   }

   combined.parts.fourthByte = Serial.read();
   // combine all the bytes
   // to create the long integer
   combined.merged =
   ((unsigned int)
   (combined.parts.firstByte)<<24)|
   (unsigned int)
   (combined.parts.secondByte<<16)|
   (unsigned int)
   (combined.parts.thirdByte<<8)|
   (unsigned int)
   (combined.parts.fourthByte);
   // delta encoding
   cur_val= combined.merged;
   delta = cur_val-prev_val;
   buff.push(delta);
   prev_val=cur_val;
   length_buff++;
  }
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2022 doi:10.20944/preprints202203.0114.v1

}

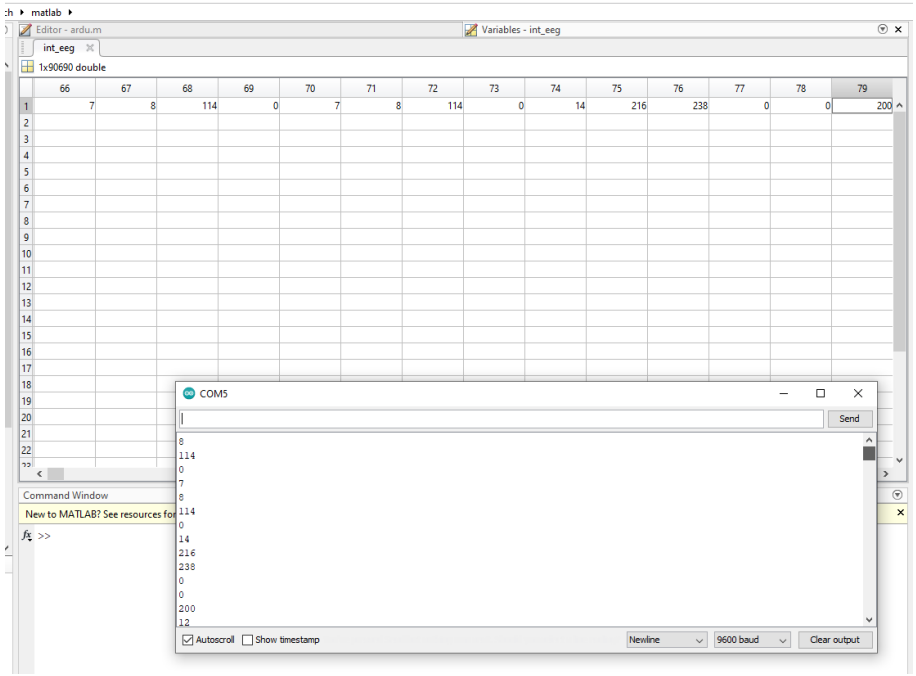

Figure 2. Evaluation of accurate EEG data transmission with Arduino. The values received by Arduino from the computer are the identical values that it began transmitting to the smartphone.

3. Results

The essential criteria for assessing the specific study is the management of the system's functioning in terms of its design parameter, i.e., the low energy consumption. The system takes a proportion of energy to operate, and the primary purpose is to conserve as much as possible so that battery replacement and frequent recharge are unnecessary. The power and how it is consumed in the system are considered for this reason. However, the question of how the evaluation would be conducted emerges. As described in earlier sections, the electroencephalographic signal transmitter was implemented in three fundamental ways: modelling, integrated circuit programming and Arduino programming. Energy consumption may be calculated both at the modelling and data transmission levels using Arduino. In the first one, the energy reduction between coded and non-coded channels is computed using a virtual technique, while in the other, a reasonably simple method is to connect the Arduino to a power bank and monitor the time it takes to transmit a signal before and after processing. No energy evaluation is undertaken in the second level because the VHDL programming level concerns memory structure.

Consumption is calculated as a percentage of energy consumed per unit of time and can take two forms: dynamic consumption and static consumption. The equation for dynamic consumption is:

$$P_D = C_L \times V_{DD}^2 \times F_{clk} \times a \tag{1}$$

, which is related to the proportion of transitions (switching power a) from zero to one, the capacitance (C), the voltage (V_{CC}) and the frequency (F_{clk})) of the circuit. Various parameters may be addressed to obtain the energy reductions necessary to power circuits. The transition factor influences the transition frequency. In general, the lower this rate, the fewer transitions required and less energy spent because the switches 0-to-1 are fewer.

The second type is static power consumption, which occurs while the system is idle. This type of consumption depends on the leakage currents that occur when the device is not operating. It is determined by the following equation:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2022 doi:10.20944/preprints202203.0114.v1

$$P_S = V_{CC}^2 \times V_{CC} \quad (2)$$

, where I_{CC} and V_{CC} correspond to the average current circulating at the circuit and the voltage [39].

All transitions affect static consumption almost as much as dynamic consumption. Voltage and capacitance, for example, are circuit properties, so what can be easily monitored since it depends on the signal are the many forms of transitions. The total number of transitions was multiplied by an estimated weight to evaluate their contribution to consumption. The weight for 0-to-1 transitions was calculated by adding the number of different types of transitions (0-to-1, 1-to-1, 0-to-0, 1-to-0) and the contribution of static consumption. As a result, the weight for the 0-to-1 transitions was five times greater than the weight for the other transitions.

Table 1. Comparison of the transitions before and after Delta encoding.

A0	A1	A2	A3	A4	A5	A6
1	673,220	549,515	1,264,096	1,164,980	7.8409	18.3751
2	669,515	527,270	1,260,845	1,147,182	9.0148	21.246
3	664,420	513,600	1,257,039	1,136,120	9.6194	22.6995
4	667,430	528,945	1,259,597	1,148,392	8.8286	20.749
5	686,040	571,775	1,274,460	1,182,678	7.2016	16.6557
6	684,600	567,005	1,273,248	1,178,986	7.4033	17.1772
7	667,360	535,950	1,259,546	1,154,084	8.373	19.691
8	674,605	561,470	1,265,401	1,174,766	7.1626	16.7706
9	684,020	571,605	1,272,830	1,182,544	7.0933	16.4345
10	685,315	554,900	1,273,815	1,169,322	8.2032	19.0299
11	671,190	542,465	1,262,620	1,159,265	8.1858	8.1858
12	679,765	627,555	1,269,519	1,227,736	3.2912	7.6806
13	674,980	601,435	1,265,526	1,206,553	4.66	10.8959
14	672,665	595,725	1,263,714	1,202,193	4.8683	11.4381
15	673,250	534,130	1,264,391	1,152,470	8.8518	20.6639
16	676,825	636,815	1,267,159	1,235,278	2.5159	5.9114
17	691,235	568,575	1,278,594	1,180,287	7.6887	17.7451
18	685,225	568,765	1,273,732	1,180,522	7.3179	16.9959
19	668,810	513,760	1,260,744	1,136,281	9.8722	23.183
20	661,190	586,285	1,254,508	1,194,316	4.7981	11.3288
21	676,470	584,705	1,266,865	1,193,110	5.8219	13.5653
22	659,300	528,790	1,253,064	1,148,467	8.3473	19.7952
23	673,250	534,130	1,264,391	1,152,470	8.8518	20.6639

In Table 1, the columns correspond to the following:

- A0: The channels being studied
- A1: Transitions from 0 to 1 prior to Delta coding (calculated by multiplying all transitions
- A2: The transitions from 0 to 1 following Delta coding (estimated by multiplying the total transitions by weight)
- A3: Total number of transitions prior to coding ((0-0) + (1-1) + (1-0) + 5 * (0- 1))
- A4: The total number of transitions following coding ((0-0) + (1-1) + (1-0) + 5 * (0-1))
- A5: The percentage of reduction of total transitions
- A6: The percentage of reduction of transitions 0-1

As a result of the implementation, the attempt to assess the consumption of the 23 channels resulted in an overall reduction of transitions from 0 to 1, ranging from 5.9114% to almost 23.2% between non-coded and encoded signals (Figure 5). The reduction was not significant in some channels (channels 12 and 16), but this is not causing concern because these results are dependent on the characteristics of the signal received from the specific leads. The use of Delta coding reduced the number of transitions by up to 10% (Figure 6),

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2022 doi:10.20944/preprints202203.0114.v1

indicating that the specific solution extends the life of the neurostimulation equipment, which is significant because every time the device's battery runs out, surgery is required to replace it.

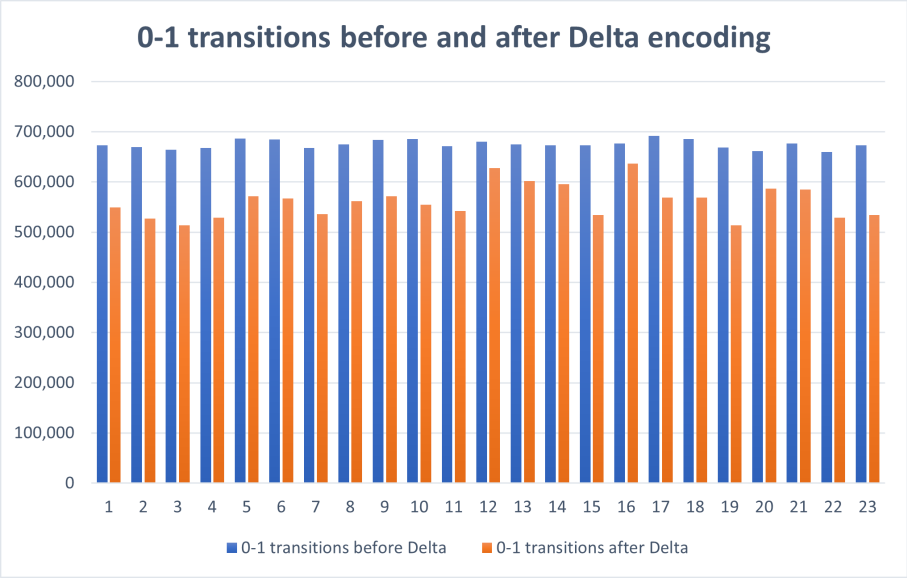

Figure 3. Bar chart of 0-1 transitions before and after Delta encoding for all the leads examined.

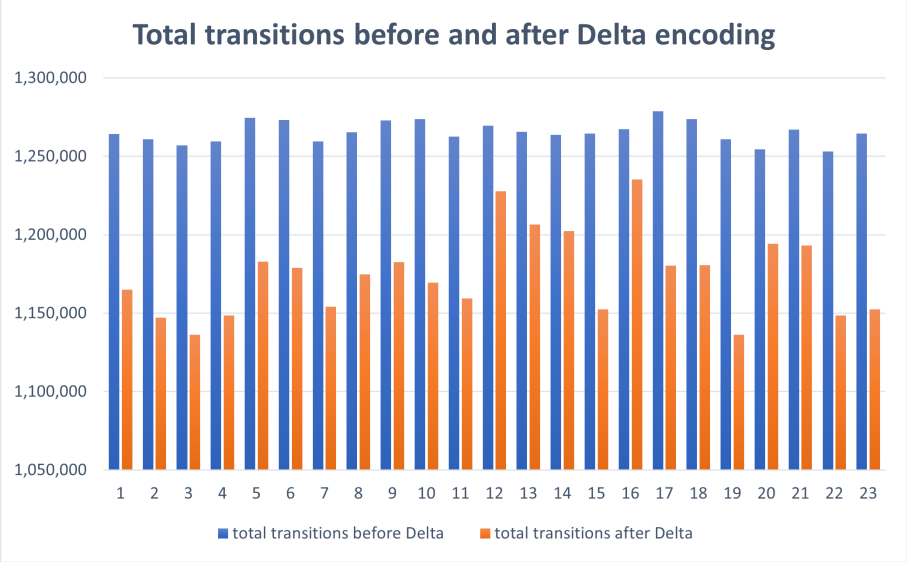

Figure 4. Bar chart of 0-0, 0-1, 1-0 and 1-1 transitions before and after Delta encoding for all the leads examined.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2022 doi:10.20944/preprints202203.0114.v1

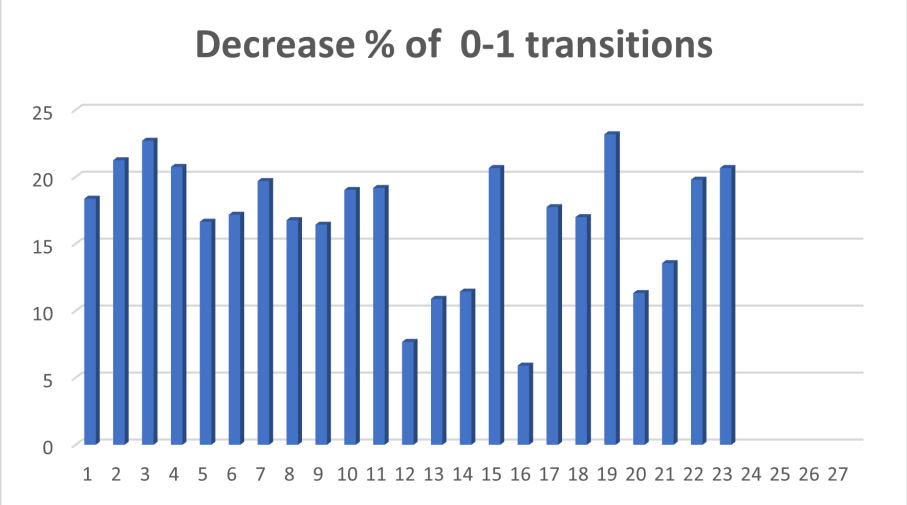

Figure 5. Bar chart of % reduction of 0-1 transitions after the encoding.

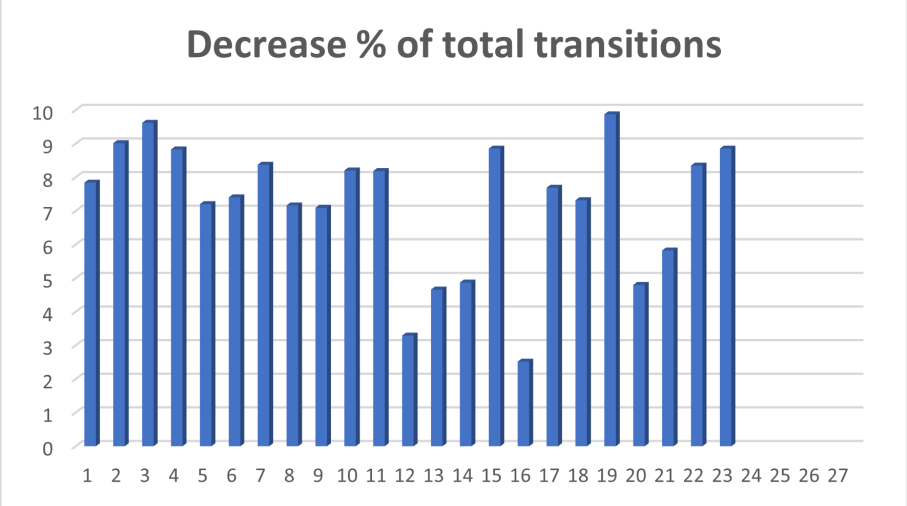

Figure 6. Bar chart of % reduction of 0-0, 0-1, 1-0 and 1-1 transitions after the encoding.

4. Discussion

The encoding wireless data transmission system from implanted brain stimulation devices was created using MATLAB modelling, VHDL coding, and Arduino simulation. This implementation is targeted to analyse the brain signals — captured in patients with epilepsy— which will be elements that the relevant medical team will utilise to investigate and treat seizures.

The subsequent actions that will occur as an outcome of these results provide the scientific team of physicians with the confidence to manage and alter the system's operations to the demands and specific features of the patient. A simulation effort was made, which was rather challenging because there was an issue with the consumption measuring procedure, and any data passed to the Arduino after disconnection was lost. Of course, it is essential to emphasise that if we had completed the development of this gadget and had it activated, the information flow would be seamless, with no data loss or deletion, because the means of supply would not alter. The employment of lossless coding methods, the usage of the BLE protocol, the writing of code for the proper operation of the system, and its assessment are all key factors that draw attention. The data was temporarily saved in the buffer and recovered from it was the same as the values from the original signal, according to the tests done with the Arduino.

 doi:10.20944/preprints202203.0114.v1

Furthermore, in the MATLAB simulation, the decrease of transitions owing to coding resulted in a significant 23.2% reduction of the signal transmission, with the system's autonomy increasing as a result. The desired results were not only achieved, but they also generated circumstances and opportunities for future growth. These development perspectives are related to the quality of its product materials time, the study of further upgraded coding methods, real-time consumption measurement, the evolution of medical science, and the connection and complete presentation of the interior and exterior seizures measurement-stimulation system, and data transmission.

Author Contributions: Conceptualization, V.K. and A.K.; methodology, A.K.; software, A.F.; validation, A.F. and A.K.; formal analysis, A.F.; investigation, A.F.; resources, V.K.; data curation, A.F.; writing—original draft preparation, AF.; writing—review and editing, AK.; visualization, A.F.; supervision, A.K. and V.K. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: The EEG data used in this research are available at https://physionet.org/content/chbmit/1.0.0/.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RNS	Responsive Neurostimulation
EEG	Electroencephalographic
FDA	Food and Drug Administration
DBS	Deep Brain Stimulation
VNS	Vagus Nerve Stimulation
LZ77	Lempel Ziv 77
LZ78	Lempel Ziv 78
CS	Compressive Sensing
MCEEG	Multichannel EEG
RLE	Run Length Encoding
FIFO	First-in-First-out
BLE	Bluetooth Low Energy

References

1. Fragkou, A. A.; Kakarountas, A. P.; Kokkinos, V. Low-Power Electroencephalographic Data Encoding System for Implantable Brain Stimulation Systems. In 2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM) (September 2021); 1–5; IEEE.
2. ins Internation Neuromodulation Society . Available online: https://www.neuromodulation.com/about-neuromodulation (accessed on 18 February 2022).
3. EPILEPSY FOUNDATION AND EPILEPSY TOGETHER . Available online: https: https://www.epilepsy.com (accessed on 18 February 2022).
4. Kokkinos, V.; Urban, A.; Sisterson, N. D.; Li, N.; Corson, D.; Richardson, R. M. Responsive neurostimulation of the thalamus improves seizure control in idiopathic generalized epilepsy: a case report. *Neurosurgery* **2020**, *87(5)*, E578–E583.
5. UCSF HEALTH . Available online: https://www.ucsfhealth.org/treatments/responsive-neurostimulation (accessed on 18 February 2022).
6. University of Pittsburgh . Available online: https://www.neurosurgery.pitt.edu/centers/epilepsy/responsive-neurostimulation (accessed on 18 February 2022).
7. NEUROPACE . Available online: https://www.neuropace.com/ (accessed on 18 February 2022).
8. American Association of Neurological Surgeons . Available online: https://www.aans.org/en/Patients/Neurosurgical-Conditions-and-Treatments/Epilepsy (accessed on 18 February 2022).
9. Cleveland Clinic . Available online: https://my.clevelandclinic.org (accessed on 18 February 2022).
10. Shoeb, A. H. Application of machine learning to epileptic seizure onset detection and treatment (Doctoral dissertation, Massachusetts Institute of Technology).
11. Goldberger, A. L.; Amaral, L. A.; Glass, L.; Hausdorff, J. M.; Ivanov, P. C.; Mark, R. G.; ... & Stanley, H. E. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *circulation* **2000**, *101(23)*, e215–e220.

12. Sisterson, N. D.; Wozny, T. A.; Kokkinos, V.; Constantino, A.; Richardson, R. M. Closed-loop brain stimulation for drug-resistant epilepsy: towards an evidence-based approach to personalized medicine. *Neurotherapeutics* **2019**, *16(1)*, 119–127.
13. Shannon, C. E.; Weaver, W. The mathematical theory of information. *Urbana: University of Illinois Press* **1949**, *97*.
14. Fano, R. M. The transmission of information. *Cambridge, Mass, USA: Massachusetts Institute of Technology, Research Laboratory of Electronics* **1949**.
15. Huffman, D. A. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE* **1952**, *40(9)*, 1098–1101.
16. Rigler, S.; Bishop, W.; Kennings, A. FPGA-based lossless data compression using Huffman and LZ77 algorithms. In 2007 Canadian conference on electrical and computer engineering (April 2007); 1235–1238; IEEE.
17. Freschi, V.; Bogliolo, A. A faster algorithm for the computation of string convolutions using LZ78 parsing. *Information processing letters* **2010**, *110(14-15)*, 609–613.
18. Yan-li, Z.; Xiao-ping, F.; Shao-qiang, L.; & Zhe-yuan, X. Improved LZW algorithm of lossless data compression for WSN. In 2010 3rd International Conference on Computer Science and Information Technology (July 2010); 523–527; IEEE.
19. Zhang, Z.; Jung, T. P.; Makeig, S.; Rao, B. D. Compressed sensing of EEG for wireless telemonitoring with low energy consumption and inexpensive hardware. *IEEE Transactions on Biomedical Engineering* **2012**, *60(1)*, 221–224.
20. Gurve, D.; Delisle-Rodriguez, D.; Bastos-Filho, T.; Krishnan, S. Trends in compressive sensing for EEG signal processing applications. *Sensors* **2020**, *20(13)*, 3703.
21. Luo, K.; Cai, Z.; Du, K.; Zou, F.; Zhang, X.; Li, J. A digital compressed sensing-based energy-efficient single-spot bluetooth ecg node. *Journal of healthcare engineering* **2018**.
22. Tao, D.; Di, S.; Guo, H.; Chen, Z.; Cappello, F. Z-checker: A framework for assessing lossy compression of scientific data. *The International Journal of High Performance Computing Applications* **2019**, *33(2)*, 285–303.
23. Di, S.; Cappello, F. Optimization of error-bounded lossy compression for hard-to-compress HPC data. *IEEE transactions on parallel and distributed systems* **2017**, *29(1)*, 129–143.
24. Titus, G.; Sudhakar, M. S. A simple but efficient EEG data compression algorithm for neuromorphic applications. *IETE Journal of Research* **2020**, *66(3)*, 303–314.
25. Karimu, R. Y.; Azadi, S. Lossless EEG compression using the DCT and the Huffman coding. **2016**.
26. Yu, B.; Yang, L.; Chong, C. C. ECG monitoring over bluetooth: data compression and transmission. In 2010 IEEE Wireless Communication and Networking Conference (April 2010); 1–5; IEEE.
27. Smith, S. Digital signal processing: a practical guide for engineers and scientists. *Elsevier* **2013**.
28. Sharma, S.; Chopra, A.. he Study: LZW Compression on SEP Protocol.
29. Hussain, S. R.; Wang, C.; Sultana, S.; Bertino, E. Secure data provenance compression using arithmetic coding in wireless sensor networks. In 2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC) (December 2014); 1–10; IEEE.
30. Kim, S.; Kim, J.; Chun, H. W. Wave2vec: Vectorizing electroencephalography bio-signal for prediction of brain disease. *International journal of environmental research and public health* **2018**, *15(8)*, 1750.
31. VHDL whiz . Available online: https://vhdlwhiz.com/ring-buffer-fifo/ (accessed on 18 February 2022).
32. Nelson, B. D.; Karipott, S. S.; Wang, Y.; Ong, K. G. Wireless technologies for implantable devices. *Sensors* **2020**, *20(16)*, 4604.
33. CABOT . Available online: https://www.cabotsolutions.com/ble-vs-wi-fi-which-is-better-for-iot-product-development (accessed on 18 February 2022).
34. IoT Lab . Available online: https://iotlab.tertiumcloud.com/2020/08/19/classic-bluetooth-vs-bluetooth-low-energy-ble/ (accessed on 18 February 2022).
35. Bluetooth . Available online: https://www.bluetooth.com/ (accessed on 18 February 2022).
36. MATLAB. Available online: https://www.mathworks.com/products/matlab.html (accessed on 18 February 2022).
37. intel. Available online: https://www.intel.com/ (accessed on 18 February 2022).
38. Arduino. Available online: https://www.arduino.cc/ (accessed on 18 February 2022).
39. EETimes. Available online: https://www.eetimes.com/power-dissipation-in-portables-design-considerations-using-low-power-cmos-ics/ (accessed on 18 February 2022).
40. Nordic Semiconductor. Available online: https://www.nordicsemi.com/ (accessed on 18 February 2022).