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

Start and Stop Conditions :

START and STOP can be generated by keeping the SCL line high and changing the level of SDA. To generate START condition the SDA is changed from high to low while keeping the SCL high. To generate STOP condition SDA goes from low to high while keeping the SCL high, as shown in the figure below.

Repeated Start Condition :

Between each start and stop condition pair, the bus is considered as busy and no master can take control of the bus. If the master tries to initiate a new transfer and does not want to release the bus before starting the new transfer, it issues a new START condition. It is called a REPEATED START condition.

Read/Write Bit :

A high Read/Write bit indicates that the master is sending the data to the slave, whereas a low Read/Write bit indicates that the master is receiving data from the slave.

ACK/NACK Bit :

After every data frame, follows an ACK/NACK bit. If the data frame is received successfully then ACK bit is sent to the sender by the receiver.

Addressing :

The address frame is the first frame after the start bit. The address of the slave with which the master wants to communicate is sent by the master to every slave connected with it. The slave then compares its own address with this address and sends ACK.

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.

Limitations :

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

Pull-up Resistors

The most common method of ensuring that the inputs of digital logic gates and circuits can not self-bias and float about is to either connect the unused pins directly to ground (0V) for a constant low “0” input, (OR and NOR gates) or directly to Vcc (+5V) for a constant high “1” input (AND and NAND gates).

Pull-down Resistors

A Pull-down resistor works in the same way as the previous pull-up resistor, except this time the logic gates input is tied to ground, logic level “0” (LOW) or it may go HIGH by the operation of a mechanical switch. This pull-down resistor configuration is particularly useful for digital circuits like latches, counters and flip-flops that require a positive one-shot trigger when a switch is momentarily closed to cause a state change.

opendrain

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.

A common example is an n-channel transistor which pulls the signal to ground when the transistor is on or leaves it open when the transistor is off.

When the transistor is off, the signal can be driven by another device or it can be pulled up or down by a resistor. The resistor prevents an undefined, floating state.

Active low, Active high

Simply put, this just describes how the pin is activated. If it's an active-low pin,**you must "pull" that pin LOW by connecting it toground.** For an active high pin, you connect it to your HIGH voltage (usually 3.3V/5V).

For example, let's say you have a shift register that has a chip enable pin, CE.

Linux booting process

Booting a Linux system involves different components and tasks. The hardware itself is initialized by the BIOS or the UEFI, which starts the kernel by means of a boot loader. After this point, the boot process is completely controlled by the operating system and handled by system .

Linux role of kernel

The Linux kernel is the main component of a Linux operating system and is the core interface between a computer's hardware and its processes. It communicates between the 2, managing resources as efficiently as possible.

First impression on zephyr RTO

- A small kerneL
- A flexible configuration and build system for compile-time definition of required resources and modules
- A set of protocol stacks
- A virtual file system interface with several flash file systems for non-volatile storage
- Management and device firmware update mechanisms

---

regards

-A.Arunmani.