I2C Communication Protocol

I2C stands for **Inter-Integrated Circuit.** It is a bus interface connection protocol incorporated into devices for serial communication. It was originally designed by Philips Semiconductor in 1982. Recently, it is a widely used protocol for short-distance communication. It is also known as Two Wired Interface(TWI).

Working of I2C Communication Protocol :

It uses only 2 bi-directional open-drain lines for data communication called SDA and SCL. Both these lines are pulled high.

Serial Data (SDA) – Transfer of data takes place through this pin.
Serial Clock (SCL) – It carries the clock signal.
I2C operates in 2 modes –

- Master mode
- Slave mode

Each data bit transferred on SDA line is synchronized by a high to the low pulse of each clock on the SCL line.

According to I2C protocols, the data line can not change when the clock line is high, it can change only when the clock line is low. The 2 lines are open drain, hence a pull-up resistor is required so that the lines are high since the devices on the I2C bus are active low. The data is transmitted in the form of packets which comprises 9 bits. The sequence of these bits are

1. **Start Condition** – 1 bit
2. **Slave Address** – 8 bit
3. **Acknowledge** – 1 bit

I2C Packet Format :

In the I2C communication protocol, the data is transmitted in the form of packets. These packets are 9 bits long, out of which the first 8 bits are put in SDA line and the 9th bit is reserved for ACK/NACK i.e. Acknowledge or Not Acknowledge by the receiver.

START condition plus address packet plus one more data packet plus STOP condition collectively form a complete Data transfer.

Features of I2C Communication Protocol :

- **Half-duplex Communication Protocol –**
  Bi-directional communication is possible but not simultaneously.

- **Synchronous Communication** –
  The data is transferred in the form of frames or blocks.
- Can be configured in a multi-master configuration.
- **Clock Stretching –**
  The clock is stretched when the slave device is not ready to accept more data by holding the SCL line low, hence disabling the master to raise the clock line. Master will not be able to raise the clock line because the wires are AND wired and wait until the slave releases the SCL line to show it is ready to transfer next bit.
- **Arbitration** –
  I2C protocol supports multi-master bus system but more than one bus can not be used simultaneously. The SDA and SCL are monitored by the masters. If the SDA is found high when it was supposed to be low it will be inferred that another master is active and hence it stops the transfer of data.
- **Serial transmission –**
  I2C uses serial transmission for transmission of data.
- Used for low-speed communication.

Advantages :

- Can be configured in multi-master mode.
- Complexity is reduced because it uses only 2 bi-directional lines (unlike SPI Communication).
- Cost-efficient.
- It uses ACK/NACK feature due to which it has improved error handling capabilities

Disadvantages :

- 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.

Limitations :

- Slower speed.
- Half-duplex communication is used in the I2C communication protocol.

Pull up resistor :

A pull-up resistor connects unused input pins (AND and NAND gates) to the dc supply voltage, (Vcc) to keep the given input HIGH. Pull-up resistors are fixed value resistors used between the connection of a voltage supply and a particular pin in a digital logic circuit. More commonly paired with switches, its purpose is to ensure the voltage between Ground and Vcc is actively controlled when the switch is open.

Pull down resistor :

A pull-down resistor connects unused input pins (OR and NOR gates) to ground, (0V) to keep the given input LOW. Pull-down resistors ensure the voltage between VCC and a microcontroller pin is actively controlled when the switch is open. However, instead of pulling a pin to a high value, such resistors pull the pin to a low valued instead.

Open drain:

An open-drain or open-collector output pin is driven by a single transistor, which pulls the pin to only one voltage (generally, to ground). When the output device is off, the pin is left floating (open, or hi-z).

Active low:

A signal is 'active low' means that signal will be performing its function when its logic level is 0. If it's anactive-low pin, you must “pull” that pin LOW by connecting it to ground.

Active high:

A signal is 'active high' means that signal will be performing its function when its logic level is 1. For an active high pin, you connect it to your HIGH voltage (usually 3.3V/5V).

Linux Booting Process

An operating system (OS) is the low-level software that manages resources, controls peripherals, and provides basic services to other software. In Linux, there are 6 distinct stages in the typical booting process.

1. BIOS

BIOS stands for Basic Input/Output System. In simple terms, the BIOS loads and executes the Master Boot Record (MBR) boot loader.

When you first turn on your computer, the BIOS first performs some integrity checks of the HDD or SSD.

Then, the BIOS searches for, loads, and executes the boot loader program, which can be found in the Master Boot Record (MBR). The MBR is sometimes on a USB stick or CD-ROM such as with a live installation of Linux.

Once the boot loader program is detected, it's then loaded into memory and the BIOS gives control of the system to it.

2. MBR

MBR stands for Master Boot Record, and is responsible for loading and executing the GRUB boot loader.

The MBR is located in the 1st sector of the bootable disk, depending on your hardware. The MBR also contains information about GRUB, or LILO in very old systems.

3. GRUB

Sometimes called GNU GRUB, which is short for GNU GRand Unified Bootloader, is the typical boot loader for most modern Linux systems.

The GRUB splash screen is often the first thing you see when you boot your computer. It has a simple menu where you can select some options. If you have multiple kernel images installed, you can use your keyboard to select the one you want your system to boot with. By default, the latest kernel image is selected.

The splash screen will wait a few seconds for you to select and option. If you don't, it will load the default kernel image.

4. Kernel

The kernel is often referred to as the core of any operating system, Linux included. It has complete control over everything in your system.

In this stage of the boot process, the kernel that was selected by GRUB first mounts the root file system that's specified in the grub.conf file. Then it executes the /sbin/init program, which is always the first program to be executed. You can confirm this with its process id (PID), which should always be 1.
The kernel then establishes a temporary root file system using Initial RAM Disk (initrd) until the real file system is mounted.

5.Init

At this point, your system executes runlevel programs. At one point it would look for an init file, usually found at /etc/inittab to decide the Linux run level.

6. Runlevel programs

Depending on which Linux distribution you have installed, you may be able to see different services getting started. For example, you might catch starting sendmail …. OK.
These are known as runlevel programs, and are executed from different directories depending on your run level. Each of the 6 runlevels described above has its own directory.

Zephyr RTOS

Zephyr is a small real-time operating system (RTOS) for connected, resource-constrained and embedded devices (with an emphasis on microcontrollers) supporting multiple architectures and released under the Apache License 2.0. Zephyr includes a kernel, and all components and libraries, device drivers, protocol stacks, file systems, and firmware updates, needed to develop full application software.

Zephyr intends to provide all components needed to develop resource-constrained and embedded or microcontroller-based applications. This includes, but is not limited to

- A small kernel
- A flexible configuration and build system for compile-time definition of required resources and modules.
- A set of protocol stacks (IPv4 and IPv6, Constrained Application Protocol (CoAP), LwM2M, MQTT, 802.15.4, Thread, Bluetooth Low Energy, CAN).
- A virtual file system interface with several flash file systems for non-volatile storage (FATFS, LittleFS, NVS.)
- Management and device firmware update mechanisms.