ANSWER 1

Bluetooth works in pretty versatile modes some of them are Advertise, Scanner, Slave, Master, Initiator in which the master is converted by the initiator and the scanner & the slave device is converted by the broadcaster.  
**Master mode:**   
Works in master mode and can be connected to a slave.   
In this mode, you can search for surrounding devices and select the slave devices you want to connect to, A master device can communicate up to seven Bluetooth slave devices at the same time.  
**Slave mode:** The Bluetooth module working in slave mode can only be searched by the host and cannot be actively searched.  
After the device is connected to the host, it can also send and receive data with the host device.  
the major difference between Master and Slave is The host/master refers to the party that can search for others and actively establish a connection. Whereas a slave cannot establish a connection actively, and can only wait for others to connect with themselves.  
  
ANSWER 2  
  
#include <SoftwareSerial.h> //Library for Serial Communication via GPIO

#define ledPin D5 // Pin connected to LED

char data; //Variable to store incoming data

SoftwareSerial mybt(D2, D1); // RX, TX

void setup() {

Serial.begin(9600);

pinMode(ledPin,OUTPUT);

while (!Serial) {

; // wait for serial port to connect. Needed for native USB port only

}

mybt.begin(9600); // Work in Normal Mode

}

void loop() {

data = mybt.read();

Serial.println(data);

int val = data;

val = map(val, 0,100, 0, 255);

for(int val = 0;val<100;val++)

{

analogWrite(ledPin,val);

delay(5);

}  
  
ANSWER 3  
I2C is abbreviated as Inter-Integrated Circuit Protocol it combines the best features of SPI and UARTs. I2C is a serial protocol for two-wire interface to connect low-speed devices like microcontrollers, EEPROMs, A/D and D/A converters, I/O interfaces and other similar peripherals in embedded systems. It was invented by Philips and now it is used by almost all major IC manufacturers.  
I2C uses only two wires those are for SCL (serial clock) and SDA (serial data). Both need to be pulled up with a resistor to +Vdd. There are also I2C level shifters which can be used to connect to two I2C buses with different voltages.   
here some advantages of I2C protocol

* maintains low pin/signal count even with numerous devices on the bus
* adapts to the needs of different slave devices
* readily supports multiple masters
* incorporates ACK/NACK functionality for improved error handling

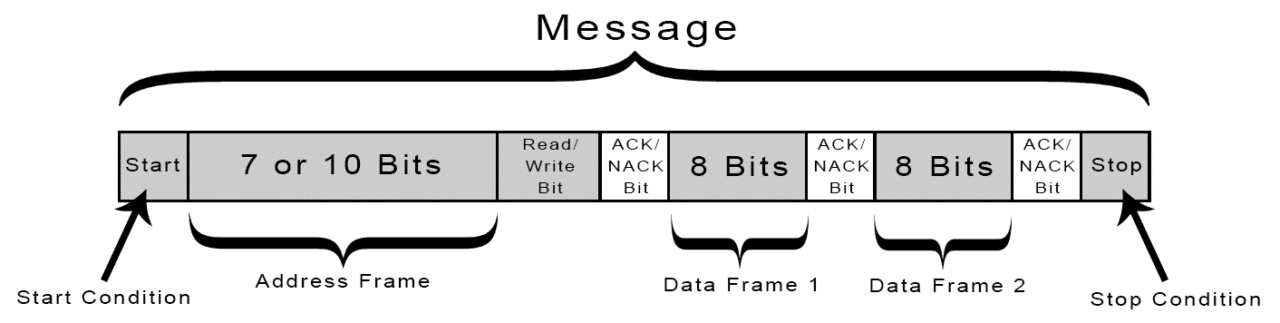
And here are some disadvantages of I2C protocol

* increases the complexity of firmware or low-level hardware
* imposes protocol overhead that reduces throughput
* requires pull-up resistors, which
* limit clock speed
* consume valuable PCB real estate in extremely space-constrained systems
* increase power dissipation

ANSWER 6

I2C bus is popular because it is simple to use, there can be more than one master and can connect almost unlimited number of I2C devices. Each slave device has a unique address. Transfer from and to master device is serial and it is split into 8-bit packets. I2C communication is using transfers of 8 bits or bytes. Each I2C slave device has a 7-bit address that needs to be unique on the bus. In this 7-bit address represents bits 7 to 1 while bit 0 is used to signal reading from or writing to the device. If bit 0 (in the address byte) is set to 1 then the master device will read from the slave I2C device.

data is transferred in messages and Messages are broken up into frames of data   
includes start and stop conditions, read/write bits, and ACK/NACK bits between each data frame  
**Start Condition:** The SDA line switches from a high voltage level to a low voltage level before the SCL line switches from high to low.  
**Stop Condition:** The SDA line switches from a low voltage level to a high voltage level after the SCL line switches from low to high.  
**Address Frame:** A 7 or 10 bit sequence unique to each slave that identifies the slave when the master wants to talk to it.



**Read/Write Bit:** A single bit specifying whether the master is sending data to the slave (low voltage level) or requesting data from it (high voltage level).

**ACK/NACK Bit**: Each frame in a message is followed by an acknowledge/no-acknowledge bit. If an address frame or data frame was successfully received, an ACK bit is returned to the sender from the receiving device.  
After the master detects the ACK bit from the slave, the first data frame is ready to be sent. The data frame is always 8 bits long, and sent with the most significant bit first. Each data frame is immediately followed by an ACK/NACK bit to verify that the frame has been received successfully. The ACK bit must be received by either the master or the slave (depending on who is sending the data) before the next data frame can be sent.

ANSWER 4   
Let’s break the terminology in three parts. The communication is very well known terminology which involves the exchange of information between two or more mediums. This exchange of data bits in microcontroller is done by some set of defined rules known as communication protocols. Now if the data is sent in series i.e. one after the other then the communication protocol is known as Serial Communication Protocol. More specifically, the data bits are transmitted one at a time in sequential manner over the data bus or communication channel in Serial Communication. Some examples of Serial Communication Protocols such as CAN, ETHERNET, I2C, SPI, RS232, USB, 1-Wire, and SATA etc. There are some difference in both the protocols and those are:

Synchronous mode requires both data and a clock.   
Asynchronous mode requires only data.

In synchronous mode, the data is transmitted at a fixed rate.   
In asynchronous mode, the data does not have to be transmitted at a fixed rate.

Synchronous data is normally transmitted in the form of blocks.   
In asynchronous data is normally transmitted one byte at a time.

Synchronous mode allows for a higher DTR (data transfer rate) than asynchronous mode does, if all other factors are held constant.

ANSWER 7

An Internet Protocol address is also known as IP address. It is a number which assigned to each device connected to a computer network which uses the IP for communication.

IPv4

* IPv4 was the first version of IP. It was deployed for production in the ARPANET in 1983. Today it is most widely used IP version. It is used to identify devices on a network using an addressing system.
* The IPv4 uses a 32-bit address scheme allowing to store 2^32 addresses which is more than 4 billion addresses. Till date, it is considered the primary Internet Protocol and carries 94% of Internet traffic.

IPv6

* It is the most recent version of the Internet Protocol. Internet Engineer Taskforce initiated it in early 1994. The design and development of that suite is now called IPv6.
* This new IP address version is being deployed to full fill the need for more Internet addresses. It was aimed to resolve issues which are associated with IPv4. With 128-bit address space, it allows 340 undecillion unique address space. IPv6 also called IPng (Internet Protocol next generation).

The major difference in IPv4 and IPv6 are :

1. IPv4 is 32-Bit IP address.  
   IPv6 is a 128-Bit IP address.
2. IPv4 is a numeric addressing method.  
   IPv6 is an alphanumeric addressing method.
3. IPv4 binary bits are separated by a dot(.).  
   IPv6 binary bits are separated by a colon (:).
4. IPv4 offers 12 header fields.  
   IPv6 offers 8 header fields.
5. IPv4 supports broadcast.   
   IPv6 doesn’t support broadcast.
6. IPv4 has checksum fields.  
   IPv6 doesn’t have checksum fields
7. IPv4 supports VLSM (Virtual Length Subnet Mask).  
   IPv6 doesn’t support VLSM.
8. IPv4 uses ARP (Address Resolution Protocol) to map to MAC address.  
   IPv6 uses NDP (Neighbour Discovery Protocol) to map to MAC address.

An application layer protocol defines how application processes client and server running on different end systems and pass message to each other, there are so many protocols for application layer

ANSWER 9

TCP/IP helps you to determine how a specific computer should be connected to the internet and how you can transmit data between them. It helps us to create a virtual network when multiple computer networks are connected.  
TCP/IP stands for Transmission Control Protocol/ Internet Protocol. It is specifically designed as a model to offer highly reliable and end-to-end byte stream over an unreliable internetwork.

UDP is a Datagram oriented protocol. It is used for broadcast and multicast type of network transmission. The full form of UDP is User Datagram Protocol A datagram is a transfer unit associated with a packet-switched network. The UDP protocol works almost similar to TCP, but it throws all the error-checking stuff out, all the back-and-forth communication and deliverability.

There are some key difference in both the protocol those are:

1. TCP is a connection-oriented protocol.  
   UDP is a connectionless protocol.
2. The speed for TCP is slower.  
   The speed of UDP is faster
3. TCP uses handshake protocol like SYN, SYN-ACK, ACK.  
   UDP uses no handshake protocols
4. TCP does error checking and also makes error recovery.  
   UDP performs error checking, but it discards erroneous packets.
5. TCP has acknowledgment segments.  
   UDP does not have any acknowledgment segment.
6. TCP is heavy-weight, TCP needs three packets to set up a socket connection before any user data can be sent.  
   and UDP is lightweight, There are no tracking connections, ordering of messages.

**Advantages of Transmission control protocol**

1. It helps you to establish/set up a connection between different types of computers.
2. Operates independently of the operating system
3. Supports many routing-protocols.
4. It enables the internetworking between the organizations.
5. It can be operated independently.
6. Supports several routing protocols.
7. TCP can be used to establish a connection between two computers

**Advantages of User Datagram Protocol**

1. It never restricts you to a connection-based communication model; that's why startup latency in distributed applications is low.
2. The recipient of UDP packets gets them unmanaged, which also includes block boundaries.
3. Broadcast and multicast transmission are also available with UDP
4. Data loss can be made
5. Small transaction (DNS lookup)
6. Bandwidth intensive app which endures packet loss.

**Disadvantages of Transmission control protocol**

1. TCP never conclude a transmission without all data in motion being explicitly asked.
2. You can't use for broadcast or multicast transmission.
3. TCP has no block boundaries, so you need to create your own.
4. TCP offers many features that you don't want. It may waste bandwidth, time, or effort.
5. In this, model the transport layer does not guarantee delivery of packets.
6. Replacing protocol in TCP/IP is not easy.
7. It doesn't offer clear separation from its services, interfaces, and protocols.

**Disadvantages of User Datagram Protocol**

1. In UDP protocol, a packet may not be delivered or delivered twice. It may be delivered out of order, so you get no indication.
2. Routers are quite careless with UDP, so they never retransmit it if it collides.
3. UDP has no Congestion Control, and flow control, so implementation is the job of a user application.
4. UDP mostly like to suffer from worse packet loss

ANSWER 8

Advantages of IOT

1. Data: The more the information, the easier it is to make the right decision. Knowing what to get from the grocery while you are out, without having to check on your own, not only saves time but is convenient as well.
2. Tracking: The computers keep a track both on the quality and the viability of things at home. Knowing the expiration date of products before one consumes them improves safety and quality of life. Also, you will never run out of anything when you need it at the last moment.
3. Time: The amount of time saved in monitoring and the number of trips done otherwise would be tremendous.
4. Money: The financial aspect is the best advantage. This technology could replace humans who are in charge of monitoring and maintaining supplies.

Disadvantages of IOT

1. Compatibility: As of now, there is no standard for tagging and monitoring with sensors. A uniform concept like the USB or Bluetooth is required which should not be that difficult to do.
2. Complexity: There are several opportunities for failure with complex systems. For example, both you and your spouse may receive messages that the milk is over and both of you may end up buying the same. That leaves you with double the quantity required. Or there is a software bug causing the printer to order ink multiple times when it requires a single cartridge.
3. Privacy/Security: Privacy is a big issue with IoT. All the data must be encrypted so that data about your financial status or how much milk you consume isn’t common knowledge at the work place or with your friends.
4. Safety: There is a chance that the software can be hacked and your personal information misused. The possibilities are endless. Your prescription being changed or your account details being hacked could put you at risk. Hence, all the safety risks become the consumer’s responsibility.

Applications of IOT

Wearables  
Smart Home Applications  
Health Care  
Smart Cities  
Agriculture  
Industrial Automation

ANSWER 10

In any way Parity is a method of detecting errors in transmission. When parity is used with a serial port, an extra data bit is sent with each data character, arranged so that the number of 1 bits in each character, including the parity bit, is always odd or always even. If a byte is received with the wrong number of 1s, then it must have been corrupted. However, an even number of errors can pass the parity check.

ANSWER 2

#include <ESP8266WiFi.h> //Library for WiFi Connection

int outputpin = D5;

int v0,res;

int port = 8888; //Port number

WiFiServer server(port);

//Server connect to WiFi Network

const char \*ssid = "xxxxx"; //Enter your wifi SSID

const char \*password = "xxxxxx"; //Enter your wifi Password

void setup()

{

Serial.begin(9600);

WiFi.mode(WIFI\_STA); // WIFi on Station Mode

WiFi.begin(ssid, password); //Connect to wifi

// Wait for connection

Serial.println("Connecting to Wifi");

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

delay(500);

}

Serial.print("Connected to WiFi");

Serial.print("IP address: ");

Serial.println(WiFi.localIP()); // To find local IP

server.begin(); // Start the server

Serial.print("Open Telnet and connect to IP:");

Serial.print(WiFi.localIP());

Serial.print(" on port ");

Serial.println(port);

}

void loop()

{

int analogValue = analogRead(outputpin);

v0 = analogValue\*3.3/1024;

res = (10000\*v0)/(3.3-v0);

Serial.print("resistance: ");

Serial.println(res);

Serial.print(v0);

WiFiClient client = server.available();

if (client) {

if(client.connected())

{

Serial.println("Client Connected");

}

while(client.connected()){

while(client.available()>0){

// read data from the connected client

Serial.write(client.read());

}

//Send Data to connected client

while(Serial.available()>0)

{

client.write(Serial.read());

}

}

client.stop(); // If client disconnects

Serial.println("Client disconnected");

}

}