Research Work-1

MQTT PROTOCOL:-

MQTT is a standard -based messaging protocol,or set of the rules ,used for machine to machine communication.Smart sensors and some other Internet of things(IOT) devices are used MQTT protocol for their communication.MQTT is type of communication protocl which will transmit the data from one device to another device.MQTT supports messaging between devices to the cloud and the cloud to the device.By using MQTT protocol it is easy to implement and can communicate data efficiently.

Features of MQTT Protocol:-

1.Lightweight and efficient:MQTT implementation on the IoT device reqiures the minimal resources,so it can even be used on small microcontrollers.

2.Scalable: MQTT has built -in features to support communication with a large number of IoT devices.Hence,you can implement the MQTT protocol to connect millions of these devices.

3.Reliable: MQTT has the built in features that reduce the time of the IoT devices to reconnect to the cloud.

4.Secure: MQTT makes it easy for developers to encrypt messages and authenticate devices using Modern authenication protocols.

5.Well-supported: MQTT protocol can support all type of advanced programming languages for the implementation (eg. Python).

What is the Principle Behind MQTT?

The MQTT protocol works on the Principles of the Publish/subscribe model.The cilents request resources or data from the server,then the server processes and sends back a response.Howerver the MQTT uses a publish/subscribepattern to decouple the messege sender publisher from the messege receiver suscriber. Instead , a third component called a

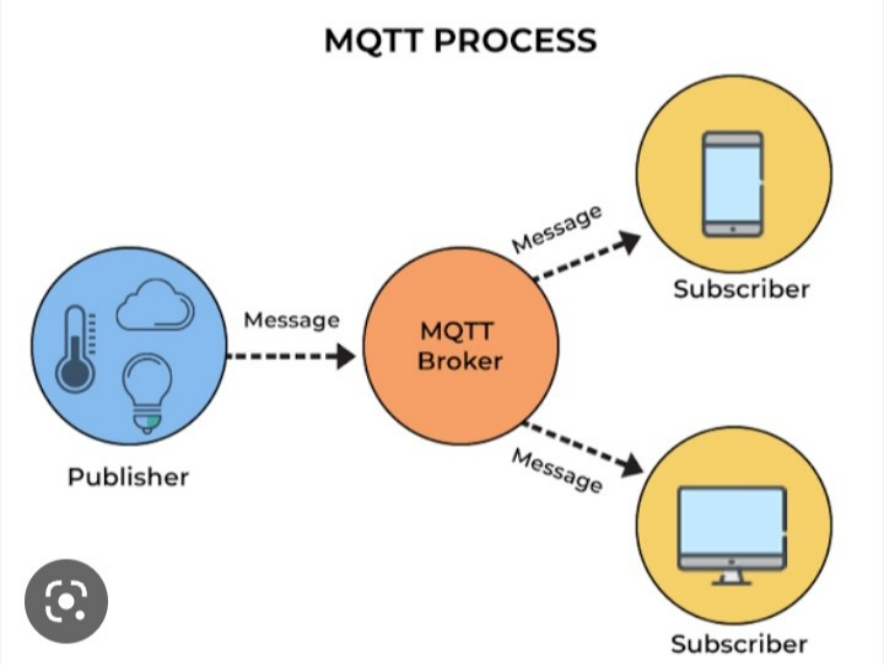
messege broker ,handles the communication between publishers and subscribers.

The broker decouples the publishers and subscribers as below:

i).Space decoupling

ii).Time decoupling

iii).synchronization decoupling



Components of MQTT Protocol:-

1.MQTT CILENT:-

An MQTT cilent is any device from a server to a microcontroller that runs an MQTT Library.If the cilent is sending messages,its acts a Publisher.and if it is receving messages ,it acts as a recevier.Any device that communicates using MQTT over a network is called an MQTT cilent device.

2.MQTT broker:-

The MQTT broker is the backend system which coordinates messages between the differnt cilents.Responsibities of the broker include receving and filtering the messages,identifying cilents and subscribed to each message,and sending them the messages,It is also responsible for other tasks such as:

1.Authorizing and authenticating MQTT cilents.
2.Passing messages to other systems for further analysis.
3.Handling missed messages and cilent sessions.

3.MQTT Connection:-

Clients and brokers begin communicating by using an MQTT connection. Clients initiate the connection by sending a *CONNECT* message to the MQTT broker. The broker confirms that a connection has been established by responding with a *CONNACK* message. Both the MQTT client and the broker require a TCP/IP stack to communicate. Clients never connect with each other, only with the broker.

How does MQTT works?

The MQTT protocol works as below:
1.An MQTT client establishes a connection with the MQTT broker.

2.Once connected, the client can either publish messages, subscribe to specific messages, or do both.

3.When the MQTT broker receives a message, it forwards it to subscribers who are interested.

MQTT TOPIC:-

The term ‘topic’ refers to keywords the MQTT broker uses to filter messages for the MQTT clients. Topics are organized hierarchically, similar to a file or folder directory. For example, consider a smart home system operating in a multilevel house that has different smart devices on each floor.
MQTT broker may organize topics as:
eg:-ourhome/groundfloor/livingroom/light

MQTT PUBLISH:-

MQTT clients publish messages that contain the topic and data in byte format. The client determines the data format such as text data, binary data, XML, or JSON files.
For example, a lamp in the smart home system may publish a message *on* for the topic *livingroom/light*.

MQTT SUBSCRIBER:-

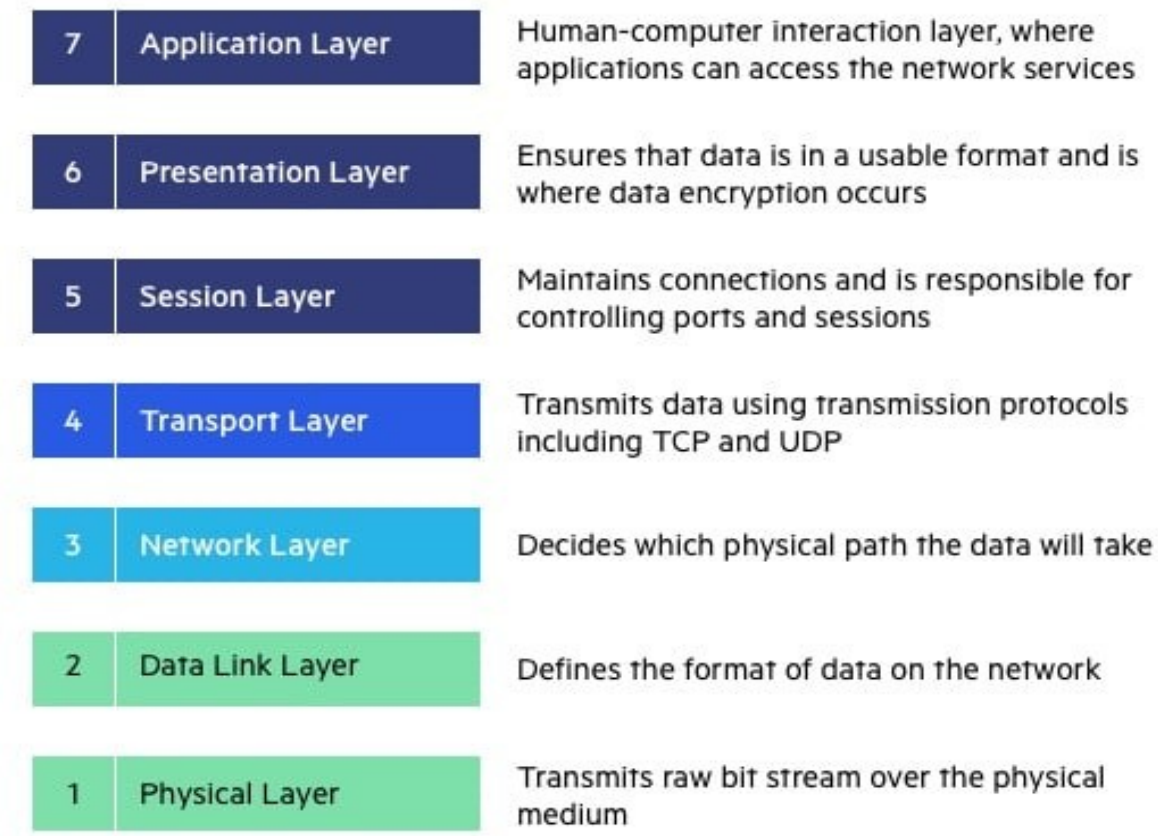
MQTT clients send a *SUBSCRIBE* message to the MQTT broker, to receive messages on topics of interest. This message contains a unique identifier and a list of subscriptions.
For example, the smart home app on your phone wants to display how many lights are on in your house. It will subscribe to the topic *light* and increase the counter for all *on* messages.

OSI MODEL:-

The OSI mode is known as the Open Systems Interconnection (OSI) model describes seven layers that computer systems use to communicate over a network. It was the first standard model for network communications

However, the OSI 7-layer model is still widely used, as it helps visualize and communicate how networks operate, and helps isolate and troubleshoot networking problems.

In the OSI model there are 7 layers as below:-



1.Application Layer:-

The application layer is used by end-user software such as web browsers and email clients. It provides protocols that allow software to send and receive information and present meaningful data to users. A few examples of application layer protocols are the Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Post Office Protocol (POP), Simple Mail Transfer Protocol (SMTP), and Domain Name System (DNS).
The application layer is the responsible for the data transmision from sender to to user.It will provide it's services to all layers.

2.Presentation layer:-

The presentation layer prepares data for the application layer. It defines how two devices should encode, encrypt, and compress data so it is received correctly on the other end. The presentation layer takes any data transmitted by the application layer and prepares it for transmission over the session layer.
It is concerned about syntax and semantics of information exchanged between two devices.
i)Translation
ii) encryption
iii)compression.

3.Session layer:-

The session layer creates communication channels, called sessions, between devices. It is responsible for opening sessions, ensuring they remain open and functional while data is being transferred, and closing them when communication ends. The session layer can also set checkpoints during a data transfer—if the session is interrupted, devices can resume data transfer from the last checkpoint.

4.Transport layer:-

The transport layer takes data transferred in the session layer and breaks it into "segments" on the transmitting end. It is responsible for reassembling the segments on the receiving end, turning it back into data that can be used by the session layer. The transport layer carries out flow control, sending data at a rate that matches the connection speed of the receiving device, and error control, checking if data was received incorrectly and if not, requesting it again.

Services provided by the transport layer:-

i) Port addressing
ii)Segmentation and reassembly
iii)Connection control
iv) End to End flow control
v) Error control

5.Network layer:-

The network layer has two main functions. One is breaking up segments into network packets, and reassembling the packets on the receiving end. The other is routing packets by discovering the best path across a physical network. The network layer uses network addresses (typically Internet Protocol addresses) to route packets to a destination node.
It is responsible for delivery of data from the original source to destination source

Services provided by the network layer:-

i)Logical addressing
ii) Routing
iii)Packages
iv)Flow controlling
v) Error controlling

6.Datalink layer:-

The data link layer establishes and terminates a connection between two physically-connected nodes on a network. It breaks up packets into frames and sends them from source to destination. This layer is composed of two parts—Logical Link Control (LLC), which identifies network protocols, performs error checking and synchronizes frames, and Media Access Control (MAC) which uses MAC addresses to connect devices and define permissions to transmit and receive data.

7.Physical layer:-

The physical layer is responsible for the physical cable or wireless connection between network nodes. It defines the connector, the electrical cable or wireless technology connecting the devices, and is responsible for transmission of the raw data, which is simply a series of 0s and 1s, while taking care of bit rate control.It is responsible for transmitting bits over a medium also provides electrical and mechanical specification.

Services provided by the Physical layer:-

i) Data rate
ii) line configuration

iii)Physical topology
iv) sychronization of bits.

Advantages of OSI Model:

The OSI model helps users and operators of computer networks:

- Determine the required hardware and software to build their network.
- Understand and communicate the process followed by components communicating across a network.
- Perform troubleshooting, by identifying which network layer is causing an issue and focusing efforts on that layer.

The OSI model helps network device manufacturers and networking software vendors:

- Create devices and software that can communicate with products from any other vendor, allowing open interoperability
- Define which parts of the network their products should work with.
- Communicate to users at which network layers their product operates – for example, only at the application layer, or across the stack.

ETHERNET FRAME:-

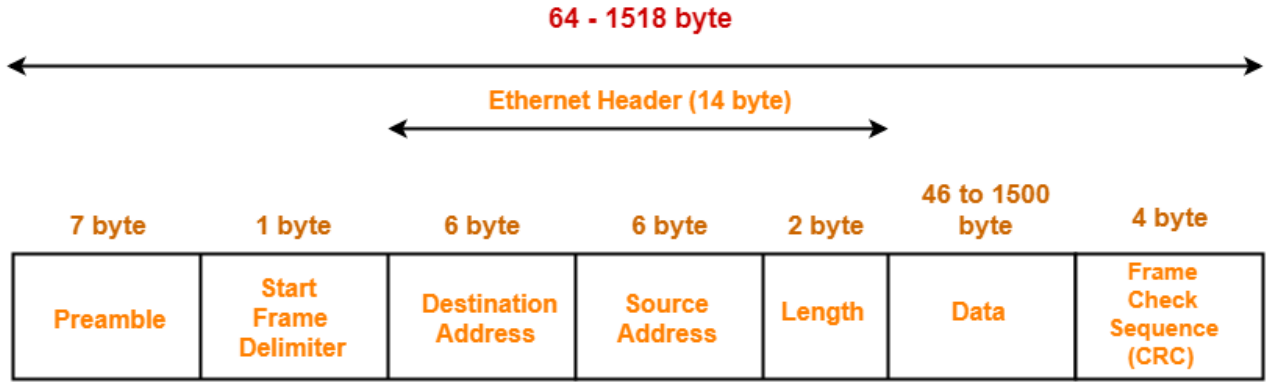
Ethernet is the traditional technology for connecting devices in a wired local area network (LAN) or wide area network (WAN). It enables devices to communicate with each other via a protocol, which is a set of rules or common network language.Ethernet is a networking technology that includes the protocol, port, cable, and computer chip needed to plug a desktop or laptop into a local area network (LAN) for speedy data transmission via coaxial or fiber optic cables.
It connects local area network (LAN) and wide area network(WAN) systems (WAN). With LAN and WAN, several devices, such as printers and laptops, may be connected across buildings, residences, and even small communities.

Ethernet was created in the early 1970s at the Xerox Palo Alto Research Center (PARC) by a group that included David Boggs and Robert Metcalfe. In 1983, the Institute of Electrical and Electronics Engineers (IEEE) ratified it as a standard.

The Ethernet protocol employs a star topology or linear bus, which is the basis for the IEEE 802.3 standard. In the OSI network structure, this protocol works bot.h the physical layer and data link layer, the first two levels. Ethernet divides the data connection layer into two distinct layers: the logical link control tier and also the medium access control (MAC) tier. Ethernet also transmits data using two components: packets and frames. The frame contains the sent data payload as well as the following:

- Both the MAC and physical addresses of the sender and recipient
- Error correction data for identifying transmission faults
- Information on Virtual LAN (VLAN) tagging, as well as the quality of service(QOS).

ETHERNET FRAME FORMAT:-



IEEE 802.3 Ethernet Frame Format

The IEEE 802.3 standard defines the fundamental frame format that is necessary for all MAC implementations.

Preamble and SFD, which operate at the physical layer, begin an Ethernet frame. The packet's payload follows the Ethernet header, which includes the MAC addresses for the source and destination. CRC, the final field, is utilized to find errors.

PREAMBLE:-However, the frame bits in high-speed Ethernet today are protected without the need for a preamble.Ethernet frames begin with a 7-byte. This is a sequence of alternate 0s and 1s that denotes the beginning of the frame and enables bit synchronization between the sender and receiver.

Start of frame Delimiter (SFD) - This 1-byte field is always set to **10101011**. The destination address is the next set of bits that will begin the

frame, as indicated by SFD. The preamble is frequently referred to as 8 Bytes since SFD is sometimes seen as a component of PRE. The SFD notifies the station or stations that synchronization is now impossible.

Destination Address - This 6-Byte element contains the MAC address of the device for which the data is intended.

Source Address - This 6-byte element contains the source machine's MAC address. Since Source Address is always a unique address (Unicast), 0 is always the least significant bit of the first byte.

Length - A 2-Byte field called Length represents the size of an Ethernet frame as a whole. Due to some inherent constraints of Ethernet, this 16-bit field can store length values from 0 to 65534, but length values greater than 1500 are not permitted.

Data - This area, sometimes referred to as the Payload, is where the real data is placed. If Internet Protocol is utilised via Ethernet, both the IP header and data will be placed here. The longest possible piece of data might be 1500 bytes long. If the data length is less than the minimum length, which is 46 bytes, padding 0's are appended to make up the difference.

Cyclic Redundancy Check (CRC) - CRC is a field of 4 bytes. The data in this field is a 32-bit hash code created using the fields for the destination address, source address, length, and data. Data is damaged if the checksum calculated by the destination differs from the checksum value supplied.

Advantages of using Ethernet:

1. Simple to implement
2. Maintenance is Easy
3. Less cost

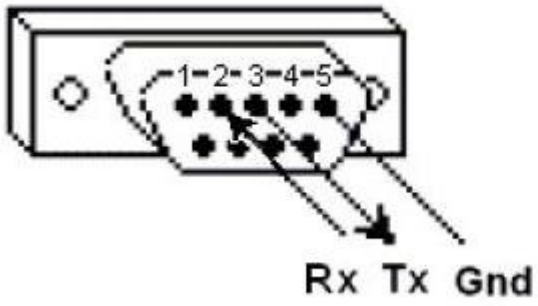
MODBUS:-

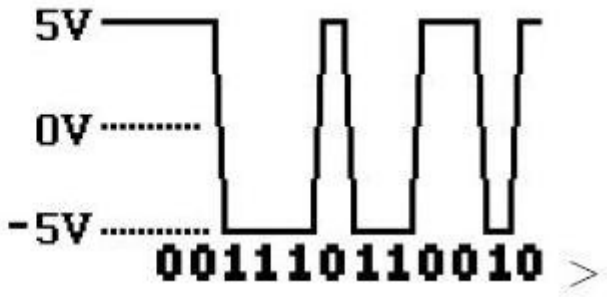
Modbus is a serial communication protocol developed by Modicon published by Modicon® in 1979 for use with its programmable logic controllers (PLCs). In simple terms, it is a method used for transmitting information over serial lines between electronic devices.

Modbus is an open protocol, meaning that it's free for manufacturers to build into their equipment without having to pay royalties. It has become a standard communications protocol in industry, and is now the most commonly available means of connecting industrial electronic devices.Modbus is typically used to transmit signals from instrumentation and control devices back to a main controller or data gathering system, for example a system that measures temperature and humidity and communicates the results to a computer. Modbus is often used to connect a supervisory computer with a remote terminal unit (RTU) in supervisory control and data acquisition (SCADA) systems. Versions of the Modbus protocol exist for serial lines (Modbus RTU and Modbus ASCII) and for Ethernet (Modbus TCP).

HOW DOES IT WORK?

Modbus is transmitted over serial lines between devices. The simplest setup would be a single serial cable connecting the serial ports on two devices, a Client and a Server.





The data is sent as series of ones and zeroes called bits. Each bit is sent as a voltage. Zeroes are sent as positive voltages and a ones as negative. The bits are sent very quickly. A typical transmission speed is 9600 baud (bits per second).

The Modbus frame structure has two types of Modbus message format has ASCII mode and RTU mode.

Modbus Frame Structure-ASCII Mode

START	ADDRESS	FUNCTION	DATA	LRC CHECK	END
1 CHAR :	2 CHARS	2 CHARS	*n* CHARS	2 CHARS	2 CHARS CRLF

Modbus Frame Structure-ASCII Mode

Each byte is encoded on serial link as 2 ASCII characters. Each ASCII character is transmitted as 1 start bit, 7 data bits, zero or 1 parity bit, one or two stop bits.

Modbus frame Structure in ASCII mode=
{
Start byte (0x3A)
Device Address (2 bytes) , Function code(2 bytes), Query Data (variable), ErrorCheck (2 bytes)
2 End Bytes (0x0D0A)
}

Modbus Frame Structure-RTU Mode

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1–T2–T3–T4*	8 BITS	8 BITS	*n* x 8 BITS	16 BITS	T1–T2–T3–T4*

*For T1–T2–T3–T4, 3.5 character times at no communication.

Modbus Frame Structure-RTU Mode

The figure RTU mode of Modbus frame. As shown in RTU (Remote terminal unit) mode, the message is transmitted in a continuous stream format. Each 8 bit byte is framed by 1 start bit, 8 data bits, 0 or 1 parity bit, 1 or 2 stop bits. The message itself starts after a silent period of at least 3.5 character times.

Let us understand the different fields of Modbus frame structure.

Modbus Address: Modbus message starts with 8 bit target address. This can take any value from 0 to 247. Here 0 is used as broadcast address and rest are used as unique device addresses.

Modbus Functions: The function code contains 2 characters (in ASCII mode) and 8 bits (in RTU mode)/ It takes any value from 1 to 255 and are selected as per application profile.

Modbus Data Field: This data field convey application level information as desired by different Modbus function. If function contains variable size of data, it begins with "byte count" in this position.

Modbus/TCP standard defines TCP/IP access to the Modbus protocol functionality.

WIRESHARK:-

Wireshark is an open-source packet analyzer, which is used for education, analysis, software development, communication protocol development, and network troubleshooting.

It is used to track the packets so that each one is filtered to meet our specific needs. It is commonly called as a sniffer, network protocol analyzer, and network analyzer. It is also used by network security engineers to examine security problems.

Wireshark is a free to use application which is used to apprehend the data back and forth. It is often called as a free packet sniffer computer application. It puts the network card into an unselective mode, i.e., to accept all the packets which it receives.

Uses of Wireshark:

1.It is used by network security engineers to examine security problems.

2.It allows the users to watch all the traffic being passed over the network.

3.It is used by network engineers to troubleshoot network issues.

4.It also helps to troubleshoot latency issues and malicious activities on your network.
5.It can also analyze dropped packets.
6.It helps us to know how all the devices like laptop, mobile phones, desktop, switch, routers, etc., communicate in a local network or the rest of the world.

Functionality of Wireshark:

Wireshark is similar to tcpdump in networking. **Tcpdump** is a common packet analyzer which allows the user to display other packets and TCP/IP packets, being transmitted and received over a network attached to the computer. It has a graphic end and some sorting and filtering functions. Wireshark users can see all the traffic passing through the network.

Wireshark can also monitor the unicast traffic which is not sent to the network's MAC address interface. But, the switch does not pass all the traffic to the port. Hence, the promiscuous mode is not sufficient to see all the traffic. The various network taps or **port mirroring** is used to extend capture at any point.

Port mirroring is a method to monitor network traffic. When it is enabled, the switch sends the copies of all the network packets present at one port to another port.

What is color coding in Wireshark?

The packets in the Wireshark are highlighted with **blue**, **black**, and **green color**. These colors help users to identify the types of traffic. It is also called as **packet colorization**.

Features of Wireshark

- It is multi-platform software, i.e., it can run on Linux, Windows, OS X, FreeBSD, NetBSD, etc.
- It is a standard three-pane packet browser.
- It performs deep inspection of the hundreds of protocols.
- It often involves live analysis, i.e., from the different types of the network like the Ethernet, loopback, etc., we can read live data.

- It has sort and filter options which makes ease to the user to view the data.
- It is also useful in VoIP analysis.
- It can also capture raw USB traffic.
- Various settings, like timers and filters, can be used to filter the output.
- It can only capture packet on the PCAP (an application programming interface used to capture the network) supported networks.
- Wireshark supports a variety of well-documented capture file formats such as the PcapNg and Libpcap. These formats are used for storing the captured data.
- It is the no.1 piece of software for its purpose. It has countless applications ranging from the **tracing down, unauthorized traffic, firewall settings, etc**.

RS-485:-

RS-485 is used as the physical layer underlying many standard and proprietary automation protocols used to implement industrial control systems, including the most common versions of Modbus and Profibus. DH 485 is a proprietary communications protocol used by Allen-Bradley in their line of industrial control units.

DEFINITION:

RS-485 is an industrial specification that defines the electrical interface and physical layer for point-to-point communication of electrical devices. The RS-485 standard allows for long cabling distances in electrically noisy environments and can support multiple devices on the same bus.

The noise immunity offered by the RS-485 standard makes the interface very versatile. RS-485 has been used in a wide range of computer automation systems dating back to when the standard was created in 1998. With the standard allowing for multi-drop (multiple devices on the same bus) and long cabling lengths, it is easy to understand its frequent use in the industrial and automation spaces. RS-485 can also be found in theater applications where many devices are spread out across a huge space.

The ability to use RS-485 at high speeds, over long cabling lengths, in electrically noisy environments, and with multiple devices on the same bus, makes it a smart implementation for most applications requiring a serial interface.

RS-485, also known as TIA-485 or EIA-485, is the standard that defines the electrical characteristics of the drivers and receivers for the communication protocol.

Physical layer of RS-485:-

The physical layer of the OSI model is responsible for the transfer of raw data between a device and a physical transmission medium. It handles the conversion of electrical signals into digital data, while defining voltages, timing, data rates, etc.

RS-485 uses two signal lines, ‘A’ and ‘B’, which must be balanced and differential. Balanced signals are two lines that share a pair in a twisted pair cable with the same impedance on each line. Along with matched impedance of the lines, there must also be matched impedance at the receiver and transmitter. Figure 2 shows a typical multi-drop RS-485 network where each device has a differential RS-485 transceiver and the link between devices is comprised of twisted pair cabling and termination resistors.

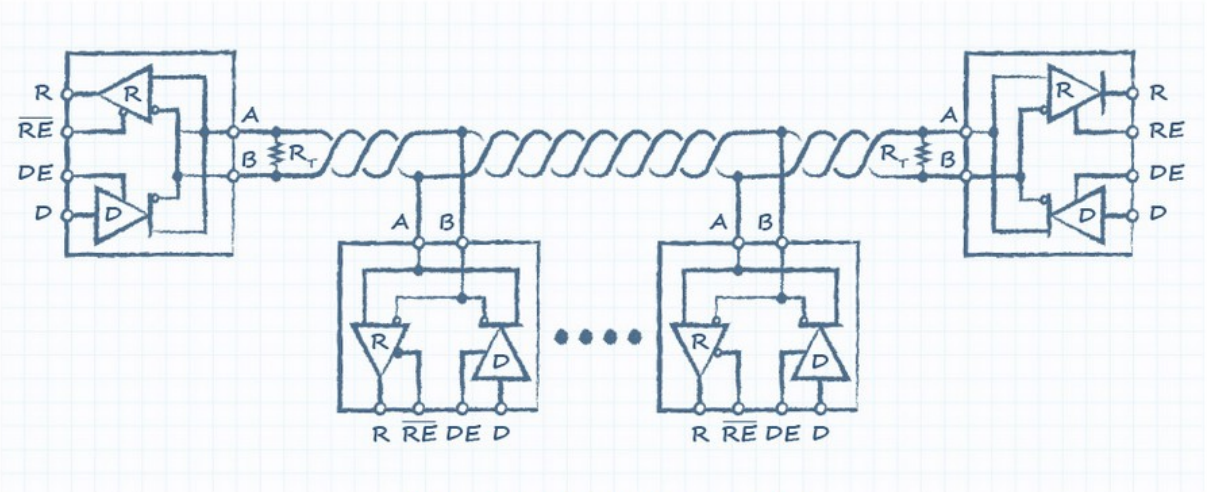


Figure 2: Typical RS-485 network topology

Note, there are various topologies that can be used to arrange devices because not all networks are created equal and termination requirements as well as device arrangement will vary. For example, in Figure 2 below termination is only used at the beginning and end of the cable.

Balanced cabling allows for noise reduction when using differential signals. These signals, ‘A’ and ‘B’ are referred to as a differential pair; one of the signals matches the original signal, while the other is entirely inverted, which is why it is sometimes referred to as a complementary signal.

In a single-ended interface, the receiver references the signal to ground, and resolves the signal state based on predetermined voltage levels (these are referred to as logic levels as they determine if the signal is logic high or logic low). However, over long cabling distances where voltages tend to drop and slew rates decrease, signal errors often occur. In a differential application, the host generates the original single-ended signal, which then goes to a differential transmitter. This transmitter creates the differential pair to be sent out over the cabling. With two signals generated, the receiver no longer references the voltage level to ground, but instead references the signals to each other. This means that rather than looking for specific voltage levels, the receiver is always looking at the *difference* between the two signals. The differential receiver then reconstructs the pair of signals back into one single-ended signal that can be interpreted by the host device using the proper logic levels required by the host, Figure 3. This type of interface also allows devices of differing voltage levels to operate together by way of communication between the differential transceivers. All this works together to overcome the signal degradation that would have occurred with a single-ended application over long cabling distances.

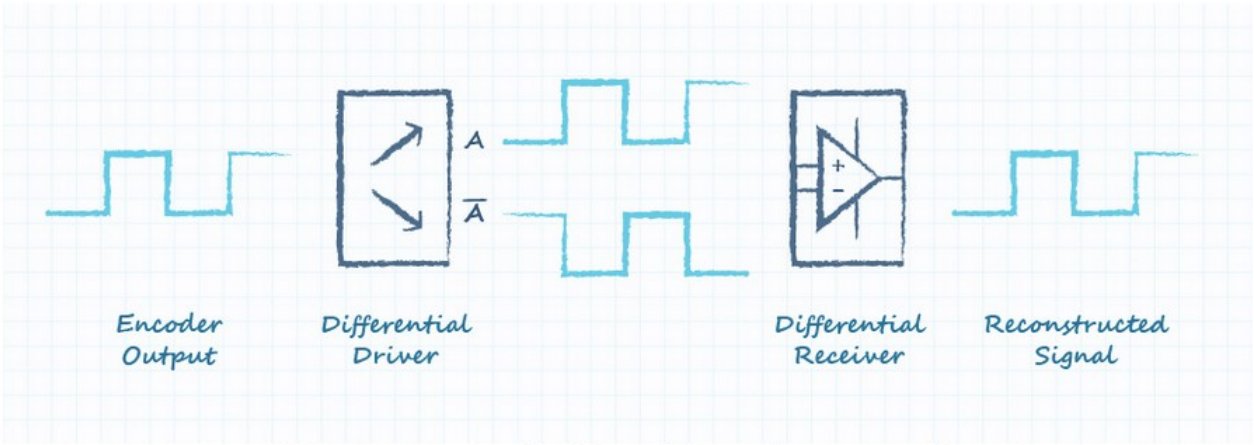


Figure 3: Encoder output driven by differential driver and reconstructed by receiver

Signal degradation is not the only issue that arises over long cabling distances. The longer the cabling is within a system, the higher the chances that electrical noise and interference will make its way onto the cables and ultimately into the electrical system. When noise couples onto cabling it shows up as voltages of varying magnitudes, but the benefit of using a balanced twisted pair cable is that the noise couples to the cable equally on each line. For instance, a positive 1-volt spike would result in +1 V on A and +1 V on B. Because the differential receiver subtracts the signals from each other to get the reconstructed signal, it would ignore the noise shown equally on both wires, Figure 4. The ability of the differential receiver to ignore voltages that are the same on both signal lines is referred to as common mode rejection.

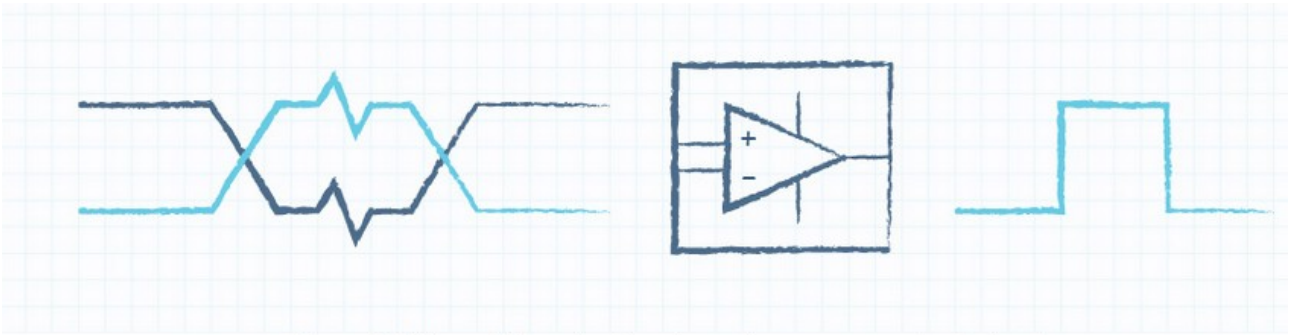


Figure 4: Differential receiver ignoring noise common to both signals

One of the other major physical layer benefits of RS-485 is the signal voltage specification. RS-485 does not require the use of a specific bus

voltage, but instead specifies the minimum required differential voltage, which is the *difference* between the signal A and B voltages. The bus requires a minimum differential voltage of +/- 200 mV at the receiver and generally all RS-485 devices will have the same input voltage range despite transmitting at various voltages. This means that any RS-485 device is able to receive the voltage range of -7 to 12 V, so an engineer can design the host system with any transmission voltage in that range. This allows designers to create RS-485 systems using their existing board voltages.

With that being said, it is important to verify the product specifications to ensure that the device supports the full voltage range of the standard. For example, CUI Devices' RS-485 encoders use 3.3 V on the board, so they use an RS-485 3.3 V transmitter. However, they are also input tolerant between 0 and 12 V. This allows them to share the same RS-485 bus at multiple different transmission voltages between 0 and 12 V without issue if the minimum differential voltage of +/- 200 mV can be met at both the receiver and the transmitter. This is especially important because as cable length increases, so does the voltage drop on the signal lines. A host device may transmit with a differential voltage of +/- 1 V but over a long cable length that voltage could diminish down to +/- 200 mV, which is still perfectly acceptable for RS-485, Figure 5.

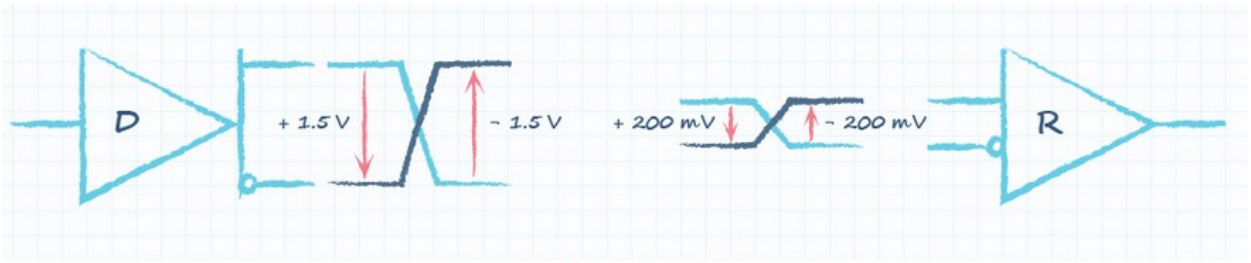


Figure 5: RS-485 minimum bus signal levels