**Prelab-6**

1. What are the applications of Bluetooth protocol?

* In laptops, notebooks and wireless PCs
* In mobile phones and PDAs (personal digital assistant).
* In printers.
* In wireless headsets.
* In wireless PANs (personal area networks) and even LANs (local area networks)
* To transfer data files, videos, and images and MP3 or MP4.
* In wireless peripheral devices like mouse and keyboards.
* In data logging equipment.
* In the short-range transmission of data from sensors devices to sensor nodes like mobile phones.

1. What are Bluetooth profiles?

Bluetooth Profiles are a set of rules that allow the technology to complete a particular task. For example, to connect a pair of headphones to another device, a specific Bluetooth profile (or rules) is used. A different Bluetooth profile is needed to transfer files from one device to another.

1. How is Bluetooth security implemented?

Bluetooth security is used to protect services offered by devices as well as enforce exclusivity, permitting only very specific devices to connect. In accomplishing this end, the security troika was introduced consisting of authentication, authorization, and [encryption](https://www.sciencedirect.com/topics/engineering/cryptography). Specific use of these fundamental building blocks was then discussed in context of three different security modes; Mode 1 was the easiest to understand as it refers to no security, Mode 2 enforces the security troika at the L2CAP and RFCOMM protocol layers, while Mode 3 enforces authentication and encryption at the Link Manager level.

1. Can we use Bluetooth products on airlines? Justify your answer.

Even if the airline allows passengers to use bluetooth devices during flight, be mindful of the fact that using cellular data is still [forbidden](https://getaway.10best.com/12782911/what-are-tsa-prohibited-items). Bluetooth devices and the electronics you connect them to must be switched to airplane mode. If switching the device to airplane mode doesn't allow you to connect via bluetooth, manually turning off cellular service might be a workable alternative.Also keep in mind that traveling with bluetooth headphones won't allow you to connect to the airplane's entertainment system, if it has one. These headphones also run on battery power. If the battery dies before the flight ends, you might be stuck sitting in silence. Stash a pair of cheap wired earbuds in your carry-on bag to use during flight; then switch back to high-quality wireless headphones once you're back on the ground.

**Experiment-6: Program to blink LED using Arduino Uno Board**

1. **Aim:** To blink LED using Arduino Uno Board

**Description:** The Arduino Uno is an [open-source](https://en.wikipedia.org/wiki/Open-source) [microcontroller board](https://en.wikipedia.org/wiki/Microcontroller_board) based on the [Microchip](https://en.wikipedia.org/wiki/Microchip_Technology) [ATmega328P](https://en.wikipedia.org/wiki/ATmega328P) microcontroller and developed by Arduino. The board is equipped with sets of digital and analog [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O) pins that may be interfaced to various [expansion boards](https://en.wikipedia.org/wiki/Expansion_board) (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the [Arduino IDE](https://en.wikipedia.org/wiki/Arduino#Software) (Integrated Development Environment) via a type B [USB cable](https://en.wikipedia.org/wiki/USB_cable).[[4]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-priceton-4) It can be powered by the USB cable or by an external [9-volt battery](https://en.wikipedia.org/wiki/9-volt_battery), though it accepts voltages between 7 and 20 volts.



**Source Code :**

/\* Blink Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain. \*/

// Pin 13 has an LED connected on most Arduino boards.

int led = 13;

void setup() {

// initialize the digital pin as an output.

pinMode(led, OUTPUT);

}

 // the loop routine runs over and over again forever:

void loop() {

digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(led, LOW); // turn the LED off by making the voltage LOW

delay(1000); // wait for a second

}

**Input:** Connecting LED to Pin 13

**Output:** LED blink for every 1sec

**Experiment-VII: Demonstrate communication protocol Bluetooth**

**7.(a) Aim:** Program to demonstrate Mobile to PI communication using Bluetooth protocol

**Description:** Bluetooth is a wireless technology used to transfer data between different electronic devices. It is mainly designed for communicating over short distances less than about 10m or 30ft.A computer network in which one centralized, powerful computer (called the server) is a hub to which many less powerful personal computers or workstations (called clients) are connected Radio frequency communication (RFCOMM) The Bluetooth protocol RFCOMM is a simple set of transport protocols, made on top of the L2CAP protocol, providing emulated RS-232 serial ports. RFCOMM is sometimes called serial port emulation.

**Procedure:**

**Package Installation Codes**

* sudo apt-get install bluetooth
* sudo apt-get install bluez
* sudo apt-get install python-bluez

**Step 1**: First you have to install Pi3 Bluetooth terminal in your Android device through playstore

**Step 2:** Open the LX-Terminal in Raspberry PI and enter the below commands one by one

* sudo bluetoothctl
* power on
* agent on
* default-agent
* discoverable on
* scan on

After the scan command it will show the nearby Bluetooth devices

Step3: Pair the mobile device with Raspberry PI. Once devices got connected you can list out the paired devices enter the below command

* paired-devices

Step 4: Go to Menu🡪Programming🡪Python 2.7 IDLE🡪File🡪New

Step 5: Write your server program in Raspberry PI

Step 6: Save & Run the program (Run🡪Run Module)

Step 7: First open and run the python script from Server device.

**SourceCode:**

**#Server**

import bluetooth

import time

server\_sock=bluetooth.BluetoothSocket( bluetooth.RFCOMM )

port = 1

server\_sock.bind(("",port))

server\_sock.listen(1)

client\_sock,address = server\_sock.accept()

print"accepted connection from",address

while True:

data=client\_sock.recv(1024)

print"received[%s]" % data

time.sleep(0.5)

text=raw\_input("enter your msg")

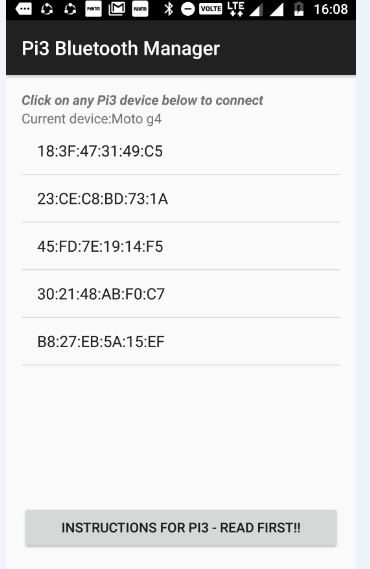
client\_sock.send(text)

time.sleep(0.5)

client\_sock.close()

server\_sock.close()

**Output:**



**7.(b) AIM:** Program to demonstrate PI to PI communication using Bluetooth protocol

**Description:** Bluetooth is a wireless technology used to transfer data between different electronic devices. It is mainly designed for communicating over short distances less than about 10m or 30ft.A computer network in which one centralized, powerful computer (called the server) is a hub to which many less powerful personal computers or workstations (called clients) are connected Radio frequency communication (RFCOMM) The Bluetooth protocol RFCOMM is a simple set of transport protocols, made on top of the L2CAP protocol, providing emulated RS-232 serial ports. RFCOMM is sometimes called serial port emulation.

**Procedure:**

**Package Installation Codes**

* sudo apt-get install bluetooth
* sudo apt-get install bluez
* sudo apt-get install python-bluez

**Step 1**: First you have to pair the two devices

Open the LX-Terminal in two Raspberry pi and enter the below commands one by one

* sudo bluetoothctl
* power on
* agent on
* default-agent
* discoverable on
* scan on

After the scan command it will show the nearby Bluetooth devices

**Step 2:** Then pair the device with your client using the below command

* pair <MAC address of client>

It will ask pairing request and click OK to connect

**Step 3:** Once devices got connected you canlist out the paired devices enter the below command

* paired-devices

**Step 5:** Go to Menu🡪Programming🡪Python 2.7 IDLE🡪File🡪New

**Step 6:** Write your program in separate file for server and client

**Step 7:** Save & Run the program (Run🡪Run Module)

**Step 8:** First open and run the python script from Server device. Then open and run the python script from client device

**Source Code:**

import bluetooth

import time

server\_sock=bluetooth.BluetoothSocket( bluetooth.RFCOMM )

port = 1

server\_sock.bind(("",port))

server\_sock.listen(1)

client\_sock,address = server\_sock.accept()

print"accepted connection from",address

while True:

data=client\_sock.recv(1024)

print"received[%s]" % data

time.sleep(0.5)

text=raw\_input("enter your msg")

client\_sock.send(text)

time.sleep(0.5)

client\_sock.close()

server\_sock.close()

**#Client**

import bluetooth

import time

bd\_addr = "E4:46:DA:A0:E2:50" # MAC address of server

port = 1

sock=bluetooth.BluetoothSocket( bluetooth.RFCOMM )

sock.connect((bd\_addr,port))

while True:

text=raw\_input("enter your msg:")

sock.send(text)

time.sleep(0.5)

data=sock.recv(1024)

print"received[%s]" % data

time.sleep(0.5)

sock.close()

**Output:**

