COMMUNICATION SYSTEMS

# LAB 1
# ASSIGNMENT 1

Hend Diaa Eldin 4805

Sara Anwar Mostafa 4923

Athar Ashraf Khorshid 4563

Zeyad Mahmoud Fathy 4988

Mohamed Khaled 4783

# Problem 1: Report

## Introduction

Bluetooth is a wireless technology for communication over distances of up to 10m Offering reasonably fast data transfer rates of around 1 Mb/s, principally between Battery-powered devices. Bluetooth's primary intent is to support the creation of adhoc Personal area networks (PANs) for small data transfers (or voice communication) Between devices such as Laptop, PDA, mobile phones, PC's, printers, digital cameras and video game consoles over a secure globally unlicensed short-range radio frequency.

Bluetooth wireless technology is an open specification for a low-cost, low-power, short-range radio technology for ad hoc wireless communication of voice and data anywhere in the world. An open specification means that the specification is publicly available and royalty free. Bluetooth wireless technology works anywhere in the world because it operates at 2.4 GHz in the globally available, license free, industrial, scientific, and medical (ISM) band.

Bluetooth wireless technology was originally developed as a cable replacement technology for connecting devices. The specification defines the over-the-air behavior to ensure compatibility of Bluetooth devices from different vendors. It defines the complete system from the radio up to the application level, including the software stack.

## Evolution of Bluetooth

When we talk about each iteration of Bluetooth, three factors help distinguish between the different versions: **range**, **data speed and power consumption**. These factors are determined by the modulation scheme and data packet being used. When the first version of Bluetooth came out, it paved the way for the wireless headphones, speakers and game controllers that we use today. However, back then Bluetooth 1.0 **was far slower than what we have now**. Data speeds capped off at 1 Mbps and the range only reached as far as 10 meters.

The first version of Bluetooth used a modulation scheme called Gaussian Frequency Shift Keying (GFSK). With GFSK, the modulated carrier shifts between two frequencies representing 1s and 0s.

When Bluetooth 2.0 came out, GFSK was taken out in favor of two newer schemes: p/4-DQPSK and 8DPSK, which used changes in the waveforms' phase to carry information, as opposed to frequency modulation. These two schemes resulted in unprecedented data speeds of 2 Mbps and 3 Mbps, respectively. Bluetooth 3.0 further improved data speeds with the addition of 802.11 for up to 24 Mbps of data transfer, although this was not a mandatory part of the 3.0 specification.

The results were game-changing. Short-range wireless solutions could now provide reliable, high speed connection, opening up to possibilities of major technological advancement in wireless devices.

However, one significant still prevented early versions of Bluetooth from widespread IoT integration: **power consumption**. Because of the large amount of energy that was required from Bluetooth versions 1.0 – 3.0, also known as Bluetooth Classic, small devices would continue to suffer from short battery life, making early versions of Bluetooth impractical for IoT use.

In order to meet the increasing demand for wireless connectivity between small devices, Bluetooth 4.0 was introduced to the market with a new category of Bluetooth: **Bluetooth Low Energy (BLE)**. Geared towards applications requiring low power consumption, BLE returns to a lower data throughput of 1Mbps using the GFSK modulation scheme.  Although BLE's max data throughput of 1Mbps may not be suitable for products that require a continuous stream of

data like wireless headphones, other IoT applications only need to send small bits of data periodically. An example are fitness wearables that relay small amounts of temperature data to your smartphone only when requested (from a mobile app, perhaps). With the focus on keeping energy demands low, Bluetooth Low Energy makes many coin-cell battery-operated IoT applications (e.g., beacons) feasible.

## Bluetooth 1.0 - 1.2

Bluetooth 1.0 and 1.0B had many problems and manufacturers had difficulty making their products interoperable. Versions 1.0 and 1.0B also included mandatory Bluetooth hardware device address (BD_ADDR) transmission in the connecting process (rendering anonymity impossible at the protocol level), which was a major setback for certain services planned for use in Bluetooth environments.

Bluetooth 1.1 was ratified as IEEE Standard 802.15.1 in 2002. Bluetooth 1.1 is the earliest version with about 748 - 810 Kb/s transfer rate. Since it is early design, versions 1.1 tends to be disturbed by same-frequency products, thus influencing communication quality.

Bluetooth 1.2 version is compatible with version 1.1, major enhancements include the following:

- Faster connection and discovery.
- Anonymous method: Shield Bluetooth hardware device address (BD_ADDR) to protect the users from identity sniffing attack and tracking. It has rendered hardware anonymity from version 1.1, but unfortunately, it is not carried out, so this function is useless for common clients.
- Adaptive frequency-hopping spread spectrum (AFH): It improves resistance to radio frequency interference by avoiding the use of crowded frequencies in the hopping sequence.
- Higher transmission speeds in practice, about 24Kb/s (192Kbps) in actual test.
- Introduce flow control and retransmission modes for L2CAP.

## Bluetooth 2.0 - 2.1

Though Bluetooth itself (at this point dubbed Bluetooth 1.0 or 1.0B) had been around for years, the first major version of the Bluetooth Core Specification that consumers would have interacted with was released in 2004. It fixed a lot of the problems inherent in those early Bluetooth devices – like the fact that the connections would sometimes make accessories inoperable – and fixed them. It also used a technology called EDR (or "Enhanced Data Rate") to allow for transfer speeds of up to 2.1 Mbit/s.

Bluetooth 2.1 + EDR: Bluetooth core specification version 2.1 + EDR was put forward by the Bluetooth SIG (Bluetooth Special Interest Group) on 26 July 2007.
The headline feature of Bluetooth 2.1 is secure simple pairing (SSP): this improves the pairing experience for Bluetooth devices, while increasing the use and strength of security. Version 2.1 allows other improvements, including "Extended inquiry response" (EIR), which provides more information during the inquiry procedure to allow better filtering of devices before connection; and sniff subrating, which reduces the power consumption in low-power mode.

## Bluetooth 3.0 - 3.1

The next major revision came in the form of Bluetooth 3.0, which was originally released to the world in April of 2009. Bluetooth 2.0 provided theoretical data transfer speeds of up to 24 Mbit/s, though not over the Bluetooth connection itself. With Bluetooth 3.0, the technology used the actual Bluetooth link to simply negotiate and establish a connection between two devices. The actual high rate data transfer was handled via a collocated 802.11 link, similar to the one that the wireless router in your home is probably using right now.

## Bluetooth 4.0 - 4.1 - 4.2

Next came Bluetooth 4.0, which made its debut in June of 2010. It included a few different ways of transferring data, like Classic Bluetooth and Bluetooth High Speed. Classic Bluetooth used legacy Bluetooth protocols to transfer data, meaning the same ones that had been in use for years prior. Bluetooth High Speed was actually based on current Wi-Fi standards and transferred data in largely the same way.

The Bluetooth SIG announced formal adoption of the Bluetooth 4.1 specification on 4 December 2013. This specification is an incremental software update to Bluetooth 4.0, and not a hardware update. The update incorporates Bluetooth Core Specification Addenda (CSA 1, 2, 3 & 4) and adds new features that improve consumer usability. These include increased co-existence support for LTE, bulk data exchange rates - and aid developer innovation by allowing devices to support multiple roles simultaneously.
New features of this specification include:

- Mobile wireless service coexistence signaling
- Train nudging and generalized interlaced scanning
- Low duty cycle directed advertising
- L2CAP connection oriented and dedicated channels with credit based flow control
- Dual mode and topology
- LE link layer topology
- 802.11n pal
- Audio architecture updates for wide band speech
- Fast data advertising interval
- Limited discovery time

Bluetooth 4.2 was released on December 2, 2014. It Introduces some key features for IoT. Some features, such as Data Length Extension, require a hardware update. But some older Bluetooth hardware may receive some Bluetooth 4.2 features, such as privacy updates via firmware.

## Bluetooth Protocol Stack

The Bluetooth protocol can be broken up into two main items: layers and profiles. All the layers Of the Bluetooth protocol form the protocol stack. The following layers of the Bluetooth protocol "stack up":

**1. Host Controller Interface (HCI):**

The Host Controller Interface is a layer of software that passes all your data from your computer to your attached Bluetooth device.

**2. Logical Link Control and Adaptation Protocol (L2CAP)**

The Logical Link Control and Adaptation Protocol is the core layer of the stack through which all data must pass. L2CAP boasts some powerful features like packet segmentation and reassembling of data, as well as protocol multiplexing. If you are trying to pass a very large packet of data, L2CAP breaks up the packet and sends smaller ones.

**3. Service Discovery Protocol (SDP)**

A Bluetooth device uses Service Discovery Protocol in order to discover services. 4. RFCOMM RFCOMM is commonly known as the wireless serial port, or the cable replacement protocol. The name is derived from the fact that your serial ports are called COMM1, COMM2, etc. RFCOMM simulates the functionality of a standard serial port.

**4. RFCOMM**

RFCOMM is commonly known as the wireless serial port, or the cable replacement protocol. The name is derived from the fact that your serial ports are called COMM1, COMM2, etc. RFCOMM simulates the functionality of a standard serial port.

**5. Telephony Control Protocol Specification (TCS, TCS Binary, TCSBIN)**

Telephony Control Protocol Specification (TCS, TCS Binary, TCS-BIN) is used to send control signals to devices that want to employ the audio capabilities within Bluetooth.

**6. Wireless Access Protocol (WAP)**

In Bluetooth, this is an adopted protocol, so the Bluetooth SIG has incorporated the existing WAP protocol into the Bluetooth protocol to fit Blue tooth's needs. WAP requires that PPP, IP, and UDP be present in the stack.

**7. Object Exchange (OBEX)**

OBEX is a communication protocol initially defined by the Infrared Data Association (IrDA). OBEX is pretty useful when you want to transfer objects like files between Bluetooth devices. OBEX does not require that TCP and IP be present in the stack, but the manufacturer is free to implement OBEX over TCP/IP.

**8. Bluetooth Network Encapsulation Protocol (BNEP)**

The Bluetooth Network Encapsulation Protocol is a layer in the Bluetooth stack that allows other Networking protocols to be transmitted over Bluetooth, namely Ethernet. BNEP is a popular choice because it encapsulates TCP/IP packets in L2CAP packets before handing off the data to the L2CAP layer in the stack.

**9. Human Interface Device Protocol (HID)**

The Human Interface Device Protocol is another adopted protocol in the Bluetooth specification. It was originally defined in the USB specification, and it lists the rules and guidelines for transmitting information to and from human interface devices like keyboards, mice, remote controls, and video game controllers.

# Problem 2: Questions

### 1- What is the frequency range of the Bluetooth?

2.4 **GHz** ISM spectrum band (2400 to 2483.5 **MHz**)

### 2- What is the modulation type used in Bluetooth?

GFSK, DQPSK, 8DPSK

### 3- How many channels are used in Bluetooth and what is the Bandwidth of each one?

79 Bluetooth channels each channel has a bandwidth of 1 MHz.

### 4- What are the Bluetooth power classes?

**Class 1**: maximum output power 100 mW (20 dBm). Used for extended range up to about 100 m.

**Class 2**: maximum output power 2,5 mW (4 dBm). Normal usage ranges up to about 10 m.

**Class 3**: maximum output power 1 mW (0 dBm). Short range communications from 10 cm up to 1 m.

### 5- What is the difference between master and slave devices, and what is the max number of slave devices within the piconet?

piconet uses a master/slave model to control when and where devices can send data. In this model, a single master device can be connected to up to seven different slave devices. Any slave device in the piconet can only be connected to a single master.

The master coordinates communication throughout the piconet. It can send data to any of its slaves and request data from them as well. Slaves are only allowed to transmit to and receive from their master. They can't talk to other slaves in the piconet.

## 6- Describe the frequency hopping spread spectrum concept.

Frequency-hopping spread spectrum (FHSS) is a method of transmitting radio signals by rapidly changing the carrier frequency among many distinct frequencies occupying a large spectral band. The changes are controlled by a code known to both transmitter and receiver. FHSS is used to avoid interference, to prevent eavesdropping, and to enable code-division multiple access (CDMA) communications.

## 7- What is Bluetooth low energy (LE)? Describe the main features of it.

BLE technology provides an easy and reliable interface, which is highly appreciated by consumer electronics manufacturers, mobile application developers and engineers. Bluetooth Low Energy technology operates in the same spectrum range (the 2.400–2.4835 GHz ISM band ) as classic Bluetooth technology, but uses a different set of channels. Instead of the classic Bluetooth 79 1-MHz channels, Bluetooth Low Energy has 40 2-MHz channels. Within a channel, data is transmitted using Gaussian frequency shift modulation , similar to classic Bluetooth's Basic Rate scheme. The bit rate is 1 Mbit/s (with an option of 2 Mbit/s in Bluetooth 5), and the maximum transmit power is 10 mW (100 mW in Bluetooth 5).

# Problem 3: Questions Related To The Experiment

### 1- What are the specification of RN41/RN41N Bluetooth module (data rate, coverage distance, power class, .....) .

The RN41 module is a small form factor, low power, class 1 Bluetooth radio that is ideal for designers who want to add wireless capability to their products without spending significant time and money developing Bluetooth-specific hardware and software. The RN41 supports multiple interface protocols, is simple to design in, and is fully certified, making it a complete embedded Bluetooth solution. With its high-performance, on-chip antenna and support for Bluetooth EDR, the RN41 delivers up to a 3-Mbps data rate for distances up to 100 meters. The RN41 is also available without an antenna (RN41N).

### 2- In the following questions, identify the line of code that needs to be changed and write the new commands.

### a. How to change the Bluetooth module name in the given code.

```
do {
        UART3_Write_Text("SN,Bluetooth-1111");     // Name of device
        UART3_Write(13);                           // CR
        Delay_ms(500);
    } while(BT_Get_Response() != BT_AOK);
```

### b. How to change the Bluetooth module from Slave to Master?

```
do {
        UART3_Write_Text("SO,Slave");              // Name of device
        UART3_Write(13);                           //CR
        Delay_ms(500);
    } while(BT_Get_Response() != BT_AOK);
```

# Problem 4: Mini Simulations

## Code:

```
BERarr = [];

BERarrAWGN = [];

SNRarr = [];

order = 8;

numberOfBits = 1e6;

bits = randi([0 1], numberOfBits, 1);

modulatedBits = dpskmod(bits, order);

for SNR = 0:15


   modulatedBitsWithErrAWGN = awgn(modulatedBits,SNR,'measured');

   modulatedBitsWithErr = modulatedBits + (randn(numberOfBits, 1) + 1i*randn(numberOfBits,
1)/(2*sqrt(SNR)));

   demodulatedBitsWithErr = dpskdemod(modulatedBitsWithErr,order);

   demodulatedBitsWithErrAWGN = dpskdemod(modulatedBitsWithErrAWGN,order);


   for i = 1:numberOfBits

     if (demodulatedBitsWithErr(i)>0)

        demodulatedBitsWithErr(i) = 1;

     end

   end


   for i = 1:numberOfBits

     if (demodulatedBitsWithErrAWGN(i)>0)

        demodulatedBitsWithErrAWGN(i) = 1;

     end

   end
```

```
    numberOfErrs = biterr(bits,demodulatedBitsWithErr);
    numberOfErrsAWGN = biterr(bits,demodulatedBitsWithErrAWGN);
    BERAWGN = numberOfErrsAWGN/numberOfBits;
    BER = numberOfErrs/numberOfBits;
    SNRarr = [SNRarr SNR]; %#ok<AGROW>
    BERarr = [BERarr BER]; %#ok<AGROW>
    BERarrAWGN = [BERarrAWGN BERAWGN]; %#ok<AGROW>
end

%plot(SNRarr, BERarr)
semilogy(SNRarr,BERarr)
xlabel('SNR ');
ylabel('BER');
title('8DPSK. Noise: (randn(numberOfBits, 1) + 1i*randn(numberOfBits, 1)/(2*sqrt(SNR)))');

figure
semilogy(SNRarr, BERarrAWGN)
xlabel('SNR');
ylabel('BER');
title('8DPSK. Noise: AWGN');
```

## Screenshots: