Report 1

Mechatronics

Communication protocoles

Name:Omar Ahmed Elsayed

ID: 20010967

Third academic year

Introduction:

Communication serves as the lifeblood of organizations, driving collaboration, innovation, and efficiency across all levels. In today's digital world, communication protocols are the rules that allow devices to talk to each other. They can be defined as a set of rules and conventions that govern the exchange of data between devices, systems, or entities in a networked environment.

Communication Protocoles:

1) **USART**
2) **I2C**
3) **SPI**
4) **CAN**

At this report we will discuss the diffrance between each one

1)US ART

A USART (universal synchronous/asynchronous receiver/transmitter) is hardware that enables a device to communicate using serial protocols.USART connections were originally common in desktop computers and would be implemented in the serial port. Later, USB became the standard interface used for serial communication, and so modern desktops and laptops no longer include serial ports. In the past, the serial port would be provided over a D-sub connector (DB9) on the back side of the computer as shown below.

USART is versatile, supporting both synchronous and asynchronous communication, and can operate in full-duplex or half-duplex modes as both master and slave.

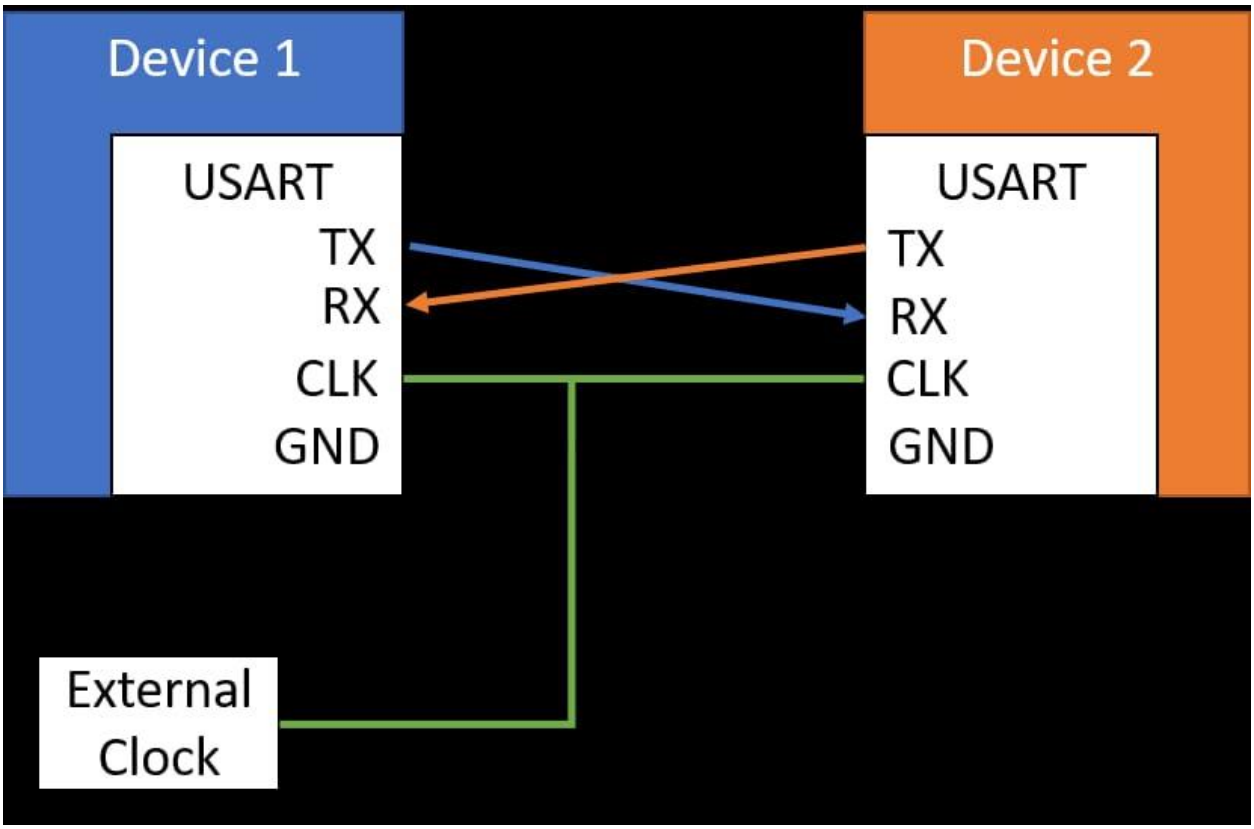

Fig1 USART-Interface.

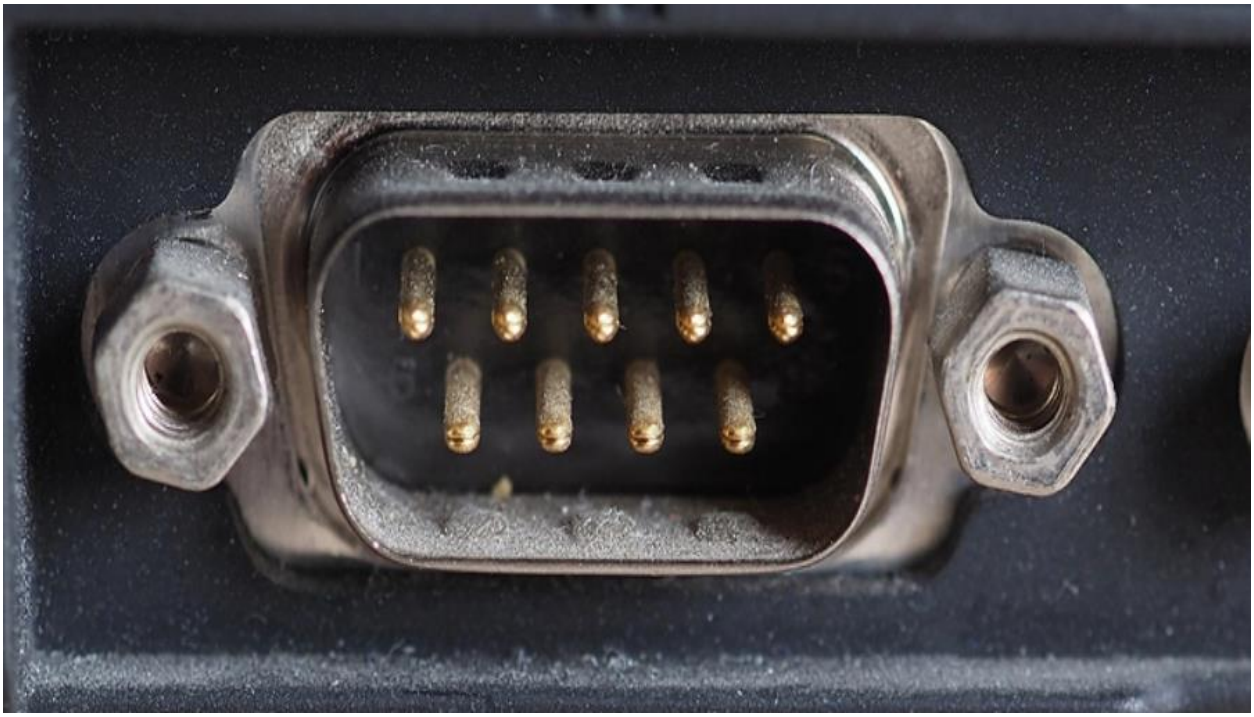

Fig2 PC serial ports were formerly used to implement USART.

The UART and USART communication happens using a data frame. The data frame consists of a start bit, data bits, parity bit and one or two stop bits. The start bit informs the receiving device that the data bits are coming. The data bits are numbers of bits of varying lengths (i.e 8-bits).

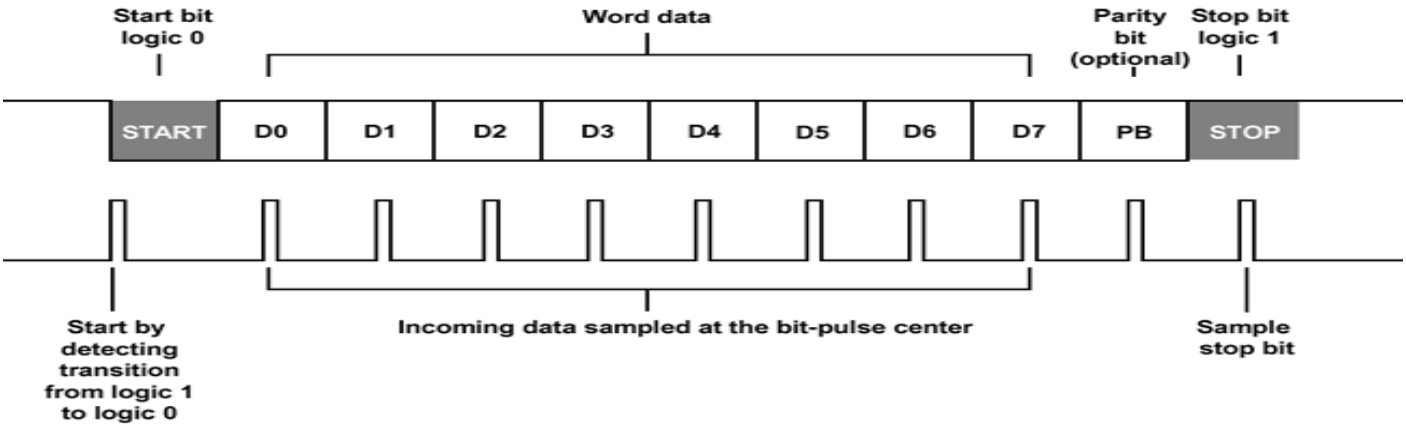

Fig3 USART-data frame.

2)I2C

I2C is a two-wire serial communication protocol using a serial data line (SDA) and a serial clock line (SCL). The protocol supports multiple target devices on a communication bus and can also support multiple controllers that send and receive commands and data.

I2C is suitable for applications where multiple devices need to communicate over a shared bus with a master controlling the communication.

Fig4 I2C connected system.

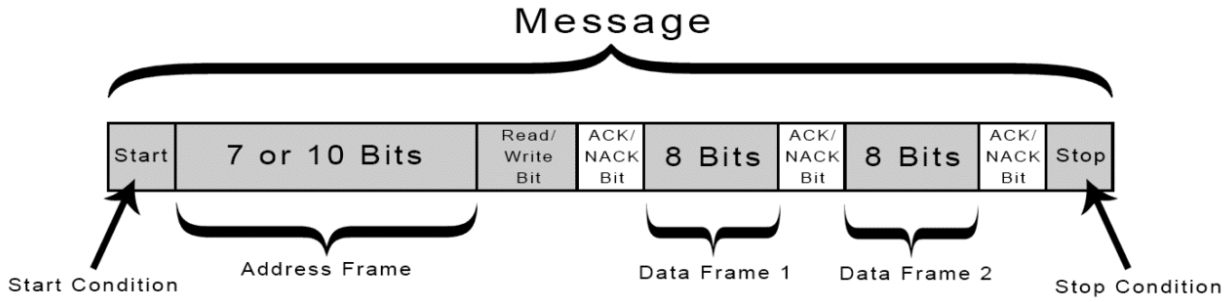

Fig5 I2C-data frame.

3)SPI

Serial Peripheral Interface (SPI) is an interface bus commonly used to send data between microcontrollers and small peripherals such as shift registers, sensors, and SD cards. It uses separate clock and data lines, along with a select line to choose the device you wish to talk to.SPI is a synchronous communication protocol that transmits and receives information simultaneously with high data transfer rates and is designed for board-level communication over short distances. The SPI communication interface is advantageous when needing to communicate between multiple devices.

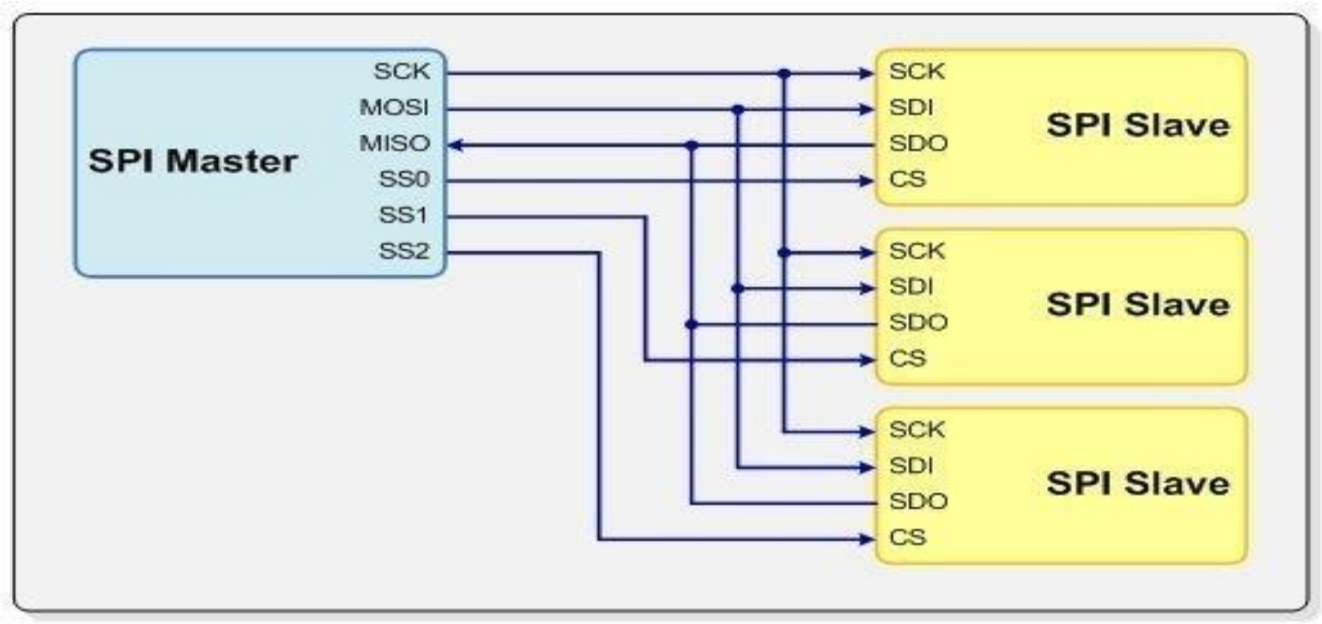

Fig6 SPI-Interface

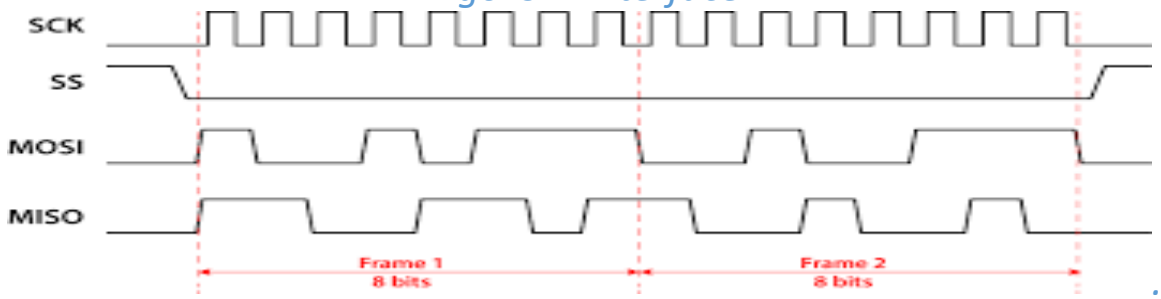

Fig7 SPI-data frame.

4)CAN

A controller area network (CAN bus) is a vehicle bus standard designed to allow microcontrollers and devices to communicate with each other. It is a message-based protocol, designed originally for multiplex electrical wiring within automobiles to save on copper, but it can also be used in many other contexts. For each device, the data in a frame is transmitted serially but in such a way that if more than one device transmits at the same time, the highest priority device can continue while the others back off. Frames are received by all devices, including by the transmitting device.

CAN has four frame types:

- Data frame: a frame containing node data for transmission.
- Remote frame: a frame requesting the transmission of a specific identifier.
- Error frame: a frame transmitted by any node detecting an error.
- Overload frame: a frame to inject a delay between data or remote frame.

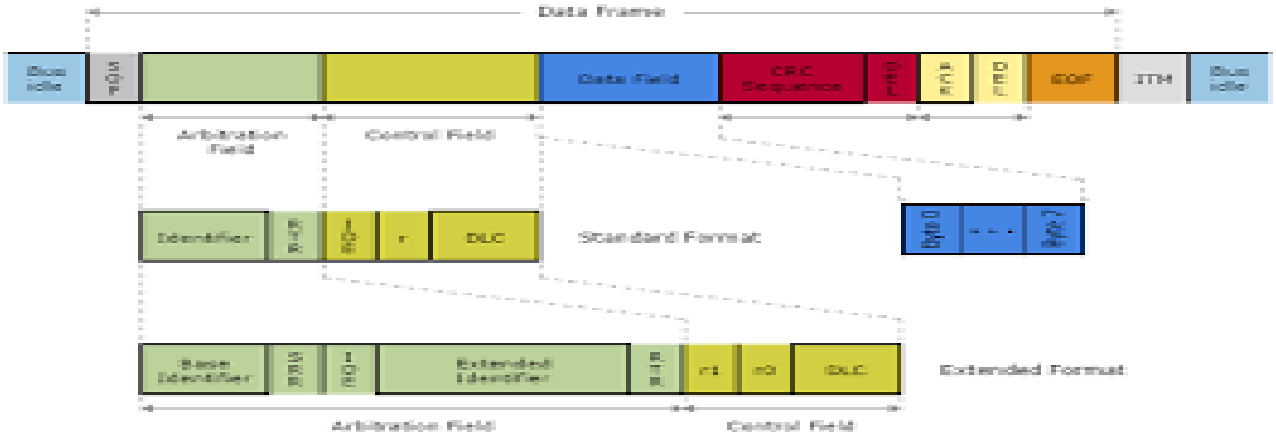

Fig8 CANI-Interface.

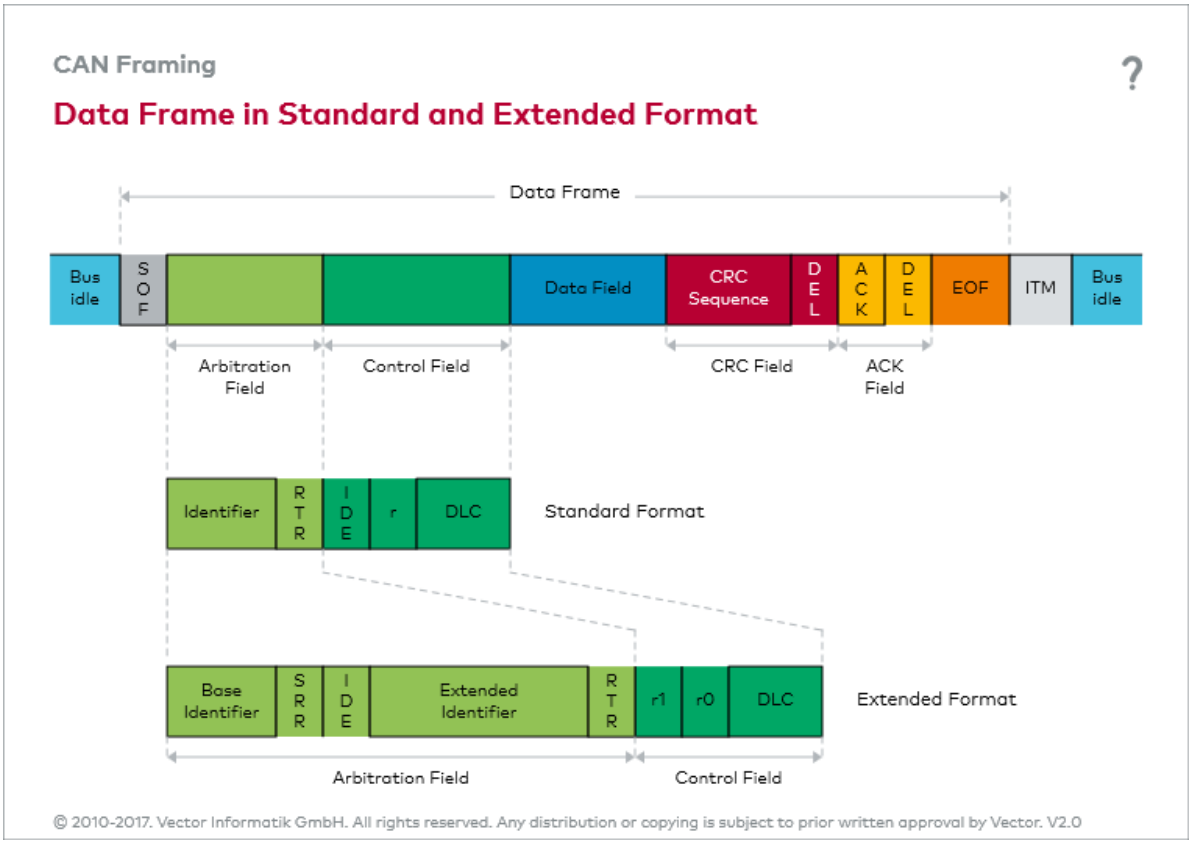

Fig9 CAN-data frame

	Duplex	Synchronous/ Asynchronous:	Master /Slave:	Arbitration:
USART	Full-duplex (simultaneous transmission and reception).	Can operate in both synchronous and asynchronous modes.	Can operate in both master and slave modes.	Typically doesn't have built-in arbitration mechanisms.
I2C	Half-duplex (alternating between transmission and reception, not simultaneous).	Synchronous (clock is shared between devices).	Master and slave devices.	Arbitration is used to resolve conflicts on the bus.
SPI	Full-duplex (simultaneous).	Synchronous.	Master and slave devices.	Generally does not have arbitration; typically, the master has full control.
CAN	Full-duplex.	Synchronous.	Master and slave devices, but often referred to as nodes.	Built-in arbitration to handle collisions on the bus.