Rugved Task-6

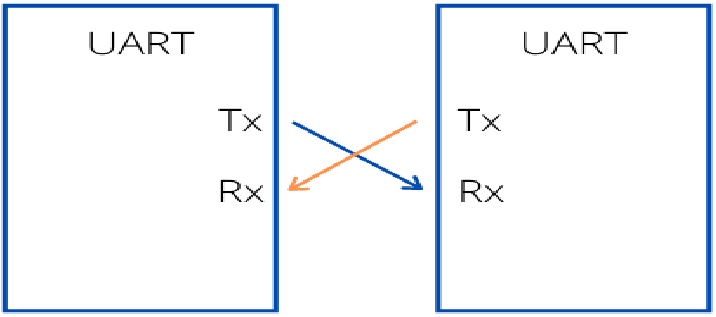
1. Inter-Systems (USB, UART)

a) Introduction

USB and UART are common communication interfaces utilized in embedded systems to send and receive data between devices. UART is one of the earliest communication interfaces applied to computers. The newer USB protocol, however, has replaced many UART systems, but both are still prevalent in the embedded industry. In this article, we will provide insight into both USB and UART communications and background on their relation.

i. UART, or Universal Asynchronous Receiver/Transmitter

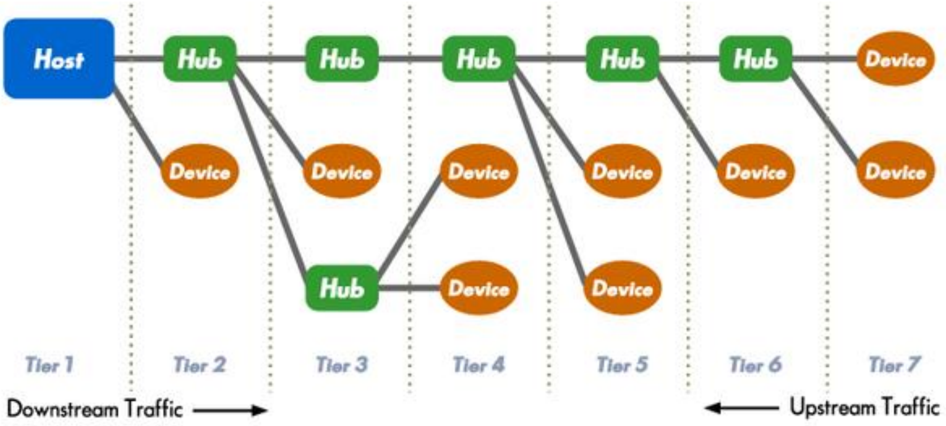
It is a hardware communication protocol that uses asynchronous serial communication with configurable speed. Asynchronous means it doesn’t use a clock signal to synchronize the output bits from the transmitting UART to the sampling bits on the receiving UART. UART is also considered to be “universal” because the parameters including transfer speed and data speed are configurable by the developer.
UART communication consists of two UART devices that communicate directly with each other. The UART on the sender device, or the transmitting UART, receives parallel data from the CPU (microprocessor or microcontroller) and converts it to serial data. This serial data is transmitted to the UART on the receiver device, or the receiving UART. The receiving UART converts the received serial data back to parallel data and sends it to the CPU.



ii. USB, or Universal Serial Bus,

It is a host-scheduled, token-based serial bus protocol that was designed to replace the multitude of cables and connectors required to connect peripheral devices to a host computer and instead implement a more standardized connection. USB addresses many of the requirements needed for this to occur including being able to self-identify on the bus and being hot-plugged for plug-and-play capabilities.

For USB communication, there is a USB host, or the master device, that initiates all communication that occurs over the bus. This is typically a computer or other controller. The peripheral device, or the slave device, is connected to the host and is programmed to provide the host with the information required to successfully operate.



b) Advantages and Disadvantages

Advantages of Using UART

- Simple to operate, well documented as it is a widely used method with a lot of resources online
- No clock needed
- Parity bit to allow for error checking

Disadvantages of Using UART

- Size of the data frame is limited to only 9 bits
- Cannot use multiple master systems and slaves
- Baud rates of each UART must be within 10% of each other to prevent data loss.
- Low data transmission speeds

Advantages of universal serial bus (USB)

- The universal serial bus is easy to use.
- It has robust connector system.
- It has low cost.
- It has variety of connector types and size available.
- It has true plug and play nature.
- It has **Low power consumption.**
- Daisy chain up to 127 USB components / peripherals at the same time to one PC.
- Fits almost all devices that have a USB port.

Disadvantages of universal serial bus (USB) are given below,

- It has limited capability and overall performance.
- Universal Serial Bus does not provide the broadcasting feature, only individual messages can be communicated between host and peripheral.
- The data transfer not as fast as some other systems.

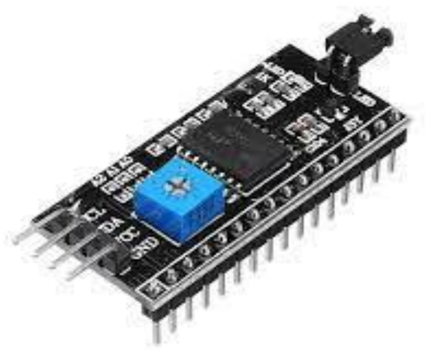
2. Intra-Systems (I2C, SPI, CAN)

a) Introduction

I. I2C

I2C stands for the inter-integrated controller. This is a serial communication protocol that can connect low-speed devices. It is a master-slave communication in which we can connect and control multiple slaves from a single master. In this, each slave device has a specific address.

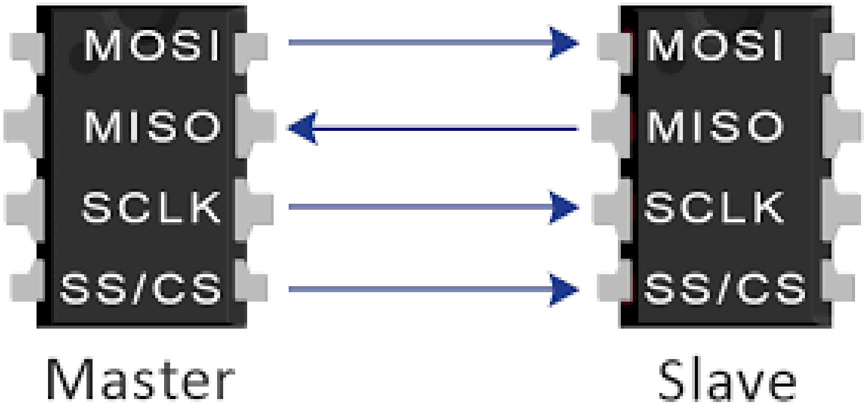
The I2C is developed to overcome the complexities of transmitting data through other communication protocols, including Universal Asynchronous Receiver Transmitter (UART) and Serial Peripheral Interface (SPI).



II. SPI

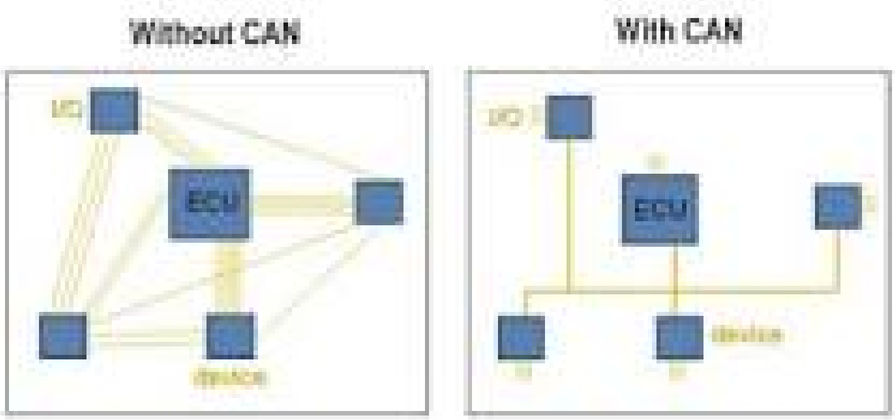
It stands for Serial Peripheral Interface (SPI). It is like I2C and it is a different form of serial-communications protocol specially designed for microcontrollers to connect.
Operates at full-duplex where data can be sent and received simultaneously.
Operate at faster data transmission rates = 8Mbits or more
It is typically faster than I2C due to the simple protocol. Even if data/clock lines are shared between devices, each device will require a unique address wire.

Used in places where speed is important. (eg. SD cards, display modules or when info updates and changes quickly like thermometers)



III. CAN

A controller area network (CAN) bus is a high-integrity serial bus system for networking intelligent devices. CAN busses and devices are common components in automotive and industrial systems. Using a CAN interface device, you can write LabVIEW applications to communicate with a CAN network. It is a message-based protocol designed to allow the Electronic Control Units (ECUs) found in today's automobiles, as well as other devices, to communicate with each other in a reliable, priority-driven fashion.



b) Advantages and Disadvantages

Advantages of using I2C

- Has a low pin/signal count even with numerous devices on the bus
- Flexible, as it supports multi-master and multi slave communication.
- Simple as it only uses 2 bidirectional wires to establish communication among multiple devices.
- Adaptable as it can adapt to the needs of various slave devices.
- Support multiple masters.

Disadvantages of using I2C

- Slower speed as it requires pull-up resistors rather than push-pull resistors used by SPI. It also has an open-drain design = limited speed.
- Requires more space as the resistors consume valuable PCB real estate.
- May become complex as the number of devices increases.

Advantages of using SPI

- The protocol is simple as there is no complicated slave addressing system like I2C.
- It is the fastest protocol compared to UART and I2C.
- No start and stop bits unlike UART which means data can be transmitted continuously without interruption
- Separate MISO and MOSI lines which means data can be transmitted and received at the same time

Disadvantages of using SPI

- More Pin ports are occupied, the practical limit to a number of devices.
- There is no flow control specified, and no acknowledgement mechanism confirms whether data is received unlike I2C
- Uses four lines – MOSI, MISO, NCLK, NSS
- No form of error check unlike in UART (using parity bit)
- Only 1 master

CAN Benefits

Low-Cost, Lightweight Network

CAN provides an inexpensive, durable network that helps multiple CAN devices communicate with one another. An advantage to this is that electronic control units (ECUs) can have a single CAN interface rather than analog and digital inputs to every device in the system. This decreases overall cost and weight in automobiles.

Broadcast Communication

Each of the devices on the network has a CAN controller chip and is therefore intelligent. All devices on the network see all transmitted messages. Each device can decide if a message is relevant or if it should be filtered. This structure allows modifications to CAN networks with minimal impact. Additional non-transmitting nodes can be added without modification to the network.

Priority

Every message has a priority, so if two nodes try to send messages simultaneously, the one with the higher priority gets transmitted and the one with the lower priority gets postponed. This arbitration is non-destructive and results in non-interrupted transmission of the highest priority message. This also allows networks to meet deterministic timing constraints.

Error Capabilities

The CAN specification includes a Cyclic Redundancy Code (CRC) to perform error checking on each frame's contents. Frames with errors are disregarded by all nodes, and an error frame can be transmitted to signal the error to the network. Global and local errors are differentiated by the controller, and if too many errors are detected, individual nodes can stop transmitting errors or disconnect itself from the network completely.

c) Applications

CAN Applications & Examples

CAN was first created for automotive use, so its most common application is in-vehicle electronic networking. However, as other industries have realized the dependability and advantages of CAN over the past 20 years, they have adopted the bus for a wide variety of applications. Railway applications such as streetcars, trams, undergrounds, light railways, and long-distance trains incorporate CAN. You can find examples of CAN devices linking the door units, brake controllers, passenger counting units, and more on different levels of the multiple networks within these vehicles. CAN also has applications in aircraft with flight-state sensors, navigation systems, and research PCs in the cockpit. In addition, you can find CAN buses in many aerospace applications, ranging from in-flight data analysis to aircraft engine control systems such as fuel systems, pumps, and linear actuators.

Medical equipment manufacturers use CAN as an embedded network in medical devices. In fact, some hospitals use CAN to manage complete operating rooms. Hospitals control operating room components such as lights, tables, cameras, X-ray machines, and patient beds with CAN-based systems. Lifts and escalators use embedded CAN networks, and hospitals use the CAN open protocol to link lift devices, such as panels, controllers, doors, and light barriers, to each other and control them. CAN open also is used in nonindustrial applications such as laboratory equipment, sports cameras, telescopes, automatic doors, and even coffee machines.

Applications of SPI

- Memory: SD Card, MMC, EEPROM, and Flash.
- Sensors: Temperature and Pressure.
- Control Devices: ADC, DAC, digital POTS, and Audio Codec.
- Others: Camera Lens Mount, Touchscreen, LCD, RTC, video game controller, etc.

Why use I2C?

- The I2C bus is currently still a common communication peripheral used by various circuits and is simple to implement. No matter how many devices are connected to the bus, only two signal lines (clock SCL and data SDA) are needed.

- It is a true multi-master bus, which is superior to SPI.
- In addition, the I2C interface is also flexible which allows it to communicate with slow devices while also having high-speed modes to transmit large data. The transmission rate can reach 100kbit/s in standard mode, 400kbit/s in fast mode, and 3.4Mbit/s in high-speed mode;
- Because of how flexible it is, I2C will always remain as one of the best communication peripherals to connect devices.